

Direction des bibliothèques

AVIS

Ce document a été numérisé par la Division de la gestion des documents et des archives de l'Université de Montréal.

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

This document was digitized by the Records Management & Archives Division of Université de Montréal.

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal

Modélisation et réalisation de la couche physique du système de
communication numérique sans fil, WiMax, sur du matériel reconfigurable

par

Mazen Ezzeddine

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Mars, 2009

© Mazen Ezzeddine, 2009



Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

Modélisation et réalisation de la couche physique du système de
communication numérique sans fil, WiMax, sur du matériel reconfigurable

présenté par :
Mazen Ezzeddine

a été évalué par un jury composé des personnes suivantes :

Stefan Monnier, président-rapporteur
El Mostapha Aboulhamid, directeur de recherche
Gabriela Nicolescu, membre du jury

Résumé

La technologie de communication numérique sans fil WiMax présente plusieurs avantages par rapport aux systèmes de communication filaires et sans fils traditionnels. Parmi ces avantages nous identifions : une réduction du coût d'installation et du service, une plus grande couverture et débit, une meilleure qualité de service et efficacité spectrale. Ces avantages exceptionnels sont le résultat de l'intégration de plusieurs techniques avancées à la norme de WiMax. Une de ces techniques est la correction d'erreur directe (forward error correction, FEC). En effet, FEC est l'une des fonctions, les plus sophistiquées et les plus exigeantes en termes de ressources et de calcul, spécifiées au niveau de la couche physique de WiMax. La technique FEC, qui permet au récepteur de détecter et de corriger les erreurs de transmission, inclut l'encodeur Reed-Solomon et l'encodeur convolutionnel ainsi que leurs modules de décodage correspondants (décodeur de Viterbi et décodeur Reed-Solomon). Ainsi, pour montrer l'importance et l'avantage de l'introduction de la technique FEC en dépit de sa complexité et de son coût de mise en œuvre, le premier objectif de ce mémoire est d'étudier l'effet de cette technique sur la performance globale de la couche physique du WiMax. Un modèle de la couche physique de WiMax développé en Simulink, y compris tous les modules obligatoires spécifiés dans la norme 802.16d, sera montré. D'abord, le modèle sera simulé avec tous les modules obligatoires spécifiés dans la norme 802.16d et la performance de tous les profils de modulation et de codage correspondants sera montrée et discutée. Ensuite, le modèle sera simulé sans les modules de correction d'erreur directe et l'effet de la technique FEC sera ainsi présenté et discuté. Et finalement, l'effet de l'entrelaceur, qui améliore la performance de la technique FEC, sera aussi étudié. Les courbes BER vs. SNR seront utilisées pour comparer les résultats.

D'autre part, un autre objectif de ce travail de recherche est de montrer que les normes de communication sans fil récentes, telles que 802.15 (réseau personnel sans fil), 802.20 (accès sans fil mobile et à large bande) et 802.22 (réseau régionale sans fil) et en particulier, 802.16 (WiMax) peuvent être efficacement conçues, implémentées et réalisées sur du

matériel reconfigurable plutôt que d'utiliser les circuits intégrés dédiés (ASICs). De plus, dans ce mémoire, nous allons explorer une approche de haut niveau pour l'implémentation des systèmes de communication numérique sans fil à large bande sur une architecture reconfigurable. Pour cela, nous allons présenter une implémentation efficace et pipelinée de tous les modules de la couche physique de la norme 802.16d (WiMax) sur un FPGA en utilisant la plateforme de prototypage rapide Xilinx System Generator for DSP (XSG) qui est une partie de la plateforme de développement de Simulink. Un système émetteur-récepteur à bande de base sera montré. Chaque module implémenté sera testé et comparé aux résultats obtenus avec le modèle Simulink standard. De plus, l'espace occupé par chaque module ainsi que sa vitesse de fonctionnement (fréquence de l'horloge) seront présentés et discutés.

Mots-clés : WiMax, FPGA, matériel reconfigurable, Simulink, Xilinx System Generator for DSP, correction d'erreur directe.

Abstract

The digital wireless communication technology WiMax introduces several advantages compared to traditional wired and wireless communication technologies. Among these benefits, we identify: a reduced installation and service cost, a higher coverage and data rate, a better quality of service and spectral efficiency. These exceptional advantages are the result of the integration of several advanced techniques into the WiMax standard. One of these techniques is the forward error correction (FEC). FEC is one of the most sophisticated, resource demanded and computationally intensive functions specified at the physical layer of WiMax. The FEC technique, which allows the receiver to detect and correct transmission errors, includes the Reed-Solomon encoder and the convolutional encoder as well as their corresponding decoding modules at the receiver (Viterbi decoder and Reed-Solomon decoder). Therefore, to demonstrate the importance and the advantage of introducing the FEC technique despite of its implementation complexity and cost, the first objective of this thesis is to study the effect of the FEC technique on the overall performance of the physical layer of WiMax. An end-to-end Simulink WiMax model, including all the mandatory modules specified in the standard 802.16d will be shown. First, the model will be simulated with all the obligatory modules specified in the standard and the performance of all the corresponding modulation and coding profiles will be shown and discussed. Next, the model will be simulated without the FEC modules and the effect of the FEC technique will be presented and discussed. Finally, the effect of the interleaving process, which increases the performance of the FEC technique, will be addressed. The curves BER vs. SNR will be used to compare the results.

Moreover, another principal objective of this research is to show that the recent digital wireless communication standards such as 802.15 (Wireless Personal Area Networks), 802.20 (Mobile Broadband Wireless Access) and 802.22 (Wireless Regional Area Networks) and in particular, 802.16 (WiMax) can be efficiently designed, implemented and realized using reconfigurable hardware rather than using application specific integrated circuits (ASIC). In addition, in this thesis we will explore a high level approach for the

implementation of broadband digital wireless communication systems on a reconfigurable architecture. Therefore, we will present an efficient and pipelined implementation of all the obligatory modules specified at the physical layer of WiMax on FPGA using the rapid system prototyping platform “Xilinx System Generator for DSP” (XSG) which is a part of the Simulink development platform. A baseband WiMax transmitter-receiver system will be shown. Each implemented module will be tested and compared against the results obtained with the standard Simulink model. In addition, the space occupied by each module and its operating speed (clock frequency) will be presented and discussed.

Keywords: WiMax, FPGA, reconfigurable hardware, Simulink, Xilinx System Generator for DSP, forward error correction.

Table des matières

RÉSUMÉ	iii
ABSTRACT	v
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xii
LISTE DES ABRÉVIATIONS	xiv
DÉDICACE	xvi
REMERCIEMENTS	xvii
CHAPITRE 1: Introduction	1
1.1 WiMax.....	1
1.2 Matériels reconfigurables	2
1.3 Objectifs et motivations.....	4
1.4 Approche et méthodologie du travail de recherche	8
1.5 Contributions	9
1.6 Organisation du mémoire	10
CHAPITRE 2: Description de la couche physique de WiMax	12
2.1 La couche physique de WiMax	12
2.2 Détails de l'émetteur WiMax	13
2.2.1 Mélangeur	14
2.2.2 La correction d'erreur directe dans WiMax	15
2.2.2.1 Encodeur Reed-Solomon	16
2.2.2.2 Codage convolutif.....	18
2.2.3 Perforateur	19
2.2.4 Entrelaceur	20
2.2.5 Modulateur	22

2.2.6	OFDM	24
2.2.7	Les sous-porteuses pilotes	26
2.2.8	Les sous-porteuses de garde	27
2.2.9	Préfixe cyclique	27
2.2.10	Effet des modules de la chaîne de transmission sur une trame exemple	28
2.3	Détails du récepteur WiMax	31
2.3.1	Égaliseur du canal	32
2.3.1.1	Erreur quadratique minimale LSE	33
2.3.2	Estimation du rapport signal/bruit	34
2.3.3	Décodeur de Viterbi	36
2.4	Résumé du chapitre	37
CHAPITRE 3: Revue de la littérature		33
3.1	Université Guelph	33
3.2	L'académie chinoise des sciences	34
3.3	Institut national d'astrophysique, d'optique et d'électronique	34
3.4	La société ALTERA	35
3.5	Lattice Semiconductor	36
3.6	Innovative radio Systems	36
3.7	Université de Kansas	36
3.8	Contributions par rapport aux travaux précédemment cités	37
3.10	Performance et efficacité de l'encodeur RS et l'encodeur convolutionnel	38
CHAPITRE 4: Technologie de réalisation		47
4.1	Introduction	47
4.2	Les contraintes de l'implémentation matérielle	42
4.3	Généralités sur les FPGAs	48
4.4	Avantages de l'utilisation des FPGAs	49

4.5	Choix de la technologie d'implémentation	50
4.6	La famille des FPGAs Virtex	50
4.7	Xilinx System Generator for DSP (XSG)	52
4.7.1	Fonctions disponibles dans XSG.....	53
4.7.2	Concept de l'horloge dans XSG	55
4.7.3	Simulation des modèles dans XSG	55
4.7.4	Précision du calcul dans XSG	51
4.8	Le flux de conception VLSI traditionnel	56
4.9	Avantages et flux de conception de XSG	57
4.10	Résumé du chapitre	59
CHAPITRE 5: Résultats de simulation du modèle de la couche physique de WiMax		61
5.1	Introduction	61
5.2	Description générale du modèle implémenté	62
5.3	Environnement de la simulation	64
5.4	Les courbes BER vs SNR pour les différents profils de modulation et de codage ..	64
5.5	Effet de la technique de correction d'erreur directe (FEC)	66
5.6	Effet de l'entrelaceur	68
5.7	Analyse générale des résultats de simulation	69
5.8	Résumé du chapitre	70
CHAPITRE 6: Implémentation matérielle de la couche physique de WiMax		72
6.1	Introduction	72
6.2	Détails de l'implémentation des modules de l'émetteur	73
6.2.1	Mélangeur	73
6.2.2	Encodeur Reed-Solomon	75

6.2.3	Encodeur Convolutionnel	77
6.2.4	Entrelaceur	78
6.2.5	Modulateur	80
6.2.6	Insertion des symboles pilotes et bandes de garde	81
6.2.7	La transformée inverse de Fourier	82
6.2.8	Préfixe cyclique (CP)	83
6.3	Détails de l'implémentation des modules du récepteur	84
6.4	Analyse générale des résultats	87
6.5	Comparaison	89
6.6	Résumé du chapitre	90
	CHAPITRE 7: Conclusion et travaux Futurs	91
7.1	Résumé du travail accompli	91
7.2	Liste des travaux futurs suggérés	92
7.2.1	Longueur de préfixe cyclique adaptif dans WiMax	93
7.2.2	Synchronisation entre l'émetteur et le récepteur	93
7.2.3	Évaluation de WiMax avec les profils 256-QAM et 512-QAM	93
7.2.4	Le rapport de la puissance moyenne-maximale (PAPR)	94
7.2.5	Entrées multiples sorties multiples MIMO.....	94
7.2.6	Partitionnement matérielle/logicielle de l'implémentation	95
	Bibliographie	96

Liste des tableaux

TABLEAU 2.1 : Perforation en fonction de taux de transfert	15
TABLEAU 2.2 : Les sept profils de modulation et de codage adoptés par WiMax.....	24
TABLEAU 2.3 : Les paramètres de l'OFDM dans WiMax.	26
TABLEAU 2.4 : Indices des sous-porteuses pilotes et leurs valeurs correspondantes.....	26
TABLEAU 3.1 : Résultats de l'université Guelph vs. INAOE	39
TABLEAU 4.1 : Statistique de la plateforme Virtex II pro XC2VP30	52
TABLEAU 6.1 : Ressources consommées et période minimale du mélangeur	74
TABLEAU 6.2 : Ressources consommées et période minimale de l'encodeur RS.....	77
TABLEAU 6.3 : Ressources utilisées par l'encodeur convolutionnel et le perforateur.	77
TABLEAU 6.4 : Ressources utilisées et période minimale de l'entrelaceur	79
TABLEAU 6.5 : Ressources utilisées et période minimale du modulateur	80
TABLEAU 6.6 : Ressources utilisées et période minimale de l'insertion des symboles pilotes et gardes	82
TABLEAU 6.7 : Ressources utilisées et période minimale de l'IFFT	82
TABLEAU 6.8 : Ressources consommées par les modules du récepteur	86
TABLEAU 6.9 : Ressources consommées et période minimale des modules des chaînes de transmission et de réception.....	88
TABLEAU 6.10 : Comparaison entre ce travail et le travail de Garcia	90

Liste des figures

FIGURE 1.1 : Approche de recherche	8
FIGURE 2.1 : Schéma d'un émetteur WiMax à bande de base	14
FIGURE 2.2 : Architecture du mélangeur	15
FIGURE 2.3 : Encodeur Reed-Solomon	17
FIGURE 2.4 : Architecture de l'encodeur Reed-Solomon	18
FIGURE 2.5 : Architecture du Convolueur	19
FIGURE 2.6 : RS, CC, entrelaceur et modulateur BPSK implémenté en Simulink	22
FIGURE 2.7 : Points de constellations BPSK, QPSK, 16QAM et 64QAM dans WiMax.....	23
FIGURE 2.8 : Les sous-porteuses de données, gardes et pilotes dans un symbole OFDM	25
FIGURE 2.9 : Émetteur OFDM dans Simulink	28
FIGURE 2.10 : Erreur quadratique minimale dans Simulink	34
FIGURE 2.11 : EVM dans la constellation I-Q	35
FIGURE 2.11 : Estimation du rapport/signal bruit dans Simulink	36
FIGURE 2.12 : Démodulateur, déentrelaceur, décodeur de Viterbi et décodeur Reed Solomon en Simulink	37
FIGURE 3.1: Performance des codes : CC, RS(15,11) et RS (15,9) dans un système à base OFDM.....	44
FIGURE 3.2 : Performance de RS et CC dans les systèmes UWB	45
FIGURE 3.3 : Performance de l'encodeur RS par rapport à la combinaison de RS et CC	46
FIGURE 4.1 : Architecture générale des FPGAs.	49
FIGURE 4.2 : Architecture générale du FPGA VIRTEX	51
FIGURE 4.3 : Xilinx System Generator for DSP	54
FIGURE 4.4 : Flux de conception VLSI conventionnel	57
FIGURE 4.5 : Flux de conception VLSI avec XSG	58
FIGURE 5.1 : Schéma bloc de la couche physique de WiMax implémenté en Simulink	63
FIGURE 5.2 : BER vs SNR pour les différents profils de modulation et de codage sur un canal AWGN.....	64
FIGURE 5.3 : BER vs SNR pour les différents profils de modulation et de codage sur un canal sélectif en fréquence	65

FIGURE 5.4 : Effet de la technique FEC dans QPSK1/2 sur un canal sélectif en fréquence ..	67
FIGURE 5.5 : Effet de la technique FEC dans 16QAM1/2 sur un canal sélectif en fréquence	67
FIGURE 5.6 : Effet de la technique FEC dans 64QAM2/3 sur un canal sélectif en fréquence	68
FIGURE 5.7 : Effet de l'entrelaceur dans 16QAM1/2 sur un canal sélectif en fréquence.....	69
FIGURE 5.8 : Effet de l'entrelaceur dans 64QAM3/4 sur un canal sélectif en fréquence.....	69
FIGURE 6.1 : Circuit du mélangeur dans XSG	74
FIGURE 6.2 : Circuit de l'encodeur Reed –Solomon dans XSG.....	76
FIGURE 6.3 : Circuit de l'encodeur convolutionnel dans XSG.....	78
FIGURE 6.4 : Circuit de l'entrelaceur dans XSG	79
FIGURE 6.5 : Circuit du modulateur dans XSG	80
FIGURE 6.6 : Circuit de l'insertion des symboles pilotes et gardes dans XSG	81
FIGURE 6.7 : Le bloc IFFT dans XSG	83
FIGURE 6.8 : Architecture du préfixe cyclique	84
FIGURE 6.9 : Modules de la chaîne de réception implémentés	85

Liste des abréviations

AMC	Adaptive Modulation and Coding
ASIC	Application Specific Integrated Circuit
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BSID	Base Station ID
CC	Codage Convolutif
CDMA	Code Division Multiple Access
CLB	Configurable Logic Block
CMOS	Complementary Metal Oxide Semiconductor
D/UTUC	Downlink/Uplink Interval Usage Code
DLL	Digital Locked Loop
DSL	Digital Subscriber Line
DSP	Digital Signal Processing
EVM	Error Vector Magnitude
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GSM	Global System for Mobile
IFFT	Inverse Fast Fourier Transform
INAOE	National Institute for Astrophysics, Optics and Electronics
IP	Intellectual property
JTAG	Joint Test Action Group
KUAR	Kansas University Agile Radio
LASSO	Laboratoire d'Analyse et de Synthèse des Systèmes Ordines
LOS	Line Of Sight
MAC	Media Access Control
MANET	Mobile Ad hoc Network
MIMO	Multiple Input Multiple Output
OFDM	Orthogonal Frequency Division Multiplexing

OFDMA	Orthogonal Frequency Division Multiple Access
RAM	Random Access Memory
ROM	Read Only Memory
SDR	Software Defined Radio
SRAM	Static Random Access Memory
TDMA	Time Division Multiple Access
UWB	Ultra Wide Band communication
VHDL	(Very High Speed Integrated Circuits) Hardware Description Language
VLSI	Very Large Scale Integration
WiMax	Worldwide Inter-operability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
XSG	XILINX System Generator for DSP
3G	Third Generations of telecommunication standards.

À la mémoire de mon père
Que DIEU l'entoure de sa miséricorde
À ma famille

Remerciements

D'abord et avant tout, je remercie DIEU pour m'avoir accordé la santé, la patience, la connaissance et tout ce qui est nécessaire pour terminer ce mémoire.

Je tiens également à exprimer ma gratitude à mon directeur de recherche, Monsieur ElMostapha Aboulhamid, pour son soutien inconditionnel qu'il m'a apporté tout au long de ce travail, pour son enthousiasme et son dynamisme constant. Merci surtout pour sa patience et sa compréhension surhumaine lors des dernières semaines. Sans ses suggestions et son aide ce mémoire ne pourrait pas être complété.

Je tiens à remercier vivement mes amis et collègues de recherche au sein du laboratoire LASSO surtout Mohamad, Amine et Joe, merci pour leurs encouragements, conseils et longues discussions. Mes vifs remerciements à mon amie Khadija Benadada, du laboratoire de recherche HERON, d'avoir consacré son temps pour lire, réviser et corriger ce mémoire malgré que le mot FPGA n'est qu'un acronyme pour elle.

J'exprime aussi ma profonde reconnaissance à toute ma famille, pour leur soutien sans faille et leur présence. Un remerciement particulier pour ma chère maman pour ses encouragements et ses recommandations pleines de persévérance.

Mon dernier mot sera pour les amis qui sont loin, mais qui sont restés très proches ; parmi eux Radi, Mohamad, Mahdi, Hussein, et Jihad.

Pour finir, je souhaite aussi remercier les professeurs Stefan Monnier et Gabriela Nicolescu d'avoir accepté pour juger ce mémoire.

Chapitre 1 :

Introduction

1.1 WiMax

La technologie de communication sans fil devient de plus en plus demandée même lorsque les communications filaires sont disponibles. Récemment, de nombreuses recherches dans différents secteurs de la communication numérique sans fil ont été menées. Ces recherches ont conduit vers l'évolution de nombreux standards. Nous parlons, en particulier, des réseaux locaux sans fil (WLANs), réseaux personnels sans fil (WPANs), réseaux urbains sans fil (WMANs) et les réseaux régionaux sans fil (WRANs). Les effets de cette révolution dans le monde de la communication sans fil ont été ressentis dans presque tous les secteurs y compris l'éducation, la santé, le commerce, la fabrication et le gouvernement. Certainement, ces technologies vont être la force pivot qui structurera les économies et les sociétés au cours des prochaines années [5].

En effet, les technologies de communication classique telle que 3G, câble/DSL et même WiFi présentent quelques inconvénients. Par exemple, DSL force les utilisateurs à se fixer pour obtenir une connexion internet à haut débit. De plus, WiFi ne prévoit la couverture qu'aux utilisateurs dans un rayon de quelques centaines de mètres. Toutefois, le réseau cellulaire 3G permet aux utilisateurs d'être mobiles, cependant cette technologie a été principalement conçue pour les services voix et ne peut pas supporter la plupart des services multimédias (vidéo-conférence en temps réel, jeux vidéo en ligne etc.). Pour surmonter ces limitations, WiMax a été développé. WiMax est un système de communication numérique sans fil également connu sous le nom d'IEEE 802.16. La technologie WiMax peut fournir l'accès sans fil à large bande jusqu'à 50 kilomètres pour

les stations fixes, et jusqu'à 5 à 15 kilomètres pour les stations mobiles [2]. WiMax a été conçu pour la transmission de services multimédias (voix, internet, courriel, jeux...) à hauts débits (jusqu'à 75 Mbit/s). Cette nouvelle technologie apporte des avantages potentiels au niveau de la couverture, l'installation, la consommation d'énergie, la qualité de service, la réutilisation des fréquences, l'efficacité spectrale, la sécurité et le coût. Comme indiqué dans [3], la technologie WiMax constitue une infrastructure qui, au cours des prochaines années, pourrait remplacer les technologies de communication sans fil classique telles que GSM, TDMA et CDMA.

1.2 Matériels reconfigurables

Les matériels reconfigurables, en particulier les FPGA (field programmable gate array), sont des circuits numériques à logiques programmables c.à.d. pouvant être programmés par le concepteur plutôt que par le fabricant du périphérique pour exécuter une fonction logique. Ces matériels fournissent une plateforme flexible et efficace pour répondre aux exigences des conceptions complexes en termes de vitesse d'opération, d'espace occupé, de consommation d'énergie, du coût de conception et de temps de mise sur le marché. Dans ce contexte, l'implémentation d'une conception sur une architecture reconfigurable (FPGA) fournit une accélération de l'ordre de 10X à 100X par rapport à une implémentation logicielle équivalente (implémentation sur un processeur à usage général) [74]. Ces avantages des architectures reconfigurables sont basés sur le fait que le matériel utilisé est personnalisé pour réaliser un certain algorithme. De plus, ces architectures offrent la possibilité de l'exploitation du parallélisme inhérent qui apparaît dans de nombreux problèmes algorithmiques. Dans ce contexte, en plus de leur capacité de calcul élevé et de leur flexibilité, plusieurs autres raisons ont motivé les ingénieurs à considérer les FPGA comme une technologie principale pour le prototypage et la mise en œuvre des systèmes complexes. Parmi ces raisons :

- L'accélération du temps de mise sur le marché.
- La reprogrammabilité, ce qui permet la mise à jour de la conception ainsi que la réutilisation de la puce.

- La réutilisabilité en termes de propriétés intellectuelles et du code développé.
- Le support à l'évolution des algorithmes et des normes.
- La facilité de correction des erreurs de la conception.
- Les petites équipes de conception.
- La réduction du coût globale de conception.

De plus, comme indiqué dans [74], les puces FPGA les plus récentes atteignent une performance similaire à celle que les circuits électroniques dédiés (ASIC) fournissent dans de nombreuses applications. Pour toutes ces raisons, les architectures reconfigurables se présentent comme une solution remarquable pour l'implémentation des normes de communication sans fil récentes telle que WiMax.

Ainsi, on peut définir le thème principal de ce mémoire qui est l'exploration d'une approche de haut niveau pour l'implémentation de la couche physique de WiMax sur une architecture reconfigurable. Pour cela, nous allons présenter une implémentation efficace et pipelinée de tous les modules de la couche physique de WiMax sur un FPGA en utilisant l'outil de synthèse de haut niveau Xilinx System Generator for DSP (XSG) qui est une partie de la plateforme de développement de Simulink. Par conséquent, nous allons fournir les résultats qui montrent la pertinence de cette approche en fonction de certains paramètres tels que l'espace occupé, la vitesse d'opération et le temps de conception. Cependant, puisque la technologie WiMax est complexe comme en témoigne la norme 802.16 qui est de plus de 900 pages [1][2], la construction d'un modèle de simulation de la couche physique de WiMax était nécessaire pour le déchiffrement des complexités de cette couche avant une implémentation définitive sur une architecture reconfigurable. De plus, en utilisant le modèle de simulation déjà implémenté, nous avons étudié l'effet de la technique de correction d'erreur directe (FEC) sur la performance globale de la couche physique de WiMax. Les objectifs détaillés ainsi que les motifs et les défis de ce travail de recherche seront présentés en détail dans la section suivante.

1.3 Objectifs et Motivations

WiMax, ou le standard IEEE 802.16 pour l'accès sans fil à large bande, est de plus en plus en popularité comme une solution alternative aux technologies de communication filaires et sans fils classiques. En effet, WiMax est la nouvelle tendance qui permet l'accès sans fil à large bande pour les utilisateurs fixes et mobiles, avec des déploiements qui devraient augmenter de trois fois en 2010 [4]. Comme indiqué dans la section 1.1, la technologie WiMax apporte des avantages potentiels au niveau de la couverture, l'installation, la consommation d'énergie, la qualité de service, l'efficacité spectrale et la mobilité. Ces avantages exceptionnels sont le résultat de l'intégration de plusieurs techniques avancées à la norme de WiMax. Parmi ces techniques nous identifions : le multiplexage par division orthogonale de la fréquence (orthogonal frequency division multiplexing, OFDM), l'entrée multiple sortie multiple (multiple input multiple output, MIMO) et la correction d'erreur directe (forward error correction, FEC). Dans ce contexte, la correction d'erreur directe (FEC) est l'une des fonctions, les plus sophistiquées et les plus exigeantes en termes de ressources et de calcul, spécifiées au niveau de la couche physique de WiMax. La technique FEC permet au récepteur de détecter et de corriger les erreurs de transmission et ainsi, cette dernière augmente la capacité du canal de communication. Comme spécifié dans le standard de WiMax [1], le processus de correction d'erreur directe est réalisé en concaténant l'encodeur Reed-Solomon avec l'encodeur convolutionnel, à l'émetteur. Par conséquent, le décodeur de Viterbi qui effectue le décodage d'une séquence codée avec un encodeur convolutionnel ainsi que le décodeur Reed-Solomon doivent être implémentés au récepteur.

De ce fait, pour montrer l'importance et l'avantage de l'introduction de la technique FEC en dépit de sa complexité et de son coût de mise en œuvre, le premier objectif de ce mémoire est d'étudier l'effet de cette technique sur la performance globale de la couche physique de WiMax. Un modèle de la couche physique de WiMax développé en Simulink, y compris tous les modules obligatoires spécifiés dans la norme 802.16d, sera montré. En premier lieu, le modèle sera simulé avec tous les modules obligatoires spécifiés

dans le standard et la performance de tous les profils de modulation et de codage correspondants sera présentée et discutée. Ensuite, le modèle sera simulé sans les modules de correction d'erreur directe et l'effet de la technique FEC sera ainsi montré et discuté. Et finalement, l'effet de l'entrelaceur qui améliore la performance de la technique FEC, sera aussi étudié. Les courbes BER vs. SNR seront utilisées pour comparer les résultats.

D'autre part, un autre objectif principal de ce travail de recherche est de montrer que les normes de communication sans fil récentes telles que 802.15 (réseau personnel sans fil), 802.20 (accès sans fil mobile et à large bande) et 802.22 (réseau régionale sans fil) et en particulier, 802.16 (WiMax) peuvent être efficacement conçues, implémentées et réalisées sur du matériel reconfigurable (FPGA) plutôt que d'utiliser les circuits intégrés dédiés (ASIC) pour la réalisation des tels systèmes. En effet, toutes les normes et les spécifications, y comprises les normes de communication sans fil, sont inutiles si elles ne peuvent pas être ciblées vers une plateforme matérielle. À ce propos, une norme spécifique peut être implémentée sur trois types de circuits: (1) les processeurs à usage général et les processeurs de traitement numérique (digital signal processor), (2) les circuits intégrés dédiés (ASIC) et (3) les matériels reconfigurables, en particulier, les FPGA. D'abord, les processeurs DSP ainsi que les processeurs à usage général ne peuvent pas répondre aux exigences de calcul intensif requises par les normes de communication sans fil récentes. Par exemple, ces processeurs ne sont pas en mesure d'atteindre des performances suffisantes pour décoder, à haute vitesse, les codes complexes comme l'encodeur Reed-Solomon et l'encodeur convolutionnel. Les conceptions ASIC classiques, d'autre part, sont très performantes et elles dominent le marché. Cependant, les conceptions ASIC impliquent des coûts récurrents et un temps de conception très élevé, ce qui exige un investissement initial élevé. De plus, les circuits ASIC souffrent du manque de la flexibilité et de la reprogrammabilité.

Aujourd'hui, de plus en plus, les architectures reconfigurables, en particulier les FPGA, se présentent comme la meilleure solution pour l'implémentation des systèmes de

communication sans fil à large bande tel que WiMax. Ce fait est dicté non seulement par les limitations des circuits ASIC et des processeurs DSP (ainsi que les processeurs à usage générale), mais aussi par les caractéristiques exceptionnelles des ces architectures telles que : la capacité de calcul, la flexibilité, la simplicité et la facilité de l'utilisation, le cycle de développement réduit, la reprogrammabilité et la réutilisabilité [7]. Dans ce contexte, la plupart des nouveaux standards de communication numérique sans fil, comme le WiMax, sont actuellement en cours de développement et de déploiement. Donc, la reprogrammabilité de la plateforme de réalisation matérielle est un facteur très important. Cela permettra la reprogrammation de la puce lorsque la norme évolue. De plus, la conception des systèmes de communication sans fil à large bande, avec des débits de données élevés, est gouvernée par la vitesse de traitement de la plateforme de réalisation matérielle. Par conséquent, cette dernière doit avoir une capacité de traitement importante. D'autre part, le temps de la mise sur le marché influe directement le choix de la plateforme matérielle. Pour cela, les ingénieurs cherchent souvent la facilité d'utilisation des outils de développement ainsi que la disponibilité des propriétés intellectuelles qui permettront de minimiser le cycle de développement et par conséquent, d'accélérer le temps de mise sur le marché. Ainsi, l'exploration d'une approche de haut niveau pour l'implémentation de la couche physique de WiMax sur une architecture reconfigurable est un sujet intéressant qui mérite d'être abordé dans ce mémoire.

Toutefois, plusieurs défis sont associés à la mise en œuvre de la couche physique du WiMax sur une architecture reconfigurable. Le premier défi dans la mise en œuvre de la couche physique de WiMax est la complexité de cette couche. Dans ce contexte, la plupart des ouvrages publiés, qui ont abordé l'implémentation de la couche physique de WiMax sur une architecture reconfigurable, se concentrent sur certaines fonctions et ne traitent pas tous les modules de cette couche. D'autre part, la majorité des fonctions de la couche physique de WiMax (FFT/IFFT, décodeur de Viterbi, encodeur/décodeur Reed-Solomon) sont soumises à des contraintes temps-réel très critiques en termes de débit et de la latence. Par exemple, pour soutenir le débit spécifié dans la norme 802.16, un récepteur WiMax

typique devrait être en mesure de traiter des données à un débit jusqu'à 75 Mbps. Ces contraintes imposent la nécessité d'introduire la plus grande quantité du parallélisme à la conception pour atteindre une vitesse d'opération alignée avec les spécifications du standard. Cependant, l'avantage d'une structure parallèle est généralement accompli au prix de l'augmentation des ressources consommées. Ainsi, un autre défi de ce travail est de maintenir l'équilibre entre la vitesse de fonctionnement requise (fréquence de l'horloge) et les ressources consommées par la conception proposée.

De même, le temps de la mise sur le marché est un défi principal. Dans ce contexte, l'utilisation des langages de description de matériels classiques, comme le VHDL et le VERILOG, pour la mise en œuvre de la couche de WiMax pourrait conduire à un temps de conception et d'implémentation élevé. Pour cela, contrairement à la plupart des ouvrages qui ont abordés l'implémentation de la couche physique de WiMax sur un FPGA, nous allons explorer une approche de haut niveau pour la conception et la réalisation de cette couche. Par conséquent, nous allons fournir les résultats qui montrent la pertinence de notre approche en fonction de certains paramètres tels que l'espace occupé, la vitesse d'opération et le temps de conception.

Un dernier objectif de ce travail de recherche est de fournir une bibliothèque des blocs qui abstraient les détails de l'implémentation des modules obligatoires de la couche physique de WiMax. Ces blocs sont hautement optimisés, pipelinés, synthétisables, réutilisables, efficaces, simples et personnalisables. En outre, les modules implémentés supportent tous les profils de modulation et de codage spécifiés au niveau de la couche physique de WiMax. Par conséquent, ces blocs peuvent être utilisés au niveau commercial et académique pour une implémentation très simple et très rapide de la couche physique de WiMax et plus tard, pour l'intégration de ces modules aux modules de la couche MAC (media access control).

Avant de terminer ce paragraphe, il est important de mentionner que, bien que notre étude se concentre sur la norme 802.16 (WiMax), les conclusions de ce mémoire restent toujours utiles pour d'autres normes de communication numérique sans fil telles que 802.22 (réseau régional sans fil) et 802.20 (accès sans fil mobile et a large bande). Dans ce contexte, la majorité des fonctions de la couche physique de WiMax traitées dans ce travail sont encore définies dans les normes 802.20 et 802.22 [72][73].

1.4 Approche et méthodologie du travail de recherche

L'approche et la méthodologie de recherche utilisées dans ce mémoire sont illustrées dans la figure 1.1. Tout d'abord, ce travail de recherche a commencé par l'étude, la compréhension, et l'analyse détaillée du standard 802.16-2004 (WiMax) avec concentration sur les spécifications de la couche physique. Sûrement, sans cette étude, ni la modélisation, ni la réalisation de la couche physique du WiMax n'auraient été possibles.

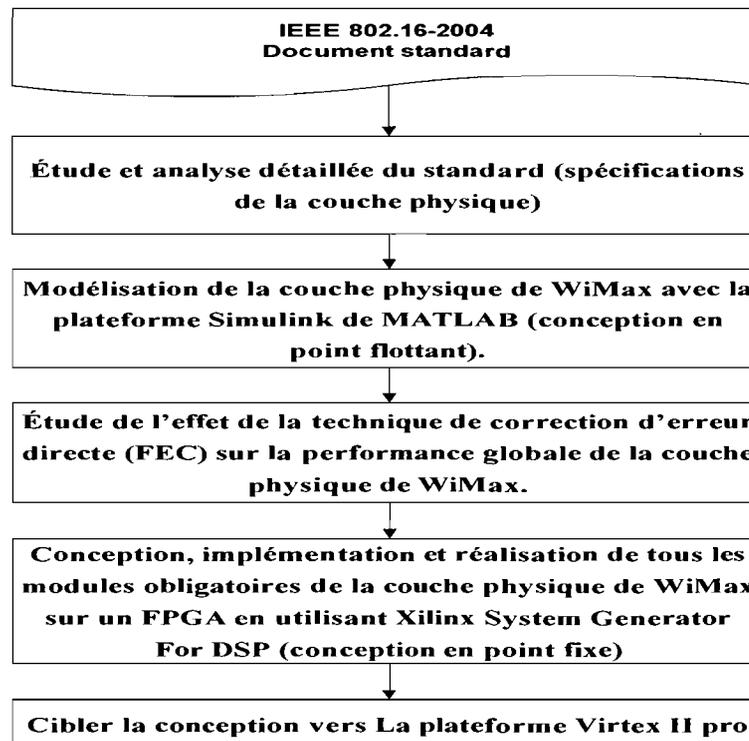


Figure 1.1: Approche de Recherche.

En deuxième lieu, nous avons modélisé la couche physique de WiMax en utilisant les blocksets standards de Simulink (point flottant). En effet, cette modélisation constitue la première étape dans le flux de la conception matérielle avec Xilinx System Generator for DSP (XSG). De plus, comme déjà indiqué, trois objectifs ont été les raisons de cette modélisation : (1) obtenir une compréhension plus précise et plus détaillée du fonctionnement de la couche physique de WiMax (avant une implémentation définitive sur le matériel), (2) présenter et évaluer la performance de tous les profils de modulation et de codage spécifiés au niveau de la couche physique de WiMax et (3) étudier l'effet de la technique de correction d'erreur directe ainsi que la technique d'entrelacement sur la performance globale de cette couche. Ainsi, après l'obtention d'une excellente compréhension de la couche physique de WiMax, nous avons abordé l'implémentation de tous les modules de cette couche sur une architecture reconfigurable en utilisant l'outil de synthèse de haut niveau Xilinx System Generator For DSP (XSG) qui est une partie de la plateforme de développement de Simulink. En outre, le modèle implémenté avec les blocksets standards de Simulink était un modèle de référence pour l'implémentation matérielle de la couche physique de WiMax. Dans ce contexte, chaque module implémenté en XSG (point fixe) a été testé et vérifié contre le même module implémenté en Simulink (point flottant).

1.5 Contributions

Ce travail de recherche est le premier à être mené au sein du laboratoire LASSO au sujet de l'implémentation d'un système de communication numérique sans fil sur du matériel reconfigurable. Dans ce contexte, ce mémoire a apporté les contributions suivantes:

- Développement d'un modèle de la couche physique du système de communication sans fil, WiMax, conformément au standard 802.16d. Ce modèle, développé en Simulink, comporte tous les modules obligatoires spécifiés dans le standard avec une unité de contrôle responsable de la modulation adaptative.

- Étude de l'effet de la technique de correction d'erreur directe (FEC) ainsi que la technique d'entrelacement sur la performance globale de la couche physique de WiMax.
- Exploration d'une approche de haut niveau pour la mise en œuvre de la couche physique de WiMax sur une architecture reconfigurable et présentation des résultats qui montrent la pertinence de cette approche en fonction de certains paramètres tels que l'espace occupé et la vitesse d'opération.
- Fournir une bibliothèque des blocs qui abstraient les détails de l'implémentation de la couche physique de WiMax. Ces blocs, qui sont hautement optimisés, pipelinés, synthétisables, et personnalisables, peuvent être utilisés au niveau académique et commercial pour une implémentation très rapide de la couche physique de WiMax.

1.6 Organisation de ce mémoire

Ce mémoire est organisé comme suit : le chapitre 2 présente un aperçu général sur la couche physique de WiMax. Le chapitre 3 présente les recherches antérieures pertinentes au niveau de l'implémentation et la réalisation des systèmes basés sur le multiplexage par répartition orthogonale de la fréquence (OFDM) sur FPGA. Dans ce chapitre, nous allons aussi présenter brièvement l'efficacité de l'encodeur Reed-Solomon et l'encodeur convolusionnel au niveau de l'amélioration de la performance des systèmes de communication sans fil basés sur OFDM. Dans le chapitre 4, nous abordons le flux de la conception des systèmes numériques en utilisant la plateforme de prototypage rapide Xilinx System Generator for DSP (XSG). Dans ce chapitre, nous abordons également, les FPGAs ainsi que leurs architectures et les éléments principaux qui les caractérisent. Le chapitre 5 présente quelques résultats de simulation du modèle de la couche physique de WiMax implémenté avec Simulink. Le chapitre 6 explique en détail l'implémentation de tous les modules de la couche physique de WiMax sur FPGA en utilisant XSG. Ce chapitre liste les

résultats obtenus et les compare aux travaux précédemment effectués. Le mémoire conclut dans le chapitre 7 avec un résumé du travail accompli ainsi que des suggestions pour les travaux futurs.

Chapitre 2 :

Description de la couche physique de WiMax

Ce chapitre vise à donner un aperçu sur la couche physique de WiMax. Chaque module de la couche physique, spécifié dans le standard 802.16d, sera présenté et l'implémentation de ces modules en Simulink sera abordée. De plus, le rôle de chaque module au niveau de l'amélioration de la performance de WiMax sera discuté. D'autre part, l'architecture matérielle générale de chaque module sera décrite. Cependant, l'implémentation et la réalisation de ces modules sur le matériel reconfigurable, ainsi que les contraintes de chaque module au niveau de l'espace occupé et la fréquence de l'horloge seront présentées en détail dans le chapitre 6.

2.1 La couche physique de WiMax

La couche physique de WiMax (PHY) est basée sur les normes IEEE 802.16-2004 [1] et IEEE 802.16e-2005 [2]. Cette couche a été conçue en se basant sur les standards de WiFi, en particulier IEEE 802.11a. PHY est basée sur le principe de multiplexage par répartition orthogonale de la fréquence (OFDM). La série des normes IEEE 802.16 définit dans son champ d'application quatre couches PHY, une de ces couches peut être utilisée avec la couche MAC pour développer un système sans fil à large bande. Les couches PHY définies dans IEEE 802.16 sont :

- **WirelessMAN SC** : une seule porteuse destinée à des fréquences au-delà de 11GHz nécessitant une condition ligne de vue (LOS).
- **WirelessMAN SCa** : une seule porteuse pour des fréquences entre 2GHz et 11GHz destinées aux opérations point à multipoint.
- **WirelessMAN OFDM** : une couche PHY OFDM à base de 256 point FFT, pour les conditions NLOS avec des fréquences entre 2GHz et 11GHz, également connue sous le nom de 802.16d.
- **WirelessMAN OFDMA** : basée sur OFDMA avec une taille FFT qui peut prendre une des valeurs suivantes : 128, 512, 1024, 2048, également connue sous le nom de IEEE 802.16e.

Comme tous les systèmes de communication, la couche physique de WiMax est formée de deux sous-systèmes : un émetteur et un récepteur. La section 2.2 explique en détail la série d'étapes reliées à l'émetteur. Plus tard dans ce chapitre, le récepteur sera décrit brièvement puisque la plupart des blocs du récepteur consiste simplement à effectuer la fonction inverse de son correspondant du côté émetteur.

2.2 Détails de l'émetteur WiMax

La figure 2.1 montre un émetteur WiMax à bande de base idéale. La première série d'étapes de l'émetteur WiMax est le codage du canal. Cette première série comprend le mélangeur (randomizer), l'encodeur Reed Solomon (Reed Solomon encoder), l'encodeur convolutionnel (convolutional encoder), le perforateur (puncturer), l'entrelaceur (interleaver) et le modulateur (modulator). La deuxième série d'étapes est liée à la construction du symbole OFDM dans le domaine des fréquences. Cette étape comprend l'insertion des sous-porteuses pilotes ainsi que les bandes de garde (pilot/guard insertion), la transformée inverse de Fourier et le préfixe cyclique (cyclic prefix). Bien que la figure 2.1 montre les composants logiques d'un émetteur, des composants similaires existent également au niveau du récepteur, dans l'ordre inverse, pour reconstruire la séquence des

informations transmises. Dans les sections suivantes, nous allons décrire en détail le fonctionnement de chaque bloc de la figure 2.1.

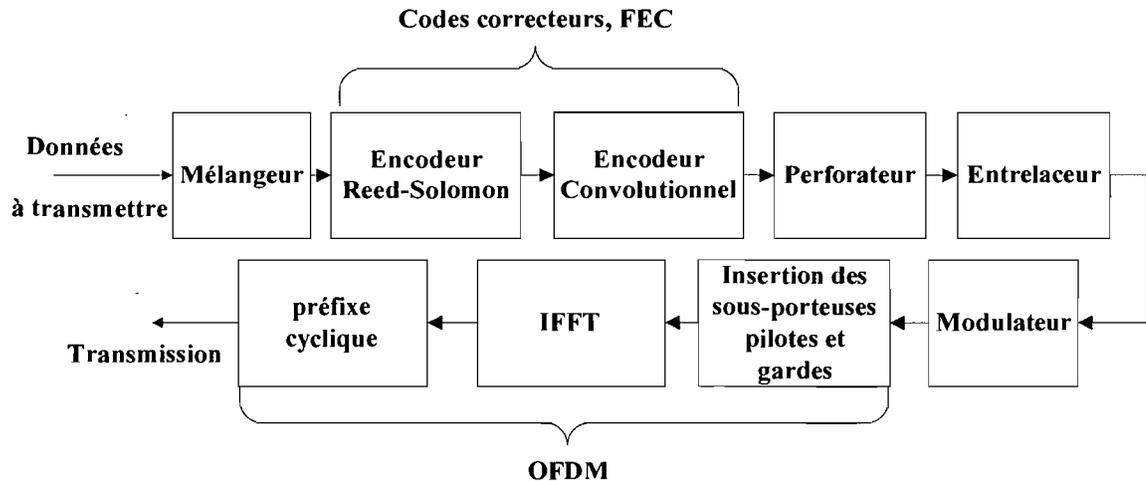


Figure 2.1 : Schéma d'un émetteur WiMax à bande de base.

2.2.1 Le mélangeur

La plupart des systèmes de télécommunication donne la performance optimale quand des données non corrélées sont transmises. Cependant, les données de l'utilisateur sont fortement corrélées (les chaînes de 1 et de 0 consécutifs). La transmission de telles chaînes pourrait se traduire par une perte de synchronisation (synchronization drift) entre l'émetteur et le récepteur. Comme une contre mesure à ce phénomène, le premier module de WiMax est le mélangeur. Il équilibre le nombre de '1' et de '0' dans la transmission pour permettre au récepteur d'extraire l'horloge du système, même si une longue chaîne corrélée est transmise.

En effet, le mélangeur est constitué de 15 registres à décalage avec deux « ou exclusifs ». Au début de chaque trame de données, ces registres sont initialisés par la séquence : [100101010000000]. Par contre, au début de chaque rafale (burst), le mélangeur est initialisé par 3 paramètres : l'ID de la station de base (BSID), l'intervalle de l'usage du

code (D/UIUC), l'index de la trame de données. Dans ce cas, les registres X_4 , X_5 et X_{10} sont mis à '1' (voir figure 2.2).

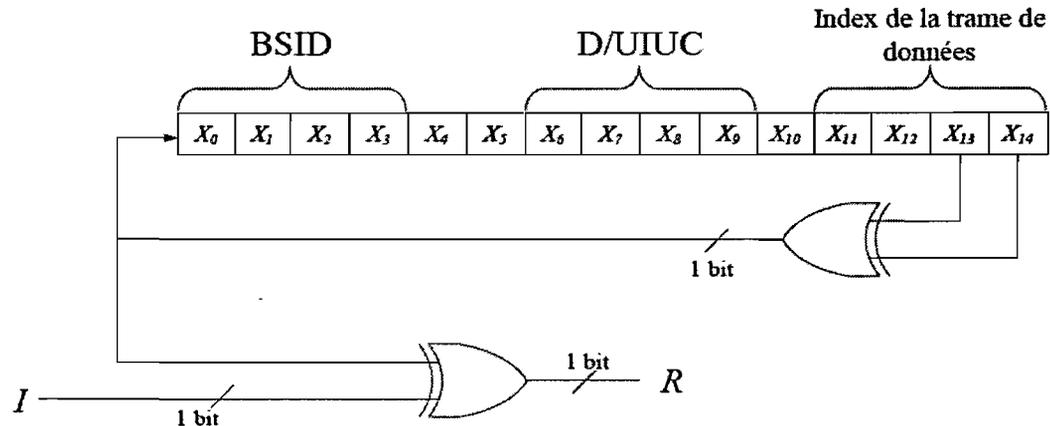


Figure 2.2 : Architecture du mélangeur.

2.2.2 La correction d'erreur directe dans WiMax (Forward Error Correction, FEC)

Dans la théorie de l'information, il existe deux stratégies pour la correction des erreurs dans les systèmes de communication [8]. La première stratégie est la répétition automatique de la demande (Automatic Repeat Request, ARQ). Dans ARQ, les codes du canal sont consacrés à la détection des erreurs dans les paquets transmis. Dans ce cas, si le récepteur n'a pas détecté d'erreur, les bits originaux sont récupérés et envoyés à l'utilisateur. Autrement, le récepteur rejette le paquet correspondant et envoie un accusé de réception négatif à l'émetteur pour la retransmission. D'autre part, la deuxième stratégie pour la correction d'erreur dans les systèmes de communication est la correction d'erreur directe, FEC (Forward Error Correction). En FEC, les bits sont encodés, non seulement pour la détection des erreurs dans les bits transmis, mais aussi pour la correction de ces erreurs. FEC peut être considéré comme une technique de diversité temporelle qui peut améliorer la fiabilité d'un système de communication. Principalement, il y a deux types des codes correcteurs qui réalisent la technique FEC : codes convolutifs (convolutional coding) et les codes en bloc (block codes). Dans le codage convolutif, le système fonctionne sur la séquence d'information continuellement sans la séparer en blocs. Par contre, l'encodeur

pour les codes en bloc transforme la séquence d'information continue en blocs de k symboles. Ensuite, l'encodeur opère sur ces blocs séparément. Au niveau de la couche physique, WiMax a adopté l'utilisation de la stratégie FEC pour la correction des erreurs dans les bits transmis. De plus, pour améliorer la performance et la fiabilité, WiMax utilise le principe des codes correcteurs concaténés (concatenated error correction codes) qui consiste à la combinaison d'un code intérieur et un code extérieur. Dans ce contexte, WiMax considère le code en bloc Reed Solomon comme un code correcteur extérieur et l'encodeur convolusionnel comme un code intérieur. L'encodeur Reed Solomon et l'encodeur convolusionnel ainsi que leurs paramètres dans WiMax seront présentés dans les sections suivantes.

2.2.2.1 Encodeur Reed-Solomon

Les codes Reed-Solomon (RS) ont été découverts par Irvin Reed et Gus Solomon en 1959 [8]. Récemment, les codes RS ont été largement utilisés dans de nombreuses applications telles que la communication par satellite, la diffusion de vidéo numérique (digital video broadcast) et la communication mobile sans fil. Ces codes, qui fonctionnent en bloc, sont basés mathématiquement sur les champs finis de Galois. Les codes RS permettent de corriger les erreurs et les effacements grâce à des symboles de contrôle ajoutés après l'information. Un code Reed-Solomon est spécifié par ses paramètres $RS(N, K, T)$ avec un symbole de L bits. L'encodeur RS prend K symboles de L bits chacun et ajoute $2T$ symboles de parité pour construire N symboles. Ainsi N, K, T sont définis comme suit :

- N : le nombre de symboles après l'encodage.
- K : le nombre de symboles avant l'encodage.
- T : le nombre de symboles qui peuvent être corrigés avec $T = \frac{N-K}{2}$.

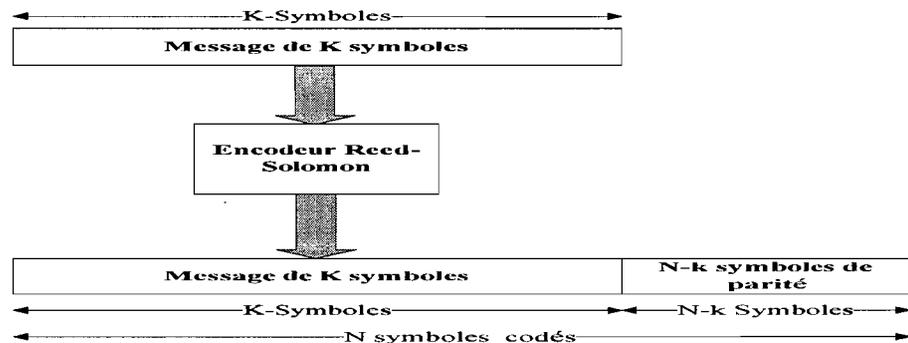


Figure 2.3 : Encodeur Reed Solomon

La norme de WiMax spécifie que l'encodeur de Reed-Solomon doit être tiré systématiquement de RS ($N = 255, K = 239, T = 8$). Ainsi, le RS fait ajouter 16 symboles de redondance à la fin d'un bloc de 239 symboles afin de pouvoir corriger 8 symboles au maximum. Les $2T$ symboles de parité sont donnés par l'équation suivante :

$$p(x) = (i(x) * x^{N-K}) \text{ modulo } g(x) \quad [2.1]$$

Où $g(x)$ est le polynôme générateur du code, $i(x)$ est le bloc d'information à coder et $p(x)$ sont les $2T$ symboles de parité. Il est à noter que $p(x)$, $i(x)$ et $g(x)$ sont des séquences binaires représentées en polynômes. Par exemple, si $i(x) = x^3 + x^2 + 1$, alors le polynôme $i(x)$ est équivalent à la séquence binaire «1101».

Le nombre de bit L par symbole est donné par :

$$N = 2^L - 1 \quad [2.2]$$

$$L = \ln(N + 1) / \ln(2) \quad [2.3]$$

En remplaçant la valeur de N dans l'équation 2.3, on obtient la valeur de L qui est égale à 8. Ainsi, nous pouvons déduire que dans le contexte de l'encodeur RS dans WiMax, un symbole représente un octet. Une autre note à signaler au niveau de l'encodeur RS, c'est que les spécifications de WiMax consistent à extraire les premiers $2T'$ octets de parité à partir des 16 octets de parité générés par l'encodeur RS. $2T'$ est égale à 8, 4, 16, 8, 12, 12

pour les profils de modulation QPSK $\frac{1}{2}$, QPSK $\frac{3}{4}$, 16QAM $\frac{1}{2}$, 16QAM $\frac{3}{4}$, 64QAM $\frac{2}{3}$, 64QAM $\frac{3}{4}$ respectivement (voir tableau 2.1). Par conséquent, les octets restants sont éliminés.

Le polynôme générateur de l'encodeur RS est de la forme :

$$g(x) = (x + \lambda^0)(x + \lambda^1) \dots (x + \lambda^{2T-1}) \quad [2.4]$$

Avec $\lambda = 02_{hex}$ (hexadécimal).

La figure 2.4 montre l'architecture de l'encodeur Reed Solomon. Dans cette architecture, les opérateurs arithmétiques effectuent une addition ou une multiplication finie sur un octet complet. La norme 802.16 [1] spécifie que chaque trame de données à encoder est préalablement divisée en sous-groupes. Chaque sous-groupe est constitué d'un certain nombre d'octets qui varie selon le type de modulation et le taux de transfert sélectionné dans WiMax (voir tableau 2.2).

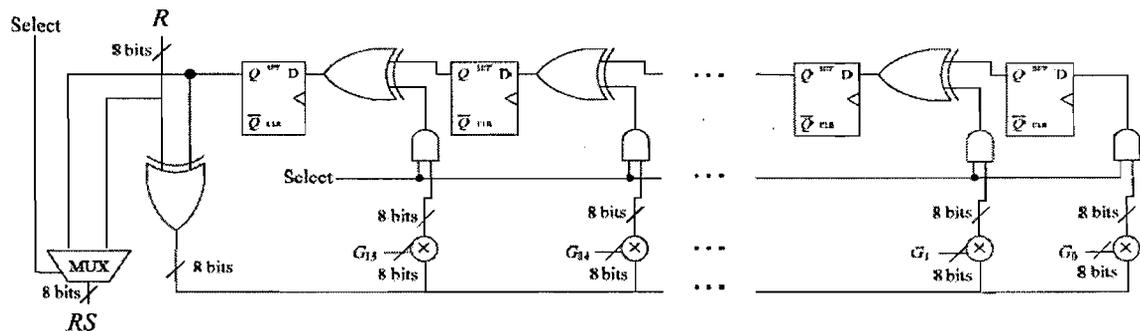


Figure 2.4 : Architecture de l'encodeur Reed Solomon.

2.2.2.2 Codage Convolutif (CC)

Le codage convolutif (encodeur convolutionnel) est une technique de correction d'erreur directe (FEC) qui est particulièrement adaptée à un canal de communication dans lequel le signal transmis est altéré essentiellement par le bruit blanc gaussien (AWGN) [9]. Un encodeur convolutionnel a les propriétés suivantes : (1) chaque m bits à encoder est

transformé en n bits, où m/n est le taux de codage ($n \geq m$) et (2) la transformation est une fonction des derniers k bits, où k est la longueur du code. Le convolveur, similairement à l'encodeur Reed-Solomon, ajoute de la redondance dans les bits transmis afin de permettre au récepteur de corriger les bits erronés. Dans WiMax, le convolveur produit un taux de transfert de $1/2$, c'est-à-dire que pour chaque bit à l'entrée, deux bits de sortie sont produits. La norme de WiMax spécifie que l'encodeur doit utiliser les polynômes $G1$ et $G2$ pour générer ses deux bits de sortie, X et Y :

$$G1 = 171_{oct} \text{ pour } X \quad [2.5]$$

$$G2 = 133_{oct} \text{ pour } Y \quad [2.6]$$

L'architecture du convolveur est constituée de six registres à décalage et de deux « ou exclusifs », comme illustrée à la figure 2.5.

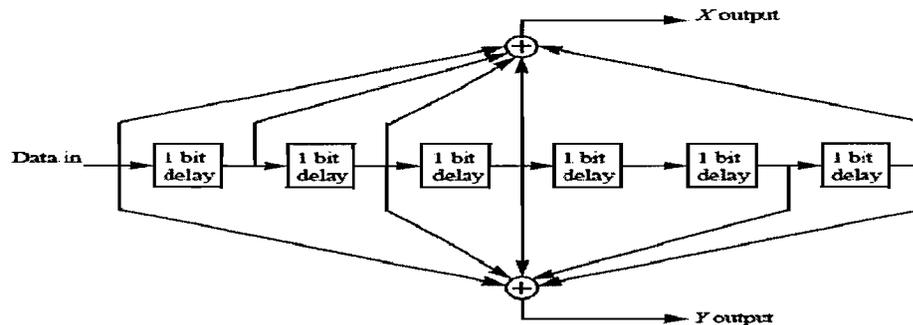


Figure 2.5 : Architecture du Convolveur.

La stratégie principale pour le décodage d'une séquence codée en utilisant un encodeur convolutionnel est basée sur l'algorithme de Viterbi qui sera présenté plus tard dans ce chapitre. Finalement, une analyse détaillée des codes Reed-Solomon et des codes convolutionnels, ainsi que d'autres méthodes de codage classique, peut être trouvée dans [8-9].

2.2.3 Perforateur

Dans la théorie de l'information, la perforation est le processus qui consiste à enlever quelques uns des bits de parité, après l'encodage avec un code de correction d'erreur.

Alors, le rôle du perforateur est de réduire la redondance dans la séquence transmise. En effet, le perforateur augmente considérablement la flexibilité du système sans augmenter sa complexité d'une manière significative. Dans WiMax, le perforateur ajuste le taux de transfert spécifié dans le standard à partir du taux de 0.5 produit par l'encodeur convolutionnel. Cinq taux de transfert sont supportés par le perforateur dans WiMax : 1/2, 2/3, 3/4, 5/6 et 7/8. Le tableau 2.1 présente la perforation effectuée en fonction du taux de transfert.

Tableau 2.1 : Perforation en fonction de taux de transfert.

Taux de Transfert	1/2	2/3	3/4	5/6	7/8
Données X sélectionnées (X_{N-1}, \dots, X_1, X_0)	1	01	101	10101	1010001
Données Y sélectionnées (Y_{N-1}, \dots, Y_1, Y_0)	1	11	011	01011	0101111
P sérialisé	Y_0X_0	$Y_1Y_0X_0$	$X_2Y_1Y_0X_0$	$X_4Y_3X_2Y_1Y_0X_0$	$X_6Y_5X_4Y_3Y_2Y_1Y_0X_0$

2.2.4 Entrelaceur

La performance des codes de correction d'erreur sera considérablement améliorée si les erreurs sont distribuées de façon aléatoire [4]. Ainsi, le rôle de l'entrelaceur est de réordonner les bits transmis de façon à ce que les bits consécutifs se transmettent sur différentes fréquences et phases. Ainsi, le bruit qui affecte une bande de fréquence sera réparti sur plusieurs symboles de données transmis. En conséquent, les bits erronés seront uniformément réparti. Dans ce contexte, la probabilité de récupération des données erronées sera augmentée.

Dans WiMax, les bits sont entrelacés en deux permutations, la première permutation consiste à ce que les bits adjacents sont mis en correspondance avec des sous-porteuses

non-adjacentes. La deuxième permutation garantit que les bits adjacents sont à tour de rôle mappés aux bits les plus et moins importants de la constellation, évitant ainsi les longues séries de bits non fiables [2].

Soit N_{CPC} le nombre de bits codés par sous-porteuse, c'est-à-dire, 1, 2, 4 ou 6 pour BPSK, QPSK, 16-QAM, 64-QAM, respectivement. Soit $s = \text{floor}\left(\frac{N_{CPC}}{2}\right)$, k est l'indice du bit encodé avant la première permutation, m_k l'indice de ce bit après la première et avant la deuxième permutation, j_k est l'indice du même bit après la deuxième permutation, juste avant la modulation. N_{cbps} est le nombre de bits encodés par symbole, c'est-à-dire, 192, 384, 768 ou 1152 pour BPSK, QPSK, 16-QAM et 64-QAM respectivement. La première permutation est alors accomplie par l'équation suivante :

$$m_k = \left(\frac{N_{cbps}}{12}\right) \cdot k_{mod12} + \text{floor}\left(\frac{k}{12}\right) \quad [2.7]$$

La seconde permutation est accomplie par l'équation suivante :

$$j_k = s \cdot \text{floor}\left(\frac{m_k}{s}\right) + \left(m_k + N_{cbps} - \text{floor}\left(12 \cdot \frac{m_k}{N_{cbps}}\right)\right) \text{mod}(s) \quad [2.8]$$

Le déentrelaceur, qui effectue l'opération inverse de l'entrelaceur, est également défini par deux permutations. Soit j l'indice de bit reçu avant la première permutation, m_j l'indice de ce bit après la première et avant la deuxième permutation, k_j l'indice de ce même bit après la deuxième permutation, et juste avant le décodage. m_j et k_j sont donnés par l'équation [2.9] et [2.10] respectivement.

$$m_j = s \cdot \text{floor}\left(\frac{j}{s}\right) + \left(j + \text{floor}\left(12 \cdot \frac{j}{N_{cbps}}\right)\right) \text{mod}(s) \quad [2.9]$$

$$k_j = 12m_j - (N_{cbps} - 1) \text{ floor } (12m_j/N_{cbps}) \quad [2.10]$$

Nous remarquons que la première permutation dans le déentrelaceur est l'inverse de la deuxième permutation dans l'entrelaceur, et inversement. La figure 2.6 montre l'encodeur Reed-Solomon suivi par l'encodeur convolutionnel, l'entrelaceur et le modulateur (BPSK) implémentés en Simulink.

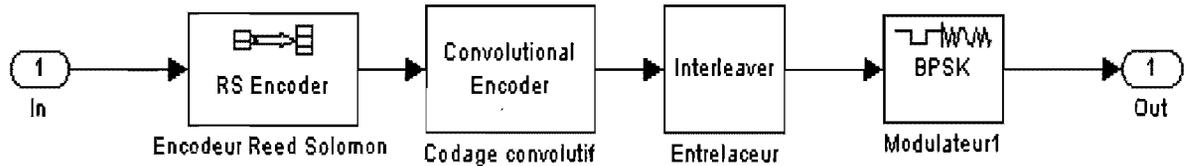


Figure 2.6 : RS, CC, entrelaceur et modulateur BPSK implémenté en Simulink.

En matériel, l'entrelaceur est réalisé en utilisant une mémoire RAM. Pour chaque bit k , la valeur de ce bit doit être écrite à l'adresse j_k dans la RAM. Après avoir écrit le nombre de bits qui correspond à la valeur N_{cbps} . Les bits sont lus séquentiellement de la mémoire RAM. L'architecture de l'entrelaceur sera discutée en détail dans le chapitre 6.

2.2.5 Modulateur

Le processus de la modulation consiste à convertir la séquence de bits binaire en une séquence de symboles (nombres complexes). La norme 802.16 [1] spécifie que quatre schémas de modulation sont adoptés par WiMax : *Binary Phase Shift Keying* (BPSK), *Quadrature Phase Shift Keying* (QPSK), *16 Quadrature Amplitude Modulation* (16-QAM) et *64 Quadrature Amplitude Modulation* (64-QAM). Selon le type de modulation sélectionné, des groupes de 1, 2, 4, et 6 bits sont considérés en entrée pour les modulations BPSK, QPSK, 16-QAM et 64-QAM respectivement. La figure 2.7 montre les quatre schémas de modulation adoptés par WiMax. Les points de la constellation de la figure 2.7 doivent être normalisés en multipliant les points de constellation avec le facteur c indiqué pour assurer une puissance moyenne égale.

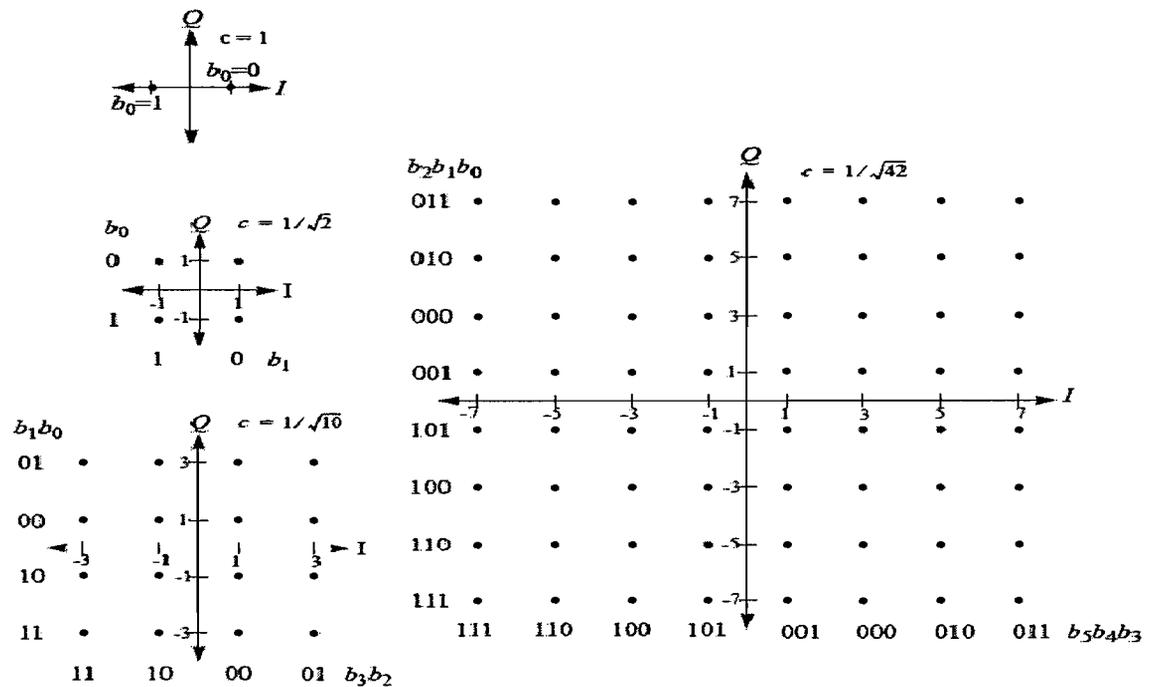


Figure 2.7 : Les points de constellation BPSK, QPSK, 16-QAM, et 64-QAM dans WiMax

En effet, WiMax est caractérisé par la modulation et le codage adaptif (Adaptive modulation and coding, AMC). Récemment, la technique AMC a beaucoup attiré l'attention en raison de sa capacité à améliorer la performance et le débit des systèmes sans fils [5]. Cette technique améliore la performance du système en augmentant le taux de transmission et en réduisant le taux d'erreur dans les bits transmis. La norme 802.16 [1] spécifie que sept profils de modulation et de codage doivent être supportés par WiMax. Donc, un système WiMax typique comporte sept profils de transmission-réception (voir tableau 2.2). Le choix du profil correspondant se base sur la valeur du rapport signal/bruit (SNR) estimé au récepteur. Lorsque les conditions du canal sont favorables (rapport SNR élevé), les schémas de modulation à haute efficacité spectrale comme le 64-QAM et le 16-QAM sont employés. Par contre, dans le cas où la valeur du rapport SNR est faible, l'utilisation des schémas de modulation de faible efficacité spectrale (BPSK, QPSK) est

favorisée. Le tableau 2.2 montre les différents profils de modulation et de codage adoptés par WiMax

Tableau 2.2 : Les sept profils de modulation et de codage adoptés par WiMax.

Profil	Modulation	Taille du bloc non codé	Code RS	Taux de CC	Taux général
0	BPSK	12	(12, 12, 0)	1/2	1/2
1	QPSK	24	(32, 24, 4)	2/3	1/2
2	QPSK	36	(40, 36, 2)	5/6	3/4
3	16QAM	48	(64, 48, 8)	2/3	1/2
4	16QAM	72	(80, 72, 4)	5/6	3/4
5	64QAM	96	(108, 96, 8)	3/4	2/3
6	64QAM	108	(120, 108, 6)	5/6	3/4

2.2.6 OFDM

OFDM est une technique de transmission par porteuses multiples qui consiste à diviser la largeur de la bande disponible en une multitude de bandes étroites pour les exploiter individuellement et simultanément. L'idée générale de la transmission en parallèle a été proposée à la fin des années 50 [11, 12, 13]. Cependant, en raison de la complexité de l'implémentation des systèmes OFDM, la modulation OFDM n'a pas été utilisée qu'au début des années 90. En effet, ce n'est que récemment, après le développement des circuits électroniques VLSI, que la technique OFDM a été utilisée commercialement. La modulation OFDM peut être décrite comme étant un multiplexage fréquentiel où N symboles sont modulés simultanément par N sous-porteuses orthogonales. Les différents signaux individuels ainsi formés sont additionnés pour former le signal OFDM. Ce dernier est donné par l'équation suivante :

$$S(n) = \sum_{k=0}^{N-1} D(k) e^{j2\pi n.k/N} \quad [2.11]$$

Avec $n = 0, 1, \dots, N - 1$.

Où $D(k)$ représentent les symboles complexes à transmettre (issues de la constellation de la modulation) et N est le nombre de sous-porteuses employées. Dans ce contexte, nous remarquons que l'équation 2.11 n'est rien d'autre que la transformée inverse de Fourier (IFFT). À ce propos, la norme de WiMax, pour la couche OFDM PHY avec 256 points FFT, spécifie que le nombre de sous-porteuses (N) à l'entrée de l'IFFT est 256. Ces sous-porteuses sont de 3 types : données, pilotes et gardes. La figure 2.8 montre les 256 sous-porteuses insérées tout au long du spectre du symbole OFDM. 192 sous-porteuses sont utilisées pour la transmission des données, 8 sous-porteuses sont utilisées pour l'estimation du canal (channel estimation) et la compensation du décalage de la fréquence porteuse (carrier frequency offset estimation), et les sous-porteuses restantes sont utilisées comme bandes de garde pour prévenir les interférences.

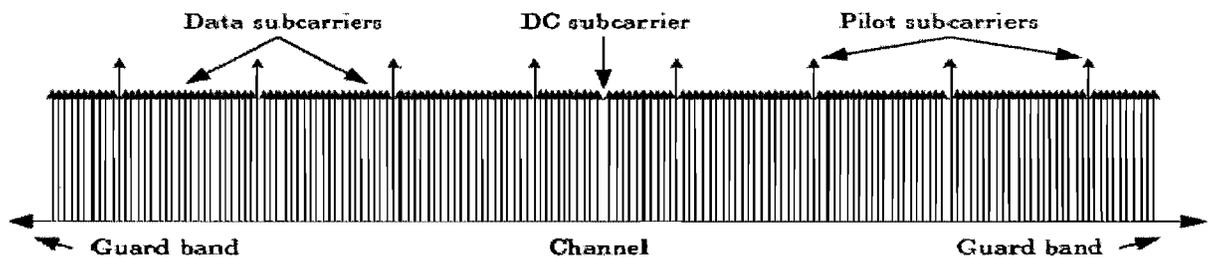


Figure 2.8 : Les sous-porteuses de données, de gardes et pilotes dans un symbole OFDM

Donc, le symbole OFDM est constitué de 256 sous-porteuses, à chaque sous-porteuse on associe un indice fréquentiel, ces indices sont répartis de -128 à +127. La sous-porteuse d'indice zéro est connue comme la sous-porteuse DC. Les différents paramètres de l'OFDM comme spécifiés dans le standard de WiMax sont donnés dans le tableau ci-contre.

Tableau 2.3 : Les paramètres de l'OFDM dans WiMax.

Paramètres	Valeur
Taille de FFT	256
Nombre de sous-porteuse de données	192
Nombre de sous-porteuse de garde	56
Nombre de sous-porteuse pilotes	8
Fraction de garde (préfixe cyclique)	1/32,1/16,1/8
facteur d'échantillonnage, n	8/7
Bande du canal	3.5 MHZ
Espacement entre les sous porteuses	15.625 KHZ
Temps du symbole OFDM	72 μ s

2.2.7 Les sous-porteuses pilotes

Les sous-porteuses pilotes sont des signaux spéciaux insérés entre chaque symbole OFDM. L'insertion de ces sous-porteuses permet d'effectuer une compensation du décalage de la fréquence porteuse (carrier frequency offset compensation) au récepteur. En outre, les récentes recherches [30] ont montré que les sous-porteuses pilotes peuvent aussi être utilisés pour l'estimation et la compensation des canaux qui varient rapidement dans le temps (channel estimation and compensation in fast time-varying channels). Le tableau 2.4 montre les fréquences dédiées aux sous-porteuses pilotes ainsi que leurs valeurs correspondantes.

Tableau 2.4 : Indices des sous-porteuses pilotes et leurs valeurs correspondantes.

Indice	Valeur
-88	1
-63	-1
-38	1
-13	-1
13	1
38	1
63	1
88	1

2.2.8 Les sous-porteuses de garde (bandes de garde)

Dans le but de prévenir les interférences, les sous-porteuses qui sont aux deux extrémités du spectre du symbole OFDM, ne sont pas utilisées et sont mises à zéro. Le standard de WiMax spécifie que les sous-porteuses d'indice -128 à -101 inclusivement et de 101 à 127 inclusivement sont utilisées comme bandes de garde.

2.2.9 Le préfixe cyclique

Le retard de la propagation (delay spread) est un type de distorsion qui est provoquée quand un signal identique arrive à différents moments à sa destination. Ainsi, pour surmonter le problème du retard de la propagation, le standard de WiMax spécifie qu'un préfixe cyclique doit être ajouté pour chaque symbole OFDM. En effet, le préfixe cyclique est un intervalle de garde au début du symbole OFDM. Cet intervalle consiste à répéter les m derniers échantillons de la transformée inverse de Fourier au début de cette transformée. La valeur de m doit être choisie de façon à ce que la durée du préfixe cyclique soit supérieure à la durée maximale du retard de la propagation causée par les phénomènes de propagation à trajets multiples.

La figure 2.9 montre la conception d'un émetteur OFDM en Simulink. Tout d'abord, une matrice de taille $[192 \times 1]$, qui correspond aux sous-porteuses des données, est reçue du modulateur. Ensuite, les sous-porteuses pilotes et gardes sont insérées, à leurs adresses correspondantes, pour former une matrice de taille $[256 \times 1]$. Cette dernière est passée vers le bloc IFFT dans Simulink pour calculer la transformée inverse de Fourier. Finalement, les dernières m lignes de la matrice sont répétées au début de cette matrice pour former le préfixe cyclique.

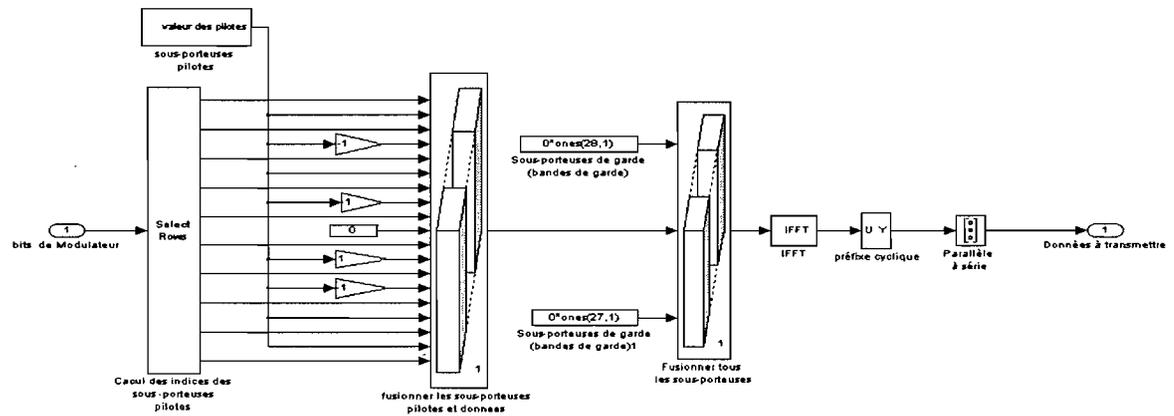


Figure 2.9 : Émetteur OFDM dans Simulink.

2.2.10 Effet des modules de la chaîne de transmission sur une trame exemple

Alors que les modules de la chaîne de transmission de WiMax ont été largement et profondément présentés et discutés dans les sections précédentes, la complexité de ces modules peut entraîner le lecteur à ne pas pleinement comprendre le contexte de la chaîne de transmission dans WiMax. Ainsi, pour mieux illustrer les différentes étapes dans l'émetteur WiMax, nous allons considérer une trame reçue de la couche MAC qui doit être transmise suivant le profil QPSK 3/4. Donc, cette trame doit passer par tous les modules de la chaîne de transmission montrés dans la figure 2.1 (la trame sera passée au mélangeur et la sortie du mélangeur sera passée à l'encodeur Reed-Solomon etc.). Par conséquent, l'effet de chaque module de la chaîne de transmission sur cette trame sera brièvement montré et discuté. Cela permettra aux lecteurs de mieux comprendre le contexte global de la couche physique du WiMax. Comme déjà indiqué, cette trame sera transmise suivant le profil QPSK 3/4, ainsi la longueur de la trame doit être 35 octets (spécification du standard). Les octets de cette trame ont été choisis d'une façon aléatoire. Cette trame est égale à [45 29 C4 79 AD 0F 55 28 AD 87 B5 76 1A 9C 80 50 45 1B 9F D9 2A 88 95 EB AE B5 2E 03 4F 09 14 69 58 0A 5D].

Pour préparer la trame pour la transmission, cette dernière doit passer d'abord par l'intermédiaire du mélangeur. Comme le montre le circuit de la figure 2.2, le fonctionnement du mélangeur est très simple. Chaque octet à transmettre entre séquentiellement dans le circuit (le bit le plus significatif en premier). Par conséquent, la trame résultante après le mélangeur sera égale à [D4 BA A1 12 F2 74 96 30 27 D4 88 9C 96 E3 A9 52 B3 15 AB FD 92 53 07 32 C0 62 48 F0 19 22 E0 91 621A C1]. On remarque que le mélangeur n'a pas d'effet sur la longueur de la trame. Ainsi, le nombre d'octets à l'entrée du mélangeur est le même que celui à la sortie. Il est à noter que pour l'initialisation des registres du mélangeur, nous avons supposé que les paramètres BSID, D/UIUC et l'index de la trame de données sont égales respectivement à 1,7et 1.

En deuxième lieu, cette trame doit passer par l'encodeur Reed-Solomon. Comme déjà indiqué l'encodeur RS ajoute quelques octets de redondance à la trame. Ainsi, après l'encodage, la trame résultante sera égale à : [49 31 40 BF D4 BA A1 12 F2 74 96 30 27 D4 88 9C 96 E3 A9 52 B3 15 AB FD 92 53 07 32 C0 62 48 F0 19 22 E0 9162 1A C1 00]. On peut observer que l'encodeur RS n'a pas d'effet sur les octets originaux. Cet encodeur ajoute quatre octets de redondance au début de la trame originale (ainsi qu'un zéro octet à la fin de la trame). Ensuite, les octets originaux ainsi que les octets de redondance sont envoyés vers l'encodeur convolutionnel. Par conséquent, la longueur de la trame à la sortie de l'encodeur RS est égale à 40 octets.

Ensuite, les 40 octets reçus de l'encodeur RS, doivent être encodés (avec le CC) et perforés. Dans ce contexte, alors que le CC ajoute des bits de redondance (pour chaque bit à l'entrée, deux bits sont produits), le perforateur élimine quelques un de ces bits pour améliorer la performance. Ainsi, après le traitement avec l'encodeur convolutionnel et le perforateur, la trame résultante sera : [3A 5E E7 AE 49 9E 6F 1C 6F C1 28 BC BD AB 57 CD BC CD E3 A7 92 CA 92 C2 4D BC 8D 78 32 FB3 BF DF 23 ED 8A 94 16 27 A5 65 CF 7D 16 7A 45 B8 09 CC]. On peut observer que la trame résultante est de longueur 48 octets. Par conséquent, huit octets de redondance ont été ajoutés à cette trame.

Pour améliorer la performance des codes correcteurs RS et CC, les 48 octets reçus du CC doivent être entrelacés. Comme indiqué précédemment, l'entrelaceur simplement réorganise les indices des bits dans une trame suivant les équations 2.7 et 2.8. Par conséquent, après le traitement de l'entrelaceur, la trame résultante sera : [77 FA 4F 17 4E 3E E6 70 E8 CD 3F 76 90 C4 2C DB3 F9 B7 F13 43 6C F19A BD ED OA 1C D8 1B EC 9B 30 15 BA DA 31 F5 50 49 7D 56 ED B4 88 CC 72 FC 5C]. Nous remarquons, dans ce contexte, que le nombre d'octets à la sortie de l'entrelaceur est 48. Ainsi, le processus d'entrelacement n'a pas d'effet sur la longueur de la trame.

Après l'entrelaceur, les 48 octets (384 bits) reçus de l'entrelaceur doivent être modulés en nombres complexes (modulateur). Comme déjà indiqué, nous supposons que la trame sera transmise suivant le profil QPSK 3/4, ainsi chaque deux bits seront modulés en un nombre complexe. Pour le profil QPSK 3/4, la norme de WiMax [1] spécifie les transformations suivantes [00, $\sqrt{2} + i\sqrt{2}$] (la séquence 00 sera transformée en $\sqrt{2} + i\sqrt{2}$), [01, $\sqrt{2} - i\sqrt{2}$], [10, $-\sqrt{2} + i\sqrt{2}$], [11, $-\sqrt{2} - i\sqrt{2}$]. Par exemple, l'octet 5C [01011101] sera transformé en quatre nombre complexe [$\sqrt{2} - i\sqrt{2}$, $\sqrt{2} - i\sqrt{2}$, $-\sqrt{2} - i\sqrt{2}$, $\sqrt{2} - i\sqrt{2}$]. Ainsi, après la modulation, les 384 bits reçus de l'entrelaceur seront modulés en 192 symboles (nombres complexes). La trame résultante après la modulation est montrée dans l'annexe C.

Ensuite, les sous-porteuses pilotes ainsi que les bandes de garde doivent être ajoutés aux 192 symboles modulés (reçus du modulateur) pour former les 256 symboles à l'entrée de la transformée inverse de Fourier. Les bandes de garde sont obtenus par l'ajout de 28 zéro au début des 192 symboles modulés et de 27 zéro à la fin de ces symboles. Les sous-porteuses pilotes, de l'autre coté, sont obtenues en insérant des signaux spéciaux à des indices spécifiques. Les valeurs de ces signaux ainsi que leurs indices correspondants sont montrés dans le tableau 2.4. La trame résultante après l'insertion des sous-porteuses pilotes et les bandes de garde est montrée dans l'annexe C.

À ce point, les 256 symboles déjà traités sont prêts pour la transformée inverse de Fourier. Ainsi, après le traitement de l'IFFT, 256 échantillons (nombres complexes) sont obtenus. Ces 256 échantillons constituent un symbole OFDM. La valeur de ce symbole OFDM est montrée dans l'annexe C.

Enfin, le préfixe cyclique doit être ajouté. Ce dernier est accompli simplement en répétant les derniers m échantillons de la transformée inverse de Fourier au début de cette transformée (m est la longueur du préfixe cyclique).

2.3 Détails du récepteur WiMax

Comme dans la plupart des normes de communication, le corps de la norme définit les détails de conception de l'émetteur, tandis que les détails du récepteur sont laissés au fabricant pour les définir. Toutefois, les blocs de la chaîne de réception doivent être le miroir des blocs de la chaîne de transmission. Dans ce travail, les mêmes blocs de la chaîne de transmission ont été inversés et légèrement modifiés pour implémenter la chaîne de réception. Cependant, un seul bloc de la chaîne de réception a été modifié d'une manière significative. Ce bloc est le décodeur de Viterbi qui effectue la fonction inverse de l'encodeur convolutionnel. De plus, deux nouveaux blocs ont été ajoutés à la chaîne de réception. Ces deux blocs sont l'égaliseur du canal et l'estimateur du rapport signal/bruit. Le module égaliseur du canal permet au récepteur de corriger les perturbations des signaux transmis (par exemple la modification de l'énergie, l'amplitude et la phase du signal originale) causées par le canal de communication. Dans ce contexte, la méthode de l'erreur quadratique minimale est utilisée pour implémenter l'égaliseur du canal. Le bloc estimateur du rapport signal bruit calcule le rapport signal bruit (SNR) et transmet la valeur correspondante vers l'émetteur pour que ce dernier puisse appliquer le profil de modulation et de codage correspondant. Les deux nouveaux blocs du récepteur (égaliseur du canal et estimateur du rapport signal bruit) ainsi que le décodeur de Viterbi seront présentés en détail dans les sections suivantes. Finalement, il est à noter que les modules égaliseur du

canal et estimateur du rapport signal bruit ont été modélisés en Simulink et n'ont pas été considérés lors de l'implémentation de la couche physique de WiMax en matériel.

2.3.1 L'égaliseur du canal

En pratique, les canaux de transmission sans fil sont souvent variables dans le temps ou inconnus a priori. Les paramètres des canaux de communication qui peuvent varier en cours de la transmission incluent, parmi d'autres, le bruit (noise) et le retard de la propagation (delay spread). En effet, l'évolution de l'environnement de transmission au cours du temps, modifie aléatoirement l'énergie, l'amplitude, et la phase du signal transmis. En outre, les systèmes de communication sans fil sont limités en capacité et en performance à cause de certains facteurs liés au canal de transmission [33]. Dans ces cas, l'estimation et l'égalisation du canal en temps réel sont essentielles afin de garantir une bonne détection des signaux transmis. Le rôle de l'estimateur du canal est de trouver approximativement les valeurs des atténuations et des distorsions que les signaux subissent lors de leurs propagations dans les canaux sans fils. L'égaliseur, en se basant sur les informations fournies par l'estimateur, essaie de corriger, en temps réel, les signaux reçus. Dans ce contexte, il existe une multitude de méthodes d'estimation du canal notamment la méthode d'estimation aveugle (blind estimation) [27-28] et la méthode d'estimation turbo [29]. Cependant, pour les systèmes de transmission OFDM, on utilise souvent des méthodes d'estimation basées sur l'insertion des sous-porteuses pilotes ou d'un symbole OFDM redondant dont la valeur est connue a priori par le récepteur [30-31]. Au niveau de WiMax, l'estimation de l'état du canal est accomplie en insérant une seule sequence pilote (symbole OFDM pilote) x_{ref} . La valeur du symbole pilote x_{ref} est spécifiée dans [1]. x_{ref} est exprimé comme suit :

$$x_{ref} = [x_{ref}(0) \ x_{ref}(1) \ \dots \ x_{ref}(N - 1)] \quad [2.12]$$

Tout comme les autres symboles transmis, x_{ref} est altéré par les perturbations de transmission dues au canal, de telle sorte que le récepteur reçoit une version altérée du symbole OFDM pilote x_{ref} . Connaissant a priori ce symbole, le récepteur est en mesure

d'estimer les facteurs d'atténuation du canal qui le guideront dans ses prises de décision pour détecter les symboles émis. Dans ce travail, nous avons utilisé l'erreur quadratique minimale LSE (Least Square Error) pour l'égalisation du canal. La méthode LSE était choisie grâce à sa simplicité et sa performance ainsi que son efficacité pour les systèmes WiMax [30]. Cette méthode est présentée en détail dans la section suivante.

2.3.1.1 Erreur quadratique minimale (LSE)

Le signal OFDM reçu peut être mathématiquement exprimé comme suit [30] :

$$r_{ref}(k) = x_{ref}(k).h(k) + n(k) \quad [2.13]$$

Où, k est l'indice de la sous porteuse, $r(k)$ est le signal reçu, $x(k)$ est le signal de référence transmis (séquence pilote), $n(k)$ est le bruit blanc gaussien et $h(k)$ est la réponse fréquentielle du canal. L'objectif de l'estimateur du canal est de calculer approximativement $h(k)$. La méthode de l'erreur quadratique minimale consiste à estimer la valeur de $h(k)$ pour laquelle la valeur de $[r(k) - x(k).h(k)]$ sera minimale. Mathématiquement parlant, l'erreur quadratique minimale résout l'équation suivante :

$$\widehat{h}(k) = \operatorname{argmin} |r(k) - x(k)h(k)|^2 \quad [2.14]$$

Où $\widehat{h}(k)$ est la réponse fréquentielle du canal estimée. Simplement la solution de [2.14] est :

$$\widehat{h}(k) = \frac{r(k)}{x(k)} \quad [2.15]$$

Après l'estimation de la réponse fréquentielle du canal pour chaque sous-porteuse, l'égalisation du canal est accomplie en multipliant chaque symbole modulé reçu par son $\widehat{h}(k)$ correspondant. Le circuit Simulink correspondant à la méthode de l'erreur quadratique minimale est montré dans la figure 2.10.

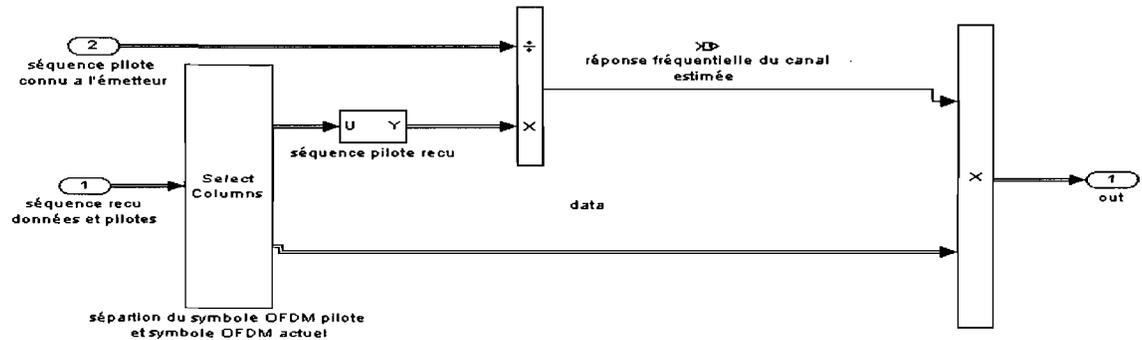


Figure 2.10 : Erreur quadratique minimale dans Simulink

2.3.2 Estimation du rapport signal/bruit

Pour estimer le rapport signal/bruit, nous avons utilisé la métrique de l'amplitude du vecteur d'erreur (Error Vector Magnitude, EVM). EVM est une métrique largement utilisée dans les systèmes de communication numérique. Elle est généralement utilisée pour évaluer la qualité du signal de l'émetteur et pour déterminer le degré d'altération du signal, ce qui offre une excellente vue sur la qualité de l'émetteur [26]. En effet, quand un signal est transmis par un émetteur idéal, tous les symboles modulés transmis prendront leurs places idéales dans la constellation I-Q. Cependant, le signal transmis par l'émetteur subit des imperfections et des dégradations diverses qui causent les symboles modulés reçus à s'éloigner de leurs places idéales (voir figure 2.12). L'amplitude du vecteur d'erreur (EVM) est donc définie comme la différence entre la place idéale du symbole modulé reçu et la place observée du même symbole [26].

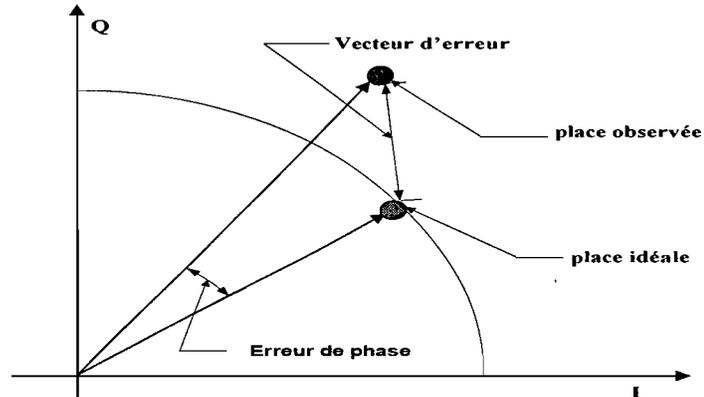


Figure 2.11 : EVM dans la constellation I-Q.

Soit \hat{c}_k le vecteur qui correspond à la place observée du symbole modulé reçu et c_k le vecteur correspondant à la place idéale du même symbole. L'EVM est définie par la formule suivante:

$$EVM = |c_k - \hat{c}_k| \quad [2.16]$$

Dans le cas de l'OFDM, où N symboles modulés sont encapsulés dans un symbole OFDM, l'EVM est définie comme suit:

$$EVM_{RMS} = \sqrt{\sum_{k=1}^N \frac{|c_k - \hat{c}_k|^2}{N}} \quad [2.17]$$

À ce propos, [25] a prouvé que la relation entre le rapport signal/bruit (SNR) et l'amplitude du vecteur d'erreur peut être exprimé comme suit :

$$EVM = \sqrt{\frac{1}{SNR}} \quad [2.18]$$

La figure 2.12 montre le circuit Simulink qui correspond à l'estimation du rapport signal bruit au récepteur. Lors de la réception d'un symbole OFDM, ce dernier est capturé

puis passé vers le modulateur pour extraire les places idéales des symboles modulés portés par le symbole OFDM. Ensuite, une soustraction vectorielle entre les symboles idéaux et les symboles reçus, est accomplie. Pour calculer l'amplitude du vecteur d'erreur d'un symbole OFDM, la racine carrée moyenne de l'EVM des symboles modulés portés par le symbole OFDM est calculée. Finalement, le rapport SNR est calculé en utilisant l'équation [2.18].

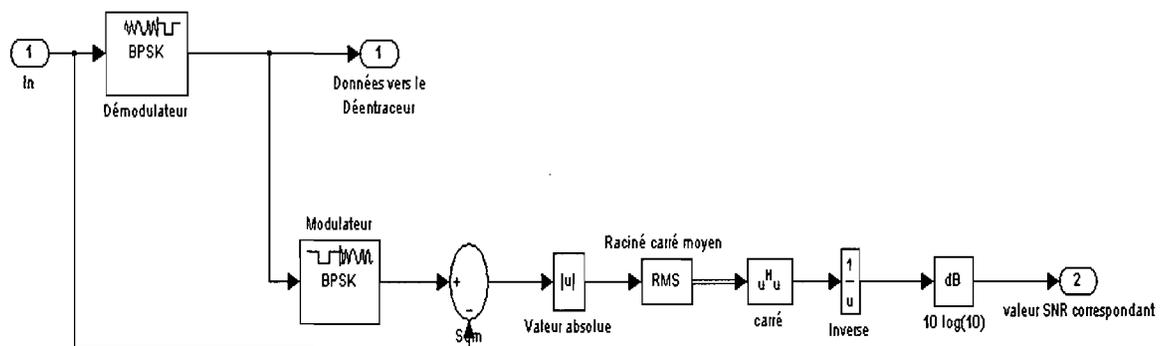


Figure 2.12 : Estimation du rapport signal/bruit dans Simulink.

2.3.3 Le décodeur de Viterbi

Pour effectuer un décodage optimal, Viterbi a proposé un algorithme de décodage pour les codes convolutifs en 1967 [32]. Le décodeur de Viterbi utilise l'algorithme de Viterbi pour le décodage d'une séquence codée avec un encodeur convolutionnel. En effet, plusieurs algorithmes ont été proposés dans la littérature pour le décodage d'une séquence codée convolutionnellement, cependant l'algorithme de Viterbi est le plus populaire et performant [33]. Le seul inconvénient du décodeur de Viterbi réside dans le grand nombre des ressources consommées par une implémentation matérielle. L'étude théorique, l'analyse détaillée et la performance de l'algorithme de Viterbi sont largement couverts dans divers livres et articles [32-34]. La figure 2.13 illustre l'implémentation du démodulateur (QPSK), déentrelaceur, décodeur de Viterbi et décodeur Reed-Solomon en Simulink.

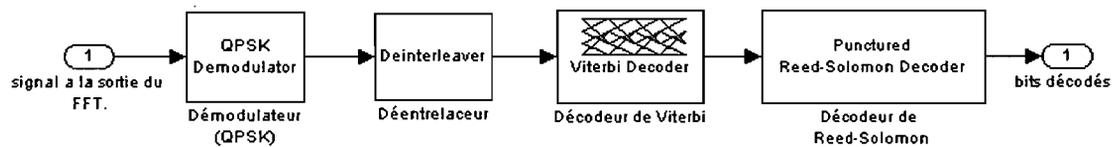


Figure 2.13 : Démodulateur, déentrelaceur, décodeur de Viterbi et décodeur RS en Simulink.

Comme on peut le voir dans la figure 2.13, après la transformée de Fourier, le signal obtenu est démodulé pour récupérer les bits correspondants. En deuxième lieu, la séquence binaire obtenue est désentrelacée pour réorganiser les bits selon leurs indices originaux (avant l'entrelacement). Ensuite, la séquence binaire est traitée par le décodeur de Viterbi qui utilise les bits de redondance ajoutés à l'émetteur pour détecter et corriger les bits erronés. Finalement, la sortie du décodeur de Viterbi est passée vers le décodeur de Reed-Solomon qui utilise les octets de redondance ajoutés à l'émetteur pour corriger les erreurs de transmission.

2.4 Résumé du chapitre

Dans ce chapitre, nous avons présenté tous les modules de la couche physique de WiMax. La théorie et le rôle de chaque module au niveau de l'amélioration de la performance de WiMax ont été présentés et discutés. De plus, nous avons abordé brièvement l'implémentation de ces modules en Simulink. Les résultats de simulation du modèle implémenté en Simulink seront présentés dans le chapitre 5. L'implémentation de ces modules en matériel sera présentée dans le chapitre 6.

Chapitre 3 :

Revue de la littérature

Ce chapitre est conçu pour fournir au lecteur un bref aperçu sur les recherches pertinentes qui se font au niveau de l'implémentation et la réalisation des systèmes basés sur le multiplexage par répartition orthogonale de la fréquence (OFDM) sur du matériel reconfigurable. Ceci n'est pas une liste exhaustive de toutes les recherches en cours, mais elle devrait donner au lecteur un aperçu de la façon dont ce mémoire s'intègre avec la communauté de recherche en cours. De l'autre côté, la section 3.9 de ce chapitre introduit brièvement l'efficacité de l'encodeur Reed-Solomon ainsi que l'encodeur convolutionnel au niveau de l'amélioration de la performance des systèmes de communication sans fil basés sur le multiplexage par répartition orthogonale de la fréquence.

3.1 Université Guelph

A. Sghaier et al. présentent une pure conception VHDL, intégrée avec certaines propriétés intellectuelles (IP) pour l'implémentation d'un émetteur OFDM conformément à la norme IEEE 802.11a [21]. Le modèle conçu est bien pipeliné. Le pipelining a été principalement réalisé en doublant les éléments de mémoire (RAM ou registres). Cette architecture permet de garder le nouveau flux de bits dans le tampon pendant que le flux précédent est en cours de traitement. La conception a été réalisée en utilisant la plateforme matérielle Virtex-II de Xilinx. Le premier bloc réalisé est le mélangeur qui est suivi par l'encodeur convolutionnel (l'encodeur Reed Solomon n'est pas implémenté). L'encodeur convolutionnel est suivi par l'entrelaceur qui est constitué d'une RAM où les données sont écrites en lignes, et ensuite lues en colonnes. Avant de générer le signal OFDM, les bits sont modulés en utilisant la technique de modulation appropriée (BPSK, QPSK, 16-QAM

ou 64-QAM). Ces techniques de modulation sont implémentées en utilisant quatre mémoires ROMs. Finalement, le module IFFT avec 64 points a été implémenté en utilisant la propriété intellectuelle (IP) de Xilinx. Les auteurs mentionnent les résultats obtenus et les comparent avec ceux de [18] (voir tableau 3.1). La conception proposée par [21] a été conforme avec les débits de données obligatoires spécifiés dans le standard (6, 12 et 24 Mbits/s).

Tableau 3.1: Résultats de l'université de Guelph vs. INAOE

Ressources	travail de [21]	travail de [18]
Tranches (Slices)	1828	1924
Tranche Flip-Flops	2840	2367
LUTs	2805	2500
BRAMs	12	12
Multiplieurs 18*18	12	0

3.2 L'académie chinoise des sciences

Le travail de Xu Jinsong et al. se concentre sur la conception d'un émetteur OFDM compatible avec les systèmes de communication UWB (Ultra Wideband Communication) [15]. Les fonctionnalités réalisées sont respectivement un mélangeur, un convolveur de taux 1 sur 3, un perforateur et un entrelaceur. La technique de modulation utilisée est le QPSK, la modulation adaptative n'est pas employée. Et finalement, la taille de la transformée de Fourier FFT utilisée est 128. Selon les auteurs, cette taille est optimale pour les systèmes UWB.

Les auteurs présentent les résultats de simulation de chaque bloc à part, la plateforme matérielle utilisée pour la synthèse est le Virtex II de Xilinx. Le résumé de l'utilisation du matériel est encore présenté. Cependant, aucune contrainte de temps n'est mentionnée.

3.3 Institut national d'astrophysique, d'optique et d'électronique (INAOE, Mexico)

G. Garcia et al. présentent la conception, la validation et la mise en œuvre d'un modulateur OFDM conforme avec le standard IEEE 802.11a [18]. Le modèle a été réalisé

en utilisant les outils de conception de haut niveau comme le System Generator for DSP (XSG) de Xilinx, MatLab et Simulink. Les modules implémentés sont le modulateur (BPSK, QPSK, 16-QAM et 64-QAM), la transformée inversée de Fourier de taille 64 points et le préfixe cyclique. La plateforme matérielle utilisée est le virtex-II, selon l'auteur seulement 10% des ressources disponibles ont été utilisées. La fréquence d'opération de l'architecture proposée était d'environ 92 MHz. La consommation d'énergie pour cette architecture est de 9,06 mW / MHz, cette valeur a été estimée en utilisant l'outil XPower de Xilinx.

Dans une autre publication [19], Garcia et al. présentent la conception, la validation et la mise en œuvre d'un modulateur OFDM conformément à la norme IEEE 802.16, ce modulateur a été complètement conçu en utilisant l'outil de conception de haut niveau System generator for DSP de Xilinx. Les modules implémentés sont respectivement le modulateur (QPSK, 16-QAM, 64 QAM), l'IFFT et le préfixe cyclique. La plateforme de synthèse utilisée est le Virtex-II, les résultats obtenus montrent que 18% des ressources disponibles ont été utilisées et la fréquence maximale atteinte est de 98.376 MHz (fréquence d'opération de la conception proposée).

3.4 La société ALTERA

La société ALTERA décrit dans [22] comment utiliser ses FPGAs et ses outils de conception disponibles pour surmonter les défis associés à la mise en œuvre des couches PHY et MAC de WiMax. Le rapport met en lumière la capacité des FPGAs d'ALTERA ainsi que les propriétés intellectuelles disponibles. Le rapport souligne l'abondance des blocs arithmétiques et mémoires dans les FPGAs d'ALTERA. De plus, ALTERA suggère l'utilisation de la famille des FPGAs startix-II qui contient plus de 384 18*18 multiplicateurs. Selon ALTERA, startix-II est la meilleure option pour la mise en œuvre des couches PHY et MAC des normes de communication récentes, y compris WiMax. Pour réduire le temps de la mise sur le marché, ALTERA suggère l'utilisation de ses propriétés

intellectuelles (FEC, FFT, MIMO) ainsi que les outils de conception de haut niveau tels qu'ALTERA DSP builder.

3.5 Lattice Semiconductor UK Ltd

Lattice Semiconductor a produit plusieurs papiers, y compris [16] et [17] qui explorent les problèmes majeurs dans la conception et l'implémentation de la norme 802.16. Lattice suggère l'utilisation de la famille des FPGAs Lattice-ECP33 pour la réalisation de WiMax. De plus, Lattice montre l'importance de la suite logicielle ispLEVER pour effectuer toutes les tâches requises pour la réalisation des couches PHY et MAC de WiMax, y compris la conception, la simulation, la synthèse, et la programmation finale du FPGA. D'autre part, Lattice propose un certain nombre de blocs de propriété intellectuelle tels que l'encodeur/décodeur Reed-Solomon, le décodeur de Viterbi, et la transformée de Fourier qui peuvent être utilisés pour accélérer le processus de développement. L'étude fournie par Lattice Semiconductor est très intéressante, elle montre la possibilité et la faisabilité de l'utilisation de la famille des FPGA Lattice ECP pour la conception des systèmes de communication sans fil basés sur le multiplexage par répartition orthogonale de la fréquence (OFDM).

3.6 Innovative radio Systems IAF, Allemande

IAF est une société allemande spécialisée dans le domaine de la communication numérique sans fil [24]. Ils ont implémenté plusieurs bancs d'essai OFDM sur un FPGA, y compris un qui est basé sur la norme IEEE 802.11a. Une grande partie de leur travail se base sur les systèmes OFDM pour la technologie 4G, y compris un système qui prétend détenir le record du monde sur la vitesse de transmission radio, avec un débit de données d'environ 1 Gigabit par seconde.

3.7 Université de Kansas

Jordan Guffey à l'université de Kansas a introduit la conception et l'implémentation d'un système de multiplexage par répartition orthogonale de la fréquence (OFDM) en se basant sur la norme IEEE 802.16 2004 [23]. Ce système se compose d'un émetteur et d'un récepteur qui ont été implémentés en VHDL pour les utiliser sur le « Kansas University Agile Radio » (KUAR). Le KUAR est une plateforme expérimentale SDR (Software Defined Radio) qui est capable de la mise en œuvre des composants radio en logiciel ou/et en matériel reconfigurable. Le modèle conçu s'intéresse à l'implémentation d'un émetteur-récepteur OFDM, sans se concentrer sur les codes correcteurs et l'entrelacement. Cependant, le système conçu s'intéresse plus aux modules de synchronisation entre l'émetteur et le récepteur comme l'estimation du canal, la synchronisation des trames, et la détection des paquets. La conception VHDL a été vérifiée et validée avec une conception MatLab. Les résultats mentionnés montrent que l'implémentation matérielle atteint des résultats similaires à ceux de la conception Matlab.

3.8 Contribution par rapport aux travaux précédemment cités

Dans les sections précédentes, nous avons présenté quelques recherches pertinentes au niveau de la conception et la mise en œuvre des systèmes OFDM reconfigurables. Dans cette section, nous allons mettre l'accent sur notre contribution par rapport aux travaux précédemment cités. Tout d'abord, nous remarquons qu'au niveau de la mise en œuvre des systèmes OFDM, en particulier la couche physique de WiMax, la plupart des ouvrages publiés se concentrent sur certaines fonctions et ne traitent pas tous les modules de la couche physique, ceci est dû à la complexité de cette couche. Dans ce contexte, dans [21] et [15], seulement les modules de la chaîne de transmission ont été réalisés et le langage VHDL a été utilisé pour l'implémentation. Alors que, dans ce mémoire nous présentons la conception de tous les modules de la couche physique de WiMax (chaîne de transmission et de réception) en utilisant l'outil de conception de haut niveau System Generator for DSP. De plus, le travail de [21] et [15] ciblent respectivement les technologies WiFi et UWB. Par contre, nous ciblons la technologie WiMax. En fait, la recherche effectuée par Garcia [19],

est la plus liée à la notre. Cependant, [19] se concentre sur la conception d'un émetteur WiMax non codé. De plus, [19] n'a pas implémenté les modules de la chaîne de réception. Cependant, notre travail présente tous les modules de la chaîne de transmission ainsi que ceux de la chaîne de réception, y compris les modules du codage du canal. Finalement, [23] traite les problèmes de synchronisation entre l'émetteur et le récepteur dans WiMax. Cependant, dans ce travail nous avons supposé une synchronisation parfaite entre l'émetteur et le récepteur. Ainsi, les modules de synchronisation entre l'émetteur et le récepteur tels que la synchronisation des trames et la détection des paquets ne seront pas traités dans ce mémoire.

3.9 Performance et efficacité de l'encodeur Reed-Solomon et l'encodeur convolutionnel

Presque tous les systèmes de communication numérique bénéficient d'une forme de correction d'erreur directe. Dans ce contexte, il existe généralement quatre types des codes correcteurs employés dans les systèmes de communication filaires et sans fils : les codes LDPC (low-density parity-check codes), les codes Turbo, les codes Reed-Solomon et les codes convolutionnels. Récemment, des efforts de recherche considérables ont porté sur deux aspects de ces codes correcteurs : (1) l'exploration et l'évaluation des performances de ces codes dans des conditions différentes et pour différentes applications et (2) l'optimisation et le pipelining de l'implémentation matérielle de ces codes (ainsi que leurs modules de décodage correspondants).

À ce propos, puisque la norme de WiMax spécifie que, pour accomplir la correction d'erreur directe (FEC), l'encodeur Reed-Solomon (RS) doit être concaténé avec l'encodeur convolutionnel (CC), nous allons mener une brève revue sur les travaux qui ont porté sur l'efficacité des codes RS et CC au niveau de l'amélioration de la performance des systèmes à base OFDM. Une note est à signaler avant de commencer notre revue, c'est que la performance des codes correcteurs d'erreurs dépend fortement de plusieurs paramètres tels que le schéma de modulation employé, le taux de codage, la taille du bloc, le polynôme

générateur, la longueur du code et le type de canal de communication. Ainsi, des configurations différentes de ces paramètres sont utilisées dans les systèmes de communication conformément aux spécifications de la norme ou selon l'application en question [68].

Dans ce contexte, une étude intéressante qui traite l'effet des codes correcteurs d'erreurs (en particulier RS et CC) dans les systèmes de communication à base OFDM a été faite par Elkhettabi dans [42]. L'auteur mentionne qu'en utilisant la structure de modulation QPSK avec un encodeur convolutionnel (CC), le taux d'erreur binaire de 10^{-5} a été atteint à un rapport signal / bruit (SNR) équivalent à 10.3 dB. De même, [42] signale qu'un gain de 0.2 dB a été observé lorsque le CC a été remplacé par le code Reed Solomon RS (15,11). Ce gain a été réduit à 0.15 lors de l'utilisation de code RS (15,9) (voir figure 3.1). D'autre part, l'auteur explore l'idée de concaténer deux encodeurs convolutionnels en série. Cependant, l'amélioration dans la performance était négligeable (environ 0.3 dB) et la complexité du système a considérablement augmenté. Par conséquent, la combinaison de deux codes CC en série était injustifiée. De plus, dans sa conclusion, Elkhettabi affirme que les codes de Reed-Solomon améliorent la performance d'un système de communication d'environ 0.3 dB par rapport aux codes convolutionnels classiques.

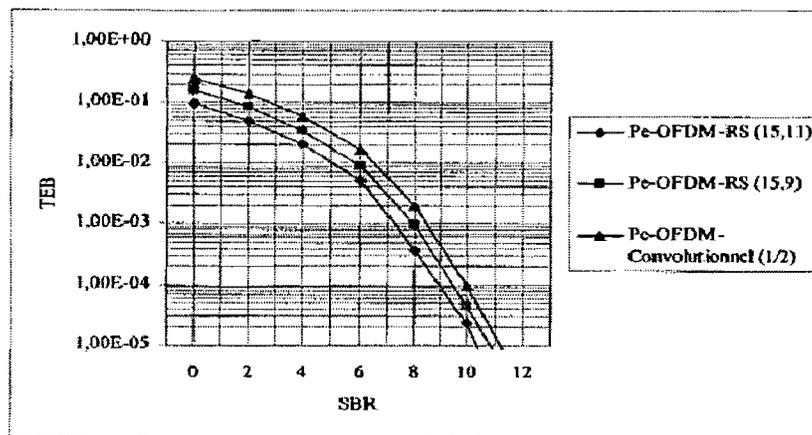


Figure 3.1: Performance des codes : CC, RS(15,11) et RS (15,9) dans un système à base OFDM

Le travail de H. Zhou et al. [70] se concentre sur l'étude de la performance de la technique de correction d'erreur directe dans les systèmes UWB (ultra wide band communication). Similairement à WiMax, les systèmes UWB emploient un code RS(255,233) concaténé avec un code CC pour la correction d'erreur directe. Comme le montre la figure 3.2, la performance du système avec le code RS et le code CC s'est améliorée par rapport au cas sans codage. De même, on peut observer que le code RS donne plus de performance en comparaison avec le code CC. Cependant, comme prévu, la combinaison de RS et CC fournit la meilleure performance. Dans ce contexte, à un taux d'erreur binaire égale à 10^{-2} , la combinaison du code RS et CC fournit une amélioration dans la performance de l'ordre de 6 dB par rapport au cas sans codage. Cette amélioration a été réduite à environ 4 dB lorsque le code CC a été enlevé (RS seulement). De même, l'amélioration a été réduite à environ 3.5 dB lorsque le code RS a été enlevé (CC seulement).

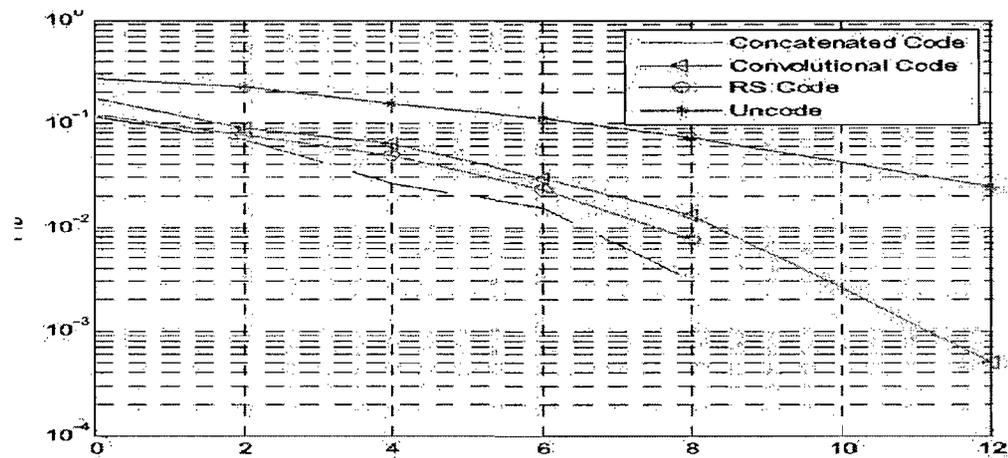


Figure 3.2 : Performance de RS et CC dans les systèmes UWB.

Haque et al. dans [69] traite l'effet du taux de codage dans un encodeur convolusionnel sur la performance globale des systèmes à base OFDM. Les auteurs proposent un système de communication constitué d'un code RS (255, 239) concaténé avec encodeur convolusionnel. Plusieurs taux de codage pour le CC ont été employés. Dans le cas d'un taux de codage égale à 1/2, la performance du système avec la structure de modulation

BPSK était tout à fait satisfaisante. Cependant, les performances atteintes pour les structures de modulation QPSK, 16QAM et 64QAM n'étaient pas efficaces. Ainsi, les auteurs suggèrent que l'utilisation d'un taux de codage équivalent à $1/2$ soit limitée à la structure de modulation BPSK. De même, les auteurs signalent que la performance obtenue dans le cadre d'un taux de codage équivalent à $2/3$ avec la structure de modulation 64QAM n'était pas fiable. Par conséquent, les auteurs ne recommandent pas l'utilisation de ce taux de codage avec la structure de modulation 64QAM. Aussi, il a été observé d'après les résultats de la simulation que la performance du système a augmentée avec un taux de codage et une longueur du code plus élevés.

Une dernière contribution dans le contexte de la performance des codes correcteurs a été faite par S. G. Lee dans [71]. L'auteur traite l'évaluation des performances des codes correcteurs concaténés (RS, CC et LDPC) dans les systèmes à base OFDM. Dans son étude, l'auteur considère le code RS (255,239,8) ainsi que l'encodeur convolutionnel de taux $2/3$. La figure 3.3 montre la performance du code RS par rapport à la concaténation de RS et CC. Comme prévu, sans l'encodeur convolutionnel, une dégradation dans la performance d'environ 3 dB (à un taux d'erreur binaire égale à 10^{-2}) a été observée par rapport au cas où le CC a été combiné avec l'encodeur RS.

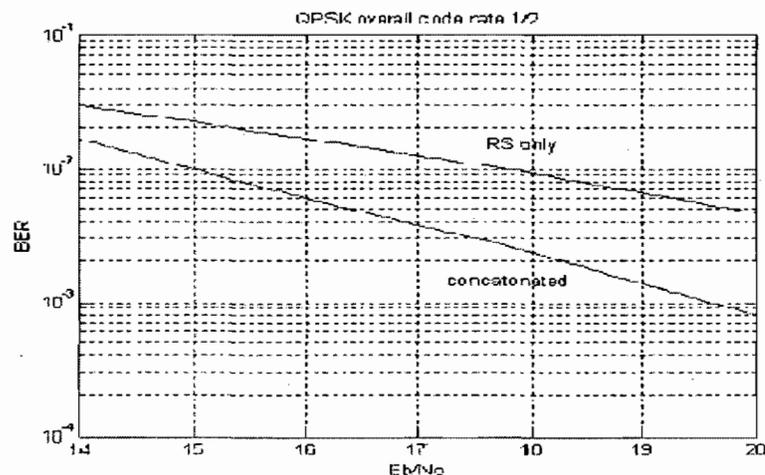


Figure 3.3 : Performance de l'encodeur RS par rapport à la combinaison de RS et CC.

Chapitre 4 :

Technologie de réalisation : Plateforme matérielle et outils de conception

4.1 Introduction

Ce chapitre présente un aperçu général sur les matériels reconfigurables (FPGA), ainsi que les contraintes à respecter lors de la réalisation d'un design sur des tels matériels. Nous abordons également le flux de la conception des systèmes numériques en utilisant la plateforme de prototypage rapide Xilinx System Generator for DSP (XSG) qui est une partie de la plateforme de développement de Simulink. De plus, nous allons justifier les raisons pour lesquelles nous avons choisi la famille des FPGAs « VIRTEX II pro » de XILINX comme plateforme de réalisation. L'utilisation de l'environnement de prototypage rapide XSG, comme plateforme de réalisation, sera aussi justifiée. De plus, les différences principales entre la modélisation d'un design avec la plateforme Simulink de MATLAB (blocksets standards de Simulink) et la réalisation de ce design sur un FPGA en utilisant XSG (blocksets de XILINX) seront mises en évidence.

4.2 Les contraintes de l'implémentation matérielle

Contrairement à la modélisation d'un design avec un logiciel de modélisation (par exemple Simulink, ns2 ou c++), la réalisation de ce design sur une plateforme matérielle reconfigurable implique quelques contraintes. En effet, l'espace disponible sur le FPGA n'est pas illimité. Ceci impose la minimisation de l'espace occupé par le circuit. D'autre part, la fréquence de fonctionnement du circuit conçu peut être réduite à cause des délais

introduits par les divers éléments logiques, mémoires et interconnexions du circuit. D'où la nécessité d'introduire une plus grande quantité de parallélisme dans la conception pour atteindre une fréquence d'opération maximale. Ces deux contraintes (l'espace occupé par le circuit et la fréquence de l'opération du système conçu) imposent une très bonne expertise au niveau de l'architecture et des ressources des FPGA, ainsi qu'au niveau des outils de conception utilisés pour le prototypage du système.

Dans la section suivante, nous allons présenter brièvement l'architecture et le fonctionnement des FPGAs. Plus tard dans ce chapitre, nous allons décrire le flux de conception des systèmes numériques en utilisant la plateforme de prototypage rapide Xilinx System Generator for DSP.

4.3 Généralités sur les FPGAs

Le dernier né de la famille ASIC (Application Specific Integrated Circuit) est le FPGA (Field Programmable Gate Array). Les FPGAs, apparus commercialement en 1985, sous l'initiative de la société XILINX, sont des circuits numériques à logiques programmables, c.à.d. pouvant être programmés par le concepteur plutôt que par le fabricant du périphérique pour exécuter une fonction logique.

L'architecture des FPGAs est constituée d'un tableau de blocs logiques configurables (configurable logic blocks, CLB). Ces blocs logiques permettent de réaliser des fonctions combinatoires et séquentielles (voir figure 4.1). Autour de ce tableau de CLBs, on trouve des blocs d'entrées/sorties (Input/output, IOs). Ces blocs sont responsables de l'interfaçage avec le monde externe. Chaque bloc configurable est aussi entouré par une matrice de connexion qui permet de router les signaux entre les différents CLBs. La programmation du FPGA est un processus très simple. Ce processus consiste à charger une trame de bits dans les cellules de mémoire SRAM pour établir les différentes connexions entre les CLBs et les IOBs ainsi qu'entre les CLBs eux même.

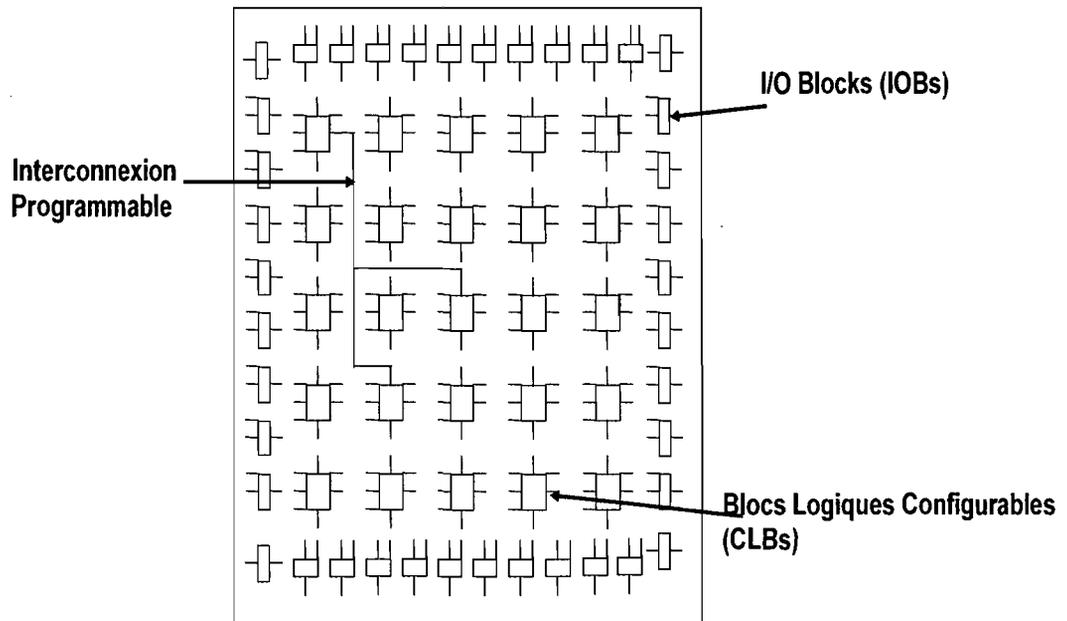


Figure 4.1 : Architecture générale des FPGAs.

4.4 Avantages de l'utilisation des FPGAs

Bien que les matériels reconfigurables sont souvent utilisés comme des circuits de prototypage avant une implémentation définitive sur des circuits dédiés. L'utilisation des FPGAs offre aux ingénieurs plusieurs avantages par rapport à la fabrication d'un circuit intégré dédié. Parmi ces avantages, on identifie une plus grande flexibilité, une diminution du temps globale de conception, et une réduction du coût globale de conception. D'autre part, contrairement à un microprocesseur ou un processeur DSP où la performance est liée à la fréquence maximale à laquelle le processeur peut fonctionner, la performance du FPGA est liée à la quantité du parallélisme qui peut être introduite dans la conception. Cette dernière caractéristique des FPGAs permet de construire des modules plus rapides et à plus haute intégration, ouvrant ainsi la voie au développement des applications importantes comme le traitement des signaux, le codage et le décodage, la reconnaissance des formes et la cryptologie [53].

4.5 Choix de la technologie d'implémentation

Sur le marché des matériels reconfigurables, on trouve plusieurs fournisseurs, tels qu'ALTERA, ACTEL, LATTICE et XILINX. Cependant, les familles des FPGAs de XILINX sont les plus utilisées en raison de leurs caractéristiques et leurs performances [47]. En effet, La réalisation de notre projet est faite sur des composantes de XILINX. Ce choix est dicté non seulement par la large utilisation des FPGAs de XILINX au niveau commercial, mais aussi par l'expertise développée au sein du laboratoire LASSO ainsi que la disponibilité des outils matériels et logiciels nécessaires pour la réalisation des différents projets. Donc, au niveau du matériel nous avons utilisé la série des FPGAs Virtex II pro XC2VP30 de XILINX. Ce dernier (Virtex II pro XC2VP30) a été choisi grâce à sa capacité au niveau CLBs, ainsi que ses multiples ressources de routage et de calcul arithmétique. D'autre part, au niveau des logiciels et outils de conception, nous avons utilisé le logiciel Xilinx System Generator for DSP (XSG). Ce dernier a été choisi parce qu'il offre un environnement de prototypage rapide, simple et efficace pour la conception des systèmes numériques. XSG et « Virtex II pro XC2VP30 » seront présentés en détail dans les sections suivantes.

4.6 La famille des FPGAs Virtex

En 1998, l'entreprise XILINX a introduit une nouvelle famille des FPGAs nommée « VIRTEX ». Cette nouvelle famille est basée sur un procédé *CMOS* $0.2 \mu m$ avec cinq couches de métal [54]. En effet, La série des FPGAs « VIRTEX » a été la première qui offre l'équivalent de 1 million de portes logiques au niveau de densité d'intégration. La série « VIRTEX » est caractérisée par les propriétés suivantes : une fréquence d'horloge pouvant aller jusqu'au 500 MHz, des DLLs intégrés (Digital Locked Loop) qui permettent un contrôle plus précis d'horloge, et les modules « VersaRing » qui permettent d'isoler les portes entrées/sorties de la matrice des CLBs et fournissent à l'utilisateur plus de flexibilité lors de l'affectation des portes d'entrées/sorties [59]. D'autre part, la série des FPGAs VIRTEX est équipée avec des mémoires RAMs dédiées, ainsi que des modules logiques

dédiées pour l'exécution rapide des opérations arithmétiques (multiplication et division). Un autre avantage de la série VIRTEX est le grand nombre des ports d'entrées/sorties qui permettent de supporter une grande variété de standards de communication.

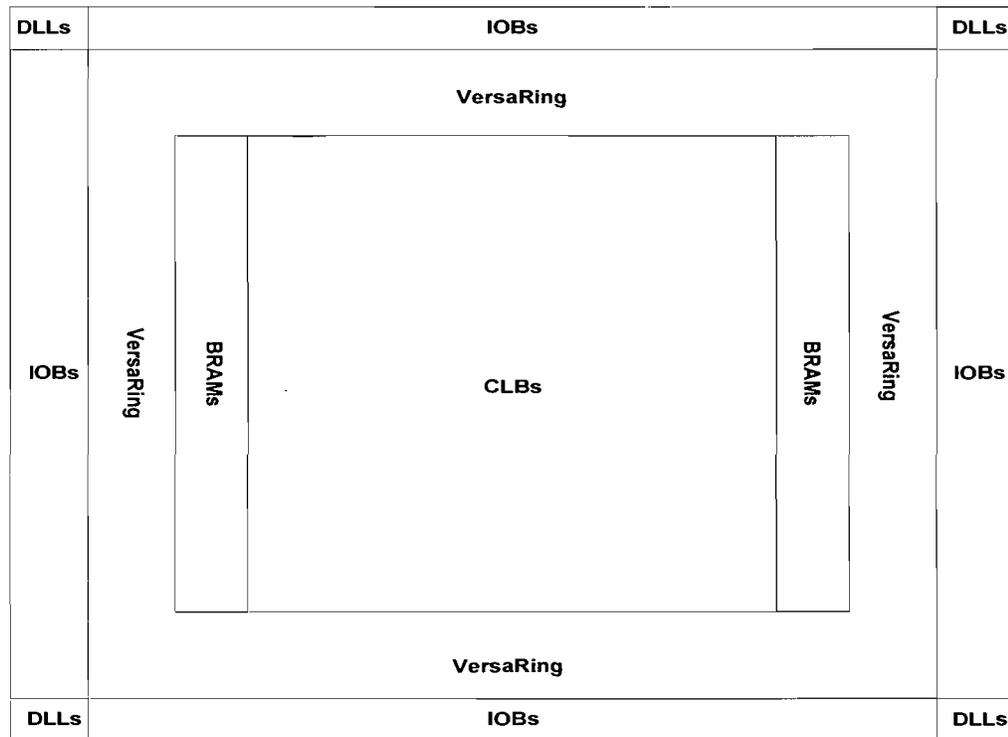


Figure 4.2 : Architecture générale du FPGA VIRTEX

En 2002, XILINX a introduit la série VIRTEX II pro. Contrairement à la première série de VIRTEX, VIRTEX II pro est basée sur un procédé *CMOS* $0.13 \mu m$. Les statistiques concernant la densité et l'architecture de la plateforme Virtex II pro XC2VP30 sont présentées dans le tableau suivant :

Tableau 4.1: Statistique de la plateforme Virtex II pro XC2VP30

Propriété	Valeur
Tranches (slices)	13,696
Cellules logiques (1 LUT à 4 entrées + bascule)	30,816
blocs RAMs de 18 Kbits	136
Multiplieurs dédiées (18*18)	136
Blocs entrées/sorties	624
PowerPC Processeur	2
Fréquence de l'horloge	430 MHZ

En effet, l'architecture de la série des FPGAs VIRTEX II pro est optimisée pour supporter les applications qui nécessitent une grande vitesse d'opération avec une faible consommation d'énergie [52]. Ceci les rend un choix optimal pour les applications qui nécessitent un traitement en temps réel, comme le traitement des signaux, la cryptologie, et la communication sans fil.

4.7 Xilinx System Generator for DSP (XSG)

Bien que les langages de description des matériels récents, comme le VHDL et le VERILOG, simplifient le cycle de développement des systèmes numériques, l'implémentation et la réalisation d'une conception sur une plateforme matérielle réelle reste toujours un cauchemar face aux ingénieurs [49-50]. En effet, l'utilisation des approches de conception classique pour l'implémentation des systèmes numériques sophistiqués est indésirable. Ce type d'approches rend le cycle de développement d'un projet long et complexe [50]. D'où la nécessité d'une plateforme de prototypage rapide qui permet aux ingénieurs de mettre l'accent sur la manière de concevoir le système, au lieu de se concentrer sur l'écriture du code VHDL et le débogage du code généré. Pour ces raisons, XILINX a introduit la plateforme SYSTEM GENERATOR FOR DSP (XSG).

En effet, XSG n'est pas un programme en soi. Cependant, il s'agit plutôt d'une librairie qui étend l'environnement SIMULINK de MATLAB. En effet, c'est l'intégration avec

SIMULINK qui fait la force de XSG. Les ingénieurs, qui utilisent SIMULINK pour modéliser et évaluer la performance d'une conception, peuvent facilement synthétiser leur conception vers un FPGA, en utilisant les blocs fournis par XILINX.

Évidemment, il y a plusieurs contraintes à respecter et la méthodologie de design au niveau de XSG n'est pas exactement la même que celle de SIMULINK. Par exemple, en Simulink les opérations arithmétiques sont effectuées en point flottant de 64 bits, alors qu'au niveau de XSG, ces opérations sont effectuées en point fixe. De plus, en SIMULINK, le traitement et le calcul des signaux se fait à base matricielle, alors qu'avec XSG ce traitement se fait à base scalaire. Finalement, les blocs fournis par XSG ne permettent pas de concevoir toutes les fonctions voulues, ce qui limite la flexibilité du designer. XSG sera présenté en détail dans la section suivante.

4.7.1 Fonctions disponibles dans XSG

XSG fournit une librairie des blocs synthétisables, optimisés, simples, flexibles, efficaces et personnalisables. Ces blocs abstractent l'implémentation des fonctions mathématiques et logiques complexes (sinus et cosinus, racine carrée, fonctions logarithmiques, machines à états...), ainsi que certains algorithmes du traitement des signaux et de communication sophistiquée (transformée de Fourier, encodeur Reed Solomon, décodeur de Viterbi...). La majorité de ces blocs imitent des fonctions déjà présentes dans SIMULINK. Cependant, l'avantage majeur des blocs fournis par XSG est que, contrairement aux blocs standards de SIMULINK, ils peuvent être synthétisé vers un matériel reconfigurable. En effet, les blocs du XSG sont groupés, selon leurs fonctions, en 8 sections : les éléments de base, les modules de communication, les modules du traitement des signaux, les fonctions logiques, les types des données, les modules mathématiques, les mémoires et les outils divers.

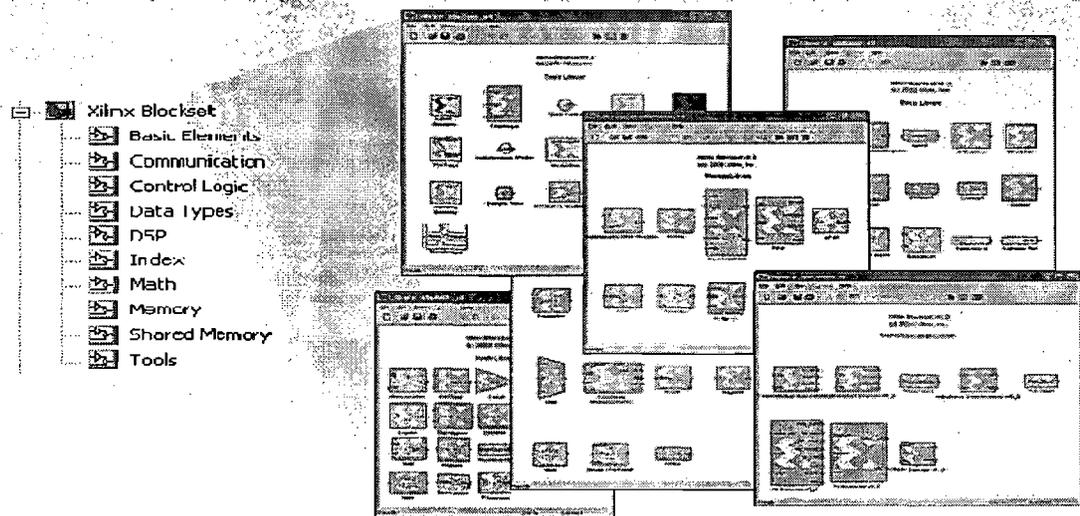


Figure 4.3: Xilinx System Generator for DSP

Les éléments de bases comprennent des fonctions simples dont les noms décrivent bien leurs fonctions. Cependant, Il y en a trois qui sont particulier et sont les plus utilisés. Le bloc « system generator » sert à transformer le modèle conçu en code VHDL (ou VERILOG) synthétisable et de niveau RTL. Les deux autres blocs sont le « Gateway in » et le « Gateway out ». Le bloc « Gateway in » est utilisé pour passer du domaine continu en virgule flottante au domaine en temps discret à point fixe. « Gateway in » permet de spécifier la fréquence à laquelle le signal sera échantillonné, ainsi que la précision et le type de donnée (data type) du signal. De l'autre coté, le « Gateway out » reconvertit automatiquement les valeurs en point flottant ce qui permet de les reconnecter aux blocs standards de SIMULINK. La section mémoire contient des éléments de mémoire typiques comme les FIFOs, les RAMs et les ROMs. Les librairies de traitement des signaux et de communication contiennent des modules qui abstractent l'implémentation des algorithmes sophistiqués comme la transformée de Fourier, l'encodeur/décodeur Reed Solomon, le décodeur de Viterbi etc. La librairie « outils » contient des modules à utilisations générales comme l'estimateur des ressources d'un circuit et le microcontrôleur PicoBlaze.

4.7.2 Concept de l'horloge dans XSG

Contrairement aux autres blocs de SIMULINK, Les blocs de XSG fonctionnent uniquement en temps discret. Ceci est pour imiter l'architecture des FPGAs qui ne peuvent pas fonctionner en temps continu. Donc, pour représenter l'horloge matérielle du FPGA dans Simulink, XSG introduit le concept de la période d'échantillonnage (sampling period). Si par exemple, la période d'échantillonnage d'un bloc est égale à un, la sortie de ce bloc n'est calculée qu'une seule fois, avec les valeurs présentés à l'entrée, au début de chaque cycle de l'horloge matérielle. Par contre, si la période d'échantillonnage d'un bloc est égale à un entier m , la sortie de ce bloc n'est calculée qu'après m cycles de l'horloge matérielle.

Un autre concept essentiel au niveau de XSG est la latence, c'est une propriété qu'il faut spécifier pour certains blocs. En effet, si dans un circuit la latence de tous les blocs est mise à zéro, le circuit est totalement combinatoire. Ceci fait en sorte que l'algorithme implémenté en FPGA donnera un résultat en un seul cycle d'horloge. Cependant, la présence de logique combinatoire, sans des éléments de latence, limite énormément la fréquence possible de l'horloge.

4.7.3 Simulation des modèles dans XSG

La simulation des circuits composés des blocs de Xilinx est très simple. En effet, les blocs de XILINX se comportent comme les autres blocs de SIMULINK, sauf qu'ils sont toujours en temps discret. Cependant, le temps de simulation d'un circuit construit avec les blocs de XILINX est beaucoup plus élevé par rapport à un circuit identique composé des blocs standards de SIMULINK.

Un autre avantage de l'utilisation de XSG est que ce dernier garantit que lorsqu'une conception est implémentée sur FPGA, le circuit se comportera exactement comme lors de la simulation du modèle dans SIMULINK [47].

4.7.4 Précision du calcul dans XSG

Au niveau du MATLAB et SIMULINK les calculs sont effectués en point flottant de 64 bits. Cette propriété de MATLAB fournit une grande précision et une plage de valeurs énorme. Du côté théorique, les FPGAs peuvent avoir la même précision, cependant le nombre d'unités de calcul serait réduit, car les opérations en point flottants sur 64 bits nécessitent un grand nombre de logique. Pour cela, XSG utilise une représentation en point fixe. Cette représentation prend la forme suivante : (M.N) où M bits sont utilisés pour représenter la partie réelle et les N bits sont utilisés pour représenter la fraction. Plus la précision est élevée, plus la quantité de portes logiques nécessaires est grande. Il faut donc toujours établir un compromis entre la précision et l'espace nécessaire.

4.8 Le flux de conception VLSI traditionnel

La figure 4.4 montre le flux de conception VLSI traditionnel. Dans cette approche plusieurs plateformes sont utilisées pour compléter le cycle de développement d'un projet. Tout d'abord, la modélisation et l'évaluation de la performance du design est généralement faite en utilisant un logiciel ou un système de simulation (par exemple Simulink ou C++). Au niveau de l'implémentation, les ingénieurs doivent transférer vers une autre plateforme (par exemple LABVIEW, ACTIVE HDL) pour l'écriture et la simulation du code HDL. Ce code correspond au modèle déjà conçu. De plus, la synthèse du code et la vérification sur puce nécessitent, parfois, une troisième plateforme. Ceci rend le processus de conception plus long et complexe. Ainsi, une grande équipe avec une très bonne expertise sera nécessaire pour la conception et la réalisation du système. De même, le temps et le coût de développement seront plus élevés.

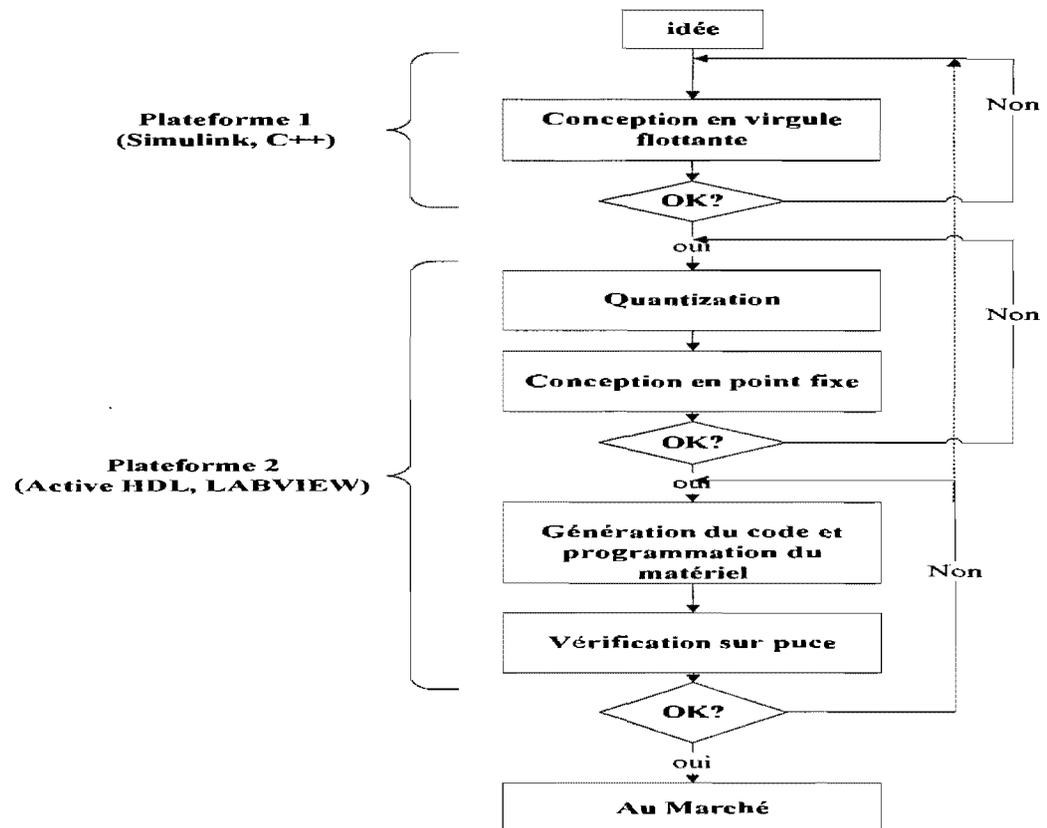


Figure 4.4 : Flux de conception VLSI conventionnel.

4.9 Avantages et flux de conception de XSG

Les ingénieurs de conception favorisent l'utilisation de XSG par rapport aux plateformes traditionnelles pour les raisons suivantes :

- XSG fournit une plateforme unique pour l'analyse, la conception, l'implémentation et la vérification du système.
- XSG fournit aux ingénieurs une plateforme de prototypage rapide qui permet aux ingénieurs de mettre l'accent sur la manière de concevoir le système, au lieu de se concentrer sur l'écriture de code VHDL.
- Le modèle conçu avec les blocs de XSG est traduit en langage de description du matériel, VHDL ou VERILOG, simplement par la pression d'un bouton. (Il est

aussi possible de générer directement le « bitstream » qui sera utilisé pour la programmation du FPGA).

- Le temps de développement d'un projet sera mesuré en journées ou semaine. Alors qu'en utilisant les approches de conception traditionnelles, le temps de développement sera calculé en mois.
- XSG permet l'échange de données entre l'environnement de SIMULINK et les puces de XILINX par l'intermédiaire de l'interface JTAG, ceci donne plus de flexibilité et permet au concepteur d'accomplir facilement le processus de co-simulation en matériel (hardware co-simulation).

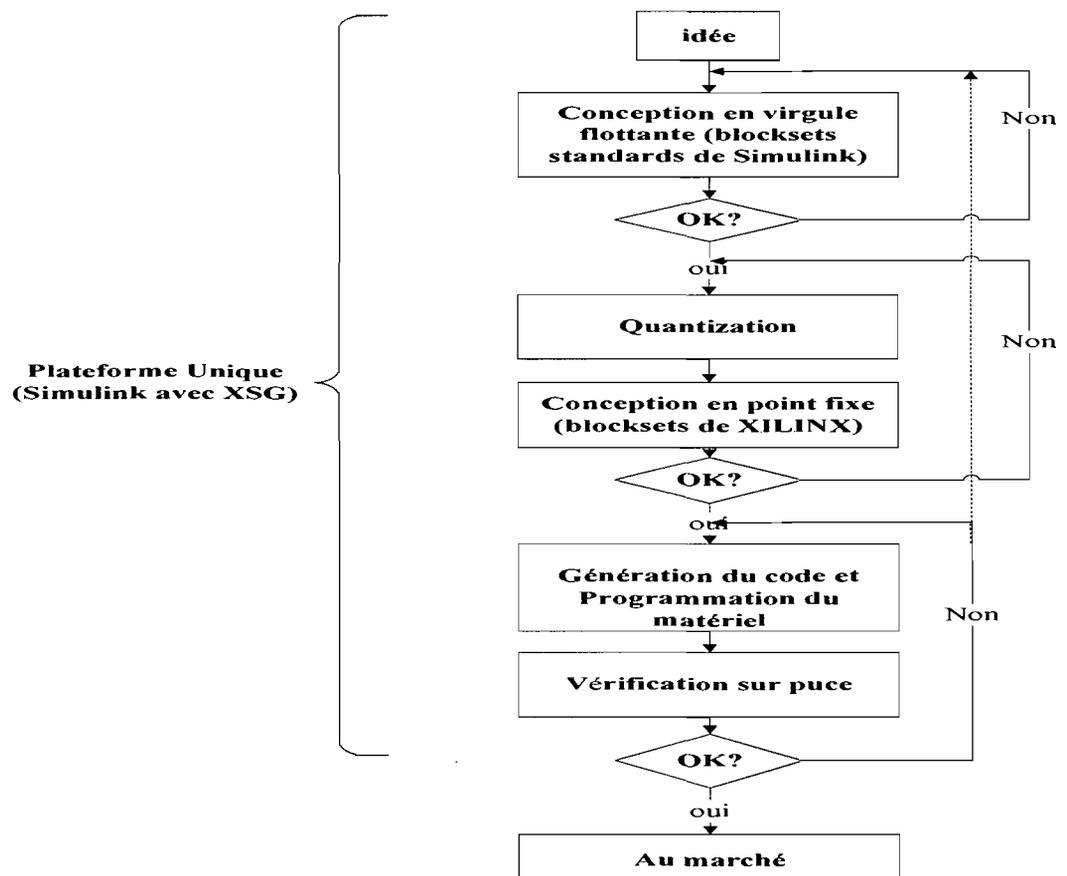


Figure 4.5 : Flux de conception VLSI avec XSG.

La figure 4.5 montre le flux de conception en utilisant XSG. En effet, le flux de conception d'un système numérique en utilisant XSG passe par les étapes suivantes :

- Étude et analyse des spécifications du système (débit, taux d'erreur binaire etc.).
- Conception et modélisation du système en utilisant les blocs standards de SIMULINK (point flottant). Le but de cette étape est d'évaluer la performance du système pour s'assurer que le modèle conçu répond aux spécifications de l'étape précédente.
- Reprendre le design du système, cette fois en utilisant uniquement les blocs de XILINX (point fixe). Après la simulation du modèle conçu, les résultats obtenus doivent être comparés aux résultats obtenus durant l'étape précédente.
- Génération du code HDL (VHDL ou VERILOG) qui correspond au modèle conçu. Cette étape est réalisée simplement par la pression d'un bouton. Le concepteur a plusieurs choix dans cette étape. En effet, au lieu de générer du code HDL, l'ingénieur peut générer le « netlist » ou le « bitstream » correspondant.
- La dernière étape est le téléchargement du « bitstream » obtenu vers le FPGA. Cependant avant la commercialisation du système, le processus de "vérification sur puce" doit être accompli.

Donc, il est bien clair que le cycle de développement VLSI avec XSG est réduit et demande moins d'expertise. De plus, des petites équipes peuvent maintenant concevoir des systèmes plus complexes en peu de temps.

4.10 Résumé de chapitre

Dans ce chapitre, nous avons présenté un aperçu général sur les outils de conception utilisés dans la réalisation de la couche physique de WiMax. Nous avons introduit en détail le fonctionnement du système de prototypage rapide « Xilinx System Generator for DSP ». Les avantages de l'utilisation de cet outil et le flux de conception VLSI en utilisant XSG étaient encore présentés. Du côté matériel, nous avons décrit brièvement les FPGAs ainsi

que leurs architecture et leurs fonctionnement. Les caractéristiques de la puce Virtex II pro XC2VP30 qui a été utilisé pour l'implémentation du design ont été présentées en détail. Les résultats de simulation du modèle de la couche physique de WiMax implémenté en SIMULINK seront présentés dans le chapitre 5. L'implémentation matérielle de tous les modules de la couche physique en utilisant XSG sera montrée et discutée dans le chapitre 6.

Chapitre 5 :

Résultats de simulation du modèle de la couche physique de WiMax

5.1 Introduction

Dans ce chapitre, les résultats de simulation du modèle de la couche physique de WiMax implémenté en Simulink seront présentés. L'objectif de ce travail est :

- D'obtenir une compréhension plus précise du fonctionnement du système de communication sans fil, WiMax, avant une implémentation définitive sur le matériel reconfigurable.
- D'évaluer la performance de la couche physique de WiMax.

De plus, nous allons étudier l'effet de la technique de correction directe (FEC) ainsi que la technique d'entrelacement sur la performance globale de la couche physique de WiMax. Nous allons, aussi, présenter la performance de tous les profils de modulation et du codage spécifiés dans WiMax. Ainsi, des interprétations sur l'utilisation de ces profils seront données pour offrir une vue complète sur la meilleure manière de la performance du WiMax.

Lors de la simulation du modèle, les points suivants seront pris en considération :

- Dans ce modèle, nous supposons que la synchronisation est parfaite entre l'émetteur et le récepteur. Les techniques de l'estimation du décalage de la fréquence porteuse

(Carrier Frequency Offset Estimation) et la synchronisation du début des trames (Frame Start Synchronization) ne sont pas traitées dans ce mémoire.

- Les observations durant la simulation sont faites en présence du module « estimation et égalisation du canal ». Ce module offre une meilleure résistance par rapport aux perturbations engendrées par le canal et peut lutter contre les effets du canal en utilisant un égaliseur.
- Le module « mélangeur » n'est pas implémenté dans ce modèle. En effet, les données binaires produites par l'émetteur sont déjà distribuées aléatoirement en utilisant la distribution de Bernoulli.
- Les techniques de modulation multi-porteuses (OFDM) sont très robustes au bruit blanc gaussien. Chaque sous-porteuse est affectée par un bruit indépendant du reste des sous-porteuses. Donc, une perte d'un symbole due à un bruit n'affecte pas les autres symboles d'une trame OFDM.

5.2 Description générale du modèle implémenté

La figure 5.1 représente le schéma bloc du modèle implémenté avec l'outil SIMULINK de MATLAB. Les détails sur le fonctionnement de ces modules sont largement expliqués dans le chapitre 2. Comme décrit dans ce dernier, au niveau de l'émetteur, les données binaires sont codées avec un code Reed-Solomon suivi d'un code convolutionnel. Après le codage on applique un processus d'entrelacement sur la séquence obtenue. Le bloc entrelaceur est représenté par une matrice où les bits rentrent ligne par ligne et lus colonne par colonne. Une division de cette séquence en groupe de bits et une transformation de ces groupes en nombre complexes est effectuée suivant la modulation appropriée. À ce niveau, on commence la phase de la construction du symbole OFDM, cette phase est effectuée en 3 étapes : la première étape consiste à ajouter les huit sous-porteuses pilotes et la sous-porteuse nulle. La deuxième étape est l'ajout des bandes de garde, ainsi que la séquence utilisée pour l'estimation du canal au récepteur. La troisième étape consiste à transformer les sous-porteuses obtenues au domaine temporel en utilisant l'inverse de la transformée de Fourier (IFFT). Par la suite, on ajoute à la transformée de Fourier une extension due à

l'intervalle de garde (préfixe cyclique). À la réception, les opérations inverses sont faites et le rapport signal/bruit est calculé en utilisant la métrique EVM. Ensuite, la valeur du rapport SNR est passée vers l'émetteur pour que ce dernier puisse adopter le profil de modulation et du codage approprié.

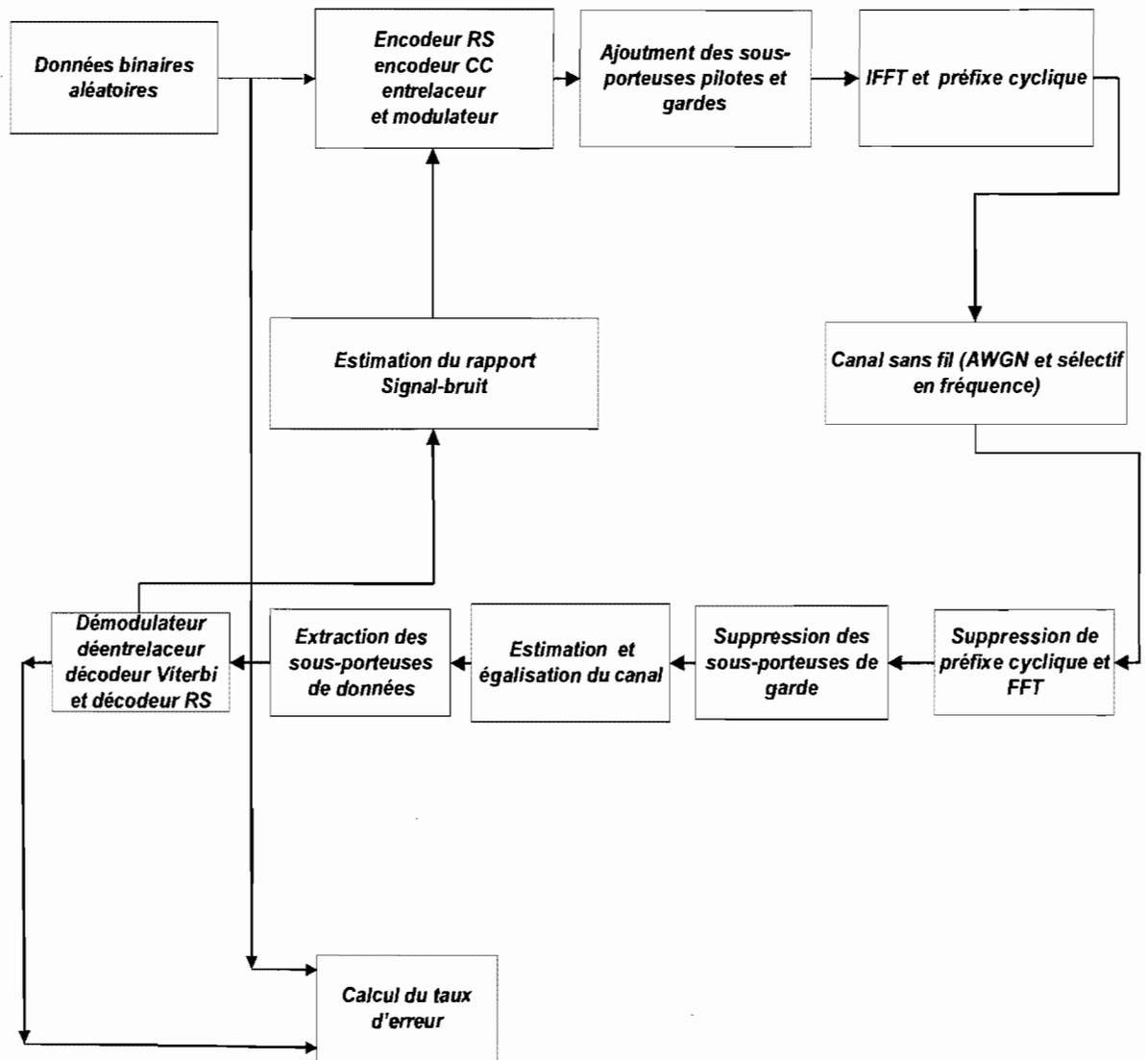


Figure 5.1 : Schéma bloc de la couche physique de WiMax implémenté en SIMULINK

5.3 Environnement de la Simulation

Les simulations mises en œuvre dans ce chapitre sont toutes faites dans MATLAB. Le système a été testé en utilisant la technique de simulation « Monte-Carlo » [39]. La simulation de Monte Carlo est utilisée pour estimer le taux d'erreur binaire (BER) que le système peut atteindre. Dans ce modèle la simulation du système est répétée, et le nombre de bits transmis ainsi que le nombre de bits erronés sont calculés pour chaque simulation. Enfin, le BER est estimé en calculant le rapport du nombre total d'erreurs observés sur le nombre total de bits transmis [39].

5.4 Les courbes BER vs SNR pour les différents profils de modulation

Dans cette section, nous présentons les différentes courbes de BER vs SNR pour tous les profils de modulation et de codage spécifiés dans WiMax. Les figures 5.2 et 5.3 montrent la performance des différents schémas de modulation respectivement sur un canal AWGN et un canal sélectif en fréquence. Il est important de mentionner que les profils de modulation les plus inférieurs (BPSK, QPSK) offrent une meilleure performance avec moins de SNR. En effet, si pour un système de modulation donné, une plus grande valeur de SNR est nécessaire pour le transfert de données, cela signifie que plus d'énergie est nécessaire pour le transfert de chaque bit.

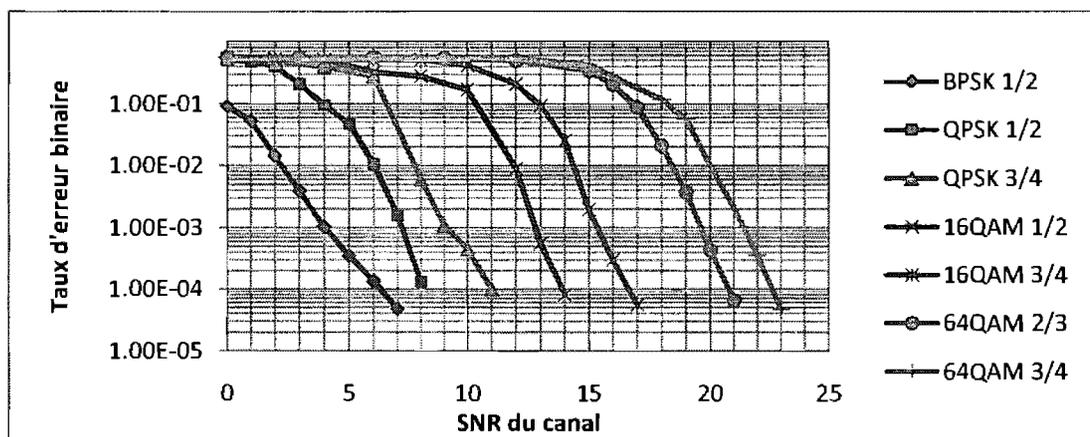


Figure 5.2 : BER vs SNR pour les différents profils de modulation et de codage sur un canal AWGN

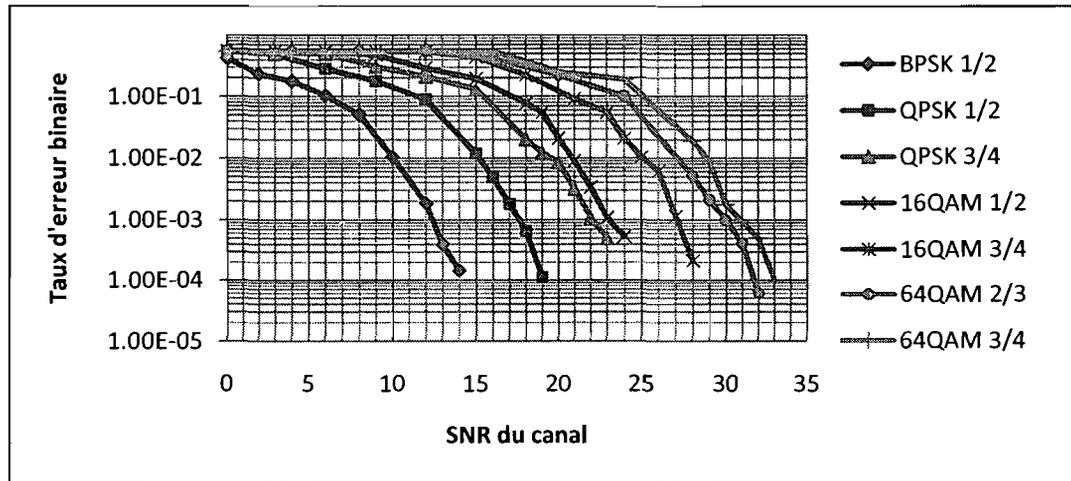


Figure 5.3 : BER vs SNR pour les différents profils de modulation et de codage sur un canal sélectif en fréquence.

Comme on peut le voir dans les figures 5.2 et 5.3, les techniques de modulation avec faible efficacité spectrale, 2-PAM (BPSK) et 4-QAM (QPSK), nécessitent un rapport signal/bruit relativement faible pour offrir un service fiable et, par conséquent, sont plus économiques en énergie et moins vulnérables aux erreurs de bits. En effet, les régimes de modulation BPSK et 4-QAM doivent être utilisés lorsque les conditions du canal sont sévères (rapport SNR faible), cependant ces régimes de modulation offrent des débits de données plus faibles. Ainsi, un compromis entre la fiabilité et le débit de données doit être considéré.

De l'autre côté, on peut constater que les régimes de modulation 16-QAM et 64-QAM nécessitent un rapport signal/bruit plus grand pour atteindre un taux d'erreur binaire acceptable. Ainsi, ces régimes de modulation ne doivent pas être appliqués à moins que la qualité du canal sans fil estimée soit bonne (rapport SNR élevé).

Les valeurs du rapport SNR requises pour atteindre un BER équivalent à 10^{-3} dans un canal AWGN sont résumées dans le tableau suivant :

Modulation	BPSK	QPSK	QPSK	16-QAM	16-QAM	64-QAM	64-QAM
Taux de codage	1/2	1/2	3/4	1/2	3/4	2/3	3/4
Canal	Rapport signal / bruit (dB) requis pour un BER égale à 10^{-3}						
AWGN	3.9	6.2	9	12.8	15.3	19.6	21.4

5.5 Effet de la correction d'erreur directe (Forward Error Correction)

Lors de l'étude de la performance de WiMax, une question intéressante se pose : combien de dégradation dans la performance apparaît dans une conception qui n'utilise pas la technique de correction d'erreur directe (encodeur Reed-Solomon suivi par l'encodeur convolutionnel). Pour visualiser l'importance et l'avantage de l'introduction de cette technique au niveau de l'amélioration de la performance de WiMax, nous avons testé la couche physique du standard 802.16 implémentée, avec et sans l'introduction des codes correcteurs (RS et CC), suivant les différents profils de modulation et de codage.

Comme prévu, les résultats obtenus avec correction d'erreur directe (RS-CC) sont nettement améliorés par rapport au cas sans correction d'erreur. Les figures 5.4, 5.5 et 5.6 représentent la variation du taux d'erreur binaire en fonction du rapport signal/bruit en utilisant, respectivement, les schémas de modulation QPSK 1/2, 16 QAM 1/2 et 64 QAM 2/3, avec et sans correction d'erreur, sur un canal sélectif en fréquence. Comme on peut le constater, le système avec correction d'erreur directe (FEC) montre une meilleure résistance face aux perturbations engendrées par le canal. En effet, on observe un gain plus grand pour les systèmes avec la technique de correction d'erreur directe par rapport aux systèmes sans cette technique. Dans la figure 5.4, on observe une amélioration de la performance d'environ 6.5 dB au niveau d'un taux d'erreur binaire équivalent à 10^{-3} . Dans le cas du 16 QAM 1/2, la figure 5.5 présente une amélioration d'environ 7.4 dB au même niveau du taux d'erreur binaire. Dans le cas de 64 QAM 3/4, les courbes de la figure 5.6 montrent une amélioration d'environ 8 dB à un taux d'erreur binaire équivalent à 10^{-4} .

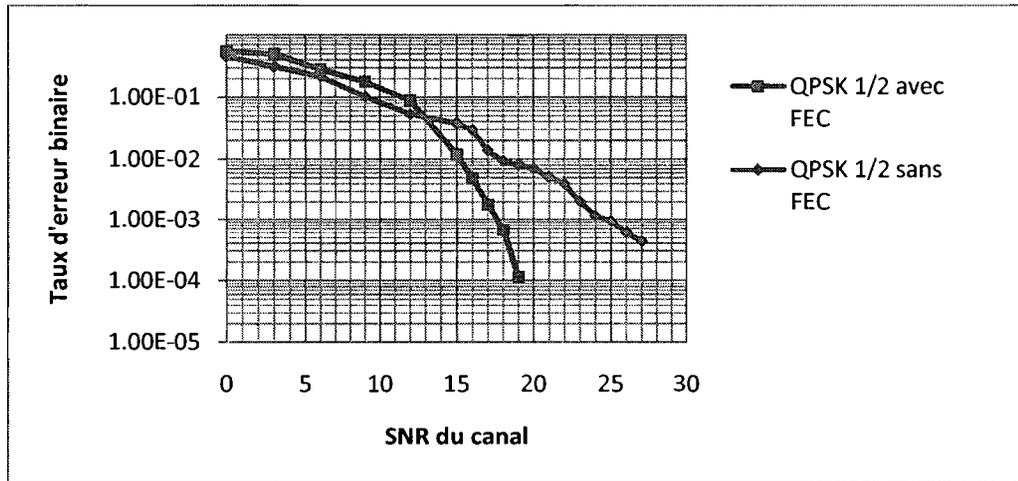


Figure 5.4 : Effet de la technique FEC dans QPSK 1/2 sur un canal sélectif en fréquence

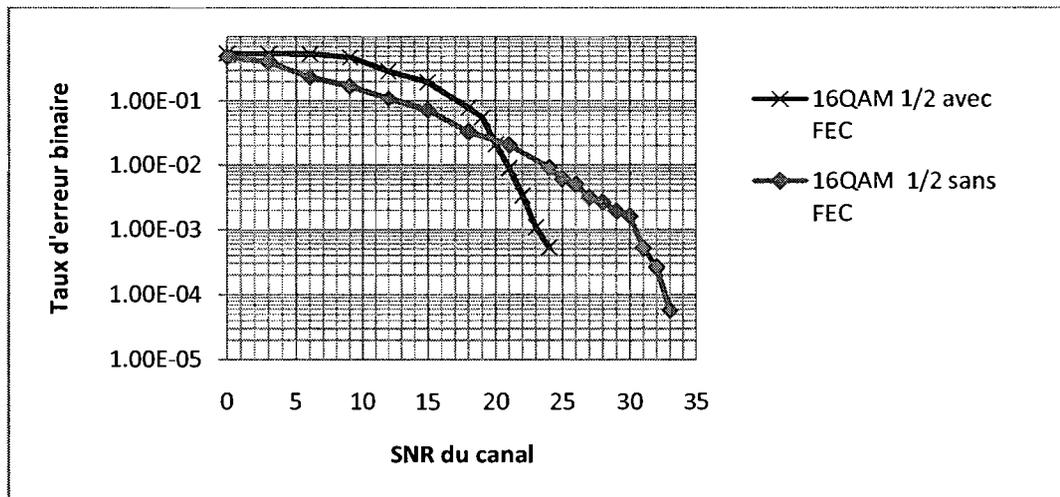


Figure 5.5: Effet de la technique FEC dans 16-QAM1/2 sur un canal sélectif en fréquence

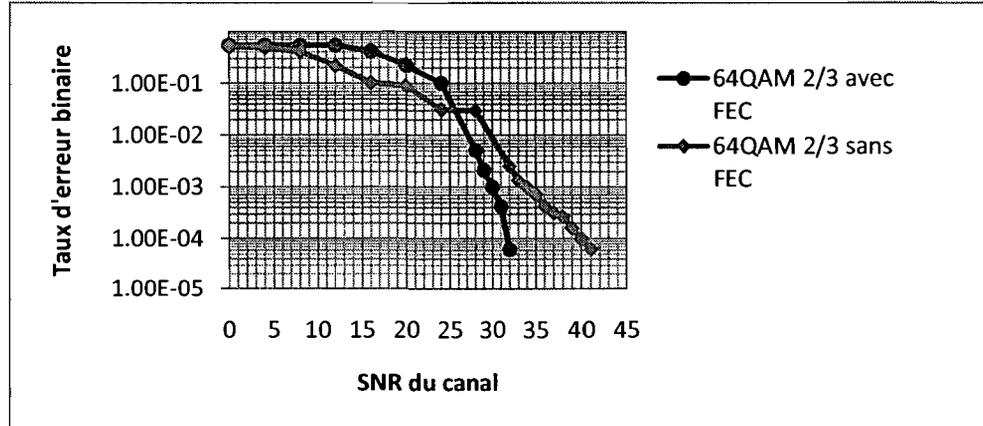


Figure 5.6 : Effet de la technique FEC dans 64 QAM 2/3 sur un canal sélectif en fréquence

Ces résultats de simulation ont montré qu'on devrait utiliser la technique de correction d'erreur directe (un code correcteur simple ou concaténé) dans presque tous les systèmes de communication numériques pour pouvoir atteindre un taux d'erreur binaire (BER) acceptable avec un rapport signal/bruit raisonnablement bas. Notons que la complexité du système augmente avec l'utilisation des techniques de correction d'erreur directe. Finalement, des compromis doivent être acceptés, si on considère la compensation entre la complexité et la performance du système.

5.6 Effet de l'entrelaceur

Les simulations effectuées avec les techniques d'entrelacement et de désentrelacement ont démontré une meilleure performance de la couche physique de WiMax. Ces techniques permettent de répartir d'une façon aléatoire les erreurs afin de faciliter leur correction au décodage. Les courbes de la figure 5.7 et 5.8 montrent les meilleurs résultats obtenus. On observe clairement un gain plus grand pour les profils de modulation avec l'entrelaceur par rapport à ceux sans le processus d'entrelacement.

On observe dans la figure 5.7 une détérioration de la performance lorsque l'entrelaceur n'est pas employé. En effet, pour un taux d'erreur binaire équivalent à 10^{-3} , la détérioration

de la performance vaut 1.7 dB. Par contre, dans la figure 5.8 la dégradation dans la performance était moins sévère, une détérioration d'environ 0.8 dB a été observée pour le même taux d'erreur binaire.

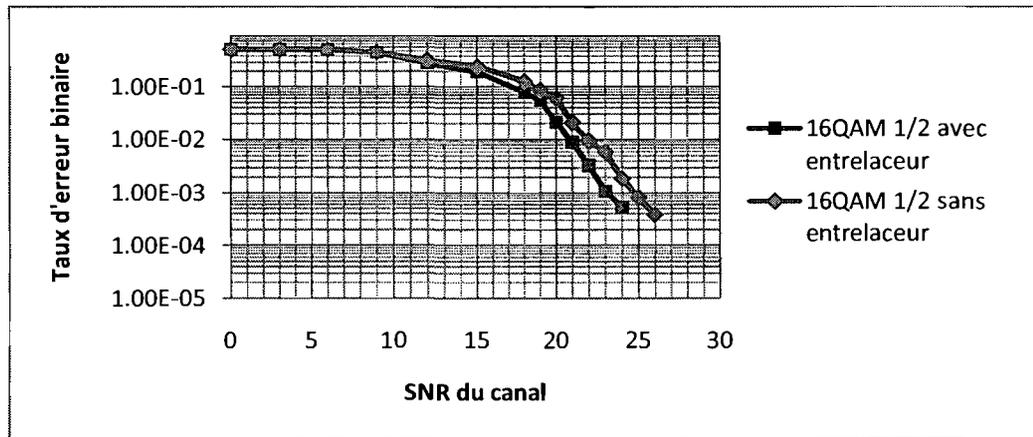


Figure 5.7 : Effet de l'entrelaceur dans 16 QAM 1/2 sur un canal sélectif en fréquence

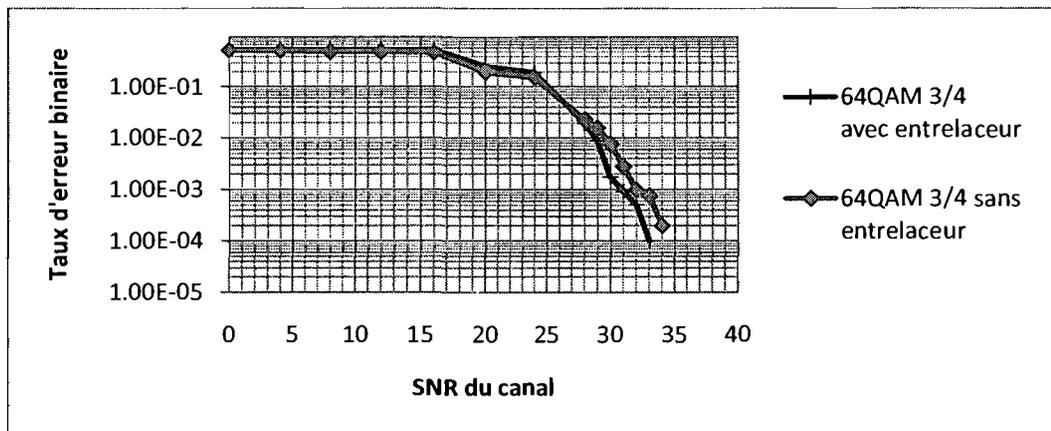


Figure 5.8 : Effet de l'entrelaceur dans 64 QAM 3/4 sur un canal sélectif en fréquence

5.7 Analyse générales des résultats de simulation

Pour la couche physique de WiMax, nous avons comparé la performance des différents profils de modulation et de codage spécifiés. Nous avons constaté qu'avec les profils de

modulation 64-QAM et 16-QAM le système subit une dégradation considérable de performance par rapport aux systèmes avec la structure de modulation QPSK ou BPSK.

Dans ce contexte, la performance du profil de modulation QPSK 1/2 était supérieure d'environ 12.3 dB par rapport à la performance du profil de modulation 64 QAM 2/3, et d'environ 6 dB par rapport à la performance du profil de modulation 16 QAM 1/2. On a vérifié que les hauts profils de modulation (16-QAM et 64-QAM) sont plus adaptés aux environnements de propagations moins sévères, alors que les bas profils de modulation (BPSK et QPSK) sont adaptés aux environnements de propagation plus sévères. En effet, puisque l'efficacité spectrale s'améliore avec l'augmentation de l'ordre de la modulation, les ingénieurs doivent contrebalancer entre l'efficacité spectrale minimale requise et les performances d'erreur considérées acceptables selon l'application en question.

D'autre part, nous avons montré par simulation l'importance et l'avantage de l'introduction de la technique de correction d'erreur directe (FEC). En effet, un gain moyen supérieur à 7 dB a été observé pour le cas avec la technique de correction d'erreur directe par rapport au cas sans cette technique. Nous avons conclu qu'on doit utiliser une technique de correction d'erreur directe (un code correcteur simple ou concaténé), dans presque tous les systèmes de communication numériques pour pouvoir atteindre un BER acceptable. Dans ce contexte, Il est à noter que la complexité et le coût de la mise en œuvre d'un système de communication sans fil augmente avec l'utilisation des codes correcteurs et des techniques de diversité temporelles comme l'entrelaceur. Comme déjà introduit dans ce mémoire, des compromis doivent être acceptés, si l'on considère la compensation entre la complexité et la performance du système.

5.8 Résumé du chapitre

Dans ce chapitre, nous avons présenté quelques résultats de simulation du modèle de la couche physique de WiMax implémenté en Simulink. L'objectif principal de ce travail était

d'obtenir une compréhension plus précise du fonctionnement de la couche physique de WiMax. Dans ce contexte, nous avons présenté la performance de tous les profils de modulation et de codage adoptés par WiMax et des interprétations sur l'utilisation de ces profils étaient présentées. De plus, nous avons montré par simulation l'importance de l'introduction de la technique de correction d'erreur directe (encodeur Reed Solomon et encodeur convolutionnel), qui sont adoptées dans presque tous les standards de communication numérique sans fil récents (Wifi, UWB et WiMax...), sur la performance générale de la couche physique de WiMax. Nous avons conclu que WiMax n'était pas en mesure d'offrir une qualité de service efficace et fiable sans l'introduction de cette technique. À partir du chapitre suivant nous allons traiter l'implémentation de tous les modules de la couche physique de WiMax sur un FPGA en utilisant Xilinx System Generator for DSP (XSG).

Chapitre 6 :

Implémentation matérielle des modules de la couche physique de WiMax

6.1 Introduction

Les bases théoriques de la couche physique de WiMax ont été présentées dans le chapitre 2. D'autre part, le flux de la conception matérielle en utilisant la plateforme de prototypage rapide Xilinx System Generator for DSP (XSG) a été présenté en détail dans le chapitre 4. De plus, les résultats de simulation du modèle de la couche physique de WiMax implémenté en Simulink ont été présentés et discutés dans le chapitre 5. L'objectif de ce chapitre est alors de présenter et traiter la réalisation et l'implémentation de tous les modules obligatoires de la couche physique de WiMax sur un FPGA en utilisant XSG. Chaque module de la chaîne de transmission (émetteur) sera présenté, et les caractéristiques principales de la conception de ces modules seront mises en évidence. Plus tard dans ce chapitre, la chaîne de réception sera discutée brièvement, puisque la plupart des blocs du récepteur consistent simplement à effectuer la fonction inverse de son correspondant du côté émetteur. À la fin de ce chapitre, nous allons comparer nos résultats avec ceux des autres travaux de recherche traitant le même sujet.

Toutefois, avant de commencer ce chapitre, il est intéressant de mentionner que notre but est de fournir une bibliothèque des blocs qui abstractent les détails de l'implémentation des modules obligatoires de la couche physique de WiMax. Ces blocs sont hautement optimisés, pipelinés, synthétisables, réutilisables, efficaces, simples et personnalisables. En outre, les modules implémentés supportent tous les profils de modulation et de codage

spécifiés au niveau de la couche physique de WiMax avec une longueur de préfixe cyclique variable. Par conséquent, ces blocs peuvent être utilisés au niveau commercial et académique pour une implémentation très simple et très rapide de la couche physique de WiMax et plus tard pour l'intégration de ces modules avec les modules de la couche MAC. D'autre part, tous les modules implémentés ont été vérifiés au niveau de composants. Dans ce contexte, chaque module implémenté dans XSG a été vérifié et testé contre le module correspondant en SIMULINK, qui à son tour, a été testé et vérifié contre le vecteur de test disponible dans le standard.

6.2 Détails de l'implémentation des modules de l'émetteur

Comme détaillé dans le chapitre 2, l'émetteur WiMax à bande de base est composé de sept modules : mélangeur, encodeur Reed-Solomon, encodeur convolutionnel (avec perforation), entrelaceur, modulateur, insertion des sous-porteuses pilotes et bandes de garde, la transformée inverse de Fourier et le préfixe cyclique. Dans les sections suivantes, nous allons traiter l'implémentation matérielle de ces modules en utilisant Xilinx System Generator for DSP.

6.2.1 Mélangeur

Comme le montre la figure 6.1, le circuit du mélangeur est très simple. Ce circuit est constitué de quinze registres à décalage avec deux «ou exclusifs». Deux signaux contrôlent le fonctionnement du mélangeur. Tout d'abord, le signal « *initialize* » permet de charger les registres avec les valeurs initiales correspondantes. Ces valeurs dépendent du signal « *initialization type* ». Lorsque « *initialization type* » est mis à 1, la séquence [100101010000000] est chargée aux registres, sinon la séquence [BSID 1 1 D/UIUC 1 index_du_trame] sera chargée. Les résultats de synthèse du mélangeur vers le FPGA ont montré que les ressources utilisées par ce module étaient négligeables. De plus, la période minimale estimée était de 4.301 ns, qui correspond à une fréquence maximale de 232.51 MHz. Le résultat de synthèse détaillé du mélangeur est donné dans le tableau 6.1.

Tableau 6.1 : Ressources consommées et période minimale du mélangeur.

Tranches	Tranches FFs	LUTs à 4 entrées	BRAMs	Multiplieurs 18* 18	période minimale
18	27	23	0	0	4.301 ns

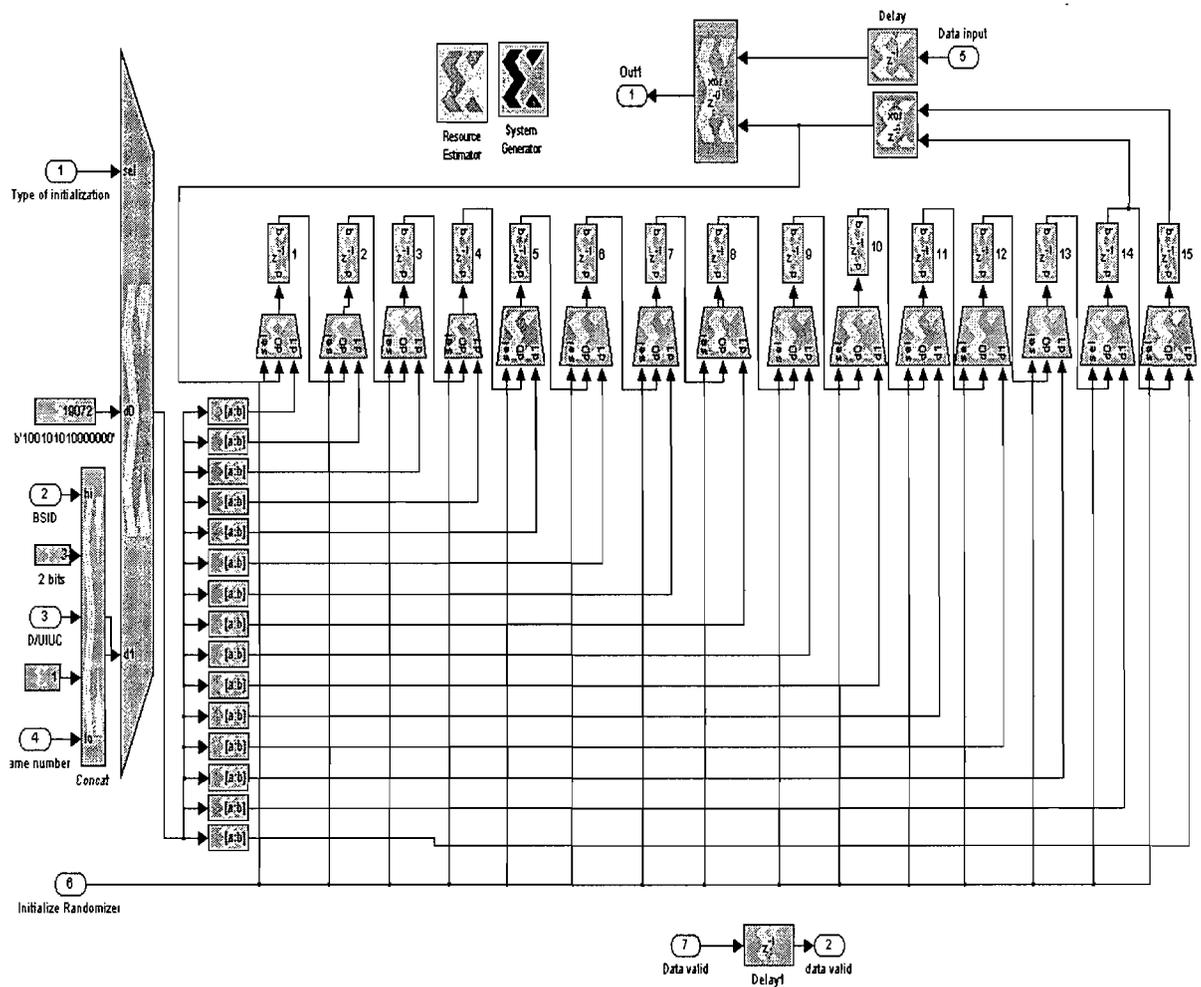


Figure 6.1: Circuit du mélangeur dans XSG.

6.2.2 Encodeur Reed-Solomon (RS)

Dans l'implémentation de l'encodeur RS, la propriété intellectuelle (RS-Encoder IP) fournie par la bibliothèque génératrice principale de XILINX (Xilinx core generator library) a été utilisée. Cependant, la propriété intellectuelle de XILINX fournit un encodeur RS générale qui ne supporte pas les processus de perforation et de raccourcissement spécifiés dans WiMax. Pour cela, deux phases sont nécessaires (pré et post l'encodage RS) afin de personnaliser l'encodeur selon la norme 802.16. La première phase (pré-RS) est le raccourcissement (shortening). Le raccourcissement consiste à ajouter $239-K$ zéros octets avant le bloc de données. Dans ce contexte, K est la taille du bloc à l'entrée de l'encodeur RS, K est égale à 24, 36, 48, 72, 96, 108 octets pour les profils QPSK $\frac{1}{2}$, QPSK $\frac{3}{4}$, 16QAM $\frac{1}{2}$, 16QAM $\frac{3}{4}$, 64QAM $\frac{2}{3}$, 64QAM $\frac{3}{4}$ respectivement. L'architecture de la première phase (pré-RS) est illustrée dans la figure 6.2. À la sortie des données de l'encodeur RS, la deuxième phase commence. Cette phase est atteinte en deux étapes. La première étape consiste à extraire les premiers $2T'$ octets de parité à partir des 16 octets de parité générés par l'encodeur RS. $2T'$ est égale à 8, 4, 16, 8, 12, 12 pour les profils de modulation QPSK $\frac{1}{2}$, QPSK $\frac{3}{4}$, 16QAM $\frac{1}{2}$, 16QAM $\frac{3}{4}$, 64QAM $\frac{2}{3}$, 64QAM $\frac{3}{4}$ respectivement. La deuxième étape sert à la réorganisation des octets de façon à ce que les octets de parité ($2T'$) soient lus en premier suivis par les octets des données. Les zéros octets sont éliminés. Donc, la taille du bloc valable à la sortie de l'encodeur RS sera égale à $K + 2T'$. Pour réaliser la deuxième phase avec ses 2 étapes, nous avons utilisé une mémoire RAM pour enregistrer les données à leurs sorties de l'encodeur RS. Ensuite, les octets de parité sont lus à partir de l'adresse 239 jusqu'à $239 + 2T' - 1$ et les octets de données sont lus à partir de l'adresse $239 - K$ jusqu'à 238. Étant donné que le délai de l'encodeur RS est équivalent à deux cycles [67], le délai principal dans la conception décrite plus haut est équivalent à 255 cycles qui correspondent au temps nécessaire pour l'écriture de 255 octets dans la RAM. En ajoutant le délai qui correspond à la lecture de $K + 2T'$ octets de la RAM, le délai du circuit devient $255 + k + 2T'$. Toutefois, pour exploiter le parallélisme, une deuxième RAM est ajoutée. Cette deuxième RAM est utilisée pour stocker les données provenant de l'encodeur RS, tandis que les données codées sont en cours de lecture de la première RAM. Ainsi, une

structure pipeliné est atteinte ce qui permet d'ajuster le délai du circuit à $255 - k - 2T'$ cycles.

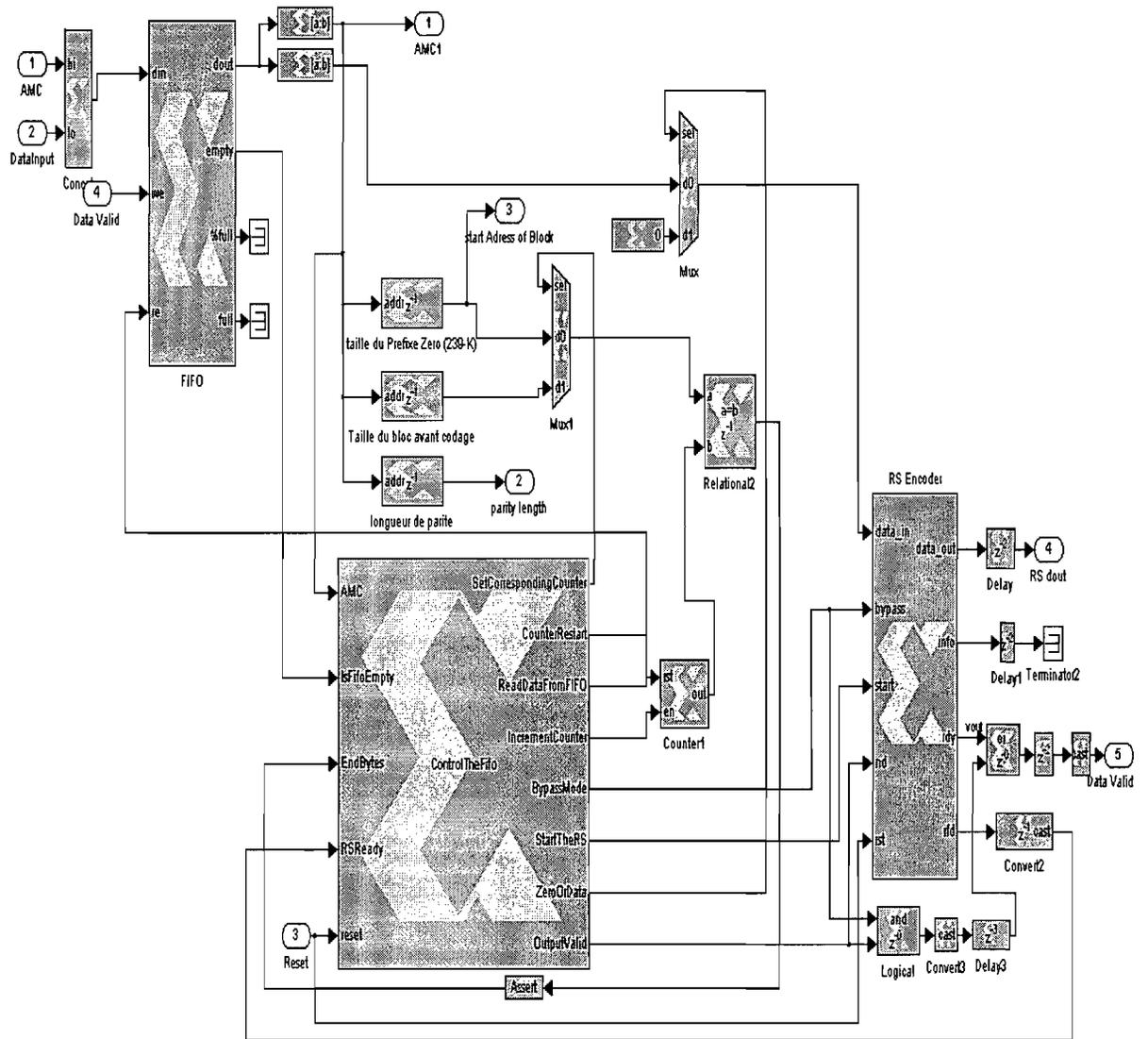


Figure 6.2 : Circuit de l'encodeur Reed-Solomon dans XSG.

Tableau 6.2 : Ressources consommées et période minimale de l'encodeur RS

Tranches	Tranches FFs	LUTs à 4 entrées	BRAMs	Multiplieurs 18* 18	période minimale
428	408	603	2	0	5.245 ns

6.2.3 Encodeur Convolutionnel

Tel que décrit dans la norme 802.16, l'encodeur convolutionnel (CC) est constitué de six registres à décalage avec deux «ou exclusifs». L'architecture décrite dans la norme était traduite vers une conception en XSG. La figure 6.3 montre la conception de l'encodeur convolutionnel en XSG. Dans ce contexte, il ya quelques points à souligner. Tout d'abord, l'encodeur Reed-Solomon transmet les données au CC en format octets, alors que l'encodeur convolutionnel opère sur la séquence de données bit par bit. Pour cela, nous avons utilisé le bloc «convertisseur parallèle-série » dans XSG afin de fournir les données à l'encodeur convolutionnel sous format bit. De plus, pour chaque bit à l'entrée, l'encodeur convolutionnel produit deux bits de sortie. Toutefois, la norme 802.16 spécifie que certains bits à la sortie du CC doivent être rejetés (voir tableau 2.2). Pour cela, nous avons défini une fonction MatLab par l'intermédiaire du bloc « Mcode » dans XSG. Pour chaque pair de bits à la sortie du CC, la fonction MatLab produit deux signaux (*Xvalid*, *Yvalid*) pour indiquer si la sortie du CC doit être considérée ou non. Les ressources consommées par l'encodeur convolutionnel ainsi que sa vitesse de fonctionnement sont présentées dans le tableau 6.3.

Tableau 6.3 : Ressources utilisées par l'encodeur convolutionnel et le perforateur.

Tranches	Tranches FFs	LUTs à 4 entrées	BRAMs	Multiplieurs 18* 18	Période minimale
33	56	26	0	0	3.271 ns

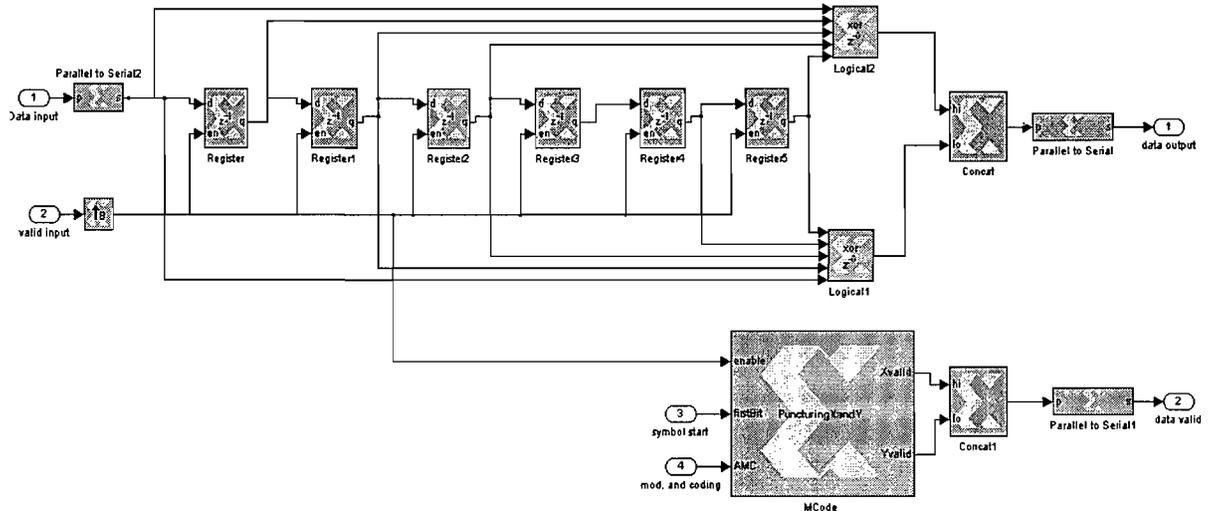


Figure 6.3: Circuit de l'encodeur convolutionnel dans XSG.

6.2.4 Entrelaceur

Comme indiqué dans le chapitre 2, le rôle de l'entrelaceur est de réordonner les bits d'un symbole. En effet, le processus d'entrelacement est réalisé en trois étapes. La première étape consiste à définir une mémoire ROM qui, pour chaque indice d'un bit dans un symbole, produit l'indice de ce bit après l'entrelacement. La deuxième étape consiste à sauvegarder le bit reçu dans une mémoire RAM à l'adresse générée du ROM. Après avoir écrit tous les bits d'un symbole dans la RAM, la troisième étape consiste à lire ces bits séquentiellement de la RAM. En effet, cette conception fournit une fonctionnalité correcte, cependant, le retard introduit dans une telle conception sera considérablement élevé. Ceci est clair, parce que nous ne pouvons pas commencer le traitement d'un nouveau symbole jusqu'à ce que le symbole en cours sera écrit et ensuite lu de la RAM. Pour exploiter le parallélisme, nous avons utilisé la RAM à double port (Dual port RAM) qui permet un accès simultané à la RAM. En utilisant la RAM à double port, nous pouvons lire le symbole de la RAM et en même temps commencer l'écriture d'un nouveau symbole dans la

RAM à différentes adresses du premier symbole (nous pouvons aussi considérer l'utilisation de deux RAMs au lieu d'un RAM à deux ports). Avec cette stratégie, il est évident que le retard introduit dans la conception sera juste le temps de l'écriture du premier symbole qui est équivalent à 192, 384, 768, 1152 pour les schémas de modulation BPSK, QPSK, 16-QAM et 64-QAM respectivement. L'architecture de l'entrelaceur est présentée dans la figure 6.4. Cette architecture supporte tous les profils de modulation et de codage spécifiés dans WiMax. L'utilisation des ressources et la période minimale estimée de l'entrelaceur sont données dans le tableau 6.4.

Tableau 6.4 : Ressources utilisées et période minimale de l'entrelaceur

Tranches	Tranches FFs	LUTs à 4 entrées	BRAMs	Multiplieurs 18* 18	période minimale
72	58	101	3	0	4.753

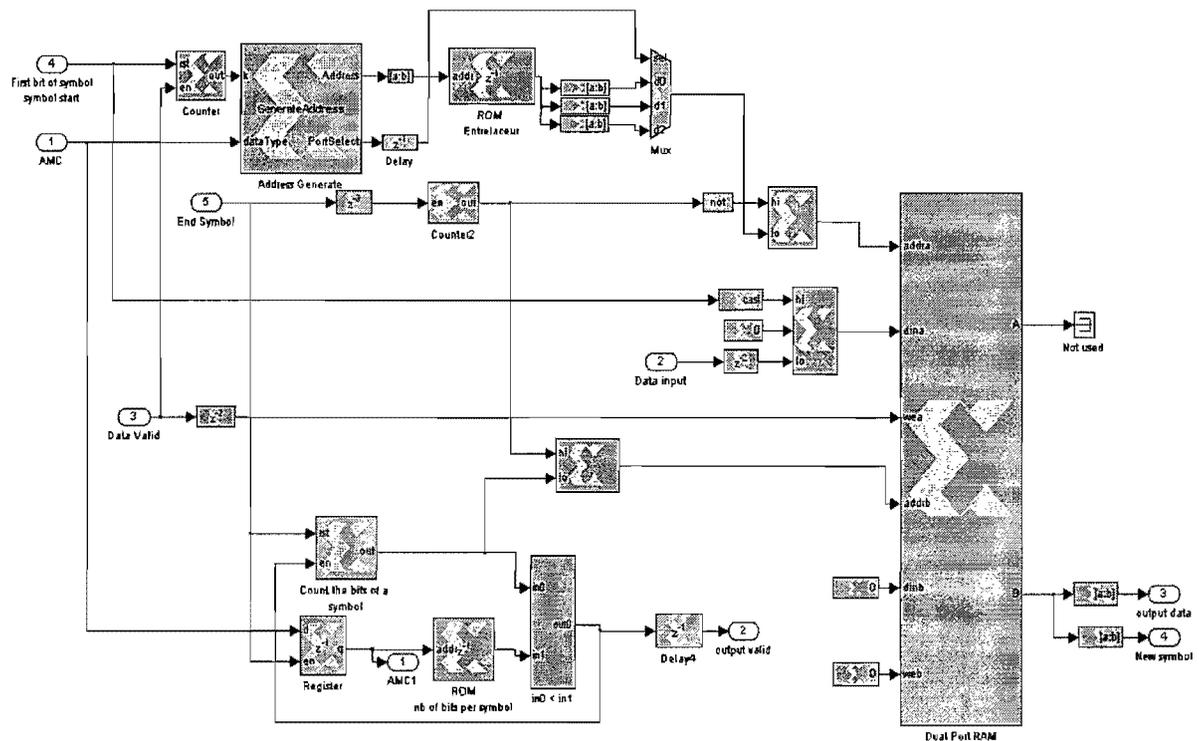


Figure 6.4 : Circuit de l'entrelaceur dans XSG.

6.2.5 Modulateur

Le modulateur transforme les bits reçus en nombres complexes. La partie réelle du nombre complexe à la sortie du modulateur est appelée I, alors que la partie imaginaire est appelée Q. L'architecture du modulateur dans XSG est montrée dans la figure 6.5. Tout d'abord, les bits sont organisés en groupe de 1, 2, 4, ou 6 bits pour les schémas de modulation BPSK, QPSK, 16-QAM, 64-QAM respectivement. Ensuite, les bits I et Q sont passés vers deux mémoires ROMs pour extraire leurs points de constellation correspondants. Dans ce contexte, contrairement à la conception de Garcia [19] qui utilise quatre mémoires ROM pour implémenter le modulateur. Notre conception utilise deux ROMs. Le tableau 6.5 montre le résultat de la synthèse détaillée du modulateur.

Tableau 6.5 : Ressources utilisées et période minimale du modulateur.

tranches	Tranches FFs	LUTs à 4 entrées	BRAMs	Multiplieurs 18* 18	Période minimale
29	47	50	0	0	4.685 ns

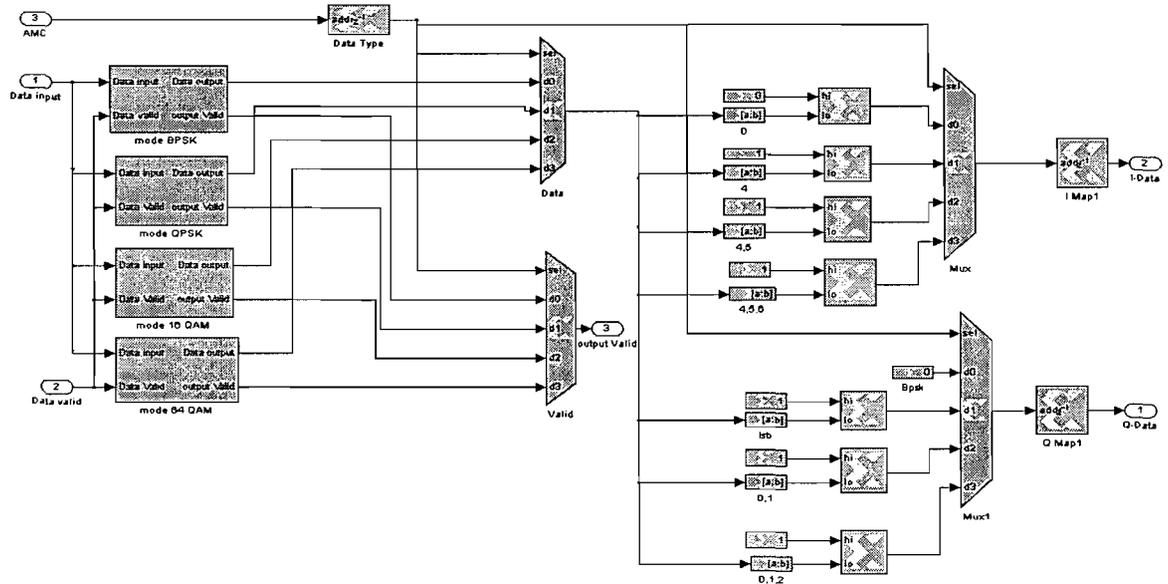


Figure 6.5 : Circuit du modulateur dans XSG.

6.2.6 Insertion des sous-porteuses pilotes et bandes de garde

Ce module ajoute les sous-porteuses pilotes et gardes à la séquence des symboles modulés pour former les 256 échantillons à l'entrée de la transformée inverse de Fourier. Les valeurs des sous porteuses pilotes et gardes ainsi que leurs indices correspondants ont été présentés dans le chapitre 2 (voir tableau 2.4). L'architecture de l'insertion des sous-porteuses pilotes et gardes est montrée dans la figure 6.6. Tout d'abord, la partie réelle et la partie imaginaire à la sortie du modulateur sont écrites dans deux FIFO. Ensuite, un compteur est utilisé pour le suivi de l'indice de la sous-porteuse en cours. Ainsi, en utilisant une logique combinatoire, un multiplexeur est choisi pour prendre les sous-porteuses de données, pilotes ou gardes en fonction de l'indice de la sous-porteuse donné par le compteur.

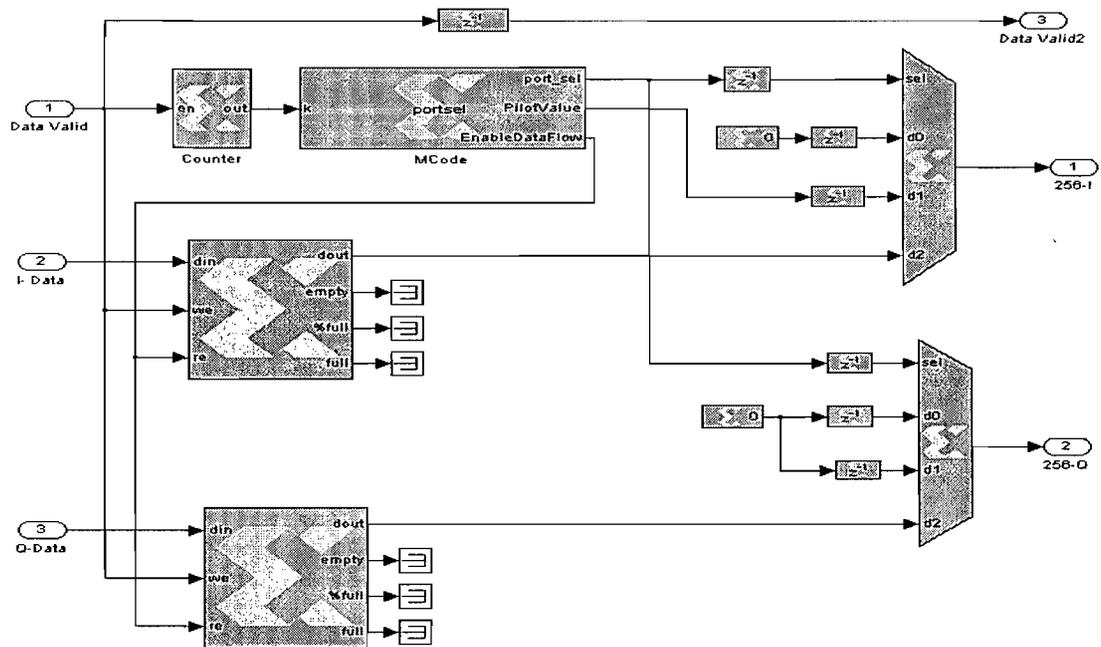


Figure 6.6 : Circuit de l'insertion des sous-porteuses pilotes et gardes dans XSG.

Tableau 6.6 : Ressources utilisées et période minimale de l'insertion des sous-porteuses pilotes et gardes

Tranches	Tranches FFs	LUTs à 4 entrées	BRAMs	Multiplieurs 18* 18	période minimale
72	51	96	2	0	6.06 ns

6.2.7 La transformée inverse de Fourier

Pour répondre aux spécifications de la norme 802.16, le bloc IFFT exige une architecture bien pipelinée. À cette fin, la propriété intellectuelle de XILINX, qui est une partie de la plateforme de développement de XSG, a été utilisée. Ce bloc calcule la FFT ou l'IFFT à base de 256 points. La FFT/IFFT a été employée dans le mode diffusion pipeliné d'entrée/sortie (pipelined streaming input/output) qui fournit un traitement continu des données reçues. Les valeurs à l'entrée et la sortie du bloc FFT/IFFT sont des nombres à 24 bits représentés en complément à deux. Au niveau de l'émetteur, il est évident que le module IFFT est le bloc qui consomme la plus grande partie des ressources. Les résultats de synthèse de ce bloc ont montrés que l'IFFT occupe plus que 65% des ressources totales consommées par l'émetteur. De plus, la période minimale du module IFFT était de 6.923 qui correspondent à une fréquence d'environ 144 MHZ. Étant donné que le module avec la plus petite fréquence gouvernera la fréquence de fonctionnement de l'émetteur, nous pouvons déduire que la fréquence de l'opération de l'émetteur est équivalente à 144 MHZ.

Tableau 6.7 : Ressources utilisés et période minimale de la transformée inverse de Fourier.

Tranches	Tranches FFs	LUTs à 4 entrées	BRAMs	Multiplieurs 18* 18	période minimale
2805	4122	4503	4	12	6.923 ns

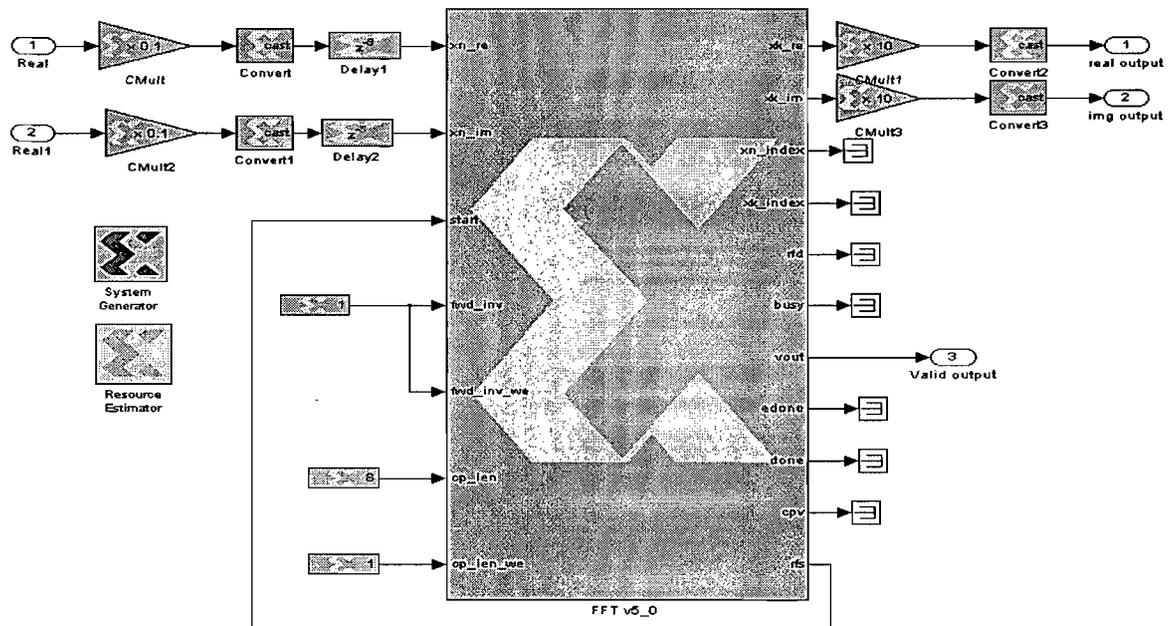


Figure 6.7 : Le bloc IFFT dans XSG.

6.2.8 Préfixe cyclique (CP)

Le préfixe cyclique est la répétition des m derniers échantillons du symbole OFDM au début de ce symbole. L'architecture de l'insertion du préfixe cyclique est présentée dans la figure 6.8. Cette architecture est basée sur deux RAMs. La première RAM est utilisée pour stocker les 256 échantillons d'un symbole OFDM à leurs sorties de l'IFFT. La deuxième RAM est utilisée pour stocker les m derniers échantillons du même symbole OFDM. Lors de la lecture des données, un multiplexeur avec une logique de contrôle sont utilisés pour transmettre le contenu de la deuxième RAM (RAM du préfixe cyclique) suivi par le contenu de la première RAM (RAM de données). Encore une fois, pour surmonter le problème du retard causé par la lecture et l'écriture des données dans les deux RAMs, une autre copie de chaque RAM (RAM de préfixe cyclique et RAM de données) est ajoutée pour fournir le pipelining. Ainsi, la sortie de l'IFFT est écrite dans la deuxième copie des RAMs pendant que la sortie de la première copie des RAMs est en cours de lecture et vice-versa. L'architecture de l'insertion du préfixe cyclique est illustrée dans la figure 6.8.

Toutefois dans ce travail, le préfixe cyclique a été ajouté directement à partir du bloc IFFT. Dans ce contexte, la propriété intellectuelle de Xilinx, utilisée pour calculer la transformée inverse de Fourier, fournit l'option d'ajouter un préfixe cyclique de longueur variable avant la transformée inverse de Fourier.

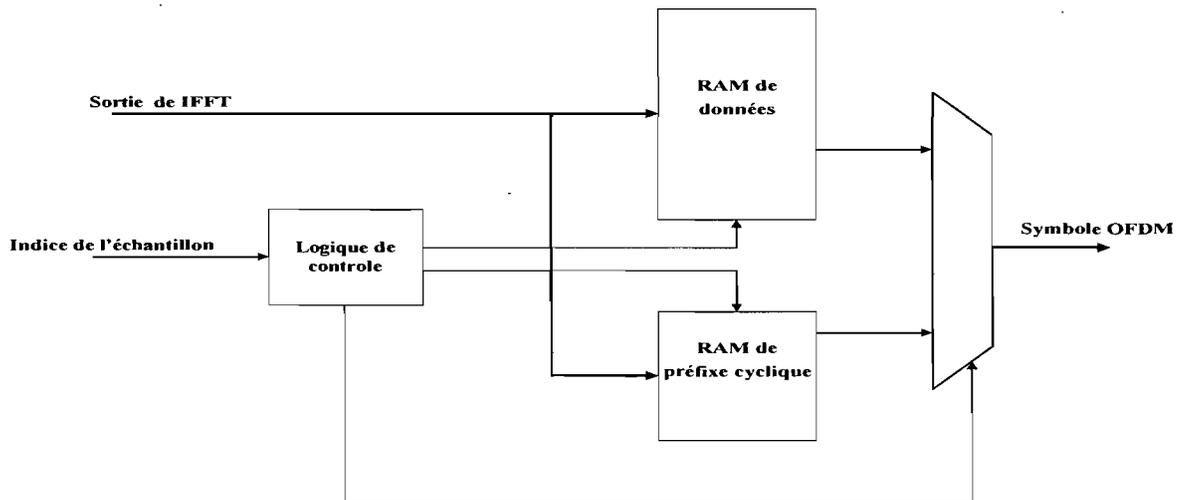


Figure 6.8 : Architecture de préfixe cyclique

6.3 Détails de l'implémentation des modules du récepteur

Comme dans la plupart des systèmes de communication, les composants logiques de l'émetteur, existent également au niveau du récepteur, dans l'ordre inverse, pour reconstruire la séquence des informations transmises. Dans ce travail, les mêmes blocs de la chaîne de transmission ont été inversés et légèrement modifiés avec quelques blocs remplacés ou modifiés d'une manière significative pour l'implémentation des modules de la chaîne de réception. La figure 6.9 montre les modules de la chaîne de réception implémentés dans ce travail. L'architecture de ces modules dans XSG est montrée dans l'annexe A.

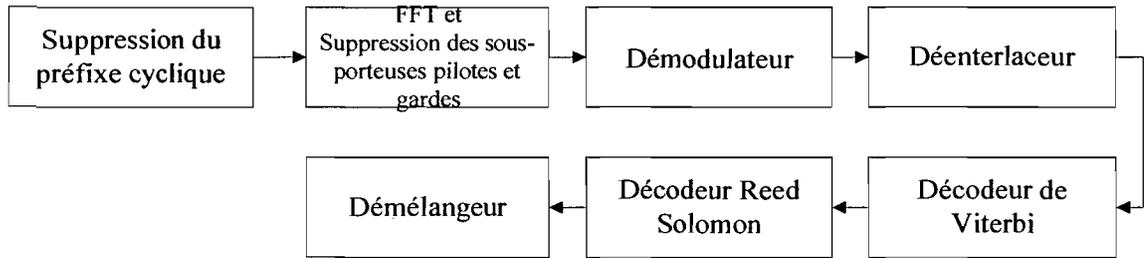


Figure 6.9 : Modules de la chaîne de réception implémentés.

Au niveau de la chaîne de réception, le seul bloc qui a été remplacé était le décodeur de Viterbi, qui effectue la fonction inverse de l'encodeur convolutionnel. En effet, la propriété intellectuelle de XILINX, qui est une partie de la plateforme de conception dans XSG a été utilisée pour implémenter le décodeur de Viterbi. Cette propriété intellectuelle fournit le décodage d'une séquence codée en utilisant un encodeur convolutionnel de taux $1/2$ avec une longueur de 7. De plus, la propriété intellectuelle de XILINX (décodeur de Viterbi) supporte également les différents taux de perforation requis par la norme 802.16d. Un autre bloc qui a été modifié d'une manière importante est la suppression du préfixe cyclique. Dans ce contexte, contrairement à l'insertion du préfixe cyclique à l'émetteur où quatre mémoires RAMs ont été utilisées pour stocker les données à leur sortie du bloc IFFT. La suppression du préfixe cyclique ne nécessite pas de mémoire RAM. Cette unité fonctionne simplement en ignorant les premiers 32, 16 ou 8 échantillons du symbole OFDM reçu. Ces échantillons correspondent au préfixe cyclique inséré à l'émetteur. Le bloc démodulateur était aussi modifié d'une manière significative. En effet, les signaux (à la sortie du bloc FFT) ont été quantifiés et démodulés en utilisant l'algorithme de vraisemblance logarithmique simplifié (simplified log likelihood ratio) [66]. Ce simple algorithme a été mis en œuvre par l'intermédiaire du bloc « Mcode » dans XSG. Le détail du fonctionnement de l'algorithme de vraisemblance logarithmique simplifié est présenté dans l'annexe B. Le reste des blocs du récepteur sont principalement basés sur les blocs déjà implémentés dans l'émetteur avec une légère modification. Plus précisément, pour l'implémentation du démélangeur, qui effectue la fonction inverse du mélangeur, le même circuit montré dans la figure 6.1 a été utilisé. De plus, dans l'implémentation du

déentrelaceur, la seule partie qui a été modifiée était le contenu de la ROM, afin de refléter les indices générés par les équations 2.9 et 2.10. Les autres blocs, comme le FFT et le décodeur Reed-Solomon ont été implémentés en utilisant les propriétés intellectuelles de XILINX qui forment aussi une partie de la plateforme de développement dans XSG. Similairement, des légères modifications ont été nécessaires pour refléter les différences entre ces blocs et les blocs de l'émetteur. Dans ce contexte, l'étape post RS, qui consiste à la réorganisation des données à leurs sorties de l'encodeur RS, n'était pas nécessaire.

En effet, bien que certaines étapes dans la conception de l'émetteur aient été omises au niveau du récepteur, la complexité des autres blocs du récepteur, surtout le décodeur de Viterbi, a conduit à un résultat qui montre que la conception du récepteur occupe plus d'espace que celui de l'émetteur. Les résultats de synthèse des blocs de la chaîne de réception sont donnés dans le tableau 6.8.

Tableau 6.8 : Ressources consommées par les modules du récepteur

Module	Tranches	Tranches FFs	LUTs à 4 entrées	Multiplieurs 18*18	Bloc RAMs	période minimale (ns)
Supprimer le CP	40	34	76	0	0	3.572
FFT (avec suppression des sous-porteuses pilotes et gardes)	2943	4296	4903	12	5	6.828
Démodulateur	587	499	1075	0	1	5.94
Déentrelaceur	84	77	100	0	3	4.662
Décodeur de Viterbi	1493	1206	2460	0	5	7.363
Décodeur RS	645	782	1093	0	3	5.832
Démélangeur	18	27	23	0	0	4.301

6.4 Analyse générale des résultats

Après la vérification de chaque module, les résultats de synthèse ont montré que les familles des FPGAs de densité moyenne, comme le virtex II Pro de XILINX, étaient en mesure d'accommoder efficacement la conception de tous les modules de la couche physique de WiMax. Dans ce contexte, nous pouvons affirmer qu'approximativement 50% des ressources disponibles sur le FPGA (xc2vp30-7fg676) étaient suffisantes pour l'implémentation de tous les modules de la couche physique de WiMax. De plus, il est évident, d'après les résultats de l'utilisation des ressources que les blocs qui exigent le plus grand nombre des ressources étaient le FFT/IFFT ainsi que le décodeur de Viterbi. Du côté de l'émetteur, nous avons constaté que le taux d'occupation du module IFFT représente environ 65% du total des ressources occupées par l'émetteur. Similairement, au récepteur, il est clair que les modules FFT et décodeur de Viterbi représente respectivement 65% et 25% (approximativement) du total des ressources occupées par le récepteur. Une autre observation à signaler au niveau du récepteur, c'est que le démodulateur et le décodeur Reed-Solomon occupe environ le double des ressources occupées par le modulateur et l'encodeur Reed-Solomon dans l'émetteur. D'autre part, nous avons remarqué que le nombre de bloc RAM utilisé dans la conception était élevé. En effet, 28 blocs RAMs ont été utilisés dans cette conception. Cependant, l'utilisation d'un grand nombre de blocs RAMs ne dégrade pas l'efficacité de la conception puisque tous les FPGAs récents sont équipés d'un grand nombre de bloc RAM. Dans ce travail, le pourcentage de l'utilisation des blocs RAMs représente seulement 20% du total des blocs RAMs disponibles.

D'autre part, bien que les ressources utilisées forment un facteur important pour le succès de design, la vitesse d'opération de la conception constitue aussi un facteur important et vital. Dans ce travail, la contrainte temporelle principale est d'être en mesure de produire un symbole OFDM en $72 \mu s$ (temps du symbole OFDM). Ceci exige l'utilisation d'une horloge fonctionnant à 14 MHz environ. Toutefois, il est clair d'après les résultats que le module le plus lent déterminera la période minimale et la fréquence maximale de la chaîne de transmission ainsi que la chaîne de réception. Dans ce contexte, du côté de l'émetteur, le

module IFFT était le plus lent avec une période minimale de 6.923 ns et une fréquence maximale d'environ 144 MHz. De l'autre côté, au récepteur, le module le plus lent était le décodeur de Viterbi avec une période minimale de 7.363 ns et une fréquence maximale d'environ 135 MHz. Un résultat détaillé sur l'espace occupé et la vitesse d'opération de tous les modules de la chaîne de transmission ainsi que la chaîne de réception est donné dans le tableau 6.9.

Tableau 6.9 : Ressources consommées et période minimale des modules des chaînes de transmission et de réception

Module	Tranches	Tranches FFs	LUTs à 4 entrées	Multiplieurs 18*18	Bloc RAMs	période minimale (ns)
Mélangeur	18	27	23	0	0	4.301
Encodeur RS	428	408	603	0	2	5.245
Encodeur convolutionnel	33	56	26	0	0	3.271
Entrelaceur	72	58	101	0	3	4.753
Modulateur	29	47	50	0	0	4.685
Insertion des symboles pilotes et gardes	72	51	96	12	2	6.06
IFFT (avec insertion du préfixe cyclique)	2805	4122	4503	0	4	6.923
Ressources consommées (chaîne de transmission)	3457	4769	5402	12	11	6.923
Supprimer le CP	40	34	76	0	0	3.572
FFT (avec suppression des sous-porteuses pilotes)	2943	4296	4903	12	5	6.828
Démodulateur	587	499	1075	0	1	5.94
Déentrelaceur	84	77	100	0	3	4.662
Décodeur de Viterbi	1493	1206	2460	0	5	7.363
Décodeur RS	645	782	1093	0	3	5.832
Démélangeur	18	17	23	0	0	4.301
Ressources consommées (chaîne de réception)	5810	6921	9730	12	17	7.363
Total des ressources	9267	11690	15132	24	28	-
Ressources disponibles : (virtex II pro : xc2vp30-7fg676)	13696	27392	27392	136	136	-

En effet, l'objectif de ce travail est de montrer la pertinence des matériels reconfigurables (FPGAs) pour la réalisation des systèmes de communication sans fil à large bande tel que WiMax. Dans ce contexte, les résultats de la synthèse des modules implémentés ont montré que les contraintes de temps et d'espace ont été atteintes. De plus, ce travail vise à souligner l'importance des outils de synthèse de haut niveau pour la réalisation des systèmes complexes tels que WiMax, afin d'accélérer le temps de développement et de minimiser l'équipe du design. Dans ce contexte nous pouvons affirmer que, en utilisant la plateforme de prototypage rapide XSG, des petites équipes peuvent concevoir des systèmes plus complexes, comme la couche physique de WiMax, en peu de temps.

6.5 Comparaison

Peu d'études rapportent les ressources utilisées dans l'implémentation de WiMax sur un FPGA. L'étude faite par Garcia [19] est comparée à notre travail. Tout d'abord, puisque le travail de Garcia est limité à la partie modulation de la chaîne de transmission (modulateur, insertion des symboles pilotes et bandes de garde, IFFT, préfixe cyclique), les résultats obtenus dans ce travail montrent une augmentation raisonnable dans le nombre des ressources utilisés si les blocs supplémentaires ajoutés (les blocs du codage du canal) sont considérés. De plus, contrairement au travail de Garcia dans lequel seulement la partie de la modulation de la chaîne de transmission est implémentée, ce travail implémente à la fois la partie modulation et la partie codage du canal ainsi que les modules de la chaîne réception. Enfin, une note concernant les résultats de Garcia, c'est qu'aucun multiplicateur n'est mentionné dans les résultats, alors que Garcia utilise la propriété intellectuelle de XILINX, qui est une partie de la plateforme de développement dans XSG, pour implémenter le module IFFT. Toutefois, cette propriété intellectuelle exige l'utilisation de douze multiplicateurs quand elle est instanciée dans XSG. Le tableau ci-contre montre une brève comparaison entre nos résultats et ceux de Garcia.

Tableau 6.10 : Comparaison entre ce travail et le travail de Garcia

Ressource	Ce travail (chaîne de transmission et de réception)	Ce travail (chaîne de transmission)	Le travail de Garcia (modulation de la chaîne de transmission)
Tranches	9267	3457	2614
Tranches FFs	11690	4769	3566
LUTs à 4 entrées	15132	5402	4304
BRAMs	28	11	12
Multiplieurs 18*18	24	12	0

6.6 Résumé du chapitre

Dans ce chapitre, tous les modules de la couche physique de WiMax conformément à la norme 802.16d ont été implémentés en utilisant la plateforme de prototypage rapide Xilinx System Generator for DSP (XSG). La conception a été ciblée vers la famille des FPGA de densité moyenne virtex II Pro (xc2vp30-7fg676) de XILINX. Les résultats de la synthèse des modules implémentés ont montré que les contraintes de temps et d'espace ont été atteintes. Une discussion plus approfondie sur le travail accompli dans ce mémoire ainsi que les travaux futurs suggérés seront présentées dans le chapitre suivant.

Chapitre 7 :

Conclusion et travaux futurs

7.1 Résumé du travail accompli

Dans ce mémoire, nous avons en premier temps implémenté un modèle complet de la couche physique de WiMax. Ce modèle, implémenté en Simulink, comporte tous les modules obligatoires spécifiés dans le standard 802.16d. En effet, la compréhension approfondie de la structure et la fonctionnalité de la couche physique de la technologie WiMax (avant une implémentation définitive sur une architecture reconfigurable) était un objectif important pour la modélisation de cette couche. De plus, un autre objectif de cette modélisation était d'évaluer la performance de la couche physique de WiMax ainsi que d'étudier l'effet de la technique de correction d'erreur directe (codes correcteurs concaténés RS et CC) sur la performance globale de cette couche. Dans ce contexte, nous avons présenté la performance de tous les profils de modulation et de codage spécifiés au niveau de la couche physique de WiMax sur un canal AWGN et sélectif en fréquence. En outre, les résultats de simulation du modèle implémenté ont montré qu'un gain moyen supérieur à 8 dB a été observé lors de l'introduction de la technique FEC (encodeur Reed Solomon et encodeur convolutionnel). Nous avons conclu que WiMax n'était pas en mesure d'offrir une qualité de service efficace et fiable sans l'introduction de cette technique. En deuxième lieu, nous avons présenté l'implémentation de tous les modules de la couche physique de WiMax sur une architecture reconfigurable en utilisant la plateforme de prototypage rapide Xilinx System Generator for DSP (XSG) qui est une partie de la plateforme de développement de Simulink. Deux objectifs ont été la raison de l'implémentation de tous les modules de la couche physique de WiMax sur une plateforme reconfigurable. D'abord,

nous voudrions montrer que les normes de communication sans fil récentes telles que 802.15 (réseau personnel sans fil), 802.20 (Accès sans fil mobile et à large bande), 802.22 (réseau régionale sans fil) et en particulier 802.16 (WiMax) peuvent être efficacement conçues, implémentées et réalisées sur du matériel reconfigurable plutôt qu'utiliser les circuits intégrés dédiés (ASICs) pour la réalisation des tels systèmes. Dans ce contexte, nous avons présenté une implémentation matérielle efficace et pipelinée de tous les modules de la couche physique de WiMax en utilisant XSG. Les résultats d'implémentation et de la synthèse des modules implémentés ont montré qu'approximativement 50% du total des ressources de la plateforme virtex II pro (FPGA de densité moyenne) ont été utilisés par cette implémentation. D'autre part, la vitesse d'opération de la chaîne de transmission était environ 144 MHz (vitesse d'opération de l'IFFT) et la vitesse d'opération de la chaîne de réception était environ 135 MHz (vitesse d'opération du décodeur de Viterbi). Par conséquent, nous pouvons affirmer que les contraintes d'espace et de temps ont été atteintes. Le deuxième objectif était de souligner la pertinence des outils de synthèse de haut niveau pour la mise en œuvre des systèmes complexes comme la couche physique de WiMax. Dans ce contexte, nous pouvons affirmer qu'en utilisant la plateforme de prototypage rapide Xilinx System Generator for DSP (XSG), des petites équipes avec moins d'expérience peuvent concevoir des systèmes plus complexes, comme la couche physique de WiMax, en peu de temps.

7.2 Liste des travaux futurs suggérés

Une des contributions principales de ce mémoire est la mise en œuvre d'une infrastructure et d'une plateforme excellente pour aborder les secteurs de recherche récents et promettants dans le domaine de la communication numérique sans fil et les matériels reconfigurables. Ci-dessous une liste des travaux futurs suggérés qui n'ont pas fait l'objet de nos études ou n'ont pas été considérés lors de la modélisation et la réalisation du standard 802.16d. Cette liste permettra aux nouveaux étudiants des cycles supérieurs de se baser sur ce travail pour s'intégrer simplement et rapidement avec la communauté de

recherche en cours au niveau de la communication numérique sans fil et les matériels reconfigurables.

7.2.1 Longueur de préfixe cyclique adaptif dans WiMax

Le standard de WiMax spécifie plusieurs longueurs pour l'intervalle de garde (préfixe cyclique). De plus, cette longueur dépend essentiellement du retard de la propagation (delay spread). Récemment, des algorithmes efficaces pour l'estimation du délai de la propagation ont été proposés [63]. L'implémentation de l'algorithme de [63] en un module Simulink, et ensuite l'addition de ce module au modèle WiMax proposé dans ce travail permettra d'évaluer la performance de WiMax avec une longueur de préfixe cyclique adaptif. Sans doute, le système proposé conduira à des résultats meilleurs au niveau de l'augmentation de l'efficacité spectrale et la minimisation de l'interférence entre les symboles. Cependant, le coût de la réalisation d'un tel système doit être considéré pour évaluer la faisabilité de l'idée proposée. Plus tard, l'implémentation de l'idée proposée sur FPGA peut être considérée.

7.2.2 Synchronisation entre l'émetteur et le récepteur

Dans notre travail, nous avons supposé une synchronisation parfaite entre l'émetteur et le récepteur lors de la simulation et la réalisation de WiMax. Alors qu'en pratique cela arrive rarement. Récemment, une multitude d'algorithmes de synchronisation pour les systèmes de communications multi-porteuses ont été proposés [58-59]. Dans le cadre de WiMax, on peut mener une étude comparative sur l'effet de ces algorithmes au niveau de l'amélioration de la performance de WiMax. L'implémentation de ces algorithmes sur du matériel reconfigurable fait aussi l'objet des recherches en cours [57].

7.2.3 Évaluation de la performance de WiMax avec les profils de modulation 256-QAM et 512-QAM

Les profils de modulation adoptés par WiMax sont les BPSK, QPSK, 16-QAM et 64-QAM (on a implémenté ces schémas de modulation dans ce travail). Dans ce contexte,

le standard spécifie que le profil de modulation 256-QAM est optionnel, alors que le 512-QAM n'est pas spécifié dans le standard. Un des travaux futurs qu'on propose est l'évaluation de la performance de WiMax en utilisant les profils de modulation 256-QAM et 512-QAM. On peut prédire qu'en utilisant les profils de modulation déjà mentionnés, l'efficacité spectrale sera considérablement augmentée, alors qu'un taux d'erreur binaire acceptable ne sera pas atteint à moins que le rapport signal bruit sera considérablement élevé. De plus, le coût de la réalisation sera aussi élevé. Dans ce contexte, une étude détaillée est précisée sur l'effet des profils de modulation 256-QAM et 512-QAM au niveau de la performance de WiMax est bien nécessaire.

7.2.4 Le rapport de la puissance moyenne-maximale (Peak to Average Power Ratio, PAPR)

Alors que le problème de PAPR n'était pas traité dans ce mémoire, la réduction du PAPR constitue un des désavantages principaux reliés à la technique de multiplexage par répartition orthogonale des fréquences (OFDM). Plusieurs techniques de réduction du PAPR existent déjà dans la littérature [61-62]. De plus, plusieurs publications intéressantes [60] ont abordé l'implémentation de quelques algorithmes de réduction du PAPR sur FPGA. Dans ce secteur, il serait intéressant de comparer les algorithmes proposés dans la littérature pour extraire celui le plus adapté à la mise en œuvre sur FPGA. Dans une telle étude, on doit considérer la performance de l'algorithme au niveau de la réduction du PAPR, l'espace occupé par cet algorithme, sa vitesse de fonctionnement et sa consommation d'énergie. Cependant, il est à noter qu'en général, la réduction du PAPR ne fait pas partie du domaine de recherche en communication numérique. Par contre, PAPR est plus liée au secteur de recherche en traitement des signaux.

7.2.5 Entrées multiples sorties multiples (Multiple Input Multiple Output MIMO)

Bien que la technique d'entrées-sorties multiples MIMO n'ait pas été considérée dans ce travail, MIMO est une technologie promettante considérée comme l'une des

technologies les plus avancées dans le domaine de communication. La technique MIMO a montré l'effet le plus profond au niveau de la capacité, la fiabilité et l'efficacité spectrale des systèmes sans fil. Cependant, l'amélioration de la performance résultante de MIMO implique l'augmentation de la complexité et la consommation d'énergie au récepteur [64]. Toutefois, au cours de ma maîtrise, je n'ai pas eu l'occasion de bien découvrir le monde du MIMO. Cependant, à mon avis, notre travail ouvre la voie aux zones des recherches suivantes : (1) la conception et la réalisation d'un système WiMax MIMO à haut débit moins complexe avec faible consommation d'énergie, (2) La modélisation et la réalisation des schémas d'allocations de la puissance et des canaux pour les systèmes MIMO et (3) la modélisation et l'implémentation des couches physiques multi-antennes pour les nouvelles technologies de communication sans fil telles que 802.11n (MIMO WIFI), 802.16e (mobile single hop WiMax) et 802.16j (mobile multi-hop ou mesh WiMax). La modélisation de ces couches peut être accomplie en utilisant les outils OPNET, NS-2 ou Simulink. De plus, la construction des modèles analytiques pour ces nouvelles technologies, si possible, sera aussi un secteur de recherche intéressant.

7.2.6 Partitionnement matérielle/logicielle de l'implémentation la couche physique

Le partitionnement matériel/logiciel est un processus qui se produit fréquemment au niveau de la conception des systèmes embarqués. Il s'agit essentiellement d'une procédure pour déterminer si une partie d'un système devrait être mise en œuvre en logiciel ou en matériel. Alors que dans ce travail tous les modules de la couche physique de WiMax ont été implémentés en matériel avec XILINX system Generator (XSG). Parfois, quelques modules de WiMax seront plus flexibles et plus simples s'ils sont implémentés en logiciel, en utilisant, par exemple, le processeur Microblaze de XILINX. Dans ce contexte, des outils de haut niveau comme le VisualSim de Mirabilisdesign [65] peuvent être utilisés pour l'exploration de la conception et la description architecturale du modèle proposé avant une implémentation définitive sur FPGA.

Bibliographie

- [1] IEEE. Standard 802.16-2004. Part16: "Air interface for fixed broadband wireless access systems", October 2004.
- [2] IEEE. Standard 802.16e-2005 Part16: "Air interface for fixed and mobile broadband wireless access systems— Amendment for physical and medium access control layers for combined fixed and mobile operation in licensed band", 2005.
- [3] Loutfi Nuaymi: "WiMAX: Technology for Broadband Wireless Access," John Wiley & Sons, ISBN: 978-0-470-02808-7, 2007.
- [4] Deepak Pareek, "Wimax: Taking Wireless to the Max", Auerbach Publications, ISBN: 0849371864, 2007.
- [5] Jeffrey G. Andrews, Arunabha Ghosh, Rias Muhamed: "Fundamentals of WiMAX: understanding broadband wireless networking," Pearson Education, ISBN: 0-13-222552-2, 2007.
- [6] Ian Grout, "digital systems design with FPGAs and CPLDs", Elsevier Ltd, ISBN: 978-0-7506-8397-5, 2008.
- [7] Steve Kilts, "Advanced FPGA Design: Architecture, Implementation, and Optimization", John Wiley & Sons, ISBN 978-0-470-05437-6, 2007.
- [8] Martin Bossert, "Channel Coding for Telecommunications", Wiley publications, ISBN 0471982776, 1998.
- [9] A. M. Michelson, et A. H. Levesque: "error Control Techniques for Digital Communications," Wiley-Interscience Publications, 1985.
- [10] S. B. Wicker: "Error Control Systems for Digital Communication and Storage," School of Electrical and Computer Engineering, Georgia Institute of Technology, Prentice Hall, 1995.
- [11] R. W. Chang, "Synthesis of band-limited orthogonal signals for multichannel data transmission", Bell Syst. Tech, 1966
- [12] S. Weinstein et P. Ebert, "Data transmission by frequency-division multiplexing using the discrete Fourier transform", IEEE Trans. Comm., Volume:19,Pages: 628-634, 1971.
- [13] L. Cimini, "Analysis and Simulation of a Digital Mobile Channel using Orthogonal Frequency Division multiplexing", IEEE trans. Comm., vol. 33, pages: 665-675, 1985.

- [14] Marc-André, Cantin Laurent Moss, Guy Bois, “WiMax 802.16 Version 5.0“, technical report, space co-design ,2007.
- [15] Xu Jinsong, Lu Xiaochun, Wu Haitao, Bian Yujing, Zou Decai, Zou Xiaolong, Wang Chaogang, “Implementation of MB-OFDM Transmitter Baseband Based on FPGA”, 4th IEEE International Conference on Circuits and Systems for Communications, 2008.
- [16] “Implementing WiMAX OFDM Timing and Frequency Offset Estimation in Lattice FPGAs,” Lattice Semiconductor White Paper, November 2005.
- [17] “Implementation of an OFDM Wireless Transceiver using IP Cores on an FPGA,” Lattice Semiconductor White Paper, August 2005.
- [18] J. Garcia et R. Cumplido, “On the design of an FPGA-Based OFDM modulator for IEEE 802.11a”, 2nd International Conference on Electrical and Electronics Engineering (ICEEE) and XI Conference on Electrical Engineering (CIE 2005), Mexico City, Mexico, 2005.
- [19] J. Garcia et R. Cumplido, “on the design of an FPGA-based OFDM modulator for IEEE 802.16-2004”, International Conference on Reconfigurable Computing and FPGAs, 2005
- [20] M. Serra, J. Ordiex, P. Marti et J. Carrabina, “OFDM Demonstrator: Transmitter” in Proc 7th International OFDM Workshop 2002, Sep 2002.
- [21] Sghaier Ahmad, Areibi Shawki, Dony Bob, “A pipelined implementation of OFDM transmission on reconfigurable platforms”, Canadian Conference on Electrical and Computer Engineering, 2008.
- [22] ALTERA, ”Accelerating WiMAX System Design with FPGAs”, Technical Report, Altera Corporation, 2005
- [23] Jordan Douglas Guffey, “OFDM Physical Layer Implementation for the Kansas University Agile Radio”, MS thesis, university of Kansas April 2007.
- [24] English homepage for IAF <http://iaf-bs.de/index.en.html>
- [25] K.M. Gharaibeh, K.G. Gard, M.B. Steer, “Accurate estimation of digital communication system metrics - SNR, EVM and ρ in a nonlinear amplifier environment”, 64th *Microwave Measurements Conference*, 2004.
- [26] T. Alexander, “Optimizing and Testing WLANs: Proven Techniques for Maximum Performance”, Newnes, 2007, ISBN 0750679867.
- [27] J. van de Beek, O. Edfors, M. Sandell, S.K Wilson, P.O. Borjesson,” On channel estimation in OFDM systems”, 45th IEEE Vehicular Technology Conference, 1995

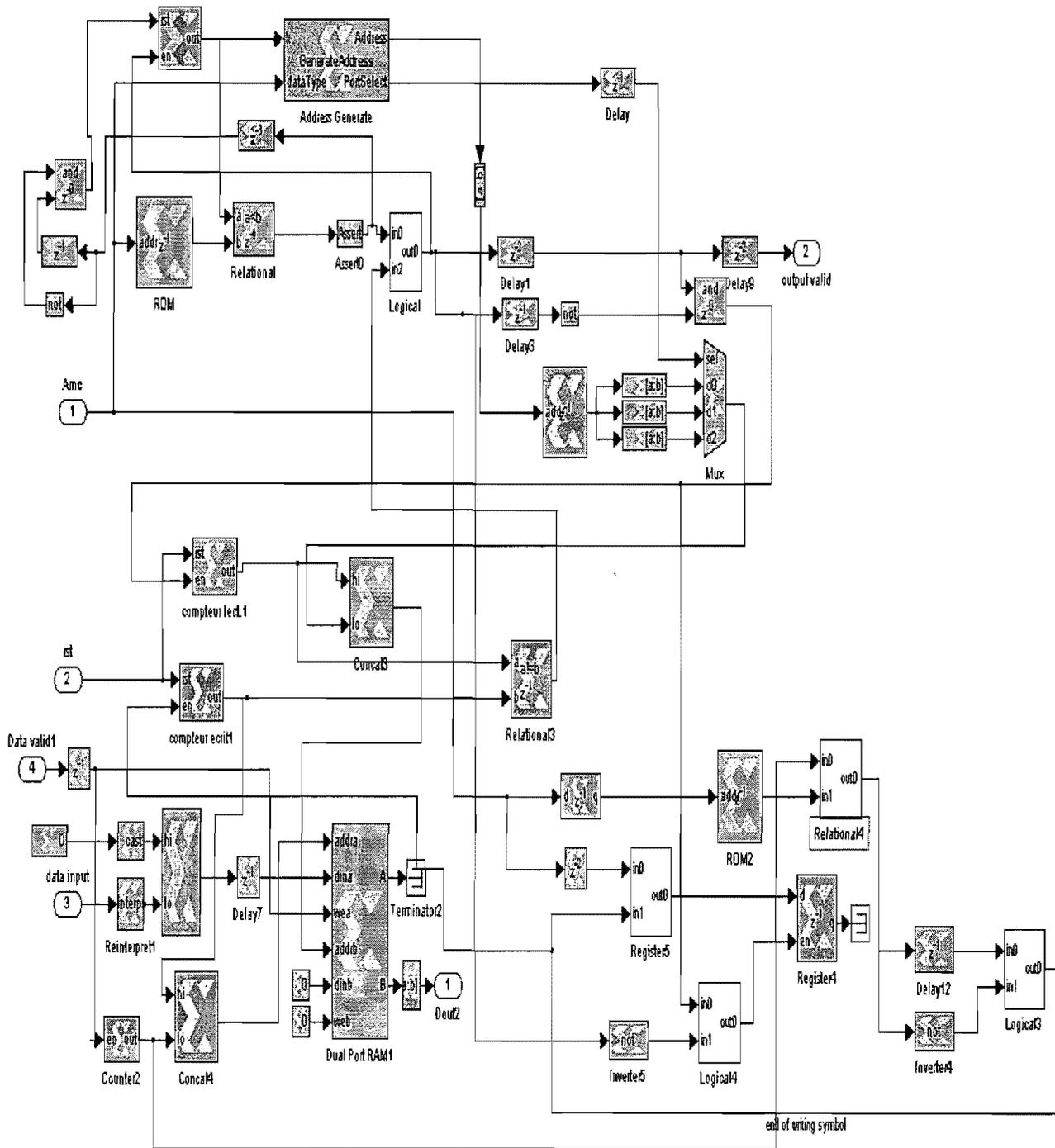
- [28] U. Tureli, H. Liu, "Blind carrier synchronization and channel identification for OFDM communications", Proceedings of the IEEE Acoustics, Speech and Signal Processing conference, 1998
- [29] E. Jaffrot, M. Siala, "Turbo channel estimation for OFDM systems on highly time and frequency selective channels", IEEE Computer Society, 2000
- [30] Wookwon Lee, "On channel estimation for OFDM systems in multipath environments with relatively large delay spread", 57th IEEE Semiannual Vehicular Technology Conference, 2003
- [31] W. Ng, V. Dubey, "On coded pilot based channel estimation for OFDM in very fast multipath fading channel", Fourth International Conference on Information, Communications and Signal Processing, 2003.
- [32] Y. Zhang, H. Chen, "Mobile WiMAX: Toward Broadband Wireless Metropolitan Area Networks", ISBN: 9780849326240, 2007.
- [33] Ian A. Glover et Peter M. Grant, Digital Communications, 2nd edition. Person Education, ISBN 0130893994, 2004.
- [34] Andrew J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", IEEE Transactions on Information Theory, 1967.
- [35] Charles L.H., "Error-Control Block Codes for Communications Engineers", Artech House, 2000. ISBN: 1-58053-032-X.
- [36] W. Cary Huffman, "Fundamentals of Error Correcting Codes", Vera Pless, 2003, ISBN: 9780521782807.
- [37] Shafik, M., El Badawy, H.M., Selim, G., "Simulation and assessment of WiMAX network performance in different environments," 3rd IEEE/IFIP International Conference in Central Asia on Internet, December 2007.
- [38] William H. Tranter., "Principles of Communication Systems Simulation: with Wireless Applications," Virginia Polytechnic Institute, Prentice Hall, 2004.
- [39] Dr. B.-P. Paris, Simulation of Wireless Communication Systems using MATLAB, fall 2007.
- [40] Alim O.A., Elboghdady N., Ashour M.M., Elaskary A.M., "Simulation of channel estimation and equalization for WiMAX PHY layer in Simulink". International Conference on Computer Engineering & Systems, November 2007.

- [41] Nicolas GRESSET, « Nouvelles techniques de codage spatio-temporel avec des modulations codées à bits entrelacés, » thèse de doctorat, Ecole nationale supérieure de télécommunication, décembre 2004.
- [42] Abdelali El Khettabi, « Conception du système de transmission OFDM codé pour les applications à haut débit, » Thèse M. Ing., École de technologie supérieure, 2008.
- [43] Alim O.A., Abdallah H.S., Elaskary A.M, “Simulation of WiMAX Systems”. Communications Workshop LCW 2008, IEEE Lebanon.
- [44] Simulink: communications and electronics Application examples, demos and recorded webinars. <http://www.mathworks.com/products/simulink/demos.html>.
- [45] The MathWorks - HIPERMAN-WiMAX Physical Layer A case study in creating an executable specification using MATLAB® and Simulink®.
- [46] Khan, M.N., Ghauri, S., “The WiMAX 802.16e physical layer model”, International Conference on Wireless, Mobile and Multimedia Networks, 2008.
- [47] Xilinx® Inc., “System Generator for DSP user guide”, March 2008.
- [48] Ian Grout, “Digital Systems Design with FPGAs and CPLDs”, University of Limerick, 2008, ISBN: 978-0-7506-8397-5.
- [49] Evgeni Perelroyzen,” Digital Integrated Circuits: Design-for-test Using Simulink and Stateflow”, CRC Press, 2006, ISBN 0849330572.
- [50] Ying Yi et Woods, R., “FPGA-based system-level design framework based on the IRIS synthesis tool and System Generator”, IEEE International Conference on Field-Programmable Technology, 16-18 Dec. 2002 Page(s): 85 – 92.
- [51] Ownby, M. et Mahmoud, W.H. “A design methodology for implementing DSP with Xilinx® System Generator for MatLab®”, Proceedings of the 35th Southeastern Symposium on System Theory, 16-18 March 2003 Page(s): 404 – 408.
- [52] The website of the company Xilinx: videos, webcast, and documentation <http://www.xilinx.com/>
- [53] L. Dutrieux, D. Demigny, “Logique programmable”, Paris: Editions, 2000.
- [54] Xilinx® Inc., “XILINX VIRTEX-II SERIES FPGAs”, 2003 <http://www.xilinx.com/publications/matrix/virtexmatrix.pdf>
- [55] Danesh Tavana, Wilson Yee, Steve Young, and Bradly Fawcett, “Logic Block and Routing Considerations for a New SRAM-Based FPGA Architecture” Proceedings of the IEEE Custom Integrated Circuits Conference, 1995.
- [56] Xilinx® Inc,” Virtex™-II Platform FPGAs: Complete Data Sheet”, 2003.

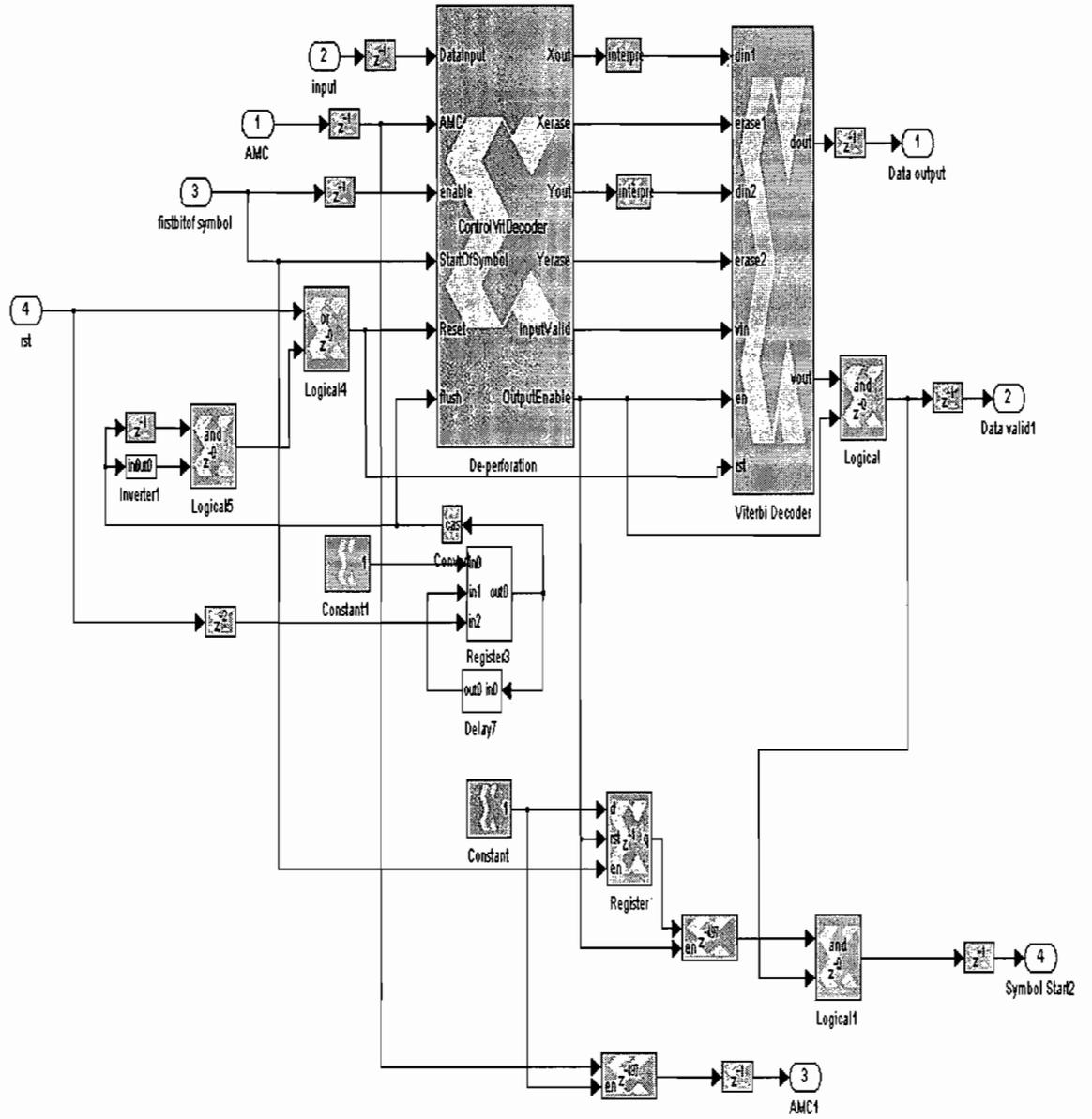
- [57] K. Tae-Hwan, I. Park, "Low-Power and High-Accurate Synchronization for IEEE 802.16d Systems", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2008
- [58] Y Chunlin, F. Jiayi, T. Youxi, L. Shaoqian, L. Yingtao, "OFDM synchronization using PN sequence and performance", *IEEE Proceedings on Personal, Indoor and Mobile Radio Communications*, 2003
- [59] W. Hui, H. Qingsheng, W. Xiaoyan, "A New OFDM Synchronization Scheme for PLC", *International Seminar on Future Information Technology and Management Engineering*, 2008
- [60] M. Ohta, K. Mizutani, N. Fujita, K. Yamashita, "Complexity suppression of neural networks for PAPR reduction of OFDM signal and its FPGA implementation", *IEEE International Joint Conference on Neural Networks*, 2008
- [61] J. Yang, L. Chen, W. Mao, "A Modified Selected Mapping Technique to Reduce the Peak-to-Average Power Ratio of OFDM Signal", *IEEE Transactions on Consumer Electronics*, 2007.
- [62] B. Kang, H. Ryu, B. Sang; "A PAPR Reduction Method using New ACE (Active Constellation Extension) with Higher Level Constellation", *IEEE International Conference on Signal Processing and Communications*, 2007.
- [63] T. Yücek, H. Arslan, "Time Dispersion and Delay Spread Estimation for Adaptive OFDM Systems", *IEEE transactions on vehicular technology*, 2008
- [64] Chitranjan Kumar Singh, "DESIGN OF HIGH PERFORMANCE MIMO RECEIVER: ALGORITHMS AND VLSI ARCHITECTURES", these de doctorat, The University of Texas at Dallas, 2008.
- [65] <http://www.mirabilisdesign.com/>
- [66]Filippo Tosato et Paola Bisaglia "Simplified Soft-Output Demapper for Binary Interleaved COFDM with Application to HIPERLAN/2", *IEEE International Conference on Communications*, 2002.
- [67] Xilinx logicore, "Reed-Solomon Encoder v5.0", 2005
- [68] Afshin Niktash, "RECFEC: A reconfigurable forward error correction engine", thèse de doctorat, Université de California Irvine, 2007.
- [69] D. Haque, R. Yasmin, S Ullah, M. Rashidul hasan, "performance evaluation of a Concatenated interleaved forward error correction scheme based orthogonal frequency division multiplexing", *Asian journal of information technology*, ISSN : 1682-3915, 2008.
- [70] H. Zhou, S. Yin, J. Zheng, H. Qiu, "Performance Analysis of Error-Correcting Codes in Ultra-Wideband system", *International Conference on Communication Technology*, 2006.

- [71] S. G. Lee, "Performance of concatenated FEC under fading channel in wireless-MAN OFDM system", *International Conference on Advanced Information Networking and Applications, AINA 2005*.
- [72] C. Stevenson, G. Chouinard, Lei Zhongding, Hu Wendong, S. Shellhammer, W. Caldwell, "IEEE 802.22: The first cognitive radio wireless regional area network standard", *Communications Magazine, IEEE, January 2009*.
- [73] Justin Thiel, "Metropolitan and Regional Wireless Networking: 802.16, 802.20 and 802.22". <http://www.cs.wustl.edu/~jain/cse574-06/ftp/wimax.pdf>
- [74] M. Gohkale et P. Graham, "Reconfigurable computing: Accelerating Computation with Field-Programmable Gate Arrays", Springer, 2005.

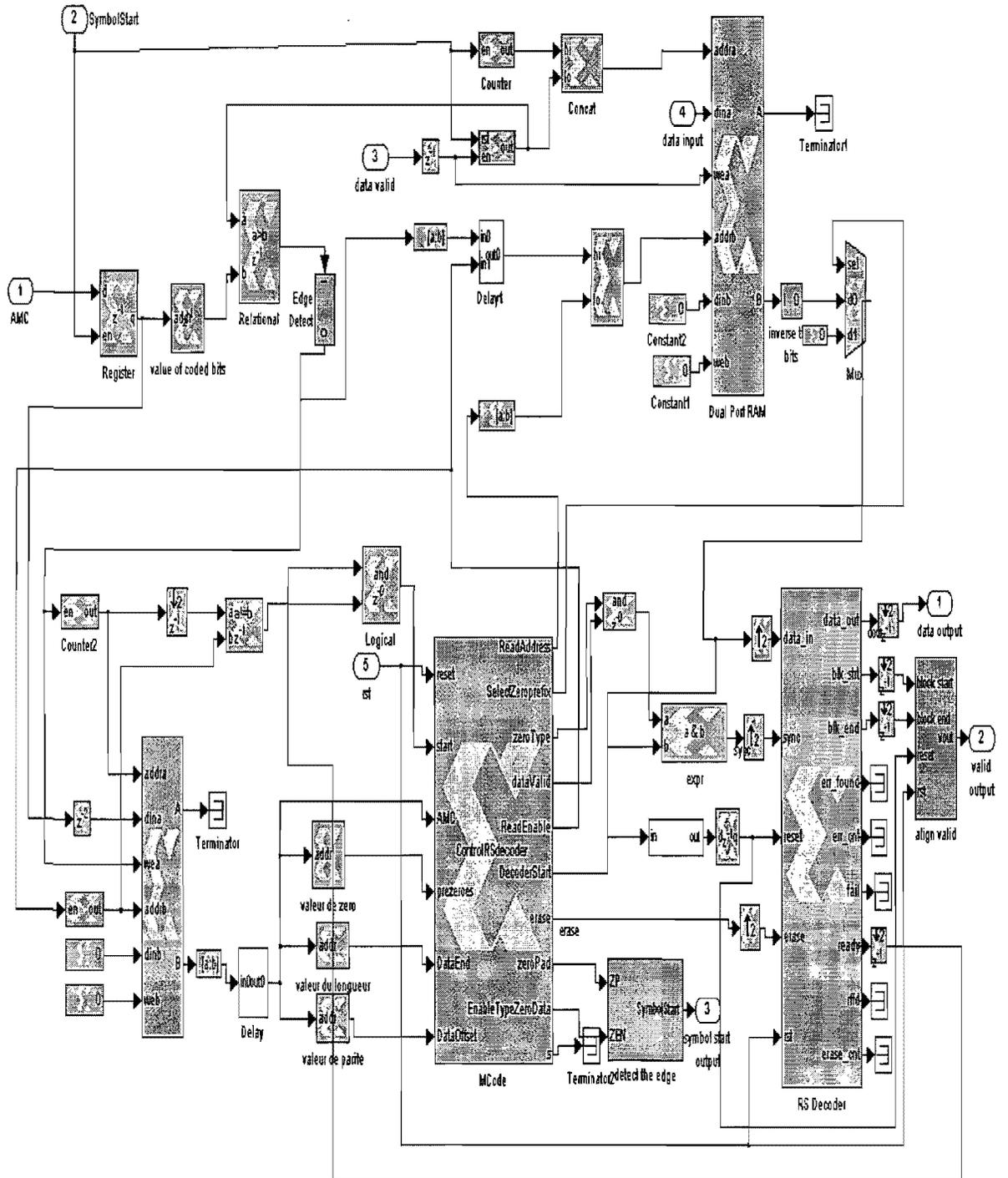
A.3 Circuit du déentrelaceur



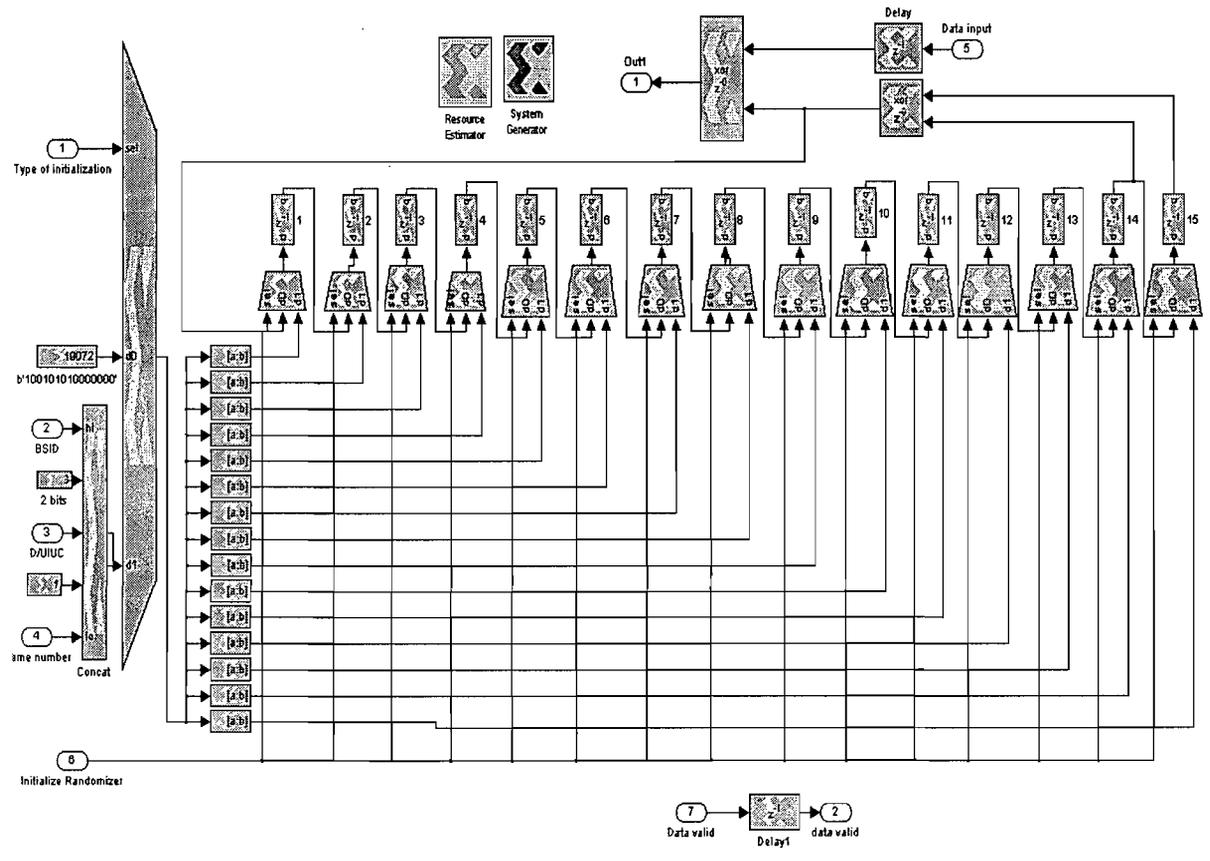
A.4 Décodeur de Viterbi



A.5 Décodeur Reed-Solomon



A.6 Démélangeur



Annexe B : Algorithme du rapport de vraisemblance logarithmique simplifié

Le démodulateur prend $\alpha + \beta i$ (sortie du FFT) comme un signal d'entrée, puis reconstruit 1, 2, 4 ou 6 bits pour les schémas de modulation BPSK, QPSK, 16 QAM, 64 QAM respectivement. Puisque le canal de transmission ajoute du bruit sur le signal transmis, $\alpha + \beta i$ ne sera pas sur le point de la constellation exacte. Par conséquent, le démodulateur doit estimer la valeur possible du signal original pour pouvoir récupérer les bits portés par ce signal. Généralement, il existe deux stratégies pour la récupération des bits portés par le signal transmis : la décision dure et la décision douce. Si les bits sont quantifiés en deux niveaux, zéro ou un, le processus de démodulation est appelé décision dure (hard decision). Par contre, dans la décision douce le démodulateur attribue une valeur de confiance à chaque bit de sortie, donc ces bits seront quantifiés en plus que deux niveaux. En effet, le processus du décodage avec décision douce est bien plus performant que celui avec la décision dure, cependant la mise en œuvre du démodulateur ainsi que le décodeur de Viterbi avec décision douce est plus complexe et coûteuse. Dans ce contexte, l'étude théorique, l'analyse détaillée et la performance du processus de démodulation avec décision douce ainsi que la performance du décodeur de Viterbi sont largement couverts dans divers livres et articles [34-36]. Dans ce travail, le démodulateur avec décision douce a été implémenté. L'algorithme du rapport de vraisemblance logarithmique simplifié (simplified log likelihood ratio) [66] a été utilisé pour récupérer les bits doux. Soit b_0, b_1, b_2, b_3, b_4 et b_5 les bits doux correspondants au signal $\alpha + \beta i$, en utilisant l'algorithme du rapport de vraisemblance logarithmique simplifié ces bits sont donnés par les équations suivantes :

$$b_0 = |\beta| \quad [\text{B.1}]$$

$$b_1 = -|\beta| + 4 \quad [\text{B.2}]$$

$$b_2 = -||\beta| - 4| + 2 \quad [\text{B.3}]$$

De même

$$b_3 = |\alpha| \quad [\text{B.4}]$$

$$b_4 = -|\alpha| + 4 \quad [\text{B.5}]$$

$$b_5 = -||\alpha| - 4| + 2 \quad [\text{B.6}]$$

Annexe C : Effet des modules de la chaîne de transmission sur une trame exemple

Nous supposons que la trame sera transmise suivant profil QPSK 3/4 et que les paramètres BSID, D/IUIC et l'index de la trame de données sont égales respectivement à 1,7 et 1.

La trame à transmettre (35 octets):

[45 29 C4 79 AD 0F 55 28 AD 87 B5 76 1A 9C 80 50 45 1B 9F D9 2A 88 95 EB AE B5 2E 03 4F 09 14 69 58 0A 5D]

La trame après le mélangeur (35 octets):

[D4 BA A1 12 F2 74 96 30 27 D4 88 9C 96 E3 A9 52 B3 15 AB FD 92 53 07 32 C0 62 48 F0 19 22 E0 91 62 1A C1]

La trame après le RS (40 octets):

[49 31 40 BF D4 BA A1 12 F2 74 96 30 27 D4 88 9C 96 E3 A9 52 B3 15 AB FD 92 53 07 32 C0 62 48 F0 19 22 E0 91 62 1A C1 00]

La trame après le CC et le perforateur (48 octets):

[3A 5E E7 AE 49 9E 6F 1C 6F C1 28 BC BD AB 57 CD BC CD E3 A7 92 CA 92 C2 4D BC 8D 78 32 FB3 BF DF 23 ED 8A 94 16 27 A5 65 CF 7D 16 7A 45 B8 09 CC]

La trame après l'enterlaceur (48 octets):

[77 FA 4F 17 4E 3E E6 70 E8 CD 3F 76 90 C4 2C DB3 F9 B7 F13 43 6C F19A BD ED OA 1C D8 1B EC 9B 30 15 BA DA 31 F5 50 49 7D 56 ED B4 88 CC 72 FC 5C].

La trame après le modulateur (192 nombres complexes):

$[\sqrt{2}-\sqrt{2}i, -\sqrt{2}-\sqrt{2}i, \sqrt{2}-\sqrt{2}i, -\sqrt{2}-\sqrt{2}i, -\sqrt{2}-\sqrt{2}i, -\sqrt{2}-\sqrt{2}i, -\sqrt{2}+\sqrt{2}i, -\sqrt{2}+\sqrt{2}i, \sqrt{2}-\sqrt{2}i, \sqrt{2} + \sqrt{2}i, -\sqrt{2}-\sqrt{2}i, -\sqrt{2}-\sqrt{2}i, \sqrt{2} + \sqrt{2}i, \sqrt{2}-\sqrt{2}i, \sqrt{2}-\sqrt{2}i, -\sqrt{2}-\sqrt{2}i, \sqrt{2}-\sqrt{2}i, \sqrt{2} + \sqrt{2}i, -\sqrt{2}-\sqrt{2}i, -\sqrt{2} + \sqrt{2}i, \sqrt{2} + \sqrt{2}i, -\sqrt{2}-\sqrt{2}i, -\sqrt{2}-\sqrt{2}i, -\sqrt{2} + \sqrt{2}i, -\sqrt{2}-\sqrt{2}i, -\sqrt{2} + \sqrt{2}i, \sqrt{2}-\sqrt{2}i, -\sqrt{2} + \sqrt{2}i, \sqrt{2}-\sqrt{2}i, -\sqrt{2}-\sqrt{2}i, \sqrt{2} + \sqrt{2}i, \sqrt{2} + \sqrt{2}i, -\sqrt{2}-\sqrt{2}i, -\sqrt{2} + \sqrt{2}i, -\sqrt{2} + \sqrt{2}i, \sqrt{2} + \sqrt{2}i, -\sqrt{2}-\sqrt{2}i, \sqrt{2} + \sqrt{2}i, -\sqrt{2}-$

0.0298i, -0.0435 - 0.0429i, 0.0425 + 0.0294i, 0.0132 + 0.0093i, -0.0537 + 0.0183i, 0.0450 -
 0.0475i, -0.0296 + 0.0298i, -0.0164 + 0.0404i, 0.0495 - 0.0756i, -0.0150 + 0.0127i, 0.0280
 + 0.0035i, -0.0748 + 0.0675i, 0.0812 - 0.0413i, -0.0198 - 0.0625i, -0.0467 + 0.0477i,
 0.0031 + 0.0436i, 0.0010 - 0.0344i, 0.0359 - 0.0101i, 0.0259 - 0.0411i, -0.0499 + 0.0229i,
 0.0348 + 0.0699i, -0.0147 - 0.0005i, -0.0256 - 0.0402i, 0.0052 - 0.0316i, -0.0245 -
 0.0203i, 0.0257 - 0.0164i, 0.0664 + 0.1318i, -0.0410 - 0.0051i, -0.0019 - 0.1179i, -0.0348
 + 0.0949i, -0.0164 - 0.0751i, 0.0476 + 0.0126i, 0.0026 + 0.0216i, -0.0013 + 0.0094i, -
 0.0150 - 0.0148i, 0.0117 - 0.0152i, 0.0044 + 0.0696i, -0.0325 - 0.0450i, 0.0118 +
 0.0029i, 0.0003 - 0.0806i, 0.0199 + 0.0866i, -0.0098 + 0.0274i, 0.0507 - 0.0568i, -0.0520 +
 0.0139i, -0.0469 + 0.0007i, 0.0253 - 0.0133i, 0.0024 + 0.0316i, 0.0371 + 0.0070i, 0.0349 -
 0.0495i, -0.0253 + 0.0296i, -0.0510 - 0.0063i, 0.0036 - 0.0192i, -0.0146 + 0.0277i, 0.0305
 - 0.0025i, 0.0280 - 0.0119i, -0.0007 + 0.0548i, -0.0438 - 0.0393i, 0.0551 - 0.0325i, -0.0357
 - 0.0224i, -0.0182 + 0.0443i, -0.0391 + 0.0271i, 0.1075 - 0.0092i, -0.0264 + 0.0130i, -
 0.0253 - 0.0196i, -0.0015 - 0.0651i, -0.0116 + 0.0563i, -0.0044 + 0.0009i, 0.0529 +
 0.0264i, -0.0527 - 0.0424i, 0.0449 + 0.0516i, -0.0085 - 0.0697i, -0.0256 + 0.0410i, -
 0.0071 - 0.0273i, -0.0018 + 0.0279i, 0.0011 + 0.0075i, 0.0499 - 0.0232i, -0.0169 +
 0.0131i, -0.0230 + 0.0189i, 0.0158 - 0.0442i, 0.0110 + 0.0011i, -0.0397 + 0.0019i, -0.0099
 + 0.0089i, 0.0109 + 0.0248i, 0.0293 + 0.0525i, 0.0369 - 0.0416i, -0.0405 - 0.1178i, -
 0.0361 + 0.0721i, 0.0386 + 0.0047i, 0.0312 + 0.0549i, -0.0716 - 0.0513i, -0.0188 +
 0.0361i, 0.0427 - 0.0197i, 0.0435 - 0.0056i, 0.0012 - 0.0254i, -0.0408 + 0.0501i, 0.0002 -
 0.0391i, 0.0002 - 0.0548i, -0.0323 + 0.0722i, 0.0321 + 0.0677i, 0.0161 - 0.0812i, 0.0228 +
 0.0514i, -0.0537 - 0.0408i, 0.0207 - 0.0567i, -0.0152 + 0.0311i, -0.0083 + 0.0197i, 0.0529
 + 0.0021i, -0.0217 + 0.0187i, -0.0052 + 0.0100i, -0.0005 + 0.0332i]

Annexe D : Glossaire

Bitstream : données binaires utilisées pour configurer le FPGA.

Digital Locked Loop : un circuit qui permet la multiplication et la division des signaux de l'horloge dans un FPGA.

Intervalle de l'usage du code (interval usage code, D/UIUC) : une valeur de 4 bits qui désigne un profil de modulation et de codage spécifique.

LUT (Look Up Table) : circuit qui réalise des fonctions combinatoires dans un FPGA.

Netlist : Une liste de portes logiques et de leurs interconnexions qui constituent un circuit.

Propriété intellectuelle (intellectual property, IP) : désigne les droits légaux qui résultent d'une activité intellectuelle dans les domaines industriels, scientifiques, littéraires et artistiques. Ces activités intellectuelles incluent, parmi d'autres, les travaux littéraires, les découvertes scientifiques et les designs industriels.

RAM à double port (Dual Port RAM, DPRAM) : est un type de RAM qui permet un accès simultané à la RAM.

Rapport signal / bruit (signal to noise ratio, SNR) : désigne la qualité d'une transmission d'information par rapport aux parasites. Autrement, SNR désigne le rapport entre la grandeur d'un signal (information utile, significative) et celle du bruit (information inutile, non significative).

Rafale (burst) : est un groupe de trames transmises suivant le même profil de modulation et de codage (par exemple BPSK1/2 ou QPSK3/4).

Taux d'erreur binaire (BER, bit error rate) : BER est le pourcentage de bits erronés, divisé par le nombre total de bits qui ont été transmis, reçus ou traités sur une période donnée.

Tranches (Slices) : un terme introduit par Xilinx qui spécifie l'unité de traitement de base dans les FPGA de Xilinx. Généralement, cette unité comprend 2 LUT, 2 bascules D, des multiplexeurs, et des portes logiques arithmétiques.

Symbole : désigne une trame de bits à transmettre. Le nombre de bits dans cette trame est égale à 192, 384, 768 et 1152 pour les profils BPSK, QPSK, 16-QAM et 64-QAM respectivement.

Symbole modulé : est un nombre complexe qui représente un groupe de bits. Ce groupe de bits est égale 1, 2, 4 ou 6 pour BPSK, QPSK, 16-QAM, 64-QAM.