

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي



BADJI MOKHTAR ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA

جامعة باجي مختار - عنابة
Année 2019

Faculté des sciences de l'ingéniorat
Département d'électronique

THÈSE

Présentée en vue de l'obtention du diplôme de DOCTORAT DE TROISIEME CYCLE

Étude et Optimisation de Système de Transmission de Données à Haut Débit en Télécommunication

Option : Télécommunications avancées

Par :

BOUDJEMA Souhila

Directeur de thèse :

Pr. SAOUCHI Kaddour

Université Badji Mokhtar d'Annaba

DEVANT LE JURY

PRESIDENT:

Pr. TOUMI Salah

Université Badji Mokhtar d'Annaba

EXAMINATEURS:

Pr. BENNACER Layachi

Université de Guelma

Pr. BOUGHAZI Mohammed

Université Badji Mokhtar d'Annaba

Dr. MASSAOUDI Kamel

Université de Souk Ahras

Remerciements

*Je remercie dieu le tout puissant qui m'a donné force, volonté et courage afin
d'élaborer ce mémoire.*

*Je tiens à remercier mon directeur de thèse Monsieur SAOUCHI Kaddour pour
le soutien moral et intellectuel et la haute compréhension qu'il m'a apporté tout
au long de mon travail, je lui exprime ici mes sincères et profondes
reconnaisances.*

*Mes remerciements s'adressent à l'ensemble des membres de mon jury d'avoir
accepté de juger et d'évaluer ces travaux de thèse. Je remercie sincèrement
TOUMI Salah, Professeur à l'université d'Annaba, de m'avoir fait l'honneur de
présider mon jury ; qu'il trouve ici l'expression de mon profond respect.*

*Je remercie également BENNACER Layachi, Professeur à l'université de
Guelma, BOUGHAZI Mohammed, Professeur à l'université d'Annaba ainsi que
MASSAOUDI Kamel, Docteur à l'université de Souk - Ahras d'avoir accepté de
me donner un peu de leurs précieux temps pour faire part de mon jury en tant
qu'examineurs.*

*Je remercie toutes les personnes qui ont contribué de près ou de loin à la
réalisation de ce Mémoire spécialement Mlle Handouzi Wahida.*

*Enfin, je remercie très vivement les personnes
Qui ont accepté de juger ce travail.*

Dédicaces

*Avec un immense plaisir, je dédie ce travail à
Mes très chers père et mère qui n'ont jamais cessé de
m'encourager et qui ont su me donner du baume au cœur
dans les moments les plus difficiles et de doutes.*

*A la mémoire de mes frères : Kamel, Karim et ma belle-sœur
Naïma*

A ma fille : Asma

Mon mari Miloud et sa famille

*Manissa, Djamila, Leïla, Samir et chaïma, Amir, Walid et
Samira*

Ases parents, Zina et Abd ALLAH

A mes sœur : Fatma, Nacira.

*A mes frère : Noureddine, Abd el Malek, Hichem, Chaouki,
Hamza*

Ma belle-sœur Saïda

A toute la famille sans exception.

*A mes amis : Handouzi Wahida, Sana, Latifa, Nadjiba,
Saïda, Narjiss.*

A tous mes fidèles professeurs.

الملخص

الهدف من هذا العمل هو دراسة وإختبار المضمنات الرقمية للاتصالات. لقد ركزنا على التحديات التي تفرضها المضمنات الرقمية من خلال تطوير المضمنات عالية السرعة للسعة و الطور بتوفير أساليب أبسط تزيد من سرعة المعالجة بشكل أفضل من تلك المتوفرة بالفعل لذلك. من خلال هذه الدراسة، يتمثل الهدف الرئيسي في العثور على أفضل نقل للبيانات وكذلك الإنتاجية المثلى.

تحقيقا لهذه الغاية ، قمنا بإجراء إختبار للبنيات الثلاثة من المضمنات الرقمية المختلفة وعمليات التنفيذ المقابلة لها على دارة FPGA من النوع Xilinx Virtex5 لتقييم أدائها.

سمحت لنا هذه الدراسة باقتراح وتطوير نظام يعتمد على خوارزمية CORDIC المعدلة ونفذ على دارة FPGA قابلة للبرمجة وبالتالي تشير نتائج الاختبار التي تم الحصول عليها إلى أن اقتراحنا فعال من حيث حساب البيانات وتداولها (خوارزمية CORDIC) ، على وجه الخصوص لتوليد الإشارات الجيبية في المضمنات الرقمية QAM.

الكلمات المفتاحية :

تعديل CORDIC, المضمنات الرقمية, اتصالات, مضمن السعة و الطور, نقل للبيانات, FPGA , التردد الأقصى.

Résumé

L'objectif de ce travail est l'étude et la simulation des modulateurs numériques destinés aux télécommunications. Nous nous sommes concentrés sur les défis engendrés par les modulateurs numériques en développant le modulateur à haut débit d'amplitude et de phase (QAM) dont on propose des méthodes plus simples qui accélèrent mieux les vitesses de traitement qu'avec celles déjà existantes. Donc par cette étude, le but principal est de trouver la meilleure transmission de données ainsi que le débit optimal.

A cet effet, nous avons entrepris la simulation d'architectures des trois modulateurs numériques différents et à leurs implémentations correspondantes sur un circuit FPGA de type Xilinx Virtex5 pour évaluer leurs performances.

Cette étude nous a permis de proposer et de développer un système basé sur l'algorithme CORDIC modifié et implanté sur un circuit programmable FPGA. Ainsi les résultats de simulation obtenus indiquent que notre proposition est efficace en termes de calcul et de transmission de données, en particulier pour la génération des signaux sinusoïdaux dans les modulateurs numériques QAM.

Mots-clés : CORDIC modifié, modulateurs numériques, télécommunications, modulateur d'amplitude et de phase, transmission de données, FPGA, Fréquence maximale.

Abstract

The objective of this work is the study and the simulation of numerical modulators for telecommunications. We have focused on the challenges of numerical modulators by developing the High -speed Amplitude and Phase Modulator (QAM). For which simpler methods are proposed that accelerate processing speeds better than existing ones. So through this study, the overall goal is to find the best data transmission as well as the optimal bit rate.

For this purpose, we undertook the simulation of the architectures of three different numerical modulators and their corresponding implementations on a circuit FPGA of type Xilinx Virtex5 to evaluate their performance.

This study allowed us to propose and develop a system based on the CORDIC algorithm modified and implemented on a programmable FPGA circuit. Thus, the simulation results obtained indicate that our proposition is efficient in terms of calculation and data transmission (CORDIC algorithm), in particular for the generation of sinusoidal signals in the QAM numerical modulators.

Keywords: modified CORDIC, digital modulator, telecommunications, amplitude and phase modulator, data transmission, FPGA, maximum frequency, CORDIC algorithm.

Liste des acronymes

FPGA: Field Programmable Gate Array.

QAM: Quadrature Amplitude Modulation.

CORDIC: COordinate Rotation DIgital Computer.

ISE: Integrated Systems Europe.

VHDL: Very High Speed Integrated Circuit Hardware Description Language.

CAN: Convertisseur Analogique Numérique.

CNA : Convertisseur Numérique Analogique.

DSP: Digital Signal Processing.

WPAN: Wireless Personal Area Network.

WLAN: Wireless Local Area Network.

WMAN: Wireless Metropololitan Area Network.

WWAN : Wireless Wide Area Network.

OSI : Open System Interconnexion.

TCP/IP: Transmission Control Protocol/ Internet Protocol.

UDP: User Datagram Protocol.

ROM: Read Only Memory.

ASIC : Application Specific Integrated Circuit.

LUT: Look Up Table.

IOB: Input Output Bloc.

BER: Bit Error Rate.

SNR: Signal to Noise Ratio.

MHz: Méga Hertz

IEEE: Institute of Electrical and Electronics Engineers.

Std: Standard.

MDA: Modulations par Déplacement d'Amplitude.

ASK: Amplitude Shift Keying.

MDF: Modulation par Déplacement de Fréquence.

FSK: Frequency Shift Keying.

MDP: Modulation par déplacement de Phase.

PSK: Phase Shift Keying.

QAM: Quadrature Amplitude Modulation.

RTL: Register Transfer Level.

CPLD: Complex Programmable Logic Device.

PROM: Programmable Read Only Memory.

PAL: Programmables Array Logic.

PCB : Printed Circuit Board.

LUT: Look Up Tables.

VLSI: Very Large Scale Integration.

ASMBL: Advanced Silicon modular Block.

DCM: Digital Clock Manager.

PLL: Phase Locked Loop.

EPROM: Erasable Programmable Read Only Memory.

EEPROM: Electrical Erasable Programmable Read Only Memory.

SRAM: Static Random Mémory.

LAB: Logical Array Bloc.

LE: Logic Element.

CLB : Configurable Logic Bloc.

MATLAB: Matrix Laboratory.

Liste des figures

N°	Titre de la figure	Page
1.1	Schéma synoptique d'un système de transmission numérique.	04
1.2	Les réseaux sans fil.	09
1.3	Modèle de référence OSI.	10
1.4	Schéma synoptique de la couche session	13
1.5	Schéma synoptique de la couche présentation	14
1.6	Schéma synoptique de la couche application	15
1.7	Modèle de référence TCP/IP en parallèle avec le modèle OSI	17
2.1	Schéma bloc d'un modulateur	17
2.2	Exemple de signal modulant $e(t)$ et modulé ASK $s(t)$.	24
2.3	Exemple de signal modulant $e(t)$ et modulé FSK $s(t)$.	26
2.4	Exemple de signal modulant $e(t)$ et modulé PSK $s(t)$.	27
2.5	Modulation d'amplitude de deux porteuses en quadrature MAQ	28
2.6	principe de modulateur QAM.	29
2.7	Exemple d'un signal modulé QAM.	29
2.8	Représentation schématique de l'utilisation de la modulation QAM.	30
2.9	Constellation (BPSK), (16QAM), (32QAM), (64QAM).	32
2.10	Conversion des trames au format QAM 1024.	33
2.11	Propriété de symétrie dans la constellation QAM.	34
3.1	L'architecture d'une modulation numérique QAM.	34
3.2	Exemple du codeur.	35
3.3	Architecture de l'accumulateur de phase.	37
3.4	Schéma de l'accumulateur de phase avec le générateur de sinusoïde.	38
3.5	Décompositions de l'angle θ en plusieurs micro-rotations.	40
3.6	Le processeur CORDIC itératif.	43
3.7	Le processeur CORDIC pipeline.	44
3.8	Le processeur CORDIC pipeline sans accumulateur de phase.	44
4.1	Structure d'un programme sous VHDL	50
4.2	Les étapes d'implémentation d'un circuit sur FPGA sous l'outil ISE.	51
4.3	Flot de conception d'un circuit intégré en VHDL.	54
4.4	Les différents circuits programmables.	56

4.5	Exemple d'implémentation sur LUT	58
4.6	Exemple d'implémentation sur des multiplexeurs.	59
4.7	Critères de choix du circuit logique programmable FPGA.	60
4.8	Structure générale des FPGAs	64
4.9	Architecture d'un bloc logique.	65
3.10	Cellule logique d'un FPGA.	66
4.11	Blocs d'entrée/sortie.	67
4.12	Les interconnexions.	68
4.13	Les deux grandes familles de circuit FPGA.	69
4.14	Statistiques marchées occupés par vendeurs des FPGAs	69
4.15	Signal continu.	71
4.16	Signal échantillonné.	71
4.17	Résultat obtenu sous Matlab de l'échantillonnage d'un signal continu.	72
4.18	Représentation de la condition d'échantillonnage (Shannon).	73
4.19	Vue interne l'implémentation d'un modulateur QAM1	75
4.20	Vue externe (Schémas RTL de l'implémentation sur FPGA Virtex 5 du codeur.	75
4.21	Vue interne l'implémentation du codeur.	76
4.22	Vue interne de l'implémentation d'accumulateur de phase.	76
4.23	Vue externe (schémas RTL) de l'implémentation du CORDIC itératif à 8 étages.	77
4.24	Vue interne de l'implémentation du CORDIC itératif à 8 étages.	77
4.25	Résultat de simulation de l'algorithme CORDIC itératif à étages.	78
4.26	Vue interne de l'implémentation du modulateur QAM ₂	78
4.27	Vue interne de l'implémentation du CORDIC pipeline.	79
4.28	Résultat de simulation d'algorithme CORDIC pipeline.	79
4.29	Vue interne de l'implémentation d'un modulateur QAM3	80
4.30	Vue externe de l'implémentation du CORDIC pipeline sans accumulateur de phase.	81
4.31	Vue interne de l'implémentation sur FPGA Virtex 5 du CORDIC pipeline sans accumulateur de phase.	81
4.32	La fréquence d'échantillonnage en différents architectures.	82

4.33	Le nombre de cellules logique sur le circuit FPGA.	83
4.34	Vue externe de l'implémentation d'un modulateurs QAM ₁ .	84
4.35	Vue interne de l'implémentation d'un modulateurs QAM ₁ .	84
4.36	Résultats de simulation d'un modulateur numérique d'amplitude et de phase à base d'opérateur CORDIC pipeline.	85

Liste des tableaux

N°	Titre du tableau	Page
3.1	Mode de fonctionnement d'un processeur CORDIC.	41
3.2	Indique les différences entre les trois algorithmes	45
3.3	Étude comparative entre trois modulateurs QAM.	46
4.1	Caractéristiques du langage VHDL.	51
4.2	Présentation les avantages et inconvénients FPGA et DSP	57
4.3	Évolution des FPGAs XILINX famille Virtex.	60
4.4	Comparaison entre (Virtex5-LX50 et Virtex5-VLX110T)	61
4.5	Variations des paramètres du générateur de sinusoïde à base d'un CORDIC pipeline.	
4.6	Variations des performances obtenue en utilisant le processeur CORDIC pipeline sans accumulateur de phase.	80
4.7	résultats comparatifs des générateurs de sinusoides considérés pour 8 bits	82
4.8	Résultats de synthèse des architectures proposées en 16 bits.	85

Table des matières

Remerciements	I
Dédicaces	II
الملخص	III
Résumé	IV
Abstract	V
Liste des Acronymes	VI
Liste des figures	VIII
Liste des tableaux	XII
Introduction générale	01
Chapitre 1: Généralités sur la transmission numérique de l'information	
1. Introduction	04
2. Modèle de Transmission de l'information	05
2.1. Système de transmission de l'information sans fil (réseaux sans fil)	06
2.1.1. Techniques de transmission dans les réseaux sans fil	06
2.1.2. Avantages des réseaux sans fil	07
2.1.3. Domaines d'application	07
2.1.4. Les catégories de réseaux sans fil	07
a. Les réseaux personnels sans fil (WPAN: Wireless Personal Area Network)	07
b. Réseaux locaux sans fil (WLAN: Wireless Local Area Network)	08
c. Les réseaux métropolitains sans fil (WMAN : Wireless Metropololitan Area Network)	08
d. Les réseaux étendus sans fil (WWAN : Wireless Wide Area Network)	08
2.2. Les modèles de référence	09
2.2.1. Le modèle OSI	09
2.2.1.1 Rôle et caractéristiques de chacune des 7 couches du modèle OSI	11
2.2.2. Le modèle TCP/IP	16
3. Les avantages et les inconvénients des systèmes numériques	17
4. Conclusion	21
Chapitre II : Quelques différents types de modulations fréquemment utilisés	
1. Introduction	22
2 Les avantages Les avantages et les inconvénients de la modulation numérique	22
3 Critères de choix d'une modulation numérique	23

3.1. Taux d'erreur sur les bits	23
3.2. L'efficacité spectrale	23
3.3. Rapport Eb/No	23
3.4. Débit binaire	24
4. Différents types de modulations numériques	24
4.1. Objectif de la modulation	
4.2. Modulation par déplacement d'amplitude MDA	24
4.2 .1. Les avantages et les inconvénients de la modulation numérique ASK	25
4.2 .2. Application ASK	25
4. 3. Modulation par déplacement de fréquence MDF	25
4.3.1. Les avantages et les inconvénients de la modulation numérique FSK	26
4.3.2. Utilisation du FSK	26
4.4. Modulation par déplacement de phase MDP (Phase Shift Keying PSK)	26
4. 4.1. Les avantages et les inconvénients de la modulation numérique PSK	27
4.4.2 Utilisation de la modulation numérique PSK	27
4.5. Modulation d'amplitude de deux porteuses en quadrature MAQ	28
4.5.1. Définition	28
4.5.2. Principe de la modulation QAM	28
4.5. 3. Domaine d'application de la modulation QAM	30
4.5.4. Schéma bloc d'un modulateur	31
4.5. 5. Les avantages et les inconvénients de QAM	31
4.5.6. Conversion des trames au format QAM 1024	31
5. Conclusion	32
 Chapitre III: Architecture du modulateur QAM à base du processeur CORDIC	
1. Introduction	34
2. synthèse numérique de sinusoïdes existant	34
3 L'architecture d'une modulation numérique QAM	34
3.1. Le codeur	35
3.2. L'accumulateur de phase	37
3.3. Le convertisseur numérique analogique associé	38
3. 4. Le générateur de sinusoïde	38
3. 4. 1. L'algorithme CORDIC	39
a. Définition de l'algorithme CORDIC	39
b. Principe de l'algorithme CORDIC	39

c .Les différentes utilisations de CORDIC	40
d. Les modes de fonctionnement d'un processeur CORDIC	41
e. L'architecture de l'algorithme CORDIC	41
4. Les différentes techniques de la modulation QAM	41
4.1. Description du première technique QAM ₁	42
4.2. La deuxième technique QAM ₂	42
4.3. L'architecture QAM ₃	43
5. Conclusion	46

Chapitre IV

Simulation et implémentation du modulateur QAM adapté au haut débit

1. Introduction	
2. présentation du langage VHDL et le circuit FPGA : Partie A	48
A. Le langage VHDL	48
A.1. Définition	48
A.2. Structure d'un programme sous VHDL	49
A. 3. Description en VHDL de l'architecture	50
A.4. Caractéristiques du VHDL	51
A.5. Le flot de conception	52
A.5.1. Description du circuit	52
A.5.2. Vérification fonctionnelle par simulation	52
A.5.3. Synthèse	52
A.5.4. Implémentation	52
A.5.5. Vérification par simulation	53
A.5.6. Extraction des métriques d'implémentation	53
A. 5.7. Programmation et vérification de la puce	53
A.6. Présentation de l'environnement de développement (Xilinx ,ISE)	54
A.7. Les circuits logiques programmables FPGAs	55
4.7.1. Introduction	55
A.7.2. Les deux grandes familles architecturales de FPGA	58
A.7. 3. Critères de choix du circuit programmable FPGA	59
A.7.4. Evolution des FPGAs XILINX famille Virtex	60
4. 7.5. Etude de la famille VIRTEX 5 de type (XC5LX110T)	60
A.7. 6. Les domaines d'applications des FPGAs	61
A.7.7. Les technologies de FPGA	62

A.8. Technologie EPROM	62
A.8.1 Technologie EEPROM	62
A.8. 2. Technologie antifusible	62
A.8.3. Technologie Flash	63
A.8.4 Technologie SRAM (Static Random Mémory)	63
A.8.5. Architecture interne du FPGA	64
A.8.6. Les blocs logiques	65
A.8.7. Architecture d'une cellule logique	66
A.8. 8. Les éléments d'entrées/sorties	66
A.8.9. Les interconnexions	67
A.8. 10.Circuit d'horloge	68
A.8. 11. Les principaux fabricants de FPGA	68
A.8.12. Exemple du premier fabricant xilinx	70
A.9. Le logiciel MATLAB	70
A.9.1. Echantillonnage des signaux	71
A.9.1.1. Introduction	71
A.9.1.2. Le résultat en Matlab d'un signal échantillonné	72
A.9.1.3 Le théorème de l'échantillonnage	72
A.10. Simulation (Les TEST-BENCHS)	73
A.11. Consommation de puissance	74
3. Résultats de simulation et implémentation : PARTIE B	74
B.1. Simulation et implémentation du modulateur QAM	74
B.2. Résultats de l'implantation de trois modulateurs QAM sur le circuit FPGA	74
B.2.1. L'implémentation d'un modulateur numérique QAM ₁	75
B.2.2. Le codeur	75
B.2.3. L'accumulateur de phase	76
B.2.4.Générateur de sinusoïde basé sur CORDIC itératif à 8 étages	76
B.3. L'implémentation d'un modulateur numérique QAM ₂	78
B.4. L'implémentation d'un modulateur numérique QAM ₃	80
B.5. Résultats d'implémentation de deux algorithmes sur le circuit FPGA	82
B.6. Résultats d'implantation de trois modulateurs QAM	84
4. Conclusion	86
Conclusion générale	87
Références	89



***INTRODUCTION
GENERALE***

1. Introduction

Les systèmes numériques trouvent leurs applications de plus en plus nombreuses dans notre vie quotidienne. Cette dernière décennie, le monde rencontre une révolution technologique remarquable, en particulier, dans le domaine de la miniaturisation des appareils électroniques tels que des modems. Les nouvelles technologies offrent un taux d'intégration de plusieurs centaines de millions de transistors sur le même substrat. Cette évolution technologique d'intégration permet la réalisation des circuits de plus en plus complexes [1].

A l'heure actuelle, on retrouve sur le même circuit tout un ensemble de composants à savoir des processeurs, des mémoires, des accélérateurs matériels et des composants périphériques. L'ensemble de ces composants est interconnecté par un réseau de communication. Le développement de cette communication est appuyé sur les progrès rapidement réalisés dans le domaine du circuit numérique FPGA (Field Programmable Gate Array) de traitement des signaux. Ainsi, l'utilisation de solutions intégrées devient indispensable au fur et à mesure que le niveau de complexité des systèmes augmente et que le prix consenti par le consommateur diminue [2].

Les années 80 ont connu la naissance du tout premier FPGA. Ce composant de la logique programmable regroupe un nombre fixe de portes logiques de base qui peuvent ou non être utilisées, afin de former un circuit spécifique.

L'implémentation sur ce circuit numérique FPGA dépend également de la fréquence d'horloge et des quantités de cellules disponibles [3]. Généralement le modulateur numérique complexe QAM est composé de cinq parties principales : trois fonctionnent en mode numérique (le codeur, l'accumulateur de phase et le générateur de sinusoïde) et les deux autres fonctionnent en mode analogique (le convertisseur numérique analogique et le filtre passe bas) [4]. Le modulateur QAM a pour rôle de faire varier l'amplitude et la phase d'une onde sinusoïdale selon les symboles transmis [5].

Dans cette thèse on étudie des architectures de modulateurs compatibles avec l'évolution des débits et des fonctionnalités des protocoles actuels, Pour cela, nous avons adapté des méthodes de traitement, plus simples que celles existantes. Le système développé est implanté sur circuit FPGA et piloté avec une fréquence maximale de 550 MHz. Les résultats obtenus ont été comparés à des travaux récents, et ont montré une amélioration significative en termes de temps de calcul, notamment dans le cas

d'augmentation de la fréquence du système de transmission à un débit très élevé.

2. Objectifs de la thèse

Le but préalable de cette étude est l'optimisation de l'implémentation de trois générateurs de sinusoïde dont l'un est basé sur le processeur CORDIC itératif, l'autre sur un processeur CORDIC pipeline avec accumulateur de phase et le dernier sur un processeur CORDIC pipeline sans accumulateur de phase. Chacun de ces générateurs de sinusoïdes est intégré successivement dans l'architecture du modulateur numérique QAM pour évaluer ses performances [6]. Cette implémentation se fait par le langage VHDL (Very High Speed Integrated Circuit Hardware Description Language) qui est un logiciel de description destiné à représenter le comportement architectural d'un système électronique numérique [7], de simulation, de programmation des circuits sur des composants programmables ISE12.2[8] sur circuit FPGA du type Xilinx Virtex5–XC5VLX110T [9].

3. Contributions de la thèse

Les travaux réalisés dans cette thèse s'inscrivent dans le domaine du développement, de la conception et de l'implémentation des différentes techniques pour la modulation numérique sur des cibles reconfigurables de type FPGA.

Afin d'atteindre les objectifs de cette thèse, le travail réalisé a entrepris les tâches suivantes :

- Une analyse des différentes architectures.
- Proposition d'une plateforme pour la génération de modulateur numérique QAM, en langage de description matériel VHDL.
- Proposition d'une architecture complète de modulateur QAM.
- Une simulation pour tester la validité des algorithmes mis en œuvre.
- Un rapport sur la consommation des ressources des différentes architectures.

4. Plan du manuscrit

Ce manuscrit de thèse s'articule autour de quatre chapitres :

Afin de situer la couche concernée par la production des signaux dans le domaine des télécommunications, des modélisations sont rappelés pour permettre la séparation des différents niveaux de fonctionnalité d'un système de traitement numérique de l'information, plus particulièrement dans le cadre de transmissions réseau.

Le premier chapitre, à cet égard, traite les généralités sur la transmission de l'information

Au deuxième chapitre nous présentons donc les différents types de modulations fréquemment utilisées dans ce domaine.

Le troisième chapitre, après avoir étudié les modulateurs existants dans ce contexte pour situer notre contribution, traite l'architecture de modulateurs QAM à base du processeur CORDIC modifié. Pour cela, une étude comparative entre les architectures construites sur la base de différents algorithmes CORDIC pour la génération de sinusoides est entreprise afin d'étudier les performances du modulateur qui sera abordé au chapitre suivant.

Le quatrième chapitre est consacré à la simulation et implémentation du modulateur QAM adapté au haut débit. La description matérielle est obtenue par le logiciel ISE et l'implantation se fera sur circuits programmables. Pour cela, ce chapitre comporte deux parties A et B dont la première est consacrée à la présentation du langage VHDL [64], le circuit FPGA [68] et la deuxième traite les résultats de simulation et d'implantation.

Enfin nous clôturons ce document avec une conclusion générale ainsi que des perspectives d'application et d'exploitation de la technologie programmable FPGA.

CHAPITRE I
GENERALITES SUR LA
TRANSMISSION
NUMERIQUE DE
L'INFORMATION

Généralités sur la transmission numérique de l'information

1. Introduction

Généralement la transmission numérique transfère les données entre une source et un destinataire en utilisant un support physique (câble, la fibre optique et ligne téléphonique,...etc.). Les signaux transportés peuvent être soit directement d'origine numérique (les réseaux) ou bien analogique avec convertisseur analogique-numérique, (parole, image,... etc) [10].

Dans la figure 1.1 nous présentons le schéma synoptique d'un système de transmission numérique et ses éléments définis comme ce qui suit :

- La source émettrice d'une suite d'éléments binaires d'un message numérique.
- Le codeur qui permet de convertir les séries de M-bits composant un symbole, pour les rendre compatibles avec le type de modulation choisi.
- Le modulateur qui a pour rôle d'adapter le signal à émettre au canal de transmission.
- Le démodulateur et le décodeur dont les fonctions respectives de démodulation et de décodage sont les inverses aux fonctions de modulation et de codage situées du côté émetteur.

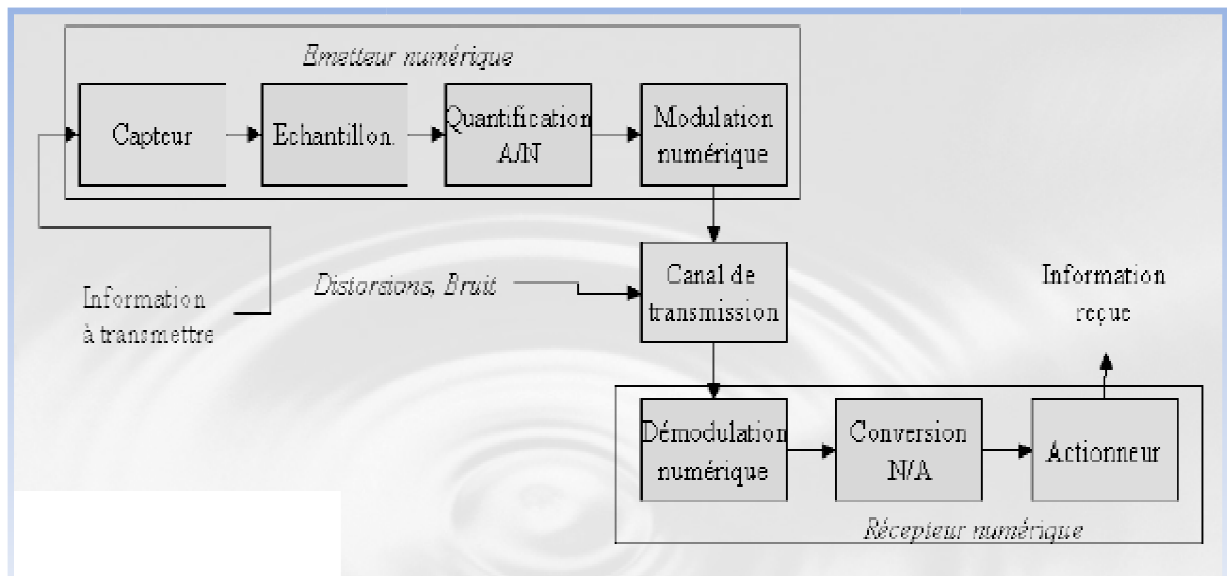


Figure 1.1. Schéma synoptique d'un système de transmission numérique.

2. Modèle de Transmission de l'information

Il y'a deux modèles différents :

- Filaire (conducteur électrique, guide optique).
- Sans fils (le support est l'air ou le vide).

2.1. Système de transmission de l'information sans fil (réseaux sans fil)

Aujourd'hui, la majorité des ordinateurs et des appareils mobiles (tels que les téléphones portables) disposent de moyens de connexion à un ou plusieurs types de réseaux sans fil comme le Wifi, le Bluetooth ou l'infrarouge. Ainsi, il est très facile de créer en quelques minutes un réseau sans fil permettant à tous ces appareils de communiquer.

Un réseau sans fil est un ensemble d'appareils connectés entre eux et qui peuvent s'envoyer et recevoir des données sans qu'aucune connexion filaire physique reliant ces différents composants entre eux ne soit nécessaire [11].

2.1.1. Techniques de transmission dans les réseaux sans fil

a. La transmission sans fil par ondes

Un réseau sans fil standard fonctionne pratiquement comme un réseau câblé : une carte réseau sans fil dotée d'un émetteur- récepteur (périphérique transmettant et recevant des signaux analogiques et numériques) est installée dans chaque ordinateur. L'utilisateur communique avec le réseau comme s'il s'agissait d'un ordinateur câblé [12].

Il existe principalement deux méthodes pour la transmission dans les réseaux sans fil (ondes), transmission infrarouge et la transmission radio:

a.1. Transmission par les ondes infrarouges

La transmission par les ondes infrarouges nécessite que les appareils soient face à face et aucun obstacle ne sépare l'émetteur du récepteur (car la transmission est directionnelle). Cette technique est utilisée pour créer des petits réseaux de quelques dizaines de mètres (télécommande, les jouets, voitures...).

a.2. Transmission par les ondes radio

La transmission par les ondes radio est utilisée pour la création des réseaux sans fil séparés par plusieurs km de distance. Les ondes radio ont l'avantage de ne pas être arrêtées par les obstacles et sont émises d'une manière omnidirectionnelle. Le problème de cette technique se manifeste par les perturbations extérieures qui peuvent affecter la communication à cause de l'utilisation de la même fréquence par exemple.

2.1.2. Avantages des réseaux sans fil

- Liberté de mouvement des usagers et des terminaux.

- Moins de risques de rupture de liens que dans le filaire (coupure du câble, abîmer les connecteurs et prises, ...etc).
- Délais de mise en service du réseau plus faibles.
- Des économies à long terme (retour sur investissements).

2.1.3. Domaines d'application

1. Tous les métiers dont la mobilité est nécessaire pour augmenter la productivité
 - Hôpitaux (gestion des fichiers patients, ...).
 - Restaurants (communications rapides entre serveurs et cuisiniers, ..).
 - Maison (éviter de câbler, coûts réduits, ...).
2. Environnement d'installations difficiles des médias filaires.

2.1.4. Les catégories de réseaux sans fil

On distingue quatre catégories selon le périmètre géographique offrant une connexion (appelé zone de couverture).

a. Les réseaux personnels sans fil (WPAN: Wireless Personal Area Network)

Les réseaux personnels sans fil ou Wireless Personal Area Network (WPAN), sont des réseaux sans fil à très faible portée, de l'ordre d'une dizaine de mètres. Ils sont le plus souvent utilisés à faire communiquer entre eux des matériels présents sur une personne (par exemple une oreillette et un téléphone portable). Ils sont également utilisés pour relier des équipements informatiques entre eux sans liaison filaire : par exemple pour relier une imprimante ou un PDA (Personal Digital Assistant) à un ordinateur de bureau ou faire communiquer deux machines très peu distantes [13].

b. Réseaux locaux sans fil (WLAN: Wireless Local Area Network)

Depuis le développement des normes qui offrent un haut débit, les réseaux locaux sans fil ou Wireless Local Area Network (WLAN) sont généralement utilisés à l'intérieur d'une entreprise [14], d'une université, mais également chez les particuliers:

- Portée de 50 à quelques centaines de mètres.
- ayant une station relais.
- IEEE 802.11 (standards d'USA) (entre 4 et 54 Mbps, bande 2.4GHz à 2,4835 GHz).

- Hiper Lan (High Performance LAN, European Telecommunications Standards Institute), HiperLan1: débit 20 Mbps, portée: 100 mètres, bande de fréquences = 5.3Ghz et HiperLan2: débit 54Mbps.

c. Les réseaux métropolitains sans fil (WMAN : Wireless Metropolitan Area Network)

Les réseaux métropolitains sans fil ou Wireless Metropolitan Area Network (WMAN) sont aussi connus sous l'appellation de boucle locale radio (BLR). Les réseaux basés sur la technologie IEEE 802.16 ont une portée de l'ordre de quelques dizaines de kilomètres (50km de portée théorique annoncée) et un taux de transmission radio théorique pouvant atteindre 74 Mbit/s pour IEEE 802.16, plus connu sous le nom commercial de WiMAX [13].

d. Les réseaux étendus sans fil (WWAN : Wireless Wide Area Network)

Les réseaux sans fil (WWAN pour Wireless Wide Area Network) Cette catégorie possède assez peu de technologies à l'heure actuelle. Les seules technologies de WWAN disponibles sont des technologies utilisant les satellites géostationnaires ou en orbite basse pour relayer l'information entre plusieurs points du globe [13].

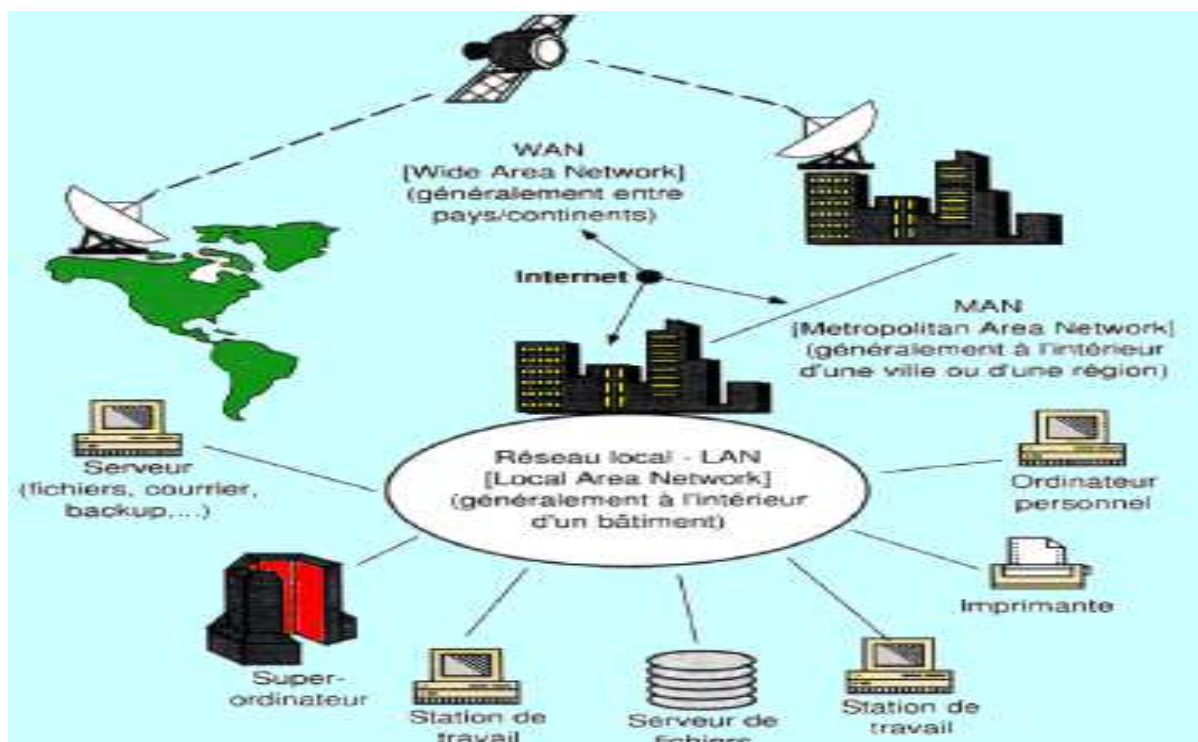


Figure 1.2. Les réseaux sans fil [12].

2.2. Les modèles de référence

Des efforts de Modélisation ont été effectués pour permettre de séparer les différents niveaux de fonctionnalité d'un système de traitement numérique de l'information, plus particulièrement dans le cadre de transmission réseau. Il existe deux principaux modèles :

Le modèle OSI est souvent employé pour décrire cette segmentation, bien que peu de protocoles réels soient effectivement basés sur lui [15].

Le modèle TCP/IP, antérieur au précédent, dispose d'une architecture modulaire ressemblant à celle d'OSI. Il est à la base du réseau mondial internet [16].

2.2.1. Le modèle OSI

- OSI (Open System Interconnexion), est un standard ouvert générique débute en 1978.
- Les vendeurs peuvent utiliser leur propre implémentation et protocoles.
- Le modèle OSI comporte 7 couches, et est présenté sur la figure (1.4).

Les principes qui ont conduit à ces 7 couches sont les suivants :

- ✓ une couche doit être créée lorsqu'un nouveau niveau d'abstraction est nécessaire.
- ✓ chaque couche a des fonctions bien définies.
- ✓ les fonctions de chaque couche doivent être choisies dans l'objectif de la normalisation internationale des protocoles.
- ✓ les frontières entre couches doivent être choisies de manière à minimiser le flux d'information aux interfaces.
- ✓ le nombre de couches doit être tel qu'il n'y ait pas cohabitation de fonctions très différentes au sein d'une même couche et que l'architecture ne soit pas trop difficile à maîtriser.
- ✓ Les couches basses (1, 2, 3 et 4) sont nécessaires à l'acheminement des informations entre les extrémités concernées et dépendent du support physique.
- ✓ Les couches hautes (5, 6 et 7) sont responsables du traitement de l'information relative à la gestion des échanges entre systèmes informatiques. Par ailleurs, les couches 1 à 3 interviennent entre machines voisines, et non entre les machines d'extrémité qui peuvent être séparées par plusieurs routeurs. Les couches 4 à 7 sont au contraire des couches qui n'interviennent qu'entre hôtes distants.

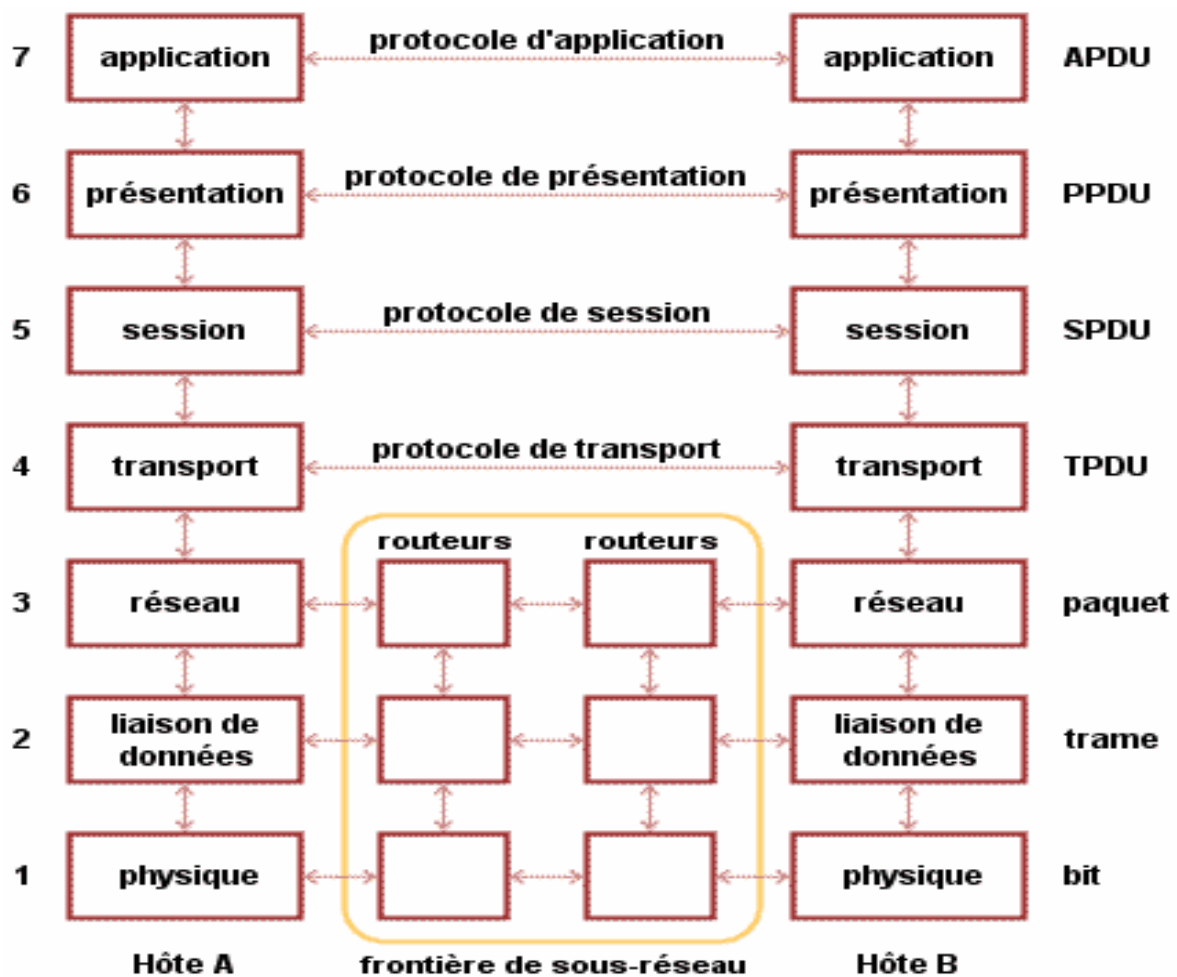


Figure 1.3. Modèle de référence OSI [15].

2.2.1.1. Rôle et caractéristiques de chacune des 7 couches du modèle OSI

- Service : est la tâche exécutée par une couche.
- Interface : comment accéder au service.
- Protocole : comment le service est implémenté.
- Ensemble de règles et de formats pour les communications.

a. La couche physique

Elle est chargée de la transmission effective des signaux entre les interlocuteurs. Son service est typiquement limité à l'émission et la réception d'un bit ou d'un train de bit continu (notamment pour les supports synchrones).

- Transférer des bits entre deux systèmes connectés par un câble physique.
- Schéma de codage pour représenter un bit : niveaux de tension, durée d'un bit, etc.

b. La couche liaison de données

Gère les communications entre deux machines adjacentes, directement reliées entre elles par un support physique.

- Définit des règles pour l'émission et la réception de données.
- Assure un transport fiable des données par le média.
- Connectivité et sélection du chemin entre les systèmes hôtes.
- Découpage en frames.
- Envoyer des frames de données entre pairs liés au même medium physique.
- Gérer l'accès au support partagé.
- Détection et correction des erreurs dues au support physique.

c. La couche réseau

Gère les communications de proche en proche, généralement entre machines : routage et adressage des paquets.

- Adresse réseau et détermination du chemin.
- Assure un transport fiable des données par le média.
- Connectivité et sélection du chemin entre les systèmes hôtes.
- Adressage logique géré par l'administrateur du réseau.
- Acheminement au mieux.
- Unité de couche 3 : routeur.
- Unité d'échange : paquet.
- Envoyer un paquet à une destination précise.
- Découpage / réassemblage.
- Contrôle de flux.
- Ordonnancement de paquets, congestion, routage.
- Détection et correction d'erreurs supplémentaires.

d. La couche transport

Gère les communications de bout en bout entre processus (programmes en cours d'exécution).

- Concerne les problèmes de transport entre les hôtes.
- Détection des pannes et reprise du contrôle du flux d'information.
- L'émetteur segmente les messages de données en paquets et le récepteur reconstitue les messages en remplaçant les paquets dans le bon ordre.
- Multiplexage de plusieurs connexions transport en une seule connexion réseau.
- Partage d'une connexion transport en plusieurs connexions réseau.

- Exemples : TCP et UDP.

e. La couche session

Gère la synchronisation des échanges et les transactions, permet l'ouverture et la fermeture de session (figure.1.5)

- Couche orientée traitement.
- Communication entre les hôtes.
- Gestion des accès (jetons).
- Synchronisation
- Interface : dépend des services.

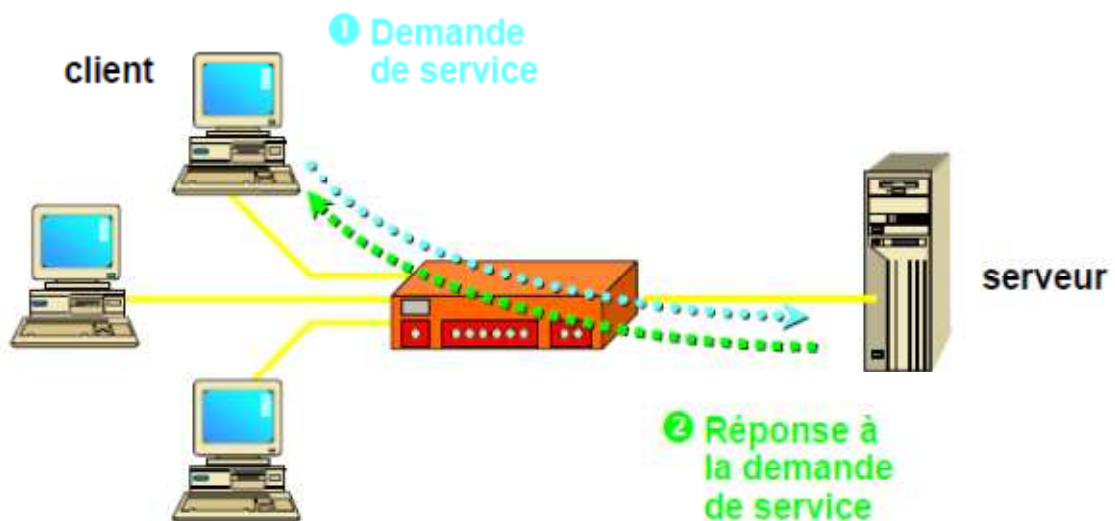


Figure 1.4. Schéma synoptique de la couche session. [17]

f. La couche présentation

Elle est chargée du codage des données applicatives, précisément de la conversion entre données manipulées au niveau applicatif et chaînes d'octets effectivement transmises. (Figure 1.6)

- Permet de formater les données dans un format compréhensible par les deux systèmes émetteur et récepteur.
- Structure les données.
- Négocie la syntaxe du transfert des données pour la couche application.
- Assure éventuellement la compression et le cryptage/décryptage des données.
- Conversion de données entre diverses représentations.

- Définit les formats de données et les règles de conversion.

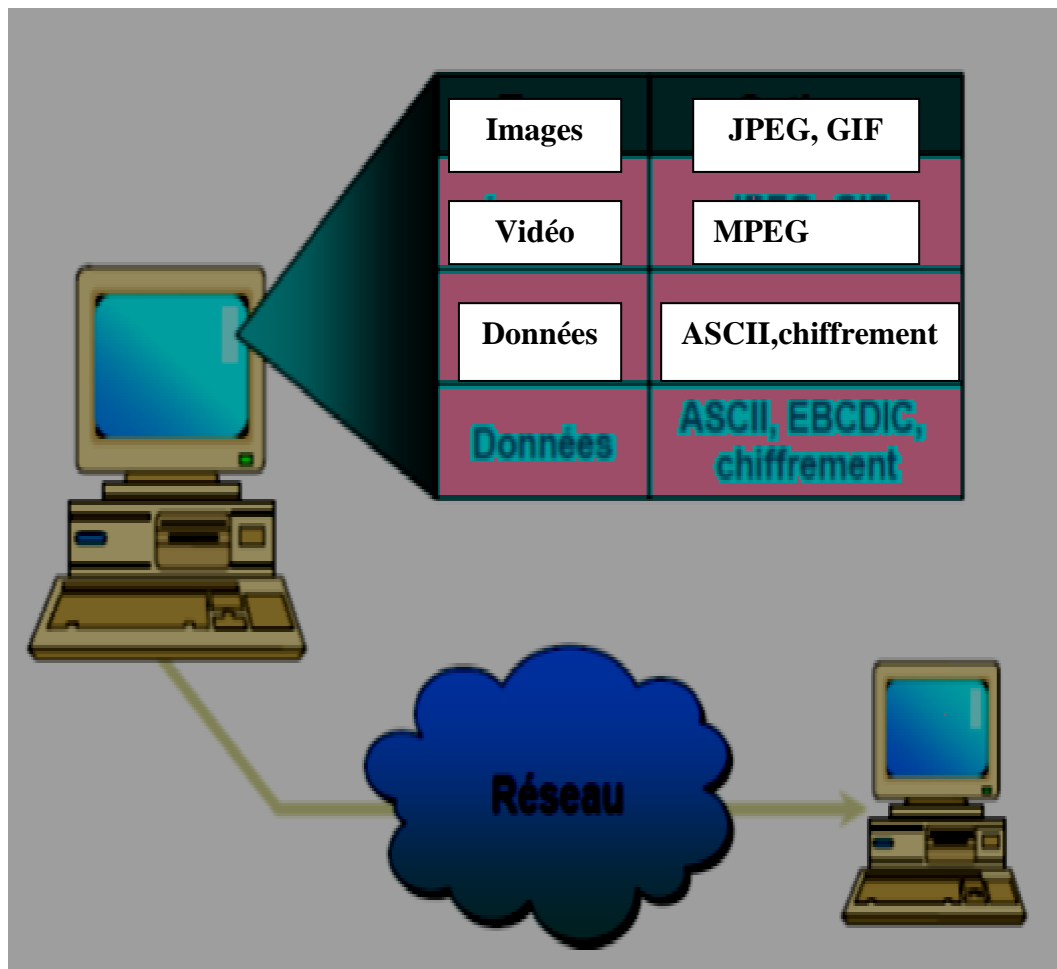


Figure 1.5. Schéma synoptique de la couche présentation [17].

g. La couche application

Elle est le point d'accès aux services réseaux. Elle n'a pas de service propre spécifique et entre dans la portée de la norme (figure.1.7).

- Fournit des services au module de communication d'une application.
- Crée une interface directe avec le reste du modèle OSI par le biais d'applications réseau (navigateur Web, messagerie électronique, transfert de fichiers (FTP), terminal virtuel (Telnet), client-serveur) ou crée une interface indirecte, par le biais d'applications autonomes (comme les traitements de texte, les logiciels de présentation ou les tableurs), avec des logiciels de redirection réseau.

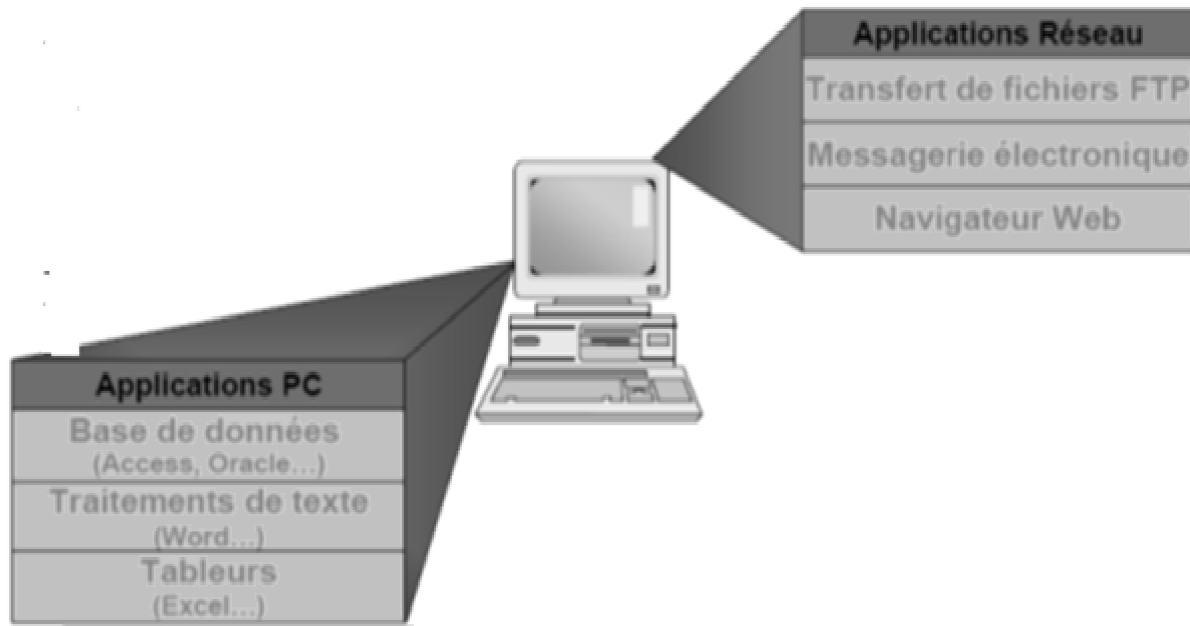


Figure 1.6. Schéma synoptique de la couche application [17].

2.2.2. Le modèle TCP/IP

- TCP/IP est une pile de protocoles travaillant sur un modèle en couches particulier DOD (fig.1.8).
- Le modèle DOD recouvre les différentes couches du modèle OSI.
- Au contraire d'OSI, TCP/IP est né après les protocoles qu'il modélise, et il ne fait que peu de distinctions entre les concepts de services, d'interfaces ou encore de protocoles. Par conséquent, la segmentation en couches indépendantes d'OSI n'est pas présente de façon aussi stricte dans TCP/IP. Le modèle TCP/IP permet simplement de positionner les protocoles existants et futurs dans un cadre théorique [18].

a. La couche application

Ce modèle contient entre autres tous les protocoles de haut niveau, comme Telnet ou FTP5.

b. La couche transport

Permet aux applications d'échanger des données indépendamment du réseau utilisé, grâce aux protocoles TCP et UDP.

- Même rôle que la couche Transport du modèle OSI : transport correct de messages de manière fiable et de bout en bout entre l'émetteur et le récepteur.
- 2 implémentations : le protocole TCP (Transmission Control Protocol) et le protocole UDP (User Datagram Protocol).

c. la couche internet (réseau)

C'est la pierre angulaire de cette architecture, et permet aux hôtes d'envoyer des paquets élémentaires indépendants les uns des autres, sans se préoccuper des détails concernant leur acheminement vers l'hôte destination.

- Réalise l'interconnexion des réseaux hétérogènes distants.
- permettre l'injection de paquets dans n'importe quel réseau et acheminement de ces paquets indépendamment les uns des autres jusqu'à destination.
- Gère le routage des paquets au travers des réseaux empruntés.
- Protocole IP (Internet Protocol).

d. la couche d'accès au réseau

- Elle est mal définie par le protocole. Elle regroupe tous les éléments nécessaires pour accéder à un réseau physique, quel qu'il soit.
- Recouvre les couches Physique et liaison du modèle OSI, exemple : Ethernet
- Elle contient en particulier les spécifications concernant la transmission de données sur le réseau physique, tout comme la première couche du modèle OSI.

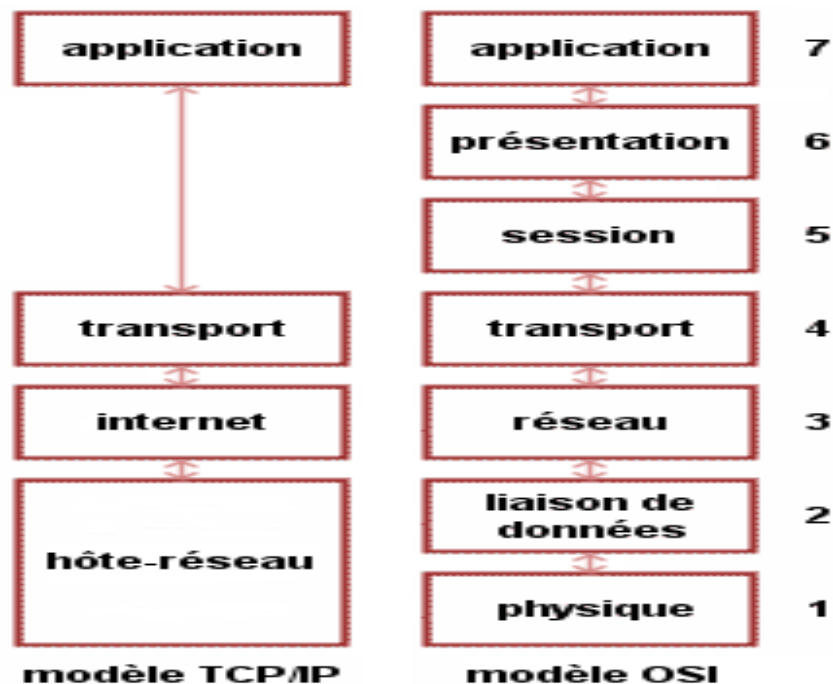


Figure 1.7. Modèle de référence TCP/IP en parallèle avec le modèle OSI [15].

3. Les avantages et les inconvénients des systèmes numériques

Il y a plusieurs avantages et inconvénients des systèmes de communications numériques :

3.1. Les avantages

- Faibles tolérances des composants.
- Sensibilité réduite, Précision contrôlée.
- Reproductibilité, pas de réglage.
- Souplesse, nombre d'opérations illimité.
- Systèmes non réalisables en analogique
- Les impulsions émises peuvent facilement être régénérées.
- Différents types de signaux (voix, image, vidéo, etc.) sont numérisés et traités de la même façon.
- Plus robuste aux distorsions et aux interférences.
- Souvent plus économiques que les systèmes analogiques.
- Plus flexibles (exemple : logiciel radio).
- Meilleure immunité au bruit.
- Optimisation de la bande passante.
- Facilité de traitement de l'information.
- Modification des paramètres f , Φ , A .
- Distance plus grande.
- Possibilité de codage des erreurs de transmission

3.2. Les inconvénients

- Source d'énergie nécessaire.
- Limitations en haute fréquence.
- Convertisseur analogique numérique / Convertisseur numérique analogique.
- Bande passante nécessaire importante.
- Les algorithmes ont besoin d'une bonne puissance de traitement.
- Le système doit être synchronisé.
- Les dégradations d'un système numérique ne sont pas progressives.
- Séparation d'une application en sous-ensembles.
- Utilisation de composants à grande tolérance.
- Circuits de codage et décodages requis

4. Conclusion

Nous avons présenté dans ce chapitre les généralités sur les systèmes de transmission de l'information numérique avec leurs avantages et inconvénients de ces systèmes sans fil. La

position et le rôle du modulateur dans une chaîne de transmission seront explicités dans le chapitre suivant, ainsi que les difficultés d'implantation posées aux architectures numériques pour la transmission d'un signal numérique.

***CHAPITRE II
QUELQUES
DIFFERENTS TYPES
DE MODULATIONS
FREQUEMMENT
UTILISEES***

1. Introduction

Depuis quelques années, la diversification et la croissance des services télématiques (vidéotex, télécopie, microordinateurs communicants ...) a contribué au développement des services de transmissions de données [19]. L'idée d'assurer, au sein d'un même réseau, une large gamme d'applications a conduit ces dernières années au développement d'un réseau à caractère universel. Cependant, soit par souci économique (récepteurs grand public) soit par impossibilité physique (récepteurs mobiles), certaines applications ne peuvent prétendre à un raccordement à ce type de réseau [20]. Le mode de transmission par voie hertzienne permet d'accéder à n'importe quel type de récepteur. Il permet une vaste gamme d'applications en particulier celles prévues pour être exploitées dans les récepteurs d'automobiles. De nouvelles applications telles que le téléaffichage, la radio texte ou la diffusion d'images fixes présentent un intérêt croissant. Elles imposent des débits plus importants et conduisent à envisager de nouveaux systèmes de diffusion sur support FM. Le choix d'une technique de modulation repose sur les différentes contraintes physiques telles que le débit, la bande allouée liée à la réglementation et les performances en présence des différentes distorsions subies par le signal [21]. Cet aspect est présenté dans ce chapitre où quatre types de modulation sont sélectionnés. Les principes de chaque modulation sont présentés dans ce chapitre.

2. Les avantages et les inconvénients de la modulation numérique

Il y a plusieurs avantages et inconvénients de la modulation numérique :

- Tout d'abord, le rapport signal sur bruit est meilleur avec un système numérique car, même si un signal numérique est bruité, distordu ou parasité, il est facile de le reconstruire en comparant ce signal déformé à un seuil. On peut atteindre ainsi des taux d'erreurs (nombre de bits erronés divisé par le nombre de bits total) de 10^{-13} . Ensuite, les densités spectrales des signaux modulés numériquement ont des largeurs moindres qu'en analogique, ce qui permet d'augmenter le nombre de canaux utilisables par Hz pour les transmissions d'informations.
- La numérisation des signaux offre la possibilité de réaliser à la fois du multiplexage temporel, fréquentiel et par conséquent de transporter de plus en plus d'informations tout en conservant le même encombrement spectral [22].
- Adaptation du signal modulé aux caractéristiques fréquentielles du canal de transmission.
- Rayonnement possible dans une antenne.
- Transmission possible à longue distance (exemple : satellites).
- Systèmes plus complexes risquent d'augmentation de la dégradation du signal due aux équipements.

- Bande de fréquences à l'émission plus importante que celle du message.

3. Critères de choix d'une modulation numérique

Avant d'argumenter ce choix, nous commençons par définir les principaux critères caractérisant une modulation numérique à savoir [23] :

- Le taux d'erreur sur les bits.
- L'efficacité spectrale
- Rapport E_b/N_0
- Le débit binaire.

3.1. Taux d'erreur sur les bits

La qualité d'une liaison est liée au taux d'erreur par bit (TEB) obtenu en divisant le nombre de bits faux reçus par le nombre de bits transmis.

$$\text{TEB} = \frac{\text{Nombres de bits faux}}{\text{Nombres de bits transmis}} \quad (2.1)$$

3.2. L'efficacité spectrale

L'efficacité spectrale (en Bit/Seconde/Hertz) d'un signal numérique est le nombre de bits par seconde de données qui peuvent être supportées pour chaque hertz de la bande de fréquence utilisée (B en Hertz) :

$$\eta = \frac{\text{Débit}}{\text{Bande occupée}} = \frac{D(\text{Bit/s})}{B(\text{Hertz})} \quad (2.2)$$

On peut aussi noter que pour des applications où la bande passante est limitée par des contraintes physiques, il faut choisir une technique qui donne la plus haute efficacité spectrale, laquelle doit permettre d'obtenir de faibles taux d'erreur sur le bit en sortie du système.

3.3. Rapport E_b/N_0

E_b/N_0 est le rapport entre l'énergie émise par bit E_b et la densité spectrale de puissance de bruit N_0 .

Ce rapport est lié au rapport signal sur bruit par l'expression suivante [24] :

$$\text{SNR} = \frac{S}{N} = \eta \times \log_2 (M) \times \frac{E_b}{N_0} \quad (2.3)$$

Où : M est la taille de l'alphabet et η l'efficacité spectrale.

3.4. Débit binaire

Le débit binaire (D) est défini comme le nombre d'informations élémentaires (ou Bits) que le canal transporte par unité de temps. Son unité est le bit.s⁻¹.

Il s'exprime en fonction de la durée du bit d'information T_b (exprimée en seconde) :

$$D = \frac{1}{T_b} \quad (2.4)$$

4. Différents types de modulations numériques

4.1. Objectif de la modulation

La modulation a pour objectif d'adapter le signal à émettre au canal de transmission. Cette opération consiste à modifier un ou plusieurs paramètres d'une onde porteuse centrée sur la bande de fréquence du canal (figure. 2.1).

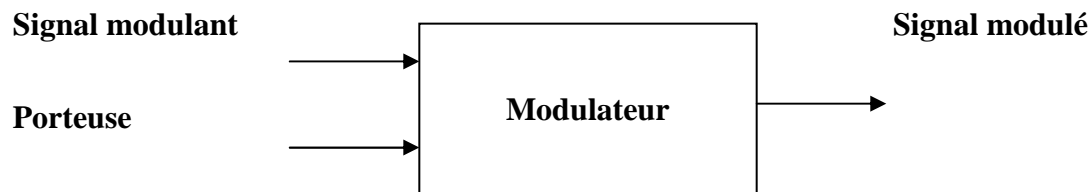


Figure 2.1. Schéma bloc d'un modulateur.

La porteuse est de forme sinusoïdale.

$$S(t) = A \cos (\omega.t + \Phi) \quad (2.5)$$

Le signal modulant sous forme numérique.

Les paramètres modifiables en fonction du signal modulant numérique selon le type de modulation employé sont :

- L'amplitude : A
- La fréquence : $f = \frac{\omega}{2\pi}$
- La phase: Φ

Les types de modulation les plus fréquemment rencontrés sont les suivants [25] :

4.2. Modulation par déplacement d'amplitude MDA

Les Modulations par Déplacement d'Amplitude (MDA) sont aussi souvent appelées par leur abréviation anglaise : ASK pour "Amplitude Shift Keying".

L'amplitude de la porteuse sinusoïdale varie avec les 0 et les 1 du signal numérique, la phase et la fréquence de la porteuse restent fixes (figure 2.2).

$$s_i(t) = \sqrt{\frac{2E_i}{T}} \cos(2\pi f_c t + \varphi), \begin{cases} 0 \leq t \leq T \\ i = 1, \dots, M \end{cases} \quad (2.6)$$

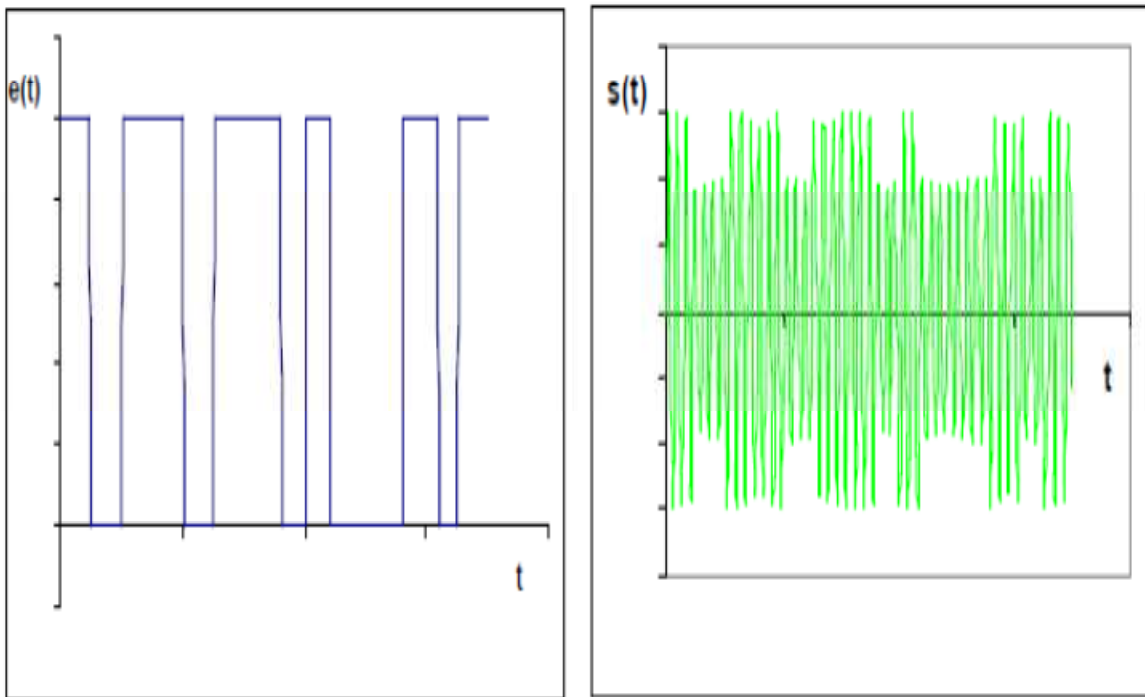


Figure 2.2.Exemple de signal modulant $e(t)$ et modulé ASK $s(t)$.

4.2.1. Les avantages et les inconvénients de la modulation numérique ASK

- Simplicité de réalisation.
- Sensibilité aux multi-trajets
- Sensibilité à la non-linéarité de la fonction de transfert des amplificateurs de puissance (enveloppe non-constante).
- Faible efficacité énergétique : beaucoup de puissance requise pour obtenir une probabilité d'erreur faible.

4.2.2. Application ASK : est souvent utilisée pour transmettre des données numériques sur fibre optique.

4.3. Modulation par déplacement de fréquence MDF

Elles sont aussi souvent appelées par leur abréviation anglaise : FSK pour Frequency Shift Keying.

La fréquence de la porteuse sinusoïdale varie en fonction des 0 et des 1 du signal numérique, l'amplitude et la phase de la porteuse restent fixes. (figure.2.3)

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos(2\pi f_i t + \varphi), \begin{cases} 0 \leq t \leq T \\ i = 1, \dots, M \end{cases} \quad (2.7)$$

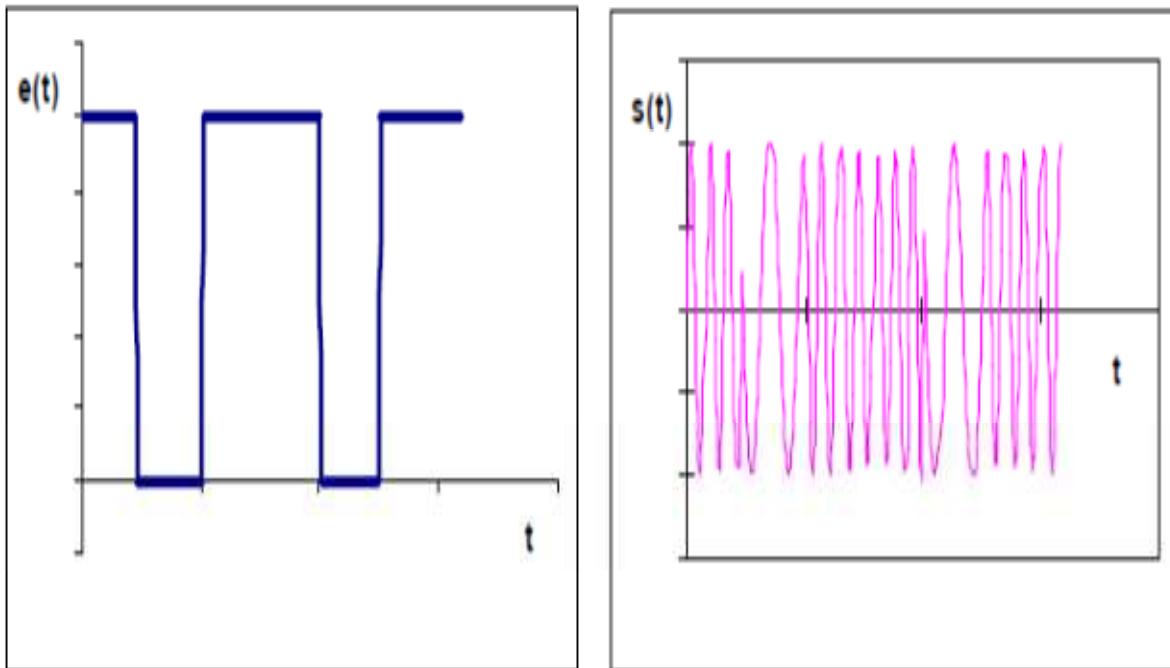


Figure 2.3.Exemple de signal modulant $e(t)$ et modulé FSK $s(t)$.

4.3.1. Les avantages et les inconvénients de la modulation numérique FSK

- Enveloppe du signal constante (énergie constante).
- Permet l'utilisation d'amplificateur non-linéaire (efficacité énergétique augmentée).
- Moins sensible aux atténuations causées par le canal de transmission.
- Efficacité spectrale diminuée (nécessite plus de bande passante que le ASK ou PSK).
- Probabilité d'erreur supérieure au PSK.
- Peu sensible aux bruits.

4.3.2. Utilisation du FSK

La modulation FSK est utilisée pour la transmission de données. Elle n'est plus beaucoup appliquée de nos jours. Néanmoins, on la retrouve encore dans certaines antennes de transmissions, dans les applications de télémétrie ne nécessitant pas de haut débit, ou encore dans des systèmes miniaturisés tels que des portes clés pour voiture. Son plus grand succès fut son application dans les premiers modems.

4.4. Modulation par déplacement de phase MDP (Phase Shift Keying PSK)

La phase de la porteuse sinusoïdale varie avec les 1 et les 0 du signal numérique (fig. 2.4).

- L'amplitude et la fréquence de la porteuse restent fixes,
- Principe de modulation le plus employé car sa technique est simple,
- Les états des phases sont répartis régulièrement sur l'intervalle $[0 ; 2\pi]$
- Possibilité de représenter plusieurs bits simultanément : 2, 4 ou 8 phases en pratique.

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left(2\pi f_c t - \frac{2\pi i}{M}\right), \begin{cases} 0 \leq t \leq T \\ i = 1, \dots, M \end{cases} \quad (2.8)$$

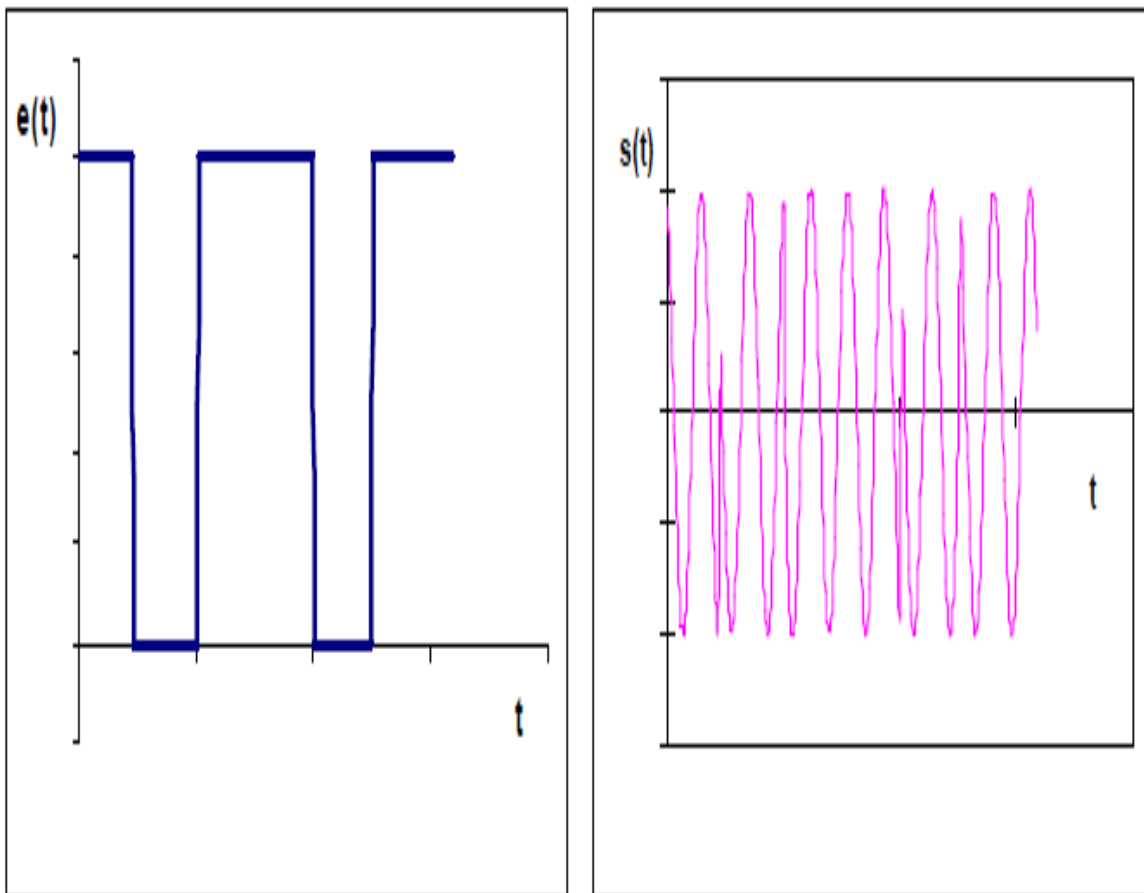


Figure 2.4.Exemple de signal modulant $e(t)$ et modulé PSK $s(t)$.

4.4.1 .Les avantages et les inconvénients de la modulation numérique PSK

- PSK est moins susceptible d'erreurs qu'ASK, alors qu'elle nécessite une occupation de la même bande passante que la modulation ASK.
- Une utilisation plus efficace de la bande passante (débit plus élevé) est possible par rapport à la modulation FSK.
- Signal plus complexe de processus de détection / récupération, que dans ASK et FSK.

4.4.2..Utilisation de la modulation numérique PSK

- La modulation PSK est utilisée souvent pour la transmission vers les satellites car dans ce cas les vitesses de transmission sont plus élevées qu'en modulation FSK pour la même bande passante et avec des codes à plusieurs bits, on peut augmenter le débit sans changer la fréquence de modulation.

4.5. Modulation d'amplitude de deux porteuses en quadrature MAQ (Quadrature Amplitude Modulation QAM).

4.5.1. Définition

Ce dernier type de modulation peut également être considéré comme une modulation mixte d'amplitude et de phase (figure 2.5) pour augmenter le nombre de symboles de la constellation.

- Exemple QAM-32
- Symboles: 5 bits
- 32 points dans la constellation

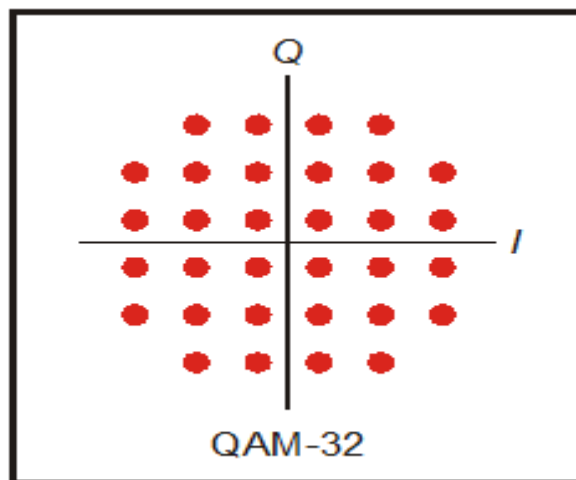


Figure 2.5. Modulation d'amplitude de deux porteuses en quadrature MAQ [25].

4.5.2 Principe de la modulation QAM

En effet , l'expression du signal modulé est telle que (figure 2.6.) :

$$S(t) = A(n) \cos (wt + \Phi) + B(n) \sin(wt + \Phi) \quad (2.9)$$

$$s(t) = \sqrt{A^2(n) + B^2(n)} \cos (wt + \Phi - \arctan (B(n)/A(n))) \quad (2.10)$$

$$\text{L'amplitude est égale } \sqrt{A^2(n) + B^2(n)} = M(n) \quad (2.11)$$

$$\text{et la phase est : } \varphi_m(n) = \arctan\left(\frac{B(n)}{A(n)}\right) \quad (2.12)$$

Le signal réel passe-bande transmis dans le canal est le suivant :

$$s(t) = \text{Re}\{ M(n) e^{j(\omega t + \Phi - \varphi_m(n))} \} = M(n) \cos(\omega t + \Phi - \arctan(B(n)/A(n))) \quad (2.13)$$

Où : $e^{j(\omega t + \Phi - \varphi_m(n))}$ représente le signal de la porteuse en notation complexe, caractérisé par sa pulsation ω (rad/s) et sa phase $\Phi - \varphi_m(n)$ en (rad).

De plus, on a :

$$s(t) = M(n) \cdot \cos(\omega t + \Phi - \varphi_m(n)) \quad (2.14)$$

L'équation (2.10) montre que la modulation QAM est équivalente à la modulation simultanée de l'amplitude et de la phase d'une porteuse (fig. 2.7).

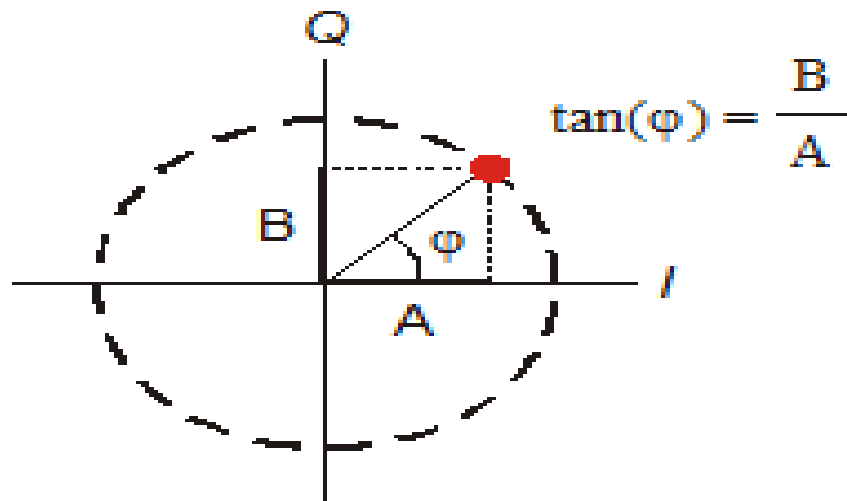


Figure 2.6. Principe de modulateur QAM.

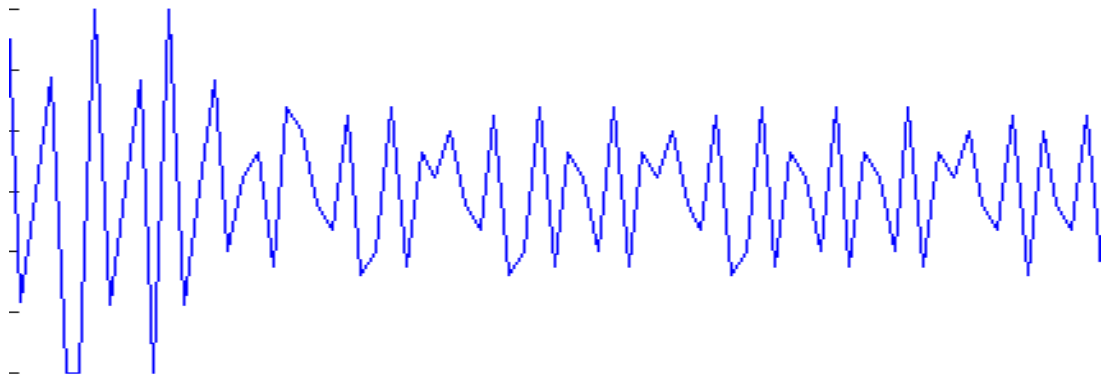


Figure 2.7.Exemple d'un signal modulé QAM.

4.5.3. Domaine d'application de la modulation QAM

La modulation QAM est utilisée dans plusieurs domaines. Nous pouvons citer par exemple l'audiovisuel pour le transfert des composantes d'informations en couleur, comme la télévision (PAL et NTSC) par câble et de modem câble [26]. Au royaume- Uni, 16 QAM et 64 QAM sont actuellement utilisés pour la télévision numérique terrestre à l'aide de DVB-Digital Vidéo Broad casting. Aux États-Unis, 64 QAM et 256 QAM sont les schémas de modulation prescrits pour le câble numérique normalisés par le SCTE dans la norme ANSI / SCTE 07 2000 [27]. Le modem d'internet qui sert à générer et faciliter la transmission entre deux personnes (appel, sms...etc) [28] (figure 2.8).

- La modulation QAM est présente dans de nombreuses applications de communication radio et de transmission de données. Cependant, certaines variantes spécifiques de QAM sont utilisées dans des applications et normes spécifiques et restreintes [29].
- Pour les applications de diffusion domestique, par exemple, 64 QAM et 256 QAM, de plus, des variantes de QAM sont également utilisées pour de nombreuses applications de technologie sans fil et cellulaire [30].

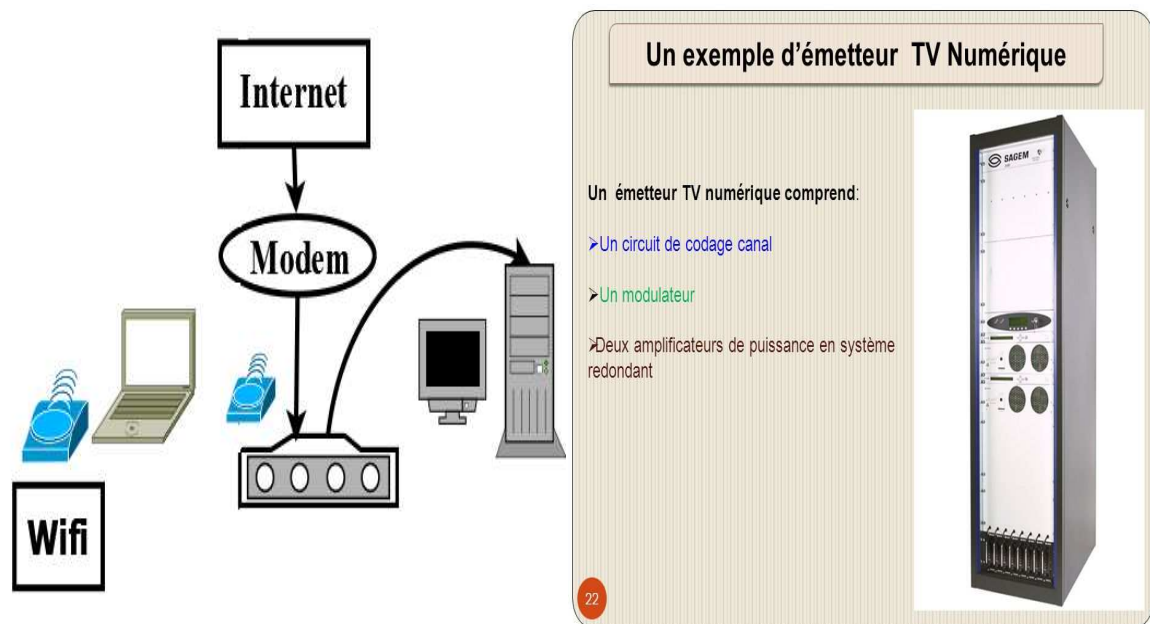


Figure 2.8.Représentation schématique de l'utilisation de la modulation QAM [26].

4.5.4. Les avantages et les inconvénients de QAM

- Ce type de modulation transmet plus de bits d'information par symbole.
- Il fournit un bon niveau pour les taux de bits élevés en utilisant des formes supérieures de QAM.
- Cette modulation nécessite une démodulation cohérente avec la phase et la fréquence exactement.
- Elle est plus sensible au bruit car les états sont plus proches les uns des autres.
- La deuxième limitation est également associée à la composante d'amplitude du signal :
Lorsqu'un signal modulé en phase ou en fréquence est amplifié dans une radio émettrice, il n'est pas nécessaire d'utiliser des amplificateurs linéaires, alors que lors de l'utilisation de QAM (qui contient une composante d'amplitude), la linéarité doit être maintenue. Malheureusement les amplificateurs linéaires sont moins efficaces et consomment plus de puissance, ce qui les rend moins attrayants pour les applications mobiles.

4.5.5. Diagrammes de constellation pour QAM

Les diagrammes de constellation montrent les différentes positions pour les états dans différentes formes de QAM (modulation d'amplitude en quadrature). Au fur et à mesure que l'ordre de la modulation augmente, le nombre de points sur le diagramme de constellation

QAM augmente également (Fig. 2.9).. Les diagrammes ci-dessous montrent des diagrammes de constellation pour une variété d'ordres de modulation.

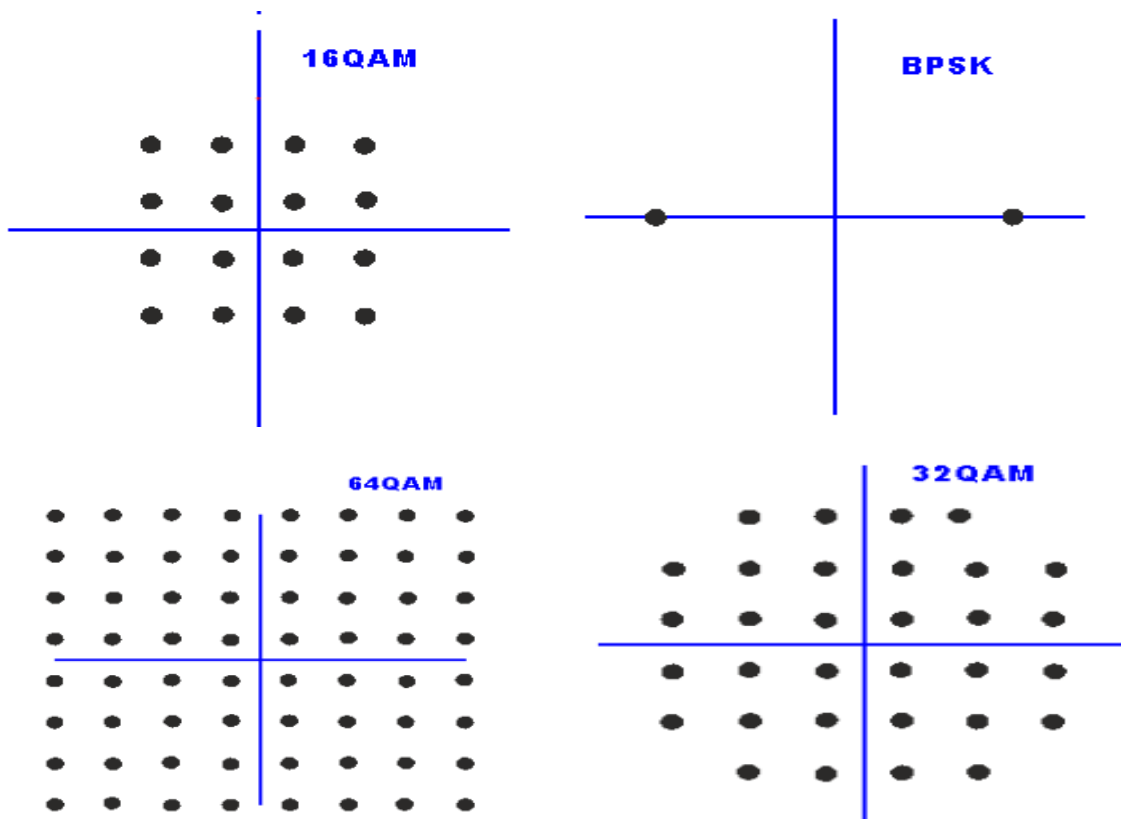


Figure 2.9. Constellations (BPSK), (16QAM), (32QAM), (64QAM).

Lors de l'utilisation du QAM, les points de constellation sont normalement disposés dans un carré avec des espacements verticaux et horizontaux égaux et, par conséquent, la plus courante forme de QAM utilise une constellation avec le nombre de points égal à une puissance de 2, c'est-à-dire 2, 4, 8, 16. . .

En utilisant des formats de modulation d'ordre supérieur, c'est-à-dire plus de points sur la constellation, il est possible d'émettre plus de bits par symbole. Cependant, il est préférable d'avoir des points moins proches et donc moins sensibles aux erreurs de bruit et de données [31].

4.5.6. Conversion des trames au format QAM 1024

Pour mettre une architecture de modulation M-QAM qui doit s'adapter facilement à plusieurs types de constellation, nous utilisons un codeur générique. Nous choisissons la modulation de type QAM 1024 comme une base de représentation des autres types de modulation (QAM 512, QAM 256...QPSK). La première étape du codage consiste en la conversion série / parallèle sur 10 bits du signal en entrée, car le QAM 1024 est de $n = 10$ bits ($M = 2^n = 2^{10} = 1024$). Puis en fonction du type de modulation choisi, les données sont

converties de manière à pouvoir être placées sur une constellation de type QAM 1024. Pour pouvoir placer les autres types de modulation dont leur nombre de bits est inférieur à 10 (par exemple le QAM 16 est de 4 bits seulement), on insère des bits nuls (zéro) pour compléter la trame de 10 bits. Ces modifications sont représentées sur la figure 2.9.

De plus, le placement des bits dans la trame utilise les propriétés de symétrie des différentes constellations [35,36]. En effet, si on regarde l'exemple de la figure (2.10) pour la modulation 16-QAM, on remarque que la répartition des points dans les différents quadrants de la constellation peut être obtenue simplement par une rotation d'un angle multiple de $\pi/2$ du quadrant principal.

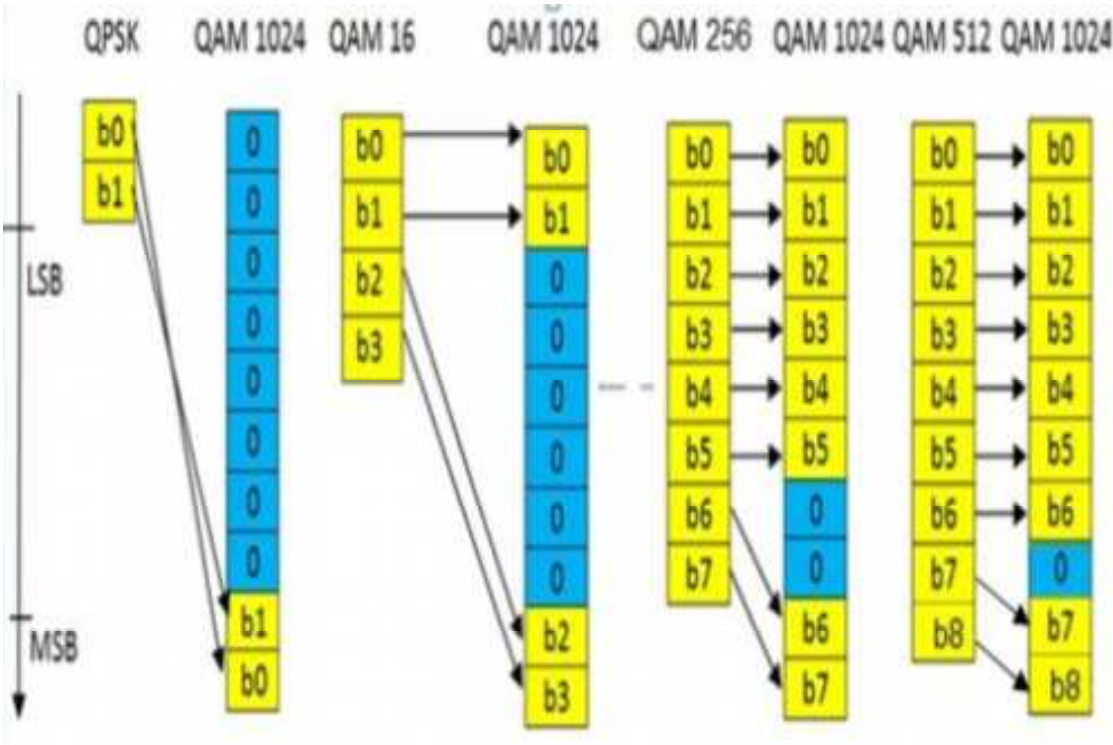


Figure 2.9. Conversion des trames au format QAM 1024.

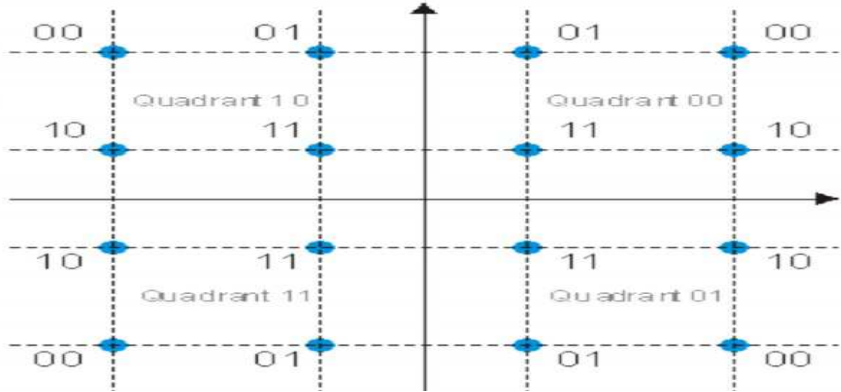


Figure 2.10. Propriété de symétrie dans la constellation QAM.

Ainsi, on remarque que les deux bits de poids forts de chaque symbole représentent le quadrant dans lequel il va se positionner:

- quadrant 1 ($\alpha = 0$): bits de poids forts 00.
- quadrant 2 ($\alpha = +\pi/2$): bits de poids forts 01.
- quadrant 3 ($\alpha = +\pi$): bits de poids forts 11.
- quadrant 4 ($\alpha = -\pi/2$): bits de poids forts 10.

En sortie du codeur, nous obtenons donc un bus véhiculant des trames QAM1024 à 10 bits, indépendamment du type de modulation souhaité en entrée.

5. Conclusion

Les performances pouvant être atteintes par les systèmes numériques dépendent fortement de la complexité des opérations arithmétiques à réaliser.

C'est pour cette raison qu'une nouvelle méthodologie de conception est proposée pour les systèmes numériques de transmission de signaux à haut débit.

Elle propose de transcrire dans le domaine numérique les fonctionnalités nécessaires à la modulation, plutôt que de transcrire les architectures analogiques existantes, trop complexes et mal adaptées au numérique.

Dans le chapitre suivant, nous abordons des techniques différentes permettant la conception d'architectures adaptées aux circuits numériques pour la modulation QAM des signaux.

CHAPITRE III
ARCHITECTURE DU
MODULATEUR QAM A
BASE DU PROCESSEUR
CORDIC

1. Introduction

L'étude et la conception de modulateurs numériques en utilisant le circuit FPGA est très intéressant en raison de la possibilité de validation des fonctionnalités et de l'adaptation versatile en cas de changement du type de modulation. En outre l'implantation sur circuit FPGA engendre des contraintes dues à la vitesse de fonctionnement limitée de ce circuit et à la surface occupée par le nombre de cellules élémentaires utilisées. Ces contraintes engendrent un chemin de parcours critique du signal pouvant brider la vitesse de traitement du modulateur. Cependant, en tenant compte des spécificités architecturales du système de modulation considéré, on peut optimiser ces limitations voire même les compenser en utilisant des opérations arithmétiques élémentaires [32].

Pour cela, dans ce chapitre, on s'intéresse à la modulation QAM dont la particularité consiste à contrôler la phase et l'amplitude de la sinusoïde de sortie. Elle est donc l'une des modulations les plus complexes utilisées de nos jours dans les protocoles de transmission. Plusieurs travaux concernant l'architecture de générateurs de sinusoïdes existent.

2. synthèse numérique de sinusoïdes existant

Plusieurs études ont été menées sur l'architecture des modulateurs numériques QAM dans le système de transmission. Ils se sont concentrés sur l'élément le plus nécessaire dans cette architecture qui est le générateur de sinusoïde dont différents résultats sur le circuit FPGA ont été obtenus.

En 2005, Gérald A. [55] a étudié la génération de sinusoïdes par l'étude comparative entre mémoire ROM avec compression, CORDIC pipeline et CORDIC pipeline en supprimant l'étage d'accumulateur de phase. En utilisant le circuit FPGA de type Altera FLEX10K200E-2 dont la fréquence maximale de fonctionnement annoncée par le constructeur pour ce type de circuit est de 150 MHz. Ces techniques employées ont donné les résultats suivants sur ce circuit en 16 bits : La fréquence d'échantillonnage pour le générateur à mémoire ROM avec compression est égale 41M-échantillons/s et nécessite 5543 cellules élémentaires ; la fréquence d'échantillonnage de processeur CORDIC pipeline est égale 101M-échantillons/s et 1430 cellules ; la fréquence d'échantillonnage de processeur CORDIC pipeline sans accumulateur de phase est égale 110M-échantillons/s et 980 cellules.

En 2007, Liu Zhi [56] à étudié la récupération de porteuse efficace pour haut-ordre QAM et en plus a étudié le détecteur de phase classique piloté par décision, l'algorithme emploie un détecteur de phase amélioré. En utilisant le câble coaxial et canal AWGN comme canal de

transmission. La fréquence résultante est 960 KHZ (canal AWGN) et 690 KHz (Câble coaxial) et génère un débit de transmission 6.9 Mbps.

En 2010, Mihra [58] a étudié l'implantation de processeur CORDIC pipeline pour le générateur de sinusoïde en utilisant le circuit FPGA spartan3- XC3S200-5 ft 256 de technologie xilinx, le logiciel ISE 10.1 et Model Sim 6.3XE pour la simulation temporelle. Ce logiciel est un outil de Mentor Graphics. Il fournit un environnement complet de simulation et débogage pour les designs complexes en ASIC et en FPGA. Il supporte plusieurs langages de description, dont le system verilog, le VHDL et le system C. Le résultat de puissance de consommation en 16 bits est 0.03942 watt, avec utilisation de 10 % des slices, 9 % des mémoires de décodages (LUT), 21 % des blocs d'entrées/sorties (IOB), 8% de slice flip flop et 12 % de GCLK. Ces résultats montrent que ce processeur n'est pas satisfaisant.

En 2011, B.V.V. Satyanarayana [59], a étudié la conception QAM modulation d'amplitude en quadrature pour l'application sans fil. Il utilise le circuit FPGA spartan-3e, le langage VHDL, ModelSim XE III 6.0d et Xilinx ISE8.1i. Ce modulateur 16 QAM basé sur l'algorithme CORDIC en transmission et réception, les résultats obtenus sont les suivants : 2922 nombre de slice et 4674 LUT.

En 2012, Mandal [60] a étudié la conception reconfigurable d'un processeur CORDIC pipeline pour le générateur numérique de fréquence, en utilisant pour l'implantation le circuit FPGA (Field Programmable Gate Array) spartan3-XC3S50pq208-5 de technologie xilinx, le logiciel ISE 10.1 et Model Sim pour la simulation temporelle. Cette simulation permet de vérifier le comportement du circuit en appliquant des vecteurs test ou stimuli (test bench) et en analysant les réponses obtenues. La simulation de l'ensemble comportemental permet de vérifier le fonctionnement d'un circuit avant son implémentation mais ne tient pas compte des délais de propagation et des caractéristiques temporelles du circuit. Cet algorithme n'a pas obtenu de bonnes performances car il consommé en 16 bits une puissance de 0.096 watt, 3% des registres, 2% des mémoires de décodages (LUT), et 35 % des blocs d'entrées/sorties (IOB). Ces résultats indiquent que ce processeur est de moyenne rapidité.

En 2014, kavya sharat [61] a étudié le générateur de sinusoïde à base d'opérateur CORDIC avec recodage d'angle, en employant le circuit FPGA Xilinx Virtex 5 XC5VF70T, le logiciel ISE 12.2 et langage Verilog. Le résultat de ces travaux se traduit par l'obtention d'une fréquence d'échantillonnage de 241.106 MHz et 123 cellules logiques utilisées en 8 bits.

En 2014, yidong liu [62] a étudié l'implémentation de générateur de sinusoïde à base d'algorithme CORDIC pipeline, en utilisant le circuit FPGA Xilinx spartan xc3s500e et le logiciel ISE 12.3. Il obtient un générateur de sinusoïde dont les performances en 32 bits sont les suivantes : BER= 10^{-8} dB, SNR=2dB et la fréquence=40 MHz.

Tous les travaux cités précédemment se sont tous concentrés sur le même but qui est l'implantation de générateurs de sinusoïdes sur circuit FPGA. Dans ce manuscrit nous proposons différents algorithmes pour générer le modulateur QAM adapté au haut débit.

Aujourd'hui il est possible, de présenter, développer et implanter des différents modulateurs numériques d'amplitude et de phase QAM, pour la transmission rapide [37].

L'architecture proposée dans notre étude introduit des méthodes permettant l'efficacité et la rapidité du modulateur numérique.

. Dans cette thèse sont étudiées de nouvelles architectures compatibles avec l'évolution des débits et des fonctionnalités des protocoles actuels. L'architecture pour la modulation est basée sur une version modifiée de l'algorithme CORDIC. Dans ce contexte ; nous avons mené trois études :- La première est une étude portant sur un modulateur complet QAM₁, à base de CORDIC itératif pour le fonctionnement de son algorithme. Ceci augmente le nombre des cellules utilisées dans le circuit FPGA et réduit ainsi la fréquence fonctionnement du modulateur.

- La deuxième est une étude portant sur un modulateur complet QAM₂ à un étage dont l'architecture comporte un nombre de cellules élémentaire dans le circuit FPGA inférieur que le cas précédent mais la fréquence des échantillons est ainsi augmentée.

- La troisième étude est l'architecture QAM₃ semblable au modulateur QAM₂ mais avec suppression de l'accumulateur de phase du générateur de sinusoïdes. Cette configuration réduit le nombre de cellules et augmente la vitesse de production des échantillons.

3. L'architecture d'un modulateur numérique QAM

Couramment, le modulateur numérique QAM est composé de trois éléments fonctionnant en mode numérique comme le codeur, l'accumulateur de phase et le générateur de sinusoïde (Figure 3.1), mais le convertisseur numérique analogique et le filtre passe bas fonctionnent en mode analogique [33,34]. Les détails concernant ces éléments sont indiqués dans ce qui suit.

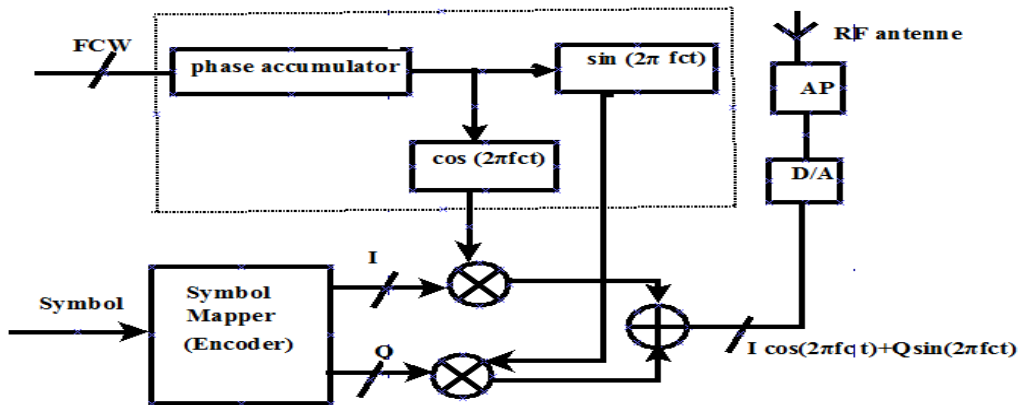


Figure 3.1.L'architecture d'une modulation numérique QAM.

3.1. Le codeur

Pour tout transport de données numériques, le codage en ligne est souvent utilisé.

Le codage est la première étape qui consiste à ramener les informations que nous voulons échanger à un ensemble d'informations binaires, à l'aide de techniques de codage. Il se fait en utilisant un codeur. Ce dernier est constitué d'une mémoire de petite taille d'adressage qui permet de convertir les séries de M-bits composant un symbole, pour les rendre compatibles avec le type de modulation choisi. Pour une modulation de phase par exemple, une série de M-bits sera remplacée par la valeur correspondante de phase du signal sinusoïdal à produire en sortie du modulateur. La figure 3.2 présente des exemples de codeur.

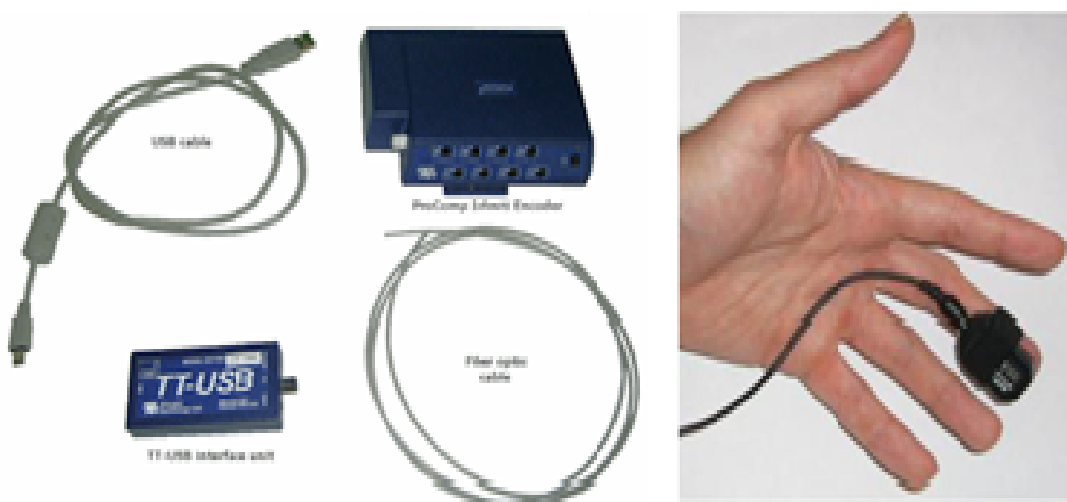


Figure 3.2.Exemples de codeur.

3.2. L'accumulateur de phase

La deuxième étape pour la génération d'un modulateur numérique QAM est le choix de l'accumulateur de phase. L'architecture générale de l'accumulateur de phase est présentée sur la figure 3.3. Elle est composée principalement de deux additionneurs et d'un registre. L'accumulateur de phase est piloté par un mot de contrôle de la fréquence F_{cw} correspondant à l'incrément de phase appliqué à chaque période d'horloge du système. Ce dernier est nécessaire pour générer les valeurs de phase (thêta), afin d'avoir en sortie un signal sinusoïdal. Son architecture est présentée sur la figure (3.3). La fréquence effective « f » résultante du signal modulé est égale à :

$$f = \frac{f_{clk} \cdot F_{cw}}{2\pi} \quad (3.1)$$

f_{clk} : fréquence d'horloge.

F_{cw} détermine la fréquence f effective résultante du signal modulé, en fonction de la fréquence d'horloge f_{clk} de l'architecture hôte.

Θ_n : déphasage lorsque la modulation considérée utilise ce paramètre pour transmettre des informations. Le k^e mot de phase produit par cet étage vaut :

$$\theta(k) = k \cdot F_{cw} + \theta_n \quad (3.2)$$

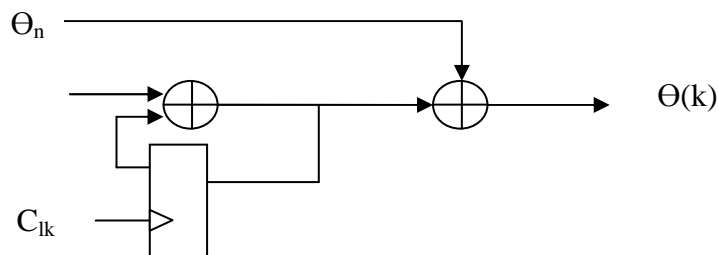


Figure 3.3. Architecture de l'accumulateur de phase.

L'accumulateur de phase applique également, le cas échéant, un déphasage θ_n lorsque la modulation considérée utilise ce paramètre pour transmettre des informations.

On ajoute la phase « θ_n » qui correspond au symbole transmis k . Pour cette dernière opération, on utilise un additionneur rapide. Il y a plusieurs architectures d'additionneurs, mais il faut noter que plus une architecture permet d'effectuer le calcul rapidement, plus sa surface

augmente. C'est pour cela que l'utilisation d'un additionneur dans l'architecture est nécessaire pour obtenir un compromis entre la surface d'implantation et la rapidité de calcul.

3.3. Le convertisseur numérique analogique associé

Ces convertisseurs sont les derniers éléments de la chaîne de traitement. Ils convertissent les échantillons en signaux adaptés au médium de transmission. Il s'agit de la seule partie de l'architecture qui fonctionne en mode analogique.

3.4. Le générateur de sinusoïde

Le générateur de sinusoïde est la partie la plus critique de l'architecture du modulation numérique. Nous avons utilisé plusieurs techniques pour obtenir une meilleure fréquence d'échantillonnage [37,38]. Cette technique permet de produire des échantillons numériques à partir des paramètres que lui fournissent le codeur et l'accumulateur de phase [39]. La figure (3.6) présente la sortie de l'accumulateur de phase qui est reliée au générateur de sinusoïdes, qui doit calculer le cosinus du terme de phase produit par l'étage accumulateur [40]. Habituellement, le générateur de sinusoïde ne génère pas un signal, il donne une seule valeur en fonction de la valeur d'entrée « théta » ce qui donne comme sortie $\cos(\theta)$ ou bien $\sin(\theta)$.

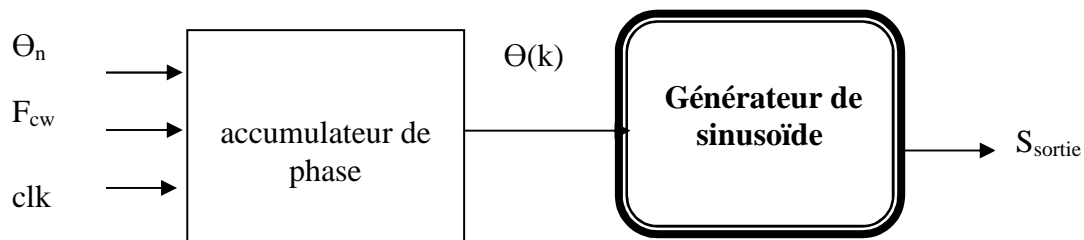


Figure 3.4. Schéma de l'accumulateur de phase avec le générateur de sinusoïdes.

Afin d'accélérer la vitesse de transmission du générateur de sinusoïdes nous utilisons l'algorithme CORDIC modifié pour la génération de sinusoïdes.

3.4.1. L'algorithme CORDIC

a. Introduction

CORDIC: COordinate Rotation DIGital Computer, proposé par Jack Volder. Il permet de calculer les fonctions trigonométriques avec une suite d'opérations arithmétiques très simples: addition, soustraction et décalage. Il a été généralisé pour calculer des fonctions exponentielles, la division, la multiplication et la racine carrée. Ce processeur est utilisé dans

les calculatrices de poche, Intel 80x87, 80486 et Motorola 68881[41,42]. CORDIC s. Il a été développé par Jack E. Volder , mais a bénéficié de nombreuses améliorations et extensions par la suite. L'intérêt de cet algorithme est qu'il permet l'évaluation de fonctions complexes sans employer de multiplieurs. Il a donc un usage conforme sur les FPGA pour lesquels, l'implantation d'un multiplieur, même avec des cellules dédiées, reste coûteuse en ressources et limite grandement la fréquence de fonctionnement du circuit cible.

b.Principe de l'algorithme CORDIC

Le principe de l'algorithme CORDIC comme décrit par Volder, est relativement simple [48 ,49].Il s'agit d'effectuer une rotation matricielle d'angle Θ du vecteur $\vec{v} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ sur le cercle trigonométrique. Cette rotation peut s'écrire sous la forme suivante :

$$\vec{v}' = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \cdot \vec{v} \quad (3.3)$$

En plaçant $\cos \Theta$ en facteur et cela peut également s'écrire :

$$\vec{v}' = \cos\theta \begin{pmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{pmatrix} \vec{v} \quad (3.4)$$

En effet, pour une valeur de Θ prise dans l'intervalle $]-\frac{\pi}{2}; \frac{\pi}{2}[$, Volder a démontré qu'il existe une suite de valeurs d_i , avec $d_i \in \{-1, 1\}$, pour lesquelles

$$\theta = \sum_{i=0}^{\infty} d_i \arctan 2^{-i} \quad (3.5)$$

Ainsi, effectuer une rotation d'angle Θ peut s'écrire sous la forme matricielle suivante

$$\vec{v}' = \prod_{i=0}^{\infty} \cos (d_i \cdot \arctan 2^{-i}) \cdot \prod_{i=0}^{\infty} \begin{pmatrix} 1 & \cos (d_i \cdot 2^{-i}) \\ d_i \cdot 2^{-i} & 1 \end{pmatrix} \cdot \vec{v} \quad (3.6)$$

Pour simplifier cette expression, on peut poser :

$$C_{\infty} = \prod_{i=0}^{\infty} \cos(d_i \arctan 2^{-i}) \prod_{i=0}^{\infty} \cos (\arctan 2^{-i}) \quad (3.7)$$

Car la fonction \cos est paire

Par exemple dans le figure (3.7), il y'a deux échantillons M_1 et M_2 , nous calculons les coordonnées de $M_2(X_2, Y_2)$ par rapport aux coordonnées de $M_1(X_1, Y_1)$ de CORDIC pipeline comme suit

X_2 est égale X_1 moins l'espace entre X_1 et X_2 .

Y_2 est égale Y_1 plus l'espace entre Y_1 et Y_2 .

Espace entre X_1 et X_2 est égale $d_k \cdot 2^{-k}$, avec 2^{-k} est une valeur et $d_i \in \{-1, +1\}$ est le sens de micro-rotation. Il y'a deux cas :

$\left\{ \begin{array}{l} -1 \text{ la direction inverse.} \\ +1 \text{ la même direction.} \end{array} \right.$

Aussi calculons la phase instantanée qui est égale à la phase précédente entre les deux échantillons moins la valeur du sens de micro produit par la valeur $\arctan 2^{-k}$. La même chose pour le CORDIC pipeline sans accumulateur, mais dans ce cas nous supprimons l'accumulateur de phase \arctan qui devient $\arctan x \approx x$ qui génère une nouvelle architecture de ce processeur. Dans la figure (3.7), on voit qu'il y a X et Y seulement et on n'a pas d'accumulateur de phase.

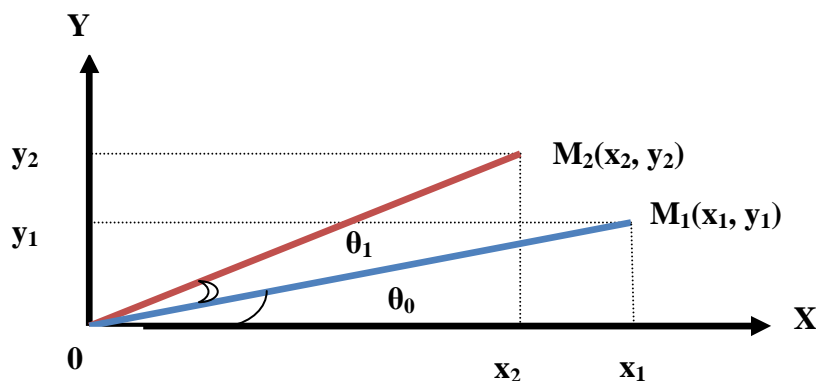


Figure 3.5. Décomposition de l'angle θ en plusieurs micro-rotations.

c. Les différentes utilisations de CORDIC

L'algorithme CORDIC est utilisé dans plusieurs domaines, nous pouvons citer par exemple ces applications :

- Calcul des fonctions trigonométriques dans les calculatrices scientifiques,
- La modulation en bande latérale unique,
- La transformée de Fourier discrète, directe et rapide, le filtrage fréquentiel (filtres orthogonaux et filtres d'onde),
- La modélisation adaptative de processus non stationnaire (filtrage récursif optimal, filtre de Kalman), ... etc.

d. Les modes de fonctionnement d'un processeur CORDIC

Suivant son mode de fonctionnement, le CORDIC peut être utilisé à différents modes.

- soit connaître l'angle d'un vecteur $\vec{M}(x,y)$ -mode vecteur.
- soit lui demander de réaliser la rotation d'un vecteur d'un angle θ -mode rotation.

Mode	Pré-conditions	Post-conditions	Sens de la progression
Rotation	$x_0 = x$ $y_0 = y$ $z_0 = \theta$	$x' = x \cdot \cos\theta - y \cdot \sin\theta$ $y' = y \cdot \cos\theta + x \cdot \sin\theta$ $z' = 0$	$d_i =$ $\begin{cases} -1 & \text{si } z_i < 0 \\ +1 & \text{sinon} \end{cases}$
	$x_0 = 0$ $y_0 = 1$ $z_0 = \theta$	$x' = -y \cdot \sin\theta$ $y' = x \cdot \sin\theta$ $z' = 0$	
Vecteur	$x_0 = x$ $y_0 = y$ $z_0 = \theta$	$x' = 0$ $y' = \sqrt{x^2 + y^2}$ $z' = \tan^{-1}\left(\frac{y}{x}\right)$	$d_i = \begin{cases} -1 & \text{si } x_i < 0 \\ +1 & \text{sinon} \end{cases}$

Tableau 3.1. Mode de fonctionnement d'un processeur CORDIC.

e. L'architecture de l'algorithme CORDIC

L'algorithme CORDIC présente donc un très gros avantage qui est l'absence d'opération arithmétique complexe pour parvenir au résultat final

Pour le calcul rapide des différents paramètres des échantillons du générateur de sinusoïde, on utilise l'algorithme CORDIC pour les trois architectures [48,49] :

- CORDIC itératif à 8 étages.
- CORDIC pipeline [43].
- CORDIC pipeline sans accumulateur de phase.

On génère ainsi trois architectures différentes de modulateurs : QAM₁, QAM₂ et QAM₃.

4. Les différentes techniques de modulation QAM

Dans cette partie, nous définissons en détail les trois techniques, puis une étude comparative entre les architectures est présentée.

4.1. Description de la première technique QAM₁

Pour le modulateur numérique QAM, le générateur de sinusoïde est à base de CORDIC itératif. Avec cette technique, il faut environ n+1 itérations pour obtenir la valeur d'un échantillon de sinusoïde avec une précision de n bits. La fréquence des échantillons diminue

donc. Ce modulateur QAM₁ est composé de codeur, d'accumulateur de phase (vu précédemment) et le générateur de sinusoïde à base de CORDIC itératifs (figure 3.6).

CORDIC itératif

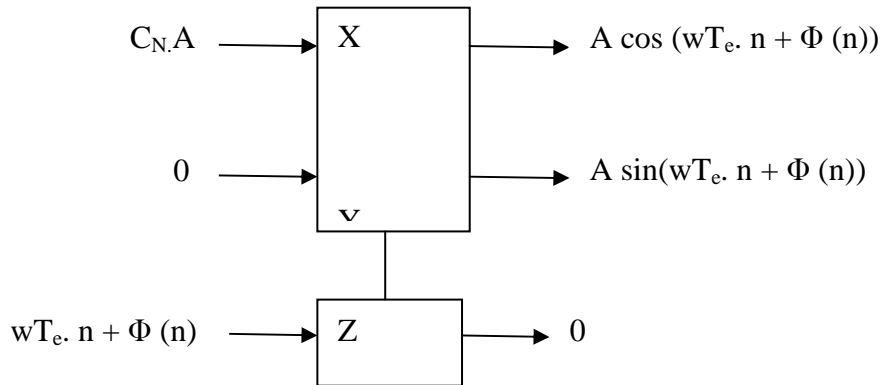


Figure. 3.6. Le processeur CORDIC itératif.

Au début du fonctionnement du circuit, les blocs X et Y sont pré-chargés avec les valeurs $C_n.A$ et 0 respectivement, et Z est chargé avec la valeur de l'angle dont il faut calculer le cosinus, à savoir θ . À chaque itération, les nouvelles valeurs de X et Y sont calculées en fonction des valeurs de l'étape précédente, et du sens de la rotation donné par le signe de Z. La valeur de Z est modifiée pour refléter la rotation qu'il reste à effectuer pour atteindre la rotation d'angle θ . Le calcul est terminé lorsque $Z = 0$, à la précision près du circuit sur lequel cette architecture est implantée. Il faut environ $n+1$ itérations pour obtenir la valeur d'un échantillon de sinusoïde avec une précision de n bits. La position des registres est telle qu'à chaque cycle d'horloge, X Y et Z sont mis à jour une fois. Par conséquent, $n+1$ cycles d'horloges seront nécessaires pour produire un échantillon. La fréquence d'échantillonnage théorique correspondante sera donc de $f/(n+1)$, où f est la fréquence de fonctionnement du circuit sur lequel est implantée cette architecture. De plus, cette architecture utilise :

- trois additionneurs soustracteurs complets sur n bits ;
- une ROM comportant $n+1$ valeurs $\arctan 2^{-i}$, chacune ayant une précision de n bits, soit $n^2 + 1$ bits au total ;
- deux décaleurs à barillet sur n bits, dont la complexité est comparable à celle d'un additionneur de même taille

L'architecture fait appel à des décaleurs à barillet qui engendrent un bloc combinatoire volumineux et dont le chemin critique est important. De plus, nous cherchons à obtenir la plus grande fréquence d'échantillonnage possible, et cette architecture pour une précision de 8 bits

nécessite 9 cycles d'horloge pour produire un seul échantillon, ce qui n'est généralement pas satisfaisant.

La principale limitation de l'architecture précédente est qu'elle restreint fortement la fréquence d'échantillonnage théoriquement possible, d'un facteur au moins égal à la précision en bits des échantillons de sortie. Il est cependant possible, mais au prix d'une occupation de surface plus élevée, de produire un échantillon de sinusöide par cycle d'horloge, en utilisant une version pipeline de l'algorithme CORDIC

4.2. La deuxième technique QAM₂

Cette technique est meilleure par rapport à l'architecture précédente QAM₁, Elle possède une architecture dont le générateur de sinusöide est à base de CORDIC pipeline. Cette configuration permet d'obtenir un échantillon de sinusöide en 1 seul cycle d'horloge.

CORDIC pipeline

Cette architecture est composée d'une mémoire pour contenir les valeurs des $\arctan 2^{-k}$, et (N+1) additionneur soustracteurs pour chacun des trois chemins de calcul (X, Y et Z) et aussi 2(N + 1) décalages [45,46].

Pour n bits il y'a n+1 étages, chaque étage du pipeline n'est chargé d'effectuer qu'une seule micro-rotation, toujours la même, dont seul le sens peut être modifié [47].

C'est-à-dire que le k^{ème} étage correspond au système d'équations :

$$\begin{cases} x_{k+1} = x_k - y_k \cdot d_k \cdot 2^{-k} \\ y_{k+1} = y_k + x_k \cdot d_k \cdot 2^{-k} \\ z_{k+1} = z_k - d_k \cdot \arctan 2^{-k} \\ d_k = \text{signe}(z_k) \end{cases} \quad (3.7)$$

Cette fois, la fin du calcul n'est plus déterminé par le fait que le contenu de Z vaut 0, mais par le fait que n + 1 micro-rotations ont été effectuées. Cela est suffisant pour obtenir un échantillon de sinusöide à la précision binaire désirée. En plus du fait de produire un échantillon à chaque cycle d'horloge, cette architecture possède l'avantage de ne pas utiliser de décaleurs à barillet puisque pour chaque étage du pipeline, le décalage à effectuer est connu par avance et ne dépend que de k (figure 3.7).

Ce processeur augmente la fréquence d'échantillonnage ainsi que le débit (figure 3.7). La fréquence d'échantillonnage est égale à la fréquence d'horloge, mais nous recherchons une architecture pour générer une fréquence supérieure à la fréquence d'horloge [48,49].

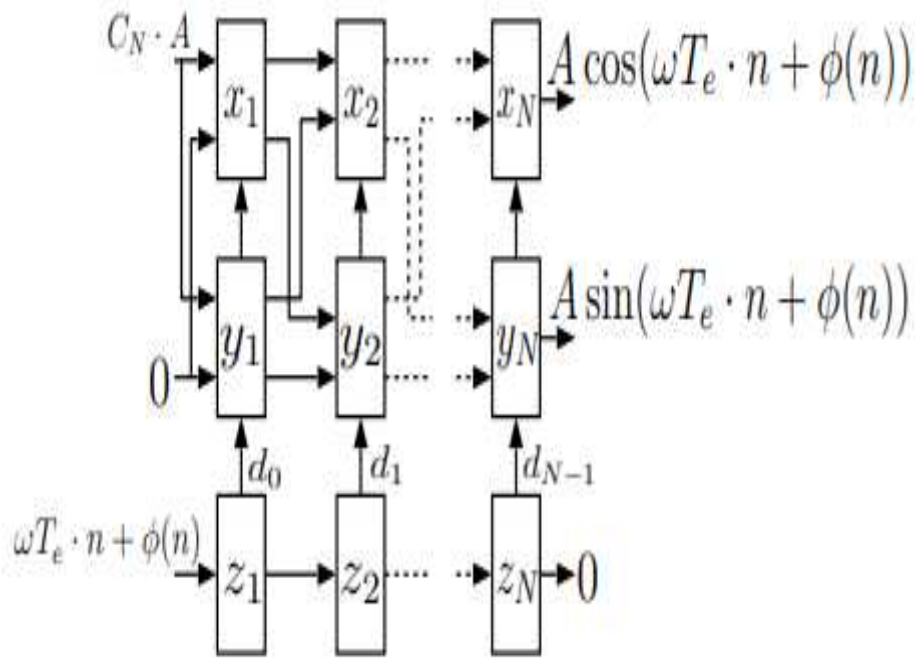


Figure 3.7. Le processeur CORDIC pipeline.

4.3. L'architecture QAM₃

L'architecture QAM₃ est la même que celle de QAM₂ mais avec CORDIC pipeline sans accumulateur de phase constituant le générateur de sinusoïde. La mise en œuvre de l'algorithme de fonctionnement permet d'obtenir en un seul cycle d'horloge une fréquence d'échantillonnage supérieure à celle du précédent. En supprimant l'accumulateur de phase, le nombre de cellules requis diminue. Il s'ensuit ainsi une augmentation de l'efficacité et la rapidité de ce modulateur.

3.3.1. CORDIC pipeline sans accumulateur de phase

Dans ce processeur, on utilise (N+1) additionneurs soustracteurs complets pour chacun des trois axes de calcul (X, Y, Z) et aussi 2(N+1) décalages, ne nécessitant pas d'élément de logique combinatoire.

Pour n bits, il y a (n+1) étages.

Le k^{ème} étage correspond au système d'équations :

$$\begin{cases} x_{k+1} = x_k - y_k \cdot d_k \cdot 2^{-k} \\ y_{k+1} = y_k + x_k \cdot d_k \cdot 2^{-k} \\ \alpha_0 = \sum_{k=1}^{\infty} 2^{-k} \\ \Theta = \alpha_0 + \sum_{k=1}^{\infty} C_k 2^{-k} \end{cases} \quad (3.8)$$

On peut pour cela constater que, pour une valeur x suffisamment petite, $\arctan x \approx x$

$$\theta = \sum_{i=1}^{\infty} d_i 2^{-i} \quad (3.9)$$

$$\theta = \sum_{i=1}^{\infty} (1 + c_i) 2^{-i} \quad (3.10)$$

$$\theta = \sum_{i=1}^{\infty} 2^{-i} + \sum_{i=1}^{\infty} c_i 2^{-i} \quad (3.11)$$

C_i coefficient $\in \{-1,1\}$,

$$\alpha_0 = \sum_{i=1}^{\infty} 2^{-i} \quad (3.12)$$

$$\theta = \alpha_0 + \sum_{i=1}^{\infty} c_i 2^{-i} \quad (3.13)$$

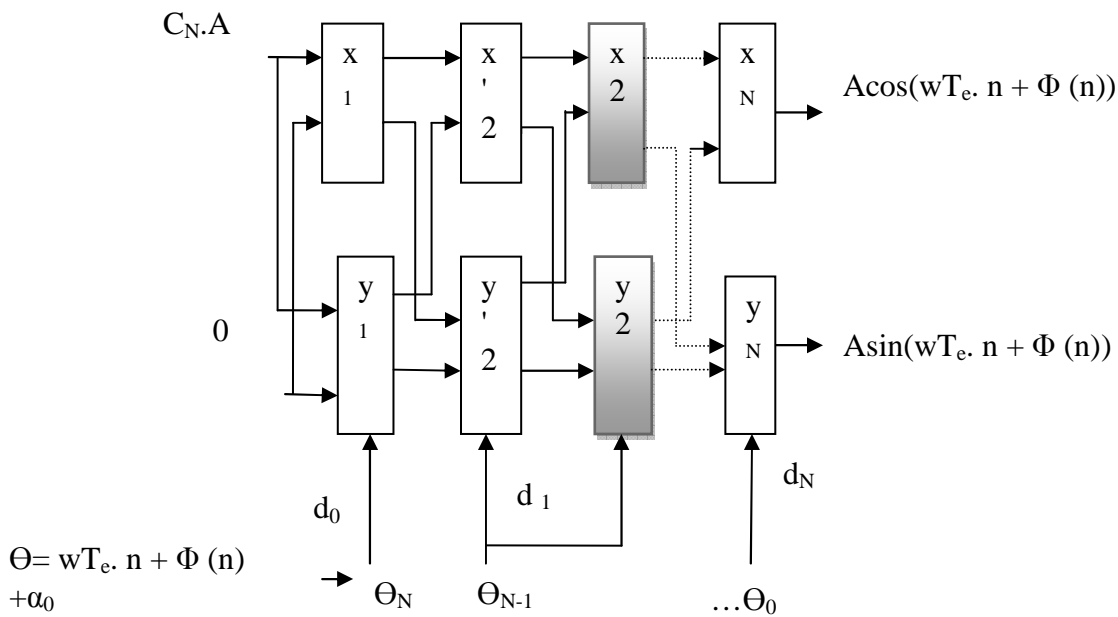


Figure 3.8. Le processeur CORDIC pipeline sans accumulateur de phase.

L'architecture résultante est présentée sur la figure 3.8. La partie grisée correspond à l'une des micro-rotations ajoutées pour compenser l'erreur sur le facteur d'échelle. L'architecture exposée ici permet au circuit sur lequel elle est implantée d'atteindre une fréquence d'horloge supérieure à celle obtenue avec les autres algorithmes présentés, car elle réduit le chemin critique de la partie combinatoire de chacun des étages du pipeline en supprimant l'étage de détection de signe, et l'un des additionneurs / soustracteurs. Le tableau (2.2) illustre l'étude comparative entre CORDIC itératif, CORDIC pipeline et CORDIC pipeline sans accumulateur de phase:

CORDIC Itératif	CORDIC pipeline	CORDIC pipeline sans accumulateur de phase
Occupation de cellule élevée	Occupation de cellule élevée.	Faible occupation de cellule
Fréquence d'échantillonnage est égale la fréquence d'horloge de circuit FPGA $\div (n+1)$ itérations pour n bits	Fréquence d'échantillonnage est égale la fréquence d'horloge de circuit FPGA.	Fréquence d'échantillonnage supérieure aux deux précédents
Débit minimal	Débit moyen	Débit maximal
n itérations, pour (n-1) nombres de bits	n étages, pour (n-1) nombres de bits.	n étage, pour (n-1) nombre de bit avec l'étage d'accumulateur de phase supprimé.
trois additionneurs soustracteurs complets sur n bits ; – une ROM comportant n+1 valeurs $\arctan 2^{-i}$, ayant une précision de n bits, soit $n^2 + 1$ bits au total ;	(n+1) additionneurs soustracteurs pour chacun des trois chemins de calcul (X, Y et Z).	(n+1) additionneurs soustracteurs complets pour chacun des trois axes de calcul (X, Y, Z).
2 décaleurs à barillet sur n bits a complexité identique avec additionneur de même taille	2(n+1) décalages et une mémoire pour contenir les valeurs des $\arctan 2^{-k}$.	2(n+1) décalages

Tableau 2.2. Indique les différences entre les trois algorithmes.

Ensuite nous présentons la comparaison entre les trois modulateurs QAM.

Les modulateurs	QAM1	QAM2	QAM3
Les paramètres			
Le nombre de cellules (cell)	Large occupation	moyenne occupation de cellules	Faible occupation de cellules
La fréquence de fonctionnement (MHz)	Faible	Moyenne	Grande
Le débit (Mbits/s)	Faible	Moyenne	Très haut débit
La puissance de consommation(Watt)	Grande	Moyenne	Faible

Tableau 3.3. Étude comparative entre trois modulateurs QAM.

Toutes les architectures seront implantées sur le circuit FPGA, en utilisant le langage VHDL et le logiciel ISE 12.2.

5. Conclusion

Les performances atteintes par les systèmes numériques synchrones, et les FPGA en particulier dépendent fortement de la complexité des opérations arithmétiques à réaliser, et de la longueur maximale du chemin critique entre deux registres. L'analyse de la structure interne d'un FPGA nous a permis de comprendre les avantages, mais aussi les contraintes posées par ces circuits. C'est pour cette raison qu'une nouvelle méthodologie de conception est proposée pour les systèmes numériques de transmission des signaux à haut débit. Elle propose de transcrire dans le domaine numérique les fonctionnalités nécessaires à la modulation, plutôt que de transcrire les architectures analogiques existantes, trop complexes et mal adaptées au numérique. Dans ce chapitre nous avons présenté les différentes techniques pour proposer une architecture QAM adaptée aux circuits numériques pour la modulation de signaux à haut débit.

Toutes les architectures seront implantées sur le circuit FPGA, en utilisant le langage VHDL et le logiciel ISE 12.2.

***CHAPITRE IV
SIMULATION ET
IMPLEMENTATION DU
MODULATEUR QAM***

1.Introduction

L'objectif de ce chapitre est la conception et l'implantation à partir du langage VHDL, un modulateur d'amplitude et de phase QAM à base d'opérateurs CORDIC adapté à la transmission de données à très haut débit en télécommunications.

La simulation du système est faite pour vérifier la validation de l'algorithme, prévoir le comportement du système sous l'action d'un événement particulier et visualiser l'évolution temporelle [50]. La dernière phase de cette conception consiste à faire implanter ce modulateur sur un circuit FPGA afin de vérifier sa faisabilité technique et d'évaluer ses performances. Donc, ce chapitre s'occupe de la conversion vers un format exécutable (langage de description matériel) du modulateur QAM [51], puis de la vérification du circuit intégré correspondant à ce modulateur en utilisant l'outil ISE 12.2 de Xilinx [52]. Pour cela, ce chapitre comporte deux parties A et B dont la première est consacrée à la présentation du langage VHDL [53] et le circuit FPGA [54] et la deuxième traite les résultats de simulation et d'implantation.

2. présentation du langage VHDL et circuit FPGA : Partie A :

A .Le langage VHDL

A .1. Définition

Le langage VHDL permet la description de tous les aspects d'un système matériel: son comportement, sa structure et ses caractéristiques temporelles. Par système matériel, on entend un système électronique arbitrairement complexe réalisé sous la forme d'un circuit intégré ou d'un ensemble de cartes. Le comportement définit la ou les fonctions que le système remplit (par exemple le comportement d'un microprocesseur comporte, entre autres, des fonctions arithmétiques et logiques). La structure définit l'organisation du système en une hiérarchie de composants (par exemple un microprocesseur est constitué d'une unité de contrôle et d'une unité opérative. Cette dernière est elle-même, entre autres, constituée d'un composant réalisant les opérations arithmétiques entières et d'un composant réalisant les opérations arithmétiques en virgule flottante). Les caractéristiques temporelles définissent des contraintes sur le comportement du système (par exemple les signaux d'un bus de données doivent être stables depuis un temps minimum donné par rapport à un flanc d'horloge pour qu'une opération d'écriture dans la mémoire soit valable).

La description d'un système matériel en VHDL est en plus simulable. Il est possible de lui appliquer des stimuli (également décrits en VHDL) et d'observer l'évolution des signaux du modèle dans le temps. La sémantique d'un modèle VHDL est basée sur un modèle de

simulation discrète dirigée par les événements. Le modèle VHDL ne peut prendre un nouvel état en simulation que lorsqu'un stimuli change de valeur et que ce changement est propagé dans le modèle. Le langage VHDL définit des règles précises pour l'évaluation de l'état d'un modèle en présence d'un changement de stimuli. Ces règles garantissent que l'évaluation aboutisse au même résultat quelque soit l'outil de simulation utilisé.

Le langage VHDL est aussi très utilisé pour la synthèse, par exemple pour dériver automatiquement un circuit à base de portes logiques optimisé à partir d'une description au niveau RTL (Register-Transfer Level) ou algorithmique. Cette application très importante du langage sort toutefois du cadre de sa définition standard et comporte des limitations dont les plus importantes seront présentées dans ce travail.

Le langage VHDL est un standard IEEE depuis 1987 sous la dénomination IEEE Std. 1076-1987 (VHDL-87). Il est sujet à une nouvelle version tous les cinq ans. La dernière version est celle de 1993 (IEEE Std. 1076-1993, VHDL-93). Elle corrige certaines incohérences de la version initiale et ajoute de nouvelles fonctionnalités [63,64].

A.2. Structure d'un programme sous VHDL

Un programme écrit sous VHDL obéit à la structure suivante [65, 66] :

-Entête

C'est une partie facultative, elle renferme des informations concernant le programmeur, la description du programme en général, la date de rédaction et toute information qui semblera importante pour celui qui rédige le programme.

-Déclaration des librairies

En deuxième lieu vient la déclaration des librairies et des paquetages que le programmeur juge nécessaire pour son programme.

-Déclaration d'entité

En troisième lieu vient la déclaration d'entité avec les signaux à utiliser dans tout le programme avec leur direction.

-Déclaration d'architecture

Juste après l'entité vient la déclaration de l'architecture qui décrit l'entité. Dans cette partie les signaux internes sont déclarés et les variables propres à cette même architecture. Puis le programme est rédigé prêt pour l'emploi (Voir la Figure 4.1.)

```

Library IEEE;
Use IEEE.std_logique_1164.all;
Entity <entity name> is port
    (<list of ports or design inputs and outputs> );
End <entity name>;
Architecture < architecture name> of < entity name> is
    < In this section define signals and constants>
Signal <signal name> : Data type;
Begin
    < Concurrent statements>
    <Process name>: process (sensitivity list)
Begin
    <sequential statements>
End ;
End < architecture name>;

```

Figure 4.1. Structure d'un programme sous VHDL [67].

A .3. Description en VHDL de l'architecture

Pour implémenter une spécification HDL sur un FPGA, avec la plateforme ISE12.2, quatre étapes importantes (expliquées dans la partie suivante) sont illustrées par la Figure 4.2.

Ces étapes sont communes à tous les fabricants de circuits logiques programmables. Chaque fabricant établit un schéma en fonction de ses produits proposés et des logiciels offerts pour pouvoir implémenter une spécification HDL [68].

La démarche que nous avons suivie pour décrire et vérifier le circuit est résumé comme suit :

- Analyse et décomposition de l'architecture proposée.
- Description en VHDL et validation de chaque bloc.
- Rassembler les blocs élémentaires du système.

Nous avons modélisé chaque composant séparément et nous avons stocké les modèles dans des bibliothèques de composants réutilisables afin d'économiser le temps de développement. Des modules supplémentaires peuvent être ajoutés si nécessaire car le langage de description matérielle VHDL facilite la modification et la réutilisation d'un design.

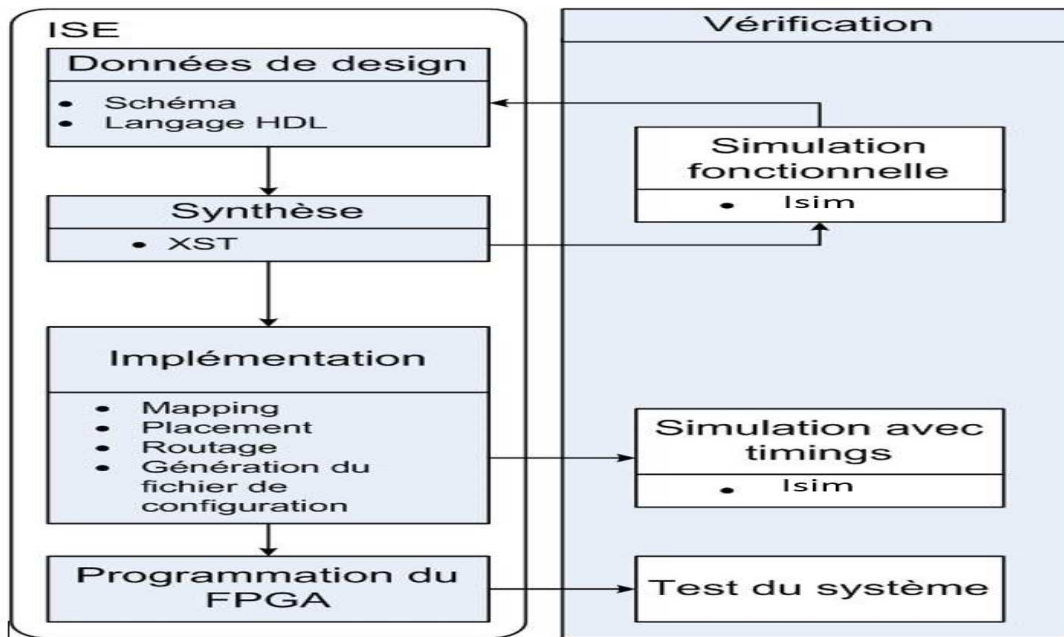


Figure 4.2. Les étapes d'implémentation d'un circuit sur FPGA sous l'outil ISE.

A .4. Caractéristiques du VHDL

Les avantages et les inconvénients sont multiples, on trouve :

Avantages	Inconvénients
<ul style="list-style-type: none"> ➤ Simulation de systèmes complexes : <ul style="list-style-type: none"> - Macroscopiques ou microscopiques - Modélisation d'ensembles de circuits ➤ Description structurée : <ul style="list-style-type: none"> - Travail en équipe, séparation des taches - Rapidité de conception. ➤ Adaptation aux projets Multi-entreprise : <ul style="list-style-type: none"> - Indépendance vis à vis de la technologie - Sécurité grâce à des modèles compilés. 	<ul style="list-style-type: none"> ➤ Description complexe. ➤ Tout n'est pas synthétisable (analogique). ➤ Prise en compte avec le VHDL-AMS (1997).

Tableau 4.1. Caractéristiques du langage VHDL.

A .5. Le flot de conception

A .5.1. Description du circuit

❖ La description du circuit combine:

- Du code HDL (VHDL, Verilog, SystemC, etc.)
- Des schémas d'interconnexions de composants

- Des diagrammes d'états
- Des fichiers de contraintes
- Famille et type de puce utilisés
- fréquence d'horloge cible
- Emphase sur la latence de calcul ou l'espace utilisé.
- Choix de pattes de la puce pour les ports du circuit.
- ❖ Les schémas et les diagrammes d'états sont convertis en code HDL par un outil.
- ❖ On utilise un environnement de développement Project Navigator de Xilinx.

A .5.2. Vérification fonctionnelle par simulation

- La vérification par simulation consiste à stimuler le modèle du circuit, à observer ses réponses et à en vérifier l'exactitude.

On fait une vérification fonctionnelle du modèle. Si des erreurs sont trouvées, on corrige le modèle.

- Une simulation correcte ne garantit (hélas) pas le fonctionnement le flot des autres étapes.

A .5.3. Synthèse

Un outil appelé synthétiseur prend en entrée le code HDL et des contraintes:

- Famille et type de puce utilisés
- Fréquence d'horloge cible
- Emphase sur la latence de calcul ou l'espace utilisé
- ❖ Il convertit le code en modules disponibles sur la puce utilisée.
- ❖ Le processus de synthèse repose sur la reconnaissance de certains patrons de code. Par exemple, la norme IEEE Std 1076.6-2004 IEEE Standard for VHDL Register Transfer Level (RTL) Synthesis spécifie la portion synthétisable du langage VHDL.

A .5.4. Implémentation

- L'implémentation se fait en deux étapes:
- ❖ Le placement
- ❖ Le routage
- Le placement consiste à choisir des structures spécifiques sur un modèle de la puce et à y fixer des modules produits par la synthèse.
- Le routage consiste à établir des connexions entre les structures.
- Les outils d'implémentation utilisent les contraintes
- Fréquence d'horloge cible

- Choix de pattes de la puce pour les ports du circuit

A .5.5. Vérification par simulation

On peut simuler la description produite par l’outil de synthèse et la description produite par les outils d’implémentation, annoter des délais estimatifs de la puce une fois programmée

A .5.6. Extraction des métriques d’implémentation

On peut extraire les métriques d’implémentation suite à la synthèse ou l’implémentation:

- ❖ Le nombre de blocs logiques et autres ressources utilisées
- ❖ Le délai et la fréquence maximale d’horloge estimés
- ❖ La puissance consommée estimée.

- Si ces métriques ne rencontrent pas les spécifications, il faut modifier la description du circuit.

A .5.7. Programmation et vérification de la puce

- La dernière étape consiste à
 - ❖ Générer un fichier de configuration pour la puce utilisée
 - ❖ Programmer la puce en la chargeant avec le fichier de configuration
 - ❖ Vérifier le fonctionnement de la puce programmée.

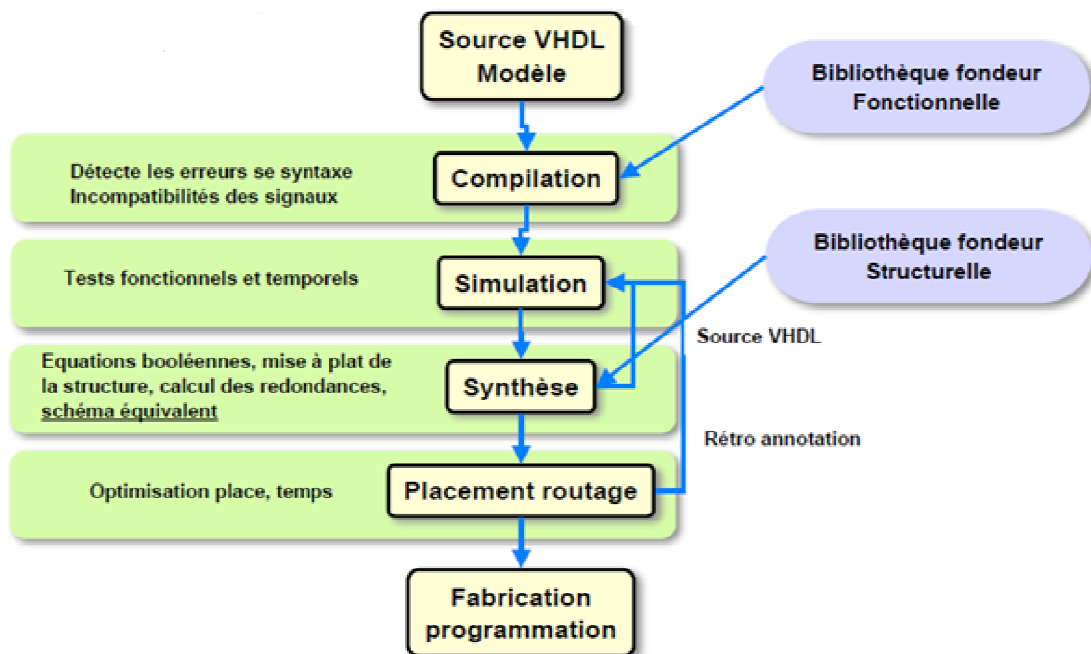


Figure 4.3. Flot de conception d’un circuit intégré en VHDL.

Puis les descriptions VHDL des circuits sont développées avec l'outil Xilinx ISE 12.2.

A.6. Présentation de l'environnement de développement (Xilinx Integrated Software Environment ,ISE)

Le logiciel ISE est un environnement intégré (de la compagnie Xilinx) du développement de systèmes numériques sur des composants reconfigurables ayant pour but une implémentation matérielle sur FPGA. ISE est disponible sur le site www.xilinx.com sous plusieurs versions (... , 8.1, 9.2, ... 10.1, 12.2, 12.3 ...)

Le logiciel ISE de XILINX (Integreted Software Environment) est un logiciel multitâche qui possède dans son soft, différents outils permettant la création de systèmes sous circuits numériques. L'introduction de projets se fait de deux manières qui sont textuelle ou graphique en vue d'une intégration dans un circuit logique programmable (CPLD ou FPGA), sachant que la saisie graphique est une alternative à la saisie textuelle mais limitée. Ce logiciel Xilinx ISE permet la simulation de la description et la synthèse du circuit logique équivalent puis placer et router ce circuit sur un prototype correspondant à une technologie FPGA bien précise et enfin, lorsque toute les vérifications sont faites, vient l'implantation sur un FPGA réel. Ceci correspond à générer le fichier de configuration du circuit cible choisi afin d'établir les interconnexions des cellules logiques correspondantes au circuit logique conçu avec optimisation de ressources disponibles au niveau circuit programmable FPGA. D'une manière générale, le XILINX ISE permet de réaliser toutes les étapes de conception et de programmation des FPGA de XILINX et même pour d'autres circuits programmables tels que les CPLD.

La conception de circuits sur XILINX ISE met en œuvre quatre outils : un éditeur de texte ou entrée graphique, un simulateur, un synthétiseur et un placeur-routeur. L'éditeur de texte ou entrée graphique est pour faire introduire la description dans les logiciels CAO c'est-à-dire de dessiner ou décrire le circuit avec une interface graphique ou textuelle. La simulation du système est faite pour vérifier la validité du code avant-synthèse, après-synthèse et même après le placement et routage. Les deux étapes synthèse et routage succéderons par la suite où la synthèse consiste à faire la transcription de description d'une forme texte vers une autre forme graphique (RTL) à base de portes logiques et pour la deuxième étape nommée routage n'est qu'une adaptation du circuit logique synthétisé sur les ressources disponibles dans le circuit FPGA ciblé [69,70].

Ensuite nous utilisons le circuit FPGA pour l'implantation des différentes architectures des modulateurs numériques.

A .7. Les circuits logiques programmables FPGAs

A .7.1. Introduction

Un circuit logique programmable, ou réseau logique programmable, est un circuit logique intégré qui peut être reprogrammé après sa fabrication. Il est composé de nombreuses cellules logiques élémentaires librement assemblables. Un circuit logique programmable permet la réalisation de circuits logiques à grand nombre de variables d'entrée et de sortie, selon la spécification de l'utilisateur et en utilisant un même support matériel [71].

Les premiers circuits programmables ont été les PROM (Programmable Read Only Memory). Dans ces circuits, le bus d'adresses est utilisé comme entrée et le bus de données comme sortie du circuit logique. Les limitations de ce type de composants pour la réalisation d'architectures logiques ont vite trouvé leurs limites et nous avons vu apparaître les FPLA (Field Programmable Logic Array) [72].

Actuellement, on trouve différentes familles de circuits programmables tels que les PLD (Programmable Logic Device), les CPLDs (Complex Logic Programmable Device) et les FPGAs. La différence entre ces composants est structurelle. Les CPLDs sont des composants pour la plupart reprogrammables électriquement ou à fusibles, peu chers et très rapides (fréquence de fonctionnement élevée) mais avec une capacité fonctionnelle moindre que les FPGA. La figure suivante montre les différentes familles des circuits programmables :

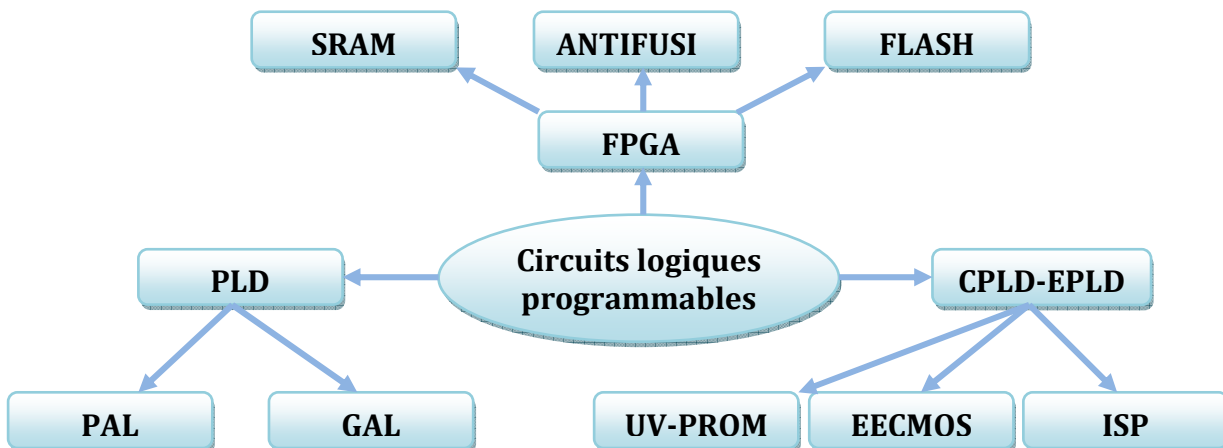


Figure 4.4. Les différents circuits programmables.

Les FPGAs (field programmables gate array) font partie de la famille de composants programmables électroniquement. A l'origine basé sur un réseau de matrice de porte élémentaire ET/OU programmables Array logic (PAL), ces circuits programmables sont devenues au milieu des années 90 des circuits plus complexes grâce à l'intégration de ressources spécifiques, associées de la mémoire interne et des entrées/sorties flexibles.

Les nouvelles générations de circuits FPGA fournissent sur une même puce l'équivalent de plusieurs centaines de milliers de portes logiques, ainsi qu'un certain nombre de bancs de mémoires vives. Le tableau 4.2 présente les avantages et inconvénients respectifs des architectures les plus couramment utilisées pour le traitement numérique du signal [73].

	FPGA	DSP
Avantages	<ul style="list-style-type: none"> - programmation plus flexible. - bus de données plus flexible. -fréquence d'horloge plus flexible -Grande vitesse (100 MHz à quelques GHz). - destiné pour des applications à Hautes performances (plus gourmandes en ressources). -Pour les critères économiques (prix, disponibilité, ...etc.). -multicouches. 	<ul style="list-style-type: none"> -simple à programmer. -moins couteux pour des applications à faible performance. -destiné pour des applications spécifiques (exemples : vidéo).
Inconvénients	<ul style="list-style-type: none"> - plus complexe à programmer. - plus couteux pour des applications à faible performance. - Très haute densité. - Très grand nombre d'entrées/sorties (boîtiers BGA). - Prix élevé (mais en baisse). -Alimentation difficile (plusieurs tensions, courants élevés, connexions multiples). ➤ Volatiles (cellules RAM). Circuits imprimés (PCB - Printed Circuit Board) 	<ul style="list-style-type: none"> -Architecture non flexible. -Espace mémoire faible. -Largeur de bus de données fixe. -Fréquence d'horloge fixe.

Tableau 4.2. Présentation des avantages et inconvénients des circuits FPGA et DSP.

Les circuits FPGAs ont été développés récemment grâce aux progrès de la technologie VLSI .L'apparition de ce type de circuit est une révolution du système digital et ouvrant des

perspectives de traitement numérique inaccessible auparavant. La fin des années 80 a vu l'apparition des premiers circuits FPGA qui sont des circuits intégrés que l'on peut configurer en un temps relativement court pour réaliser n'importe quelle fonction logique « câblée » à bas coût par une programmation de ses cellules logiques et ses interconnexions avec une restriction de ne pas épuiser les ressources de FPGA. Typiquement, un circuit FPGA haute densité peut contenir jusqu'à plusieurs millions d'éléments programmables. Pour réussir une application à base d'FPGA et afin d'obtenir un système plus performant, consommant un minimum de puissance, il est nécessaire de respecter un certain nombre de règles comme :

- Bien connaître les caractéristiques du FPGA ciblé pour assurer son adéquation avec les besoins du projet.
- Elaborer une méthodologie de conception.
- Maîtriser les outils d'implémentation et de choisir des outils de synthèse de qualité. La conception sur les circuits FPGA est un challenge dans lequel l'objectif est de trouver le bon compromis entre densité, flexibilité et performances temporelles.

A .7.2. Les deux grandes familles architecturales de FPGA

Les familles des FPGA peuvent se regrouper en deux groupes :

- ✓ Les circuits FPGA à base de « LUT » (Look Up Tables)

Les LUT ressemblent aux tables de vérité des fonctions logiques et réalisables par des mémoires de type SRAM qui permettent de mettre en œuvre le concept de prototypage (ou maquette) pour la vérification fonctionnelle de systèmes sur puce pour certaines applications. La fonction de la LUT est de stocker la table de vérité de la fonction combinatoire à implémenter comme le montre la figure suivante :

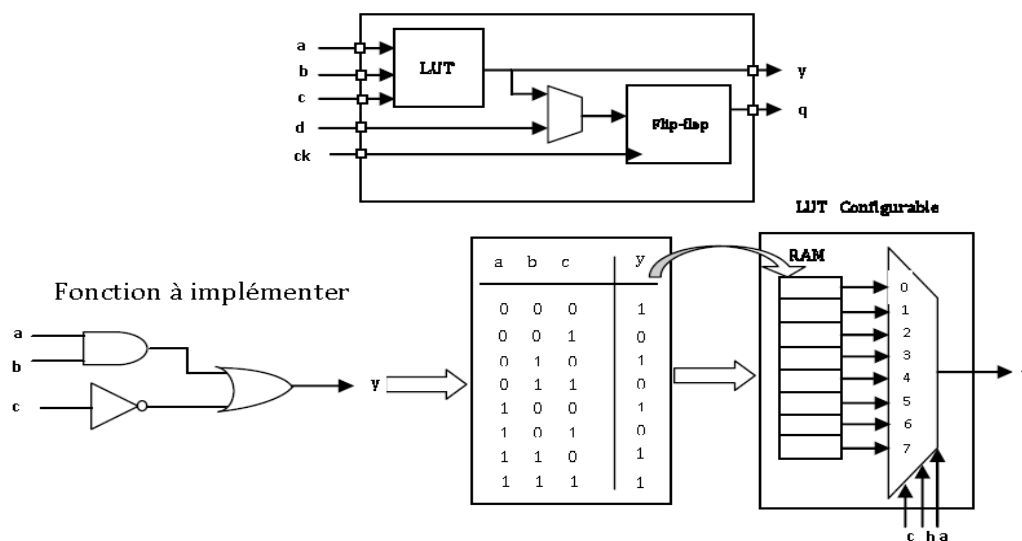


Figure 4.5. Exemple d'implémentation sur LUT [52].

✓ Les circuits FPGA à base de multiplexeurs « MUX »

Les FPGA à base de multiplexeurs sont des microcellules à trois entrées capables de réaliser la fonction suivante :

Equation logique	Symbole
$F=(a \text{ AND } b) \text{ OR } (\bar{a} \text{ AND } b)$	

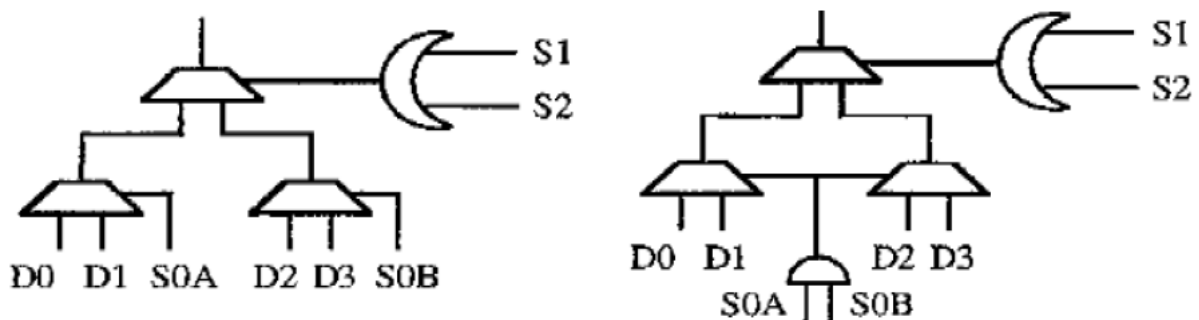


Figure 4.6.Exemple d'implémentation sur des multiplexeurs.

A .7.3. Critères de choix du circuit programmable FPGA

Les FPGA sont développés récemment grâce aux progrès de la technologie VLSI [74]. L'apparition de ce type de circuit est une révolution des systèmes digitaux et ouvrant de nouvelles perspectives de traitement numérique inaccessibles auparavant. La fin des années 80 a vu l'apparition des premiers circuits FPGA qui sont des circuits intégrés que l'on peut configurer en un temps relativement court pour réaliser n'importe quelle fonction logique « câblée » à bas coût par une programmation de ses cellules logiques et ses interconnexions avec une restriction de ne pas épuiser les ressources du FPGA [75]. Typiquement, un circuit FPGA haute densité peut contenir jusqu'à plusieurs millions d'éléments programmables. Pour réussir une application à base de FPGA et afin d'obtenir un système plus performant, consommant un minimum de puissance, il est nécessaire de respecter un certain nombre de règles comme :

- Bien connaître les caractéristiques du FPGA ciblé pour assurer son adéquation avec les besoins du projet.
- Elaborer une méthodologie de conception.
- Maîtriser les outils d'implémentation et de choisir des outils de synthèse de qualité.

La conception sur les circuits FPGA est un challenge dans lequel l'objectif est de trouver le bon compromis entre densité, flexibilité et performance temporelles [76].

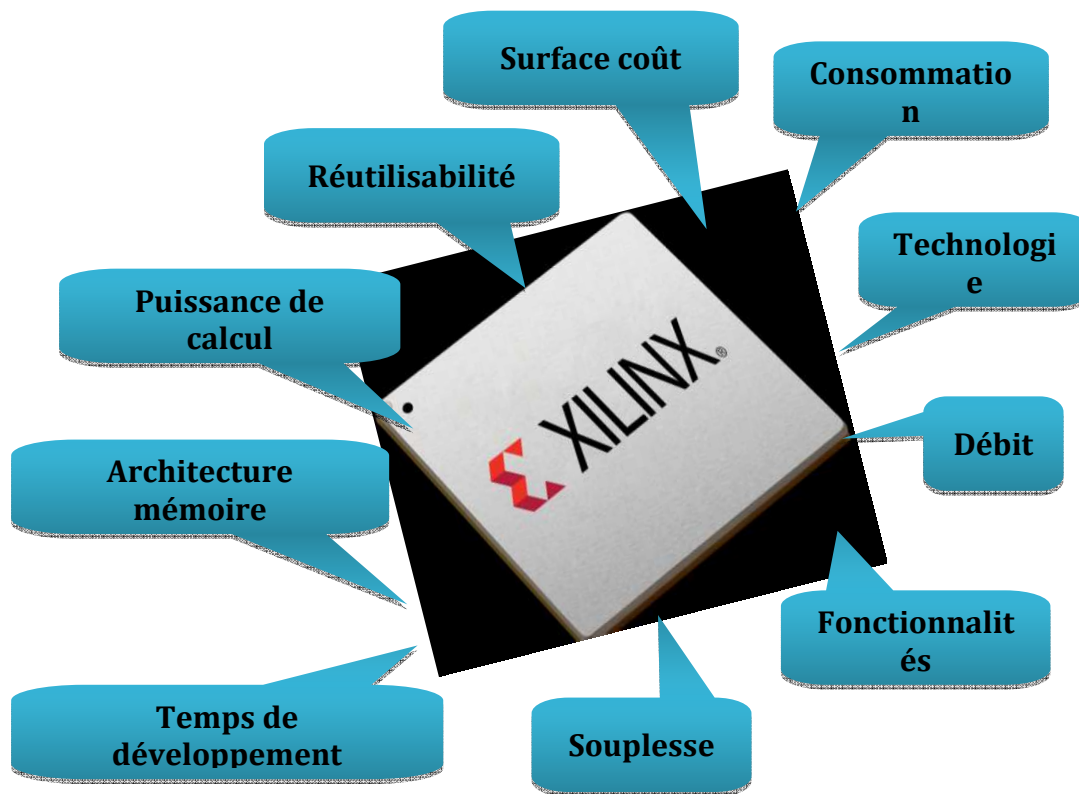


Figure 4.7. Critères de choix du circuit logique programmable FPGA.

A .7.4. Evolution des FPGAs XILINX famille Virtex

Famille	Année	Technologie	Max cellules
Virtex4	2004	90nm	200000LUT4
Virtex5	2007	65nm	330000LUT6

Tableau 4.3. Evolution des FPGAs XILINX famille Virtex.

A .7.5. Etude de la famille VIRTEX 5 de type (XC5LX110T)

La famille Virtex 5 fournit les nouvelles fonctionnalités les plus puissantes du marché des FPGA. En utilisant la deuxième génération ASMBL (Advanced Silicon modular Block), l'architecture en colonnes, la famille Virtex-5 contient cinq plates-formes distinctes (sous-familles), le plus grand choix offert par une famille de FPGA.

Chaque plate-forme contient une proposition différente de fonctionnalités pour répondre aux besoins d'une grande variété de logique avancée, dessins et modèles. En plus de la plus avancée, la logique de tissu haute performance. FPGA Virtex-5 contient beaucoup de dur-IP blocs au niveau du système, y compris la RAM bloc puissant de 36 Kbit/FIFO, la deuxième génération 25 x18 tranches DSP, sélectionnent IO avec la technologie intégrée dans digital

contrôle impédance ,chip sync source-synchrone interfacent des blocs ; la fonctionnalité de surveillance du système, améliore les carreaux de gestion d'horloge intégrant DCM (Digital Clock Manager) et à verrouillage de phase en boucle de phase (PLL) des générateurs d'horloge, et les options de configuration avancées.

Le tableau 4.4 montre les détails des 2 circuits FPGA (Virtex5-LX50 et Virtex5-LX110T)

circuits FPGA	CLB (Configurable Logic Block)			DSP 48E	Blocs BRAM			Total I/O Bank s	Max User I/O
	(Row x col)	Slices	Max RAM distr		18 Kb	36 Kb	Max(Kb)		
XC5LX 50	120x30	7200	480Kb	48	96	48	1.728	17	560
XC5LX 110T	160x54	17280	1.120Kb	64	296	148	5.328	20	680

Tableau 4.4. Comparaison entre (Virtex5-LX50 et Virtex5-VLX110T).

A .7 .6. Les domaines d'applications des FPGAs

Les FPGA ont fait révolutionner le contrôle numérique et sont de plus en plus utilisés pour intégrer des architectures complexes. Ils sont devenus les plus populaires en matière d'implantation et de prototypage des circuits numériques après leur apparition sur le marché en 1984. La clé maitresse de leurs réussites est l'aspect programmable de ces derniers.

Leurs utilisations actuelles couvrent les deux domaines : domaine civil et militaire. Parmi ces applications nous citons :

- 1-Informatique : Périphériques spécialisés.
- 2-Machinerie industrielle : Contrôleur pour machines.
- 3-Télécommunications : traitement d'image, filtrage.
- 4-Instrumentation : équipement médical, prototypage.
- 5- Transport : Contrôle d'avions et métros.
- 6- Aérospatiale: Satellites, Radar, Communication protégée, la détection ou la surveillance.....etc.

A .7.7. Les technologies de FPGA

Le choix de la technologie de programmation affecte plusieurs caractéristiques du circuit programmable :

- Densité
- Délais de propagation (Cellules, Routage)
- La consommation
- Reprogrammation du circuit On distingue deux types de circuits programmables :
- Circuit programmables une seule fois OTP pour « One Time Programmable ».
- Circuit programmable plusieurs fois.

A .8. Technologie EPROM

EPROM (Erasable Programmable Read Only Memory), cette technologie utilisée dans grille flottante pour programmer les transistors de transmission.

Il y'a plusieurs avantages et inconvénients qui sont :

- Circuits programmables plusieurs fois
- Programmation sur site
- Consommation statique
- Nécessite une grande surface
- Plusieurs tensions d'alimentation
- Déprogrammation lente.
- Boîtier d'encapsulation spécial (avec du verre)

A.8.1. Technologie EEPROM

EEPROM (Electrical Erasable Programmable Read Only Memory) est la même méthode de programmation que pour EPROM et utilisée aussi pour la déprogrammation (Utilisation de champs électriques).

Les Avantages sont multiples, on trouve :

- Déprogrammation plus rapide que pour EPROM
- Pas besoin d'utiliser un boîtier spécial (avec du verre)
- Déprogrammation sur site.

A .8.2 . Technologie antifusible

- A l'état non passant il s'agit d'un isolant entre deux semi-conducteurs fortement dopés.
- Le passage à l'état passant se fait par claquage de l'isolant par diffusion du semi-conducteur dans l'isolant. Pour cela, il faut appliquer une forte tension à l'isolant (16V pendant 1ms)
- La configuration du fusible est irréversible (non-reconfigurable)

- La plus ancienne des technologies (initialement cellule fuse mais tension de programmation trop importante) Voici quelques exemples de FPGA utilisant la technologie anti-fuse:

Actel série EX , Actel série SX-A

Les avantages et les inconvénients sont comme ce qui suit :

- Le point de configuration est simple et très petit => haute densité d'intégration.
- La configuration est rapide.
- Technologie la moins chère.
- Très bonnes performances électriques (R et C faible) => faible consommation (Watt).
- Technologie très sûre pour la propriété intellectuelle (espionnage industriel).
- Peu sensible aux radiations (milieux hostiles : espace par exemple).
- Circuit programmable une seule fois.
- Tension élevée pour la configuration.

A .8.3. Technologie Flash

Cette Technologie est une version plus évoluée des EEP-ROM avec avantage d'être plus facile à programmer et à effacer.

Leurs avantages et des inconvénients sont les suivants:

- Densité d'intégration importante (mais 10 fois moins que la technologie antifuse)
- Temps de configuration faible (d'où le nom de "Flash") de quelques μ s pour un temps d'effacement 1000 fois plus grand (qqms)
- Plusieurs cellules sont configurées simultanément
- Mémoire non volatile (principe de la mémoire ROM)
- Disponible à la mise sous tension
- Les cellules élémentaires ne permettent pas une configuration partielle du circuit.
- Nécessite une alimentation supplémentaire (la tension de programmation et d'effacement est de 12 Volts) qui peut être intégrée au composant sous la forme d'un système de pompe de charge. Le composant peut alors être programmé sur la carte (ISP : In Situ Programmable).
- Peu sensible aux radiations.

A.8.4. Technologies SRAM (Static Random Memory)

Mémoires volatiles avec cellule de base à plusieurs transistors (accès rapide, consommation plus, coûteux). La volatilité correspond au non disponibilité de l'information lorsqu'il n'y a pas d'alimentation.

Les avantages et les inconvénients sont multiples, on trouve :

- Configuration très rapide de qqns.

- Nombre de configurations illimité (idéal pour le prototypage).
- Chargement d'une configuration partielle ou totale possible en cours d'utilisation (Run Time Configuration = reconfiguration dynamique).
- Utilisation des dernières technologies (90nm, 9 métallisations cuivre) - Programmation In Situ.
- Grande surface occupée environ deux fois plus grande que celle d'une cellule flash et 20 fois plus que celle d'une cellule à antifusible.
- Mauvaises performances électriques (R et C) élevé=> consommation de puissance élevée (20% au moins que la technologie Flash).
- Mémoire volatile, nécessite donc une programmation à chaque mise sous tension (mémoire de configuration ROM externe)
- L'échange du bitstream est une faille en termes de sécurité (protection du design).
- Grande sensibilité aux radiations.

A.8.5. Architecture interne du FPGA

Chaque fournisseur d'FPGA a sa propre architecture, mais en termes généraux ils sont tous une variante de la structure représentée par la figure 4.6. L'architecture se compose de blocs logiques LAB, blocs E/S, cellules logiques LE, et interconnexion programmable. En outre, il aura un circuit d'horloge pour conduire les signaux d'horloge à chaque bloc logique.

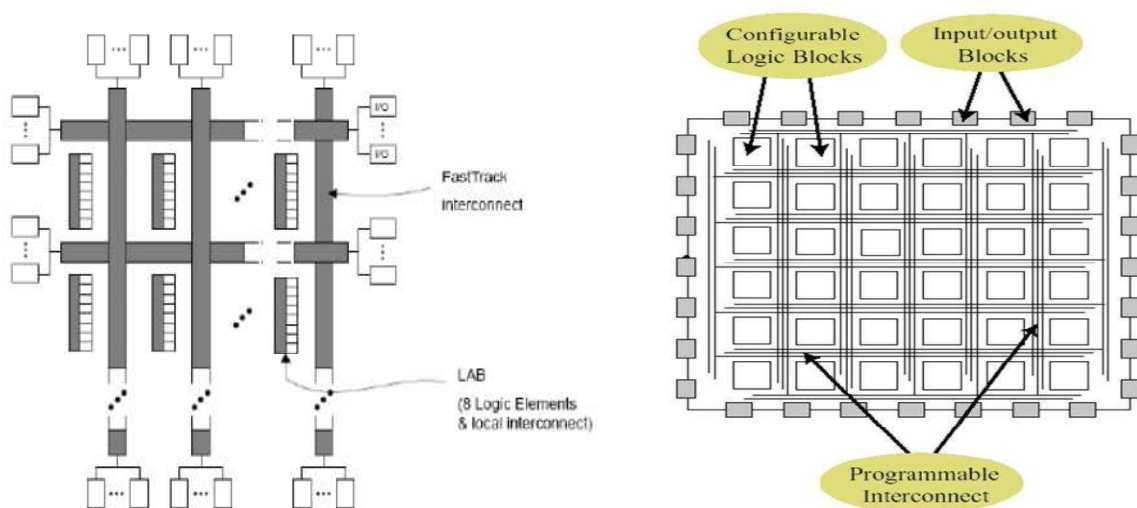


Figure 4.8. Structure générale des FPGAs [77].

A 8.6. Les blocs logiques

Les LAB facilitent la combinaison de plusieurs LE et donc la réalisation de fonctions combinatoires ou séquentielles plus complexes. En profitant des chemins de propagation intégrés, les fonctions implantées dans un seul LAB bénéficient de délais de propagation réduits entre les différents LE intégrés. Mais un LAB ne regroupe qu'un nombre restreint de LE. Les fonctions trop complexes, ou qui nécessitent un trop grand nombre d'entrées/sorties sont donc implantées sur plusieurs LAB distincts, et les délais de propagation entre cellules logiques s'en trouvent dégradés.

Le plus petit élément structurel de cette architecture est l'élément logique décrit dans la section suivante.

Les blocs logiques permettent de réaliser des fonctions qui vont de la simple porte au microprocesseur (Figure 4.9.). Ils sont typiquement capables d'implanter plusieurs fonctions logiques combinatoires ou séquentielles. Ces blocs sont des :

- Paires de transistors
- Porte logiques NAND et XOR à deux entrées
- Des Multiplexeurs
- Des Look_Up Table (LUT)
- Des structures AND_OR

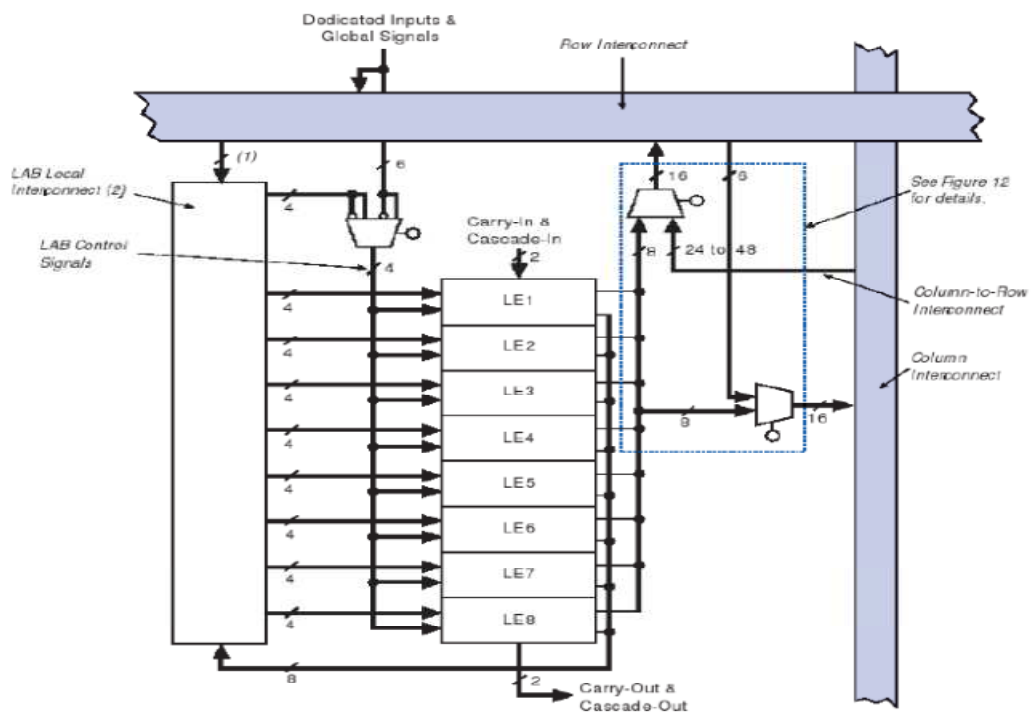


Figure 4.9. Architecture d'un bloc logique.

A.8.7 Architecture d'une cellule logique

Les éléments logiques sont les blocs de base de tout circuit FPGA (Figure 4.10). On peut réaliser dans ces blocs toutes les opérations de logique combinatoire (dans la limite du nombre d'entrées). Ces blocs ont souvent la même constitution et cela malgré la différence de fabricants et d'architectures. La structure la plus courante est la table LUT (Look Up Table) qui contient, après configuration, la table de vérité de la fonction logique qu'elle doit réaliser ou alors un ensemble de valeurs qui sont mémorisées comme dans une mémoire ROM.

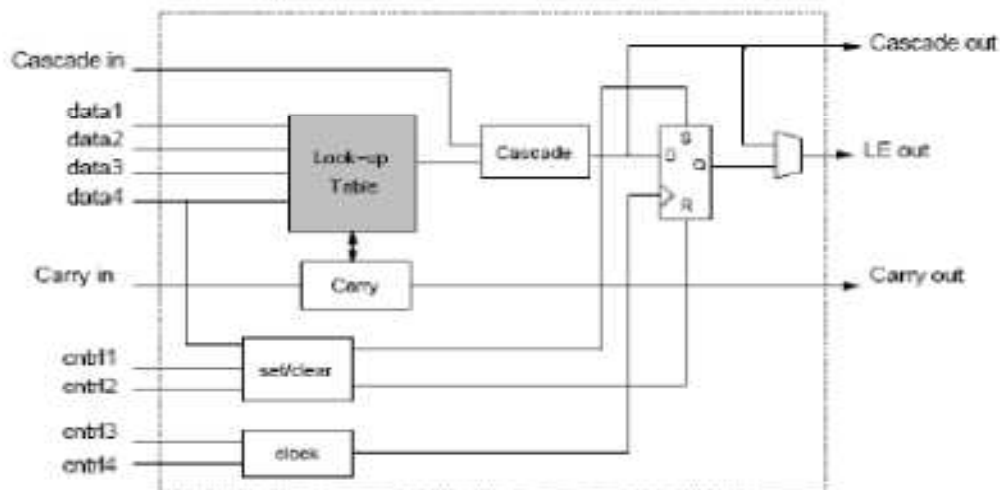


Figure 4.10. Cellule logique d'un FPGA.

A.8.8. Les éléments d'entrées/sorties

Un bloc d'entrée / sortie configurable, illustré à la figure 4.11, est utilisé pour amener les signaux sur la puce et les renvoyer. Il se compose d'un tampon d'entrée et un tampon de sortie avec trois états et ouvrir les commandes de sortie de collecteur.. En outre, il y a souvent un flip-flop sur les résultats de sorte que les signaux de sortie peuvent être cadencés directement aux broches sans rencontrer de retard important.

Le but des éléments d'entrées/sorties est de relier un circuit avec son environnement extérieur. Ceux-ci peuvent bénéficier de protections, de buffer ou d'autres éléments permettant la gestion des entrées et des sorties. En particulier, il est à noter que les circuits actuels proposent différentes normes pour les niveaux d'entrées et de sorties qui par configuration peuvent être choisies afin de s'adapter à l'environnement.

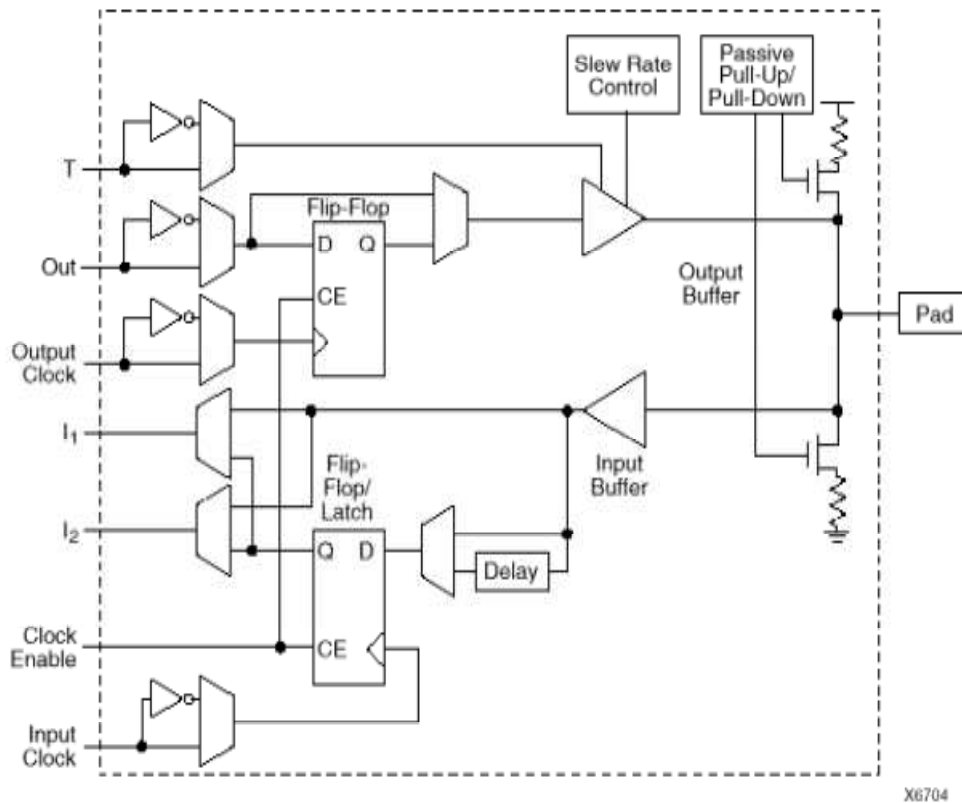


Figure 4.11. Blocs d'entrée/sortie.

A.8.9. Les interconnexions

Ces interconnexions permettent l'établissement de liaisons entre les CLB et les IOB avec un maximum d'efficacité de vitesse et d'occupation du circuit (Figure 4.12). De plus, il est possible de connecter directement certaines entrées d'un CLB aux sorties d'un autre. Pour chaque bloc logique configurable, la sortie X peut être connectée directement aux entrées C ou D du CLB situé au-dessus et les entrées A ou B du CLB situé au-dessous. Quant à la sortie Y, elle peut être connectée à l'entrée B du CLB placé immédiatement à sa droite. Pour chaque bloc logique adjacent à un bloc entrée/sortie, les connexions sont possibles avec les entrées I ou les sorties O suivant leur position sur le circuit.

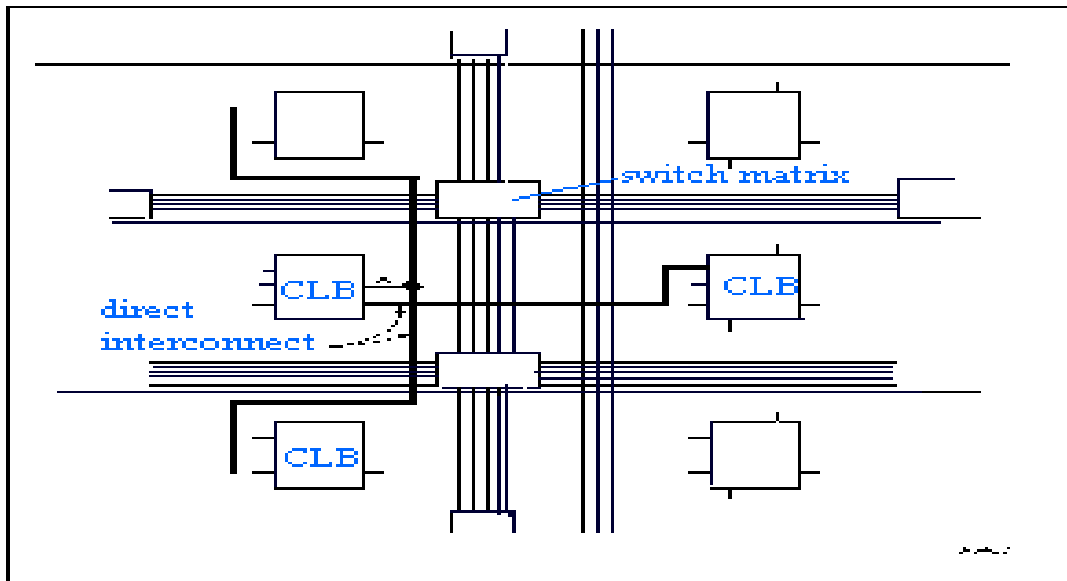


Figure 4.12. Les interconnexions.

4.A.8.10 Circuit d'horloge

Des entée/sortie spatiales de haut tampon d'horloge, comme l'horloge conductrice, qui sont réparties autour de la puce. ces tampons sont connectés à l'horloge d'entrée dans le but de conduire les signaux d'horloge sur les lignes d'horloge globale. ces lignes d'horloge sont conçues spécialement pour des faibles impédances.

A.8.11. Les principaux fabricants de FPGA

Comme un des plus grands segments croissants de l'industrie du semi-conducteur, le marché de FPGA est volatile. Comme tel, l'équipe de compagnies impliquée change rapidement et c'est difficile de dire quels produits seront les plus considérables quand l'industrie atteint un état stable.

Il y a deux catégories de base de FPGAs sur le marché aujourd'hui: FPGA basé sur SRAM et FPGA antifusible (Figure 4.13).

Dans la première catégorie, Xilinx et Altera sont les principaux fabricants quant au nombre d'utilisateurs avec le concurrent majeur qui est AT&T (Figure 4.11). Pour les FPGA antifusible Actel, Quicklogic, Cypress et Xilinx offrent des produits en concurrence.



Figure 4.13. Les deux grandes familles de circuit FPGA.

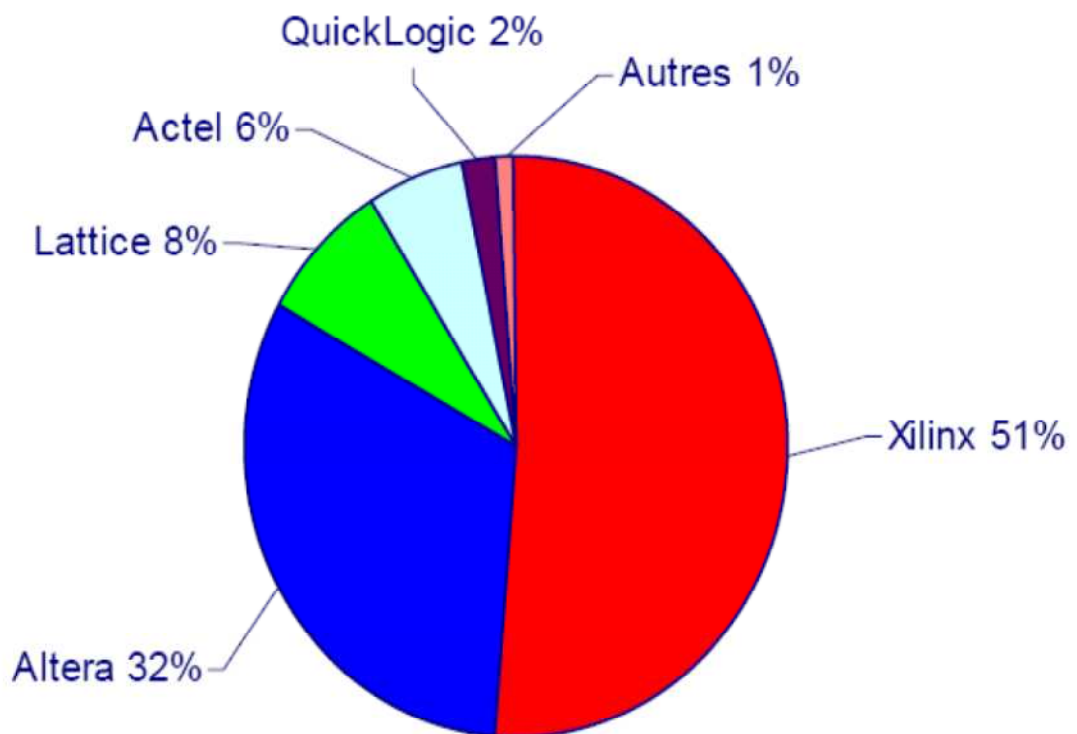


Figure 4.14. Statistiques des marchés occupés par les vendeurs des FPGAs [78].

A.8.12. Exemple du premier fabricant XILINX

Le plus gros constructeur du marché est XILINX qui a introduit la série XC2000 (de 600 à 1500 portes) entre 1984 et 1985 avec des fréquences de fonctionnement pouvant atteindre les 420MHZ (Figure 4.14). Depuis, d'autres séries sont apparues comme les XC3000 ,XC4000,XC5200 et XC6200,ainsi que les XC9500 et la mise sur le marché du 1^{er} FPGA XILINX en 1985.puis l'apparition en 1992 du premier FPGA du constructeur ALTERA qui est le concurrent le plus important de XILINX , avec un type de circuit assez différent le FLEX 8000 (1500 portes max).L'exploitation de la technologie EEPROM en 1993 puis le lancement du VIRTEX II/XILINX présente leur produit avec jusqu'à 10millions de portes en 2001 et rapidement des FPGA présente le leur avec des capacités supérieures à 50 millions de portes fonctionnant à des fréquences dépassant les 500MHZ.

Le principe des FPGA de XILINX est de stocker la configuration dans une mémoire vive statique SRAM. Aujourd'hui des blocs de fonctionnalités supplémentaires dans quelques versions évoluées sont ajoutés et dédiés à des applications spécifiques.Ces fonctionnalités supplémentaires qui sont : mémoire RAM, petits multiplieurs, blocs DSP, cœurs de processeurs RISC et arbres de distribution d'horloge pour générer différents domaines d'horloge pour un bon synchronisme en cas de FPGA a grande capacité.

Finalement nous utilisons le logiciel MATLAB pour afficher les paramètres de fréquence d'échantillonnage et nombre de cellules à chaque architecture.

A.9. le logiciel MATLAB

MATLAB (« matrix laboratory ») est un langage de programmation de quatrième génération et un environnement de développement .Il est utilisé à des fins de calcul numérique. Développé par la société The MathWorks, ce langage permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++ et Java. Les utilisateurs de matlab (environ 2 millions en 2019) sont de milieux très différents comme l'ingénierie, les sciences et l'économie dans un contexte aussi bien industriel que pour la recherche. Il interprète (donc sans phase préliminaire de compilation) puis exécute les opérations demandées séquentiellement, avec possibilité de boucle, test et saut. Il ne manipule que des données numériques et ne sait effectuer aucun calcul formel à priori. Matlab peut s'utiliser seul ou bien avec des toolbox (boîte à outils) et pour développer des solutions nécessitant une très grande puissance de calcul. Son objectif, par rapport aux autres langages, est de simplifier au maximum la transcription en langage informatique d'un problème

mathématique, en utilisant une écriture la plus proche possible du langage naturel scientifique. Le logiciel fonctionne sous Windows et sous Linux.

Finalement, il y'a plusieurs paramètres traités dans cette étude qui sont :

Echantillonnage des signaux (fréquence d'échantillonnage), le nombre de cellules, le débit, la consommation de puissance et le simulation des signaux (test bench)

A.9.1. Echantillonnage des signaux

A.9.1.1 Introduction

L'échantillonnage est l'opération qui permet de représenter de façon discrète un signal continu dans le temps (Figure 4.15). Le signal échantillonné correspond aux valeurs du signal original, prélevées à des intervalles de temps fixes, multiples d'une durée donnée et qui est la période d'échantillonnage T_e (Figure 4.16)..

Soit un signal continu :

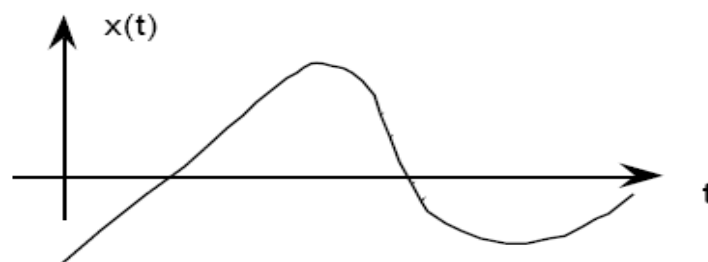


Figure 4.15.Signal continu.

Un échantillonnage consiste à représenter un signal par une suite de valeurs ponctuelles :

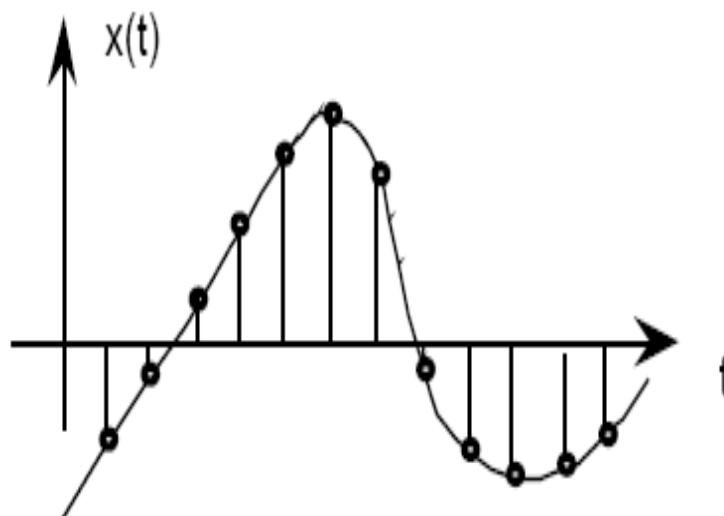


Figure 4.16.Signal échantillonné.

A.9.1.2. Exemple d'un signal échantillonné obtenu par Matlab

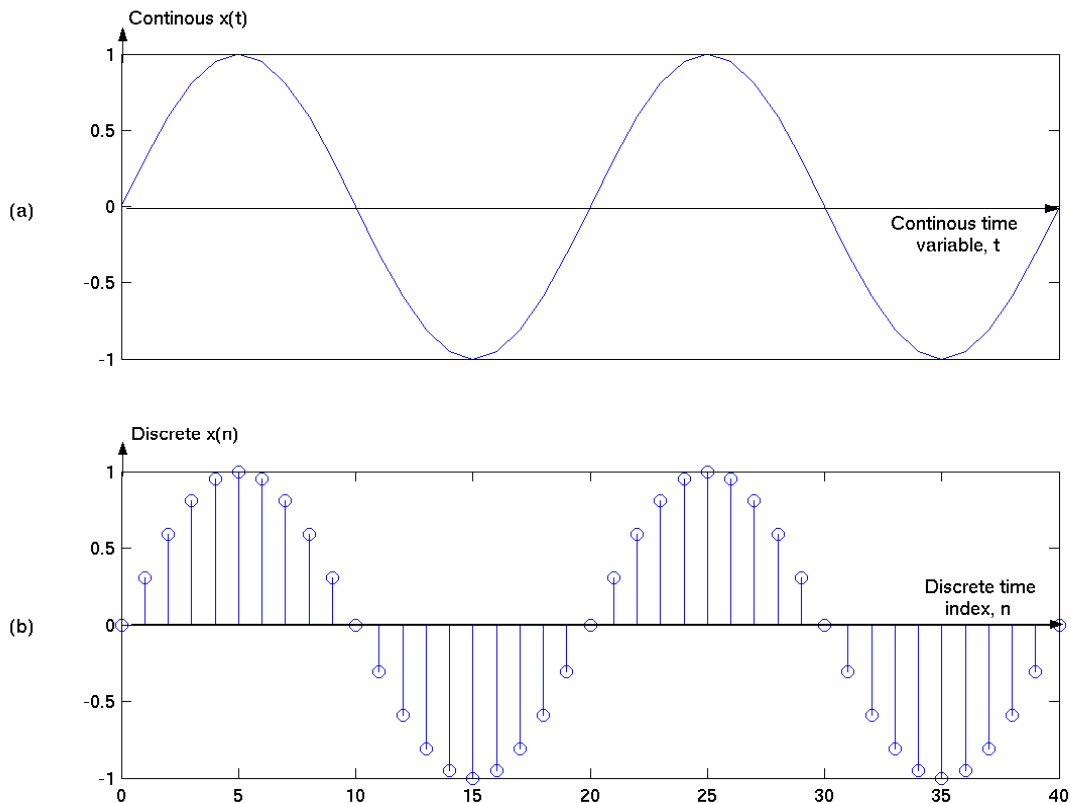


Figure 4.17. Résultat obtenu sous Matlab de l'échantillonnage d'un signal continu.

A.9.1.3. Le théorème d'échantillonnage

Ce théorème présente les conditions dans lesquelles un signal continu peut être complètement représenté par une suite discrète de ses échantillons [79].

En d'autres termes, un signal est complètement représenté si à partir de ses échantillons prélevés à intervalles T_e , il est possible de reconstruire l'intégralité du signal initial.

Ce théorème s'exprime de la façon suivante : un signal dont le spectre ne contient aucune composante de fréquence supérieure ou égale à une fréquence maximale f_{\max} peut être entièrement représenté par une suite de valeurs constituées par ses échantillons, à condition que ceux-ci soient prélevés à des intervalles régulièrement espacés, dont la durée T_e n'excède pas

$$T_e = \frac{2}{f_{\max}} \quad (4.1)$$

On peut alors définir la fréquence d'échantillonnage par :

$$f_e = \frac{1}{T_e} \quad (4.2)$$

La condition de Shannon est satisfaite dans les signaux représentés dans la figure suivante.

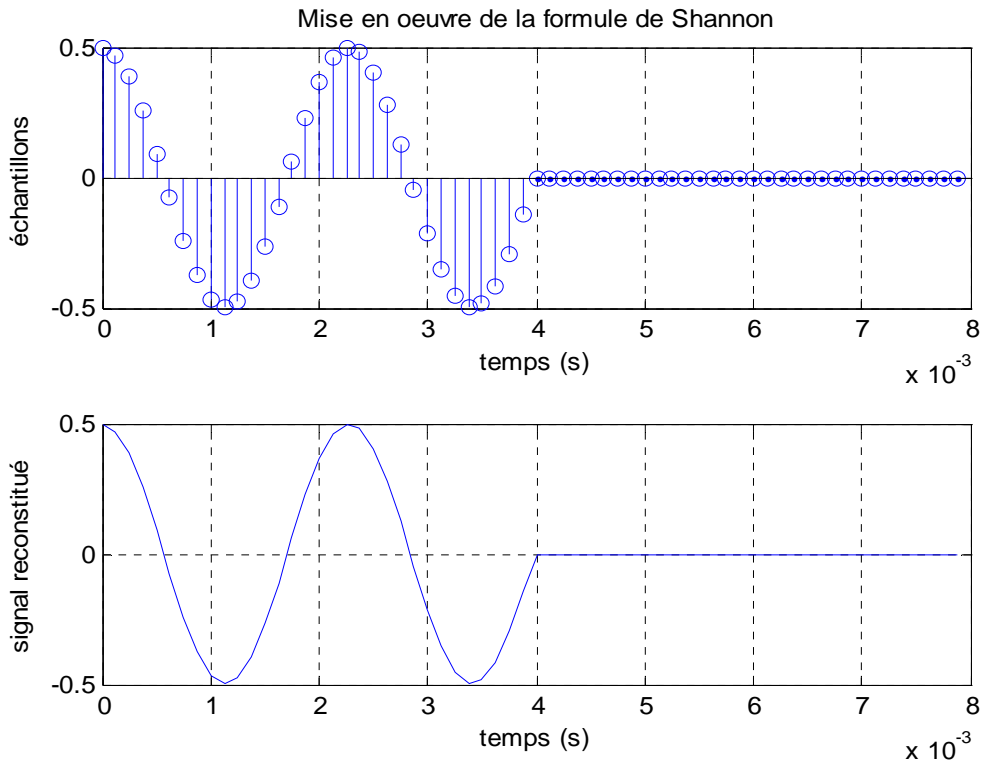


Figure 4.18 : Représentation de la condition d'échantillonnage (Shannon).

A.10. Simulation (Les TEST-BENCHS)

La plupart des simulateurs logiques permettent de créer des stimuli pour vérifier le comportement d'un modèle. Souvent un langage de commande donne à l'utilisateur la possibilité d'automatiser, par la création de fichiers contenant des instructions de ce langage, l'enchaînement des opérations nécessaires aux tests. Le langage VHDL est un langage de modélisation et de simulation. Il contient tous les éléments nécessaires à la création de stimuli et surtout à l'exploitation des résultats.

Pour élaborer un TEST BENCH, il faut établir la liste des cas le plus complètement possible. Les stimuli sont les entrées appliquées au programme via le simulateur pour imiter le comportement des vraies entrées. Après exécution du simulateur, les sorties sont observées si elles sont comme prévues par le programme et ceci sans aucun risque sur le matériel FPGA [80].

A.11. Consommation de puissance

La consommation de puissance est une spécification cruciale dans les systèmes

numériques modernes, en particulier pour les appareils portables. Elle est classée comme statique ou dynamique, où le premier qualificatif est relatif à la puissance consommée du circuit à l'état statique, le deuxième désigne la puissance consommée lorsque le circuit change d'état.

La consommation de puissance totale (P_T) est donc donnée par:

$$P_T = P_{\text{static}} + P_{\text{dynamique}} \quad (4.3)$$

3. Résultats de simulation et implémentation : PARTIE B

B.1.Introduction

L'architecture proposée dans ce travail, consiste à assurer un débit maximum de données binaires, avec un taux d'erreur acceptable par les protocoles OSI ; par exemple en se basant sur l'algorithme CORDIC amélioré pour l'implémenter sur carte FPGA (XC5VLX110T).

L'implantation de notre architecture sur FPGA concerne la programmation du circuit à l'aide du langage de description de haut niveau VHDL. Cette programmation est donc la première étape. La deuxième étape consiste à effectuer la simulation qui est validée à l'aide de logiciel ISE 12.2 de la société Xilinx . Ce logiciel fournit, à la fin de ces phases, divers résultats sous forme de rapports et schémas. Parmi ces documents, le rapport de compilation (tableau 4.4) nous donne les quantités de ressources logiques utilisées pour l'implémentation physique du système, tandis que la fonction RTL viewer (Register Transfert Level) présente une description hiérarchique.

B .2. Résultats de l'implantation de trois modulateurs QAM sur le circuit FPGA

Dans cette partie, nous allons présenter les résultats de l'implantation de la synthèse et la simulation de l'architecture numérique générique pour les trois modulateurs QAM₁, QAM₂ et QAM₃.

La troisième architecture donne le meilleur résultat, car elle utilise un nombre minimal de cellules ainsi qu'une fréquence de traitement maximale. Ces derniers paramètres permettront d'avoir un modulateur plus rapide que ceux obtenus par les autres architectures. Dans cette partie nous prenons en compte les contraintes suivantes :

$$\left\{ \begin{array}{l} \text{fréquence}_{\text{QAM3}} > \text{fréquence}_{\text{QAM2}} > \text{fréquence}_{\text{QAM1}} \\ \text{cellules}_{\text{QAM3}} < \text{cellules}_{\text{QAM2}} < \text{cellules}_{\text{QAM1}} \\ \text{débit}_{\text{QAM3}} > \text{débit}_{\text{QAM2}} > \text{débit}_{\text{QAM1}} \end{array} \right. \quad (4.4)$$

B.2.1. L'implémentation d'un modulateur numérique QAM₁

Cette architecture est composée des éléments suivants : un codeur, un accumulateur de phase et un générateur de sinusoïde à base de CORDIC itératif. Son algorithme permet d'obtenir un seul échantillon en 8 cycles d'horloge et le nombre de cellules nécessaires du circuit FPGA a ainsi augmenté. De plus, nous cherchons à obtenir la plus grande fréquence d'échantillonnage possible, et cette architecture nécessite plusieurs cycles d'horloge pour produire un seul échantillon, ce qui n'est généralement pas satisfaisant.

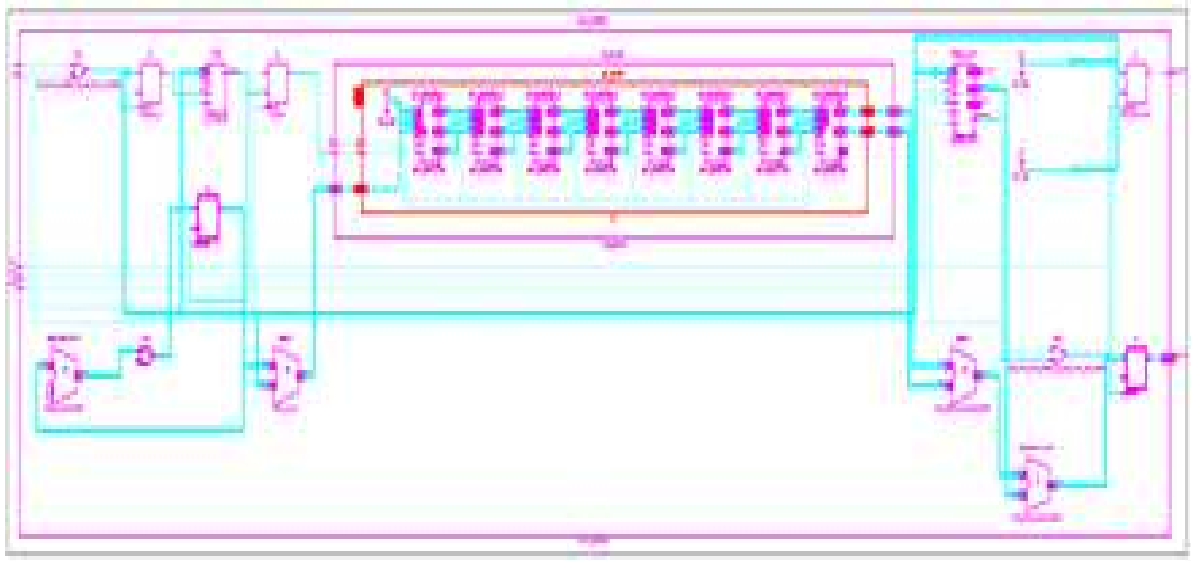


Figure 4.19 : Vue interne de l'implémentation d'un modulateur QAM₁

B.2.2. Le codeur

Résultat de synthèse

L'implémentation RTL du codeur est présentée dans les figures (4.19) et (4.20) :

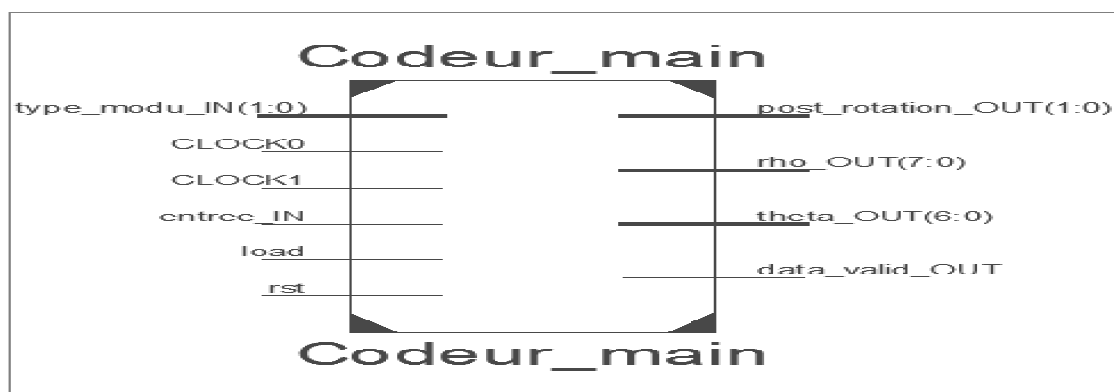


Figure 4.20. Vue externe (Schémas RTL) de l'implémentation sur FPGA Virtex 5 du codeur.

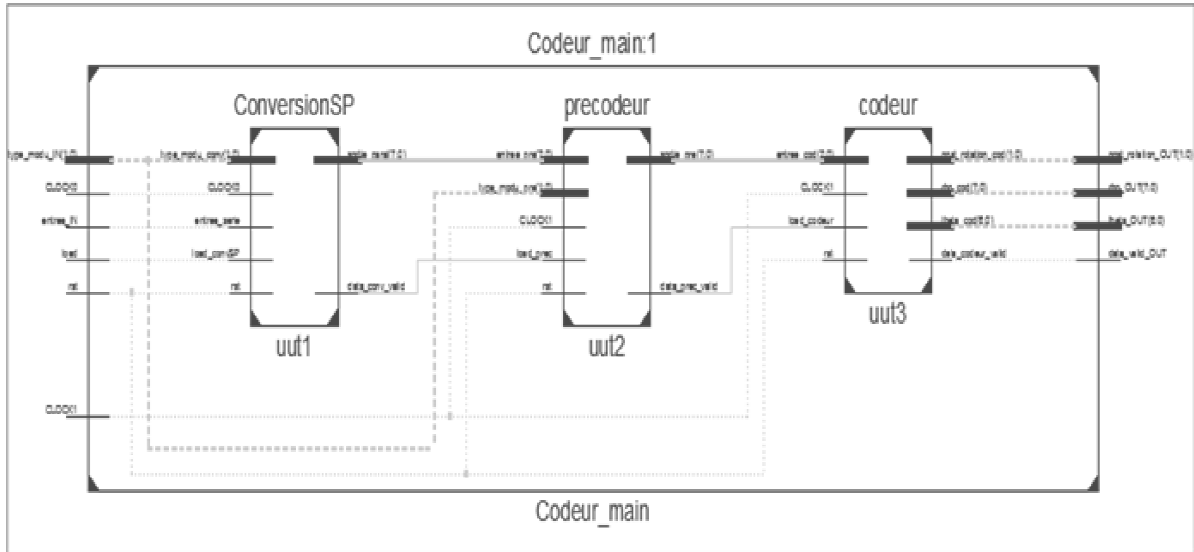


Figure 4.21. Vue interne de l'implémentation du codeur.

B.2.3. L'accumulateur de phase

Résultat de synthèse

La figure ci-dessous présente l'implémentation RTL d'accumulateur de phase :

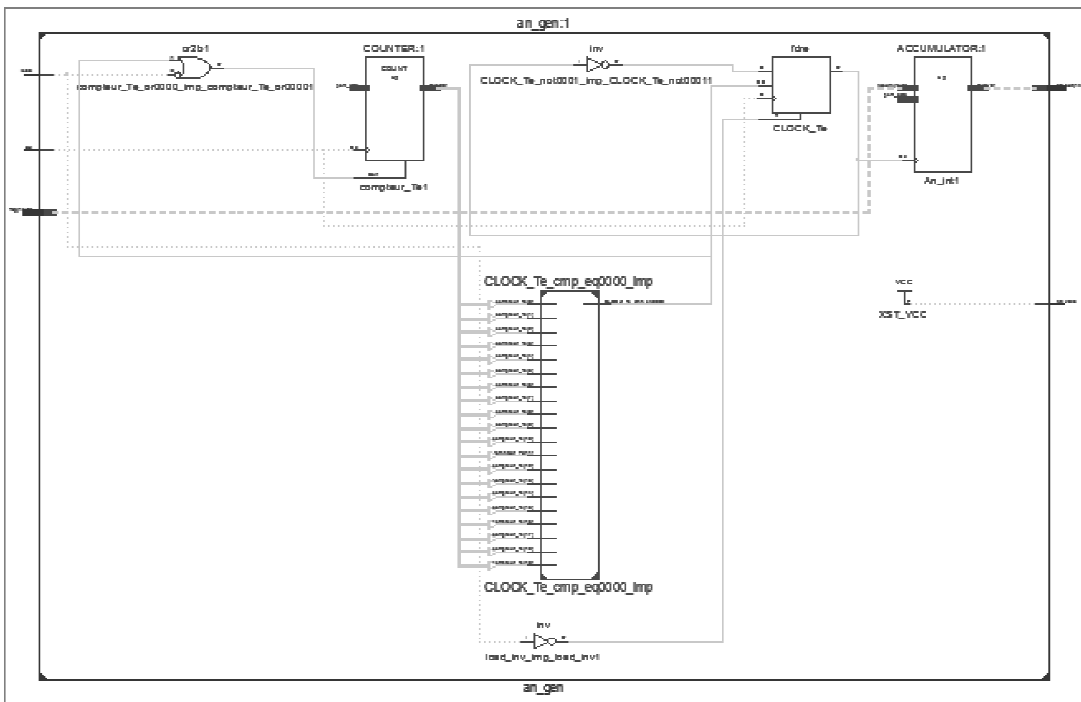


Figure 4.22. Vue interne de l'implémentation de l'accumulateur de phase.

B.2.4. Générateur de sinusoïde basé sur CORDIC itératif à 8 étages

Résultat de synthèse

L'implémentation RTL est illustrée dans la figure (4.22) :

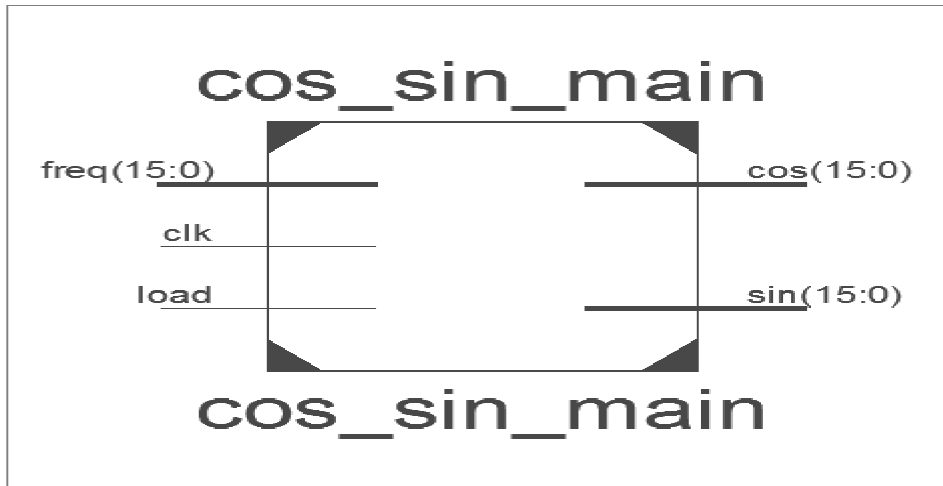


Figure 4.23. Vue externe (schémas RTL) de l'implémentation du CORDIC itératif à 8 étages.

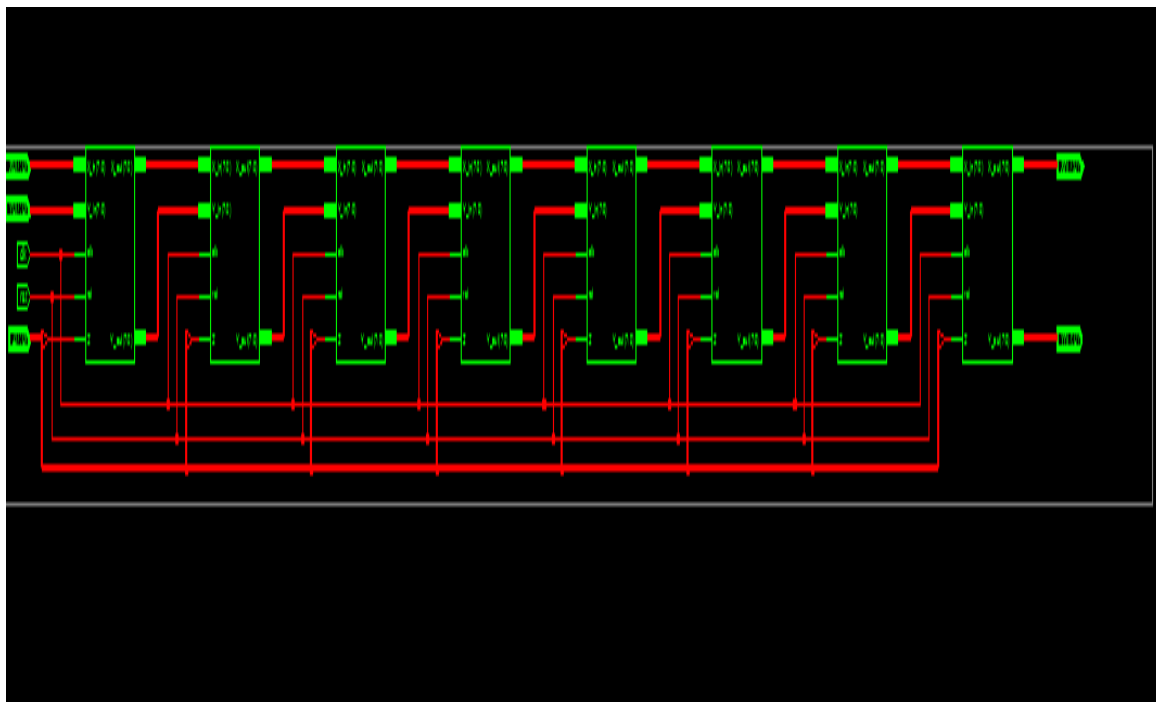


Figure 4.24. Vue interne de l'implémentation du CORDIC itératif à 8 itérations.

Simulation

Sur la figure (4.24), nous présentons le résultat de simulation affiché par le logiciel

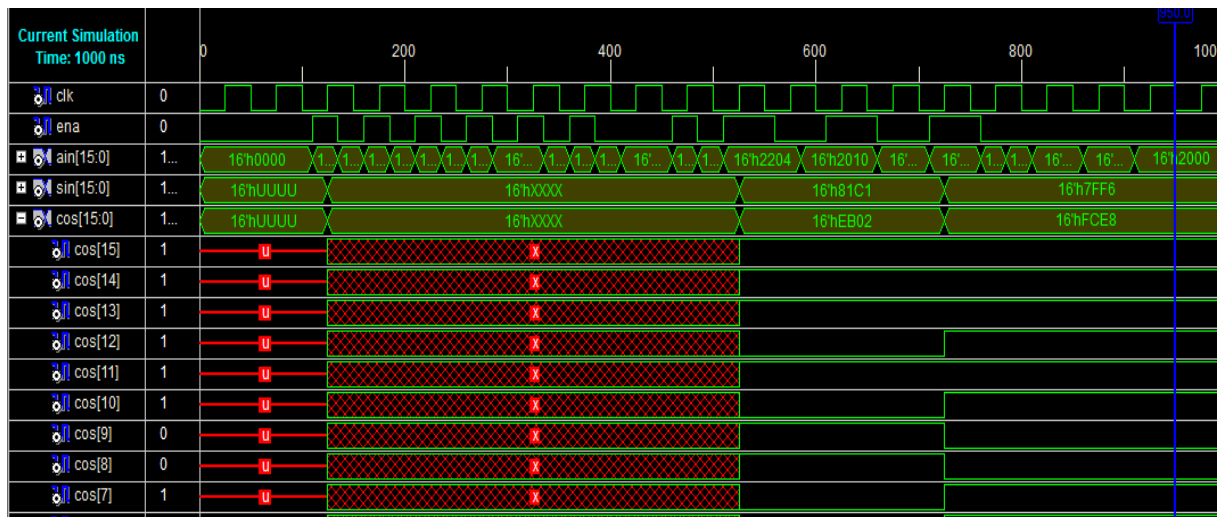


Figure 4.25. Résultat de simulation de l’algorithme CORDIC itératif à 8 étages.

Dans ce chronogramme, le cosinus et le sinus de phase sont disponibles après 8 cycles d’horloge.

La position des registres est telle qu’à chaque cycle d’horloge, les registres sont mis à jour une fois. Par conséquent, 8 cycles d’horloges seront nécessaires pour produire un échantillon. La fréquence d’échantillonnage théorique correspondante sera donc de $f/(n+1)$, où f est la fréquence de fonctionnement du circuit sur lequel est implantée cette architecture. De plus, nous cherchons à obtenir la plus grande fréquence d’échantillonnage possible, et cette architecture nécessite plusieurs cycles d’horloge pour produire un seul échantillon, ce qui n’est généralement pas satisfaisant

B.3. L’implémentation du modulateur numérique QAM₂

La limitation de l’architecture précédente est qu’elle restreint fortement la fréquence d’échantillonnage théoriquement possible, d’un facteur au moins égal à la précision en bits des échantillons de sortie. Il est cependant possible, de produire un échantillon de sinusoïde par cycle d’horloge, en utilisant l’architecture QAM₂ à base du processeur CORDIC pipeline. Cette caractéristique améliore les performances en augmentant la fréquence de transmission des échantillons et diminue le nombre de cellules logiques. Cependant, elle reste insuffisante pour une transmission plus rapide.

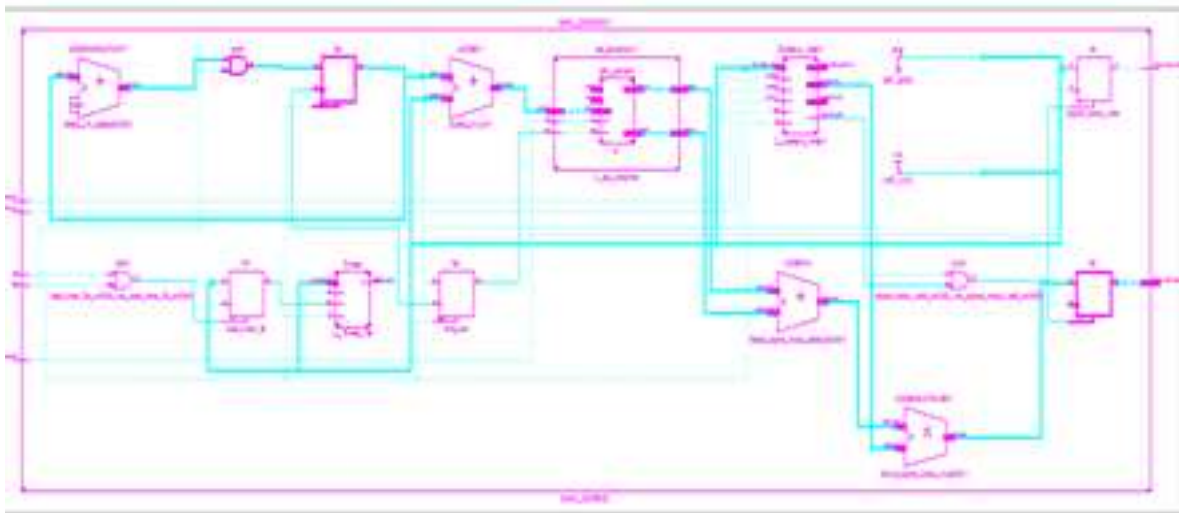


Figure 4.26. Vue interne de l'implémentation d'un modulateur QAM₂

Résultat de synthèse

La Figure (4.25) présente le schéma RTL de processeur CORDIC pipeline :

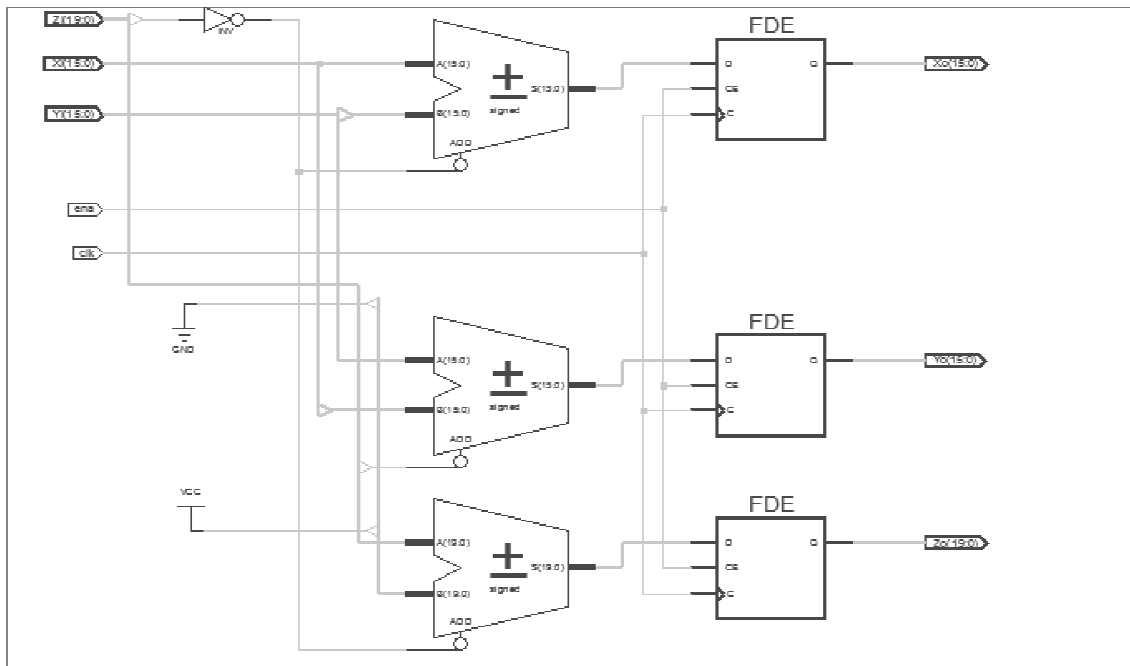


Figure 4.27. Vue interne de l'implémentation du CORDIC pipeline.

Simulation

Nous présentons les résultats de simulation dans les figures suivantes :

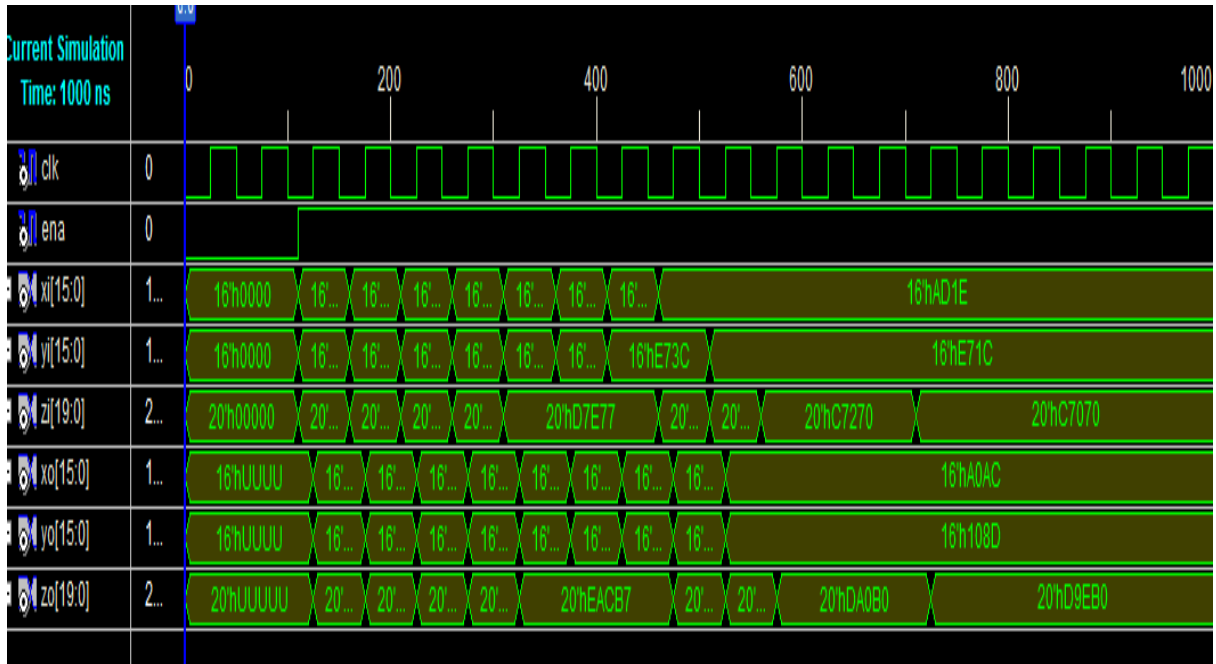


Figure 4.28. Résultat de simulation d’algorithme CORDIC pipeline .

Dans le chronogramme ci-dessus on a :

- Les signaux X_i et y_i représentent l’amplitude des signaux à générer.
- Le signal Z_i représente la phase de signal à générer.
- Les signaux X_0 et Y_0 représentent le cosinus et le sinus de la phase.

Dans le tableau 4.5 nous présentons les résultats des performances selon le nombre de bits

CORDIC Pipeline n (bits)	Fréquence d’échantillonnage (M-échantillons/s)	Cellules (cell)
02	202.80	102
04	162.66	112
08	104.45	130
16	78.12	1410
32	10.33	3535

Tableau 4.5. Variations des paramètres du générateur de sinusoïde à base d’un CORDIC pipeline.

L'examen de ce tableau indique que lorsque la précision en nombre de bits augmente la fréquence des échantillons transmis par le générateur de sinusoïde diminue en raison de l'augmentation du nombre de cellules exigées par l'architecture considérée

B.4. L'implémentation du modulateur numérique QAM₃

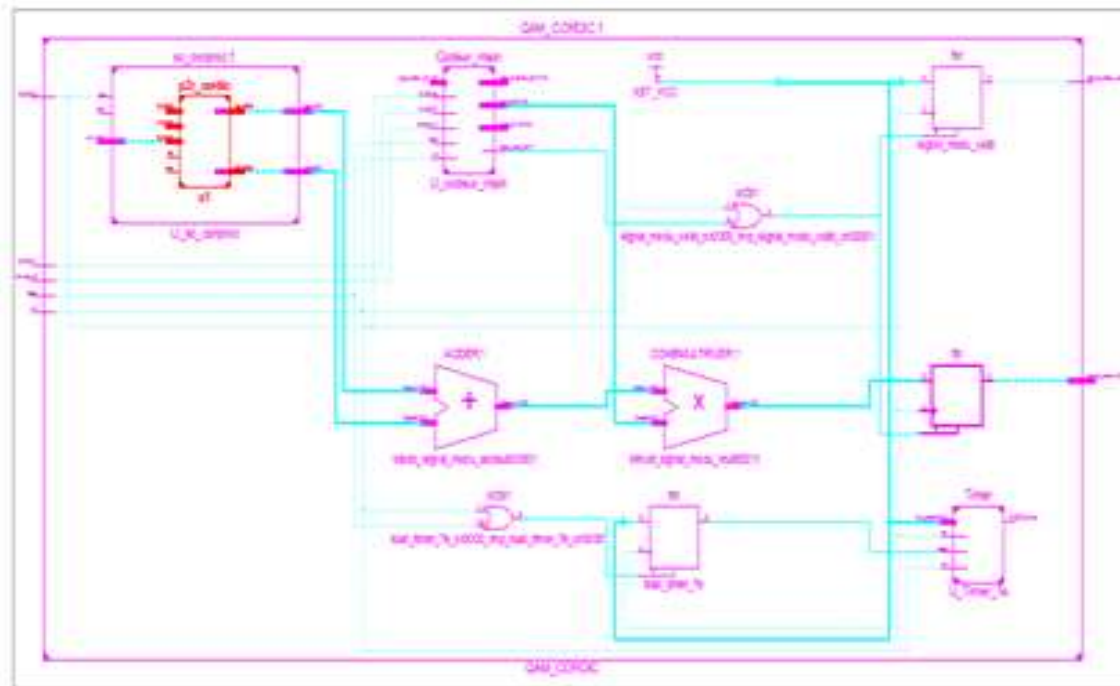


Figure 4.29. Vue interne de l'implémentation du modulateur QAM₃

Résultat de synthèse

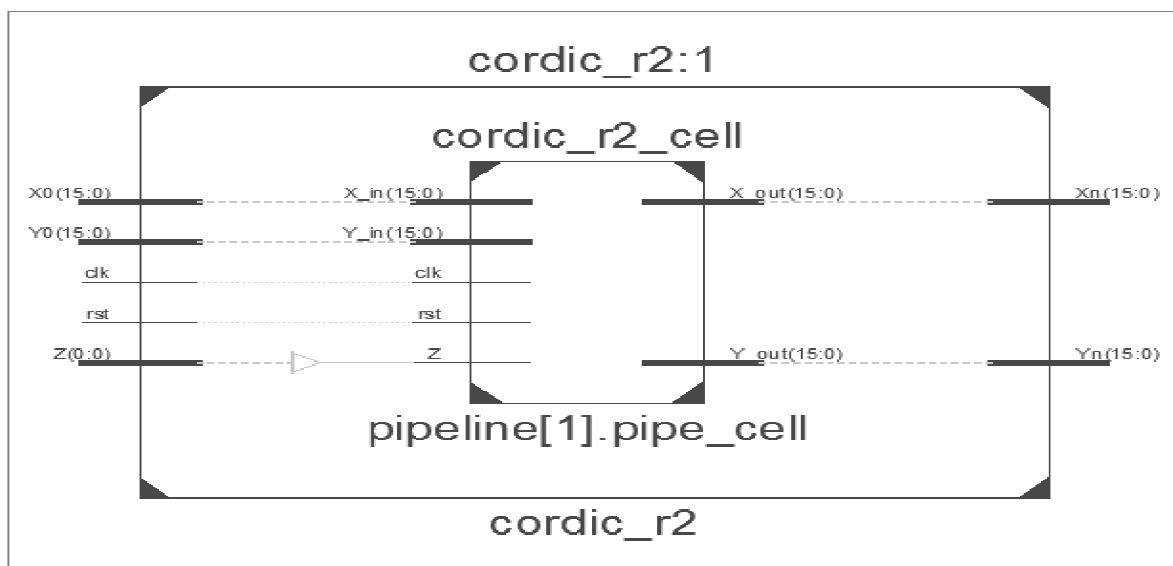


Figure 4.30. Vue externe de l'implémentation du CORDIC pipeline sans accumulateur de phase.

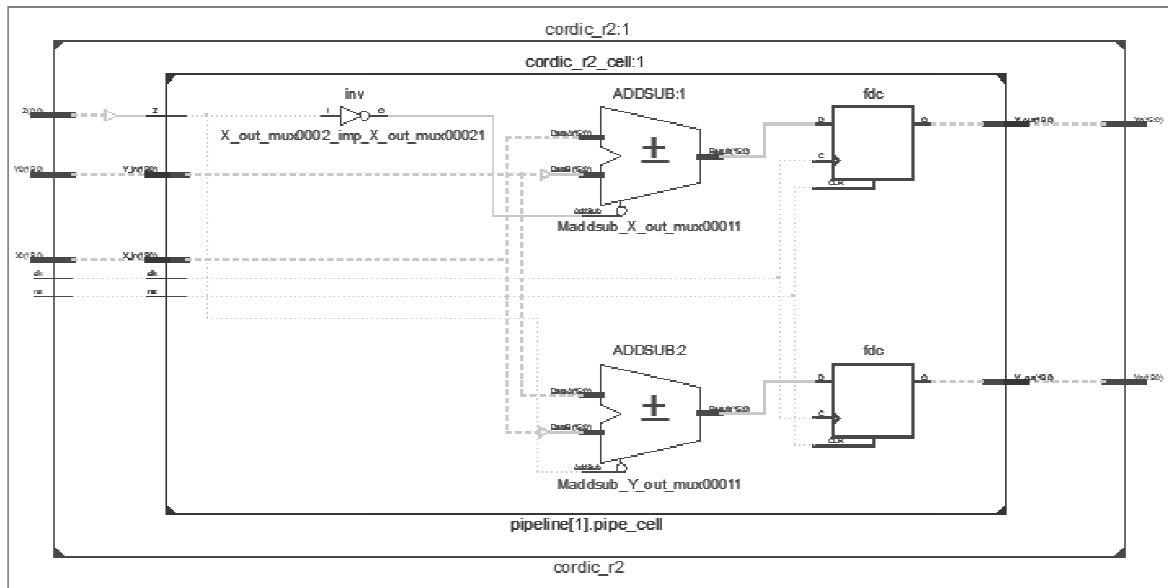


Figure 4. 31. Vue interne de l'implémentation sur FPGA Virtex 5 du CORDIC pipeline sans accumulateur de phase.

Dans le tableau suivant est indiquée la variation des performances du CORDIC pipeline sans accumulateur de phase en fonction du nombre de bits. Cela nous a permis d'atteindre des fréquences d'échantillonnage améliorées par rapport aux autres architectures et acceptables sur FPGA.

CORDIC pipeline sans accumulateur de phase	Fréquence d'échantillonnage (M-échantillons/s)	Cellules (cell)
n (bits)		
02	966.464	8
04	635.805	36
08	602.518	136
16	542.977	528
32	455.154	1953

Tableau 4.6 : Variations des performances obtenues en utilisant le processeur CORDIC pipeline sans accumulateur de phase.

B.5. Résultats d'implémentation des deux architectures de générateurs de sinusoïdes sur le circuit FPGA

Ces générateurs de sinusoïdes à base de processeur CORDIC ont été implantés sur un FPGA de type XC5LX110T.

Dans ce qui suit (figure 4.32), nous représentons graphiquement les variations des paramètres (fréquences d'échantillonnage et des nombres de cellules consommées) en fonction du nombre de bits pour les deux processeurs CORDIC (pipeline avec accumulateur de phase puis sans).

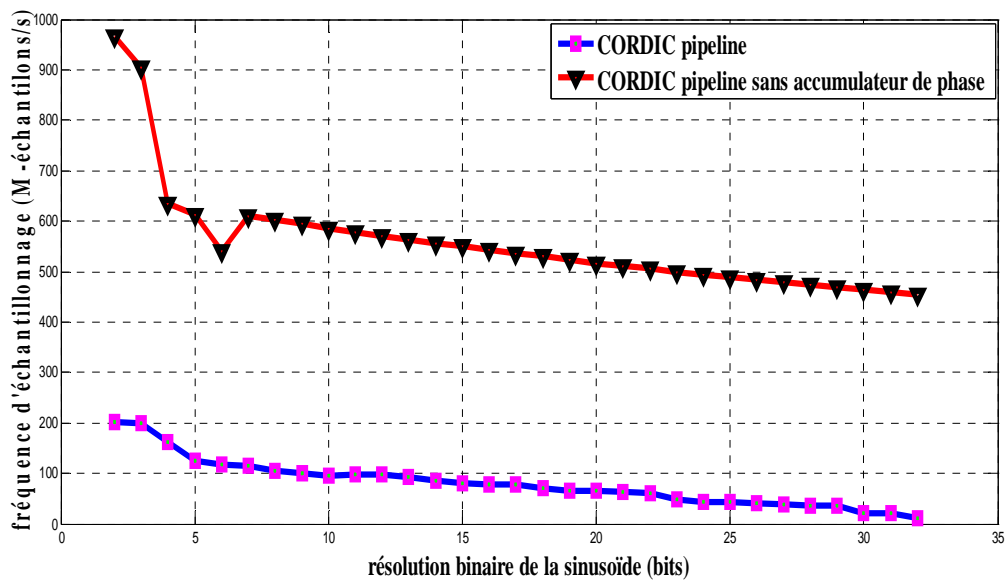


Figure 4.32. Variation de la fréquence d'échantillonnage en fonction de la résolution binaire pour les deux architectures.

La figure 4.33 montre variation du nombre de cellules logiques élémentaires consommées par les deux architectures, en fonction du nombre de bits injectés.

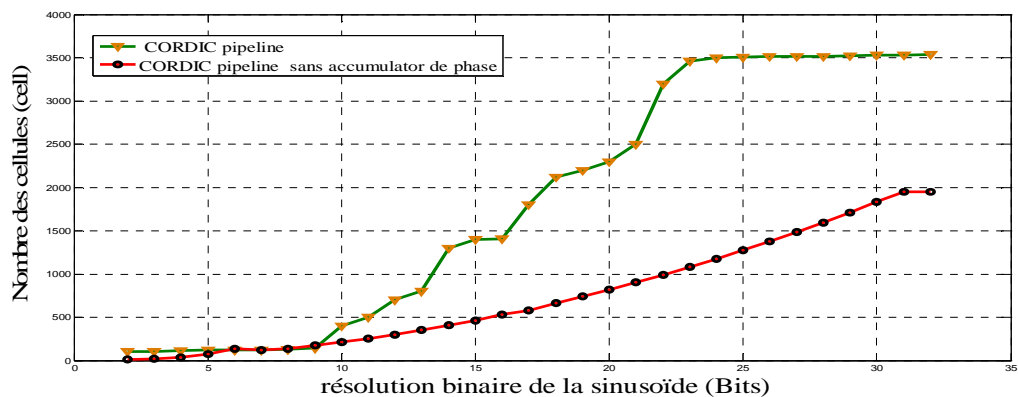


Figure 4.33. Variation (pour les architectures considérées) du nombre de cellules logiques requis sur le circuit FPGA en fonction du nombre de bits en entrée.

Nous déduisons des deux figures précédentes les résultats des performances obtenues par les deux générateurs en 8 bits dans le tableau 4.7

Paramètres Architectures	Fréquence d'échantillonnage (M-échantillons/s)	Cellules (cell)
CORDIC pipeline	104.45	130
CORDIC pipeline sans accumulateur de phase	602.518	136

Tableau 4.7. résultats comparatifs des générateurs de sinusoides considérés pour 8 bits.

Les résultats précédents indiquent que la meilleure performance en 8 bits est obtenue avec l'architecture CORDIC pipeline sans accumulateur de phase débitant une fréquence de 602.512 M-échantillons/s tandis que les nombres de cellules logiques requis par les deux architectures sont légèrement différents.

Avec chacune des différentes architectures des générateurs de sinusoides décrits précédemment, trois modulateurs différents (QAM_1 , QAM_2 et QAM_3) sont implantés sur circuit FPGA

B.6. Résultats d'implantation des trois modulateurs QAM

Résultat de synthèse

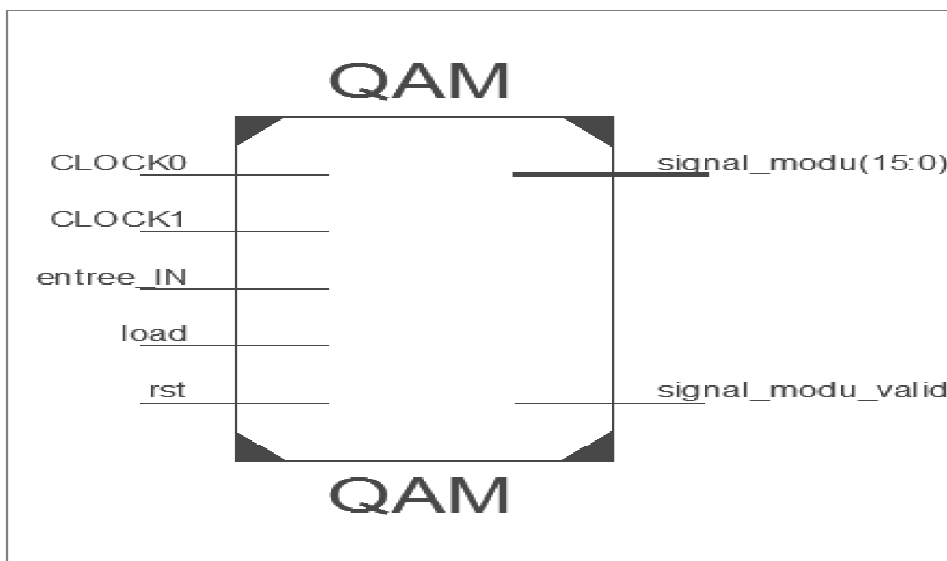


Figure 4.34. Vue externe de l'implémentation du modulateur QAM.

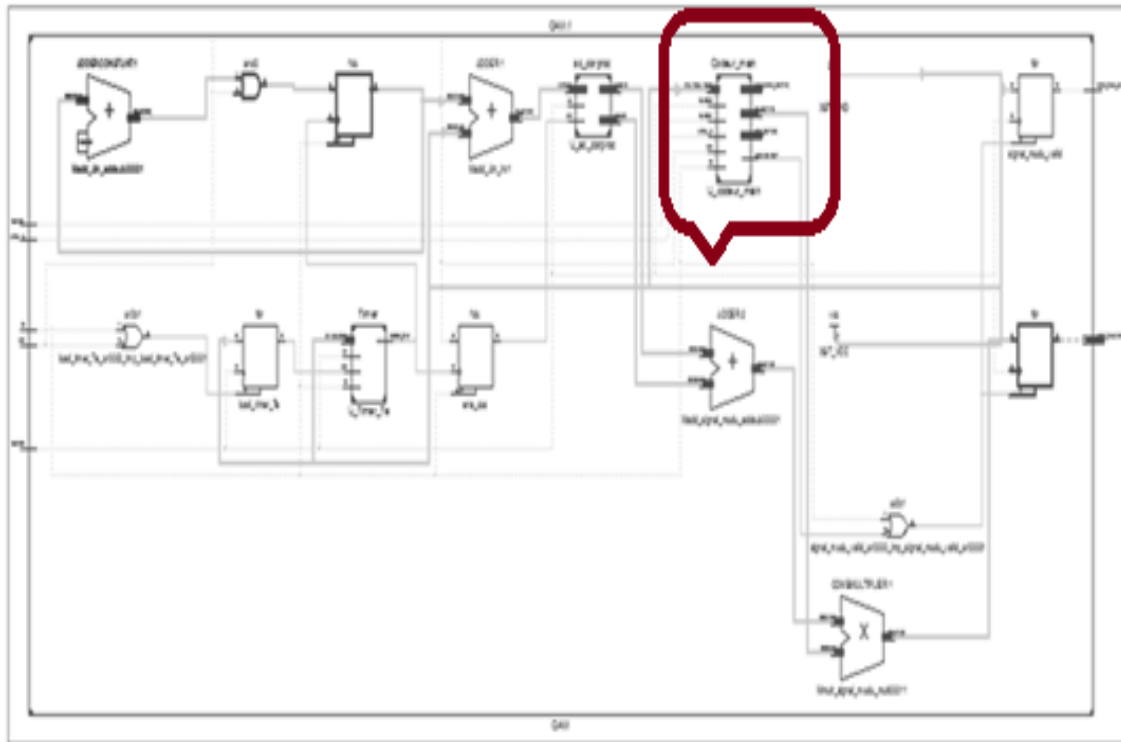


Figure 4.35. Vue interne de l'implémentation d'un modulateur QAM.

Le tableau 4.8 indique les résultats d'implantation de différents modulateurs :

(16 bits)	fréquence de fonctionnement (MHz)	Slices luts (19200)	Slices registres (19200)	Puissance (mW)
QAM ₁	232.520	467	384	267.18
QAM ₂	237.102	102	51	266.84
QAM ₃	284.511	92	51	266.84

Tableau 4.8. Résultats de synthèse des architectures proposées en 16 bits.

Le tableau ci-dessus indique que l'architecture QAM₃ présente une faible consommation de puissance, car nous avons utilisé le générateur de sinusoïde à base de CORDIC pipeline, ainsi qu'une diminution du nombre de cellules (slice lut et slice register) et une augmentation de la fréquence de transmission des échantillons par rapport aux autres architectures en 16 bits.

Simulation

La figure 4.36 montre le résultat de simulation de l'implémentation du modulateur complet QAM₃.

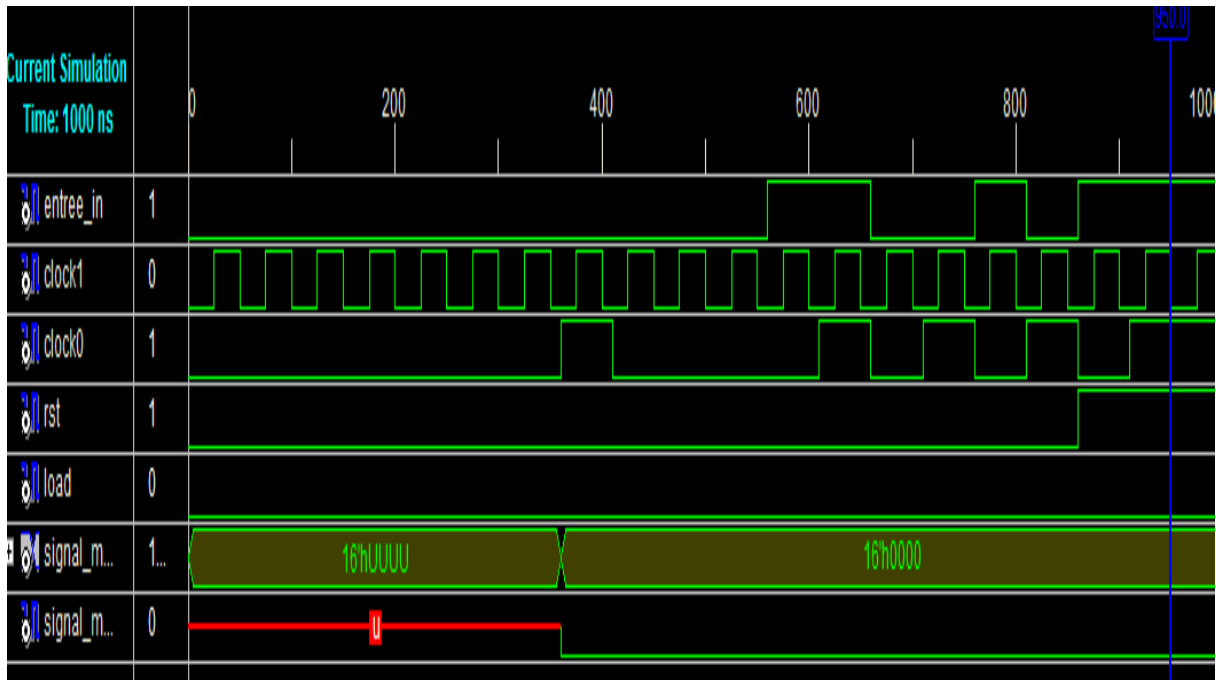


Figure 4.36. Résultats de simulation d'un modulateur numérique d'amplitude et de phase à base d'opérateur CORDIC pipeline.

4. Conclusion

À performances comparées, les résultats pour l'ensemble des architectures présentées sont résumés sur le tableau précédent. Le circuit FPGA de destination est de type XC5LX110T. La fréquence maximale annoncée est celle du QAM₃, ce qui correspond à une fréquence de fonctionnement de 284.511Mhz.

Nous avons procédé à la simulation temporelle du modulateur numérique complet pour les trois techniques dont les schémas RTL ont été présentés. Dans cette simulation, le générateur de sinusoïde est à base du processeur CORDIC pour répondre aux exigences d'une implantation optimale sur un circuit numérique de type FPGA.

Les résultats de simulation ont permis de montrer que l'architecture pourvue d'un processeur CORDIC pipeline sans accumulateur de phase est la plus performante. Ceci améliore donc l'efficacité de ce modulateur QAM.

***CONCLUSION
GENERALE
ET
PERSPECTIVES***

L'objectif de ce travail de thèse a été d'optimiser un système de transmission des données à très haut débit en télécommunications.

Le système étudié est basé sur un modulateur numérique. Pour cela, nous avons développé et proposé des architectures de générateur de sinusoïde à base de processeur CORDIC pipeline, pour obtenir différentes architectures de modulateur numériques d'amplitude et de phase QAM.

Après la présentation des différents langages et outils de description du matériel ainsi que celui du développement, nous avons choisi les outils de conception de Xilinx simple et rapide.

Nous avons ensuite présenté des différentes technologies utilisées pour l'implémentation d'applications numériques (FPGA, PLD, CPLD), en essayant d'avoir des éléments de comparaison entre-elles. En effet, pour la validation de notre implémentation, nous avons utilisé le circuit FPGA virtex 5(XC5VLX110T).

Toutes les architectures ont été testées et simulées sur la technologie FPGA, pour une comparaison équitable. La solution retenue et proposée a montré son efficacité en termes de performance et coût. La performance a été illustrée par une fréquence de fonctionnement supérieure, et le coût par une consommation inférieure en ressources logiques. L'idée de supprimer l'accumulateur de phase du générateur de sinusoïde est d'augmenter la vitesse du modulateur.

Les travaux réalisés dans le cadre de cette thèse se sont déroulés comme ce qui suit :

- 1- Conception architecturale détaillée
- 2- Test et intégration
- 3- Validation.
- 4- Réalisation d'un modulateur numérique complet d'amplitude et de phase adapté au haut débit

Les résultats de simulation obtenus ont montré que l'architecture proposée, génère une fréquence maximale jusqu'à 284.51 MHz, réduit le nombre de cellules logiques utilisées à 829 cellules et ceci avec un minimum de puissance de consommation de 266.84 mw. Ces caractéristiques améliorent les performances en augmentant la rapidité et l'efficacité du modulateur. Afin de répondre aux exigences des multimédias et de mobilité, dans une perspective future on escompte étudier l'architecture du modulateur QAM à base de processeur CORDIC parallèle avec son démodulateur numérique correspondant adaptées au très haut débit



REFERENCES

Références bibliographique

- [1] Yamada E., Kikuchi N., Sano A., Kobayashi T., Shibata Y., Ishii H. & Miyamoto Y., 2012. 86 Gbit/s Pdm 16-qam signal transmission using inp optical iq modulator, Electronics Letters, Vol . 48 (23), 1486 – 1487.
- [2] Paulsson K., Hübner M. 2008. Cost-and power optimized FPGA based system integration:methodologies and integration of a low-power capacity-based measurement application on Xilinx FPGAs, Proceedings of Design, automation and test'2008, Europe, 50-55.
- [3] Andrew moore,2014,FPGA for dummies Altera special edition, John wiley and sons Inc, unites states of America.
- [4] Arnould G., 2006. Etude et conception d'architectures haut-débit pour la modulation et la démodulation numériques, Manuscript of thesis, LICM, University of Metz, 142 p.
- [5] Peter J. Ashenden The student's guide to VHDL (2nd edition) Morgan Kaufmann,2008.
- [6] Gerald Arnould, Abbas Dandache and fabric Montero, " digital frequency synthesizer architectures comparaison for à high- speed quadrature amplitude modulator" DCIS2005-xx conference on design of circuit and integrated system, lisboa, portogal, novembre 23-25-2005.
- [7] Xilinx. ISE 12.2i software manuals. Xilinx; 2008.
- [8] Xilinx, Embedded Processor Block in Virtex-5 FPGAs, Xilinx User Guide,UG200, February 2010.
- [9] Peter J. Ashenden The student's guide to VHDL (2nd edition) Morgan Kaufmann, 2008.
- [10]GLAVIEUX Alain / JOINDOT Michel, Communications numériques, Paris, Masson, 1996.
- [11] <https://perso.liris.cnrs.fr/florent.dupont/Enseignement/Reseaux-sans-fil>.
- [12]. Luis Alejandro ANDIAMONTES. « Amplificateur de puissance en classe commutée pour application dans un émetteur multiradio à haut rendement » Thèse doctorat, Spécialité: Electronique, Optronique et Systèmes (CNU 63) Laboratoire ESYCOM de l'Université de Paris-EST, octobre 2010.
- [13] Boudjaadar Amina ; « Plateforme basée Agents pour l'aide à la conception et la simulation des réseaux de capteurs sans fil ». Thèse de Magistère ; Université de Skikda ; 2009/2010.
- [14] LAN-MAN; Standards Committee of the IEEE Compter Society -802.11 IEEE Standard for Information technology; 1999.
- [15] A. Tanenbaum, Réseaux, 4ème édition, Pearson Education France, 2003.

- [16] D. E. Comer, TCP/IP architecture, protocoles, applications, 3me édition, Dunod, Paris, 2000.
- [17] Site web : www.etsmtl.ca/rlandry
- [18] M. Busby, Introduction à TCP/IP, Osman EyrollesMultimedia, Paris, 2000.
- [19] page web : nicolas.baudru.perso.esil.univmed.fr
- [20] <http://www.editionsorganisation.com/Livre/9782212091557/initiation-aux-reseaux>
- [21] <http://spoken-tutorial.org/NMEICT-Intro>. This Textbook Companion and Scilab.
- [22] John G. Proakis, Digital communications, USA, McGraw-Hill 1989.
- [23] Michel Degauque, Transmission numérique sur porteuse : ASK, FSK et PSK. Probatoire du CNAM de Bordeaux, juillet 1998.
- [24] <http://www.enseirb.fr/~kadionik>
- [25] J. G. Proakis, *Digital Communications*, McGraw Hill, 1995.
- [26] Kurakake T., Nakamura. & Oyamada K., 2004.A blind 1024-qam demodulator for cable television, Proceedings of International Zurich Seminar on Communications'2004, Zurich, Switzerland, Switzerland, 136 – 139.
- [27] Trung H. D, Tuan D. T. & Pham A. T., 2014. Pointing error effects on performance of free-space optical communication systems using sc-qam signals over atmospheric turbulence channels, International Journal of Electronics and Communications, Vol . 68 (9), 869-876.
- [28] Mohini., Priya ., Sur S.N., Bera S., 2019. labview-based software and hardware implementation of 16-qam modem, Proceedings of Advances in Communication, Devices and Networking'2019, Singapore, 221-227.
- [29]Instructor: N. Vlajic:Analog Transmission of Digital Data: ASK, FSK, PSK, QAM ;Required reading: Garcia 3.7. CSE 3213, Fall 2010.
- [30]Mamadou.Moustapha.NDIAYE:supinfo.com/articles/single/2618-modulation-numerique.Publié le 13/10/2016 à 19:06:59.
- [31] Patrice Kadionik :[modulations-numeriques.pdf_BASES DE TRANSMISSIONS NUMERIQUES](#)Les modulations numériques. Publié le 18 avril 2005.
- [32].S. Kumar Das, MOBILE HANDSET DESIGN, John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop, 02–01, Singapore 129809 Copyright 2010.
- [33]. W. T. Webb and L. Hanzo, Modern Quadrature Amplitude Modulation: Principles and Applications for Fixed and Wireless Channels, Wiley-IEEE Press, New York, NY, USA, 1994.
- [34]. Christos Ttofis, Agathoklis Papadopoulos, Theocharis Theocharides, Maria K. Michael, Demosthenes Doumenis: An MPSoC-Based QAM Modulation Architecture

with Run-Time Load-Balancing. EURASIP J. Emb. Sys. 2011 (2011).

[35]. Maamar Touiza, Gilberto Ochoa-Ruiz, El-Bay Bourennane, Abderrezak Guessoum, Kamel Messaoudi. « A novel methodology for accelerating bitstream relocation in partially reconfigurable systems», *Microprocessors and Microsystems*,

Volume 37, 2013, Pages 358–372, ISSN 10.1016/j.micpro.2012.07.004.

[36]. <http://docs-europe.electrocomponents.com/webdocs/0de1/0900766b80de1aa8.pdf>

[37]. J. E. Volder, "The CORDIC trigonometric computing technique", *IRE Trans. Electron. Comput.*, vol. EC-8, pp. 330–334, Sept. 1959.

[38]. G.ARNOULD, " Etude et Conception d'Architectures Haut-Débit pour la Modulation et la Démodulation Numériques", Manuscript of thesis, LICM, University of Metz, 2006.

[39] Naik R. & Nadaf, R., 2015. Sine-cosine computation using cordic algorithm, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol . 4, (9), 44-48.

[40] Boudjema, S. and S. Kaddour, 2012. Optimisation d'architecture de modulation QAM par Processeur CORDIC. Guelma1945.

[41] J. E. Volder, "The CORDIC trigonometric computing technique", *IRETrans. Electron. Comput.*, vol. EC-8, pp. 330–334, Sept. 1959.

[42] J. E. Volder, "The birth of CORDIC," *J. VLSI Signal Processing*, vol. 25,no. 2, pp. 101–105, June 2000.

[43] Kavya S., Uma B.V. &Sagar D.M., 2014. Calculation of sine and cosine of an angle using the cordic algorithm, *International Journal of Innovative Technology and Research*, Vol . 2 (2), 891 – 895.

[44] Xu R., Jiang Z., Huang H. & Dong C., 2015. An optimization of cordic algorithm and fpga implementation, *International Journal of Hybrid Information Technology*, Vol . 8 (6), 217-228.

[45] C. Y. Kang and E. E, Swartzlander, Jr., "A digit-pipelined direct digital frequency synthesis architecture", in *Proc. IEEE Workshop on Signal Processing Systems (SIPS 2003)*, Aug. 2003, pp. 224–229.

[46] C. Y. Kang and E. E, Swartzlander, Jr., "Digit-pipelined direct digital frequency synthesis based on differential CORDIC", *IEEE Trans. Circuits and Systems I : Regular Papers*, vol. 53, no. 5, pp. 1035–1044, May 2006.

[47] J. Duprat and J.Muller, "The CORDIC algorithm : new results for fast VLSI implementation", *IEEE Trans. Comput.*, vol. C-42, pp. 168–178, Feb. 1993.

- [48] D. S. Phatak, "Double step branching CORDIC : a new algorithm for fast sine and cosine generation", IEEE Trans. Comput., vol. 47, no. 5, pp. 587– 602, May 1998.
- [49] A. Singh et al., "Comparison of branching CORDIC implementations", in Proc. IEEE Int. Conf. on Application-Specific Systems, Architectures, and Processors, June 2003, pp. 215–225.
- [50] A. Benyamina, "Ordonnancement Hiérarchique Multi-Objectif D'Application Embarquées Intensives (CAO/IAO&Simulation)", thèse de doctorat de l'université d'Oran-Es-Senia, 2009.
- [51] David R. Smith, Introduction to digital transmission systems Springer, 3 th ed., 1985; 1-20.
- [52] Xilinx. ISE 12.2i software manuals. Xilinx; 2008.
- [53] Peter and Ashenden, The student's guide to VHDL, 2nd ed., Morgan Kaufmann, 2008.
- [54] Worchel, J., <http://www.instat.com/abstract.asp?id=68&SKU=0603187SI/> The Field-Programmable Gate Array (FPGA): Expanding Its Boundaries, InStat Market Research; 2006.
- [55] Arnould, G., A. Dandache and F. Montero, 2005. Digital frequency synthesizer architectures comparison for a high speed quadrature amplitude modulator DCIS 2005. Proceedings of the Conference on Design of Circuit and Integrated System, Nov. 23-25, Lisbon, Portugal.
- [56] Liu Zhi, Jiang Zhou, Li Qing, Zeng Xiaoyang (2007). Efficient carrier recovery for high-order QAM, "IEEE Xplore. Journal of Consumer Electronics, 25(1), 1-2.
- [57] Winzer, P.J., Gnauck, A.H., Doerr, C.R., Magarini, M. (2009). Spectrally efficient long-haul optical networking using 112-gb/s polarization-multiplexed 16-QAM. Journal of Light Wave Technology IEEE, 28(4), 547-556.
- [58] Mihra, R. and B. Kamboj, 2010. FPGA based design of digital wave generator using CORDIC algorithm. Int. J. Comput. Technol. Applic., 1: 54-48.
- [59] B.V.V. Satyanarayana, A.Pravin, Shafivullah Mohd, S.kanaka Durga, 2011, Design of Quadrature Amplitude Modulation System for Wireless Applications, International Journal of Electronics & Communication Technology.
- [60] Mandal, A. and R. Mishra, 2012. Reconfigurable design of pipelined CORDIC processor for digital sine-cosine, J. Signal Process. Theory Applic. DOI: 10.7726/jspta.2012.1003.
- [61] Sharat et al.(2014), 'Calculation of Sine and Cosine of an Angle Using the Cordic Algorithm', *International Journal of Innovative Technology and Research*, vol 2, No2, 891-895.

- [62] Yidong et al. (2014), 'A Modified Cordic FPGA Implementation for Wave Generation', journal of circuits systems and signal processing, Vol 33, No1, 321-329.
- [63] IEEE Standard Multivalued Logic System for VHDL Model Interoperability, IEEE Std 1164-1993.
- [64] J. WEBER, "Le langage VHDL". PARIS: DUNOD, 2000.
- [65] R. Munden, "ASIC AND FPGA VERIFICATION: A GUIDE TO COMPONENT MODELING", ELSEVIER, 2005.
- [66] W. KAFIG, VHDL 101. OXFORD: ELSEVIER, 2011.
- [67] J. ROSE , A. EL GAMAL and A. SANGIOVANNI-VICENTILLI, "Architecture of Field-Programmable Gate Array" in Proceeding of the IEEE Vol. 81 Issue N° 7, 1993.
- [68] Z. Ait Ouali, "Application des FPGA à la commande d'un moteur asynchrone", thèse de magister de l'université MOULOUD MAMMARI, TIZI-OUZOU, 2010.
- [69] Xilinx. ISE 12.2i software manuals. Xilinx; 2008.
- [70] site internet,
http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/v2012_2---14_2.html
- [71] Paulsson K., Hübner M. & 2008. Cost-and power optimized FPGA based system integration: methodologies and integration of a low-power capacity-based measurement application on Xilinx FPGAs, Proceedings of Design, automation and test'2008, Europe, 50-55.
- [72] J. DETREY, "Arithmétiques réelles sur FPGA virgule fixe, virgule flottante et système logarithmique", Thèse de doctorat de l'Ecole Normale Supérieure de Lyon, Janvier 2007.
- [73] D. SMITH, "HDL Chip Design: A practical Guide for Designing, Synthesis & Simulating Asics & FPGAs using VHDL or Verilog", Doone Pubns, 2006.
- [74] V. Betz and J. Rose, "FPGA Routing Architecture: Segmentation and Buffering to Optimize Speed and Density", In International Symposium on Field Programmable Gate Arrays, 1999.
- [75] K.-J. Lee, T.-Y. Hsieh, C.-Y. Chang, Y.-T. Hong, and W.-C. Huang, "On-Chip SOC Test Platform Design Based on IEEE 1500 Standard", IEEE Trans. Very Large Scale Integr. Syst., vol. 18, no. 7, pp. 1134–1139, Jul. 2010.
- [76] Benaïcha ramzi "les télécommunications sans fil et les réseaux de communication embarquées". thèse de doctorat, université badji mokhtar annaba, 2016.

- [77] S. SNAIDERO, "Modélisation multidisciplinaire VHDL-AMS de systèmes complexes vers le prototypage virtuel", thèse de doctorat de l'université Louis Pasteur Strasbourg I, décembre 2004.
- [78] Xilinx, Embedded Processor Block in Virtex-5 FPGAs, Xilinx User Guide, UG200, February 2010.
- [79] C. E. Shannon, "A mathematical theory of communication", Bell System
- [80] IEEE Standard for VHDL Register Transfer Level Synthesis, IEEE 1076.6-1999.



***LISTE DES
COMMUNICATIONS***

Liste des publications

- [1] S.BOUDJEMA,K.SAOUCHI : Optimisation De L'architecture De Modulation QAM Par Processeur CORDIC Hybride Parallèle.(ICESTI international conférence on Embedded Systems in télécommunications and instrumentation Annaba- Algérie ICESTI), novembre 5,07,2012), plus une participation au membre du comité d'organisation et aussi obtenu le meilleur papier dans cette conférence.
- [2] S.BOUDJEMA, K.SAOUCHI : Etude des systèmes de transmission/réception des données à haut débit en télécommunication. (JIDATIM journées doctorales sur l'automatique, les télécommunications, l'instrumentation), 2012.
- [3] S.BOUDJEMA, K.SAOUCHI : Optimisation De L'architecture De Modulation QAM Par Processeur CORDIC. (PCNTE'12 première conférence on télécommunications de Guelma Algérie), novembre 10-11, 11,2012).
- [4] S.BOUDJEMA, K.SAOUCHI : Sine generator based of parallel cordic. (ICNCRE international conférence on nanoelectronics, communications and renewable energy jijel, Algérie,septembre, 22-23,9,2013.
- [5] S.BOUDJEMA, K.SAOUCHI : Étude le Système de Transmission des Données a Haut Débit en Télécommunication. (ICESTI international conférence on Embedded Systems in télécommunications and instrumentation Annaba- Algérie, 27-29 octobre 2014.
- [6] S.BOUDJEMA, K.SAOUCHI : 'Study a more effective technique for the digital modulation amplitude and phase. (ICESTI international conférence on Embedded Systems in télécommunications and instrumentation Annaba- Algérie, 24-27 octobre 2016
- [7] S.BOUDJEMA, K.SAOUCHI : The Optimal Implementation of a Generator of Sinusoid. Revue american journal of applied sciences 2015, 12 (8): 593.600DOI: 10.3844/ajassp.2015.593.600.
- [8] S.BOUDJEMA, K.SAOUCHI : Architectures of a digital modulator for high-speed data transmission in telecommunication system. Revue des sciences et de la technologie (revue synthèse), de l'université badji mokhtar annaba,2019:
- [9] S.BOUDJEMA, K.SAOUCHI: A novel method for digital modulator amplitude and phase using cordic algorithm.(ICESTI'19 (international conférence on Embedded Systems in télécommunications and instrumentation Annaba- Algérie), 28-29 octobre 2019.