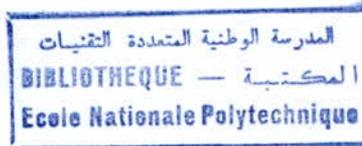


ECOLE NATIONALE POLYTECHNIQUE



DEPARTEMENT D'ELECTRONIQUE

Mémoire de  
MAGISTER EN ELECTRONIQUE

Option : Traitement du signal et Communication

Thème

ETUDE ET IMPLEMENTATION SUR FPGA  
D'UN SYSTEME DE COMMUNICATIONS A SPECTRE ETALE  
BASE SUR UNE MODULATION CHIRP

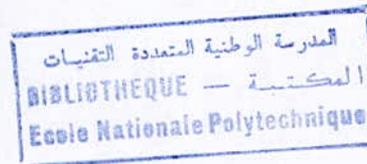
Présenté par

*M<sup>r</sup>* SARNI YACINE  
Ingénieur d'état en électronique (ENP)

Devant le jury composé de :

<i>M<sup>r</sup></i> M. Mehenni	(Maitre de Conférences ENP) Maitre de Recherche	<b>Président</b>
<i>M<sup>r</sup></i> A. Belouchrani	(Maitre de Conférences ENP) Maitre de Recherche	<b>Rapporteur</b>
<i>M<sup>r</sup></i> R. Sadoun	(Chargé de cours ENP) Chargé de Recherche	<b>Rapporteur</b>
<i>M<sup>r</sup></i> O. Stihi	(Chargé de cours ENP) Chargé de Recherche	<b>Examineur</b>
<i>M<sup>r</sup></i> H. Bousbia-Salah	(Chargé de cours ENP) Chargé de Recherche	<b>Examineur</b>
<i>M<sup>lle</sup></i> N. Izeboudjen	Chargé de Recherche CDTA	<b>Examineur</b>
<i>M<sup>r</sup></i> A. Oudjida	Chercheur CDTA	<b>Invité</b>

ECOLE NATIONALE POLYTECHNIQUE



DEPARTEMENT D'ELECTRONIQUE

Mémoire de  
MAGISTER EN ELECTRONIQUE

**Option** : Traitement du signal et Communication

Thème

ETUDE ET IMPLEMENTATION SUR FPGA  
D'UN SYSTEME DE COMMUNICATIONS A SPECTRE ETALE  
BASE SUR UNE MODULATION CHIRP

Présenté par

*M<sup>r</sup>* SARNI YACINE  
Ingénieur d'état en électronique (ENP)

Devant le jury composé de :

<i>M<sup>r</sup></i> M. Mehenni	(Maitre de Conférences ENP) Maitre de Recherche	<b>Président</b>
<i>M<sup>r</sup></i> A. Belouchrani	(Maitre de Conférences ENP) Maitre de Recherche	<b>Rapporteur</b>
<i>M<sup>r</sup></i> R. Sadoun	(Chargé de cours ENP) Chargé de Recherche	<b>Rapporteur</b>
<i>M<sup>r</sup></i> O. Stihi	(Chargé de cours ENP) Chargé de Recherche	<b>Examineur</b>
<i>M<sup>r</sup></i> H. Bousbia-Salah	(Chargé de cours ENP) Chargé de Recherche	<b>Examineur</b>
<i>M<sup>lle</sup></i> N. Izeboudjen	Chargé de Recherche CDTA	<b>Examineur</b>
<i>M<sup>r</sup></i> A. Oudjida	Chercheur CDTA	<b>Invité</b>

## Abstract

In this work, we present a novel method for enhancing the interference immunity of a direct sequence spread spectrum system by applying a chirp signal to modulate the spreaded sequence and by using a time varying filter to recover the transmitted signal. An analytic expression of the receiver SNR is derived and compared with simulation results. Also, we implement our application in FPGA using the VHDL programming language and an extensive simulations of the system have been performed to test the effectiveness of the implementation.

**Key words :** Chirp Modulation, Spread spectrum, SNR, FPGA

## Résumé

Dans ce travail, on présente une nouvelle méthode pour améliorer les performances d'un système de communication à spectre étalé en terme d'immunité aux interférences en lui appliquant une modulation avec un signal chirp. A la réception, un filtre à réponse impulsionnelle variable dans le temps est utilisé pour la récupération du signal d'information, une expression analytique du SNR à la réception a été calculée, et comparée avec les résultats de simulation. Aussi, une implémentation sur FPGA du système proposé a été effectuée en utilisant le langage de programmation VHDL, suivit des simulations pour valider le fonctionnement du circuit implémenté.

**Mots clés:** Modulation chirp, Spectre étalée, SNR, FPGA

## ملخص

يخص هذا البحث دراسة نظام إتصال رقمي قائم على تمديد طيف الإشارة الرقمية وتعديلها بواسطة إشارة ذات تردد يتغير خطيا بدلالة الزمن. في الإستقبال، نستعمل مرشح ذات إجابة منتهية متغيرة بدلالة الزمن و متزامنة مع إشارة التعديل. عبارة تحليلية لنسبة الإشارة و تشوش SNR تم إستنتاجها و مقارنتها مع النتائج التطبيقية. قمنا كذلك ببرمجة النظام المدروس باللغة الوصفية VHDL .

كلمات مفتاحية : إتصال رقمي، تمديد طيف، نسبة الإشارة و تشوش، مرشح.

# Sommaire

<b>1</b>	<b>Système de Communication à Spectre étalé</b>	<b>1</b>
1.1	Modèle d'un système de communication numérique . . . . .	1
1.2	Concept de la Direct Sequence Spread Spectrum . . . . .	2
1.3	Génération des séquences PN . . . . .	4
1.3.1	Les codes m-sequence . . . . .	5
1.3.2	Les codes Gold . . . . .	6
1.3.3	Les codes Hadamard-Walch . . . . .	7
1.4	Avantage d'un système DSSS . . . . .	7
1.4.1	Suppression des interférences . . . . .	7
1.4.2	Elimination des trajets multiples . . . . .	9
1.4.3	Accès multiples . . . . .	10
1.5	Conclusion . . . . .	10
<b>2</b>	<b>Synchronisation</b>	<b>11</b>
2.1	Introduction . . . . .	11
2.2	Processus d'acquisition . . . . .	12
2.2.1	Architecture de la boucle d'acquisition . . . . .	13
2.2.2	Performance d'une boucle d'acquisition . . . . .	16
2.3	Tracking . . . . .	19
2.4	Conclusion . . . . .	21

<b>3</b>	<b>Système DSSS modulé par une chirp</b>	<b>22</b>
3.1	Introduction . . . . .	22
3.2	Description du système . . . . .	23
3.3	Synchronisation . . . . .	26
3.3.1	Processus d'acquisition . . . . .	26
3.3.2	Tracking . . . . .	27
3.4	Performance du système . . . . .	28
3.5	Rapport signal sur bruit et interférence du récepteur . . . . .	29
3.5.1	Système DSSS classique . . . . .	29
3.5.2	Système proposé . . . . .	30
3.6	Rendement spectral . . . . .	33
3.7	Evaluation des performances . . . . .	33
3.8	Conclusion . . . . .	40
<b>4</b>	<b>Les FPGAs et l'arithmétique binaire</b>	<b>41</b>
4.1	Field Programmable Gate Arrays (FPGA) . . . . .	41
4.2	Programmation en VHDL . . . . .	42
4.2.1	Niveaux de description comportemental . . . . .	43
4.2.2	Le niveau RTL . . . . .	43
4.2.3	Description structurelle . . . . .	43
4.3	Format des données . . . . .	44
4.4	Multiplication en virgule fixe . . . . .	45
4.5	L'arrondi . . . . .	45
4.6	La saturation arithmétique . . . . .	48
4.7	Méthodologie d'implémentation . . . . .	48
<b>5</b>	<b>Implémentation sur FPGA du système proposé</b>	<b>50</b>
5.1	Implémentation de l'émetteur . . . . .	50
5.1.1	Générateur de séquence PN . . . . .	51

5.1.2	Génération du signal chirp . . . . .	51
5.1.3	Compteur modulo-31 . . . . .	52
5.2	Implémentation du récepteur . . . . .	53
5.2.1	Implémentation du filtre . . . . .	53
5.2.2	Implémentation du démodulateur . . . . .	59
5.2.3	Implémentation du bloc de synchronisation . . . . .	63
5.2.4	Tracking . . . . .	65
5.3	Conclusion . . . . .	67
<b>6</b>	<b>Evaluation des performances</b>	<b>68</b>
6.1	Implémentation . . . . .	68
6.2	Vérification des résultats de simulation . . . . .	71
6.2.1	Résultats de filtrage et de démodulation . . . . .	71
6.2.2	Vérification de la boucle de synchronisation . . . . .	72
6.3	Conclusion . . . . .	73
<b>7</b>	<b>Conclusion générale</b>	<b>74</b>
<b>8</b>	<b>Bibliographie</b>	<b>76</b>
<b>9</b>	<b>Annexes</b>	<b>78</b>
<b>A</b>	<b>Génération Des Séquences PN</b>	<b>78</b>
<b>B</b>	<b>Simulation</b>	<b>80</b>
<b>C</b>	<b>Système Sous Simulink</b>	<b>85</b>
<b>D</b>	<b>Publication</b>	<b>89</b>

# Introduction

Les systèmes de communications à spectre étalé sont des systèmes utilisant une bande de fréquence beaucoup plus grande que le débit de la source d'information. Ils sont utilisés dans les applications multi-utilisateurs, les communications sécurisées et ils permettent de réduire l'effet des interférences et des trajets multiples. Cependant, les performances de ces systèmes peuvent être améliorées en terme d'immunité aux interférences.

Dans ce travail, nous proposons une méthode pour augmenter les performances de ces systèmes en appliquant une modulation chirp au signal d'information étalée. A la réception, un filtre à réponse impulsionnelle finie variable dans le temps est utilisé pour retrouver le signal d'information.

Le premier chapitre présente une introduction générale des systèmes de communications à spectre étalé, leurs caractéristiques et leurs avantages par rapport à d'autres systèmes de communications. Le deuxième chapitre, traite des méthodes de synchronisation qui sont indispensables dans ce type de système.

Dans le troisième chapitre, nous présentons le système que nous avons proposé. Les performances en terme de probabilité d'erreur de réception, obtenues analytiquement, sont comparées avec les résultats obtenus par simulation. Aussi, nous proposons une méthode pour la synchronisation du système. Les chapitres qui suivent, traitent de l'implémentation du système proposé sur FPGA en utilisant le langage VHDL. Une description des différents composants entrant dans la réalisation du circuit est présentée. Le dernier chapitre donne les résultats obtenus après implémentation du système et les simulations qui attestent du bon fonctionnement du système proposé.

# Chapitre 1

## Systeme de Communication à Spectre étalé

Les systèmes de communications à spectre étalé sont généralement utilisés dans les communications numériques dans différents domaines d'applications, telle que la suppression d'interférence et les systèmes multi-utilisateurs. Dans ce chapitre, on présentera le concept de ce type de système, leurs caractéristiques, les techniques de modulation et de démodulation, et leurs avantages comparés à des systèmes de communications numériques classiques utilisant une modulation binaire.

### 1.1 Modèle d'un système de communication numérique

Soit un système de communications numériques, utilisé pour la transmission d'une séquence d'information binaire  $b_m \in \{\pm 1\}$  équiprobable, à travers un canal de transmission, de sorte que le signal  $r_m$  reçu par le récepteur est de la forme :

$$r_m = \mathcal{E}b_m + w_m \quad (1.1)$$

Où :  $\mathcal{E} > 0$  est l'énergie de l'impulsion qui représente chaque bits  $b_m$ ,  $w_m$  est un bruit blanc additif Gaussien de moyenne nulle et de variance  $\sigma^2$ .

Dans ce cas, le récepteur optimal est un simple détecteur à seuil qui base sa décision sur

le signe de la variable  $r_m$  :

$r_m \geq 0$  : alors +1 a été transmit

$r_m < 0$  : alors -1 a été transmit

En général, en présence de bruit additif Gaussien, un récepteur à corrélation est un détecteur optimal "au sens du maximum de vraisemblance", il détermine le bit reçu en fonction de la valeur d'une variable de décision  $y_m$ , qui est une fonction du signal reçu  $r_m$ . Dans le cas précédant,  $y_m = r_m$  est la variable de décision dont les statistiques déterminent les performances du récepteur.

À partir de l'équation (1.1), il est claire que  $r_m$  est une variable gaussienne de moyenne  $\mathcal{E}b_m$  et de variance  $\sigma^2$ , les performances du système, mesurées en terme de probabilité d'erreur de réception de cette modulation d'amplitude binaire sont déterminées par le rapport  $\frac{\mathcal{E}}{\sigma}$  [1].

## 1.2 Concept de la Direct Sequence Spread Spectrum

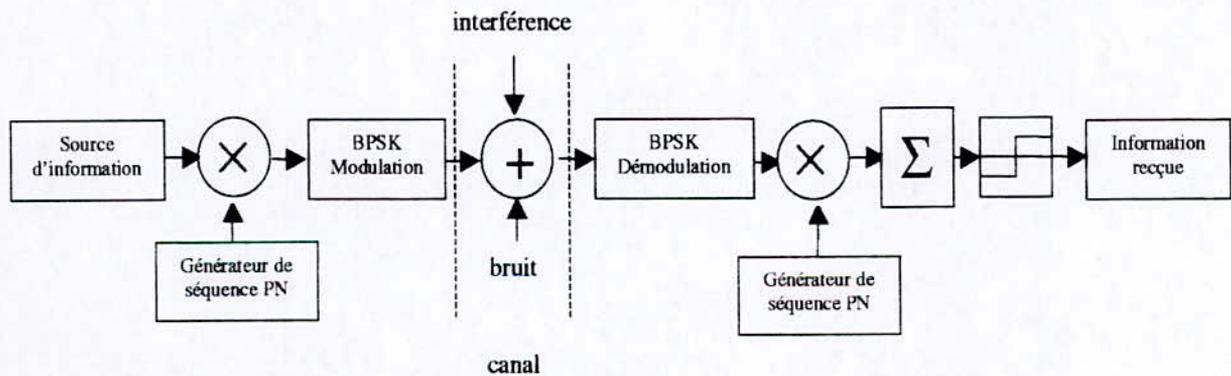


Figure 1.1: Modèle d'un système de communications Direct Sequence Spread Spectrum

Le système *Direct Sequence Spread Spectrum* (DSSS) utilisé pour la transmission d'informations numériques, dont la figure (1.1) donne le bloc diagramme [1], se caractérise par le fait qu'il utilise une bande de transmission  $R_c = \frac{1}{T_c}$ , beaucoup plus grande que le débit de la source d'information  $R_b \text{ bit/s} = \frac{1}{T_b}$ . Le facteur d'expansion de la bande de fréquence du signal transmis est  $N = \frac{R_c}{R_b} = \frac{T_b}{T_c}$ . L'extension de la bande de fréquence est

obtenue en modulant le signal d'information avec une séquence binaire  $\{c_n\}_{n=0}^{N-1}$ , ayant une bande de fréquence plus grande que celle du signal d'information. Cette modulation est un simple produit entre le signal d'information et les éléments de la séquence  $\{c_n\}$  appelés **chip** de durée  $T_c$ , l'opération de multiplication des deux séquences est connue sous le nom de *spreading* [1]. Le signal transmis prend la forme :

$$s_n = \mathcal{E}_c c_n b, \quad n = 0, \dots, N-1. \quad (1.2)$$

Ici,  $N$  impulsions sont transmises pour représenter un seul bit d'information ( $b$ ). La durée  $T_b$  de chaque bit est divisée en  $N$  intervalles de durée  $T_c = \frac{T_b}{N}$  durant lesquels une impulsion d'énergie  $\mathcal{E}_c = \mathcal{E}/N$  est transmise. Le résultat de cette opération est l'étalement du spectre du signal sur une bande de fréquence beaucoup plus grande puisque  $T_c \ll T_b$ .

La figure (1.2) donne la relation entre le signal d'information et la séquence  $\{c_n\}_{n=0}^{N-1}$ .

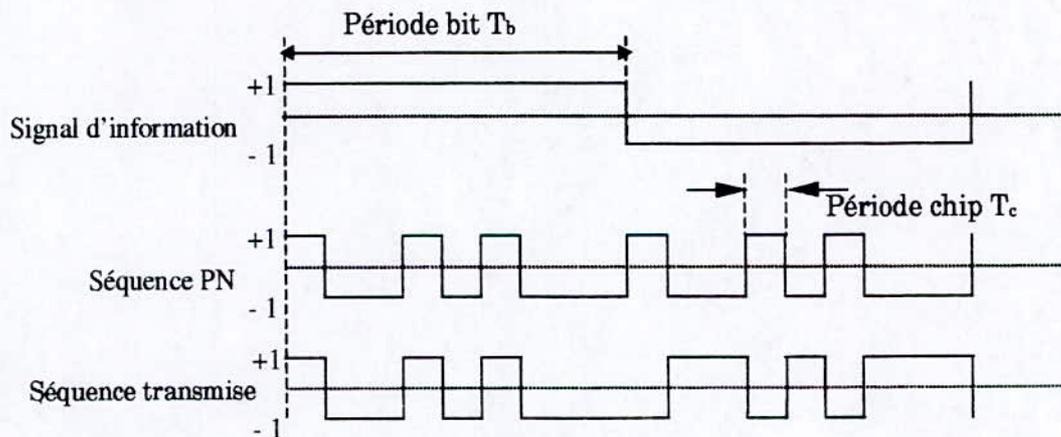


Figure 1.2: Relation entre la séquence PN et les bits d'informations

On suppose que la séquence  $\{c_n\}$  est périodique de période  $N$  et doit satisfaire les deux propriétés suivantes :

$$\sum_{n=0}^{N-1} c_n \approx 0 \quad (1.3)$$

$$\sum_{n=0}^{N-1} c_n c_{n+i} \approx \begin{cases} N & i = 0 \\ 0 & 0 < |i| < N \end{cases} \quad (1.4)$$

Les deux propriétés de cette séquence **déterministe** sont analogues aux propriétés d'un bruit blanc à moyenne nulle, c'est pour cela qu'on les appelle des séquences pseudo-aléatoires (PN).

Ces conditions sont idéales, mais en pratique les séquences générées se caractérisent par le fait que le nombre de +1 et de -1 dans une période diffère de un.

Dans ce cas, les équations ( 1.3) et ( 1.4) deviennent :

$$\sum_{n=0}^{N-1} c_n = \pm 1 \quad (1.5)$$

$$\sum_{n=0}^{N-1} c_n c_{n+i} \approx \begin{cases} N & i = 0 \\ -1 & 0 < |i| < N \end{cases} \quad (1.6)$$

A la réception, le récepteur fait la corrélation entre le signal reçu et la séquence de son générateur local de séquence PN. Cette opération est connue sous le nom de *despreading*, le résultat de cette opération est la variable de décision  $y$  :

$$y = \sum_{n=0}^{N-1} r_n c_n = \sum_{n=0}^{N-1} (\mathcal{E}_c b c_n + w_n) c_n, \quad (1.7)$$

En se basant sur les propriétés de la séquence  $\{c_n\}$  (1.6), l'équation (1.7) devient :

$$y = N \mathcal{E}_c b + \sum_{n=0}^{N-1} w_n c_n \quad (1.8)$$

La variable de décision  $y$  est gaussienne de moyenne  $Nb\mathcal{E}_c = \mathcal{E}b$  et de variance  $\sigma^2$ . Si on compare ce résultat avec un système sans Spread Spectrum, on remarque que dans un canal avec un bruit blanc additif gaussien, il n'y a pas d'amélioration des performances: Si le débit de la source est augmenté d'un facteur  $N$ , la bande du signal augmente et la puissance du bruit augmente d'un facteur  $N$ .

### 1.3 Génération des séquences PN

Il existe une multitude de classes de code qui peuvent être utilisés dans les système DSSS, les codes les plus utilisés sont les *maximum length sequence* ou m-sequence, les codes Gold et les codes Hadamard-Walch [2]

### 1.3.1 Les codes m-sequence

Les codes m-sequence sont des codes périodiques de période  $N = 2^m - 1$ , telle que la valeur de  $m$  détermine l'ordre du code. Leur fonction d'autocorrélation présente un pic de  $2^m - 1$  pour un décalage nul. Les séquences PN de ce type sont caractérisées par le fait que pour chaque période  $N$ , elles contiennent  $2^{m-1} - 1$  (+1) et  $2^{m-1}$  (-1).

Ce type de code est très utilisé dans les systèmes DSSS parce que sa fonction d'autocorrélation présente une valeur très faible pour un décalage différent de zéro. Un autre avantage de ce type de code est leur simplicité de génération. En effet, comme le montre la figure (1.3), ces codes sont obtenus à partir d'un simple registre à décalage et une fonction de retour  $f(x_1, x_2, \dots, x_m)$  qui est une somme modulo 2 des éléments  $x_i$  du registre à décalage.

Le résultat de la fonction  $f(x_1, \dots, x_m)$  représente l'entrée du registre et la sortie de ce dernier donne la séquence PN. Comme indiqué dans la figure (1.3), les coefficients  $a_i$  sont les connections qu'on utilise pour la boucle de retour :

Si,  $a_i = 0$  l'élément  $x_i$  n'est pas utilisé dans la boucle de retours, si par contre, il est utilisé  $a_i = 1$ . Des combinaisons différentes des coefficients  $a_i$  permettent de générer des séquences PN différentes.

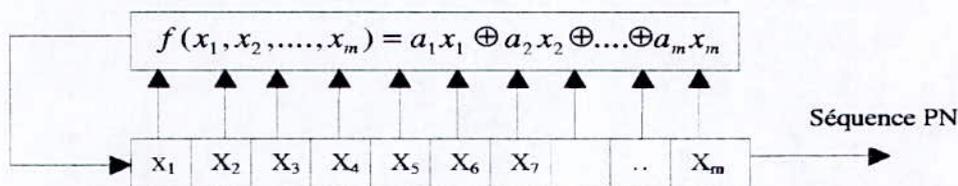


Figure 1.3: générateur de séquences PN

Cependant, un des inconvénients de ce type de séquence est que pour un ordre  $m$  donné, il n'existe pas beaucoup de m-séquence différente, ce qui limite leurs utilisations dans les applications multi-utilisateurs.

Pour remédier à la limitation imposée par ce type de code, une autre classes de code est utilisée pour les applications multi-utilisateurs, ces codes sont appelés *Gold codes*.

### 1.3.2 Les codes Gold

Dans un système multi-utilisateurs, on a besoin d'avoir un ensemble de code de même longueur mais avec de bonnes propriétés de la fonction d'intercorrélation.

Les codes Gold sont très utiles du fait qu'un grand nombre de codes (avec la même longueur et une fonction d'intercorrélation contrôlée) peuvent être générés. Ils ont des fonctions d'intercorrélations meilleures que celle des codes m-séquence, de plus ils ne requièrent qu'une paire de registre à décalage avec une boucle de retour.

Les codes Gold sont obtenus par une addition modulo 2 de deux codes m-sequence de même longueur. Les deux séquences sont additionnées chip par chip avec une même horloge. Puisque les deux codes sont de même longueur, les deux générateurs de code maintiennent la même phase, et le code Gold généré est de même longueur, chaque changement de la phase de l'un des deux générateurs de m-sequence, génère un nouveau code Gold.

La figure (1.4) donne une représentation d'un générateur de séquence Gold.

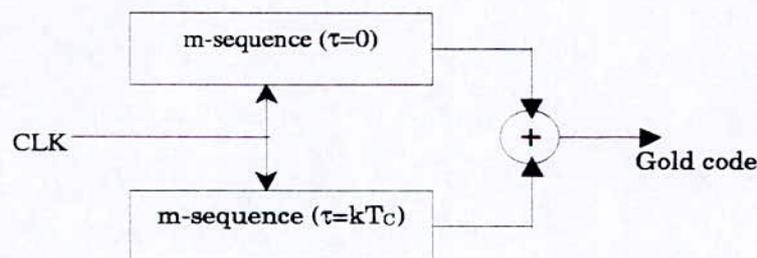


Figure 1.4: générateur de séquences Gold

Cependant, les séquences produites ne sont pas maximales. Leur fonction d'autocorrélation ne prend pas que deux valeurs comme dans le cas des m-séquences.

Avec chaque paire de registres de longueurs  $L$ , on peut générer un ensemble de  $L$  codes différents, en faisant la somme modulo 2 entre la sortie d'un registre et les  $L$  phases possibles du deuxième registre ou vis versa. Si en plus, on ajoute les deux m-séquences, on obtient pour chaque paire de registres, un total de  $L + 2$  codes différents de période  $2^L - 1$ . En plus du grand nombre de code qu'on peut générer, les codes Gold peuvent être choisis de telle sorte que la fonction d'intercorrélation entre ces codes ne prend que les trois

valeurs suivantes  $-1, -t(m), t(m) - 2$  telle que  $t(m)$  est définie comme suit :

$$t(m) = \begin{cases} 2^{(m+1)/2} + 1 & \text{pour } m \text{ pair} \\ 2^{(m+2)/2} + 1 & \text{pour } m \text{ impair} \end{cases} \quad (1.9)$$

### 1.3.3 Les codes Hadamard-Walch

Les codes Hadamard-Walch (H.W) sont générés dans un ensemble de  $N = 2^n$  codes de longueur  $N = 2^n$ , suivant l'algorithme de génération suivant :

$$H_N = \begin{vmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{vmatrix}$$

avec :  $H_1 = 1$ . Les lignes ou les colonnes de la matrice  $H_N$  sont des codes (H.W).

Dans chaque cas, la première colonne de la matrice contient uniquement des uns, chacune des autres colonnes contient  $N/2$  zéros et  $N/2$  un, et toutes les colonnes sont mutuellement orthogonales

$$\sum_{k=0}^{N-1} h_{ik}h_{jk} = 0 \quad (1.10)$$

Pour toutes les colonnes  $i$  et  $j$ , l'intercorrélacion entre deux codes (H.W) de la même matrice est nulle.

## 1.4 Avantage d'un système DSSS

L'avantage d'un système basé sur la (DSSS) est son effet sur les signaux corrélés et à bande limitée, telle que les interférences, les trajets multiples et les interférences multi-utilisateurs "applications multi-utilisateurs" [3]

### 1.4.1 Suppression des interférences

Supposons que le canal contient une interférence, une constante inconnue  $I$  qui s'additionne au signal reçu :

$$r_n = \mathcal{E}_c b c_n + I + w_n, \quad n = 0, 1, \dots, N - 1. \quad (1.11)$$

à la sortie du corrélateur, on aura la variable de décision :

$$\begin{aligned}
 y &= N\mathcal{E}_c b + I \sum_{n=0}^{N-1} c_n + \sum_{n=0}^{N-1} w_n c_n \\
 &\approx N\mathcal{E}_c b + 0 + \sum_{n=0}^{N-1} w_n c_n
 \end{aligned} \tag{1.12}$$

La variable de décision  $y$  est de moyenne  $\mathcal{E}b$  et de variance  $\sigma^2$ , l'interférence est éliminée par l'opération de *despreading*.

Dans le cas d'un système classique la variable de décision aurait une moyenne  $\mathcal{E}b + I$ , ce qui rendra le système inutilisable pour de larges valeurs de  $|I|$ .

Dans le cas d'une interférence  $i(t)$  à bande limitée, la partie du signal reçu qui contient l'interférence est de la forme :

$$v(t) = i(t)c(t) \tag{1.13}$$

L'interférence  $i(t)$  et la séquence PN  $c(t)$  sont statistiquement indépendant, le produit  $v(t)$  est aussi un processus aléatoire avec une fonction d'autocorrélation égale au produit des fonctions d'auto corrélation des deux signaux. La densité spectrale de puissance (dsp) de  $v(t)$  est égale à la convolution des dsp de  $i(t)$  et de  $c(t)$ . Puisque le signal  $c(t)$  occupe une bande  $R_c$ , le résultat de la convolution des deux spectres est l'étalement de la dsp du signal  $v(t)$  sur une bande de fréquence  $R_c$ . Si  $i(t)$  est une interférence qui occupe une bande de fréquence plus petite que  $R_c$ , la dsp de  $v(t)$  occupe une bande de fréquence qui est au minimum égale à  $R_c$ .

A la réception, le corrélateur a une bande passante égale à la bande du signal d'information  $\frac{1}{T_b}$ . Puisque  $1/T_b \ll R_c$ , seulement une fraction de la puissance de l'interférence apparaît à la sortie du corrélateur, cette fraction est égale au rapport entre les deux bandes de fréquence  $\frac{1}{T_b}$  et  $R_c$  :

$$\frac{1/T_b}{R_c} = \frac{1}{R_c T_b} = \frac{T_c}{T_b} = \frac{1}{L_c} \tag{1.14}$$

En d'autres termes, la multiplication de l'interférence par la séquence PN permet de réduire la puissance de l'interférence et augmente les performances du système avec un facteur  $L_c$  qui est le gain du processus. La figure (1.5) donne une représentation de l'effet de multiplier l'interférence par la séquence PN.

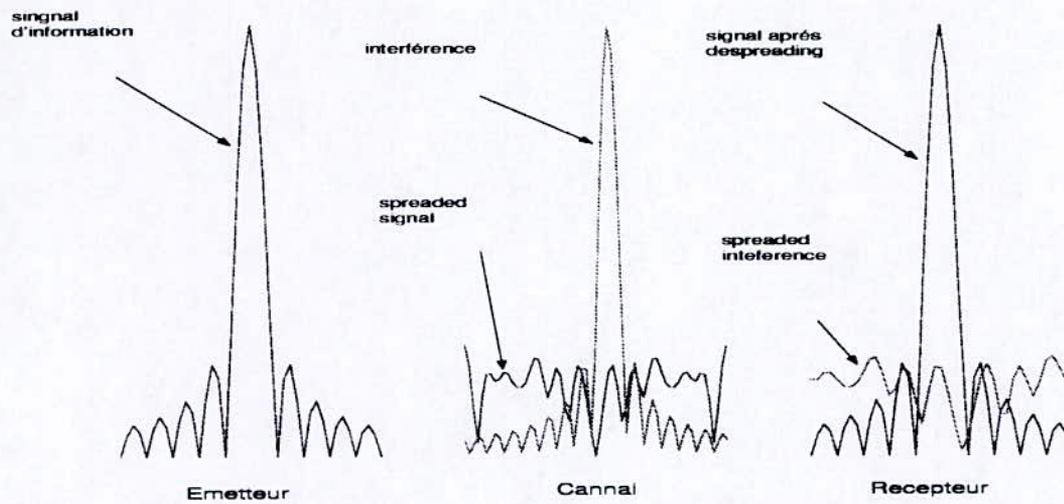


Figure 1.5: effet d'un système DSSS sur une interférence

### 1.4.2 Elimination des trajets multiples

Supposant que le canal introduit des trajets multiples, un trajet direct et un autre trajet qui cause une seconde copie du signal, qui arrive au récepteur avec un retard  $l$  et une atténuation  $\beta$ . Pour simplifier les notations, on supposera que  $\mathcal{E}_c = 1$ , le signal reçu durant l'intervalle du bit transmit  $b_m$  est :

$$r_n = \begin{cases} b_m c_n + \beta b_{m-1} c_{N-l+n} & n = 0, \dots, l-1 \\ b_m c_n + \beta b_m c_{n-l} & n = l, \dots, N-1 \end{cases} \quad (1.15)$$

On suppose que le retard est inférieur à la durée d'un bit  $l < N$ . Le second trajet crée une interférence sur la version retardée du bit désiré  $b_m$  et sur l'ancien bit transmit  $b_{m-1}$ .

A la réception on aura :

$$\begin{aligned} y &= N b_m + \beta b_{m-1} \sum_{n=0}^{l-1} c_{N-l+n} c_n + \beta b_m \sum_{n=l}^{N-1} c_{n-l} c_n + \sum_{n=0}^{N-1} w_n c_n \\ &\approx N b_m + 0 + 0 + \sum_{n=0}^{N-1} w_n c_n \end{aligned} \quad (1.16)$$

Les interférences inter symboles sont éliminées par un système SS

### 1.4.3 Accès multiples

Supposons le cas où  $k$  utilisateurs qui transmettent les bits  $b_m^{(k)}$  avec des séquences  $\{c_n^{(k)}\}$  différentes, telle que leur fonction d'autocorrélation est comme suit :

$$\sum_{n=0}^{N-1} c_n^{(k)} c_{n+i}^{(j)} \approx \begin{cases} N & k = j, i = 0 \\ 0 & \text{ailleurs} \end{cases} \quad (1.17)$$

On s'intéresse qu'au signal transmit par l'utilisateur  $k = 1$ , qui utilise la séquence  $\{c_n^{(1)}\}$ .

Le signal reçu par le récepteur est de la forme :

$$r_n = \sum_{k=1}^K b_m^{(k)} c_n^{(k)} + w_n \quad (1.18)$$

la réception du signal du premier utilisateur génère la variable de décision  $y_m^{(1)}$  telle que :

$$\begin{aligned} y_m^{(1)} &= b_m^{(1)} \sum_{n=0}^{N-1} (c_n^{(1)})^2 + \sum_{k=2}^K b_m^{(k)} \sum_{n=0}^{N-1} c_n^{(k)} c_n^{(1)} + \sum_{n=0}^{N-1} w_n c_n^{(1)} \\ &\approx N b_m^{(1)} + 0 + \sum_{n=0}^{N-1} w_n c_n^{(1)} \end{aligned} \quad (1.19)$$

La propriété d'intercorrélation des séquences  $\{c_n^{(k)}\}$  (orthogonalité mutuelle) permet l'utilisation de la SS dans les systèmes à accès-multiples qui sont appelés *Code-Division Multiple Access (CDMA)*.

## 1.5 Conclusion

Dans ce chapitre nous avons décrit le principe des systèmes de communications à spectre étalé et leurs utilisations. L'étalement du spectre du signal d'information est réalisé avec différentes catégories de séquence PN, qui donnent au système des propriétés lui permettant d'être plus robuste aux interférences. Aussi, on peut les utiliser dans les applications multi-utilisateurs et les communications sécurisées puisqu'il faut connaître la séquence PN transmise pour pouvoir réaliser la démodulation.

Toutefois, les propriétés des séquences PN imposent, pour avoir un rendement maximum, une synchronisation parfaite entre l'émetteur et le récepteur.

# Chapitre 2

## Synchronisation

### 2.1 Introduction

Dans un système DSSS, avant que le récepteur puisse commencer à démoduler le signal reçu, il doit aligner son générateur de séquence PN avec la séquence PN de l'émetteur. La nature des systèmes DSSS exige un alignement le plus précis possible entre les deux séquences ; pour avoir un gain maximum en puissance, le temps relatif entre les séquences de l'émetteur et du récepteur doit être ajusté pour compenser les variations de phase dues au décalage en fréquence, au temps de transmission et au décalage initial des phases des générateurs PN du récepteur et de l'émetteur.

Le processus de synchronisation entre l'émetteur et le récepteur s'effectue en deux étapes distinctes. Premièrement, un alignement initial approximatif est effectué de telle sorte que les deux séquences soient alignées avec un offset relativement petit. Cette étape est connue sous le nom *d'acquisition*. Durant cette étape, l'émetteur transmet une séquence PN connue du récepteur pour qu'il puisse effectuer l'acquisition ; la seconde étape consiste à affiner la synchronisation entre les deux séquences et à maintenir continuellement le meilleur alignement possible entre ces dernières. Cette étape est connue sous le nom de *tracking*.

## 2.2 Processus d'acquisition

Soit la séquence PN  $\{c^{(0)}\}$  de période  $L$ , avec les éléments :

$(\dots, c^{(0)}(0), c^{(0)}(1), c^{(0)}(2), \dots), c^{(0)}(j) \in \{-1, +1\}$ . La  $k^{ieme}$  phase  $\{c^{(k)}\}$  de la séquence  $\{c^{(0)}\}$  est le  $k^{ieme}$  décalage à gauche de cette dernière, telle que :

$$c^{(k)}(i) = c^{(0)}(i + k), i = \dots, -2, -1, 0, 1, 2, \dots \quad (2.1)$$

Soit le train d'impulsion transmit par l'émetteur pour effectuer la synchronisation :

$$c(t) = \sum_{j=-\infty}^{j=\infty} c^{(0)}(j) \Pi_{T_c}(t - jT_c) \quad (2.2)$$

Où :  $\Pi_{T_c}$  est une impulsion rectangulaire de période  $T_c$  centrée sur  $\frac{T_c}{2}$  ; le signal reçu est de la forme :

$$r(t) = c(t + \delta T_c) + w(t) \quad (2.3)$$

Où  $w(t)$  est un bruit blanc additif de moyenne nulle et de variance  $\sigma^2$  et  $\delta T_c$  est un décalage inconnu dans le temps. Le processus d'acquisition consiste à trouver un estimé  $\delta_{est} T_c$  du déphasage  $\delta T_c$  telle que  $|\delta T_c - \delta_{est} T_c|$  soit inférieur à une constante  $\zeta$ .

Puisque  $c(t)$  est de période  $LT_c$ , on suppose que  $\delta T_c \in [0, LT_c]$  ou bien  $\delta \in [0, L]$ , de manière similaire on suppose que  $\delta_{est} \in [0, L]$ .

On dit que l'acquisition est effectuée si  $|\delta - \delta_{est}| \approx 0$  ou, si  $|\delta - \delta_{est}| \approx L$ . Plus précisément, le signal est acquis si :

$$\min\{|\delta - \delta_{est}|, L - |\delta - \delta_{est}|\} \leq \zeta \quad (2.4)$$

Pour faire un compromis entre le processus d'acquisition et de tracking, la valeur de  $\zeta$  est prise égale à  $\frac{T_c}{2}$ . En effet, plus la valeur de  $\zeta$  est petite plus la synchronisation est précise, ce qui réduit le décalage initial dans la boucle de tracking, et permet une convergence plus rapide vers une synchronisation plus affinée dans la boucle de tracking, d'un autre coté, l'implémentation d'une boucle d'acquisition pour des valeur de  $\zeta < \frac{T_c}{2}$  est compliquée, mais reste relativement simple pour  $\zeta = \frac{T_c}{2}$ .

L'estimation du décalage entre les deux séquences est basée sur les propriétés de la fonction d'autocorrélation de la séquence PN. La valeur de la fonction d'intercorrélation entre les deux séquences est grande si les deux séquences sont synchronisées, faible dans l'autre situation. Il suffit de comparer la corrélation entre le signal reçu et la séquence locale du récepteur à un seuil pour décider de la synchronisation des deux séquences.

La boucle d'acquisition de la séquence PN peut faire des erreurs de détection :

Occasionnellement, la boucle d'acquisition peut déclarer que les deux séquences sont alignées alors qu'elles ne le sont pas dans le cas où le seuil de détection est trop bas. D'autre part, si la corrélation entre les deux séquences alignées est inférieure au seuil prédéfini  $\zeta$ , une erreur est commise par la boucle d'acquisition en ne déclarant pas l'acquisition.

### 2.2.1 Architecture de la boucle d'acquisition

Le principe des boucles d'acquisition est basé sur la recherche de la séquence transmise par l'émetteur, sur l'ensemble des décalages possible de la séquence PN. Cette recherche est divisée en  $q$  cellules, chacune d'elle correspond à une période d'intégration,  $q$  peut être égal à la longueur de la séquence PN ou un multiple d'elle, par exemple si le pas de décalage est de  $\frac{T_c}{2}$ ,  $q$  est égal au double de la longueur de la séquence PN. Le processus d'acquisition est basé sur la recherche de la cellule qui correspond à la situation où les deux codes sont synchronisés, le temps de recherche pour chaque cellule correspond au temps d'intégration nécessaire pour décider si oui ou non les deux séquences sont synchronisées. Il existe plusieurs types de boucle d'acquisition pour les séquences PN, les plus utilisées sont les boucles à base de corrélateur et les boucles à base de filtre adapté qui a la même structure qu'un filtre FIR.

#### 2.2.1.1 Corrélateur sériel

À la réception, la sortie du corrélateur sériel est de la forme :

$$R(\delta) = \frac{1}{T_b} \int_0^{T_b} c(t + \delta)r(t)dt \quad (2.5)$$

Avec :  $c(t + \delta)$ , la séquence PN du récepteur avec un décalage  $\delta$ ,  $r(t)$  est le signal reçu qui contient la séquence PN de l'émetteur et un bruit additif.

Le principe d'un corrélateur sériel est basé sur la recherche de la séquence PN de l'émetteur avec un pas de  $\frac{T_c}{2}$ , la corrélation est effectuée sur toute la durée de la séquence PN ( $LT_c$ ) ( $L$  chips). Après chaque intervalle d'intégration, la sortie du corrélateur est comparée à un seuil pour déterminer si les deux séquences sont synchronisées. Si le seuil n'est pas dépassé, la séquence PN du récepteur est avancée de  $\frac{T_c}{2}$  et le processus de corrélation est répété. Cette opération est effectuée jusqu'à la détection du signal [4]. La figure (2.1) donne le schéma bloc d'une boucle d'acquisition basée sur un corrélateur sériel.

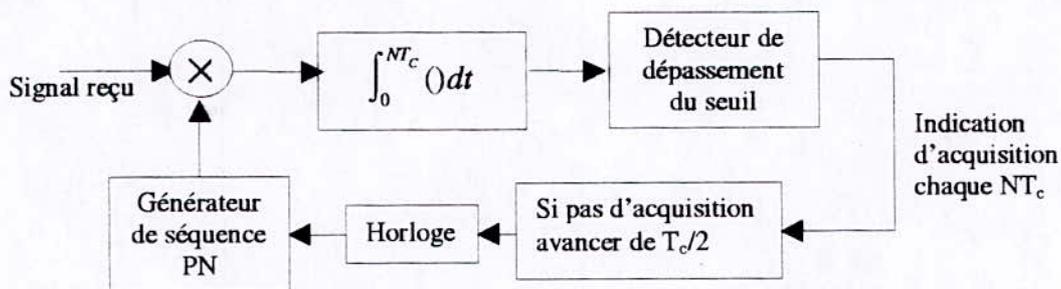


Figure 2.1: boucle d'acquisition basée sur un corrélateur sériel

Dans le cas le plus défavorable où le décalage entre les deux séquences est de  $LT_c$ , le temps minimum d'acquisition  $T_{acq}$  est donné par la relation :

$$T_{acq} = \left( \frac{LT_c}{T_c/2} LT_c \right) = 2L^2 T_c \quad (2.6)$$

Donc pour effectuer l'acquisition, l'émetteur doit transmettre la séquence d'acquisition durant une durée égale à au moins  $2L^2 T_c$  pour que le récepteur ait le temps suffisant pour faire la recherche nécessaire. Cette méthode devient très longue pour des valeurs de  $L$  très grandes.

Afin de réduire le temps d'acquisition, on peut utiliser plusieurs corrélateurs placés en parallèles [5], la figure (2.2) donne un exemple de cette méthode. Ici, 3 corrélateurs en parallèle sont utilisés avec des séquences PN décalées de  $\frac{T_c}{2}$ . Après une période d'intégration, si aucune des valeurs de sorties des corrélateurs ne dépassent le seuil de détection, la

séquence est décalée de  $\frac{3T_c}{2}$ , si le seuil est dépassé, la séquence avec la valeur de corrélation la plus importante est choisie.

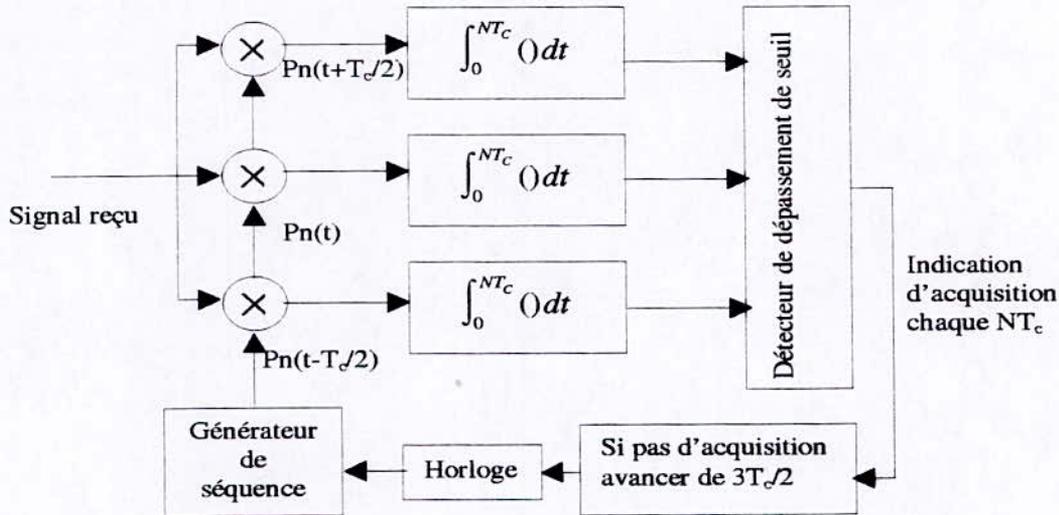


Figure 2.2: boucle d'acquisition basée sur un corrélateur sériel \ parallèle

Dans ce cas, le temps minimum de transmission pour l'acquisition se réduit à :

$$T_{acq} = \frac{LT_c}{3T_c/2} = 2/3L^2T_c \quad (2.7)$$

Le temps de recherche est réduit mais l'implémentation devient plus compliquée et consomme plus de ressources.

### 2.2.1.2 Filtre adapté

Dans une architecture utilisant un filtre adapté [6], le signal reçu  $r(t)$  est passé à travers un filtre FIR avec une réponse impulsionnelle  $c(L - k)$ , qui est l'inverse dans le temps de la séquence PN  $c(k)$ . Si la sortie du filtre est échantillonnée à  $T_b$  le résultat est exactement équivalent à un corrélateur sériel. Cependant, avec une période d'échantillonnage de  $\frac{T_c}{2}$ , à chaque nouvel échantillon à l'entrée du filtre une nouvelle valeur de la fonction de corrélation est obtenue avec un nouvel offset. Ce qui permet d'avoir un temps d'acquisition plus court, puisqu'on détecte directement la séquence  $c(k)$ . Il suffit par la

suite d'appliquer le décalage adéquat au générateur local pour aligner les deux séquences. Cependant, l'implémentation du filtre nécessite plus de ressources. Celles-ci augmentent considérablement avec la longueur de la séquence PN et le facteur de sur-échantillonnage ( $s$ ) à l'entrée du filtre. Ce qui nécessite une réponse impulsionnelle de longueur ( $s$ ) fois la longueur de la séquence PN. La figure (2.3) donne l'architecture utilisant un filtre adapté.

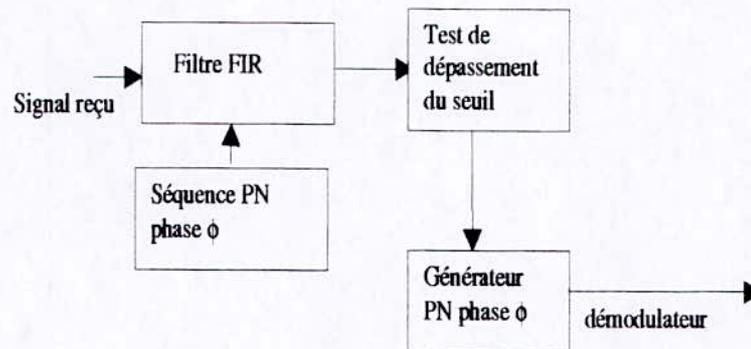


Figure 2.3: boucle d'acquisition basée sur un filtre adapté

Pour résumer, les boucles d'acquisition sérielles sont utilisées pour de longues séquences PN, par contre, les boucles à base de filtre adapté sont utilisées pour des séquences PN courtes.

### 2.2.2 Performance d'une boucle d'acquisition

La boucle d'acquisition utilise le temps de convergence ou le temps d'acquisition "en général le temps moyen d'acquisition", comme une mesure standard de performance. Le rapport signal sur bruit, la puissance des interférences et la valeur du seuil de détection sont des paramètres qui influent sur le temps moyen d'acquisition par le fait de leurs influences sur les probabilités de détection et de fausse alarme. Pour déterminer le temps moyen d'acquisition, plusieurs variables sont requises ; la probabilité de fausse alarme, la longueur du code, pénalité de fausse alarme qui est définie comme le temps requis pour détecter une fausse acquisition, probabilité de détection qui correspond à une décision

correcte sur tout le processus. En supposant que la probabilité de fausse alarme  $P_{fa}$  et la probabilité de détection  $P_d$  sont connues et en modélisant la pénalité d'obtenir une fausse alarme par  $k\tau_D$ , où  $\tau_D$  est le temps d'intégration, qui dans le cas d'un corrélateur sériel est égale à la période de la séquence PN. Le temps moyen d'acquisition est donné par la relation [7] :

$$\bar{T} = \frac{(2 - P_D) - (1 + kP_{fa})}{2P_D} (q\tau_D) \quad (2.8)$$

Dans le cas d'une boucle d'acquisition utilisant un filtre adapté, l'expression du temps moyen de détection est exactement la même, la seule différence est que  $\tau_D = T_c$ , (la période chip).

Le seuil de détection détermine les performances de la boucle d'acquisition. Le seuil optimum est défini comme étant le seuil qui permet des temps d'acquisitions relativement petits dans une grande gamme de rapport signal sur bruit.

La figure (2.4) donne les variations du temps moyen d'acquisition en fonction du rapport chip sur bruit (CNR) pour une séquence PN de longueur  $L = 31$ . Les simulations sont effectuées sur  $10^4$  bits, avec un offset initial aléatoire. Aussi nous avons supposé que le décalage entre les deux séquences est un nombre entier de chip  $\delta_{est} \in [0, 1, 2, \dots, L]$ .

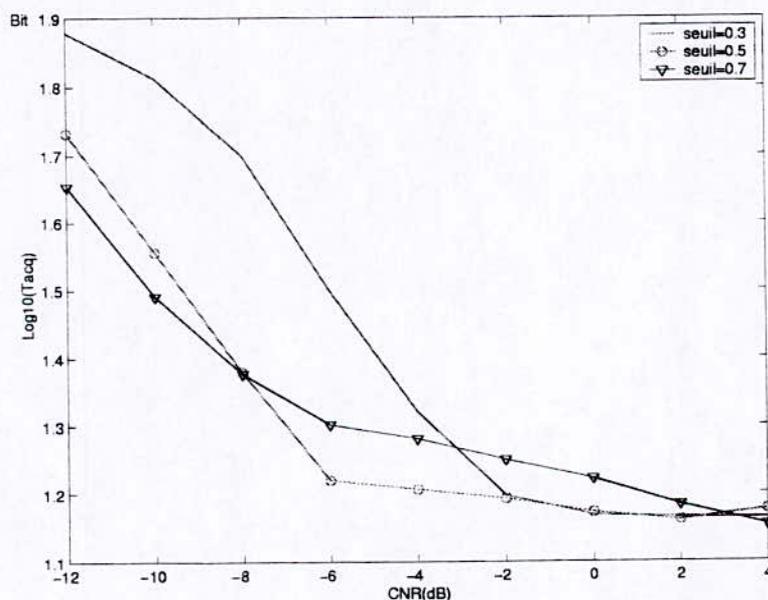


Figure 2.4: temps moyen d'acquisition pour une boucle d'acquisition sérielle

Comme le montre la figure (2.4), pour des valeurs de CNR très bas il est plus souhaitable d'utiliser des seuils hauts, parce que le nombre de fausse alarme est réduit. Par contre pour d'important CNR, les seuils hauts tendent à ne pas détecter le signal et donnent des résultats moindres, par rapport à ceux obtenus avec des seuils plus bas. La raison pour laquelle les seuils hauts mettent plus de temps à détecter le signal est que dans le cas de grand rapport CNR, la variance du signal ne varie pas beaucoup, donc il y a moins de probabilité pour que le résultat de la corrélation dépasse le seuil prédéfini. D'après la dernière figure, le seuil optimal est égal à 0.5.

La figure (2.5) donne les probabilités d'erreurs d'acquisition en fonction du CNR pour différents seuils, avec la même séquence PN. Comme on peut le remarquer, plus le seuil est grand plus la probabilité d'erreur diminue, ce qui est prévisible. Cependant, les meilleurs résultats obtenus avec des seuils hauts se traduisent par une augmentation du temps moyen d'acquisition.

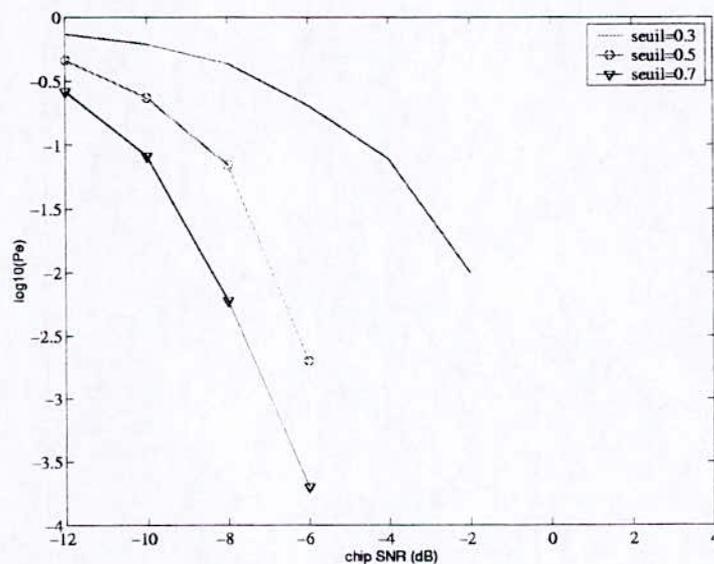


Figure 2.5: probabilité d'erreur d'acquisition pour un corrélateur sériel

Une fois la séquence PN détectée, un algorithme de vérification doit être exécuté pour prévenir les fausses détections. Il est nécessaire d'intégrer durant un certain temps pour tester l'acquisition.

Après le mode de vérification, la boucle d'acquisition est terminée et le récepteur commute sur la boucle de tracking.

## 2.3 Tracking

La boucle de tracking maintient le générateur de code PN en synchronisme avec la séquence reçue. Il inclue la synchronisation chip et, pour une démodulation cohérente, la synchronisation de la phase de la porteuse.

La boucle de tracking la plus utilisée dans un système DSSS est une *early-late gate bit tracking loop* [1] qui est représentée dans la figure (2.6).

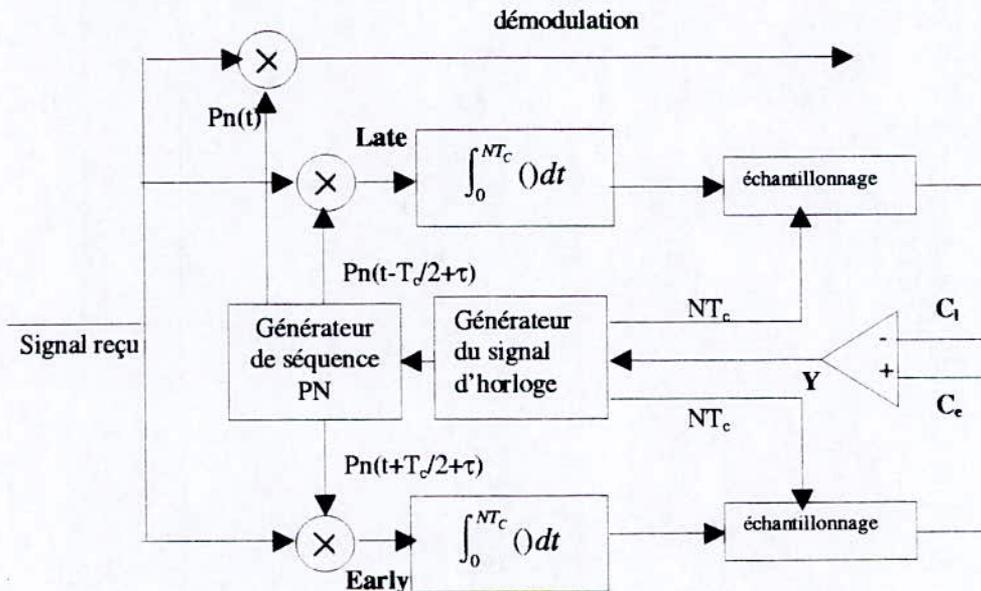


Figure 2.6: early late bit tracking loop

Dans cette boucle, le signal reçu qui a un décalage maximum par rapport à la séquence du récepteur de  $\tau < \frac{T_c}{2}$  est multiplié par des versions décalées de la séquence PN du générateur local :  $p_e(t) = c(t + \frac{T_c}{2} + \tau)$  et  $p_l(t) = c(t - \frac{T_c}{2} + \tau)$ , chaque séquence est une version avancée ou retardée de la séquence PN.

Le résultat de la multiplication passe par un intégrateur pour évaluer la fonction de cor-

relation à deux points différents :  $c_l = R(\tau - \frac{T_c}{2})$  et  $c_e = R(\tau + \frac{T_c}{2})$ , telle que :

$$c_l = R(\tau - \frac{T_c}{2}) = \frac{1}{T_b} \int_0^{T_b} c(t)c(t - \frac{T_c}{2} + \tau)dt \quad (2.9)$$

$$c_e = R(\tau + \frac{T_c}{2}) = \frac{1}{T_b} \int_0^{T_b} c(t)c(t + \frac{T_c}{2} + \tau)dt \quad (2.10)$$

Les équations (2.10) et (2.9) représentent la fonction de corrélation dans les branches *early* et *late* respectivement. La différence entre les deux corrélations est :

$$y(t) = R(\tau - \frac{T_c}{2}) - R(\tau + \frac{T_c}{2}) \quad (2.11)$$

La boucle de tracking minimise la fonction erreur  $y(t)$  en contrôlant le générateur du signal d'horloge, si  $\tau$  est positif, le générateur du signal d'horloge augmente sa fréquence ce qui force  $\tau$  à décroître, si  $\tau$  est négatif la fréquence du générateur est diminuée ce qui fait croître  $\tau$ , la figure(2.7) présente une simulation du processus de *tracking*.

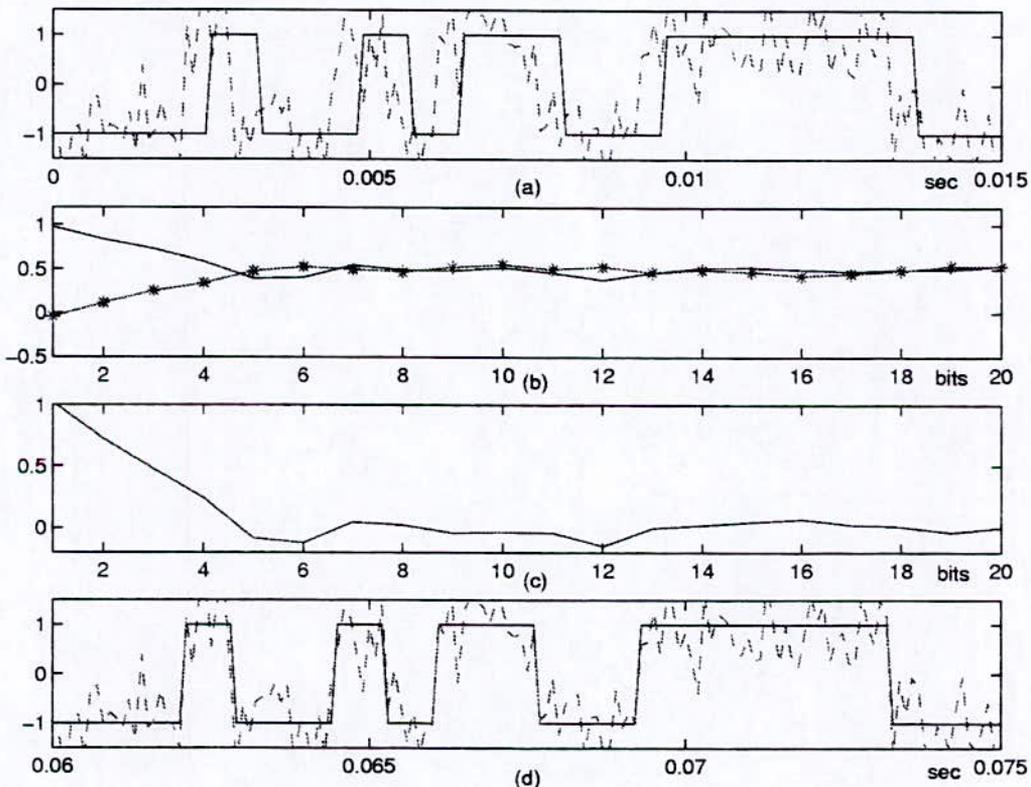


Figure 2.7: processus de tracking

Avec :

(a) : décalage initiale entre les deux séquences.

(b) :  $(*)c_e, (-)c_l$ .

(c) : fonction erreur  $y(t)$ .

(d) : synchronisation des deux séquences.

La figure (2.7)(a) montre le décalage entre les deux séquences obtenu après la phase d'acquisition et qui est égale à  $T_c/2$ . Les valeurs de la fonction de corrélation des deux branches représentées dans la figure (2.7)(b) sont différentes. A la fin de la première période, la séquence PN est décalée vers la droite ce qui permet de diminuer la fonction erreur représentée dans la figure (2.7)(c) jusqu'à ce que la fonction erreur tende vers zéro. La précision de la boucle de synchronisation dépend du nombre d'échantillons pris pour le calcul de la fonction de corrélation dans les deux branches, et de la sensibilité du générateur du signal d'horloge.

## 2.4 Conclusion

Dans ce chapitre on a étudié le processus de synchronisation dans les systèmes DSSS, dont la précision détermine les performances du système. Pour avoir un gain maximum, il faut que la synchronisation soit la plus parfaite possible. Une autre source de dégradation des performances d'un système DSSS est l'effet des interférences et du bruit.

# Chapitre 3

## Systeme DSSS modulé par une chirp

### 3.1 Introduction

Dans le chapitre 1, on a démontré qu'un système basé sur la DSSS a une capacité inhérente pour réduire l'effet des interférences. Cependant, les performances des systèmes de communications DSSS peuvent être augmentées en terme d'immunité aux interférences par l'estimation de l'interférence et sa soustraction du signal reçu avant l'opération de *de-spreading*.

Plusieurs techniques peuvent être utilisées à cet effet, parmi elles on peut citer le filtrage adaptatif [8], [9],[10], les approches dans le domaine fréquentiel [11], les approches temps-fréquence [12], les méthodes basées sur le filtrage spatial [13].

Dans ce chapitre, on propose une nouvelle approche afin d'augmenter les performances d'un système DSSS en terme d'immunité aux bruits et aux diverses interférences qui peuvent détériorer la qualité de réception.

Le système proposé est basé sur la modulation du signal d'information avec un signal chirp en plus de la séquence PN. A la réception, on utilise un filtre réjecteur à réponse impulsionnelle finie (FIR) variable dans le temps et synchronisée avec la fréquence instantanée de la chirp pour la démodulation.

### 3.2 Description du système

La figure(3.1) donne une représentation du schéma de principe du système proposé.

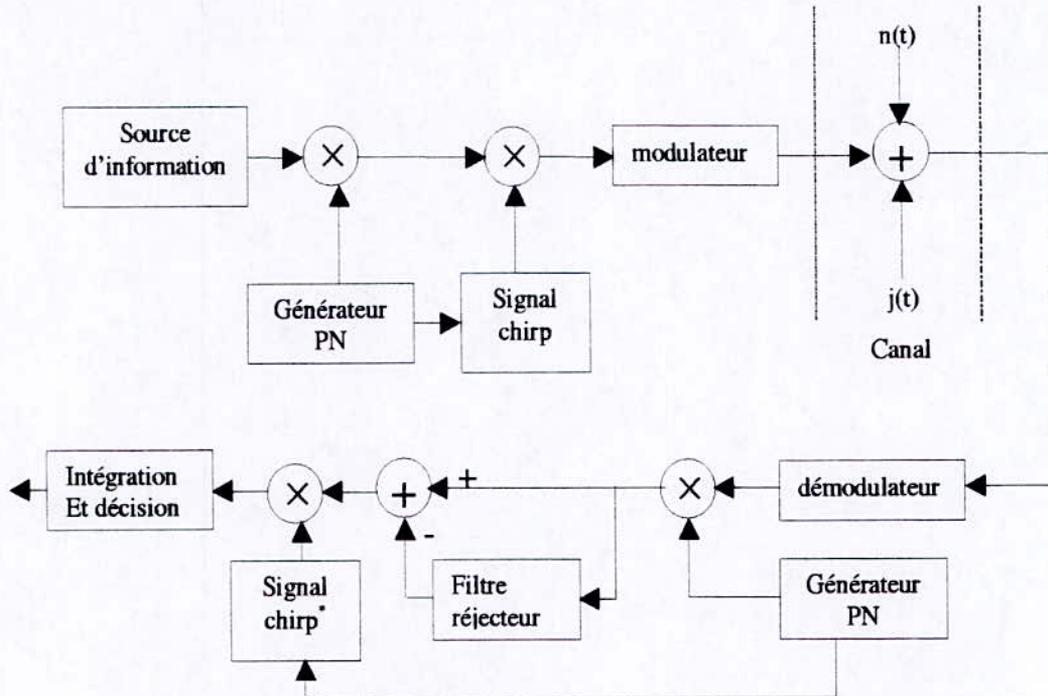


Figure 3.1: bloc diagramme du système DSSS utilisant une chirp modulation

$n(t)$  et  $j(t)$  sont respectivement le bruit du canal et l'interférence.

Le signal transmis par l'émetteur peut s'écrire sous la forme

$$s(t) = \sum_{k=-\infty}^{\infty} I_k b_k(t - kT_b) \quad (3.1)$$

Où,  $I_k$  représente la séquence d'information binaire,  $T_b$  est la durée de chaque bit.

$b_k(t)$  est le signal DSSS modulée par la chirp et est donné par :

$$b_k(t) = \sum_{i=0}^{L-1} p_k(i) q(t - iT_c) ch(t) \quad (3.2)$$

$p_k(n)$  représente la séquence de sortie du générateur de séquence pseudo-aléatoire du  $k^{ieme}$  bit d'information  $b_k(t)$ ,  $q(t)$  est une impulsion rectangulaire de durée  $T_c = \frac{T_b}{L}$  et d'énergie

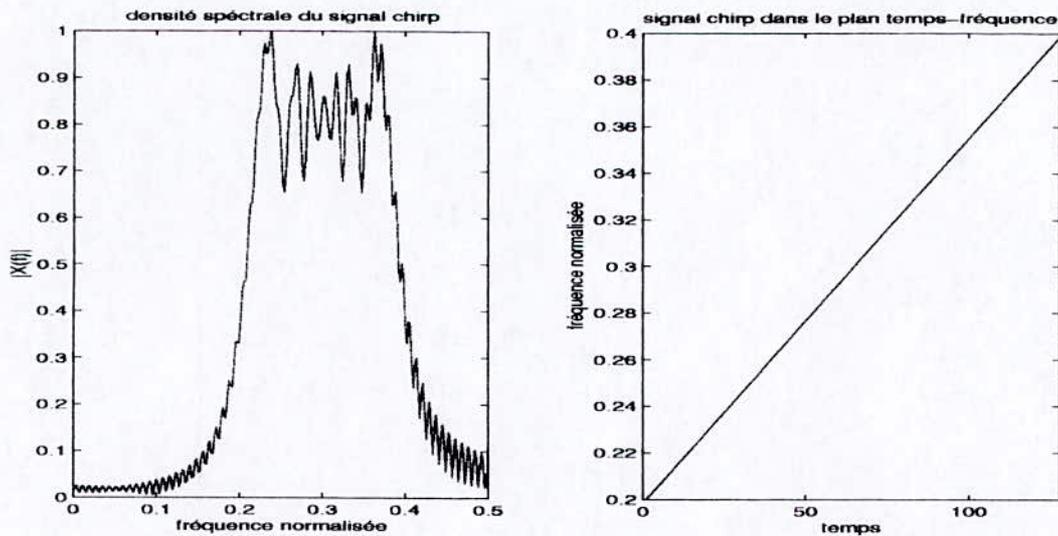


Figure 3.2: représentation du signal chirp dans les plans fréquence et temps-fréquence

unitaire ;  $L$  est la longueur de la séquence PN et  $ch(t)$  est le signal chirp périodique de période  $LT_c$  avec une variation de fréquence linéaire telle que chaque *chip* de la séquence PN a une fréquence instantanée ( $IF$ ) différente.

Le signal chirp a la forme suivant :

$$ch(t) = \exp(j(\alpha t + f_{st})t) \quad (3.3)$$

$(\alpha t + f_{st})$  est la fréquence instantanée du signal chirp  $ch(t)$ .  $f_{st}$  est la fréquence initiale de la chirp,  $\alpha$  est donné par la relation suivante :

$$\alpha = \frac{f_{end} - f_{st}}{L} \quad (3.4)$$

$f_{end}$  est la fréquence finale de la chirp qui correspond à la fréquence instantanée du signal à la fin de la période de la séquence PN. La figure (3.2) donne une représentation de la chirp dans les plans fréquence et temps-fréquence.

A la réception, le signal après l'opération de *despreading* passe par un filtre symétrique réjecteur avec une réponse impulsionnelle finie variable dans le temps, telle que la fréquence rejetée par ce filtre à l'instant  $t$  correspond à la fréquence instantanée de la chirp au même instant.

Ce filtre ne doit pas amplifier le signal à l'entrée et ne doit rejeter que la fréquence IF de la chirp.

En supposant une synchronisation parfaite entre l'émetteur et le récepteur, le signal à la sortie du filtre qui ne contient que le bruit et les différentes interférences est soustrait du signal reçu, ce qui permet dans le cas idéal de retrouver le signal chirp sans les interférences et une puissance de bruit réduite.

L'opération suivante consiste à retrouver le signal original (signal d'information) par la démodulation du signal chirp en multipliant ce dernier par son complexe conjugué, suit ensuite les opérations d'intégration et de décision sur la partie réelle du signal obtenu.

La séquence PN et le signal chirp sont connus du récepteur. On suppose que 1 est transmis, le signal reçu par le récepteur échantillonné à  $T_c$  durant un intervalle bit  $T_b$  est de la forme :

$$r(i) = p(i)ch(i) + j(i) + n(i) \quad (3.5)$$

Où  $j(i)$  représente l'interférence qui est supposée à moyenne nulle et avec une fonction de covariance  $R_j(i)$  et  $n(i)$  est un bruit blanc de moyenne nulle et de variance  $\sigma^2$ .

Suite aux propriétés de la séquence PN, de l'interférence  $j(i)$  et du bruit  $n(i)$ , les trois séquences de l'équation (3.5) sont indépendantes.

Soit un filtre réjecteur FIR d'ordre M avec une réponse impulsionnelle  $h(i)$  variable dans le temps, utilisé pour l'élimination de la chirp, la réponse impulsionnelle du filtre peut être écrite sous forme matricielle  $H$ , telle que chaque ligne correspond à la réponse du filtre à l'instant  $i$  :

$$H = \begin{pmatrix} h_0(1) & h_1(1) & h_2(1) & \dots & h_{M-1}(1) \\ h_0(2) & h_1(2) & h_2(2) & \dots & h_{M-1}(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_0(L) & h_1(L) & h_2(L) & \dots & h_{M-1}(L) \end{pmatrix}$$

l'entrée de ce filtre est le signal  $r(i)p(i)$  et sa sortie est le signal  $x(i)$  telle que :

$$x(i) = \sum_{m=0}^{M-1} h_m(i-m)r(i-m)p(i-m) \quad (3.6)$$

Comme cité précédemment, la sortie du filtre réjecteur est soustraite du signal  $r(i)p(i)$ , le résultat de cette opération est démodulé, ensuite suit l'opération d'intégration et de décision, ce qui permet d'avoir la variable de décision  $\xi$  suivante :

$$\xi = \text{Re}\left(\sum_{i=1}^L (r(i)p(i) - x(i))ch^*(i)\right) \quad (3.7)$$

### 3.3 Synchronisation

Pour permettre une bonne réjection de la chirp contenue dans le signal reçu et sa démodulation, en plus de la synchronisation des deux générateurs PN de l'émetteur et du récepteur la synchronisation des deux signaux chirp est nécessaire, pour ce faire, il suffit de contrôler le générateur de la chirp avec l'état du registre interne du générateur de séquence PN, donc le début de la chirp coïncide avec le début de la séquence PN qui détermine le début de chaque période bit.

Du fait de la relation entre l'état de la séquence PN et le signal chirp, la synchronisation des deux séquences PN implique la synchronisation des deux chirps. Etant donné que la transmission du signal se fait sur deux canaux : un pour la partie réelle et un pour la partie imaginaire, la procédure de synchronisation doit prendre en compte cette caractéristique du signal reçu. De ce fait, les algorithmes d'acquisition et de tracking diffèrent un petit peu de ceux décrits dans le chapitre précédent mais gardent le même principe. Le processus de synchronisation se fait avant l'opération de filtrage, parce que lorsque les deux séquences ne sont pas synchronisées, le signal à la sortie du filtre présente des pics qui provoquent une augmentation du taux de fausse détection et ne permettent pas l'acquisition de la séquence PN.

#### 3.3.1 Processus d'acquisition

Le principe du processus d'acquisition est identique à celui décrit dans le chapitre précédent, mais il prend en compte la partie imaginaire de la chirp. La figure(3.3) donne une représentation du bloc diagramme de tracking.

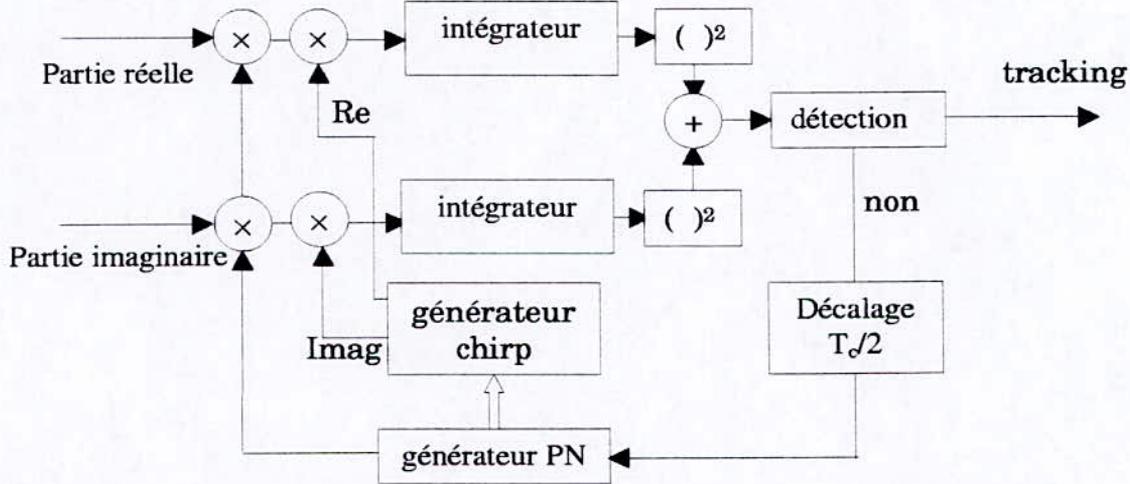


Figure 3.3: processus d'acquisition

La partie réelle du signal reçu est multipliée par la séquence PN et par la partie réelle du générateur du signal chirp du récepteur, de même, la partie imaginaire du signal reçu est multipliée par la séquence PN et par la partie imaginaire du signal chirp. Le résultat de cette opération est intégré sur chacune des deux branches durant une période bit  $T_b$ , le résultat obtenu est élevé au carré puis sommé ce qui permet d'obtenir la variable de décision  $\Gamma$  qui permet de décider si oui ou non les deux séquences sont synchronisées :

$$\Gamma = \left( \sum_0^{L-1} Re(r(i))pn(i)Re(ch(i)) \right)^2 + \left( \sum_0^{L-1} Im(r(i))pn(i)Imag(ch(i)) \right)^2 \quad (3.8)$$

Où :  $r(i)$  est le signal reçu,  $p(i)$  et  $ch(i)$  sont respectivement la séquence PN et le signal chirp.

L'élevation au carré des deux sommes permet d'augmenter le rapport entre le pic principal de la fonction  $\Gamma$  et ses lobes secondaires ce qui permet au système d'être plus robuste au bruit. La figure (3.4) donne une représentation de la variable de décision.

### 3.3.2 Tracking

Une fois la phase de tracking terminée, le système bascule sur le mode tracking ce qui permet de converger vers une synchronisation complète entre l'émetteur et le récepteur.

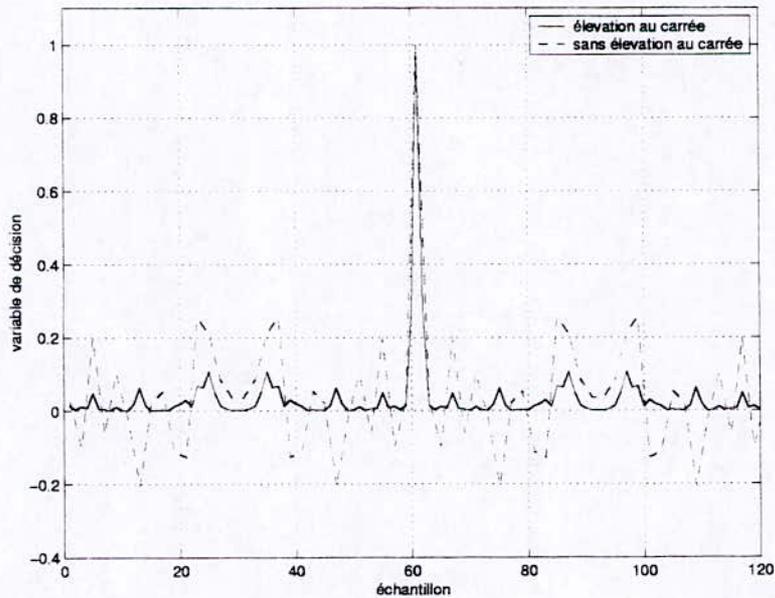


Figure 3.4: variable de décision  $\Gamma$  en fonction du nombre de bit

Le système de tracking utilisé dans notre cas est similaire à celui décrit dans le chapitre précédent en utilisant uniquement la partie réelle du signal reçu. La seule différence réside dans le fait qu'après avoir multiplié la partie réelle du signal reçu par les séquences *early* et *late* du générateur PN, le signal résultant est multiplié également avec les signaux avancé et retardé de  $T_c/2$  de la partie réelle du générateur du signal chirp avant l'opération d'intégration.

La synchronisation de la partie réelle du signal reçu implique la synchronisation de la partie imaginaire.

### 3.4 Performance du système

Les performances d'un système de communications numériques sont déterminées par la probabilité d'erreur de réception. L'objectif de tout système est de minimiser cette probabilité d'erreur qui est fonction du rapport signal sur bruit et interférence à l'entrée du récepteur. Selon [14] la probabilité d'erreur du récepteur dépend des statistiques de la

variable de décision  $\xi$  et est déterminée par la relation :

$$P_e = Q(\sqrt{SNIR}) = Q\left(\sqrt{\frac{E^2[\xi]}{\text{var}[\xi]}}\right) \quad (3.9)$$

où :

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{y^2}{2}} dy \quad (3.10)$$

## 3.5 Rapport signal sur bruit et interférence du récepteur

### 3.5.1 Système DSSS classique

Dans un système DSSS classique la variable de décision est donnée par la relation suivante :

$$\xi = \sum_{i=1}^L p(i)^2 + \sum_{i=1}^L n(i)p(i) + \sum_{i=1}^L j(i)p(i) \quad (3.11)$$

Compte tenu des propriétés du bruit et de l'interférence qui sont de moyenne nulle, la valeur moyenne de la variable de décision  $\xi$  est donnée par :

$$E[\xi] = E\left[\sum_{i=1}^L (p(i)^2 + n(i)p(i) + j(i)p(i))\right] = L \quad (3.12)$$

Le terme  $E[.]$  représente l'espérance mathématique. La variance de  $\xi$  est donnée par la relation :

$$\text{var}[\xi] = L\sigma^2 + L\rho^2 - \frac{1}{L} \sum_{i=1}^L \sum_{\substack{k=1 \\ k \neq i}}^L R_j(i-k) \quad (3.13)$$

Telle que :

$\rho^2 = R_j(0)$  est la puissance de l'interférence à l'entrée du corrélateur.

Dans le cas où la séquence PN est de moyenne nulle l'équation (3.13) se réduit à :

$$\text{var}[\xi] = L(\sigma^2 + \rho^2) \quad (3.14)$$

à partir des deux équations (3.12) et (3.14), le rapport signal sur bruit et interférence à la sortie du corrélateur  $SNIR_{ds}$  peut s'écrire sous la forme :

$$SNIR_{ds} = \frac{L}{(\sigma^2 + \rho^2)} \quad (3.15)$$

### 3.5.2 Système proposé

A partir des équations (3.6) et (3.7), la variable de décision  $\xi$  peut s'écrire sous la forme :

$$\xi = \xi_1 + \xi_2 + \xi_3 \quad (3.16)$$

avec :

$$\xi_1 = Re\left\{\sum_{i=1}^L (p(i))^2 - \sum_{m=0}^{M-1} h_m(i-m)ch(i-m)ch^*(i)\right\} \quad (3.17)$$

$$\xi_2 = Re\left\{\sum_{i=1}^L (j(i)p(i) - \sum_{m=0}^{M-1} h_m(i-m)j(i-m)p(i-m))ch^*(i)\right\} \quad (3.18)$$

$$\xi_3 = Re\left\{\sum_{i=1}^L (n(i)p(i) - \sum_{m=0}^{M-1} h_m(i-m)p(i-m)n(i-m))ch^*(i)\right\} \quad (3.19)$$

telle que, \* représente le complexe conjugué.

Les trois différents termes  $\xi_1$ ,  $\xi_2$  et  $\xi_3$  correspondent, respectivement, au signal utile, l'interférence et le bruit additif. Suite aux propriétés d'indépendances entre  $p(i)$ ,  $j(i)$  et  $n(i)$ , la valeur moyenne de la variable de décision  $\xi$  est déterminée par la relation :

$$E[\xi] = E[\xi_1] + E[\xi_2] + E[\xi_3] \quad (3.20)$$

On suppose une synchronisation parfaite entre l'émetteur et le récepteur, donc le signal à la sortie du filtre ne contient pas le signal d'information.

$$\sum_{m=0}^{M-1} h_m(i-m)ch(i-m) \approx 0 \quad (3.21)$$

Et la moyenne de la variable de décision de l'équation (3.17) se réduit à :

$$E[\xi] = E[\xi_1] = \sum_{i=1}^L E[p(i)^2] = L \quad (3.22)$$

Le second et le troisième terme de l'équation (3.20) sont nuls parce que les différents signaux sont indépendants entre eux et à moyenne nulle.

Puisque les variables  $\xi_1$ ,  $\xi_2$  et  $\xi_3$  sont décorrélées, la variance de  $\xi$  est la somme des variances des trois variables des équations (3.17), (3.18) et (3.19) :

$$\text{var}(\xi) = \text{var}(\xi_1) + \text{var}(\xi_2) + \text{var}(\xi_3) \quad (3.23)$$

avec :

$$\text{var}[\xi_1] = 0 \quad (3.24)$$

$$\begin{aligned} \text{var}[\xi_3] = & L\sigma^2 \left\{ 1 - \sum_{i=1}^L \sum_{m=0}^{M-1} h_m(i-m)ch^*(i)ch(i-m) \right. \\ & - \sum_{i=1}^L \sum_{m=0}^{M-1} h_m^*(i-m)ch(i)ch^*(i-m) \\ & \left. + \sum_{i=1}^L \sum_{m=1}^{M-1} \sum_{o=1}^{M-1} h_m(i-m)h_o^*(i-m)ch^*(i)ch(i+o-m) \right\} \quad (3.25) \end{aligned}$$

$$\begin{aligned} \text{var}[\xi_2] = & L\rho_j^2 \left\{ 1 - \sum_{i=1}^L \sum_{m=0}^{M-1} h_m(i-m)ch^*(i)ch(i-m) \right. \\ & - \sum_{i=1}^L \sum_{m=0}^{M-1} h_m^*(i-m)ch(i)ch^*(i-m) \\ & \left. + \sum_{i=1}^L \sum_{m=1}^{M-1} \sum_{o=1}^{M-1} h_m(i-m)h_o^*(i-m)ch^*(i)ch(i+o-m) \right\} \\ & - \frac{1}{L} \left\{ \sum_{i=1}^L \sum_{\substack{k=1 \\ i \neq k}}^L (R_j(i-k)ch^*(i)ch(k)) \right. \\ & + \sum_{k=1}^L \sum_{\substack{i=1 \\ k \neq (i-m)}}^L \sum_{m=0}^{M-1} h_m^*(i-m)R_j(k-i+m)ch^*(k)ch(i) \\ & \left. + \sum_{k=1}^L \sum_{\substack{i=1 \\ k \neq (i-m)}}^L \sum_{m=0}^{M-1} h_m(i-m)R_j(i-m-k)ch^*(i)ch(k) \right\} \end{aligned}$$

$$\begin{aligned}
& - \sum_{k=1}^L \sum_{i=1}^L \sum_{m=0}^{M-1} \sum_{o=0}^{M-1} \\
& \quad k \neq i - m + o \\
& h_m(i - m)h_o^*(k - o)R_j(i - m - k + o)ch^*(i)ch(k) \} \quad (3.26)
\end{aligned}$$

avec :  $\rho_j^2$  est la puissance de l'interférence.

Le rapport signal sur bruit et interférence (SNIR) est :

$$SNIR = \frac{L^2}{var[\xi]} \quad (3.27)$$

Dans les équations précédentes, on a considéré le cas où :

$$\sum_{i=1}^L p(i)p(i+k) = \begin{cases} L & \text{pour } k = i \\ -1 & \text{pour } k \neq i \end{cases} \quad (3.28)$$

Si on suppose que la séquence PN est à moyenne nulle (cas idéal), l'équation (3.28) devient :

$$\sum_{i=1}^L p(i)p(i+k) = \begin{cases} L & \text{pour } k = i \\ 0 & \text{pour } k \neq i \end{cases} \quad (3.29)$$

Et l'expression (3.26) devient :

$$\begin{aligned}
var[\xi_2] & = L\rho_j^2 \left\{ 1 - \sum_{i=1}^L \sum_{m=0}^{M-1} h_m(i - m)ch^*(i)ch(i - m) \right. \\
& \quad - \sum_{i=1}^L \sum_{m=0}^{M-1} h_m^*(i - m)ch(i)ch^*(i - m) + \sum_{i=1}^L \sum_{m=1}^{M-1} \sum_{o=1}^{M-1} \\
& \quad \left. h_m(i - m)h_o^*(i - m)ch^*(i)ch(i + o - m) \right\} \quad (3.30)
\end{aligned}$$

Dans ce cas, le rapport signal sur bruit et interférence (SNIR) prend la forme suivante :

$$SNIR = \frac{L}{(\sigma^2 + \rho_j^2)\gamma} \quad (3.31)$$

avec :

$$\begin{aligned}
\gamma & = 1 - \sum_{i=1}^L \sum_{m=0}^{M-1} h_m(i - m)ch^*(i)ch(i - m) \\
& \quad - \sum_{i=1}^L \sum_{m=0}^{M-1} h_m^*(i - m)ch(i)ch^*(i - m) + \sum_{i=1}^L \sum_{m=1}^{M-1} \sum_{o=1}^{M-1} \\
& \quad h_m(i - m)h_o^*(i - m)ch^*(i)ch(i + o - m) \quad (3.32)
\end{aligned}$$

ce qui permet de déterminer les performances du système en terme de probabilité d'erreur de réception suivant la relation :

$$P_e = Q(\sqrt{SNIR}) = Q\left(\sqrt{\frac{L}{(\sigma^2 + \rho_j^2)\gamma}}\right)$$

### 3.6 Rendement spectral

Le rendement spectral "*Spectral efficiency*", est la mesure du débit en bit de l'émetteur obtenue dans une bande passante donnée suivant la relation [15] :

$$v = \frac{\text{débit}}{\text{bande passante en Hz}} \quad (3.33)$$

Le rendement spectrale est mesuré en bit/sec.Hz.

Pour faire une comparaison efficace entre deux systèmes, il faut prendre en considération le rendement spectral de chaque système.

Comparé à un système DSSS classique, le système proposé utilise une bande passante plus grande que celle d'un système DSSS classique du fait de la modulation chirp. Donc, pour pouvoir faire les comparaisons des performances entre les deux systèmes, il faut utiliser une séquence PN dans un système DSSS classique plus grande par rapport à la séquence PN du système proposé, pour obtenir le même rendement spectral.

### 3.7 Evaluation des performances

L'évaluation des performances dans cette section a deux buts. Le premier est d'évaluer l'effet de la chirp dans un système DSSS en comparant le système proposé à un système DSSS classique. Le second est de valider les performances théoriques du système calculées dans la section précédente.

On considère le système représenté dans la figure (3.1) en utilisant un signal chirp dont la fréquence instantanée  $f_n$  varie linéairement de 0 à 0.2 (normalisée par rapport à la fréquence d'échantillonnage).

Dans les simulations, on a utilisé une séquence PN de 63 chips pour étaler le spectre du signal d'information modulé avec le signal chirp, le signal résultant est transmis à travers un canal avec un bruit blanc additif.

A la réception, on applique un filtre réjecteur avec une réponse impulsionnelle d'ordre 3 :  $h_n(1) = h_n(3) = .25$ ,  $h_n(2) = -0.5 \cos(2\pi f_n)$  [16], ce filtre permet de rejeter la fréquence  $f_n$ . La figure(3.5) donne la réponse impulsionnelle du filtre.

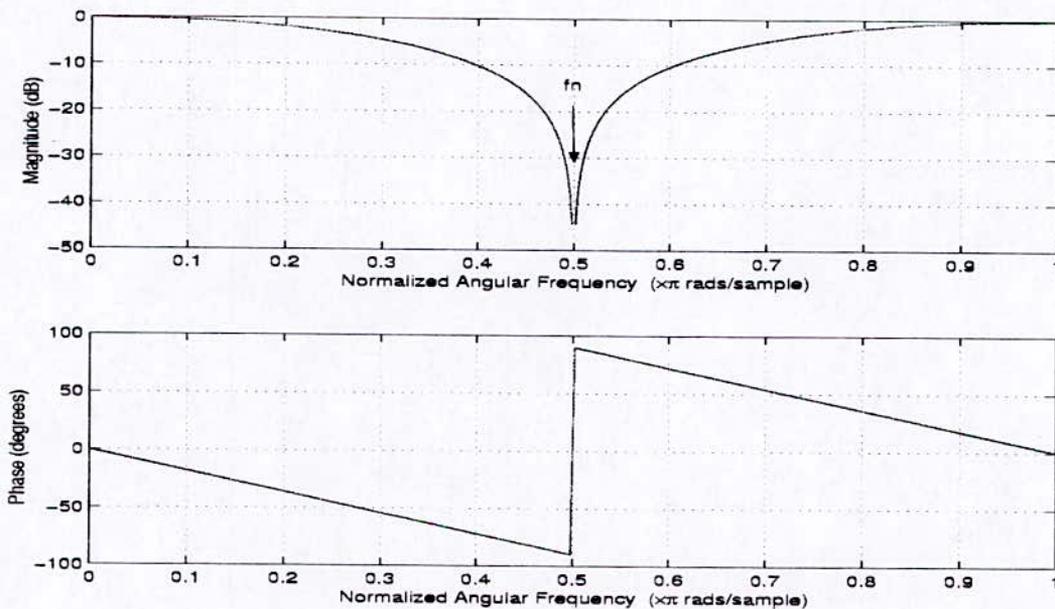


Figure 3.5: réponse impulsionnelle du filtre utilisé

Dans les simulation, on a utilisé différents types d'interférences, afin de tester le système sous différentes conditions. La première simulation consiste à voir la probabilité d'erreur du système dans le cas d'un bruit blanc additif avec un SNR variant entre  $-4$  et  $8$  dB avec un pas de  $2$  dB. La figure (3.6) représente la probabilité d'erreur du système proposé comparée à un système DSSS classique avec une séquence PN de 127 chips.

On notera que les performances du système proposé sont supérieures par rapport à un système classique avec une amélioration de près de  $3$  dB. On remarquera également que la probabilité d'erreur obtenue par simulation est proche de la probabilité d'erreur calculée analytiquement.

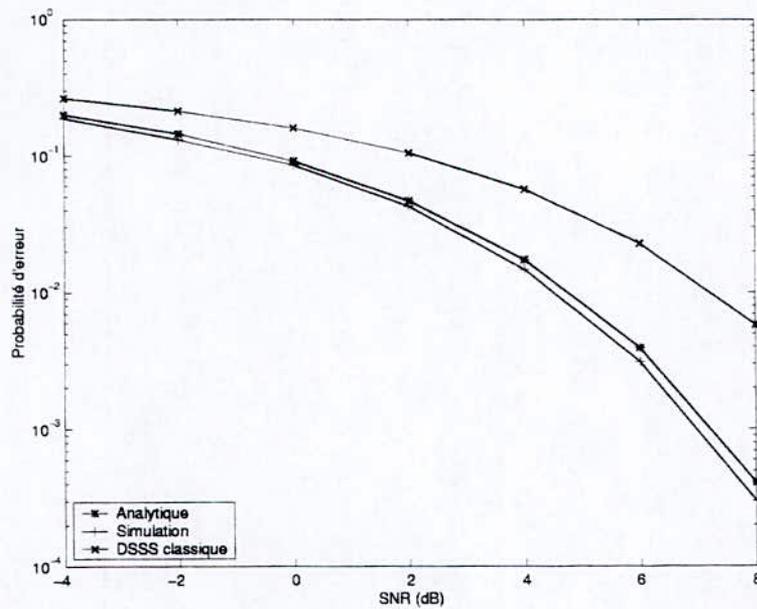


Figure 3.6: probabilité d'erreur du système proposé

Pour la deuxième simulation, on applique au signal d'information une interférence sinusoïdale avec une fréquence  $f = 0.33$  (en fréquence normalisée) avec un rapport signal sur interférence de  $-20dB$  et une phase  $\phi$  qui varie dans l'intervalle  $[0, 2\pi]$  suivant une distribution uniforme.

Les simulations sont effectuées sur  $10^6$  bits. Un bruit blanc est ajouté avec un rapport signal sur bruit (SNR) variant entre  $-4$  et  $8$  dB avec un pas de  $2dB$ , la comparaison a été faite avec un système DSSS classique ayant une séquence PN de longueur 127, pour avoir le même rendement spectral. La figure (3.7) représente les résultats des simulations et le résultat obtenu analytiquement dans le cas d'une interférence sinusoïdale.

Pour voir l'effet de l'interférence sur le système proposé la figure (3.8) montre la probabilité d'erreur de réception dans le cas d'un rapport signal sur bruit de  $8$  dB avec un rapport signal sur interférence (SIR) variant entre  $-45$  et  $-20$  dB, avec un pas de  $5$  dB.

La comparaison entre les deux systèmes montre que les performances du système proposé dans le cas où il y a une interférence, sont meilleures que celle obtenue avec un système DSSS classique même dans le cas où ce dernier est testé sur un canal sans interférence.

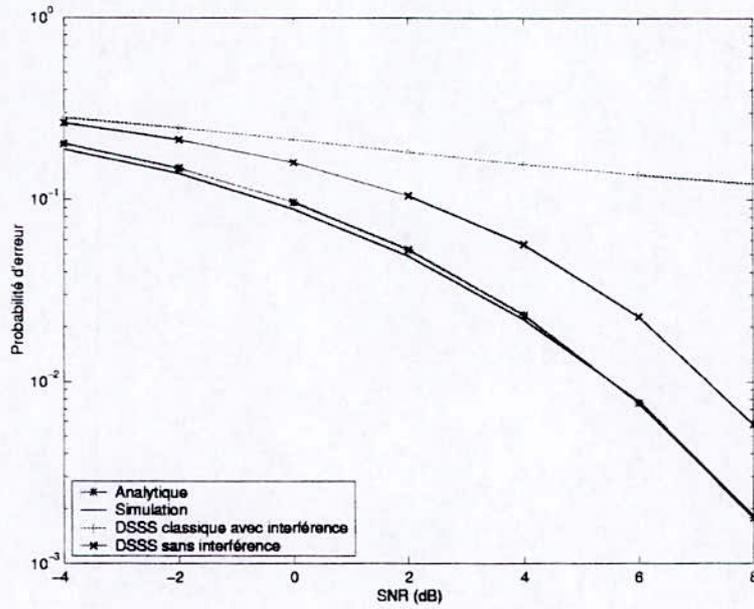


Figure 3.7: probabilité d'erreur(interférence sinusoïdale  $SIR = -20dB, w = 2.07$  rad)

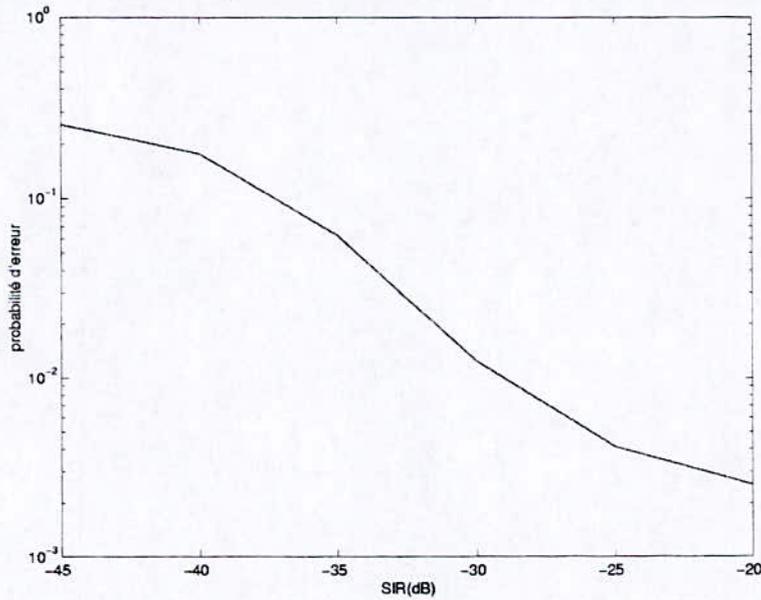


Figure 3.8: probabilité d'erreur (interférence sinusoïdale  $w = 2.07rad, SNR = 8dB$ )

Dans la troisième simulation, on considère une interférence de type chirp dont la fréquence varie linéairement entre 0.1 et 0.5 (en fréquence normalisée) et on l'applique au même système décrit dans les précédentes simulations sur chaque intervalle bit avec une phase  $\phi$  suivant une distribution uniforme dans l'intervalle  $[0, 2\pi]$ , avec un SIR de  $-20\text{dB}$ . La figure(3.9) donne la probabilité d'erreur en fonction du rapport signal sur bruit.

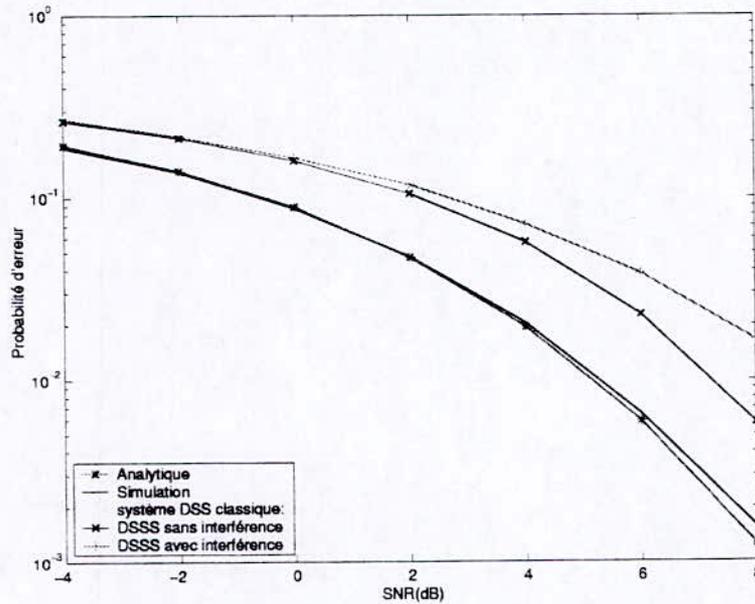


Figure 3.9: probabilité d'erreur(chirp interférence  $w = 0.62$  à  $\pi$  rad,  $JSR = -20\text{dB}$ )

La dernière simulation consiste à appliquer au système une interférence de la forme :

$$j(n) = A_j \exp(j \frac{2\pi}{3} n - \frac{(n - \alpha)^2}{\beta}) \tag{3.34}$$

C'est une interférence qui a une localisation en temps et en fréquence. La figure (3.10) donne une représentation de cette interférence. Dans les simulations, la position du pic de l'interférence  $\alpha$  varie suivant une distribution normale entre  $[0 \dots 62]$ . La figure(3.11) donne la probabilité d'erreur en fonction du rapport signal sur bruit qui varie entre  $-4$  dB et  $8$  dB avec un pas de  $2$  dB. Le pic de l'interférence est de  $20$  dB par rapport au signal.

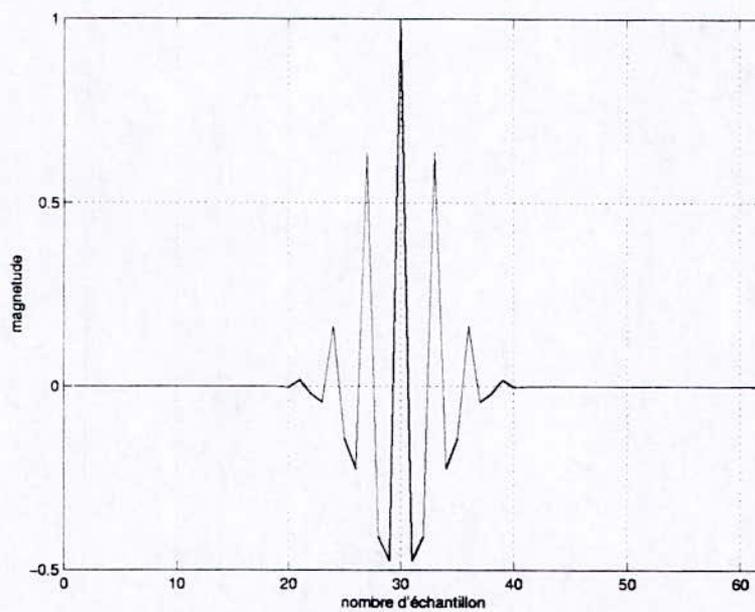


Figure 3.10: interférence  $j(n)$  avec  $\alpha = 30, \beta = 20$

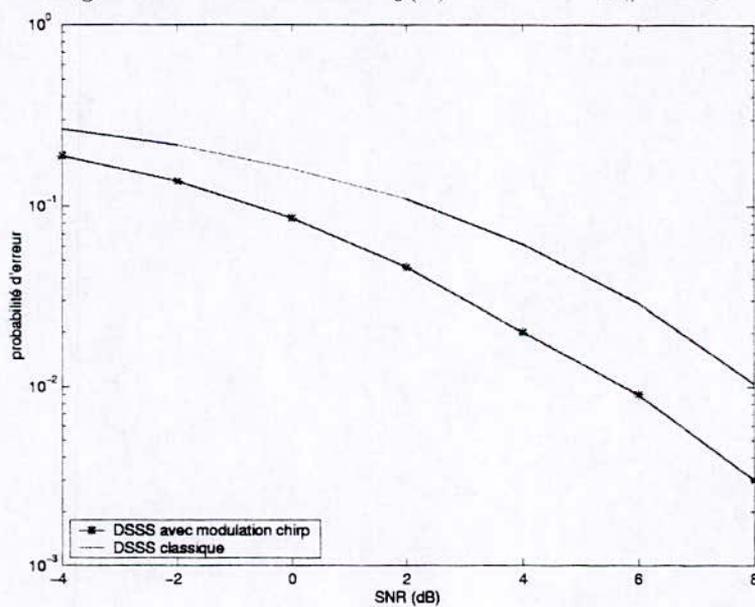


Figure 3.11: probabilité d'erreur  $\beta = 20, SIR = -20dB$

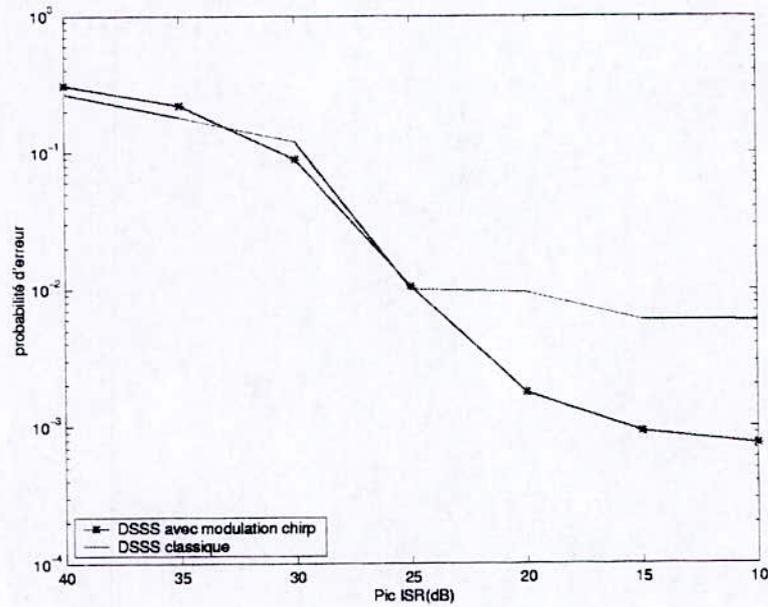


Figure 3.12: probabilité d'erreur  $\beta = 20$ ,  $SNR = 8dB$

La figure(3.12), représente la probabilité d'erreur de réception dans le cas d'un SNR de 8 dB et en faisant varier le rapport signal sur le pic de l'interférence entre -10 dB et -40 dB avec un pas de 5 dB. Dans les deux simulations  $\beta = 20$ .

Toutes les simulations indiquent que les performances du système proposé dépassent celles d'un système DSSS classique, même dans le cas où il n'y a pas d'interférences. Les résultats obtenus confirment les performances théoriques présentées dans la section précédente. l'hypothèse de la sommation gaussienne de l'interférence et du bruit semble raisonnable pour les puissances du bruit simulées.

L'utilisation d'une chirp modulation, permet au système d'avoir une capacité inhérente de réduire l'effet des différentes interférences et même dans le cas d'un bruit blanc.

### 3.8 Conclusion

Dans ce chapitre, on a décrit une nouvelle approche basée sur l'application d'une chirp modulation qui permet de donner à un système DSSS une capacité inhérente de réduire l'effet des interférences et du bruit sans l'utilisation d'un système de réjection d'interférence avant le despreading. Le rapport signal sur bruit et interférence à la sortie du corrélateur a été calculée théoriquement et validé par les simulations avec différent type d'interférences.

# Chapitre 4

## Les FPGAs et l'arithmétique binaire

Le développement des circuits logiques programmables tel que les FPGAs et les langages de programmations qui leurs sont associés permettent l'intégration de tout type de systèmes numériques. Dans ce chapitre, nous allons présenter l'architecture des FPGAs, on présentera l'arithmétique binaire que nous allons utiliser dans notre système, et on donnera la méthodologie d'implémentation sur ce type de circuit.

### 4.1 Field Programmable Gate Arrays (FPGA)

Les logiciels de développements tels que *Xilinx Fondation* contiennent plusieurs catégories de FPGA pour permettre plusieurs configuration d'implémentation sans modification du programme d'implémentation. Suivant le nombre de ressources que l'application nécessite et la vitesse à laquelle on veut qu'elle fonctionne, différents packages FPGA de différentes tailles avec des vitesses de fonctionnement différentes peuvent être utilisées, pour donner au programmeur la possibilité d'augmenter ou de réduire le nombre de ressource, afin d'avoir le meilleur compromis possible entre la taille et la vitesse d'utilisation.

L'architecture FPGA consiste en un ensemble de colonne et de ligne de bloc logique configurable (CLBs) qui réalise les fonctions logiques spécifiées par l'utilisateur. Ces derniers composent une matrice noyau qui est entourée de cellules d'entrée sortie (IOB) qui constituent une interface entre les blocs logiques et les pins du FPGA. Pour permettre le

transport des signaux entre les différents CLB, des ressources d'interconnexions sont utilisées pour réaliser les liens entre les différents blocs logiques afin de réaliser les opérations effectuées par le circuit. La figure (4.1) donne une représentation de cette architecture.

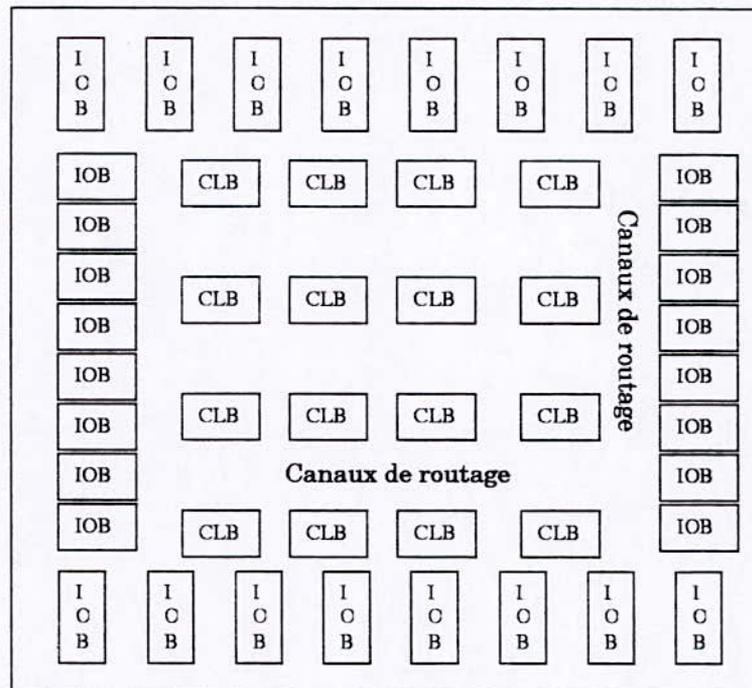


Figure 4.1: distribution des CLB et des IOBs dans un FPGA

## 4.2 Programmation en VHDL

Le VHDL "*VHSIC (Very High Speed Integrate Circuit) Hardware Description Langage*" est un langage de haut niveau pour la description matérielle. C'est à dire des circuits allant d'une simple porte logique jusqu'à un processeur, voire tout un système à base d'un ou de plusieurs microprocesseurs, il possède trois niveaux de description qui sont :

- Niveaux de description comportemental .
- Register Transfert Level (RTL).
- Niveaux de description structurelle.

### 4.2.1 Niveaux de description comportemental

Dans ce niveau, le code en VHDL est écrit comme une description de l'algorithme de l'opération qu'effectue le circuit. C'est une description comportementale qui décrit le fonctionnement du circuit sans se soucier d'un éventuel découpage proche de la réalisation. Un exemple de type de description est donné ci dessous.

```
for k in 0 to 2 loop
  C(k) ← A(k) + B(k) ;
end loop;
```

Dans cet exemple, le code en VHDL ne permet pas de connaître le nombre d'additionneur ou de cycle d'horloge nécessaire à l'opération. Le seconde niveaux de description en VHDL est le niveau RTL

### 4.2.2 Le niveau RTL

Dans cette description, le code en VHDL est une description de l'architecture du circuit, cette description utilise fondamentalement des registres, des portes logiques, des multiplexeurs, des mémoires, des additionneurs, des multiplicateurs.

Le processus de conception est ramené à une opération de transfert de donnée, par exemple la description RTL de l'exemple précédent est comme suit :

```
wait clk'event and clk='1';
  C(k) ← A(k) + B(k);
  k = k + 1;
```

Ce code RTL, définit une architecture qui nécessite trois cycles d'horloge avec un additionneur et quelques logiques de contrôle pour compléter les opérations.

### 4.2.3 Description structurelle

Une description structurelle d'un modèle est défini par sa structuration. Elle est vue comme un ensemble d'éléments interconnectés et défini surtout la connexion entre ses

éléments. Les modèles ou les descriptions de ces éléments doivent être définis. C'est un bas niveau de description qui nous place près de la réalisation physique d'un circuit ou d'un système.

Après avoir décrit, testé et optimisé les différents composants entrant dans la réalisation d'un circuit donné, il suffit de décrire la connexion entre les différents éléments.

L'exemple ci dessous donne une représentation de cette description.

U1 : nand2 port map ( $in1 \Rightarrow A, in2 \Rightarrow B, out \Rightarrow C$ ) ;

U2 : add port map ( $in1 \Rightarrow C, in2 \Rightarrow D, out \Rightarrow D$ ) ;

Les composants *nand2* et *add* sont des circuits définis, testés et optimisés. La définition de chaque composant permet d'avoir un meilleur contrôle du processus de réalisation. C'est pour cette raison que pour la réalisation de notre système nous avons opté pour une description structurelle combinée à une description RTL pour la description du fonctionnement de chaque bloc.

### 4.3 Format des données

Pour la représentation des données de notre système nous avons choisi un format à virgule fixe. Le codage des nombres fractionnaires sur N bits utilise le même concept que pour la représentation des entiers signés. Il suffit, pour coder un nombre fractionnaire F, de décaler la virgule qui se trouve à droite du bit de poids faible de l'entier signé de  $N - 1$  position vers la gauche. Le  $N^{ième}$  bit étant le bit de signe, celui-ci n'est pas à modifier. Ce décalage à gauche de la virgule revient à décaler l'entier signé vers la droite. On obtient alors l'expression suivante qui permet de convertir une fraction binaire en une fraction décimale :

$$F = (-2b_{N-1} + 1)(2^{-1}b_{N-2} + \dots + 2^{-N+1}b_0) \quad (4.1)$$

Le nombre fractionnaire F est donc codé sur N bit avec une précision de  $2^{-N+1}$ , le bit  $b_{N-1}$  représente le bit de signe et les bits  $[b_{N-2} \dots b_0]$  représente la partie fractionnaire,

avec entre les bits  $b_{N-1}$  et  $b_{N-2}$  la virgule dont la position reste toujours la même. Dans ce type de représentation, les nombres fractionnaires qu'on peut représenter dépendent du nombre de bit  $N$ . Pour une représentation sur 8 bits, on peut représenter des valeurs variants entre  $\pm 0.9921$  avec une précision de  $2^{-7}$ , cette dernière est donc limitée ; elle sera d'autant plus petite que le nombre de bits sera élevé.

## 4.4 Multiplication en virgule fixe

Si l'addition et la soustraction ne posent aucun problème particulier pour leurs réalisations, il n'est pas de même avec la multiplication. S'il est vrai que le produit de deux nombres binaires de  $N$  bits fournit un résultat sur  $2N$ , cette règle n'est pas valable pour les nombres signés. Pour trouver le nombre total de bits, calculons le produit  $P$  du plus petit nombre entier négatif codé sur  $N$  bits avec lui même, soit  $-2^{N-1}$ . Ce nombre  $P$  positif occupe  $2(N-1)$  bits. A ce nombre binaire, il faut ajouter le bit de signe à gauche du résultat précédent. Nous obtenons donc une taille totale de  $2N-1$  bits pour coder le résultat d'une multiplication de deux nombres de  $N$  bits. Ces règles restent les mêmes avec les nombres fractionnaires en binaire.

Du point de vue matériel, la multiplication est réalisée avec des circuits logiques simples tels que des portes "non ET" et des additionneurs 1 bits représentés dans la figure (4.2). Un exemple d'un multiplicateur (câblée) de deux opérands  $x$  et  $y$  sur 4 bits avec un résultat  $P$  sur sept bits est donné dans la figure (4.3).

## 4.5 L'arrondi

L'arrondi est une opération de mise à l'échelle qui permet, dans un système numérique, de réduire le nombre de bits significatifs. Prenons l'exemple de la multiplication de deux nombres binaire sur  $N$  bits ; cette opération fournit un résultat sur  $2N$  bits, alors que le calculateur mémorise des données sur  $N$  bits. Il faut donc, avant la sauvegarde du résultat de la multiplication, limité le nombre de bits au format de la mémoire. Une

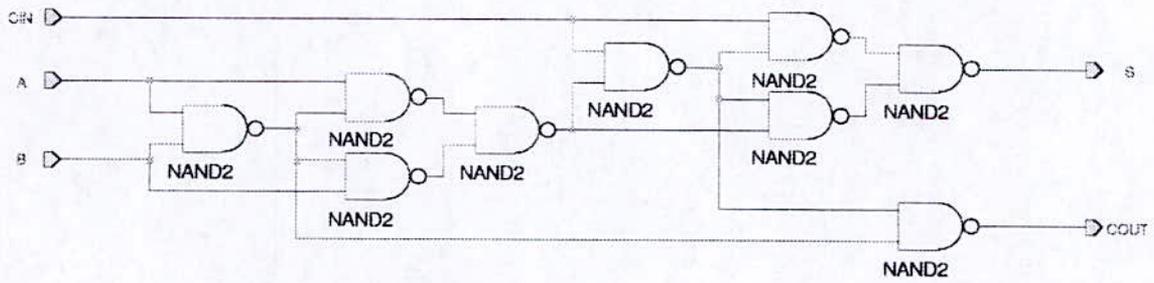


Figure 4.2: additionneur 1 bit

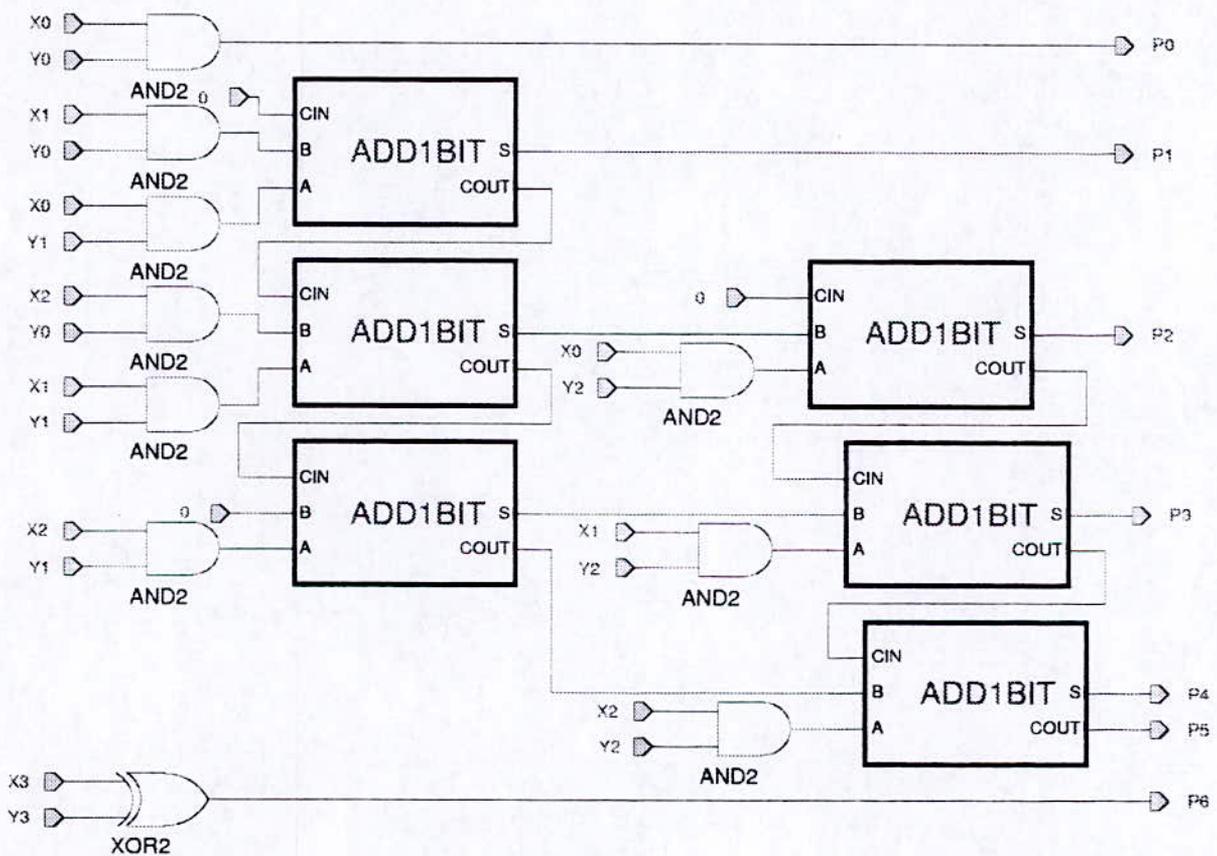


Figure 4.3: multiplication câblée 4 × 4 bits

solution simple consiste à tronquer les  $N$  bits les moins significatifs.

Mais une solution plus élégante consiste à additionner 1 ou 0 selon l'état des bits de la partie que l'on veut arrondir et des bits de la partie que l'on veut garder. Nous allons examiner les quatre configurations possibles d'arrondi qui peuvent se produire, par exemple sur un mot de 16 bits constitués de deux parties de 8 bits, contenues dans deux registres que nous appellerons  $A1$  et  $A0$ , qui correspondent respectivement aux 8 bits les plus significatifs et les huit bits les moins significatifs.

En fonction de la valeur contenue dans le registre  $A0$ , l'unité arithmétique opère un arrondi en additionnant 1 ou 0 à  $A1$ . Le premier cas d'arrondi se produit pour  $A0 < \$80$ , comme la valeur  $A0$  est inférieure à la moitié de la valeur la plus faible  $2^{-7}$  de  $A1$  soit  $2^{-8}$ , rien n'est additionné à  $A1$ , par contre dans le cas contraire  $A0 > \$80$ , un 1 est additionné à la valeur de  $A0$ . Dans le cas où la valeur de  $A0$  est égale à  $\$80$ , deux cas peuvent se présenter selon que le bit de poids faible de  $A1$  est égal à 1 ou 0. L'arrondi provoque l'addition de 1 à  $A1$  si son bit de poids faible est à 1, si non on ne fait pas d'addition. Cependant, les opérations d'arrondi et de troncature introduisent des erreurs qui dépendent de la représentation choisie.

Pour une représentation à virgule fixe "bit de signe et amplitude", l'erreur de troncature est symétrique par rapport à zéro et est donnée par :

$$-(2^{-(N-1)} - 2^{-tr}) \leq E_t \leq (2^{-(N-1)} - 2^{-tr}) \quad (4.2)$$

Où :  $E_t$  est l'erreur de troncature et  $tr$ , est le nombre de bit avant cette opération. L'erreur d'arrondi peut être représentée par la relation :

$$E_r = Q_r(x) - x \quad (4.3)$$

telle que  $Q_r(x)$  est la valeur après l'opération d'arrondi et  $E_r$  est l'erreur d'arrondi qui est indépendante du format binaire choisi et affecte uniquement l'amplitude de la valeur. Puisque l'erreur maximale dans l'opération d'arrondi ne dépasse pas la valeur  $(2^{-(N-1)} - 2^{-tr})/2$ , l'erreur d'arrondi est symétrique par rapport à zéro et est donnée par

la relation suivante :

$$-(2^{-(N-1)} - 2^{-tr})/2 \leq E_r \leq (2^{-(N-1)} - 2^{-tr})/2 \quad (4.4)$$

## 4.6 La saturation arithmétique

L'opération de saturation arithmétique constitue une solution possible au problème de dépassement dans les calculs, qui se produit lorsqu'un résultat est en dehors de la gamme arithmétique ou encore lorsqu'il nécessite un nombre plus important de bits dans sa représentation. Ainsi dans notre système, si un résultat dépasse les limites autorisées du format, sa valeur reste bloquée à la limite correspondant à son signe.

## 4.7 Méthodologie d'implémentation

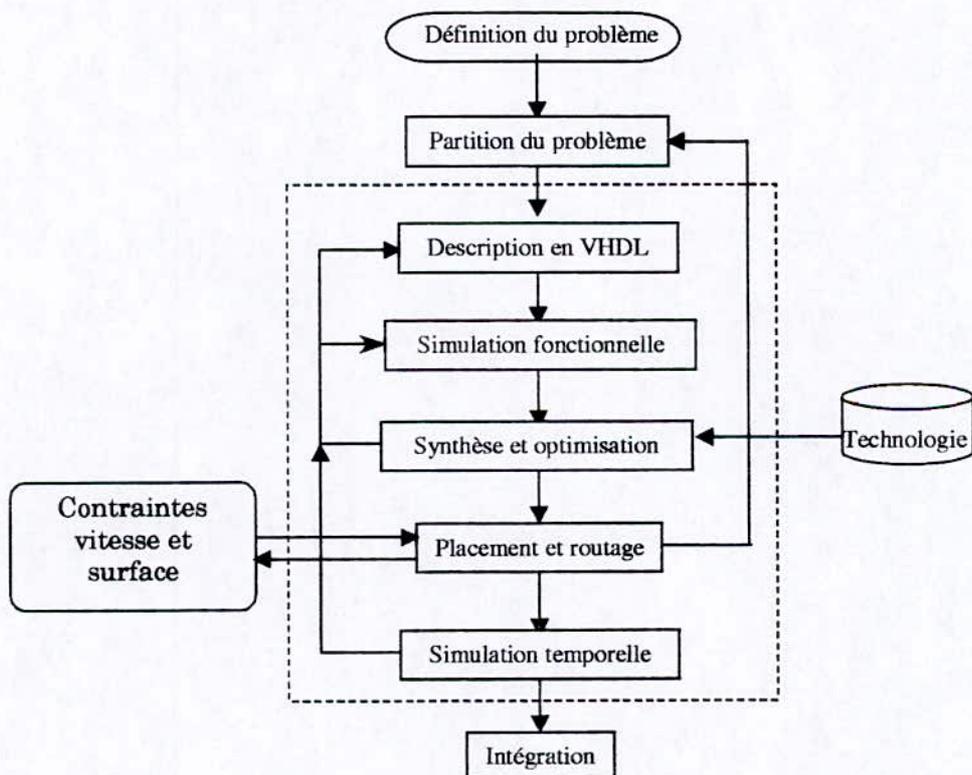


Figure 4.4: méthodologie d'implémentation

Le processus d'implémentation s'effectue à partir de langage de haut niveau tel que le VHDL, selon une méthodologie d'implémentation qui peut être résumée dans le diagramme de la figure (4.4).

Une fois le fonctionnement du circuit défini, il est préférable de le diviser en un ensemble de bloc réalisant chacun une fonction. La division du circuit en un ensemble de sous circuit, permet de mieux optimiser le système en optimisant chaque fonction indépendamment.

Une fois, la description VHDL des différents composants terminée, on passe aux simulations fonctionnelles du circuit. Ces simulations ne prennent pas en compte les retards introduits par les différents blocs logiques et les temps de stabilisation du signal. L'étape suivante est l'opération de synthèse et d'optimisation.

Cette étape, permet de rendre compatible le code VHDL avec la technologie utilisée.

La phase suivante est l'opération de placement et routage, elle permet l'implémentation des différentes fonctions dans les CLBs en prenant en compte les contraintes imposées par l'utilisateur telles que la fréquence de fonctionnement. Le programme teste plusieurs distributions possibles des blocs CLBs pour avoir le meilleur résultat possible qui satisfait ou s'approche le plus possible des contraintes imposées.

Le résultat de cette opération permet de faire les simulations temporelles qui permettent de vérifier le fonctionnement réel du circuit puisqu'elles prennent en compte tous les délais et retards imposés par l'opération de placement et de routage.

Dans ce qui suit, on se propose l'implémentation d'un système de communications à spectre étalé avec une modulation chirp. La longueur de la séquence PN est prise égale à 31, le signal chirp varie entre 0 et  $\pi$  rad/sec.

Pour la quantification des données, on utilise une représentation à virgule fixe sur huit bits, bit de signe et amplitude.

# Chapitre 5

## Implémentation sur FPGA du système proposé

Ce chapitre présente une description de l'architecture utilisée pour l'implémentation des différents blocs constituant le système DSSS avec une modulation chirp et leurs fonctionnements.

### 5.1 Implémentation de l'émetteur

L'émetteur possède trois entrées, la première est pour le signal *clk8* de période  $T_c$ , utilisé comme signal d'horloge pour le générateur de séquence PN et le compteur binaire modulo-31 qui détermine le début de chaque bit d'information. La deuxième entrée *data in* est utilisée pour les bits d'informations. Le dernier signal (*clear*) est utilisé pour la remise à zéro du compteur binaire et le chargement de l'état initial du registre interne du générateur de séquence PN.

L'émetteur contient également deux bus de sortie 8 bits, qui correspondent au produit entre le bit d'information, les chips de la séquence PN et les parties réelles et imaginaires de la chirp.

Le produit entre les différents termes cité précédemment n'affecte que le bit de signe

du signal de sortie. Il est réalisé avec une porte **XOR** à trois entrées, une pour le bit d'information, la deuxième pour les chips de la séquence PN, la dernière entrée est le bit de signe du signal chirp. Les 7 autres bits de chaque sortie correspondent aux bits  $[0 \dots 6]$  des parties réelles et imaginaires, à la sortie du générateur du signal chirp.

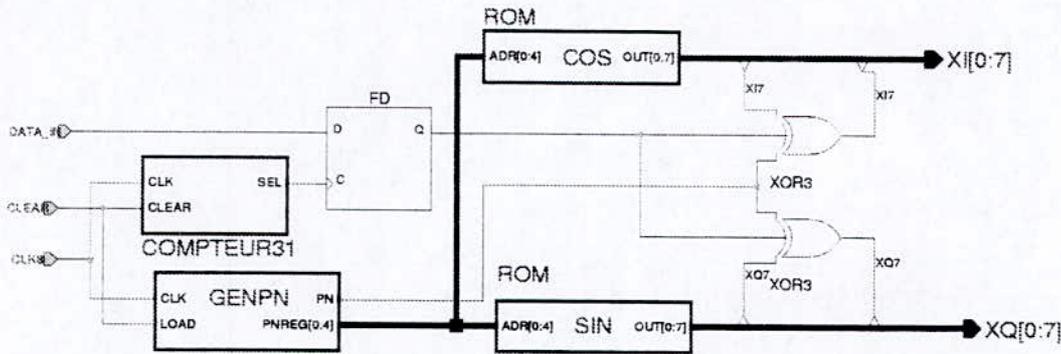


Figure 5.1: émetteur

La figure (5.1), donne une représentation de l'architecture de l'émetteur qui se compose des éléments suivants :

### 5.1.1 Générateur de séquence PN

L'architecture utilisée pour l'implémentation du générateur de séquence PN est la même décrite dans la figure (1.3). Les additionneurs modulo-2 sont remplacés par des portes **XOR**. La figure (5.2) donne une représentation du circuit utilisé, avec une boucle de retour composée des éléments  $[1, 0]$  du registre interne.

Le vecteur  $L[4 \dots 0]$  est utilisé pour initialiser la phase du générateur PN. Cette dernière doit coïncider avec l'adresse contenant la valeur correspondante à la fréquence initiale du signal chirp.

### 5.1.2 Génération du signal chirp

Les composantes réelles et imaginaires du signal chirp sont mémorisées dans deux ROMs ( $8 \times 31$ ), une pour chaque partie. Les deux ROMs sont adressées par l'état (la phase) du

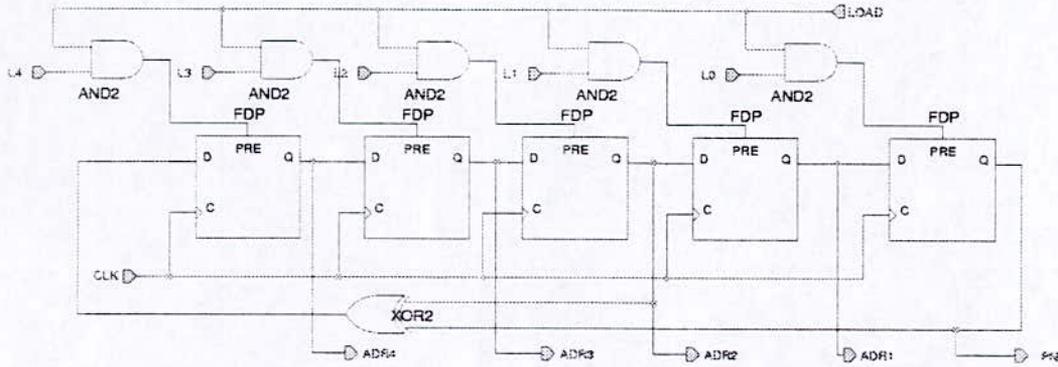


Figure 5.2: générateur de séquence PN modulo 31

générateur de séquence PN. Pour avoir une variation linéaire de la fréquence de la chirp, on doit connaître tous les états du registre du générateur de séquence PN, de telle sorte à ordonner les valeurs du signal chirp dans les ROMs suivant les variations de l'état du registre. L'état suivant du registre, correspond à la fréquence suivante du signal chirp, comme indiqué dans le tableau (5.1).

L'adressage des ROMs dépend de la boucle de retour utilisée dans le générateur de séquence PN ; pour des boucles de retour différentes, l'adressage des ROMs est différent.

état du générateur PN	valeur de la chirp $ch(n)$
11111	$ch(0)$
01111	$ch(1)$
⋮	⋮
11110	$ch(31)$

Tableau 5.1: adressage des ROMs

### 5.1.3 Compteur modulo-31

L'émetteur comporte également un compteur modulo 31 qui délivre le signal de chargement du nouveau bit d'information à chaque nouvelle période bits  $T_b = 31T_c$ .

Le signal numérique à la sortie de l'émetteur passe ensuite par un convertisseur numérique analogique puis par un filtre de transmission, pour obtenir le signal analogique transmis  $x(t)$  qui peut s'écrire sous la forme :

$$x(t) = \sqrt{2}(\cos(w_c t) \sum_{-\infty}^{\infty} Re(I_m)) - \sqrt{2}(\sin(w_c t) \sum_{-\infty}^{\infty} Im(I_m)) \quad (5.1)$$

Où :  $w_c$  représente la fréquence de la porteuse.  $I_m$  représente le signal d'information multiplié par la chirp et la séquence PN.

## 5.2 Implémentation du récepteur

Le signal à la réception passe par un filtre de réception (phase splitter) pour obtenir les parties réelles et imaginaires du signal reçu. Après la conversion analogique numérique, les deux signaux numériques résultants constituent l'entrée du démodulateur, la sortie de ce dernier donne le bit reçu.

Comme indiqué dans le chapitre 3, le récepteur se compose des éléments suivants :

### 5.2.1 Implémentation du filtre

Cette section décrit l'implémentation du filtre utilisé pour la réjection du signal chirp.

Les entrées du filtre sont deux bus 8 bits qui représentent le signal reçu, un pour la partie réelle et l'autre pour la partie imaginaire, les deux bus de sortie donnent le résultat du filtrage de chaque partie (réelle et imaginaire).

#### 5.2.1.1 Algorithme et considération pratique

L'algorithme d'implémentation du filtre doit prendre en considération les contraintes imposées par le package utilisé telles que la vitesse de fonctionnement du FPGA choisie, le nombre de CLBs disponibles pour l'implémentation et le routage des différents composants entrant dans la réalisation du filtre, tels que les additionneurs, les multiplicateurs

et les différents registres pour l'introduction des retards.

A défaut d'utiliser une architecture parallèle qui permet des vitesses d'utilisation plus grande, mais nécessite un grand nombre de ressources du fait qu'elle utilise autant de blocs de multiplications que de coefficients. On a opté pour une architecture sérielle afin de minimiser le nombre de ressources consommées.

L'implémentation du filtre comprend une unité de multiplication addition, une unité de contrôle pour gérer l'exécution des différentes opérations, une logique de génération d'adresses et des mémoires pour le stockage des coefficients du filtre et les parties réelles et imaginaires du signal reçu.

À chaque nouvel échantillon  $r(i)$ , une opération de multiplication addition est réalisée entre les échantillons reçus stockés dans des RAMs et les coefficients du filtre  $h(i)$  issues d'une mémoire ROM interne suivant la relation :

$$y(n) = \sum_{i=0}^{M-1} h_m(i-m)r(i-m) \quad (5.2)$$

La sélection des échantillons des RAMs se fait à partir d'un générateur d'adresse.

Le calcul de la partie réelle et imaginaire du résultat se fait d'une manière sérielle. A chaque nouvel échantillon, on sélectionne la RAM (RAM Réel) contenant les échantillons de la partie réelle. Une fois le résultat de l'opération du filtrage de cette partie effectué, on commute sur la RAM (RAM Imag) contenant les échantillons de la partie imaginaire pour le filtrage de cette dernière.

Une fois le résultat de l'opération de filtrage de la partie réelle et imaginaire du premier échantillon sauvegardé dans des registres, on passe à un nouvel échantillon.

Dans l'implémentation VHDL, on a utilisé le filtre décrit dans le chapitre 3 qui est un filtre symétrique à 3 coefficients. Le code VHDL du filtre comprend une logique de contrôle, l'unité de multiplication addition, deux RAM et une ROM pour le stockage du deuxième coefficient du filtre  $h_2(n)$ .

La figure (5.3) donne une représentation de l'architecture utilisée.

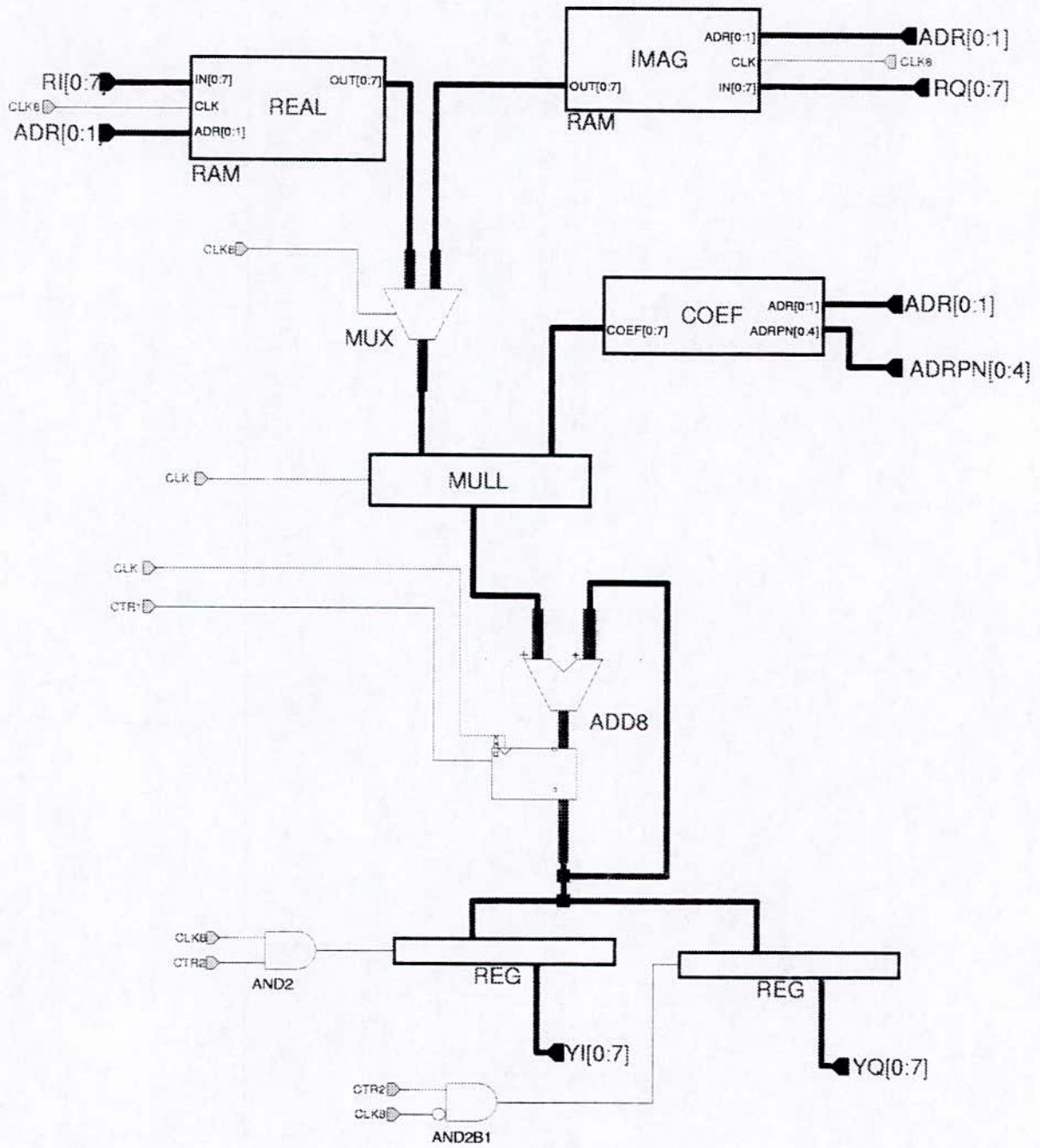


Figure 5.3: architecture du filtre

### 5.2.1.2 Logique de contrôle

La logique de contrôle a pour fonction la gestion des différentes opérations, elle fournit les différents signaux pour le contrôle des différents blocs.

Elle a pour entrée, un signal d'horloge (*clk*) qui constitue le signal de référence pour la génération des différents signaux de sortie, obtenus à partir de diviseur de fréquences et des portes logiques.

Le bloc de contrôle possède aussi un bus de sortie 2 bits issus d'un compteur binaire, contrôlé par le signal *clk*. Il est utilisé comme générateur d'adresse pour la sélection des coefficients du filtre et des valeurs de sortie des deux RAMs.

La figure (5.4) donne le chronogramme des différents signaux de contrôle.

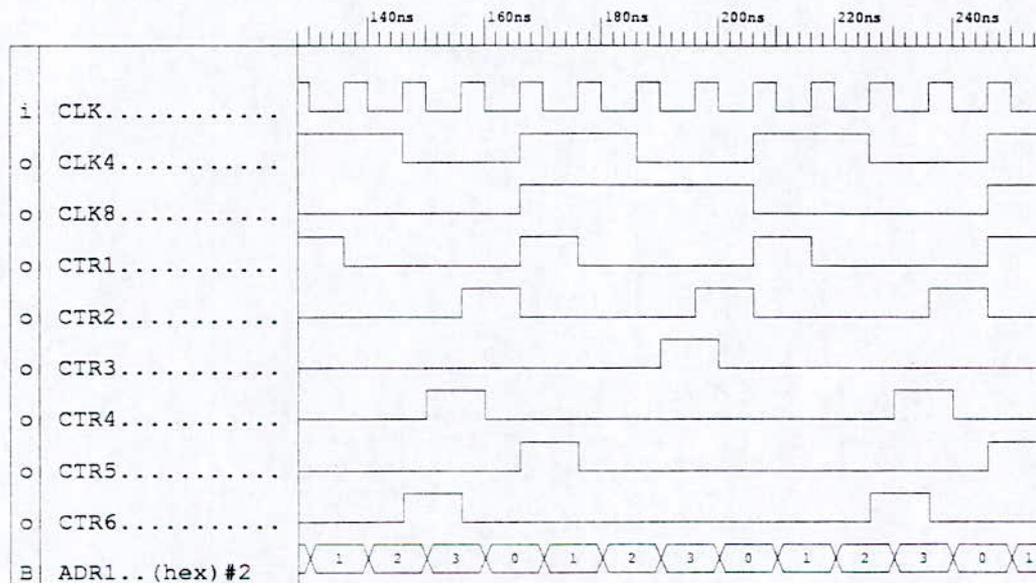


Figure 5.4: signaux de contrôle

### 5.2.1.3 Les RAMs

Les deux RAMs sont utilisées pour le stockage des échantillons nécessaires au filtrage. Puisque le filtre utilisé est d'ordre 3 la profondeur de chaque RAM est de  $3 \times 8$  bits. La première position de la RAM correspond à l'échantillon reçu  $r(i)$ , les deux autres positions correspondent aux échantillons  $r(i-1)$  et  $r(i-2)$ .

#### 5.2.1.4 Générateur des coefficients du filtre

Le générateur des coefficients de filtrage comporte une ROM de taille égale à la longueur de la séquence PN, utilisée pour le stockage du deuxième coefficient.

Puisque le filtre utilisé est symétrique, un registre 8 bits est utilisé pour le stockage du premier coefficient.

Le générateur comporte deux entrées et une sortie :

La première entrée est la valeur du registre interne du générateur de séquence PN. Elle est utilisée pour la sélection de la valeur de sortie de la ROM contenant le deuxième coefficient du filtre.

La deuxième entrée est utilisée pour la sélection du coefficient de filtrage, elle provient du bus 2 bits de la logique de contrôle. Le coefficient de filtrage à la sortie dépend de l'état du générateur d'adresses, si elle est égale à 01 la valeur de sortie est le coefficient du filtre  $h_2(i)$ , sinon la valeur de sortie est le premier coefficient  $h_1$ .

#### 5.2.1.5 Unité de multiplication addition

Cette unité comprend les modules pour les opérations de multiplication et d'addition. Elle comporte deux entrées, une provenant des RAMs et l'autre du générateur de coefficients de filtrage. L'opération de multiplication est contrôlée par le front montant du signal ( $clk$ ) et l'opération d'addition par le front descendant. Les opérations de multiplication et d'addition ne se font pas sur le même front d'horloge, pour donner à l'additionneur et au multiplicateur le temps d'effectuer leurs opérations respectives et pour que les signaux à l'entrée de chaque blocs soient stables.

Les signaux utilisés dans l'opération de filtrage sont les suivants :

- Le signal ( $clk$ ) est un signal d'horloge de fréquence égale à  $1/8T_c$ , utilisé dans l'unité de multiplication addition. Durant les quatre premiers cycles de  $clk$ , le filtrage de la partie réelle est effectué, les 4 cycles suivants sont utilisés pour le calcul de la partie imaginaire.

- Le deuxième signal ( $clk8$ ) est utilisé pour la lecture des RAMs, sa période est égale à la période chip  $T_c$ . Si  $clk8 = 1$  on sélectionne la RAM qui contient la partie réelle du signal reçu, si  $clk8 = 0$ , la RAM de la partie imaginaire est sélectionnée.
- Le troisième signal ( $ctr1$ ) est utilisé pour la remise à zéro du résultat de l'unité de multiplication addition pour que le résultat de la partie réelle ou imaginaire n'interfère pas avec le nouveau résultat.
- Le dernier signal ( $ctr2$ ) contrôle les registres de sauvegarde des résultats de filtrage de chaque partie. A chaque nouveau front montant du signal ( $ctr2$ ), le résultat de l'opération de filtrage est stocké dans un registre durant une période d'échantillonnage.

Le processus de multiplication et d'addition se fait comme suit :

- À l'instant initial (nouvel échantillon) qui correspond au front montant du signal ( $clk8$ ), le signal ( $ctr1$ ) est égal à 1, ce qui permet la remise à zéro du registre de sortie de l'additionneur.
- Au top suivant du signal ( $clk$ ), le signal ( $ctr1$ ) prend la valeur 0 et on fait la multiplication entre la sortie de la RAM de la partie réelle  $r(i)$  et la sortie du générateur de coefficient de filtrage  $h_1$ . Au suivant front descendant de ( $clk$ ), on fait l'addition entre la sortie de l'opération de multiplication  $r(i)h_1$  et la valeur de sortie de l'additionneur qui est nulle. Donc le premier résultat est  $r(i)h_1$ .
- Au top ( $clk$ ) suivant, les signaux sélectionnés pour l'opération de multiplication sont : le signal  $r(i-1)$  à la sortie de la RAM et le coefficient  $h_2(i)$ . La multiplication entre ces deux valeurs est réalisée au front montant de ( $clk$ ). Au front descendant de ce dernier, on fait l'addition entre le résultat de la multiplication  $r(i-1)h_2(i)$  et la sortie précédente de l'additionneur  $r(i)h_1$ . Ce qui donne  $r(i)h_1 + r(i-1)h_2(i-1)$  et le processus se répète pour le troisième coefficient.

A la fin de cette étape, le signal  $ctr2 = 1$ , ce qui permet de stocker le résultat du filtrage de la partie réelle dans un registre auxiliaire.

- Au top ( $clk$ ) suivant, le signal ( $clk8$ ) est à zéro, ce qui permet de sélectionner la RAM qui contient la partie imaginaire du signal ; et le même processus se répète pour cette partie.

## 5.2.2 Implémentation du démodulateur

Les opérations effectuées par le démodulateur sont l'addition entre le signal à la sortie du filtre et le signal reçu ; et la multiplication du résultat par le conjugué du signal chirp.

Puisque la variable de décision est la partie réelle du résultat, il suffit de faire la multiplication entre le résultat de l'opération d'addition de la partie réelle avec le cosinus de la chirp, et le résultat imaginaire de l'addition par le sinus de la chirp.

Les valeurs du signal chirp se trouvent dans deux ROMs, une pour le sinus l'autre pour le cosinus. La taille des ROMs dépend de la longueur de la séquence PN, pour une séquence de longueur  $L = 31$  la taille de la ROM est de  $(31 \times 8)$  bits. La valeur de sortie des ROMs est contrôlée par l'état du registre interne du générateur de séquence PN.

Les modules entrant dans la réalisation du démodulateur sont : une unité de multiplication addition, et une pile pour la sauvegarde des résultats intermédiaires nécessaires pour la démodulation. Le tout est géré par les signaux de contrôle (figure(5.4)) :  $clk$ ,  $clk8$ ,  $ctr3$ ,  $ctr4$ ,  $ctr5$  et le signal  $ctr6$ .

La figure (5.5) présente l'architecture utilisée pour l'opération de démodulation.

### 5.2.2.1 Unité de multiplication accumulation

Cette unité est identique à celle utilisée dans le filtre, la seule différence réside dans les fronts de déclenchements des opérations de multiplication et d'addition.

Pour la multiplication, elle s'effectue au front descendant du signal ( $clk$ ) par contre, l'opération d'addition se fait au front montant de ( $clk$ ).

Les entrées de cette unité dépendent des signaux de contrôles. Elles peuvent être :

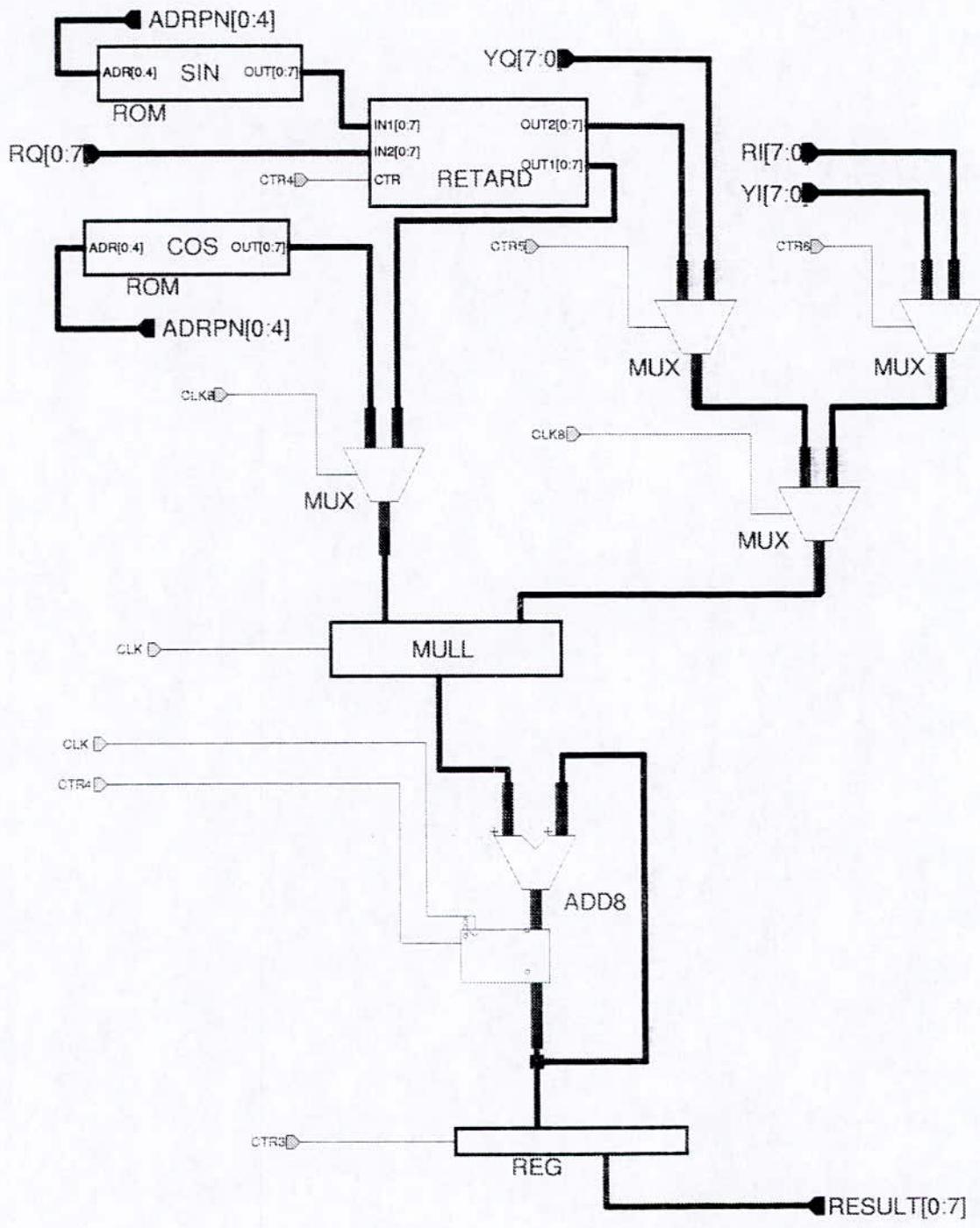


Figure 5.5: architecture du démodulateur

les sorties du filtre, le signal reçu ou les valeurs de sortie des deux ROMs contenant la partie réelle et imaginaire de la chirp.

### 5.2.2.2 Les signaux de contrôle $clk8$ , $ctr5$ , $ctr6$ , $ctr4$ , $ctr3$

Comme indiqué précédemment, ces signaux de contrôle sont utilisés pour la sélection des entrées de l'unité de multiplication accumulation du démodulateur et pour la validation du résultat de sortie.

Dans ce qui suit, on donnera la fonction de chaque signal :

- le signal  $clk8$  :

Il est utilisé pour la sélection des parties réelles et imaginaires à l'entrée du multiplicateur. Si  $clk8 = 0$ , les signaux d'entrée sont les signaux de la partie réelle, s'il est égal à 1, les signaux sélectionnés sont ceux de la partie imaginaire. Entre autre, il permet la sélection de la première entrée du multiplicateur. S'il est égal à 0, l'entrée du multiplicateur est la sortie de la ROM de la partie réelle de la chirp. Dans le cas où  $clk8 = 1$ , l'entrée est la sortie de la ROM de la partie imaginaire de la chirp.

- le signal  $ctr6$  :

Le signal ( $ctr6$ ), valide uniquement quand le signal  $clk8 = 0$ , a pour fonction la sélection de la deuxième entrée du multiplicateur. Dans le cas où  $ctr6 = 1$ , l'entrée du multiplicateur est la partie réelle du signal reçu, dans le cas contraire ( $ctr6 = 0$ ), l'entrée du multiplicateur est la sortie réelle du filtre.

- le signal  $ctr5$  :

Il joue le même rôle que le signal ( $ctr6$ ) ; valide quand  $clk8 = 1$ , il permet la sélection de la deuxième entrée du multiplicateur qui est dans ce cas, la partie imaginaire du signal reçu si  $ctr5 = 0$ .

Si  $ctr5 = 1$ , la deuxième entrée du multiplicateur est le résultat de la partie imaginaire de l'opération de filtrage.

- Le signal ( $ctr4$ ) :

Il joue le même rôle que le signal ( $ctr1$ ) dans l'unité de filtrage (voire section 5.2.1.5).

- le signal  $ctr3$  :

Il permet la validation du résultat de l'opération de démodulation durant une période chip, la validation du résultat se fait durant le niveau haut du signal.

### 5.2.2.3 Processus de démodulation

L'opération de démodulation se réalise selon le processus suivant :

Au début de l'opération, le signal  $ctr4 = 1$ , ce qui permet la remise à zéro de l'accumulateur de l'unité de multiplication addition. Les signaux  $clk8 = 0$  et  $ctr6 = 1$ , permettent la sélection des entrées du multiplicateur qui sont respectivement : Le signal  $cos(n)$  issue de la ROM cosinus et la partie réelle du signal reçu  $ri(n)$ .

Durant le front descendant du signal ( $clk$ ), la multiplication entre ces deux valeurs est réalisée. Donc, le résultat à la sortie de l'additionneur au front montant de ( $clk$ ) est :  $ri(n)cos(n) + 0$ .

Au suivant front descendant de ( $clk$ ), le signal ( $ctr6$ ) prend la valeur zéro et la deuxième entrée du multiplicateur devient la sortie réelle du filtre  $yi(n)$ . Après l'opération d'addition, le résultat de l'accumulateur est :  $cos(n)ri(n) + cos(n)yi(n)$ .

Au top ( $clk$ ) suivant, le signal  $clk8 = 1$ . Dans ce cas, la première entrée du multiplicateur est la sortie de la ROM sinus  $sin(n)$  retardée d'un échantillon du fait que le registre de la séquence PN change de valeur au front montant de ( $clk8$ ). Le signal ( $ctr5$ ) qui est égal à 1, sélectionne le deuxième signal à l'entrée du multiplicateur, qui est dans ce cas, la sortie imaginaire du filtre  $yq(n)$ . Une fois l'opération de multiplication et d'addition effectuée, le résultat de l'accumulateur est :  $cos(n)ri(n) + cos(n)yi(n) + sin(n)yq(n)$ .

Lorsque  $clk5 = 0$ , la deuxième entrée du multiplicateur est le signal reçu  $rq(n)$  retardé lui aussi d'un échantillon. Après l'opération de multiplication addition, on obtient le résultat suivant:  $cos(n)ri(n) + cos(n)yi(n) + sin(n)rq(n) + sin(n)yq(n)$ . La validation de ce résultat est effectuée avec le signal ( $ctr3$ ) qui est égale à 1.

Le résultat de l'opération de démodulation de chaque chips s'obtient un échantillon et demi après sa réception.

Le résultat de la démodulation une fois obtenu, on utilise un intégrateur pour obtenir la variable de décision  $\xi$ , dont le bit de signe détermine le bit reçu.

### 5.2.3 Implémentation du bloc de synchronisation

Comme indiqué dans le chapitre 2, le processus de synchronisation se décompose en deux étapes distinctes, l'acquisition suivie du processus de tracking.

#### 5.2.3.1 Acquisition

L'unité d'acquisition a pour rôle la synchronisation initiale des deux séquences de réception et d'émission. Comme indiqué précédemment (chapitre 2), la décision de l'acquisition du signal se fait après chaque période bit. Pour cela, on utilise un compteur modulo-31 qui délivre deux signaux :

Le signal (*sel*), qui définit la fin de chaque période bit et le signal (*clr*), qui détermine le début de la période bit. Le compteur s'incrémente à chaque front montant du signal (*clk8*).

L'unité d'acquisition se compose de deux unités de multiplication addition. Une pour la partie réelle du signal reçu et l'autre pour la partie imaginaire. Elles sont contrôlées par le signal (*clr*), qui permet la remise à zéro des additionneurs à chaque nouvelle période. Les opérations de multiplications et d'additions sont cadencées par le signal (*clk*), ce qui permet d'avoir huit échantillons pour chaque chip.

Les entrées de la première unité de multiplication addition sont d'une part, la partie réelle du signal reçu, et d'autre part le résultat du produit entre la séquence PN présente et la sortie de la ROM cosinus de la chirp.

Pour la deuxième unité, elle a comme entrée la partie imaginaire du signal reçu et le produit entre la séquence PN présente et la sortie de ROM sinus de la chirp.

Le circuit de tracking suit le même principe décrit dans la figure (3.3).

Pour des raisons de normalisation du résultat de sortie, les entrées des deux unités sont décalées de 3 positions vers la droite ce qui équivaut à une divisions par huit de chaque entrée.

### 5.2.3.2 Module d'élevation au carré

Contrairement à l'opération de multiplication de deux nombres binaires, la multiplication d'un nombre par lui-même réduit considérablement le nombres de ressources utilisées.

Pour donner une idée de la simplicité d'implémentation de l'opération d'élevation au carré, le tableau (5.2) donne un exemple pour un nombre binaire X de 4 bits, avec un résultat P sur 7 bits.

$P_6$	$P_5$	$P_4$	$P_3$	$P_2$	$P_1$	$P_0$
			$x_0.x_3$	$x_0.x_2$	$x_0.x_1$	$x_0.x_0$
		$x_1.x_3$	$x_1.x_2$	$x_1.x_1$	$x_1.x_0$	
	$x_2.x_3$	$x_2.x_2$	$x_2.x_1$	$x_2.x_0$		
$x_3.x_3$	$x_3.x_2$	$x_3.x_1$	$x_3.x_0$			
$x_2.x_3$	$x_1.x_3$	$x_0.x_3$	$x_0.x_2$	$x_1$	0	$x_0$
$x_3$		$x_1.x_2$		$x_0.x_1$		
		$x_2$				

Tableau 5.2: élévation au carré d'un nombre binaire

La partie supérieure du tableau (5.2), représente des opérations **and** logiques entre les différents éléments du nombre X. La deuxième partie du tableau, représente les opérations d'additions binaires.

Comme on peut le constater, le nombre d'additionneurs nécessaires pour l'obtention du résultat est moins important que dans le cas d'une multiplication classique. De plus, certains résultats s'obtiennent directement. En plus, la logique combinatoire se trouve simplifiée.

Par exemple, le terme  $x_0.x_1$  dans la colonne  $P_1$  est décalé dans la colonne  $P_2$ . Le décalage des sommes des termes identiques est effectué dans toutes les colonnes.

### 5.2.3.3 Processus d'acquisition

Une fois que le résultat de l'opération d'intégration de chaque unité est validé par le signal *sel* du compteur, les résultats sont élevés au carré puis additionnés avec un autre additionneur.

Le résultat de cette sommation constitue l'entrée d'un comparateur, qui compare le résultat de l'addition avec une valeur pré enregistrée qui définit le seuil d'acquisition.

Si le résultat de cette opération est négatif, le signal de sortie du comparateur *sync* est à un niveau bas, ce qui permet d'avancer le signal *clk8* d'une demi-période, lorsque le signal *sel* est à 1.

Cette opération se répète jusqu'à ce que le signal *sync* devienne égal à 1. Une fois ce résultat obtenu, on passe à la phase de tracking.

### 5.2.4 Tracking

Dans la phase de tracking, les unités de multiplication accumulation utilisées dans le processus d'acquisition sont utilisées pour la phase de tracking.

Pour cette phase, les entrées de ces unités deviennent :

Pour la première unité, la première entrée est la partie réelle du signal reçu. La deuxième entrée est le produit entre les valeurs *early* de la séquence PN et la sortie de la ROM contenant la partie réelle du signal chirp.

Les entrées de la deuxième unité sont d'une part, la partie réelle du signal reçu et d'autre part, le produit entre les valeurs *late* de la séquence PN et la valeur de sortie de la ROM cosinus de la chirp.

Les valeurs *early et late* du signal PN sont obtenues à partir de 3 bascules D, contrôlées par le signal (*clk4*) de période  $T_c/2$ , issu de l'unité de contrôle. Ces bascules, sont placées en série, comme indiqué dans la figure (5.6).

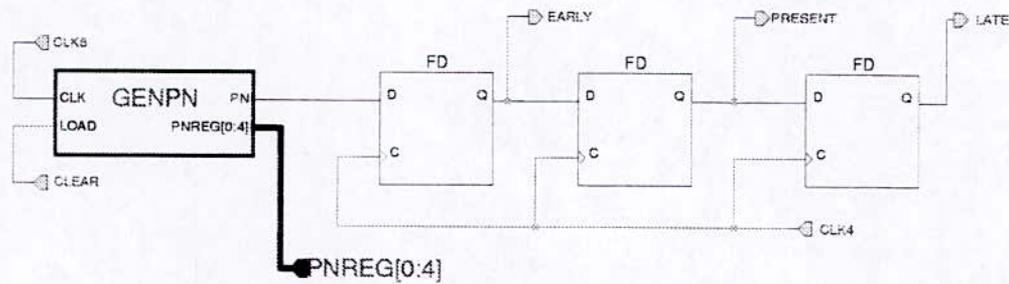


Figure 5.6: signaux early, present, late de la séquence PN

La sortie de la première bascule donne la valeur *early* de la séquence PN, la sortie de la deuxième bascule donne la valeur *présente* de la séquence PN, utilisée dans l'opération d'acquisition et de despreading. La sortie de la troisième bascule donne la valeur *late* de la séquence ; les trois séquences sont décalées entre elle de  $T_c/2$ . Pour la génération des trois valeurs *early*, *late* et *présente* à la sortie du générateur du signal chirp, on utilise le même principe sauf que chaque bascule D est remplacée par un registre huit bits, contrôlées toujours par le signal ( $clk_4$ ).

Une fois que le résultat de chaque branche est validé par le signal *sel* du compteur, le résultat de la différence entre les deux branches est entré dans un comparateur, qui détermine, en fonction de l'entrée si les deux séquences sont bien synchronisées ou pas. La comparaison est faite avec un facteur de tolérance qui est fixe et égal à  $\pm 0.03125$ .

Si la différence entre les deux branches est supérieure à cette valeur, le bus de sortie du comparateur prend la valeur '01', cela indique que la branche *early* est en avance par rapport à l'autre branche. S'il est dans l'intervalle de tolérance la valeur de sortie est '00', dans ce cas, la synchronisation entre les deux séquences est acceptable. Dans le dernier cas, la valeur du bus de sortie de l'unité de comparaison est égale à '10', cela signifie que la branche *late* est en avance par rapport à la branche *early*.

La valeur du bus de sortie de l'unité de comparaison est une entrée de l'unité de contrôle qui délivre les différents signaux de contrôle. Si le signal du comparateur est égal à '01', les compteurs internes de cette unité sont incrémentés de 1. Dans le cas où le résultat de

la comparaison est '10', les compteurs sont décrémentés de 1. Dans le dernier cas, si la bus de sortie de l'unité de comparaison prend la valeur '00', aucune opération n'affecte les compteurs puisque la synchronisation est acceptable entre les deux séquences.

Les opérations d'avancement ou de retardement des différents signaux de contrôles, notamment les signaux d'horloge (*clk8*) et (*clk4*) sont effectués au début de chaque nouvelle période bit.

### 5.3 Conclusion

Dans ce chapitre, nous avons donné une description des différents blocs constituant notre système. Une fois la synthèse des différents programmes réalisée, l'opération suivante est l'implémentation du système et la vérification des résultats.

# Chapitre 6

## Evaluation des performances

### 6.1 Implémentation

Une fois la description VHDL des différents éléments entrant dans la réalisation de notre système est terminée, l'opération suivante est la synthèse du circuit. Cette opération permet la traduction du code VHDL en un ensemble de fonctions logiques de base et leurs optimisations, afin qu'il soit compatible avec l'architecture FPGA.

Le résultat de cette opération permet de vérifier le fonctionnement du circuit sans prendre en considération les contraintes temporelles (retards), liées à chaque circuit logique. Ces retards sont dus essentiellement au temps de réponse propre à chaque circuit logique et aux délais de transmission du signal.

Après la vérification du bon fonctionnement du circuit, on passe à l'opération d'implémentation proprement dite. Cette opération permet la distribution des différentes fonctions logiques obtenues dans la synthèse sur les différents CLB et leurs interconnexions avec les ressources de routage disponibles dans le FPGA choisi.

Pour l'implémentation de notre système, nous avons opté pour le package **4013XLBG256** dans la famille **XC4000XL**, avec un *speed grade* de -08. Ce dernier, est une unité de mesure du retard interne, plus il est petit plus les fréquences de fonctionnements sont grandes.

Après introduction des contraintes temporelles, telle que la fréquence maximale de fonctionnement du circuit, nous avons obtenu les résultats de l'opération d'implémentation résumés dans les tableaux suivants :

Ressources	nombre utilisé	nombre disponible	%
IOBs :	37	192	19
CLBs :	497	576	86
Latches :	102	1152	8
Flops :	220	1152	19
LUTs 4 entrées :	874	1152	75
LUTs 3 entrées :	137	576	23

Tableau 6.1: ressources utilisées

Retard moyen de connexion :	2.214 ns
Retard moyen de connexion dans les nets critiques :	0.000 ns
Retard moyen d'horloge :	0.162 ns
Retard maximum pour les pins :	8.077 ns
Retard moyen de connexion pour les plus mauvais nets :	6.606 ns
Fréquence maximale d'utilisation :	14.177 MHZ

Tableau 6.2: différents retards

Les tableaux précédents, donnent une évaluation des ressources utilisées pour l'implémentation du circuit et les retards et délais qui en découlent. Ces derniers, fixent la fréquence maximale d'utilisation, qui dans notre cas impose que la période du signal *clk* doit être au minimum égale à 70.538 ns, pour avoir une fréquence maximale d'utilisation de 14.177 MHz. A partir de ces résultats, le débit maximum est de 1.77 M chip / seconde.

La figure (6.1) donne une représentation de la distribution des différents composants entrants dans l'implémentation du circuit dans le FPGA cité ci-dessus.

Une fois l'implémentation du circuit terminée, les résultats obtenus imposent des contraintes de propagation du signal et des délais de réponse des différents composants au signal d'horloge.

Dans ce qui suit, on donne les résultats de simulation du système proposé

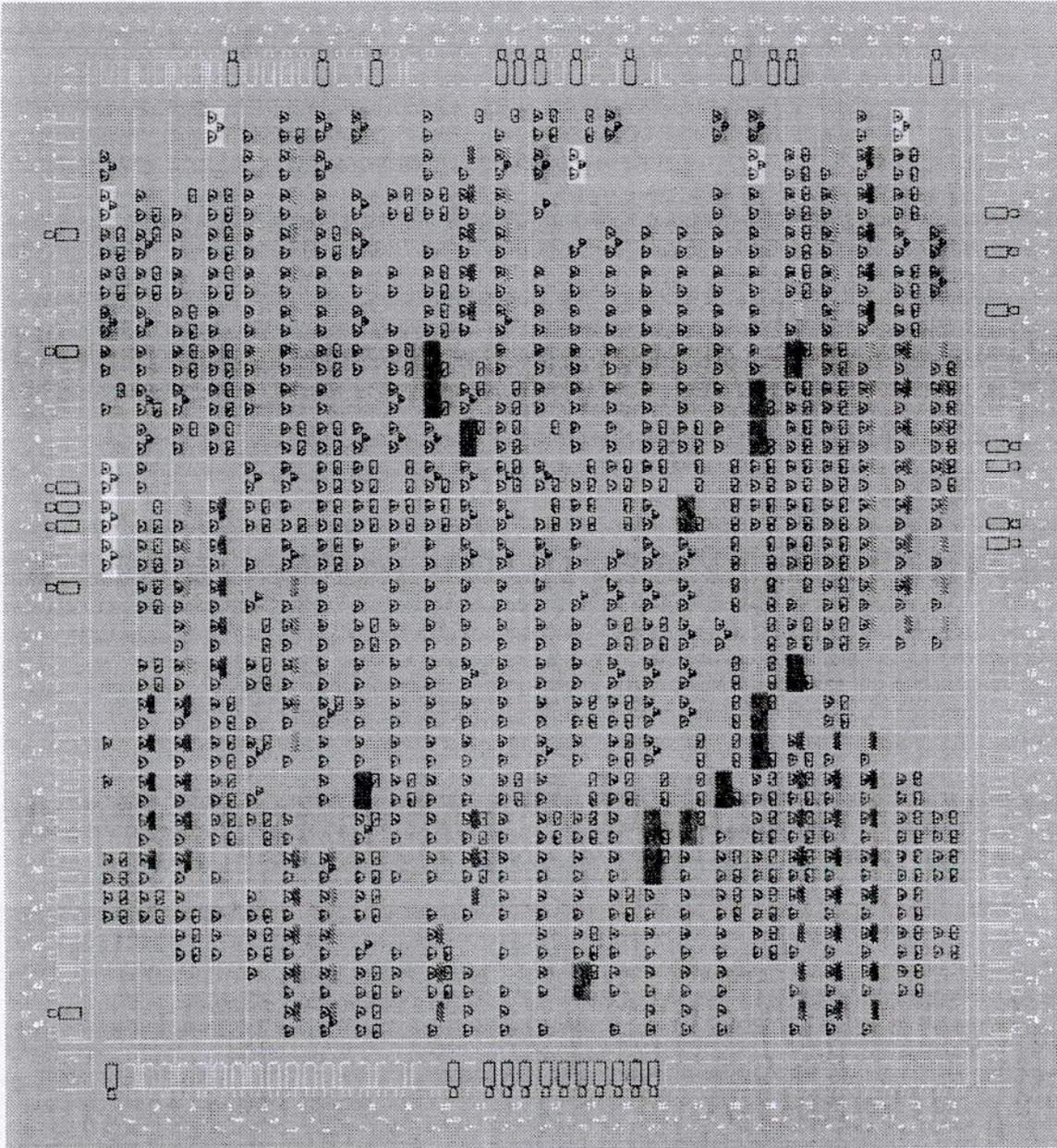


Figure 6.1: circuit après implémentation

## 6.2 Vérification des résultats de simulation

### 6.2.1 Résultats de filtrage et de démodulation

La vérification de l'opération de filtrage et de démodulation est basée sur la comparaison entre les résultats obtenus avec le système après implémentation et ceux obtenus avec MATLAB. Pour ce faire, on utilise un vecteur test qui contient le signal chirp multiplié par la séquence PN, après l'opération de mise au format du vecteur (quantification en virgule fixe), il est utilisé comme entrée dans les simulations du circuit. Les résultats obtenus après leurs conversions en nombres réels  $y_{sim}$ , sont comparés à ceux obtenus avec MATLAB  $y_{mat}$ . La figure (6.2) donne l'erreur quadratique moyenne  $\zeta = 20\log_{10}(|y_{sim} - y_{mat}|)$  obtenue dans les opérations de filtrage de la partie réelle et imaginaire et dans l'opération de démodulation.

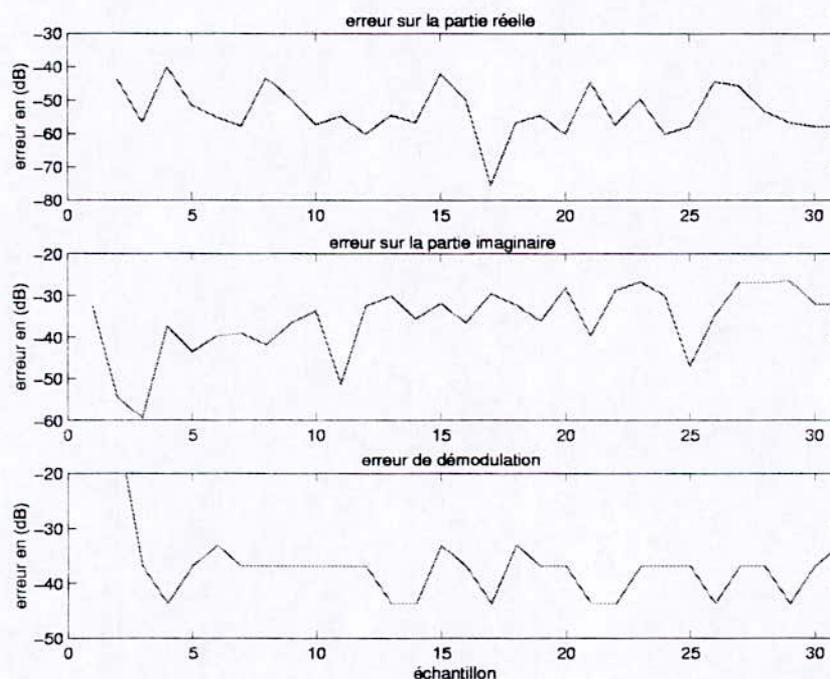


Figure 6.2: analyse de l'erreur en (dB)

Comme on peut le constater, la différence entre les deux résultats de démodulation varie en moyenne entre  $-20\text{dB}$  et  $-40\text{dB}$ , ce qui est un résultat acceptable si on prend en considération la précision limitée par le format de donnée choisie (8 bits).

## 6.2.2 Vérification de la boucle de synchronisation

La seconde partie des tests concerne la vérification du processus d'acquisition. Cette vérification se divise en deux étapes. La première est la vérification du processus d'acquisition, la seconde est la vérification du processus de tracking.

En ce qui concerne la phase d'acquisition, on utilise une séquence PN avec un offset prédéfini par rapport à la séquence PN du récepteur, l'opération d'acquisition doit s'effectuer après un nombre connu de cycle puisqu'on connaît le décalage entre les deux séquences.

Après la phase d'acquisition, la boucle de synchronisation bascule sur le mode tracking. Dans ce mode, on doit vérifier si les horloges de l'émetteur et du récepteur se synchronisent. Les figures suivantes, donnent le résultat des simulations pour le mode de synchronisation.

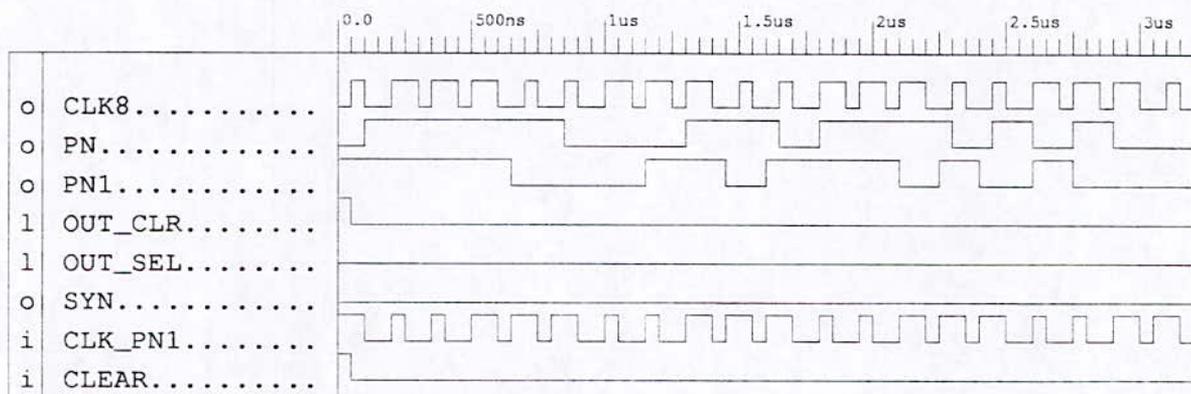


Figure 6.3: décalage initial

La figure (6.3) représente le décalage initial entre la séquence PN du récepteur *pn* et la séquence de l'émetteur *pn1*. Les deux séquences ne sont pas synchronisées, car comme on peut le constater, elles sont décalées de  $1.5T_c$ . Le système doit faire la synchronisation après un nombre prédéfini de périodes bit, qui dans notre cas doit être égal à 3 périodes.

La figure (6.4) représente la fin de la phase d'acquisition. En effet, le signal *syn* qui indique l'acquisition est à 1 après trois périodes bit. Les signaux *out\_clr* et *out\_sel* représentent les signaux de fin et de début d'intégration.

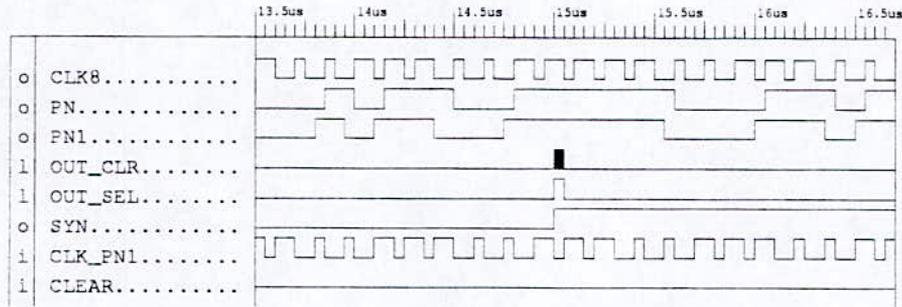


Figure 6.4: acquisition

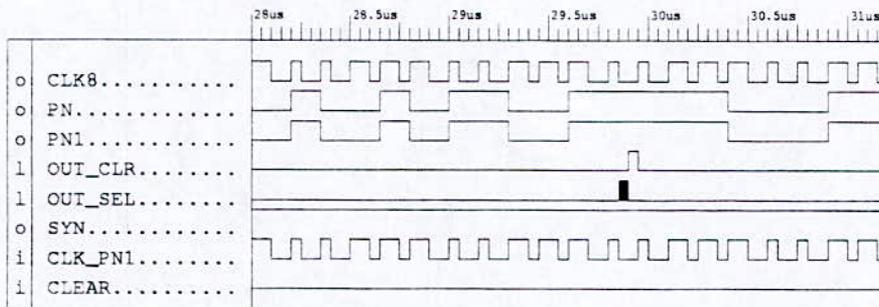


Figure 6.5: synchronisation

Comme on peut le constater, les deux séquences ne sont pas synchronisées, puisqu'il reste un décalage d'une demi-période entre les deux séquences. Comme le montre la figure (6.4), le signal *clk8* qui représente la période *chip* n'est pas synchronisé avec le signal d'horloge de l'émetteur *clk\_pn1*.

La mise à 1 du signal *sync* permet au système de passer à la phase de tracking pour avoir la synchronisation la plus parfaite possible entre les deux séquences.

La figure (6.5) représente l'exécution de cette phase. On remarque bien que les deux séquences sont parfaitement synchronisées, de même que leurs signaux d'horloges.

### 6.3 Conclusion

Dans ce chapitre, nous avons présenté les résultats d'implémentation du circuit, qui permettent de confirmer la fonctionnalité du circuit et donnent les performances obtenues en terme de fréquence de fonctionnement et d'erreur de quantification.

## Conclusion générale

Dans ce travail, nous avons présenté un système de communications à spectre étalé qui utilise une modulation chirp en plus de la séquence PN. Le but d'utiliser une modulation chirp est d'exploiter la distribution temps-fréquence de ce signal. On donne à chaque échantillon du signal d'information une fréquence instantanée propre.

L'utilisation à la réception d'un filtre réjecteur, avec une réponse impulsionnelle finie variable dans le temps, synchronisé avec la fréquence instantanée du signal chirp, permet d'augmenter les performances du système.

Les expressions de performance en terme de probabilité d'erreur de réception dans un canal avec un bruit blanc Gaussien, ont été calculées analytiquement et comparées avec les résultats obtenus par simulation avec différents types d'interférences possédant des distributions différentes.

Les résultats montrent clairement que la méthode proposée augmente les performances du système comparées à un système DSSS classique, même dans le cas où il n'y a pas d'interférence (bruit blanc uniquement).

L'utilisation de la chirp donne au système une capacité inhérente à réduire l'effet des interférences, sans utiliser un autre système pour rejeter les interférences.

Aussi, nous avons présenté les méthodes de synchronisation dans les systèmes DSSS, et les simulations du temps moyen et de la probabilité d'erreur d'acquisition.

Une méthode pour la synchronisation du système avec une modulation chirp a été présentée. Elle est basée sur une *early late bit tracking loop*, que nous avons modifiée pour qu'elle soit compatible avec la modulation utilisée.

Nous avons consacré la dernière partie de ce travail à l'implémentation du système, en utilisant un format de donnée huit bits. Cette implémentation a été réalisée en utilisant le langage de description VHDL et le logiciel de développement XILINX Fondation V1.5. Les résultats des simulations obtenus après implémentation du système permettent de valider le fonctionnement du circuit avec des fréquences de fonctionnement de l'ordre de 14MHZ.

# Bibliographie

- [1] J.G. Proakis, 'Digital Communication', 3<sup>rd</sup> Edition , McGraw-Hill, New York,1995.
- [2] R.L. Pickholtz, D.L.Schilling, and L.B. Milstein,'Theory of Spread-Spectrum Communication - A Tutorial'. IEEE tras on Communication, Vol. COM-25, No 5, May 1982.
- [3] P.G. Flikkema, ' Spread Spectrum Technique for Wireless Communication', IEEE Signal Processing Magazine, pp 26-36, May 1997.
- [4] C.Deng and C.Chien, ' Interference-Robust Serial Dual-path Threshold Referenced Architecture for PN Acquisition', IEEE GlobeCom, Sydney, Australia, NOv, 1998.
- [5] K.K. Chawla, ' Parallel Acquisition of PN Sequences in DSSS Systems', IEEE tras On Communications. Vol 42, NO 5, pp 2155-2164, May 1994.
- [6] B.B. Ibrahim, A.H. Aghvami, ' Direct Sequence Spread Spectrum Code Acquisition in Mobile Fading Channel Using Matched Filter With Reference Filtering', IEEE Global Telecommunication Conference, VOI 2, pp 1085-1089, 1993.
- [7] Jack K. Holmes, 'Acquisition Time Performane of PN Spread-Spectrum Systems', IEEE trans. On Communication , pp 778-783 august 1977.
- [8] J.D. Laster, J.H. Reed, ' Interference Rejection in Digital Wireless Communications', IEEE Signal Processing Magazine, pp 37-61, May 1997.

- [9] Mehmet V. Tazebay, Ali N. Akansu, ' A Performance Analysis of Interference Excision Techniques in Direct Sequence Spread Spectrum Communications', IEEE Transactions On Signal Processing, VOL. 46, NO. 9, pp 2530-2535 September 1998.
- [10] M.V Tazebay, A.L. Akansu. 'Performance Analysis of Direct Sequence Spread Spectrum Communication System Employing Interference Excision' Proc IEEE Digital Signal Processing Workshop, pp 125-128 Spetember 1996.
- [11] C.Wang, M.G. Amin, ' Performance Analysis of Instantaneous Frequency-Based Interference Excision Techniques in Spread Spectrum Communication', IEEE trans. On Signal Processing Vol 46. NO 1 pp 70-82, January 1998.
- [12] M.G. Amin, ' Interference Mitigation in Spread Spectrum Communication Systems Using time-frequency Distribution', IEEE trans. On Signal. Proc. Vol 45, pp 90-101, jan 1997.
- [13] A.Belouchrani and MG Amin, ' A Tow-Sensor Array Blind Beamformer for Direct Sequence Spread Spectrum Communication', IEEE trans. On Signal Proc, Vol 47 pp 2191-2199, August 1999.
- [14] J. Ketchum, J.Proakis, ' Adaptive algorithms for estimating ans suppressing narrow-band interference in PN Spread Spectrum Systems'.IEEE trans. On Communication, pp 913-924, May 1982.
- [15] Edward A.Lee, ' Digital Communication', second edition, Kluwer Academic Publishers, 1994.
- [16] M.G. Amin, Chenshu Wang, Alan R. Lindsey, 'Optimum Interference Excision in Spread Spectrum Communication Using Open-Loop Adaptive Filters', IEEE trans. On Signal. Proc. Vol 47, NO. 7, pp 1966-1976, July 1999.

# Annexe A

## Génération Des Séquences PN

L	$N = 2^L - 1$	boucle de retour pour les m-séquences	# m-sequence
2	3	[2,1]	2
3	7	[3,1]	2
4	15	[4,1]	2
5	31	[5,3][5,4,3,2][5,4,2,1]	6
6	63	[6,1][6,5,2,1][6,5,3,2]	6
7	127	[7,1] [7,3] [7,3,2,1] [7,4,3,2] [7,6,4,2] [7,6,3,1] [7,6,5,2] [7,6,5,4,2,1] [8,6,4,3,2,1]	18
8	255	[8,4,3,2] [8,6,5,3] [8,6,5,2] [8,5,3,1] [8,6,5,1] [8,7,6,1] [8,7,6,5,2,1] [8,6,4,3,2,1]	16
9	511	[9,4] [9,6,4,3] [9,8,5,4] [9,8,4,1] [9,5,3,2] [9,8,6,5] [9,8,7,2] [9,6,5,4,2,1] [9,7,6,4,3,1] [9,8,7,6,5,3]	48
10	1023	[10,3] [10,8,3,2] [10,4,3,1] [10,8,5,1] [10,8,5,4] [10,9,4,1][10,8,4,3] [10,5,3,2] [10,5,2,1] [10,9,4,2] [10,6,5,3,2,1] [ 10,9,8,6,3,2] [10,9,7,6,4,1] [10,7,6,4,2,1] [10,9,8,7,6,5,4,3] [10,8,7,6,5,4,3,1]	60

Tableau A.1: boucle de retour pour la génération des m-séquences

L	$N = 2^L - 1$	m-sequences	inter corrélation		
5	31	[5,3] [5,4,3,2]	7	-1	-9
6	63	[6,1] [6,5,2,1]	15	-1	-17
7	127	[7,3] [7,3,2,1] [7,3,2,1] [7,5,4,3,2,1]	15	-1	-17
8	255	[8,7,6,5,2,1] [8,7,6,1]	31	-1	-17
9	511	[9,4] [9,6,4,3] [9,6,4,3] [9,8,4,1]	31	-1	-33
10	1023	[10,9,8,7,6,5,4,3] [10,9,7,6,4,1] [10,8,7,6,5,4,3,1] [10,9,7,6,4,1] [10,8,5,1] [10,7,6,4,2,1]	63	-1	-65
11	2047	[11,2] [11,8,5,2] [11,8,5,2] [11,10,3,2]	63	-1	-65

Tableau A.2: génération des séquences Gold

L	pic d'inter corrélation $\phi_{max}$	$\phi_{max}/\phi(0)$	$t(m)$	$t(m)/\phi(0)$
3	5	0.71	5	0.71
4	9	0.60	9	0.60
5	11	0.35	9	0.29
6	23	0.36	17	0.27
7	41	0.32	17	0.13
8	95	0.37	33	0.13
9	113	0.22	33	0.06
10	383	0.37	65	0.06
11	287	0.14	65	0.03

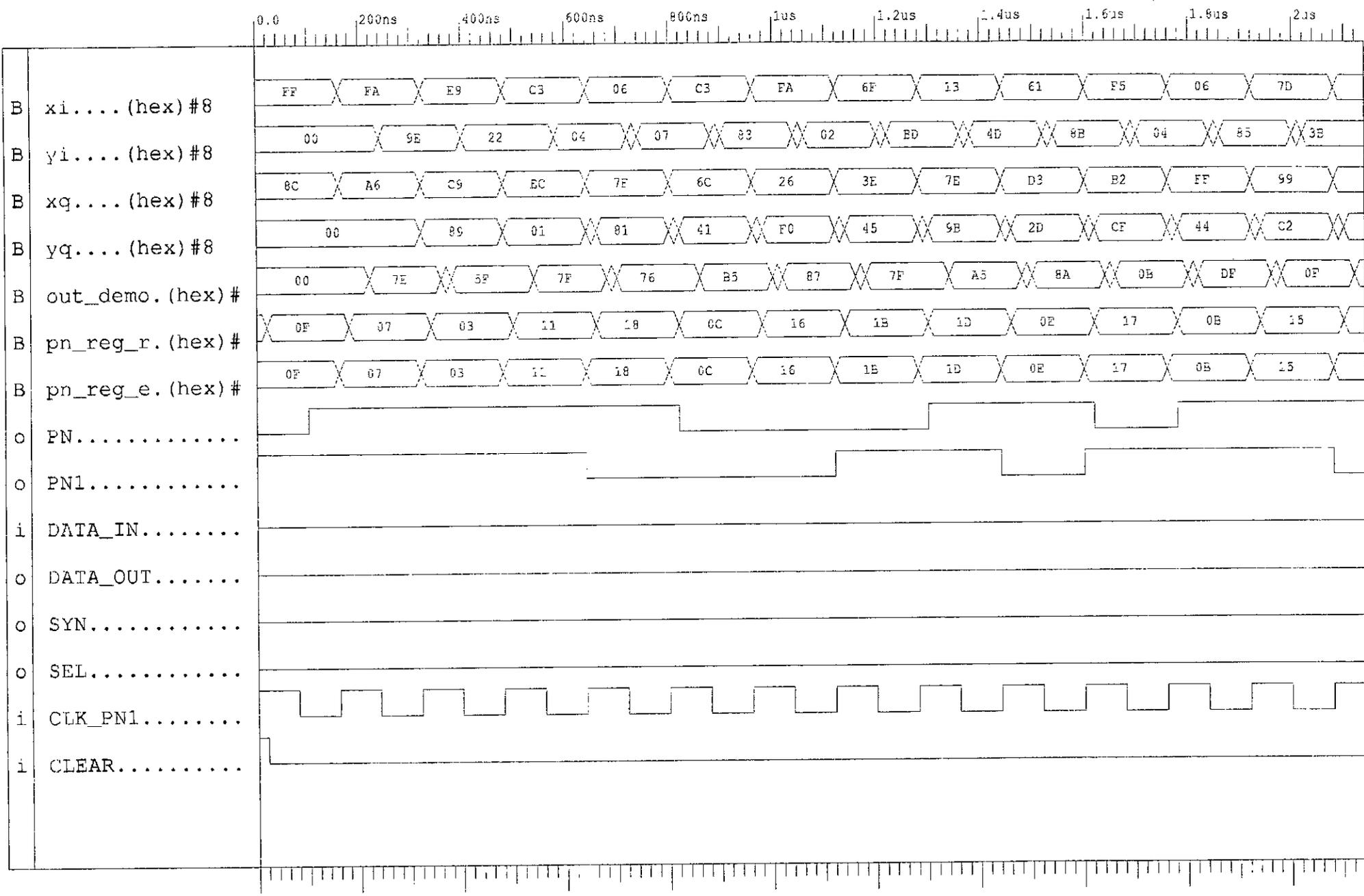
Tableau A.3: pic d'inter corrélation des m-séquences et des codes Gold

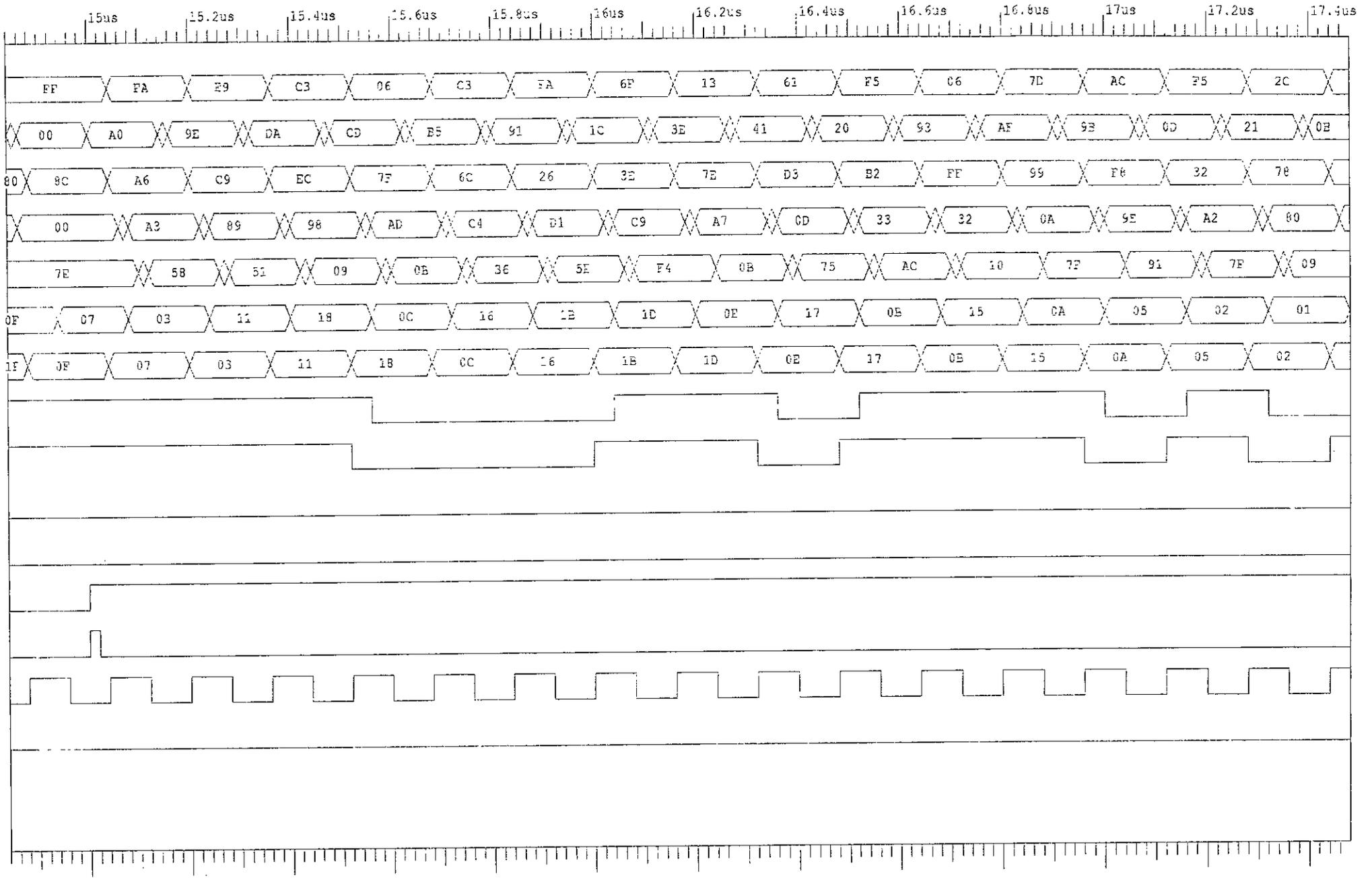
# Annexe B

## Simulation

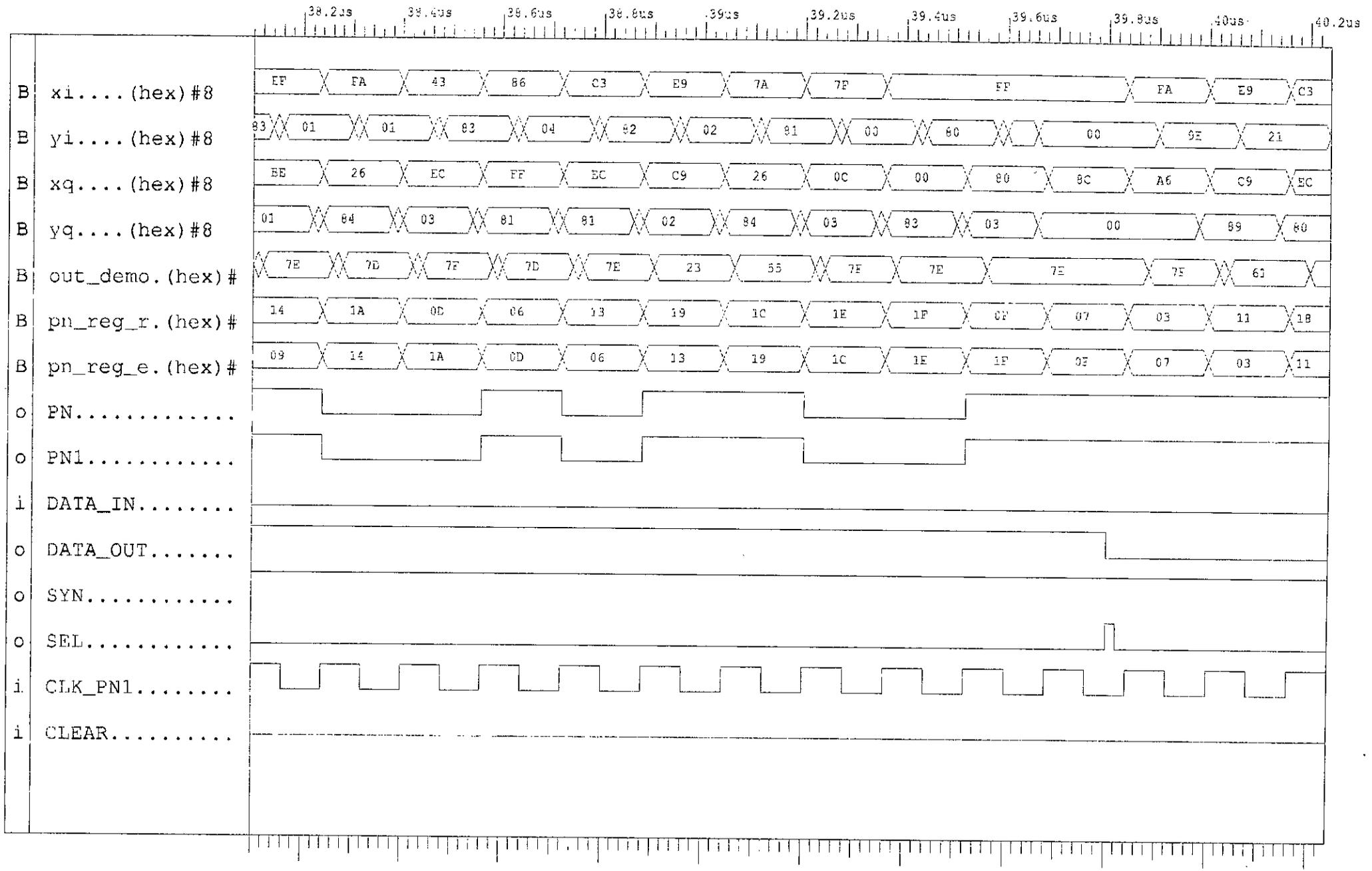
clk	signal d'horloge
clk_pn1	signal d'horloge du générateur PN
xi	partie réelle du signal reçu
xq	partie imaginaire du signal reçu
yi	filtrage de la partie réelle
yq	filtrage de la partie imaginaire
demo_out	résultat de la démodulation
pn_reg_e	état du registre du générateur PN de l'émetteur
pn_reg_r	état du registre du générateur PN du récepteur
PN	séquence PN du récepteur
PN1	séquence PN de l'émetteur
data_in	bit d'information
data_out	bit reçu
syn	signal de synchronisation
sel	signal du début de la période bit

Tableau B.1: les signaux d'entrée sortie









# Annexe C

## Système Sous Simulink

Dans cette annexe, on présente le système de communications à spectre étalé avec une modulation chirp réalisé sous simulink.

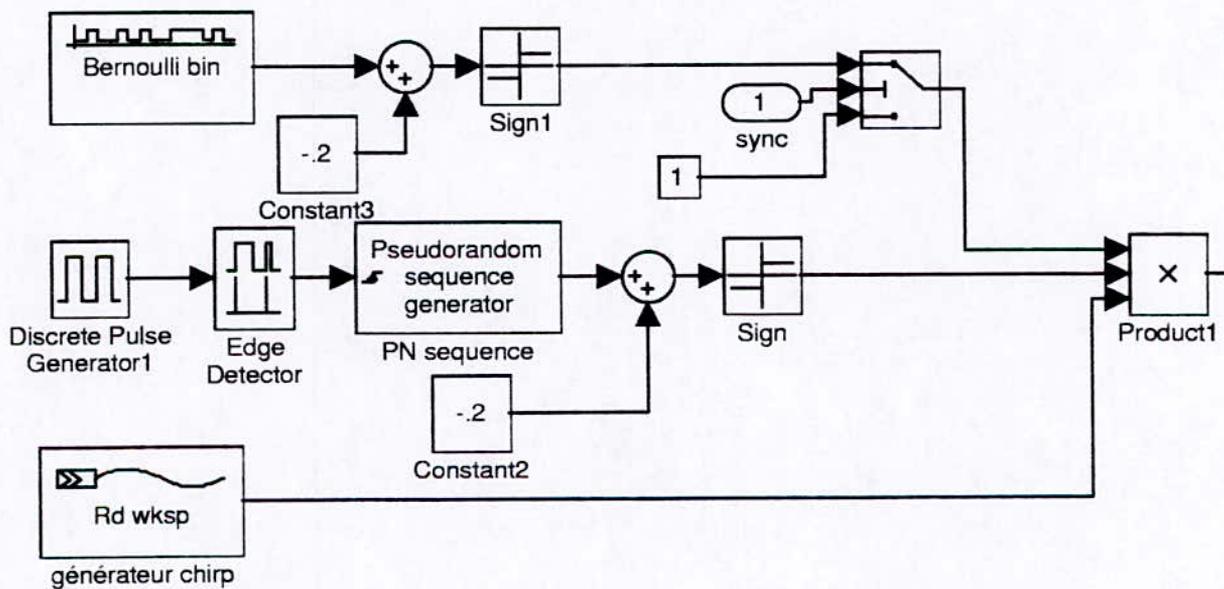


Figure C.1: emetteur

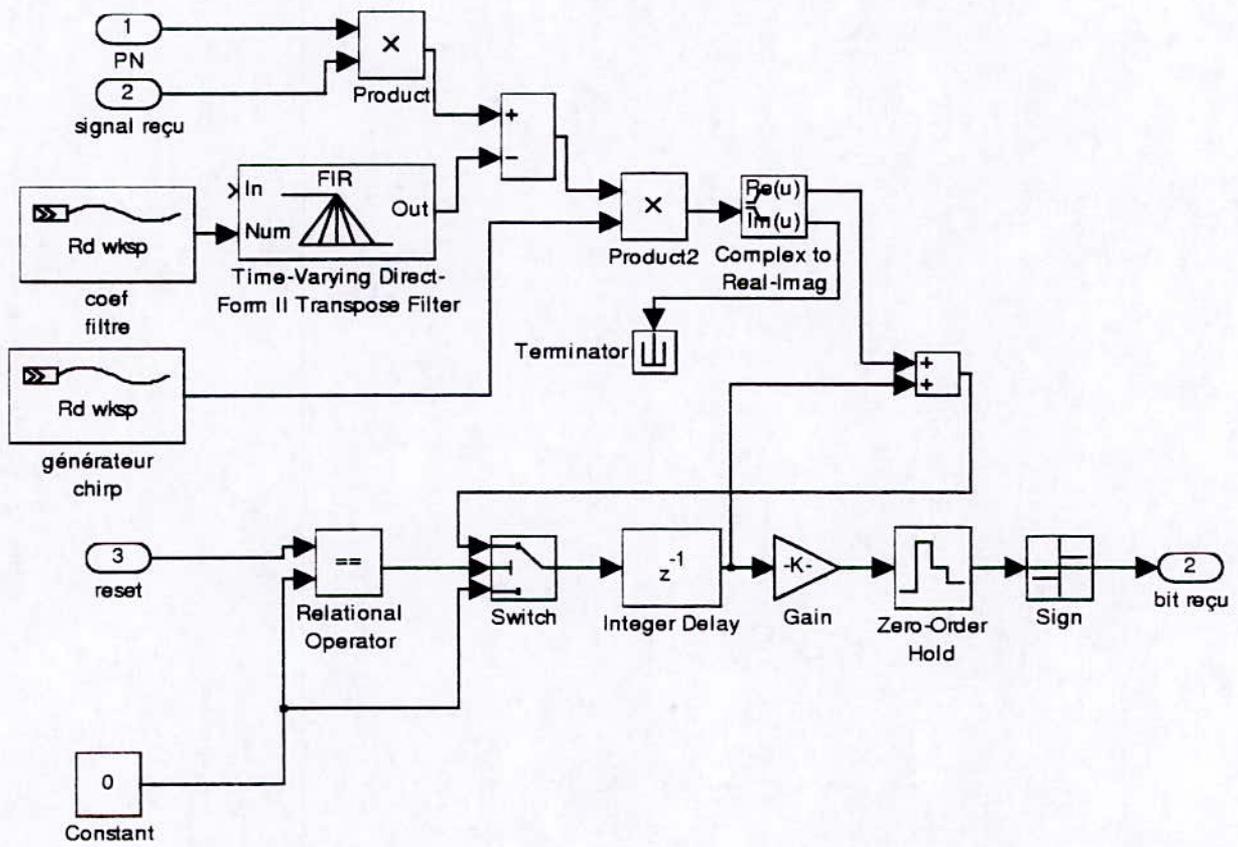


Figure C.2: demodulateur

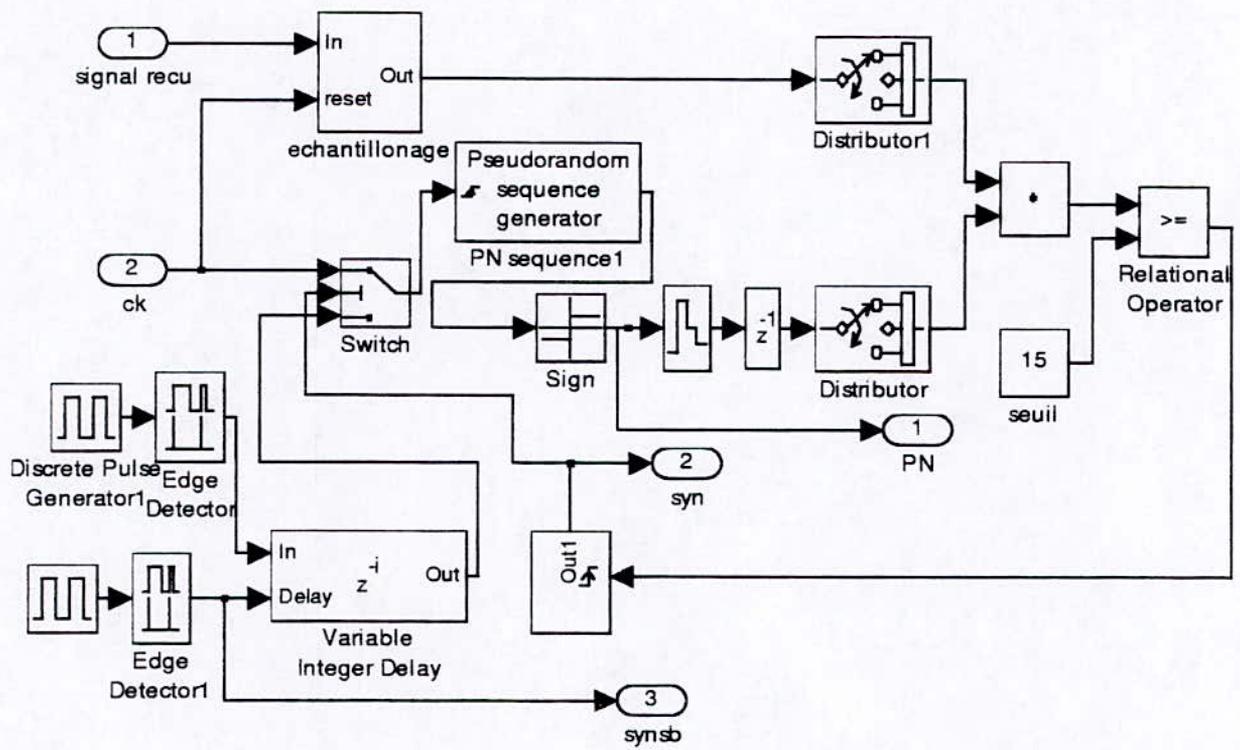


Figure C.3: bloc d'acquisition

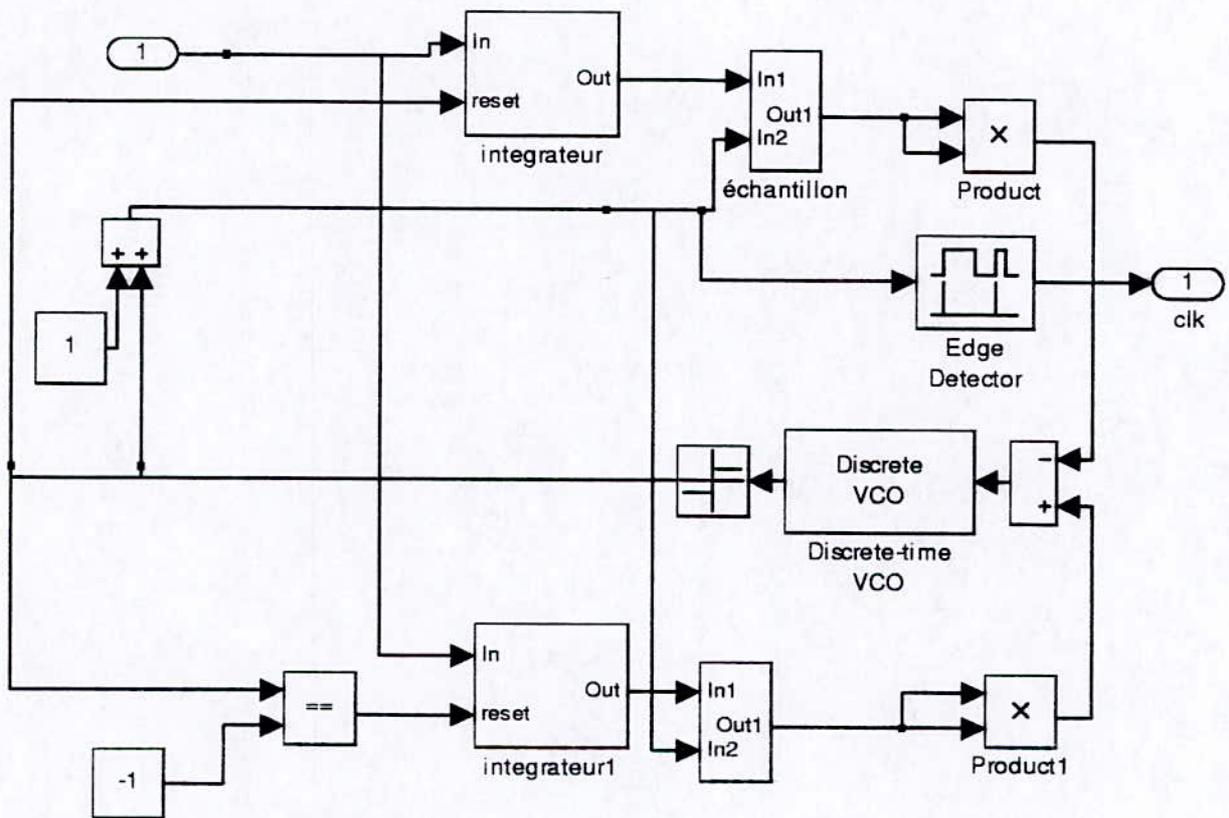


Figure C.4: bloc de tracking

# Annexe D

## Publication

- Y. Sarni, R. Sadoun and A. Belouchrani

"ON the Application of Chirp Modulation in Spread Spectrum Communication System",

Sixth International Symposium on Signal Processing and its Applications 'ISSPA'

13-16 August 2001 Kuala-Lumpur, Malaysia