

UNIVERSITÉ BADJI
MOKHTAR-ANNABA

THESE
DE DOCTORAT EN SCIENCE

Présentée pour obtenir le grade de Docteur en science
de l'université Badji Mokhtar de Annaba
Spécialité: Télécommunications

Par

BENHAOUES Atef

**Architectures numériques versatiles avancées
pour la transmission sans fil de données à très
haut débit**

Soutenue le

devant la commission d'examen composée de:

Mr. LARBI Allal	Professeur, Université Badji Mokhtar d'Annaba	Président
Mr. FEMMAM Smaïn	Professeur, Ecole Polytechnique de Paris	Examineur
Mr. CHAOUI Allaoua	Professeur, Université Mentouri-Constantine	Examineur
Mr. SAOUCHI Kaddour	Maître de conférence, Université Badji Mokhtar d'Annaba	Examineur
Mr. BOURENNANE El-Bay	Professeur, Université de Bourgogne-Dijon	Invité
Mr. TOUMI Salah	Professeur, Université Badji Mokhtar d'Annaba	Directeur
Mr. TANOUGAST Camel	Professeur, Université Paul Verlaine-Metz	Co-Directeur

Remerciements

D'abord et avant tout, je remercie mon DIEU « **Allah** » pour m'avoir accordé la santé, la patience, la connaissance et tout ce qui est nécessaire pour terminer cette thèse.

Je tiens à remercier toutes les personnes qui m'ont accompagné et guidé pendant ma thèse. Ma gratitude va d'abord au Pr. *TOUMI Salah* et Pr. *TANOUGAST Camel*, qui m'ont proposé le sujet et qui pendant ces années m'ont encadré dans mes activités de recherche. Merci donc de m'avoir écouté et guidé, vous m'avez beaucoup appris dans le domaine scientifique et professionnel évidemment, mais aussi vous avez contribué à ma croissance personnelle.

Je tiens également à remercier Monsieur *BOURENNANE El-Bay*, Professeur à l'université de Bourgogne-Dijon pour avoir su me guider avec attention et gentillesse pendant ces années.

Je remercie Monsieur *LARBI Allal*, Professeur à l'université de Badji Mokhtar d'Annaba, qui m'a fait l'honneur de présider le jury.

Mes remerciements vont également à Monsieur *FEMMAM Smaïn* Professeur à l'école Polytechnique de Paris, Monsieur *CHAOUI Allaoua* Professeur à Université Mentouri de Constantine, et Monsieur *SAOUCHI Kaddour* Maître de Conférences à Université de Badji Mokhtar d'Annaba pour avoir accepté d'examiner ce travail et de l'enrichir par leurs grands savoirs et leurs grandes expériences.

A mes amis: *Hichem MAYACHE*, *MESSAOUDI Kamel*, *El Hadi MHELEL*, je vous remercie car pendant ces années vous m'avez donné votre support de différentes manières, en m'écoutant, en discutant, en me donnant vos avis, en me faisant rire, en me faisant découvrir le monde. J'exprime aussi mes sincères remerciements à tous les *membres du laboratoire LERICA*.

Enfin, à mes êtres les plus chers, un grand merci, à ma femme *Rym* évidemment qui jour après jour m'a soutenu dans cette aventure et qui m'a permis de garder mon optimisme et ma volonté dans les moments de stress et de difficulté. A ma famille pour le support, pour m'avoir appris les valeurs de la vie. Un remerciement particulier pour *maman* et *papa* pour leurs encouragements et leurs recommandations pleines de persévérance. Je les remercie pour leur amour, leur foi et pour m'avoir encouragé dans tous les moments de ma vie à faire toujours mieux. Je remercie aussi mon fils *Abd El Matine* malgré il est encore un bébé, mon frère *Bassem*, mes sœurs *Zahra* et *Loubna*, mon oncle *Ahmed* pour leur soutien constant.

Résumé

Dans un contexte de multiplication des normes de télécommunications aux spécifications distinctes, nos travaux de recherche définissent une architecture reconfigurable pour des terminaux radio multistandard. Aujourd'hui les multiples normes et standards de télécommunications envahissent nos vies quotidiennes. Par conséquent le défi actuel est de concevoir de nouvelles architectures entièrement numériques, en plus elles sont capables de prendre en charge de nombreuses fonctions, cela conduit à éviter l'utilisation de nombreux circuits pour ces différentes fonctions. La radio logicielle a été créée pour fournir à l'utilisateur un terminal unique capable de naviguer entre les multiples standards de radiocommunication. Les premiers travaux de la radio logicielle supposent qu'avec un processeur généraliste on peut supporter l'ensemble des traitements nécessaires de plusieurs standards. Cependant l'utilisation d'un processeur généraliste ne répond pas à plusieurs contraintes tels que, la consommation, l'occupation de ressources matérielles et la puissance de calcul. Afin d'accéder à la combinaison "reconfiguration temps réel/généricité du terminal", nous avons choisi la technique de la paramétrisation et l'approche d'opérateur commun. Cette technique demande à faire une synthèse des travaux concernant les architectures numériques pour la modulation et le codage canal pour les différents standards et normes, dont le but est de trouver les similarités entre les différentes architectures. Ces similarités permettent de définir des blocs génériques pouvant être réutilisés facilement par un simple téléchargement de paramètres. Ceci permet d'obtenir un équipement multistandards reconfigurable et évolutif tout en limitant l'occupation en ressource matérielle et le temps de reconfiguration. Dans ce travail de thèse, nous avons identifié les similarités entre les unités de désétalement dans le récepteur Rake, la cellule de l'algorithme Cordic et l'élément de traitement (PE) pour le FFT-SDF (Fast Fourier Transform path Single Delay Feedback) pour proposer une architecture générique capable de prendre en charge les trois fonctionnalités. L'architecture proposée est codée en utilisant le VHDL sur un FPGA Virtex-5 et les résultats sont comparés à des travaux similaires. La mise en œuvre démontre que l'architecture proposée peut fournir une forte réduction de la consommation en logique dans le FPGA avec une très haute fréquence de travail. L'architecture proposée peut facilement prendre en charge d'autres fonctions comme le décodage de Viterbi. Dans la deuxième partie de ce manuscrit nous avons proposée deux architectures versatiles et génériques : la première architecture proposée est utilisée pour la modulation numérique de plusieurs types de modulation (QAM, PSK et FSK). La deuxième architecture proposée est utilisée pour la démodulation numérique, le code de convolution et le code de brouillage (Gold). Les deux architectures montrent une très faible consommation en ressource dans le FPGA et une fréquence de fonctionnement très grande.

Mots-clés:

Radio reconfigurable, architectures génériques, paramétrisation, opérateurs communs, FFT-SDF, récepteur Rake, algorithme CORDIC, synthèse architecturale, décodage de Viterbi.

Abstract

In a context of increasing communication standards with distinct specifications, our research defines a reconfigurable architecture for multi-standard radio terminals. Today the multiple communication standards invade our daily lives. Therefore, the current challenge is to design new digital and adaptive generic architectures. In addition these versatile architectures are able to take over of many functions, which leads to avoid the use of many circuits for these different functions. The software radio was created to provide to the user a single terminal that capable of navigating between multiple radio standards. The ideal software radio envisages that by using a general-purpose processor, we can do the entire necessary signal processing to several standards. However, the use of a general-purpose processor doesn't respond to several constraints such as energy consumption, the hardware resources occupation and power computing. To access the combination of "real time reconfiguration / generic terminal", we chose parameterization technique and common operator approach. This technique requires to make a synthesis on digital architectures of modulation and channel coding for the different standards, which aims to find the similarities between different architectures. These similarities make it possible to define generic blocks that can be reused easily by simple parameter download. This allows to obtain reconfigurable and scalable equipment while limiting the occupation in material resource and the reconfiguration time.

In this thesis, we have identified the similarities between the despreading units in the Rake receiver, the Cordic cell and the Processing Element (PE) for the FFT-SDF (Fast Fourier Transform path Single Delay Feedback) to propose a generic architecture capable to support the functionality of the three algorithms. The proposed architecture is coded using VHDL on a Virtex-5 FPGA and the results are compared to similar works. The implementation demonstrates that the proposed architecture can deliver a high reduction of the FPGA logic requirements with high maximum work frequency. The proposed architecture can easily support other functions such as Viterbi decoding.

In the second part of this thesis we proposed two versatile architectures: the first generic architecture is used for several types of digital modulation (QAM, PSK et FSK). The second architecture is used for digital demodulation, convolutional code and scrambling code (Gold). Both architectures show very low resource consumption in FPGA and high operating frequency.

Keywords:

Reconfigurable radio, generic architectures, parameterization, common operators, FFT-SDF, Rake receiver, CORDIC algorithm, Optimized architectural synthesis, Viterbi decoding.

الملخص

في سياق الزيادة المستمرة في عدد معايير الاتصالات مع مواصفات مختلفة، فإن بحثنا يتلخص في تشكيل بنية متعددة الوظائف و متعددة المعايير تستعمل في محطات الراديو وأجهزة النقل. اليوم لا يخفى على أحد، أن معايير اتصالات متعددة تغزو حياتنا اليومية. وبالتالي فإن التحدي الحالي هو تصميم أبنية رقمية جديدة، بالإضافة أنها تكون قادرة على تولي العديد من المهام و الوظائف، وهذا يؤدي إلى تجنب استخدام العديد من الدوائر لهذه الوظائف . تم إنشاء الراديو المُبرمج (radio logicielle) لتوفير محطة واحدة للمستخدم قادرة على التنقل بين العديد من معايير الراديو المختلفة. الأعمال الأولى للراديو المُبرمج تقتض بأنهُ يمكن دعم جميع الوظائف اللازمة لهذه المعايير المتعددة باستخدام معالج رقمي ذو أغراض عامة (processeur généraliste). ولكن استخدام هذا المعالج (généraliste processeur) لا يستجيب إلى العديد من القيود التي يجب توفرها في هذا الجهاز مثل استهلاك الطاقة، استهلاك الموارد والقدرة الحاسوبية . للوصول إلى الثنائية: " إعادة تشكيل في الوقت الحقيقي / محطة عامة"، فقد اخترنا العمل بتقنية الثوابت مع نهج العامل المشترك (la paramétrisation et l'approche opérateur commun). تتطلب هذه التقنية القيام بدراسة العديد من الأبنية الرقمية للتضمين و الترميز المستخدمة في المعايير المختلفة، و من ثم نهدف إلى إيجاد أوجه الشبه بين الأبنية المختلفة. التشابهات التي تم تحديدها تجعل من الممكن تحديد و تشكيل بنية عامة ذات قدرة على تغيير وظائفها بسهولة باستخدام بعض القيم المُنزلة فقط. هذا يسمح بالحصول على بنية متعددة الوظائف وقابلة للتطوير مع الحد من استهلاك الموارد المادية و وقت إعادة التشكيل. في هذه الأطروحة، حددنا أوجه التشابه بين وحدات desreading في المتلقي Rake ، خوارزمية CORDIC و عنصر المعالجة (PE) (FFT-SDF) و اقتراح بنية عامة قادرة على دعم وظيفة الخوارزميات الثلاثة. تم ترميز الهيكل المقترح باستخدام VHDL على Virtex 5 FPGA وتم مقارنة النتائج مع أعمال مماثلة. يوضح التنفيذ أن الهيكل المقترح يمكن أن يوفر خفض كبير في استهلاك الموارد مع وتيرة عمل عالية. الهيكل المقترح يمكن أن يدعم بسهولة وظائف أخرى مثل تشفير فيتربي (Viterbi decoding).

في الجزء الثاني من هذه الأطروحة اقترحنا بنيتين متعددة الاستخدامات، العمارة الرقمية العامة الأولى تستخدم في الحصول على العديد من أنواع التضمين الرقمية. العمارة الرقمية العامة الثانية، تقوم بالعملية العكسية للتضمين و تقوم بالتشفير للحماية من الضجيج (code de convolution) و أيضا تقوم بتوفير تشفير غولد (code de brouillage Gold)). كلتا البنيتين المقترحتين تتمتعان باستهلاك منخفض جدا للموارد مع ارتفاع وتيرة التشغيل في FPGA.

الكلمات المفتاحية: إعادة التشكيل؛ أبنية عامة، FFT-SDF؛ عامل مشترك، خوارزمية CORDIC؛ parameterization؛ فك فيتربي، مستقبل Rake.

LISTE DES ACRONYMES

2G: Mobile de 2^{ème} Génération
3G: Mobile de 3^{ème} Génération
4G: Mobile de 4^{ème} Génération

A

ACS: Add-Compare-Select
ADC: Analog-to-digital converter
AMS: Analog/Mixed Signal
AP: puissance amplification
ASK: Amplitude Shift Keying
ASIC: Application Specific Integrated Circuit
ASIP: Application Specific Instruction set Processor
ATM: Asynchronous Transfer Mode

B

BMC: Branch Metric Calculation
BOC: Banc d'Opérateurs Communs
BPF: Band-pass filter
BRAM: Block RAM

C

CORDIC: Coordinate Rotation Digital Computer
CRC: Cyclic Redundancy Check

D

DAC: Digital-to-Analog Converter
DSP: Digital Signal Processor
DVB: Digital Video Broadcasting

E

EDGE: Enhanced Data rates for GSM Evolution

F

FDD: Frequency-Division Duplexing
FDM: Frequency Division Multiplexing
FEC: Forward Error Correction
FFT: Fast Fourier Transform
FIFO: First-In-First-Out

FIR: Finite Impulse Response
FPGA: Field Programmable Gate Array

G

GF: Galois Field
GIPS: Giga Instruction per second
GMSK: Gaussian Minimum Shift Keying
GPP: General Purpose Processor
GPRS: General Packet Radio Service
GSM: Global System for Mobile communications

H

HW: Hardware

I

IF: Intermediate Frequency
IFFT: Inverse Fast Fourier Transform
IP: Intellect Property
IOE: Input/Output Element

L

LAB: Logical Array Bloc
LFSR: Linear Feedback Shift Register
LNA: Low Noise Amplifiers
LO: local oscillator
LSB: Least Significant Bit
LTE: Long Term Evolution
LUT: Look Up Table

M

M: Multiplier
MAC: Media Access Control
M-FSK: M-ary Frequency Shift Keying
MIPS: Million Instruction per Seconde
M-PSK: M-ary Phase Shift Keying
MPSoC: Multiprocessor System-on-Chip
MSB: Most Significant Bit
MUX: Multiplexer

N

NoC: Network-on-Chip

Q

OC: Opérateur Commun

OFDM: Orthogonal Frequency Division Multiplexing

OL: Oscillateur Local

OSI: Open System Interconnection

P

PCI: Peripheral Component Interconnect

PE: Processing Element

PLL: Phase-Locked Loop

Q

QAM: Quadrature Amplitude Modulation

QPSK: Quadrature Phase Shift Keying

R

R: Register

RF : Radio fréquence

RI: Radio Intelligente

RL : Radio Logicielle

RS: Reed-Solomon

RTL: Register Transfer Level

S

SDF: Single-path Delay Feedback

SDR: Software Defined Radio

SIMD: Single Instruction stream, Multiple Data stream

SMM: Survivor Memory Management

SoC: System-on-Chip

SW: Software

U

UIT : Union International de Télécommunications.

ULB: Ultra Large bande

UMTS: Universal Mobile Telecommunications System

V

VCO: Voltage Controlled Oscillator

VHDL: Very High Speed Integrated Circuit Hardware Description Language

VLSI: Very-Large-Scale Integration

W

W-CDMA: Wideband Code Division Multiple Access

WIFI: Wireless Fidelity ou IEEE802.11

WIMAX: Worldwide Interoperability For Microwave Access

LISTE DES FIGURES

N°	Titre de la figure	Page
1.1	Principaux systèmes de communication sans-fil grand public	12
1.2	Le modèle de référence OSI	13
1.3	Représentation d'un récepteur superhétérodyne.	17
1.4	Vision idéale de Mitola: la radio logicielle idéale	20
1.5	Représentation d'un récepteur de SDR	22
1.6	Adaptation de la modulation dans un système WiMAX [28]	24
1.7	Le cycle de cognition de Mitola et Maguire [28].....	25
1.8	Le cycle de cognition simplifié [28]	25
1.9	Modèle OSI simplifié pour la radio cognitive	26
2.1	Réseau d'antenne patch configurable	31
2.2	L'antenne agile en diagramme de rayonnement proposée dans [35]	32
2.3	Filtre passe bande du second ordre avec transistors MOS	34
2.4	Linéarisation par pré-distorsion	36
2.5	Schéma fonctionnel d'un CAN.....	39
2.6	Synoptique simplifié d'un convertisseur numérique/analogique N bits	41
2.7	Les différentes opérations et traitements en bande de base	41
2.8	Exemple d'architecture de FPGA (Architecture, Interconnexions et CLB).....	45
2.9	Cycle de programmation d'un FPGA	46
2.10	Architectures CPU et GPU	50
2.11	Structure possible d'une radio logicielle	53
2.12	Fonction de modulation paramétrable	56
2.13	Vers une chaîne de transmission multi-standard unifiée.....	57
2.14	Chaîne de transmission multi-standard unifiée.....	58
3.1	Conception de l'unité de désétalement	65
3.2	L'architecture RTL pipeline d'un multiplexeur (MUX 4:1) pour l'unité de désétalement.....	66
3.3	Une architecture pipeline FFT-SDF (N=1024 points).....	68
3.4	Adition et soustraction complexe	69
3.5	Schéma de la multiplication par j	70
3.6	Multiplication complexes: (a) Multiplication par w_8^1 (b) Multiplication par w_8^3	70
3.7	Implémentation matériel de la multiplication avec $\sqrt{2}/2$	71
3.8	Multiplicateur complexe avec trois multiplications réelles et cinq additions réelles	71
3.9	L'architecture RTL pipeline pour le multiplicateur complexe	72
3.10	Cellule de base d'un processeur CORDIC	74
3.11	Cellule de base d'un processeur CORDIC avec la modification de [93].....	75
3.12	Structure Butterfly Commun.....	76
3.13	L'opérateur universel proposé pour l'unité de désétalement, processeur élémentaire PE de FFT et	77

la Cellule Cordic.....	80
3.14 Les trois modes de fonctionnement de la cellule proposée et ses ressources: (a) Mode FFT, (b) Mode désétalement, (c) Mode cellule Cordic.....	80
3.15 Schémas RTL de l'implémentation sur FPGA Virtex V de : a) RMAC [95] et b) CO [3].....	83
3.16 Schémas RTL de l'implémentation sur FPGA Virtex V de l'architecture proposée.....	84
3.17 Structure globale d'un décodeur de Viterbi.....	88
3.18 L'opérateur universel proposé pour l'unité de désétalement, processeur élémentaire PE de FFT papillon, la Cellule Cordic, et le Radix 2-FFT.....	90
3.19 L'opérateur universel proposé adapté pour l'unité ACS de Viterbi.....	92
3.20 Illustration de la mise en œuvre de la technique de l'opérateur commun par la synthèse architecturale.....	93
4.1 Structure d'un modulateur QAM numérique	98
4.2 Architecture de l'accumulateur de phase	99
4.3 Architecture du modulateur numérique QAM proposé	100
4.4 Conversion des trames au format QAM 1024.....	101
4.5 Propriété de symétrie dans la constellation QAM	102
4.6 L'architecture pipeline de l'additionneur	104
4.7 Parallélisation de l'architecture proposée pour la modulation M-QAM	106
4.8 Architecture universelle pour la modulation M-QAM, M-PSK et B-FSK	108
4.9 Les deux architectures de LFSR	111
4.10 Codeur convolutif MTO pour $m = 3$	112
4.11 Codeur convolutif OTM pour $m = 3$	113
4.12 Turbo-code parallèle	113
4.13 Principe du compteur LFSR	114
4.14 Architecture générique proposée	115
4.15 Les deux architectures proposées LFSR configurables	117
4.16 Technique de reconfiguration du registre LFSR proposé	117
4.17 Durées des états haut et bas pour un signal non déphasé, et un signal avec un déphasage.....	119
4.18 Schémas RTL de l'implémentation sur FPGA Virtex 5 du codeur	122
4.19 Schémas RTL de l'implémentation sur FPGA Virtex 5 de LUT de phase et le LUT d'amplitude.....	122
4.20 Schémas RTL de l'implémentation sur FPGA Virtex V du CORDIC de 8 étages.....	122
4.21 Schémas RTL d'un étage d'algorithme de CORDIC proposé par [93]	123
4.22 Schémas RTL de l'additionneur pipeline	123
4.23 Résultats de simulation de l'additionneur pipeline	124
4.24 Résultats de simulation Résultats de simulation du CORDIC de 8 étages	125
4.25 Schémas RTL de l'implémentation sur FPGA Virtex V de l'architecture proposée	126
4.26 Résultats de simulation dans le mode démodulation	130
4.27 Résultats de simulation dans le mode codage	131

LISTE DES TABLEAUX

N°	Titre du tableau	Page
1.1	Caractéristiques des principaux systèmes de communications sans fil.....	14
2.1	Comparaison entre les principaux CAN de télécommunication	40
2.2	Comparatif des approches de conception des multi-traitements	60
3.1	Opérations entre les échantillons d'entrée pour différentes valeurs de code.	65
3.2	Comparaison de l'implémentation de l'architecture proposée avec l'architecture Velcro sur FPGA Virtex-V (XC5VLX50T).....	81
3.3	Comparaison de l'implémentation sur FPGA Virtex-V (XC5VLX50T) de l'architecture proposée avec les travaux précédents	85
4.1	Résultats de synthèse de l'architecture proposée	123
4.2	Polynômes générateurs de codes Convolutifs.....	127
4.3	Polynômes générateurs de codes Convolutifs ($R=1/2$).....	127
4.4	Comparaison de l'architecture proposée avec l'architecture Velcro	128

Table des matières

Remerciement	II
Résumé	III
Abstract	IV
Liste des Acronymes	VI
Liste des figures	X
Liste des tableaux	XII
Introduction générale	1
Chapitre I: Prolifération des normes et standards: Vers un terminal multistandard.	7
1. Introduction	7
2. Définition d'un système de transmission sans fil.....	8
2.1. Système de radiocommunication.....	8
2.2. Évolution des standards et normes de communication.....	8
3. Modèle de référence (OSI).....	12
4. Vers des architectures de plus en plus flexibles et versatiles (multistandards).....	13
4.1. Les exigences d'un terminal multistandard	16
4.2. Les intérêts d'un terminal multistandard flexible.....	16
5. Solutions pour la radio flexible/reconfigurable.....	16
5.1. Radio classique (matérielle).....	17
5.2. Radio reconfigurable.....	19
5.2.1. La radio logicielle (Software Radio).....	19
5.2.2. La radio logicielle restreinte (Software-Defined Radio - SDR).....	21
5.3. La radio cognitive (La radio intelligente).....	22
5.3.1. Terminal de radio cognitive.....	23
5.3.2. Relation entre la radio cognitive et la radio logicielle restreinte.....	24
5.3.3. Cycle de cognition.....	25
5.3.4. Modèle OSI de la radio cognitive.....	26
5.3.5. Radio opportuniste.....	27
6. Conclusion.....	27
Chapitre II : Description détaillée d'une Software-Defined- Radio (SDR)	29
1. Introduction	29
2. Le frontal RF (radio fréquence) d'une SDR.....	30
2.1. Antenne	30
2.1.1. Antennes agiles en fréquence	30
2.1.2. Antennes Agiles en Diagramme de Rayonnement	32
2.1.3. Antennes Agiles en Polarisation	32
2.2. Filtres analogiques.	33
2.2.1. Les filtres passifs LC intégrés.....	33
2.2.2. Filtres actifs à simulation d'inductance.....	34

2.2.3.	D'autres filtres actifs agiles.....	35
2.3.	Les amplificateurs à faible bruit et les amplificateurs de puissance	35
2.4.	Les mélangeurs (Mixers: Up-converters et Down-Converters).....	37
2.5.	Synthétiseur de fréquence.....	37
2.6.	Oscillateurs locaux agiles.....	38
3.	Convertisseurs analogique/numérique et numérique/ analogique.....	38
3.1.	Convertisseurs analogique/numérique (CAN).....	38
3.2.	Convertisseurs numérique/analogique (CNA).....	40
4.	Le frontal numérique (Traitement en bande de base).....	41
5.	Les plateformes de la radio reconfigurable.....	43
5.1.	Composants matériels.....	43
5.1.1.	Les ASIC (Application-Specific Integrated Circuit).....	43
5.1.2.	Les FPGA (Field-Programmable Gate Array).....	44
5.2.	Composants logiciels.....	47
5.2.1.	Les DSP	47
5.2.2.	Les processeurs à usage général (GPP).....	48
5.2.3.	L'accélération matérielle: GPU (Graphics Processing Unit).....	50
5.3.	Analyse des plateformes de la radio reconfigurable.....	51
6.	Solutions hybrides (hétérogènes).....	52
7.	L'approche de paramétrisation.....	54
7.1.	Fonctions communes (FC).....	55
7.2.	Opérateurs communs (OC).....	58
8.	Notre solution préconisée dans le travail de thèse	59
9.	Conclusion.....	61
	Chapitre III : Architecture numérique versatile pour un terminal mobile	63
1.	Introduction	63
2.	Récepteur Rake et l'unité de désétalement.....	64
2.1.	Proposition d'une architecture pipeline pour les deux multiplexeurs 4:1	66
3.	L'architecture de FFT-SDF.....	67
3.1.	Processeur Butterfly Radix-2/4/8.....	68
3.2.	Addition et soustraction complexe.....	69
3.3.	Multiplication Complexe.....	70
3.4.	Architecture du multiplicateur complexe proposée.....	72
4.	L'Algorithme CORDIC.....	72
5.	Opérateur universel pour la FFT, l'unité de désétalement et l'algorithme Cordic.....	75
6.	Implémentation sur FPGA, résultats et comparaisons de la performance	80
7.	Discussion.....	87
8.	Conclusion.....	94
	Chapitre IV : Architecture numérique et générique pour la transmission de données	96
1.	Introduction	96
2.	Traitement numérique vs traitement analogique	97

3.	Architectures génériques pour la modulation numérique.....	97
3.1.	Un modulateur numérique M-QAM.....	97
3.2.	Architecture proposée pour un modulateur QAM numérique.....	99
3.2.1.	Le Codeur.....	101
3.2.2.	Le LUT d'amplitude et le LUT de phase.....	102
3.2.3.	L'accumulateur de phase.....	103
3.2.4.	Additionneur pipeline proposée pour le terme $(\omega t + \varphi_k)$	103
3.2.5.	Générateurs de forme d'onde (Générateurs sinusoïdes).....	104
3.2.6.	Principe de fonctionnement de l'architecture proposée.....	105
3.3.	Parallélisation de l'architecture proposée pour la modulation M-QAM.....	105
3.4.	Architecture générique pour la Modulation QAM, PSK et FSK.....	106
4.	Architecture générique pour la démodulation numérique et le codage.....	110
4.1.	Registres à décalage à rétroaction linéaire (LFSR).....	110
4.2.	Le LFSR dans le codage de canal (code de convolution et turbo code).....	111
4.2.1.	Architecture MTO (Many To One).....	112
4.2.2.	Architecture OTM (One To Many).....	113
4.2.3.	Un turbo-code.....	113
4.3.	Le LFSR pour le générateur de codes d'embrouillage dans le WCDMA.....	114
4.4.	LFSR pour la démodulation numérique.....	114
4.5.	Architecture générique proposée pour le codage et la démodulation numérique	114
4.5.1.	Un registre LFSR reconfigurable comme un opérateur commun.....	116
4.5.2.	Fonctionnement de l'architecture flexible.....	118
4.5.2.1.	Mode démodulation numérique.....	118
4.5.2.2.	Mode codage.....	120
5.	Implémentation sur FPGA.....	121
5.1.	Implémentation de l'architecture générique proposée pour la modulation numérique.....	121
5.2.	Implémentation de l'architecture générique pour le codage et la démodulation numérique.....	125
6.	Conclusion.....	132
	Conclusion et perspectives	133
	Références	138
	Liste des Publications	147
	Annexe A : Système de transmission WCDMA.....	149
	Annexe B : Système de transmission OFDM.....	157
	Annexe C : Conversion d'un signal modulé en un signal binaire.....	162

Introduction générale

Durant ces dernières années nous faisons face à une mutation rapide et profonde des technologies de communications, notamment des réseaux sans fil. Cette mutation est dictée par le besoin incessant de débits des données croissants liés aux interactions et échanges de données entre senseurs interconnectés en réseau et selon un spectre de fréquences déjà bien encombré et non extensible. Cela n'est possible que par l'emploi de nouvelles techniques de transmission adaptatives et de partage de canaux particulièrement sophistiqués (ultra-large bande, approches MIMO, codages de multiplexage et codages correcteurs d'erreurs plus complexes, communications multistandards à très faibles niveaux de puissance de transmission), pour lesquelles les architectures existantes ne sont plus adaptées. Les contraintes correspondantes en termes de capacité de traitement temps réel, de flexibilité et versatilité, de faible consommation, d'indépendance envers les technologies matérielles employées, ainsi que l'orientation vers la « radio logicielle » constituent de réels défis que seules de nouvelles approches architecturales sont en mesure de relever.

En plus nous assistons actuellement à la multiplication des normes et des standards de télécommunications. Le nombre croissant des standards normalisés permet d'élargir l'éventail des offres et des services disponibles pour chaque application et structure de communication. Cependant si cette prolifération des normes et standards est considérée comme un avantage pour les utilisateurs, elle représente pour les concepteurs un défi technique qui est: comment pouvons-nous fournir à l'utilisateur un terminal *unique* capable de naviguer entre les multiples standards de radiocommunication.

En effet, différents par les services proposés, les standards se distinguent par les bandes fréquentielles utilisées, les modulations, les codages et un ensemble de paramètres qui est spécifique à chacun d'eux. En somme la réalisation d'un terminal multistandard ou nœud communicant intelligent ("*smart adaptive node*") permettant une fonctionnalité et adaptabilité de communication à usage universel au sein d'un même réseau devient indispensable.

La méthode dite "Velcro" est l'une des possibilités pour la réalisation d'un nœud de réseau multistandards, qui consiste en la coexistence de toutes les entités architecturales nécessaires et d'une structure de type "routeur" ou "Switch", permettant de passer d'un mode à l'autre ou d'une fonction à une autre selon une approche de reconfigurabilité technologique. Cependant son évolutivité est limitée par les standards considérés et la complexité de sa mise en œuvre reste maximale selon les réseaux actuels [1][2][3]. En

effet, la complexité de la solution classique correspond à la somme des complexités de chaque standard à laquelle s'ajoute celle des organes de reconfiguration [2].

Une autre alternative est la mise en œuvre de radios dites logicielles. L'atout de la Radio logicielle est de pouvoir ajouter, mettre à jour des fonctionnalités du système radio par logiciel grâce à la reprogrammabilité ou la reconfigurabilité de structures micro-programmées sans ajouter les coûts de changement du matériel. Le signal radio fréquence capté par une antenne large bande est directement converti en numérique par un convertisseur analogique/numérique qui travaille à une haute fréquence d'échantillonnage, puis ce signal numérique est traité par un processeur très rapide capable de prendre en charge les traitements numériques aux différents standards par simple téléchargement d'un logiciel ou d'un code. Ce type de radio logicielle baptisé « idéal » est loin d'être réel, en effet il existe de nombreuses limitations technologiques dans les processeurs et les convertisseurs analogiques/numériques actuels. Les processeurs génériques présents dans le marché montrent une faible puissance de calcul et une grande consommation de puissance, alors les convertisseurs analogiques/numériques présents aussi dans le marché ne sont pas capable de traiter des signaux de grande fréquence comme celles des systèmes de transmissions sans fil actuels. Ces limitations ne permettant pas de réaliser de la radio logicielle idéal. Ce qui nous oblige de ne pas directement numériser le signal issu de l'antenne sans faire diminuer leur fréquence par des composants analogiques. En plus, les limitations des processeurs génériques nous obligent d'envisager l'utilisation d'autres composants numériques dans la partie bande de base, qui montrent une puissance de calcul très grande et une consommation de puissance plus faible.

Les FPGA constituent une cible technologique très intéressante de par leurs caractéristiques dont certaines des plus importantes sont le faible coût, grande fréquence de fonctionnement et la possibilité de les reconfigurer in-situ. Ces cibles technologiques permettent notamment d'intégrer des "soft-cores" microprogrammés au sein d'une structure configurable logique pouvant ainsi mettre en œuvre simultanément une solution reconfigurable « radio logicielle ». La reconfiguration dynamique des FPGA, peut implémenter n'importe quel algorithme sur la même surface logique. Ceci va réduire la surface utilisée dans le FPGA, et réduire le nombre de circuits, poids et volume lors de la conception de systèmes embarqués. Cependant le chargement d'un code d'un nouveau algorithme puis d'envoyer des bitstreams pour reconfigurer le terminal peut affecter le fonctionnement temps réel d'un terminal mobile. La recherche de solutions généralisables et facilement réutilisables recommande des approches facilement reconfigurables.

Une chaîne de communications numériques intégrables sur puce, supportant différentes normes et standards de communication réseau, emploie des opérations de traitement des

signaux typiques telle que la modulation, le codage de canal, l'égalisation, etc... Ces opérations ont de nombreuses similarités entre elles [4][2][3][5]. Ces similarités peuvent être exploitées pour proposer de nouvelles architectures numériques génériques et «universelles», capables de se reconfigurer par un téléchargement de simples paramètres. C'est le principe de la paramétrisation [6] [7], en effet cette technique avec l'approche des opérateurs communs (OC) [7][8] vise une implémentation multistandards plus efficace et reconfigurable par un simple chargement de paramètres tout en limitant le nombre d'éléments physiques à implémenter. Les blocs génériques ou les opérateurs communs identifiés peuvent effectuer des opérations sans savoir leurs applications. Par conséquent, ces opérateurs peuvent être adaptables et évolutifs à n'importe quel standard et leurs réutilisations deviennent très optimales.

Problématique et objectif de la thèse :

De nos jours, il existe une multitude de standards et de normes dans la transmission numérique dont les architectures numériques pour la transmission sans fils de données sont différentes les unes des autres. De plus pour un même standard ou norme on y trouve des architectures distinctes. L'objectif principal de notre travail est de mettre en place une architecture tout numérique, générique à usage universel capable de prendre en charge des fonctionnalités précédemment gérées par des circuits différents:

- cette architecture peut être générique dans le même standard.
- cette architecture peut être générique entre plusieurs standards.
- cette architecture doit être tout numérique.

Au vu de cet objectif une question principale se pose: comment intégrer une multitude de protocoles et standards, sachant que pour chacun d'entre eux on y trouve plusieurs solutions et architectures, sur un même module ? La réponse à cette question nécessite sa décomposition en questions secondaires:

- a- la diversité des solutions engendre une différence au niveau des composants de base. Comment résoudre les conflits issus d'une densité d'intégration très élevée sur une même puce miniature?
- b- certains modules importants dans la chaîne de transmission sont encore analogiques. Comment on peut les faire passer dans le domaine numérique ?

Afin de résoudre les conflits de la densité d'intégration très élevée, on propose de rechercher des points de similarité entre les différentes architectures étudiées, en suivant la technique de la paramétrisation [6] [7] et l'approche des opérateurs communs (OC) [7][8]. Cette recherche des points de similarités permet de définir un cahier des charges concernant les caractéristiques principales de l'architecture générique à réaliser. Pour réussir l'intégration tout en assurant un fonctionnement optimal, l'idée est d'exploiter ces

points de similarité pour mettre en place des architectures flexibles et versatiles «universelles», capables de prendre en charge des fonctionnalités précédemment gérées par des circuits différents dans un contexte de réseau de communication adaptatif. Celles-ci devront, en outre, être suffisamment flexibles pour supporter des normes de transmissions multiples et s'adapter facilement aux évolutions futures dans le cadre de communications en réseaux de systèmes sur puce à base de capteurs. Ainsi pour faire la numérisation des architectures qui sont encore analogiques on propose des architectures qui traduisent directement les fonctionnalités souhaitées en un algorithme adapté aux circuits numériques, sans passer par une étape analogique susceptible d'introduire des contraintes en contradiction avec la démarche de conception de ces circuits.

Contributions

Les contributions réalisées dans ces travaux de thèse sont tout à fait originales, en effet elles s'inscrivent totalement dans le domaine du développement expérimental, de la conception architecturale et de l'implantation d'architectures numériques avancées pour la transmission réseau sans fils à très haut débit de données audio et vidéo (signaux, paroles et images) sur une technologie reconfigurable de type FPGA (Field Programmable Gate Array), et permettant la mise en œuvre d'une versatilité robuste de modulations, protocoles et codages selon des compromis entre ressources et performances des communications des systèmes embarqués à contrainte temps réel et tout en répondant à des contraintes d'embarquabilité.

Pour atteindre ces objectifs, le principal apport et la nouveauté dans le présent document peuvent être résumées comme suit:

- la définition détaillée d'une chaîne de transmission radio logicielle.
- la proposition d'une nouvelle architecture reconfigurable qui implémente des algorithmes largement utilisés dans les systèmes de communications sans fil. À cette fin, nous avons identifié les similarités entre les unités de désétalement dans le récepteur Rake, le processeur élémentaire dans l'algorithme FFT-SDF (Fast Fourier Transform- Single path Delay Feedback), et la cellule d'un processeur Cordic, puis proposer un opérateur générique commun entre les trois algorithmes couramment utilisés. L'opérateur générique proposé est basé sur une structure RTL pipeline pour offrir le meilleur compromis entre les ressources, les performances et la consommation de puissance.
- l'architecture proposée est capable de prendre facilement en charge d'autres algorithmes de la FFT ainsi la fonction de décodage de Viterbi.
- la proposition d'une architecture pipeline pour le multiplexeur et le multiplicateur complexe peuvent atteindre une plus grande fréquence de fonctionnement avec une consommation de ressource faible.

- la proposition d'une nouvelle approche qui consiste à combiner deux techniques : l'opérateur commun (OC) et la synthèse architecturale (SA).
- la proposition d'une architecture versatile et numérique pour plusieurs types de modulations numériques tel que la PSK, FSK et la QAM.
- la proposition d'une nouvelle architecture universelle utilisée pour la démodulation de phase (PSK), démodulation de fréquence (FSK), la génération de code de brouillage (code Gold utilisé dans les systèmes WCDMA) et le codage de canal.
- la proposition d'un registre LFSR reconfigurable par des simples paramètres.

Cadre Conceptuel

Ce sujet entre dans le cadre des thématiques du LERICA (Université Badji Mokhtar, Annaba) et de l'équipe ASEC du laboratoire LCOMS (Université de Lorraine, Metz) en développant des systèmes de traitement numérique pour la transmission fiable et à haut débit de données sur des réseaux numériques, notamment sans fils. L'équipe ASEC possède une longue expérience dans le domaine de la conception d'architectures purement numériques à base de technologie adaptative par reconfiguration dynamique pour la modulation, le codage canal et le codage correcteur d'erreurs pour des transmissions sur différents supports (filaire, sans-fils et fibre optique). Ces travaux portent sur la réalisation de systèmes microélectroniques numériques embarqués selon une approche structurale SoC (système sur puce) pour les applications embarquées fortement communicantes où l'adaptabilité et la flexibilité sont des paramètres essentiels.

Plan du mémoire

Ce manuscrit se décompose en quatre chapitres (deux chapitres introductifs suivis de deux chapitres présentant nos contributions). Ces chapitres se répartissent comme suit :

Le premier chapitre donne au lecteur un aperçu sur le contexte de notre travail de recherche. Nous commençons par une mise en évidence de la problématique et des exigences d'un terminal multistandards en mettant l'accent sur la diversité des normes et standards de radio communication et en présentant la solution de la radio logicielle. Ce chapitre propose un état de l'art sur les technologies candidates à la constitution d'un système radio logicielle où un aperçu détaillé sur la Radio Logicielle, la Radio Logicielle Restreinte et la Radio Intelligente fait l'objet de la deuxième partie de ce chapitre.

Le deuxième chapitre présente successivement un état de l'art sur la chaîne de transmission d'une radio logicielle restreinte. Dans la première partie de ce chapitre il nous a semblé important de donner au lecteur un aperçu sur les différents éléments constitutifs d'une radio logicielle restreinte en commençant par l'antenne jusqu'à la partie bande de base. Dans la seconde partie de ce chapitre, les plateformes de la radio

reconfigurable disponible seront discutées. Ensuite, nous avons fait le choix de présenter particulièrement la technique de paramétrisation et l'approche des opérateurs communs, qui ont fait l'objet de nos recherches durant cette thèse.

Le troisième chapitre propose une architecture générique et universelle capable de prendre en charge les trois fonctions : l'unité de désétalement pour le récepteur Rake, le processeur élémentaire (PE) pour l'algorithme FFT-SDF (Fast Fourier Transform Single path Delay Feedback) et la cellule de l'algorithme Cordic. À cette fin, nous avons identifié les similarités entre les trois architectures couramment utilisées pour proposer un opérateur générique basé sur une structure RTL pipeline. Nous discutons à la fin de ce chapitre les comparaisons entre notre architecture proposée et d'autres cellules proposées dans la littérature. Ainsi, nous montrons que notre architecture peut facilement prendre en charge d'autres fonctionnalités comme le décodage de Viterbi.

Dans le quatrième chapitre, après une synthèse des architectures numériques pour la modulation et le codage canal, nous proposons d'abord une architecture générique et universelle pour la modulation numérique en utilisant la technique de paramétrisation. Ensuite nous proposons une nouvelle architecture universelle utilisée pour quatre fonctionnalités : la démodulation numérique des signaux modulés en phase et en fréquence, la génération de code de brouillage connu sous le code Gold et le codage de canal. Nous discutons ainsi à la fin de ce chapitre la consommation en ressource et la performance dans le FPGA des architectures proposées.

Nous concluons notre travail par une synthèse des différents travaux présentés dans ce document. La synthèse sera suivie d'une discussion sur les principales contributions apportées. Enfin, nous ouvrirons des perspectives sur les travaux futurs.

Chapitre I

Prolifération des normes et standards: Vers un terminal multi-standards

1. Introduction

Aujourd'hui, les multiples normes et standards de télécommunications envahissent nos vies quotidiennes. En effet, à cause de cette prolifération des normes et standards, nous parlons de la notion de Handover vertical qui s'ajoute au Handover horizontal déjà existant. Si le Handover horizontal qualifie la capacité d'un terminal à "naviguer" entre les différentes zones cellulaires d'un même standard, le Handover vertical qualifie la capacité d'un terminal à "naviguer" entre les différents réseaux et standards. Cependant si cette nouvelle notion permet d'élargir l'éventail des offres et des services pour les utilisateurs, elle représente pour les concepteurs un défi technique qui est le suivant: comment pouvons-nous fournir à l'utilisateur un terminal unique capable de naviguer entre les multiples standards de radiocommunication ? La première solution qui vient à l'esprit est l'utilisation dans un même terminal des composants dédiés à chacun de ces standards, cette technique s'appelle la technique «Velcro». Celle-ci, présente deux inconvénients majeurs qui sont: l'augmentation des coûts de fabrication et elle n'autorise pas la prise en compte de nouveaux standards et normes. La radio logicielle a été créée pour répondre à ces deux problèmes de la technique « Velcro ». Dans la première partie de ce chapitre, et après une introduction au concept du réseau sans-fil, le document s'attarde sur la diversification des normes et standards. Ensuite, et dans le but de proposer des architectures génériques et universelles utilisables pour plusieurs standards et normes, un aperçu détaillé sur la Radio Logicielle, la Radio Logicielle Restreinte et la Radio Intelligente fait l'objet de la deuxième partie de ce chapitre.

2. Définition d'un système de transmission sans fil

Un réseau sans fil (en anglais Wireless network) est un réseau dans lequel au moins deux terminaux peuvent communiquer sans liaison filaire. L'avantage essentiel que représentent les réseaux sans fil est la mobilité. Celle-ci permet aux usagers de communiquer sur une zone plus ou moins étendue et de pouvoir poursuivre une communication tout en se déplaçant avec toutefois, des limites en vitesse et en distance. Celle-ci permet aussi aux fournisseurs d'éviter la complexité d'installation de tels réseaux (creusement pour acheminer les câbles, câblage, connecteurs, équipements lourds des bâtiments etc. comme dans le cas des réseaux filaires).

Les principaux paramètres qui permettent de définir les exigences d'un système de communication sans fil sont: la distance entre l'émission et la réception, le débit de la communication c'est-à-dire la quantité d'informations à transmettre et enfin le type de communication qui peut être un transfert de données comme dans l'internet ou une communication vocale en télécommunication mobile.

2.1.Système de radiocommunication

Les systèmes de radiocommunications sont apparus à la fin du XIX^{ème} siècle avec l'invention du téléphone filaire (fixe) par le Canadien G. BELL et E. GRAY en 1876, puis la découverte des ondes radios par le physicien allemand H. HERTZ en 1878 et l'établissement par Guglielmo Marconi des premières liaisons radio en 1895 sur un peu plus de 2 km et en 1901 à travers l'Atlantique [9] [10].

Un système de radiocommunication est basé sur l'utilisation d'ondes radioélectriques pour transmettre des informations entre deux points distants. L'onde radio (ou onde radio électrique ou aussi onde hertziennes) est une onde électromagnétique dont la fréquence est inférieure à 300 GHz soit une longueur d'onde dans le vide supérieure à 1 millimètre. Afin d'éviter toutes formes d'anarchies des communications par ondes radios, leur utilisation est régie par l'UIT (*Union Internationale des Télécommunications*). On parle de la réglementation du domaine des radiocommunications. Cette organisation affecte les bandes de fréquences à des services différents tels que la transmission de données, la télémesure, la télécommande, la téléphonie, la radiodiffusion sonore, la télévision (vidéo), etc. Par conséquent, le spectre des fréquences a été découpé en plusieurs bandes de fréquences dont les noms sont normalisés [10] [11]. Ainsi, chaque bande est découpée en plusieurs sous-bandes fréquentielles, où chacune de ces sous-bandes est réservée à une utilisation spécifique.

2.2. Évolution des standards et normes de communication

Depuis les premières expériences de Marconi à la fin du XIXe siècle, les communications sans fil entre deux éléments mobiles ont été limitées à des applications professionnelles et

ce, jusqu'à la fin des années soixante dix. Il a fallu attendre les années quatre vingt avant de voir la naissance des systèmes de téléphonie mobile cellulaire, de première génération, entièrement analogiques comme l'AMPS (*Advanced Mobile Phone Service*). Néanmoins, les ressources spectrales réduites, la courte durée de vie des batteries et le coût du terminal, ont été des obstacles qui ont restreint les débuts du développement des téléphones portables de première génération.

La seconde génération des systèmes cellulaires est apparue au début des années 90 avec le GSM (*Global System for Mobile communications*, Europe) [12], le PDC (*Personal Digital Cellular*, Japon), et le PCS (*Personal Communication System*, États-Unis). Par rapport à la norme de première génération, cette norme a permis l'essor des communications mobiles en plus, les performances comme l'efficacité spectrale, la sécurité et la fonctionnalité sont les plus importantes. Dans le cas de GSM, le faible débit de ce standard permet principalement une émission-réception vocale, ou des services simples comme les SMS et MMS (prévus initialement pour des applications de transport de la parole et de données à faibles débits (9600 bits/s) autour des fréquences (935-960 MHz) ou (890-915 MHz)).

Toutefois, la demande de nouveaux services de transmission de données plus rapides a été vite reconnue et la norme GSM a été dotée de nouvelles fonctionnalités comme le HSCSD (High Speed Circuit Switched Data), GPRS (General Packet Radio Service) et EDGE (Enhanced Data Rates for GSM Evolution). Le HSCSD est basé sur la commutation de circuits qui n'a pas eu de succès commercial et a été abandonné. Le GPRS est basé sur la commutation de paquets qui a été introduite à large échelle en 2002, il est considéré comme des systèmes d'une génération nommée 2.5G. L'EDGE, est une norme pour le téléphone mobile, il représente l'évolution du GPRS et qui est présenté comme la génération 2,75G.

Après l'apparition de ces nouveaux standards, l'ITU-R (International Telecommunications Union-Regulatory) a cherché à établir un nouveau standard encore plus performant. Le standard UMTS (Universal Mobile Telecommunications System) fut créé. La troisième génération (3G) de systèmes cellulaires, a été principalement conçue et imaginée afin de satisfaire l'acheminement de données multimédia (accès à Internet, aux réseaux d'entreprise, à des services de messageries, de visioconférence, de jeux,...). Cette troisième génération de communication mobile est basée sur la technologie W-CDMA et permet un transfert de données important allant jusqu'à 2 Mbits/s théorique [13]. Au cours d'une communication, l'UMTS ajuste sa puissance de sortie en fonction de la distance entre le téléphone et la station, afin d'économiser l'énergie.

L'évolution de l'UMTS est le HSDPA (*High-Speed Downlink Packet Access*) considéré comme le standard 3.5G des télécommunications mobiles. Alors que le débit maximum

permis sur une liaison UMTS est de 2 Mbits/s pour une bande passante de 5MHz, le HSDPA, grâce à sa modulation en 16-QAM, autorise des débits de 10 Mbits/s. Ce dernier a évolué vers L'HSPA+ (*High Speed Packet Access+*), parfois nommée 3G++ ou 3G Dual Carrier ou aussi 3.75G, est une norme de téléphonie mobile permettant d'atteindre des débits théoriques crêtes de 21 Mbit/s par cellule radio en téléchargement.

La norme LTE (Long Term Evolution) est une norme de la quatrième génération (4G parfois on dit 3.9G), qui vise des débits élevés, de l'ordre de 70 Mbps en voie montante (le sens des communications permet de distinguer les voies montantes et descendantes; les premières se réfèrent aux communications depuis le terminal mobile vers la station de base et les secondes évoquent les communications depuis la station de base vers le terminal mobile) pour un canal de 20 MHz jusqu'à 300 Mbps en voie descendante pour une configuration 4x4 MIMO et un canal de 20 MHz (calculs théoriques) [14]. Le LTE fait appel à des technologies récentes de radio transmission (OFDMA en réception, SC-FDMA [14] et MIMO pour l'émission).

Après le LTE, vient le LTE-Advanced qui représente la « vraie » 4G , il est capable de fournir des débits descendants (téléchargement) supérieurs à 1 Gb/s à l'arrêt et à plus de 100 Mb/s pour un terminal en mouvement.

Dans ce même sens, le réseau IEEE802.16 (WiMAX) peut être considéré comme faisant partie de la quatrième génération (4G) des systèmes de téléphonie cellulaire. Le WiMAX permet un débit théorique de 70 Mbits/s sur un rayon de 50 km maximum tout en se basant sur des modulations différentes en fonction de l'état du canal radio et de nouvelles méthodes d'accès au canal.

En plus de tous les réseaux mobiles et leurs générations, nous avons aussi les réseaux locaux sans fil qui font l'actualité dans la vie quotidienne à travers le WIFI, le Bluetooth, etc... La Figure I.1 montre les principaux systèmes de communication sans-fil grand public ainsi que les fréquences de fonctionnement [15].

L'essentiel des systèmes fonctionnant en dessous de 6 GHz peut se résumer en quatre catégories selon le périmètre géographique (appelé zone de couverture) offrant une connexion:

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

- ayant une faible portée: quelques dizaines de mètres autour de l'utilisateur.
- se déplaçant avec l'utilisateur.
- servant à relier des périphériques (imprimante, téléphones portables avec un ordinateur). Ou deux machines très peu distantes.
- n'ayant pas de station relais.

- exemple le Bluetooth : (IEEE 802.15, débit: 1Mbps, bande: entre 2400 et 2483.5 Mhz).

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

- de 50 à quelques centaines de mètres.
- ayant une station relais.
- IEEE 802.11 (standards de USA) (entre 4 et 54 Mbps, bande 2.4GHz à 2,4835 GHz).
- HiperLan (High Performance LAN, European Telecommunications Standards Institute), HiperLan1: débit 20 Mbps, portée: 100 mètres, bande de fréquences = 5.3Ghz et HiperLan2: débit 54Mbps.
- IEEE 802.11a, b, g, n.

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

- IEEE 802.16e (Wimax).
- Débit =110Mbps.
- Portée de 4 à 10 kilomètres.
- Connexion des immeubles par radio à travers une antenne extérieure avec des stations centrales reliées au réseau filaire. Ils permettent aussi des raccordements à des réseaux à large bande dans les secteurs qui ne sont pas servis par le câble.

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

- Ces connexions peuvent être maintenues sur de vastes zones géographiques, comme des villes ou des pays, grâce à l'utilisation de plusieurs sites d'antennes ou de systèmes satellites gérés par les fournisseurs de services sans fil.
- GSM (Global System for Mobile communications).
- GPRS (General Packet Radio Service).
- UMTS (Universal Mobile Telecommunications System).
- TD-SCDMA, et LTE.

La plage de fréquence considérée dans la figure I.1 comprend les systèmes ultra large bande (UWB), le WiMAX fixe (ou IEEE 802.16 a), elle comprend aussi les systèmes de diffusion générale «broadcast» comme: la télévision et vidéo numérique (DVB/T), la

radio numérique (DAB) et le système de positionnement global (GPS). Pour ces derniers, le système se réduit essentiellement à un récepteur.

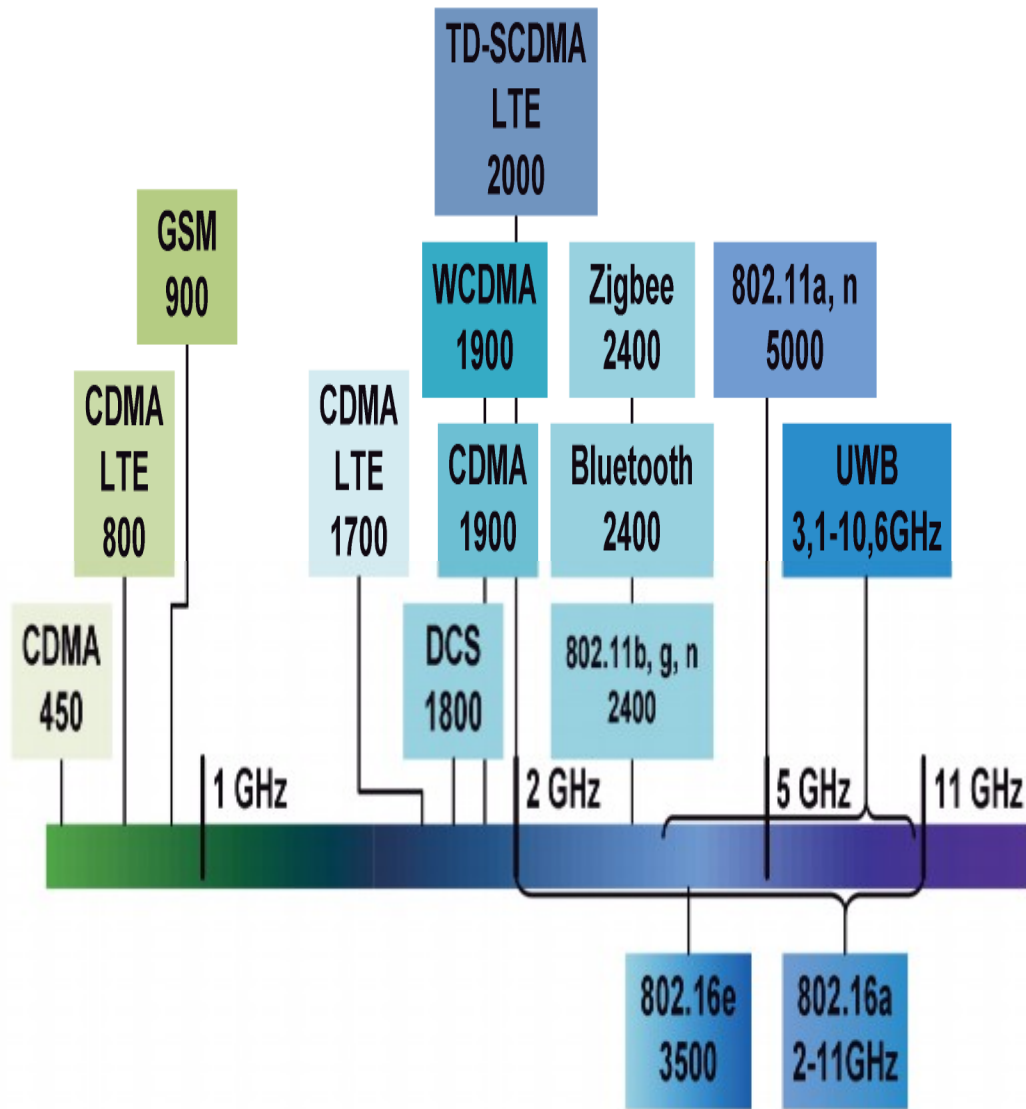


Figure I.1. Principaux systèmes de communication sans-fil grand public [15].

3. Modèle de référence (OSI)

Les normes réseaux actuelles sont divisées selon un modèle en couche, le modèle Open Systems Interconnection (OSI), ce qui se traduit par « interconnexion de systèmes ouverts ». Ce modèle divise l'ensemble des protocoles en sept couches indépendantes entre lesquelles sont définis deux types de relations: les relations verticales entre les couches d'un même système (interfaces) et les relations horizontales relatives au dialogue entre deux couches de même niveau (les protocoles) (figure I.2). Les couches 1,2, 3 et 4 sont orientées en transmission et les couches 5,6 et 7 sont orientées en traitement [16].

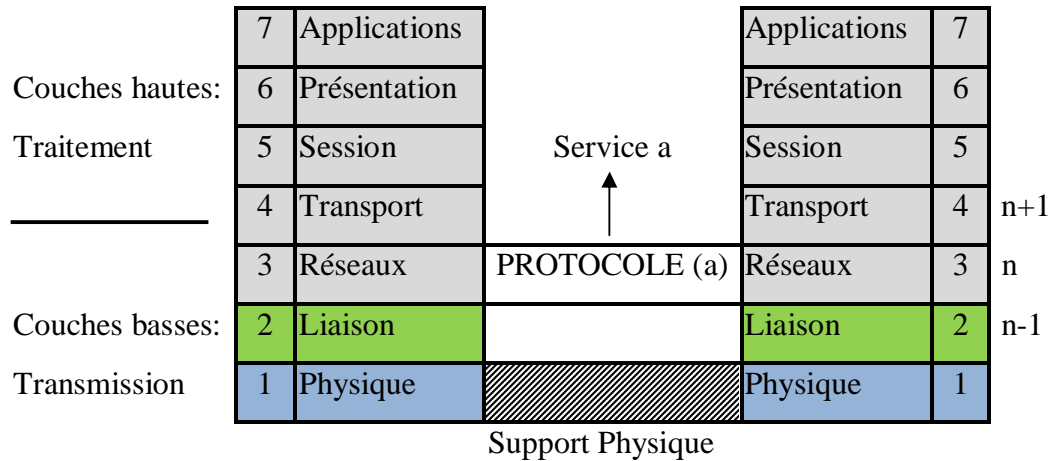


Figure. I.2. Le modèle de référence OSI.

Dans ce travail de thèse, on s'intéresse aux couches basses de ce modèle, qui sont particulièrement la couche physique (appelée PHY) et la couche liaison de données.

La couche physique (1) se charge de la transmission de bits à l'état brut sur un canal de transmission. Les problèmes de conception de cette couche concernent les interfaces électriques, la synchronisation, ainsi que les spécifications du support physique de transmission.

La couche liaison (2) a pour but de transmettre les données sans erreur. Elle décompose les données sur l'émetteur en trames de données et les envoie en séquence. Cette couche s'occupe aussi de contrôler l'accès au canal partagé par une sous-couche MAC (Medium Access Control), surtout dans le cas d'accès multiples.

Ainsi pour les réseaux de communication de nouvelle génération, l'UIT-T (Union Internationale des Télécommunications standardisation du secteur des Télécommunications) a développé un modèle de référence composé de trois plans: le plan utilisateur, le plan contrôle, et le plan gestion. Le plan utilisateur est destiné au transport de l'information des utilisateurs. Le plan contrôle s'occupe de la signalisation. Le plan gestion offre les fonctions de surveillance du réseau, les fonctions de gestion de plan et les fonctions de gestion de couche.

4. Vers des architectures de plus en plus flexibles et versatiles (multistandards).

Il est clair qu'à partir des paragraphes précédents, il existe un nombre important de normes et standards de télécommunication et ceci sans parler des futurs standards. Par conséquent et comme il est dit dans l'introduction de ce chapitre; le défi technique réside dans la conception d'un terminal unique flexible et versatile, capable de naviguer entre les multiples standards de radiocommunication actuels et futurs [4].

Cette tâche semble un peu difficile, car chaque standard se distingue par les services proposés, les bandes fréquentielles utilisées, les modulations, les codages et un ensemble de paramètres qui est spécifique à chacun d'eux. Pour mettre en évidence cette diversité des spécifications des normes radio, nous présentons dans le Tableau I.1, les principales caractéristiques des systèmes de communication sans fil les plus connues.

Tableau I.1 : Caractéristiques des principaux systèmes de communications sans fil.

Groupe	Standard		Tx (MHz)	Rx (MHz)	Technique d'accès	Modulation	Débit
WLAN	WiFi	802.11a	5150-5350		CSMA/CA	OFDM, BPSK, QPSK, 16QAM, 64QAM	54 Mbit/s
		802.11b	2412-2472			DQPSK, DBPSK, CCK	11 Mbit/s
		802.11g	2412-2472			OFDM, BPSK, QPSK, 16QAM, 64QAM	54 Mbit/s
	HiperLAN2		5150-5350	5470-5725	TDMA	OFDM, xQAM, BPSK	54 Mbit/s (/utilisateur)
WPAN	UWB		3100-10600		CDMA	QPSK (MBOA) BPSK, QPSK, OFDM	480 Mbit/s
	Bluetooth		2402-2480		FHSS	GFSK	723 kbit/s
	Zigbee		868-868,6	2412-2472	TDMA	GFSK, O-QPSK	28/250 kbit/s
WMAN	WiMAX		2000-11000		TDMA, OFDMA	BPSK, QPSK, xQAM	70 Mbit/s
WWAN	GSM	Gsm900	880-915	925-960	TDMA	GMSK	14.5 kbit/s
		Dcs1800	1710-1785	1805-1880			115 kbit/s
		Pcs1900	1850-1910	1930-1990			
	EDGE		1710-1785	1805-1880	TDMA, FDMA	GMSK, 8-PSK	384 kbit/s
	UMTS		1920-1980	2110-2170	W-CDMA	QPSK, HPSK	2 Mbit/s
LTE		2500-2700		OFDMA (DL)	QPSK, 16-QAM, 64-QAM	100 Mbit/s	
				SC-FDMA (UL)		50 Mbit/s	

Par conséquent, d'après les spécifications des systèmes de communication sans-fil (standards de communications mobiles, les réseaux sans fils métropolitains, les réseaux

sans fils locaux, les réseaux sans fils personnels, les systèmes de diffusion générale et le système de positionnement global) [11] et comme le montre le tableau I.1, on peut remarquer que les différentes caractéristiques de la couche physique (PHY) et la couche de contrôle d'accès au média (MAC) pour les différentes normes de télécommunication, peuvent-être résumées dans les points suivants:

- **la fréquence de travail**, qui impacte les antennes, les premiers étages analogiques (les amplificateurs de puissance, les filtres analogique, les mélangeurs, les oscillateurs...) et les fonctions de conversion analogique/numérique;
- **la modulation et le codage employés**, peuvent être simples (FSK, GFSK) ou très complexes à mettre en œuvre (OFDM, SC-FDMA, WCDMA, QAM de haute densité);
- **les codes détecteurs et correcteurs d'erreurs**, qui peuvent aller du simple code CRC aux codes les plus complexes (code convolutif, Reed-Solomon, Turbo Codes, LDPC...);
- **la méthode d'accès au média (MAC)**, qui peut se faire par synchronisation avec les autres émetteurs et acquittements des paquets reçus pour éviter les collisions (nécessite des temps de réponse très courts), ou par la détection de collision en utilisant des étages de décodages spécifiques (ceci accroît les besoins de traitement temps réel).

Ces différences dans la couche physique (PHY) et la couche de contrôle d'accès au media (MAC) pour les normes et standards de télécommunications, montrent la nécessité de disposer d'un élément reconfigurable et flexible. Par conséquent, cet élément ne doit pas être figé dans un mode de fonctionnement, ce qui permet la cohabitation de plusieurs systèmes radios dans le même équipement [3] [4] [17]. Par exemple, un terminal mobile est capable sur la même plate-forme de répondre aux besoins de changements de services et/ou changements de réseaux d'accès (changement de l'accès dû à la mobilité, changement du taux d'erreur binaire, changement des traitements de la couche physique).

On peut imaginer donc retrouver dans un terminal de nombreux standards afin d'offrir un accès aux systèmes mobiles (GSM, UMTS, WIMAX et LTE), un accès à un réseau WPAN (Bluetooth) et WLAN (WiFi) et même, de diffusion (DVB-S/-H/-T et DAB) et de positionnement (GPS, Galiléo). Parfois ce terminal est qualifié de terminal «multistandard», d'une architecture générique, flexible et reconfigurable [17].

4.1.Les exigences d'un terminal multistandard

Les exigences d'un terminal flexible multistandard concernent d'abord les aspects fonctionnels qui peuvent se résumer en deux points:

- Le premier point impose qu'un terminal multistandard nécessite des circuits capables de traiter des signaux large bande, à grande plage dynamique et autour de différentes fréquences porteuses, il est décrit comme un système multi-bandes. La notion de multi-bandes est importante pour la partie Front-End RF du système. Cette problématique n'est pas l'objet de cette thèse, bien que nous allions essayer dans le deuxième chapitre de donner plus de clarification sur ce sujet.
- Le deuxième point est que l'architecture matérielle d'un terminal multistandard doit pouvoir supporter les différentes techniques de communication (modulation, codage,...) ainsi que les différentes techniques d'accès (TDMA, FDMA, CDMA, OFDM, OFDMA...) et différents protocoles et services de communication. Cette notion de flexibilité est importante pour la partie bande de base du système qui est nécessaire pour le travail de la thèse.

Ces exigences, concernant les aspects fonctionnels et la mise en œuvre pratique d'un tel concept, nécessitent aussi le respect d'autres exigences en termes de performances, de portabilité, de consommation en puissance et de coût.

4.2.Les intérêts d'un terminal multistandard flexible

En effet, le premier intérêt de cette flexibilité est économique, si on prend l'exemple d'un récepteur unique qui doit avoir la possibilité de recevoir simultanément plusieurs standards par seulement une reconfiguration ou modification des paramètres, il présente une importante réduction de la taille, de la complexité, de la consommation et du prix.

Le deuxième intérêt, est que l'utilisateur possède un terminal universel compatible avec différents standards de radiocommunications [18]. On parle ici de l'intérêt d'exploitation pour l'utilisateur des services radio mobile. Ceci présente un avantage à la fois de confort et d'économie pour l'utilisateur qui aura la souplesse d'utiliser ce terminal universel et unique pour différents réseaux et dans différents pays. De même, ce type de concept favorisera d'avantage l'interopérabilité des différents réseaux radio: cellulaires, diffusion, positionnement, WLL, WLAN,...

5. Solutions pour la radio flexible/reconfigurable

La flexibilité [19] introduit des notions distinctes telles que la reconfigurabilité/reprogrammabilité, l'adaptabilité, la modularité (une définition des termes flexibilité, adaptabilité et reconfigurabilité (FAR) est proposée dans [19] et [20]). Par exemple selon [19] [20] *l'adaptabilité* est qualifiée par un terminal qui a la capacité de s'adapter à l'environnement de travail en faisant varier un ensemble de valeurs connues, comme par

exemple le taux de codage ou la densité de la modulation, alors que **la reconfigurabilité** est qualifiée par un terminal qui a la capacité de faire de grandes modifications de la structure de l'application ou bien des éléments qui la composent.

Actuellement, deux types de démarches sont à l'étude pour la répartition de la flexibilité: l'une est pour le réseau lui-même et l'autre pour le terminal, bien que ce que nous cherchons dans ce travail de thèse, c'est la flexibilité d'un terminal mobile ou son élément. Plusieurs solutions existent pour la radio multistandard ou la radio flexible entre une radio classique toute simple sans composante logicielle, et une radio intelligente.

5.1. Radio classique (matérielle)

Dans un terminal radio classique ou matérielle, la chaîne de transmission prend en entrée un flux de données à transmettre, et fournit en sortie une onde électromagnétique. La chaîne de réception fait l'opération inverse, elle prend en entrée l'onde électromagnétique du domaine radio-fréquentiel (RF), et fournit en sortie un flux de données, qui correspond aux données transmises par l'émetteur. L'architecture la plus utilisée par les récepteurs radios actuelles est le superhétérodyne, dont la description générale est donnée dans la Figure I.3.

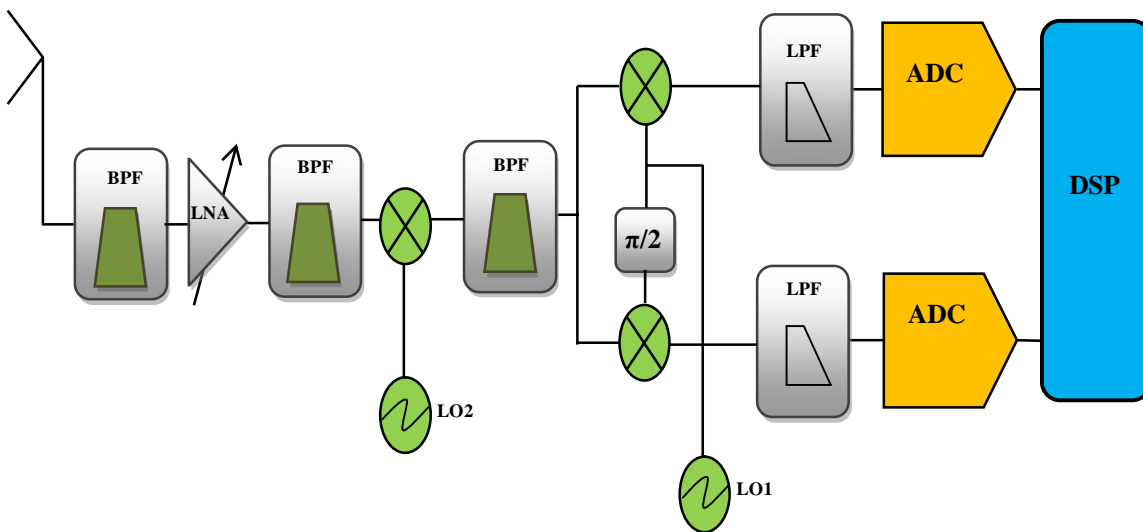


Figure I.3– Représentation d'un récepteur superhétérodyne.

Le domaine RF est la sortie directe de l'antenne [9]. Ensuite, on doit isoler la bande de réception en utilisant un filtre RF, puis le signal est amplifié par un amplificateur faible bruit (LNA), ensuite, on multiplie le signal RF avec un signal issu d'un oscillateur local pour faire le transposé autour d'une fréquence intermédiaire (FI) fixe, (dans le cas du récepteur superhétérodyne, la transposition se fait en deux étapes ou plus). La partie analogique se termine avec la transposition du signal résultant (de fréquence FI) autour d'une fréquence centrée sur la fréquence centrale du canal souhaité (bande de base) par un

démodulateur quadratique. Le signal bande de base est ensuite échantillonné, afin de réaliser les opérations de synchronisation, demapping, désentrelacement, et correction d'erreur en numériques...[9].

L'inconvénient d'un terminal radio classique, est que la plupart des composants analogiques ne sont pas réglables et le gabarit des filtres est lié à la norme choisie. De plus, l'architecture se compose de plusieurs composants analogiques coûteux et encombrants (filtres, oscillateurs, les filtres de réjection d'image placés avant le mélangeur etc.). Mais le principal inconvénient de cette approche est que la conception d'un terminal "multistandards", imposerait la multiplication d'implantations dédiées, c'est-à-dire la juxtaposition au sein d'un même équipement des différentes chaînes de traitement des modes et/ou standards ciblés. Cette solution dite Velcro s'explique par la coexistence de toutes les normes, les standards et un commutateur sélectionnant les éléments appropriés à la réception du standard sélectionné [4] [18] [21]. On parle dans ce cas d'une radio logicielle à récepteur multistandards classique.

Dans cette architecture à radio classique (matérielle), le matériel est optimisé pour le standard et/ou l'application ciblée, ceci permet d'avoir une fréquence de fonctionnement très élevée. Bien qu'elle ait cet avantage, cette architecture à radio classique a comme défaut majeur son manque de flexibilité. Par exemple, que ce soit dans la partie RF ou dans la partie bande de base, il faut prévoir autant de chaînes RF, modulation/démodulation et codage/décodage de canal, que de normes et standards de communication utilisés. On peut imaginer donc l'utilisation de plusieurs composants analogiques (antenne, filtres RF... etc.) dans la partie RF et aussi l'utilisation de plusieurs processeurs de traitement de signal (DSP) ou plusieurs circuits Intégrés Spécialisés (Application Specific Integrated Circuit ou ASIC) dans la partie bande de base, où chacun est dédié à un standard. Pour changer un standard par un autre, il suffit d'activer les composants analogiques RF et le DSP de la chaîne correspondant à ce standard. Cela a pour principales conséquences d'augmenter les coûts de fabrication, la consommation en surface et l'encombrement des systèmes de communication. Il y a donc une limite rapidement atteinte quant au coût de la multiplication des implantations matérielles dédiées [1] [2] [3] [4].

De plus, ce type de conception n'est pas évolutif et n'autorise pas la prise en compte de nouveaux standards et/ou services. En conséquence, un équipement radio ne peut réaliser qu'un nombre donné de standards. En effet pour l'utilisateur, il doit changer l'équipement à chaque changement technologique pour profiter des dernières avancées. Les fournisseurs doivent ainsi mettre à jour les équipements au sein des stations de base afin d'offrir ces nouveaux services. Ceci donc pose un problème pour l'utilisateur et le fournisseur.

L'exemple du SmartPhone est un bon exemple de ce type de radio. Ces appareils sont capables de surfer sur plusieurs réseaux et normes tel que le GSM, GPRS, EDGE, UMTS, HSPA, WiFi et le Bluetooth par la coexistence de plusieurs ASIC figés, chacun étant spécifié pour exécuter un standard.

On comprend alors que l'intérêt pour les utilisateurs et les fournisseurs est d'avoir un seul équipement comprenant une chaîne de traitement unique et flexible, capable de prendre en charge plusieurs fonctionnalités et/ou standards, et de se reconfigurer selon les besoins de manière dynamique.

5.2. Radio reconfigurable

Imaginez un téléphone portable capable de fonctionner partout dans le monde, de surfer sur tous les réseaux sans fil, et de se transformer en système de positionnement par satellite ou en téléviseur miniature. Ce terminal à tout faire devrait devenir une réalité grâce à la radio flexible et reconfigurable. Ce terminal a la capacité d'utiliser le même matériel pour exécuter plusieurs chaînes de communications différentes en téléchargeant un simple logiciel ou en modifiant des paramètres. La radio reconfigurable vient contourner les problèmes de la radio classique. La radio reconfigurable se base essentiellement sur l'idée de la Radio Logicielle (RL) introduite par Mitola [17].

5.2.1. La radio logicielle (Software Radio)

La radio logicielle ("software radio" en anglais) qui est une mise en œuvre de la radio reconfigurable, vise à concevoir un système de communication universel, versatile capable de supporter différents standards de télécommunication. Afin d'atteindre cet objectif de la radio logicielle, ce système universel possède la capacité d'être facilement reconfigurable, paramétrable ou reprogrammable. Plus précisément, l'idée de base de la radio logicielle est "un ensemble de techniques qui permettent la reconfiguration d'un système de communication sans nécessité de modifier les éléments matériel du système" [4]. Ce qui signifie qu'un seul circuit universel (générique) serait capable de prendre en charge des fonctionnalités précédemment gérées par des circuits différents correspondants aux différents standards radios à supporter par l'équipement de télécommunication [4].

L'origine de la radio logicielle (RL) remonte aux années 70 avec les radios militaires définies par le département américain de la défense (DoD). A cette époque, ils ont décrit une radio VLF (Very Low Frequency) qui utilise un convertisseur analogique numérique et un processeur 8085 [22]. Cependant, c'est le Dr. Joseph Mitola qui a fait connaître la «radio logicielle» et l'a intégrée dans le domaine civil pour la première fois au début des années 90 [23]. Connue comme le « Père de la radio logicielle», Dr. Mitola a présenté l'architecture RL idéale, qui est caractérisée par le passage d'un système radio classique où 80% de leur fonctionnalités sont accomplies par du « hardware », vers un système

radio logicielle multi-mode où plus de 80% de leur fonctionnalités sont accomplies par du logiciel [17] (figure I.4). En effet, un système de transmission comprenant plus de composants numériques (plus de logiciel) sera plus flexible qu'un dispositif radio classique comprenant plus de composants analogiques.

On peut imaginer donc par exemple qu'un seul récepteur générique multistandard est composé d'une partie radio fréquence (RF) universelle (composée d'une antenne, d'un filtre RF et d'un amplificateur faible bruit travaillant dans une gamme fréquentielle très large), d'un convertisseur analogique numérique très rapide, et d'une partie numérique capable de faire le traitement du signal de plusieurs normes et standards (figure I.4) (la partie numérique contient des cibles différentes: ASIC, FPGA, DSP, GPP, ou leur combinaison).

La première version de la radio logicielle proposée par Mitola s'appelle la radio logicielle idéale, qui suppose que la transposition du signal RF vers le signal BB se fasse de façon directe, c'est-à-dire qu'il n'y a plus de fréquence intermédiaire comme dans l'architecture de la figure I.4. Dans ce cas, l'antenne est presque reliée directement à un échantillonneur (convertisseur analogique numérique - ADC), et toutes les opérations sont réalisées directement en numérique.

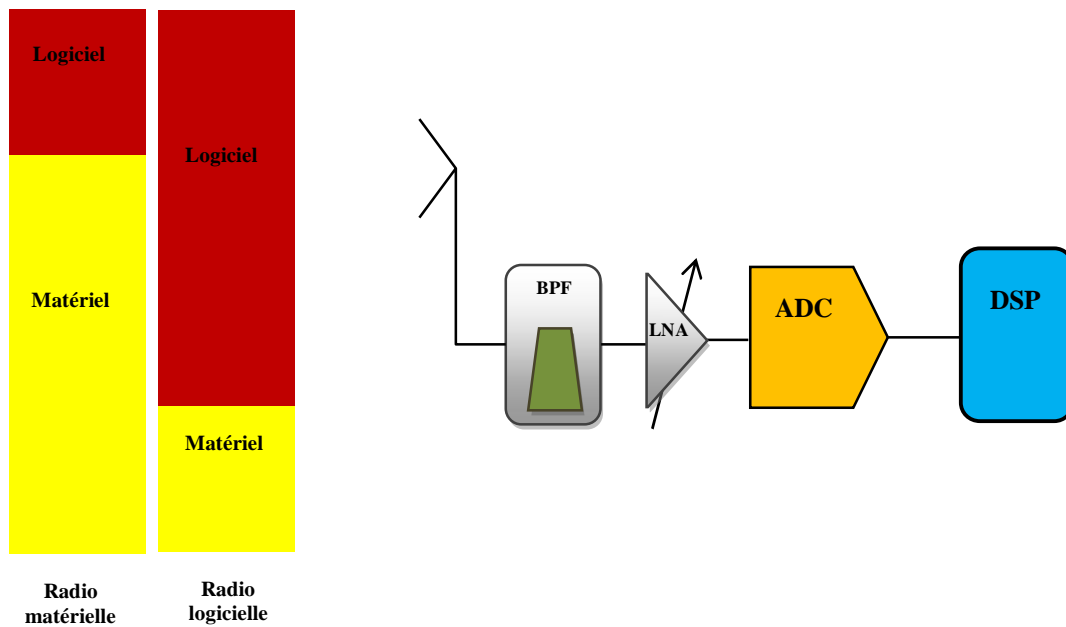


Figure I.4. Vision idéale de Mitola: la radio logicielle idéale.

Cependant, dans les systèmes radio, on travaille sur des fréquences dont l'ordre de grandeur varie entre la dizaine de mégahertz (10^7 Hz), et la dizaine de gigahertz (10^{10} Hz). Afin de numériser le signal tout en respectant le théorème d'échantillonnage de Shannon, qui dit que la fréquence d'échantillonnage (f_s) doit être supérieure où égale au

double de la fréquence de l'onde porteuse (f_c): $f_s \geq 2 \times f_c$ avec f_c la fréquence de l'onde porteuse, et f_s la fréquence d'échantillonnage, il faudrait par exemple disposer d'un convertisseur CAN travaillant à une cadence très élevée avec une haute résolution [5]. Ceci pose des contraintes très élevées sur les CAN. Même s'il existe de tels convertisseurs rapides, nous rencontrerons le problème du coût, à titre d'exemple, les convertisseurs de Texas Instruments, sur 8 bits, à 3 Giga-échantillon/s (référence: ADC083000CIYB-ND), se vendent aux alentours de 802.62 dollars l'unité et 7,902.72 dollars pour 10 circuits (en 2015) [24].

Le problème de la rapidité et de la haute résolution des convertisseurs analogique/numérique n'est pas unique. Le traitement d'un nombre très important des échantillons (en utilisant un DSP par exemple) est un autre obstacle qui s'ajoute au premier. D'après le tableau I.1 si on prend à titre d'exemple la norme Bluetooth ou le WIFI (802.11b), on trouve qu'ils s'opèrent à 2,4 GHz, donc traiter $2.4 \times 2 = 4.8 \times 10^9$ échantillons par seconde est un défi difficile à relever avec les capacités de traitement actuelles. D'après une étude de la norme UMTS (Universal Mobile Telecommunication System) en [25], l'auteur estime les besoins en calcul à 100 GIPS (porteuse autour de 900 MHz). La réalisation de la radio logicielle avec ses contraintes de traitement, peut donc se faire avec des processeurs puissants comme les récents processeurs d'Intel ou aussi avec une architecture multiprocesseurs intégrée. Malheureusement ceci pose des problèmes de coûts, de consommation et de surface. Ces difficultés imposent aux concepteurs de chercher un compromis ou *une solution intermédiaire*.

5.2.2. La radio logicielle restreinte (Software-Defined Radio - SDR)

La radio logicielle idéale cherche à convertir directement le signal radio en numérique, réduisant ainsi la composante de traitement analogique. Il faudrait donc disposer juste d'un excellent convertisseur A/N connecté directement à l'antenne et ensuite un puissant calculateur pour le traitement du signal en bande de base. Cette solution implique des contraintes extrêmes sur la bande passante, sur la dynamique des CAN et également sur la puissance de calcul des composants numériques. À l'état actuel, les circuits disponibles ne sont pas encore capables de traiter l'ensemble du spectre de 0.....30 MHz avec suffisamment de dynamique. Même s'il existe de tels circuits puissants, ils constituent une solution assez coûteuse. Ce qui fait qu'il y a toujours besoin de circuits analogiques, et par conséquent, la plupart des radios logicielles utilisent encore les mêmes techniques de transposition en FI que les radios matérielles classiques. Ce traitement préliminaire de l'onde, va diminuer la fréquence en entrée et donc réduire les contraintes discutées auparavant sur les convertisseurs ADC et les calculateurs.

Cette classe de la radio située entre la radio classique et la radio logicielle idéale, s'appelle la radio logicielle restreinte (Software-Defined Radio ou SDR en anglais) [5]

[26] dont l'architecture (récepteur) est représentée sur la figure I.5. Le signal RF capté par l'antenne large bande est amplifié par un LNA, puis ce signal est converti à un signal de fréquence intermédiaire (FI) plus basse par rapport à la fréquence de la porteuse (RF), réduisant ainsi les contraintes sur le convertisseur (CAN). Le signal numérisé résultant subit encore des traitements numériques divers (démodulation, décodage de canal,...) exécuté sur un FPGA, un DSP, un processeur classique d'un PC, ou encore une combinaison de ces différentes cibles.

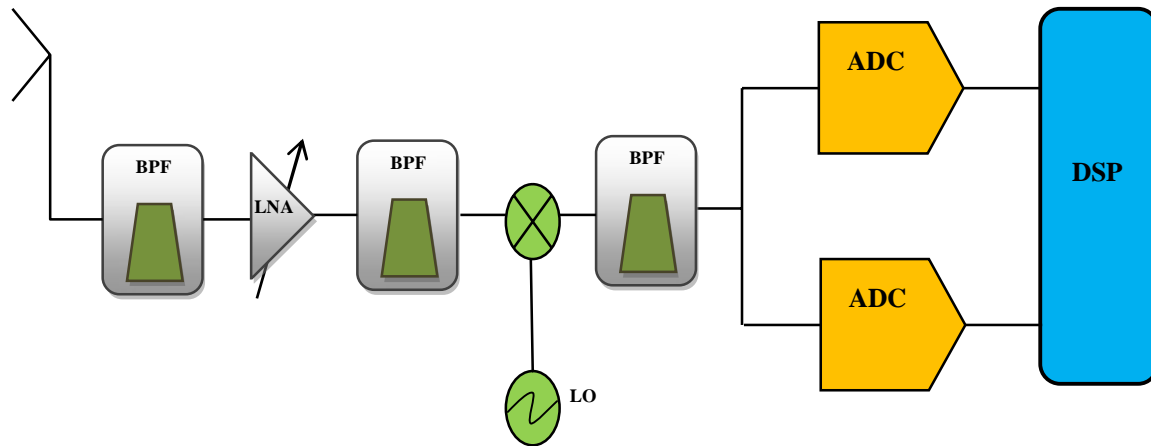


Figure I.5 – Représentation d'un récepteur de SDR.

L'architecture d'un SDR présenté sur la figure I.5 applique aussi d'autres opérations de filtrage minimisant ainsi les signaux parasites, les signaux images et les interférences.

Dans cette architecture, on trouve une partie analogique que l'on tend à remplacer progressivement et une partie numérique que l'on veut rendre de plus en plus importante.

De même, l'émetteur est composé d'un circuit électronique produisant une représentation numérique du signal à transmettre, puis un convertisseur CNA générant une représentation en FI du signal. Ce signal est alors transposé vers la fréquence porteuse désirée, amplifié jusqu'à un niveau de puissance approprié par un amplificateur de puissance et transmis par une antenne agile.

5.3. La radio cognitive (La radio intelligente)

Le mot cognitif est un adjectif qualifiant ce qui est relatif [5]:

- a la cognition ; c'est-à-dire aux grandes fonctions de l'esprit comme la perception, le langage, la mémoire, la décision, le mouvement, le raisonnement logique, le jugement moral ou l'esthétique....
- aux sciences cognitives ; on parle aussi de psychologie cognitive ou de thérapie cognitivo-comportementale [5].

5.3.1. Terminal de radio cognitive

Mitola qui est le père aussi de cette nouvelle idée, après la combinaison de la radio logicielle avec l'apprentissage automatique et l'intelligence artificielle, a produit la « radio cognitive ». Cette nouvelle notion ou idée a été présentée officiellement par lui à un séminaire à KTH, l'Institut royal de technologie, en 1998. Ensuite, il a publié cette idée dans un article en 1999 avec Gerald Q. Maguire [27] [28]. D'après Mitola:

« Une radio cognitive peut connaître, percevoir et apprendre de son environnement puis agir pour simplifier la vie de l'utilisateur ».

D'après la définition de Mitola, la radio cognitive (CR) se résume comme:

- un système conscient de son environnement, et ayant la capacité de détecter et de reconnaître son cadre d'utilisation,
- un dispositif doté d'au moins quatre entrées principales, à savoir une compréhension de l'environnement dans lequel il opère, une compréhension des besoins de communication de l'utilisateur, une compréhension des politiques réglementaires qui lui sont applicables ainsi qu'une compréhension de ses propres capacités,
- un système capable de prendre des décisions concernant leurs comportements radio suivant les informations disponibles,
- un système capable après la décision d'ajuster ses paramètres de fonctionnement radio (de façon dynamique et autonome), et de modifier son comportement (modifier l'architecture nécessaire comme la méthode d'émission/réception),
- un système capable d'apprendre des résultats de ses actions.

Nous pouvons prendre un exemple très illustratif de Laurent Alhaus [5], en considérant une voiture qui roule sur une route à double sens. Le conducteur est obligé en permanence non seulement de jauger de la distance entre son véhicule et celui qui le précède mais aussi d'anticiper les mouvements de l'ensemble des véhicules et la configuration du terrain. Toute action de ce dit conducteur ne pourra se faire que lorsque la prise de connaissance de son environnement sera effective afin de ne pas créer d'accident. Dans le monde des télécommunications, un terminal, évoluant au sein d'un ensemble d'équipements dans une même cellule, doit satisfaire au mieux les besoins des utilisateurs sans perturber les autres équipements.

On peut prendre aussi l'exemple de la norme « WiMAX », où la densité de la modulation du signal transmis peut varier en fonction des conditions de la liaison radio, plus

particulièrement le rapport signal à bruit (figure I.6). Près de la station de base, la qualité du signal reçu est meilleure puisque la puissance du signal utile est supérieure à celle du bruit, dans ce cas le système intelligent utilise une modulation dense (64-QAM) qui garantit un débit de transmission élevé malgré une robustesse moindre face aux perturbations. Cependant, la qualité du signal se dégrade de plus en plus avec l'éloignement de l'utilisateur de la station de base, dans ce cas le système intelligent utilise une modulation moins dense (16-QAM ou 4-QAM) qui donne un débit faible, mais elle est plus robuste face aux perturbations qu'elle procure à la liaison.

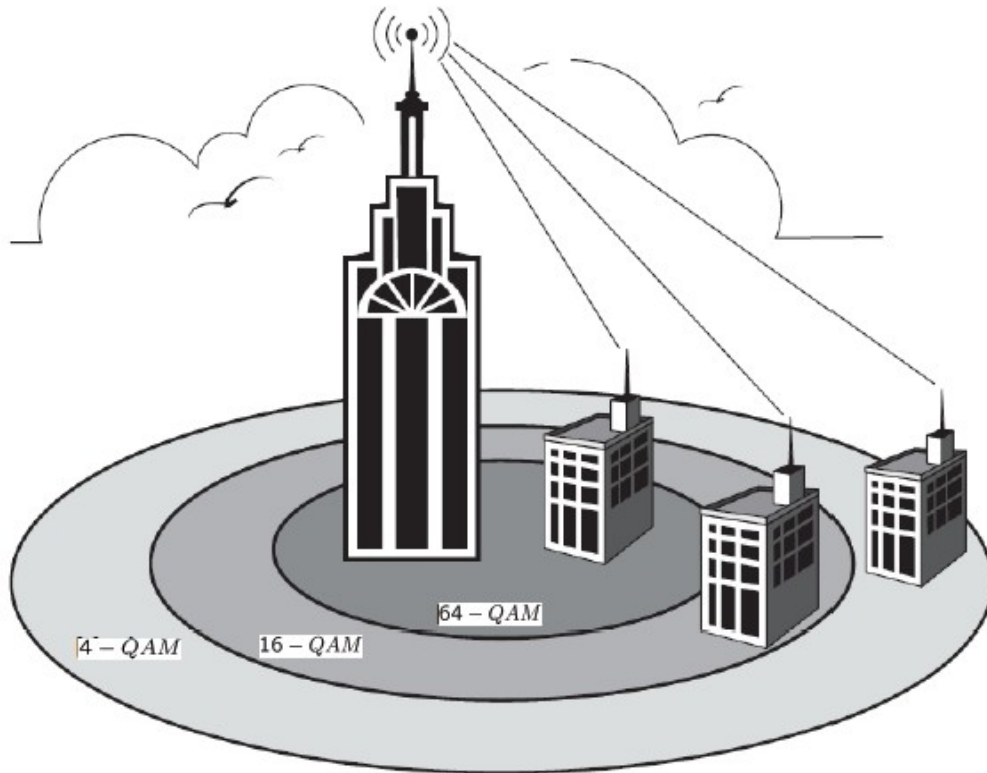


Figure 1.6 – Adaptation de la modulation dans un système **WiMAX** [28].

5.3.2. Relation entre la radio cognitive et la radio logicielle restreinte

L'une des principales caractéristiques de la **radio cognitive** est la capacité d'adaptation de façon dynamique où les paramètres de la radio (fréquence porteuse, puissance, modulation, bande passante, codage de canal) peuvent être modifiés en fonction de: l'environnement radio, de la situation, des besoins de l'utilisateur, de l'état du réseau et de la géo-localisation,...etc. La radio logicielle peut également offrir la flexibilité et la reconfigurabilité nécessaires pour obtenir l'adaptation de la radio cognitive. En d'autres termes, la radio logicielle est une "**technologie habilitante**" pour la radio cognitive [5][27], c'est-à-dire cette dernière doit être mise en œuvre autour d'une radio logicielle.

5.3.3. Cycle de cognition

La figure I.7 illustre le cycle de cognition ainsi appelé par Mitola et Maguire [29] qui synthétise les fonctions de cognition de l'architecture d'une radio intelligente [28] [30].

L'observation (observe) consiste à prendre conscience de l'environnement par la capture de métriques; la compréhension de l'environnement radio, des besoins des utilisateurs, des contraintes existantes. L'orientation (orient) consiste à classer les métriques par priorités et aiguiller le traitement (normal, urgent, immédiat). La planification (plan) consiste en la prise de décisions à long terme c'est-à-dire le système planifié les meilleures configurations possibles suivant les métriques observées. Le processus de décision (decision) se sert des observations pour produire une action ou un ensemble d'actions, on parle ici à allouer les ressources nécessaires. Le processus agit (act) pour effectuer la reconfiguration de l'équipement. Enfin, l'apprentissage (learn) où toutes les perceptions sensorielles ainsi que les actions sont continuellement comparées à l'ensemble des expériences antérieures. En effet, l'apprentissage a pour but d'apprendre des échecs ou des réussites des précédentes reconfigurations.

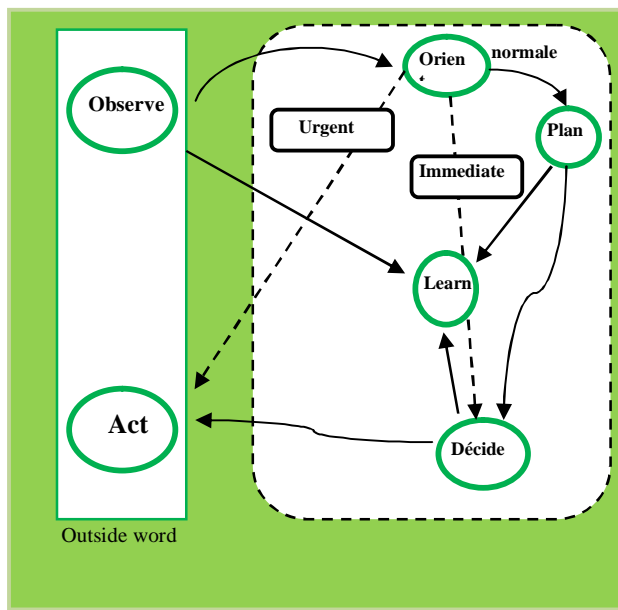


Figure I.7 : Le cycle de cognition de Mitola et Maguire [28].

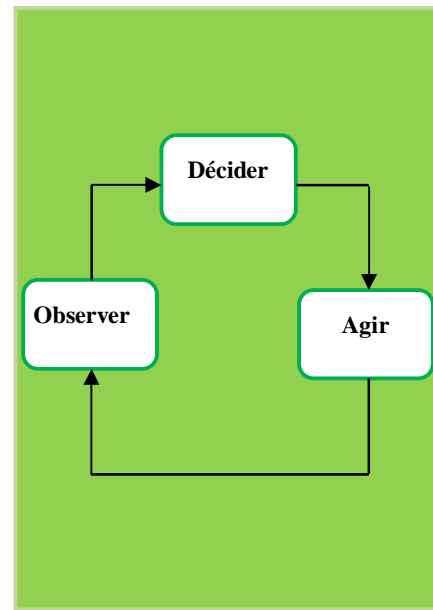


Figure I.8 : Le cycle de cognition simplifié [28].

Il est possible de simplifier le schéma de la figure I.7 par celui de la figure I.8. Ce schéma simplifié intègre les notions d'intelligence comme "orient", "plan", "learn" dans la notion "decide". Selon la figure I.8 simplifiée, le terminal observe en permanence son environnement, puis et selon les résultats de l'observation, le terminal transmet à l'unité de reconfiguration les modifications à effectuer après la décision.

5.3.4. Modèle OSI de la radio cognitive.

La figure I.9 montre l'approche générale qui peut aider la radio afin de mieux adapter sa fonctionnalité pour un service donné dans un environnement donné. La radio cognitive est structurée suivant trois couches du modèle OSI (figure I.9).

Sensors (capteurs)	La couche
Le profil de l'utilisateur (Prix. Des choix personnels), localisation, Son, vidéo, vitesse, sécurité.	Application
Handover vertical, standards.	Transport Réseau
Mode d'accès, puissance, modulation, codage, fréquence, handover, estimation de canal.	Liaison de données physique

Figure. I.9. modèle OSI simplifié pour la radio cognitive

Dans les couches basses du modèle OSI, nous trouvons toutes les informations de détection (sensing information) liées à la couche physique: la propagation, la consommation d'énergie, le schéma de codage, la modulation, etc. Au niveau intermédiaire du modèle OSI, on trouve toutes les informations qui participent au handover vertical, ou peut aider à faire un choix de standard, comme un capteur de détection du standard. La couche la plus élevée est liée aux applications, et tout ce qui concerne l'utilisateur, ses habitudes, les préférences, les politiques, la position de l'utilisateur, sa langue maternelle, le profil et peut apprendre les routines et les habitudes de voyage de l'utilisateur. Si un utilisateur a l'habitude de se connecter à un service de vidéo à la demande tous les soirs en rentrant à la maison de son travail par le métro, un terminal CR devrait être consciente de planifier toutes les exigences en termes de vie de la batterie, une quantité suffisante de crédit sur son contrat, par succession de handover vertical en fonction de chaque zone pendant le voyage, etc.

L'équipement peut être conscient de son environnement avec l'aide de capteurs comme le microphone, la caméra vidéo, les biocapteurs, etc. Une caméra vidéo par exemple peut être utilisée pour indiquer si le terminal est à l'extérieur ou à l'intérieur d'un bâtiment. Cela peut avoir un impact sur les caractéristiques de propagation, mais également de la capacité ou non à recevoir des signaux GPS. Nous pouvons penser aussi que notre équipement peut être sécurisé par des informations biométriques sûres de l'utilisateur et/ou la reconnaissance de visage.

5.3.5. Radio opportuniste

Nous pouvons dire que la radio opportuniste représente une application de la radio intelligente, qui a pour but d'optimiser l'utilisation des bandes de fréquence, sous licence ou non (lorsque la radio opportuniste cherche à utiliser des fréquences sous licences, elle s'appelle radio opportuniste fréquentielle ou spatiale). Dans ce type de radio, nous avons un réseau opportuniste (un réseau secondaire), et un réseau officiel (un réseau primaire). Un utilisateur dit secondaire pourra à tout moment accéder à une bande de fréquence d'un autre utilisateur primaire qui possède une licence sur cette bande. L'utilisateur secondaire peut accéder seulement à cette bande, s'il trouve que celle-ci est libre et non occupée par l'utilisateur primaire. Une fois, le service terminé ou l'utilisateur primaire a besoin de faire une connexion, l'utilisateur secondaire devra libérer cette bande.

Nous pouvons reprendre l'exemple de Laurent Alhaus [5] relatif à la circulation routière. Si un conducteur, circule sur une route à double sens et qu'il désire doubler le véhicule qui le précède, il n'a d'autre choix que d'emprunter la voie venant en sens opposée. Ceci n'est possible que si cette voie est libre et si aucun autre véhicule n'arrive en sens inverse. Nous avons comme application de cette radio l'exemple de [31], qui est un projet s'appelant « Oracle ». Dans ce projet, le réseau secondaire permet l'échange d'informations alors que le réseau primaire est de type WiFi.

Un réseau opportuniste a besoin donc de connaître les différentes fréquences de travail présentes dans le réseau afin de se reconfigurer intelligemment et convenablement, c'est pour cela qu'il nécessite un équipement capable de travailler sur une large gamme de fréquences.

Afin de gérer les interactions entre les utilisateurs, nous avons deux approches distinctes: **la première est l'approche Centralisée** où l'intelligence est centralisée dans une centrale de décisions. Dans ce cas, le terminal ne possède pas d'intelligence propre, son rôle consiste seulement à examiner son environnement et à communiquer ses résultats avec la centrale de décisions. Ce dernier agrège toute la connaissance de l'environnement pour ensuite répartir les allocations spectrales aux différents terminaux. Les terminaux recevront donc les paramètres opérationnels nécessaires pour faire la reconfiguration convenable en fonction de cette allocation spectrale. **La deuxième approche est dite l'approche Décentralisée** où chaque terminal est autonome dans la prise de décision. Cette approche implique, que chaque équipement possède sa propre intelligence qui lui permet d'examiner son environnement et de sélectionner une bande de fréquence sur laquelle il peut transmettre.

6. Conclusion

Dans ce premier chapitre, nous avons d'abord mis l'accent sur l'intérêt de la réalisation d'un terminal multistandard qui permettrait aux utilisateurs de disposer d'un

émetteur/récepteur universel et unique capable d'opérer selon plusieurs normes et standards de radiocommunications. La radio logicielle est une technique prometteuse pour répondre aux besoins de la future génération de la radio communication. Nous avons présenté donc dans ce chapitre les différents concepts liés à des notions importantes concernant la radio logicielle idéale (SR), la radio logicielle restreinte (SDR) ainsi la radio cognitive (RC).

Ainsi il est clair qu'à partir de ce chapitre, nous considérons que la radio logicielle restreinte et la radio cognitive, nécessitent des architectures avec des contraintes variant en fonction des composants qui composent la chaîne de transmission/réception SDR. Dans le deuxième chapitre, nous détaillerons les différentes architectures et solutions proposées dans les littératures pour répondre à ces exigences et besoins d'un terminal multistandard. Au troisième et au quatrième chapitre, nous allons mettre en évidence notre travail, en proposant des architectures génériques capables de prendre en charge plusieurs fonctionnalités.

Chapitre II

Description détaillée d'une Software-Defined- Radio (SDR)

1. Introduction

L'intérêt des systèmes radio logicielle et radio cognitive a été discuté dans le chapitre précédent. Le principal avantage de ces nouveaux systèmes de transmissions est de pouvoir communiquer en utilisant plusieurs standards avec une seule architecture universelle sur un même terminal mobile. Par conséquent, il est très important pour ces systèmes, de posséder de nouvelles architectures agiles, flexibles et versatiles. Celles-ci doivent donc comporter des couches physiques génériques, capables de générer/recevoir toutes les formes d'ondes, et également faire le traitement en bande de base (modulation, démodulation, filtrage, codage et décodage de canal...) de plusieurs normes et standards.

Dans ce chapitre, nous détaillerons les différents éléments constitutifs d'une radio logicielle restreinte, en commençant par l'antenne jusqu'à, la partie bande de base. Dans la seconde partie de ce chapitre, les plateformes de la radio reconfigurable disponible seront discutées. Ensuite, nous décrivons l'approche de paramétrisation qui se situe parmi les techniques de reconfiguration actuelles les plus prometteuses, avec différents niveaux envisagés: fonctions communes et opérateurs communs.

2. Le frontal RF (radio fréquence) d'une SDR

Dans une architecture d'émission SDR typique, la partie numérique (DSP par exemple) produit une représentation numérique du signal à transmettre, puis le convertisseur CNA génère une représentation analogique de ce signal à la fréquence FI. Ce signal est ensuite transposé à la fréquence RF, subit une amplification jusqu'à un niveau de puissance approprié et transmis par l'antenne. De même, dans une architecture de réception SDR typique, le frontal RF amplifie puis convertit le signal récupéré par l'antenne à la fréquence porteuse (RF) en une fréquence intermédiaire (FI) de sorte que le signal puisse être numérisé facilement par un CAN, et ensuite traité par la partie numérique (un processeur de traitement numérique du signal ou un FPGA par exemple).

2.1. Antenne

La première composante matérielle d'une radio logicielle restreinte (SDR) est l'antenne, qui est un composant électronique permettant de rayonner et/ou de capter les ondes électromagnétiques. Comme décrit dans le chapitre I, nous avons plusieurs standards et normes, qui travaillent dans une gamme fréquentielle différente les uns par rapport aux autres. Par conséquent, les antennes dans les systèmes de transmission SDR, travaillent autrement que les antennes utilisées dans les systèmes de transmission classique. En effet, il faut disposer des antennes large bande et/ou agiles en fréquence, agiles en diagramme de rayonnement et en polarisation. Donc une antenne est dite agile (reconfigurable ou aussi paramétrable) si elle est capable de modifier dynamiquement sa caractéristique de fonctionnement (en terme de fréquence, et/ou de polarisation et/ou de diagramme de rayonnement) après sa fabrication. Cette modification de la fonctionnalité se fait selon les besoins dictés par l'environnement de l'antenne et les besoins de l'application [32]. Le progrès remarquable de la microélectronique a proposé de nouvelles approches plus efficaces pour obtenir des antennes agiles (reconfigurables). Nous pouvons citer les diodes PIN (Positive Intrinsic Negative), les diodes Varicap, les MEMS (Micro Electro Mechanical Systems), et les commutateurs optiques.

2.1.1. Antennes agiles en fréquence

L'augmentation importante et sans cesse du nombre des standards et normes de télécommunications, tels que le GSM, GPS, UMTS, LTE, BlueTooth, WiFi, WLAN, WiMAX, ainsi le besoin de faire la cohabitation entre eux sur une même antenne acquiert un intérêt croissant. C'est pour cette raison, l'agilité en fréquence de l'antenne devienne une fonctionnalité très importante et indispensable. En plus de cela, le spectre fréquentiel est mal utilisé, là où il ya un gaspillage sur quelques bandes fréquentielles qui ne sont pas utilisées, alors que d'autres le sont de façon intense. Pour une utilisation très efficace de la bande fréquentielle, nous avons les nouveaux systèmes extrêmement flexibles et intelligents tel que la radio cognitive (section 5.3 du chapitre I). L'implémentation de la

radio cognitive impose l'utilisation d'un composant capable de travailler dans une bande fréquentielle très large (une antenne large bande pour la détection et une antenne reconfigurable pour la communication).

Parmi les approches proposées pour réaliser ces types d'antennes, nous citons les réseaux d'antennes patch configurables à base de résonateurs électromécaniques. Dans cette approche chaque élément résonne à une fréquence différente (figure II.1), et l'activation ou la désactivation des éléments résonants est assurée par les commutateurs micro-électro-mécaniques (MEMS). Grâce à cette technique, il est possible de reconfigurer l'antenne pour s'adapter à différentes bandes spectrales.

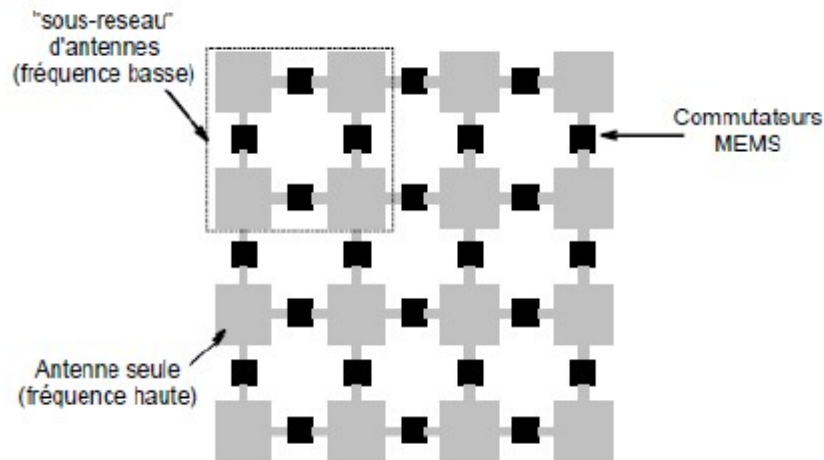


Figure II.1 : Réseau d'antenne patch configurable.

D'autres approches utilisent des matériaux (substrats) "intelligents" ou encore "agiles". L'importance de ces matériaux est que leurs propriétés électromagnétiques comme la permittivité et/ou la perméabilité, peuvent être modifiées par l'application d'un champ électrique et/ou magnétique externe. Ces matériaux utilisés comme un substrat peuvent donc assurer l'agilité en fréquence des antennes.

Y. Hawk et al. [33] proposent un schéma d'antenne pour la radio cognitive composé d'une antenne ULB de structure elliptique permettant de couvrir une large bande utilisée pour la détection, et une antenne reconfigurable en fréquence destinée à la communication (les deux antennes utilisées, sont imprimées sur un même substrat). L'agilité en fréquence est obtenue par deux commutateurs photoconducteurs [33]. BEN TRAD Imen [34] propose une antenne carrée multi-bandes avec fente fractale capable de commuter entre quinze bandes de fréquences de 0.5 à 6 GHz. La reconfigurabilité en fréquence est obtenue en intégrant seize diodes PIN le long de la fente fractale. Selon les états des diodes PIN intégrées dans la fente (On ou Off), l'antenne est capable de rediriger le flux de courant et commuter entre quinze bandes de fréquences [34]. En général, le travail dans ce domaine est vaste et sort du cadre de notre travail.

2.1.2. Antennes agiles en diagramme de rayonnement

La diversité de diagrammes de rayonnement est une caractéristique importante pour la conception d'antennes dédiées à la radio cognitive. Une antenne qui doit avoir la capacité de modifier la forme et la direction des diagrammes de rayonnement et de privilégier ou non certaines directions, présente plusieurs avantages comme l'amélioration des débits de transmission, la limitation des interférences avec d'autres utilisateurs. De façon générale, elle augmente l'autonomie afin d'améliorer la performance globale du système. G.-M. Zhang et al. [35] par exemple, proposent une antenne compacte large bande à diagramme de rayonnement reconfigurable (figure II.2).

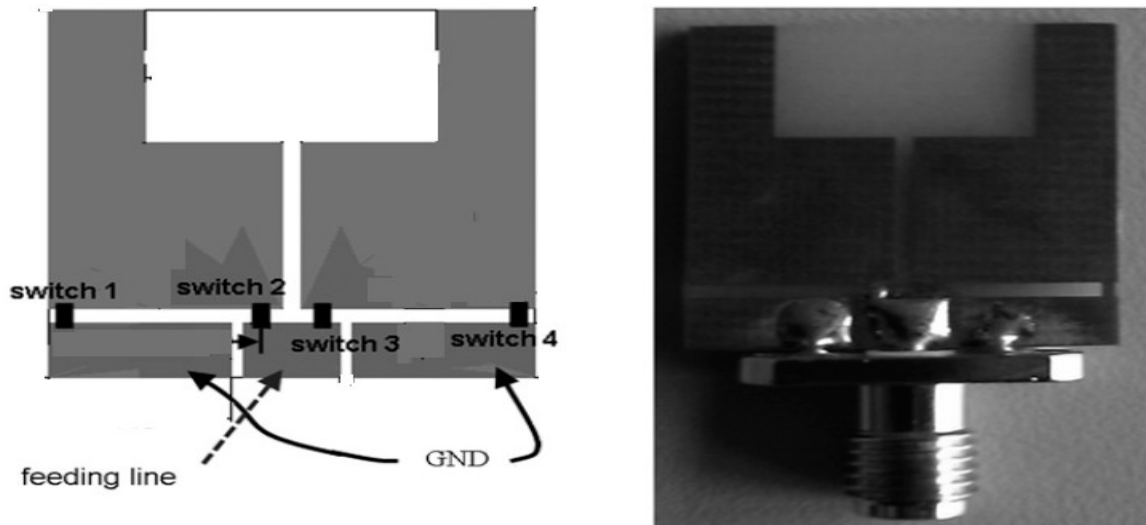


Figure II.2. L'antenne agile en diagramme de rayonnement proposée dans [35].

Ils ont utilisé une paire de patch, quatre commutateurs et une ligne d'alimentation guide d'ondes coplanaire (CPW) comme l'indique la figure II.2. En réglant les commutateurs à l'état On ou Off, les deux patches peuvent alternativement basculer entre le radiateur et le réflecteur (Figure II.2). L'antenne donc peut fournir des diagrammes de rayonnement directionnels de deux faisceaux dirigés plus ou moins dans des directions opposées [35].

2.1.3. Antennes agiles en polarisation

Le troisième type de l'agilité, est la modification de la polarisation linéaire ou circulaire. En effet, pour optimiser les performances du multiplexage spatial tirant partie des phénomènes multi trajets dans les systèmes MIMO, les signaux reçus doivent être non-corrélés, donc la diversité spatiale nécessite une distance minimale entre des sources rayonnantes identiques, ceci augmente l'encombrement des antennes. Pour limiter cet encombrement, la diversité spatiale peut être remplacée par la diversité en polarisation et/ou en diagramme de rayonnement. De manière générale, pour modifier la polarisation de l'antenne, on cherche à modifier le sens et/ou la phase de circulation des courants sur l'élément rayonnant [34].

2.2. Filtres analogiques.

Les filtres jouent un rôle important dans plusieurs applications RF. Ils sont distribués dans les différentes parties d'un émetteur-récepteur sans fils: dans la partie bande de base (BB), dans la partie à la fréquence intermédiaire (IF) et dans la partie radiofréquence (RF). De plus, compte tenu que le spectre électromagnétique est limité et a besoin d'être partagé parmi toutes les applications RF existantes, les filtres sont utilisés donc pour sélectionner/rejeter ou séparer/combiner des signaux de différentes fréquences.

Un filtre analogique agile (reconfigurable) est un filtre dont la fréquence centrale peut être volontairement déplacée, dans une gamme de fréquences pouvant couvrir une fraction importante d'une octave. Pour qu'un tel filtre soit parfaitement adapté à la radio cognitive, cette agilité doit être accompagnée aussi de:

- un changement très rapide de la fréquence centrale, pour ne pas perturber le traitement du signal,
- un coefficient de qualité réglable,
- supporter des niveaux de puissance variables: à l'émission, il a pour rôle d'éviter à l'émetteur de polluer les canaux voisins, le filtre dans ce cas est placé en sortie des amplificateurs d'émission, donc il fonctionne sous forte puissance; à la réception, il permet au récepteur de n'amplifier que le signal utile,
- le filtre qui ne doit pas être uniquement agile en fréquence mais aussi en bande passante afin de pouvoir s'adapter aux gabarits liés aux différents standards.

Dans les paragraphes qui suivent, nous donnons quelques exemples sur les travaux dans les filtres analogiques agiles.

2.2.1. Les filtres passifs LC intégrés

Les filtres passifs intégrés sur silicium sont conçus à base des composants selfs et de capacités. L'agilité de la fréquence centrale, est obtenue par la modification des valeurs de ces composants. Leur avantage réside dans le faible bruit, la grande linéarité et la faible consommation. En revanche, cette agilité est difficilement obtenue. Les seules techniques qui permettent leur reconfiguration est l'utilisation des composants micromécaniques déformables appelés "Micro Electro Mechanical Systems" (MEMS). Dans la première application des MEMS, les capacités d'un filtre agile sont constituées de deux électrodes l'une est fixe et la deuxième est mobile. L'électrode mobile se déplace en utilisant les MEMS ce qui va modifier la valeur de la capacité (variation entre C_{max} et C_{min}). La variation de la capacité permet de varier la fréquence centrale [37]. Les MEMS sont utilisés aussi pour commuter différentes capacités, où le filtre agile est constitué d'une banque de capacités, dont certaines sont remplacées par des capacités "MEMS

digitales". La plage de fréquence couverte par ce type de reconfiguration dépend des valeurs des capacités qui sont connectées [36] [37].

2.2.2. Filtrés actifs à simulation d'inductance

Les problèmes des filtres passifs résident dans le faible facteur de qualité (ceci revient à l'utilisation des selfs), ils sont relativement volumineux et leurs procédés de fabrication sont encore mal maîtrisés. Dans les filtres actifs à simulation d'inductance, les selfs passives intégrées sont remplacées par des inductances simulées en utilisant le principe du "Gyrateur. La figure II.3, représente un filtre actif à simulation d'inductance à partir de transistors Bipolaires ou CMOS, où la valeur de l'inductance peut être modifiée à l'aide des différents courants de polarisation des éléments [36] [37].

Dans cet exemple (figure II.3) qui représente un filtre passe bande du second ordre, les transistors M1 et M2 permettent d'adapter l'impédance d'entrée du circuit à 50 Ω. Les transistors M3 et M4 constituent l'inductance dont la valeur est modifiable en agissant sur la tension de grille v_{g5} de M5. Les transistors M6 à M10 jouent le rôle d'une résistance négative dont les tensions de contrôle v_{q1} et v_{q2} permettent d'en modifier la valeur. Ceci permet ainsi d'agir sur le coefficient de qualité du filtre. L'étage de sortie M11 et M12 joue le même rôle que celui de l'étage d'entrée [37].

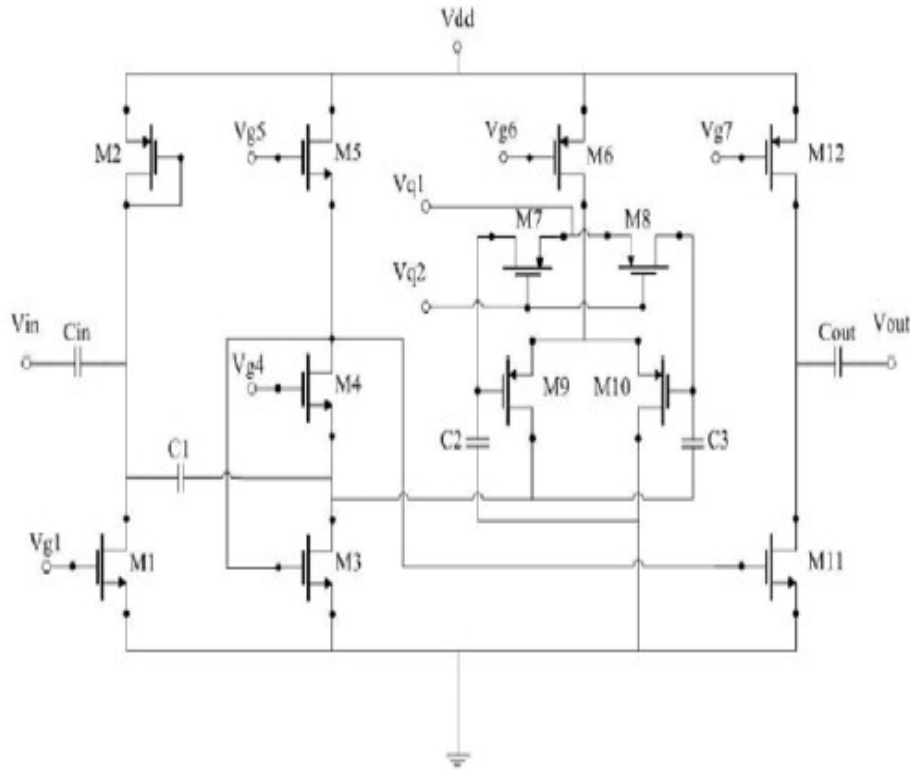


Figure II.3. Filtre passe bande du second ordre avec transistors MOS, [37].

Les résultats de simulation en technologie AMS 0.35 μm montrent que si le filtre est polarisé entre 0 et 5V, la fréquence centrale est variée entre 0.5 et 1.3 GHz, ainsi qu'avec un facteur de qualité supérieur à 60 [37]. D'autres Gyrateurs à base des amplificateurs opérationnels à transconductance (OTA) sont possibles aussi, où les réglages de la fréquence centrale et le facteur de qualité s'effectuent par la variation des courants de polarisation.

2.2.3. D'autres filtres actifs agiles

Yahya LAKYS [37], a démontré que l'ajout d'un amplificateur de contre réaction à gain 'A' variable à une cellule de filtrage classique de second ordre, permet d'obtenir un filtre totalement actif et agile. La fréquence centrale du filtre est facilement réglable si on modifie la valeur du gain 'A' de l'amplificateur de contre réaction. Il a démontré aussi que la variation du gain 'A' de l'amplificateur permet non seulement de varier la fréquence centrale, mais aussi de varier les paramètres caractéristiques du filtre agile: coefficient de qualité QA , bande passante et gains des sorties [37]. D'autres travaux proposent ce qu'on appelle les LNA filtrants. Ceux-ci sont en réalité des amplificateurs à faible bruit (Low Noise Amplification) et un résonateur LC accordable, ils représentent: une fonction de filtrage, une amplification et un faible facteur de bruit.

2.3. Les amplificateurs à faible bruit et les amplificateurs de puissance

Les amplificateurs sont des éléments importants dans les chaînes d'émission-réception radiofréquences et microondes. Dans la chaîne de réception, on trouve les amplificateurs en régime linéaire à faible bruit (LNA ou Low Noise Amplification), et dans la chaîne de transmission, on trouve les amplificateurs de puissance en régime non linéaire (AP) [9].

Les amplificateurs à faible bruit sont surtout utilisés en réception après l'antenne à cause de l'influence de leur faible facteur de bruit sur le facteur de bruit total de la chaîne. En plus, la gamme de puissance des signaux que peut traiter la voie de réception est limitée; pour les faibles puissances, l'amplitude minimale du signal détectable est déterminée par le bruit de l'amplificateur «LNA», et pour les fortes puissances, l'amplitude du signal est limitée par les signaux brouilleurs tombant par intermodulation dans le canal utile, qui gênent ainsi la réception.

Les amplificateurs de puissance RF, sont utilisés en émission car leur rôle est d'amplifier le signal radiofréquence (RF). L'amplificateur de puissance convertit la puissance fournie par l'alimentation (P_{DC}) en puissance émise autour d'une fréquence RF (P_{RF_S}) suivant l'information du signal d'entrée à la fréquence RF (P_{RF_E}). Une partie de la puissance fournie peut être dissipée (P_{dissp}) par l'amplificateur. Le bilan des puissances entrantes et sortantes est donc établi par:

$$P_{DC} + P_{RF_E} = P_{RF_S} + P_{dissp} \quad (1)$$

La puissance dissipée est une puissance perdue et elle doit être minimisée. Le rendement en puissance ajoutée doit actuellement dépasser les 50 % car l'amplificateur de puissance consomme près de 80 % de l'énergie du système radiofréquence complet. L'utilisation de classes de fonctionnement à haut rendement, comme les classes D, E et F, est donc indispensable [38].

Les non-linéarités de l'AP provoquent des interférences dans les canaux adjacents au canal d'émission/réception. La pré-distorsion est l'une des techniques qui peut être utilisée pour rendre linéaire l'amplificateur de puissance. L'idée est de créer une distorsion à l'entrée de l'AP qui annule les distorsions créées par l'AP lui-même (figure II.4) [39].

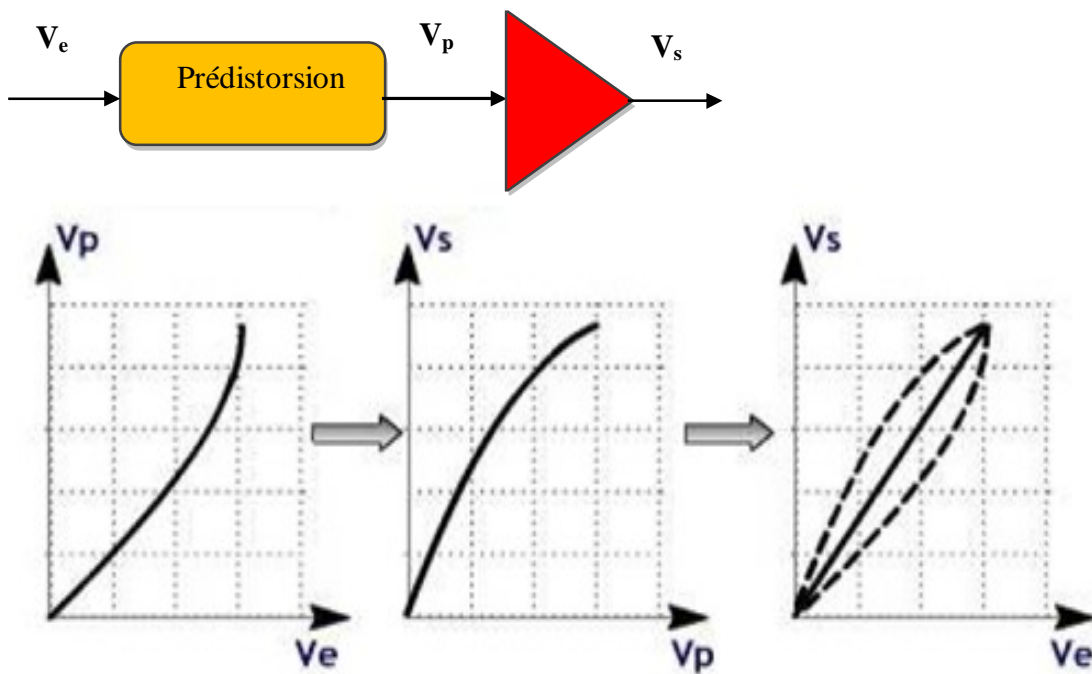


Fig. II.4. Linéarisation par pré-distorsion

La pré-distorsion est réalisée par un composant non linéaire générant des distorsions inverses de celles générées par l'amplificateur de puissance (AP).

Un amplificateur de puissance reconfigurable multistandards est un amplificateur capable de modifier dynamiquement, sa linéarité, son rendement et sa fréquence (la bande d'amplification). Cette agilité (reconfigurabilité) se fait suivant le besoin de l'application et les spécifications de chaque standard. N. Deltimple et al. [40], proposent un amplificateur de puissance pour trois standards. Ils utilisent un Varicap pour le contrôle de la fréquence, et ils modifient la classe de fonctionnement pour le contrôle de la linéarité avec le rendement.

2.4. Les mélangeurs (Mixers: Up-converters et Down-Converters)

Le mélangeur réalise la multiplication temporelle de deux signaux, l'un étant une sinusoïde pure appelée OL (Oscillation Locale) et l'autre étant le signal utile, ce dernier, peut être un signal de haute fréquence (Radio Fréquences RF) ou un signal de basse fréquence (IF ou BB). Cette multiplication va donc transposer le spectre du signal utile, soit à une fréquence plus basse (Down-conversion) soit à une fréquence plus haute (Up-conversion) [9]. Par exemple, dans la chaîne de réception, le mélangeur transpose la fréquence porteuse du signal radiofréquence vers une fréquence plus basse, appelée FI (Fréquence Intermédiaire) ou directement en bande de base (BB) (voir figure I.4 du chapitre I).

Dans le cas « Up-conversion », la fréquence d'échantillonnage en sortie du mélangeur doit être égale à la fréquence d'échantillonnage du CNA. L'OL fonctionne déjà à la fréquence d'échantillonnage du CNA, mais la fréquence d'échantillonnage en entrée du mélangeur numérique est généralement beaucoup plus faible. Pour pallier à ce problème on utilise un filtre à interpolation, qui augmente d'un facteur N (facteur d'interpolation) la fréquence d'échantillonnage du signal d'entrée bande de base.

Dans le cas « Down-conversion », on peut diminuer le taux d'échantillonnage d'un facteur N par la décimation. En effet, en réduisant la fréquence d'échantillonnage, on peut réduire considérablement le coût et la complexité des processeurs de traitement signal. En général, les contraintes imposées au mélangeur sont fortes en termes de gain, de bruit, de linéarité et d'isolation [41].

2.5. Synthétiseur de fréquence

Généralement, la synthèse de fréquence se réalise à l'aide d'une boucle à verrouillage de phase (PLL) et les éléments les constituants sont :

- L'Oscillateur Contrôlé en Tension (OCT ou Voltage Controlled Oscillator : VCO),
- Le diviseur de fréquences,
- Le comparateur de phase/fréquence,
- Le filtre de boucle.

La boucle à verrouillage de phase est un système à contre-réaction qui permet d'asservir la phase instantanée du signal de sortie du diviseur sur la phase instantanée du signal d'entrée. Le comparateur phase/fréquence qui est en réalité un multiplieur, génère un signal proportionnel à l'erreur de phase entre les signaux qui lui sont appliqués. Ensuite, le circuit de pompe de charge délivre un courant positif ou négatif pendant le temps où l'erreur subsiste. La tension résultante vient alors rétroagir sur le VCO afin d'ajuster sa

fréquence à la fréquence instantanée du signal d'entrée. Le filtre de boucle limite le bruit du système, mais aussi le temps de verrouillage.

2.6. Oscillateurs locaux agiles

Les oscillateurs locaux (OL) sont utilisés pour la génération ou la transposition de fréquence. Un oscillateur local doit avoir une très grande pureté spectrale et une forte stabilité en fréquence dans le temps. Un oscillateur local agile pour les systèmes SDR, doit être capable de couvrir plusieurs bandes de fréquence. Pour réaliser de tels oscillateurs agiles, il existe deux techniques qui sont actuellement couramment employées: la première c'est l'utilisation de matériaux de type Grenat d'Yttrium (YIG) qui présentent de nombreux avantages mais répondent à une commande magnétique. La deuxième technique est l'utilisation de diodes Varactor qui répondent à une commande électrique. La deuxième technique est la plus utilisée puisque elle est plus facile à intégrer, malgré le défaut d'une consommation de puissance importante et un facteur de bruit assez élevé. Une autre solution basée sur l'utilisation de MEMS existe [42]. Une autre approche remplace totalement les VCO (Voltage Controlled Oscillator) dans le synthétiseur de fréquence par une structure à base de DLL (Delay Locked Loop – Boucle à verrouillage de délai) [43] qui a permis de générer les fréquences de tous les standards visés.

3. Convertisseurs analogique/numérique et numérique/ analogique

L'étage de conversion analogique/numérique et numérique /analogique joue un rôle très important dans tout système de communication numérique, ces convertisseurs sont considérés comme des éléments critiques avec les amplificateurs et les filtres dans un système de transmission radio logicielle.

3.1. Convertisseurs analogique/numérique (CAN)

Un convertisseur analogique/numérique, est un montage électronique dont la fonction est de faire la transposition d'un signal analogique vers un signal numérique et codé sur plusieurs bits. Un CAN comporte essentiellement quatre éléments (figure II.5): un filtre anti-repliement pour supprimer les fréquences indésirables risquant de se mélanger au signal utile, un circuit d'échantillonnage/blocage pour échantillonner et maintenir le signal d'entrée à un niveau constant et stable pour la quantification, un quantificateur qui transforme le signal analogique en un signal numérique, et une mémoire tampon (*buffer*).

Pour la radio logicielle ou bien la radio cognitive, un convertisseur CAN doit avoir la possibilité de faire la numérisation de diverses formes d'onde avec des performances très élevées en précision. En plus ils doivent travailler à une cadence très élevée comme cela a été évoqué dans le chapitre I.

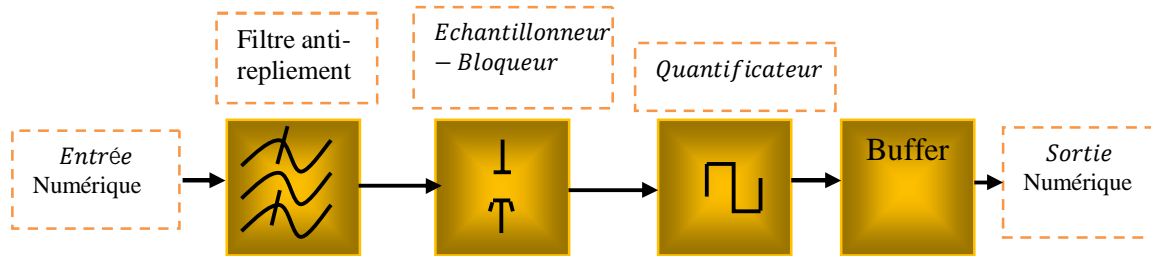


Figure II.5 : Schéma fonctionnel d'un CAN

Les CAN les plus communément utilisés dans les systèmes de transmission numérique sont les convertisseurs "à approximations successives", "flash", "pipeline" et "sigma-delta" [44] [45]. Les convertisseurs de type flash sont les convertisseurs les plus rapides, ils sont adaptés à des fréquences d'échantillonnage très élevées. L'origine de cette rapidité est que le CAN possède plusieurs comparateurs (ils quantifient le signal en le comparant à une suite de tensions de référence). Ce type d'architecture de CAN est capable de faire la quantification du signal à une résolution de n bits en un coup d'horloge. Cependant, cette rapidité s'obtient au prix d'une surface et d'une consommation très élevées. Pour un CAN de résolution n bits cette architecture utilise $2^n - 1$ comparateurs et 2^n résistances. C'est pour cette raison que leur utilisation est limitée aux applications qui n'exigent qu'une résolution modérée, n'excédant pas 8 bits.

Les convertisseurs à approximations successives réalisent la conversion par comparaison en plusieurs étapes, le résultat de la comparaison est stockée dans un registre à chaque étape, jusqu'à l'obtention du résultat final. Contrairement au CAN flash, les CAN à approximations successives nécessitent n cycles d'horloge pour une résolution de n bits.

Par rapport au CAN flash, elles sont donc très lentes, bien qu'elles présentent l'avantage d'une surface très réduite, une réalisation simple et une résolution très élevée (supérieures à 14 bits) [44].

Les CAN pipeline réalisent la conversion en plusieurs étages mis en cascade. Ainsi, à chaque front d'horloge, chaque étage réalise en parallèle une partie de la conversion. Chaque étage est composé d'un échantillonneur bloqueur, d'un CAN flash de faible résolution de k bits, d'un CNA de k bits aussi, d'un soustracteur et d'un amplificateur. Les CAN pipeline présentent un bon compromis entre la rapidité et la résolution.

Le fonctionnement des convertisseurs de type Sigma-Delta " $\Sigma\Delta$ " est fondé sur les principes de suréchantillonnage. Le but du suréchantillonnage qui est en effet un échantillonnage à des fréquences supérieures à la fréquence de Nyquist, est d'étaler le bruit de quantification sur une bande de fréquences plus large dont l'étalement spectral abaisse ainsi le niveau de bruit. Leur architecture comporte un décimateur qui représente une opération nécessaire dans le cas de « Down-conversion » (voir section 2.4), il peut

donc occuper une double fonction. Le tableau suivant effectue un comparatif des performances de ces différents types de CAN [44] [45].

Tableau II.1. Comparaison entre les principaux CAN de télécommunication.

Type de CAN	Flash	Approximations successives	Pipeline	Sigma-delta ($\Sigma \Delta$)
Avantages	Très rapide. Excellente bande passante.	Hautes résolution et précision. Très bonne bande passante. Faible consommation.	Débit élevé. Faible consommation.	Haute résolution. Filtrage numérique sur puce. Excellente résolution.
Inconvénients	Consommation électrique. Taille de la puce. Faible résolution.	Faible taux d'échantillonnage.	Nécessite une fréquence d'horloge élevée.	Faible taux d'échantillonnage.
Usage	Très haute vitesse sans contrainte de consommation.	Une résolution moyenne à élevée (8 à 16 bits). Faibles puissance et taille.	Haute vitesse 8 bits à 16 bits. Consomme moins que le flash.	Haute résolution. Vitesse basse à moyenne.

Parmi les architectures présentées, seuls les CAN pipeline et $\Sigma \Delta$ représentent les solutions les plus pertinentes en terme de résolution et de fréquence. Le convertisseur $\Sigma \Delta$ présente des caractéristiques indispensables pour une chaîne multi-standard, car il permet de réaliser la numérisation du signal ainsi que la sélection du canal de manière programmable [46].

3.2. Convertisseurs numérique/analogique (CNA)

Comme le CAN, le convertisseur numérique/analogique peut être considéré aussi comme un élément indispensable dans un émetteur SDR. Un convertisseur numérique/analogique, est un montage électronique dont la fonction est de faire la transposition d'un signal numérique, codé sur plusieurs bits vers un signal analogique (figure II.6) selon l'équation (2).

$$I = I_0[a_N 2^0 + a_{N-1} 2^1 + a_{N-2} 2^2 + \dots a_1 2^{N-1}] \quad (2)$$

Où I_0 est un courant de référence, et les coefficients « a_1 », « a_2 », ... « a_N » sont les entrées d'un convertisseur N bits, prenant les états logiques « 0 » ou « 1 » (le bit de poids fort $a_i = 0$ si le niveau logique est 0, et le bit de poids faible $a_i = 1$ si le niveau logique est 1).

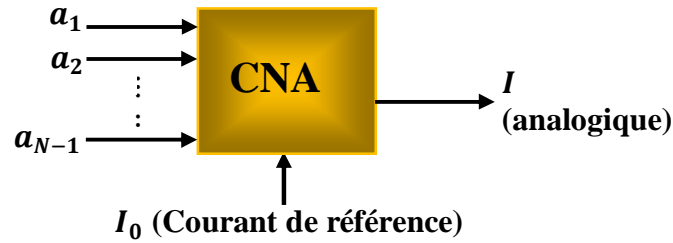


Figure II.6. Synoptique simplifié d'un convertisseur numérique/analogique N bits.

La précision des convertisseurs réside essentiellement dans la précision des résistances utilisées. Pour caractériser un CNA, on utilise généralement la notion de monotonie (la tension de sortie est une fonction croissante du code d'entrée).

4. Le frontal numérique (Traitement en bande de base).

Cette partie de la chaîne de transmissions SDR est la base de notre travail de thèse. Dans la figure II.7, un aperçu global sur les différentes opérations et traitements pouvant être effectués en bande de base ayant été donnés pour les différents normes et standards.

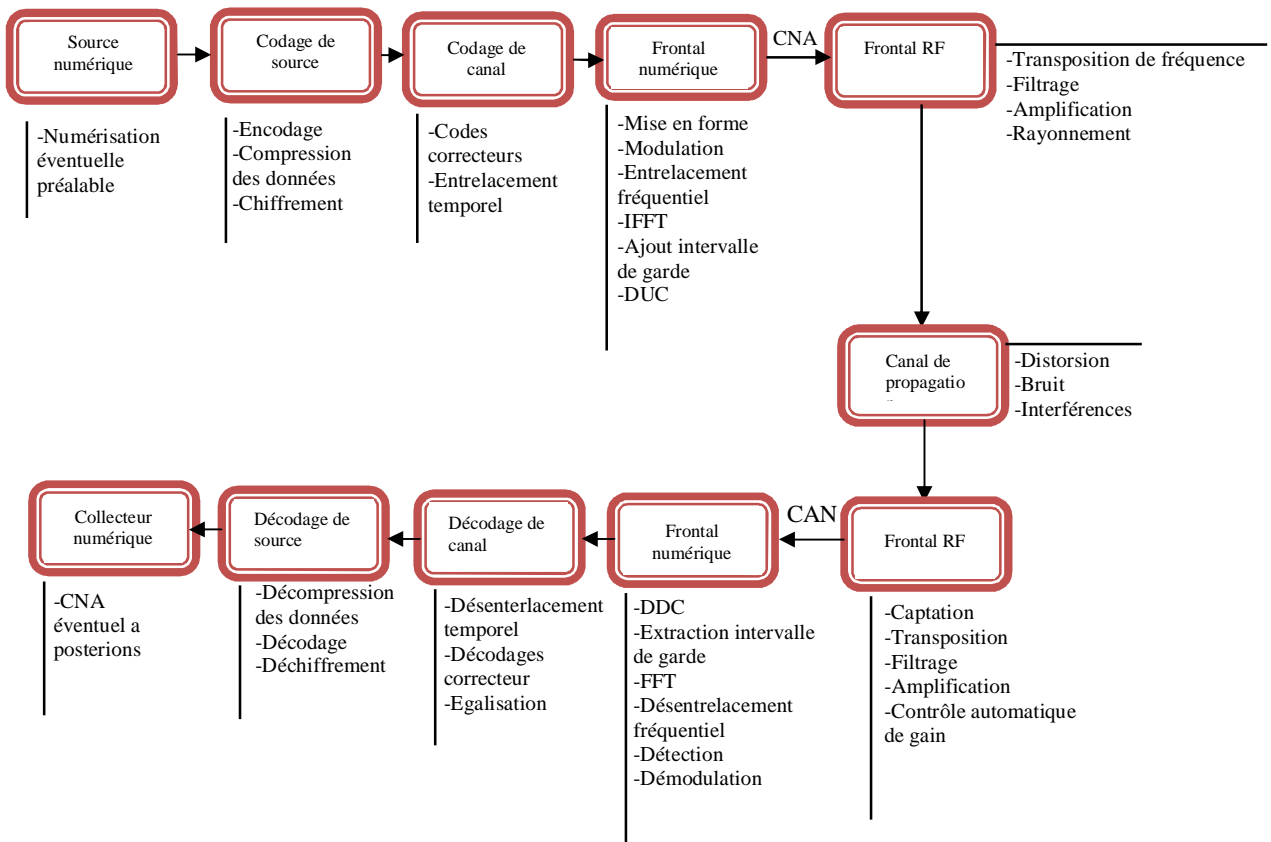


Figure II.7. Les différentes opérations et traitements en bande de base.

Cette figure présente à la fois des fonctions appartenant à la couche physique, et d'autres à la couche de liaison de données. La fonction du codage canal peut appartenir aux deux niveaux simultanément. D'une façon générale, la couche physique dans la chaîne d'émission reçoit les données de la couche Media Access Control - MAC (dont le rôle est

cit  dans la section 3 du chapitre I) et effectu e de multiples transformations. En r ception les  chantillons num riques bande de base obtenus apr s la conversion analogique num rique font l'objet de plusieurs traitements afin d'extraire les informations d sir es. Ces transformations et traitements sont le plus souvent d pendants des normes de transmission employ es.

Parmi les diverses op rations r alis es, on peut citer:

- **Le codage de source:** le r le consiste   r duire la taille du message avec ou sans perte d'information afin de minimiser l'occupation de bande du signal et permettre d'envoyer plus de donn es;

- **Le codage de canal:** l'objectif du codage de canal est de prot ger le message contre les perturbations du canal de transmission. En effet, la mobilit  dans les communications sans fils est une source de nombreux probl mes qui peuvent nuire   la bonne r ception des donn es. Par cons quent, en ajoutant en  mission de la redondance au message, il se forme des bits ajout s (codage canal) ceci afin de permettre au r cepteur de d tecter et/ou de corriger les erreurs induites par le canal de transmission (d codage canal). Parmi les codes les plus utilis s, on a les codes de redondance cyclique ou CRC, les codes convolutifs, les turbocodes, qui se basent sur des codages convolutifs en parall le (ou s rie), et les codes LDPC (*Low Density Parity Check codes*). Le code convolutif est le plus utilis  dans les communications modernes, il se base sur des op rations r cursives simples qui g n rent de l'information en fonction des donn es cod es. Pour faire le d codage du code convolutif, on utilise l'algorithme de Viterbi, qui vise   minimiser l'incertitude des donn es re ues. Par rapport au codage, le d codage se r alise avec une forte complexit , tant en puissance de calcul qu'en m moire.

- **l'entrelacement des donn es:** la technique de l'entrelacement revient   permuter les symboles cod s avant leur transmission. Cette op ration rend les codes correcteurs plus rentables et efficaces;

- **la modulation:** la modulation effectu e une conversion de bits en symboles. Pour une modulation M-aire, on associe   chaque mot de n bits un signal $x_i(t)$, $i = 1, 2, \dots, M$ de dur e T choisi parmi les $M = 2^n$ signaux. Par cons quent, la modulation permet d'augmenter le d bit par la transmission d'une plus grande quantit  d'informations sur une bande passante d finie. Par exemple dans les modulations BPSK, O-QPSK, 16-QAM et 64-QAM, chaque point de la constellation repr sente respectivement un, deux, 4 et 6 bits.

Les modulations QAM sont plus sensibles aux modifications de phases et de fr quences, ainsi la probabilit  d'erreur en r ception augmente avec l'augmentation de la taille de la constellation. Cependant, gr ce aux algorithmes de synchronisation sophistiqu s, au d veloppement des filtres de grande capacit  de filtrage et   la combinaison du codage

aux modulations QAM, on peut utiliser des modulations QAM d'une constellation de grande taille.

- **les techniques de multiplexage:** plusieurs techniques cherchent à optimiser la bande passante, nous citons en particulier, les techniques basées sur l'utilisation conjointe de plusieurs porteuses, et celles basées sur l'étalement du spectre. Dans le premier cas, les données sont transmises sur plusieurs porteuses orthogonales: *OFDM (Orthogonal Frequency Division Multiplexing)*. L'OFDM avec cette orthogonalité peut maximiser l'utilisation du spectre (nombre élevé de bit par Hertz). Dans le second cas, on utilise des codes d'étalement pour étaler le signal sur une large bande passante. Ce signal étalé devient sous le niveau de bruit. Cette technique est utilisée dans les communications 3G, qui l'implémentent au travers du *WCDMA (Wideband Code Division Multiple Access)*.
- **au niveau du récepteur,** les fonctions de démodulation et de décodage sont les fonctions inverses respectives des fonctions de modulation et de codage situées du côté émetteur.

5. Les plateformes de la radio reconfigurable

Nous étudions dans ce paragraphe les circuits utilisés en bande de base pour le SDR: les circuits spécialisés de type Application-Specific Integrated Circuit (ASIC) et dérivés, les processeurs de type General Purpose Preprocessor (GPP), les FPGA et les DSP (Digital Signal Processor).

5.1. Composants matériels

Les composants matériels sont utilisés pour leurs performances très élevées. Ils permettent de résoudre de fortes contraintes temps réel. On peut distinguer deux types: les ASIC et les FPGA. Les ASIC sont des composants spécialisés développés pour une application spécifique. Les FPGA, sont des circuits intégrés qui peuvent être reconfigurés après sa fabrication.

5.1.1. Les ASIC (Application-Specific Integrated Circuit)

Les ASIC (circuit intégré propre à une application, en français), ce sont des circuits intégrés spécialisés, que sont développés pour un seul client et selon les spécifications de ce dernier. En général, un ASIC regroupe un grand nombre de fonctionnalités uniques ou sur mesure. Les ASIC ne sont pas reconfigurables (reprogrammables), c'est-à-dire ils sont figés (fixés) pour un nombre de fonctionnalités spécifiques. En raison de leurs caractéristiques spéciales, ils sont utilisés couramment dans la SDR, bien qu'ils soient des composants non modifiables contredisant ainsi le principe de base des SDR ; c'est-à-dire la reconfigurabilité et la flexibilité. En effet, les ASIC, présentent une puissance de calcul très élevée, une faible consommation d'énergie, une haute intégration et une petite taille. Ils montrent aussi un coût de fabrication faible pour des quantités élevées. C'est pour ces

raisons que les portables représentent un domaine d'application typique de ces composants.

Le manque de la flexibilité et le cycle de développement très long des ASIC constituent leur principal inconvénient [47]. À noter qu'il existe également des ASSP (*Application Specific Standard Product*), circuits type ASIC légèrement programmables, regroupant un grand nombre de fonctionnalités pour satisfaire à une application généralement standardisée. En général, les ASIC et les ASSP sont basés sur les mêmes processus de conception et technologies de fabrication. La différence est que les ASIC sont destinés à une compagnie spécifique, tandis que les ASSP sont vendus à de multiples clients. Un ASSP pour GSM issu d'un fabricant unique est utilisé comme circuit de base par différents fabricants de téléphones portables qui se distinguent sur d'autres aspects tels que le logiciel, l'écran, le boîtier, la batterie, etc. La programmation d'un ASIC s'effectue avec un langage de bas niveau, et les tests de validation sont intensifs. Une erreur qui apparaît après sa production peut provoquer la nécessité de repasser par la phase de développement et coûte par conséquent très cher [47].

5.1.2. Les FPGA (Field-Programmable Gate Array)

Les FPGA (Les réseaux de portes programmables *in situ*, en français), est un ensemble de blocs logiques élémentaires que l'utilisateur peut interconnecter pour réaliser les fonctions logiques de son choix. Un FPGA est composé d'un nombre élevé de blocs logiques configurables ou CLB (*Configurable Logic Block*). Un CLB est constitué au minimum d'une table de correspondance (LUT – *Look-Up Table*) et d'une bascule. Une LUT est considérée comme la partie fondamentale du FPGA, elle agit en tant que générateur de fonction et sert à implémenter des équations logiques (généralement de 4 à 6 entrées et une sortie), une petite mémoire, un multiplexeur ou un registre à décalage. L'architecture d'un CLB se diffère en fonction du FPGA et de la technologie utilisée. Sur une famille de FPGA Xilinx Virtex 4 par exemple, chaque CLB dispose de deux LUT à quatre entrées (LUT4), alors que pour les Virtex 5, et 6, chaque CLB dispose de quatre LUT à six entrées (LUT6), soit la possibilité d'utiliser six variables d'entrée pour chaque fonction au lieu de quatre, et deux fois plus de fonctions combinatoires, alors que dans le Virtex 7 chaque CLB dispose de quatre LUT, qui peuvent être configurés soit comme une LUT à 6 entrées avec une sortie, ou comme deux LUT à 5 entrées avec des sorties séparées. Les CLB sont interconnectés par un système matriciel configurable de liaisons haute vitesse, et sont reliés à des blocs d'E/S programmables ou (IOB – *Input Output Blocks*). Les blocs d'entrées/sortie sont utilisés pour relier le FPGA à des signaux externes, et ainsi à des mémoires externes rapides dans le cas où la mémoire interne est insuffisante. Deux types de mémoire interne existent dans le FPGA; de petites mémoires rapides double accès (pour les tampons et les tables de coefficient), et de grandes mémoires simple accès (pour le stockage de grandes tables de données). Les blocs dédiés

dans le FPGA, permettent de gérer l'horloge, l'aident à la programmation, à réaliser facilement de nombreuses opérations tels qu'une PLL et les traitements numériques des signaux (blocs DSP). En effet, la présence des multiplicateurs, des accumulateurs et des mémoires facilite la réalisation de la convolution, la corrélation et le filtrage. La figure II.8 présente un exemple d'architecture de FPGA.

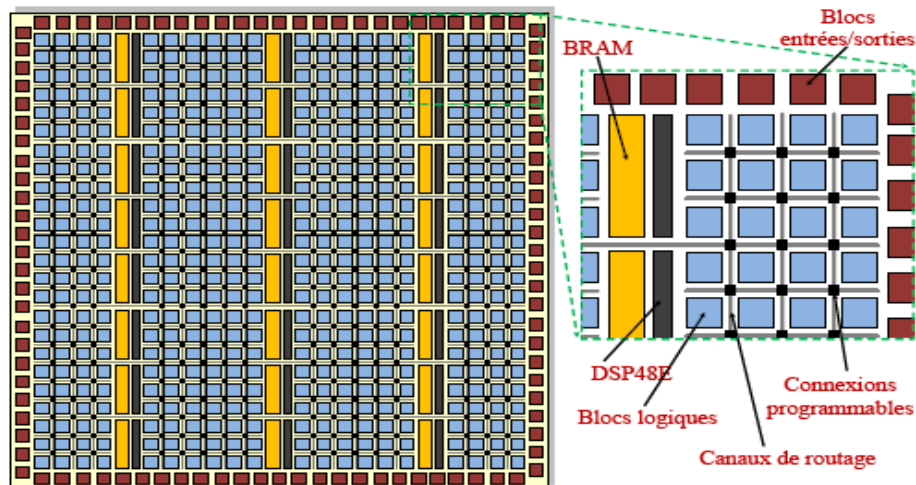


Figure II.8. Exemple d'architecture de FPGA (Architecture, Interconnexions et CLB).

L'aspect le plus séduisant des FPGA est leur flexibilité et leur reconfigurabilité, ce qui respecte pleinement le principe de base des SDR. En plus, ils présentent une consommation d'énergie faible et une puissance de calcul grande par rapport aux processeurs DSP et GPP. Ceci revient à un grand nombre d'unités de traitements et de mémoires tampons (*buffers*). Par exemple, la famille des FPGA Virtex-7 de Xilinx [48], contient entre environ 45 000 et 300 000 slices (CLB), chaque slice contient quatre LUT et huit bascules. Nous avons aussi comme exemple le FPGA Stratix V de Altera, qui est un FPGA de haute performance [49]. Cette énorme ressource permet d'implémenter plusieurs fonctions et même des standards et normes. Par exemple L. Bisdounis et al [50] ont implémenté sur un FPGA la partie bande de base et la couche MAC du standards HIPERLAN/2 et IEEE 802.11a.

La concurrence entre les principales firmes des FPGA notamment Xilinx et Altera contribuent à une amélioration permanente des performances de ces composants. Pour cette raison, la performance actuelle des FGPA est un peu approchée de celle des ASIC. En plus, la tendance pour les circuits FPGA est la possibilité de réaliser des systèmes sur puce (SoC) en utilisant des "composants virtuels" et de concevoir ainsi des blocs de propriété intellectuelle IP qui sont par exemple des fonctions VHDL/Verilog génériques réutilisables, on parlera alors de « System On Programmable Chip » (SoPC). Les systèmes de type SoPC peuvent intégrer sur le même FPGA un ou plusieurs processeurs

softcore ou hardcore avec ses périphériques et une mémoire interne. Cette mémoire est utilisée pour sauvegarder les données (Data) ou le code exécutable correspondant à l'application logicielle déployée sur le ou les processeurs. Parmi les processeurs embarqués intégrables dans un système SoPC, on peut citer les solutions softcore propriétaires: MicroBlaze et nios développées par Xilinx et Altera respectivement. D'autres processeurs performants (hardcore) sont issus du monde opensource tels que: processeurs Leon 2/3, plasma, ARM,...

Les FPGA se programment grâce à leurs LUT et leur réseau d'interconnexion en utilisant le plus souvent un langage de description de matériel tel que le VHDL (acronyme de VHSIC-HDL: Very High Speed Integrated Circuit Hardware Description Language) ou Verilog. L'outil de développement (ISE par exemple [51]) transforme cette description en un fichier de configuration du FPGA en plusieurs étapes (figure II.9): Lors de l'étape de synthèse, la description HDL du circuit est transformée en un assemblage (Netlist) de primitives de base (portes logiques, bascules, etc.).

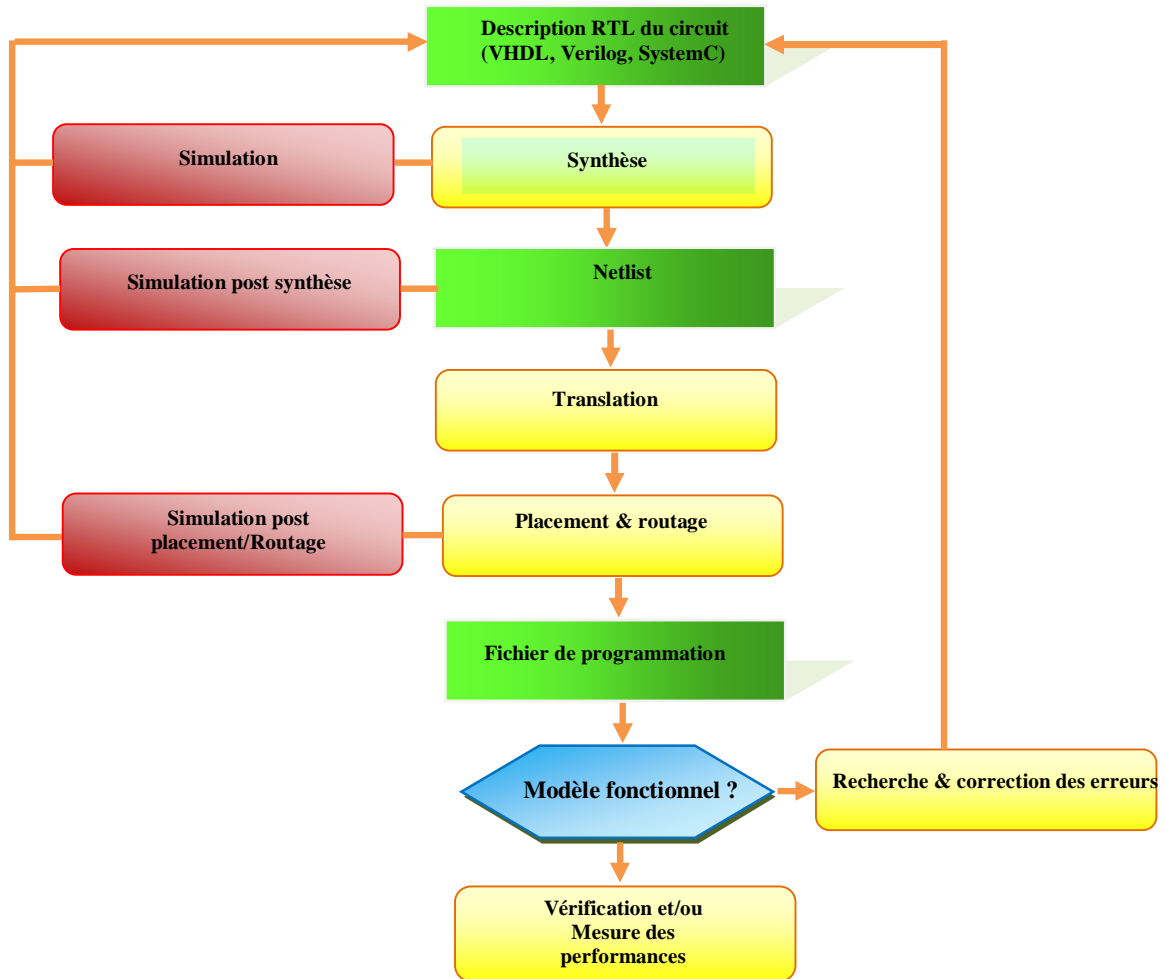


Figure II.9. Cycle de programmation d'un FPGA.

A ce point, il est possible d'effectuer une étape de simulation post-synthèse, l'intérêt de cette étape étant de vérifier les propriétés temporelles du circuit. Il vient ensuite l'étape de placement qui place les éléments de la Netlist (correspondant maintenant à des cellules) physiquement sur le circuit FPGA. Une fois l'emplacement physique de chaque cellule fixé, le routeur effectue le calcul des interconnexions entre les cellules physiques en respectant la connectivité des cellules comme indiqué dans la Netlist donnée en entrée. Une fois la phase de placement et routage terminée, le fichier de configuration (bitstream) du FPGA peut être généré. Le bitstream est une image binaire du circuit FPGA, il y détermine pour chaque élément (cellule logique, table de correspondance LUT, élément de routage, cellule mémoire) l'état logique. L'exécution proprement dite du circuit est possible après que le fichier ait été chargé sur le circuit FPGA (on parle ici de la reconfiguration du FPGA soit pour placer une nouvelle application, soit pour gérer l'évolution d'une application). Il est alors possible de vérifier les fonctionnalités du modèle et d'effectuer des mesures de performances.

Pendant l'étape de reconfiguration, un FPGA doit être dans un état inactif, puis se reconfigurer, enfin être remis en état opérationnel, bien que d'autres séries de FPGA supportent un mode de configuration partielle, ne nécessitant pas le composant dans un état inactif [52].

5.2. Composants logiciels

Ils offrent par rapport aux composants matériels une flexibilité très élevée. On peut distinguer les DSP et les GPP. La programmation s'effectue le plus souvent en langage de haut niveau, ce qui simplifie le processus de développement.

5.2.1. Les DSP

Les processeurs de traitement numérique du signal ou DSP (*Digital Signal Processor*) sont des microprocesseurs optimisés et adaptés pour exécuter des applications de traitement numérique du signal (filtrage, convolution, etc.). Ils sont classés donc comme des processeurs spécifiques. Les DSP occupent de nombreux secteurs de l'électronique, car leur architecture interne est dimensionnée pour le calcul intensif, ce qui permet d'exécuter efficacement des opérations de multiplications/accumulations (MAC), avec des vitesses de traitement plus rapides.

Les processeurs DSP ont des architectures capables de réaliser des opérations sur des nombres à virgule fixe, et à virgule flottante. En effet, les unités arithmétiques et logiques ne sont pas construites de la même manière. Le calcul flottant requiert une architecture plus complexe qui conduit à une fréquence de fonctionnement généralement plus faible. Par conséquent, les flottants sont utilisés seulement lorsqu'une grande précision et une grande dynamique sont requises sur les valeurs traitées [47]. L'architecture peut ainsi inclure des interfaces série, PCI, DMA (Direct Memory Access) et une mémoire.

À l'origine, les DSP étaient composés d'un seul cœur qui exécute des opérations MAC aussi vite que possible. Les DSP modernes peuvent utiliser plusieurs cœurs de calcul, en plus la possibilité d'embarquer des coprocesseurs dédiés (par exemple coprocesseurs de décodage Viterbi). Le DSP de Texas Instruments TMS320C6416 par exemple est un DSP à deux cœurs de calcul (A et B) qui ont chacun quatre unités (L, S, M et D). Huit instructions peuvent donc être exécutées en un cycle (voir datasheet de TMS320C6416). Le DSP d'Analog Devices « Blackfin ADSP-BF561 » est aussi un DSP à double cœur optimisé pour le traitement du signal de haute performance, il peut effectuer jusqu'à 2,4 Giga-MAC (milliards de multiplication/accumulation) par seconde avec une fréquence d'horloge maximale de 600 MHz [53] [54].

Ces DSP récents comportent trois techniques qui permettent d'augmenter leurs performances:

- l'architecture VLIW (Very Long Instruction Word) lit des instructions de 256 bits pour fournir à chaque cycle d'horloge jusqu'à huit instructions de 32 bits aux huit unités fonctionnelles;
- les unités de calcul 32 bits du TMS320C6416 par exemple, permettent d'effectuer une opération 32 bits, deux opérations 16 bits ou quatre opérations 8 bits, grâce aux instructions SIMD (Single Instruction Multiple Data). Les unités fonctionnelles peuvent donc être exploitées de manière intensive en utilisant toute la largeur disponible;
- les transferts entre les périphériques et la mémoire et également entre la mémoire interne et la mémoire externe sont réalisés par le DMA (Direct Memory Access), qui permet de décharger le cœur de calcul, et de paralléliser l'accès mémoire et les traitements.

Les DSP peuvent être programmés avec un langage de haut niveau, tel que C ou C++, et sont accompagnés d'outils de développement performants. Ces outils de développement, les compilateurs et les simulateurs, sont capables d'optimiser fortement un programme et d'informer le développeur sur le résultat en ajoutant des commentaires dans le programme (nombre de cycles par exemple). Ils permettent aussi d'obtenir des informations sur la charge de calcul, le comportement des mémoires caches, et le temps d'exécution [47]. D'après Jean-Philippe Delahaye [55], les DSP modernes de haute performance avec leurs souplesses et leurs productions en masse, constituent un choix plausible pour le SDR.

5.2.2. Les processeurs à usage général (GPP)

Les processeurs à usage général ou GPP (*General Purpose Processor*) sont les processeurs qui se trouvent dans la plupart des ordinateurs. Ils sont classés comme des processeurs généralistes [47]. Les architectures des GPP les plus répandues sont les CISC (Complex Instruction Set Computer) comme le processeur Pentium d'Intel ou AMD

Athlon, et les RISC (Reduced Instruction Set Computer), comme les Sparc, MIPS, ARM et PowerPc. Ces instructions effectuent des opérations telles que la multiplication, l'addition ou le stockage, mais ne sont toutefois pas optimisées pour un usage particulier.

Comme pour les DSP, la plupart des GPP sont accompagnés de nombreux outils de développement puissants (compilateurs, debuggers) [55].

Plusieurs avantages des GPP les placent parmi les composants plausibles pour le SDR, tels que leur grande flexibilité, leur simplicité de configuration et surtout la possibilité de réutilisation des programmes pour d'autres applications. Cependant, les inconvénients des GPP, sont décrits par le manque d'une puissance de calcul suffisante pour gérer le flux du signal radio de haut débit et l'inadaptation au temps réel pour les applications complexes [47]. Par conséquent, il est préférable d'utiliser les GPP lorsque le signal est en bas débit, c'est à dire dans les opérations finales d'un récepteur ou les opérations initiales de l'émetteur. Les GPP présentent d'autres inconvénients tels que la grande consommation, la dissipation d'énergie élevée, l'encombrement et le prix.

Toutefois, les GPP subissent une grande évolution au niveau des performances. En effet, les nouveaux processeurs disposent d'unités de traitement vectoriel (SIMD « Single Instruction Multiple Data ») haute vitesse. Ils sont ainsi capables d'optimiser l'exécution d'un programme à la volée pour améliorer les performances des applications qui ne sont pas optimisées à la base, pour un type de processeur particulier. On peut résumer les techniques utilisées par les GPP modernes comme suit:

- **Pipeline d'exécution:** elle permet de découper les opérations en plusieurs étapes afin d'augmenter la fréquence des processeurs, chaque étape étant plus courte. Le processeur peut alors contenir plusieurs instructions, chacune à une étape différente.
- **Super-pipeline:** elle permet aux nouvelles versions des GPP d'effectuer plus d'un milliard d'opérations mathématiques par seconde.
- **Super-scalaires:** plusieurs unités de calcul dédiées sont embarquées (calcul arithmétique et logique, calcul flottant, accès mémoire, branchement). Cela permet d'exécuter plusieurs instructions par cycle grâce au fonctionnement en parallèle de ces unités.
- **Unité de ré-ordonnement des instructions et des accès mémoire:** Les dépendances de données entre les instructions séquentielles d'un programme sont analysées, puis les instructions sont parallélisées sur les différentes unités de calcul, et réordonnées si besoin. De même que pour les instructions, les accès aux données en mémoire sont réordonnés.
- **Prédiction de branchement:** Un branchement conditionnel dans un programme est très préjudiciable. En effet, la condition doit être calculée avant d'effectuer le

branchement, induisant une perte de cycles. Pour les limiter, le branchement est prédit. Une analyse statistique en temps réel permet de prédire un branchement et de garder les unités de calcul actives.

- **Mémoire cache:** La mémoire cache est utilisée afin de limiter le nombre d'accès à la mémoire centrale du système et de minimiser, par conséquent, l'utilisation du bus et de la consommation d'énergie du système.

Le microprocesseur Core i7 multi-cœur, multi-thread (multi-tache) d'Intel, est un processeur pouvant s'avérer utile dans les applications SDR.

5.2.3. L'accélération matérielle: GPU (Graphics Processing Unit).

L'accélération matérielle est un circuit intégré dédié qui effectue une fonction spécifique (initialement effectuée par le processeur CPU) de façon plus efficace. L'accélération matérielle est implémentée soit sur la carte mère, soit sur une carte fille, soit intégrée dans le CPU. Il existe différents types d'accélérateurs matériels pour différents usages, les plus reconnus pour les utilisateurs des PC, sont les cartes d'accélération audio (cartes son) et les cartes d'accélération vidéo. Les premières sont utilisées pour la numérisation et le traitement du signal audio et les deuxièmes sont utilisées pour le traitement du signal vidéo, décompression MPEG, etc. La carte d'accélération vidéo embarque un ou plusieurs processeurs graphiques, ou GPU (*Graphics Processing Unit*). Généralement, les GPU s'interfaçent avec la carte mère d'un ordinateur via un port PCI-express. De plus, les GPU partagent la mémoire vive avec la carte mère, bien qu'il existe des GPU qui disposent de leur propre mémoire vive.

Le GPU se compose d'une multitude (plusieurs centaines, voir milliers) de processeurs indépendants capables d'exécuter simultanément plusieurs calculs (figure II.10).

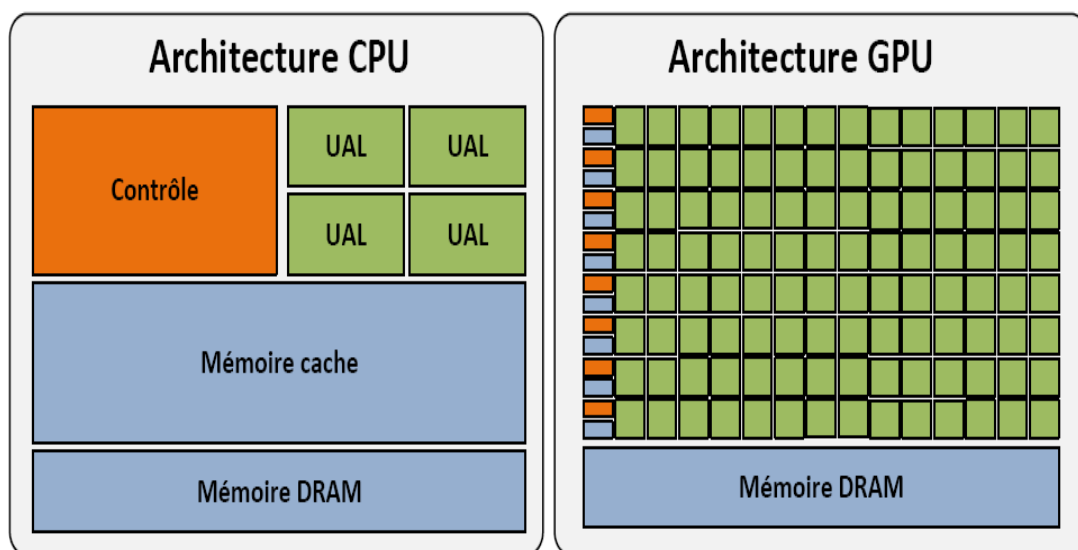


FIGURE II.10. Architectures CPU et GPU.

C'est pour cela que les GPU sont parfaitement adaptés aux traitements parallèles avec une performance très supérieure par rapport aux processeurs GPP. Un GPU offre une puissance de calcul de six fois supérieure à un coût relativement réduit que celle des GPP [47]. Cela les rend très utilisables à la radio logicielle restreinte [56].

Il a été montré que l'utilisation d'un GPU peut améliorer les performances et l'exécution dans une application SDR et ceci beaucoup mieux que l'utilisation d'un CPU [56] [57]. De même, J. KIM et al [58] réalisent un terminal SDR pour le WiMAX à l'aide d'un ordinateur équipé à la fois d'un PC et d'un GPU. En comparant les performances d'une carte graphique GeForce 9800GTX de Nvidia et d'un DSP TMS320C6416 de Texas Instruments, ils ont trouvé que le décodage d'un code convolutif avec l'algorithme de Viterbi par la carte graphique est 90 fois plus rapide que le DSP. Ainsi, le débit de décodage de Viterbi implémenté sur le GPU est de 181.6 Mb/s comparé à 3.07 Mb/s pour le DSP [58]. Il faut noter que Nvidia fournit également un kit de développement basé sur le langage C (CUDA: Compute Unified Device Architecture) qui permet le débogage et la simulation sur CPU.

5.3. Analyse des plateformes de la radio reconfigurable.

La radio flexible est clairement dépendante de la capacité d'un système à modifier son mode de fonctionnement. Les composants logiciels présentent plus de flexibilité par rapport aux composants matériels, il semble donc que les cibles logicielles sont les plus adaptés pour le SDR. En effet, un processeur comme les General Purpose Processor (GPP), les GPU et les DSP (Digital Signal Processor) sont virtuellement capables d'exécuter n'importe quelle opération, par changement du programme qui la décrit. Cependant, il est difficile d'ignorer les aspects de performances et de consommation dans ces composants. Dans les systèmes embarqués, cette contrainte de consommation est, et restera forte, en effet elle est supérieure à celle d'un ASIC et FPGA, surtout pour les traitements intensifs de données. En plus, les performances de ces processeurs sont souvent insuffisantes face aux besoins de traitement du signal des applications récentes. Bien qu'il existe des processeurs spécialisés de traitement du signal comme les DSP, ils ne peuvent pas prendre en charge l'ensemble des traitements d'une chaîne radio [55].

Dans une application réelle de la radio logicielle avec les cibles logicielles telles que les GPP, GPU et DSP, la modification d'une application est faite par un simple téléchargement d'un nouveau programme (code) applicatif. Cependant, cela pose le problème de la limitation de mémoire disponible pour le stockage des données et des fichiers de configuration.

D'autres problèmes résident dans le risque sur l'intégrité de l'équipement lors de la phase de téléchargement et sont multiples et bien connus du monde de l'informatique tels que:

- bogues logiciels,

- transmission de virus informatiques par téléchargement,
- failles logicielles.

Par conséquent, il n'est pas toujours nécessaire d'utiliser une cible logicielle afin d'avoir la flexibilité pour le SDR. Les cibles matérielles comme les ASIC sont des implantations figées et spécifiques à l'application développée, qui montrent des performances extraordinaires avec une consommation réduite. Cependant, leur point faible est le manque de la flexibilité, qui est une caractéristique indispensable dans la définition de la radio reconfigurable. Pour réaliser un compromis entre la flexibilité et la performance, il existe des solutions reconfigurables comme les circuits de type FPGA. Les FPGA montrent une puissance de calcul élevée par rapport aux DSP ou aux GPP, cette performance résulte du fait que les FPGA autorisent des traitements quasiment en parallèle, c'est à dire qu'après un ou quelques cycles d'horloge, leur valeur de sortie est disponible en fonction de la valeur d'entrée. En revanche, les DSP et les GPP travaillent essentiellement en mode série. Ainsi les FPGA sont flexibles par rapport aux ASIC et les performances des FPGA modernes ont presque atteint le niveau de performance des ASIC. Ils peuvent donc remplir les compromis flexibilité/performance et constituer un choix indispensable pour le SDR. Il existe d'autres techniques et des solutions prometteuses pour avoir la flexibilité et la performance attendue par la radio logicielle (SDR) que nous verrons dans les sections suivantes.

6. Solutions hybrides (hétérogènes)

Comme toute évolution technologique, son utilisation (GPP, DSP/GPU, ASIC, FPGA) présente des avantages et des inconvénients. L'une des solutions envisagées pour compenser les inconvénients est l'utilisation de la plateforme hybride. La radio logicielle flexible nécessite l'utilisation des processeurs généralistes pour obtenir une flexibilité maximale du système qui nécessite alors un coût élevé en consommation. Dans les systèmes embarqués, cette contrainte de consommation est forte. Or les ASIC, sur lesquels sont basés les systèmes actuels, souffrent d'un manque de flexibilité à cause de sa paramétrisation réduite. Les systèmes à base de FPGA offrent des performances intéressantes en plus de la flexibilité. C'est pour cette raison que les systèmes de transmissions dédiés aux applications de la radio logicielle possèdent généralement une architecture hétérogène, car ces plateformes sont constituées de divers éléments ayant beaucoup d'avantages complémentaires (figure II.11).

Il existe quelques travaux sur les plateformes hétérogènes, nous citons à titre d'exemple la plateforme Kansas University Agile Radio (KUAR) [59] qui est une plateforme radio logicielle disposant d'un processeur Pentium M de fréquence de 1.4 GHz avec une mémoire DDR2 SDRAM de 1GB et un FPGA Virtex2 de Xilinx [59].

L'Institut national japonais de la technologie de l'information et des communications (NTIC) a construit une plate-forme de radio logicielle restreinte (NTIC plateforme SDR) pour les réseaux mobiles de prochaines générations. La plate-forme dispose de deux processeurs embarqués, quatre FPGA Virtex 2 de Xilinx avec module RF qui pourrait soutenir 1,9 GHz à 2,4 GHz et de 5,0 GHz à 5,3 GHz. Le traitement du signal a été partagé entre le CPU et le FPGA. À cette fin, un certain nombre de normes commerciales ont été mises en œuvre dans cette plate-forme, par exemple 802.11a/b/g, la radiodiffusion numérique terrestre, le WCDMA et le système de communication OFDM.

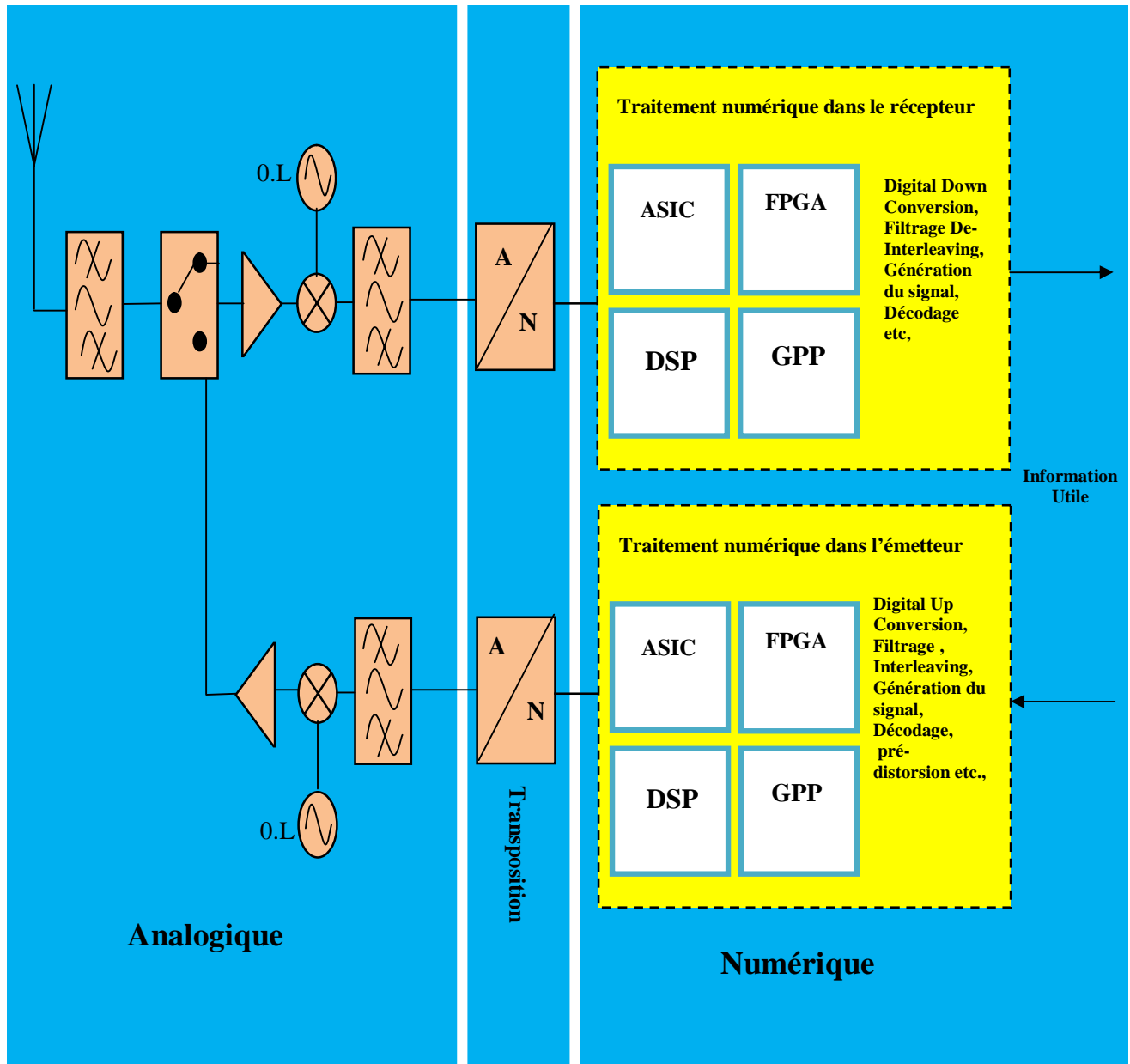


Figure. II.11. Structure possible d'une radio logicielle.

D'autres plateformes sont à base d'un FPGA et d'un DSP où le FPGA assure la conversion entre le haut débit des CAN/CNA et un débit plus faible, permettant ainsi au DSP de réaliser l'intégralité des traitements.

Le problème des plateformes hétérogènes réside dans la difficulté à les utiliser (difficulté pour gérer cette intégration). En effet, l'intégration de processeurs ayant chacun leur propre interface d'accès à la mémoire, leur propre jeu d'instruction, leur propre ordre dans lequel les octets sont organisés en mémoire ou dans une communication ("boutisme" ou endianness), leur vitesses de traitement, leur méthode d'acheminement des données dans l'architecture, présente de grandes difficultés et même un défi. Certaines approches utilisent un GPP pour gérer cette intégration, où le GPP joue le rôle d'un contrôleur qui donne les ordres et les données aux autres processeurs. De plus, il n'existe pas un environnement de travail unique permettant de concevoir de tels systèmes, donc malheureusement, le concepteur doit recourir à de nombreux logiciels et langages de programmation lui permettant de programmer les différents composants.

7. L'approche de paramétrisation

Cette approche est beaucoup plus générale, pourtant c'est une technique plus efficace pour l'implantation des terminaux multistandards [60]. Cette approche vise à trouver des points ou caractéristiques communs entre les traitements de différents standards dans un premier temps, puis d'en proposer des architectures communes (génériques) et flexibles. Ces architectures de traitements communs restent figées après la conception, mais elles peuvent prendre en charge un ensemble de fonctionnalités grâce à un simple jeu de paramètres, d'où le nom de paramétrisation [5] [60]. L'opposé d'une approche de type Velcro permet de diminuer le nombre d'éléments à implémenter et constitue une méthode de reconfiguration en elle-même [6] [7] [61]. Selon la méthodologie de paramétrisation, les aspects communs des différents standards deviennent un élément de traitement commun qui pourrait être installé dans le dispositif et exécuté par un simple appel.

La paramétrisation se décline selon deux approches: l'approche théorique et l'approche pragmatique. **L'approche théorique** consiste à lister de façon hiérarchique tous les appels de fonctions possibles dans un terminal (Par exemple, la modulation OFDM fait appel à la FFT, qui elle même fait appel à l'opérateur papillon et ce dernier faisant appel aux opérateurs arithmétiques classiques). La démodulation WCDMA comme un autre exemple, fait appel à un récepteur Rake, qui lui même fait appel à l'unité de désétalement et cet dernier faisant appel aux opérateurs arithmétiques classiques. Ensuite, l'ensemble des fonctions réalisées par un terminal multi-standards est représenté par un diagramme dont chaque sommet représente un élément de traitement (Processing Element- PE) fonctionnel qui occupe un niveau de granularité donné. Cette représentation en graphe permet de sélectionner un chemin optimal selon un processus d'optimisation basé sur des

coûts (consommation, surface, délais...). Ce chemin optimal permettant de privilégier l'appel à certains opérateurs, qui deviennent alors communs. Le processus d'optimisation est basé sur une fonction de coût/performance, qui calcule le choix de chaque jeu d'opérateurs. Cette fonction de coût lorsqu'elle est optimisée permettra de sélectionner le jeu d'opérateurs ayant le coût minimal. La fonction de coût qui a été proposée par S.T. Gul [62] prend comme paramètre: le coût de fabrication (Building Cost, BC), le coût de calcul (Computational Cost, CC), et le nombre d'appels (Number of Calls, NoC). En général, par cette approche on peut définir de nouveaux opérateurs à partir d'une combinaison d'opérateurs déjà présents dans le graphe considéré. Par conséquent, cette approche nécessite une deuxième approche qui permet de définir les opérateurs communs. Cette approche dite **l'approche pragmatique**, que nous privilégions dans cette thèse. Elle consiste dans un premier temps à identifier dans la littérature les traitements similaires qui peuvent partager des ressources communes (tant au niveau algorithmique qu'architectural), puis dans un second temps à réaliser un opérateur générique qui devra alors être reconfigurable par changement de paramètres. C'est pour cela que cette approche arbore un côté plus pratique dans l'application de la technique des opérateurs communs (section 7.2) [60][62]. Mais il faut noter que les deux approches convergent vers le même objectif i.e l'identification des meilleurs points communs entre les différentes normes, ce qui conduit à une conception multi-standard reconfigurable flexible.

Deux approches peuvent être distinguées en paramétrisation: l'approche fonction commune et l'approche opérateur commun.

7.1. Fonctions communes (FC)

Le niveau de ce type de paramétrisation est de granularité élevé, puisque la paramétrisation se fait au niveau fonction. Ici, on cherche à définir des fonctions paramétrables communes à plusieurs standards, où tous les composants dédiés à la même fonctionnalité ont été regroupés dans le même FC. Par exemple, on dédie à la fonction commune « codage canal », les différents types de « codage canal » utilisés pour les différents standards. Une liste de paramètres est associée à la fonction et détermine son mode de fonctionnement.

On peut prendre l'exemple d'Anne Wiesler et F. Jondral [6] [7] [63], qui sont considérés comme pionniers dans ce domaine. En effet, dans leurs travaux, ils ont proposé des structures paramétrables pour les fonctions de modulation et d'égalisation communes aux standards GSM, DECT (Digital Enhanced Cordless Telecommunications), UMTS UTRA/FDD et IS-136. Cet exemple de fonction commune est illustré dans la figure II.12. A.-R. Rhiemeier [64] qui est dans la même équipe de recherche, a proposé une fonction

commune de codage canal. Nous avons aussi l'architecture nommée VITURBO, qui est une structure unifiée de turbo décodage et de Viterbi pour les systèmes 3G [65].

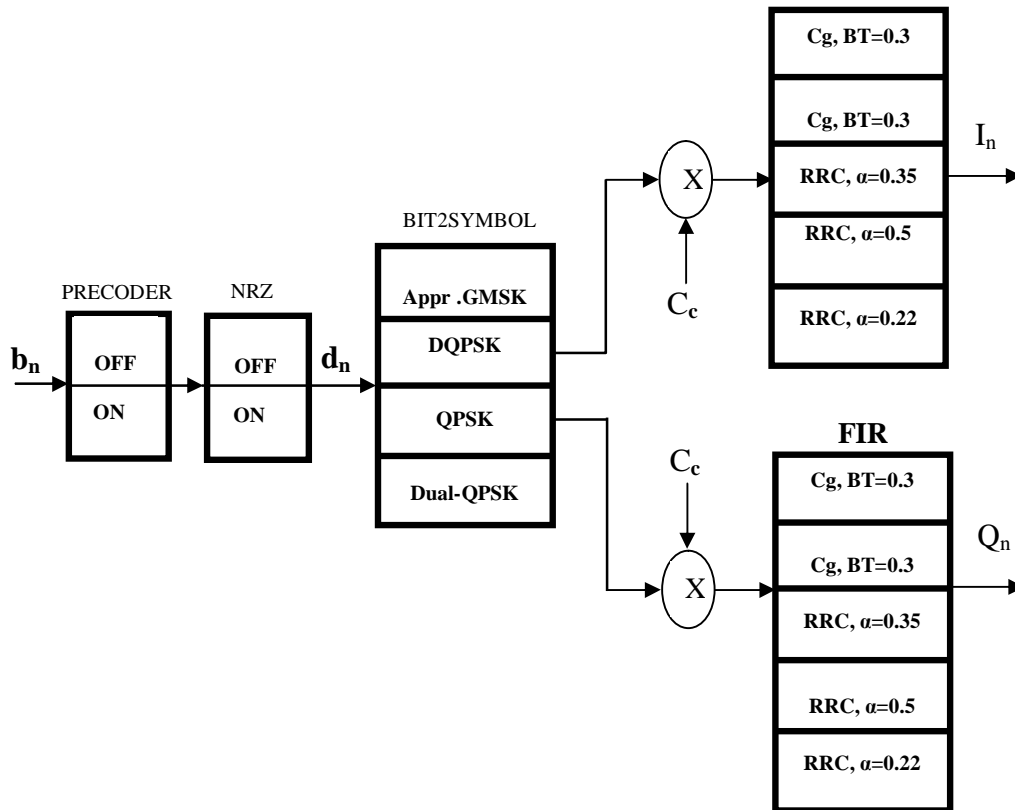


Figure. II.12. Fonction de modulation paramétrable [7].

Jean-Philippe Delahaye a proposé sur la base de cette technique une chaîne de transmission multistandards pour le GSM, UTRA/FDD et 802.11g [55]. Il a étudié chaque chaîne de traitement des 3 standards, puis il a remarqué que ces chaînes présentent un certain nombre de fonctions similaires appelées fonctions multi-standards, et fonctions spécifiques à chaque standard (figure II.13). Enfin, il a proposé d'ajouter les parties spécifiques à la partie fonctions multi-standards afin d'obtenir une chaîne multi-standards unifiée [55] (figure II.14).

L'approche « fonction commune » présente l'avantage d'un temps de changement de modes négligeable, et ainsi par un nombre limité de paramètres, on peut décrire plusieurs modes de fonctionnement.

Cependant, cette approche souffre d'un problème d'évolution et d'un problème d'efficacité énergétique. En effet, tous les exemples présentés ci-dessus, nous permettent de conclure que cette technique montre une forte dépendance des normes choisies, avec un faible partage de ressources. La technique « fonction commune » (FC) consiste en

l'agrégation des composants nécessaires dans une seule fonction, ce qui signifie que l'ajout d'une nouvelle norme implique l'ajout des composants distincts de chaque fonction dans les FC associées. Donc cette structure ne peut pas s'adapter facilement aux nouveaux besoins des standards [62][64]. Selon L. Alaus et al [67] cette technique est considérée comme une technique fixe.

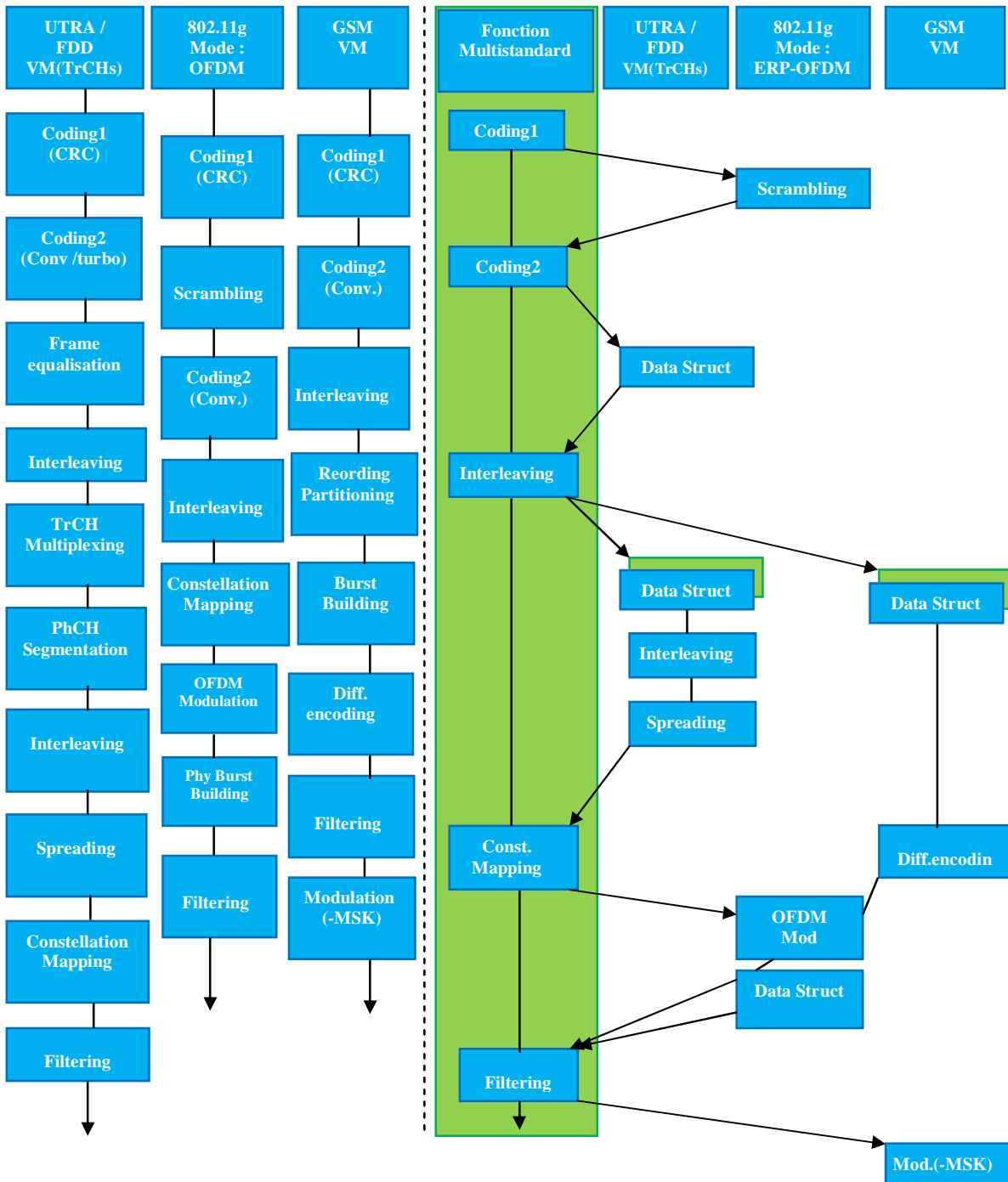


Figure. II.13. Vers une chaîne de transmission multi-standard unifiée [55].

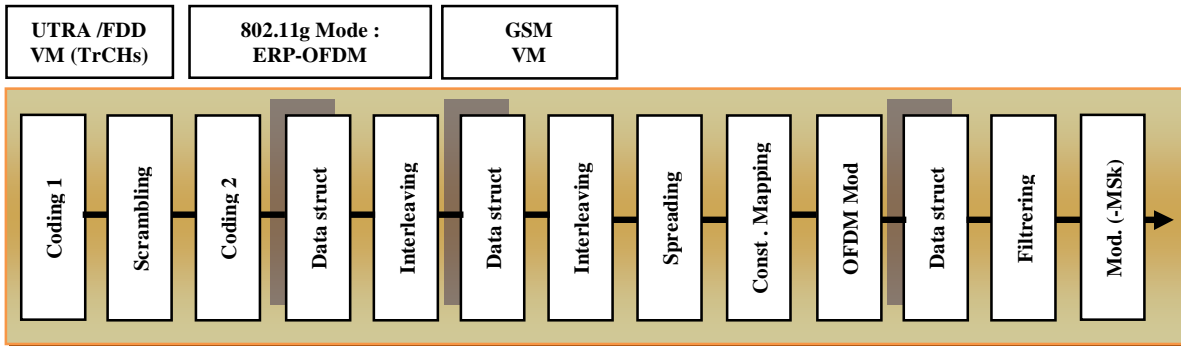


Figure. II.14. Chaîne de transmission multi-standard unifiée [55].

De plus, au moment de l'exécution, tous les modules de traitement quelque soit leur état actif (les modules requis par les paramètres sélectionnés) ou inactif (les modules non requis par les paramètres sélectionnés) consomment de l'énergie, ceci montre alors la perte de l'efficacité énergétique. Bien qu'il existe des mécanismes pour couper l'alimentation sur des fonctions entières, il est plus compliqué de couper l'alimentation sur des sous-blocs d'une même fonction [18]. Le coût en surface, qui peut aussi être significatif [18] [64], s'ajoute aux deux problèmes rencontrés précédemment. Les limites de la technique « fonction commune » ouvrent la voie à l'émergence d'une autre approche définie comme indépendante de la norme qui les utilise.

7.2. Opérateurs communs (OC)

La technique de l'opérateur commun est une deuxième approche de la paramétrisation située à un niveau de granularité intermédiaire entre fonction commune et opérateur arithmétique classique. Ce niveau de granularité est considéré comme idéal, car il fournit le meilleur compromis entre performances et complexité [62].

Cette technique consiste à faire une étude bibliographique sur les différentes architectures de transmission/réception (codage de canal, décodage, modulation, etc...) pour chaque standard afin d'identifier les points de similarité entre les différentes fonctions et architectures étudiées [2][4][66] [68]. Les points de similarité identifiés nous permettent ensuite de proposer des éléments communs (opérateurs ou structures de calcul) de granularité intermédiaire pour les appliquer à un grand nombre de fonctions.

Les opérateurs considérés ont pour objectif d'être communs à différents standards et aussi à différentes fonctions d'un même standard. Ils sont capables de passer d'un mode de fonctionnement à un autre c'est-à-dire reconfigurable par simple téléchargement de paramètres. Ces architectures génériques sont paramétrables en cours d'exécution et peuvent aussi être implantées sur n'importe quelle cible, logicielle ou matérielle [18] [68]. Les opérateurs peuvent être adaptables et évolutifs à n'importe quel standard, si les opérateurs communs sont définis comme indépendants des fonctions et par conséquent des normes qui les utilisent. En d'autres termes, un OC peut être défini par un opérateur, qui peut être réutilisé par chaque fonction ou bien le traitement exige la fonctionnalité de

l'opérateur, indépendamment du contexte d'application. Dans ce cas, la réutilisation des ces opérateurs génériques devient très optimale. En essence, un CO est utilisé pour effectuer des opérations sans savoir leur application [62]. Selon L. Alaus et al [67] la technique de l'opérateur commun est considérée comme une technique ouverte (open technique). En plus de l'aspect évolutif, cette technique aspire à être un design optimisé permettant d'obtenir un gain en complexité vis-à-vis de la méthode Velcro Classique [2] [4] [69].

Palicot et al [68] introduisent cette notion d'opérateur commun en radio logicielle par l'exemple de l'opérateur FFT. Ils se sont basés sur l'approche pragmatique, ils constatent que plusieurs traitements intéressants dans la chaîne de transmission/réception tels que la (dé)modulation OFDM, l'égalisation, le filtrage, etc., peuvent être effectués avec l'opérateur FFT. Cette étude présentée peut être considérée comme pionnière sur le sujet des opérateurs communs [68]. Dans la littérature [8][70], les auteurs développent quatre opérateurs communs à base des registres LSFR: (1) Reconfigurable Fibonacci LFSR (RF-LFSR), (2) Reconfigurable Galois LFSR (RG-LFSR), (3) Reconfigurable LFSR (R-LFSR) et (4) Extended Reconfigurable LFSR (ER-LFSR) qui peuvent être utilisés pour plusieurs fonctions tel que le codage canal et filtrage. A.Al Ghouwayel [69] propose une paramétrisation de l'opérateur FFT dans le champ de Galois pour la réalisation d'un encodeur Reed-Salomon. Cet encodeur est réalisé dans le domaine fréquentiel pour bénéficier de l'architecture matérielle performante du papillon FFT. L'opérateur proposé par A.Al Ghouwayel [69] est appelé double mode FFT (DMFFT), car ce dernier est capable de fonctionner sur deux domaines différents (champ complexe (C) et le champ de Galois (GF)). Finalement, M. Naoues et al. [3] [71] ont proposé des cellules de traitement matériel reconfigurables pour les algorithmes FFT et Viterbi, capables de prendre en compte leurs différences fonctionnelles et de s'y adapter à l'utilisation.

La technique des Opérateurs Communs est une approche ambitieuse dans le design de terminaux multistandards mais nécessite une gestion complexe (scheduling). Ce point faible de la méthode affecte non seulement la capacité du terminal à gérer les opérateurs mais aussi la complexité résultante de leurs implémentations en introduisant le besoin des structures de contrôle [72].

8. Notre solution préconisée dans le travail de thèse

Si on fait une analyse des plateformes de la radio reconfigurable et des solutions existantes pour avoir la flexibilité, la performance, l'efficacité énergétique et l'efficacité en surface attendue par la radio logicielle (SDR), nous pouvons tirer essentiellement la remarque suivante: la plate-forme hybride offre un bon compromis en termes de flexibilité et de complexité, mais la configuration d'une telle plate-forme revient à traiter les problèmes de synchronisation et de répartition des tâches entre les différentes

composantes. De plus, la condition d'avoir un environnement unique de travail se pose également. Pourtant, on peut remarquer que la technologie FPGA laisse la place à une solution intermédiaire, où on allie flexibilité de la programmation et puissance de calcul des architectures spécialisées. Sachant que, le développement des FPGA de haute performance et de capacités de calculs élevées est en pleine évolution offrant ainsi la possibilité de concevoir des System On Programmable Chip (SoPC) intégrant des processeurs embarqués, de mémoire de large capacité, des accélérateurs dédiés, et les différents types de flexibilité logicielle et matérielle.

En effet, selon M. Cummings [73] le compromis performance/flexibilité apporté par les FPGA en fait un bon candidat pour les applications radio logicielle par rapport aux ASIC et DSP. Aussi Selon Laurent Alhaus [5] les FPGA sont en fait un bon candidat pour la réalisation d'une plateforme hybride pour la Radio Logicielle, puisque les FPGA ressemblent de plus en plus à des systèmes sur puce, et présentent l'ensemble des niveaux de granularité: **(1) petit grain:** des systèmes de LUTs, **(2) grain moyen:** Multiplieurs à X-bit, additionneurs, et un accumulateur à K-bit, et **(3) gros grain:** DSPs, processeurs PowerPC. C'est pour cette raison que nous nous sommes particulièrement intéressés à ce type de circuits et à leur mise en œuvre dans les applications « radio logicielle ».

De plus, la paramétrisation et la technique d'opérateurs communs sont des techniques clés dans la radio logicielle [68][69]. Selon Malek NAOUES [18] la technique d'opérateurs communs se situe parmi les techniques de reconfiguration actuelles et vise une implémentation multistandards plus efficace. J.P. Delahaye [55], a fait dans sa thèse une comparaison de différentes approches de réalisation de traitements multi-standards (tableau II.2.).

Tableau II.2. Comparatif des approches de conception des multi-traitements [55].

	Fonction Fixe (Velcro)	Fonction reconfigurable	Paramétrisation Operateur commun	Paramétrisation Fonction commune
Performance	+++	++ (+)	+	++ (+)
Surface	--	++	++	-
Consommation Energétique	--	++	++	-
Temps de changement de contexte	+++	-	+	+++
Potentiel de réutilisabilité	--	++	+++	+
Complexité de conception	+++	++	--	+

À partir de cette comparaison, on peut remarquer que les deux approches: la reconfiguration et la Paramétrisation avec la technique d'opérateurs communs, présentent le meilleurs choix pour la radio logicielle. Cependant, le temps de changement dans l'approche reconfigurable représente son grand point faible. En effet, le temps de changement de modes doit être négligeable dans la radio logicielle, puisque les couches physiques travaillent en temps réel. Par contre, la technique de paramétrisation avec l'approche opérateur commun apparaît comme une technique très efficace pour le SDR et constitue elle même une méthodologie de conception de la Plateforme Hybride (hétérogène). De plus, la Paramétrisation avec la technique d'opérateurs communs peut constituer une méthode de reconfiguration en elle-même ou une technique complémentaire aux méthodes existantes telle que la Reconfiguration Partielle d'un FPGA.

Prenons l'exemple de [5] qui suppose qu'une technique de paramétrisation définisse une entité commune (COM1) entre deux standards S1 et S2. Supposons toujours que le paramètre α appliqué à COM1 permet d'exécuter une opération de S1 et que le paramètre β appliqué à COM1 permet d'exécuter S2. Alors, passer de S1 à S2 ne nécessitera plus de reconfigurer le terminal, de charger un code ou d'envoyer un bitstream mais juste de fournir en plus des données le paramètre α ou β selon le cas. Maintenant, supposons l'existence d'une deuxième entité commune COM2, suivant le même principe. En combinant Paramétrisation et reconfiguration partielle, nous pouvons sur la même sous partie du FPGA, faciliter la reconfiguration partielle en autorisant à reconfigurer seulement deux fois (soit COM1, soit COM2) au lieu de passer par quatre reconfigurations partielles.

Cette technique vise à ajouter aussi de la flexibilité pour les composants qui ne sont pas flexibles tel que les ASICs ou encore d'améliorer plus cette flexibilité pour les FPGA. Ceci nous permet d'utiliser des composants matériels de haute performance présentant une grande efficacité énergétique. Cette technique est surtout performante dans un FPGA pouvant lier la flexibilité logicielle et matérielle aussi bien à la conception que durant l'utilisation. C'est pour cette raison que l'approche opérateur commun a été privilégiée dans notre travail.

9. Conclusion

Dans ce chapitre nous avons présenté le besoin des terminaux reconfigurables pour la radio logicielle restreinte et la radio intelligente. Nous avons également donné le détail des différents éléments constitutifs d'une radio logicielle restreinte, en commençant par l'antenne agile jusqu'à, la partie de bande de base. Cette description détaillée des différents éléments de SDR, est renforcée par des exemples de quelques travaux de recherches récentes dans ce domaine.

En termes de reconfiguration du terminal multistandard, nous avons remarqué que celui-ci ne peut se réaliser exclusivement sur des cibles logicielles mais demande l'utilisation des cibles matérielles. La plateforme hétérogène est donc une solution prometteuse. Cependant il présente ainsi quelques interrogations à éclaircir.

Dans ce contexte, la technique d'opérateur commun définit des éléments qui peuvent être "reconfigurés" en temps réel en ne modifiant que des simples paramètres. Cette technique constitue elle-même une méthode de reconfiguration et une méthodologie de conception de la plateforme hybride. En plus si elle est combinée avec la technique de reconfiguration partielle, nous pouvons sur la même sous partie du FPGA faciliter ce type de reconfiguration.

Dans le chapitre qui suit, nous proposons un opérateur générique et versatile pour trois fonctions : l'algorithme CORDIC, FFT-SDF et l'unité de désétalement. Il peut être considéré comme un opérateur générique (commun) utilisé dans un grand nombre d'algorithmes classiques de traitement du signal. En plus dans le chapitre 4 nous proposons des architectures universelles pour la modulation et la démodulation.

Chapitre III

Architecture numérique versatile pour un terminal mobile

1. Introduction

Le récepteur Rake, le processeur FFT et l'algorithme Cordic sont devenus des éléments très importants dans de nombreux systèmes de transmission actuels. Dans les systèmes de radio-communication CDMA (Code Division Multiple Access), la propagation est caractérisée par des atténuations et de multiples réflexions du signal. Ces phénomènes sont dus aux obstacles naturels, tels que les immeubles et les montagnes qui provoquent une propagation caractérisée par des trajets multiples. Dans un système CDMA, cette propriété est considérée comme un avantage, car une forme de diversité appelée "la diversité multi-path" peut être exploitée. L'objectif est de détecter le plus grand nombre de chemins à l'aide d'un récepteur spécifique nommé récepteur RAKE pour les combiner [4]. Par conséquent, le signal final est renforcé ce qui conduit à une amélioration du rendement. La transformée de Fourier rapide (FFT) et son inverse (IFFT) sont des algorithmes très importants dans les techniques de modulation les plus prometteuses tels que l'OFDM (Orthogonal Frequency Division Multiplexing). Actuellement, plusieurs standards de télécommunication sont basés sur l'OFDM à cause de plusieurs avantages. En effet la modulation OFDM peut éliminer le problème de l'interférence entre symboles (IES) dans le cas d'un canal à trajets multiples sans l'utilisation d'une égalisation de grande complexité.

Le processeur Cordic est ainsi un algorithme nécessaire pour calculer les valeurs des fonctions élémentaires (tels que les fonctions trigonométriques et hyperboliques) avec la précision désirée, en utilisant seulement des additionneurs, soustracteurs, registres à décalage, et des constantes stockées.

Dans ce chapitre, nous proposons une architecture générique et universelle capable de prendre en charge les trois fonctions: l'unité de désétalement pour le récepteur Rake, le Processeur Élémentaire (PE) pour l'algorithme FFT-SDF (Fast Fourier Transform Single path Delay Feedback) et la cellule de l'algorithme Cordic. À cette fin, nous avons identifié les similarités entre les trois architectures couramment utilisées, pour proposer un opérateur générique basé sur une structure RTL pipeline. Nous discutons à la fin de ce chapitre les comparaisons entre notre architecture proposée et d'autres cellules proposées dans la littérature.

2. Récepteur Rake et l'unité de dés-étalement

Le récepteur Rake correspondant à un détecteur de CDMA développé pour les trajets multiples, est en d'autres termes une radio récepteur conçue pour minimiser les effets de l'évanouissement du signal afin de contrer les effets des canaux à trajets multiples. Le récepteur Rake effectue l'opération de désétalement et peut également être considéré comme une opération de corrélation ou une opération de filtrage adapté [4]. Par conséquent, un récepteur RAKE est composé de corrélateurs (doigts), chacun d'eux recevant un trajet multiple. Après le désétalement par corrélation avec une copie locale et retardée de manière appropriée du code d'étalement de l'émetteur, les signaux sont combinés afin d'améliorer la qualité du signal combiné global et les performances. La composition matérielle de la partie en bande de base d'un système WCDMA est fournie en annexe A.

De nombreuses implémentations de récepteur Rake ont été publiées au cours de ces dernières années. L'une des premières implémentations est le récepteur RAKE conventionnel [74]. Nous citons aussi le récepteur FlexRake [75] [76] et le récepteur Time Multiplexed Parallel RAKE ou «RAKE parallèle à multiplexage temporel» (TmpRAKE) [77]. Une autre architecture de récepteur RAKE baptisée CodeRAKE, a été proposée [78] [79] [80]. Cette architecture est partitionnée en fonction du nombre d'utilisateurs et du nombre de codes par utilisateur. En effet, avec cette approche de partition, nous parvenons à réduire les ressources matérielles nécessaires à son établissement, ce qui améliore l'efficacité du système. C'est pourquoi ce récepteur offre un bon compromis entre le débit et la surface par rapport à d'autres architectures. En outre, des débits plus importants peuvent être obtenus en utilisant une architecture parallèle. Ce parallélisme permet l'augmentation des cadences, en augmentant la surface, avec un rapport des débits / surfaces restant légèrement supérieur par rapport à d'autres architectures [79].

L'unité de désétalement est un élément très important dans le récepteur RAKE [75] [78] [79]. En effet, elle effectue la multiplication complexe entre les échantillons I/Q ($D_{in_i} + jD_{in_q}$) et les codes générés par le générateur de code conformément à l'équation (1).

$$D_{out_i} + jD_{out_q} = (D_{in_i} + jD_{in_q}) \cdot (C_{s_i} + jC_{s_q}) \cdot C_0 \quad (1)$$

Les codes générés par le générateur de code sont construits à partir des codes complexes de brouillage ($C_{s_i} + jC_{s_q}$) et de la valeur réelle du code OVVSF ' C_0 ' (voir Annexe A). La multiplication des codes de brouillage avec les bits de code OVVSF est donnée par l'équation suivante (2):

$$R_{c_i} + jR_{c_q} = (C_{s_i} + jC_{s_q}) \cdot C_0 = (C_{s_i} \cdot C_0) + j(C_{s_q} \cdot C_0) \quad (2)$$

En effet, les valeurs des deux parties (réelle et imaginaire) du code de brouillage sont égales à ± 1 , ainsi que la valeur de la partie réelle du code OVSF (où $+1$ et -1 illustrent respectivement les valeurs logiques 0 et 1). Cela signifie que R_{c_i} et R_{c_q} peuvent être soit $+1$ ou -1 , et le résultat de la multiplication des R_{c_i} et R_{c_q} avec l'échantillon d'entrée de données $D_{in_i} + jD_{in_q}$ changera le signe du signal d'entrée ($D_{in_i} + jD_{in_q}$) ou le laissera inchangé.

Ainsi, D_{out_i} et D_{out_q} peuvent prendre l'une des valeurs suivantes: $(+D_{in_i} + D_{in_q})$, $(+D_{in_i} - D_{in_q})$, $(-D_{in_i} - D_{in_q})$ et $(-D_{in_i} + D_{in_q})$, en fonction des valeurs des codes d'étalement et de brouillage tel que spécifié dans le tableau III.1.

Tableau III.1. Opérations entre les échantillons d'entrée pour différentes valeurs de code.

R_{c_i}	R_{c_q}	D_{out_i}	D_{out_q}
0	0	$+D_{in_i} - D_{in_q}$	$+D_{in_i} + D_{in_q}$
0	1	$+D_{in_i} + D_{in_q}$	$-D_{in_i} + D_{in_q}$
1	0	$-D_{in_i} - D_{in_q}$	$+D_{in_i} - D_{in_q}$
1	1	$-D_{in_i} + D_{in_q}$	$-D_{in_i} - D_{in_q}$

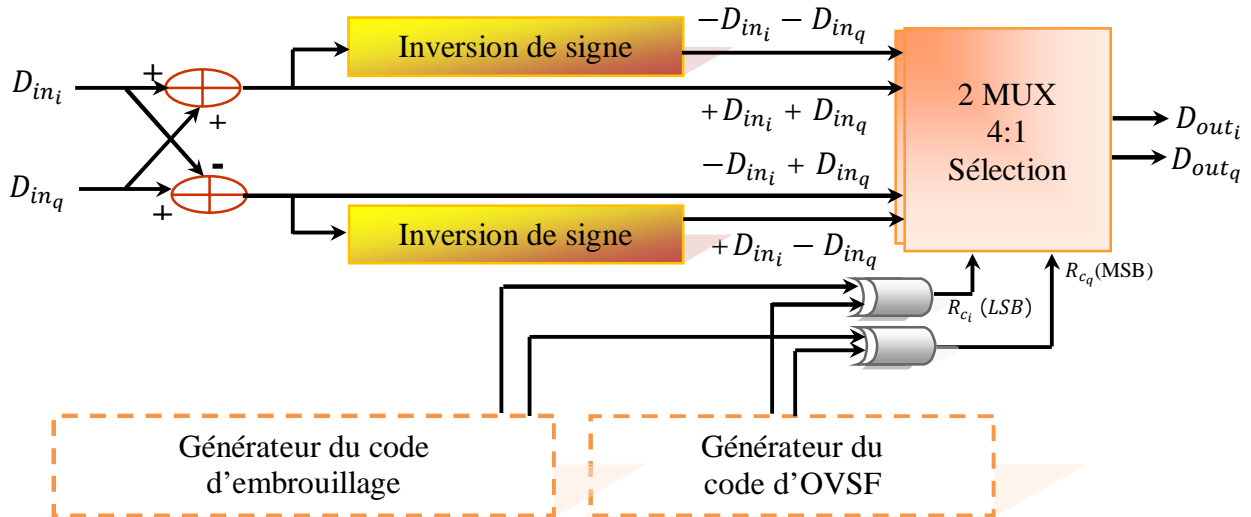


Figure. III.1. Conception de l'unité de désétalement.

En conséquence, sur la base des équations ci-dessus et le tableau III.1, l'opération de désétalement peut être effectuée à l'aide des éléments logiques suivants: deux multiplexeurs 4:1 (un multiplexeur pour chaque composant I ou Q), des portes XOR, des additionneurs/soustracteurs et des inverseurs de signe (l'inversion du signe est implémentée par un soustracteur et un "élément zéro») [81]. La figure III.1 représente la

conception matérielle de l'unité de désétalement. Cette unité au début calcule les parties réelle et imaginaire de l'échantillon d'entrée; puis par l'utilisation d'un additionneur, soustracteur et deux inverseurs de signe, on ajoute ou on soustrait les ensembles des résultats du calcul; enfin, les deux multiplexeurs sélectionnent les sorties par l'utilisation des codes d'embrouillage et le code OVSF. Après l'étape de désétalement, les données sont accumulées sur une période de symbole défini par T_s . En d'autres termes, cela signifie que la quantité totale de données correspondant à SF (facteur d'étalement) sera accumulée pour produire les symboles à la sortie du corrélateur.

2.1. Proposition d'une architecture pipeline pour les deux multiplexeurs 4:1

Les principaux composants de l'unité de désétalement sont les deux multiplexeurs 4:1 (voir figure III.1.), qui sont utilisés pour la sélection de données. Leur vitesse de fonctionnement avec également les vitesses de fonctionnement des additionneurs/soustracteurs ont un effet important dans l'unité de désétalement [4]. C'est pour cette raison que nous avons proposé une nouvelle architecture pipeline pour le multiplexeur 4:1, celle ci peut atteindre une grande fréquence de fonctionnement. La figure III.2 présente notre architecture RTL pour les deux multiplexeurs 4:1 pipeline utilisée dans l'unité de désétalement [4]. L'architecture est basée sur trois multiplexeurs 2:1 et des registres (REG dans la figure III.2.) [4].

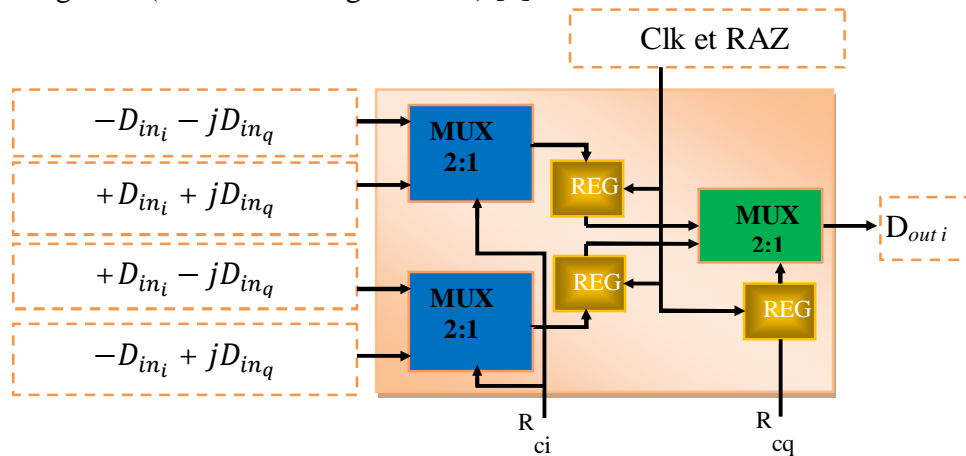


Figure. III.2. L'architecture RTL pipeline d'un multiplexeur (MUX 4:1) pour l'unité de désétalement.

Pour respecter strictement le tableau III.1 et fournir la sortie D_{out_i} , le premier multiplexeur pipeline 4:1 proposé (Voir figure III.2.) effectue les opérations de l'unité de désétalement suivantes:

Si $R_{c_i} = "0"$ les deux premiers MUX 2:1 (de couleur bleu dans la figure III.2) donnent respectivement $(+D_{in_i} + D_{in_q})$ et $(+D_{in_i} - D_{in_q})$ puis

Si $R_{c_q} = "0"$ le dernier MUX 2:1 (de couleur vert) donne $(+D_{in_i} - D_{in_q})$;

Sinon ($R_{c_q} = "1"$) le dernier MUX 2:1 (de couleur vert) donne $(+D_{in_i} + D_{in_q})$;

Sinon $R_{c_i} = "1"$ les deux premiers MUX 2:1 (de couleur bleu) donnent respectivement $(-D_{in_i} - D_{in_q})$ et $(-D_{in_i} + D_{in_q})$ puis
 Si $R_{c_q} = "0"$ le dernier MUX 2:1 (de couleur vert) donne $(-D_{in_i} - D_{in_q})$
 Sinon ($R_{c_q} = "1"$) le dernier MUX 2:1 (de couleur vert) donne $(-D_{in_i} + D_{in_q})$,

De même, pour fournir la sortie D_{out_q} , on a utilisé un deuxième multiplexeur pipeline 4:1 proposé (comme celle de la figure III.2) effectue les opérations de l'unité de désétalement suivantes:

Si $R_{c_i} = "0"$ les deux premiers MUX 2:1 (de couleur bleu) donnent respectivement $(+D_{in_i} + D_{in_q})$ et $(-D_{in_i} + D_{in_q})$ puis
 Si $R_{c_q} = "0"$ le dernier MUX 2:1 (de couleur vert) donne $(+D_{in_i} + D_{in_q})$
 Sinon ($R_{c_q} = "1"$) le dernier MUX 2:1 (de couleur vert) donne $(-D_{in_i} + D_{in_q})$;
 Sinon $R_{c_i} = "1"$ les deux premiers MUX 2:1 (de couleur bleu) donnent respectivement $(-D_{in_i} - D_{in_q})$ et $(+D_{in_i} - D_{in_q})$ puis
 Si $R_{c_q} = "0"$ le dernier MUX 2:1 (de couleur vert) donne $(+D_{in_i} - D_{in_q})$
 Sinon ($R_{c_q} = "1"$) le dernier MUX 2:1 (de couleur vert) donne $(-D_{in_i} - D_{in_q})$.

La principale originalité de notre architecture pipeline proposée pour les deux multiplexeurs 4:1, est non seulement d'atteindre une grande fréquence de fonctionnement, mais aussi de renforcer les points de similarité entre l'unité de désétalement, le processeur élémentaire (PE) dans le FFT-SDF et la cellule élémentaire dans l'algorithme CORDIC (détaillées aux prochaines sous-sections) [4]. Ces points de similarité entre les différentes architectures étudiées nous permettent de définir précisément un cahier des charges concernant les caractéristiques principales de l'architecture générique à réaliser.

3. L'architecture de FFT-SDF

Le cœur des systèmes multi porteuse orthogonaux est l'opération de FFT/IFFT (cf. Annexe B), qui représente la partie de calcul la plus gourmande en ressource de calcul [82] [83]. Il existe principalement deux types d'architectures pour la réalisation de la FFT, les architectures de type pipeline et les architectures à base de mémoires. Les plus utilisées sont les architectures pipeline, car elles offrent un bon compromis complexité matérielle/taux de traitement de données pour les systèmes de communications à haut débit [84]. Les FFT à base des architectures pipeline peuvent atteindre un débit élevé avec un contrôle relativement simple [85] [86]. De plus, les architectures pipeline ont des structures régulières et nous permettent de faire l'implémentation d'une FFT de longue taille (size). Plusieurs architectures pipeline ont été développées, comme le Multi-path Delay Commutator (MDC) [87], Single path Delay Feedback (SDF) [87], [88], et Single path Delay Commutator (SDC). Compte tenu de ces travaux antérieurs, l'architecture pipeline FFT-SDF peut être facilement adaptée et paramétrée dans une conception

matérielle [87], [88], ce qui la rendra plus appropriée pour les systèmes OFDM. Au niveau de la complexité matérielle, elle utilise un seul multiplieur complexe après chaque Radix-2/4/8 à cause du chemin unique d'accès, ce qui va réduire la complexité matérielle.

De plus, elle utilise les registres de manières plus efficaces en stockant les sorties des processeurs élémentaires (PE) dans les registres à décalage de la chaîne à rétroaction [89]. Toutefois, pour les systèmes MIMO-OFDM, l'architecture nécessite l'utilisation de plusieurs coprocesseurs FFT-SDF ce qui augmente la complexité matérielle dans ce type de système. L'architecture MDC est la plus adaptée aux systèmes MIMO-OFDM parce qu'elle possède de multiples chemins d'entrées et de sorties, cependant l'inconvénient de la méthode MDC est que la taille de la mémoire par PE est plus élevée comparativement au SDF. Par conséquent, une architecture pipeline de type SDF permet d'atteindre 8-8192-points de la FFT en ajoutant plusieurs processeurs élémentaires (PE). La figure III.3 montre une architecture pipeline FFT de 1024 points basée sur le Radix-2/4/8.

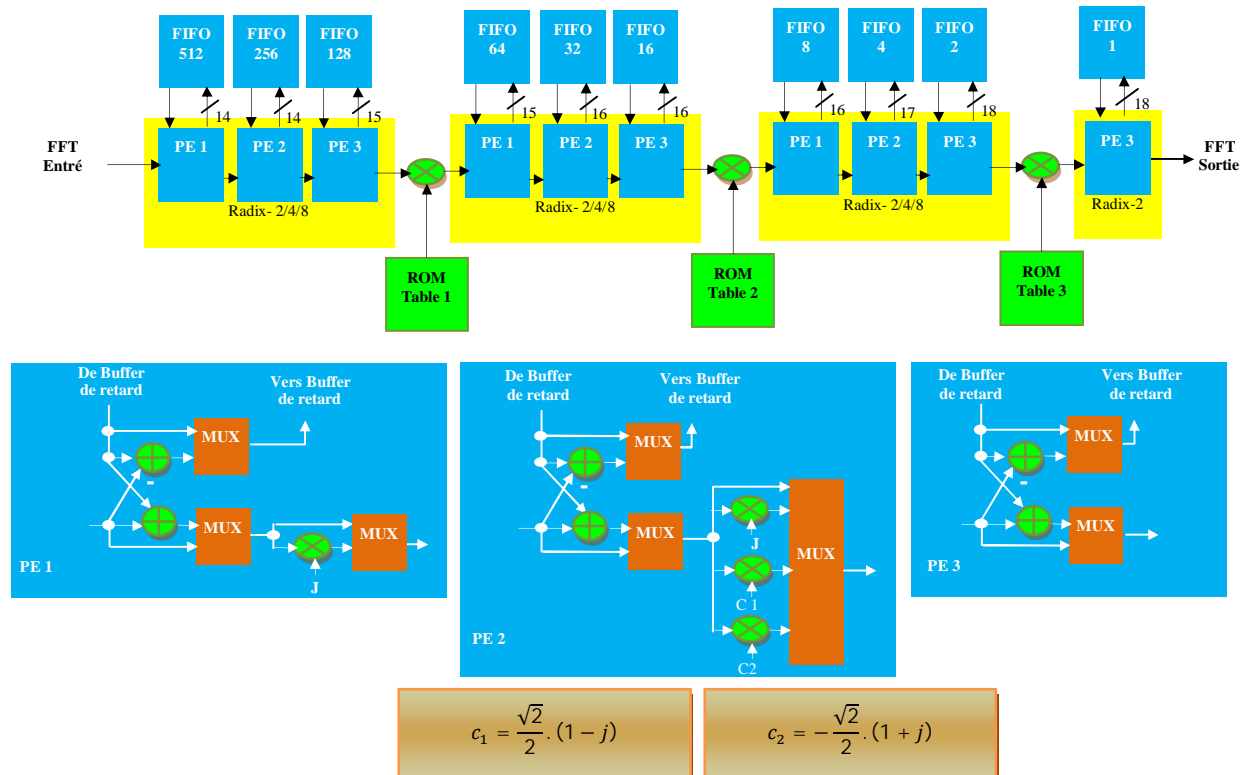


Figure. III.3. Une architecture pipeline FFT-SDF (N=1024 points).

3.1. Processeur Butterfly Radix-2/4/8.

Le Radix-2/4/8 signifie en français radical-2/4/8 ou base-2/4/8 dont la structure est très simple, et appelé également structure Butterfly (papillon) à cause de son schéma en forme de papillon. Dans l'algorithme Radix-2/4/8, l'unité de calcul de base est divisée en trois petits processeurs élémentaires (PE) qui sont désignés PE1, PE2, et PE3 respectivement.

L'algorithme de FFT radix-2/4/8 minimise le nombre des multiplications complexes avec l'extraction de $\pm j$ et $\sqrt{2}/2.(1 \pm j)$, sachant qu'une multiplication prend plus d'espace en silicium qu'une simple addition. En outre, l'algorithme de FFT Radix-2/4/8 est plus régulier et approprié pour les implémentations d'architecture pipeline VLSI. Figure III.3 détaille l'architecture FFT à 1024 points en utilisant un processeur à base de 2/4/8 Butterfly. Dans notre exemple, le module de FFT ne nécessite que trois multiplications complexes et des tableaux associés pour le facteur de rotation (twiddle factor). Chaque multiplicateur complexe est réalisé par trois multiplications réelles et cinq additions/soustractions réelles [90], et chaque table de « twiddle factor » stocke uniquement les valeurs de sinus/cosinus de 0 à $\pi/4$ (en utilisant la symétrie des fonctions sinus/cosinus). Pour mémoriser temporairement les données, il existe plusieurs Buffer de retard (Delay Buffer) de différentes longueurs dans l'architecture FFT. Les plus longs Buffers de retard sont implémentés avec des modules SRAM à port unique ou des mémoires simple accès (single-port SRAM modules), et les plus courts Buffers de retard sont implémentés par des bascules 'D' qui sont configurées comme des FIFO (first-in-first-out).

3.2. Addition et soustraction complexe

À partir des processeurs élémentaires Butterfly (PE_x) ($x = 1, 2, 3$) (voir Figure III.3), nous définissons la méthode de calcul des parties réelles et imaginaires en utilisant les équations suivantes (3) et (4):

$$(a + jb) + (c + jd) = (a + c) + j(b + d) \quad (3)$$

$$(a + jb) - (c + jd) = (a - c) + j(b - d) \quad (4)$$

Par conséquent, nous pouvons concevoir le Butterfly PE_x présenté dans la figure III.3 à partir de deux architectures: la première est pour le calcul de la partie réelle représentée par $(a + c)$ et $(a - c)$, la seconde est pour le calcul de la partie imaginaire représentée par $(b + d)$ et $(b - d)$, comme représentées dans la figure III.4.

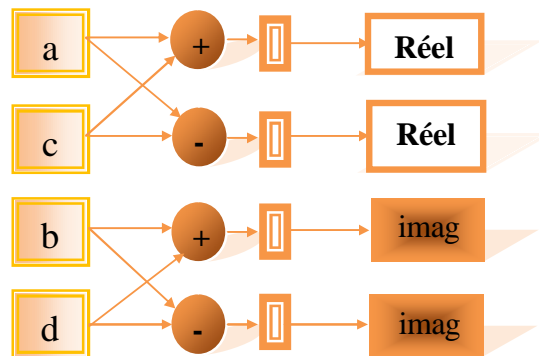


Figure. III.4. Addition et soustraction complexe.

3.3. Multiplication Complexe

Il existe trois types de multiplications dans l'architecture FFT-SDF à base de Radix-2/4/8: la multiplication par «j», la multiplication par un facteur de rotation constant et la multiplication par un facteur de rotation complexe.

a) **Multiplication par «j»:** La multiplication par «j» peut être réalisée sans aucun coût de matériel supplémentaire, on change simplement les parties réelles et imaginaires des entrées, et la partie réelle change de signe, comme représentée dans l'équation 5 et la figure III.5.

$$(a + bj) \times (j) = -b + aj \quad (5)$$

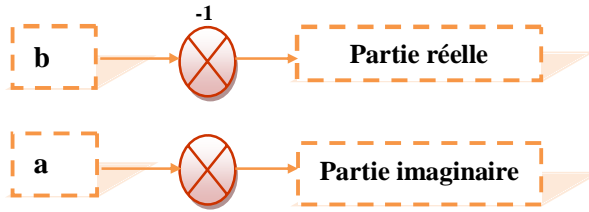


Figure. III.5. Schéma de la multiplication par j.

b) **La multiplication par un facteur de rotation constant:** Les multiplications complexes avec w_8^1 et w_8^3 (C_1 et C_2 dans la figure III.3.) peuvent être exprimées respectivement dans les équations suivantes:

$$(a + jb) * W_N^{N/8} = (a + jb) * \left(\frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}\right) = \frac{\sqrt{2}}{2} [(a + b) + j(b - a)] \quad (6)$$

$$(a + jb) * W_N^{3N/8} = (a + jb) * \left(-\frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}\right) = \frac{\sqrt{2}}{2} [(b - a) - j(b + a)] \quad (7)$$

Ainsi, on peut réaliser ces opérations de multiplication avec deux additions réelles et deux multiplications réelles comme le montre la figure III.6.

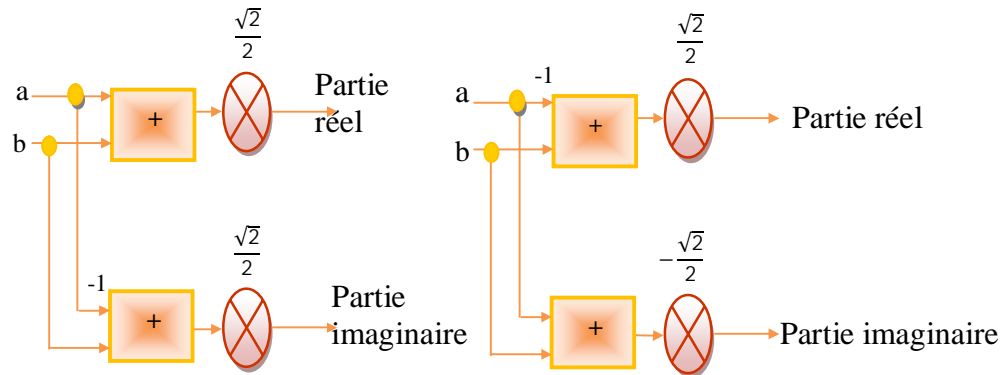


Figure. III.6. Multiplication complexes: (a) Multiplication par w_8^1 . (b) Multiplication par w_8^3 .

En outre, la multiplication réel par $\frac{\sqrt{2}}{2}$ peut également être réécrite comme $\frac{\sqrt{2}}{2} = 0,70710678 = 2^{-1} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-8} + 2^{-9}$, ce qui signifie que la multiplication

par $\frac{\sqrt{2}}{2}$ peut être remplacée par une opération de décalage et cinq additions réelles comme indiqué dans la figure III.7.

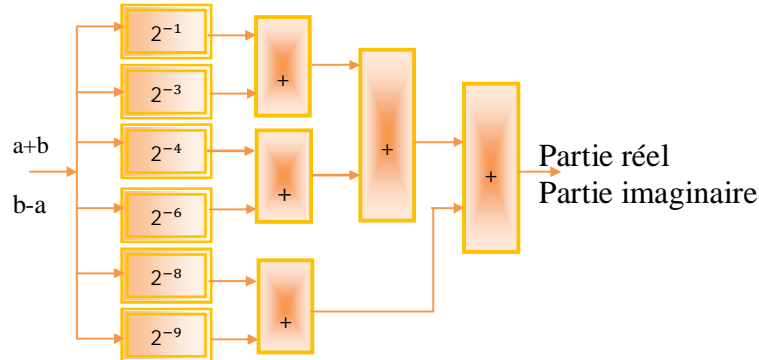


Figure. III.7. Implémentation matériel de la multiplication avec $\frac{\sqrt{2}}{2}$.

c) **La multiplication par le facteur de rotation complexe:** Un multiplieur complexe peut être réalisé par quatre multiplications réelles et deux additions réelles. Sa forme mathématique peut être exprimée comme suit:

$$(a + bj)(c + dj) = (ac - bd) + j(bc + ad) \quad (8)$$

Ce multiplieur complexe occupe une grande surface dans une implémentation matérielle (la consommation du multiplieur augmente avec l'augmentation de la représentation binaire des opérandes). Heureusement, ce multiplieur complexe peut être réalisé avec trois multiplications réelles et cinq additions réelles comme le montre la figure III.8 sur la base de l'équation 9:

$$\begin{aligned} (a + bj)(c + dj) &= (ac - bd) + j(bc + ad) \\ &= (ac - bc + bc - bd) + j(bc + bd - bd + ad) \\ &= \{(ac - bc) + (bc - bd)\} + j\{(ad + bd) + (bc - bd)\} \\ &= \{c(a - b) + b(c - d)\} + j\{d(a + b) + b(c - d)\} \end{aligned} \quad (9)$$

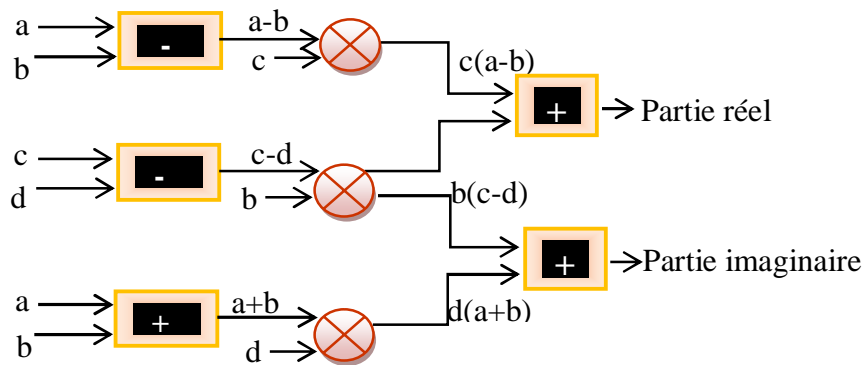


Figure. III.8. Multiplieur complexe avec trois multiplications réelles et cinq additions réelles.

3.4. Architecture du multiplicateur complexe proposée

Il est clair donc que l'architecture à trois multiplications, représentée dans la figure III.8 permet de réduire la surface et la consommation électrique du circuit par rapport à une architecture à 4 multiplications, mais diminue la fréquence maximale du fonctionnement du papillon. Pour réduire encore la surface et la consommation par rapport à l'architecture à 3 multiplications, nous avons proposé une architecture pipeline du multiplicateur complexe. La figure III.9 présente notre architecture RTL pipeline pour le multiplicateur complexe.

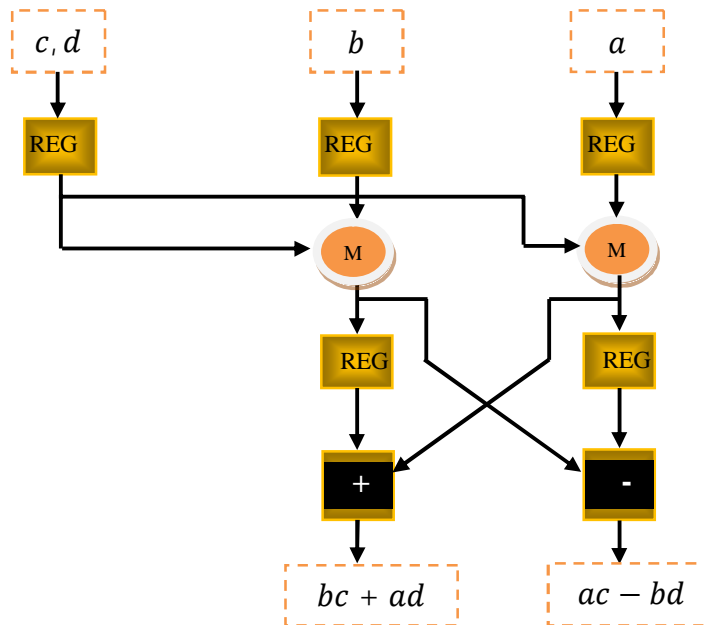


Figure. III.9. L'architecture RTL pipeline pour le multiplicateur complexe.

L'idée de cette architecture est inspirée d'une architecture pipelinée du papillon FFT Radix-2 proposée par la référence [91]. Cette architecture nécessite seulement 2 multiplicateurs réels, de plus, elle permet de réaliser l'opération de la multiplication en 2 cycles d'horloge selon l'équation 3.10.

$$\left. \begin{aligned}
 \text{Cycle\#1 : } A1 &= ac; B1 = bc \\
 \text{Cycle\#2 : } A2 &= ad; B2 = bd \\
 \text{Partie Réelle} &= ac - bd \\
 \text{Partie Imaginaire} &= bc + ad
 \end{aligned} \right\} \quad (10)$$

4. L'Algorithme CORDIC

L'algorithme CORDIC a été développé par Jack E. Volder dans [92], mais a bénéficié de nombreuses améliorations et extensions par la suite. L'algorithme Cordic utilise une transformation dynamique plutôt que des ROM statiques, ce qui rend cette technique très intéressante pour le calcul des fonctions trigonométriques et hyperboliques. Il existe deux

modes différents de calcul dans le CORDIC, le mode vecteur et le mode rotation. Le mode vecteur calcule le module et la phase à partir d'un vecteur de coordonnées x et y . Dans ce mode, l'algorithme cherche à annuler la partie imaginaire par rotations successives. Les variables x_0 , y_0 et θ_0 sont initialisées à x , y et 0 respectivement, et on effectue des rotations jusqu'à l'annulation de la partie imaginaire y . Le mode rotation qui nous intéresse, calcule le sinus et le cosinus d'un angle initial. L'algorithme cherche à annuler θ_0 (initialisé par l'argument de rotation souhaitée θ). L'angle de rotation est d'abord déterminé par le mode vecteur puis calculé par le mode rotation. Cette rotation peut être écrite comme suit [92]:

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (11)$$

En utilisant l'identité : $\cos\theta = 1/\sqrt{1 + \tan^2\theta}$ et en plaçant $\cos\theta$ en facteur, l'équation (11) peut être modifiée comme suit :

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \frac{1}{\sqrt{1+\tan^2\theta}} \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (12)$$

Dans la méthode CORDIC, la rotation par un angle θ est implémentée comme un processus itératif, composée de plusieurs micro-rotations dans laquelle le vecteur initial est mis en rotation par des angles de pas (*step*) prédéterminés α_i . Si on spécifie le sens de rotation ou " d_i ", la somme des pas (step) α_i se rapproche à l'angle θ comme suit:

$$\theta = \sum_{i=0}^{n-1} d_i \alpha_i, \quad d_i \in \{-1, 1\} \quad (13)$$

Une variable z_i auxiliaire est introduite pour contenir la somme partielle cumulée de pas d'angles. Cette variable est également utilisée pour déterminer le signe de la prochaine micro-rotation. Pour simplifier le calcul du produit matriciel donné par l'équation (13), les angles de pas α_i sont choisis tels que $\tan\alpha_i$ représente une série de puissances de 2: $\tan\alpha_i = 2^{-i}$, $i=0, 1, 2, \dots, n-1$.

Dans le mode rotation, l'algorithme CORDIC a trois entrées qui sont initialisées à la coordonnée du vecteur x_0 , y_0 et de l'angle de rotation $z_0 = \theta$. Ce mode est décrit par les équations suivantes [92]:

$$\begin{cases} x_{i+1} = x_i - y_i d_i 2^{-i} \\ y_{i+1} = y_i + x_i d_i 2^{-i} \\ z_{i+1} = z_i - d_i \arctan 2^{-i} \end{cases} \quad (14)$$

$$\text{Où } d_{i+1} = \begin{cases} +1 & \text{if } z_{i+1} < 0 \\ -1 & \text{if } z_{i+1} > 0 \end{cases} \text{ et } i = 0, 1, 2, \dots, n-1.$$

D'après les considérations précédentes, la figure III.10 donne la cellule de base d'un processeur CORDIC. Cette architecture RTL présente une complexité arithmétique faible

[4], étant seulement basée sur trois additionneurs/soustracteurs, deux registres à décalage (de couleur jaune dans la figure III.10) effectuant la division par 2 pour x et y, des registres pour mémoriser les valeurs des angles de rotation, alors que les valeurs de arctan 2^{-i} sont pré-calculées (stockées dans une mémoire).

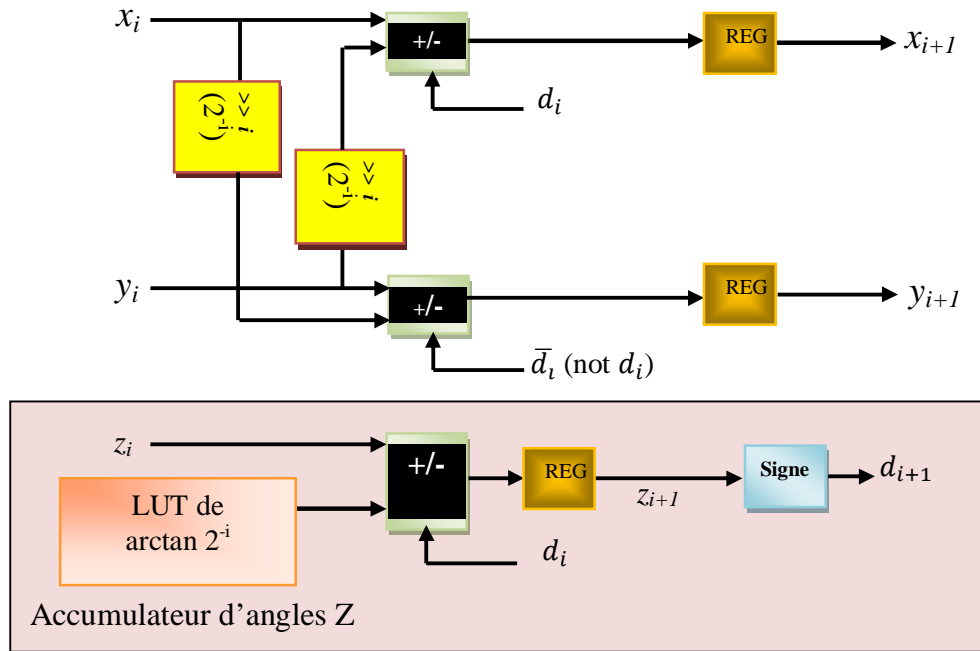


Figure. III.10. Cellule de base d'un processeur CORDIC.

Cette cellule CORDIC de base transmet à chaque cycle d'horloge les nouvelles valeurs aux additionneurs/soustracteurs et les registres à décalage. Les résultats sont ensuite stockés dans les registres [4].

Les sorties du x_n , y_n et z_n en mode de rotation, sont données par les expressions suivantes:

$$\begin{cases} x_n = K_n(x_0 \cos z_0 - y_0 \sin z_0) \\ y_n = K_n(y_0 \cos z_0 + x_0 \sin z_0) \\ z_n = 0 \end{cases} \quad (15)$$

Où x_n et y_n sont les coordonnées du vecteur de rotation (par l'angle θ) et $K_n = \prod_{i=0}^{n-1} \sqrt{1 + 2^{-2i}}$. La constante K_n , est le facteur d'échelle qui représente l'augmentation de l'amplitude du vecteur pendant le processus de rotation. Il existe deux méthodes principales pour l'implémentation matérielle de l'algorithme CORDIC. La première est l'architecture CORDIC itérative où chaque branche est constituée d'une combinaison d'addition/soustraction, unités de décalage et registres pour stocker les résultats. La seconde méthode est l'architecture CORDIC pipeline; au lieu d'utiliser les mêmes

ressources, on peut simplement faire la cascade de plusieurs architectures CORDIC itérative. Par conséquent, la sortie d'un étage constitue l'entrée de la suivante.

Afin de réduire la surface occupée dans la version pipeline, G.ARNOULD [93] propose une optimisation de cet algorithme par l'élimination totale de l'accumulateur d'angles Z (figure III.10). La suppression de l'accumulateur d'angle nécessite la connaissance, pour chaque étage de pipeline, du sens de rotation à faire pour un angle donné. Au lieu d'utiliser le signe de z_k pour déterminer le sens de rotation de l'ordre k, G.ARNOULD propose donc d'utiliser le numéro d'ordre k dans la représentation binaire de l'angle θ [93]. Par exemple, pour un angle dont la représentation binaire est la suivante $0_11_21_30_41_50_6$, la première rotation est contrôlée par le nombre 0, la deuxième est contrôlée par le nombre 1, et ainsi de suite, enfin la sixième et la dernière rotation est contrôlée par le nombre 0. L'architecture résultante est présentée sur la figure III.11.

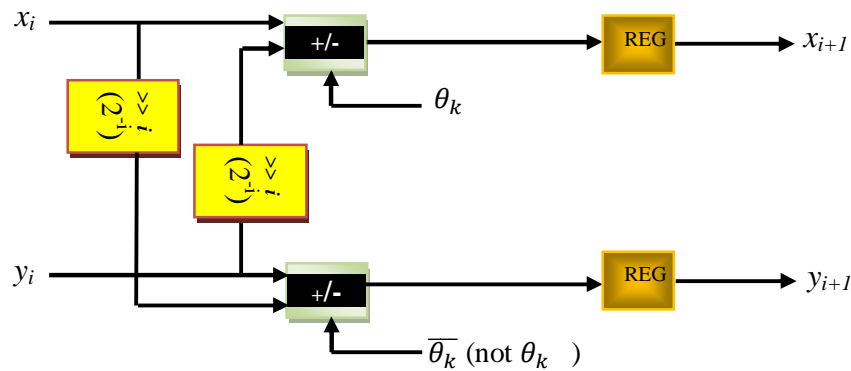


Figure. III.11. Cellule de base d'un processeur CORDIC avec la modification de [93].

Où θ_k dans la figure III.11 représente le numéro d'ordre k dans la représentation binaire de l'angle θ . Cette suppression de l'étage de détection de signe réduit la surface occupée, dans le FPGA, et ceci nous permet donc de paralléliser plusieurs étages CORDIC capables de générer en parallèle plusieurs échantillons de sinusoïde sur un seul FPGA.

5. Opérateur universel pour la FFT, l'unité de désétalement et l'algorithme Cordic

Dans cette section, nous étudions les similarités entre le processeur élémentaire PEx pour le FFT-SDF, l'unité de désétalement pour le récepteur Rake et la cellule élémentaire d'un processeur CORDIC. L'idée principale est de définir et de proposer une structure générique (opérateur commun) qui permet de basculer entre les trois fonctions par changement de paramètre, en suivant la technique de paramétrisation et l'approche d'opérateurs communs. Sachant que l'approche d'opérateurs communs (CO) suit deux étapes :

- (1) l'étape de recherche (Existing Search),

- (2) l'étape de la construction (Constructive Search).

L'étape de recherche consiste à identifier dans la littérature et les travaux existants, certains opérateurs qui peuvent mutualiser ou partager des ressources communes (tant au niveau algorithmique qu'architectural). La construction est la prochaine étape dans la conception du CO, qui suit l'étape « Existing Search », elle consiste donc à la construction de CO générique qui devra alors réaliser les traitements ciblés [62]. Par conséquent, le premier objectif de la technique de l'opérateur commun est de rechercher les points de similarité entre les différentes architectures étudiées pour définir précisément un cahier des charges concernant les caractéristiques principales de l'architecture générique à réaliser. Suivant les structures présentées précédemment de l'unité de désétalement, le Radix 2/4/8 pour le FFT-SDF, et la cellule Cordic, on remarque l'existence de trois principales similarités [4]:

- La première similarité qui est la principale, est que les trois structures ont la même forme « structure Butterfly ». Cette remarque est apparente à travers les figures (III.1, III.3 et III.4), ainsi les équations qui décrivent leurs fonctionnements (les équations (1), (2), (3), (4) et l'équation (14)). Cette similarité est bien illustrée dans la figure III.12 (pour la partie réelle seulement de FFT), où on peut voir qu'il existe dans les trois fonctions, une opération d'addition et de soustraction pour deux opérands (avec une légère modification pour le Cordic).
- La seconde similarité est la présence des multiplexeurs 2:1 entre l'unité de désétalement et le processeur élémentaire (PE_x) de FFT-SDF. Ces multiplexeurs peuvent être partagés entre les deux fonctions.
- La troisième similarité est la présence de registres entre l'unité de désétalement et la cellule Cordic.

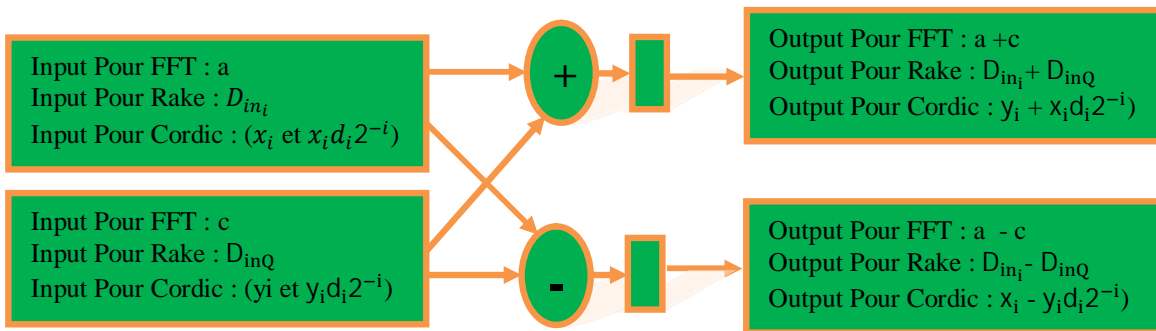


Figure. III.12. Structure Butterfly Commun.

Par contre les blocs qui ne peuvent pas se partager entre les trois architectures sont les deux registres à décalage effectuant la division par 2 pour x et y qui se trouvent seulement dans la cellule Cordic, et les deux inverseurs de signe qui se trouvent seulement dans l'unité de désétalement.

Après avoir identifié les points de similarité entre les trois architectures étudiées, nous essayons de proposer une architecture générique unique capable de prendre en charge les trois fonctions par les mêmes ressources. Notre architecture proposée se compose d'additionneurs, de soustracteurs, de multiplexeurs et de registres comme des blocs fonctionnels partagés entre les trois fonctionnalités [4]. La figure III.13 représente notre architecture universelle proposée pour l'unité de désétalement, le processeur élémentaire PE de FFT, et la cellule Cordic.

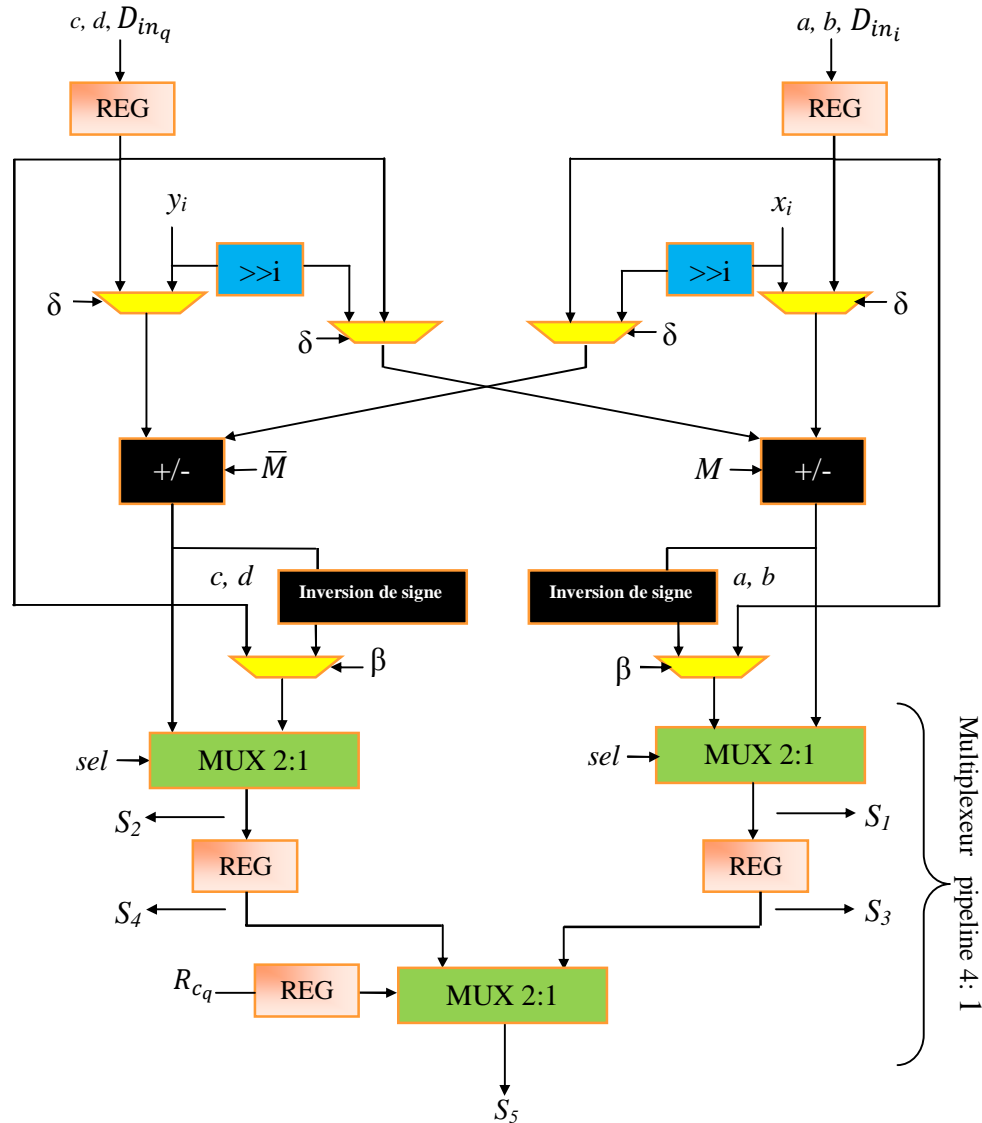


Figure. III.13. L'opérateur universel proposé pour l'unité de désétalement, processeur élémentaire PE de FFT, et la Cellule Cordic [4].

Elle est composée de deux blocs d'addition/soustraction, deux soustracteurs pour l'inversion de signe (inversion du signe dans la figure III.13), deux décaleurs (registres à décalage de couleur bleu), registres de données et des multiplexeurs 2:1. Il faut noter que les blocs additionneur/soustracteur sont capables de traiter différents types de données

(quantification) de la FFT, Cordic et de l'unité de désétalement [4]. En effet, il y a une différence dans le codage (le nombre de bits utilisés) des échantillons entre les différentes applications [18].

Dans cette architecture RTL proposée, β et δ sont des paramètres booléens (implémentés par un seul bit de sélection) qui permettent la configuration de la structure générique en utilisant des multiplexeurs (colorés en jaune dans la figure III.13) pour basculer entre le calcul de: PE-FFT, l'unité de désétalement et la cellule Cordic. Le signal de commande (SEL) est commun entre les trois fonctionnalités, et il peut avoir trois valeurs en fonction du mode choisi: R_{c_i} (signal de sélection pour le désétalement), F_{FFT} (signal de sélection pour la FFT), et δ (utilisée pour la sélection du calcul Cordic). M est un signal de commande permettant de fonctionner comme un additionneur ou un soustracteur. Enfin le signe (d_i) utilisé pour la cellule Cordic, est commandé par la représentation binaire de l'angle θ [93], ceci nous permet de supprimer l'accumulateur d'angle et donc de réduire l'occupation de surface de notre architecture générique.

La fonction de l'architecture proposée (voir figure III.13) peut être comprise, en considérant les équations ci dessous comme une synthèse de calculs obtenus à partir des équations (1), (2), (3), (4) et l'équation (14):

$$\begin{cases} \alpha = \left((a, b, D_{in_i}).\delta + (1 - \delta).x_i \right) + M. \left((c, d, D_{in_q}).\delta + (1 - \delta).y_i.2^{-i} \right) \\ \alpha' = \left((a, b, D_{in_i}).\delta + (1 - \delta).y_i \right) + \bar{M}. \left((c, d, D_{in_q}).\delta + (1 - \delta).x_i.2^{-i} \right) \end{cases} \quad (16)$$

$$\begin{cases} k = \beta.(a, b) + (1 - \beta).Inversion\ sign(\alpha) \\ k' = \beta.(c, d) + (1 - \beta).Inversion\ sign(\alpha') \end{cases} \quad (17)$$

$$\begin{cases} S_1 = sel.\alpha + (1 - sel).k \\ S_2 = sel.\alpha' + (1 - sel).k' \end{cases} \quad (18)$$

$$\begin{cases} S_3 = S_1.Z^{-1} \\ S_4 = S_2.Z^{-1} \end{cases} \quad (19)$$

où Z^{-1} représente le retard de registre.

$$S_5 = R_{c_q}.S_3 + (1 - R_{c_q}).S_4 \quad (20)$$

$$M = \delta + d_i \rightarrow \bar{M} = not(M) \quad (21)$$

$$sel = \left(R_{c_i}.(1 - \beta) + F_{FFT}.(\beta) \right) (\delta) + (1 - \delta) \quad (22)$$

L'architecture proposée permet de commuter entre les modes suivants:

- **Le mode FFT.** Ce mode est sélectionné par les paramètres de configuration ' β ' = 1 et ' δ ' = 1. Selon les équations développées précédemment (pour la partie réelle comme illustration), nous avons les valeurs des paramètres et les signaux de

configuration suivants: $M = \delta = 1 \rightarrow \bar{M} = 0$; $sel = F_{FFT}$ (signal de sélection pour la FFT); $\alpha = a + c$; $\alpha' = a - c$; $k = a$; $k' = c$; $S_1 = F_{FFT} \cdot \alpha + (1 - F_{FFT}) \cdot k$ (vers le reste du PEX) et $S_2 = F_{FFT} \cdot \alpha' + (1 - F_{FFT}) \cdot k'$ (vers le buffer de retard). où (a & c) représentent les données d'entrées, et S_1 et S_2 sont les sorties de PEX dans le mode FFT. Nous pouvons obtenir le résultat de la partie imaginaire (b & d) par le même mode. La figure III.14 (a) illustre l'implémentation du mode FFT et les ressources utilisées (de couleur bleu). Le mode FFT nécessite deux opérations d'addition/soustraction. Dans ce mode également, l'architecture peut être: un élément de traitement "1" (PE1 sur la figure III.3) si on ajoute une multiplication par « j », ou un élément de traitement "2" (PE2 sur la figure III.3) si on ajoute une multiplication par « j » et une multiplication par le facteur de rotation constant $(\frac{\sqrt{2}}{2} \cdot (1 \pm j))$, et enfin, un élément de traitement "3" (PE3 sur la figure III.3).

- **Le mode désétalement :** ce mode est sélectionné par les paramètres de configuration de valeurs ' β ' = 0 et ' δ ' = 1. Les équations développées précédemment donnent les valeurs des paramètres et les signaux de configuration suivants: $M = \delta = 1 \rightarrow \bar{M} = 0$; $sel = R_{c_i}$ (signal de sélection pour le désétalement, voir le tableau III.1); $\alpha = +D_{in_i} + D_{in_q}$; $\alpha' = +D_{in_i} - D_{in_q}$; $k = Inversion\ sign(\alpha) = -D_{in_i} - D_{in_q}$; $k' = Inversion\ sign(\alpha') = -D_{in_i} + D_{in_q}$; $S_1 = R_{c_i} \cdot \alpha + (1 - R_{c_i}) \cdot k$; $S_2 = R_{c_i} \cdot \alpha' + (1 - R_{c_i}) \cdot k'$; $S_3 = S_1 \cdot Z^{-1}$; $S_4 = S_2 \cdot Z^{-1}$; $S_5 = R_{c_q} \cdot S_3 + (1 - R_{c_q}) \cdot S_4 = D_{out_i}$. où (D_{in_i} & D_{in_q}) représentent les données d'entrées, et S_5 représente la première sortie (D_{out_i}) dans le mode désétalement. Nous pouvons obtenir la seconde sortie (D_{out_q}) en utilisant un deuxième MUX 4:1 pipeline comme proposé dans la section 2.1. La figure III.14 (b) représente l'architecture proposée en mode désétalement (de couleur bleu). Le mode désétalement nécessite deux opérations d'addition / soustraction et deux inversions de signe (deux inverseurs de signe).
- **Le mode cellule Cordic :** ce mode est sélectionné si ' δ ' = 0 et lorsque le signal de commande sel est égal à 1. Les équations développées précédemment donnent les paramètres et les signaux de configuration de valeurs suivantes: $M = d_i \rightarrow \bar{M} = \bar{d}_i$; $sel = 1$; $\alpha = x_i + d_i \cdot y_i \cdot 2^{-i}$; $\alpha' = y_i + \bar{d}_i \cdot x_i \cdot 2^{-i}$; $S_1 = sel \cdot \alpha + (1 - sel) \cdot k$; $S_2 = sel \cdot \alpha' + (1 - sel) \cdot k'$; $S_3 = S_1 \cdot Z^{-1}$ (vers la cellule Cordic suivante); $S_4 = S_2 \cdot Z^{-1}$ (vers la cellule Cordic suivante); Où (x_i & y_i) représentent les données d'entrée, et S_3 et S_4 sont les sorties de la cellule de traitement dans le mode Cordic. La figure III.14 (c) illustre la mise en œuvre du mode cellule Cordic et les ressources utilisées (de couleur bleu). Comme présenté dans la figure III.14 (c), le mode Cordic nécessite deux opérations d'addition/soustraction et nécessite deux décaleurs (deux registres à décalage).

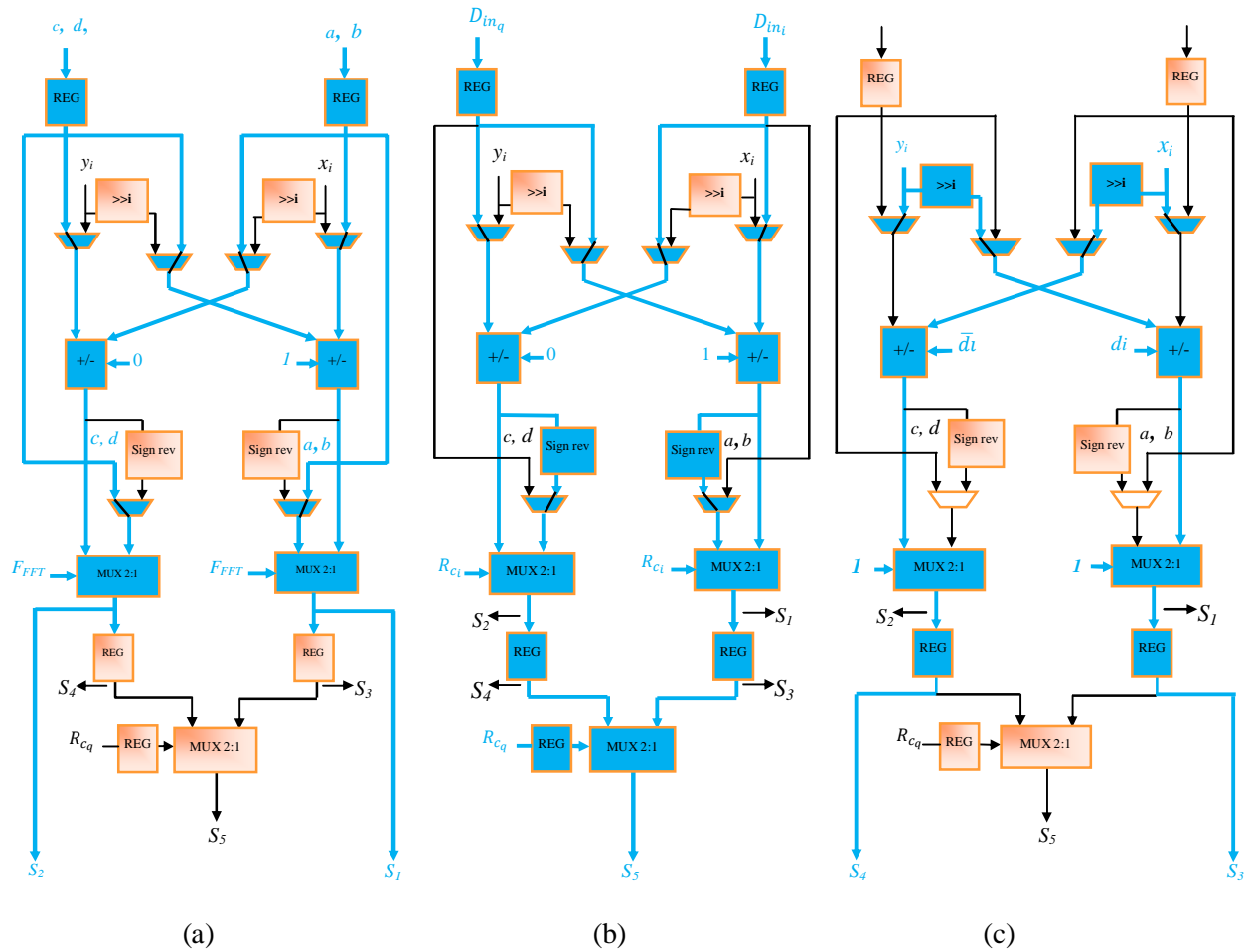


Figure. III.14. Les trois modes de fonctionnement de la cellule proposée et ses ressources:
 (a) Mode FFT, (b) Mode désétalement, (c) Mode cellule Cordic.

6. Implémentation sur FPGA, résultats et comparaisons de la performance

La description RTL de l'architecture proposée a été implémentée sur un FPGA Virtex-V de Xilinx [94] en utilisant la description structurale VHDL (VHSIC Hardware Description Language). L'outil ISE 10.1i de Xilinx [51] a été utilisé pour cette implémentation numérique permettant d'obtenir les ressources logiques nécessaires et les performances associées. Les résultats de synthèse de notre implémentation après le processus "place and route" de l'outil ISE 10.1i, sont détaillés dans le tableau III.2 pour différentes longueurs (tailles) de mots.

Plus précisément, le tableau III.2 spécifie, la fréquence de fonctionnement maximale et la consommation de ressources en termes de "logic Slice" ou "Slice Flip-Flops" (Slice registers) et les LUTs (Look Up Tables) pour différentes longueurs-mots. Ces résultats montrent que l'architecture générique proposée peut être implémentée facilement et

efficacement de façon optimisée sur la technologie FPGA tout en réduisant la complexité de l'architecture [4].

Table III.2. Comparaison de l'implémentation de l'architecture proposée avec l'architecture Velcro sur FPGA Virtex-V (XC5VLX50T).

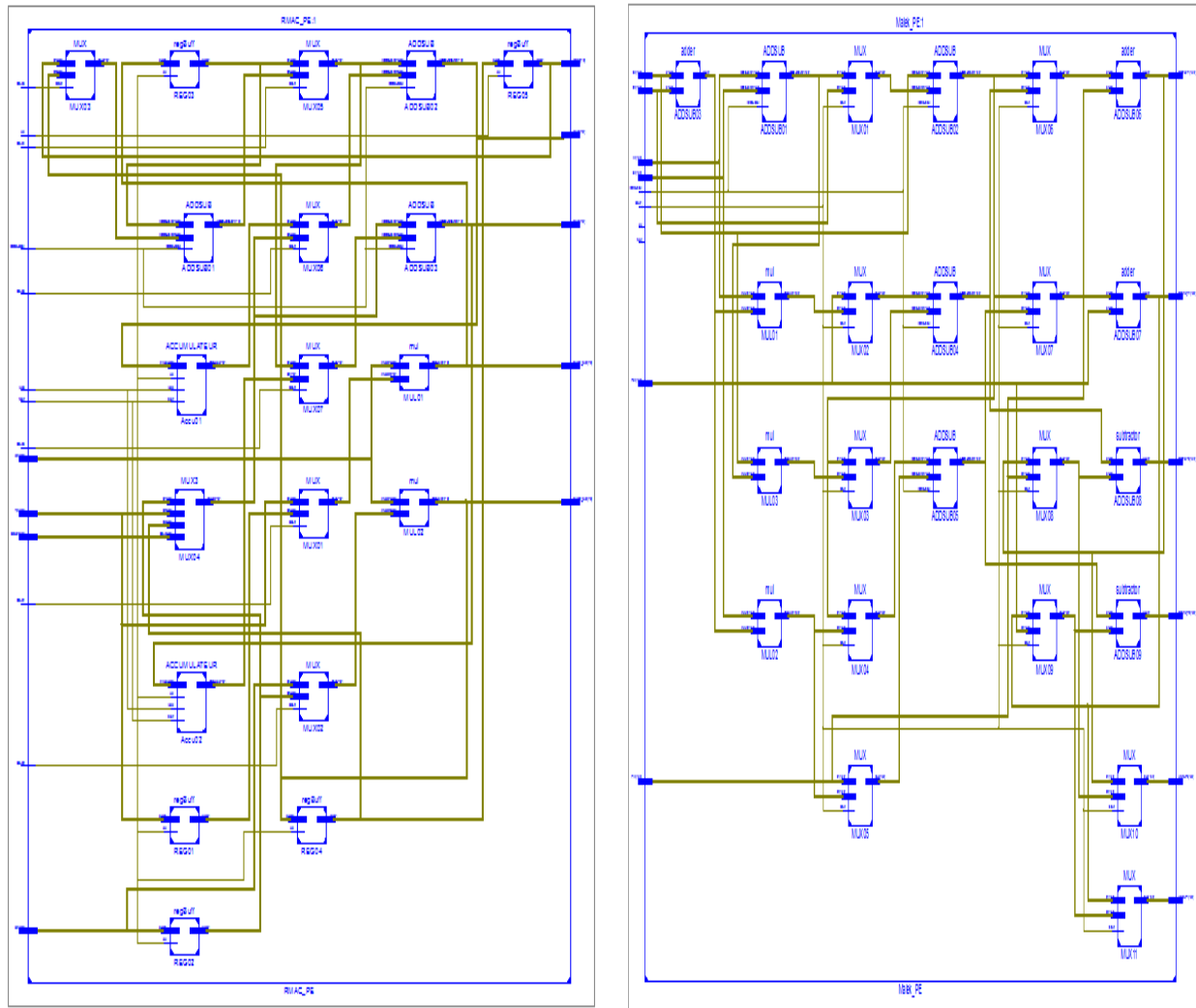
	Longueur de mots (n)					
	8 bits		12 bits		16 bits	
Paramètre	Architecture proposée	Velcro	Architecture proposée	Velcro	Architecture proposée	Velcro
Nombre de Slice Registers (19200)	32	16	48	24	64	32
Nombre de Slice LUTs (19200)	99	123	114	205	150	273
Fréquence maximale (MHz)	329.826	Unité de Désétalement : 376.605 Cellule de Cordic : 602.518 Cellule de FFT: 472.903	308.780	Unité de Désétalement : 308.447 Cellule de Cordic : 571.200 Cellule de FFT: 453.391	300.341	Unité de Désétalement : 300.026 Cellule de Cordic : 542.977 Cellule de FFT: 435.426

Cette réduction de la complexité est souhaitable pour effectuer des opérateurs pour l'unité de désétalement, le papillon FFT, et la cellule Cordic. En outre, notre architecture présente un bon compromis entre la vitesse de fonctionnement et la consommation des ressources logiques. Par exemple le résultat de l'implémentation pour une longueur de mot de 8 bits nécessite seulement 99 Slices CLB, aucun multiplicateur ou blocs DSP ou des RAM, avec une fréquence de fonctionnement maximale de 329,8 MHz.

Pour une évaluation réelle de la performance et la consommation en ressources, nous comparons les implémentations de notre architecture générique avec la technique Velcro [21] et avec quelques travaux similaires disponibles dans la littérature qui sont basés sur la technique des opérateurs communs et les architectures reconfigurables [3] [95]. Ces travaux similaires disponibles comportent certaines fonctionnalités communes avec celle proposée. Cependant la comparaison entre les opérateurs communs et/ou les cellules reconfigurables qui sont basées sur des architectures différentes et/ou ont été implémentées sur différentes technologies n'est pas simple. Comme l'implémentation dans la technologie FPGA n'a pas été introduite dans les travaux de [3] et [95] ([3] utilise un DSP et [95] utilise un ASIC), nous proposons de refaire ces travaux avec une implémentation sur FPGA de type Virtex 5. Ce qui nous permet de faire directement la comparaison de nos résultats avec ces architectures précédentes de façon équitable et appropriée. Nous avons également utilisé les mêmes métriques de la comparaison (consommation des ressources et la fréquence maximale) pour différentes précisions (longueurs de mots). La figure III.15 présente les schémas RTL pour les travaux précédents présentés dans [3] et [95] respectivement. La figure III.16 présente le schéma RTL de notre architecture proposée.

La méthode "Velcro" est la méthode conventionnelle pour créer un terminal multistandard comme nous avons dit dans le chapitre I. Dans notre cas, c'est la juxtaposition de trois architectures; une architecture pour l'unité de désétalement, une architecture pour le processeur élémentaire (FFT) et une architecture pour la cellule Cordic, et la "reconfiguration" est simplement réalisée par un simple switch pour basculer d'une architecture à une autre. H. Lange et al. [95], ont proposé un élément de traitement (PE) reconfigurable pour les algorithmes qui se basent sur la multiplication-accumulation ou multiplication-addition, le RMAC_PE (Reconfigurable Multiply-Accumulate-based Processing Element). Il se compose de deux multiplicateurs, trois additionneurs/soustracteurs, deux accumulateurs et plusieurs registres de données. Cette architecture montre une bonne flexibilité qui permet la réalisation de différentes classes d'algorithmes comme FFT/IFFT, filtrage FIR et la multiplication matrice-vecteur [95]. M. Naoues et al. [3] ont suivi la même approche que dans [95], ils ont proposé un opérateur commun pour le FFT et l'algorithme de Viterbi (FFT/Viterbi Common Operator). Le CO

FFT/Viterbi est composé de deux multiplicateurs, deux additionneurs/soustracteurs, des additionneurs, des soustracteurs, multiplexeurs et plusieurs registres de données.



a)

b)

Figure. III.15. Schémas RTL de l'implémentation sur FPGA Virtex V de : a) RMAC [95] et b) CO [3].

Le CO FFT/Viterbi [3] définit trois configurations: 1^{ère} config: FFT Radix-2 papillon, 2^{ème} config: unité ACS (Add Compare Select) pour l'algorithme de Viterbi, effectue le calcul des métriques de chemin et la sélection du chemin survivant, 3^{ème} config: unité BMC (Branch Metric Calculation) pour l'algorithme de Viterbi, effectue le calcul des métriques de branche pour chaque état du treillis [3].

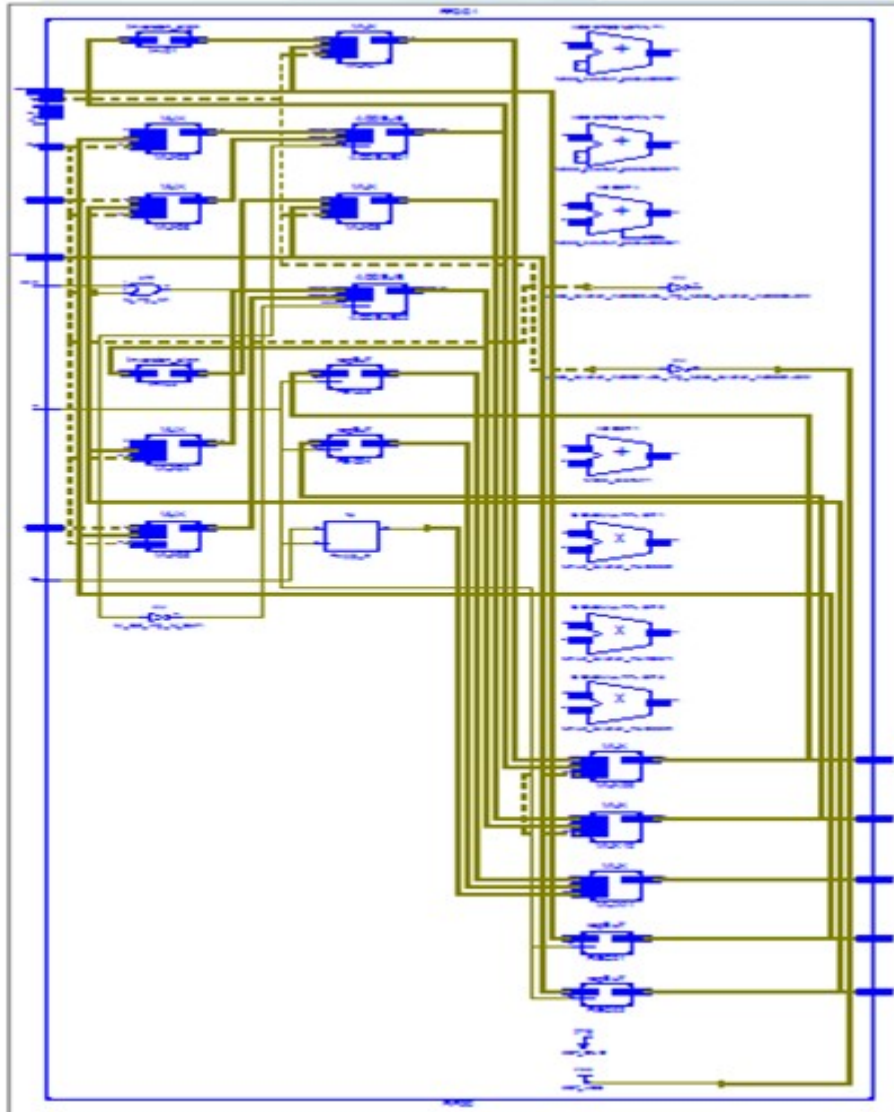


Figure. III.16. Schémas RTL de l'implémentation sur FPGA Virtex V de l'architecture proposée [4].

Les résultats de la comparaison entre l'architecture proposée et l'architecture Velcro et entre l'architecture proposée et les travaux similaires sont donnés dans le tableau III.2 et le tableau III.3 respectivement pour différentes longueurs-mots (n).

Comme le montre le tableau III.2, si on compare avec l'architecture Velcro, notre architecture fonctionne au minimum avec une fréquence de 300 MHz et offre un gain de surface allant de 5,7%, 29,25% et 29,83% pour une longueur de mots de 8, 12 et 16 bits respectivement. Par rapport à l'approche Velcro, l'architecture proposée présente une consommation de surface FPGA raisonnable, ce qui rend possible d'envisager une implémentation complète d'un processeur de réseau pour les systèmes WCDMA et OFDM.

Table III.3. Comparaison de l'implémentation sur FPGA Virtex-V (XC5VLX50T) de l'architecture proposée avec les travaux précédents [3] [95].

Fréquence maximale (MHz)	Nombre de Slice LUTs (19200)	Nombre de Slice Registers (19200)	Paramètre			Longueur (tailles) de mots n
			Architecture proposée	RMAC [95]	FFT/Viterbi CO [3]	
329.826	99	32	Architecture proposée	RMAC [95]	FFT/Viterbi CO [3]	8 bits
142.835	88	56	Architecture proposée	RMAC [95]	FFT/Viterbi CO [3]	
240.941	40	48	Architecture proposée	RMAC [95]	FFT/Viterbi CO [3]	
308.780	114	48	Architecture proposée	RMAC [95]	FFT/Viterbi CO [3]	12 bits
141.002	132	84	Architecture proposée	RMAC [95]	FFT/Viterbi CO [3]	
235.771	60	72	Architecture proposée	RMAC [95]	FFT/Viterbi CO [3]	
300.341	150	64	Architecture proposée	RMAC [95]	FFT/Viterbi CO [3]	16 bits
139.216	176	112	Architecture proposée	RMAC [95]	FFT/Viterbi CO [3]	
230.819	80	96	Architecture proposée	RMAC [95]	FFT/Viterbi CO [3]	

Bien que cette réduction de la complexité, n'est pas plus grande, et la fréquence maximale est inférieure à une architecture matérielle dédiée (Velcro). Le grand avantage de l'architecture proposée est la plus grande flexibilité, et son utilisation dans une architecture régulière. Cette flexibilité augmente le parallélisme du traitement en partageant des ressources communes entre les algorithmes, ce qui conduit à améliorer la vitesse de calcul [1][71].

De même, si on fait la comparaison avec le processeur RMAC [95] qui partage avec notre architecture proposée la fonction Cordic et une partie de la fonction FFT, notre architecture présente un gain significatif en termes de surface de FPGA d'environ 9%, 33,3% et 25,69% pour une taille de mots de 8, 12 et 16 bits, respectivement comme le montre le tableau III.3. La fréquence maximale de l'architecture proposée est également supérieure à la fréquence maximale de l'architecture RMAC [95] Ainsi, les résultats de l'implémentation démontrent clairement que notre architecture proposée peut fournir une réduction de la surface logique totale nécessaire dans le FPGA. Par conséquent, par rapport à l'architecture Velcro et l'architecture proposée dans [95], nous montrons l'intérêt de l'architecture adoptée, qui propose une solution permettant d'optimiser les ressources matérielles embarquées. La raison principale est que notre architecture est basée sur une structure RTL pipeline. Comme le montre le tableau III.3, bien que la surface logique soit faible par rapport à l'architecture FFT/Viterbi CO [3], notre architecture proposée qui effectue plus de fonctionnalités (FFT, Cordic, désétalement) améliore les performances en augmentant la vitesse de fonctionnement de 36,89%, 30,96% et 30,11% pour une longueur de mots respectivement de 8, 12 et 16 bits. Cette comparaison montre également que l'opérateur proposé offre un bon compromis entre la fréquence de fonctionnement, la consommation de la surface de FPGA et la flexibilité. En outre, notre architecture fournit une réutilisation efficace des ressources. Par exemple, dans le mode de désétalement, le pourcentage d'utilisation des ressources, est de 90% des ressources totales (figure III.14 (b)). De même, dans les modes Cordic et FFT, le pourcentage d'utilisation des ressources, est de 60% des ressources totales de l'opérateur (figure III.14 (a) et (c)), respectivement. Avec cette réutilisation efficace de ressources et la réduction de la consommation de surface de FPGA obtenu, nous attendons une consommation raisonnable de la puissance [3] [4].

En outre, le principal avantage de la conception proposée est la versatilité; au lieu de trois architectures différentes pour l'unité de désétalement, la cellule Cordic et le processeur élémentaire (PE), l'architecture proposée est capable de faire toutes ces fonctionnalités par la même structure, avec une utilisation optimale de la surface logique de FPGA et ainsi une consommation de puissance raisonnable. Cette réduction de la consommation en ressource et en puissance permet d'envisager une parallélisation éventuelle en multipliant le nombre des opérateurs communs, ce qui conduit à améliorer la fréquence

de fonctionnement. Il est ainsi possible avec cette réduction de la consommation en ressource de réutiliser notre opérateur commun par blocs dans le cadre d'un processeur de réseau complet, pour un récepteur Rake, un module de FFT-SDF, et un module Cordic [4].

7. Discussion

* **T**out d'abord, si nous considérons le coût de l'implémentation sur DSP (Digital Signal Processing) et ASIC (Circuit Application-Specific Integrated) sur lesquels sont basés les travaux précédents [3] et [95], nous pouvons faire plusieurs observations et remarques. Comme nous avons dit dans le deuxième chapitre, le DSP est flexible, mais il manque d'une puissance de calcul suffisante. Si on considère la surface (zone) de silicium, la performance de FPGA est améliorée par rapport aux DSP existants. Cependant, par rapport aux ASIC, le FPGA nécessite entre cinq et dix fois plus d'espace de silicium au même nombre de portes équivalent (gates count), avec une vitesse inférieure (En moyenne, les FPGA sont environ 7,2 fois plus lent qu'un ASIC), et une consommation de puissance encore grande (En moyenne, un FPGA consomme en puissance 12 fois plus qu'un ASIC) [96]. Néanmoins, le facteur clé pour la réduction de cet écart (gap) entre le FPGA et l'ASIC, est la disponibilité des blocs hétérogènes dans les FPGA modernes comme la mémoire et les multiplicateurs, où ces blocs peuvent réduire cet écart (gap) de surface de façon significative. En effet, l'écart de la surface, le retard du chemin critique (critical-path delay) et la consommation de puissance dynamique sont potentiellement réduits à cinq (divisés par cinq). Mais en général, ce coût n'est pas très important si on considère les autres avantages de FPGA, à savoir la flexibilité et la capacité de faire de grandes modifications des fonctions sans aucun changement de la partie matérielle. En outre, le parallélisme peut être exploité pour obtenir une grande vitesse de fonctionnement dans le FPGA. En dehors de la flexibilité et les performances, le FPGA a un autre avantage: en effet les différentes précisions (longueurs ou tailles de mots), peuvent être facilement logées à différents nœuds dans le système de telle sorte que le traitement requis, soit exactement réalisé [4].

* **N**ous pouvons comparer les fonctions accomplies par les solutions proposées dans [3], [95] et notre solution dans le cas de l'implémentation dans la même technologie FPGA. En conséquence, H. Lange et al [95], ont proposé un élément de traitement reconfigurable (RMAC_PE) pour les algorithmes basé sur la multiplication-accumulation ou multiplication-addition. Ce processeur montre une bonne flexibilité qui permet la réalisation de différentes classes d'algorithmes comme le FFT/IFFT, le filtrage FIR de valeur réelle ou complexe, les multiplications matrice-vecteur et le Cordic-PE. La bonne flexibilité dans le RMAC_PE est obtenue au détriment de la consommation de surface. En revanche, notre approche proposée dans ce travail, fournit une meilleure fréquence de

fonctionnement avec une consommation de surface faible au détriment de la flexibilité par rapport à l'architecture de [95], car notre approche à première vue est limitée à trois fonctionnalités; unité de désétalement, élément de traitement FFT-SDF et la cellule Cordic. Cela est dû au fait que la technique des opérateurs communs développée fournit des blocs logiques spécifiques adaptés aux fonctionnalités implémentées. Par conséquent, par rapport à [95], notre architecture est flexible, et montre une bonne efficacité en surface et en puissance. M. Naoues et al [3], ont suivi également la même approche que [95], mais en limitant leur approche pour deux fonctionnalités; le Radix-2 FFT butterfly et l'algorithme de Viterbi. L'architecture de décodeur de Viterbi peut être divisée en trois unités (figure III.17).

- 1- l'unité BMC (Branch Metric Calculation) qui calcule les métriques de branche associée à chaque transition du treillis,
- 2- l'unité d'addition et sélection (ACS) qui calcule les métriques accumulées et sélectionne le chemin survivant entrant pour chaque état du treillis,
- 3- l'unité de gestion de survivant (SMM) qui stocke la décision prise par l'unité ACS dans le but de fournir le chemin le plus probable à la sortie du décodeur.

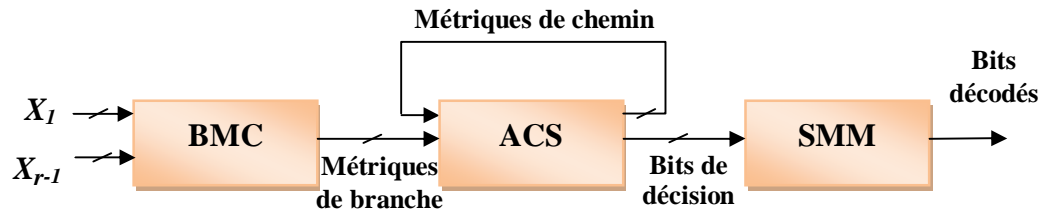


Figure III.17 – Structure globale d'un décodeur de Viterbi

En ce qui concerne l'unité BMC, on peut remarquer qu'il existe une similitude (similarité) apparente avec l'unité de désétalement dans notre architecture générique [4]. L'opération de l'unité ACS nécessite les métriques de branche BM_{00} , BM_{01} , BM_{10} et BM_{11} . Ces mesures consistent à de simples opérations de comparaison qui peuvent être obtenues par l'unité BMC, comme représentée dans l'équation (23) [3] [4].

$$\begin{cases} bm_{00} = +E_1 + E_2 \\ bm_{01} = +E_1 - E_2 \\ bm_{10} = -E_1 - E_2 \\ bm_{11} = -E_1 + E_2 \end{cases} \quad (23)$$

Selon les équations (23), l'unité BMC utilise de simples opérations d'addition/soustraction et leurs inverses, donc c'est la même conception (design) qui se trouve dans l'unité de désétalement (voir section 2). Par conséquent, nous pouvons en déduire que notre opérateur commun proposé, est capable facilement de prendre en charge le

fonctionnement de l'unité BMC [4]. De même, avec l'unité ACS, celle ci nécessite deux opérations d'additions et une opération de soustraction comme illustré dans [3], notre architecture proposée est également capable de prendre en charge l'opération de l'unité ACS, avec une légère modification de la configuration. Ce qui nous conduit à dire aussi que notre architecture, a plus de fonctionnalités donc plus de flexibilité avec un gain de surface par rapport à [3].

En ce qui concerne l'algorithme de FFT, [3] et [95] ont proposé des architectures qui implémentent le Radix-2 FFT butterfly composés d'additionneurs et multiplicateurs, tandis que notre approche est spécifique pour la méthode de FFT-SDF (Fast Fourier Transform-Single path Delay Feedback). Ici aussi, il faut noter que notre architecture peut facilement prendre la fonction de Radix-2 FFT butterfly avec une légère modification. D'abord d'après Cooley-Tukey [97], la représentation habituelle de la FFT radix-2, est le diagramme en treillis qui partitionne récursivement une FFT en éléments de calculs intermédiaires également nommés papillons. L'algorithme FFT radix-2 met à jour les échantillons intermédiaires selon l'équation 24 qui illustre les opérations réalisées par un papillon FFT Radix-2 Décimation-en-Temps (DIT) où les $y_k^{[i]}$ sont les échantillons à l'entrée du papillon et W_n^k les « Twiddle factors » de la FFT.

$$\begin{cases} y_{k+1}^{[0]} = y_k^{[0]} + y_k^{[1]} \times W_n^k \\ y_{k+1}^{[1]} = y_k^{[0]} - y_k^{[1]} \times W_n^k \end{cases} \quad (24)$$

$$\text{Si } \begin{cases} y_k^{[1]} = a + jb \\ W_n^k = c + jd \\ y_k^{[0]} = e + jf \end{cases} \quad (25)$$

Donc l'équation 24 devient :

$$\begin{cases} y_{k+1}^{[0]} = y_k^{[0]} + y_k^{[1]} \times W_n^k = e + (ac - bd) + j[f + (ad + bc)] \\ y_{k+1}^{[1]} = y_k^{[0]} - y_k^{[1]} \times W_n^k = e - (ac - bd) + j[f - (ad + bc)] \end{cases} \quad (26)$$

Pour adapter notre opérateur commun au calcul de radix 2 FFT butterfly représenté par l'équation 26 on utilise le multiplicateur complexe pipeline proposé dans le paragraphe 3.4 (initialement proposé pour l'architecture FFT-SDF représentée dans la figure III.9). Le nouvel opérateur commun adapté pour le calcul de radix 2 FFT est représenté dans la figure III.18.

L'architecture de multiplicateur complexe pipeline (figure III.9) permet d'obtenir les résultats de la multiplication de $y_k^{[1]} \times W_n^k = (a + bj)(c + dj) = (ac - bd) + j(bc + ad) = R1 + jR2$. Puis notre opérateur commun calcule les résultats finaux comme décrit dans l'équation 26. Cette nouvelle fonctionnalité ajoutée à notre opérateur

commun (mode **radix 2 FFT butterfly**) est sélectionnée par les mêmes paramètres de contrôle que dans le mode FFT-SDF, c'est-à-dire par ' β ' = 1 et ' δ ' = 1. Donc, selon les équations développées précédemment (pour la partie réelle comme illustration), nous avons les valeurs des paramètres et les signaux de configuration suivants: $M = \delta = 1 \rightarrow \bar{M} = 0$; $\alpha = e + R1$; $\alpha' = e - R1$; $S_6 = \alpha = e + (ac - bd)$ et $S_7 = \alpha' = e - (ac - bd)$. Où $(e \ \& \ R1)$ représentent les données d'entrées, et S_6 et S_7 sont les sorties dans le mode Radix 2 FFT. Nous pouvons obtenir le résultat de la partie imaginaire ($f \ \& \ R2$) par le même mode.

La Figure III.18 illustre l'implémentation du mode FFT radix 2 et les ressources utilisées (de couleur bleu). Ce mode nécessite un multiplicateur complexe pipeline et deux opérations d'addition/soustraction.

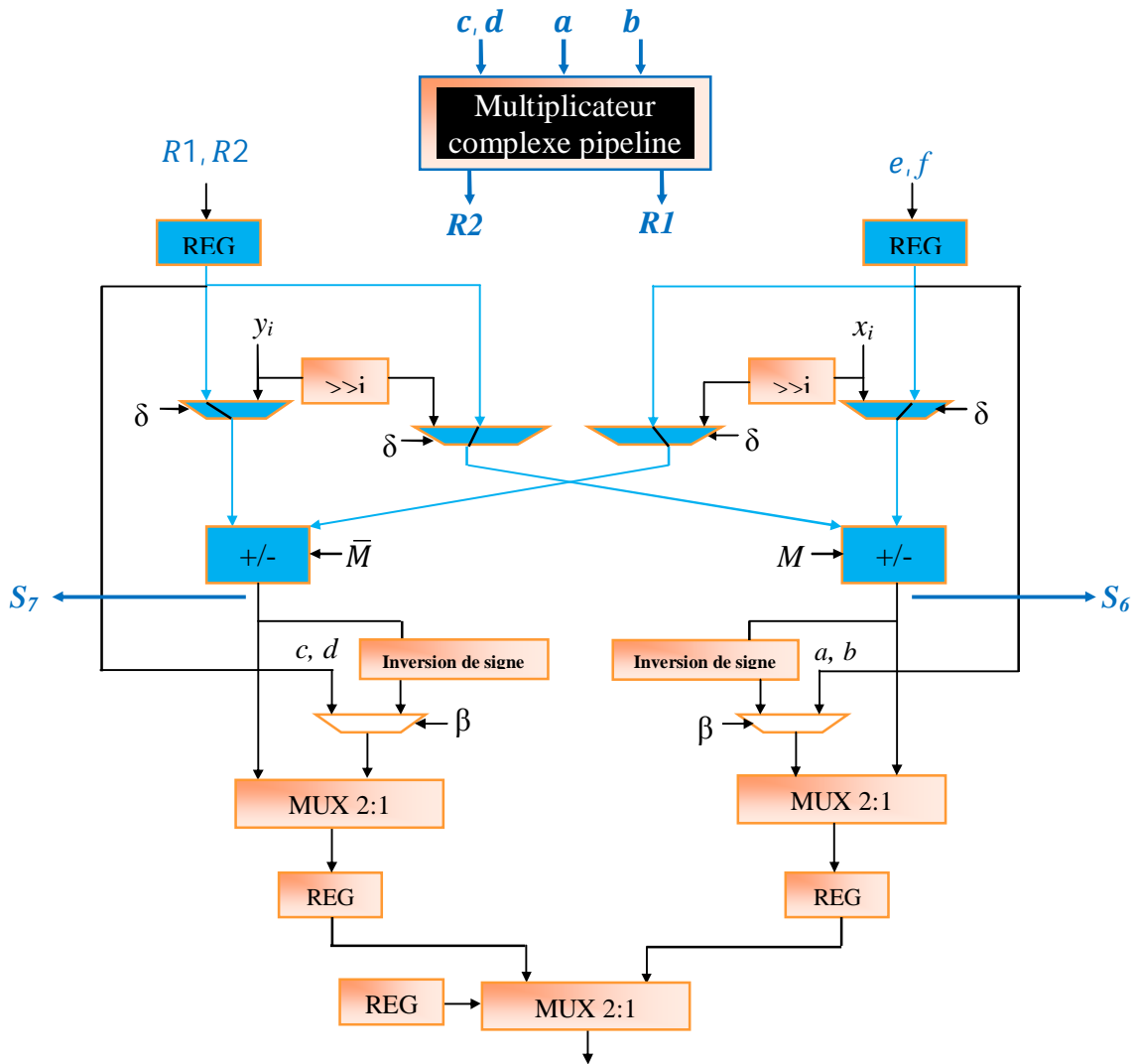


Figure. III.18. L'opérateur universel proposé pour l'unité de désétalement, processeur élémentaire PE de FFT papillon, la Cellule Cordic, et le Radix 2-FFT.

Cela nous permet de dire que notre architecture peut prendre aussi le calcul de radix 2 FFT sans aucun changement ni pour les paramètres de contrôle ni pour la structure de notre opérateur commun proposée dans la figure III.13. Néanmoins, on profite seulement de notre architecture pipeline pour la réalisation de la multiplication complexe. En plus, le Radix 2 butterfly représente une partie clé pour l'accomplissement des autres architectures des butterfly comme le radix 4 et le radix 8.

Cela nous permet de dire aussi qu'avec cette nouvelle fonctionnalité, notre opérateur commun apporte plus de flexibilité. En fait, il est devenu très convenable à implémenter non seulement les architectures FFT-SDF mais aussi d'autres architectures pipeline comme le FFT-MDC (qui est l'architecture la plus adaptée aux systèmes MIMO-OFDM), les architectures parallèles et les architectures à base de mémoire.

Revenons maintenant à l'unité ACS, celle-ci comme nous avons dit nécessite deux opérations d'additions et une opération de soustraction [3]. On peut encore profiter de l'architecture proposée pour l'opérateur commun (figure III.13) et pour le multiplicateur complexe pipeline (figure III.9) pour prendre en charge la fonctionnalité de l'unité ACS. L'opération de l'unité ACS peut se résumer par les équations suivantes [3]:

$$\begin{cases} S_8 = E_3 + E_4 \\ S_{10} = E_3 + E_4 - E_1 - E_2 = S_8 - S_9 \\ S_9 = E_1 + E_2 \end{cases} \quad (27)$$

Où les E_i et S_i sont les entres/sorties de cette unité.

L'architecture proposée pour la multiplication complexe est utilisée maintenant pour faire le calcul de $S_8 = E_3 + E_4$ et $S_9 = E_1 + E_2$ comme représenté dans la figure III. 19, puis l'opérateur commun fait le calcul final $S_{10} = S_8 - S_9$. Pour que cette nouvelle fonctionnalité ajoutée à notre opérateur commun fonctionne, il suffit de sélectionner 'δ' = 1 comme représenté dans la figure III.19. Cette figure montre aussi les ressources utilisées en couleur bleu pour l'unité ACS.

Là aussi on peut dire que notre architecture peut prendre le calcul nécessaire pour l'algorithme de Viterbi (unité BMC et unité ACS) sans aucun changement de la structure de notre opérateur commun proposée. Comme pour le FFT Radix 2 on a réutilisé l'architecture pipeline proposée pour le multiplicateur complexe.

L'architecture pipeline pour le multiplicateur complexe qui a été proposée initialement pour le radix-2/4/8 est devenue un opérateur commun pour d'autres fonctionnalités comme le Viterbi. Cela ouvre la voie aussi sur l'utilité de la cohabitation entre les différents opérateurs communs, dans le but de maximiser l'utilisation de ces opérateurs dans une implémentation SDR et ainsi d'obtenir d'autres fonctionnalités.

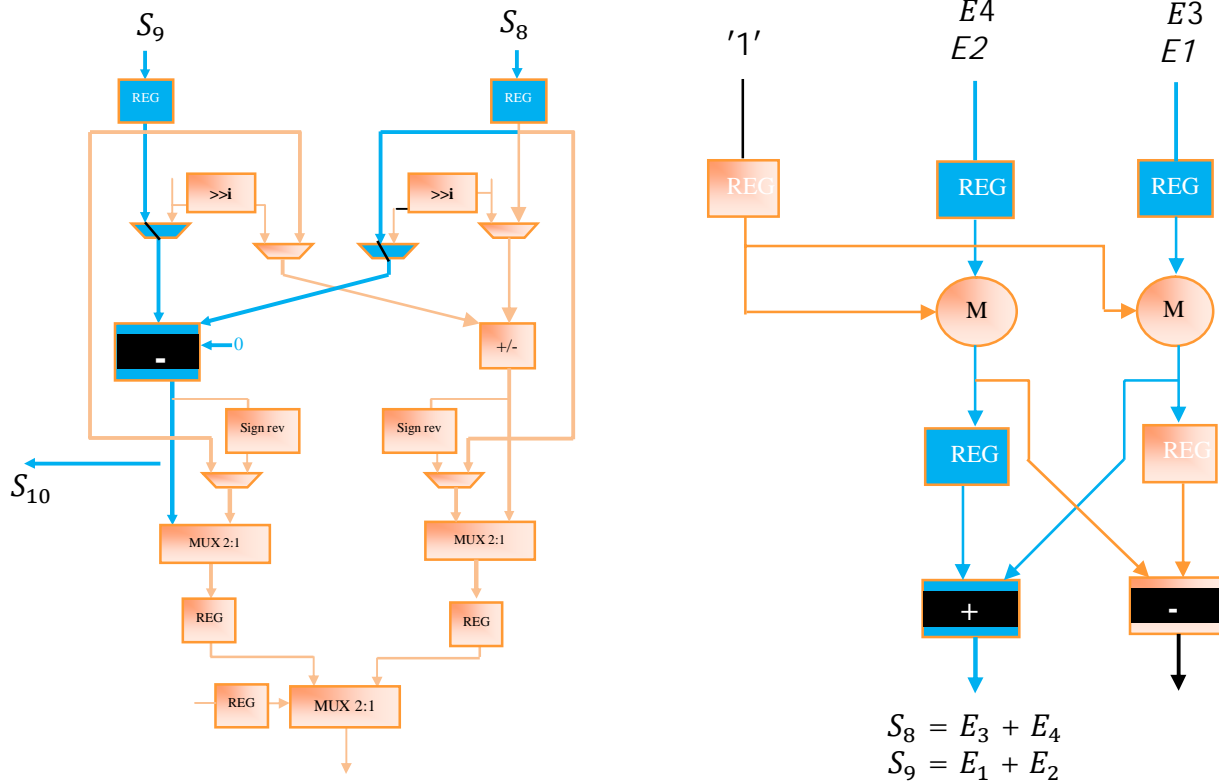


Figure. III.19. L'opérateur universel proposé adapté pour l'unité ACS de Viterbi.

* Si on combine la technique de l'opérateur commun avec la technique de synthèse architecturale qui est basée sur la réutilisation des opérateurs en ajoutant un contrôle sur le moment différent, nous pouvons également comparer notre approche versatile à la méthode basée sur le partitionnement temporel (temporal partitioning). Plus précisément, la synthèse architecturale est basée sur la réutilisation d'opérateurs à différents instants pour exécuter l'ensemble de l'algorithme [98][99][100]. Cette réutilisation est mise en œuvre par l'addition supplémentaire de ressources de contrôle. L'approche qui est basée sur le partitionnement temporel, permet de minimiser la taille du réseau (array size) d'un FPGA, en utilisant la reconfiguration dynamique (la reconfiguration dynamique consiste en l'exécution fractionnée et successive d'un algorithme par reconfiguration successive de la partie matérielle configurable) [101]. Dans le partitionnement temporel, il est nécessaire de replier temporellement différentes parties de l'algorithme à implémenter sur la même surface logique FPGA. Ceci va augmenter l'efficacité en silicium par le fonctionnement à la fréquence maximale autorisée sur la plus petite surface possible, laquelle satisfait la contrainte temps réel, mais au détriment de la consommation de puissance et parfois d'un accroissement du temps d'exécution.

Bien que le partitionnement temporel permette d'éviter un surdimensionnement des ressources nécessaires pour une application et reflète les contraintes de conception

imposées par la technologie cible, notre approche de combinaison implémente plusieurs algorithmes par les mêmes contraintes tout en maintenant la consommation de puissance [4]. La figure III.20 illustre un exemple de la mise en œuvre de la technique des opérateurs communs par la synthèse architecturale.

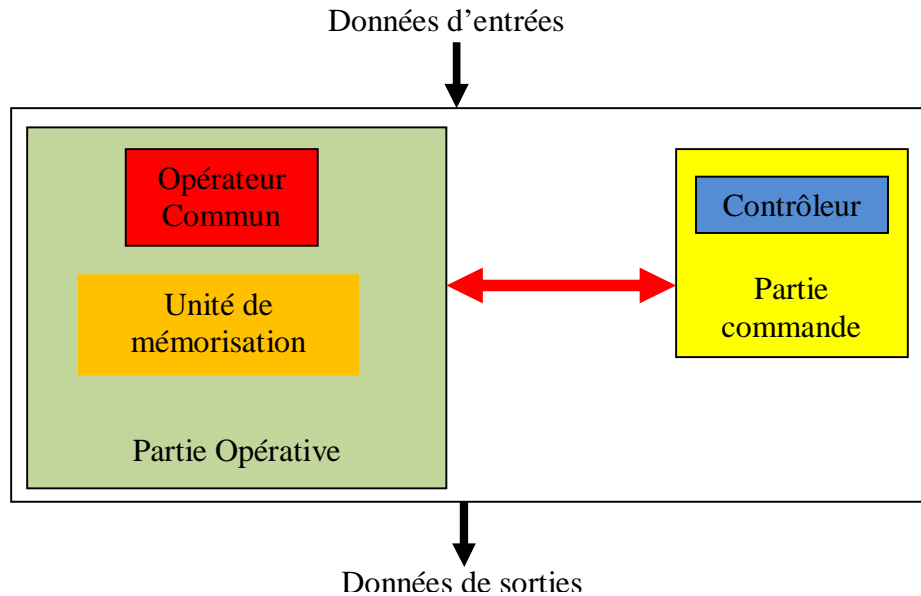


Figure. III.20. Illustration de la mise en œuvre de la technique des opérateurs communs par la synthèse architecturale.

En effet, l'approche de l'opérateur commun (OC), nous permet d'exécuter plusieurs fonctionnalités en utilisant les mêmes ressources par un simple téléchargement des paramètres. Avec l'approche de **synthèse architecturale (SA)**, on est capable de réutiliser l'opérateur commun à différents instants. Par conséquent, la combinaison de ces deux approches peut apporter les avantages suivants:

- minimiser encore le nombre de partitions temporelles nécessaires pour une application exécutée en reconfiguration dynamique, et donc apporter plus de souplesse et faciliter l'approche de la Reconfiguration Partielle d'un FPGA.
- atteindre une meilleure utilisation des ressources et atteindre la meilleure Adéquation-Algorithme-Architecture (AAA) [102][103].
- cette combinaison nous permet aussi de réaliser une implémentation optimale qui s'adapte par rapport à l'environnement et qui satisfait les contraintes (temps réel, zone logique, consommation de puissance, etc...). Ceci permet de répondre aux critères d'intégration et d'embarquabilité.

Donc, notre approche proposée de la mise en œuvre des opérateurs communs basés sur la synthèse architecturale peut être utile pour la conception d'une application générique de communications multi standard [4].

* **E**n outre, il est nécessaire de prendre en compte la consommation de puissance qui est un paramètre important dans les systèmes embarqués. En effet, dans les communications sans fil, les mobiles sont limités par leurs batteries d'alimentation. Par conséquent, la consommation de puissance est un facteur important qu'on ne peut pas négliger facilement. Dans notre cas d'étude, cette estimation de la consommation dépend de la technologie cible et l'environnement de travail comme la fréquence de travail et les ressources nécessaires pour chaque partie de la fonction commune [4]. En effet, les FPGAs sont devenus un choix intéressant par rapport aux ASICs, en raison d'un délai de mise sur le marché (time-to-market) plus court, une conception faible coût et la diminution de l'écart (gap) en termes de consommation de puissance. Ceci est dû grâce aux nouvelles générations des FPGAs de faible puissance et aux nouvelles techniques de routage [104],[105]. Bien que les ASICs ont été largement utilisés pour la conception des stations de base cellulaire, mais avec de longs cycles de conception, un coût élevé et fixe et le manque de flexibilité, ils deviennent impropres pour le prototypage et ne sont pas adaptés à un marché qui est en évolution [106].

* **U**n autre critère important est la complexité de la reconfiguration, la figure III.13 ((a), (b) et (c)) montre que l'architecture proposée peut fournir une commutation rapide entre les algorithmes et réduire la complexité de la reconfiguration par rapport aux travaux de [95].

8. Conclusion

Dans ce chapitre, on a proposé une nouvelle architecture générique et versatile pour l'unité de désétalement, le processeur élémentaire (PE) de FFT (en utilisant la méthode SDF), et la cellule Cordic. L'architecture proposée est basée sur la technique des opérateurs communs. Cette architecture est évolutive, scalable et peut être facilement adaptée pour traiter des mots de longueurs plus grandes. Pour démontrer l'efficacité de l'architecture proposée, nous avons fait la comparaison avec la technique Velcro et quelques opérateurs communs et des cellules reconfigurables disponibles dans la littérature. Toutes les architectures ont été implémentées sur la technologie FPGA Virtex-V pour une comparaison équitable. L'approche proposée améliore de façon importante le temps de traitement et montre une faible consommation de surface. En effet, les résultats de l'implémentation prouvent clairement que notre architecture proposée peut fournir une réduction de la surface logique totale nécessaire dans le FPGA, alors qu'elle fonctionne à une fréquence d'horloge supérieure à 300 MHz. En outre, l'implémentation de

l'architecture proposée nécessite peu de ressources matérielles par rapport à la technique Velcro et aux travaux précédents qui portent une certaine fonctionnalité commune. En effet, elle est capable de fournir un gain de surface logique (% des slices) sur FPGA (Xilinx Virtex- XC5VLX50T) de 5,7% , 29,25 et 25,69% pour un mots de longueurs de 8, 12 et 16 bits, respectivement. Ce gain de surface peut être amélioré, si on combine notre architecture qui est basée sur la technique de l'opérateur commun avec la technique de synthèse architecturale.

Ainsi, la conception de notre approche présente de bonnes capacités d'évolution, où on a montré que notre architecture peut facilement prendre en charge d'autres fonctionnalités. Ainsi, notre opérateur commun est défini pour effectuer des opérations élémentaires de traitement du signal indépendamment de la fonction qui l'exécute.

Enfin, et afin d'explorer des nouvelles architectures efficaces pour les terminaux mobiles, notre approche présente des performances attrayantes et peut être utilisée comme un opérateur commun dans les plus importantes fonctions pour plusieurs normes de transmission sans fil.

Chapitre IV

Architecture numérique et générique de modulation et codage pour la transmission de données

1. Introduction

Dans le chapitre I nous avons montré que les différences essentielles entre les différents standards et normes de télécommunication sont les types de modulations et les codages utilisés. En effet, la modulation employée, peut varier entre une modulation de phase (M-PSK), de fréquence (FSK, GFSK), ou une modulation d'amplitude et de phase (M-QAM). De la même manière, les codes détecteurs et correcteurs d'erreurs peuvent aller du simple code CRC aux codes les plus complexes (code convolutif, Reed-Solomon, Turbo Codes).

Le développement des architectures génériques peut générer plusieurs types de modulation dont l'intérêt touchera les systèmes RL et RC, ce qui est un objectif très intéressant. Dans ce chapitre, après une synthèse des architectures numériques pour la modulation et le codage canal, nous proposons d'abord une architecture générique et universelle pour la modulation numérique en utilisant la technique de paramétrisation.

Ensuite, nous proposons une nouvelle architecture universelle utilisée pour quatre fonctionnalités; la démodulation numérique des signaux modulés en phase et en fréquence, la génération de code de brouillage connue par le code Gold et le codage de canal.

2. Traitement numérique vs traitement analogique

Dans les chapitres I et II, nous avons remarqué que les couches les plus basses de la chaîne de transmission sont les plus sensibles aux changements de protocoles de transmission ou du standard, car elles doivent produire un signal dont les caractéristiques physiques et temporelles sont très dépendantes des spécifications du canal de communication ou du standard ciblé.

Habituellement, la plupart des architectures de la couche physique repose sur des composants électroniques analogiques. Bien que les composants analogiques soient compatibles avec les hautes fréquences de fonctionnement, leurs champs d'application et les possibilités d'évolution restent très limités. Par conséquent, les composants analogiques sont peu adaptés à l'évolution des normes, des algorithmes et des protocoles utilisés.

Plusieurs tentatives ont été effectuées pour transposer sous forme de circuits numériques les solutions technologiques ayant fait leurs preuves dans le domaine analogique [107] [108]. Cependant, la transposition des fonctionnalités analogiques en numérique introduit des chemins critiques difficiles à réduire, qui limitent la fréquence de fonctionnement et restreignent l'emploi de tels systèmes aux applications à faible débit. C'est notamment le cas pour les boucles à verrouillage de phase (PLL), présentes dans la plupart des modulateurs et démodulateurs analogiques, et dont les fonctionnalités sont très difficiles à transposer efficacement en numérique [93] [109].

Une nouvelle méthodologie proposée [109] consiste à faire une traduction directe de la fonctionnalité souhaitée en un algorithme adapté aux circuits numériques, sans passer par une étape analogique susceptible d'introduire des contraintes en contradiction avec la démarche de conception de ces circuits. Dans ce qui suit, nous proposons des architectures génériques pour la modulation et démodulation numérique en suivant la technique de la paramétrisation et la méthodologie proposée dans [109].

3. Architectures génériques pour la modulation numérique

Dans ce paragraphe nous utilisons une architecture M-QAM donc la densité de la modulation du signal transmis peut varier facilement, puis sur la base de cette architecture, nous proposons une autre architecture générique numérique pour la modulation de phase, la modulation de fréquence et la modulation QAM.

3.1. Un modulateur numérique M-QAM

La modulation QAM présente la particularité de nécessiter le contrôle de deux paramètres de la sinusoïde de sortie, à savoir, sa phase et son amplitude [9]. Un signal $s(t)$ de QAM, est représenté par l'équation (1):

$$s(t) = A_k \cos(\varphi_k) \cos(\omega t) - A_k \sin(\varphi_k) \sin(\omega t) = I \cos(\omega t) + Q \sin(\omega t) \quad (1)$$

où I et Q sont les composants in-phase et quadrature respectivement, ω est la fréquence de la porteuse, A_k et φ_k sont respectivement l'amplitude et la phase d'un signal QAM.

Un modulateur numérique QAM est composé de cinq parties principales (figure IV.1) [110][111]:

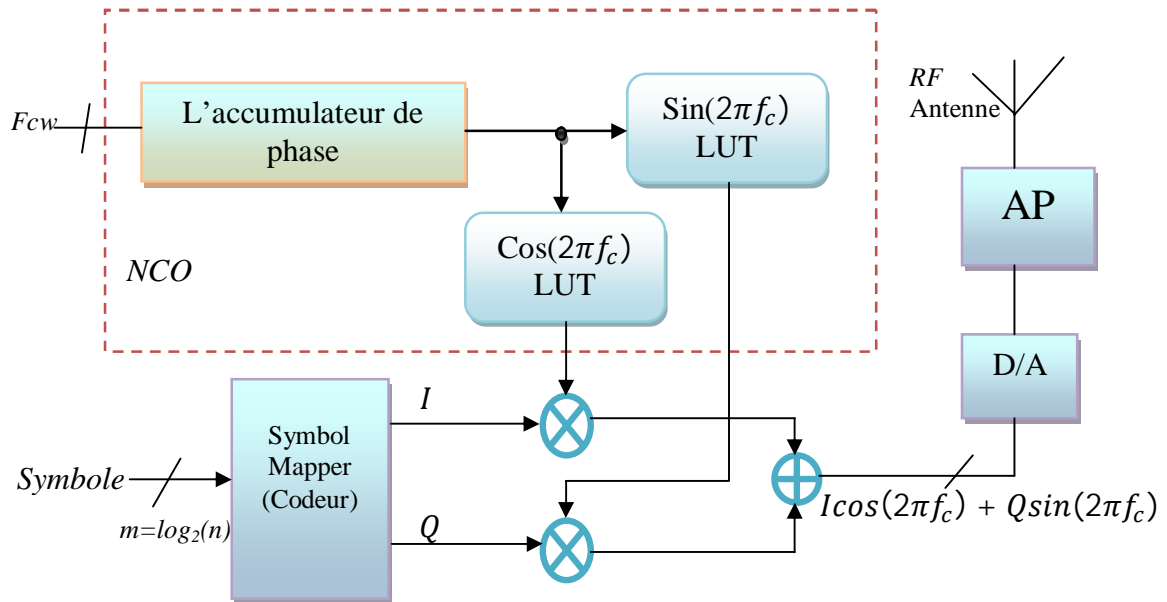


Figure IV. 1. Structure d'un modulateur QAM numérique.

– **le codeur (Symbol Mapper)**: en groupant, sous forme d'un bloc n symboles binaires indépendants, on obtient un alphabet de $M = 2^n$ symboles M-aires. Ainsi un symbole M-aire véhicule l'équivalent de $n = \log_2 M$ bits. Par exemple, pour la modulation QAM-16, chaque symbole représente 4 bits. Généralement, on utilise le codage de Gray, de tel sorte que chaque deux symboles successifs se diffèrent par un seul bit. Ensuite, le codeur (Symbol Mapper) génère les deux composants in-phase (I) et quadrature (Q) correspondant de ce symbole.

– **l'accumulateur de phase** dont le rôle est de produire le terme de phase ωt . Ce terme correspond à la phase instantanée de l'échantillon de la sinusoïde produit par le modulateur (parfois on ajoute à ce terme une phase constant φ_0). En effet, le principe de l'accumulateur de phase consiste à *recréer numériquement*, point par point, le signal que l'on désire obtenir. L'architecture de l'accumulateur de phase est représentée dans la figure IV.2.

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.

L'incrément de phase est ajouté à l'incrément de phase résultant pour chaque période d'horloge du système.

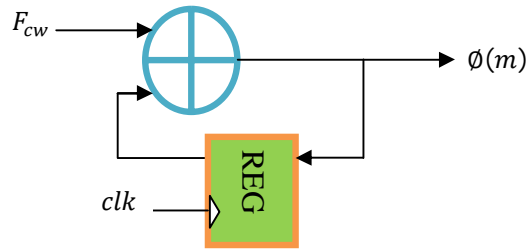


Figure. IV.2. Architecture de l'accumulateur de phase

« F_{cw} » détermine la fréquence f effective résultante du signal modulé, en fonction de la fréquence d'horloge f_{clk} de l'architecture hôte. Le m^e mot de phase produit par cet étage vaut :

$$\phi(m) = m \cdot F_{cw}. \quad (2)$$

– **le générateur numérique de fréquence** qui permet de produire les échantillons numériques de sinusoides (cosinus et sinus) à partir des paramètres que lui fournit l'accumulateur de phase. Le générateur numérique de fréquence joue le rôle d'un oscillateur à commande numérique (NCO). Généralement, il est implémenté par deux tables de correspondance LUT (ROM), dans lesquelles sont stockés les échantillons pré-calculés de la fonction cosinus et sinus. En effet, dans la mémoire sont stockées N valeurs correspondantes à une période du signal. La lecture se fait par incrément F_{cw} pour permettre de sélectionner l'adresse de la valeur désirée. Puis par une lecture successive de ces valeurs, on peut restituer le signal désiré en sortie [93]. On obtient un signal sinusoidal de fréquence [111] :

$$f_s = \frac{f_{clk} \cdot F_{cw}}{N} \quad (3)$$

– **deux multiplieurs** sont utilisés pour faire la multiplication de la composante in-phase par le cosinus issue de la première LUT, et la composante quadrature par le sinus issue de la deuxième LUT (les deux composantes I et Q sont filtrées par des filtres FIR avant d'être multiplié). Les produits sont finalement ajoutés afin de générer un signal modulé QAM conformément à l'équation (1).

– **le convertisseur numérique analogique** est le dernier élément de la chaîne de traitement numérique, dont le rôle est de convertir les échantillons en signaux adaptés au médium de transmission.

3.2. Architecture proposée pour un modulateur QAM numérique

Dans l'architecture numérique pour la modulation QAM de la figure IV.1, on remarque deux inconvénients:

- l'architecture (figure IV.1) utilise deux multiplieurs complexes pour la multiplication de la composante in-phase par le cosinus, et de la composante quadrature par le sinus. Malheureusement, la consommation d'un multiplieur complexe en ressource est grande dans le FPGA.
- l'architecture utilise des LUT (ROM), pour la génération de la porteuse (le signal sinus et cosinus). Pour obtenir une bonne précision, à la fois en phase et en amplitude de la porteuse, il est nécessaire d'utiliser une ROM de grande taille dont les conséquences sont néfastes sur la fréquence d'horloge maximale du circuit et sur la surface occupée [93].

Dont le but de réaliser une architecture pour la modulation QAM à fréquence de fonctionnement très rapide, moins occupante en surface sur FPGA et à usage universel, nous avons proposé une nouvelle architecture QAM numérique et générique représentée par la figure IV.3. La nouvelle architecture est basée totalement sur l'équation (4) qui représente un signal modulé QAM:

$$s(t) = A_k \cdot \cos(\omega t + \varphi_k) = A_k \cdot \cos(m \cdot F_{cw} + \varphi_k) \quad (4)$$

Où A_k et φ_k sont respectivement l'amplitude et la phase correspondant au symbole 'k'.

En effet l'équation (1) représente le développement de l'équation 4 car :

$$s(t) = A_k \cos(\omega t + \varphi_k) = A_k \cos(\varphi_k) \cos(\omega t) - A_k \sin(\varphi_k) \sin(\omega t) = I \cos(\omega t) + Q \sin(\omega t) \quad (5)$$

avec aussi $A_k = \sqrt{I^2 + Q^2}$ et $\varphi_k = \arctan\left(\frac{Q}{I}\right)$.

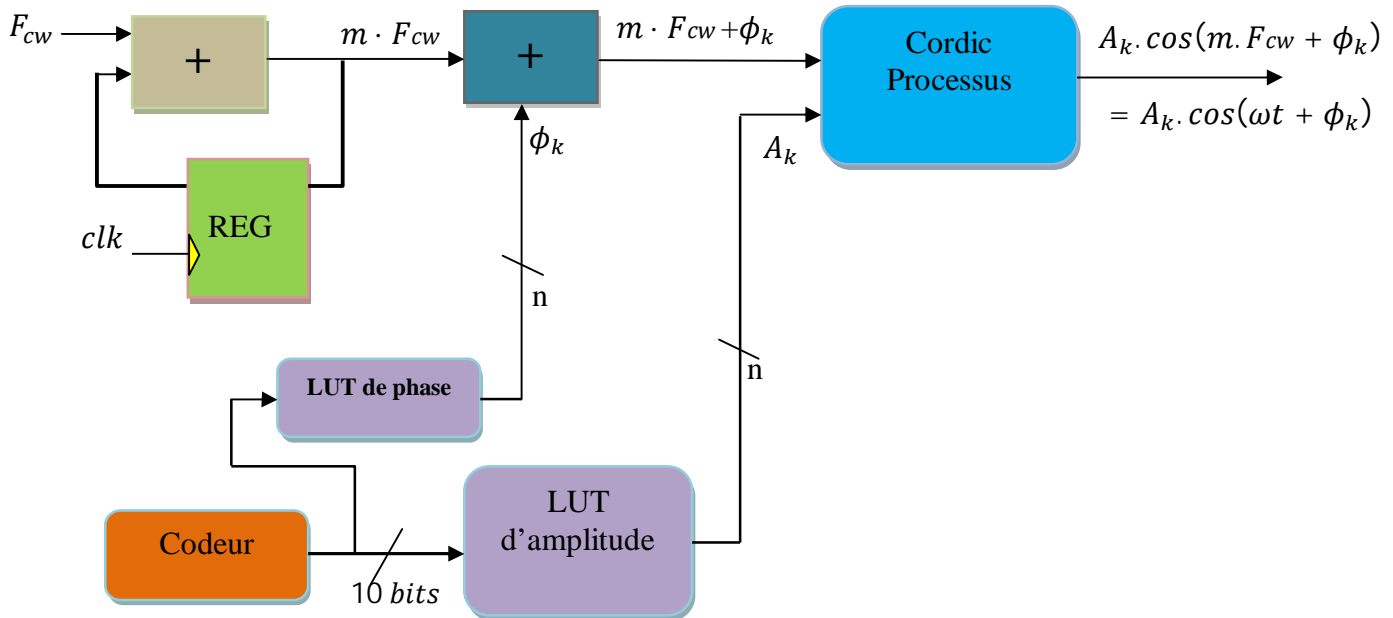


Figure IV.3. Architecture du modulateur numérique QAM proposé.

Pour comprendre le fonctionnement de cette architecture, nous détaillons ci-dessous les principales modifications ainsi que le principe de fonctionnement des composants constituant l'architecture IV.3.

3.2.1. Le Codeur

Pour mettre une architecture de modulation M-QAM universelle qui doit s'adapter facilement à plusieurs types de constellation, nous avons implémenté un codeur générique. Nous avons alors choisi 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 IV.4.

De plus, le placement des bits dans la trame utilise les propriétés de symétrie des différentes constellations [4] [90]. En effet, si on regarde l'exemple de la figure IV.5 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.

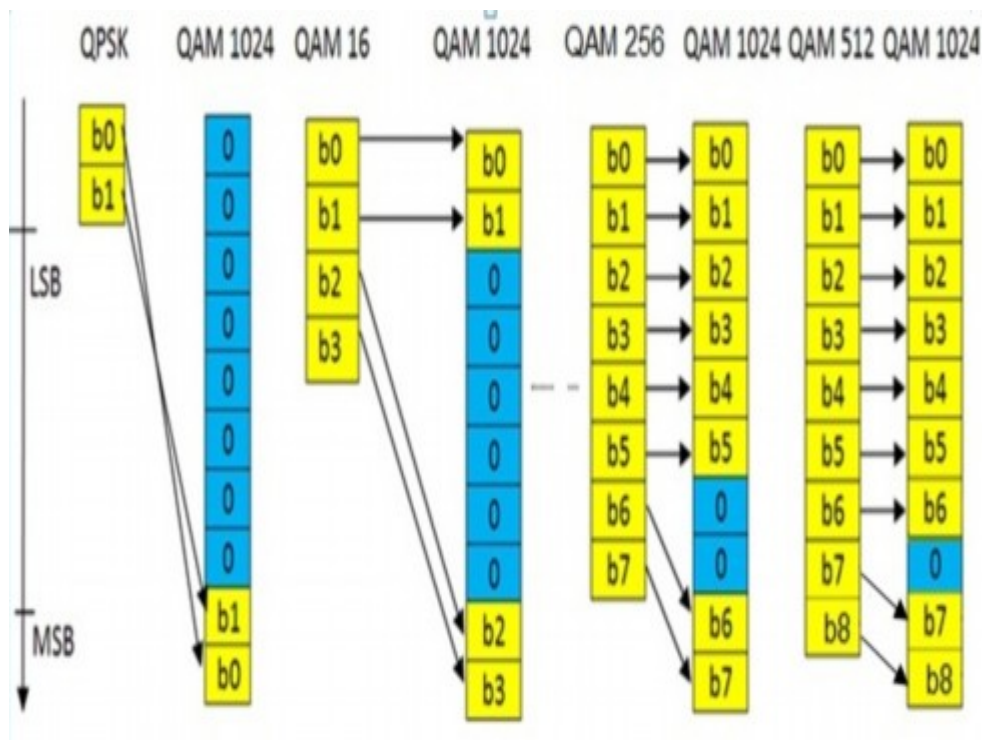


Figure IV.4. Conversion des trames au format QAM 1024.

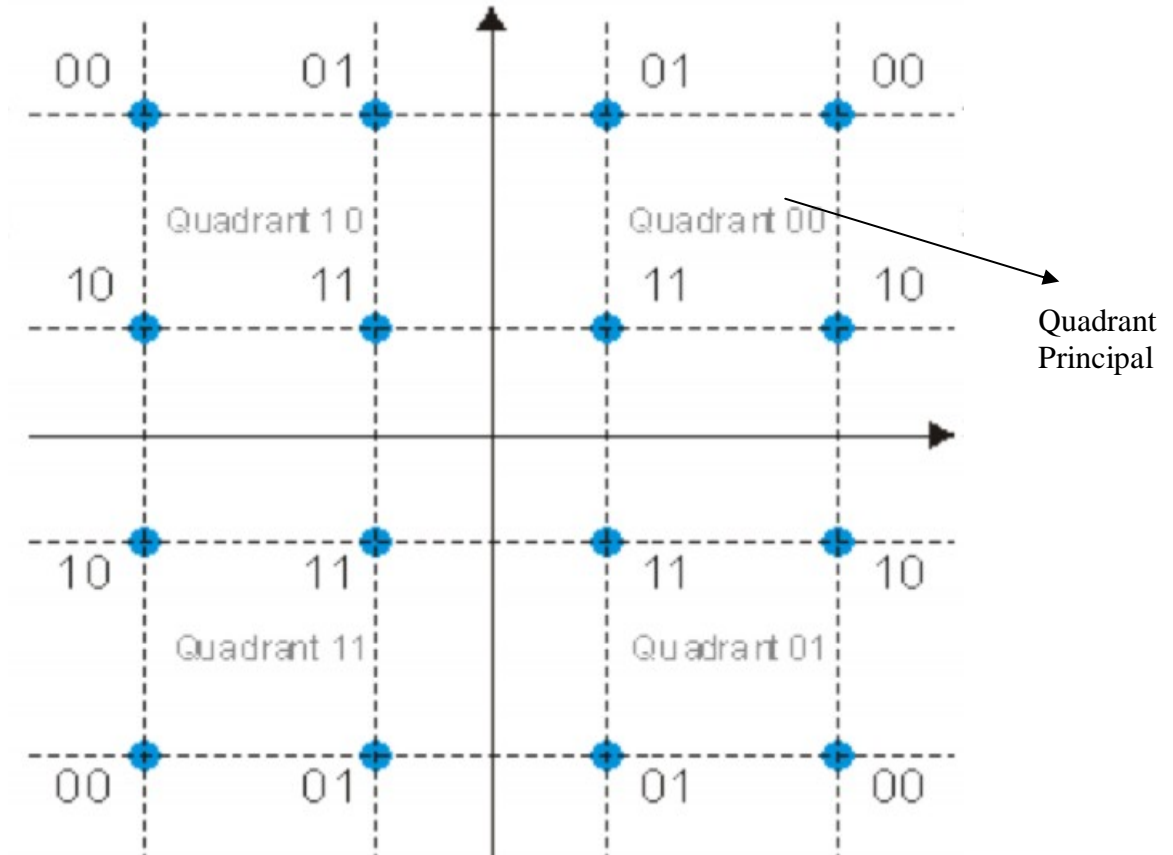


Figure IV.5. Propriété de symétrie dans la constellation QAM.

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

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

3.2.2. Le LUT d'amplitude et le LUT de phase

La sortie de ce codeur est utilisée comme adressage pour deux LUTs. La première LUT génère la phase φ_k qui correspond à chaque symbole k , et la deuxième LUT génère en fonction de cette valeur d'adressage l'amplitude A_k . Dans notre architecture proposée, nous avons donc généré directement la phase φ_k et l'amplitude A_k au lieu de générer les deux composants in-phase 'I' et quadrature 'Q'. Ceci nous permet donc de s'affranchir de

l'utilisation des deux multiplieurs complexes comme dans le cas de la figure (IV.1). La phase φ_k et l'amplitude A_k obtenues, nous permettent de générer directement le signal QAM en utilisant l'équation 4.

Puisque l'information sur le quadrant est véhiculée par les deux bits de poids fort de la trame QAM 1024, seuls les huit autres bits seront utiles pour déterminer la valeur de l'angle correspondant à la phase. Ainsi, nous avons développé un petit programme pour générer directement la valeur de l'angle correspondant à la phase.

Pour les valeurs des amplitudes, nous avons aussi développé un petit programme qui génère automatiquement une matrice contenant les valeurs nécessaires.

3.2.3. L'accumulateur de phase

Le rôle de l'accumulateur de phase est de produire le terme de phase ωt . Si on regarde l'équation (2) qui donne le m_e mot de phase produit par l'étage de l'accumulateur de phase, on remarque que celle-ci produit les valeurs suivantes: $F_{cw0}, F_{cw1}, \dots, F_{cwn}$. où $F_{cw1} = 2F_{cw0}, F_{cw2} = 3F_{cw0} \dots F_{cwn} = mF_{cw0}$.

Le m_e mot de phase produit par l'étage de l'accumulateur de phase correspond au terme « ωt ». Enfin pour produire le terme $(\omega t + \varphi_k)$, on ajoute la phase « φ_k » qui correspond au symbole transmis k . Pour cette dernière opération, on utilise un additionneur rapide. Il y a plusieurs architectures des 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 pipeline, permet d'obtenir un compromis entre la surface d'implantation et la rapidité.

3.2.4. Additionneur pipeline proposé pour le terme $(\omega t + \varphi_k)$

Cet additionneur utilisé dans l'architecture universelle de la figure IV.3, calcule la somme de la phase « φ_k » (correspondant au symbole transmis k) et le terme produit par l'accumulateur de phase, donnant ainsi le résultat sous le terme: $\omega t + \varphi_k$. Plus précisément, à chaque cycle d'horloge, l'additionneur fait l'opération d'addition de la phase φ_k avec un nouveau terme produit par l'accumulateur de phase (F_{cw}), jusqu'à la formation d'une période du signal à transmettre.

Nous avons ainsi, proposé un additionneur rapide de structure pipeline adapté à cette opération d'addition. Leur architecture interne est composée de trois additionneurs et de registres.

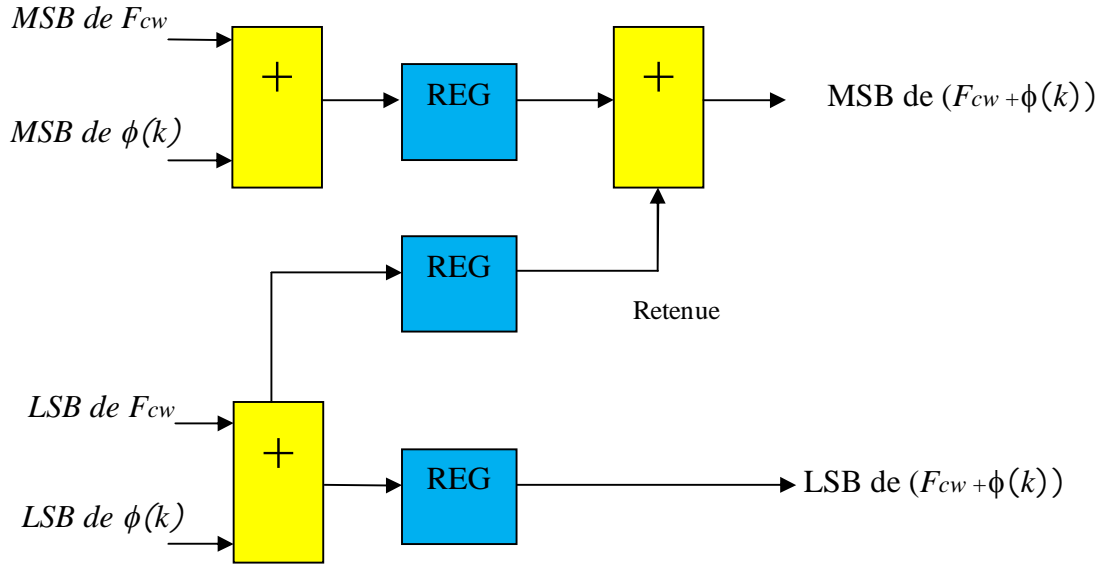


Figure IV.6. L'architecture pipeline de l'additionneur.

Dans notre cas, les deux entrées sont la phase et le terme F_{cw} . Un additionneur pour le calcul de la somme du point fort (MSB) pour les deux entrées, un additionneur pour le calcul de la somme du point faible (LSB) pour les deux entrées, et le troisième additionneur calcule la somme du résultat du premier additionneur (MSB) et la retenue (figure IV.6).

3.2.5. Générateurs de forme d'onde (Générateurs sinusoïdes)

Notre architecture proposée est basée sur l'équation 4, il reste donc à générer le cosinus du terme $\omega t + \varphi_k$, puis multiplier le cosinus avec l'amplitude A_k afin d'obtenir un signal qui représente la modulation M-QAM. Dans le modulateur QAM (figure IV.1), les valeurs des cosinus et sinus sont stockées dans des mémoires (ou LUTs). Cette solution nécessite de grandes mémoires mortes si on cherche à augmenter la précision. Cette solution ne peut être parallélisée que par le dédoublement des mémoires ou l'utilisation de mémoires à accès multiples simultanés [93]. Afin d'éviter l'utilisation des mémoires, nous avons proposé l'algorithme CORDIC pour le calcul du cosinus du terme $\omega t + \varphi_k$ (l'algorithme Cordic est détaillé précédemment). Le CORDIC permet de faire le calcul des fonctions trigonométriques comme le cosinus et le sinus par approximation successives [4][92]. Par conséquent, le terme $\omega t + \varphi_k$ avec le terme A_k représentent les deux entrées du module Cordic. Le principe du Cordic est simple [92][93]: Soit un vecteur \vec{v} , et un angle θ tels que:

$$\vec{v} = \begin{pmatrix} A_k \\ 0 \end{pmatrix} \quad (6)$$

$$\theta = \omega t + \varphi_k \quad (7)$$

L'algorithme produit une rotation qui conduit au vecteur résultant:

$$\vec{v}' = \begin{pmatrix} A_k \cos(\omega t + \phi_k) \\ A_k \sin(\omega t + \phi_k) \end{pmatrix} \quad (8)$$

Nous avons utilisé l'architecture du processeur CORDIC pipeline optimisé [93] qui permet de minimiser la consommation en ressource dans le FPGA.

3.2.6. Principe de fonctionnement de l'architecture proposée

Le principe de fonctionnement de cette architecture est simple car toutes les données, quel que soit le type de modulation QAM choisi (4-QAM, 16-QAM, ..., où 1024-QAM), seront représentées dans un format compatible avec la modulation QAM 1024. Cette tâche est réalisée par le codeur, puis les 10 bits sortant du codeur sont à la fois l'entrée d'une mémoire (LUT) contenant la phase (ϕ_k) correspondant à chaque symbole et l'entrée d'une autre LUT contenant l'amplitude (A_k) correspondant à chaque symbole. En même temps, l'accumulateur de phase, assure la génération des N points nécessaires à la description discrète du signal sinusoïdal désiré. La phase générée par la mémoire (LUT) est additionnée avec la sortie de l'accumulateur de phase en utilisant l'architecture de l'additionneur pipeline proposée. Enfin, le résultat de l'addition de $(\omega t + \phi_k)$ et de l'amplitude (A_k) est réalisé par le processus Cordic. Ce dernier calcule le cosinus de $(\omega t + \phi_k)$, puis il génère le signal QAM " $A_k \cdot \cos(\omega t + \phi_k)$ " à transmettre.

3.3. Parallélisation de l'architecture proposée pour la modulation M-QAM

Pour produire ' m ' fois plus d'échantillons d'une même sinusoïde, on parallélise ' m ' processeurs CORDIC, où chaque processeur génère un échantillon comme représenté dans la figure (IV.7). En effet, plusieurs processeurs CORDIC parallélisés en ' m ' étages sont capables de produire ' m ' fois plus d'échantillons d'une même sinusoïde d'un seul processeur CORDIC.

Cette architecture parallèle augmente la fréquence de fonctionnement malgré l'augmentation de la consommation de ressource en FPGA. Le sérialiseur rapide dans la figure (IV.7) est utilisé pour reconstituer successivement les échantillons produits par chacun des étages afin de générer le signal de sortie.

L'architecture nécessite aussi la parallélisation de plusieurs étages d'accumulateurs de phase et d'additionneurs au même nombre que celui des étages de CORDIC. On peut aussi supprimer les étages d'accumulateurs de phase et lire directement les valeurs « $m * F_{cw}$ » depuis une table d'allocation (figure IV.7). Ceci permettra de réduire la consommation de surface et de ressource de cette architecture parallèle. On peut aussi imaginer que l'utilisation du processeur Cordic optimisé de [93] va réduire la consommation en ressource mieux que si on utilise un processeur Cordic classique.

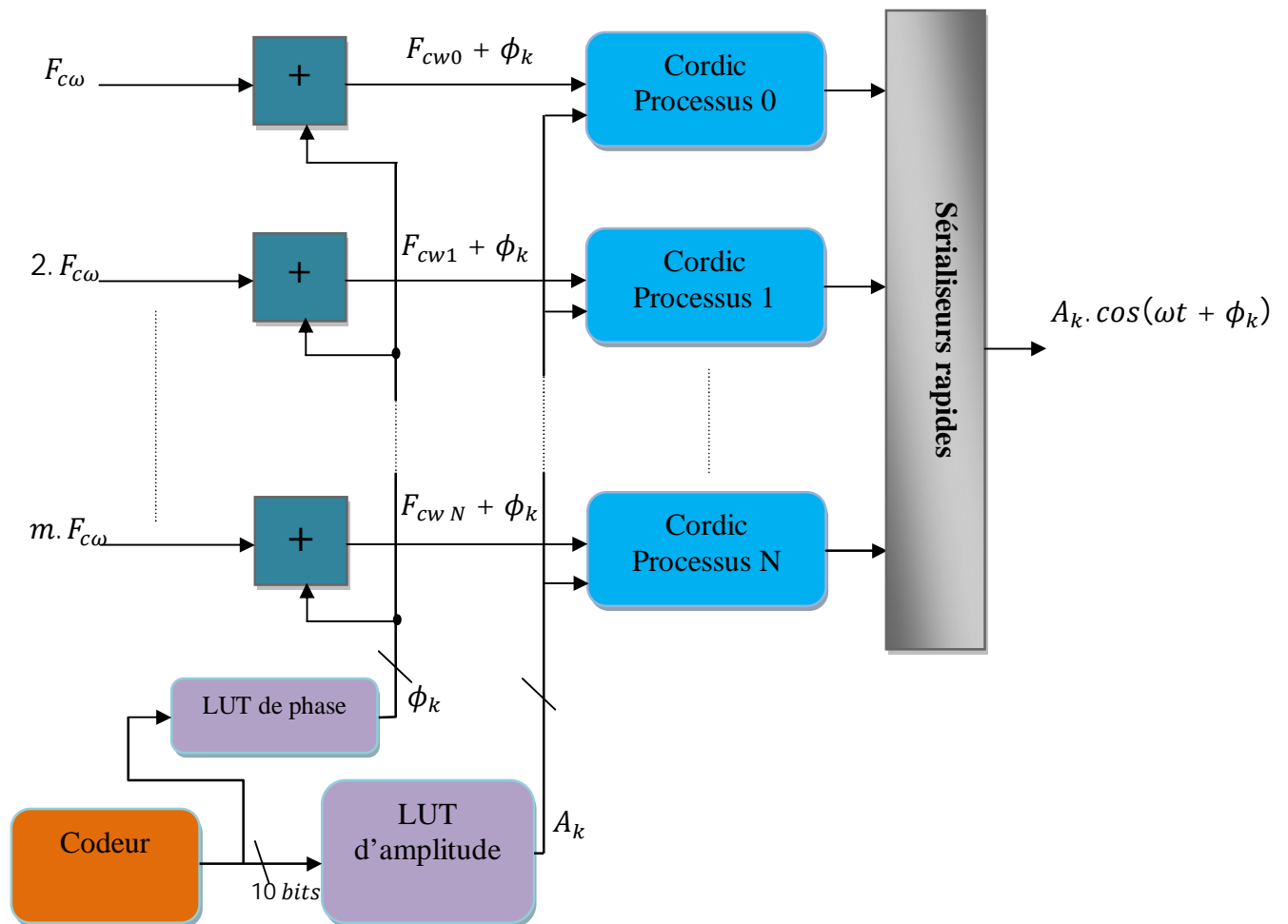


Figure IV.7. Parallélisation de l'architecture proposée pour la modulation M-QAM.

3.4. Architecture générique pour la Modulation QAM, PSK et FSK

Sur la base de l'architecture de modulation QAM numérique développée dans la section précédente, nous avons proposé une nouvelle architecture générique à usage universel (figure IV.8), qui peut prendre en charge les modulations M-QAM, M-PSK et B-FSK. Les équations des signaux modulés en phase (PSK), la modulation QAM, et la modulation FSK sont ainsi décrites [9] [112] :

- Pour la Modulation par Déplacement de Phase (MDP) qui est aussi appelée par son abréviation anglaise: PSK pour "Phase Shift Keying", l'amplitude et la fréquence sont fixes par contre la phase change avec le symbole à transmettre. On pourrait imaginer que les symboles seraient disposés de façon quelconque sur un cercle. L'équation qui décrit un signal modulé en phase (PSK) est donnée par :

$$s(t)_{PSK} = A \cdot \cos(\omega t + \theta_k) \quad (9)$$

- Pour la modulation QAM, nous avons décrit son principe de fonctionnement, et on donne l'équation qui décrit un signal modulé en modulation QAM:

$$s(t)_{QAM} = A_k \cdot \cos(\omega t + \phi_k) \quad (10)$$

- Pour la Modulation par Déplacement de Fréquence (MDF) qui est aussi appelée par son abréviation anglaise: FSK pour "Frequency Shift Keying", l'amplitude est fixe par contre c'est la fréquence qui change avec le symbole à transmettre.

Nous supposons que la modulation FSK traitée par l'architecture proposée est la modulation FSK binaire c'est-à-dire $M = 2$ (1 ou 0). Dans ce cas, le signal modulé peut être écrit très simplement comme étant composé de deux porteuses différentes.

$$\begin{cases} s_1(t)_{FSK} = A \cdot \cos(2\pi f_1 t) \\ s_2(t)_{FSK} = A \cdot \cos(2\pi f_2 t) \end{cases} \quad (11)$$

$s_1(t)_{FSK}$ la réponse d'un bit égal à 1, et $s_2(t)_{FSK}$ la réponse d'un bit égal à 0. Ceci peut être aussi écrit comme une déviation (Δf) par rapport à la fréquence porteuse.

$$\begin{cases} s_1(t)_{FSK} = A \cdot \cos\left(2\pi \left(f - \frac{\Delta f}{2}\right) t\right) = A \cdot \cos(2\pi(\Delta f_1)t) \\ s_2(t)_{FSK} = A \cdot \cos\left(2\pi \left(f + \frac{\Delta f}{2}\right) t\right) = A \cdot \cos(2\pi(\Delta f_2)t) \end{cases} \quad (12)$$

Par conséquent, l'équation qui décrit un signal modulé en fréquence (FSK) est donnée par :

$$s(t)_{FSK} = A \cdot \cos\left(\omega t + \frac{2\pi\Delta f}{2} a_k t\right) \quad (13)$$

L'architecture proposée représentée par la figure IV.8 est composée de:

- une LUT qui est utilisée pour la modulation QAM et également pour la modulation PSK, car la modulation de phase PSK peut être considérée comme une modulation MAQ où l'amplitude est constante. Donc la LUT (LUT-QAM-PSK dans la figure IV.8) contient les phases d'un signal modulé selon les modulations QAM et PSK,
- un processus CORDIC chargé de générer le signal à transmettre,
- un accumulateur de phase assure la génération des n points nécessaires à la description discrète du signal sinusoïdal désiré,
- un codeur utilisé pour la modulation QAM et la modulation PSK,
- Un bloc LUT d'amplitude, qui est utilisé seulement pour la modulation QAM puisque dans le cas de la modulation de phase et de la modulation de fréquence, l'amplitude est fixe. Ce bloc génère donc l'amplitude A_k .
- Un multiplexeur (couleur noir) est utilisé seulement pour la modulation B-FSK commandé par le symbole a_k (ce symbole peut prendre deux valeurs '0' ou '1').
- Trois commandes C1, C2 et C3 (figure IV.8) nous permettent de changer le mode de fonctionnement (type de modulation) à travers trois multiplexeurs de couleur grise dans la figure (IV.8).

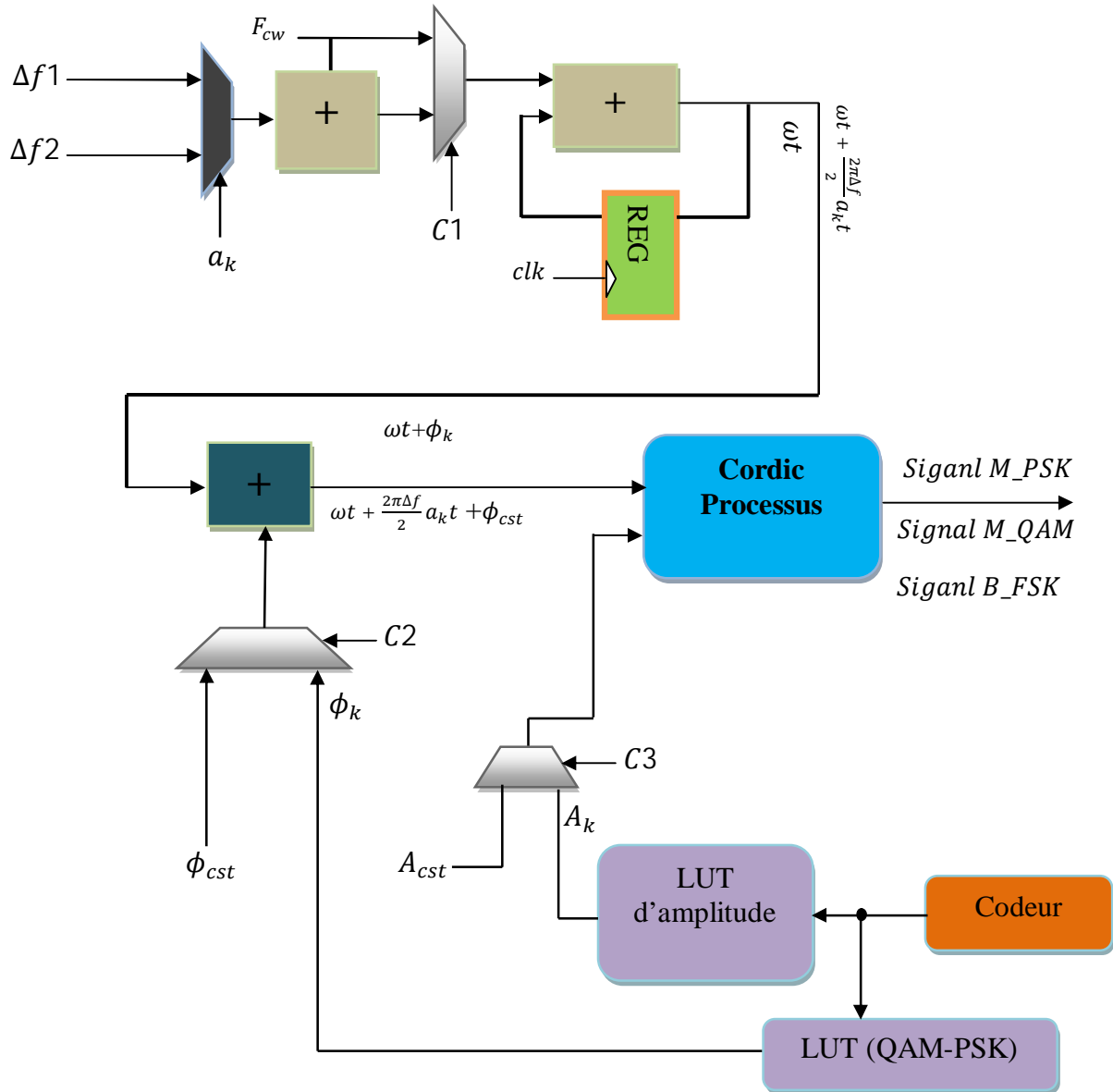


Figure IV.8. Architecture universelle pour la modulation M-QAM, M-PSK et B-FSK.

Le principe de fonctionnement est décrit par les trois modes suivants:

- **Mode 1: Modulation M-QAM**

Dans ce cas deux paramètres sont variables en fonction du symbole transmis; la phase et l'amplitude, par contre la fréquence reste fixe. Le codeur génère une trame de 10 bits pour la modulation M-QAM, où M peut varier de 2 jusqu'à 1024. La sortie de ce codeur représente une adresse de lecture pour la mémoire (LUT-QAM-PSK) qui génère la phase correspondante à chaque symbole. Selon cette adresse, le bloc LUT d'amplitude génère l'amplitude (A_k) correspondante à ce symbole. Dans ce mode, les paramètres de reconfiguration C1, C2, et C3 dans la figure (IV.8) permettent de sélectionner le mot

« F_{cw} », la phase " ϕ_k " et l'amplitude " A_k " respectivement. Simultanément, l'accumulateur de phase assure la génération des N points nécessaires à la description discrète du signal sinusoïdal désiré. Dans ce cas, l'accumulateur de phase est piloté par le mot de contrôle de la fréquence « F_{cw} » correspondant à l'incrément de phase appliqué à chaque période d'horloge du système. La phase générée par la mémoire (LUT) est additionnée avec la sortie de l'accumulateur de phase. Enfin, la sortie de l'additionneur ($\omega t + \phi_k$) et l'amplitude (A_k) sont utilisées par le processus Cordic. Ce dernier calcule le cosinus de ($\omega t + \phi_k$), puis il génère le signal M-QAM ($A_k \cdot \cos(\omega t + \phi_k)$) à transmettre.

- **Mode 2: Modulation M-PSK**

Dans ce cas, la phase ϕ_k est variable en fonction du symbole transmis; donc l'amplitude et la fréquence sont fixes. Le codeur génère une trame de 10 bits pour la modulation PSK. Cette trame de 10 bits représente une adresse de lecture pour la mémoire (LUT-QAM-PSK) contenant la phase correspondante de chaque symbole. Dans ce mode, les paramètres de reconfiguration C1, C2, et C3 dans la figure (IV.8) permettent de sélectionner le mot « F_{cw} », la phase " ϕ_k " et l'amplitude constant " A_{cst} " respectivement. Simultanément, l'accumulateur de phase assure la génération des N points nécessaires à la description discrète du signal sinusoïdal désiré. La phase générée par la mémoire (LUT-QAM-PSK) est additionnée avec la sortie de l'accumulateur de phase. Enfin, la somme ($\omega t + \phi_k$) et la valeur A_{cst} représentent les deux entrées nécessaires pour le processus Cordic qui va générer le signal à transmettre modulé en phase (M-PSK).

- **Mode 3 : Modulation B-FSK**

Ce mode utilise le changement de la fréquence en fonction du symbole transmis a_k . Dans notre architecture, la valeur du symbole à transmettre a_k (0 ou 1), permet de sélectionner la valeur de la déviation qui correspond à la modulation de fréquence (équations 12 et 13) en utilisant un multiplexeur (coloré en noir dans la figure IV.8). Simultanément, la valeur de la déviation correspondant au symbole a_k est ajoutée au mot de contrôle de la fréquence « F_{cw} » correspondant à l'incrément de phase appliqué à chaque période d'horloge du système. Ainsi, le paramètre de configuration C1 permet de sélectionner le résultat de l'addition de ($F_{cw} + a_k \cdot \Delta f$). Dans ce cas, l'accumulateur de phase est piloté par le mot de contrôle de la fréquence « $F_{cw} + a_k \cdot \Delta f$ », il assure donc avec ce nouveau mot de contrôle, la génération des N points nécessaires à la description discrète du signal sinusoïdal désiré. Les paramètres de reconfiguration C2, et C3 dans la figure (IV.8) permettent de sélectionner respectivement une phase constant " ϕ_{cst} " et une amplitude constant " A_{cst} ". Enfin, la sortie de l'accumulateur de phase et la valeur de l'amplitude représentent les deux entrées nécessaires pour le processus Cordic qui va générer et transmettre le signal modulé en fréquence (B-FSK).

Cette architecture générique pour la modulation M-QAM, M-PSK et B-FSK peut être parallélisée pour augmenter la fréquence de fonctionnement et générer plus d'échantillons.

4. Architecture générique pour la démodulation numérique et le codage.

Dans cette partie, nous proposons une nouvelle architecture universelle utilisée pour quatre fonctionnalités. Elle peut être utilisée comme un démodulateur de phase (PSK), comme un démodulateur de fréquence (FSK), comme un générateur de code de brouillage connu sous le nom de code Gold utilisé dans les systèmes WCDMA, et comme un code convolutif (codage de canal). Cette architecture est basée sur un simple registre LFSR (*Linear Feedback Shift Register*) configurable. L'idée principale est de proposer une structure générique qui permet de basculer entre les quatre fonctions par un simple changement de paramètres, en suivant la technique de paramétrisation et l'approche d'opérateurs communs.

4.1.Registres à décalage à rétroaction linéaire (LFSR)

Lorsqu'une étude bibliographique est faite sur les architectures numériques pour la modulation, le codage de canal, le filtrage etc., nous pouvons trouver de nombreux points de similarités entre ces architectures [2] [4] [113] [114]. Parmi les points de similarités intéressants, on remarque que les registres LFSR sous forme binaire, sont presque présents dans tous les systèmes électroniques actuels. Ils jouent un rôle essentiel dans la génération de séquences, le codage de canal (correction et détection d'erreurs), les codes de brouillages et le filtrage [2][115]. Nous pouvons les trouver dans les téléphones mobiles, les modems et presque toutes les applications de codage [2].

Un registre à décalage à rétroaction linéaire, ou LFSR (acronyme de l'anglais *linear feedback shift register*) est un registre où lorsque l'horloge avance (l'horloge du circuit où l'architecture est implantée), le signal se déplace à travers le registre, d'une bascule à une autre (figure IV.9). Un LFSR peut être formé en réalisant un **OU** exclusif sur les sorties des deux ou plusieurs bascules (r_i sur la figure IV.9) qui alimentent de nouveau l'entrée de l'une des bascules (figure IV.9). Comme le registre possède un nombre fini d'états, il doit entrer dans un cycle répétitif. Cependant, un LFSR avec une fonction de rétroaction bien choisie peut produire une séquence de bits qui apparaît dans la nature aléatoire et qui a un très long cycle [93] [116].

Les deux types d'architectures (Many-To-One et One-To-many) permettant de réaliser des LFSR sont présentés sur la figure (IV.9). Un choix judicieux des coefficients $\{c_i\}_{1 \leq i < n}$ et $\{c'_i\}_{1 \leq i < n}$ fournit l'état de ces deux architectures. En outre, les coefficients " c_i " peuvent prendre deux valeurs "1" ou "0". Ainsi, quand " $c_i = 1$ " il y a une connexion physique, et lorsque " $c_i = 0$ ", il n'y a pas de connexion physique.

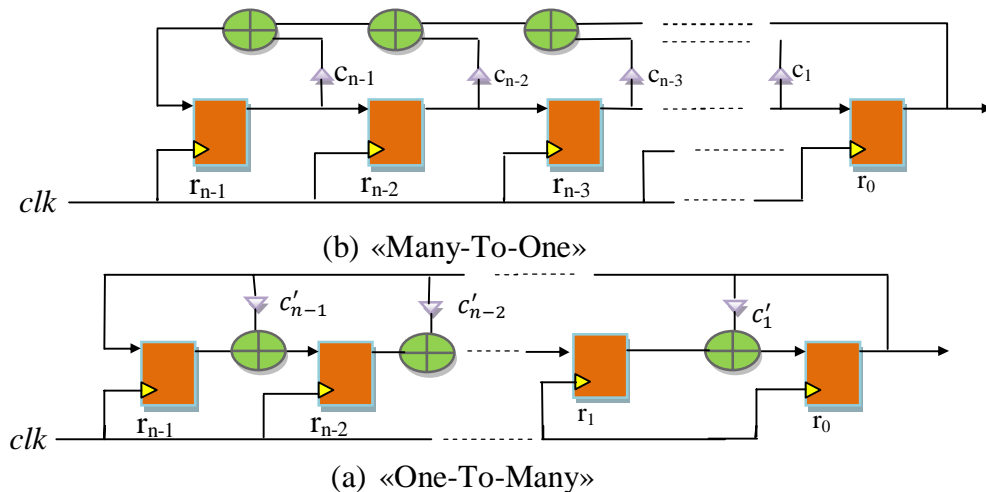


Figure IV.9. Les deux architectures de LFSR.

4.2. Le LFSR dans le codage de canal (code de convolution et turbo code)

Nous avons sans cesse besoin d'augmenter les débits de transmission tout en gardant ou en améliorant leur qualité. Malheureusement, différents types de bruits existent dans le canal de transmission ou dans les composants de la chaîne de transmission. Par conséquent, nous devons appliquer des techniques et envisager des stratégies [114] [116] pour assurer une transmission fiable des données:

- les stratégies ARQ (*Automatic Repeat Request*) qui sont basées sur la détection d'erreur et la retransmission; une demande envoyée à l'émetteur pour répéter le message sur la détection d'erreur par le récepteur,
- les stratégies Forward Error Correction (FEC) qui corrigent automatiquement les erreurs détectées au niveau du récepteur,
- Stratégies hybrides qui combinent ARQ et FEC.

Un code correcteur d'erreur (stratégies FEC) permet de corriger une ou plusieurs erreurs dans un mot de code en ajoutant aux informations des symboles redondants. Différents codes possibles existent mais dans notre travail, nous traitons seulement les codes Convolutifs et les Turbo code car ils présentent une similarité avec le code Gold et les démodulateurs numériques à base de registres LFSR. De plus ce sont les codes les plus utilisés dans les standards de télécommunications [11].

Le principe du codage convolutionnel a été introduit en 1955 par Elias comme alternative aux codes blocs [117]. Les codes convolutionnels ajoutent une certaine redondance aux bits de la séquence d'information à transmettre moyennant une opération logique (OU exclusif). Chaque information d'un symbole de m bits se transforme en un symbole de n bits, où m/n est le taux de code ($n \geq m$). Chaque codeur convolutionnel est caractérisé aussi par la longueur de contrainte K qui représente le nombre de cellules du registre à

décalage. Le code convolutif peut être divisé en deux catégories principales: le codeur non systématique (NSC) et le code convolutif récursif et systématique (RSC). Un code non systématique est celui dont la sortie ne contient pas les bits d'entrée. Dans sa forme la plus commune, un NSC est représenté par un *registre à décalage linéaire* et chaque code de sortie dépend d'un polynôme générateur dont l'équation pourrait être considérée comme l'équation d'un filtre de type FIR (Finite Impulse Response) [118]. Au contraire, un code systématique récursif est celui dans lequel la sortie contient les bits d'entrée plus une récurrence linéaire de la première sortie. Ainsi, une RSC est un codeur qui utilise un *registre à décalage à rétroaction linéaire (LFSR)*. Dans ce type de registre l'équation du polynôme générateur est une équation d'un filtre de type IIR (infinie Réponse Impulsee):

$$y[n] = \sum_0^N b_k \cdot x_{n-k} - \sum_1^N a_k \cdot y_{n-k} \quad (14)$$

où b_k et a_k sont respectivement les coefficients directs et de retour, N étant l'ordre du filtre.

Les codeurs convolutifs peuvent être mis en œuvre en utilisant l'une des deux structures équivalentes présentées dans la section suivante; le MTO et l'OTM.

4.2.1. Architecture MTO (Many To One).

La forme canonique d'un codeur convolutif MTO est représentée dans la figure IV.10 pour une mémoire du codeur de taille $m = 3$. où r_i représente la bascule i du registre R, g_i et h_i sont respectivement les coefficients du polynôme de la boucle de retour et du polynôme de la branche directe. Il y a deux configurations possibles :

- un codeur Convolutif non récursif (CNR) est obtenu, lorsque $(g_0, g_1, g_2, g_3) = (0, *, *, *)$ or $(1, 0, 0, 0)$, * égal à 0 ou 1.
- un codeur convolutif récursif (CR) est obtenu, lorsque $g_0 = 1$ et $(g_1, g_2, g_3) \neq (0, 0, 0)$.

La boucle de rétroaction est inséré uniquement, lorsque $g_0 = 1$ et au moins un des coefficients g_1, g_2, g_3 est égal à 1.

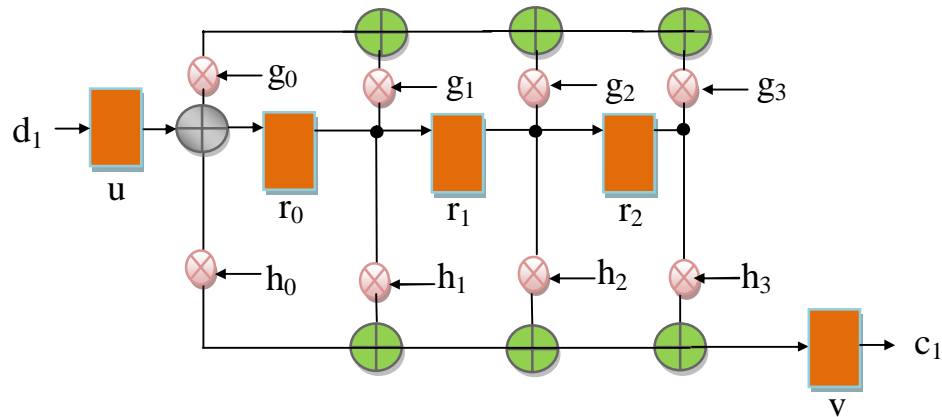


Figure IV.10. Codeur convolutif MTO pour $m = 3$.

4.2.2. Architecture OTM (One To Many)

La forme canonique d'un codeur convolutif série OTM est représenté dans la figure IV.11 pour $M = 3$. La configuration de la structure de l'encodeur OTM est similaire au MTO avec deux configurations possibles: Un codeur CNR est obtenu, lorsque $(g_0, g_1, g_2, g_3) = (*, *, *, 0)$ or $(0, 0, 0, 1)$. * égal à 0 ou 1, Un codeur convolutif récursif (CR) est obtenu, lorsque $g_3 = 1$ et $(g_0, g_1, g_2) = (0, 0, 0)$. La boucle de rétroaction est insérée uniquement lorsque $g_3 = 1$ et au moins l'un des coefficients g_0, g_1, g_2 est égal à 1.

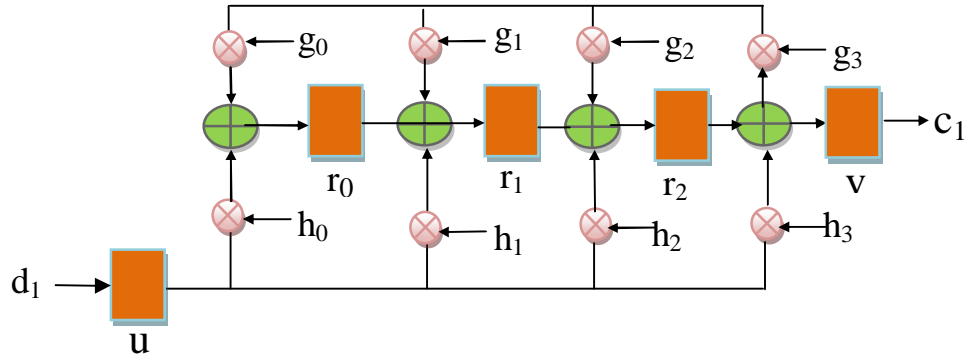


Figure IV.11. Codeur convolutif OTM pour $m = 3$.

4.2.3. Un turbo-code

Les turbo-codes peuvent être considérés comme une classe de codes concaténés. Un turbo-code est la concaténation d'un ou de plusieurs codes (convolutifs ou blocs) séparés par des blocs d'entrelacement. Suivant le type de la concaténation, nous avons les turbo-codes série et les turbo-codes parallèle.

Un turbo-code parallèle est construit par concaténation parallèle de deux codes de convolution séparés par entrelacement. Le premier code fonctionne avec la séquence d'informations et le second fonctionne avec la même séquence d'informations, mais entrelacées. La figure (IV.12) représente un turbo-code composé de deux codes constitutifs C_1 et C_2 de type RSC.

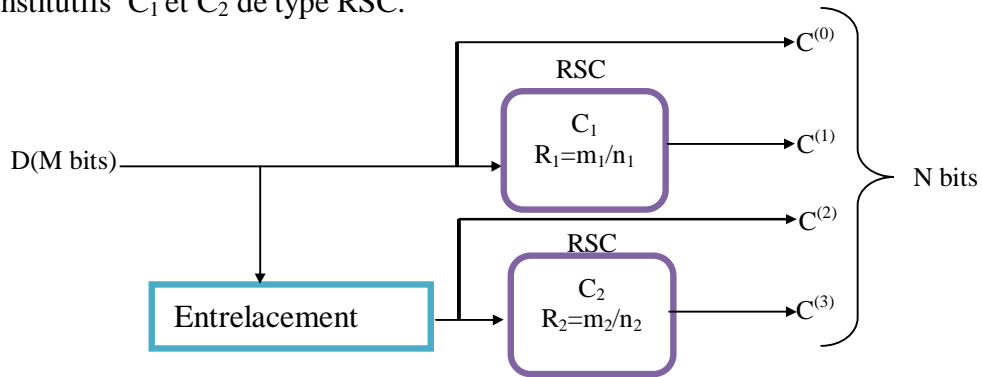


Figure. IV.12. Turbo-code parallèle.

Dans la pratique, nous choisissons les deux codes similaires et de structure RSC. En outre, afin d'augmenter la production, il est recommandé de ne pas utiliser la sortie systématique $C^{(2)}$.

4.3. Le LFSR pour le générateur de codes d'embrouillage dans le WCDMA

L'architecture du générateur de code Gold, basée sur une combinaison de deux registres à décalage à rétroaction linéaire (LFSR), est détaillée dans l'annexe A (section A.5.2).

4.4. LFSR pour la démodulation numérique

Les registres LFSR sont capables de produire des séquences fixes et périodiques, en fait c'est un choix pertinent pour une utilisation en tant que compteurs programmables [93] [109]. L'organisation fonctionnelle de l'un de ces compteurs (figure. IV.13) est composée de deux parties principales:

- un LFSR registre contrôlé par un polynôme primitif de degré n ,
- une ROM décodage.

Le rôle de cette ROM est de permettre la conversion entre l'ordre de parcours des éléments de corps de Galois $GF(2^n)$ et l'ordre «traditionnel» croissant utilisé dans les compteurs classiques. Une architecture, en respectant ces critères, se comporte comme un compteur modulo $2^n - 1$ [93][109].

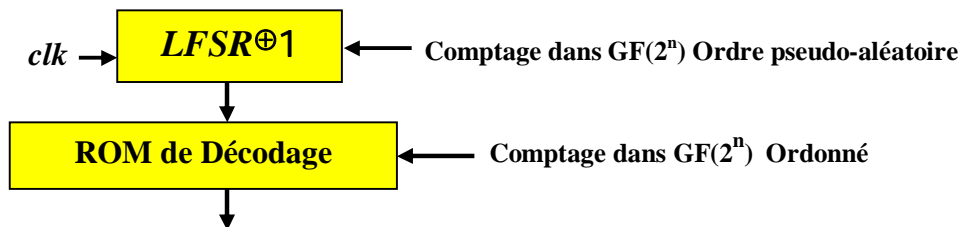


Figure. IV.13. Principe du compteur LFSR.

De nouvelles architectures [93][109] traduisent directement les fonctionnalités nécessaires à la démodulation des signaux en algorithmes adaptés aux circuits numériques et aux FPGA en particulier, ceci basant sur les registres LFSR. Les architectures des démodulateurs [93] utilisent ces compteurs pour mesurer directement la phase ou la fréquence des signaux qui lui sont fournies. L'architecture de quatre registres LFSR est conçue pour mesurer la phase dans le cas de la démodulation de phase, et celle de deux registres LFSR pour mesurer la fréquence dans le cas de la démodulation de fréquence [93][109].

4.5. Architecture générique proposée pour le codage et la démodulation numérique

A partir des architectures présentées précédemment pour la démodulation (phase et fréquence), pour le générateur de code de brouillage et pour le codage de type turbo code, nous pouvons remarquer que la principale similarité entre les quatre architectures est la

présence des registres LFSR [2]. Après avoir identifié ce point de similarité entre les architectures étudiées, nous essayons de proposer une architecture générique unique capable de prendre en charge les quatre fonctions tout en réduisant l'occupation de surface sur FPGA.

L'architecture universelle proposée (figure IV.14), est composée de quatre registres LFSR configurables, deux portes XOR, un bloc d'entrelacement, deux LUT (ROM), un bloc pour le calcul de la différence, un registre de données et un multiplexeur de configuration [2].

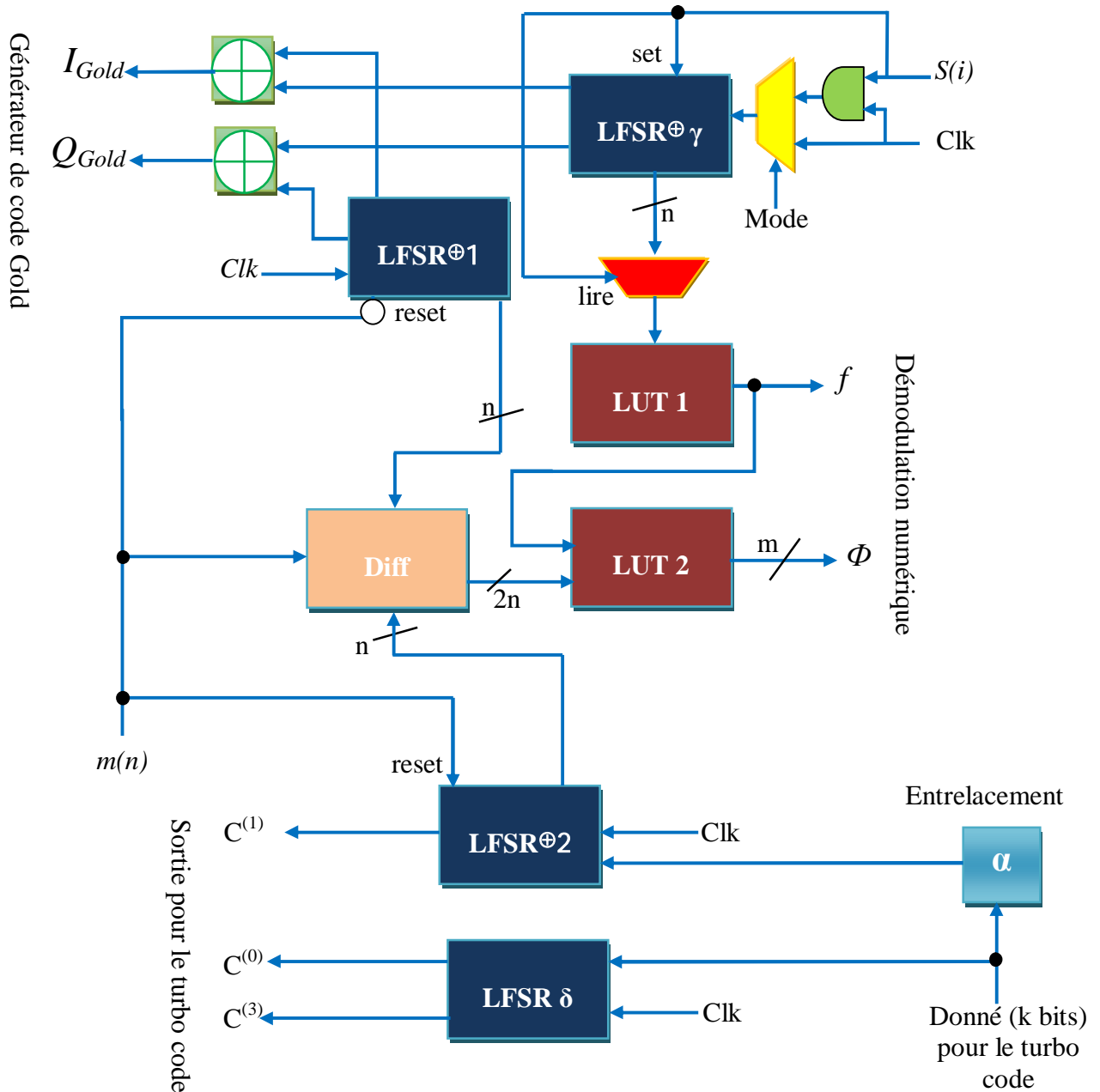


Figure. IV.14. Architecture générique proposée [2].

Dans cette architecture, le signal *mode* (figure IV.14) est un paramètre booléen (mis en œuvre par le biais d'un de bit de sélection), il permet de configurer l'architecture en utilisant un multiplexeur de configuration (couleur jaune dans la figure IV.14).

Dans la section suivante, nous proposons une structure d'un registre LFSR reconfigurable pour l'architecture générique (figure IV.14). Ce registre reconfigurable peut être considéré comme une architecture commune pour la démodulation numérique et le codage.

4.5.1. Un registre LFSR reconfigurable comme un opérateur commun

L'architecture du registre LFSR utilisée se diffère en fonction de l'application et de la fonctionnalité, par exemple le générateur de code Gold utilise deux registres où chaque registre possède une largeur (taille) de 18 bits et les deux se diffèrent dans les polynômes générateurs. La longueur du registre utilisée dans le cas de code Gold n'est pas forcément la longueur qui doit être utilisée pour les LFSR du codeur convolutif ou le démodulateur numérique. En effet, cette différence dans l'architecture de LFSR peut se résumer comme suit:

- le nombre de cellules du registre à décalage ou la longueur de ce registre,
- la différence dans le degré du polynôme générateur,
- la différence dans les polynômes générateurs (les coefficients g_i et h_i),

Par conséquent, nous avons proposé un registre (figure IV.15), reconfigurable considéré comme un opérateur commun pour plusieurs fonctions [2]. Nous remarquons que les registres dans notre architecture proposée (figure IV.14) se retrouvent dans l'architecture du LFSR reconfigurable (figure IV.15) [2].

L'architecture proposée (figure IV.15) se compose de " n " flip-flop D, des portes logiques, des coefficients (g_i/h_i) et des paramètres booléens " S_i " (mis en œuvre par le biais d'un bit de sélection). Ce registre peut s'adapter facilement à la nouvelle application par une simple configuration de ces paramètres " S_i " et ces coefficients (g_i/h_i) [2]. L'entrée d'horloge de chaque bascule « r_i » est le résultat de la multiplication du signal *CLK* (l'horloge de FPGA) et un paramètre booléen " S_i ", où i représente l'ordre de la bascule considérée dans le registre. Donc si " S_i " est égal à '0' la bascule « r_i » ne fonctionne plus (désactivée), et si " S_i " est égal à '1' la bascule « r_i » est active (elle fonctionne). Ceci permet de changer le degré du polynôme du registre LFSR ainsi que leur structure interne. De plus, les coefficients (g_i/h_i) dans le registre universel sont modifiables et reconfigurables en fonction de l'application. Le changement des coefficients (g_i/h_i) permet de changer le polynôme générateur de ce registre. De cette façon, notre registre proposé peut être adapté au changement de la fonction par la paramétrisation des coefficients (g_i/h_i) et les paramètres " S_i ".

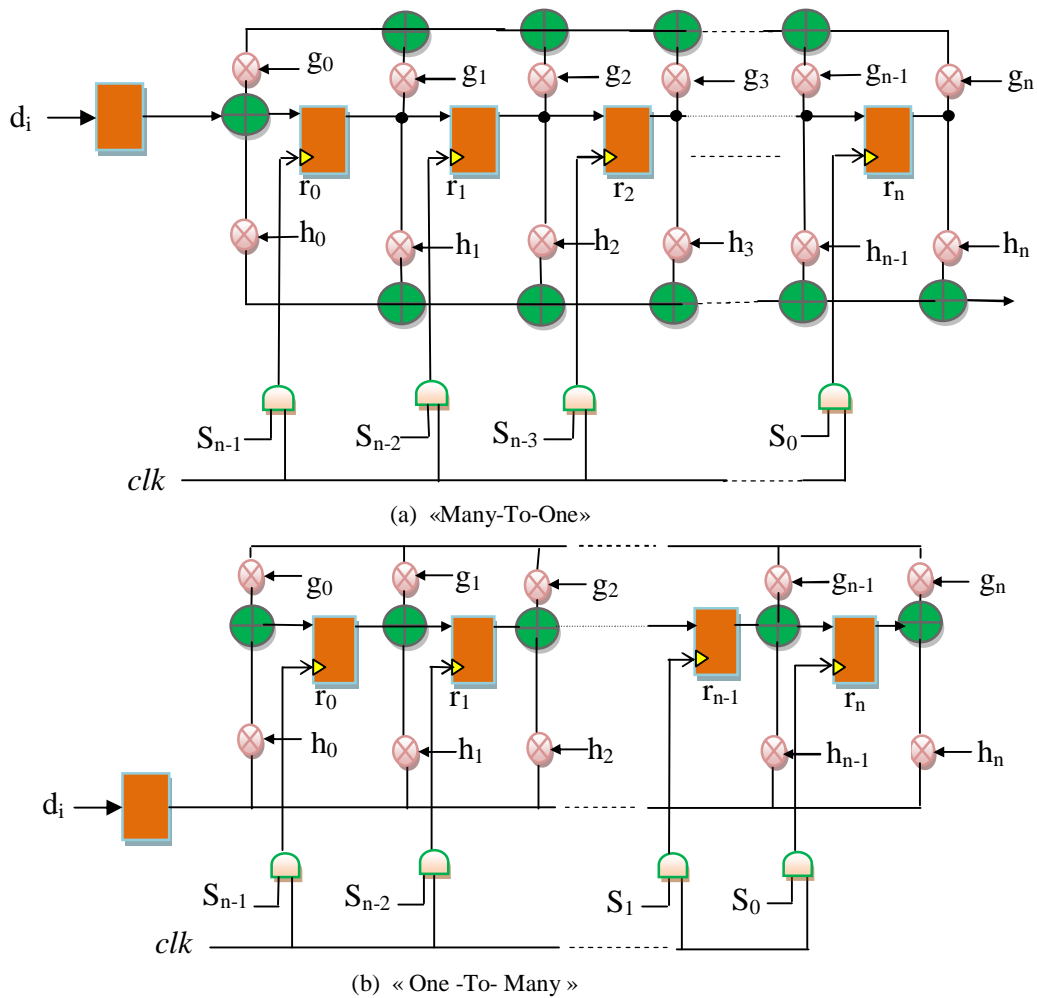


Figure. IV.15. Les deux architectures proposées LFSR configurables [2].

La configuration de ce registre reconfigurable est faite par un autre registre externe (registre paramètre de reconfiguration dans la figure IV.16) qui contient les valeurs des paramètres " S_i " et les coefficients (g_i/h_i) (figure IV.16).

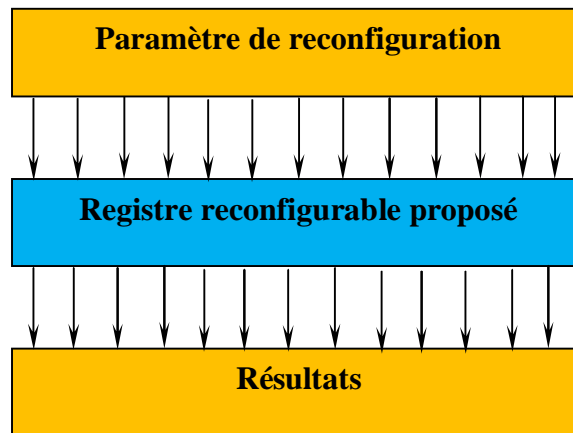


Figure. IV.16. Technique de reconfiguration du registre LFSR proposé.

Ainsi, la sortie de chaque bascule de ce registre reconfigurable est reliée avec un autre registre 'Résultats' (figure IV.16). Ce registre (Résultats) contient le résultat du registre reconfigurable proposé.

4.5.2. Fonctionnement de l'architecture flexible

L'architecture proposée (figure IV.14) peut basculer entre deux modes qui sont le mode démodulation numérique et le mode codage. L'architecture peut prendre en charge l'un des deux modes, mais pas les deux modes en même temps; c'est-à-dire lorsque l'architecture est en mode démodulation, elle ne peut pas fonctionner en mode codage et vice versa.

4.5.2.1. Mode démodulation numérique

Dans ce mode, l'architecture peut fonctionner comme un démodulateur de phase et un démodulateur de fréquence.

- **Mode: démodulation de fréquence FSK**

Dans ce mode, le registre (figure IV.14) nommé $LFSR^{\oplus\gamma}$ est utilisé pour la démodulation de fréquence de ce signal. La première étape est la conversion du signal échantillonné et quantifié en un signal purement binaire. Cette première étape est indispensable avant de commencer la démodulation, puisque les LFSR fonctionnent avec des signaux binaires seulement. Il faut noter aussi que ce signal converti doit présenter la même fréquence et la même phase que le signal original, dont l'amplitude ne peut prendre d'autres valeurs que 0 ou 1 (voir annexe C). Le signal à démoduler et converti en forme binaire est noté $S(i)$. Ce signal représente une des deux entrées d'un multiplexeur (couleur jaune dans la figure IV.14). Ce multiplexeur a une sortie et deux entrées: l'horloge (Clk) et le signal $S(i)$ multiplié par l'horloge ($S(i) \times Clk$). Le signal de contrôle de ce multiplexeur est le signal *mode*. Ceci permet de sélectionner selon le mode de fonctionnement soit le signal à démoduler en fréquence $S(i)$ multiplié par l'horloge (clk) soit seulement l'horloge (Clk) [2]. Par exemple, dans ce mode le signal *mode* permet pour le signal ($S(i) \times Clk$) de passer dans le registre $LFSR^{\oplus\gamma}$.

Le polynôme générateur du registre $LFSR^{\oplus\gamma}$ est de degré n , il pourra passer par (2^n-1) états différents. Le signal de comptage qui anime le $LFSR^{\oplus\gamma}$ est le signal à démoduler ($S(i) \times Clk$), donc le $LFSR^{\oplus\gamma}$ est incrémenté sur les fronts montants du signal ($S(i) \times Clk$). De plus, on a utilisé le signal à démoduler $S(i)$ comme un signal « reset » pour ce registre ($LFSR^{\oplus\gamma}$).

Si un front descendant (l'état logique «0») est détecté du signal $S(i)$, le contenu de ce registre est lu par un autre registre de données (couleur rouge dans la figure IV.14) juste à l'instant du déclenchement de ce signal « reset ». Le registre de données (couleur rouge

dans la figure IV.14), stocke la valeur de l'état interne du registre $LFSR^{\oplus\gamma}$ à chaque incrémentation. En même temps, le registre $LFSR^{\oplus\gamma}$ est remis à sa valeur initiale par ce signal *reset* ($S(i)$).

Enfin, la valeur lue est utilisée pour adresser une *LUT* (LUT 1) qui contient une table de correspondance entre l'ensemble des valeurs possibles pour le $LFSR^{\oplus\gamma}$ et la fréquence associée (f)[2][109]. Cette fréquence mesurée, représente le symbole qui a été transmis par la partie modulation.

- **Mode: Démodulation de phase PSK**

Dans ce mode, les deux registres notés $LFSR^{\oplus 1}$ et $LFSR^{\oplus 2}$, fonctionnant en mode compteur peuvent mesurer la phase de ce signal. D'abord, comme pour la modulation de fréquence, le signal d'entrée est converti en une forme purement binaire, mais conserve la même fréquence et la même phase que le signal original (voir annexe C). Le signal M-PSK après une conversion en binaire est noté $m(n)$ (figure IV.14). Le principe de la démodulation de phase est basé sur la mesure de la *régularité* du signal $m(n)$ [93]. En d'autres termes, un changement de phase sur un signal binaire carré se traduit par une rupture de la régularité des périodes pendant lesquelles le signal est à l'état haut et à l'état bas. Dans le cas d'un signal ne subissant aucun déphasage, la durée de ces périodes doit être identique (figure IV.17) [93]. Sur le chronogramme (IV.17a), $T^+ = T^-$ le signal conserve la même phase sur toute la durée du graphe. Par contre, sur la figure (IV.17b), $T'_+ < T^-$ ce qui dénote un déphasage $\Delta\theta$. Celui-ci peut-être évalué de la façon suivante [93]:

$$\Delta\theta = 2\pi f(T'_+ - T^-) \quad (15)$$

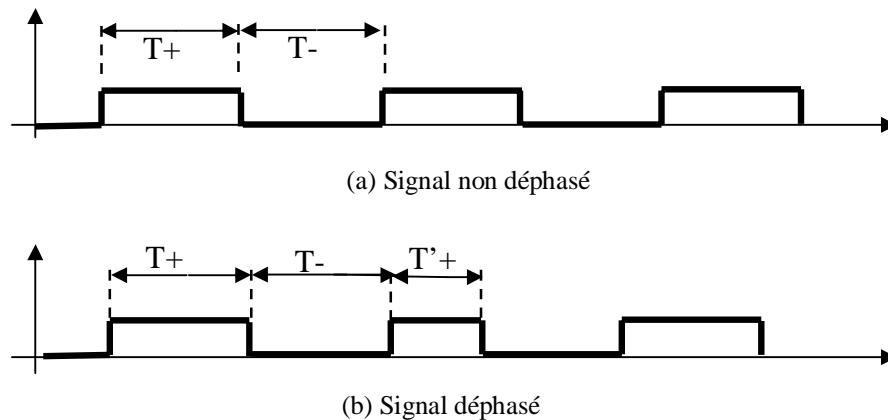


Fig. IV.17: Durées des états haut et bas pour un signal non déphasé, et un signal avec un déphasage.

Par conséquent, on a utilisé deux registres: le registre $LFSR^{\oplus 1}$ permet de mesurer la durée pendant laquelle le signal $m(n)$ reste à l'état haut et le $LFSR^{\oplus 2}$ permet de mesurer la durée pendant laquelle le signal $m(n)$ reste à l'état bas [2]. Les deux registres ($LFSR^{\oplus 1}$ et $LFSR^{\oplus 2}$) fonctionnent comme deux compteurs synchrones incrémentés sur le front montant de l'horloge clk . Pour mesurer la durée de l'état haut et l'état bas, on a utilisé le signal à démoduler $m(n)$ comme un signal « reset » pour les deux registres de cette façon:

- si un front montant du signal $m(n)$ (état logique «1») est détecté, le $LFSR^{\oplus 1}$ commence l'incrément, après détection d'un front descendant de $m(n)$ le contenu de ce registre est lu par le bloc « *Diff* », et ce registre est remis à sa valeur initiale par le signal reset $m(n)$ (figure IV.14).
- si un front descendant du signal $m(n)$ (état logique «0») est détecté, le $LFSR^{\oplus 2}$ commence l'incrément, après détection d'un front montant de $m(n)$ le contenu de ce registre est lu par le bloc « *Diff* », et ce registre est remis à sa valeur initiale par le signal reset $m(n)$ (figure IV.14).

Le module « *Diff* », permet de calculer la différence des résultats des deux compteurs pour le calcul de déphasage du signal $m(n)$. Ce déphasage est calculé à l'aide de l'équation (15), et la conversion finale est effectuée par une LUT (LUT 2) contenant la table de conversion entre les couples de valeurs (R_i, f) et la phase θ . Pour la fréquence qui entre dans l'équation (15), nous supposons qu'elle est évaluée en utilisant une autre architecture de démodulation de fréquence comme celle présentée dans la section suivante. Enfin, la phase mesurée représente le symbole qui a été transmis par la partie modulation.

Nous pouvons utiliser deux LUTs qui contiennent les valeurs des symboles correspondants, par exemple la première LUT est adressée par la valeur de la fréquence et la deuxième LUT est adressée par la valeur de la phase.

4.5.2.2. Mode codage

Dans ce mode l'architecture peut fonctionner comme un générateur de code Gold et un turbo code de type parallèle.

- **Mode générateur de code de Gold**

Le générateur de code Gold nécessite deux registres LFSR (section 3.3), donc nous avons utilisé deux registres configurables nommée $LFSR^{\oplus \gamma}$ et $LFSR^{\oplus 1}$. Ainsi, nous devons adapter les coefficients h_i, g_i et les paramètres S_i pour la génération de code de Gold, car les deux registres ont une longueur de 18 bits, et les deux polynômes utilisés (deux séquences m) sont décrits par les équations A.12 et A.13 respectivement (voir annexe A). Donc, le premier LFSR noté $LFSR^{\oplus \gamma}$ est de polynôme générateur: $1 + X^7 + X^{18}$, et le

deuxième LFSR noté $LFSR^{\oplus 1}$ est de polynôme générateur: $1 + X^5 + X^7 + X^{10} + X^{18}$. Le signal de comptage qui anime les deux registres ($LFSR^{\oplus \gamma}$ et $LFSR^{\oplus 1}$) est l'horloge (CLK). Par conséquent, le signal de commande $Mode$ permet de sélectionner le signal CLK comme une sortie du multiplexeur (couleur jaune dans la figure IV.14). Enfin, nous avons besoin de deux portes XOR (figure IV.14) pour la génération de la partie réelle et imaginaire du code Gold [2][9].

- **Mode codage de canal.**

L'architecture universelle proposée peut prendre en charge la fonction d'un turbo code de type parallèle en utilisant deux registres LFSR noté $LFSR^{\oplus \delta}$ et $LFSR^{\oplus 2}$ (figure IV.14). À ce propos, l'architecture des deux registres doit être modifiée conformément aux architectures d'un codeur convolutif récursif et systématique (RSC) (section 4.2). Ce changement est facilement obtenu car tous les LFSR dans l'architecture proposée sont des registres reconfigurables (figure IV.15).

Nous pouvons donc changer le degré du polynôme du registre (ou la taille de la mémoire du codeur), soit par exemple les paramètres S_i comme $(S_{n-1} S_{n-2} S_{n-3} \dots S_0) = (11100 \dots 0)$ pour obtenir un codeur convolutif de taille (de mémoire) $m = 3$. Les coefficients (g_i) et (h_i) dans le registre configurable sont modifiables et reconfigurables en fonction de l'application [2]. En plus, l'entrée pour le deuxième registre est entrelacée par le bloc d'entrelacement.

5. Implémentation sur FPGA.

La description RTL des architectures proposées dans ce chapitre a été implémentée en utilisant la description structurale VHDL (VHSIC Hardware Description Language). Le FPGA ciblé pour implémentation est la famille Virtex-V de Xilinx [94]. L'outil ISE 10.1i de Xilinx [51] a été utilisé pour cette implémentation numérique permettant d'obtenir les ressources logiques nécessaires et les performances associées.

5.1. Implémentation de l'architecture générique proposée pour la modulation numérique

Dans cette section, nous présentons les résultats de la synthèse et la simulation de l'architecture numérique générique pour la modulation M-QAM, M-PSK et B-FSK. L'architecture matérielle proposée de cette architecture générique est écrite en langage VHDL, en utilisant l'outil de conception Xilinx ISE 10.1i pour la synthèse. La conception RTL pour le codeur, la LUT de phase et la LUT d'amplitude, le processor Cordic et l'additionneur pipeline sont montrés dans les figures IV.18, IV.19, IV.20, IV.21 et IV.22 respectivement. Le tableau IV.1 montre les résultats de synthèse de l'ensemble du système sur des données de 8 bits.

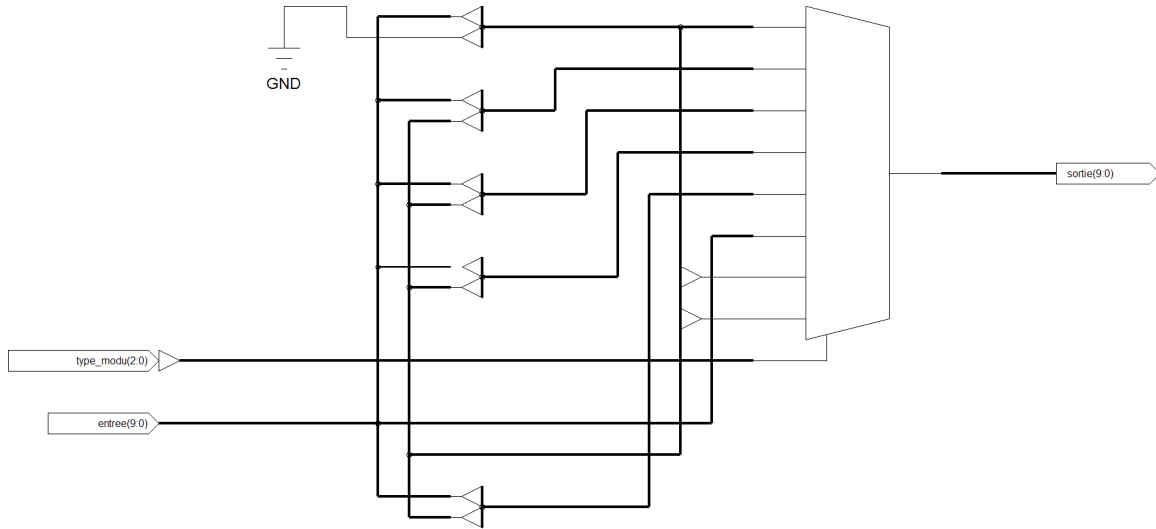


Figure IV.18. Schémas RTL de l'implémentation sur FPGA Virtex 5 du codeur.

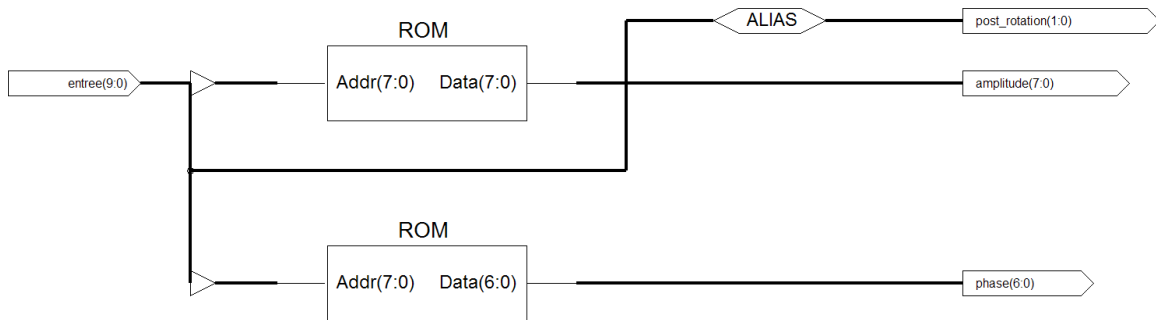


Figure IV.19. Schémas RTL de l'implémentation sur FPGA Virtex 5 de LUT de phase et la LUT d'amplitude.

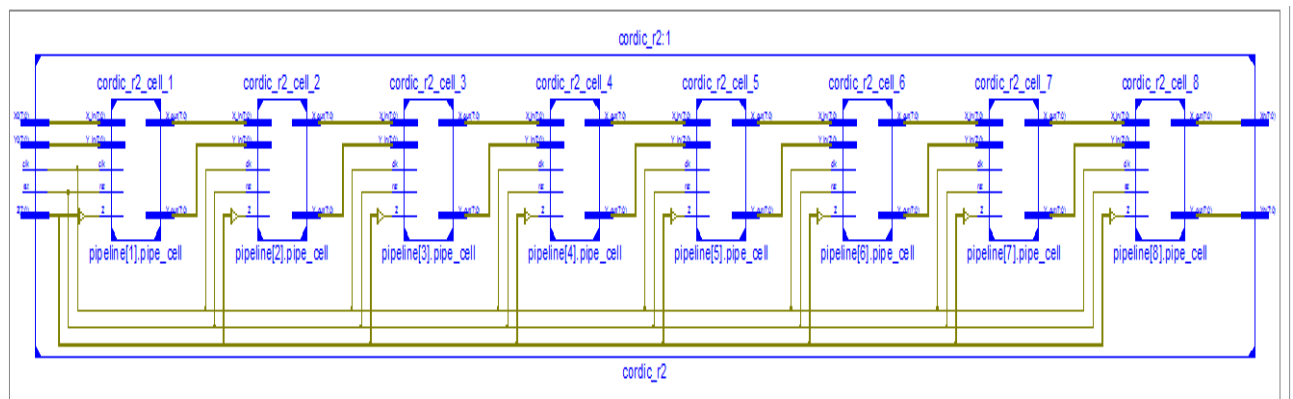


Figure IV.20. Schémas RTL de l'implémentation sur FPGA Virtex V du CORDIC de 8 étages.

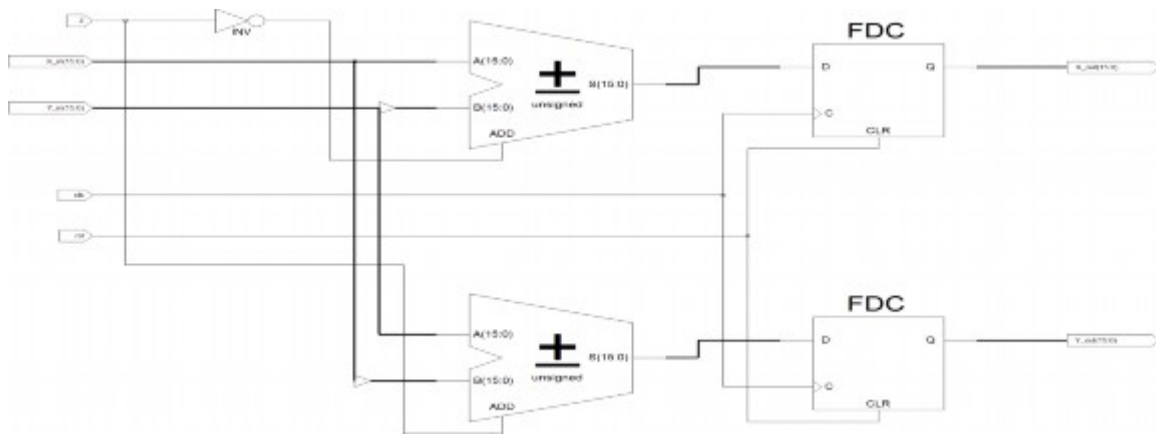


Figure IV.21. Schémas RTL d'un étage d'algorithme de CORDIC proposé par [93].

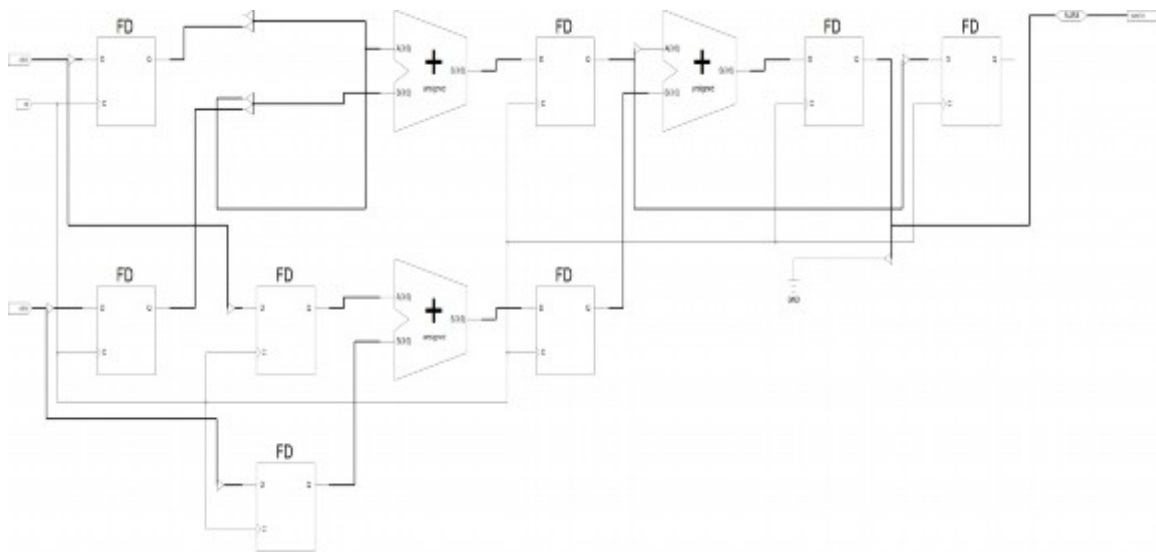


Figure IV.22. Schémas RTL de l'additionneur pipeline.

Tableau IV.1. Résultats de synthèse de l'architecture proposée

	Slices registres (19200)	LUT (19200)	Fréquence de fonctionnement (MHz)
Codeur	----	12	-----
LUT de phase et LUT d'amplitude	----	15	-----
CORDIC	128	136	602.518
Additionneur pipeline	29	13	645.953

Nous pouvons déduire de ce tableau que l'architecture proposée possède une faible consommation de ressource avec des données de 8 bits. En effet, chaque bloc de cette architecture utilise un très faible pourcentage (%) des éléments logiques disponibles dans le circuit ciblé, conduisant à un circuit de très petite dimension. On peut voir même avec le processeur Cordic, que nous avons une faible consommation, ceci revient à l'utilisation du Cordic optimisé proposé par [93]. Cette très faible consommation en ressource de cette architecture donne la possibilité de faire la parallélisation pour augmenter la fréquence de fonctionnement et la fréquence d'échantillonnage apparente ou d'implanter d'autres architectures en utilisant le reste des ressources.

- **Résultats de simulation**

Pour la vérification de nos implémentations, nous avons utilisé ISE 10 pour la simulation des résultats. Le but de cette simulation consiste à vérifier le fonctionnement du bloc additionneur pipeline proposé et le processeur Cordic car leur vitesse de fonctionnement est déterminante dans l'architecture proposée. Un fichier testbench a été écrit spécialement pour cette simulation. Les figures IV.23 et IV.24 montrent les résultats de simulation de l'implémentation proposée.

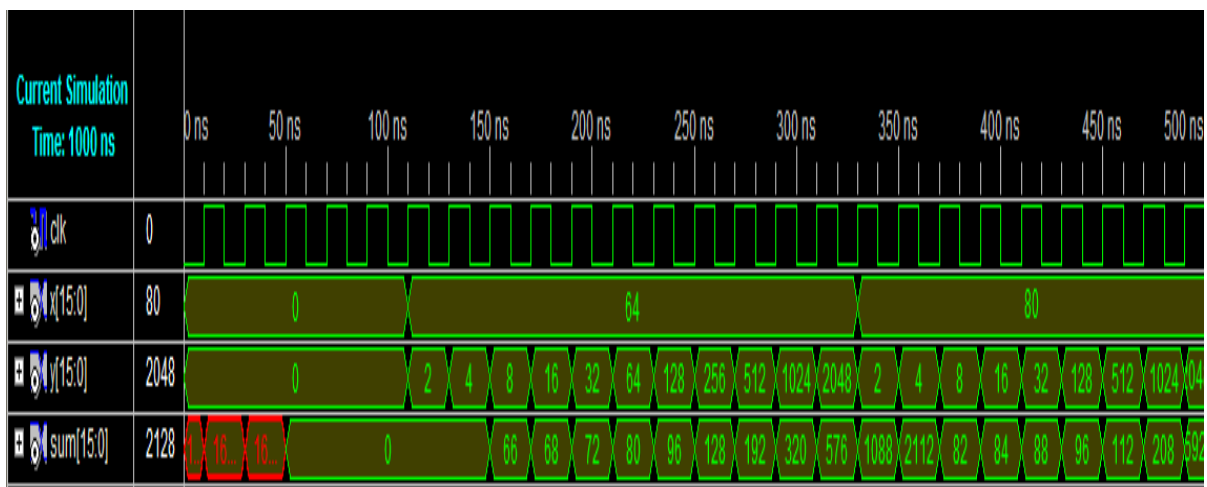


Figure IV.23. Résultats de simulation de l'additionneur pipeline.

Sur le chronogramme de la figure IV.23,

- Le signal «X» représente la phase issue de la LUT de phase,
- Le signal «Y» sont les N sorties de l'accumulateur de phase, elles assurent la description discrète du signal sinusoïdal désiré,
- Le signal «sum» représente la sortie de l'additionneur pipeline.

On remarque que le résultat de l'addition est disponible après 2 cycles de retards, ce qui est clair car nous avons dans le premier cycle une opération d'addition des MSB pour les

deux signaux d'entrée et une opération d'addition des LSB pour les deux signaux d'entrée, puis dans le deuxième cycle nous avons l'opération finale (figure IV.6).

Pour le Cordic sur le chronogramme de la figure IV.24 on a :

- le signal « X » représente l'amplitude du signal à générer,
- le signal « Z » représente la phase du signal à générer,
- le signal « Xn et Yn » représentent la sortie de Cordic, où exactement le signal « Xn » représente le cosinus de la phase et « Yn » représente le sinus de la phase.

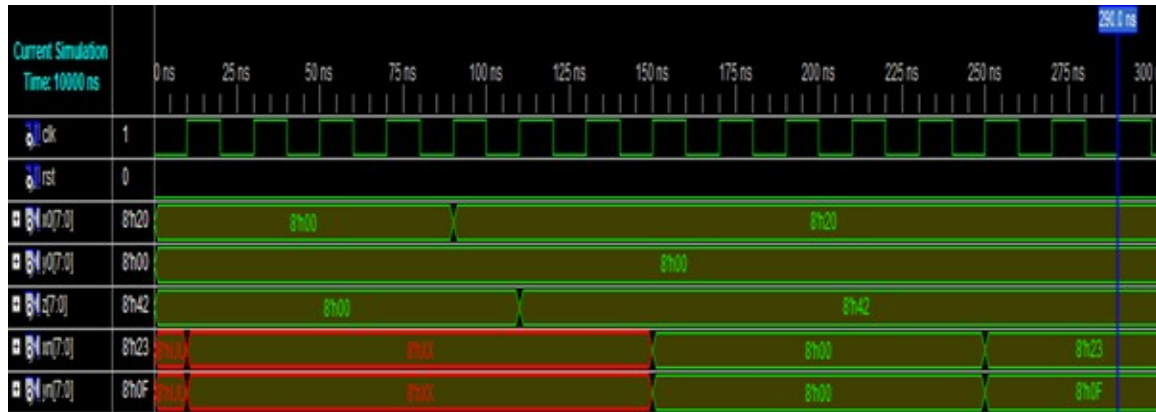


Figure IV.24. Résultats de simulation du Cordic.

Sur le chronogramme de la figure IV.24 nous avons montré le calcul d'un seul angle d'un seul échantillon (de valeur 42 en hexa), on remarque que le résultat est disponible après 8 cycles d'horloge. Ceci s'explique par le fait que nous avons utilisé un Cordic de 8 étages. Ce résultat montre bien l'intérêt de paralléliser plusieurs processeurs Cordic afin de générer plusieurs échantillons d'une même sinusoïde en parallèle. On peut par exemple paralléliser 8 processeurs Cordic pour obtenir 8 échantillons en 8 cycles d'horloge au lieu d'un seul échantillon en 8 cycles d'horloge.

5.2.Implémentation de l'architecture générique pour le codage et la démodulation numérique.

Dans cette section, nous présentons les résultats de la synthèse et la simulation de l'architecture universelle proposée pour le codage canal, le code de brouillage et la démodulation numérique. De plus, pour une évaluation réelle de la performance et la consommation, nous comparons les implémentations de notre architecture générique avec la technique Velcro.

L'architecture de Velcro utilisée pour la comparaison est composée de quatre architectures différentes :

- les architectures de démodulation de phase [93] de quatre registres,

- les architectures de démodulation de la fréquence [93][109] de deux registres,
- le générateur de code de Gold avec deux registres,
- le Turbo code avec deux registres.

Toutes les architectures sont codées en VHDL, simulées et synthétisées en utilisant le ModelSim SE v6.5 et l'outil Xilinx ISE 10. La figure IV.25 présente le schéma RTL pour notre architecture proposée.

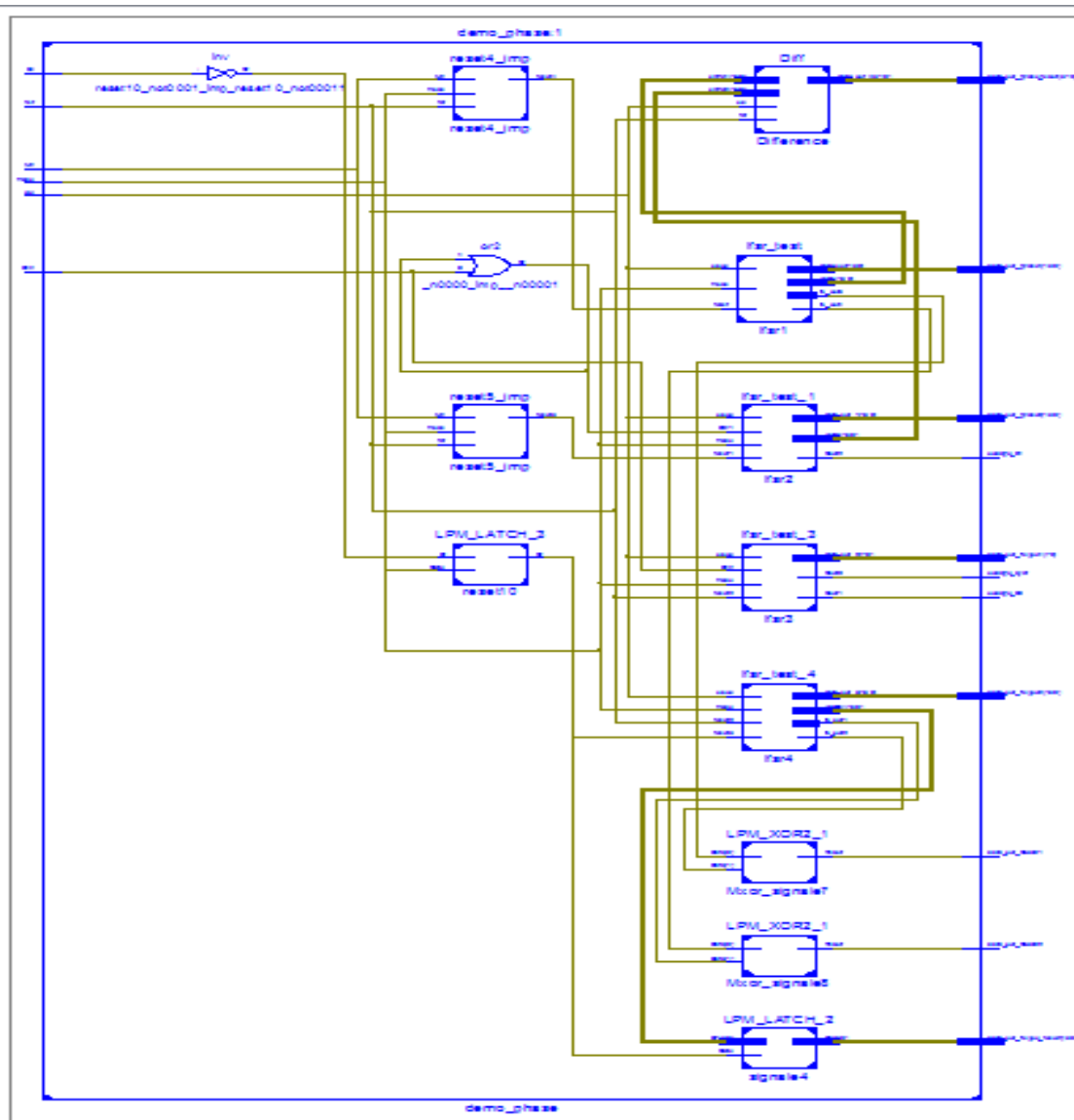


Figure. IV.25. Schémas RTL de l'implémentation sur FPGA Virtex V de l'architecture proposée [2].

Les résultats de l'implémentation après la synthèse par l'outil ISE.10 de Xilinx sont donnés dans le tableau IV.4 pour différentes tailles (longueur de LFSR) en termes de

surface occupée (le nombre de Slice LUTs (Look Up Tables) et le nombre de slice registres), la fréquence de fonctionnement maximal et la consommation de puissance.

Les polynômes générateurs pour les codes Convolutifs utilisés pour cette simulation sont donnés dans le tableau IV.2 et le tableau IV.3. Leur choix est justifié par le fait qu'ils présentent la meilleure distance libre pour une taille de mémoire donnée [118].

Notations: Chaque codeur est défini par la notation $(k, n, m) (G, H_1, \dots, H_i)_8$; où:

- k représente le nombre d'entrées du codeur,
- n représente le nombre de sorties,
- M est la taille de la mémoire du codeur,
- G représente les coefficients du polynôme de la boucle récursive,
- H_1, \dots, H_i sont les i polynômes correspondant aux i sorties codées,
- $()_8$ signifie que les polynômes générateurs sont donnés en octal.

Tableau. IV.2. Polynômes générateurs de codes Convolutifs.

R=1/2
(1,2,2)(7,5) ₈
(1,2,4)(23,33) ₈

Tableau. IV.3. Polynômes générateurs de codes Convolutifs (R=1/2)

R=1/2		
Notation	Degré	Pôlynomes de la boucle récursif
G(X)	16	$1 + X^2 + X^{15} + X^{16}$
Notation	Degré	Pôlynomes de la branche directe
H(X)	16	$1 + X^1 + X^2 + X^4 + X^7 + X^9 + X^{10} + X^{14} + X^{16}$

Dans le tableau IV.4, on observe que la performance de l'architecture proposée est meilleure que celle de l'architecture Velcro. L'occupation de surface est réduite de 40% à 50% comparée à l'architecture Velcro. Notre architecture proposée pour 4 bits prend seulement 65 « slice registers » et atteint une fréquence de 644 MHz, en fonctionnement maximum avec la technologie Virtex 5.

Notre architecture comparée à l'architecture Velcro, offre un gain de LUT entre 11,32% et 35,07%. La consommation de surface de notre architecture progresse linéairement avec la précision en bit, elle présente toujours une consommation raisonnable qui permet d'envisager une parallélisation éventuelle en multipliant les étages de calcul. Nous

pouvons considérer aussi qu'il reste suffisamment d'espace afin d'implémenter des algorithmes additionnels.

Tableau. IV.4. Comparaison de l'architecture proposée avec l'architecture Velcro.

Xilinx Virtex 5						
	Résolution					
	Longueur du LFSR (bits) de 4 bits pour la démodulation PSK et FSK Turbo code (1,2,2)(7,5)8 LFSR pour Gold de 18 bits		Longueur du LFSR (bits) de 8 bits pour la démodulation PSK et FSK Turbo code (1,2,4)(23,33)8 LFSR pour Gold de 18 bits		Longueur du LFSR (bits) de 16 bits pour la démodulation PSK et FSK Turbo code de Polynômes générateurs du tableau IV.3 LFSR pour Gold de 18 bits	
Paramètres	Architecture proposée	Velcro	Architecture proposée	Velcro	Architecture proposée	Velcro
Number of Slice Registers (19200)	65	103	93	171	138	312
Number of Slice LUTs (19200)	47	53	59	79	87	134
Maximum Frequency (MHz)	644.300	Démodulateur de phase [90] : 1084.952 Démodulateur de fréquence [122]: 1025.010 Gold code: ----- Code convolutif 1025.010	644.300	Démodulateur de phase [90] : 911.826 Démodulateur de fréquence [122]: 1080.030 Gold code: ----- Code convolutif 1080.030	644.300	Démodulateur de phase [90] : 911.826 Démodulateur de fréquence [122]: 1080.030 Gold code: 644.330 Code convolutif 1080.030
Power (mW) Quiescent: Dynamic: Total :	379.12 2.38 381.50	379.10 1.43 380.53	379.11 1.90 381.01	379.14 4.05 383.19	379.14 4.28 383.43	379.21 9.28 388.49

On remarque aussi que la fréquence maximale de l'architecture proposée est inférieure à une architecture matérielle dédiée (le tableau IV.4), mais le grand avantage est la flexibilité offerte qui augmente le parallélisme du traitement en partageant des ressources communes entre les algorithmes, ce qui conduit à améliorer la vitesse de calcul [1], [71].

La puissance consommée par l'architecture a été estimée par l'outil de Xilinx «XPower analyzer», après le processus «place and route». Le tableau montre les rapports détaillés de la puissance, dans Xilinx Virtex5. La comparaison de notre architecture proposée avec l'architecture Velcro, indique que la notre réalise une consommation de puissance appropriée, notamment lorsque la taille des registres augmente (tableau IV.4).

- **Résultats de simulation**

Pour la vérification de nos implémentations, nous avons utilisé ISE.10 pour la simulation des résultats. Le but de cette simulation consiste à vérifier le fonctionnement global de l'architecture proposée. Un fichier testbench est écrit spécialement pour cette simulation. Les figures (IV.26) et (IV.27) montrent les résultats de simulation de l'implémentation proposée respectivement dans le mode démodulation et le mode codage.

Dans le mode démodulation, l'architecture peut mesurer la fréquence et la phase d'un signal modulé. Dans cette configuration, le signal « Mode » qui permet de configurer l'architecture, est égal à '1'. Sur le chronogramme de la figure (IV.26), le signal PSK à démoduler est noté « Mn » après conversion en binaire. Le signal FSK à démoduler est noté « Si » après conversion en binaire.

- Data_out_phase01 est la sortie du premier LFSR pour le démodulateur de phase.
- Data_out_phase02 est la sortie du deuxième LFSR pour le démodulateur de phase.
- Data_out_freque02 est la sortie du LFSR pour le démodulateur de fréquence.
- Data_out_freque est la valeur sortie du deuxième LFSR pour le démodulateur de fréquence.

Dans le cas de la démodulation de fréquence, on remarque sur le chronogramme que le registre $LFSR^{\oplus\gamma}$ est incrémenté sur les fronts montants du signal ($Si \times clk$). Lorsqu'un front descendant du signal Si est détecté, on lit la valeur d'état du registre $LFSR^{\oplus\gamma}$. Dans cet exemple, cette valeur représente la sixième itération du registre $LFSR^{\oplus\gamma}$ (de valeur 0800 en Hexa sur le chronogramme). La LUT de la figure (IV.14) est ainsi adressée avec cette valeur. En effet, le registre $LFSR^{\oplus\gamma}$ fait six itérations, c'est-à-dire six périodes d'horloge ($6 T_{clk}$) pendant une demi-période du signal FSK ($\frac{T_{signal}}{2}$ ou $\frac{1}{2f_q}$) après la

conversion binaire (voir annexe C). Par conséquent, nous avons $(6 \cdot T_{clk} \rightarrow \frac{T_{sigantl}}{2})$, donc la fréquence mesurée résultante fournie par cette LUT est $f_q = \frac{1}{12} f_{clk}$.

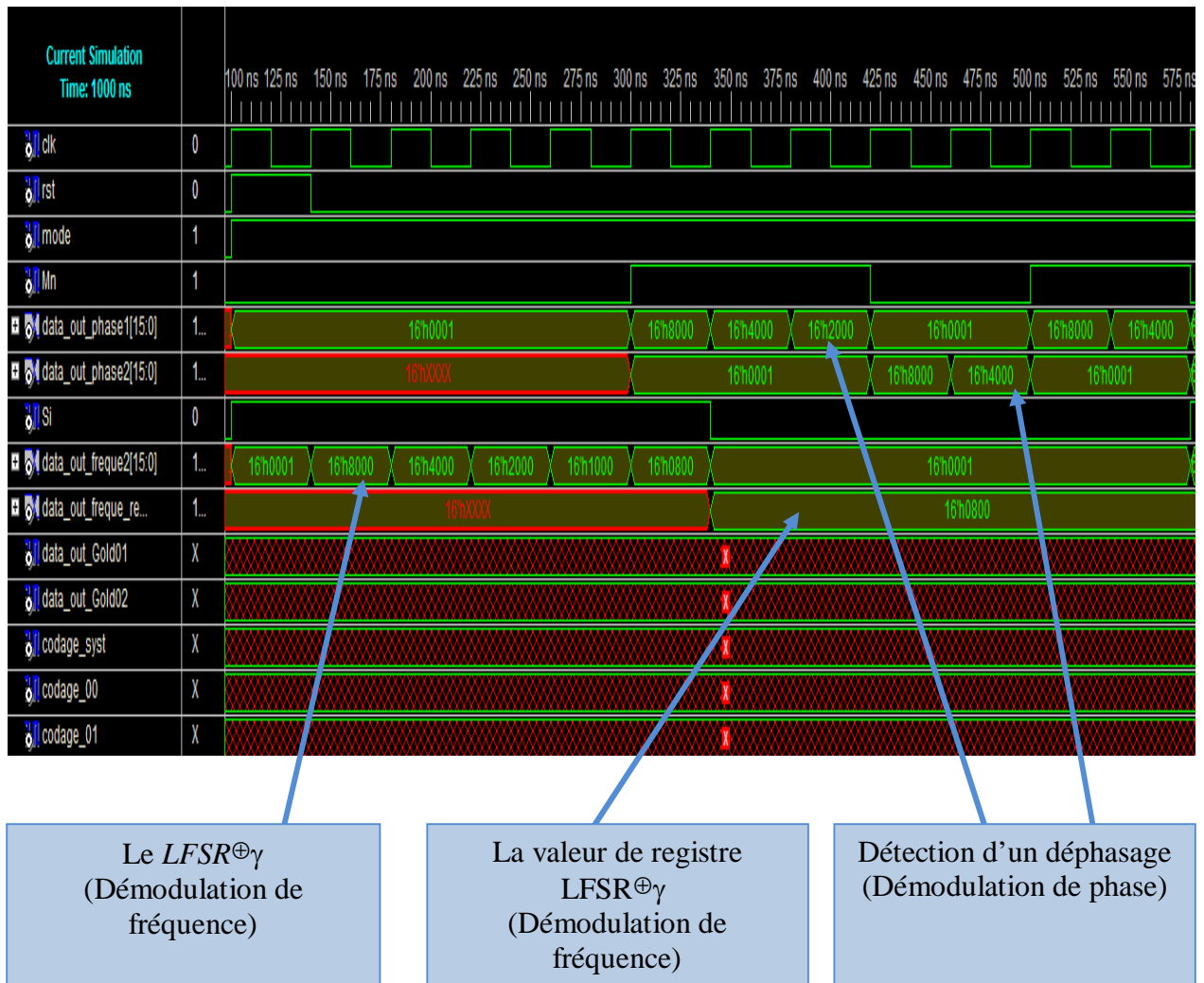


Figure IV.26. Résultats de simulation dans le mode démodulation.

Dans le cas de la démodulation de phase, on remarque sur le chronogramme qu'il y a un déphasage détecté par les deux registres, où le nombre d'itérations du registre $LFSR^{\oplus 1}$ (qui permet de mesurer la durée pendant laquelle le signal $m(n)$ reste à l'état haut) et le nombre d'itérations du registre $LFSR^{\oplus 2}$ (qui permet de mesurer la durée pendant laquelle le signal $m(n)$ reste à l'état bas) ne sont pas les mêmes. Le nombre d'itérations du registre $LFSR^{\oplus 1}$ représente le nombre de cycles d'horloge pendant lesquelles le signal $m(n)$ reste à l'état haut, et le nombre d'itérations du registre $LFSR^{\oplus 2}$ représente le nombre de cycles d'horloge pendant lesquelles le signal $m(n)$ reste à l'état bas. Donc, ceci permet de mesurer la valeur de la phase en utilisant l'équation (15) et la valeur de la fréquence mesurée par le registre $LFSR^{\oplus \gamma}$.

Dans le mode codage, l'architecture peut fonctionner comme un générateur de code Gold et comme un turbo code pour le codage canal. Dans cette configuration, le signal « Mode » prend maintenant la valeur '0' (figure IV.27).

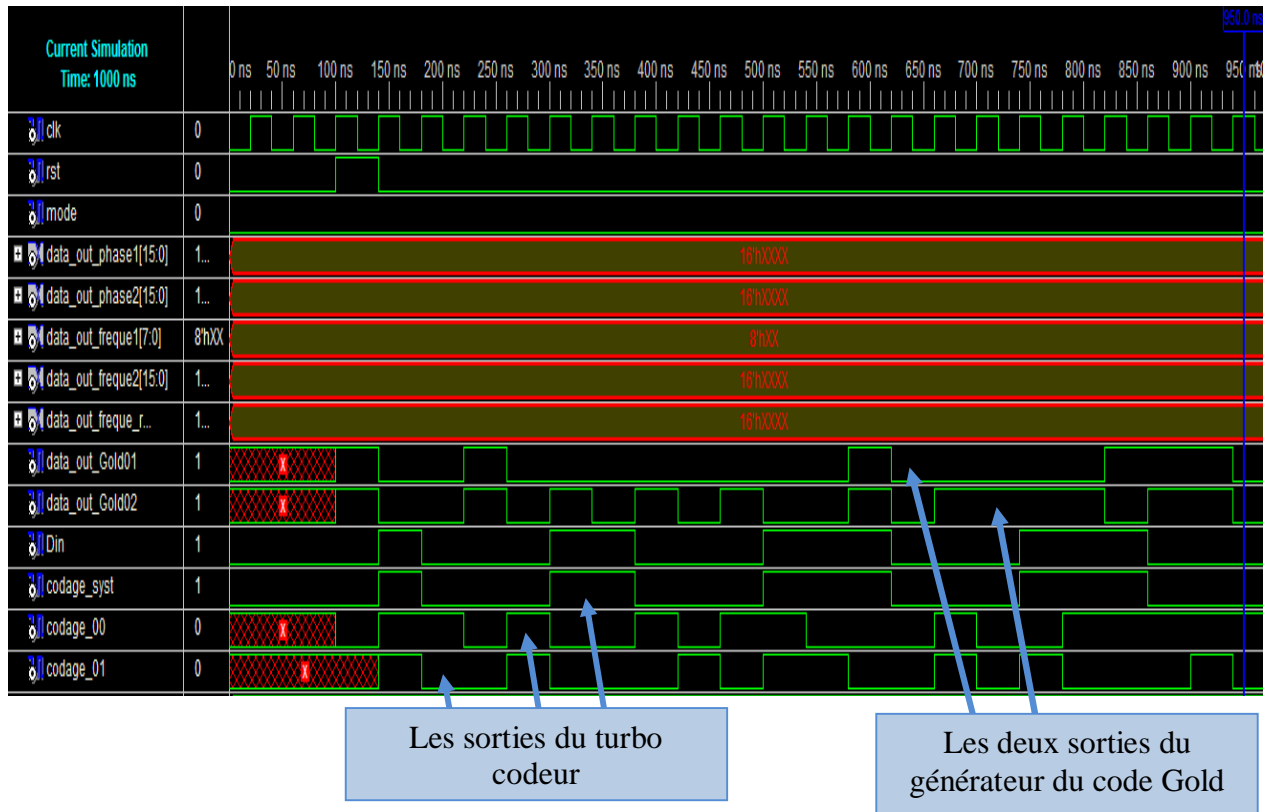


Figure IV.27. Résultats de simulation dans le mode codage.

Le chronogramme de la figure IV.27 montre :

- le signal *Din* qui est le flux de données d'entrée,
- le *Codage_Sys* qui est le flux de sortie systématique,
- le *Codage_00* qui est le flux de sortie codé par le premier codeur RSC,
- le *Codage_01* qui est le flux de sortie codé par le deuxième codeur RSC après entrelacement du flux de données d'entrée (*Din*),
- le *Data_out_Gold01* qui est la sortie du premier LFSR dans le générateur de code Gold,
- le *Data_out_Gold02* qui est la sortie du deuxième LFSR dans le générateur de code Gold.

Les polynômes générateurs pour le code convolutif utilisés pour la réalisation d'un Turbo code sont les polynômes générateurs du tableau (IV.3).

On remarque sur le chronogramme de la figure (IV.27) que pour le turbo code, la sortie systématique est similaire au flux de données d'entrée, car dans un code systématique la sortie doit contenir les bits d'entrée. On remarque aussi, que la sortie du premier codeur est différente de celle du deuxième codeur qui utilise les données d'entrée après entrelacement.

Pour le générateur de code Gold, nous avons deux sorties (représentant les sorties des deux portes XOR) qui génèrent la partie réelle et la partie imaginaire.

6. Conclusion

Dans ce chapitre, nous avons proposé deux architectures numériques et universelles :

- la première architecture est pour la modulation M-QAM, M-PSK et B-FSK. Le nombre de bites par symbole est ainsi facilement obtenu, avec la modification du codeur. Cette architecture utilise en son cœur un algorithme de type CORDIC, ainsi, leur implémentation en FPGA montre une très faible consommation en ressource, ce qui donne la possibilité de faire la parallélisation pour augmenter la fréquence de fonctionnement ou d'implanter d'autres architectures en utilisant le reste des ressources,

- la deuxième structure proposée est une architecture numérique et générique de démodulation/codage de transmission de données, qui fournissent une variété de services. Cette architecture adaptative repose principalement sur l'utilisation de structures à compteurs, basée sur des registres LFSR, dont la dimension polynômiale caractéristique permet de contrôler la durée du comptage correspondant aux durées imposées au sein des protocoles de transmission. De plus, cette architecture permet par simple configuration de modifier la structure de ces registres permettant son adaptation à de nombreuses applications. Selon la configuration, cette architecture peut réaliser la démodulation de phase et la démodulation de fréquence et également, elle est capable de générer le code Gold nécessaire dans les systèmes WCDMA et de prendre en charge la fonctionnalité d'un turbo code de type parallèle. L'analyse des ressources matérielles logiques nécessaires à son implantation dans un circuit FPGA, à la consommation d'énergie (consommation électrique) et à la fréquence maximale, prouve clairement que l'architecture proposée présente de meilleures performances comparée à une architecture de type Velcro.

Les architectures présentées dans ce chapitre présentent de bonnes capacités d'évolution, et permettent d'envisager une parallélisation possible en multipliant les étages de calcul.

Conclusion et perspectives

Conclusion

L'objectif de ce travail se manifeste dans le contexte de « la radio reconfigurable » à travers la conception de nouvelles architectures entièrement numériques, plus génériques et « universelles » utilisables dans la couche physique et plus précisément dans la partie bande de base d'une chaîne de transmission multistandard.

La réponse à la problématique

Aujourd'hui les concepteurs des systèmes de transmissions sans fil sont en face à la multiplication des normes de télécommunications, en effet cette multiplication oblige les concepteurs à fournir pour les utilisateurs, un terminal unique capable de faire deux types de Handover :

- le Handover horizontal qui qualifie la capacité d'un terminal à "naviguer" entre les différentes zones cellulaires d'un même standard,
- le Handover vertical qui qualifie la capacité d'un terminal à "naviguer" entre les différents réseaux et standards.

La méthode classique utilisée pour implémenter de tels terminaux est la méthode « Velcro », qui se base sur l'utilisation dans un même terminal des composants dédiés à chacun de ces standards. Cette méthode, présente des coûts importants de fabrication et n'autorise pas la prise en compte de nouveaux standards et normes. Toutefois, le concept de « la radio logicielle » permet à un seul équipement de surfer sur plusieurs standards et normes, par un simple téléchargement d'un logiciel ou d'un code qui sera exécuté dans un processeur généraliste.

Cependant le manque d'une puissance de calcul et la grande consommation de puissance de ces processeurs nous imposent à chercher d'autres cibles pour l'implantation de la radio logicielle. Les circuits reconfigurables de type FPGA sont particulièrement séduisants, par leurs faibles coûts, leurs possibilités d'évolution et leurs intérêts économiques pour les productions industrielles.

Ainsi, nos objectifs sont:

- concevoir de nouvelles architectures entièrement numériques, plus génériques et « universelles »
- proposer une architecture reconfigurable à faible temps de reconfiguration.
- être suffisamment flexibles pour supporter des normes de transmissions multiples et pour s'adapter facilement aux évolutions futures.

L'approche suivie pour atteindre ces objectifs, appelée « la paramétrisation » en particulier, nous a permis d'explorer dans ce manuscrit la technique des opérateurs communs. Cette technique demande une étude bibliographique et une synthèse des travaux concernant les architectures numériques pour la modulation et le codage canal pour différents standards. Cette étude nous permet de chercher les similarités entre les différentes architectures pour définir un cahier des charges concernant les caractéristiques principales de l'architecture générique à réaliser. Cette technique est très intéressante puisque par un simple téléchargement de paramètres, l'architecture générique peut facilement s'adapter à la nouvelle application. De plus, la paramétrisation avec la technique d'opérateurs communs peut constituer une technique complémentaire à la reconfiguration partielle d'un FPGA. En effet si elle est combinée avec la reconfiguration partielle, nous pouvons sur la même sous partie du FPGA, faciliter ce type de reconfiguration.

Résultats

Nous avons présenté dans ce mémoire des architectures adaptées au domaine de la radio logicielle et la radio cognitive reposant sur l'approche de la paramétrisation.

Dans la première partie de ce manuscrit nous avons proposée une architecture générique qui cible deux grands systèmes de transmissions sans fils: les systèmes WCDMA et les systèmes OFDM. Plus précisément les systèmes WCDMA sont surtout utilisés dans les standards de troisième génération, par contre les systèmes OFDM couvrent plusieurs standards tels que le WIFI ainsi que les standards de quatrième génération comme le LTE- Advanced.

En pratique, le récepteur RAKE est le récepteur le plus utilisé dans les systèmes WCDMA, soit dans la station mobile en cas de code multiple par utilisateur, soit dans la station de base pour la détection multiutilisateur, alors que l'algorithme FFT est considéré comme le cœur des systèmes OFDM. De plus le récepteur Rake et l'algorithme FFT présentent une occupation de surface très importante dans la chaîne de transmission, et un traitement lourd en termes de calcul. En outre l'algorithme Cordic devient un outil très intéressant de calcul des fonctions élémentaires tels que les fonctions trigonométriques et hyperboliques avec la précision désirée. Cet algorithme est présent dans plusieurs architectures pour la modulation et la démodulation, on le retrouve aussi même dans le calcul de l'algorithme FFT.

Après avoir fait une étude détaillée des architectures sur le récepteur Rake, l'algorithme FFT et le Cordic, nous avons trouvé des ressemblances structurelles entre les trois algorithmes. Plus précisément nous avons cherché les similarités entre l'unité de désétalement dans le récepteur Rake, le processeur élémentaire (Butterfly) dans le FFT-SDF et la cellule dans le processeur Cordic. Les ressemblances structurelles trouvées

permettent de proposer une nouvelle structure reconfigurable (opérateur commun) en suivant la technique de la paramétrisation et l'approche de l'opérateur commun.

En parallèle nous avons aussi proposé une architecture pipeline du multiplicateur complexe utilisé dans le FFT-SDF, et une autre architecture pipeline pour le multiplexeur utilisé dans l'unité de désétalement.

Pour démontrer l'efficacité de l'architecture proposée, nous avons réalisé une comparaison avec la technique Velcro et également avec quelques opérateurs communs et des cellules reconfigurables disponibles dans la littérature. Toutes les architectures ont été implémentées sur la technologie FPGA Virtex-V pour une comparaison équitable. L'opérateur commun proposé fournit un gain important du point de vue de la complexité avec une bonne performance en termes de fréquence de fonctionnement.

L'opérateur commun a une structure très versatile de telle sorte qu'il est capable de prendre en charge facilement la fonctionnalité d'autres algorithmes comme le Viterbi et le Radix.2 FFT. Ces nouvelles fonctionnalités peuvent être obtenues, sans aucun changement de la structure de l'opérateur commun proposé, mais avec une légère modification des paramètres de configuration et avec la collaboration de notre architecture pipeline proposé pour le multiplicateur complexe.

Nous avons proposé une nouvelle approche qui consiste à combiner deux techniques; l'opérateur commun (OC) et la synthèse architecturale (SA), dans le but de réaliser une implémentation optimale qui satisfait les contraintes (temps réel, zone logique, consommation de puissance, etc...).

La mise en œuvre de la technique des opérateurs communs par la synthèse architecturale apporte plus de souplesse et de faciliter pour l'approche de la reconfiguration partielle d'un FPGA, et par conséquent peut être utile pour la conception d'une application générique de communications multi standard.

Dans la deuxième partie de ce manuscrit nous avons proposée deux architectures versatiles:

- la première architecture générique est réalisée pour la modulation numérique de plusieurs types de modulations telles que la M-QAM, M-PSK, et la B-FSK. Cette architecture repose sur l'algorithme Cordic et une version universelle du codeur (Symbol Mapper).
- la deuxième architecture proposée peut prendre en charge quatre fonctionnalités, qui selon la configuration peut réaliser la démodulation de phase, la démodulation de fréquence, générer le code Gold nécessaire dans les systèmes WCDMA et prendre en charge la fonctionnalité d'un turbo code de type parallèle. Le cœur de cette architecture

est un registre reconfigurable dont le polynôme générateur et le degré du registre sont modifiables selon l'application.

Les deux architectures montrent une très faible consommation en ressource dans le FPGA et une fréquence de fonctionnement très grande. Cette faible consommation de ressource est nécessaire pour réaliser la parallélisation ou pour implanter d'autres architectures en utilisant le reste des ressources.

Perspectives et valorisation des résultats

Dans l'avenir, il faut mettre en place un réseau de processeurs pour mettre en œuvre une gestion optimisée des communications multi-standards, et adapter un système de transmission de données sans fil très haut débit. Par conséquent, une architecture générique spécifique peut être réalisée si nous faisons l'interconnexions de plusieurs modules de l'opérateur commun proposé (en pipeline ou en parallèle) avec des blocs de mémoires. Celle-ci doit être capable de prendre en charge plusieurs fonctions:

- un récepteur Rake utilisé pour les systèmes WCDMA.
- un processeur Cordic utilisé pour le calcul des fonctions trigonométriques.
- un processeur FFT de type pipeline tel que le FFT-SDF pour les systèmes SISO-OFDM.
- un décodeur de Viterbi.

Par conséquent la première démarche pour la réalisation de telle architecture est d'affecter un nombre N de structures universelles (opérateur commun) proposées précédemment (figure III.12). Mais les contraintes de chaque standard imposent des besoins spécifiques pour chaque fonction de cette architecture générique. En effet, dans les standards de communications, il y a une différence dans le nombre de bits utilisés pour le codage. Donc l'architecture générique doit être capable de prendre en compte cette différence. L'architecture générique doit aussi être capable de prendre en compte le besoin de faire un changement par exemple:

- la taille (size) de la FFT,
- la précision dans le processeur Cordic,
- le nombre de trajet et code par utilisateur traités dans le récepteur Rake,
- le nombre d'états des décodeurs Viterbi qui peuvent varier en fonction des standards et des applications avec des besoins de débits très différents.

Ces contraintes qui changent en fonction des modes et des standards considérés, posent un problème à l'affectation et à la connexion des opérateurs communs.

La poursuite de ces travaux conduisent ainsi à l'élaboration d'un système complexe en réseaux de MPSoC à base de NoC (Réseau sur Puce adaptatif) et de nœuds numériques

versatiles pour la transmission de données en réseaux sans fil à très haut débit (mécanismes d'auto-modulation et codage).

Bibliographie

- [1].A. Chamas Al. Ghouwayel, A.Haj-Ali and Z. El-Bazzal "Towards a Triple Mode Common Operator FFT for SoftWare Radio Systems" ICT 2012, Beyrouth : Lebanon (2012).
- [2].A. Benhaoues, C. Tanougast, H. Mayache, S. Toumi, A. Dandache, Digital synthesis architecture for modulation and demodulation, in: The 25th Int. Conference on Microelectronics, IEEE Advancing Technology for Humanity, 2013, pp. 1–4.
- [3].M. Naoues, D. Noguét, Y. Loüet, K. Grati, A. Ghazel, An efficient flexible common operator for FFT and Viterbi algorithms, in: IEEE73rd Vehicular Technology Conference, VTC Spring 2011, Budapest, Hungary.
- [4].A. Benhaoues, S. Toumi, C. Tanougast, E. Bourennane, K. Messaoudi, H. Mayache , « Versatile digital architecture for mobile terminal », Microprocessors and Microsystems, Volume 39, Issue 6, August 2015, Pages 405-417, ISSN 10.1016/j.micpro.2015.05.008. Elsevier.
- [5].Laurent Alhaus. Architecture Reconfigurable pour un Équipement Radio Multistandard. PhD thesis, Université de Rennes 1, 2010.
- [6].Anne Wiesler, H. Schober, R. Machauer et Friedrich Jondral, « Software radio structure for UMTS and second generation mobile communication systems ». In 50th IEEE Vehicular Technology Conference (VTC), vol. 2, pages 939–942, September 1999.
- [7].A. Wiesler and F.K. Jondral. A software radio for second- and third generation mobile systems. Vehicular Technology, IEEE Transactions on, 51(4): 738–748, 2002.
- [8].L. Alaus, D. Noguét and J. Palicot, A Reconfigurable Linear Feedback Shift Register Operator for Software Defined Radio Terminal, In International Symposium on Wireless Pervasive Computing, 2008, May 2008, Santorini, Greece.
- [9].S. Kumar Das, MOBILE HANDSET DESIGN, John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop, # 02–01, Singapore 129809 Copyright 2010.
- [10]. G. Baudoin et coll., Radio-communications numériques; Principes, modélisation et simulation,. Ed. Dunod, 2eme edition, Paris 2007.
- [11]. ITU ou UIT, online : <http://www.itu.int>
- [12]. GSM world website : <http://www.gsmworld.com>
- [13]. H. Holma and A. Toskala, « W-CDMA for UMTS », 3rd Edition, John Wiley & Sons, West Sussex, England, 2004.
- [14]. LTE world website : <http://www.lte-world.com>
- [15]. 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.
- [16]. A. Tânenbaum, Réseaux, 4th Edition, Pearson Education France, 2003.

- [17]. J. Mitola. The software radio architecture. *Communications Magazine*, IEEE, 33(5): 26–38, 1995.
- [18]. Malek NAOUES, «Management d'opérateurs communs dans les architectures de terminaux multistandards », Thèse de doctorat, LICM, Université de Carthage Ecole Supérieure des Communications de Tunis Ecole Doctorale en TIC ; 2013.
- [19]. Ioannis Dages, Andreas Zalonis, Nikos Dimitriou, Konstantinos Nikitopoulos, and Andreas Polydoros. Flexible Radio: A Framework for Optimized Multimodal Operation via Dynamic Signal Design. *EURASIP Journal on Wireless Communications and Networking*, 3: 284–297, 2005.
- [20]. Andreas Polydoros, Jukka Rautio, Giuseppe Razzano, Hanna Bogucka, Diego Ragazzi, Panos I. Dallas, Aarne Mämmelä, Michael Benedix, Manuel Lobeira, and Luigi Agarossi. WIND-FLEX : Developing a Novel Testbed for Exploring Flexible Radio Concepts in an Indoor Environment. *IEEE Communications Magazine*, July 2003.
- [21]. Frank H.P. Fitzek, *Cooperation in Wireless Networks, Principles and Applications: Real Egoistic Behavior Is to Cooperate*, Kluwer Academic Publishers, 2006.
- [22]. Dr. Joseph Mitola III. Technical challenges in the globalization of software radio. *IEEE Communications Magazine*, 37(2): 84–89, 1999.
- [23]. Dr. Joseph Mitola III. Software radios: Survey, critical evaluation and future directions. *Aerospace and Electronic Systems Magazine*, IEEE, 8(4): 25–36, Apr 1993.
- [24]. Texas Instruments website: <http://www.digikey.com/product-detail/en/ADC083000CIYB%2FNOPB/ADC083000CIYB%2FNOPB-ND/1628673>.
- [25]. D Greifendorf, J Stammen, S Sappok, M Ackeren Van, and P Jung. A novel hardware design paradigm for mobile "software defined radio" terminals. In *IEEE Seventh International Symposium on Spread Spectrum Techniques and Applications*, 2002.
- [26]. Jacques Palicot et Christian Roland, « La radio logicielle: enjeux, contraintes et perspectives ». In *Revue de l'Electricité et de l'Electronique*, 2001, n°11, pp. 60-67 (12 ref.), ISSN 1265-6534.
- [27]. Joseph Mitola and Gerald Q. Maguire. *Cognitive Radio: Making Software Radios More Personal*. *IEEE Personal Communications*, pages 13–18, August 1999.
- [28]. D. Linda "Essentials of cognitive radio," *Cambridge Wireless Essentials Series*, Cambridge University Press, 2009.
- [29]. J. Mitola. *Cognitive Radio: An integrated Agent Architecture For Software Defined Radio*. PhD thesis, Royal Institute of Technology of Stockholm, 2000.
- [30]. <http://hal.archives-ouvertes.fr/docs/00/68/23/44/PDF/RC-Pres-Benmammar.pdf>.
- [31]. D. Noguét, « ORACLE: the wiser way to communicate ». *eStrategies*, 2008. www.ist-oracle.org.

- [32]. Sylvain Loizeau, «Conception et Optimisation d'antennes reconfigurables multifonctionnelles et ultra large bande», Thèse de Doctorat L'UNIVERSITE PARIS-SUD XI, décembre 2009.
- [33]. Y. Hawk, S. Hemmady, G. Balakrishnan, and C. G. Christodoulou, "Demonstration of a cognitive radio frontend using optically pumped reconfigurable antenna systems (OPRAS)," *IEEE Trans. Antennas Propag.*, vol. 60, no. 2, pp. 1075–1083, Feb. 2012.
- [34]. BEN TRAD Imen, 'Antennes Agiles pour les Télécommunications Multistandards', thèse DOCTEUR IETR-INSA Rennes, le 29.10.2014.
- [35]. G.-M. Zhang, J.-S. Hong, G. Song, B.-Z. Wang, "Design and Analysis of a Compact Wideband Pattern Reconfigurable Antenna with Alternate Reflector and Radiator," *IET Microwaves, Antennas and Propagation*, Vol. 6, Iss. 15, pp. 1629–1635, 2012.
- [36]. Ludovic COLLOT, «Étude de nouvelles architectures de filtres RF intégrés dans le contexte de la radio opportuniste», Thèse de Doctorat DE L'UNIVERSITÉ DE LIMOGES, novembre 2011.
- [37]. Yahya LAKYS, «FILTRES A FREQUENCE AGILE TOTALEMENT ACTIFS: THEORIE GENERALE ET CIRCUITS DE VALIDATION EN TECHNOLOGIE SiGe BiCMOS 0.25 μ m », Thèse de Doctorat L'UNIVERSITE BORDEAUX 1, décembre 2009.
- [38]. Jérémy DUFRAISSE, 'Étude des classes de fonctionnement à haut rendement pour l'amplification de puissance en hyperfréquence en utilisant la technologie HEMT à base de nitrure de gallium', thèse DOCTEUR DE L'UNIVERSITÉ DE LIMOGES, le 21 décembre 2012.
- [39]. J. K. Cavers, Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements, *IEEE Trans. Vehicular Technology*, VOL. 39, NO. 04, pp. 374–382, Nov. 1990.
- [40]. N. Deltimple, E. Kerhervé, Y. Deval and P. Jarry, "A Reconfigurable RF Power Amplifier Biasing Scheme", *Proceedings of the 2nd annual IEEE Northeast Workshop on Circuits and Systems (NEWCAS2004)*, pp. 365-368, Montréal, Canada, June 20-23, 2004.
- [41]. Bert Henderson and Edmar Camargo, *Microwave Mixer Technology and Applications* (Artech House Microwave Library). Artech House (July 31, 2013).
- [42]. Emmanuel Dupouy, 'Etude et réalisation d'un oscillateur bande X, contrôlé en tension, à varactor MEMS, pour application spatiale', thèse DOCTEUR DE L'UNIVERSITÉ DE LIMOGES, le 27 février 2009.
- [43]. C. Majek, N. Deltimple, H. Lapuyade, J.-B. Bégueret, E. Kerhervé and Y. Deval, "A Programmable CMOS RF Frequency Synthesizer for Multi-standard Wireless Applications", *Proceedings of the 2nd annual IEEE Northeast Workshop on Circuits and Systems (NEWCAS2004)*, pp. 289-292, Montreal, Canada, June 20-23, 2004.

- [44]. Jan Sevenhans and Zhong-Yuan Chang « Analog Circuit Design » Chapter “Architectures and Circuits for A/D and D/A Conversion in CMOS Integrated Systems for Telecom Applications” pp 65-88, Springer US 1997.
- [45]. Arthur H. M. van Roermund, Herman Casier Michiel Steyaert « Analog Circuit Design; Smart Data Converters, Filters on Chip, Multimode Transmitters » Springer Science+Business Media B.V. 2010.
- [46]. Researchgate « Article; Architecture reconfigurable pour la numérisation du signal radio de récepteurs mobiles multi-standards »
http://www.researchgate.net/publication/29975133_Architecture_reconfigurable_pour_la_numrisation_du_signal_radio_de_recepteurs_mobiles_multi-standards.
- [47]. Fabrice URBAN, ‘Implantation optimisée d’estimateurs de mouvement pour la compression vidéo sur plates-formes hétérogènes multi-composants’, Thèse, Institut National Des Sciences Appliquées De Rennes, Décembre 2007.
- [48]. <http://www.xilinx.com/products/silicon-devices/fpga/virtex-7.html>
- [49]. <https://www.altera.com/products/fpga/stratix-series/stratix-v/overview.html>
- [50]. L. Bisdounis, C. Dre, S. Blionas, D. Metafas, A. Tatsaki, F. Ieromnimon, E. Macii, P. Rouzet, R. Zafalon and L. Benini « Low-power system-on-chip architecture for wireless LANs » [Computers and Digital Techniques, IEE Proceedings - \(Volume:151, Issue: 1\)](#) 15 Jan. 2004.
- [51]. Xilinx. ISE 10.1i software manuals. Xilinx; 2008.
- [52]. 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.
- [53]. <http://docs-europe.electrocomponents.com/webdocs/0de1/0900766b80de1aa8.pdf>
- [54]. <http://www.analog.com/en/products/processorsdsp/blackfin/adspbf561.html#product-overview>.
- [55]. Jean-Philippe Delahaye « Plate-forme hétérogène reconfigurable : application à la radio logicielle » thèse Docteur de l’Université de Rennes I. 18 avril 2007.
- [56]. Pierre-Henri Horrein, Christine Hennebert, Frédéric Pétrot « Integration of GPU computing in an Software Radio environment » *Journal of Signal Processing Systems* October 2012, Volume 69, Issue 1, pp 55-65. DOI 10.1007/s11265-011-0639-1.
- [57]. Joel Gregory Millage « GPU Integration into a Software Defined Radio Framework » *Theses MASTER OF SCIENCE*. Iowa State University Ames, Iowa 2010.
- [58]. J. KIM, S. HYEON ET S. CHOI. Implementation of an SDR System Using Graphic Processing Unit. *IEEE Communications Magazine* • March 2010
- [59]. G.J. Minden, J.B. Evans, L. Searl, D. DePardo, V.R. Petty, R. Rajbanshi, T. Newman, Q. Chen, F. Weidling, J. Guffey, D. Datla, B. Barker, M. Peck, B. Cordill, A.M. Wyglinski, and A. Agah. “KUAR: A flexible software-defined radio

- development platform”. In *New Frontiers in Dynamic Spectrum Access Networks*, 2007. DySPAN 2007. 2nd IEEE International Symposium on, pages 428–439, April 2007.
- [60]. Friedrich Jondral. Chapitre 8. Parametrization- A technique for SDR implementation. In Walter Tuttlebee, editor, *Software Defined Radios: Enabling Technologies*, pages 233–256. Wiley, 2002.
- [61]. Hiroshi Harada, Yuki Yoshi Kamio et Masayuki Fujise, « Multimode Software Radio System by Parameter Controlled and Telecommunication Component Block Embedded Digital Signal Processing Hardware ». *EICE TRANSACTIONS on Communications*, vol. E83-B, no6, pages 217–228, 2000.
- [62]. S.T. Gul. Optimization of Multi-standards Software Defined Radio Equipments: A Common Operators Approach. PhD thesis, SUPELEC Rennes, 2009.
- [63]. F. Jondral. «A software radio view of modulation and spreading for UTRA FDD and UTRA TDD » *3G Mobile Communication Technologies*, 2002. Third International Conference on (Conf. Publ. No. 489).
- [64]. Arnd-Ragnar Rhiemeier, « Benefits and limits of parameterised channel coding for software radio ». In *Proc. 2nd Workshop on Software Radios*, (Karlsruhe, Germany), pages 107–112, March 2002.
- [65]. J. Cavallaro et M. Vaya, « VITURBO: A Reconfigurable Architecture for Viterbi and Turbo Decoding ». In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, (Hong Kong, China), April 2003.
- [66]. V. Rodriguez, C. Moy, and J. Palicot, “Install or invoke?: The optimal trade off between performance and cost in the design of multi-standard reconfigurable radios,” *Wiley InterScience, Wireless Communications and Mobile Computing Journal*, vol. 7, pp. 1143–1156, November 2007.
- [67]. L. Alaus, J. Palicot, C. Roland, Y. Louët, and D. Noguét, “Promising Technique of Parametrisation For Reconfigurable Radio, the Common Operators Technique: Fundamentals and Examples,” *Springer Science + Business Media LLC*, 2009.
- [68]. J. Palicot et C. Roland, « FFT: a basic function for a reconfigurable receiver ». *Telecommunications*, 2003. *ICT 2003. 10th International Conference on*, vol. 1, pages 898–902 vol.1, Feb.-1 March 2003.
- [69]. A. Al Ghouwayel, Y. Louët et J. Palicot, « A reconfigurable architecture for the FFT operator in a software radio context ». *Circuits and Systems*, 2006. *ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pages 4 pp.–, 0-0 2006.
- [70]. L. Alaus, D. Noguét and J. Palicot. An Extended Reconfigurable Linear Feedback Shift Register Operator for Software Defined Radio. In *IEEE International Symposium on Spread Spectrum Techniques and Applications*, 2008
- [71]. M. Naoues, D. Noguét, L. Alaus, Y. Louët, “A common operator for FFT and FEC decoding,” *Microprocessors and Microsystems*, Volume 35, Issue 8, November 2011, Pages 708-715, ISSN 0141-9331, 10.1016/j.micpro.2011.08.007.

- [72]. L. Alaus, D. Noguét, and Jacques Palicot. A common operator bank to resolve scheduling issue on a complexity optimized sdr terminal. In Telecommunications (AICT), 2010 Sixth Advanced International Conference on, pages 142–146, 2010.
- [73]. Mark Cummings et S. Haruyama, « FPGA in the Software Radio ». IEEE Communications Magazine, vol. 37, no2, pages 108–112, Feb 1999.
- [74]. BEN TRAD Imen, ‘Antennes Agiles pour les Télécommunications Multistandards’, thèse DOCTEUR IETR-INSA Rennes, le 29.10.2014.
- [75]. L. Harju, M. Kuulusa, and J. Nurmi, « Flexible Implementation of a W-CDMA RAKE Receiver », Journal of VLSI Signal Processing, vol. 39, p. 147–160, 2005.
- [76]. L. Harju, M. Kuulusa, and J. Nurmi. “A Flexible RAKE Receiver Architecture for WCDMA Mobile Terminals”. Proc. IEEE Third Workshop on Signal Processing Advances in Wireless Communications (SPAWC '01), 2001, pp. 9–12.
- [77]. M. Chugh, D. Bhatia and T. Balsara, «Design and Implementation of Configurable W-CDMA RAKE Receiver architectures on FPGA», Proc. of 19th IEEE Int. Parallel and Distributed Proc. Symposium, p. 145b, avr. 2005.
- [78]. M. Youssef, F. Monteiro, A. Dandache, C. Diou, « An Effective and Scalable Multiuser Architecture for the Base Station Receiver », 2nd IEEE International Conference on Signal Processing and Communication Systems, December 15-17, 2008, Gold Coast, Australia.
- [79]. M. Youssef, F. Monteiro, A. Dandache, C. Diou, «A Cost-Effective Parallel Architecture for the CodeRAKE Receiver », 14th IEEE International Conference on Electronics Circuits and Systems, December 11-14, 2007, Marrakech, Morocco.
- [80]. M. Youssef, C. Diou, F. Monteiro, A. Dandache, « CodeRake : a new small-area scalable architecture for the multi-user/multi-code Rake receiver », 13th IEEE International Conference on Electronics Circuits and Systems, December 10-13, 2006, Nice, France.
- [81]. M. Chugh, D. Bhatia, T. Balsara, Design and Implementation of Configurable WCDMA RAKE Receiver architectures on FPGA, in: Proc. of 19th IEEE Int. Parallel and Distributed Proc. Symposium, p. 145b, avr. 2005.
- [82]. Bo Fu, Paul Ampadu, An area efficient FFT/IFFT processor for MIMOOFDMWLAN 802.11n, J. Sign. Process Syst. 56 (2009) 59–68, <http://dx.doi.org/10.1007/s11265-008-0264-9>.
- [83]. J.M. Rudagi, R. Lobo, P. Patil, N. Biraj, An efficient 64-point pipelined FFT engine, in: 2010 International Conference on Advances in Recent Technologies in Communication and Computing 2010, pp. 204–208.
- [84]. Hsiang S. Hu, Hsiao-Y. Chen and Shyh-J. Jou, 'Novel FFT Processor with Parallel-In-Parallel-Out in Normal Order', VLSI Design Automation and Test Symposium, Page(s): 150 - 153, 2009.
- [85]. G. Santella, A frequency and symbol synchronization system for OFDM signals: architecture and simulation results, IEEE Trans. Veh. Technol. 49 (1) (2000) 254–275.

- [86]. C. Vennila, G. Lakshminarayanan, Seok-Bum Ko, Dynamic partial reconfigurable FFT for OFDM based communication systems, *Circ. Syst. Signal Process* 31 (2012) 1049–1066, <http://dx.doi.org/10.1007/s00034-011-9367-9>.
- [87]. M. Turrillas, A. Cortés, J.F. Sevillano, I. Vélez, C. Oria, A. Irizar, V. Baena, Comparison of area-efficient FFT algorithms for DVB-T2 receivers, *Electron. Lett.* 46 (15) (2010).
- [88]. Waqar Hussain, Fabio Garzia, Tapani Ahonen, Jari Nurmi, Designing fast fourier transform accelerators for orthogonal frequency-division multiplexing systems, *J. Sign. Process Syst.* 69 (2012) 161–171, <http://dx.doi.org/10.1007/s11265-011-0642-6>.
- [89]. YOUSSEF ACHOURI, 'IMPLÉMENT A TI ON EFFICACE DE LA FFT POUR DES COMMUNICATIONS OFDM', MAÎTRISE EN GÉNIE ÉLECTRIQUE de L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES, décembre 2010.
- [90]. Pei-Yun Tsai, Tzi-Dar Chiueh « A Low-Power Multicarrier-CDMA Downlink Baseband Receiver for Future Cellular Communication Systems» *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS*, VOL. 54, NO. 10, OCTOBER 2007.
- [91]. J. Takala and K Punkka. Butterfly unit supporting radix-4 and radix- 2 fft. In *Proceedings of The 2005 International TICSP Workshop on Spectral Methods and Multirate Signal Processing, SMMSP 2005, Riga, Latvia, volume 30, pages 47–54, 20-22 June 2005.*
- [92]. J. E. Volder, "The CORDIC trigonometric computing technique", *IRE Trans. Electron. Comput.*, vol. EC-8, pp. 330–334, Sept. 1959.
- [93]. G.ARNOULD, " Etude et Conception d'Architectures Haut-Débit pour la Modulation et la Démodulation Numériques", Manuscript of thesis, LICM, University of Metz, 2006.
- [94]. Xilinx Inc., Virtex 5 FPGAs datasheet (Xilinx Inc., San Jose, 2008)
- [95]. H. Lange, O. Franzen, H. Schrder, M. Bücken, B. Oelkrug, Reconfigurable Multiply-Accumulate-based Processing Element, in: *Proc. of the IEEE Workshop on heterogeneous Systems on a Chip, Hamburg, Germany, 2002.*
- [96]. I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs", *IEEE Trans. on Computer-aided design of integrated circuits and systems*, Vol. 26, N°2, 2007, pp. 203-215.
- [97]. James W. Cooley and John W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 19(90) :297–301, 1965.
- [98]. T. Liu, C. Tanougast, S. Weber, A framework of architectural synthesis for dynamically reconfigurable FPGAs, in: *IEEE International SOC Conference, IEEE Circuits and Systems Society, 2008, pp. 283–286.*
- [99]. Ting LIU. « Optimisation par synthèse architecturale des méthodes de partitionnement temporel pour les circuits reconfigurables» Thèse doctorat,

Spécialité: Instrumentation et Micro-Electronique) l'Université Henri Poincaré – Nancy 1, Mai 2008.

- [100]. Ting Liu et al: Toward a methodology for optimizing algorithm-architecture adequacy for implementation reconfigurable system. In Proc. of 13th IEEE International Conference on Electronics, Circuits and Systems (ICECS2006), IEEE Computer Society Press, December 2006.
- [101]. C. Tanougast, Y. Berviller, P. Brunet, S. Weber and H. Rabah, "Temporal partitioning methodology optimizing FPGA resources for dynamically reconfigurable embedded real-time system," *Microprocessors and Microsystems*, Elsevier, Vol. 27, Issue 3, April 2003, pp 115-130.
- [102]. N. Togawa, M. Ienaga, M. Yanagisawa, T. Ohtsuki, An area/time optimizing algorithm in high-level synthesis for control-base hardware, in: *Proceedings of the ASP-DAC 2000, IEEE circuits and systems, Yokohama, Japan, 2000*, pp. 309–312.
- [103]. A. Sharma, R. Jain, Estimating architectural resources and performance for high-level synthesis applications, *IEEE Trans. VLSI Syst.* 1 (2) (1993) 175–190.
- [104]. A. Priadarshini, M. Jagadeeswari, "Low power reconfigurable FPGA based on SRAM", 2013 International Conference on Computer Communication and Informatics, 2013.
- [105]. Katarina Paulsson, Michael Hübner, Jürgen Becker, "Cost-and Power Optimized FPGA based System Integration: Methodologies and Integration of a Low-Power Capacity - based Measurement Application on Xilinx FPGAs", *Design, Automation and Test in Europe*, 2008. pp; 50-55.
- [106]. T. Suresh and K. L. Shunmuganathan "Efficient Resource Sharing Architecture for Multistandard Communication System" Hindawi Publishing Corporation *VLSI Design* Volume 2011, Article ID 328640, doi:10.1155/2011/328640.
- [107]. F. Curticapean and J. Niittylahti, "Low-power direct digital frequency synthesizer", in *Proc. 43rd IEEE Midwest Symp. Circuits and Systems*, vol. 2, Aug. 2000, pp. 822–825.
- [108]. Tierney, C. M. Rader, and B. Gold, "A digital frequency synthesizer", *IEEE Trans. Audio Electroacoust.*, vol. 19, pp. 48–57, Mar. 1971.
- [109]. G. Arnould, F. Monteiro, and A. Dandache, "A digital frequency shift keying demodulator", in *Proc. ICECS'06, Nice, Dec. 2006*.
- [110]. 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.
- [111]. 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).

- [112]. Fuqin Xiong, "Digital Modulation Techniques", p. cm. - (Artech House telecommunications library) Includes bibliographical references and index. ISBN 0-89006-970-0 (alk. paper) Copyright 2000.
- [113]. P. Altke, "Efficient Shift Registers, LFSR Counters, and Long Pseudo-Random Sequence Generators," Xilinx Application Note XAPP-052, 1996.
- [114]. M. M'sir, « Conception d'architectures rapides pour codes convolutifs en télécommunications : Application aux turbo-codes », Manuscript of thesis, LICM, University of Metz, 2003.
- [115]. S Gul, L Alaus, J Palicot, C Moy, and N Noguet. Linear feedback shift registers as common operators in cognitive radio. In International Journal of Autonomous and Adaptive Communications, 2010.
- [116]. H. Jaber, et al., «Design of Parallel Fault-Secure Encoders for Systematic Cyclic Block Transmission Codes », Microelectronics Journal, volume 40, issue 12, pp. 1686–1697, Dec. 2009.
- [117]. P. Elias, « Error Free Coding », IRE Transaction on Inf. Theory, vol. IT-4, pp. 29–37, September 1954.
- [118]. A. M'sir, F. Monteiro, A. Dandache, B. Lepley, « Design of a high speed parallel encoder for convolutional codes », Microelectronics Journal, vol. 35(2), pp. 151–166, Feb. 2004.
- [119]. R. Gold, « Maximal recursive sequences with 3-valued recursive cross-correlation function », IEEE trans. Info. Theory, vol. 4, p. 154–156, janv. 1968.
- [120]. M. A. Mohamed « FPGA Synthesis of VHDL OFDM System », Wireless Pers Commun (2013) 70:1885–1909. DOI 10.1007/s11277-012-0786-0.

Publications de l'auteur

Revue:

A. Benhaoues, S. Toumi, C. Tanougast, E. Bourennane, K. Messaoudi, H. Mayache , « Versatile digital architecture for mobile terminal », Revue **Microprocessors and Microsystems**, Éditeur Elsevier, Volume 39, Issue 6, August 2015, Pages 405-417, ISSN 10.1016/j.micpro.2015.05.008.

Communications internationales :

A. Benhaoues, H.Mayache, C.Tanougast, E. Bourennane, S. Toumi, "UNIVERSAL ARCHITECTURE FOR DEMODULATION AND MODULATION " **International Conference on Embedded Systems in Telecommunications and Instrumentation** Annaba, Algeria, "ICESTI'12" November , 05-07, 2012.

A. Benhaoues, S. Toumi, C.Tanougast, H.Mayache," Architecture numérique générique et universel" 80ème Congrès de l'ACFAS « **Association Francophone pour le savoir** » du 7 au 11 Mai 2012 Montréal. CANADA.

A. Benhaoues, H. Mayache, C.Tanougast, A. Dandache, S. Toumi, "Digital synthesis architecture for modulation and demodulation", **The International Conference on Microelectronics (ICM'13)**, *IEEE advancing Technology for Humanity*, 2013, p.1-4 Beirut Lebanon.

A. Benhaoues, Z.Asradj, C.Tanougast, H.Mayache, S. Toumi, "Digital synthesis architecture for modulation and demodulation", **International Congress on Telecommunication and Application 2014, ICTA'14**. Bejaia – Algeria.

A. Benhaoues, E. Bourennane, C.Tanougast, H. Mayache, S. Toumi, K. Messaoudi, "Implementation of Universal Digital Architecture using 3D-NoC for Mobile Terminal ", *The International Conference on Control, Decision and Information and Technologies. Codit'14, IEEE France*, 2014.

Communications nationales:

A. Benhaoues, « Architectures numériques versatiles avancées pour la transmission sans fil de données à très haut débit ». **1ères Journées Doctorales sur l'Automatique, les Télécommunications, l'Instrumentation et les Multimédia « JIDATIM'12 »** 16 et 17 Janvier 2012, Annaba. Algérie.

Liste des autres publications et conférences:

Kamel Messaoudi, Hichem Mayache, **Atef Benhaoues**, El-Bay Bourennane, Salah Toumi, 'Connection of H.264/AVC Hardware IPs Using a Specific Networks-on-Chip', Elsevier, *Microprocessors and Microsystems*, Volume 39, Issue 8, November 2015, Pages 609-620.

Mayache Hichem, **Benhaoues Atef**, Messaoudi Kamel, Bourennane El Bey, Toumi Salah, 'Designing Mesh 2x2 network on chip adapted for Wishbone protocol', **ICESTI'12, International Conference on Embedded Systems in Telecommunications and Instrumentation**, November 5–7, 2012, Annaba, Algeria.

H. Mayache, E. Bourennane, **A. Benhaoues**, S. Toumi, K. Messaoudi, " A scalable Router and Effective XYZ Routing Algorithm", **The 3rd International Conference on Information, Processing and Electrical Engineering (IEEE - ICIPEE'14)**, Tebessa-Algeria.

H. Mayache, **A. Benhaoues**, E. Bourennane, K. Messaoudi, S. Toumi, 'Network on Chip Platform Generator', **International Conférence on Embedded Systems in Télécommunications and Instrumentation (ICESTI'14)**, Annaba, 27-29 octobre 2014, Algérie.

Projets

PROJET : TASSILI N° 11MDU844. Titre du projet: **Modélisation, Simulation et Génération d'une Architecture Multiprocesseur Basée sur un NoC de type Wishbone.** Responsables du projet : **En France : Professeur BOURENNANE El Bay** Université de Bourgogne Dijon, Laboratoire d'Electronique, d'Informatique et Image (**Le2i**). **En Algérie : Professeur TOUMI Salah** Université Badji Mokhtar Annaba, Laboratoire d'Etude et de Recherche en Instrumentation et en Communication d'Annaba (**LERICA**).

Annexe A

Système de transmission WCDMA

A. Description d'un système WCDMA

l'UMTS est basée sur une technique d'accès multiple par répartition en code (AMRC) large bande ou W-CDMA ("Wideband Code Division Multiple Access"). Cette technique permet de multiplexer par code sur une bande de 5 Mhz les canaux logiques, voix, données ou services. Une partie importante dans le W-CDMA est la chaîne de transmission des données qui sera réalisée à l'aide des composants analogiques et numériques.

A.1. Emetteur WCDMA.

Le synoptique de l'émetteur devant être implémenté au niveau du terminal mobile est donné sur la figure A.1.

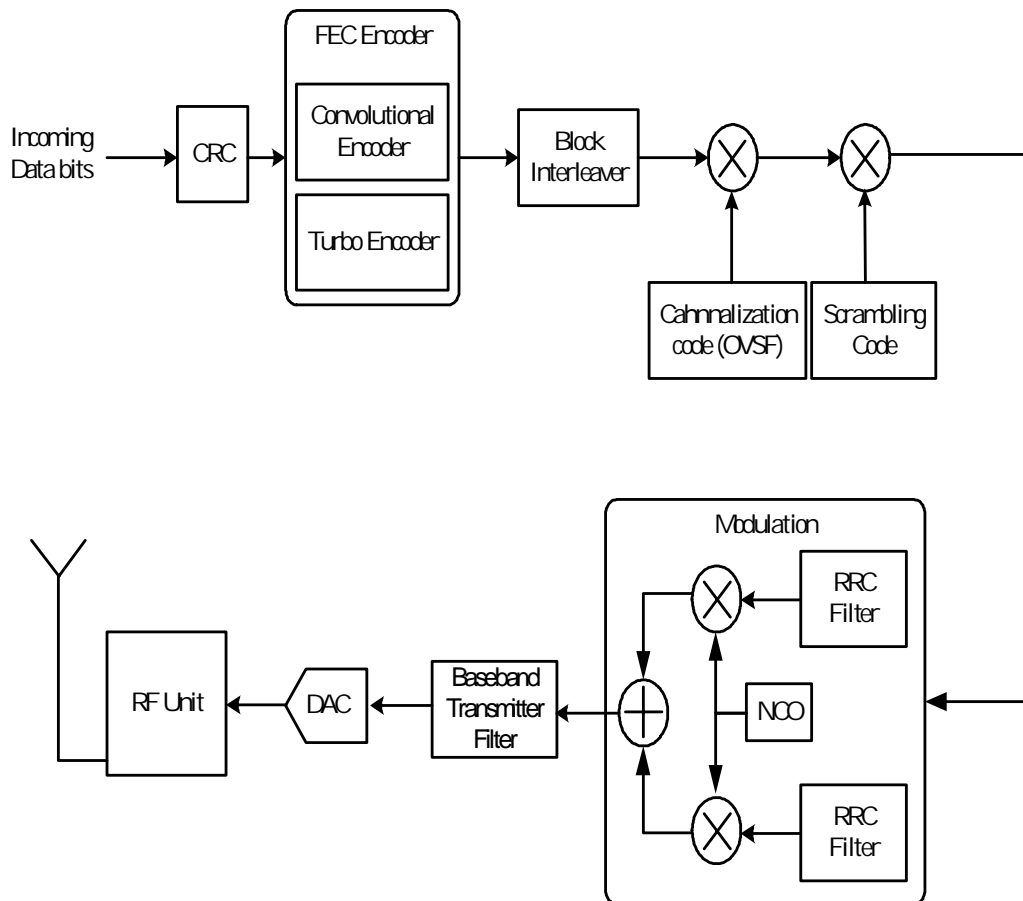


Figure. A.1. – Diagramme en blocs des traitements en bande de base de l'UMTS.

Au début et pour chaque bloc de transport (TB) qui est l'unité de base de données, on ajoute des bits **CRC (Cyclic Redundancy Check)**. Ensuite, le bloc **codage de canal** (de type code convolution ou turbo code) va ajouter de la redondance dont le but de protéger les informations. Après le codage de canal, nous avons un premier **entrelacement** qui agit au niveau bit et renforce les signaux contre les interférences. On ajoute à la séquence de bits entrelacés des bites pilotes pour l'estimation de canal et des bits pour le contrôle de puissance. Un **étalement**, utilisant un code approprié (Code OVSF), permet d'atteindre les 3,84 MegaChips par seconde requis en émission, puis un deuxième code dit le code de brouillage est utilisé pour améliorer les propriétés statistiques des codes de canalisation et aussi pour différencier les utilisateurs présents dans une même cellule. Le **mapping (modulation)** permet de moduler le signal en QPSK ou autre type de modulation. Le signal numérique résultant est converti en un signal analogique à la fréquence intermédiaire IF par un **convertisseur D/A**. Les signaux IF est ensuite **transposé** en signaux RF à 2 GHz, amplifié par un **amplificateur de puissance** et envoyé par une **antenne [9]**.

A.2. Récepteur WCDMA

Le synoptique d'un récepteur WCDMA est représenté sur la figure A.2.

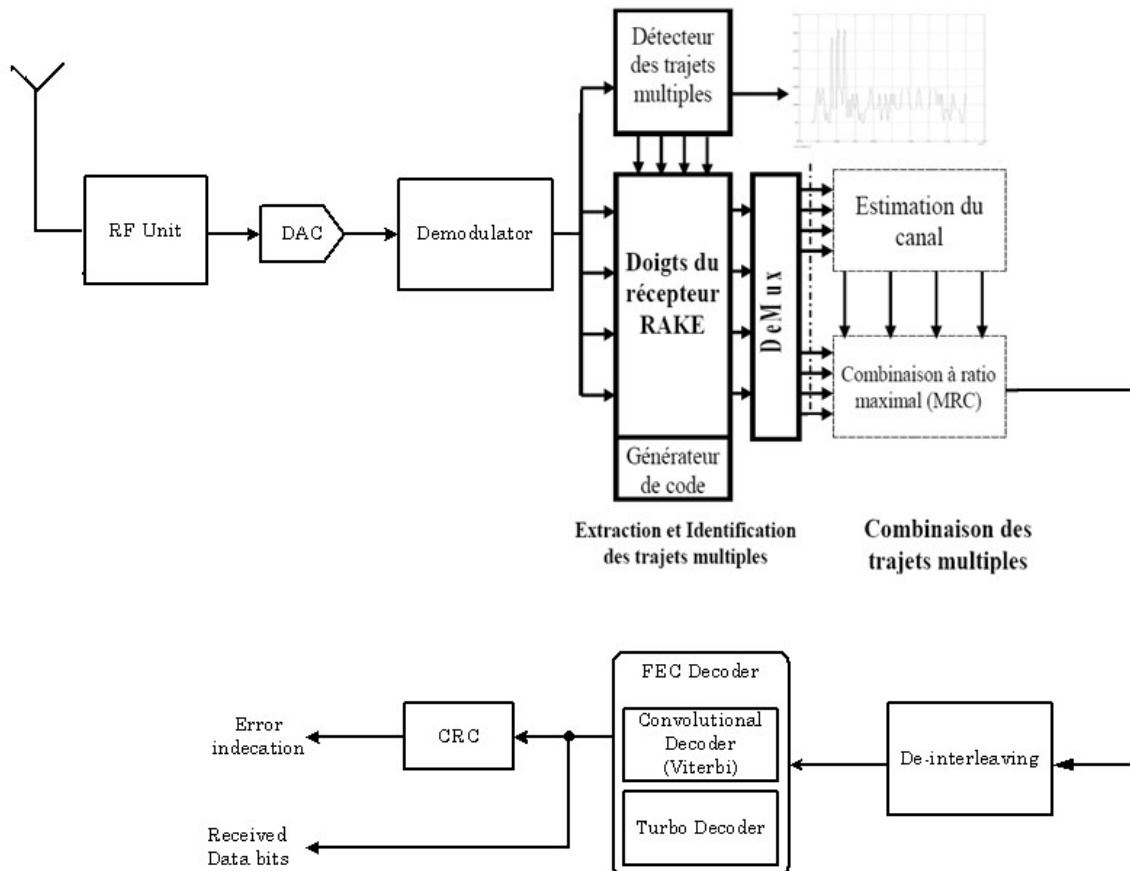


Figure. A.2. Les blocs fonctionnels de l'architecture du récepteur RAKE

Dans la **partie RF**, le signal reçu après filtrage par un filtre RF est amplifié par un **amplificateur faible bruit (LNA)**, ensuite le signal est **translaté** en bande de base (si il s'agit d'un récepteur **hétérodyne** le signal est transposé vers un signal de fréquence intermédiaire IF puis vers le bande de base). Le processus de **conversion analogique-vers-numérique** produit un signal sur lequel travaille la partie bande de base.

Ensuite, les différents copies d'un même signal qui proviennent de différents trajets multiple subissent un désétalement par un composant très important dans le récepteur WCDMA; le **récepteur Rake** (voir section **A.3**). À la sorte du **récepteur Rake**, les séquences de données qui en résultent sont **désentrelacés** (L'opération de désentrelacement est l'inverse de l'opération d'entrelacement faite par l'émetteur) et subissent un **décodage de canal**. En général, le décodage est basé sur l'**algorithme de Viterbi**. Ce dernier est basé sur des diagrammes en treillis qui nécessitent un calcul intensif. La séquence est ensuite récupérée par un block de **décision binaire**, qui est ensuite soumis à un bloque de **détection d'erreur** et enfin transmis à la couche supérieure.

A.3. Récepteur en râteau (Rake)

Dans les systèmes de radiocommunication CDMA (Code Division Multiple Access), le signal reçu est composé de plusieurs répliques du signal émis, liées à la présence de multiples chemins de propagation de l'onde radioélectrique entre l'émetteur et le récepteur. Chaque trajet est caractérisé par un retard et une amplitude complexe. Puisque l'émetteur ou le récepteur n'est pas fixe, le nombre de trajets, leurs amplitudes et retards varient au cours du temps. Ainsi le canal multi-trajets peut être modélisé par un filtre linéaire dont la réponse impulsionnelle varie au cours du temps:

$$h(\tau, t) = \sum_{i=1}^L h_i(t)\delta(t - \tau_i(t)) \quad \text{A.1}$$

$h_i(t)$ et $\tau_i(t)$ sont respectivement l'amplitude complexe et le retard du $i^{\text{ème}}$ trajet, et L désigne le nombre de trajets emprunté par le signal.

Dans un système CDMA, la propriété de trajets multiples est considérée comme un avantage, car un récepteur spécifique nommé récepteur RAKE détecte le plus grand nombre de chemins pour les combiner. Par conséquent, le signal final est renforcée ce qui conduit à une amélioration du rendement.

Comme il est présenté à la figure **A.3**, le récepteur Rake est composé d'un banc de corrélateurs représentant les doigts (Finger) de celui-ci, d'un estimateur de canal et d'un combineur. Chaque doigt permet de désétalement un trajet multiple du signal reçu, pris en compte par le détecteur de trajets. Le signal reçu est corrélé avec le code d'étalement sur une période correspondante au facteur d'étalement. Chaque doigt contient une unité très important qui est le corrélateur ou l'unité de désétalement. Le nombre de corrélateurs

dans le récepteur RAKE dépend du nombre de doigts (c'est-à-dire du nombre de trajets multiples traités) et du nombre de codes par utilisateur; dans le cas conventionnel nous utilisons un corrélateur pour chaque doigt et un corrélateur pour chaque code de canal (code par utilisateur).

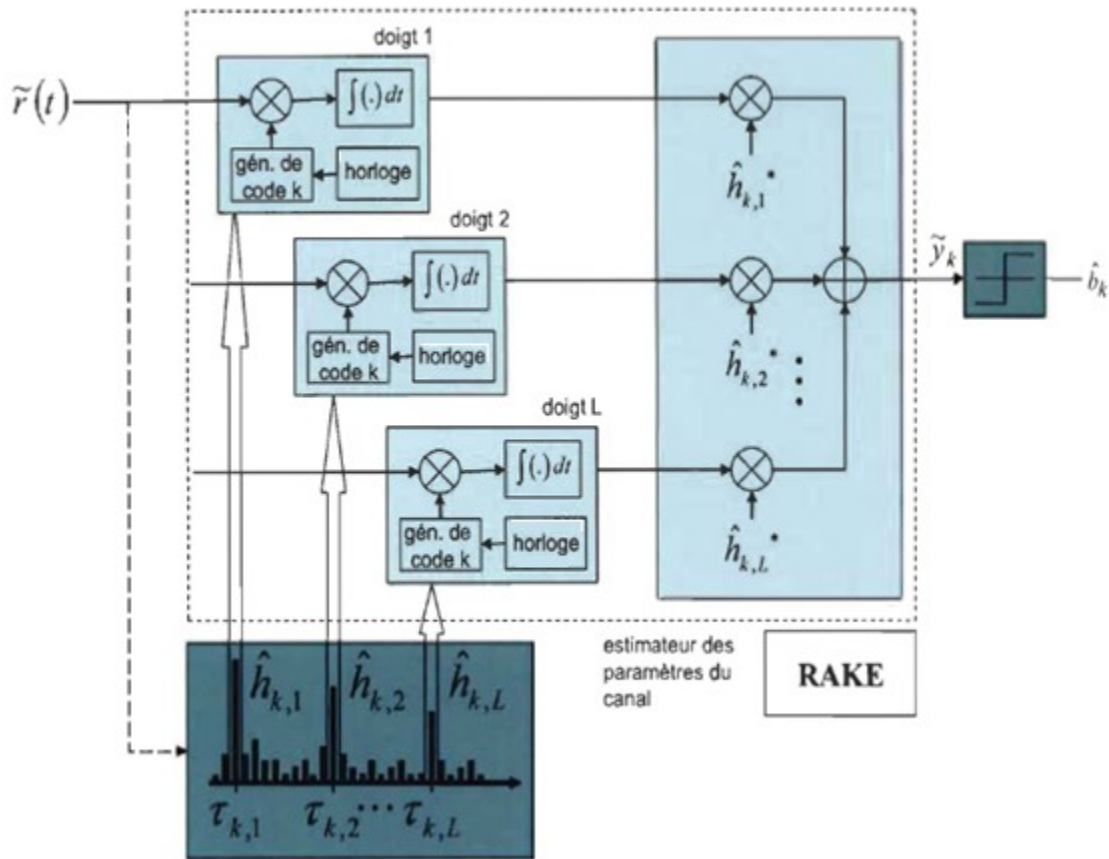


Figure A.3: Structure du récepteur en râteau.

Le rôle du combineur est d'additionner d'une manière constructive les différentes copies des symboles d'information disponible à la sortie de chacun des doigts, permettant d'améliorer les performances du récepteur.

A.4. Détecteur de trajets multiples

Une estimation des retards très précise est nécessaire pour le récepteur Rake, car le générateur de code doit aligner dans le temps avec le signal reçu. Cette estimation est obtenue par une corrélation entre des symboles pilotes dans le signal reçu et les symboles pilotes d'un signal connu par le récepteur. Les résultats de corrélation sont des pics, où les plus forts représentent les trajets les plus puissants.

A.5. Générateur de code

Nous avons deux générateurs de code; le générateur du code de canalisation (OVSF ou Orthogonal Variable Spreading Factor) et le générateur du code de brouillage.

A.5.1. Implantation de générateur de code OVSF

Les codes de canalisation, appelés aussi codes orthogonaux, rendent possible le passage de symboles aux *chips*, c'est-à-dire d'un signal en bande étroite à un signal en large bande. La fonction d'étalement est accomplie en multipliant la partie réelle (voie *I*) et imaginaire (voie *Q*) de chaque symbole d'information par le code de canalisation alloué à l'utilisateur et dont la taille est égale au facteur d'étalement «SF». Ce code peut être le même pour chaque voie suivant que l'on se place dans la voie montante ou dans la voie descendante.

Il existe deux méthodes pour l'implémentation d'un générateur de code de canalisation (OVSF). La première approche est basée sur des tables, elle présente des inconvénients. La deuxième technique est basée sur l'arbre du code (OVSF). Dans la figure A.4, nous pouvons voir l'architecture du générateur de code OVSF utilisé dans le récepteur RAKE. Ce générateur produit le code souhaité au taux d'un *chip* par cycle d'horloge. Le générateur de code OVSF est composé de cinq blocs: un bloc de décalage, un contrôleur, un compteur, un réseau des portes logiques (AND, XOR) et un multiplexeur [75] [76].

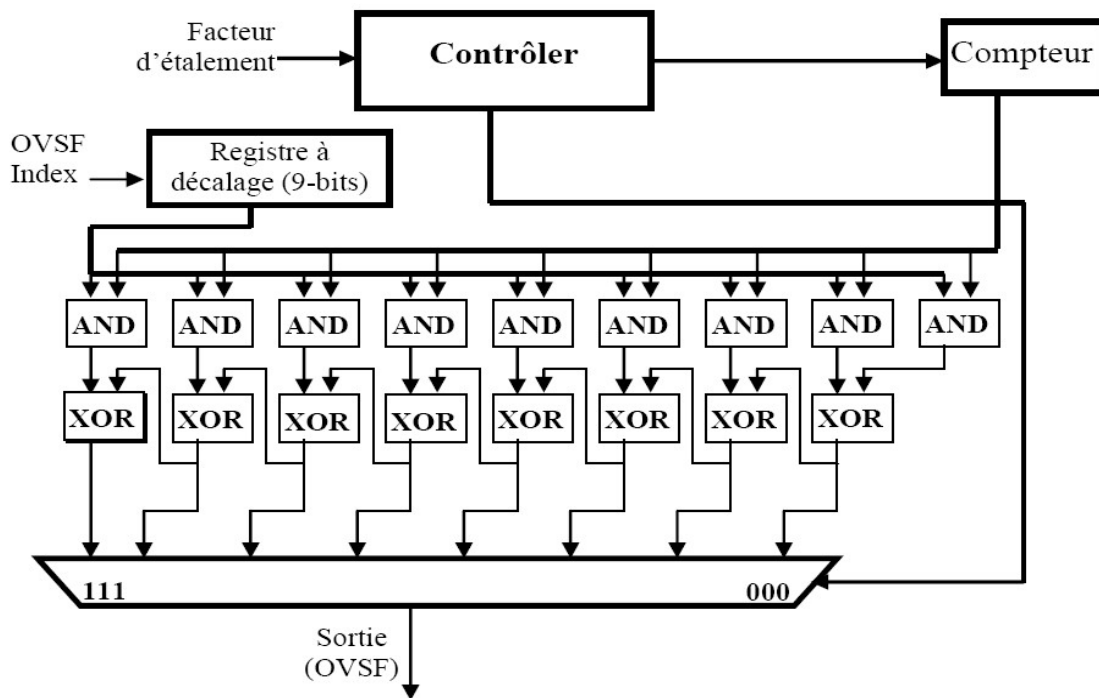


Figure A.4: Générateur de code OVSF.

On peut exprimer le fonctionnement de ce générateur de code par les étapes suivantes: tout d'abord, le bloc de contrôle reçoit le facteur d'étalement et le convertit en un code binaire de largeur 3 bits. Ce code est utilisé comme signal de contrôle pour le bloc de décalage et le multiplexeur. En fait, ce contrôle des deux blocs sert à sélectionner le code OVSF en sortie du générateur. Ensuite, le bloc de décalage stocke les index du code

sélectionné sous forme binaire de largeur 9 bits (0–511). L'index sera décalé à gauche selon le signal de commande de 3 bits, ce signal provenant du bloc de contrôle.

De son côté, le compteur va compter jusqu'à atteindre la valeur correspondant au facteur d'étalement indiqué par l'index au bloc de contrôle. Les neuf bits en sortie du bloc de contrôle sont reliés aux paires de portes (AND/XOR) correspondantes pour être employés comme signaux de commande. Finalement, le réseau de portes logiques (AND, XOR) produit les véritables séquences d'OVSF qui seront disponibles à la sortie de multiplexeur [75].

A.5.2. Implantation de générateur de code de Gold

Le signal étalé par les codes de canalisation dans un système WCDMA est multiplié *chip* par *chip* par un autre code dit d'embrouillage. Ce code fait partie des codes pseudo-aléatoires dont les propriétés d'intercorrélacion sont supérieures à celles obtenues par les seuls codes de canalisation. Ce sont ces deux mots « pseudo » et « aléatoire » qui confèrent à ce type de codes des propriétés assez intéressantes justifiant leur utilisation dans un système de radiocommunication. En effet, le code n'est pas complément aléatoire, car il est heureusement possible de le régénérer à la réception pour récupérer l'information transmise, pourvu que l'émetteur et le récepteur soient parfaitement synchronisés. Un registre à décalage binaire représente l'une des manières les plus courantes pour générer des codes pseudo-aléatoires. Comme le registre comprend « n » états, il est possible de générer « 2^n » codes pseudo-aléatoires. Il en résulte aussi que la période de la séquence n'est jamais supérieure à « $2^n - 1$ ». Une séquence binaire de période « $P = 2^n - 1$ », générée avec un registre à décalage de type *LFSR*, est appelée m-séquence ou encore, séquence à longueur maximale (*Maximal Length Sequence*). Afin d'améliorer plus les propriétés d'intercorrélacion des m-séquences tout en pouvant générer un nombre plus important de codes avec les mêmes propriétés pseudo-aléatoires, un autre type de codes a été proposé: les codes de Gold [9] [119].

J. Gold [119] propose d'additionner en modulo 2 deux m-séquences de mêmes longueurs, générées par deux polynômes primitifs différents. Si la période des deux séquences est « $2^n - 1$ », la séquence résultante aura aussi « $2^n - 1$ » comme période. Si nous considérons tous les décalages possibles entre les deux m-séquences, on peut générer un total de « $2^n + 1$ » codes (les « $2^n - 1$ » codes précédents plus les deux m-séquences génératrices). L'architecture du générateur du code de Gold utilisé dans le système W-CDMA est basée sur une combinaison de deux séries de registres linéaires à décalage (m-séquences). Ces registres ont une largeur 18 bits et sont combinés avec des portes logiques. Les deux polynômes utilisés pour générer ces deux m-séquences sont décrits dans les deux expressions suivantes [9]:

$$1+X^7+X^{18} \quad \text{A.2}$$

$$1+X^5+X^7+X^{10}+X^{18} \quad \text{A.3}$$

La figure A.5 montre l'implantation de l'architecture du générateur de code de Gold composée de deux m-séquences de 18 bits et de portes XOR [9] [75].

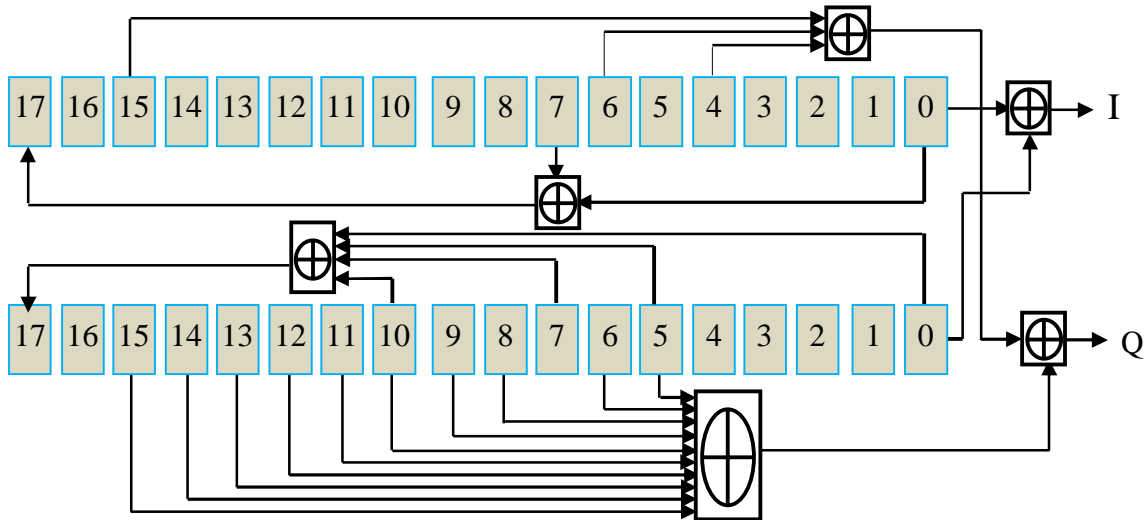


Figure A.5. Générateur de code de Gold.

A.6. Estimation de canal

Dans les systèmes de communication W-CDMA, pour combiner les différents trajets de façon raisonnable, le récepteur RAKE a besoin des paramètres de canal, par exemple: le nombre de trajets, leurs positions respectives (dans le temps) et leurs atténuations (en valeur complexe). Pour cela, les paramètres du canal doivent être estimés et surveillés pendant toute la durée de transmission. L'estimation de canal est une opération complexe qui détermine les coefficients complexes du canal (les paramètres du canal), c'est-à-dire la phase et l'atténuation de chaque composante des trajets multiples.

Annexe B

Système de transmission OFDM

B. Les systèmes OFDM

Les modulations multiporteuses OFDM ont été conçues sur le principe suivant: en émission, le signal fréquentiel est transmis sur un certain nombre N d'exponentielles complexes à des fréquences différentes appelées sous porteuses. On répartit ainsi un flux binaire (ou de symboles) à un rythme T_d sur l'ensemble de sous-porteuses ayant chacune un débit réduit. On obtient donc un symbole OFDM de durée $T_s = N \cdot T_d$ [120].

Cela pose deux obstacles pour la réalisation de la modulation multiporteuses OFDM; le premier est que le nombre N pouvant être très grand, donc il n'est pas envisageable de réaliser un système OFDM comportant N oscillateurs, mélangeurs, filtres, démodulateurs...

Le deuxième est que pour obtenir une bonne efficacité spectrale il faut rapprocher au maximum les sous-porteuses, donc la complexité des filtres devant séparer les N sous-porteuses devient intolérable.

Pour remédier à ces deux problèmes, il a été montré que l'opération de transformée de Fourier permet de simplifier la génération d'un signal OFDM si l'espacement fréquentiel entre les sous-porteuses est correctement choisi.

La sous-porteuse k d'un système OFDM peut s'écrire de la manière suivante:

$$\Psi_k(t) = \exp\left(j \cdot 2\pi \cdot k \cdot \frac{t}{T_s}\right) \quad \text{B.1}$$

Si deux sous porteuses Ψ_k et Ψ_m sont espacées de $1/N \cdot T_d$, elles sont orthogonales au sens mathématique et vérifient alors la propriété suivante:

$$\int_{\tau}^{\tau+T_s} \Psi_k(t) \Psi_m^*(t) dt = 0 \quad \text{si } k \neq m$$
$$= N \cdot T_d \quad \text{si } k = m \quad \text{B.2}$$

Les porteuses sont donc conçues de manière à ce qu'elles soient orthogonales entre elles. Le spectre relatif à chaque porteuse voit, alors, ses zéros correspondre aux fréquences centrales des autres porteuses (cf. figure B.1). Le sinus cardinal s'annule pour les fréquences de la forme:

$$\Psi = \Psi_i \pm \frac{k}{T_s} \quad \text{B.3}$$

Cela leur permet d'être placées très proches les unes des autres sans interférence. (cf. figure **B.1**). En plus il est donc en théorie possible de démoduler un signal OFDM sans filtrage explicite, si l'espacement entre les sous-porteuses est respecté.

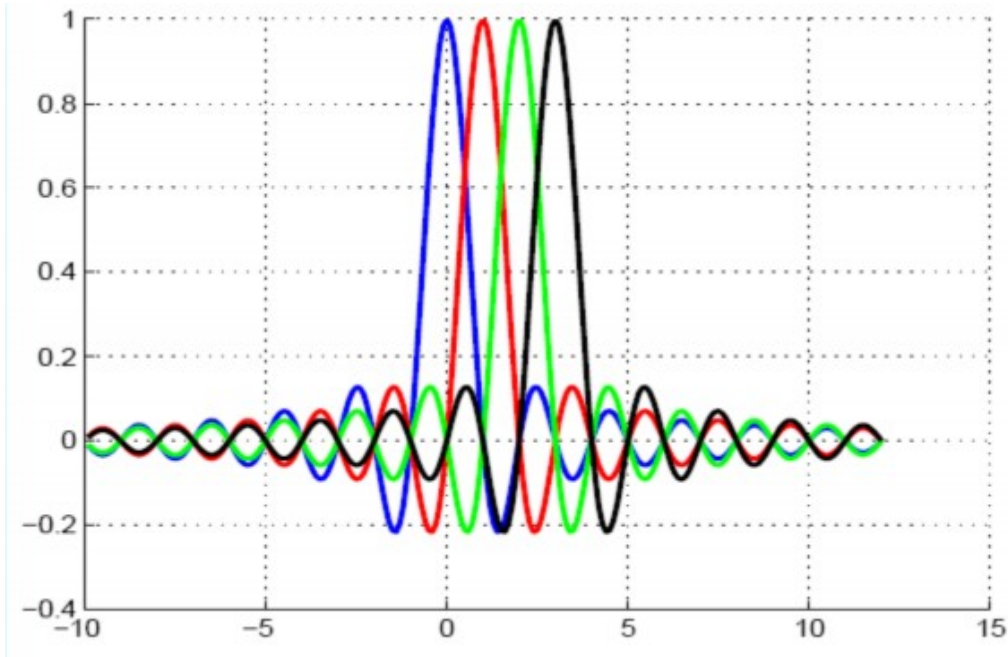


FIGURE B.1 : Spectres de quatre porteuses orthogonales.

L'expression du signal OFDM sur l'intervalle temporel $[0, T_s]$ peut s'exprimer comme [9][120]:

$$x(t) = \frac{1}{N} \sum_{k=0}^{N-1} C_k e^{j2\pi k \Delta f t}, \quad 0 \leq t \leq T_s \quad \mathbf{B.4}$$

Où C_k est le symbole numérique, qui a été choisi de l'ensemble de constellation de la modulation numérique de type M-QAM, modulé par la $k^{\text{ème}}$ sous porteuse, Δf c'est la différence de fréquence entre les sous-porteuses et il est mis à $\Delta f = 1/T_s$ afin de produire des sous porteuses orthogonales, où T_s est la durée de symbole OFDM. On peut remplacer la variable t en (B.4) par la variable n/N , nous obtenons la relation suivante:

$$x(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} C_k e^{j2\pi k \Delta f \frac{n}{N}}, \quad 0 \leq n \leq N - 1 \quad \mathbf{B.5}$$

Le terme en droite dans l'équation (B.5) c'est exactement l'opération IFFT (Inverse Fast Fourier Transform) [90]. Où

- k l'indice de sous-porteuse (domaine fréquentiel).
- C_k le symbole modulé sur la sous-porteuse k .
- N la taille de la FFT.

Par convention, l'expression du signal OFDM est normalisée par l'inverse de la racine carrée de N .

B.1. La chaîne de transmission OFDM

Le diagramme en bloc de la chaîne de transmission OFDM est représenté en figure B.2. Cette chaîne de transmission est constituée des éléments suivants:

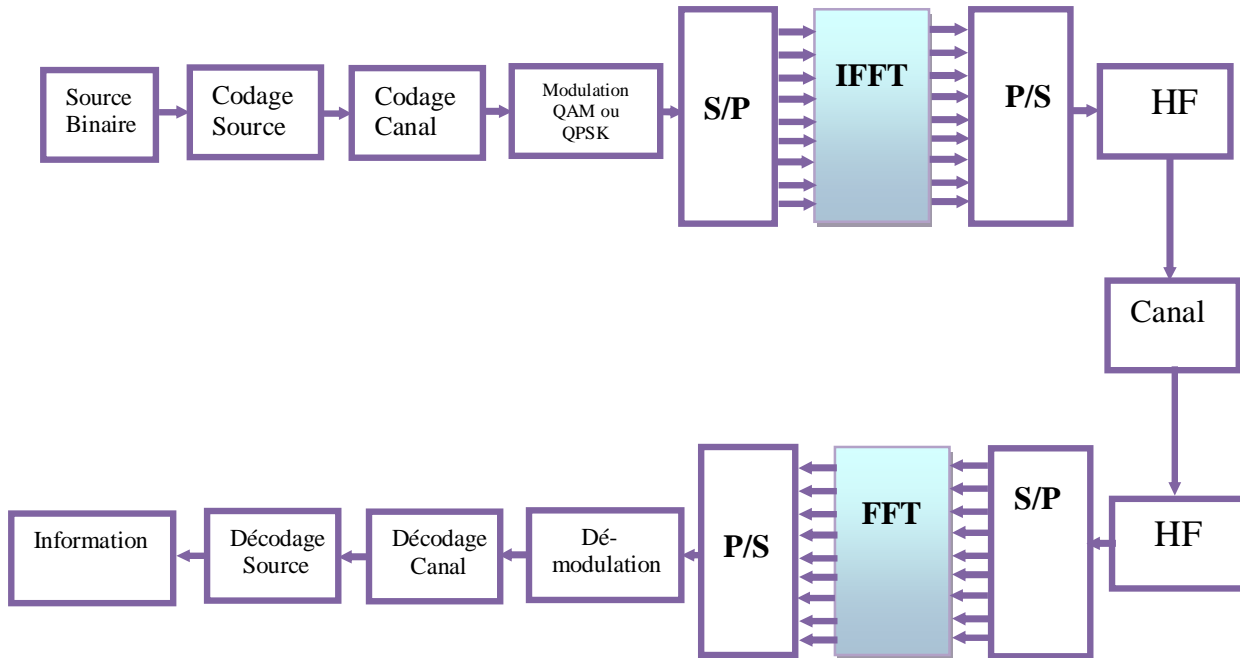


Figure. B.2– Diagramme en bloc de la chaîne de transmission OFDM.

- **Source d'information:** ce sont par exemple les capteurs qui transforment un signal physique d'information en un signal électrique. Le signal électrique est ensuite transformé en séquences de bits d'information.
- **Codage source:** cette étape représente la compression des données qui enlève la redondance dans les bits d'information.
- **Codage canal:** cette étape va maintenant ajouter de la redondance dont le but de protéger les informations.
- **Modulation binaire/M-aire (Mapping):** Le flux binaire (b_0, b_1, \dots), dont le temps bit est égal à T_d , est regroupé par paquets de l bits : 1, 2, 4 ou 6 en fonction de la modulation numérique employée (B-PSK, Q-PSK, 16-QAM ou 64-QAM respectivement). Par conséquent des symboles complexes ($c_k = a_k + jb_k$) de durée $T_C = l \times T_B$ sont générés suivant un alphabet donné, en modifiant l'amplitude et/ou la phase des symboles en fonction des paquets de bits d'information.
- **Modulation OFDM:** le schéma classique d'une modulation multiporteuses comprenant une conversion série/parallèle, une IFFT et une conversion parallèle/série. En premier

lieu, les données d'entrée sont converties en parallèle par un convertisseur série/parallèle (le bloc S/P). Ensuite, les N parallèles sorties du bloc S/P alimentent les entrées de bloc IFFT (Inverse Fast Fourier Transform) afin de créer un symbole OFDM, parfois appelé le modulateur OFDM. Après, les N nombres complexes à la sortie du bloc IFFT sont converti en série par un bloc convertisseur parallèle-série (block P/S). Ensuite, le préfixe cyclique est inséré afin de lutter contre l'interférence entre symboles (ISI) et l'interférence entre porteuses (ICI) provoqué par le canal de propagation à trajets multiples.

- **Insertion du préfixe cyclique ou du zero padding:** on ajoute une redondance ou des zéros à chaque symbole OFDM permettant d'éviter l'interférence entre symboles OFDM utile. Ce préfixe cyclique est parfois appelé l'intervalle de garde. Afin de créer le préfixe cyclique, le vecteur complexe de longueur Δ à la fin de la durée de symbole T_s est copiée et ajoutée à l'avant du bloc de signal [9] [120]. En pratique, le préfixe cyclique est choisi pour être plus long que le retard maximum de propagation du canal.

- **Conversion numérique/analogique:** le signal numérique est converti en un signal analogique.

- **Filtrage:** le signal électrique bande de base est filtré afin de supprimer les répétitions du spectre obtenues lors de la conversion numérique/analogique.

- **Transposition:** Le spectre du signal utile est transposé vers une fréquence plus haute par le bloc Upconverter.

- **Amplificateur de puissance:** la puissance du signal est augmentée pour qu'il puisse résister à l'atténuation du canal.

- **Antenne d'émission:** le signal électrique est transformé en une onde électromagnétique pour la propagation dans le canal.

- **Canal de propagation:** le canal de propagation dans le cas des télécommunications mobiles introduit plusieurs sortes de distorsions comme l'effet Doppler ou l'effet multi-trajets.

- **Antenne de réception:** l'onde électromagnétique est transformée en un signal électrique.

- **Amplificateur faible bruit:** le signal qui a subi l'atténuation du canal est amplifié.

- **Transposition:** le spectre du signal utile est transposé vers une fréquence plus basse par le bloc Downconverter,

- **Filtrage:** le signal électrique bande de base est filtré afin d'éviter le repliement spectral lors de l'échantillonnage effectué par la conversion analogique/numérique.

- **Conversion analogique/numérique:** le signal électrique analogique est converti en un signal numérique.
- **Suppression du préfixe cyclique ou du zero padding.**
- **Démodulation OFDM:** cette opération est réalisée par la FFT. Le schéma classique d'une démodulation multiporteuses comprenant une conversion série/parallèle, une FFT et une conversion parallèle/série.
- **Estimation et égalisation:** la dispersion du canal est estimée grâce à des symboles connus du récepteur. Les symboles reçus affectés par le canal sont ensuite compensés.
- **Démodulation M-aire/binaire:** les symboles reçus sont reconvertis en paquets de bits.
- **Décodage canal :** C'est la correction de certaines erreurs en utilisant la redondance ajoutée au niveau de l'émetteur.
- **Décodage source:** Les données sont ensuite décompressées en insérant les redondances enlevées lors du codage source à l'émission.
- **Informations:** les données sont transformées de forme électrique en forme physique.

B.2. Avantages et inconvénients de l'OFDM

Les avantages de l'OFDM sont nombreux:

- l'utilisation de la bande de fréquence allouée est optimale par orthogonalisation des porteuses.
- la modulation est basée sur un algorithme bien connu: la FFT.
- l'OFDM permet une égalisation simple grâce à l'ajout du "préfixe cyclique" ou du "zéro padding", même en présence de canaux multi-trajets denses.

L'OFDM possède néanmoins des inconvénients qu'il est important d'appréhender:

- l'orthogonalité des sous-porteuses est l'élément clef de la modulation OFDM. Le bruit de phase ou le désaccord en fréquence entre les oscillateurs locaux de l'émetteur et du récepteur (appelé offset fréquentiel) impliquent une perte d'orthogonalité entre sous porteuses et une forte dégradation des performances du système.
- si le récepteur OFDM est mal synchronisé temporellement, un phénomène d'interférence entre symboles OFDM peut intervenir dégradant considérablement les performances du système global.

Annexe C

Conversion d'un signal modulé en un signal binaire

C. 1. Modulation de fréquence (FSK)

Un signal modulé en fréquence répond à l'équation

$$S(t) = A \cdot \cos(2\pi f_q \cdot t + \phi_0) \quad \text{C.1}$$

Avec $q \cdot \Delta t \leq t < (q + 1)\Delta t$.

Dans cette équation Δt est la durée d'un symbole FSK, ϕ_0 le déphasage du signal et f_q la fréquence du signal pour ce symbole.

Pour la conversion du signal modulé en un signal binaire dont l'amplitude ne peut prendre d'autres valeurs que 0 ou 1 on utilise un système à détection de seuil intégré au convertisseur analogique numérique. Par conséquent, le signal devient [93][109]:

$$S'(i) = \begin{cases} 1, & \text{si } \frac{1}{2\pi f_q} (2m\pi - \phi_0) \leq t < \frac{1}{2\pi f_q} ((2m + 1)\pi - \phi_0) \\ 0, & \text{dans les autres cas} \end{cases} \quad \text{C.2}$$

Où $m \in \mathbb{Z}$.

Le résultat de cette transformation est présenté sur la figure C.1.

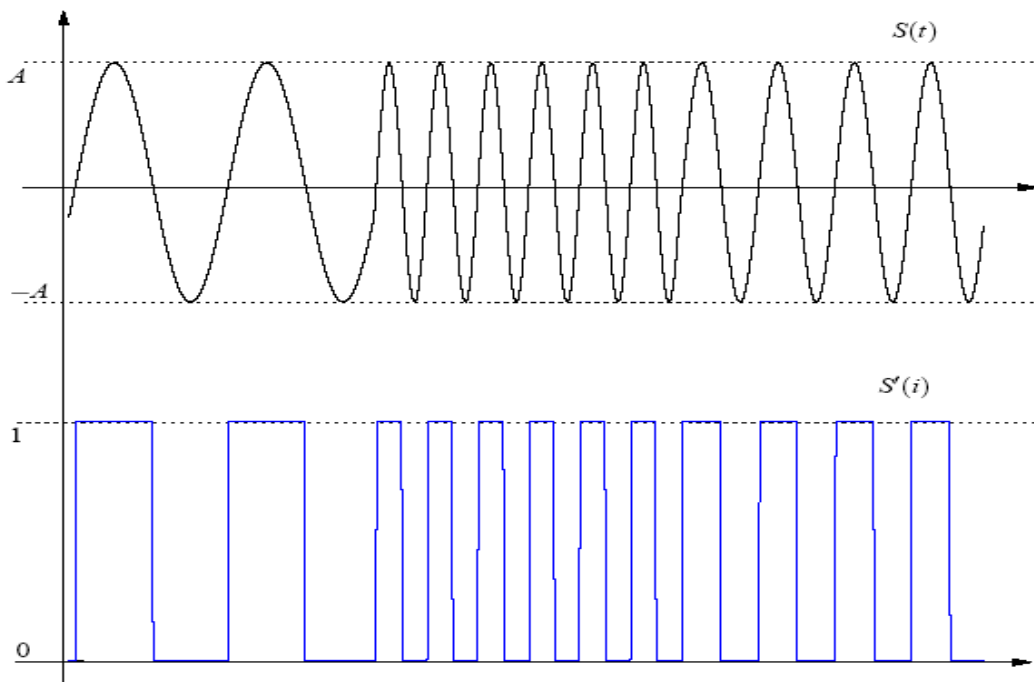


FIG. C.1: Normalisation d'un signal M-FSK.

C.2. Modulation de phase (PSK)

Un signal modulé M-PSK répond à l'équation :

$$S(t) = A. \cos(2\pi f. t + \phi_q) \quad \text{C.3}$$

Avec $q. \Delta t \leq t < (q + 1)\Delta t$.

Dans cette équation Δt est la durée d'un symbole M-PSK, ϕ_q la phase associée à ce symbole et f la fréquence du signal.

Après traitement par le système à détection de seuil, le signal devient [93]:

$$m(n) = \begin{cases} 1, & \text{si } \frac{1}{2\pi f}(2k\pi - \phi_q) \leq t < \frac{1}{2\pi f}((2k + 1)\pi - \phi_q) \\ 0, & \text{dans les autres cas} \end{cases} \quad \text{C.4}$$

Où $k \in \mathbb{Z}$.

Le résultat de cette transformation est présenté sur la figure C.2.

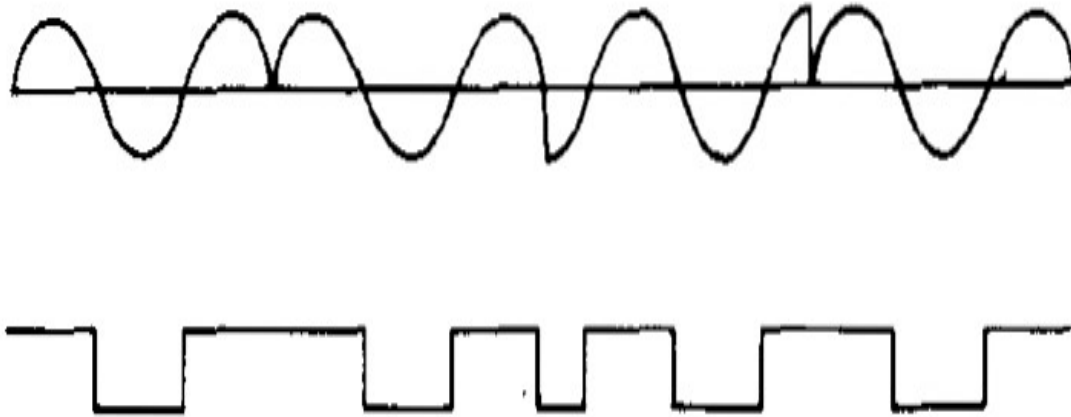


FIG. C.2: Normalisation d'un signal M-PSK.