

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN GÉNIE ÉLECTRIQUE

PAR
KHALED ATOUB

OPTIMISATION DE LA FFT À VIRGULE FIXE BASÉE SUR LES ALGORITHMES
GÉNÉTIQUES

NOVEMBRE 2013

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

RÉSUMÉ

Aujourd'hui l'arrivée de la 4^{ème} génération représentée, entre autres, par les réseaux LTE à améliorer le monde de la télécommunication mobile grâce à l'incorporation des techniques OFDM et le MIMO-OFDM. La modulation multiporteuse OFDM consiste à répartir les informations à transmettre sur un grand nombre de porteuses orthogonales, individuellement modulées à débit réduit. L'implémentation de cette modulation est basée sur le principe d'orthogonalité assurée par la transformée rapide de Fourier (FFT – *Fast Fourier Transform*) pour la génération des sous-porteuses. L'exécution en arithmétique à virgule fixe des IFFT/FFT se traduit par la présence de sources de bruits. Ces derniers se propagent au sein du système et provoquent une dégradation de la précision mathématique en sortie du système. Ainsi, il est nécessaire de maîtriser la dégradation de la précision afin de garantir l'intégrité de l'algorithme et les performances du système.

Ce mémoire, présente l'étude de quantification en virgule fixe de l'IFFT/FFT en se basant sur une architecture matérielle de type pipeline radix-4 MDC (*Multi-path Delay Commutator*). Nous présentons une méthode d'optimisation en virgule fixe, basée sur les

RÉSUMÉ

algorithmes génétiques (AG), afin de calculer les coefficients FFT tout en minimisant les erreurs de quantifications. La précision de l'implémentation est évaluée en déterminant l'expression du Rapport Signal à Bruit de Quantification (SQNR : *Signal-to-Quantisation-Noise Ratio*). Une étude de comparaison a été faite entre la méthode qui consiste à simplement tronquer les coefficients de Fourier et celle basée sur les AG. Les résultats obtenus montrent que pour une même largeur binaire des coefficients de Fourier, notre méthode d'optimisation à base des algorithmes génétiques présente un gain en moyenne de 1.5 dB comparativement à la troncature des coefficients de Fourier et cela sans aucun coût matériel supplémentaire.

Nous présentons dans ce travail aussi l'évaluation de la FFT radix-4 MDC sur FPGA afin de montrer le gain en SQNR obtenu ainsi que la consommation en puissance et ressources matérielles utilisées.

REMERCIEMENT

Je tiens à exprimer mes plus vifs remerciements à Dieu le tout puissant pour la volonté, la santé, et la patience qu'il m'a donné toutes ces longues années d'études, que nous puissions arriver là.

Je tiens à exprimer ma profonde reconnaissance et ma gratitude à mon professeur, M Daniel Massicotte pour m'avoir encadré, son orientation et ses précieux conseils durant l'élaboration de ce travail, aussi aux membres de jury pour avoir accepté de juger mon travail.

A mes deux êtres les plus chers au monde Ma chère mère et mon cher père

A toute mes sœurs et mes frères pour leurs sincère aide morale

A toute ma famille sans exception

A tous mes collègues et amis. A tous ceux qui avec de chaleur et d'amour se sont dévoués pour ma réussite et mon bonheur, je dédie ce travail.

TABLE DES MATIÈRES

TABLE DES MATIÈRES

RÉSUMÉ	II
REMERCIEMENT	IV
TABLE DES MATIÈRES.....	V
LISTES DES FIGURES	XI
LISTE DES TABLEAUX	XVI
LISTE DES ABRÉVIATIONS	XVIII
CHAPITRE I	1
INTRODUCTION	1

TABLE DES MATIÈRES

I.1. Problématique liée au sujet de recherche	5
I.2. Objectifs du sujet de recherche	7
I.3. Méthodologie du sujet de recherche.....	8
I.4. Organisation de mémoire	9
CHAPITRE II	11
FFT QUANTIFIÉES AU RÉCEPTEUR AU OFDM.....	11
II.1. Les systèmes multi-porteurs OFDM.....	12
II.1.1. Historique – Principe.....	12
II.1.2. Modélisation des systèmes OFDM.....	14
II.1.2.1. Modulations OFDM.....	15
II.1.2.2. Orthogonalités.....	17
II.1.3. La FFT dans les communications OFDM	19
II.1.3.1 Implémentation numérique de la Modulation/Démodulation OFDM	19
II.1.3.2. Quelques standards OFDM à base FFT	21
II.1.2. Avantages et inconvénients de l'OFDM	22
II.2. Algorithme FFT et l'effet de quantification.....	24
II.2.1 La Transformée de Fourier Discrète.....	24

TABLE DES MATIÈRES

II.2.2. La Transformée Rapide de Fourier.....	28
II.2.3. Approche diviser-pour-régner	29
II.3. Architecture pipeline pour le calcul de la FFT	32
II.3.1. Classification des algorithmes FFT	33
II.3.2. Papillon radix-r FFT	34
II.3.3. Architecture pipeline radix-r FFT	35
II.3.4. Algorithme FFT radix-4 MDC	38
II.3.4.1. Algorithme radix-4 DIT	38
II.3.4.2. Algorithme radix-4 DIF	41
II.3.4.2. Architecture pipeline radix-4MDC	44
II.4. Effet de quantification.....	45
II.4.1. Arithmétique en virgule fixe conventionnelle.....	47
II.4.1.1. Représentation des nombres	47
II.4.1.2. Erreur de quantification	49
CHAPITRE III.....	52
ALGORITHME GÉNÉTIQUE POUR L’OPTIMISATION DE LA FFT	52
III.1. Algorithme génétique pour l’optimisation en virgule fixe des coefficients de la FFT	53

TABLE DES MATIÈRES

III.2. Algorithme génétique et stratégie d'évolution	54
III.2.1. Présentation des algorithmes génétiques.....	54
III.2.2. Principe de base des AG	56
III.3. Algorithme génétique pour l'optimisation des coefficients de la FFT	58
III.3.1. Génération de la population initiale	58
III.3.2. La phase d'évaluation	60
III.3.3. La reproduction	61
III.3.4. Sélection.....	61
III.3.4.1. Sélection par échantillonnage stochastique universel	62
III.3.4.2. Variante linéaire	64
III.3.4.2. Variante non linéaire	64
III.3.5. Croisements (Recombinaison)	65
III.3.6. Mutation	67
III.3.7. Critère d'arrêt.....	68
III.3.8 Réglage des paramètres des AG.....	68
III.4. résultats des coefficients de Fourier optimisés en virgule fixe.....	69
III.4.1. Métrique pour évaluer la qualité d'optimisation.....	69
III.4.2. Initialisation des paramètres et conditions de simulation	71
III.4.3. Optimisation des coefficients FFT	72

TABLE DES MATIÈRES

III.4.4. Étude de comparaison de SQNR.....	78
III.5. Analyse de variance des SQNR.....	82
III.6. Algorithme génétique à objectif multiple	87
3.7. Conclusion.....	88
CHAPITRE IV	89
IMPLEMENTATION D'UNE FFT RADIX-4 MDC À 16 POINTS	89
IV.1. La consommation en puissance dans les circuits FPGA.....	90
IV.2. Processus d'estimation de puissance de la FFT radix-4	91
IV.3. Conception et architecture pipeline radix-4 FFT.....	95
IV.3.1. BPE radix-4 FFT.....	98
IV.3.2. Multiplication complexes.....	99
IV.3.3. Le commutateur	100
IV.3.4 Evaluation de SQNR de l'architecture pipeline radix-4 MDC à 16 points	101
IV.4 Implémentation FPGA et analyse des résultats	103
IV.4.1 Résultats de synthèses de radix-4 FFT à 16 points.....	103

TABLE DES MATIÈRES

IV.4.2 Représentation schématique au niveau RTL de Processeur FFT R4MDC	106
IV.5. Analyse de résultats de consommation en puissance de radix-4 MDC à 16 point	109
CHAPITRE V.....	111
CONCLUSION.....	111
BIBLIOGRAPHIES	114

LISTE DES FIGURES

LISTES DES FIGURES

FIGURE 1.1: ÉVOLUTION DU MONDE DES TECHNOLOGIES SANS FIL [LER09]	3
FIGURE 1.2: ARCHITECTURE D'UN SYSTEME OFDM	5
FIGURE 1.3: PRÉVISION DE LA COMPLEXITÉ ALGORITHMIQUE ET DES PERFORMANCES DES PROCESSEURS [DSP05]	6
FIGURE 2.1: MULTIPLEXAGE DE FRÉQUENCE A) CLASSIQUE FDM B) ORTHOGONALE OFDM	13
FIGURE 2.2: SCHÉMA DE PRINCIPE D'UN MODULATEUR OFDM.	15
FIGURE 2.3: SCHÉMA DE PRINCIPE D'UN DÉMODULATEUR OFDM.	16

LISTE DES FIGURES

FIGURE 2.4: (A) LA FORME DES IMPULSIONS PER-SOUS-PORTEUSE. (B) SPECTRE POUR LA TRANSMISSION OFDM BASIC [DAH11].....	18
FIGURE 2.5: SPECTRE DU SIGNAL EN SORTIE DU MODULATEUR OFDM, DÉCOMPOSÉ SUR CHAQUE PORTEUSE. [DAH11].....	19
FIGURE 2.6: SCHÉMA DE PRINCIPE DU MODULATEUR OFDM NUMÉRIQUE	20
FIGURE 2.7: SCHÉMA DE PRINCIPE D'UN DÉMODULATEUR OFDM NUMÉRIQUE	21
FIGURE 2.8: LES COEFFICIENTS DE LA DFT POUR $N=8$	26
FIGURE 2.9: DÉCOMPOSITION DE 15 POINTS FFT EN 3 POINTS ET 5 POINTS TFD [PRO96] ..	32
FIGURE 2.10: PAPILLON (BPE) RADIX- R	34
FIGURE 2.11: ARCHITECTURE PIPELINE DE S STAGES RADIX- R FFT [JAB09].....	35
FIGURE 2.12: DIAGRAMME TYPIQUE DE PAPILLON RADIX-4 DIT.....	40
FIGURE 2.13: LA FORME SIMPLIFIÉE DE PAPILLON RADIX-4 DIT.....	40
FIGURE 2.14: DIAGRAMME DE LA FFT À BASE DE BUTTERFLY RADIX4 DE TYPE DIT $N=16$.	41
FIGURE 2.15: DIAGRAMME DE LA FFT À BASE DE BUTTERFLY RADIX4 DE TYPE DIF $N=16$.	44
FIGURE 2.16: ARCHITECTURE R4MDC ($N = 64$ POINTS).....	45
FIGURE 2.17: QUELQUES MÉTHODES DE REPRÉSENTATIONS EN VIRGULE FIXE	46
FIGURE 2.18: REPRÉSENTATION DES DONNÉES EN VIRGULE FIXE	47
FIGURE 2.19: MODÉLISATION DU PROCESSUS DE QUANTIFICATION DU SIGNAL x	50
FIGURE 2.20: BPE RADIX-4 FFT AVEC MISE À L'ÉCHELLE DE $1/4$	51
FIGURE 3.1: REPRÉSENTATION BINAIRE D'UN INDIVIDU DE DIMENSIONS 6.	56
FIGURE 3.2: PRINCIPES DE FONCTIONNEMENT D'UN ALGORITHME GÉNÉTIQUE	57

LISTE DES FIGURES

FIGURE 3.3: DÉCLARATION DES DÉFINITIONS À VIRGULE FIXE SANS MATLAB	59
FIGURE 3.4: ENCODAGES DES CHROMOSOMES D'UNE FFT DE 16 POINTS	59
FIGURE 3.5: PRINCIPES DE LA TECHNIQUE DE SÉLECTION SUS.....	63
FIGURE 3.6: OPÉRATION DE CROISEMENT.....	66
FIGURE 3.7: OPÉRATION DE MUTATION.....	68
FIGURE 3.8: MÉTHODOLOGIES D'ÉVALUATION DU SQNR	70
FIGURE 3.9: ÉVOLUTION DE FONCTION OBJECTIVE DANS LE CAS OÙ $N=16$, $wl_x = 10$, $wl_w = 10$	72
FIGURE 3.10: ÉVOLUTION DE FONCTION OBJECTIVE DANS LE CAS OÙ $N=64$, $wl_x = 10$, $wl_w = 10$	73
FIGURE 3.11: ÉVOLUTION DE FONCTION OBJECTIVE DANS LE CAS OÙ $N=256$ $wl_x = 10$, $wl_w = 10$	73
FIGURE 3.12: CONVERGENCE DE SQNR EN FONCTION DE NOMBRE DE GÉNÉRATION ($N=16, wl_x=v.floTTANT, wl_w=10$).....	78
FIGURE 3.13: CONVERGENCE DE SQNR EN FONCTION DE NOMBRE DE GÉNÉRATION ($N=64, wl_x=v.floTTANT, wl_w=10$).....	79
FIGURE 3.14: HISTOGRAMMES DE LA DYNAMIQUE DE SQNR ($N=16, wl_x=v.f, wl_w=10$)	80

LISTE DES FIGURES

FIGURE 3.15: HISTOGRAMMES DE LA DYNAMIQUE DE SQNR (N=64, WL_X=v.F, WL_W=10)	81
FIGURE 3.16: HISTOGRAMMES DE LA DYNAMIQUE DE SQNR (N=256, WL_X =v.F, WL_W=10)	81
FIGURE 3.17: ANOVA DES DISTRIBUTIONS STATISTIQUES DES SQNR (N=16, WL_X=v.F, WL_W=10).....	83
FIGURE 3.18: ANOVA DES DISTRIBUTIONS STATISTIQUES DES SQNR (N=64, WL_X=v.F, WL_W=10).....	84
FIGURE 3.19: ANOVA DES DISTRIBUTIONS STATISTIQUES DES SQNR (N=256, WL_X=v.F, WL_W=10).....	84
FIGURE 3.20: ANOVA DE DISTRIBUTIONS STATISTIQUES DES SQNR (N=16, WL_X=v.F, WL_W=10)	86
FIGURE 3.21: ANOVA DE DISTRIBUTIONS STATISTIQUES DES SQNR (N=64, WL_X=v.F, WL_W=10).....	86
FIGURE 4.1: ORGANIGRAMME DE CYCLE D'IMPLÉMENTATION ET L'ESTIMATION DE PUISSANCE.....	92
FIGURE 4.2: GRAPHE DE FLUENCE DES ENTRÉES/SORTIES DE LA FFT RADIX-4 MDC DE TYPE DIF POUR N=16 [RAO10].....	96
FIGURE 4.3: ARCHITECTURE PIPELINE RADIX-4 MDC DE TYPE DIF À DEUX ÉTAGES	97

LISTE DES FIGURES

FIGURE 4.4: STRUCTURE INTERNE DU BPE RADIX-4 FFT À BASE D'OPÉRATEURS RÉELS.	98
FIGURE 4.5: STRUCTURE INTERNE DE MULTIPLIEUR COMPLEXE	99
FIGURE 4.6: STRUCTURE INTERNE DE COMMUTATEUR	100
FIGURE 4.7: LES TRANSFORMATIONS DE COMMUTATEUR.....	101
FIGURE 4.8: BLOC TECHNOLOGIQUE RTL COMPLET DU R4MDC À 16 POINTS.....	107
FIGURE 4.9: SCHÉMA TECHNOLOGIQUE COMPLET DU R4MDC À 16 POINTS	108
FIGURE 4.10: SCHÉMA DE DESIGN INTERNE DE PROCESSEUR FFT R4MDC SUR FPGA.....	109

LISTE DES TABLEAUX

LISTE DES TABLEAUX

TABLEAU 2.1: PARAMÈTRE FFT DANS DIFFÉRENTES NORMES OFDM [TSA11]	22
TABLEAU 2.2: COMPARAISON ENTRE LES ARCHITECTURES DES PROCESSEURS FFT	33
TABLEAU 3.1: COMPARAISON EN VIRGULE FIXE EN NOMBRE SIGNÉS ENTRE LES COEFFICIENTS TRONQUÉS ET LES COEFFICIENTS GÉNÉTIQUES POUR UNE FFT À 16 POINTS	74
TABLEAU 3.2: COMPARAISON EN VIRGULE FLOTTANTE EN NOMBRE SIGNÉS ENTRE LES COEFFICIENTS TRONQUÉS ET LES COEFFICIENTS GÉNÉTIQUES POUR UNE FFT À 16 POINTS	74

LISTE DES TABLEAUX

TABLEAU 3.3: COMPARAISON EN VIRGULE FIXE EN NOMBRE SIGNÉS ENTRE LES COEFFICIENTS TRONQUÉS ET LES COEFFICIENTS GÉNÉTIQUES POUR UNE FFT À 64 POINTS	75
TABLEAU 3.4 COMPARAISON EN VIRGULE FLOTTANTE EN NOMBRE SIGNÉS ENTRE LES COEFFICIENTS TRONQUÉS ET LES COEFFICIENTS GÉNÉTIQUES POUR UNE FFT À 64 POINTS	76
TABLEAU 4.1 : ÉVALUATION DE SQNR EN UTILISANT LES W_{TRONC}	102
TABLEAU 4.2: ÉVALUATION DE SQNR EN UTILISANT LES W_{AG}	102
TABLEAU 4.3: ÉVALUATION DES RESSOURCES MATÉRIELLES (N=16, WL_X=16,WL_W=16)	104
TABLEAU 4.4: ÉVALUATION DES RESSOURCES MATÉRIELLES (N=16, WL_X=10,WL_W=10)	105
TABLEAU 4.5: ÉVALUATION DES RESSOURCES MATÉRIELLES (N=16, WL_X=16,WL_W=10)	105
TABLEAU 4.6 :ÉVALUATION DE LA CONSOMMATION EN PUISSANCE OBTENUS AVEC LES COEFFICIENTS TRONQUÉS	110

LISTE DES ABRÉVIATIONS

LISTE DES ABRÉVIATIONS

2G	:	Deuxième Génération des systèmes cellulaire
3G	:	Troisième Génération des systèmes cellulaire
4G	:	Quatrième Génération des systèmes cellulaire
3GPP	:	3 rd Generation Partnership Program
3GPP-LTE	:	3 rd Generation Partnership Program-Long Term Evolution

LISTE DES ABRÉVIATIONS

AG	:	Algorithmes Génétiques
ANOVA	:	ANalysis Of Variance
BPE	:	Butterfly Processing Element
CDMA	:	Code Division Multiple Access
DVB	:	Digital Video Broadcasting
DSP	:	Digital Signal Processing
DIT	:	Decimation- In-Time
DIF	:	Decimation- In-Frequency
DFT	:	Discrete Fourier Transform
DMT	:	Discrete Multi Tone
FFT	:	Fast Fourier Transform
FDMA	:	Frequency Division Multiplexing Access
FPGA	:	Field Programmable Gate Array
GSM	:	Global System for Mobile communications
HSPA	:	High Speed Packet Access
HDTV	:	High Definition Television
ISI	:	Inter Symbol Interference

LISTE DES ABRÉVIATIONS

ICI	:	Inter-carrier-Interference
IFFT	:	Invers Fast Fourier Transform
IDFT	:	Invers Discrete Fourier Transform
IMT	:	International Mobile Télécommunications
IMT-2000	:	International Mobile Télécommunications-2000
LTE	:	Long Term Evolution
LTE avancés	:	Long Term Evolution Advanced
LSSI	:	Laboratoire de Signaux et Systèmes Intégrés
MIMO	:	Multi-input-multi-output
MDC	:	Multi-path Delay Commutator
MC-FDMA	:	Multi-Carrier Frequency Division Multiple Access
MC-TDMA	:	Multi-Carrier Time Division Multiple Access
MC-CDMA	:	Multi-Carrier Code Division Multiple Access
NCD	:	Native Circuit Description
OFDM	:	Orthogonal Frequency Division Multiple
PAPR	:	Peak to Average Powerratio
PCF	:	Physical Constraints File

LISTE DES ABRÉVIATIONS

PAR	:	Place-and-Route
R2MDC	:	Radix2 Multipath Delay Commutator
R4MDC	:	Radix4 Multipath Delay Commutator
RTL	:	register transfer level
SDF	:	Single-path Delay Feedback
SDC	:	Single-path Delay Commutator
SUS	:	Stochastic Universal Sampling
SQNR	:	Signal Quantization Noise Ratio
TDMA	:	Time Division Multiple Access
UMTS	:	Universal Mobile Telecommunications System
UIT	:	Union internationale des télécommunications mobile
UWA	:	Underwater Acoustics
VLSI	:	Very Large Scale Integration
VCD	:	Value Change Dump
WiMAX	:	Worldwide Interoperability for Microwave Access
WIFI	:	Wireless Fidelity
XML	:	Extensible Markup Language

LISTE DES ABRÉVIATIONS

XPA : Xpower Analyzer de la compagnie Xilinx

CHAPITRE I

INTRODUCTION

L'évolution progressive du monde des télécommunications mobile est en train d'envahir la quasi-totalité des domaines d'activité, et la demande pour les systèmes de transmission assurant de très hauts débits avec une qualité de services importants est en croissance exponentielle. Ceci a motivé les chercheurs d'apercevoir des modèles de transmission capable de supporter des communications à large bande. Le développement des réseaux mobiles a traditionnellement été considéré d'une part comme une séquence de générations successives de réseaux de télécommunications [YAY11], essentiellement consacrés à la téléphonie (2^{ème}G, *GSM: Global System for Mobile Communications*) basée sur les systèmes mobiles numériques. L'orientation vers la transmission multimédia complète des données ainsi que les communications vocales étaient le fruit des

CHAPITRE I – INTRODUCTION

recherches pour les 3^{ème} génération de système de communication mobile nommé UMTS (*Universal Mobile Telecommunication System*) et IMT-2000 (*International Mobile Telecommunication-2000*).

Comparés aux réseaux GSM, les systèmes UMTS sont les premiers systèmes mobiles de traitement des données à haut débit, typiquement dans la gamme de 64 à 384 kbit/s, tandis que le taux maximum de données pour une faible mobilité ou des applications intérieures est de 2 Mbit/s. Avec l'extension du HSPA (*High Speed Packet Access*), les débits des données allant jusqu'à 10 Mbit/s sont disponibles dans la liaison descendante. Le rythme actuel, qui peut être observé dans le marché des communications mobiles, montre déjà que les systèmes 3G ne seront pas les systèmes à solution ultime. Dans un futur proche, selon les critères de l'Union internationale des télécommunications (UIT), qui établissent les normes pour les réseaux cellulaires, les systèmes de communication devront proposer un accès à l'internet avec des débits allant jusqu'à 1 Gbits/s dans une zone locale de couverture (environnement quasi statique) et offrant des débits de données jusqu'à 100 Mbits/s en environnement mobile ; c'est la quatrième génération (4^{ème}G) de réseaux cellulaires (voir la figure 1.1). Les technologies préconisées qui devraient satisfaire à ces critères sont les réseaux mobiles WiMAX (*Worldwide Interoperability for Microwave Access*), et les réseaux LTE avancés (*Long Term Evolution Advanced*). Ce type de réseau de haute gamme est voué à remplacer les réseaux de 3^{ème} génération basée sur le CDMA (Code Division Multiple Access) qui commence à atteindre leurs limites. De plus, ces normes ont été conçues pour durées longtemps, car le déploiement d'un nouveau réseau est très coûteux pour les opérateurs.

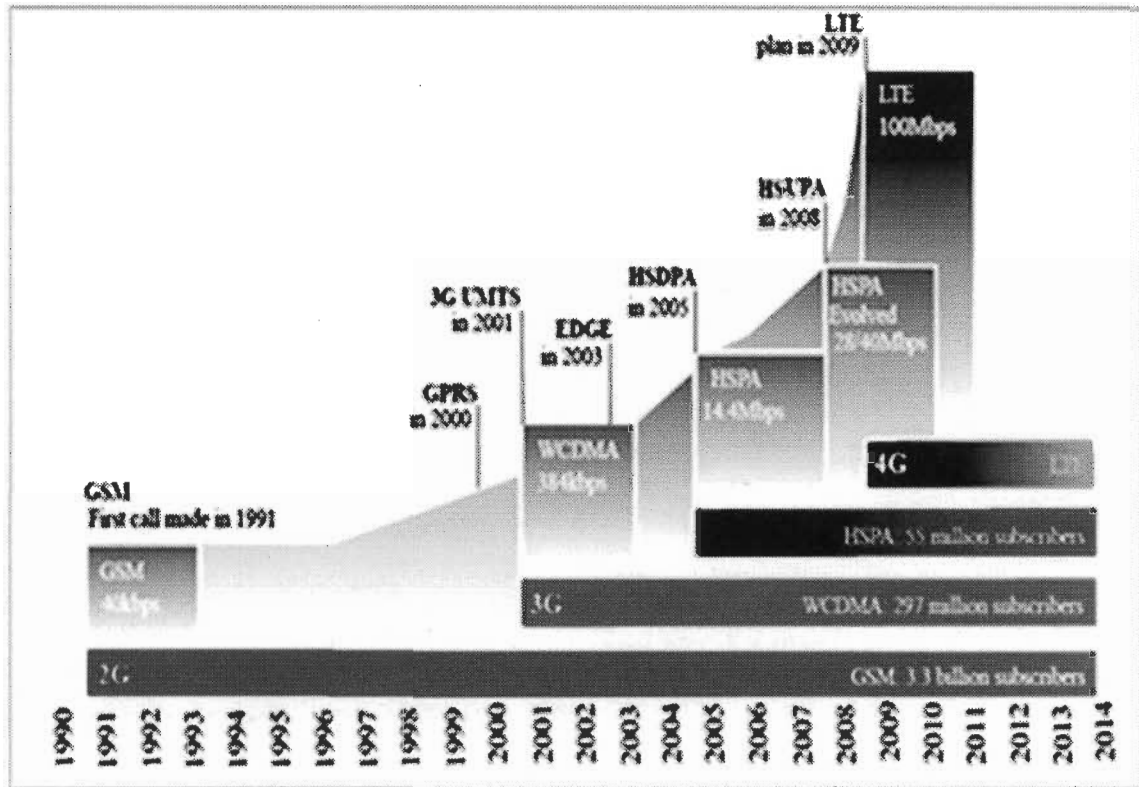


Figure 1.1: Évolution du monde des technologies sans fil [LER09]

Grâce à ces qualités remarquables, en particulier des débits élevés entre la station de base et les terminaux, et afin d'accroître les débits de données et l'efficacité du spectre, l'industrie a développé le nouveau système de radiocommunication mobile LTE (*Long Term Evolution*) qui est une évolution des techniques IMT (International Mobile Telecommunications : UMTS, HSPA, HSPA+). Le LTE vise une efficacité spectrale environ 3 à 4 fois supérieures à l'UMTS HSPA, pour un coût de réseau relativement bas (c'est-à-dire un faible coût par bit transféré). En outre, la réduction de l'intervalle de transmission des données (latence) devrait améliorer considérablement la réactivité du réseau. Enfin, la consommation d'énergie des réseaux LTE doit être minimale que l'UMTS, notamment au niveau du terminal.

CHAPITRE I – INTRODUCTION

Par rapport à la norme UMTS actuelle, la principale innovation de la technologie LTE se base sur l'introduction du procédé de codage par répartition en fréquences orthogonales sous forme de multiples sous-porteuses OFDM (*Orthogonal Frequency Division Multiple*) [MOL11]. En raison de sa flexibilité et son adaptabilité, l'OFDM a été choisi comme une interface pour la liaison descendante dans le cadre des standards LTE.

La spécificité de l'OFDM vient du recouvrement mutuel des différentes sous-porteuses, d'une manière dite orthogonale. Cette orthogonalité permet une utilisation optimale des ressources spectrales et facilite l'implantation numérique. Basant sur la figure 1.2 l'aspect nouveau et moderne de la technique de transmission OFDM est que les signaux sous-porteurs différents sont générés numériquement et conjointement par une transformée de Fourier rapide inverse (*en anglais : Inverse Fast Fourier Transform; IFFT¹*) dans l'émetteur et par une transformée de Fourier directe (*en anglais : Fast Fourier Transform ; FFT²*) dans le récepteur [ROL03]. En conséquence, la génération du signal d'émission est simplifiée et l'efficacité de la bande passante du système est considérablement améliorée.

¹ L'acronyme *IFFT*, plus largement utilisé comme terme technique, sera utilisé dans le mémoire

² L'acronyme *FFT*, plus largement utilisé comme terme technique, sera utilisé dans le mémoire

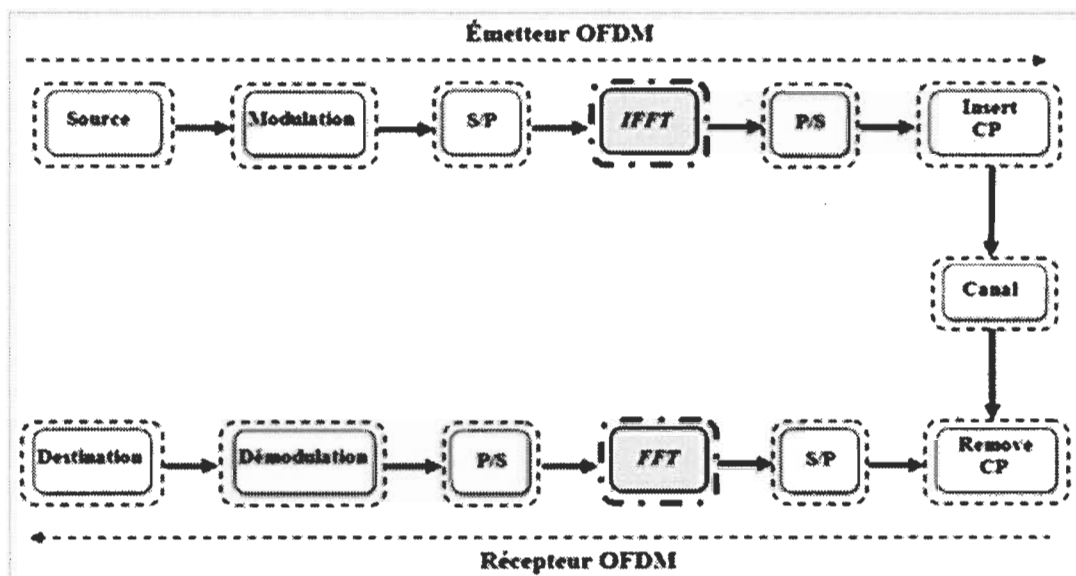


Figure 1.2: Architecture d'un système OFDM

I.1. PROBLÉMATIQUE LIÉE AU SUJET DE RECHERCHE

En se basant sur la figure 1.3, le monde des télécommunications a connu des avancées spectaculaires ces dernières décennies. La complexité algorithmique est un facteur fondamental qui ne cesse d'augmenter chaque fois avec l'évolution des systèmes télécommunications. Les futurs systèmes de communication sans fil sont amenés à proposer une qualité de service très importante en termes de débit d'information élevé, fiabilité, ainsi qu'une faible consommation électrique. L'apparition des réseaux d'accès sans fil large bande tels que le WIFI (*Wireless Fidelity*) ou le WiMAX, la HDTV (*High Definition Television*), ainsi les systèmes de communication 4G, a accru l'intérêt pour de nouvelles techniques de transmission afin de relever un tel défi.

Une des techniques de modulation les plus utilisées dans les systèmes de communication sans fil est l'OFDM, aussi appelée DMT (*Discrete Multi Tone*) dans le cas des

communications filaires. Cette technique de transmission possède une grande efficacité spectrale et semble être la plus adaptée à la demande en termes de débit.

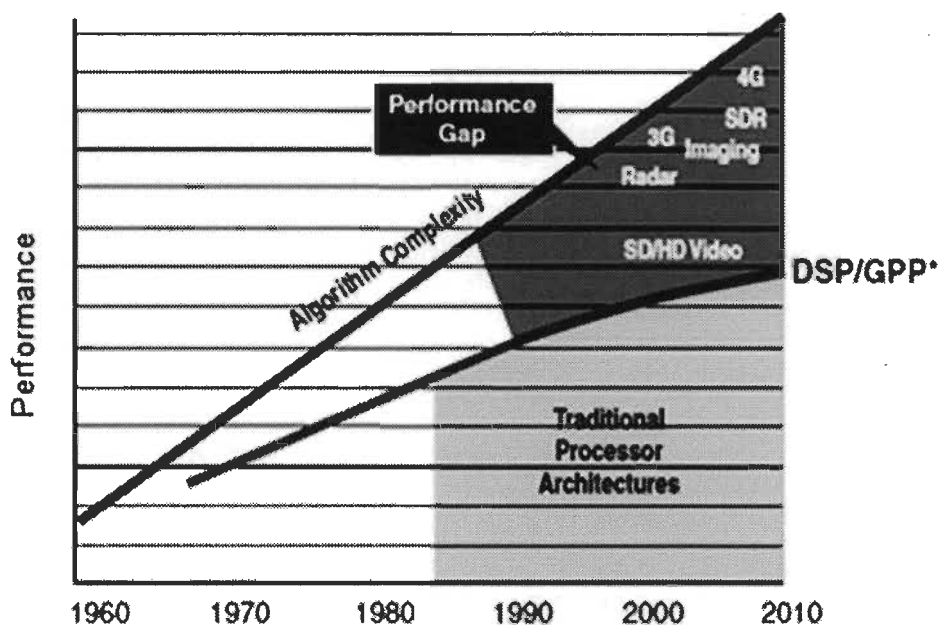


Figure 1.3: Pr evision de la complexit e algorithmique et des performances des processeurs [DSP05]

La synth ese du signal OFDM est typiquement r ealis ee en utilisant la transform ee de Fourier Rapide inverse (IFFT)  a la transmission et directe  a la r eception permettant de simplifier l' egalisation et par cons equent r eduire la complexit e du r ecepteur. G en erale­ment, les modules de FFT/IFFT exigent beaucoup de ressources mat erielles,  a savoir les registres, les additionneurs/soustracteurs et les multiplieurs le tout en arithm etique  a valeur complexe. De plus, dans la technologie VLSI (*Very Large Scale Integration*), l'augmentation du nombre d'op erations arithm etiques entra enera l'augmentation de la consommation d' energie ce qui rend l'impl ementation des modules FFT/IFFT tr es co uteuse. Par ailleurs, chaque op eration arithm etique est r ealis ee en virgule fixe et engendre une erreur due  a la

troncature, appelée erreur au bruit de quantification. Ce dernier est mesuré par le rapport signal sur bruit de quantification (*SQNR: Signal Quantization Noise Ratio*). Respectant les contraintes d'implémentation en technologie VLSI, telles que la faible consommation en puissance et en surface d'implémentation, l'exécution en arithmétique à virgule fixe des transmetteurs-récepteurs s'avère essentielle. Ainsi, une solution efficace de l'implémentation de la FFT consiste à optimiser la largeur binaire des coefficients de Fourier. Pour ce faire, dans un premier temps, une solution est présentée en utilisant les algorithmes génétiques afin de représenter les coefficients de Fourier en tenant compte des contraintes d'implémentation. Dans un second temps, nous comptons implémenter la FFT radix-4 conventionnelle de type MDC (*Multi-path Delay Commutator*) sur FPGA (*Field Programmable Gate Array*). Par la suite, nous estimerons la consommation de la puissance consommée en utilisant le Xpower analyzer de Xilinx (XPA).

I.2. OBJECTIFS DU SUJET DE RECHERCHE

L'objectif principal de ce projet consiste à proposer une méthode d'optimisation en virgule fixe d'un processeur IFFT/FFT basé sur les algorithmes génétique pour avoir de bonnes performances en termes de probabilité d'erreur avec une faible complexité du traitement, ainsi qu'une faible consommation d'énergie. Pour ce faire, quatre sous-objectifs peuvent être envisagés :

1. Études de l'effet de quantification de l'algorithme Radix-4 FFT de type MDC.
2. Développement des algorithmes d'optimisation de la largeur binaire pour les coefficients de la FFT. A cet effet l'utilisation des algorithmes génétique pour

l'optimisation de la qualité de signal vs compromis de complexité d'implémentation est appliquée dans notre étude.

3. Synthèse et implémentation de la FFT pipeline radix-4 mdc à 2 étages sur FPGA suivie par l'étude d'estimation de la consommation en puissance.

I.3. MÉTHODOLOGIE DU SUJET DE RECHERCHE

L'ensemble des travaux de ce mémoire a été réalisé au laboratoire LSSI (Laboratoire de Signaux et Systèmes Intégrés) de l'Université du Québec à Trois-Rivières.

Afin d'atteindre nos objectifs, notre méthodologie de travail consiste dans un premier temps, de faire une étude approfondie de la littérature des algorithmes et les différentes architectures pipelines pour le calcul de la FFT. Cette étude nous a fourni un paquet d'informations qui caractérisent chaque algorithme FFT (complexité, quantité de mémoire...). Basé sur cette étude, nous allons choisir l'algorithme radix-4 et son architecture pipeline du type MDC pour le calcul de la FFT. Cependant, son étude est nécessaire pour bien saisir son bon fonctionnement. Ensuite, nous programmons la FFT radix-4 MDC conventionnelle sur Matlab® et nous comparons les résultats en virgule flottante et celle en virgule fixe obtenue en utilisant la fonction fi^3 de Matlab®.

Le module IFFT/FFT est une composante non négociable en complexité dans les systèmes de télécommunication mobiles. Ces performances en termes de puissance, la surface et la vitesse d'exécution sont affectées par la largeur binaire des données et des coefficients de FFT [SUL04].

³ Une fonction Matlab permettant la conversion virgule flottante en virgule fixe

CHAPITRE I – INTRODUCTION

Nous étudions les algorithmes génétiques comme outil d'optimisation des coefficients de la FFT. Les bons choix des paramètres qui commandent ces méthodes telles que, le choix de codage des chromosomes, des opérateurs génétiques (sélection, croisement, et la mutation), le critère d'arrêt et la fonction sélective sont les opérations principales pour avoir des résultats optimaux de cet algorithme. La mise œuvre des algorithmes génétiques appliqués pour concevoir un nouveau processeur FFT, permettra également de programmer cette méthode sur Matlab® et de l'appliquer pour l'optimisation des coefficients de la FFT en tenant compte des contraintes d'implémentation VLSI telle que, la complexité de calcul, et la surface d'implémentation. L'étape suivante représente l'implémentation matérielle pour l'estimation en puissance. Dans un premier temps, nous implémentons les blocs de la FFT pipeline de type radix-4 MDC en VHDL. L'outil d'implémentation et de simulation est Modelsim PE Student® de Mentor Graphics®. Secondement, une fois le code VHDL est vérifié, nous passons à la synthèse de bloc FFT sur FPGA en utilisant le logiciel Xilinx ISE. Ces outils de simulation fournissent la puissance consommée en utilisant le XPower Analyzer.

I.4. ORGANISATION DE MÉMOIRE

Le reste de mémoire est structuré comme suite :

Le chapitre II met en évidence l'intérêt de la modulation OFDM et examine la DFT comme l'élément de base de la chaîne OFDM. Ce chapitre introduit une revue sur le concept de la DFT et son développement mathématique. Ainsi le modèle mathématique de la FFT radix-4 MDC et son architecture pipeline seront examinés. Le chapitre III discutera

CHAPITRE I – INTRODUCTION

les étapes des algorithmes génétiques appliqués pour l'optimisation de la FFT. Ce chapitre expose également quelques résultats obtenus pour l'optimisation des coefficients de la FFT. Dans le chapitre VI nous discuterons l'implémentation de la FFT radix-4 MDC sur FPGA (Simulation VHDL, Synthèse FPGA). Enfin, le chapitre V présentera la conclusion générale du projet.

CHAPITRE II

FFT QUANTIFIÉES AU RÉCEPTEUR AU OFDM

Nous commençons ce chapitre par la présentation de la technique OFDM comme technique robuste pour les systèmes de communication à haut débit. Son principe de fonctionnement et les éléments constitutants de la chaîne OFDM sont présentés aussi. Ensuite, un point est fait sur la fiabilité de ces composants, en particulier sur les éléments sensibles comme la FFT et IFFT. Après, nous exposerons l'histoire de la FFT, et une étude approfondie sur l'algorithme FFT adopté dans ce travail. Nous concluons cette partie par la représentation des nombres en virgule fixe et l'effet de quantification

II.1. LES SYSTÈMES MULTI-PORTEURS OFDM

L'OFDM est une technique de transmission multi-porteuse consiste à répartir l'information à transmettre à haut débit sur plusieurs sous-porteuses, modulées à bas débit. L'intérêt de cette modulation est l'exploitation optimale du spectre, grâce à la propriété d'orthogonalité des porteuses, ce qui augmente l'efficacité spectrale du système.

II.1.1. Historique – Principe

L'idée de l'OFDM est de diviser une bande fréquentielle du signal transmise en un groupe de bandes adjacentes, cette idée a pu être dépistée à la fin des années 50 avec la société Collins Radio Co. Kineplex system [DOE57]. Dans le multiplexage fréquentiel classique (FDM), la bande totale est divisée en N sous canaux qui ne se chevauchent pas, alors que dans l'OFDM la bande est divisée en un certain nombre de sous canaux superposés avec des fréquences orthogonales.

La figure 2.1 illustre l'efficacité de la largeur de bande requise pour un signal OFDM par rapport à un signal FDM. En utilisant la modulation OFDM, nous économisons près de 50% de la bande passante. Pour réaliser la technique de modulation à porteuses multiples avec chevauchement, il est toutefois nécessaire de réduire l'interférence entre les sous porteuses. Ceci est réalisé par l'orthogonalité fréquentielle des sous porteuses [PRA04].

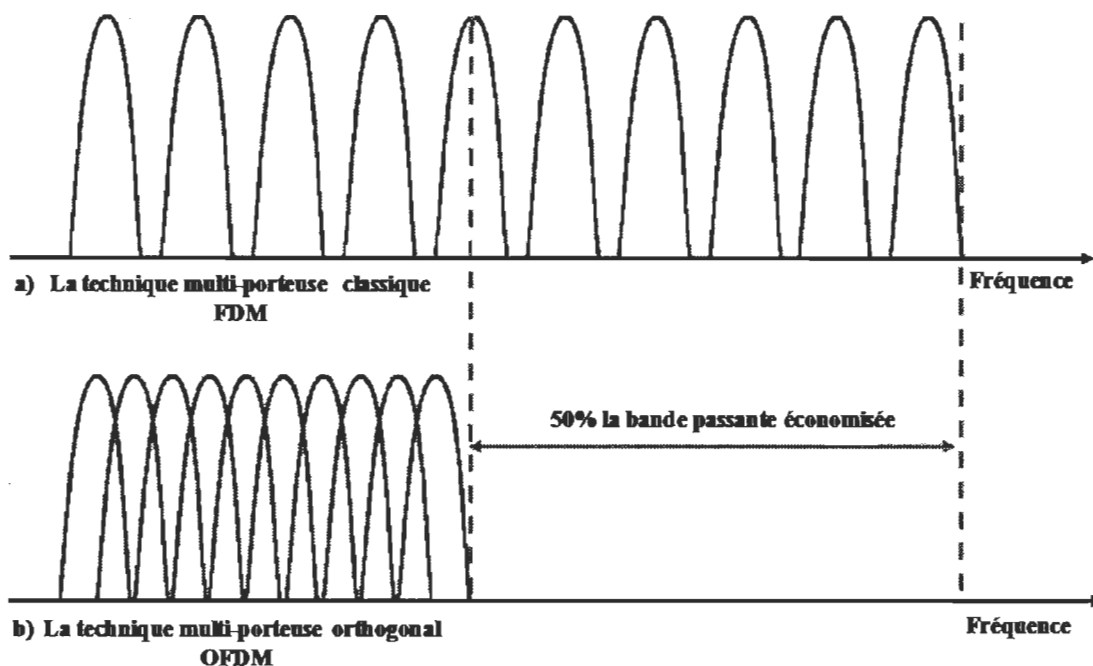


Figure 2.1: Multiplexage de fréquence a) Classique FDM b) Orthogonale OFDM [PRA04]

L'ensemble des sous-porteuses forme un symbole OFDM. Grâce à l'orthogonalité de l'OFDM, les différentes sous-porteuses se chevauchent dans le domaine fréquentiel, mais sans causer d'interférence entre sous-porteuses ICI (Inter-carrier-Interference). Cette propriété rend ce système robuste contre le problème des multi-trajets.

L'utilisation des porteuses dont le spectre est un sinus cardinal a permis une orthogonalité entre elles pour éviter l'interférence entre canaux [CHA66]. Peu après, B. Saltzberg a expérimenté la performance d'un tel système [SAL67]. En 1971, S. B. Weinstein et P. M. Ebert [WEI71] introduisent l'idée d'utiliser une DFT pour la mise en œuvre de la génération et la réception des signaux OFDM. Après avoir été certifiée, la technique OFDM a été longtemps mise à l'écart des applications commerciales en raison de la complexité que

revêt son implantation. Plus particulièrement, le fait de réaliser la transformée de Fourier en temps réel. Cette opportunité présente une implémentation facile de l'OFDM, spécialement avec l'utilisation de la transformée de Fourier rapide (FFT), qui est une implémentation de la DFT. Pour la première fois, les applications de cette technique ont été proposées en 1985 pour la radiophonie mobile [CIM85], et plus tard implémentaient pour la diffusion numérique [ALA87].

En 1989, cette technique a pris le nom « Modulation à Répartition en Fréquences Orthogonales » (OFDM) [ZER89]. Dans le contexte de la téléphonie mobile, l'OFDM peut être utilisée en combinaison avec d'autres formes d'accès multiple comme le FDMA, le TDMA et le CDMA pour donner lieu, respectivement, aux systèmes FDMA multi-porteuse (MCFDMA), TDMA multi-porteuse (MC-TDMA), et CDMA multi-porteuse (MC-CDMA). Les premières idées pour utiliser l'OFDM en combinaison avec le CDMA ont été présentées dans [YEE93] et [FAZE93].

II.1.2. Modélisation des systèmes OFDM

L'OFDM peut-être modélisé de plusieurs manières, et la représentation de ce type de système a évolué au cours du temps avec les innovations technologiques. Nous présenterons donc en premier lieu la représentation continue du système OFDM de laquelle nous ferons surgir la modélisation discrète en bande de bases, ainsi qu'une modélisation bidimensionnelle dans le plan temps-fréquence.

II.1.2.1. Modulations OFDM

Le principe de la modulation multi-porteuse de types OFDM consiste à transmettre les données de manière simultanée sur N porteuses. Elle peut être modélisée de plusieurs manières. La figure 2.2 décrit le schéma de principe d'un modulateur OFDM en bande de base.

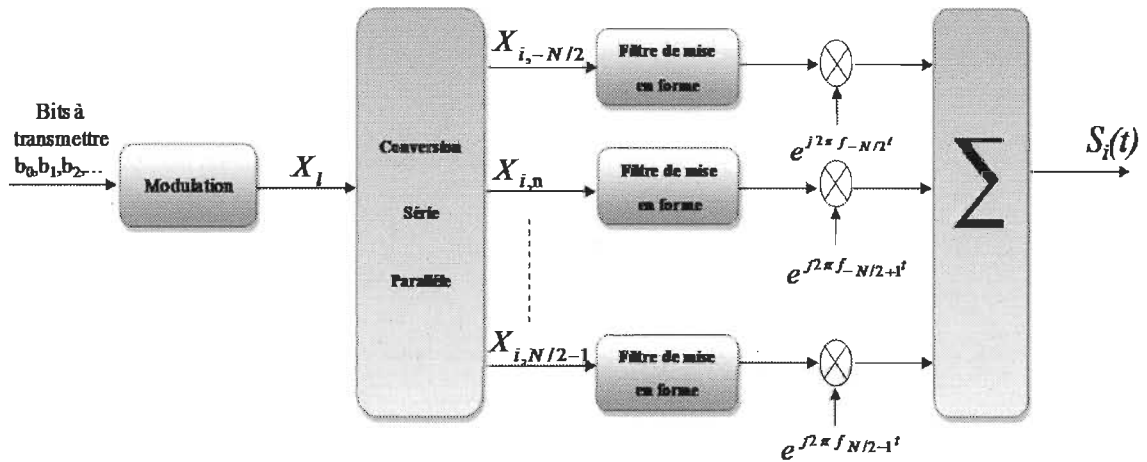


Figure 2.2: Schéma de principe d'un modulateur OFDM.

Considérons des données binaires (b_0, b_1, b_2, \dots) de période T_b . Ces données seront transformées en symboles complexes $\{X_i\}$ en utilisant des modulations numériques, telles que M-QAM de durée $T_i = \log_2 MT_b$, où M est la taille de la constellation. Le convertisseur série-parallèle dispose les symboles complexes $\{X_i\}$ en groupe de N symboles, qui sont mis en forme sur une durée de symbole $T = N T_b$, puis envoyés sur les N porteuses $\{f_n\}$ afin de former le signal de sortie [TAI09].

Durant l'intervalle du temps $[iT, (i+1)T]$, le signal OFDM généré s'écrit [MOL11] :

$$S_i(t) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} X_{i,n} \underbrace{g(t-iT)}_{\psi_{i,n}(t)} e^{j2\pi f_n t} \quad (2.1)$$

où $g(t)$ est la forme d'onde de la modulation. $S_i(t)$ représente le $i^{\text{ième}}$ symbole OFDM, tandis que les $\{X_{i,n}\}$ représentent les symboles complexes transmis sur le $n^{\text{ième}}$ porteuse et le $i^{\text{ième}}$ symbole OFDM de durée T . La fréquence inter porteuse f_n est ici égale à $1/T$. L'équation (2.2) décrit le signal reçu pour le cas d'un canal de transmission sans bruit additif [MOL11].

$$y(t) = \sum_{i=-\infty}^{\infty} S_i(t) = \sum_{i=-\infty}^{\infty} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} X_{i,n} \psi_{i,n}(t) \quad (2.2)$$

où $\{\psi_{i,n}(t)\}_{i,n}$ est la base orthonormale de l'espace des signaux.

Les symboles transmis peuvent être retrouvés en réception à l'aide d'un filtre adapté suivi d'un échantillonneur. La figure 2.3 illustre le schéma de principe d'un récepteur OFDM.

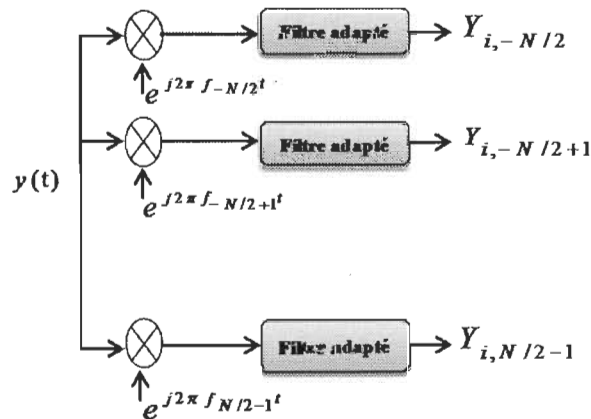


Figure 2.3: Schéma de principe d'un démodulateur OFDM.

II.1.2.2. Orthogonalités

La propriété d'orthogonalité est fondamentale en OFDM. Pour que le signal modulé autorise un fort recouvrement spectral entre les sous-porteuses, il faut que les fréquences des porteuses soient les plus proches possible, tout en garantissant que le récepteur soit capable de les séparer et retrouver le symbole numérique émis sur chacune d'entre elles. Ceci est vérifié si le spectre d'une porteuse est nul aux fréquences des autres porteuses (voir la figure 2.1 (b), et la figure 2.5).

En considérons le signal OFDM donnée par l'équation 2.1. $\psi_{i,n}(t)$, $\psi_{i,n'}(t)$ sont des porteuses orthogonales si :

$$\int_t^{t+T} \psi_{i,n}(t) \overline{\psi_{i,n'}(t)} dt = \begin{cases} 0, & n \neq n' \\ T & n = n' \end{cases} \quad (2.3)$$

L'équation 2.3 démontre que l'ensemble des N sous-porteuses d'une trame OFDM sont orthogonales. L'orthogonalité temporelle de la fonction $\psi_{i,n}(t)$ doit être vérifiée dans la mise en œuvre d'un signal OFDM.

Lors des travaux de Chang [CHA66] ont permis de démontrer que l'orthogonalité de la fonction $\psi_{i,n}(t)$ se traduit par des conditions sur le module et la phase de $g(t)$. Parmi les fonctions disponibles, celle qui est la plus utilisée est la fonction porte $g(t) = \text{Rect}[0, T]$. Une simple impulsion rectangulaire élaborée est représentée dans la figure 2.4 (a). Cela

correspond à une Sinc-carrée formée par sous-porteuse du spectre, comme indiqué dans la figure 2.4 (b) [DAH11].

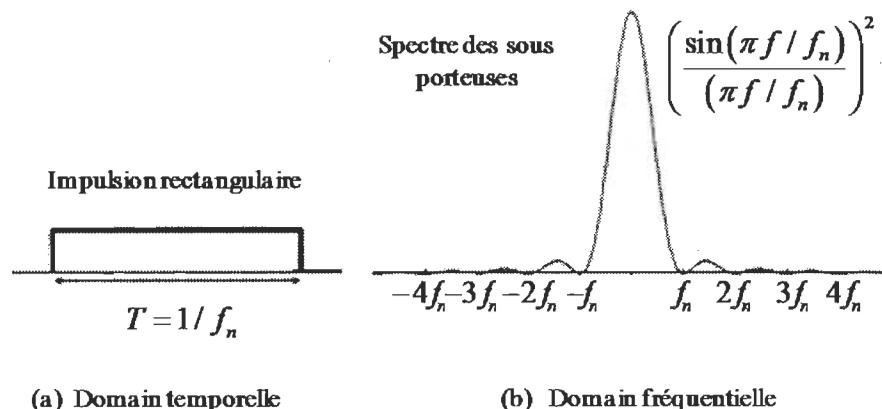


Figure 2.4: (a) la forme des impulsions par sous-porteuse. (b) Spectre pour la transmission OFDM basic [DAH11]

Supposons que les symboles émis sont de moyenne nulle et de variance σ^2 , le spectre du signal modulé sur la porteuse n s'écrit de la façon suivante:

$$S_n(f) = \frac{\sigma^2}{T} \text{sinc}^2[\pi(f - f_n)T] \quad (2.4)$$

Ce spectre s'annule aux fréquences $f = \{f_n + k/T\}$ pour tout entier k avec f_n est la fréquence inter porteuse. $1/T$ est donc l'espacement inter porteuse minimal qui garantit à la fois l'orthogonalité entre les porteuses et une efficacité spectrale optimale. Les spectres des différentes porteuses sont présentés sur la figure 2.5 [TAI09] [DAH11]. Le spectre d'un signal OFDM est la somme de tous ces spectres. Notons que lorsque la mise en forme est une fonction rectangulaire de longueur T , les filtres adaptés en réception sont des intégrateurs sur la durée T .

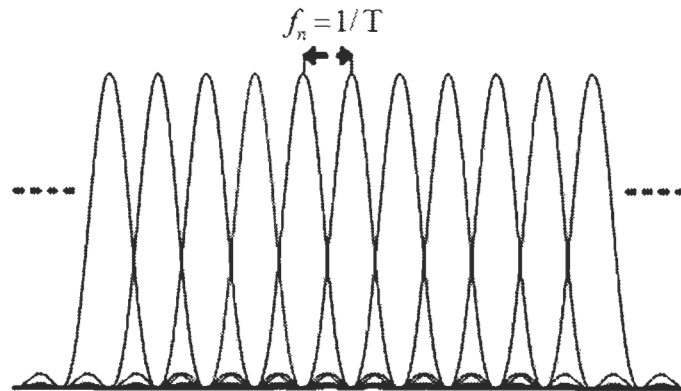


Figure 2.5: Spectre du signal en sortie du modulateur OFDM, décomposé sur chaque porteuse. [DAH11]

II.1.3. La FFT dans les communications OFDM

D'un point de vue de l'implémentation pratique de l'orthogonalité, la modulation OFDM se fait avec une IDFT/DFT. Ceci est très intéressant puisque les opérations de la IDFT et DFT peuvent s'implanter de façon efficace en utilisant les algorithmes de transformée de Fourier rapide IFFT et FFT. Ces algorithmes permettent de passer d'un nombre de multiplications à (N^2) vers un nombre de multiplications à $O(N \log_2(N))$, ce qui est très intéressant d'un point de vue matériel.

II.1.3.1 Implémentation numérique de la Modulation/Démodulation OFDM

La réalisation analogique d'un modulateur OFDM est très complexe car il faut utiliser un banc de modulateurs/démodulateurs synchronisés et un banc de filtres de mise en forme/filtres adaptés avec un grand nombre de voies. En effet, lorsque les porteuses sont orthogonales et que : $f_n = \frac{n}{T}$ pour $n = -\frac{N}{2} \dots \frac{N}{2} - 1$, le signal généré en bande de base dans

l'intervalle de temps $[iT, (i+1)T]$ peut s'écrire de la façon suivante l'équation (2.5)

[TAI09].

$$S_i(t) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} X_{i,n} e^{j2\pi \frac{nt}{T}} \quad (2.5)$$

Nous discrétisons ce signal, nous obtenons :

$$S_i(k) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} X_{i,n} e^{j2\pi \frac{nk}{N}} \quad (2.6)$$

$\{S_i(k)\}$, où $k = -\frac{N}{2} \dots \frac{N}{2} - 1$ Correspond aux N échantillons du $i^{\text{ième}}$ symbole OFDM. Nous

les noterons $\{S_{i,k}\}$. Ils peuvent être obtenus grâce à une transformée de Fourier discrète

inverse des symboles $\{X_{i,n}\}$ à transmettre. La figure 2.6 représente un modulateur OFDM.

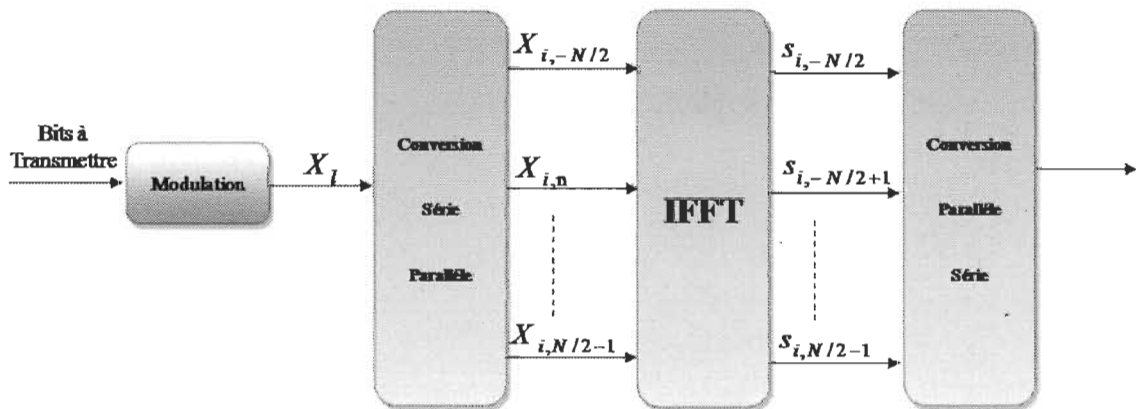


Figure 2.6: Schéma de principe du modulateur OFDM numérique

A la réception, la procédure inverse est appliquée. La démodulation consiste à effectuer une transformée de Fourier discrète directe des symboles reçus. Ceci peut être réalisé à l'aide de l'algorithme de la FFT. La figure 2.7 décrit le schéma d'un démodulateur OFDM numérique. Nous notons $\{Y_{i,n}\}$ les symboles reçus après la FFT du récepteur, correspondant aux symboles émis $\{X_{i,n}\}$ placés avant l'IFFT de l'émetteur.

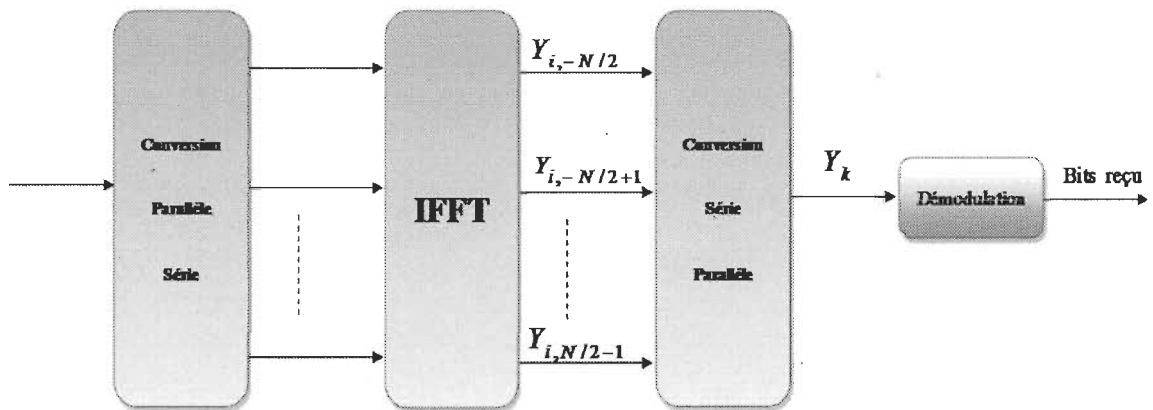


Figure 2.7: Schéma de principe d'un démodulateur OFDM numérique

II.1.3.2. Quelques standards OFDM à base FFT

La technique de modulation OFDM est robuste contre la sélectivité en fréquence et au bruit impulsif du canal de propagation, ce qui permet d'atteindre des débits élevés. La modulation OFDM a émergé comme un signal attrayant pour le modem de communication sans fil large bande telle que les 3GPP LTE, WIMAX, les réseaux locaux sans fil, la télévision numérique (DVB), et a récemment été considéré pour les communications acoustiques sous-marines (UWA: Underwater Acoustics) et les réseaux sans fil de la 4^{ème} génération.

CHAPITRE II – FFT QUANTIFIÉES AU RÉCEPTEUR OFDM

Le tableau ci-dessous, résume les paramètres clés de quelques systèmes de communication multi-porteuses.

Tableau 2.1: Paramètre FFT dans différentes normes OFDM [TSA11]

Technologie	Taille de la FFT	Taux d'échantillonnage (MHz)
DVB-T/H	2048-8192	9
802.11a	64	20
802.11n	64-128	40
UWB (MB-OFDM)	128	40
802.16 ^e (OFDM)	256	32.7
802.16 ^e (OFDMA)	128-2048	20
3GPP-LTE	128-2048	30.7
802.20	512-2048	20

II.1.2. Avantages et inconvénients de l'OFDM

L'utilisation de la technique OFDM pour les systèmes de communications sans fil conduit à certains avantages et inconvénients. La décision d'utiliser une telle technique est toujours basée sur l'évaluation du rapport coût/performances. Dans certains scénarios tels les communications mobiles intérieures, la réalisation de la technique OFDM s'avère avantageuse sinon la communication n'est pas fiable.

Les principaux avantages de l'OFDM sont les suivants:

- Une haute efficacité spectrale et une grande robustesse à l'évanouissement sélectif en fréquence.
- Une réalisation numérique simple par utilisation d'IFFT/FFT. cela a conduit à son utilisation massive dans plusieurs standards
- Réduction de la complexité des récepteurs évitant les ISI et ICI grâce à l'ajout d'un intervalle de garde.
- L'OFDM permet une égalisation simple grâce à l'ajout du préfixe cyclique, même en présence de canaux multi trajets denses.

Mais le système utilisant la technique d'OFDM n'est pas parfait. L'OFDM possède néanmoins des inconvénients qu'il est important d'appréhender :

- Les signaux multi porteuses ont un coefficient PAPR (*Peak to Average Power ratio*) élevé, ce qui nécessite l'utilisation des amplificateurs à haute linéarité.
- La perte dans l'efficacité spectrale due à l'addition d'un intervalle de garde.
- La sensibilité à l'effet Doppler : quand le récepteur est en mouvement relatif par rapport à la source, la fréquence du signal reçu ne sera pas la même que celle du signal émit. La fréquence du récepteur est supérieure à celle de l'émetteur et diminue au fur et à mesure qu'ils se rapprochent entre eux. Ceci s'appelle « l'effet Doppler ».
- Si le récepteur OFDM est mal synchronisé temporellement, un phénomène d'interférence entre symboles OFDM peut intervenir dégradant considérablement les performances du système global.

II.2. ALGORITHME FFT ET L'EFFET DE QUANTIFICATION

Dans cette partie, nous commençons par un aperçu de la DFT qui est utilisée pour produire l'analyse de fréquence des signaux périodiques non discrets. Par la suite, nous présentons la FFT comme une autre méthode pour atteindre le même résultat de la DFT tout en apportant moins de complexité de calculs. Enfin, le reste de cette partie se concentre sur la représentation des nombres en virgule fixe et l'effet de quantification.

II.2.1 La Transformée de Fourier Discrète

Un signal numérique dans le temps peut être transformé dans le domaine fréquentiel au moyen de la transformée en Z ou de la transformée de Fourier [PRO96]. De manière générale, un signal périodique $x(t)$ peut se représenter sous la forme d'une somme de signaux sinusoïdaux, selon l'expression (2.7) appelée également transformée de Fourier inverse :

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad (2.7)$$

où :

- $X(f)$ est un nombre complexe définissant l'amplitude et la phase de chaque composante de fréquence. Le terme $X(f)$ constitue la transformée de Fourier du

signal temporel $X(t)$ éventuellement complexe, sa définition est donnée par l'équation (2.8) :

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \quad (2.8)$$

L'expression (2.8) correspond à la forme continue de la transformée de Fourier. Le calcul de cette dernière sur une série échantillonnée conduit alors à la définition de la DFT noté $X[k]$. Ainsi, pour une série $x(n)$ constituée d'un nombre N , de valeurs finie, l'expression de la DFT d'un signal $x(n)$ est définie dans l'équation (2.9) [PRO96] [RAO10]:

$$X[k] = \sum_{n=0}^{N-1} x(n) e^{-j2\pi k \frac{n}{N}}, \quad 1 \leq k \leq N \quad (2.9)$$

- n et k représentent respectivement les variables discrètes et normalisées de l'espace original et de l'espace transformé.
- N est une puissance de 2, représente le nombre des valeurs discrètes et successives des variables n et k . Et sont inverse associé; IDFT qui effectue l'opération inverse, c'est-à-dire qu'elle convertit le spectre fréquentiel $X[k]$ dans le domaine du temps échantillonné $x(n)$, est donnée par l'équation (2.10) :

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi k \frac{n}{N}}, \quad 1 \leq n \leq N \quad (2.10)$$

- $x(n)$ représente le vecteur temporel $X[k]$ désigne le vecteur fréquentiel,

- N représente la longueur de la DFT/IDFT, et $1/N$ est un facteur de pondération utilisé dans l'expression de la IDFT afin d'obtenir les mêmes coefficients que dans la décomposition en série de Fourier.

aussi :

$$W_N^{kn} = e^{-j2\pi k \frac{n}{N}} = \cos\left(\frac{2\pi k n}{N}\right) + j \sin\left(\frac{2\pi k n}{N}\right) \quad (2.11)$$

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

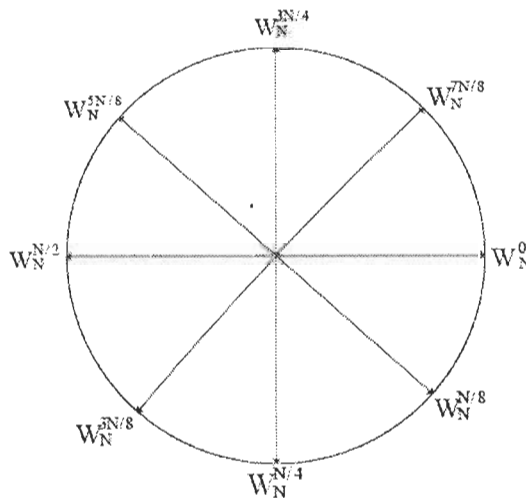


Figure 2.8: Les coefficients de la DFT pour $N=8$

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

La matrice de la DFT est composée d'exponentielles complexes, elle est définie comme suite :

$$X[k] = \underbrace{\begin{bmatrix} 1 & 1 & 1 & \dots & w^0 \\ 1 & e^{-j2\pi\frac{1}{N}} & e^{-j2\pi\frac{2}{N}} & \dots & e^{-j2\pi\frac{N-1}{N}} \\ 1 & e^{-j2\pi\frac{2}{N}} & e^{-j2\pi\frac{4}{N}} & \dots & e^{-j2\pi\frac{2(N-1)}{N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi\frac{N-1}{N}} & e^{-j2\pi\frac{1}{N}} & \dots & e^{-j2\pi(N-1)\frac{N-1}{N}} \end{bmatrix}}_{\text{La matrice DFT}} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ \vdots \\ \vdots \\ x(N-1) \end{bmatrix} \quad (2.12)$$

Ce qui peut être représenté sous la forme matricielle suivante :

$$X[k] = [B_N][x_N] \quad (2.13)$$

Où : $[B_N]$ la matrice des coefficients et $[x_N]$ la matrice d'entrée de la DFT.

Le calcul direct d'une DFT correspond à la sommation d'un produit pour chaque point. Ceci est équivalent à une quantité de calculs considérable. En effet, si N est élevé, selon les formules données en (2.9) et (2.10), la DFT revient à calculer un produit matrice-vecteur où chaque élément est de type complexe. La complexité de calcul de la DFT est donc de : N^2 multiplications et $N(N-1)$ additions sur des nombres complexes. Ceci revient à une complexité de $4N^2$ multiplications réelles et $N(4N-2)$ additions réelles. Par conséquent, la complexité de l'algorithme est de $O(N^2)$, ceci demeure élevé en quantité de ressources

matérielles nécessaire. Donc, en raison de l'implémentation de traitement numérique de signal en temps réel, optimiser ce calcul devient très avantageux [PRO96] [RAO10].

Une autre forme de la DFT est utilisée pour les processus de modulation et de démodulation, appelée la transformée de Fourier rapide (FFT), qui est un algorithme de DFT développé en 1965 [COO65]. Cette transformée rapide réduit considérablement le temps de calcul et la complexité arithmétique.

II.2.2. La Transformée Rapide de Fourier

De nombreux travaux ont essayé de réduire la complexité de la DFT. En effet, Cooley et Tukey introduisent en 1965 le premier algorithme de la FFT qui permet de réduire considérablement le temps de calcul de la DFT d'une suite dont le nombre d'échantillons N est décomposable en facteurs typiquement, une puissance de 2, connu sous le nom de Radix-2 [COO65]. Cette publication allait engendrer de nombreuses recherches sur les algorithmes de calculs des transformées. Depuis, et encore de nos jours, les algorithmes de FFT ont révolutionné le traitement numérique de signal (Digital Signal Processing : DSP). Leur efficacité réside dans la réduction du nombre d'opérations nécessaires, en particulier le nombre des multiplications ce qui réduit davantage le temps d'exécution.

Tous ces algorithmes sont basés sur un même principe qui consiste à décomposer le calcul de la DFT en plusieurs sous ensemble de DFT de longueur plus petite. La mise en œuvre de ce principe conduit à différentes méthodes.

En profitant des propriétés de symétrie et de périodicité suivant les équations (2.14) et (2.15) des facteurs de phases W_N^k .

$$\begin{aligned} \text{Symétrie :} \quad & W_N^{k+\frac{N}{2}} = -W_N^k \\ & (W_N^k)^* = W_N^{-k} \end{aligned} \quad (2.14)$$

$$\text{Périodicité :} \quad W_N^{k+N} = W_N^k \quad (2.15)$$

Où (*) désigne le complexe conjugué.

Nous pouvons réduire significativement les nombres d'opérations arithmétiques. Cependant, la complexité globale reste $O(N^2)$ [OPP75].

Les algorithmes de FFT permettent de faire une DFT de manière efficace ainsi que de réduire la charge de calcul en termes de multiplications et additions complexe à l'ordre $O(N \log_2 N)$. En effet, il existe de nombreux algorithmes FFT qui découlent tous de l'approche diviser pour régner (en anglais : divide-and-conquer approach). L'idée de cette approche consiste à représenter les vecteurs $X[k]$ et $x(n)$ sur deux dimensions (cas du radix-2, radix-4, radix-8, etc ...) ou bien plusieurs dimensions (cas du radix- 2^i , radix- 4^i).

II.2.3. Approche diviser-pour-régner

L'algorithme de la FFT et ses variantes sont détaillées dans de nombreux ouvrages, notamment (Proakis et Manolakis) [PRO96]. Le principe de base de l'algorithme consiste à faire un changement de variable de l'indice n sur deux dimensions entières. Par exemple, avec M pour les colonnes et L pour les lignes, on a :

$$N = ML \quad (2.16)$$

$$n = Ml + m; \quad 0 \leq l \leq L-1, \text{ et } 0 \leq m \leq M-1 \quad (2.17)$$

$$k = Mp + q; \quad 0 \leq p \leq L-1, \text{ et } 0 \leq q \leq M-1 \quad (2.18)$$

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

$$X[p, q] = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} X[l, m] W_N^{(Mp+q)(mL+l)} \quad (2.19)$$

Le terme $W_N^{(Mp+q)(mL+l)}$ peut être simplifié de la manière suivante :

$$W_N^{(Mp+q)(mL+l)} = W_N^{MLmp} W_N^{mLq} W_N^{MPl} W_N^{lq} \quad (2.20)$$

Toutefois, $W_N^{MLmp} = 1$, $W_N^{mLq} = W_M^{mq} = W_{(N/L)}^{mq}$ et $W_N^{MPl} = W_L^{lp} = W_{(N/M)}^{lp}$

De là on obtient l'équation :

$$X[p, q] = \sum_{l=0}^{L-1} \underbrace{\left\{ W_N^{lq} \left[\underbrace{\sum_{m=0}^{M-1} X(l, m) W_M^{mq}}_{\text{DFT à M point}} \right] \right\}}_{\text{Rotation}} \underbrace{\left. \right\}}_{\text{DFT à L points}} W_L^{lp} \quad (2.21)$$

L'équation (2.21) illustre le fonctionnement de la FFT. Soit la décomposition de la FFT à N points, en L FFT à M points. Pour résoudre l'équation (2.21), nous procédons d'abord à des

TFD de M points correspondant à la sommation à l'intérieur entre crochets. Ensuite, la matrice résultante subit une multiplication point à point par les facteurs de phase W_L^{lp} et enfin des TFD sur L points sont effectuées en suivant l'autre dimension. En effectuant ces étapes, la complexité passe de l'ordre N^2 à l'ordre $N(M+L)$.

Nous pouvons aussi récursivement faire d'autres changements de variables, en prenant un nombre premier pour L. Par la suite, nous factorisons M jusqu'à obtenir seulement des TFD ayant des tailles de nombre premier [PRO96].

L'algorithme diviser pour régner peut se résumer dans les étapes suivantes [PRO96]:

1. Calcul de M-point DFT, $F(l,q)$ de chaque ligne selon l'équation (2.22).

$$F(l, q) = \sum_{m=0}^{M-1} X(l, m) W_M^{mq} \quad (2.22)$$

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

$$G(l, q) = W_N^{lq} F(l, q) \quad (2.23)$$

3. Finalement, calcul de L-points DFT de chaque colonne selon l'équation (2.24):

$$X(p, q) = \sum_{l=0}^{L-1} G(l, q) W_L^{lp} \quad (2.24)$$

L'algorithme peut aussi être représenté graphiquement, en utilisant un exemple d'une FFT à 15 points, avec $m = 3$ et $l = 5$. La figure 2.9 explique la méthodologie utilisée :

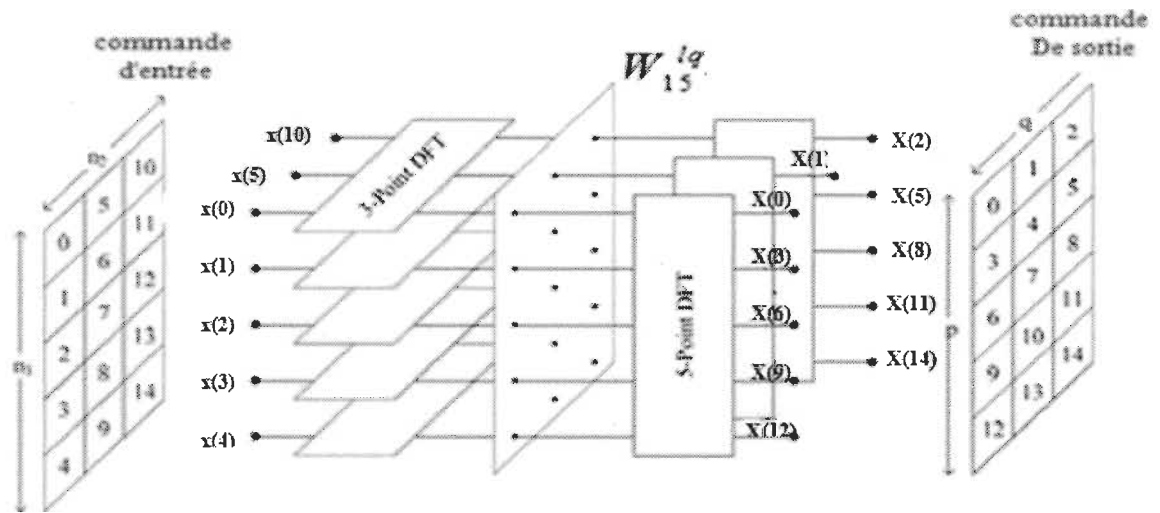


Figure 2.9: Décomposition de 15 points FFT en 3 points et 5 points TFD [PRO96]

II.3. ARCHITECTURE PIPELINE POUR LE CALCUL DE LA FFT

L'algorithme FFT est simplement une DFT calculée selon un algorithme permettant de réduire le nombre d'opérations et, en particulier, le nombre de multiplications à effectuer. Cependant, le bon choix de l'architecture qui exécute le traitement, permet de réduire le nombre d'opérations arithmétiques et le temps d'exécution. Il existe une variété d'architecture pour la réalisation de la FFT proposée pour l'amélioration de la vitesse de calcul et la puissance consommée. Il s'agit principalement des architectures à base de mémoires [CHN04][BAA99], en pipeline [SHO98], et les architectures parallèles [ZHA90].

Tableau 2.2: Comparaison entre les architectures des processeurs FFT

Architecture	Avantages	Désavantages
Architecture pipeline	1- Débit élevé 2- Régularité	coût d'implémentation élevé
Architecture à base de mémoire	Faible coût d'implémentation	Faible débit

Parmi les aspects les plus importants dans l'étude des performances que pourrait obtenir un modulateur OFDM à très haut débit multi-standard, est la réalisation matérielle de la FFT. Chacun des algorithmes considérés pour la FFT peut être réalisé par différents types d'architectures. Par conséquent, une conception performante en termes de débit et de puissance consommée passe par une bonne adéquation algorithme-architecture. L'architecture pipeline est adoptée dans le présent travail, car elle est certainement la plus appréciée pour les systèmes MIMO-OFDM (MIMO-OFDM : Multi Input Multi Output OFDM) et peut atteindre un rendement relativement élevé avec des coûts modérés du matériel.

II.3.1. Classification des algorithmes FFT

Les algorithmes FFT peuvent être classés en trois catégories, à savoir radix-r, split-radix et radix-mixte. Pour chaque algorithme, le principe fondamental de la FFT repose sur un processus de décomposition de la DFT à calculer en DFT plus courte. Selon la manière

dont ce principe est implanté, nous considérons la décomposition dans le temps connus aussi sous le nom de l'entrelacement temporel (*Decimation-In-Time DIT*), et celle basée sur la décomposition dans le domaine de la fréquence, est aussi appelé l'entrelacement en fréquence (*Decimation-In-Frequency DIF*). Ces deux transformations reposent sur une décomposition récursive d'une transformée de N points en une séquence successive plus petites.

II.3.2. Papillon radix- r FFT

Pour une DFT à N points puissance de 2 (2, 4, 8, etc.), il existe au moins un facteur commun entre M et L , avec $N = ML$. Selon les valeurs de M et L , nous considérons plusieurs variantes appelées algorithmes radix- r (base- r). Dans chaque étage de radix- r FFT, nous trouvons un module appelé, opérateur papillon (en l'anglais *Butterfly*); en le notera BPE (*Butterfly Processing Element*), où r désigne le nombre de points d'entrée du BPE, pour réaliser une FFT complète à N points. Le BPE est le cœur de calcul de la FFT, il permet de réaliser le calcul du papillon de radix- r de la FFT.

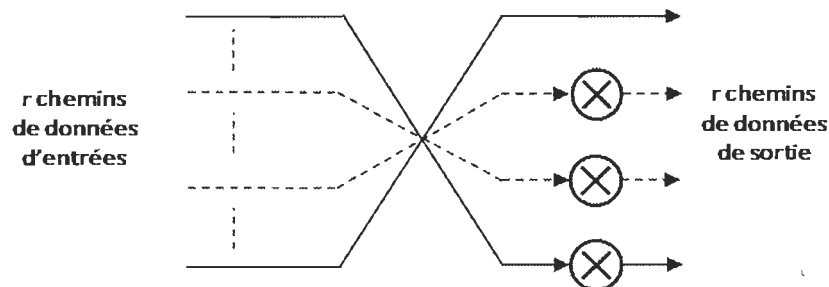


Figure 2.10: Papillon (BPE) radix- r

La Figure 2.10 illustre le schéma de base d'un BPE radix- r . Il faut noter que le BPE radix- r utilise les opérations de base comme la multiplication complexe et l'addition ou soustraction complexe pour calculer la FFT.

II.3.3. Architecture pipeline radix- r FFT

L'architecture en pipeline radix- r FFT (figure 2.11) est caractérisée par un traitement continu des entrées séquentielles de la FFT. Une telle architecture est composée de $\log_r N$ modules de calcul (MC), un par étage, qui correspond aux opérateurs papillon. La valeur de r correspond au radix de l'algorithme utilisé. Chaque MC traite N/r opérations papillon successives. Par conséquent, la taille maximale réalisable pour ce type d'architecture est dictée par le nombre de MC. En effet, ce type d'architecture offre un bon compromis complexité matérielle/taux de traitement de données pour les systèmes de communication à haut débit [JUN03].

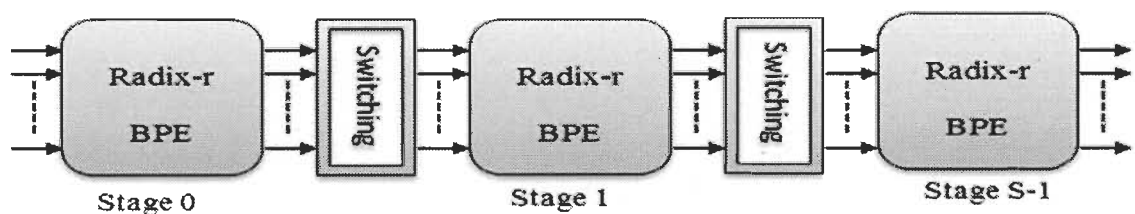


Figure 2.1: Architecture pipeline de S stages radix- r FFT [JAB09]

Selon la figure 2.11, une FFT de longueur $N = r^S$ est exécuté d'une façon générale par S étages de calcul, constituées chacune de N/r ensembles de calcul élémentaires. Chacun de ces ensembles est effectué par un papillon radix- r . Cela mène à $\frac{N}{r} \log_r N$ ensembles de calculs élémentaires, dont les expressions différentes selon la version choisie de l'algorithme et sa base. Les blocs de commutation (en anglais : Swtching) correspond aux bus de données à partir de $(S-1)^{i\text{ème}}$ à $(S)^{i\text{ème}}$, où $S = \log_r N$, pour $s^{i\text{ème}}$ itérations ($s = 0, 1, \dots, S-1$). Lorsque r chemins de données sont utilisés, le BPE pipeline atteint un taux de donné S fois la fréquence d'horloge inter-module. De plus, selon la figure 2.13 une FFT radix- r pipeline de taille N produira les premières r sorties en S cycles. Donc, le nombre de cycles nécessaires pour compléter une FFT de taille N est par conséquent $S + \left(\frac{n}{r} - 1\right)$ cycles [JAB09].

Il existe plusieurs architectures en pipeline basé principalement selon deux points :

- Le premier point de différence est le nombre de chemins de données utilisés pour traiter les échantillons. Nous trouvons ainsi deux types d'architectures, à chemin unique (Single-path; S), ou à chemin multiple (Multi-path; M). Il est intéressant de noter qu'un plus grand nombre de chemins permet d'augmenter le débit de traitement au détriment d'une augmentation des ressources nécessaires.
- Le deuxième point de différence consiste en la stratégie de mémorisation pour créer les différents délais nécessaires à l'ordonnancement des échantillons.

Selon ces deux critères, nous pouvons diviser les architectures pipelines en trois types [SHO98] :

- Radix-r SDF: *Single-path Delay Feedback*.
- Radix-r SDC: *Single-path Delay Commutator*.
- Radix-r MDC: *Multi-path Delay Commutator*.

L'architecture radix-r MDC (Multi-path Delay Commutator) est l'une des approches les plus utilisées et les plus appréciées dans l'implémentation des systèmes de communication à haut débit binaire. Elle comprend un commutateur pour permuter les échantillons et des délais pour synchroniser les mêmes échantillons. Selon l'algorithme radix-r utilisé, nous aurons $r - 1$ délais avant et après chaque module BPE. De plus, cette architecture nécessite $r - 1$ multiplieur par module BPE. Plusieurs architectures radix-r MDC existent, nous trouvons par exemple le R2MDC et le R4MDC qui sont les architectures les plus populaires, et aussi les radix-supérieures comme le radix- 2^2 et radix- 2^3 . Chacun de ces algorithmes est étudié de façon à être optimisé suivant certaines conditions. De plus, l'approche MDC est l'architecture la plus adaptée aux systèmes MIMO-OFDM parce qu'elle possède un multiple chemin d'entrées et de sorties [FU09]. Cette propriété permet aux modules FFT utilisant cette architecture d'opérer à haute vitesse avec un haut débit. Par contre, cette architecture utilise plus de surface silicium et par conséquent un cout plus élevé en matériel. Le choix d'algorithme de haut degré pour la FFT est en lui-même un choix permettant de réduire la consommation. Plus le degré est haut, plus le nombre de multiplications complexes triviales augmente. Dans ce travail, nous avons choisi

l'algorithme radix-4MDC pour le calcul de la FFT dans le but d'optimiser ces performances tel que la surface d'implémentation et la consommation d'énergie qui sont particulièrement important pour les communications sans fil basées sur les systèmes MIMO-OFDM.

II.3.4. Algorithme FFT radix-4 MDC

Depuis l'algorithme initial [COO65], différentes architectures, plus ou moins rapides et plus ou moins coûteuses, ont été proposées pour effectuer le calcul et réduire davantage la complexité de la FFT. Dans la suite de ce travail, nous utiliserons l'algorithme de radix-4 détaillé dans [PRO96], où 4 factorise N .

II.3.4.1. Algorithme radix-4 DIT

Nous pouvons toujours utiliser l'algorithme radix-2 pour le calcul de la FFT. Toutefois, l'utilisation des algorithmes radix- r supérieurs permet de réduire le nombre d'étage pipeline et le nombre d'opérations. L'amélioration de la complexité de calcul nous a conduits à mettre en application le radix-4 FFT dont le nombre de points de données est une puissance de 4 c.-à-d. $N = 4^v$ où v est un entier positif.

Brièvement, la description de l'algorithme radix-4 FFT DIT est obtenue en utilisant l'approche divisée pour régner décrite précédemment. Nous considérons les données des constantes suivantes :

$$\left\{ \begin{array}{l} L = 4, \quad M = \frac{N}{4} \\ l, p = 0, 1, 2, 3, 4 \\ m, q = 0, 1, \dots, \frac{N}{4} - 1 \\ n = 4m + 1, \quad k = \frac{N}{4} \end{array} \right. \quad (2.25)$$

Nous remplaçons les données de la relation (2.25) dans l'équation (2.21) tout en séparant

une séquence de N points d'entrées en 4 sous-séquences $X(4n), X(4n+1), X(4n+2),$

$X(4n+3)$ où $n = 0, 1, \dots, \frac{N}{4} - 1$. En obtient donc :

$$X(p, q) = \sum_{i=0}^4 [W_N^{iq} F(l, q)] W_4^{lp} \quad p = 0, 1, 2, 3, 4 \quad (2.26)$$

Avec $F(l, q)$ est donné par l'équation (2.27) :

$$F(l, q) = \sum_{m=0}^{\frac{N}{4}} X(l, m) W_{N/4}^{mq} \quad \begin{array}{l} l = 0, 1, 2, 3, 4 \\ q = 0, 1, \dots, \frac{N}{4} - 1 \end{array} \quad (2.27)$$

Et :

$$\left\{ \begin{array}{l} x(l, m) = x(4m + 1) \\ X(p, q) = X\left(\frac{N}{4}p + q\right) \end{array} \right. \quad (2.28)$$

Ainsi, la DFT à quatre $N/4$ points $F(l, q)$ est obtenus à partir de l'équation (2.26) sont organisés pour donner les N points DFT. L'expression pour arranger les $N/4$ points DFT sont exprimés sous forme matricielle comme suite :

$$\begin{bmatrix} X(0,q) \\ X(1,q) \\ X(2,q) \\ X(3,q) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} W_N^0 F(0,q) \\ W_N^q F(1,q) \\ W_N^{2q} F(2,q) \\ W_N^{3q} F(3,q) \end{bmatrix} \quad (2.29)$$

L'algorithme FFT radix-4, est donc une décomposition de DFT en quatre sous transformés indépendants de taille N/4. La structure typique de BPE radix-4 suivant l'algorithme décrit dans l'équation 2.28 est représenté par la figure 2.12. Ainsi sa forme simplifier est illustrée dans la figure 2.13.

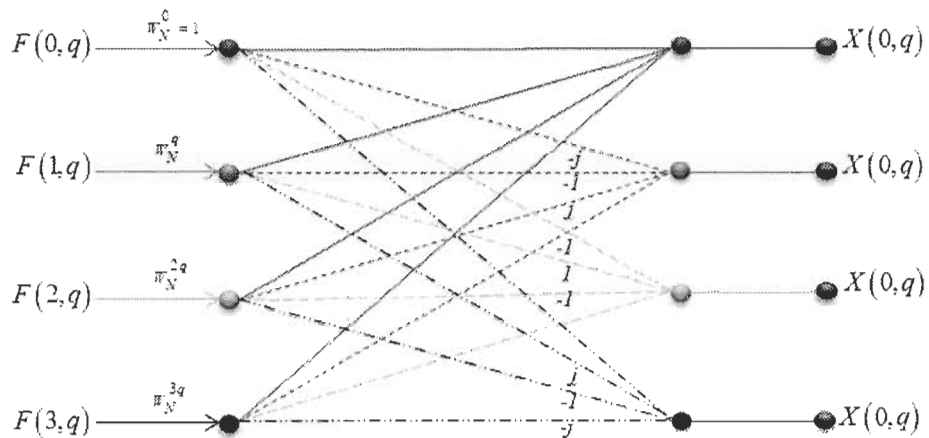


Figure 2.22: Diagramme typique de papillon radix-4 DIT

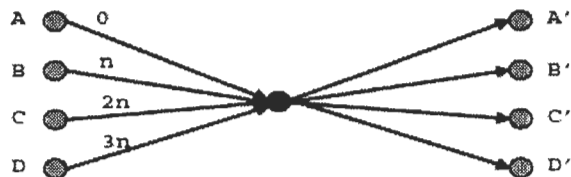


Figure 2.3: la forme simplifiée de papillon radix-4 DIT

Il est à noter que chaque BPE implique 3 multiplications complexes et 12 additions complexes. Nous excluons les opérations impliquant les opérateurs triviaux j et $-j$ pouvant être résolue par une simple manipulation de signes des partie réel et imaginaire. La figure 2.14 représente la structure de radix-4 DIT avec $N=16$, les entrées sont à bits inversées tandis que les sorties sont présentés dans un ordre normal.

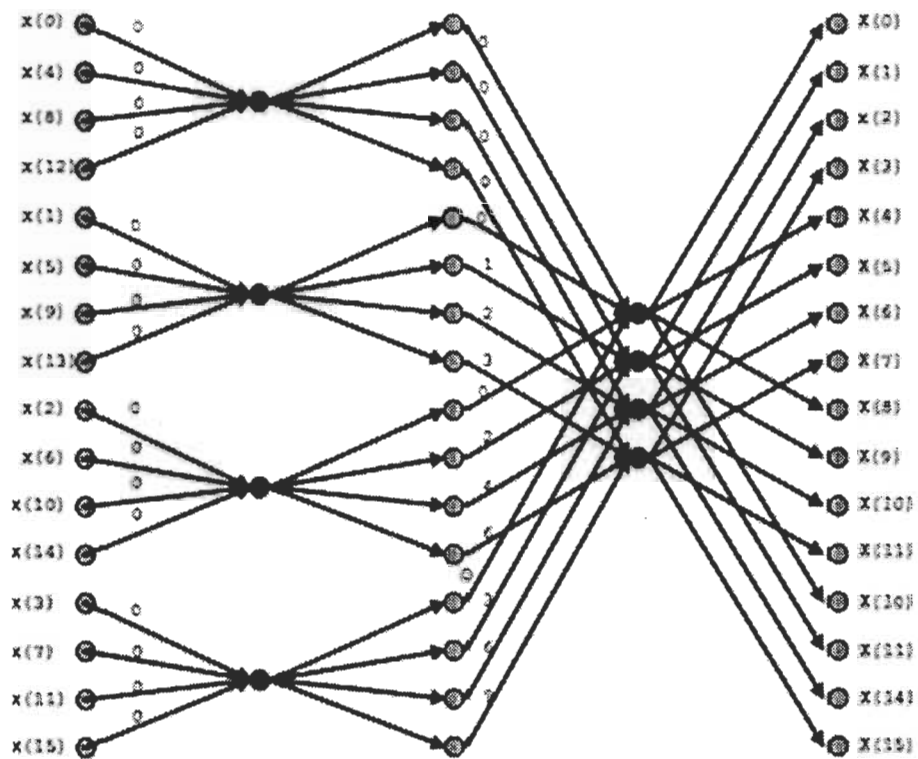


Figure 2.4: Diagramme de la FFT à base de Butterfly Radix4 de type DIT N=16

II.3.4.2. Algorithme radix-4 DIF

si nous voulons avoir la structure de radix-4 DIF, nous remplaçons dans l'équation (2.21) $L = N/4$ et $M = 4$. Nous dérivons l'algorithme radix-4 DIF en divisant les N données d'entrée en quatre petites séquences élémentaires, nous avons donc :

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{kn} \\
 &= \sum_{n=0}^{\frac{N-1}{4}} x(n) W_N^{kn} + \sum_{n=\frac{N}{4}}^{\frac{2N-1}{4}} x(n) W_N^{kn} + \sum_{n=\frac{N}{2}}^{\frac{3N-1}{2}} x(n) W_N^{kn} + \sum_{n=\frac{3N}{4}}^{N-1} x(n) W_N^{kn} \\
 &= \sum_{n=0}^{\frac{N-1}{4}} x(n) W_N^{kn} + W_N^{\frac{kn}{4}} \sum_{n=0}^{\frac{N-1}{4}} x\left(n + \frac{N}{4}\right) W_N^{kn} + W_N^{\frac{2kn}{4}} \sum_{n=0}^{\frac{N-1}{4}} x\left(n + \frac{N}{2}\right) W_N^{kn} \\
 &\quad + W_N^{\frac{3kn}{4}} \sum_{n=0}^{\frac{N-1}{4}} x\left(n + \frac{3N}{4}\right) W_N^{kn}
 \end{aligned}$$

(2.30)

où : $k = 0, 1, \dots, N-1$

D'après la définition des facteurs de phase nous pouvons démontrer que :

$$W_N^{\frac{kN}{4}} = (-j)^k, \quad W_N^{\frac{kN}{2}} = (-1)^k, \quad W_N^{\frac{3kN}{4}} = (j)^k \quad (2.31)$$

Avec cette substitution, l'équation (2.29) peut récrit selon

$$X(k) = \sum_{n=0}^{\frac{N-1}{4}} \left[x(n) + (-j)^k x\left(n + \frac{N}{4}\right) + (-1)^k x\left(n + \frac{N}{2}\right) + (j)^k x\left(n + \frac{3N}{4}\right) \right] W_N^{kn} \quad (2.32)$$

Si nous remarquons dans cette décomposition de l'équation (2.30), $X(k)$ n'est pas encore une FFT de longueur $N/4$, car W_N^{kn} dépend de N et non pas de $N/4$. Pour ce faire, la séquence $X(k)$ est décomposée en quatre sous-ensembles dans le cas où :

$$\begin{cases} k = 4r, 4r + 1, 4r + 2, 4r + 3 \\ r = 0, 1, \dots, \frac{N}{4} - 1 \end{cases} \quad (2.33)$$

Nous remplaçons dans l'équation (2.30) pour chaque valeur de k , On obtient donc :

$$\begin{cases} X(4r) = \sum_{n=0}^{\frac{N}{4}-1} \left[x(n) + x\left(n + \frac{N}{4}\right) + x\left(n + \frac{N}{2}\right) + x\left(n + \frac{3N}{4}\right) \right] W_N^0 W_N^{kn} \\ X(4r+1) = \sum_{n=0}^{\frac{N}{4}-1} \left[x(n) - jx\left(n + \frac{N}{4}\right) + x\left(n + \frac{N}{2}\right) + jx\left(n + \frac{3N}{4}\right) \right] W_N^n W_{N/4}^{kn} \\ X(4r+2) = \sum_{n=0}^{\frac{N}{4}-1} \left[x(n) - x\left(n + \frac{N}{4}\right) + x\left(n + \frac{N}{2}\right) - x\left(n + \frac{3N}{4}\right) \right] W_N^{2n} W_{N/4}^{kn} \\ X(4r+3) = \sum_{n=0}^{\frac{N}{4}-1} \left[x(n) + jx\left(n + \frac{N}{4}\right) - x\left(n + \frac{N}{2}\right) + jx\left(n + \frac{3N}{4}\right) \right] W_N^{3n} W_{N/4}^{kn} \end{cases} \quad (2.34)$$

Chaque équation de la relation (2.34) représente une séquence de $N/4$ points FFT qui peut être calculée en répétant la procédure précédente jusqu'à $N = 4$. La figure 2.15 représente les structures Radix-4 DIF. Les entrées sont normalement commandé tandis que les sorties sont présentées dans un ordre inverse.

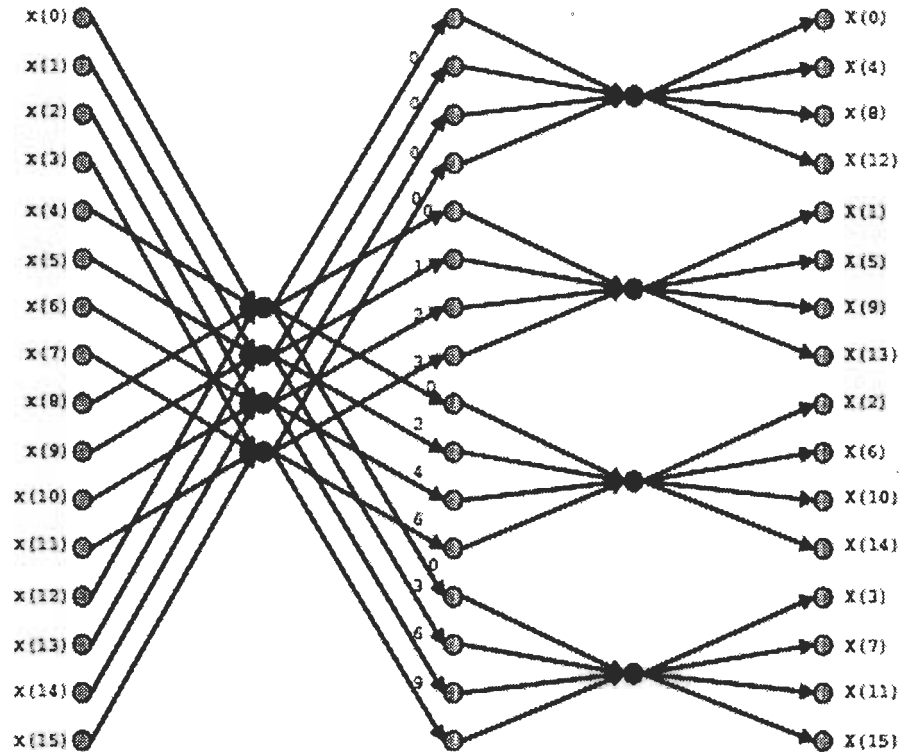


Figure 2.5: Diagramme de la FFT à base de Butterfly Radix4 de type DIF N=16

Nous constatons que le Radix-4 DIF possède les mêmes propriétés que le Radix-4 DIT sauf que la division se fait au niveau de la fréquence ou le temps. De plus, le Radix-4, DIT ou DIF, réduit l'ordre de calcul de N^2 à $3\log_4 N - 1$ multiplications complexes et de $N^2 - N$ à $8\log_4 N$ additions complexes.

II.3.4.2. Architecture pipeline radix-4MDC

La figure 2.16 représente l'architecture pipeline R4MDC à 4 canaux pour $N = 64$ points. Cette architecture utilise quatre chemins d'entrées en parallèles, et en trouve à la fin quatre chemins de sorties. Le flux de données d'entrée et les données inter-étage sont contrôlés

avec un extra commutateur. Ce dernier est implémenté par de simples éléments de délais et un multiplexeur. Donc, sa complexité matérielle lors de son implémentation est minimale. Le rôle de cet extra commutateur est de réorganiser les quatre flux d'entrées de données avant de les injecter dans l'architecture [JUN03].

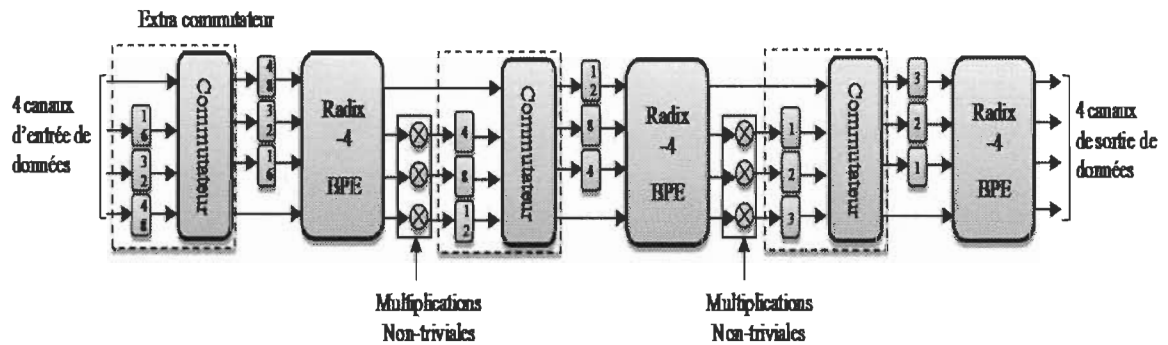


Figure 2.6: Architecture R4MDC ($N = 64$ points)

Comparée à plusieurs autres architectures pipeline, le radix-4 MDC requière une faible manipulation de mémoire, son compromis vitesse/surface rend le processeur radix-4 MDC préférable pour l'implémentation des systèmes MIMO-OFDM [SAN05].

II.4. EFFET DE QUANTIFICATION

Les unités de calcul en virgule flottante sont des unités trop complexes est présente un coût matériel trop important. A cet effet, le calcul est réalisé souvent avec des nombres représentés en virgule fixe afin de satisfaire les coûts en VLSI pour les implémentations FPGA. La figure 2.17 résume quelques méthodes de représentations en virgule fixe :

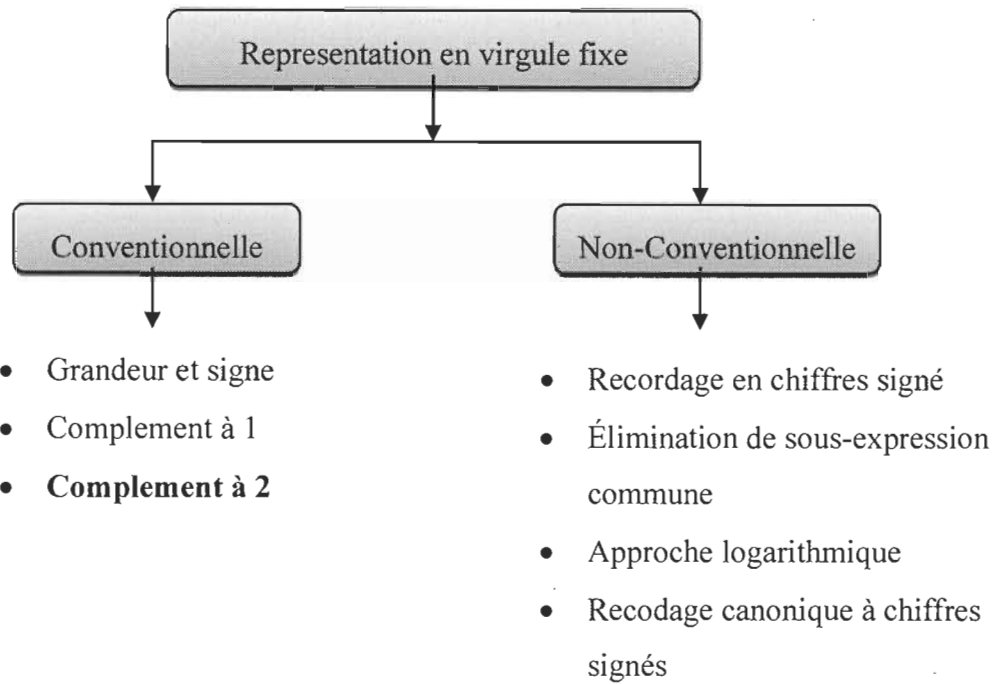


Figure 2.7: Quelques méthodes de représentations en virgule fixe

En raison des performances d'implémentations avantageuses en termes de consommation d'énergie, de surface et de latence de l'architecture dédiée ; l'arithmétique en virgule fixe est préférée pour les applications de DSP. De plus, toutes ces performances dépendent de la représentation en virgule fixe utilisée, ainsi, un choix judicieux de la largeur binaire dans la conception FPGA peut entraîner des économies substantielles. Dans cette section, nous discutons les différentes méthodes de l'arithmétique en virgule fixe, et évaluer la précision existante.

II.4.1. Arithmétique en virgule fixe conventionnelle

II.4.1.1. Représentation des nombres

La plupart des algorithmes de DSP manipulent des données définies sur un intervalle de nombres réels ou un domaine de nombres complexes, mais ces derniers sont équivalents à des couples de réels. Dans la représentation des nombres dans un format dite en virgule fixe, les données codées sont composées d'une partie fractionnaire et d'une partie entière.

La figure 2.18 représente une donnée composée de b bits ($b = m + l + 1$) pour représenter les nombres. La base choisie est 2, car il s'agit d'une représentation binaire de deux chiffres 0 et 1 [KUN91][BOI97]. Soit donc :

$$b_i \in \{0, 1\}, i \in \{m, m-1, \dots, 0, 1, \dots, -l\} \quad (2.35)$$

Chaque donnée se compose d'un bit de signe S et de $(b - 1)$ bits répartie en m bits pour la partie entière et l bits pour la partie fractionnaire.

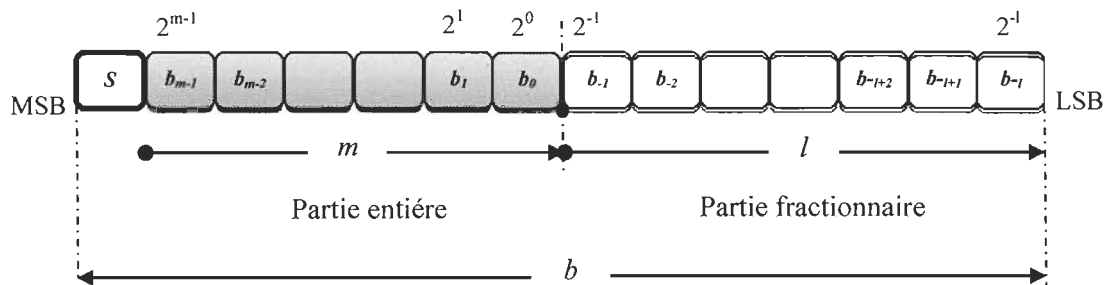


Figure 2.8: Représentation des données en virgule fixe

Le format d'une donnée en virgule fixe est entièrement défini par la représentation choisie et la largeur de sa partie entière et de sa partie fractionnaire. En ce qui concerne la représentation grandeur et signe d'un nombre x , le module est représenté par $(m - 1)$ bits, tandis que $S = 0$ pour un nombre positif et $S = 1$ pour un nombre négatif. Nous pouvons donc écrire [WAN99][[KUN91]][[BOI97]] :

$$x = (-1)^S \cdot \sum_{i=-1}^{m-1} b_i \cdot 2^i \quad (2.36)$$

Ce type de codage dans l'équation (2.36) possède deux représentations de la valeur zéro, ainsi le nombre de valeurs représentable est égale à $(2^b - 1)$. Le domaine de définition de cette représentation avec un pas de quantification noté par q est donné par l'équation (2.37)

$$-2^m + q \leq x \leq 2^m - q, \quad q = 2^1 \quad (2.37)$$

La représentation en complément à deux (2c) est détaillée ci-dessous. Cette représentation est très utilisée car elle possède des propriétés arithmétiques très intéressantes pour l'addition et la soustraction. D'une façon générale la représentation d'une donnée x en complément à 2 est illustrée dans l'équation (2.38) comme suite:

$$x = -2^m \cdot S + \sum_{i=-1}^{m-1} b_i \cdot 2^i \quad (2.38)$$

Par rapport au code grandeur et signe, ce type de code a l'avantage de ne posséder qu'une seule représentation de la valeur zéro. En conséquence, le domaine de définition D de ce

code n'est pas symétrique par rapport à l'origine, il est composé de $(2^{b-1} - 1)$ valeurs positives de 2^{b-1} valeurs négatives :

$$D = [-2^m, -2^m - 2^l] \quad (2.39)$$

Où $(q = 2^{-1})$, est le pas de quantification.

L'exécution d'un circuit intégré dédié apporte des limitations dans le calcul arithmétique du fait de la représentation des données en virgule fixe ce qui est nécessaire pour satisfaire les contraintes de coût et de consommation.

En général, l'inconvénient d'utiliser l'arithmétique à virgule fixe conduit à une dégradation de la précision des calculs réalisés. Pour satisfaire les performances associées à l'application, il est nécessaire que l'arithmétique en virgule fixe implantée garantisse une précision minimale.

II.4.1.2. Erreur de quantification

Lorsqu'une FFT est effectuée en arithmétique à virgule fixe, des erreurs de quantification interviennent, liées à la longueur certainement finie des valeurs numériques codées à l'intérieur de la machine sous la forme de mots binaires. Elles impliquent une incertitude sur le résultat qui est ainsi affecté des sources de bruits. La quantification d'un signal continu x est modélisée par la somme de ce signal et d'une variable d'erreur aléatoire e , comme illustré à la figure 2.19

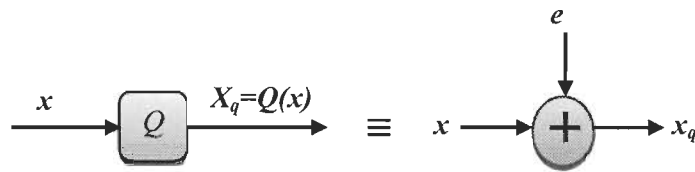


Figure 2.9: Modélisation du processus de quantification du signal x

Nous pouvons distinguer trois sources d'erreurs de quantification initiales [KUN91] :

- L'erreur de quantification dus au signal d'entrée liée à la résolution du convertisseur analogique-numérique.
- L'erreur de quantification dus aux coefficients de multiplication W de la FFT.
- L'erreur de quantification dans les unités de calculs; après multiplications et/ou les additions.

Ces trois types d'erreur se propagent de l'entrée vers la sortie au sein du système, la FFT dans notre cas, et modifient la précision des calculs en sortie de l'application. Dans de nombreux cas, il est raisonnable de traiter les effets de troncatures et d'arrondis par un modèle statistique remplaçant chaque source d'erreur par un bruit blanc. En effet la dégradation de la précision des calculs doit être maîtrisée afin de garantir l'intégrité de l'algorithme et les performances de l'application.

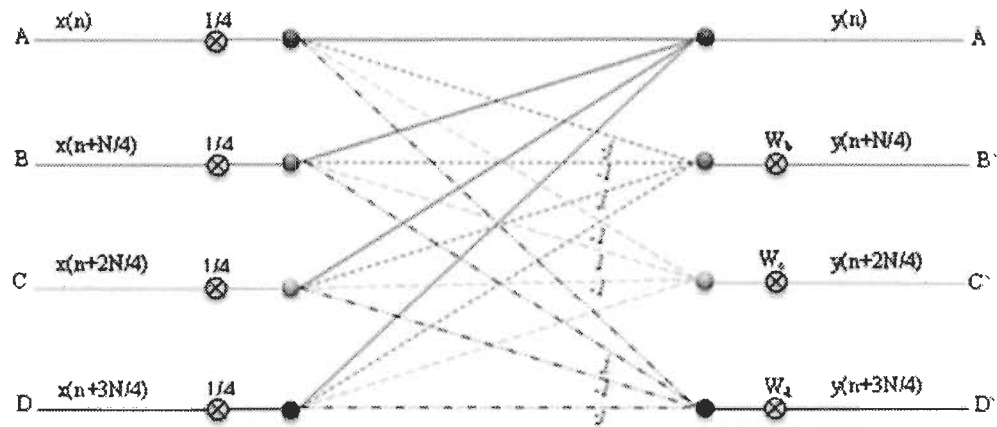


Figure 2.20: BPE radix-4 FFT avec mise à l'échelle de 1/4

Dans une FFT radix-4, le domaine de définition des données en virgule fixe est $[-1,1]$. Par conséquent, les entrées de papillons doivent être réduites pour éviter des résultats en dehors de la plage dynamique, cette instabilité est intitulée le débordement qui peut causer des erreurs importantes dans la sortie. La mise à l'échelle à chaque étage est suffisante pour garantir qu'aucun débordement ne se produira pour une entrée. À cet égard, comme la figure 2.20 illustre, ce débordement peut être évité en associant un facteur de $1/r$ à l'entrée de la FFT où r est le radix.

CHAPITRE III

ALGORITHME GÉNÉTIQUE POUR L'OPTIMISATION DE LA FFT

Les ingénieurs rencontrent fréquemment des problèmes de plus en plus complexes, qui surgissent dans des secteurs variés. Parmi ces problèmes, nous notons les problèmes d'optimisations difficiles. L'optimisation est le fait d'obtenir le meilleur résultat de compromis dans un ensemble de données. Le compromis s'exprime par une ou plusieurs fonctions objectives quantitatives, ou fonctions sélectives, que l'on cherche à maximiser. Dans cette section, nous comptons à utiliser une méthode d'optimisations, basée sur les algorithmes génétiques (AG). L'application des AG a été étudiée pour l'optimisation des largeurs des coefficients de la FFT qui assurent les meilleures performances.

III.1. ALGORITHME GÉNÉTIQUE POUR L'OPTIMISATION EN VIRGULE FIXE DES COEFFICIENTS DE LA FFT

La précision et la complexité de matériel des processeurs FFT pipelines en virgule fixe sont affectées par la largeur des coefficients de la FFT [HAN06]. Évidemment, la précision de calcul de la FFT est proportionnelle à la largeur de données utilisées dans ces calculs. Cependant, le coût en ressource d'implémentation et également la consommation en puissance sont aussi proportionnels à la largeur des données. Dans le but de trouver la largeur optimale qui assure les meilleures performances, nous proposons d'utiliser une approche d'optimisation de la performance basée sur les algorithmes génétiques (AG).

Ces dernières années, l'optimisation basée sur la largeur des données a reçu une attention considérable. Dans la littérature, nous trouverons les travaux de Sulaiman et Arslan et ces collègues qui ont utilisé les AG sur des modules FFT pipelines pour l'optimisation de ces paramètres. À cet effet, l'optimisation mono-objectif a été étudiée dans [SUL04]. Toutefois, la largeur des coefficients FFT a été optimisée par l'AG d'une FFT radix-4 avec l'étude limitée à une FFT de 16 points. De plus, certains papiers ont utilisé l'optimisation multi-objectif pour les processeurs FFT pipeline. Dans [SUL05] [SUL06], les chercheurs ont employé les AG pour optimiser la largeur des coefficients d'une FFT 16 points afin de minimiser la consommation en puissance dans un récepteur MC-CDMA. Dans les travaux de [SUL05], les auteurs ont considéré la méthode des sommes pondérées (*The weighted-sum method*) comme la fonction d'adaptation [GEN00]. Dans cette technique le rapport

signal sur bruit de quantification (SQNR)⁴ représenté par un coefficient α et la consommation en puissance représentée par le coefficient β . La technique de la roulette a été appliquée dans l'opérateur de la sélection. Ainsi la probabilité de croisement et de mutation est fixée respectivement à 90% et 10%, avec le critère de performance est le SQNR. D'ailleurs, les données d'entrées utilisées dans les travaux cités précédemment [SUL04, 05, 06] ne sont pas clairement définies, à savoir s'il s'agit d'une seule trame de données d'entrées de la FFT fixe (stationnaire) ou si l'étude considère le passage d'un ensemble de trame de données aléatoires. Dans ce chapitre, nous explorons uniquement les AG, leurs principes de fonctionnement, et leurs applications pour l'optimisation de la FFT.

III.2. ALGORITHME GÉNÉTIQUE ET STRATÉGIE D'ÉVOLUTION

III.2.1. Présentation des algorithmes génétiques

Les algorithmes génétiques (AG) reposent sur l'analogie entre la théorie de l'évolution naturelle de Darwin et l'optimisation. Selon la théorie de Darwin, les individus d'une population les mieux adaptés à leur environnement ont une plus grande probabilité de survivre et de se reproduire, en donnant des descendants encore mieux adaptés. Les AG sont des stratégies d'optimisation qui font partie des algorithmes évolutionnaires. Ils ont été proposés par J. Holland [HOL75], puis développés par d'autres chercheurs tels que De Jong

⁴ SQNR : Signal-to-Quantization-Noise Ratio

[DEJ75] et Goldberg [GOL89]. Leurs principes théoriques a été exposés et popularisés par Goldberg et Holland [GOL88] [GOL89].

D'après Goldberg et Holland [GOL88] [GOL89], les AG ont des particularités spéciales résumées dans trois points principaux:

1. Utilisation d'un codage pour les paramètres sous forme de gène.
2. L'opérateur de sélection des individus les mieux adaptés ou les plus compétitifs
3. Les opérateurs de reproduction tels que la recombinaison et la mutation qui agissent sur les gènes.

Présentés comme un outil d'optimisation puissant, et grâce à leurs simplicités d'implantation, la diversité de leur mise en application, et leur efficacité même pour des problèmes complexes, les AG ont été utilisés pour résoudre les problèmes d'optimisation et ont été conduits à un nombre croissants de travaux. Les AG ont exploré un grand nombre d'applications comme, l'intelligence artificielle, recherche opérationnelle, optimisation combinatoire, l'optimisation de fonctions numériques difficiles (discontinues, multimodales, bruitées...) [BEA93] [DOR95]. Les AG sont également utilisés en traitement d'image (alignement des photos satellites, la reconnaissance de suspects...) [BEA93], dans le contrôle de systèmes industriels [BEA93], la cryptographie et l'apprentissage des réseaux de neurones [REN95]... etc. Ils ont aussi été employés pour optimiser des réseaux (câbles, fibres optiques...), des circuits VLSI [BEA93], des antennes [REI97]. D'une manière générale, les AG sont des algorithmes s'appuyant sur des techniques dérivées de la génétique et des mécanismes d'évolution de la nature. Voici

quelques définitions et terminologies distinctes relatives à la génétique résumée dans la figure 3.1 [SZW11].

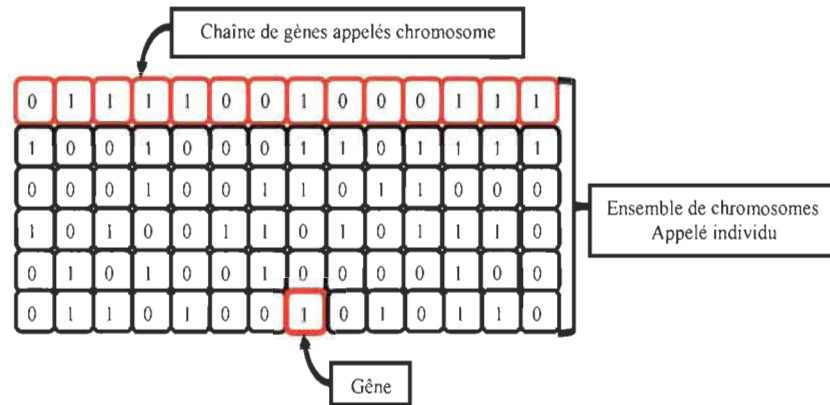


Figure 3.1: Représentation binaire d'un individu de dimensions 6.

Comme la figure 3.1 illustre, chaque individu est caractérisé par un génotype ou bien chromosome représentant le codage d'une solution au problème auquel est appliqué un algorithme génétique. Toutefois, un chromosome est une unité organisatrice qui code la structure des êtres vivants. Il correspond au codage sous forme de gènes d'une solution potentielle à un problème d'optimisation. Cependant, le gène est la brique de base du chromosome. C'est un point de chromosome responsable d'un caractère et peut prendre n'importe quelle valeur de l'alphabet du codage. Dans le codage binaire, un gène vaut soit, 0 soit 1.

III.2.2. Principe de base des AG

L'organigramme de la figure 3.2 montre brièvement le principe de fonctionnement de l'AG et indique clairement ses différents opérateurs de base.

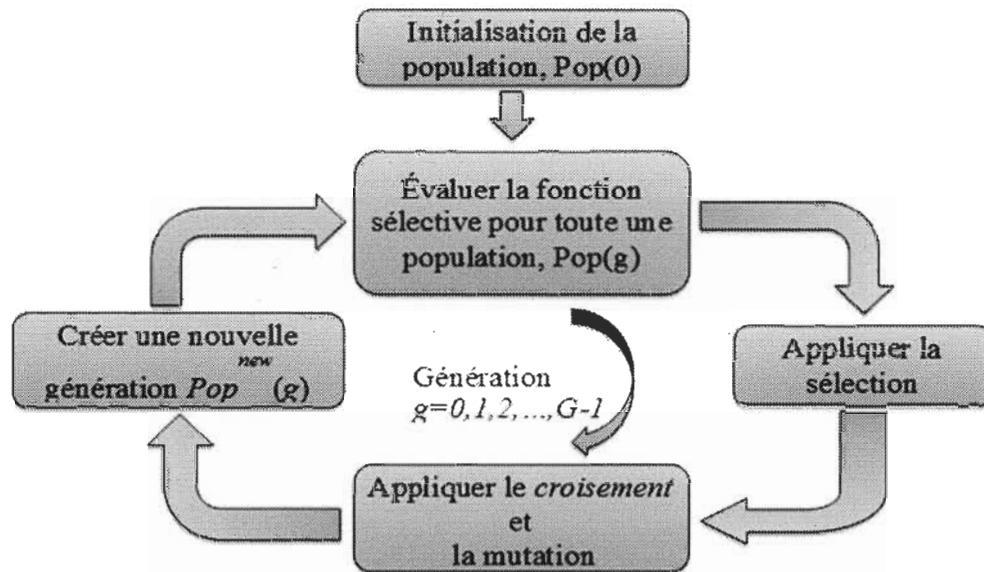


Figure 3.2: Principes de fonctionnement d'un algorithme génétique

Dans un premier temps, un AG nécessite le codage des paramètres du problème d'optimisation en une chaîne de longueur finie. Selon la figure 3.2, le principe de fonctionnement d'un AG est simple, il s'agit de manipuler une population P de m individus de taille constante généralement triée aléatoirement au départ. La population évolue sur plusieurs générations. À chaque génération g , les individus de la population $P(g)$ sont soumis à une compétition entre eux. Chaque individu est donné sous forme d'une seule chaîne de caractères appelés chromosome et qui représentent un point de l'espace de recherche [GOL89]. L'AG fait évaluer cette population d'individus au cours des générations à l'aide d'opérateurs de sélection, de croisement et de mutation qui s'inspirent des phénomènes naturels, ils ont tout en commun de produire une population $Pop^{new}(g)$. Le croisement et la mutation sont chargés d'explorer l'espace de recherche en construisant de

nouveaux individus à partir de la génération précédente, alors que la sélection favorise les individus qui possèdent une adaptation élevée. Les paramètres et les opérateurs de l'AG sont discutés en détail dans la section ci-après.

III.3. ALGORITHME GÉNÉTIQUE POUR L'OPTIMISATION DES COEFFICIENTS DE LA FFT

III.3.1. Génération de la population initiale

Selon l'algorithme décrit dans la figure 3.2, nous commençons par la génération de m éléments de population initiale, $Pop(0)$, ou m , est un nombre préalablement fixé qui représente le nombre d'individus W (coefficients FFT) générés aléatoirement. Chaque chromosome se compose d'une partie réelle binaire et une partie imaginaire binaire. Cependant, nous distinguons deux types d'initialisation ; initialisation des données d'entrées et l'initialisation des coefficients de Fourier W . Dans un premier temps, nous créons aléatoirement une séquence de N données d'entrées à valeurs complexes. Ces données qui sont en virgule flottante sont converties en nombres binaires à l'aide de la fonction « f_i^5 » en Matlab®. La figure 3.3 illustre le code de la fonction f_i utilisé dans les simulations.

⁵ f_i : C'est une fonction en Matlab, son rôle est la conversion de virgule flottante en virgule fixe

```

Fw = fimath('OverflowMode','Saturate',...
    'RoundMode','nearest',...
    'CastBeforeSum', 0,...
    'ProductBias', 0,...
    'SumBias', 0,...
    'MaxProductWordlength',wl_w,...
    'ProductMode','SpecifyPrecision',...
    'ProductWordLength',wl_w,
    'ProductFractionLength',wlf_w,...
    'MaxSumWordLength', wl_w,...
    'SumMode','SpecifyPrecision',...
    'SumWordLength',wl_w,...
    'SumFractionLength',wlf_w);
    
```

```

W_FixP=fi(W,1,wl_w,wlf_w,'fimath',Fw);
    
```

Figure 3.3 Déclaration des définitions à virgule fixe dans Matlab

Ensuite, une population initiale est formée aléatoirement par une chaîne de valeurs binaires représentant les coefficients de Fourier W . Chaque coefficient est codé en complément à 2 avec une largeur binaire wl (figure 3.4).

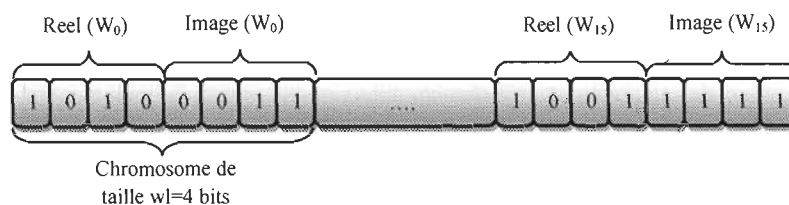


Figure 3.4: Encodages des chromosomes d'une FFT de 16 points

Réellement, si nous comptons a calculé une FFT de 16 points, le nombre de coefficients utilisés est différents de la longueur de la FFT. Selon les figures 2.14, 2.15, le calcul de 16 points FFT utilise seulement sept coefficients ($W_0, W_1, W_2, W_3, W_4, W_6, W_9$). À cet égard, notre tâche de recherche se limite à optimiser juste sept coefficients en écartant les autres coefficients ($W_{10}, W_{11}, W_{12}, W_{13}, W_{14}, W_{15}$). Cependant, le temps de calcul va diminuer et la complexité de l'algorithme aussi.

III.3.2. La phase d'évaluation

Le principe des AG est la recherche de l'individu le plus apte. Ce dernier est évalué par une fonction performance appellait fonction sélective (en anglais : *Fitness Function*). L'individu le plus performant sera celui pour lequel la fonction sélective est maximale. Dans un premier temps, notre but étant l'optimisation des coefficients de la FFT afin de maximiser la précision de la FFT défini par le SQNR en tant que fonction sélective proposée notée $f(x, x_q)$ selon l'équation (3.1):

$$f(x, x_q) = \text{SQNR}_{\text{dB}} = 10 \log_{10} \left(\frac{\text{Re}(x_\infty)^2 + \text{Im}(x_\infty)^2}{\left[\text{Re}(x_\infty) - \text{Re}(x_q) \right]^2 + \left[\text{Im}(x_\infty) - \text{Im}(x_q) \right]^2} \right) \quad (3.1)$$

où :

- x_∞ et x_q sont les signaux de sortie en virgule flottante et virgule fixe respectivement.

- $\text{Re}(x_q)$ et $\text{Im}(x_q)$ sont la partie réelle et imaginaire en virgule fixe de signal x_q .
- $\text{Re}(x_\infty)$ et $\text{Im}(x_\infty)$ sont la partie réelle et imaginaire en virgule flottante de signal x_∞ .

À chaque début de génération, tous les individus sont évalués en utilisant l'équation (3.1).

La phase d'évaluation mesure le degré d'adaptation des individus, et permet aux techniques de sélection de comparer les individus et de choisir les plus pertinents à partir des grandeurs à optimiser.

III.3.3. La reproduction

Afin, de passer d'une génération à une autre, la phase de la reproduction joue un rôle fondamental. Elle représente un cycle de l'AG, et elle se fait en appliquant des opérations génétiques telles que la sélection, le croisement et la mutation. Ces derniers sont présentés en détail dans les sections qui suivent.

III.3.4. Sélection

La sélection joue un rôle capital dans le bon fonctionnement des AG. Son objectif est de retenir et favoriser les meilleurs individus, alors que les individus les moins adaptés sont exclus progressivement de la population précédente selon le critère à optimiser. Toutefois, avant de procéder à la sélection, dans un premier temps nous commençons par le classement des individus qui ont un SQNR maximal pour chaque génération. Ceci permet de donner aux individus dont la valeur est plus grande, une probabilité plus élevée de participer à l'élaboration de la population suivante, c'est à dire de participer à l'opération de recombinaison.

La sélection des individus s'effectue sur la base de leur fonction d'évaluation. Plusieurs opérateurs de sélections existent et peuvent être mis en œuvre sous forme algorithmique de différentes façons. Dans la littérature, les techniques de sélection les plus connus et les plus utilisés sont [GEN00]:

- La sélection par la roulette russe.
- La sélection par rang.
- La sélection par tournoi.
- La sélection par troncature.
- La sélection par échantillonnage stochastique universel. Cette dernière nous l'avons choisie pour notre approche d'optimisation.

III.3.4.1. Sélection par échantillonnage stochastique universel

Dans un processus de recherche stochastique tel que les AG, la variété des chromosomes dans la population, la pression sélective, et la diversité génétique sont des facteurs importants. Puisque ces facteurs sont dépendants, l'augmentation de la pression sélective diminue la diversité dans la population, et vice versa. De plus, une forte pression sélective entraîne une convergence prématurée de l'algorithme, par contre une faible pression sélective rend la recherche inefficace. De ce fait, il est important de trouver un compromis entre ces deux facteurs, ce qui est le rôle de la méthode proposé dans le paragraphe qui suit.

La sélection par échantillonnage stochastique universel (*Stochastic Universal Sampling* : *SUS*), a été le fruit des travaux de J.Baker [BAK87]. Nous l'avons appliquée dans ce

travail, car elle permet d'établir un compromis entre la diversité génétique et la pression sélective. La sélection SUS donne plus de chance aux moins bons individus de participer à la plage de reproduction. Ainsi, les meilleurs ont toujours plus de chance d'être choisis. Dans cette méthode chaque individu de la population est associé à un secteur rectangulaire d'une longueur égale à sa fonction sélective (voir figure 3.5). L'ensemble des rectangles est placé de manière contiguë pour former une ligne de longueur égale à la somme des fitness. Le secteur total est marqué par des pointeurs désignés par des flèches qui sont régulièrement espacés.

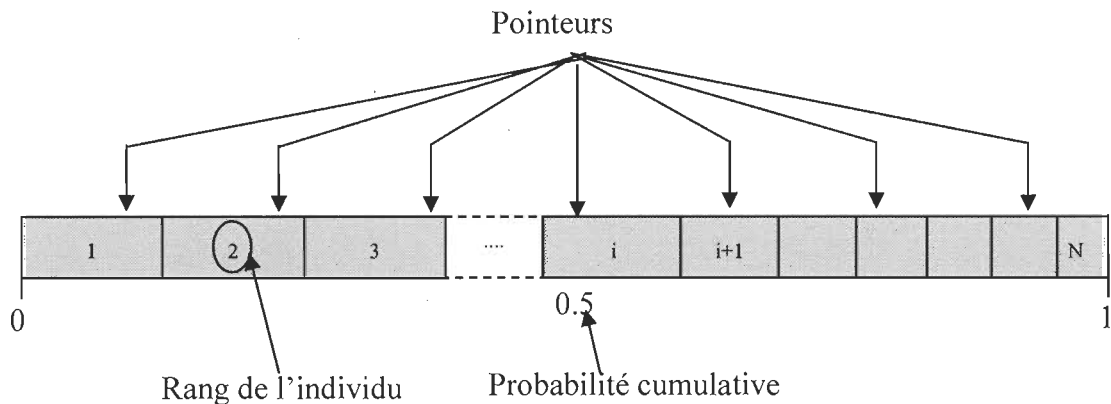


Figure 3.5: Principes de la technique de sélection SUS

Le nombre de pointeurs doit être égal au nombre d'individus à sélectionner m . Ces pointeurs ont la même distance entre eux, égale ainsi, à l'inverse du nombre d'individus à sélectionner $1/m$. L'emplacement du premier pointeur est défini aléatoirement par un nombre compris entre 0 et $1/m$. Le principe de la méthode SUS consiste à introduire dans la nouvelle population tous les individus indiqués par des pointeurs (les flèches). Si nous

considérons une population de taille m , les individus sont classés par rang de 1 à m . le premier rang est attribué a l'individu le plus pertinent et le rang m a l'individu le moins adapter. La probabilité de sélection d'un individu est proportionnelle à une performance artificielle calculée en fonction du rang de cet individu. Plusieurs variantes linéaire ou non linéaire ont été utilisées pour le calcul de la performance artificielle en fonction du rang.

III.3.4.2. Variante linéaire

Nous considérons m , le nombre des individus dans la population, r_i la position d'un individu dans cette population (l'individu le moins adapté a $r_i=1$, l'individu le plus apte ($r_i=m$), et SP est la pression sélective. En effet, la sélection des individus selon le cas linéaire est donnée par la relation suivante :

$$\text{Fitness}_{\text{Rank}}(x_i) = 2 - \text{SP} + 2 \cdot (\text{SP} - 1) \cdot \frac{(r_i - 1)}{(m - 1)} \quad (3.2)$$

Noté que : $\text{SP} \in [1, 2]$ dans le cas linéaire.

III.3.4.2. Variante non linéaire

L'utilisation de la variante non linéaire permet des pressions sélectives plus élevées que la méthode de classement linéaire. La distribution non linéaire utilisée dans notre travail est donné par la formule suivante :

$$\text{Fitness}_{\text{Rank}}(x_i) = \text{SP} \cdot \exp\left(-2 \cdot (\text{SP} - 1) \cdot \frac{r_i}{(m-1)}\right) \quad (3.3)$$

Noté que : $\text{SP} \in [1, m-2]$ dans le cas non linéaire.

Une fois la distribution non linéaire calculée, nous déduisons la probabilité de sélection selon l'équation (3.4) comme suite :

$$P(j) = \frac{\text{Fitness}_{\text{Rank}}(j)}{\sum_{j=1}^m \text{Fitness}_{\text{Rank}(j)}} \quad (3.4)$$

Ainsi, la probabilité cumulative pour chacun des chromosomes est décrit selon l'équation (3.5)

$$P_{\text{cum}}(j) = \sum_{j=1}^m P(j) \quad (3.5)$$

III.3.5. Croisements (Recombinaison)

Le croisement est un opérateur primordial dans les applications des AG. Il permet de créer de nouveaux individus par l'échange d'information entre deux autres chromosomes appelés parents. Classiquement, le croisement s'effectue en deux étapes. Dans la première étape, deux individus sont choisis aléatoirement selon une probabilité de croisement, noté P_c . Dans la deuxième étape, en localisant aléatoirement aussi des points de coupure sur les deux chromosomes Parent, notés Pr_1 et Pr_2 . L'opération de croisement échange le contenu d'information de ces points entre les deux chromosomes. Cette opération produit deux

individus enfants modifiés génétiquement qui font partie de la génération ultérieure. La figure 3.6 résume l'opération de croisement.

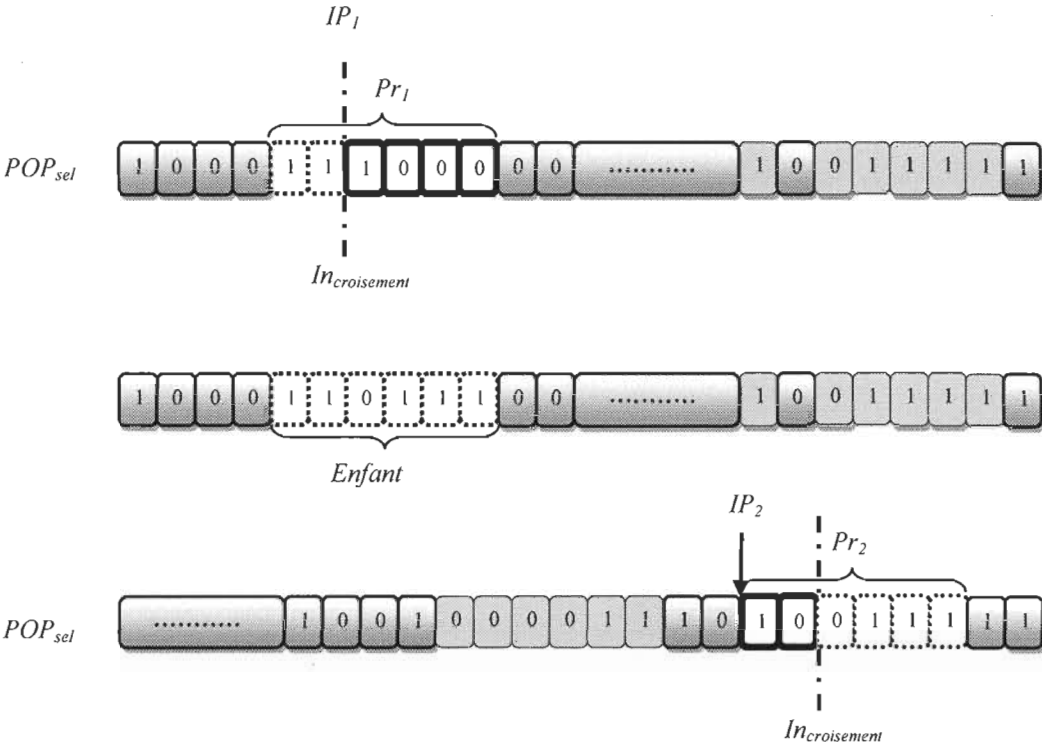


Figure 3.6: Opération de croisement

Afin de respecter ce principe, et après avoir sélectionné les individus les plus adaptés, nous aurions à la fin une nouvelle population POP_{sel} , patient prêt à l'opération de croisement. Cela peut être réalisé en gérant la probabilité de croiser les individus, noté P_c , la probabilité de croisement. En effet, le nombre de chromosomes à croiser dépendra de cette probabilité et de la taille de la population. Dans un premier temps, nous conservons les chromosomes les plus performants comme parents idéals Pr_{ideals} . Dans un deuxième temps, nous

choisirons aléatoirement deux indices de positions IP_1 et IP_2 pour les deux parents dans la population sélectionnée au préalable, soit Pr_1 et Pr_2 le parent1 et le parent2 successivement.

Nous pourrions traduire cette description sous forme d'équation donnée comme suite :

$$\begin{aligned} Pr_1 &= POP_{sel}(IP_1) \\ Pr_2 &= POP_{sel}(IP_2) \end{aligned} \quad (3.6)$$

Une fois, le choix fait, nous choisissons aléatoirement aussi une position de coupure $In_{croisement}$ dans chacun des parents. Nous permutons ensuite les deux sous chaîne terminale de chacun des deux parents, ce qui produit des enfants. Finalement, la nouvelle population après croisement se compose des individus enfants obtenus et les parents Pr_{ideals} conserver auparavant. L'opérateur de croisement doit participer à la convergence de l'algorithme en produisant une population plus performante.

III.3.6. Mutation

C'est une étape très importante dans le fonctionnement des AG. La mutation a pour but d'éviter les erreurs de recopie. Elle est aussi un processus aléatoire consiste à altérer le codage d'un chromosome. Elle se traduit à des modifications à une portion des individus en modifiant aléatoirement la valeur d'un gène ou plusieurs dans un Chromosome et en respectant une probabilité de mutation, noté P_m . Dans notre cas, nous choisirons aléatoirement un pointeur qui indique le placement des gènes à muté $In_{mutation}$ dans la population précalculée après le croisement. Chaque bit à muter est remplacé par son

complémentaire. Cet opérateur est donc d'une grande importance et il est loin d'être marginal. La figure 3.7 illustre une opération de mutation.

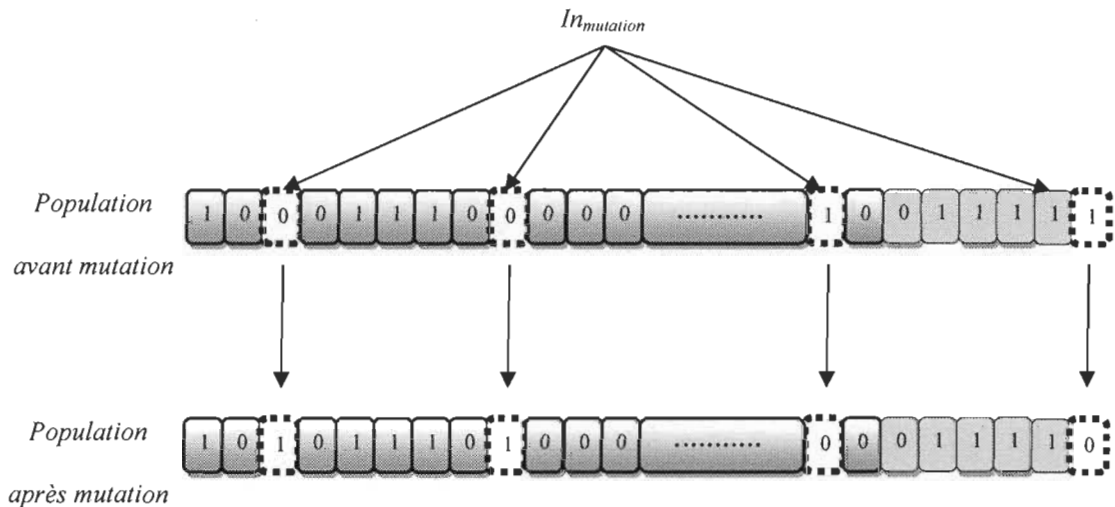


Figure 3.7: Opération de mutation

III.3.7. Critère d'arrêt

Le processus génétique recommencera une fois la nouvelle génération est créée. Les individus de la population sont évalués selon la fonction de forme à optimiser, de nouveaux parents sont sélectionnés et ainsi de suite. Les étapes de cycle génétique seront répétées pendant le nombre maximum de générations G fixé au préalable, ou selon un critère d'arrêt.

III.3.8 Réglage des paramètres des AG

L'AG fait intervenir de différents paramètres fixés à l'avance qui influent beaucoup sur sa convergence. Il s'agit de :

1. La taille de la population : elle doit être choisie de façon à réaliser un bon compromis entre le temps de calcul et la qualité de résultat.
2. Le ratio de croisement ou la probabilité de croisement : il détermine la portion des chromosomes qui sont croisés parmi ceux qui remplaceront l'ancienne génération.
3. Le ratio de mutation ou la probabilité de mutation : il détermine le ratio des chromosomes qui ne seront pas affectés par la mutation.

III.4. RÉSULTATS DES COEFFICIENTS DE FOURIER OPTIMISÉS EN VIRGULE FIXE

III.4.1. Métrique pour évaluer la qualité d'optimisation

De nos jours, la technologie microélectronique des circuits VLSI et le développement des processeurs DSP (Digital Signal Processing), ont connu une révolution remarquable. En contrepartie, la complexité et les erreurs de quantification restent des facteurs décisifs pour le développement des systèmes de télécommunications. D'ailleurs, la FFT pipeline est une opération de propagation, des erreurs de quantifications provenant des opérations arithmétiques. Les coefficients W sont utilisés avec une largeur binaire limitée, donc ils sont soumis à des pertes dues aux valeurs inexacts des coefficients W . Cependant, l'utilisation des longueurs de mot courtes assure un coût du matériel bas, mais une perte de précision [SUL05, 06]. D'autre part en aura une meilleure résolution en utilisant une plus grande largeur binaire. Mais dans cette situation, elle va augmenter la complexité du

matériel et aussi gaspille plus d'énergie et augmente son temps d'exécution. Ainsi, il est important de trouver la meilleure largeur des coefficients FFT tout en maximisant le SQNR. Ceci est l'objectif de notre expérience. Le critère de validation choisie est le SQNR calculé selon l'équation (3.1). Le bruit de quantification en sortie est engendré par la propagation des différents bruits présents au sein du système. Nous allons schématiser la méthode d'évaluation selon la figure 3.8.

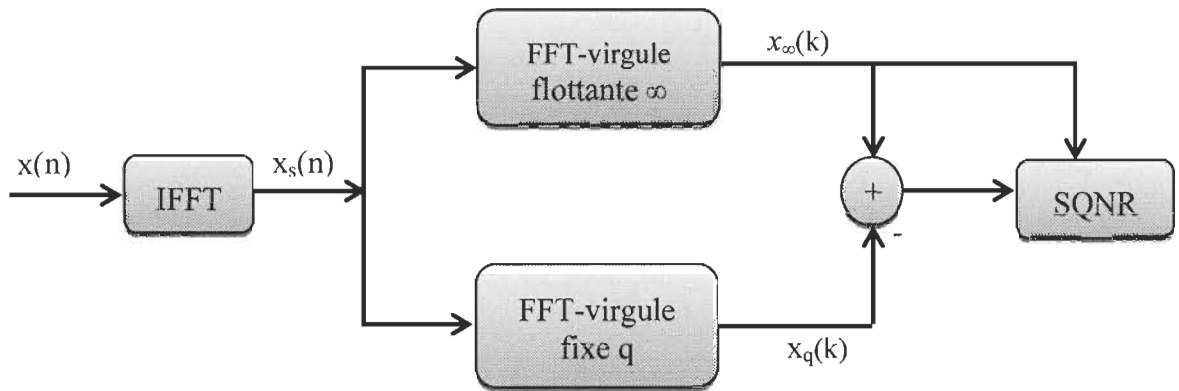


Figure 3.8: Méthodologies d'évaluation du SQNR

Dans la figure 3.7 ci-dessus, le signal $x(n)$ est passé par une IFFT et il traduit le principe d'un émetteur OFDM, le signal $x_s(n)$ est le signal reçu. Le signal $x_\infty(k)$ est en virgule flottante par contre le signal $x_q(k)$ est en virgule fixe. Le SQNR est le résultat obtenu selon l'équation (3.1).

Afin d'étudier l'effet de quantifications, nous avons soumis l'implémentation de l'algorithme génétique détaillée précédemment à une série de tests. Ces tests visent notamment à la mesure de l'efficacité de l'AG et de la qualité des solutions qu'il fournit en

comparaison avec la méthode de troncature (référence). Nous avons effectué des analyses de sensibilité relatives aux paramètres intrinsèques de l'algorithme génétique, puis mesuré l'évolution de la performance en fonction des variations de ces paramètres.

III.4.2. Initialisation des paramètres et conditions de simulation

Afin de maximiser le SQNR, il est indispensable d'initialiser tous les paramètres d'entrée.

Les conditions de simulation sont les suivantes :

- Nous travaillons sur une population de chromosomes $m=200$.
- Le nombre de génération appliqué est varié selon le nombre de points FFT utilisé $N=16, 64, 256$.
- La probabilité de mutation $P_m=30\%$.
- La probabilité de croisement $P_c=50\%$.
- Largeur binaire des coefficients $w_l=10$ bits.

Il est à noter que le choix des valeurs de P_c et P_m joue un rôle d'importance dans les opérations de croisement et de mutation afin d'améliorer les performances de l'algorithme génétique utilisé. Le nombre de générations utilisé varie avec le nombre de points FFT utilisés. Dans le texte qui suit, nous discuterons les résultats expérimentaux pour différents points FFT ($N=16, 64, 256$). Dans ces simulations, les données d'entrées utilisées sont en nombres complexes, aléatoires et non-stationnaires. De plus, nous avons conservé la largeur binaire des coefficients de Fourier à $w_l=10$ bits dans toutes les simulations. L'étude de

SQNR est vérifiée dans deux cas, la largeur de signal d'entrée $wl_x=10$ bits et $wl_x=16$ bits.

III.4.3. Optimisation des coefficients FFT

Nous commençons dans un premier temps à évaluer la fonction sélective donnée par l'équation (3.1) utilisée pour différent cas de simulation. L'échelle de mesure de cette figure représente le résultat de la fonction sélective normalisée selon l'équation suivante

$$f_n(x, xq) = \frac{\text{SQNR}_{\text{dB}} \text{ par AG}}{\text{SQNR}_{\text{dB}} \text{ par troncature}} \quad (3.7)$$

Le but de cette expérience était de trouver la largeur binaire optimisée pour les coefficients de la FFT. Les résultats obtenus sont représentés à la figure 3.9

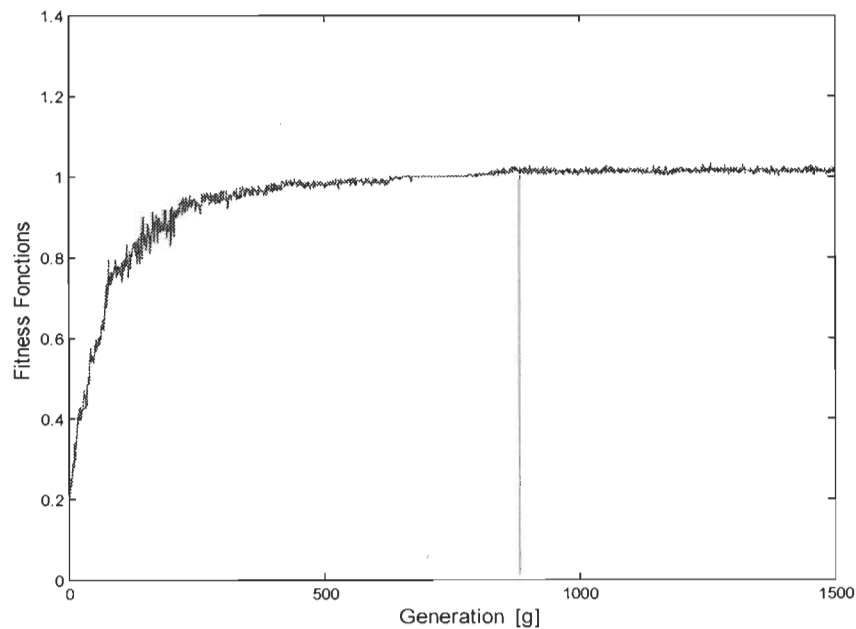


Figure 3.9: Évolution de fonction objective dans le cas où $N=16$, $wl_x=10$, $wl_W=1$

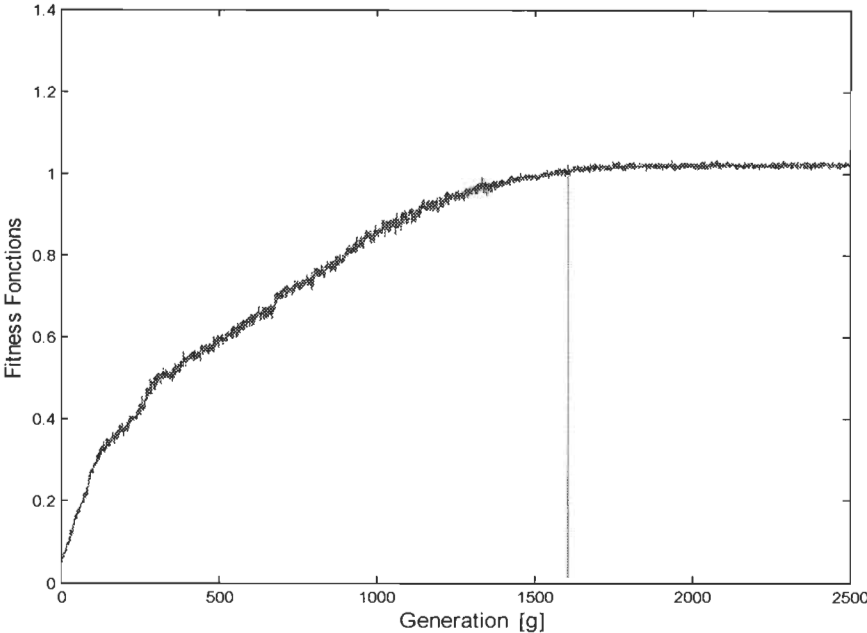


Figure 3.10: Évolution de fonction objective dans le cas où $N=64$, $wl_x=10$, $wl_W=10$

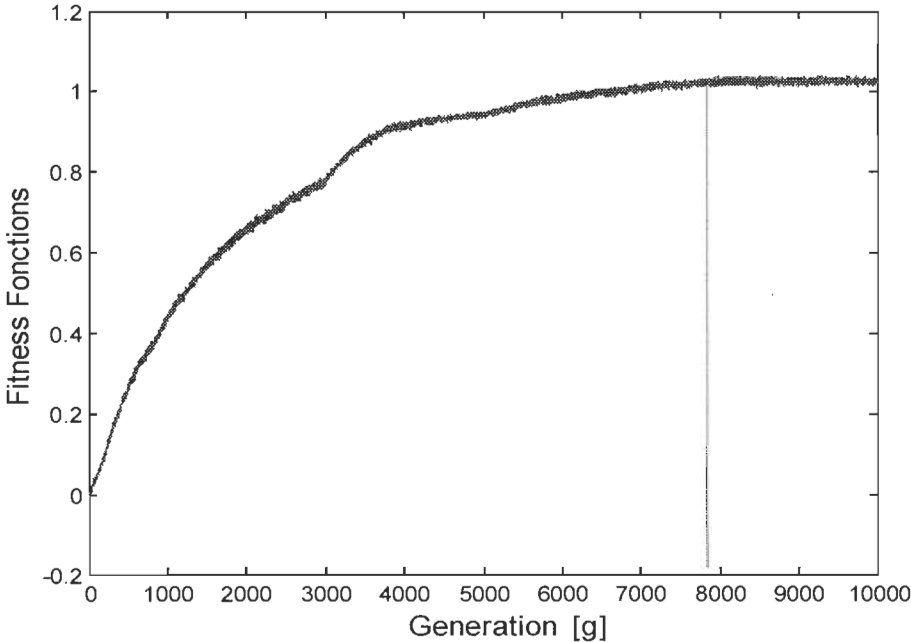


Figure 3.4: Évolution de fonction objective dans le cas où $N=256$ $wl_x=10$, $wl_W=10$

CHAPITRE III- ALGORITHME GÉNÉTIQUE POUR L'OPTIMISATION DE LA FFT

Les figures 3.9, 3.10, 3.11 illustrent l'évolution des performances de la fonction d'adaptation (SQNR) au cours des générations. Il est bien de remarquer que l'algorithme converge après un certain nombre de générations, mais cette convergence varie selon la longueur de la FFT utilisée. Dans cette recherche, les premiers meilleurs coefficients ont été trouvés dans la génération 900 pour une FFT de 16 points, et la génération 2000 pour une FFT de 64 points. Nous constaterons que le nombre de générations augmente lorsque la longueur de FFT augmente aussi.

Les tableaux 3.1 et 3.2 résument les coefficients génétiques obtenus à une FFT de longueur $N=16$ comparés aux coefficients tronqués, représentés en virgule fixe et en virgule flottante respectivement.

Tableau 3.1: Comparaison en virgule fixe en nombre signés entre les coefficients tronqués et les coefficients génétiques pour une FFT à 16 points

TWF	WFixP_real	WGA_real	WFixP_imag	WGA_imag
W^0	0111111111	0111111111	0000000000	0000000000
W^1	0111011001	0111011010	1100111100	1100111100
W^2	0101101010	0101101011	1010010110	1010010101
W^3	0011000100	0011000100	1000100111	1000100110
W^4	0000000000	0000000000	1000000000	1000000000
W^6	1010010110	1010010101	1010010110	1010010101
W^9	1000100110	1000100110	0011000100	0011000100

Les résultats correspondants au tableau 3.1 en virgule flottante sont donnés par le tableau 3.2

CHAPITRE III- ALGORITHME GÉNÉTIQUE POUR L'OPTIMISATION DE LA FFT

Tableau 2.2: Comparaison en virgule flottante en nombre signés entre les coefficients tronqués et les coefficients génétiques pour une FFT à 16 points

TWF	WFixP_real	WGA_real	WFixP_imag	WGA_imag
W^0	0.9980	0.9980	0.0000	0.0000
W^1	0.9238	0.9258	-0.3828	-0.3828
W^2	0.7070	0.7090	-0.7070	-0.7090
W^3	0.3828	0.3828	-0.9238	-0.9258
W^4	0.0000	0.0000	-1.0000	-1.0000
W^6	-0.7070	-0.7090	-0.7070	-0.7090
W^9	-0.9238	-0.9258	0.3828	0.3828

Les tableaux 3.3 et 3.4 résument les coefficients génétiques obtenus à une FFT de longueur $N=64$ comparés aux coefficients tronqués, représentés en virgule fixe et en virgule flottante respectivement.

Tableau 3.3: Comparaison en virgule fixe en nombre signés entre les coefficients tronqués et les coefficients génétiques pour une FFT à 64points

TWF	WFixPreal	WGAreale	WFixPimag	WGAimag
W^0	0111111111	0111111111	0000000000	0000000000
W^1	0111111110	0111111101	1111001110	1111001110
W^2	0111110110	0111110110	1110011100	1110011100
W^3	0111101010	0111101010	1101101011	1101101011
W^4	0111011001	0111011010	1100111100	1100111100
W^5	0111000100	0110101010	1100001111	1100001111
W^6	0110101010	0110101010	1011100100	1011100100
W^7	0110001100	0110001100	1010111011	1010111011

CHAPITRE III- ALGORITHME GÉNÉTIQUE POUR L'OPTIMISATION DE LA FFT

W^8	0101101010	0101101011	1010010110	1010010101
W^9	0101000101	0101000101	1001110100	1001110100
W^{10}	0100011100	0100011100	1001010110	1001010110
W^{11}	0011110001	0011110001	1000111100	1000111100
W^{12}	0011000100	0011000100	1000100111	1000100110
W^{13}	0010010101	0010010100	1000010110	1000010110
W^{14}	0001100100	0001100100	1000001010	1000001010
W^{15}	0000110010	0000110010	1000000010	1011001001
W^{16}	0000000000	0000000000	1000000000	1000000000
W^{18}	1110011100	1110011100	1000001010	1000001010
W^{21}	1100001111	1100001111	1000111100	1000111101
W^{22}	1011100100	1011100100	1001010110	1001010110
W^{24}	1010010110	1010010101	1010010110	1010010101
W^{27}	1000111100	1000111101	1100001111	1100001111
W^{30}	1000001010	1000001010	1110011100	1110011100
W^{32}	1000000000	1000000000	0000000000	0000000000
W^{33}	1000001010	1000000011	0000110010	0000110010
W^{36}	1000100111	1000100110	0011000100	0011000100
W^{39}	1001110100	1001110100	0101000101	0101000101
W^{42}	1011100100	1011100100	0110101010	0110101010

Les résultats correspondants au tableau 3.3 en virgule flottante sont donnés par le tableau 3.4

Tableau 3.4 Comparaison en virgule flottante en nombre signés entre les coefficients tronqués et les coefficients génétiques pour une FFT à 64 points

TWF	WFixP_real	WGA_real	WFixP_imag	WGA_imag
W^0	0.9980	0.9980	0.0000	0.0000

CHAPITRE III- ALGORITHME GÉNÉTIQUE POUR L'OPTIMISATION DE LA FFT

W^1	0.9961	0.9941	- 0.0977	- 0.0977
W^2	0.9805	0.9805	- 0.1953	- 0.1953
W^3	0.9570	0.9570	- 0.2910	- 0.2910
W^4	0.9238	0.9258	- 0.3828	- 0.3828
W^5	0.8828	0.8809	- 0.4707	- 0.4707
W^6	0.8320	0.8320	- 0.5547	- 0.5547
W^7	0.7734	0.7734	- 0.6348	- 0.6348
W^8	0.7070	0.7090	- 0.7070	- 0.7090
W^9	0.6348	0.6348	- 0.7734	- 0.7734
W^{10}	0.5547	0.5547	- 0.8320	- 0.8320
W^{11}	0.4707	0.4707	- 0.8828	- 0.8828
W^{12}	0.3828	0.3828	- 0.9238	- 0.9258
W^{13}	0.2910	0.2891	- 0.9570	- 0.9570
W^{14}	0.1953	0.1953	- 0.9805	- 0.9805
W^{15}	0.0977	0.0977	- 0.9961	- 0.9941
W^{16}	0.0000	0.0000	- 1.0000	- 1.0000
W^{18}	-0.1953	-0.1953	- 0.9805	- 0.9805
W^{21}	-0.4707	-0.4707	- 0.8828	- 0.8809
W^{22}	-0.5547	-0.5547	- 0.8320	- 0.8320
W^{24}	-0.7070	-0.7090	- 0.7070	- 0.7090
W^{27}	-0.8828	-0.8809	- 0.4707	- 0.4707
W^{30}	-0.9805	-0.9805	- 0.1953	- 0.1953
W^{32}	-1.0000	-1.0000	0.0000	0.0000
W^{33}	-0.9961	-0.9941	0.0977	0.0977
W^{36}	-0.9238	-0.9258	0.3828	0.3828
W^{39}	-0.7734	-0.7734	0.6348	0.6348
W^{42}	-0.5547	-0.5547	0.8320	0.8320

Seuls les coefficients nécessaires pour résoudre la FFT sont présentés, les autres coefficients ne se sont simplement pas utiles. Nous constatons que les coefficients obtenus

à partir des AG sont proches de ceux fournis par la méthode tronquée. Cependant, les légères variations apportées par l'optimisation obtenue procurent un SQNR supérieure.

III.4.4. Étude de comparaison de SQNR

En respectant la méthodologie d'évaluation du SQNR d'écrit par la figure 3.7, le $SQNR_{max}$ est calculé de façon moyennée en générant des données d'entrées qui se propagent à travers les étages de FFT pour déterminer la précision des l'algorithme.

Les courbes de SQNR en fonction du nombre de générations sont données aux figures 3.12 et 3.13 pour deux longueurs de FFT, 16 et 64 points.

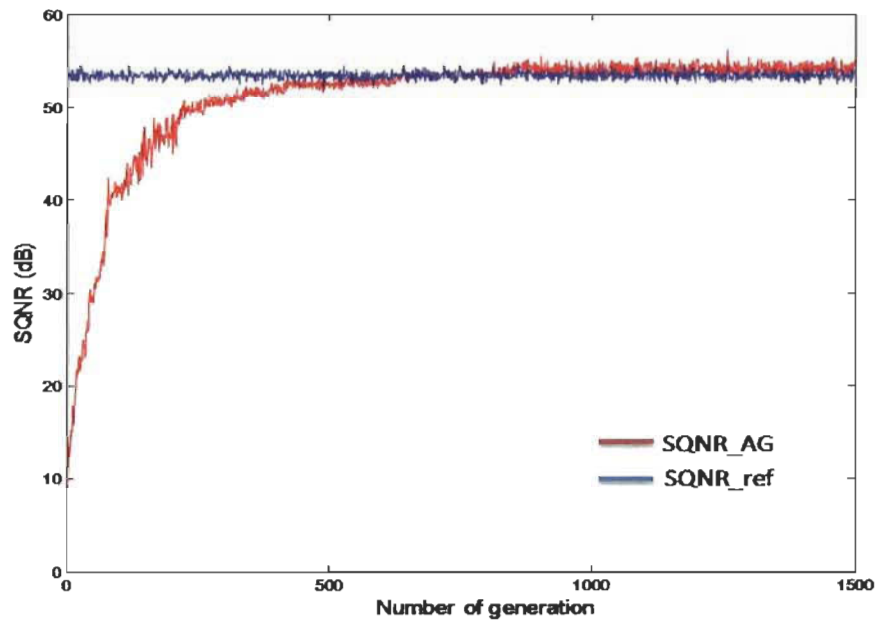


Figure 3.5: Convergence de SQNR en fonction de nombre de génération

($N=16$, $wl_x = v.$ flottante, $wl_W=10$)

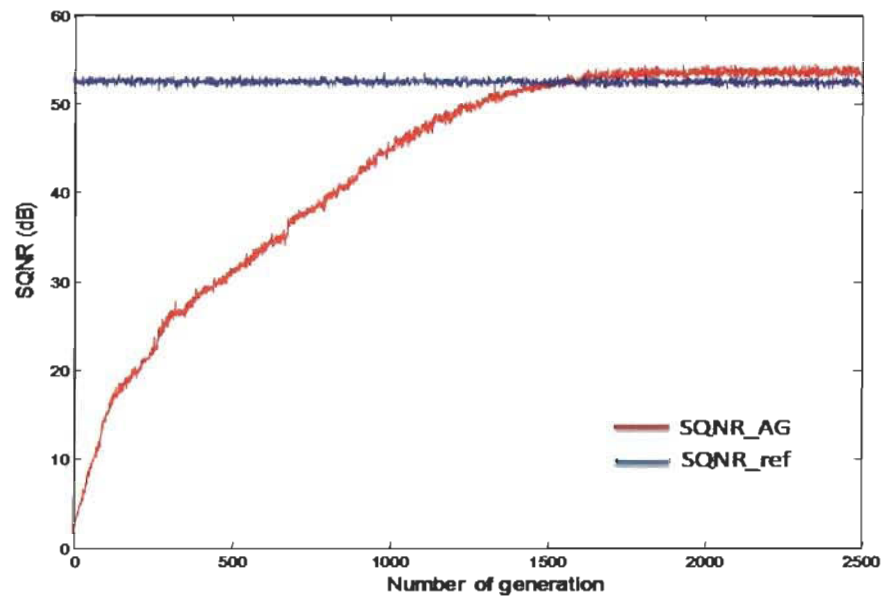


Figure 3.63: Convergence de SQNR en fonction de nombre de génération

(N=64, wl_x = v. flottante, wl_W=10)

Les tracer des figures 3.12 et 3.13 représentent l'évolution du SQNR dans l'espace de recherche par rapport au nombre de générations. Nous constatons que la méthode proposée (SQNR_AG) converge progressivement jusqu'à ce qu'il atteigne le SQNR_ref obtenu par la méthode de troncature. Avec une FFT de 16 points, à partir de la génération 900, nous apercevrons un gain de 0.8 dB fourni par la méthode proposée (AG). Ainsi avec une FFT de 64 points nous pourrions bien remarquer un gain de 1.2 dB par la méthode des AG par rapport à la méthode de troncature.

Pour valider les coefficients à l'aide de notre méthode d'optimisation, nous avons généré 10^5 signaux $x(n)$ aléatoires de longueur N . Nous avons propagé ces signaux à travers la FFT en virgule fixe en utilisant d'une part les coefficients tronqués et les coefficients obtenus

par les AG. Le SQNR a été calculé selon l'équation (3.1) pour l'ensemble des signaux propagés.

Afin de bien visualiser la qualité de nos résultats obtenus, nous avons représenté la répartition de la dynamique de SQNR en fonction de nombres des unités qu'elles conviennent pour le cas de troncature et AG sous forme d'histogramme (voir les figures 3.14, 3.15, 3.16). Ces résultats sont réalisés pour $x(n)$ en virgule flottante et les coefficients de Fourier en virgule fixe de longueur de 10-bit.

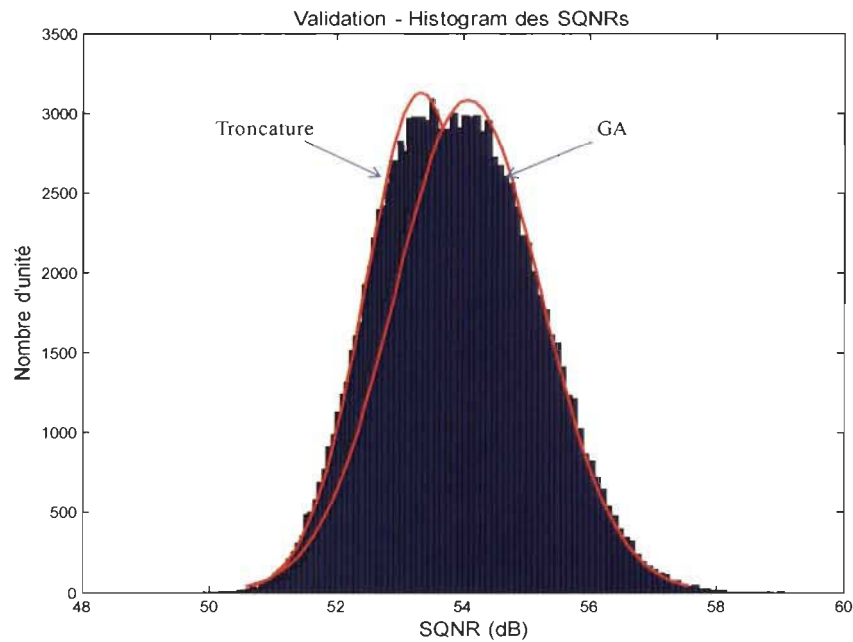


Figure 3.7: Histogrammes de la dynamique de SQNR (N=16, $wl_x = v.f$, $wl_W=10$)

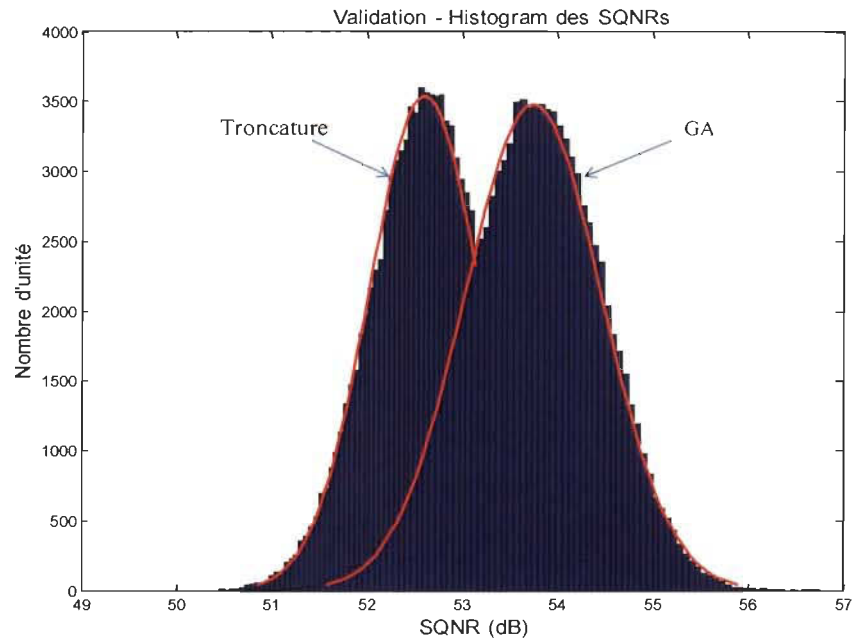


Figure 3.8: Histogrammes de la dynamique de SQNR (N=64, $wl_x = v.f$, $wl_W = 10$)

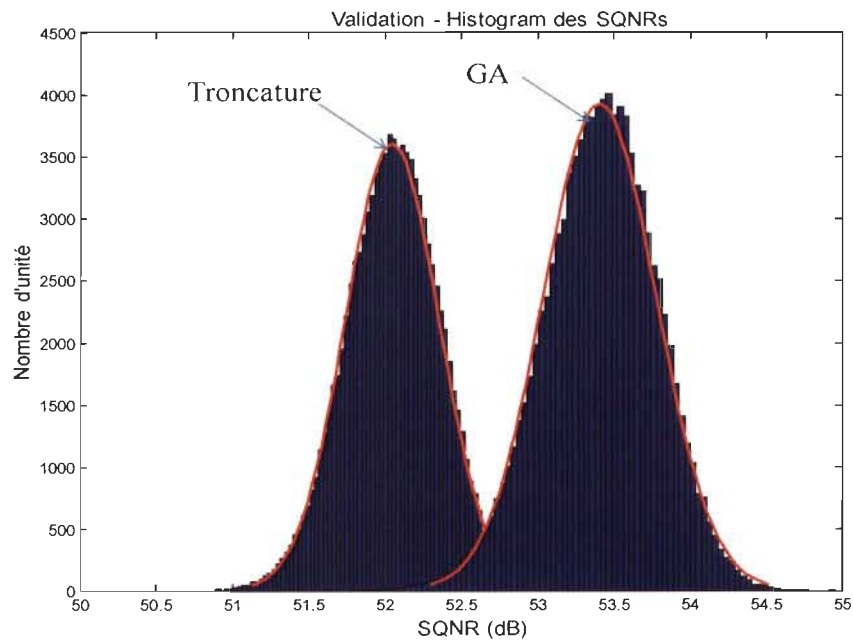


Figure 3.9: Histogrammes de la dynamique de SQNR (N=256, $wl_x = v.f$, $wl_W = 10$)

Les histogrammes des figures (3.14, 3.15, 3.16) montrent un mélange de deux répartitions de SQNR ayant une moyenne différente présentée sous la forme de deux courbes en cloche pour différentes FFT de longueur ($N=16, 64, 256$) respectivement. Il est clair que le SQNR_moyen indiqué par le centre de la cloche associée à la méthode de l'algorithme génétique a une valeur plus grande que la méthode de troncature. Nous remarquerons un gain de SQNR égale à 0.8 dB, 1.2 dB et 1.5 dB obtenus par la méthode proposée basée sur les algorithmes génétiques par rapport à la méthode de troncature pour une FFT de longueur ($N=16, 64, 256$) respectivement. Nous distinguerons aussi que le gain de SQNR augmente avec la longueur de la FFT, se qui reflète la supériorité de la méthode proposée basée sur les algorithmes génétiques par rapporte à la méthode de troncature.

Nous avons aussi étudié statistiquement les valeurs max des SQNR obtenues par la méthode des AG et celui de la méthode de troncature à l'aide d'une analyse de la variance (ANOVA).

III.5. ANALYSE DE VARIANCE DES SQNR

L'analyse de la variance (ANOVA : *ANalysis Of VAriance*) est une approche statistique portant sur des moyennes servant à exploiter la distribution des échantillons des résultats d'expérimentations disposées suivant certains plans [MOR55]. Elle a pour objectif d'étudier l'influence d'un ou plusieurs facteurs sur une variable quantitative.

D'une manière générale, dans notre travail, l'objectif d'utiliser ANOVA vise à évaluer les différences significatives de SQNR. L'outil d'ANOVA disponible sur Matlab a été utilisé

pour représenter les distributions des SQNR pour les deux méthodes de troncature et celle des AG. Les résultats obtenus pour différentes longueurs de FFT sont présentés dans la figure 3.17, la figure 3.18, et la figure 3.19 pour $x(n)$ en virgule flottante et une largeur binaire de 10 bits les coefficients.

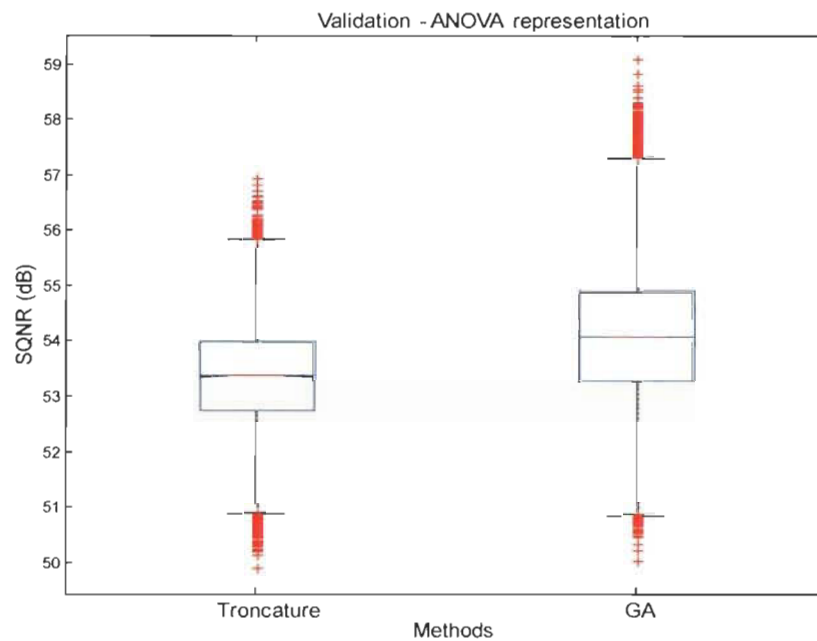


Figure 3.10: ANOVA des distributions statistiques des SQNR ($N=16$, $wl_x=v.f$, $wl_W=10$)

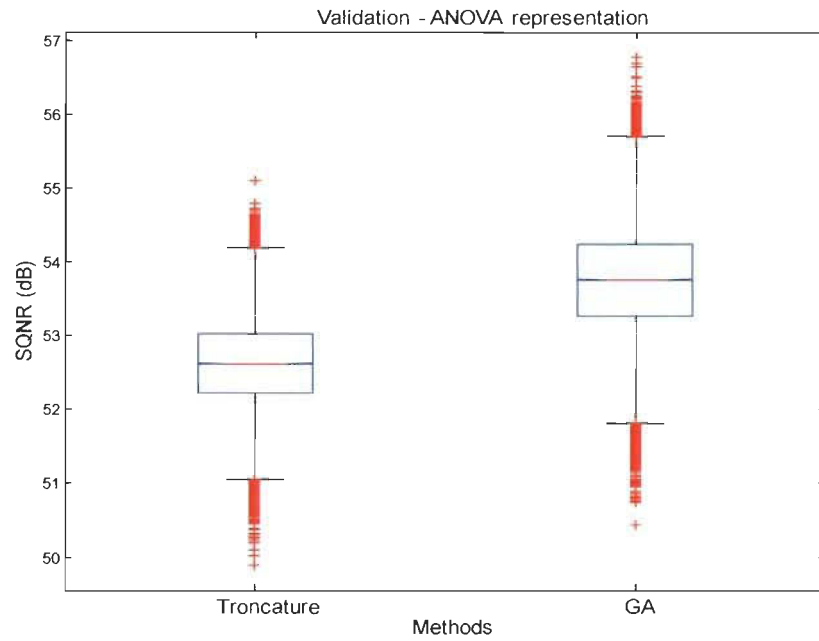


Figure 3.11: ANOVA des distributions statistiques des SQNR (N=64, $wl_x=v.f$, $wl_W=10$)

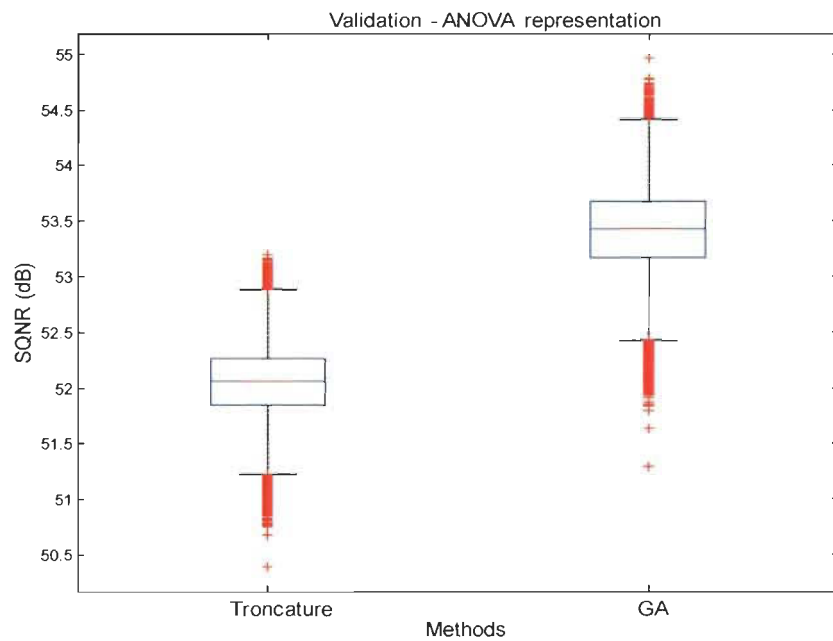


Figure 3.12: ANOVA des distributions statistiques des SQNR (N=256, $wl_x=v.f$, $wl_W=10$)

Sur ces boxplots d'ANOVA, nous constatons facilement que les moyennes varient avec la longueur de la FFT. Les valeurs moyennes de SQNR obtenues par la méthode des AG sont plus grandes que celle de la méthode tronquée. Suivant les figures 3.17, 3.18 et 3.19, il est bien clair que la méthode proposée basée sur les AG permettrait d'obtenir des gains égaux à 0.8 dB, 1.2 dB et 1.5 dB en SQNR pour une FFT de longueur ($N=16, 64, 256$) respectivement.

De plus, la distribution des SQNR par rapport à la moyenne est répartie en deux groupes supérieur et inférieur. À partir de ces résultats, nous constatons que le nombre de points de SQNR obtenus avec les AG et le maximum de ces points est plus grand que le nombre de points de la méthode tronquée et le maximum de ces points de même pour le groupe de points inférieur. Afin de concrétiser au mieux les résultats obtenus, nous considérons par la suite la transmission d'un paquet de données d'entrée d'une largeur binaire $wl_x=16$ bits. Dans les mêmes conditions décrites précédemment, l'évaluation de SQNR selon ANOVA est représentée dans la figure 3.20, et la figure 3.21 pour une largeur binaire de 16 bits pour $x(n)$ et de 10 bits pour les coefficients.

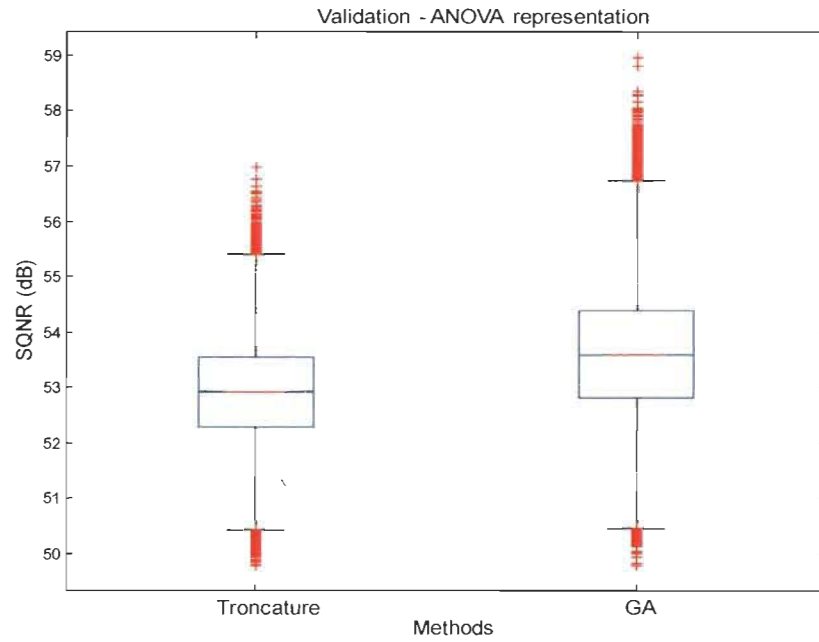


Figure 3.20: ANOVA de distributions statistiques des SQNR (N=16, $wl_x=v.f$, $wl_W=10$)

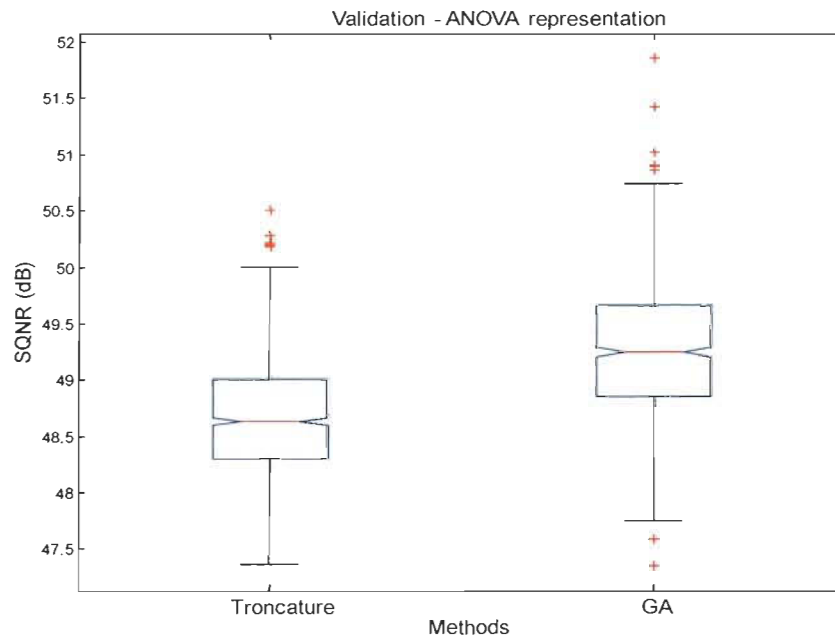


Figure 3.13: ANOVA de distributions statistiques des SQNR (N=64, $wl_x=v.f$, $wl_W=10$)

Dans les figures 3.20 et 3.21, nous remarquerons un gain de 0.6 dB et 0.8 dB obtenu par la méthode proposée basée sur les AG pour une FFT de longueur (N=16, 64) respectivement.

Si nous comparons les résultats d'ANOVA illustré par la figure 3.20, et la figure 3.21 correspondent à $wl_x=10$, et celles de la figure 3.17, et la figure 3.18 correspondent à $wl_x=16$, pour une FFT de 16 et 64 points, nous constatons d'une part que les SQNR_moyen obtenus par la méthode des AG et plus grands à celui de troncature. D'autre part, nous observons une augmentation des points SQNR dans le groupe supérieur et une diminution des points dans le groupe inférieur pour les deux cas (N=16,64) en utilisant les AG.

III.6. ALGORITHME GÉNÉTIQUE À OBJECTIF MULTIPLE

Dans les sections précédentes, nous avons considéré l'optimisation à objectif unique où le problème à traiter était de maintenir le SQNR à sa valeur maximale. Pratiquement, de plus en plus des problèmes exigent la considération simultanée de plusieurs objectifs s'opposant. En effet, le SQNR augmente avec la largeur binaire des coefficients cependant la consommation de puissance augmente aussi avec la largeur binaire des coefficients. Deux critères ayant des directions opposées à ce que nous désirons.

Nous avons ajouté un deuxième critère d'optimisation afin d'optimiser la consommation d'énergie. L'équation (3.7) représente la fonction multi sélective utilisée à la fois pour maximiser le SQNR et minimiser la consommation d'énergie.

$$f = \alpha \cdot \text{SQNR} + \beta \cdot \frac{1}{\text{consommation}} \quad (3.7)$$

où: α et β sont des constantes appelées, les poids ; seront adaptés par tâtonnement jusqu'à l'obtention d'une solution acceptable. C'est-à-dire, aura un SQNR élevé et une faible consommation et vice-versa.

Des essais de simulations ont été réalisés avec une fonction multi sélective représenté par l'équation (3.7), mais elle présente une optimisation non triviale rendant l'optimisation à la fois du SQNR et consommation extrêmement sensible et créant rapidement des divergences. C'est pourquoi dans ce mémoire nous allons nous concentrer sur le gain de SQNR. Ce problème est devenu très difficile à résoudre, et une étude séparée sera réalisée par d'autres sujets de recherche à la maîtrise ou doctorat au laboratoire LSSI.

3.7. CONCLUSION

Dans ce chapitre, les algorithmes génétiques ont été utilisés pour optimiser la largeur des coefficients FFT. Les améliorations en termes de SQNR, proposées par la technique des AG par rapport à l'approche de troncature qui est la référence ont été détaillées. Les résultats des expérimentations montrent l'intérêt des algorithmes génétiques pour l'optimisation des coefficients FFT pour améliorer la qualité de la solution. Dans toutes nos applications, nous avons obtenu un SQNR_max par la méthode des AG plus grande que le SQNR_max obtenue par la méthode de troncature en fonction des paramètres de la FFT. Cela signifie qu'en peut utiliser une largeur binaire des coefficients plus faibles que celle utilisée dans nos calculs. Les résultats prouvent aussi qu'il est souhaitable d'utiliser une largeur binaire des données d'entrées plus grande que celle des coefficients FFT.

CHAPITRE IV

IMPLEMENTATION D'UNE FFT RADIX-4 MDC À 16 POINTS

Sachant que le processeur IFFT/FFT est l'une des composantes intensives en calcul dans les communications OFDM, son implémentation est soumise à de fortes contraintes de consommation électrique et de coûts matériels. Aussi, l'implémentation des algorithmes FFT est une tâche indispensable et difficile. De plus, les puces FPGA deviennent de plus en plus une cible architecturale envisageable pour la conception des différents algorithmes FFT en raison de son rapide progrès dans la technologie d'intégration à très grande échelle (VLSI). Ainsi, l'estimation de puissance des algorithmes FFT est une étape nécessaire dans l'implémentation. Dans ce chapitre, nous étudierons l'implémentation de la FFT pipeline de

type radix-4 MDC pour fin de démonstration avec une longueur limitée à 16 points sur une plateforme FPGA ainsi que les résultats obtenus pour l'estimation de la consommation de puissance.

IV.1. LA CONSOMMATION EN PUISSANCE DANS LES CIRCUITS FPGA

Grâce à ces avantages grandissants en capacité de traitement, leur programmabilité, et ces motivations pour répondre aux exigences concernant la puissance de calcul. Les circuits FPGA sont de plus en plus utilisés pour le prototypage ou la fabrication des systèmes numériques. Actuellement, les FPGA présentent beaucoup de perspectives et des performances remarquables (vitesse, surface occupée, consommation d'énergie) pour l'implémentation de plusieurs algorithmes en arithmétique à virgule fixe. En effet, la consommation en puissance est un facteur majeur dans le cadre des applications et la conception des systèmes numériques et les fabricants ont porté beaucoup d'importance pour ce facteur critique.

La consommation de puissance est définie comme la quantité d'énergie consommée durant certains temps. Elle a deux sources principales [GAR99]:

- La consommation statique est dominée par le courant de fuite. Elle est liée aux ressources utilisées et le coût en consommation de chaque ressource, elle peut être définie comme suit :

$$P_s = V_{cc} \cdot A \quad (4.1)$$

- La puissance dynamique fait référence à l'énergie consommée durant le fonctionnement du circuit. Elle est due à l'activité du signal dans le circuit ou le taux de commutation. La puissance dynamique peut être modélisée en utilisant l'équation (4.2):

$$P_d = V_{cc}^2 \cdot F \cdot \alpha \cdot C \quad (4.2)$$

où : V_{cc} est la tension d'alimentation, F est la fréquence de fonctionnement (α) est l'activité de commutation, c 'est la capacité de charge, et A est la surface occupée.

VI.2. PROCESSUS D'ESTIMATION DE PUISSANCE DE LA FFT RADIX-4

Le processus de développement pour l'estimation de puissance utilise le langage de description matériel VHDL et des outils de Xilinx pour synthétiser ce langage.

Ces mêmes outils fournissent un logiciel qui permet d'estimer la consommation de puissance dite, XPower Analyzer (XPA). La méthode d'implantation d'algorithmes au sein de FPGA spécifié en utilisant l'arithmétique virgule fixe est détaillée à la figure 4.1.

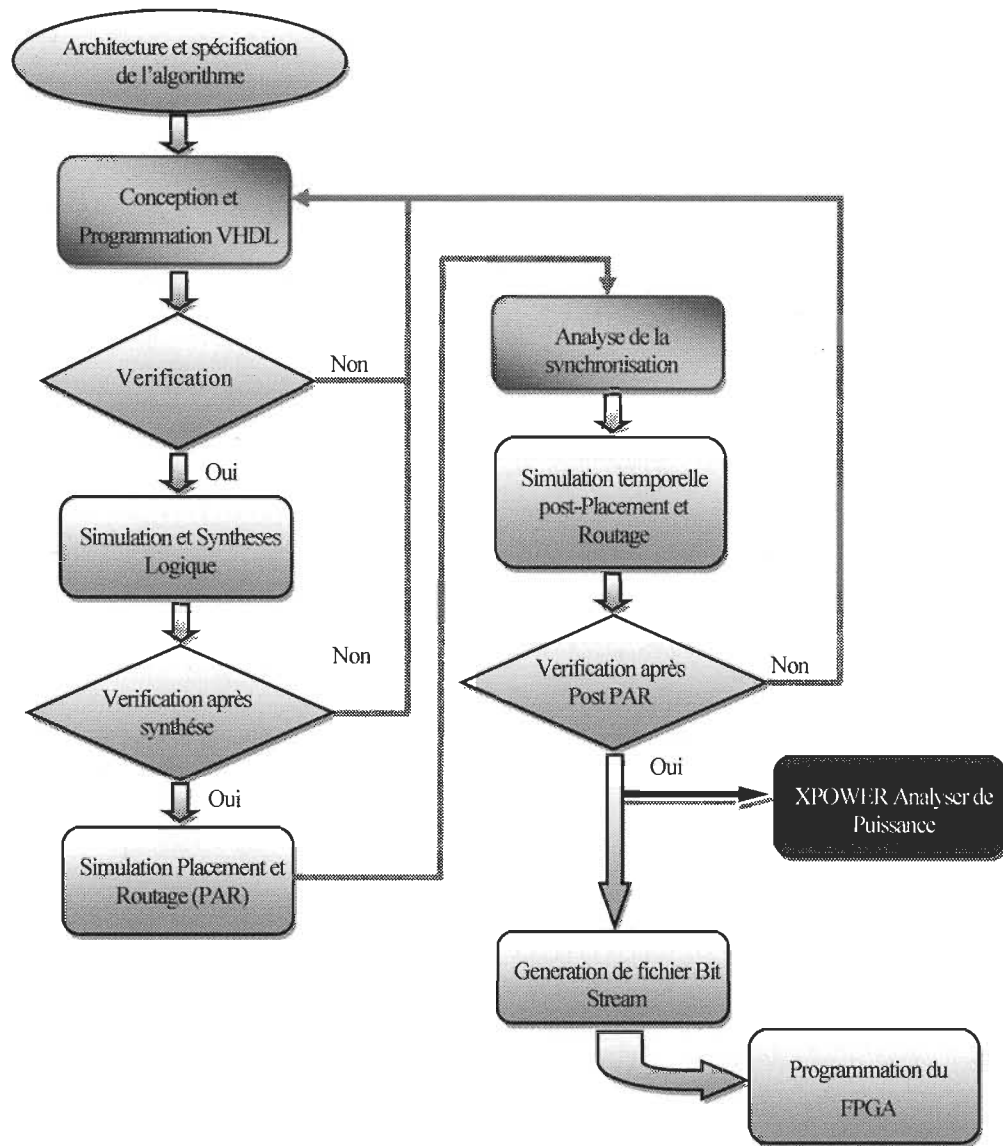


Figure 4.1: Organigramme de cycle d'implémentation et l'estimation de puissance

La méthodologie proposée suivant la figure 4.1 se décompose en trois étapes distinctes. La première étape intitulée la description hiérarchique et spécification des blocs qui caractérisent l'architecture pipeline radix-4 MDC à 16 points. Par la suite, nous créons une

description textuelle en langage VHDL en utilisant le logiciel Modelism de Mentor Graphics. Afin d'assurer le bon fonctionnement de code VHDL, l'étude comparative en virgule fixe VHDL vs Matlab nous a permis de bien vérifier la fonctionnalité du design. Si le code est fonctionnel, nous passons directement à la deuxième étape.

L'outil Xilinx ISE prend comme entrée la description synthétisable en VHDL pour produire la liste des interconnexions (netlist) composées d'instances des cellules élémentaires ciblant le circuit FPGA dédié. La synthèse d'architecture repose sur une bibliothèque d'opérateurs arithmétiques virgule fixe associée à une famille de FPGA ciblée. Cependant, des erreurs peuvent apparaître, dans ce cas, il est nécessaire de vérifier et corriger le code VHDL et refaire la synthèse logique. Une fois, la synthèse est réalisée, nous passons au placement routage, et nous procurant un rapport d'information avec les pourcentages de surfaces utilisées, le temps de propagation et les nombres des portes FPGA utilisés. Ces temps de propagation déterminent la fréquence maximale du design. Nous passons après à la prochaine phase de post placement et routage (Post-PAR) ou la simulation temporelle, ici nous pourrions vérifier dynamiquement la fonctionnalité du design à l'aide d'un fichier test fourni. L'étape de PAR consiste à tenter d'augmenter la fréquence d'horloge et de diminuer les ressources matérielles, ce qui diminue la puissance nécessaire à la conception. Finalement, après la génération de bits-Stream qui est le fichier de configuration chargé dans le circuit FPGA choisi, le dessin peut être réalisé dans le matériel ciblé, et nous testons le design dans son vrai environnement.

CHAPITRE IV- IMPLÉMENTATION D'UNE FFT RADIX-4 MDC À 16 POINTS

Dans l'étape trois, la puissance moyenne consommée a été caractérisée avec l'outil ISE/Xpower Analyzer (XPA) de Xilinx. XPA rassemble des informations de conception dans la liste suivante des fichiers optionnels :

- Le fichier XML, les informations telles que le nombre de ressources utilisées (cellules logiques, multiplieurs dédiés, etc.), le temps de traversée du composant où sa consommation sont finalement sauvegardées sous la forme d'une base de données XML.
- Le fichier NCD (Native Circuit Description), est un fichier résultat de processus de PAR. Il représente le détail de circuit physique du dispositif FPGA spécifié.
- la fiche contrainte physique PCF (Physical Constraints File), il contient des informations pour aider à déterminer les fréquences d'horloge et le voltage. Le fichier PCF est important pour des résultats précis de la puissance
- Le fichier d'activité VCD (Value Change Dump), ce scripte est généré en utilisant ModelSim pour pouvoir obtenir de l'information de l'activité. Il englobe des informations de commutations telles que le taux de bascule, taux de signalisation et des informations de fréquences.

Le XPA accepte les fichiers résultats de PAR NCD, et PCF afin de déterminer les interconnexions I/O et les informations de l'horloge du design et avec ces informations. Le XPA construit une représentation hiérarchique du modèle. En outre XPA utilise le fichier VCD provenant de ModelSim™. Le fichier VCD fournit un paquet de renseignements descriptifs de l'activité de commutation, ceci donne une estimation précise de consommation de puissance.

IV.3. CONCEPTION ET ARCHITECTURE PIPELINE RADIX-4 FFT

Nous avons bien détaillé dans le chapitre II la description mathématique de la FFT radix-4. En se basant sur cette analyse, cette section explicite toutes les informations pertinentes sur les étapes de conception de la FFT radix-4 à 16 points. L'architecture pipeline R4MDC et les processeurs élémentaires constitutants, ainsi que le rôle de chaque processeur, seront discutés individuellement et en détaille. L'évolution de l'ensemble des calculs, incluant les différentes étapes, peut donc être mieux visualisée par l'utilisation de graphe de fluence qui fait apparaître tous les BPE intervenant dans la FFT radix-4. La figure 4.2 donne à titre d'exemple le diagramme de fluence de la FFT radix-4 MDC de type DIF pour $N=16$. Ce diagramme de fluence peut être généralisé pour tout N puissance de 4. Nous constaterons que le calcul de la FFT est réalisé en deux étages. Les variables $x(0)$ jusqu'à $x(15)$ sont les valeurs d'entrée pour le calcul FFT et $X(0)$ jusqu'à $X(15)$ sont désignés comme des sorties.

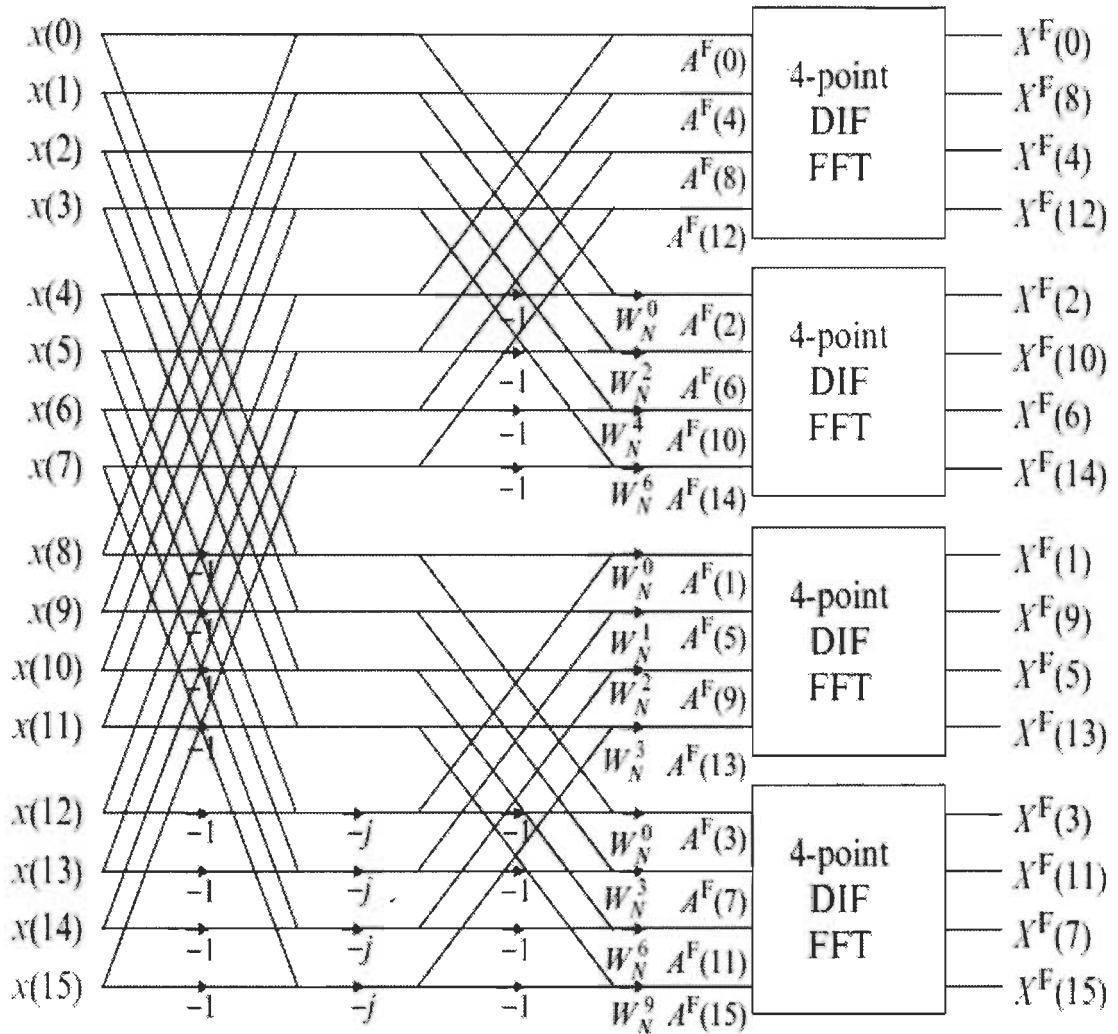


Figure 4.2: Graphe de fluence des entrées/sorties de la FFT radix-4 MDC de type DIF pour $N=16$ [RAO10]

Le graphe de fluence de la figure 4.2 sera d'ailleurs notre point de départ pour l'implémentation de la FFT sur des architectures pipeline. Plaçons-nous maintenant dans le cas d'un design qui comporte plusieurs parties fonctionnantes, les blocs fondamentaux qui

constituent la FFT radix-4 MDC à 16 points sont implémentés selon la structure donnée par la figure 4.3.

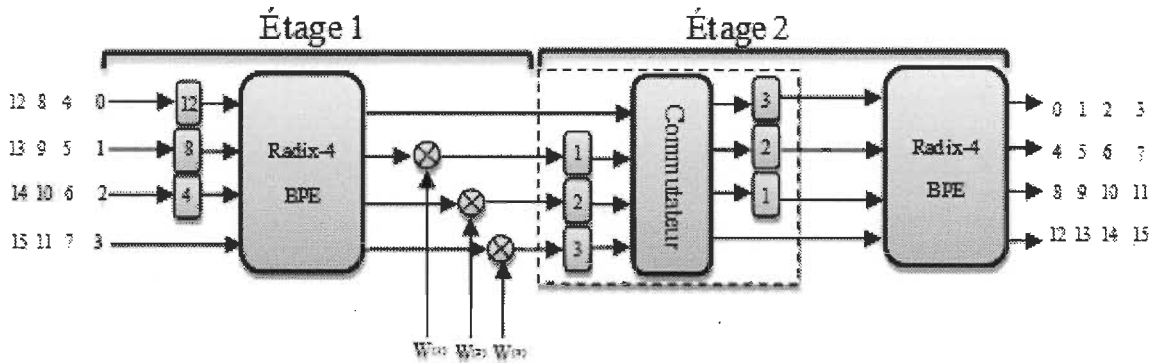


Figure 4.3: Architecture pipeline radix-4 MDC de type DIF à deux étages

Nous pouvons bien saisir que l'architecture pipeline de radix-4 MDC à 16 points a étendu en deux étages, chaque étage inclut des processeurs élémentaires (PE). Le premier étage traite quatre trames de quatre données d'entrée, connectées chacune avec des éléments de délais au début de l'architecture. Ces dernières ont utilisé pour organiser les données d'entrée. Les sorties des éléments de délais sont les entrées de premier processeur élémentaire (PE1), qui est le BPE radix-4 suivi par des multiplieurs complexes avec W coefficients de la FFT ou « twiddle factor ». Les sorties du premier étage sont reliées aux entrées de deuxième étage. Ce dernier se constitue d'un deuxième processeur élémentaire (PE2), qui est le commutateur inclut des éléments de délais avant et après le bloc, son rôle est de mettre les données qui viennent de premier étage dans le bon ordre afin de faire les opérations papillon de deuxième BPE. Les sorties du deuxième étage sont les sorties de la FFT pipeline.

IV.3.1. BPE radix-4 FFT

Le premier processeur élémentaire nommé, BPE radix-4 FFT est le cœur de calcul de l'opérateur FFT. Il est composé de trois opérateurs réels indispensables aux circuits de traitement du signal permettant l'exécution des opérations arithmétiques complexe, tels que: les multiplieurs, les additionneurs, et les soustracteurs. La structure interne qui définit le principe de l'opération papillon est montrée dans la figure 4.4 [SWA84].

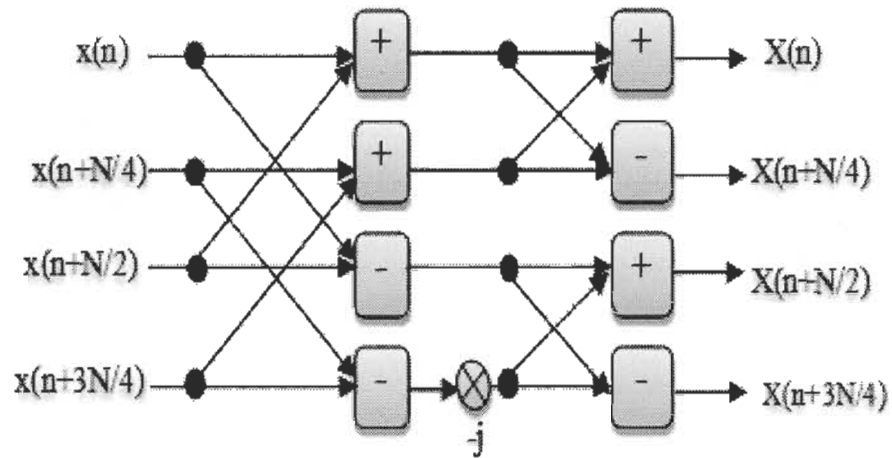


Figure 4.4: Structure interne du BPE radix-4 FFT à base d'opérateurs réels.

Les opérations papillon illustrées par la figure 4.3 consistent à prendre quatre entrées en nombres complexes, et calculer les quatre sorties. ces opérateurs exécutent les opérations binaires sur des nombres complexes. Cette opération requiert quatre additions et trois soustractions complexes, ainsi qu'une multiplication triviale par $-j$. La multiplication par $-j$ trivial est réalisée sans coût matériel supplémentaire par une simple permutation de la partie réelle et imaginaire du produit comme le montre l'équation (4.1).

$$(a + bj) \times (-j) = b - aj \quad (4.1)$$

IV.3.2. Multiplication complexes

Le multiplieur complexe est le bloc le plus couteux dans l'architecture pipeline radix-4. Il doit être optimisé maximum possible et représenté avec un minimum de multiplieur. L'équation (4.2) représente la multiplication classique de deux nombres complexes.

$$(A + Bj)(C + Dj) = (AC - BD) + j(BC + AD) \quad (4.2)$$

L'équation (4.2) réclame quatre multiplieurs et quatre additionneurs réels. D'un point de vue matériel cette représentation utilise une surface d'implémentation importante. L'équation (4.3) montre une autre représentation avantageuse d'un multiplieur complexe.

$$(A + Bj)(C + Dj) = \{C(A - D) + B(C - D)\} + j\{D(A + B) + B(C - D)\} \quad (4.3)$$

La structure interne de multiplieur adopté dans notre travail est donc réalisée avec trois multiplieurs et cinq additionneurs réels représentés par la figure 4.5. Soit une addition de plus mais un multiplieur de moins que la version précédente

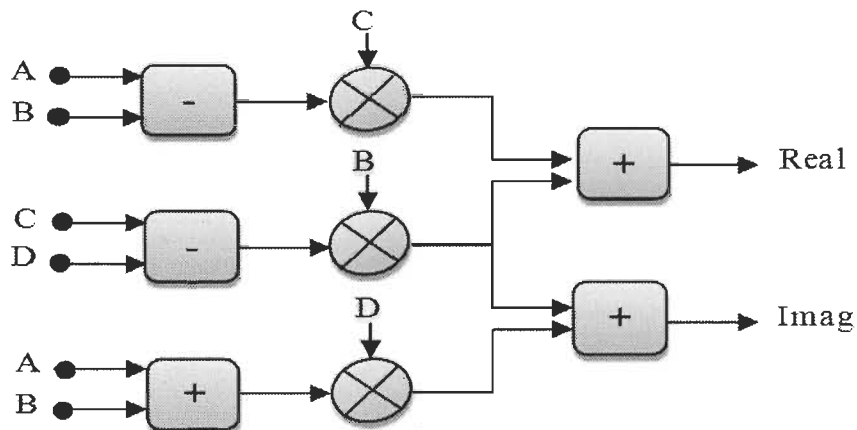


Figure 4.5: Structure interne de multiplieur complexe

IV.3.3. Le commutateur

Le commutateur est un bloc indispensable et non trivial dans le calcul de la FFT radix-4 MDC. Il est utilisé entre les étages de radix-4 papillon dans l'architecture pipeline. Il stocke temporairement une partie des données durant le cycle de calcul FFT afin de les réorganiser de nouveau. Le bloc commutateur se constitue des multiplexeurs, de simples éléments de délais, et un compteur à deux bits pour l'activation des multiplexeurs. Le nombre des éléments de délais dépend exactement de la longueur de la FFT. La structure interne de commutateur est donnée par la figure 4.6 [JUN03].

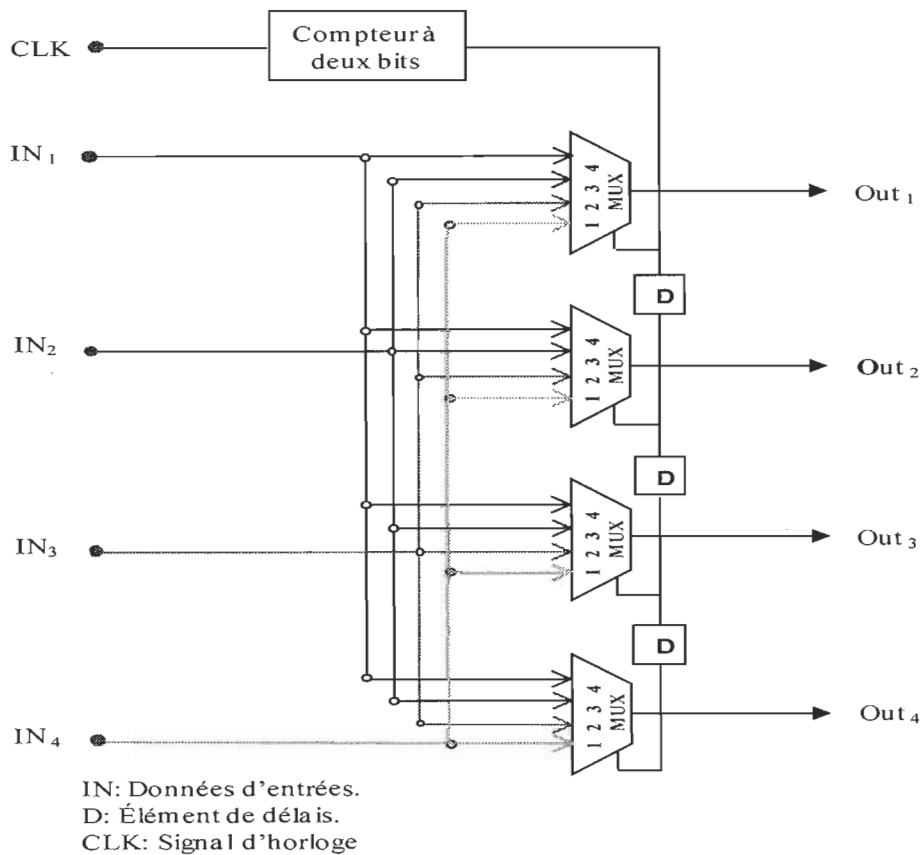


Figure 4.6: Structure interne de commutateur

La structure de commutateur se constitue de quatre multiplexeurs en parallèle. Chaque multiplexeur a quatre entrées et une sortie. Les multiplexeurs sont aussi connectés par un compteur à deux bits pour l'activation de chaque multiplexeur. Un autre schéma représentatif des différentes transformations de commutateur est donné dans la figure 4.7 [SWA84] :

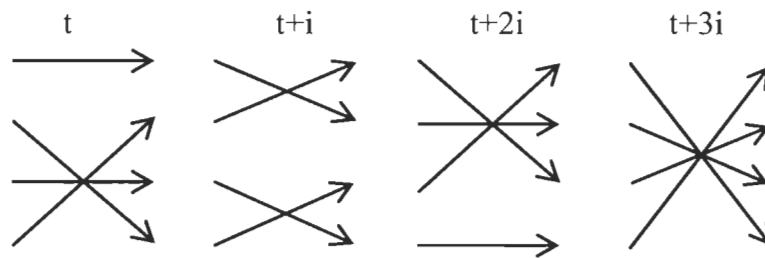


Figure 4.7: Les transformations de commutateur

IV.3.4 Evaluation de SQNR de l'architecture pipeline radix-4 MDC à 16 points

Nous avons effectué une série de simulation afin d'analyser l'erreur de quantification de l'architecture radix_4MDC à 16 points en terme de SQNR. Pour cela, nous avons programmé en VHDL le module FFT R4MDC avec une représentation en virgule fixe complément à deux des données et des coefficients W , et nous avons comparé les résultats obtenus en virgule fixe avec celle obtenue en virgule flottante.

Le tableau 4.1, et le tableau 4.2 ci-dessous résument les différentes valeurs de SQNR. Ces résultats sont obtenus dans les simulations de la FFT pipeline R4MDC à deux étages en utilisant les coefficients tronqués (W_{Tronc}) et les coefficients génétiques (W_{AG}) présentés au

CHAPITRE IV- IMPLÉMENTATION D'UNE FFT RADIX-4 MDC À 16 POINTS

chapitre III, tableau 3.1. Dans nos simulations, nous considérons le cas de la largeur binaire fixée à 10 bits. De plus, les résultats sont obtenus pour 10 réalisations aléatoires du signal d'entrée d'une largeur binaire de 10 bits et 16 bits.

Tableau 4.1 : Évaluation de SQNR en utilisant les W_{Tronc}

Largeur des données FFT	largeur de TWF	SQNR (dB)										Moy (dB)	Écart type
		X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀		
10	10	52.38	54.15	61.10	55.03	56.65	55.14	53.73	56.71	56.84	55.64	55.73	2.24
16	10	52.41	54.15	61.08	54.98	57.65	56.30	53.40	56.71	56.76	55.62	55.90	2.32

Tableau 4.2: Évaluation de SQNR en utilisant les W_{AG}

Largeur des données FFT	largeur de TWF	SQNR (dB)										Moy (dB)	Écart type
		X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀		
10	10	53.15	55.22	61.77	55.83	56.79	57.74	55.03	58.11	56.98	56.94	56.75	2.17
16	10	54.01	56.42	61.63	56.12	58.35	56.39	55.85	59.71	57.06	56.92	57.24	2.04

D'après les résultats des tableaux 4.1 et 4.2, on remarque que le SQNR maintient une valeur plus élevée pour différents signaux d'entrées X. De plus, nous enregistrons une augmentation de SQNR en utilisant les coefficients génétiques W_{AG} par rapport aux coefficients tronqués W_{Tronc} .

Nous constatons aussi que l'emploi d'une largeur binaire des données d'entrée plus grande que la largeur binaire des coefficients W donne plus de performance en terme de SQNR.

Donc, dans une FFT pipeline il est avantageux d'utiliser une largeur binaire des données circulant dans les étages supérieurs à celle des coefficients W . Ces résultats de SQNR confirment aussi que notre module R4MDC est fonctionnel et procure les mêmes conclusions que les résultats obtenus par simulation.

IV.4 IMPLÉMENTATION FPGA ET ANALYSE DES RÉSULTATS

Dans cette section, les résultats de simulation de l'implémentation de l'architecture pipeline de la FFT Radix-4 MDC à 16 points décrite précédemment sur une puce FPGA choisie sont présentés, ainsi que les performances de l'algorithme FFT en termes de surface et la fréquence maximale de fonctionnement.

IV.4.1 Résultats de synthèses de radix-4 FFT à 16 points

L'algorithme FFT est modélisé en utilisant le langage VHDL et mis en œuvre sur FPGA du type Virtex-4 (xq4vsxx55-10ff1148) de Xilinx. L'environnement de travail nommé ISE pour «Integrated synthesis environment» ou environnement de synthèse intégré est utilisé pour synthétiser et simuler le code VHDL conçu. Les ressources matérielles utilisées lors de l'implémentation de processeur FFT radix-4 à $N=16$ sont représentés dans le tableau 4.3, pour des largeurs binaires des données d'entrées et de coefficient égal à 16 bits.

CHAPITRE IV- IMPLÉMENTATION D'UNE FFT RADIX-4 MDC À 16 POINTS

Tableau 4.3: Évaluation des ressources matérielles (N=16, wl_x=16,wl_W=16)

Utilisation logique	Utilisé	Disponible
Nombre de Slices Flip Flops	1162	49152
Nombre de Slices	1413	24576
Nombre de LUT à 4 entrées	2806	49152
Nombre de IOBs	354	640
GCLKs	1	32
Nombre de DSP48 18×18 bits	9	512

Les ressources logiques des FPGA de Xilinx sont constituées de blocs logiques configurables basés sur les slices. Nous constatons que 1413 des ressources logiques du FPGA sont utilisées, obtenues avec une FFT à 16 points pour une largeur binaire des données et des coefficients égale à 16 bits. Sachant que chaque slice est composé de deux tables appeler Look Up Table (LUT) et deux registres de 16 bits.

Noter que les registres à décalages jouent un rôle principal dans la technologie VLSI, ils sont utilisés dans les décalages de bit réalisé dans la génération des adresses des échantillons et des coefficients. La synthèse logique indique que seulement 1162 flips flops qui sont configurés en registres à décalage. De plus, l'architecture utilise 354 des entrées/sorties (input/output block IOB). Il est à noter que les blocs DSP48 se composent d'un multiplieur complément à deux entrées de 18 bits, suivi par un additionneur/soustracteur/ accumulateur. D'après les résultats de tableau 4.1, nous constatons aussi que le processeur FFT obtenu utilise 9 DSP48, et chaque DSP48 peut donc réaliser une opération de multiplication, multiplication/addition, multiplication/accumulation, ou simplement une

CHAPITRE IV- IMPLÉMENTATION D'UNE FFT RADIX-4 MDC À 16 POINTS

addition. Finalement, les outils de placement et routage permettent d'estimer la fréquence maximale de fonctionnement du R4MDC à 16 points. La fréquence maximale d'opération sur FPGA obtenu est 56.8 MHZ.

Nous fixons maintenant la largeur des coefficients à 10 bits. Les tableaux 4.4 et 4.5 représentent les résultats des ressources matérielles pour la largeur des données d'entrées égale à 16 bits et 10 bits respectivement. Nous constatons, d'une part, que la FFT utilise 730 des ressources logiques disponibles avec une largeur binaire des données égale à 10 bits, et 1179 pour une largeur binaire des données égales à 16 bits. D'autre part, si nous comparons les résultats des tableaux 4.3 et 4.5, nous constatons, que pour une FFT avec une largeur binaire égale à 16 bits des données et 10 bits pour les coefficients, utilise moins des ressources matérielles par rapport à la FFT qui utilise une largeur binaire égale à 16 bits pour les données et 10 bits pour les coefficients.

Tableau 4.4: Évaluation des ressources matérielles (N=16, wl_x=10,wl_W=10)

Utilisation logique	Utilisé	Disponible
Nombre de Slices Flip Flops	730	49152
Nombre de Slices	884	24576
Nombre de LUT à 4 entrées	1744	49152
Nombre de IOBs	222	640
GCLKs	1	32
Nombre de DSP48 18×18 bits	9	512

Tableau 4.5: Évaluation des ressources matérielles (N=16, wl_x=16,wl_W=10)

Utilisation logique	Utilisé	Disponible
Nombre de Slices Flip Flops	946	49152
Nombre de Slices	1179	24576
Nombre de LUT à 4 entrées	2266	49152
Nombre de IOBs	288	640
GCLKs	1	32
Nombre de DSP48 18×18 bits	9	512

IV.4.2 Représentation schématique au niveau RTL de Processeur FFT R4MDC

Le bloc FFT R4MDC avec une longueur limitée à 16 points a été simulé et synthétisé en utilisant le logiciel Xilinx Design ISE. Le bloc technologique RTL ainsi du circuit complet généré par Xilinx est représenté en figure 4.8. Nous distinguerons toutes les entrées et sorties globales du système complet.

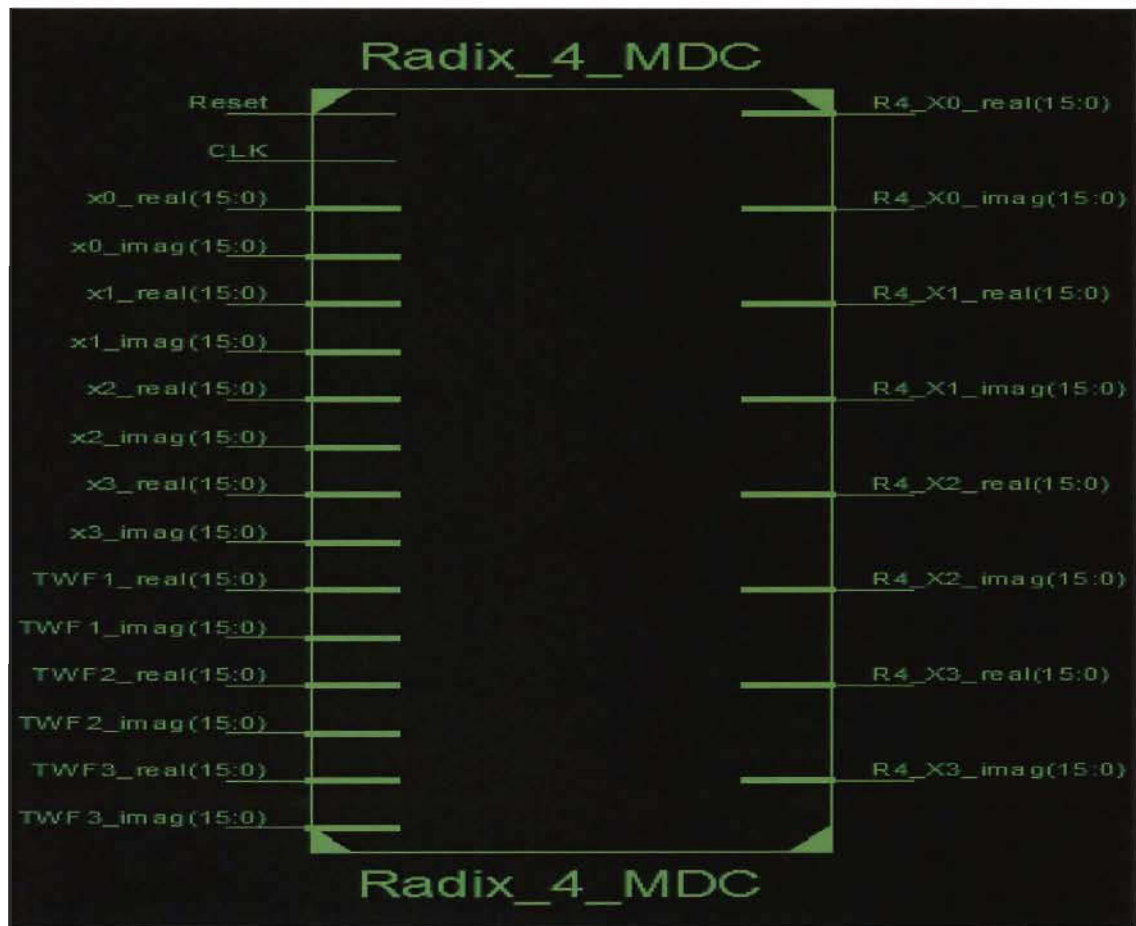


Figure 4.8: Bloc technologique RTL complet du R4MDC à 16 points

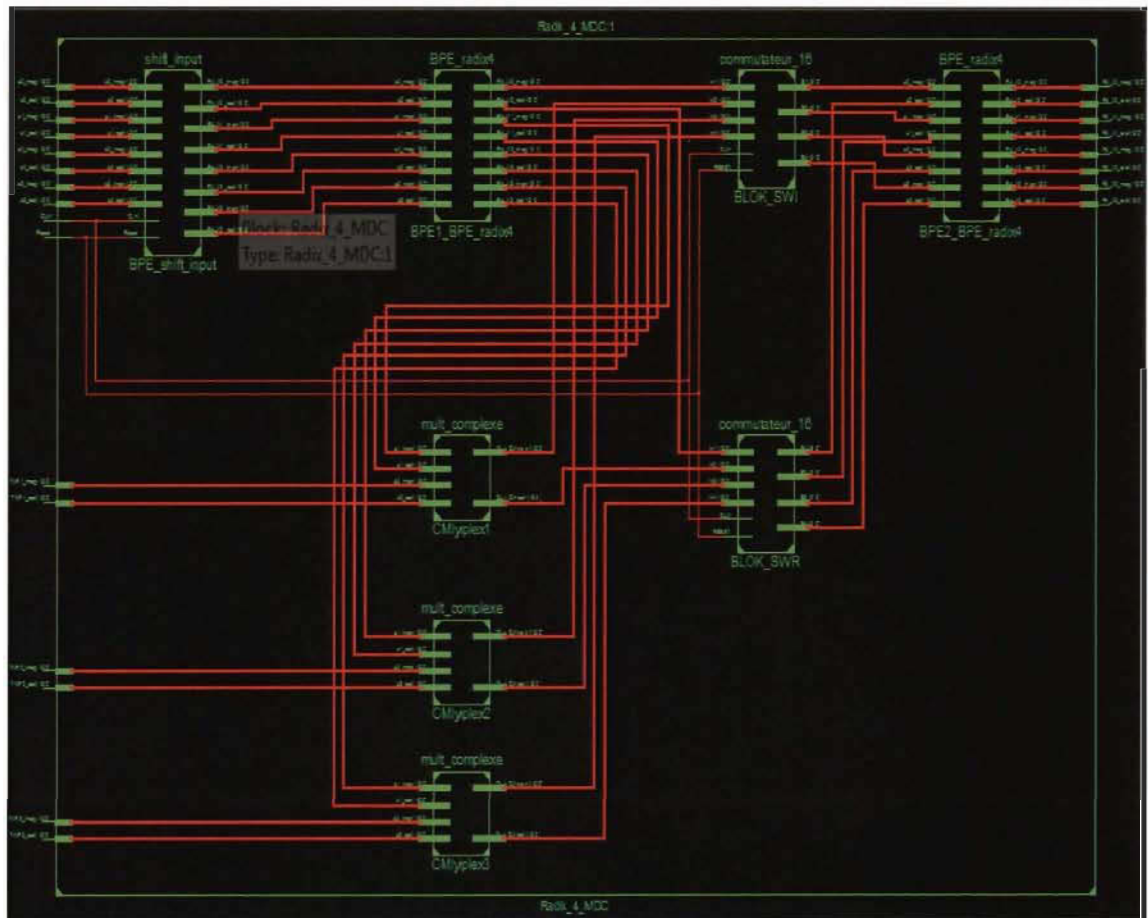


Figure 4.9: Schéma technologique complet du R4MDC à 16 points

Après avoir synthétisé et ensuite passé l'étape de placement et routage, nous aurions le design intérieur de circuit implémenté présenté dans la figure 4.10. Dans cette figure, nous constatons toutes les interconnexions des primitives constituant le processeur FFT dans la puce FPGA ciblée avant.

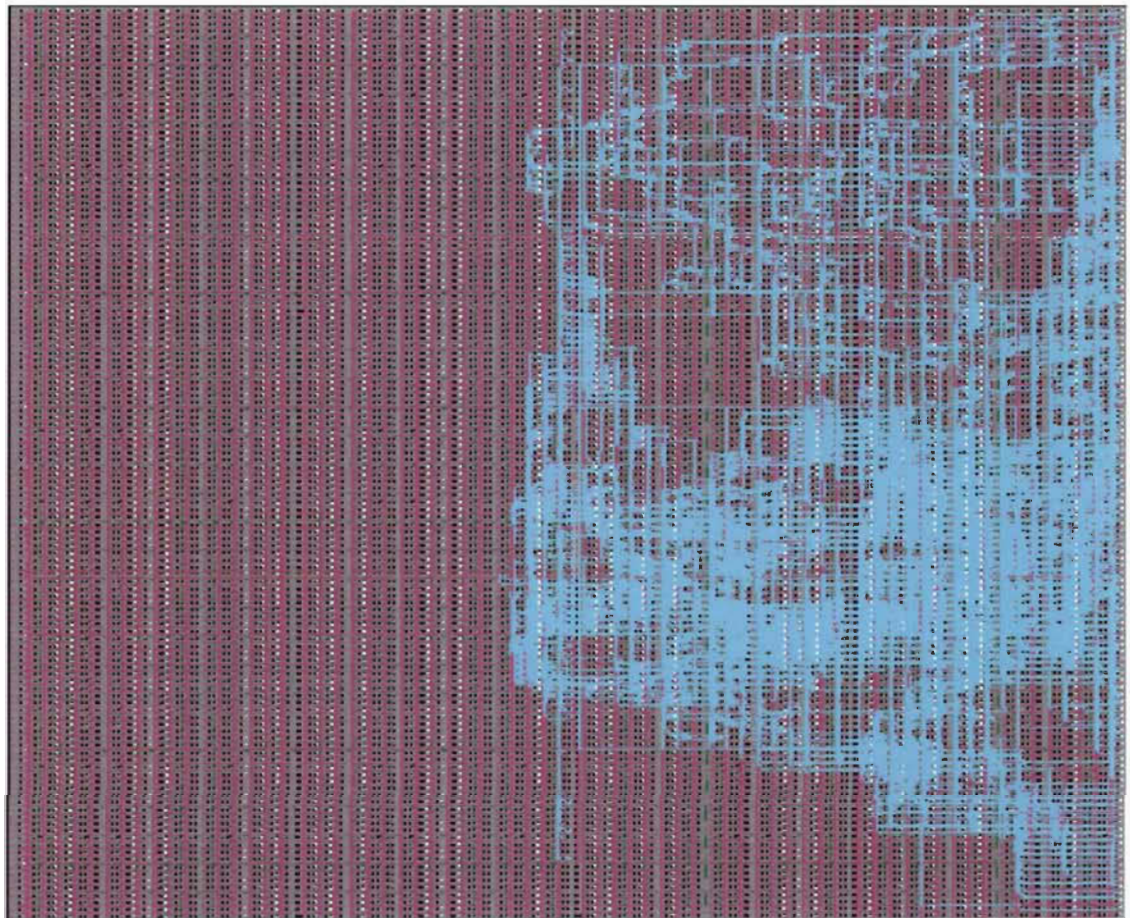


Figure 4.10: Schéma de design interne de processeur FFT R4MDC sur FPGA

IV.5. ANALYSE DE RÉSULTATS DE CONSOMMATION EN PUISSANCE DE RADIX-4 MDC À 16 POINT

La consommation en puissance est caractérisée grâce à l'outil Xpower Analyser de Xilinx après synthèse sur un FPGA virtex-4. Ainsi il est possible d'obtenir les statistiques de l'activité de circuit et calculer au fur et à mesure la consommation d'énergie de circuit. Seule la puissance dynamique est considérée car la puissance statique dépend uniquement de la catégorie de composant FPGA utilisé. Les résultats sont obtenus à une série de

CHAPITRE IV- IMPLÉMENTATION D'UNE FFT RADIX-4 MDC À 16 POINTS

simulations pour différentes largeurs binaires (8, 10, 12 et 16 bits). Le tableau 4.2 illustre les résultats de consommation de puissance.

Tableau 4.6 : Évaluation de la consommation en puissance obtenus avec les coefficients tronqués

largeur de données et TWF	Puissance Dynamique										Moy (mW)	Écart type
	P(X ₁)	P(X ₂)	P(X ₃)	P(X ₄)	P(X ₅)	P(X ₆)	P(X ₇)	P(X ₈)	P(X ₉)	P(X ₁₀)		
8	0.204	0.194	0.207	0.205	0.200	0.195	0.199	0.190	0.202	0.198	0.1994	0.00533
10	0.255	0.239	0.252	0.258	0.257	0.248	0.253	0.253	0.254	0.250	0.2519	0.00552
12	0.331	0.309	0.339	0.332	0.336	0.323	0.327	0.311	0.334	0.332	0.3274	0.0101
16	0.445	0.423	0.460	0.455	0.442	0.440	0.437	0.418	0.449	0.442	0.441	0.0136

D'après le tableau 4.6 nous constatons que la puissance consommée augmente lorsqu'on augmente la taille des données et des TWF; elle est proportionnelle avec la taille des données et des TWF. D'autre part, la puissance consommée change avec le changement des entrées, ce changement est dû à l'activité de commutation.

CHAPITRE V

CONCLUSION

De nos jours, les applications de télécommunication telles que les nouvelles normes de la quatrième génération sans fil doivent répondre à des exigences grandissantes en termes de débit de transmission. Les systèmes OFDM sont une solution clef pour les communications à très hauts débits de transmission. Son principe, qui consiste à utiliser un grand nombre de sous-porteuses orthogonales permet une exploitation efficace du spectre. De plus, l'implémentation de cette modulation multi-porteuse est aujourd'hui faisable en numérique grâce à l'introduction de la structure de la transformée rapide de Fourier inverse (IFFT) pour la génération des sous-porteuses. D'un autre côté, le frein majeur à l'intégration des modules FFT est lié à sa grande complexité de calcul et sa surface d'implémentation non négligeable occasionnant une consommation en puissance. En outre, l'exécution des

CHAPITRE V- CONCLUSION

circuits dédiés provoque des limitations dans le calcul arithmétique du fait de la représentation des données en virgules fixe, ce qui est nécessaire pour satisfaire les contraintes de coût et de consommation.

Depuis que la IFFT/FFT est implémentée, l'optimisation des largeurs de ces coefficients est dotée nécessaire en utilisant des algorithmes spécifiques afin de concevoir un processeur FFT moins complexe, et qui peut répondre aux besoins des normes de télécommunication actuels.

Dans cette thèse, une nouvelle méthodologie en virgule fixe basée sur les algorithmes génétiques sous des contraintes de précision a été présentée. La FFT optimisée est le radix-4 MDC. Un processus itératif d'optimisation de la largeur des coefficients et la synthèse d'architecture est mis en œuvre. De plus, l'évaluation de la précision est réalisée par une approche analytique basée sur le SQNR. Les résultats de quantification en virgule fixe pour différentes longueurs de la FFT (16,64, 256 points), montrent l'intérêt d'une telle approche pour optimiser les coefficients de la FFT afin de minimiser les erreurs de quantification dues à l'opération d'arrondi et de troncature. Dans tous les cas, nous avons obtenu un SQNR maximal par rapport au SQNR référence. Cela signifie qu'il est possible de représenter les coefficients avec moins de bits pour un SQNR maximal et nous aurions à la fin une FFT moins complexe. La deuxième partie de ce travail se repose sur l'implémentation de la FFT radix-4 MDC sur FPGA avec une longueur limitée à 16 points. Les ressources matérielles propres utilisées ont été calculées, et la consommation en puissance a été estimée en utilisant les outils de Xilinx-ISE.

CHAPITRE V- CONCLUSION

Les résultats obtenus, ce n'est qu'un début pour faire d'autres expérimentations pour plus de validation et d'explorer d'autres perspectives. En effet, on considère, le problème est bien clarifié, et au début de sa résolution, et par conséquent certains sous problèmes de la FFT peuvent être approchés et résolus autrement. C'est ce qui va constituer l'objet d'autres travaux de maîtrise et doctorat au laboratoire LSSI dans cet axe.

BIBLIOGRAPHIES

BIBLIOGRAPHIES

- [ALA87] Alard. M and R. Lassalle, «Principles of modulation and channel coding for digital broadcasting for mobile receivers», EBU Tech, Rev, 224, pp.168-190, august 1987.
- [BAA99] Baas. B. M, «A low-Power, high-performance, 1024-point FFT processor», IEEE Journal of Solid-State Circuits, Vol 34(3), pp. 380–387, Mars 1999.
- [BOU05] Boullis. N, A. Tisserand, «Some optimizations of hardware multiplication by constant matrices», IEEE Transactions on Computers, Vol. 54, No. 10, pp. 1271–1282, October 2005.
- [BEA93] Beasley. D, D. R. Bull and R. R. Martin, «An Overview of Genetic Algorithms: Part 1, Fundamentals», University Computing, vol. 15, No. 2, pp. 58-59, 1993.
- [BOI97] Boite. R, Hasler. M, Dedieu .H, «Effet Non Linéaire Dans les filtres Numériques», 1ère édition, Presses polytechnique et universitaires romandes et CENT-ENST, Lausanne, 1997.

BIBLIOGRAPHIES

- [BAK87] Baker. J, «Reducing bias and inefficiency in the selection algorithm». In Proc. of Intl. Conf. on Genetic Algorithms (ICGA'87), 1987.
- [CHA66] Chang. R, «Synthesis of band-limited orthogonal signals for multichannel data transmission», The Bell Systems Technical Journal, pp. 1775–1796, December 1966.
- [CIM85] Cimini Jr. L, «Analysis and Simulation of a Digital Mobile Channel Using Orthogonal Frequency Division Multiplexing», IEEE Transactions on Communications, vol 7, pp. 665-675, July 1985.
- [CHN04] H. Chung-Ping, C. Sau-Gee and C. Kun-Lung, «Design of an efficient variable-length FFT processor», In Circuits and Systems IEEE ISCAS, Vol. 2, pp. 833-836, May 2004.
- [CHA95] Chandrakasan. A. P. and Brodersen. R. W, «Minimizing power consumption in digital CMOS circuits», Proceedings of the IEEE, vol. 83, pp. 498–523, Apr. 1995.
- [COO65] Cooley, J. W, Tukey, J. W. «An algorithm for the machine calculation of complex Fourier series». Mathematics of computation, vol. 19, pp. 297-30. April 1965.
- [DSP05] DSP: Designing for Optimal Results High-Performance DSP Using Virtex-4 FPGAs, Xilinx, March 2005.

BIBLIOGRAPHIES

- [DOE57] Doeltz. M, Heald. E, Martin. D, «Binary data transmission techniques for linear systems», Proceedings Institute of Radio Engineers (IRE), vol. 45, pp. 656–661, May 1957.
- [DAH11] E. Dahlman, S. Parkvall, J. Sköld, «4G LTE/LTE-Advanced for Mobile Broadband: LTE/LTE-Advanced for Mobile Broadband », Academic Press, pp.27-35, 2011.
- [DEJ75] De Jong. K., «The analysis of the behaviour of class of genetic adaptative systems», thèse de doctorat, Department of computer Science, University of Michigan, Ann Arbor, Michigan, 1975.
- [FOU03] Fourier .J, «Théorie Analytique de la Chaleur/The Analytical Theory of Heat» translated by Alexander Freeman. Dover Publications, 1822, translated 1878, re-released 2003.
- [FAZ93] Faze. I, Papke. K. L, “On the Performance of Convolutionally Coded CDMA/ OFDM for Mobile Communication Systems», Proc. Of IEEE PIMRC 93, pp. 468–47, september. 1993.
- [FU09] Fu.B, Ampadu.P, «An Area Efficient FFT/IFFT Processor for MIMO-OFDM WLAN 802.11n», Journal of Signal Processing System, Springer, Vol. 56, pp 59-58, september 2009.
- [GOL89] Goldberg. D.E, «Genetic Algorithms in Search, Optimization, and Machine Learning», Addison-Wesley Publishing company, 1989.

BIBLIOGRAPHIES

- [GOL88] Goldberg. D.E, Holland. J. H, «Genetic Algorithms and Machine Learning», Machine Learning 3, pp.95-99, 1988.
- [GEN00] Gen. M, Cheng. R, «Genetic algorithms and engineering optimization», New York: J.Wiley, and Sons, c2000.
- [GAR99] Garcia .A, Burleson. W, Jean-Luc. D, «Power modelling in Field Programmable Gate Arrays (FPGA) », International Workshop on Field Programmable Logic and Applications, Vol. 1673, pp. 396-404, Springer Berlin Heidelberg 1999.
- [HWA79] Hwang. K, «Computer Arithmetic, Principles, Architecture, and Design», New York: Wiley, 1979.
- [HOL75] Holland, J.H, «Adaptation in natural and artificial systems», Université de Michigan, Ann Arbor, Tech.Rep. 1975.
- [HAN06] Han. K, Evans. B. L, «Optimum word length search using sensitivity information», EURASIP Journal on Applied Signal Processing 5, pp.1-14, (2006).
- [JAB09] Jaber. M. A, Massicotte. D, «A New FFT Concept for Efficient VLSI Implementation: Part-II Parallel Pipelined Processing», Digital Signal Processing, 16th International Conference on 10.1109/ICDSP, pp.1-5, Aout 2009.
- [JUN03] Jung. Y, Yoon. H, Kim. J, «new efficient FFT algorithm pipeline implementation results ofdm/dmt applications», IEEE Transaction on Consumer Electronics, Vol. 49, No. 1, pp.14-20, February 2003.

BIBLIOGRAPHIES

- [KUN91] Kunt. M, Bellanger. M, De-Coulon. F, Guegen. C, «Techniques modernes de traitement numérique des signaux», volume 1 of Traitement de l'information. Presses polytechniques et universitaires romandes et CNET-ENST, 1ère édition, 1991.
- [LER09] Leresteux. J, Lory.J-M., le Jacues. O, «FFT Parallelization for OFDM Systems», Aalborg University, Aalborg East, 9 Sep 2009.
- [MOL11] Molisch Andreas. F, «Wireless Communications». Second Edition, John Wiley 2011.
- [MOR55] Morice. E, «les méthodes de l'analyse de la variance», Revue de statistique appliquée, tome 3, No. 2, pp.65-82. 1955.
- [OPP75] Oppenheim, A. V, Schafer. R. W, «Digital Signal Processing», Prentice-Hall, 1975.
- [PRO96] Proakis. J. G, Manolakis. D. G, «Digital Signal Processing: Principles, algorithms, and application», (3th ed).Upper Saddle River, N.J: Prentice Hall, 1996
- [PRA04] Prasad.R, «OFDM for Wireless Communications Systems», (Artech House, Boston, 2004).
- [ROL03] Palicot. J, Roland.C, «FFT: a Basic Function for a Reconfigurable Receiver», 10th International Conference on Telecommunications, ICT'03, vol. 1, pp. 898–902, March 2003.

BIBLIOGRAPHIES

- [RAO10] Rao. K.R, Kim, D.N, Hwang J.J, «Fast Fourier Transform: Algorithms and Applications», Signals and Communication Technology, Springer Science Business Media B.V. 2010.
- [REN95] Renders. J.M, «Algorithmes génétiques et réseaux de neurones», Paris: Hermès, 1995.
- [REI97] Reineix. A, Eclercy.D, Jecko.B, « FDTD/genetic algoritm coupling for antennas optimization», Annales de Télécommunications, vol. 52, pp. 9-10, 1997.
- [SHO98] Shousheng. H, Torkelson.M, «Designing pipeline FFT processor for OFDM (de)modulation» Proc. IEEE URSI. Int. Symp. Sig. Syst. Electron, pp. 257–262, october 1998.
- [SAN05] Sansaloni. T, Perez-Pascual. A, Torres. V, Valls. J, «Efficient pipeline FFT processors for WLAN MIMO OFDM systems», Electronics Letters, vol. 41, pp. No. 19 pp 1043-1044, 2005.
- [SUL04] Sulaiman. N, Arslan. T, « A genetic algorithm for the optimisation of a reconfigurable pipelined FFT processor» Evolvable Hardware, Proceedings, NASA/DoD Conference on, pp. 104-108, june 2004.
- [SUL05] Sulaiman. N, Arslan. T, «A multi-objective genetic algorithm for on-chip real-time optimisation of word length and power consumption in a pipelined FFT processor targeting a MC-CDMA receiver», Evolvable Hardware, Proceedings, NASA/DoD Conference on, pp. 154–159, june 2005.

BIBLIOGRAPHIES

- [SUL06] Sulaiman. N, Erdogan. A.T, «A multi-objective genetic algorithm for on-chip real time adaptation of a multi-carrier based telecommunications receiver» In: Proceedings of the 1st NASA/ESA Conference on Adaptive Hardware and Systems, pp. 424–427, June 2006.
- [SZW11] Szwarc. M, Czyzewski. A, «New approach to railway noise modeling employing Genetic Algorithms». *Appl Acoust*, Vol. 72, pp. 611–622, 2011.
- [SWA84] Swartzlander. E. E, Young. W. K.W, and Joseph, S. J, «A radix-4 delay commutator for fast Fourier transform processor implementation», *IEEE J. Solid-state Circuits*, vol. 19, No. 5, pp. 702-709, Oct. 1984.
- [SAL67] Saltzberg. B, «Performance of an Efficient Parallel Data Transaction System», *IEEE Transactions on Communication Technology*, Vol. COM-15, pp. 805-813, December. 1967.
- [TUK65] Cooley. J. W, Tukey. J. W, «An algorithm for the machine calculation of complexe Fourier series», *Mathematics Computation*, vol. 19, pp. 297–301, April 1965.
- [TSA11] Tsai. P-Y, Chen. C-W, Huang. M-Y, «Automatic IP Generation of FFT/IFFT Processors with Word-Length Optimization for MIMO-OFDM Systems», *Hindawi Publishing Corporation EURASIP Journal on Advances in Signal Processing*, Vol. 2011, Article ID 136319, 15 pages, November 2010
- [TAI09] Tai. A. H, «Application des techniques multi-porteuses de type OFDM pour les futurs systèmes de télécommunications par satellite», *Institut National*

BIBLIOGRAPHIES

Polytechnique de Toulouse, Thèse Doctorat, Université de Toulouse, France, Mars 2009.

- [WAN99] Wanhammar. L, «DSP Integrated Circuits», Academic, New York, 1999
- [WEI71] Weinstein. S. B, Ebert. P. M, «Data transmission by frequency-division multiplexing using the discrete Fourier transform», IEEE Transactions on Communication Technology, Vol. COM-19, No 5, pp. 628-634, Oct. 1971.
- [YAY11] Ali-Yahiya. T, «Understanding LTE and its Performance», Springer New York Dordrecht Heidelberg London, pp. 3-14, Jan. 2011
- [YEE93] Yee. N, Linnatz. J-P, Fettweis. G, «Multi-carrier CDMA in indoor wireless radio networks” Proc. of IEEE PIMRC 93, pp. 109–113, september. 1993.
- [ZER89] Zervos. N, Kalet. I, «Optimized Decision Feedback Equalization Versus Optimized Orthogonal Frequency Division Multiplexing For High-Speed Data Transmission Over The Local Cable Network», Proceeding of IEEE Transactions conference on Communications (ICC), pp. 1080-1085, june 1989.
- [ZHA90] Zhang. N, Zhang. X, Zhang. C, «Two kinds of array architecture for FFT processor», Proc. Int. Conf. Signal Processing, pp.1161–1162, 1990.