République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Dr YAHIA FARES de MEDEA



Faculté de la Technologie

Département de Génie Electrique

Mémoire de fin d'études pour l'obtention du diplôme de Master en Génie électrique

Option : Systèmes de Télécommunication

THEME :

Implémentation sur FPGA d'une architecture FFT Parallèle pour les systèmes de télécommunication MIMO-OFDM

Proposé et dirigé par :

Mr. DALI Mohamed

Etudié par:

Mr. BEN HAMANI Bachir Mouin

Mr. LAHLOUB Akram

Promotion 2016/2017

Remerciements

Nous tenons en premier lieu à remercier « Dieu le tout puissant » de nous avoir donné le courage de surmonter tous les problèmes, la santé et la patience pour mener à bien ce modeste travail.

Ensuite, nous remercions nos parents qui nous ont soutenus beaucoup pendant toute la vie et qui continuerons à nous aider dans tous les projets de l'avenir.

Nos plus sincères remerciements à notre promoteur Mr. DALI Mohamed pour son précieux temps qu'il a consacré à notre écoute, à notre orientation et correction tout au long de notre projet.

Nous tenons à remercier : le président et les membres de jury pour avoir accepter de juger notre travail.

Enfin, nous remercions tous ceux qui ont contribué de loin ou de prés à la réalisation de ce modeste travail.

DEDICACES

Je dédie ce mémoire, le fruit de mon travail durant tous les années de mes étude à:

A mon père et ma mère que dieu les protège, qui mon offert tous les moyens,

Ainsi leurs encouragements avec lesquelles j'ai pu atteindre ce niveau.

A mes sœurs Romaissa, Hiba et Maroua.

A mon neveu Razin.

A mon ZIYADE.

A tous mes amis (es), touts ma famille (BEN HAMANI et YAHAOUI)

A tous ceux qui ont contribué de loin ou de prés a la réalisation de ce mémoire.

Enfin, a tous ceux qui m'aiment.

A vous.....

BEN HAMANI Bachir Mouine

DEDICACES

Je dédie ce mémoire, le fruit de mon travail durant tous les années de mes étude à:

A mon père et ma mère que dieu les protège, qui mon offert tous les moyens,

Ainsi leurs encouragements avec lesquelles j'ai pu atteindre ce niveau.

A mon frère Youcef

A ma sœur Amani

A tous mes amis (es), touts ma famille (LAHLOUB et BEN MOUSSA).

A tous ceux qui ont contribué de loin ou de prés a la réalisation de ce mémoire.

Enfin, a tous ceux qui m'aiment.

A vous.....

LAHLOUB Akram

Liste des abréviations…….....xi

Notations……...xii

Liste des figures …….....xiv

Liste des tableaux……...xv

Introduction générale…….……1

Chapitre I : Les techniques de communications

I)- Introduction ……...4

II)- Les chaînes de transmission Numériques ……...4

II.1)- La chaîne d'émission …….5

II.1.1)- La source d'information ……5

II.1.2)- Le codeur de source ……...5

II.1.3)- Le codeur de canal ……6

II.1.4)- Le Modulateur ……...6

II.2)- La chaîne de réception……..6

II.2.1)- Le démodulateur ……6

II.2.2)- Le décodeur de canal ……...7

II.2.3)- Le décodeur source ……..7

II.2.4)- Le destinataire …..……....7

II.2.5)- Le canal de transmission ……..7

III)- La Technique OFDM ……...…9

III.1)- Avantages et inconvénients de l'OFDM ……....10

IV)- Les Système MIMO ……..11

V)- Système MIMO-OFDM ……..13

IV)- Conclusion ……15

Chapitre II : L'FFT : Algorithme et Architecture

I)- Introduction ……17

II) -Concepts de base ……..18

III)- Algorithmes FFT …….…… 18

III.1)- Radix 2……19

III.1.1) –l'algorithme FFT Radix-2 DIT ……...20

III.1.2)- Décimation fréquentielle (DIF) ……22
III.2) – Radix-4 …… ……25
III.2.1)- Décimation temporelle(DIT) ……..……25
III.2.2) – Décimation fréquentielle (DIF) ……26
III.3)- Bit reversal ……29
IV)- Architectures d'implémentation de L'FFT en pipeline …… 29
IV.1)- Architecture Radix-r MDC …….. 30
IV.2) Architecture Radix-r(SDF) 31
IV.3) –Architecture Radix-r SDC …… 32
V)-Comparaison …….. 32
VI)-Conclusion ……33

Chapitre III : Généralité sue les circuits FPGA

I)- Introduction ……35
II)-Architecture du FPGA ……..35
II.1)-Les CLB (Configurable Logic Block) ……...…….37
II.2)-Les cellules d'entrée sortie (IOB) ……..37
II.2.1)-Configuration en entrée ……37
II.2.2)-Configuration en sortie ……37
II.3)-Les Interconxions ……38
III)- les avantages est les applications de FPGA ……38
IV)-Les outils de dévloppement…… ….41
IV.1)-Loutil de conceptino ISE 14.3 ……..41
VI.2)-System Generator ……..41
V)- Application des FPGA ……42
VI)-Les étapes d'implémentation ……43
VI.1)-Spécification de design et développement de design…….43
VI.2)- Simulation(FPGA) ……43
VI.3)-Synthèse et implémentation du projet ……...44
VII)-Conclusion ……..45

Chapitre IV : Conception et implémentation

I)-Introduction ……..46
II)- L'architecture implémentée ……..46
III)-l'organigramme de développement notre projet ……..49
III.1)- description de l'organigramme ……...50
a)- Génération des entrées par Matlab ……..50
b)- Conception du schéma ……..50
c)-Calcule de la DFT par le processeur FFT conçu en utilisant « système generator » ……51
d)- Calcul de la FFT par MATLAB ……...……53
e)- Comparaison ……...…..53
f)-modification ……...……...53
g) -Calcul du rapport Signal sur Erreur de quantification : (SQNR) ...……...53
h)-génération du modèle HDL équivalent ..……..…...……54
i)- Implémentation ……...……...……55
j)- Analyse temporal & détermination de la fréquence maximale ……56
k)-Estimation de la puissance dynamique ……57
IV)-Présentation des résultats de simulation et d'implémentation ……..57
V) -Conclusion ……..60

Liste des figures

Chapitre I:Les techniques de communications

Figure I.1 : *Schéma synoptique d'une chaîne de transmission numérique* ***5***

Figure I.2 : *Schéma bloc d`un système OFDM* ***10***

Figure I.3 : *Schéma d`un système MIMO* ***12***

Figure I.4 : *Schema simplifie d`un système MIMO-OFDM* ***14***

Chapitre II : L'FFT : Algorithme et Architecture

Figure II.1: *Structure DIT butterfly* ***21***

Figure II.2 : *Représente la structure de l'algorithme Radix-2 DIT pour N=8* ***22***

Figure II.3 : *Représente le schéma d'une structure butterfly DIF* ***23***

Figure II.4 : *Diagramme de la FFT à base de Butterfly Radix-2 de type DIF pour N=8* ***24***

Figure II.5 : *Organigramme du Butterfly Radix-4 de type DIT* ***26***

Figure II.6 : *Schéma d'une structure de Butterfly Radix-4 de type DIF* ***28***

Figure II.7 : *Architecture pipeline de base* ***30***

Figure II.8 : *Architecture R4SDC (N = 16 points)* ***31***

Figure II.9 : *Architecture R2SDF (N = 8 points)* ***31***

Figure II.10 : *Architecture R4SDC (N = 256 points)* ***32***

Chapitre III : Généralité sur les circuits FPGAs

Figure III.1 : *Cellule de base d'un circuit FPGA* ***35***

Figure III.2 : *Architecteur interne d'un FPGA* ***36***

Chapitre IV : Conception et implémentation

Figure IV.1 : *une architecture SDF radix-2 à deux voies pour l'FFT de taille 8* ***48***

Figure IV.2 : *présente un organigramme simplifié qui résume les étapes de conception de notre projet* ***49***

Figure IV.3 : *montrent les composants est « simulink Libray browser »* ***51***

Figure IV.4 : *schéma global du processeur FFT de taille 8 conçu en utilisant « System Generator »* ***52***

Figure IV.5 : *présentation des trois étages d'un processeur FFT de taille 8.* ***52***

Figure IV.6 : *Generation du modèle HDL par « System Generator »* ***54***

Figure IV.7 : *Lancement du modèle d'implémentation dans l'environnement « Project Navigator »* ***55***

Figure IV.8 : *Exemple de résultats d'implémentation* ***56***

Figure IV.9 : *présentation de la fréquence maximale obtenue après PAR.* ***56***

Figure IV.10 : *Exemple d'estimation de la puissance par « XPower Analyzer »* ***57***

Figure IV.11 : *passage de l'FFT de taille 8 vers l'FFT de taille 16.* ***58***

Liste des tableaux

Chapitre II : L'FFT : Algorithme et Architecture

Tableau II.1: *Comparaison du nombre d'opérations dans un calcul direct de la DFT et dans la FFT* **19**

Tableau II.2 : *Bit reversal pour N=8* **29**

Tableau II.3 : *Ressources matérielles selon l'architecture en pipeline utilisée* **32**

Tableau II.4 : *Taux d'utilisation de ressource des différentes architectures* **33**

Chapitre III : Généralité sur les circuits FPGAs

Tableaux III.1 : *Comparaison entre les circuits de type FPGA et les circuits de type ASIC* **39**

Tableaux III.2 : *Exemple de quelque circuit FPGA* **40**

Tableaux III.3 : *Quelques « familles »FPGA XILINX* **43**

Chapitre IV : Conception et implémentation

Tableau IV.1 : *Comparaison entre DFT calculée par Matlab et DFT calculée par notre modèle* **53**

Tableau IV.2 : *Présente les résultats de l'implémentation des processeurs FFT sur le circuit Virtex 7 XC7VX485t-2ffg1761* **58**

Tableau IV.3 : *Présente les résultats de l'implémentation des processeurs FFT sur le circuit Spartan 6 xc6slx45-3csg324* **59**

Tableau IV.4 : *Présenté le SQNR pour les différentes taille de l'FFT* **59**

Liste des abréviations

4M2A : *4 Multiplications réelles et 2 Additions réells.*

ALM: *Adaptation Logic Modules.*

ASIC: *Application Spécific Integrated Circuit.*

AWGN: *Additive White Gaussian Nois.*

BPE: *Bulterfly Processor Elément.*

CAO : *Conception Assistée par Ordinateur.*

CLB : *Configurable Logic Bloc.*

CMOS : *Complementary Metal Oxide Semi-conductor.*

CPLD : *Complex Programmable Logic Device.*

DFT : *Discrete Fourier Transform.*

DIF: *Decimation In Frequency.*

DIT: *Decimation In Time.*

DSP: *Digital Signal Processor.*

EEPROM: *Electrically Erasable Programmable Read Only Memary.*

FFT: *Fast Fourier Transform.*

FPGA: *Field Programmable Gate Arrays.*

HDL: *Hardware Description Langage.*

ICI: *Inter-Carrier Interference.*

IES: *Interférences Entre Symboles.*

IFFT: *Inverse Fast Fourier Transform.*

IOB: *Input Output Bloc. ISE: Integrated Software Environment.*

IP: *Intellectual Property.*

ISE: *Integrated Software Environment.*

LCA: *Logic Cell Array.*

LUT: *Look Up Table.*

MC: *Modules de Calcul.*

MDC: *Multi-path Delay Commutator.*

MIMO: *Multiple Input Multiple Output.*

MOS: *Metal Oxide Semiconductor.*

OFDM: *Orthogonal Frequency-Division Multiplexing.*

PAL: *Programmable Array Logic.*

PAPR: *Peak to Average Power Ratio.*

PE: ***P**rocessing **E**lement.*

PIP: ***P**rogramme **I**nterconnect **P**oints.*

ROM: ***R**ead **O**nly **M**emory.*

RTR: ***R**un **T**ime **R**econfiguration.*

SDC: ***S**ingle-path **D**elay **C**ommutator.*

SDF: ***S**ingle-path **D**elay **F**eedback.*

SFG: ***S**ignal **F**low **G**raphe.*

SISO-OFDM: ***S**ingle **I**mput **S**ingle **O**utput **O**rthogonal **F**requency **D**ivision **M**ultiplex.*

SOC: ***S**ystem **O**n **C**hips.*

SQNR: ***S**ignal **T**o **Q**antization noise ratio.*

SRAM: ***S**tatic **R**andom **A**ccess **M**emory.*

TEB : ***L**e **T**aux **D`e**rreur **B**inaire.*

TTL: ***T**ransistor-**T**ransistor **L**ogic.*

UPLD: ***P**rogrammable **L**ogic **D**evice.*

VHDL: ***V**ery high speed integrated circuit **H**ardware **D**escription **L**anguage.*

.

Notations

C : Célérité de la lumière.

V : Vitesse radiale entre l'émetteur est le récepteur.

Θ : l`angle entre le vecteur vitesse et la ligne émetteur récepteur.

$\mathbf{f_c}$: La fréquence de l`onde incident.

$\mathbf{f_d}$: La fréquence Doppler

N : la longueur de la transformée.

$\mathbf{W_N^{nk}}$ **:** Coefficients de base de la DFT.

$\mathbf{0(N^2)}$ **:** La complexité de calcul de la DFT.

K : le numéro de l'étage

b(t) : le bruit AWGN

e(t) : le signal émis

r(t) : le signal reçu

$\mathbf{X_{fl}(j)}$: les échantillons de la DFT éxacte (en utilisant la virgule Flottante).

$\mathbf{X_{fi}(j)}$: les échantillons de la DFT approximative (en utilisant la virgule Fixe).

N : la taille de la FFT.

x(n) et **X(k)** : les séquences d'entrée et de sortie respectivement

N : la longueur de la transformée

G(k) : la sequence pair

H(k) : la sequence impair

INTRODUCTION GÉNÉRAL

CHAPITRE I

Les Techniques De Communications

CHAPITRE II

La FFT : Algorithme et Architecture

CHAPITRE III

Généralité sur les circuits $FPGA_S$

CHAPITRE

IV

Conception Et Implémentation

CONCLUSION GÉNÉRAL

SOMMAIRE

الملخص

الهدف من هذا المشروع هو تصميم و محاكات خوارزمية FFT ثم برمجتها على دارة قابلة لإعادة البرمجة من نوع FPGA في البداية تناولنا دراسة عامة حول الاتصالات الرقمية ثم تحدثنا عن تقنية .MIMO-OFDM و بعدها تطرقنا بالدراسة إلى خوارزمية FFT كما أعطينا نبذة مختصرة حول الدارات القابلة لإعادة البرمجة من حيث بنيتها الداخلية و من حيث الأدوات المسهلة لعملية البرمجة. وفي الأخير قمنا بعرض النماذج التي صممناها مرفوقا بشرح مفصل للخطوات التي تم اتباعها هذا بالإظافة إلى عرض مختلف النتائج المتحصل عليها.

الكلمات المفتاحية: دارة قابلة لإعادة البرمجة FPGA خوارزمية FFT تقنية .MIMO-OFDM .

Résumé :

L'objectif de ce projet de fin d'études consiste à l'implémentation sur FPGA de l'algorithme FFT à base de Radix-2 en utilisant l'architecture SDF pour le calcul de la DFT de deux séquences d'entrée parallèle.

Nous avons tous d'abord commencé par donner des généralités sur les communications numériques et la technique MIMO-OFDM. Puis nous avons présenté une étude sur l'algorithme FFT du point de vue algorithmique et architecturale. Ensuite nous avons donné un aperçu sur les circuits FPGA, leurs architectures, leurs avantages et les outils de leur programmation. Finalement nous avons présenté les modèles que nous avons développés avec une explication des différentes étapes que nous avons suivies pour l'implémentation des modèles sur FPGA. En plus nous avons présenté les résultats de simulation et d'implémentation de chaque modèle.

Mots clés: FFT, MIMO-OFDM, FPGA, SDF.

ABSTRACT:

The main objective of our project is to implement the Radix-2 FFT algorithm on FPGA device using the SDF architecture to process in parallel two independent streams.

First, we presented generalities about communication systems and MIMO-OFDM technique. Then we provided a study about FFT technique from architectural and algorithmic aspect. After that, we presented a preview on FPGA circuits, their global architecture, advantages applications and their developing tools. Finally, we presented the designed FFT processors for various sizes with explanation of different steps followed during the implementation process. The simulation and implementation results are also presented for each designed processor.

Keywords: FFT, SDF, MIMO-OFDM, FPGA.

Introduction Générale

De nos jours nous vivons dans un monde où les télécommunications jouent un rôle primordial, tant par la place qu'elles occupent dans le quotidien de chacun, que par les enjeux économiques et technologiques dont elles font l'objet. Le succès des télécommunications se repose sur plusieurs paramètres comme l'efficacité spectrale, la robustesse par rapport aux propagations multitrajets et la complexité d'implémentation. Parmis les techniques qui a permettent d'atteindre ces performances, est l'utilisation de la modulation multiporteuse OFDM « orthogonal frequency-division multiplexing » combinée avec la technologie MIMO « Multiple-Input Multiple-Output » .

Les circuits de type FPGA « Field-Programmable Gate Array » est une technologie très promoteuse pour l'implementation des algorithmes de traitement numérique du signal utilisés dans le domaine des télécommunications. Cette technologie offre plusieurs avantages comme la reconfigurabilité, le coût de réalisation et le temps de développement.

Parmi les algorithmes les plus adaptés aux circuits FPGA on trouve l'algorithme FFT, les recherches dans ce domaine montrent qu'on peut implémenter l'algorithme FFT dans un circuit FPGA en assurant de meilleures performances avec une consommation minimale de ressource et de puissance.

L'objectif de notre projet de fin d'études est l'implémentation sur FPGA de l'algorithme FFT à base de Radix-2 en utilisant l'architecture SDF pour le calcul de la DFTs de deux séquences d'entrée en parallèle. Le processeur FFT obtenu est utilisé dans un système de communication de type MIMO-OFDM.

Cette étude est composée de quatre chapitres répartis comme suit:

Le premier chapitre porte sur des généralités sur communications numériques. Il décrit brièvement le fonctionnement d'une chaîne de transmission. Il présente aussi la technique OFDM, la technique MIMO et leur combinaison appelée MIMO-OFDM.

La présentation de l'algorithme FFT fait l'objet du second chapitre. Nous étudions les algorithmes conventionnels les plus communs: Radix-2, Radix-4. Nous donnons ainsi un aperçu sur la complexité arithmétique des algorithmes de la FFT. Finalement nous présentons les différentes architectures conventionnelles de l'implémentation de la FFT.

Le troisième chapitre a pour objectif de donner un aperçu général sur les circuits FPGAs. Nous commençons par la présentation de l'architecture de base des FPGAs où

nous expliquons les différents blocs constituant le circuit. Nous donnons ensuite une petite comparaison entre le FPGA et l'ASIC ainsi que les avantages et les inconvénients de chaque circuit. Enfin nous décrivons les outils et les étapes de développement des applications à base de circuit FPGA.

Le quatrième chapitre est la phase pratique de notre projet. Ce chapitre est consacré à la présentation de l architecture utilisée pour l'implémentation ainsi que les différents résultats de simulation et d'implémentation de l'algorithme FFT sur circuit FPGA Virtex7 et Spartan 6.

Ce mémoire se termine par une conclusion générale et des perspectives pour la continuité de ce travail.

I)- Introduction

La communication numérique est une nouvelle discipline de la communication qui désigne l'ensemble des actions visant à diffuser des messages par le biais d'un média numérique, web, médias sociaux, application mobile ou autres.

L'apparition des nouvelles technologies ainsi que les progrès effectués dans le domaine des communications sans fil permettant de transmettre de la voix, des données ou des contenus multimédias ont entraîné l'apparition des systèmes MIMO-OFDM, permettant une meilleure exploitation de la diversité spatiale et de la diversité fréquentielle. L'objectif de ce chapitre est de donner d'abord une vue générale sur les chaînes de transmission, en présentant la composition d'une chaîne de communication numérique. Nous abordons en suite la technique OFDM et la technique MIMO. On parlera enfin d'un système MIMO utilisant la modulation OFDM appelé MIMO-OFDM.

II)- Les chaînes de transmission Numériques

Les chaînes de transmission numériques véhiculent de l'information sous forme numérique entre une source et un ou plusieurs destinataires, en utilisant un support physique comme le câble, la fibre optique ou encore la propagation sur un canal radioélectrique. Les signaux transportés peuvent être soit directement d'origine numérique, comme dans les réseaux de données, soit d'origine analogique (parole, image...) mais convertis sous une forme numérique. La tâche de la chaîne de transmission est d'acheminer l'information de la source vers le destinataire avec la plus grande fiabilité possible [1].

On peut identifier trois éléments qui permettent d'acheminer l'information de la source vers un destinataire à savoir la chaîne d'émission, la chaîne de réception et le canal de transmission, comme le montre le schéma synoptique donné par la figure (I.1).

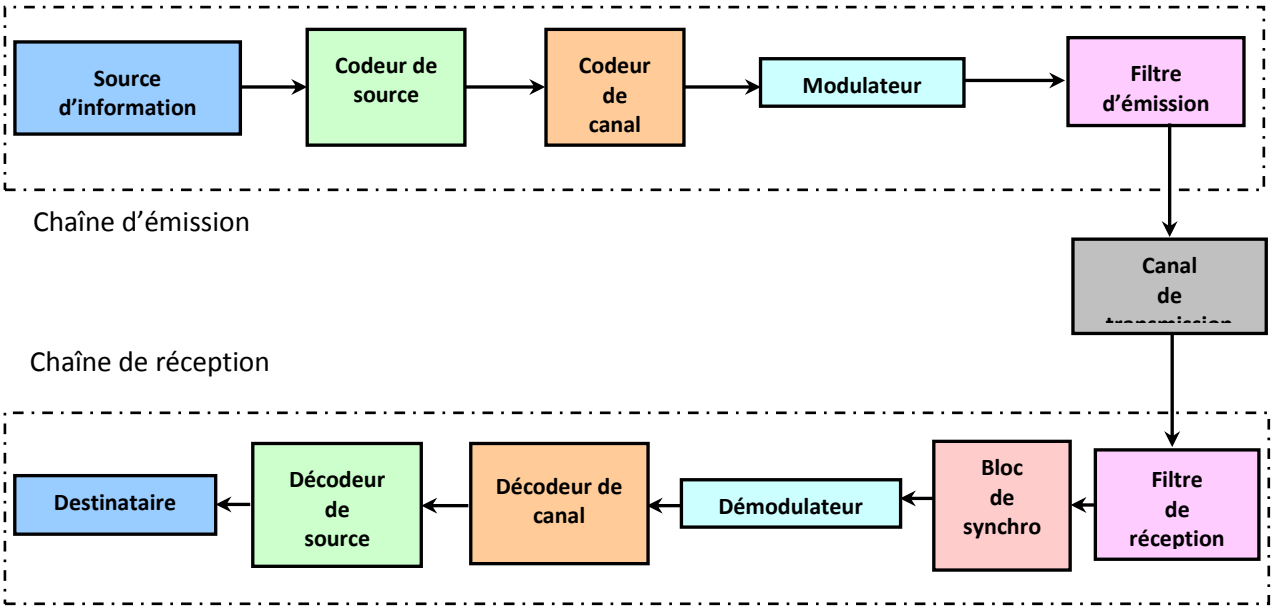


Figure (I.1) : Schéma synoptique d'une chaîne de transmission numérique

II.1)- La chaîne d'émission :

Elle se compose essentiellement des éléments suivants :

II.1.1)- La source d'information :

Le signal existe naturellement sous forme analogique, par exemple : électrique, sonore, vidéo, etc. Le signal analogique est défini comme étant une quantité physique qui varie continuellement dans le temps.

La transmission numérique des signaux comme son nom l'indique, nécessite la numérisation du signal analogique en entrée et cela se fait en trois étapes : échantillonnage, quantification et codage

II.1.2)- Le codeur de source :

Le codeur de source a pour but la compression des données. Elle consiste à supprimer toute redondance inutile à la récupération de l'information afin de raccourcir le message à transmettre et diminuer ainsi le temps de transmission où la taille du volume de stockage.

Le nom de code de source correspond à la source des données numériques, c'est à dire au train binaire qui vient du convertisseur analogique numérique. Le terme de compression est aussi couramment utilisé pour désigner cette opération. Il s'agit de comprimer des données numériques, sous forme binaire et non une compression du signal physique comme c'est le

cas pour la quantification. La compression du signal de la source peut être avec ou sans pertes.

II.1.3)- Le codeur de canal :

L'objectif du codeur de canal est d'établir un système de contrôle des erreurs par un nouveau codage du message. Ceci se réalise en insérant des éléments binaires (de la redondance) au message. Cette redondance permet éventuellement au récepteur de détecter voir de corriger les erreurs de transmission. Il agit donc dans un sens antagoniste au codage de source qui vise à enlever des éléments binaires. Il se place dans l'architecture de la chaîne avant la modulation.

II.1.4)- Le Modulateur :

La modulation a pour objectif d'adapter le signal à émettre au canal de transmission. Cette opération consiste à modifier un ou plusieurs les paramètres d'une onde porteuse centrée sur la bande de fréquence du canal.

Les paramètres qu'on peut modifier sont : l'amplitude, la fréquence et la phase de la porteuse.

Dans le cas où le signal à transmettre est analogique, on effectue une modulation d'amplitude, de fréquence ou de phase (AM, FM PM) ; pour le signal numérique, on parle plutôt de modulation par saut d'amplitude, de fréquence ou de phase (ASK, FSK, PSK) car les modifications d'amplitude, de phase ou de fréquence, sont brutales et discrètes

II.2)- La chaîne de réception :

Elle se compose essentiellement des éléments décrits ci-après :

II.2.1)- Le démodulateur :

Le démodulateur joue le rôle dual du modulateur, il consiste à transformer le signal reçu en un train binaire du signal modulé. La technique de démodulation dépend du type de modulation utilisée lors de la transmission et on distingue :

La démodulation cohérente : la porteuse locale est reconstituée en synchronisme et en phase avec la porteuse reçue (ce procédé est optimal vis-à-vis de la probabilité d'erreur) et la démodulation se fait en appliquant un battement entre la porteuse et le signal reçu.

La démodulation non cohérente : la démodulation se fait sans utilisation d´une porteuse

locale.

II.2.2)- Le décodeur de canal :

Le décodeur de canal effectue la fonction inverse de la fonction du codeur de canal. Connaissant la loi de codage utilisé, le récepteur est donc capable de détecter puis éventuellement corriger les données binaires erronées, ce qui permet de retrouver dans la mesure du possible l'information binaire initiale. En générale les codes correcteurs ne peuvent pas tout corriger sinon ils demanderaient trop de bits redondants. Ils se contentent de corriger les erreurs les plus probables.

II.2.3)- Le décodeur source :

Ayant la séquence binaire ou message associé à l'information, ce décodeur doit reconstituer l'information transmise.

II.2.4)- Le destinataire :

Le destinataire inclut tout appareil adéquat capable de traiter, de lire ou d'afficher le signal reçu contenant l'information émise préalablement par la source. Il constitue aussi l'entité à laquelle le message est destiné. Cela peut être un ordinateur, un téléphone, un poste radio ou de télévision et autres.

II.2.5)- Le canal de transmission :

C'est l'élément physique par lequel les informations de la source seront transmises au récepteur. Selon la nature des signaux, les canaux sont de nature différente :

- ✓ Atmosphère : ondes électromagnétiques ou ondes Hertziennes.
- ✓ Câble coaxial , paire torsadée : signaux électriques (tensions, courants).
- ✓ Fibre optique : ondes électromagnétiques optiques (lumière visible, infrarouge, …).

Quel que soit le canal utilisé, ils ont tous un point commun qui sont la perturbation du signal due au bruit, ce qui dégrade la qualité du message.

Les canaux de transmission peuvent être classés selon l'effet qu'ils ont sur le signal.

On distingue :

-Canal à bruit additif blanc gaussien

Le modèle du canal le plus fréquemment utilisé pour la simulation de transmissions numériques, qui est aussi un des plus faciles à générer et à analyser, est le canal à bruit blanc additif gaussien (en anglais : AWGN). Ce bruit modélise à la fois les bruits d'origine interne (bruit thermique dû aux imperfections des équipements...) et le bruit d'origine externe (bruit d'antenne...). Ce modèle est toutefois associé à une transmission filaire, puisqu'il représente une transmission quasi-parfaite de l'émetteur au récepteur. Le signal reçu s'écrit alors [1] :

$$r(t)=e(t)+b(t) \qquad \text{I.1}$$

Où $b(t)$ représente un bruit AWGN caractérisé par un processus aléatoire gaussien, *e(t)* est le signal émis et *r(t)* est le signal reçu.

- Canal à évanouissements

Les communications radio ont souvent besoin d'un modèle élaboré prenant en compte les différences de propagation du milieu, appelées encore atténuations ou évanouissements, qui affectent la puissance du signal transmis. Cette atténuation est principalement due à:

- **L'effet Doppler** : qui est une variation temporelle du canal due à un mouvement relatif de l'émetteur et du récepteur. Le signal reçu est sujet à un décalage aléatoire de fréquence variant entre F_d et $-F_d$ avec :

$$F_d = f_c\left(\frac{v}{c}\right) . Cos\,\Theta \qquad \text{(I-2)}$$

Avec : c : Célérité de la lumière.

v : Vitesse radiale entre l'émetteur est le récepteur.

Θ : l`angle entre le vecteur vitesse et la ligne émetteur récepteur.

f_c : La fréquence de l`onde incidente.

f_d : La fréquence Doppler.

- **L'effet multi trajets** : dû à un environnement de propagation riche en échos, qui provoque la variation en phase et en amplitude du signal, ainsi que la variation fréquentielle du canal de propagation (sélectivité fréquentielle).

On peut distinguer deux types de canaux à effet multi trajets qui sont :

- **Le canal de Rayleigh** : c'est le modèle du canal le plus couramment utilisé pour les communications radio mobiles si on considère qu'il n'existe pas de trajets directs entre l'émetteur et le récepteur [1].

- **Le canal de Rice :** c'est un canal multi trajets avec l'existence d'un trajet direct prépondérant en puissance entre l'émetteur et le récepteur [1].

Un autre effet dû aux canaux de transmission est celui des interférences entre symboles (*IES*) qui affaiblissent les performances de la détection qui peut être réduit par l'adjonction d'un filtre de Nyquist dans la chaîne de transmission et le problème de synchronisation qui peut être réglé en introduisant un bloc de récupération du rythme.

III)- La Technique OFDM

Les techniques qu'on appelle multi porteuses consistent à transmettre des données numériques simultanément en les modulant sur un grand nombre de porteuses. Ce sont des techniques de multiplexage en fréquence qui existent depuis longtemps. Le regain d'intérêt actuel réside dans l'amélioration apportée pour augmenter l'efficacité spectrale en « orthogonalisant » les porteuses, ce qui permet d'obtenir un meilleur recouvrement spectral. L'implémentation de la modulation et de la démodulation s'effectue à l'aide de circuits performants basée sur la transformée de Fourier rapide (FFT). En résumé, l'idée principale de l'OFDM consiste à diviser la bande spectrale disponible en sous-canaux (sous-porteuses). Cette subdivision spectrale conduit à des sous canaux dont la bande de fréquence reste très faible par rapport à la bande de cohérence du canal. Ceci garantit la propriété de non sélectivité en fréquence du canal [2]. Pour obtenir une efficacité spectrale élevée, les réponses fréquentielles des sous-canaux sont en partie non disjointes mais orthogonales, d'où l'appellation OFDM.

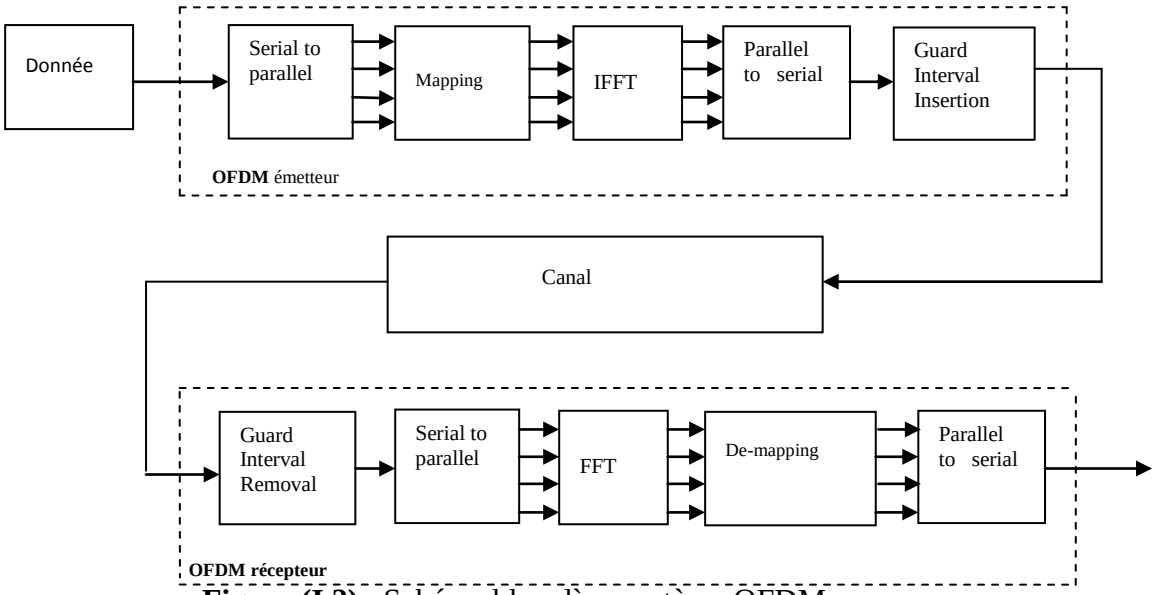


Figure (I.2) : Schéma bloc d`un système OFDM

III.1) -Avantages et inconvénients de l'OFDM :

Les avantages de la technique OFDM sont nombreux :

1. Le modulateur et le démodulateur peuvent être réalisés en utilisant une IFFT et une FFT respectivement.
2. La technique OFDM dispose d'une très grande flexibilité tant dans l'allocation de bits par sous-porteuse que dans l'allocation de débit dans un contexte multi-utilisateurs.
3. Les interférences entre symboles (ISI) et entre porteuses (ICI) peuvent être évitées à l'émission et corrigées en réception.
4. Les techniques multi-porteuses sont robustes en termes d'égalisation.

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

1. L'OFDM est sensible au décalage fréquentiel qui peut causer une perte d'orthogonalité entre les sous-porteuses.

2. L'OFDM est sensible aux non linéarités des amplificateurs. Le signal peut présenter un PAPR (Peak to Average Power Ratio) très élevé, engendrant la saturation des blocs analogiques [3].

IV) - Les Système MIMO

La technologie MIMO consiste à augmenter le débit des transmissions sans fil à puissance émise donnée en multipliant le nombre d'antennes tant au niveau de l'émetteur que du récepteur.

Les systèmes MIMO introduisent une nouvelle forme de diversité connue sous le nom de diversité spatiale. Cette technique cherche à exploiter les phénomènes d'évanouissement et de multi-trajets, longtemps considérés comme des handicaps. En effet, le récepteur reçoit et combine plusieurs copies d'un même signal envoyées à partir de plusieurs antennes et subissant des évanouissements indépendants. Par conséquent, le récepteur prend la bonne décision permettant de reconstituer le message d'origine sans erreur. En résumé, l'utilisation de plusieurs antennes au niveau des systèmes MIMO peut servir à améliorer la fiabilité de transmission en réduisant la probabilité d'erreur (gain en diversité) et à augmenter le débit de transmission (gain de multiplexage) [4,5].

Le système MIMO est un système ayant N_t antennes d'émission et N_r antennes de réception, le signal reçu par l'antenne j (j=1, 2, …,N_r) est la somme des contributions des symboles émis (s_1, …,s_{N_t}) multipliée par le gain complexe h_{ij} des liaisons correspondantes :

$$y_i=\sum_{j=1}^{N_t} h_{ij}\, s_j + n_j \qquad \text{I.3}$$

Où h_{ij} représente le gain, qui n'est autre que la réponse impulsionnelle du canal, entre l'antenne émettrice i et 'antenne réceptrice j.

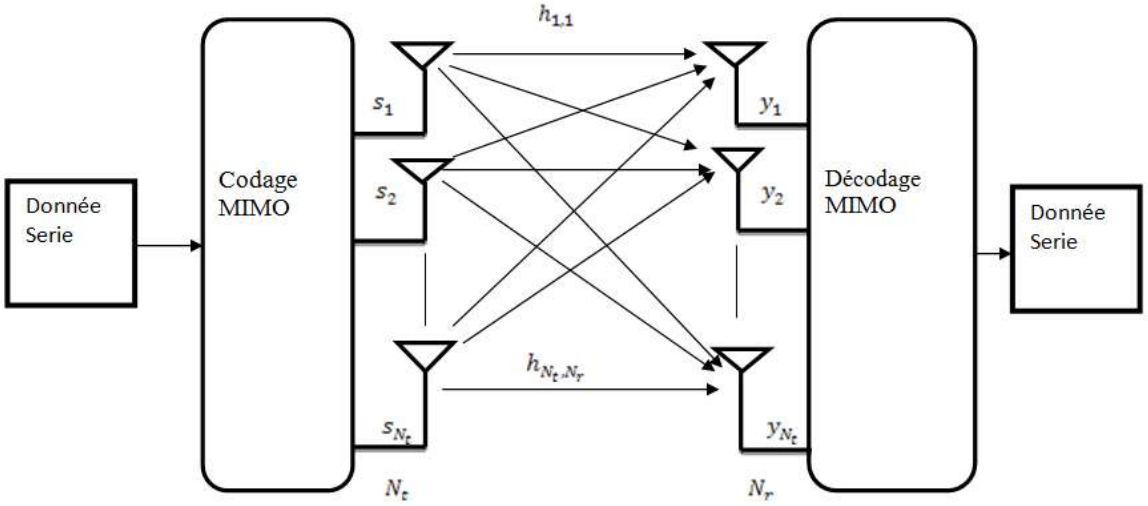
La figure I.3, représente-le modéle d'un système MIMO avec le démultiplexage (mise en parallèle des données série) des symboles sj sur chaque antenne et le multiplexage (mise en série des données parallèles).

L'équation I.3 peut être réécrite d'une manière matricielle afin de la rendre plus synthétique :

$$Y = H * S + n \quad \text{I.4}$$

Ou $Y = [y_1 \dots y_{N_r}]^T$, $S = [s_1 \dots s_{N_r}]^T$, $N = [n_1 \dots n_{N_r}]^T$ I.5

Et $H = \begin{pmatrix} h_{11} & h_{12} & \dots h_{1N_t} \\ h_{21} & h_{22} & \dots h_{2N_t} \\ \vdots & \ddots & \ddots \vdots \\ h_{N_{r1}} & h_{N_{r2}} & \dots h_{N_r N_t} \end{pmatrix}$ I.6



Figure(I.3) : schéma d`un système MIMO

Les techniques MIMO mises en œuvre se divisent en trois catégories :

- **La diversité spatiale MIMO :** On transmet simultanément un même message sur différentes antennes à l'émission. Les signaux reçus sur chacune des antennes de réception sont ensuite combines d`une façon cohérente. Cela permet de diminuer le taux d`erreur binaire (TEB) du système.

- **Le multiplexage spatial MIMO** : Chaque message est segmenté en sous-messages. On transmet simultanément les sous-messages différents sur chacune des antennes d'émission. Les signaux reçus sur les antennes de réception sont réassemblés pour reformer le message entier d'origine. Le multiplexage MIMO permet d'augmenter les débits de transmission.

- **Le MIMO Beamforming** : On utilise le réseau d'antenne MIMO pour orienter et

contrôler le faisceau d'onde radio. On peut ainsi créer des lobes et optimiser une transmission entre l'émetteur et la cible. Les techniques de beamforming permettent à la fois d'étendre une couverture radio (d'une station de base ou d'un point d'accès par exemple) et de limiter les interférences entre utilisateurs.

V)- Système MIMO-OFDM

Dans les systèmes sans fil à large bande, les Canaux MIMO sont gravement touchés par les évanouissements sélectifs en fréquence, pour y remédier, MIMO peut être combiné avec un système OFDM, pour donner naissance à un système MIMO-OFDM. L'association MIMO-OFDM est la base du système de communication de la quatrième génération 4G, puisque la technique MIMO

Peut apporter un gain en diversité ainsi qu'améliorer la capacité du système, et l'OFDM peut atténuer les effets des évanouissements sélectifs en fréquence. Le schéma bloc d`un système de communication utilisant la technique MIMO-OFDM est donnée dans la figure I.4.

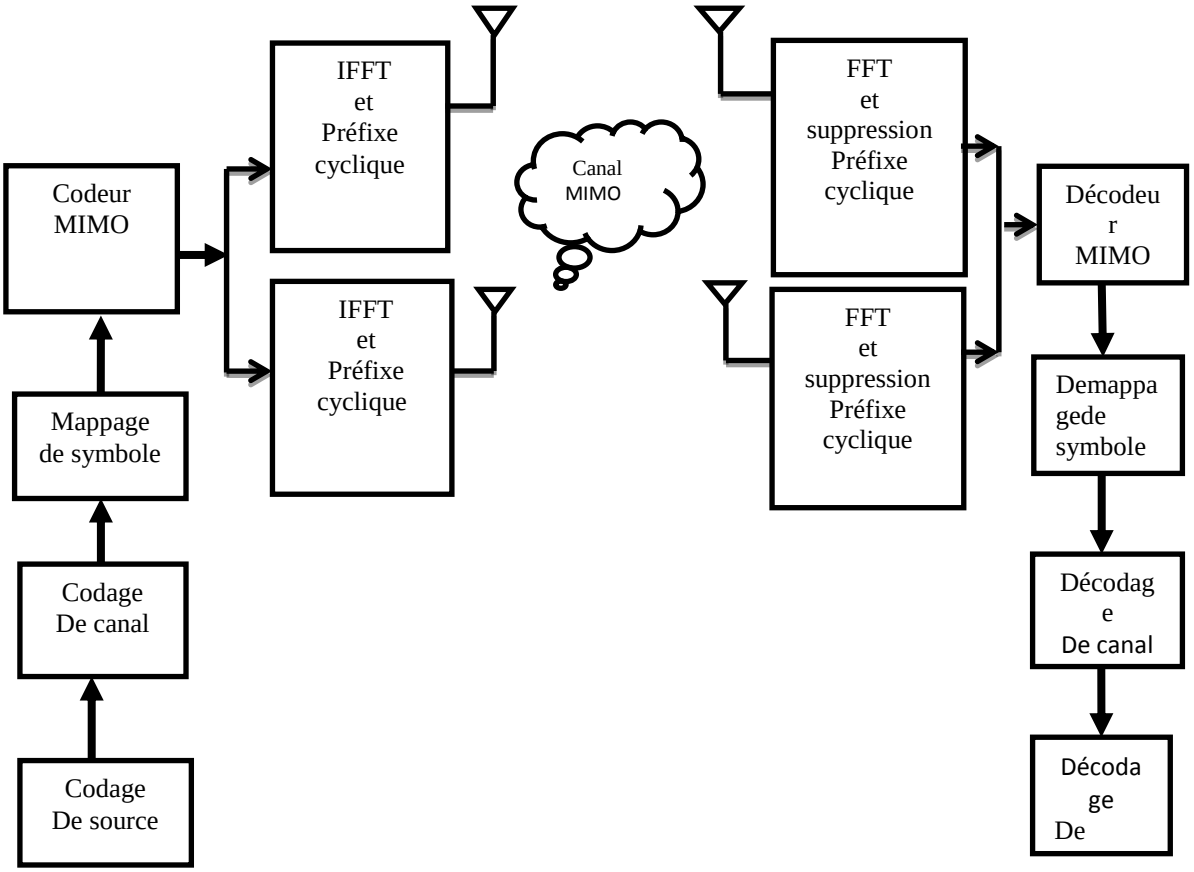


Figure (1.4) : schema simplifie d`un système MIMO-OFDM

Les systèmes MIMO sont l'un des principaux axes de développement pour augmenter les débits des communications sans fil et en effet nous assistons à un très rapide développement de cette technologie avec des applications dans les réseaux locaux sans fil et les réseaux de communication de 3ème, 4ème et 5ème génération.

IV)-CONCLUSION

Nous avons présenté dans ce premier chapitre quelques généralités sur les transmissions numériques. Nous avons donné les principaux blocs de chaîne de transmission, de la source binaire jusqu'au destinataire. Les modèles de canaux AWGN et de Rayleigh ont été décrits.

La technique OFDM apparait comme une bonne solution pour combattre le problème d'interférence dans les communications haut débit. Cette technique consiste à diviser la bande de transmission en N sous-canaux orthogonaux afin de rendre le canal non sélectif en fréquence.

La technique MIMO représentent un moyen efficace pour augmenter le débit du système de communication à travers le multiplexage spatiale et aussi pour améliorer le taux d'erreur binaire à travers le codage spatio-temporelle.

Un système de communication peut bénéficier des avantages de la modulation OFDM et des avantages de la technique MIMO par l'utilisation conjointe des deux techniques c'est le MIMO-OFDM.

I)- Introduction

Dans le domaine du traitement numérique du signal, la Transformée de Fourier Discrète(TFD), ou en anglais DFT *(Discrete Fourier Transform)* joue et jouera un rôle très important dans l'analyse, la conception et l'implémentation des algorithmes et systèmes futurs du traitement du signal temporel discret. La DFT est un procédé mathématique qui reste au centre du traitement prenant place à l'intérieur d'un processeur numérique, Son application est aussi utilisée dans plusieurs domaines technologiques tels que les télécommunications, le biomédical, le traitement séismique, etc ....

La DFT peut, par exemple, calculer la réponse fréquentielle d'un signal, peut servir comme une étape intermédiaire dans des techniques plus élaborées de traitement numérique du signal comme la modulation OFDM (Orthogonal Frequency Division Multiplexing).

Plusieurs algorithmes de transformées de Fourier rapides, plus communément appelées FFT pour Fast Fourier Transform, ont pour but d'accélérer le calcul de la transformée de Fourier discrète.

En 1965 J. W. Cooley et de J. W. Tukey publient le célèbre algorithme construit autour d'une propriété de la transformée de Fourier et démontrent que le calcul d'une transformée de Fourier sur N échantillons (sachant que $\boldsymbol{N = r^n}$) peut être factorisé par le calcul de $\boldsymbol{r}$ transformées de Fourier sur n échantillons.

Aujourd'hui nous disposons d'un arsenal important d'algorithmes FFT, dont le choix et l'efficacité de ce dernier, dépendent du respect des hypothèses permettant d'exploiter les propriétés correspondantes et d'obtenir le maximum de performances de ces algorithmes.

Le but de ce chapitre est de :

- ✓ Présenter l'algorithme FFT ainsi que les différents algorithmes conventionnelles radix-2, radix-4 ;
- ✓ Illustrer la complexité de l'algorithme FFT ;
- ✓ Donner un aperçu sur les architectures d'implémentation les plus connues.

II) -Concepts de base

La DFT d'un signal discret *x(n)* peut être directement calculée par l'équation. [7]

$$X_F(k) = \sum_{n=0}^{N-1} x_F(n)\, W_N^{nK} \qquad \text{II .1}$$

Avec *x(n)* et *X(k)* sont les séquences d'entrée et de sortie respectivement. Les deux séquences *x(n)* et *X(k)* sont des séquences de nombres complexes. *N* est la longueur de la transformée, elle est de valeur finie pour limiter les calculs exécutés par le processeur numérique qui lui aussi est limité par la taille de sa mémoire.

$$W_N^{nk} = e^{-j\left(\frac{2\pi}{N}\right)nk} \qquad \text{II.1}$$

$$j^2 = -1 \qquad \text{k=0,1,…, N-1.}$$

LA DFT inverse est données par :

$$x_F(n) = \frac{1}{N}\sum_{K=0}^{N-1} X_F(k) W_N^{-nk} \qquad \text{II.2}$$

III)- Algorithmes FFT

La transformée de Fourier rapide(FFT) est une façon très efficace de calculer la transformée de Fourier discrète (DFT). Le terme FFT ne désigne pas une nouvelle transformée, ni un algorithme, mais un ensemble d'algorithmes qui permettent d'accélérer le traitement de la DFT. Tous ces algorithmes sont basés sur un même principe qui consiste à décomposer le calcul de la DFT en plusieurs DFT de longueur plus petite. Les algorithmes les plus connus et les plus utilisés sont radix-2, radix-4, radix-8, etc. Pour chaque algorithme, il existe deux versions duales et de même complexité, à savoir l'entrelacement en fréquence (Décimation-In-Frequency DIF) et l'entrelacement en temps (Decimation-In-Time DIT).

L'utilisation de l'algorithme FFT au lieu de l'application directe de la relation de l'équation II.1, permet de réduire considérablement le nombre d'opération, par exemple pour le radix-2 la complexité de calcul est réduite $0(N^2)$ à $0(N\log_2(N))$,

Le tableau II .1 donne un résumé des valeurs N^2, $Nlog_2(N)$, $N/log_2(N)$ pour diverses valeurs de *N*. La dernière colonne montre de combien la vitesse de calcul peut être augmentée par l'utilisation de la FFT au lieu du calcul direct de DFT.

Tableau (II .1) : Comparaison du nombre d'opérations dans un calcul direct de la DFT et dans la FFT [6]

N	N^2	$N.\log_2(N)$	$N/\log_2(N)$
2	4	2	2.00
4	16	8	2.00
8	64	24	2.67
16	256	64	4.00
32	1024	160	6.40
64	4096	384	10.67
128	16384	896	18.29

Dans la littérature, on emploie le mot Radix-r ou base-b, qui signifient le nombre d'entrée de chaque sous-système DFT qui compose l'algorithme FFT. Toutefois les algorithmes sont nommés Radix-r. Par exemple si $r = 2$, l'algorithme s'appellera Radix-2 où

$$N = r^n = 2^n \qquad \text{II.3}$$

Dans la section suivante, nous verrons de plus près, les deux algorithmes conventionnels FFT les plus communs : Radix-2, Radix-4[6].

III.1)- Radix 2 :

L'algorithme FFT à base de Radix-2 signifie radical-2 ou base-2, possède la structure la plus simple pour le calcul de la DFT, il se caractérise par :

- ✓ Une taille N de la FFT à calculer soit à base de 2.
- ✓ Une structure simple nommée Structure Butterfly à cause de son schéma en forme de papillon.

Il existe deux algorithmes :

-DIT FFT (Decimation In Time, Décimation dans le temps).

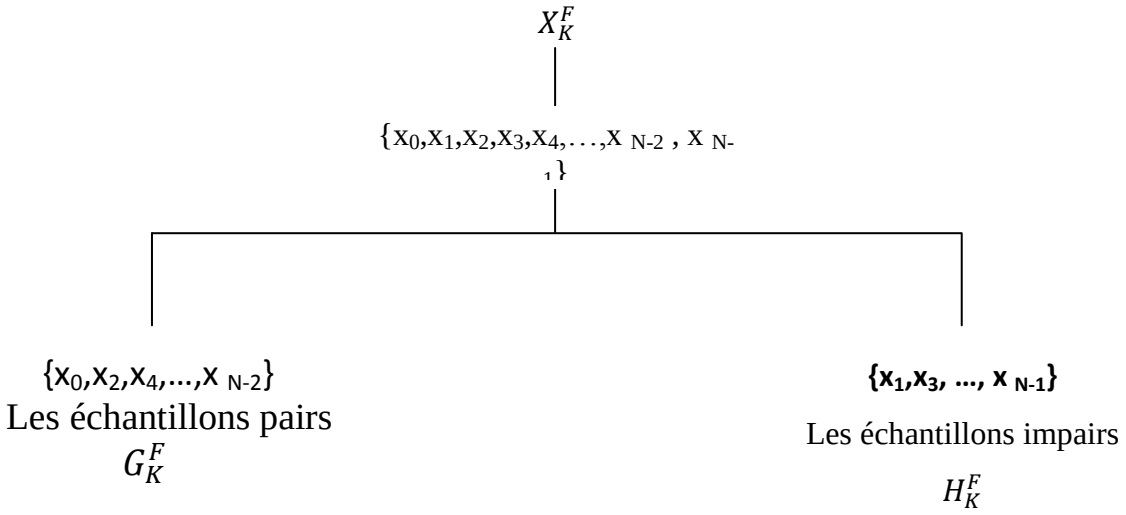
-DIF FFT (Decimation In Frequency, Décimation en fréquence).

Le Radix-2 se base sur le principe « diviser pour régner », autrement dit-il fractionne 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.

III.1.1) –l'algorithme FFT Radix-2 DIT

Cet algorithme est basé sur la décomposition de la séquence de N point en deux séquences de N/2 point (l'un pour échantillons pairs et les autres pour les échantillons impairs) [6]



ona:

$$X^F(\mathrm{k}) = \sum_{n=0}^{N-1} x\ [n]W_N^{nK}, k = 0,1, \ldots., N-1$$

$$= \sum_{npair} x(n)W_N^{nk} + \sum_{nimpir} x(n)W_N^{nk}$$

$$= \sum_{r=0}^{\left(\frac{N}{2}\right)-1} x[2r]W_N^{2rk} + \sum_{r=0}^{\left(\frac{N}{2}\right)-1} x[2r+1]W_N^{(2r+1)k}$$

$$= \sum_{r=0}^{\left(\frac{N}{2}\right)-1} x(2r)(W_N^2)^{rk} + W_N^k \sum_{r=0}^{\left(\frac{N}{2}\right)-1} x(2r+1) \quad \text{II.3}$$

Noter que:

$$\mathrm{W_N^2} = \exp\left[\frac{-\mathrm{j}2(2\pi)}{\mathrm{N}}\right] = \exp\left(\frac{-\mathrm{j}2\pi}{\mathrm{N}/2}\right) = \mathrm{W_{N/2}}$$

$$X^F(\mathrm{K}) = \sum_{r=0}^{\left(\frac{N}{2}\right)-1} x(2r)W_{\frac{N}{2}}^{rk} + W_N^k \sum_{r=0}^{\left(\frac{N}{2}\right)-1} x(2r+1)W_{\frac{N}{2}}^{rk} = G^F(k) + W_N^k H^F(k), k = 0,1, \ldots, \frac{N}{2} - 1 \quad \text{II.4}$$

$X^F(k)$: $Periodique\ de\ periode\ N,\ \ X^F(k) = X^F(k+N)$

$G^F(k), H^F(k)$: $Periodique\ de\ periode \frac{N}{2}$:

$$G^F(k) = G^F\left(k+\frac{N}{2}\right) H^F(k) = H^F\left(k+\frac{N}{2}\right) \qquad \text{II.5}$$

Donc :

$$X^F(k) = G^F(k) + W_N^k H^F(k)\ , K = 0,1,\dots,\frac{N}{2}-1 \qquad \text{II .6}$$

$$X^F\left(k+\frac{N}{2}\right) = G^F(K) + W_N^{k+\frac{N}{2}} H^F(k) \qquad \text{II .7}$$

Avec :

$$W_N^{\frac{N}{2}} = exp\left[-\frac{j2\pi N}{2N}\right] = \exp(-j\pi) = -1$$

$$W_N^{k+N/2} = exp\left[-\frac{j2\pi}{N}\left(k+\frac{n}{2}\right)\right] = W_N^k W_N^{N/2} = -W_N^k$$

Donc :

$$X^F\left(k+\frac{N}{2}\right) = G^F(k) - W_N^k H^F(k),$$
$$K = 0,1,\dots,\frac{N}{2}1 \qquad \text{II .8}$$

Les équations II.6 et II.8 montrent que la DFT d'une séquence de N points (X(k)) peut être obtenu en calculant les deux DFT des séquences pair et impaire (G(k) et H(k)) chaqu' une de taille N/2.

La Figure II .1 représente le schéma d'une structure butterfly DIT

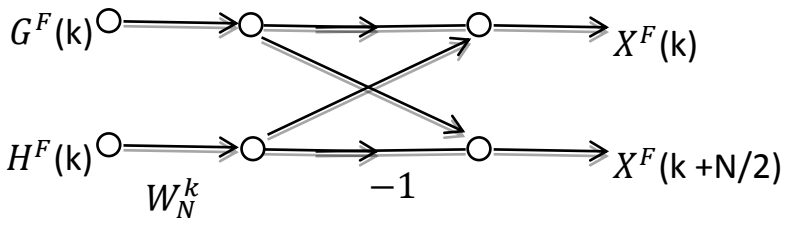


Figure (II .1**)** : Structure DIT butterfly

La Figure II .2 représente la structure de l'algorithme Radix-2 DIT pour N=8 :

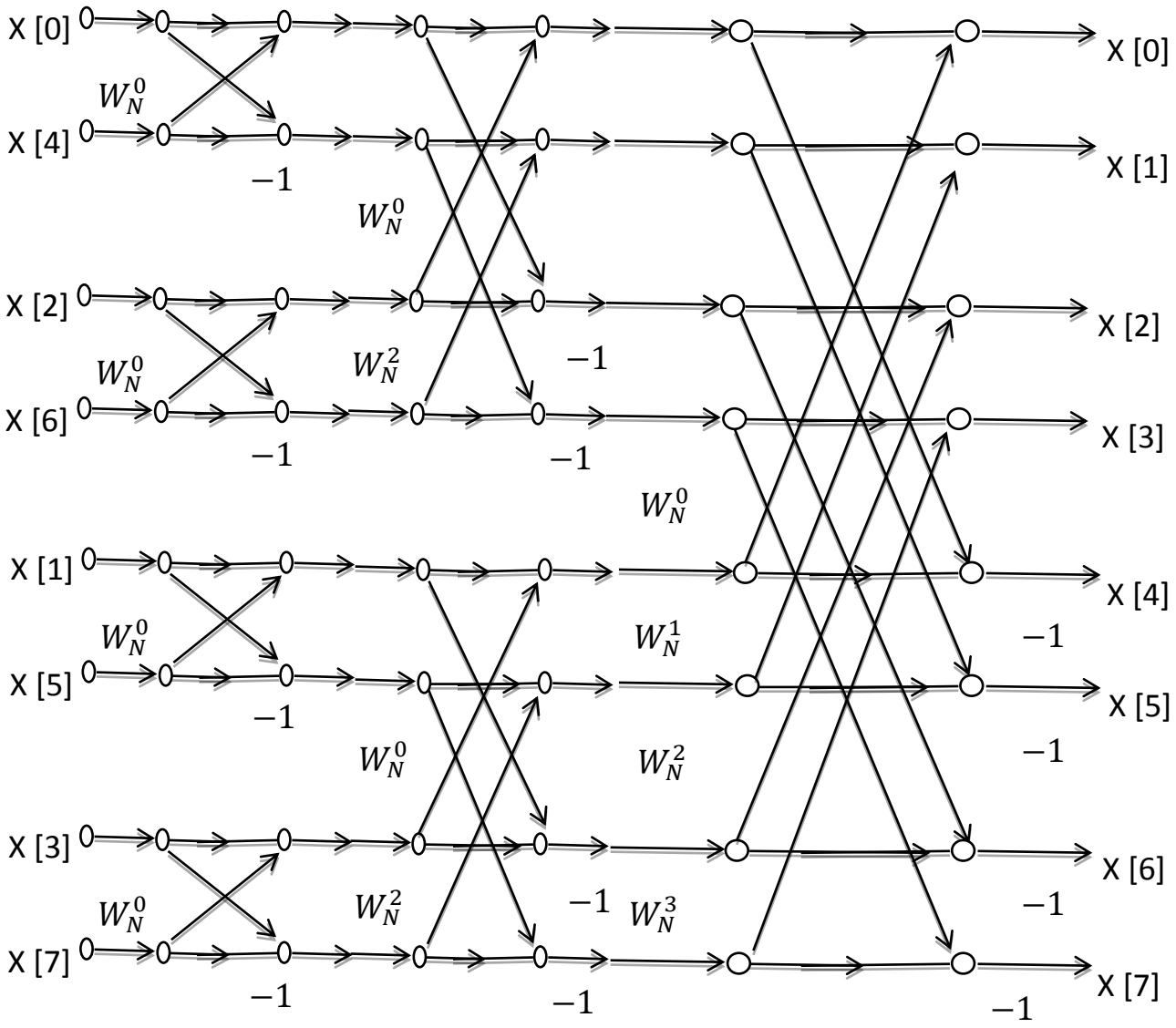


Figure (II .2**)** : Diagramme de la FFT à base de Butterfly Radix-2 de type DIT pour N=8

III.1.2)- Décimation fréquentielle (DIF)

L'algorithme radix-2 décimation en fréquence réorganise la transformée de Fourier discrète (DFT) de l'équation (II.1) en deux parties : calcul des indices paires X (k) pour k = [0, 2, 4, … N- 2], (X (2r)) et calcul des indices impairs k = [1,3, 5, ….N - 1] (X (2r + 1)) [7]

$$X(2r) = \sum_{n=0}^{N-1} (x(n) w_N^{2rn})$$

$$= \sum_{n=0}^{\frac{N}{2}-1} (x(n) w_N^{2rn}) + \sum_{n=0}^{\frac{N}{2}-1} \left(x\left(n + \frac{N}{2}\right) w_N^{2r(n+\frac{N}{2})} \right)$$

$$=\sum_{n=0}^{\frac{N}{2}-1}(x(n)w_N^{2rn})+\sum_{n=0}^{\frac{N}{2}-1}\left(x\left(n+\frac{N}{2}\right)w_N^{2rn}\right)$$

$$=\sum_{n=0}^{\frac{N}{2}-1}\left(x(n)+x\left(n+\frac{N}{2}\right)w_{N/2}^{rn}\right) \qquad \text{II.9}$$

$$=DFT_{\frac{N}{2}}\left[x(n)+x\left(n+\frac{N}{2}\right)\right]$$

$$X\,(2r+1)=\sum_{n=0}^{N-1}\,(x(n)w_N^{(2r+1)n}\,)$$

$$=\sum_{n=0}^{\frac{N}{2}-1}\left(\left(x(n)+W_N^{\frac{N}{2}}x\left(n+\frac{N}{2}\right)\right)W_N^{r(n+1)N}\right)$$

$$=\sum_{n=0}^{\frac{N}{2}-1}\left[\left(\left(x(n)-x\left(n+\frac{N}{2}\right)\right)W_N^n\right)W_{\frac{N}{2}}^m\right]$$

$$=DFT_{N/2}\left[\left(x(n)-x\left(n+\frac{N}{2}\right)\right)W_N^n\right] \qquad \text{II .10}$$

La Figure II .3 représente le schéma d'une structure butterfly DIF

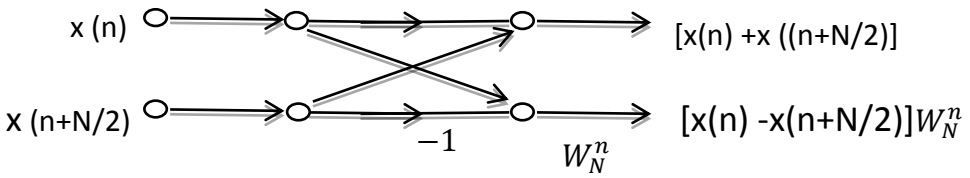


Figure (II .3**) :** Schéma d'une structure butterfly DIF

La Figure (II .4) représente la structure de l'algorithme Radix-2 DIF pour N=8 :

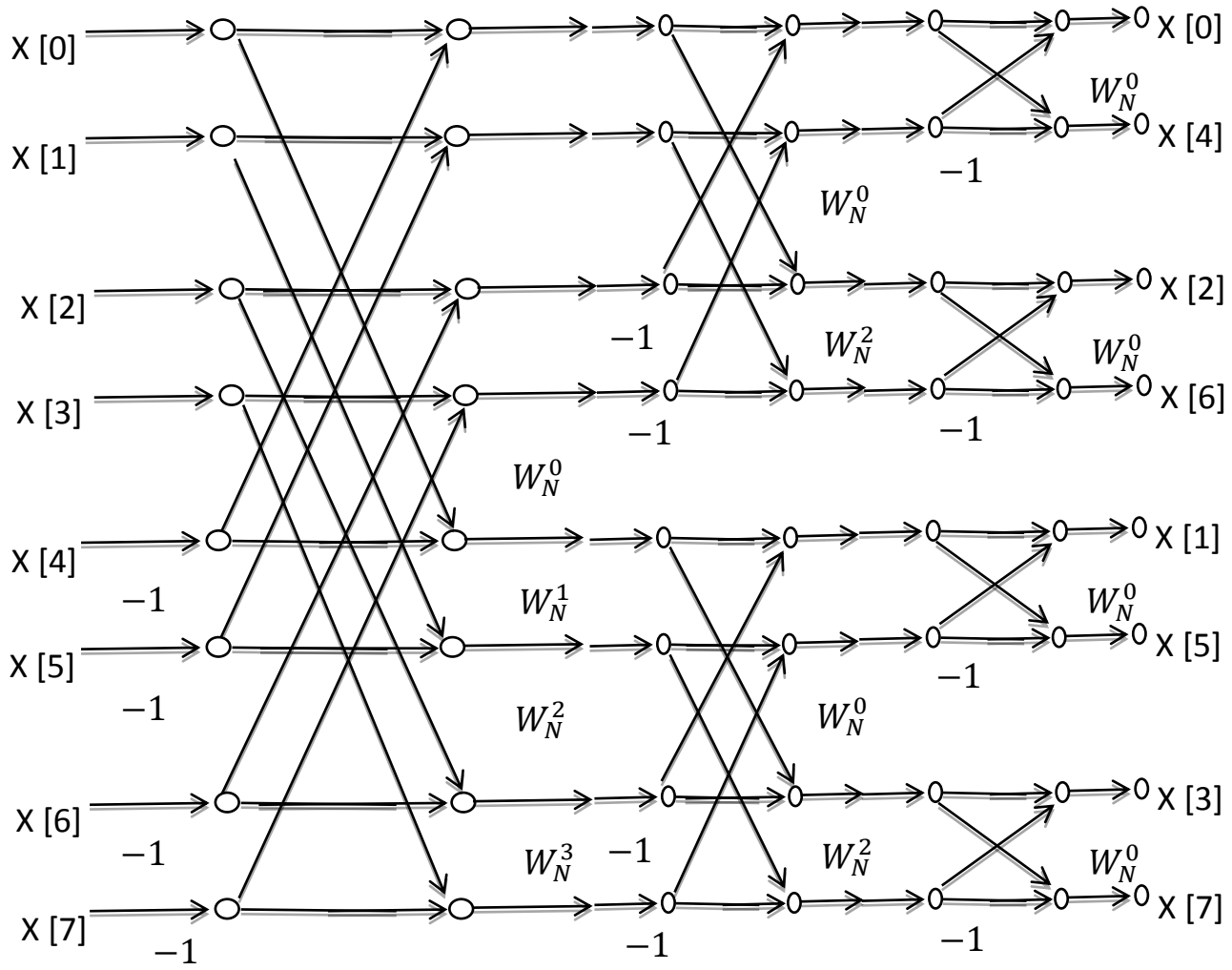


Figure (II .4**)** : Diagramme de la FFT à base de Butterfly Radix-2 de type DIF pour N=8 [6].

Une FFT Radix-2 sur N point utilise *N/2* lignes et *log2 N* colonnes de papillons, ce qui fait au total *N*log2(N) additions complexes et N/2log2 N multiplications complexes. Ce qui se compare avantageusement aux additions et multiplications complexes de la DFT. Sur la Figure (II.4), on remarque que la séquence de points en entrée est en ordre naturel, mais que la sortie n'est pas ordonnée. L'inverse se produit pour l'algorithme DIT Figure (II.2), Dans la littérature, cet ordre non naturel est nommé en anglais « Bit reverse order ».

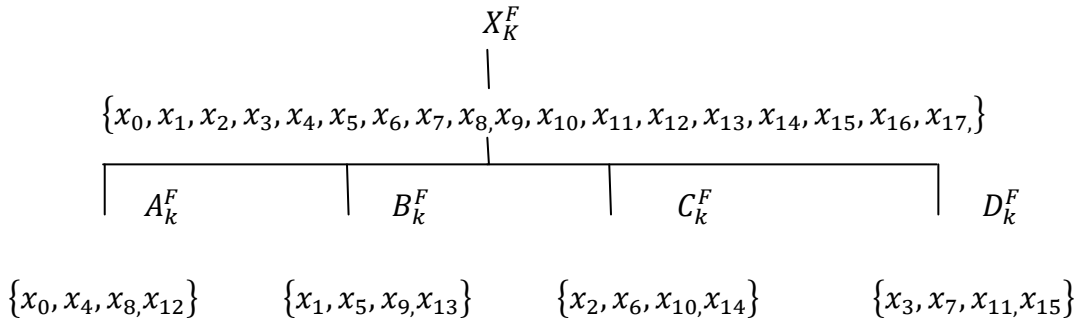
Le Radix-2 est l'algorithme FFT le plus simple à l'implémentation. Il apparaît dans plusieurs articles de l'IEEE, grâce à sa simple architecture. Plusieurs travaux sont effectués sur cet algorithme afin de trouver des méthodes plus élaborées pour réduire le nombre de multiplications et d'additions complexes.

III.2) – Radix-4 :

L'algorithme FFT radix-4 est une décomposition de l'algorithme de transformée de Fourier discrète en 4 transformées indépendantes de taille N/4., où quatre factorise N. Il vise à réduire la complexité multiplicative de la DFT de 25% par rapport au Radix-2. À condition que la taille de la FFT soit une puissance de 4. On peut tirer profit des multiplications triviales par 1,-1, j et –j qui n'ont pas à être réalisées par une vraie multiplication.

III.2.1)- Décimation temporelle(DIT) :

Cet algorithme est basé sur la décomposition de la séquence de N point en quatre séquences de N/4 point (x (4n), x (4n+1), x (4n+2), x (4n+3), n = 0, 1, ..., N/4-1). [7]



L'équation (II.1) peut être exprimée par :

$$X_k^F = \sum_{n=0}^{\frac{N}{4}-1} x_{4n} W_N^{4nk} + \sum_{n=0}^{\frac{N}{4}-1} x_{4n+1} W_N^{(4n+1)k} + \sum_{n=0}^{\frac{N}{4}-1} x_{4n+2} W_N^{(4n+2)k} + \sum_{n=0}^{\frac{N}{4}-1} x_{4n+3} W_N^{(4n+3)k} \quad \text{II.11}$$

$$X_k^F = \left(\sum_{n=0}^{\frac{N}{4}-1} x_{4n} W_N^{4nk}\right) + W_N^k \left(\sum_{n=0}^{\frac{N}{4}-1} x_{4n+1} W_N^{4nk}\right) + W_N^{2k} \left(\sum_{n=0}^{\frac{N}{4}-1} x_{4n+2} W_N^{4nk}\right) + W_N^{3k} \left(\sum_{n=0}^{\frac{N}{4}-1} x_{4n+3} W_N^{4nk}\right) \quad \text{II.12}$$

Elle peut être exprimée comme :

$$X_k^F = A_k^F + W_N^k B_k^F + W_N^{2k} C_k^F + W_N^{3k} D_k^F, k = 0,1, \dots, N-1 \quad \text{II.13}$$

Ou A_k^F,B_k^F, C_k^F, et D_k^F sont des DFT a N/4 point. Donc ils sont périodiques avec période N/4

$$X_{k+\frac{N}{4}}^F = A_k^F + W_N^{k+\frac{N}{4}} B_k^F + W_N^{2\left(k+\frac{N}{4}\right)} C_k^F + w_N^{3\left(k+\frac{N}{4}\right)} D_k^F \quad \text{II.14}$$

$$X^F_{k+\frac{N}{2}} = A^F_k - JW^k_N B^F_k - W^{2K}_N C^F_k + jW^{3k}_N D^F_k,\ k=0,1\ldots\frac{N}{4}-1 \qquad \text{II.15}$$

De même

$$X^F_{k+\frac{3N}{4}} = A^F_k - W^k_N B^F_k + W^{2K}_N C^F_k - W^{3k}_N D^F_k, k = 0,1,\ldots,\frac{N}{4}-1 \qquad \text{II.16}$$

$$X^F_{k+\frac{3N}{4}} = A^F_k + JW^k_N B^F_k - W^{2K}_N C^F_k - jW^{3k}_N D^F_k \quad k=0,1\ldots,\frac{N}{4}-1 \qquad \text{II.17}$$

Les équations (II.14) à (II.17) peuvent être exprimées sous la forme de matrice comme suivant :

$$\begin{bmatrix} X^F_K \\ X^F_{K+N/4} \\ X^F_{K+N/2} \\ X^F_{K+3N/2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} A^F_K \\ W^k_N B^F_k \\ W^{2K}_N C^F_k \\ W^{3k}_N D^F_k \end{bmatrix}$$

La figure II.5 représente le schéma d'une structure de Butterfly Radix-4 de type DIT

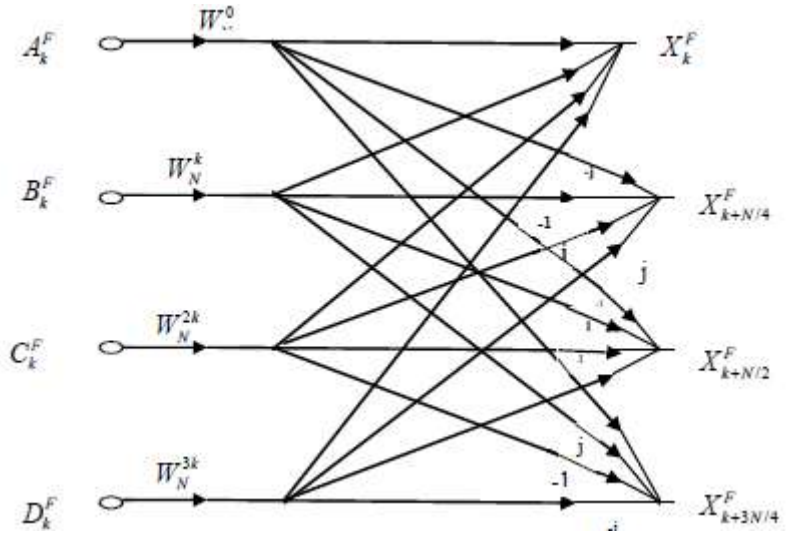


Figure (II.5**)**: Organigramme du Butterfly Radix-4 de type DIT

III.2.2) – Décimation fréquentielle (DIF) :

Partant de l'expression générale de la DFT [18] :

$$X^F_K = \sum_{N=0}^{N-1} x_n\, W^{nk}_N \quad , k = 0,1,\ldots N-1$$

$$X^F_K = \sum_{n=0}^{\frac{N}{4}-1} x_n W^{nk}_N + \sum_{n=\frac{N}{4}}^{\frac{N}{2}-1} x_n W^{nk}_N + \sum_{n=\frac{N}{2}}^{\frac{3N}{4}-1} x_n W^{nk}_N + \sum_{n=\frac{3N}{4}}^{N-1} x_n W^{nk}_N \qquad \text{II.18}$$

$X_K^F = \mathrm{I} + \mathrm{II} + \mathrm{III} + \mathrm{IV}$

Posons n=m+N/4 dans *II*, n=m+N/2 dans *III* et n=m+3N/4 dans*IV* on a :

$$\sum_{n-\frac{N}{4}}^{\frac{N}{2}-1} x_n W_N^{nk} = \sum_{m=0}^{\frac{N}{4}-1} x_{m+\frac{N}{4}} w_N^{\left(m+\frac{N}{4}\right)k} = w_N^{\frac{N}{4}k} \sum_{n=0}^{\frac{N}{4}-1} x_{n+\frac{N}{4}} W_N^{nk} = (-j)^k \sum_{n=0}^{\frac{N}{4}-1} x_{n+\frac{N}{4}} W_N^{nk} \quad \text{II.19}$$

$$\sum_{n-\frac{N}{2}}^{\frac{3N}{4}-1} x_n W_N^{nk} = \sum_{m=0}^{\frac{N}{4}-1} x_{m+\frac{N}{4}} w_N^{\left(m+\frac{N}{2}\right)k} = w_N^{\frac{N}{2}k} \sum_{n=0}^{\frac{N}{4}-1} x_{n+\frac{N}{2}} W_N^{nk} = (-1)^k \sum_{n=0}^{\frac{N}{4}-1} x_{n+\frac{N}{2}} W_N^{nk} \quad \text{II.20}$$

De même $\sum_{n=3N/4}^{N-1} x_n W_N^{nk} = (-j)^k \sum_{n=0}^{\frac{N}{4}-1} x_{n+\frac{3N}{4}} W_N^{nk}$ II.21

Remplacent (II.19),(II.20) et (II.21) dans (II.18) :

$$X_K^F = \sum_{n=0}^{\frac{N}{4}-1} \left[x_n + (-j)^k x_{n+\frac{N}{4}} + (-j)^k x_{n+\frac{N}{2}} + (j)^k x_{n+\frac{3N}{4}} \right] W_N^{nk} \quad \text{II.22}$$

Remplacent K=4m dans (II.22), (m=0, 1, …, N/4 − 1) :

$$X_{4m}^F = \sum_{n=0}^{\frac{N}{4}-1} [x_n + x_{n+N/4} + x_{n+N/2} + x_{n+3N/4}\ ]\ W_{N/4}^{nm} \quad \text{II.23}$$

X_{4m}^F *est une DFT des* $\frac{N}{4}$ *points,*

Pour k=4m+1 dans (II.22) on obtient :

$$X_{4m+1}^F = \sum_{n=0}^{\frac{N}{4}-1} x_n w_N^{n(4m+1)} + \sum_{n=N/4}^{\frac{N}{2}-1} x_n w_N^{n(4m+1)} + \sum_{n=N/2}^{\frac{3N}{4}-1} x_n w_N^{n(4m+1)} + \sum_{n=\frac{3N}{4}}^{N-1} x_n w_N^{n(4m+1)}$$

X_{4m+1}^F=V+VI+VII+VIII

V, VI, VII et VIII représente les quatre sommations de l'équation (II.24).

Posons n=m+N/4 dans VI, n=m+N/2 dans VII et n=m+3N/4 dans VIII, on obtient :

$$X_{4m+1}^F = \sum_{n=0}^{\frac{N}{4}-1} \left(\left[x_n + j x_{n+\frac{N}{4}} + x_{n+\frac{N}{2}} + j x_{n+\frac{3N}{4}} \right] W_N^n \right) W_N^{nk} \quad \text{II.25}$$

X_{4m+1}^Fest aussi une DFT de N/4 points : $\left(\left[x_n - j x_{n+\frac{N}{4}} + x_{n+\frac{N}{2}} + j x_{n+\frac{3N}{4}} \right] W_N^n \right)$

Pour k=4m+2 on obtient :

$$X_{4m+2}^{F} = \sum_{n=0}^{\frac{N}{4}-1} \left(\left[x_n + x_{n+\frac{N}{4}} + x_{n+\frac{N}{2}} + x_{n+\frac{3N}{4}} \right] W_N^{2n} \right) W_{N/4}^{nm} \qquad \text{II. 27}$$

X_{4m+1}^{F} est aussi une DFT de N/4 points : $\left(\left[x_n - x_{n+\frac{N}{4}} + x_{n+\frac{N}{2}} + x_{n+\frac{3N}{4}} \right] W_N^{2n} \right)$

Pour k=4m+3 on obtient :

$$X_{4m+3}^{F} = \sum_{n=0}^{\frac{N}{4}-1} \left(\left[x_n + jx_{n+\frac{N}{4}} - x_{n+\frac{N}{2}} - jx_{n+\frac{3N}{4}} \right] W_N^{3n} \right) W_{N/4}^{nm} \qquad \text{II.28}$$

X_{4m+1}^{F} est une DFT de $\frac{N}{4}$ points : $\left(\left[x_n + jx_{n+\frac{N}{4}} - x_{n+\frac{N}{2}} jx_{n+\frac{3N}{4}} \right] W_N^{3n} \right)$

En conclusion, une FFT Radix-4 DIT est développée à base des DFT N/4-points.

Chaque DFT de N/4 points est encore décompose sous forme de 4 DFT de N/4 points chacune. L'opération se répète jusqu'à l'obtention des DFTs à 4 points.

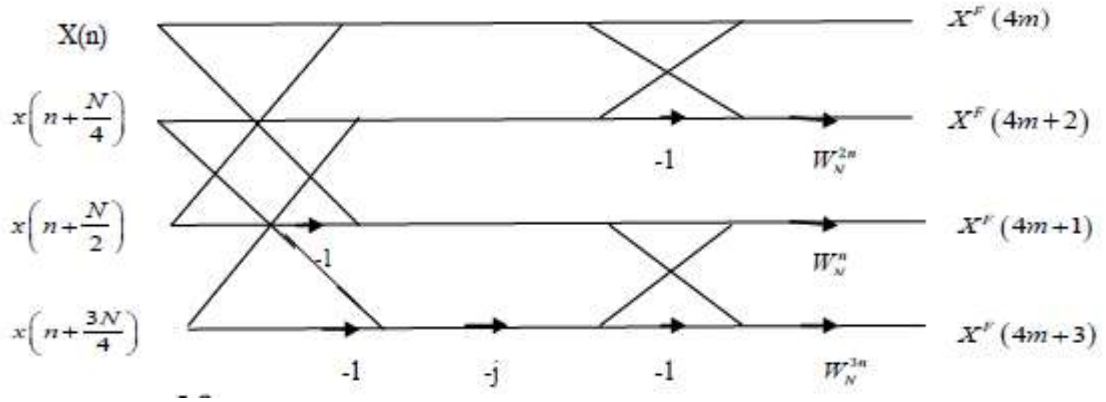


Figure (II.6) : Schéma d'une structure de Butterfly Radix-4 de type DIF.

III.3)- Bit reversal

Dans les mathématiques appliquées, une permutation de renversement de bit est une permutation d'une séquence de *N* éléments, où $N = 2^n$ est une puissance de deux. Il est défini en indexant les éléments de la séquence par les nombres de 0 à *N* - 1, puis en inversant les représentations binaires de chacun de ces nombres (rembourrés de sorte que chacun de ces nombres binaires a une longueur exactement *n*). Chaque élément est ensuite mappé à la nouvelle position donnée par cette valeur inversée. La permutation de renversement de bit est une involution, alors la répétition de la même permutation revient deux fois à la commande initiale sur les éléments. [9] Par exemple pour N = 8:

Tableau (II.2) : Bit reversal pour N=8.

Séquence monotone	***Codé en binaire***	***Bit inverse***	*Succession de bit-reversé*
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

IV)- Architectures d'implémentation de L'FFT en pipeline

L'architecture pipeline est l'une des architectures les plus populaires et les plus adaptées pour les applications du traitement de signal où le haut débit de données est une condition dominante.

Depuis, l'architecture pipeline offre un bon compromis complexité matérielle/taux de traitement de données pour les systèmes de communications à haut débit la figure II.7 représente une architecture pipeline de base [6]

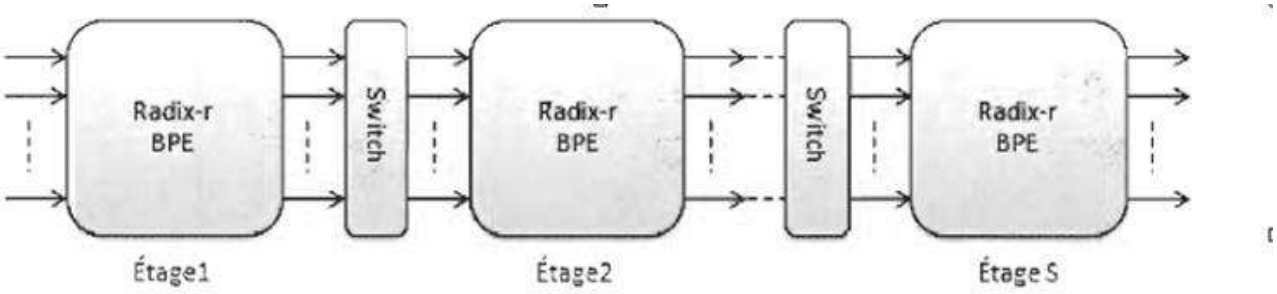


Figure (II.7) : architecture pipeline de base

Comme le montre la figure II.7, l'architecture se compose d'un nombre de BPE (Butterfly Processor Elément) qui sont les unités de calcul des butterfly et d'un nombre de modules 'Switch'. Ces derniers réorganisent les données entre les BPE et peuvent contenir des modules de délais. À chaque étage du pipeline, un BPE, prend *r* données de valeurs complexes, effectue son calcul, puis génère *r* données intermédiaires à la sortie. On a donc un débit de données *r* fois la fréquence d'horloge. Par exemple avec un BPE Radix-2 fonctionnant à *R* MHz, les données sont traitées à une fréquence de *2R* MHz, puisque deux données sont traitées au cours de chaque période d'horloge. De même pour un Radix-4 fonctionnant à *R* MHz, la

fréquence de traitement sera de *4R* MHz. Donc plus le Radix du BPE est grand, plus on obtient un meilleur débit binaire. Il existe plusieurs structures d'architecture pipeline, on peut les divisés en trois types :

- Architecture SDF *(Single-path Delay Feedback);*
- Architecture SDC (Single-path Delay Commutator);
- Architecture MDC *(Multi-path Delay Commutation).*

IV.1)- Architecture Radix-r MDC :

L'architecture pipeline Radix-r MDC *(Multi-path Delay Commutator)* estl'architecture pipeline la plus populaire et la plus utilisée pour l'implémentation des systèmes de communications qui demandent un haut débit binaire, Plusieurs implémentations existent. On trouve le R2MDC *(Radix-2 Multi-path Delay Commutator)*, le R4MDC, le R8MDC et d'autres architectures plus élaborées comme le R23MDC et le MC-MRMDC *(Multi-Channel- Mixed Radix Multi-path Delay Commutator).* Cette architecture utilise plus de surface silicium que les autres architectures pipeline, mais reste très attrayante et reste l'implémentation pipeline la plus classique grâce à son compromis vitesse/surface silicium.

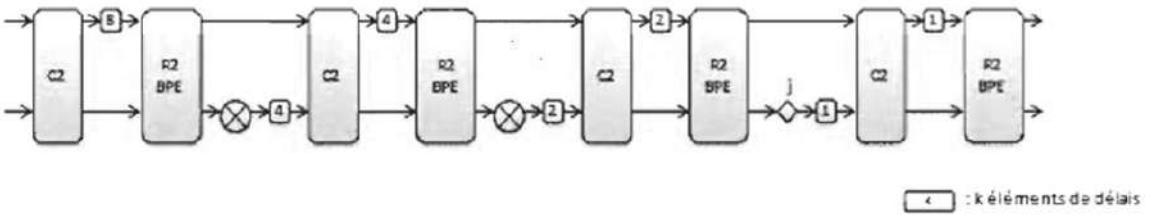


Figure (II.8) : Architecture R2MDC (N=16 point)

IV.2) Architecture Radix-r(SDF) :

L'architecture SDF (Single-path Delay Feedback) est l'architecture la plus appropriée pour les systèmes OFDM au niveau de la complexité matérielle. Puisqu'il existe plusieurs algorithme Radix-r, où r est le nombre d'entrées du BPE, on retrouve plusieurs architectures SDF, tel que la R2SDF (Radix-2 Single-path Delay Feedback), la R4SDF, la R8SDF et la R2SDF. L'architecture SDF est considérée comme la meilleure approche afin de calculer la FFT pour un système OFDM à un seul canal parce qu'elle requière le moins de ressources matérielles et qu'elle possède une seule entrée et sortie ce qui la rend très appropriée pour un

système SISO-OFDM (Single Input Single Output Orthogonal Frequency Division Multiplex). [6]

La figure II.8, représente Architecture R2SDF (N = 8 points) ;

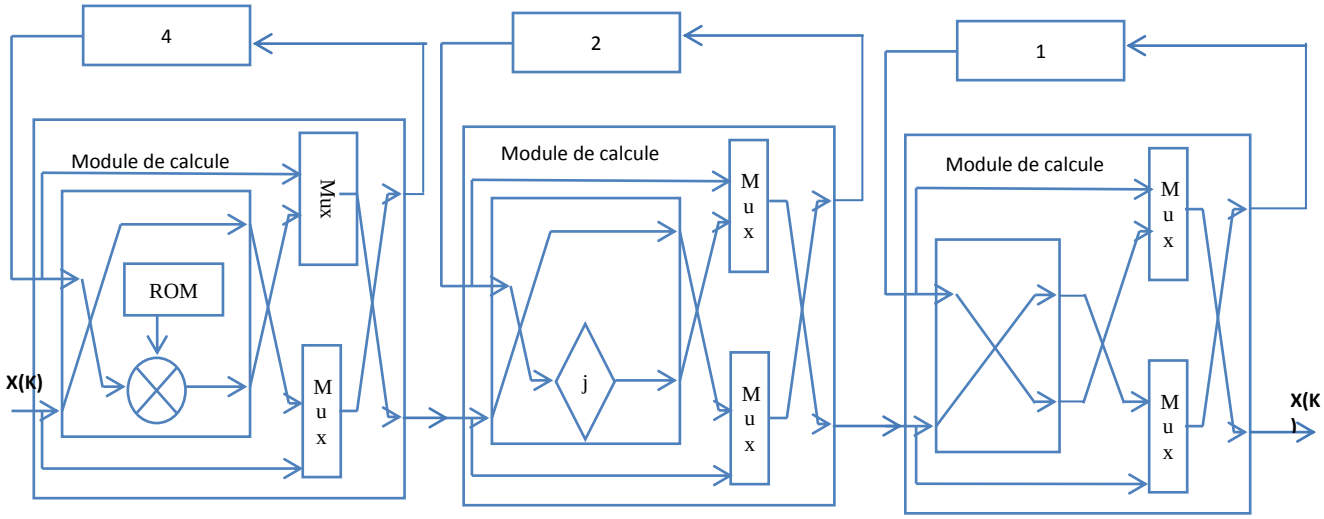


Figure II.9 Architecture R2SDF (N = 8 points)

IV.3) –Architecture Radix-r SDC :

L'architecture pipeline Radix-r SDC utilise le même nombre de multiplieurs complexes que l'architecture pipeline SDF grâce à son chemin unique d'accès. Elle utilise par contre un commutateur de délais qui possède une seule entrée et plusieurs sorties. Le nombre de sorties du commutateur varie selon le nombre d'entrée du BPE. La figure II.8 représente une architecture pipeline R4SDC.

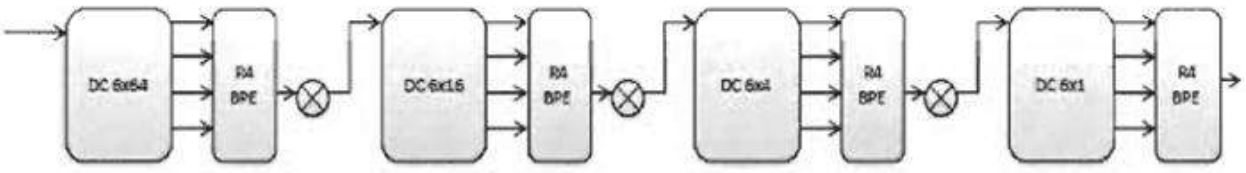


Figure (II.10) : Architecture R4SDC *(N* = 256 points)

À chaque coup d'horloge, le commutateur de délais injecte les données au BPE. Après avoir effectué les calculs, le BPE injecte le résultat dans le chemin unique d'accès pour l'étage suivant. L'architecture SDC requière un système de contrôle plus complexe que les autres architectures pipeline. Et comme l'architecture SDF, elle opère à un faible débit binaire

puisqu'elle possède un seul chemin d'accès d'entrée et de sortie. Pour un système MIMO-OFDM, elle a aussi donc besoin de plusieurs coprocesseurs pour chaque canal du système. Ceci augmente la complexité matérielle de son implémentation.

V)-Comparaison :

Tableau (II.3) : Ressources matérielles selon l'architecture en pipeline utilisée [9].

Architecture		#Multiplieurs	#additionneurs	Mémoire	Contrôle	$f^{(1)}op$
		Complexes	Complexes			
R2	MDC	$2\log_4 N-1$	$4\log_4 N$	3N/2-2	Simple	$f^{(2}Echant$
R2	SDF	$2\log_4 N-1$	$4\log_4 N$	N-1	Simple	*f Échant*
R4	MDC	$3(\log_4 N-1)$	$8\log_4 N$	5N/2-2	Simple	*f Échant*
R4 SDF		$\text{Log}_4 N-1$	$8\log_4 N$	N-1	Moyen	*f Échant*
R4	SDC	$\text{Log}_4 N-1$	$3\log_4 N$	2N-2	Complexe	*f Échant*
$R2^2$	SDF	$\text{Log}_4 N-1$	$4\log_4 N$	N-1	Simple	*f Échant*

(1) Fréquence d'opération. (2) Fréquence d'échantillonnage.

L'approche radix-r MDC est ***r*** fois plus rapide que l'approche radix-r SDF. Toutefois en termes de mémoires, l'approche SDF est plus économe que l'approche MDC (qui souffre d'un surplus de surface par rapport à l'approche SDF). En termes de ressources, les architectures Radix- MDC et Radix- SDF sont plus performantes que n'importe quelle autre approche. Elles nécessitent toujours le moins de ressources arithmétiques et de mémoires. Ceci est dû entre autres au fait qu'elles utilisent plusieurs multiplieurs par des valeurs constantes optimisées et caractérisées par une complexité inférieure. En termes de complexité de contrôle, les architectures pour le radix-4-SDC et split-radix sont les plus complexes à contrôler [7].

Tableau (II.4) : Taux d'utilisation de ressource des différentes architectures.

Architecture	Utilisation des multiplicateurs	Utilisation des additionneurs (en PE)	Utilisation des registres
Radix-2 SDF	50%	50%	100 %
Radix-4 SDF	75%	25%	100 %
Radix-2 MDC	50%	50%	50 %
Radix-4 MDC	25%	25%	25 %

VI-Conclusion :

Le présent chapitre a permis l'étude des algorithmes FFT pour le calcul de la DFT. L'algorithme FFT s'avère être efficace pour calculer la DFT car il réduit considérablement le nombre d'opérations arithmétiques par rapport à la DFT, de ce fait, la FFT devient indispensable dans les nouveaux systèmes de communications.

Pour l'implémentation de l'algorithme FFT, les architectures en pipeline sont les plus adaptées pour les communications numériques à haut début.

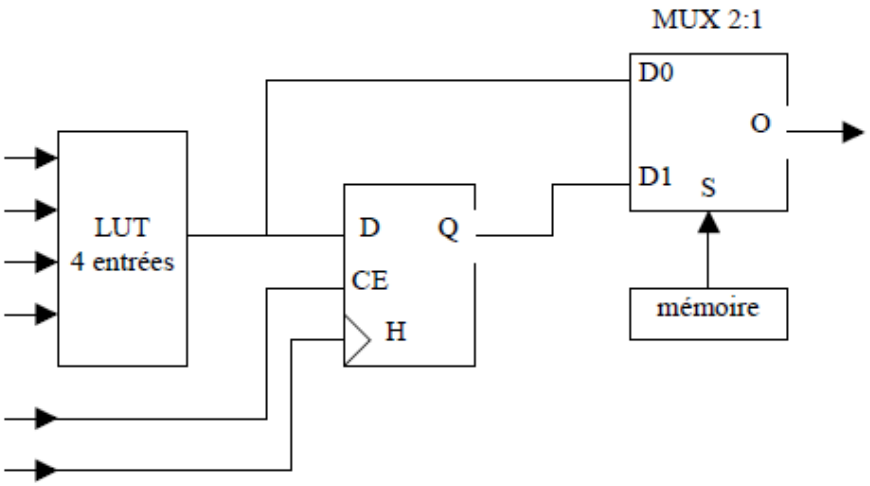
Les architectures pipeline conventionnelles sont : le SDF, MDC et SDC. Parmi les différents architectures FFT pipelines et parallèles nous avons choisi l'architecture SDF, elle possède un seul chemin d'accès et est considérée comme la meilleure approche permettant l'implémentation efficace de l'FFT.

I)- Introduction

Ce chapitre est consacré à la présentation des circuits de type FPGA. Nous commençons par la présentation de l'architecture de base des $FPGA_s$ puis nous présentons les avantages et les applications des circuits FPGA.Par la suite nous présentons une comparaison de FPGA avec les circuits$ASIC_s$. Enfin nous donnons quelque outil utilisé pour développer une application à base de circuit FPGA.

II)-Architecture des circuits FPGA : [6]

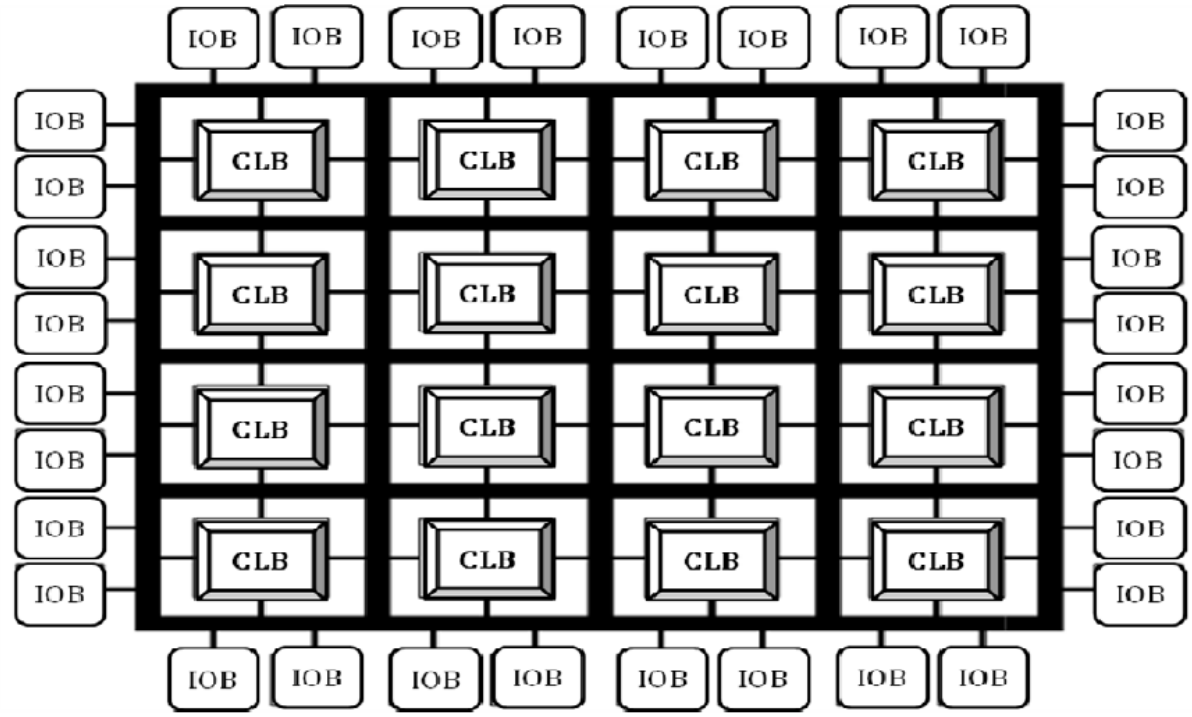
Lancer sur le marché en 1984 par la firme XILINX, le FPGA (Field Programmable GateArrays) est un circuit prédiffusé programmable. Le concept du FPGA est basé sur l'utilisation d'une LUT (LookUp Table) comme élément combinatoire de la cellule de base.
Une cellule LUT est l'élément qui permetde générer n'importe quelle fonction logique combinatoire de *n*variables d'entrées. Chez Xilinx, on appelle cela un générateur de fonction ou « Fonction Generator ». La figure suivante représente la cellule type de base d'un FPGA.



Figure(III.1) : cellule de base d'un circuit FPGA

Elle comprend une LUT 4 entrées et une bascule D (D Flip-Flop) et un Multiplexeur. La bascule D permet la réalisation de fonctions logiques séquentielles. La configuration du multiplexeur 2 vers 1 de sortie autorise la sélection des deux types de fonction, combinatoire ou séquentielle. Les cellules de base d'un FPGA sont disposées en lignes et en colonnes. Des lignes d'interconnexions programmables traversent le circuit, horizontalement et

verticalement, entre les diverses cellules. Ces lignes d'interconnexions permettent de relier les cellules entre elles, et avec les plots d'entrées/sorties. Les connexions programmables sur ces lignes sont réalisées par des transistors MOS dont l'état est contrôlé par des cellules mémoires SRAM. Ainsi, toute la configuration d'un FPGA est contenue dans des cellules SRAM.



Figure(III.2) : Architecteur interne d'un FPGA [6].

Tous les FPG Assont fabriqués en technologie CMOS Par rapport aux prédiffusés classiques, les interconnexions programmables introduisent des délais plus grands que la métallisation (environ 3 fois plus lents). Par contre, les cellules logiques fonctionnent à la même vitesse. Pour minimiser les délais de propagation dans un FPGA, il faut donc réduire le nombre de cellules logiques utilisées pour réaliser une fonction.

Par conséquent, les cellules logiques d'un FPGA sont plus complexes que celles d'un prédiffusé.

Les circuits FPGA du fabricant Xilinx utilisent deux types de cellules de base :

- Les cellules d'entrées/sorties appelés IOB (input output bloc),
- Les cellules logiques appelées CLB (configurable logic bloc). Ces différentes cellules sont reliées entre elles par un réseau d'interconnexions configurable.

II.1)-Les CLB (Configurable Logic Block)[7]

Les CLBs constituent le cœur du FPGA. Ce sont des cellules constituées d'éléments logiques programmables où l'on trouve des bascules (registres), des LUTs (Look-Up Table), des multiplexeurs. Chaque cellule est identique aux autres et peut être reliée à ses voisins par le biais de bus d'interconnexion. Ces cellules peuvent être utilisées pour créer des fonctions logiques complexes mais aussi comme éléments de stockage de variables. Au cours du temps et grâce à l'évolution de la technologie, l'architecture de ces cellules a évolué en complexité. Par ailleurs, les FPGAs intègrent de plus en plus de cellules sur une même surface. On trouve par exemple jusqu'à 400000 CLB appelés ALM (high performance Adaptation Logic Modules) dans les Stratix V de Altera et 1 956 000 CLBs pour les Virtex 7 de Xilinx.

II.2)-Les cellules d'entrée sortie (IOB) : [16]

Ce sont des cellules permettant l'acheminement des données de ou vers le circuit FPGA, ces cellules peuvent être configurées en entrée ou en sortie.

II.2.1)-Configuration en entrée :

Premièrement, le signal d'entrée traverse un buffer qui selon sa programmation peut détecter soit des seuils TTL ou soit des seuils CMOS. Il peut être routé directement sur une entrée directe de la logique du circuit FPGA ou sur une entrée synchronisée. Cette synchronisation est réalisée à l'aide d'une bascule de type D, le changement d'état peut se faire sur un front montant ou descendant. De plus, cette entrée peut être retardée de quelques nanosecondes pour compenser le retard pris par le signal d'horloge lors de son passage par l'amplificateur. Le choix de la configuration de l'entrée s'effectue grâce à un multiplexeur (program controlled multiplexer). Un bit positionné dans une case mémoire commande ce dernier.

II.2.2)-Configuration en sortie :

Nous distinguons les possibilités suivantes :

- inversion ou non du signal avant son application à l'IOB,
- synchronisation du signal sur des fronts montants ou descendants d'horloge,
- mise en place d'un " pull-up " ou " pull-down " dans le but de limiter la consommation des entrées sorties inutilisées,
- signaux en logique trois états ou deux états. Le contrôle de mise en haute impédance et la réalisation des lignes bidirectionnelles sont commandés par le signal de commande Out

Enable lequel peut être inversé ou non. Toutes ces possibilités permettent au concepteur de connecter au mieux une architecture avec les périphériques extérieurs.

II.3) -Les Interconnexions :

Les connexions internes dans les circuits FPGA sont composées de segments métallisés. Parallèlement à ces lignes, nous trouvons des matrices programmables réparties sur la totalité du circuit, horizontalement et verticalement entre les divers CLB. Elles permettent les connexions entre les diverses lignes, celles-ci sont assurées par des transistors MOS dont l'état est contrôlé par des cellules de mémoire vive ou RAM. Le rôle de ces interconnexions est de relier avec un maximum d'efficacité les blocs logiques et les entrées/sorties afin que le taux d'utilisation dans un circuit donné soit le plus élevé possible. Pour parvenir à cet objectif, Xilinx propose trois sortes d'interconnexions selon la longueur et la destination des liaisons. Nous disposons :

- d'interconnexions à usage général ;
- d'interconnexions directes ;
- de longues lignes.

III)- Les avantages et les applications de FPGA [14]

Les $FPGA_s$ sont utilisés dans diverses applications nécessitant de l'électronique numérique (télécommunications, aéronautique, transports…). Ils sont également utilisés pour le prototypage d'ASIC.

Les FPGA sont généralement plus lents et consomment davantage d'énergie que leur équivalent en ASIC (*Application Specific Integrated Circuit*). Cependant, ils ont plusieurs avantages :

- délai de mise sur le marché plus court, car ce sont des composants standards,
- temps de conception plus court, car on réutilise des fonctions de base dont la reconfiguration autorise une validation préalable moins stricte,
- coût inférieur pour de petites séries (moins de 10 000 unités). Avec l'évolution technologique, cette quantité tend à augmenter : en effet, le prix d'une puce est proportionnel à sa surface, qui diminue avec la finesse de gravure, tandis que les coûts initiaux pour fabriquer un ASIC (conception, tests, masques de gravure) sont en forte augmentation.

Il est parfois possible de transformer directement un FPGA en une version ASIC plus rapide, moins chère et consommant moins (car les matrices de routage sont remplacées par une couche de métallisation fixe).

Plusieurs FPGA modernes possèdent la possibilité d'être *reconfigurés* (on parle de configuration lorsqu'il s'agit de programmation du matériel) partiellement à la volée. Ceci permet d'obtenir des *systèmes reconfigurables* - par exemple une unité centrale dont les instructions changent dynamiquement en fonction des besoins.

Les FPGA modernes sont assez vastes et contiennent suffisamment de mémoire pour être configurés pour héberger un cœur de processeur ou un DSP, afin d'exécuter un logiciel. On parle dans ce cas de processeur softcore, par opposition aux microprocesseurs *hard-core* enfouis dans le silicium. Aujourd'hui, les fabricants de FPGA intègrent même un ou plusieurs cœurs de processeur « *hard-core* » sur un même composant afin de conserver les ressources logiques configurables du composant. Ceci n'exclut pas l'utilisation de processeur softcore possédant de nombreux avantages. On tend donc vers des « *Systems On Chip* », comme pour le microcontrôleur il y a quelques décennies, avec en plus de la logique configurable selon l'utilisateur. Si la mémoire du circuit FPGA est encore insuffisante pour exécuter des logiciels embarqués un peu complexes on doit avoir recours à des mémoires externes (ROM, RAM).

Les tableaux suivant représente une comparaison entre les circuits de type FPGA et les circuits de type ASIC. [10]

	Avantages	**Désavantages**
ASIC	• Grande capacité potentielle • Moins de silicium • Consommation plus faible • Plus grande fiabilité • S/B plus faible	• Investissement initial plus grand • Pas de flexibilité • Délai emportant entre la conception et la mise en marché
FPGA	• Grande flexibilité matérielle • Peut réutiliser des blocs matériels déjà faits et testés • Peut être plus performant	• Consommation plus grande • Surtout utilisé pour des applications en virgule fixe • Apprentissage plus long

Tableaux (III.1) : une comparaison entre FPGA et ASIC [10]

Les deux firmes qui dominant la fabrication des circuits FPGA aussi que les logiciels de développement dans la firme Xilinx et la firme Altera .

-XILINX : [16]

Xilinx a été fondée en 1984 par trois anciens employés de Zilog : Ross Freeman ; Bernie Vonderschmitt et Jim Barnett, dont le plan d'entreprise se résumait à la commercialisation de composants électroniques basé sur un concept alors nouveau : celui de la logique programmable.

La société a commercialisé son premier produit en 1985, le FPGA XC2064. Deux ans plus tard, elle ouvrait des bureaux de vente en Europe et au Japon. En 1990, Xilinx est introduite en bourse, et réalise en 2000 un chiffre d'affaires dépassant le milliard de Dollars.

- ALETRA :

Fondé en 1983. La société a développé le premier dispositif logique reprogrammable (PLD) en1984. Altera a augmenté sa direction de technologie en1988 avec l'architecture de max et, en 1992, avec l'architecture basée sur du les (LUT)Il a récemment renforcé encore autre avec l'introduction de plus nouveau, des architectures plus puissantes et plus efficaces, le système de développement deQuartus II, et une offre étendue d'IP. Le partenaire stratégique d'Altera est TSMC.

Les deux sociétés offrent une gamme étendue de circuit $FPGA_s$, avec des capacités approximativement équivalentes elles offrent qu'aussi des logiciels de développement, avec beaucoup de blocs d'IP (propriété intellectuelle) pour les besoins spécifiques des ingénieurs et de chercheurs dans le domaine de conception à base de circuit FPGA.

Le tableau suivant donne quelques exemples de circuit FPGA développé par les deux firmes.

	Aletra	**xlinx**
1997		APEX
1998		Virtex
2000		Spartax II
2005	Stratix II GX	Virtex-4 FX & Spartan-3A
2008	Stratix IV	Virtex-5 TXT & Spartan-6
20013	Stratix 10 & Cyclone 10	

Tableaux (III.2) : exemple de quelque circuit FPGA.

IV)-Outils De Développement

IV.1)-L'outil de conception ISE 14.3 : [6]



Le logiciel Xilinx ISE est un outil de conception de circuit pour FPGA de Xilinx. Ce logiciel permet essentiellement d'effectuer les différentes étapes propres à la synthèse de circuits numériques sur FPGA. Il est alors possible d'en faire l'implémentation sur les différentes familles de puces fournies par Xilinx.

L`outil de conception ISE intègre plusieurs logiciel nécessaire au développement des applications sur circuit FPGA.

Les logiciels les plus communément intégré avec ISE Sant : [16]

- ✓ Xilinx EDK : Environnement de développement intégré ciblant les processeurs soft cores (microblaze) et hard cores (PowerPC) intégrés au FPGA. Intégrant lui-même deux sous ensembles, XPS pour l'intégration d'IP et SDK pour le développement de programme embarqué en langage C/C++ pour la cible ainsi créée.
- ✓ XilinxChipScope Pro : Outil de déboguage *temps-réel* de FPGA
- ✓ Impact : Pour le chargement du code compilé sur les cibles FPGA via les interfaces JTAG
- ✓ System Generator : pour le développement à haut niveau d'abstraction basé sur Matlab Simulink
- ✓ Logicore : Pour la génériond'IP

IV.2)-System Generator:[7]

System Generator est un outil qui permet de développer, d'une façon simple et rapide des applications sur FPGA. Constituant une interface entre MATLAB/Simulink et ISE de Xilinx, il permet de concevoir et de simuler des systèmes numériques dans un environnement graphique à partir de blocs IP (Intellectual Property) et des codes MATLAB.

Des blocs DSP sont fournis dans le jeu de blocs Xilinx DSP pour Simulink. Ces,blocs comprennent les blocs de construction communs tels que les additionneurs, les multiplicateurs et les registres. Sont également inclus des blocs de construction complexes tels que blocs Forward Error Correction, FFT, filtres ...

Dans System Generator sont inclus des blocs Mcode qui permettent l'utilisation de programmes MATLAB pour la modélisation et l'implémentation des opérations de contrôle simple.

System Generator fournit aussi un bloc estimateur de ressources qui permet de faire une estimation rapide des ressources nécessaires pour l'implémentation d'un design sur FPGA.

V)- Applications des FPGAs [10]

Les FPGA sont utilisés dans de nombreuses applications, on en cite dans ce qui suit quelques-unes:

- ✓ Prototyping de nouveaux circuits ;
- ✓ Fabrication de composants spéciaux en petite série ;
- ✓ Adaptation aux besoins rencontres lors de l'utilisation ;
- ✓ Systèmes de commande à temps réel ;
- ✓ DSP (Digital Signal Processor) ;
- ✓ Imagery medical.
- ✓ contrôle d'avions et métros.
- ✓ radar, communication protégée, la détection ou la surveillance

- Quelques « familles »FPGA XILINX :

Famille	portes	E /S	bascules	RAM bits	CLB
XC2000	0.6 - 1.5K	58 - 74	122 - 174	0	64-100
XC3000	1.3 - 9K	64 - 176	256 - 1320	0	64-484
XC4000	2 - 180K	80 - 448	360 - 7168	3K - 100K	100-3136
XC6200	9K - 100K	192 - 512	2K - 16K	0	2K - 16K
XC4000XV	75K - 500K	448	10K - 20K	131K - 270K	2.5K-5K
SPARTAN	2K - 200K	77 – 284	30 - 2016	16K - 57K	238 – 5292
VIRTEX	58K - 1M	*180 - 512*	1728 - 27K	32K - 131K	384-6144
VIRTEX2	40K - 10M	88 - 1108	256 - 61K	72K - 3.4M	64-15360

Tableau (III.3) : Quelques « familles »FPGA de XILINX [13].

VI)-Les étapes d'implémentation

Il contient deux étapes :

VI.1)-Spécification de design et Développement de design :

-Le nombre de broche d'entrée-sortie et leur localisation dans la puce FPGA.

- La spécification de la fréquence d'horloge du système.
- La spécification de la mémoire requise pour l'application.
- Spécification de la méthodologie de design (outil de développement utilisé.
- La saisie du circuit (codage RTL, VHDL, Verilog …), Graphique (Machine à états) et Saisie HDL (Hardware Description langage) .

VI.2)- Simulation(FPGA) :

La simulation est la partie la plus important du flux de conception sur FPGA. A ce point, la conception doit être vérifiée. Sans souci du type de conception, le but de la simulation est de vérifier que la conception effectue les fonctions requises. Typiquement, la simulation peut être effectuée à trois endroits dans le processus de développement sur la sortie de la conception, la synthèse et après la phase d'implémentation. La plupart des temps, la simulation est effectuée uniquement sur la conception et non pas sur le netlist ou un fichier produit par synthèse ou après la phase d'implémentation. Une vérification de la conception peut également être effectuée en utilisant l'équipement de laboratoire, tels que des analyseurs

logique, cependant, cela peut prendre plus de temps et d'une manière moins efficace, surtout pour les nouveaux conceptions. La vérification de laboratoire est moins flexible, ces erreurs peuvent endommager le matériel. La simulation n'est pas la seule méthode pour vérifier une conception de FPGA. Même si la simulation n'est pas nécessaire, elle ne devrait jamais être complètement omise de la phase de développement de FPGA, en particulier pour les nouvelles conceptions. Il semble qu'il n'est pas nécessaire de simuler petites et simples modifications, mais ce sont ceux qui peuvent causer des problèmes les plus importants [6].

VI.3)-Synthèse et implémentation du projet :

La synthèse d'un circuit consiste à traduire la description du circuit en blocs disponibles dans la technologie utilisée. Par exemple, pour un circuit décrit avec un schéma et qui doit être réalisé sur un FPGA, le processus de synthèse convertit et regroupe les portes logiques du schéma en composantes réalisables sur le FPGA choisi. L'implémentation du circuit est divisée en quatre sous étapes :

• la transformation (mapping) : regrouper les composantes obtenues lors de la synthèse dans des blocs spécifiques du FPGA ;

• la disposition (placement) : choisir des endroits spécifiques sur le FPGA où disposer les blocs utilisés, et choisir les pattes du FPGA correspondant aux ports d'entrée et de sortie ;

• le routage (routing) : établir des connexions électriques entre les blocs utilisés ;

• la configuration (configuration) : convertir toute cette information en un fichier pouvant être téléchargé sur le FPGA pour le programmer.

VII)-Conclusion :

Ce chapitre est dédié à la présentation des circuits FPGA qui d'inscrivent au sommet de l'évolution des composants logiques et le besoin croissant de composants plus performants, plus économiques et disponibles en grandes quantités.

Nous avons commencé par la présentation de l'architecture de base qui est basée sur trois éléments suivant : les CLB ; les IOB et les interconnexions. Nous avons en suite, donnée les avantages des circuits FPGA qui sont nombreuses ce qui rend ce type de circuit très utilisé dans tous les domaines, et spécialement dans le domaine des communications numériques.

Nous avons vue aussi que le développement d'une application à base de FPGA est facilite par l'introduction de plusieurs logiciels de conception. Enfin nous avons présenté les différentes étapes de développement d'une application à base de circuit FPGA.

I)-Introduction

Il existe plusieurs architectures pour réaliser l'algorithme FFT, comme il a été expliqué dans le chapitre 2. Ce chapitre est dédié à présenter notre simulation et le résultat de l'implémentation de l'architecture SDF pour le calcul parallèl de la DFT de deux séquences indépendantes. L'architecture est conçu pour la taille de la DFT allant de 8 jusqu'à 1024 est est implémenté sur deux circuits FPGA de type SPARTAN6 et Virtex 7.

Ce chapitre se compose essentiellement de trois parties. Dans la première partie nous allons donner une description détaillée des architectures SDF utilisées. La deuxième partie présente l'organigramme décrivant les différentes étapes de conception de notre projet. Ce chapitre se termine par les résultats finaux de simulation et de l'implémentation.

II)- L'architecture implémentée

Dans une architecture SDF conventionnelle ; pour implémenter un FFT de taille $N=2^n$ on utilise n étages en pipeline dont chaque étage effectue les opérations de l'étage qui lui correspond dans le SFG (Signal Flow Graphe). Par exemple pour un FFT de taille 8 on utilise 3 étages en pipeline. Le premier étage réalise les opérations de l'étage 1 du SFG de la figure IV.1. De même, le deuxième étage de l'architecture réalise les opérations de l'étage 2 du SFG et la même chose pour le troisième étage

Dans l'architecture SDF chaque étage ; à l'exception du dernier et de l'avant dernier ; contient un multiplicateur complexe. Ce multiplicateur est utilisé pour multiplier les sorties du soustracteur par les coefficients W_N^{nk} correspondants. Les multiplicateurs sont utilisés uniquement pendant 50% du temps de traitement puisque uniquement les sorties du soustracteur sont multipliées.

Le taux d'utilisation des multiplicateurs peut être augmenté à 100% par la mise en commun de multiplicateur pour deux étages SDF identiques en parallèle pour le calcul parallèle de la DFT de deux séquences indépendantes.

Pour deux étages identiques en parallèle on utilise un seul multiplicateur qui fonctionne comme suivant :

- Durant la première moitié du temps de traitement les sorties du soustracteur de la séquence supérieure sont d'abord multipliées par les coefficients W_N^{nk} puis injectées dans les

Delay, alors que les sorties du soustracteur de la séquence inférieure sont directement injectées dans les Delay (sans être multipliées par les coefficients W_N^{nk}).

- Durant la deuxième moitié du temps de traitement les sorties des Delay de la séquence supérieure sont acheminées vers l'étage suivant, alors que les sorties du Delay de la séquence inférieure sont d'abord multipliées par les coefficients W_N^{nk} puis acheminées vers l'étage suivant.

La figure suivante donne l'architecture détaillée pour une FFT de taille 8.

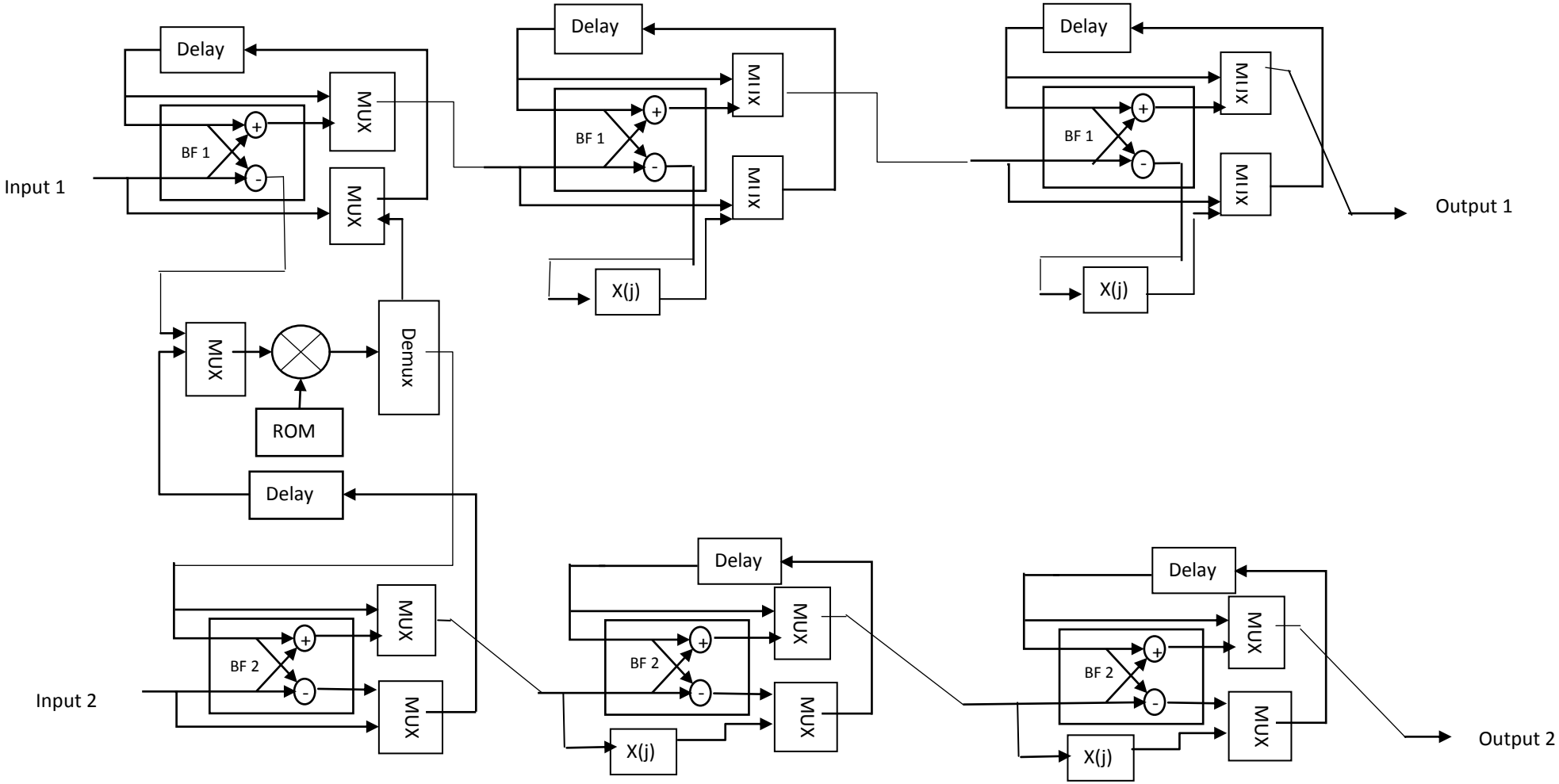


Figure (VI.1) : une architecture SDF radix-2 à deux voies pour l'FFT de taille 8.

III)-l'organigramme de développement notre projet

La figure VI.2 présente un organigramme simplifié qui résume les étapes de conception de notre projet. Ces étapes seront expliquées dans les sections suivantes.

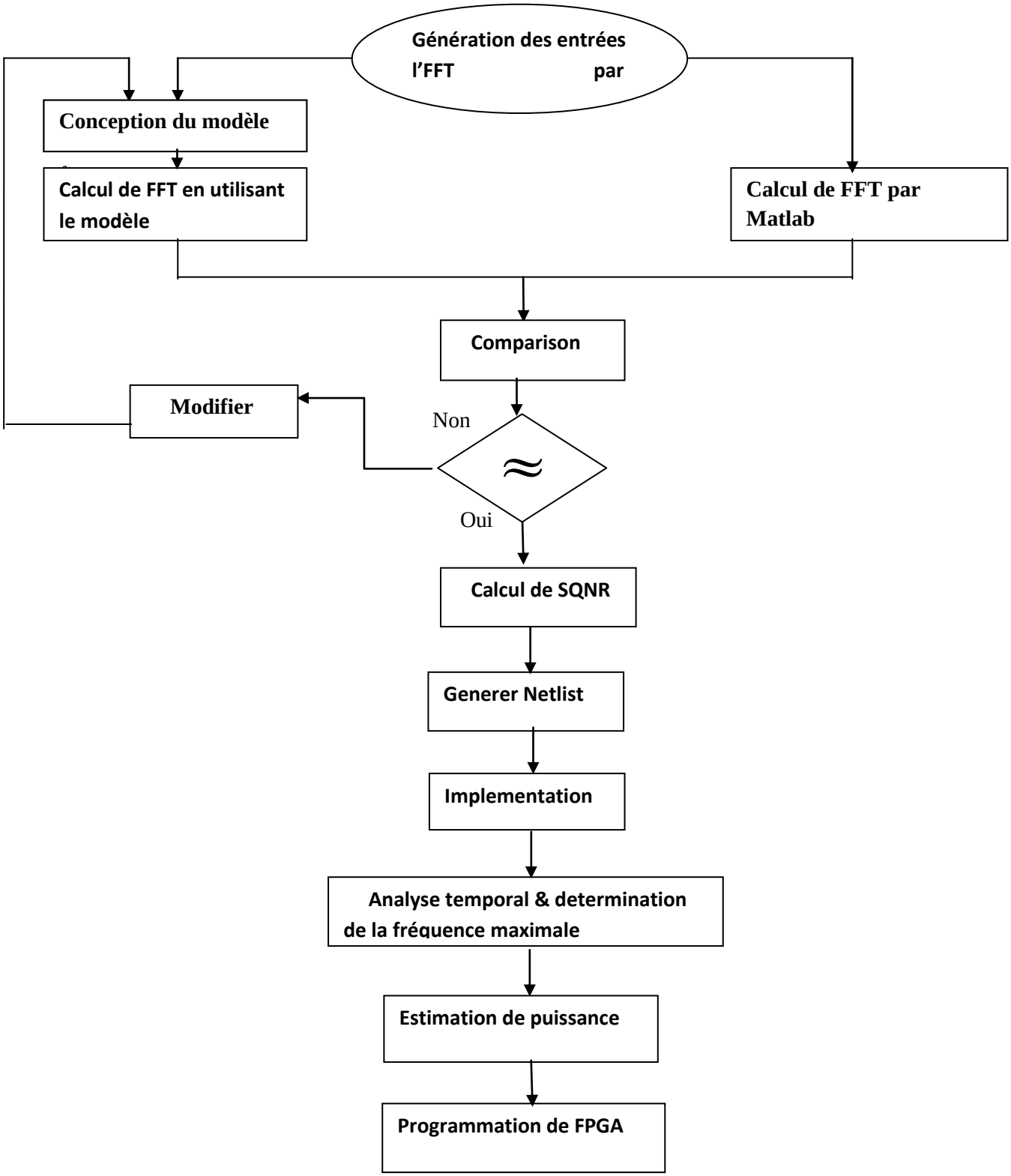


Figure (VI.2) : présente un organigramme simplifié

III.1)- description de l'organigramme :

Dans Cette section on vas faire une description détaillée .pour chaque étape de l'organigramme :

a)- Génération des entrées par Matlab :

Un programme Matlab est développé pour la génération de deux séquences aléatoires complexes de taille allant de 8 à 1024. Chaque séquence possède une partie réelle et une partie imaginaire.

b)- **Conception du schéma :**

Pour la conception du modèle FFT implémenté sur FPGA on a utilisé le logiciel « system générateur » de Xilinx. Ce logiciel est compatible et interactif avec l'environnement Simulink et l'environnement Matlab. Le logiciel « system générateur » peut importer des données soit de l'environnement Matlab soit de l'environnement Simulink comme il peut exporter les données vers Matlab ou Simulink. Cette propriété permet de faciliter la simulation la détection et la correction des erreurs de conception. On outre logiciel « system générateur » fournie une bibliothèque très riche en composants optimisés et configurables selon le besoin du concepteur. Cette bibliothèque contient une panoplie de composant allant des composants les plus simple comme les portes logiques jusqu'aux éléments compliqués dédiés au traitement numérique du signal (DSP) et à la communication numérique comme le codeur convolutif, le décodeur de Viterbi, filtre FIR.

En plus le logiciel « system générateur » permet de générer automatiquement le modelé HDL (Hardware Discription Language) en langage VHDL ou Verilog afin de passer rapidement vers l'étape d'implémentation.

Pour notre modèle de conception nous avons utilisé les composants suivants :

- ROM : la ROM est utilisée pour sauvegarder les valeurs des coefficients W_N^{nk}.
- Chaque étage : à l'exception du dernier et de l'avant dernier ; utilise deux ROMs, l'une pour les parties réelle des coefficients et l'autre pour les parties imaginaires. La ROM peut être implémentée soit en utilisant des mémoires distribuées « distributed memory » soit en utilisant des RAM en bloc « bloc RAM ».
- Complex Multipliers 3.1 : Ce composant est utilisé pour implémenter la multiplication complexe. Dans chaque étage ; à l'exception du dernier et de l'avant dernier ; on utilise un seul

- Complex Multipliers : ce composant peut être implémenté soit en utilisant des LUT soit en utilisant composant de type DSP48 disponible dans les versions récentes des circuits FPGA. En plus le module peut être optimisé soit pour minimiser les ressources utilisées soit pour maximiser sa fréquence de travail.
- ADD/SUB : c'est un module qui permet d'effectuer les opérations d'addition et de soustraction aux niveaux de chaque unité de calcul BF. Chaque BF utilise 4 modules de type ADD/SUB. Deux pour l'addition des parties réelles et des parties imaginaires et deux autres pour la soustraction des parties réelles et des parties imaginaires.
- Mux : ce sont des multiplexeurs qu'on utilise pour acheminer les données dans les différentes parties de l'architecture.
- Delay : ce sont des éléments qui permettent de retarder les données avec un nombre fini de cycle horloge, afin de combiner d'une façon convenable les données.
- Counter : c'est un compteur binaire qui permet de générer les signaux nécessaires pour la commande des multiplexeurs dans chaque étage de l'architecture ainsi que les signaux de l'adressage des ROMs dans chaque étage.

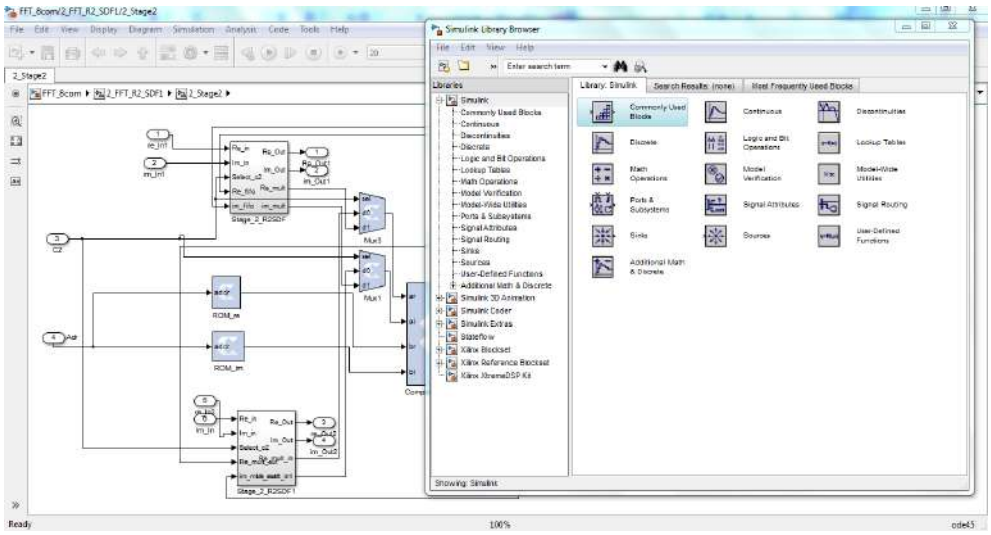


Figure (VI.3) : montrent les composants est « simulink Libray browser »

c)-Calcule de la DFT par le processeur FFT conçu en utilisant « système generator » :

Les deux séquences complexes générées par Matlab sont d'abord exporter vers Simulink en utilisant l'intermidaire « from workspace » disponible dans la bibliothèque de Simulink puis passées vers « System Generator » par le biais des portes « Gateway in ». Après lancement de la simulation le modèle permet de calculer la DFT des deux séquences complexes. Les deux DFTs des deux

séquences sont exportées vers Simulink par des portes « Gateway out ».puis vers Matlab par l'intermédiaire du bloc « To workspace » disponible lui aussi dans la bibliothèque de Simulink.

La figure suivante donne un exemple pour le modèle FFT de taille 8.

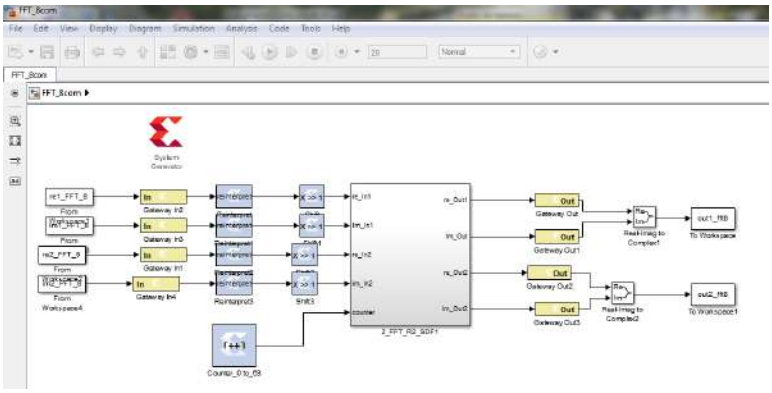


Figure (VI.4) : schéma global du processeur FFT de taille 8 conçu en utilisant « System Generator »

Le processeur FFT de taille 8 est composé de trois étages comme illustré dans la figure suivante :

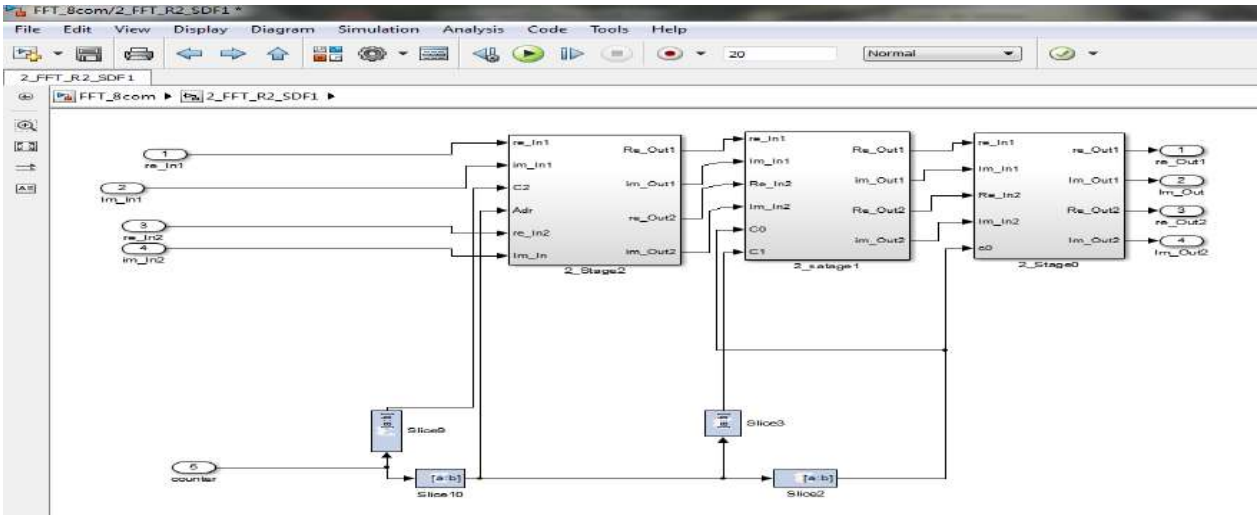


Figure (VI.5) : présentation des trois étages d'un processeur FFT de taille 8.

d)- Calcul de la FFT par MATLAB :

Dans cette étape les DFTs des deux séquences complexes sont calculées en utilisant la fonction *FFT* de Matlab. On note ici que la fonction *FFT* du logiciel Matlab utilise la virgule flottante pour la représentation des données alors que notre modèle développé sous « System Generator » utilise la représentation en virgule fixe. Ce qui signifie que les DFTs calculées en utilisant Matlab présentent une précision meilleure que les DFTs calculées en utilisant le modèle de « System Generator ».

e)- Comparaison :

Après calcul des DFTs par Matlab et par le modèle de « System Generator » on fait une comparaison entre les deux résultats en calculant la différence entre les deux résultats. Cette différence est tout simplement l'erreur de quantification due à l'utilisation de la virgule fixe au lieu d'utiliser la virgule flottante.

f)-modification :

Si les résultats obtenus par le modèle de « System Generator » ne sont pas proches des résultats obtenus en utilisant Matlab, on précède à la modification et à la correction d'éventuelles erreurs dans notre modèle et en revient à l'étape ***a***. On répète la procédure jusqu'à obtenir des erreurs acceptable.

Le tableau suivant donne un exemple de résultat obtenu pour un FFT de taille 8

DFT calculée par Matlab	**DFT calculée par notre modèle**	**différence**
0.6322 - 2.0204i	0.6313 - 2.0220i	0.0009 + 0.0016i
-1.2220 + 2.7872i	-1.2222 + 2.7876i	0.0001 - 0.0004i
2.2026 - 0.1047i	2.2031 - 0.1045i	-0.0005 - 0.0002i
1.9634 - 1.9321i	1.9639 - 1.9326i	-0.0005 + 0.0005i
1.1628 - 3.2578i	1.1606 - 3.2578i	0.0021 - 0.0000i
-0.0500 + 1.9441i	-0.0493 + 1.9443i	-0.0007 - 0.0002i
-0.4699 - 1.9842i	-0.4702 - 1.9844i	0.0003 + 0.0002i
0.8161 - 0.5415i	0.8159 - 0.5420i	0.0002 + 0.0005i

TABLEAU (VI.1) : comparaison entre DFT calculée par Matlab et DFT calculée par notre modèle

g) -Calcul du rapport Signal sur Erreur de quantification : (SQNR) :

Le rapport SQNR permet de donner une idée précise sur l'effet du bruit de quantification sur la puissance du signal quantifié. Le SQNR est donnée par :

$$SQNR = 10log_{10}\left(\frac{\sum_{j=0}^{N-1}|X_{fl}(j)|^2}{\sum_{j=0}^{N-1}|X_{fi}(j) - X_{fl}(j)|^2}\right)$$

Avec $X_{fl}(j)$ sent les échantillons de la DFT éxacte (en utilisant la virgule Flottante).

$X_{fi}(j)$ sont les échantillons de la DFT approximative (en utilisant la virgule Fixe).

N: est la taille de la FFT.

h)-génération du modèle HDL équivalent :

Après vérification du bon fonctionnement du modèle développé en utilisant le logiciel « System Generator » on passe à l'étape de la génération du modèle HDL équivalent à notre modèle « System Generator ». Dans cette étape on doit spécifier le language HDL (Verilog ou VHDL), le type du circuit FPGA cible de l'implémentation (Artix7, spartan3,spartan 6 ,virtex5,virtes 7……ect) et la periode de signal. Pour notre projet et à afin de faire une comparaison nous avons utilisé deux types de circuits FPGA :

- Spartan 6 xc6slx45-3csg324
- Virtex 7 xc7Vx485t-2ffg1761



Figure (VI.6) : Génération du modèle HDL par **«** System Generator »

i)- Implémentation :

Après génération du modèle HDL de notre processeur FFT on passe à l'environnement ISE afin de faire l'implémentation du processeur sur circuit FPGA .dû on lance l'implantation en cliquant sur « GENERATE».



On lance l'outil ISE projet Navigator puis on ouvre le fichier de conception correspondant à notre processeur comme le montre la figure suivante :

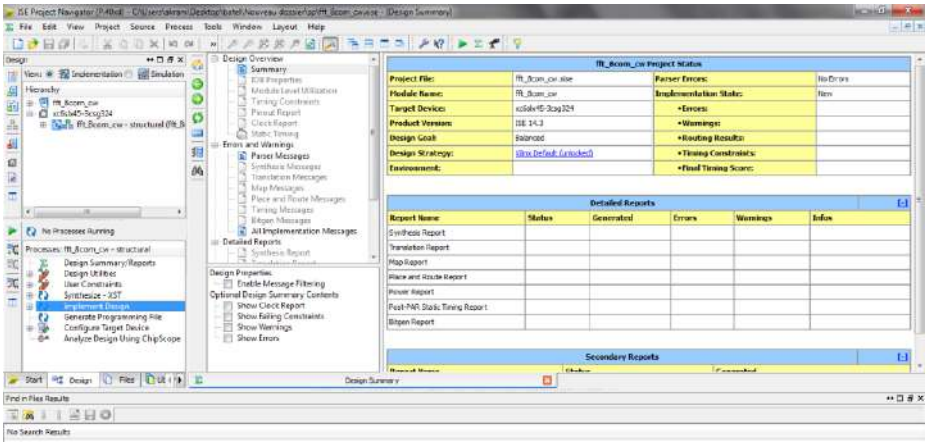


Figure (VI.7) : Lancement du modèle d'implémentation dans l'environnement « Project Navigator »

Maintenant on lance l'implémentions par double click sur « Implement Design », à la fin de cette étape l'outil permet d'offrir un rapport détaillé sur les différents résultats d'implémentation. Les résultats le plus significatifs sont les ressources utilisées « Device Utilisation Summary », les erreurs les avertissements, les contraintes temporelles ainsi que le rapport de timing après l'étape de placement et de routage « PAR timing repport »

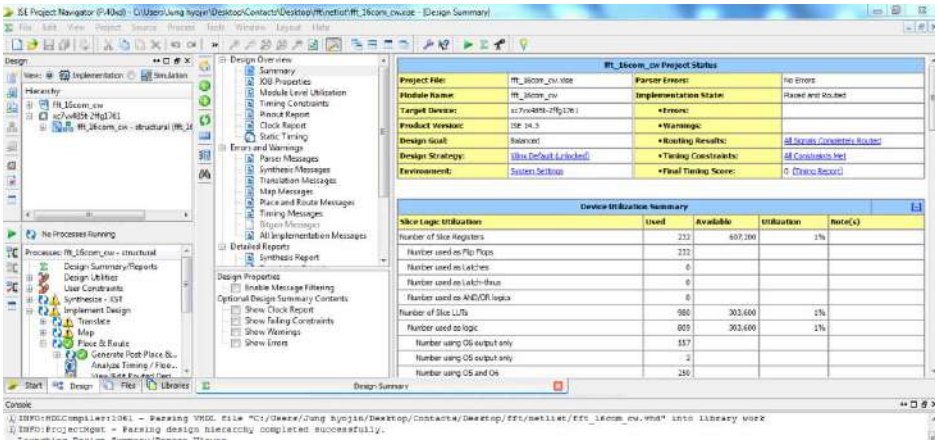


Figure (VI.8) Exemple de résultats d'implémentation

j)- Analyse temporal & détermination de la fréquence maximale :

Pour déterminer la fréquence maximale de signal horloge du circuit FPGA, on précède à un changement graduelle de la période jusqu'à l'obtention d'une période minimale très proche de celle choisie dans la contrainte de temps, comme le montre la figure suivante :

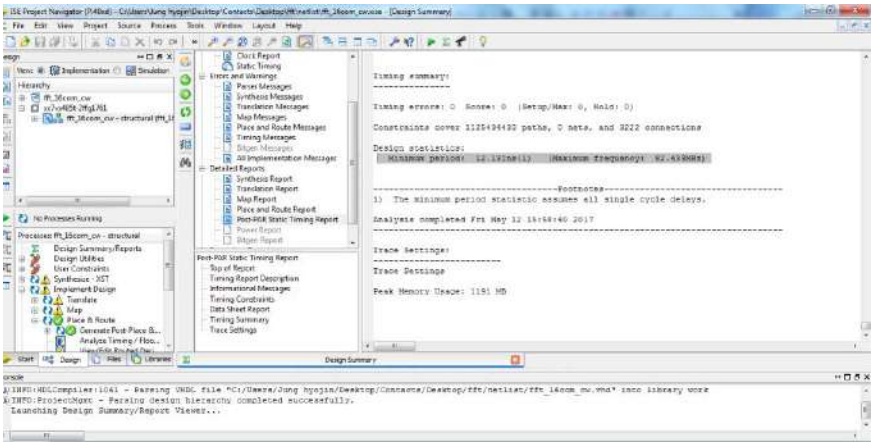


Figure (VI.9) : présentation de la fréquence maximale obtenue après PAR.

k)-Estimation de la puissance dynamique :

L'estimation de la puissance est effectuée par le logiciel « XPower Analyzer » qu'on peut lancer directement de l'outil « ISE Project Navigator », ce logiciel permet l'estimation de la puissance statique et de la puissance dynamique consommée par notre processeur. En plus il permet de fournir un rapport détaillé sur la puissance consommée par chaque type de ressource dans le circuit FPGA (DSP, Logic, IO…)

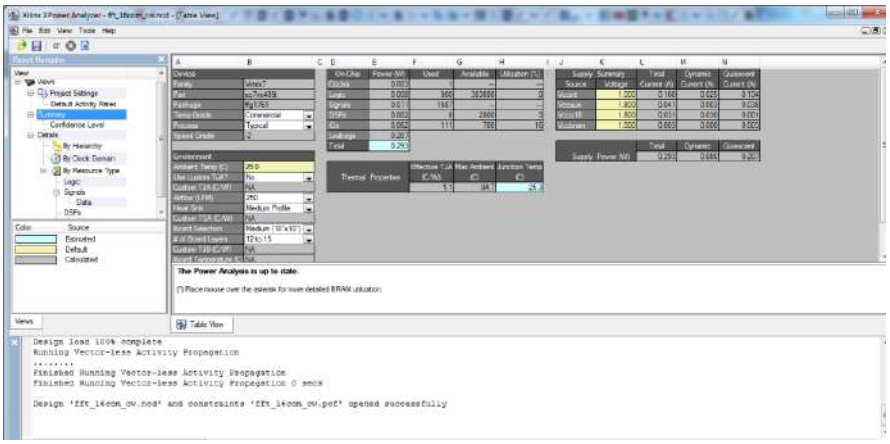


Figure (VI.10) Example d'estimation de la puissance par « XPower Analyzer »

IV)-Présentation des résultats de simulation et d'implémentation :

- Les étapes de développement décrites dans le paragraphe précédent sont répétées pour les FFT de taille 8 jusqu'à la taille 1024. L'implémentation été réalisée sur un circuit FPGA de type Spartan 6 xc6slx45-3csg324 et un autre circuit de type Virtex 7 XC7VX485t-2ffg1761

On note ici que le passage d'un processeur FFT de taille 2^i vers un processeur FFT de taille 2^{i+1} s'effectue en ajoutant un étage en amont. La figure suivante illustre le passage d'un FFT de taille 8 vers un FFT de taille 16.

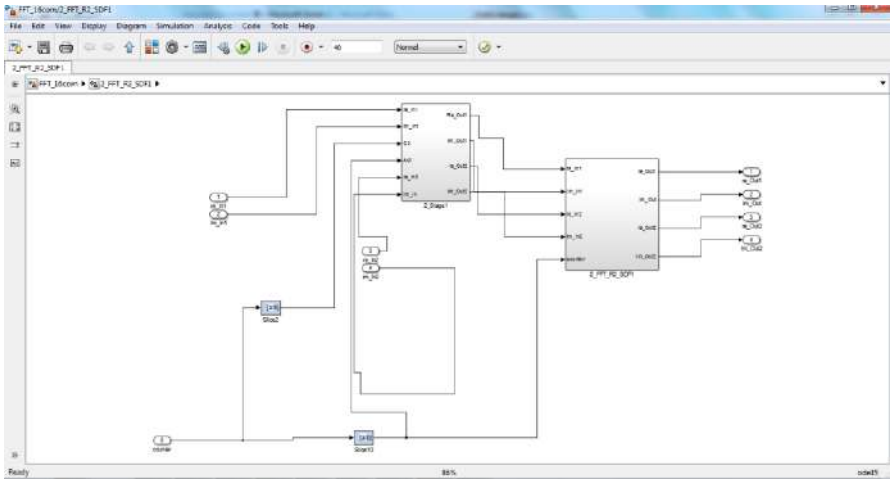


Figure (VI.11) passage de l'FFT de taille 8 vers l'FFT de taille 16.

Tableau(VI.2) : présente les résultats de l'implémentation des processeurs FFT sur le circuit Spartan 6 xc6slx45-3csg324

FFT	**8**	**16**	**32**	**64**	**128**	**256**	**512**	**1024**
SLICE REGESTERS	175	232	289	402	627	1076	1975	3710
SLICE LUT	724	980	1239	1554	1993	2680	3802	5904
OCCUPIED SLICE	238	352	468	564	680	923	1209	1738
DSP 48E	4	8	12	16	20	24	28	32
FREQUENCE (MHz)	98.135	82.43	70.13	62	56.167	51.54	45.508	42.139
PUISSANCE TOTAL (mW)	302	297	293	290	280	284	282	291
TIME (ns)	10.19	13	14.25	16.12	17.804	19.45	21.97	27.73

Le Tableau(VI.3) : présente les résultats de l'implémentation des processeurs FFT sur le circuit Virtex 7 XC7VX485t-2ffg1761

N	**8**	**16**	**32**	**64**	**128**	**256**	**512**	**1024**
SLICE REGESTERS	175	232	289	402	627	1076	1973	3710
SLICE LUT	662	896	1127	1386	1713	2176	2875	4111
OCCUPIED SLICE	235	326	457	552	664	788	953	1375
DSP 48E	4	8	12	16	20	24	28	32
FREQUENCE (MHz)	65.011	50.98	41.682	38.372	31.22	30.601	25.654	26.322
PUISSANCE TOTAL (mW)	227	172	181	165	151	151	150	159
TIME (ns)	14.99	19.612	23.991	26.193	30.313	32.679	35.916	37.991

Les résultats des deux tableaux précédents montrent que les ressources utilisées augmentent avec l'augmentation de la taille de l'FFT. De l'autre part la fréquence maximale diminue avec l'augmentation de la taille *N*. Les résultats montrent aussi que la puissance consommée varie légèrement avec la taille de l'FFT. En outre les résultats montrent que l'implémentation sur le circuit Virtex 7 XC7VX485t-2ffg1761 permet d'obtenir une meilleure fréquence par rapport au circuit Spartan 6 xc6slx45-3csg324

Tableau (VI.4) : présenté le SQNR pour la différente taille de l'FFT. On remarque que le SQNR est supérieur à 40 dB et que la valeur du SQNR diminue avec l'augmentation de la taille de l'FFT.

FFT	**SQNR (dB)**
8	66,8872
16	63,7035
32	60,6077
64	55,5094
128	52,4064
256	48,8546
512	45,7345
1024	42,8379

V) -Conclusion :

Dans ce chapitre, nous avons commencé par la présentation de l'architecture SDF RADIX-2 à deux voies parallèles qui permet de calculer simultanément les DFTs de deux séquences complexes et indépendantes. Ensuite nous avons présenté avec une explication détaillée l'organigramme représentant les différentes étapes de développement en commençant de la génération des entrées jusqu'à l'implémentation. Nous avons développé sous « System Generator » des processeurs FFT pour le calcul des DFTs de taille allant de 8 jusqu'à 1024. Pour l'implémentation nous avons ciblé deux circuits FPGA, qui sont le circuit Virtex 7 XC7VX485t-2ffg1761 et le circuit Spartan 6 xc6slx45-3csg324. Les résultats de l'implémentation montrent que le circuit Virtex 7 est plus performant que le circuit Spartan 6 en terme de vitesse.

L'objectif principal de notre projet de fin d'études était l'implémentation sur FPGA de l'algorithme FFT à base de radix-2 en utilisant l'architecture SDF pour le calcul de la DFT de deux séquences d'entrée parallèle. le processeur FFT obtenu est utilisé dans un système de communication de type MIMO-OFDM.

Un système de communication utilisant la technique MIMO-OFDM permet d'augmenter le débit, d'améléorer l'efficacité spectrale et de diminuer le taux de l'erreur binaire BER « Bit Error Rate ».

Un système de communication à base de la technique MIMO-OFDM nécessite le calcul de la DFT à la réception et la DFT inverse à l'émission. Le FFT est un algorithme très efficace pour le calcul de la DFT car il permet de réduire significativement la complexité de calcul de la DFT.

Il existe plusieurs version de l'algorithme FFT (Radix-2, Radix-4, Radix-8, Radix-r) chaque version est caractérisée par sa complexité de calcul, et par la forme de sa structure de base appelé module élémentaire, la forme la plus simple est celle du papillon « Butterfly » pour la version Radix-2.

Pour l'implémentation de l'algorithme FFT, On trouve dans la littérature plusieurs architectures, les plus communes sont les architectures SDF, MDC et SDC. Chaque architecture est caractérisée par le nombre de ressources qu'elle consomme (additionneur, multiplieur, mémoire), par ces performance (nombre d'échantillons traités par cycle horloge et retard de calcul) et par la complexité de contrôle.

Les circuits programmables de type FPGA représentent une cible privilégiée pour l'implémentation des algorithmes de traitement numérique du signal en général et pour les communications numériques en particulier. Ceci est dû aux avantages que présente cette technologie d'une part et à l'ensemble des outils de développement disponible et qui facilite énormément la tache de développement des applications de l'autre part.

Dans notre projet nous avons implémenté l'algorithme FFT radix-2 à base de l'architecture SDF à deux voies pour une taille variable allant de 8 jusqu'à 1024. Nous avons commencé par la conception des modèles en utilisant le logiciel « Xilinx System Generator ». Durant la phase de conception nous avons constaté que « Xilinx System Generator » est un outil très efficace et très attractif à cause des facilités qu'il offre pour les concepteurs. Après vérification du bon fonctionnement de chaque modèle par comparaison avec les résultats de Matlab, nous avons passé à

l'étape de l'implémentation. Pour l'implémentation nous avons ciblé deux circuits qui sont le circuit Virtex 7 XC7VX485t-2ffg1761 et le circuit Spartan 6 xc6slx45-3csg324.

La réalisation de ce projet nous a permet de :

- Avoir une vue globale sur les systèmes de communications numériques et spécialement ceux utilisant la technique MIMO-OFDM.
- Etudier la technique FFT du point de vu algorithmique et du point de vu architecturale.
- Avoir une idée sur les circuits FPGA et se familiariser avec l'outil « System Generator » et l'outil « Xilinx ISE ».

Les perspectives de ce travail sont nombreux on peut citer à titre d'exemple :

- L'implémentation sur FPGA de l'algorithme FFT à base de radix-2 en utilisant l'architecture SDF pour le calcul de la DFT de deux séquences d'entrée parallèle à haut fréquence : la fréquence peut être amélioré considérablement en insérant des registre de pipeline à la sortie des composant. Cette étape a été commencée mais malheureusement n'a pas été achevée.
- L'intégration du processeur FFT système de communication de type MIMO-OFDM et l'évaluation de ses performance.
- L'implémentation sur FPGA des autres modules qui constituent le système MIMO-OFDM.

RÉFÉRENCE BIBLIOGRAPHIE

[1] M.LOUNICI, N.DJEBRANE, « implémentation d`une chaine de communication numérique sur carte dsp tms 320c6711 dsk, » mémoire de master école militaire polytechnique, 2011.

[2] HENRIK SCHULZE, CHRISTIAN LÜDERS.SCHOLTZ, « Theory and Applications of OFDM and CDMA»,Edition J. Wiley and sons, 2005.

[3] Ali RACHINI, « Etude de la synchronisation temporelle dans les systèmes MIMO-OFDM appliqués aux réseaux mobiles », Université INSA de rennes,2014.

[4] A.BENAMAR, N. BELKHATIR, F.T. BENDIMERAD, « Adaptive and Context-Aware Scenarios for Pervasive Technology-Enhanced Learning System», International Arab Journal of e-Technology- A JeT Vol. 3 N° 1, pp 58-67, 2013.

[5] S. A. DJENNAS, F. T. BENDIMERAD « Proposition of an Unexploited Multiple Access Technique», International Journal of Computer Science Issues, Vol. 9, No. 3, pp. 346-353, 2012.

[6] CHELLALI Saida, SERRAH Meriem « *Implémentation de l'algorithme FFT sur FPGA*» mémoire de master yahia farés de médea.

[7] YOUSSEF ACHOURI, « implémentation efficace de l'FFT pour des communications OFDM », thèse de doctorat, université du Québec à Trois-Rivières, 2010.

[8] CHAWKI SAHNINE, " architecture de circuit intégré reconfigurable, très haut débit et basse consommation pour le traitement numérique de l'OFDM avancé",l'institut polytechnique de Grenoble le 30 janvier 2009.

[9] en.wikipedia.org/wiki/Bit-reversal permutation. DATE 19/03/2017.

[10] Adrien BLANCHARDON « Synthèse d'architectures de circuits FPGA tolérants aux Défauts » Docteur de l'Université de thèse le 15/09/2015.

[11] Mr. FREDERIC MORAIN-NICOLIER « Conception d'un outil de prototypage rapide sur le FPGA pour des applications de traitement d'images » le 04 novembre 2011.

[12] MOUNIR BOUKADOUM « Traitement numérique des signaux »UQAM, 2014.

[13] Françoi Verdier « Cours Master Recherche » Université de cerg-Pontoise page 18.

Référence bibliographie

Site web:

[14] https://fr.wikipedia.org/wiki/Circuit_logique_programmable DATE/ 18/03/2017 PM 23.00.

[15] http://www.ni.com/white-paper/8043/fr/ DATE 03/05/2017 AM 13.15.

[16] xilinx. the programmable logic data book, 1995. DATE 27/05/2017 AM. 10.00

[17] https://fr.wikipedia.org/wiki/Xilinx. DATE 20/05/2017 15.30.