

Ministère de l'enseignement supérieur et de la recherche scientifique



**UNIVERSITE SAAD DAHLEB DE BLIDA I**

**Institut d'aéronautique et des études spatiales**

**Spécialité : Navigation aérienne**

**Option: CNS / ATM**

**Thème :**

**Etude, comparaison et implémentation des détecteurs dédiés à l'acquisition d'un signal GNSS.**

En vue d'obtention du diplôme de Master.

**Présenté par :**

Lakel Bilal Nabil

**Promotrice :** Dr. S. Dehouche

**Co-Promoteur :** Pr. M. Hamadouche

**- Blida 2019 -**

*L'homme raisonnable s'adapte au monde ; l'homme déraisonnable s'obstine à essayer d'adapter le monde à lui-même. Tout progrès dépend donc de l'homme déraisonnable.*  
*George Bernard Shaw*

## Abstract

As part of this work, we have acquired a methodology for designing an SDR program and implementing it on an electronic map, in the field of satellite navigation of the constellation GNSS. The approaches used in this research allowed us to identify the different performance parameters of a software design and those of an implementation on an FPGA card, which can be generalized for several applications in communication and aeronautical navigation.

The example of application of the research focused on the analysis of the performance of the detection stage and the techniques of increases in the acquisition of a signal, a very important operation in the GNSS receiver, which determines reliability and availability of a system, key parameters for accuracy, safety and efficiency of a system in aeronautics.

## RESUME

Dans le cadre de ce travail, nous avons acquis une méthodologie de conception d'un programme en *SDR* et de son implémentation sur carte électronique, dans le domaine de la navigation par satellites de la constellation *GNSS*. Les démarches utilisées dans cette recherche nous ont permis d'identifier les différents paramètres de performance d'une conception logicielle et ceux d'une implémentation sur carte *FPGA*, pouvant se généraliser pour plusieurs applications en communication et navigation aéronautique .

L'exemple d'application des recherches s'est porté sur l'analyse des performances de l'étage de détection et techniques d'augmentations de l'acquisition d'un signal, une opération très importante dans le récepteur *GNSS*, qui détermine la fiabilité et la disponibilité d'un système, des paramètres clés pour la précision, la sécurité et l'efficacité d'un système en aéronautique.

## ملخص

كجزء من هذا العمل قد اكتسبنا منهجية لتصميم برمجيات محددة و تنفيذه على خريطة الكترونية , في مجال الملاحة الساتلية في نظام المجموعة العالمية *GNSS* . سمحت لنا المناهج المستخدمة في هذا البحث بتحديد مختلف الأداء و تطوير البرامج و بارامترات تنفيذ خريطة صيف البوابة القابلة للبرمجة التي يمكن تعميمها لعدة تطبيقات في مجال الإتصالات و الملاحة الجوية . ركز مثال تطبيق البحث على تحليل أداء مرحلة الكشف و تقنيات الزيادات في اكتساب اشارة , و هي عملية هامة جدا في مستقبل النظم العالمية لسوائل الملاحة التي تحدد الموثوقية و توافر النظام , بارامترات مفتاحية بدقة و سلامة و كفاءة نظام في مجال الطيران.

## REMERCIEMENTS

Je remercie d'abord ALLAH le bon dieu pour la vie qui m'a offert et qui m'as permis d'accomplir ce travail. Mes remerciements du fond du cœur à mes promoteurs Dr. DEHOUCHE.S et Prof. HAMADOUCHE. M qui m'ont orienté et booster le long de mon travail, un encadrement de qualité et des plus professionnelles, je vous dois beaucoup de respect et gratitude.

J'adresse ma reconnaissance, ma gratitude et mon respect à celui sans lequel cette thèse n'aurait pas eu lieu d'être, au Professeur Monsieur KAMEL BENACHENHOU, grâce à vous et à vos qualités techniques et humaines singulières j'ai pu m'épanouir professionnellement et espérer un avenir meilleur, un grand merci pour votre dévouement et votre soutien.

J'adresse également mes remerciements à mes parents et à ma sœur qui m'ont chéries et ont voulu toujours le meilleur pour moi, et qui ont veillé à ce que je ne manque de rien en dépit de leurs vie entière, vous êtes une bénédiction pour moi.

Je tiens à exprimer ma gratitude envers mes enseignants qui m'ont soutenu et qui m'ont toujours encouragé à aller de l'avant. Une grande estime pour eux qui ont une grande responsabilité.

Merci aux membres de ma famille, particulièrement ma grand-mère Beloui Aicha et mon Oncle Azouz Abderrahim, Un exemple pour moi d'humanité et de sagesse, mes chaleureuses pensées à vous.

Un Immense respect pour toi Ryad Abdelaziz pour l'ami, le frère l'exemple de sagesse et de persévérance que tu es, tu étais l'épaule sur laquelle je me suis accroché par les temps difficiles, et tu étais toujours présent lors des moments de plaisirs. Je termine par remercier mes amis qui ont été présent et qui m'ont soutenu et aider à la construction de ce mémoire Sidahmed Madani, Menira Abdelhamid, Islem Khenifer, Djamila Oubaiche, Sami Khellaoui, , Fares Kaouadji....ainsi qu'à tout mes collègues Damia, Soheib, Meriem....



# TABLE DES MATIERES

REMERCIEMENTS

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABREVIATIONS

LISTE DES SYMBOLES & NOTATIONS

INTRODUCTION GENERALE

## CHAPITRE I: Généralités sur la chaîne de transmission GNSS dédiée à l'acquisition

I.1. INTRODUCTION.....	18
I.2. Constellation.....	19
I.3. Principe de détermination de la position.....	20
I.4. SIGNAL GPS.....	23
I.4.1 Le signal au niveau du satellite.....	24
I.4.4 Propagation du signal dans l'espace.....	28
I.5. Evolution du signal dans le récepteur.....	33
I.6 Conclusion .....	43

## CHAPITRE II: L'ETAGE D'ACQUISITION D'UN SIGNAL GNSS

II. 1.INTRODUCTION .....	46
--------------------------	----

II. 2. Technologie FPGA du Xilinx.....	49
II. 2.1. La famille FPGA Xilinx Virtex-II.....	52
II. 2.2. FPGA Virtex-II Pro.....	53
II. 3. Flot de conception FPGA .....	56
II. 4. Architecture d'un Circuit FPGA.....	65
II. 5. Architecture d'un Circuit FPGA.....	66

**CHAPITRE III:                   Analyse de performance de l'étage de détection**

III. 1. INTRODUCTION.....	73
III. 2. Acquisition .....	74
III. 3. Analyse de la Corrélation .....	75
III. 3.3 Espace de recherche.....	79
III. 4. La détection du signal GPS.....	81
III. 4.1 Strategies de détection .....	82
III. 4.2 Les courbes ROC .....	82
III. 5. Détecteur et Probabilités.....	83
III.6. Intégration .....	90
III.7. CONCLUSION.....	98

**CHAPITRE IV: IMPLEMENTATION**

IV. 1. INTRODUCTION.....	102
IV. 2. Outils d'implémentation.....	102
IV. 3. DESCRIPTION DE LA CARTE Xilinx Virtex <u>6</u> .....	105
IV. 4. IMPLEMENTATION SUR FPGA.....	107
IV. 4. 1. Implémentation de l'acquisition avec détecteur quadratique.....	107
IV. 4. 2. Implémentation de l'acquisition avec intégration.....	103
IV. 5. Comparaison.....	119
IV. 6. CONCLUSION.....	120

## CONCLUSION GENERALE

### Annexe IV.1

### Annexe IV.2

#### LISTE DES FIGURES

Figure I.1 :	Illustration des principaux espaces de modélisations du signal	28
Figure I.2 :	Dispositions de satellites et la relation avec la précision de mesure	22
Figure I.3 :	répartition des fréquences pour le système GNSS	24
Figure I.4 :	Message de navigation d'un signal GPS	25
Figure I.5 :	Forme d'un code PRN	26
Figure I.6 :	Spectre de fréquence du code PRN	26
Figure I.7 :	Echantillon d'un signal GPS. Code PRN modulé par une porteuse.	27
Figure I.8 :	Gain de l'antenne d'émission	29
Figure I.9 :	Vitesse du satellite et effet doppler	31
Figure I.10 :	Fréquence Doppler causée par le mouvement du satellite	32
Figure I.12 :	Composantes d'un récepteur GNSS et leurs rôles dans la chaîne	33
Figure I.13 :	Architecture généralisée d'un étage radio fréquence d'un récepteur GNSS	35
Figure I.14 :	Rôle du mélangeur et du filtre passe bas	35
Figure I.15.a :	Figure de bruit du « FRONT END »	41
Figure I.15.b :	Figure de bruit du « FRONT END » avec une digitalisation direct	41

Figure I.16 :	Paramètres du « Front end »	42
Figure II.1 :	Comparaison des temps de routage entre circuits configurables	47
Figure II.2 :	Réseau d'aiguillage typique d'un FPGA	48
Figure II.3 :	Zones de congestion dans le routage des FPGA	49
Figure II.4 :	Organisation des FPGA de Xilinx	50
Figure II.5 :	Architecture simplifiée d'un « Slice ».	51
Figure II.6 :	Les modes de configuration des LUT	51
Figure II.7 :	Configuration des blocs d'E/S	52
Figure II.8 :	Détail de l'architecture d'un composant Virtex-II	53
Figure II.9 :	Vue Globale du VIRTEX-II PRO	54
Figure II. 10 :	Architecture standard autour du PowerPc405 sur VIRTEX-II PRO	55
Figure II.11:	Flot de conception générique	57
Figure II.12 :	Flot de synthèse sous Xilinx	59
Figure II.13 :	Implémentation sous Xilinx	60
Figure II.14 :	Le processus de vérification sous Xilinx.	62
Figure II.15:	Le processus « Back-Annotation ».	63
Figure II.16 :	Architecture interne du FPGA	66
Figure III.1 :	Schéma bloqué de l'étage de détection	73
Figure III.2 :	Schéma simplifié du bloqué d'acquisition	74
Figure III.3 :	Auto corrélation de deux codes PRN similaires décalés	75

Figure III.4 :	Inter corrélation de deux codes PRN différents	76
Figure III.5 :	corrélateur.	76
Figure III.6 :	Résultat obtenu en acquisition GPS	80
Figure III.7 :	Schéma bloc d'un détecteur linéaire	83
Figure III.9 :	ROC pour détecteur linéaire et quadratique (probabilité de détection en fonction de Pfa pour plusieurs SNR)..	88
Figure III.10 :	Probabilité de détection en fonction de SNR pour plusieurs Pfa	89
Figure III.11 :	Schéma d'un récepteur avec une intégration non cohérente	93
Figure III.12 :	Probabilité de détection en fonction de SNR pour Pfa=10 <sup>-5</sup> pour un détecteur avec intégration non cohérente	94
Figure III.14 :	<i>Détecteur différentiel cohérent</i>	95
Figure III.15 :	Courbe ROC pour le détecteur différentiel cohérent.	97
Figure III.16 :	Comparaison entre le détecteur non cohérent et le détecteur différentiel.	98
Figure IV.1 :	Flot de conception	103
Figure IV.2 :	Co-design	103
Figure IV.3 :	FPGA Virtex-VI ML605	106
Figure IV.4 :	Schéma modulaire du détecteur quadratique	109
Figure IV.5 :	Résultat de la simulation du détecteur quadratique	110
Figure IV.6 :	Bloc RTL Schematic principal	110
Figure IV.7 :	Description "RTL Shematic"	111
Figure IV.8 :	« Technologie Shematic » détecteur quadraique	112
Figure IV.9 :	détecteur série non cohérent	115

Figure IV.10 :	Résultat de la simulation du détecteur série non cohérent	116
Figure IV.11 :	Bloc “RTL Shemati”c principal	117
Figure IV.12 :	Description « RTL Shematic » du detecteur avec integration	117
Figure IV.13 :	« Thecnologie shematic » du detecteur avec integration	118

## LISTE DES TABLEAUX

Tableau I.1 :	Paramètres de la constellation GPS.	19
Tableau I.2 :	Bilan de liaison émission réception.	38
Tableau I.3 :	Bilan de liaison entrée sortie de l’étage RF et SNR à l’entrée de l’acquisition.	42
Tableau II.1 :	Famille FPGA Virtex-II	52
Tableau II.2 :	Ressources du circuit VIRTEX-II PRO/ VIRTEX-II PRO X	56
Tableau IV.1 :	Ressources utilisées pour détecteur quadratique.	108
Tableau IV.2 :	Ressources utilisées pour détecteur avec intégration	114

## LISTE DES SYMBOLES ET NOTATIONS

AWGN	Bruit blanc gaussien additive
BPSK	Binary Phase Shift Keying
$C/N_0$	Rapport signal sur bruit
C/A	Coarse Acquisition Code
CDMA	Code Division Multiple Access
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
UIT	Union international des telecommunication
LNA	Amplificateur faible bruit
SDN	Software Defined Navigator
SDR	Software Defined Radio

$A_i$	Amplitude du signal GNSS
$A_G$	Gain de l'amplificateur AGC
$az_k$	Azimut du satellite $k$
$B_{IF}$	Bande passante de l'étage radio fréquence
$\Delta b_r$	Biais d'horloge du récepteur
$\Delta b_{si}$	Biais d'horloge du satellite $i$
$C$	Vitesse de la lumière
$d_{euc}$	Distance euclidienne entre le satellite $i$ et le récepteur
$d_i(t)$	Données de navigation
$el_k$	Elévation du satellite $k$
$f_e$	Fréquence d'échantillonnage
$f_{d\min}$	Fréquence Doppler minimale dans l'espace de recherche
$f_{d\max}$	Fréquence Doppler maximale dans l'espace de recherche
$\Delta f$	Résolution fréquentielle dans l'espace de recherche
$\Delta f_f$	Décalage Doppler sur la fréquence de la porteuse
$\delta f_d$	Erreurs résiduelles entre le centre de la cellule et la vraie position de la fréquence Doppler
$\Delta f_f$	Décalage Doppler sur la fréquence de la porteuse
$f_E$	Fréquence du signal émis par le satellite
$f_c$	Fréquence du code d'étalement
$f_L$	Fréquence de la porteuse du signal GNSS
$f_{IF}$	Fréquence intermédiaire d'un récepteur GNSS
$f_{c,d}$	Fréquence du code d'étalement affecté par le Doppler
$f_d$	Décalage Doppler
$\hat{f}_d$	Décalage Doppler estimé
$f_0(s)$	Densité de probabilité de la variable $s$ sous $H_0$
$f_1(s)$	Densité de probabilité de la variable $s$ sous $H_1$
$G_t$	Gain de transmission

$G_r$	Gain de réception
$H_1$	Hypothèse alternative
$H_0$	Hypothèse nulle
$N$	Longueur de la période d'intégration
$N_\tau$	Nombre de cellules dans la direction du temps dans l'espace de recherche
$N_r$	Position du bit de transition
$N_f$	Nombre de cellules fréquentielles dans l'espace de recherche
$N_0$	Densité spectrale de puissance du bruit
$P_r$	Puissance reçue
$P_t$	Puissance transmise
$P_d$	Probabilité de détection
$P_{fa}$	Probabilité de fausses alarmes
$R_c(\hat{\tau}, \hat{f}_d)$	Fonction d'ambiguïté
$R_n(\hat{\tau}, \hat{f}_d)$	Fonction de corrélation en présence du bruit seul
$r[n]$	Signal échantillonné
$r_q[n]$	Signal quantifié
$r_{RF}$	Signal reçu au niveau de l'antenne du récepteur
$S(\hat{\tau}, \hat{f}_d)$	Variable de décision
$S_L(\hat{\tau}, \hat{f}_d)$	Variable de décision en considérant l'intégration non cohérente
$T_c$	Temps d'intégration
$T$	Longueur du code d'étalement
$T_s$	Période d'échantillonnage
$T_{ch}$	Durée d'un chip du code d'étalement
$t_r$	Temps de réception du signal au niveau du récepteur
$t_{si}$	Temps d'émission au niveau du satellite $i$
$V_r$	Composante radiale de la vitesse du satellite



$Y_I(\hat{\tau}, \hat{f}_d)$	Partie réelle de la variable de décision
$Y_Q(\hat{\tau}, \hat{f}_d)$	Partie imaginaire de la variable de décision
$\alpha$	Rapport d'amplitude multitrajet / signal direct
$\beta$	Seuil de détection fixe
$\eta_{RF}$	Bruit Gaussien à l'étage radio fréquence
$\eta_{IF}$	Le bruit à la sortie de l'étage $IF$
$\rho_i$	Pseudo-distance entre le récepteur et le satellite $i$
$\rho_{MS,k}$	Pseudo-distance séparant le récepteur du satellite $k$
$\sigma_n^2$	Variance du bruit reçu après corrélateur
$\tau$	Retard de propagation
$\hat{\tau}$	Retard de propagation estimé
$\delta\tau$	Erreurs résiduelles entre le centre de la cellule et la vraie position du code phase
$\varphi_i$	Phase du signal reçu
$\hat{\varphi}_i$	Phase du signal estimé

## INTRODUCTION GENERALE

En 1983 le conseil de l'OACI a affirmé que les systèmes et procédures supportés par l'aviation civile ont atteint leurs limites de performances et poseront problème sur la capacité de l'espace aérien international. Ce premier a créé le comité FANS pour des systèmes de navigations futurs. Le comité FANS proposa à son tour le concept CNS/ATM à la 10<sup>ème</sup> conférence de la navigation aérienne en 1991. Prévu pour déploiement en 2020, le concept CNS/ATM vise à utiliser plusieurs technologies de communications, de navigation et surveillance dans un système interopérable gérant le trafic aérien en toute transparence, pour un moindre coût et sans compromettre les niveaux de sécurité établis au préalable[15].

La mission du déploiement sera d'améliorer la sécurité, la régularité et l'efficacité des vols, augmenter la capacité et l'efficacité des espaces aériens en plus de minimiser les équipements. Cela est réalisable en maîtrisant les transferts d'informations, la précision de la navigation et l'étendue de la surveillance. D'autres bénéfices adviendront de ceux cités. Les systèmes de trafic aérien basés sur le CNS/ATM dépendent étroitement des satellites ! [15]

Le Radiorepérage par satellites consiste à déterminer la position, la vitesse ou d'autres caractéristiques d'un objet, ou obtention de données relatives à ces paramètres, et cela à l'aide des propriétés de propagation des ondes radioélectriques. Le système *GNSS* est un système à *Diffusion* qui transmet des renseignements concernant la navigation aérienne. Les satellites *GNSS* émettent en continu des signaux que le récepteur exploite pour extraire leur temps de parcours ainsi qu'un certain nombre d'informations utiles. Afin de répondre aux nouvelles exigences des récepteurs *GNSS* en termes d'interopérabilité, l'implémentation logicielle s'exécutant sur ordinateur ou sur carte électronique est la nouvelle tendance des recherches.

L'Acquisition dans les systèmes *GNSS*, étant la 1ère étape dans le traitement logiciel, et qui consiste à déterminer la présence des signaux des satellites visibles aux récepteurs et de donner une rugueuse estimation des paramètres du signal (décalage temporel et fréquentiel) requises pour l'estimation de la position. Afin de déterminer les paramètres de performance de l'acquisition et les limites des techniques utilisées, il faudra modéliser le signal le long de ses transitions dans le récepteur pour ensuite établir un programme de détection fiable, en temps pratique selon les besoins de notre application de navigation aéronautique.

## But et approche pédagogique du projet

Le but de ce travail de PFE est l'évaluation de performance des différentes techniques d'acquisition, plus précisément la partie détection, afin de tirer des conclusions sur les avantages et les inconvénients associés à chaque méthode. Par la suite, on réalise ces détecteurs afin de d'établir un bilan lié à leur implémentation.

L'application d'un tel projet a nécessité, d'abord une orientation de l'étude par apport aux besoins de l'aviation civile et aux performances requises RNP du projet CNS/ATM de l'OACI ainsi qu'aux besoins futurs de la radionavigation, et cela en termes d'interopérabilité car de nos jours des milliers de satellites sont en orbite autour de la Terre. Parmi eux, nous pouvons citer les satellites des constellations **GPS** Américain, **GLONASS** Russe, **Galileo** Européen, **Beidou** Chinois. Avec plusieurs signaux différents l'implémentation devient un défi vu la complexité des fonctions à exécuter. Sous contraintes temporelles, une réalisation d'un tel système demande un travail laborieux et très méthodique.

Deux approches logiques seront utilisées :

La première serait de présenter les composantes du récepteur *GNSS* et les paramètres recherchés pour l'acquisition du signal, et cela de façon générale, permettant ainsi de bien définir le but du travail et schématiser la chaîne de transmission ainsi que déterminer l'emprise de chaque espace de modélisation sur le signal.

- ✓ La deuxième approche consiste en l'étude des traitements et opérations élémentaires sur le signal séparément dans chaque bloc. Cela nous aidera à mettre en évidence les opérations liées directement ou indirectement aux performances du système d'acquisition et de choisir ensuite la meilleure adaptation selon les performances requises pour notre application. Ainsi, le signal va être poursuivi de sa génération puis sa propagation et enfin dans les phases du traitement par le récepteur.

Dans ce sens la structure de ce travail s'est faite sous quatre chapitres :

- ❖ Dans le premier chapitre nous étudierons la chaîne de transmission des signaux GPS en nous intéressant au signal et ses paramètres jusqu'au bloc d'acquisition.

- ❖ Le deuxième chapitre est une introduction aux FPGA's et leur caractéristiques. La connaissance des performances des cartes électroniques ainsi que les méthodes de conception est primordiale pour leur exploitation à juste valeur.
- ❖ Dans le troisième chapitre nous entamerons l'analyse des performances de l'acquisition en définissant le choix de la stratégie à entreprendre pour la simulation de notre récepteur en mettant en évidence les contraintes auxquelles on peut s'attendre avant l'implémentation.
- ❖ Le quatrième chapitre est dédié à la l'implémentation sous logiciels de notre acquiiteur, ou nous analyserons les difficultés d'une conception en apportant les conclusions nécessaires pour notre maitrise.

# *Chapitre I*

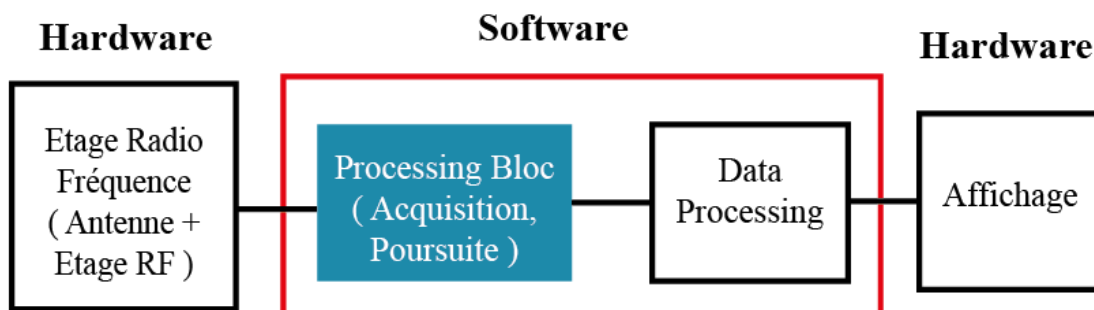
## *Généralités sur la chaîne de transmission*

### *GNSS dédiée à l'acquisition*

- ❖ **Introduction**
- ❖ **Constellation**
- ❖ **Principe de détermination de la position**
- ❖ **Modélisation du signal**
- ❖ **Conclusion**

## I.1. INTRODUCTION

Le concept traitement radio software est réalisé à l'aide d'un convertisseur analogique numérique. Une fois l'obtention d'un signal numérique, les données sont traitées pour recueillir les informations recherchées [3]. L'avantage primordial du traitement soft est de minimiser les composants hardware ou un récepteur de plusieurs signaux, issus de systèmes différents, encombrant un système hard. D'autres avantages peuvent être cités : comme pouvoir simuler la chaîne de transmission sur logiciel afin d'analyser les performances du système et exercer plusieurs tests et visualiser les schémas d'implémentation pour améliorer le système avant conception, mais aussi enregistrer des séquences de signaux reçus.



Les généralités, dans ce chapitre, porteront attention sur les paramètres clés pris en compte pour l'analyse de l'acquisition du signal GPS, et les conséquences de la transition du signal à travers le canal de transmission et les composantes électroniques du récepteur, sur ces paramètres. Ainsi, nous pourrions passer à l'étape suivante qui consiste à choisir l'architecture de notre système numérique d'acquisition, et entreprendre l'analyse de la stratégie à modéliser. On s'intéressera au traitement du récepteur à savoir les opérations accomplies pour accommoder le signal au bloc d'acquisition visé par l'étude, en détaillant les transformations subies par le signal.

Ainsi, nous entamerons avec le principe de positionnement de la constellation *GNSS*, puis nous introduirons les éléments de la transmission d'un signal *GNSS* dans le but de relever les paramètres clés de l'acquisition. Pour cela on prendra cas du signal *GPS* afin d'évaluer un signal concret. Ce chapitre comprendra les généralités du système *GPS*, notamment la structure du signal *GPS* et ses caractéristiques. Puis on évoquera la transmission du signal, dans le but d'établir un *bilan de liaison*.

## I.2. Constellation

Il y a un total de 24 satellites *GPS*, divisés en six orbites, chaque orbite a quatre satellites. Chaque orbite fait un angle de 55 degrés avec l'équateur, c'est l'*inclinaison*. Les orbites sont séparées de 60 degrés pour couvrir 360 degrés. Le rayon de l'orbite du satellite est de 26 560 km et il tourne autour de la terre deux fois par *jour sidéral* [1]. À tout moment et en tout lieu sur la terre, en négligeant des obstacles tels que les montagnes et les immeubles de grandes hauteurs, un récepteur *GPS* doit avoir une visibilité directe, et recevoir des signaux de 4 à 11 satellites [2].

Le rayon de la terre est de 6 378 km autour de l'équateur et de 6 357 km en passant à travers les pôles, et le rayon moyen peut être considéré comme 6 368 km. Le rayon de l'orbite du satellite est de 26 560 km, ce qui correspond à environ 20 192 km (26560 - 6 368km) au-dessus de la surface de la Terre [1]. Cette hauteur est approximativement la distance la plus courte entre un utilisateur sur la surface de la terre et un satellite, qui se produit vers *le zénith* ou une élévation angle d'environ 90 degrés [1]. La distance depuis un satellite vue *d'horizon* par l'utilisateur est de 25 785 km [1].

On peut voir que les retards de la transmission satellites-terre sont dans la plage de 67 ms (20 192 km /  $c$ ) à 86 ms (25 785 km /  $c$ ), où  $c$  est la vitesse de la lumière. Si l'utilisateur est à la surface de la terre, le temps de retard différentiel maximal de deux satellites différents doit être compris entre 19 (86-67) ms.

**Tableau I.1** : paramètres de la constellation *GPS*.

### *Constellation*

Nombre de satellites	24
Nombre de plans orbitaux	6
Inclinaison de l'orbite	55°
Rayon de l'orbite	26560 km
Période	11hrs 57min 57.26 sec
Distance la plus proche de la surface	20192 km
Distance la plus longue de la surface	25785 km
<b>Temps de propagation</b>	<b>67-86 ms</b>

*Source: Fundamentals of Global Positioning System Receivers: A Software Approach*

James Bao-Yen Tsui

L'estimation du temps de propagation s'interprète, géométriquement, par une distance entre le satellite et le récepteur, une donnée nécessaire pour le calcul de la position. Le signal émis du satellite, contenant les informations requises pour le calcul de position (I.4.1), est porté par un code d'une période d'une milliseconde (I.4.3) Doté de 1023 bits sur une fréquence d'un Méga Hertz. Ainsi, connaître la position du 1<sup>e</sup> bit sur le code reçu indique le retard sur une milliseconde de plus que l'estimation initiale du temps de propagation est donc une estimation précise de la distance. Sachant que 1 milliseconde d'erreur a pour effet 300 km d'erreur sur la position, le système doit être précis et fiable (III.3.1) (III.3.2).

### **I.3. Principe de détermination de la position**

En deux dimensions, l'estimation d'une distance à partir d'un point fixe est représentée par un cercle. Avec deux points et deux distances, l'intersection entre deux cercles donne deux points, un troisième point et une troisième distance sont nécessaires pour déterminer une solution unique : un point. De même dans les systèmes de positionnement par satellites en trois dimensions, l'intersection de deux sphères donne un cercle, le cercle intersecté avec une autre sphère donne deux points. L'un des deux points est considéré obsolète par le système géométrique vu que l'un sera positionné sur la surface terrestre tandis que l'autre sera beaucoup plus loin dans l'espace [2].

Dans la technologie *GPS*, la position du satellite est connue à partir des données des éphémérides transmises par le satellite. La distance mesurée entre l'utilisateur et le satellite est supposée être très précise et non biaisée. Cependant, la distance mesurée entre le récepteur et le satellite a un biais inconnu constant, parce que l'horloge de l'utilisateur est généralement différente de l'horloge des satellites. Afin de résoudre cette erreur de biais, un satellite supplémentaire est requis [8].

Le principe pour déterminer la position du récepteur est d'initialiser une position qui pourrait être le centre de la terre par exemple, puis, effectuer des itérations aux équations de pseudo distances.

#### **I.3.1. Calcul des distances**

L'estimation de la distance via le système *GNSS* repose sur la mesure du temps de propagation, de l'émission à la réception du signal électromagnétique. La mesure du temps de propagation du signal provenant de plusieurs satellites permet, par multilatération, de



déterminer la position du récepteur. Le signal voyageant à la vitesse de la lumière  $c$ . La distance recherchée est donnée par :

$$R = c \times T \quad (\text{I.1})$$

Chaque satellite envoie un signal à un instant  $t_{si}$ . Le récepteur reçoit le signal à l'instant  $t_u$  après une certaine durée. La distance entre le satellite  $i$  et le récepteur est :

$$\rho_{iT} = c \times (t_u - t_{si}) \quad (\text{I.2})$$

Où  $c$  est la vitesse de la lumière ;  $\rho_{iT}$  est souvent désignée comme la vraie valeur de pseudo-distance d'un utilisateur à un satellite  $i$ ,  $t_{si}$  est appelée l'instant réel de transmission depuis le satellite  $i$ ,  $t_u$  est l'instant de réception réel. Du point de vue pratique, il est difficile, voire impossible, d'obtenir l'heure correcte du satellite ou de l'utilisateur. L'heure satellite d'horloge réelle  $t'_{si}$ , et l'instant d'horloge utilisateur réel  $t'_u$  est lié à l'instant réel par :

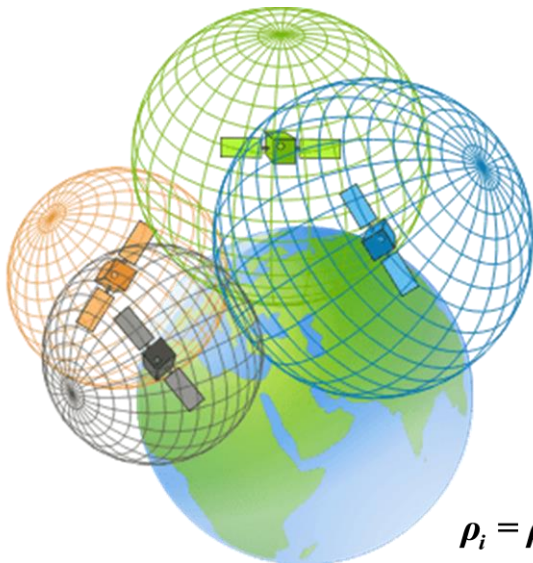
$$t'_{si} = t_{si} + \Delta bi \quad (\text{I.3})$$

$$t'_u = t_u + but$$

Où  $\Delta bi$  : l'erreur d'horloge du satellite ;  $but$  : l'erreur d'horloge du récepteur. D'autres facteurs peuvent affecter la mesure de la pseudo-distance dont son équation est la suivante [1] [2] [3] [15][19] :

$$\rho_i = \rho_{iT} + \Delta D_i - c(\Delta bi - but) + c(\Delta T_i + \Delta I_i + v_i + \Delta v_i) \quad (\text{I.4})$$

Où  $\Delta D_i$  : l'effet de l'erreur de position du satellite sur la distance ;  $\Delta T_i$  : Le retard du à la troposphère ;  $\Delta I_i$  : Le retard du à l'ionosphère ;  $v_i$  : Bruit ;  $\Delta v_i$  : Correction temps.



Pseudo distance

$$\rho_{iT} = c \times (t_u - t_{si})$$

Biais

$$t'_{si} = t_{si} + \Delta bi$$

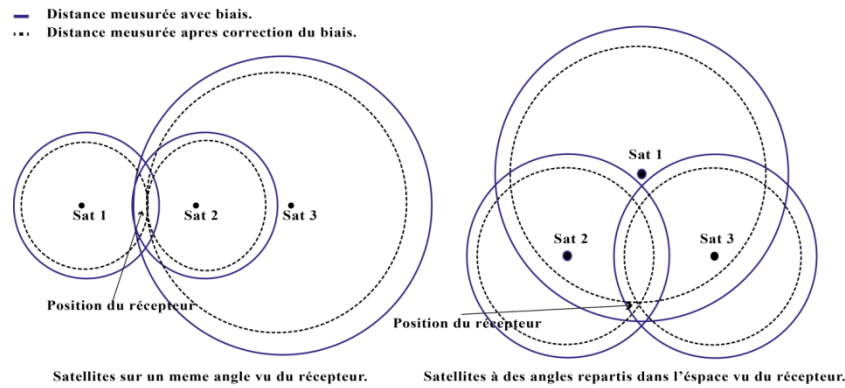
$$t'_u = t_u + but$$

Pseudo distance corrigée

$$\rho_i = \rho_{iT} + \Delta D_i - c(\Delta bi - but) + c(\Delta T_i + \Delta I_i + v_i + \Delta v_i)$$

### I.3.2. La Dilution de Précision (DOP)

Un récepteur peut recevoir des signaux de plus de quatre satellites. Pour calculer la position, on pourrait éventuellement utiliser tous les signaux, cela dit, les positions des combinaisons des satellites influents sur la précision des mesures, cela est dû à leurs positions dans l'espace [1].



**Figure I.2:** Dispositions de satellites et la relation avec la précision de mesure.

La **Figure I.2** montre deux combinaisons possibles et illustre leurs impacts sur la précision de la position. Les cercles en gras représentent la distance mesurée par le récepteur en considérant le biais d'horloge, et les cercles en pointillés représentent les cercles des distances après la correction du biais. Sur **Figure I.2** à gauche, l'erreur de la position est plus grande que celle de la figure à droite, car dans le 1<sup>er</sup> cas les positions des satellites sont sur une seule ligne, tandis que dans l'autre figure, les satellites sont répartis avec des angles bien clairs, et il est difficile de mesurer la position avec précision à partir de points tangentiels. On peut également mesurer les triangles formés par les satellites, la plus grande surface de ces triangles détermine la position la plus proche du réel. Donc dans un cas en 3D, il est préférable de maximiser le volume défini par les satellites pris en compte pour le calcul de la position du récepteur.

*The Dilution Of Precision DOP* est utilisé pour évaluer la précision des mesures. Plusieurs définitions existent pour le *DOP*, et qui sont en fonction de la géométrie des satellites dans l'espace de tel que leurs positions déterminent les valeurs des *DOP's* [2] [19].

$$\left\{ \begin{array}{l} \text{GDOP} = \frac{1}{\sigma} \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2 + \sigma_b^2} \\ \text{PDOP} = \frac{1}{\sigma} \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \\ \text{HDOP} = \frac{1}{\sigma} \sqrt{\sigma_x^2 + \sigma_y^2} \\ \text{VDOP} = \frac{\sigma_z}{\sigma} \\ \text{TDOP} = \frac{\sigma_b}{\sigma} \end{array} \right. \quad (\text{I.5})$$

Ou  $\sigma$  est la variance de l'erreur de la pseudo distance,  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_z$  correspondent aux erreurs de mesures dans les axes x,y,z.  $\sigma_b$  est la variance de l'erreur de l'horloge exprimée en distance. La plus petite valeur du *DOP* révèle la meilleure combinaison pour le calcul de la position avec précision.

#### I.4. Espaces de modélisation du signal

Le système *GPS* comprend trois segments : le segment de control, le segment espace et le segment utilisateur. Le segment de control comprend plusieurs stations au sol dont la station mère principale située à *Falcon Air Force Base, Colorado Springs, CO*, et qui traite les rapports de fonctionnement et positionnement des satellites, reçus des autres stations. Ainsi, la station mère gère les données qui vont être transmises aux satellites, ces données comprennent principalement la mise à jour des messages de navigation et corrections (orbite, temps, etc.)[4].

La répartition des satellites dans l'espace est faite en sorte qu'au minimum 11 satellites soient visibles, et que la constellation englobe la terre en termes de couverture, en plus de minimiser l'impact de panne d'un satellite sur les performances du système en termes de couverture et du *DOP* [1][2][4]. Au moins trois espaces de modélisations et de mesures des paramètres du signal sont nécessaires pour bien évaluer et expliquer les changements que subit le signal dans la chaîne de transmission, au niveau du satellite, de l'espace et du récepteur [4].

##### I.4.1. Le signal au niveau du satellite

Afin de comprendre les opérations établies sur le signal reçu, il est d'abord nécessaire de connaître sa forme, ses composantes, et ses caractéristiques au niveau du satellite émetteur.

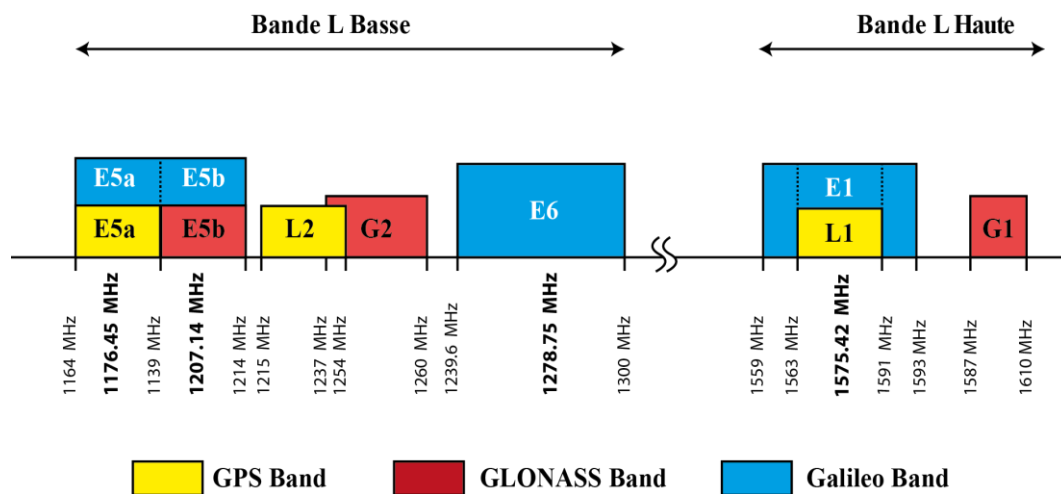
### ➤ Fréquence du signal

Le signal *GPS civile* est envoyé sur deux fréquences porteuses  $F_{L1}$  et  $F_{L2}$ .

$$F_{L1} = 154 F_0 = 1575.42 \text{ [MHz]}$$

$$F_{L2} = 120 F_0 = 1227.60 \text{ [MHz]}$$

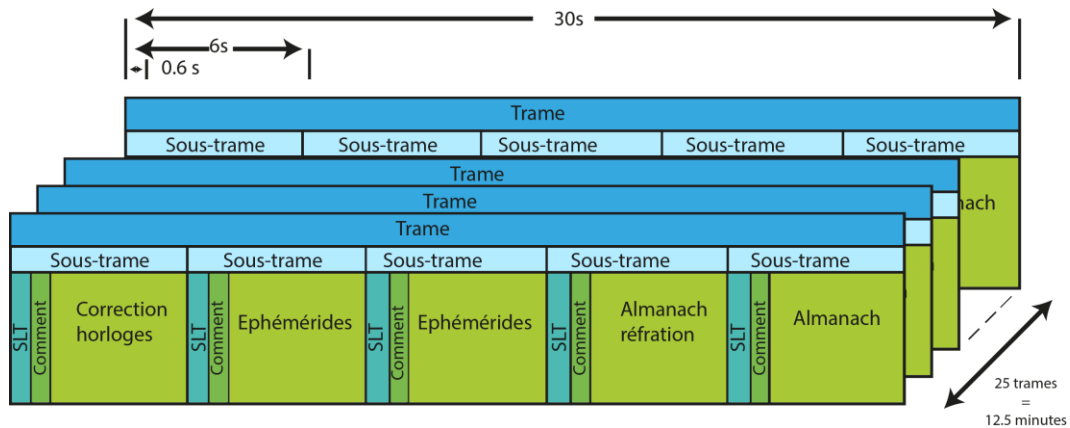
Ces fréquences sont en *UHF* () spécifiquement la bande L. La bande L est choisie car l'atmosphère est transparente pour la gamme *UHF* et *SHF*. Pour les gammes moindres, l'ionosphère devient de plus en plus opaque pour les ondes radio. Au-dessus de la bande X l'onde devient tellement petite qu'elle est d'autant susceptible d'être affectée par l'oxygène et la vapeur d'eau. A noter que la fréquence d'émission est très précise vu l'utilisation de l'horloge atomique au sein des satellites [8].



**Figure I.3 :** Répartition des fréquences pour le système GNSS.

### ➤ Donnée de navigation

Les informations contenues dans le message de navigation ont la structure de trames de base représentées ci-dessous



**Figure I.4 :** Message de navigation d'un signal GPS.

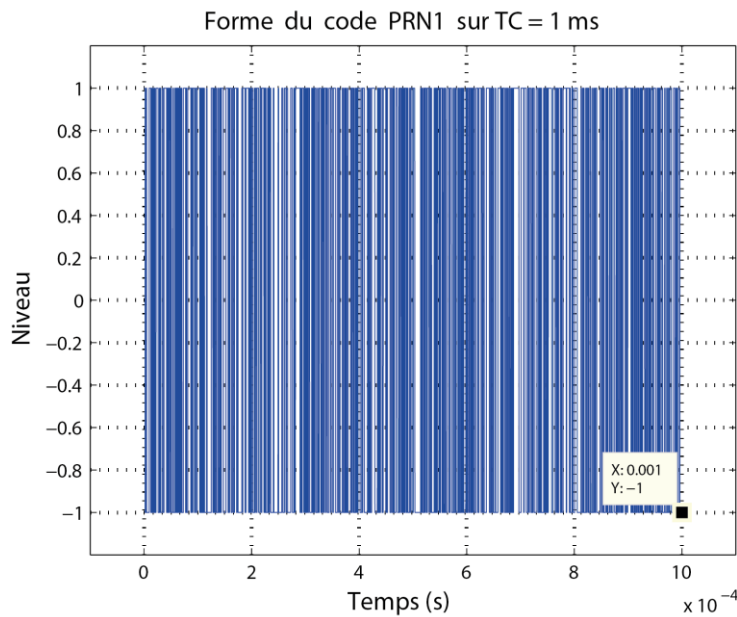
Un message complet se compose de 25 *trames*, chacune contenant 1500 bits à 50 Hz (ce qui signifie que l'on transmet 50 bits par secondes). Chaque trame est subdivisée en cinq *sous-trames* de 300 bits : 6sec et chaque sous-trame se compose de 10 *mots* de 30 bits : 0,6sec. Le premier mot de chaque sous-trame est *SLT* (*mot de télémétrie*), il contient des informations sur l'actualité des *éphémérides*. Il est suivi du mot *Comment* (*mot de transfert*) contenant le nombre *Z époques* compté. Ces données incluent le temps écoulé depuis le dernier redémarrage. La transmission du message complet de navigation de 25 trames nécessite 750s ou 12,5 min [3] [5].

Les données restantes de la première sous-trame contiennent des données sur l'état et la précision du satellite ainsi que les valeurs de correction d'horloge. Les deuxièmes et troisièmes sous-trames contiennent des paramètres de l'éphéméride, et les sous-trames 4 et 5 comprennent enfin les almanachs [3] [5].

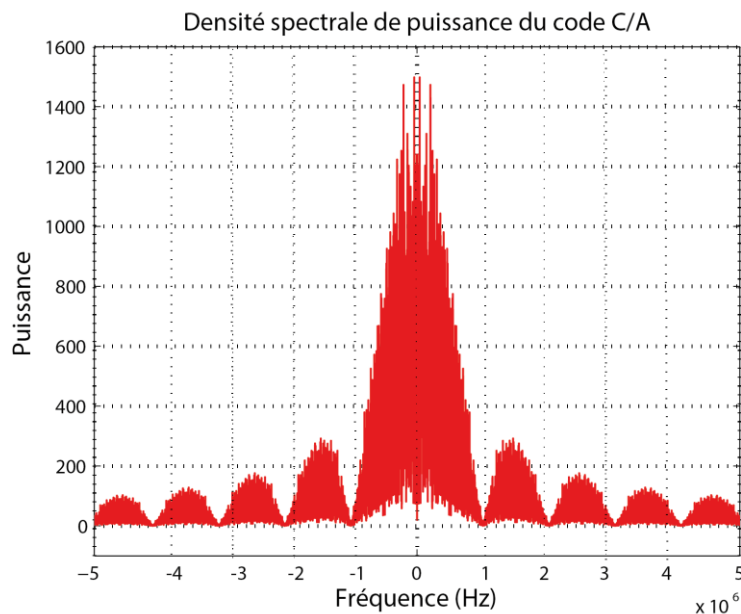
Les systèmes *GNSS* sont en général des systèmes *DSSS* « *DIRECT SEQUENCE SPREAD SPECTRUM* » étalement de spectre à séquence directe, une technique d'étalement de spectres utilisée dans le but du *DSSS* et d'une part est de rendre les signaux, occupants une bande de fréquence, plus résistants aux brouillages et aux interférences, et d'autre part de permettre à plusieurs équipements de partager la même fréquence porteuse et pour cela ils sont combinés à un code d'étalement. Les propriétés de corrélation des codes d'étalement permettent d'identifier le signal émis et de calculer les temps de transmission [3] [8] [1].

### ➤ Code C/A

Acronyme de « *Coarse Acquisition* », en français : « acquisition brute » est un code binaire pseudo- aléatoire (code qui revient périodiquement à la même structure de bit (un et zéro) ou *PRN* (*en anglais Pseudo Random Noise*) de débit 1,023 Mbit/sec de période 1 ms [3].



**Figure I.5 :** *Forme d'un code PRN.*



**Figure I.6 :** *Spectre de fréquence du code PRN.*

A partir du la **Figure I.6** on remarque que le lobe principal, qui contient une grande puissance est contenu dans une bande de 2 MHz.

La transmission des données se fait selon le principe *CDMA* (Code Division Multiple Access), c'est-à-dire que chaque satellite de la constellation dispose d'une clé (mot d'un code) à l'aide de laquelle il code l'information à transmettre. Connaissant la clé d'un satellite donné, le récepteur est capable d'extraire le message de ce satellite du signal composite reçu. Le message transmis résulte de l'addition modulo-2 (message de navigation et des codes). Avec le

code modulé ci-dessus la porteuse subit une modulation de type *BPSK* (*Binary Phase Shift keying*). Ainsi la porteuse fait un saut de phase de  $180^\circ$  à chaque changement d'état et codé en NRZ. Le signal GNSS reçu peut être écrit [1][2][3][19] :

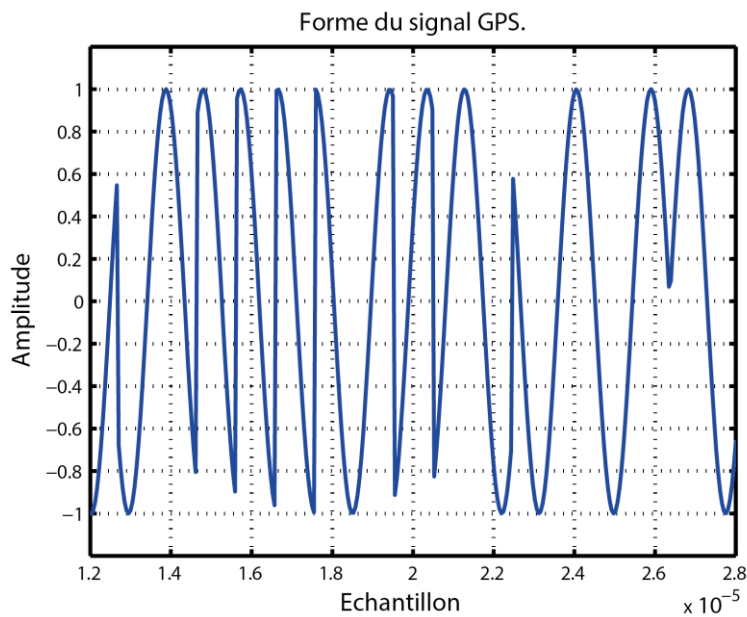
$$r_{RF}(t) = \sum_{i=1}^L y_i(t) + \eta_{RF}(t) \quad (\text{I.6})$$

Qui est la somme de  $L$  signaux différents pour  $L$  satellites émetteurs et  $n(t)$  désigne le bruit environnant.

$$y_i(t) = \sqrt{2P} \cdot \left( c_i(t) \oplus d_i(t) \right) \cdot \cos(2\pi F_{L1}t) + \sqrt{2P} \cdot \left( p_i(t) \oplus d_i(t) \right) \cdot \sin(2\pi F_{L1}t) + \sqrt{2P} \cdot \left( p_i(t) \oplus d_i(t) \right) \cdot \sin(2\pi F_{L2}t) \quad (\text{I.7})$$

$P$  : représente la puissance totale ;  $c_i$  est le code d'étalement du satellite  $i$  ;  $d_i$  représente le message de navigation ;  $p_i$  est le code d'étalement dédié au signal militaire appelé code P au lieu de code C/A [3]. Dans notre étude on prendra en considération le code C/A uniquement.

$$y_i(t) = \sqrt{2P} \cdot \left( c_i(t) \oplus d_i(t) \right) \cdot \cos(2\pi F_{L1}t) \quad (\text{I.8})$$



**Figure I.7 :** Echantillon d'un signal GPS. Code PRN modulé par une porteuse

On remarque qu'un changement de phase se produit à plusieurs reprises, signifiant un changement de signe du code *PRN* à chaque changement de phase de  $180^\circ$ , c'est une modulation *BPSK*.

### I.4.2. Propagation du signal dans l'espace

Dans cette section on parlera de la propagation du signal de l'émetteur au récepteur, afin d'établir un bilan de liaison de la transmission, cela nous permettra de déterminer la puissance reçue au niveau du récepteur, les changements subis par signal et de définir les éléments susceptibles d'atténuer le signal ou de modifier sa fréquence dans l'espace.

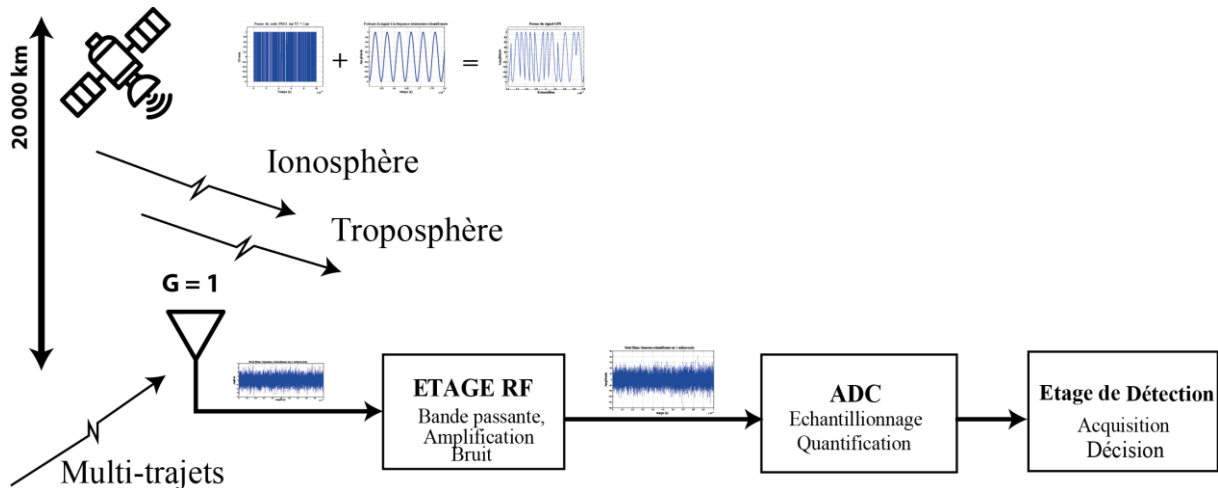


Figure I.1 : Illustration des principaux espaces de modélisations du signal.

#### ➤ Gain de l'antenne d'émission

Dans un bilan de liaison, le gain est la caractéristique la plus importante d'une antenne. Il est défini comme étant le rapport de la puissance rayonnée par l'antenne, dans une direction, par unité d'angle solide, sur la puissance rayonnée par une antenne référence par unité d'angle solide. Ce gain est dit absolu si l'antenne référence est isotrope, c'est-à-dire qu'elle rayonne uniformément dans toutes les directions. Il est exprimé en dBi. Le gain absolu est souvent utilisé dans le calcul d'une liaison par satellite. Pour un satellite de la constellation GPS, on considère que la puissance générée par le satellite est de 33 watts, et pour une antenne isotropique [2], la densité de puissance sera :

$$P_t = P \times G \text{ [watts]} = P + G \text{ [dBw]} \quad (\text{I.9})$$

$$P = \frac{P_t}{4\pi R^2} \text{ [watts / m}^2\text{]} \quad (\text{I.10})$$

$$P[\text{dB}] = 10 \cdot \log\left(\frac{P_t}{4\pi R^2}\right) = 10 \log(P_t) - 10 \log(4\pi R^2)$$

$P_t$  : Puissance total émise par le satellite



P : Puissance reçu au niveau du récepteur

Puisque le niveau de puissance est inversement proportionnel au carré de la distance en dB, la différence de niveau de puissance peut être trouvée comme :

$$\Delta P = 10 \cdot \log \left( \frac{25785^2}{20192^2} \right) = 2.1 [dB] \quad (I.11)$$

L'antenne, , ayant une directivité et une ouverture du lobe principal, à un gain qui représente la fraction ou le rapport géométrique de son ouverture sur celle de la sphère de la puissance isotopique dissipée [2].

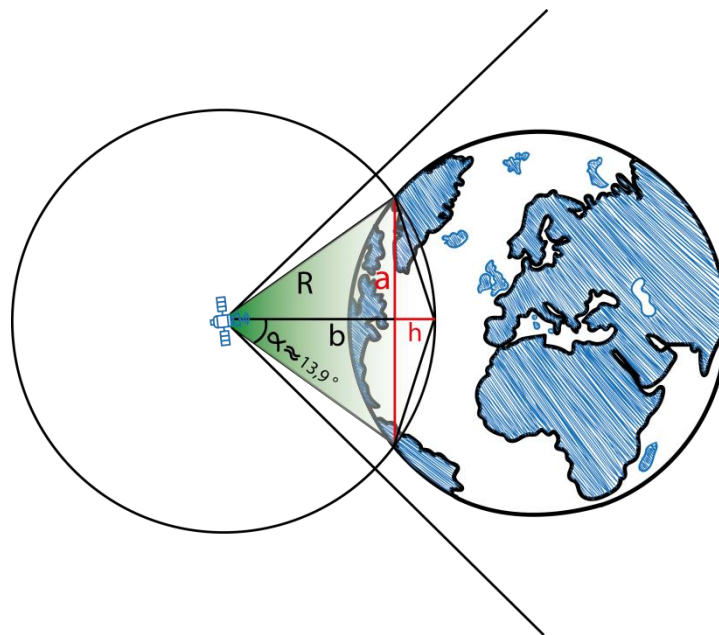


Figure I.8 : Gain de l'antenne d'émission.

Surface d'une sphère :  $4\pi R^2$

Surface d'un arc sphérique :  $a = R \sin(\alpha)$ ,  $b = R \cos(\alpha)$

On a  $h = R - b = R(1 - \cos \alpha)$  ;  $p = \sqrt{h^2 + a^2} = R\sqrt{2 - 2\cos \alpha}$

Le rapport s'écrit :

$$G(\alpha) = \frac{4\pi R^2}{\pi p^2} = \frac{2}{1 - \cos \alpha}$$

La valeur approximative de  $\alpha$  est de  $13.87^\circ$  qui est l'angle solide pour couvrir la terre [2].

**Note :** le gain d'antenne s'ajuste afin recevoir une puissance minimum garantie sur toute la surface terrestre rayonnée

### ➤ Atténuation du signal

La puissance reçue, garantie aux récepteurs, peut être calculée à partir de la puissance émise par le satellite, largeur de bande de l'antenne émettrice, la distance entre le satellite et le récepteur ainsi que la surface effective de l'antenne de réception. On aura la distance du

satellite de la surface  $R = 20000 \text{ Km}$  la surface effective  $A$  d'une antenne  $A_{eff} = \frac{\lambda^2}{4\pi}$  avec

$\lambda = 0.19 \text{ m}$  : longueur d'onde. On notera également le gain de l'antenne de réception  $G_a = 1 = 0 \text{ dB}$  La puissance reçue s'écrit :

$$P_{reçue} = \frac{P_t \cdot G_t \cdot A}{4 \cdot \pi \cdot R^2} = 1.41 \cdot 10^{-16} [\text{w}] = -158.5 [\text{dBw}] = -128.5 [\text{dBm}] \quad (\text{I.12})$$

La puissance reçue comme estimée est très faible et le spectre est étalé, donc il ne peut être directement analysé par le récepteur [1] [2][8].

### ➤ Pertes de propagation

L'atmosphère autour de la Terre affectera la vitesse de déplacement du signal GPS et provoque des erreurs de mesure. Ces erreurs devraient être corrigées. Pour GPS application, l'atmosphère est généralement divisée en deux parties : *l'ionosphère* et la *troposphère*. La *troposphère* est la plus proche des deux à la surface de la terre tandis que l'ionosphère est au-dessus de la troposphère. La troposphère contient les particules neutres et l'ionosphère contiennent des ions libres.

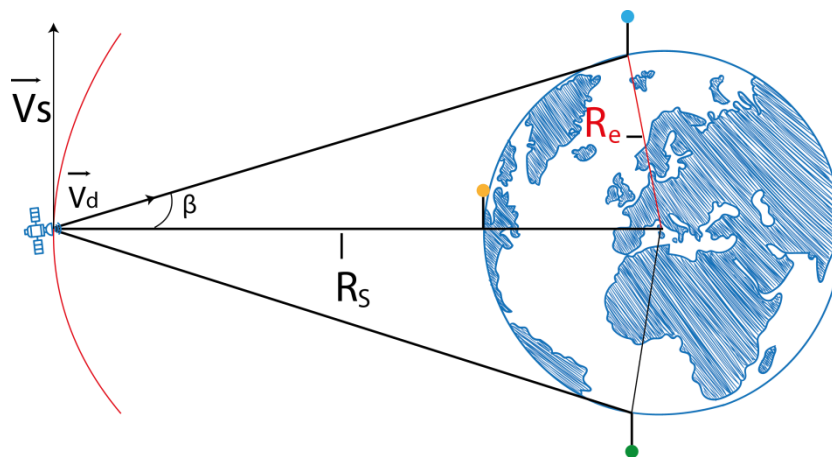
*L'ionosphère*, couche haute de l'atmosphère, contient des particules chargées qui réduisent la vitesse de propagation et allongent le chemin parcouru par réflexion (diminution de la vitesse de groupe et avance de la phase) causant des erreurs pouvant atteindre plusieurs dizaines de mètres. Le modèle de correction ionosphérique utilisé en aéronautique [15] (*Annexe 10 volume 13 OACI 3.1.2.4 Correction du retard ionosphérique*)

*La troposphère*, couche basse de l'atmosphère, contient de la vapeur d'eau. Elle ralentit le signal et peut causer des erreurs de l'ordre de 2 à 25m, et des atténuations inférieures à 0.5dB. Ces effets varient en fonction de l'angle d'élévation, de la pression et de la température du milieu. Des modèles mathématiques, prenant en compte ces paramètres, peuvent apporter des corrections sur les retards induits par la troposphère. Il est aussi important de noter que la troposphère est un milieu non dispersif et donc, affecte les signaux transmis à différentes fréquences de la même manière.

**Remarque :** Il est à noter que l'impact de l'ionosphère sur la transmission des signaux dépend directement de la fréquence (évolution du retard en  $1/f^2$ ). L'utilisation de 2 fréquences différentes (par exemple L1 et L2 en GPS) permet alors d'estimer et de compenser les retards induits par l'ionosphère. Cependant, les récepteurs classiquement rencontrés sont mono fréquence.

➤ **Effet Doppler**

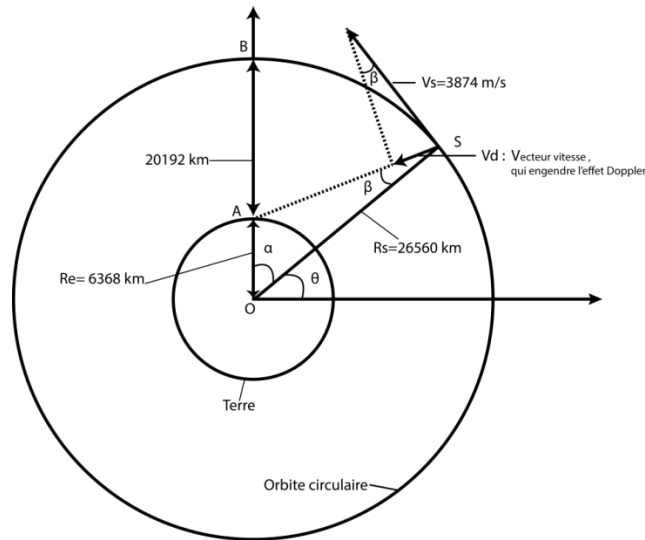
La vitesse du satellite et celle du récepteur influent sur le signal reçu. Ces deux vitesses génèrent une déviation de la fréquence du signal reçu, de celle émise [8]. Cette déviation est en fonction de plusieurs paramètres, c'est l'effet Doppler. On illustrera le phénomène comme suit, nous nous intéresserons d'abord, à la vitesse du satellite :



**Figure I.9 :** vitesse du satellite et effet doppler.

$V_s$  : est la vitesse radiale du satellite ;  $R_e$  : Rayon de la terre  $R_s$  : Rayon de l'orbite ;  $\beta$  : Angle de position du satellite par rapport au zénith.

Les trois illustrations des positions de récepteur sont indiquées par les trois points, bleu jaune et vert, ces trois points sont considérés comme les points aux limites, vues du récepteur. Le satellite est au zénith pour le point jaune, et à l'horizon aux points vert et bleu.



**Figure I.10 :** *Frequence Doppler causée par le mouvement du satellite.*

Dans la figure **I.10** le satellite est à la position S et l'utilisateur est à la position A. La vitesse du satellite peut être calculée à partir de sa vitesse angulaire ( $d\alpha/dt$ ) et la vitesse radiale [1].

$$\frac{d\alpha}{dt} = \frac{2\pi}{12 \times 3600} = 1.458 \times 10^{-4} \quad [rad / s] \quad (I.13)$$

$$V_s = R_s \frac{d\alpha}{dt} = 26560 \text{ km} \times 1.458 \times 10^{-4} = 3874 \quad [m / s]$$

La fréquence Doppler est due à la vectrice vitesse en direction du récepteur :

$$V_d = V_s \sin \beta \quad (I.14)$$

En utilisant la loi de cosinus et du sinus sur le triangle OAS, on aura :

$$AS^2 = R_s^2 + R_e^2 - 2 R_e R_s \cos \alpha = R_s^2 + R_e^2 - 2 R_e R_s \sin \theta \quad (I.15)$$

$$\frac{\sin \beta}{\sin \alpha} = \frac{\sin \beta}{\cos \theta} = \frac{R_e}{AS}$$

En remplaçant dans l'équation **(I.15)** :

$$V_d = \frac{V_s R_e \cos \theta}{AS} = \frac{V_s R_e \cos \theta}{\sqrt{R_s^2 + R_e^2 - 2 R_e R_s \sin \theta}} \quad (I.16)$$

Quand  $\theta = \pi/2$ , la vitesse doppler  $V_d$  est nulle. La vitesse est à son maximum à l'horizon et  $\theta$  est donné comme suit :

$$\sin \theta = \frac{R_e}{R_s} \quad \text{or} \quad \theta = \sin^{-1} \left( \frac{R_e}{R_s} \right) \approx 0.242 [rad] \quad (I.17)$$

Le module du vecteur vitesse  $V_d$  à son maximum se calcule comme suit :

$$V_{dm} = \frac{V_s \text{Re}}{R_s} = \frac{3874 \times 6368}{26560} \approx 929 [m/s] \quad (\text{I.18})$$

On obtient la fréquence Doppler à partir de la fréquence RF [1][8]:

$$f_D = \frac{f_{RF} \times V_{dm}}{c} = \frac{1575.42 \times 929}{3 \times 10^8} \approx 4.9 [KHz] \quad (\text{I.19})$$

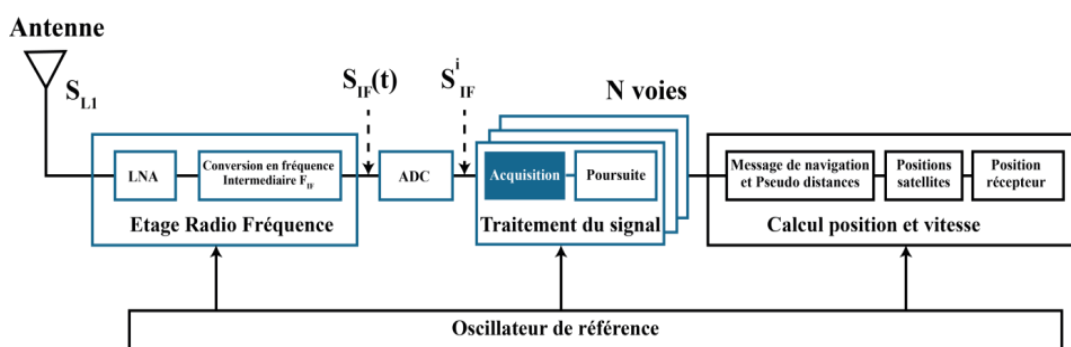
C'est pourquoi pour un récepteur statique la déviation est entre  $\pm 5$ khz. Si un récepteur se déplace à une grande vitesse l'effet doppler dû au déplacement du récepteur doit être pris en considération, ces valeurs déterminent la plage de recherche de la fréquence dans l'acquisition.[1]

Pour la fréquence du code qui est relativement faible, à partir de sa fréquence, mais peut causer un désalignement du code généré avec le code reçu qui n'est pas désiré .Si on échantillonne le signal à 5MHz or la fréquence du code pourrait se décaler de 15 H donc à un temps 1/15hz il y aura un décalage d'un chip entre les codes.

### I.4.3/ Evolution du signal dans le récepteur

Les récepteurs radio fréquence sont caractérisés par leur sensibilité et leur sélectivité [1] :

- *La sensibilité* définit la capacité du système à détecter et à amplifier le signal recherché quand son amplitude est faible.
- *La sélectivité* évalue la capacité du système à détecter et à amplifier le signal recherché lorsque celui-ci est entouré de signaux parasites d'amplitude plus forte.

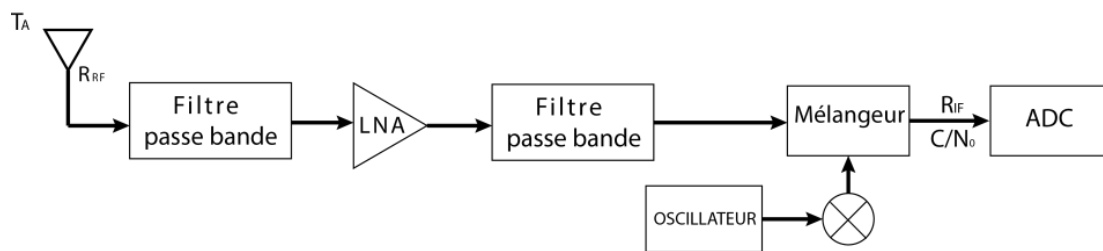


**Figure I.11** : composantes d'un récepteur GNSS et leurs rôles dans la chaîne.

Un récepteur GNSS peut être décomposé en trois blocs significatifs[8][9][3], comme illustré dans la **Figure I.12**.

- Le premier étant l'antenne et l'étage radio fréquence « *FRONT END* » (filtres, amplificateurs, Mélangeur, ADC) qui est chargé de conditionner le signal pour les traitements *soft*, c'est la partie hardware.
- Pour le deuxième bloque du traitement de signal appelé « *PROCESSING BLOC* » le traitement consiste à l'estimation des paramètres du signal, à savoir le code du satellite émetteur, le temps de propagation et la fréquence de réception, c'est « l'acquisition » l'autre opération de ce bloc est la poursuite ou des boucles d'asservissement poursuivent les paramètres obtenus de l'acquisition, ce qui demande moins de temps de traitement, néanmoins l'acquisition est nécessaire.
- Le 3ème bloque traite les données de navigation pour but d'acquérir le message de navigation, contenant les informations permettant de calculer les équations de navigation et obtenir la position et la vitesse du récepteur.

#### I.4.3.1. Etage Radio Fréquence



**Figure I.12** : Architecture généralisée d'un étage radio fréquence d'un récepteur GNSS.

Il est nécessaire d'étudier le comportement du signal à travers l'antenne et l'étage Radio Fréquence nommé ensemble « *FRONT END* », afin d'établir un bilan de liaison entre le signal reçu et celui qu'on va traiter dans l'étage d'acquisition, en définissant les variables pondérantes sur ce bilan. On s'intéressera également au  $C/N_0$  et on cherchera à exprimer la variable de sortie de l'étage RF en fonction du  $C/N_0$  à l'entrée du récepteur. Ce  $C/N_0$  sera important pour évaluer la sensibilité des détecteurs dans le chapitre III. Le signal intercepté par l'antenne subit deux traitements l'un analogique et l'autre Numérique. Le signal étant numérisé, l'étage d'acquisition qu'on verra en détails par la suite interviendra afin de relever les informations de temps et de fréquence recherchées.

Il existe deux manières de convertir le signal analogique en signal discret [1][3][5] :

- ✓ *Une digitalisation directe*, directement à la fréquence L1 : Pour réaliser une digitalisation directe, un convertisseur analogique numérique à fréquence élevée est nécessaire, ce qui n'est pas simple à concevoir, aussi l'amplificateur à fréquence

élevée est plus onéreux, de plus à une fréquence haute, un filtre à bande étroite cause des erreurs.

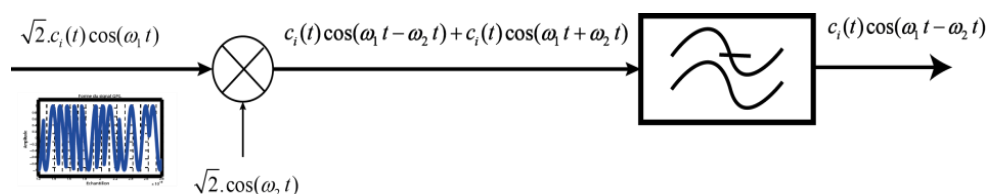
- ✓ Convertir le signal de la fréquence reçue à une fréquence intermédiaire  $f_{IF}$ , puis procéder à la digitalisation : Le passage à une fréquence intermédiaire détériore le signal en niveau (SNR) et en fréquence, tel que des atténuations (faibles) dues à la non linéarité des filtres engendrant des erreurs de quantification et peuvent erroner l'information reçue). A une fréquence basse  $f_{IF}$ , les filtres à bande étroite sont précis et engendrent une moindre perte tandis que l'oscillateur et le mixeur sont encombrants et causent des erreurs sur la fréquence, et donc auront un impact sur le signal reçu.

**Note :** Pour un traitement soft sur un signal GPS, il est requis d'avoir un signal numérique échantillonné, cet échantillon sera enregistré pour être traité, un minimum de 30 secondes d'enregistrement est nécessaire et ainsi l'espace de la mémoire influx sur la latence du processus.

### ➤ Fonctionnement général

Il existe plusieurs types d'architectures de récepteurs, dont les plus connus sont les architectures hétérodynes, homodynes ou les architectures à faible fréquence intermédiaire. Toutefois, on peut généraliser le fonctionnement de l'étage radio fréquence en notant les caractéristiques des composantes et leur effet sur le signal [20].

La partie *Front end* est constituée de filtres, amplificateurs, de mélangeurs et d'oscillateurs locaux. Le canal désiré est sélectionné par le biais d'un filtre passe bande, puis un amplificateur faible bruit amplifie le signal, très faible pour être quantifié directement, et cela tout en limitant le bruit à l'entrée du mélangeur. Le signal est ensuite transposé à une ou plusieurs fréquences plus basses, grâce à un mélangeur qui multiplie ce signal par une sinusoïde de fréquence  $F_{ol}$  fournie par l'oscillateur local. Le résultat de ce mélange est deux composantes aux fréquences  $F_{porteuse} \pm f_{ol}$ . Le filtre passe bas en sortie du mélangeur est alors utilisé pour ne retenir que la fréquence haute. [20]



**Figure I.13 :** Rôle du mélangeur et du filtre passe bas.

### ➤ Antenne de réception

Une antenne GPS doit avoir une grande ouverture pour recevoir un nombre maximal de signaux. L'exigence commune est de recevoir les signaux de tous les satellites à environ 5 degrés au-dessus de l'horizon. Toutefois, la Combinaison de satellites à faible angle d'élévation et des angles d'élévation élevés peut engendrer une faible valeur de dilution géométrique de précision (*GDOP*) [2][1][21]. De plus, d'un faible angle d'élévation viennent beaucoup d'interférences et multi trajet. Par conséquent, en choisissant une antenne GPS, un compromis entre le nombre maximal de satellites captés par l'antenne et le brouillage causé par des signaux non désirés doit être soigneusement évalué [2] [5].

On peut citer les principales caractéristiques d'une antenne GNSS [2] :

- ✓ **Le gain** : Dans les systèmes CDMA il est préférable de recevoir des signaux d'amplitudes similaires, car un signal à amplitude supérieur par rapport aux autres pourrait les rendre indétectables, pour cette raison on considérera notre antenne réceptrice isotopique.[2]
- ✓ **La bande passante** : est choisie selon les fréquences dédiées, une bande étroite diminue l'impact d'interférences.

**La polarisation et le diagramme de rayonnement** : sont aussi un atout pour atténuer les Multi trajets d'angles plats par rapport à l'angle d'ouverture de l'antenne GPS. Par conséquent, même les signaux des satellites avec une faible élévation seront atténués ce qui explique en partie l'angle de garde que spécifie l'ouverture de lobe de l'antenne d'émission.

**Remarque** : Si l'antenne est jumelée avec un amplificateur (antenne active) on peut relier le récepteur à l'antenne avec un long câble sans qu'il y est de pertes, la raison est apparente dans (.4.3.4).

### ➤ Mélangeur Oscillateurs locaux

Le mélangeur est utilisé pour la conversion de la fréquence du signal reçu. Étant donné la non-linéarité de ce module, il génère un très grand nombre de signaux parasites dont certains peuvent se retrouver à l'intérieur de la bande utile, qui ne vont pas être filtrés. La perte de conversion du mélangeur, généralement **entre 3 et 9 dB** [2].

- L'oscillateur local fournit au mélangeur le signal nécessaire à la conversion en fréquence du signal reçu. La fréquence du signal de l'oscillateur local doit être très stable, précise et comporter des signaux parasites extrêmement faibles. Il n'empêche



qu'en réalité il existe toujours une erreur (un résidu de fréquence), cela est un inconvénient pour la corrélation qu'on verra plus loin.[19]

### ➤ **Filtre passe bande**

Un filtre passe bande est un filtre ne laissant passer qu'une bande ou intervalle de fréquences, compris entre une fréquence de coupure basse et une fréquence de coupure haute du filtre. Caractérisé par sa fréquence de coupure, conventionnellement, cette limite est souvent fixée à la fréquence pour laquelle la puissance de sortie est réduite à moitié pour un signal d'entrée de même amplitude par rapport à la puissance de sortie à la fréquence de référence [1].

Vu le spectre du signal *GPS*, nous prenons en considération lors de la simulation que le filtre a une bande de fréquence de 2MHz autour de la fréquence porteuse, ou on pourra récupérer la majeure partie de la puissance du signal. En pratique, ce type de filtre laisse passer des résidus de fréquence au-delà de la bande voulue, également le filtrage se fait graduellement, ce qui provoque une atténuation du signal et un ajout de bruit. On notera cette atténuation  $\Delta_{IF}$  qu'on soustrait directement au  $(SNR)_0$  idéale (sans pertes).

Si  $X(t)$  est un processus aléatoire gaussien, appliqué à un système LTI de réponse impulsionnelle  $h(t)$ , alors le processus de sortie est également gaussien. Par conséquent, le processus de sortie  $Y(t)$  sera complètement spécifié, étant donné le processus d'entrée  $X(t)$  et la réponse impulsionnelle  $h(t)$ . [9]

### ➤ **Amplificateurs**

La plus importante des caractéristiques de ce module est son gain. Celui-ci doit être suffisamment élevé pour minimiser la contribution des modules subséquents au bruit, particulièrement celle du mélangeur. Il faudra toutefois s'assurer de ne pas saturer le mélangeur, avec un gain trop élevé [1][2].

L'amplificateur utilisé dans le récepteur GNSS est un amplificateur «*LNA : Low Noise Amplifier* ». La caractéristique de cet amplificateur est son très faible bruit. Ce module est conçu de telle sorte que sa contribution au bruit soit minimale permettant ainsi la détection de signaux très faibles. C'est cet amplificateur qui détermine la sensibilité du récepteur. L'amplification est nécessaire car la sensibilité du convertisseur analogique numérique requiert un niveau du signal beaucoup plus important que le niveau reçu, toute fois le bruit omniprésent, une trop forte amplification risque de saturer l'*ADC* avec le bruit.[1][2][3]

**Tableau I.2 : Bilan de liaison émission réception. :**

Paramètres	Valeur en SI	En dB	
Gain d'antenne d'émission	10		
Puissance émise	294 W		
Pertes d'espace	1,95 e-16 m <sup>-2</sup>		
Pertes atmosphériques	0,9	-0,5	
Puissance reçue	4,92 e-14 W/m <sup>2</sup>		
Gain d'antenne de réception	1	0	
Surface effective	2,87 e-3 m <sup>2</sup>		
Pertes de polarisation	0,50	-3	
Puissance reçue effective			1,41*10 <sup>-16</sup> W = -158,5dBw

L'amplification concerne le niveau du signal, tel que le niveau minimum reçu garanti, est de -130 dbm pour le signal GPS ce qui revient à choisir un amplificateur sensible à cette valeur.[2] [1] [3] [5]

#### I.4.3.2. Le Convertisseur Analogique-Numérique (ADC)

La conversion de données de l'analogique vers le numérique permet de passer d'un signal continu dans le temps et de valeurs continues à un signal à temps discret et à valeurs discrètes. *Un échantillonneur bloqueur* par exemple pourra prélever à période fixe la valeur du signal à convertir et maintenir la valeur constante (bloquer) le temps nécessaire à la quantification. Quantifier une valeur de  $s(n)$  c'est l'approximer grâce à un multiple entier d'une quantité élémentaire  $\Delta = 2V_{\max}/N_{\text{bit}}$  dite « *Pas de quantification* ». La discrétisation, à l'aide d'un quantificateur, des valeurs continues provoque une perte d'informations dû à des erreurs de quantification qui diffèrent d'un dispositif à un autre. Nous pouvons dès lors assimiler l'erreur de quantification  $e(t)$  est une variable aléatoire de répartition uniforme et on assume qu'il s'agit d'un bruit blanc gaussien sa puissance s'écrit [20] :

$$P_e = \int_{-\Delta/2}^{\Delta/2} e(t)^2 p_e(t) dt$$

$$P_e = \int_{-\Delta/2}^{\Delta/2} e^2 \frac{1}{\Delta} dt = \frac{\Delta}{12} \quad (\text{I.22})$$

Pour minimiser cet effet on peut [1] [20] :

- ✓ **Soit augmenter la précision du quantificateur** ce qui réduirait le pas de quantification : l'ajout d'un bit de précision divise la valeur de  $\Delta$  par un facteur 2.
- ✓ **Soit augmenter la fréquence d'échantillonnage :**

Les variables aléatoires de répartition uniforme engendrent un bruit de type blanc, c'est-à-dire de Densité Spectrale de Puissance (DSP) constante vis-à-vis des fréquences. Le bruit de quantification  $e(t)$  est donc un bruit blanc, nous noterons sa  $DSP|E(f)|^2$ . Pour calculer cette valeur, nous considérons que la condition de Shannon est remplie.

$$Pe = \int_{-F_{ech}/2}^{F_{ech}/2} |E(f)|^2 df = |E(f)|^2 F_{ech} \quad (I.23)$$

$$|E(f)|^2 = \frac{\Delta^2}{12F_{ech}}$$

La puissance de bruit de quantification associée au signal dans la bande utile  $[-fb,fb]$  est :

$$Pe = \int_{-f_b}^{f_b} |E(f)|^2 df = 2f_b \frac{\Delta^2}{12F_{ech}} \quad (I.24)$$

Il est donc plus avantageux d'augmenter la résolution du quantificateur. Malheureusement la diminution du pas de quantification est limitée par la technologie et l'architecture du quantificateur. On peut donc utiliser des systèmes utilisant une fréquence d'échantillonnage largement supérieure à  $2.f_b$  (fréquence de Nyquist) : c'est la technique dite de « sur-échantillonnage ». Cependant, une augmentation de la fréquence d'échantillonnage pose des problèmes de consommation ou de limitation de la fréquence de commutation des interrupteurs MOS. Ces pertes ont un impact négatif sur le  $(SNR)_0$  que nous noterons  $\Delta_Q$ . En négligeant les effets de la quantification, le signal sera de la forme [8][17] :

$$r(nT_e) = \sum_{i=1}^L A_i c_i(nT_e - \tau_i) d_i(nT_e - \tau_i) \cos \left[ \left( 2\pi(f_{IF} + f_{d,i})nT_e + \varphi_i \right) \right] + \eta_{IF}(nT_e) \quad (I.25)$$

### ➤ Détails sur la fréquence d'échantillonnage

Si un seul ADC est utilisé par un canal RF, le signal est réel, parallèlement avec un signal complexe, et la bande du signal est limitée par la fréquence d'échantillonnage selon  $Nyquist f_s/2 \geq B$  ou,  $f_s$  : fréquence d'échantillonnage ;  $B$  : bande passante du signal. Les emplacements des filtres dans un récepteur affectent les performances du frontal RF. Dans

plusieurs récepteurs *GPS*, le signal est converti dans deux canaux I et Q. A partir de là les données collectées sont complexes et la fréquence *Nyquist* sera donc  $f_s = B$ .

Choisir la fréquence d'échantillonnage se fait en prenant en considération la fréquence du code *PRN*, de manière que l'échantillonnage ne doit pas être synchronisé avec la fréquence du code *C/A* (1.023Mhz), car en considérant le biais d'horloge la résolution de la distance liée à cette erreur sera moins importante que si on choisit une fréquence d'échantillonnage asynchrone ou une fréquence différente que le multiple de la fréquence du code *PRN*. Toutefois pour notre cas, lors de la simulation d'une acquisition le but recherché n'est pas de calculer la position du récepteur et donc on peut négliger le biais d'horloge. Le signal reçu et échantillonné s'écrit [14] :

$$r[n] = \sum_{i=1}^L A_i e_i \left[ n - \frac{\tau_i}{T_e} \right] d_i \left[ n - \frac{\tau_i}{T_e} \right] \cos \left[ 2\pi \frac{(f_{IF} + f_{d,i})}{f_e} n + \varphi_i \right] + \eta_{IF}[n] \quad (I.26)$$

On simplifie pour un satellite :

$$r[n] = A e_i [n - \tau_0] d_i [n - \tau_0] \cos [2\pi F_D n + \varphi] + \eta_{IF}[n] \quad (I.27)$$

Dans cette équation,  $F_D = (f_{IF} + f_{d,i}) / f_e$  et  $\tau_0 = \tau_i / T_e$ .  $\eta_{IF}[n]$  est un bruit Gaussien centré discrétisé à la fréquence ' $f_e$ ' de densité spectrale de puissance ' $N_0$ ' et variance  $\sigma_{\eta}^2 = N_0 B_{IF}$ .  $B_{IF}$  est la bande passante après l'étage radio fréquence.

#### I.4.3.3. Bruit dans le Front End

Le bruit thermique par unité de largeur de bande d'un objet est équivalent au bruit thermique généré par une résistance de 50 ohms à une température absolue T, alors le bruit est exprimé en termes de température absolue T et devient équivalent à l'énergie thermique moyenne rayonnée à la température absolue T. La puissance du bruit thermique par unité de largeur de bande,  $N_0$  peut être exprimée par  $kT$  avec  $k$  désigne la constante de Boltzmann ( $k = 1,38 \times 10^{-23}$  J/K). La température T qui correspond à ce bruit thermique est appelée température de bruit [20] (*Noise temperature*).

La figure de bruit est une quantité qui exprime la qualité de point de vue bruit d'un dispositif tel qu'un circuit, un système ou une liaison. La formule de Friis nous permettra de calculer la température effective à la sortie de l'étage RF. Le niveau de bruit, dans un système de transmission par satellite doit être extrêmement faible puisque le signal désiré est souvent assez faible. Ce niveau du bruit peut être exprimé en termes de température absolue fictive.

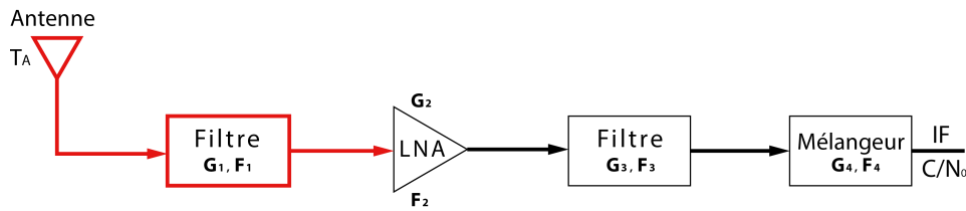


Figure I.14.a : Figure de bruit .

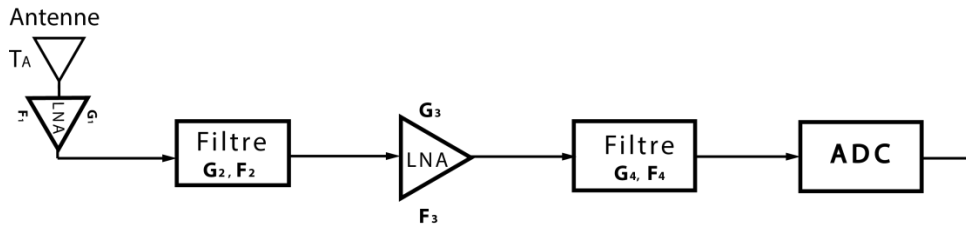


Figure I.14.b : Figure de bruit du « FRONT END » avec une digitalisation direct.

$$T_{eff} = T_a + (F_1 - 1) \times T + \frac{(F_2 - 1) \times T}{G_1} + \frac{(F_3 - 1) \times T}{G_1 \times G_2} + \frac{(F_4 - 1) \times T}{G_1 \times G_2 \times G_3} \tag{I.28}$$

T : température ambiante ; F : figure de bruit, circuit passif : F=1/G circuit actif. La figure de bruit est spécifiée par le constructeur.

Une antenne de réception capte en plus du signal utile des ondes radio nuisibles. De surcroît, les pertes thermiques de l’antenne sont considérées comme bruit thermique, noté Ta[19]. D’après l’équation le premier composant qui suit l’antenne est soit un filtre soit un amplificateur, ainsi la deuxième architecture citée requiert que l’amplificateur soit intégré avec l’antenne. On déduit que si l’amplification est placée en 1<sup>er</sup>, la figure de bruit est faible (approximativement égale à la figure de bruit du 1<sup>er</sup> ampli). La figure de bruit qui caractérise le composant qui vient après le 1<sup>er</sup> composant sera atténué par le gain du 1<sup>er</sup> amplificateur. Toutefois, un signal avec une grande bande passante affecte l’ampli et génère ainsi des fréquences résiduelles. Avec un filtre en premier l’atténuation du signal sera plus importante et la figure du bruit du récepteur plus grande (-2 3 dB) [2] il s’agira également d’un filtre volumineux Le 1<sup>er</sup> cas est généralement établi quand le système n’est pas soumis à beaucoup d’interférences. Cela nous permettra de calculer la densité et la puissance du bruit

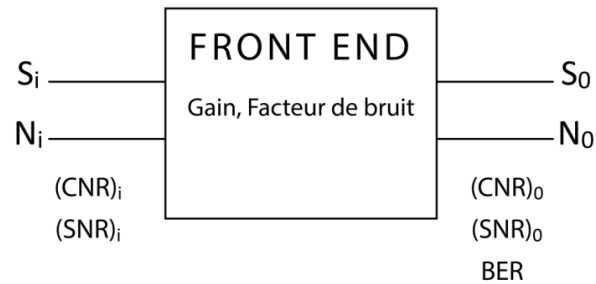
$$N_0 = K \times T_{eff} \quad \left[ \frac{w}{Hz} \right] \tag{I.29}$$

$$N_i = K \times T_{eff} \times B \quad [w]$$

Ni est la puissance de bruit en watts.

La température du bruit est à la température ambiante où  $T \approx 290\text{K}$  exprimé en dBm  
 $N_0 \text{ (dBm)} = -174 \text{ dBm /Hz}$  ou  $N_0 \text{ (dBm)} = -114 \text{ dBm /MHz}$ . Le signal est de 19 dB (-130 + 111) en dessous du bruit de fond. On prendra en compte le niveau de puissance par unité de la bande passante [dBw/Hz].

$$CNR = \frac{A^2}{N_0} = -160 \text{ [dBw]} - (-)204 \text{ [dBw/Hz]} = 45 \text{ [dB - Hz]} \quad (\text{I.30})$$



**Figure I.15:** Paramètres de l'étage RF.

*Note :* La formule du facteur de bruit renseigne que le 1<sup>er</sup> élément de la cascade RF dans le résultat du facteur de bruit ce qui indique que les composantes passives qui viennent avant les éléments actifs vont avoir un effet négatif sur le facteur. Une solution est d'incorporer un amplificateur avec l'antenne, antenne qui nécessite une alimentation pour l'antenne.

**Tableau I.3 :** Bilan de liaison entrée sortie de l'étage RF et SNR à l'entrée de l'acquisition.

Paramètres	Valeur en SI	En dB
Constante de Boltzmann	$1,38 \cdot 10^{-23} \text{ J/K}$	
Température d'antenne	130 K	
Température	290 K	
F1	1	
gain LNA	316,2	25
F LNA	1,5	1,9
F3	1,1	0,5
F4	7,9	9
$T_{\text{eff}}$	296,4 K	
Figure de bruit	1.6	2
$N_0$		-203,9 dBw/Hz
Bande de fréquence	2 Mhz	
Puissance du bruit		

C/N0	45,4 dB-Hz
SNR	-20dB pour B = 2 Mhz

*Remarque : La majorité du bruit provient de l'étage radio fréquence par conséquent le choix de ce dernier est très important dans notre récepteur GPS.*

## I.5. CONCLUSION

Le système GPS a été étudié afin d'en tirer les éléments clés d'une simulation d'une chaîne d'acquisition sur logiciel. Ces paramètres consistent en la forme du signal et sa modulation, sa puissance, son spectre (bande passante), son mode de transmission, ses variations en temps de transmission et les perturbations que peuvent influencer ces paramètres notamment la variation de la fréquence doppler. Le choix d'une fréquence d'échantillonnage bien déterminée et une estimation de la quantité de bruit par unité de fréquence à l'entrée du récepteur a mené à choisir un modèle mathématique pour la simulation.

La partie *RF* constitue une partie importante dans le récepteur hybride, plusieurs études sont disponibles en littérature. L'impact de l'étage RF sur le traitement du signal a été notifié en analysant les différents composants de cet étage. L'analyse de la chaîne de réception nous a permis d'exprimer le modèle mathématique en fonction du rapport signal sur bruit par unité de fréquence, ce paramétrage nous permet de négliger l'étage RF pour l'analyse des performances de l'acquisition en prenant directement plusieurs valeurs de cette métrique.

# *Chapitre II*

## *Fonctionnement, conception et implémentation du FPGA*

- ❖ Introduction
- ❖ Flot de conception FPGA
- ❖ Architecture d'un Circuit FPGA
- ❖ Le module ADM-XRC-II
- ❖ Le langage VHDL



## II.1.INTRODUCTION

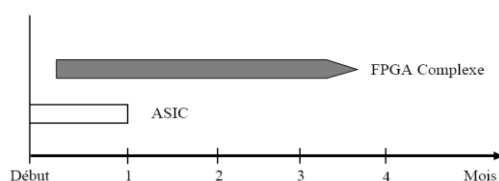
Durant ces deux dernières décennies, les méthodes de conception et d'implémentation des fonctions numériques ont connu un développement important. Dans les années soixante-dix, les applications de la logique câblée étaient des circuits intégrés standards, pris assez souvent de la famille *TTL*. C'était dans les années quatre-vingt que les premiers circuits programmables furent apparus, ils peuvent être simples pour les applications ordinaires, ou des circuits intégrés spécifiques (*ASIC*) pour les fonctions complexes. La complexité de ces circuits a nécessité la création des logiciels de hauts niveaux, pour leur description structurelle. Actuellement, le langage de programmation pour l'implémentation des circuits le plus utilisé est le *VHDL* [13].

Les circuits *FPGA* (*Field Programmable Gate Array*) demeurent toujours les circuits reprogrammables les plus développés, les plus fiables et les plus fertiles. Ils sont en effet, entièrement configurables par programmation, et ils permettent d'implanter physiquement, par simple programmation, n'importe quelle fonction logique. De plus, ils ne sont pas limités à un mode de traitement séquentiel de l'information comme avec les microprocesseurs ; et en cas d'erreur, ils sont reprogrammables électriquement sans avoir à extraire le composant de son environnement [6].

Ainsi, cette technologie, qui ciblait principalement les domaines : public, médical, télécommunication et automobile, devient intéressante pour des applications plus simples et portant sur des volumes moins importants. Et ceci particulièrement depuis que les *FPGA* sont proposés à un prix très faible (0.0001 dollar par porte).

### II.1.1/ Limites de la FPGA

L'emploi de *FPGA* est devenu, à juste titre, une solution privilégiée pour produire rapidement des circuits. Pourtant, la complexité croissante des plus grandes matrices engendre de graves problèmes à l'étape de routage. Entre les *ASIC* et les *FPGA* de haute complexité, la durée du procédé de placement-routage sous contrainte temporelle varie beaucoup avant de converger vers la solution optimale (Figure II.1).

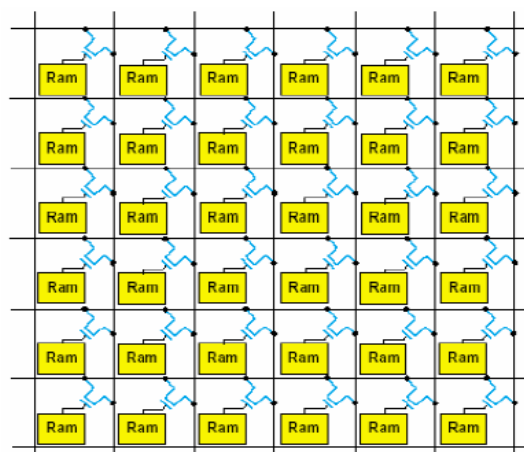


**Figure II.1 :** Comparaison des temps de routage entre circuits configurables

Les *FPGA* ont certes de nombreux atouts, mais les concepteurs de systèmes ont tendance à se tourner, par habitude, vers cette solution sans regarder si elle répond correctement aux objectifs de performances et de coût, voire si elle s'inscrit bien dans le calendrier de développement de leurs projets. Pourtant, les *FPGA* de forte complexité induisent des limitations sévères qui peuvent pénaliser le bon déroulement d'une conception [6].

Certaines applications, dans le domaine des réseaux et communications notamment, exigent des fonctions complexes et un nombre élevé de broches, ceci assorti de performances toujours croissantes. De ce fait, les plus grands réseaux logiques programmables choisis pour réaliser ces projets sont fréquemment employés au-delà de leurs possibilités physiques, des limitations inhérentes à leurs architectures. Avec de tels *FPGA*, il faut souvent plusieurs mois avant de converger vers un routage satisfaisant au niveau des performances temporelles. Dans ce cas, la convergence vers un timing optimal est presque toujours un procédé long, imprévisible, onéreux, et extrêmement laborieux.

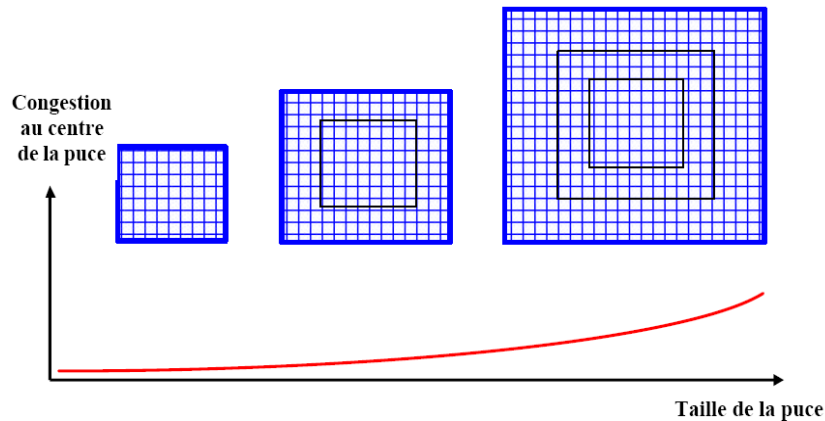
Au niveau des *FPGA* de forte complexité, la difficulté d'utilisation de ces matrices dans des applications de hautes performances réside dans le principe d'interconnexions programmables pour le routage des signaux. Et ceci n'est pas entièrement dû au fait que la quantité de ressources de routage disponibles est plus faible dans les *FPGA* basés sur des points SRAM que dans les technologies concurrentes. Entre en jeu également le retard substantiel, de type R-C, inhérent à ces circuits programmables par SRAM (Figure II.2).



**Figure II.2 :** Réseau d'aiguillage typique d'un *FPGA*

Chaque point d'aiguillage consomme au moins cinq transistors pour la Ram et un transistor interrupteur de grande taille. Ce délai de propagation rend les performances du composant hautement sensibles au moindre changement dans un chemin de routage. Comme

l'illustre la *Figure II.3*, le point de congestion des équipotentielles, situé au centre de la puce, est d'une taille beaucoup plus importante pour les circuits les plus complexes d'une famille de *FPGA*. En outre, quand une interconnexion doit prendre un chemin détourné pour éviter cette congestion, l'opération se traduit par une augmentation conséquente des délais de propagation.



**Figure II.3 :** Zones de congestion dans le routage des *FPGA*

De ce fait, alors les matrices de petite ou moyenne taille permettront de converger rapidement vers une solution de routage temporellement satisfaisante, cette bonne conclusion demande souvent trois ou quatre mois pour les grands *FPGA* [10] .

### II.1.2/ Technologie FPGA du Xilinx

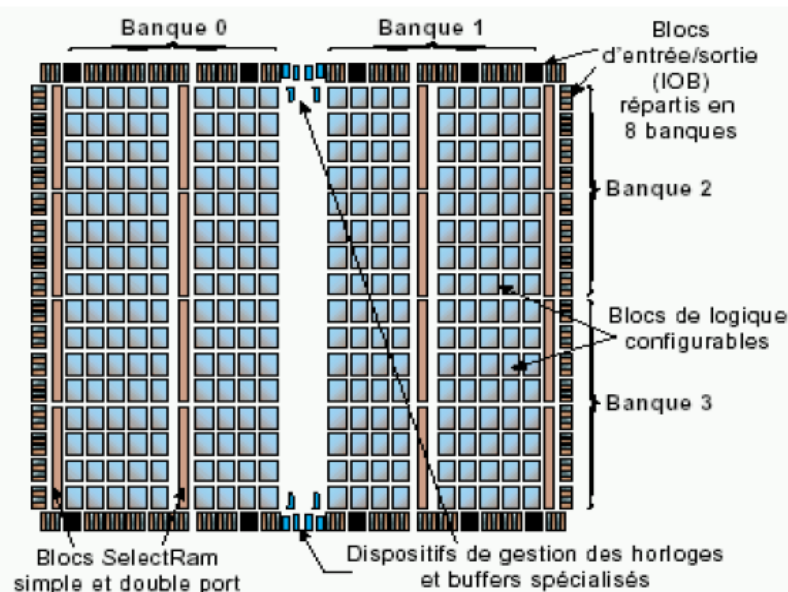
Dès l'origine, les *FPGA*, tels que *Xilinx* les a inventés, avaient la réputation de mettre à disposition de l'utilisateur une conception rapide, fiable et simple. Les progrès technologiques ont permis d'accéder à des matrices logiques programmables de plusieurs millions de portes. Cette complexité actuelle reste absolument gérable et permet la réalisation d'applications très performantes moyennant une bonne connaissance des ressources offertes et le respect d'une méthodologie de conception.

La Figure II.4, montre l'architecture simplifiée d'un *FPGA Xilinx* (pour les familles *Spartan-IIE*, *Virtex-E* et *Virtex-II*). Les principales caractéristiques de ces trois familles sont :

- Complexités allant de 1500 à plus de 8 millions de portes ;
- Faible consommation ;
- Grande souplesse d'utilisation des entrées/sorties avec adaptation d'impédance (*Virtex-II*) et configuration en mode différentiel (*Spartan-IIE*, *Virtex-E* et *Virtex-II*) ;
- Fonctions mémoire (distribuée et blocs de Ram) ;
- Dispositifs de gestion des horloges (*DLL* et *DCM*) ;

➤ Multiplieurs câblés (*Virtex-II*) ;

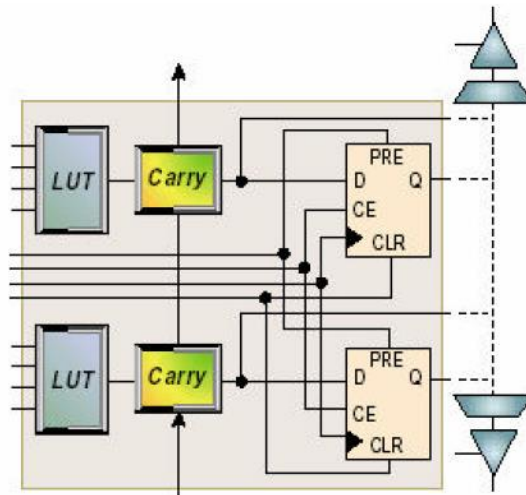
Et bien d'autres possibilités permettant d'optimiser à la fois la performance et la densité des fonctions logiques et arithmétiques.



**Figure II.4 :** Organisation des FPGA de Xilinx

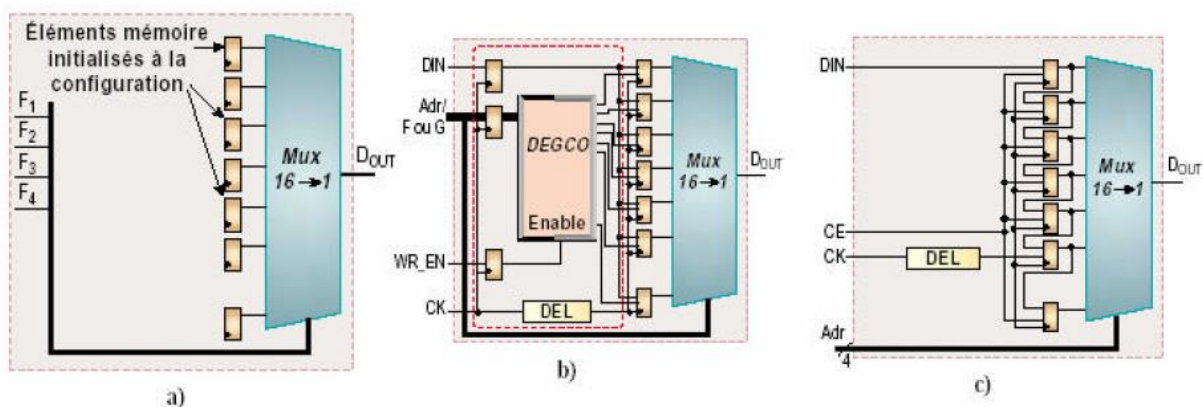
Nous y voyons qu'un des modules de base est le bloc logique configurable (ou *CLB*). Lui-même constitué de *slice*. La *Figure II.5*, illustre l'architecture simplifiée d'un slice.

La logique combinatoire est implantée grâce aux *LUT* (*Look-Up Table*) contenus dans chaque slice. Ces *LUT* peuvent également être configurés comme éléments de mémoire synchrone, simple ou double-port de 16 bits, ou encore comme registres à décalage de 16 bits. Il existe donc trois modes de configuration de ces *LUT*. Plus précisément, le fonctionnement en mode combinatoire est obtenu en lisant le contenu pointé par les signaux d'entrée (*Figure II.5a*). Autrement dit, les *LUT* sont des mémoires dont le contenu est initialisé lors de la configuration du FPGA. De ce fait, elles permettent à l'utilisateur d'en disposer en mode « élément mémoire » dans chacun des slices si nécessaire (*Figure II.5b*). La *Figure II.8c* décrit le mode de configuration particulière en registre à décalage de longueur programmable jusqu'à 16 bits. Par ailleurs, une logique supplémentaire utile pour la réalisation des fonctions arithmétiques est disponible dans chaque slice. Grâce à ces éléments, et au style d'écriture adapté, des modules de type accumulateur chargeable en addition/soustraction pourront être implantés à raison de 2 bits par slice [10].



**Figure II.5 :** Architecture simplifiée d'un « Slice »

Les bascules dans chaque slice ont aussi des caractéristiques importantes pour le concepteur. En particulier, elles sont initialisées systématiquement à la mise sous tension (par défaut à valeur '0'), et sont utilisables indépendamment de la logique combinatoire disponible dans le même slice. En outre, chaque bascule bénéficie de broches de contrôle telles que : entrée dédiée de validation de l'horloge (*Clock Enable*) permettant d'activer ou de suspendre le fonctionnement de chacune des bascules individuellement, et ceci sans avoir à insérer de la logique combinatoire sur le chemin de l'horloge (entrée de « set » et de « reset » synchrones ou asynchrones). La polarité des signaux d'horloge, de « *Clock Enable* », de set et reset est programmable pour chacune des bascules. Autrement dit, ces signaux peuvent être individuellement actifs au niveau haut ou au niveau bas [10].

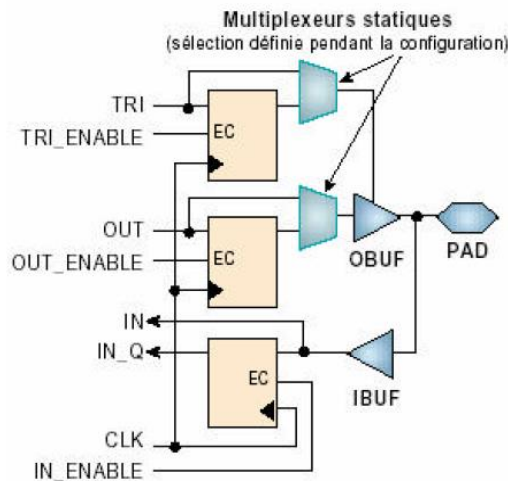


**Figure II.6 :** Les modes de configuration des LUT

L'échange des données avec les circuitries externes (microprocesseur, *DSP*, mémoires rapides, convertisseurs A/N et N/A) est souvent un critère important dans la

perspective des performances à atteindre. Les blocs d'entrées/sorties disposent de bascules sur les chemins d'entrées/sorties et de contrôle à trois-états. Les outils de synthèse les plus sophistiqués permettent l'utilisation systématique de ces bascules.

Enfin, les *FPGA* de la famille *Virtex-II* intègrent des registres doubles sur chacun de leurs trois chemins, autorisant ainsi la réalisation d'interfaces *DDR (Double Data Rate)* pour la communication avec les mémoires *SDRAM* du même nom ou autres dispositifs de même type (Figure II.6).



**Figure. II.7 :** Configuration des blocs d'E/S

La configuration électronique des blocs d'E/S donne la possibilité d'ajuster la raideur des fronts sur les étages de sortie, la sortance, les seuils de communication et les standards de communication (*LVTTL, LVCMos, SSTL, PCI, LVDS...*). Par ailleurs, la famille *Virtex-II* offre d'adapter en impédance aussi bien les entrées que les sorties, sans avoir besoin d'insérer les traditionnelles résistances d'adaptation [10] .

### II.1.3/ La famille FPGA Xilinx Virtex-II

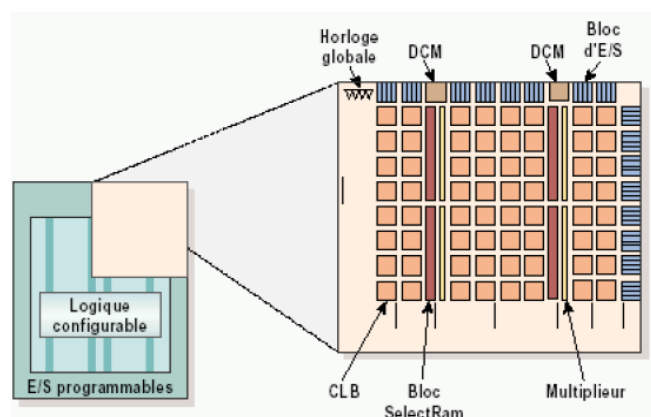
La famille *FPGA Xilinx Virtex-II* est constituée de onze composants dont les caractéristiques sont résumées par le *Tableau II.1*.

Device	System Gates	CLB (1CLB=4slices=Max128 bits)			Multipliers Blocks	Select RAM Blocks		DCMs	Max I/O pads
		Array Row X col	Slices	Maximum Distributed RAM Kbits		18Kbits Blocks	Max RAM (Kbits)		
XC2V40	40K	8 x 8	256	8	4	4	72	4	88
XC2V80	80K	16 x 8	512	16	8	8	144	4	120
XC2V250	250K	24 x 16	1.536	48	24	24	432	8	200

XC2V500	500K	32 x 24	3.072	96	32	32	576	8	264
XC2V1000	1M	40 x 32	5.120	160	40	40	720	8	432
XC2V1500	1.5M	48 x 40	7.860	240	48	48	864	8	528
XC2V2000	2M	56 x 48	10.752	336	56	56	1.008	8	624
XC2V3000	3M	64 x 56	14.336	448	96	96	1.728	12	720
XC2V4000	4M	80 x 72	23.040	720	120	120	2.160	12	912
XC2V6000	6M	96 x 88	33.792	1.056	144	144	2.592	12	1.104
XC2V8000	8M	112 x 104	46.592	1.456	168	168	3.024	12	1.108

**Tableau II.1 : Famille FPGA Virtex-II**

Les blocs « *SelectRam* » dans l'architecture *Virtex-II* sont des mémoires Ram double port de 18Kbits, configurables de différentes manières entre 18x1Kbit et 512x36 bits. Chaque port est totalement synchrone et indépendant. Les blocs « *Selectram* » peuvent être mis en cascade pour réaliser des larges zones des stockages enfouies dans le *FPGA*. Un module multiplieur 18x18 bits est disposé à proximité de chaque bloc « *SelectRam* » et optimisé pour opérer sur le contenu d'un port de la mémoire [11].



**Figure II.8 : Détail de l'architecture d'un composant Virtex-II**

Les fonctions mémoire « *SelectRam* » et les multiplieurs sont connectés aux matrices d'interconnexions, cette configuration leur ouvrant l'accès aux ressources de routage globales de la matrice *FPGA*. La *Figure II.8* présente en détail l'architecture d'un composant *Virtex-II*. Dans cette famille de *Virtex-II*, on s'intéresse particulièrement au composant *XC2V6000*, puisque c'est le composant supporté par la plateforme *ADM-XRC-II* [11].

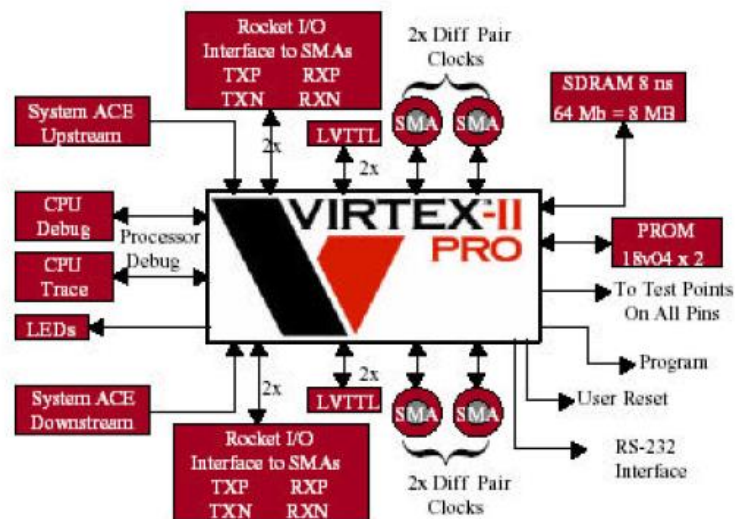
#### II.1.4/ FPGA Virtex-II Pro

*Xilinx* a mis depuis peu sur le marché, une nouvelle technologie de circuits *FPGA* intégrant de nouvelles fonctionnalités, ces dernières permettent la réalisation d'architectures



qui auparavant s'avéraient très compliquées à entreprendre. Parmi ces nouveaux circuits, le *VIRTEX-II PRO* (Figure II.9) qui est un composant *FPGA* intégrant un cœur de processeur PowerPC 405 de chez IBM pouvant fonctionner à plus de 300MHz. Afin de rendre le co-design plus simple à réaliser, la famille de composant *Virtex-II-Pro* possède plusieurs atouts tel que :

- L'intégration d'un ou deux cœurs de processeur *powerPC405* ;
- Un grand nombre de slices pouvant atteindre 44.096 unités permet d'implémenter des algorithmes dont la complexité les rend gourmand en consommation des ressources logiques ;
- Des blocs de mémoire de 18Kb sont aussi disponible ce qui diminuera l'accès vers des ressources externes et ce qui peut être fatal pour le circuit surtout du point de vue consommation d'énergie ;
- Des multiplieurs 18x18 bits, des *DCM* (*Digital Clock Manager*).



**Figure II.9 : Vue Globale du VIRTEX-II PRO**

Pour cela *Xilinx* fournit des IPs soft (Figure II.9) écrites en langage *VHDL* qui permettent d'exploiter le PowerPC405 et de lui fournir tous les périphériques nécessaires à son fonctionnement. Les premières IPs à implémenter autour du processeur sont le *PLB* « *Processor Local Bus* » et le « *On Chip Memory Controller* », la première constitue le bus de communication sur lequel viennent se greffer d'autres IPs, quant au « *On Chip Memory Controller* » il permet de connecter le PowerPC405 avec les *Bram's* du *FPGA* ou bien avec des *RAMs* externes où sera logé le code à exécuter. [11]



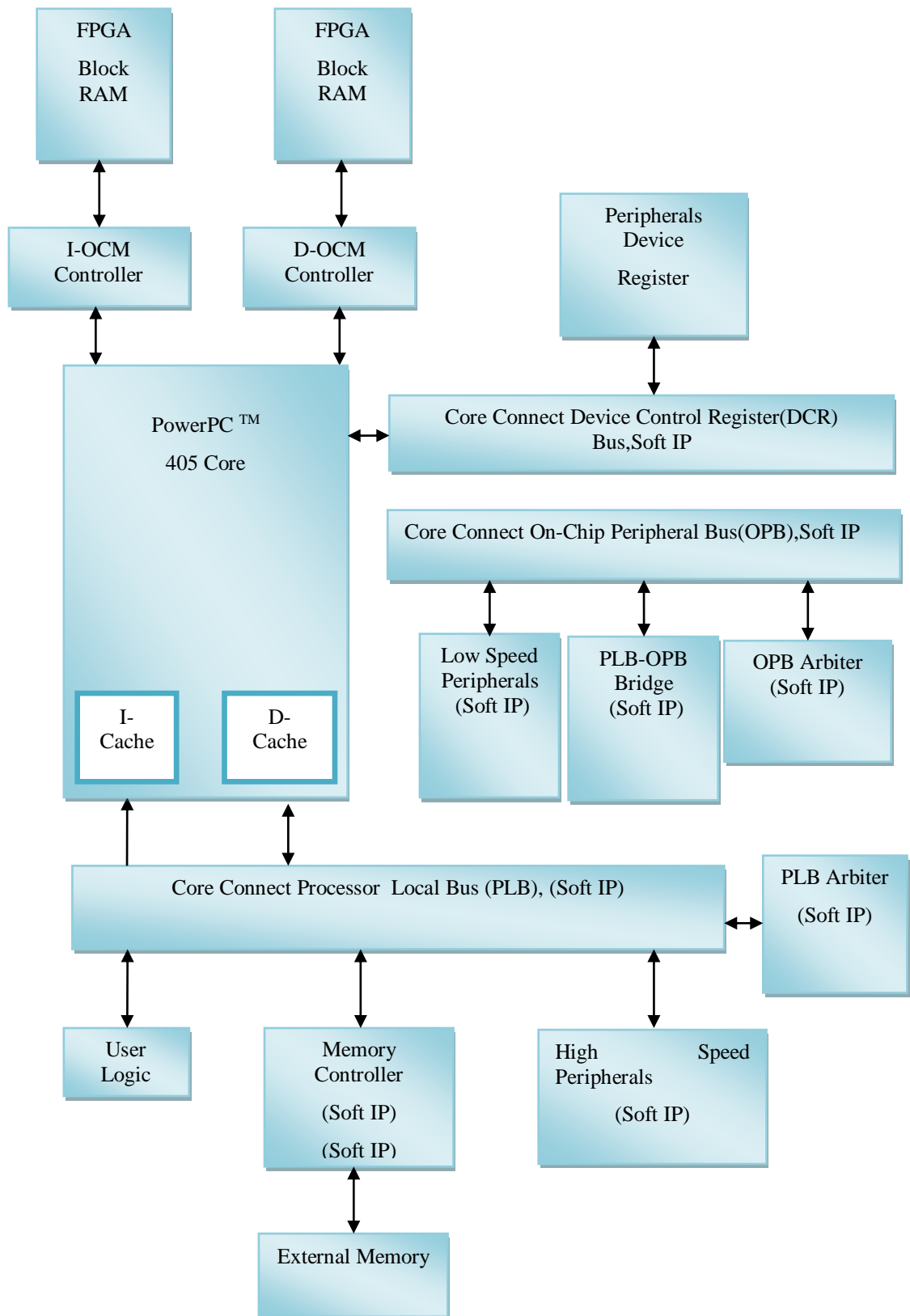


Figure II .10 : Architecture standard autour du PowerPc405 sur VIRTEX-II PRO

Une application peut se limiter à une configuration avec un PowerPC405, une IP « *Processor Local Bus* » et un « *On Chip Memory Controller* », l'utilisateur peut ajouter des bus ou bien des contrôleurs de mémoire externe ou autres IPs, selon les exigences de son architecture.

Device	RocketIO Transceiver Blocks	PowerPC Processo r Blocks	Logic Cells	CLB		18X18 Multi Blocks	Select RAM Blocks		DCMs	Max I/O pads
				Slices	Max Dist RAM		18KB Blocks	Max RAM (KB)		
XC2VP2	4	0	3.168	1.408	44	12	12	216	4	204
XC2VP4	4	1	6.768	3.008	94	28	28	504	4	348
XC2VP7	8	1	11.088	4.928	154	44	44	792	4	396
XC2VP20	8	2	20.880	9.280	290	88	88	1.584	8	564
XC2VPX20	8	1	22.032	9.792	306	88	88	1.584	8	552
XC2VP30	8	2	30.816	13.696	428	136	136	2.448	8	644
XC2VP40	0, 8, or 12	2	43.632	19.392	606	192	192	3.456	8	804
XC2VP50	0 or 16	2	53.136	23.616	738	232	232	4.176	8	852
XC2VP70	16 or 20	2	74.448	33.088	1.034	328	328	5.904	8	996
XC2VPX70	20	2	74.448	33.088	1.034	308	308	5.544	8	992
XC2VP100	0 or 20	2	99.216	44.096	1.378	444	444	7.992	12	1.164

**Tableau II.2 :** Ressources du circuit VIRTEX-II PRO/ VIRTEX-II PRO X

La famille *VIRTEX-II PRO* dispose d'un nombre suffisant de ressources permettant la réalisation d'applications « *System On Chip* » et ce avec une électronique associée très réduite. Le *Tableau II.2* donne un récapitulatif des ressources dont dispose le composant de la famille *VIRTEX-II PRO* [11].

## II.2. Flot de conception FPGA

*Xilinx* fournit dans son pack d'outils différents soft permettant la création de systèmes embarqués sur puce, parmi ces softs on dénombre *ISE (Integrated Software Environment)* et *EDK (Embedded Development Kit)*, tous deux nous offrent la possibilité d'avoir un bitstream pour la programmation des *FPGA* suivant l'application ciblée.

### II.2.1/ Vue Générale

Le flot de conception standard est composé de plusieurs étapes qui peuvent se résumer en :

- Conception et synthèse du design ;
- L'implémentation et la vérification du design.

Une vue générale du flot de conception est présentée par la figure suivante

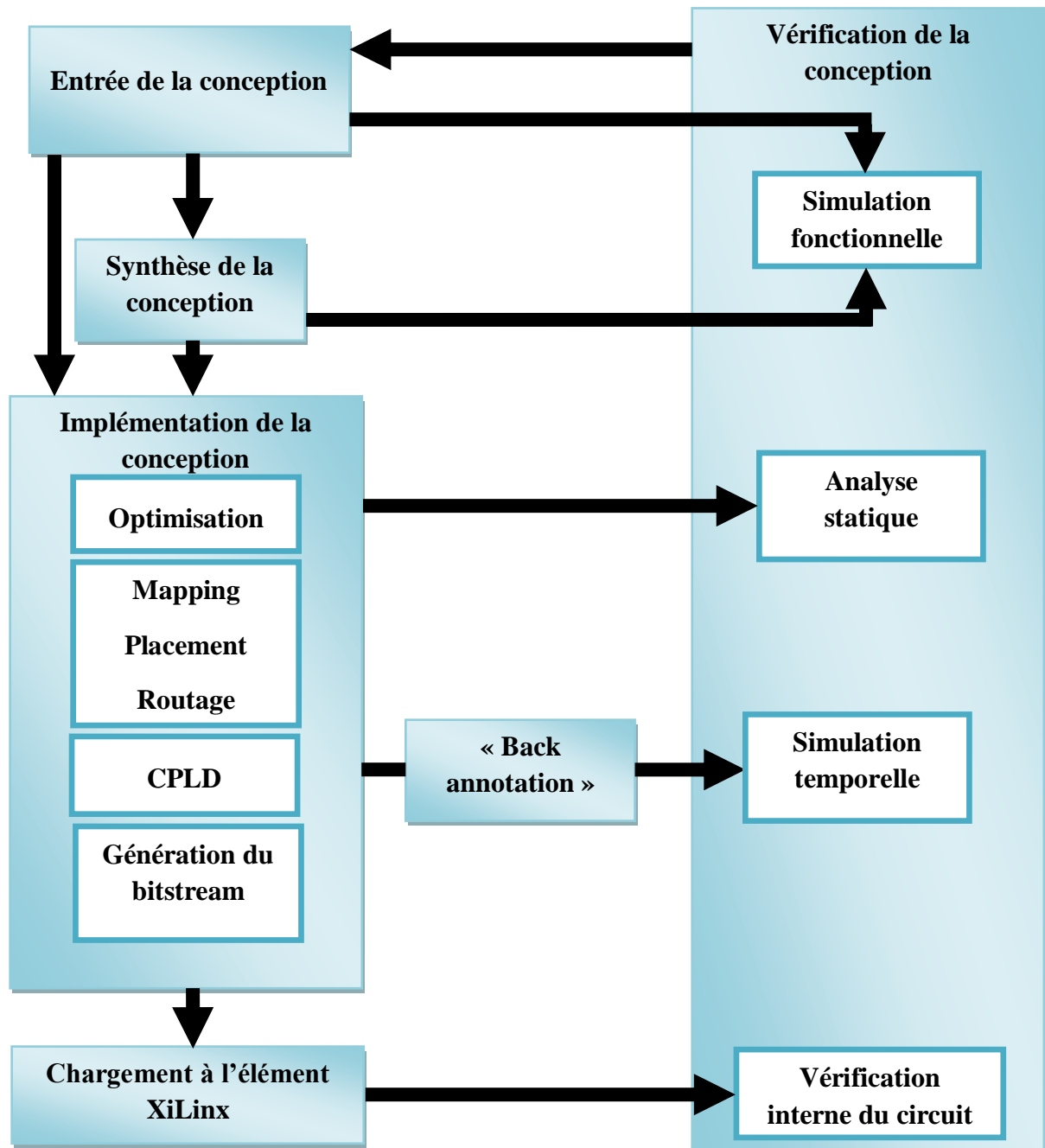


Fig.II.11 :Flot de conception générique [6]

## II.2.2/ Les différents étages du flot de la conception

### II.2.2.1. La synthèse :

Un design peut être conçu à l'aide d'un éditeur schématique ou d'un outil de traitement de textes. La conception commence par un concept exprimé par un schéma ou une description fonctionnelle. A partir du design original, une *netlist* est créée, synthétisée puis traduite en fichier *NGO* (*Native Generic Object*). Ce fichier est utilisé ensuite par un logiciel appelé *NGDBuild* qui produira alors un fichier *NGD* (*Native Generic Database*) [11] .

#### ➤ Design Hiérarchique

Une hiérarchie du design est importante dans les deux modes de conception : Schématique et HDL. Et ceci pour plusieurs raisons :

- Pour avoir un design bien structuré et facile à déboguer
- Combinaison de plusieurs types de conceptions (Schématiques, *VHDL*, *Verilog*, etc.)
- Design incrémental qui consiste à concevoir, implémenter et vérifier des parties de système individuellement.
- Réduit le temps d'optimisation et facilite les designs concurrents par une répartition du système en parties qui seront développées simultanément par plusieurs personnes (comme de design modulaire) [11] .

#### ➤ Schématiques

Les outils schématiques produisent des interfaces graphiques pour décrire le design. Ces outils peuvent être utilisés pour connecter des symboles représentant des instances des composants utilisés dans le design. Le design peut être construit à l'aide des portes individuelles, ou à l'aide de blocks fonctionnels en utilisant des éléments de bibliothèques et des outils comme *Core Generator* et *LogiBLOX* [11] .

#### ➤ Les éléments des Bibliothèques

Les primitives et les macros sont les constituants fondamentaux des bibliothèques de composants. Les bibliothèques de *Xilinx* offrent des primitives et des macros de haut niveau. Une primitive est une partie élémentaire du circuit (portes logiques, Flip-Flops, etc.). Chaque primitive a des propriétés caractéristiques (nom de bibliothèque, symbole et description). Un macro contient plusieurs éléments de bibliothèques qui peuvent être primitives ou des macros. Deux types de macros sont disponibles à l'utilisation avec les FPGA de *Xilinx* :

- Les *soft-macros* dont les fonctionnalités sont prédéfinies, mais qui sont flexibles du point de vue *mapping*, placement et routage.
- Les *RPMs* (*Relationally Placed Macros*) qui possèdent un *mapping* et un placement relatif fixés.

Les macros ne sont pas disponibles pour la synthèse parce que les outils de synthèses possèdent leurs propres générateurs de modules et ne requièrent pas de RPMs [11].

### ➤ Outil de Génération de Cores

L'outil de conception *CORE Generator* de Xilinx offre des cores paramétrables qui sont optimisés pour les *FPGAs* de Xilinx. La librairie contient des éléments qui varient entre de simples registres à des applications très complexes (filtres et multiplexeurs *DSP*).

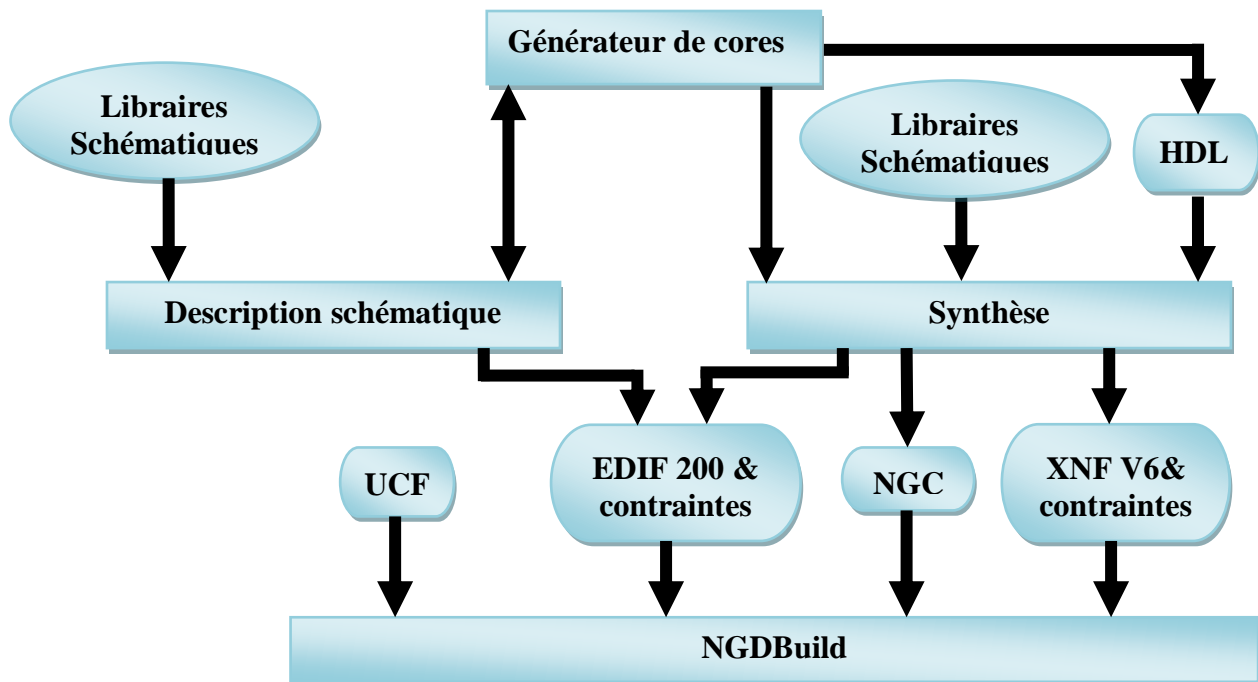


Figure II.12 :Flot de synthèse sous Xilinx [11]

### II.2.2.2. Simulation fonctionnelle :

Après sa conception et sa description, la simulation fonctionnelle teste la logique du design pour détecter et corriger tout dysfonctionnement dû à une mauvaise conception ou à une description erronée. Le logiciel utilisé pour la simulation fonctionnelle est le *ModelSim*.

### II.2.2.3. Translation en NetList :

C'est le format requis par les outils de développement de Xilinx pour compléter le flot de conception est *NGD* (*Native Generic Database*). Deux outils sont disponibles pour effectuer la translation d'autres formats en format *NGD*. Par exemple *EDIF2NGD* traduit un fichier *EDIF* (*Electronic Data Interchange Format*) en *NGD* et le programme *NGDBuild* convertit plusieurs formats (*XNF*, ou *EDIF*) en *NGD*. A noter que ce dernier outil, utilise le premier pour la traduction des fichiers *EDIF* [11].

### II.2.2.4. Implémentation :

L'implémentation du design commence avec le *mapping* qui se traduit par l'association des éléments logiques avec les éléments physiques d'un composant donné, et se termine lorsque le design physique est routé totalement et traduit en série de bits chargeable dans le composant.

Des contraintes peuvent toujours être ajoutées durant le processus d'implémentation pour que le design satisfasse le cahier de charge. Le processus d'implémentation est illustré par la *Figure II.13*.

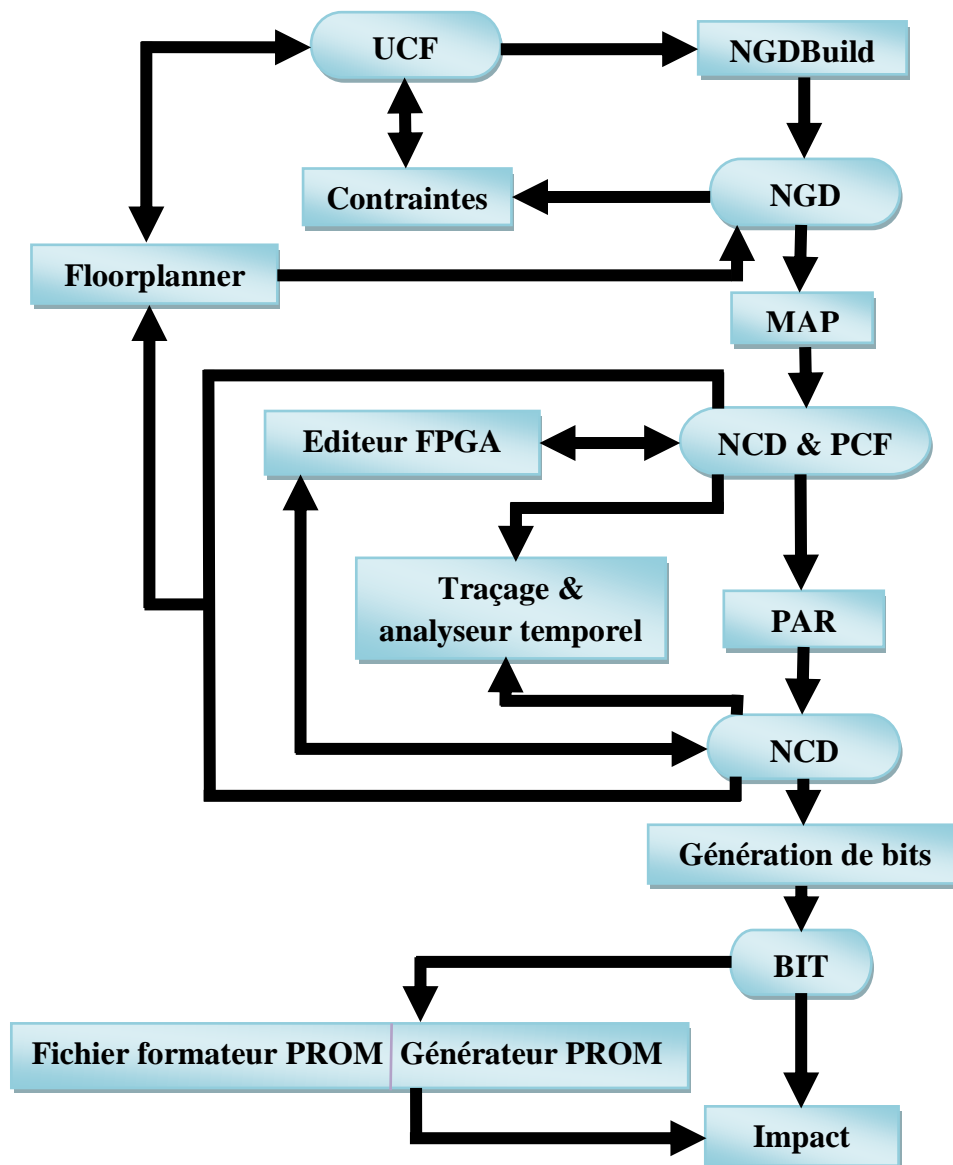


Figure II.13 : Implémentation sous Xilinx

### II.2.2.5. Mapping :

L'outil *MAP* arrange les éléments logiques d'un design dans un *FPGA* de *Xilinx* et les associe à des éléments physiques. L'outil *MAP* prend en entrée un fichier *NGD* qui contient une description logique du design en terme de composants hiérarchiques utilisés dans le design et de primitives de *Xilinx* de bas niveau, et d'un nombre quelconque de fichiers *NMC* (*hard placed and routed macros*), chacun contenant la définition d'un macro physique. *MAP* associe alors des composants (cellules logiques, cellules d'entrée/sortie, etc.) aux différentes parties logiques du design. La sortie du *MAP* est un fichier de description de circuit d'extension *NCD* (*Native Circuit Description*). Ce fichier est une représentation physique du design à l'aide des éléments constitutifs du *FPGA* cible. Le fichier *NCD* peut être placé et routé.

### II.2.2.6. Placement et Routage :

L'outil *PAR* (*Place And Route*) prend à l'entrée une description physique du design (un fichier *NCD*), le place et le route pour donner un autre fichier *NCD*, utilisable par *BitGen*, afin qu'il soit transformé en données de configuration. Le fichier *NCD* de sortie peut être aussi utilisé comme fichier guide pour refaire le placement routage après avoir effectué de petits changements au design. L'outil *FPGA editor* permet de placer et router des composants critiques avant le lancement du placeur/routeur automatique, et permet aussi de modifier manuellement le placement et le routage.

### II.2.2.7. Génération de flot de bits :

L'outil *BitGen* produit une série de bits pour la configuration d'un *FPGA* de *Xilinx*. *BitGen* prend un fichier *NCD* totalement placé et routé pour générer la série de bits de configuration sous la forme d'un fichier d'extension *.bit*. Le fichier *BIT* contient toutes les informations contenues dans le fichier *NCD* (la logique interne et les interconnexions entre les éléments du *FPGA*), et des informations caractéristiques du composant cible à partir d'autres fichiers associés avec le composant cible. Après la génération du fichier *BIT*, il peut être chargé dans le *FPGA* à l'aide de l'outil *iMPACT*. Un fichier *PROM* peut aussi être créé à partir du fichier *BIT*, dans le but de le charger dans une *PROM*, afin d'être utilisé ultérieurement par le *FPGA*.

### II.2.2.8. Vérification :

La vérification du design est un processus de test des fonctionnalités et des performances du design. *Xilinx* offre plusieurs méthodes pour ce but :

- La simulation fonctionnelle et temporelle ;
- Une analyse temporelle statique ;

- la vérification sur le circuit.

La procédure de vérification du design se fait normalement comme l'illustre la figure :

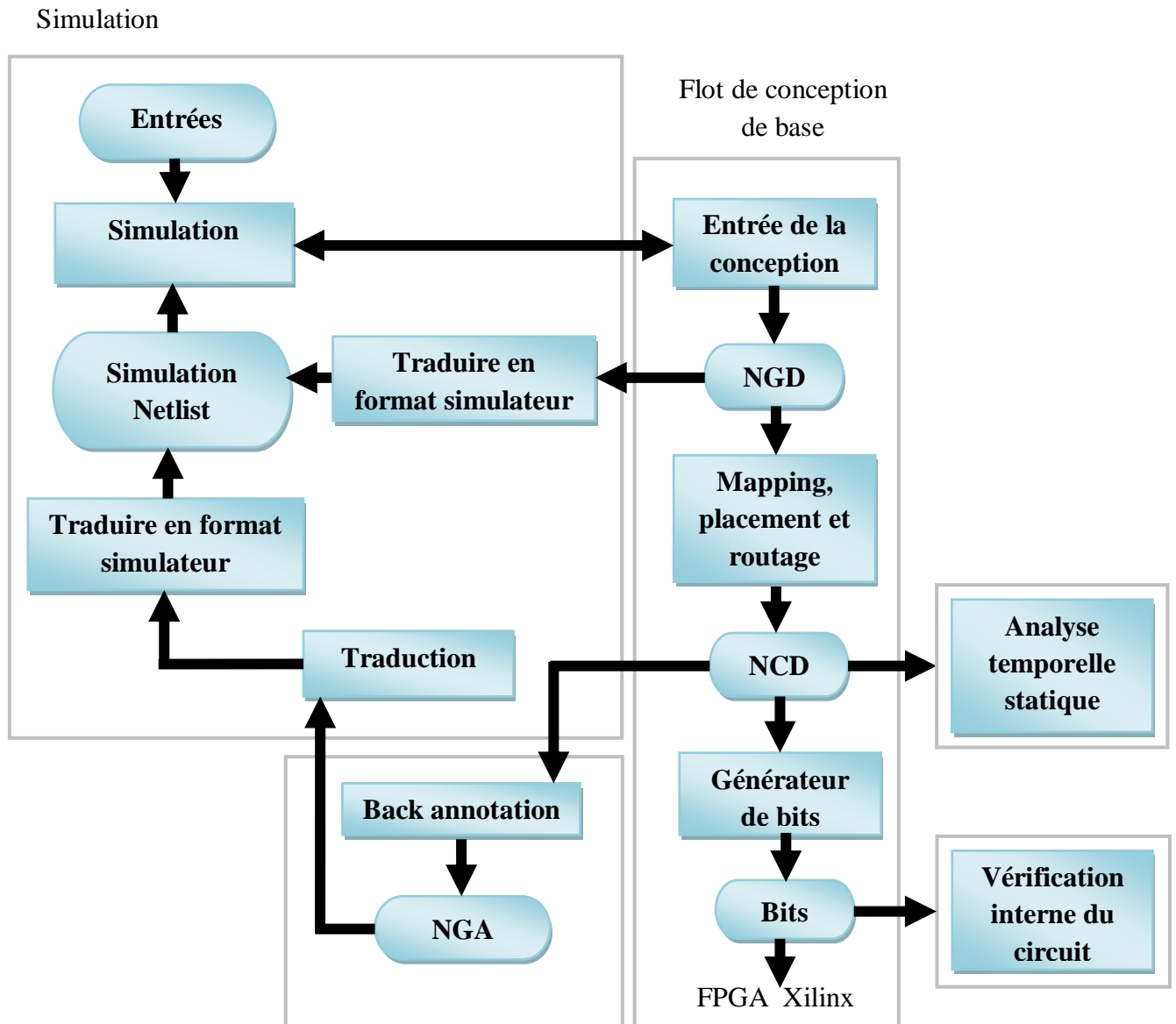


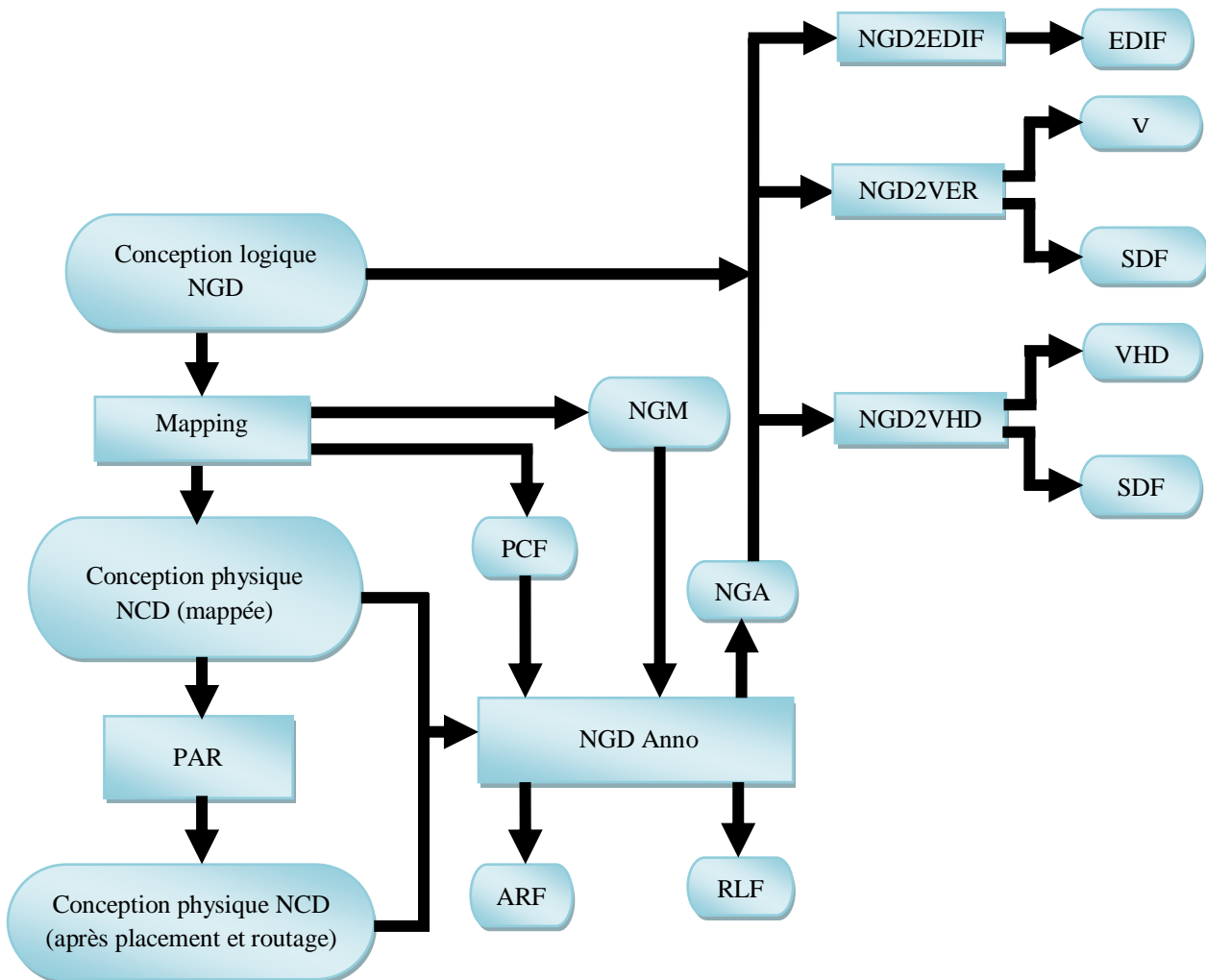
Figure II.14 :Le processus de vérification sous Xilinx

### II.2.2.9. Le processus « Back annotation » :

Pour vérifier le design, une simulation fonctionnelle ou temporelle peut être exécutée. Un processus de *Back-Annotation* doit avoir lieu avant la simulation temporelle. Avant la phase de la simulation temporelle, la description physique du design doit être traduite en



design logique compréhensible par le simulateur. Cette tâche est appelée *Back-Annotation process* en anglais. Un outil appelé *NGDAnno* est chargé de créer une base de données afin qu'elle soit utilisée par l'outil créateur de *netlist*, qui traduit ces informations en format *netlist* compréhensible par le simulateur. La figure suivante illustre le *Back-Annotation process*.



**Figure II.15 :**Le processus « Back-Annotation »

*NGDAnno* est un outil qui traduit les caractéristiques physiques (délais, temps de propagation des signaux, temps de réponses, etc.) trouvées dans un fichier de description de circuit *NCD*, en fichier de description logique du design *NGD* (en tenant compte de toutes les caractéristiques physiques du circuit). Le fichier *NCD* à l'entrée peut être un design totalement ou partiellement placé et routé, ou qui n'est ni placé ni routé (*NCD* à la sortie de l'outil *MAP*). Un fichier *NGM* créé aussi par *MAP* est une entrée optionnelle pour *NGDAnno* qui fusionne les informations de *mapping* contenues dans ce dernier fichier avec les informations de placement, routage et les informations temporelles provenant du fichier *NCD* employé. A la sortie, cet outil génère un fichier *NGA* (*Native Generic Annotation*) qui est un

fichier *NGD* reconstruit à partir d'un *NCD*. Ce fichier constituera une entrée pour l'outil de génération de netlist approprié, qui convertira le design du format *NGA* binaire, en format netlist *ASCII*. Les *netlist writers* (*NGD2EDIF*, *NGD2VER* ou *NGD2VHDL*) prennent la sortie de l'outil *NGDAnno* et créent une *netlist* de simulation au format spécifié. Un fichier *NGD* ou *NGA* peut être une entrée d'un *netlist writer*. *NGD2EDIF* traduit un fichier *NGD* ou *NGA* en *netlist* au format *EDIF* (fichier *EDN*). *NGD2VER* traduit un fichier *NGD* ou *NGA* en netlist au format *Verilog* (fichier *V*). Cet outil génère aussi un autre fichier *SDF* (*Standard DelayFormat*) qui contient des informations temporelles qui est utilisable uniquement avec le fichier *Verilog* créé par le même outil, à partir du même fichier *NGD* ou *NGA*. *NGD2VHDL* traduit un fichier *NGD* ou *NGA* en *netlist VHDL* (fichier *VHD*). Si le fichier d'entrée est du format *NGA*, cet outil génère aussi un fichier *SDF* spécifique pour ce fichier *VHD*.

#### **II.2.2.10. Simulation fonctionnelle :**

La simulation fonctionnelle ou comportementale détermine si la logique du design est correcte avant la phase d'implémentation. Ce type de simulation peut avoir lieu tôt dans le processus de conception du système. Et puisque les informations temporelles ne sont pas disponibles à ce moment, le simulateur teste la logique en utilisant des délais élémentaires comme unités. Le simulateur utilisé (*ModelSim*) est un simulateur intégré dans l'environnement de développement de *Xilinx* : le passage entre les outils de conceptions (éditeurs *HDL* ou schématiques) et le simulateur se fait automatiquement sans besoin d'utilisation intermédiaire d'outils de translation.

#### **II.2.2.11. Simulation temporelle :**

La simulation temporelle examine le temps d'exécution du design dans les pires conditions. Ce processus peut avoir lieu après le *mapping*, le placement et le routage du design. A ce moment-là, tous les délais du design sont bien connus. La simulation temporelle est très importante parce qu'elle peut vérifier les relations temporelles et détermine les chemins critiques du design dans les pires conditions. Avant la simulation temporelle, il faut passer par le *Back-Annotation process* déjà mentionné.

#### **II.2.2.12. Analyse temporelle statique :**

L'analyse temporelle statique est favorisée dans le cas de vérification rapide d'un design après placement et routage. Elle permet de déterminer les délais des chemins du design. Les deux buts majeurs de l'analyse temporelle sont : la vérification temporelle (vérifier que le design obéit aux contraintes temporelles) et le rapportage (décrire le

comportement temporel d'une façon technique qui peut être utilisée comme documentation pour évaluer le design). L'outil *TRACE (Timing Reporter And Circuit Evaluator)* est responsable de faire cette analyse.

#### **II.2.2.13. Vérification sur le circuit :**

La vérification du comportement de design dans l'application cible est considérée comme un test final. La vérification du design sur le circuit test, circuit dans les conditions typiques de fonctionnement. Ce type de test est simple du fait que les composants de *Xilinx* sont reprogrammables (plusieurs itérations du design peuvent être essayées).

Avant la création du fichier *BIT* final, il est utile d'utiliser l'option *DRC* dans l'outil *BitGen* pour évaluer le fichier *NCD*, et chercher les problèmes qui peuvent empêcher le design de fonctionner correctement sur le composant cible.

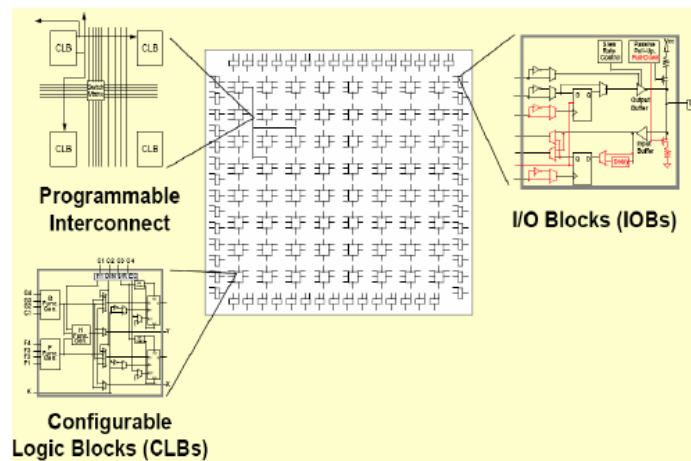
### **II.3. Architecture d'un Circuit FPGA**

Une structure *FPGA* est composée généralement de deux blocs. Des sous blocs d'entrées/sorties et des sous blocs logiques programmables. Quelques circuits *FPGA* possèdent aussi des mémoires *RAM* et des noyaux de processeurs.

De nombreux constructeurs élaborent des structures *FPGA* dont la technologie diffère d'un constructeur à un autre. Nous nous intéresserons dans ce qui suivra à des structures *FPGA* développées par la société *XILINX*. Une structure *FPGA* construite par *XILINX* se présente sous forme de deux couches :

#### **II.3.1/ Le circuit configurable :**

Elle est constituée de blocs logiques configurables (*CLB*), elle contient des fonctions combinatoires et des fonctions séquentielles. Autour de ces circuits configurables existent des blocs d'entrées/sorties (*IOB*) qui ont pour rôle la gestion des entrées et des sorties réalisant ainsi une interface entre le circuit et les modules extérieurs. La figure suivante illustre ces différents blocs.



**Figure II.16 :** Architecture interne du FPGA.

### II.3.2/ La couche réseau mémoire

C'est dans cette partie que la configuration du circuit est mémorisée, on déduit facilement que la programmation d'un circuit *FPGA* est volatile puisqu'elle s'enregistre sur une *SRAM*, mais le contenu est stocké tout de même sur une *ROM* externe. Le processus est simple ; à chaque mise sous tension, la *SRAM* interne est chargée à partir de la *ROM* externe. Cette caractéristique fait que le contenu interne n'est jamais définitif, ainsi un même circuit peut être exploité avec des *ROM* différentes, de ce fait une erreur quelconque en phase de programmation est facilement corrigible.

## II.4. Le module ADM-XRC-II

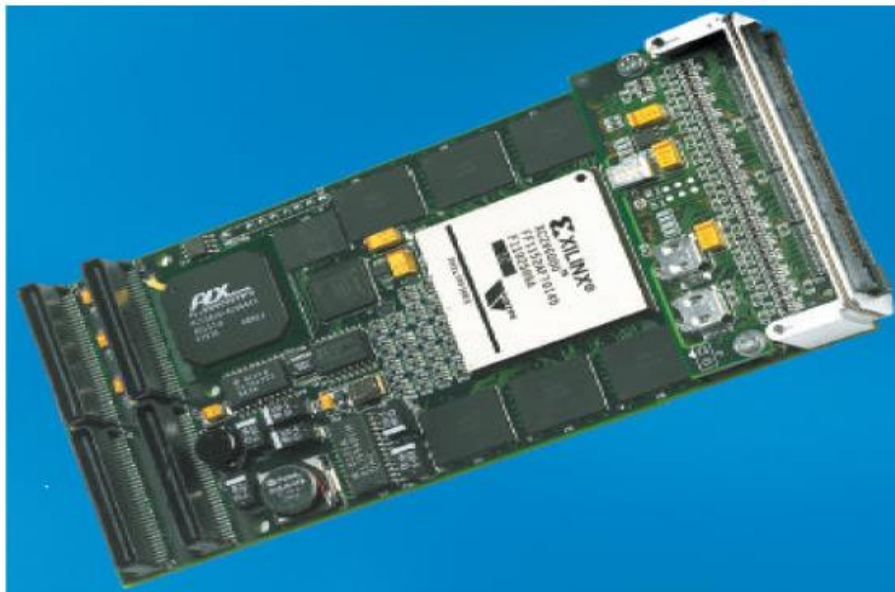
### II.4.1/ Présentation de la plateforme ADM-XRC-II

La plateforme *ADM-XRC-II* [43] utilise un *FPGA* de type *Xilinx Virtex-II XC2V6000* (Tableau II.2). Ce type de composant est très utile pour les applications de télécommunications car en plus de sa grande densité d'intégration, sa faible consommation et sa haute fréquence de travail, le *Virtex-II* possède des entrées/sorties compatibles pour les interfaces *LVDS* et le *PCI*. Le *FPGA Virtex-II XC2V6000* employé par la plateforme est un boîtier de type *FF1152* (boîtier BGA de 35mm x 35mm avec 1152 broches).

Les caractéristiques techniques de ce composant (Tableau II.2) sont les suivantes :

- Nombre de portes : 6 Millions
- Horloge interne maximale : 420 MHz
- Alimentation du cœur  $VCC_{int} = 1,5V$
- 824 entrées/sorties
- Il possède 16 multiplexeurs d'horloge interne.

La particularité de ce type de technologie (famille *FPGA Virtex-II*) est d'accepter la reconfiguration dynamique partielle. C'est-à-dire qu'en cours d'exécution d'une application sur le *FPGA*, nous pouvons reconfigurer une ou plusieurs zones sans devoir reconfigurer entièrement le *FPGA*. Dans ce cas, nous configurons le *FPGA* avec une IP de processeur possédant des paramètres déterminés, ensuite au cours de l'exécution nous modifions une partie de ce processeur avec des paramètres différents. Ceci permet un gain de temps au niveau de la phase de la synthèse de design (re-synthétiser et configurer uniquement la partie concernée sur l'*FPGA*).



**Figure II.17** : Plateforme de prototypage ADM-XRC-II (Alpha Data)

D'autre part, le module *ADM-XRC-II* (Figure II.17) intègre une interface *PCI* (à travers un connecteur *PMC*), permettant à un *PC*, de reconfigurer et/ou communiquer avec le *FPGA*. Avec ce module, sont fournis tous les drivers et bibliothèques, permettant au *PC* de communiquer avec celui-ci sous les environnements *Linux*, *Windows*...

#### **II.4.2/ Spécifications techniques du module ADM-XRC-II**

Le module *ADM-XRC-II* supporte les hautes performances du bus *PCI* sans avoir à utiliser une *IP PCI* dans le *FPGA*. Une interface *PCI Hard PLX 9656* [46] gère les communications entre le *FPGA* et le *PC*. A travers cette interface *PCI*, gère aussi les commandes du module tel que l'horloge programmable.

Les caractéristiques techniques sont les suivantes :

- Bus *PCI* 64bits – 66 MHz

- 6 banques indépendantes de *ZBT SSRAM* 256K x 32bits
- 64 entrées/sorties via un connecteur *PMC*.
- 146 entrées/sorties via un connecteur *XRM*.
- Sur la carte, une horloge programmable fournit au bus local (entre *FPGA* et interface *PCI*) une fréquence variable : 400kHz à 66 MHz.
- Une autre horloge programmable fournit une fréquence au *FPGA* : 0 à 100 MHz.
- Possibilité de connecter 256 MO de *DDRSDRAM* via le connecteur *XRM*.

## II.5. Le langage VHDL :

Le développement colossal qu'avait connu l'électronique a fait que la description d'un circuit programmable a connu un saut dans le temps, il fut un temps, en effet, les techniques utilisées pour traduire une structure en un circuit électrique étaient des outils de saisie de schémas, ou des langages de bas niveaux (*ABEL, PALASM, ORCAD/PLD, ...*). Actuellement, on ne peut plus parler de ces outils simples pour une simple raison ; les circuits d'aujourd'hui ne sont plus ceux d'autre fois ; ils sont plus compliqués et beaucoup plus denses. C'est à partir de là que le besoin de langages de hauts niveaux a eu lieu à savoir le *VHDL* et le *VERILOG*.

Le programme *VHSIC* (*Very High Speed Integrated Circuits*), développé par le département de la défense des Etats-Unis dans les années 1970-1980 a donné naissance au langage *VHSIC-HDL* (*VHSIC Hardware Description Language*) ou *VHDL*. En 1987, la première version du langage a été standardisée par l'*IEEE* (*Institute of Electrical and Electronics Engineers*) portant ainsi le nom *IEEE Std. 1076-1987 (VHDL 87)*, en 1993 quelques ambiguïtés ont été supprimées et de nouvelles commandes ont été mises en place dans la version (*IEEE Std. 1076-1993 ou VHDL-93*), en 1997, de nouvelles fonctions pour la conception des circuits ont été ajoutées. La dernière version du *VHDL* est celle de 2002 nommée (*IEEE Std. 1076-2002 ou VHDL-2002*).

Un langage de description Hardware (*HDL*) offre donc la possibilité de description à plusieurs niveaux dans laquelle les portes élémentaires et les *netlists* peuvent être utilisés avec une description fonctionnelle. Cette variation de niveau permet de décrire l'architecture du système à un niveau d'abstraction plus élevée, puis d'une façon incrémentale raffiner l'implémentation du design jusqu'au niveau des portes.

La description HDL offre les avantages suivants : la fonctionnalité du design peut être vérifiée immédiatement ce qui permet d'évaluer les décisions architecturales, une description

HDL est lue et comprise plus facilement qu'une *netlist* ou une description schématique et elle constitue une documentation du design et de sa fonctionnalité indépendante de la technologie, et sans oublier qu'une description *HDL* facilite la manipulation de larges designs par rapport à une description schématique.

### II.5.1/ Structure d'une description VHDL

Une structure *VHDL* comporte généralement deux parties indissociables à savoir l'entité (*ENTITY*) et l'architecture (*ARCHITECTURE*). La première est définie comme un bloc contenant des entrées et des sorties ; c'est une vue externe de la description. L'architecture contient les instructions écrites en *VHDL* ayant pour but la réalisation du fonctionnement attendu ; c'est la description interne de la structure.

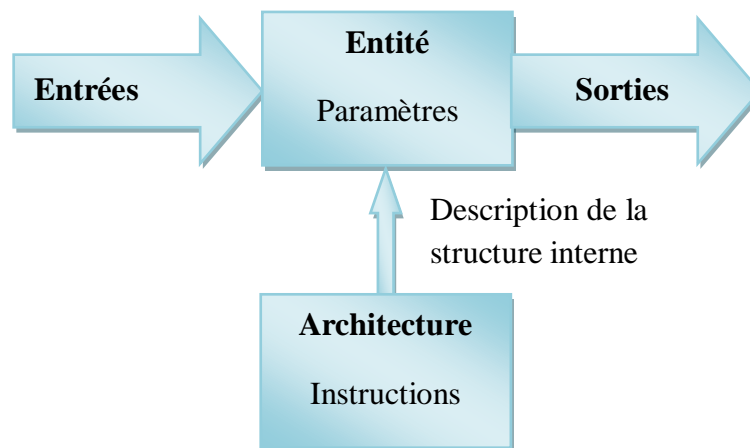


Figure II.18 : Structure de base d'une description VHDL

#### II.5.1.1. Déclaration des bibliothèques :

Les bibliothèques sont un élément indispensable pour toute description VHDL. Elle contiennent les définitions des signaux électroniques, des fonctions et sous programmes à utiliser pour réaliser les opérations arithmétiques et logiques. La commande *use* permet de désigner les bibliothèques à utiliser.

#### Exemple :

```
library ieee;
```

```
use ieee.std_logic_1164.all;
```

```
use ieee.std_logic_arith.all;
```

```
use ieee.std_logic_unsigned.all;
```

### II.5.1.2. Déclaration de l'entité et des entrées/sorties :

C'est une opération qui permet de définir le nom de la description et les entrées et sorties utilisées, l'instruction utilisée est *port*.

Exemple :

```
entityessai
```

```
Port (a:in std_logic;  
      b: in std_logic;  
      y: out std_logic  
      );  
endessai;
```

Un signal peut être défini en entrée (*in*), en sortie (*out*), en entrée/sortie (*inout*) ou en sortie pouvant être utilisé en entrée à l'intérieur de la description (*buffer*). Le langage VHDL ne fait pas la distinction entre la majuscule et la minuscule.

Un signal peut être :

- *std\_logic* pour un signal à un bit ;
- *std\_logic\_vector* pour un registre composé de plusieurs bits.

### II.5.1.3. Déclaration de l'architecture correspondante à l'entité :

L'architecture définit le fonctionnement tracé pour un circuit donné. Ce sont tout simplement des instructions écrites en langage *VHDL* décrivant les différentes relations entre les entrées et les sorties.

Exemple:

```
architecture Behavioral of essai is
```

```
begin
```

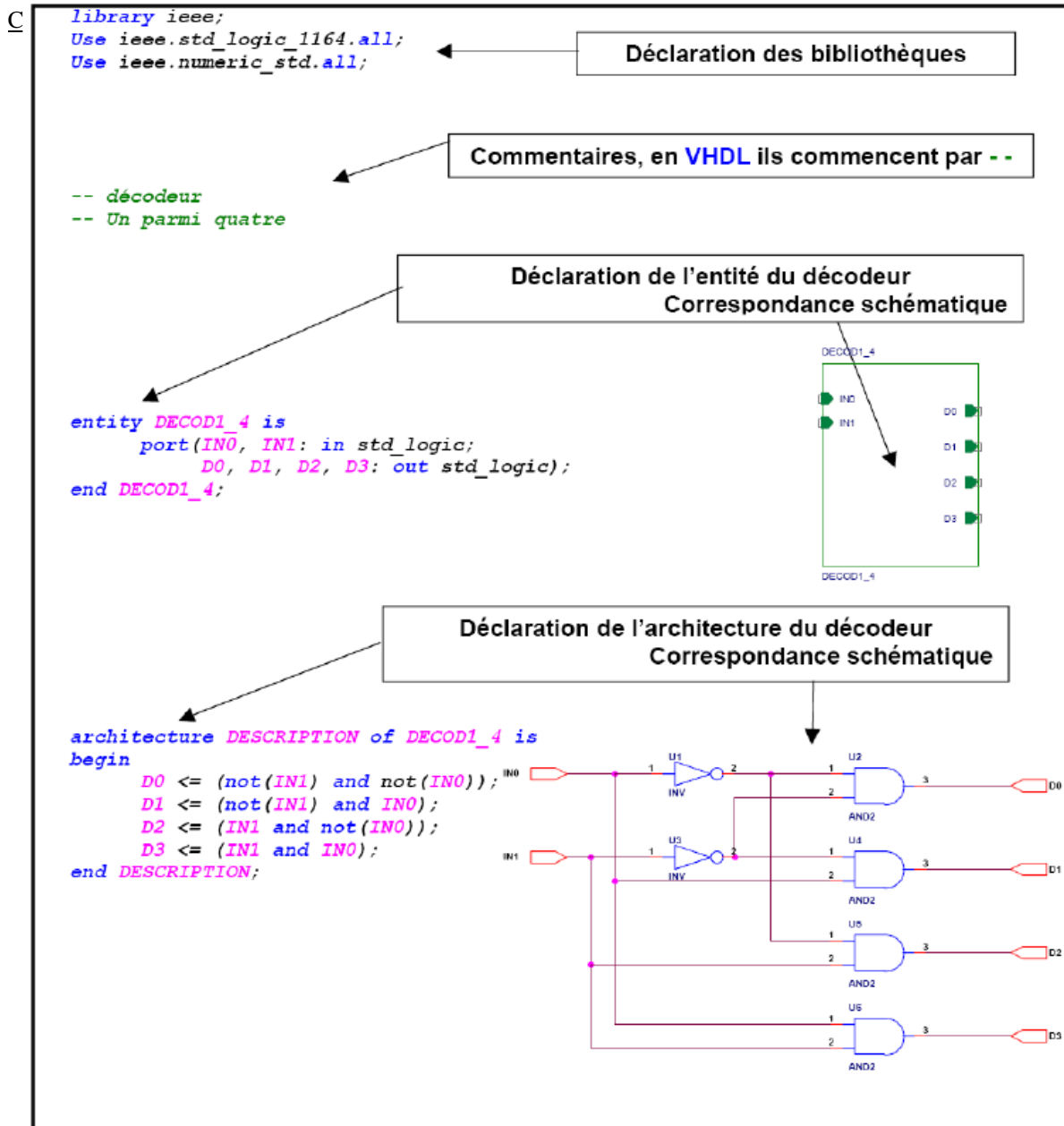
```
y<=a and b;
```

```
end Behavioral;
```

## II.5.2/ Les Opérateurs VHDL

L'opérateur le plus utilisé dans les structures *VHDL* est l'opérateur d'affectation (*<=>*). Il permet de modifier l'état d'un signal donné. L'opérateur (*&*) permet d'associer deux





signaux pour former un nouveau signal dont le nombre de bits est égal à la somme des signaux à associer.

Le VHDL dispose également des opérateurs logiques (*and*, *or*, *not*, *nor*, *xor*, ...), arithmétiques (+, -, \*, /) et relationnels (=, >, <, >=, <=, /=) qui sont prédéfinis dans les bibliothèques *ieee.std\_logic\_1164.all* et *ieee.numeric\_std.all*.

# *Chapitre III*

## *Analyse de performance de l'étage de détection*

- ❖ Introduction
  
- ❖ Acquisition
  
- ❖ Analyse des propriétés de Corrélation et de l'espace de  
recherche
  
- ❖ Détection
  
- ❖ Intégration
  
- ❖ Conclusion

### III.1. Introduction

Le signal GPS reçu, amplifié, filtré puis numérisé passe au traitement du signal numérique par l'opération de l'acquisition nécessaire pour sa démodulation. A ce niveau, les composantes de la variable qui interprète la présence du signal appartiennent au domaine de l'aléatoire. Dans ce chapitre nous allons traiter l'aspect des statistiques et de probabilités du signal, dans le bloc d'acquisition. Dans cette approche, nous allons définir deux domaines d'analyses stochastiques.

- ✓ Le premier adviendra du domaine des cellules de recherche. Ces dernières sont formées d'intervalles de temps et de fréquences soit une grille de recherche en deux dimensions. L'analyse du niveau du signal révèle la combinaison temps et fréquence du signal reçu après le test de détection, qui sera le deuxième domaine. *Le signal* étant mélangé au bruit, le paramètre signal sur bruit sera la métrique de référence
- ✓ Dans le domaine de détection, les cellules sont soumises à un test qui détermine la présence ou l'absence du signal recherché. La variable dans la cellule dite de décision [3][8], résulte de différentes opérations élémentaires et suit des lois de distributions bien connu pour plusieurs détecteurs et que l'on peut démontrer. L'analyse de cette variable sous deux hypothèses, étant l'absence et la présence d'un signal utile respectivement, permettra d'analyser les performances de différents détecteurs selon les opérations établies, et cela en termes de probabilité de détection et de fausse alarme en fonction de condition du niveau du signal reçu.

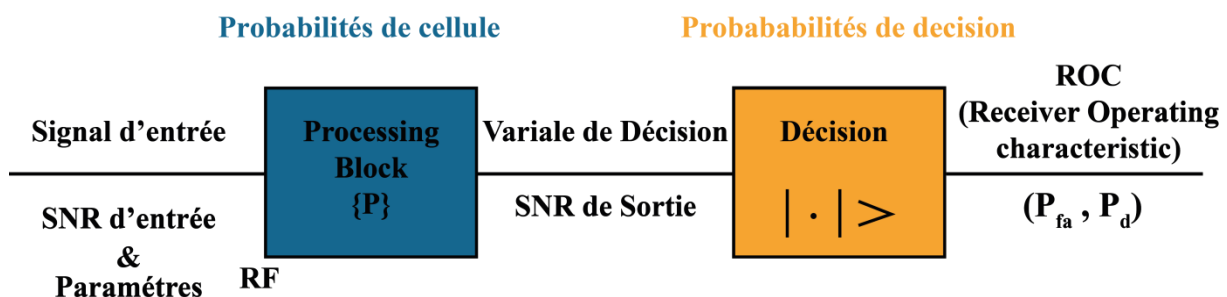


Figure III.1 : Schéma bloc de l'étage de détection.

## III.2. Acquisition

La première étape du traitement logiciel d'un récepteur GNSS consiste en l'acquisition du signal, qui détermine la présence d'un signal parmi plusieurs signaux satellitaires, et garantit une estimation rugueuse de ses paramètres notamment, le temps entre l'émission et la réception, qui permettra par la suite d'estimer la distance entre le satellite et le récepteur. Le deuxième paramètre est la déviation de fréquence entre celle du signal émis et reçu, cette déviation doit être compensée pour pouvoir appliquer les opérations élémentaires dans le bloc, afin d'étudier le signal et relever les paramètres désirés. Ces paramètres vont être utilisés pour calculer les pseudo-distances et estimer la position du récepteur [8].



**Figure III.2 :** Schéma simplifié du bloc d'acquisition.

Plusieurs techniques d'implémentations et méthodes de recherches sont approuvées, et engendrent des estimations très précises, il n'empêche que chaque méthode a des avantages et inconvénients et donc le choix de la technique dépend des performances requises par l'application à réaliser [8] [3]. Les performances de l'opération d'acquisition peuvent se résumer en [1] :

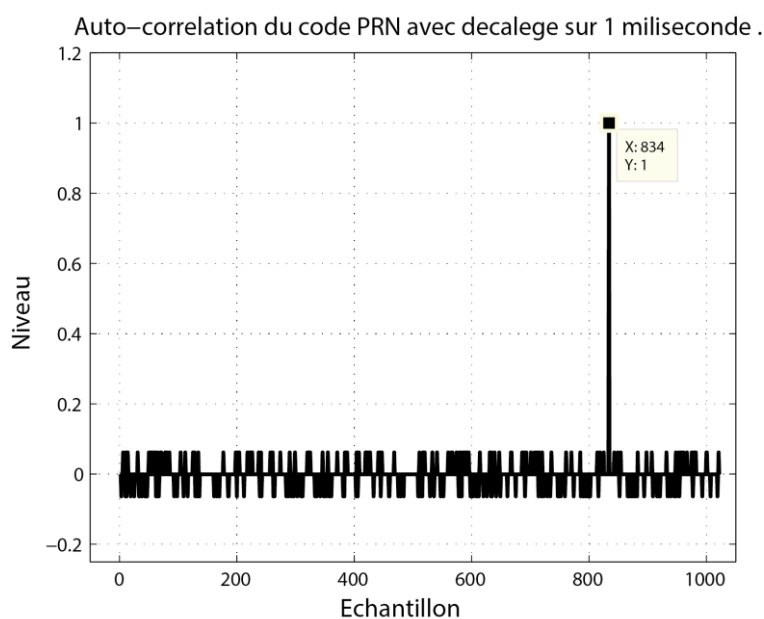
- ✓ *La sensibilité* de détection, ou le niveau de puissance minimum du signal reçu pour l'acquérir en fonction de probabilités de détection désirées.
- ✓ *Le temps d'opération* de l'acquisition.

Afin de définir les éléments qui contribuent à la variation de ces deux paramètres, toute la chaîne de transmission doit être évaluée, à savoir l'émission du signal au niveau du satellite, la propagation atmosphérique du signal et la chaîne de réception jusqu'à détection, une partie étant réalisée dans le **chapitre I**. La métrique de référence utilisée pour évaluer la sensibilité est le rapport signal sur bruit, cela permettra d'orienter notre étude de manière homogène. Ainsi on pourra déterminer un *SNR* minimum pour une probabilité de détection voulu.

### III.3. Analyse des propriétés de Corrélation et de l'espace de recherche

#### III.3.1. Corrélation

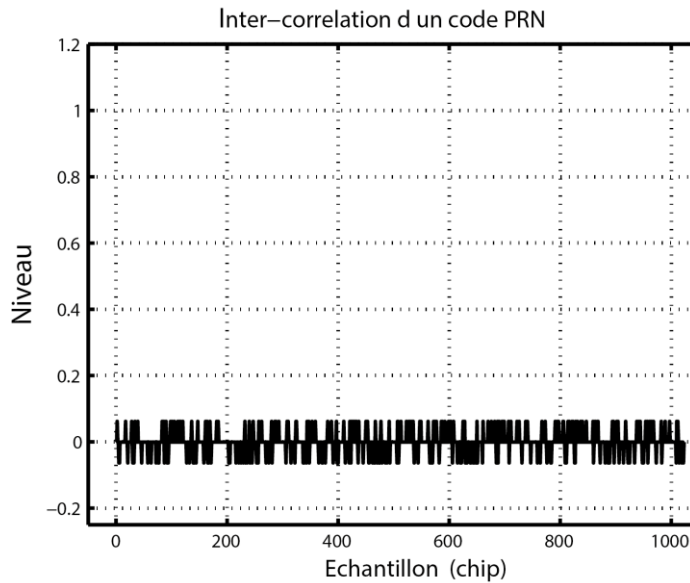
La corrélation est l'opération basique du traitement logiciel des récepteurs *GNSS*, et dépend essentiellement des caractéristiques du code d'étalement. C'est l'opération qui compare le signal reçu avec un signal généré localement, ainsi le système peut garantir la reconnaissance des satellites visibles, ainsi que l'estimation du retard et de la déviation de la fréquence due aux phénomènes vus précédemment. L'obtention de ces informations revient aux propriétés de corrélation générant un pic pour un alignement des deux signaux de séquences identiques.



**Figure III.3 :** *Auto corrélation de deux codes PRN similaires décalés.*

La **Figure III.3** Montre un pic de corrélation au chip  $\tau = X - 1$  ; et de valeurs presque nulles pour les autres décalages, ce qui révèle que les deux codes sont alignés après un temps  $t = 834 -$

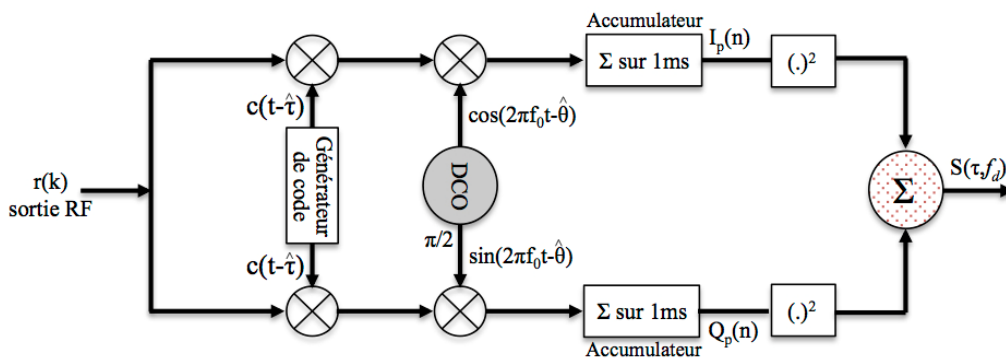
1



**Figure III.4 :** *Inter corrélation de deux codes PRN différents.*

On distingue que le résultat de la corrélation de deux codes PRN différents dans la **Figure III.4** a des valeurs se rapprochant de 0 pour les 1022 décalages possibles, qui révèle qu'il n'y a pas d'alignement des deux codes. On conclut à partir des deux simulations que l'alignement d'un code généré avec un code reçu peut nous révéler de quel code s'agit ou de quel satellite reçoit-on le signal ou non, ainsi que le retard de transmission.

L'opération de corrélation commence par la multiplication du signal reçu par une réplique locale de la porteuse et du code d'étalement, le résultat du signal échantillonné est ensuite intégré pour un temps donné (plusieurs instances du signal pourront être prises en comptes pour la décision finale). Deux signaux se distinguent de la corrélation, le signal *I* qui advient de la multiplication du signal par une porteuse en phase avec le signal généré au niveau du récepteur et un signal *Q* où la porteuse locale est en quadrature avec la première.



**Figure III.5 :** *Corrélateur utilisé dans l'étape d'acquisition.*

### III.3.1.1. Etapes de la corrélation :

- **1ere étape :** Multiplier le signal d'entrée par les codes PRN générés localement dans le but de déterminer le degré de similitude entre le code du signal reçu et les 32 codes locaux c'est l'opération de corrélation. Si le code généré est le même que celui reçu un pic en forme de triangle (propriété de l'auto corrélation d'un signal carré) émergera indiquant que le code généré est le même que celui reçu et donc déterminer le satellite émetteur. Les codes locaux sont décalés d'un chip ,1022 fois, afin de centrer le code reçu qui est décalé (entre 0-1022) avec le code (correct) généré.

$$r[n] = \{A_i c_i[n - \tau_0] d_i[n - \tau_0] \cos[2\pi F_D n + \varphi] + \eta_{IF}[n]\} \times c(t - \hat{\tau}) \quad (\text{III.2})$$

Si le code généré utilisé n'est pas aligné avec le signal ou n'est pas le même que celui reçu, le résultat de la corrélation sera proche de zéro vu les caractéristiques de la corrélation du code PRN.

- **2ème étape :** Cette étape consiste en la multiplication du signal reçu avec une porteuse générée localement à l'aide d'un *DCO*, dans le but d'annuler cette porteuse et récupérer le signal modulant. Cela dit, entre les mouvements des satellites et ceux du récepteur (effet doppler), qui engendrent une déviation de la fréquence de réception par rapport à la fréquence d'émission, et les résidus additionnels de fréquence dus aux transitions du signal à travers les dispositifs électroniques qui engendrent une autre déviation de la fréquence  $f_{IF}$ , les déviations résumées en la fréquence doppler noté  $f_D$ , étant inconnu et se situant entre -5 et 5khz (récepteur statique) il suffit de chercher dans cette marge la fréquence correcte, pour déduire le décalage.

$$\begin{aligned} I[n] &= \{A_i c_i[n - \tau_0] d_i[n - \tau_0] \cos[2\pi F_D n + \varphi] + \eta_{IF}[n]\} \times c(t - \hat{\tau}) \times \cos(2\pi(f_{IF} + f_{d,i})n) \\ Q[n] &= \{A_i c_i[n - \tau_0] d_i[n - \tau_0] \cos[2\pi F_D n + \varphi] + \eta_{IF}[n]\} \times c(t - \hat{\tau}) \times \sin(2\pi(f_{IF} + f_{d,i})n) \end{aligned} \quad (\text{III.3})$$

Avoir un niveau maximum de corrélation peut révéler la  $f_D$  adéquate en notant la fréquence ou le pic est à son maximum. Pour  $\tau = \hat{\tau}$  et  $f_D = \hat{f}_D$  on aura les équations des deux chaînes  $I$  et  $Q$  suivantes [8] :

$$\begin{aligned} I[n] &= A_i d_i(\Delta\tau) c_i(\Delta\tau) \cos(2\pi\Delta f_D n + \Delta\theta) + \eta_I[n] \\ Q[n] &= A_i d_i(\Delta\tau) c_i(\Delta\tau) \sin(2\pi\Delta f_D n + \Delta\theta) + \eta_Q[n] \end{aligned} \quad (\text{III.4})$$

**Remarque :** Le signal reçu est déphasé à son arrivée au détecteur, et lors du traitement des étages précédents, c'est pour cela que la porteuse locale est générée en phase et en

quadrature ainsi l'énergie du signal peut apparaitre dans l'un ou les deux canaux  $I$  et  $Q$ . c'est l'effet de la variation de la phase [2].

$$\begin{aligned} I(n) &= \frac{A_i}{2T_c} d_i(t-\tau) R_c(\Delta\tau[n]) \cos(\pi\Delta f_D[n]T_c + \Delta\theta[n]) \text{sinc}(\pi\Delta f_D[n]T_c) + n_I[n] \\ Q(n) &= \frac{A_i}{2T_c} d_i(t-\tau) R_c(\Delta\tau[n]) \sin(\pi\Delta f_D[n]T_c + \Delta\theta[n]) \text{sinc}(\pi\Delta f_D[n]T_c) + n_I[n] \end{aligned} \quad (\text{III.5})$$

Sinc : sinus cardinal.

En réalité on ne peut pas décomposer le bruit du signal utile, donc on modélisera à présent le signal sous sa forme statistique. Le signal soumis au test de détection peut s'écrire :

$$S(\hat{\tau}_i, \hat{f}_{d,i}) = \left| \frac{1}{N} \sum_{n=0}^{N-1} r(n) \cdot c_i(n - \hat{\tau}_i) \cdot e^{-j2\pi\hat{f}_{d,i}n} \right|^2 \quad (\text{III.6})$$

Ainsi, la variance et la moyenne de la variable de détection peut se calculer en prenant la partie réelle et imaginaire pour en déduire le résultat des chaînes  $I$  et  $Q$  respectivement [14] :

$$\begin{aligned} \text{Var} [I(\hat{\tau}_i, \hat{f}_{d,i})] &= \text{Var} \left[ R_e \left\{ \sum_{n=0}^{N-1} r(n) \cdot c_i(n - \hat{\tau}_i) \cdot e^{-j2\pi\hat{f}_{d,i}n} \right\} \right] \\ \text{Var} [I(\hat{\tau}_i, \hat{f}_{d,i})] &= \text{Var} \left[ \frac{1}{N} \sum_{n=0}^{N-1} r(n) \cdot c_i(n - \hat{\tau}_i) \cdot \cos(2\pi\hat{f}_{d,i}n) \right] \\ \text{Var} [I(\hat{\tau}_i, \hat{f}_{d,i})] &= \frac{1}{N^2} \sum_{n=0}^{N-1} \text{Var} [r(n) \cdot c_i(n - \hat{\tau}_i) \cdot \cos(2\pi\hat{f}_{d,i}n)] = \frac{1}{N^2} \sum_{n=0}^{N-1} \frac{\sigma_{IF}^2}{2} \\ \text{Var} [I(\hat{\tau}_i, \hat{f}_{d,i})] &= \frac{\sigma_{IF}^2}{2N} = kT \frac{B}{2N} = kT \frac{fs}{4N} = \frac{N_0}{4Tc} \end{aligned} \quad (\text{III.7})$$

De même pour Q :

$$\begin{aligned} \text{Var} [Q(\hat{\tau}_i, \hat{f}_{d,i})] &= \text{Var} \left[ I_m \left\{ \sum_{n=0}^{N-1} r(n) \cdot c_i(n - \hat{\tau}_i) \cdot e^{-j2\pi\hat{f}_{d,i}n} \right\} \right] \\ \text{Var} [Q(\hat{\tau}_i, \hat{f}_{d,i})] &= \frac{\sigma_{IF}^2}{2N} = kT \frac{B}{2N} = kT \frac{fs}{4N} = \frac{N_0}{4Tc} \end{aligned} \quad (\text{III.8})$$

La variance correspond à la puissance du bruit à l'entrée du détecteur, sachant que la variance du signal utile recherché est nulle vu qu'il est supposé déterministe, à présent nous dénoterons la puissance du bruit :

$$\text{Var} [I(\hat{\tau}_i, \hat{f}_{d,i})] = \text{Var} [Q(\hat{\tau}_i, \hat{f}_{d,i})] = \sigma_n^2 \quad (\text{III.9})$$

La moyenne sert à déterminer les valeurs de non centralité des variables aléatoires, elle se calcule comme suit [14] :



$$\begin{aligned}
E\left[I(\hat{\tau}_i, \hat{f}_{d,i})\right] &= E\left[\sum_{n=0}^{N-1} r(n).c_i(n - \hat{\tau}_i).\cos(2\pi \hat{f}_{d,i}n)\right] \\
E\left[I(\hat{\tau}_i, \hat{f}_{d,i})\right] &= \frac{1}{N} \sum_{n=0}^{N-1} E[y_i(n) + \eta(n)].c_i(n - \hat{\tau}_i).\cos(2\pi \hat{f}_{d,i}n) \\
E\left[I(\hat{\tau}_i, \hat{f}_{d,i})\right] &= \frac{1}{N} \sum_{n=0}^{N-1} E[y_i(n)].c_i(n - \hat{\tau}_i).\cos(2\pi \hat{f}_{d,i}n)
\end{aligned} \tag{III.10}$$

Pour  $\tau = \hat{\tau}$  et  $f_D = \hat{f}_D$  on aura :

$$\begin{aligned}
\left[I(\hat{\tau}_i, \hat{f}_{d,i})\right] &= \frac{A_i}{N} \sum_{n=0}^{N-1} c_i^2(n - \hat{\tau}_i).\cos(2\pi \hat{f}_{d,i}n + \varphi).\cos(2\pi \hat{f}_{d,i}n) \\
E\left[I(\hat{\tau}_i, \hat{f}_{d,i})\right] &= \frac{A_i}{2N} \sum_{n=0}^{N-1} \cos(\varphi).\cos(4\pi \hat{f}_{d,i}n + \varphi)
\end{aligned} \tag{III.11}$$

Le résultat des moyennes des deux chaînes  $I$  et  $Q$  sont :

$$\begin{aligned}
E\left[I(\hat{\tau}_i, \hat{f}_{d,i})\right] &= \frac{A_i}{2} \cos(\varphi) \\
E\left[Q(\hat{\tau}_i, \hat{f}_{d,i})\right] &= \frac{A_i}{2} \sin(\varphi)
\end{aligned} \tag{III.12}$$

$A_i$  représente l'amplitude du signal utile. A partir des résultats de la moyenne qui représente l'amplitude du signal *GPS* et la variance qui indique la puissance du bruit, le rapport  $C/N_0$  en fonction de la fréquence d'échantillonnage, peut être déduit comme suit :

$$CNR = \frac{A^2}{4\sigma^2} = N \frac{A^2}{N_0 f_s} = \frac{A^2}{N_0} \times TC \tag{III.13}$$

La fréquence d'échantillonnage a été choisie à deux fois la bande passante.

### III.3.2. Espace de recherche

L'espace de recherche correspond à toutes les valeurs possibles que peuvent prendre les décalages temps et fréquences du signal reçu. Ainsi cet espace est partagé en intervalle sur deux axes,  $N_f$  intervalles de fréquences, et  $N_\tau$  intervalles pour le décalage temps, dans ce sens il y aura  $N_\tau * N_f$  cellules de recherche, chacune soumise au test de détection. La matrice de recherche est représentée par  $(f_i, \tau_j)$  ou  $i=1, \dots, N_f$  et  $j=1, \dots, N_\tau$  sont les valeurs **centrales** des cellules [3][8].

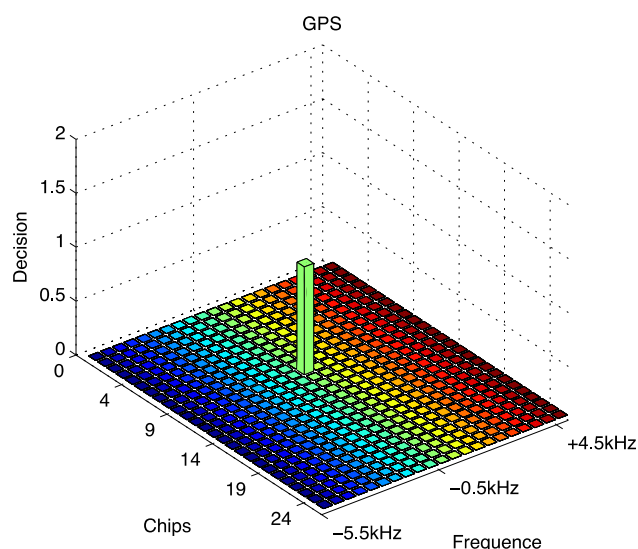
On ne peut pas assumer qu'une cellule est infiniment petite [2], cela induit une erreur  $\delta_f$  et  $\delta_\tau$  à partir du centre de la cellule contenant le signal utile et il va falloir l'analyser de plus près.

Des lors, un compromis doit être trouvé pour atteindre des performances voulues : entre la précision du calcul et la valeur du pic obtenu, advient de choisir des dimensions d'une cellule telle que la dimension est inversement proportionnelle aux nombre de cellules globales et donc au temps de recherche. De plus , on notera l'effet du résultat de la fonction de corrélation sur l'espace de recherche, en temps et en fréquence, qu'on appellera Matrice de recherche série [8] et sur quels principes nous nous baserons pour limiter les cellules de la matrice de recherche en tenant compte de la puissance que contient le signal utile dans la cellule ,ainsi que le temps de recherche. Afin de bien mettre en évidence notre travail on tiendra compte d'un seul satellite visible, puis on pourra généraliser l'étude pour plusieurs.

La plage de recherche de la fréquence doppler se situe entre  $-5$  et  $+5\text{kHz}$  et il est important de définir le pas, ce pas est relié à la longueur des données, pour  $1\text{ms}$ ,  $1\text{KHz}$  de déviation décale le code d'un bit, l'erreur maximum étant définie a  $0.55$  chip, un pas maximum de  $0.5\text{kHz}$  est nécessaire. La puissance du signal croit en fonction du nombre d'échantillon au carré et le bruit en fonction du nombre d'échantillon car il est non corrélé avec le code PRN( $n^2/n$ ) donc le SNR croit proportionnellement à  $n$  [17][2].

$$SNR_1 = n \frac{A^2}{\sigma^2} \quad (\text{III.14})$$

*Note : Dans le but de l'analyse on étudie le signal et le bruit séparément .On considère également dans cette analyse que le signe du bit de donnée ne change pas pendant le temps d'intégration puis on introduira la notion de « Data bit transition » en considérant les données dans l'intégration.*



**Figure III.6 : Résultat obtenu en acquisition GPS.**

### III.4. La détection du signal GPS :

Le but d'un détecteur de signal est de déterminer la présence ou l'absence d'un signal utile à partir d'un signal bruité. Le « *Processing Block* » définit la stratégie de détection par des paramètres comme [8] :

- La forme mathématique de la variable de décision.
- Temps d'intégration cohérente.
- Nombre d'intégrations non cohérentes.

A la sortie du bloc d'acquisition le signal est quantifié en une variable de décision  $D$ , qui est utilisée pour définir si un signal utile est présent. La variable de décision est une variable aléatoire caractérisée par deux *fonctions de densité de probabilité* qui se réfère à la présence ou l'absence du signal respectivement, on note :

$$S_{IF}[n] = \begin{cases} \eta[n] & \text{sous } H_0 \\ u[n] + \eta[n] & \text{sous } H_1 \end{cases} \quad (\text{III.15})$$

Où  $H_0$  et  $H_1$  sont les hypothèses de l'absence et présence du signal respectivement.  $\eta[n]$  : représente le bruit ;  $u[n]$  : représente le signal utile. On note  $\theta(f_D, \tau)$  le vecteur constituant les paramètres de l'acquisition  $f_D$  et  $\tau$ . En comparant la variable de décision  $D$  à un seuil  $\beta$  le système pourra calculer les probabilités tel que si  $D(\theta) > \beta$  alors il y a une détection correcte ou bien une fausse alarme. Cela se traduit par les équations suivantes :

$$\begin{cases} P_{FA}(\beta) = P(D(\theta) > \beta | H_0) \\ P(\beta) = P(X > \beta | H_0) = P(X > \beta | \tau \neq \hat{\tau} \cup f_D \neq \hat{f}_D) \end{cases} \quad (\text{III.16})$$

$$\begin{cases} P_d(\beta) = P(D(\theta) > \beta | H_1) \\ P(\beta) = P(X > \beta | H_1) = P(X > \beta | \tau = \hat{\tau} \cap f_D = \hat{f}_D) \end{cases}$$

Cela n'empêche pas qu'un signal soit détecté sans qu'il n'y est réellement de signal utile, la subtilité est là, C'est pour cela qu'on parlera en terme de probabilité de détection ou de fausse alarme en présence et en l'absence du signal utile, on utilisera le paramètre ROC pour définir la performance de la détection.

**Note :** Plusieurs stratégies ont été développées pour déterminer la cellule contenant le signal dans ce projet on a choisit de comparer les signaux des cellules à un seuil.

### III.4.1. Stratégie de détection et critère de décision

Plusieurs stratégies de détection peuvent être considérées pour la comparaison de l'amplitude pour les cellules de la matrice de l'acquisition à un seuil comme [8] :

- ✓ *La stratégie du maximum* : compare l'amplitude maximale seulement, au seuil.
- ✓ *La stratégie série* : ou toutes les cellules sont soumises au test.
- ✓ *La stratégie hybride* : l'évaluation se tient pour une multitude de cellules ensemble (lignes ou colonnes par exemple), puis la décision est prise en prenant le maximum des sélections.

Dans notre cas on utilisera la méthode la plus simple à modéliser et à implémenter qui est la stratégie série.

La théorie des tests permet de confronter deux hypothèses, formulées à priori, relatives à une ou plusieurs populations. Ces deux hypothèses font l'objet d'un test qui permet de contrôler leur validité à partir d'un ou plusieurs échantillons. Il y a plusieurs types de tests en fonction de l'hypothèse. On retiendra dans notre étude le test paramétrique composite (multiple) ou les hypothèses seront justes ou fausses en comparant ces paramètres à un seuil, dans ce sens on appliquera la règle de décision de *Neyman-Pearson*.

Dans l'optique de *Neyman et Pearson*, on accroît la dissymétrie du problème de test en considérant que l'erreur la plus grave consiste à rejeter à tort l'hypothèse nulle. On fixe donc un seuil maximum  $\alpha_0$  au risque de première espèce et on cherche un test qui minimise le risque de seconde espèce. Le test d'hypothèses est  $H_0 : \theta = \theta_0$  et  $H_1 : \theta = \theta_1$ . La règle de décision proposée par *Neyman et Pearson* consiste à déterminer la région critique pour laquelle la puissance (la probabilité de rejeter l'hypothèse nulle avec raison) est maximum sous la contrainte  $\alpha \leq \alpha_0$ .

### III.4.2. Les courbes ROC ( Receiver Operating Characteristics)

*ROC* pour « *Receiver Operating Characteristics* » représente les deux courbes des probabilités, de détection et de fausse alarme respectivement qui caractérisent la performance du détecteur et supprime les informations non pertinentes [17]. Les courbes ROC peuvent être utilisés pour comparer deux ou plusieurs détecteurs [16]. Elles peuvent également être utilisées pour explorer les conséquences de la variation du seuil d'un capteur unique. Bien entendu, le choix d'un seuil optimal

dépendra d'une variable de détection, et est utilitaire pour la sensibilité. Si la variable de décision est plus grande que le seuil résulte une correcte décision ou une fausse alarme les paramètres du *ROC* s'écrivent comme suit

$$P_{fa} = \int_{\beta}^{\infty} P(D / H_0) dD$$

$$P_d = \int_{\beta}^{\infty} P(D / H_1) dD$$
(III.17)

Étant donné que la courbe *ROC* dépend également de  $H_0$  et  $H_1$ , la modification de l'un ou l'autre changera également la forme de la courbe *ROC*. Cette fonctionnalité fournit un moyen pour comparer deux ou plusieurs capteurs, chacun faisant sa propre hypothèse de distribution sur la variable de diagnostic  $D$ .

### III.4.3. Détecteurs et probabilités

En connaissant désormais, la nature du signal à l'entrée de l'étage de détection et ces paramètres stochastiques, ainsi que les formules de l'expression de la fonction d'auto corrélation, on pourra examiner les détecteurs visés par l'étude. Il existe plusieurs détecteurs, leurs modélisations dépendent de la variable de décision finale, et de ses caractéristiques aléatoires.

➤ **Détecteur linéaire**

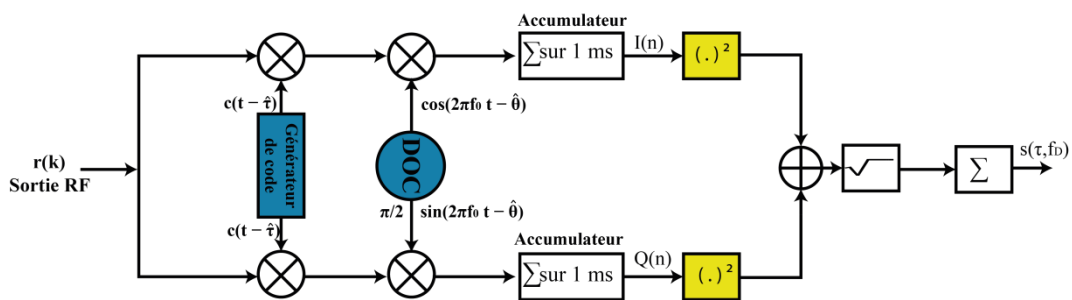


Figure III.7 : Schéma bloc d'un détecteur linéaire

On remarque que la variable de décision est la somme des deux variables des canaux  $I$  et  $Q$  sous racine carré. La variable de décision s'écrit :

$$S(f_D, \tau) = r = \sqrt{I^2 + Q^2}$$
(III.18)

Sous l'hypothèse nulle  $H_0$  les canaux  $I$  et  $Q$  sont deux variables gaussiennes centrées, ou leurs moyennes nulles coïncident avec le niveau nul du signal déterministe. La variance est celle du bruit de sortie qui est gaussien, on écrit :

$$\begin{cases} I(f_D, \tau) \sim G(0, \sigma_n^2) \\ Q(f_D, \tau) \sim G(0, \sigma_n^2) \end{cases} \quad (\text{III.19})$$

Afin de déterminer la densité de probabilité de la variable  $r$ , on établit un changement de variable :

$$\begin{cases} P(r, \theta) dr d\theta = P(I, Q) dI dQ \\ P(r, \theta) = P(I, Q) \frac{dI dQ}{dr d\theta} \\ \frac{dI dQ}{dr d\theta} = r \end{cases} \quad (\text{III.20})$$

$\frac{dI dQ}{dr d\theta} = \mathbf{r}$  : représente le jacobien.

$I$  et  $Q$  sont indépendants donc  $P(I, Q) = P(I) P(Q)$  (multiplication de deux processus gaussiens).

$$P(I, Q) = \frac{1}{2\pi\sigma_n^2} \exp\left\{-\left(\frac{I^2 + Q^2}{2\sigma_n^2}\right)\right\} \quad (\text{III.21})$$

On remplace dans (III.20):

$$P(r, \theta) = \frac{r}{2\pi\sigma_n^2} \exp\left\{-\left(\frac{r^2}{2\sigma_n^2}\right)\right\} \quad (\text{III.22})$$

Et on déduit les densités  $P(r)$  et  $P(\theta)$  :

$$\begin{cases} P(r) = \int_0^{2\pi} \frac{r}{2\pi\sigma_n^2} \exp\left\{-\left(\frac{r^2}{2\sigma_n^2}\right)\right\} d\theta \\ P(r) = \frac{r}{\sigma_n^2} \exp\left\{-\left(\frac{r^2}{2\sigma_n^2}\right)\right\} \end{cases} \quad (\text{III.23})$$

Qui correspond à une loi de Rayleigh.

$$\left\{ \begin{array}{l} P(\theta) = \int_0^r \frac{r}{2\pi\sigma_n^2} \exp\left\{-\left(\frac{r^2}{2\sigma_n^2}\right)\right\} dr \\ P(\theta) = \frac{1}{2\pi} \end{array} \right. \quad (\text{III.24})$$

Qui correspond à une loi uniforme. La probabilité de fausse alarme se déduit de la densité de probabilité :

$$\left\{ \begin{array}{l} P_{fa} = \int \frac{r}{\beta\sigma_n^2} \exp\left\{-\left(\frac{r^2}{2\sigma_n^2}\right)\right\} dr \\ P_{fa}(\beta) = \exp\left\{-\frac{\beta^2}{2\sigma_n^2}\right\} \end{array} \right. \quad (\text{III.25})$$

La  $P_{fa}$  est en fonction du seuil, donc il est évident d'obtenir un seuil avec une  $P_{fa}$  prédéfinie.

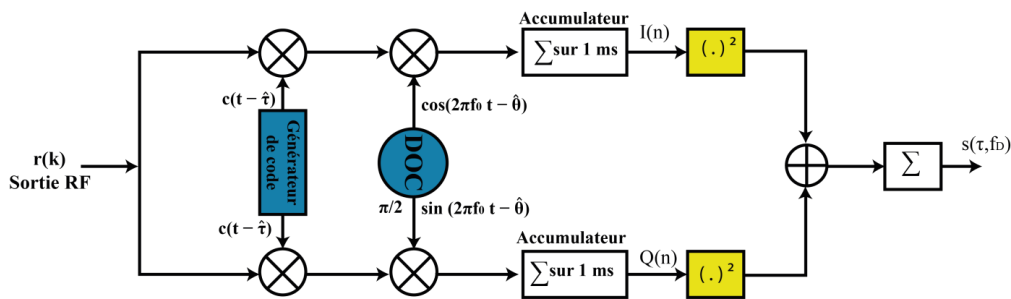
Sous l'hypothèse adjacente  $H_1$ , la valeur moyenne n'est plus nulle, mais la variance reste similaire puisque le signal déterministe est de variance nulle, de plus qu'on considère qu'il n'y a pas de corrélation entre le bruit et le signal utile. On aura :

$$\left\{ \begin{array}{l} I(f_D, \tau) \sim G\left(\frac{A}{2} \cos \varphi, \sigma_n^2\right) \\ Q(f_D, \tau) \sim G\left(\frac{A}{2} \sin \varphi, \sigma_n^2\right) \\ P(r) = \frac{r}{\sigma_n^2} \exp\left\{-\left(\frac{r^2 + \frac{A^2}{2}}{2\sigma_n^2}\right)\right\} \exp\left(\frac{Ar}{2\sigma_n^2}\right) \end{array} \right. \quad (\text{III.26})$$

Ce qui correspond à une loi de *Rice*, ou sa fonction de répartition peut être évaluée à l'aide de la fonction de Macrum [17] [14] [8]:

$$\begin{cases} P_{fa}(\beta) = \exp\left(-\frac{\beta}{2\sigma_n^2}\right) \\ P_d(\beta) = Q_1\left(\sqrt{\frac{\lambda}{\sigma_n^2}}, \frac{\beta}{\sqrt{\sigma_n^2}}\right) \end{cases} \quad (III.27)$$

➤ **Détecteur quadratique**



**Figure III.8 :** Schéma bloc d'un détecteur quadratique

Dans ce détecteur la variable de décision est la racine carré de la fonction d'auto corrélation complexe :

$$R(f_D, \tau) = I + j Q$$

$$S(f_D, \tau) = |R(f_D, \tau)|^2 = I^2 + Q^2$$

Puisque  $r_{IF}[n]$  est bruité avec un bruit gaussien,  $I(f_D, \tau)$  et  $Q(f_D, \tau)$  sont deux variables gaussiennes ou leurs moyennes dépendent de la présence ou de l'absence du signal utile et son alignement avec le signal généré localement. Si le signal n'est pas présent ou n'est pas aligné  $I(f_D, \tau)$  et  $Q(f_D, \tau)$  sont de moyennes nulles et de variances  $\sigma_n^2$ , qui dépendent du bruit d'entrée et du bruit additionnel dans la chaîne de réception. Puisqu'on additionne deux variables suivant une loi normale, de moyennes nulles et mis au carré,  $S(f_D, \tau)$  est une variable aléatoire *KHI 2* centrale [14] [17].

$$\text{Sous } H_0 \begin{cases} I(f_D, \tau) \sim (0, \sigma_n^2) \\ Q(f_D, \tau) \sim (0, \sigma_n^2) \end{cases} \quad (III.28)$$



$$S(f_D, \tau) = Z = r^2 = I^2 + Q^2 \quad (\text{III.29})$$

La densité de probabilité sera :

$$\begin{cases} dZ = 2r dr P(Z) \\ P(r) \frac{dr}{dZ} = \frac{P(r)}{2r} \\ P(Z) = \frac{1}{2\sigma_n^2} \exp\left\{-\left(\frac{z}{2\sigma_n^2}\right)\right\} \end{cases} \quad (\text{III.30})$$

Ce qui correspond à une loi *KHI 2* centrée  $\chi^2 \sim (0, \sigma_n^2)$ , Avec les paramètres :

$$\begin{cases} P_{fa} = \int_{\beta}^{\infty} \frac{r}{\sigma_n^2} \exp\left\{-\left(\frac{r^2}{2\sigma_n^2}\right)\right\} dr \\ P_{fa}(\beta) = \exp\left\{-\frac{\beta}{2\sigma_n^2}\right\} \\ \beta = -2\sigma_n^2 \ln(P_{fa}) \end{cases} \quad (\text{III.31})$$

Sous H1 :

$$P(z) = \frac{1}{2\sigma_n^2} \exp\left\{-\left(\frac{z + \frac{A^2}{4}}{2\sigma_n^2}\right)\right\} \exp\left\{\left(\frac{A\sqrt{z}}{2\sigma_n^2}\right)\right\} \quad (\text{III.32})$$

L'équation correspond à la fonction de *KHI 2* non centrée de paramètre de non centralité  $\lambda$ .

$$P(z) = \frac{1}{2\sigma_n^2} \exp\left\{-\left(\frac{z + \lambda}{2\sigma_n^2}\right)\right\} \exp\left\{\left(\frac{\sqrt{\lambda z}}{\sigma_n^2}\right)\right\} \quad (\text{III.33})$$

Les paramètres du ROC obtenus sont [14] :

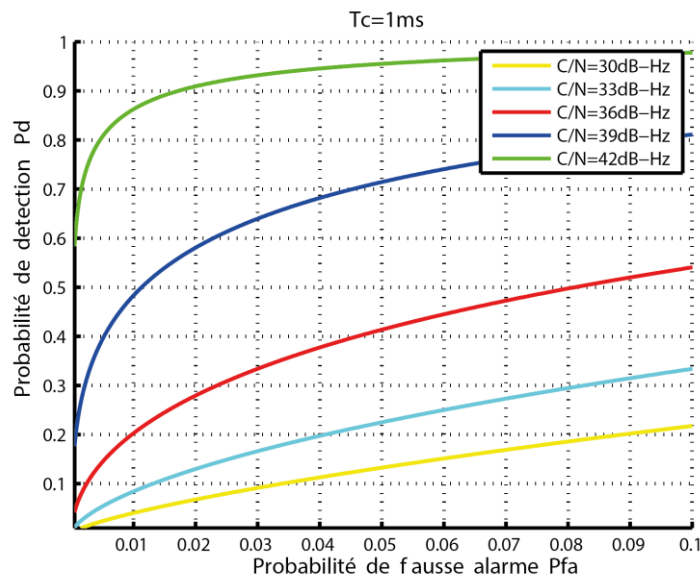
$$\begin{cases} P_{fa}(\beta) = \exp\left(-\frac{\beta}{2\sigma_n^2}\right) \\ P_d(\beta) = Q_1\left(\sqrt{\frac{\lambda}{\sigma_n^2}}, \sqrt{\frac{\beta}{\sigma_n^2}}\right) \end{cases} \quad (\text{III.34})$$

A partir de l'équation (III.13) on trouve une relation de la probabilité de détection avec le  $C/N_0$ :

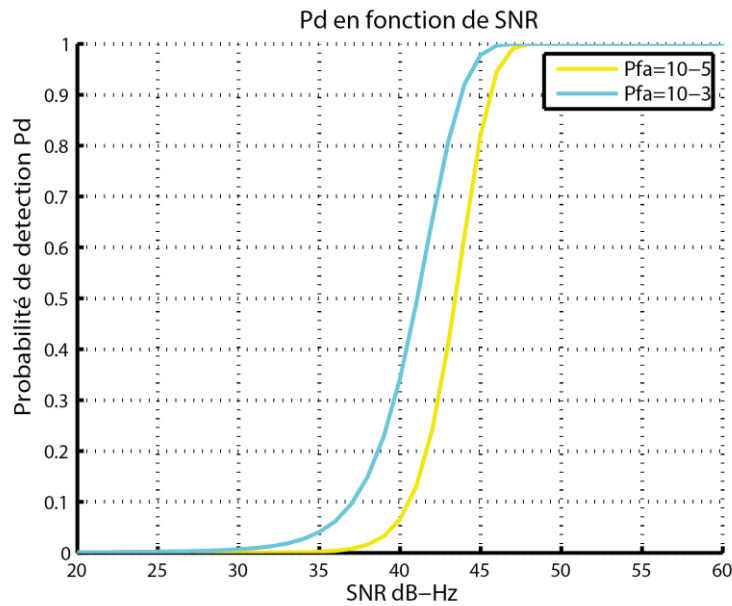
$$\begin{cases} P_d(\beta) = Q_1\left(\sqrt{2P_{reçue}/N_0}, \sqrt{\beta'}\right) \\ P_d(\beta) = Q_1\left(\sqrt{C/N_0 \times T_c}, \sqrt{\beta'}\right) \end{cases} \quad (\text{II.35})$$

**Remarque :** L'acquisition normalisée des détecteurs linéaire et quadratique sous  $H_1$  correspond à une fonction de distribution identique, qui est une fonction  $Q$  de Marcum d'ordre 1. Toutefois, avec le détecteur linéaire on obtient une meilleure performance en terme de probabilité de fausse alarme sous  $H_0$  ou  $\beta_l = \sqrt{\beta_q}$  pour une même  $P_{fa}$ .

On remarque que les probabilités de détection sous  $H_1$  des deux détecteurs linéaire et quadratique ont un résultat similaire, le tracé du ROC ci-dessous mettra en évidence les performances de ces deux techniques et pour plusieurs CNR à l'entrée de l'étage.



**Figure III.9 :** ROC pour détecteur linéaire et quadratique (probabilité de détection en fonction de  $p_{fa}$  pour plusieurs SNR).



**Figure III.10 :** *Probabilité de détection en fonction de SNR pour plusieurs Pfa.*

A partir de la **Figure III.9**, on remarque que pour un niveau acceptable de probabilité de détection pour le système, c'est-à-dire se rapprochant de 1, un SNR de 42dB-Hz pour une probabilité de fausse alarme au-dessus de  $10^{-2}$  est nécessaire, ce qui est un résultat satisfaisant garantissant une détection du signal vu le niveau minimum garanti par le système GPS 45dB-hz. Toutefois, pour une probabilité de fausse alarme en dessous de  $10^{-2}$  le système ne permet pas d'avoir un système performant. Des techniques d'augmentation doivent être appliquées au système pour améliorer les performances de détection.

Dans la **Figure III.10**, on note que pour les détecteurs linéaire et quadratique que le test sous  $H_1$  peut atteindre une probabilité de valeur approché de 1 avec une Pfa donnée relativement faible et cela avec un niveau d'SNR moins de 45 db-hz pour une Pfa de  $10^{-3}$  et plus que 45db-Hz pour une pfa de  $10^{-5}$ .

A partir des deux ensembles de courbes ci-dessus, on peut apporter la conclusion que le SNR d'entrée joue un grand rôle pour définir la sensibilité du système, cela dit, il ne peut être considéré seul. La courbe du ROC nous donne une meilleure vue de la performance.

*Note : Du point de vue de l'implémentation, le calcul de  $r$  nécessite une opération supplémentaire (la racine carrée) et les fonctions de distributions Rice et Rayleigh (telles que*

*CDF, PDF...)* ne sont pas nécessairement implémentées (dans Matlab) et devrait être évalué avec la fonction Marcum  $Q$ ) lorsque les fonctions de distributions  $\chi^2$  sont disponibles, le détecteur d'acquisition est choisi d'être le détecteur quadratique.

### III.5. Intégration

Lorsque le signal est faible (faible SNR), ne permettant pas d'obtenir les performances requises de l'acquisition (malgré le gain du dés-étalement vu précédemment sur une période), des techniques d'intégrations de plusieurs échantillons du signal sont effectuées afin de réduire le bruit et augmenter la puissance du signal (augmenter le SNR) [17].

#### III.5.1/. L'intégration cohérente

Le terme cohérent désigne une cohérence entre le temps et la phase du signal. Ainsi, sur la durée  $T_I = N \times T_C$  (multiple du temps d'un code PRN) le signal est sommé de telle sorte que les pics d'auto-corrélation s'additionnent sur  $T_I$  par période  $T_C = 1$  ms. Le bruit, qui est non corrélé avec le code PRN, reste de moyenne nulle et le gain peut atteindre les 100 % [2].

**Remarque :** en l'absence du bruit le pic croît linéairement avec le nombre d'échantillon du signal, mais en sa présence et avec une fréquence d'échantillonnage élevée, il se peut que le bruit soit corrélé avec le code PRN et ainsi dégradant les pics de corrélation.

En tenant compte d'une seule mesure (intervalle d'1 ms), notons la puissance du signal utile  $A^2$  et la variance du bruit blanc gaussien  $\sigma^2$  (puissance), le SNR de cette unique mesure est noté  $SNR_1 = \frac{A^2}{\sigma^2}$  pour  $n$  échantillons. Avec  $N$  intégrations cohérentes, en sortie on aura un signal constant, d'amplitude  $N \times A$  et de puissance  $N^2 \times A^2$ , le bruit reste de moyenne nulle et de puissance  $N \times \sigma^2$  se qui advient d'une expression du gain d'intégration cohérente dite idéale suivante [17] :

$$\begin{aligned}
 SNR_{ci} &= n \times \frac{N^2 A^2}{N \sigma^2} = N \times n \times \frac{A^2}{\sigma^2} \\
 SNR_{ci} &= N \times SNR_1 \\
 G_{ci} &= N \\
 SNR_{ci} &= SNR_1 \times G_{ci}
 \end{aligned}
 \tag{III.36}$$

D'où l'efficacité de 100%. Dans le jargon GPS ce gain est appelé « *PROCESSING GAIN* » noté  $G_{ci} = N$ . En réalité, le signal subit plusieurs pertes après les opérations établies dans les composantes électroniques avant l'intégration, ces pertes peuvent être additionnées et résumées dans un seul paramètre noté « *Implementation loss* » qui se soustrait (en dB) directement au gain d'intégration cohérente idéale ce qui donne l'expression de l'intégration cohérente suivante [17] :

$$L_{nli} = \frac{G_{ci}}{G_{nli}} \quad (III.37)$$

$$SNR_{nli} = SNR_1 \times G_{nli}$$

**Note :** Dans notre cas le SNR en sortie n'est pas bien défini car on ne peut pas séparer idéalement le bruit du signal après des opérations non linéaires de ce fait le résultat de l'intégration de sa variance dépend du signal, Néanmoins, Cette technique réduit considérablement la variance du bruit car à la sortie du corrélateur (acquisition), la puissance du bruit est inversement proportionnelle à  $T_c \left( \frac{N_0}{4T_c} \right)$ .

#### ➤ . Limites (inconvenients)

D'une part le gain élevé de l'intégration cohérente la caractérise, en contrepartie les performances de ce type d'intégration sont limitées. Les limites sont en rapport avec le temps de traitement pour les raisons suivantes [1][2][8][14] :

- ✓ *L'erreur de fréquence :* Le signal intégré subit un changement de phase au cours du temps  $T_c$  (dû aux résidus de fréquences ajoutés ou soustraits au signal). Ce changement de phase conduit à l'affaiblissement du pic en fonction de la phase et pourrait même engendrer un inversement du signe du pic d'auto corrélation (phase de 180°).

Le terme  $R_c(\Delta\tau(k))$  designant la fonction d'autocorrélation qui représente un pic en forme de triangle, où il atteint son sommet au décalage temps nul c'est-à-dire  $\tau = \hat{\tau}$ . Ainsi, le niveau du pic croît avec le nombre d'échantillons  $n$  pris en compte lors de l'intégration, si on prend comme paramètre la puissance on aura :

$$I(\mathbf{k}) = \frac{\sqrt{2p_{\text{prec}}}}{2T_s} \mathbf{d}(\mathbf{t} - \boldsymbol{\tau}) R_c(\Delta\boldsymbol{\tau}(\mathbf{k})) \cos(\pi\Delta f_D(\mathbf{k})T_s + \Delta\theta(\mathbf{k})) \text{sinc}(\pi\Delta f_D(\mathbf{k})T_s) + \mathbf{n}_I(\mathbf{k})$$

Le terme  $\text{sinc}(\pi\Delta f_D(k)Ts)$  révèle la variation du niveau d'amplitude ou de puissance prise en compte.

Exemple : Avec  $\Delta f_D = 500\text{Hz}$  et  $Tc = 1\text{ms}$   $\text{sinc}(\pi 0.5) = 0.9 = -1\text{dB}$  Donc on aura une perte de 1dB. En d'autres termes  $\Delta f_D = \frac{1}{2}Tc$  correspond à la dégradation du SNR de 0.9 dB ce qui correspond à la plus mauvaise estimation de  $f_D$  soit de  $\Delta f_D \leq \frac{1}{4}Tc$

Ainsi en sommant sur une longue durée, le résultat d'intégration peut conduire à l'annullement des pics entre eux (+1-1). On devra alors considérer une limite du temps d'intégration cohérente afin d'éviter la répercussion de ce changement de phase.

- ✓ *Transition des bits de données* : Si on considère un temps d'intégration plus long que le temps d'un bit de donnée, le signe du pic de corrélation peut changer avec un éventuel changement de signe du bit de donnée, et donc dégrader le niveau du pic après l'accumulation puisque le pic changera de signe également. Sachant que le code d'étalement et les données de navigation sont rigoureusement synchronisées, le mieux serait d'effectuer le test d'acquisition sur deux périodes d'intégration cohérente de 10ms ce qui nous garantira qu'il n'y a pas de changement de signe sur une éventuelle transition des bits de données sur l'un des deux intervalles et donc d'augmenter notre chance d'acquisition avec succès et avec un bon SNR (une des solutions).

#### ➤ **Conséquences sur l'espace de recherche**

Le nombre de cellules de la matrice ( $N_T * N_f$ ) est lié au temps d'intégration cohérente. Un temps d'intégration plus long réduit l'intervalle de fréquence puisqu'il est inversement proportionnelle à cette dernière et donc donne une estimation plus rugueuse de la fréquence recherchée, en contrepartie augmente les intervalles de fréquence donc augmente le temps de recherche.

Dans cette première approche l'intégration est dite idéale ce qui a pour conséquence que le bruit ne sera pas corrélé avec le code PRN et donc l'écart type croît avec la racine carrée du nombre d'échantillons et sa variance (puissance) avec N.

### **III.5.2. Solutions d'augmentation Alternatives :**

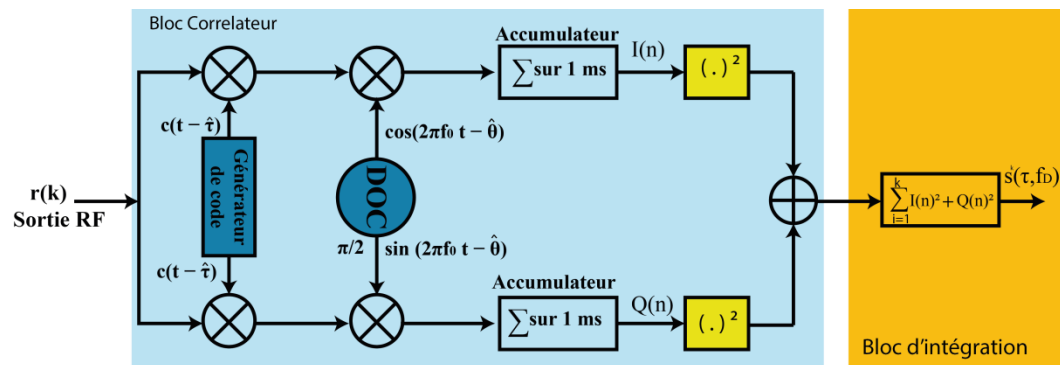
Les contraintes de l'intégration cohérente sont nombreuses, et diminuent les performances de l'acquisition, en terme de temps de traitement (beaucoup d'échantillons à intégrer) et de

calculs, en plus des limites de cette techniques [9]. D'autres solutions alternatives d'intégrations, interviennent afin d'améliorer la sensibilité des détecteurs ainsi que leur fiabilité en termes de ROC, nous citerons dans ce travail :

- ✓ Intégration non-cohérente
- ✓ Intégration cohérente différentielle

➤ **Détecteur quadratique avec Intégration non-cohérente :**

L'intégration non cohérente est l'une des techniques d'augmentation des performances de l'acquisition et qui consiste à sommer plusieurs instances de variables de décisions afin d'augmenter le niveau du signal.



**Figure III.11** : Schéma d'un récepteur avec une intégration non cohérente.

Non cohérent implique que la phase du signal n'influe pas sur la sommation (pas de dégradation due aux changements de phase) du fait de la mise des deux chaînes I et Q au carré puis les intégrant sur une durée  $T_I = K * T_C$ . Cette méthode est moins efficace (gain) que l'intégration cohérente, expliquée par le phénomène non linéaire de la mise au carré du signal en présence du bruit, en effet la moyenne du bruit étant nulle devient une valeur positive et dégrade le SNR. De plus le bruit au carré perd sa notoriété de gaussienne et peut se corrélérer avec code PRN et engendrer une dégradation du SNR, toutefois le théorème de la limite centrale couvre cette dégradation. Après l'opération au carré quel que soit le changement de phases les pics d'auto corrélation resteront positifs, on peut résumer que les canaux  $I$  et  $Q$  intégrés sont de la forme :

$$S(f_D, \tau) = \sum_{i=1}^k |R(f_D, \tau)|^2 \quad (\text{III.39})$$

Comme :

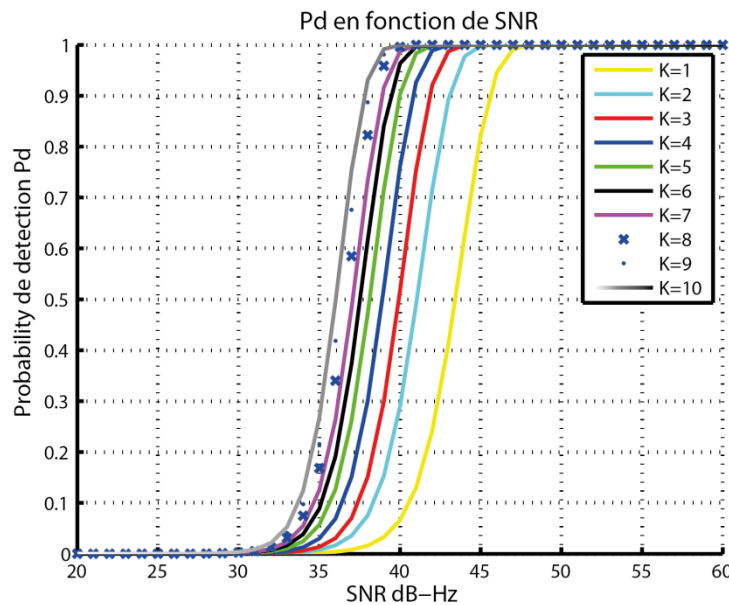
$$\begin{cases} I = cte.\cos(\omega t) \\ Q = cte.\sin(\omega t) \\ I^2 + Q^2 = cste^2(\cos^2 + \sin^2) = cte^2(\text{puissance}) \end{cases} \quad (\text{III.40})$$

Comme vu précédemment la variable de décision issue d'un détecteur quadratique est de loi de  $\chi^2$  avec 2 degrés de libertés, ainsi la somme de K variable indépendantes  $\chi^2$  avec 2 degrés de libertés  $S_K(f_D, \tau)$  est une variable aléatoire  $\chi^2$  avec 2K degrés de libertés et le paramètre de non centralité  $\lambda$  devient :

$$\lambda_K = K \times \lambda = K \times \frac{A^2}{4} \quad (\text{III.41})$$

Les paramètres du ROC s'écrivent [14]:

$$\begin{cases} P_{fa,K} = \exp\left\{-\frac{\beta}{2}\right\} \sum_{i=0}^{K-1} \frac{1}{i!} \left(\frac{\beta}{2}\right)^i \\ P_d(\beta) = Q_K\left(\sqrt{K \frac{\lambda}{\sigma_n^2}}, \sqrt{\frac{\beta}{\sigma_n^2}}\right) = Q_K\left(\sqrt{K \times C/N_0 \times T_c}, \sqrt{\beta'}\right) \end{cases} \quad (\text{III.42})$$



**Figure III.12** : Probabilité de détection en fonction de SNR pour  $P_{fa}=10^{-5}$  pour un détecteur avec intégration non cohérente.



De la **Figure III.12**, Pour  $K=1$  l'équation de la variable de décision ainsi que les courbes du ROC correspondent à un détecteur quadratique. On peut identifier la croissance de la performance avec le nombre d'intégrations  $K$ . De plus, on déduit que pour une  $P_{fa}$  donnée la probabilité de détection atteint un niveau maximum avec un SNR de moins en moins important avec un nombre d'intégration non cohérente croissant.

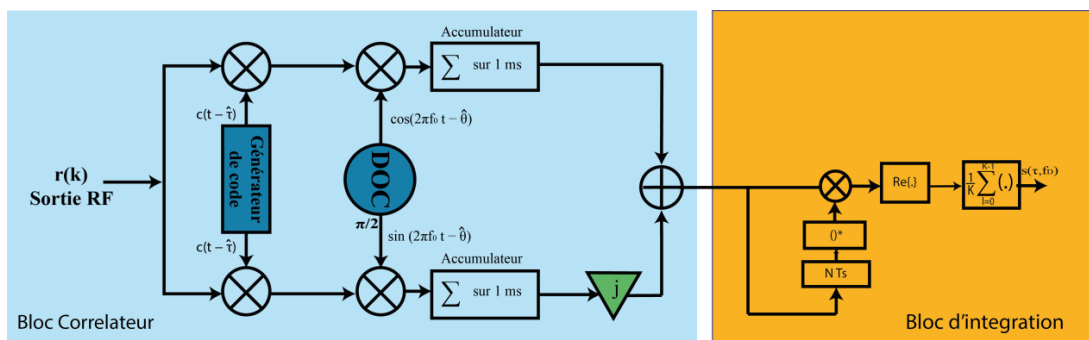
*Note : La dégradation du pic d'auto corrélation dû à l'intégration non cohérente est moindre pour la première intégration et se dégrade en fonction du nombre d'intégrations. On notera qu'avec un nombre plus important d'intégrations la fonction d'auto corrélation a tendance à se courber d'avantage même avec une déviation faible de la fréquence. Sachant qu'avec un doppler nul il n'y a pas de pertes quelque soit le nombre d'intégrations.*

### ➤ Détecteur quadratique avec Intégration cohérente différentielle

Avec l'intégration non cohérente, l'effet l'atténuation du niveau du signal utile engendré par la phase initiale est relevé. Cependant, la moyenne du bruit n'est plus nul de faite de la mise au carré du signal sur les deux chaines et engendre comme convenu précédemment, des pertes en termes de signal sur bruit. Pour faire face à cet inconvénient une solution qui est l'intégration cohérente différentielle peut être modélisée à travers la formule de la variable de décision suivante [17][14] :

$$S_K(f_D, \tau) = \sum_{k=1}^K R_e \left\{ \left[ Y_{I,k}(\tau, f_D) + j Y_{Q,k}(\tau, f_D) \right] \left[ Y_{I,k-1}(\tau, f_D) + j Y_{Q,k-1}(\tau, f_D) \right]^* \right\} \quad (\text{III.43})$$

Où  $k$  détermine l'instance prise en compte pour la corrélation utilisé pour former la variable de décision, la fonction de corrélation complexe  $\left[ Y_{I,k-1}(\tau, f_D) + j Y_{Q,k-1}(\tau, f_D) \right]$  évalué pour l'instance du signal  $(k-1)$  est utilisée pour corriger la phase de la  $k^{\text{eme}}$  instances, dans ce sens les composantes du bruit multipliés sont indépendant et la moyenne du bruit résultant reste nulle.



**Figure III.13** : *Le détecteur différentiel cohérent.*

Le résultat de l'équation est un complexe ,donc on prend la valeur réelle de la variable, l'expression de la variable de décision sera [17][14] :

$$\begin{aligned}
 S_k(f_D, \tau) &= \sum_{k=1}^K R_e \left\{ \left[ Y_{I,k}(\tau, f_D) + jY_{Q,k}(\tau, f_D) \right] \left[ Y_{I,k-1}(\tau, f_D) + jY_{Q,k-1}(\tau, f_D) \right]^* \right\} \\
 &= \sum_{k=1}^K R_e \left\{ X_{2k}(\tau, f_D) X_{2k-1}^*(\tau, f_D) \right\} \\
 &= \frac{1}{4} \sum_{k=1}^K \left[ \left| X_{2k}(\tau, f_D) + X_{2k-1}(\tau, f_D) \right|^2 - \left| X_{2k}(\tau, f_D) - X_{2k-1}(\tau, f_D) \right|^2 \right] \\
 &= \sum_{k=1}^K \left[ \left| \frac{X_{2k}(\tau, f_D) + X_{2k-1}(\tau, f_D)}{2} \right|^2 - \sum_{k=1}^K \left| \frac{X_{2k}(\tau, f_D) - X_{2k-1}(\tau, f_D)}{2} \right|^2 \right]
 \end{aligned} \tag{III.44}$$

$$\text{Les deux expressions } \sum_{k=1}^K \left| \frac{X_{2k}(\tau, f_D) + X_{2k-1}(\tau, f_D)}{2} \right|^2 \text{ et } \sum_{k=1}^K \left| \frac{X_{2k}(\tau, f_D) - X_{2k-1}(\tau, f_D)}{2} \right|^2$$

représentent deux variable aléatoires khi2 centrées sous  $H_0$  et une est centrée et l'autre non centrée sous  $H_1$ . La modélisation statistique de ce détecteur se révèle compliquée, nous avons alors considéré le cas spécial  $k=1$ , ce qui revient à chercher la distribution de la différence entre deux variables khi2 à deux degrés de liberté [14][17].

En utilisant les caractéristiques des variables khi2, après quelques développements, nous aurons sous  $H_0$  la densité suivante :

$$f_0(s) = \begin{cases} \frac{1}{2\sigma_n^2} \exp\left(-\frac{s}{\sigma_n^2}\right) & s < 0 \\ \frac{1}{2\sigma_n^2} \exp\left(-\frac{s}{\sigma_n^2}\right) & s \geq 0 \end{cases} \tag{III.45}$$

Ce qui donnera la probabilité de fausses alarmes suivante [14] :

$$P_{fa}(\beta) = \begin{cases} \int_{\beta}^{+\infty} \frac{1}{2\sigma_n^2} \exp\left(-\frac{s}{\sigma_n^2}\right) ds = 1 - \frac{1}{2} \exp\left(-\frac{\beta}{\sigma_n^2}\right) & \beta < 0 \\ \int_{\beta}^{+\infty} \frac{1}{2\sigma_n^2} \exp\left(-\frac{s}{\sigma_n^2}\right) ds = \frac{1}{2} \exp\left(-\frac{\beta}{\sigma_n^2}\right) & \beta \geq 0 \end{cases} \tag{III.46}$$

Sous l'hypothèse  $H_1$ , la densité est [14] :

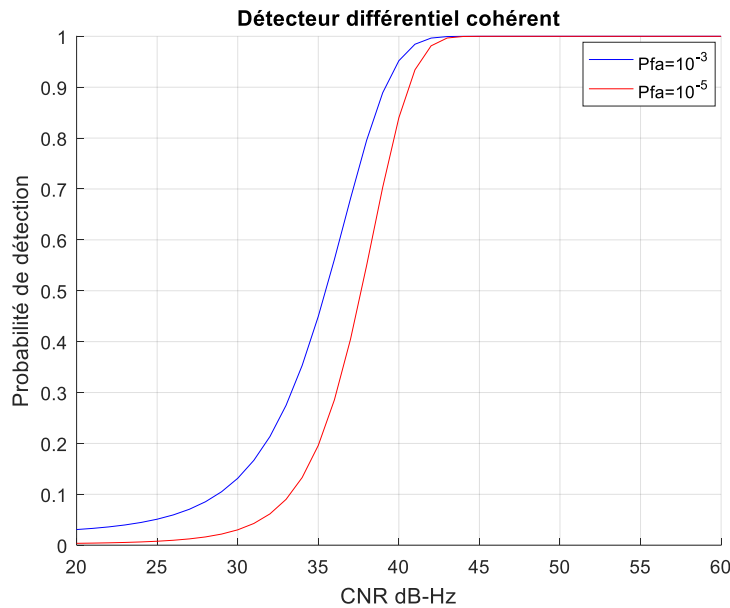
$$f_1(s) = \begin{cases} \frac{1}{2\sigma_n^2} \exp\left(\frac{2s-\lambda}{2\sigma_n^2}\right) & s < 0 \\ \frac{1}{2\sigma_n^2} \exp\left(\frac{2s-\lambda}{2\sigma_n^2}\right) Q_1\left(\sqrt{\frac{\lambda}{\sigma_n^2}}, \sqrt{\frac{4s}{\sigma_n^2}}\right) & s \geq 0 \end{cases} \quad (\text{III.47})$$

La probabilité de détection est [14] :

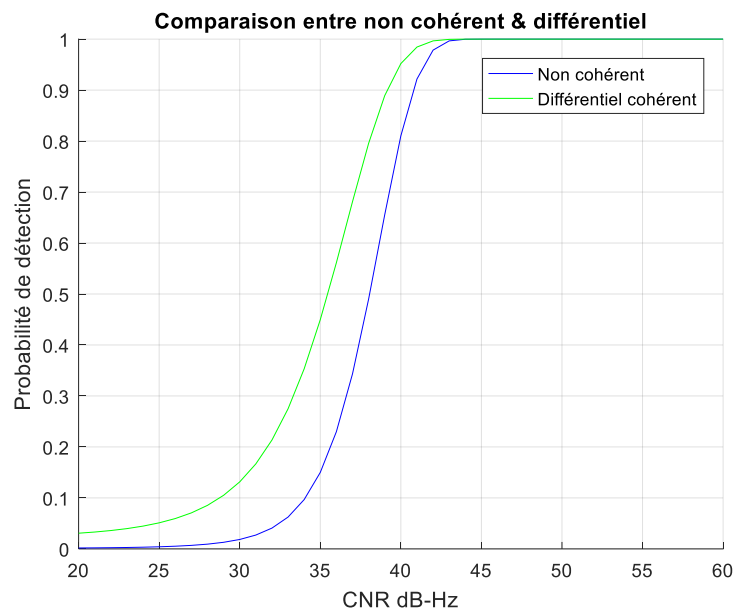
$$P_d(s) = \begin{cases} 1 - \frac{1}{2} \exp\left(\frac{2\beta-\lambda}{2\sigma_n^2}\right) & \beta < 0 \\ Q_1\left(\sqrt{\frac{\lambda}{\sigma_n^2}}, \sqrt{\frac{\beta}{\sigma_n^2}}\right) - \frac{1}{2} \exp\left(\frac{2\beta-\lambda}{4\sigma_n^2}\right) Q_1\left(\sqrt{\frac{\lambda}{2\sigma_n^2}}, \sqrt{\frac{2\beta}{\sigma_n^2}}\right) & \beta \geq 0 \end{cases} \quad (\text{III.48})$$

Dans la **Figure III.14** nous présentons la performance du détecteur différentiel en termes de probabilité de détection en fonction du CNR, pour des probabilités de fausses alarmes de  $10^{-5}$  et  $10^{-3}$ . En diminuant la probabilité de fausses alarmes la probabilité de détection diminue.

Nous présentons enfin une comparaison générale entre les détecteurs non cohérent et différentiel dans la **Figure III.15**. Le détecteur différentiel présente la meilleure performance, pour avoir une probabilité de détection de 0.9 le CNR requis est de 42 dB-Hz pour le détecteur non cohérent et seulement de 38 dB-Hz pour le détecteur différentiel.



**Figure III.14** : Courbe ROC pour le détecteur différentiel cohérent.



**Figure III.15:** Comparaison entre le détecteur non cohérent et le détecteur différentiel.

### III.5. CONCLUSION

Les résultats de performances obtenus pour les détecteurs linéaire est quadratique répondent favorablement à la réception des signaux GPS qui garantissent un niveau minimal du signal de 158.5 dBw. Ainsi, la méthode d'acquisition série utilisée est efficace et simple à simuler par rapport à d'autres méthodes qui utilisent d'autres fonctions plus complexes. Le seul inconvénient pondérant de cette dernière est le temps de traitement, plus exactement le temps de recherche qui augmente en cherchant une meilleure précision des paramètres de l'acquisition.

Lorsque le signal est faible (faible SNR), ne permettant pas d'obtenir les performances requises de l'acquisition des techniques d'intégrations de plusieurs échantillons du signal sont effectuées afin de réduire le bruit et augmenté la puissance du signal. Les solutions d'intégrations proposées sont bien connues dans la littérature. L'analyse des performances de ces systèmes reste précieuse pour pouvoir les comparer. L'intégration cohérente avec un temps d'intégration de plus d'une milliseconde semble inapproprié dans un processus en

temps réel vu le temps trop long du procédé ainsi qu'aux atténuations engendrées par la transition des bits de données.

La méthode d'intégration non cohérente améliore nettement les performances en fonction du nombre d'intégrations ou la dépendance de la phase est éliminée par la quadrature des sorties des corrélateurs. Cela dit, cette technique accumule le bruit avec le nombre d'intégrations et dégrade le SNR.

L'intégration différentielle cohérente relève la contre performance du bruit du détecteur non cohérent en utilisant la corrélation complexe sur deux portions du signal, pour compenser l'erreur de phase. Le gain de ce détecteur peut atteindre les 100%, Cependant, ce détecteur est sensible à une erreur de phase du signal qui peut rendre la détection inefficace.

Bien que le ROC caractérise complètement les performances du détecteur, il est souvent utile d'avoir une seule métrique. Le SNR encapsule autant d'informations que possible sur les performances du détecteur. Ce paramètre caractérise la qualité de la variable aléatoire de la cellule et, dans un certain sens résume les informations portées par le ROC.



# *Chapitre IV*

## *Implémentation du FPGA*

- ❖ **Introduction**
- ❖ **Flot de conception FPGA**
- ❖ **Implémentation FPGA**
- ❖ **Comparaison**
- ❖ **Conclusion**

## **IV. 1. INTRODUCTION**

L'implémentation d'un logiciel ou matériel englobe tous les processus impliqués dans le bon fonctionnement d'un élément dans son environnement, y compris l'analyse des besoins, l'installation, la configuration, la personnalisation, l'exécution, le test, l'intégration des systèmes, la formation des utilisateurs et l'application d'éventuelles modifications. Le terme « déploiement » est parfois employé dans le même sens [9].

Deux défis majeurs pour appliquer les technologies et qui sont l'aspect d'efficacité et de performance, et l'aspect temporel. Il est essentiel que notre récepteur réponde aux performances étudiées en théorie, du moins arriver à un comportement approché du système. Il y a deux catégories de systèmes en temps réel : des systèmes dits critiques et des systèmes dits non critiques. Un système non critique a peu de contraintes temporelles ou ces dernières sont relativement larges par exemple un système de réservations. Dans notre cas, il est essentiel que notre récepteur réponde dans les plus brefs délais à cause du mouvement des satellites et du mobile. Le récepteur en mouvement doit connaître sa position de manière instantanée, sinon l'information s'avère peu utile et même désuète, ce qui peut entraîner des erreurs graves (dans le cas, par exemple, de l'atterrissage d'un avion). Il s'agit donc d'un système critique.[1] [9]

Afin d'implanter notre système, une méthodologie rigoureuse doit être suivie pour une conception efficace, fiable et réalisable en temps pratique.

## **IV. 2. OUTILS D'IMPLEMENTATION**

Afin de faciliter les tâches d'implémentation on a adopté le co-design software-hardware. Il s'agit d'utiliser les logiciels Matlab, Simulink, Xilinx ISE et Xilinx System Generator avec la carte FPGA comme illustré par la figure IV.1, combinaison qui nous permet



d'atteindre l'objectif désiré avec une vérification et une visualisation des résultats simples et efficaces.



Figure IV.1: Flot de conception.

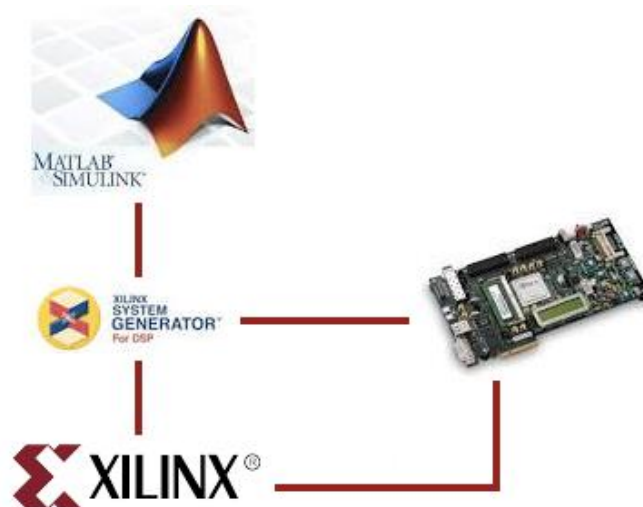


Figure IV.2: Co-design.

#### IV. 2. 1. Matlab & Simulink

MATLAB est un logiciel de calcul matriciel à syntaxe simple. Avec ses fonctions spécialisées et sa bibliothèque très riche, Matlab peut être aussi considéré comme un langage de programmation adapté pour les problèmes scientifiques. Les instructions sont interprétées et exécutées ligne par ligne selon deux modes de fonctionnement:

- Mode interactif: où les instructions sont exécutées au fur et à mesure qu'elles sont données par l'utilisateur ;
- Mode exécutif: MATLAB exécute ligne par ligne un 'fichier .m'.

Simulink est l'extension graphique de Matlab permettant de représenter les fonctions mathématiques et les systèmes sous forme de diagramme en blocs et de simuler le fonctionnement de ces systèmes []. Une option importante récemment développée est la capacité de générer les programmes HDL des systèmes simulés pour une éventuelle réalisation FPGA.

#### IV. 2. 2. Xilinx ISE

C'est l'outil développé par Xilinx pour leurs cartes FPGA, il permet de réaliser les différentes fonctions []:

- **La synthèse:** Permet d'estimer grossièrement le besoin en ressources et en vitesse pour une carte FPGA désirée;
- **Implémentation:** Elle concerne le 'mapping' ainsi que le routage dans une carte FPGA. Elle donne une estimation plus précise que la synthèse en plus l'analyse temporelle et l'analyse de puissance;
- **La programmation:** Le résultat de cette étape est un fichier à transférer vers la carte FPGA.

#### IV. 2. 3. Xilinx system generator

C'est un nouvel outil très intéressant qui permet la connexion entre Simulink et les blocs Xilinx sous Simulink, ce qui facilite considérablement les tâches d'implémentation et de visualisation des signaux. Un fichier ainsi conçu peut aussi générer un bit stream directement, on parle de la transparence entre les deux logiciels.[11]

### **IV. 3. DESCRIPTION DE LA CARTE Xilinx Virtex 6**

Xilinx, Inc. est une entreprise américaine de semi-conducteurs. Inventeur du FPGA avec un premier produit en 1985, Xilinx fait partie des plus grandes entreprises spécialisées dans le développement et la commercialisation des composants logiques programmable [11]. Xilinx fabrique une large gamme de FPGA, Spartan pour une fabrication en grande série et Virtex pour les hautes performances.

En 2009, Xilinx a lancé les FPGA Virtex 6, en technologie 40nm selon trois catégories différentes LX, LXT et SXT. Elle est optimisée spécialement pour les applications nécessitant un traitement numérique du signal important selon treize modèles fournissant une haute densité d'intégration se situant entre 74500 et 566800 portes logiques. On s'intéresse à la version XC6VLX240T où chaque CLB est constitué de 2 slices, ces derniers sont de nombre 37680. Chaque slice contient 4 générateurs de fonctions 'LUT' à 6 entrées, 4 bascules mémoires, 4 bascules additionnelles, une logique de traitement de la retenue, des portes logiques arithmétiques, des larges multiplexeurs pouvant implémenter des fonctions à 7 ou 8 entrées. De plus, cet FPGA possède 416 blocs de RAM dont chacun est de 36 Kbits ce qui donne une capacité totale de stockage interne de 14976 Kbits. Les blocs d'entrées et de sorties sont principalement composés de buffers, leur nombre maximum pour utilisateur est de 720.[11]

Le FPGA Virtex 6 XC6VLX240T dispose de 768 blocs DSP48 permettant d'effectuer les calculs suivants :

- Multiplication 25x18bits ;
- Multiplication avec accumulation -MAC- ;
- Additionneur à 3 bloc d'entrées 48 bits ;
- Registre à décalage 48bits ;
- Module de calcul logique (et, ou, ou exclusif) sur 48bits ;
- Comparateur sur 48bits ;
- Détecteur de 'pattern'.

Dans ce travail de mémoire, nous avons utilisé une plateforme de prototypage avec un FPGA Vitex6, il s'agit de la plateforme ML605 présentée par la figure IV.3.

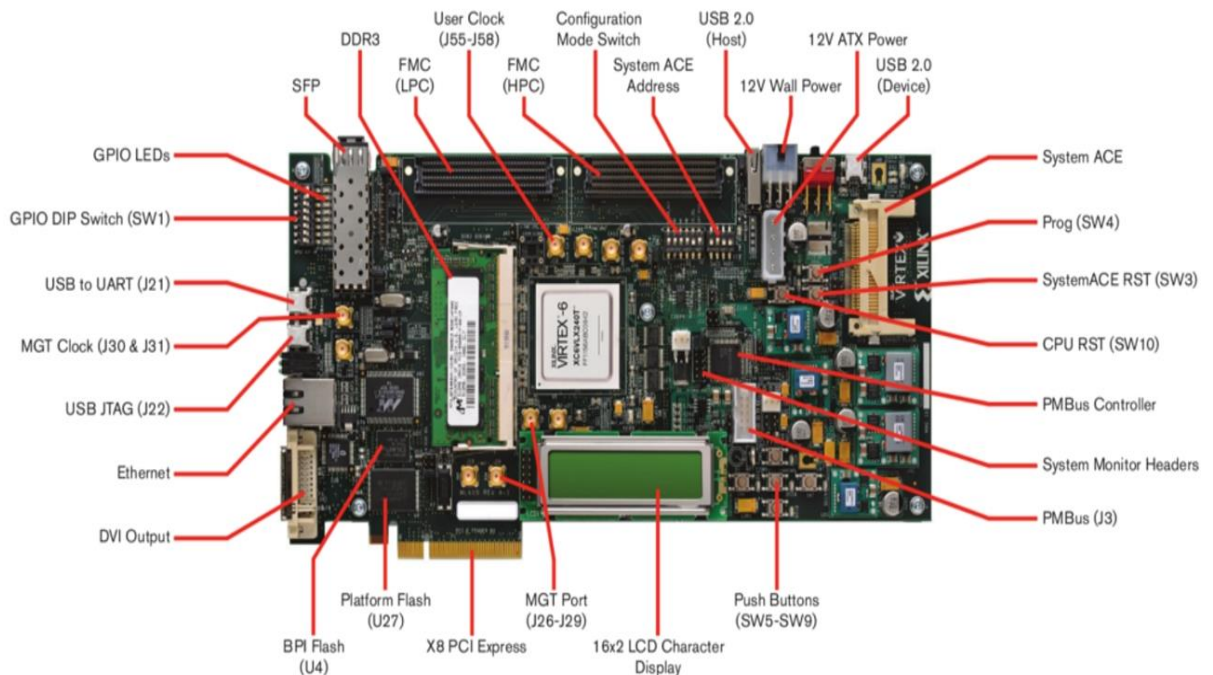
La plateforme ML605 comprend tous les composants de base, matériel, outils de conception et des IP's, avec une connectivité série haute performance et une interface mémoire très avancée.

Un port USB JTAG est disponible, il permet de programmer le FPGA en plus utiliser l'outil ChipScope™ de Xilinx. Une deuxième interface USB permet d'établir une communication UART *-UniversalAsynchronousReceiverTransmitter-* entre le FPGA et un ordinateur à une fréquence de communication pouvant atteindre 921,600 kHz.

La plateforme ML605 possède deux ports d'extension FMC *-FPGA Mezzanine Card-*. Ce type de connecteur est un standard développé afin de faciliter la comptabilité entre des modules d'extension et la carte contenant l'électronique programmable.

Le FPGA étant entièrement numérique, les signaux analogiques doivent être échantillonnés et discrétisés avant d'y être traités. L'utilisation de convertisseurs analogique-numériques *-ADC pour analog-to-digital converter-* est donc requise.

Finalement, la carte de développement ML605 dispose des interfaces de communication USB, Gigabit Ethernet et PCI Express.



**Figure IV.3 :** FPGA Virtex-VI ML605.

## **IV. 4. IMPLEMENTATION SUR FPGA**

### **IV. 4. 1. Implémentation de l'acquisition avec détecteur quadratique**

Notre implémentation repose sur un bloc de base, il s'agit de l'acquisition série non cohérente sur un seul canal, aménagée pour l'implémentation en introduisant des ALU's. Cette entité se compose:

#### ➤ **Des entrées:**

- **Signal:** Correspond au signal *GNSS* reçu quantifié sur quatre bits;
- **Code local:** C'est le code *PRN* généré localement quantifié sur deux bits;
- **Cos & Sin:** Deux signaux sinusoïdaux en quadrature générés localement quantifiés sur deux bits;

#### ➤ **Des sous- modules:**

- **Décodeur:** Le décodeur accomplit la multiplication du signal reçu par le code *GNSS* local, à sa sortie le nombre de bits sera par conséquent égal à six bits ;
- **Démodulateur:** Sert à supprimer la porteuse *FI* du signal reçu; en utilise deux signaux sinusoïdaux générés localement quantifiés sur deux bits chacun. Le signal à la sortie du démodulateur est réparti sur deux voies en phase et en quadrature, chaque voie est quantifiée sur huit bits ;
- **Intégrateur:** Les deux sorties issues du démodulateur se trouvent cumulées sur la longueur du code pour chacune, sachant la fonction du décodeur qui précède, il s'agit de l'implémentation d'un corrélateur. A ce niveau le nombre de bits est de vingt-cinq afin d'éviter la saturation de ce module;
- **Module:** Dans notre schéma d'acquisition, on a opté pour une intégration non cohérente: il s'agit d'évaluer le module de chaque voie issue de l'intégration, mais avant de réaliser cette opération on a normalisé sur la longueur 'N' du code. La valeur de normalisation est quantifiée sur 26 bits ce qui donne après division cinquante-un bits et après évaluation du module cent deux bits ; Ce module permettra aussi de combiner les deux voies en phase et en quadrature de par une opération d'addition, il se trouve quantifié sur cent deux bits ;

#### ➤ **Des sorties**

- **Signal de sortie:** Signal résultant de l'acquisition;

- **Seuil:** Le seuil fixe calculé;
- **Décision:**  $H_0$  ou  $H_1$  pour absence ou présence du code recherché.

On note que les codes d'étalement utilisés ne sont pas implémentés mais générés sous Matlab puis utilisés comme des entrées au niveau de la carte. Réellement, ces codes peuvent être stockés en mémoire RAM accessible pendant l'opération d'acquisition. Sous Simulink / System generator, notre implémentation des détecteurs quadratique et quadratique avec intégration non cohérente est réalisée selon les figures sous la forme modulaire décrite ci-dessous.

L'exécution est réalisée pour un signal GPS défini selon les paramètres de simulation : Pour un Doppler nul, un décalage de 10 chips et un rapport *CNR* de 40dB, le résultat obtenu à partir de l'outil 'Scope' du Simulink. Nous observons que notre étage d'acquisition détecte correctement le satellite visible avec le retard correspondant.

La compilation a été validée à l'aide de l'outil ISE –version 14.1- de la société Xilinx. Ce logiciel fournit, à la fin de compilation, divers rapports et schémas, parmi eux, le résumé des ressources utilisées qui est donné sous forme de tableau précisant les quantités de ressources matérielles utilisées pour l'implémentation physique du système. Ce résumé est donné en annexe IV-1, nous en avons retenu un extrait de valeurs dans les tableaux ci-dessous. L'analyse de ces valeurs montre que les ressources disponibles permettent une implémentation sur notre carte.

**Tableau IV.1 :** Ressources utilisées pour détecteur quadratique.

Ressources	Disponibles	Utilisées	Pourcentage
Les slices registres	44,800	112	1%
Les slices LUTs	44,800	247	1%
Number of BUFG	32	1	3%
Unused Flip Flop	292	85	61%
DSP48E1s	128	22	17%
Entrées / Sorties	640	122	19%

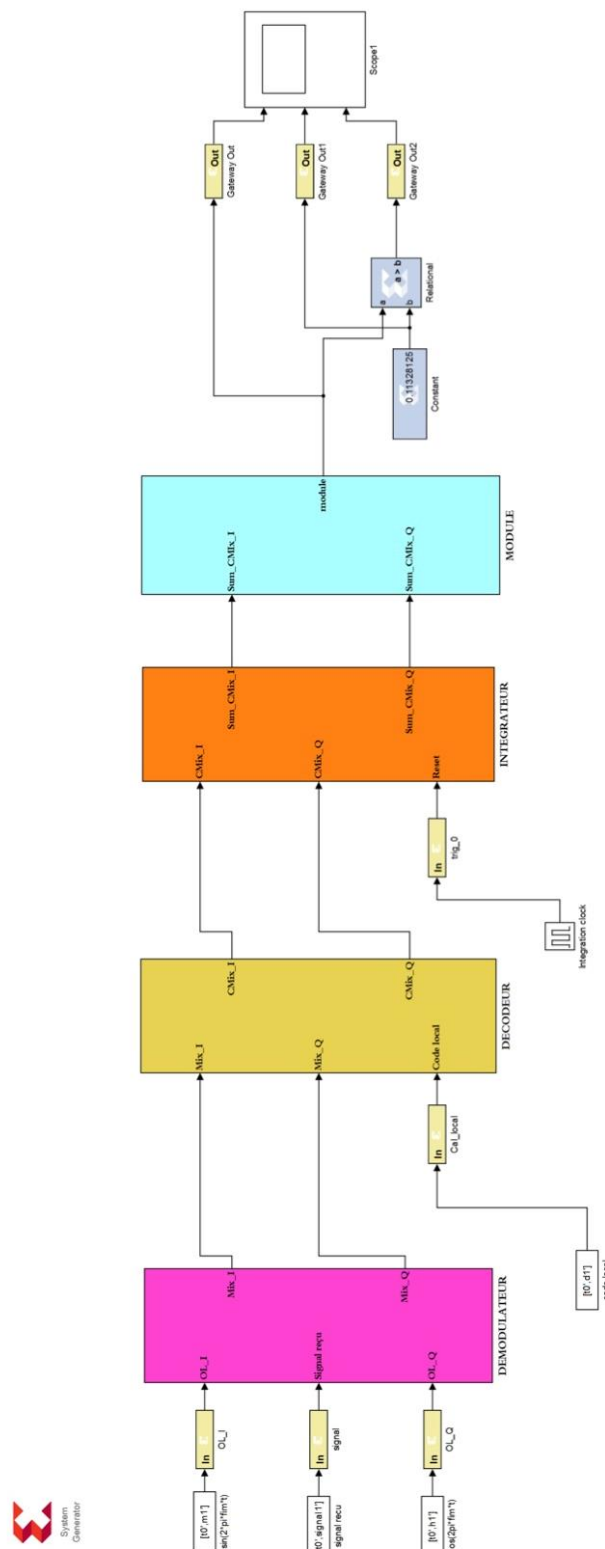


Figure IV.4 : Schéma modulaire du détecteur quadratique, Simulink/System generator.

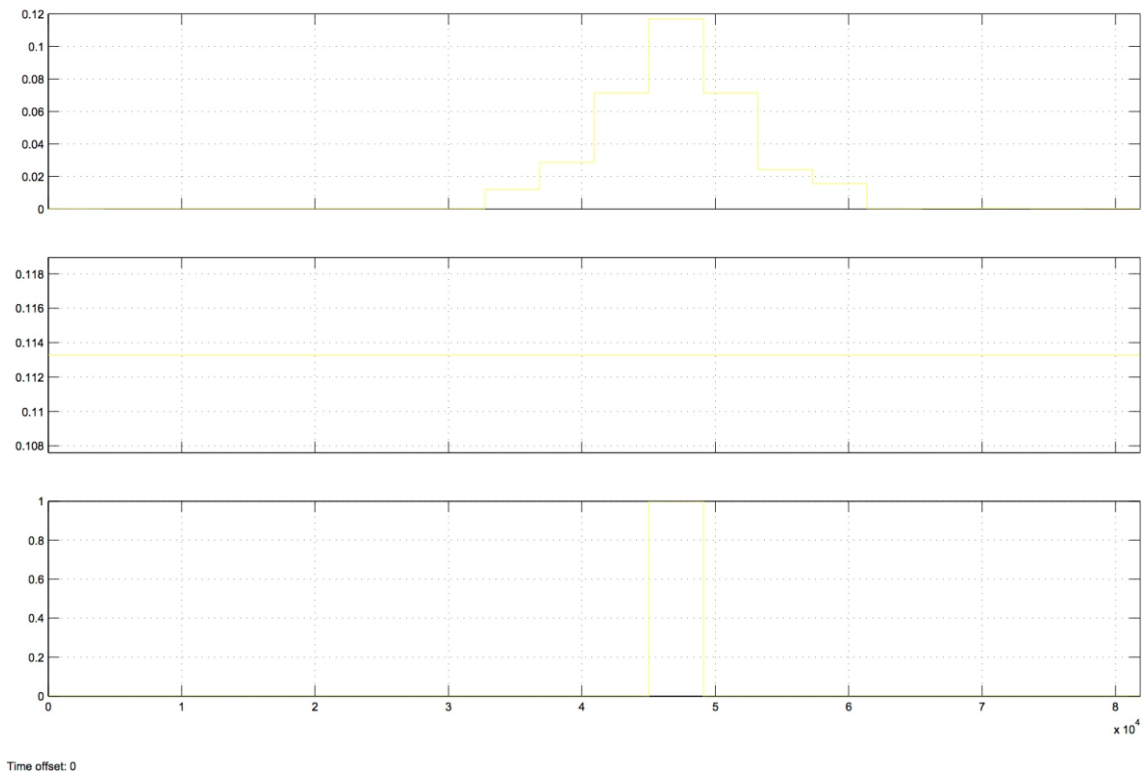


Figure IV.5 : Résultat de la simulation du détecteur quadratique, Simulink/System generator.

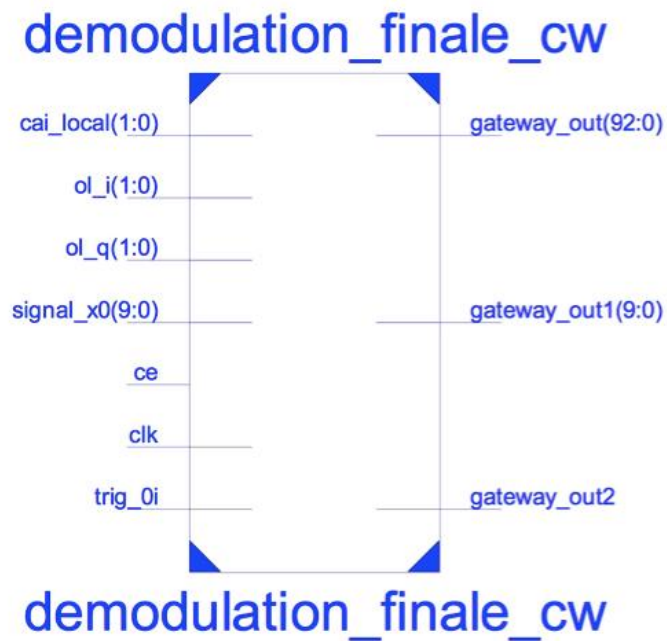


Figure IV.6 : Bloc “RTL Schematic” principal, Xilinx ISE.



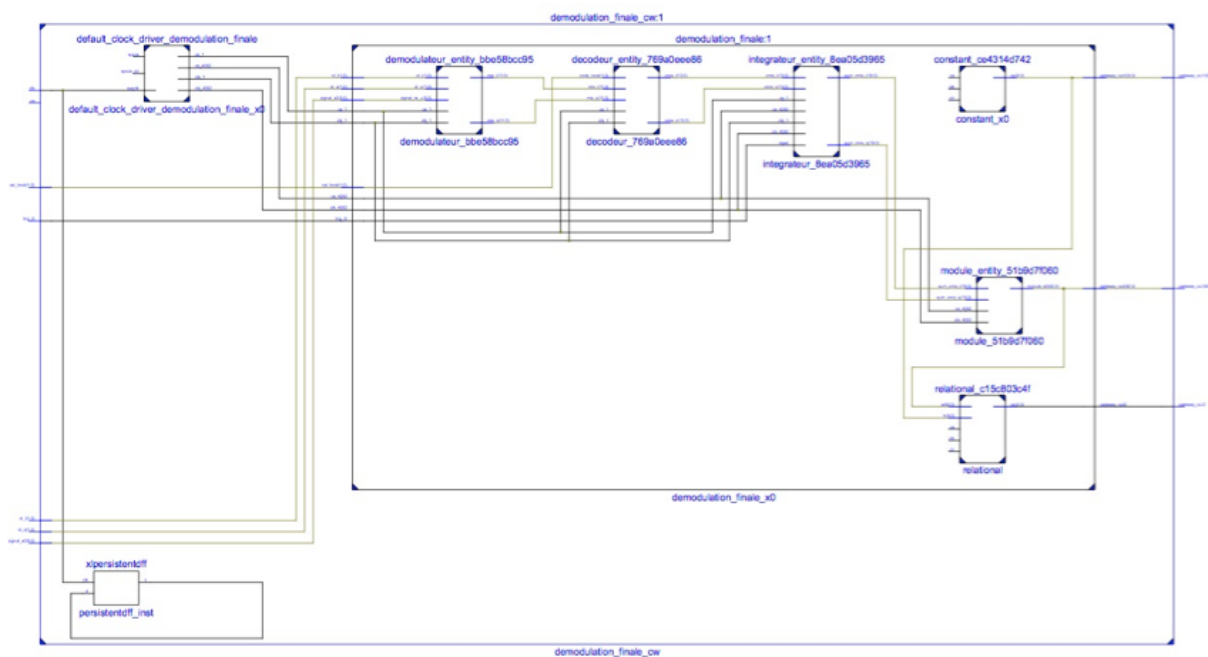
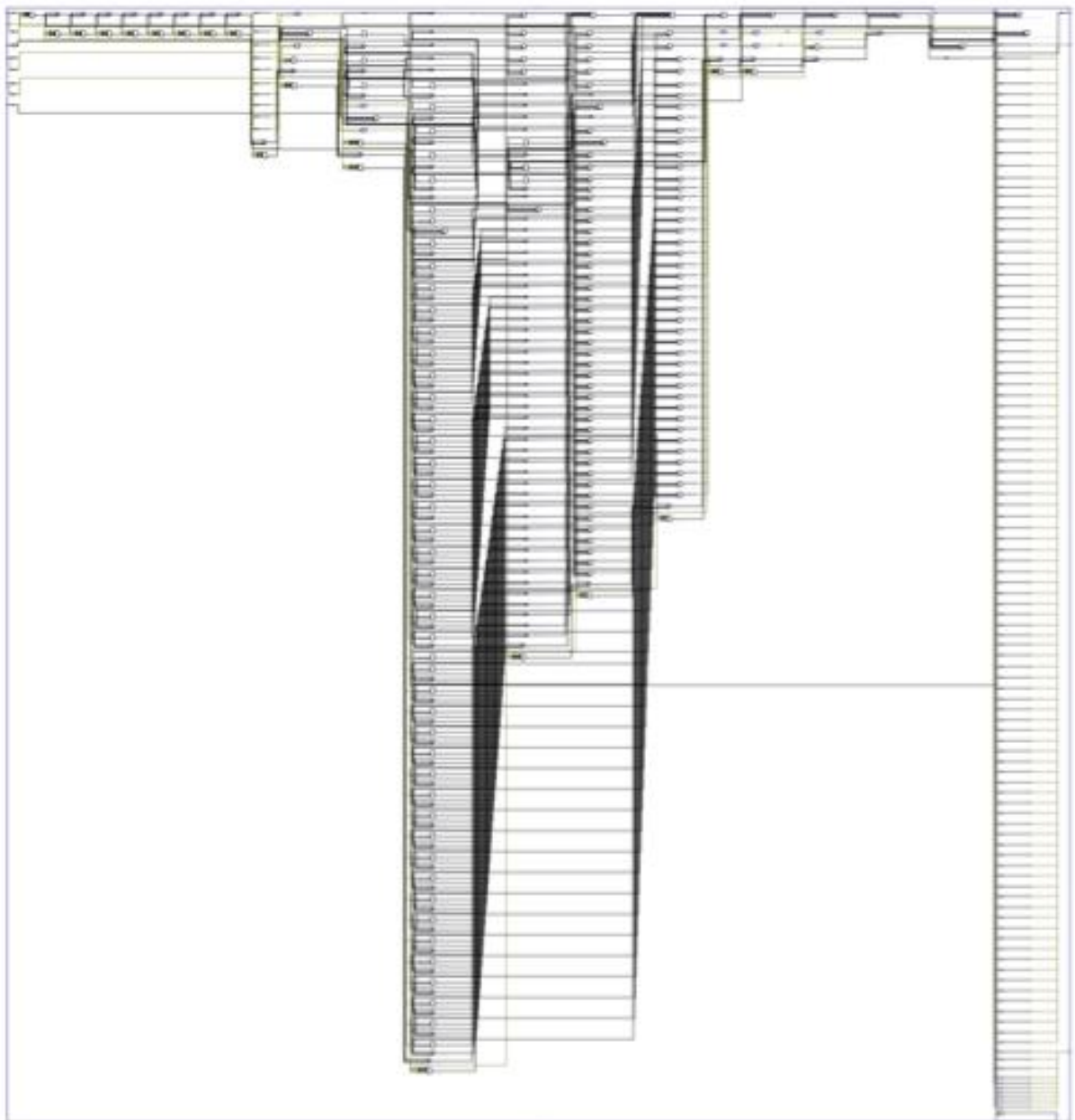


Figure IV.7 : Description “RTL Schematic” Xilinx ISE.



**Figure IV.8 :** « Technologie Schematics » détecteur quadratique, Xilinx ISE.

#### **IV. 4. 2. Implémentation de l'acquisition avec intégration**

Notre implémentation s'agit de l'acquisition série non cohérente sur un seul canal, aménagée pour l'implémentation en introduisant des ALU's. La structure repose sur des blocs modulaires, suivant un classement hiérarchique. Cette entité se compose des modules du détecteur quadratique en plus du module d'intégration. Ce module a pour tâche d'évaluer la somme de 'k' sorties d'acquisition. :

➤ **Entrées:**

- **Signal:** Correspond au signal *GNSS* reçu quantifié sur quatre bits;
- **Code local:** C'est le code *PRN* généré localement quantifié sur deux bits;
- **Cos & Sin:** Deux signaux sinusoïdaux en quadrature générés localement quantifiés sur deux bits;

➤ **Sorties**

- **Signal de sortie:** Signal résultant de l'acquisition;
- **Seuil:** Le seuil fixe calculé;
- **Décision:**  $H_0$  ou  $H_1$  pour absence ou présence du code recherché.

Les codes d'étalement utilisés ne sont pas implémentés mais générés sous Matlab puis utilisés comme des entrées au niveau de la carte. Sous Simulink / System generator, notre implémentation est réalisée selon les figures sous la forme modulaire décrite ci-dessous.

L'exécution est réalisée pour un signal GPS défini selon les paramètres de simulation : Pour un Doppler nul, un décalage de 10 chips et un rapport *CNR* de 40dB, le résultat obtenu à partir de l'outil 'Scope' du Simulink. Nous observons que notre étage d'acquisition détecte correctement le satellite visible avec le retard correspondant.

La compilation a été validée à l'aide de l'outil ISE –version 14.1- de la société Xilinx. Le résumé ressources matérielles utilisées pour l'implémentation physique du système est donné en annexe IV-2, nous en avons retenu un extrait de valeurs dans les tableaux ci-dessous.

**Tableau IV.2** : Ressources utilisées pour détecteur avec intégration.

<b>Ressources</b>	<b>Disponibles</b>	<b>Utilisées</b>	<b>Pourcentage</b>
<b>Les slices registres</b>	44,800	334	1%
<b>Les slices LUTs</b>	44,800	527	1%
<b>Number of BUFG</b>	32	1	3%
<b>Unused Flip Flop</b>	600	266	61%
<b>DSP48E1s</b>	128	22	17%
<b>Entrées / Sorties</b>	640	123	19%

La fonction RTL viewer -Register Transfert Level- présente une description hiérarchique par niveaux d'abstraction du système sous forme de schémas. Nous présentons dans l'annexe, la partie corrélation et la partie détection, à partir de ces deux figures nous observons bien la correspondance entre notre description du synoptique et les schémas réalisés.

La fonction viewer -Technologie- présente une description est une représentation schématique du fichier NGC, elle figure en termes d'éléments logiques optimisés pour une architecture ou une technologie bien définie. Par exemple, schématisé avec des LUTs, buffers d'I/O ...etc. Elle est générée après l'étape d'optimisation et la spécification de la technologie utilisée par le processus de synthèse. Ce fichier permet de visualiser la conception figure le schéma tel qu'il sera implémenté sur le composant FPGA.

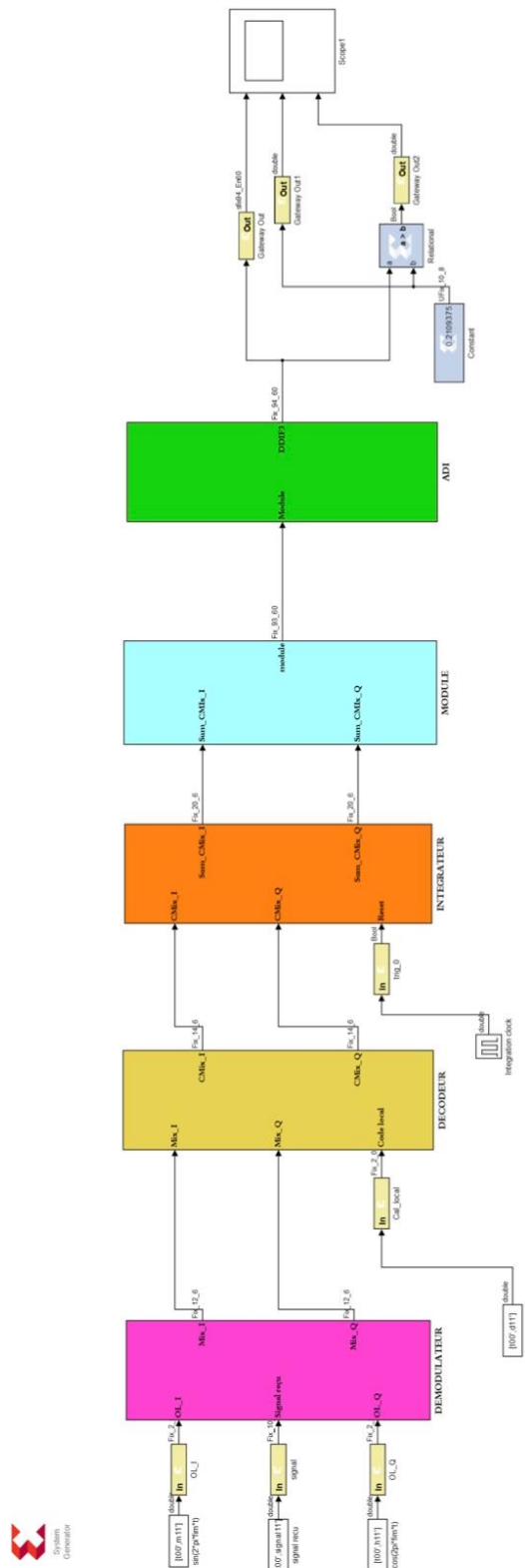
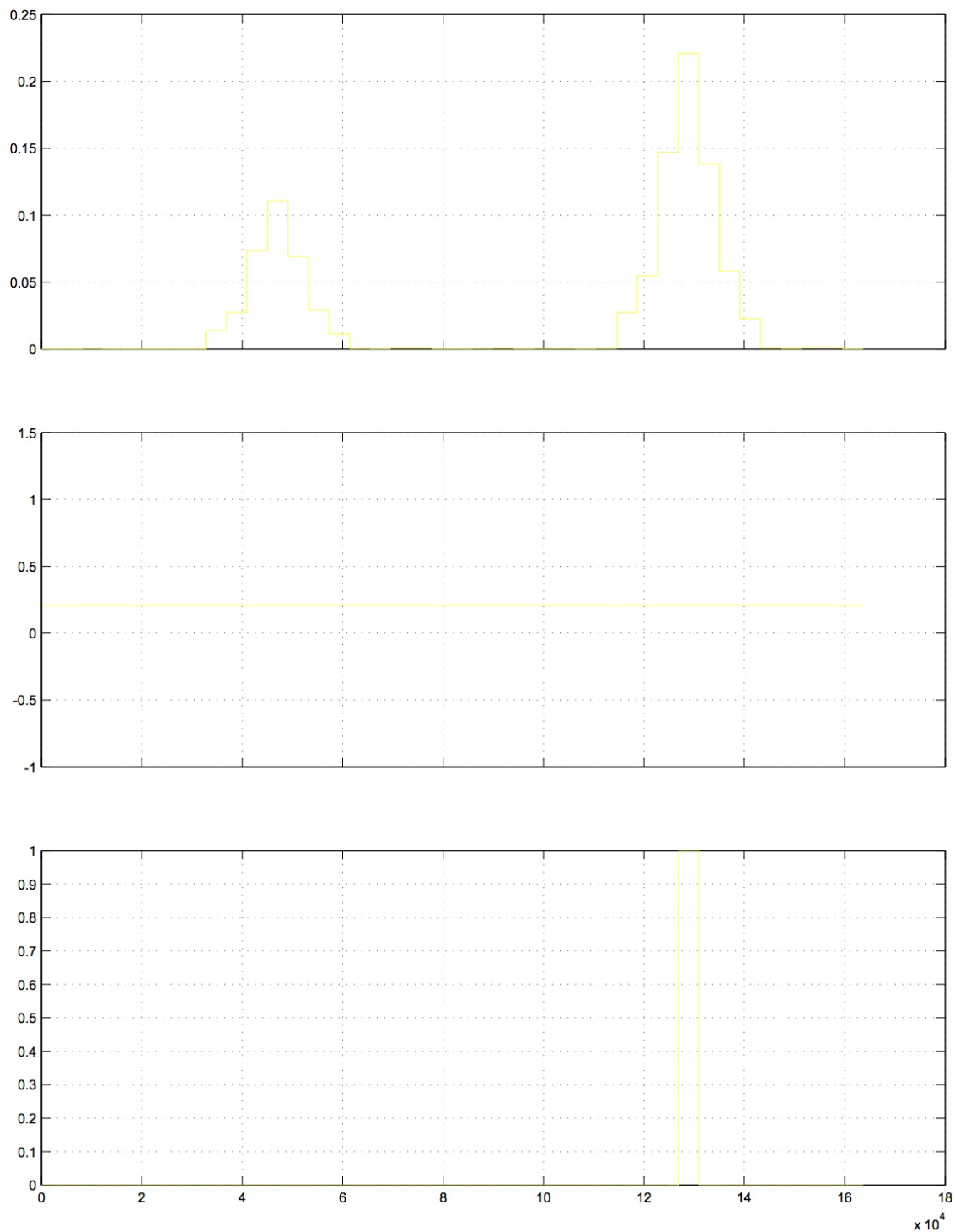


Figure IV.9 : Schéma modulaire du détecteur série non cohérent sous Simulink/System generator.



Time offset: 0

**Figure IV.10 :** Résultat de la simulation du détecteur série non cohérent Simulink/System generator.

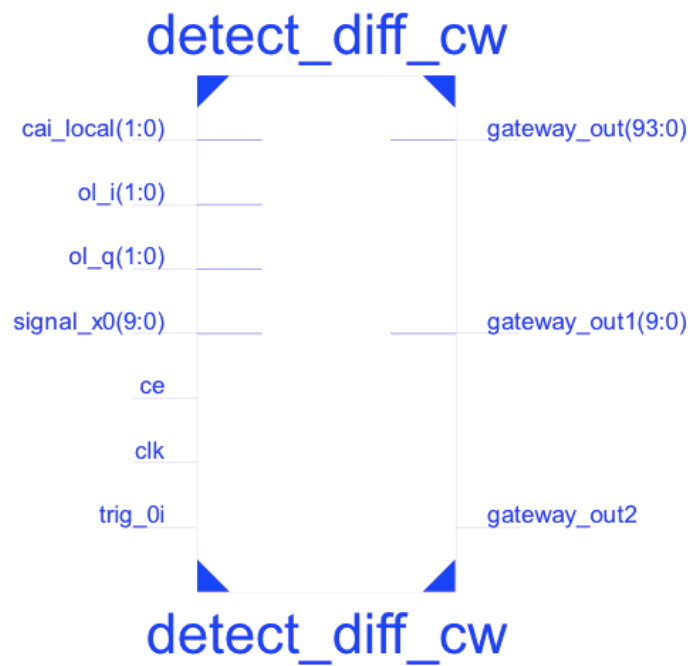


Figure IV.11 : Bloc “RTL Schematics” principal, Xilinx ISE.

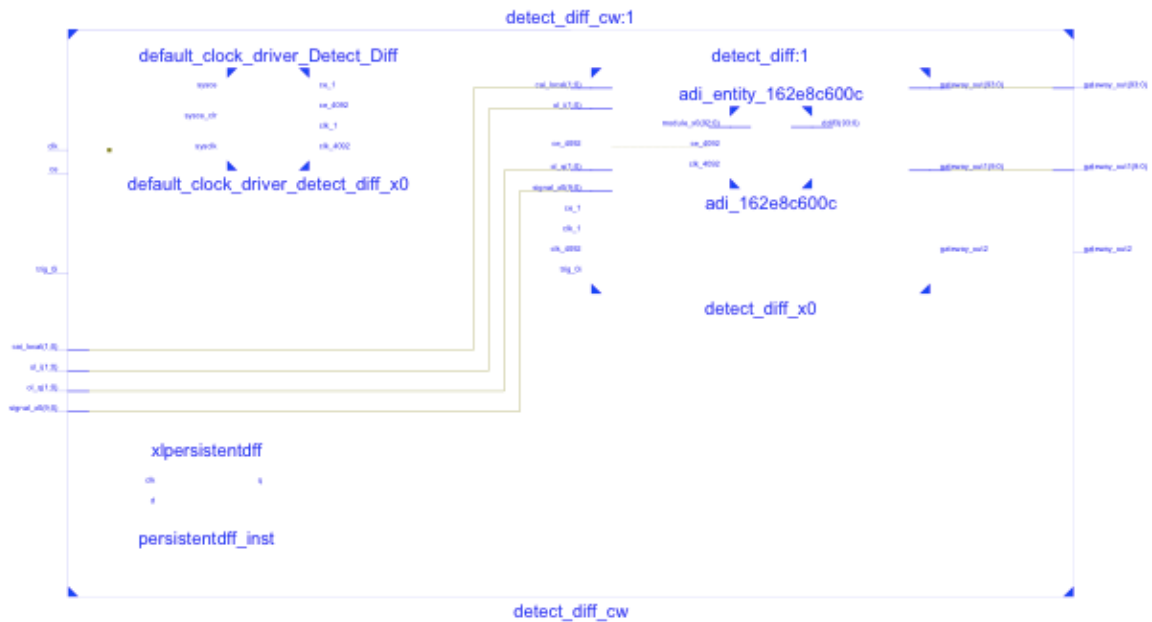


Figure IV.12 : Description « RTL Schematics » du détecteur avec intégration, Xilinx ISE.

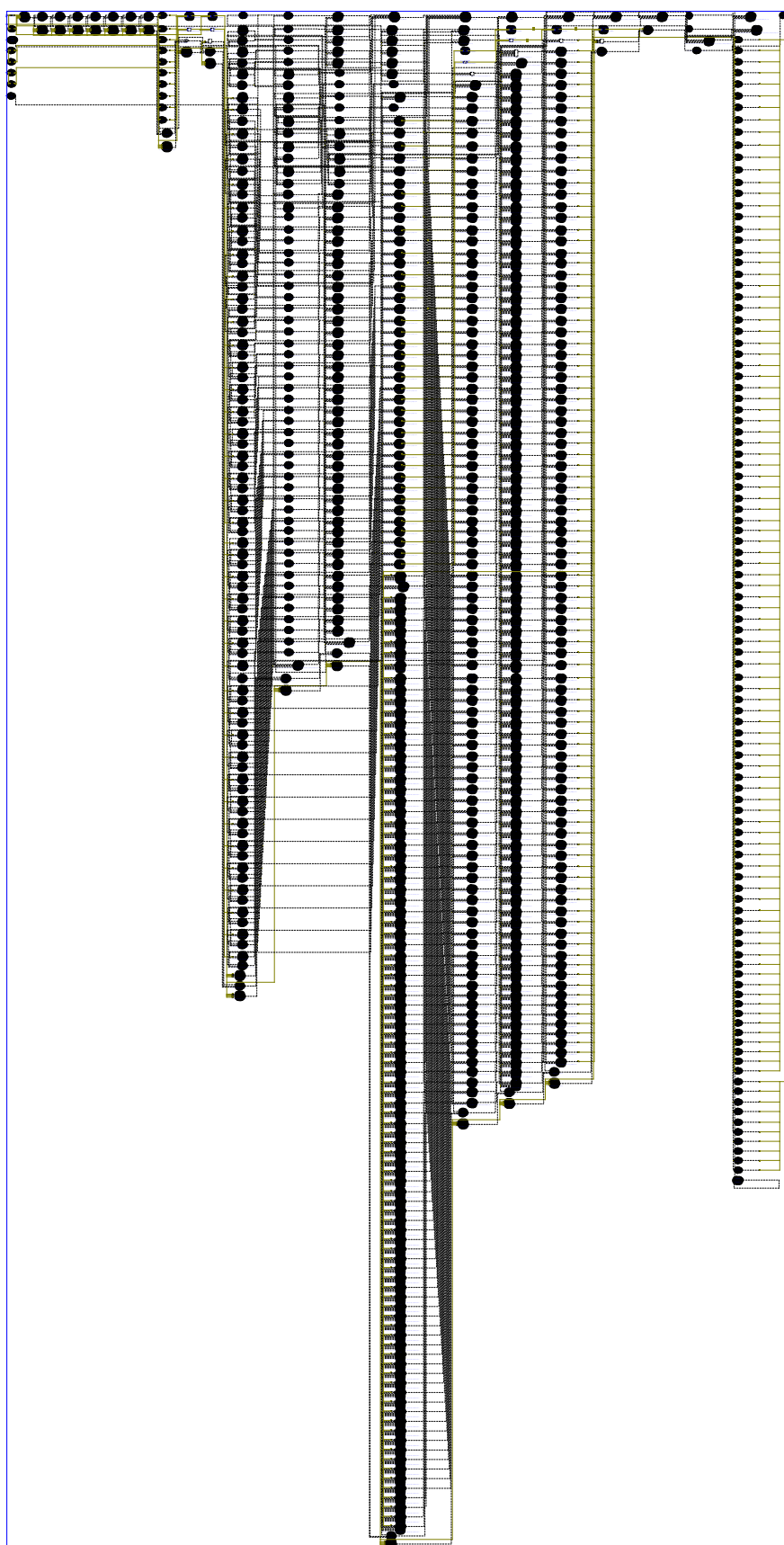


Figure IV.13 : « Technology schematics » du detecteur avec intégration, Xilinx ISE.



## **IV. 5. Comparaison**

Au niveau de la description structurelle, le détecteur avec intégration requiert un sous module d'intégration en plus par rapport au détecteur quadratique. Ce module a pour rôle l'accumulation de plusieurs instances d'une milliseconde issues de la corrélation du signal. En comparant les tableaux des ressources, l'ajout d'un module implique évidemment l'utilisation de composants supplémentaires dans la carte. Toute fois, du point de vue de l'occupation sur la carte, on peut dire que les deux implémentations peuvent être largement supportées par la carte.

La simulation des deux détecteurs a conduit à l'obtention d'un pic de corrélation au décalage incorporé au signal simulé, et prouve que le système est efficace. Les figures de l'outil scope de Simulink illustrés dans les figures IV.5 et IV.10 ont montré que le pic de corrélation du détecteur avec intégration atteint un niveau plus important que celui du détecteur quadratique, avec un gain presque idéal pour  $K=2$  (deux instances) : 0.12 pour le détecteur quadratique et 0.22 pour le détecteur quadratique avec intégration.

Pour le cas du détecteur avec intégration, on remarque un deux pics qui apparaissent, lors de la première et de la deuxième instance d'intégration. Cependant, la réponse du détecteur de la première instance est négative car le pic est en dessous du seuil fixé. Cela implique que le temps de détection du détecteur quadratique est moindre comparé au détecteur intégrateur.

La décision a pris en compte seulement les pic dépassants les seuils fixés. Cela implique d'une part, que la valeur du seuil fixé a un grand rôle dans une décision correcte, évitant une détection fausse ou fausse alarme. D'autre part, on conclut que les seuils calculés en fixant une probabilité de fausse alarme, répondent favorablement aux tests de détection fournis dans l'approche théorique, pour les deux détecteurs.

## **IV. 6. CONCLUSION**

Le nombre de ressources utilisé par le détecteur avec intégration est plus important que le premier, les chiffres sont évidents dans la partie « Device Utilisation Summary ». Dans la même partie des tableaux les éléments de la cartes qui ont été utilisés, exprimés en pourcentage, indiquent que les ressources utilisées sont beaucoup moindre que la capacité de la carte, des lors nous pourrons ajoutés plusieurs fonctions à savoir l'implémentation de tous les codes satellitaires, et l'inclusion des détecteurs des autres systèmes (GLONASS, GALILEO, BEIDOU...). Cette perspective entre dans un projet futur nécessitant plus de temps et de ressources technologiques plus performantes de celles auxquelles nous avons eu accès.

Les figures de simulations obtenues sous Simulink, montrent que les résultats théoriques sont confirmés par le biais de la simulation, ainsi le calcul des seuils et la comparaison des performances des détecteurs et également confirmé. Cependant, un seuil fixe peut atténuer les probabilités de détection et augmenter celle de fausses alarmes.

Le temps d'exécution du système n'a pas de répercutions sur un récepteur statique et est conforme à une utilisation en temps réel. Cela dit pour un récepteur dynamique, le temps de détection et de propagation de la carte peut engendrer des erreurs. Dans ce sens une investigation du système et du routage sur la carte FPGA se doit d'être établis pour se qui advient du futur.

Les outils disponibles utilisés pour l'implémentation offrent plusieurs fonctions de conception, de simulation et de test. Le choix de la combinaison de ces dernieres est une méthode très prometteuse pour l'accomplissent de système complexes et performants puisque cela nous permet d'avoir plus de fonction a notre disposition (tests, modules de conception, plusieurs descriptions possibles du système.....).

Les résultats de ce chapitre ont montré que la méthodologie adoptée pour notre conception a donné des résultats convaincants de l'efficacité du système, cohérent avec l'étude théorique. Cela est motivant pour la réalisation d'un récepteur programmable complet.



## **CONCLUSION GENERALE**

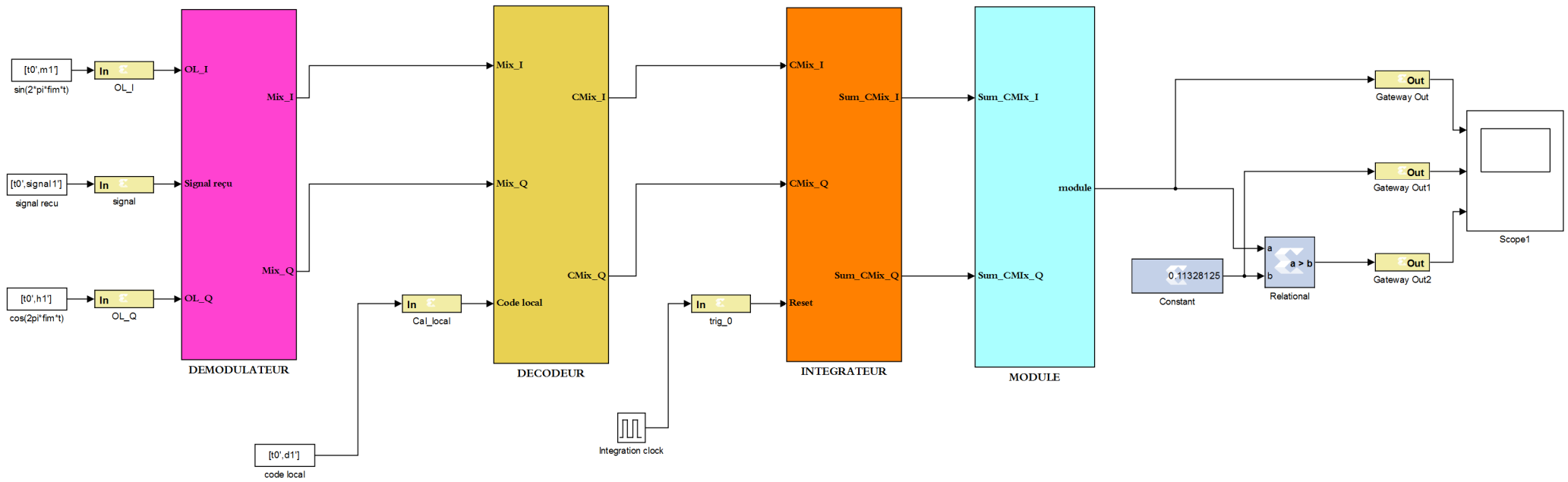
L'exigence de la couverture mondiale d'un système de positionnement, ainsi que les besoins futurs de la radionavigation ne peuvent être satisfaits par une seule constellation. L'utilisation de plusieurs systèmes implique le besoin d'interopérabilité des récepteurs GNSS, une telle conception peut être extrêmement complexe vu la multitude d'opérations complexes du fonctionnement interne d'un récepteur. Avec un nombre élevé d'opérations, pour parvenir à des performances requises d'un système fiable et la rentabilité en temps de conception avec les limites du matériel et des outils dont nous disposons, l'implémentation du système sur carte électronique demande un travail laborieux et une méthodologie rigoureuse ainsi que la maîtrise de plusieurs logiciels.

Par ce projet, conjuguer une approche logicielle et matérielle et superposer les deux méthodologies a été acquise. Cela a fallu un travail de recherche approfondie sur le système GNSS, dans le domaine du traitement du signal, des mathématiques statistiques et probabilitaires et en électronique, pour pouvoir cibler l'opération d'acquisition visée par l'étude.

L'étude théorique des récepteurs dans le chapitre un et de l'acquisition série dans le chapitre trois a été essentiel pour la conception du modèle. La connaissance de la partie analogique du traitement du signal nous a donné un aperçu des contraintes dans la pratique et a été pris en compte lors de la simulation ainsi qu'aux limites des fréquences d'échantillonnage. Dans le chapitre trois l'interprétation mathématique du signal et de ses performances grâce aux courbes ROC et leur dépendance au SNR est un résultat très utile sachant qu'il existe des outils de mesure du SNR en pratique.

Tel qu'indiqué dans le dernier chapitre, les recherches pourront se diriger vers le développement d'un récepteur GNSS hybride vu que les FPGA actuels proposent de hautes performances. L'investigation d'un système d'acquisition au seuil adaptatif pour les récepteurs GNSS est également envisageable.

## **Annexe IV.1 : Simulation du détecteur quadratique**



demodulation_finale_cw Project Status (10/02/2013 - 02:05:57)			
<b>Project File:</b>	demodulation_finale_cw.xise	<b>Parser Errors:</b>	No Errors
<b>Module Name:</b>	demodulation_finale_cw	<b>Implementation State:</b>	Placed and Routed
<b>Target Device:</b>	xc5vfx70t-1ff1136	<b>• Errors:</b>	No Errors
<b>Product Version:</b>	ISE 14.1	<b>• Warnings:</b>	<a href="#">243 Warnings (243 new)</a>
<b>Design Goal:</b>	Balanced	<b>• Routing Results:</b>	<a href="#">All Signals Completely Routed</a>
<b>Design Strategy:</b>	<a href="#">Xilinx Default (unlocked)</a>	<b>• Timing Constraints:</b>	<a href="#">All Constraints Met</a>
<b>Environment:</b>	<a href="#">System Settings</a>	<b>• Final Timing Score:</b>	0 ( <a href="#">Timing Report</a> )

Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	112	44,800	1%	
Number used as Flip Flops	110			
Number used as Latch-thrus	2			
Number of Slice LUTs	247	44,800	1%	
Number used as logic	242	44,800	1%	
Number using O6 output only	174			
Number using O5 output only	44			
Number using O5 and O6	24			
Number used as exclusive route-thru	5			
Number of route-thrus	56			
Number using O6 output only	49			
Number using O5 output only	7			
Number of occupied Slices	85	11,200	1%	
Number of LUT Flip Flop pairs used	292			
Number with an unused Flip Flop	180	292	61%	
Number with an unused LUT	45	292	15%	
Number of fully used LUT-FF pairs	67	292	22%	
Number of unique control sets	10			
Number of slice register sites lost to control set restrictions	2	44,800	1%	
Number of bonded IOBs	122	640	19%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number used as BUFGs	1			
Number of DSP48Es	22	128	17%	
Average Fanout of Non-Clock Nets	1.67			

Performance Summary			
<b>Final Timing Score:</b>	0 (Setup: 0, Hold: 0, Component Switching Limit: 0)	<b>Pinout Data:</b>	<a href="#">Pinout Report</a>
<b>Routing Results:</b>	<a href="#">All Signals Completely Routed</a>	<b>Clock Data:</b>	<a href="#">Clock Report</a>
<b>Timing Constraints:</b>	<a href="#">All Constraints Met</a>		

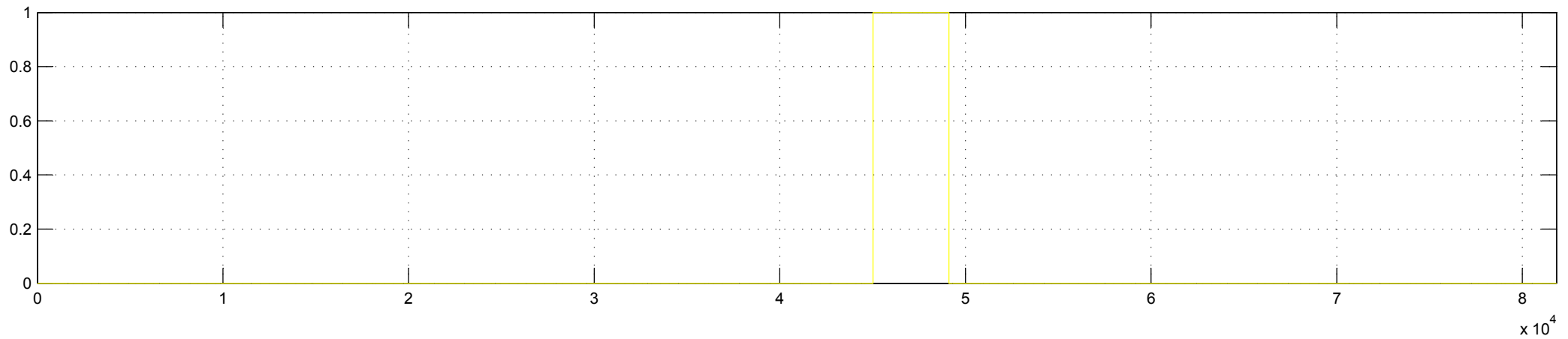
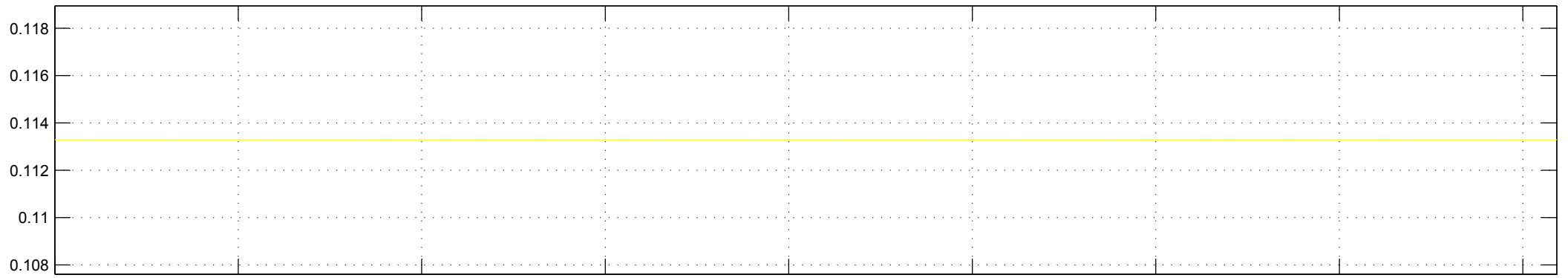
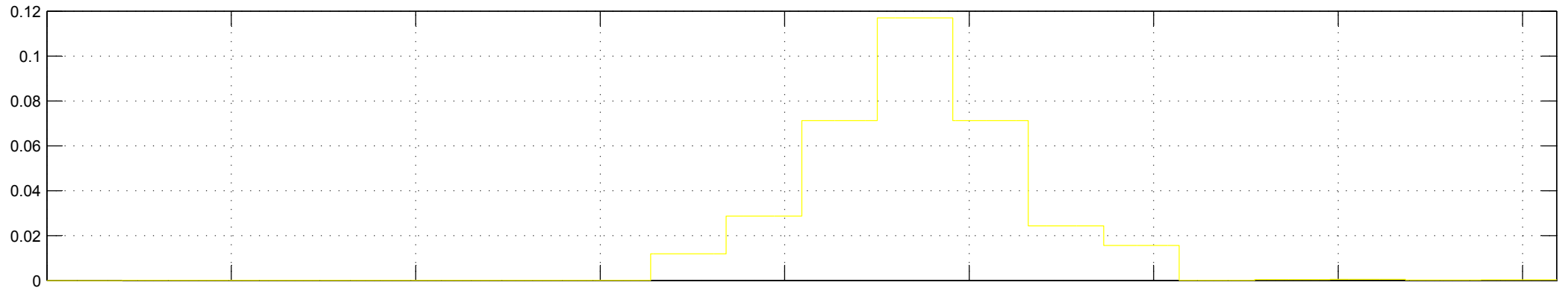
Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
<a href="#">Synthesis Report</a>	Current	mer. 2. oct. 01:57:33 2013	0	<a href="#">238 Warnings (238 new)</a>	<a href="#">1 Info (1 new)</a>
<a href="#">Translation Report</a>	Current	mer. 2. oct. 02:03:47 2013	0	<a href="#">2 Warnings (2 new)</a>	0
<a href="#">Map Report</a>	Current	mer. 2. oct. 02:04:37 2013	0	<a href="#">1 Warning (1 new)</a>	<a href="#">19 Infos (19 new)</a>

<a href="#">Place and Route Report</a>	Current	mer. 2. oct. 02:05:38 2013	0	<a href="#">1 Warning (1 new)</a>	<a href="#">1 Info (1 new)</a>
Power Report					
<a href="#">Post-PAR Static Timing Report</a>	Current	mer. 2. oct. 02:05:55 2013	0	<a href="#">1 Warning (1 new)</a>	<a href="#">3 Infos (3 new)</a>
Bitgen Report					

Secondary Reports			[-]
Report Name	Status	Generated	
<a href="#">Physical Synthesis Report</a>	Current	mer. 2. oct. 02:04:37 2013	

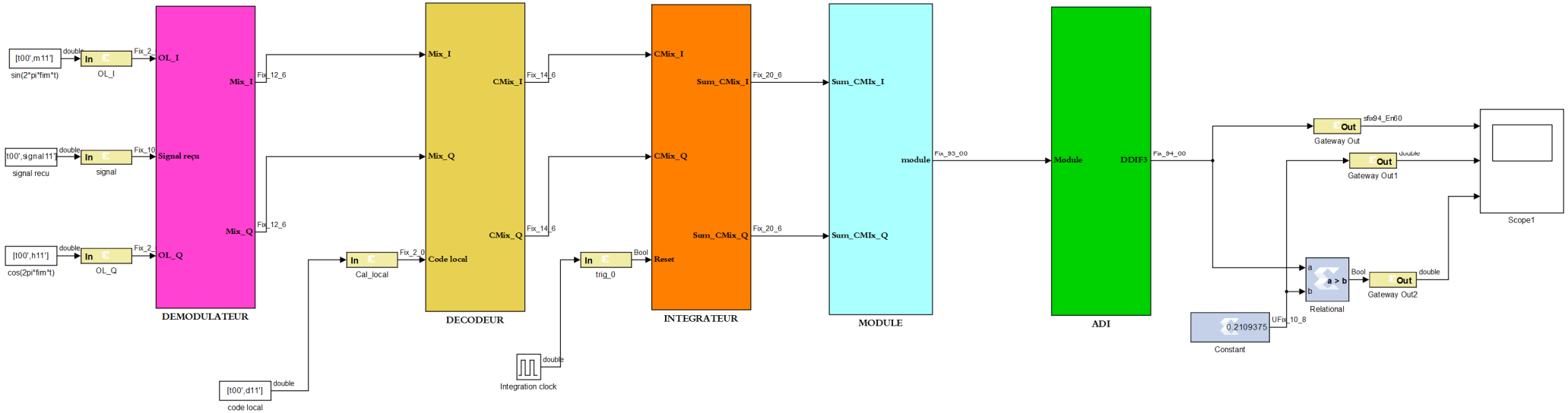
**Date Generated:** 10/02/2013 - 02:05:57





Time offset: 0

## **Annexe IV.2 : Simulation du détecteur quadratique avec intégration**



detect_diff_cw Project Status (10/02/2013 - 15:00:50)			
<b>Project File:</b>	detect_diff_cw.xise	<b>Parser Errors:</b>	No Errors
<b>Module Name:</b>	detect_diff_cw	<b>Implementation State:</b>	Mapped
<b>Target Device:</b>	xc5vfx70t-1ff1136	<b>• Errors:</b>	No Errors
<b>Product Version:</b>	ISE 14.1	<b>• Warnings:</b>	<a href="#">998 Warnings (998 new)</a>
<b>Design Goal:</b>	Balanced	<b>• Routing Results:</b>	
<b>Design Strategy:</b>	<a href="#">Xilinx Default (unlocked)</a>	<b>• Timing Constraints:</b>	
<b>Environment:</b>	<a href="#">System Settings</a>	<b>• Final Timing Score:</b>	

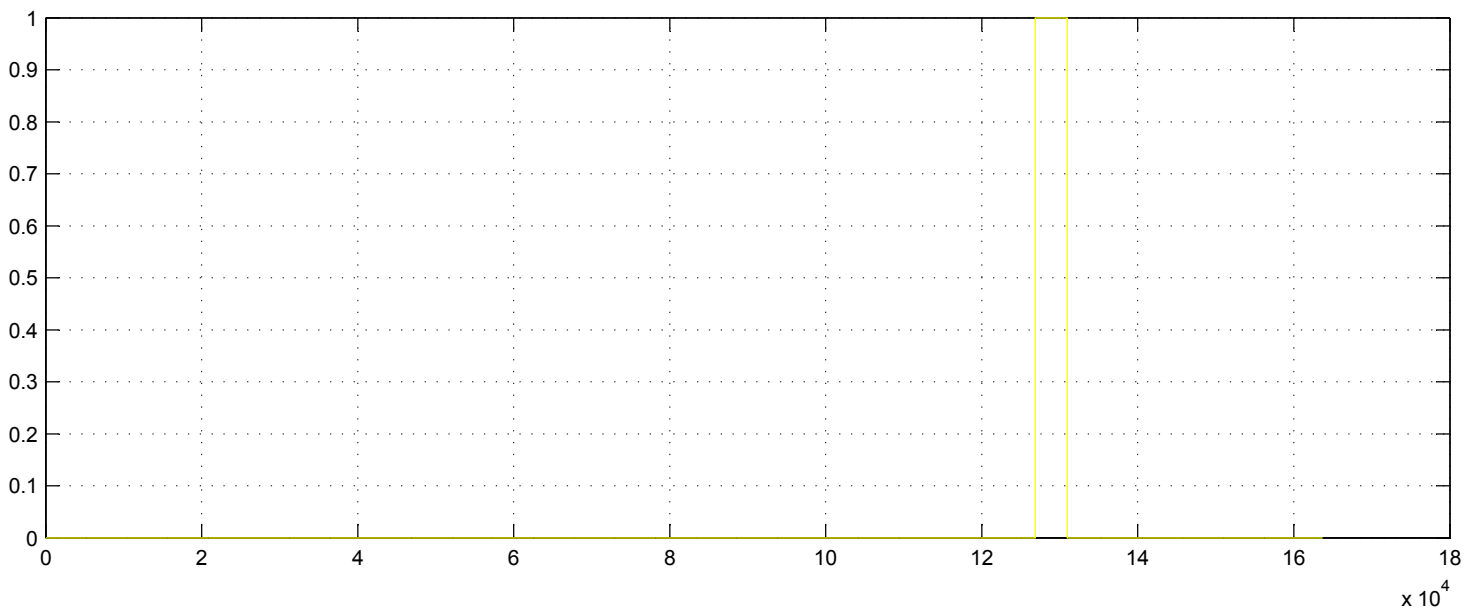
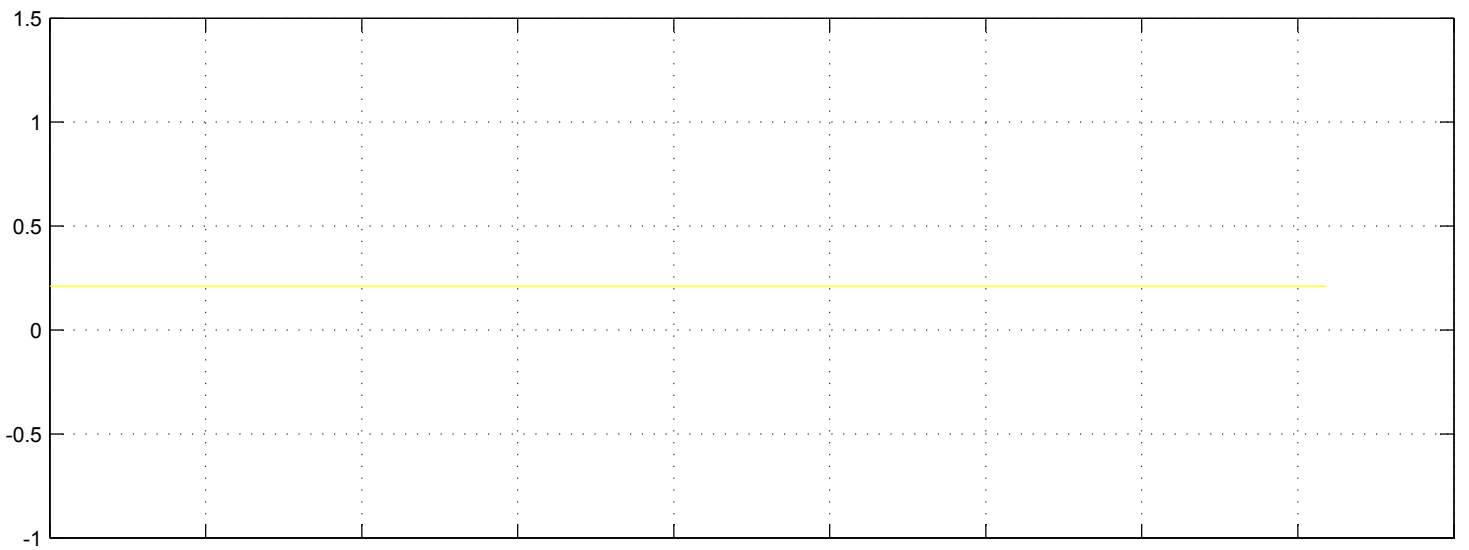
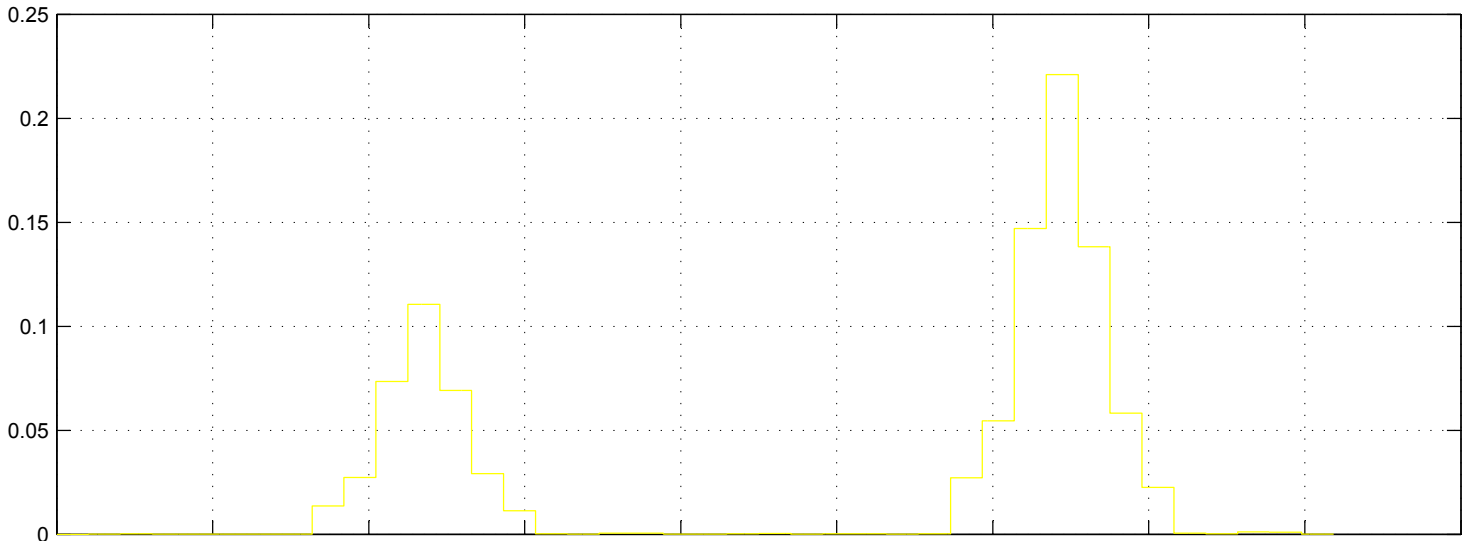
Device Utilization Summary				
<a href="#">[-]</a>				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	334	44,800	1%	
Number used as Flip Flops	332			
Number used as Latch-thrus	2			
Number of Slice LUTs	527	44,800	1%	
Number used as logic	336	44,800	1%	
Number using O6 output only	268			
Number using O5 output only	44			
Number using O5 and O6	24			
Number used as Memory	186	13,120	1%	
Number used as Shift Register	186			
Number using O6 output only	186			
Number used as exclusive route-thru	5			
Number of route-thrus	61			
Number using O6 output only	49			
Number using O5 output only	12			
Number of occupied Slices	212	11,200	1%	
Number of LUT Flip Flop pairs used	600			
Number with an unused Flip Flop	266	600	44%	
Number with an unused LUT	73	600	12%	
Number of fully used LUT-FF pairs	261	600	43%	
Number of unique control sets	37			
Number of slice register sites lost to control set restrictions	34	44,800	1%	
Number of bonded IOBs	123	640	19%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number used as BUFGs	1			
Number of DSP48Es	22	128	17%	
Average Fanout of Non-Clock Nets	1.75			

Detailed Reports					
<a href="#">[-]</a>					
Report Name	Status	Generated	Errors	Warnings	Infos
<a href="#">Synthesis Report</a>	Current	mer. 2. oct. 14:52:17 2013	0	<a href="#">996 Warnings (996 new)</a>	<a href="#">1 Info (1 new)</a>
<a href="#">Translation Report</a>	Current	mer. 2. oct. 14:58:39 2013	0	<a href="#">2 Warnings (2 new)</a>	0
<a href="#">Map Report</a>	Current	mer. 2. oct. 15:00:49 2013	0	0	<a href="#">55 Infos (55 new)</a>
Place and Route Report					
Power Report					

Post-PAR Static Timing Report					
Bitgen Report					

Secondary Reports		
Report Name	Status	Generated
<a href="#">Post-Synthesis Simulation Model Report</a>	Current	mer. 2. oct. 14:58:24 2013
<a href="#">Physical Synthesis Report</a>	Out of Date	mer. 2. oct. 15:00:48 2013

**Date Generated:** 10/02/2013 - 15:00:50



## REFERENCES BIBLIOGRAPHIQUES

- [01]. James Bao-Yen Tsui, “Fundamentals of Global Positioning System Receivers: A Software Approach”, John Wiley & Sons, ISBN 0-471-20054-9, 2000.
- [02]. Frank van Diggelen, “A-GPS: Assisted GPS, GNSS, and SBAS”, Artech House, ISBN-13: 978-1-59693-374-3, 2009.
- [03]. Kai Borre and al, “A Software-Defined GPS and Galileo Receiver A Single Frequency Approach”, ISBN-13: 978-0-8176-4540-3, 2007.
- [04]. <https://www.gps.gov/>
- [05]. G.S.Rao, “Global Navigation Satellite System with essentials of satellite communication”, Tata McGraw- Hill, ISBN: 978-0-07-070029-1, 2010.
- [06]. Steven C, Fisher and Kamran Ghassemi, “ GPS IIF- The Next Generation”, Proceedings Of The IEEE, Vol. 87, No. 1, January 1999.
- [07]. Peter J.G.Teunissen, Oliver Montenburk (Eds), “ Handbook of Global Navigation Satellite system”, Springer, e-ISBN: 978-3-319-42928-1, 2017.
- [08]. <https://tel.archives-ouvertes.fr/tel-01169567>
- [09]. BERNARD DIONNE, “ ÉTUDE ET ANALYSE TEMPS RÉEL D'UN CANAL DE RÉCEPTION GPS”, ÉCOLE DE TECHNOLOGIE SUPÉRIEURE MONTRÉAL, LE 22 DÉCEMBRE 2006.
- [10]. [www.mvd-fpga.com](http://www.mvd-fpga.com)
- [11]. [www.Xilinx.com](http://www.Xilinx.com)
- [13]. [www.mathworks.com](http://www.mathworks.com)
- [14]. Daniele Borio “A Statistical Theory for GNSS Signal Acquisition” Tesi di Dottorato POLITECNICO DI TORINO Marzo 2008.
- [15]. Global Navigation Satellite System (GNSS) Manual, Second Edition, ICAO Doc 9849, June 2012.
- [16]. Anna S. Gordon “The Stochastics of Diagnostic and Threat Detection Tests “A Dissertation Submitted to The Faculty of The Columbian College of Arts and Sciences of The George Washington University in partial fulfillment of the requirements for the degree of Doctor of Philosophy May 20, 2012
- [17]. Chun Yang, Mikel Miller, Erik Blasch, and Thao Nguyen “COMPARATIVE STUDY OF COHERENT, NONCOHERENT, AND SEMI-COHERENT INTEGRATION SCHEMES FOR GNSS RECEIVERS (PREPRINT)”, Sigtem

Technology, Inc. AFRL-RY-WP-TP-2008-1160.

[19]. HAL Id: tel-01886043  
<https://hal.archives-ouvertes.fr/tel-01886043>

[20]. HAL Id: tel-01220020  
<https://hal.archives-ouvertes.fr/tel-01220020>