MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS – SETIF
UFAS (ALGERIE)

MEMOIRE

Présenté à la Faculté des Sciences de L'ingénieur

Département d'électronique

Pour l'Obtention du Diplôme de

MAGISTER

Option : Contrôle

Par

Mr DRIF MAKHLOUF

Thème :

Etude de performance des architectures parallèles dédiées aux filtres numériques

Soutenu le : / / devant la commission d'examen :

Dr : BEKKA Raïs Elhadi	**Prof à l'université de Sétif**	**Président**
Dr : CHIKOUCHE Djamel	**Prof à l'université de M'sila**	**Rapporteur**
Dr : KHENFER Nabil	**Prof à l'université de Sétif**	**Examinateur**
Dr : DJAHLI Farid	**Prof à l'université de Sétif**	**Examinateur**
Dr : FERHAT HAMIDA Abdelhak	**MC à l'université de Sétif**	**Examinateur**

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

REMERCIEMENTS

Je tiens à remercier vivement Prof. Djamel CHIKOUCHE qui m'a proposé ce travail et suivi dans sa réalisation. Je tiens également à la remercier pour les conseils qu'il m'a prodigués et la patience dont il a fait preuve, qui ont été d'un apport précieux pour l'accomplissement de ce travail.

J'adresse mes vifs remerciements à monsieur le président du jury, professeur BEKKA Raïs Elhadi, ainsi qu'à ses membres, Prof. N. KHENFER, Prof. F. DJAHLI, et Dr. A. FERHAT HAMIDA, qui m'ont fait l'honneur d'accepter de juger ce travail.

Je tiens à remercier Prof. MERZOUKI A/Aziz directeur du laboratoire de recherche L.I.S, qui nous a permis d'utiliser les moyens du laboratoire pour réaliser ce travaille de magister.

Je témoigne aussi ma profonde gratitude aux enseignants qui ont participé à notre formation et à tous ceux qui m'ont aidé.

Je prie le personnel du département d'électronique et de la bibliothèque, de trouver dans ce passage l'expression de ma profonde gratitude pour leur sympathie et assistance.

Enfin, je remercie infiniment tous mes amis et en particulier ceux de l'institut d'électronique.

Drif Makhlouf

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Résumé

Ce travail de magister présente une étude sur la représentation des filtres numériques RIF par la courte convolution rapide. Il consiste également à implémenter les algorithmes parallèles de filtre numérique RIF, sur l'architecture systolique et étudier leur performance par rapport à l'architecture VLSI de Parhi.

Les tests ont révélé la complexité des structures VLSI de Parhi durant la phase d'implémentation. Nous avons alors proposé l'implémentation des filtres RIF sur l'architecture systolique afin d'améliorer la complexité des structures de filtrage RIF.

La procédure d'implémentation est basée sur un algorithme itératif de filtrage qui effectue la multiplication des matrices.

Les tests de simulation de l'architecture systolique démontrent ses bonnes performances en terme de temps d'exécution et complexité de mise en œuvre.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Sommaire

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Chapitre II - La Courte Convolution Itérative (ISC) : Application à la représentation d'un filtre RIF

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Chapitre III : Implémentation des filtres RIF sur les réseaux systoliques

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

INTRODUCTION GENERALE

Un filtre numérique peut être implémenté sur un ordinateur à usage général, ou sur un microprocesseur soigneusement programmé. On peut également réaliser un filtre numérique construit spécifiquement pour appliquer l'algorithme de filtrage désiré au signal d'entrée. Toutefois, la vitesse de traitement maximale proche de celle des filtres analogiques ne peut être atteinte que par les circuits numériques dédiés conçus spécifiquement pour réaliser un seul algorithme de filtrage. Les applications de traitement de signaux en temps réel peuvent être actuellement satisfaites efficacement par la technologie VLSI d'intégration à très grande échelle et à un coût raisonnable [**1, 2, 3,4 ,5**]. Le réseau processeur VLSI, ayant un grand nombre d'éléments processeurs (PE) régulièrement connectés entre eux et utilisant un traitement concurrent, est devenu une architecture très attirante pour la réalisation des systèmes de traitement du signal[**6, 7**]. Parmi les réseaux processeurs VLSI, la structure systolique a reçu une attention particulière ces dernières années pour les bonnes performances qu'elle offre en terme de temps d'exécution et complexité de mise en œuvre[**8, 9,10,11,12**].

Des progrès considérables ont été réalisés pour le développement de structures parallèles destinées aux calculs les plus complexes tels que : la convolution, la corrélation, la transformée de Fourier rapide, la triangularisation [**13, 14, 15, 16**]. Le produit matriciel, opérateur fondamental de la plupart des algorithmes de calculs linéaires, a été intensivement étudié dans les applications de réseaux de processeurs. La majorité des algorithmes de traitement du signal utilise des opérations matrice/vecteur qui peuvent être effectuées à l'aide d'architectures parallèles [**3,4,12,17,18**]. La vitesse de ces opérations matrice/vecteur est proportionnelle à la dimension des matrices. Pour une matrice N x N, une opération matrice/vecteur est proportionnelle à la dimension des matrices. Pour une matrice N x N, une opération matrice/vecteur, utilisant N multiplieurs en parallèle, peut être

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

réalisée en $O(N)$ unités de temps. Malheureusement, pour les grandes valeurs de N, cette vitesse de calcul n'est pas toujours adaptée aux applications en temps réel [**12, 19,20]**.

L'objectif de ce travail consiste à adapter des architectures systoliques, aussi performantes que possible, pour la réalisation des filtres numériques RIF qui nécessitent des calculs non récursifs. Les architectures parallèles dédiées aux filtres RIF ont fait l'objet d'un grand nombre de travaux publiées par Parhi et autres ces dernières années [**1, 4, 9, 21, 22, 23,24**].

Dans ce travail, nous utilisons une autre méthodologie de mise en œuvre des filtres non récursifs basée sur les réseaux systoliques qui permettent de calculer le produit d'une matrice par une colonne de l'algorithme de filtrage demandant moins de calculs.

Le mémoire s'articule autour de trois chapitres. Le premier chapitre présente une description sur les différentes méthodes de synthèse des filtres numériques RIF.

Le second chapitre est réservé à l'étude de la technique d'implémentation de la convolution rapide par différents algorithmes de calcul et l'utilisation de la technologie VLSI selon la technique de Parhi.

Le troisième chapitre illustre le principe d'application des réseaux systoliques conventionnels à l'implémentation des filtres non récursifs représentés sous forme matricielle, le principe d'implantation VLSI des processeurs élémentaires du réseau et l'estimation du temps d'exécution. L'objectif de ce chapitre consiste à déterminer les architectures les plus performantes en termes du temps d'exécution, et de la complexité de la mise en œuvre.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Chapitre I

Les filtres numériques RIF et leur méthode de synthèse

Introduction

Le traitement numérique du signal est une discipline scientifique destinée à l'étude des systèmes et signaux par rapport aux contraintes imposées par la machinerie de calcul numérique. Il est devenu un élément essentiel de la technologie moderne surtout dans des domaines comme le filtrage, la communication de données numériques, le traitement d'images, la biomédecine, la parole, le radar, le sonar, et encore d'autres disciplines**[25]**.

On distingue deux grandes classes de filtres numériques : Les filtres à réponse impulsionnelle finie (RIF) et les filtres à réponse impulsionnelle infinie (RII). Un filtre RIF répond toujours à une impulsion appliquée à son entrée par un signal de sortie de durée finie, c'est-à-dire formé d'un nombre fini de valeurs numériques. Les filtre RIF sont stables et présentent l'avantage de posséder une phase exactement linéaire. Par contre, ils ont l'inconvénient de nécessiter un grand nombre de multiplications comparées aux filtres RII qui satisfont des spécifications similaires. Mais, le progresse dans la technologie des semi-conducteurs a rendu l'opération d'implantation matérielle directe plus rapide et moins chère. C'est pourquoi, les filtres RIF ont pris de l'importance ces dernières années. Dans l'implantation matérielle directe, la précision dans la représentation des coefficients du filtre par différentes méthodes et techniques est nécessaire **[26, 27, 28, 29]**.

Il existe plusieurs méthodes permettant de déterminer les valeurs des coefficients de façon appropriée pour satisfaire la réponse fréquentielle spécifiée. Parmi ces techniques, on peut citer la synthèse par série de Fourier, l'échantillonnage en fréquence **[25,29]**, la méthode des fenêtres **[26, 29,25]**, la transformation Mc-Clellan **[27, 30,31]**, l'approximation numérique, etc.

En général, un filtre RIF peut être réalisé soit sous forme récursive (avec boucle de retour) soit sous forme non récursive. Les RIF non récursifs sont stables, ont une architecture simple et sont insensibles aux conditions extérieures (chaleur, humidité, etc.). Le grand avantage des structures RIF est leur capacité d'engendrer une phase linéaire dans leur comportement fréquentielle. Cette propriété est d'un grand usage dans les domaines d'applications, comme :

- Le traitement de la parole,
- Le traitement d'images,

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

- L'égalisation de phase pour les systèmes de communications numériques.

Toutefois, on peut également signaler certains inconvénients liés au filtre RIF. Le plus important de ses inconvénients est son incapacité à engendrer une réponse fréquentielle en amplitude sélective, à moins que l'ordre du filtre soit très élevé.

Dans ce chapitre, nous allons présenter les notions fondamentales et les caractéristiques des filtres RIF à phase linéaire, ainsi que les différentes méthodes de synthèse des filtres RIF.

I.1. Caractéristiques des filtres RIF à phase linéaire [25, 31, 32,33]

Soit *{h(n)}* une séquence causale à durée finie définie sur l'intervalle $0 \leq n \leq N$-1.

La transformée de Fourier de *{h(n)}* est*:*

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n} \tag{I.1}$$

Posons $$H(e^{j\omega}) = \left|H(e^{j\omega})\right| . e^{-j\theta(\omega)} \tag{I.2}$$

Avec $\left|H(e^{j\omega})\right|$ et $\theta(\omega)$ respectivement l'amplitude et la phase de $H(e^{jw})$.

Pour qu'un filtre RIF ait une phase exactement linéaire, il faut que la phase $\theta(\omega)$ soit de la forme suivante :

$$\theta(\omega) = -\alpha.\omega \quad \text{avec} -\pi < \omega < \pi \tag{I.3}$$

òu α est un retard de groupe constant.

En combinant (I.1) et (I.2), nous avons

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n} = \left|H(e^{j\omega})\right| e^{-j\theta(\omega)} \tag{I.4}$$

Nous obtenons

$$\tan(\alpha.\omega) = \frac{\sum_{n=0}^{N-1} h(n).\sin \omega n}{h(0) + \sum_{n=1}^{N-1} h(n).\cos \omega n} \tag{I.5}$$

Il existe deux solutions possibles :

- $\alpha = 0$: qui entraîne que *h(0)* est arbitraire et *h(n) =0* pour $n \neq 0$, résultat peu utile.
- $\alpha \neq 0$ *:* qui entraîne que l'équation (I.5) s'écrit

$$\sum_{n=0}^{N-1} h(n).\sin(\alpha - n) = 0 \tag{I.6}$$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

La solution de (I.6) si elle existe est unique et est de la forme $\alpha = (N\text{-}1)/2$ *et* $h(n) = h(N\text{-}1\text{-}n)$ *avec* $0 \leq n \leq N\text{-}1$.

Remarque :

Pour chaque cas *N*, il y a une seule valeur de retard de groupe '$\alpha=(N\text{-}1)/2$', la réponse impulsionnelle doit être d'une symétrie spéciale.

- Si '*N*'est *impaire*, 'α' est *entier*, alors, le centre de symétrie de la réponse impulsionnelle du filtre à phase linéaire est l'échantillon du milieu (figure.I.1).
- Si '*N*'est *paire*, alors, le centre de symétrie de la réponse impulsionnelle du filtre à phase linéaire est le milieu entre deux échantillons (figure.I.2).

On dit qu'un filtre est à phase exactement linéaire, s'il possède un retard de groupe constant et un retard de phase constant. Si on désire seulement que le retard de groupe soit constant (comme c'est souvent la cas), alors le filtre à phase linéaire se définit comme suit

$$H(e^{j\omega}) = \left|H(e^{j\omega})\right| e^{-j(\beta-\alpha\omega)} \qquad \text{(I.7)}$$

Les seules nouvelles solutions pour *{h(n)}*, α et β sont :

$$\alpha = (N\text{-}1)/2,\ \beta = \pm\pi/2 \quad et \quad h(n) = -h(N\text{-}1\text{-}n)\ avec\ 0 \leq n \leq N\text{-}1 \qquad \text{(I.8)}$$

Les filtres solutions ont un retard de $(N\text{-}1)/2$ échantillons avec des réponses impulsionnelles antisymétriques autour du centre de la séquence opposée à la séquence réelle à phase linéaire.

- Pour *N* impaire, $h(N\text{-}1)/2$ doit égale à 0. Le centre de symétrie de la réponse impulsionnelle du filtre à phase linéaire est l'échantillon du milieu égal à 0 (figure.I.3).
- Pour *N* paire, le centre de symétrie de la réponse impulsionnelle du filtre à phase linéaire est le milieu entre deux échantillons (figure.I.4).

Dépendant de la valeur de *N (paire ou impaire)* et du type de la symétrie de la séquence à réponse impulsionnelle (symétrique ou antisymétrique), il existe quatre cas possibles des filtres RIF à phase linéaire **[25,34]**.

I.2. Réponse en fréquence des filtres RIF à phase linéaire

Soit la réponse impulsionnelle (I.7), nous posons :

$H^*(e^{j\omega}) = \left|H(e^{j\omega})\right|$. α, β sont définies suivant l'équation (I.8).

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$H(e^{j\omega})$ peut être exprimée en terme des coefficients de la réponse impulsionnelle pour chacun des quatre cas du filtre à phase linéaire.

- cas 1 *(réponse impulsionnelle symétrique et symétrie impaire) :*

$$H(e^{j\omega}) = e^{-j\omega((N-1)/2)} \sum_{n=0}^{N-1} a(n)\cos \omega n \qquad \text{(I.9)}$$

avec $a(0)=h((N-1)/2)$ et $a(n) = 2h((N-1)/2-n)$ pour $n=1\ldots\ldots(N-1)/2$

- cas 2 *(réponse impulsionnelle symétrique et symétrie paire) :*

$$H(e^{j\omega}) = e^{-j\omega((N-1)/2)} \sum_{n=0}^{N-1} b(n)\cos \omega(n-1/2) \qquad \text{(I.10)}$$

avec $b(n) = 2h(N/2-n)$ *pour* $n=1\ldots\ldots(N)/2$

à $\omega = \pi$ nous avons $H^*(e^{jw}) = 0$, ce qui signifie que les filtres passe- haut ne peuvent pas être évalués avec ce type de filtre.

- cas 3 *(réponse impulsionnelle antisymétrique et symétrie impaire) :*

$$H(e^{j\omega}) = e^{-j\omega((N-1)/2)} e^{j\pi/2} \sum_{n=0}^{N-1} c(n).\cos \omega n \qquad \text{(I.11)}$$

avec $c(n) = 2h((N-1)/2-n)$ pour $n=1\ldots\ldots(N-1)/2$

Ce filtre à phase linéaire possède une réponse en fréquence imaginaire.

- cas 4 *(réponse impulsionnelle antisymétrique et symétrie paire) :*

$$H(e^{j\omega}) = e^{-j\omega((N-1)/2)} e^{j\pi/2} \sum_{n=0}^{N-1} d(n)\sin \omega(n-1/2) \qquad \text{(I.12)}$$

avec $d(n) = 2h(N/2-n)$ *pour* $n=1\ldots\ldots N/2$

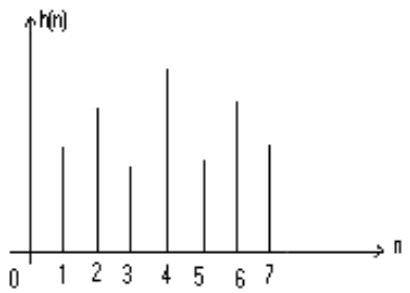


Fig.I.1. Réponse impulsionnelle symétrique et symétrie impaire.

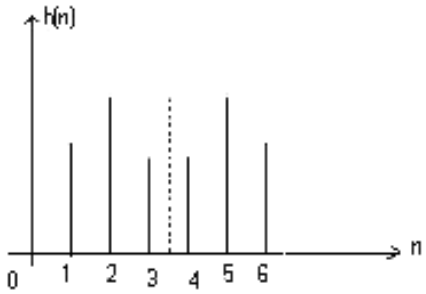


Fig.I.2. Réponse impulsionnelle symétrique et symétrie paire.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

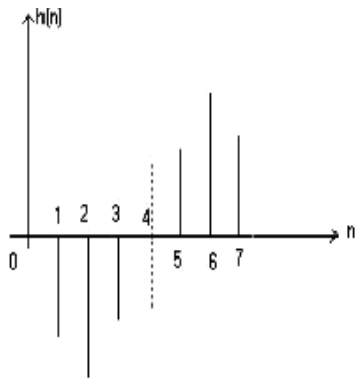


Fig.I.3. Antisymétrique et symétrie impaire.

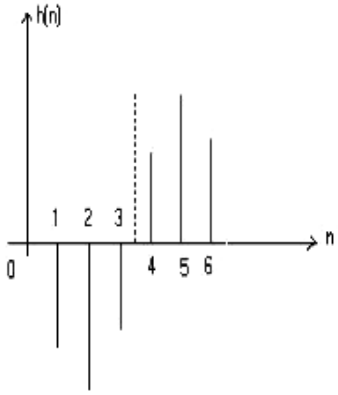


Fig.I.4. Antisymétrique et symétrie paire.

Les figures (I.1., I.2., 1.3. *et* I.4) représentent la réponse impulsionnelle de chaque cas étudié pour les filtres RIF à phase linéaire. Le trait en pointillé représente l'axe de symétrie. La répartition des coefficients est relative au cas considéré.

En pratique, on impose que la réponse impulsionnelle *h(n)* soit réelle, ce qui veut dire que la partie réelle de *H(ω)* doit être paire, tandis que sa partie imaginaire doit être impaire. La contrainte de linéarité par la phase *θ(ω)* conduit à quatre types possibles de filtres RIF.

Les propriétés des filtres RIF sont résumées dans le Tableau.I.1 **[25]**.

RIF	*N*	*h (n)*	$H_0(\omega)$	$\theta(\omega)$	$H(\omega)$ ou $H_0(\omega)$	
					$\omega = 0$	$\omega = \pi$
1	impair	Symétrique $h[n] = h[N-1-n]$	$\sum_{n=0}^{(N-1)/2} a_n.\cos[n\omega]$	$\frac{N-1}{2}\omega$	Sans contr	Sans contr
2	pair	Symétrique $h[n] = h[N-1-n]$	$\sum_{n=1}^{N/2} b_n.\cos[(n-1/2)\omega]$	$-\frac{N-1}{2}\omega$	Sans contr	0
3	impair	Anti-symétrique $h[n] = -h[N-1-n]$ $h[(N-1)/2] = 0$	$\sum_{n=0}^{(N-1)/2} a_n.\sin[n\omega]$	$\pi/2-[(N-1)/2]\omega$	0	0
4	pair	Anti-symétrique $h[n] = -h[N-1-n]$	$\sum_{n=1}^{N/2} b_n.\sin[(n-1/2)\omega]$	$\pi/2-[(N-1)/2]\omega$	0	Sans contr

Tableau.I.1. Propriétés des filtres RIF.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Les coefficients a_i et b_i sont donnés par:

$$a_0 = h(\frac{N-1}{2}),\ a_n = 2.h(\frac{N-1}{2}) - n\ ,\ n = 1, \frac{N-1}{2}\ ,\quad b_n = 2.h(\frac{N}{2} - n),\quad n = 1, \frac{N}{2}$$

Les filtres de type 1 et 2 sont généralement utilisés pour des applications de filtrage à proprement parlé, alors que les modèles 3 et 4 sont souvent utilisés pour des applications de filtrage avec déphasage (intégrateur, différentiateur, transformateur de Hilbert) **[25]**.

Conformément au tableau.I.1, on peut réécrire $H(\omega)$ du filtre de type 1 sous la forme *:*

$$H(\omega) = \sum_{n=0}^{N-1} h(n) e^{-jn\omega} = e^{-j\frac{N-1}{2}\omega} \sum_{n=0}^{\frac{N-1}{2}} a_n \cos(n\omega) = e^{-j\frac{N-1}{2}} M(\omega) \qquad \text{(I.13)}$$

Ces filtres conservent leur stabilité pour tout jeu de coefficients, ce qui les rend très adéquats pour la réalisation des systèmes adaptatifs. Au niveau de synthèse, il est à noter également qu'il est possible d'obtenir facilement des filtres RIF d'ordre élevé par combinaison de plusieurs cellules élémentaires **[25,35]**.

N.B : Si *N -1* est l'ordre du filtre.

N : la dimension du filtre (longueur) ou nombre de coefficients.

La conception d'un filtre RIF comporte trois grandes phases **[25]**.

- L'élaboration des spécifications du filtre.
- La détermination des paramètres du filtre pour satisfaire ces spécifications (la synthèse).
- La réalisation du filtre par une structure spécifique.

I.3. Méthodes de synthèse

Les méthodes de conception d'un filtre RIF peuvent être regroupées en trois classes **[25, 27,29]**:

- La méthode des fenêtres.
- La méthode de l'échantillonnage en fréquence.
- Les méthodes d'optimisation.

I.3.1. La méthode des fenêtres [25, 26,28]

Dans ce chapitre, nous allons décrire deux techniques de synthèse des filtres RIF à phase linéaire : la méthode des fenêtres et celle dite des bandes étroites. Cette dernière est une méthode simple et rapide. Elle a été proposée en 1988 par P.Janardhanan et M.N. Neelakantan. **[26]**.

I.3.2. Synthèse par la méthode des fenêtres

I.3.2.1 Généralité

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Sachant que la fonction de transfert (réponse en fréquence) d'un filtre numérique $H(\omega)$ est une fonction périodique de période 2π, une méthode immédiate pour le calcul d'un filtre non récursif consiste en un simple développement en séries de Fourier. A partir des spécifications $H_d(\omega)$ du filtre désiré, on détermine la réponse impulsionnelle $h_d(n)$ du filtre par **[25,28]** :

$$h_d(n) = (1/2\pi)\int_{-\pi}^{\pi} H_d(\omega)e^{j\omega n}d\omega \qquad \text{(I.14)}$$

La relation liant $H_d(\omega)$ à $h_d(n)$ est donné par :

$$H_d(\omega) = \sum_{n=0}^{\infty} h_d(\omega)e^{-jwn} \qquad \text{(I.15)}$$

L'évaluation de l'intégrale (I.14) permet d'obtenir la réponse impulsionnelle $h_d(n)$ du filtre. Cette dernière sera d'ordre infinie $(n=0 \text{ à } \infty)$. Il est nécessaire, pour rendre ce filtre utilisable, de la tronquer à un ordre fini N. Le principe de troncature est équivalent à multiplier la réponse impulsionnelle $h_d(n)$ par la fenêtre rectangulaire définie par :

$$W(n) = \begin{cases} 1 & pour \quad n = 0,1,...,N-1 \\ 0 & ailleurs \end{cases} \qquad \text{(I.16)}$$

L'expression de la réponse impulsionnelle devient :

$$h(n) = h_d(n)w(n) = \begin{cases} h_d(n) & pour \quad n = 0,1,...,N-1 \\ 0 & ailleurs \end{cases} \qquad \text{(I.17)}$$

Cette troncature se traduit pour la réponse en fréquence $H(\omega)$ du filtre par la convolution de $H_d(\omega)$ par $W(\omega)$ donnée par [1] :

$$H(\omega) = (1/2\pi)\int_{-\pi}^{\pi} H_d(\gamma)W(\omega-\gamma)d\gamma \qquad \text{(I.18)}$$

Remarque

La troncature du développement en séries de Fourier de la réponse en fréquence du filtre souhaité conduit au phénomène de Gibbs. L'approximation sera d'autant meilleure que la fonction fenêtre W sera constituée par des impulsions aussi étroites que possible. Ceci entraîne un choix du nombre d'échantillons N suffisamment grand. Il faut donc rechercher un compromis entre la précision de l'approximation et le degré aussi faible que possible pour la transmitance du filtre. Ce compromis est d'ailleurs conditionné par un choix judicieux de la fonction fenêtre.

Parmis les fenêtres les plus utilisées on peut citer **[25,28]** :

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

La fenêtre de *Hanning* définie par :

$$W_{han}(n) = 0.5[1-\cos(2\pi n/N)] \qquad \text{avec } n = 0,1,\ldots,N\text{-}1 \tag{I.19}$$

La fenêtre de *Hamming :*

$$W_{ham}(n) = 0.54 - 0.46\cos(2\pi n/N) \quad \text{avec } n = 0,1,\ldots,N\text{-}1 \tag{I.20}$$

La fenêtre de *Kaiser :*

$$\begin{cases} W_k(n) = \dfrac{I_0(\beta\sqrt{1(1-n/Ns)^2})}{I_0(\beta)} & si\ 0 \le n \le N \\ 0 & \text{si } non \end{cases} \tag{I.21}$$

Avec I_0 : fonction de Bessel.

β : paramètre de forme de la fenêtre

Ns = N/2 : point de symétrie de la fenêtre.

La fenêtre de *Blackman :*

$$Wb(n) = 0.42 + 0.5\cos(2\pi/(N-1)) + 0.08\cos(4\pi/(N-1)) \tag{I.22}$$

avec $n = 0,1,\ldots,N$ -1.

I.3.3. Technique des bandes [26]

I.3.3.1. Description et principe.

La figure.I.5 représente la réponse en fréquence idéale d'un filtre numérique à bande étroite. La largeur de la bande de transition est nulle, ω_i la fréquence centrale de la bande, les valeurs désirées dans la bande passante (ω_1-$\delta\omega$/2 à ω_1+$\delta\omega$/2*)* et les bandes affaiblies sont respectivement *D* et *zéro*. La largeur de la bande étroite est $\delta\omega$.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

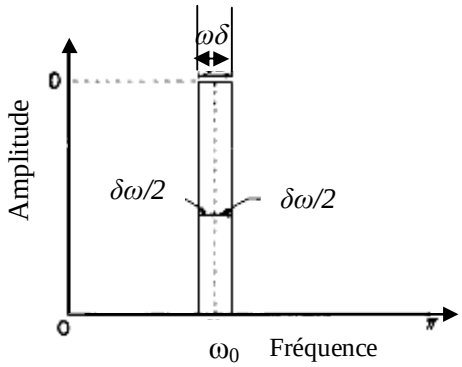


Fig. I.5. Réponse en fréquence idéale d'une bande étroite.

La caractéristique de phase est supposée linéaire ; elle est définie par $\theta(\omega) = k\,\omega$ radians ou k est une constante et ω une fréquence numérique variable.

Quand $\delta\omega$ est très petit, les coefficients du filtre $h_d(n)$ sont donnés par **[26]** :

$$h_d(n) = ((D.\delta\omega)/\pi)[\cos\{(n+k).\omega_i\}] \tag{I.23}$$

L'équation (I.23) représente un filtre RII non causal qui est sans aucune utilité pratique.

Cependant, en utilisant (I.23), les coefficients du filtre RIF causal d'ordre N approximant la réponse idéale de la figure. I.5. peuvent être obtenus comme suit **[26]** :

$$h_b(n) = (D/k).\cos\{\omega_i[n-(N-1)/2]\} \tag{I.24}$$

Òu $n = 0,1,\ldots, N-1$ avec N un entier impair.

$$\text{et } k = (N+1)/2 + \sum_{n=0}^{(N-2)/2} \cos\{2\omega_i[n-(N-1)/2]\} \tag{I.25}$$

Le retard de phase pour un tel filtre est égal à $\omega(N-1)/2$.

La largeur de la bande de transition est prise comme une valeur finie $T\omega$, donnée par la relation empirique **[26]** :

$$T\omega = 6.2197/N \tag{I.26}$$

Le principe d'approximation d'une bande étroite d'un filtre RIF peut être utilisé pour la conception des filtres RIF à phase linéaire avec des caractéristiques arbitraires de la réponse en fréquence.

Pour mieux comprendre cette technique de conception, on considère un filtre passe-bande dont les limites inférieure et supérieure de la bande passante sont notées respectivement ω_L et $\omega_\gamma.T\omega$ est la largeur de la bande de transition, $\overline{\gamma}$ est la valeur désirée de la réponse fréquentielle dans la bande passante et zéro dans les bandes atténuées.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

La figure.I.6 représente la décomposition de la réponse en fréquence d'un tel filtre en un nombre fini noté *J* de réponses fréquentielles individuelles. Chaque réponse individuelle peut être approximée par une bande étroite òu $\omega_1, \omega_2, \ldots, \omega_j$ sont les fréquences centrales de chaque bande étroite.

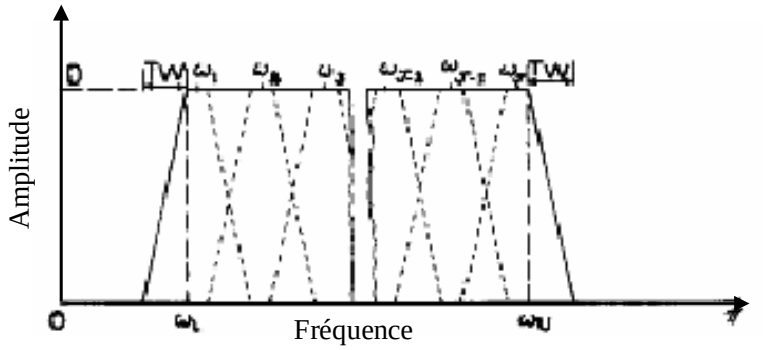


Fig.I.6. Réponse fréquentielle d'un filtre passe-bande divisée en *J* parties.

La valeur de *J*, représentant le nombre de réponse individuelle, est calculée suivant la relation **[23,37]** :

$$J = \text{partie entière de } \{(\omega_\gamma - \omega_L)/T_{\omega+1}\} \quad \text{(I.27)}$$

Les fréquences centrales $\omega_1, \omega_2, \omega_3, \ldots, \omega_j$ sont données par **[26]** :

$$\omega_m = \omega_L + (m-0.5)B\omega + (m-1)T\omega \quad \text{(I.28)}$$

Ou $m = 1,2,\ldots,J$. $B\omega$ est la largeur de la bande passante du $J^{iéme}$ filtre à bande étroite. Elle est donnée par l'expression **[37]** :

$$B\omega = [\omega_\gamma - \omega_L - (J-1)\,T\omega]/J \quad \text{(I.29)}$$

Les coefficients $h_m(n)$ ($m=1,2,\ldots, J$ et $n=0,1,\ldots, N-1$) de chaque filtre à bande étroite sont obtenus en utilisant (I.24), (I.25), (I.27), (I.29)

Les coefficients du filtre passe-bande seront la somme de tous les coefficients de chaque bande étroite. Ils sont donnés par **[37]** :

$$h(n) = \sum_{m=1}^{j} h_m(n) \quad \text{(I.30)}$$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

La figure.I.7 montre l'organigramme détaillé de ce programme.

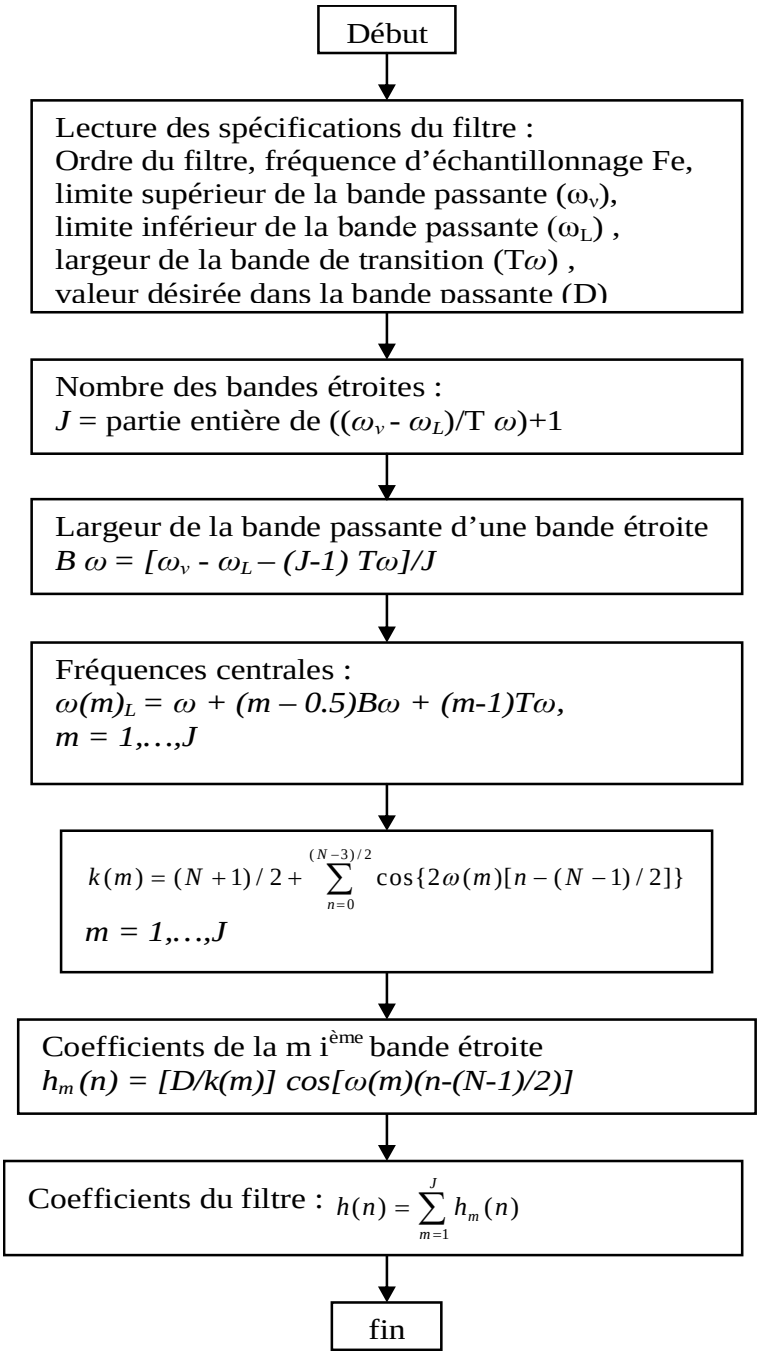


Fig.I.7. Organigramme de synthèse par la technique des bandes étroites.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

I.3.3.2. Exemples de synthèse d'un filtre RIF

I.3.3.3. Synthèse d'un filtre passe-bande par la méthode de fenêtrage pour N = 24, 64-taps

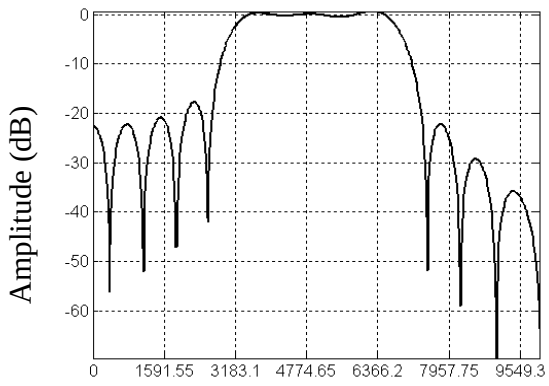
Les spécifications du filtre sont :

f_1 = 3000 Hz: première fréquence de coupure, f_2= 7000 Hz: deuxième fréquence de coupure, f_e = 10000Hz: fréquence d'échantillonnage.

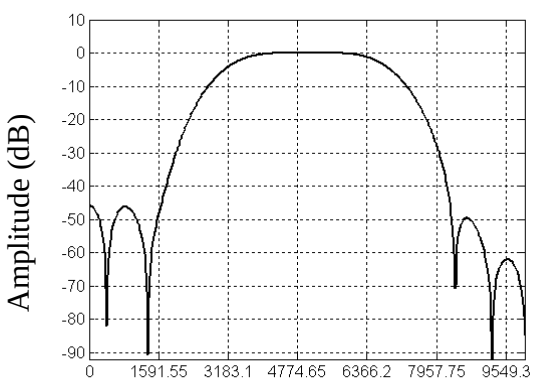
Les figures.I.8 et I.9 représentent les réponses fréquentielles en amplitude des filtres passe-bande synthétisés par la méthode des fenêtres pour N = 24, 64-taps, et la réponse impulsionnelle donnée sur le tableau.I.2.

	Rectangulaire	**Hamming**
***N* = 24**	$h_0 = h_{23} = 0.0328$, $h_6 = h_{17} = 0.0262$, $h_1 = h_{22} = -0.0137$, $h_7 = h_{16} = 0.0320$, $h_2 = h_{21} = 0.0152$, $h_8 = h_{15} = 0.1079$, $h_3 = h_{20} = -0.0444$, $h_9 = h_{14} = -0.1867$, $h_4 = h_{19} = -0.0622$, $h_{10} = h_{13} = -0.2517$, $h_5 = h_{18} = 0.0581$, $h_{11} = h_{12} = 0.2884$	$h_0 = h_{23} = 0.0025$, $h_6 = h_{17} = 0.0144$, $h_1 = h_{22} = -0.0013$, $h_7 = h_{16} = 0.0214$, $h_2 = h_{21} = 0.0021$, $h_8 = h_{15} = 0.0834$ $h_3 = h_{20} = -0.0096$, $h_9 = h_{14} = -0.1608$, $h_4 = h_{19} = -0.0196$, $h_{10} = h_{13} = -0.2326$, $h_5 = h_{18} = 0.0249$, $h_{11} = h_{12} = 0.2759$
***N* = 64**	$h_0 = h_{63} = 0.0117$, $h_{16} = h_{47} = -0.0091$, $h_1 = h_{62} = -0.0046$, $h_{17} = h_{46} = -0.0097$, $h_2 = h_{61} = 0.0048$, $h_{18} = h_{45} = -0.0273$, $h_3 = h_{60} = -0.0129$, $h_{19} = h_{44} = 0.0364$, $h_4 = h_{59} = -0.0166$, $h_{20} = h_{43} = 0.0320$, $h_5 = h_{58} = 0.0139$, $h_{21} = h_{42} = -0.0134$, $h_6 = h_{57} = 0.0055$, $h_{22} = h_{41} = 0.0148$, $h_7 = h_{56} = 0.0057$, $h_{23} = h_{40} = -0.0433$, $h_8 = h_{55} = 0.0157$, $h_{24} = h_{39} = -0.0607$, $h_9 = h_{54} = -0.0202$, $h_{25} = h_{38} = 0.0567$, $h_{10} = h_{53} = -0.0171$, $h_{26} = h_{37} = 0.0256$, $h_{11} = h_{52} = 0.0069$, $h_{27} = h_{36} = 0.0313$, $h_{12} = h_{51} = -0.0072$, $h_{28} = h_{35} = 0.1052$, $h_{13} = h_{50} = 0.0199$, $h_{29} = h_{34} = -0.1821$, $h_{14} = h_{49} = 0.0260$, $h_{30} = h_{33} = -0.2456$, $h_{15} = h_{48} = -0.0223$, $h_{31} = h_{32} = 0.2814$,	$h_0 = h_{63} = 0.0009$, $h_1 = h_{62} = -0.0004$, $h_2 = h_{61} = 0.0004$, $h_3 = h_{60} = -0.0013$, $h_4 = h_{59} = -0.0019$, $h_5 = h_{58} = 0.0019$, $h_6 = h_{57} = 0.0009$, $h_7 = h_{56} = 0.0011$, $h_8 = h_{55} = 0.0034$, $h_9 = h_{54} = -0.0051$, $h_{10} = h_{53} = -0.0049$, $h_{11} = h_{52} = 0.0022$, $h_{12} = h_{51} = -0.0027$, $h_{13} = h_{50} = 0.0082$, $h_{14} = h_{49} = 0.0118$, $h_{15} = h_{48} = -0.0112$, $h_{16} = h_{47} = -0.0050$, $h_{17} = h_{46} = -0.0057$, $h_{18} = h_{45} = -0.0173$, $h_{19} = h_{44} = 0.0247$, $h_{20} = h_{43} = 0.0231$, $h_{21} = h_{42} = -0.0134$, $h_{22} = h_{41} = -0.0102$, $h_{23} = h_{40} = -0.0362$, $h_{24} = h_{39} = -0.0527$, $h_{25} = h_{38} = 0.0508$, $h_{26} = h_{37} = 0.0236$, $h_{27} = h_{36} = 0.0295$, $h_{28} = h_{35} = 0.1012$, $h_{29} = h_{34} = -0.1776$, $h_{30} = h_{33} = -0.2417$, $h_{31} = h_{32} = 0.2782$,

Tableau.I.2. Réponse impulsionnelle d'un filtre passe-bande RIF par la méthode de fenêtrage pour N = 24, 64-taps.

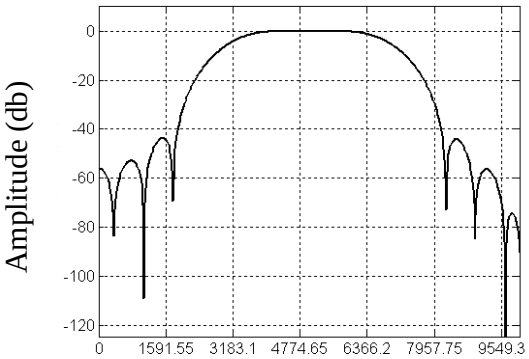


(a) Rectangulaire

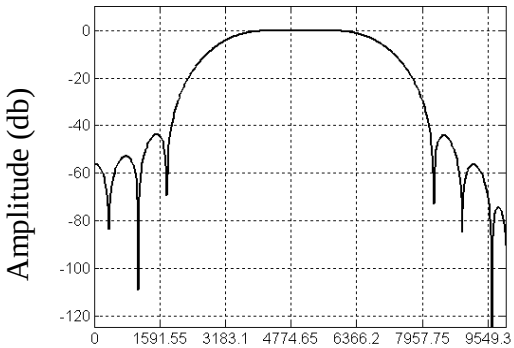


(b) Hamming

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

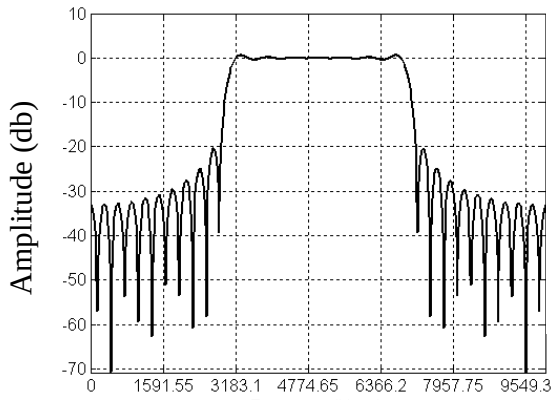


(c) Hanning

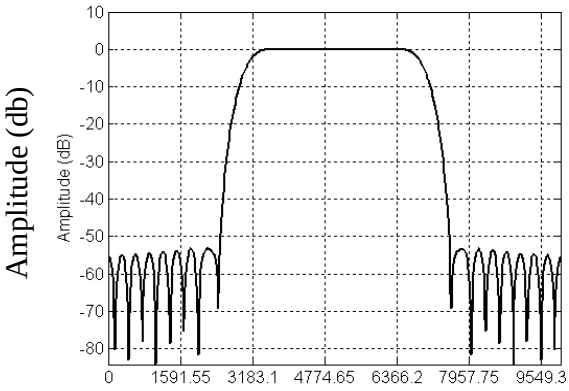


(d) Blakman

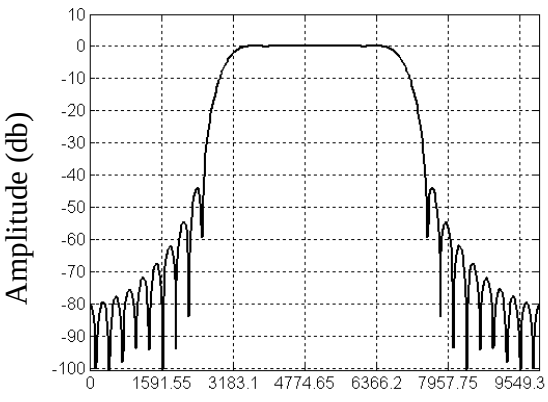
Fig.I.8. Synthèse par la méthode de fenêtrage pour *N* = 24-taps.



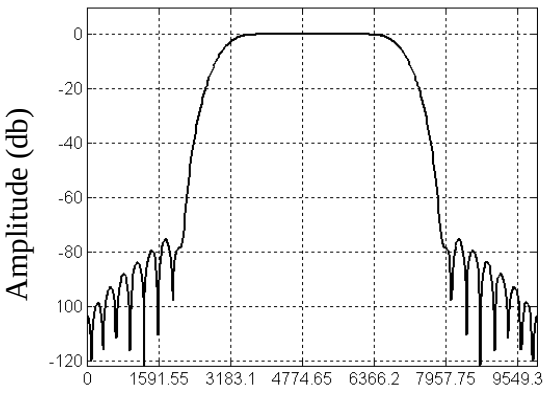
(a) Rectangulaire



(b) Hamming



(c) Hanning



(d) Blakman

Fig.I.9. Synthèse par la méthode de fenêtrage pour *N* = 64-taps.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

I.3.3.4. Synthèse d'un filtre passe-bas Par la méthode de fenêtrage pour N = 24, 64-taps

Les spécifications du filtre sont :

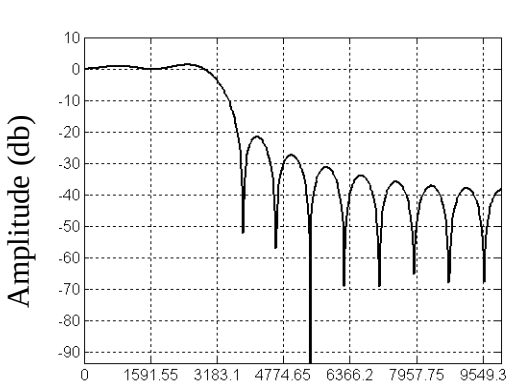
f_1 = 3000 Hz : fréquence de coupure.

f_e = 10000Hz : fréquence d'échantillonnage.

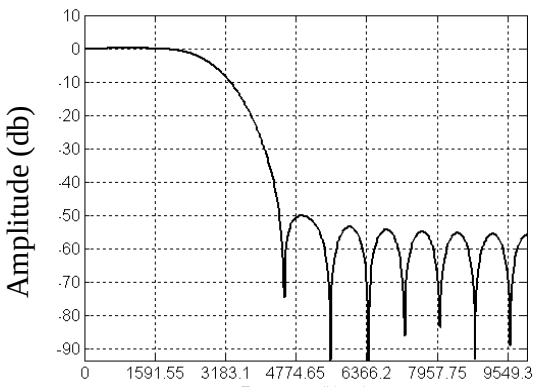
Les figure.I.10 et I.11 représentent les réponses fréquentielles en amplitude des filtres passe-bas synthétisés par la méthode des fenêtres pour N = 24, 64-taps, et la réponse impulsionnelle donnée sur le tableau.I.3.

	Rectangulaire	**Hamming**
***N* = 24**	$h_0 = h_{23}$ =-0.0277, $h_1 = h_{22}$ =-0.0140 ,$h_2 = h_{21}$ =0.0154 $h_3 = h_{20}$ = 0.0375, $h_4 = h_{19}$ =0.0304,$h_5 = h_{18}$ =-0.0078 $h_6 = h_{17}$ =-0.0523, $h_7 = h_{16}$ = -0.0639,$h_8 = h_{15}$ =-0.0144 $h_9 = h_{14}$ = 0.0913, $h_{10} = h_{13}$ = 0.2125,$h_{11} = h_{12}$ = 0.2930	$h_0 = h_{23}$ =-0.0022,$h_1 = h_{22}$ =-0.0013, $h_2 = h_{21}$= 0.0022, $h_3 = h_{20}$ =0.0084,$h_4 = h_{19}$ =0.0099,$h_5 = h_{18}$ =-0.0034, $h_6 = h_{17}$ =-0.0295,$h_7 = h_{16}$ =-0.0438,$h_8 = h_{15}$ = -0.0115 $h_9 = h_{14}$ =0.0809,$h_{10} = h_{13}$ = 0.2020,$h_{11} = h_{12}$ =0.2884.
***N* = 64**	$h_0 = h_{63}$ = -0.0100,$h_1 = h_{62}$ = -0.0048,$h_2 = h_{61}$= 0.0049 $h_3 = h_{60}$ = 0.0111,$h_4 = h_{59}$ = 0.0082,$h_5 = h_{58}$ = -0.0019 $h_6 = h_{57}$= -0.0112, $h_7 = h_{56}$ =-0.0116, $h_8 = h_{55}$ =-0.0021 $h_9 = h_{54}$=0.0101, $h_{10} = h_{53}$ = 0.0147, $h_{11} = h_{52}$ = 0.0071 $h_{12} = h_{51}$ = -0.0075,$h_{13} = h_{50}$ =-0.0171,$h_{14} = h_{49}$=-0.0129 $h_{15} = h_{48}$ = 0.0030, $h_{16} = h_{47}$ = 0.0184, $h_{17} = h_{46}$= 0.0197 $h_{18} = h_{45}$= 0.0037, $h_{19} = h_{44}$= -0.0181,$h_{20} = h_{43}$= -0.0275 $h_{21} = h_{42}$= -0.0138,$h_{22} = h_{41}$ = 0.0153,$h_{23} = h_{40}$ = 0.0372 $h_{24} = h_{39}$ = 0.0302,$h_{25} = h_{38}$ =-0.0077,$h_{26} = h_{37}$ = -0.0519 $h_{27} = h_{36}$=-0.0634,$h_{28} = h_{35}$ = -0.0143,$h_{29} = h_{34}$= 0.0906 $h_{30} = h_{33}$ = 0.2109;$h_{31} = h_{32}$ = 0.2908,	$h_0 = h_{63}$ =-0.0008,$h_1 = h_{62}$ =-0.0004,$h_2 = h_{61}$ = 0.0004 $h_3 = h_{60}$ = 0.0011,$h_4 = h_{59}$ =0.0010,$h_5 = h_{58}$ = -0.0003 $h_6 = h_{57}$= -0.0018,$h_7 = h_{56}$=-0.0022, $h_8 = h_{55}$ =-0.0005 $h_9 = h_{54}$=0.0025, $h_{10} = h_{53}$ = 0.0042, $h_{11} = h_{52}$ = 0.0023 $h_{12} = h_{51}$=-0.0028,$h_{13} = h_{50}$=-0.0071,$h_{14} = h_{49}$ = -0.0059 $h_{15} = h_{48}$ =0.0015,$h_{16} = h_{47}$=0.0101, $h_{17} = h_{46}$ = 0.0117 $h_{18} = h_{45}$=0.0024,$h_{19} = h_{44}$=-0.0124, $h_{20} = h_{43}$= -0.0199 $h_{21} = h_{42}$ =-0.0106,$h_{22} = h_{41}$ = 0.0123,$h_{23} = h_{40}$ = 0.0312 $h_{24} = h_{39}$=0.0263,$h_{25} = h_{38}$ =-0.0069, $h_{26} = h_{37}$=-0.0481, $h_{27} = h_{36}$=-0.0602,$h_{28} = h_{35}$= -0.0138,$h_{29} = h_{34}$= 0.0888, $h_{30} = h_{33}$ = 0.2086,$h_{31} = h_{32}$ = 0.2890

Tableau.I.3. Réponse impulsionnelle d'un filtre passe-bas RIF par la méthode de fenêtrage pour N = 24, 64-taps.

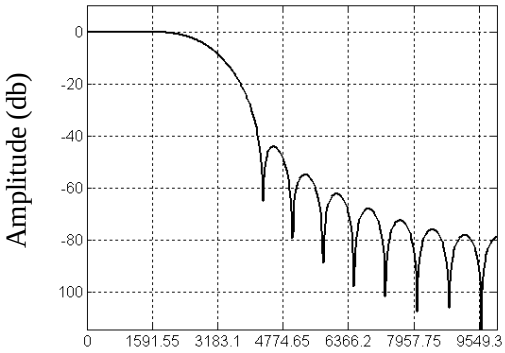


(a) Rectangulaire

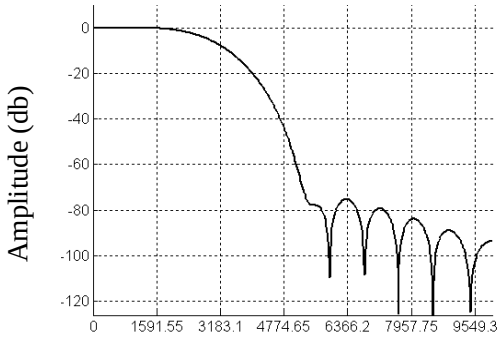


(b) Hamming

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

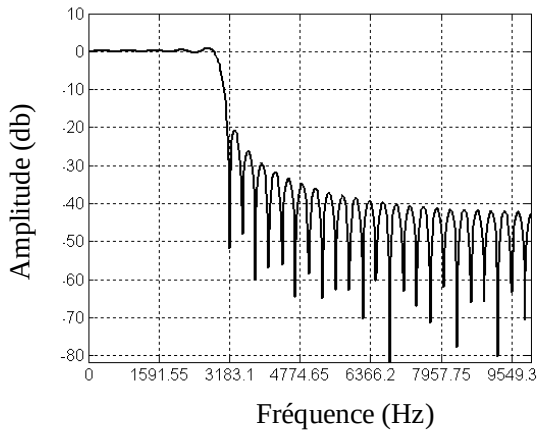


(c) Hanning

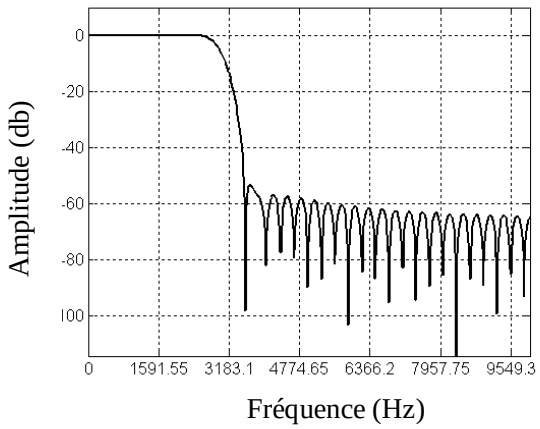


(d) Blakman

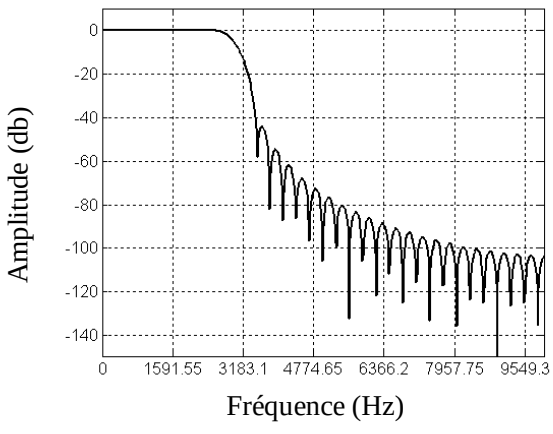
Fig.I.10. Synthèse par la méthode de fenêtrage pour *N* = 24-taps.



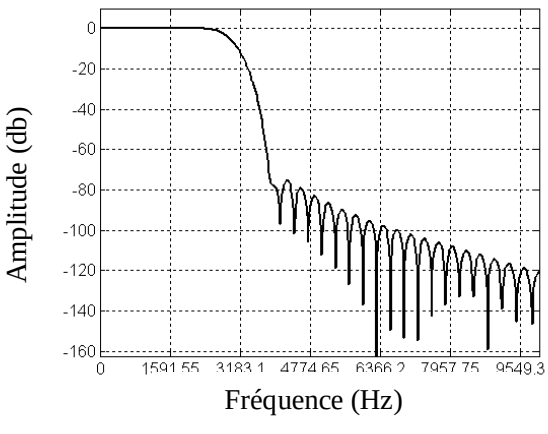
(a) Rectangulaire



(b) Hamming



(c) Hanning



(d) Blakman

Fig.I.11. Synthèse par la méthode de fenêtrage pour *N* = 64-taps.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

I.3.3.5. Synthèse d'un filtre multi-bande par la méthode de fenêtrage pour N = 65, 120-taps

Les spécifications du filtre sont :

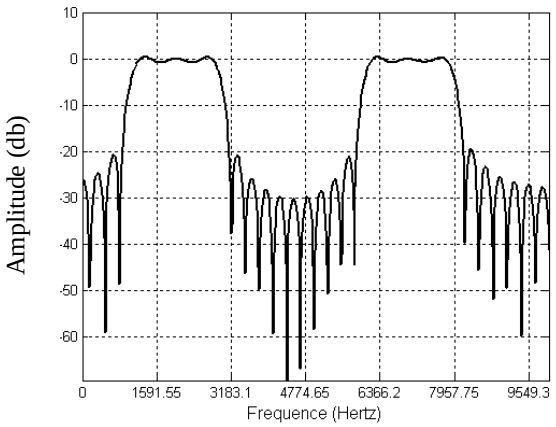
f_1 = 1000Hz : 1ère fréquence de coupure, f_2 = 3000Hz : 2ème fréquence de coupure.

f_3 = 6000Hz: 3ème fréquence de coupure, f_4 = 8000Hz : 4ème fréquence de coupure, f_e = 10000Hz.

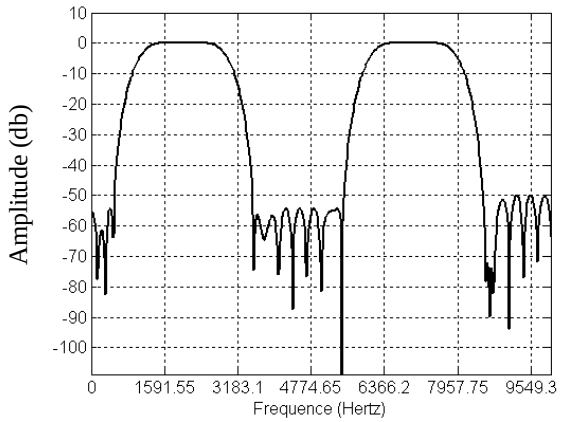
Les figure.I.12 et I.13 représentent les réponses fréquentielles en amplitude des filtres Multi-bandes synthétisés par la méthode des fenêtres pour N = 65, 120-taps, et la réponse impulsionnelle donnée sur le tableau.I.4.

	Rectangulaire
N = 65	$h_0 = h_{63}$ =-0.0138, $h_1 = h_{62}$ =-0.0016, $h_2 = h_{61}$ = 0.0016, $h_3 = h_{60}$ =0.0153, $h_4 = h_{59}$ = -0.0111, $h_5 = h_{58}$ = -0.0152, $h_6 = h_{57}$= -0.0014, $h_7 = h_{56}$ =-0.0452, $h_8 = h_{55}$ = -0.0100, $h_9 = h_{54}$= 0.0135, $h_{10} = h_{53}$ =-0.0051, $h_{11} = h_{52}$ = 0.0065, $h_{12} = h_{51}$ =-0.0069, $h_{13} = h_{50}$ =0.0060, $h_{14} = h_{49}$ = -0.0174, $h_{15} = h_{48}$ = 0.0142, $h_{16} = h_{47}$ =0.0715, $h_{17} = h_{46}$ =0.0025 $h_{18} = h_{45}$= 0.0299, $h_{19} = h_{44}$= 0.0244, $h_{20} = h_{43}$= -0.0379, $h_{21} = h_{42}$ =-0.0045, $h_{22} = h_{41}$=0.0050, $h_{23} = h_{40}$ =0.0513 $h_{24} = h_{39}$ =-0.0406, $h_{25} = h_{38}$ =-0.0622, $h_{26} = h_{37}$ = -0.0066, $h_{27} = h_{36}$= -0.2463, $h_{28} = h_{35}$ =-0.0669, $h_{29} = h_{34}$ = 0.1218 $h_{30} = h_{33}$ = -0.0737, $h_{31} = h_{32}$ = 0.2678.
N = 120	$h_0 = h_{60}$ =-0.0024, $h_1 = h_{61}$ =0.0020, $h_2 = h_{62}$ =-0.0057, $h_3 = h_{63}$ = 0.0045, $h_4 = h_{64}$ =0.0215, $h_5 = h_{65}$ = 0.0007 , $h_6 = h_{66}$=0.0081, $h_7 = h_{67}$ = 0.0063, $h_8 = h_{68}$ =-0.0091, $h_9 = h_{69}$= -0.0010, $h_{10} = h_{70}$ = 0.0010 , $h_{11} = h_{71}$ = 0.0097, $h_{12} = h_{72}$ =-0.0069, $h_{13} = h_{73}$ =-0.0094, $h_{14} = h_{74}$=-0.0009, $h_{15} = h_{75}$ =-0.0268, $h_{16} = h_{76}$ =-0.0058, $h_{17} = h_{77}$ = 0.0077, $h_{18} = h_{78}$=-0.0029, $h_{19} = h_{79}$= 0.0036 , $h_{20} = h_{80}$=-0.0037, $h_{21} = h_{81}$= 0.0031, $h_{22} = h_{82}$=-0.0088, $h_{23} = h_{83}$= 0.0069, $h_{24} = h_{84}$ =0.0336 , $h_{25} = h_{85}$=0.0011, $h_{26} = h_{86}$ =0.0130, $h_{27} = h_{87}$= 0.0101, $h_{28} = h_{88}$ =-0.0149 , $h_{29} = h_{89}$ =-0.0017, $h_{30} = h_{90}$ =-0.0017, $h_{31} = h_{91}$=0.0165, $h_{32} = h_{92}$ = -0.0119, $h_{33} = h_{93}$ =-0.0164, $h_{34} = h_{94}$ =-0.0015 , $h_{35} = h_{95}$=-0.0488, $h_{36} = h_{96}$ =-0.0107, $h_{37} = h_{97}$ =0.0146, $h_{38} = h_{98}$ =-0.0055, $h_{39} = h_{99}$ =0.0070, $h_{40} = h_{100}$ =-0.0074, $h_{41} = h_{101}$=0.0064, $h_{42} = h_{102}$= -0.0188, $h_{43} = h_{103}$=0.0153, $h_{44} = h_{104}$=0.0771, $h_{45} = h_{105}$=0.0027, $h_{46} = h_{106}$=0.0322, $h_{47} = h_{107}$=0.0263, $h_{48} = h_{108}$=-0.0408, $h_{49} = h_{109}$=-0.0049, $h_{50} = h_{110}$= 0.0054, $h_{51} = h_{111}$= 0.0552, $h_{52} = h_{112}$=-0.0438, $h_{53} = h_{113}$=-0.0670, $h_{54} = h_{114}$=-0.0071, $h_{55} = h_{115}$=-0.2654, $h_{56} = h_{116}$=-0.0721, $h_{57} = h_{117}$ =0.1313, $h_{58} = h_{118}$=-0.0795, $h_{59} = h_{119}$ = 0.2886

Tableau.I.4. Réponse impulsionnelle d'un filtre multi-bandes RIF par la méthode de fenêtrage pour N = 65, 120-taps.

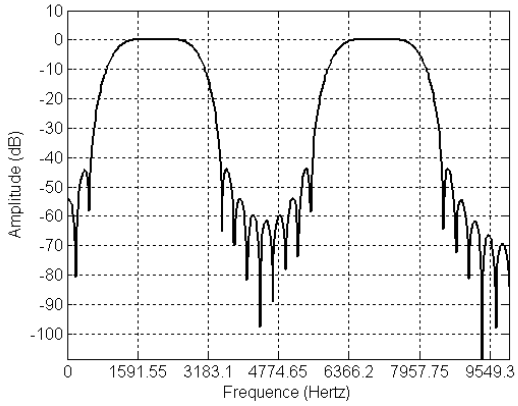


(a) Rectangulaire

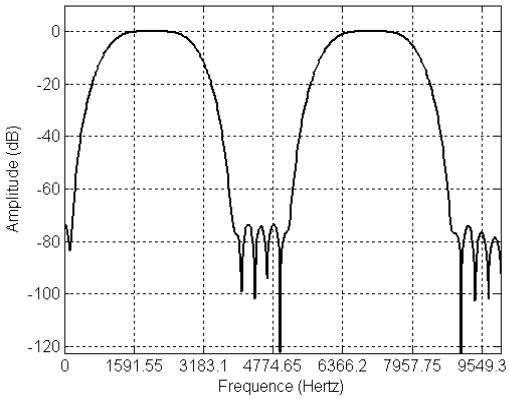


(b) Hamming

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

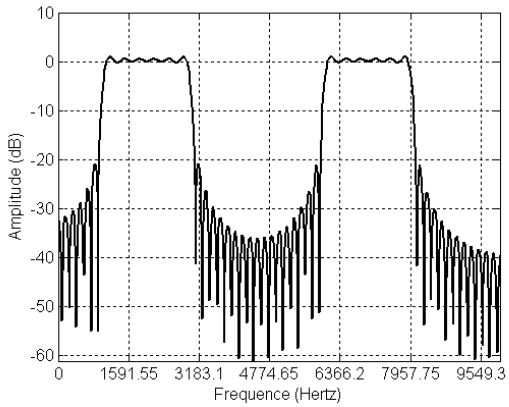


(c) Hanning

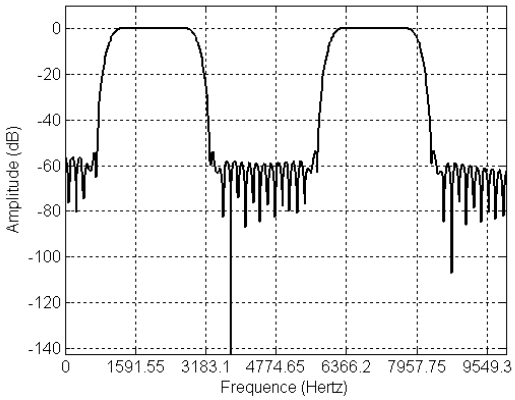


(d) Blakman

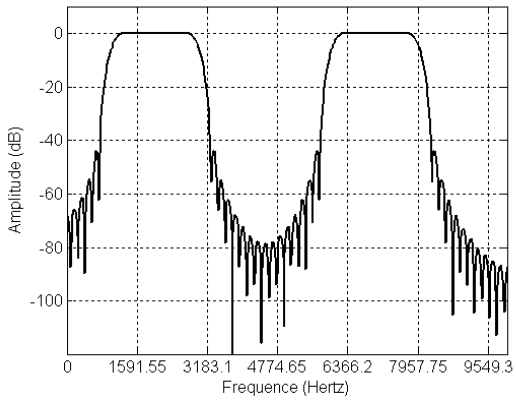
Fig.I.12. synthèse par la méthode de fenêtrage pour *N* = 65-taps.



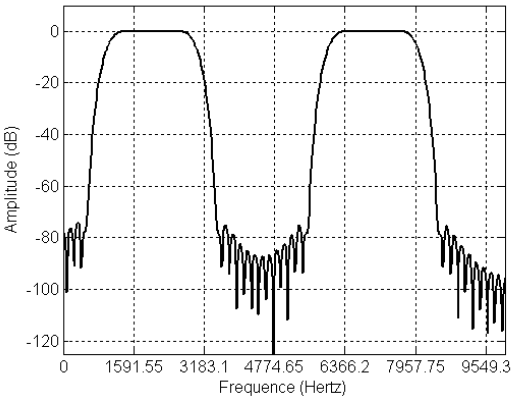
(a) Rectangulaire



(b) Hamming



(c) Hanning



(d) Blakman

Fig.I.13. synthèse par la méthode de fenêtrage pour *N* = 120-taps.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

I.3.4. La méthode de l'échantillonnage en fréquence [25,28]

Pour cette méthode, uniquement *N* échantillons de la fonction $H\infty(\omega)$ sont pris en compte :

$$H(k) = H_{\infty}(\omega) \; avec \; \omega = k(2\pi / N) \quad k = 0,\ldots, N-1 \tag{I.31}$$

Ensuite, la réponse impulsionnelle du filtre RIF correspondant est calculée comme la TFD inverse de la séquence *H(k).* La fonction de transfert du filtre conçu sera donnée par :

$$H(\omega) = e^{-j\omega\frac{N-1}{2}} \sum_{k=0}^{N-1} A(\omega,k).H_d(k) \tag{I.32}$$

Òu $H_d(k)$ représente les échantillons de la fonction de transfert de phase nulle du modèle idéal et :

$$A(\omega,k) = \frac{1}{N}\frac{\sin\left(\frac{N}{2}\left(\omega - k\frac{2\pi}{N}\right)\right)}{\sin\left(\frac{1}{2}\left(\omega - k\frac{2\pi}{N}\right)\right)} = \frac{1}{N}Sa\left(\omega - k\frac{2\pi}{N}\right) \tag{I.33}$$

Il en résulte la fonction de transfert de phase nulle du filtre RIF :

$$H_0(\omega) = \frac{1}{N}\sum_{k=0}^{N-1} H_d(k).S_a\left(\omega - k\frac{2\pi}{N}\right) \tag{I.34}$$

Il est à noter que $H_0(\omega)$ s'obtient par une procédure d'interpolation de la réponse en fréquence échantillonnée, à savoir que chaque échantillon $H_d(k)$ pondère une fonction $S_a(\omega)$, centrée sur la position de l'échantillon. Par conséquent, l'erreur d'approximation sera nulle pour les fréquences $\omega_k = k(2\pi / N)$ et finie entre elles.

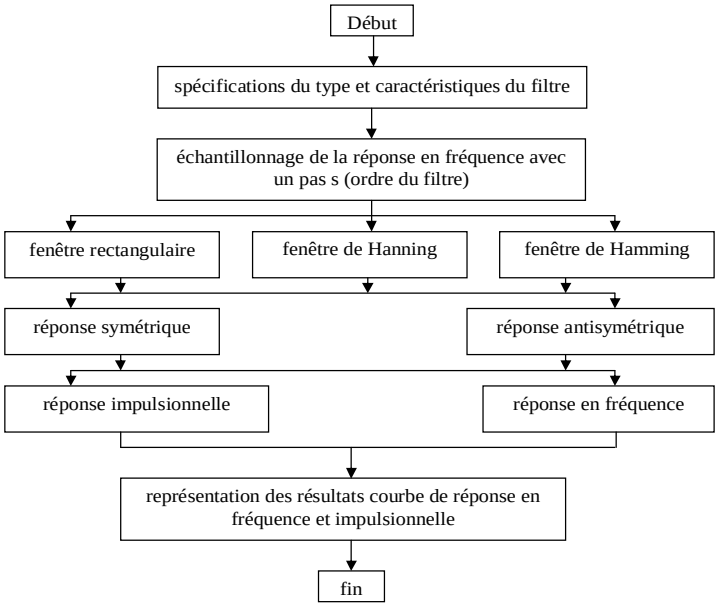


Fig. I.14. Organigramme de synthèse par la méthode d'échantillonnage en fréquence.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

I.3.4.1. Exemple de synthèse d'un filtre RIF

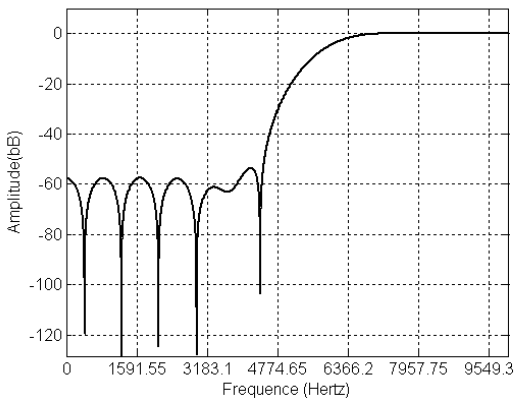
I.3.4.2. Synthèse d'un filtre passe-haut par la méthode d'échantillonnage en fréquence pour N = 25, 65,129-taps

Les spécifications du filtre sont : f_1 = 6000Hz : fréquence de coupure, f_e = 10000Hz.

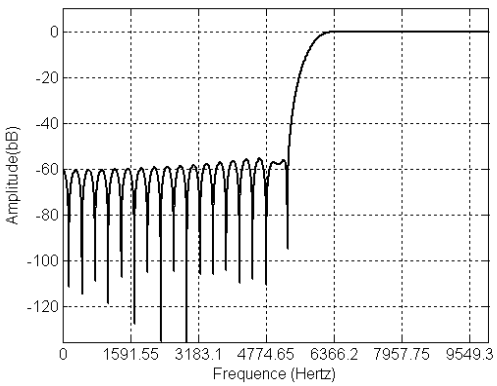
la figure.I.15 représente la réponse fréquentielle du filtre RIF passe-haut synthétisé par la méthode d'échantillonnage en fréquence pour N = 25, 65, 129-taps, et la réponse impulsionnelle donnée sur le tableau.I.5.

	Méthode d'échantillonnage en fréquence
N = 25-taps	$h_0 = h_{13} = 0.0001, h_1 = h_{14} = -0.0027, h_2 = h_{15} = 0.0022, h_3 = h_{16} = 0.0055, h_4 = h_{17} = -0.0106, h_5 = h_{18} = -0.0052,$ $h_6 = h_{19} = 0.0286, h_7 = h_{20} = -0.0104, h_8 = h_{21} = -0.0533, h_9 = h_{22} = 0.0645, h_{10} = h_{23} = 0.0752, h_{11} = h_{24} = -0.3025,$ $h_{12} = 0.4160,$
N = 65-taps	$h_0 = h_{33} = -0.0006, h_1 = h_{34} = -0.0003, h_2 = h_{35} = 0.0009, h_3 = h_{36} = -0.0002, h_4 = h_{37} = -0.0011, h_5 = h_{38} = 0.0010,$ $h_6 = h_{39} = 0.0010, h_7 = h_{40} = -0.0022, h_8 = h_{41} = -0.0001, h_9 = h_{42} = 0.0033, h_{10} = h_{43} = -0.0019, h_{11} = h_{44} = -0.0036,$ $h_{12} = h_{45} = 0.0049, h_{13} = h_{46} = 0.0020, h_{14} = h_{47} = -0.0079, h_{15} = h_{48} = 0.0021, h_{16} = h_{49} = 0.0094, h_{17} = h_{50} = -0.0085,$ $h_{18} = h_{51} = -0.0075, h_{19} = h_{52} = 0.0158, h_{20} = h_{53} = 0.0005, h_{21} = h_{54} = -0.0212, h_{22} = h_{55} = 0.0122, h_{23} = h_{56} = 0.0211,$ $h_{24} = h_{57} = -0.0295, h_{25} = h_{58} = -0.0111, h_{26} = h_{59} = 0.0489, h_{27} = h_{60} = -0.0150, h_{28} = h_{61} = -0.0668, h_{29} = h_{62} = 0.0731,$ $h_{30} = h_{63} = 0.0794, h_{31} = h_{64} = -0.3066, h_{32} = 0.4160$
N = 129-taps	$h_0 = h_{65} = 0.0003, h_1 = h_{66} = -0.0002, h_2 = h_{67} = -0.0002, h_3 = h_{68} = 0.0004, h_4 = h_{69} = 0.0001, h_5 = h_{70} = -0.0005,$ $h_6 = h_{71} = 0.0002, h_7 = h_{72} = 0.0004, h_8 = h_{73} = -0.0005, h_9 = h_{74} = -0.0002, h_{10} = h_{75} = 0.0007, h_{11} = h_{76} = -0.0001,$ $h_{12} = h_{77} = -0.0008, h_{13} = h_{78} = 0.0006, h_{14} = h_{79} = 0.0007, h_{15} = h_{80} = -0.0011, h_{16} = h_{81} = -0.0001, h_{17} = h_{82} = 0.0015,$ $h_{18} = h_{83} = -0.0007, h_{19} = h_{84} = -0.0014, h_{20} = h_{85} = 0.0016, h_{21} = h_{86} = 0.0007, h_{22} = h_{87} = -0.0023, h_{23} = h_{88} = 0.0005,$ $h_{24} = h_{89} = 0.0025, h_{25} = h_{90} = -0.0020, h_{26} = h_{91} = -0.0019, h_{27} = h_{92} = 0.0034, h_{28} = h_{93} = 0.0003, h_{29} = h_{94} = -0.0041,$ $h_{30} = h_{95} = 0.0020, h_{31} = h_{96} = 0.0037, h_{32} = h_{97} = -0.0044, h_{33} = h_{98} = -0.0018, h_{34} = h_{99} = 0.0061, h_{35} = h_{100} = -0.0013,$ $h_{36} = h_{101} = -0.0063, h_{37} = h_{102} = 0.0050, h_{38} = h_{103} = 0.0044, h_{39} = h_{104} = -0.0083, h_{40} = h_{105} = -0.0005, h_{41} = h_{106} = 0.0098,$ $h_{42} = h_{107} = -0.0050, h_{43} = h_{108} = -0.0086, h_{44} = h_{109} = 0.0106, h_{45} = h_{110} = 0.0040, h_{46} = h_{111} = -0.0146,$ $h_{47} = h_{112} = 0.0036, h_{48} = h_{113} = 0.0151, h_{49} = h_{114} = -0.0127, h_{50} = h_{115} = -0.0106, h_{51} = h_{116} = 0.0213, h_{52} = h_{117} = 0.0006,$ $h_{53} = h_{118} = -0.0262, h_{54} = h_{119} = 0.0145, h_{55} = h_{120} = 0.0243, h_{56} = h_{121} = -0.0329, h_{57} = h_{122} = -0.0121, h_{58} = h_{123} = 0.0520,$ $h_{59} = h_{124} = -0.0156, h_{60} = h_{125} = -0.0686, h_{61} = h_{126} = 0.0742, h_{62} = h_{127} = 0.0800, h_{63} = h_{128} = -0.3071, h_{64} = 0.4160,$

Tableau.I.5. Réponse impulsionnelle d'un filtre passe-haut RIF par la méthode d'échantillonnage en fréquence pour N = 25, 65, 129-taps.

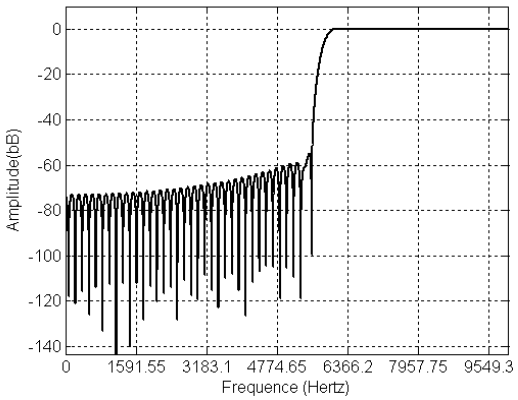


(a) N = 25-taps



(b) N = 65-taps

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



(c) N = 129-taps

Fig.I.15. Synthèse par la méthode d'échantillonnage en fréquence.

I.3.5. Les méthodes d'optimisation [25, 30, 31,38]

Pour améliorer la qualité de l'approximation dans la méthode de l'échantillonnage en fréquence, on peut utiliser de 1 à 3 échantillons situés dans la bande de transition comme variables supplémentaires, qui peuvent être utilisées dans une procédure d'optimisation de la solution, en vue de minimiser l'erreur. On obtient alors la méthode des moindres carrés pour la conception des filtres RIF lorsque cette procédure est utilisée pour l'optimisation. Cette méthode est également très utile dans le cas òu les échantillons ne sont pas uniformément espacés **[29,25].**

Une autre méthode très utilisée de la conception des filtres RIF par optimisation est la méthode de Remez, qui est basée sur le théorème de l'alternance **[27,30]**. Cette méthode réalise la meilleure approximation, au sens de Chebychev, de la fonction $H_d(\omega)$. Cette troisième classe de méthodes, basée sur des outils *CAO*(calcul assisté par ordinateur), permet donc l'optimisation de la solution à partir d'un critère d'erreur donné et d'un ensemble de contraintes initiales. L'utilisation de ces outils réclame néanmoins une bonne compréhension des méthodes de base afin d'être capable d'une attitude critique face aux résultats.

I.3.5.1. Méthode de Parks-McClellan [27, 30,40]

L'approche mathématique la plus utilisée jusqu'à présent dans le cas des coefficients à précision infinie est celle de Parks-McClellan (P.M.C.). Le filtre de P.M.C. possède la meilleure approximation au sens de Chebyshev. Son fondement mathématique et ses propriétés sont développés dans la section suivante.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

I.3.5.1.1. Calcul des coefficients d'un filtre à phase linéaire par Parks-McClellan *(*P.M.C*)*

Le filtre RIF à phase linéaire conçu par Parks-McClellan utilisant l'algorithme de Remez représente la meilleure approximation au sens de Chebyshev avec des coefficients à précision infinie. Auparavant, plusieurs chercheurs ont étudié les problèmes de conception (RIF) pour certains types de filtre en utilisant des algorithmes différents.

L'importance de cette nouvelle approche réside dans la combinaison entre la rapidité de la procédure de Remez avec la capacité de conception d'une grande classe de types de filtre, avec des filtres peu communs tels que les filtres passe bande à bandes multiples, filtre avec la transformée de Hilbert et les différentiateurs et des filtres plus communs tels que les filtres passe bande, coupe bande, passe bas et passe haut. La réponse en fréquence peut ainsi être approximée.

Dans cette section, nous allons présenter le fondement mathématique de la méthode P.M.C. pour des coefficients à précision infinie.

I.3.5.1.2. Formulation du problème d'approximation

Soit la réponse en fréquence d'un filtre RIF :

$$H(f) = \sum_{k=0}^{N-1} h(k) e^{-j2\pi kf} \tag{I.35}$$

La réponse en fréquence d'un filtre R.I.F à phase linéaire s'écrit :

$$H(f) = G(f) e^{j(L\pi/2 - ((N-1)/2).2\pi f)} \tag{I.36}$$

G(f) : fonction à valeur réelle. $L = 0$ ou 1.

Il existe quatre cas de filtre RIF à phase linéaire, qui dépendent de la parité de la longueur et de la symétrie de la réponse impulsionnelle (paire ou impaire), (positive ($L = 0$) ou négative ($L = 1$)) respectivement.

Par symétrie positive, nous avons $h(k) = h(N\text{-}1\text{-}k)$.

Par symétrie négative, nous avons $h(k) = -h(N\text{-}1\text{-}k)$.

Dans tous les cas, on ne s'intéresse qu'à *G(f)* fonction réelle qui est utilisée pour approximer les spécifications de l'amplitude idéale désirée, puisque le terme de la phase linéaire n'a aucun effet sur la réponse en amplitude. Auparavant, tous les algorithmes déjà établis, se sont concentrés sur le cas 1. Dans cette approche, on a pu combiné les quatre cas dans un seul algorithme en notant que *G(f)* s'écrit comme :

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$$G(f) = Q(f).P(f) \tag{I.37}$$

Où *P(f)* est une combinaison de fonctions cosinus qui dépend de chaque cas.

$$P(f) = \sum_{k=0}^{r-1} \alpha(k)\cos(2\pi kf) \tag{I.38}$$

et α *(k)* est une réponse implusionnelle dépendante du cas considéré.

Les quatre cas ont été écrits sous une forme commune afin que l'algorithme de Remez s'accomplisse convenablement. Le problème d'approximation d'origine réside dans la minimisation du maximum de l'erreur absolue pondérée définie comme :

$$\|E(f)\| = \max\{W(f)|D(f) - G(f)|\} \quad \text{avec} \quad f \in F \tag{I.39}$$

W(f) : fonction de pondération.

F : sous ensemble de fréquences dans les bandes d'intérêt (bande passante et bande atténuée).

D(f) : réponse en amplitude désirée.

En remplaçant *G(f)* par sa valeur, on aura :

$$\|E(f)\| = \max\{W'(f)\left|D'(f) - P(f)\right|\} \text{ avec } f \in F' \tag{I.40}$$

avec $W'(f) = W(f)\,Q(f)$, $D'(f) = D(f)/Q(f)$ et $F' \subset F$.

I.3.5.1.3. Théorème de l'alternance

Soit *G(f)* et *P(f)* des fonctions définies respectivement suivant(I.37) et (I.38). Une condition suffisante et nécessaire pour que *P(f)* soit l'unique et meilleure approximation au sens de Chebyshev à une fonction continue *D'(f)* dans une gamme de fréquence *F'* est que :

$E(f) = (W'(f)|D'(f) - P(f)|)$, expose $r+1$ fréquences extrêmes dans *F'* notées par 'F_i'.

$i = 1, 2, \ldots, r+1$,

Où $F_1 < F_2 < \ldots < F_r < F_{r+1}$.

avec $E(F_i) = -E(F_{i+1})$, $i = 1, \ldots, r$.

et $|E(F_i)| = max\,(E(f))$ pour $f \in F'$.

Un algorithme peut donc être conçu pour satisfaire la condition sur l'erreur du filtre dans le théorème d'alternance **[6]**.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

I.3.5.1.4. Description de l'algorithme de conception du filtre de Parks-McClellan sous l'algorithme d'échange de Remez [27]

La conception de cet algorithme consiste en une section de lecture des données, formulation du problème par une approximation appropriée équivalente, solution du problème d'approximation en utilisant la méthode de Remez, et enfin le calcul de la réponse impulsionnelle du filtre. La figure.I.16 représente l'organigramme général de L'algorithme de Parks-McClellan

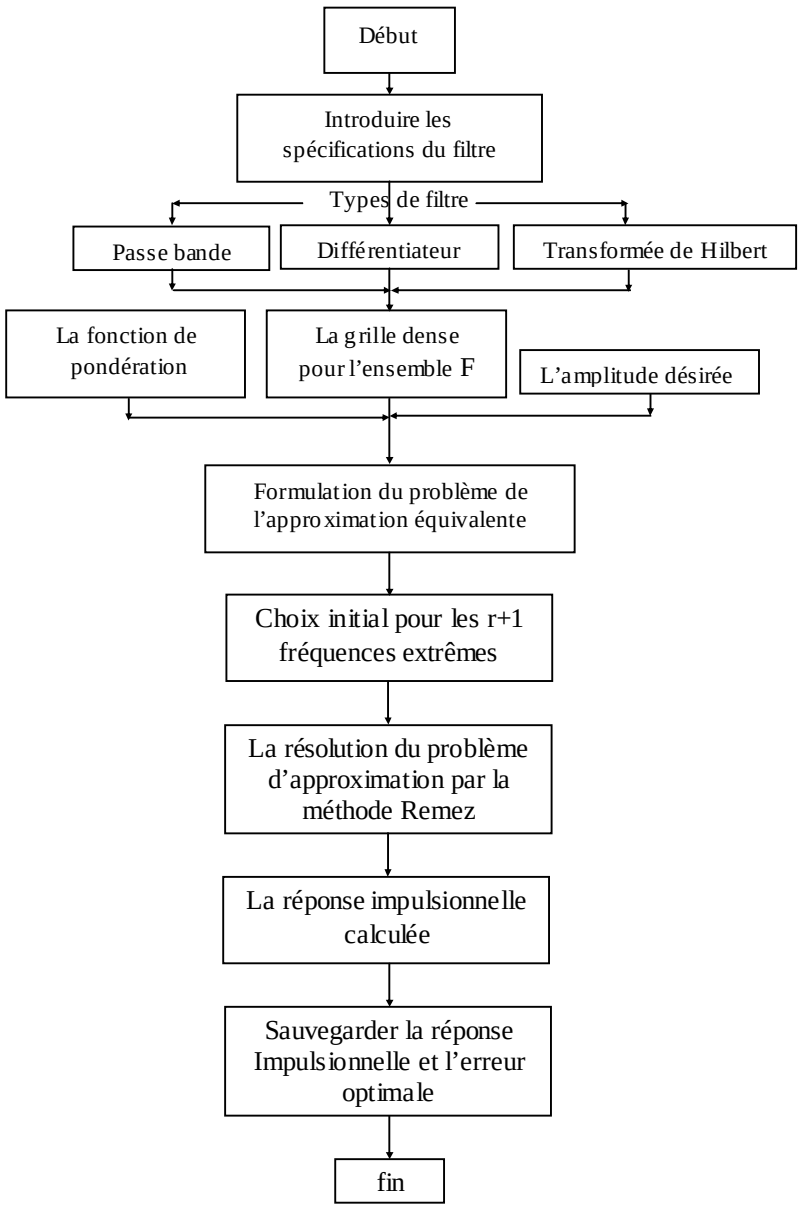


Fig. I.16. Organigramme de conception de filtre de Parks-McClellan à l'aide de la méthode Remez.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

L'algorithme d'échange de Remez comporte quatre étapes :

1. Spécifications de la réponse en fréquence désirée $D(\omega)$, de la fonction pondération $W(f)$ et du degré N du filtre.

2. Détermination du problème d'approximation équivalent par calcul de $W'(f)$, $D'(f)$, et $P(f)$.

3. Solution du problème d'approximation en utilisant l'algorithme d'échange de Remez.

4. Calcul des coefficients du filtre par transformée discrète de Fourier inverse.

Dans la première étape, il existe une ambiguïté. N étant fixé, rien n'assure que la solution optimale satisfera le gabarit imposé. Il existe des formules empiriques qui donnent une valeur approximative de N minimal. Ensuite, par augmentation ou diminution de celui-ci en fonction des résultats, on pourra trouver la valeur de N optimal. La figure.I.17 décrit l'organigramme de l'algorithme d'échange de Remez.

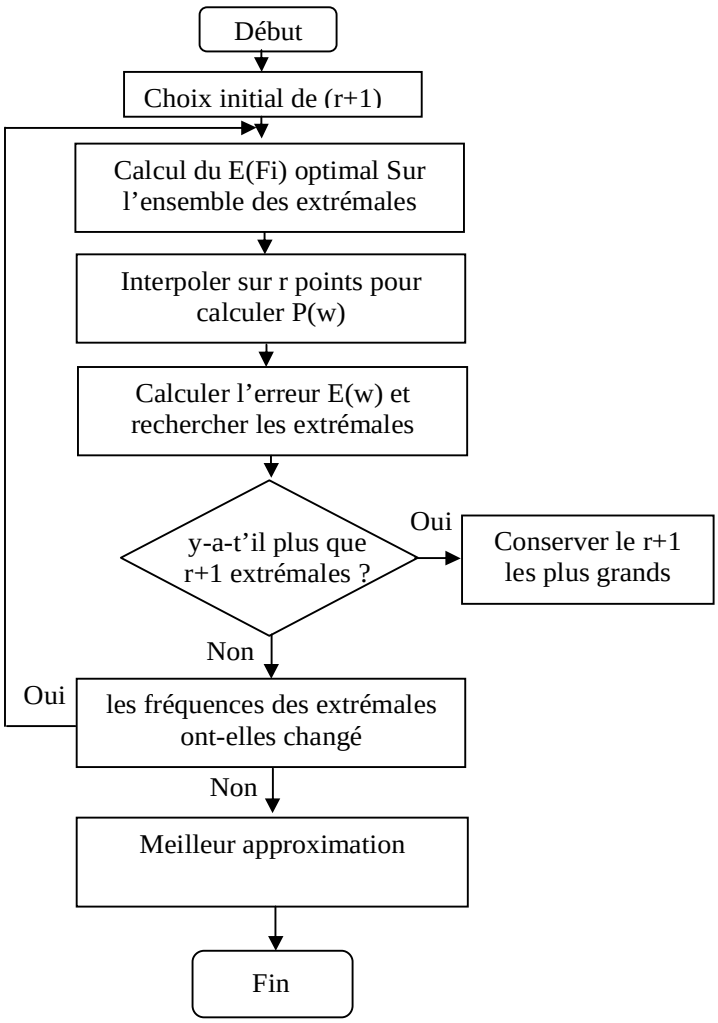


Fig. I.17. Organigramme de l'algorithme d'échange de Remez.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Cet organigramme présente l'approche utilisée par l'algorithme d'échange de Remez pour obtenir une solution au problème d'approximation.

Le filtre de Parks-McClellan (PMC) sous l'algorithme d'échange de Remez présente la meilleure approximation au sens de Chebyshev pour les coefficients à précision infinie. Il peut concevoir une large variété de filtres standard pour n'importe quelle réponse en amplitude désirée spécifiée par l'utilisateur. La rapidité de cet algorithme lui offre l'intérêt d'un large domaine d'application.

I.3.5.1.5. Exemple de synthèse d'un filtre RIF

I.3.5.1.5.1. Synthèse d'un filtre passe-bande Par la méthode de Parks-McClellan pour N = 24, 64,128-taps

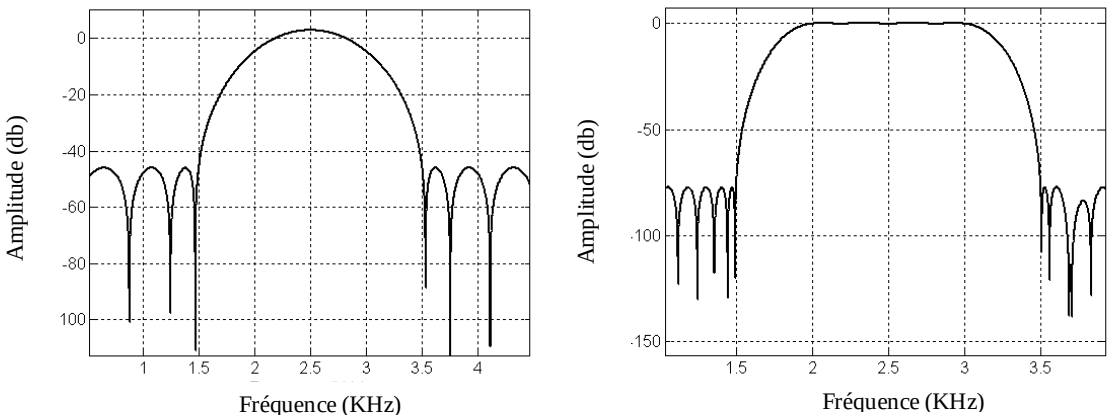
Les spécifications du filtre sont : f_1 = 1500Hz, f_2 = 2000Hz, f_3 = 3000Hz, f_4 = 3500Hz, f_e = 10000Hz : Fréquence d'échantillonnage.

La figure I.18 représente la réponse fréquentielle du filtre RIF passe-bande synthétisé par la méthode de Parks-McClellan pour N = 24, 64,128-taps et la réponse impulsionnelle donnée sur le tableau.I.6.

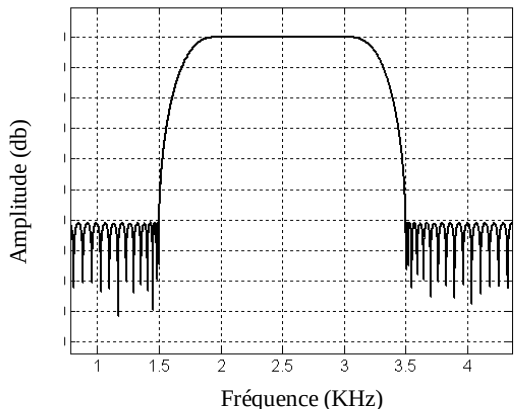
	méthode de Parks-McClellan
N = 24	$h_0 = h_{23} = 0.0003$, $h_1 = h_{22} = -0.0037$, $h_2 = h_{21} = -0.0117$, $h_3 = h_{20} = 0.0244$, $h_4 = h_{19} = 0.0405$, $h_5 = h_{18} = -0.0607$, $h_6 = h_{17} = -0.0851$, $h_7 = h_{16} = 0.1115$, $h_8 = h_{15} = 0.1368$, $h_9 = h_{14} = -0.1585$, $h_{10} = h_{13} = -0.1748$, $h_{11} = h_{12} = 0.1837$.
N = 64	$h_0 = h_{63} = -0.0002$, $h_1 = h_{62} = 0.0006$, $h_2 = h_{61} = 0.0012$, $h_3 = h_{60} = -0.0020$, $h_4 = h_{59} = -0.0029$, $h_5 = h_{58} = 0.0038$, $h_6 = h_{57} = 0.0042$, $h_7 = h_{56} = -0.0042$, $h_8 = h_{55} = -0.0030$, $h_9 = h_{54} = 0.0011$, $h_{10} = h_{53} = -0.0019$, $h_{11} = h_{52} = 0.0055$, $h_{12} = h_{51} = 0.0090$, $h_{13} = h_{50} = -0.0119$, $h_{14} = h_{49} = -0.0127$; $h_{15} = h_{48} = 0.0113$, $h_{16} = h_{47} = 0.0065$, $h_{17} = h_{46} = 0.0007$, $h_{18} = h_{45} = 0.0102$, $h_{19} = h_{44} = -0.0204$, $h_{20} = h_{43} = -0.0289$, $h_{21} = h_{42} = 0.0339$, $h_{22} = h_{41} = 0.0323$, $h_{23} = h_{40} = -0.0234$, $h_{24} = h_{39} = -0.0055$, $h_{25} = h_{38} = -0.0205$, $h_{26} = h_{37} = -0.0530$, $h_{27} = h_{36} = 0.0894$, $h_{28} = h_{35} = 0.1251$, $h_{29} = h_{34} = -0.1572$, $h_{30} = h_{33} = -0.1807$, $h_{31} = h_{32} = 0.1934$.

Tableau.I.6. Réponse impulsionnelle d'un filtre passe-bande RIF par la méthode de Parks-McClellan pour N = 24, 64-taps.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



(a) N = 24-taps (b) N = 64-taps



(a) N =128-taps

Fig.I.18. Synthèse par la méthode de Parks-McClellan.

I.3.6. La méthode des moindres carrés

I.3.6.1. Calcul des coefficients par la méthode des moindres carrés [29,31]

Soit à calculer les N coefficients h_i d'un filtre RIF de manière à ce que la fonction de transfert approche une fonction donnée suivant un critère des moindres carrés. La Transformée de Fourier Discrète appliquée à la suite h_i, avec ($0 \leq i \leq N-1$), fournit une suite H_k telle que:

$$H_k = \frac{1}{N}\sum_{i=0}^{N-1} h_i e^{-j2\pi ik/N} \qquad \text{(I.41)}$$

L'ensemble des H_k, $0 \leq k \leq N-1$, constitue un échantillonnage de la réponse en fréquence du filtre avec le pas f_e/N. Réciproquement les coefficients h_i sont liés à l'ensemble des H_k par la relation.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$$h_i = \sum_{k=0}^{N-1} H_k e^{j2\pi ik/N} \qquad \text{(I.42)}$$

Par suite le problème du calcul des *N* coefficients est équivalent au problème de la détermination de la réponse en fréquence du filtre en *N* points de l'intervalle (0 , f_e). La fonction *H*(*f*) est ensuite obtenue par la formule d'interpolation qui exprime le produit de convolution de la suite d'échantillons $H_k\ \delta(\ f\ -\ k/N\ f_e)$ par la transformée de Fourier de la fenêtre rectangulaire échantillonnée :

$$H(f) = \sum_{k=0}^{N-1} H_k \frac{\sin\left[\pi N(\frac{f}{f_e} - \frac{k}{N})\right]}{N\sin\left[\pi(\frac{f}{f_e} - \frac{k}{N})\right]} \qquad \text{(I.43)}$$

Il faut remarque que cette expression constitue simplement un autre type de développement en série de la fonction *H*(*f*), à nombre limité de termes.

La fonction à approcher *D(f)* étant donnée, une première possibilité consiste à choisir les H_k tels que :

$$H_k = D(\frac{k}{N}.f_e) \quad \text{Pour } 0 \leq k \leq N\text{-}1 \qquad \text{(I.43)}$$

La fonction de transfert du filtre *H(f)*, obtenue par interpolation, présente des ondulations en bandes passante et affaiblie, comme le montre la figure I.19.

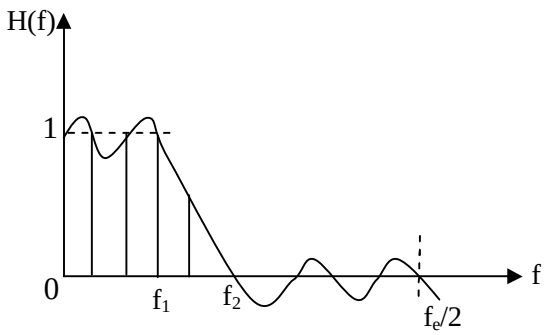


Fig. I.19. Fonction de transfert interpolée.

L'écart entre cette fonction et celle qui est donnée représente une erreur $\varepsilon(f) = H(f) - D(f)$ qu'il est possible de minimiser au sens des moindres carrés.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

La procédure commence par une évaluation de l'erreur quadratique E qui est la norme L_2 de la fonction d'écart. A cet effet la réponse $H(f)$ est échantillonnée avec un pas de fréquence Δ inférieur à f_e/N, de façon à faire apparaître les valeurs interpolées, par exemple :

$\Delta = f_e / NL$ avec L entier supérieur à 1.

La fonction $\varepsilon(f)$ est calculée aux fréquences multiples de Δ.

En général dans l'évaluation de l'erreur quadratique, E une partie seulement de la bande $(0 , f_e/2)$

est à prendre en compte : pour un filtre passe-bas ça peut être la bande passante, la bande affaiblie ou l'ensemble des deux. Pour exposer le principe du calcul, on suppose que la minimisation porte sur la bande passante $(0 , f_1)$ d'un filtre passe-bas. Il vient dans cette hypothèse :

$$E = \sum_{n=0}^{N_0-1} \varepsilon^2(n\frac{f_e}{NL}) \quad \text{avec} \quad (f_1/f_e)\, NL < N_0 \leq (f_1/f_e)\, NL+1 \tag{I.44}$$

De plus il est souvent utile d'affecter un coefficient de pondération $P_0(n)$ à l'élément d'erreur d'indice n, afin de pouvoir modéliser la réponse en fréquence. On obtient alors :

$$E = \sum_{n=0}^{N_o-1} P_0^2(n)\varepsilon^2(n.\frac{f_e}{NL}) = \sum_{n=0}^{N_0-1} P_0^2(n)\varepsilon^2(n) \tag{I.45}$$

La fonction erreur étant obtenue à partir de la formule d'interpolation (I.44), l'erreur quadratique E est fonction de l'ensemble des H_k avec $0 \leq k \leq N$ -1 et est exprimée par : $E(H)$. Si on donne à ces échantillons de la réponse en fréquence des accroissements ΔH_k, on obtient une nouvelle valeur de l'erreur quadratique qui s'exprime par l'égalité :

$$E(H+\Delta H) = E(H) + \sum_{k=0}^{N-1} \frac{\partial E}{\partial H_k}.\Delta H_k + \frac{1}{2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} \frac{\partial^2 E}{\partial H_k \partial H_l}.\Delta H_k.\Delta H_l \tag{I.46}$$

Compte tenu de la relation de définition de E et de la relation d'interpolation, il vient :

$$\frac{\partial E}{\partial H_k} = 2\sum_{n=0}^{N_0-1} P_0^2(n)\varepsilon(n)\frac{\partial \varepsilon(n)}{\partial H_k} \tag{I.47}$$

$$\frac{\partial^2 E}{\partial H_k \partial H_l} = 2\sum_{n=0}^{N_0-1} P_0^2(n)\frac{\partial \varepsilon(n)}{\partial H_l}\frac{\partial \varepsilon(n)}{\partial H_k} \tag{I.48}$$

Ces équation s'écrivent sous une forme matricielle, soit A la matrice à N lignes et N_0 colonnes telle que :

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$$A=\begin{bmatrix} a_{0.0} & a_{0.0} & \dots & a_{(N_0-1).0} \\ a_{1.0} & a_{1.1} & \dots & a_{(N_0-1).1} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{0.(N-1)} & a_{1.(N-1)\dots} & & a_{(N-1)(N_0-1)} \end{bmatrix} \qquad \text{avec} \qquad a_{ij}=\frac{\partial \varepsilon(j)}{\partial H_i}$$

Soit P_0 la matrice diagonale d'ordre N_0 dont les éléments sont les coefficients de pondération $P_0(n)$, il vient :

$$\frac{\partial E}{\partial H_k}=2AP_0^2[\varepsilon(n)] \tag{I.49}$$

L'ensemble des termes $\frac{\partial^2 E}{\partial H_k \partial H_l}$ constitue une matrice carrée d'ordre N telle que :

$$\frac{\partial^2 E}{\partial H_k \partial H_l}=2AP_0^2A^t \tag{I.50}$$

La condition pour que $E(H+\Delta H)$ soit le minimum de la fonction est que toutes ses dérivées par rapport aux $H_k(0 \le k \le N-1)$ s'annulent en ce point. Or:

$$\frac{\partial}{\partial H_k}E(H+\Delta H)=\frac{\partial E}{\partial H_k}+\sum_{l=0}^{N-1}\frac{\partial E}{\partial H_k}\frac{\partial E}{\partial H_l}\Delta H_l \tag{I.51}$$

La condition des moindres carrés s'écrit alors :

$$AP_0^2[\varepsilon(n)]+AP_0^2A^t[\Delta H]=0 \tag{I.52}$$

Dans ces conditions, les accroissements ΔH_k $(0 \le k \le N-1)$ qui permettent de passer des valeurs initiales des échantillons de la réponse en fréquence, aux valeurs optimales forment un vecteur colonne qui s'écrit :

$$\Delta H=-[AP_0^2A^t]^{-1}.AP_0^2[\varepsilon(n)] \tag{I.53}$$

Finalement le calcul des coefficients du filtre par la méthode des moindres carrés demande les opérations suivantes :

1. Echantillonner la fonction à approcher en N point pour obtenir N nombres H_k $(0 \le k \le N-1)$.

2. Dans la bande de fréquences òu l'erreur doit être minimisée, interpoler la réponse entre les H_k pour obtenir N_0 nombres $\varepsilon(n)$ $(0 \le n \le N_0-1)$ qui représentent l'écart entre la réponse du filtre et la fonction à approcher.

3. En fonction des contraintes de l'approximation, déterminer N_0 coefficients de pondération $P_0(n)$

4. Calculer à l'aide de l'équation d'interpolation les éléments de la matrice A.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

5. Résoudre l'équation matricielle qui donne les ΔH_k.

6. Opérer sur l'ensemble des nombres (H_k + ΔH_k) avec $0 \leq n \leq N$ -1 une transformation de Fourier inverse pour obtenir les coefficients du filtre.

Les coefficient de pondération $P_0(n)$ permettent d'introduire des contraintes particulières, par exemple d'obtenir des ondulations en bandes passantes et affaiblies qui soient dans un rapport donné ou encore d'imposer à la réponse en fréquence de passer par un point particulier.

La mise en œuvre de la procédure de calcul ne présente pas de difficultés particulières, elle permet de calculer un filtre d'une manière directe. Cependant le filtre obtenu a des ondulations qui n'ont pas une amplitude constante, or c'est l'objectif qui se rencontre le plus fréquemment. Pour l'atteindre il faut faire appel à une technique itérative.

I.3.6.2. Exemple de synthèse

I.3.6.2.1 Synthèse d'un filtre passe-bande par la méthode de moindre carrée pour N = 24, 64,128-taps

Les spécifications du filtre sont: f_1 = 1000 Hz, f_2 = 1200 Hz, f_3 = 2000Hz, f_4 = 2200 Hz.

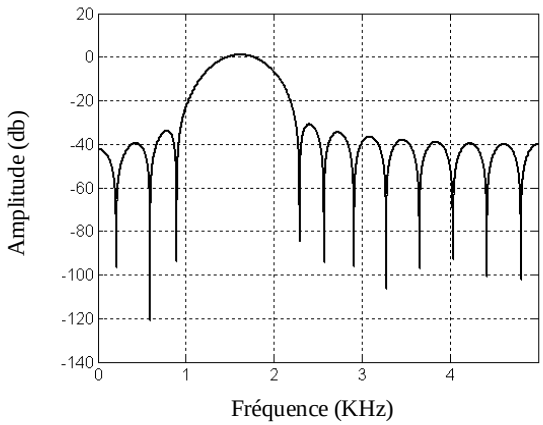
f_e =10000Hz : Fréquence d'échantillonnage.

La figure .I.20 représente la réponse fréquentielle du filtre RIF passe bande synthétisé par la méthode des moindres carrées pour N = 24, 64, 128-taps, et la réponse impulsionnelle donnée sur le tableau.I.7.

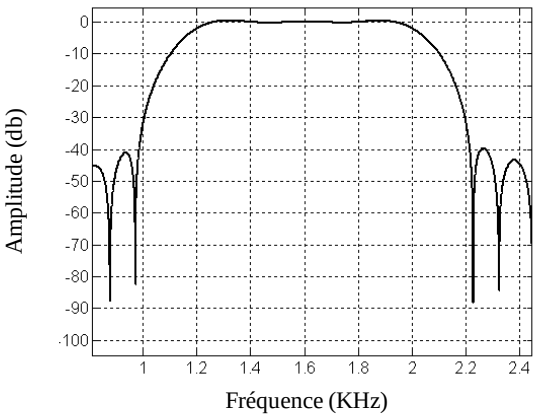
	méthode de moindre carrée
N = 24	$h_0 = h_{23} = 0.0078$, $h_1 = h_{22} = -0.0092$, $h_2 = h_{21} = -0.0356$, $h_3 = h_{20} = -0.0330$, $h_4 = h_{19} = 0.0193$, $h_5 = h_{18} = 0.0798$, $h_6 = h_{17} = 0.0729$, $h_7 = h_{16} = -0.0204$, $h_8 = h_{15} = -0.1186$, $h_9 = h_{14} = -0.1126$, $h_{10} = h_{13} = 0.0088$, $h_{11} = h_{12} = 0.1318$.
N = 64	$h_0 = h_{63} = 0.0037$, $h_1 = h_{62} = 0.0046$, $h_2 = h_{61} = -0.0009$, $h_3 = h_{60} = -0.0088$, $h_4 = h_{59} = -0.0094$, $h_5 = h_{58} = 0.00005$, $h_6 = h_{57} = 0.0099$, $h_7 = h_{56} = 0.0094$, $h_8 = h_{55} = 0.0010$, $h_9 = h_{54} = -0.0028$, $h_{10} = h_{53} = 0.0011$, $h_{11} = h_{52} = 0.0013$, $h_{12} = h_{51} = -0.0101$, $h_{13} = h_{50} = -0.0199$, $h_{14} = h_{49} = -0.0085$, $h_{15} = h_{48} = 0.0194$, $h_{16} = h_{47} = 0.0333$, $h_{17} = h_{46} = 0.0146$, $h_{18} = h_{45} = -0.0155$, $h_{19} = h_{44} = -0.0225$, $h_{20} = h_{43} = -0.0063$, $h_{21} = h_{42} = -0.0014$, $h_{22} = h_{41} = -0.0213$, $h_{23} = h_{40} = -0.0273$, $h_{24} = h_{39} = 0.0198$, $h_{25} = h_{38} = 0.0865$, $h_{26} = h_{37} = 0.0827$, $h_{27} = h_{36} = -0.0247$, $h_{28} = h_{35} = -0.1434$, $h_{29} = h_{34} = -0.1378$, $h_{30} = h_{33} = 0.0111$, $h_{31} = h_{32} = 0.1635$,

Tableau.I.7. Réponse impulsionnelle d'un filtre passe-bande RIF par la méthode des moindres carrées pour N = 24, 64-taps.

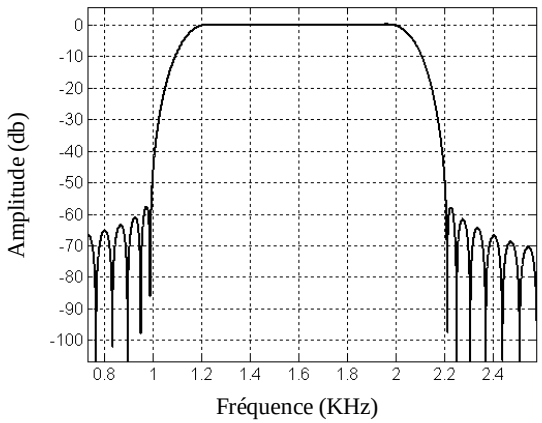
This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



N = 24-taps



N = 64-taps



N = 128-taps

Fig.I.20. Synthèse par la méthode de moindre carrée.

I.3.6.2.2. Exemple d'un filtre passe-bas par la méthode des moindres carrées pour N = 64,128-taps

Les spécifications du filtre sont :

f_1 = 1000 Hz : Fréquence de coupure.

f_2 = 1200 Hz : Limite de la bande atténué.

f_e = 10000Hz : Fréquence d'échantillonnage.

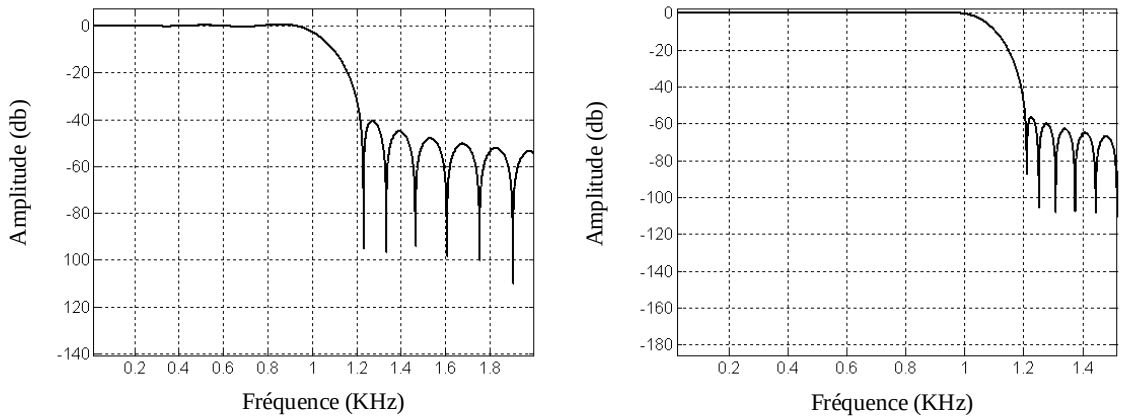
La figure.I.21 représente la réponse fréquentielle du filtre RIF passe-bas synthétisé par la méthode des moindres carrées pour N = 64, 128, et la réponse impulsionnelle donnée sur le tableau.I.8.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

	méthode de moindre carrée
***N* = 64**	$h_0 = h_{63} = 0.0008$, $h_1 = h_{62} = 0.0004$, $h_2 = h_{61} = -0.0010$, $h_3 = h_{60} = -0.0033$, $h_4 = h_{59} = -0.0056$, $h_5 = h_{58} = -0.0069$, $h_6 = h_{57} = -0.0061$, $h_7 = h_{56} = -0.0028$, $h_8 = h_{55} = 0.0022$, $h_9 = h_{54} = 0.0073$, $h_{10} = h_{53} = 0.0103$, $h_{11} = h_{52} = 0.0093$, $h_{12} = h_{51} = 0.0040$, $h_{13} = h_{50} = -0.0039$, $h_{14} = h_{49} = -0.0117$, $h_{15} = h_{48} = -0.0156$, $h_{16} = h_{47} = -0.0131$, $h_{17} = h_{46} = -0.0040$, $h_{18} = h_{45} = 0.0087$, $h_{19} = h_{44} = 0.0202$, $h_{20} = h_{43} = 0.0245$, $h_{21} = h_{42} = 0.0182$, $h_{22} = h_{41} = 0.0015$, $h_{23} = h_{40} = -0.0204$, $h_{24} = h_{39} = -0.0389$, $h_{25} = h_{38} = -0.0441$, $h_{26} = h_{37} = -0.0285$, $h_{27} = h_{36} = 0.0096$, $h_{28} = h_{35} = 0.0650$, $h_{29} = h_{34} = 0.1261$, $h_{30} = h_{33} = 0.1784$, $h_{31} = h_{32} = 0.2085$.
***N* = 128**	$h_0 = h_{127} = -0.0000$, $h_1 = h_{126} = -0.0000$, $h_2 = h_{125} = 0.0000$, $h_3 = h_{124} = 0.0003$, $h_4 = h_{123} = 0.0006$, $h_5 = h_{122} = 0.0007$, $h_6 = h_{121} = 0.0006$, $h_7 = h_{120} = 0.0002$, $h_8 = h_{119} = -0.0003$, $h_9 = h_{118} = -0.0009$, $h_{10} = h_{117} = 0.0012$, $h_{11} = h_{116} = -0.0010$, $h_{12} = h_{115} = -0.0003$, $h_{13} = h_{114} = -0.0003$, $h_{14} = h_{113} = 0.0015$, $h_{15} = h_{112} = 0.0018$, $h_{16} = h_{111} = 0.0013$, $h_{17} = h_{110} = 0.0001$, $h_{18} = h_{109} = -0.0013$, $h_{19} = h_{108} = -0.0024$, $h_{20} = h_{107} = -0.0026$, $h_{21} = h_{106} = -0.0026$, $h_{22} = h_{105} = 0.0004$, $h_{23} = h_{104} = 0.0025$, $h_{24} = h_{103} = 0.0036$, $h_{25} = h_{102} = 0.0036$, $h_{26} = h_{101} = 0.0012$, $h_{27} = h_{100} = -0.0016$, $h_{28} = h_{99} = -0.0041$, $h_{29} = h_{98} = -0.0051$, $h_{30} = h_{97} = -0.0051$, $h_{31} = h_{96} = -0.0004$, $h_{32} = h_{95} = 0.0035$, $h_{33} = h_{94} = 0.0064$, $h_{34} = h_{93} = 0.0066$, $h_{35} = h_{92} = 0.00369$, $h_{36} = h_{91} = -0.0013$, $h_{37} = h_{90} = -0.0064$, $h_{38} = h_{89} = -0.0091$, $h_{39} = h_{88} = -0.0079$, $h_{40} = h_{87} = -0.0026$, $h_{41} = h_{86} = 0.0045$, $h_{42} = h_{85} = 0.0106$, $h_{43} = h_{84} = 0.0125$, $h_{44} = h_{83} = 0.0125$, $h_{45} = h_{82} = 0.0125$, $h_{46} = h_{81} = -0.0098$, $h_{47} = h_{80} = -0.0166$, $h_{48} = h_{79} = -0.0164$, $h_{49} = h_{78} = -0.0083$, $h_{50} = h_{77} = 0.0052$, $h_{51} = h_{76} = 0.0189$, $h_{52} = h_{75} = 0.0260$, $h_{53} = h_{74} = 0.0217$, $h_{54} = h_{73} = 0.0056$, $h_{55} = h_{72} = -0.0174$, $h_{56} = h_{71} = -0.0382$, $h_{57} = h_{70} = -0.0460$, $h_{58} = h_{69} = -0.0321$, $h_{59} = h_{68} = 0.0056$, $h_{60} = h_{67} = 0.0623$, $h_{61} = h_{66} = 0.0623$, $h_{62} = h_{65} = 0.1806$, $h_{63} = h_{64} = 0.2123$,

Tableau.I.8. Réponse impulsionnelle d'un filtre passe-bas RIF par la méthode des moindres carrées pour *N* = 64, 128-taps.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



(a) N = 64-taps (b) N = 128-taps

Fig.1.21. Synthèse par la méthode des moindres carrées.

Conclusion

Dans ce chapitre nous avons décrit les différentes méthodes de synthèse des filtres numériques RIF, la méthode des fenêtres, la méthode d échantillonnage en fréquence, la méthode de Parks-McClellan, et la technique des moindres carrées.

Dans le chapitre suivant, nous allons appliquer la technique de la convolution rapide de Parhi pour développer des algorithmes de filtrage RIF rapides à partir des coefficients de la réponse impulsionnelle du filtre RIF synthétisé par les différentes techniques de synthèse des filtres RIF considérées dans cette étude.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Chapitre II

La Courte Convolution Itérative (ISC) : Application à la représentation d'un filtre RIF

Introduction

Le traitement numérique des signaux suppose la réalisation de certaines opérations telle que : la convolution, la corrélation, la transformée de Fourier, le filtrage, la décimation et l'interpolation des séquences etc. La convolution numérique s'avère un outil très pratique pour le calcul de la réponse d'un système numérique linéaire et invariant, défini par sa réponse impulsionnelle à une séquence quelconque appliquée à son entrée.

Dans ce chapitre, nous allons utiliser la technique de la courte convolution itérative (iterate short convolution) de Parhi pour développer des algorithmes de filtrage RIF rapides.

II.1. Définition de la convolution linéaire [25, 29,42]

Soient deux séquences finies, x et h, représentées par :

$$x = \{x(n), n = 0,\ldots, N\text{-}1\} \tag{II.1}$$

$$h = \{h(n), n = 0,\ldots, L\text{-}1\} \tag{II.2}$$

avec x : la séquence d'entrées.

h : la séquence de réponse impulsionnelle d'un système linéaire invariant. La convolution linéaire est définie par les relations suivantes :

$$y(n) = (x * h)(n) = (h * x)(n) = \sum_{m=0}^{N-1} x(m).h(n-m) = \sum_{m=0}^{L-1} h(m).x(n-m) \tag{II.3}$$

avec $n = 0,\ldots, N + L\text{-}2$.

Si les séquences x(n) et h(n) sont représentées par les polynômes :

$$X(z) = \sum_{i=0}^{N-1} x_i.Z^i \quad \text{avec degré } \{X(z)\} = N\text{-}1 \tag{II.4}$$

$$H(z) = \sum_{i=0}^{L-1} h_i.Z^i \quad \text{avec degré } \{H(z)\} = L\text{-}1 \tag{II.5}$$

avec Z : l'opérateur de transformation en Z.

Alors, la séquence de la convolution $y = x * h$ est représentée par le polynôme :

$$Y(z) = \sum_{i=0}^{N+L-2} y_i.Z^i = X(z).H(z) \tag{II.6}$$

avec degré $\{Y(z)\} = N + L\text{-}2$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

La définition de la convolution linéaire sous forme matricielle est définie comme :

$$Y_{2n-1} = C(n).x(n) \qquad \text{(II.7)}$$

Òu $C(n)$ la matrice de convolution définie par **[41,43]** :

$$C(n) = \begin{bmatrix} h_0 & 0 & \dots & 0 \\ h_1 & h_0 & \dots & 0 \\ \dots & \dots & \dots & \\ h_{n-1} & h_{n-2} & \dots & h_0 \\ 0 & h_{n-1} & \dots & h_1 \\ \dots & \dots & \dots & \\ 0 & 0 & \dots & h_{n-1} \end{bmatrix}$$

La factorisation de la matrice C(n) en plusieurs matrices permet de simplifier l'implémentation de la convolution par des algorithmes rapides.

On peut calculer la convolution linéaire par plusieurs algorithmes comme : l'algorithme de Cook-Toom, l'algorithme de Winograd **[1,25]**, ainsi que la méthode de calcul par transformée de Fourier.

II.2. Description de la technique de la courte convolution itérative [1,2,4,9]

Toute convolution $M \times M$, soit $M = m \times n$ peut être décomposée en une convolution ($m \times m$) et une convolution ($n \times n$). Les algorithmes de calcul peuvent être construits avec des algorithmes rapides de la convolution comme l'algorithme de Cook-Toom ou l'algorithme de Winograd, qui sont respectivement décrits par : $S_{2m-1}=Q_m H_m P_m X_m$ et $S_{2n-1}=Q_n H_n P_n X_n$ respectivement.

Q_m et Q_n sont les matrices de post-addition.

P_m et P_n les matrices de pré-addition.

H_m et H_n les matrices diagonales, définies par : $H_m = diag[P_m . [h_0,h_1,\dots,h_{m-1}]^T]$ et $H_n = diag[P_n . [h_0, h_1,\dots,h_{n-1}]^T]$ respectivement. Ils Déterminent le nombre de multiplications utilisées dans l'algorithme de la courte convolution itérative.

X_m et $[h_0, h_1,\dots, h_{m-1}]^T$ sont deux vecteurs colonnes contenant les deux séquences d'entrée pour la convolution $m \times m$.

X_n et $[h_0, h_1,\dots, h_{n-1}]^T$ sont deux vecteurs colonnes contenant les deux séquences d'entrée pour la convolution $n \times n$. Ces deux convolutions S_{2m-1} et S_{2n-1} sont de longueurs respectives $2m$-1 et $2n$-1 [1].

L'utilisation de l'algorithme mélange de base **[3,41,42,43,44]** permet d'aboutir à l'algorithme de la courte convolution itérative, qui est représenté par l'équation tensorielle suivante :

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

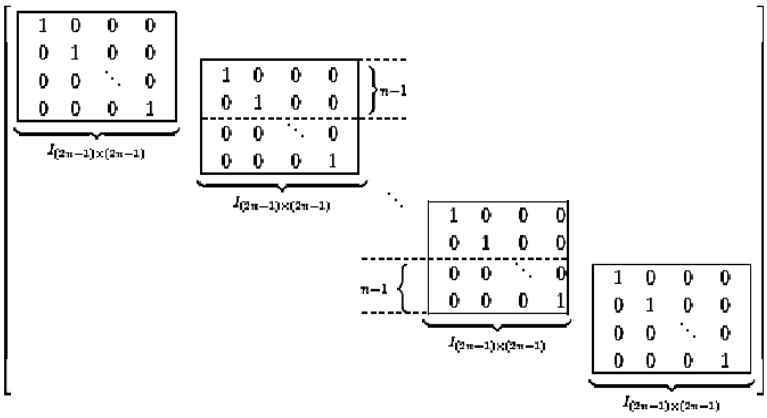
$$S_{2M-1} = A_{M\_mn}(Q_m \otimes Q_n).H_{M\_mn}(Pm \otimes Pn).X_M \qquad \text{(II.8)}$$

Les matrices H_{M_mn} et A_{M_mn} sont respectivement définies en (II.9) et (II.10),

$$H_{M\_mn} = diag[Pm \otimes Pn \,.\, [h_0, h_1, \ldots, h_{M-1}]^T] \qquad \text{(II.9)}$$

A_{M_mn} est de dimension de $2M$-1 par $(2m$-1$)($ $2n$-1$)$ **[1 ,42]**

$A_{M\_mn} =$



(II.10)

$\otimes$ Représente le produit tensoriel.

La convolution $N \times N$ ($N = Mk = mnk$) peut être encore décomposée en trois convolutions de dimensions respectives $m \times m$, $n \times n$ et $k \times k$.

Un des algorithmes de la convolution itérative résultants peut être représenté par **[1]** :

$$S_{2N-1} = A_{N\_Mk}(A_{M\_mn} \otimes I_{(2k-1)x(2k-1)})(Q_m \otimes (Qn \otimes Q_k)).H_{N\_mnk}(P_m \otimes (Pn \otimes P_k).X_N \qquad \text{(II.11)}$$

Où $$H_{N\_mnk} = diag\,[(P_m \otimes (Pn \otimes P_k)).[h_0, h_1, \ldots, h_{N-1}]^T \qquad \text{(II.12)}$$

Nous pouvons décomposer une longue convolution en n'importe quelle combinaison de deux ou trois convolutions courtes par l'utilisation de la convolution itérative (II.11).

Soit une convolution linéaire $L \times L$ ($L=L_1L_2\ldots L_r$), elle peut être décomposée en (r) courtes convolutions **[1,2,4]**.

Un des algorithmes de la convolution itérative résultants est représenté par :

$$S_{2N-1} = A_L\,(Q_{L1} \otimes (\ldots (Q_{Lr-1} \otimes Q_{Lr}))).H_L\,(P_{L1} \otimes (\ldots (P_{Lr-1} \otimes P_{Lr}))).X_N \qquad \text{(II.13)}$$

A_L est calculé en utilisant le procédé suivant:

$A_1 = A\,(L_1, L_2)$.

$A_2 = A\,(L_1L_2, L_3).\,(A_1 \otimes I_{(2L3-1)\,x\,(2L2-1)})$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

……

$A_{r\text{-}1} = A\,(L_1 L_2 \ldots L_{r-1}, L_r).(A1 \otimes I_{(2Lr-1)x(2Lr-1)})$ (II.14)

$A_L = A_{r-1}$.

L'équation (II.13) est l'algorithme de la courte convolution itérative proposé par Parhi **[1,2,4]**. L'algorithme mélangé de base **[41,42]** combine deux convolutions courtes pour obtenir une plus longue convolution, tandis que l'algorithme de la courte convolution itérative de Parhi peut combiner tous les nombres de convolutions courtes et il est ainsi plus efficace.

II.3. Algorithme parallèle rapide du filtre RIF basé sur la courte convolution itérative [1,2,4]

La structure de la convolution itérative (II.13) peut être transposée pour obtenir une architecture parallèle rapide du filtre RIF.

Soit un filtre RIF L-Parallèle (avec $L = L_1 L_2 \ldots L_r$) à N-taps basé sur la convolution itérative $L_i \times L_i$.

$S_{2Li-1} = Q_{Li} H_{Li} P_{Li} X_{Li}$ avec $(i = 1, 2, ..., r)$, l'algorithme peut être exprimé sous la forme :

$$Y_L = P^T H_L Q^T A^T X_L \tag{II.15}$$

Avec

$$Y_L = \left[Y_{L-1}\ Y_{L-2} \ldots Y_0\right]^T \tag{II.16}$$

$$X_L = \left[X_{L-1} \ldots X_1\ X_0\ Z^{-L} X_{L-1} \ldots Z^{-L} X_2\ Z^{-L} X_1\right]^T \tag{II.17}$$

$X_i (i = 0,1,..., L-1)$ la séquence d'entrées qui contient les échantillons x_{Lk+i}, $(k = 0,1,2...)$

$$H_L = diag\left[P \times \left[H_0, H_1, \ldots H_{L-1}\right]^T\right] \tag{II.18}$$

$$P = (P_{m1 \times n1} \otimes (P_{m2 \times n2} \otimes (\ldots (P_{mr-1 \times nr-1} \otimes P_{mr \times nr})))) \tag{II.19}$$

Hi, $(i = 0,1,..., L-1)$ le sous filtre qui contient les coefficients h_{Lk+i}, $k = (0,1,2...)$

$$P^T = (P^T{}_{m1 \times n1} \otimes (P^T{}_{m2 \times n2} \otimes (\ldots (P^T{}_{mr-1 \times nr-1} \otimes P^T{}_{mr \times nr})))) \tag{II.20}$$

$$Q^T = (Q^T{}_{m1 \times n1} \otimes (Q^T{}_{m2 \times n2} \otimes (\ldots (Q^T{}_{mr-1 \times nr-1} \otimes Q^T{}_{mr \times nr})))) \tag{II.21}$$

A^T définie par l'équation (II.14).

Le calcul de Q_m, Q_n,, P_m et P_n se fait par un algorithme de la convolution rapide comme l'algorithme de Cook- Toom ou l'algorithme de Winograd, ou d'une autre façon qui mène à la factorisation de la matrice de convolution H en plusieurs sous matrices.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Exemple1 :

Soit le produit de deux nombres complexes $(a + jb).(c + dj) = e+jf$, on peut l'exprimer sous forme d'un produit matriciel qui exige 4 multiplications et 2 additions :

$$\begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} c & -d \\ d & c \end{bmatrix} . \begin{bmatrix} a \\ b \end{bmatrix} \tag{II.22}$$

Le nombre de multiplications peut être réduit à 3 au dépend de 3 additions supplémentaires en utilisant les relations : $\begin{cases} ac - bd = a(c-d) + d(a-b) \\ ad + bc = b(c+d) + d(a-b) \end{cases}$ (II.23)

Afin de l'exprimer sous forme matricielle, sa matrice coefficient peut être décomposée en un produit de trois matrices : C (2 × 3), H (3 × 3) et D (3 × 2):

$$s = \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} c-d & 0 & 0 \\ 0 & c+d & 0 \\ 0 & 0 & d \end{bmatrix} . \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix} . \begin{bmatrix} a \\ b \end{bmatrix} = C.H.D.x \tag{II.24}$$

Où C est une matrice de post-addition (exigeant 2 additions).

D est une matrice pré-addition (exigeant 1 addition).

H est une matrice diagonale (exigeant 2 additions pour obtenir ses éléments diagonaux)

– Ainsi, la complexité arithmétique est réduite à 3 multiplications et 3 additions (sans inclure les additions dans la matrice H).

On distingue deux approches bien connues pour le développement des algorithmes rapides de la convolution de longueur courte :

a) l'algorithme de Cook -Toom (basé sur l'interpolation de Lagrange).

b) et l'algorithme de Winograd (basé sur le théorème du reste chinois).

II.4. Algorithme de Cook-Toom [25,44,45,46]

II.4.1. Définition

C'est un algorithme de la convolution linéaire pour la multiplication polynômiale basée sur le théorème d'interpolation de Lagrange.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

II.4.2. Théorème de l'interpolation de Lagrange

Soit $\beta_{0,\ldots,}\beta_n$ un ensemble de n+1 points distincts, et $f(\beta_i)$ donné pour $i = 0, 1, \ldots, n$. Il existe un polynôme $f(p)$ de degré inférieur ou égal à n et qui a la valeur $f(\beta_i)$ lorsqu'on l'évalue en β_i pour $i = 0, 1, \ldots, n$. Ce polynôme $f(p)$ est donné par :

$$f(p) = \sum_{i=0}^{n} f(\beta_i) \frac{\prod_{j \neq i} (p - \beta_i)}{\prod_{j \neq i} (\beta_i - \beta_j)} \qquad \text{(II.25)}$$

II.4.3. L'application du théorème de l'interpolation de Lagrange dans la convolution linéaire [25,46]

Considérons une séquence $h = \{h_0, h_1,\ldots, h_{N-1}\}$ de N *points*, et une séquence $x = \{x_0, x_1,\ldots,x_{L-1}\}$ de L points, la convolution de h et x est exprimé en termes de multiplication des polynômes comme : $s(p) = h(p)\, x(p)$.

Où $h(p) = h_{N-1}p^{N-1}+\ldots+h_1p+h_0$.

$x(p) = x_{L-1}p^{L-1}+\ldots+x_1p+x_0$.

$s(p) = s_{L+N-2}\, p^{L+N-2} +\ldots+s_1p+s_0$

Le polynôme de sortie $s(p)$ est de degré $L+N$-2, et a $L+N$ - 1 points différents.

$s(p)$ peut être déterminé d'une manière unique à partir de ses valeurs en $L+N$ -1 points distincts.

Soit $\{\beta_0, \beta_{1,\ldots,}\beta_{L+N-2}\}$ les $L+N$-1 nombres réels différents, Si $s(\beta_i)$ pour $i = \{0,1,\ldots, L+N-2\}$ sont connus, alors $s(P)$ peut être calculé en utilisant le théorème d'interpolation de Lagrange:

$$s(p) = \sum_{i=0}^{L+N-2} s(\beta_i) \frac{\prod_{j \neq i} (p - \beta_i)}{\prod_{j \neq i} (\beta_i - \beta_j)} \qquad \text{(II.26)}$$

On peut montrer que cette équation est la solution unique pour calculer la convolution linéaire pour $s(p)$ étant donné les valeurs de $s(\beta_i)$, pour $i = \{0,1,\ldots, L+N-2\}$.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

II.4.4. Description de l'algorithme Cook-Toom

1. Choisir $L+N-1$ nombres réels différents $\beta_0, \beta_1, \ldots, \beta_{L+N-2}$

2. Calculer $h(\beta_i)$ et $x(\beta_i)$, pour $i = \{0,1,\ldots,L+N-2\}$

3. Calculer $s(\beta_i) = h(\beta_i).x(\beta_i)$, pour $i = \{0,1,\ldots,L+N-2\}$

4. Calcul de $s(p)$ en utilisant :

$$s(p) = \sum_{i=0}^{L+N-2} s(\beta_i) \frac{\prod_{j \neq i} (p - \beta_i)}{\prod_{j \neq i} (\beta_i - \beta_j)} \qquad \text{(II.27)}$$

II.4.5. Complexité de l'Algorithme Cook-Toom

- Le but de l'algorithme de convolution rapide est de réduire la complexité de multiplication. Ainsi, si les β_i (i=0,1,…, $L+N-2$) sont choisis correctement, le calcul à la seconde étape implique quelques additions et multiplications par des petites constantes.

- Les multiplications sont utilisées à la troisième étape seulement pour calculer $s(\beta_i)$, le nombre de multiplications nécessaires est donc $L+N-1$.

- Par l'algorithme de Cook –Toom, le nombre de multiplications est réduit de LN à $L+N-1$ aux dépend d'une augmentation du nombre d'additions.

- Un additionneur a une surface d'intégration et un temps de calcul inférieurs à un multiplicateur. Ainsi, l'algorithme de Cook –Toom peut mener à un gain considérable dans la complexité matérielle (VLSI) et génère une implémentation efficace en termes de temps de calcul.

Exemple 2 :

Construction d'un algorithme de la convolution 2 × 2 en utilisant l'algorithme de Cook -Toom avec $\beta = \{0, 1, -1\}$. Ecrire la convolution 2 × 2 sous la forme d'une multiplication polynômiale :

$s(p) = h(p)\,x(p)$, où $h(p) = h_0 + h_1 p$ et $x(p) = x_0 + x_1 p$

$s(p) = s_0 + s_1 p + s_2 p$

Le calcul direct exige 4 multiplications et 1 addition selon l'expression :

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} h_0 & 0 \\ h_1 & h_0 \\ 0 & h_1 \end{bmatrix} . \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \qquad \text{(II.28)}$$

Nous employons l'algorithme Cook-Toom pour obtenir un calcul efficace de la convolution avec une réduction du nombre de multiplications.

$\beta_0 = 0, \quad h(\beta_0) = h_0, \quad x(\beta_0) = x_0.$
$\beta_1 = 1, \quad h(\beta_1) = h_0 + h_1, \quad x(\beta_0) = x_0 + x_1.$
$\beta_1 = -1, \quad h(\beta_2) = h_0 - h_1, \quad x(\beta_2) = x_0 - x_1.$

$s(\beta_0)$, $s(\beta_1)$, et $s(\beta_2)$ sont calculés, en utilisant 3 multiplications:

$s(\beta_0) = h(\beta_0)\, x(\beta_0), \qquad s(\beta_1) = h(\beta_1)\, x(\beta_1), \qquad s(\beta_2) = h(\beta_2)\, x(\beta_2).$

D'après le théorème d'interpolation de Lagrange, nous obtenons :

$$s(p) = s(\beta_0)\frac{(p-\beta_1)(p-\beta_2)}{(\beta_0-\beta_1)(\beta_0-\beta_2)} + s(\beta_1)\frac{(p-\beta_0)(p-\beta_1)}{(\beta_1-\beta_0)(\beta_1-\beta_2)} + s(\beta_2)\frac{(p-\beta_0)(p-\beta_1)}{(\beta_2-\beta_0)(\beta_2-\beta_1)}.$$

$$s(p) = s(\beta_0) + p\left(\frac{s(\beta_1)-s(\beta_2)}{2}\right) + p^2\left(-s(\beta_0) + \frac{s(\beta_1)+s(\beta_2)}{2}\right).$$

Ce qui conduit à la forme matricielle suivante :

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix} . \begin{bmatrix} s(\beta_0) \\ s(\beta_1)/2 \\ s(\beta_2)/2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} h(\beta_0) & 0 & 0 \\ 0 & h(\beta_1)/2 & 0 \\ 0 & 0 & h(\beta_2)/2 \end{bmatrix} \begin{bmatrix} x(\beta_0) \\ x(\beta_1) \\ x(\beta_2) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} h_0 & 0 & 0 \\ 0 & (h_0+h_1)/2 & 0 \\ 0 & 0 & (h_0-h_1)/2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} h_0 & 0 & 0 \\ 0 & (h_0+h_1)/2 & 0 \\ 0 & 0 & (h_0-h_1)/2 \end{bmatrix} \begin{bmatrix} x_0 \\ x_0+x_1 \\ x_0-x_1 \end{bmatrix} \text{ (\textit{deux additions})}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} h_0.x_0 \\ [(h_0+h_1)/2].(x_0+x_1) \\ [(h_0-h_1)/2].(x_0-x_1) \end{bmatrix} \text{ (\textit{troix multiplications})}$$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$$= \begin{bmatrix} h_0.x_0 \\ [(h_0+h_1)/2].(x_0+x_1)-[(h_0-h_1)/2].(x_0-x_1) \\ -h_0.x_0+\ [(h_0+h_1)/2].(x_0+x_1)+[(h_0-h_1)/2].(x_0-x_1) \end{bmatrix} \text{ (troix additions)}$$

Le calcul est effectué comme suit (5 additions, 3 multiplications)

1. $H_0 = h_0$, $H_1 = (h_0 + h_1)/2$, $H_2 = (h_0 - h_1)/2$, (pré-calculé).

2. $X_0 = x_0$, $X_1 = x_0 + x_1$, $X_2 = x_0 - x_1$, (2 additions).

3. $S_0 = H_0X_0$, $S_1 = H_1X_1$, $S_2 = H_2X_2$, (3 multiplications).

4. $s_0 = S_0$, $s_1 = S_1 - S_2$, $s_2 = -S_0 + S_1 + S_2$, (3 additions).

Par conséquent, cet algorithme a besoin de 3 multiplications et 5 additions (en ignorant les additions dans le pré calcul), c-à-d, le nombre de multiplications est réduit de 1 aux dépens de 4 additions supplémentaires.

- Quelques additions dans les matrices pré-addition ou post-addition peuvent être partagées. Ainsi, quand nous comptons le nombre d'additions, nous comptons seulement une au lieu de deux ou trois.

- Si nous prenons h_0, h_1 en tant que coefficients du filtre RIF et x_0, x_1 comme la séquence du signal d'entrée, alors les termes H_0, H_1 n'ont pas besoin d'être recalculées chaque fois que le filtre est utilisé, Ils peuvent être pré-calculé une seule fois en différé et stocké, donc, nous ignorons ces calculs en comptant le nombre d'opérations.

D' après l'exemple 2, nous pouvons comprendre l'algorithme de Cook-Toom comme une décomposition matricielle.

Remarque:

Une convolution 2×2 est exprimée sous forme du produit d'une matrice par un vecteur :

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} h_0 & 0 \\ h_1 & h_0 \\ 0 & h_1 \end{bmatrix} . \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \quad ou \quad s = T.x \qquad \text{(II.29)}$$

Généralement, l'équation peut être exprimée comme: $S = T.\,x = C.\,H.\,D.\,x$

Où C est la matrice de post-addition.

D est la matrice de pré-addition.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

H est une matrice diagonale avec H_i, $i = 0, 1, \ldots, L+N-2$ sur la diagonale principale.

D'après $T = C.H.D$, il implique que l'algorithme Cook-Toom fournit une manière de factoriser la matrice T de convolution en une multiplication de trois matrices, une matrice C de post-addition, une matrice diagonale H et une matrice D de pré-addition, tels que tout le nombre de multiplications est déterminé seulement par les éléments différents de zéro sur la diagonale principale de la matrice diagonale H.

Bien que le nombre de multiplications soit réduit, le nombre d'additions a augmenté. L'algorithme Cook-Toom peut être modifié afin de réduire d'avantage le nombre d'additions.

II.4.6. Algorithme Cook-Toom modifié [25,46]

II.4.6.1. Description De l'algorithme Cook-Toom modifié

On définit $s'(p) = s(p) - s_{L+N-2}\, p^{L+N-2}$. Noter que le degré de $s(p)$ est $L + N - 2$ et s_{L+N-2} est son coefficient d'ordre le plus élevé, Par conséquent le degré de $s'(p)$ est $L + N - 3$.

Considérons maintenant l'algorithme Cook-Toom modifié :

1. Choisir $L+N-2$ nombres réels différents $\beta_0, \beta_1, \ldots, \beta_{L+N-3}$.

2. Calculer $h(\beta_0)$ et $x(\beta_0)$, pour $i = 0, 1, \ldots, L+N-3$.

3. Calculer $s(\beta_i) = h(\beta_i).x(\beta_i)$, pour $i = 0, 1, \ldots, L+N-3$.

4. Calculer $s'(\beta_i) = s(\beta_i) - s_{L+N-2}.\beta_i^{L+N-2}$ pour $i = 0, 1, \ldots, L+N-3$.

5. Calcul de $s'(p)$ en utilisant :

$$s'(p) = \sum_{i=0}^{L+N-2} s'(\beta_i) \frac{\prod_{j \neq i} (p - \beta_i)}{\prod_{j \neq i} (\beta_i - \beta_j)} \qquad \text{(II.30)}$$

6. Calculer $s(\beta_i) = s'(\beta_i) + s_{L+N-2}\, \beta_i^{L+N-2}$.

Exemple 3 : Déduire un algorithme de la convolution 2×2 en utilisant l'algorithme Cook-Toom modifié avec $\beta = \{0,-1\}$.

Considérons l'interpolation de Lagrange pour $s'(p) = s(p) - h_1.x_1.p^2$ avec $\{\beta_0 = 0, \beta_1 = -1\}$.

Premièrement, on trouve $s'(p) = h(\beta_i)\, x(\beta_i) - h_1\, x_1\, \beta_i^2$.

$\beta_0 = 0$, $\quad h(\beta_0) = h_0$, $\quad x(\beta_0) = x_0$.

$\beta_1 = -1$, $\quad h(\beta_1) = h_0 - h_1$, $\quad x(\beta_1) = x_0 - x_1$.

et $\quad s'(\beta_0) = h(\beta_0)\, x(\beta_0) - h_1\, x_1.\, \beta_0^2 = h_0\, x_0$.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$s'(\beta_1) = h(\beta_1)\, x(\beta_1) - h_1 . x_1\, \beta_1{}^2 = (h_0 - h_1)(x_0 - x_1) - h_1\, x_1.$

Ce qui demande 2 multiplications (sans compter les multiplications $h_1.x_1$). En appliquant l'algorithme d'interpolation de Lagrange, nous obtenons :

$$s'(p) = s'(\beta_0)\frac{(p-\beta_1)}{(\beta_0-\beta_1)} + s'(\beta_1)\frac{(p-\beta_0)}{(\beta_1-\beta_0)}.$$

$$s'(p) = s'(\beta_0) + p(s'(\beta_0) - s'(\beta_1)).$$

Par conséquent $s(p) = s'(p) + h_1 x_1 p^2 = s_0 + s_1 p + s_2 p^2$.

Enfin, nous avons l'expression matricielle :

Noter que $\begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} s'(\beta_0) \\ s'(\beta_1) \\ h_1.x_1 \end{bmatrix}$

Donc $\begin{bmatrix} s'(\beta_0) \\ s'(\beta_1) \\ h_1 x_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} s(\beta_0) \\ s(\beta_1) \\ h_1 x_1 \end{bmatrix}$

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} s(\beta_0) \\ s(\beta_1) \\ h_1 x_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} h_0 & 0 & 0 \\ 0 & h_0 - h_1 & 0 \\ 0 & 0 & h_1 \end{bmatrix} . \begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} . \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

Ce qui donne : $S_3 = P_2.H_2.Q_2.X_2$.

- Le calcul est effectué comme suit:

1. $H_0 = h_0$, $H_1 = (h_0 - h_1)$, $H_2 = h_1$, (pré calculée).

2. $X_0 = x_0$, $X_1 = x_0 - x_1$, $X_2 = x_1$, (1 addition).

3. $S_0 = H_0 X_0$, $S_1 = H_1 X_1$, $S_2 = H_2 X_2$, (3 multiplications).

4. $s_0 = S_0$, $s_1 = S_0 - S_1 + S_2$, $s_2 = S_2$, (2 additions).

Le nombre total d'opérations est 3 multiplications et 3 additions.

Par comparaison à l'algorithme de la convolution de l'exemple 2, le nombre d'additions a été réduit à 2 tandis que le nombre de multiplications reste le même.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Remarque :

L'algorithme Cook-Toom est efficace en termes du nombre de multiplications, toutefois lorsque la taille du problème augmente, il n'est pas efficace parce que le nombre d'additions augmente considérablement si β prend les valeurs différentes de $\{0, \pm1, \pm2, \pm4\}$. Ce qui peut résulter en une complication des matrices de pré-addition et post-addition.

Pour des problèmes de grande taille, l'algorithme de Winograd est plus efficace.

II.5. Algorithme de Winograd [25,45,46]

II.5.1. Définition

L'algorithme de Winograd est basé sur le Théorème des restes Chinois (CRT). Il est possible de déterminer uniquement un nombre entier non négatif étant donné seulement son reste par rapport aux modules donnés, à condition que les modules soient relativement premiers et le nombre entier est supposé être plus petit que le produit des modules.

-Théorème des restes Chinois pour des nombres entiers

Soit $C_i = R_{mi}[C]$(représente le reste de C lorsqu'il est divisé par m_i), pour $i = 0,1,\ldots,k$.

$$C = \left(\sum_{i=0}^{k} C_i N_i M_i \right) \bmod M \text{ , ou } M = \Pi_{i=o}^{k} m_i, \quad M_i = M / m_i .$$

et N_i est la solution de $N_i M_i + n_i m_i = GCD(M_i, m_i) = 1$, à condition que $0 \leq C < M$.

II.5.2.Théorème des restes Chinois (CRT) pour des polynômes

Soit $C^{(i)}(p) = R_{m^{(i)}}(p)\left[C(p)\right]$ pour $i = 0, 1, \ldots, k$ ou $m^{(i)}(p)$

Sont relativement premier, alors

$$C(p) = \left(\sum_{i=0}^{k} C^{(i)}(p).N^{(i)}(p).M^{(i)}(p) \right) \bmod M(p) \qquad \text{(II.31)}$$

$M(p) = \prod_{i=0}^{k} m^{(i)}(p)$, $M^{(i)}(p) = M(p) / m^{(i)}(p)$ *et* $N^{(i)}(p)$ est la solution de l'équation :

à condition que le degré de $N^{(i)}(p)M^{(i)}(p) + n^{(i)}(p)m^{(i)}(p) = GCD(M^{(i)}(p), m^{(i)}(p)) = 1$.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$C(p)$ soit inférieur au degré de $M(p)$.

Exemple 4 :

En utilisant le théorème CRT pour les entiers, on choisit les modules :

$m_0 = 3,\ m_1 = 4,\quad m_2 = 5,\qquad M = m_0.m_1.m_2 = 60$ et $M_i = M/m_i$.

Alors $m_0 = 3,\qquad M_0 = 20,\qquad (-1)20 + (3)7 = 1.$

$m_1 = 4,\qquad M_1 = 15,\qquad (-1)15 + (4)\,4 = 1.$

$m_2 = 5,\qquad M_2 = 12,\qquad (-2)12 + (5)\,5 = 1.$

Où N_i et n_i sont obtenus en utilisant l'algorithme Euclidien *GCD*.

Le nombre entier *C* doit satisfaire les conditions: $0 \le C < M,$ et $C_i = R_{mi}[C]$.

Le nombre entier *C* peut être calculé comme suit :

$$C = \left(\sum_{i=0}^{k} C_i N_i M_i\right).\mathrm{mod}\, M = (-20*C_0 - 15*C_1 - 24*C_2)\bmod 60.$$

Pour C = 17, $C_0 = R3(17) = 2,$ $C_1 = R4(17) = 1,$ $C_2 = R5(17) = 2.$

C = (-20*2 – 15*1 – 24*2).mod60 = (- 103).mod60 = 17.

- CRT pour des polynômes : Le reste d'un polynôme par rapport au module $p' + f(p)$, où $\deg(f(p)) \le i - 1$, peut être évalué par la substitution de p' par $- f(p)$ dans le polynôme.

Exemple 5 :

$R_{x+2}[5x^2 + 3x + 5] = 5(-2)^2 + 3(-2) + 5 = 19.$

$R_{x^2+2}[5x^2 + 3x + 5] = 5(-2) + 3x + 5 = 3x - 5.$

$R_{x^2+x+2}[5x^2 + 3x + 5] = 5(-x - 2) + 3x + 5 = -2x - 5.$

II.5.3. Description de l'Algorithme

1. Choisir un polynôme $m(p)$ de degré plus élevé que le degré de $h(p).x(p)$, et le factoriser en $(k+1)$ polynômes relativement premiers de coefficients réels, c'est-à-dire :

$m(p) = m^{(0)}m^{(1)}...m^{(k)}(p)$.

2. Soit $M^{(i)}(p) = m^{(p)} / m^{(i)}(p)$ Utiliser l'algorithme Euclidien GCD pour résoudre l'équation :

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$N^{(i)}(p)M^{(i)}(p)+n^{(i)}(p)m^{(i)}(p)=1$ pour $N^{(i)}(p)$.

3. Calculer $h^{(i)}(p)=h(p)\bmod m^{(i)}(p),\quad x^{(i)}(p)=x(p)\bmod m^{(i)}(p)$ pour i = 0, 1, …, k.

4. Calculer $s^{(i)}(p)=h^{(i)}(p).x^{(i)}(p).\bmod m^{(i)},\quad pour\ \ i=0,1,...,k$.

5. Calculer $s(p)$ en utilisant $s(p)=\sum_{i=0}^{k}s^{(i)}(p)N^{(i)}(p)M^{(i)}\bmod m^{(i)}(p)$.

Exemple 6:

Considérons la convolution linéaire 2 × 3, Construire une réalisation efficace en utilisant l'algorithme de Winograd.

Avec $m(p)=p(p-1)(p^2+1)$.

$m^{(o)}(p)=p,\quad m^{(1)}(p)=p-1,\quad m^{(2)}(p)=p^2+1.$

On construit le tableau suivant en utilisant les relations

$M^{(i)}(p)=m(p)/m^{(i)}(p)$ et $N^{(i)}(p)M^{(i)}(p)+n^{(i)}(p)m^{(i)}(p)=1$ pour i = 0, 1, 2.

i	$m^{(i)}(p)$	$M^{(i)}(p)$	$n^{(i)}(p)$	$N^{(i)}(p)$
0	p	p^3-p^2+p-1	p^2-p+1	-1
1	$p-1$	p^3+p	$-(½)\ (p^2+p+2)$	1/2
2	p^2+1	p^2-p	$-(½)\ (p-2)$	$(½)\ (p-1)$

On Calcule les résidus à partir de :

$h(p) = h_0 + h_1 p,\qquad x(p) = x_0 + x_1 p + x_2 p^2.$

$h^{(0)}(p) = h_0,\qquad x^{(0)}(p) = x_0.$

$h^{(1)}(p) = h_0 + h_1,\qquad x^{(1)}(p) = x_0 + x_1 + x_2.$

$h^{(2)} = h_0 + h_1 p,\qquad x^{(2)} = (x_0 - x_2) + x_1 p.$

$s^{(0)}(p)=h_0x_o=s_0^{(0)},\ s^{(1)}(p)=(h_0+h_1)(x_0+x_1+x_2)=s_0^{(1)}.$

$s^{(2)}(p)=(h_0+h_1p)((x_0-x_2)+x_1p)\bmod(p^2+1).$

$=h_0(x_0-x_2)-h_1x_1+(h_0x_1+h_1(x_0-x_2))p.$

$=s_0^{(2)}+s_1^{(2)}p.$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Alors

$$s(p) = \sum_{i=0}^{2} s^{(i)}(p) N^{(i)}(p) M^{(i)}(p) \bmod m^{(i)}(p).$$

Par

$$s(p) = \left[-s^{(o)}(p)(p^3 - p^2 + p - 1) + \frac{s^{(1)}(p)}{2}(p^3 + p) + \frac{s^{(2)}(p)}{2}(p^3 - 2p^2 + p) \right] \bmod (p^4 - p^3 + p^2 - p).$$

substitution de $s^{(0)}(p)$, $s^{(1)}(p)$, $s^{(2)}(p)$ dans $s(p)$,on obtient le tableau suivant :

p^0	p^1	p^2	p^3
$s_0^{(0)}$	$-s_0^{(0)}$	$s_0^{(0)}$	$-s_0^{(0)}$
0	$(½)s_0^{(1)}$	0	$(½)s_0^{(1)}$
0	$(½)s_0^{(2)}$	$-s_0^{(2)}$	$(½)s_0^{(2)}$
0	$(½)s_1^{(2)}$	0	$(-½)s_1^{(2)}$

Par conséquent, on a :

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 1 \\ 1 & 0 & -2 & 0 \\ -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} s_0^{(0)} \\ (1/2)s_0^{(1)} \\ (1/2)s_0^{2} \\ (1/2)s_1^{(2)} \end{bmatrix}$$

$$\begin{bmatrix} s_0^{(0)} \\ \frac{1}{2}s_0^{(1)} \\ \frac{1}{2}s_0^{(2)} \\ \frac{1}{2}s_1^{(2)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} h_0 & 0 & 0 & 0 & 0 \\ 0 & \frac{h_0+h_1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{h_0}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{h_1-h_0}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{h_0+h_1}{2} \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_0 + x_1 + x_2 \\ x_0 + x_1 - x_2 \\ x_0 - x_2 \\ x_1 \end{bmatrix}$$

Finalement, on obtient :

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 2 & 1 & -1 \\ 1 & 0 & -2 & 0 & 2 \\ -1 & 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} h_0 & 0 & 0 & 0 & 0 \\ 0 & \frac{h_0+h_1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{h_0}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{h_1-h_0}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{h_0+h_1}{2} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

- Dans cet exemple, l'algorithme de convolution de Winograd exige 5 multiplications et 11 additions comparées à 6 multiplications et 2 additions pour l'implémentation directe.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Remarque:

– Le nombre de multiplications impliquées dans l'algorithme de Winograd dépend fortement du degré de chaque $m^{(i)}(p)$. Par conséquent, le degré de $m(p)$ devrait être aussi petit que possible.

– Une forme plus efficace (ou une forme modifiée) de l'algorithme de Winograd peut être obtenu en posant $deg[m(p)] = deg[s(p)]$ et en appliquant le Théorème des restes Chinois (CRT) à : $s'(p) = s(p) - h_{N-1} X_{L-1} m(p)$.

II.5.4. Algorithme modifié de Winograd [25,46]

II.5.4. 1. Description:

1. Choisir un polynôme $m(p)$ de degré égal au degré de $s(p)$ et le factoriser en $(k+1)$ polynômes, relativement premiers de coefficients réels.

c.à.d, $m(p) = m^{(0)}(p)m^{(1)}(p)\ldots m^{(k)}(p)$.

2. Soit $M^{(i)}(p) = m(p)/m^{(i)}(p)$, utiliser l'algorithme Euclidien GCD pour résoudre l'équation

$N^{(i)}(p)M^{(i)}(p) + n^{(i)}(p)m^{(i)}(p) = 1$ pour $N^{(i)}(p)$.

3. Calculer $h^{(i)}(p) = h(p) \bmod m^{(i)}(p)$, $x^{(i)}(p) = x(p) \bmod m^{(i)}(p)$ pour $i = 0, 1, \ldots, k$.

4. Calculer $s'^{(i)}(p) = h^{(i)}(p)x^{(i)}(p) \bmod m^{(i)}(p)$ pour $i = 0, 1, \ldots, k$.

5. Calculer $s'(p)$ en utilisant la relation : $s'(p) = \sum_{i=0}^{k} s'^{(i)}(p)N^{(i)}(p)M^{(i)}(p) \bmod m^{(i)}(p)$.

6. Calculer $s(p) = s'(p) + h_{N-1}x_{L-1}m(p)$.

Exemple 7:

Construire un algorithme de la convolution 2 × 3 en utilisant l'algorithme modifié de Winograd avec $m(p) = p(p-1)(p+1)$.

Soit $m^{(0)}(p) = p$, $m^{(1)}(p) = p-1$, $m^{(2)}(p) = p-1$.

Construire le tableau suivant en utilisant les relations :

$M^{(i)}(p) = m(p)/m^{(i)}(p)$ *et* $N^{(i)}(p)M^{(i)}(p) + n^{(i)}(p)m^{(i)}(p) = 1$.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

i	$m^{(i)}(p)$	$M^{(i)}(p)$	$n^{(i)}(p)$	$N^{(i)}(p)$
0	p	p^2-1	p	-1
1	$p-1$	p^2+p	$-\frac{1}{2}(p+2)$	$\frac{1}{2}$
2	$p+1$	p^2-p	$-\frac{1}{2}(p-2)$	$\frac{1}{2}$

Calculer les résidus à partir de :

$h(p)=h_0+h_1p, \quad x(p)=x_0+x_1p+x_2p^2$.

$h^{(0)}(p)=h_0, \qquad x^{(0)}(p)=x_0$.

$h^{(1)}(p)=h_0+h_1, \quad x^{(1)}(p)=x_0+x_1+x_2$.

$h^{(2)}(p)=h_0-h_1, \quad x^{(2)}(p)=x_0-x_1+x_2$.

$s'^{(0)}(p)=h_0x_0, \quad s'^{(1)}(p)=(h_0+h_1)(x_0+x_1+x_2).$

$s'^{(2)}(p)=(h_0-h_1)(x_0-x_1+x_2).$

Puisque le degré de $m^{(i)}(p)$ est égal à 1, $s'^{(i)}(p)$ est un polynôme de degré 0 (une constante). Par conséquent, nous avons :

$s(p)=s'(p)+h_1x_2m(p).$

$$s(p)=\left[-s'(0)(-p^2+1)+\frac{s'(1)}{2}(p^2+p)+\frac{s'(2)}{2}(p^2-p)+h_1x_2(p^3-p)\right].$$

$$s(p)=s'(0)+p(\frac{s'(1)}{2}-\frac{s'(2)}{2}-h_1x_2)+p^2(-s'^{(0)}+\frac{s'(1)}{2}+\frac{s'(2)}{2})+p^3(h_1x_2).$$

L'algorithme peut être exprimé sous forme matricielle :

$$\begin{bmatrix}s_0\\s_1\\s_2\\s_3\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&1&-1&-1\\-1&1&1&0\\0&0&0&1\end{bmatrix}\cdot\begin{bmatrix}s'^{(0)}\\\frac{s'^{(1)}}{2}\\\frac{s'^{(2)}}{2}\\h_1x_2\end{bmatrix}$$

Où $$\begin{bmatrix}s_0\\s_1\\s_2\\s_3\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&1&-1&-1\\-1&1&1&0\\0&0&0&1\end{bmatrix}\cdot\begin{bmatrix}h_0&0&0&0\\0&\frac{h_0+h_1}{2}&0&0\\0&0&\frac{h_0-h_1}{2}&0\\0&0&0&h_1\end{bmatrix}\cdot\begin{bmatrix}1&0&0\\1&1&1\\1&-1&1\\0&0&1\end{bmatrix}\cdot\begin{bmatrix}x_0\\x_1\\x_2\end{bmatrix}$$

– Conclusion : Cet algorithme exige 4 multiplications et 7 additions.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

II.6. La courte convolution itérative [1,4,46]

Les algorithmes efficaces de la convolution de longueur courte sont utilisés itérativement pour construire les longues convolutions. Cet algorithme n'atteint pas la complexité de multiplication minimale, mais assure un bon équilibre entre les complexités de multiplications et d'additions.

II.6.1. Description de l'algorithme de la convolution itérative [1,4,46]

1- Décomposer la longue convolution en plusieurs niveaux de courtes convolutions.
2- Construire les algorithmes de calcul rapide de la convolution pour des courtes convolutions.
3- Utiliser les algorithmes de courtes convolutions pour implémenter itérativement la longue convolution.

Remarque :

L'ordre des courtes convolutions dans la décomposition affecte la complexité de la longue convolution résultante.

Exemple 8 :

Construire un algorithme linéaire de la convolution 4×4 en utilisant la courte convolution 2 × 2.

Soit $h(p) = h_0 + h_1 p + h_2 p^2 + h_3 p^3$, $x(p) = x_0 + x_1 p + x_2 p_2 + x_3 p^3$ et $s(p) = h(p)x(p)$.

D'abord, nous devons décomposer la convolution *4 × 4* en une convolution *2 × 2*.

$$h'_0(p) = h_0 + h_1 p, \quad h'_1(p) = h_2 + h_3 p$$
$$x'_0(p) = x_0 + x_1 p, \quad x'_1(p) = x_2 + x_3 p$$

Nous avons alors:

$$h(p) = h'_0(p) + h'_1(p)p^2, \qquad ,h(p) = h(p,q) = h'_0(p) + h'_1(p)q$$
$$x(p) = x'_0(p) + x'_1(p)p^2, \qquad x(p) = x(p,q) = x'_0(p) + x'_1(p)q$$
$$\begin{aligned} s(p) &= h(p)x(p) = h(p,q)x(p,q) \\ &= [h'_0(p) + h'_1(p)q] \cdot [x'_0(p) + x'_1(p)q] \\ &= h'_0(p)x'_0(p) + [h'_0(p)x'_1(p) + h'_1(p)x'_0(p)]q + h'_1(p)x'_1(p)q^2 \\ &= s'_0(p) + s'_1(p)q + s'_2(p)q^2 = s(p,q) \end{aligned}$$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Par conséquent, la convolution 4×4 est décomposée en deux niveaux de convolution 2×2. Commençons à partir de la première convolution $s'_0(p) = h'_0(p).x'_0(p)$, nous avons :

$$h'_0(p)\cdot x'_0(p) \equiv h'_0 \cdot x'_0 = (h_0 + h_1 p)\cdot(x_0 + x_1 p)$$
$$= h_0x_0 + h_1x_1p^2 + p[(h_0 + h_1)\cdot(x_0 + x_1) - h_0x_0 - h_1x_1]$$

Nous avons l'expression suivante pour la troisième convolution

$$s'_2(p) = h'_1(p)\cdot x'_1(p) \equiv h'_1 \cdot x'_1 = (h_2 + h_3 p)\cdot(x_2 + x_3 p)$$
$$= h_2x_2 + h_3x_3p^2 + p[(h_2 + h_3)\cdot(x_2 + x_3) - h_2x_2 - h_3x_3]$$

Pour la seconde convolution, on obtient l'expression suivante :

$$s'_1(p) = h'_0(p)\cdot x'_1(p) + h'_1(p)\cdot x'_0(p) \equiv h'_0 \cdot x'_1 + h'_1 \cdot x'_0$$
$$= [(h'_0 + h'_1)\cdot(x'_0 + x'_1) - h'_0 \cdot x'_0 - h'_1 \cdot x'_1]$$

—: multiplication ▼: addition

Si nous réécrivons les trois convolutions comme expressions suivantes, alors nous pouvons obtenir le tableau suivant :

p^0	p^1	p^2	p^3	p^4	p^5	p^6
a_1	a_2	a_3		b_1	b_2	b_3
		c_1	c_2	c_3		
		$-b_1$	$-b_2$	$-b_3$		
		$-a_1$	$-a_2$	$-a_3$		

▲ ▲ Au total 8 additions à ce niveau.

Ceci exige 9 multiplications et 11 additions.

$$h'_0 x'_0 \equiv a_1 + pa_2 + p^2a_3$$
$$h'_1 x'_1 \equiv b_1 + pb_2 + p^2b_3$$
$$(h'_0 + h'_1)\cdot(x'_0 + x'_1) \equiv c_1 + pc_2 + p^2c_3$$

Par conséquent, le nombre total d'opérations utilisées dans cet algorithme de la convolution itérative 4 × 4 est de 9 multiplications et 19 additions.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

II.7. Conception d'algorithmes rapides de la convolution par inspection [46]

Lorsque les algorithmes de Cook-Toom ou Winograd ne peuvent pas générer un algorithme efficace, parfois une factorisation intelligente par inspection peut produire le meilleur algorithme.

Exemple 9 :

Construire un algorithme rapide de la convolution 3×3 par inspection. La convolution linéaire 3×3 peut être écrite comme suit (elle exige 9 multiplications et 4additions) :

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} h_0 x_0 \\ h_1 x_0 + h_0 x_1 \\ h_2 x_0 + h_1 x_1 + h_0 x_2 \\ h_2 x_1 + h_1 x_2 \\ h_2 x_2 \end{bmatrix}$$

Utiliser les identités suivantes:

$$s_1 = h_1 x_0 + h_0 \cdot x_1 = (h_0 + h_1) \cdot (x_0 + x_1) - h_0 x_0 - h_1 x_1$$
$$s_2 = h_2 x_0 + h_1 x_1 + h_0 x_2 = (h_0 + h_2)(x_0 + x_2) - h_0 x_0 + h_1 x_1 - h_2 x_2$$
$$s_3 = h_2 x_1 + h_1 x_2 = (h_1 + h_2)(x_1 + x_2) - h_1 x_1 - h_2 x_2$$

La convolution linéaire 3 × 3 peut être écrite comme suit :

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 0 & 0 \\ -1 & 1 & -1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} h_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & h_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_0 + h_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 + h_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & h_1 + h_2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

Cet algorithme, qui ne peut pas être obtenu en faisant appel aux algorithmes Cook-Toom ou Winograd, exige 6 multiplications et 10 additions.

Pour l'algorithme de la courte convolution itérative on définit l'opérateur produit tensoriel et sa propriété.

II.8. Le produit tenseur (produit de Kronecker) [1,2,4,29,41,42,44]

II.8.1. Définition

Le produit tensoriel est un opérateur binaire entre deux matrices qui fournit un mécanisme pour combiner deux formes de matrices à une plus grande matrice.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

L'opération matricielle est le coeur de la plupart des algorithmes de traitement numérique du signal (DSP), et le produit tensoriel est un outil extrêmement puissant pour modéliser et implémenter des algorithmes efficaces.

Une définition mathématique précise du produit tensoriel est établie comme:

Le produit tensoriel ou produit de Kronecker de la matrice $A = [a_{i,j}]_{m1\times n1}$ et la matrice $B = [b_{i,j}]_{m2\times n2}$ est la matrice $C = [c_{i,j}]_{[m1.m2 \times\ n1.n2]}$ de forme :

$$C = A \otimes B = \begin{bmatrix} a_{0,0}B \ldots a_{0,n}B \\ \ldots \quad \ldots \quad \ldots \\ a_{m,0}B \ldots a_{m,n}B \end{bmatrix} \qquad \text{(II.32)}$$

Nous énumérons certaines des propriétés les plus importantes du produit tensoriel **[19]**

1. $A \otimes I_1 = I_1 \otimes A = A$ avec I_{mn}: La matrice identité.
2. $A \otimes (B \otimes C) = (A \otimes B) \otimes C$
3. $(A + B) \otimes C = (A \otimes C)\,(B \otimes C)$
4. $A \otimes (B + C) = (A \otimes B)\,(A \otimes C)$
5. $(A \otimes B)(C \otimes D) = AC \otimes BD$
6. $I_m \otimes I_n = I_{mn}$
7. $(A \otimes B)^{\mathrm{T}} = A^{\mathrm{T}} \otimes B^{\mathrm{T}}$
8. $A^{-1} \otimes B^{-1} = (A \otimes B)^{-1}$
9. $A_{m1\times n1} \otimes B_{m2\times n2} = (A \otimes I_{m2})(I_{n1} \otimes B) = (I_{m1} \otimes A)(B \otimes I_{n2})$
10. $A \otimes B \neq B \otimes A$ (Ce produit n'est pas généralement commutatif).

(II.33)

Les algorithmes de calcul rapide de la transformée de Fourier Discrète (TFD) son basés sur une factorisation de la matrice de cette transformée. Il est nécessaire de faire appel à un outil mathématique bien adapté, le produit de Kronecker des matrices.

En combinant ce produit avec le produit au sens habituel, il est possible de décomposer simplement toute matrice de TFD [4].

II.8.2. Implémentation de la matrice tenseur [1,4, 41, 42]

D'après l'expression (II.32), on a:

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$A_{n1\times n2}$: la matrice d'identité de l'ordre *s*

$B_{m1\times m2}$: la matrice carrée de dimension *r*.

La matrice résultante $C_n = (A \otimes B)$ est diagonale de dimension $n = rs$, sa structure est donnée par:

$$C_n = (I_S \otimes B_r) = \begin{bmatrix} B_r & 0 & 0 & 0 & 0 \\ 0 & B_r & 0 & 0 & 0 \\ 0 & 0 & B_r & 0 & 0 \\ 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & 0 & B_r \end{bmatrix} \qquad \text{(II.34)}$$

Supposons que C_n fait partie d'un certain algorithme de traitement du signal et qu'il est nécessaire de calculer le produit $Y_n = (C_n).X_n$, avec X_n le vecteur d'entrée.

Le produit tenseur est utilisé fréquemment pour déduire des algorithmes rapides pour le traitement numérique du signal. Il vaudrait mieux exploiter les symétries de $(I_s \otimes B_r)$ en partageant le vecteur X_n en *s* sous vecteur x_i de longueur *r* et en effectuant le produit B_r (x_i) de la matrice B_r sur chacun de ces derniers.

Exemple 10 :

Soit I_3 la matrice identité d'ordre 3 et *B* une matrice de dimension 2 × 2, c.à.d

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, B_2 = \begin{pmatrix} b_{0.0} & b_{0.1} \\ b_{1.0} & b_{1.1} \end{pmatrix} \qquad \text{soit l'algorithme :} \qquad Y = (C_6)X_6 = (I_3 \otimes B_2)X_6$$

qui donne:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} b_{0.0} & b_{0.1} & & & & \\ b_{1.0} & b_{1.1} & & & & \\ & & b_{0.0} & b_{0.1} & & \\ & & b_{1.0} & b_{1.1} & & \\ & & & & b_{0.0} & b_{0.1} \\ & & & & b_{1.0} & b_{1.1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \qquad \text{(II.35)}$$

Ce produit de matrices peut être réalisé en partageant le vecteur d'entrée X_6 en 3 sous vecteurs de longueur 2 et en effectuant le produit B_2X_2 de la matrice B_2 sur chacune de ces vecteurs:

$$\begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} b_{0.0} & b_{0.1} \\ b_{1.0} & b_{1.1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \begin{pmatrix} y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} b_{0.0} & b_{0.1} \\ b_{1.0} & b_{1.1} \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \text{ et } \begin{pmatrix} y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} b_{0.0} & b_{0.1} \\ b_{1.0} & b_{1.1} \end{pmatrix} \begin{pmatrix} x_4 \\ x_5 \end{pmatrix}.$$

La structure de $(I_s \otimes B_r)$ donne l'idée de faire une boucle sur une routine de calcul du produit de la matrice B_r, avec chacun des vecteurs 2[**41**].

for (i=0 to s-1)

$B_r(x+ ir);$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

La deuxième forme: $C_n = (A_r \otimes I_s)$, est donne par

$$C_n = \begin{pmatrix} a_{0.0}I_s & a_{0.1}I_s & \dots & a_{0.r-1}I_s \\ a_{0.0}I_s & a_{0.1}I_s & \dots & a_{0.r-1}I_s \\ \dots & \dots & \dots & \dots \\ a_{0.0}I_s & a_{0.1}I_s & \dots & a_{0.r-1}I_s \end{pmatrix} \qquad \text{(II.36)}$$

Le calcul de $Y_n = (A_r \otimes I_s)X_n$ se fait en partageant le vecteur X_n par r sous vecteurs de longueur s.

Exemple 11 :

On à $Y_6 = (C_6)X_6 = (A_2 \otimes I_3)X_6$ donne:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} a_{0.0} & a_{0.1} & & & & \\ & a_{0.0} & a_{0.1} & & & \\ & & a_{0.0} & a_{0.1} & & \\ a_{1.0} & a_{1.1} & & & & \\ & a_{10} & a_{1.1} & & & \\ & & a_{1.0} & a_{1.1} & & \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}$$

Le produit représenté avec 2 sous vecteurs comme:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix} = (a_{0.0}) \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} + (a_{0.1}) \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} \quad et \quad \begin{pmatrix} y_3 \\ y_4 \\ y_5 \end{pmatrix} = (a_{1.0}) \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} + (a_{1.1}) \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix}$$

Pour comprendre les relations entre les vecteurs et les étages parallèles, il nécessaires de faire appel à la notion de pas de permutation.

II.9. Pas de permutation [29,41,44,47]

II.9.1. Définition

A_s et B_r sont deux matrices carrées de dimension s et r, respectivement, et $n = r.s$ ou

$(A_s \otimes B_r) = P_{n.s}\,(B_r \otimes A_s)P_{n.r}$.

On peut prendre quelques exemples de pas de permutation et leur effet sur les données d'entrée:

$$y = (P_{4,2})\,x = \begin{pmatrix} x_0 \\ x_2 \\ x_1 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1000 \\ 0010 \\ 0100 \\ 0001 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$$y = (P_{6,2}) = \begin{pmatrix} x_0 \\ x_2 \\ x_4 \\ x_1 \\ x_3 \\ x_5 \end{pmatrix} = \begin{pmatrix} 100000 \\ 001000 \\ 000010 \\ 010000 \\ 000100 \\ 000001 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}, \quad y = (P_{8,4}) = \begin{pmatrix} x_0 \\ x_4 \\ x_1 \\ x_5 \\ x_2 \\ x_6 \\ x_3 \\ x_7 \end{pmatrix} = \begin{pmatrix} 10000000 \\ 00001000 \\ 01000000 \\ 00000100 \\ 00000010 \\ 00010000 \\ 00000001 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}.$$

$(I_s \otimes A_r)$ peut être exprimé comme: $(I_s \otimes A_r) = P_{n.s}\,(A_r \otimes I_s) P_{n.r}$.

L'introduction du pas de permutation donne:

$$Y_n = (C_n)\, x_n = [P_{n.s}\,(I_r \otimes A_s)\, P_{n.r}]\,(I_s \otimes B_r)\, x_n \qquad \text{(II.37)}$$

$$[\mathrm{P_{n.s}}\ (\mathrm{I_r} \otimes \mathrm{A_s}) \mathrm{P_{n.r}}] \Leftrightarrow \begin{bmatrix} (P_{n,s}) \begin{pmatrix} As \\ As \\ \dots \\ As \end{pmatrix} (P_{n,r}) \end{bmatrix}, \qquad (I_s \otimes B_r) \Leftrightarrow \begin{bmatrix} B_r & & & & \\ & B_r & & & \\ & & B_r & & \\ & & & \dots & \\ & & & & B_r \end{bmatrix}$$

On peut le programmer comme suit:

for (*i = 0 to s-1*)

B_r (*input, output, i*);

for (*i=0 to r-1*)

A_s (*input, output, i, r, s*)

La performance de l'algorithme est spécifiée par le nombre d'opérations. Le calcul direct de C_n exige n^2 multiplications et $n(n-1)$ additions avec le nombre total d'opérations $2n^2-n$. L'utilisation d'un algorithme qui exige sr^2+rs^2 multiplications et $sr(r-1) + rs(s-1)$ additions avec un nombre total d'opérations $s(2r^2- r) +r\,(2s^2-s)$, améliore la performance.

Pour $r = s = 10$, l'implémentation directe exige 19990 opérations, tandis que l'algorithme (II.37) exigerait 3800 opérations.

Le produit tensoriel est un outil puissant pour la modélisation et l'implémentation des algorithmes rapides pour le traitement numérique du signal.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

II.10. Implémentation de convolutions 4 x 4 selon l'approche de Parhi [1,4]

D'après la technique de Parhi on peut implémenter la courte convolution itérative 4 x 4 par les étapes suivantes :

Nous considérons l'architecture d'un filtre RIF 4-parallèles à 4-taps.

La convolution 4 x 4 peuvent être décomposée en deux courtes convolutions 2 x 2 selon l'équation :

$Y_4 = (P_2^T \otimes P_2^T) H_4 (Q_2^T \otimes Q_2^T) A_{4-22}^T X_4$ (Exige 22 additions, 9 multiplications).

$H_4 = diag\left(\left[H_0, H_0 - H_1, H_1, H_0 - H_2, H_0 - H_1 - H_2 + H_3, H_1 - H_3, H_2, H_2 - H_3, H_3\right]\right)$.

$X_4 = \left[X_3 \;\; X_2 \;\; X_1 \;\; X_0 \;\; Z^{-4}X_3 \;\; Z^{-4}X_2 \;\; Z^{-4}X_1\right]^T$.

Avec $H_i = \{h_{Lk+i}\}$, $L = 4$, $i = 0,1,2,\ldots,L\text{-}1$, $k = 0,1,2,\ldots..$

$$A_{4-22}^T = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix},$$

$$Q_2^T \otimes Q_2^T = \begin{bmatrix} 1&1&0&1&1&0&0&0&0 \\ 0&-1&0&0&-1&0&0&0&0 \\ 0&1&1&0&1&1&0&0&0 \\ 0&0&0&-1&-1&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&-1&-1&0&0&0 \\ 0&0&0&1&1&0&1&1&0 \\ 0&0&0&0&-1&0&0&-1&0 \\ 0&0&0&0&1&1&0&1&1 \end{bmatrix}, \; P_2^T \otimes P_2^T = \begin{bmatrix} 1&1&0&1&1&0&0&0&0 \\ 0&-1&1&0&-1&1&0&0&0 \\ 0&0&0&-1&-1&0&1&1&0 \\ 0&0&0&0&1&-1&0&-1&1 \end{bmatrix}.$$

L'implémentation directe du produit tensoriel pour $Y_9 = (Q_2^T \otimes Q_2^T).X_9$ exige 12 additions.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \end{bmatrix} = \begin{bmatrix} 1&1&0&1&1&0&0&0&0 \\ 0&-1&0&0&-1&0&0&0&0 \\ 0&1&1&0&1&1&0&0&0 \\ 0&0&0&-1&-1&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&-1&-1&0&0&0 \\ 0&0&0&1&1&0&1&1&0 \\ 0&0&0&0&-1&0&0&-1&0 \\ 0&0&0&0&1&1&0&1&1 \end{bmatrix} . \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}.$$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$y_1 = x_0 + x_1 + (x_3 + x_4)$ 3 additions.

$y_2 = -x_1 + x_4$ 1 addition.

$y_3 = x_1 + x_2 + (x_4 + x_5)$ 3 additions.

$y_4 = -(x_3 + x_4)$ calculée une seule fois.

$y_5 = x_4$.

$y_6 = -(x_4 + x_5)$ calculée une seule fois.

$y_7 = x_3 + x_4 + x_6 + x_7$ 2 additions.

$y_8 = -x_4 - x_7$ 1 addition.

$y_9 = (x_4 + x_5) + x_7 + x_8$ 2 additions.

Nombre total d'additions : 12 additions (implémentation directe du produit tensoriel).

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \end{bmatrix} = diag(H_0, H_0 - H_1, H_1, H_0 - H_2, H_0 - H_1 - H_2 + H_3, H_1 - H_3, H_2, H_2 - H_3, H_3]. \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \end{bmatrix}.$$

$s_1 = H_0\, y_1$.

$s_2 = (H_0 - H_1)\, y_2$.

$s_3 = H_1\, y_3$.

$s_4 = (H_0 - H_2)\, y_4$.

$s_5 = (H_0 - H_1 - H_2 + H_3)\, y_5$.

$s_6 = (H_1 - H_3)\, y_6$.

$s_7 = H_2\, y_7$.

$s_8 = H_0\, y_8$.

$s_9 = H_0\, y_9$.

} Neuf opération de filtrage.

Neuf multiplications sont nécessaires pour l'implémentation directe du produit tensoriel.

L'implémentation directe du produit tensoriel pour $Z_9 = (P_2^T \otimes P_2^T).S_9$ exige 10 additions.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \\ z_9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \end{bmatrix}.$$

$z_1 = s_1 + s_2 + (s_4 + s_5)$, 3 additions.

$z_2 = -s_2 + s_3 - s_5 + s_6$, 3 additions.

$z_3 = -(s_4 + s_5) + s_7 + s_8$, 2 additions.

$z_4 = -(-s_5 + s_6) - s_8 + s_9$, 2 additions.

Nombre total d'additions = 10 additions pour l'implémentation directe du produit tensoriel.

Nombre total d'additions et multiplications : 22 additions et 9 multiplications (qui est égal au nombre de sous filtres) pour l'implémentation direct du produit tensoriel.

La figure.II.6 représente l'architecture d'un filtre RIF 4-parallèle à 4-taps conçu selon la technique de Parhi **[1,4].**

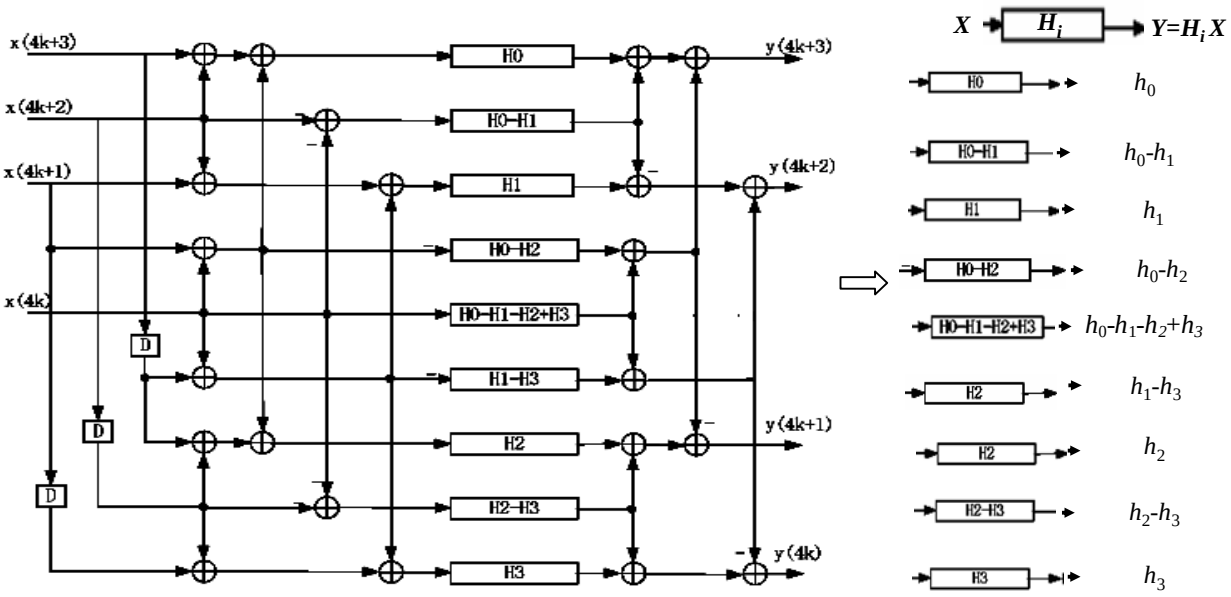


Fig.II.6. Implémentation du filtre RIF 4- Parallèles à 4-taps selon la structure de Parhi.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

L'implémentation de sous filtres pour 16-taps est représenté sur la figure.II.7 [**1,34**].

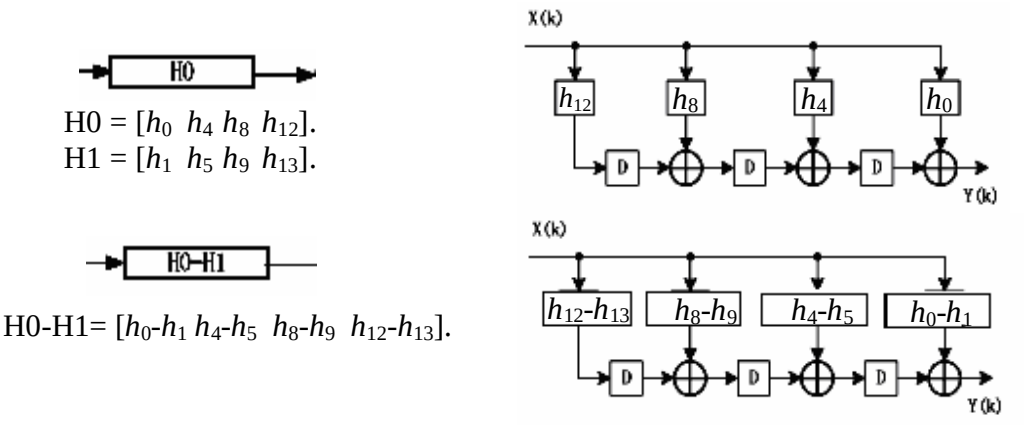


Fig.II.7. Implémentation de sous filtre H0 et H0-H1 (L = 4, N = 16)

II.11. Mise en application de l'algorithme de la courte convolution itérative ''Technique de Parhi'' [1,4]

On prend comme exemple $L = 4$.

La convolution 4 × 4 peut être décomposée en deux courtes convolutions *2 × 2* comme suit:

$$S_7 = A_{4\_22}\,(Q_2 \otimes Q_2).H_{4\_22}.\,(P_2 \otimes P_2).X_6 \qquad \text{(II.38)}$$

Cet algorithme peut être transposé pour obtenir une structure rapide d'un filtre RIF 4-Parallèle à 4-tape. On peut représenter l'algorithme de filtrage par l'expression suivante:

$$Y_4 = (P_2^T \otimes P_2^T) H_4 (Q_2^T \otimes Q_2^T) A_{4-22}^T X_4 \qquad \text{(II.39)}$$

Avec $A_{4-22}^T = \begin{bmatrix} 1000000 \\ 0100000 \\ 0010000 \\ 0010000 \\ 0001000 \\ 0000100 \\ 0000100 \\ 0000010 \\ 0000001 \end{bmatrix}$, $Q_2^T \otimes Q_2^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$.

$H_4 = diag\left(\left[H_0, H_0 - H_1, H_1, H_0 - H_2, H_0 - H_1 - H_2 + H_3, H_1 - H_3, H_2, H_2 - H_3, H_3\right]\right)$.

$X_4 = \left[X_3 \;\; X_2 \;\; X_1 \;\; X_0 \;\; Z^{-4}X_3 \;\; Z^{-4}X_2 \;\; Z^{-4}X_1\right]^T$.

Z : l'opérateur de la transformation en *Z.*

H_i *:* représentent les sous filtres (contenant les coefficients h_{LK+i}).

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

On peut représenter l'algorithme dans le domaine discret par la forme matricielle suivante :

$$\begin{bmatrix} y_{Lk+3} \\ y_{Lk+2} \\ y_{Lk+1} \\ y_{Lk} \end{bmatrix} = (P_2^T \otimes P_2^T)[(P_2 \otimes P_2).[h_{Lk}\ h_{Lk+1}\ h_{Lk+2}\ h_{Lk+3}]^T](Q_2^T \otimes Q_2^T)A_{4-22}^T \begin{bmatrix} x_{Lk+3} \\ x_{Lk+2} \\ x_{Lk+1} \\ x_{Lk} \\ x_{Lk-1} \\ x_{Lk-2} \\ x_{Lk-3} \end{bmatrix} \quad \text{(II.40)}$$

Pour L = 4, k = 0, 1,2,… (k =0,…, N -1, N : représente le nombre de taps).

Pour k = 0, $$\begin{bmatrix} y_3 \\ y_2 \\ y_1 \\ y_0 \end{bmatrix} = (P_2^T \otimes P_2^T)[(P_2 \otimes P_2).[h_0\ h_1\ h_2\ h_3]^T](Q_2^T \otimes Q_2^T)A_{4-22}^T \begin{bmatrix} x_3 \\ x_2 \\ x_1 \\ x_0 \\ x_{-1} \\ x_{-2} \\ x_{-3} \end{bmatrix} \quad \text{(II.41)}$$

Qui donne : $$\begin{bmatrix} y_3 \\ y_2 \\ y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} h_0\ h_1\ h_2\ h_3\ 0\ 0\ 0 \\ 0\ h_0\ h_1\ h_2\ h_3\ 0\ 0 \\ 0\ 0\ h_0\ h_1\ h_2\ h_3\ 0 \\ 0\ 0\ 0\ h_0\ h_1\ h_2\ h_3 \end{bmatrix} \begin{bmatrix} x_3 \\ x_2 \\ x_1 \\ x_0 \\ x_{-1} \\ x_{-2} \\ x_{-3} \end{bmatrix}$$

Avec *:* x_{-1}, x_{-2}, x_{-3} séquence d'entrée avec retard.

Dans l'exemple précédent, on voie que l'algorithme donne une fonction de filtrage qui représente la fonction de transfert (dans le cas de N = 4-taps).

II.12. Simulation de la structure de Parhi issue de la (ISC) pour différentes méthodes de synthèse des RIF

La simulation se fait selon les étapes suivantes:

1) Entrée des spécifications du filtre à synthétiser.

2) Synthèse par différentes méthodes (fenêtre, Remez, moindre carré...)

3) Extraction des coefficients du filtre RIF.

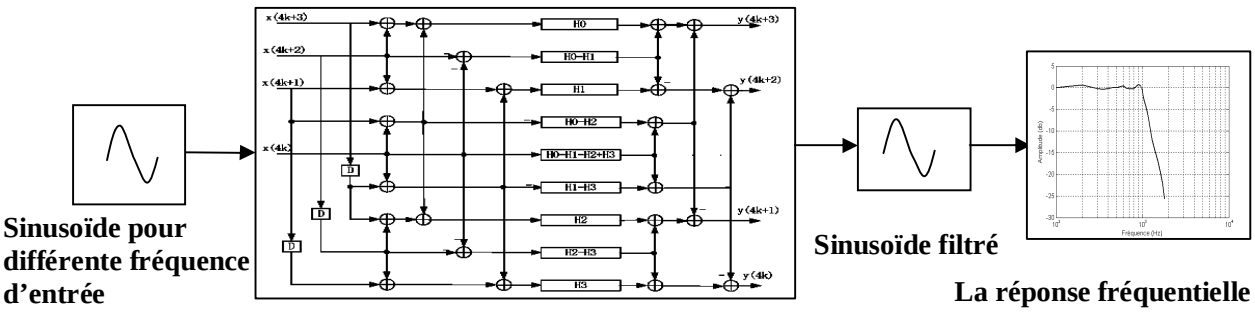
4) Test de l'algorithme de filtrage (ISC) avec les coefficients obtenus (fonction de filtrage).

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

5) Implémentation de l'algorithme de filtrage sur la structure VLSI selon la technique de Parhi (exemple L = 4).

6) Filtrage d'une sinusoïde pour différentes fréquences d'entrée.

7) Détermination de la réponse fréquentielle du filtre simulé.



Structure de Parhi pour L = 4.

Fig.II.8. Simulation de la structure de Parhi issue de la (ISC)

a) Cas d'un filtre passe-bas

Considérons un filtre RIF passe bas de fréquences caractéristiques :

f_1 = 1000Hz : fréquence de coupure.

f_2 = 1300Hz : limite de la bande atténué.

f_e = 4000Hz : fréquence d'échantillonnage.

En utilisant les coefficients de la réponse impulsionnelle du filtre RIF passe-bas obtenues par la méthode de Remez (tableau.II.1), nous avons implémenté sur Matlab 7.1 (SIMULINK) la structure de Parhi pour $L = 4$ et $N = 24$.

La figure II.10 illustre les résultats de simulation en représentant la réponse du filtre à la sinusoïde d'entrée pour les fréquences : f = 100, 600, 900, 990, 1000, 1060, 1100, 1200, 1250, 1300, 1400, 1800 Hz.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Méthode de synthèse	Les coefficient h_i
Remez	$h_0 = -0.0130, h_1 = -0.03278, h_2 = -0.0173,$ $h_3 = 0.0241, h_4 = 0.0113, h_5 = -0.0382,$ $h_6 = -0.0026, h_7 = 0.0627, h_8 = -0.0207,$ $h_9 = -0.1132, h_{10} = 0.1101, h_{11} = 0.4828,$ $h_{12} = 0.4828, h_{13} = 0.1101, h_{14} = -0.1131,$ $h_{15} = -0.0207, h_{16} = 0.06278, h_{17} = -0.0026,$ $h_{18} = -0.0382, h_{19} = 0.0113, h_{20} = 0.0241,$ $h_{21} = -0.0173, h_{22} = -0.0327, h_{23} = -0.0130,$

Tableau II.1. Réponse impulsionnelle d'un filtre passe-bas RIF par la méthode de Remez pour N = 24-taps.

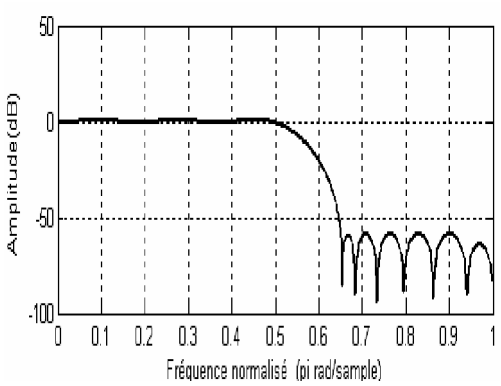
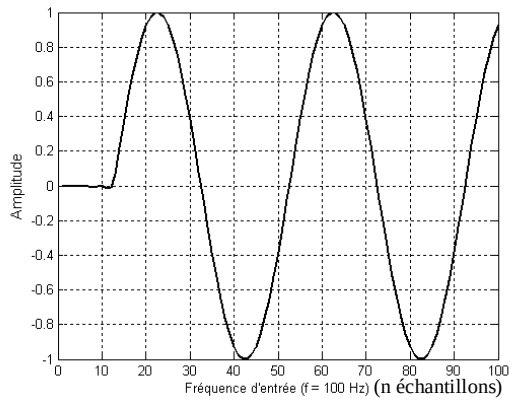
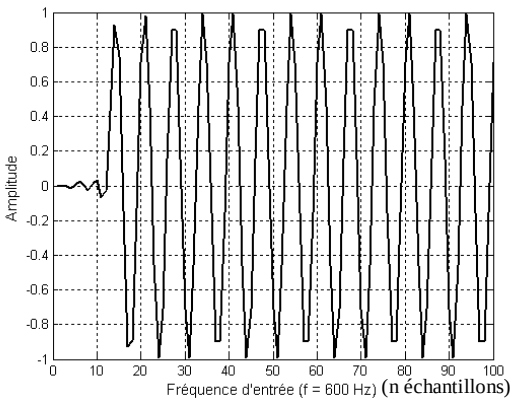


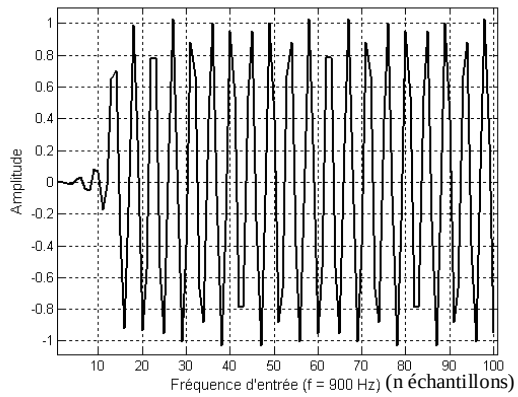
Fig.II.9. Réponse fréquentielle du filtre RIF synthétisé par la méthode de Remez



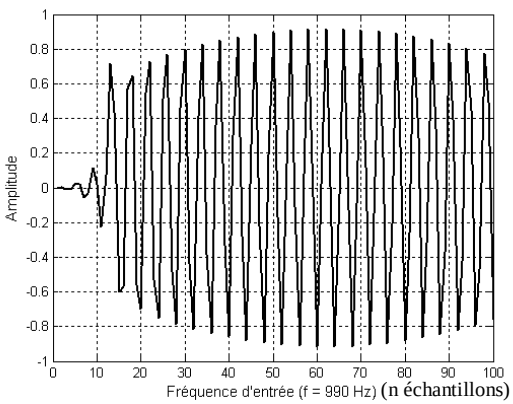
(a)



(b)

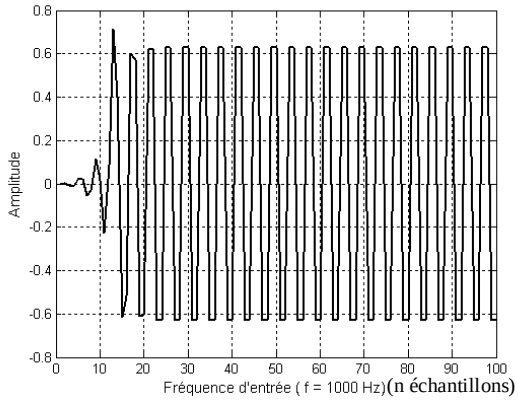


(c)

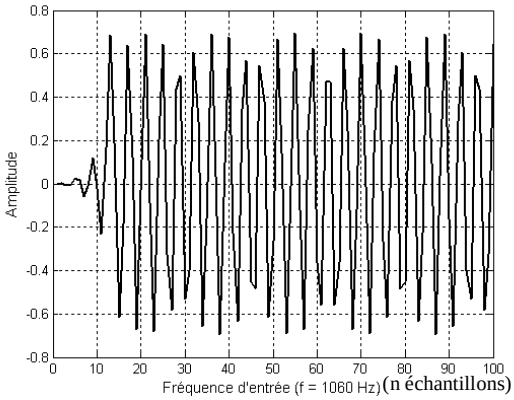


(d)

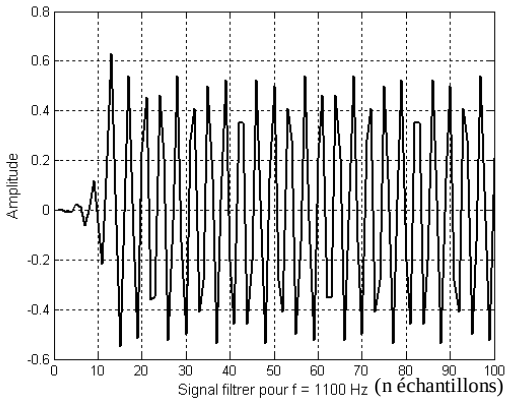
This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



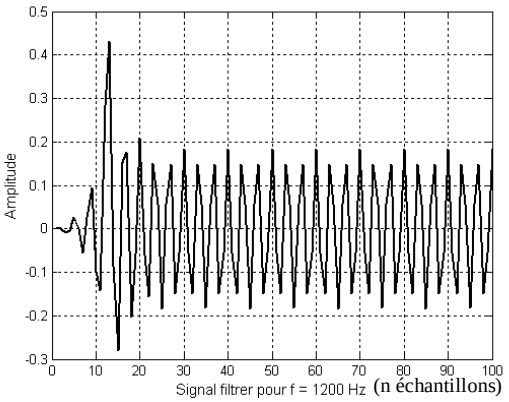
(e)



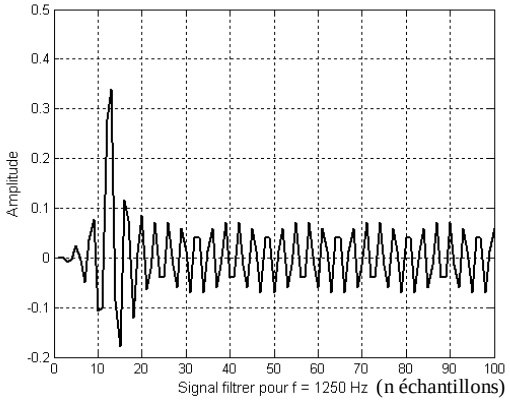
(f)



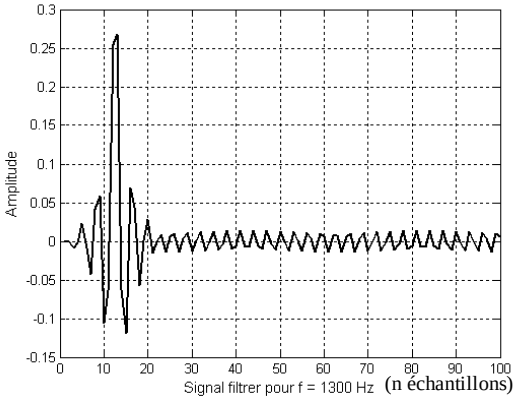
(g)



(h)

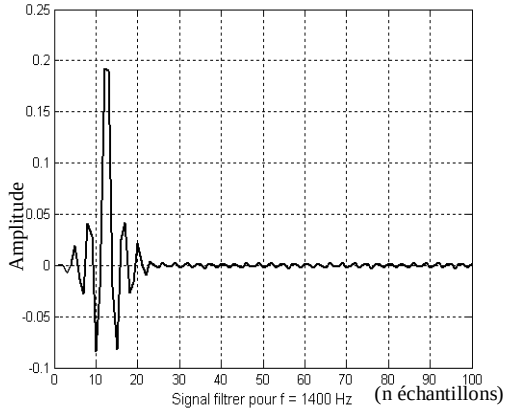


(i)

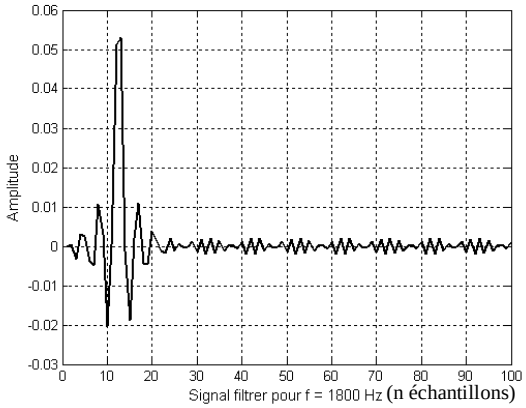


(j)

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



(k)



(l)

Fig.II.10. Filtrage du signal sinusoïdal pour un filtre RIF passe-bas implémenté sur la structure de Parhi issue de la ISC avec L = 4 et N = 24-taps.

Nous remarquons que l'algorithme de la courte convolution simule bien la fonction de filtrage passe-bas et que la fréquence de coupure est de 1060 Hz conformément aux spécifications du filtre synthétisé.

La synthèse du filtre RIF par la méthode de Remez donne la réponse fréquentielle en amplitude représentée sur la figure.II.11et la réponse impulsionnelle donnée sur le tableau II.2.

Méthode de synthèse	Les coefficient h_i
Remez	h_0 =-0.0130, h_1 =-0.0327, h_2 = -0.0173, h_3 = 0.0241, h_4 = 0.0113, h_5 = -0.0382, h_6 = -0.0026, h_7 = 0.0627, h_8= -0.0207, h_9 = -0.1131, h_{10} = 0.1101, h_{11} = 0.4828, h_{12} =0.4828, h_{13}= 0.1101, h_{14} = -0.1131, h_{15} =-0.0207, h_{16}= 0.0627, h_{17} = -0.0026, h_{18} =-0.0382, h_{19} = 0.0113, h_{20} = 0.0241, h_{21} =-0.0173,h_{22} = -0.0327, h_{23} = -0.0130.

Tableau II.2. Réponse impulsionnelle d'un filtre passe-bas RIF par la méthode de Remez pour N = 24-taps.

A partir de l'amplitude du signale filtré, en représente la réponse fréquentielle de filtre RIF 4-parallèle à 24-taps (figure.II.11)

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

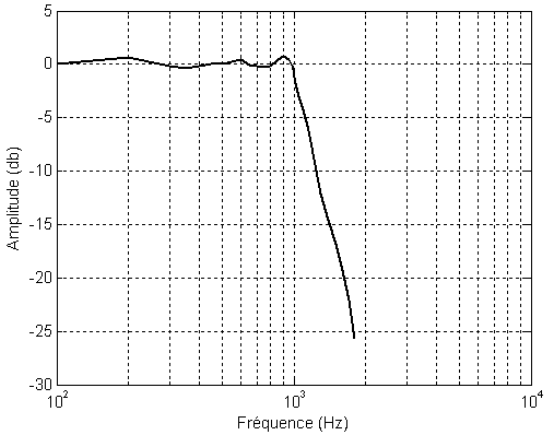


Fig.II.11. Réponse fréquentielle du filtre RIF implémenté sur la structure de Parhi pour L = 4, N = 24-taps (Méthode de Remez, $f_{coupure}$ = 1050 Hz).

La synthèse du filtre RIF par la méthode des moindres carrées donne la réponse impulsionnelle sur le tableau.II.3. et la réponse fréquentielle implémenté sur donnée de la figure.II.12 et

Méthode de synthèse	Les coefficient h_i
Moindres carrées	$h_0 = 0.0012$, $h_1 = -0.0078$, $h_2 = -0.0087$, $h_3 = 0.0154$, $h_4 = 0.0117$, $h_5 = -0.0314$, $h_6 = -0.0079$, $h_7 = 0.05918$, $h_8 = -0.0129$, $h_9 = -0.1139$, $h_{10} = 0.1024$, $h_{11} = 0.4879$, $h_{12} = 0.48793$, $h_{13} = 0.10242$, $h_{14} = -0.1139$, $h_{15} = -0.0129$, $h_{16} = 0.0591$, $h_{17} = -0.0079$, $h_{18} = -0.0314$, $h_{19} = 0.0117$, $h_{20} = 0.0154$, $h_{21} = -0.0087$, $h_{22} = -0.0078$, $h_{23} = 0.0012$.

Tableau II.3. Réponse impulsionnelle d'un filtre passe bas RIF par la méthode des moindres carrées pour N = 24-taps.

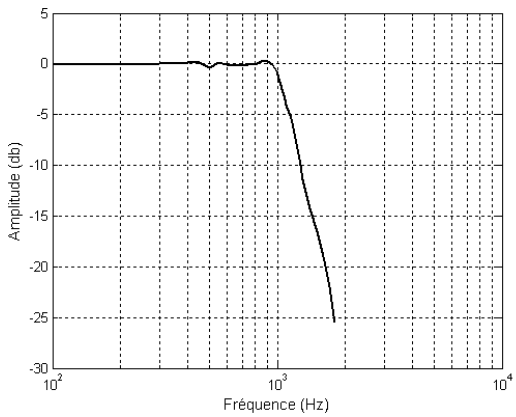


Fig.II.12. Réponse fréquentielle du filtre RIF 4-parallèle à 24-taps implémenté sur la structure de Parhi (moindre carrées, $f_{coupure}$ = 1062 Hz)

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Conclusion

Dans ce chapitre, nous avons présenté la technique de la courte convolution itérative et son application pour le développement d'algorithmes rapides réalisant la fonction filtrage RIF. Nous avons également simuler ces algorithmes de filtrage rapide pour examiner les performances de la technique ISC. Dans le chapitre suivant, nous allons utiliser les structures systoliques pour l'implémentation VLSI des algorithmes rapides de filtrage RIF obtenus par la méthode ISC.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Chapitre III
Implémentation des filtres RIF sur les réseaux systoliques

Introduction

Dans ce chapitre, nous allons appliquer les réseaux systoliques pour développer des architectures rapides dédiées à l'implémentation des filtres RIF représentés par la courte convolution itérative.

III.1. Les réseaux systoliques

Un réseau systolique est un système composé d'un ensemble de cellules processeurs interconnectées, chacune étant capable d'effectuer une certaine opération élémentaire. Ces processeurs élémentaires calculent et font transmettre d'une manière rythmique les données numériques à travers tout le réseau. Les structures systoliques présentent une simplicité et une régularité de contrôle et de communication par rapport à d'autres structures complexes en termes de conception et d'implémentation. Dans un système systolique, l'information s'écoule d'une manière pipeline entre les différentes cellules du réseau et la communication avec le monde extérieur s'établit simplement au niveau des cellules situées à la frontière du réseau systolique. Seules ces dernières cellules peuvent être donc utilisées pour les entrées/sorties du système à réaliser. Kung **[51,52]** a démontré que des cellules (PE) effectuant l'opération de base du produit scalaire de deux vecteurs ($y \longleftarrow y + A \times B$) peuvent être localement connectées d'une façon linéaire afin de réaliser l'opération de filtrage RIF. De plus, il est établi que des réseaux systoliques bidimensionnels, formés de la même catégorie de cellules PE, peuvent être construits afin d'exécuter, avec une grande efficacité, toute l'arithmétique des matrices.

Le principe de base d'une architecture systolique est illustré sur la figure .III.1 en remplaçant un simple processeur élémentaire (PE) par un réseau de PEs, ou cellules selon la terminologie couramment utilisée, ainsi un débit en données plus élevées peut être obtenu sans augmentation de la capacité mémoire. Le rôle de la mémoire dans ce diagramme est similaire à celui du cœur humain, elle pompe les données (au lieu du sang) à travers tout le réseau de cellules. Selon cette approche, chaque ensemble de données qui sort de la mémoire sera efficacement utilisé par chaque cellule rencontrée lors de son passage à la suite de son pompage d'une cellule à une autre le long du réseau. Ce principe de fonctionnement est valable pour une large classe d'algorithmes

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

présentant un nombre multiple d'opération à effectuer sur une séquence de données d'une manière répétitive **[51,52,53,54]**.

Bien que la transmission de données soit synchrone au niveau algorithmique **[51]**, une machine implémentant un algorithme systolique peut fonctionner soit d'une manière synchrone soit d'une manière asynchrone. La conception synchrone est souvent préférée pour des raisons de simplicité. Le fait d'être capable d'utiliser chaque ensemble de données d'entrée un certain nombre de fois est justement un des avantages multiples de l'approche systolique. Les réseaux systoliques sont devenus très célèbre pour leurs propriétés distingués d'interconnexion locale, et de multitraitement hautement pipeline et synchrone. D'autres avantages, comme l'extension modulaire, les flots de données et de contrôle simples et réguliers, l'utilisation de cellules simples et uniformes, l'élimination d'émission globale, et le temps de réponse rapide, sont aussi les caractéristiques d'un système systolique.

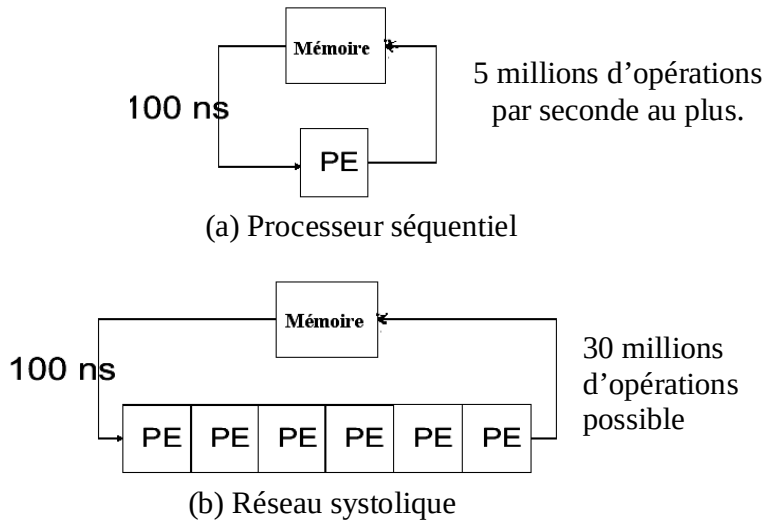


Fig.III.1. Principe de base d'un système systolique.

Toutefois, il subsiste encore un certain nombre de problèmes sans solution liés à l'usage des réseaux systoliques. En premier lieu, les systèmes systoliques demandent une synchronisation globale du réseau (c'est-à-dire une distribution globale du signal horloge). Ce qui peut donner naissance à des décalages d'horloge indésirables dans les implémentations de systèmes VLSI d'ordre élevé. En plus, les réseaux systoliques (parfaits) tendent à allouer les mêmes unités de temps à toutes les opérations de traitement. A titre d'exemple, dans les réseaux systoliques dédiés à la convolution, un transfert de données local s'effectue pendant la même période de temps qu'une opération de multiplication/accumulation (c'est-à-dire une unité de temps complète). Ce

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

qui résulte certainement en une perte de temps inutile. L'autre problème est surtout lié à la description d'un langage parallèle et la programmation des flots de données complexes.
Une liste de certains problèmes pouvant être résolus par l'approche systolique est donnée comme suit **[19,51, 55,56]**:

1) Traitement du signal et de l'image.
- Le filtrage RIF et RII et la convolution unidimensionnelle.
- La convolution et la corrélation bidimensionnelle.
- La transformée de fourrier discrète.
- L'interpolation.
- Le filtrage médiane unidimensionnel.

2) Arithmétique des matrices.
- Multiplication d'une matrice par un vecteur.
- Produit de deux matrices.
- Triangularisation d'une matrice (solution de systèmes linéaires, inversion matricielle).
- Solution de systèmes linéaires triangulaires.
- Solution de systèmes Toeplitz linéaires.
- Calculs des moindres carrées.
- Décomposition sur les valeurs singulières (méthode SVD).
- Problèmes des valeurs propres.

III.2. Principe des structures systoliques

Notre objectif consiste à obtenir des architectures systoliques simples et rapides dédiées aux filtres RIF en adoptant la représentation matricielle, il est donc indispensable d'étudier d'abord l'implémentation du produit de deux matrices sur un réseau systolique **[19,20,53,57,58]**.

Plusieurs algorithmes réalisant une telle implémentation ont été proposés par Kung **[34]**. Nous utiliserons dans ce travail l'algorithme proposé par Kung qui exige moins de temps de calcul.

Pour illustrer le principe de cet algorithme, nous considérons l'exemple du produit de deux matrices carrées (2 x 2). Soit A x $B = C$ la matrice produit.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, \quad C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}.$$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

La figure.III.2.a décrit une version systolique de l'algorithme et le flot de données dans cette structure. Le calcul des éléments de la matrice *C* est décomposé en taches élémentaires effectuées par des cellules de base dont le principe de fonctionnement est donnée à la figure.III.2.b Chaque cellule possède trois registres : deux destinés à stocker les entrées a_{ij} et b_{ij} et le troisième est destiné à stocker le résultat c_{ij}. Le contenu initial des registres est zéro. A chaque étape de calcul les cellules du réseau sont activées simultanément.

A la première étape de fonctionnement du réseau, seules les données présentes aux entrées de la cellule c_{11} sont différentes de zéro (a_{11} et b_{11}) (figure.III.2.b). Ainsi, uniquement le produit calculé par cette cellule ($a_{11}b_{11}$), différent de zéro, est mémorisé au niveau local. Ensuite, la donnée a_{11} présente à l'entrée horizontale est transmise à la cellule horizontale immédiatement voisine, et la donnée b_{11} présente à l'entrée verticale est transmise à la cellule verticale immédiatement voisine.

Pendant la seconde étape, les données présentes aux entrées du réseau sont : a_{12}, a_{21}, b_{21} et b_{12} (figure.III.2.c). Le produit formé par la cellule c_{11} (a_{12} b_{21}) est accumulé au produit précédent. Le premier élément c_{11} de la matrice produit *C*, alors obtenu, est mémorisé dans la cellule c_{11}. Au niveau de la cellule c_{12}, le produit calculé $a_{11}b_{12}$ est mémorisé. A la cellule c_{21}, le premier terme formé $a_{21}b_{11}$ est mémorisé. Les données présentes aux entrées des cellules sont ensuite transmises aux cellules voisines selon le principe décrit.

A la troisième étape, les données présentes aux entrées du réseau sont : a_{22} et b_{22}. Les éléments c_{12} et c_{21} sont obtenus selon la même procédure et ces données sont transmises à la dernière cellule.

Enfin durant la dernière étape, on obtient l'élément c_{22} de la même manière.

Si chaque étapes de calcul ou unité de temps comporte une multiplication et une addition, elle prend donc (m + l) cycles d'horloge pour s'exécuter (m et l étant le nombre de cycles d'horloge pour exécuter une multiplication et une addition respectivement)

Le nombre d'étapes de calcul nécessaire pour effectuer le produit de deux matrices carrées *A* et *B* de dimensions (N x N), sur un réseau systolique de dimension (N x N), est de $(2N\text{-}1) + (N\text{-}1) = 3N\text{-}2$ étapes.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Le terme 2*N*-1 de l'équation est le nombre d'étapes nécessaire pour alimenter le réseau avec les éléments des deux matrices et le terme *N*-1 est le nombre d'étapes supplémentaire pour la propagation des données à travers toutes les cellules du réseau (voir tableau.III.1).

Le nombre total d'étapes de calcul nécessaire pour effectuer la multiplication d'une matrice carrée *A* (*N* x *N*) par une matrice colonne *X* (*N* x 1), sur un réseau systolique (*N* x *N*), est de 2*N*-1 étapes. La version systolique de cet algorithme est représentée sur la figure.III.3. Ce réseau ne fonctionne pas selon le même principe que celui de la figure.III.2. Dans ce cas, les éléments de la matrice *A* sont enregistrés dans les cellules du réseau et les éléments du vecteur *X* sont introduit à l'entrée du réseau.

Cette dernière structure réalisant le produit d'une matrice carrée par un vecteur sera adoptée pour la réalisation des filtres RIF.

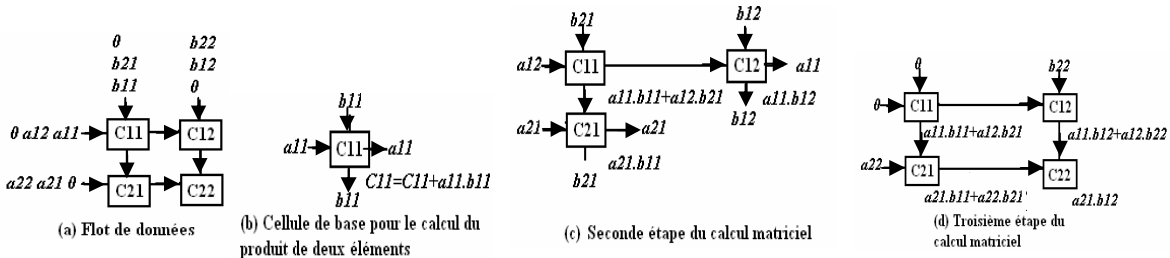


Fig.III.2. Structure systolique du produit de 2 matrices carrées.

Etape	Entrées						Sorties		
1		a_{11}			b_{11}				
2	a_{12}		a_{21}	b_{12}		b_{21}		c_{11}	
3		a_{22}			b_{22}		c_{12}		c_{21}
4								c_{22}	

Temps d'exécution = 4 unités de temps (étapes de calcul)

Tableau.III.1. Activité des entrées/sorties du réseau systolique de la figure.III.2.

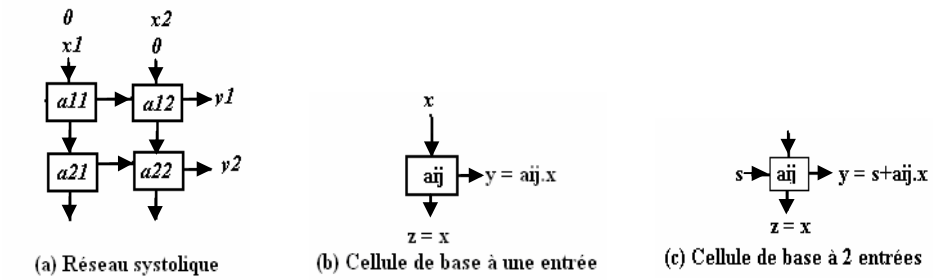


Fig.III.3. Structure systolique du produit d'une matrice carrée par un vecteur *Y = AX*.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Etape	Entrées		Sorties
1	$x1$		
2		$x2$	$y1$
3			$y2$

Temps d'exécution = 3 unités de temps (étapes de calcul)

Tableau.III.2. Activité des entrées/sorties du réseau systolique de la figure.III.3.

III.3. L'architecture systolique du type cylindrique [20,53,58,59,60]

Une architecture cylindrique est formée par un réseau de processeurs ou cellules élémentaires placés sur la surface d'un cylindre d'une manière régulière. Ces cellules élémentaires jouent le même rôle que celle utilisées dans les réseaux systoliques conventionnels de la section précédente. Nous allons utiliser cette architecture cylindrique pour réaliser les filtres RIF.

III.3.1. Principe des structures cylindriques [12,20]

L'exemple du produit de deux matrices de la section III.2 est repris pour illustrer le principe du calcul matriciel à l'aide d'un réseau cylindrique associé au produit de deux matrice (2 x 2) et son flot de données(figure.III.4). Les activités du réseau cylindrique sont résumées dans le tableau.III.3.

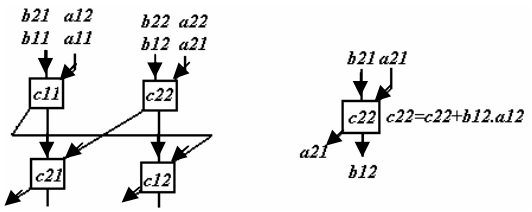


Fig.III.4. Structure cylindrique du produit de deux matrices (2x2).

Etape	Entrées				sorties	
1	a_{11}	a_{21}	b_{11}	b_{12}		
2	a_{12}	a_{22}	b_{22}	b_{22}	c_{11}	c_{22}
3					c_{21}	c_{12}

Temps d'exécution = 3 unités de temps

Tableau.III.3. Activité des entrées/sorties du réseau cylindrique de la figure.III.4.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Le nombre total d'étapes nécessaire pour déterminer tous les éléments du produit matriciel $C = A$ x B, (A et B sont des matrices (N x N)), dans un réseau cylindrique de dimension (N x N) est de $N + (N - 1) = 2N - 1$ étapes. Le produit d'une matrice carrée (N x N) par une matrice colonne (N x 1) nécessite N étapes pour calculer tous les éléments sur un réseau cylindrique (N x N).

On remarque donc que les réseaux cylindriques sont plus rapides que les réseaux systoliques de Kung pour le même nombre de cellules.

III.3. 2. Principe des structures toriques [12,58]

Le nombre total d'étapes de calcul nécessaire pour effectuer la multiplication d'une matrice carrée A (N x N) par une matrice colonne X (N x 1), sur un réseau systolique du type torique (N x N), est de N étapes de calcul. La version torique de cet algorithme est représentée sur la figure.III.5. Ce réseau ne fonctionne pas selon le même principe que celui de la figure.III.4. Dans ce cas, les éléments de la matrice A sont enregistrés dans les cellule du réseau et les éléments du vecteur X sont introduit à l'entrée du réseau. Les éléments du vecteur $Y = A$ x X sont obtenus en même temps sur les sorties transversals des cellules de sortie à la dernière étape.

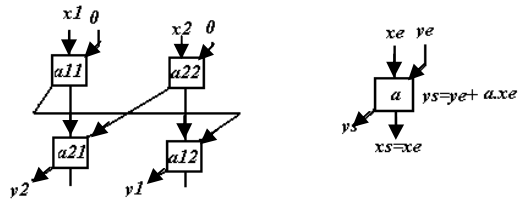


Figure.III.5. Structure torique du produit de deux matrices (2 x 2).

Etape	Entrées		Sorties	
1	x_1	x_2		
2			y_2	y_1

Temps d'exécution = 2 unités de temps

Tableau.III.4. Activité des entrées/sorties du réseau torique de la figure 3.5.

III.4. Application des architectures systoliques pour la réalisation des filtres numériques RIF

Les réseaux de processeurs parallèles ont reçu une grande attention dans le domaine du traitement numérique du signal ces dernières années **[8,9,10,11,12,17,21,22,23]**. Leurs caractéristiques de

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

grande vitesse, de régularité et de modularité les rendent très adaptées pour la plupart des algorithmes de traitement numérique du signal. En particulier, les réseaux de processeurs pouvant exploiter réellement les possibilités de la technologie VLSI constituent une solution aux exigences de traitement numérique en temps réel.

L'objectif de cette section consiste à adapter des architectures systoliques à l'implémentation des filtres numériques RIF représentés par une convolution rapide qui nécessite des calculs matriciels. Les architectures systoliques ont été particulièrement appliquées aux filtres non récursifs RIF **[10,22,51,52,63,64,67]**.

III.4.1. Principes des structures systoliques dédiées au produit matrice/vecteur [60,61,62]

Nous avons vu, qu'une architecture systolique est définie comme un réseau de cellules (processeur) élémentaires interconnectées entre elles. Une cellule élémentaire peut réaliser des simples opérations telles que la multiplication, l'addition et la mémorisation.

Plusieurs algorithmes réalisant l'implémentation du produit de deux matrices ont été proposés par Kung **[51,63,68]**. Nous utiliserons dans ce travail l'architecture systolique de Kung qui exige moins de temps de calcul.

Il y a beaucoup d'algorithmes séquentiels pour calculer un produit de matrice, comme l'algorithme standard de multiplication, l'algorithme de Winograd, et les algorithmes de Strassen [**59**].

Le calcul direct du produit de deux matrices par la multiplication standard se fait selon la formule:

$$C_{ij} = C_{ij} + \sum_{k=1}^{n} a_{ik} . b_{kj} \quad \text{Pour } i \geq 1,\ j \leq n \qquad \text{(III.1)}$$

Nous considérons l'exemple, utile à la réalisation des filtres numérique RIF sous forme de produit matricielle, du produit d'une matrice *A* carrée (2 x 2) par un vecteur *X*. soit *Y* = *A.X* le vecteur produit, avec :

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},\ X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},\quad Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

Les filtres non récursifs, étant caractérisés par une matrice carrée formée des coefficients fixes qui doivent être enregistrés dans les mémoires des cellules du réseau systolique.

La figure.III.6 décrit une version systolique de l'algorithme de calcul du produit d'une matrice A carrée (2 x 2) par un vecteur *X* et le flot de données correspondant. Les éléments a_{ij} de la matrice

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

A sont chargés dans les cellules du réseau. Le calcul des éléments du vecteur *Y* est décomposé en taches élémentaires effectuées par des cellules dont le principe de fonctionnement est donné à la figure.III.6.b.

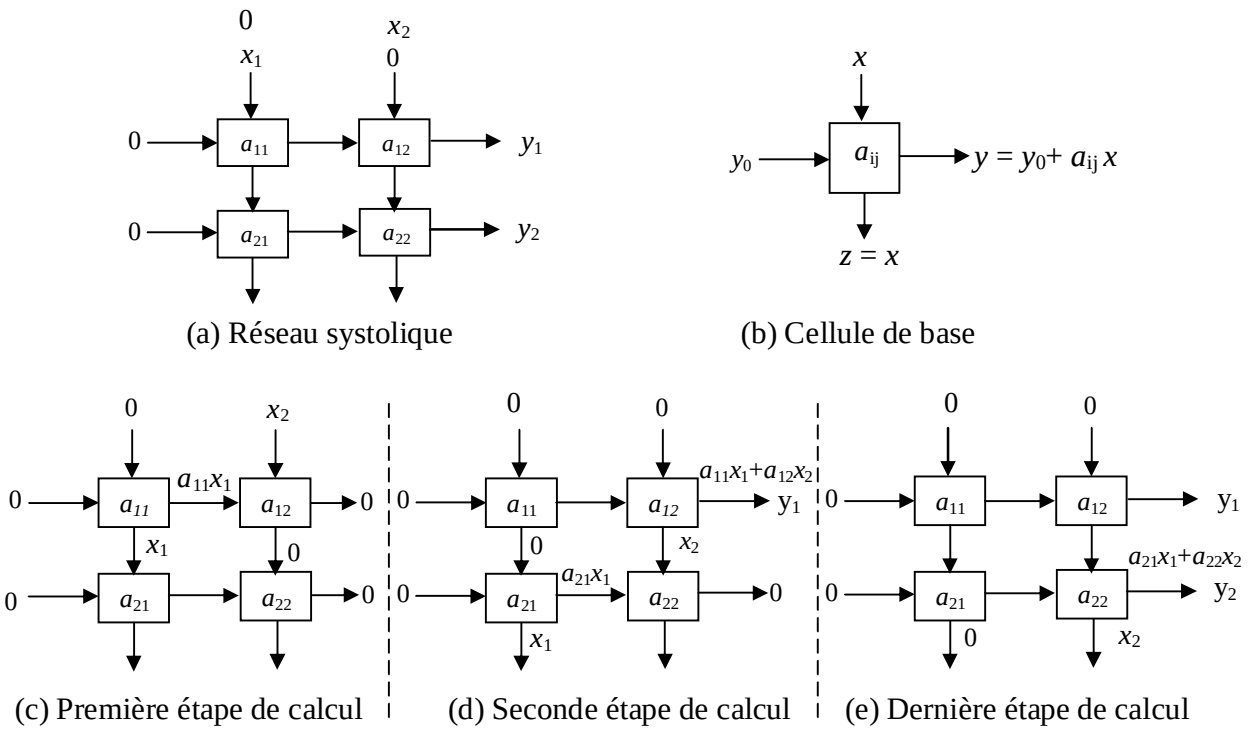
Chaque cellule multiplie la donnée présente à l'entrée verticale par le scalaire stocké dans son registre interne. Ce produit est accumulé à la donnée présente à l'entrée horizontale puis le résultat est transmis horizontalement à la cellule suivante. La donnée présente à l'entrée verticale est transmise verticalement sans modification à la cellule voisine.

La figure.III.6.c illustre le fonctionnement du réseau à la fin de la première étape de calcul.

La figure.III.6.d illustre le fonctionnement du réseau à la fin de la seconde étape de calcul, l'élément y_1 est déterminé. A la dernière étape l'élément y_2 est obtenu. Le tableau.III.5. résume les activités des entrées/sorties du réseau systolique et le nombre d'étapes nécessaire pour obtenir tous les éléments du vecteur produit *Y*.

En général, le nombre total d'étapes de calcul nécessaire pour effectuer la multiplication d'une matrice carrée *A* (*N* x *N*) par une matrice colonne *X* (*N* x *N*), sur un réseau systolique (*N* x *N*), est de 2*N*-1 étapes.

Cette dernière structure réalisant le produit d'une matrice carrée par un vecteur sera adoptée pour l'implémentation des filtres numérique RIF.



(a) Réseau systolique (b) Cellule de base

(c) Première étape de calcul (d) Seconde étape de calcul (e) Dernière étape de calcul

Fig.III.6. Structure systolique du produit d'une matrice carrée par un vecteur *Y=AX*.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Etape	Entrées		Sortie
1	x_1		
2		x_2	y_1
3			y_2

Tableau.III.5. Activité des entrées/sorties du réseau systolique de la figure.III.6.

III.4.2. Structure systolique directe des filtres numériques RIF représentés par une convolution ordinaire

Dans cette section, nous allons présenter la structure systolique pour la mise en œuvre des filtres RIF avec les améliorations en terme d'implémentation et la simplicité durant l'utilisation de l'architecture par rapport à l'architecture VLSI de Parhi.

Un filtre RIF peut être décrit par la forme matricielle $Y_{2n-1}= C_n X_n$, avec C_n de dimension $(2n-1 \times n)$, après transformation par les algorithmes de Cook-Toom ou Winograd on obtient la forme suivante: $Y_L= C_L X_L$ avec C_L de dimension $(L \times L)$ [**1, 4, 45,46**].

$$Y_L = C_L X_L \tag{III.2}$$

$$\begin{bmatrix} y_n \\ y_{n-1} \\ y_{n-2} \\ y_{n-3} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_n \\ x_{n-1} \\ x_{n-2} \\ x_{n-3} \end{bmatrix} \tag{III.3}$$

Ainsi, l'opération de filtrage réalisée par un filtre RIF est équivalente au produit d'une matrice carrée (L x L) par un vecteur colonne (L x 1). L'utilisation du réseau systolique de la figure.III.6 calculant le produit d'une matrice par un vecteur permet la mise en œuvre directe du filtre non récursif. Les cellules élémentaires doivent être chargées initialement par la séquence suivante :

$$C_L = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}.$$

Pour que le réseau systolique fonctionne selon l'algorithme de filtrage RIF. Des cellules a_{ij} sont chargées par les éléments de la matrice C_L. Dans ce type de structures, les entrées du réseau sont utilisées pour charger le signal d'entrée du filtre pour produire le signal de sortie du réseau. La figure.III.7 décrit le principe de fonctionnement de la cellule de base et son implémentation

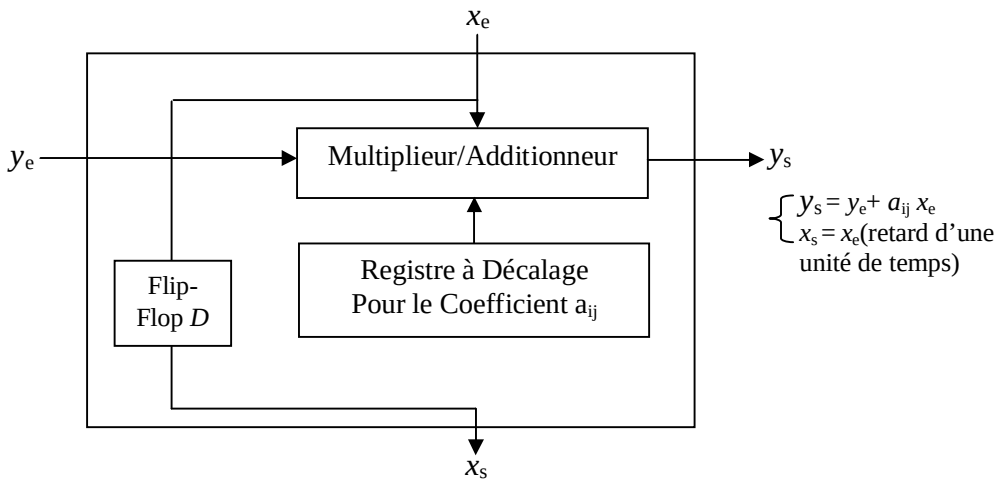
This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

VLSI. La cellule calcule le produit du coefficient stocké dans sa mémoire avec son entrée verticale et l'accumule à son entrée horizontale.

Le nombre d'étapes de calcul nécessaires pour obtenir un échantillon $y(n)$ à la sortie d'un filtre RIF (4 x 4) sur un réseau systolique de dimension (L x L) est de $2L$-1 à l'initialisation du réseau (Calcul du premier échantillon de sortie) et de L étapes pour le reste du fonctionnement du réseau (Calcul des échantillons suivants).

III.4.3. Construction de la cellule élémentaire du réseau systolique

Chaque cellule du réseau systolique est construite à partir d'un multiplieur/additionneur, un registre à décalage, et un flip-flop du type *D*. le multiplieur/additionneur assure le calcul, $y_s = y_e + a_{ij}\, x_e$, le registre à décalage permet la mémorisation du coefficient a_{ij}, alors que le flip-flop *D* permet la transmission de l'entrée verticale de la cellule vers sa sortie verticale avec un retard d'une unité de temps $x_s = x_e$(figure.III.7).



(a) Construction de la cellule du réseau systolique.

Fig .III.7. Construction et fonctionnement des cellules du réseau systolique.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

La figure.III.8 représente le réseau systolique d'un filtre RIF à 4-taps à partir de la convolution ordinaire et le fonctionnement de cellule.

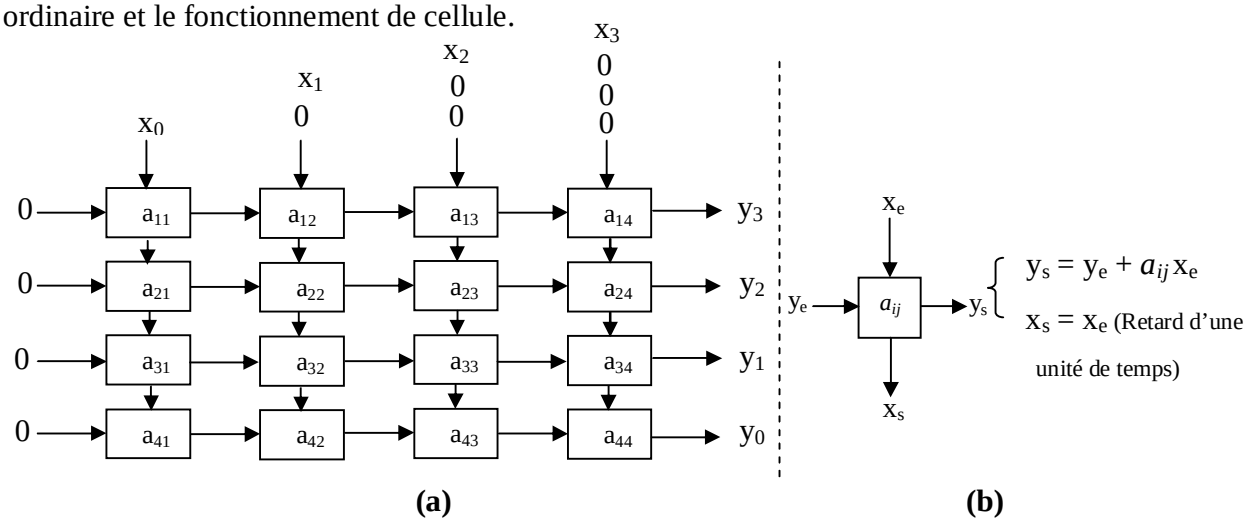
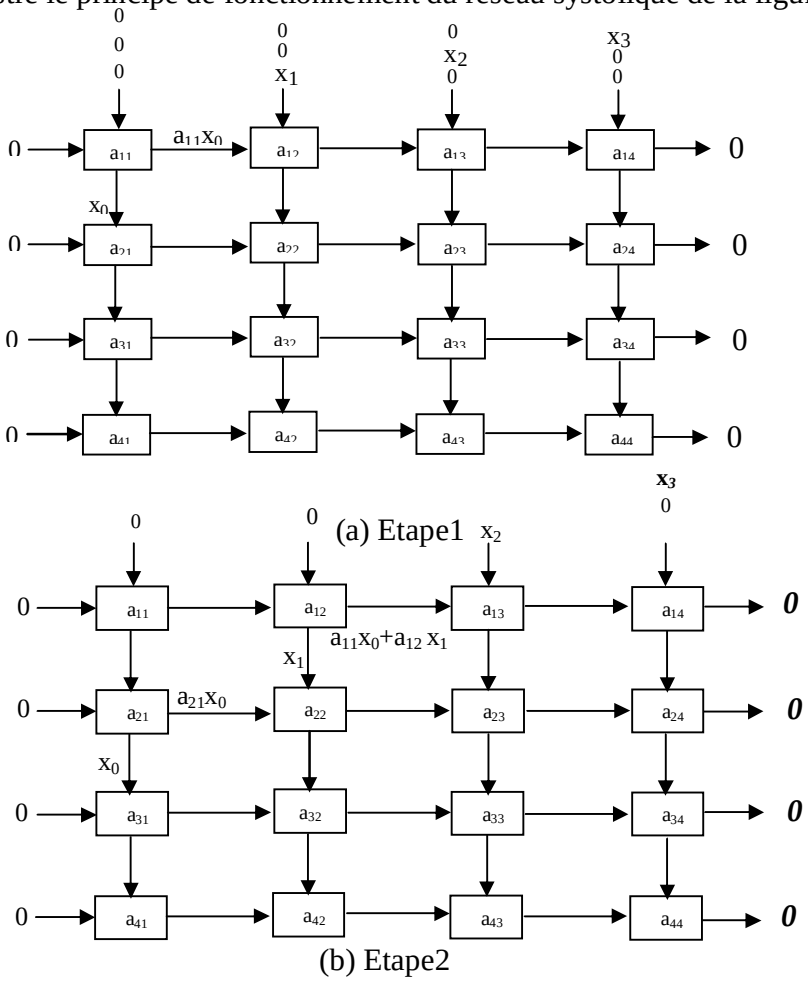
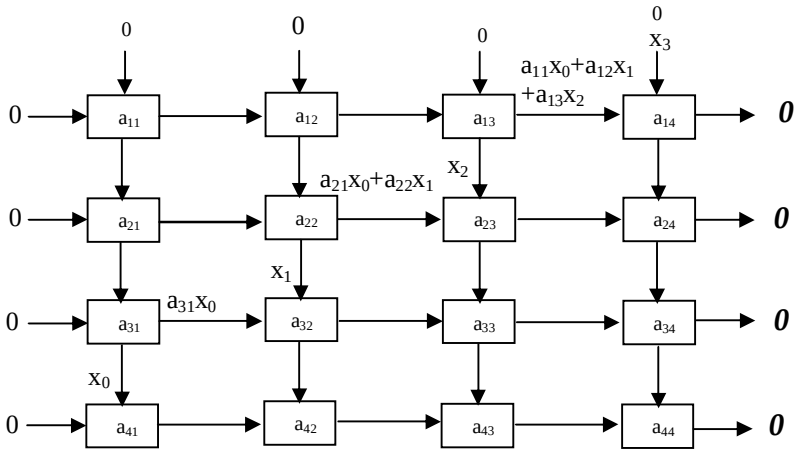


Fig.III.8. (a) Réseau systolique d'un filtre RIF 4-parallèle type Kung à partir de la convolution ordinaire, (b) Fonctionnement de la cellule élémentaire.

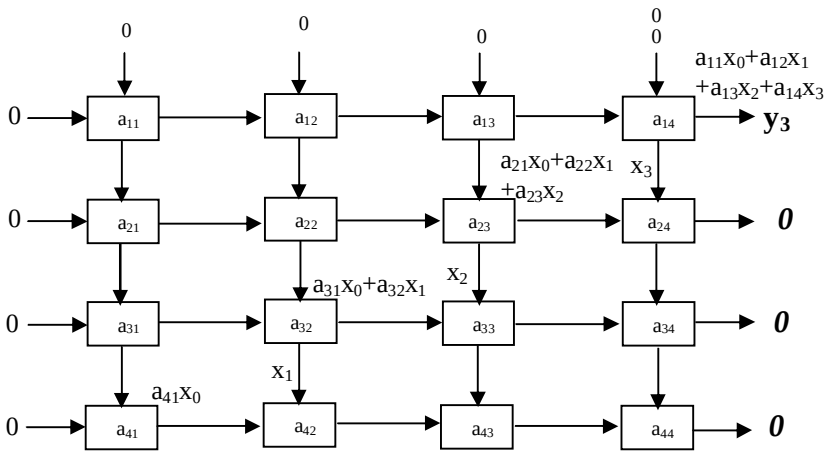
La figure.III.9 illustre le principe de fonctionnement du réseau systolique de la figure.III.8.



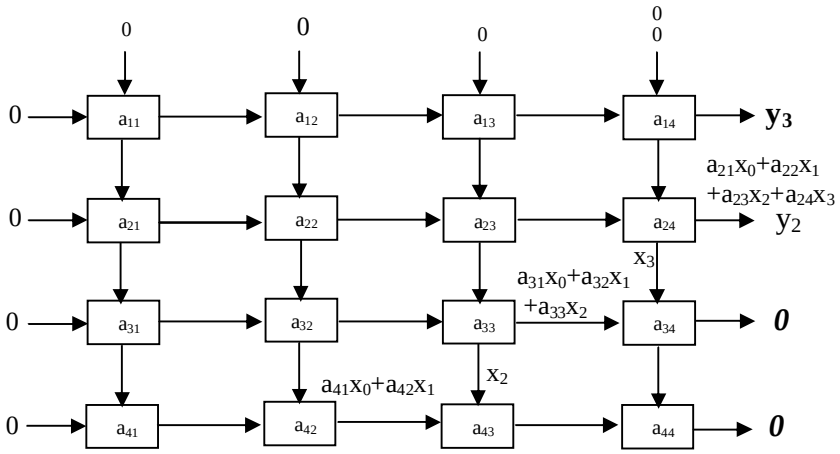
This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



(c) Etape3

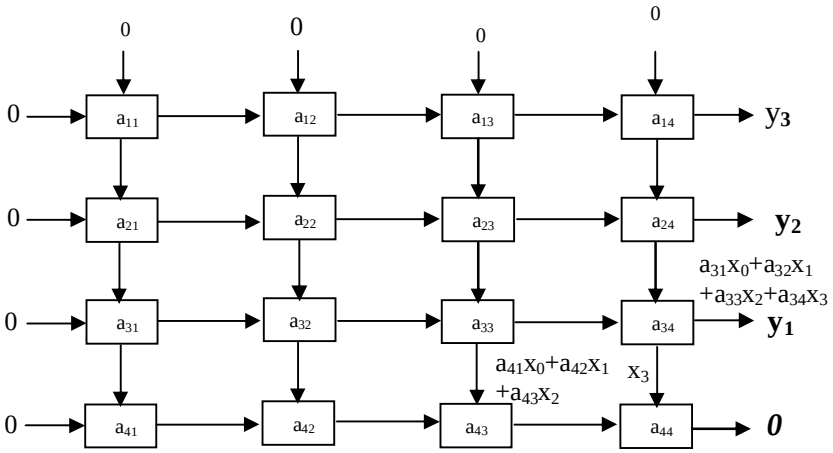


(d) Etape 4

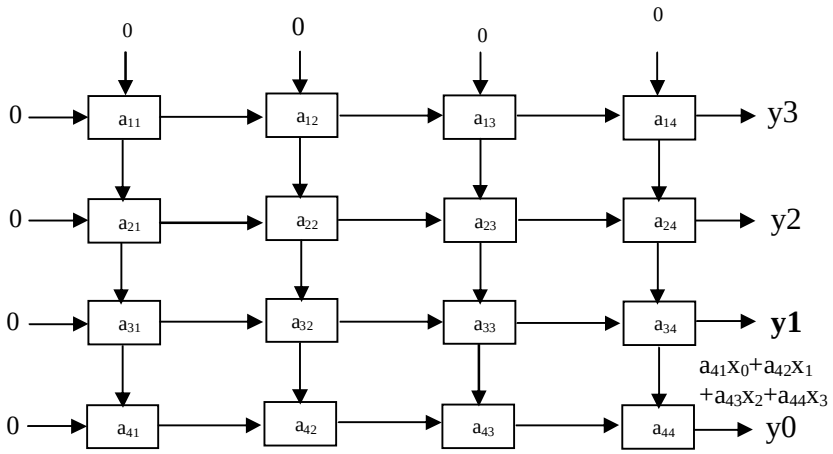


(e) Etape 5

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



(f) Etape 6



(g) Etape 7

Fig.III.9. Principe de fonctionnement du réseau systolique de type Kung.

III.5. Estimation du temps d'exécution des calculs sur réseau systolique directe de type Kung (pour un filtre à 4-taps)

Si chaque étapes de calcul ou unité de temps comporte une multiplication et une addition, elle prend donc ($m + l$) cycles d'horloge pour s'exécuter (m et l étant le nombre de cycles d'horloge pour exécuter une multiplication et une addition respectivement).

- Pour la figure.III.8. le temps d'exécution est : $t_{4\text{-taps-direct}} = 7(m + l)$
- Pour le cas de N-taps le temps d'exécution est : $t_{N\text{-taps}} = (2N\text{-}1)(m + l)$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

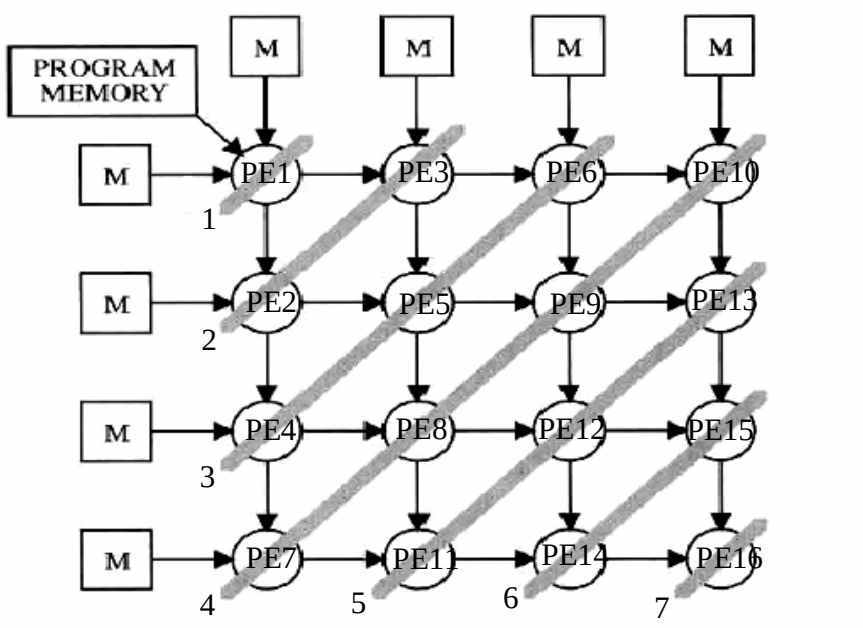


Fig.III.10. Estimation du temps d'exécution des calculs sur réseau systolique type Kung

III.6. Implémentation directe du filtre RIF sur un réseau type cylindrique à partir de la convolution ordinaire

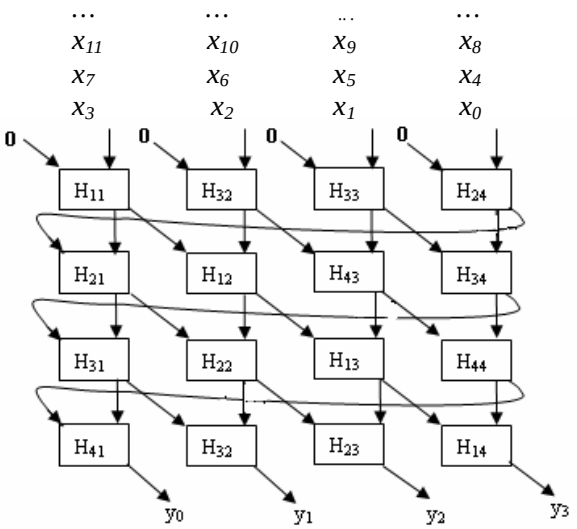


Fig. III.11. Implémentation d'un filtre RIF 4-parallèle sur un réseau systolique de type cylindrique.

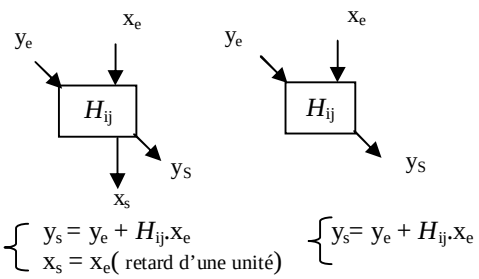
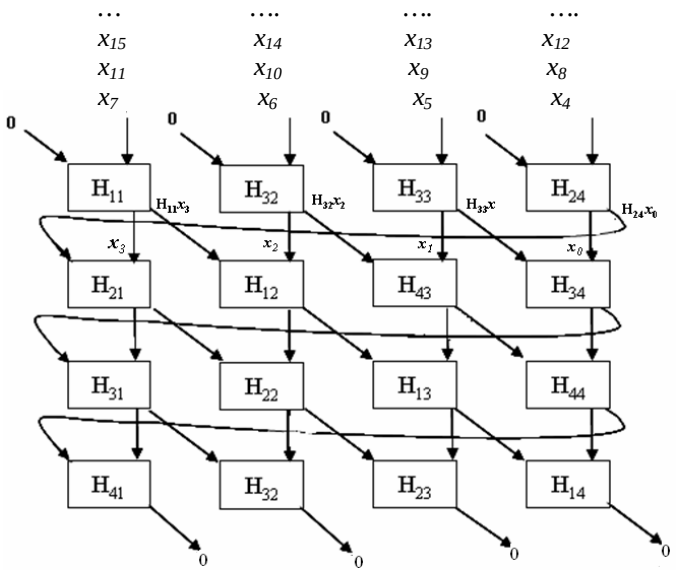


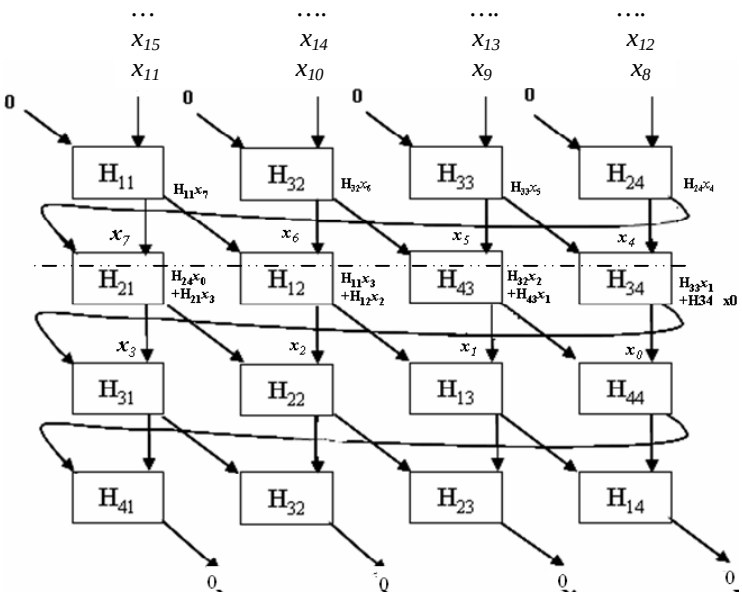
Fig. III.12. Fonctionnement des cellules du réseau systolique de la figure.III.22.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

La figure.III.13. illustre le principe de fonctionnement du réseau systolique de type cylindrique.

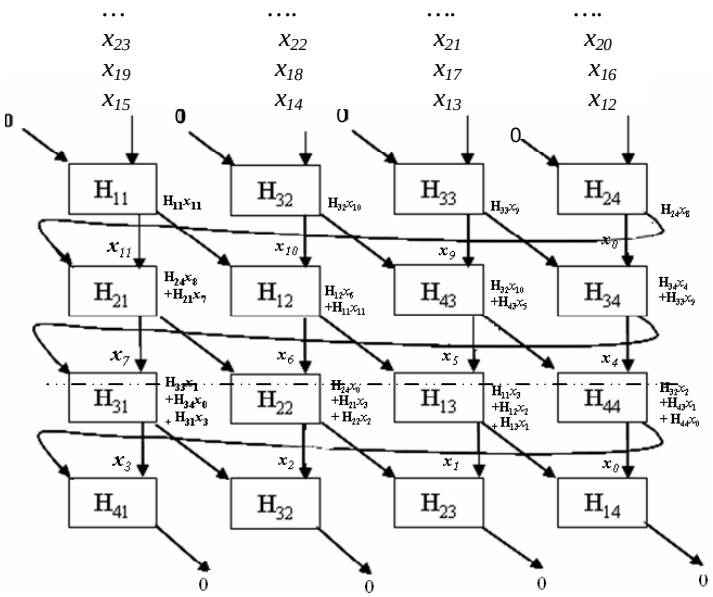


(a) étape 1

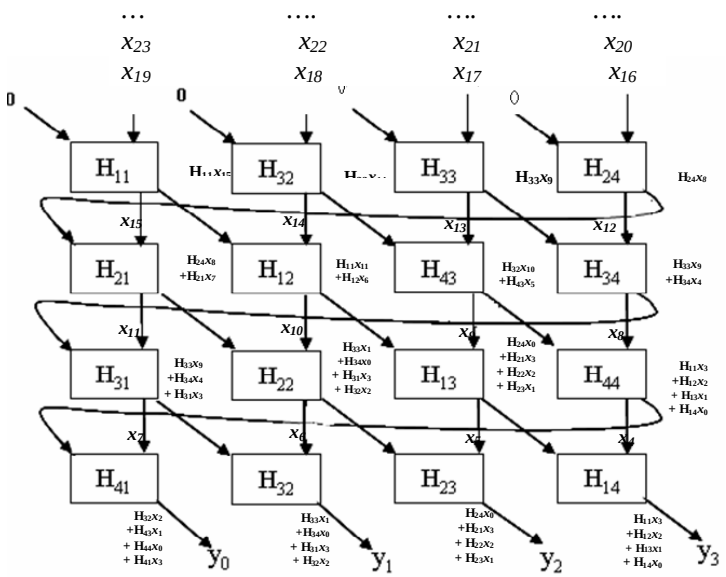


(a) étape 2

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



(c) étape 3



(d) étape 4

Fig.III.13. Principe de fonctionnement du réseau systolique de type cylindrique.

Le temps d'exécution pour un réseau systolique de type cylindrique est (figure.III.11):

$$t_{Cyl} = 4\,(m + l) \qquad \text{(III.15)}$$

- à l'initialisation du réseau le temps de calcul est : $t_{Cyl} = 4\,(m + l)$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

- Pour L = N_{-taps} le temps d'exécution est : $t_{Cyl} = L(m + l)$

- Pour le fonctionnement permanent une seule étape de calcul : $t_{Cyl} = (m + l)$

Le temps d'exécution sur réseau systolique type Kung et cylindrique à partir d'une convolution ordinaire est donné sur le tableau.III.6 :

L	***N*-taps**	**type Kung (t_{kung})**	**type Cylindrique (t_{Cyl})**
4	4	$7(m + l)$	$4(m + l)$
6	6	$11(m + l)$	$6(m + l)$
8	8	$15(m + l)$	$8(m + l)$
12	12	$23(m + l)$	$12(m + l)$
16	16	$31(m + l)$	$16(m + l)$
18	18	$35(m + l)$	$18(m + l)$
24	24	$47(m + l)$	$24(m + l)$
36	36	$71(m + l)$	$36(m + l)$
48	48	$95(m + l)$	$48(m + l)$
54	54	$107(m + l)$	$54(m + l)$
72	72	$143(m + l)$	$72(m + l)$

Tableau.III.6. Le temps d'exécution pour différents types des réseaux systoliques (cas de *L* = *N*) pour l'implémentation direct d'un filtre RIF *L*-parallèle à N-taps à partir d'une convolution ordinaire.

III.7. Implémentation du filtre RIF représenté par la courte convolution itérative sur l'architecture systolique type Kung

Nous considérons l'architecture d'un filtre RIF 4-parallèle à 4-taps, Les convolutions 4 x 4 (L = 4) peuvent être décomposées en deux courtes convolutions 2 x 2 selon l'équation:

$$Y_4 = (P^T{}_2 \otimes P^T{}_2).H_{4\_22}.\ (Q^T{}_2 \otimes Q^T{}_2).\ A^T{}_{4\_22}.X_4 \qquad \text{(III.4)}$$

La matrice $A^T{}_{4\_22}$ est utilisée pour arranger les entrées.

La matrice H_{4_22} définit les sous filtres (qui contient les coefficients du filtre).

$P^T{}_2$, $Q^T{}_2$: sont les matrice de pré addition et post addition, respectivement.

Y_4, X_4: sont deux séquence de sortie et d'entrée, respectivement.

$$P_2^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix},\ Q_2^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

1ère étape: Calcul du produit $A^T_{4\_22}.X_4$ pour arranger les entrées.

$$X_4 = \begin{bmatrix} x3 & x2 & x1 & x0 & x3z^{-4} & x2z^{-4} & x1z^{-4} \end{bmatrix}^T, \qquad A_{4\_22} = \begin{bmatrix} 1000000 \\ 0100000 \\ 0010000 \\ 0010000 \\ 0001000 \\ 0000100 \\ 0000100 \\ 0000010 \\ 0000001 \end{bmatrix}.$$

$A_{4\_22}.X_4 = \begin{bmatrix} x_3 & x_2 & x_1 & x_1 & x_0 & z^{-4}x_3 & z^{-4}x_3 & z^{-4}x_2 & z^{-4}x_1 \end{bmatrix}^T$, qui donne 9 éléments partagés en 3 sous vecteur: $X_3 = \begin{bmatrix} x_3 & x_2 & x_1 \end{bmatrix}^T$, $X_2 = \begin{bmatrix} x_1 & x_0 & z^{-4}x_3 \end{bmatrix}^T$, $X_1 = \begin{bmatrix} z^{-4}x_3 & z^{-4}x_2 & z^{-4}x_1 \end{bmatrix}^T$.

On pose $D = Z^{-4}$ (Z : Opérateur de transformée en Z, D : élément de retard)

2ème étape:

Nous pouvons écrire l'algorithme (III.4) comme suivant :

$$[y_3\, y_2]^T = P^T_2.H'_1.Q^T_2.\,(X_3 + X_2) - P^T_2.H'_2.Q^T_2.\,X_2 \qquad \text{(III.5)}$$

$$[y_1\, y_0]^T = P^T_2.H'_2.Q^T_2.\,X_2 + P^T_2.H'_3.Q^T_2.\,(X_1 + X_2) \qquad \text{(III.6)}$$

Avec $X_3 = [x_3 \quad x_2 \quad x_1]^T$, $X_2 = [x_1 \quad x_0 \quad z^{-4}x_3]^T$, $X_1 = [\, z^{-4}x_3 \quad z^{-4}x_2 \quad z^{-4}x_1]^T$.

$$H'_1 = \begin{bmatrix} H_0 & 0 & 0 \\ 0 & H_0 - H_1 & 0 \\ 0 & 0 & H_1 \end{bmatrix}, H'_2 = \begin{bmatrix} H_0 - H_2 & 0 & 0 \\ 0 & H_0 - H_1 - H_2 + H_3 & 0 \\ 0 & 0 & H_1 - H_3 \end{bmatrix} et\ H'_3 = \begin{bmatrix} H_2 & 0 & 0 \\ 0 & H_2 - H_3 & 0 \\ 0 & 0 & H_3 \end{bmatrix}.$$

L'algorithme final représentant le filtre RIF:

$$\begin{cases} \begin{bmatrix} y_3 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} H_0 & 0 & 0 \\ 0 & H_0 - H_1 & 0 \\ 0 & 0 & H_1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_3 + x_1 \\ x_2 + x_0 \\ x_1 + z^{-4}x_3 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} H_0 - H_2 & 0 & 0 \\ 0 & H_0 - H_1 - H_2 + H_3 & 0 \\ 0 & 0 & H_1 - H_3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_0 \\ z^{-4}x_3 \end{bmatrix} \text{(III.7)} \\ \begin{bmatrix} y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} H_0 - H_2 & 0 & 0 \\ 0 & H_0 - H_1 - H_2 + H_3 & 0 \\ 0 & 0 & H_1 - H_3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_0 \\ z^{-4}x3 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} H_2 & 0 & 0 \\ 0 & H_2 - H_3 & 0 \\ 0 & 0 & H_3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} z^{-4}x_3 + x_1 \\ z^{-4}x_2 + x_0 \\ z^{-4}x_1 + z^{-4}x_3 \end{bmatrix} \text{(III.8)} \end{cases}$$

3ème étape: Implémentation de l'algorithme Y_4 par l'architecture systolique.

L'évaluation de Y_4 exige le calcul de :

1) $P^T_2.H'_1.Q^T_2.\,(X_3 + X_2)$.

2) $P^T_2.H'_2.Q^T_2.\,X_2$.

3) $P^T_2.H'_3.Q^T_2.\,(X_1 + X_2)$.

L'implémentation directe de l'algorithme de filtrage des l'équations (III.7) et (III.8) par un réseau systolique de type Kung est donnée par la figure.III.14.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

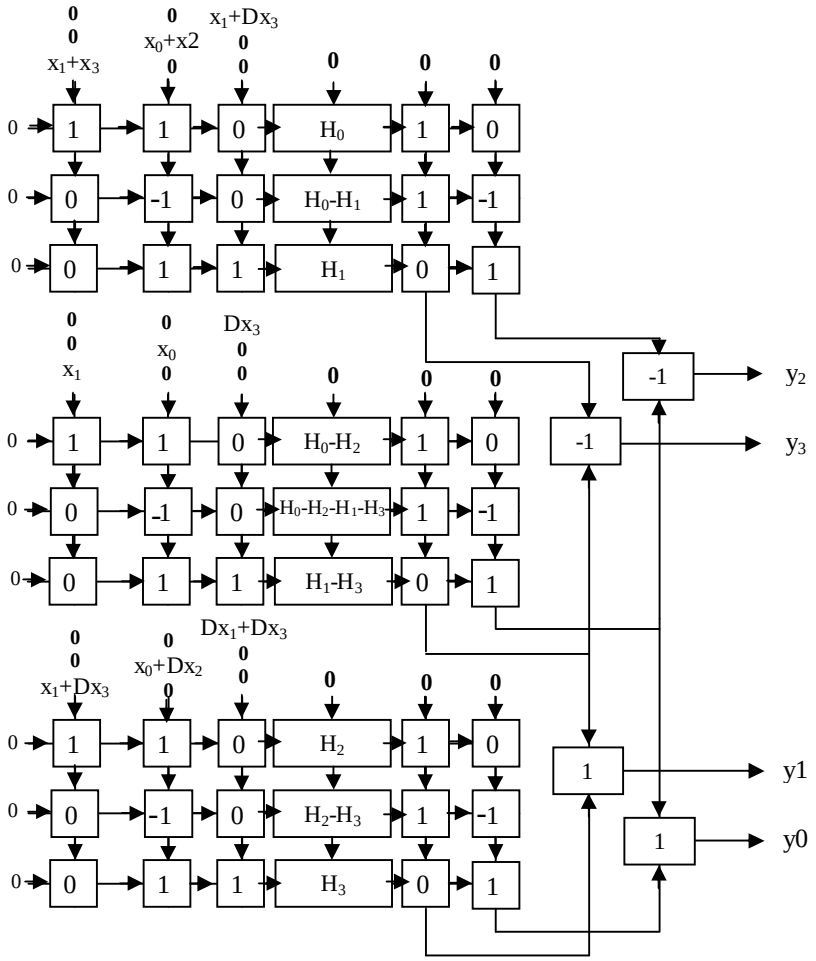


Fig. III.14. Implémentation d'un filtre RIF 4-parallèle sur un réseau systolique de type Kung à partir de la courte convolution itérative.

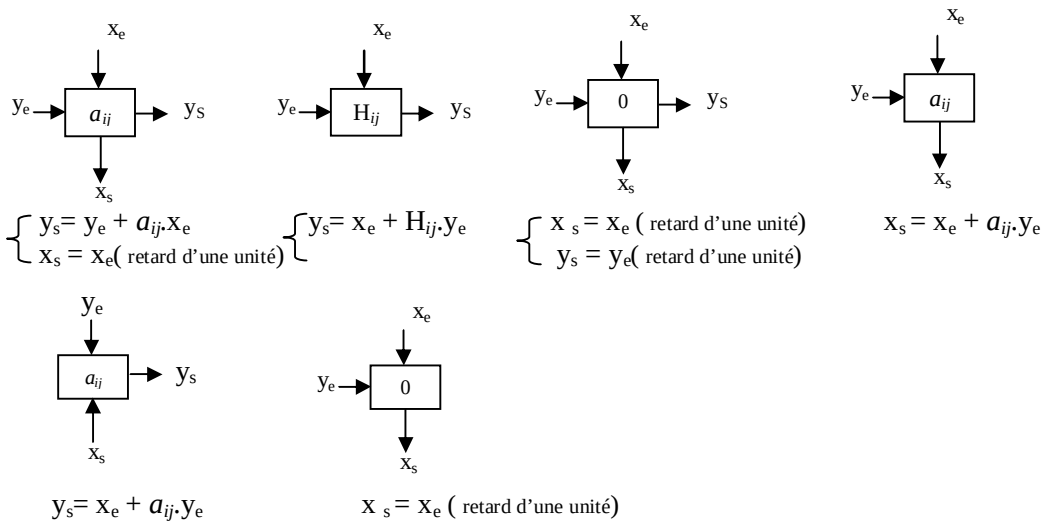


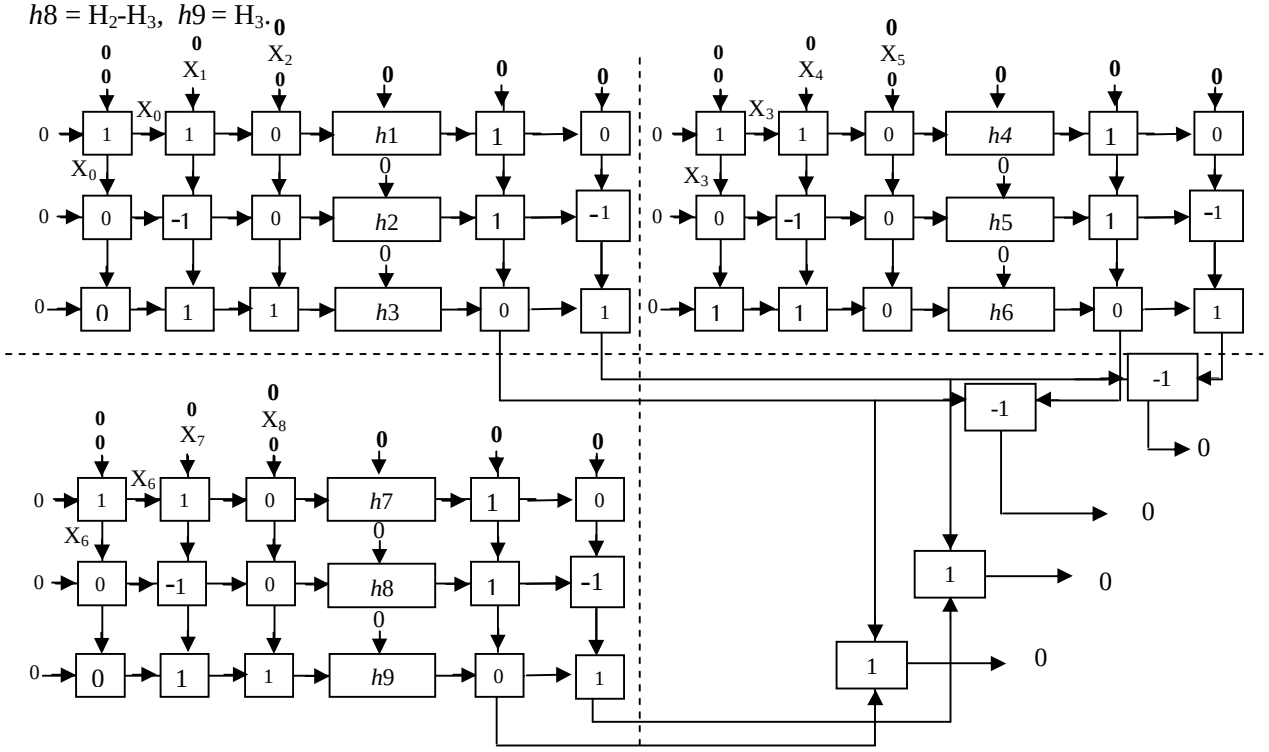
Fig. III.15. Fonctionnement des cellules du réseau systolique de la figure.III.14.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

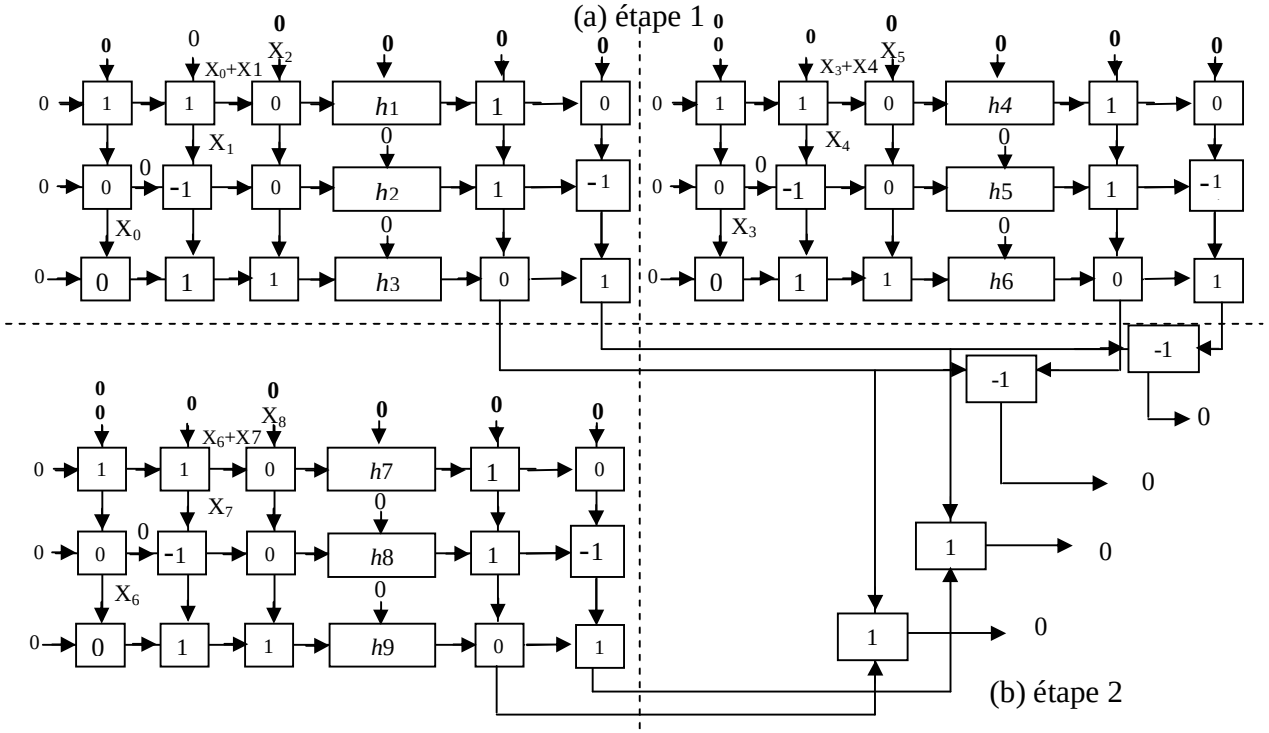
Le principe de fonctionnement de l'architecture systolique de la figure.III.14 est donnée sur a figure.III.16.

Pour simplifier le principe de fonctionnement de l'architecture systolique de type Kung, on pose :$X_0 = x_1$, $X_1 = x_0$, $X_2 = D\,x_3$, $X_3 = x_1 + x_3$, $X_4 = x_0 + x_2$, $X_5 = x_1 + D\,x_3$, $X_6 = x_1 + D\,x_3$, $X_7 = x_0 + D\,x_2$, $X_8 = D\,x_1 + Dx_3$

et $h1 = H_0\text{-}H_2$, $h2 = H_0\text{-}H_1\text{-}H_2\text{+}H_3$, $h3 = H_1\text{-}H_3$, $h4 = H_0$, $h5 = H_0\text{-}H1$, $h6 = H_1$, $h7 = H_2$, $h8 = H_2\text{-}H_3$, $h9 = H_3$.

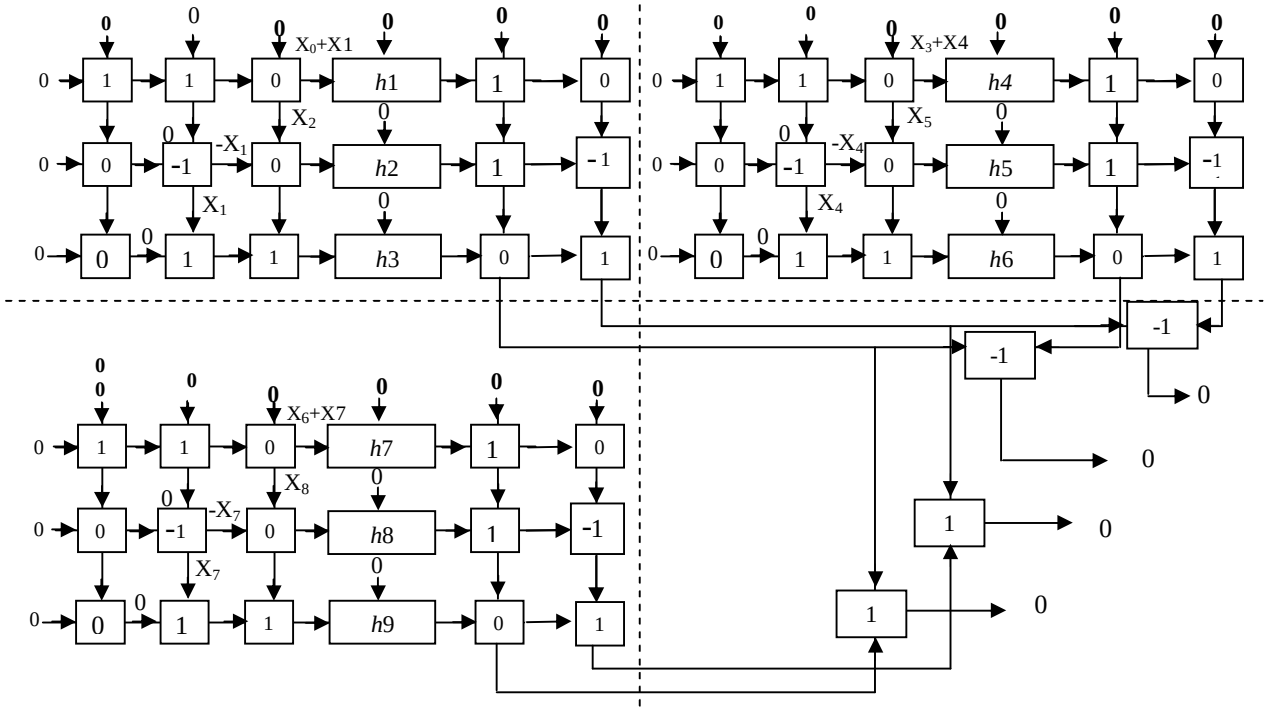


(a) étape 1

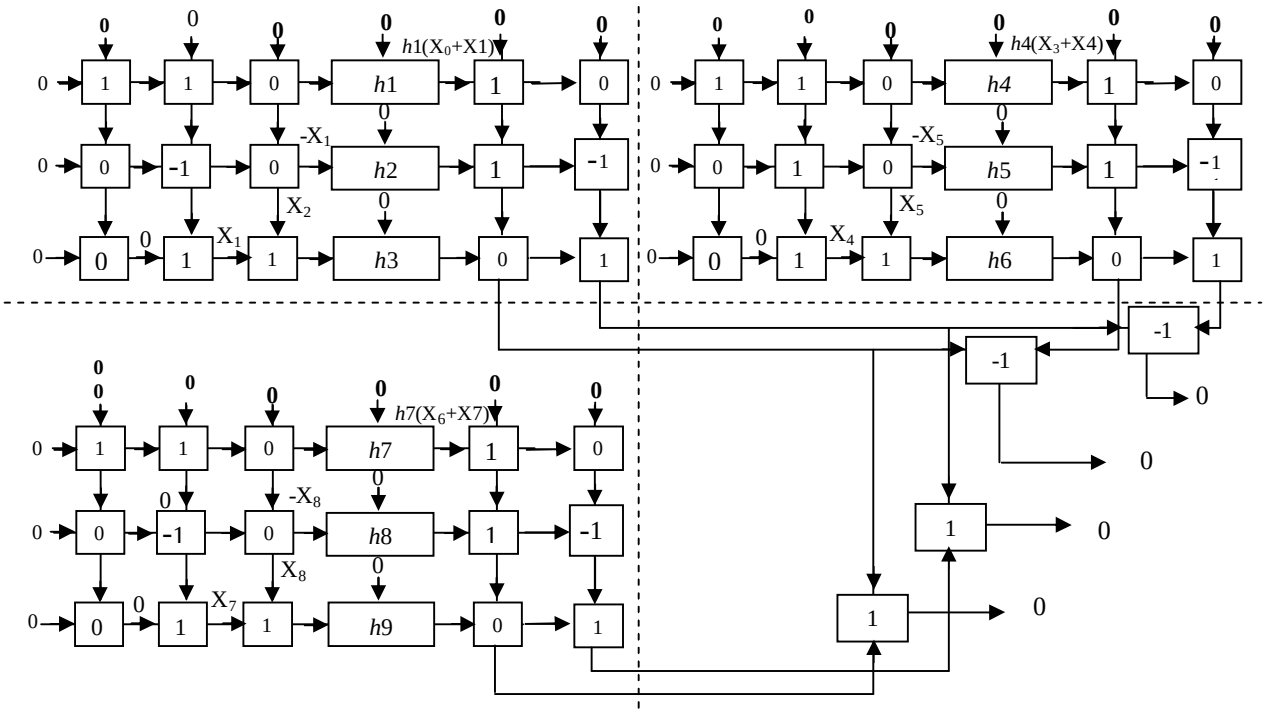


(b) étape 2

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

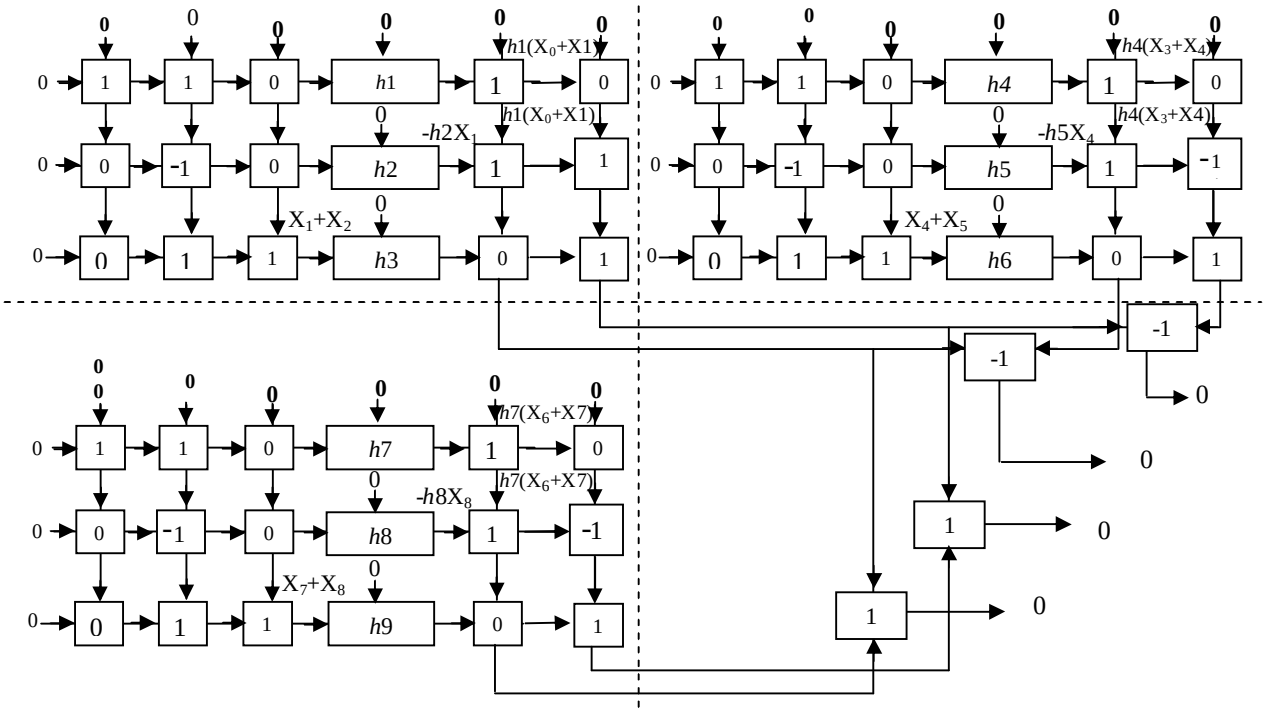


(c) étape 3

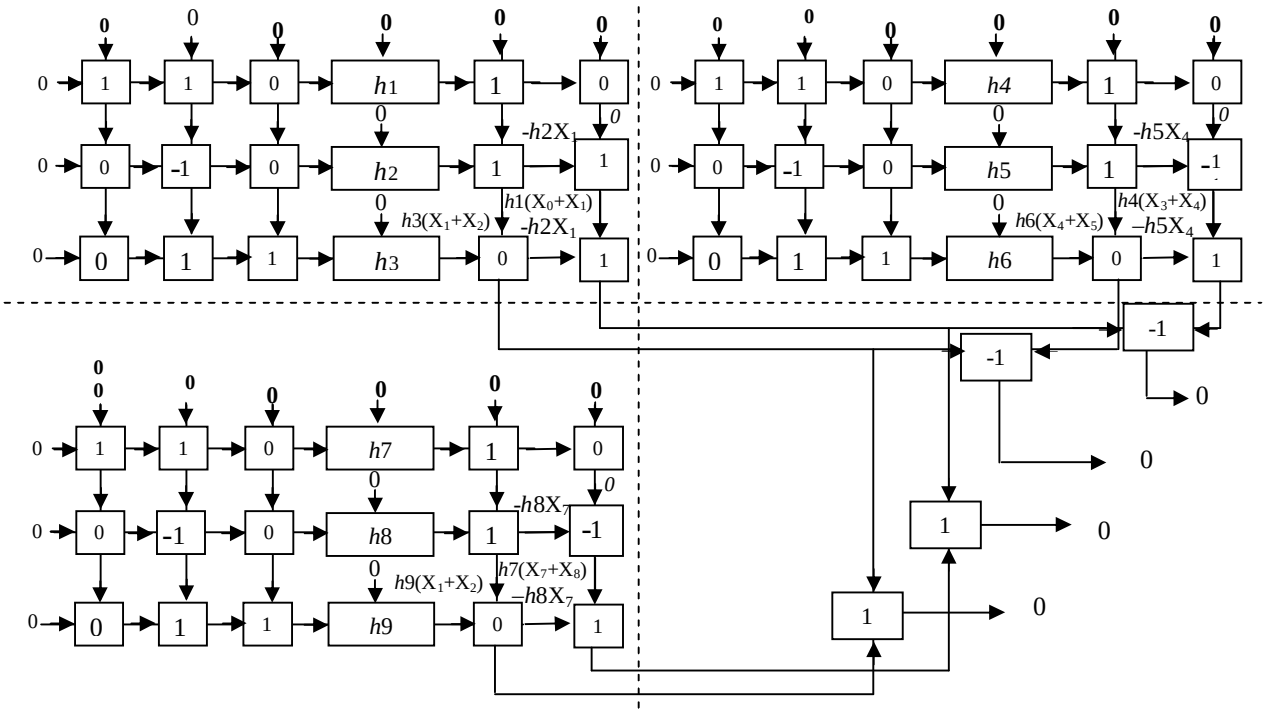


(d) étape 4

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

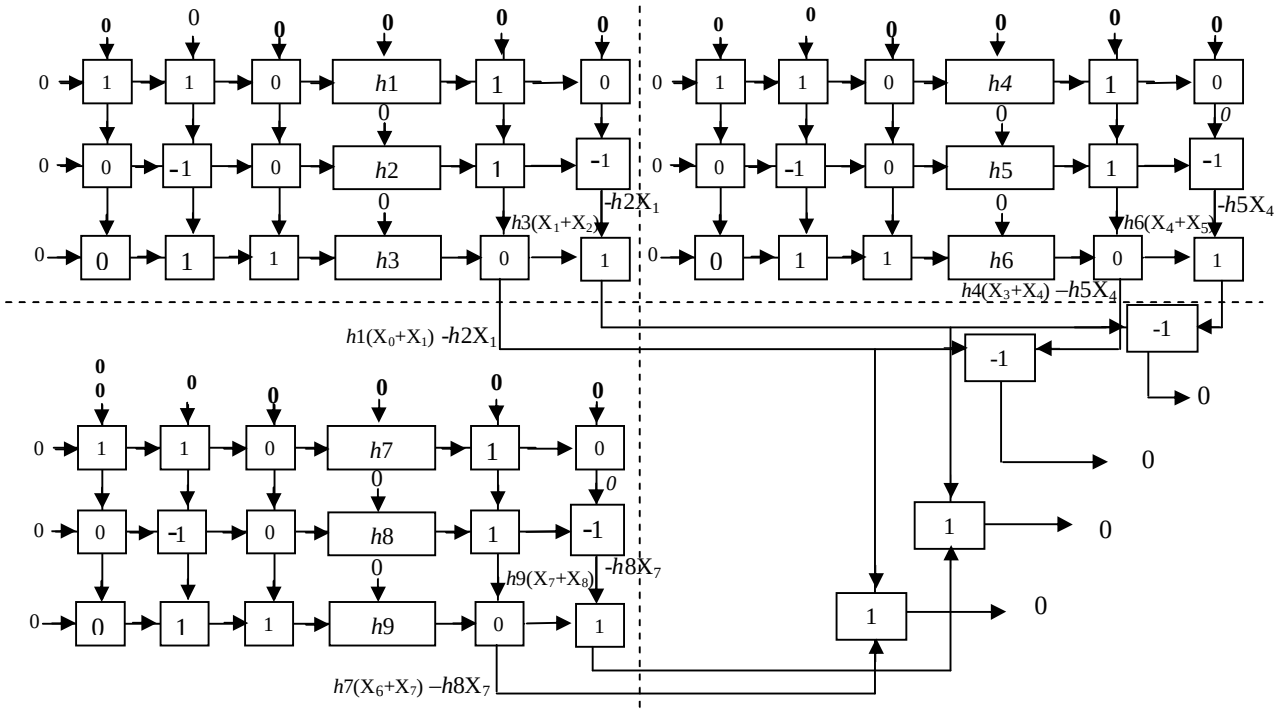


(e) étape 5

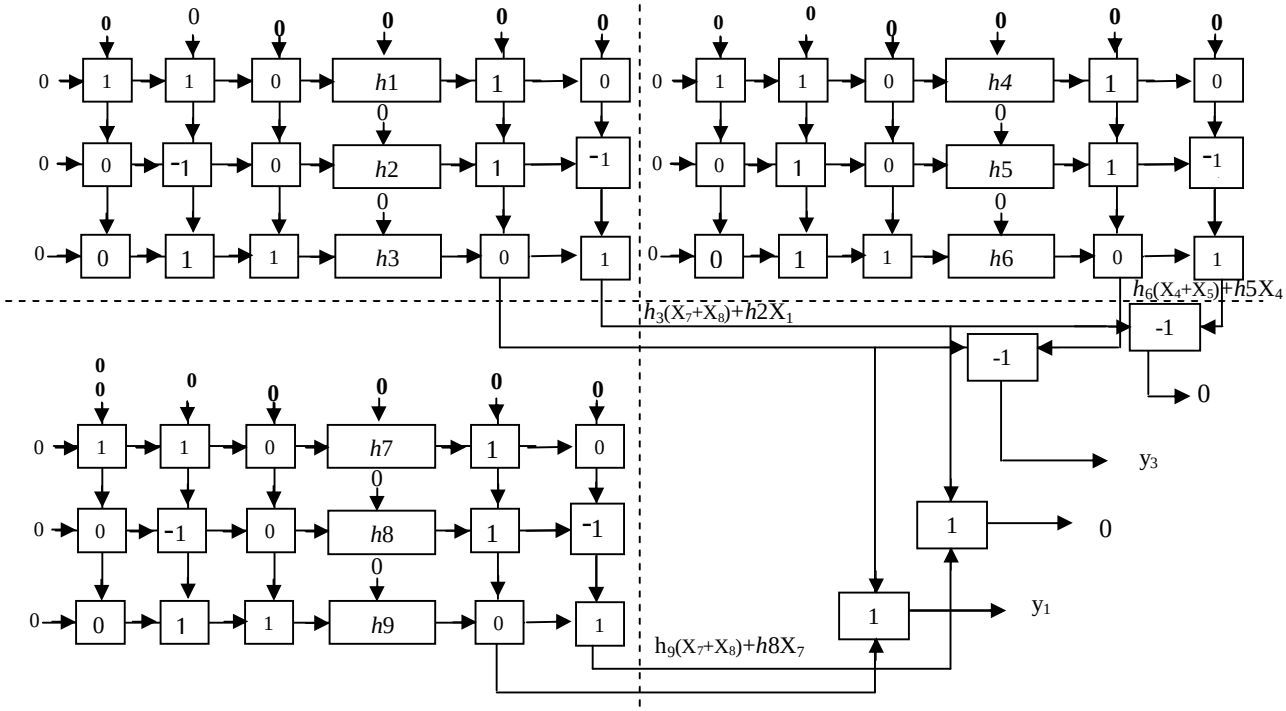


(f) étape 6

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

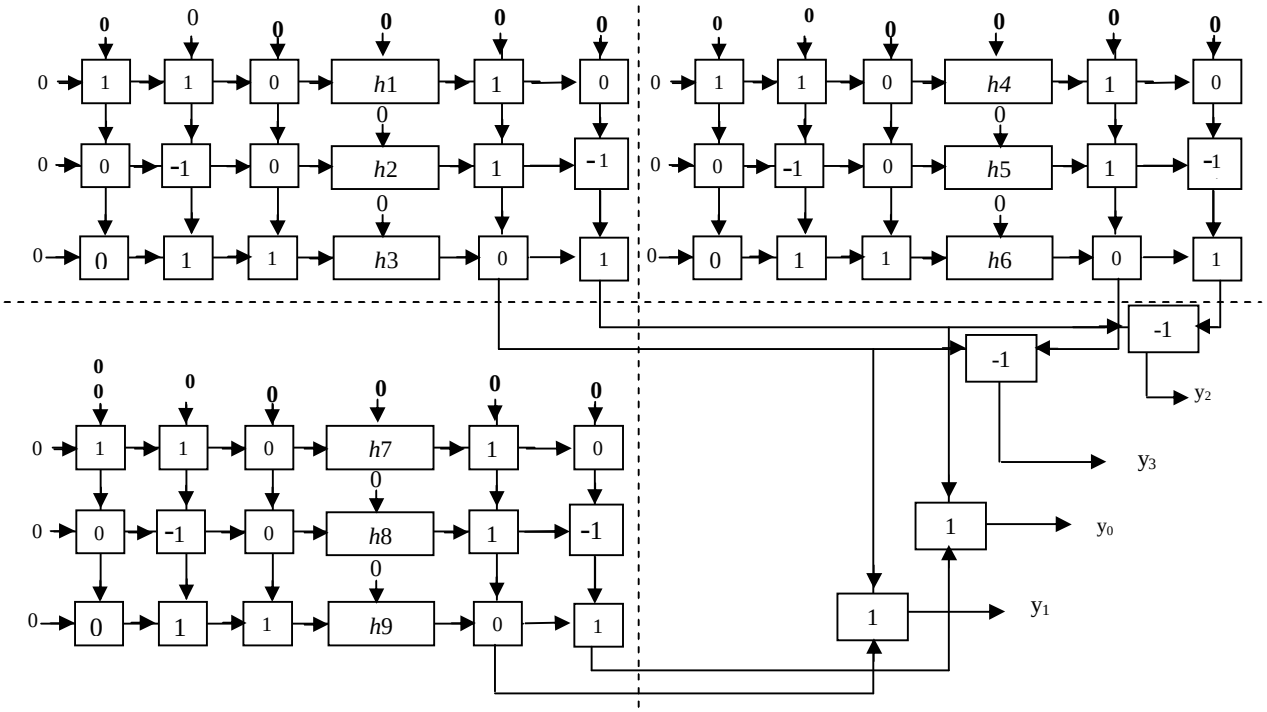


(g) étape 7



(h) étape 8

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



(i) étape 9

Fig. III.16. Principe de fonctionnement de l'architecture systolique de type Kung à partir de la courte convolution itérative

Le temps d'exécution de la figure.III.14 est 9 étapes de calcul (unité de temps).

Une unité de temps = (l + m), l et m étant le nombre de cycles d'horloge pour exécuter une multiplication et une addition respectivement.

Le temps d'exécution pour un réseau systolique de type Kung (cas de $L = N$) pour l'implémentation d'un filtre RIF à partir d'une ISC est donné sur le tableau.III.7.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

L	N_{taps}	Type Kung (t_{kung})
4	4	$9(m + l)$
6	6	$10(m + l)$
8	8	$11(m + l)$
12	12	$13(m + l)$
16	16	$15(m + l)$
18	18	$16(m + l)$
24	24	$19(m + l)$
36	36	$25(m + l)$
48	48	$31(m + l)$
54	54	$34(m + l)$
72	72	$43(m + l)$

Tableau.III.7. Le temps d'exécution pour un réseau systolique de type Kung (cas de $L = N$) pour l'implémentation d'un filtre RIF à partir d'une ISC

III.8. Implémentation du filtre RIF 4-taps et 4-parallèle représenter par ISC sur réseau systolique de type cylindrique

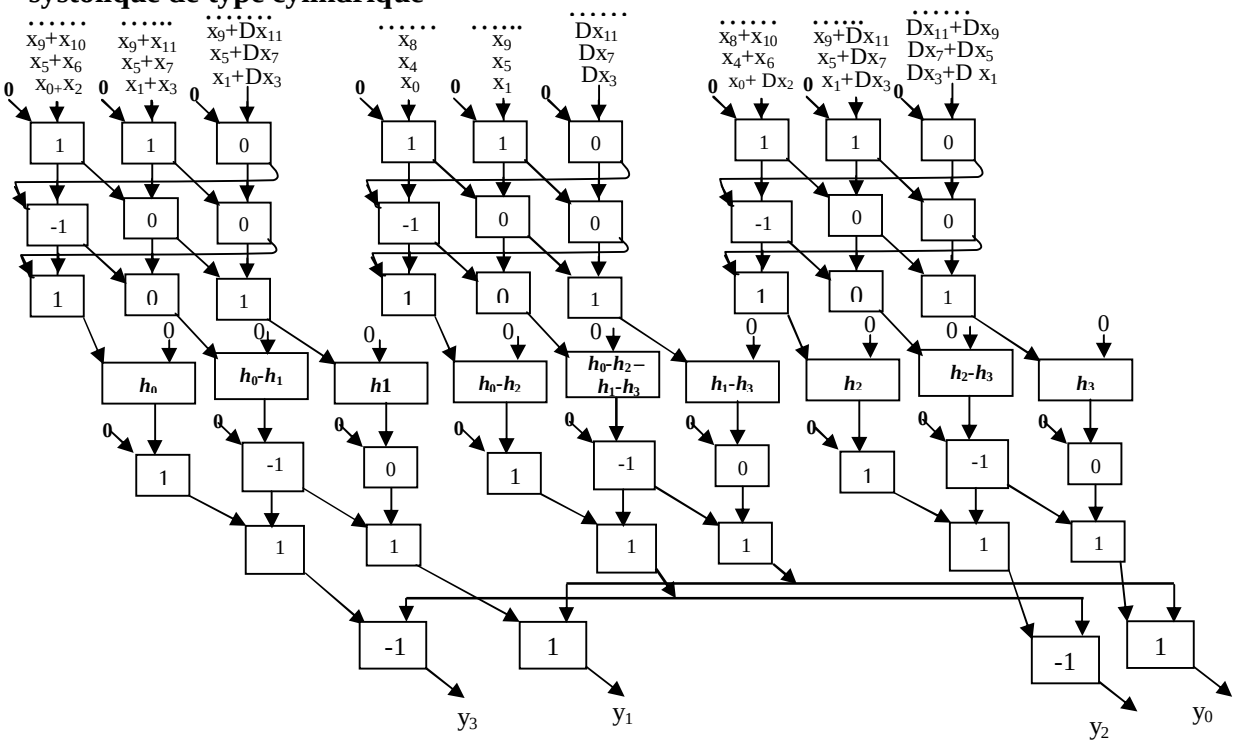


Fig. III.17. Implémentation d'un filtre RIF 4-parallèle sur un réseau systolique de type cylindrique.

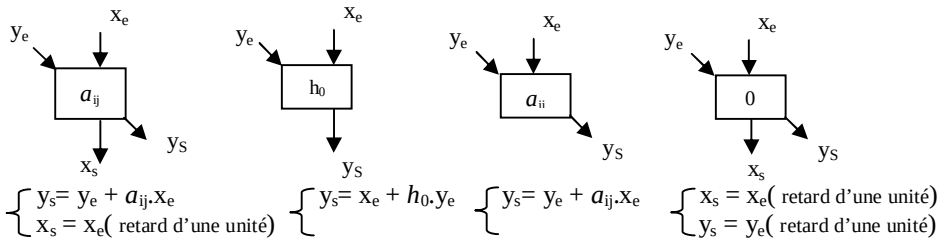
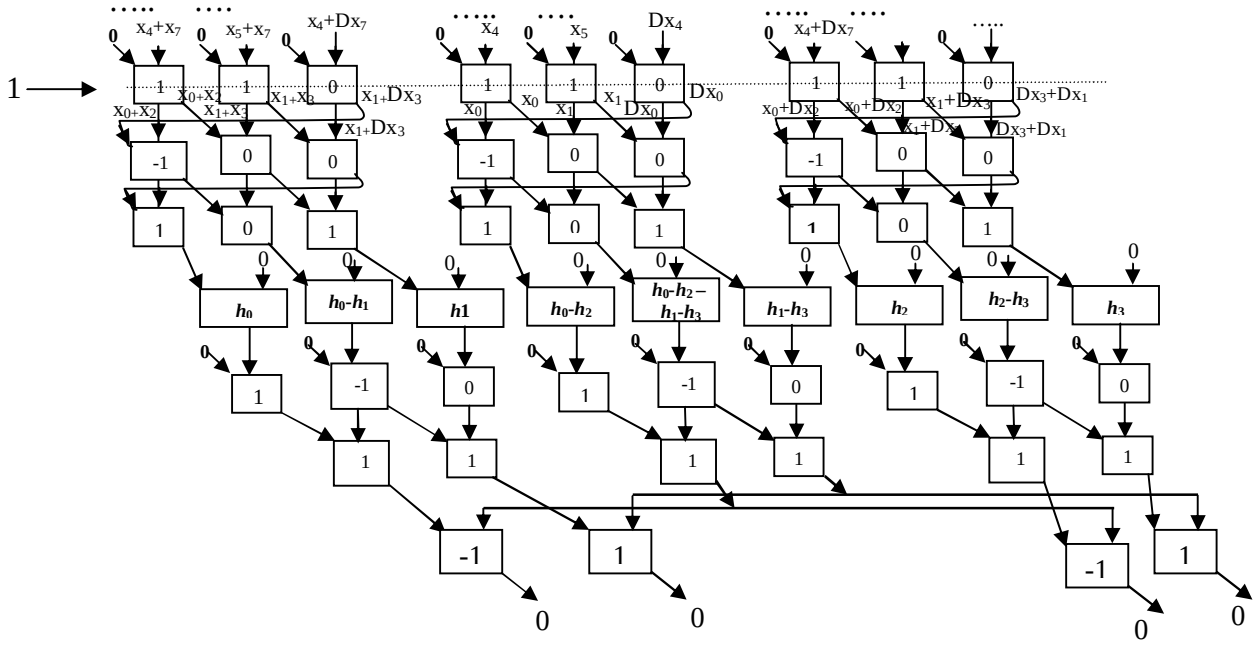


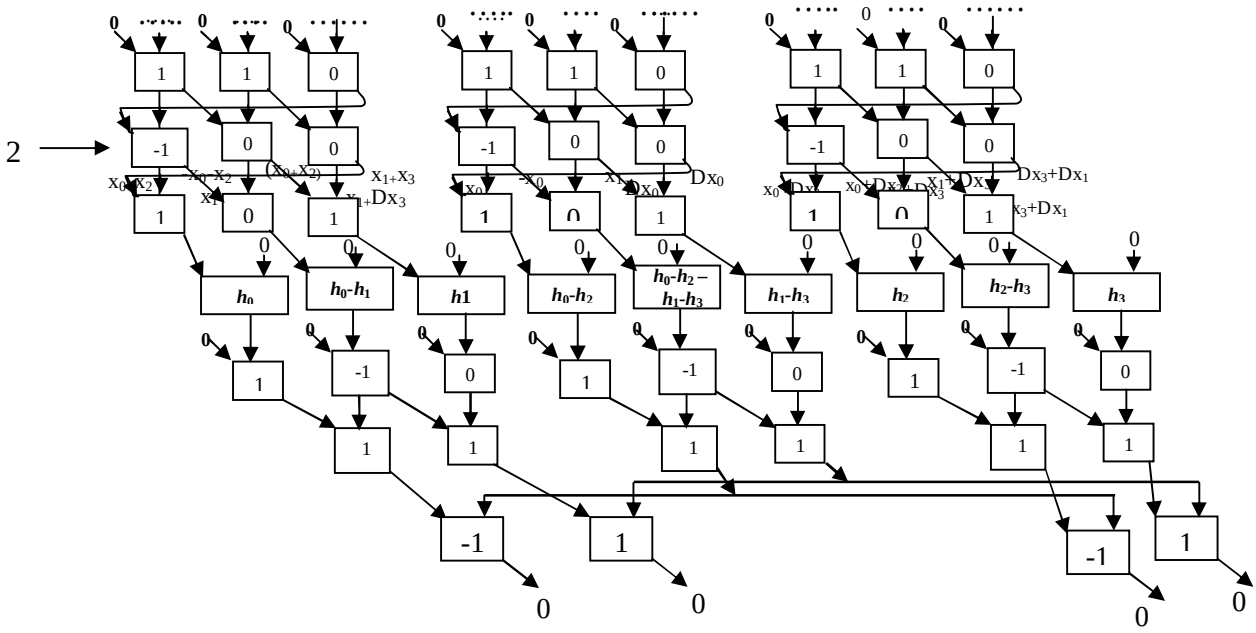
Fig. III.18. Fonctionnement des cellules du réseau systolique de type cylindrique.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Le Principe de fonctionnement de réseau systolique de type cylindrique à partir de la courte convolution itérative et donné par la Figure.III.19.

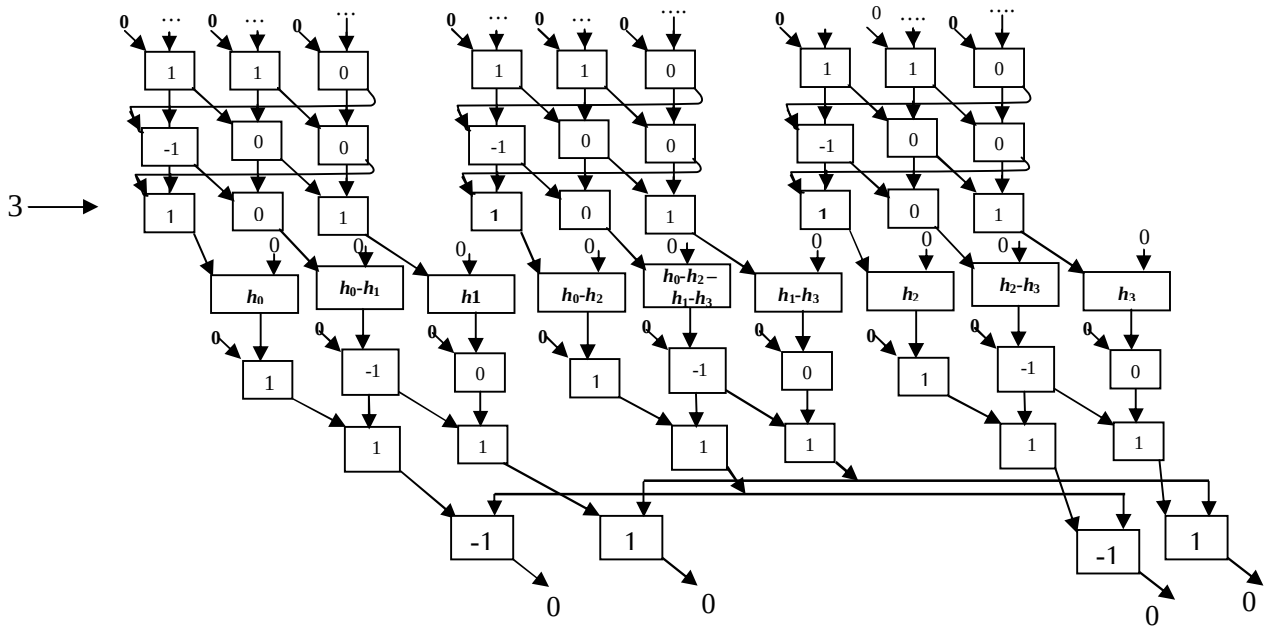


(a) étape 1

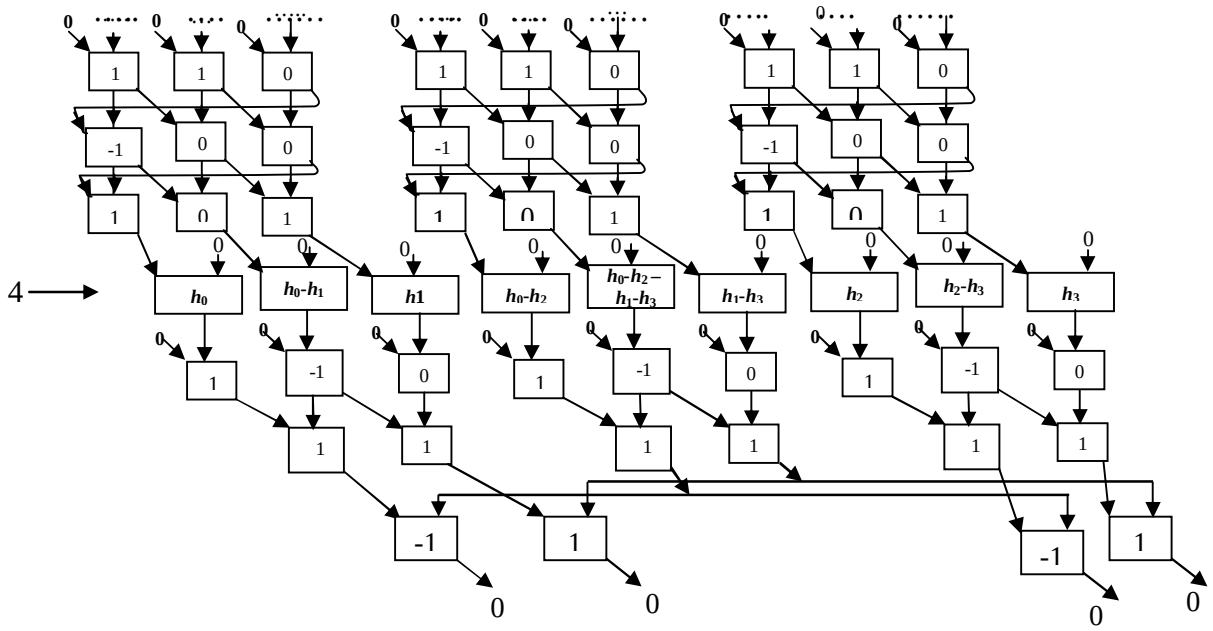


(b) étape 2

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

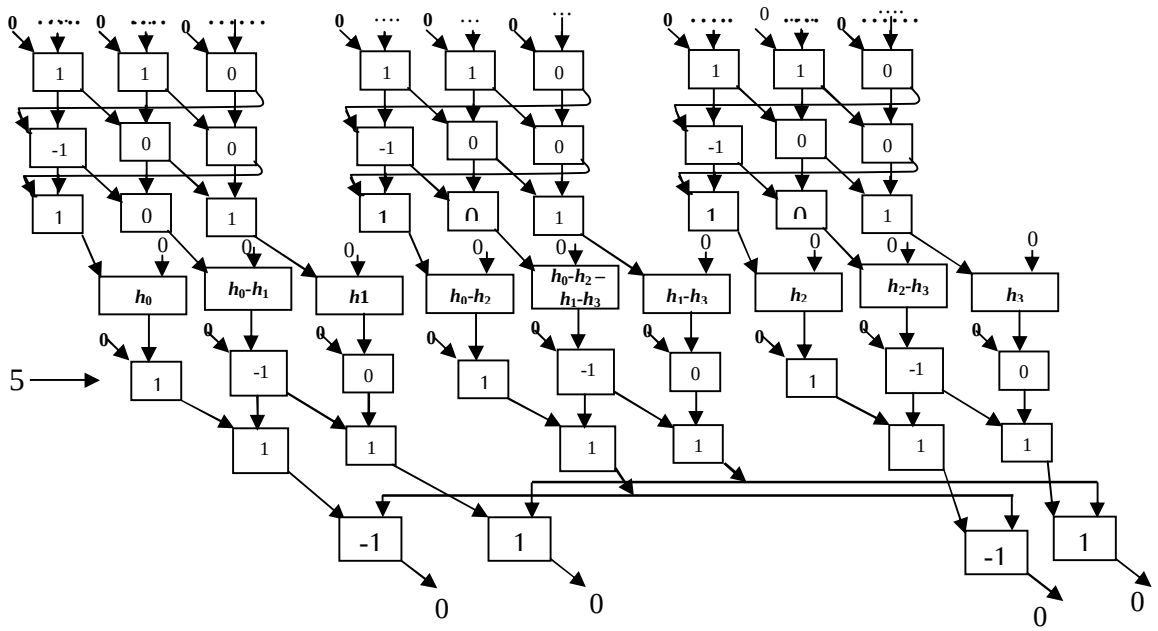


(b) étape 3

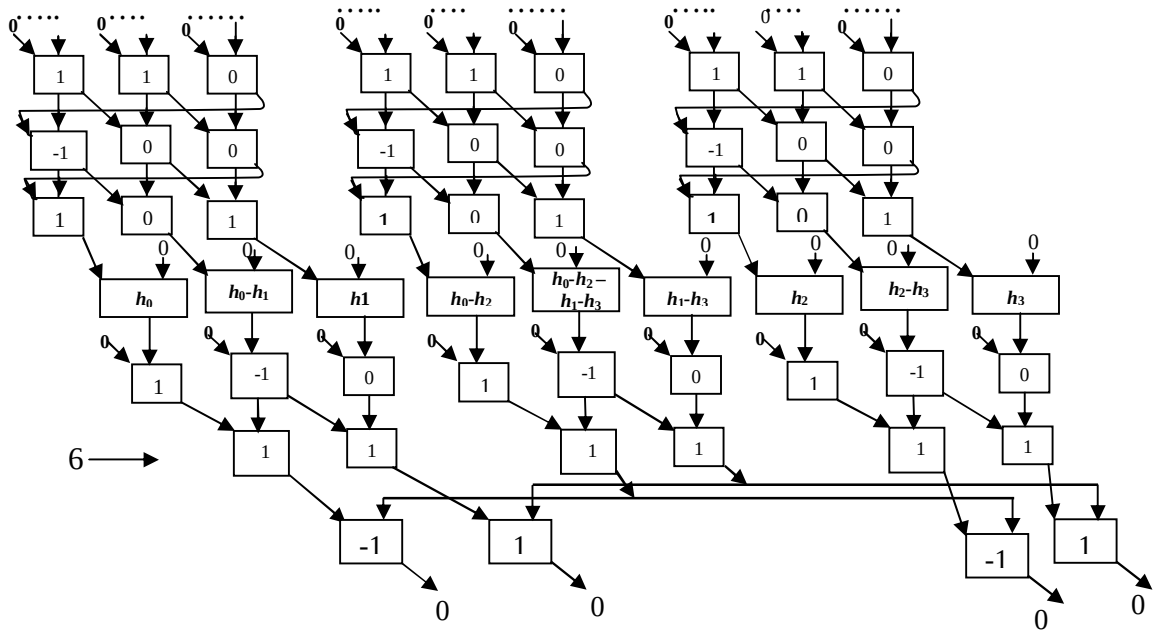


(c) étape 4

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

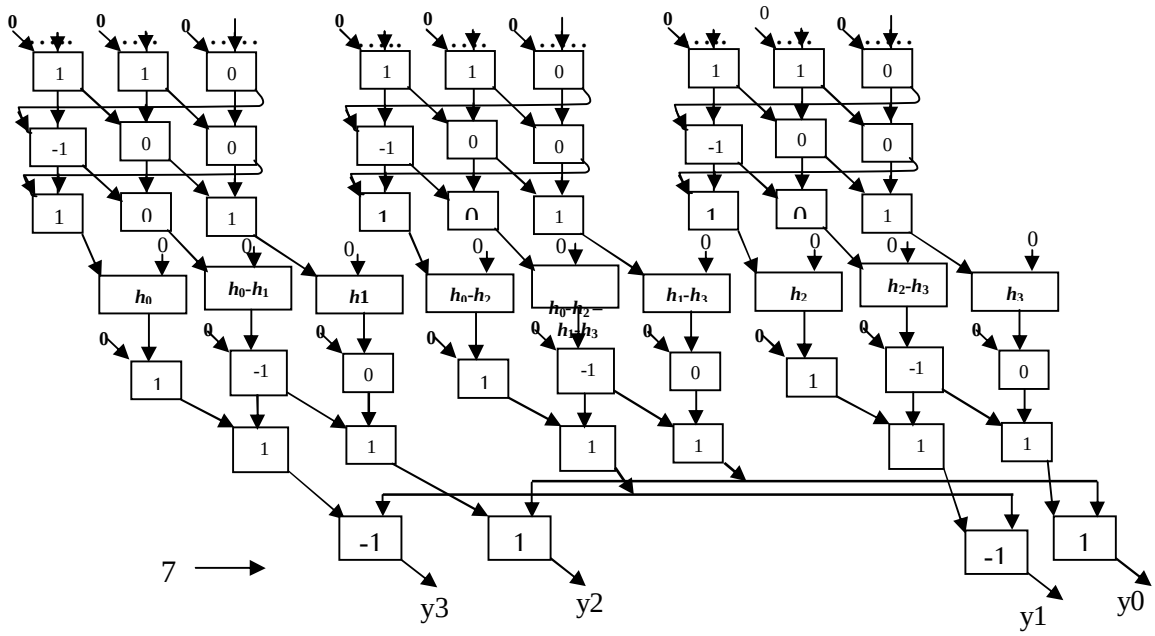


(e) étape 5



(f) étape 6

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



(g) étape 7

Fig. III.19. Principe de fonctionnement de réseau systolique de type cylindrique à partir de la courte convolution itérative.

- Le temps d'éxecution (type cylindrique) Pour 4-Ntaps :

$$t_{cyl} = 7\ (m + l)$$

- A l'initialisation du réseau le temps de calcul : $t_{cyl} = 7(m + l)$

Pour L = N-taps : $t_{4\text{-taps-cyl}} = 7(m + l)$

- Pour fonctionnement permanent une seul étape de calcul: $t_{cyl} = (m + l)$

Le temps d'exécution pour un réseau de type cylindrique est donnée sur le tableau. III.9.

L	N_{taps}	**type cylindrique (t_{Cyl})**
4	**4**	$7(m + l)$
6	**6**	$7(m + l)$
8	**8**	$7(m + l)$
12	**12**	$7(m + l)$
16	**16**	$7(m + l)$
18	**18**	$7(m + l)$
24	**24**	$7(m + l)$
36	**36**	$7(m + l)$
48	**48**	$7(m + l)$
54	**54**	$7(m + l)$
72	**72**	$7(m + l)$

Tableau.III.8. Le temps d'exécution pour un réseau systolique de type cylindrique (cas de *L = N*) pour l'implémentation d'un filtre RIF à partir d'une ISC.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

III.9. Simulation les réseaux systolique issue de la (ISC)

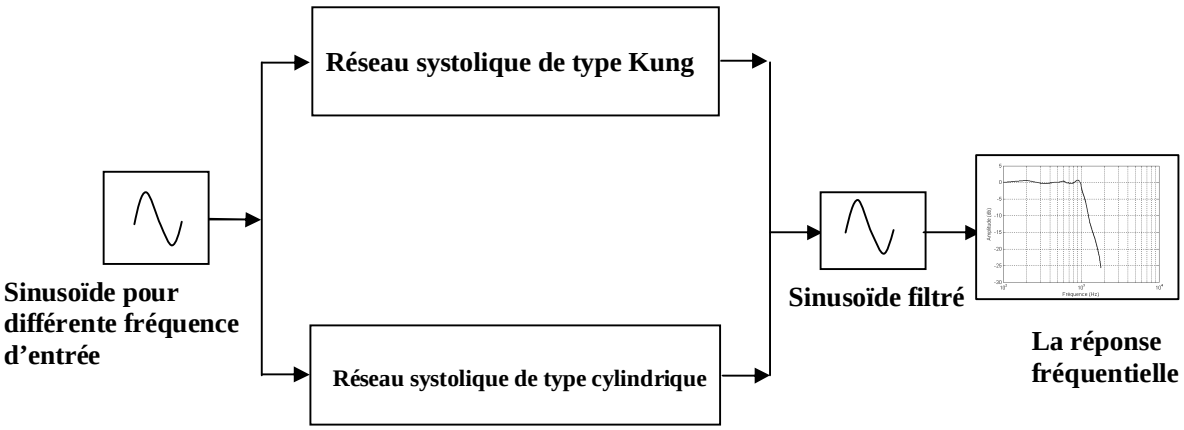


Fig.III.20. Implémentation de l'algorithme de filtrage (ISC) pour différents types d'architecture systolique.

a) Cas d'un filtre passe-bas pour L = 4, N = 24-taps

Considérons l'implémentation d'un filtre passe-bas ayant les caractéristiques suivantes :

f_1 = 1000Hz : fréquence de coupure.

f_2 = 1500Hz : limite de la bande atténuée.

f_e = 5000Hz : fréquence d'échantillonnage.

Les cœfficients du filtre sont calculés par Matlab pour différentes méthodes de synthèse (méthode de remez, fenêtrage, moindres carrées).

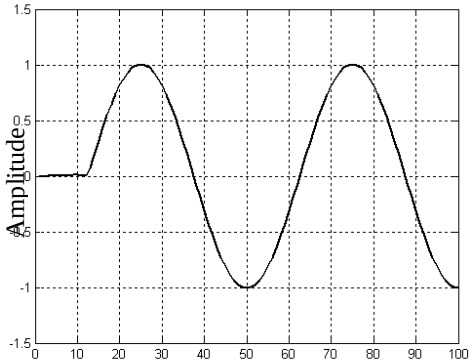
La liste des cœfficients des filtres passe-bas synthétisées est donnée sur le tableau. III.9.

Méthode de synthèse	Méthode de Remez	Méthodes des Moindres carrées
Les coefficients pour $N = 4$ (Fig.III.16)	$h_0 = h3 = 0.2461$, $h1 = h_2 = 0.5435$,	
Les coefficients pour $N = 8$ (Fig.III.17)	$h_0 = h_7 = 0.0084$, $h_2 = h_5 = 0.0609$, $h_3 = h_4 = 0.5133$, $h_1 = h_6 = -0.1504$,	$h_0 = h_7$= -0.0479, $h_1 = h_6$=-0.0281, $h_2 = h_5$=0.1716, $h_3 = h_4$=0.4094. **Fig.III.17.)**
Les coefficients pour $N = 24$ (Fig.III.18)	$h0 = h23 = 0.0028$, $h2 = h21 = 0.0168$, $h4 = h19 = -0.0248$, $h6 = h17 = 0.0420$, $h15 = h8 = -0.0729$, $h13 = h10 = 0.1662$, $h12 = h11 = 0.4273$, $h9 = h14 = -0.0622$, $h7 = h16 = 0.0196$, $h5 = h18 = -0.0039$, $h3 = h20 = -0.0021$, $h22 = h1 = 0.0125$,	

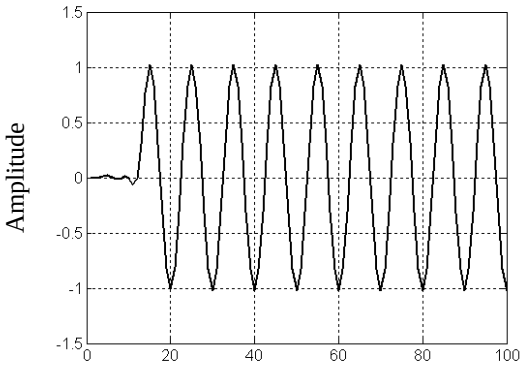
Tableau.III.9. Réponse impulsionnelle d'un filtre passe-bas pour N = 4, 8,24-taps synthétisé par la méthode de Remez et moindres carrées.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

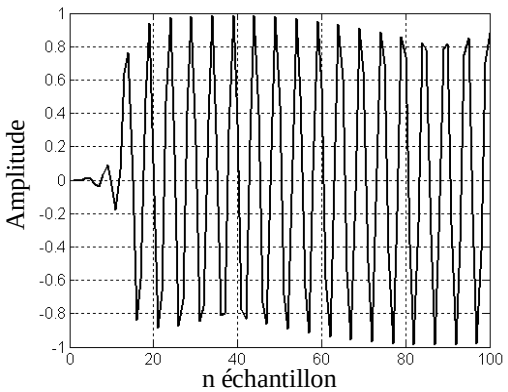
La figure.III.21 illustre les résultats de simulation en représentant la réponse du filtre à la sinusoïde d'entrées pour les fréquences : f = 100, 500, 990, 1000, 1100, 1200, 1300, 1400, 1500, 1600 Hz.



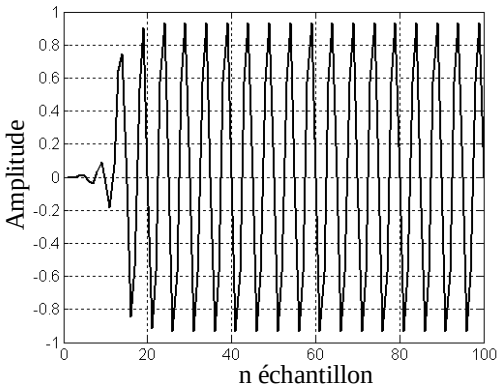
(a) f = 100 Hz



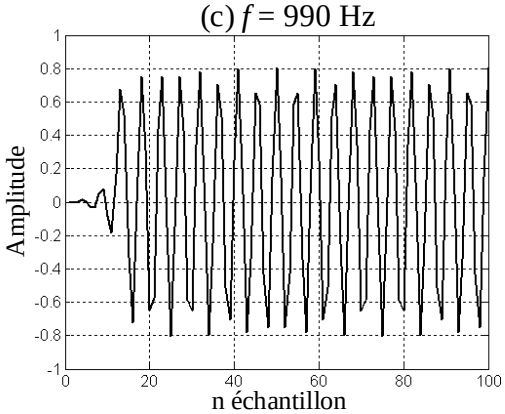
(b) f = 500 Hz



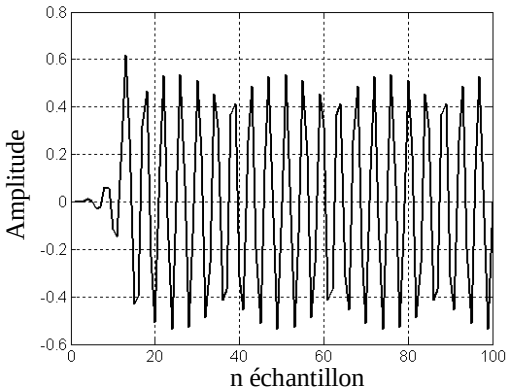
(c) f = 990 Hz



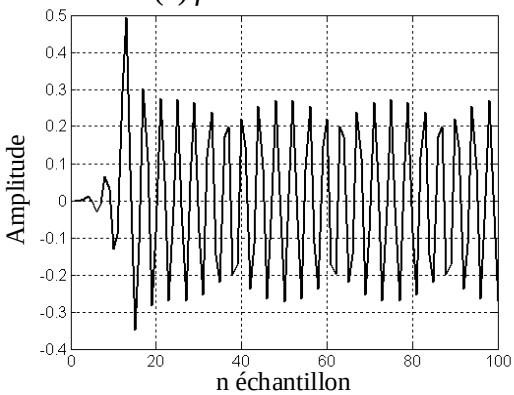
(d) f = 1000 Hz



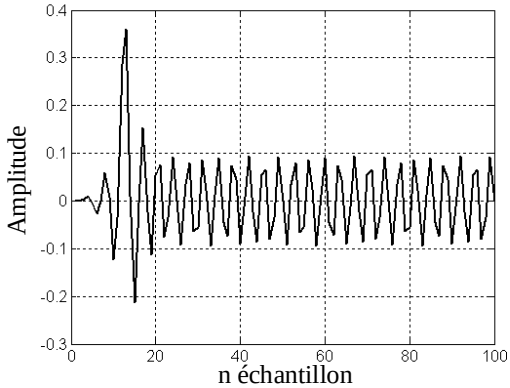
(e) f = 1100 Hz



(f) f = 1200 Hz

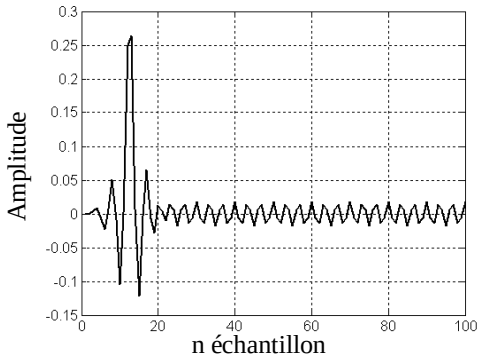


(g) f =1300 Hz

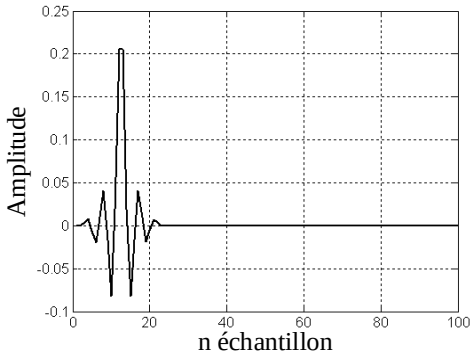


(h) f = 1400 Hz

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



(i) f = 1500 Hz



(k) f = 1600 Hz

Fig. III.21. Filtrage d'une sinusoïde par l'architecture systolique d'un filtre passe-bas RIF 4-parallèl à 24-taps.

Le filtrage d'une sinusoïde par les architectures systoliques d'un filtre RIF 4-parallèle à 24-taps est illustré par les différentes valeurs de l'amplitude du signal de sortie représentées sur le tableau.III.10.

***f*(Hz)**	400	800	900	1000	1100	1200	1300	1400	1500	1600	1700	1900	2000
A(f)	1.0087	0.9924	1.0217	0.9303	0.8056	0.6182	0.4925	0.3600	0.2632	0.2052	0.1684	0.1144	0.0921
A_{dB}(f)	0.075	-0.06	0.18	-0.62	-1.87	-4.17	-6.15	-8.87	-11.59	-13.75	-15.47	-17.12	-20.71

Tableau.III.10. Amplitude du signal filtré par l'architecture systolique d'un filtre RIF 4-parallèle pour N = 24-taps.

La réponse fréquentielle de l'architecture systolique d'un filtre passe-bas RIF 4-parallèle à 24-taps et donnée par la figures .III.22 ($f_{coupure}$ = 1148Hz), et pour N = 64-taps (figure.III.23) ($f_{coupure}$ = 1053Hz), pour différents valeur de N en utilisant les méthodes de synthèse (Remez et moindres carrées).

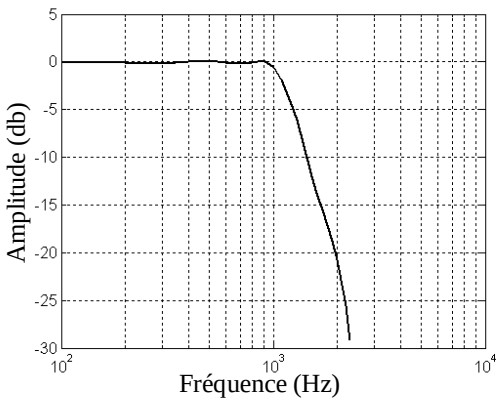


Fig.III.22. Réponse fréquentielle d'un filtre passe-bas RIF 4-parallèl à 24-taps (méthode de Remez).

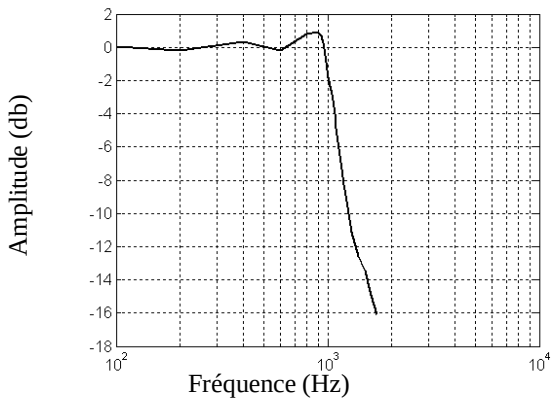


Fig.III.23. Réponse fréquentielle d'un filtre passe-bas RIF 4-parallèl à 64-taps (f_1=1000Hz, f_2 = 1200Hz, f_e = 5000Hz (méthode de Remez)).

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

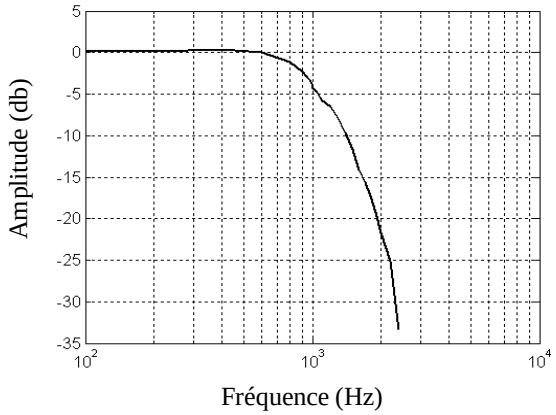


Fig.III.24. Réponse fréquentielle d'un filtre passe-bas RIF 4-parallèl à 8-taps (méthode des moindres carrées).

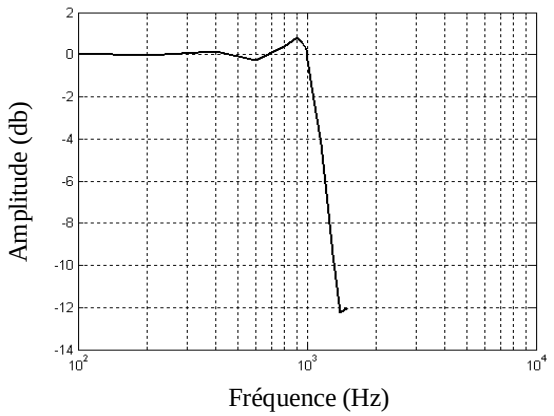


Fig.III.25. Réponse fréquentielle d'un filtre passe-bas RIF 4-parallèl à 64-taps (méthode des moindres carrées).

b) Cas d'un filtre passe-haut

Considérons maintenant l'implémentation systolique d'un filtre passe-haut ayant les caractéristiques :

f_1 =1000Hz : fréquence de coupure.

f_2 = 1500Hz : limite de la bande atténuée.

f_e = 5000Hz : fréquence d'échantillonnage , N =24-taps (Figure.III.33).

La liste des cœfficients de filtre passe-haut synthétisé est donnée sur le tableau. III.11.

Méthode de synthèse	Méthode de Remez
Les coefficients pour **N = 4** (Fig.III.33)	h_0 =0.002365450051559, h_1 =– 0.001080828889070 h_1 =– 0.012533256616747 , h_3 =– 0.023647030401219 h_4 = 0.00242292714277, h_5= 0.031443097744307 h_6= – 0.010373191550343, h_7= –0.054523435346591 h_8=0.035454961230119, h_9= 0.102061400679954 h_{10}= – 0.124989868987548, h_{11}= 0.469378882750645 h_{12} = – 0.469378882750645, h_{13} =– 0.124989868987548 h_{14} =0.102061400679954, h_{15} = 0.035454961230119 h_{16}= – 0.054523435346591 , h_{17}=– 0.010373191550343 h_{18}=0.031443097744307, h_{19}= 0.002422927142773 h_{20}=0.023647030401219, h_{21}=– 0.012533256616747 h_{22}= – 0.001080828889070, h_{23}=0.002365450051559

Tableau.III.11. Réponse impulsionnelle d'un filtre RIF passe-haut pour N = 24-taps synthétisé par la méthode de Remez.

La figure.III.26 représente la réponse fréquentielle du filtre passe-haut RIF 4-parallèle à 24-taps à partir de la simulation sur l'architecture systolique.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

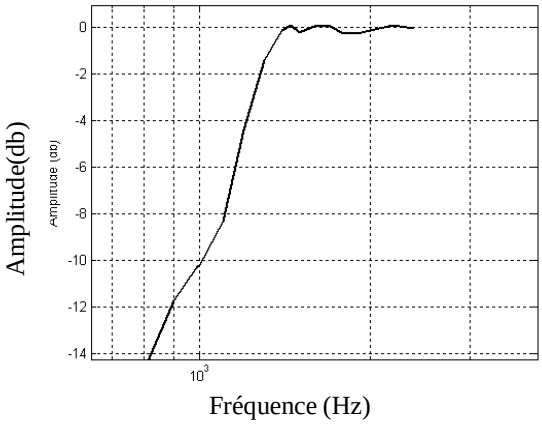


Fig.III.26. Réponse fréquentielle d'un filtre passe-haut RIF 4-parallèl à 24-taps implémenté sur réseau systolique (méthode de Remez).

c) Cas d'un filtre passe-bande

Le filtre passe bande à implémenter est défini par :

f_1 = 700 Hz: limite de la bande atténuée.

f_2 = 760 Hz : 1ère fréquence de coupure.

f_3 = 1200 Hz : 2ème fréquence de coupure.

f_4 = 1260 Hz : limite de la bande atténuée.

f_e = 3500 Hz: fréquence d'échantillonnage.

La figure.III.27 représente la réponse fréquentielle du filtre passe-bande RIF à 144-taps implémenté sur réseau systolique par la méthode de Remez.

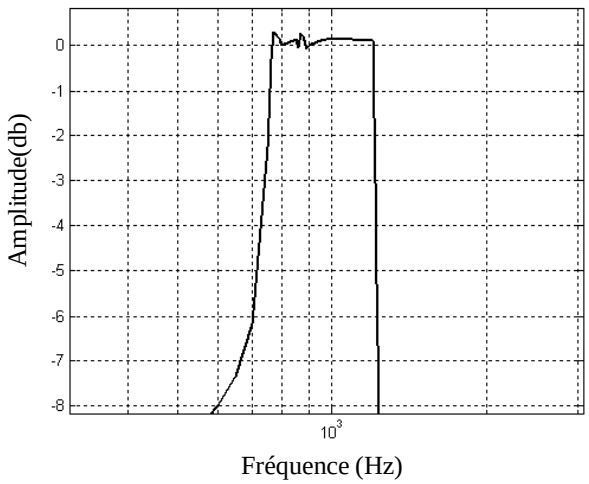


Fig.III.27. Réponse fréquentielle d'un filtre passe bande RIF 4-parallèl à 144-taps implémenté sur réseau systolique(méthode de Remez).

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

d) Cas d'un filtre passe-bas pour L = 6 à 24-taps

Considérons l'implémentation d'un filtre passe-bas ayant les caractéristiques suivantes :

f_1 = 1000Hz : fréquence de coupure.

f_2 = 1500Hz : limite de la bande atténuée.

f_e = 4000Hz : fréquence d'échantillonnage.

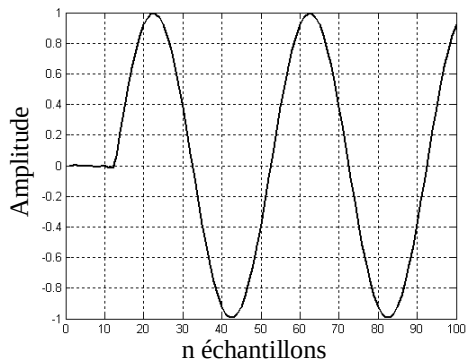
Les cœfficients du filtre sont calculés par Matlab par différentes méthodes de synthèse (méthode de Remez, moindres carrées) (tableau.III.12).

Méthode de synthèse	Méthode de Remez	Méthodes des Moindres carrées
Les coefficients pour N = 6 (Fig.III.36)	$h_0 = h_5 = -0.1498$, $h_4 = h_1 = 0.0617$, $h_2 = h_3 = 0.5134$,	
Les coefficients pour N = 12 (Fig.III.37)	$h_0 = h_{11} = 0.0306$, $h_1 = h_{10} = 0.0802$, $h_2 = h_9 = 0.0063$, $h_3 = h_8 = -0.1160$, $h_4 = h_7 = 0.0843$, $h_5 = h_6 = 0.4993$,	
Les coefficients pour N = 24(Fig.III.38)	$h_0 = h_{23} = 0.0025$, $h_{22} = h_1 = 0.0004$ $h_2 = h_{21} = -0.0112$, $h_3 = h_{20} = -0.0022$, $h_4 = h_{19} = 0.0187$, $h_5 = h_{18} = -0.0125$ $h_{17} = h_6 = -0.0290$, $h_{16} = h_7 = 0.0460$, $h_8 = h_{15} = 0.0199$, $h_{14} = h_9 = -0.1174$ $h_{10} = h_{13} = 0.0676$, $h_{11} = h_{12} = 0.5109$,	$h_0 = h_{23} = 0.0019$, $h_1 = h_{22} = 0.0026$, $h_2 = h_{21} = -0.0064$, $h_{20} = h_3 = -0.0026$ $h_{19} = h_4 = 0.0173$, $h_{18} = h_5 = -0.0079$, (Fig.III.39) $h_{17} = h_6 = -0.0297$, $h_{16} = h_7 = 0.0415$, $h_8 = h_{15} = 0.0248$, $h_{14} = h_9 = -0.1160$, $h_{10} = h_{13} = 0.0612$, $h_{11} = h_{12} = 0.5146$,

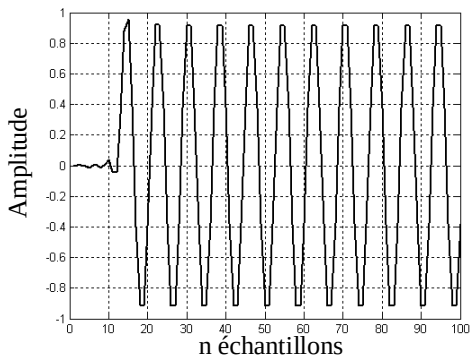
Tableau.III.12. Réponse impulsionnelle d'un filtre passe bas pour N = 6, 12, 24-taps par Remez et moindres carrées.

Le filtrage d'un sinusoïde sur l'architecture systolique du filtre passe-bas RIF 6-parallel à 24-taps est représenté sur la figure.III.28 pour différentes fréquences d'entrée f = 100, 500 , 990, 1000, 1100 , 1200, 1300 , 1400, 1500 , 1600 Hz.

f_e = 3000Hz : fréquence d'échantillonnage.

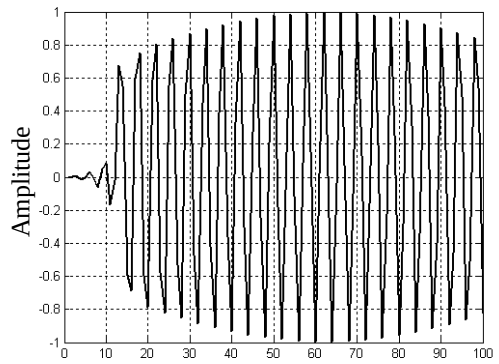


(a) f = 100 Hz

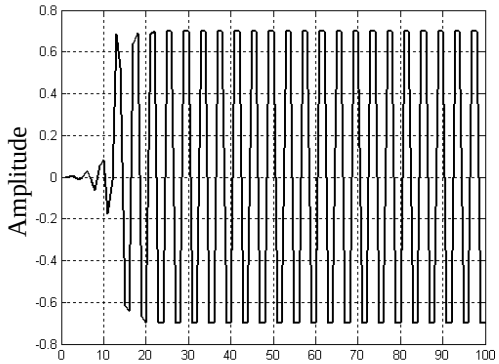


(b) f = 500 Hz

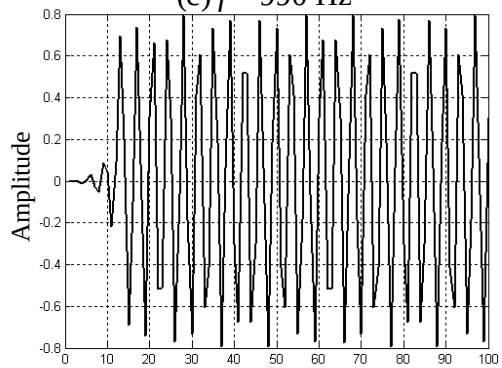
This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!



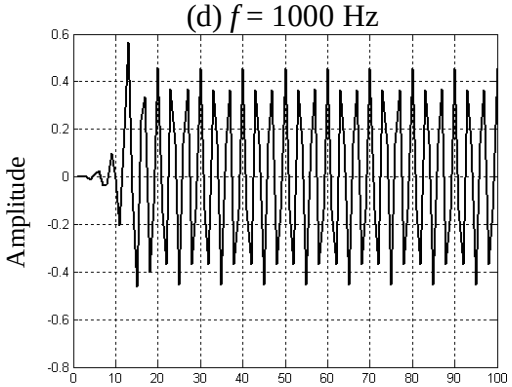
(c) f = 990 Hz



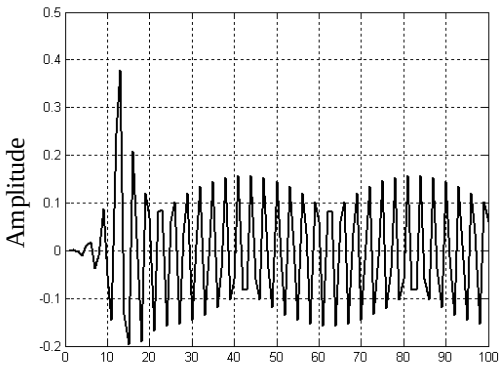
(d) f = 1000 Hz



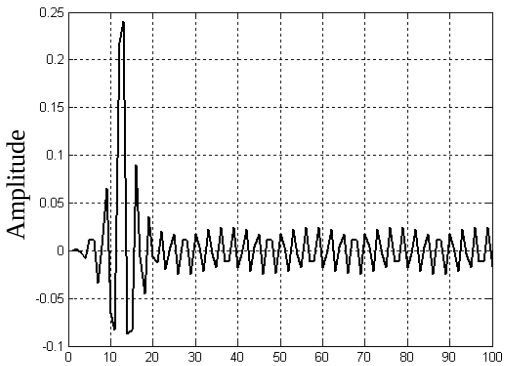
(e) f = 1100 Hz



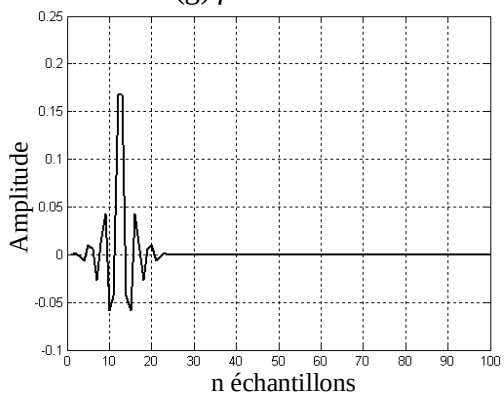
(f) f = 1200 Hz



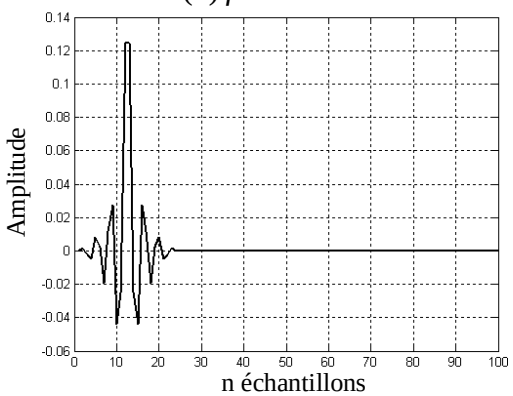
(g) f =1300 Hz



(h) f = 1400 Hz



(i) f = 1500 Hz



(k) f = 1600 Hz

Fig. III.28. Filtrage d'une sinusoïde sur l'architecture systolique du filtre passe-bas d'un RIF 6-parallèle à 24-taps.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Le filtrage d'une sinusoïdale sur l'architecture systolique d'un filtre RIF 6-parallèle à 6, 12,24-taps est représenté par les différentes valeurs de l'amplitude du signal de sortie est reporté sur le tableau.III.13.

***f*(Hz)**	400	800	900	1000	1100	1200	1300	1400	1500	1600	1700	1900	2000
A_{dB}(f)	0.416	0.330	-0.008	-3.565	-3.424	-5.350	-7.310	-10.16	-14.18	-14.95	-16.52	-24.04	-148.6

(a)

***f*(Hz)**	400	800	900	1000	1100	1200	1300	1400	1500	1600	1700	1900	2000
A_{dB}(f)	-0.540	0.880	0.4265	-3.152	-4.347	-6.549	-9.505	-14.78	-15.88	-18.49	-21.40	-31.20	-165.2

(b)

***f*(Hz)**	400	800	900	1000	1100	1200	1300	1400	1500	1600	1700	1900	2000
A_{dB}(f)	0.178	-0.338	0.069	-3.111	-2.031	-4.935	-8.427	-12.93	-15.50	-18.10	-21.02	-31.00	-165.7

(c)

Tableau .III.13. Amplitude du Signal filtre par l'architecture systolique d'un filtre passe-bas RIF 6-parallèle (a) 6-taps (b) 12-taps (c) 24-taps

La réponse fréquentielle de l'architecture systolique du filtre passe-bas RIF 6-parallèle est donnée par les figures.III.29 à III.33 pour différents taps en utilisant les méthodes de synthèse (Remez et moindres carrées).

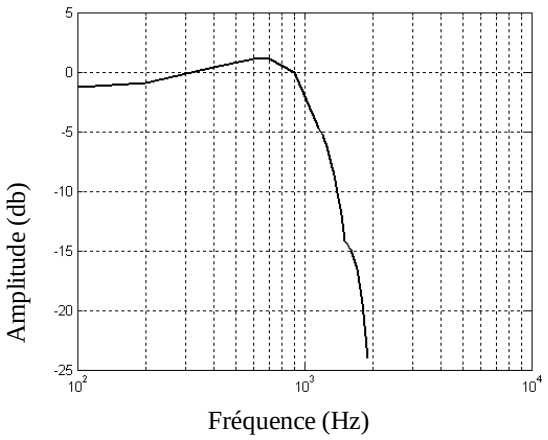


Fig.III.29. Réponse fréquentielle d'un filtre RIF 6-parallèle à 6-taps implémenté sur réseau systolique (méthode de Remez).

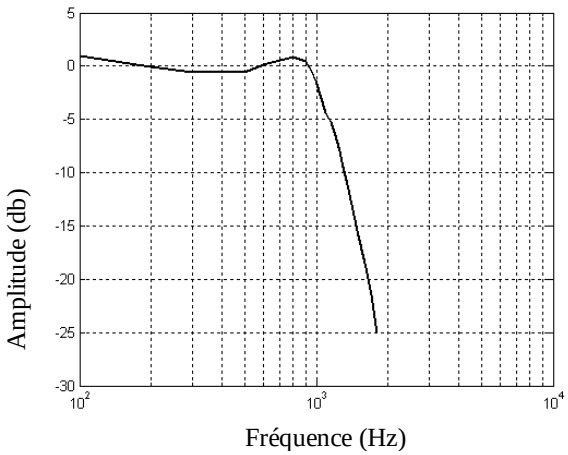


Fig.III.30. Réponse fréquentielle d'un filtre passe- bas RIF 6-parallèle à 12-taps (méthode de Remez).

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

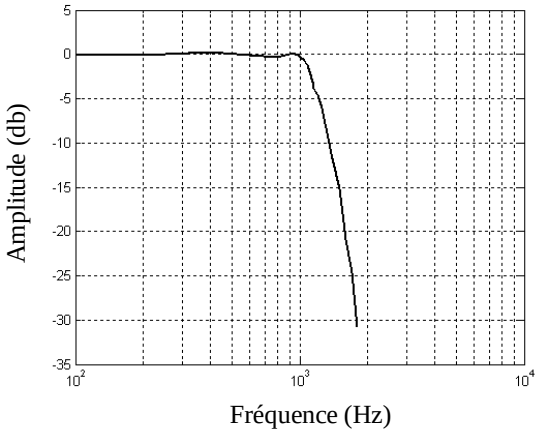


Fig.III.31. Réponse fréquentielle d'un filtre RIF 6-parallèl à 24-taps (méthode de Remez).

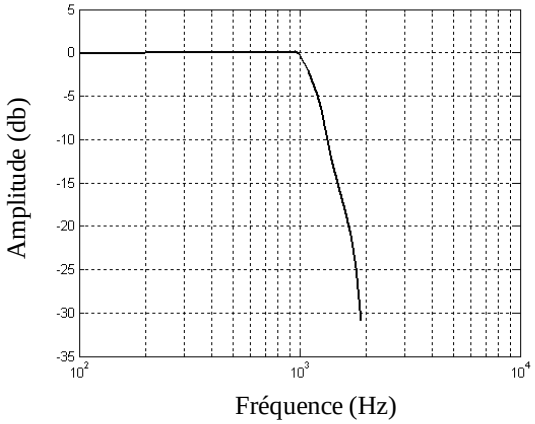


Fig.III.32. Réponse fréquentielle d'un filtre RIF 6-parallèl à 24-taps (moindre carrées).

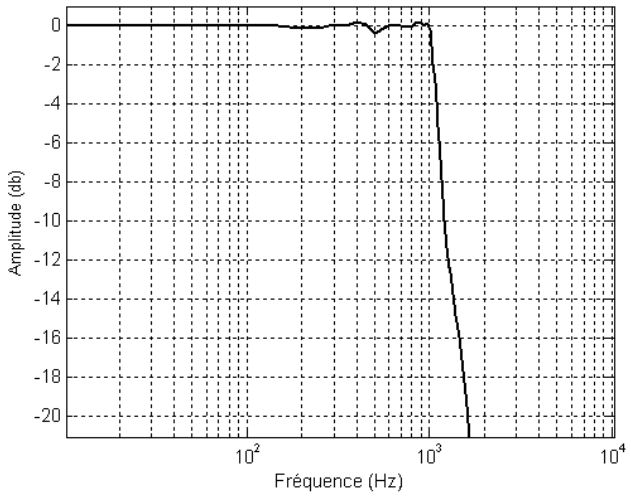


Fig.III.33. Réponse fréquentielle d'un filtre passe-bas RIF 6-parallèl à 72-taps implémenté sur réseau systolique (méthode de moindre carrées)

e) Cas d'un Filtre passe-haut

Le filtre passe-haut RIF à représenter par une courte convolution est défini par :

f_1 = 1000Hz : fréquence de coupure, f_2 = 1500Hz : limite de la bande atténuée,

f_e = 5000Hz : fréquence d'échantillonnage , N =24-taps.

La réponse fréquentielle de l'architecture systolique du filtre RIF pass-haut est illustrée sur la figure.III.34.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

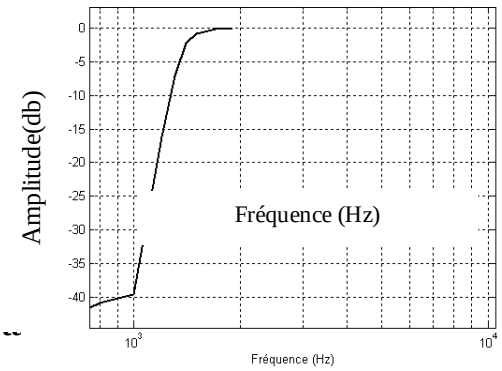


Fig.III.3 -haut RIF 6-parallèl à 24-.. éthode de Remez)

f) Cas d'un filtre passe-bande

Le filtre passe bande à concevoir est défini par :

f_1 = 700 Hz, f_2 = 760 Hz, f_3 = 1200 Hz, f_4= 1260 Hz, f_e = 3500 Hz, N_{taps} =72-taps.

La figure.III.35 représente la réponse fréquentielle de l'architecture systolique du filtre RIF passe-bande.

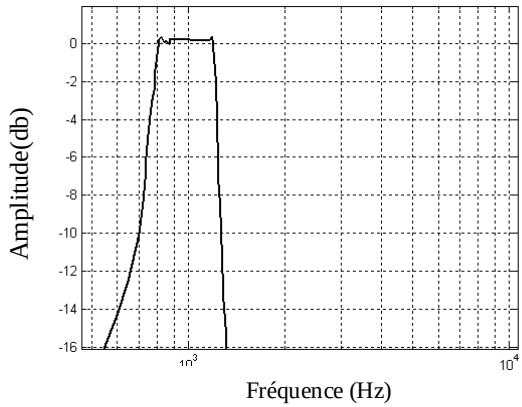


Fig.III.35. Réponse fréquentielle d'un filtre passe-bande RIF 6-parallèl à 24-taps (méthode de Remez).

g) Cas d'un filtre Multi-bande

Le filtre Multi-bande à concevoir est défini par :

f_1 = 200 Hz, f_2 = 250 Hz, f_3 = 400 Hz, f_4 = 450 Hz, f_5 = 600 Hz, f_6 = 850 Hz, f_7 = 900 Hz, f_8 = 1000 Hz, f_e = 3500 Hz , N_{taps} = 72-taps .

($f_{coupure1}$ = 232Hz, $f_{coupure2}$ = 416Hz, $f_{coupure3}$ = 635Hz, $f_{coupure4}$ = 870Hz).

La figure.III.36 illustre la réponse fréquentielle du filtre RIF multi-bande sur l'architecture systolique.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

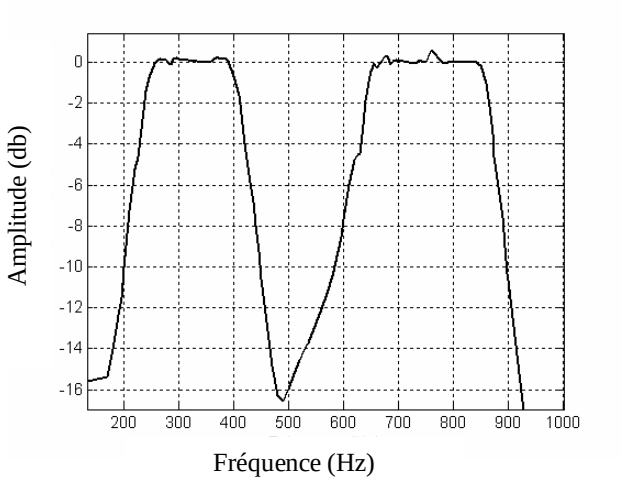


Fig.III.36. Réponse fréquentielle d'un filtre multi-bande RIF 6-parallèl à 72-taps implémenté sur l'architecture systolique (méthode de moindre carrée).

III.10. Résultats et Commentaire

Les résultats de simulation du filtre RIF implémenté sur un réseau systolique montre que le gabarit du filtre réalisé est en conformité avec celui du filtre désiré pour les méthodes de synthèse utilisées Remez et les moindres carrées. Lorsque le nombre de taps de la courte convolution du filtre RIF augmente, le nombre d'éléments processeurs augmente résultant en une architecture plus complexe.

a) Dans le cas de l'implémentation directe de la convolution ordinaire sur réseau systolique :

Pour L = N, lorsque L prend des valeurs importantes, le temps de calcul sur un réseau cylindrique est presque la moitié du temps de calcul sur un réseau systolique de type Kung pour la même complexité matérielle (tableaux.III.9).

b) Dans le cas de l'implémentation de la courte convolution itérative sur réseau systolique :

Pour L = N, lorsque L augmente, le temps de calcul sur un réseau cylindrique garde la même valeur $7(m + l)$ alors qu'il augmente sur réseau de type Kung pour la même complexité matérielle (tableaux.III.8)

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Une étude comparative entre la technique de Parhi (implémentation par technologie VLSI), et résumée sur le tableau.III.14

Conv	Structure d'implémentation		M		A		D		Temps de calcul		
										Phase transitoire	phase. permanent
4 X 4	SYS	2x2, 2x2	355	-23 %	355	-15%	333	-18%	t_{Cyl}	$42(m+l)$	$(m+l)$
									t_{kung}	$44(m+l)$	
	PAR	2x2, 2x2	288		308		283		t_{PAR}	$36l+42m$	
6 x 6	SYS	3x3, 2x2	492	-37%	492	-21%	450	-29%	t_{Cyl}	$30(m+l)$	$(m+l)$
									t_{kung}	$33(m+l)$	
	PAR	2x2, 3x3	360		406		350		t_{PAR}	$24l+37m$	
8 x 8	SYS	2x2,2x2,2x2	575	-33%	575	-13%	513	-24%	t_{Cyl}	$24(m+l)$	$(m+l)$
									t_{kung}	$28(m+l)$	
	PAR	4x4, 2x2	432		509		415		t_{PAR}	$18l+44m$	
12 x 12	SYS	3x3,2x2,2x2	822	-52%	822	-16%	702	-39%	t_{Cyl}	$18(m+l)$	$(m+l)$
									t_{kung}	$24(m+l)$	
	PAR	4x4, 3x3	540		706		506		t_{PAR}	$12l+37m$	
16 x 16	SYS	2x2,2x2,2x2, 2x2	988	-71%	988	-7%	810	-37%	t_{Cyl}	$15(m+l)$	$(m+l)$
									t_{kung}	$23(m+l)$	
	PAR	4x4, 4x4	576		926		591		t_{PAR}	$9l+46m$	
18 x 18	SYS	3x3,3x3,2x2	1206	-101%	1206	-20%	972	-79%	t_{Cyl}	$14(m+l)$	$(m+l)$
									t_{kung}	$23(m+l)$	
	PAR	3x3,3x3,2x2	600		1007		542		t_{PAR}	$8l+48m$	

Tableau .III.14. Nombre de multiplication (M), additions (A), éléments de retard (D). Temps d'exécution (t_{kung}, t_{Cyl}, t_{PAR}) d'un Filtre RIF *L*- parallèles à 144-Taps. (SYS: réseau systolique d'un filtre RIF issu de la courte convolution itérative) (PAR: structure de Parhi d'un filtre RIF issu de la ISC).

Dans le cas d'une implémentation de la courte convolution itérative sur un réseau systolique et pour un nombre de taps fixe de 144, le temps de calcul sur un réseau cylindrique diminue avec l'augmentation de L (nombre d'entrée parallèles).

Malgré que la structure de Parhi demande moins d'éléments de calcul (additionneurs, multiplieurs, retard) que le réseau systolique, elle reste plus complexe à réaliser en technologie VLSI que les réseaux systolique qui respectent le principe de la localité et la régularité.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

CONCLUSION GENERALE

L'objectif de ce travail était l'étude de la technique d'implémentation des algorithmes de filtrage RIF sur les architectures systoliques et la comparaison de ses performances par rapport à la technique d'implémentation VLSI de Parhi.

Nous avons d'abord utiliser la technique de la courte convolution pour développer des algorithmes de filtrage RIF rapides à partir de la réponse impulsionnelle du filtre obtenue par une des méthodes de synthèse : méthode des fenêtre, méthode d'échantillonnage en fréquence, méthode de Remez ou méthode des moindres carrées.

Nous avons également utiliser Matlab pour implémenter les algorithmes de filtrage RIF à base de la courte convolution sur les structures VLSI parallèles de Parhi. Nous avons simulé et testé le fonctionnement de ces structures à l'aide d'un signal sinusoïdal de fréquence variable. Les résultats de simulation ont montré la bonne concordance avec les spécifications du filtre désiré.

Nous avons procédé aux tests des circuits VLSI de Parhi en faisant appel aux différentes méthodes de synthèse du filtre numérique RIF pour extraire les coefficients du filtre RIF (filtre passe-bas, filtre passe-haut, filtre passe-bande). Ces tests ont révélé la complexité des structures VLSI de Parhi durant la phase d'implémentation. En effet, l'architecture de Parhi devient complexe lorsque la longueur du filtre augmente (nombre de taps) en exigeant un grand nombre de sous filtres. Ainsi, nous avons pensé à exploiter l'architecture systolique pour améliorer la complexité des structures de filtrage RIF.

La phase d'implémentation a été donc modifié en appliquant les réseaux systoliques qui sont connus pour leur simplicité et régularité de contrôle et de communication par rapport à d'autres structures.

Deux approches d'implémentation sont proposées:

Une approche basée sur l'implémentation directe de la convolution ordinaire sur les réseaux systoliques de Kung ou cylindrique.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Et une approche basée sur l'implémentation de la courte convolution itérative sur les réseaux de Kung et cylindrique.

Pour le cas de la courte convolution itérative (L, N) nous avons proposé une architecture parallèle composée d'un certain nombre de réseaux systolique connectés en parallèle, qui calculent simultanément la sortie du filtre RIF désiré.

La structure systolique développée est simple, régulière et vérifie le principe de localité qui la rend plus prête à une implantation VLSI. Le filtre RIF est représenté par un algorithme rapide issu de la technique de la courte convolution.

a) Dans le cas de l'implémentation directe de la convolution ordinaire sur réseau systolique :

Pour $L = N$, lorsque L prend des valeurs importantes, le temps de calcul sur un réseau cylindrique est presque la moitié du temps de calcul sur un réseau systolique de type Kung pour la même complexité matérielle(tableaux.III.9).

b) Dans le cas de l'implémentation de la courte convolution itérative sur réseau systolique pour $L = N$, lorsque L augmente, le temps de calcul sur un réseau cylindrique garde la même valeur $7(m + l)$ alors qu'il augmente sur réseau de type Kung pour la même complexité matérielle.

c) Dans le cas d'une implémentation de la courte convolution itérative sur un réseau systolique et pour un nombre de taps fixe de 144, le temps de calcul sur un réseau cylindrique diminue avec l'augmentation de L (nombre d'entrée parallèles).

Malgré que la structure de Parhi demande moins d'éléments de calcul(additionneurs, multiplieurs, retard) que le réseau systolique, elle reste plus complexe à réaliser en technologie VLSI que les réseaux systolique qui respectent le principe de la localité et la régularité.

Les tests de simulation de l'architecture systolique associée à différentes méthodes de synthèse du filtre numérique RIF démontrent ses bonnes performances en terme de complexité de l'implémentation. Nous pouvons également conclure que l'architecture

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

systolique associée à l'algorithme de filtrage RIF constitue une bonne solution pour réaliser le filtrage numérique en temps réel à moindre coût.

En perspectives, nous espérons étendre la technique de filtrage par l'architecture systolique pour implémenter les filtres numériques RIF 2-D pour le traitement des images numériques.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

Bibliographie

[1] Chao Cheng and Keshab K. Parhi, "Hardware Efficient Fast Parallel FIR Filter Structures Based on Iterated Short Convolution", IEEE Transactions, on Circuits and System-I: Regular Paper, vol. 51, No.8, August 2004.

[2] Chao Chang and Keshab K. Parhi, "Low Cost Parallel FIR Filter Structures with 2-stage parallelism", IEEE, TCASI-3380, February 2006.

[3] D.A. Parker and Keshab K. Parhi, "Low-area/power parallel FIR digital filter implémentations", Journal of VLSI Signal processing Systems, vol. 17, no. 1, pp. 75-92, Sept. 1997.

[4] Chao Cheng and Keshab K. Parhi, "Further complexity reduction of parallel FIR filters", Proceedings of 2005 IEEE Intenational Symposium on Circuits and Systems (ISCAS2005), Kobe, Japan, May 2005.

[5] Y.N. Chang, J.H. Satyanarayana and KeshabK. Parhi, "Design and Implémentation of Low-Power Digit-Serial Multipliers", Proc. of IEEE Conf. on Computer Design, pp. 186-195, Austin, October 12-15, 1997.

[6] Y.N. Chang and Keshab K. Parhi, "Efficient FFT implementation using digit-serial arithmetic", in Proc. 1999 IEEE Workshop Signal Processing Systems, Taipei, Taiwan, R.O.C., Oct. 1999, pp. 645-653.

[7] Keshab k. Parhi, "Approaches to Low-Power Implementations of DSP Systems", IEEE Transactions on Circuits and Systems-I Fundamental Theory and Applicatons, vol. 48, No. 10, October 2001.

[8] Paul Bougas, Paraskevas Kalivas, Andreas Tsirikos and Kimamal. Z. Pekmestzi, "Pipelined Array-Based FIR Filter Folding", IEEE Transaction on Circuit and Systèms-Vol. 52. No. 1. January 2005.

[9] Chao Chang and Keshab K. Parhi, "Hardware Efficient Fast DCT Based on Novel Cyclic Convolution Structures", IEEE Transactions on Signal Processing, Vol. 54. No. 11, November 2006.

[10] C. Cheng and K. K. Parhi, "A novel systolic array structure for DCT ", IEEE Trans. Circuits Syst. II: Express Briefs, vol. 52, pp. 366-369, Jul. 2005.

[11] D. F. Chiper, M. N. S. Swarny. M. O. Ahmad and T. Stouraitis, "A systolic array architecture for the discrete sine transform", IEEE Trans. Signal Process., vol. 50. 2002.

[12] Djamel Chikouche, "Contributions à l'implémentation des filters numériques récursifs unidimentionnels et bidimensionnels sur des réseaux processeurs concurrents : Réalisation de hauts débits en données par l'usage de la technique CTP", Doctorat thesis, University of setif, Algeria, 2000.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

[13] H. T. Kung, "Let's design algorithms for VLSI systems," Proceed. Conf. Very Large Scale Integration: Architecture, Design, Fabrication, California Institute of Technology, Jan. 1979, pp. 65-90.

[14] D. W. L. Yen and A. V. Kulkarni, "Systolic processing and implementation for signal and image processing ", IEEE Trans. Comput. , vol. C-31, No 10, Oct. 1982, pp. 1000-1009.

[15] H.J. Nussbaumer, "Fast Fourier Transform and Convolution Algorithms", 2nd ed. Berlin. Germany: Springer-Verlag, 1982.

[16] W. M. Gentleman and H. T. Kung, "Matrix triangularization by systolic arrays", Proceed. SPIE Symp., vol. 298: Real–time signal processing IV, Society of photo-optical instrumentation engineers, Aug. 1981, pp. 19-26.

[17] T. S. Chang, J. I. Guo, and C. W. Jen, "Hardware-Efficient DFT designs with cyclic convolution and sub expression sharing", IEEE Trans. Circuits Syst. II: Analog Digital Signal Process, vol. 47, Sep. 2000.

[18] Tsay. J. C and Yuan. S, "Some combinatorial aspects of parallel algorithm design for matrix multiplication", IEEE Trans. Comput 1992. 41, (3). pp. 355-361.

[19] William A. Porter, "Fast Forms of Banded Maps ", IEEE Transactions on acoustics. Speech, and Signal Processing. Vol. 38, No. 7, July 1990.

[20] Porter. W. A. and Aravena. J. L. "Cylindrical arrays for matrix multiplication", Proceedings of the 24th Annual Allerton Conference, October 1986, pp. 595 – 602.

[21] Parakevas Kalivas, Paul Bougas, Andreas Tsirikos, George Economakos and Kiamal Z. Pekmestzi, "New Systolic and Low Latency Parallel FIR Filter Schemes", Transactions on engineering, Computing and Technology V2, December 2004.

[22] Lin li, Shu-qin Lou, Xiaolin Liu and Jie Liu,"The Implementation of Digit-Serial FIR Filters Based on FPGA", IEEE international Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communication Procedings, 2005.

[23] Chao Cheng and Keshab K. Parhi, "A novel systolique array structure for DCT", IEEE Trasns. Circuits Syst. II: Express Briefs, vol. 52. pp. 366-369. Jul 2005. w

[24] O.T. Chen and W. L. Liu, "A FIR filter processor with programmable dynamic data ranges"**,** IEEE Trans. on VLSI Systems, vol. 8, no. 4. August 2000, pp. 440-447.

[25] André Quinquis, "Le Traitement du Signal sous Matlab, pratique et application", Hermès, Paris 2000.

[26] P.Janardhanan and M.N.Neelakatnan, "Very fast method for the design of FIR digital filters with linear phase", IEEE Transaction on circuits and systems, vol.35, No.2, February 1988.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

[27] J. H. McClellan, T. W. Parks and L. R. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters" IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 506-526, Dec 1973.

[28] Mustapha Rouha, "Etude comparative des principales méthodes de synthèse des filtres R.I.F à phase lineaire", Master thesis, University of setif, Algeria, 1996.

[29] M. Bellanger, "Traitement Numérique Du Signal", masson, Paris 1990.

[30] A.E, O.N and Y.Y, "Equiripple FIR Filter Design by the FFT Algorithm", IEEE Signal processing Magazine, March 1997.

[31] A. N. Belbachir, "Conception des Filtres Numériques RIF à Phase Linéaire dans l'Espace Discret des Coefficients", Master thesis, University of Oran, Algeria, 2000.

[32] Lawrence R. Rabiner, Bernard Gold, "Theory and Application of Digital Signal Processing", PRENTICE-HALL, INC. 1975.

[33] Steven W. Smith, "Digital Signal Processing", Elsevier Science 2003.

[34] Lars Wanhammar, "DSP Integrated circuits", Academic press 1999.

[35] Douglas. F. Elliott, "Handbook of Digital Signal Processing", Academic press 1987.

[36] C-M. Liu, C-W. Jen, "Design of algorithm-based fault-tolérant VLSI array processor", IEE Proceedings, Vol. 136, Pt, E, No. 6, November 1989.

[37] M.Kunt, "Traitement numérique du signal", Masson, Paris 1984.

[38] Xi Zhang and Toshinori Yoshikawa, "Design of FIR Nyquist Filters with Low Group Dealy", IEEE Transaction on Signal Processing, Vol, 47, No. 5, May 1999.

[40] Edmund Lai, "Practical Digital Signal Processing for Engineers and Technicians", Elsevier Science 2003.

[41] J. Granata, M. Conner and R. Tolimieri, "Recursive Fast Algoritms and the Role of the Tensor product", IEEE Trans. on Signal Processing, vol. 40, No. 12, December 1992.

[42] J. Granata, M. Conner and R. Tolimieri, "A tensor product factorisation of the linear convolution matrix", IEEE Trans. Circuits Syst. Vol. 38, pp. 1364 – 1366, 1991.

[43] Ayman Elnaggar, H. M. Alnuweiri and M. R. Ito, "A New Tensor Product Formulation for Toom's Convolution Algorithm", IEEE Transactions on Signal Processing, vol. 47, No. 4.April 1999.

[44] Shen-Yi Lin, Chih-Shen Chen, Li Liu and Chua-Huang Huang, "Tensor Product Formulation for Hilbert Space-Filling Curves", IEEE, Proceedings of the 2003 International Conference on Parallel Processing, (ICPP'03).

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

[45] www.textbookbuys.com, Keshab. K. Parhi/DSP.Html, "VLSI digital signal processing systems".

[46] www.ece.umn.edu/users/parhi/slides/chap8.pdf, Keshab. K. Parhi,**" Chapter 8: Fast Convolution".**

[47] L-S. Lin and S. K. Mitra, "Overlapped block digital filtering", IEEE Trans. Circuits Syst. II, vol. 43, pp. 586-596, Aug.1996.

[48] K. David. A and Keshab K.Parhi, "Area-Efficient Parallel FIR Digital Filter Implementations", IEEE Proceedings of International Conference on Application Specific Systems, Architectures and Processors, 19-21. Aug 1996. pp. 93-111.

[49] J. I. Acha, "Computational structures for fast implementation of L-path and L-block digital filters", IEEE Trans. Circuit Syst., vol. 36.pp.805-812, June 1989.

[50] Z -J. Mou and P. Duhamel, " Short-length FIR filters and their use in fast nonrecursive filtering", IEEE Trans. Signal processing, vol. 39, pp. 1322-1332, June 1991.

[51] H. T. Kung, "Why systolic architectures? ", IEEE Computer, vol. 15, No1, pp 37- 46, 1982.

[52] H.T. Kung, C. E. Leiserson, " Systolic arrays (for VLSI)", in I. S. Duff, and G. W. Stewart, eds, Sparse Matrix Proceedings 1978, SIAM, Philadelphia, 1979, pp. 256-282.

[53] Djamel Chikouche and R. Elhadi Bekka, " Cylindrical architecture for 1-D recursive digital filters: A state space approach", IEE Proc. Comput. Digit. Tech. Vol, 145, No. 5, September 1998.

[54] H. T. Kung, "Let's design algorithms for VLSI systems", Proceed. Conf. Very Large Scale Integration: Architecture, Design, Fabrication, California Institute of Technology, Jan. 1979, pp. 65-90.

[55] Jagadih. H. V and Kailath. T, "A family of new efficient arrays for matrix multiplication", IEEE Trans. Comput. 1989. 38, (1), pp 149 – 155.

[56] Djamel Chikouche, R. Elhadi Bekka, A. Khellaf and A. Boucenna, "Orbital architectures for recursive digital filters", Proc. Maghrebean Conf. on Soft. Eng. And Art. Intel, Algiers, Apr. 1996, pp. 273-277.

[57] GUO-JIF.L and Wah. B , "The design of optimal systolic array", IEEE Trans. Comput. 1985, C.34,(1). pp. 66-77.

[58] Porter. W.A and Aravena. J. L , "Orbital architectures with dynamic reconfiguration", Proc IEE, part E, Comput. Digit Tech, vol. 134, N°6, pp. 281-287, Nov1987.

[59] S. Yuan and J. C. Tsay, "Unified approach to designing parallel Winograd algorithms", IEE proceed. Comput. Digit. Tech, vol. 141, No. 3. May 1994.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

[60] Jorge L. Aravena and Abdulkader O. Barbir, "A Class of Low Complexity High Concurrence Algorithms", IEEE Transaction on Parllel and Distibuted Systems, Vol. 2, No. 04, October 1991.

[61] D. F Chiper, M. N. S Swamy. M. O. Ahmad and T. Stouraitis, "A systolic array architecture for the discrete cosine transform", IEEE. Trans. Signal Process, vol. 50, 2002.

[62] SHEK-W. C and CHIN-L. W, "The Design of Concurrent Error Diagnosable Systolic Arrays for Band Matrix Multiplications", IEEE Transactions on Computer-Aided design, Vol. 7. No. 1, January 1988.

[63] W. A. Porter and J. L. Aravena, "Array based design of digital filters", IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 38, N° 9, Sep. 1990, pp. 1628-1632.

[64] W. A. Porter, "Concurrent forms of signal processing algorithms", IEEE Trans. on Circuits and Systems, Vol. 36, No 4, Apr. 1989, pp. 553-560.

[65] M.O. Esonu, A.j. Al-Khalili, S.Hariri and D. Al-Khalili, "Systolic arrays: how to choose them", IEE Proceedings-E, Vol. 139, No.3, May 1992.

[66] H-J. Kang and I-C. Park, "FIR filter synthesis algorithms for minimizing the delay and the number of adders", IEEE Trans. on Circuit and Systems-II, vol. 48, No. 8, August 2001, pp. 770-777.

[67] K. Z.Pekmestzi, P. Kalivas and N. Moshopoulos", Long unsigned number systolic serial multipliers and Squarers", IEEE Transactions on Circuits & Systems II, Vol. 48, no. 3, pp. 316-321, March 2001.

[68] J. Guo, C. M. Liu and C. W. Jen, "A new array architecture for prime length discrete cosine transform", IEEE Trans. Signal Process., vol. 41, pp. 436-442, Jan. 1993.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!

ملخص

يهتم هذا العمل المقدم في هذه المذكرة بدراسة المرشحات الرقمية (RIF) بإستعمال جداء التزويج السريع. كذلك بإنجاز خوارزمية على التوازي للمرشحات الرقمية (RIF)، بالتصميم الإنقباضي و دراسة الجودة بالنسبة للتصميم بتقنية (VLSI).

كشفت معاينات بتقنية (VLSI) الجودة المحدودة لهذه التقنية خلال مرحلة الإنجاز، لهذا افترضنا الإنجاز بالتصميم الانقباضي لتحسين زمن تنفيذ خوارزمية الترشيح وتعقيد التصميم. تقوم طريقة الانجاز المعتمدة على خوارزمية ترشيح مكررة بضرب المصفوفات.

تبين معاينات التصميم الانقباضي النتائج ذات جودة لتعقيد التصميم و زمن التنفيذ.

الكلمات المفتاحية **:** انجاز, مرشح (RIF) , تصميم إنقباضي, زمن التنفيذ, تقنية(VLSI).

Abstract

This master thesis presents an investigation on the representation of the FIR filter using the iterated short convolution. It consists also to implement parallel algorithms of FIR digital filters, on systolic architectures and investigate their performances relativity to Parhi VLSI architecture. The tests reveal the performance limitations of Parhi architecture during the implementation phase. We have then proposed the implementation of FIR filters on systolic architectures in order to improve their hardware complexity and time execution. The procedure of implementation is based on an iterative filtering algorithm which is based on multiplication of matrices. The tests of systolic architecture show its good performances in term of time execution an hardware complexity.

Key words: Implementation, filter (FIR), systolic architecture, time execution, technology VLSI.

This PDF was created using Adolix PDF Converter PRO. Register to remove this watermark!