

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY

UNIVERSITE BADJI MOKHTAR - ANNABA



جامعة باجي مختار- عنابة

Faculté: Sciences de l'ingénierat

Année : 2021

Département: Electronique

## THÈSE

Présentée en vue de l'obtention du diplôme de Doctorat 3<sup>ème</sup> Cycle

### Intitulé

**Conception et implémentation d'une technique de communication Numérique OFDM Sur une cible FPGA  
(OFDM Transceiver design with FPGA)**

Option : **Multimédia et Communications Numériques**

**Par : AOUISSI Samiha**

**DIRECTEUR DE THÈSE : Benouaret Mohamed Professeur Université de Annaba**

### DEVANT Le JURY

Président : Pr. DOGHMANE Nouredine U - Badji Mokhtar – Annaba

Examineur : Pr . FARROUKI Attef U – Constantine

Examineur : Pr LARBI Allal U - Badji Mokhtar – Annaba

Examineur : Pr FEZARI Mohamed U - Badji Mokhtar – Annaba

## **Remerciements**

*Cette thèse a été effectuée au sein du laboratoire d'Automatique et Signaux Annaba (LASA) au niveau de l'université de Badji Mokhtar-Annaba, faculté science de l'ingénierie, département d'électronique.*

*En premier lieu, je tiens à remercier profondément mon directeur de thèse, Monsieur BENOUARET Mohamed , professeur à l'Université d'Annaba pour les précieuses informations apportées tout au long du projet mais aussi pour ses conseils avisés et son optimisme à toute épreuve qui ont été de précieux atouts durant ces années , et surtout pour le temps qu'il m'a accordé et la patience dont il a fait preuve.*

*Je remercie plus particulièrement Madame HARIZ Saliha, , docteur à l'Université d'Annaba, pour les précieuses informations apportées tout au long du projet mais aussi pour ses conseils avisés et son optimisme à toute épreuve qui ont été de précieux atouts durant ces années. pour son aide précieuse, son intérêt pour ce travail, ses encouragements .*

*Mes remerciements vont également aux membres du jury pour l'honneur qu'ils m'ont fait en participant à l'évaluation de ce travail :*

*Monsieur . DOGHMANE Noureddine, professeur à l'Université d'Annaba.*

*Monsieur FARROUKI Attef , professeur à l'Université de Constantine.*

*Monsieur LARBI Allal , professeur à l'Université d'Annaba.*

*Monsieur FEZARI Mohamed, professeur à l'Université d'Annaba.*

*De manière plus personnelle, mes pensées les plus profondes et intimes vont vers ma famille*

*Je remercie mes collègues et amis qui sont toujours présents pour m'encourager : Wassila A, Boualag S, Chebout A, Ribahi h, Mahmoud.B , Meriem.O, Oulaya B, Nardjes.M, Meriem.B*

## Résumé

L'un des problèmes majeurs en télécommunications consiste à adapter l'information à transmettre au canal de propagation. En revanche, la modulation OFDM semble être l'alternative probante la plus adaptée et également l'approche préconisée pour résoudre un tel problème. Ainsi, la mise en œuvre du composant OFDM est destinée à répondre aux besoins imposés par la transmission à haut débit. D'autre part, ces besoins devraient être cohérents avec ce paradigme : performances métriques optimales, qualité de service attrayante et un coût de conception très réduit. La transformée de Fourier rapide (FFT) est parmi les pièces maîtresse de la modulation OFDM.

En effet, Il s'est avéré que le choix de l'algorithme de la FFT joue un rôle primordial lors de la phase de son implémentation en temps réel. Certes, le calcul de la FFT est une tâche qui sera souvent confiée à une architecture appropriée faisant intervenir le matériel reconfigurable ciblant les applications OFDM. Bien entendu, elle est considérée comme l'opération la plus gourmande en termes de temps et de ressources car elle réalise des opérations complexes, c'est-à-dire l'addition et la multiplication, ce qui plaide en faveur le choix adéquat du multiplieur pour satisfaire les besoins et répondre aux exigences imposées par cette application.

Ce travail traite principalement la conception et la mise en œuvre d'un processeur DFT avec des tailles spécifiques portant sur l'algorithme de Rader. Ainsi, la méthode suggérée visant l'implémentation du composant sus-indiqué sur une cible de type FPGA est parfaitement décortiquée et amplement décrite dans cette thèse, de même, les performances métriques sont évaluées et comparées. Pratiquement, la description est faite en ayant recours au langage VHDL et les tests réalisés à l'aide du logiciel ModelSim-Altera 6.3g.

**Mots clés :** OFDM, FFT, DFT, algorithme Rader, VHDL, FPGA

## **Abstract**

One of the major problems in telecommunications consists in adapting the information to be transmitted to the propagation channel. On the other hand, OFDM modulation seems to be the most suitable convincing alternative and also the recommended approach to solve such a problem. Thus, the implementation of the OFDM component is intended to meet the needs imposed by high-speed transmission. On the other hand, these needs should be consistent with this paradigm: optimal metric performance, attractive quality of service and a very low design cost. Among the core elements of OFDM is the Fast Fourier Transform (FFT).

Indeed, it turned out that the choice of the FFT algorithm plays a key role during the phase of its real-time implementation. Admittedly, calculating the FFT is a task that will often be left to an appropriate architecture involving reconfigurable hardware targeting OFDM applications. Of course, it is considered to be the most time-consuming and resource-intensive operation because it performs complex operations, i.e. addition and multiplication, which pleads in favor of the adequate choice of the multiplier to satisfy the needs and meet the requirements imposed by this application.

Of course, the calculation of the FFT is a task which will often be entrusted to an appropriate architecture involving the reconfigurable hardware targeting OFDM applications. It is considered to be the most time-consuming operation in terms of time and resources because it performs complex operations, i.e. addition and multiplication, which pleads in favor of the appropriate choice of the multiplier to meet the needs and meet the requirements imposed by this application.

This work mainly deals with the design and implementation of a DFT processor with specific sizes relating to the Rader algorithm. Thus, the suggested method aiming at the implementation of the above-mentioned component on a target of FPGA type is perfectly analyzed and amply described in this thesis; likewise, the metric performances are evaluated and compared. In practice, the description is made using the VHDL language and the tests carried out using the ModelSim-Altera 6.3g software.

**keywords :** OFDM, FFT, DFT, algorithm Rader, VHDL, FPGA

## ملخص

تتمثل إحدى المشكلات الرئيسية في الاتصالات في تكييف المعلومات لإرسالها إلى قناة الانتشار. من ناحية أخرى ، يبدو أن تعديل OFDM هو البديل الأكثر إقناعًا والأسلوب الموصى به لحل مثل هذه المشكلة. وبالتالي ، فإن تنفيذ مكون OFDM يهدف إلى تلبية الاحتياجات التي يفرضها الإرسال عالي السرعة. من ناحية أخرى ، يجب أن تكون هذه الاحتياجات متناسقة مع هذا النموذج: الأداء المتري الأمثل ، والجودة الجذابة للخدمة ، وتكلفة التصميم المنخفضة للغاية. يعتبر تحويل فورييه السريع (FFT) من بين العناصر الأساسية لتشكيل OFDM.

في الواقع ، اتضح أن اختيار خوارزمية FFT يلعب دورًا رئيسيًا خلال مرحلة التنفيذ في الوقت الفعلي. من المعروف أن حساب FFT هو مهمة غالبًا ما تُترك للمعماريات المناسبة التي تتضمن أجهزة قابلة لإعادة التشكيل تستهدف تطبيقات OFDM. بالطبع ، هذه هي العملية الأكثر استهلاكًا للوقت والموارد لأنها تؤدي عمليات معقدة مثل الجمع والضرب التي تتطلب الاختيار المناسب للمضاعف لتلبية الاحتياجات وتلبية المتطلبات التي يفرضها هذا التطبيق.

يتضمن هذا العمل بشكل أساسي تصميم وتنفيذ معالج DFT بأحجام محددة تتعلق بخوارزمية RADER. وبالتالي ، فإن الطريقة المقترحة التي تهدف إلى تنفيذ المكون المذكور أعلاه على هدف من نوع FPGA يتم تحليلها ووصفها بدقة في هذه الأطروحة ، وبالمثل ، يتم تقييم الأداء المتري ومقارنته. في الممارسة العملية ، تم إجراء الوصف باستخدام لغة VHDL والاختبارات التي تم إجراؤها باستخدام برنامج ModelSim-Altera 6.3g.

الكلمات المفتاحية: OFDM ، FFT ، DFT ، خوارزمية Rader ، VHDL ، FPGA

## Table des figures

Figure1.1: Comparaison entre le spectre de fréquence d'un system (a) FDM et (b)OFDM.....	8
Figure1. 2: Différentes utilisations des sous -porteuses d'un symbole OFDM .....	8
Figure 1.3: systèmes de communication numérique.....	9
Figure 1.4 : Schéma d'un système OFDM .....	10
Figure1. 5 : La bande passante occupée .....	11
Figure1. 6 : Spectre d'un signal OFDM .....	13
Figure 1.7 : Schéma de principe d'un modulateur OFDM.....	14
Figure 1.8: le schéma de la modulation BPSK.....	15
Figure1. 9: le schéma de la modulation QPSK.....	16
Figure 1.10: Effets du canal à trajets multiples sur des symboles reçus dans le cas (a) mono- porteuse (b) multi-porteuses.....	17
Figure 1.11: Illustration de l'effet de l'intervalle de garde précédent chaque OFDM.....	18
Figure1.12 présentation d'intervalle de garde de type (a) préfixe cyclique et (b) padding dans un symbole OFDM .....	19
Figure 1.13 : Interférence inter-porteuse (ICI) en OFDM dans les domaines :(a)temporel et (b) fréquentiel dans le cas d'un canal à deux trajets.....	20
Figure 2.1:Structure du papillon DIT .....	32
Figure 2.2 : Structure du papillon DIF.....	32
Figure 2.3 : Diagramme de la FFT à base de radix de type DIT.....	33
Figure 2.4 : Diagramme de la FFT à base de radix de type DIF.....	33
Figure 2.5 : Structure du papillon DIT radix -4 .....	34
Figure 2.6 : Exemple de FFT 8 points avec l'algorithme mixed-radix.....	34
Figure 2.7 : Signal- split-radix FFT de taille 8.....	35
Figure 2.8 Multiplicateur de tableau RCA (a) , CSA(b) et Multiplicateur de poids corporel(c).....	37
Figure 2.9 : Organigramme standard du multiplicateur Booth.....	39
Figure 2.10 : Élément de base Vedic 2x2 bits et Élément de base Vedic 8x8 bits.....	39

Figure 2.11:Rotation du vecteur V par l'angle $\Phi$ .....	39
Figure 3.1:Structure de base d'un circuit FPGA.....	44
Figure 3.2: Instruction de code VHDL.....	48
Figure 3.3:mise en œuvre d'une chaîne de transmission OFDM.....	49
Figure 3.4: Bloc de codage convolutif.....	50
Figure 3.5: Données codées par codage convolutif sous modelsim.....	50
Figure 3.6 : Exemple d'entrelacement.....	51
Figure 3.7 : Chronogramme d'entrelacement .....	51
Figure 3.8: Résultat de convertisseur binaire -décimale.....	52
Figure 3.9: Constellation de la modulation QAM16.....	53
Figure 3.10: Bloc mapping.....	53
Figure 3.11: Résultat de modulation QAM16.....	53
Figure 3.12 :Fenêtre de chargement de fonction et testbench.....	54
Figure 3.13 : Génération de code vhdl d' ifft.....	54
Figure 3.14 :Génération de testbench d'ifft.....	55
Figure 3.15 : top level de l'IFFT & Déclaration des donnes d'ifft 64.....	55
Figure 3.16: Résultat d'IFFT.....	55
Figure 3.17: partie réel et imaginaire du préfixe cyclique.....	56
Figure 3.18: Conversion série-parallèle.....	57
Figure 3.19 : Simulation de Conversion série-parallèle.....	57
Figure 3.20: Restitution de signal émis sous Matlab et Modelsim .....	57
Figure 4.1: Structure papillon de 2 point.....	61
Figure 4.2: Diagramme de bloc de multiplieur complexe.....	62
Figure 4.3 : Schéma de principe du bloc 8 points FFT.....	63
Figure 4.4. Résultat de simulation Modelsim pour un modèle de données.....	63
Figure 4.5 : Organigramme de l'algorithme CORDIC.....	65

Figure 4.6: Module CORDIC.....	66
Figure 4.7: Machine d'états de contrôleur.....	67
Figure 4.8: Diagramme de schéma de Registre _X.....	68
Figure 4.9: Diagramme de schéma de bloc ADD.....	68
Figure 4.10: Module principal d'FFT à base d'algorithme CORDIC.....	69
Figure 4.11: Résultat de Simulation Modelsim d'exemple d'angle ( $\Phi = 60^\circ$ ).....	69
Figure 4.12 : FFT à 2 Points avec CORDIC.....	70
Figure 4.13: Diagramme fonctionnel de l'algorithme de Rader avec un processus de mappage basé sur des racines primitives.....	73
Figure 4.14: Conception holistique de l'algorithme Rader.....	74
Figure 4.15. La structure du filtre FIR.....	74
Figure.4.16 : Le multiplicateur RAG des coefficients de filtre quantifiés.....	77
Figure. 4.17 Structure transposée du filtre proposé.....	78
Figure 4.18: Machine à états finis du contrôleur.....	79
Figure 4.19: Schéma de mise en œuvre de la cartographie d'index.....	80
Figure 4.20:Chronogramme de simulation de 7 point DFT.....	80

## Liste des tableaux

Tableau 3.1 Consommation des ressources de bloc de codage convolutif .....	50
Tableau 3.2 : Consommation des ressources de l'entrelacement.....	52
Tableau 3.3 : Consommation des ressources de 16QAM .....	53
Tableau 3.4 : Consommation des ressources d'IFFT .....	56
Tableau 4.1 : Comparaison des résultats fournis par Matlab et ceux obtenus auprès Modelsim.....	64
Tableau 4.2 : Tableaux des angles .....	66
TABLEAU 4.3 : EXEMPLE D'ANGLE ( $\Phi = 60^\circ$ ).....	70
Tableau 4.4 : Exemple d'FFT a 2 points.....	70
TABLEAU 4.5 : COMPARAISON ENTRE 2-POINT FFT AVEC ET SANS CORDIC.....	71
Tableau 4.6: coefficients quantifiés à 8 bits signé.....	76
Tableau 4.7: Mise en œuvre du filtre FIR proposé.....	76
Tableau 4.8 Chronogramme de l'opération de mappage d'index.....	78
Tableau 4.9: Comparaison des performances (Xilinx Virtex 6).....	81
Tableau 4.10 : Comparaison des performances de la mise en œuvre sur FPGA Xilinx.....	82
Tableau 4.11 : Comparaison des performances (Altera Cyclone II).....	82
Tableau 4.12 : Performances de latence pour différents algorithmes.....	83

## Liste des abréviations

## **A**

ASIC: Application Specific Integrated Circuit

## **B**

BPE :Butterfly Processing Element

BPSK: Binary Phase Shift Keying

## **C**

CLA: Carry look-ahead Adder

CSA: Carry Save Adder

CZT: Chirp Z-transform

CPLD: Complex Programmable Logic Device

## **D**

DA: Distributed Arithmetic

DFT - Discrete Fourier Transform

## **F**

FFT: Fast Fourier Transform

FIR : Finite Impulse Response

FDM : Frequency Division Multiplexing

FPGA; Field Programmable Gate Array

## **I**

IFFT: Inverse Fast Fourier Transform

## **L**

LUT : Look-Up Table

## **R**

RAM: Random Access Memory

RAG: Reduced adder graph

RCA: Ripple Carry adder

ROM: Read Only Memory

RTL : Register Transfer Level

## **O**

OFDM :Orthogonal Frequency Division Multiplexing

## **V**

VHDL: VHSIC Hardware Description Language

VM: Vedic Multiplier

## Q

QPSK: Quadrature Phase Shift Keying

QAM : Quadrature Amplitude Modulation

## Table des matières

Introduction générale

1

0.1 Contexte général du travail et problématique . . . . .	1
0.2- Objectifs et Principales contributions de la thèse. ....	2
0.3 Organisation de la thèse . . . . .	3
<b>1 Généralités sur la modulation OFDM</b>	<b>6</b>
1.1 Introduction . . . . .	7
1.2 Histoire et développement de l'OFDM.....	7
1.3 Structure du système de communication numérique.....	9
1.4 Contexte de modulation OFDM. ....	9
1.5 Notion d'orthogonalité.....	12
1.6 Implantation numérique du modulateur/démodulateur . . . . .	13
• 1.6.1 Principes de la modulation.....	15
1.7 Caractéristiques du signal OFDM.....	17
• 1.7.1 Intervalle de garde . . . . .	17
• 1.7.2 Interférence entre symboles . . . . .	18
• 1.7.3 Interférence entre sous porteuses.....	19
• 1.7.4. Synchronisation en OFDM . . . . .	21
1.8. Domaines d'application de l'OFDM . . . . .	21
1.9. Avantages d'OFDM.....	22
1.10 Conclusion.....	23
1.11 Référence de chapitre 1.....	24
<b>2 Étude théorique des algorithmes FFT</b>	
2 . 1 Introduction.....	28
2.2 Transformée de Fourier . . . . .	28
2.3 Transformée de Fourier Discrète . . . . .	29
2.4. La Transformée de Fourier Rapide . . . . .	30
2.5. Algorithmes conventionnels . . . . .	31

• 2.5.1 Radix-2 .....	31
• 2.5.2 Radix-4.....	34
• 2.5.3 Mixed radix .....	34
• 2.5.4 Split-Radix FFT.....	35
2.6. Autre Algorithmes de la transformée de Fourier discrète.....	35
2.7. Multiplicateurs disponibles.....	36
2.7.1 Multiplicateur de tableau .....	36
2.7.2 Multiplicateur de booth.....	38
2.7.3. Le multiplicateur Vedic.....	38
2.7.4. CORDIC .....	39
2.8 Conclusion .....	40
2.9 Référence de chapitre 2.....	41
<b>3 Simulation de la modulation OFDM</b>	
3.1.Introduction .....	43
3.2 Les circuits programmables .....	43
3.3 Les éléments d'un circuit FPGA .....	44
• a)Les éléments logiques.....	44
• b) Les éléments de mémorisation .....	45
• c) Les éléments de routage.....	45
• d) Les éléments d'entrées/sorties .....	45
• e) Les éléments de contrôle et d'acheminement des horloges.....	45
3.4. Les deux grandes familles des FPGA.....	45
• 3.4.1. La famille Xilinx.....	45
• 3.4.2. La famille Altera.....	46
3.5. Les outils de conception.....	45
•3.5.1 Le langage de description matérielle.....	46
•3.5.2. Modelsim et Quartus II .....	48
•3.5.3. Logiciel Xilinx ISE .....	48
3.6. Simulation de la chaine de transmission OFDM .....	49
3.7. Conclusion .....	58

3.8 Référence de chapitre 3.....	59
<b>4 Méthodologies d'implémentation d'FFT et discussion des résultats.</b>	
4.1 Introduction .....	60
4.2 Multiplicateur complexe .....	60
4.3 L'architecture proposée de 8 points FFT .....	62
4.4 Synthèse matérielle pour FFT à 8 points .....	64
4.5. L'architecture proposée d 'FFT à base d'algorithme CORDIC .....	64
• 4.5.1 Résultats et discussion .....	69
4. 6. L'architecture proposée d 'FFT pour des taille premier à base d'algorithme Rader .....	71
• 4.6.1 L'algorithme Rader .....	71
• 4.6.2 Structure de filtre .....	72
• 4.6.3 Méthodologie d'implémentation de processeur DFT .....	74
• 4.6.4. Résultats et discussion .....	80
4.7. Conclusion.....	83
4.8 Référence de chapitre 4.....	85
<b>Conclusion et perspectives</b> .....	87

# Introduction générale

## 0.1 -Contexte général du travail et problématique

Les domaines des télécommunications, de transmission des informations ont connu une évolution fulgurante. Ainsi, leurs performances s'améliorent chaque jour pour répondre aux exigences de plus en plus élevées en termes de qualité et temps de traitement [1]. En effet, La modulation **OFDM** (**O**rtogonal **F**requency **D**ivision **M**ultiplexing) est un type particulier de transmissions multi-porteuses dont l'originalité est de multiplexer l'information sur des sous-porteuses orthogonales. Les bandes passantes de ces sous porteuses sont suffisamment étroites, les distorsions induites par un canal sélectif en fréquence sont alors limitées à une simple atténuation sur chacune d'elles. En outre, la condition d'orthogonalité des sous-porteuses permet leur recouvrement réciproque sans interférence de l'une sur l'autre et accorde donc une haute efficacité spectrale au système. Enfin, les interférences entre sous-porteuses et les interférences entre trames induites par le canal étant fortement limitées, l'OFDM est une modulation particulièrement appréciée pour les transmissions sans-fil à hauts-débits. Elle est utilisée dans la plupart des standards de communication [2].

Le bloc de la transformée de Fourier discrète (DFT) est le bloc le plus important dans les systèmes de communications numériques OFDM. Durant ces dernières années, plusieurs algorithmes ont été élaborés dans le but d'alléger le nombre d'opérations de la DFT. Ils sont connus sous le nom d'algorithmes FFT. Ces derniers rendent possible l'implémentation de la DFT sur une puce électronique. Son exécution 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, de coût d'implémentation et de vitesse de calcul.

L'évolution de la technologie VLSI a facilité l'implémentation des divers algorithmes FFT sur des processeurs dédiés, appelés 'Processeurs FFT' [3]. L'intérêt de la présente étude porte sur l'implémentation au niveau matériel d'un processeur FFT ; un élément essentiel pour la modulation OFDM. La cible matérielle visée est le FPGA (Field Programmable Gate Array)[4]. Les FPGA sont de plus en plus utilisés dans les applications télécommunications. Leur flexibilité, leur rapidité et leurs architectures parallèles et séries les rendent des composants microélectroniques parfaits pour la

modulation OFDM. Afin de réaliser une FFT performante, la conception et la réalisation de sa structure interne efficacement sont les enjeux clés de ce travail de recherche.

Dans la littérature, plusieurs architectures d'implémentation d'FFT sur FPGA ont été proposées afin d'améliorer la vitesse et de réduire l'utilisation de ressource élevée. Il s'avère qu'il existe deux applications principales de la FFT sur FPGA : la conception basée sur les structures parallèles et les architectures pipeline. La FFT parallèles n'utilise qu'un seul papillon et de grandes ressources pour le stockage des données. L'architecture pipeline utilise de nombreux papillons pour améliorer la vitesse.

Les travaux proposés dans [1, 2] examinent le problème particulier des tailles et les exigences supplémentaires de la Digital Radio Mondiale liées à la diffusion radio standard. Dans [3], une conception de pipeline basée sur FPGA de processeur FFT inverse utilisé dans la diffusion de télévision numérique a été présentée. Dans cette tentative, l'attention est focalisée, de manière efficace, sur la tâche d'implémentation matérielle du cœur DFT limité aux tailles premières [4, 5]. Par conséquent, l'algorithme devient moins régulier et son implémentation plus compliquée. La plupart des travaux existants sur l'implémentation matérielle des algorithmes DFT avec une taille non puissance de deux se concentrent sur des implémentations pour répondre à un coût d'application spécifique et à des objectifs de performance [6].

De plus, l'algorithme DFT peut être exploité pour effectuer une opération de convolution, ce qui est très utile pour les principales tâches de traitement du signal. En effet, il existe un intérêt considérable dans les systèmes embarqués pour développer un nouveau processeur DFT qui doit être cohérent et conforme à ce paradigme : performances métriques optimales en termes de latence, débit, zone occupée et consommation électrique [7]. De plus, il doit fournir des architectures adéquates, des algorithmes de codage rapides, un protocole de communication fiable et efficace, une qualité de service attrayante ainsi que le contrôle de flux adaptatif et la congestion [8].

Dans ce contexte, l'algorithme CZT nécessite  $N$  convolutions linéaires et  $2N$  multiplications complexes pour une DFT de longueur  $N$ . Ainsi, la capacité exceptionnelle de cet algorithme le prédispose à calculer la DFT pour n'importe quelle taille de transformée. Cependant, l'émergence de part et d'autre de deux multiplicateurs complexes peut considérablement altérer et affecter les performances [10].

L'algorithme de Rader souffre également de certaines limitations telles que seule la transformation de longueur première peut être considérée et les données de flux d'entrée et de sortie apparaissent dans un ordre permuté [11]. Par conséquent, l'utilisation de cette méthode est d'une importance cruciale, car elle est ostensiblement un prélude pour prédire des améliorations de sa structure afin d'être bien adaptée à toutes les circonstances et d'être un filtre à réponse impulsionnelle finie (FIR) approprié [12]. Avec l'avènement du schéma de conception du multiplicateur basé sur l'architecture d'adder graphe réduit (RAG), le coût et la complexité de l'algorithme de Rader peuvent être considérablement réduits, conduisant à une implémentation prometteuse et avantageuse [13].

## **0.2- Objectifs et Principales contributions de la thèse.**

Notre objectif est la conception et l'implémentation d'une technique de communication numérique OFDM sur une cible FPGA. Étant donné que le point clé (sensible) de la modulation OFDM est la transformée de Fourier rapide, cette thèse est intéressée par l'implémentation d'un processeur FFT sur une cible FPGA.

Cependant, l'implémentation temps réel d'un processeur spécifique en se basant sur la FFT conduit à un problème de complexité algorithmique qui va se répercuter sévèrement sur les performances métriques telles que :

- ❖ L'augmentation du nombre de coefficients (espace mémoire important)
- ❖ Consommation considérable en termes de ressources internes de la cible FPGA
- ❖ Un temps de traitement plus long

Divers algorithmes et architectures FFT ont été proposés pour concevoir un processeur FFT efficace. Afin de répondre aux exigences des applications, principalement deux types de FFT, dont la dimension de la séquence puissance de 2 et de la séquence n'est pas une puissance de 2. Certes, la conception de deux processeurs FFT est plus difficile car la gestion et le traitement des données ne sont pas réguliers.

Beaucoup des travaux ont été fait pour résoudre le problème de la complexité imposée par le choix de l'algorithme. Certains auteurs préconisent l'algorithme de décomposition à base 2,4 et 8.... Ainsi une autre alternative qui se base sur la séparation qualifie Split Radix, qui utilise un mélange des bases, a d'autre coté une démarche faisant intervenir l'algorithme de Chirp-z qui s'appuie sur le principe de la convolution. La conception du processeur FFT

pour une taille à puissance de 2 est un sujet relativement simple et bien connu. Cependant, la conception des architectures FFT ayant des tailles de nombre premier pose encore quelques défis.

En fait, notre choix porte sur l'algorithme de Rader qui se présente sous la forme d'un filtre FIR, ce qui permet de profiter amplement de la propriété de la convolution et la nature des nombres premiers de points candidats.

Dans tout dispositif de traitement de signal numérique, la pièce maîtresse est l'architecture de son multiplicateur. Il s'ensuit que la vitesse de ces dispositifs est intimement liée à la vitesse de leurs multiplicateurs. Donc il est important de choisir un multiplicateur approprié en évitant de recourir à une architecture complexe et coûteuse en termes de ressources matérielles. Pour cette raison, la mise en œuvre de l'algorithme de Rader consiste à exprimer la structure transposée du filtre FIR en utilisant le multiplicateur de graphe d'addition réduit (RAG) avec une quantification des coefficients en virgule fixe. Cette démarche nous semble prometteuse et elle va booster ostensiblement les performances métriques.

### **0.3 Organisation de la thèse**

Ce manuscrit s'articule autour de quatre chapitres :

**Chapitre 1 : Généralités sur la modulation OFDM.** Ce chapitre présente l'histoire et développement de l'OFDM ainsi que le principe de fonctionnement et les différents éléments de la chaîne d'émission et réception OFDM et finalement les avantages et les inconvénients de ce système.

**Chapitre 2 : Étude théorique des algorithmes FFT .** Ce chapitre présente les différents algorithmes de la FFT de la littérature ainsi que les différents types de multiplicateurs utilisés dans l'implémentation matérielle et le rôle primordial de ce dernier pour faciliter l'implémentation.

**Chapitre 3 : Simulation de la modulation OFDM.** Ce chapitre est divisé en deux parties. La première présente d'abord ces circuits programmables et leurs caractéristiques ainsi que les outils d'aide à la conception de circuits sur FPGA. Dans la seconde partie, une simulation de la modulation OFDM est présentée en détails.

**Chapitre 4: Méthodologies d'implémentation d'FFT et discussion des résultats.** Le quatrième chapitre est consacré aux méthodes proposées pour l'implémentation de bloc FFT

qui est le noyau de la modulation multi-porteuses OFDM. Ce chapitre est constitué de deux parties : la première partie présente l'implémentation d'FFT à base de Radix 2 d'un côté et à base d'algorithme CORDIC a d'autre coté. La seconde partie de chapitre présente la contribution effective de cette thèse, ont été donnés les détails sur la solution proposée, analyse et discute les différents schémas de calcul et propose une évaluation des performances en comparant les résultats obtenus avec 3 études publiées.

# Chapitre 1 :

## Généralités sur la modulation OFDM.

1.1 Introduction . . . . .	7
1.2 Histoire et développement de l'OFDM.....	7
1.3 Structure du système de communication numérique.....	9
1.4 Contexte de modulation OFDM. ....	9
1.5 Notion d'orthogonalité.....	12
1.6 Implantation numérique du modulateur/démodulateur . . . . .	13
• 1.6.1 Principes de la modulation.....	15
1.7 Caractéristiques du signal OFDM.....	17
• 1.7.1 Intervalle de garde . . . . .	17
• 1.7.2 Interférence entre symboles . . . . .	18
• 1.7.3 Interférence entre sous porteuses.....	19
• 1.7.4. Synchronisation en OFDM . . . . .	21
1.8. Domaines d'application de l'OFDM . . . . .	21
1.9. Avantages d'OFDM.....	22
1.10 Conclusion.....	23
1.11 Référence de chapitre 1.....	24

## **1.1 Introduction:**

Les différentes techniques de modulation sur les canaux à trajets multiples sont devenues l'outil incontournable pour concevoir et construire les futurs systèmes universels de communications mobiles. Ces dernières années, l'intérêt pour la modulation OFDM, une solution parfaite pour les canaux à trajets multiples a été rétabli. Ce premier chapitre présente l'état de l'art de l'OFDM. Nous commençons avec une vue d'ensemble historique des communications multi-porteuses et en particulier l'OFDM. Ensuite, nous expliquons en détails le principe de fonctionnement de l'OFDM. Enfin, nous mettons en évidence les avantages, les limitations de cette méthode et ses domaines d'applications.

## **1.2-Histoire et développement de l'OFDM**

Au cours des dernières années, le domaine des télécommunications numériques a connu une croissance spectaculaire à tous égards [5]. L'OFDM est l'une des modulations multi-porteuses les plus utilisées dans ce domaine. L'origine de la transmission multi-porteuses a débuté avec la technique de multiplexage fréquentielle FDM (Frequency Division Multiplexing), qui a été proposé pour la première fois à la fin des années 50 par [6]. Dans le multiplexage par division de fréquence (FDM), la gamme de fréquence utilisable est divisée en plusieurs petites plages appelées canaux, dont chacun est caractérisé par une fréquence porteuse différente et destiné en permanence à un utilisateur. Le mécanisme FDM utilise une bande de garde entre les canaux afin d'éviter les interférences entre les porteuses (ICI), mais ce processus présente un gaspillage du spectre. Quelques années plus tard la FDM a été améliorée vers une transmission multi-porteuse OFDM (Orthogonal Frequency Division Multiplexing) par [7], permettant à plusieurs fréquences porteuses, de se chevaucher les unes avec les autres afin de maximiser l'utilisation du spectre de fréquence. Ceci est obtenu en présentant une orthogonalité entre les sous porteuses de telle sorte à ne pas avoir des interférences entre celles-ci (ICI). La figure 1.1 illustre une comparaison entre le spectre de fréquence d'un système FDM et OFDM [8 ,9].

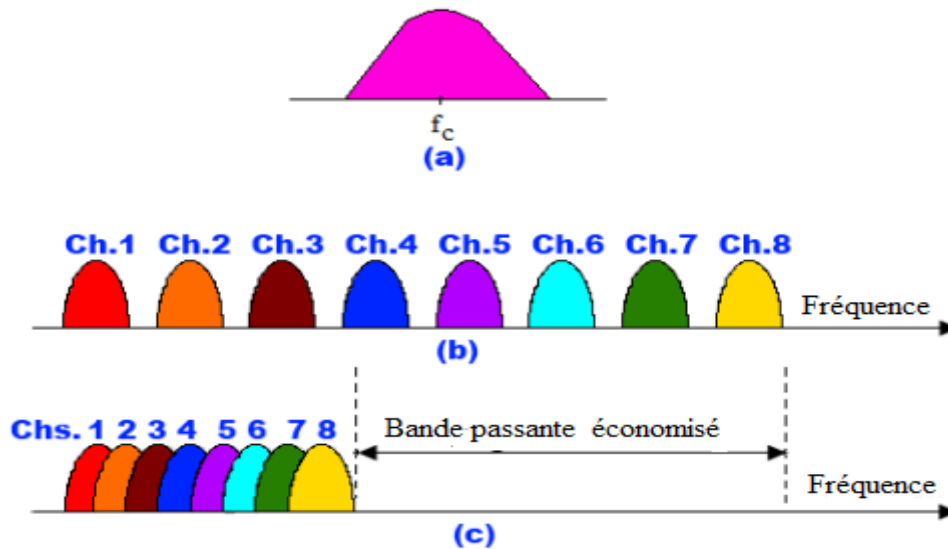


Figure1.1: Comparaison entre le spectre de fréquence d'un system (b) FDM et (c)OFDM

L'OFDM nécessite une complexité d'implémentation élevée pour générer un modulateur et un démodulateur à base des sous-porteuses sinusoïdales, cette obligation limite l'effet de l'OFDM jusqu'à 1971 lorsque [10] ont inspiré la transformée de Fourier discrète pour l'implémentation numérique de la modulation et la démodulation OFDM. Par la suite, d'autres chercheurs s'intéressèrent de plus en plus sur le développement de l'OFDM, au début des années quatre-vingt que [11] ont proposé la technique multi-porteuses avec l'insertion du préfixe cyclique. Dans un dispositif OFDM, les sous porteuses peuvent être utilisés de différentes façons comme le montre la figure 1.2. La majorité des sous porteuses sont utilisées dans la transmission des données, certains sous porteuses sont utilisées comme des pilotes qui servent à l'estimation du canal et à la synchronisation, et d'autres sont utilisées au début (préfixe) ou à la fin (suffixe) de chaque symbole OFDM comme un intervalle de garde [12].

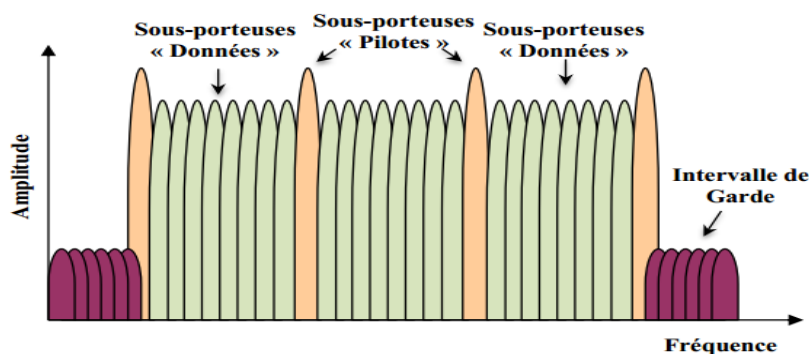


Figure1. 2: Différentes sous -porteuses d'un symbole OFDM [5]

### 1.3 Structure du système de communication numérique

Un système de communication numérique implique la transmission de la forme numérique d'un point à un autre, comme illustré à la figure 1.3.

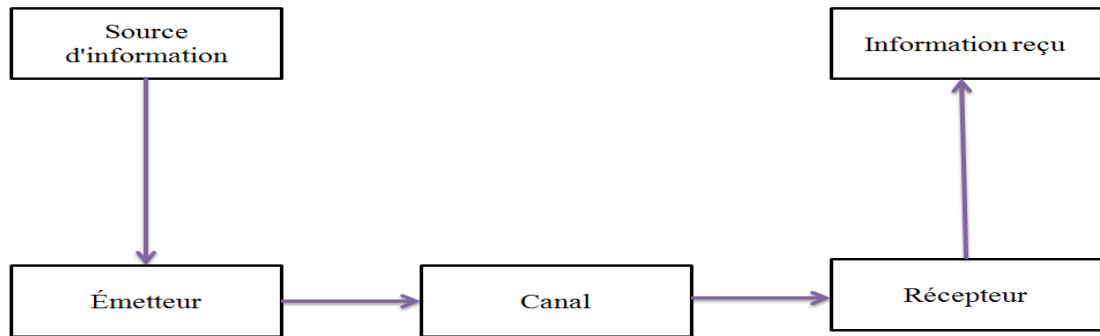


Figure 1.3 systèmes de communication numérique

Quelle que soit la forme de la méthode de communication, les trois éléments de base d'un système de communication sont l'émetteur, le canal et le récepteur.

La source d'information est constituée des messages à transmettre à l'autre bout du récepteur. Un émetteur peut comprendre un codeur source, un codeur de canal et une modulation. Le codeur source utilise une présentation efficace de l'information permettant de conserver les ressources. Un codeur de canal peut inclure une détection d'erreur et un code de correction. L'objectif est d'augmenter la redondance des données pour améliorer la fiabilité de la transmission. Un processus de modulation convertit le signal de la bande de base en un signal passe-bande avant la transmission [13]. Pendant la transmission, le signal subit une dégradation qui atténue l'amplitude du signal et déforme la phase du signal. De plus, les signaux transmis via un canal sont également perturbés par le bruit, supposé être une composante gaussienne distribuée. Au niveau du récepteur, l'ordre inverse des étapes de l'émetteur est exécuté. Idéalement, les mêmes informations doivent être décodées chez le destinataire.

### 1.4 Contexte de modulation OFDM

Parmi les différentes opérations réalisées dans une chaîne de transmission OFDM, on peut citer :

- **Le codage de source:** le rôle consiste à réduire la taille du message avec ou sans perte d'information afin de minimiser l'occupation de bande du signal et permettre d'envoyer plus de données.

- **Le codage de canal:** l'objectif du codage de canal est de protéger le message contre les perturbations du canal de transmission. En effet, la mobilité dans les communications sans fils est une source de nombreux problèmes qui peuvent nuire à la bonne réception des données. Par conséquent, en ajoutant en émission de la redondance au message, il se forme des bits ajoutés (codage canal) ceci afin de permettre au récepteur de détecter et/ou de corriger les erreurs induites par le canal de transmission (décodage canal). Le code convolutif est le plus utilisé dans les communications modernes, il se base sur des opérations récursives simples qui génèrent de l'information en fonction des données codées. Pour faire le décodage du code convolutif, on utilise l'algorithme de Viterbi, qui vise à minimiser l'incertitude des données reçues. Par rapport au codage, le décodage se réalise avec une forte complexité, tant en puissance de calcul qu'en mémoire.

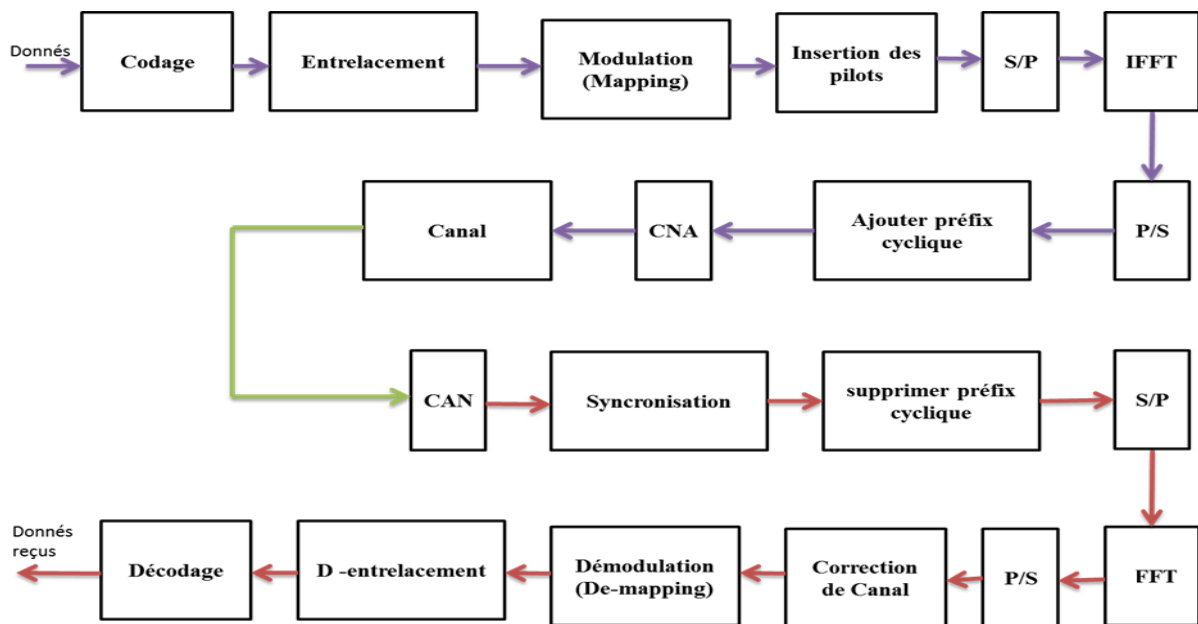


Figure 1.4 : Schéma d'un système OFDM

- **la modulation:** la modulation effectue une conversion de bits en symboles. Pour une modulation M-aire, on associe à chaque mot de n bits un signal  $x_i(t)$ ,  $i = 1, 2, \dots, M$  de durée T choisi parmi les  $M = 2^n$  signaux. Par conséquent, la modulation permet d'augmenter le débit par la transmission d'une plus grande quantité d'informations sur une bande passante définie.

La figure 1.4 présente un système détaillé de communication entre émetteur et récepteur OFDM. Dans ce travail, l'axe principal est la partie FFT et IFFT du système OFDM.

**Les symboles OFDM :** Les symboles d'entrée sont entrés dans l'émetteur en série à  $R$  symboles / seconde. Ces symboles passent par un convertisseur série-parallèle et transmettent des données sur  $M$  lignes en parallèle. Le débit de données sur chaque ligne  $M$  est  $R / M$  symboles / seconde.

Un symbole dans ce flux de données parallèle est désigné par  $X_{i,k}$ . L'indice  $i$  fait référence au sous-canal auquel appartient le symbole, et  $i$  est compris entre 1 et  $M$ . L'ensemble des sous-symboles de  $X_{1,k}$  à  $X_{M,k}$  constitue un symbole OFDM.

Les symboles  $M$  sont envoyés à un bloc IFFT (Inverse Fast Fourier Transform) qui effectue une opération IFFT à  $N$  points. L'IFFT transforme un spectre (d'amplitude et phase de chaque composant) en un signal du domaine temporel.

Afin de préserver l'orthogonalité des sous-porteuses et l'indépendance des symboles OFDM ultérieurs, un intervalle de garde cyclique est introduit.

Le modulateur et le démodulateur utilisent les transformées de Fourier discrètes [15]. Pour discrétiser, il faut choisir une fréquence d'échantillonnage. À l'autre côté, la démodulation impose cette fréquence et le signal occupe la bande passante  $B$  autour de la fréquence d'échantillonnage  $f_0$  comme le montre le schéma suivant.

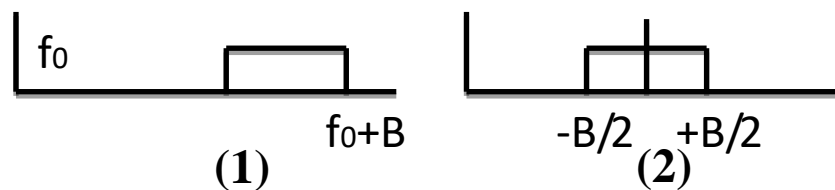


Figure 1.5 : la bande passante occupée

À la démodulation, on va d'abord transformer le signal en bande de base, donc faire une translation de  $f_0 + B/2$ , fréquence médiane de la bande. Le spectre occupera la bande  $[-B/2, B/2]$ , voir le schéma (2).

Donc la bande passante du signal étant  $B/2 = N/2T_s$ , la fréquence d'échantillonnage doit être supérieure ou égale à  $2B/2$  soit  $N/T_s$ . L'échantillonnage aux temps  $t_n = nT_s/N$ .

## 1.5 Notion d'orthogonalité

Les signaux sont orthogonaux s'ils sont mutuellement indépendants les uns des autres. L'orthogonalité est une propriété qui permet à plusieurs signaux d'information d'être parfaitement transmis sur un canal commun et détectés, sans interférence. La perte d'orthogonalité entraîne un flou entre ces signaux d'information et une dégradation des communications. De nombreux schémas de multiplexage courants sont intrinsèquement orthogonaux. Le multiplexage par répartition dans le temps (TDM) permet la transmission de plusieurs signaux d'information sur un seul canal en attribuant des tranches de temps uniques à chaque signal d'information distinct. Pendant chaque tranche de temps, seul le signal d'une seule source est transmis, évitant toute interférence entre les multiples sources d'informations. En raison de ce TDM est de nature orthogonale. Dans le domaine fréquentiel, la plupart des systèmes FDM sont orthogonaux car chacun des signaux de transmission séparés est bien espacé en fréquence empêchant les interférences. Bien que ces méthodes soient orthogonales, le terme OFDM a été réservé à une forme particulière de FDM. Les sous-porteuses dans un signal OFDM sont espacées aussi près que cela est théoriquement possible tout en conservant une orthogonalité entre elles [16]. OFDM réalise une orthogonalité dans le domaine fréquentiel en attribuant chacun des signaux d'information séparés à différentes sous-porteuses. Les signaux OFDM sont constitués d'une somme de sinusoides, chacun correspondant à une sous-porteuse. La fréquence de bande de base de chaque sous-porteuse est choisie pour être un multiple entier de l'inverse du temps de symbole, ce qui fait que toutes les sous-porteuses ont un nombre entier de cycles par symbole. En conséquence, les sous-porteuses sont orthogonales les unes par rapport aux autres[17]. Le principe d'orthogonalité est présenté dans la figure 1.6.

Les fréquences des sous-porteuses sont définies par :

$$f_k = f_0 + k\Delta f \quad \text{avec} \quad k = 0, \dots, (N-1) \quad (1.1)$$

Où  $f_0$  est la fréquence porteuse origine,  $\Delta f$  est l'écart entre deux sous-porteuses consécutives et  $N$  est le nombre des sous-porteuses. L'espacement minimal entre deux sous porteuses adjacentes permettant de conserver l'orthogonalité entre elles est de [18]:

$$\Delta f = 1 / T_u \quad (1.2)$$

Avec,  $T_u$  est la période utile du symbole

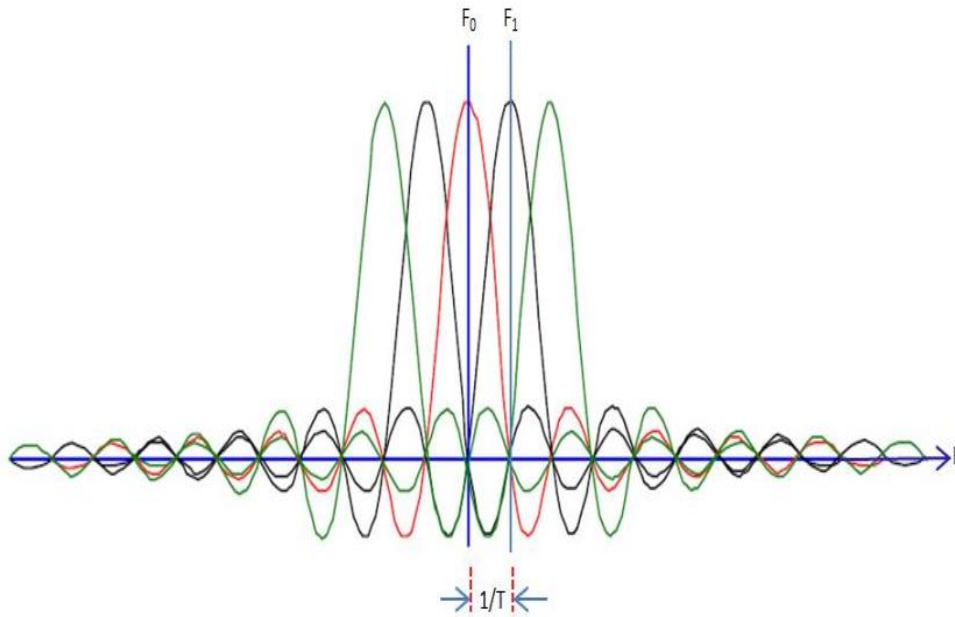


Figure1. 6 : Spectre d'un signal OFDM [16]

### 1.6 Implantation numérique du modulateur/démodulateur

La technique de transmission multi-porteuses OFDM consiste à partager le spectre de fréquence en  $N$  sous porteuses, ensuite à transmettre parallèlement et simultanément les flux de symboles sur les différentes  $N$  sous-porteuses modulées à un débit  $R_{mc} = 1/TN = RN$ , avec  $R$  est le débit binaire d'une transmission mono-porteuse. la durée d'un symbole est étendue à un facteur de  $N$ , donc le système multi-porteuses est moins sensible à la dispersion temporelle du canal par rapport à un système mono-porteuses transmettant des données au même débit [19, 20].

En générale, le signal OFDM est composé de  $N$  sous-porteuses de fréquence  $f_k = f_0 + k\Delta f$ , utilisées pour transmettre en parallèle les  $N$  symboles complexes  $X_k$  définis par une modulation PSK ou QAM. La représentation mathématique d'un signal modulé en OFDM est exprimée par [21, 12]:

$$x(t) = \sum_{k=0}^{N-1} X_k e^{j2\pi f_k t} \quad (1.3)$$

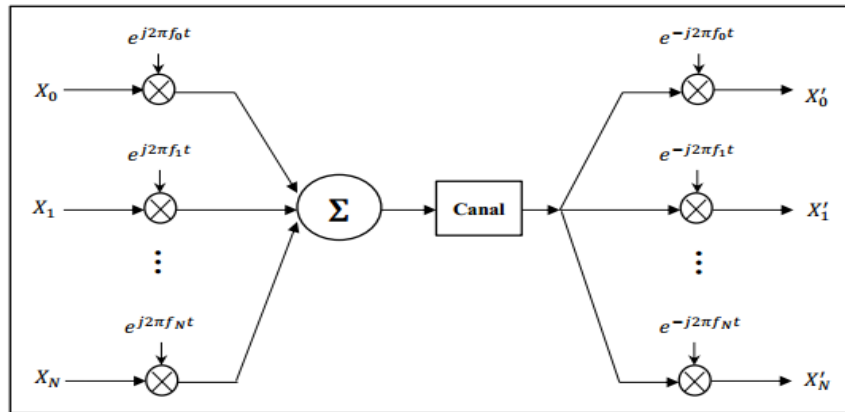


Figure 1.7 : Schéma de principe d'un modulateur OFDM

Les systèmes de transmission modernes mettent généralement en œuvre des émetteurs et des récepteurs numériques, un émetteur numérique produit une sortie sous forme de données échantillonnées. Si  $t = nT_s$ , ou  $T_s$  est la période d'un échantillonnage. La sortie numérique de l'émetteur multi-porteuse est donnée par :

$$x(nT_s) = \sum_{k=0}^{N-1} X_k e^{j2\pi f_k nT_s} \quad (1.4)$$

En suite si les fréquences porteuses sont uniformément séparées dans le domaine fréquentiel par un espacement de  $f_k = k\Delta f$ , donc :

$$x(nT_s) = \sum_{k=0}^{N-1} X_k e^{j2\pi \Delta f nT_s} \quad (1.5)$$

Avec  $\Delta f = 1/NT_s$  est l'espacement minimal pour maintenir une orthogonalité entre les signaux, le signal OFDM est donné sous forme [21]:

$$x(nT_s) = \sum_{k=0}^{N-1} X_k e^{j2\pi nk/N} \quad (1.6)$$

Le résultat obtenu ci-dessus met en certitude que le signal OFDM peut être aisément généré en employant la transformée de Fourier discrète inverse (IDFT). A la réception, les

symboles émis peuvent être recouverts en appliquant la transformée de Fourier discrète directe (DFT) aux échantillons reçus. Néanmoins la modulation et la démodulation OFDM peuvent être implémentées numériquement par un simple calcul de transformée de Fourier rapide inverse (IFFT), et directe (FFT) respectivement [12,22].

### 1.6.1 Principes de la modulation [23]

La première étape de l'émetteur OFDM consiste à fractionner le flux de données en  $K$  sous-flux parallèles, et est modulé sur sa propre sous-porteuse à la fréquence  $f_k$  dans la bande de base complexe. En fait, nous ne modulons pas le flux de données, nous mappons simplement les symboles pour une opération ultérieure. Différents types de modulation numérique sont disponibles pour ce faire, comme par exemple la modulation par décalage d'amplitude (ASK), la modulation par décalage de fréquence (FSK), la modulation par décalage de phase (PSK) et la modulation d'amplitude en quadrature (QAM), etc. Les données sont modulées en fonction de leur taille et sur la base de différents schémas de modulation tels que BPSK, QPSK, 16 QAM et 64 QAM [13]. Le nombre total de bits représentés selon la constellation mappée de différentes techniques de modulation.

- BPSK: 1 bit est transmis par symbole .
- QPSK: 2 bits sont transmis par symbole .
- QAM : 4 ou 16 bits sont transmis par symbole ;

La phase de la porteuse est modifiée en fonction du signal à émettre. Pour la transmission numérique, les 0 et les 1 provoquent une variation de  $180^\circ$  de la phase de la porteuse [24]. A la réception, un détecteur de phase permet de retrouver sur la fréquence modulée le signal numérique émis.

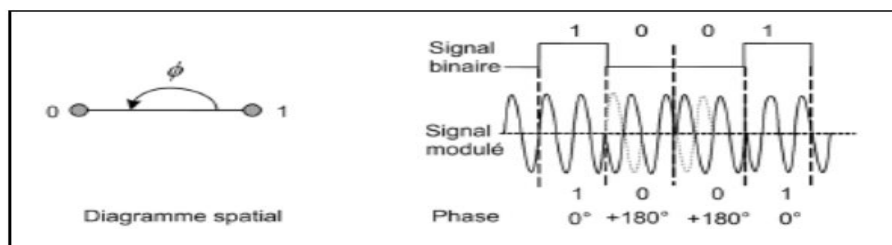


Figure 1.8: le schéma de la modulation BPSK

QPSK : Quadrature Phase

Cette modulation donne deux signaux transportant l'information, l'un est sinusoïdal (Q) et l'autre cosinusoidal (I). Le codage est fait en fonction de ces signaux [25]. Il y a deux phases possibles (séparées de  $180^\circ$ ) pour chacun des deux signaux (I et Q), ce qui permet de réaliser 4 symboles différents (ligne Signal / Data ci-dessous) [26].

La modulation QPSK est très utilisée dans les transmissions satellites car elle permet d'obtenir un signal peu sensible aux interférences tout en conservant un débit acceptable.

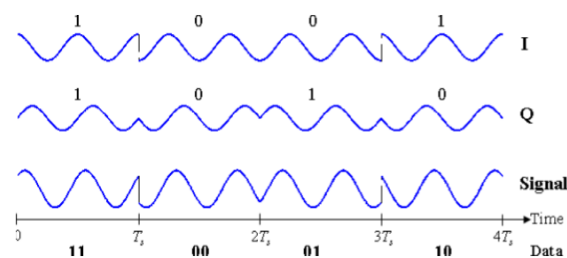


Figure1. 9: le schéma de la modulation QPSK

#### QAM : Quadrature Amplitude Modulation

Cette modulation complète la modulation QPSK. En effet, elle utilise le même système de modulation de phase mais elle ajoute, en plus, une modulation d'amplitude. Les signaux I et Q peuvent prendre différentes valeurs d'amplitude et ainsi augmenter le nombre de symboles transmissibles.

Le bloc suivant de la figure 1.10 dans l'émetteur OFDM est la partie IFFT qui est la partie principale de l'émetteur OFDM. Dans le transmetteur OFDM , IFFT (Inverse Fast Fourier Transform) utilisé pour créer une forme d'onde OFDM ou un symbole à l'aide de flux de données modulés. IFFT convertit les flux de données du domaine fréquentiel en domaine temporel sous forme discrète. Puis convertissez à nouveau le signal parallèle en série.

Avant de transmettre le signal OFDM dans le support sans fil, le signal doit être converti en signal analogique et le faire passer à travers le filtre passe-bas. Ensuite, les signaux OFDM sont transmis par l'antenne au canal multi trajet .

D'autre part, dans le récepteur OFDM, la FFT (Transformée de Fourier Rapide) utilisée pour démoduler les flux de données en tant que domaine temporel en domaine de fréquence

## 1.7 Caractéristiques du signal OFDM

### 1.7.1 Intervalle de garde [27]

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. 1.10 (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. 1.10 (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.

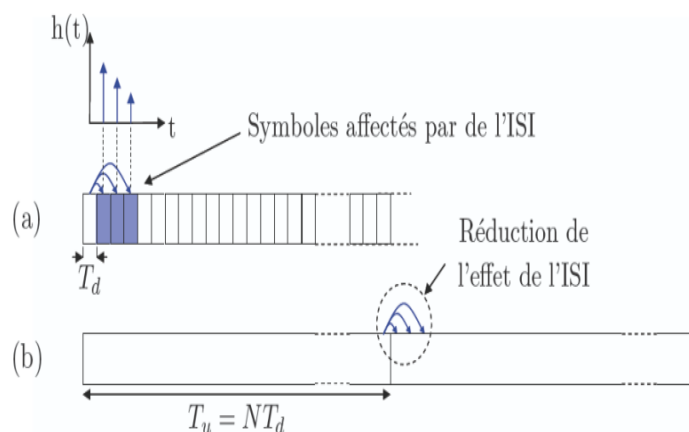


Figure 1.10: Effets du canal à trajets multiples sur des symboles reçus dans le cas

(a) mono-porteuse (b) multi-porteuses [27]

Avec :

$T_u$ : 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.

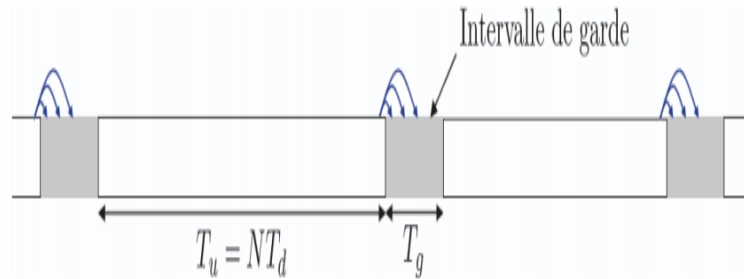


Figure 1.11: Illustration de l'effet de l'intervalle de garde précédant chaque OFDM

Avec:

$T_g$ : La durée de l'intervalle de garde.

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

### 1.7.2 Interférence entre symboles (ISI)

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 [28], 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 1.11. 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  $T_s = T_u + T_g$  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 [28] et la seconde, appelée (Zero Padding : ZP-OFDM) consiste quant à elle à insérer des échantillons de valeur nulle entre les symboles OFDM [29]. 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.

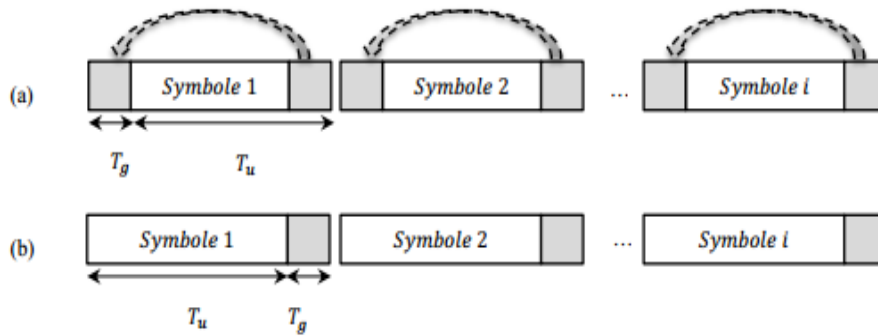
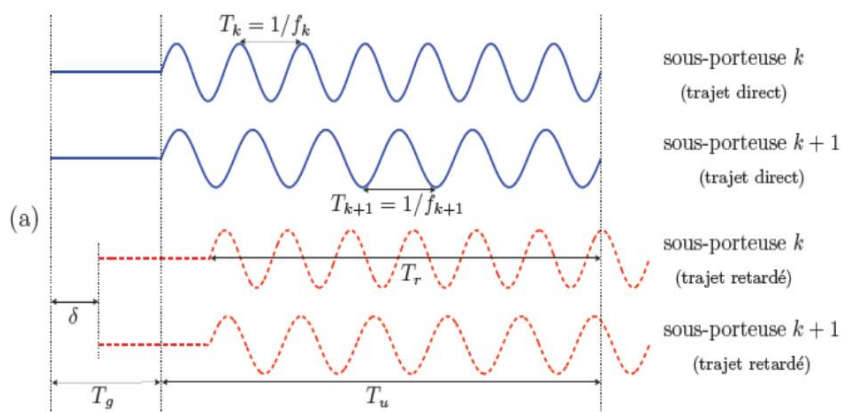


Figure 1.12 : présentation d'intervalle de garde de type (a) préfixe cyclique et (b) zero padding dans un symbole OFDM.

### 1.7.3 Interférence entre sous porteuses (ICI)

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  $\delta$ , 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 1.13 les chronogrammes de deux "voies" particulières, correspondant aux sous-porteuses de fréquences respectives  $f_k$  et  $f_{k+1}$ .



(a)

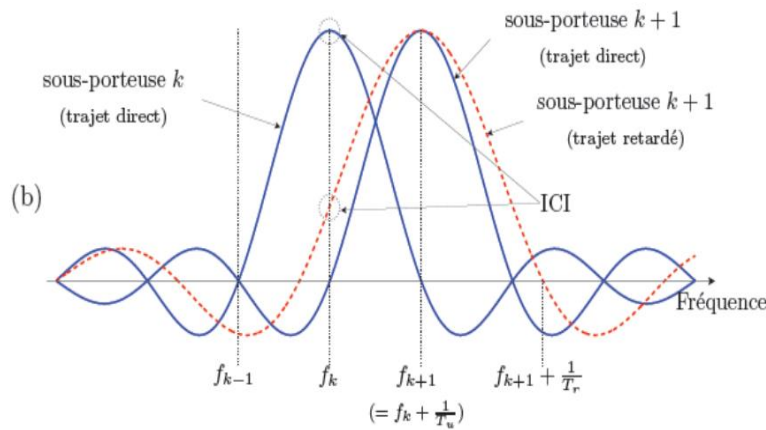


Figure 1.13 : Interférence inter-porteuse (ICI) en OFDM dans les domaines :(a)temporel et (b) fréquentiel dans le cas d'un canal à deux trajets [28 29].

Dans la figure 1.13(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  $\delta$ , 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  $Tr$  [28, 29].

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 1.13 (a).

Pour le trajet direct, la transformée de Fourier d'une sinusoïde de fréquence  $fk$ , convolutive par la fonction porte de largeur  $T_u$ , correspondra à un sinus cardinal s'annulant aux fréquences  $fk \pm 1/ T u$ , comme cité au paravent. Il en est de même pour la sous-porteuse  $fk + 1$ .

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  $Tr < T_u$ . 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 1.13 (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  $T_u$  [29].

Dans le domaine fréquentiel et grâce au préfixe cyclique, la sommation des signaux de la sous-porteuse  $f_k$  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.

#### **1.7.4. Synchronisation en OFDM**

La synchronisation en temps et en fréquence entre l'émetteur et le récepteur a un impact crucial sur la performance d'une liaison OFDM. Il existe essentiellement deux types d'erreurs de synchronisation. A savoir l'erreur de synchronisation temporelle qui est due au décalage temporel entre le signal reçu à l'entrée du récepteur et l'horloge de réception, cette erreur provoque éventuellement une rotation de la phase des porteuses et des interférences entre les symboles (ISI) conduisant à une rupture de l'orthogonalité entre les porteuses. De plus l'erreur de synchronisation fréquentielle se produit par un décalage en fréquence qui vient du fait que la fréquence du signal reçu est différente de celle de l'oscillateur local de réception ou bien par la présence de la fréquence doppler. Il en résulte notamment une atténuation et un déphasage du signal de plus une perte d'orthogonalité introduisant ainsi des interférences entre les porteuses (ICI) [20]. Une variété d'algorithmes de synchronisation a été proposée dans la littérature pour corriger à la fois les erreurs de synchronisation en temps et en fréquence tels que [30 31] :

- Algorithmes de synchronisation à base de symbole pilotes connus par l'incorporation des pilotes à l'intérieur du symbole OFDM.
- Algorithmes exploitant l'intervalle de garde.

#### **1.8. Domaines d'application de l'OFDM:**

L'OFDM est restée une technique prépondérante, puisqu'elle est utilisée pour de nombreuses applications [32]. L'OFDM offre des possibilités intéressantes de dépasser les capacités de système CDMA et de fournir la méthode d'accès sans fil pour les systèmes 4G [33]. En plus dans les normes de diffusion numérique du audio/vidéo DAB

(Digital Audio Broadcasting), DVB ,de télévision numérique terrestre DVB-T (Digital Vidéo Broadcasting terrestrial) de communications numériques hauts débits ADSL (Asynchronous Digital Subscriber Line) sur la boucle locale téléphonique et ses dérivés ([AHV]-DSL), ainsi que dans l'étude des normes de communications pour réseaux locaux à l'intérieur des bâtiments de type BRAN (Broadband Radio Access Network), qui est prévu pour des débits allant jusqu'à 54 Mbps. Grâce à sa fiabilité OFDM sera adoptée pour l'ATM sans fils. habituellement, l'OFDM est une technologie fiable pour la transmission de données ultrarapide et donc, peut être utilisée pour des réseaux à fréquence unique avec des grands échos "actifs"[34]. De tels réseaux peuvent être vus comme un arrangement cellulaire d'émetteurs qui émettent le même signal sur la même fréquence très stable et soigneusement synchronisée et avec le même chronométrage de symbole. Les autres applications courantes employant OFDM incluent le GSTN (General Switched Telephone Network), les modems de DSL et d'ADSL et les standards de réseau sans fil IEEE.

### **1.9. Avantages de l'OFDM**

- La robustesse des signaux OFDM aux canaux sélectifs en fréquence représente l'avantage principal de cette modulation [27] . 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.
- 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.
- 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.
- Une interprétation matricielle des systèmes à reconstruction parfaite permettant de déduire directement des structures de réalisation génériques et régulières, de travailler en temps

discret, ce qui permet l'application directe des filtres ainsi construits dans les modulateurs/démodulateurs, notamment sous forme numérique .

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

Malgré ces avantages, l'OFDM présente quelques points faibles.

### **1.10 Conclusion :**

Dans ce chapitre, ont été présenté un historique et développement de la modulation multi porteuse orthogonale fréquentiel OFDM. Ensuite on a présenté le principe de fonctionnement et les différents éléments de la chaine d'émission et réception OFDM. Ainsi que Les paramètres les plus importants à prendre en compte dans toute conception OFDM sont le temps de garde, la durée du symbole, le nombre de sous-porteuses, l'espacement des sous-porteuses, le type de modulation et finalement les avantages et les inconvénients de la modulation OFDM.

### **1.11. Références du chapitre 1**

- [ 1] HARIZE Saliha, Etude et implémentation d'un codeur vidéo sur un système embarqué de type FPGA , **these** de doctorat,univ Annaba,2013
- [ 2] Nicola Marchetti, Muhammad Imadur Rahman, OFDM: Principles and Challenges, springer,2009.
- [3] D.-S. Kim, S.-S. Lee, J.-Y. Song, et al.: 'Design of a mixed prime factor FFT for portable digital radio mondiale receiver'. IEEE Trans. on Consumer Electronics,2008,pp. 1590–1594
- [4] Rader C.: 'Discrete Fourier Transform when the Number of Data Samples is Prime'. Proceedings of the IEEE, 1968, pp. 1107-1108
- [5 ] Ch I, **Détection multi-utilisateurs MUD dans un système multi-antennes et à modulation multi-porteuses MIMO-OFDM par des approches méta-heuristiques**, université aboubakr belkaïd,2017.
- [6] Doelz. M.L, Heald. E.T, and Martin. D.L, “Binary data transmission techniques for linear systems,” *in Proceeding in IRE*, vol. 45, no. 5, pp. 656- 661, May 1957
- [7] Chang. R.W, “Synthesis of band-limited orthogonal signals for multichannel data transmission,” *Bell Labs Technical Journal*, vol. 45, no. 10,pp. 1775-1796, Dec. 1966.
- [8] Saltzberg. B, “Performance of an efficient parallel data transmission system,” *IEEE Transactions on Communication Technology* , vol. 15, no. 6, pp. 805-811, Dec. 1967.
- [ 9] Chiueh. T.D, and Tsai. P.Y, *OFDM baseband receiver design wireless communications*, John Wiley & Sons (Asia) Pte Ltd, Singapore, 2007
- [10] Weinstein. S, and Ebert. P, “Data transmission by frequency-division multiplexing using the discrete Fourier transform,” *IEEE Transactions on Communication Technology*, vol. 19, no. 5, pp. 628-634, Oct. 1971
- [11] Peled. A, and Ruiz. A, “Frequency domain data transmission using reduced computational complexity algorithms,” *IEEE International Conference on ICASSP'80. Acoustics, Speech, and Signal Processing*, pp. 964- 967, Apr 1980
- [12] Hogue. M.J.V, “Etude comparative des estimateurs de canaux en vue d'une implémentation dans un système de télécommunication sans fil de type MIMO-OFDM,” Thèse de Doctorat, Université du Québec à Trois Rivières. 2009
- [13] Loo Kah C , design of an OFDM transmitter and receiver using FPGA, Universiti Teknologi Malaysia, 2004
- [14] Z H ,layer design et acces multiple dans les reseaux sans fil MIMO:application au reseau wimax ,l'universite de tlemcen,2013

- [15] Modulation multi porteuse » [http://www.rennes.supelec.fr /ren/rd/scee /ftp/docs /%20 ofdm.pdf](http://www.rennes.supelec.fr/ren/rd/scee/ftp/docs/%20ofdm.pdf) , Consulter le 12/02/2015.
- [16] Saeed M, 'Study the carrier frequency offset (cfo) for wireless OFDM', these the Faculty of Engineering and Computer Science University of Denver,2013.
- [17] Guéguen. E, “Etude et optimisation des techniques UWB haut débit multibandes OFDM,” *Thèse de Doctorat*, Institut Nationale des Sciences Appliquées de Rennes. 2009.
- [18]Rohling. H, OFDM concepts for future communication systems, Springer-Heidelberg Dordrecht London New York , 2011.
- [19] Plass. S, Dammann. A, Kaiser. S, and Fazel. K, Multi-carrier spread spectrum 2007, 1st edition Springer Netherlands, 2007.
- [20] Bannour. A, and Matin. A.M, Coding for MIMO-OFDM in future wireless systems, 1st edition Springer Cham Heidelberg New York Dordrecht London, 2015.
- [21] Chiueh. T.D, and Tsai. P.Y, OFDM baseband receiver design wireless communications, John Wiley & Sons (Asia) Pte Ltd, Singapore, 2007.
- [22] Schulze. H, C, Theory and applications of OFDM and CDMA: Wideband wireless communications, Jhon Wiley & Sons, Ltd, England, 2005.
- [23] DAACHI Dalila ,Etude des systèmes mimo-ofdm communication sans fil, Mémoire de master ,université abderrahmane mira de bejaia,2012
- [ 24] Farhi Nabila, etude et simulation d'une transmission de type ofdm pour les communications sans fil, Mémoire de master, universite larbi tebessi - tebessa,2016
- [25] TAHRI,Tarik.' Systèmes radars coopératifs multimodes pour la détection, l'identification des obstacles sur les voies, la localisation et la transmission de données trains infrastructures',université de valenciennes ,2014 .
- [26][https://www.academia.edu/15220754/Objet\\_Diagramme\\_de\\_constellation\\_sur\\_7865\\_7866\\_7865HD\\_7866HD](https://www.academia.edu/15220754/Objet_Diagramme_de_constellation_sur_7865_7866_7865HD_7866HD)
- [27] LARBI Nacéra, Détection multi-utilisateurs dans le contexte de communication mobile , Exploitation de la combinaison OFDM/CDMA, Mémoire de master , Univ ORAN, 2015
- [28] E. Gueguen, Etude et optimisation des techniques UWB haut dé bit multi-bandes OFDM. Thèse de Doctorat, Institut National des Sciences Appliquées de Rennes, Janvier 2009
- [29] 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.

[30] V. Dégardin, Analyse de la faisabilité d'une transmission de données haut débit sur le réseau électrique basse tension. Thèse de Doctorat, Université des Sciences et Technologies de Lille, Décembre 2002.

[31] Jankiraman. M, Space-time codes and MIMO systems, Artech House Inc, Norwood, MA, USA, 2004.

[32] Hanzo. L, Akhtman. Y, and Wang. L, 'MIMO-OFDM for LTE, WiFi and WiMax: Coherent versus non-coherent and cooperative turbo transceivers", Jhon Wiley & Sons Ltd, IEEE Press, United Kingdom, 2011.

[33] GAL YNA PISKONOVA, "Transmission ofdm pour la téléphonie cellulaire", école de technologie supérieure université du Québec, Montréal, 19 décembre 2003.

[34] Eric Phillip, "Adaptive Techniques for Multiuser OFDM", Thèse, Electrical and Computer Engineering School of Engineering James Cook University, Montréal, Décembre 2001.

# Chapitre 2 :

## Étude théorique des algorithmes FFT

2.1 Introduction.....	28
2.2 Transformée de Fourier .....	28
2.3 Transformée de Fourier Discrète .....	29
2.4. La Transformée de Fourier Rapide .....	30
2.5. Algorithmes conventionnels .....	31
• 2.5.1 Radix-2 .....	31
• 2.5.2 Radix-4.....	34
• 2.5.3 Mixed radix .....	34
• 2.5.4 Split-Radix FFT.....	35
2.6. Autre Algorithmes de la transformée de Fourier discrète.....	35
2.7. Multiplicateurs disponibles.....	36
2.7.1 Multiplicateur de tableau .....	36
2.7.2 Multiplicateur de booth.....	38
2.7.3. Le multiplicateur Vedic.....	38
2.7.4. CORDIC .....	39
2.8 Conclusion .....	40
2.9 Référence de chapitre2 .....	41

## **2 . 1 Introduction**

La transformée de Fourier fournit les moyens de transformer un signal défini dans le domaine temporel en un signal défini dans le domaine fréquentiel. Lorsqu'une fonction est évaluée par des procédures numériques, il est toujours nécessaire de l'échantillonner d'une certaine manière. Cela signifie que pour évaluer pleinement une transformée de Fourier avec des opérations numériques, il est nécessaire que les fonctions de temps et de fréquence soient échantillonnées sous une forme ou une autre. Ainsi, la transformée de Fourier numérique ou discrète (DFT) est d'un intérêt primordial. Dans toute application FFT, la taille et le choix du type de multiplieur utilisé pose un problème plus important.

## **2.2 Transformée de Fourier**

La transformée de Fourier est utilisée pour transformer un signal temporel continu dans le domaine fréquentiel [35]. Il décrit le spectre continu d'un signal temporel non périodique. La transformée de Fourier  $X(f)$  d'une fonction temporelle continue  $x(t)$  peut être exprimée comme :

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ftdt}$$
$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ftdf} \quad (2.1)$$

## **2.3 Transformée de Fourier Discrète [36]**

La Transformée de Fourier Discrète (TFD), (*DFT : Discrete Fourier Transform*) est un élément essentiel dans l'analyse, la conception et la mise en œuvre des algorithmes et systèmes du traitement du signal à temps discret. En physique numérique, on dispose presque toujours de signaux issus d'une acquisition électronique ou de résultats de mesures discrètes. Ces signaux et résultats ne sont pas infinis, et généralement sont non périodiques. La TFD a été conçue pour traiter ce genre de données. Elle réalise une décomposition d'un signal tronqué et échantillonné (discret) en une série de Fourier, en le

"périodisant", avec une période égale à  $T$ . La TFD nous permet donc de calculer de façon approchée les coefficients de Fourier des différentes harmoniques d' un signal quelconque et ainsi d' obtenir son spectre en fréquence.

Son application est donc utilisée dans plusieurs domaines technologiques tels que les télécommunications, le biomédical, le traitement sismique, etc. [36]  
La DFT d' un signal discret  $x(n)$  peut-être directement calculée par l' équation (2.2)

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k \in [0, \dots, N-1] \quad (2.2)$$

Avec  $x(n)$  et  $X(k)$ , respectivement, représentent la séquence d' entrée et la séquence de sortie en nombre complexe et  $N$  est la longueur de la transformée. À partir de l' équation (2.2), on constate que la complexité de calcul de la TFD s'exprime en  $O(N^2)$ , il augmente avec le carré de la longueur de la transformée et donc devient cher pour des  $N$  larges. Plusieurs additions et soustractions, en plus de quelques autres opérations, sont nécessaires donc pour réaliser une seule multiplication complexe. C'est pour ces raisons que la TFD n'est pas utilisée pour le traitement du signal temps réel. Puisque le processeur numérique est généralement limité par la taille de sa mémoire. La taille de la TFD  $N$  limite les calculs exécutés le processeur [36].

L' équation de définition de la DFT fournit une relation entre deux ensembles de  $N$  nombres complexes, en posant:

$$W_N^{nk} = e^{-j\left(\frac{2\pi}{N}\right)nk} \text{ avec } j^2 = -1 \quad (2.3)$$

$W_N^p$  Est appelé Facteur de Fourier (*TWF - Twiddle Factors*), plusieurs propriétés caractéristiques de la représentation de ce coefficient sont utilisées [37], par exemple :

$$W_N^{nk} = W_N^{k(n+N)} = W_N^{(k+N)n}$$

$$W_N^{2nk} = W_{N/2}^{nk}$$

$$W_N^{k(N-n)} = (W_N^{nk})^* \quad (2.4)$$

Où  $x^*$  représente le complexe conjugué de la donnée complexe  $x$ . Une forme matricielle montrée par l'équation (2.4) de la DFT peut être donc déduite.

$$x(k) = \begin{bmatrix} W_N^0 W_N^0 & \cdots & W_N^0 \\ \vdots & \ddots & \vdots \\ W_N^0 W_N^{(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ \cdots \\ x(N-1) \end{bmatrix} \quad (2.5)$$

Elle peut être représentée sous la forme matricielle suivante:

$$X(k) = [B_N][X_n] \quad (2.6)$$

Avec  $[B_N]$  la matrice des coefficients et  $[X_n]$  la matrice d'entrée de la TFD. Donc la TFD est une décomposition d'un signal échantillonné, composé de sinusoides. En exploitant les propriétés de symétrie et de périodicité de la TFD, plusieurs méthodes efficaces ont été développées pour calculer la TFD et ainsi diminuer significativement la charge de calcul.

#### 2.4. La Transformée de Fourier Rapide

En introduisant le premier algorithme de la FFT en 1965, Cooley&Tukey ont été capable de réduire d'une façon remarquable le temps de calcul de la TFD d'une suite de nombre d'échantillons  $N$  qui est une puissance de 2, connu sous le nom de radix-2 [38]. Depuis, et encore dans les dernières années, les algorithmes de FFT ont été dans le centre d'intérêt pour de nombreuses recherches qui ont révolutionné le traitement numérique de signal (*DSP : Digital Signal Processing*). La réduction du nombre d'opérations nécessaires, en particulier le nombre des multiplications constituent leurs avantages essentiels et est devenu l'approche de base dans toutes les nouvelles applications. La grande majorité des algorithmes sont basés sur un même principe qui consiste à

décomposer le calcul de la TFD en plusieurs sous-ensembles de TFD de longueur plus petite. En profitant des propriétés de symétrie et de périodicité suivant les équations (2.7) et (2.8) des facteurs de phases:

Symétrie:

$$w_N^{k+N/2} = -w_n^k$$
$$(w_N^k)^* = w_N^{-k} \quad (2.7)$$

Périodicité:

$$w_N^{k+N} = w_N^k \quad (2.8)$$

Où (\*) désigne le complexe conjugué.

Malgré que le nombre des opérations serait réduit d'une façon remarquable, la complexité globale reste  $O(N^2)$  [39] et son flot des données demeure complexe dans l'implémentation. Les algorithmes de FFT permettent de faire une DFT d'une manière efficace ainsi que de réduire la charge de calcul en termes de multiplications et additions à valeurs complexes à l'ordre  $O(N \log_2 N)$ .

## **2.5. Algorithmes conventionnels**

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  $M$  un entier. Grâce à ces algorithmes, il est possible de réduire le nombre d'opérations nécessaires à un ordre de grandeur de  $N \log_2(N) = N \times M$  [37].

### **2.5.1 Radix-2**

Radix-2 ou base-2, signifie 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, elle peut être aussi appelée BPE (*Butterfly Processing Element*). On peut différencier entre deux algorithmes: DIT (*Decimation In Time*, Décimation dans le temps) et DIF (*Decimation In Frequency*, Décimation en fréquence). D'une part, quand la division en Butterfly, commence du côté de l'entrée (signal temporel),

on l'appelle DIT et d'autre part, quand division en Butterfly, se fait du côté de la fréquence, cet algorithme est appelé DIF. Le radix-2 utilise la technique 'diviser pour régner' [39], donc à l'aide de cette technique, il 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.

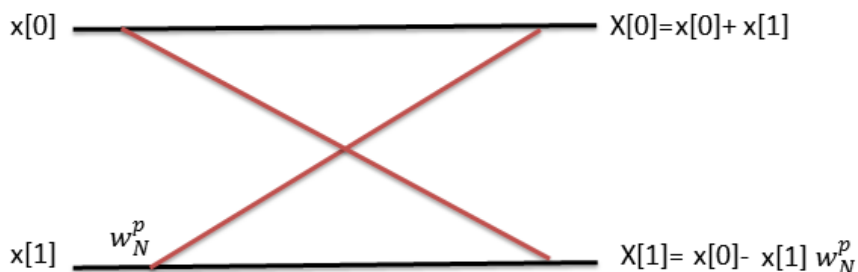


Figure 2.1: Structure du papillon DIT

La figure 2.1 représente le schéma d'une structure papillon DIT de la FFT. On remarque que cette structure, avec  $p = 0$ , représente exactement la relation qui existe entre les échantillons à l'entrée  $x[0]$  et  $x[1]$  et les échantillons en sortie  $X[0]$  et  $X[1]$ , dans une TFD à 2 points. La figure 2.2 quant à elle, représente la structure d'une papillon DIF.

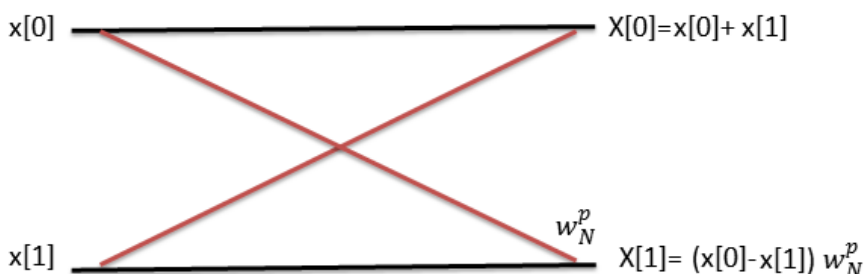


Figure 2.2 : Structure du papillon DIF

Les figures 2.1 et 2.2 représentent les deux structures papillon utilisées respectivement pour le DIT et DIF FFT. Pour le DIT, les facteurs de Fourier sont multipliés par l'entrée  $x[1]$  puis le produit est, ou ajouté ou retranché, à  $x[0]$ . D'une autre part, pour le DIF les entrées  $x[0]$  et  $x[1]$  sont additionnées pour la première sortie puis pour la deuxième sortie,  $x[0]$  est soustraite de  $x[1]$  et on multiplie par le facteur.

Les figures 2.3 et 2.4 représentent les structures des deux algorithmes radix-2 DIT et DIF :

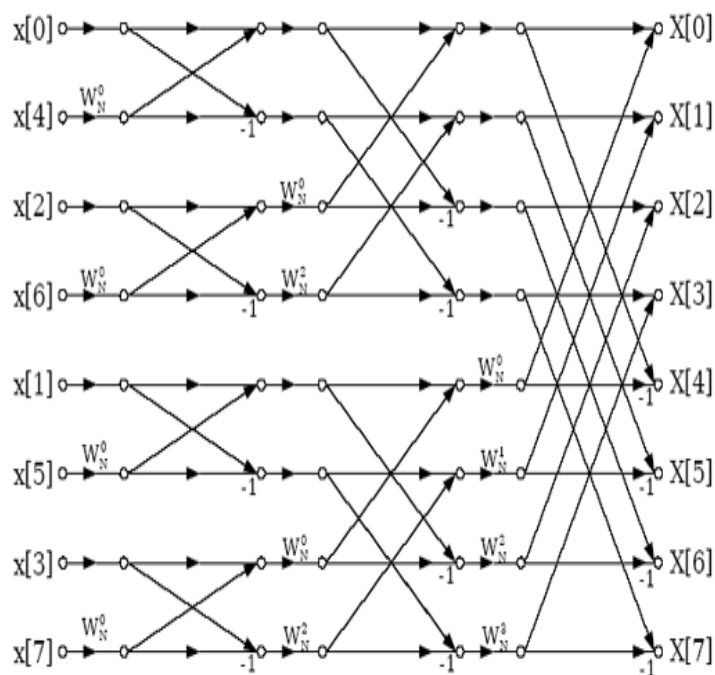


Figure 2.3 : Diagramme de la FFT à base de radix2 de type DIT

Dans les figures 2.3 et 2.4, on représente la structure du radix-2 DIT et DIF pour 8 points.

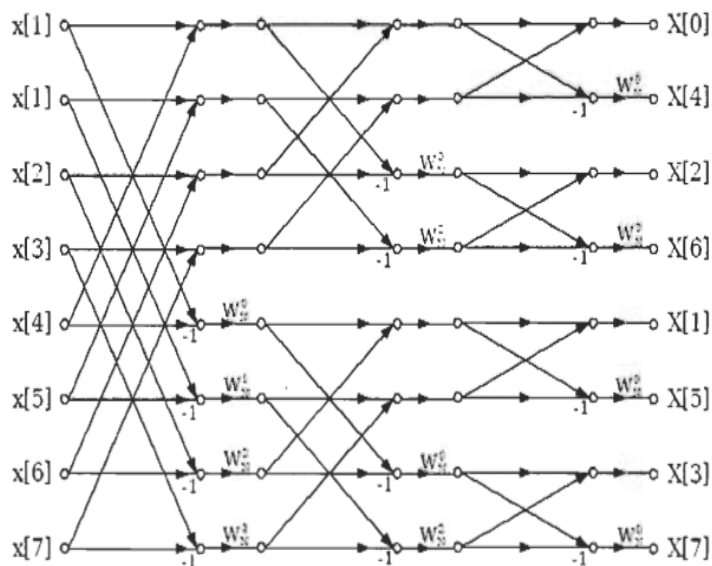


Figure 2.4 : Diagramme de la FFT à base de radix 2 de type DIF

Pour une taille  $N=8$ , pour un radix-2, on a  $\log_2 N$  étages pour calculer la FFT, donc trois étages pour ce cas  $N=8$ . Le radix-2, DIT ou DIF, réduit aussi l'ordre de calcul de  $N^2$  à

$N/2 \log_2 N$  multiplications complexes et de  $N^2 - N$  à  $N \log_2 N$  additions complexes. Grâce à ces avantages, le radix-2 est l'algorithme FFT le plus utilisé actuellement [40], [41]. .

### 2.5.2 Radix-4

Le radix-4 est un algorithme FFT qui peut être utilisé dans le cas où on désire réduire le nombre de multiplications complexes d'environ 25% par rapport au radix-2 [42], [43]. Ceci est vrai dans le cas où la taille de la FFT est une puissance de 4. La structure simplifiée du radix-4 est représenté dans la figure 5.

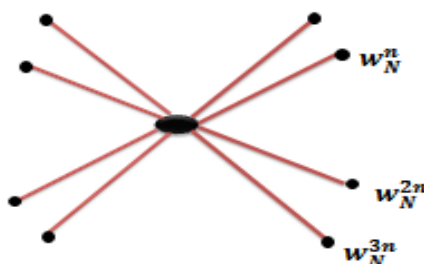


Figure 2.5 : Structure du papillon DIT radix -4

**2.5. Mixed radix** : une FFT peut être partitionnée en plusieurs sous transformées.

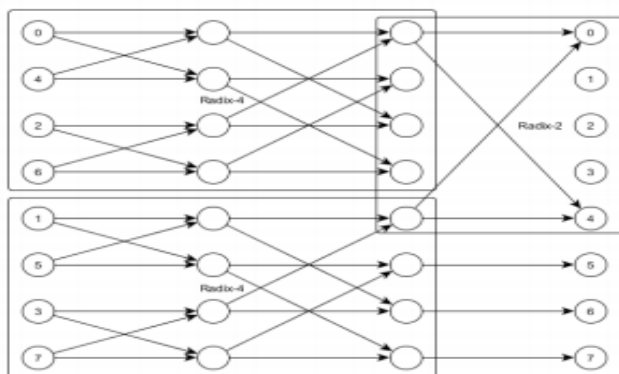


Figure 2.6 : Exemple de FFT 8 points avec l'algorithme mixed-radix [44]

La première passe est calculée avec deux radix-4 et la dernière passe avec 4 radix-2.

Si toutes les partitions ont la même taille  $r$ , alors cette version de l'algorithme FFT est dite de type radix- $r$ . Dans le cas où les partitions ont des tailles différentes, on parle de mixed-radix FFT [44]. L'approche mixed-radix permet plus de degrés de liberté dans l'implémentation et l'organisation des calculs.

## Split-Radix FFT

L'algorithme FFT split-radix trouve son origine dans une observation très simple illustrée par la figure 2.7 : le graphe d'un algorithme radix-2 à entrelacement temporel peut se transformer de manière évidente en graphe d'un algorithme radix-4 uniquement en changeant les exposants de la racine de l'unité servant de coefficients multiplicateurs (twiddle factors) [44].

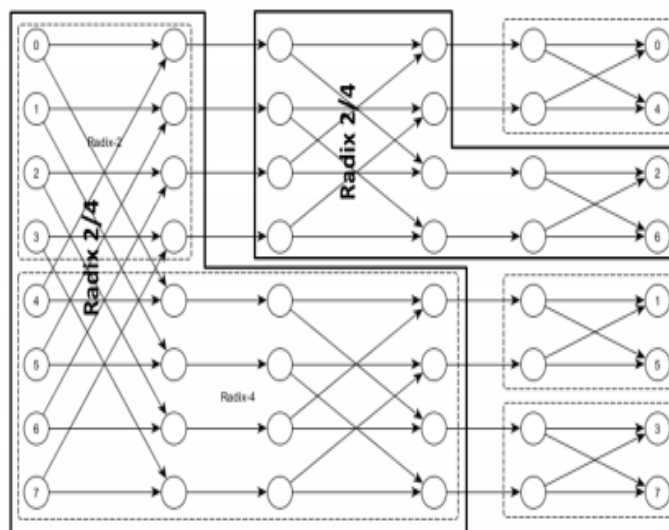


Figure 2.7 – Signal- split-radix FFT de taille 8 [44]

## 2.6. Autre Algorithmes de la transformée de Fourier discrète

**L'algorithme de Bruun** : Est un algorithme de transformée de Fourier rapide (FFT) basé sur une approche inhabituelle de factorisation polynomiale récursive, proposé par G. Bruun pour une puissance de deux en 1978 et par H. Murakami en 1996, généralisé à des tailles composites voire arbitraires [45]. jusqu'à la dernière étape de calcul, il a été initialement proposé comme moyen de calculer efficacement la transformée de Fourier discrète (DFT) de données réelles. Néanmoins, l'algorithme de Bruun illustre un cadre algorithmique alternatif pouvant s'exprimer à la fois avec l'algorithme de Cooley – Tukey et offre ainsi une perspective intéressante sur les FFT qui permet de mélanger les deux algorithmes et d'autres généralisations.

**La transformée Z en chirp**: ( CZT ) est une généralisation de la transformée de Fourier discrète (DFT). Bien que les échantillons DFT à des points uniformément espacés le long du cercle unité, la transformée en chirp Z échantillonne le long d' arcs de spirale dans le plan Z, correspondant à des lignes droites dans le plan S[45]. La DFT réelle et la DFT zoom peuvent être calculées comme des cas spéciaux du CZT.

Il existe d'autres algorithmes qui permettent de calculer la transformée de Fourier discrète. 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 [45]. Le PFA est similaire à celui de Cooley-Tukey.

**L'algorithme de Rader-Brenner** est aussi une variante de Cooley-Tukey avec des *facteurs de rotation* 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. Les algorithmes qui procèdent aussi par des factorisations successives sont celui de Bruun[45] .

**L'algorithme Winograd FFT** tend à réduire le nombre de multiplications au prix d'une augmentation des additions. Pour les grandes tailles de transformées, une application directe de l'algorithme Winograd FFT implique un nombre prohibitif d'additions, en particulier sur les architectures modernes avec des fonctions de multiplication et d'accumulation qui masquent les multiplications à l'intérieur des additions. Cependant, les stratégies hybrides peuvent être efficaces lorsqu'elles s'appuient sur des routines de cœur Winograd de petite taille et lorsque ces cœurs sont imbriqués dans la transformée de Fourier rapide à facteur premier de Good Thomas.

### 2.7. Multiplicateurs disponibles:

**2.7.1 Multiplicateur de tableau (LUT):** Le multiplicateur de tableau est bien connu en raison de sa structure régulière. Le circuit multiplicateur est basé sur des algorithmes d'addition et de décalage. Chaque produit partiel est généré par la multiplication du multiplicande avec un bit multiplicateur. Les produits partiels sont décalés en fonction de leurs commandes de bits. L'addition peut être effectuée avec un additionneur de propagation de report normal. N-1 additionneurs sont requis lorsque N est la longueur du multiplicateur [46].

#### (a) Multiplicateur RCA

Les multiplicateurs RCA fonctionnent en faisant onduler le report vers l'additionneur vers la gauche dans la même rangée. Ainsi, dans une ligne, chaque additionneur doit attendre que l'entrée de retenue effectue son calcul. En d'autres termes, il existe une contrainte intra-itération entre deux nœuds additionnels adjacents au sein d'une ligne, en supposant qu'il n'y a pas de pipelining impliqué. En raison de cette nature à effet d'entraînement, les chemins

critiques impliqués sont assez larges, ce qui limite la vitesse de multiplication. La figure la montre le schéma d'un multiplicateur de bits parallèles à virgule fixe RCA à 4 bits.

**(b) multiplicateur CSA**

Dans le multiplicateur CSA, les sorties de report sont enregistrées et utilisées dans l'additionneur de la ligne suivante. Dans ce cas, le produit partiel est remplacé par une somme partielle et un report partiel, qui sont sauvegardés et passés à la ligne suivante. L'avantage de la multiplication de sauvegarde de report est que les additions à différentes positions de bit dans la même rangée sont maintenant indépendantes les unes des autres et peuvent être effectuées en parallèle, ce qui accélère essentiellement la phase d'addition de chaque cycle, et donc accélère la multiplication. L'addition de la somme partielle et de la retenue partielle à la dernière étape est effectuée, qui peut être mis en œuvre soit comme un additionneur de retenue d'ondulation, soit comme un additionneur de retenue.

**(c) Multiplicateur de poids corporel**

La difficulté de la multiplication du complément à 2 réside dans la gestion des bits de signe du multiplicande et multiplicateur. Un moyen efficace de surmonter ce problème est fourni par l'algorithme de multiplication de Bough-Wooley. L'algorithme est un moyen efficace de gérer le signe bits et aide à concevoir des multiplicateurs réguliers en utilisant les opérandes complémentaires de 2. La multiplication Baugh-Wooley peut être implémentée sous la forme d'un tableau de report ou d'un tableau de sauvegarde de report. La figure 1c montre le schéma d'un multiplicateur BW à 4 bits implémenté en tant que tableau de sauvegarde-report.

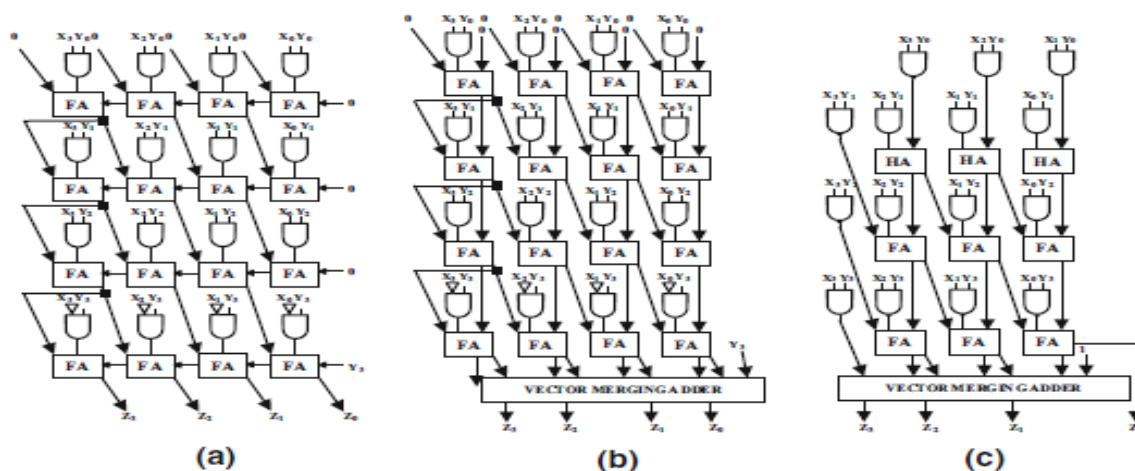


Figure 2 .8 Multiplicateur de tableau RCA (a) , CSA(b) et Multiplicateur de poids corporel(c)

[46].

**2.7.2 Multiplicateur de booth:** C'est un algorithme puissant pour la multiplication de nombres signés, qui traite les nombres positifs et négatifs de manière uniforme. C'est une opération (Add-décalage ). Si le multiplicateur est très grand, alors un grand nombre de multiplicandes doivent être ajoutés. Dans ce cas, le retard du multiplicateur est déterminé par le nombre d'addition effectué. S'il existe un moyen de réduire le nombre d'ajouts, les performances s'amélioreront. L'algorithme de Booth est une méthode qui réduit le nombre de multiplicandes.

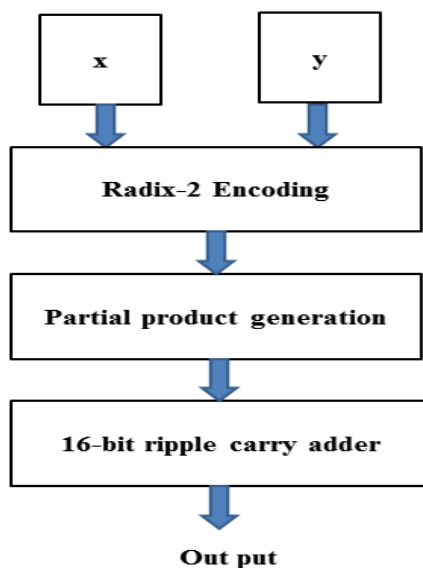


Figure 2.9 Organigramme standard du multiplicateur Booth [47]

**2.7.3. Le multiplicateur VEDIC :** Dans l'approche des mathématiques VEDIC, les produits partiels sont calculés à l'avance, avant même le début des opérations réelles de multiplication. Ensuite, le produit final est obtenu en ajoutant ces produits partiels selon l'algorithme VEDIC. Cette approche fournit à son tour une multiplication très grande. Les performances du multiplicateur peuvent être améliorées en réduisant le nombre de produits partiels générés et en accélérant l'accumulation de produits partiels.

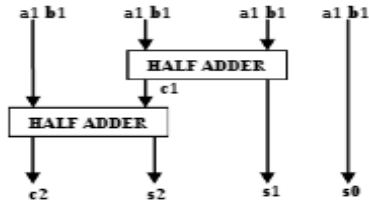


Figure 5: 2x2 Vedic Multiplier

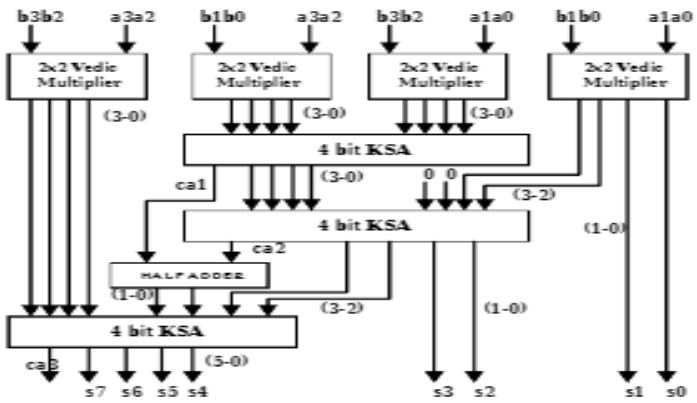


Figure 6: 4x4 Vedic Multiplier using KSA

Figure 2.10 Élément de base Vedic 2x2 bits et Élément de base Vedic 8x8 bits [48]

### 2.7.4. CORDIC

L'algorithme CORDIC a été proposé en 1959 (ordinateur de rotation coordonné) s'est avéré un moyen efficace d'évaluer les fonctions élémentaires, telles que les fonctions trigonométriques, exponentielles et logarithmiques de manière itérative. CORDIC peut être utilisé pour effectuer des rotations vectorielles complexes selon des angles arbitraires. Cette approche est utilisée dans la FFT pour calculer les facteurs des torsions. L'algorithme CORDIC fournit une méthode itérative pour effectuer des rotations vectorielles selon certains angles, pour calculer le cosinus et le sinus comme illustré à la figure 2.11, en utilisant uniquement la multiplication ou la division par deux ce qui se traduit en pratique par des décalages à gauche ou à droite.

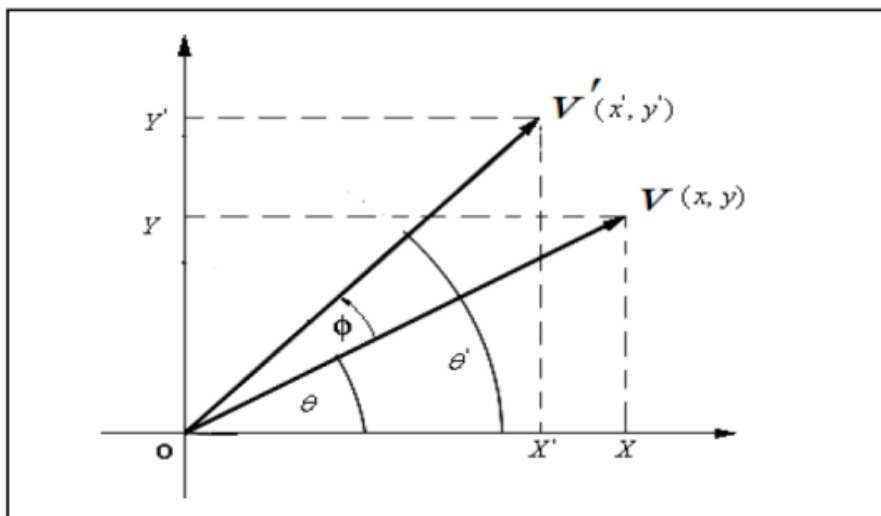


Figure 2.11 :Rotation du vecteur V par l'angle  $\Phi$

$$X' = X \cos(\Phi) - Y \sin(\Phi) \quad (2.9)$$

$$Y' = Y \cos(\Phi) + X \sin(\Phi)$$

Où (X', Y') sont les coordonnées du vecteur résultant après rotation d'un vecteur de coordonnées (X, Y) à travers un angle  $\Phi$  dans le plan rectangulaire. Ces équations peuvent être écrites comme suit:

$$X' = \cos(\Phi) \cdot [X - Y \tan(\Phi)] \quad (2.10)$$

$$Y' = \cos(\Phi) \cdot [Y + X \tan(\Phi)]$$

Si les angles de rotation sont limités de telle sorte que :  $\tan(\varphi) = 2^{-i}$ , alors le terme de multiplication tangente est réduit à une opération de décalage. On peut donc trouver des angles de rotation en effectuant des rotations élémentaires continues plus petites [49]. L'équation ci-dessus pour la rotation peut être exprimée comme suit :

$$X_{i+1} = K_i \cdot [X_i - Y_i \delta_i 2^{-i}] \quad (2.11)$$

$$Y_{i+1} = K_i \cdot [Y_i + X_i \delta_i 2^{-i}]$$

$k_i = \frac{1}{\sqrt{1+2^{-2i}}}$  et  $\delta_i = \pm 1$ . Le produit  $K_i$  approche de la valeur 0,607312 [50].

## 2.7. Conclusion

Dans ce chapitre, ont été présentée une étude théorique des différents algorithmes FFT. L'algorithme rader était la méthode la plus adapté à notre travail. Dans le terme mathématique tous les algorithmes FFT utilisent des multiplicateurs. Cette opération mathématique, lorsqu'elle est implémentée dans le matériel, nécessite généralement beaucoup de ressources, ce qui n'est pas toujours souhaité. En plus, dans la deuxième partie de ce chapitre, ont été présentés quelques types des multiplieurs peu utilisés à l'implémentation matérielle. L'utilisation d'algorithme CORDIC est très utile pour la mise en œuvre d'FFT en temps réel ce qui nous a permis de remplacer les multiplications par une séquence d'additions et de soustractions.

## **2.8.Référence de chapitre 2**

[35] natarajan, hariharan, implementation of chirp- z discrete fourier transform on virtex ii fpga,2004

[36] Chokri N Conception et implementation sur FPGA de structures multiplexes de la FFT en pipeline,université du quebec a trois-rivieres,2017

[37] A.W.M ,Traitement Numerique du signal 3e édition ,Dound 2003

[38] y. Wang, Y.Tang, Y. Jiang, 10. Chung, S.S. Song and M.S. Lim, " Novel Memory Reference Reduction Methods for FFT Implementations on DSP Processors", IEEE Transaction on Signal Processing , vol. 55, no 5, pp. 2338 - 2349,2007.

[39] Fu.B, Ampadu.P, "An Area Efficient FFT/IFFT Processor for MIMO-OFDM WLAN 802.11 n", Journal of Signal Processing System, Springer, vol. 56, pp 59- 58, september 2009.

[40] Y. Jung, H. Yoon, et J. Kim, "New Efficient FFT Aigorithrn and Pipeline Implementation Results for OFDMIDMT Applications", International Conference on Consumer Electronics, Feb 2003, Hong Kong.

[41]Youcef A ,implementation efficace de la FFT POUR des communication OFDM, universite du quebec a trois-rivieres,2010

[42] John G. Proakis, Dimitris G.Manolakis, "Digital Signal Processing, Principles, Aigorithms, and applications", Third Edition, Prentice hall.

[43] Oppenheim, A. V, Schafer. R. W, "Digital Signal Processing", Prentice-Hall, 1975.

[44]Mohamed Amine B , Adaptation du calcul de la Transformée de Fourier Rapide sur une architecture mixte CPU/GPU intégrée thèse UNIVERSITE NICE-SOPHIA ANTIPOLIS ,2015.

[45] [https://fr.wikipedia.org/wiki/Transformation\\_de\\_Fourier\\_rapide](https://fr.wikipedia.org/wiki/Transformation_de_Fourier_rapide)

[46] Burhan Khurshid “LUT based realization of fixed-point multipliers targeting state-of-art FPGAs” , Springer Science+Business Media New York 2017

[47] Sakshi R, Priya S “High Speed and Reduced Power –Radix-2 Booth Multiplier” IJCEM International Journal of Computational Engineering & Management, Vol. 16 Issue 2, March 2013

[48]aniket kumar vishikha comparative analysis of vedic & array multiplier 2017

[49] S. Younis, A. Al-Dweik1, ”CORDIC based Architecture for Blind CFO Estimation in OFDM Systems”, School of Electrical, Electronic and Computer Engineering, Newcastle University, U.K., 2011 IEEE.

[50] G.Gopikiran”FPGA Implementation of Floatingpoint Rotation Mode CORDIC Algorithm”, ECE Department National Institute of Technology,International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011).

# Chapitre 3 :

## Simulation de la modulation OFDM

3.1.Introduction .....	43
3.2 Les circuits programmables .....	43
3.3 Les éléments d'un circuit FPGA .....	44
• a)Les éléments logiques.....	44
• b) Les éléments de mémorisation .....	45
• c) Les éléments de routage.....	45
• d) Les éléments d'entrées/sorties .....	45
• e) Les éléments de contrôle et d'acheminement des horloges.....	45
3.4. Les deux grandes familles des FPGA.....	45
• 3.4.1. La famille Xilinx.....	45
• 3.4.2. La famille Altera.....	46
3.5. Les outils de conception.....	45
•3.5.1 Le langage de description matérielle.....	46
•3.5.2. Modelsim et Quartus II .....	48
•3.5.3. Logiciel Xilinx ISE .....	48
3.6. Simulation de la chaine de transmission OFDM .....	49
3.7. Conclusion .....	58
3.8 Référence de chapitre 3.....	59

### **3.1.Introduction**

Les circuits programmables existent depuis longtemps, mais les circuits de type FPGA sont en train de révolutionner le monde, tout comme l'ont fait les DSPs quelques décennies auparavant. Des algorithmes très importants en traitement numérique, implémentés sur des PDSPs (Programmable DSPs) ou des ASICs, sont maintenant implémentés sur des circuits FPGAs grâce à l'évolution de la technologie et l'augmentation fulgurante de la densité d'intégration [51]. Ce chapitre est partagé en deux parties. La première présente d'abord ces circuits programmables et leurs caractéristiques ainsi que les outils d'aide à la conception de circuits sur FPGAs. Dans la seconde partie, une simulation de la chaîne de transmission OFDM est exposée en détail.

### **3.2 Les circuits programmables :**

Actuellement, on trouve différentes familles de circuits programmables ou PLDs (Programmable Logic Devices) tels que les CPLDs (*Complex Logic Programmable Devices*), les ASICs, les DSPs et les FPGAs. Les CPLDs sont des composants pour la plupart reprogrammables électriquement ou à fusibles, peu onéreux et très rapides (fréquence de fonctionnement élevée). Cependant, ils disposent d'une capacité fonctionnelle moindre que les FPGAs. Par contre, ceux-ci sont des composants VLSI constitués de blocs mémoires vives, entièrement reconfigurables [52]. Ces blocs sont structurés en LUT (*Look Up Table*), flip-flop, RAM et l'ensemble dispose d'un vaste système d'interconnexions. Le DSP a longtemps régné en maître absolu dans les applications de traitement numérique du signal. Mais, depuis quelques années, il doit faire face au nouveau concurrent : le FPGA. Celui-ci a déjà quelques décennies de carrière mais l'intégration de nouvelles fonctions (mémoires, processeurs, etc.) lui ont permis de sortir de ses applications classiques de traitement d'opérations de logique booléenne et d'accéder aux applications complexes de traitement du signal, d'images et de vidéos. On le trouve de plus en plus sur les cartes standards. La technologie a beaucoup évolué ces derniers temps et les FPGA sont devenus aujourd'hui de véritables processeurs numériques des signaux, qui viennent concurrencer les composants DSP et ASIC. Malgré tout, les FPGA ne s'imposent pas si facilement dans les applications de traitement du signal. Il y a deux raisons à cela. La première, c'est que pour programmer efficacement un FPGA, il faut repenser les algorithmes fondamentaux du traitement du signal. De plus, il ne suffit pas d'avoir des compétences en logiciel, il faut également des compétences en électronique numérique (au niveau "matériel"). La deuxième raison qui freine

l'utilisation des FPGA, c'est que les outils de conception standard pour FPGA ne sont pas adaptés pour réaliser des applications de traitement du signal relativement complexes [53].

### **3.3 Les éléments d'un circuit FPGA [54]:**

Les éléments constitutifs d'un FPGA sont toujours approximativement les mêmes quelle que soit l'architecture choisie et chaque fabricant apporte ses variantes. Les FPGAs se composent d'une matrice de blocs logiques élémentaires (CLB) permettant de réaliser des fonctions combinatoires et des fonctions séquentielles. Tout autour de ces blocs logiques configurables, nous trouvons des blocs entrées/sorties (IOB) dont le rôle est de gérer les entrées-sorties réalisant l'interface avec les modules extérieurs et des ressources d'interconnexion (programmable interconnect) totalement flexibles (figureV.1)[55].

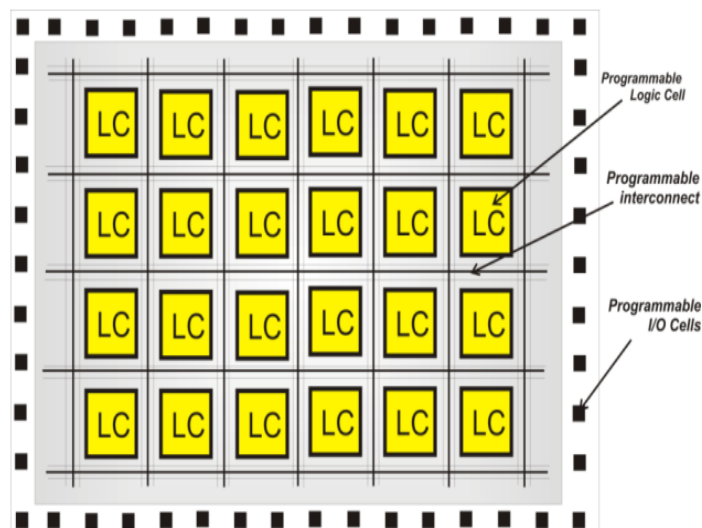


Figure 3.1:Structure de base d'un circuit FPGA

Généralement, ces éléments sont :

**a) Les éléments logiques [52] :**

Il s'agit des blocs de base de tout circuit FPGA. Ces blocs ont souvent la même constitution et cela malgré la différence de fabricants et d'architectures .Les structures sont généralement constitués d'une ou plusieurs tables LUT (*Look Up Table*) qui contiennent, après configuration, la table de vérité de la fonction logique qu'elles doivent réaliser ou alors un ensemble de valeurs qui sont mémorisées comme dans une mémoire ROM. Les tables LUT sont suivies d'un registre de sortie, ce qui permet de synchroniser, si nécessaire, la sortie sur

une horloge. La plupart des blocs logiques de bases sont munis d'une chaîne de propagation rapide de retenue afin de former de petits additionneurs rapides.

**b) Les éléments de mémorisation :**

Les FPGAs sont utilisés pour des applications plus importantes .La nécessité d'intégrer des blocs de mémoire directement dans l'architecture des FPGAs est vite devenue capitale. De cette façon, les temps d'accès à la mémoire sont réduits. En effet, il n'est plus nécessaire de communiquer avec des éléments extérieurs au circuit [52] .

**c) Les éléments de routage :**

Les éléments de routage sont les composants les plus importants dans les FPGAs. En fait, ces éléments représentent la plus grosse partie du silicium consommée sur la puce du circuit. Ces éléments sont très importants puisqu'ils vont déterminer la vitesse et la densité logique du système..

**d) Les éléments d'entrées/sorties :**

Le but des éléments d'entrées/sorties est de relier un circuit avec son environnement extérieur. Ceux-ci peuvent bénéficier de buffer ou d'autres éléments permettant la gestion des entrées et des sorties.

**e) Les éléments de contrôle et d'acheminement des horloges :**

L'horloge est un élément essentiel pour le bon fonctionnement d'un système électronique. Les circuits FPGA sont prévus pour recevoir une ou plusieurs horloges. Des entrées peuvent être spécialement réservées à ce type de signaux.

**3.4. Les deux grandes familles des FPGA**

Actuellement, on trouve sur le marché des circuits FPGA (de faible, moyenne et haute densité) produits par les deux principaux producteurs de circuits logiques programmables : Xilinx et Altera. Sur le même marché, on trouve plusieurs autres producteurs de circuits FPGA, on peut citer à titre d'exemple : Actel, Abound Logic, Achronix, Atmel, Cypress, Lattice Semiconductor, etc. L'architecture d'une cellule logique élémentaire (CLE) varie fortement d'un producteur à l'autre. Dans cette partie de notre travail, nous décrivons les deux sociétés Xilinx et Altera, avec des exemples pour chaque famille de FPGA.

**3.4.1. La famille Xilinx**

Xilinx, Inc. (1984) est une entreprise américaine de semi-conducteurs. Inventeur du FPGA avec un premier produit en 1985, Xilinx fait partie des plus grandes entreprises spécialisées dans le développement et la commercialisation de composants logiques programmables, et des services associés tels que les logiciels de CAO électroniques ; création des blocs IP et

formation. En effet, Xilinx vend également des spécifications d'architectures ("IP Cores"). Cela peut aller de fonctions très simples comme des compteurs jusqu'à des systèmes complets comme des microcontrôleurs. Xilinx fabrique une large gamme de FPGA et de CPLD pour diverses applications. En effet, l'offre commerciale de Xilinx est découpée en plusieurs gammes : (FPGA hautes performances : gamme Virtex, FPGA pour la fabrication en grande série : gamme Spartan et CPLD : gammes XC9500 et Coolrunner). Les plus onéreux sont les FPGA Virtex (Virtex II/pro, Virtex4 et Virtex5). Les composants Virtex5 sont disponibles, à bon prix, avec plusieurs catégories (LX, LXT, SXT, TXT et FXT). Les Virtex5 offrent une densité supérieure à 12 Méga portes logiques équivalente. Ils peuvent atteindre une fréquence de fonctionnement de 550 MHz. Ils sont fabriqués avec la technologie 65nm.

### **3.4.2. La famille Altera**

Fin 92, Altera (qui produit des EPLD et des MAX) lance sur le marché une famille de FPGA, FLEX 8000 (*Flexible Logic Element matrix*), dans le but de concurrencer les LCA de Xilinx et la famille AT6000 d'Atmel ; ce lancement coïncide avec l'arrivée sur le marché des EPLD de Xilinx. En 1995, Altera annonce le lancement de la famille FLEX10K qui constitue la deuxième génération de FPGA chez Altera ; cette famille vise le créneau de la forte densité d'intégration avec des capacités allant jusqu'à 250.000 portes utilisables. En 1997, Altera annonce le lancement de sa troisième génération de FPGA, la famille FLEX6K, qui vise le créneau du composant bas coût de capacité moyenne (jusqu'à 24.000 portes utilisables). Selon une enquête *Dataquest*, en 1994 25 % des conceptions d'ASIC portaient sur des complexités inférieures à 10.000 portes et 25 % entre 10.000 et 25.000 portes ; la famille FLEX, avec des complexités de composants jusqu'à 24.000 portes vise donc 50 % des conceptions d'ASIC. En 1999, Altera lance la famille APEX20K avec, comme ambition de couvrir une gamme de complexité allant de 60.000 à 1.500.000 portes.

## **3.5. Les outils de conception [52] :**

### **3.5.1 Le langage de description matérielle :**

La densité actuelle des fonctions logiques (portes et bascules) intégrées dans les PLDs est telle (plusieurs milliers, voire millions, de portes) qu'il n'est plus possible d'utiliser les outils de saisie d'un schéma pour développer les circuits. Les sociétés de développement et les ingénieurs ont voulu s'affranchir des contraintes technologiques des circuits et ont créé des langages dits de description matérielle de haut niveau (les HDL, Hardware Description Language). Deux d'entre eux ont émergé et sont

couramment utilisés : VHDL (VHSIC Hardware Description Language) et Verilog. Ces deux langages bénéficient du support de la quasi-totalité des logiciels [56]. Dans les années 80, le département de la défense aux Etats-Unis fait un appel d'offre pour développer un langage de description matérielle numérique unique. Le langage VHDL (VHSIC Hardware Description Language) est inventé pour répondre à ces critères. Ce langage se base sur le VHSIC (Very High Speed Integrated Circuit) qui est un projet de recherche national mené par le groupe IBM/Texas Instruments / Intermetrics. Ce langage, ouvert au domaine public en 1985, devient une norme en 1987 sous la dénomination d'IEEE 1076-1987. Des changements minimes ont été apportés pour la seconde normalisation en 1993 qui porte le nom VHDL'93. La dernière évolution de la norme est la norme IEEE 1076-2001. Cependant, la norme IEEE 1076-2001 ne permet pas seule la description mixte (numérique et analogique). Le Verilog et le VHDL ont des capacités techniques équivalentes [57]. Le choix du langage est souvent dicté par la " culture " de l'équipe recherche. De plus, ce choix est parfois imposé par les outils disponibles, les logiciels de simulation et de synthèse, dont les langages associés sont fixés par des aspects économiques. VHDL a continué son évolution et la description analogique s'y est intégrée sous la référence IEEE 1076.6 (VHDL-AMS).

Le langage VHDL autorise trois types de description [58] : le type structurel qui décrit le câblage des composants élémentaires.

➤ Le type flot de données qui décrit les transformations d'un flot de données de l'entrée à la sortie.

➤ Le type comportemental qui décrit le fonctionnement par des blocs programme appelés Processus qui échangent des données au moyen de signaux comprenant des instructions séquentielles.

En VHDL [59], une structure logique est décrite sous la forme d'une paire de fonctions, d'une part une entité (ENTITY) et d'autre part, l'architecture (ARCHITECTURE) de la façon suivante :

```
> LIBRARY LIBNAME;
> USE LIBNAME.PACKAGENAME.ALL;
> ENTITY ENTITY_NAME IS
> PORT (
>     SIGNAL_NAME :MODE SIGNAL_TYPE;
>     .
>     .
>     SIGNAL_NAME : MODE SIGNAL_TYPE);
> END ENTITY_NAME;
>
> ARCHITECTURE ARCHITECTURE_NAME OF ENTITY_NAME IS
> DECLARATION DE COMPOSANTS
> + DECLARATION DES SIGANUX INTERNES
> + AUTRE DECALARATION ...
> BEGIN
> INSTRUCTIONS CONCURENTES;
> PROCESSUS;
> INSTANCES DE COMPOSANT;
> END ARCHITECTURE_NAME;
```

Figure 3.2: Instruction de code VHDL

L'entité décrit les signaux d'entrées et de sorties de la structure ainsi que leurs noms et leurs types. Par contre l'architecture décrit le comportement de l'entité. Il est possible de créer plusieurs architectures pour une même entité où chacune décrit l'entité de façon différente.

### **3.5.2. Modelsim et Quartus II :**

Modelsim est un logiciel de Mentor Graphics qui permet principalement, en utilisant le langage VHDL ou Verilog, de concevoir des systèmes numériques sous forme de projets, les décrire sur plusieurs niveaux (fichiers VHDL) et les simuler à l'aide de fichiers testbench. Modelsim ne permet que l'analyse fonctionnelle.

Quartus II est un environnement de conception de systèmes en vue de les implémenter sur FPGA. Tout comme Modelsim, Quartus II permet de décrire les circuits en VHDL mais aussi sous forme graphique ou schématique [60]. Il permet de les simuler, de les analyser aussi bien fonctionnellement que temporellement afin de juger leurs performances et enfin de programmer la cible (FPGA).

### **3.5.3. Logiciel Xilinx ISE**

ISE est un environnement intégré de développement de systèmes numériques ayant pour but la synthèse/implémentation matérielle sur FPGA. Les designs peuvent être décrits sous trois formes principales : sous forme de schémas, sous forme de HDL ou bien sous forme de diagrammes d'états. Pour cela, ISE intègre différents outils permettant de passer à travers tout le flot de conception d'un système numérique.

### 3.6. Simulation de la chaîne de transmission OFDM

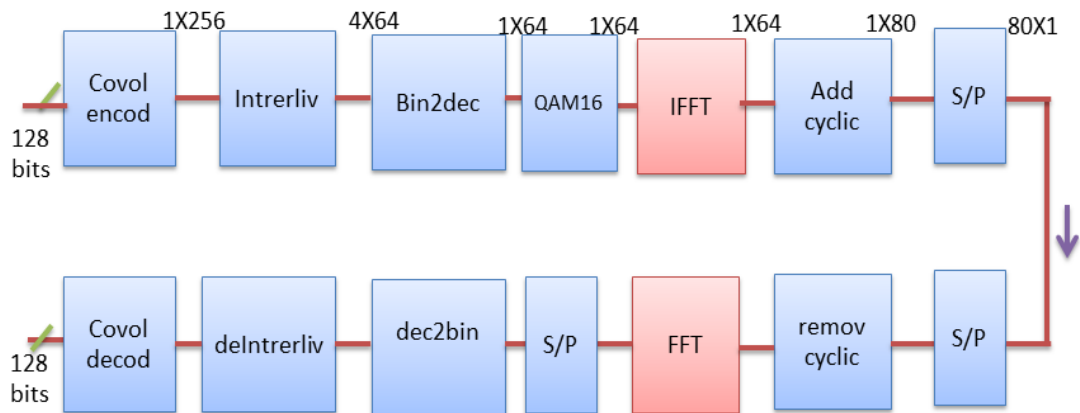


Figure 3.3:mise en œuvre d'une chaîne de transmission utilisant l'OFDM

Dans cette partie ont été testé le bon fonctionnement de la chaine OFDM classique sans passé au canal de transmission pour but de calculer les ressources matérielle FPGA du chaque bloc de la chaine OFDM. Pour cette raison ont été créés une scripte sur l'environnement matlab qui simule la chaine OFDM on utilisant les fonctions de librairie du simulateur Matlab. Les dernies versions de Matlab, Permet de convertie le code Matlab en code Vhdl à partir des étapes à suivre.

Premièrement transférer le code malab sous formes une fonction et créer un autre code qui contient le teste bench .

La deuxième étape est de choisir le mode de démarrage du HDLcoder à travers l'utilitaire APPS .Cette étape permet de créé un nouveau projet vhdl. Ont été chargés la fonction et le teste bench .

Après la génération de code vhdl de la fonction désirer ont été testé le bon fonctionnement de ce code sur l'environnement Modelsim 6.3 et ISE14.7 qui permet du donné le rapport de ressource matériel utilisé .Nous appliquons les étapes précédents sur tous les blocs de la chaine OFDM classique.

**3.5.1 Bloc de codage convolutif des données** Un codeur convolutif accepte des messages de longueur de k bits et génère des mots de code de n bits. Généralement, il est constitué d'un registre à décalage de L segments où L rétrograde la longueur de contrainte. Le codeur convolutif utilise la fonction «poly2trellis» pour créer un treillis en utilisant la longueur de contrainte. La structure en treillis est celle spécifiée dans la norme: poly2trellis (7, [37 21]) pour un taux de code de 1/2.

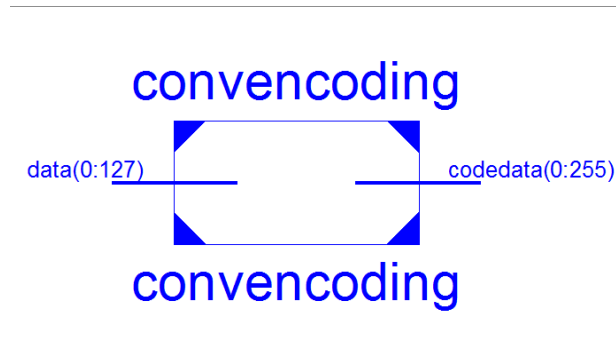


Figure 3.4: Bloc de codage convolutif

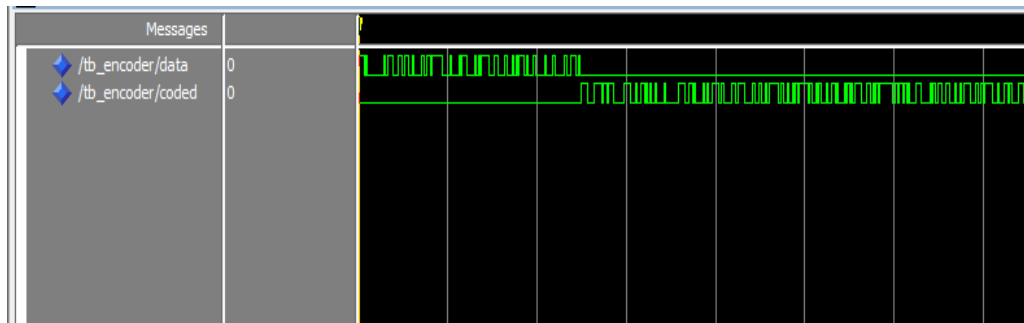


Figure 3.5: Données codées par codage convolutif sous modelsim

La figure 3.5 présente le chronogramme de codage convolutif donné par modelsim .

**datain** : Signal de chargement en entrée, il détermine l'entrée série des données résiduelles à traiter ( 128bits).

**coded** : Signal de sortie de codeur après l'insertion des bits de redondance. Ensuite, cette sortie de 256 échantillons est traitée par le bloc entrelacé

<b>DeviceUtilizationSummary (estimated values)</b>		
<b>LogicUtilization</b>	<b>Used</b>	<b>Utilization</b>
Number of Slice LUTs	3328	1%
Number of occupied Slices	2143	2%
Number of bonded IOBs	384	40%

Tableau 3.1 Consommation des ressources de bloc de codage convolutif

### 3.5.2.L'entrelacement

La principale tâche de cette opération est de modifier les flux de données afin de réduire l'effet des erreurs introduites dans le signal dans le canal de transmission. Cela dépend du système de codage de canal utilisé pour récupérer les données d'origine. Il y a deux entrelacements. Le premier entrelacement est un entrelacement par bloc appelé inter-trame. Les opérations effectuées sont des permutations de colonnes dont l'ordre est prédéfini. Le deuxième entrelacement appelé intra-frames est réalisé par blocs. Les bits d'entrée sont structurés sous forme de matrice. Les permutations interviennent entre colonnes.

L'entrelacement des données: la technique de l'entrelacement revient à permuter les symboles codés avant leur transmission. Cette opération rend les codes correcteurs plus rentables et efficaces.

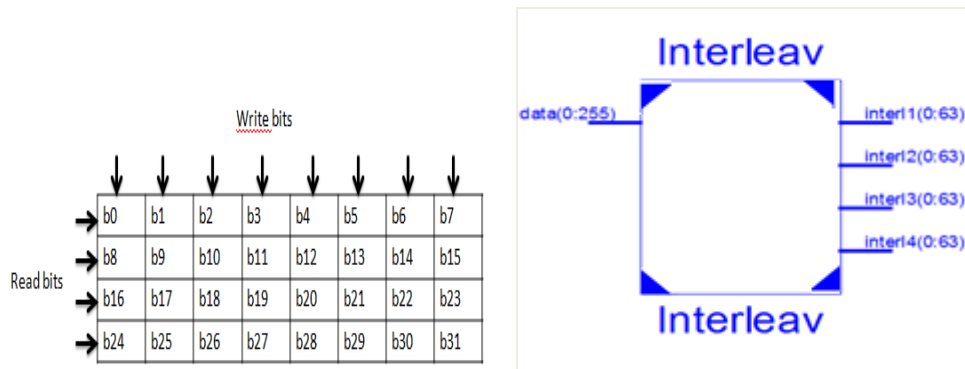


Figure 3.6 : Exemple d'entrelacement

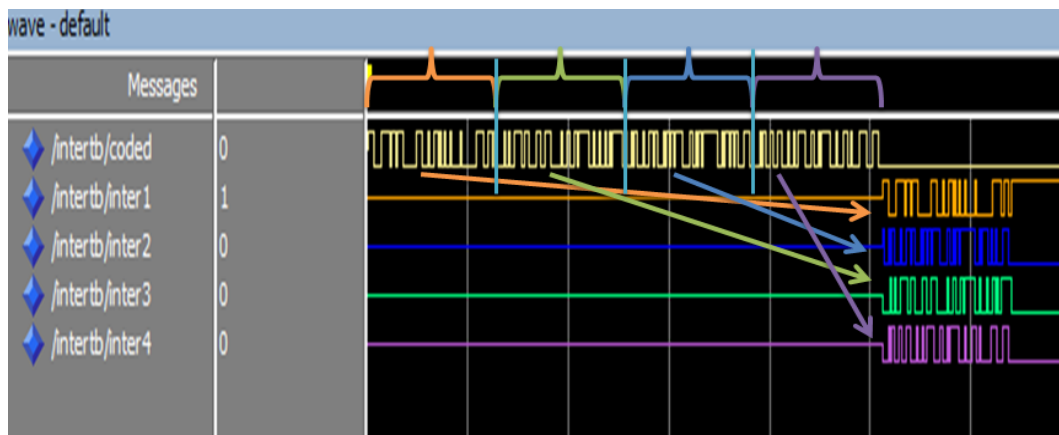


Figure 3.7 : Chronogramme d'entrelacement

Dans ce bloc d'entrelacement les 256 bits d'entrées sont structurés sous forme d'une matrice (4\*64 ).Inter1 ,Inter2, Inter3 et Inter4 sont les lignes de la matrice résultante .

DeviceUtilizationSummary (estimated values)		
LogicUtilization	Used	Utilization
Number of Slice LUTs	4251	2%
Number of occupied Slices	2143	3%
Number of bonded IOBs	512	53%

Tableau 3.2 Consommation des ressources de l'entrelacement

### 3.5.3. Convertisseur binaire -décimale

Tout d'abord le convertisseur **binaire -décimale** est utilisé pour convertir les données binaires à la forme décimale de 4 bits pour préparer l'entrée du bloc mapping.

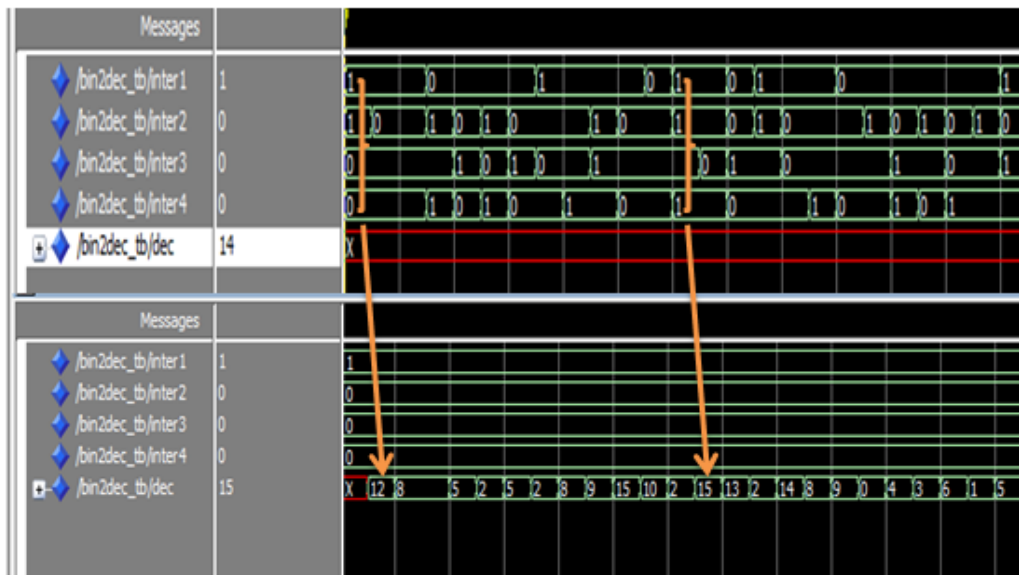


Figure 3.8: Résultat de convertisseur **binaire –décimale**

La figure3.8 présente une partie de résultat de convertisseur binaire décimale .le résultat compris entre 0 et 15.

**3.5.4. Bloc mapping** -Pour un choix de la modulation QAM16, l'entrée du bloc mapping étant le flux des données de 4 bits varié entre 0 et 15. La sortie sera séparée en deux signaux qui représentent la partie réelle et imaginaire. La figure 3.8 présente la constellation de la modulation QAM16.

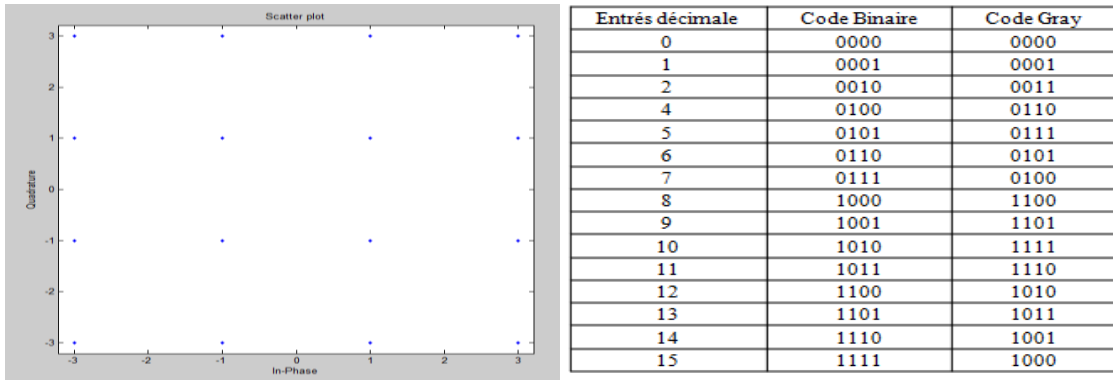


Figure 3. 9: Constellation de la modulation QAM16

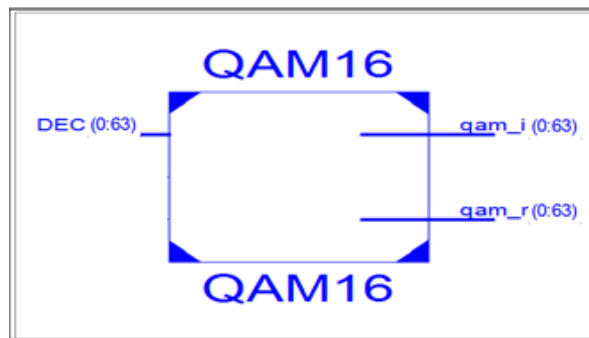


Figure 3.10: Bloc **mapping**

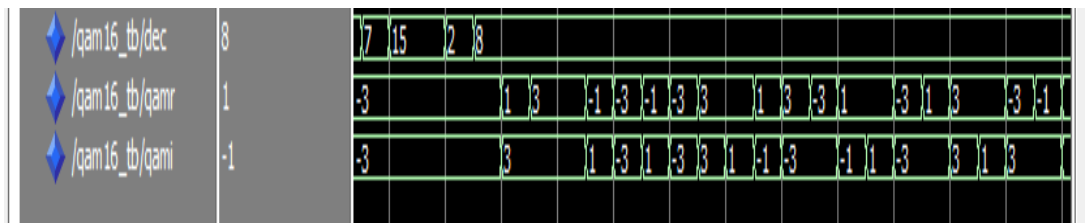


Figure 3.11: Résultat de modulation QAM16

La figure 3.11 illustre les 2 parties réelles et imaginaires des 20 premières sorties de codeur QAM16 parmi les 64 sorties.

DeviceUtilizationSummary (estimated values)		
LogicUtilization	Used	Utilization
Number of Slice LUTs	2267	1%
Number of occupied Slices	1075	1%
Number of bondedIOBs	192	20%

Tableau 3.3 Consommation des ressources de 16QAM

3.5.5. Bloc d 'IFFT /FFT: Comme les autres blocs de la chaine ofdm ont été utilisé le librairie du Matlab pour réaliser les blocs (ifft) /(fft) .

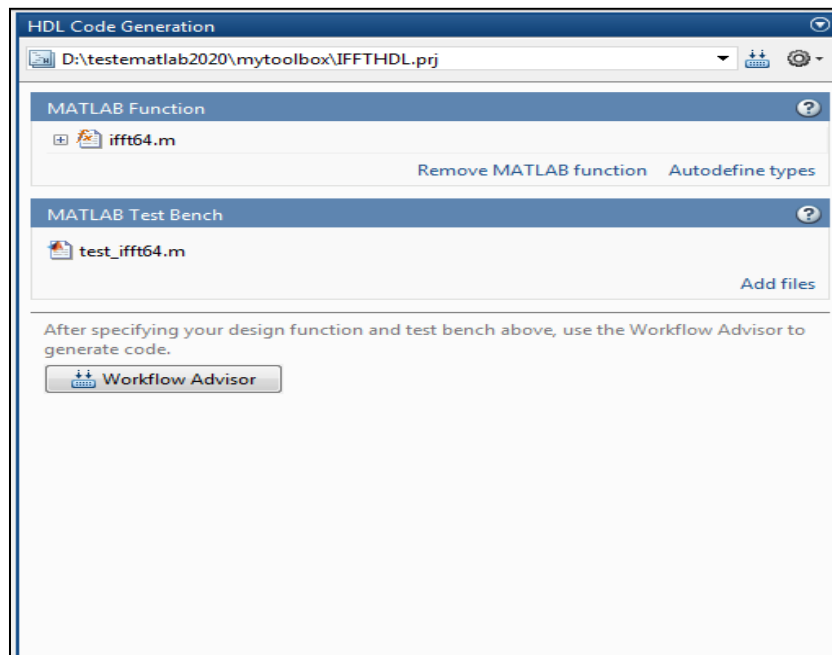


Figure 3.12 :Fenêtre de chargement de fonction et testbench

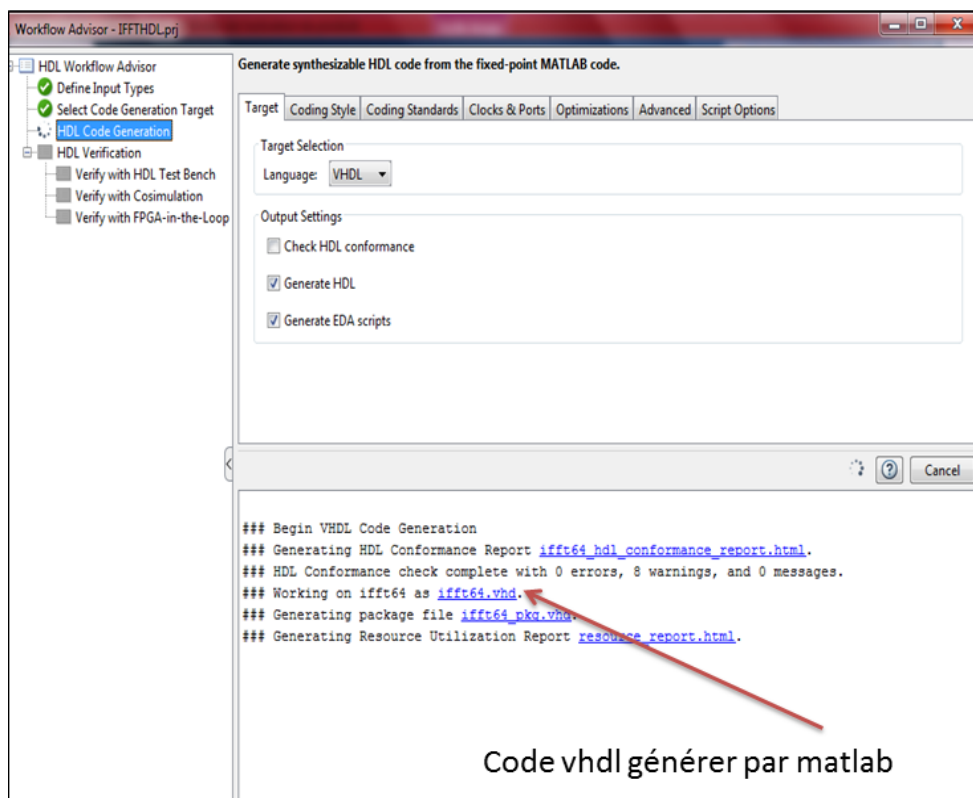


Figure 3.13 : Génération de code vhdl d' ifft

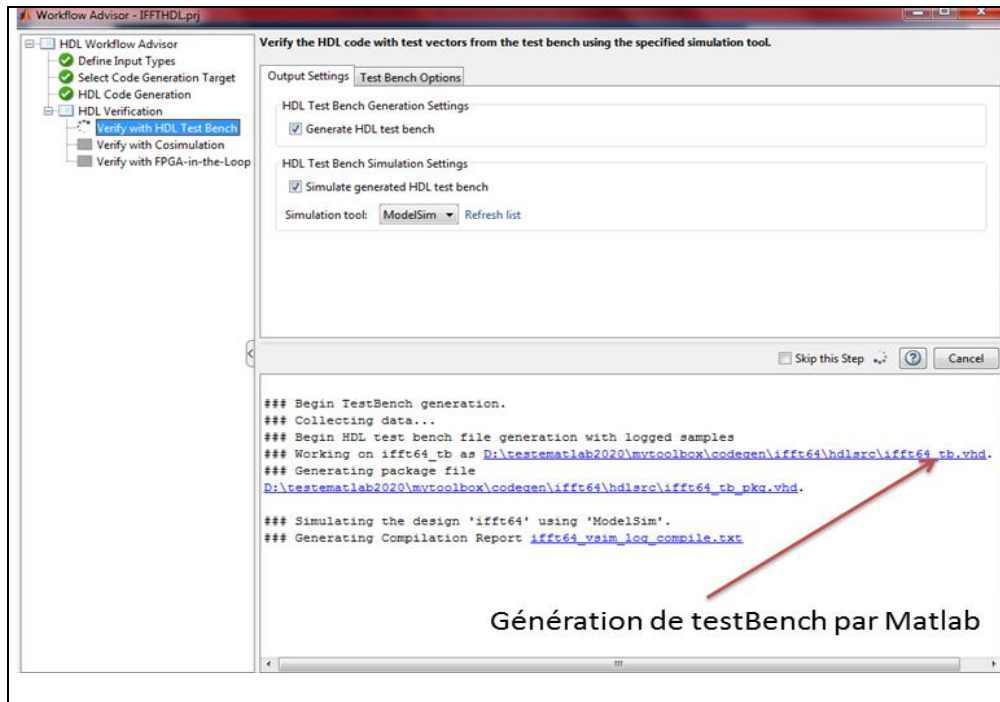


Figure 3.14 : Génération de testbench d'ifft

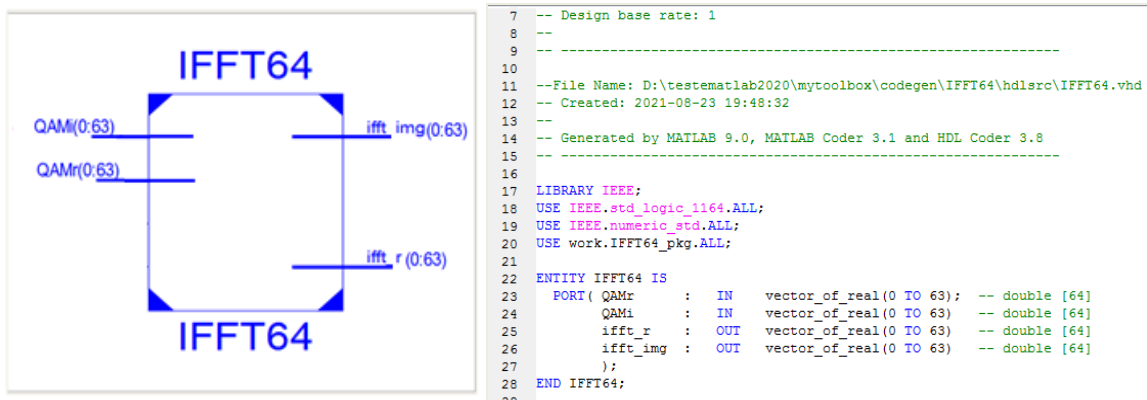


Figure 3.15 : top level de l'IFFT & Déclaration des donnes d'ifft 64

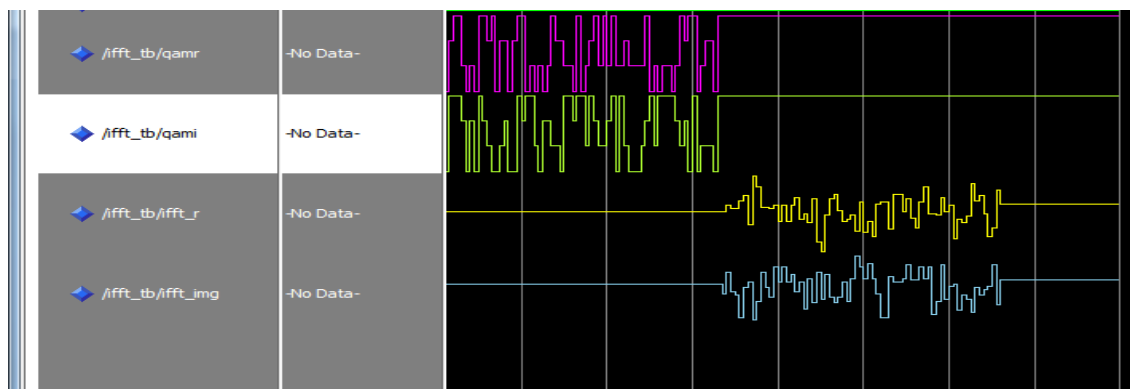


Figure 3.16: Résultat d'IFFT

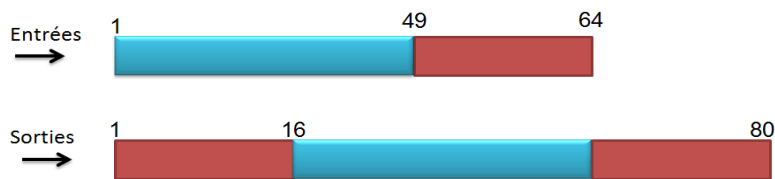
La figure 3.16 présente, les parties réelles et imaginaires d'IFFT donné par modelsim. Le traitement des données en séries, et le résultat des parties réelles et imaginaires d'IFFT sont générés après 64 cycles d'horloge.

Target Device	XC4vlx200-11ff1513	
<b>DeviceUtilizationSummary (estimated values)</b>		
<b>LogicUtilization</b>	<b>Used</b>	<b>Utilization</b>
Number of Slice LUTs	7859	4%
Number of occupied Slices	5381	6%
Number of bonded IOBs	256	27%

Tableau 3.4 Consommation des ressources d'IFFT

**3.5.6. Bloc d'insertion du préfixe cyclique**

Le préfixe cyclique est une copie des 16 derniers échantillons de l'IFFT, qui sont placés au début de la trame IFFT.



La figure 3.17 présente le résultat de simulation pour insérer le préfixe cyclique au début de signal.

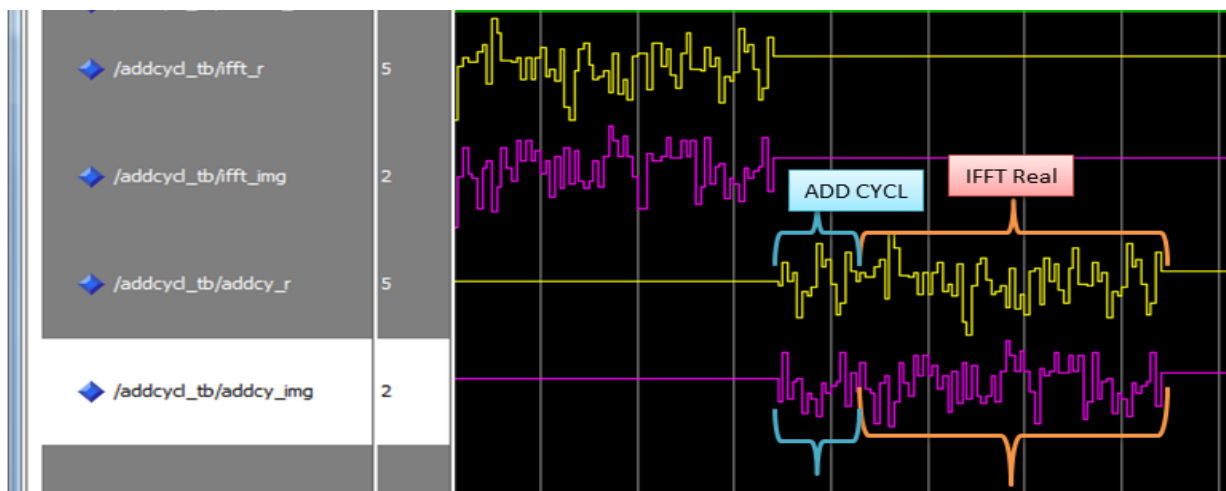


Figure 3.17: partie réel et imaginaire du préfixe cyclique

### 3.5.7. Bloc Convertisseur série-parallèle

L'objectif du convertisseur série-parallèle est de recevoir les données à transmettre. Le convertisseur série-parallèle reçoit les M bits de série à transmettre, et ces bits seront divisés en N sous-blocs de MN bits, chaque sous-bloc appelé symboles.

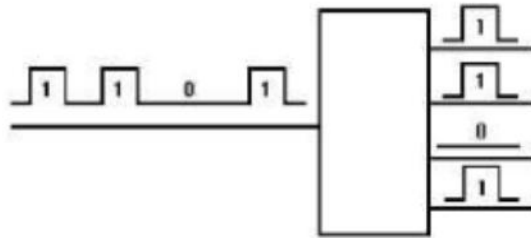


Figure 3.18: Conversion série-parallèle

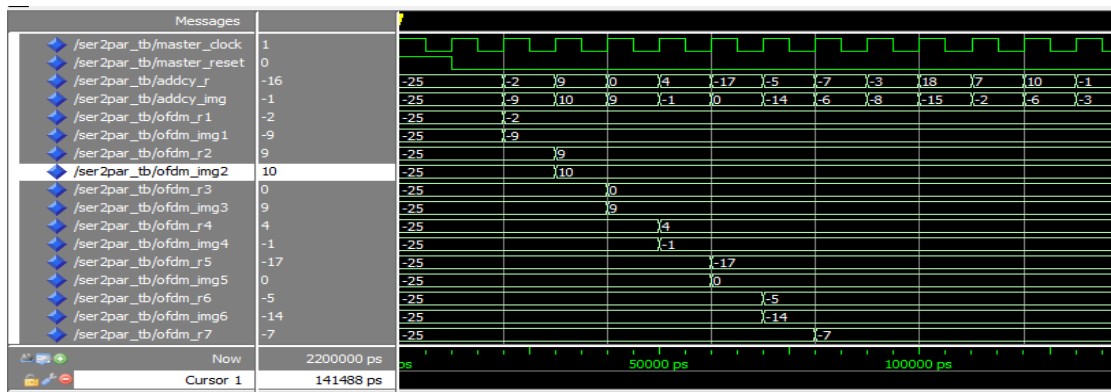


Figure 3.19 : Simulation de Conversion série-parallèle

Le résultat obtenu à la sortie du bloc **convol decod** présenté sur le chronogramme de la figure 3.20. Il s'avère que le résultat obtenu à la fin de la chaîne de réception est le même données émis c a d la restitution de signal émis .Cela nous montre que tous les blocs de la chaîne de transmission fonctionnent correctement.

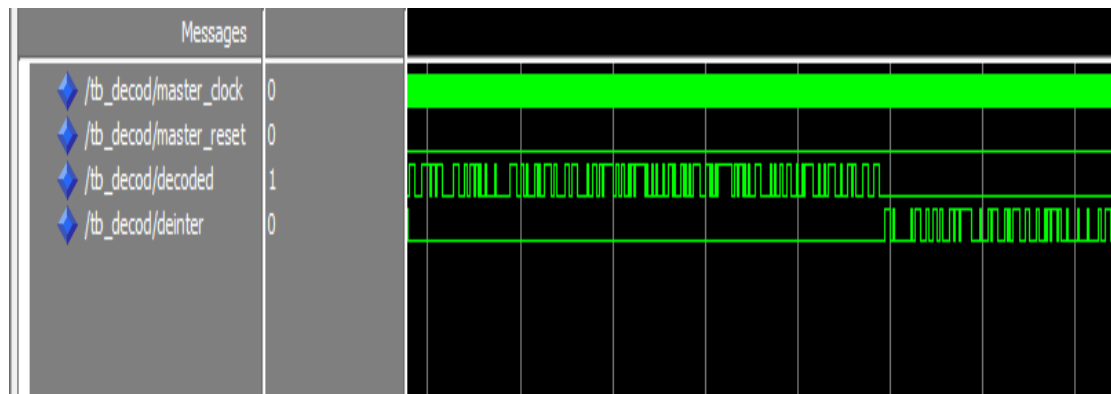


Figure 3.20: Restitution de signal émis sous Matlab et Modelsim

On remarque que les blocs IFFT et l'entrelacement présentent la plus grande consommation en ressource FPGA par rapport aux autres blocs d'une chaîne d'émission OFDM.

### **3.7. Conclusion**

Dans ce chapitre, ont été présentés différentes familles de circuits programmables et particulièrement les circuits FPGA de la famille Altera et Xilinx. Ensuite, on a présenté les étapes suivies qui nous ont permis de générer un code VHDL à partir d'un code Matlab. Dans cette partie, ont été testés le bon fonctionnement de la chaîne OFDM classique sans passer au canal de transmission pour mettre en évidence l'importance des blocs IFFT/FFT par rapport à toute la chaîne OFDM. L'estimation des ressources des conceptions d'émetteur et de récepteur a été synthétisée à l'aide de Xilinx ISE Design Suite 12.2. Il faut, enfin, signaler que l'implémentation des blocs inverses de la chaîne OFDM se fait en appliquant les mêmes procédures décrites dans ce chapitre.

### **3.7. Référence de chapitre 3**

- [51] Uwe Meyer-Baese, —Digital Signal Processing with Field Programmable Gate Arrays, Third Edition, ISBN 978-3-540-72612-8 Springer Berlin Heidelberg New York, 2007
- [ 52] NACHEF Toufik, Implémentation d'une instrumentation sur un FPGA, Memoire de Magister,univ Tizi-ouzou,2011.
- [53] F. Gohzzi, —Optimisation d'une Bibliothèque de Modules Matériels de Traitement d'Images. Conception et test VHDL, Implémentation Sous Forme FPGA, Thèse, No.2789, Ecole doctorale de Sciences Physiques et de l'Ingénieur, Université de Bordeaux, 26 janvier 2004.
- [54] Les composants FPGA s'invitent dans le traitement du signal , Patrick Méchin Directeur général de Techway, Mesures 775 - MAI 2005 - [www.mesures.com](http://www.mesures.com)
- [55] Ahmed Ben Atitallah, —Etude et Implantation d'Algorithmes de Compression d'Images dans un Environnement Mixte Matériel et Logiciel, thèse de doctorat en électronique, Université Bordeaux 1, Juillet 2007.
- [56] A. Nkesta, —informatique industrielle, circuits logiques programmables, mémoires, PLD, CPLD et FPGA | Editions Ellipses, 1998.
- [57] D.Smith — HDL Chip Design : A practical Guide for Designing, Synthesis & Simulating Asics & FPGAs using VHDL or Verilog, Doone Pubns . ISBN : 0965193438
- [58] <http://www.angelfire.com/ca/verilog/history.html>
- [59] Sanjay Churiwala , Designing with Xilinx® FPGAs ,Circuits &Systems, Springer,2017.
- [60] Etienne Messerli ,Manuel VHDL synthèse et simulation, Reconfigurable & Embedded System ,2007

**Chapitre 4:**

**Méthodologies d'implémentation d'FFT et discussion des résultats.**

4.1 Introduction .....	60
4.2 Multiplicateur complexe .....	60
4.3 L'architecture proposée de 8 points FFT .....	62
4.4 Synthèse matérielle pour FFT à 8 points .....	64
4.5. L'architecture proposée d 'FFT à base d'algorithme CORDIC .....	64
• 4.5.1 Résultats et discussion .....	69
4. 6. L'architecture proposée d 'FFT pour des taille premier à base d'algorithme Rader .....	71
• 4.6.1 L'algorithme Rader .....	71
• 4.6.2 Structure de filtre .....	72
• 4.6.3 Méthodologie d'implémentation de processeur DFT .....	74
• 4.6.4. Résultats et discussion .....	80
4.7. Conclusion.....	83
4.8 Référence de chapitre 4.....	85
<b>Conclusion et perspectives</b> .....	<b>87</b>

#### 4.1 Introduction

Le choix de l'architecture utilisée pour implémenter la FFT est un élément très important et très délicat. L'élément papillon est le bloc de construction le plus important du processeur FFT. L'addition et la multiplication sont les deux opérations arithmétiques principales impliquées dans le calcul de l'algorithme FFT. La conception du multiplicateur complexe influe sur la complexité informatique globale de l'algorithme FFT. L'objectif principal de cette partie est de mettre en œuvre la conception du papillon hautement efficace, qui pourrait être utilisée dans une architecture FFT. Dans ce chapitre on étudiera les deux architectures les plus communes en termes d'efficacité d'implémentation de la FFT : l'architecture avec multiplieur et l'architecture à base d'algorithme CORDIC. Ainsi, la deuxième partie de ce chapitre présente l'implémentation de la FFT avec des tailles premières en utilisant l'algorithme Radé.

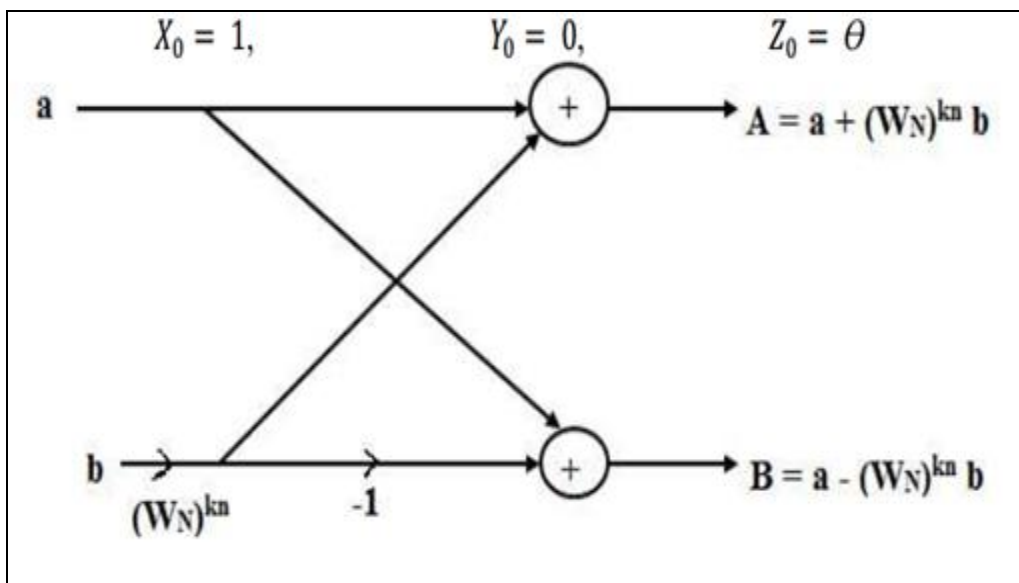


Figure 4.1: Structure papillon de 2 point

#### 4.2 Multiplicateur complexe

La multiplication de deux nombres complexes est une opération très fréquente dans de nombreux algorithmes de traitement du signal. Un multiplicateur complexe est une combinaison d'un nombre réel et d'un nombre imaginaire. En général, deux nombres complexes  $(a + jb)$  et  $(c + jd)$  sont multipliés comme suit  $(a + jb)(c + jd) = (ac - bd) + j(ad + bc)$ .

La figure 4.2 montre le schéma de principe du multiplicateur complexe. Ici, 4 multiplicateurs et 3 blocs additionneurs / soustracteurs sont utilisés. Tout d'abord, les deux nombres complexes  $(a + jb)$  et  $(c + jd)$  sont introduits à l'entrée du multiplicateur comme indiqué sur la Figure 4.1. Le multiplicateur génère les produits partiels  $ac$ ,  $bd$ ,  $ad$  et  $bc$ . Ensuite, la somme et la différence entre les produits partiels sont calculées en utilisant respectivement un additionneur et un soustracteur.

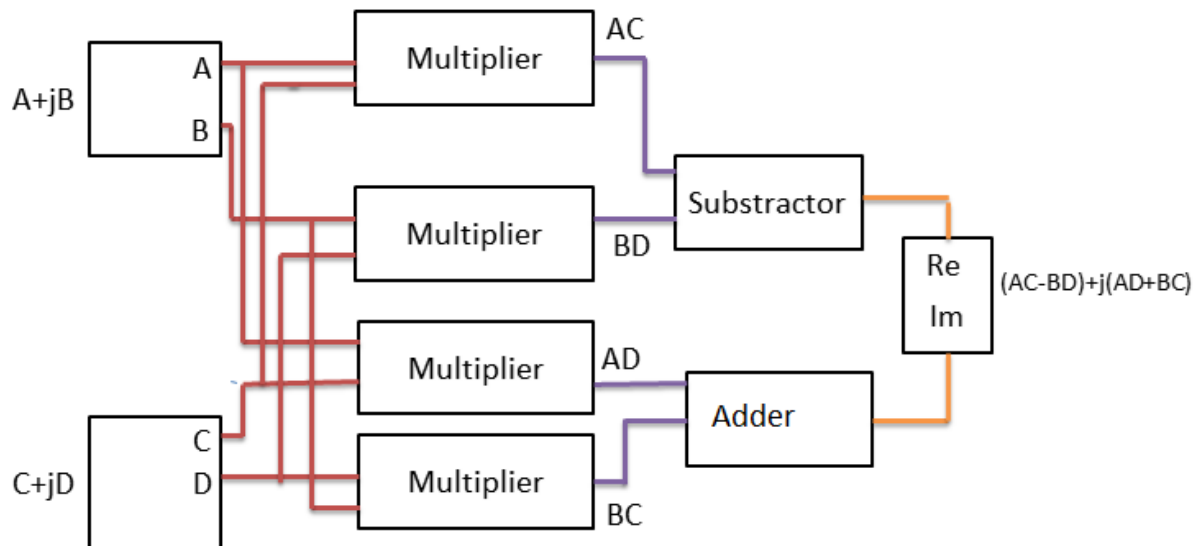


Figure 4.2: Diagramme de bloc de multiplicateur complexe

### 4.3 L'architecture proposée de 8 points FFT

Le processus de mise en œuvre a été divisé en trois composants qui sont un séparateur, bloc d'FFT à 4 points et bloc de la TFD à 8 points. Le composant séparateur est la première étape du processus de conception et sa tâche consiste à diviser le flux de données à filtrer en échantillons pairs ( $X_{\text{even}}$ ) et impairs ( $X_{\text{odd}}$ ).

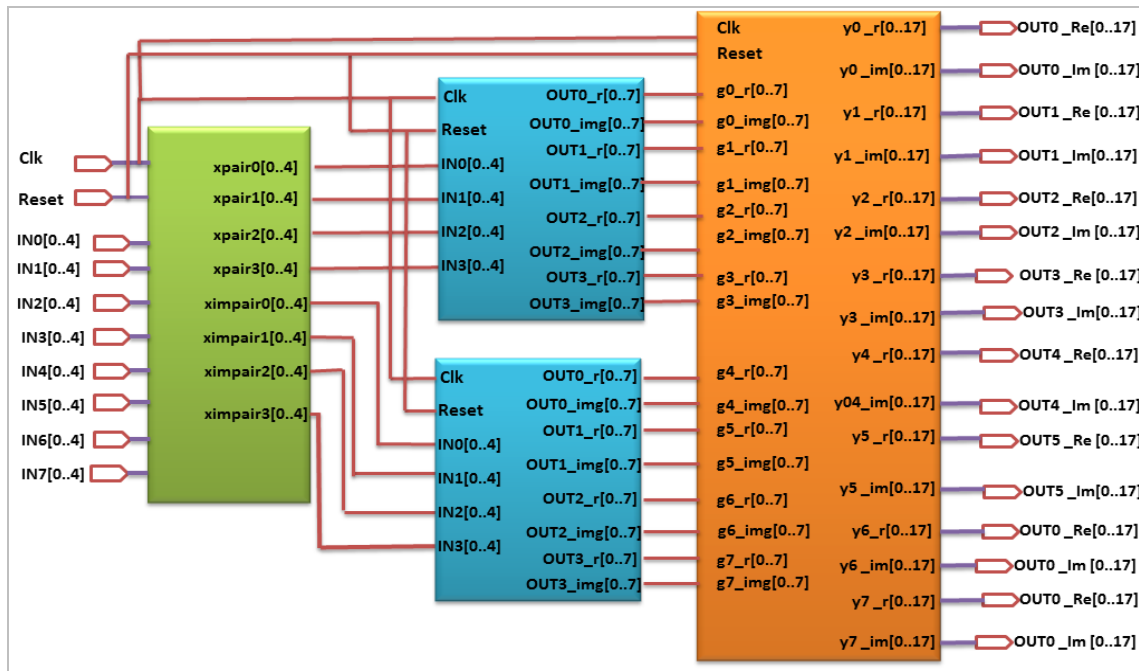


Figure 4.3. Schéma de principe du bloc 8 points FFT

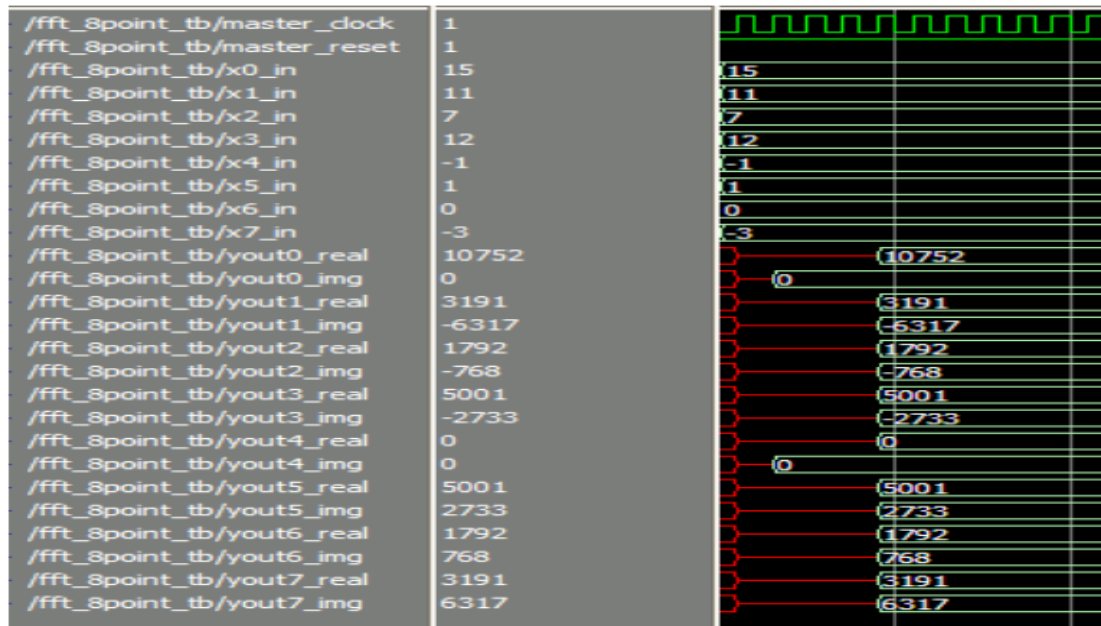


Figure 4.4. Résultat de simulation Modelsim pour un modèle de données.

Le résultat de la simulation Modelsim est illustré par la figure 4.4. Les résultats fournis par Matlab et ceux fournis par Modelsim (divisé par 28 pour obtenir le résultat correct) sont présentés dans le tableau 4.1

<b>Data in</b>	<b>15 11 7 12 -1 1 0 -3</b>
<b>Real_part_fft(matlab)</b>	<b>42.0000 12.4645 7.0000 19.5355 0 19.5355 7.0000 12.4645</b>
<b>img_part_fft(matlab)</b>	<b>0 -24.6777 -3.0000 -10.6777 0 10.6777 3.0000 24.6777</b>
<b>Real_part_fft (modelsim) normalisé</b>	<b>42.0000 12.4648 7.0000 19.5352 0 19.5352 7.0000 12.4648</b>
<b>img_part_fft (modelsim) normalisé</b>	<b>0 -24.6758 -3.0000 -10.6758 0 10.6758 3.0000 24.6758</b>

Tableau 4.1 Comparaison des résultats fournis par Matlab et ceux obtenus auprès Modelsim.

Les résultats de Matlab corroborent les résultats obtenus avec Modelsim et prouvent que le processeur FFT 8 points implémenté fonctionne correctement. La différence entre les deux échantillons de sortie ne dépasse jamais  $\pm 1$ . Cela est dû à l'opération de troncature car les divisions sont effectuées par des opérations de décalage à droite. Le processeur FFT 8 points implémenté a un débit de 4 horloges.

#### **4.4 .Synthèse matérielle pour FFT à 8 points**

Cette section présente les résultats de l'évaluation matérielle de l'architecture proposée pour la mise en œuvre d'un processeur FFT à 8 points. La conception de l'architecture proposée a été synthétisée sur un dispositif FPGA EP2C50F6726C6 basé sur Quartus II d'Altera de la famille Cyclone II. Les zones consommées par le registre, l'élément logique et les broches représentent respectivement 1%, 3% et 21% du total des ressources disponibles qui ont été mises en œuvre sur les programmes FPGA ont été écrites en VHDL.

#### **4.5. L'architecture proposée d 'FFT à base d'algorithme CORDIC**

L'algorithme de mode de rotation décrit dans le deuxième chapitre peut faire pivoter n'importe quel vecteur (pas seulement un vecteur unitaire aligné le long de l'axe) d'un angle compris entre  $-90^\circ$  et  $+90^\circ$ . Les décisions sur le sens de la rotation dépendent du fait qu'elles sont positives ou négatives. Le mode de fonctionnement vectoriel nécessite une légère modification de l'algorithme. Il commence par un vecteur dont l'une des coordonnées est

positive et l'autre coordonnée est arbitraire. Les rotations successives ont pour but de faire tourner le vecteur sur l'axe (et donc de réduire leur coordonnée à zéro). La valeur finale de  $x$  sera la magnitude du vecteur original mis à l'échelle par  $K$ . Ainsi, une utilisation évidente du mode vectoriel est la transformation des coordonnées rectangulaires en coordonnées polaires. Ne considérant que  $z$  comme angle d'entrée initial. À chaque étape,  $z$  tend à converger vers «0». Les étapes de l'algorithme sont comme suit :

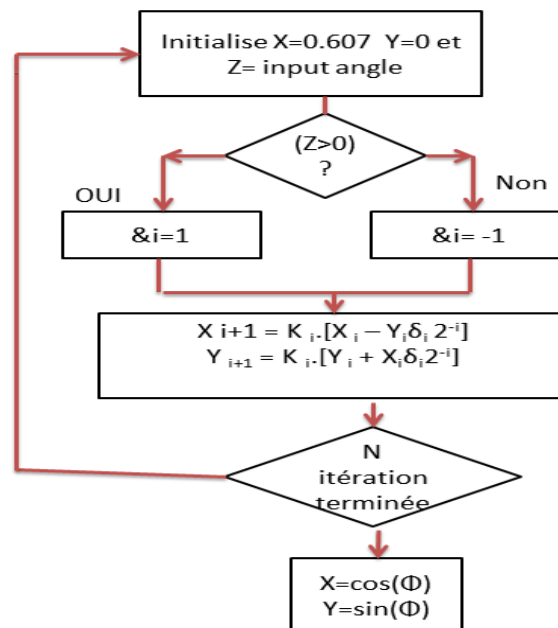


Figure 4.5 : Organigramme de l'algorithme CORDIC

Dans L'architecture proposée de l'algorithme CORDIC, 8 itérations continues sont effectuées et une représentation à 24 bits en virgule fixe est utilisée pour tous les registres. Initialement,  $x = 0,6073$  et  $y = 0$ . L'angle de rotation du vecteur à une itération particulière est fixe. À chaque itération, le vecteur pivote à l'angle le plus proche de l'angle requis. L'angle le plus proche est sélectionné parmi les angles disponibles. La table de consultation des opérations arctan a été introduite, avec une précision de 22 bits [61] comme indiqué ci-dessous.

$l$	$2^{-i} = \tan \alpha_i$	$\alpha_i = \arctan(2^{-i})$	$\alpha_i$ in radian
0	1	45°	0.7854
1	0.5	26.565°	0.4636
2	0.25	14.063°	0.2450
3	0.125	7.125°	0.1244
4	0.0625	3.576°	0.0624
5	0.03125	1.7876°	0.0312
6	0.015625	0.8938°	0.0156
7	0.007812	0.4469°	0.0078

Tableau 4.2 : Tableaux des angles

fig.4.6 Il montre les différents composants de l'algorithme CORDIC implémenté, comment ils sont interconnectés ainsi que les différents signaux de contrôle.

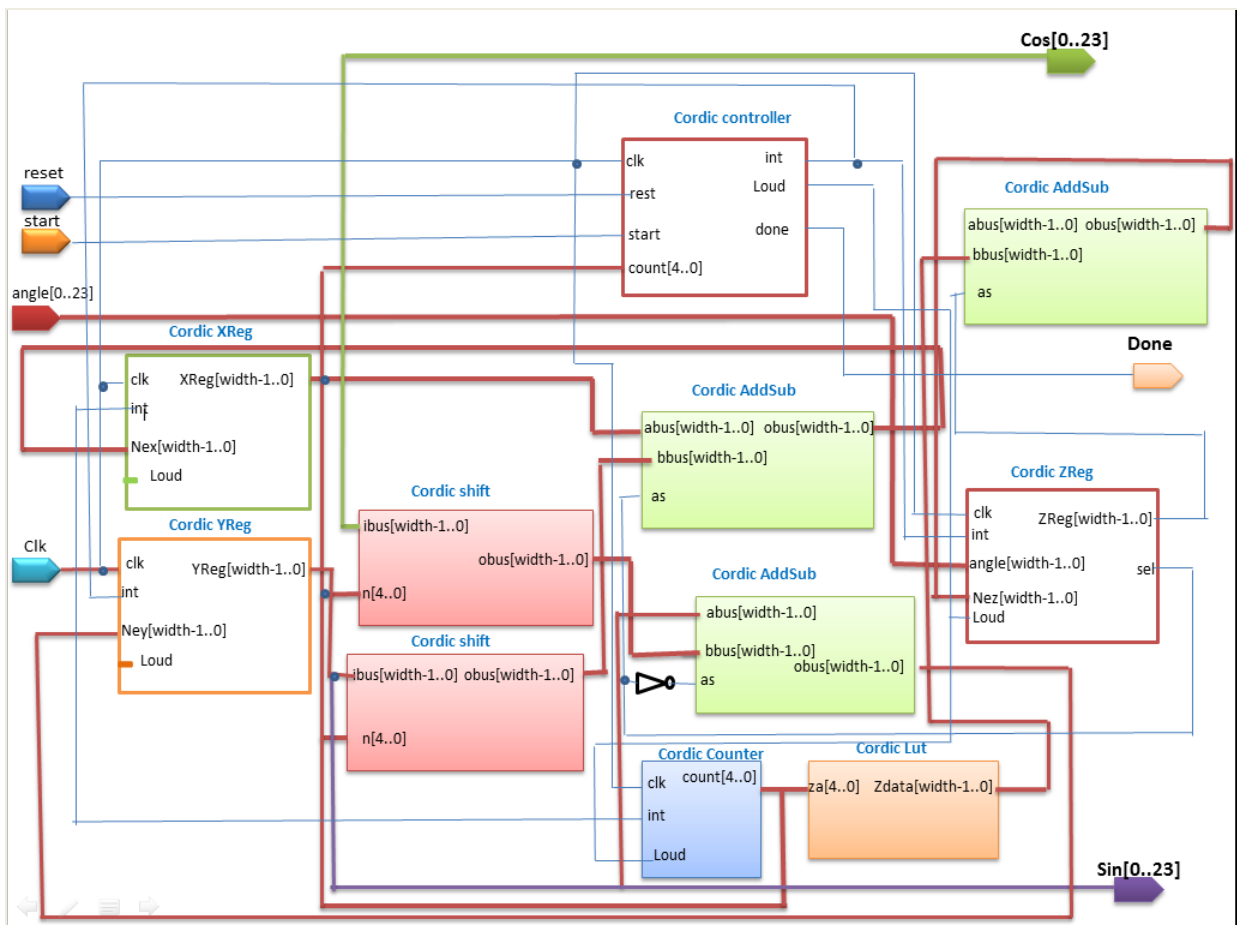


Figure 4.6: Module CORDIC

Le processus de mise en œuvre a été divisé en sept composants : contrôleur, register\_x, register\_y, register\_z, une mémoire morte (ROM) ou LUT, add\_subb et un shifn. Les valeurs des angles (rad) en représentation à virgule fixe sont répertoriées dans le tableau 4.2. Des précautions doivent être prises pour éviter tout débordement pouvant se produire dans l'additionneur. Pour cette raison, les angles qui sont supérieurs à  $2^{22}$ , doivent être écrits sur le complément à 2 en utilisant au moins un bit supplémentaire par rapport à la représentation des angles les plus grands [62].

Le contrôleur se compose d'un accumulateur de mise à l'échelle et d'un contrôleur de temps. Ce composant est également responsable de la génération du signal d'échantillonnage de l'algorithme CORDIC et de l'assertion du signal qui permet au circuit d'émettre le résultat final, le cosinus et le sinus d'angle si la valeur de compteur est égale à sept.

**Le contrôleur** est modélisé par la machine à états suivante (figure4.8).

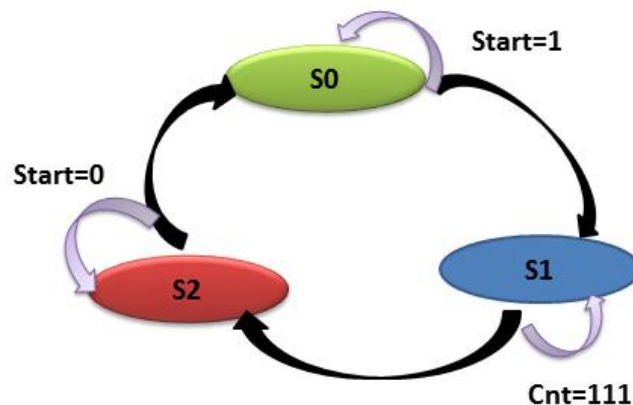


Figure 4.7: Machine d'états de contrôleur

**Register\_x** est initialisé à la valeur  $K_i = 9945$  ( $0,607 * 2^{14}$ ) après la précision. Ce composant donne la valeur de cosinus à la fin du traitement. La figure 4 est un diagramme de bloc généré par le RTL de QuartusII.

**Register\_y** est initialisé à la valeur (0) et donne la valeur du sinus à la fin du traitement.

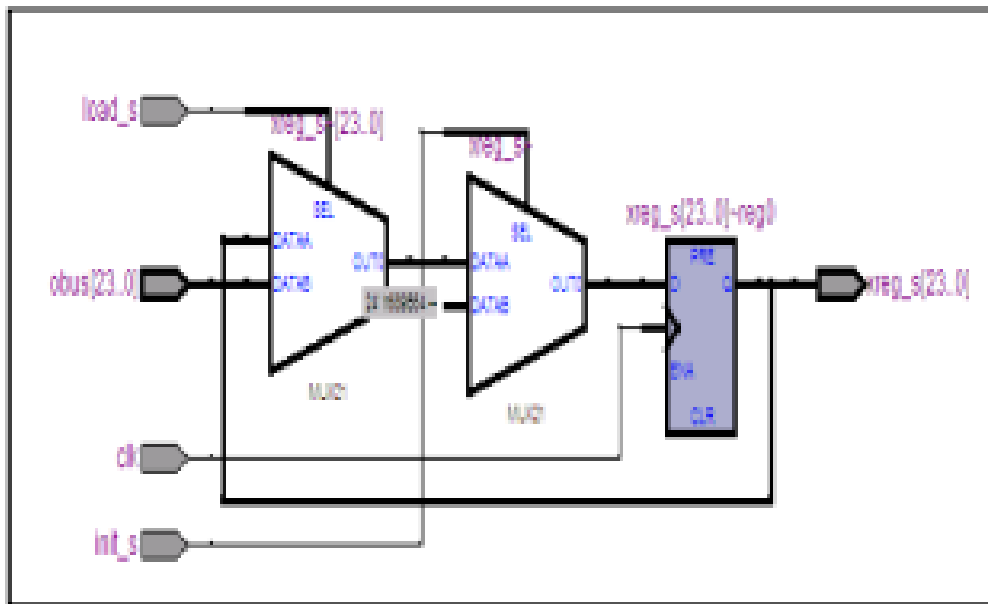


Figure 4.8: Diagramme de schéma de Registre \_X

Le rôle du bloc **ADD** est l'addition ou la soustraction en fonction de la valeur du signal de bit le plus significatif (Zreg) qui est le signal (as), si (as) = 0,  $\delta_i$  est égal à «1», sinon  $\delta_i$  est égal à «- 1» en tant que principe d'expression (2 .11).

Le rôle du bloc **Shiftn** est l'opération du décalage à droite ( $2^i$ ) .

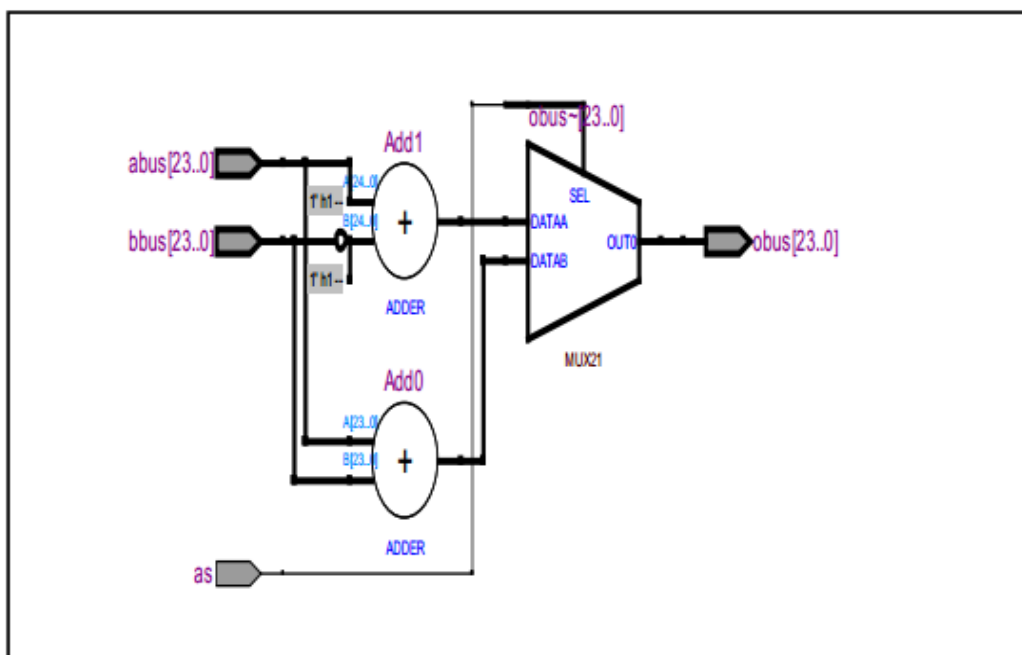


Figure 4.9: Diagramme de schéma de bloc ADD

Le délai d'obtention de la sortie (valeur sinus et cosinus) est fonction de la résolution choisie et du nombre d'itérations.

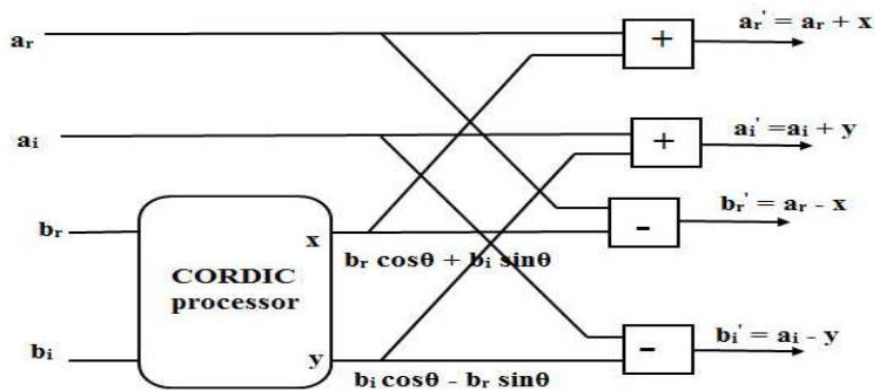


Figure 4.10: Module principal d'FFT à base d'algorithme CORDIC

Dans le module principal, les différents modules papillon nécessaires (basés sur la Figure. 4.10) ont été créés avec une indexation correcte de l'entrée et de la sortie.

L'implémentation FFT basée sur CORDIC et l'implémentation basée sur un multiplicateur complexe ont été réalisées séparément. L'architecture globale est la même pour les deux cas et la seule différence est que dans une implémentation basée sur un multiplicateur, le bloc CORDIC est remplacé par le multiplicateur.

#### 4.5.1. RÉSULTATS ET DISCUSSION

La description de haut niveau de la mise en œuvre a été écrite en langage VHDL. Les simulations ont été effectuées sur Modelsim et les résultats ont été comparés à ceux obtenus avec Matlab. Le modèle de données d'entrée (angles) choisi dans la plage  $[0, 6588397]$ .

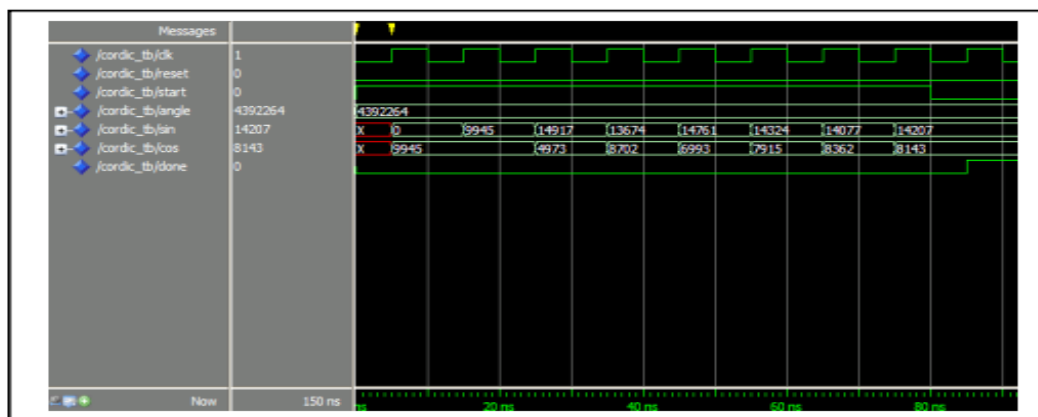


Figure 4.11: Résultat de Simulation Modelsim d'exemple d'angle ( $\Phi = 60^\circ$ )

<i>Angle deg</i>	<i>Angle rad*2<sup>22</sup></i>	<i>Cosine Result</i>	<i>Sine result</i>	<i>Cosine Normalize</i>	<i>Sine Normalize</i>
60°	4392264	8143	14207	0.5	0.86

TABLEAU 4.3 EXEMPLE D'ANGLE ( $\Phi = 60^\circ$ )

L'algorithme CORDIC implémenté a un débit de 8 cycles d'horloge. Dans l'exemple présenté dans la figure4.11, la simulation donne les résultats de cosinus et le sinus de l'angle 60°. L'angle est converti en radians et multiplié par 2<sup>22</sup> pour obtenir un résultat complet (4392264). À la fin de la simulation, le résultat est divisé par 2<sup>14</sup> pour obtenir le cosinus et le sinus corrects (tableau 4.3).

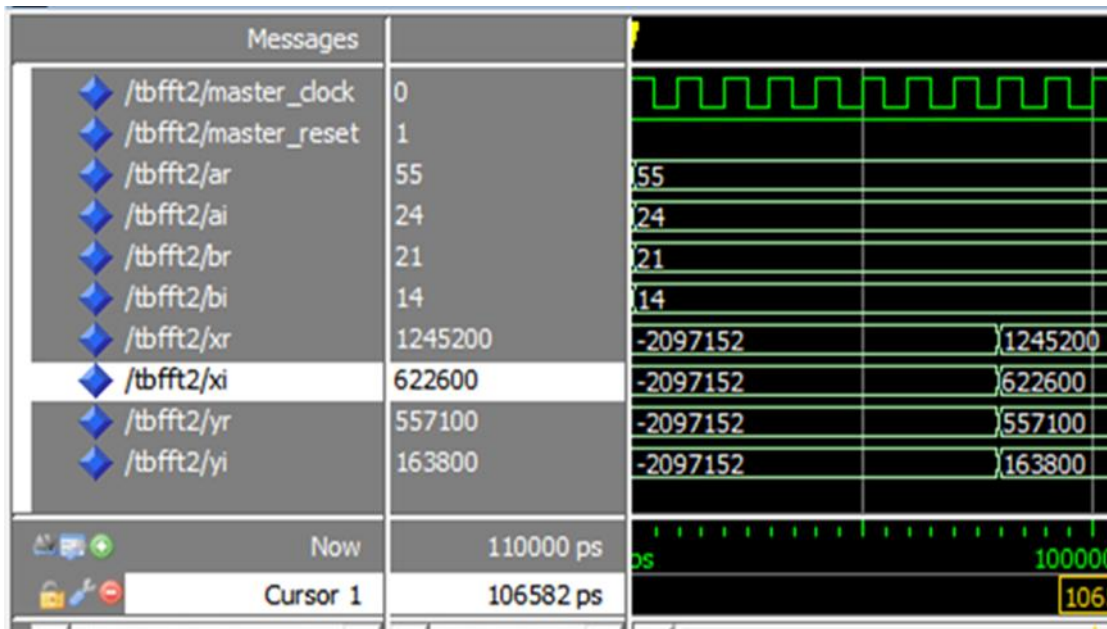


Figure 4.12 : FFT à 2 Points avec CORDIC

FFT IN		FFT OUT /2 <sup>14</sup>	
a=55+24i	b=21+14i	x=76+38i	y= 34+10i

Tableau 4.4 : Exemple d'FFT à 2 points

Le résultat de la simulation pour la FFT à 2 points avec CORDIC est présenté dans la Figure 4.12 Le résultat est obtenu après 8 cycles d'horloge, car l'algorithme CORDIC s'exécute en 8 cycles d'horloge.

La comparaison des différents paramètres tels que délai, multiplicateur, additionneur / soustracteur avec et sans CORDIC pour la FFT à 2 points est présentée dans le tableau 4.5. Le délai pour une FFT 2 points utilisant CORDIC est réduit par rapport au délai d'une FFT 2 points sans CORDIC .

Parameters	2-Point FFT sans CORDIC	2-Point FFT avec CORDIC
Delay	14.101 ns	10.231 ns
Multipliers	4	0
Adders/Subtractors	6(3 Adders ,3 Subtractors)	37(2 Adders ,2 Subtractors,33 Add/Sub)

Tableau 4.5 : Comparaison entre 2-Point FFT avec et sans CORDIC

#### **4.6. L'architecture proposée d 'FFT pour des taille premier à base d'algorithme Rader**

##### **4.6.1 L'algorithme Rader**

La séquence de transformée de Fourier discrète de N points est exprimée par:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \left[ \frac{nk}{N} \right]} \quad (4.1)$$

pour effectuer un changement adéquat, il faut utiliser la notation modulo d'entiers à N points [63].

$$k = g^k \bmod N \quad (4.2)$$

$$n = g^n \bmod N$$

$$g^n \in \mathbb{Z}_{p/\{0\}}$$

De plus, si N est un nombre premier, il produit une ou plusieurs racines génératrices primitives, appelées g.

Par conséquent, nous voulons calculer directement la fréquence comportementale de la séquence X (k) pour tout k. De plus, la valeur restante X (0), appelée composante continue, peut être calculée à l'aide d'un simple opérateur de sommation. D'autre part, on peut également observer que la valeur de l'échantillon x (0) n'est pas introduite directement dans le calcul de la convolution, et il est préférable de l'ajouter à la fin d'opération de sommation:

$$X(0) = \sum_{n=0}^{N-1} x(n).$$

### 4.6.2 Structure de filtre

Donc nous retrouvons avec la séquence DFT basée sur l'approche de Rader:  $\{X(k) - x(0)\}$ , pour  $k = 1, 2, \dots, N-1$ , qui pourrait être matérialisée par une structure de filtre appropriée, sur la base qu'il représente une expression de convolution cyclique [64] :

$$X(g^k \text{ mod } N) - x(0) = \sum_{n=0}^{N-2} x(g^n \text{ mod } n) e^{-j2\pi \left[ \frac{g^{(n+k) \text{ mod } N}}{N} \right]} \quad (4.3)$$

$$X(g^k \text{ mod } N) - x(0) = \sum_{n=0}^{N-2} x(g^n \text{ mod } N) w_N^{g^{(n+k) \text{ mod } N}} \quad (4.4)$$

Par exemple, lorsque  $N = 7$ , les racines primitives sont  $g = 3$  et  $5$ . Si  $3$  est choisi comme racine primitive candidate, son inverse est calculé comme suit:  $(3^1 \text{ mod } 7)^{-1}$  qui devient  $(3^{-1} \text{ mod } 7) = 5$ .

Pour  $k = 0, g = 3, N = 7$

$$X(3^0 \text{ mod } 7) - x(0) = \sum_{n=0}^5 x(3^n \text{ mod } 7) w_7^{3^{(0+n) \text{ mod } 7}} \quad (4.5)$$

$$X(1) - x(0) = x(1)w_7^1 + x(3)w_7^3 + x(2)w_7^2 + x(6)w_7^6 + x(4)w_7^4 + x(5)w_7^5$$

For  $k = 1$

$$X(3) - x(0) = x(1)w_7^3 + x(3)w_7^2 + x(2)w_7^6 + x(6)w_7^4 + x(4)w_7^5 + x(5)w_7^1$$

For  $k = 2$

$$X(2) - x(0) = x(1)w_7^2 + x(3)w_7^5 + x(2)w_7^4 + x(6)w_7^5 + x(4)w_7^1 + x(5)w_7^3$$

.

.

$$. X(5) - x(0) = x(1)w_7^5 + x(3)w_7^1 + x(2)w_7^3 + x(6)w_7^2 + x(4)w_7^6 + x(5)w_7^4$$

Dans la deuxième étape, il est très pratique d'utiliser la forme matricielle de l'opération de convolution cyclique, donc on obtient :  $X(k) - x(0) = \underline{W} \underline{x}$

Ce qui conduit à la représentation suivante:

$$\begin{bmatrix} X(1) \\ X(3) \\ X(2) \\ X(6) \\ X(4) \\ X(5) \end{bmatrix} - x(0) = \begin{bmatrix} w_N^1 & w_N^3 & w_N^2 & w_N^6 & w_N^4 & w_N^5 \\ w_N^3 & w_N^2 & w_N^6 & w_N^4 & w_N^5 & w_N^1 \\ w_N^2 & w_N^6 & w_N^4 & w_N^5 & w_N^1 & w_N^3 \\ w_N^6 & w_N^4 & w_N^5 & w_N^1 & w_N^3 & w_N^2 \\ w_N^4 & w_N^5 & w_N^1 & w_N^3 & w_N^2 & w_N^6 \\ w_N^5 & w_N^1 & w_N^3 & w_N^2 & w_N^6 & w_N^4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_2 \\ x_6 \\ x_4 \\ x_5 \end{bmatrix} \quad (4.6)$$

Les lignes et les colonnes sont permutées conformément au schéma de mappage indiqué dans le schéma fonctionnel de la figure 4.13, de sorte qu'il devient:  $X(k) - x(0) = [x]w$ .

$$\begin{bmatrix} X(1) \\ X(3) \\ X(2) \\ X(6) \\ X(4) \\ X(5) \end{bmatrix} - x(0) = \begin{bmatrix} x_1 & x_3 & x_2 & x_6 & x_4 & x_5 \\ x_5 & x_1 & x_3 & x_2 & x_6 & x_4 \\ x_4 & x_5 & x_1 & x_3 & x_2 & x_6 \\ x_6 & x_4 & x_5 & x_1 & x_3 & x_2 \\ x_2 & x_6 & x_4 & x_5 & x_1 & x_3 \\ x_3 & x_2 & x_6 & x_4 & x_5 & x_1 \end{bmatrix} \begin{bmatrix} w_N^1 \\ w_N^3 \\ w_N^2 \\ w_N^6 \\ w_N^4 \\ w_N^5 \end{bmatrix} \quad (4.7)$$

Lorsque vous sélectionnez une racine primitive et appelez un dispositif de réorganisation en utilisant l'équation (4.2), vous obtenez l'ensemble des index de sortie représentés sur la figure 4.14. Il existe donc un processus de mappage univoque des indices d'entrée  $n = 0, \dots, N-2$  pour générer les indices de sortie  $k = 1, \dots, N-1$ .

Il est primordial de souligner que cette matrice circulaire (4.7) possède la structure dite de Toeplitz. Généralement, cette notation matricielle est utilisée pour renforcer le niveau de mise en œuvre matérielle et réduire considérablement la complexité algorithmique [65] [66].

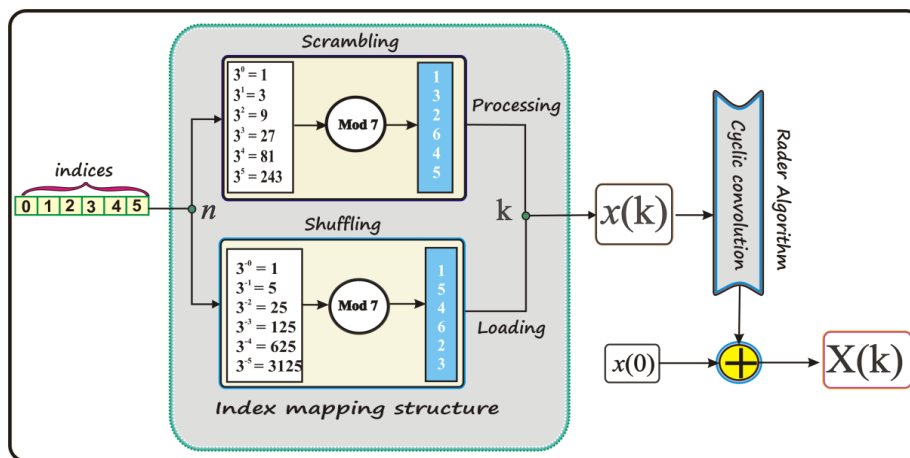


Figure 4.13: Diagramme fonctionnel de l'algorithme de Rader avec un processus de mappage basé sur des racines primitives.

### 4.6.3 Méthodologie d'implémentation de processeur DFT

Le processus de mise en œuvre a été divisé en trois composants: Rader\_Coeff, Rader\_filter et Rader\_controller, chacun a un rôle spécifique.

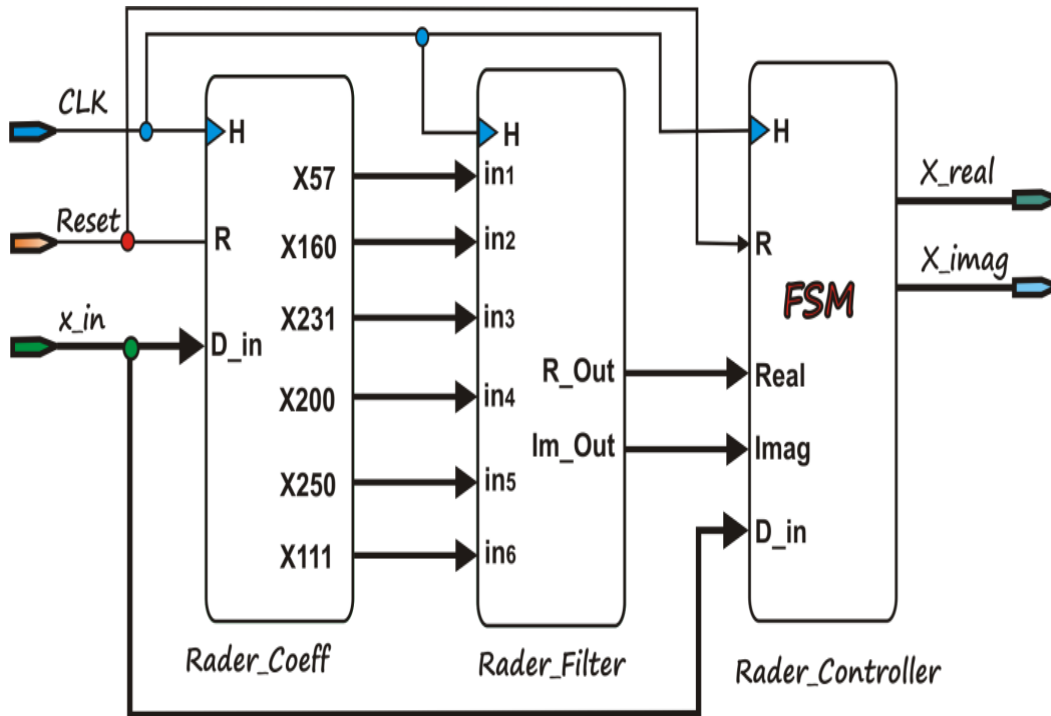


Figure 4.14: Conception holistique de l'algorithme Rader

La figure 4.14 illustre la conception suggérée de ce type de filtre. L'algorithme est utile pour de nombreuses applications dans le filtrage linéaire, l'égalisation de canaux et le multiplexage par répartition de fréquence orthogonale (OFDM) [65]. Selon l'équation (4.7), la structure adoptée de l'algorithme DFT basée sur le filtre FIR.

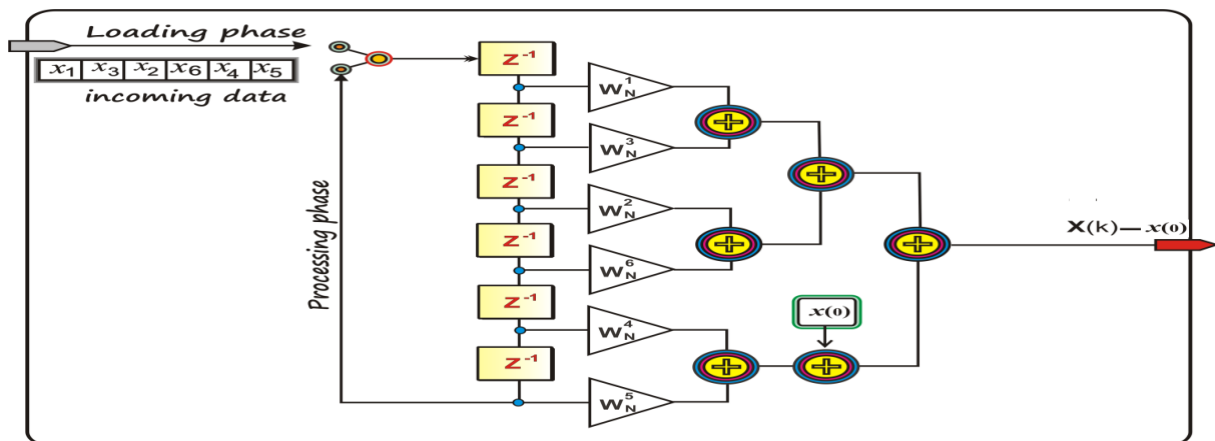


Figure 4.15: La structure du filtre FIR.

Avant de nous lancer dans la phase de mise en œuvre, il est important de rechercher un multiplicateur approprié en évitant de recourir à une architecture complexe, qui semble plus compliquée et coûteuse, particulier en termes de ressources matérielles [66]. Pour cette raison, la mise en œuvre de l'algorithme de Rader consiste à exprimer la structure transposée du filtre FIR en utilisant le multiplicateur de graphe d'addition réduit (RAG), ce qui semble plus attrayant que la méthode canonique signée (CSD) ou l'architecture arithmétique distribuée entièrement en pipeline. (DAA). La conception RAG est généralement considérée comme un simple multiplicateur basé sur l'additionneur, car les conceptions d'arithmétique distribuée (DA) sont traitées comme une configuration basée sur une table (LUT) [67] [68]. Ainsi, pour le large éventail de mises en œuvre en temps réel.

Le processeur TFD proposé est décrit dans le langage de description matérielle (VHDL) et synthétisé à l'aide de Xilinx ISE Design Suite 12.2 et Quartus II, ciblant les circuits FPGA Xilinx Virtex6 xc6vsx475t-2ff1156 et Altera Cyclone II, à des fins de comparaison.

Il est également possible d'implémenter l'intégralité de l'algorithme sous forme de circuit intégré à application spécifique (ASIC) ou de sous-programme d'exécution sur un processeur numérique standard (DSP) [69], [70].

Dans tout dispositif de traitement de signal numérique, la pièce maîtresse est l'architecture de son multiplicateur [71]. Il s'ensuit que la vitesse de ces dispositifs est intimement liée à la vitesse de leurs multiplicateurs. Par conséquent, il est nécessaire de quantifier les facteurs de torsion en tant que coefficients de filtre.

Supposons que les données d'entrée et les coefficients du filtre soient représentés par un nombre signé de 16 bits et de 8 bits, on peut noter que la charge de mise en œuvre de tous les coefficients basés sur la structure RAG nécessite environ 10 additionneurs pour satisfaire à l'exigence imposée par le paramètre. Diagramme des multiplicateurs de coefficients constants. Par conséquent, les coefficients quantifiés ou la représentation en virgule fixe sont répertoriés dans le tableau 4.6. Etant donné que l'algorithme de Rader est limité aux longueurs premières, il montre une grande symétrie dans la représentation des coefficients en virgule fixe.

$2^8 \times (W_7^k)$		
K	Real	Imaginary
1	160	-200
3	-231	-111
2	-57	-250
6	160	200
4	-231	111
5	-57	250

Tableau 4.6: coefficients quantifiés à 8 bits signé.

La mise en œuvre du filtre FIR circulaire avec sa forme transposée est illustrée dans le tableau 4.6. On peut voir dans ce tableau que l'effort pour les filtres CSD- Adder peut être calculé avec  $(M + N) * 2$ , où M représente la profondeur de bits des coefficients et N représente la longueur du filtre DFT. Ainsi, la structure RAG-Adder ne dépend que du facteur de résolution  $M / 2$ . Par exemple, lorsque  $M = 19$  bits, le facteur obtenu doit être égal à  $(19/2 \approx 10)$  additionneurs. De plus, les topologies RAG pour les filtres plus longs ne nécessitent qu'un seul additionneur supplémentaire pour chaque facteur, car le coefficient pré-synthétisé produit une grille dense de représentation de petits coefficients.

DFT Lengt h	RAG adder	CSD Adder	Auxiliary Factors
7	10	52	1,8,20,28
17	21	118	3,35,103,415,115 3,1249
31	36	206	3,9,133,797,877, 975
61	66	386	5,39,51,205,265, 3211
127	126	758	5,15,25

Tableau 4.7: Mise en œuvre du filtre FIR proposé.

Fondamentalement, la traduction appropriée d'un modèle mathématique décrivant l'évolution de l'algorithme à sa phase de mise en œuvre RTL représente un défi à surmonter.

La tâche de mise en œuvre nécessite l'utilisation d'une description structurée basée sur trois composants principaux, à savoir la multiplication complexe du facteur de torsion  $w_N^{mk}$ , le filtre proposé et le contrôleur qui coordonne et assure la synchronisation entre les différents modules. Le calcul des coefficients est la première étape du processus de conception. Les valeurs des coefficients sont calculées en utilisant le graphe additionneur réduit en tant que multiplicateur approprié à introduire à cette fin, comme illustré par le diagramme schématique de la figure 4.16.

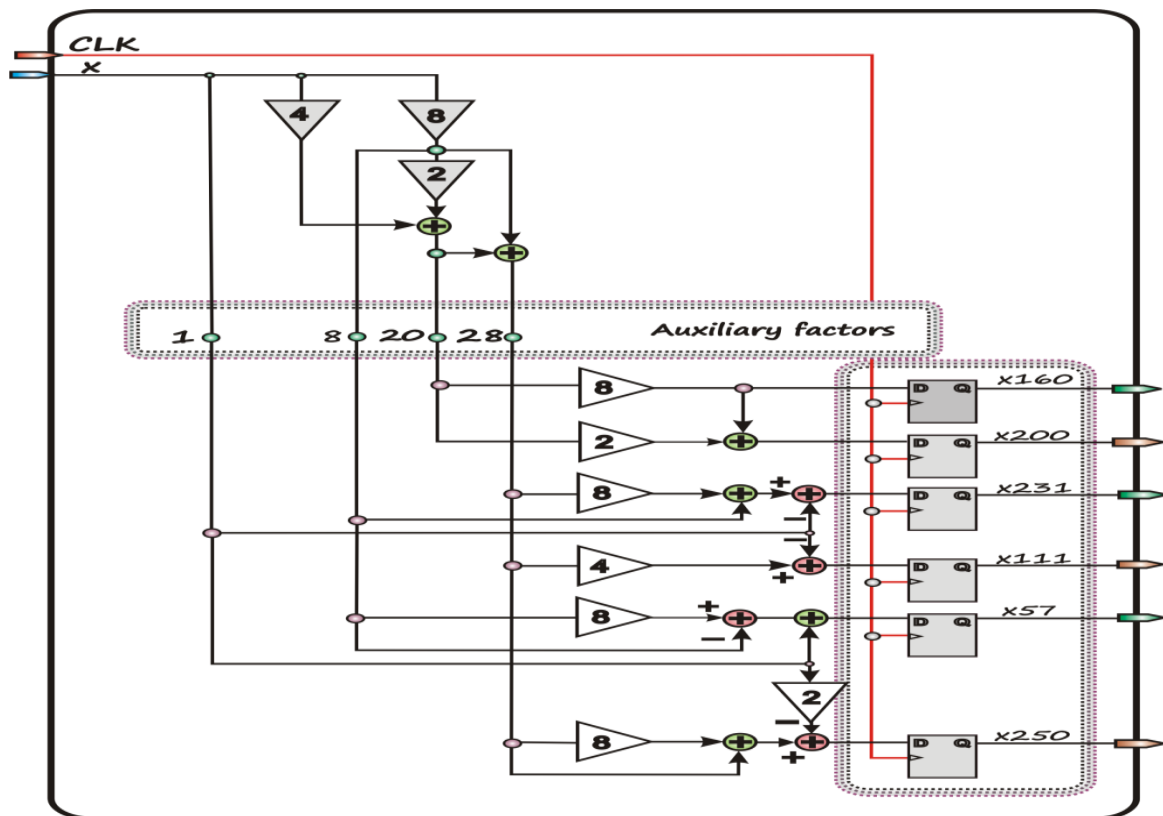


Figure.4.16 Le multiplicateur RAG des coefficients de filtre quantifiés.

Le second bloc est constitué de blocs de registres à décalage combinés avec un ensemble d'additionneurs afin de répondre aux besoins de ce type de filtre, ce qui permet de produire simultanément les valeurs imaginaire et réelle, comme illustré à la Figure 4.17.

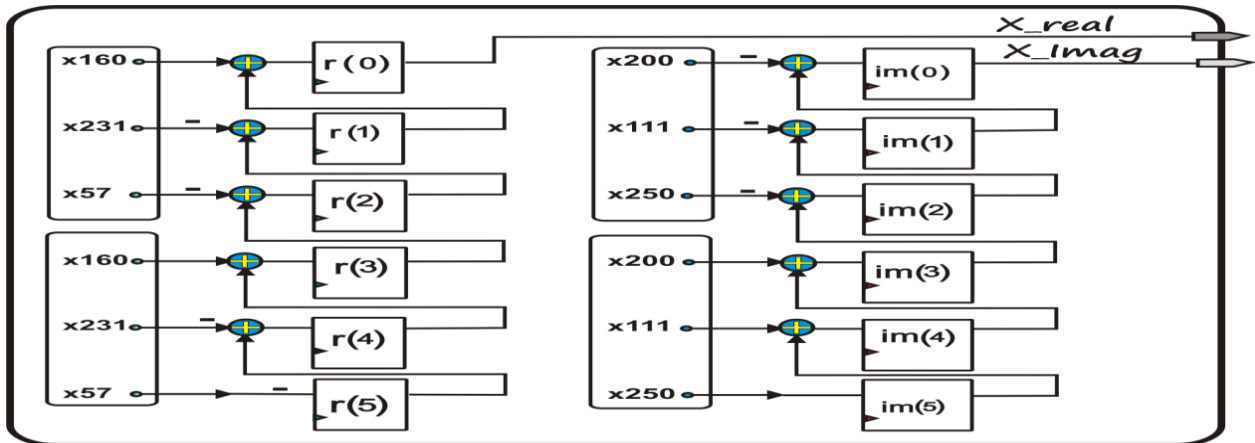


Figure. 4.17 Structure transposée du filtre proposé.

D'autre part, le mécanisme du contrôleur implique deux éléments pertinents: l'accumulateur de mise à l'échelle et le système du gestionnaire de synchronisation. Ce sous-circuit est également dédié à gérer correctement le temps de traitement adapté et valider les signaux activés. En pratique, ce composant est décrit par une machine à états finis, où les trois phases d'évolution, Idle, Load et Proc peuvent être facilement distinguées. Cet appareil nécessite donc certaines conditions spécifiques pour répondre à nos attentes. Le schéma fonctionnel de cette topologie est illustré à la figure 4.18.

Cnt	m <sub>2</sub>	m <sub>1</sub>	m <sub>0</sub>	x_out	CLK_out
1	0	X	0	x <sub>1</sub>	CLK_in
2	0	X	1	x <sub>1</sub>	0
3	0	X	0	x <sub>3</sub>	CLK_in
4	0	X	1	x <sub>3</sub>	0
5	1	1	X	x <sub>2</sub>	CLK_in
6	0	X	0	x <sub>6</sub>	CLK_in
7	1	1	X	x <sub>4</sub>	CLK_in
8	1	1	X	x <sub>5</sub>	CLK_in
9	1	0	X	x <sub>1</sub>	CLK_in
10	0	X	1	x <sub>3</sub>	CLK_in
11	0	X	1	x <sub>2</sub>	CLK_in
12	0	X	1	x <sub>6</sub>	CLK_in
13	0	X	1	x <sub>4</sub>	CLK_in
14	0	X	1	x <sub>5</sub>	CLK_in
15			Done		

Tableau 4.8 Chronogramme de l'opération de mappage d'index.

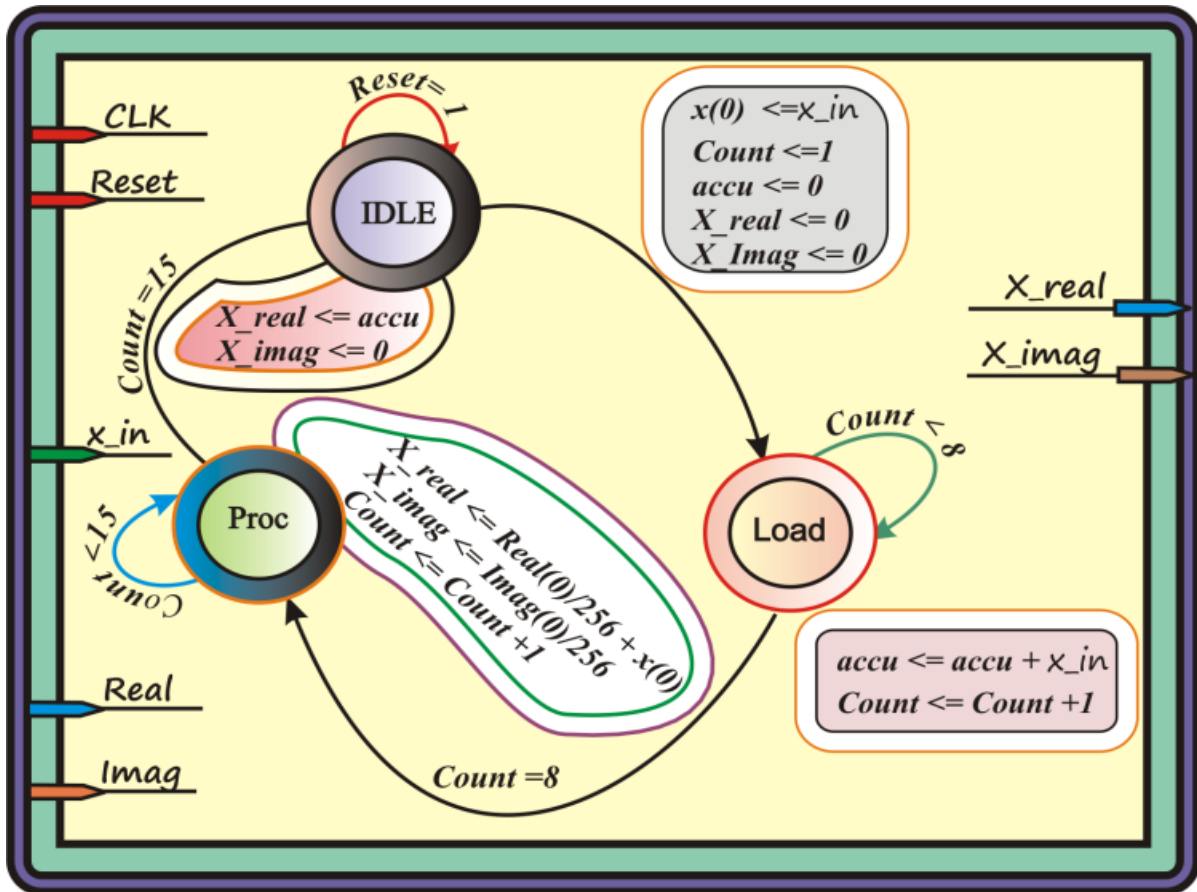


Figure 4.18: Machine à états finis du contrôleur.

La figure 4.20 montre la mise en œuvre des deux étapes respectivement appelées opérations de brouillage et de brassage qui serviront de transformation non linéaire selon la conception représentée dans la partie gauche de la figure 4.13.

D'un point de vue pratique, cette interface, qui constitue une étape de prétraitement de la séquence d'entrée, devrait être placée avant l'algorithme principal afin de faciliter sa mise en œuvre et de répondre aux exigences imposées par le choix de cette approche. En outre, le mécanisme permettant d'effectuer cette opération de mappage d'index, qui a été clarifié par le chronogramme, semble plus explicite, comme indiqué dans le tableau 4.8. Il est tout à fait approprié de veiller à l'application de la méthode de chargement et de traitement. Par conséquent, le jeu de données est présenté sous la forme  $\{x(0), x(5), x(4), x(6), x(2), x(3), x(1)\}$  Même séquence sauf échantillon  $x(0)$ .

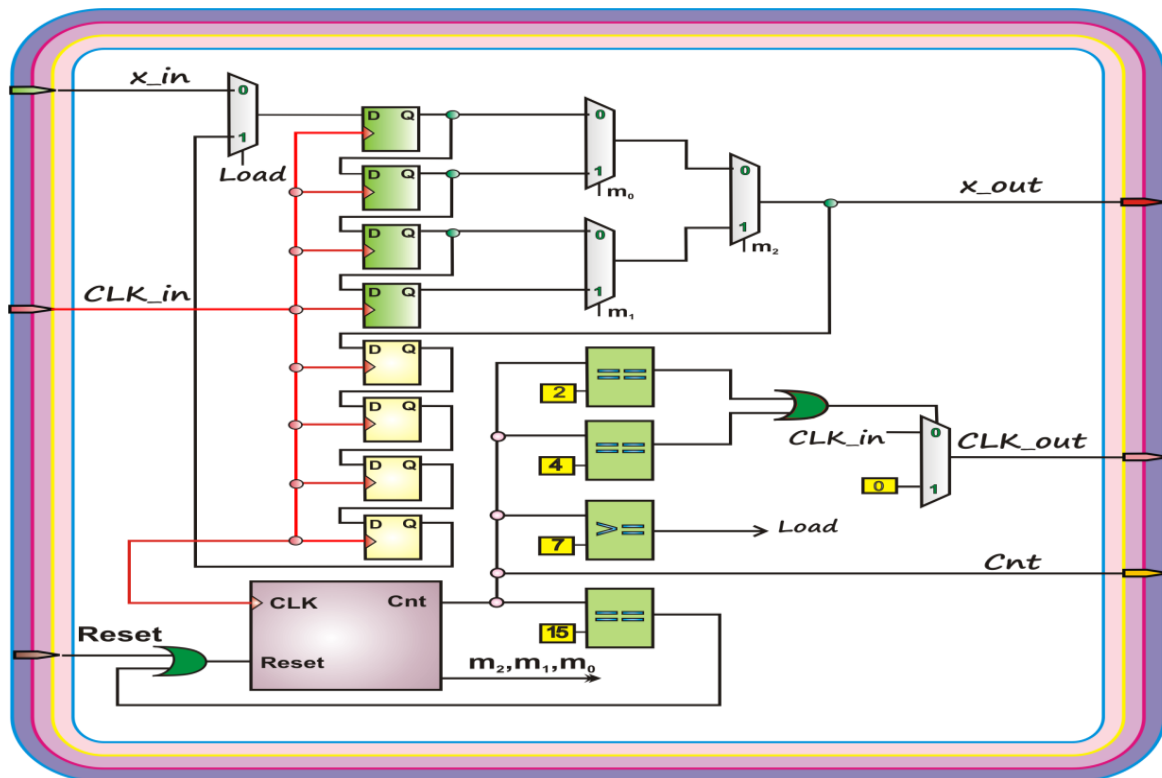


Figure 4.19: Schéma de mise en œuvre de la cartographie d'index

#### 4.6.4. Results and discussion

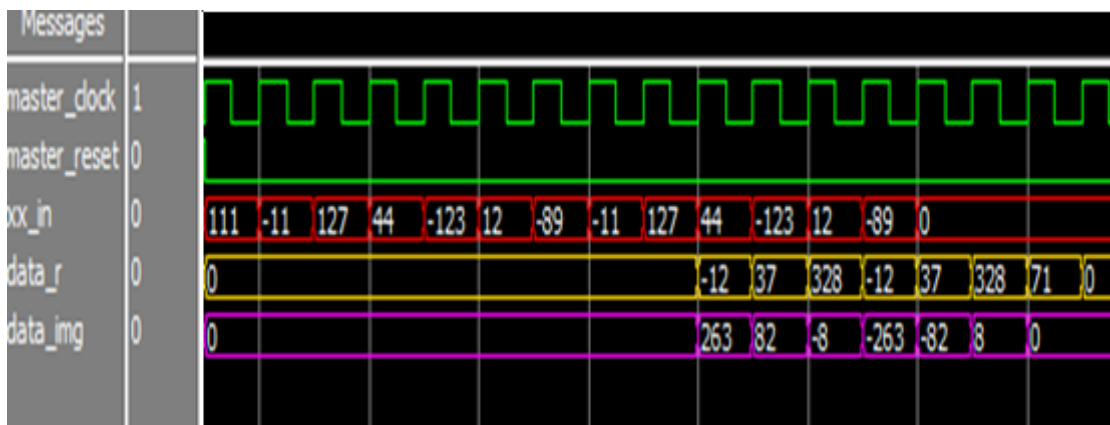


Figure 4.20: Chronogramme de simulation de 7 point DFT

Le bon fonctionnement de la conception proposée a été testé en utilisant l'environnement Modelsim et les résultats ont été comparés à ceux donnés par Matlab. Fondamentalement, la séquence de données entrante sera traitée en série et supposée varier dans la plage  $[-32768, 32767]$  avec une résolution de 16 bits par échantillon. Par conséquent, tous les coefficients sont traités en utilisant six additionneurs et quatre soustracteurs. Les séquences de sortie sont

quantifiées sur 27 bits et apparaissent dans un ordre permuté. Exactement à 16 cycles d'horloge, la dernière valeur X (0) est transmise à la sortie et le circuit est prêt à traiter le flux de données suivant.

Cette section présente une évaluation des performances de l'architecture proposée pour les tailles 7, 17, 31 et 67 DFT. Les résultats montrent qu'il y a toujours un compromis à établir entre différentes mesures. Les tableaux 4, 5 et 6 répertorient toutes les ressources nécessaires pour construire correctement l'algorithme adopté illustré à la figure 4.14 et comparent les résultats à ceux obtenus avec différentes architectures. On peut donc facilement affirmer que l'algorithme de Rader est encore moins complexe que les autres méthodes. De plus, il permet d'obtenir des circuits plus rapides et une consommation d'énergie réduite.

	<b>[72]</b>		<b>Proposed approach</b>	
<b>DFT Length</b>	N=16	N=32	N=17	N=31
<b>Slices</b>	2402	5620	1434	1911
<b>Fully used Lut -FF pairs</b>	295	545	1391	1854
<b>DSP 48E1'S</b>	51	99	0	28
<b>Bonded IOB's</b>	128	320	76	78
<b>MOF (MHz)</b>	25	19	148.169	121.45
<b>Latency (ns)</b>	39	52.3	6.75	8.2
<b>Power consumption (mW) at 100 MHz</b>	-	-	14	16

Tableau 4.9: Comparaison des performances (Xilinx Virtex 6).

Dans le tableau 4.9, les longueurs DFT sont différentes, mais 17 et 31 ont été sélectionnés comme nombres premiers les plus proches, à 16 et 32, respectivement, tout en utilisant la même longueur de données (16 bits). Les résultats montrent que l'architecture proposée basée sur l'algorithme de Rader utilise une zone plus petite, moins de DSP et est plus rapide que l'architecture Radix-2 utilisée dans [72]. Dans le tableau 4.9, la latence a été calculée comme dans [72] et est égale à 1/MOF, MOF étant la fréquence de fonctionnement maximale.

**CHAPITRE4 : Méthodologies d'implémentation d'FFT et discussion des résultats**

N	[73] SPARTAN3		[74] virtex4		Proposed approach virtex6	
	MOF (MHz)	L	MOF (MHz)	L	MOF (MHz)	L
7	50.9	-	-	-	233.33	-
17	40	-	-	-	148.17	-
31	25	-	-	-	121.45	-
64	-	-	80	192 clks 1 data/3clks	-	-
67	-	-	-	-	92.5	1 98clks 1data/2. 95clks

Tableau 4.10 Comparaison des performances de la mise en œuvre sur FPGA Xilinx

Le tableau 4.10 compare la fréquence maximale de fonctionnement (MOF) d'œuvres similaires à des implémentations ciblant différents circuits FPGA. D'une part, cela montre que le MOF décroît à mesure que la longueur de la TFD augmente et, d'autre part, cela met en évidence la supériorité de l'architecture proposée.

	DFT Length	Data length (bits)	MOF (MHz)	Logic elements	Throughput (Gbps)
[73]	8	16	12.56	16102	-
[74]	8	4	162.7	52	10.41
[74]	16	7	94.76	2548	21.23
<b>Proposed approach</b>	7	16	127.3	726	24.06
<b>Proposed approach</b>	17	16	90	1898	44.37

Tableau 4.11 Comparaison des performances (Altera Cyclone II).

Enfin, le tableau 4.11 consolide la conclusion tirée des tableaux 4.10 et 4.11 et révèle les améliorations apportées à la structure proposée par rapport aux résultats publiés dans [73] et [74] en termes de ressources matérielles, de fréquence maximale et de débit. Cette dernière métrique a été évaluée selon la formule utilisée dans [74].

En fait, la latence nécessaire pour effectuer cette transformation et produire des données fiables est intimement liée à la longueur de la TFD. Évidemment, lorsqu'un horizon de 7 points est satisfaisant, il faut neuf cycles pour produire les données attendues, comme indiqué dans le tableau 4.12. Ces valeurs représentent le temps de latence nécessaire pour produire la première sortie lorsque toutes les entrées sont disponibles. Il s'exprime en cycles d'horloge et non en temps, car un circuit peut fonctionner à une vitesse différente de celle du MOF. Les sorties suivantes apparaissent à une cadence d'une sortie par cycle d'horloge.

Du point de vue de la complexité du calcul, il est facile de constater que l'algorithme Rader fonctionne plus efficacement que les deux autres versions des approches CZT ou Winograd, comme illustré dans le tableau 4.12, mais cela n'est vrai que lorsque la longueur de la TFD est limitée à des tailles courtes. Cette prérogative favorise grandement les techniques dédiées à la transmission numérique telles que la composante OFDM ainsi que la radiodiffusion.

<b>DFT length</b>	<b>7</b>	<b>17</b>	<b>31</b>	<b>67</b>	<b>127</b>
<b>Rader algorithm</b>	9	35	41	131	233
<b>Winograd algorithm</b>	11	49	53	105	205
<b>CZT algorithm</b>	17	67	121	175	305

Tableau 4.12 Performances de latence pour différents algorithmes

#### **4.7 Conclusion :**

Ce dernier chapitre a été entièrement consacré à la présentation de l'implémentation de la transformée de Fourier discrète rapide, plusieurs propositions ont été discutées. Dans la première partie, une méthodologie basée sur la technique de radix2 pour implémenter 8points FFT parallèle a été détaillée et les résultats ont été présentés et discutés. La multiplication avec le facteur de torsion est la principale tâche critique du calcul de la FFT. Ont été présenté une conception et implémentation d'algorithme CORDIC utilisé pour implémenter l'FFT. Par la suite, une étude comparative entre l' FFT à 2 points avec et sans CORDIC. La structure d'FFT de 2points avec CORDIC est plus rapides que celles de la FFT de 2points sans CORDIC et offrent un bon compromis entre performances et coût (multiplicateur, additionneur / soustracteur).

La deuxième partie de ce chapitre ont été consacrée à la contribution effective de cette thèse. En effet, intéressé à l'implémentation de la transformée de Fourier discrète rapide dont la dimension de la séquence n'est pas une puissance de 2. L'algorithme matérialisé et analysé dans ce cadre, est celui de Rader utilisant une structure de type filtre FIR et traitant des séquences d'entrée de taille exprimée en nombres premiers. L'architecture proposé composée de 3 sous-systèmes fondamentaux, en l'occurrence : Un bloc de calcul des coefficients ainsi qu'un bloc réalisant le filtrage à proprement dit et l'ensemble étant piloté par un contrôleur a machine synchrone à états finis. Le processus peut, bien entendu, être retiré pour passer à un nombre de niveaux supérieur. L'architecture proposé est décrit dans le langage de description matérielle (VHDL) et synthétisé à l'aide de Xilinx ISE Design Suite 12.2 et Quartus II, ciblant les circuits FPGA Xilinx Virtex6 xc6vsx475t-2ff1156 et Altera Cyclone. Notre méthodologie d'implémentation beaucoup plus simples, beaucoup plus rapides que celles des travaux [72],[73],[74] publiées et offrent un bon compromis entre performances et coût (surface et latence).

#### 4.8 Référence de chapitre 4

- [61] Aouissi S, Mohamed B, Hichem Sand Saliha H. “FPGA Implementation of Sine and Cosine Value using Cordic Algorithm”,SIVA13,guelma,2013
- [62] G.Gopikiran”FPGA Implementation of Floatingpoint Rotation Mode CORDIC Algorithm”, ECE Department National Institute of Technology,International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011).
- [63] Rader, C. ‘Discrete Fourier transform when the number of data samples is prime’, Proc. IEEE, 1968, **56**, pp. 1107–1108
- [64] Aouissi S , Benouaret M ,”Hardware Implementation Of The Fast Fourier Transform With Non-Power-Of-Two Problem Size”, The International Conference on Automatic control, Telecommunication and Signals (ICATS’15),Annaba 2015.
- [65] Aouissi Samiha , Benouaret Mohamed, Harize Saliha”DFT processor implementation scheme based on Rader algorithm” IET Circuits, Devices & Systems, 2019.
- [66] Marc J, Pascal P, Serge V.: 'Efficient Generation of Prime Numbers', Cryptographic Hardware and Embedded Systems – CHES , Springer-Verlag, 2000, pp. 340–354
- [67] Saliha Harize, Mohamed Benouaret ,“A Simple and Efficient Scheme to Implement FIR Filters on FPGA” ,badji mokhtar university International Conference in khanechela ICEECA2012.
- [68] M. D. van Burgwal,P. T. Wolkotte, G. J. M. Smit.: 'Non power-of-two FFTs: Exploring the flexibility of the Montium TP',International Journal of Reconfigurable Computing. 2009, pp. 1-12
- [69] Javier Valls, Vicenç Almenar. : 'FPGA implementation of an OFDM-based WLAN receiver', ELSEVIER, Microprocessors & Microsystems ,2012, pp. 232-244
- [70] W.Zheng,K.Li. :'Split-radix algorithm for length  $6^m$  DFT'. IEEE Signal Process. 2013,pp. 713–716
- [71] S.G. Johnson, M. Frigo.: 'A modified split-radix FFT with fewer arithmetic operations',IEEE Trans. Signal Process. 2007,pp.111–119
- [72] [S. Josue S](#), [Juan J](#), [Edwin C](#) et al.: 'FPGA Design and Implementation of Radix-2 Fast Fourier Transform Algorithm with 16 and 32 Points', 2015 IEEE International Autumn Meeting on Power, Electronics and Computing , Ixtapa, Mexico ,2015,pp. 1 - 6
- [73] Arman C, Yousef S, Otto S et al.: 'Implementing FFT Algorithms on FPGA', International Journal of Computer Science and Network Security,2011, pp. 148-156

[74] Jameel A, Waseem I, Muhammad A.: ' Implementation and Performance Evaluation of Three Reconfigurable FFT Cores for Application in Software Defined Radio System'. [International Conference on Electrical Engineering \(IEEE\)](#), Lahore, Pakistan,2017,pp.1-6

# **Conclusion et perspectives**

## **Conclusion et perspectives :**

Les communications numériques d'aujourd'hui ainsi que les communications futures doivent répondre à des contraintes. En effet, il est nécessaire de travailler en temps réel tout en garantissant une très bonne qualité de transmission. Cette thèse présente les aspects théoriques et pratiques de la mise en œuvre de la modulation OFDM et notre contribution de la thèse porte sur le bloc FFT. Notre méthodologie d'implémentation d'un algorithme DFT séquentiel basé sur l'algorithme de Rader, qui intègre l'idée d'un filtre spécifique de type FIR pouvant être implémenté sur des cibles FPGA.

L'analyse du schéma de calcul qui en découle a permis de réduire la complexité globale du bloc FFT en effectuant un choix judicieux du multiplicateur numérique approprié. La structure du multiplicateur est considérée comme étant l'élément clé du système, en termes de complexité d'implémentation, de latence et de consommation des ressources matérielles. L'architecture proposée s'est avérée meilleure en taille et en vitesse sans avoir besoin de multiplicateurs intégrés.

Il est important de souligner que la consommation de ressources matérielles augmente avec la longueur DFT et la profondeur des bits de données.

En général, l'implémentation proposée peut être étendue pour fournir une variété d'autres réalisations de tailles FFT de haute performance. Cette architecture est destinée à être un noyau spécifique pour le développement de techniques plus sophistiquées utilisées dans le traitement du signal comme le multiplexage codé par répartition orthogonale de la fréquence (COFDM) ou le système de cryptage numérique. Par conséquent, cette conception vise à la mise en œuvre de cœurs de processeur DFT et à souligner le fait qu'une cible FPGA est à la fois flexible et offre de meilleures performances que les appareils classiques.

Notre conception peut être améliorée pour donner de meilleures performances.

La conception peut être optimisée pour la contrainte de synchronisation et le retard du signal. Cette conception peut être modifiée pour accepter les nombres complexes. Une implémentation de circuits numériques supplémentaires peut être réalisée de telle sorte que cette conception puisse accepter des nombres complexes comme entrées.