

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي



BADJI MOKHTAR - ANNABA UNIVERSITY
UNIVERSITÉ BADJI MOKHTAR - ANNABA

جامعة باجي مختار – عنابة

FACULTÉ DES SCIENCES

Département de physique

THÈSE

Présentée en vue de l'obtention du diplôme de Doctorat

IMPLEMENTATION D'UN PROCESSEUR FFT DEDIÉ A LA TRANSMISSION NUMÉRIQUE A HAUT DÉBIT

Option : Physique et technologie de l'optique non-linéaire
appliquée aux télécommunications optiques

Par **MOKHNACHE RAOUF**

Directeur de thèse :

Mr EL AKRMI ABD ESSETTAR

Professeur, Université Badji Mokhtar - Annaba

Co-Directrice de thèse :

Mme TRIKI HOURIA

Professeur, Université Badji Mokhtar - Annaba

Président :

GHERS MOKHTAR

Professeur, Université Badji Mokhtar - Annaba

Examineurs :

Mr GUERSI NOURREDINE

Professeur, Université Badji Mokhtar - Annaba

Mr OUCHTATI SALIM

Professeur, Université 20 aout 1955 - Skikda

Mr MOUSSAOUI ABDELKRIM

Professeur, Université 8 mai 1945 - Guelma

REMERCIEMENTS

Avant tout, je remercie mon DIEU « ALLAH » pour m'avoir accordé la santé, la patience, la connaissance et le courage nécessaires pour mener cette thèse à son terme.

Je tiens à remercier en premier toutes les personnes qui m'ont accompagné et guidé pendant ma thèse. Ma gratitude va d'abord au Professeur EL AKRMI ABD-ESSETTAR et au Professeur TRIKI HOURIA, qui m'ont proposé le sujet et qui pendant ces années m'ont encadré dans mes activités de travail et de recherche. Merci de m'avoir écouté et guidé, vous m'avez énormément appris dans le domaine scientifique et professionnel.

Je remercie Monsieur GHERS MOKHTAR, Professeur de l'université Badji Mokhtar d'Annaba, qui m'a fait l'honneur de présider le jury de soutenance de ma thèse.

Mes remerciements les plus vifs et les plus chaleureux vont à :

- Monsieur OUCHTATI SALIM, professeur de l'université 20 août 1955 de Skikda
- Monsieur MOUSSAOUI ABDELKRIM de l'université 8 mai 1945 de Guelma
- Monsieur GUERSI NOUREDDINE Professeur de l'université Badji Mokhtar d'Annaba,

Ils m'ont fait l'honneur d'accepter d'examiner mon travail, de l'enrichir par leurs grands savoirs et leurs grandes expériences. Ils ont aussi accepté d'être membres examinateurs du jury de soutenance, et à ce titre, je leur dois une gratitude éternelle et sans limites.

Qu'ils trouvent ici, encore une fois, l'expression sincère de ma reconnaissance et de mon respect.

MERCI, professeurs ! merci infiniment.

Je tiens aussi à remercier Monsieur BENOUARET MOHAMED, Professeur de l'université Badji Mokhtar d'Annaba, qui m'a fait bénéficier de ses compétences avec disponibilité, constance et bonne humeur.

DEDICACES

Ce travail est dédié à mes êtres les plus chers qui se reconnaîtront dans cette dédicace. Un grand merci, un très grand merci à vous tous.

Un très grand merci plein de reconnaissance et de gratitude.

A ma chère mère OUARDA, qui a cru en moi et m'a toujours soutenu. Avec son soutien, je me suis toujours senti plus fort. Merci encore une fois, ma Maman chérie. De tout cœur.

A ma sœur, SOFIA en lui souhaitant tout le courage du monde pour ce qui reste de son propre travail de thèse. Je suis certain de sa réussite et confiant dans ses compétences. Tu es une jouvence permanente. Merci SOFIE. Merci, Merci infiniment.

A ma sœur, NADJOUA qui m'a toujours apporté un soutien sans failles. Ton amour est une source permanente de fraîcheur et de courage. Merci NADJOUITA. Merci, Merci infiniment.

A ma sœur, MALAK qui a toujours été un soutien de tous les moments. Ton amour m'a fait vivre dans les moments difficiles. Merci MALLOUKA HABIBTI. Merci, Merci infiniment.

A mon père AZZOUZ, pour tous les moments que nous avons passés ensemble et pour le soutien qu'il m'a apporté. Merci encore.

A MERIEM, ma femme chérie, avec tout l'amour du monde et une très grande place dans mon cœur. Merci Meriouma, Merci infiniment.

A toute sa famille avec une mention toute spéciale pour mes beaux-parents Ammi HAS-SANE et Khalti NADIA.

Sans oublier bien sûr ses frères DJAMEL IMAD et MOHAMED AKRAM, notre petit ange avec des pensées toutes affectueuses pour eux tous.

Ma famille toute entière a cru en moi. Mes tantes et mes oncles tant du côté paternel que du côté maternel trouveront ici l'expression de mes remerciements. Merci, merci et merci.

A tous mes amis, je vous remercie car pendant ces années, vous m'avez donné votre support de toutes les manières possibles, en m'écoutant, en échangeant nos avis, en me faisant repartir pendant les moments de découragements, en me faisant comprendre le sens de la persévérance et du courage.

RESUME

Les applications de télécommunications sont en constante évolution. L'amélioration des normes actuelles des réseaux mobiles impose entre autres d'augmenter l'efficacité spectrale pour un meilleur taux de transmission tout en présentant une grande robustesse des systèmes aux perturbations. Ces perturbations sont pour une grande partie dues aux canaux de transmission à trajets multiples. Les techniques multi-porteuses répondent partiellement à cet objectif de robustesse aux perturbations. Parmi ces techniques de transmissions à hauts débits, le multiplexage par répartition orthogonale de fréquences (OFDM) est l'un des systèmes de communication à large bande les plus adoptés.

La technique du multiplexage par répartition orthogonale de la fréquence (OFDM) est célèbre pour sa robustesse contre les canaux à évanouissements sélectifs en fréquence. Cette technique a été largement utilisée dans de nombreux systèmes de communication filaires et sans fil. En général les opérations de transformée de Fourier rapide inverse (IFFT) et de FFT (FFT) sont utilisées comme modulation / démodulation dans les systèmes OFDM.

Dans une chaîne de transmission OFDM, le processeur de Fourier représente donc un élément clé dans la performance du système de communication. La transformation de Fourier rapide (FFT) représente la clé de ce problème dans un tel système. La mise en œuvre de l'algorithme FFT dans une cible FPGA représente un problème non négligeable lorsqu'il s'agit de respecter les contraintes de l'application en termes de consommation d'énergie, de coût de mise en place, de zone occupée, et de vitesse de calcul. Naturellement, en raison des exigences de calculs intensifs, les processeurs de calcul de la FFT occupent de grandes surfaces et consomment beaucoup d'énergie lorsqu'ils sont mis en œuvre dans le matériel.

L'objectif majeur de cette thèse porte principalement sur la conception et la mise en œuvre d'un processeur de Fourier basé sur une nouvelle approche dans le but de l'implémenter dans une cible FPGA.

Dans cette perspective, le travail proposé expose les tendances les plus largement utilisées pour réduire la consommation d'énergie, l'espace occupé et permettre des débits élevés dans les transmissions numériques afin de présenter notre nouvelle approche dans la conception d'un processeur FFT qui peut être utilisé dans une chaîne de transmission basée sur le système OFDM.

Les principes du processeur proposé reposent sur un algorithme novateur et une architecture nouvelle en termes de multiplieurs complexes. Le processeur proposé est paramétrable selon les tailles FFT à traiter et la résolution d'entrée de la suite à traiter.

Le processeur est comparé avec des réalisations dans le même domaine. Les résultats sont très satisfaisants selon les critères évoqués précédemment : consommation d'énergie, coût, simplicité de mise en place, zone occupée, vitesse de calcul et hauts débits. De plus il présente un intérêt certain en termes de simplicité d'implémentation sur une cible FPGA.

Mots clés : transformation de Fourier rapide (FFT), architecture processeur FFT, optimisation, large bande, FPGA, OFDM, mise en œuvre en temps réel.

ABSTRACT

Telecommunications applications are constantly evolving. The improvement of the current standards of mobile networks requires among other things to increase the spectral efficiency for a better transmission rate while presenting a great robustness of the systems to disturbances. These disturbances are largely due to multipath transmission channels. Multi-carrier techniques partially meet this objective of robustness to disturbances. Among these high-speed transmission techniques orthogonal frequency division multiplexing (OFDM) is one of the most widely adopted broadband communication systems. The technique of orthogonal frequency division multiplexing (OFDM) is famous for its robustness against selective frequency vanishing channels. This technique has been widely used in many wired and wireless communication systems. In general the operations of inverse fast Fourier transform (IFFT) and FFT (FFT) are used as modulation / demodulation in OFDM systems.

In an OFDM transmission chain the Fourier processor therefore represents a key element in the performance of the communication system. The fast Fourier transformation (FFT) represents the key to this problem in such a system. The implementation of the FFT algorithm in an FPGA target represents a not insignificant problem when it comes to respecting the constraints of the application in terms of energy consumption cost of implementation occupied area and computing speed. Naturally due to the requirements of intensive computing FFT computing processors occupy large areas and consume a lot of energy when implemented in hardware.

The main objective of this thesis mainly concerns the design and implementation of a Fourier processor based on a new approach in order to limit it in an FPGA target. In this perspective the proposed work exposes the trends most widely used to reduce energy consumption occupied space and allow high throughput in digital transmissions in order to present our new approach in the design of an FFT processor which can be used in a transmission chain based on the OFDM system.

The principles of the proposed processor are based on an innovative algorithm and a new architecture in terms of complex multipliers. The proposed processor can be configured according to the FFT sizes to be processed and the input resolution of the suite to be processed.

The processor is compared with realizations in the same domain. The results are very satisfactory according to the criteria mentioned above energy consumption cost simplicity of installation occupied area speed of calculation and high flow rates. In addition it has a definite advantage in terms of simplicity of implementation on an FPGA target.

Key words: Fast Fourier Transform (FFT), Architecture, Broadband, FPGA, OFDM, real time implementation

خلاصة

تتطور تطبيقات الاتصالات باستمرار. يتطلب تحسين المعايير الحالية لشبكات الهاتف المحمول ، من بين أمور أخرى ، زيادة الكفاءة الطيفية للحصول على معدل إرسال أفضل مع تقديم متانة كبيرة للأنظمة ضد الاضطرابات. ترجع هذه الاضطرابات إلى حد كبير إلى قنوات الإرسال متعددة المسارات. تلبية التقنيات متعددة الموجات الحاملة جزئيًا هذا الهدف من التصدي لقوة الاضطرابات. ومن بين تقنيات الإرسال عالية السرعة الأكثر استعمالًا ، يعد تعدد الإرسال بتقسيم تعامدي للتردد أحد أكثر أنظمة الاتصالات واسعة النطاق التي يتم تبنيها على نطاق واسع. بمتانتها ضد قنوات التلاشي الانتقائي للترددات تم استخدام هذه التقنية على نطاق واسع في العديد من أنظمة الاتصالات السلكية واللاسلكية. تشتهر تقنية تعدد الإرسال بتقسيم تعامدي " أو أف دي أم " بالصالبة و المتانة ضد الاضطرابات.

بشكل عام ، تُستخدم عمليات تحويل فورييه السريع و تحويل فورييه المعكوس السريع كتعديل / إزالة التردد في الأنظمة " أو أف دي أم". في سلسلة الإرسال " أو أف دي أم" ، يمثل معالج فورييه عنصرًا رئيسيًا في أداء نظام الاتصالات. يمثل تحويل فورييه السريع مفتاح هذه المشكلة في مثل هذا النظام. يمثل تنفيذ خوارزمية "أف أف تي" في هدف " أف بي جي أ" مشكلة كبيرة عندما يتعلق الأمر باحترام قيود التطبيق من حيث استهلاك الطاقة ، تكلفة التركيب ، المنطقة المستعملة ، وسرعة الحساب. في هذا المنظور يكشف العمل المقترح الاتجاهات الأكثر استخدامًا لتقليل استهلاك الطاقة والمساحة المشغولة والسماح بإنتاجية عالية في الإرسال الرقمي من أجل تقديم نهجنا الجديد في تصميم معالج "أف أف تي" الذي يمكن أن يكون مستخدم في سلسلة إرسال قائمة على نظام " أو أف دي أم". تستند مبادئ المعالج المقترح إلى خوارزمية مبتكرة وبنية جديدة من حيث المضاعفات المعقدة. يمكن تكوين المعالج المقترح وفقًا لأحجام "أف أف تي" المراد معالجتها ودقة مدخلات المجموعة المراد معالجتها.

تتم مقارنة المعالج مع الإنجازات في نفس المجال.
النتائج مرضية للغاية وفقاً للمعايير المذكورة أعلاه : استهلاك الطاقة ، التكلفة ، بساطة التركيب ، المنطقة المستعملة ، سرعة الحساب والسرعات العالية لتدفق المعطيات.
بالإضافة إلى ذلك ، لديها ميزة محددة و كبيرة من حيث بساطة التنفيذ على هدف " أف بي جي أ".

الكلمات المفتاحية : تحويل فوربييه السريع "أف أف تي" ، بنية معالج "أف أف تي" ، التحسين ، النطاق العريض ، " أف بي جي أ" ، " أو أف دي أم" ، التنفيذ في الوقت الفعلي.

LISTE DES ACRONYMES

A

ASIC: Application Specific Integrated Circuit

ADSL: Asymmetric Digital Subscriber Line

B

BPE: Butterfly Processor Element

C:

CP: Cyclic Prefix

CP: Cyclic Prefix - Orthogonal Frequency Division Multiplexing

D

DSP: Distributed Arithmetic

DSP: Digital Signal Processor

DSL: Digital Subscriber Line

DIT: Decimation In Time

DIF: Decimation In Frequency

DVB-T: Digital Video Broadcasting – Terrestrial

DVB-H: Digital Video Broadcasting – Handheld

DMT: Discrete Multi Tone

E

ETSI: European Telecommunications Standards Institute

F

FDD: Frequency-Division Duplexing

FDM: Frequency Division Multiplexing

FFT: Fast Fourier Transform

FIR: Finite Impulse Response

FPGA: Field Programmable Gate Array

LISTE DES ACRONYMES

HW: Hardware

I

ICI: Inter Carrier Interference

IFFT: Inverse Fast Fourier Transform

IES: Interférences Entre Symboles

IEP: : Interférences Entre Porteuses

ISI: Inter-Symbol Interference

L

LSB: Least Significant Bit

LTE: Long Term Evolution

LUT: Look Up Table

M

MAQ: Modulation D'amplitude En Quadrature

MC-CDMA: Multi-Carrier Code Division Multiple Access

MSB: Most Significant Bit

MUX: Multiplexer

MOF: Maximum Of Frequency

N

NoC: Network-on-Chip

O:

OFDM: Orthogonal Frequency Division Multiplexing

PE: Processing Element

PLL: Phase-Locked Loop

Q

QAM: Quadrature Amplitude Modulation

QPSK: Quadrature Phase Shift Keying

LISTE DES ACRONYMES

R

R: Register

RTL: Register Transfer Level

S

SBMP: Standard Bit Parallel base complex

SDF: Single-path Delay Feedback

SoC: System-on-Chip

SQNR: Signal-to-Quantization Noise Ratio

SW: Software

U

UMTS: Universal Mobile Telecommunications System

UWB: Ultra Wide Band

V

VHDL: Very High Speed Integrated Circuit Hardware Description Language

VLSI: Very-Large-Scale Integration

W

WiMAX: Worldwide Interoperability for Microwave Access

Z

ZP-OFDM : Zero Padding Orthogonal Frequency Division

LISTE DES FIGURES

| | |
|--|----|
| Figure 1-1- Influence du multi trajet sur le brouillage inter symboles..... | 25 |
| Figure 1-2- Effet positif de la propagation en contexte multi-trajets..... | 26 |
| Figure 1-3- Effets négatif de la propagation en contexte multi trajets | 26 |
| Figure 1-4- Fonction de transfert d'un canal comportant un retard τ | 27 |
| Figure 1-5- Schéma de principe de la modulation OFDM | 30 |
| Figure 1-6- Spectre en sortie du modulateur OFDM..... | 31 |
| Figure 1-7- Vision spectrale de la modulation OFDM..... | 32 |
| Figure 1-8- Schéma de principe du démodulateur OFDM. | 32 |
| Figure 1-9- Spectre du signal transmis $s(t)$ | 33 |
| Figure 1-10- Spectre du signal transmis $s(t)$ en bande de base | 33 |
| Figure 1-11- Chaîne de transmission d'une modulation OFDM | 34 |
| Figure 1-12- Chaîne de transmission d'une modulation OFDM. | 34 |
| Figure 1-13- Modulateur utilisant la IFFT..... | 35 |
| Figure 1-14- Problème des interférences entre symboles | 35 |
| Figure 1-15- Insertion d'un intervalle de garde ' Δ ' | 36 |
| Figure 1-16- Illustration de l'insertion du préfixe cyclique | 37 |
| Figure 1-17- Ajout d'un intervalle de garde | 39 |
| Figure 1-18- Exemple d'une base orthogonale | 39 |
| Figure 1-19- Principe d'orthogonalité des sous porteuses en OFDM..... | 40 |
| Figure 1-20- Spectres des différentes porteuses..... | 41 |
| Figure 1-21- Spectre du signal OFDM pour 10 porteuses..... | 41 |
| Figure 1-22- Interférence inter-porteuse (ICI) en OFDM dans les domaines | 43 |
| Figure 1-23- Effets du canal à trajets multiples sur des symboles reçus dans le cas..... | 44 |
| Figure 1-24- Illustration de l'intervalle de garde précédant chaque symbole OFDM..... | 45 |
| Figure 1-25- Effet de l'intervalle de garde précédant chaque symbole OFDM | 45 |
| Figure 1-26- Schéma synoptique des modulations OFDM | 46 |
| Figure 1-27- Synoptique d'une chaîne de transmission OFDM | 47 |
| Figure 2-1- Les coefficients de la DFT pour $N=8$ | 54 |
| Figure 2-2- Décomposition d'une FFT 15 points en deux FFT 3 points et 5 points [15]..... | 57 |
| Figure 2-3- structure DIT Butterfly | 60 |
| Figure 2-4- structure DIF Butterfly..... | 60 |
| Figure 2-5- algorithme Radix-2 DIT pour $N=8$ | 61 |
| Figure 2-6- algorithme Radix-2 DIF pour $N=8$ | 62 |
| Figure 2-7- éléments de R sur le cercle unité | 66 |
| Figure 2-8-Structure Butterfly du Split-Radix DIT | 67 |

LISTE DES FIGURES

| | |
|---|-----|
| Figure 2-9- Architecture à mémoire unique..... | 69 |
| Figure 2-10- Architecture basée sur une mémoire duale..... | 70 |
| Figure 2-11- Architecture Mémoire Cache | 71 |
| Figure 2-12- Architecture en Tableaux pour les calculs FFT..... | 72 |
| Figure 2-13- Architecture Radix-2 Multipath Delay Commutator FFT pour 16-points..... | 73 |
| Figure 2-14- Architecture Radix-4 Multipath Delay Commutator FFT pour 16-points..... | 73 |
| Figure 2-15- Architecture Radix-2 Single Path Delay Feedback FFT pour 16-points..... | 74 |
| Figure 2-16- Architecture Radix-4 Single Path Delay Feedback FFT pour 16-points..... | 74 |
| Figure 2-17- Les additionneurs utilisés dans les architectures en pipeline..... | 75 |
| Figure 2-18- Les multiplieurs complexes utilisés dans les architectures en pipeline..... | 75 |
| Figure 3-1- Diagramme fonctionnel de l'algorithme FFT | 81 |
| Figure 3-2 Décimation des échantillons sous forme de matrices..... | 83 |
| Figure 3-3- Le commutateur splitter..... | 84 |
| Figure 3-4- Le commutateur splitter et espace mémoire de stockage des données décimées.. | 84 |
| Figure 3-5- Le commutateur splitter et espace mémoire de stockage des données décimées. | 85 |
| Figure 3-6- Le commutateur splitter, espace mémoire de stockage des données décimées et blocs de facteurs twiddles..... | 85 |
| Figure 3-7- Le commutateur splitter, espace mémoire de stockage des données décimées et blocs de facteurs twiddles..... | 86 |
| Figure 3-8- Le commutateur splitter et espace mémoire de stockage des données décimées.. | 86 |
| Figure 3-9- Le commutateur splitter, espace mémoire de stockage des données décimées et blocs de facteurs twiddles avec sélecteur et mémoires de sorties..... | 87 |
| Figure 3-10- Conception graphique d'un processeur élément pour FFT Radix 64..... | 90 |
| Figure 4-1- Position des pôles sur le cercle unitaire..... | 98 |
| Figure 4-2- Modélisation des calculs..... | 99 |
| Figure 4-3- Modélisation des calculs pour $X1$ | 99 |
| Figure 4-4- Modélisation des calculs pour $X2$ | 100 |
| Figure 4-5- Modélisation des calculs pour $X3$ | 101 |
| Figure 4-6- Schéma bloc du banc de filtres numériques $N = 4$ | 103 |
| Figure 4-7- Détails du banc de filtres numériques $N = 4$ | 103 |
| Figure 4-8- schéma bloc du banc de filtres numériques N quelconque..... | 104 |
| Figure 4-9- schéma détaillé du banc de filtres numériques N quelconque | 104 |
| Figure 5-1- La structure d'une multiplication distribuée | 118 |
| Figure 5-2- La structure d'une multiplication distribuée à base d'une LUT Look-Up Table Multiplication | 119 |
| Figure 5-3- schéma bloc d'une LUT (Entrée (5 bits) * 67, sortie (15 bits))..... | 120 |
| Figure 5-4- Le schéma bloc d'une LUT (Entrée = 25, sortie = 1675)..... | 121 |
| Figure 5-5- algorithme de Booth..... | 121 |
| Figure 5-6- Diagramme du circuit multiplieur complexe | 125 |

LISTE DES FIGURES

| | |
|---|-----|
| Figure 5-7- Seconde version du diagramme du circuit multiplieur basée sur la multiplication complexe | 126 |
| Figure 5-8 - Circuit Simulink du premier facteur twiddle du multiplieur complexe..... | 127 |
| Figure 5-9 - Diagramme Simulink de l'élément processeur du premier échantillon spectral | 127 |
| Figure 5-10- Schéma d'implémentation du composant principal | 128 |
| Figure 5-11 -Script Matlab expliquant le fonctionnement | 130 |
| Figure 6-1- Evolution Du SNR pour différentes résolutions d'entrées et FFT de différentes résolutions..... | 131 |
| Figure 6-2-Valeurs des facteurs Twiddles pour une FFT-16 points | 132 |
| Figure 6-3- Multiplieur (Bit Parallel) du facteur $\text{Cos}(\pi/8)$ | 136 |
| Figure 6-4- Multiplieur (Bit Parallel) du facteur $\text{Sin}(\pi/8)$ | 136 |
| Figure 6-5- Multiplieur (Bit Parallel) du facteur $\text{Cos}(\pi/4)$ | 137 |
| Figure 6-6- Diagramme du timing de l'algorithme proposé pour le calcul de la FFT pour l'échantillon X_1 | 138 |
| Figure 6-7- La surface occupée par l'algorithme confrontée à différents algorithmes | 140 |
| Figure 6-8- Evolutions des débits pour plusieurs approches..... | 141 |

LISTE DES TABLEAUX

| | |
|--|-----|
| Tableau 1 - Comparaison du nombre d'opérations dans un calcul direct de la DFT et dans la FFT | 58 |
| Tableau 2- Opération de Bit-Reversed | 64 |
| Tableau 3- Nombre d'opérations d'une FFT de 4096 points pour différents Radix | 67 |
| Tableau 4- Représentation bidimensionnelle d'une séquence initiale décimée en temps | 79 |
| Tableau 5- Représentation bidimensionnelle de la DFT d'une séquence réarrangée | 79 |
| Tableau 6- Représentation bidimensionnelle des facteurs twiddles | 80 |
| Tableau 7- Résultats de la multiplication des facteurs Twiddles par la DFT des colonnes séparées | 80 |
| Tableau 8- réarrangement de $x(n)$ sous forme matricielle | 82 |
| Tableau 9: Décimation des échantillons sous forme de matrices | 83 |
| Tableau 10 - Evolution des calculs du schéma pour X1 | 100 |
| Tableau 11 - Evolution des calculs du schéma pour X2 | 101 |
| Tableau 12- Modélisation des calculs pour X3 | 102 |
| Tableau 13- Représentation du contenu d'une LUT à 6 entrées pour 3 x 3 bits | 119 |
| Tableau 14- Exemple de la représentation de la LUT (entrée (5 bit) * 67) | 120 |
| Tableau 15 Soustractions des multiples de deux de 237 | 134 |
| Tableau 16 - Soustractions des multiples de deux de 946 | 135 |
| Tableau 17 -Performances de l'algorithme proposé | 139 |

TABLE DES MATIERES

| | |
|---|-----------|
| Remerciements..... | 1 |
| Dédicaces... .. | 2 |
| Résumé..... | 3 |
| Abstract..... | 5 |
| خلاصة..... | 6 |
| Liste des acronymes..... | 7 |
| Liste des figures..... | 10 |
| Liste des tableaux..... | 13 |
| Table des matières..... | 14 |
| Introduction Générale..... | 17 |
| 1 TRANSMISSIONS OFDM | 22 |
| 1.1 Introduction aux transmissions OFDM..... | 22 |
| 1.2 Historique des modulations multi-porteuses et de l'OFDM | 22 |
| 1.3 Problematique..... | 24 |
| 1.4 Effets des trajets multiples | 25 |
| 1.4.1 Effets positifs des multi trajets | 26 |
| 1.4.2 Effets négatifs des multi trajets | 26 |
| 1.5 Modèle simplifié | 27 |
| 1.6 La sélectivité des canaux..... | 28 |
| 1.7 Introduction à l'ofdm | 29 |
| 1.8 Principe et fonctionnement de l'OFDM | 29 |
| 1.8.1 Modulation OFDM | 29 |
| 1.8.2 démodulation ofdm | 32 |
| 1.8.3 Principe du modulateur et du démodulateur | 33 |
| 1.8.4 Problème des interférences entre symboles | 35 |
| 1.8.5 Nécessité des porteuses orthogonales | 39 |
| 1.9 La chaîne de transmission OFDM | 46 |
| 1.10 Avantages et limites de l'OFDM | 47 |
| 1.10.1 Avantages de l'OFDM..... | 48 |
| 1.10.2 Limites de l'OFDM..... | 49 |
| 1.11 Applications de l'OFDM | 49 |
| 1.12 Conclusions | 51 |
| 2 ETAT DE L'ART FFT | 52 |

TABLE DES MATIERES

| | | |
|------------|---|------------|
| 2.1 | Introduction | 52 |
| 2.2 | Etat de l'art des algorithmes FFT | 53 |
| 2.2.1 | Introduction | 53 |
| 2.2.2 | La Transformée de Fourier Discrète | 53 |
| 2.2.3 | La Transformée Rapide de Fourier..... | 54 |
| 2.2.4 | Approche diviser-pour-régner | 56 |
| 2.3 | Classification des algorithmes FFT | 57 |
| 2.4 | Radix 2 | 59 |
| 2.4.1 | Formulation matricielle de l'algorithme Cooley-Tuckey..... | 62 |
| 2.4.2 | Opération bit reverse | 63 |
| 2.5 | Radix 4 | 64 |
| 2.6 | Radix 8 | 66 |
| 2.7 | Split-Radix FFT (SRFFT)..... | 67 |
| 2.8 | Autres versions FFT..... | 68 |
| 2.8.1 | PFA (Prime factor Algorithm) [Good-Thomas]:..... | 68 |
| 2.8.2 | Rader- Brenner..... | 68 |
| 2.8.3 | Brunn's Algorithm:..... | 68 |
| 2.8.4 | Algorithme de Winograd | 68 |
| 2.8.5 | Algorithme de Rader | 68 |
| 2.8.6 | Algorithme de Bluestein | 69 |
| 2.9 | Architectures FFT..... | 69 |
| 2.9.1 | Architectures basées sur des mémoires (Memory-based architectures)..... | 69 |
| 2.9.2 | Architectures basées sur des mémoires Cache (Cached memory architectures)..... | 70 |
| 2.9.3 | Architectures en Tableaux (Cellulaires) (Array architectures) | 71 |
| 2.9.4 | Architectures en pipeline (Pipelined architectures) | 72 |
| 2.9.5 | Multi-Path Delay Commutator | 72 |
| 3 | ALGORITHME PROPOSE | 76 |
| 3.1 | Introduction | 76 |
| 3.2 | Représentation bidimensionnelle de l'algorithme FFT..... | 76 |
| 3.3 | Reformulation de la FFT et construction de la matrice | 78 |
| 3.4 | Facteurs Twiddles..... | 79 |
| 3.5 | Exemple d'un processeur FFT Radix-16..... | 82 |
| 3.6 | Exemple d'un processeur FFT Radix-64..... | 89 |
| 3.7 | Généralisation à un processeur FFT de N points Radix-M..... | 93 |
| 4 | MODIFICATION ET AMÉLIORATION | 94 |
| 4.1 | Modification de l'algorithme proposé | 94 |
| 4.1.1 | Introduction | 94 |
| 4.1.2 | Passage de la forme matricielle à la forme vectorielle | 96 |
| 4.1.3 | Généralisation et extension de la méthode améliorée | 104 |
| 4.2 | Conclusion..... | 114 |

TABLE DES MATIERES

| | | |
|----------|---|------------|
| 5 | CONCEPTION DU MULTIPLIEUR COMPLEXE..... | 115 |
| 5.1 | Arithmétique sur FPGA | 115 |
| 5.2 | Multiplieurs série | 116 |
| 5.2.1 | La multiplication distribuée | 116 |
| 5.3 | Multiplieurs parallèles | 121 |
| 5.3.1 | Multiplieur de Booth..... | 121 |
| 5.3.2 | Multiplieur Vedic | 122 |
| 5.3.3 | Multiplieur à arbre de Wallace..... | 123 |
| 5.4 | Fondement mathématique du multiplieur complexe de notre algorithme..... | 124 |
| 5.5 | Problème de modélisation et simulation Matlab | 126 |
| 5.6 | Implémentation hardware de l'algorithme adopté | 128 |
| 6 | Résultats de la confrontation et discussion | 131 |
| 6.1 | Résolution et mise à l'échelle :..... | 132 |
| 6.1.1 | Résolution 8 bits | 133 |
| 6.1.2 | Résolution 10 bits | 134 |
| 6.2 | Structure des multiplieurs..... | 136 |
| 6.2.1 | Multiplieur (Bit Parallel) du facteur $\text{Cos}(\pi/8)$ | 136 |
| 6.2.2 | Multiplieur (bit parallel) du facteur $\text{Sin}(\pi/8)$ | 136 |
| 6.2.3 | Multiplieur (Bit Parallel) du facteur $\text{Cos}(\pi/4)$ | 137 |
| 6.3 | Performances de l'algorithme proposé..... | 139 |
| 7 | CONCLUSIONS et perspectives | 142 |
| 8 | BIBIOGRAPHIE | 144 |

INTRODUCTION GENERALE

L'avancement des systèmes numériques remplace continuellement les anciens systèmes analogiques. Les systèmes numériques sont l'épine dorsale des matériels, produits, processus et services modernes et leur qualité est de plus en plus soumise à ces systèmes. Les applications des systèmes numériques actuels incluent les systèmes de communication, les systèmes de trafic, les systèmes de contrôle, les systèmes de prévision météorologique, Internet etc. Le matériel et les logiciels de la microélectronique sont devenus un matériau essentiel pour les principaux gadgets, produits, processus et services. La technologie microélectronique moderne met en œuvre un système de traitement de l'information complexe complet sur une seule puce. La transformation des systèmes multi-puces en systèmes sur une puce (SoC) ouvre de nouveaux résultats potentiels et génère de nouveaux défis. Les progrès de la technologie microélectronique sont exceptionnellement rapides.

De nos jours les systèmes numériques sont configurés pour faire face aux exigences des applications de traitement du signal numérique (DSP) les plus exigeantes, qui imposent des conditions rigoureuses telles qu'une petite surface d'implémentation, une fréquence d'horloge élevée, un débit élevé, une consommation d'énergie réduite, une latence faible et des calculs en temps réel performants. Afin de répondre à ces exigences, des systèmes numériques nouveaux sont en cours d'implémentation sur des circuits intégrés à application spécifique (ASIC) et des circuits FPGA (Field Programmable Gate Arrays).

Dans les systèmes de transmission OFDM à hauts débits, les dispositifs souhaités doivent fonctionner à des fréquences d'horloge élevées et permettent d'atteindre de grandes performances en calculs de traitement de signal numérique (DSP) afin de permettre des débits élevés et faire preuve de robustesse face aux perturbations. Les opérations de modulation et de démodulation sont réalisées grâce à des blocs FFT / IFFT.

La recherche d'algorithmes FFT performants pour le calcul DFT reste donc d'actualité et fait l'objet de très nombreuses innovations.

L'algorithme de Cooley-Tukey est le plus populaire parmi tous les autres algorithmes FFT. La décimation en temps (DIT) et la décimation en fréquence (DIF) sont deux formes de base de l'algorithme de Cooley-Tukey. Dans les algorithmes de FFT, les décompositions sont appliquées de manière récursive jusqu'aux DFT d'un petit nombre de points requis appelés Radix.

La transformée de Fourier rapide (FFT) est un algorithme très efficace pour le calcul rapide de la transformée de Fourier discrète (DFT) et de son inverse. La FFT nécessite moins de calculs que l'évaluation directe de la DFT. Elle tire parti du fait que le calcul des coefficients peut être effectué de manière itérative. La FFT réduit la complexité de calcul de la DFT de $O(N^2)$ à $O(N \log N)$.

La FFT a diverses applications où le domaine de fréquence d'un signal doit être analysé. Elle a attiré l'attention dans les systèmes de communication car la FFT est une opération de traitement cruciale dans les systèmes de multiplexage par répartition orthogonale de la fréquence (OFDM). L'OFDM est une technique de modulation de premier plan qui permet d'élargir de manière appropriée l'utilisation des canaux et de réduire les interférences entre symboles (IES) ainsi que les interférences de porteuse (IEP) générées par l'effet de trajets multiples. Il est principalement utilisé dans les domaines de la radiodiffusion audionumérique (DAB) de la télédiffusion numérique terrestre (DVB-T) et de la télédiffusion numérique (DVB-H). En plus de cela il existe de nombreuses applications de communication basées sur l'OFDM où la FFT est l'opération de traitement la plus importante telle que l'UWB (Ultra Wide Band), le LTE, la ligne d'abonné numérique (DSL) le WLAN le WiMax etc.

Lors de la conception des systèmes de communication, le principal facteur déterminant les performances est le débit du système. En raison des immenses besoins en débit de données, les modules FFT efficaces à haut débit mais avec une surface de puce réduite et une consommation d'énergie réduite sont particulièrement recherchés.

Ainsi, le travail de recherche mené dans cette thèse présente une implémentation matérielle nouvelle et efficace d'un processeur FFT avec une implémentation sur cible FPGA.

Notre choix de la cible s'est porté sur les FPGA car ce sont des cibles technologiques très intéressantes de par leurs caractéristiques dont certaines des plus importantes sont le faible coût, la grande fréquence de fonctionnement et la possibilité de reconfiguration in-situ.

Contributions

Les apports réalisés dans ce travail de thèse sont tout à fait originaux. Ils s'inscrivent dans le domaine du développement d'un processeur FFT basé sur une idée nouvelle de traitement et de calcul de la FFT. La multiplication de deux nombres complexes, si importante dans les calculs FFT, repose aussi sur une conception architecturale nouvelle qui aboutit sur l'implantation de multiplieurs numériques avancés. La conjonction de ces deux innovations aboutit à une architecture complète d'un processeur FFT et à son implémentation sur une cible FPGA Virtex-6 xc6lx75t-2ff484.

Dans cette optique, ces idées, traduites sous forme de blocs, ont été utilisées pour mettre en œuvre l'algorithme proposé ainsi que les multiplieurs complexes qui réalisent dans leur ensemble le processeur FFT de notre approche.

Le processeur ainsi implémenté est configurable en nombre d'échantillons à traiter, en résolution d'entrée des suites à traiter et des facteurs twiddles et en structure de multiplication complexe (Parallel Base Complex Multiplication).

Pour atteindre ces objectifs, le principal apport et la nouveauté dans la présente thèse peuvent être résumés comme suit :

Une proposition d'une nouvelle architecture FFT configurable :

- 1) - En nombre d'échantillons à traiter
- 2) - En résolution des échantillons en entrée à traiter
- 3) - En résolution des facteurs twiddles
- 4) - En structure de multiplication complexe (Parallel Base Complex Multiplication).

Nous avons aussi pour objectif de nous affranchir des structures PBE (Butterfly Processor Element), ainsi que des opérations de Bit-Reverse, nécessaires pour la majorité des algorithmes. La définition détaillée d'une nouvelle forme de multiplieur complexe basée sur une idée originale est proposée. Elle se base sur l'objectif de réduire le nombre de multiplieurs complexes par rapport à une implémentation brute de la multiplication classique de deux nombres complexes, car comme le prouve la littérature et les travaux dans ce domaine, la multiplication est très gourmande en ressources, et partant de là, conditionne les performances du processeur. En fin de réalisation, le processeur proposé atteint tous les objectifs visés en termes de paramètres configurables et se libère des structures BPE (Butterfly Processor Element) qui deviennent complexes dès que la taille de la FFT augmente en fonction en particulier des Radix utilisés. L'opération Bit-Reverse est aussi éliminée, réduisant considérablement les interconnexions. Les données en entrées sont entrées dans l'ordre naturel et les sorties du processeur proposé sont données en parallèle directement et simultanément, évitant ainsi la nécessité de pré-arranger les données en entrée ou post-arranger les résultats en sortie (élimination du Bit-Reverse).

Cadre conceptuel

Ce sujet entre dans le cadre des thématiques du LPR (Laboratoire de la Physique Des Rayonnements, Département De Physique) (Université Badji Mokhtar, Annaba) et du laboratoire LASA (Université Badji Mokhtar, Annaba).

Il s'inscrit dans le cadre d'un travail proposé par le LPR et réalisé en collaboration avec le LASA.

Plan de la thèse

Ce manuscrit comporte sept chapitres (deux chapitres introductifs consacrés aux thèmes à développer, cinq chapitres consacrés à notre travail. Le dernier chapitre est consacré aux conclusions et perspectives). Ces chapitres se répartissent comme suit :

Une introduction générale expose la problématique et le cadre de notre travail et détaille les étapes de la thèse.

Le premier chapitre donne une présentation détaillée de la transmission OFDM. Cette présentation est primordiale dans la mesure où notre processeur s'intègre dans une chaîne de transmission OFDM, ce qui est le but essentiel de ce travail, bien que le processeur développé puisse être utilisé dans d'autres types d'applications comme processeur FFT.

Le deuxième chapitre donne une présentation détaillée de l'état de l'art de la FFT. Les algorithmes FFT les plus utilisés et les architectures des processeurs FFT les plus courants y sont présentés avec une comparaison des ressources utilisées et des performances atteintes. Cette présentation nous permettra d'effectuer dans les prochains chapitres une comparaison avec notre algorithme et son implémentation hardware et d'exposer les nettes performances obtenues au regard des critères les plus pertinents, en particulier les tailles FFT traitables, les registres Slice utilisés, les latences, les cycles, Les MOF (Maximum Of Frequency) et les débits atteints.

Le troisième chapitre propose notre approche pour le processeur développé. Nous y exposons les représentations bidimensionnelles de l'algorithme FFT, la reformulation de la FFT et la construction de la matrice. Des exemples de deux processeurs FFT Radix-16 et FFT Radix-64 sont clairement présentés et permettent de généraliser l'utilisation du processeur aux cas d'une taille N quelconque et un Radix quelconque. Ce chapitre traite aussi de la projection à partir de l'espace 2D de la séquence d'entrée présélectionnée sur le plan 2D formé par la matrice des facteurs twiddles pour obtenir un vecteur de la composante principale qui reflète le comportement fréquentiel de la séquence d'origine à traiter.

Le chapitre quatre traite les modifications et les améliorations apportées à notre réalisation. Notre souci majeur est d'éviter la structure matricielle pour les raisons suivantes :

- La latence, facteur déterminant pouvant influencer négativement sur la qualité de transmission en termes de débits et donc en volume des données à transmettre
- Gourmandise en termes de ressources internes (primitives ; registres internes, bascules, mémoires, additionneurs, multiplieurs...). Ceci augmente considérablement la surface occupée.
- Complexité apparente en termes de méthodologie de programmation des différents blocs nécessaires pour l'implémentation finale du processeur

Pour, notre contribution suivante consiste à créer une structure récursive de filtres numériques afin de faciliter le processus de calcul de la TFD qui devrait répondre aux exigences de transmission OFDM à grande vitesse à moindre coût et à un compromis intelligent pour éviter les contraintes susmentionnées.

Dans le but d'expliquer notre approche de cette structure récursive de filtres numériques envisagée, nous nous concentrons sur le cas $K = 1$ et $N = 4$ c'est-à-dire qu'on effectue le calcul de la première raie spectrale $X(1)$ d'une suite de $N = 4$ échantillons.

Le chapitre cinq traite des fondements mathématiques du multiplieur complexe. Avant de se lancer dans la phase de mise en œuvre d'un tel composant et de choisir ensuite une architecture appropriée il est primordial de proposer une réduction ostensible en termes de ressources primitives afin d'améliorer de manière significative les performances souhaitées. Cette réduction est obtenue par des schémas nouveaux proposés pour réduire le nombre de multiplieurs complexes au strict minimum et de les remplacer du point de vue fonctionnel par des additionneurs, plus faciles à utiliser et surtout moins gourmands en ressources.

Les problèmes de simulation sous MATLAB et SIMULINK sont mis en lumière. Une confrontation avec les résultats obtenus avec les fonctions implémentées sous MATLAB est faite et nous avons validé les résultats sous SIMULINK. Les résultats obtenus sous MATLAB et SIMULINK sont strictement identiques aux résultats donnés par notre processeur, cette concordance valide totalement notre approche.

Dans le chapitre six, nous présentons les résultats des simulations sous XILINK ISE. Un code VHDL complet décrivant le processeur a été écrit. Ceci nous a permis d'implémenter le processeur sous XILINK ISE. Le FPGA cible est un Xilinx Virtex-6 xcvlx75t-2ff484. Ce choix est motivé par les ressources proposées par cette carte et son utilisation très répandues dans le monde des FPGA, ce qui facilitera son intégration dans des projets plus étendus dans la perspective des nombreuses améliorations qui peuvent être entrevues. Nous confrontons notre architecture avec les autres architectures présentées dans le chapitre deux sous l'angle des critères de performance qui y sont énumérés, La comparaison fait ressortir un net avantage selon tous les critères en faveur de notre approche.

Le chapitre sept est consacré à une conclusion de notre travail. Cette thèse étend l'utilisation de Radix-k à une structure appropriée d'un filtre basé sur un vecteur principal d'un composant d'une matrice FFT 2D réarrangée à l'aide de la décomposition DIT (décimation en temps). En effet, il est démontré que cette structure est plus efficace que celles basées sur la rétroaction quand plusieurs séquences parallèles d'entrée doivent être traitées. En outre, le choix du vecteur principal du composant de la matrice FFT 2D réarrangée, censé être une information révélant le comportement fréquentiel de la séquence d'entrée à étudier, joue un rôle clé dans la mise en œuvre optimale de cet algorithme. Enfin, les résultats expérimentaux confirment que le modèle proposé est clairement efficace aussi bien en surface occupée, en latence et en performances de débit, rendant possible l'obtention de débits plus élevés, autour de deux Giga échantillons/s, avec de faibles latences.

1 TRANSMISSIONS_OFDM

1.1 INTRODUCTION AUX TRANSMISSIONS OFDM

L'adaptation de l'information à transmettre au canal de propagation est un des problèmes majeurs en télécommunications. Pour les canaux sélectifs en fréquence, les trajets des signaux sont multiples. Le signal est réfléchi en plusieurs endroits, et des échos apparaissent et créent des perturbations dont l'influence augmente avec le débit de transmission. Parmi les solutions étudiées pour trouver un remède à ce problème, les techniques d'accès multiple avancées [1]. et en particulier la technique OFDM (Orthogonal Frequency Division Multiplexing ont une place de choix. Cette technique consiste en l'utilisation de la modulations multi-porteuses dans laquelle un bloc d'information est modulé par une transformée de Fourier. Cette technique a connu un très vif et très grand succès ces dernières années et est en phase de normalisation avancée dans différents standards sans fils (IEEE802.11a, WiMAX, LTE, DVB). La technique OFDM a le très grand mérite de transformer un canal multi-trajet large bande en un ensemble de sous-canaux mono-trajet très simples à égaliser. De plus, l'utilisation pertinente et ingénieuse de redondance cyclique à l'émission permet la réduction de la complexité des terminaux grâce à l'utilisation d'algorithmes à base de FFT rapides.

Le présent chapitre expose les caractéristiques du canal de transmission multi trajets et introduit la modulation OFDM.

Les principes généraux (chaîne de transmission...), les avantages (simplicité de l'égalisation, utilisation d'algorithmes FFT rapides....) et les désavantages de l'OFDM sont brièvement exposés

1.2 HISTORIQUE DES MODULATIONS MULTI-PORTEUSES ET DE L'OFDM

Durant les dernières décennies, les systèmes de télécommunications se sont développés très considérablement et ont réalisé une véritable révolution. L'un des événements majeurs les plus spectaculaires est que la connexion câblée traditionnelle est, dans une très large mesure sinon totalement, remplacée par la connexion sans fil à une vitesse exponentielle [2].

Dans les années 80/90, un téléphone mobile était encore financièrement inaccessible, alors qu'aujourd'hui la majorité des gens en ont un. De plus, les missions des téléphones portables se sont diversifiées : non seulement ils sont utilisés pour les appels vocaux, mais aussi pour la transmission d'images, de données et bien d'autres informations dans des technologies que nous utilisons quotidiennement dans des réseaux locaux sans fil, de l'audio à la télévision tout est numérique [3]. Le marché des systèmes radio mobiles se développe très rapidement. Les avancées technologiques, qu'elles soient matérielles ou logicielles sont proposées quasiment chaque jour. Les nouvelles normes naissent sans cesse pour introduire les technologies les mieux adaptées à ce marché très diversifié. Ces normes permettent d'éviter les interférences entre les différents systèmes. Les développements des circuits microélectroniques permettent d'implanter les techniques, méthodes et algorithmes de traitements des signaux numériques de plus en plus complexes [4].

Parallèlement, la demande vertigineusement croissante en services pour les communications mobiles de plus en plus performants et par conséquent gourmands en termes de bande passante, oblige les concepteurs à chercher des systèmes de transmission avec de fortes efficacités spectrales [4].

L'étape de modulation est un maillon primordial dans une chaîne de transmission et son optimisation garantit un certain degré de performance. Actuellement, les solutions sont basées de plus en plus sur des schémas de transmissions multi-porteuses du fait de leur excellente robustesse vis-à-vis des canaux à trajets multiples. Parmi ces solutions, la modulation OFDM se pose comme une solution de référence [5], pour les réseaux sans fil ou encore Discrete Multi Tone (DMT) pour les réseaux filaires.

La différence déterminante et fondamentale entre les différentes techniques classiques de modulations multi-porteuses et l'OFDM est que cette dernière autorise un fort recouvrement spectral entre les porteuses. Cela permet d'améliorer grandement l'efficacité spectrale en augmentant de manière sensible le nombre de porteuses [6].

L'idée d'une transmission de données modulées en parallèle, utilisant plusieurs fréquences porteuses, n'est ni nouvelle ni récente. Le concept de modulations multi-porteuses découle de celui de multiplexage fréquentiel (FDM). Ce concept (FDM) a été introduit à la fin des années 50 et 60 et a été utilisé dans des systèmes de communications hautes fréquences militaires, tels que les systèmes Kineplex, ANDEFT et KATHRYN [2], [7],[8].

Par la suite, beaucoup d'autres chercheurs s'intéressèrent de plus en plus aux modulations multi-porteuses. En 1966 des conditions d'orthogonalité furent mises en évidence. Cela permet aux spectres des sous-porteuses respectives de se chevaucher, en optimisant ainsi la bande occupée du signal émis [3]. Durant cette même année R. W. Chang a proposé le premier schéma d'un système OFDM [9].

Quelques années plus tard, R. W. Chang et R. A. Gibby améliorèrent le concept en introduisant la notion de signaux orthogonaux à bande limitée, concept que l'on appellera par la suite OFDM [2].

En 1971, S. Weinstein et P. Ebert simplifièrent le schéma de modulation/démodulation en utilisant la transformée de Fourier discrète inverse (IFFT) à l'émetteur et (FFT) au récepteur, plus simple à utiliser et surtout plus facile à implémenter sous forme d'algorithme rapide [10].

Plus tard (1997), d'autres travaux sur cet aspect de systèmes OFDM ont également démontré que la transformée d'Hadamard pouvait remplacer le banc de modulateurs [3].

Le chevauchement en réception de plusieurs versions retardées du signal émis entraînait d'une part l'interférence entre symboles successifs (IES, en anglais ISI : Inter-Symbol Interference), et d'autres parts l'interférence entre porteuses (IEP, en anglais ICI: Inter Carrier Interference). Afin de remédier à ce problème, en 1980, A. Peled et A. Ruiz ont proposé l'ajout d'un intervalle de garde cyclique (CP: Cyclic Prefix) où la fin du signal OFDM est recopiée dans l'intervalle de garde [11].

Une bibliographie est présentée ci-après Elle n'est pas exhaustive Les ouvrages [7],[9], [10], [12]-[13] présentent la modulation OFDM. Les deux premiers exposent la théorie et les applications de l'OFDM. Le chapitre six du troisième ouvrage détaille le fonctionnement de l'OFDM.

Le quatrième ouvrage décrit le design d'un système OFDM et cherche à mettre en avant les différences entre l'OFDM et le système MC-CDMA. Le dernier ouvrage offre une vision générale sur les concepts de base, l'optimisation des performances ainsi que les problèmes liés à l'OFDM

1.3 PROBLEMATIQUE

Un signal radiofréquences est émis sur un canal, qui constitue son support physique. Les contraintes physiques de son support limite toute transmission numérique. Un canal est dit sélectif en fréquence lorsqu'il ne se comporte pas de façon identique quelle que soit la fréquence du signal à transmettre. Certaines fréquences seront transmises plus rapidement que d'autres, ou encore atténuées plus que d'autres. Le signal sera alors déformé lors de la transmission, les données seront dispersées dans le temps, pouvant mener à des interférences entre symboles.

Ce phénomène de sélectivité en fréquence est aggravé par la présence de trajets multiples pour un même signal transmis. Du fait des nombreuses réflexions que le signal peut subir en environnement urbain, le récepteur recevra une série d'échos d'amplitudes et de retards variables. Cette problématique du canal à trajets multiples est critique dans le cas d'un canal radio mobile, c'est-à-dire lorsque le récepteur et l'émetteur ne sont pas fixes relativement. Les différents échos et amplitudes varient dans l'espace, ils varieront dans le temps dans le cas d'un récepteur mobile.

Le canal de transmission à trajets multiples est caractérisé par [14] :

- T_m son retard maximum (ou étalement des retards). Si la durée d'un symbole est grand devant T_m , le canal est non sélectif en fréquence (ou dit plat dans le domaine fréquentiel) mais il peut être atténué ou amplifié.
- T_c le temps de cohérence ou Bd son pendant fréquentiel appelé spectre Doppler.

Si $B \ll Bd$ le signal ne subit pas de distorsion dans le temps.

La fonction de transfert, $h(t, \tau)$, $\tau_n(t)$ et $\alpha_n(t)$ représentant l'atténuation et le retard en fonction du temps du nième écho, et f_c la fréquence porteuse :

$$h(t, \tau) = \sum_n \alpha_n(t) \cdot e^{-2j\pi f_c \tau_n} \delta(t - \tau_n(t)) \quad (1.1)$$

Ces différents trajets pourront alors générer des interférences destructrices ou constructrices, suivant la localisation du récepteur relativement à l'émetteur et suivant les caractéristiques des obstacles rencontrés. Des interférences destructrices peuvent mener à la perte totale du signal.

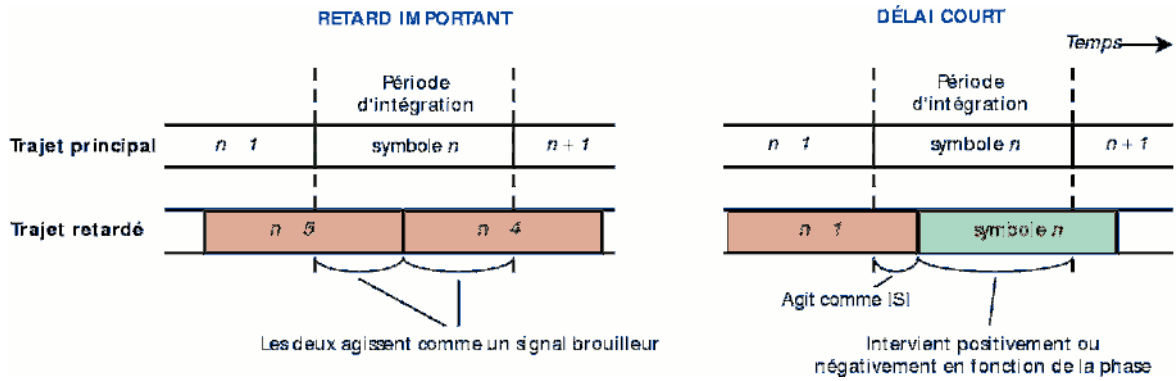


Figure 1-1- Influence du multi trajet sur le brouillage inter symboles

Ces problématiques sont d'autant plus d'actualité que les débits transmis augmentent exponentiellement, et en conséquence la bande de fréquence nécessaire pour transporter ces informations à haut débit. Or l'effet de la sélectivité en fréquence des canaux sur la dégradation des performances augmente avec la largeur de bande de fréquence du signal transmis.

Les processus d'égalisation censés compenser les effets des multi trajets et de la sélectivité en fréquence des canaux sont d'une grande complexité lorsque le canal varie beaucoup dans le temps ou suivant la fréquence du signal.

Ils nécessitent de plus la connaissance à tout instant de la fonction de transfert du canal de transmission.

Les modulations multi porteuses dont fait partie l'OFDM (Orthogonal Frequency Division Multiplexing) permet de répondre à cet enjeu en utilisant des sous-porteuses peu sensibles aux multi trajets et à la sélectivité en fréquence, faciles à égaliser.

1.4 EFFETS DES TRAJETS MULTIPLES

Une onde se propage dans tout l'espace ou, suivant le type d'environnement, elle va être réfléchie ou absorbée par les obstacles rencontrés. Le nombre d'ondes réfléchies est plus important en zone urbaine, qu'en zone rurale puisque le nombre de réflecteurs en zone urbaine est plus important qu'en zone rurale. L'onde radio peut en effet se réfléchir sur tout type d'obstacles, (bâtiment, montagne, camion, avion, discontinuité de l'atmosphère).

Les réflecteurs multiples provoquent en conséquence, plusieurs trajets entre l'émetteur et le récepteur (Multipath Propagation). Ceci a pour conséquences deux effets, l'un positif et l'autre négatif.

1.4.1 EFFETS POSITIFS DES MULTI TRAJETS

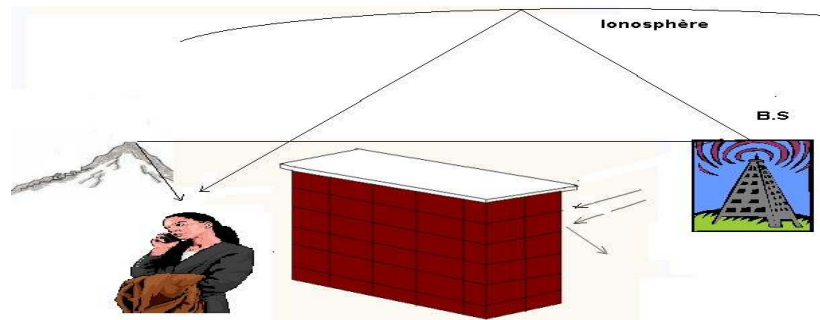


Figure 1-2- Effet positif de la propagation en contexte multi-trajets

L'avantage principal des trajets multiples est de permettre aux communications d'avoir lieu dans les cas où l'émetteur et le récepteur sont en non visibilité directe (voir figure ci dessus). En effet, les trajets multiples permettent aux ondes radio de s'affranchir des obstacles (montagnes, bâtiment, tunnels...) et donc d'assurer une certaine continuité de la couverture radio.

1.4.2 EFFETS NÉGATIFS DES MULTI TRAJETS

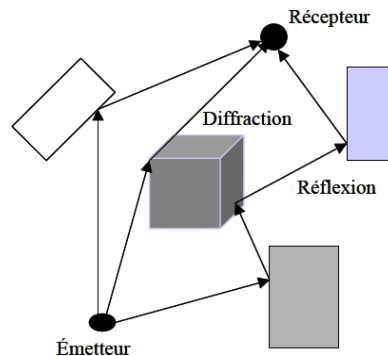


Figure 1-3- Effets négatif de la propagation en contexte multi trajets

Les trajets multiples sont aussi à l'origine de plusieurs problèmes, dont les trois principaux sont :

- L'interférence entre les trajets issus de l'émetteur qui créent des fluctuations rapides de la puissance de signal (Rayleigh fading)
- La dispersion des retards (Delay spread),
- La modulation aléatoire des fréquences due aux décalages Doppler sur les différents trajets.

1.5 MODÈLE SIMPLIFIÉ

Considérons un modèle simple du canal à l'issue duquel le récepteur reçoit la somme du signal émis et des signaux ayant subis des échos, donc retardés de τ_i et d'amplitude h_i

La réponse impulsionnelle s'écrit :

$$h(t) = \sum_i h_i \delta(t - \tau_i) \quad (I.2)$$

Sa fonction de transfert est donc :

$$H(f) = \sum_i h_i e^{-j2\pi f \tau_i} \quad (I.3)$$

Prenons l'exemple plus simple de deux trajets, le direct et un retardé de τ . La réponse impulsionnelle sera :

$$h(t) = 1 + \alpha \delta(t - \tau) \quad (I.4)$$

Et

$$|H(f)|^2 = 1 + \alpha^2 + 2\alpha \cos(2\pi f \tau) \quad (I.5)$$

La fonction de transfert pour $\alpha = 0.316$ a l'allure suivante:

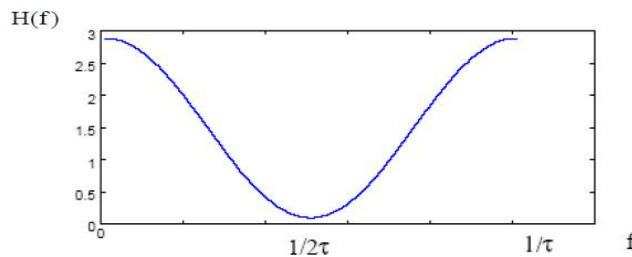


Figure 1-4- Fonction de transfert d'un canal comportant un retard τ

On remarque que la fonction de transfert comporte des zones où le signal sera amplifié ($H(f) > 1$) et des zones où le signal sera très affaibli ($H(f) < 1$) (zones d'évanouissement ou fading). Sa période de variation est de l'ordre de $1/\tau$, où τ étant l'étalement des retards.

Suivant la valeur de la bande occupée B_s par le signal, deux cas peuvent se présenter [15] :

- $B_s \ll 1/\tau$: $H(f)$ peut être considérée constante sur la bande B_s : le signal ne subit pas de distorsion, mais il peut être très affaibli si la fréquence de modulation se situe près de $1/2\tau$ (les signaux issus du trajet direct et du trajet retardé sont en opposition de phase). Mais il peut aussi être amplifié (signaux en phase).
- $B_s \gg 1/\tau$: $H(f)$ n'est pas constante sur la bande de fréquence et le signal subit des distorsions qu'il faut corriger à l'aide d'un égaliseur (l'égaliseur de canal est un estimateur de la réponse fréquentielle du canal).

1.6 LA SÉLECTIVITÉ DES CANAUX

La reconstruction des signaux transmis nécessite quelques suppositions pour un traitement numérique adéquat en aval. Les valeurs de la bande de cohérence et la fréquence de cohérence définissent la sélectivité du canal. Les multi-trajets ainsi que les déplacements de l'émetteur et/ou du récepteur sont les sources d'une sélectivité fréquentielle et temporelle. Un signal transmis peut être caractérisé par sa durée symbole T_s ainsi que sa bande B_s . La robustesse de ce signal dépendra des rapports T_s/T_c et B_s/B_c . Ainsi, quatre cas se présentent :

- $B_s \ll B_c, T_s \gg \tau$: si la bande occupée par le signal est inférieure à la bande de cohérence du canal, ou la durée du symbole émis est largement supérieure à la dispersion des retards, alors le canal est considéré comme non sélectif en fréquence.
- $B_s > B_c, T_s < \tau$: si la bande occupée par le signal est supérieure à la bande de cohérence du canal, ou la durée du symbole émis est inférieure à la dispersion des retards, alors le canal est dit sélectif en fréquence.
- $T_s < T_c, B_s > B_d$: si la durée du symbole émis est inférieure au temps de cohérence du signal, ou la bande occupée par le signal est supérieure à la bande Doppler, alors, le canal est dit non sélectif en temps. La réponse impulsionnelle du canal reste constante sur plusieurs symboles consécutifs.
- $T_s \gg T_c, B_s \ll B_d$: si la durée du symbole émis est largement supérieure au temps de cohérence du signal, ou la bande occupée par le signal est largement inférieure à la bande Doppler, alors, le canal est dit sélectif en temps. Dans ces conditions, la réponse impulsionnelle du canal varie de façon significative pendant la durée d'un symbole [16],[17].

Le canal multi-trajets à évanouissement est caractérisé par deux paramètres essentiels à savoir :

- Bande de cohérence B_s
- Temps de cohérence T_c

La variation de la réponse impulsionnelle du canal en fonction de la fréquence engendre des Interférences inter symboles (ISI).

Un signal transmis dans un canal variable dans le temps subit une nouvelle modulation due à la variation de la fréquence porteuse par l'effet Doppler. Pour remédier au problème de l'évanouissement, vient l'idée de la modulation multi porteuse (OFDM).

La modulation multiport (MC, "multi carrier") ou OFDM (Orthogonal Frequency Division Multiplexing) est abondamment utilisée pour traiter la sélectivité d'un canal. Elle permet d'effectuer l'égalisation de manière élégante, grâce à une opération de IFFT et l'ajout d'une extension cyclique (CP, "cyclique préfix") en émission, on égalise très facilement par une opération de FFT suivie d'une correction de chaque sortie par un seul coefficient complexe.

1.7 INTRODUCTION À L'OFDM

Par rapport aux modulations mono porteuses, les modulations multi porteuses présentent l'avantage d'améliorer l'efficacité spectrale. Les premières études ([18] et [19]) sur les modulations multi porteuses ont vu le jour à la fin des années 50.

Quelques années plus tard R.W.Chang et R.A. Gibby [20] introduisirent les signaux orthogonaux à bande limitée ce qui sera appelé « OFDM », Ce moyen de transmission fut ignoré pendant de nombreuses années, pour des raisons de complexité de mise en œuvre. L'usage d'algorithmes rapides de type (IFFT/FFT) ne sera proposé que plus tard [21], avec des réductions très significatives en complexité.

Peled et Ruiz [22] proposeront une version modifiée (CP-OFDM) consistant à allonger la durée du symbole OFDM par l'insertion d'un intervalle de garde (cyclique).

Grâce à ses bonnes performances et à sa complexité raisonnable, l'OFDM a été retenue dans plusieurs standards tels que les standards de diffusion numérique (DAB, DVB), les normes filaires (ADSL, PLC) et les réseaux locaux sans fil (WiFi, WiMax, etc) [23]. . Ainsi que dans l'étude es normes de communication pour réseaux locaux à l'intérieur des bâtiments.

En effet, les systèmes mono-porteuses, contrairement à l'OFDM, ne remplissaient pas les conditions de résistance aux trajets multiples et de débit élevé pour un taux d'erreur binaire faible requis par cette nouvelle application, Le multiplexage en fréquence est bénéfique pour les transmissions dans des canaux sélectifs en fréquence qui comportent des trajets multiples.

Les techniques qu'on appelle multi-porteuses consistent à transmettre des données numériques en les modulant sur un grand nombre de porteuses en même temps. Ce sont des techniques de modulation en fréquence qui existent depuis longtemps. Le regain d'intérêt actuel réside dans l'amélioration apportée pour augmenter l'efficacité spectrale en orthogonalisant les porteuses, ce qui permet d'implémenter la modulation et la démodulation à l'aide de circuits performants de transformée de Fourier rapide.

Enfin l'OFDM s'adapte parfaitement aux communications mobiles, et semble incontournable pour les futurs standards.

1.8 PRINCIPE ET FONCTIONNEMENT DE L'OFDM

1.8.1 MODULATION OFDM

Les modulations multi porteuses (OFDM) consistent à répartir les symboles sur un grand nombre de porteuses à bas débit, à l'opposé des systèmes conventionnels qui transmettent les symboles en série, chaque symbole occupant alors toute la bande passante disponible.

Ainsi dans le cas de l'OFDM, pour un train de symboles initial de période TS_i , les symboles seront répartis en N trains plus lents et auront alors une durée $TS = N \cdot TS_i$. Cette diminution du rythme symbole entraîne une diminution des interférences entre symboles d'un rapport N .

Ainsi pour un débit symbole de 10 M symboles.s⁻¹ transmis sur un canal radio de réponse impulsionnelle 250 μs, un symbole interfère avec k symboles avec $k = \frac{250}{0.1} = 2500$ symboles

Le processus d'égalisation s'effectue alors par bloc et est très complexe. En revanche, en répartissant ces symboles sur $N = 2048$ porteuses, moins de 2 symboles sont en interférence, ce qui simplifie énormément l'égalisation.

Le principe du multiplexage en fréquence est de grouper des données numériques par paquets de N , qu'on appellera symbole OFDM et de moduler par chaque donnée une porteuse différente en même temps. La modulation OFDM consiste à répartir aléatoirement des symboles de durée T_s (temps symbole utile) sur différentes porteuses modulées en QAM ou bien en QPSK [24], appelons T_s la durée symbole c'est-à-dire le temps qui sépare deux séquences de données.

Pour répartir les données à transmettre sur les N porteuses, les symboles sont groupés par paquets de N . Les c_k sont des nombres complexes définis à partir des éléments binaires par une constellation souvent de modulation MAQ à 4, 16, 64, 2^q états.

La séquence de N symboles c_0, c_1, \dots, c_{N-1} constitue un symbole OFDM. Le k ième train de symboles parmi les N trains module un signal de fréquence f_k . Le signal individuel s'écrit sous forme Complexe : $c_k e^{j2\pi f_k t}$

Le signal $s(t)$ total correspondant à toutes les données d'un symbole OFDM est la somme des signaux individuels est donné par l'expression suivante

$$s(t) = \sum_{K=0}^{N-1} c_k e^{j2\pi f_k t} \quad (1.6)$$

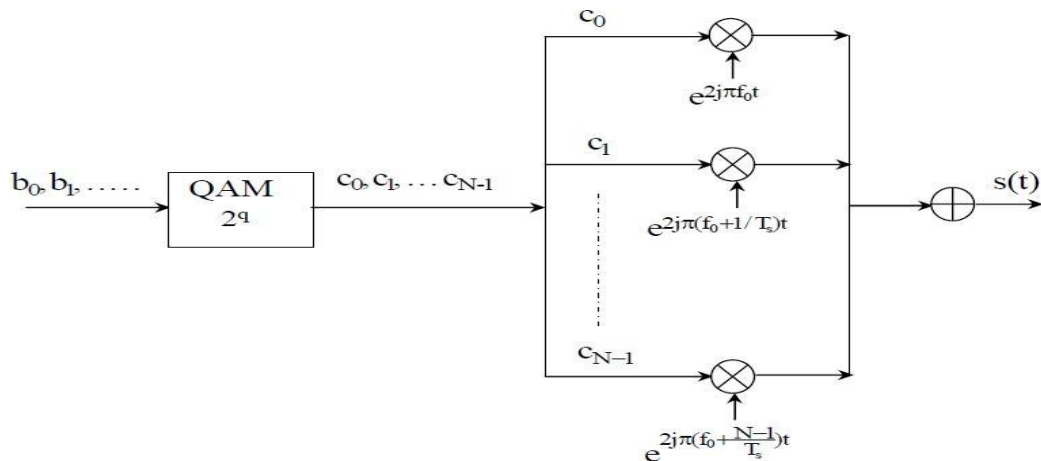


Figure 1-5- Schéma de principe de la modulation OFDM

Les fréquences sont orthogonales si l'espace entre deux fréquences f_k et f_{k+1} est $\frac{1}{T_s}$

En effet, chaque porteuse module un symbole pendant une fenêtre rectangulaire de durée

T_s son spectre en fréquences est un sinus cardinal qui s'annule tous les multiples de $\frac{1}{T_s}$, alors

$f_k = f_0 + \frac{k}{T_s}$ avec f_0 est la fréquence porteuse.

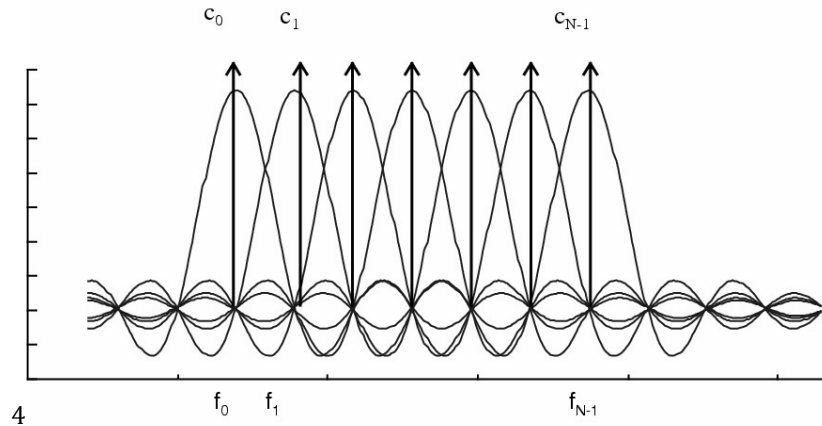


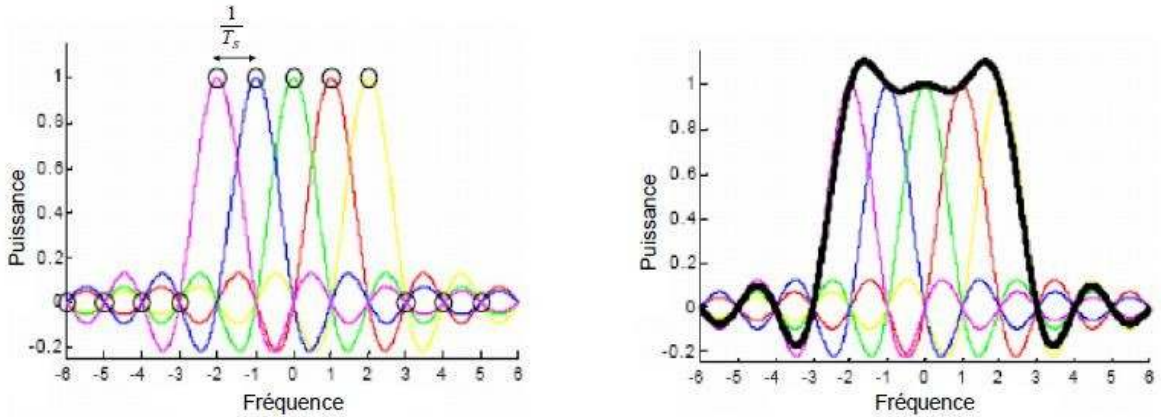
Figure 1-6- Spectre en sortie du modulateur OFDM

et

$$s(t) = e^{j2\pi f_0 t} \sum_{k=0}^{N-1} c_k e^{\frac{j2\pi k t}{T_s}} \quad (1.7)$$

Ainsi, lorsque l'échantillonnage est effectué précisément à la fréquence f_k d'une sous-porteuse, il n'y a aucune interférence avec les autres sous-porteuses. C'est ce qui permet de recouvrir les spectres des différentes porteuses et d'obtenir ainsi une occupation optimale du spectre. Le nombre de sous-porteuses N est choisi de manière à remplir les deux conditions primordiales $T_s > T_m$ afin de pouvoir considérer le canal plat, et $T_s \ll \frac{1}{B_d}$

La figure 1.7 présente les spectres des sous- système OFDM, avec $N = 5$. Les fréquences sont orthogonales si l'espace entre deux fréquences adjacentes f_k et f_{k+1} est de $\Delta f = \frac{1}{T_s}$.. En effet, chaque sous-porteuse est modulée par un symbole de donnée pendant une fenêtre rectangulaire temporelle de durée T_s , son spectre en fréquence est un sinus cardinal, fonction qui s'annule tous les multiples de Δf comme le montre la figure 1.7a. La forme du spectre multi-porteuse est illustrée sur la figure 1.7b.



(A) Orthogonalité entre les sous-porteuses OFDM (b) Spectre global du multi porteuses OFDM

Figure 1-7- Vision spectrale de la modulation OFDM

1.8.2 DÉMODULATION OFDM

Pour la démodulation, le signal parvenant au récepteur s'écrit sur une durée symbole T_s Comme suit :

$$r(t) = \sum_{k=0}^{N-1} c_k H_k(t) e^{j2\pi(f_0 + \frac{k}{T_s})t} \quad (1.8)$$

Où $H_k(t)$ est la fonction de transfert du canal autour de la fréquence $f_k = f_0 + \frac{k}{T_s}$ et au temps

t . Cette fonction varie lentement et on peut la supposer constante sur la période T_s . La démodulation classique consiste à démoduler le signal suivant les N sous-porteuses comme le montre le schéma classique ci-dessous.

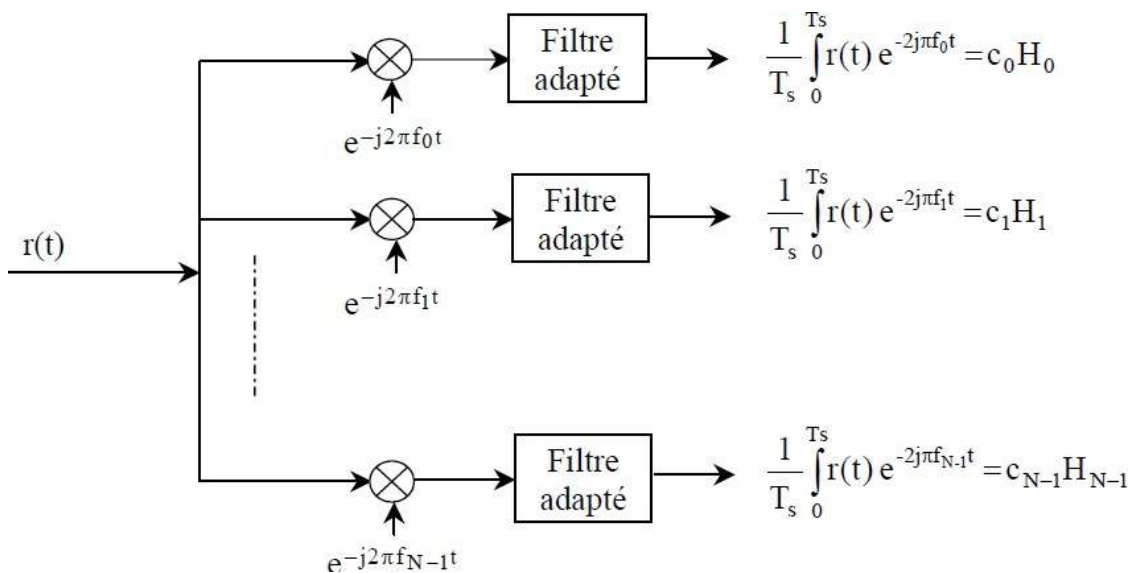


Figure 1-8- Schéma de principe du démodulateur OFDM.

La condition d'orthogonalité nous montre que :

$$\frac{1}{T_s} \int_0^{T_s} r(t) e^{-j2\pi f_i t} dt = \frac{1}{T_s} \sum_{k=0}^{N-1} \int_0^{T_s} c_k H_k(t) e^{j2\pi(k-i)\frac{t}{T_s}} dt = c_i H_i \quad (1.9)$$

Puisque

$$\frac{1}{T_s} \int_0^{T_s} e^{j2\pi(k-i)\frac{t}{T_s}} dt = \begin{cases} 0 & \text{si } k \neq i \\ 1 & \text{si } k = i \end{cases} \quad (1.10)$$

En pratique, comme pour la modulation, on remarque que la démodulation peut être réalisée par une transformée de Fourier.

1.8.3 PRINCIPE DU MODULATEUR ET DU DEMODULATEUR

Pour discrétiser, il faut choisir une fréquence d'échantillonnage. Voyons comment la démodulation impose cette fréquence. Le signal occupe la bande passante B à partir de la fréquence porteuse f_0 comme le montre la Figure suivante :

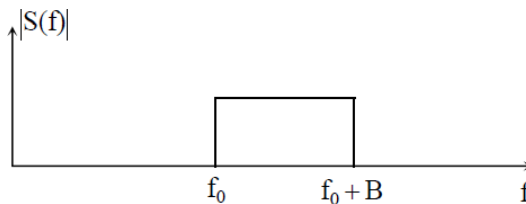


Figure 1-9- Spectre du signal transmis $s(t)$.

Pour démoduler, on va d'abord transposer le signal en bande de base, donc effectuer une translation de $f_0 + B/2$, qui est la fréquence médiane de la bande spectrale du signal. Le Spectre occupera la bande $[-B/2, B/2]$, comme indiqué ci-dessous :

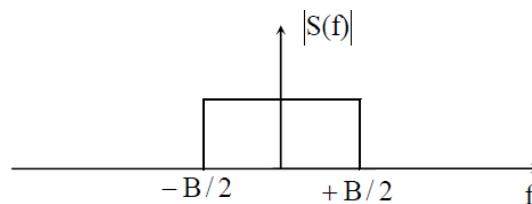


Figure 1-10- Spectre du signal transmis $s(t)$ en bande de base

La bande passante du signal étant $\frac{B}{2} = \frac{N}{2T_s}$, la fréquence d'échantillonnage doit être supérieure

ou égale à $2B/2$ soit $\frac{N}{T_s}$ s. L'échantillonnage se fera aux temps $t_n = nT_s / N$.

La chaîne de transmission est schématiquement la suivante :

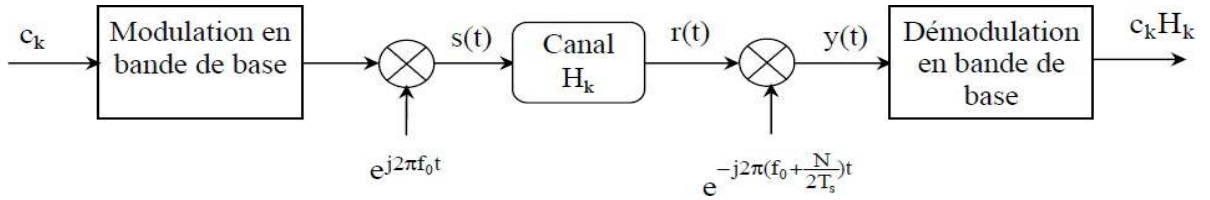


Figure 1-11- Chaîne de transmission d'une modulation OFDM

Où le signal émis est :

$$s(t) = e^{j2\pi f_0 t} \sum_{k=0}^{N-1} c_k e^{j2\pi \frac{kt}{T_s}} \quad (I.11)$$

Et le signal reçu $r(t)$ est : $r(t) = \sum_{k=0}^{N-1} c_k H_k(t) e^{j2\pi(f_0 + \frac{k}{T_s})t}$ (I.12)

Le signal reçu en bande de base après le décalage en fréquence de $f_0 + B/2$ s'écrit alors :

$$y(t) = r(t) e^{j2\pi(f_0 + \frac{N}{2T_s})t} = \sum_{k=0}^{N-1} c_k H_k(t) e^{j2\pi(\frac{2k-N}{2T_s})t} \quad (I.13)$$

Puis après échantillonnage de période T_s / N , on obtient :

$$y(t_n) = y\left(\frac{nT_s}{N}\right) = (-1)^n \sum_{k=0}^{N-1} c_k H_k e^{j2\pi \frac{kn}{N}} \quad (I.14)$$

On voit que y_n est la Transformée de Fourier discrète inverse de $C_k H_k$, la démodulation consiste donc à effectuer une Transformée de Fourier directe discrète.

L'intérêt de cette discrétisation est qu'on peut réaliser ces transformées de Fourier à l'aide d'algorithmes de FFT (direct) et IFFT (inverse).

Le schéma de principe du démodulateur se simplifie à :

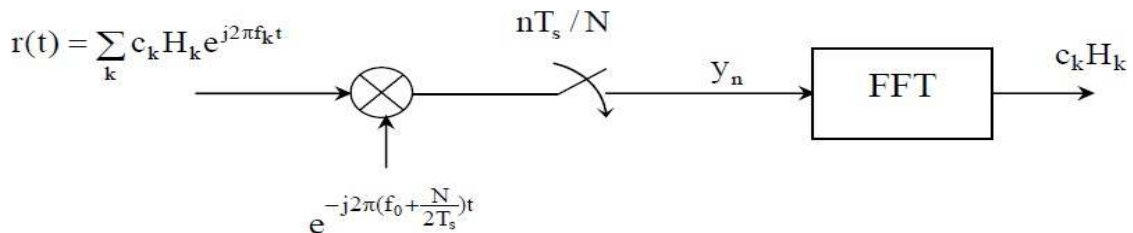


Figure 1-12- Chaîne de transmission d'une modulation OFDM.

Si on pose que le signal modulé en bande de base $s(t)$ est lui aussi discrétisé, les échantillons s_n s'écrivent alors :

$$s_n = \sum_{k=0}^{N-1} c_k e^{j2\pi \frac{kn}{N}} \quad (1.15)$$

Le schéma de principe du modulateur est le suivant :

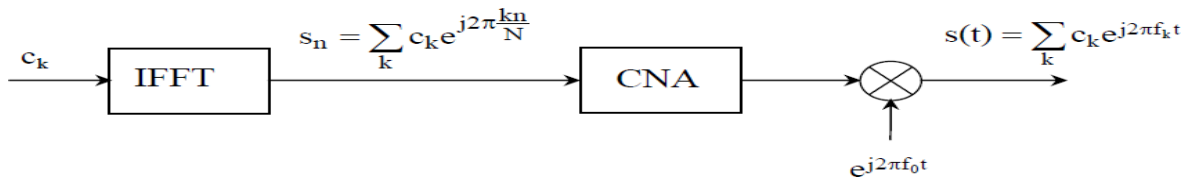


Figure 1-13- Modulateur utilisant la IFFT.

Physiquement, les symboles numériques c_k sont des données dans l'espace fréquentiel, les échantillons du signal s_n sont des données dans l'espace temporel puisqu'on passe des premières aux secondes par une transformée de Fourier inverse

1.8.4 PROBLÈME DES INTERFÉRENCES ENTRE SYMBOLES

1.8.4.1 Intervalle de garde

Comme nous l'avons vu, les symboles subissent des échos et un symbole émis parvient au récepteur sous forme de plusieurs symboles atténués et retardés. Un symbole émis lors d'une période (iTs) peut se superposer à un écho provenant du symbole émis à la période $(i - 1)Ts$. Il se produit alors des interférences comme le montre la figure ci-dessous :

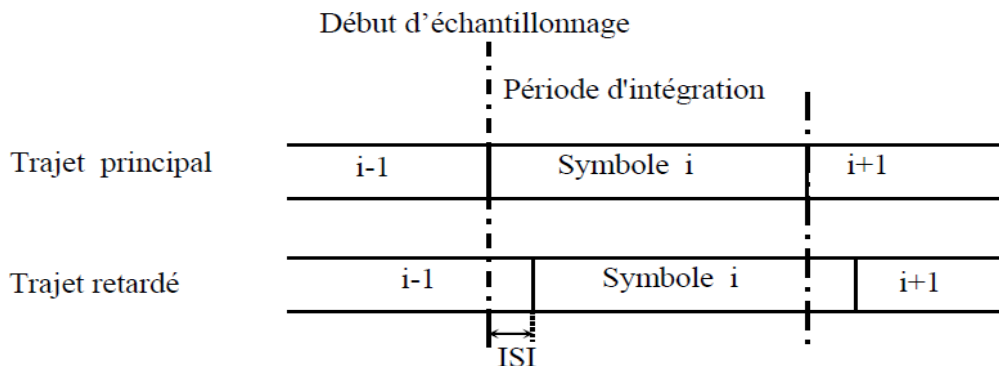


Figure 1-14- Problème des interférences entre symboles

Pour éviter ces interférences, on ajoute un intervalle de garde d'une durée Δ . La durée du Symbole totale transmis est alors ($T = T_s + \Delta$). Pour que les interférences soient éliminées, il faut que l'intervalle de garde soit plus grand que le plus grand des retards T_m qui apparaissent dans le canal comme indiqué ci-dessous.

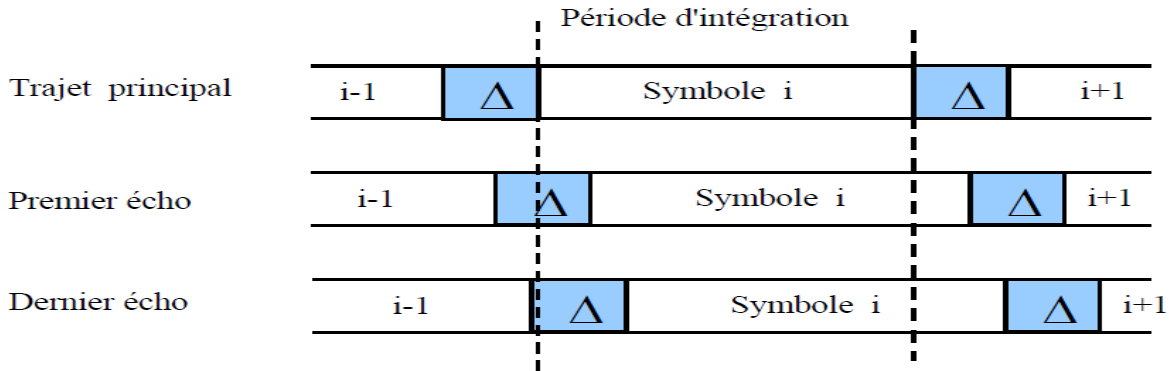


Figure 1-15- Insertion d'un intervalle de garde ' Δ '

On voit sur cette figure que si l'échantillonnage est fait au début du symbole reçu i , l'écho le plus retardé du symbole $i-1$ ne sera pas encore reçu, il faut donc que le récepteur reçoive les signaux provenant de tous les échos (ici au temps i) ce qui implique que le signal soit prolongé pendant les intervalles de garde précédant le symbole i .

Le débit qui était (qN/T_s) bits/s diminue et devient $(qN/(T_s + \Delta))$. L'intérêt de la technique OFDM est que la durée d'un symbole OFDM contenant N symboles numériques peut être grande. Si le nombre de porteuses est assez grand permettant une durée symbole T_s assez longue devant l'intervalle de garde Δ , alors le débit n'est que peu réduit [25].

1.8.4.2 Contenu de l'intervalle de garde

Puisque la durée d'émission des symboles est $T = T_s + \Delta$, le nombre d'échantillons Temporels augmente et devient $N_{total} = N + N_\Delta$.

Dans ce cas, S_n sera comme suit :

$$s_n = \sum_{k=0}^{N-1} c_k e^{j2\pi \frac{nk}{N}} \quad \text{pour } -N_\Delta \leq n \leq N-1 \quad (1.16)$$

Ceci consiste à calculer les N échantillons S_n de l'IFFT des N échantillons C_k , c.-à-d. :

$$s_n = \sum_{k=0}^{N-1} c_k e^{j2\pi \frac{nk}{N}} \quad \text{pour } 0 \leq n \leq N-1 \quad (1.17)$$

Recopier les N_Δ derniers échantillons de S_n précédemment calculés devant les N échantillons temporels comme le montre la figure ci-dessous, puisque :

($S_{n+N} = S_n$) alors, S_n pour $(-N_\Delta \leq n \leq -1)$ égales à s_n pour $(N - N_\Delta \leq n \leq N - 1)$

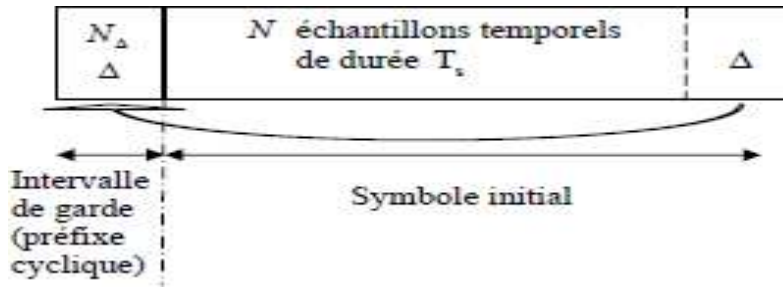


Figure 1-16- Illustration de l'insertion du préfixe cyclique

En réception, on a une convolution du signal s_n par la fonction de transfert du canal, on obtient alors :

$$r_n = \sum_{i=0}^{N_\Delta-1} h_i s_{n-i} \quad (1.18)$$

En supposant que la longueur temporelle du canal est inférieure ou égale à Δ (ici, on considère qu'elle est égale à Δ , si elle est inférieure, on complète par des échantillons nuls). La démodulation consiste à effectuer la transformée de Fourier discrète des N derniers échantillons et donc " laisse tomber " l'intervalle de garde ou le préfixe cyclique : elle fournit des échantillons y_k donnés par :

$$y_k = \sum_{n=0}^{N-1} r_n e^{-j2\pi \frac{kn}{N}} \quad (1.19)$$

En remplaçant r_n par son expression, on obtient alors :

$$y_k = \sum_{n=0}^{N-1} \sum_{i=0}^{N_\Delta-1} h_i s_{n-i} e^{-j2\pi \frac{kn}{N}} \quad (1.20)$$

$$= \sum_{i=0}^{N_\Delta-1} h_i e^{-j2\pi \frac{ki}{N}} \sum_{n=0}^{N-1} s_{n-i} e^{-j2\pi \frac{k(n-i)}{N}} \quad (1.21)$$

$$y_k = \sum_{i=0}^{N_\Delta-1} h_i e^{-j2\pi \frac{ki}{N}} \sum_{n'=-i}^{N-i-1} s_{n'} e^{-j2\pi \frac{kn'}{N}} \quad (1.22)$$

$$y_k = \sum_{i=0}^{N_\Delta-1} h_i e^{-j2\pi \frac{ki}{N}} \left\{ \sum_{n=-i}^{-1} s_{n'} e^{-j2\pi \frac{kn'}{N}} + \sum_{n=0}^{N-i-1} s_{n'} e^{-j2\pi \frac{kn'}{N}} \right\} \quad (1.23)$$

Le premier terme dans l'accolade correspond à la modification des échantillons à l'intérieur de l'intervalle de garde, le second à la modification des échantillons dans le symbole proprement

dit.

En tenant compte de $s_{n'+N} = s_{n'}$ et de $e^{-j2\pi\frac{kn'}{N}} = e^{-j2\pi\frac{k(n'+N)}{N}}$

Le résultat de la démodulation est finalement :

$s_{n'} e^{-j2\pi\frac{kn'}{N}}$ pour $-i \leq n' \leq -1$ est égale à $s_{n'} e^{-j2\pi\frac{kn'}{N}}$ pour $N-i \leq n' \leq N-1$

Donc pour retrouver les données émises, il suffit de diviser les données démodulées par la va-

$$y_k = \sum_{i=0}^{N_d-1} h_i e^{-j2\pi\frac{ki}{N}} \sum_{n'=0}^{N-1} s_{n'} e^{-j2\pi\frac{kn'}{N}} = H_k C_k \quad (1.24)$$

leur de la fonction de transfert du canal en fonction de la fréquence, ceci n'est vrai que parce que l'intervalle de garde dure plus longtemps que la fonction du canal et qu'il est constitué du préfixe cyclique. Cette opération d'estimation des symboles numériques C_k , s'appelle étape d'égalisation du canal.

1.8.4.3 Notion d'orthogonalité des porteuses

Dans la pratique, les porteuses sont modulées par des nombres complexes qui changent d'un symbole à l'autre. Si la période d'intégration porte sur deux symboles (cas des trajets retardés de la figure 1), on aura non seulement des interférences entre symboles (ISI) à l'intérieur de la même porteuse, mais aussi entre porteuses (ICI). Pour éviter ce phénomène, on ajoute un intervalle de garde pour garantir que toutes les informations intégrées viennent du même symbole et apparaissent constantes pendant sa durée.

La période du symbole T_s est prolongée de manière à être supérieure à la période d'intégration T_i . Toutes les porteuses étant cycliques à l'intérieur de T_s , il en va de même pour l'ensemble du signal modulé. Le segment ajouté au début du symbole pour former l'intervalle de garde est donc identique au segment de même longueur à la fin du symbole.

Tant que le retard d'un trajet par rapport au trajet principal (le plus court) est inférieur à l'intervalle de garde, les composantes du signal à l'intérieur de la période d'intégration viennent toutes du même symbole : le critère d'orthogonalité est satisfait. Les brouillages ICI et ISI ne se produisent que lorsque le retard relatif est plus long que l'intervalle de garde. La figure 22 illustre l'ajout d'un intervalle de garde.

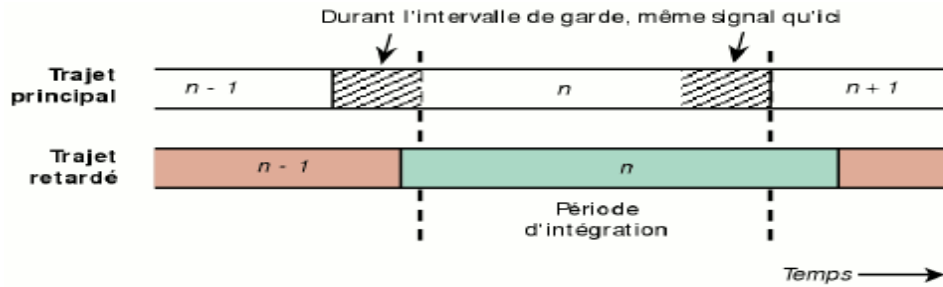


Figure 1-17- Ajout d'un intervalle de garde

1.8.5 NÉCESSITÉ DES PORTEUSES ORTHOGONALES

On définit l'orthogonalité de deux fonctions $f(t)$ et $g(t)$ dans l'intervalle $[a,b]$ par la relation suivante :

$$\int_a^b f(t).g(t)dt = 0 \quad (I.25)$$

Cela signifie que ces deux fonctions sont disjointes sur le segment $[a, b]$. Pour réaliser une base orthogonale à N dimensions, il suffit de trouver N fonctions orthogonales deux à deux. Comme la montre la figure (II.2), un ensemble de N fenêtres rectangulaires espacées d'un intervalle sur l'axe des temps constitue une base orthogonale.

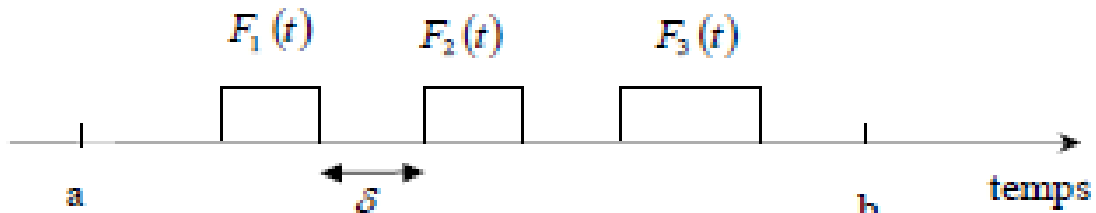


Figure 1-18- Exemple d'une base orthogonale

L'orthogonalité est la propriété fondamentale qui permet de transmettre des signaux d'informations multiples dans un même canal et de les détecter sans interférence.

1.8.5.1 Orthogonalité temporelle

Envisageons tout d'abord des signaux continus, donc non encore échantillonnés. Dans ce cas, un signal OFDM est composé d'une somme de N sinusoïdes de fréquences respectives f_k , transmises durant une durée T_s , k variant de 1 à N , et définie par $f_k = \frac{k}{T}$, cette condition permettant, d'avoir un nombre entier de sinusoïdes sur chaque sous-porteuse durant T_s .

Chaque sous-porteuse $S_k(t)$ réelle et non modulée peut se mettre sous la forme suivante pour

$$k \in [1, N]: s_k(t) = \begin{cases} \sin\left(2\pi \frac{k}{T_s} t\right) & \text{pour } 0 \leq t < T_s \\ 0 & \text{ailleurs} \end{cases} \quad (I.26)$$

Ainsi deux sous-porteuses $S_i(t)$ et $S_j(t)$, de fréquences respectives f_i et f_j , définies par l'expression (I-26), sont orthogonales sur l'intervalle $[0, T_s]$, puisqu'elles vérifient l'équation (I-25).

1.8.5.2 Orthogonalité fréquentielle

On peut aussi percevoir la notion d'orthogonalité du signal OFDM dans le domaine fréquentiel. En effet, si chaque sous-porteuse $S_k(t)$ est transmise pendant la durée T_q , cela revient à appliquer à la sous-porteuse une porte de durée T_s , dont l'enveloppe spectrale est un sinus cardinal qui s'annule aux premières fréquences

$f_1 = f_k - \frac{1}{T_s}$
 et $f_2 = f_k + \frac{1}{T_s}$. On obtient ainsi l'enveloppe spectrale représentée sur la figure (II-3), le sinus cardinal représente le spectre d'une sous-porteuse i , de bande $\Delta f = \frac{1}{T_s}$.

La figure (I-19) et la figure (I-20) montre que l'espace entre chaque sous-porteuse $\frac{1}{T_s}$ permet, lorsque le spectre d'une sous-porteuse est maximal, d'annuler le spectre de toutes les autres : C'est la condition d'orthogonalité, (Orthogonal de OFDM). Cette condition permet ainsi d'avoir une occupation spectrale idéale et d'éviter les interférences entre sous-porteuses si l'échantillonnage est fait précisément à la fréquence d'une sous-porteuse.

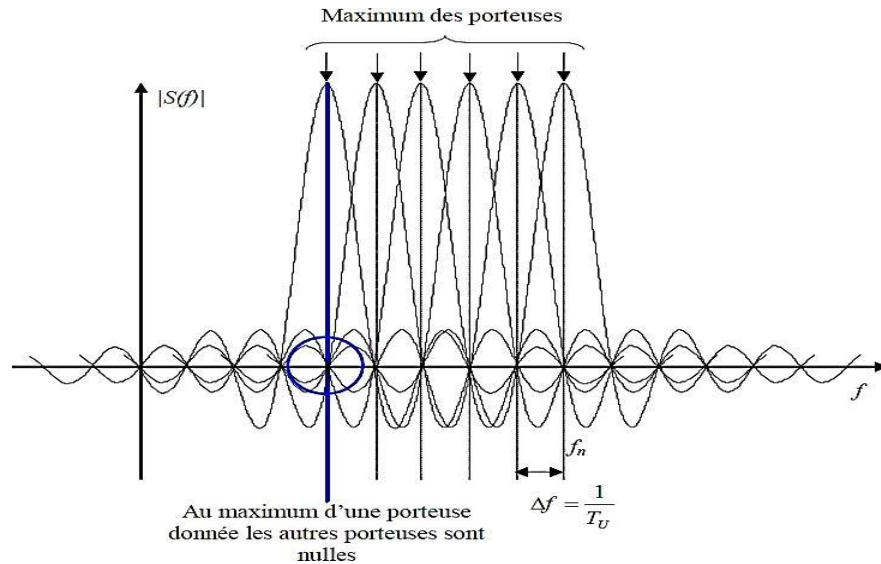


Figure 1-19- Principe d'orthogonalité des sous porteuses en OFDM.

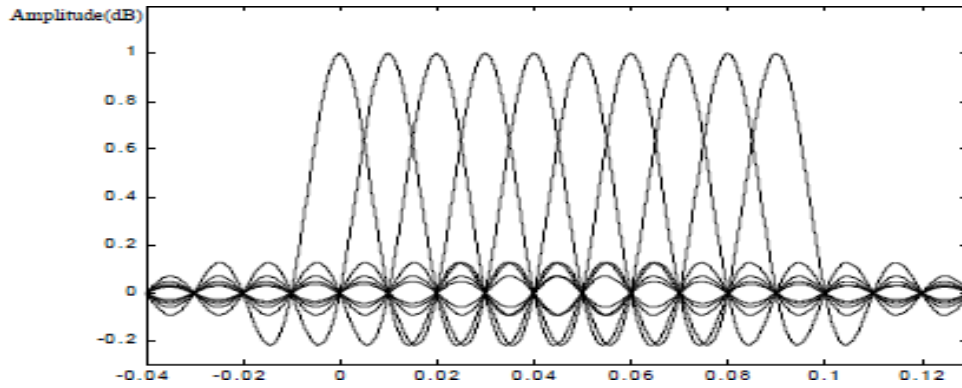


Figure 1-20- Spectres des différentes porteuses.

Tous les symboles numériques C_k sont envoyés pendant la durée symbole T_s , donc le spectre total est la somme des spectres individuels comme indiqué dans la figure (I-21).

La figure (I-21) montre qu'alors, la bande de fréquence est occupée de façon optimum, puisque le spectre est presque plat dans cette bande. La bande occupée est à peu près = N/s (en excluant les lobes secondaires de part et d'autre de la bande).

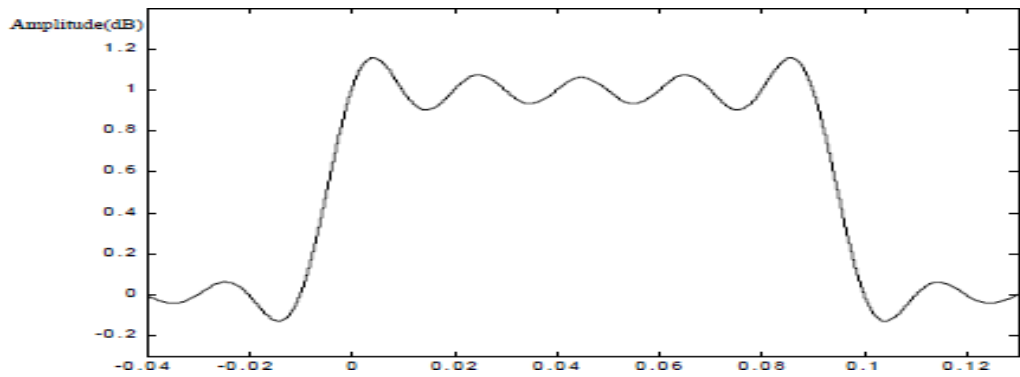


Figure 1-21- Spectre du signal OFDM pour 10 porteuses

3

Dans un environnement à trajets multiples, un symbole transmis prend différents retards pour arriver au récepteur par différents chemins de propagation. Du point de vue du récepteur, le canal présente une dispersion temporelle dans laquelle la durée du symbole reçu est étalée. Prolonger la durée de symbole fait chevaucher le symbole reçu courant avec les symboles reçus précédemment [26], ce qui donne naissance à l'interférence entre symboles (IES).

Plusieurs mécanismes sont donc présents dans une transmission OFDM pour réduire les erreurs. L'intervalle de garde réduit les interférences entre deux symboles OFDM, dues aux trajets multiples. L'intervalle de garde est un délai introduit entre la transmission de deux symboles OFDM consécutifs afin d'absorber l'étalement des retards dus aux trajets multiple [27], dont la

durée T_g doit être supérieure au retard maximum des signaux issus des trajets indirects. L'insertion de ce préfixe est présentée par la figure 1.10. La partie utile de durée T_u de chaque symbole OFDM ne sera alors pas affectée par l'ISI.

Après l'insertion de l'intervalle de garde, l'espacement entre les sous-porteuses reste égal à $\Delta f = 1/T_u$, alors que la durée des symboles OFDM est augmentée à T entraînant une perte d'orthogonalité entre les sous-porteuses. Cette orthogonalité peut être restaurée en réception sous réserve que durant le fenêtrage rectangulaire de durée T_u sur laquelle est appliquée la FFT, le nombre de périodes de chacun des signaux sinusoïdaux composant le signal OFDM soit entier.

Il existe deux techniques permettant de restaurer l'orthogonalité entre les sous-porteuses en réception. La première, appelée (préfixe cyclique : CP-OFDM) consiste à ajouter de la redondance au signal temporel à émettre, on place dans cet intervalle de garde une copie de la fin du symbole OFDM à transmettre [27] et la seconde, appelée (Zero Padding ZP-OFDM) consiste quant à elle à insérer des échantillons de valeur nulle entre les symboles OFDM [8].

En pratique on choisit pour la taille de cet intervalle de garde une durée de l'ordre du quart de celle d'un symbole OFDM, ce qui représente un bon compromis entre diminution des erreurs et perte de débit utile.

1.8.5.3 L'interférence entre porteuses (IEP)

Dans les systèmes OFDM, les spectres des sous-porteuses se recouvrent mais demeurent orthogonaux entre eux. Ceci signifie qu'au maximum de chaque spectre de sous-porteuse, tous les spectres des autres sous-porteuses sont nuls [28]. L'interférence entre porteuses (IEP) est causée par la présence des symboles de données d'une sous-porteuse sur les sous-porteuses adjacentes. L'IEP se produit aussi quand le canal à trajets multiples change pendant la durée d'un symbole OFDM [29]. Quand ceci se produit, les effets Doppler sur chaque trajet causent un décalage de fréquence, ayant pour résultat la perte d'orthogonalité.

Si le préfixe inséré au début d'une trame OFDM est muet (sans aucun signal), des interférences inter porteuses vont se produire. Pour expliquer ce phénomène, il est beaucoup plus facile de raisonner dans le domaine fréquentiel plutôt que dans le domaine temporel, non échantillonné. Prenons donc l'exemple d'une transmission OFDM à N sous-porteuses à travers un canal à deux trajets, dont le retard du trajet indirect est δ , inférieur à la longueur du préfixe T_g . La durée d'une trame OFDM sans son préfixe est, comme précédemment notée T_u . Observons sur la figure 1.22. les chronogrammes de deux "voies" particulières correspondant aux sous-porteuses de fréquences respectives f_k et f_{k+1}

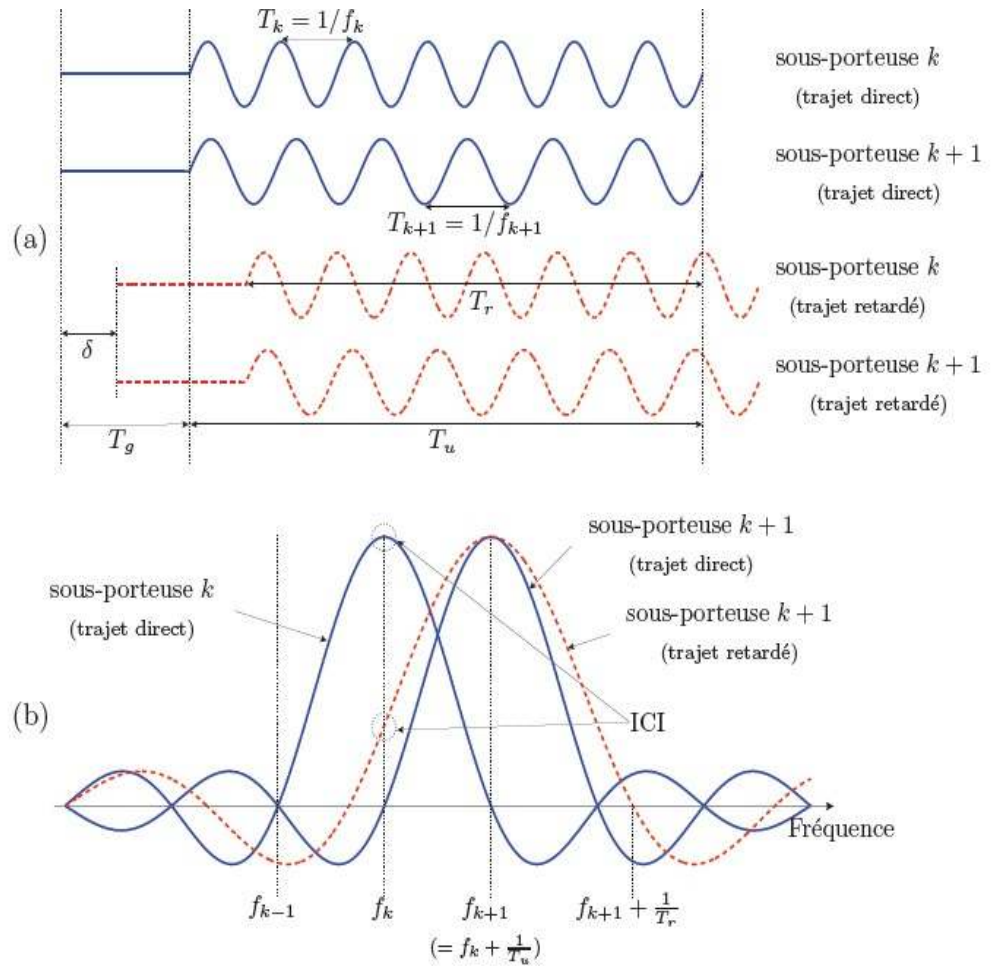


Figure 1-22- Interférence inter-porteuse (ICI) en OFDM dans les domaines

(a) temporel et (b) fréquentiel dans le cas d'un canal à deux trajets [8, 17]

Dans la figure 1.22(a), les signaux k et $k + 1$ issus soit du trajet direct, soit du trajet réfléchi sont représentés en fonction du temps. Il est important de noter que le décalage δ dû au retard de trajet, modifie notablement l'allure du signal dans la fenêtre d'observation, de largeur T_u , liée à la référence d'horloge, puisque la sinusoïde n'est présente que sur une durée T_r [8, 28]... En réception, après suppression du préfixe, on réalise la FFT sur la durée T_u de la trame OFDM, correspondant à la fenêtre visualisée sur la figure 1.22 (a)

Pour le trajet direct, la transformée de Fourier d'une sinusoïde de fréquence f_k , convoluée par la fonction porte de largeur T_u correspondra à un sinus cardinal s'annulant aux fréquences $f_k \pm 1/T_u$ comme cité au paravent. Il en est de même pour la sous-porteuse f_{k+1} . Pour le trajet indirect, les signaux ayant subi une ou plusieurs réflexions, donc décalés dans le temps, la sinusoïde n'est présente que sur une durée $T_r < T_u$. Ceci entraînera une modification de la fonction caractérisant le contenu spectral de puissance du signal, dont les passages

par zéro se produiront donc pour des valeurs différentes de celles associées au trajet direct. Les diverses courbes de la figure 1.22 (b) mettent clairement ce problème en évidence. Lors de l'échantillonnage, il n'y aura plus d'orthogonalité entre les sous-porteuses et on retrouvera des informations d'une sous-porteuse sur l'autre.

Afin d'éviter ces interférences, le préfixe ne doit pas être muet, mais être la copie des L derniers symboles de la trame OFDM. On parle dans ce cas de préfixe cyclique. L'avantage de cette copie est que chaque signal, issu d'un trajet multiple, possèdera toujours un nombre entier de sinusoides sur la durée T_u [28].

Dans le domaine fréquentiel et grâce au préfixe cyclique, la sommation des signaux de la sous-porteuse fk issus des divers trajets ne détruit donc pas l'orthogonalité des sous-porteuses, mais introduit seulement un déphasage. La valeur de L est choisie de telle façon que la durée des L symboles soit supérieure au retard maximum entre trajets.

1.8.5.4 L'insertion de l'intervalle de garde

Dans le cas d'une propagation sur un canal à trajet multiples, de nombreuses répliques de l'onde émise sont reçues avec des amplitudes et des retards différents. Il en résulte de l'interférence entre les symboles reçus ISI. Les techniques de modulation classiques transmettant sur de tels canaux sont très sensibles à ce type d'interférences qui sont d'ailleurs d'autant plus importantes que la durée d'un symbole est petite par rapport à l'étalement des retards du canal (Fig. 1.23 (a)). En d'autres termes, la fiabilité de la transmission est favorisée si la durée des symboles utiles transmis est grande par rapport à l'étalement maximum des retards du canal (Fig. 1.23 (b)). Il existe donc un compromis à trouver entre le débit lié à la durée du symbole et la fiabilité de la liaison liée à l'interférence ISI. Les modulations à porteuses multiples apportent une solution intéressante à l'optimisation de ce compromis.

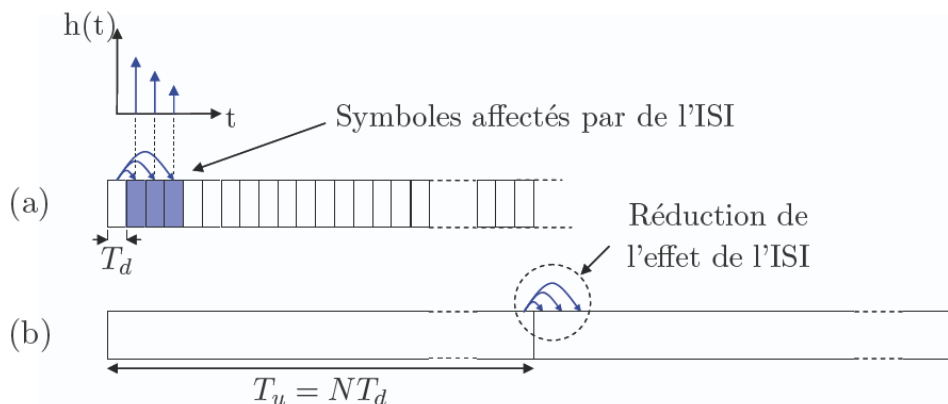


Figure 1-23- Effets du canal à trajets multiples sur des symboles reçus dans le cas

(a) mono-porteuse et (b) multi-porteuses [8]

Avec :

T : la durée de la partie utile de chaque symbole OFDM qui n'est pas affectée par l'ISI

$1/T_d$: le débit de la modulation mono-porteuse initiale.

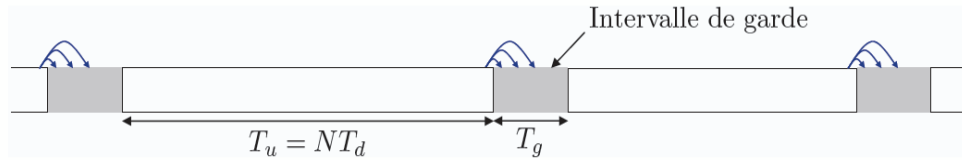


Figure 1-24- Illustration de l'intervalle de garde précédant chaque symbole OFDM

Avec :

T_g : la durée de l'intervalle de garde.

$T_s = T_u + T_g$: la durée de chaque symbole OFDM

Les perturbations du canal de propagation induisent, entre autres, la perte d'orthogonalité entre les sous-porteuses et l'apparition d'interférences entre symboles, dues aux trajets multiples. Afin d'éliminer ces interférences, une solution simple consiste à accroître le nombre N de sous-porteuses pour augmenter la durée symbole T_s . Cependant cette technique se heurte à différentes contraintes. Le temps de cohérence du canal, l'effet Doppler ou les contraintes technologiques, tel que le bruit de phase des oscillateurs, limitent l'emploi de cette technique. Une autre technique permet d'annuler ces ISI. En effet, l'ajout d'un intervalle de garde d'une durée T_g , supérieure ou égale à l'étalement τ_{\max} de la réponse impulsionnelle du canal, précédant le symbole OFDM à émettre permet de supprimer ces interférences. Dès lors la partie utile T_s de chaque symbole OFDM ne sera plus affectée par les IES. La durée totale T_{tot} du symbole OFDM se voit donc augmentée et devient égale à $T_g + T_s$. La mise en œuvre de cette technique conduit donc à une perte en efficacité spectrale n_g et en puissance I_g . Ces pertes peuvent s'exprimer comme suit [30]:

$$n_g = \frac{T_g}{T_g + T_s} \quad (1.27)$$

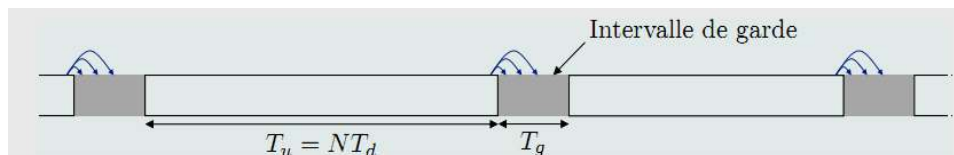


Figure 1-25- Effet de l'intervalle de garde précédant chaque symbole OFDM

$$I_g = 10 \log \left(\frac{T_g}{T_g + T_s} \right) \quad (1.28)$$

En supposant que T_g est égale à 25% de T_s , la perte en efficacité spectrale est de 20%. L'insertion de l'intervalle de garde, se fait au début du symbole OFDM et est une copie de la fin de ce même symbole. Cette solution permet de s'affranchir des termes d'ICI pour Inter-Carrier Interférence. En effet, le choix d'un intervalle de garde nul annulerait l'IES. Néanmoins, en présence de trajets multiples, le nombre de périodes des répliques retardées de chacune des sous-porteuses contenues dans la partie utile T_s de chaque symbole OFDM n'est plus entier. Par conséquent, ce phénomène provoque un élargissement du spectre des sous-porteuses correspondantes et l'apparition d'ICI, induites par la perte d'orthogonalité entre ces sous-porteuses. En réception, la suppression de l'intervalle de garde permet de restituer l'orthogonalité entre les sous-porteuses. De plus, comme l'intervalle de garde est la recopie des échantillons de fin de symbole OFDM, cet intervalle peut également être exploité en réception pour la synchronisation temporelle du signal OFDM.

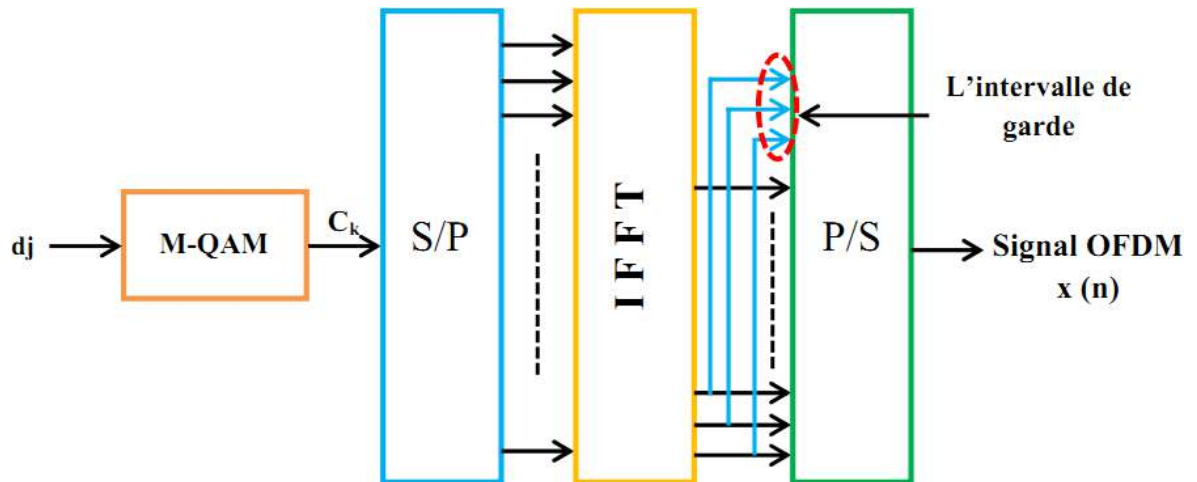


Figure 1-26- Schéma synoptique des modulations OFDM

1.9 LA CHAÎNE DE TRANSMISSION OFDM

Le synoptique de la figure (I.28) illustre les différents modules qui composent la chaîne de transmission OFDM. Le modulateur M-QAM transforme les données binaires *bide* durée T_b en symboles complexes X_k de durée $T_q = \log_2 M T_b$, où M est la taille de la constellation de la modulation QAM utilisée.

Le convertisseur série-parallèle dispose les symboles X_k en groupes (trames) de N symboles, la durée d'une trame T_u est N fois plus grande que la durée d'un symbole en série T_q . Par conséquent, l'effet de canal devient moins nuisible. En appliquant ensuite une transformée de

Fourier inverse, on obtient la trame (symbole) OFDM.

L'IFFT est utilisée afin de transformer le spectre du signal OFDM au domaine temporel pour la transmission à travers le canal. Un préfixe cyclique de durée T_g copie les N_g derniers symboles de la trame OFDM et les ajoute ensuite au début de la trame.

Après conversion parallèle-série, on obtient enfin le symbole OFDM qui contient $N_s = N + N_g$ symboles de durée totale $T_s = T_u + T_g$ que l'on transmet à travers un canal à évanouissements de Rayleigh.

À la réception, les opérations inverses sont réalisées, commençant par la suppression du préfixe cyclique. La décomposition spectrale des échantillons reçus calculée en utilisant l'algorithme FFT, et enfin la démodulation pour retrouver les données binaires transmises [32], [33],[34].

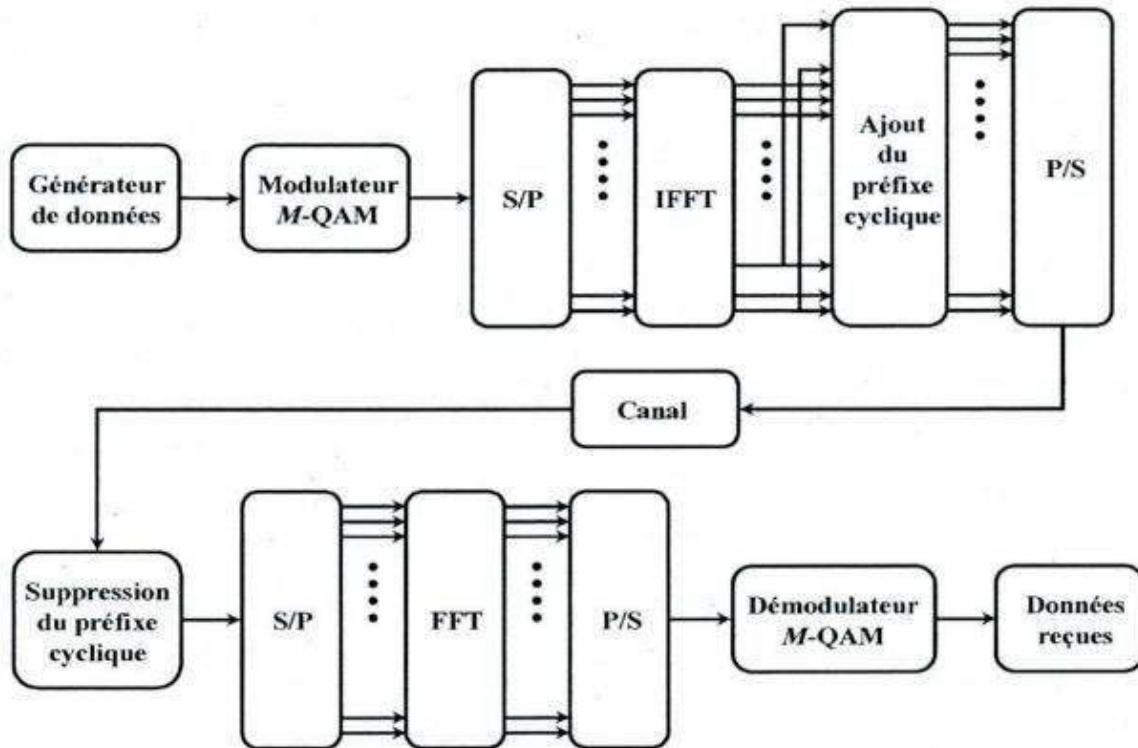


Figure 1-27- Synoptique d'une chaîne de transmission OFDM

1.10 AVANTAGES ET LIMITES DE L'OFDM

Les nouvelles technologies se basant sur les modulations multi-porteuses orthogonales, présentent des avantages ainsi que des inconvénients.

1.10.1 AVANTAGES DE L'OFDM

Les avantages de l'OFDM sont nombreux :

- l'utilisation de la bande de fréquence allouée est optimale par orthogonalité des porteuses (Une haute efficacité spectrale).
- la modulation est basée sur un algorithme bien connu et peu complexe : la FFT.
- un codage et entrelacement adapté permettent d'améliorer la qualité de la transmission des données.

l'OFDM permet une égalisation simple grâce à l'ajout du "préfixe cyclique" ou du "zéro padding", même en présence de canaux multi trajets denses.

-La diminution des taux de transmission et l'ajout de préfixes cycliques permettent d'éliminer ou de limiter l'interférence inter symboles et de simplifier l'égalisation au récepteur.

-Les effets des parcours multiples dû aux évanouissements sélectifs en fréquence sont réduits en divisant le spectre en N sous porteuses ayant des évanouissements plats.

-Le chevauchement en fréquence des sous porteuses permet de conserver une grande efficacité spectrale.

- Réduction de la complexité des récepteurs due à la possibilité d'éviter les ISI et ICI par insertion d'un intervalle de garde.

La robustesse des signaux OFDM aux canaux sélectifs en fréquence représente l'avantage principal de cette modulation. En effet, d'un point de vue fréquentiel, cette technique divise un canal large bande sélectif en fréquence en plusieurs sous-canaux à bande étroite non sélectifs avec une orthogonalité entre canaux très simples à égaliser. On peut tirer profit de la diversité fréquentielle en privilégiant les bonnes sous porteuses. Le principe du waterfilling (ou power loading) est alors utilisé : les sous porteuses qui ont un gain trop faible ne reçoivent pas de puissance.

1. Les techniques multi-porteuses sont robustes au bruit impulsif puisque chaque sous-porteuse est affectée d'un bruit indépendant des autres sous-porteuses. Contrairement à la modulation mono-porteuse où le bruit peut affecter un certain nombre de symboles transmis, la perte d'un symbole dû à un bruit important n'affecte pas les autres symboles.

2. Une égalisation numérique et un décodage simple et optimal grâce à l'utilisation de l'intervalle de garde (au prix d'une diminution du débit). De plus, l'utilisation de différents systèmes de codage correcteur d'erreur associés à un entrelacement entre fréquences permet d'atteindre les performances d'un canal sans écho.

Enfin, la modulation est réalisée par une transformée de Fourier inverse et la démodulation via une simple transformée directe.

1.10.2 LIMITES DE L'OFDM

La modulation OFDM n'a pas que des avantages, mais elle a aussi des inconvénients :

- ✓ Un des principaux inconvénients est que les signaux OFDM ont une forte fluctuation d'enveloppe qui est caractérisée par le PAPR élevé en comparaison avec les modulations monoporteuse. Un PAPR élevé rend les signaux OFDM très sensibles aux non-linéarités des composants analogiques, en particulier celles de l'amplificateur de puissance [3],[35]. Pourtant, pour des rendements élevés, les amplificateurs de puissance doivent fonctionner dans une zone dite non linéaire (ou de saturation), malheureusement, c'est dans cette zone que se présentent les non linéarités qui créent des distorsions des signaux à transmettre. Ces effets sont d'autant plus gênants quand les signaux à amplifier sont à PAPR élevés.
- ✓ L'intervalle de garde induit une perte d'efficacité spectrale.
- ✓ L'OFDM est également très vulnérable aux problèmes de synchronisation. Les erreurs de synchronisation induisent un déphasage sur les symboles reçus. Les techniques de compensation qui existent pour les modulations mono-porteuses sont mal adaptées aux modulations multi-porteuses et de nouvelles approches sont à l'étude.
- ✓ L'OFDM est également très délicate aux problèmes de décalage en fréquence (Frequency offset). Dans ce cas, le "Frequency offset" est dû aux différences de la fréquence entre les oscillateurs locaux de l'émission et de la réception, et engendre ainsi de l'interférence entre sous-porteuses qui peut détruire l'orthogonalité des sous-porteuses [36].
- ✓ Malgré ses nombreux avantages, la performance de l'OFDM est beaucoup moins satisfaisante dans un scénario de communication à grande mobilité, où l'effet Doppler joue un rôle important. Dans ce cas, les techniques traditionnelles, qui sont utilisées avec succès pour l'estimation de canal ou l'égalisation dans un environnement statique, fonctionneront de manière très dégradée [2].
- ✓ L'OFDM conventionnelle, utilise une forme d'onde rectangulaire parfaitement localisée en temps mais mal localisée en fréquence. Elle n'a pas été adoptée dans les communications radio mobiles vu sa sensibilité à la sélectivité temporelle (variations très rapides en temps) du canal de propagation, entraînée par le mouvement des stations mobiles. Le désir actuel d'utiliser cette technique dans les systèmes radio mobiles de futures générations a motivé la recherche de nouvelles formes d'onde bienlocalisées en temps et en fréquence. L'OFDM/OQAM et l'OFDM suréchantillonnées sont des techniques qui autorisent ce type de forme d'onde [5].

1.11 APPLICATIONS DE L'OFDM

Les modulations multi-porteuses sont maintenant utilisées dans diverses applications à haut débit, que ce soit en bande de base sur paire torsadée l'ADSL (Asymmetric Digital Subscriber Line) dont l'application principale est l'internet haut débit ou sur onde porteuse pour les transmissions sans fil. L'OFDM est utilisée dans la plupart des standards de communication [3] et [13].

En 1980, lorsque le projet de radiodiffusion numérique DAB (Digital Audio Broadcasting) fut lancé, il fut démontré que les modulations OFDM pouvaient garantir les performances désirées pour ce système [1],[2], [4], [6], [8].

En 1991, l'ETSI (European Telecommunications Standards Institute) retient l'OFDM comme modulation standard pour le DAB [3],[4] et [15].

En 1995, dans le domaine de la télévision numérique terrestre, le standard DVB-T (Digital Video Broadcasting - Terrestrial) s'appuie sur la modulation OFDM avec un codage de canal (COFDM) [1],[2], [4], [6], [8].

En 1996, la société TELIA proposa une interface radio basée sur l'OFDM pour les systèmes de communication mobile UMTS (Universal Mobile Telecommunications System).

Entre 1999 et 2001, on voyait apparaître les standards WLAN (Wireless Local Area Network), comme l'IEEE 802.11a/g nommé Wi-Fi qui adoptaient la modulation OFDM comme spécification principale de leur couche physique. Wi-Fi (IEEE802.11b) fournit jusqu'à 11 Mb/s de données, et plus récemment IEEE802.11a/g qui peut aller jusqu'à 54 Mb/s. L'équivalent des réseaux locaux sans fil IEEE802.11a pour les Etats-Unis est le HiperLAN/2 pour l'Europe [2-4], [6], [8].

En 2004, le projet LTE (Long Term Evolution), avait comme objectif de définir les spécifications techniques de la future norme de réseau mobile de 4G, utilise l'OFDM pour une meilleure flexibilité dans l'utilisation du spectre.

En 2005, un nouveau standard basé sur l'OFDM vit le jour ; il s'agit de l'IEEE 802.16 nommé Wi-Max [3]. Ce système garantit un débit théorique jusqu'à 80Mbps et une portée de 50km. Mobile Broadband Wireless Access (MBWA) IEEE802.20 est l'extension du standard IEEE802.16 pour les environnements mobiles.

En 2006, le procédé de modulation OFDM a été adopté pour les communications à très haut débit (480Mbps) et à courte portée (10m), basées sur la technologie Ultra Wide Band.

OFDM est aussi considéré dans l'IEEE 802.11n qui se base sur les systèmes MIMO (Multiple Input/Multiple Output) dont plusieurs antennes sont utilisées pour un multiplexage spatial.

En outre, l'OFDM a aussi récemment été considérée dans le cadre de communications acoustiques sous-marines : ce type de canal est en effet particulièrement sélectif, ce qui fait de l'OFDM une solution potentiellement intéressante.

1.12 CONCLUSIONS

Le principe de la technique OFDM consiste à répartir un flux de données à haut débit sur plusieurs flux à faible débit. Ces derniers sont transmis simultanément sur des sous-porteuses orthogonales. La somme de ces sous-porteuses constitue le signal OFDM transmis. Le signal transmis se propage dans un canal à trajets multiples et subit des distorsions. A la réception, des versions décalées du même signal sont reçues avec des interférences entre symboles OFDM. Pour éliminer cette interférence, un préfixe cyclique (CP) de durée supérieure à l'étalement maximal des retards du canal est ajouté au début de chaque symbole OFDM à l'émission. En réception, les opérations inverses sont réalisées ainsi que les opérations d'estimation et d'égalisation du canal

Dans ce chapitre, nous avons décrit et caractérisé le signal OFDM. Les systèmes multi-porteuses permettent de surmonter efficacement les dégradations introduites par le canal comme la sélectivité en fréquence.

Grace aux progrès dans la fabrication des circuits numériques, la réalisation du système OFDM devient possible. Bien que cette dernière ait été choisie comme la couche physique standard pour une diversité de systèmes importants, la théorie, les algorithmes, et techniques de sa mise en œuvre reste les sujets d'intérêt actuel.

2 ETAT DE L'ART FFT

2.1 INTRODUCTION

La transformée de Fourier rapide (FFT) est une méthode efficace pour calculer la transformée de Fourier discrète (DFT). Dans ce contexte la transformée de Fourier discrète d'un signal dans le domaine temporel $x(n)$ est donnée par l'équation (2.1).

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k = 0,1,2, \dots, N-1 \quad (2.1)$$

Où $W_N^{nk} = e^{-j2\pi(\frac{kn}{N})}$ est appelé le facteur twiddle.

C'est une valeur complexe. Le calcul direct d'une DFT de N points nécessite $O(N^2)$ opérations, alors que l'algorithme FFT nécessite approximativement un nombre total de multiplications complexes : $O((N/2) \log_2(N))$ [37]. En fait, l'algorithme FFT utilise l'approche diviser et conquérir pour réduire les calculs [38]. L'idée est de diviser le problème en sous-problèmes avec la relation suivante :

$$\sum \text{Coût (sous - problèmes)} + \text{Coût (découpage en sous - problèmes)} \\ < \text{Coût (probleme original)}$$

La méthode de division de la DFT en sous-problèmes classe les algorithmes FFT en deux familles : Cooley-Tukey et algorithmes PFA (Prime-factor algorithm). La méthode Cooley-Tukey est une méthode simple qui convient à tous les N puissance de 2 si l'on utilise des méthodes de base [39].

Chaque type d'algorithme est en outre classé en fonction d'autres caractéristiques : utilisation de mémoire supplémentaire, choix du type de décimation (Décimation En Temps) ,(Décimation En fréquence), etc.

Dans l'algorithme de Cooley-Tukey Radix- k , la DFT de N points est subdivisée en deux DFT de $(N/k \text{ points})$ qui est ensuite divisée en deux DFTs plus petites jusqu'à obtenir une DFT de taille deux points, incluant les structures papillons qui sont juste des additions et des soustractions de nombres complexes de manière récursive. En fait, c'est le meilleur algorithme spécialement adapté pour un nombre N puissance de 2 (Radix-2).

Des algorithmes de Radix supérieur à 2 peuvent être utilisés pour faciliter l'exploitation de la multiplication complexe. D'un autre côté, ils génèrent une complexité algorithmique qui impose une structure papillon plus compliquée [40]. Ainsi, un nouvel algorithme appelé Split Radix Algorithme est adopté pour tirer avantage des algorithmes Radix-2 et de Radix-4 afin d'obtenir un minimum de multiplications complexes avec la tenue d'une structure de papillon simple.

D'autres algorithmes ont vu le jour et seront exposés brièvement à la fin de ce chapitre par souci de complétude, bien que leur intérêt pour la suite de ce travail de thèse ne soit pas de premier plan.

2.2 ETAT DE L'ART DES ALGORITHMES FFT

2.2.1 INTRODUCTION

Dans cette partie, nous commençons par un aperçu de la DFT qui est utilisée pour produire l'analyse de fréquence des signaux périodiques non discrets. Par la suite, nous présentons la FFT comme une autre méthode pour atteindre le même résultat de la DFT tout en apportant moins de complexité de calculs

2.2.2 LA TRANSFORMÉE DE FOURIER DISCRÈTE

Un signal numérique dans le temps peut être transformé dans le domaine fréquentiel au moyen de la transformée en Z ou de la transformée de Fourier [15].

De manière générale, un signal périodique $x(t)$ peut être représenté sous la forme d'une somme de signaux sinusoïdaux, selon l'expression (2.3) appelée également transformée de Fourier inverse

$$x(t) = \int (X(f) e^{2\pi jft} df) \quad (2.2)$$

Où $X(f)$ est un nombre complexe définissant l'amplitude et la phase de chaque composante de fréquence. Le terme $X(f)$ constitue la transformée de Fourier du signal temporel éventuellement complexe, sa définition est donnée par l'équation (2.3) :

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-2\pi jft} dt \quad (2.3)$$

L'expression (2.8) correspond à la forme continue de la transformée de Fourier. Le calcul de cette dernière sur une série échantillonnée conduit alors à la définition de la DFT noté $X(k)$. Ainsi, pour une série $x(n)$ constituée d'un nombre N échantillons, de valeurs finies, l'expression de la DFT d'un signal $x(n)$ est définie dans l'équation (2.4) [15]

La TFD d'une suite temporelle d'échantillons $x(n)$ est donnée par la formule :

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad \forall k \in \mathbb{Z} \quad 0 \leq k \leq N - 1 \quad (2.4)$$

avec la notation suivante :

$$W_N^{kn} = e^{-2\pi j \frac{kn}{N}}$$

- n et k Représentent respectivement les variables discrètes et normalisées de l'espace original (temporel) et de l'espace transformé (fréquences).
- N est une puissance de 2, qui représente le nombre des valeurs discrètes et successives des variables n et k . L'IDFT qui effectue l'opération inverse, c'est-à-dire qu'elle convertit le spectre fréquentiel $X(f)$ dans le domaine temporel échantillonné $x(n)$, est donnée par l'équation (2.10) :

$$x(t) = \frac{1}{N} \sum_{n=0}^{N-1} X(K) e^{2\pi j \frac{kn}{N}} \quad (2.5)$$

- $x(n)$ Représente le vecteur temporel
- $X(k)$ Désigne le vecteur fréquentiel
- N Représente la longueur de la DFT/IDFT, et $1/N$ est un facteur de pondération utilisé dans l'expression de la IDFT afin d'obtenir les mêmes coefficients que dans la décomposition en série de Fourier.
- Le facteur $W_N^{kn} = e^{-2\pi j \frac{kn}{N}}$ (2.6)

est nommé facteur de phase ou coefficient de transformation (en anglais: Twiddle Factor). C'est le coefficient complexe utilisé pour combiner les résultats d'une étape précédente afin de former l'entrée de l'étape suivante. Ce facteur correspond à une coordonnée sur le cercle unitaire complexe, tel que représenté par la figure 2.8 dans le cas $N = 8$.

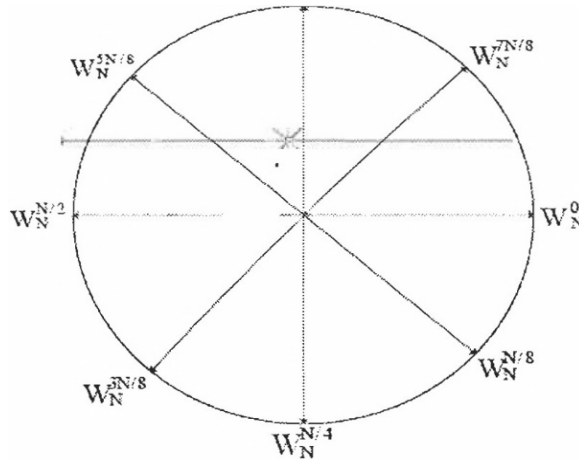


Figure 2-1- Les coefficients de la DFT pour N=8

2.2.3 LA TRANSFORMÉE RAPIDE DE FOURIER

La Transformée de Fourier Rapide (notée par la suite FFT) est simplement une TFD calculée selon un algorithme permettant de réduire le nombre d'opérations et, en particulier, le

nombre de multiplications à effectuer. Il faut noter cependant, que la réduction du nombre d'opérations arithmétiques à effectuer, n'est pas synonyme de réduction du temps d'exécution. Tout dépend de l'architecture du processeur qui exécute le traitement.

Pour calculer une TFD, on doit calculer N valeurs $X(k)$:

$$X(K) = \sum_{n=0}^{n=N-1} x(n) e^{-2\pi j \frac{kn}{N}}$$

Si on effectue le calcul directement sans algorithme efficace, on doit effectuer :

$$\begin{aligned} N^2 & \text{ Multiplications complexes} \\ N(N-1) & \text{ Additions complexes} \end{aligned}$$

La complexité globale est donc de $O(N^2)$.

Il existe différents algorithmes de FFT qui réduisent significativement les nombres d'opérations arithmétiques.

De nombreux travaux ont essayé de réduire la complexité de la DFT. En effet, Cooley et Tukey introduisent en 1965 le premier algorithme de la FFT qui permet de réduire considérablement le temps de calcul de la DFT d'une suite dont le nombre d'échantillons N est décomposable en facteurs, typiquement une puissance de 2, connu sous le nom de Radix-2. Cette publication allait engendrer de nombreuses recherches sur les algorithmes de calculs des transformées. Depuis, et encore de nos jours, les algorithmes de FFT ont révolutionné le traitement numérique de signal (Digital Signal Processing : DSP).

Leur efficacité réside dans la réduction du nombre d'opérations nécessaires, en particulier le nombre de multiplications, ce qui réduit davantage le temps d'exécution. Tous ces algorithmes sont basés sur un même principe qui consiste à décomposer le calcul de la DFT en plusieurs sous ensemble de DFT de longueur plus petite. La mise en œuvre de ce principe conduit à différentes méthodes qui profitent toutes des propriétés de symétrie et de périodicité des facteurs de phases (Twiddles) W_N^k suivant les équations (2.7) et (2.8).

$$\begin{aligned} \text{Symétrie :} \quad W_N^{k+N/2} &= -W_N^k & (2.7) \\ (W_N^{kn})^* &= W_N^{-k} \end{aligned}$$

$$\text{Périodicité : } W_N^{k+N} = W_N^k \quad (2.8)$$

Où (*) désigne le complexe conjugué.

Les algorithmes de FFT permettent donc de faire une DFT de manière efficace ainsi que de réduire la charge de calcul en termes de multiplications et additions complexe à l'ordre $O(N \log_2 N)$. En effet, il existe de nombreux algorithmes FFT qui découlent tous de l'approche

diviser pour régner (en anglais: Divide-And-Conquer Approach). L'idée de cette approche consiste à représenter les vecteurs $X[k]$ et $x(n)$ sur deux dimensions (cas du Radix-2 ou bien plusieurs dimensions (cas du Radix-2ⁱ, Radix-4ⁱ)).

2.2.4 APPROCHE DIVISER-POUR-RÉGNER

L'algorithme de la FFT et ses variantes sont détaillés dans de nombreux ouvrages, notamment (Proakis et Manolakis) [15]. Le principe de base de l'algorithme consiste à faire un changement de variables de l'indice n sur deux dimensions entières. Par exemple, avec M pour les colonnes et L pour les lignes, on a :

$$N = ML \tag{2.9}$$

$$n = Ml + m; \quad 0 < p < L - 1, \text{ et } 0 < m < M - 1 \tag{2.10}$$

$$k = Mp + q; \quad 0 < p < L - 1, \text{ et } 0 < q < M - 1 \tag{2.11}$$

Avec ce changement de variable, les points de séquence à transformer seront représentés sous forme matricielle. Ainsi, l'équation de la DFT devient :

$$X[p, q] = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} X[l, m] W^{(Mp+q)(Ml+1)} \tag{2.12}$$

Le terme $W^{(Mp+q)(Ml+1)}$ peut être simplifié de la manière suivante :

$$W_N^{(Mp+q)(Ml+1)} = W_N^{MLmp} W_N^{mLq} W_N^{MPl} W_N^{lq} \tag{2.13}$$

Toutefois, on a

$$W^{(MLmp)} = 1; \quad W_N^{mlq} = W_N^{mq} = W_{N/2}^{mq} \text{ et } W_N^{MLp} = W_L^{lp} = W_{N/M}^{lp};$$

De là on obtient l'équation :

$$X[p, q] = \sum_{l=0}^{L-1} \left\{ W_N^{lq} \sum_{m=0}^{M-1} X(l, m) W_M^{mq} \right\} W_L^{lp} \tag{2.14}$$

L'équation (2.21) illustre le fonctionnement de la FFT. Soit la décomposition de la FFT à N Points en L FFT à M points. Pour résoudre l'équation (2.14), nous procédons d'abord à des TFD de M points correspondant à la sommation à l'intérieur entre crochets. Ensuite, la matrice résultante subit une multiplication point à point par les facteurs de phase de phase (Twiddles). Enfin des TFD sur L points sont effectuées en suivant l'autre dimension. En effectuant ces étapes, la complexité passe de l'ordre N^2 à l'ordre $N(M + L)$.

Nous pouvons aussi récursivement faire d'autres changements de variables, en prenant un nombre premier pour L . Par la suite, nous factorisons M jusqu' à obtenir seulement des TFD ayant des tailles de nombre premier [15].

L'algorithme diviser pour régner peut se résumer dans les étapes suivantes [15]:

1. Calcul de M-point DFT, $F(l, q)$ de chaque ligne selon l'équation (2.15).

$$F(l, q) = \sum_{m=0}^{M-1} X(l, m) X(l, m) W_M^{mq} \quad (2.15)$$

2. Multiplier le tableau résultant par le facteur de phase W_N^{lq} , on obtient $G(l, q)$
 Défini par :

$$G(l, q) = W_N^{lq} F(l, q) \quad (2.16)$$

3. Finalement, calcul de L-points DFT de chaque colonne selon l'équation (2.17):

$$X(p, q) = \sum_{l=0}^{L-1} G(l, q) W_L^{lp} \quad (2.17)$$

L'algorithme peut aussi être représenté graphiquement, en utilisant un exemple d'une FFT à 15 points, avec $m = 3$ et $l = 5$. La figure 2.2 explique la méthodologie utilisée :

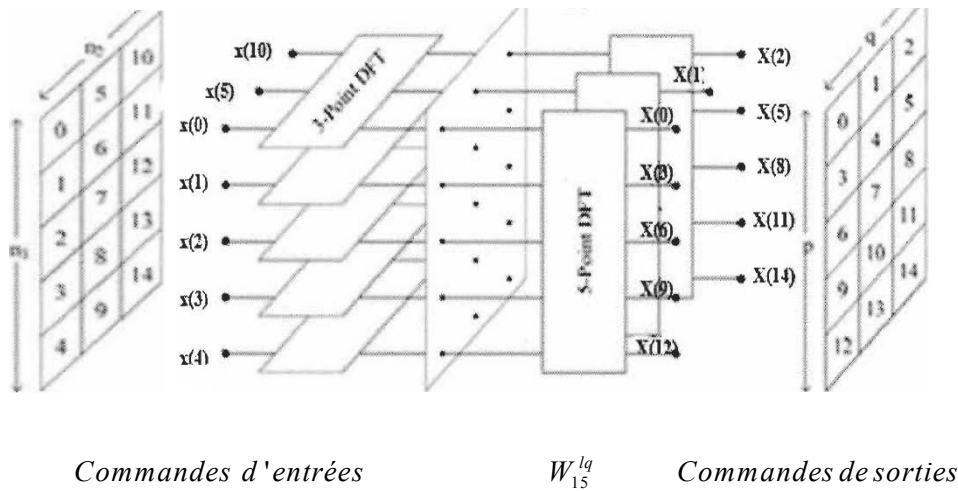


Figure 2-2- Décomposition d'une FFT 15 points en deux FFT 3 points et 5 points [15]

2.3 CLASSIFICATION DES ALGORITHMES FFT

Pendant plusieurs décennies, la principale préoccupation des chercheurs était de développer un algorithme FFT qui minimise le nombre d'opérations requises. Depuis que les deux Ingénieurs d'IBM, Cooley et Tukey, ont présenté leur approche montrant que le nombre de multiplications requis pour calculer la DFT peut être réduit significativement en exploitant les propriétés de la figure 2.5 l'intérêt a surgi à la fois pour chercher des applications pour cette puissante transformée et trouver diverses implémentations logicielles et matérielles pour la FFT.

Il existe des algorithmes FFT qui calculent la DFT plus rapidement et plus efficacement et réduisent significativement les tailles des puces où ils seront implémentés telles que les DSP (Digital Signal Processing) et les FPGA (Field Programmable Gate Array). L'exécution de ces algorithmes en technologie d'intégration à très grande échelle représente un problème non trivial quand il s'agit de respecter les contraintes de l'application en termes de consommation

de puissance, coût d'implémentation et vitesse de calcul. D'où, la nécessité d'augmenter les performances de ces algorithmes en trouvant une solution matérielle.

Tous les algorithmes cités dans ce chapitre ont été élaborés pour réduire la charge de calcul en termes de multiplications et d'additions complexes et chacun d'entre eux possède ces propres caractéristiques. Ces méthodes d'algorithmes rapides sont toutes basées sur une approche appelée 'diviser pour régner'. Comme précédemment présenté, le principe de cette méthode est de diviser un grand problème en petits sous-problèmes facile à résoudre. Dans le cas de la FFT, la division en sous-problèmes signifie que les données d'entrée $x(n)$ sont divisées en sous-ensembles sur lesquels la DFT est calculée. Ensuite, à partir des résultats intermédiaires, la DFT des données initiales est reconstruite. Si cette stratégie est appliquée récursivement sur les DFT intermédiaires, on obtient un algorithme FFT performant [14], [15].

Les algorithmes les plus connus et les plus utilisés sont les algorithmes FFT où N est une puissance entière de deux, $N = 2^M$, où M est un entier. Grâce à ces algorithmes, il est possible de réduire le nombre d'opérations nécessaires à un ordre de grandeur de $N \log_2 N = NxM$ [15]. Le tableau 1 donne une comparaison du nombre d'opérations dans un calcul direct de la DFT et dans la FFT [15]

| N | N^2 | $N \log_2 N$ | $N / \log_2 N$ |
|------|----------|--------------|----------------|
| 2 | 4 | 2 | 2.00 |
| 4 | 16 | 8 | 2.00 |
| 8 | 64 | 24 | 2.67 |
| 16 | 256 | 64 | 4.00 |
| 32 | 1024 | 160 | 6.40 |
| 64 | 4096 | 384 | 10.67 |
| 128 | 16384 | 896 | 18.29 |
| 256 | 65536 | 2048 | 32.00 |
| 512 | 262144 | 4608 | 56.89 |
| 1024 | 1048576 | 10240 | 102.40 |
| 2048 | 4 194304 | 22528 | 186.18 |

Tableau 1 - Comparaison du nombre d'opérations dans un calcul direct de la DFT et dans la FFT

La dernière colonne montre de combien la vitesse de calcul peut être augmentée par l'utilisation de la FFT au lieu du calcul direct de DFT [15].

Dans d'autres littératures, on emploie le mot Radix-r ou Base-b, qui signifient le nombre d'entrée de chaque sous-systèmes DFT qui compose l'algorithme FFT. Toutefois les algorithmes sont nommés Radix-r. Par exemple si $r = 2$, l'algorithme s'appellera Radix-2 où $N = r^M = 2^M$.

Dans la section suivante, nous verrons de plus près, les trois algorithmes conventionnels FFT les plus communs et les plus utilisés : Radix-2, Radix-4 et Radix-8.

2.4 RADIX 2

L'algorithme FFT à base de Radix-2 est un des algorithmes possédant la structure Butterfly la plus simple pour le calcul de la DFT. Radix-2 qui signifie en français radical-2 ou base-2, veut que la taille N de la FFT à calculer soit à base de 2. Sa structure est très simple, appelée structure Butterfly (papillon) à cause de son schéma en forme de papillon. Il existe deux algorithmes le DIT FFT (Decimation In Time, Décimation dans le temps) et DIF FFT (Decimation In Frequency, Décimation en fréquence). Dans ce travail nous nous intéresserons uniquement à la version DIT car elle présente, selon [CHA08], une meilleure précision en virgule fixe que la version DIF.

De plus, la suite de notre travail de thèse repose sur la décimation en temps. Nous prendrons donc soin de présenter cet algorithme dans le détail.

Le Radix-2 utilise la technique 'diviser pour régner'. Donc à l'aide de cette technique divise la FFT en sous-système de $N/2$ points, puis calcule chaque sous-système tout seul pour obtenir à la fin les composantes du spectre fréquentiel du signal.

La démonstration ci-dessous montre la méthode 'diviser pour régner' utilisée dans l'algorithme Radix-2 ainsi que la méthode appliquée pour obtenir la structure en papillons. Prenons un signal discret $x[n]$ de longueur N (N est pair). Définissons alors deux nouveaux signaux $x_1[n]$ et $x_2[n]$ chacun de longueur $N/2$ et qui sont constitués par les échantillons pairs et impairs de $x[n]$ respectivement :

$$\left. \begin{array}{l} x_1[n] = x[2n] \\ x_2[n] = x[2n + 1] \end{array} \right\} \text{avec } n = 0, 1, \dots, (N/2) - 1 \quad (2.18)$$

Notons $W_N^{kn} = e^{-2\pi j \frac{kn}{N}}$

En utilisant les coefficients W_N , on trouve comme DFT de $x[n]$:

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk} = \sum_{n=0(n \text{ pair})}^{N-1} x[n]W_N^{nk} + \sum_{n=0(n \text{ impair})}^{N-1} x[n]W_N^{nk}$$

$$X[k] = \sum_{n=0}^{\frac{N-1}{2}} x[2n]W_N^{2nk} + \sum_{n=0}^{\frac{N-1}{2}} x[2n+1]W_N^{2(n+1)k} \quad (2.19)$$

Et en utilisant les propriétés de symétrie et de périodicité (2.14) et (2.18), on trouve :

$$X[k] = \sum_{n=0}^{\frac{N-1}{2}} x_1[n]W_{N/2}^{2nk} + W_N^k \sum_{n=0}^{\frac{N-1}{2}} x_2[n]W_{N/2}^{nk} \quad (2.20)$$

$$X[k] = X_1[k] + W_N^k X_2[k]$$

$X_1[k]$ et $X_2[k]$ Représentent les TFD à $N/2$ points de $x_1[n]$ et $x_2[n]$ [15].

A partir de l'équation (2.20), on constate que le calcul de $X[k]$ peut se faire à l'aide d'une structure en treillis de la FFT, plus communément appelée 'butterfly diagram' littéralement 'structure en papillon'. Cette dernière représente une opération dans laquelle, partant de deux nombre complexes A et B , on calcule deux nouveaux nombres complexes Y et Z tels que :

$$Y = A + W_N^p B \text{ et } Z = A - W_N^p B \quad (2.21)$$

Avec W_N^p le facteur Twiddle et p un entier compris entre 0 et $N - 1$.

L'équation (2.21) est représentable schématiquement, la figure 2.3 montre son schéma.

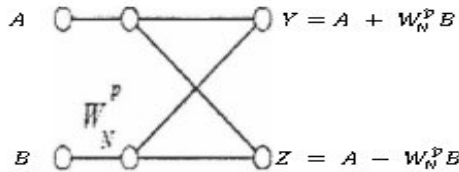


Figure 2-3- structure DIT Butterfly

La figure 2.3 représente le schéma d'une structure Butterfly DIT de la FFT. On remarque que cette structure, avec $p = 0$, représente exactement la relation qui existe entre les échantillons à l'entrée $x[0] = A$ et $x[1] = B$ et les échantillons en sortie $X[0] = Y$ et $X[1] = Z$, dans une DFT à 2 points. La figure 2.4 quant à elle, représente la structure d'une Butterfly DIF.

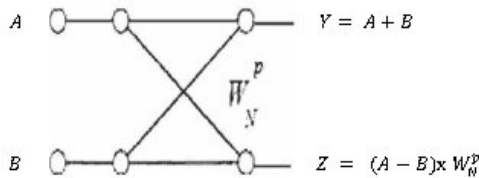


Figure 2-4- structure DIF Butterfly

Ci-dessus les deux structures Butterfly utilisées respectivement pour la DIT-FFT et la DIF FFT. Les Twiddle pour le DIT sont multipliés par l'entrée B puis le produit est, ou ajouté ou retranché, à A . Par contre pour la DIF on additionne les entrées A et B pour la première sortie puis pour la deuxième sortie, on soustrait B de A et on multiplie par le twiddle factor. En général, la structure Butterfly est l'opération de base du Radix- r PE (Processeur Élémentaire, Processor Element) qui compose l'algorithme FFT, où r entrées sont combinées pour donner r sorties via l'opération déjà décrite dans l'équation (2.17). Les figures 2.5 et 2.6 représentent à titre d'exemple les structures des deux algorithmes Radix-2 DIT et DIF pour $N=8$.

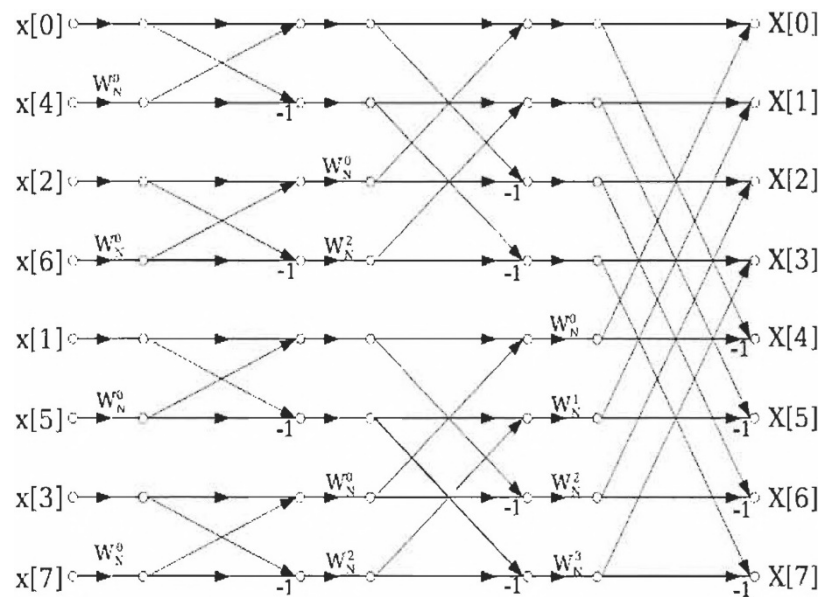


Figure 2-5- algorithme Radix-2 DIT pour $N=8$

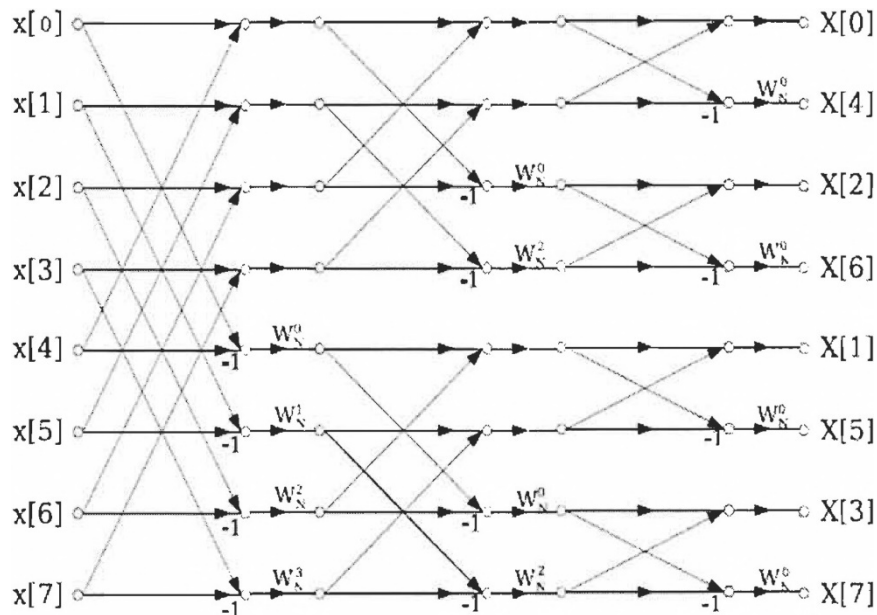


Figure 2-6- algorithme Radix-2 DIF pour N=8

Dans la figure 2.5 ci-dessus, on représente la structure du Radix-2 (a) DIT et 2.6 DIF(b). On constate que la division en petits sous-systèmes, (Butterfly, structures papillons), commence au côté de l'entrée (signal temporel), d'où le nom de l'algorithme «décimation en temps». Le Radix-2 DIF possède les mêmes propriétés que le précédent sauf que la division se fait au niveau de la fréquence (décimation en fréquence). La taille de la FFT est $N=8$. Pour un Radix-2, la FFT se calcule en $\log_2 N$ étages, soit ici en trois étages puisque $N=8$ et on aura $N/2$ Butterfly (sous-systèmes) à calculer par étage, soit 4 pour notre cas ici. On ajoute aussi que le Radix-2, DIT ou DIF, réduit l'ordre de calcul de N^2 à $\frac{N}{2} N \log_2 N$ multiplications complexes et de $N^2 - N$ à $N \log_2 N$ additions complexes.

Le Radix-2 est l'algorithme FFT le plus utilisé actuellement. Il apparaît dans plusieurs articles de l'IEEE grâce à sa simple architecture. Plusieurs travaux sont effectués sur cet algorithme afin de trouver des méthodes plus élaborées pour réduire le nombre de multiplications et d'additions complexes.

2.4.1 FORMULATION MATRICIELLE DE L'ALGORITHME COOLEY-TUCKEY

Nous pouvons réécrire l'équation (2.9) sous forme matricielle en faisant ainsi apparaître la Complexité de l'algorithme de la DFT.

La matrice DFT est composée d'exponentielles complexes, elle est définie en notation matricielle comme suit :

$$X[K] = \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N^1 & W_N^2 & \dots & W_N^{(N-1)} \\ 1 & W_N^2 & W_N^4 & \dots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \dots & W_N^{(N-1)(N-1)} \end{bmatrix} \times \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{bmatrix} \quad (2.22)$$

Matrice DFT

[B_n] est la matrice des coefficients (facteurs Twiddles)

L'expression (2.21) peut être représenté sous la forme matricielle suivante :

$$X[K] = [B_n] [X_n] \quad (2.23)$$

Où :

[B_n] est la matrice des coefficients (facteurs Twiddles)

[X_n] est la matrice des entrées de la DFT

X[K] est la matrice des sorties de la DFT

2.4.2 OPERATION BIT REVERSE

On remarque sur le schéma figure 2.5 que les données x(n) en entrée sont désordonnées en raison des décimations répétées dans les différents étages.

La séquence d'entrée naturelle est brouillée et l'ordre de la séquence d'entrée finale est obtenu comme indiqué figure 2.5 et 2.6, alors que celles de sortie X(k) sont dans l'ordre naturel. Rappelons que c'est de ce fait que cet algorithme de FFT s'appelle FFT avec entrelacement temporel.

Notons que dans les 2 cas précédents : FFT avec entrelacement temporel et FFT avec entrelacement fréquentiel, l'ordre entrelacé est obtenu à partir de l'ordre naturel en appliquant une technique dite du « bit reversal ».

Cette technique consiste à écrire en binaire l'indice dans l'ordre naturel puis à retourner l'ordre des bits pour obtenir la représentation binaire de l'indice correspondant dans l'ordre entrelacé.

Le bit reverse se traduit par une organisation selon les bits du poids fort des données d'entrée initialement ordonnées selon le bit du poids faible. Ceci est dû à la propriété des décompositions successives de la FFT en deux sous-transformées.

Il faut bien prendre en compte cette particularité car le réordonnement des données est très coûteux, surtout avec des accès en puissances de deux

| | | | | | | | | |
|--------------------------|------|------|------|------|------|------|------|-----|
| Ordre naturel d'entrée : | 000, | 001, | 010, | 011, | 100, | 101, | 110, | 111 |
| | 0, | 1, | 2, | 3, | 4, | 5, | 6, | 7 |
| Ordre Bit-Reversed : | 000, | 100, | 010, | 110, | 001, | 101, | 011, | 111 |
| | 0, | 4, | 2, | 6, | 1, | 5, | 3, | 7 |

Le tableau 2 résume l'opération de Bit-Reversed :

| Séquence d'entrée | Index | Forme binaire de l'index | Forme Bit reversed de l'index | Représentation décimale | Séquence d'entrée finale |
|-------------------|-------|--------------------------|-------------------------------|-------------------------|--------------------------|
| $x(0)$ | 0 | 000 | 000 | 0 | $X[0]$ |
| $x(1)$ | 1 | 001 | 100 | 1 | $X[4]$ |
| $x(2)$ | 2 | 010 | 010 | 2 | $X[2]$ |
| $x(3)$ | 3 | 011 | 110 | 3 | $X[6]$ |
| $x(4)$ | 4 | 100 | 001 | 4 | $X[1]$ |
| $x(5)$ | 5 | 101 | 101 | 5 | $X[5]$ |
| $x(6)$ | 6 | 110 | 011 | 6 | $X[3]$ |
| $x(7)$ | 7 | 111 | 111 | 7 | $X[7]$ |

Tableau 2- Opération de Bit-Reversed

2.5 RADIX 4

Le paradigme diviser pour régner ne se limite pas à diviser un problème en deux sous-problèmes. En fait comme expliqué, l'équation de récurrence permet de subdiviser un problème en plusieurs sous-problèmes.

L'algorithme de décimation dans le temps radix-4 réorganise l'équation de la transformée de Fourier discrète (DFT) en quatre parties sommes sur tous les groupes de chaque quatrième indice de temps discret $n = [0, 4, 8, \dots, N - 4]$, $n = [1, 5, 9, \dots, N - 3]$, $n = [2, 6, 10, \dots, N - 2]$ et $n = [3, 7, 11, \dots, N - 1]$ (Cela ne fonctionne que lorsque la longueur FFT est un multiple de quatre.)

Tout comme dans la FFT de décimation dans le temps radix-2, une manipulation mathématique supplémentaire montre que la longueur- N DFT peut être calculée comme la somme des sorties de quatre N DFT des échantillons à temps discret à indice pair et à indice impair respectivement où trois d'entre eux sont multipliés par les facteurs Twiddles W_N^k , W_N^{2k} , W_N^{3k} .

Les transformées de Fourier rapides (FFT) de décimation en temps et de décimation en fréquence de Radix-4 gagnent en vitesse en réutilisant les résultats de calculs intermédiaires plus petits pour calculer plusieurs sorties de fréquence DFT.

C'est ce qu'on appelle une décimation dans le temps parce que les échantillons de temps sont réorganisés en groupes alternés et un algorithme radix-4 parce qu'il y a quatre groupes.

L'équation de récurrence est :

$$X_r = \sum_{l=0}^{N-1} x_l w_N^{rl}, r = 0, 1, \dots, N-1 \quad (2.23)$$

Cette équation peut être décomposée comme suit :

$$X_r = \sum_{k=0}^{N/4-1} x_{4k} w_N^{r(4k)} + \sum_{k=0}^{N/4-1} x_{4k+1} w_N^{r(4k+1)} + \sum_{k=0}^{N/4-1} x_{4k+2} w_N^{r(4k+2)} + \sum_{k=0}^{N/4-1} x_{4k+3} w_N^{r(4k+3)} \quad (2.24)$$

$$= \sum_{k=0}^{N/4-1} x_{4k} w_N^{r(4k)} + w_N^r \sum_{k=0}^{N/4-1} x_{4k+1} w_N^{r(4k)} + w_N^{2r} \sum_{k=0}^{N/4-1} x_{4k+2} w_N^{r(4k)} + w_N^{3r} \sum_{k=0}^{N/4-1} x_{4k+3} w_N^{r(4k)} \quad (2.25)$$

Par conséquent, les quatre sous-problèmes ci-dessus peuvent être écrits comme

$$Y_r = \sum_{k=0}^{N/4-1} x_{4k} w_N^{r(4k)} = \sum_{k=0}^{N/4-1} x_{4k} (w_N^4)^{rk} = \sum_{k=0}^{N/4-1} y_k w_{N/4}^{rk}, r = 0, 1, \dots, N/4-1$$

Et de façon similaire

$$Z_r = \dots \quad (2.26)$$

$$G_r = \dots$$

$$H_r = \sum_{k=0}^{N/4-1} h_{4k+3} w_N^{r(4k)} = \sum_{k=0}^{N/4-1} x_{4k+3} (w_N^4)^{rk} = \sum_{k=0}^{N/4-1} h_k w_{N/4}^{rk}, r = 0, 1, \dots, N/4-1$$

Maintenant la taille de chaque sous-problème est $N/4$ ce qui est égal à la taille d'entrée. Le sous-problème ci-dessus peut être résolu récursivement pour obtenir la solution du problème principal avec une taille de $N/4$ au vu de la périodicité suivante :

$$Y_r = Y_{r+n/4} = Y_{r+n/2} = Y_{r+3n/4}$$

Le sous-problème ci-dessus peut être écrit sous la forme :

$$\begin{aligned} X_r &= Y_r + w_N^r Z_r + w_N^{2r} G_r + w_N^{3r} H_r \\ X_{r+N/4} &= Y_r + w_N^{r+N/4} Z_r + w_N^{2(r+N/4)} G_r + w_N^{3(r+N/4)} H_r \\ X_{r+N/2} &= Y_r + w_N^{r+N/2} Z_r + w_N^{2(r+N/2)} G_r + w_N^{3(r+N/2)} H_r \\ X_{r+3N/4} &= Y_r + w_N^{r+3N/4} Z_r + w_N^{2(r+3N/4)} G_r + w_N^{3(r+3N/4)} H_r \end{aligned} \quad (2.27)$$

En notant :

$$w_N^{N/4} = w_4 = e^{-j2\pi/4} = -j, w_N^{N/2} = (-j)^2 = -1$$

et 2.62

$$w_N^{3N/4} = (-j)^3 = j$$

Les quatre équations de 2.61 peuvent s'écrire finalement sous la forme :

$$\begin{aligned} X_r &= Y_r + w_N^r Z_r + w_N^{2r} G_r + w_N^{3r} H_r \\ X_{r+N/4} &= Y_r - j w_N^r Z_r - w_N^{2r} G_r + j w_N^{3r} H_r \\ X_{r+N/2} &= Y_r - w_N^r Z_r + w_N^{2r} G_r - w_N^{3r} H_r \\ X_{r+3N/4} &= Y_r + j w_N^r Z_r - w_N^{2r} G_r - j w_N^{3r} H_r \end{aligned} \quad (2.28)$$

Pour analyser le coût arithmétique, nous devons prendre en compte que $w_N^{2r} G_r, w_N^r Z_r, w_N^{3r} H_r$ sont calculés avant les sommes partielles.

Une analyse plus poussée conclut à un coût de 25 pour cent moindre que le coût de $T(N) = N \log_2 N$ d'une FFT Radix-2.

Le Radix-4 joue un rôle important dans la suite de notre travail. Il servira de base pour expliciter l'algorithme proposé par notre approche, bien que le Radix-8 serve aussi de support à une implémentation sur cette base. Il convient de souligner dès à présent que le Radix- r utilisé ($r = 2,4,8..$) n'a aucune importance particulière pour l'implémentation de l'algorithme, qui est implémentable quelque soit le Radix utilisé.

Le Radix-4 est un algorithme FFT intéressant puisqu'il réduit le nombre de multiplications complexes de 25 % par rapport au Radix-2, [15] quand la taille de la FFT est une puissance de 4. La matrice des twiddle est donnée ci-dessous.

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

Les éléments de la matrice R sont obtenus par une méthode géométrique, la figure 2.8 ci-dessous donnent les éléments de R sur le cercle unité.

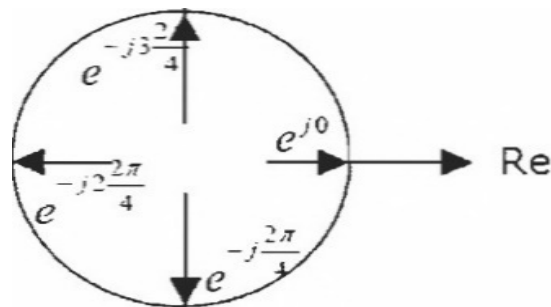


Figure 2-7- éléments de R sur le cercle unité

Les twiddle factor sont calculés à partir du cercle trigonométrique de la figure 2.7, ils prennent donc les valeurs $(1, -j, -1, j)$.

2.6 RADIX 8

La plupart des implémentations des algorithmes 'Cooley et Tukey' utilisent des versions d'algorithmes Radix-2 ou Radix-4, bien que les Radix supérieurs comme le Radix-8 ont plusieurs avantages évidents, par exemple, moins d'accès mémoire, et moins de multiplications complexes, voir ci-dessous le tableau 3. La raison est sans doute que l'implémentation de Radix supérieurs, exige une grande complexité du Processeur Élémentaire (PE) et impose une restriction sur la longueur N de la transformée, d'où la nécessité de s'intéresser de plus

près à ces Radix supérieurs pour les rendre moins complexes donc plus facile à implémenter.

| Opérations | Radix-2 | Radix4 | Radix-8 |
|---------------------------|---------|--------|---------|
| Multiplications complexes | 22528 | 15360 | 10752 |
| Multiplications réelles | 0 | 0 | 8192 |
| Additions complexes | 49152 | 49152 | 49152 |
| Additions réelles | 0 | 0 | 8192 |
| Accès mémoire | 49152 | 24576 | 16384 |

Tableau 3- Nombre d'opérations d'une FFT de 4096 points pour différents Radix

2.7 SPLIT-RADIX FFT (SRFFT)

L'algorithme Split-Radix FFT (SRFFT) a été clairement décrit et nommé pour la première fois en 1984 par Duhamel et Hollman. Cet algorithme utilise moins d'opérations arithmétiques (multiplications et d'additions) que n'importe quel autre algorithme de puissance 2. L'algorithme SRFFT est une variante de l'algorithme Cooley-Tukey, puisqu'il utilise un mélange du Radix-2 et du Radix-4. La figure 2.8 exprime de manière récursive une DFT de longueur N , en fonction d'une DFT plus petites de longueur $N/2$ et deux petites DFT de longueur $N/4$ [15].

La figure 2.8 représente la structure du Butterfly Split-Radix

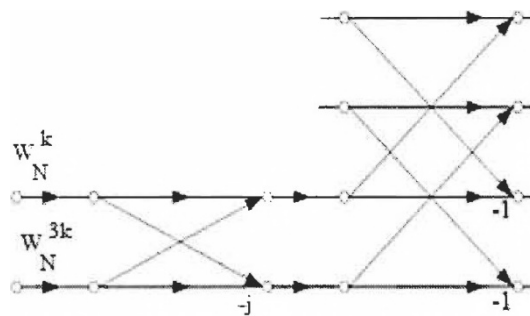


Figure 2-8-Structure Butterfly du Split-Radix DIT

Grace à sa structure mixte, le SRFFT réutilise les twiddle communs entre le Radix-2 et la Radix-4. Ainsi, le nombre de multiplications non-triviales diminue et des fois est remplacé par des additions. En effectuant un mélange approprié des équations régissant le Radix-2 et le Radix-4, on arrive à obtenir un algorithme FFT utilisant moins d'opérations arithmétiques.

2.8 AUTRES VERSIONS FFT

Il existe d'autres méthodes publiées dans la littérature, que nous ne considérerons pas dans notre travail, et mentionnées ici uniquement par souci de complétude.

2.8.1 PFA (PRIME FACTOR ALGORITHM) [GOOD-THOMAS]:

Pour une taille $n = n_1 n_2$, avec des nombres premiers entre eux n_1 et n_2 , il est possible d'utiliser l'algorithme PFA (Good-Thomas) basé sur le théorème des restes chinois. Le PFA est similaire à celui de Cooley-Tukey.

2.8.2 RADER-BRENNER

L'algorithme de Rader-Brenner est aussi une variante de Cooley-Tukey avec des facteurs twiddles purement imaginaires qui améliorent les performances en réduisant le nombre de multiplications mais au détriment de la stabilité numérique et une augmentation du nombre d'additions.

2.8.3 BRUNN'S ALGORITHM:

Les algorithmes qui procèdent aussi par des factorisations successives sont ceux de Bruun et l'algorithme QFT. Les versions originales travaillent sur des fenêtres dont la taille est une puissance de deux mais il est possible de les adapter pour une taille quelconque. L'algorithme de Bruun considère la transformée de Fourier rapide comme une factorisation récursive du polynôme $z^n - 1$ en des polynômes avec des coefficients réels de la forme $z^m - 1$ et $z^{2m} + az^m + 1$.

2.8.4 ALGORITHME DE WINOGRAD

L'algorithme de Winograd factorise $z^n - 1$ en un polynôme cyclotomique, dont les coefficients sont souvent -1, 0 ou 1 ce qui réduit le nombre de multiplications. On peut voir cet algorithme comme la version optimale en termes de multiplications. Winograd a montré que la transformée de Fourier discrète peut être calculée avec seulement $O(n)$ multiplications, ce qui représente une borne inférieure atteignable pour les tailles qui sont des puissances de deux. Toutefois, des additions supplémentaires sont nécessaires ce qui peut être pénalisant sur les processeurs modernes comportant des unités arithmétiques performantes.

2.8.5 ALGORITHME DE RADER

L'algorithme de Rader est quant à lui destiné aux fenêtres dont la taille est un nombre premier. Il profite de l'existence d'une génératrice pour le groupe multiplicatif modulo

n . La transformation discrète dont la taille est un nombre premier s'exprime ainsi comme une convolution cyclique d'une taille $n - 1$. On peut ensuite la calculer par une paire de transformation de Fourier rapide.

2.8.6 ALGORITHME DE BLUESTEIN

Finalement, un autre algorithme destiné aux transformations avec des tailles qui sont des nombres premiers est due à Bluestein. On l'appelle plus souvent l'algorithme chirp-z. Ici encore, la transformation est vue comme une convolution dont la taille est identique à la fenêtre originale. On utilise à cet effet l'identité $jk = - (j - k)^2 / 2 + j^2 / 2 + k^2 / 2$

2.9 ARCHITECTURES FFT

Il existe quatre types d'architecture FFT disponibles dans la littérature :

1. Architectures basées sur des mémoires (Memory Based Architectures)
2. Architectures basées sur des mémoires cache (Cached memory Architectures)
3. Architectures en Tableaux ou architectures cellulaires (Array Architectures)
4. Architectures en pipeline (Pipelined Architectures)

2.9.1 ARCHITECTURES BASEES SUR DES MEMOIRES (MEMORY-BASED ARCHITECTURES)



Figure 2-9- Architecture à mémoire unique

Souvent, les algorithmes FFT basés sur des mémoires sont utilisés dans les étages où les opérations de lecture et d'écriture sont réalisées en accédant à la mémoire [41].

Les architectures basées sur des mémoires sont classées en architecture à mémoire unique et architecture à mémoire duale [42]. Dans une architecture à mémoire unique, l'élément de traitement (PE) échange des informations de données avec une mémoire unique via un bus bidirectionnel d'au moins N mots comme illustré à la Fig 2.9.

De la même manière la Fig.2.10 montre l'architecture à mémoire duale où deux mémoires de taille N chacune sont attachées à l'élément de traitement (PE) avec deux bus de données bi-directionnels distincts. Le flux de données entrant d'une mémoire est transmis à travers l'élément de traitement vers une autre mémoire et inversement jusqu'à ce que la transformation soit parfaitement complétée.

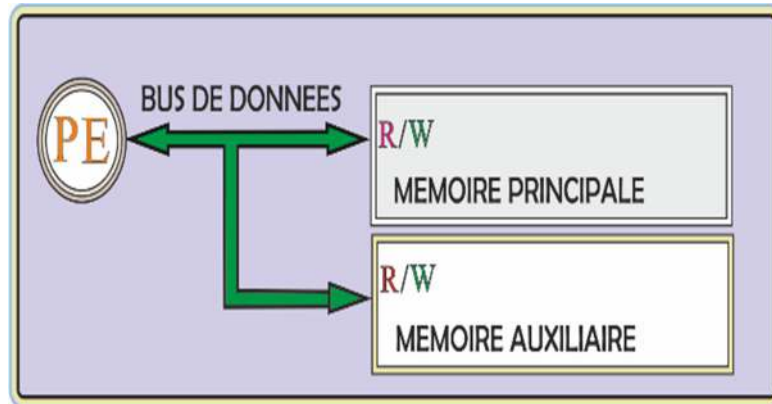


Figure 2-10- Architecture basée sur une mémoire duale

De toute évidence, le débit d'entrée peut ne pas être égal à l'horloge du processeur FFT de sorte que les données présentes en entrée doivent passer par trois phases : le bloc tampon d'entrée, l'unité de calcul et le mécanisme de sortie.

D'un point de vue pratique, les données à traiter sont d'abord stockées dans la première mémoire d'entrée jusqu'à ce que les N échantillons soient empilés. Ensuite cette mémoire est utilisée comme mémoire de calcul et l'élément de traitement (PE) y accède via son signal de commande R / W. Dans le même temps, un autre bloc de mémoire joue le rôle de tampon d'entrée pour stocker un autre ensemble de séquences de données en entrée. L'élément de traitement (PE) prend un certain temps pour effectuer les calculs et stocker les données intermédiaires dans la mémoire de calcul. Lorsque la transformation est terminée, la mémoire dite de calcul sert de tampon de sortie.

Fondamentalement, les architectures basées sur la mémoire nécessitent $(N / k) \log_k(N)$ accès mémoire pour calculer une FFT Radix-k de N points où k mots sont lus et écrits depuis et vers le circuit de mémoire à chaque accès. Le signal d'horloge doit être exprimé en tant que $\log_k(N/k)$ fois la fréquence de la séquence des données en entrée car un seul élément de traitement contrôle l'évolution des calculs.

2.9.2 ARCHITECTURES BASEES SUR DES MEMOIRES CACHE (CACHED MEMORY ARCHITECTURES)

Au lieu des topologies antérieures, certains auteurs préconisent l'utilisation de l'architecture à mémoire cache [43], comme le montre la Fig.2.11. Ainsi il comporte une zone cache de données sur l'élément processeur pour augmenter la vitesse d'accès à la mémoire et ainsi améliorer l'efficacité énergétique. Il est similaire à celui de l'architecture à mémoire unique à l'exception du cache entre le processeur et la mémoire principale par lequel l'étape de pré-extraction des données est bien réalisée. Ce type d'architecture n'est pas largement utilisé en raison de la complexité du contrôleur et du hardware supplémentaire [46].

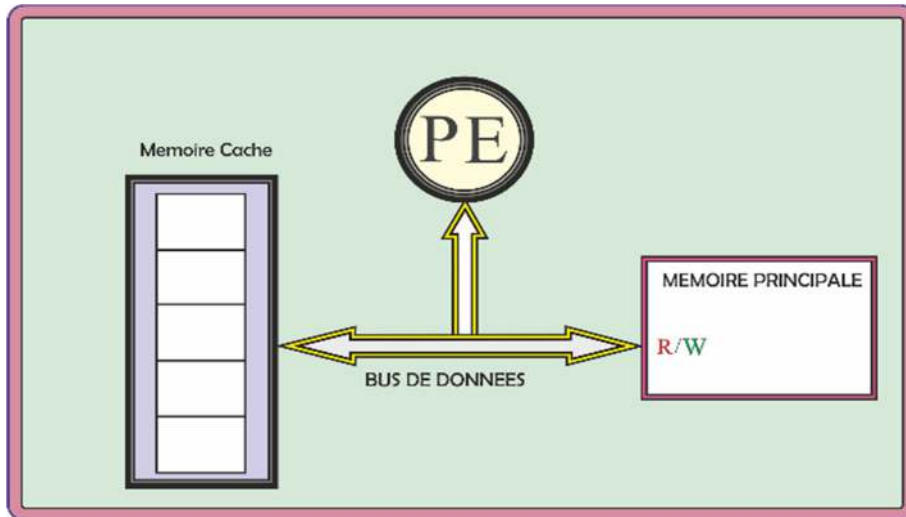


Figure 2-11- Architecture Mémoire Cache

2.9.3 ARCHITECTURES EN TABLEAUX (CELLULAIRES) (ARRAY ARCHITECTURES)

Dans les architectures matricielles [44] un certain nombre d'éléments de traitement sont interconnectés avec des buffers locaux en mode réseau comme le montre la Fig. 2.12 pour effectuer les calculs FFT. Ces structures ne sont pas non plus largement adoptées en raison des énormes besoins en surface et d'autres aménagements [46].

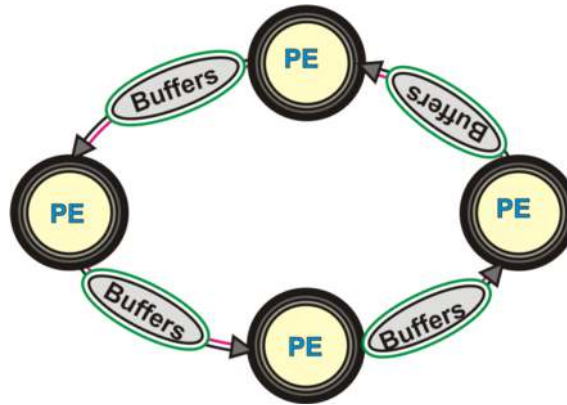


Figure 2-12- Architecture en Tableaux pour les calculs FFT

2.9.4 ARCHITECTURES EN PIPELINE (PIPELINED ARCHITECTURES)

Une autre alternative basée sur les architectures FFT en pipeline est possible. Ces dispositifs sont des architectures rapides et à haut débit avec parallélisme et pipelining [45]. Néanmoins la complexité du matériel est élevée et moins flexible que celle des autres architectures. Ils offrent un haut débit et une consommation efficace lors de la mise en œuvre [46].

Nous présentons ici certaines architectures en pipeline couramment utilisées telles que le MDC (Multi-Path Delay Commutator) et le SDF (Single-Path Delay Feedback) [46].

2.9.5 MULTI-PATH DELAY COMMUTATOR

Dans ce type d'architecture, la séquence de données entrantes est d'abord subdivisée en plusieurs flux de données parallèles par un commutateur, puis l'opération papillon est effectuée via une multiplication par un twiddle, avec des délais appropriés pour chaque flux de données.

Dans le radix-2 MDC (R2MDC), le flux de données d'entrée est divisé en deux branches parallèles comme indiqué sur la Fig. 2.13 pour N égal à 16.

Au total, $(\log_2(N) - 1)$ multiplieurs complexes, $\log_2(N)$ unités papillons et $2 \left(\frac{3N}{4} - 1 \right)$ tampons de retard spécifique. Fig. 2.14 illustre l'architecture MDC (R4-MDC) de radix-4, dans laquelle quatre tampons de données parallèles sont nécessaires. Dans ce contexte, tous les composants d'unités et de multiplieurs de papillon pourraient servir à 100 % avec les ensembles sont traités à la fois. Un total de $(\log_4(N) + 3)$ multiplieurs complexes, $\log_4(N)$ unités papillons radix-4 et $(2N + 4)$ tampons ou mots de mémoire sont requis [46].

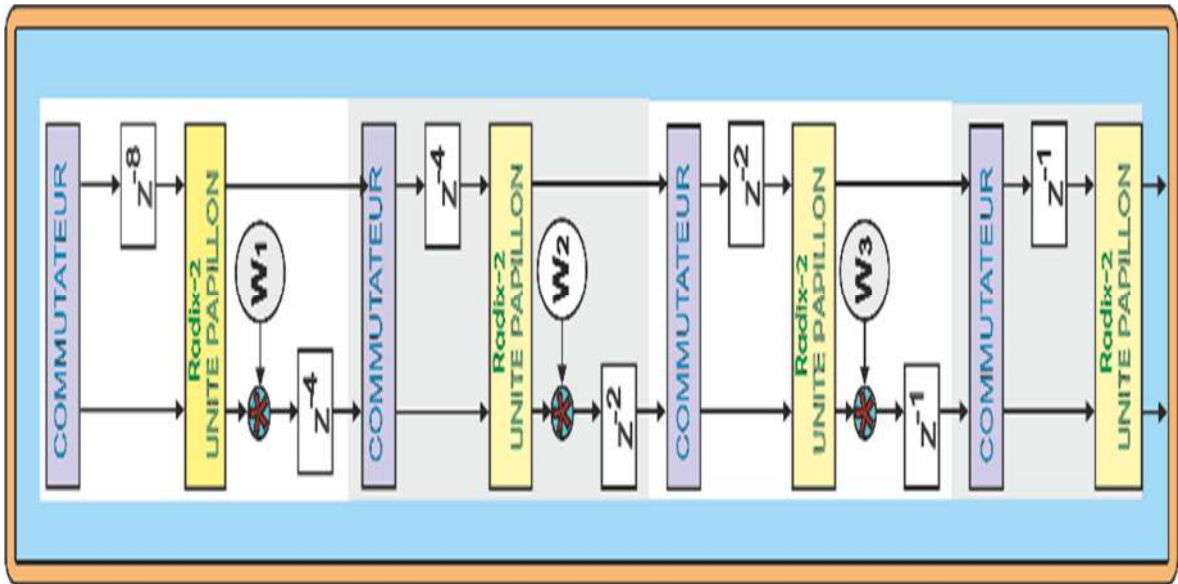


Figure 2-13- Architecture Radix-2 Multipath Delay Commutator FFT pour 16-points

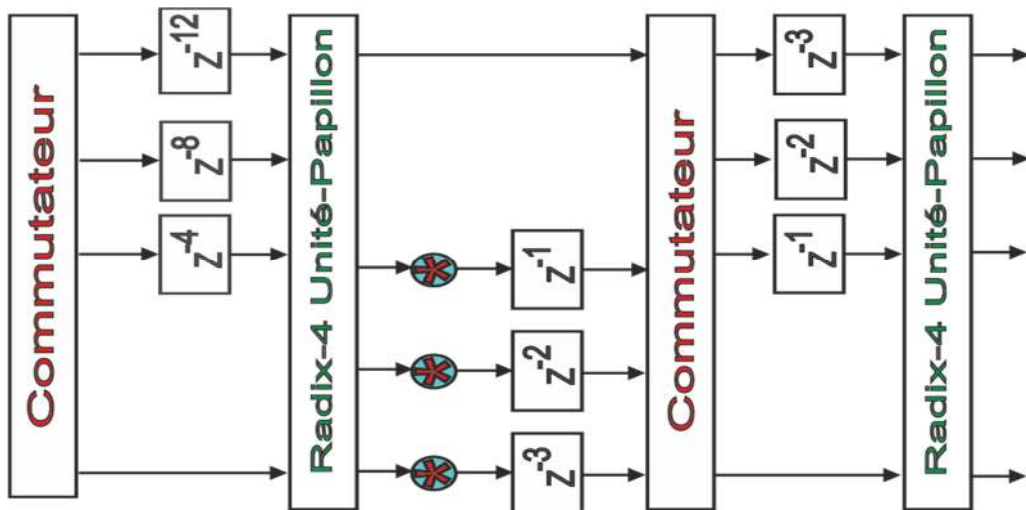


Figure 2-14- Architecture Radix-4 Multipath Delay Commutator FFT pour 16-points

Dans une architecture de type Single Path Delay Feedback, une séquence de données entrantes fait l'objet d'une opération de multiplication avant d'être dirigée vers l'étage suivant. Dans ce cas, les unités de retard sont plus efficacement utilisées en partageant le même espace mémoire entre les entrées, les sorties et les unités papillons. Les architectures Radix-2 et Radix-4 SDF sont indiquées dans la Fig. 2.13 et dans la Fig. 2.14 respectivement. Dans ce cas, il est important de noter que l'utilisation des multiplicateurs et des unités de papillon est de 50 % parce qu'ils sont contournés durant la moitié du temps [46].

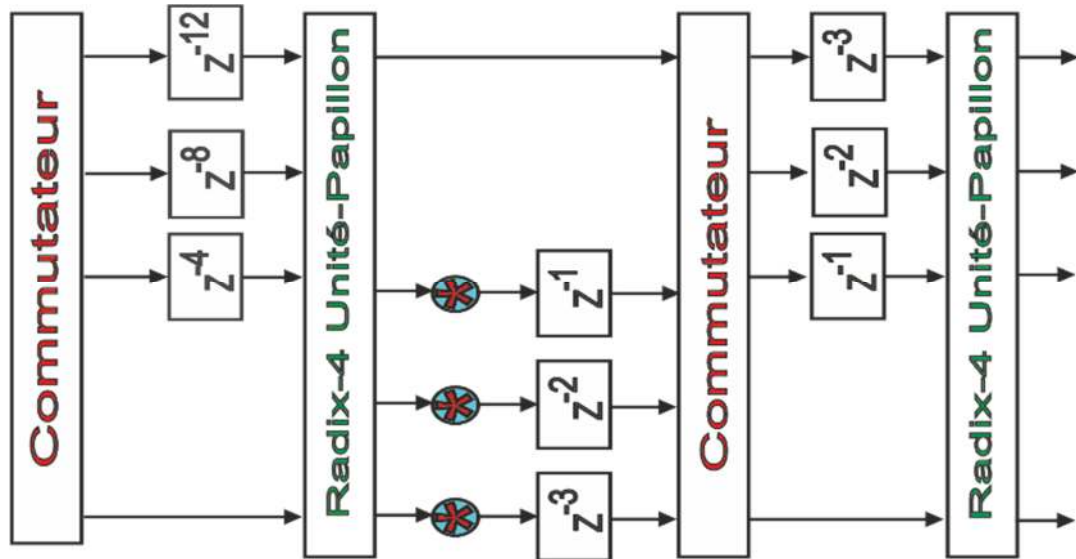


Figure 2-15- Architecture Radix-2 Single Path Delay Feedback FFT pour 16-points

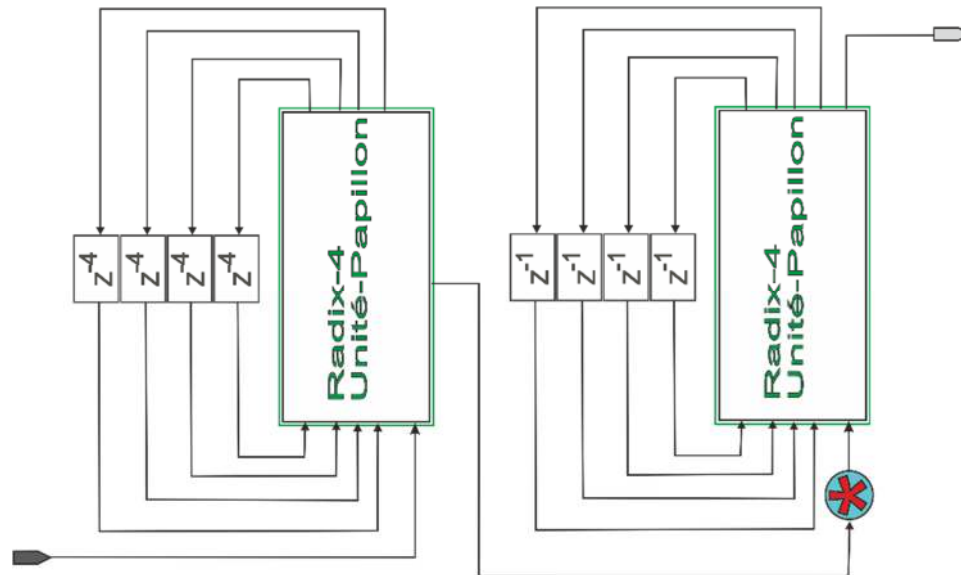


Figure 2-16- Architecture Radix-4 Single Path Delay Feedback FFT pour 16-points

Les figures Fig. 2.18 et Fig. 2.19 montrent la comparaison entre les ressources requises pour les architectures pipelines. On peut remarquer que le régime SDF a la configuration matérielle minimale requise en raison de l'utilisation efficace des multiplieurs complexes et des mémoires tampons de retard [46].

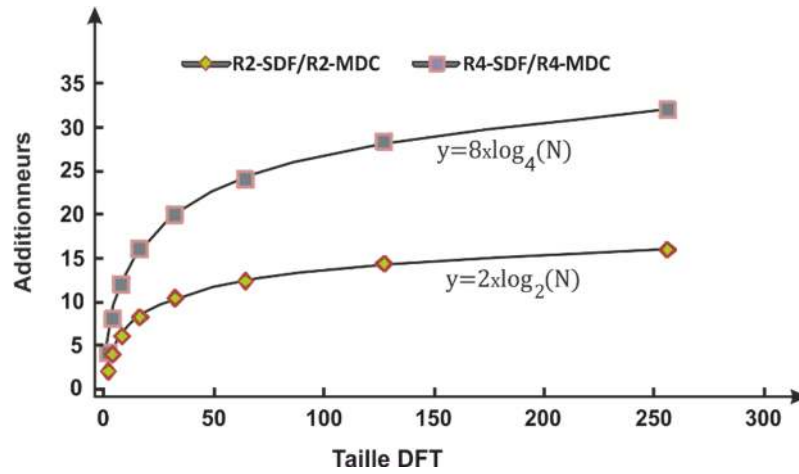


Figure 2-17- Les additionneurs utilisés dans les architectures en pipeline

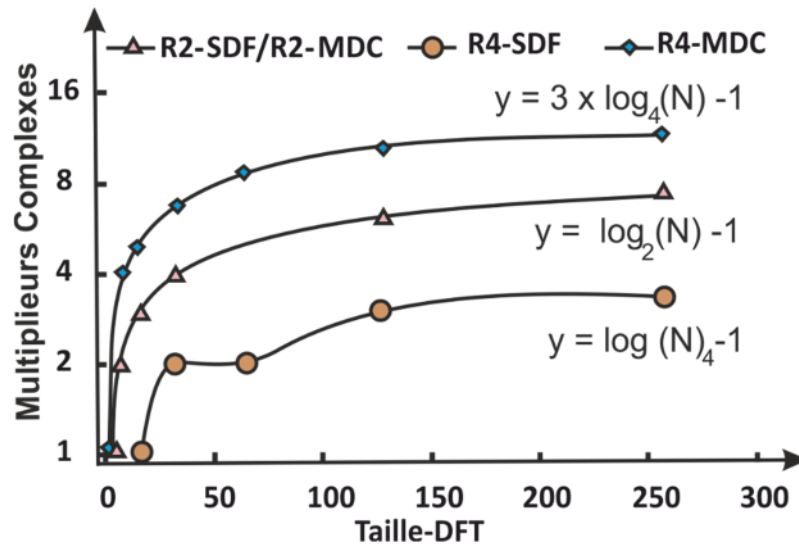


Figure 2-18- Les multiplieurs complexes utilisés dans les architectures en pipeline

3 ALGORITHME PROPOSE

3.1 INTRODUCTION

Pour la sélection d'une approche optimale de la tâche d'implémentation matérielle d'un algorithme, il est important que l'on comprenne exactement la structure de cet algorithme. Ce genre d'entité permettra au concepteur du circuit de tirer avantage de tout aspect de parallélisme ou de compilation particulière de l'algorithme et de les faire correspondre aux capacités des architectures les plus récentes

Dans la plupart des littératures de traitement numérique du signal, l'algorithme FFT peut être représenté en termes aussi bien de concept de décimation en temps (DIT) qu'en termes de concept de décimation en fréquence (DIF). Bien sûr, lorsque ces connotations sont appliquées à l'algorithme de FFT d'une séquence de données, ils sont destinés à s'appliquer à toute opération dans laquelle la TFD d'une séquence de données N points est effectuée en calculant d'abord la DFT de deux ou plusieurs sous-groupes, c.-à-d. en décimant les flux de données en entrée et puis en les combinant d'une certaine façon, avec le but que l'effort total requis pour le calcul des DFT à court terme et en combinant les résultats obtenus est toujours inférieur à celui requis pour le calcul direct de la DFT de la première séquence à N points. Nous n'allons pas commencer immédiatement avec le test de cette hypothèse, mais plutôt, nous allons d'abord étudier le paradigme de base sous-jacent de la FFT. Il sera alors indiqué que cette décomposition réduit considérablement le temps de calcul.

3.2 REPRESENTATION BIDIMENSIONNELLE DE L'ALGORITHME FFT

Considérons une séquence entrante de N points :

$$S: \{x_0, x_1, x_2, \dots, x_{N-1}\}.$$

Si N est positif et non premier, alors il peut être exprimé comme un produit d'au moins deux nombres L et M , c.-à-d., $N = L \times M$

Une séquence N -points de données peut être construite en utilisant la somme d'un certain nombre L de séquences contiguës. Ces séquences auxiliaires sont successivement prises à partir de la séquence d'origine en gardant un sous-ensemble d'échantillons et de remplir le reste avec des zéros.

Une construction spécifique peut être définie comme suit :

La séquence initiale à N points est divisée en L séquences auxiliaires somme de $L \{S^0, S^1, S^2, \dots, S^{L-1}\}$, qui sont définies dans l'équation ci-dessous :

$$\begin{bmatrix} S^0 \\ S^1 \\ \vdots \\ S^{L-1} \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & x_2 & \cdots & x_{M-1} & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & x_M & x_{M+1} & \cdots & x_{2M-1} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \cdots & 0 & 0 & \cdots & \cdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots & \cdots & 0 & x_{(L-1)M} & x_{(L-1)M+1} & \cdots & x_{N-1} \end{bmatrix} \quad (3.1)$$

Il n'y a pas de chevauchement des segments d'entrée non nuls parmi les sous-séquences auxiliaires. Partant du fait que la DFT est une combinaison linéaire, la DFT de la séquence originale d'échantillons de N points est égal à la somme des DFT des L DFT des segments auxiliaires.

$$\begin{bmatrix} X_k^0 \\ X_k^1 \\ \vdots \\ X_k^{L-1} \end{bmatrix} = \begin{cases} \sum_{m=0}^{M-1} x_m W_{LM}^{mk} \\ \sum_{m=0}^{M-1} x_{m+M} W_{LM}^{mk} (W_L^k) \\ \vdots \\ \sum_{m=0}^{M-1} x_{m+(L-1)M} W_{LM}^{mk} (W_L^{k(L-1)}) \end{cases} \quad \forall k = 0, 1, 2, \dots, N-1 \quad (3.2)$$

En utilisant le principe de superposition, nous pouvons facilement exprimer la DFT de la séquence indiquée S comme :

$$X_k = \sum_{i=0}^{L-1} (X_k^i) \quad (3.3)$$

L'équation précédente peut être réécrite sous la forme d'une double sommation avec l'introduction de l'index.

$$X_k = \sum_{l=0}^{L-1} \left(\sum_{m=0}^{M-1} x_{(m+lM)} W_{LM}^{mk} \right) W_L^{lk} \quad \text{avec } k = 0, 1, 2, \dots, N-1 \quad (3.4)$$

À partir de cette nouvelle représentation, il apparait que les propriétés de symétrie de la TFD jouent un rôle clé dans le développement de l'algorithme de FFT. Il convient de noter que le terme exponentiel de la dernière double sommation dans l'équation (3.4) se répète de manière cyclique comme $(lk \bmod L)$.

Considérons les sous-groupes de comportement en fréquence :

ALGORITHME PROPOSE

| $l \backslash m$ | 0 | 1 | 2 | ... | M-1 |
|------------------|--------------|----------------|----------------|-----|--------------|
| 0 | x_0 | x_1 | x_2 | ... | x_{M-1} |
| 1 | x_M | x_{M+1} | x_{M+2} | ... | x_{2M-1} |
| 2 | x_{2M} | x_{2M+1} | x_{2M+2} | ... | x_{3M-1} |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | |
| L-1 | $x_{(L-1)M}$ | $x_{(L-1)M+1}$ | $x_{(L-1)M+2}$ | ... | $x_{(LM)-1}$ |

Tableau 4- Représentation bidimensionnelle d'une séquence initiale décimée en temps

| $k_0 \backslash k_1$ | 0 | 1 | 2 | ... | M-1 |
|----------------------|-------------|------------|------------|-----|----------------|
| 0 | X_0 | X_L | X_{2L} | ... | $X_{(M-1)L}$ |
| 1 | X_1 | X_{L+1} | x_{2L+1} | ... | $X_{(M-1)L+1}$ |
| 2 | X_2 | X_{L+2} | X_{2L+2} | ... | $X_{(M-1)L+2}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | |
| L-1 | $X_{(L-1)}$ | X_{2L-1} | X_{3L-1} | ... | $X_{(LM)-1}$ |

Tableau 5- Représentation bidimensionnelle de la DFT d'une séquence réarrangée

3.4 FACTEURS TWIDDLES

Conformément à l'équation (3.8), les TFD de toutes les colonnes énumérées dans le Tableau 1 doivent être multipliées par un facteur de pondération : $W_N^{mk_0} = e^{\left(\frac{-j2\pi}{N}\right)mk_0}$. Ce coefficient est communément appelé facteur twiddle. En fait, ce facteur twiddle est une fonction de deux variables m et k_0 , qui peuvent également être représentées sous forme de matrice à deux dimensions comme indiqué dans le Tableau 3.

ALGORITHME PROPOSE

| $k_0 \backslash m$ | 0 | 1 | 2 | ... | M-1 |
|--------------------|----------|-------------|----------------|----------|--------------------|
| 0 | 1 | 1 | 1 | ... | 1 |
| 1 | 1 | W_N^1 | W_N^2 | ... | W_N^{M-1} |
| 2 | 1 | W_N^2 | W_N^4 | ... | $W_N^{2(M-1)}$ |
| \vdots | \vdots | \vdots | \vdots | \vdots | \vdots |
| L-1 | 1 | W_N^{L-1} | $W_N^{2(L-1)}$ | ... | $W_N^{(L-1)(M-1)}$ |

Tableau 6- Représentation bidimensionnelle des facteurs twiddles

Lorsque $W_N^{mk_0}$ est égal à l'unité ou $\pm j$, aucune multiplication réelle n'est requise.

En règle générale, nous pouvons appeler ces trois valeurs singulières du facteur twiddle twiddles triviaux [47]. Il convient de souligner que la multiplication twiddle ne doit pas être confondue avec la multiplication matricielle. Il s'agit d'une opération consistant à multiplier les entrées des positions mémoire correspondantes de la matrice DFT et de la matrice twiddle et à former une nouvelle matrice à partir du produit.

Représentons le résultat de la multiplication de twiddle sous la forme d'une matrice à deux dimensions comme indiqué à la Fig 3.1.

| $k_0 \backslash m$ | 0 | 1 | 2 | ... | M-1 |
|--------------------|-------------|-------------|-------------|----------|---------------|
| 0 | $Y_{0,0}$ | $Y_{1,0}$ | $Y_{2,0}$ | ... | $Y_{M-1,0}$ |
| 1 | $Y_{0,1}$ | $Y_{1,1}$ | $Y_{2,1}$ | ... | $Y_{M-1,1}$ |
| 2 | $Y_{0,2}$ | $Y_{1,2}$ | $Y_{2,2}$ | ... | $Y_{M-1,2}$ |
| \vdots | \vdots | \vdots | \vdots | \vdots | \vdots |
| L-1 | $Y_{0,L-1}$ | $Y_{1,L-1}$ | $Y_{2,L-1}$ | ... | $Y_{M-1,L-1}$ |

Tableau 7- Résultats de la multiplication des facteurs Twiddles par la DFT des colonnes séparées

Une grande variété d'architectures FFT peut être dérivée en introduisant ces aspects de décimation de différentes manières par rapport à la séquence d'origine. Les étapes adoptées pour les approches FFT généralisées sont résumées dans le diagramme 3.1 ci-dessous.

ALGORITHME PROPOSE

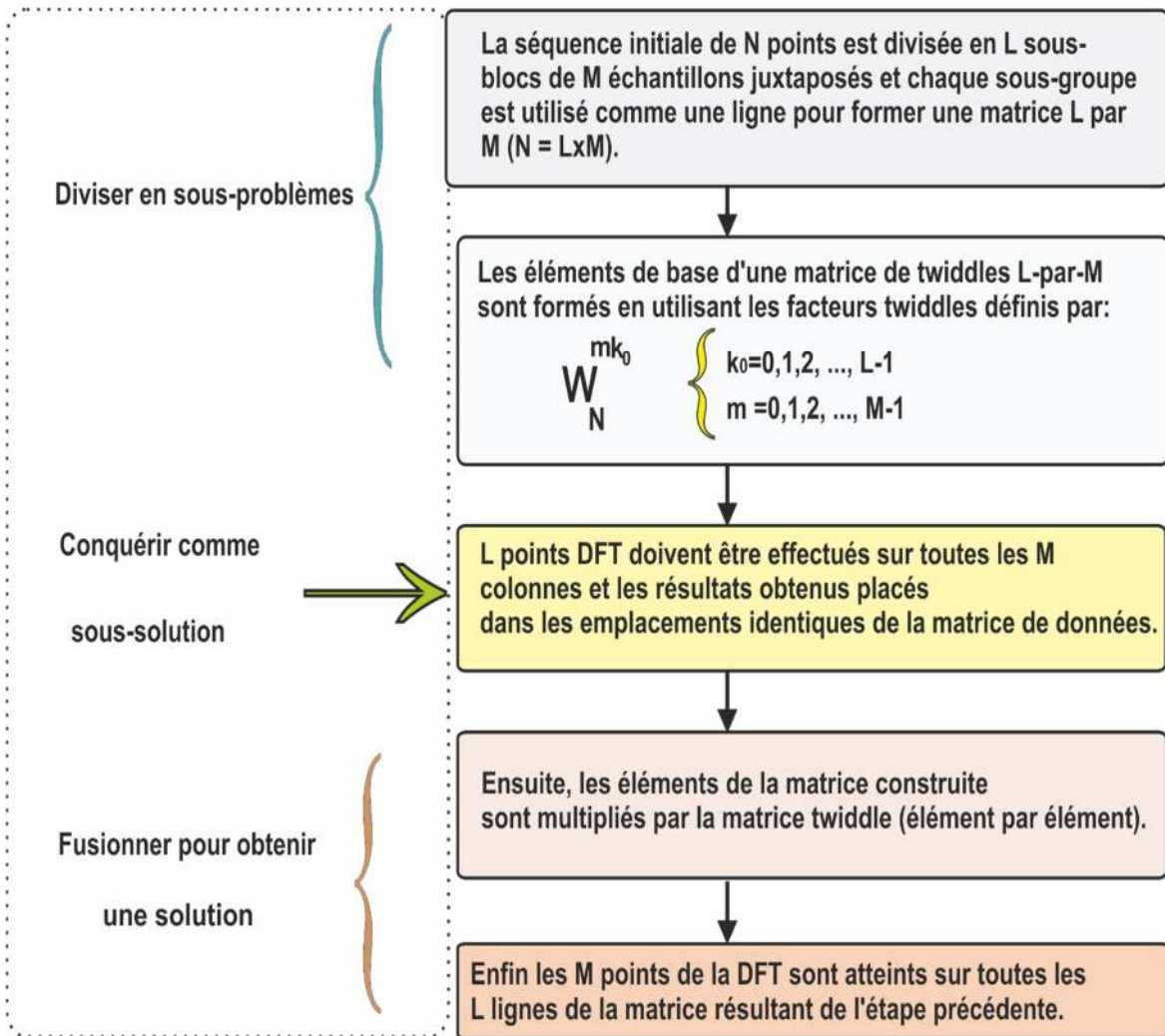


Figure 3-1- Diagramme fonctionnel de l'algorithme FFT

Partant de ce diagramme fonctionnel de l'algorithme FFT, nous nous proposons de mettre en évidence une conception graphique d'un processeur FFT Radix-16. L'exemple traite d'une suite constituée de 16 échantillons à laquelle nous appliquons les traitements induits par le diagramme fonctionnel de l'algorithme FFT. Cet algorithme est présenté comme une sorte de Radix-4 avec une architecture Single Path Delay FFT.

Nous exposerons également un exemple Radix-64. Nous considérerons une suite de 64 échantillons. Cet algorithme est présenté comme une sorte de Radix-8 avec une architecture Single Path Delay FFT.

A partir de ces exemples et à titre de généralisation, Nous montrons l'extension de cet algorithme à n'importe quelle suite et n'importe quel Radix.

3.5 EXEMPLE D'UN PROCESSEUR FFT RADIX-16

Cet algorithme est présenté comme une sorte de Radix-4 avec une architecture Single Path Delay FFT. La suite à traiter est une suite de N échantillons ($N = 16$).

N Doit être décomposé en un produit de deux entiers : $N = L * M = 16$. De toute évidence, la séquence de données d'entrée $x(n)$, $0 \leq n \leq 15$ est arrangée en une matrice 2-D en prenant les index n comme une paire d'index (l, m) $0 \leq l \leq 3$ et où $0 \leq m \leq 3$.

Ainsi, le réarrangement de $x(n)$ sous forme matricielle peut être ainsi exprimé :

$$A = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{bmatrix}$$

Tableau 8- réarrangement de $x(n)$ sous forme matricielle

Un réarrangement semblable est fait dans le domaine de fréquence par le remplacement de l'index k par une paire (k_0, k_1) d'index, d'où $0 \leq k_0 \leq 3$ et $0 \leq k_1 \leq 3$.

De ce fait, en sortie, les échantillons X_k sont stockés dans la mémoire intermédiaire (registres) en respectant l'ordre $k = 4k_1 + k_0$.

Dans ce cas la matrice 2D est réarrangée comme suit :

$$X_{\underbrace{k_0}_{+} \underbrace{4k_1}} = \sum_{m=0}^3 Y_{(mk_0)} W_4^{mk_1} \quad (3.9)$$

$$\text{où : } Y_{mk_0} = (W_{16}^{mk_0}) \sum_{l=0}^3 (x_{(m+4l)} W_L^{lk_0}) \quad (3.10)$$

Pour corroborer cette suggestion, l'architecture préliminaire décrivant cet algorithme sera détaillé dans la figure 3.8.

La topologie de l'algorithme proposé peut être décomposée en trois parties essentielles :

- Un commutateur qui nécessite un circuit décimateur avec un circuit splitter pour gérer le flux des données et ainsi former les sous-groupes de données qui peuvent être empilés dans un espace mémoire préliminaire avant d'être acheminés vers le prochain étage. Cette étape sera matérialisée par une machine d'état spécifique pour répondre à notre exigence d'une telle conception. Bien sûr, cette étape nécessite environ 16 cycles en termes de latence et d'un grand espace mémoire comme des registres internes.

ALGORITHME PROPOSE

- Une structure Radix-4 qui représente des facteurs twiddles triviaux ainsi que les facteurs twiddles principaux. Il convient de noter que cette structure est la plus gourmande en ressources internes et pour laquelle l'optimisation sera ostensiblement recommandée pour accélérer la convergence de cet algorithme de manière significative.
- Un bloc de multiplication complexe impliquant deux unités d'éléments processeurs réduisant la zone occupée ainsi que le temps de traitement. De la même manière, un espace mémoire sera prévu afin de stocker les résultats obtenus avant leur exploitation par une autre unité de traitement.
- Il est crucial de prévoir un générateur d'adresses pour sélectionner la zone mémoire correspondante et donc d'établir la structure papillon imposée par le choix de la taille de l'unité de traitement en fonction du type de Radix à adopter pour l'architecture sélectionnée.

Les étapes de la conception graphique de la topologie de l'algorithme proposé, conduisant à la Figure 3.9, sont résumées comme suit :

- 1 Les échantillons de la suite à traiter sont décimés sous forme de matrice 4x4, conformément au Tableau 9.

$$A = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{bmatrix}$$

Tableau 9: Décimation des échantillons sous forme de matrices

Cette étape est matérialisée par ce début de topologie :

- 1- Décimation des échantillons sous forme de matrices



Figure 3-2 Décimation des échantillons sous forme de matrices



Figure 3-3- Le commutateur splitter

- 2 . Le commutateur splitter gère flux de données et forme les sous-groupes de données qui peuvent être empilés dans un espace mémoire préliminaire avant d'être acheminés vers le prochain étage.

Cette étape ajoute un espace mémoire dans lequel les échantillons décimés sont stockés. Cet espace mémoire est référencé comme mémoire d'entrée dans le schéma suivant

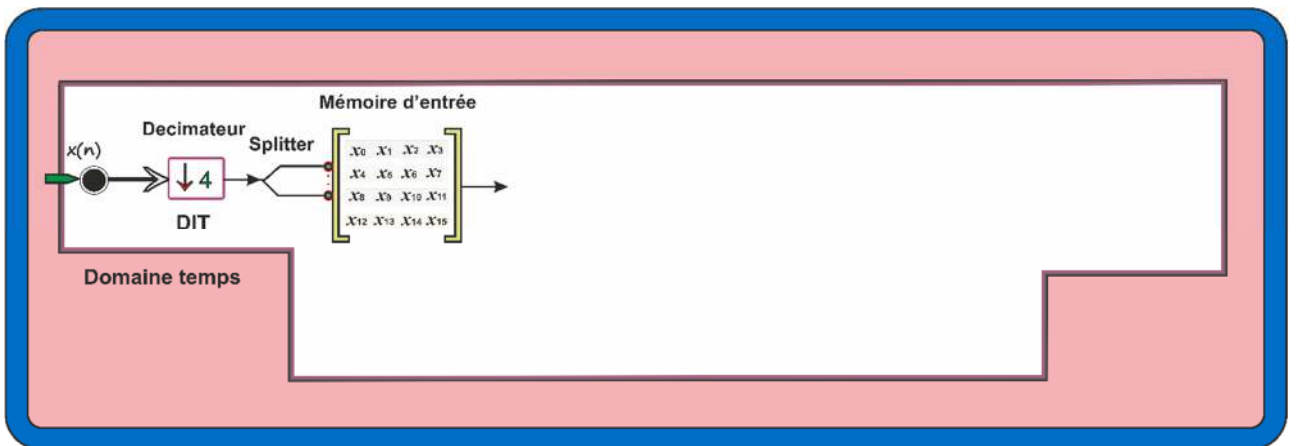


Figure 3-4- Le commutateur splitter et espace mémoire de stockage des données décimées

- 3 Une structure Radix-4 qui représente les facteurs twiddles triviaux ainsi que les facteurs twiddles principaux est ajoutée à la topologie de l'algorithme pour préparer la FFT d'un bloc décimé de L échantillons de la séquence N points d'origine S (3.8)

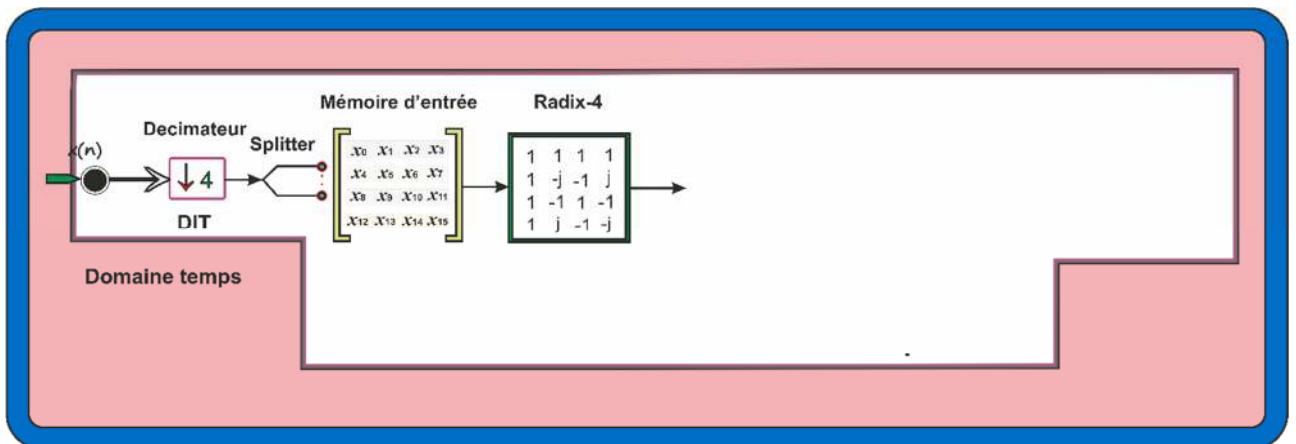


Figure 3-5- Le commutateur splitter et espace mémoire de stockage des données décimées

- 4 Un bloc de facteur de pondération (facteurs twiddles) est ajoutée à la topologie. la FFT d'un bloc décimé de L échantillons de la séquence N points d'origine S est multiplié terme à terme par les twiddles. (3.8) (3.9) (3.10)

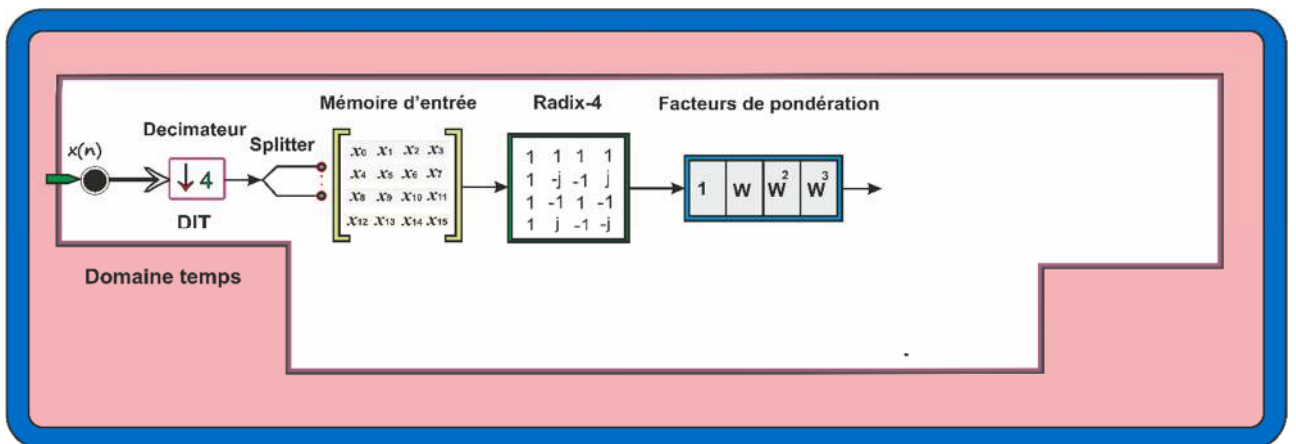


Figure 3-6- Le commutateur splitter, espace mémoire de stockage des données décimées et blocs de facteurs twiddles

- 5 Une structure Radix-4 qui représente les facteurs twiddles triviaux ainsi que les facteurs twiddles principaux est ajoutée à la topologie de l'algorithme après le bloc de facteur twiddles. (3.8)

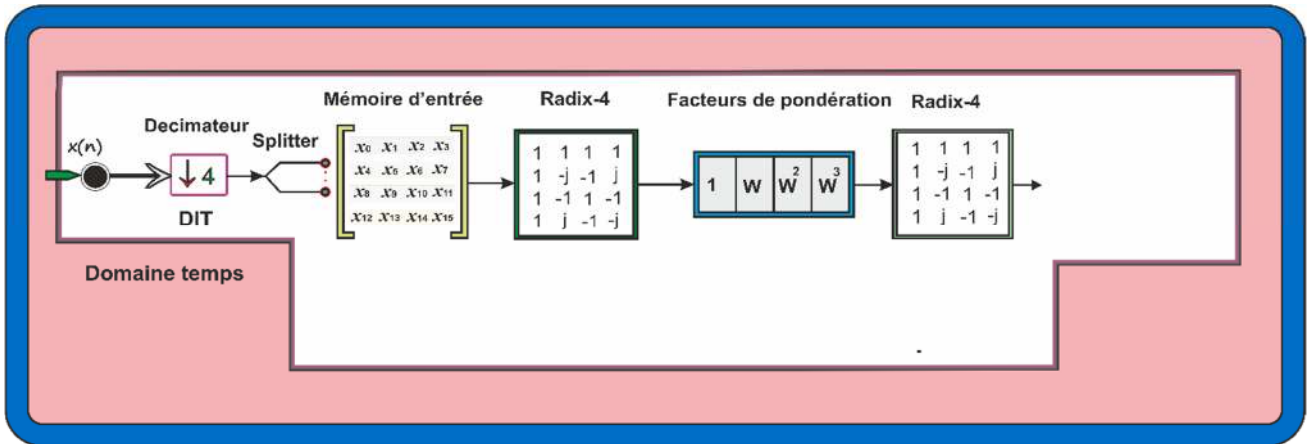


Figure 3-7- Le commutateur splitter, espace mémoire de stockage des données décimées et blocs de facteurs twiddles

6 Le multiplieur complexe avec les facteurs twiddles

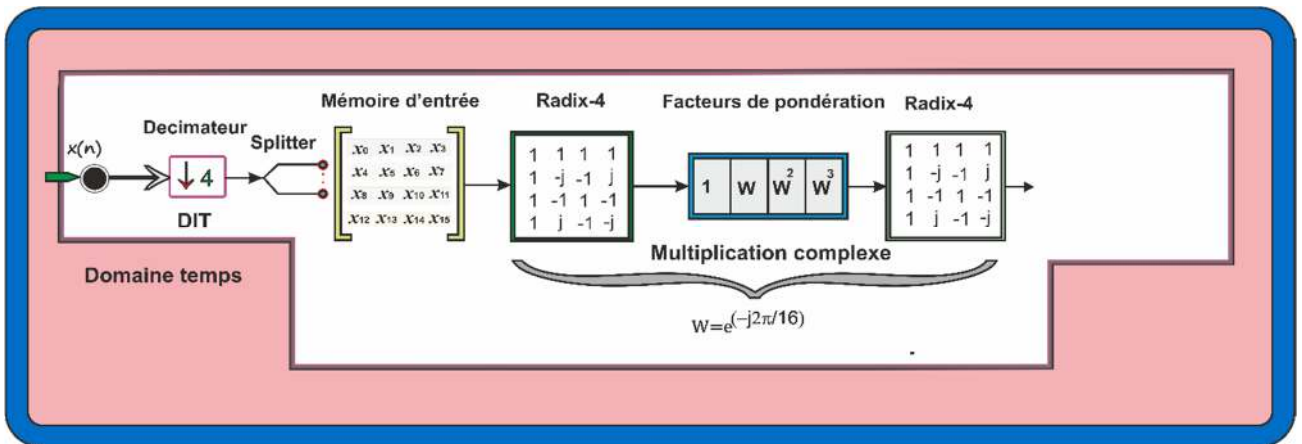


Figure 3-8- Le commutateur splitter et espace mémoire de stockage des données décimées Structures Radix-4 et blocs de facteurs twiddles arrangés en multiplieur complexe

Le facteur twiddle est $w = e^{-j2\pi/16}$ découlant du fait que l'on a pris comme base Radix-4 avec $N = 16$ échantillons.

7 La topographie complète est ainsi présentée.

La figure 3.8 est complétée par un générateur d'adresses (sélecteur) pour sélectionner l'emplacement mémoire correspondant dans la mémoire de sortie et donc d'établir la

ALGORITHME PROPOSE

structure papillon imposée par le choix de la taille de l'unité de traitement en fonction du type de Radix à adopter pour l'architecture sélectionnée.

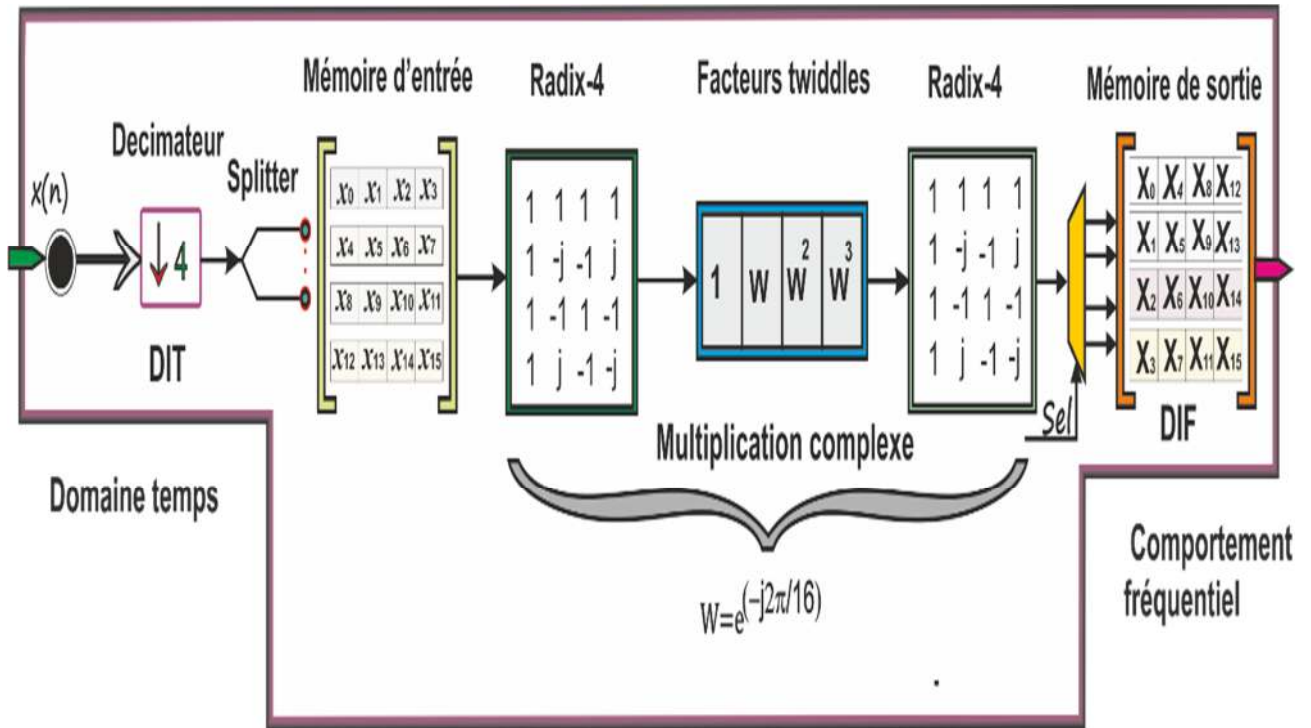


Figure 3-9- Le commutateur splitter, espace mémoire de stockage des données décimées et blocs de facteurs twiddles avec sélecteur et mémoires de sorties

Le comportement du composant est évalué avec un code MATLAB décrivant son fonctionnement. Le code est présenté ci-après. Chaque bloc fonctionnel est décrit par une séquence d'instructions MATLAB.

Une série tout à fait arbitraire est choisie pour mettre en évidence le bon fonctionnement de l'algorithme. Cette série est la suivante :

$$x = [115 \ 97 \ 91 \ 31 \ 63 \ 255 \ 153 \ 41 \ 21 \ 3 \ 135 \ 1 \ 109 \ 145 \ 59 \ 128];$$

Les résultats obtenus par comparaison sont exposés ci-après (Tableau 6) :

La colonne de gauche représente le calcul de la FFT de la série prise en exemple. Elle est le résultat de la fonction implantée sous MATLAB. Les résultats obtenus sont donnés Tableau 6.

La colonne de droite représente le calcul de la FFT effectué par le processeur proposé (Figure 2.8).

ALGORITHME PROPOSE

Code MATLAB décrivant le fonctionnement du composant

```
clc;
clear all;
close all;

% Séquence des données en entrée
x = [115 97 91 31 63 255 153 41 21 3 135 1 109 145 59 128];

% Décimation en temps des données
%et mise en mémoire

x0=x(1:4:end);
x1=x(2:4:end);
x2=x(3:4:end);
x3=x(4:4:end);

% Construction de Radix 4 et calculs
R4=[1 1 1 1;1 -j -1 j;1 -1 1 -1;1 j -1 -j];

format short;
A = x0 * R4;
B = x1 * R4;
C = x2 * R4;
D = x3 * R4;

% Générateur de facteurs Twiddles
W= exp(-2*pi*j *(0:3) / 16);

% Calcul FFT en combinant les facteurs
%twiddles et Radix-4

X0 = [A + (B.* W) + (C.* W.^2) + (D.* W.^ 3)].';
X1 = [A -j * (B.* W) - (C.* W.^2) + j * (D.* W.^ 3)].';
X2 = [A - (B.* W) + (C.* W.^2) - (D.* W.^ 3)].';
X3 = [A + j *(B.* W) - (C.* W.^2) - j *(D.* W.^ 3)].';

%Réarrangement des résultats en utilisant les
%mémoires de sortie
X=[X0;X1;X2;X3]
format short G

% Comparaison des obtenus avec les
%fonctions Matlab implentées
fft(x,16) .' ;
X ;
```

ALGORITHME PROPOSE

| le calcul de la FFT effectué par la fonction fft implantée sous MATLAB | le calcul de la FFT effectué par la processeur proposé (Figure 2.8). |
|--|--|
| [fft(x,16) .'] | x |
| 1447.00 + 0i | 1447.00 + 0i |
| 133.03 - 121.38i | 133.03 - 121.38i |
| -151.26 + 295.01i | -151.26 + 295.01i |
| 334.76 - 34.247i | 334.76 - 34.247i |
| -130 - 299i | -130 - 299i |
| 48.405 + 128.46i | 48.405 + 128.46i |
| 79.258 + 323.01i | 79.258 + 323.01i |
| -140.19 - 142.67i | -140.19 - 142.67i |
| 45 + 0i | 45 + 0i |
| 140.19 + 142.67i | 140.19 + 142.67i |
| 79.258 - 323.01i | 79.258 - 323.01i |
| 48.405 - 128.46i | 48.405 - 128.46i |
| -130 + 299i | -130 + 299i |
| 334.76 - 34.247i | 334.76 - 34.247i |
| -151.26 - 295.01i | -151.26 - 295.01i |
| 133.03 + 121.38i | 133.03 + 121.38i |

Tableau 6 : Comparaison des résultats MTLAB-Algorithmme Proposé

Les résultats sont strictement identiques. Cela valide complètement l’algorithme proposé.

3.6 EXEMPLE D’UN PROCESSEUR FFT RADIX-64

Partant de l’exemple précédent, on se propose de présenter un schéma d’implémentation pour une séquence de 64 échantillons basée sur un radix-8. Cette étape permettra de généraliser le concept à n’importe quel Radix-r et à n’importe quelle taille N de la suite à traiter.

Il suffit simplement de changer les blocs décimateurs (ils prennent une dimension 8x8 au lieu de 4x4 dans le cas de Radix-16) puis par conséquent on modifie le générateur de

facteurs twiddles qui peut prendre la forme $W = e^{\left(\frac{-j2\pi}{64}\right)}$

Si nous prenons $N = L * M$ ($64 = 8 * 8$), Le schéma d’implémentation est alors le suivant :

ALGORITHME PROPOSE

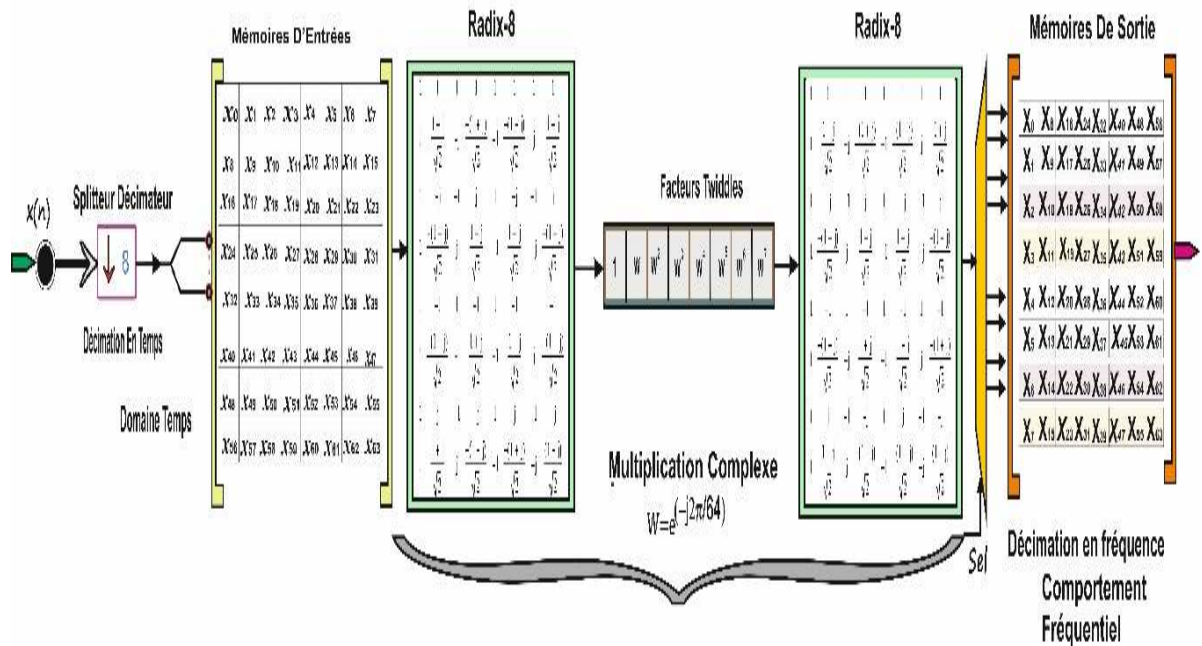


Figure 3-10- Conception graphique d'un processeur élément pour FFT Radix 64

- Les blocs décimateurs sont de dimensions 8 x 8
- L'élément processeur est basé sur Radix-8
- Le générateur de twiddles est $W = e^{\left(\frac{-j2\pi}{64}\right)}$

Le code MATLAB décrivant ce comportement est exposé ci-après :

```

clc;
close all;
clear all;
% % Séquence des données en entrée (64)
x=[ 115 97 91 31 63 255 153 41 ...
    21 3 135 1 109 145 59 128 ...
    213 258 789 852 458 1 223 669 ...
    456 521 412 458 789 85 52 48 ...
    21 28 79 82 48 10 23 669 ...
    45 51 42 48 79 82 28 41 ...
    2 29 71 820 481 100 203 66 ...
    452 58 412 486 791 824 281 410 ];

% Décimation en temps et mémorisation des données
x0=x(1:8:end);
x1=x(2:8:end);
x2=x(3:8:end);
x3=x(4:8:end);
x4=x(5:8:end);
x5=x(6:8:end);
x6=x(7:8:end);
x7=x(8:8:end);
% Construction Radix 8

```

Diviser en sous-problèmes

Radix-8=

$$\begin{bmatrix}
 W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 \\
 W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \\
 W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14} \\
 W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21} \\
 W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28} \\
 W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35} \\
 W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42} \\
 W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49}
 \end{bmatrix}$$

Radix-8=

$$\begin{bmatrix}
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 0.7 - 0.7i & -i & -0.7 - 0.7i & -1 & -0.7 + 0.7i & i & 0.7 + 0.7i \\
 1 & -i & -1 & i & 1 & -i & -1 & i \\
 1 & -0.7 - 0.7i & i & 0.7 - 0.7i & -1 & 0.7 + 0.7i & -i & -0.7 + 0.7i \\
 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
 1 & -0.7 + 0.7i & -i & 0.7 + 0.7i & -1 & 0.7 - 0.7i & i & -0.7 - 0.7i \\
 1 & i & -1 & -i & 1 & i & -1 & -i \\
 1 & 0.7 + 0.7i & i & -0.7 + 0.7i & -1 & -0.7 - 0.7i & -i & 0.7 - 0.7i
 \end{bmatrix}$$

**% Décimation en temps des données
%et mise en mémoire**

A = x0*R8;
B = x1*R8
C = x2*R8;
D = x3*R8;
E = x4*R8;
F = x5*R8;
G = x6*R8;
H = x7*R8

Conquérir comme sous-solutions

% Générateur des facteurs Twiddles

W = xp(-2 * pi * j * (0:7) / 64);

% Posons R2 = 2^{1/2}

% Calculs FFT en combinant les facteurstwiddles et Radix-8

X0 = [A + (B.* W) + (C.* W.^2) + (D.* W.^3) + (E.* W.^4) + (F.* W.^5) + (G.* W.^6) + (H.* W.^7)].';

X1 = [A + ((1 - j)/R2) * (B.* W) - j * (C.* W.^2) - ((1+j)/R2) * (D.* W.^3) - 1 * (E.* W.^4) - ((1 - j)/R2) * (F.* W.^5) + j * (G.* W.^6) + ((1+j)/R2) * (H.* W.^7)].';

X2 = [A - j * (B.* W) - 1 * (C.* W.^2) + j * (D.* W.^3) + 1 * (E.* W.^4) - j * (F.* W.^5) - 1 * (G.* W.^6) + j * (H.* W.^7)].';

X3 = [A - ((1+j)/R2) * (B.* W) + j * (C.* W.^2) + ((1 - j)/R2) * (D.* W.^3) - 1 * (E.* W.^4) + ((1+j)/R2) * (F.* W.^5) - j * (G.* W.^6) - ((1-j)/R2) * (H.* W.^7)].';

X4 = [A - ((1+j)/R2) * (B.* W) + j * (C.* W.^2) + ((1 - j)/R2) * (D.* W.^3) - 1 * (E.* W.^4) + ((1+j)/R2) * (F.* W.^5) - j * (G.* W.^6) - ((1-j)/R2) * (H.* W.^7)].';

X5 = [A - ((1-j)/R2) * (B.* W) - j * (C.* W.^2) + ((1+j)/R2) * (D.* W.^3) - 1 * (E.* W.^4) + ((1 - j)/R2) * (F.* W.^5) + j * (G.* W.^6) - ((1+j)/R2) * (H.* W.^7)].';

X6 = [A + j * (B.* W) - 1 * (C.* W.^2) - j * (D.* W.^3) + 1 * (E.* W.^4) + j * (F.* W.^5) - 1 * (G.* W.^6) - j * (H.* W.^7)].';

X7 = [A + ((1+j)/R2) * (B.* W) + j * (C.* W.^2) - ((1-j)/R2) * (D.* W.^3) - 1 * (E.* W.^4) - ((1+j)/R2) * (F.* W.^5) - j * (G.* W.^6) + ((1 - j)/R2) * (H.* W.^7)].';

**%Réarrangement des résultats en utilisant les mémoires de
%sorties**

X = [X0;X1;X2;X3;X4;X5;X6;X7];

format short G;

Y = fft(x,64);

Y

X

**% Les résultats des fonctions MATLAB et ceux de notre
%algorithme sont identiques**

% Y = 452.49 + 1512i FFT Matlab

% X = 452.49 + 1512i Calcul avec notre algorithme

Les résultats X et Y sont identiques comme indiqué à la fin du code précédent et valident le schéma.

3.7 GENERALISATION A UN PROCESSEUR FFT DE N POINTS RADIX- M

Comme mentionné dans les deux cas précédents, Il résulte de cet aspect que lorsque l'on souhaite traiter une FFT de N points en utilisant la représentation Radix- M il suffit simplement de :

- Changer les blocs décimateurs (ils deviennent $\downarrow M$ au lieu de $\downarrow 4$ ou $\downarrow M$ au lieu de $\downarrow 8$)

- Modifier le générateur de facteurs twiddles qui peut prendre la forme générale :

$$W = e^{\left(\frac{-j2\pi(0\dots M-1)}{N}\right)} \text{ av } \mathbb{C} \quad N \geq 2M.$$

Ainsi l'algorithme peut calculer une FFT de n'importe quelle taille sur la base de n'importe quel Radix.

4 MODIFICATION ET AMÉLIORATION

Selon cette description, nous pouvons facilement constater que tout type de modification à appliquer à cette approche est étroitement liée à la méthodologie adoptée lors de la phase de mise en œuvre matérielle pour satisfaire nos besoins à la fois en termes de complexité algorithmique et de gain attendu dans les performances métriques.

Ainsi cet aspect de la présentation nous incite à faire une projection à partir de l'espace 2D de la séquence d'entrée présélectionnée sur le plan 2D formé par la matrice des facteurs twiddles pour obtenir un vecteur de la composante principale qui reflète le comportement fréquentiel de la séquence d'origine à traiter.

Dans ce type de réalisation, notre souci majeur est d'éviter la structure matricielle pour les raisons suivantes :

- La latence, facteur déterminant pouvant influencer négativement sur la qualité de transmission en termes de débits et donc en volume des données à transmettre
- Gourmandise en termes de ressources internes (primitives ; registres internes, bascules, mémoires, additionneurs, multiplieurs...). Ceci augmente considérablement la surface occupée.
- Complexité apparente en termes de méthodologie de programmation des différents blocs nécessaires pour l'implémentation finale du processeur

Donc, notre contribution suivante consiste à créer une structure récursive de filtres numériques afin de faciliter le processus de calcul de la TFD qui devrait répondre aux exigences de transmission OFDM à grande vitesse à moindre coût et à un compromis intelligent pour éviter les contraintes susmentionnées.

Dans le but d'expliquer notre approche de cette structure récursive de filtres numériques envisagée, nous nous concentrons sur le cas $K = 1$ et $N = 4$ c'est-à-dire qu'on effectue le calcul de la première raie spectrale $X(1)$ d'une suite de $N = 4$ échantillons.

4.1 MODIFICATION DE L'ALGORITHME PROPOSÉ

4.1.1 INTRODUCTION

L'algorithme proposé est de nature parallèle. Il a une complexité algorithmique performante que nous quantifierons par la suite. Ses performances sont indiquées aussi au regard des critères pertinents et nous avons comparé avec des structures équivalentes.

De tous points de vue, il est clairement indiqué que les performances (après simulation et implémentation sur une carte FPGA cible Xilinx Virtex-6 xc6lx75t-2ff484) sont très satisfaisantes, comparées aux résultats obtenus par des architectures équivalentes.

La TFD d'une suite temporelle d'échantillons $x(n)$ est donnée par la formule :

$$X(K) = \sum_{n=0}^{n=N-1} x(n) W_N^{Kn} \quad \text{avec} \quad 0 \leq k \leq N-1 \quad \text{et} \quad W_N^{kn} = e^{-2\pi j \frac{kn}{N}}$$

La forme matricielle de la TFD s'écrit ainsi :

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N^1 & W_N^2 & \dots & W_N^{(N-1)} \\ 1 & W_N^2 & W_N^4 & \dots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \dots & W_N^{(N-1)(N-1)} \end{bmatrix} \times \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{bmatrix} \quad (4.1)$$

Comme dit précédemment, nous nous proposons de d'effectuer le calcul pour le cas $N = 4$ et $K = 1$ et de généraliser le cas à un N quelconque. L'équation 1 (pour $N = 4$) se réduit alors à Equation 2

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^4 & W_4^6 \\ 1 & W_4^3 & W_4^4 & W_4^9 \end{bmatrix} \times \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \quad (4.2)$$

Partant des propriétés de symétrie et de périodicité, la matrice de transformation s'écrit en remplaçant les twiddles par leurs valeurs ;

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^4 & W_4^2 \\ 1 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix} \times \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \quad (4.3)$$

Partant des résultats :

$$W_4^0 = 1, \quad W_4^1 = -j, \quad W_4^2 = -1, \quad W_4^3 = j \quad (4.4)$$

On peut écrire :

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \times \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \quad (4.5)$$

En effectuant les calculs, on obtient :

$$X(0) = x(0) + x(1) + x(2) + x(3) \quad (4.6)$$

$$X(1) = x(0) - jx(1) - x(2) + jx(3) \quad (4.7)$$

où les éléments de base du vecteur sont donnés par la relation suivante:

$$P_k = W_N^{(N-k)} = C_k + jS_k \quad (4.18)$$

La transformée en Z de cette dernière expression donne le résultat suivant :

$X_k(z) = E_k(z)H_k(z)$, où $E_k(z)$ désigne la transformée en Z de la séquence initiale d'entrée et où la fonction de transfert du filtre récursif est définie comme suit :

$$H_k(z) = \frac{1}{1 - P_k z^{-1}} \quad (4.19).$$

Pour clarifier l'évolution de cette étape, prenons un exemple introductif en utilisant la structure Radix-4 dont la matrice de transformation est exprimée par :

$$R_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \text{ Dont le vecteur composant principal est : } \begin{bmatrix} 1 \\ j \\ -1 \\ -j \end{bmatrix} = \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (4.20)$$

Grâce à cette représentation on peut constater que la complexité algorithmique en termes de nombre d'opérations requises pour le calcul de la TFD est réduite à trois opérations simples, à savoir l'addition, la soustraction et l'opération conjuguée du deuxième échantillon. En outre, Ce processus peut être modélisé par un ensemble de tampons d'entrée comme un accumulateur dont la taille est estimée à N fois la taille de la longueur des entrées et enfin un autre groupe de registres pour stocker les résultats obtenus. Ainsi, pour mieux comprendre cette approche, nous nous limitons au calcul de la première valeur spectrale qui sera évaluée par le produit scalaire direct comme suit :

$$X_1 = [1 \quad -j \quad -1 \quad j] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4.21)$$

Dans ce cas, si nous appliquons notre méthode l'expression TFD sera réduite à la notation suivante :

$$X_1 = P_1 [x_3 + P_1 [x_2 + P_1 [x_1 + P_1 x_0]]], \quad \text{où } P_1 = W_4^3 = j. \quad (4.22)$$

Sur le plan pratique, cette récursivité implique des intégrateurs et des opérateurs de dérivation en tant que circuit accumulateur pour fournir le résultat attendu.

On se propose maintenant de construire un filtre dont les pôles sont les composantes du Vecteur Composant Principal.

La figure 25 montre la position des pôles P_0, P_1, P_2, P_3 sur le cercle unitaire. Le sens de placement des pôles est indiqué figure 25. C'est le sens contraire du sens des aiguilles d'une montre (Counter ClockWise).

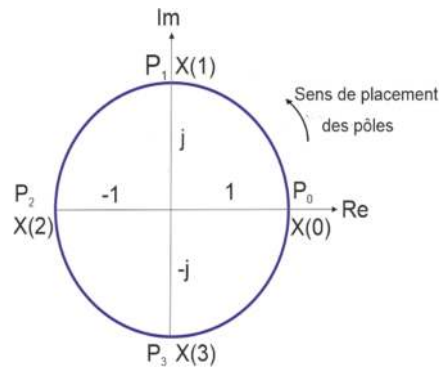


Figure 4-1- Position des pôles sur le cercle unitaire

L'expression de la fonction de transfert d'un tel filtre est :

$$H_i(z) = \frac{1}{1 - P_i Z^{-1}} \quad (4.22)$$

Cette fonction de transfert peut être écrite sous la forme :

$$H_{(i)}(z) = \frac{X(k)}{x(n)} = \frac{1}{1 - P_i Z^{-1}} \quad (4.23)$$

Il convient de noter ici que les expressions $X(k)$ et $x(n)$ sont prises au sens suite des termes, c'est-à-dire suite temporelle $x(n)$ des échantillons d'entrée et $X(k)$ suite fréquentielle des sorties.

En développant cette fonction, on obtient :

$$X(k) = \frac{x(n)}{1 - P_k Z^{-1}} \quad (4.24)$$

$$X(k)(1 - P_k Z^{-1}) = x(n)$$

$$X(k) - X(k)P_k Z^{-1} = x(n) \quad (4.25)$$

$$X(k) = x(n) + X(k)(1 - P_k Z^{-1})$$

On constate que la k ième composante spectrale $X(k)$ est calculée en sommant l'échantillon x_n et la composante spectrale $X(k - 1)$ pondéré par un coefficient P_i dépendant de k .

Le schéma 4.2 qui modélise ce calcul se réduit donc à un accumulateur à deux entrées qui somme chaque échantillon entré x_n avec le résultat $X(k - 1)$ à la sortie de cet accumulateur au moment du traitement de x_{n-1} pondéré par un coefficient P_i dépendant de k .

Le schéma suivant traduit le comportement décrit précédemment :

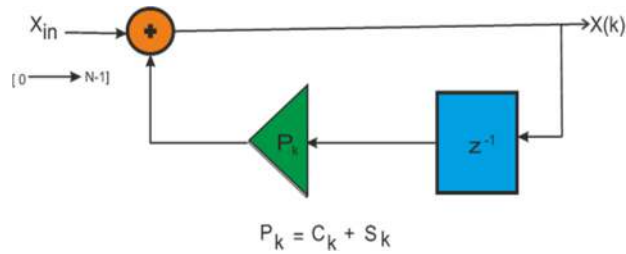


Figure 4-2- Modélisation des calculs

Où $P_k = W_N^{N-k} = C_k + jS_k$ (4.26)

Calculons alors les valeurs de $X(1), X(2), X(3)$ dans le cas d'une suite de $N = 4$ échantillons en utilisant le modèle de la figure 10 .

Avec la notation (12) $P_1 = W_N^{N-1} = W_4^{4-1} = W_4^3 = j$
 $P_2 = W_N^{N-2} = W_4^{4-2} = W_4^2 = -1$
 $P_3 = W_N^{N-3} = W_4^{4-3} = W_4^1 = 1$

Les tableaux suivants tracent les sorties des accumulateurs (et donc X_{out}) dans le cas $X(1), X(2), X(3)$.

Considérons le schéma suivant, dérivé de (10) pour $K = 1$ et $N = 4$ et calculons la valeur de $X(1)$:

Prenons $P_1 = W_4^3$ et suivons le calcul de $X(1)$

1-Calcul de $X(1)$

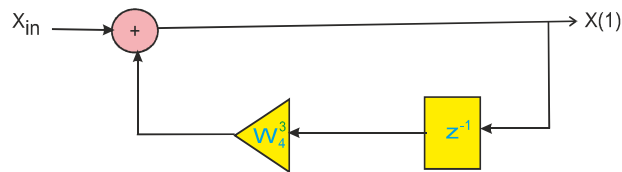


Figure 4-3- Modélisation des calculs pour X1

Le tableau suivant (Tableau 6) trace l'évolution des calculs du schéma ci-dessus

| Echantillons | Accumulateur | X_{out} |
|--------------|---|---|
| $x(0)$ | 0 | $x(0)$ |
| $x(1)$ | $W_4^3 x(0)$ | $x(1) + W_4^3 x(0)$ |
| $x(2)$ | $W_4^3 [x(1) + W_4^3 x(0)]$ | $x(2)$ + $W_4^3 [x(1)$ + $W_4^3 x(0)]$ |
| $x(3)$ | $W_4^3 [x(2) + 3[x(1)$ + $W_4^3 x(0)]]$ | $x(3) + W_4^3 [x(2)$ + $W_4^3 [x(1)$ + $W_4^3 x(0)]]$ |
| 0 | $W_4^3 [x(3) + W_4^3 [x(2)$ + $W_4^3 [x(1)$ + $W_4^3 x(0)]]]$ | |

Tableau 10 - Evolution des calculs du schéma pour X_1

Le zéro en fin de séquence sert de séparateur entre les différentes séquences à traiter. On constate que le temps machine imparti à cette opération est de $N + 1$ cycles.

En écrivant et en simplifiant avec ces notation, l'expression de $X(1)$ devient :

$$X(1) = W_4^3 [x(3) + W_4^3 [x(2) + W_4^3 [x(1) + W_4^3 x(0)]]]$$

Partant de $W_4^3 = j$

$$X(1) = j [x(3) - j[x(2) + j[x(1) + jx(0)]]]$$

$$X(1) = -jx(3) - x(2) + jx(1) + x(0)$$

Ce résultat pour $X(1)$ est identique au résultat obtenu par le calcul de la forme directe (4.7).

2-Calcul de $X(2)$

Le calcul est similaire à celui de $X(1)$, à ceci près que le gain s'écrit :

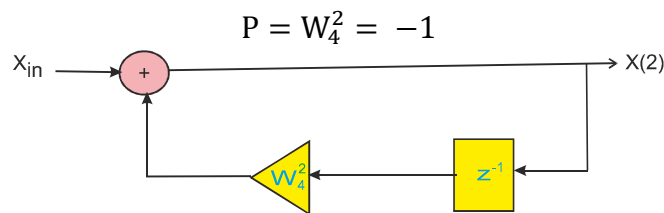


Figure 4-4- Modélisation des calculs pour X_2

La séquence de calcul est la suivante :

| Echantillons | Accumulateur | X_{out} |
|--------------|--|--|
| $x(0)$ | 0 | $x(0)$ |
| $x(1)$ | $W_4^2 x(0)$ | $x(1)$ + $W_4^2 x(0)$ |
| $x(2)$ | $W_4^2 [x(1) + W_4^2 x(0)]$ | $x(2)$ + $W_4^3 [x(1)$ + $W_4^3 x(0)]$ |
| $x(3)$ | $W_4^2 [x(2) + 3[x(1)$ + $W_4^2 x(0)]]$ | $x(3)$ + $W_4^2 [x(2)$ + $W_4^2 [x(1)$ + $W_4^2 x(0)]]$ |
| 0 | $W_4^2 [x(3) + W_4^2 [x(2) + W_4^2 [x(1)$ + $W_4^2 x(0)]]]$ | |

Tableau 11 - Evolution des calculs du schéma pour X_2

$$X(2) = W_4^2 [x(3) + W_4^2 [x(2) + W_4^2 [x(1) + W_4^2 x(0)]]]$$

Partant de $W_4^2 = -j$, $X(2)$ s'écrit alors sous la forme suivante :

$$X(2) = j [x(3) - j[x(2) + j[x(1) - jx(0)]]]$$

$$X(2) = -x(3) + x(2) - x(1) + x(0)$$

Ce résultat pour $X(2)$ est identique au résultat obtenu par le calcul de la forme directe (4.8).

On opère de manière similaire pour le calcul de $X(3)$.

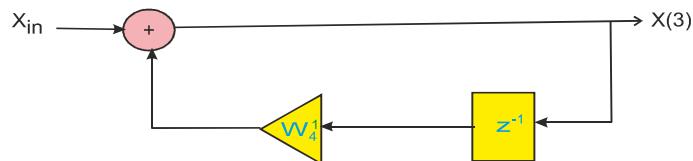


Figure 4-5- Modélisation des calculs pour X_3

Le gain est ici : $P = W_4^1 = 1$

3-Calcul de $X(3)$

| Echantillons | Accumulateur | Xout |
|--------------|---|---|
| $x(0)$ | 0 | $x(0)$ |
| $x(1)$ | $W_4^1 x(0)$ | $x(1) + W_4^1 x(0)$ |
| $x(2)$ | $W_4^1 [x(1) + W_4^1 x(0)]$ | $x(2) + W_4^1 [x(1) + W_4^1 x(0)]$ |
| $x(3)$ | $W_4^1 [x(2) + W_4^1 [x(1) + W_4^1 x(0)]]$ | $x(3) + W_4^1 [x(2) + W_4^1 [x(1) + W_4^1 x(0)]]$ |
| 0 | $W_4^1 [x(3) + W_4^1 [x(2) + W_4^1 [x(1) + W_4^1 x(0)]]]$ | |

Tableau 12- Modélisation des calculs pour X_3

On obtient :

$$X(3) = W_4^1 [x(3) + W_4^1 [x(2) + W_4^1 [x(1) + W_4^1 x(0)]]]$$

En tenant compte de la valeur de $W_4^3 = 1$, l'équation précédente devient :

$$X(3) = -jx(3) - x(2) + jx(1) + x(0)$$

Cette équation est identique à celle obtenue par le calcul classique de la TFD (4.9)

$$.X(3) = -jx(3) - x(2) + jx(1) + x(0)$$

En résumé, nous constatons que la structure du banc de filtres numériques évoqué est clairement établie et exposée dans les schémas ci-après.

Les fonctions de transferts sont indiquées pour chaque composante spectrale en sortie.

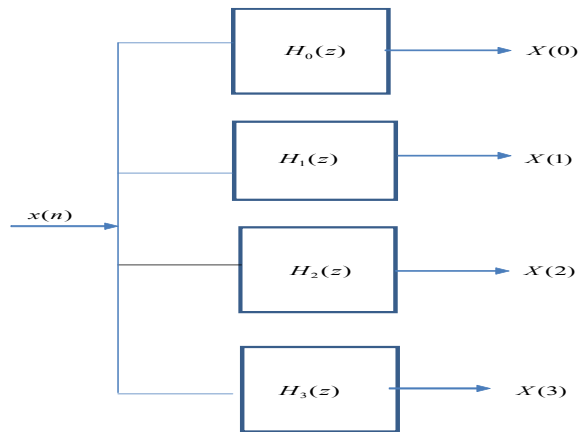


Figure 4-6- Schéma bloc du banc de filtres numériques $N = 4$

Le schéma du banc de filtres numériques a été simulé sous Simulink et les résultats mentionnés à la figure 13.

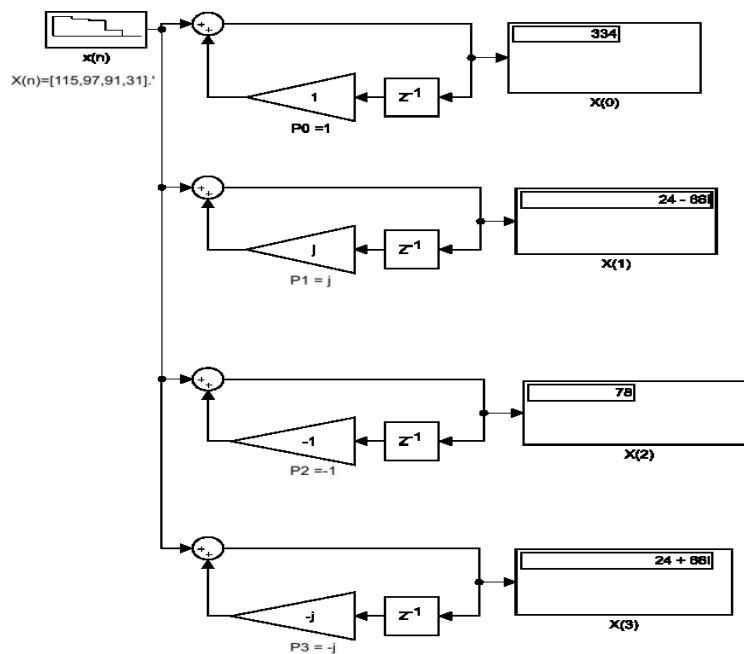


Figure 4-7- Détails du banc de filtres numériques $N = 4$

Il apparait clairement que le modèle est validé par les résultats obtenus sous Simulink montrés figure 4.7 et qui sont identiques à ceux calculés par le calcul direct de la FFT et par les méthodes FFT implémentées sous MATLAB.

4.1.3 GENERALISATION ET EXTENSION DE LA METHODE AMELIOREE

Nous pouvons étendre facilement la méthode à un nombre N quelconque d'échantillons. Les figures 4.6 et 4.7 se transforment en les figures 4.8 et 4.9.

L'angle θ qui va servir à mettre les pôles sur le cercle unité est (pour une suite à N échantillons):

$$\theta = \frac{2\pi}{N}$$

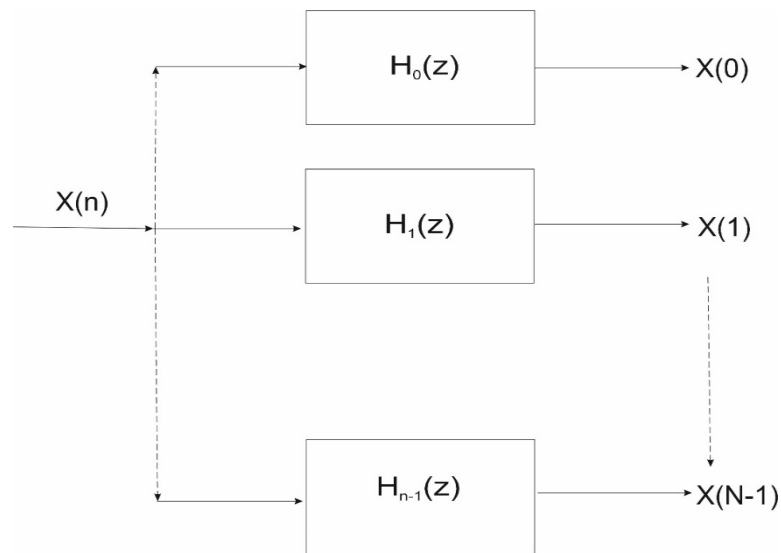


Figure 4-8- schéma bloc du banc de filtres numériques N quelconque

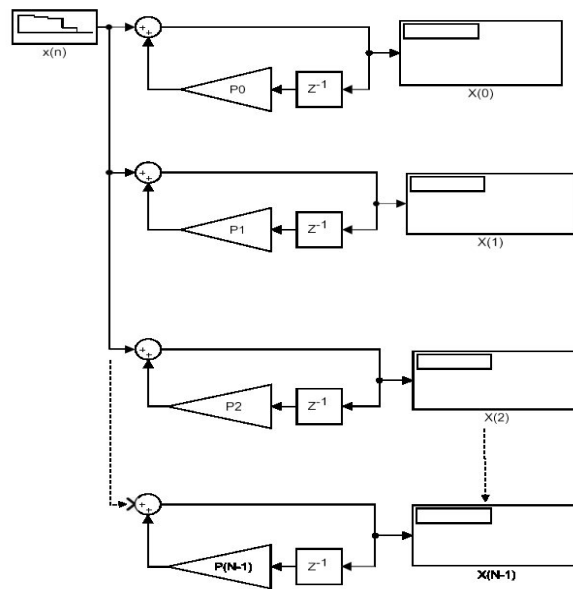


Figure 4-9- schéma détaillé du banc de filtres numériques N quelconque

Quelques cas particuliers sont exposés de manière détaillée. L'évolution de la sortie du schéma de base est suivie à l'arrivée de chaque échantillon. Le passage à un banc de filtres est assez facile. Il correspond simplement, comme déjà indiqué, à l'extension de la méthode à un nombre N quelconque d'échantillons. Les figures 4.6 et 4.7 se transforment en les figures 4.8 et 4.9.

Cela nous conforte dans la généralisation de la méthode des bancs de filtres.

1-Cas $N = 15$ et $k = 1$:

$$\mathbf{P}_1 = \mathbf{W}_N^{N-k} = \mathbf{W}_{15}^{14}$$

MODIFICATION ET AMELIORATION

| Echantillons | Accumulateur | Xout |
|--------------|---|--|
| $x(0)$ | 0 | $x(0)$ |
| $x(1)$ | $\mathbf{W}_{15}^{14}x(0)$ | $x(1) + \mathbf{W}_{15}^{14}x(0)$ |
| $x(2)$ | $\mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]$ | $x(2) + \mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]$ |
| $x(3)$ | $\mathbf{W}_{15}^{14}[x(2) + \mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]]$ | $x(3) + \mathbf{W}_{15}^{14}[x(2) + \mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]]$ |
| $x(4)$ | $\mathbf{W}_{15}^{14}[x(3) + \mathbf{W}_{15}^{14}[x(2) + \mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]]]$ | $x(4) + \mathbf{W}_{15}^{14}[x(3) + \mathbf{W}_{15}^{14}[x(2) + \mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]]]$ |
| $x(5)$ | $\mathbf{W}_{15}^{14}[x(4) + \mathbf{W}_{15}^{14}[x(3) + \mathbf{W}_{15}^{14}[x(2) + \mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]]]]]$ | $x(5) + \mathbf{W}_{15}^{14}[x(4) + \mathbf{W}_{15}^{14}[x(3) + \mathbf{W}_{15}^{14}[x(2) + \mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]]]]]$ |
| $x(6)$ | $\mathbf{W}_{15}^{14}[x(5) + \mathbf{W}_{15}^{14}[x(4) + \mathbf{W}_{15}^{14}[x(3) + \mathbf{W}_{15}^{14}[x(2) + \mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]]]]]]]$ | $x(6) + \mathbf{W}_{15}^{14}[x(5) + \mathbf{W}_{15}^{14}[x(4) + \mathbf{W}_{15}^{14}[x(3) + \mathbf{W}_{15}^{14}[x(2) + \mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]]]]]]]$ |
| $x(7)$ | $\mathbf{W}_{15}^{14}[x(6) + \mathbf{W}_{15}^{14}[x(5) + \mathbf{W}_{15}^{14}[x(4) + \mathbf{W}_{15}^{14}[x(3) + \mathbf{W}_{15}^{14}[x(2) + \mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]]]]]]]]]$ | $x(7) + \mathbf{W}_{15}^{14}[x(6) + \mathbf{W}_{15}^{14}[x(5) + \mathbf{W}_{15}^{14}[x(4) + \mathbf{W}_{15}^{14}[x(3) + \mathbf{W}_{15}^{14}[x(2) + \mathbf{W}_{15}^{14}[x(1) + \mathbf{W}_{15}^{14}x(0)]]]]]]]]]$ |

MODIFICATION ET AMELIORATION

| | | |
|---------|---|---|
| $x(12)$ | $ \begin{aligned} & \mathbf{W}_{15}^{14} [x(11) + [\mathbf{W}_{15}^{14} [x(10) \\ & + \mathbf{W}_{15}^{14} [x(9) + \mathbf{W}_{15}^{14} [x(8) \\ & + \mathbf{W}_{15}^{14} [x(7) \\ & + \mathbf{W}_{15}^{14} [x(6) \\ & + \mathbf{W}_{15}^{14} [x(5) \\ & + \mathbf{W}_{15}^{14} [x(4) \\ & + \mathbf{W}_{15}^{14} [x(3) \\ & + \mathbf{W}_{15}^{14} [x(2) \\ & + \mathbf{W}_{15}^{14} [x(1) + \mathbf{W}_{15}^{14} x(0)]]]]]]]]]] \end{aligned} $ | $ \begin{aligned} & x(12) + \mathbf{W}_{15}^{14} [x(11) \\ & + [\mathbf{W}_{15}^{14} [x(10) + \mathbf{W}_{15}^{14} [x(9) \\ & + \mathbf{W}_{15}^{14} [x(8) + \mathbf{W}_{15}^{14} [x(7) \\ & + \mathbf{W}_{15}^{14} [x(6) \\ & + \mathbf{W}_{15}^{14} [x(5) \\ & + \mathbf{W}_{15}^{14} [x(4) \\ & + \mathbf{W}_{15}^{14} [x(3) \\ & + \mathbf{W}_{15}^{14} [x(2) \\ & + \mathbf{W}_{15}^{14} [x(1) \\ & + \mathbf{W}_{15}^{14} x(0)]]]]]]]]]] \end{aligned} $ |
| $x(13)$ | $ \begin{aligned} & \mathbf{W}_{15}^{14} [x(12) + W_4^1 [x(11) \\ & + W_4^1 [x(10) + \mathbf{W}_{15}^{14} [x(9) \\ & + \mathbf{W}_{15}^{14} [x(8) \\ & + \mathbf{W}_{15}^{14} [x(7) + \mathbf{W}_{15}^{14} [x(6) \\ & + \mathbf{W}_{15}^{14} [x(5) + \mathbf{W}_{15}^{14} [x(4) \\ & + \mathbf{W}_{15}^{14} [x(3)x \\ & + \mathbf{W}_{15}^{14} [x(2) \\ & + \mathbf{W}_{15}^{14} [x(1) + \mathbf{W}_{15}^{14} x(0)]]]]]]]]]] \end{aligned} $ | $ \begin{aligned} & x(13) \\ & + \mathbf{W}_{15}^{14} [x(12) + W_4^1 [x(11) \\ & + W_4^1 [x(10) + \mathbf{W}_{15}^{14} [x(9) \\ & + \mathbf{W}_{15}^{14} [x(8) \\ & + \mathbf{W}_{15}^{14} [x(7) + \mathbf{W}_{15}^{14} [x(6) \\ & + \mathbf{W}_{15}^{14} [x(5) + \mathbf{W}_{15}^{14} [x(4) \\ & + \mathbf{W}_{15}^{14} [x(3)x \\ & + \mathbf{W}_{15}^{14} [x(2) \\ & + \mathbf{W}_{15}^{14} [x(1) \\ & + \mathbf{W}_{15}^{14} x(0)]]]]]]]]]] \end{aligned} $ |

MODIFICATION ET AMELIORATION

| | | |
|---------|--|---|
| $x(14)$ | $ \begin{aligned} & \mathbf{W}_{15}^{14}[x(13)] \\ & + \mathbf{W}_{15}^{14}[x(12) + W_4^1[x(11)] \\ & + \mathbf{W}_{15}^{14}[x(10) + \mathbf{W}_{15}^{14}[x(9)] \\ & + + \mathbf{W}_{15}^{14}[x(8)] \\ & + \mathbf{W}_{15}^{14}[x(7) + \mathbf{W}_{15_4}^{14^1}[x(6)] \\ & + \mathbf{W}_{15}^{14}[x(5) + \mathbf{W}_{15}^{14}[x(4)] \\ & + \mathbf{W}_{15}^{14}[x(3)x \\ & + \mathbf{W}_{15}^{14}[x(2) \\ & + \mathbf{W}_{15}^{14}[x(1) \\ & + \mathbf{W}_{15}^{14}x(0)]]]]]]]]]] \end{aligned} $ | $ \begin{aligned} & x(14) + \mathbf{W}_{15}^{14}[x(13)] \\ & + \mathbf{W}_{15}^{14}[x(12) + W_4^1[x(11)] \\ & + \mathbf{W}_{15}^{14}[x(10) + \mathbf{W}_{15}^{14}[x(9)] \\ & + + \mathbf{W}_{15}^{14}[x(8)] \\ & + \mathbf{W}_{15}^{14}[x(7) + \mathbf{W}_{15_4}^{14^1}[x(6)] \\ & + \mathbf{W}_{15}^{14}[x(5) + \mathbf{W}_{15}^{14}[x(4)] \\ & + \mathbf{W}_{15}^{14}[x(3)x \\ & + \mathbf{W}_{15}^{14}[x(2) \\ & + \mathbf{W}_{15}^{14}[x(1) \\ & + \mathbf{W}_{15}^{14}x(0)]]]]]]]]]] \end{aligned} $ |
| 0 | $ \begin{aligned} & \mathbf{W}_{15}^{14}[x(14) + \mathbf{W}_{15}^{14}[x(13)] \\ & + \mathbf{W}_{15}^{14}[x(12) + W_4^1[x(11)] \\ & + \mathbf{W}_{15}^{14}[x(10) + \mathbf{W}_{15}^{14}[x(9)] \\ & + + \mathbf{W}_{15}^{14}[x(8)] \\ & + \mathbf{W}_{15}^{14}[x(7) + \mathbf{W}_{15_4}^{14^1}[x(6)] \\ & + \mathbf{W}_{15}^{14}[x(5) + \mathbf{W}_{15}^{14}[x(4)] \\ & + \mathbf{W}_{15}^{14}[x(3)x \\ & + \mathbf{W}_{15}^{14}[x(2) \\ & + \mathbf{W}_{15}^{14}[x(1) \\ & + \mathbf{W}_{15}^{14}x(0)]]]]]]]]]] \end{aligned} $ | |

2- Cas $N = 16$ et $k = 1$:

$$P_1 = W_N^{N-k} = W_{16}^{15}$$

MODIFICATION ET AMELIORATION

| Echan- tillons | Accumulateur | Xout |
|-------------------|---|--|
| $x(0)$ | 0 | $x(0)$ |
| $x(1)$ | $\mathbf{W}_{16}^{15}x(0)$ | $x(1) + \mathbf{W}_{16}^{15}x(0)$ |
| $x(2)$ | $\mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]$ | $x(2) + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]$ |
| $x(3)$ | $\mathbf{W}_{16}^{15}[x(2) + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]]$ | $x(3) + \mathbf{W}_{16}^{15}[x(2) + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]]$ |
| $x(4)$ | $\mathbf{W}_{16}^{15}[x(3) + \mathbf{W}_{16}^{15}[x(2) + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]]]$ | $x(4) + \mathbf{W}_{16}^{15}[x(3) + \mathbf{W}_{16}^{15}[x(2) + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]]]$ |
| $x(5)$ | $\mathbf{W}_{16}^{15}[x(4) + \mathbf{W}_{16}^{15}[x(3) + \mathbf{W}_{16}^{15}[x(2) + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]]]]]$ | $x(5) + \mathbf{W}_{16}^{15}[x(4) + \mathbf{W}_{16}^{15}[x(3) + \mathbf{W}_{16}^{15}[x(2) + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]]]]]$ |
| $x(6)$ | $\mathbf{W}_{16}^{15}[x(5) + \mathbf{W}_{16}^{15}[x(4) + \mathbf{W}_{16}^{15}[x(3) + \mathbf{W}_{16}^{15}[x(2) + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]]]]]]]$ | $x(6) + \mathbf{W}_{16}^{15}[x(5) + \mathbf{W}_{16}^{15}[x(4) + \mathbf{W}_{16}^{15}[x(3) + \mathbf{W}_{16}^{15}[x(2) + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]]]]]]]$ |
| $x(7)$ | $\mathbf{W}_{16}^{15}[x(6) + \mathbf{W}_{16}^{15}[x(5) + \mathbf{W}_{16}^{15}[x(4) + \mathbf{W}_{16}^{15}[x(3) + \mathbf{W}_{16}^{15}[x(2) + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]]]]]]]]]$ | $x(7) + \mathbf{W}_{16}^{15}[x(6) + \mathbf{W}_{16}^{15}[x(5) + \mathbf{W}_{16}^{15}[x(4) + \mathbf{W}_{16}^{15}[x(3) + \mathbf{W}_{16}^{15}[x(2) + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)]]]]]]]]]$ |

MODIFICATION ET AMELIORATION

| | | |
|---------|--|--|
| $x(8)$ | $\begin{aligned} & \mathbf{W}_{16}^{15}[x(7) + W_{16}^{15}[x(6) + \mathbf{W}_{16}^{15}[x(5) \\ & \quad + \mathbf{W}_{16}^{15}[x(4) \\ & \quad + \mathbf{W}_{16}^{15}[x(3) \\ & \quad + \mathbf{W}_{16}^{15}[x(2) \\ & \quad + \mathbf{W}_{16}^{15}[x(1) \\ & \quad + \mathbf{W}_{16}^{15}x(0)]]]]] \end{aligned}$ | $\begin{aligned} & x(8) \\ & + \mathbf{W}_{16}^{15}[x(7) + W_{16}^{15}[x(6) \\ & + \mathbf{W}_{16}^{15}[x(5) + \mathbf{W}_{16}^{15}[x(4) \\ & + \mathbf{W}_{16}^{15}[x(3) \\ & + \mathbf{W}_{16}^{15}[x(2) \\ & + \mathbf{W}_{16}^{15}[x(1) \\ & + \mathbf{W}_{16}^{15}x(0)]]]]] \end{aligned}$ |
| $x(9)$ | $\begin{aligned} & \mathbf{W}_{16}^{15}[x(8) + \mathbf{W}_{16}^{15}[x(7) + W_{16}^{15}[x(6) \\ & \quad + \mathbf{W}_{16}^{15}[x(5) \\ & \quad + \mathbf{W}_{16}^{15}[x(4) \\ & \quad + \mathbf{W}_{16}^{15}[x(3) \\ & \quad + \mathbf{W}_{16}^{15}[x(2) \\ & \quad + \mathbf{W}_{16}^{15}[x(1) \\ & \quad + \mathbf{W}_{16}^{15}x(0)]]]]]] \end{aligned}$ | $\begin{aligned} & x(9) + +\mathbf{W}_{16}^{15}[x(8) \\ & + \mathbf{W}_{16}^{15}[x(7) + W_{16}^{15}[x(6) \\ & + \mathbf{W}_{16}^{15}[x(5) + \mathbf{W}_{16}^{15}[x(4) \\ & + \mathbf{W}_{16}^{15}[x(3) \\ & + \mathbf{W}_{16}^{15}[x(2) \\ & + \mathbf{W}_{16}^{15}[x(1) \\ & + \mathbf{W}_{16}^{15}x(0)]]]]]] \end{aligned}$ |
| $x(10)$ | $\begin{aligned} & \mathbf{W}_{16}^{15}[x(9) + +\mathbf{W}_{16}^{15}[x(8) \\ & \quad + \mathbf{W}_{16}^{15}[x(7) \\ & \quad + W_{16}^{15}[x(6) \\ & \quad + \mathbf{W}_{16}^{15}[x(5) \\ & \quad + \mathbf{W}_{16}^{15}[x(4) \\ & \quad + \mathbf{W}_{16}^{15}[x(3) \\ & \quad + \mathbf{W}_{16}^{15}[x(2) \\ & \quad + \mathbf{W}_{16}^{15}[x(1) \\ & \quad + \mathbf{W}_{16}^{15}x(0)]]]]]] \end{aligned}$ | $\begin{aligned} & x(10) + \mathbf{W}_{16}^{15}x(9) \\ & + \mathbf{W}_{16}^{15}[x(8) \\ & + \mathbf{W}_{16}^{15}[x(7) + W_{16}^{15}[x(6) \\ & + \mathbf{W}_{16}^{15}[x(5) + \mathbf{W}_{16}^{15}[x(4) \\ & + \mathbf{W}_{16}^{15}[x(3) \\ & + \mathbf{W}_{16}^{15}[x(2) \\ & + \mathbf{W}_{16}^{15}[x(1) \\ & + \mathbf{W}_{16}^{15}(0)]]]]]] \end{aligned}$ |
| $x(11)$ | $\begin{aligned} & \mathbf{W}_{16}^{15}[x(10) + \mathbf{W}_{16}^{15}[x(9) \\ & \quad + +\mathbf{W}_{16}^{15}[x(8) \\ & \quad + \mathbf{W}_{16}^{15}[x(7) \\ & \quad + W_{16}^{15}[x(6) \\ & \quad + \mathbf{W}_{16}^{15}[x(5) \\ & \quad + \mathbf{W}_{16}^{15}[x(4) \\ & \quad + \mathbf{W}_{16}^{15}[x(3) \\ & \quad + \mathbf{W}_{16}^{15}[x(2) \\ & \quad + \mathbf{W}_{16}^{15}[x(1) \\ & \quad + \mathbf{W}_{16}^{15}x(0)]]]]]] \end{aligned}$ | $\begin{aligned} & x(11) + \mathbf{W}_{16}^{15}[x(10) \\ & + \mathbf{W}_{16}^{15}[x(9) + \mathbf{W}_{16}^{15}[x(8) \\ & + \mathbf{W}_{16}^{15}[x(7) + W_{16}^{15}[x(6) \\ & + \mathbf{W}_{16}^{15}[x(5) + \mathbf{W}_{16}^{15}[x(4) \\ & + W_4^1[x(3) \\ & + \mathbf{W}_{16}^{15}[x(2) \\ & + \mathbf{W}_{16}^{15}[x(1) \\ & + \mathbf{W}_{16}^{15}x(0)]]]]]] \end{aligned}$ |

MODIFICATION ET AMELIORATION

| | | |
|---------|---|--|
| $x(12)$ | $\begin{aligned} & \mathbf{W}_{16}^{15}[x(11) + \mathbf{W}_{16}^{15}[x(10) \\ & \quad + \mathbf{W}_{16}^{15}[x(9) \\ & \quad + \mathbf{W}_{16}^{15}[x(8) \\ & \quad + \mathbf{W}_{16}^{15}[x(7) \\ & \quad + W_4^1[x(6) \\ & \quad + \mathbf{W}_{16}^{15}[x(5) \\ & \quad + \mathbf{W}_{16}^{15}[x(4) \\ & \quad + \mathbf{W}_{16}^{15}[x(3) \\ & \quad + \mathbf{W}_{16}^{15}[x(2) \\ & \quad + \mathbf{W}_{16}^{15}[x(1) \\ & \quad + \mathbf{W}_{16}^{15}x(0)]]]]]]] \end{aligned}$ | $\begin{aligned} & x(12) + W_{16}^{15}[x(11) \\ & \quad + \mathbf{W}_{16}^{15}[x(10) + \mathbf{W}_{16}^{15}[x(9) \\ & \quad + \mathbf{W}_{16}^{15}[x(8) \\ & \quad + \mathbf{W}_{16}^{15}[x(7) + W_{16}^{15}[x(6) \\ & \quad + \mathbf{W}_{16}^{15}[x(5) + \mathbf{W}_{16}^{15}[x(4) \\ & \quad + \mathbf{W}_{16}^{15}[x(3) \\ & \quad + \mathbf{W}_{16}^{15}[x(2) \\ & \quad + \mathbf{W}_{16}^{15}[x(1) \\ & \quad + \mathbf{W}_{16}^{15}x(0)]]]]]]] \end{aligned}$ |
| $x(13)$ | $\begin{aligned} & \mathbf{W}_{16}^{15}[x(12) + W_{16}^{15}[x(11) \\ & \quad + \mathbf{W}_{16}^{15}[x(10) \\ & \quad + \mathbf{W}_{16}^{15}[x(9) \\ & \quad + \mathbf{W}_{16}^{15}[x(8) \\ & \quad + \mathbf{W}_{16}^{15}[x(7) \\ & \quad + W_{16}^{15}[x(6) \\ & \quad + \mathbf{W}_{16}^{15}[x(5) \\ & \quad + \mathbf{W}_{16}^{15}[x(4) \\ & \quad + \mathbf{W}_{16}^{15}[x(3) \\ & \quad + \mathbf{W}_{16}^{15}[x(2) \\ & \quad + \mathbf{W}_{16}^{15}[x(1) \\ & \quad + \mathbf{W}_{16}^{15}x(0)]]]]]]] \end{aligned}$ | $\begin{aligned} & x(13) + \mathbf{W}_{16}^{15}[\mathbf{W}_{16}^{15} + [x(11) \\ & \quad + \mathbf{W}_{16}^{15}[x(10) + \mathbf{W}_{16}^{15}[x(9) \\ & \quad + \mathbf{W}_{16}^{15}[x(8) \\ & \quad + \mathbf{W}_{16}^{15}[x(7) + W_{16}^{15}[x(6) \\ & \quad + \mathbf{W}_{16}^{15}[x(5) + \mathbf{W}_{16}^{15}[x(4) \\ & \quad + \mathbf{W}_{16}^{15}[x(3) \\ & \quad + \mathbf{W}_{16}^{15}[x(2) \\ & \quad + \mathbf{W}_{16}^{15}[x(1) \\ & \quad + \mathbf{W}_{16}^{15}x(0)]]]]]]] \end{aligned}$ |

| | | |
|---|--|--|
| 0 | $ \begin{aligned} & \mathbf{W}_{16}^{15}[x(15)] \\ & + \mathbf{W}_{16}^{15}[x(14)] \\ & + \mathbf{W}_{16}^{15}[x(13)] + \mathbf{W}_{16}^{15}[x(12)] \\ & + \mathbf{W}_{16}^{15}[x(11)] + \mathbf{W}_{16}^{15}[x(10)] \\ & + \mathbf{W}_{16}^{15}[x(9)] + \mathbf{W}_{16}^{15}[x(8)] \\ & + \mathbf{W}_{16}^{15}[x(7)] + \mathbf{W}_{16}^{15}[x(6)] \\ & + \mathbf{W}_{16}^{15}[x(5)] + \mathbf{W}_{16}^{15}[x(4)] \\ & + \mathbf{W}_{16}^{15}[x(3)]x \\ & + \mathbf{W}_{16}^{15}[x(2)] \\ & + \mathbf{W}_{16}^{15}[x(1) + \mathbf{W}_{16}^{15}x(0)] \end{aligned} $ | |
|---|--|--|

4.2 CONCLUSION

Les résultats obtenus pour les cas $N = 4, 15, 16$ confortent cette généralisation par l'identité des résultats entre algorithme et calcul direct et valident donc l'extension à un nombre quelconque d'échantillons.

Nous voyons que l'algorithme permet de :

- Entrer les données dans l'ordre naturel
 $x(0), x(1), x(2), \dots, \dots, x(N - 1)$
- Sortir les données spectrales de façon parallèle dans l'ordre naturel
 $X(0), X(1), X(2), \dots, \dots, X(N - 1)$.
- Obtenir les sorties directement et simultanément

On voit que le choix judicieux du Vecteur Composant Principal et donc les positions des pôles sur le cercle unité telle que mentionnées (Figure 4.2 ; exemple $N = 4$) permettent de :

- S'affranchir des structures Butterfly utilisées dans les algorithmes de type Radix-r, qui deviennent rapidement très complexes quand N augmente. Un niveau de complexité plus élevé peut conduire à un schéma de configuration difficile sur cible FPGA. Plus de complexité implique aussi plus d'interconnexions qui sont beaucoup plus lentes en termes de latence sur FPGA que la logique primitive elle-même.
- S'affranchir du Bit-Reversal tout en conservant le bénéfice de l'entrée naturelle des données et une sortie parallèle directe des raies spectrales

5 CONCEPTION DU MULTIPLIEUR COMPLEXE

5.1 ARITHMÉTIQUE SUR FPGA

Dans l'arithmétique des ordinateurs, deux concepts fondamentaux ont une très grande importance :

- La représentation des nombres
- La réalisation des opérations algébriques

La décision d'utiliser la représentation en virgule fixe ou en virgule flottante doit être faite soigneusement, de préférence à une phase précoce dans le projet. En général, la représentation en virgule fixe s'exécute à des vitesses plus élevées et à faible consommation. Tandis que pour la représentation en virgule flottante à une gamme plus élevée, qui peut être attrayante pour les algorithmes les plus compliqués.

Notre choix s'est porté sur l'arithmétique en virgule fixe, car l'implantation d'applications de traitement numérique du signal dans un système embarqué requiert l'usage de l'arithmétique en virgule fixe et de minimiser le nombre de bits pour représenter les données. Cette action permet de réduire la surface et la consommation mais entraîne une perte de précision des calculs ; il est donc nécessaire de spécifier une contrainte de précision au niveau applicatif. Contrairement aux processeurs, les ASIC et les FPGA permettent une liberté totale sur les choix du nombre des unités arithmétiques et du nombre de bits pour les entrées/sorties de chacune. Ceci constitue un potentiel important pour l'optimisation d'architectures. Cependant l'optimisation est difficile car liée au problème du partage de ressources.

Dans un processeur de signal numérique, les multiplicateurs sont les composants clés, la multiplication étant l'une des fonctions fondamentales des opérations arithmétiques. Étant donné que la multiplication domine le temps d'exécution de nombreux algorithmes DSP il faut donc un multiplicateur à grande vitesse. Des opérations arithmétiques efficaces sont importantes pour atteindre les performances souhaitées dans de nombreux signaux en temps réel.

Par conséquent des efforts continus sont déployés pour améliorer les performances des multiplicateurs. Plusieurs méthodes ont été utilisées pour cela.

Par exemple pour effectuer un calcul de transformée de Fourier discrète, il est nécessaire d'utiliser des opérateurs de multiplication rapides. Donc pour développer un circuit avec une capacité de traitement élevée basée sur le mécanisme de calcul de multiplication, il est important de connaître les performances de ces opérateurs. En règle générale les critères les plus importants sont :

- La vitesse de multiplication qui est un paramètre très important car elle est déterminante en termes de performances à atteindre.

- La surface occupée en termes de ressources internes joue un rôle clé lors de la phase de mise en œuvre de l'architecture adoptée et le coût du circuit cible en dépend fortement donc.
- La consommation de puissance est également un critère de premier ordre

Ces trois critères, parmi d'autres, déterminent la nécessité d'utiliser une entité spécifique pour la topologie du multiplieur. Les techniques de multiplication rapide sont très nombreuses et se sont révélées complexes.

En général, pour améliorer la vitesse d'un multiplieur, il faut réduire le nombre de produits partiels en recodant le multiplieur et utiliser des blocs additionneurs. Dans la pratique, il existe plusieurs méthodes d'encodage du multiplieur.

Il y a la paire adjacente codante comme l'arbre de Wallace, le codeur différenciateur pur tel que l'algorithme de Booth [48],[49], le différenciateur modifié codant pour les 0 et les 1 isolés utilisés par la structure de Reitwiesner [50]. Dans ce contexte nous ne nous intéressons qu'à la structure standard SBPM (Standard Bit Parallel Base Complex Multiplication) car il s'agit de la technique la plus utilisée dans le domaine des opérations de multiplication rapide [51].

Nous exposons ci-après les multiplieurs série et les multiplieurs parallèle les plus utilisés dans l'arithmétique en virgule fixe.

5.2 MULTIPLIEURS SÉRIE

Le multiplicateur série est utilisé lorsque la zone et la puissance sont les plus importants et le retard peut être toléré. Dans celui-ci un additionneur est utilisé pour ajouter des produits partiels. Les entrées multiplicateur et multiplicande sont agencées de manière à se synchroniser avec le comportement du circuit. Selon la longueur du multiplicateur et du multiplicande les entrées peuvent être présentées à des taux différents. Ici deux signaux d'horloge sont utilisés. Un pour les données et un autre pour l'opération de réinitialisation. Le principal inconvénient de l'algorithme multiplicateur série n'est pas adapté aux grandes valeurs de multiplicateur et de multiplicande.

5.2.1 LA MULTIPLICATION DISTRIBUÉE

L'arithmétique distribuée pour (Distributed Arithmetic DA) est une technique importante pour les FPGA. Elle est énormément utilisée dans le calcul de la somme des produits :

$$y = \langle c, x \rangle = \sum_{n=0}^{N-1} c[n] \times x(n) \quad (5.1)$$

En plus de la convolution, la corrélation, TFD et la plupart de calcul numérique peut être représenté en tant que somme de produit (Sum Of Produits). Pour exécuter un cycle de filtrage, lorsqu'on utilise une unité arithmétique classique, on prend d'environ un N MAC cycles. Ce

temps peut être réduit avec le pipelining, mais peut néanmoins être extrêmement longtemps. Il s'agit d'un problème fondamental lorsque les multiplicateurs d'usage général sont utilisés. Pour de nombreuses applications DSP, la multiplication n'est pas nécessaire techniquement. Si les coefficients de filtre $c[n]$ sont connus a priori, alors techniquement, le terme de produits partiels $c[n]x[n]$ devient une multiplication par une constante. C'est une différence importante et est une condition préalable pour une conception d'une arithmétique distribuée DA. La première discussion de l'arithmétique distribuée remonte à un article par Croisier en 1973 [15] et DA a été popularisée par Peled and Liu [15]. Yiu [26] a étendu DA pour les nombres signés, et Kammeyer [26] et Taylor [26] ont étudié les effets de quantification dans les systèmes de l'arithmétique distribuée. Pour comprendre le paradigme de la conception de l'arithmétique distribuée, on considère la somme des produits produit scalaire ci-dessous :

$$\begin{aligned}
 y = \langle c, x \rangle &= \sum_{n=0}^{N-1} c[n] \times x(n) \\
 &= c[0]x[0] + c[1]x[1] + \dots + c[N-1]x[N-1]
 \end{aligned} \tag{5.2}$$

Supposons en outre que les coefficients $c[n]$ sont des constantes connues et $x[n]$ est une variable. Un système d'arithmétique distribuée non signé suppose que la variable $x[n]$ est représenté par :

$$\begin{aligned}
 x[n] &= \sum_{b=0}^{B-1} x_b[n] \times 2^b \\
 \text{avec } x_b[n] &\in [0,1],
 \end{aligned} \tag{5.3}$$

où $x_b[n]$ désigne le bit b de $x[n]$, soit le n ième élément de x . Le produit intérieur y peut, par conséquent, être représenté comme suit :

$$y = \sum_{n=0}^{N-1} c[n] \times \sum_{b=0}^{B-1} x_b[n] 2^b \tag{5.4}$$

Redistribution de l'ordre de sommation (d'où le nom arithmétique distribuée), les résultats dans :

$$\begin{aligned}
 y &= c[0](x_{B-1}[0]2^{B-1} + x_{B-2}[0]2^{B-2} + \dots + x_0[0]2^0) + \\
 &\quad + c[1](x_{B-1}[1]2^{B-1} + x_{B-2}[1]2^{B-2} + \dots + x_0[1]2^0) \\
 &\quad \vdots \\
 &\quad + c[N-1](x_{B-1}[N-1]2^{B-1} + x_{B-2}[N-1]2^{B-2} + \dots + x_0[N-1]2^0) \\
 &= (c[0]x_{B-1}[0] + c[1]x_{B-1}[1] + \dots + c[N-1]x_{B-1}[N-1])2^{B-1} \\
 &\quad + (c[0]x_{B-2}[0] + c[1]x_{B-2}[1] + \dots + c[N-1]x_{B-2}[N-1])2^{B-2}
 \end{aligned}$$

$$+ (c[0]x_0[0] + c[1]x_0[1] + \dots + c[N-1]x_0[N-1])2^0 \quad (5.5)$$

Ou bien sous une forme plus compacte :

$$y = \sum_{b=0}^{B-1} 2^b \times \sum_{n=0}^{N-1} \underbrace{c[n] \times x_b[n]}_{f(c[n] \times x_b[n])}$$

$$y = \sum_{b=0}^{B-1} 2^b \times \sum_{n=0}^{N-1} f(c[n] \times x_b[n]) \quad (5.6)$$

La figure 5.1 suivante illustre une telle structure.

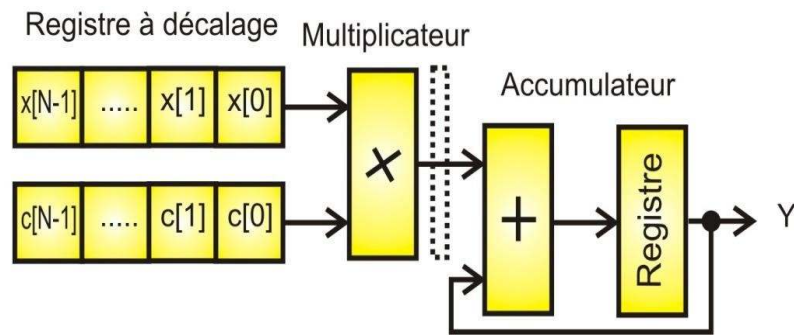


Figure 5-1- La structure d'une multiplication distribuée

Exemple :

Un produit scalaire de troisième ordre est défini par l'équation de produits scalaires suivant :

$$y = \langle c, x \rangle = \sum_{n=0}^2 c[n] \times x[n] \quad (5.7)$$

Supposons que $c[0]=2, c[1]=3, c[2]=1$. Le produit scalaires avec $x[n]=\{x[0]=1, x[1]=3, x[2]=7\}$, est obtenu par le calcul suivant :

$$y = \langle c, x \rangle = c[0]x[0] + c[1]x[1] + c[2]x[2]$$

$$= 2 \times 1 + 3 \times 3 + 1 \times 7 = 18$$

Pour remédier à toute utilisation des multiplicateurs qui sont gourmands en éléments logiques et en temps d'exécution ainsi que l'un des multipliant est une constante. L'implémentation sur FPGA de la fonction $f(c[n], x[n])$ nécessite une réflexion particulière.

La méthode de mise en œuvre préférée est de réaliser la cartographie $f(c[n], x[n])$ en utilisant une look-up tables (LUT). La LUT contient 2^N mots est préprogrammée pour accepter un vecteur d'entrée de N-bits $x_b=[x_b[0], x_b[1], \dots, x_b[N-1]]$, et une sortie $f(c[n], x_b[n])$. Les mappages

individuels $f(c[n], x[n])$ sont pondérés par la puissance appropriée de deux facteurs et accumulé. L'accumulation peut être implémentée efficacement comme il est indiqué dans figure 3.18. Après N cycles de calcul le résultat du produit scalaire est obtenu.

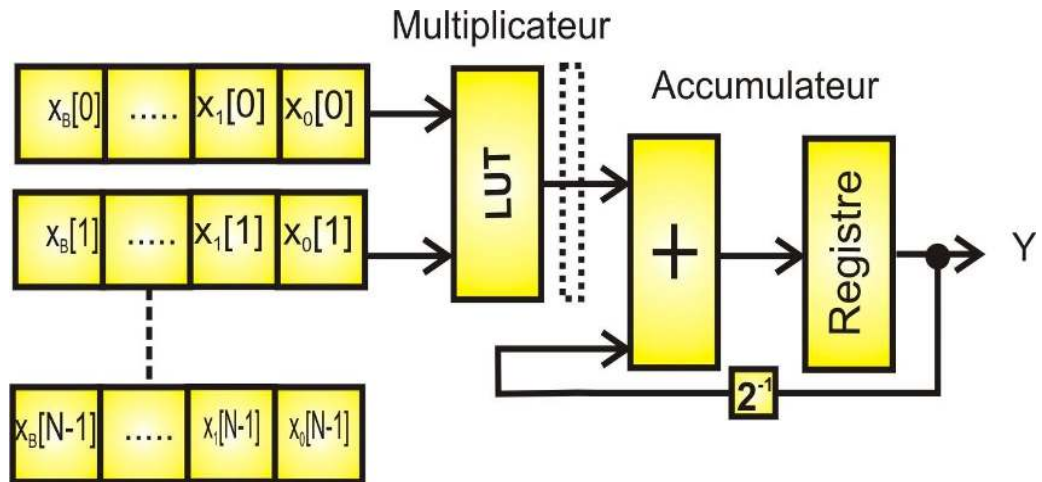


Figure 5-2- La structure d'une multiplication distribuée à base d'une LUT Look-Up Table Multiplication

La Multiplication utilisant une table de correspondance (LUT) est une alternative utilisée pour implémenter des opérations mathématiques sur FPGA, elle est tout simplement un bloc de mémoire contenant le résultat des multiplications complètes des deux opérandes. Il contient donc toutes les combinaisons possibles de cette multiplication. Les dimensions de la table requises sont considérables, et délicates, ce qui rend cette technique impraticable pour les FPGA de cette manière.

| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|-----|--------|--------|--------|--------|--------|--------|--------|--------|
| 000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 |
| 001 | 000000 | 000001 | 000010 | 000011 | 000100 | 000101 | 000110 | 000111 |
| 010 | 000000 | 000010 | 000100 | 000110 | 001000 | 001010 | 001100 | 001110 |
| 011 | 000000 | 000011 | 000110 | 001001 | 001100 | 001111 | 010010 | 010101 |
| 100 | 000000 | 000100 | 001000 | 001100 | 010000 | 010100 | 011000 | 011100 |
| 101 | 000000 | 000101 | 001010 | 001111 | 010100 | 011001 | 011110 | 100011 |
| 110 | 000000 | 000110 | 001100 | 010010 | 011000 | 011110 | 100100 | 101010 |
| 111 | 000000 | 000111 | 001110 | 010101 | 011100 | 100011 | 101010 | 110001 |

Tableau 13- Représentation du contenu d'une LUT à 6 entrées pour 3 x 3 bits

Pour une multiplication où un des multipliant est une constante, cette technique est beaucoup plus efficace sur FPGA. Il suffit de construire une table de multiplication qui ne dispose que d'une seule colonne correspondant à la valeur du résultat de la multiplication précalculée. Il est reconnu comme une multiplication par un coefficient constant (KCM). L'exemple ci-dessous

CONCEPTION DU MULTIPLIEUR COMPLEXE

multiplie une entrée à 5-bits (valeurs de 0 à 31) par une constante 67. Notons que si un multipliant est une constante, toutes les entrées sont disponibles sur LUT pour le multiplicande variable. Cela rend la KCM plus efficace qu'une multiplication complète (moins de produits partiels pour une largeur donnée). C'est une technique qui est recommandée sur FPGA car elle réduit le nombre des éléments logiques, elle réduit aussi le temps d'exécution, qui la rend plus efficace en traitement du signal.

| | Entrée (5 bits) * 67 | | | |
|--------|-----------------------|------|------|------|
| Entrée | 00 | 01 | 10 | 11 |
| 000 | 0 | 536 | 1072 | 1608 |
| 001 | 67 | 603 | 1139 | 1675 |
| 010 | 134 | 670 | 1206 | 1742 |
| 011 | 201 | 737 | 1273 | 1809 |
| 100 | 268 | 804 | 1340 | 1876 |
| 101 | 335 | 871 | 1407 | 1943 |
| 110 | 402 | 938 | 1474 | 2010 |
| 111 | 469 | 1005 | 1541 | 2077 |

Tableau 14- Exemple de la représentation de la LUT (entrée (5 bit) * 67)

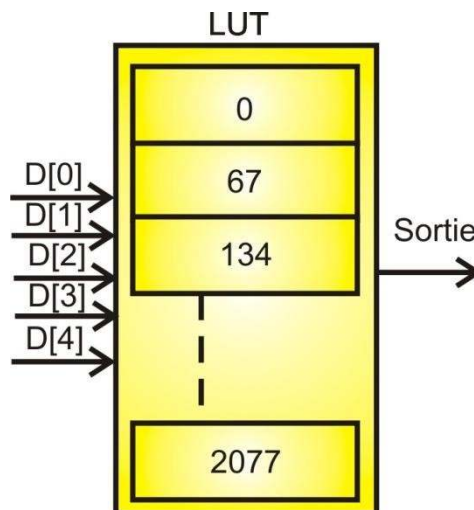


Figure 5-3- schéma bloc d'une LUT (Entrée (5 bits) * 67, sortie (15 bits))

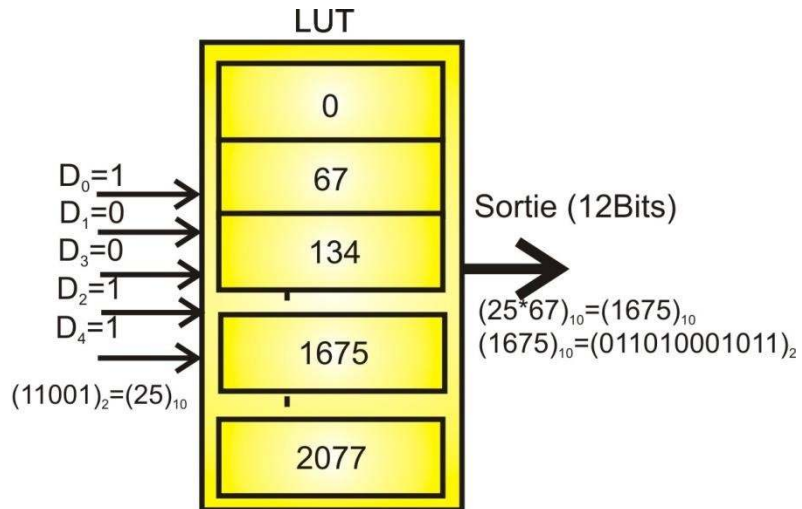


Figure 5-4- Le schéma bloc d'une LUT (Entrée = 25, sortie = 1675).

5.3 MULTIPLIEURS PARALLÈLES

5.3.1 MULTIPLIEUR DE BOOTH

Cette méthode a été donnée par Andrew Donald Booth. L'organigramme donné décrit la méthode.

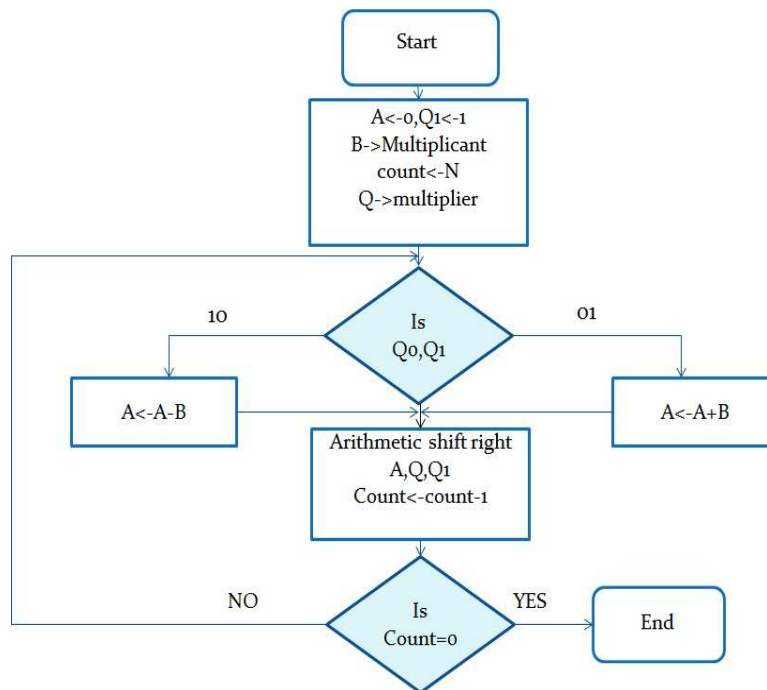


Figure 5-5-algorithme de Booth

De l'organigramme ci-dessus nous avons l'algorithme comme suit :

- Le multiplicateur et le multiplicande sont placés dans Q et B registres respectivement.
- Un registre Q (-1) à un bit est placé au bit le moins significatif de Q (0) du registre Q.
- Le résultat final apparaît en A.
- Les registres A et Q (-1) sont initialisés à 0.
- La multiplication de no est un cycle sur n.
- Dans chaque cycle Q (0) et Q (-1) sont examinés.
 - Si Q (0) et Q (-1) sont (1-1 ou 0-0) alors tous les bits des registres A Q et Q (-1) sont décalés de 1 bit vers la droite.
 - Si Q (0) et Q (-1) sont 01 alors le multiplicande est ajouté avec A. Après addition A Q et Q (-1) les registres sont décalés de 1 bit vers la droite.
 - Si Q (0) et Q (-1) sont 10 alors le multiplicande est soustrait de A. Après la soustraction les registres Q et Q (-1) sont décalés vers la droite de 1 bit.
- Le résultat final apparaît en A.

Le multiplicateur de Booth utilise l'algorithme de codage de Booth afin de réduire le nombre de produits partiels en considérant deux bits du multiplicateur à la fois obtenant ainsi un avantage de vitesse sur les autres architectures de multiplicateur. Cet algorithme est valable pour les nombres signés et non signés. Il accepte les nombres sous forme de complément à 2 basé sur le calcul de radix-2. La qualité d'une faible consommation d'énergie du multiplicateur en fait un choix préféré dans la conception de différents circuits. En implémentant le multiplicateur Radix-2 & Radix -4 et en utilisant l'algorithme de Booth, la vitesse de calcul augmente significativement.

5.3.2 MULTIPLIEUR VEDIC

Mohammed Hasmat, Ali Anil Kumar et Sahani ont présenté cet algorithme. Le multiplicateur présenté est basé sur un algorithme rdhva Tiryakbhyam (Vertical & Crosswise) des anciennes mathématiques indiennes védiques

En mathématiques védiques, deux des seize règles sont principalement utilisées pour le processus de multiplication. L'un est la règle Urdhva Tiryagbhyam (Urdhva Tiryagbhyam sutra) et l'autre est la règle Nikhilam Navatascaramam (Nikhilam Navatascaramam dasatahs). Urdhva-Tiryagbyam effectue l'opération de multiplication de deux nombres décimaux. Elle s'applique à tous les types de multiplication entre deux grands nombres. Il est également appelé algorithme vertical et transversal ("Vertically and crosswise algorithm").

Cela peut résoudre la multiplication d'un plus grand nombre ($N \times N$ bits) en le divisant en plus petites tailles. Le multiplicateur védique offre une vitesse améliorée par rapport au multiplicateur conventionnel et réduit la mémoire du système. Une très petite surface est nécessaire pour ce multiplicateur. Pour la multiplication de nombres binaires et décimaux, ce multiplicateur est utilisé exclusivement

Pour résoudre des calculs complexes par des techniques simples, les mathématiques védiques sont utilisées. La stratégie appliquée pour développer un multiplicateur védique 64 x 64 bits consiste à concevoir un multiplicateur védique 2 x 2 bits comme module de construction de base pour le système. De la même manière un multiplicateur védique 8 x 8, 16 x 16 et 32 x 32 bits est conçu. Ce type de multiplicateur joue un rôle très important dans les circuits numériques d'aujourd'hui.

La première étape de la multiplication est la multiplication verticale du LSB des deux multiplicandes puis la deuxième étape est la multiplication transversale et les ajouts des produits partiels. La troisième étape implique la multiplication verticale du MSB du multiplicande et l'addition avec le report propagé à partir de l'étape 2. Le multiplicateur védique offre une vitesse améliorée par rapport au multiplicateur conventionnel et réduit la mémoire du système. Une très petite surface est nécessaire pour ce multiplicateur par rapport à une autre architecture de multiplicateur. Pour la multiplication de nombres binaires et décimaux ce multiplicateur est utilisé et il est également utilisé dans la multiplication de nombres non signés et signés.

Le multiplicateur Vedic Urdhva est beaucoup plus efficace que le multiplicateur cellulaire (Array multiplier) et que le multiplicateur à coefficient constant (KCM) en termes de temps d'exécution. Le principal inconvénient de ce multiplicateur est que le système devient complexe pour des multiplications complexes [8]. Standard Bit Parallel base complex Multiplication

5.3.3 MULTIPLIEUR À ARBRE DE WALLACE

L'arbre de Wallace est une mise en œuvre matérielle efficace d'un circuit numérique qui multiplie deux entiers. Les arbres Wallace sont des structures irrégulières en ce que la description informelle ne spécifie pas une méthode systématique pour le compresseur interconnexions. Mais c'est quand même une mise en place de l'ajout de produits partiels en parallèle. En utilisant cette méthode, un processus en trois étapes est utilisé pour multiplier deux nombres entiers.

- La première étape consiste à multiplier chaque bit de l'un des arguments par chaque bit de l'autre donnant n^2 résultats. Sur la base de la position du bits multipliés les branches portent des poids différents.
- La deuxième étape consiste à réduire le nombre de produits partiels à deux par couches d'additionneur complet et de demi additionneur.
- La troisième étape consiste à regrouper les branches en deux nombres puis à les ajouter avec des additionneurs conventionnels.

Deux architectures différentes du multiplicateur à arbre de Wallace sont disponibles.

- La première est conçue en utilisant seulement des additionneurs et des demi-additionneurs.

- La deuxième utilise un processus plus sophistiqué : l'additionneur Carry Skip Adder (CSA).

Un multiplieur à arbre de Wallace utilisant uniquement des additionneurs complets et des demi-additionneurs est employé. Avec cette idée d'arbre Wallace, on cherche à réduire le nombre d'additionneurs en minimisant le nombre de demi-additionneurs dans n'importe quel multiplieur. Le premier produit partiel est le bit le moins significatif du résultat à la sortie du multiplicateur.

Après cela, on passe à la colonne suivante du produit partiel. S'il y a des additionneurs du produit partiel précédent, un additionneur complet est utilisé sinon un demi-additionneur est utilisé et ainsi de suite.

Un multiplieur à arbre de Wallace utilisant un Carry Skip Adder est également employé. Un additionneur Carry-Skip Adder se compose d'un simple Ripple Carry-Adder avec une vitesse spéciale. Le but de l'utilisation du CSA est d'améliorer le pire retard de trajet.

Le multiplicateur à arbre de Wallace utilisant un CSA occupe la plus petite surface tandis que le multiplicateur védique utilisant KSA consomme une grande surface. La consommation électrique des deux multiplicateurs sont convergents.

La conclusion est que les multiplieurs parallèles donnent davantage d'options que les multiplieurs série.

La surface totale est bien inférieure à celle des multiplicateurs en série. Par conséquent la consommation d'énergie est également inférieure.

Le multiplicateur d'arbre de Wallace présente de bonnes caractéristiques par rapport au multiplicateur cellulaire [Array multiplier]. Il a moins de dissipation de puissance statique et dynamique. Il a moins de retard et une bonne immunité au bruit.

Le multiplicateur à arbre de Wallace utilisant un additionneur complet et un demi-additionneur régulier a une surface minimale tandis que Vedic avec l'utilisation de KSA a une surface maximale. En termes de puissance, la consommation les deux architectures multiplicatives sont très proches. Le multiplicateur d'arbre Wallace a le plus petit retard de chemin critique par rapport au multiplicateur Vedic.

5.4 FONDEMENT MATHEMATIQUE DU MULTIPLIEUR COMPLEXE DE NOTRE ALGORITHME

Avant de se lancer dans la phase de mise en œuvre d'un tel composant et de choisir ensuite une architecture appropriée il est primordial de proposer une réduction ostensible en termes de ressources primitives afin d'améliorer de manière significative les performances souhaitées. En premier, considérons l'opération de multiplication fondamentale de deux quantités complexes.

$$P = (a + j \cdot b) \cdot (c + j \cdot s)$$
$$P = (a \cdot c - b \cdot s) + j(a \cdot s + b \cdot c)$$

Donc, cela donne la partie réelle :

$$Re(P) = a \cdot c - b \cdot s$$

et la partie imaginaire sera obtenue comme suit :

$$Im(P) = a \cdot s + b \cdot c$$

En fait ces combinaisons doivent utiliser au total quatre composants multiplieurs et deux blocs additionneurs pour réaliser cette opération. De plus, nous pouvons apporter une petite modification à la structure de cette équation pour construire un autre substitut en supprimant un composant multiplieur au profit d'un opérateur additionneur. Évidemment, cette nouvelle configuration nécessite trois multiplieurs combinés avec cinq blocs additionneurs pour effectuer l'opération de multiplication complexe. D'autre part, lorsque nous supposons qu'un terme complexe est une constante comme les facteurs twiddles dans le noyau de la FFT, le nombre d'additionneurs devient trois au lieu de cinq, ce qui permettra d'améliorer considérablement les performances de ce composant principal.

Première représentation

Seconde représentation

$$Re(P) = a \cdot c - b \cdot s + \overbrace{a \cdot s - a \cdot s}^{=0}$$

$$Im(P) = a \cdot s + b \cdot c + \underbrace{a \cdot c - a \cdot c}_{=0}$$

$$Re(P) = a \cdot c - b \cdot s + \overbrace{b \cdot c - b \cdot c}^{=0}$$

$$Im(P) = a \cdot s + b \cdot c + \underbrace{a \cdot c - a \cdot c}_{=0}$$

$$Re(P) = a \cdot (c + s) - s \cdot (a + b)$$

$$Im(P) = a \cdot (c + s) - c(a - b)$$

$$Re(P) = (a + b) \cdot c - (c + s) \cdot b$$

$$Im(P) = (a + b) \cdot c - (c - s) \cdot a$$

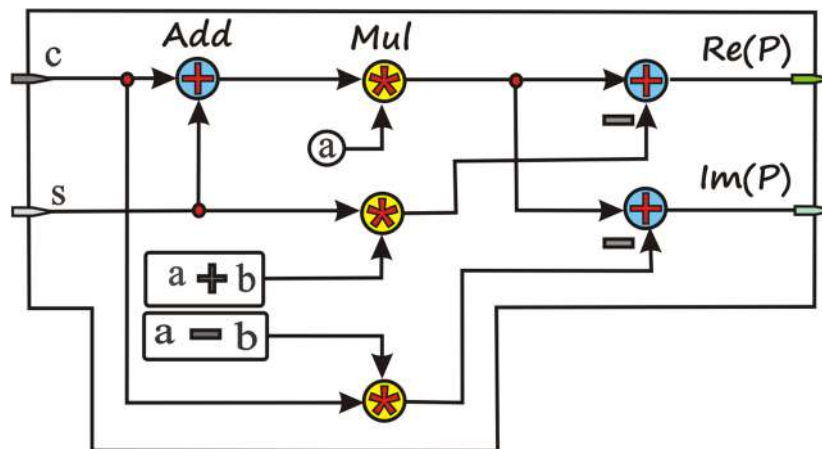


Figure 5-6- Diagramme du circuit multiplieur complexe

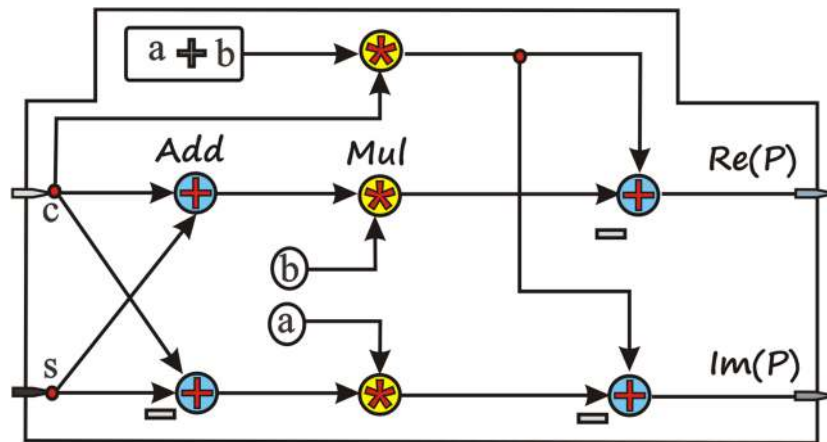


Figure 5-7- Seconde version du diagramme du circuit multiplieur basée sur la multiplication complexe

Globalement, cette configuration met en évidence le coût de ce type de multiplication en termes de composants, à savoir trois multiplieurs combinés avec quatre additionneurs. Ainsi un multiplieur a été supprimé et il a été remplacé par deux blocs d'addition concernant la multiplication classique. De ce fait, changer un bloc multiplieur va garder environ 15% de la superficie occupée par rapport au système standard à quatre multiplieurs. De plus la conception à trois multiplieurs est environ 2 % plus lente [52].

5.5 PROBLEME DE MODELISATION ET SIMULATION MATLAB

Pour bien comprendre le flux de notre approche, il convient d'élaborer une représentation graphique qui peut conduire à l'unité de traitement de DFT portée sur un échantillon spécifique en comptant sur le jeu demandé des données d'entrée. À partir du processus d'évolution de la matrice qui a été énuméré dans le Tableau 4 et en introduisant le diagramme schématique du multiplieur complexe tel qu'illustré à la Fig. 5.9, nous pouvons aisément décrire le circuit proposé concernant la première raie spectrale comme illustré à la Fig. 5.10.

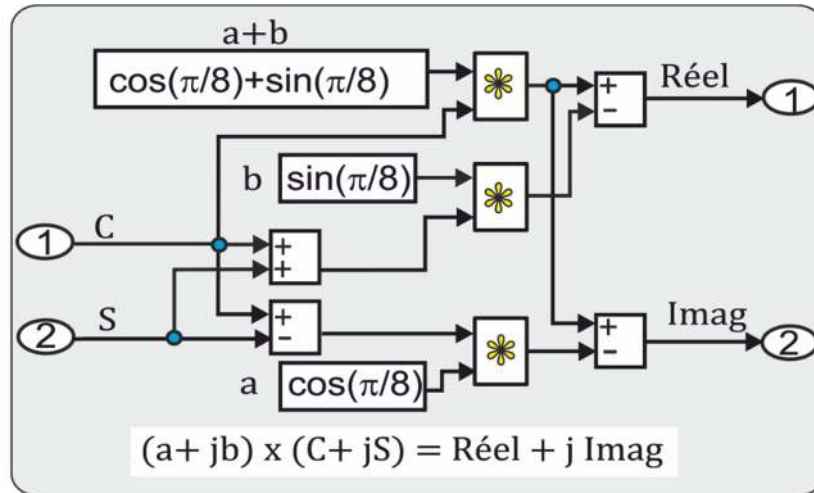


Figure 5-8 - Circuit Simulink du premier facteur twiddle du multiplieur complexe

Le résultat obtenu, après avoir appliqué cette approche sur la séquence d'entrée $x(n) = [115, 97, 91, 31, 63, 255, 153, 41, 21, 3, 135, 1, 109, 145, 59, 128]$ en conservant le premier facteur twiddle : $e^{\left(\frac{-j\pi}{8}\right)}$ du vecteur composant principal, est clairement décortiqué via le schéma fonctionnel du Fig. 14.

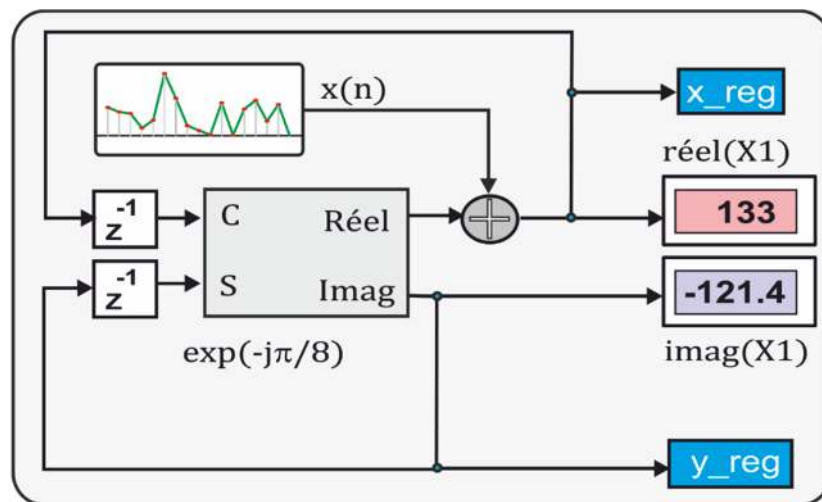


Figure 5-9 - Diagramme Simulink de l'élément processeur du premier échantillon spectral

5.6 IMPLEMENTATION HARDWARE DE L'ALGORITHME ADOPTE

Le grand défi et le lourd fardeau résident dans la façon de migrer de l'environnement Matlab vers une description hardware en introduisant une cible FPGA tout en veillant à réduire de façon significative la disproportionnalité entre le composant conçu et la simulation du modèle lui-même. Ainsi, le chemin à partir des outils de simulation à la représentation VHDL peut être résumé tel qu'indiqué dans la Fig. 5.11.

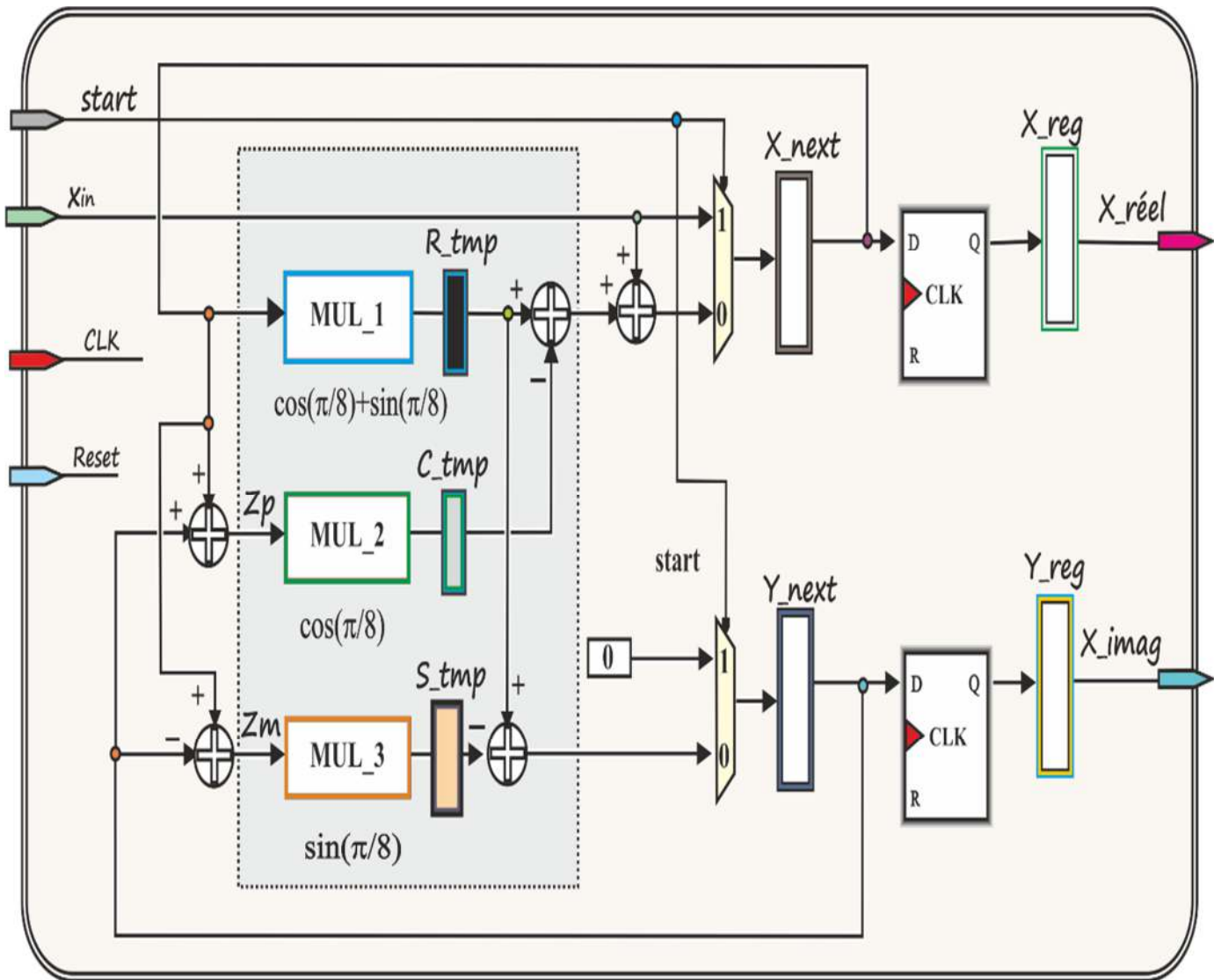


Figure 5-10- Schéma d'implémentation du composant principal

Cette architecture a un certain nombre de registres en pipeline où nous pouvons stocker les résultats intermédiaires ainsi que le multiplieur complexe et deux unités de retard pour les

CONCEPTION DU MULTIPLIEUR COMPLEXE

parties réelle et imaginaire. Donc, après activation du signal start, le composant passe de l'état initial à l'état de fonctionnement et y reste, en mode occupé, jusqu'à ce que le cycle compteur atteigne $(N + 1)$ cycles. A ce moment, le processus se met dans l'état done, puis revient à son état initial. Ainsi, un autre cycle sera réalisé.

Dans cette topologie les variables C et S désignent respectivement les parties de cosinus et de sinus qui sont évolutives. Par contre a et b sont des constantes qui expriment respectivement les parties réelle et imaginaire des facteurs twiddles comme indiqué à la figure 17.a, qui sont précalculées et utilisées directement. Il résulte de cette représentation que seuls cinq additionneurs sont nécessaires pour le circuit de calcul DFT de la première branche.

Le Script MATLAB exposé Figure 5.11 explique le fonctionnement du schéma d'implémentation du composant principal du Processeur Element Radix-16 complet.

```

clc;
clear all;
close all;

% Séquence des données en entrée
x = [115 97 91 31 63 255 153 41 21 3 135 1 109 145 59 128];

% mise à zéro des variables

zp=0;zm=0;R_tmp=0;C_tmp=0;S_tmp=0;Xreg=0;Yreg=0;

X_tmp=0;Y_tmp=0;

x=[115 97 91 31 63 255 153 41 21 3 135 1 109 145 59 128 0].';

format short g;

for k=1:17

X_tmp=Xreg/1024 ;
Y_tmp= Yreg/1024;
zp= X_tmp+Y_tmp;
zm = X_tmp-Y_tmp;

C_tmp= zp* 392;      % Représentation en point fixe
                    % de sin(pi/8)*1024 = 392

S_tmp= zm*946;      % Représentation en point fixe
                    % de cos(pi/8)

R_tmp= X_tmp*1338;  % Représentation en point fixe
                    % de (cos(pi/8)+sin(pi/8))

Xreg = R_tmp-C_tmp+x(k);
Yreg = R_tmp-S_tmp;

End

X=fft(x(1:16));

real(X(2)), Xreg, imag(X(2)),Yreg]

```

Figure 5-11 -Script Matlab expliquant le fonctionnement

6 RESULTATS DE LA CONFRONTATION ET DISCUSSION

Comme nous avons utilisé l'arithmétique en virgule fixe, certaines troncatures sont nécessaires pour conserver la plage dynamique des résultats et des sorties intermédiaires. Ces restrictions peuvent toucher la précision de l'algorithme FFT en introduisant le bruit de quantification. Il est donc important d'évaluer la précision du point fixe via la mesure du rapport du bruit signal-à-quantification (SQNR). Notre objectif dans ce cas est d'évaluer les répercussions de la troncature. En pratique nous n'avons considéré que la troncature pour les multiplieurs constants (pas pour les additionneurs). En effet, pour une multiplication avec un facteur twiddle codé en utilisant M bits, nous appliquons la troncature en utilisant M opérations de décalages à droite. En conséquence la résolution de la sortie de l'architecture FFT à N points proposée est évaluée à $N + 2\log_4(N)$ bits. D'ordinaire, le SQNR est défini par :

$$SQNR = \frac{\sum_{k=0}^{N-1} |X_k|^2}{\sum_{k=0}^{N-1} |X_{km} - X_k|^2}$$

où X_{km} désigne le résultat obtenu en utilisant la précision 64 bits de l'environnement Matlab

Selon les variations SQNR présentées à la Fig.6.1, on peut notamment montrer que pour une FFT de grande résolution, la valeur du SQNR diminue (la résolution du facteur twiddle est fixée à 10 bits). Ce phénomène est dû à la propagation du bruit de quantification. Enfin il convient d'indiquer que l'erreur quadratique moyenne maximale (MSE) obtenue en utilisant une FFT de 256 points est d'environ 2% [53].

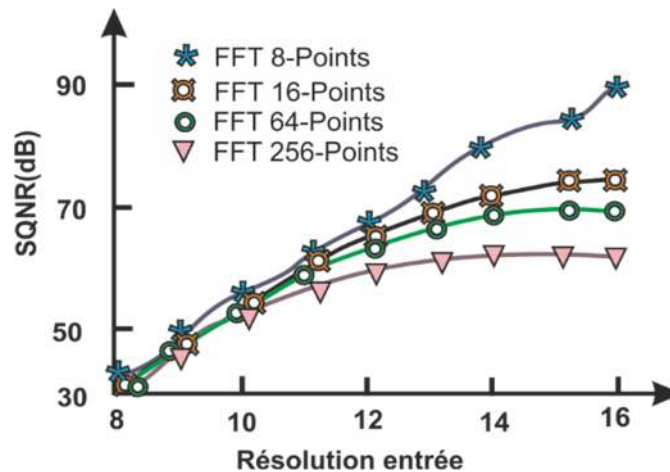


Figure 6-1- Evolution Du SNR pour différentes résolutions d'entrées et FFT de différentes résolutions

Les valeurs requises du facteur twiddle pour une multiplication pour l'algorithme proposé sont répertoriées dans le Tableau 6.2.

| Facteur Twiddle | Valeurs |
|-----------------|---------------------|
| W_{16}^0 | 1 |
| W_{16}^1 | $0.9238 - j0.3826$ |
| W_{16}^2 | $0.707 - j0.707$ |
| W_{16}^3 | $0.3826 - j0.9238$ |
| W_{16}^4 | $-j$ |
| W_{16}^5 | $-0.3826 - j0.707$ |
| W_{16}^6 | $-0.707 - j0.707$ |
| W_{16}^7 | $-0.9238 - j0.3826$ |

Figure 6-2-Valeurs des facteurs Twiddles pour une FFT-16 points

Il montre l'efficacité de cette approche par rapport à Radix-n FFT en termes de réduction du nombre de multiplications du facteur twiddle. Par exemple dans l'architecture FFT Radix-2 il faut 15 multiplieurs pour effectuer le calcul FFT [18],[19],[20]. Mais dans notre cas ; seuls 9 multiplieurs sont nécessaires pour effectuer le calcul FFT. De cette manière, l'approche proposée réduit la complexité matérielle et la consommation d'énergie pour la conception d'un processeur FFT. Les valeurs du facteur twiddle sont décrites dans le Tableau 6.2.

6.1 RESOLUTION ET MISE A L'ECHELLE :

De nombreux systèmes embarqués implémentent des applications de traitement du signal, notamment lors de communications et de processeurs FFT. Certains de ces traitements sont effectués par des filtres linéaires, qu'il est donc nécessaire de mettre en œuvre numériquement sur des cibles parfois FPGA. Les systèmes embarqués sont sujets à diverses contraintes qu'il faut optimiser tout en conservant des systèmes fiables en termes de performance et de précision. Comme précédemment indiqué, l'arithmétique virgule fixe est généralement préférée à l'arithmétique flottante pour des systèmes embarqués de traitement du signal, entre autres car elle est moins coûteuse, disponible dans tous les systèmes, permet d'utiliser des largeurs arbitraires sur des cibles matérielles et est généralement suffisante en termes de précision pour les applications de traitement du signal et de processeurs FFT. Le calcul en virgule fixe nécessite d'aligner les positions des virgules pour ainsi rendre cohérent des calculs à base de nombres entiers (pour les additions). Cela implique des quantifications et l'enjeu est donc de minimiser

la répercussion de ces arrondis sur le résultat final, en proposant une garantie sur l'erreur sur la sortie.

Certaines de ces applications nécessitant des calculs sur les nombres réels, de nombreuses questions se posent quant à la réalisation de ces calculs : avec quelle précision calculer ? à quel coût ? pour quelles performances ?

Pour estimer l'efficacité de notre algorithme, on se propose de calculer l'écart entre une valeur référence du facteur twiddle et la valeur obtenue en sortie par l'algorithme proposé.

On se place à titre d'exemple dans le cas du calcul de la partie réelle de la composante spectrale $X(1)$. L'angle $\theta = \frac{\pi}{8}$ est utilisé pour le calcul de la TFD.

La valeur de référence donnée par MATLAB pour $\cos\left(\frac{\pi}{8}\right)$ est $\cos\left(\frac{\pi}{8}\right) = 0.923879532511287$.

6.1.1 RÉOLUTION 8 BITS

La mise à l'échelle pour cette résolution se traduit alors par:

$$\cos\left(\frac{\pi}{8}\right) * 256 = 0.9238795325 * 256 = 236.513 \approx 237$$

Pour une résolution 8 bits, on peut écrire :

$$128 + 64 + 32 + 8 + 4 = 236$$

Cette expression permet d'approximer la valeur 237.

La méthodologie est de soustraire les plus grandes valeurs possibles de puissances de deux, de manière successive de $\cos\left(\frac{\pi}{8}\right) * 256 = 0.9238795325 * 256 = 236.513 \approx 237$.

On voit que de ce cette façon, multiplier ou diviser par 237 revient à multiplier ou diviser par des puissances de deux se traduit par des shifts de la valeur à multiplier. Cela permet de réaliser une SBMP (Standard Bit Parallel Base Complex) et d'implémenter des multiplieurs à base d'additionneurs uniquement, réalisant par là un gain important en termes de vitesse, de surface, de consommation d'énergie car on a éliminé les multiplieurs (gros consommateurs de ressources).

Les successions de soustractions des puissances de deux sont clairement indiquées dans le tableau suivant :

RESULTATS DE LA CONFRONTATION ET DISCUSSION

| Shifts | Valeurs soustraites de la valeur 237 | Restes |
|--|--------------------------------------|--------|
| $\frac{1}{2}$ | -128 | 109 |
| $\frac{1}{4}$ | -64 | 45 |
| $\frac{1}{8}$ | -32 | 13 |
| $\frac{1}{64}$ | -8 | 5 |
| $\frac{1}{256}$ | -4 | 1 |
| $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{64} + \frac{1}{256}$ $=$ 0.894531250000000 | 128+64+32+8+4=236 | |

Tableau 15 Soustractions des multiples de deux de 237

$$\cos\left(\frac{\pi}{8}\right) = 256 * \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{64} + \frac{1}{512}\right) = 0.894531250000000$$

On voit que pour une résolution 8 bits on a une perte de précision sur le calcul de $\cos\left(\frac{\pi}{8}\right)$. Une comparaison permet de donner la différence entre la valeur calculée et la valeur standard MATLAB.

- (a) $\cos\left(\frac{\pi}{8}\right) = 0.923879532511287$ Valeur standard MATLAB
- (b) $\cos\left(\frac{\pi}{8}\right) = 0.894531250000000$ Valeur calculée pour une résolution 8 bits

La différence est (b) - (a) = 0.029348282511287

6.1.2 RÉSOLUTION 10 BITS

Partant de 946 = 512 + 256 + 128 + 32 + 16 + 8 + 6

RESULTATS DE LA CONFRONTATION ET DISCUSSION

| Shifts | Valeurs soustraites de la valeur 946 | Restes |
|--|---------------------------------------|--------|
| $\frac{1}{2}$ | -512 | 434 |
| $\frac{1}{4}$ | -256 | 178 |
| $\frac{1}{8}$ | -128 | 50 |
| $\frac{1}{32}$ | -32 | 18 |
| $\frac{1}{64}$ | -16 | 2 |
| $\frac{1}{512}$ | -2 | 0 |
| $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{32} + \frac{1}{64} + \frac{1}{512}$ $=$ 0.92382812500000 | $512 + 256 + 128 + 32 + 16 + 2 = 946$ | |

Tableau 16 - Soustractions des multiples de deux de 946

Pour une résolution 10 bits, on peut écrire :

$$\cos\left(\frac{\pi}{8}\right) = 0.923879532511287 \text{ Valeur de référence MATLAB.}$$

De manière similaire au cas 8 bits, on procède au calcul de l'écart :

$$0.923879532511287 - 0.923828125 = 5.140751128673848 \times 10^{-5}$$

On voit que pour une résolution 10 bits on a une perte de précision sur le calcul de $\cos\left(\frac{\pi}{8}\right)$, bien moindre que celle observée dans le cas 8 bits.

La perte de précision est alors :

$$0.923879532511287 - 0.923828125 = 5.140751128673848 \times 10^{-5}$$

La structure de la multiplication SBPM (Standard Bit Parallel Base Complex) de $\cos\left(\frac{\pi}{8}\right) = 0.9238$ est illustré Fig.6.3. L'opération de multiplication d'un facteur twiddle de valeur $\cos\left(\frac{\pi}{8}\right)$ avec une valeur quelconque de donnée en entrée est réalisée comme suit :

$$in \times \cos\left(\frac{\pi}{8}\right) = in \times 0.9238 = in \times [2^{-1} + 2^{-2} + (2^{-2} + 1)(2^{-6} + 2^{-3})]$$

Le circuit multiplieur du facteur twiddle de valeur $\cos\left(\frac{\pi}{8}\right)$ requiert cinq opérations shift et quatre additionneurs.

6.2 STRUCTURE DES MULTIPLIEURS

6.2.1 MULTIPLIEUR (BIT PARALLEL) DU FACTEUR $\text{Cos}(\pi/8)$.

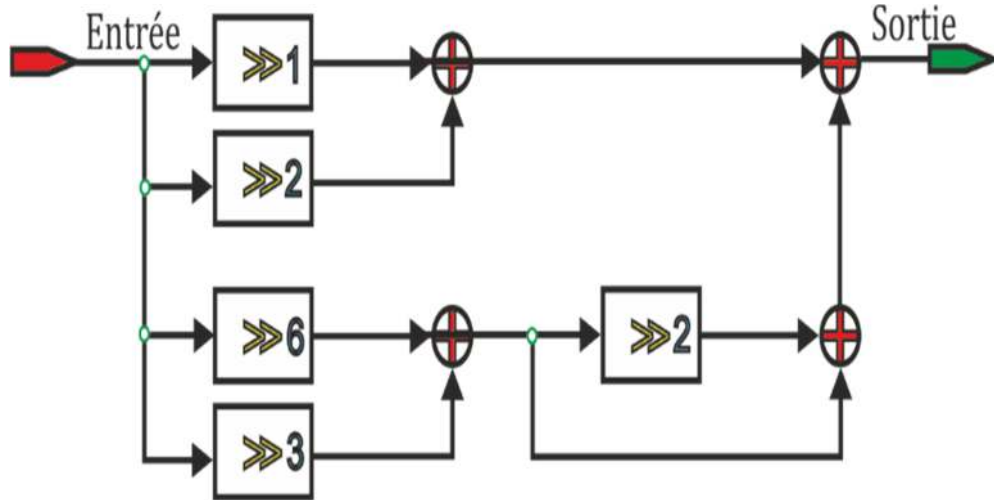


Figure 6-3- Multiplieur (Bit Parallel) du facteur $\text{Cos}(\pi/8)$.

De manière similaire, la multiplication du facteur twiddle $\text{Sin}(\frac{\pi}{8}) = 0.3826$ avec les données entrées est illustrée à la Fig 6.4. De manière significative, la multiplication de la valeur du facteur twiddle 0,3826 nécessite trois opérations de décalage et deux additionneurs pour effectuer cette multiplication. La forme pondérée de la valeur fractionnelle 0,3826 basée sur la notion de facteur commun peut réduire avantageusement la complexité algorithmique

$$in \times \text{Sin}(\frac{\pi}{8}) = in \times 0.3826 = in \times 2^{-2}[1 + 2^{-1} + 2^{-5}]$$

6.2.2 MULTIPLIEUR (BIT PARALLEL) DU FACTEUR $\text{Sin}(\pi/8)$

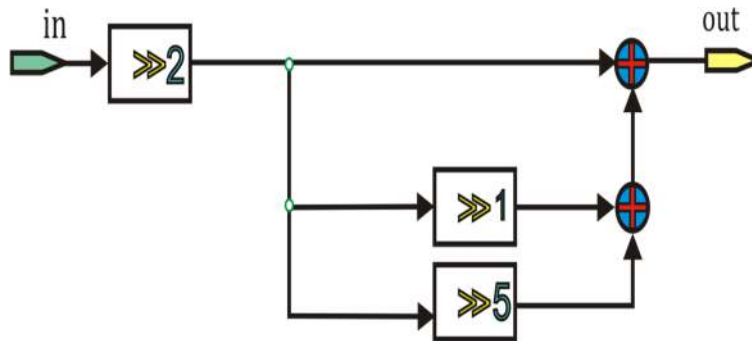


Figure 6-4- Multiplieur (Bit Parallel) du facteur $\text{Sin}(\pi/8)$

Finalement, le facteur twiddle de valeur $\frac{\sqrt{2}}{2} = 0.707$ peut être exprimé comme suit :

$$in \times \frac{\sqrt{2}}{2} = in \times 0.707 = in \times 2^{-1}[2^{-3} + (1 + 2^{-2})(1 + 2^{-5})]$$

Le circuit conçu nécessite au total quatre registres à décalage et trois blocs additionneurs pour effectuer l'opération de multiplication Fig. 6.5

6.2.3 MULTIPLIEUR (BIT PARALLEL) DU FACTEUR $\text{Cos}(\pi/4)$

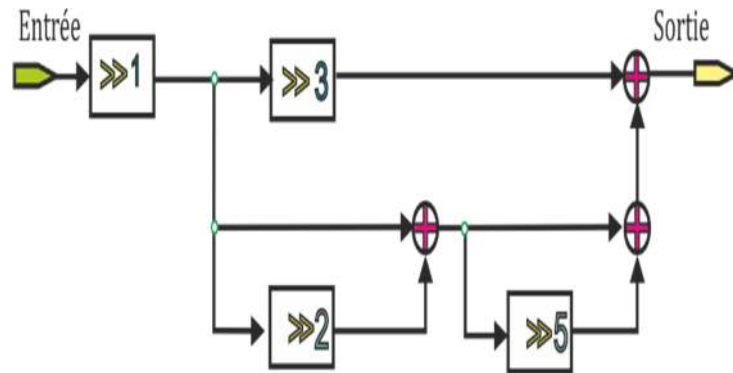


Figure 6-5- Multiplieur (Bit Parallel) du facteur $\text{Cos}(\pi/4)$

Par convention, la performance métrique est mesurée par le débit et la latence. La latence fait référence au nombre de cycles requis pour être prêt pour le prochain échantillon en entrée c'est-à-dire le temps qu'il faut pour effectuer une opération étant donné qu'il reçoit un flux de séquence de données en entrée. Parallèlement, le débit indique le nombre d'échantillons traités par cycle d'horloge ; c'est une somme de mesures pour un temps donné. Dans toutes les tâches de mise en œuvre, ce débit est égal au nombre d'échantillons pouvant être traités en parallèle.

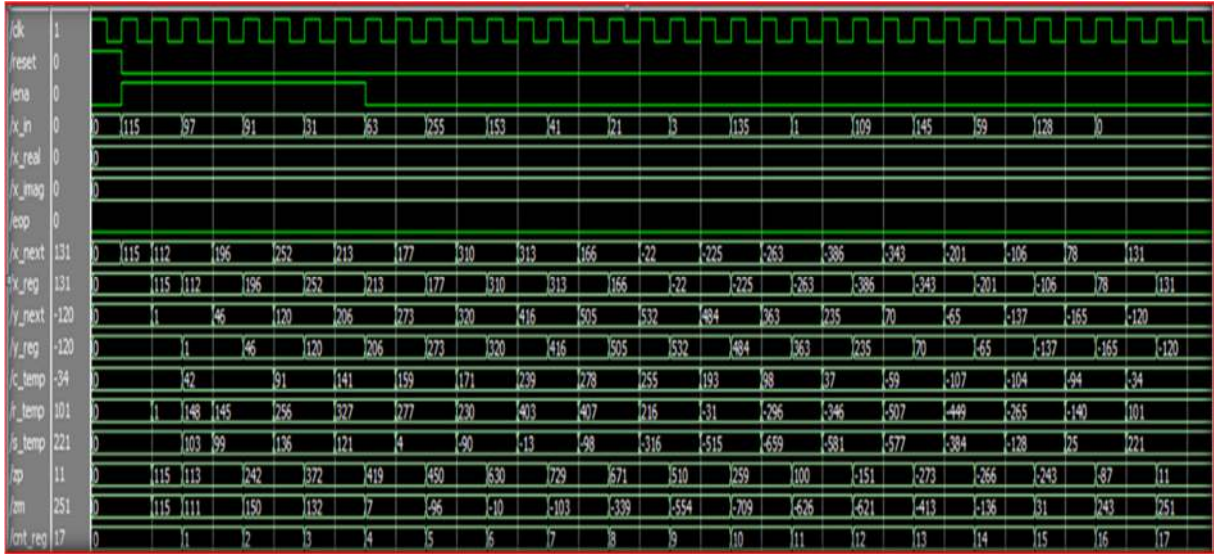


Figure 6-6- Diagramme du timing de l’algorithme proposé pour le calcul de la FFT pour l’échantillon X1

L'architecture suggérée a été élaborée pour les cibles FPGA. Les conceptions sont configurables en nombre d'échantillons à utiliser, en résolution du facteur twiddle et en structure de multiplication complexe (Parallel Base Complex Multiplication).

Le Tableau 6 présente les résultats après placement et routage pour différentes configurations de N (taille de la FFT) et en utilisant la résolution d'entrée (résolution $P = 8\ 10\ 12\ 14\ 16$ bits). Un code VHDL complet a été écrit pour tester l'implémentation sur une cible FPGA.

Le FPGA cible est un Xilinx Virtex-6 xc6vx75t-2ff484. Dans la conception proposée, ces blocs ont été utilisés pour mettre en œuvre des multiplieurs complexes qui réalisent le processeur unité de l'algorithme FFT.

Comme reportée dans le Tableau 17 et illustrée Fig. 6.7, la complexité de l'algorithme augmente lorsqu'une FFT avec un grand nombre de points N est utilisé la conception. Un niveau de complexité plus élevé peut conduire à un schéma de configuration difficile sur cible FPGA. Plus de complexité implique aussi plus d'interconnexions qui sont beaucoup plus lentes en termes de latence sur FPGA que la logique primitive elle-même. Lorsque la fréquence de fonctionnement maximale est augmentée, plusieurs registres de pipeline sont nécessaires pour satisfaire aux exigences du timing. Les registres supplémentaires ainsi que les dispositifs DSP pourraient figurer au lieu des blocs actuels. Dans ce cas, on peut augmenter la fréquence de fonctionnement et ainsi la performance est améliorée.

6.3 PERFORMANCES DE L'ALGORITHME PROPOSÉ

| Tailles FFT | Registres Slice | Latences (ns) | Cycles | MOF (MHz) | Débits (Ms/s) |
|----------------|--------------------|------------------|--------|-----------|---------------|
| 16 | 376 | 25.35 | 9 | 355 | 1893 |
| 64 | 658 | 77.91 | 21 | 263 | 1642 |
| 256 | 1034 | 213.63 | 43 | 200 | 1527 |
| 1024 | 1410 | 1000.75 | 183 | 183 | 1156 |
| 4096 | 2350 | 5898.79 | 165 | 165 | 864 |

Tableau 17 -Performances de l'algorithme proposé

RESULTATS DE LA CONFRONTATION ET DISCUSSION

À partir de ces valeurs répertoriées nous pouvons facilement découvrir que la fréquence de fonctionnement maximale est donnée par la relation :

$$(MOF = Cycles / latency)$$

Mais en ce qui concerne les valeurs de débit elles peuvent être exprimées en utilisant la représentation de la base 4 comme suit

$$Débit = \left(L_{fft} * \frac{MOF}{cycles} \right) * R_x$$

dans lequel R_x conçoit le coefficient de normalisation de la base 4 car nous avons utilisé ce schéma pour réduire l'effort de calcul lorsque nous avons manipulé la longueur de la FFT. Donc il devrait être exprimé comme suit $R_x = 4 * \left(\frac{Cycles}{L_{fft}} \right)$.

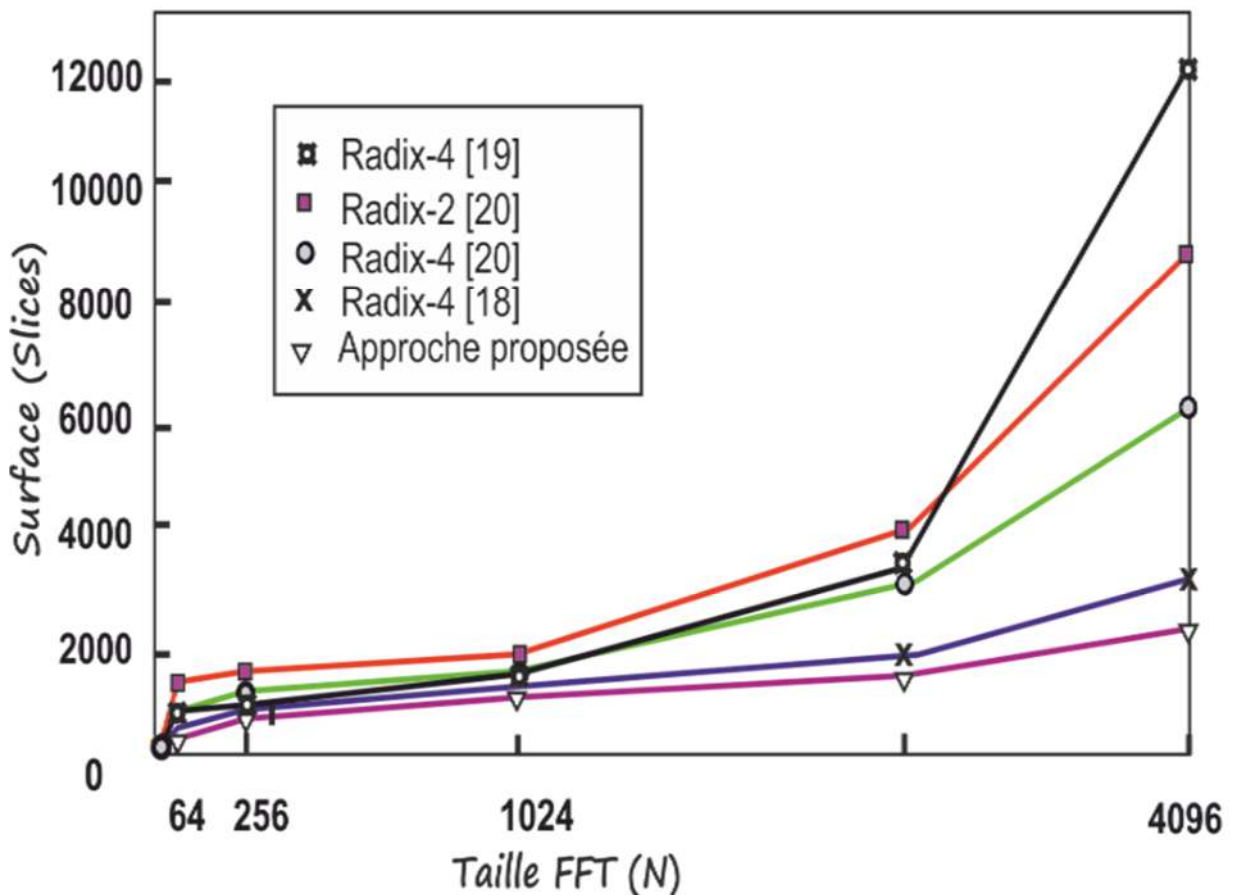


Figure 6-7- La surface occupée par l'algorithme confrontée à différents algorithmes

De la Fig.6.7, il peut être constatée que la conception proposée requiert moins d'espace que les autres architectures pour n'importe quelle taille de FFT, N. Cette amélioration augmente avec la taille de la FFT. Ainsi, cette figure montre que le concept proposé est clairement plus amélioré que les architectures de radix-4 et le plus grand N .

RESULTATS DE LA CONFRONTATION ET DISCUSSION

L'algorithme proposé requiert plus de registres en pipeline qui se traduit par une augmentation significative du nombre de slices utilisés pour le circuit de processeur FFT. L'algorithme proposé doit plus de registres en pipeline qui se traduit par une augmentation significative du nombre de tranches à servir pour le circuit de processeur FFT. En outre, la Fig.6.8 montre les comparaisons entre les débits de l'approche proposée et les diverses conceptions basées sur le concept Four-Parallel FFT en pipeline. Comme on peut l'observer, l'architecture proposée atteint le taux de transfert le plus élevé par rapport aux autres conceptions. En effet, un débit encore plus élevé peut être effectué en recourant à des topologies de radix-2 si le besoin se fait sentir, comme illustré dans cette figure 6.8.

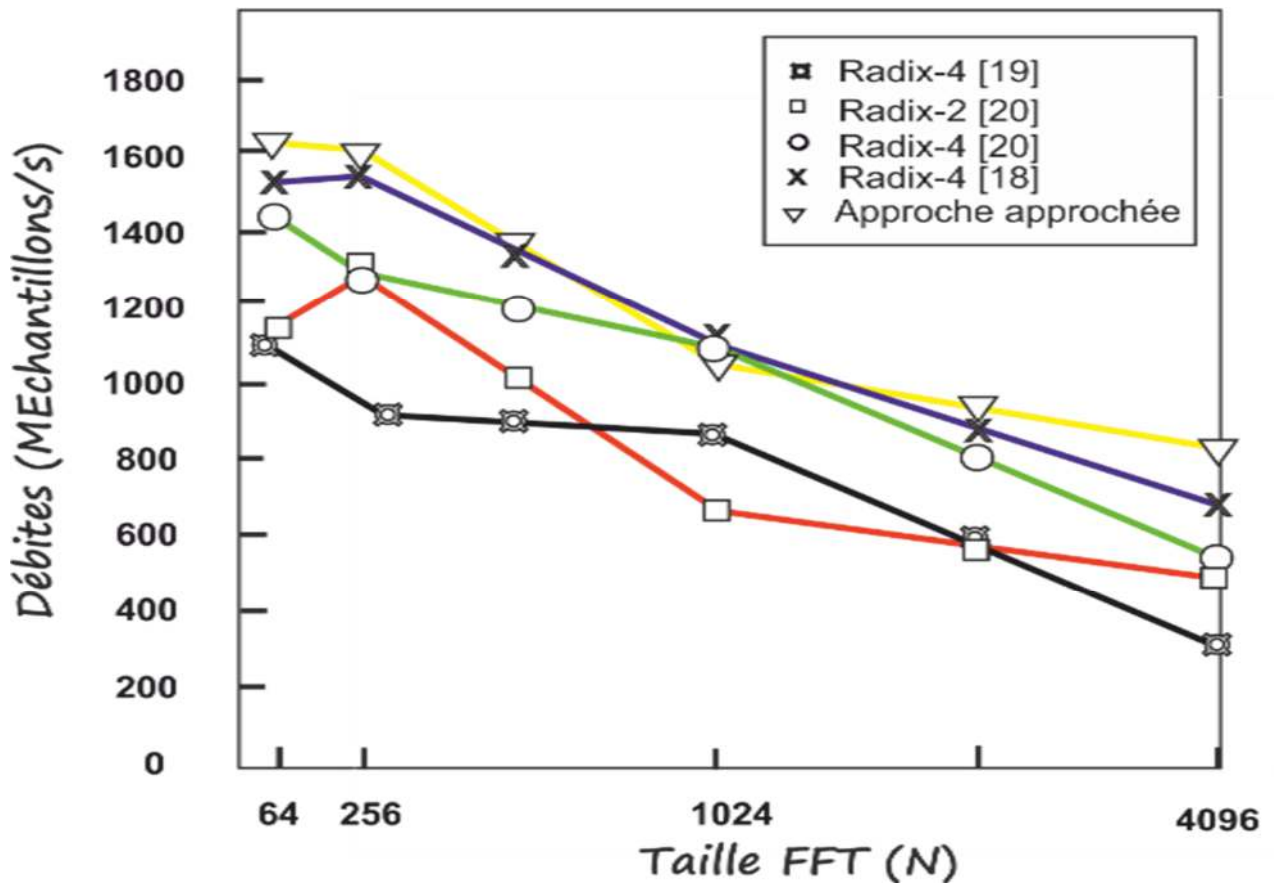


Figure 6-8- Evolutions des débits pour plusieurs approches

Cependant, la différence devient plus petite lorsque la fréquence de fonctionnement est augmentée, depuis la conception de radix-4 commence aussi à exiger plus de registres pipelines. Basés sur ces résultats, l'algorithme proposé exige ostensiblement moins de ressources que les architectures radix-4 jusqu'à 350 Mhz. En conséquence, la conception radix-4 reste une perspective prometteuse pour des fréquences de fonctionnement de près de 200 Mhz.

7 CONCLUSIONS ET PERSPECTIVES

Cette thèse étend (dans sa première partie) l'utilisation de Radix-k à une structure appropriée FFT 2D réarrangée à l'aide de la décomposition DIT (décimation en temps). Le modèle qui s'en suit est performant, implémentable et réalisable. Le modèle a été confronté avec succès à MATLAB et SIMULINK qui sont des références en la matière. Cependant deux inconvénients (opération Bit Reverse et Structure Butterfly non totalement éliminés, c.a.d la décimation en temps en entrée (décimateur splitter) et une forme de décimation en fréquence en sortie, nécessitant un générateur d'adresse comme sélecteur) nous ont poussé à repenser le processeur sous la forme d'un banc de filtres. Le modèle se simplifie considérablement et les deux inconvénients cités ont été balayés.

La mise en œuvre d'un multiplieur complexe innovant testé sous Modelsim et implémenté sous XILINX ISE sur une carte Virtex 6 permet en association avec les techniques SBPM de réduire les multiplieurs à des formes ne contenant que des additions et des shifts. Cela évite tous les inconvénients liés aux multiplieurs.

La passage d'une forme matricielle à une forme vectorielle, moyennant les notations et méthodes adoptées permet le choix du vecteur principal du composant de la matrice FFT 2D réarrangée, censé être une information révélant le comportement fréquentiel de la séquence d'entrée à étudier, joue un rôle clé dans la mise en œuvre optimale de cet algorithme.

Enfin, les résultats expérimentaux confirment que le modèle proposé est clairement efficace aussi bien en surface occupée, qu'en latence et en performances de débit, rendant possible l'obtention de débits plus élevés, autour de deux Giga échantillons/s, avec de faibles latences.

Les perspectives d'un tel travail sont larges.

En premier, elles portent sur l'algorithme lui-même :

- Cette forme est-elle optimale ? vu sa structure et sa simplicité, il semble que oui ! mais une étude mériterait d'être poussée à un stade plus avancé de manière à répondre à cette question.

En second, elles portent sur la carte choisie.

- Virtex 6 est très performante et contient un grand nombre de ressources. Mais depuis, des cartes de plus en plus performantes ont vu le jour en grand nombre (Zedboard, Zync, Spartan, Artix....). En implémentant sur de telles cartes, les performances s'en trouveront accrues, car elles sont plus rapides et plus optimisées, de sorte que certaines fonctions sont réalisées de façon plus compacte et donc plus optimisées . De plus XILINX ISE a été remplacé par VIVADO HLS. Ce dernier environnement permet des synthèses de schéma plus poussées et donc peut permettre des économies de ressources et un accroissement des performances.

CONCLUSIONS ET PERSPECTIVES

Enfin, il faut explorer les moyens tels que la création « automatique » d'accélérateur matériel pour différents types de carte.

8 BIBIOPHIE

- [1] S. Tertois, Réduction des effets des non linéarités dans une modulation multi-porteuses à l'aide de réseaux de neurones. Thèse de Doctorat, Université de Rennes 1, Décembre 2003.
- [2] H. Hijazi, Estimation de canal radio-mobile à évolution rapide dans les systèmes à modulation OFDM. Thèse de Doctorat, Institut Polytechnique de Grenoble, 2008.
- [3] D. Guel, Etude de nouvelles techniques de réduction du « facteur de crête à compatibilité descendante pour les systèmes multi-porteuses. Thèse de Doctorat, Université de Rennes 1, Décembre 2009.
- [4] S. M. J. Bilfagih, Transmission multiple porteuses utilisant un codage détecteur/correcteur d'erreur de type LDPC sur canaux MIMO : Détection multi-utilisateurs, Turbo égalisation, Diversité Temps, Espace, Fréquence. Thèse de Doctorat, Université de Limoges, Mars 2005.
- [5] A. Skrzypczak, Contribution à l'étude des modulations multi-porteuses OFDM/OQAM et OFDM suréchantillonnées. Thèse de Doctorat, Université de Rennes 1, Novembre 2007.
- [6] C. Lengoumbi, Accès multiple OFDMA pour les systèmes cellulaires post 3G : allocation de ressources et ordonnancement. Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications de Paris, Mars 2008.
- [7] A.R. S. Bahai and B. R. Saltzberg, Multicarrier Digital Communication theory and application of OFDM. Kluwer Academic Publishers, New York & Boston, 2002.
- [8] E. Gueguen, Etude et optimisation des techniques UWB haut débit multi-bandes OFDM. Thèse de Doctorat, Institut National des Sciences Appliquées de Rennes, Janvier 2009.
- [9] L. Hanzo et al., OFDM and MC-CDMA for broadband Multi-user communications, WLANs and broadcasting. John Wiley & Sons, Ltd, 2003.

BIBLIOGRAPHIE

- [10] C.R. Nassar et al., Multicarrier technologies for Wireless communication. Kluwer Academic Publishers, 2002.
- [11] H. H. Chen, The next generation CDMA technologies. John Willy & Sons, Ltd 2007.
- [12] H. Schulze and C. Lüders, Theory and Applications of OFDM and CDMA Wide-band Wireless Communications. John Wiley & Sons Ltd, 2005.
- [13] Y.G. LI and S. Gordan, Orthogonal frequency division multiplexing for Wireless communications. Springer science + business media, Inc 2006.
- [14] Geneviève BAUDOIN, Radiocommunications numériques, Tome 1: Principes, modélisation et simulation, Dunod Electronique, 2002.
- [15] Digital Communications, John G. Proakis Ed. Mac Graw Hill
- [16] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using mutiple antennas", Wireless Personal Communications, volume 6, pages 311-335, March 1998.
- [17] P.Guguen, Les Techniques Multi-Antennes pour les Réseaux sans Fil.ISBN2-74620883- 0,Hermes, Décembre 2004.
- [18] M. Doelz, E. Heald, and D. Martin, "Binary data transmission techniques for linear systems," Proceedings of the IRE, vol. 45, no. 5, pp. 656–661, May 1957.
- [19] R. R. Mosier and R. G. Clabaugh, "Kineplex, a bandwidth-efficient binary transmission system," American Institue of Electrical Engineers, vol. 76, pp. 723–728, 1958.
- [20] R. Chang and R. Gibby, "Synthesis of band-limited orthogonal signals for multi- channeldata transmission," Bell System Technical Journal, vol. 45, Dec. 1966.
- [21] S.Weinstein and P. Ebert, "Data transmission by frequency-division multiplexing using the discrete fourier transform," IEEE Transactions on Communications, vol. 628–634, no. 5, pp. 628–634, Oct. 1971.
- [22] Peled and A. Ruiz, "Frequency domain data transmission using reduced computational complexity algorithms," in International Conference on Acoustics,

BIBLIOGRAPHIE

Speech, and Signal Processing, Denver, 9-11 April 1980, 964-967.

[23] S. Wendt, P. Gelpi, D. Duponteil, " An Equivalent multipath Channel Chip Model for CDMA Systems ", IEEE ISCCSP, Mars 2004.

[24] Annick Le Glaunec " Modulations multiporteuses, " version à approfondir, 2001.

[25] ETS 300 421 : " Digital broadcasting systems for television, sound and data services, Framing structure, channel coding and modulation for 11/12 GHz services ", Dec 1999.

[26] Proakis, J.G.(2001). Digital communications (4th ed.). New York, N.Y: McGraw-Hill.

[27] H. A. Tai, Application des techniques multi-porteuses de type OFDM pour les futurs Thèse de Doctorat, Institut National Polytechnique de Toulouse

[28] R. Van Née et R. Prasad, "OFDM for Wireless Multimedia Communications" ,Artech Artech House Publishers, 2000.

[29] S. Kaiser et P. Robertson , The Effects of Doppler Spreads in OFDM a Mobile Radio Systems», IEE E Vehicular Technology Conférence , Vol. 1, pp . 329-333, septembre 1999.

[30] Arnaud Massiani, "Prototypage de Systèmes Haut Débit combinant étalement de Spectre,Multi-porteuses et Multi- antennes", Thèse de Doctorat, Institut National des Sciences Appliquées, Rennes, Novembre 2005

[31] V. Mannoni, Optimisation des codes LDPC pour les communications multi-porteuses.Thèse de Doctorat, Université de Reims champagne Ardenne, Juin 2004.

[32] B.Le Floch, M. Alard, and C. Berrou. "Coded orthogonal frequency division multiplex." IEEE Proceedings, 83(6):982-996, 1995.

[33] L. J. Cimini. "Analysis and Simulation of a Digital Mobile Channel Using Orthogonal Frequency Division Multiplexing.", IEE E Transactions on Communications , Vol. 33, No. 7, pp . 665-675, juillet 1985.

[34] J. J. van de Beek , P. Odling, S. K. Wilson et P. O . Borjesson , "Orthogonal Frequency Division Multiplexing (OFDM)" . International Union of Radio Science

(URSI), 2002.

[35] A. Latif, Hybrid QAM - FSK (HQFM) OFDM transceiver with low PAPR. Thèse de Doctorat, January, 2009.

[36] H. A. Tai, Application des techniques multi-porteuses de type OFDM pour les futurs systèmes de télécommunications par satellite. Thèse de Doctorat, Institut National Polytechnique de Toulouse, Mars 2009.

[37] Mohamed Arif , “A novel approach for DFT computation”, 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], 19-20 March 2015, **DOI:** 10.1109/ICCPCT.2015.7159528

[38] Arun C.A.; P.Prakasam, ”Design of high speed FFT algorithm For OFDM technique”, 2016 Conference on Emerging Devices and Smart Systems (ICEDSS), 4-5 March 2016, **DOI:** 10.1109/ICEDSS.2016.7587780

[39] J. Kwong and M. Goel, “A high performance split-radix FFT with constant geometry architecture,” 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE), DOI: 10.1109/DATE. 2012.6176717

[40] B V Uma; Harsha R Kamath ; S Mohith ; V Sreekar ; Shravan Bhagirath, “Area and time optimized realization of 16 point FFT and IFFT blocks by using IEEE 754 single precision complex floating point adder and multiplier”, 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI), **DOI:** 10.1109/ICSCTI.2015.7489546

[41] Ze-ke Wang; Xue Liu; Bingsheng He; Feng Yu, “A combined SDC-SDF architecture for normal I/O pipelined radix-2 FFT,” IEEE Transaction on Very Large Scale Integration (VLSI) Systems (Volume: 23, may 2015), **DOI:** 10.1109/TVLSI.2014.2319335

[42] Zhen-Guo Ma; Xiao-Bo Yin; Feng Yu, “A novel memory-based FFT architecture for real-valued signals based on a radix-2 decimation-in-frequency algorithm,” IEEE Transactions on Circuits and Systems II: Express Briefs (Volume: 62, Issue: 9, Sept. 2015), DOI: 10.1109/TCSII.2015. 2435522

[43] Hsin-Fu Luo; Yi-Ju Li; Ming-Der Shieh, “Efficient Memory-Addressing Algorithms for FFT Processor Design Sign In or Purchase”, IEEE Transactions on Very

BIBLIOGRAPHIE

Large Scale Integration (VLSI) Systems (Volume: 23, Issue: 10, Oct. 2015),
DOI: 10.1109/TVLSI.2014.2361209

[44] J.Greg Nash, “High-throughput programmable systolic array FFT architecture and FPGA implementations”, 2014 International Conference on Computing, Networking and Communications (ICNC), **DOI:** 10.1109/ICCNC.2014.6785453

[45] Mario Garrido; J. Grajel; M.A.Sanchez; Oscar Gustafsson, “Pipelined Radix-2k Feedforward FFT Architectures”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems (Volume: 21, Issue: 1, Jan. 2013),
DOI: 10.1109/TVLSI.2011.2178275

[46] Ranbeer Rathore , Navneet Kaur, “Comparison Study of DIT and DIF Radix-2 FFT Algorithm”, International journal of Computer Applications (0975 – 8887) Volume 150 – No.7, September 2016, DOI : 10.5120/ijca2016911565

[47] Trio Adiono; MuhSyafiqIrsyadi; Yan SyafriHidayat; Ade Irawan, “64-point fast efficient FFT architecture using Radix-23 single path delay feedback”, 2009 International Conference on Electrical Engineering and Informatics,
DOI: 10.1109/ICEEI.2009.5254734

[48] Tram Thi Bao Nguyen; Hanho Lee, “Shared CSD complex constant multiplier for parallel FFT processors”, 2015 International SoC Design Conference (ISOCC),
DOI: 10.1109/ISOCC.2015.7401648

[49] D. Villeger; V.G. Oklobdzija, “Evaluation of Booth encoding techniques for parallel multiplier implementation”, Electronics Letters 29(23):2016 – 2017,DOI: 10.1049/el:19931345

[50] Ms. Anuja a. bhat1 ; Prof. Mangesh n. Thakare, “Review on 32-bit IEEE 754 complex number multiplier based on FFT architecture using booth algorithm International Journal Of Engineering And Computer Science Volume 6 Issue 2 Feb. 2017, Pages 20308-20312, DOI: 10.18535/ijecs/v6i2.28

[51] Vojin G. Oklobdzija, “An integrated multiplier for complex numbers”, Journal for VLSI systems, Volume 7 Issue 3, October Pages 213-222 (2014),DOI: 10.1007/BF02409398

[52] Jian Wang; ChunlinXiong; Kangli Zhang; Jibo Wei, “Fixed-Point Analysis and Parameter Optimization of the Radix-2k Pipelined FFT Processor”, IEEE Transactions on Signal Processing (Volume: 63, Issue: 18, Sept. 2015),
DOI: 10.1109/TSP.2015.2447500

BIBLIOGRAPHIE

[53] Pankaj Gupta, “Accurate performance analysis of a fixed point FFT”, 2016 Twenty Second National Conference on Communication (NCC),**DOI:** 10.1109/NCC.2016.7561147

[54] Mario Garrido Gálvez, J Grajal, M A. Sanchez and Oscar Gustafsson, “Pipelined Radix-2(k) Feedforward FFT Architectures”, 2013, IEEE Transactions on Very Large Scale Integration (VLSI Systems), DOI: 10.1109/TVLSI.2011.2178275

[55] Miguel A. Sanchez; Mario Garrido; Marisa Lopez-Vallejo; Jesús Grajal, “Implementing FFT-based digital channelized receivers on FPGA platforms,” IEEE Transactions on Aerospace and Electronic Systems (Volume: 44, Issue: 4, Oct. 2008), **DOI:** 10.1109/TAES.2008.4667732

[56] Chao Wang ; Yuwei Yan ; Xiaoyu Fu, “A high-throughput low-complexity radix-24 -22 -23 FFT/IFFT processor with parallel and normal input/output order for IEEE 802.11ad systems,” IEEE Transactions on Very Large Scale Integration Systems (VLSI 2015), DOI: 10.1109/TVLSI.2014.2365586