

**ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC**

**MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE**

**COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE ÉLECTRIQUE
M.Ing.**

**PAR
Sacha ATWAL**

**UN SYSTÈME DE COMMUNICATION À FAIBLE PROBABILITÉ D'INTERCEPTION
BASÉ SUR LA MODULATION CHAOTIQUE**

MONTRÉAL, LE 20 DÉCEMBRE 2010

© Tous droits réservés, Sacha Atwal, 2010

PRÉSENTATION DU JURY
CE MÉMOIRE A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE

M. François Gagnon, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Claude Thibeault, codirecteur
Département de génie électrique à l'École de technologie supérieure

M. Gheorghe Marcel Gabrea, président du jury
Département de génie électrique à l'École de technologie supérieure

M. Ghyslain Gagnon, membre du jury
Département de génie électrique à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 24 NOVEMBRE 2010

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

J'aimerais remercier les professeurs François Gagnon et Claude Thibeault pour leurs aides tout au long de cette maîtrise.

UN SYSTÈME DE COMMUNICATION PEU IDENTIFIABLE BASÉ SUR LA MODULATION CHAOTIQUE

Sacha ATWAL

RÉSUMÉ

Dans un environnement militaire, il existe des situations dans lesquelles il est important d'utiliser des méthodes de communications avec faible probabilité d'interception (LPI) et de détection (LPD). Ce mémoire propose un système de communication LPI basé sur la modulation chaotique. Malheureusement, les signaux générés à partir de systèmes numériques basés sur la modulation chaotique sont cyclostationnaire [1]. Une technique d'introduction de phase aléatoire est évaluée afin de modifier les propriétés cyclostationnaires du signal modulé chaotiquement. Une analyse de la cyclostationnarité de deuxième ordre démontre que la technique d'introduction de phase aléatoire proposée réduit le degré de cyclostationnarité du signal chaotique résultant. Le modulateur chaotique est basé sur la symbolique dynamique. Le récepteur utilise un algorithme à maximum de vraisemblance afin d'estimer la séquence transmise. Une analyse des performances du système de communication en terme ("Bit Error Rate", BER) démontre une dégradation de performance de deux décibels par rapport à une modulation BPSK. De plus, afin de faire la démonstration de la faisabilité du système proposé, une analyse de l'implémentation du receveur chaotique dans un ("Field Prommable Gate Array", FPGA) est réalisée.

Mots-clés : Chaos, cyclostationnarité, Modulation, LPI

AN LPI COMMUNICATION SYSTEM BASED ON CHAOTIC MODULATION

Sacha ATWAL

ABSTRACT

In a military environment, there are situations in which it is important to use methods of communication with low probability of interception (LPI) and low probability of detection (LPD). This thesis proposes an LPI communication system based on chaotic modulation. Unfortunately, signal resulting from digital communication system based on chaotic modulation are cyclostationary [1]. A phase randomization technique is investigated in order to modify the cyclostationary properties of the chaotically modulated signal. Second-order statistical cyclostationary analysis shows that phase randomization reduce the degree of cyclostationarity of the chaotic signal. The chaotic modulation scheme is based on symbolic dynamics. The receiver estimates the transmitted sequence using the Maximum Likelihood (ML) estimation approach. The resulting bit error-rate (BER) of the proposed communication system show a performance that is two decibels lower than a BPSK modulation. In addition, the overall feasibility of such a scheme is presented by an analysis of the receiver complexity using field programmable gate array (FPGA) technologies.

Keywords: Chaos, cyclostationarity, Modulation, LPI

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 COMMUNICATION LPI/LPD	3
1.1 Communications LPI/LPD et signaux chaotiques	3
1.1.1 Résumé des caractéristiques d'un signal chaotique LPI/LPD	4
1.2 Méthodes de détection de la cyclostationnarité	6
1.2.1 Fonction de densité de corrélation spectrale	6
1.2.2 Spectre de densité de puissance de la transformation au carré du signal	9
1.3 Revue de la littérature des méthodes de suppression de la cyclostationnarité	9
1.3.1 Filtre passe bas avec largeur de bande $1/2T$	10
1.3.2 Signal avec enveloppe temporelle rectangulaire de durée T	11
1.3.3 Phase aléatoire	12
1.4 Conclusion	13
CHAPITRE 2 MODULATION CHAOTIQUE	14
2.1 Théorie du fonctionnement de la modulation chaotique	14
2.1.1 Définition du chaos	17
2.1.2 Symbolique dynamique	18
2.2 Survol des méthodes de modulation et synchronisation chaotique	19
2.2.1 Méthodes de modulation chaotique	19
2.2.2 Méthodes de synchronisation des systèmes chaotiques	22
2.3 Choix du modulateur chaotique proposé	24
2.4 Conclusion	26
CHAPITRE 3 SYSTÈME CHAOTIQUE PROPOSÉ	27
3.1 Chaîne de transmission du système chaotique	27
3.1.1 Modulateur chaotique	28
3.1.2 Introduction de phase aléatoire	31
3.1.3 Filtre de mise en forme	32
3.1.4 Démodulateur chaotique	36
3.2 Évaluation de la cyclostationnarité du signal de transmission	42
3.2.1 Résultats et analyse de la transformation au carré du signal de transmission	44
3.2.2 Résultats et analyse de la SCD du signal de transmission	48
3.3 Évaluation des performances du système de communication chaotique	51
3.3.1 Résultats et analyse des performances	52
3.4 Conclusion	52

CHAPITRE 4 ARCHITECTURE MATÉRIELLE DU DÉMODULATEUR CHAOTIQUE	54
4.1 Module série-parallèle	55
4.2 Module de calcul des métriques de branche BMU	56
4.2.1 Générateur pseudo-aléatoire	60
4.2.2 Module de filtre de mise en forme	61
4.2.3 Générateur chaotique	63
4.2.4 Module multiplication accumulation	65
4.2.5 Calcul de la métrique euclidienne	65
4.3 Module de l'Addition-Comparaison-Sélection ACS	66
4.4 Unité de gestion de l'historique	68
4.4.1 Gestion de la mémoire	69
4.4.2 Principe de décodage	71
4.5 Unité LIFO	72
4.6 Contrôle	72
4.7 Conclusion	75
CHAPITRE 5 IMPLÉMENTATION, TEST ET RÉSULTATS	76
5.1 Stratégie d'implémentation	76
5.1.1 Quantification	78
5.1.2 Plateforme de développement	78
5.2 Test et résultat	79
5.2.1 Environnement de test	79
5.2.2 Procédure de test	81
5.2.3 Résultat de co-simulation et comparaison avec modèle Matlab	82
5.2.4 Rapport d'implémentation	83
5.3 Conclusion	86
CONCLUSION	87
RECOMMANDATIONS	89
ANNEXE I RAPPORT APRÈS IMPLÉMENTATION ET ROUTAGE DU DÉMODULATEUR CHAOTIQUE ET DE SES MODULES	90
BIBLIOGRAPHIE	129

LISTE DES TABLEAUX

Page

Tableau 4.1	Initialisation des mémoires ROM	63
Tableau 5.1	Résumé des résultats après le placement et routage du démodulateur chaotique et de ses modules, ($D=4$, $M=2$, $p=0.1$)	84
Tableau 5.2	Résumé des résultats après le placement et routage du module BMU avec et sans DSP48, ($D=4$, $M=2$, $p=0.1$)	85
Tableau 5.3	Calcul du débit net du démodulateur chaotique, ($D=4$, $M=2$, $p=0.1$)	86

LISTE DES FIGURES

	Page
Figure 1.1	Implémentation de la SCD.....8
Figure 1.2	Spectre de densité de puissance de la transformation au carré du signal.....9
Figure 1.3	Modulation en amplitude avec enveloppe de durée inférieure à T.....10
Figure 1.4	Spectre de densité de puissance S_{xf}11
Figure 1.5	Modulation en amplitude avec enveloppe de durée T.12
Figure 1.6	Spectre de densité de puissance de $x(t)^2$12
Figure 2.1	Modulation chaotique.14
Figure 2.2	Exemple de fonction continue16
Figure 2.3	Diagramme de Cobweb.....16
Figure 2.4	Signal chaotique de l'orbite $0^+(0.5)$ de $f(x)$ pour les itérations 60 à 90.....17
Figure 2.5	Système de communication par masquage chaotique.....20
Figure 2.6	Système de communication CSK cohérent.....21
Figure 2.7	Système de communication de modulation chaotique basée sur la dynamique symbolique22
Figure 2.8	Principe de synchronisation identique.24
Figure 2.9	Fonction de $f(x)$25
Figure 2.10	Fonction inverse de $f(x)$26
Figure 3.1	Système de communication chaotique.....28
Figure 3.2	Fonction chaotique.....28
Figure 3.3	Séquence chaotique pour $p=0$, $p=0.5$ et $X[N]=0.65$29
Figure 3.4	Structure du modulateur chaotique.30

Figure 3.5	Fonction chaotique inverse.	30
Figure 3.6	Module de phase aléatoire.	32
Figure 3.7	Signal $x[n]$ échantillonné par un facteur de $M=5$	32
Figure 3.8	Réponse fréquentielle du filtre cosinus surélevé.	33
Figure 3.9	Réponse à l'impulsion du filtre cosinus surélevé.	34
Figure 3.10	Impulsions cosinus surélevé consécutives.	34
Figure 3.11	Mémoire induite par la phase aléatoire, $M=2$, $D=4$	36
Figure 3.12	Treillis utilisant deux états par itération de séquence chaotique.	38
Figure 3.13	Diagramme de treillis adapté pour $h(t)$ et $\Theta(t)$	39
Figure 3.14	ACS.	42
Figure 3.15	Test de la cyclostationnarité du signal.	43
Figure 3.16	Paramètres du système.	43
Figure 3.17	PSD de la transformation au carré du signal de transmission sans phase aléatoire, $M=3$	44
Figure 3.18	PSD de la transformation au carré du signal de transmission avec phase aléatoire, $M=3$	45
Figure 3.19	PSD de la transformation au carré du signal de transmission sans phase aléatoire, $M=5$	45
Figure 3.20	PSD de la transformation au carré du signal de transmission avec phase aléatoire, $M=5$	46
Figure 3.21	Module de polarité et phase aléatoire.	47
Figure 3.22	PSD de la transformation au carré du signal de transmission avec polarité et phase aléatoire, $M=3$	47
Figure 3.23	PSD de la transformation au carré du signal de transmission avec polarité et phase aléatoire, $M=5$	48
Figure 3.24	SCD sans polarité et phase aléatoire, $M=5$	49
Figure 3.25	SCD avec polarité et phase aléatoire, $M=5$	49

Figure 3.26	SCD vue alpha avec polarité et phase aléatoire, M=5	50
Figure 3.27	SCD vue alpha sans polarité et phase aléatoire, M=5.....	50
Figure 3.28	Environnement de test du système de communication chaotique.	51
Figure 3.29	Paramètres du système.	51
Figure 3.30	Eb/No (dB).....	52
Figure 4.1	Schéma bloc du démodulateur.	55
Figure 4.2	Schéma bloc(RTL) du module série-parallèle.	56
Figure 4.3	Exemple d'interrelation treillis (D=4).	57
Figure 4.4	Exemple d'initialisation treillis.	58
Figure 4.5	Diagramme bloc BMU.....	59
Figure 4.6	Diagramme bloc Branche.	60
Figure 4.7	Module générateur pseudo-aléatoire.....	61
Figure 4.8	Schéma RTL du module de filtre de mise en forme.	62
Figure 4.9	Réponse à l'impulsion h(t) (M=2 D=4).	63
Figure 4.10	Schéma RTL du modulateur chaotique, p=0.1.	64
Figure 4.11	Schéma RTL du module multiplication accumulation.	65
Figure 4.12	Schéma RTL du module distance euclidienne branche.	66
Figure 4.13	Module ACS.	67
Figure 4.14	Schéma RTL du module ACS.....	68
Figure 4.15	Flot de la mémoire du TB à quatre RAM.	69
Figure 4.16	Diagramme bloc du TB.....	70
Figure 4.17	Bloc Mux_Logic.	71
Figure 4.18	Schéma RTL du module LIFO.	72
Figure 4.19	Schéma bloc du module de contrôle.	73

Figure 4.20	Séquence des opérations des modules du décodeur.....	74
Figure 5.1	Environnement de simulation Simulink/System Generator. Test du générateur chaotique, $p=0.1$	77
Figure 5.2	Résultat de comparaison du modulateur chaotique version Matlab et System Generator.....	77
Figure 5.3	XtremeDSP Development Kit-IV.....	79
Figure 5.4	Environnement de test du démodulateur chaotique.	80
Figure 5.5	Paramètres de simulation.	81
Figure 5.6	Procédure de test.	82
Figure 5.7	Courbes BER pour simulation Matlab et co-simulation sur circuit FPGA, ($D=4$, $M=2$, $p=0.1$)	83

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ACS	Addition comparaison soustraction
BER	Bit error rate
BMU	Branch Metric Unit
BP	Bande passante
CSK	Chaos shift keying
FPGA	Field programmable gate array
FW	Featureless waveform
HIL	Hardware in the loop
HOS	Higher-order cyclostationarity statistic
LFRS	Linear feedback shift register
LIFO	Last in first out
LPD	Low probability of detection
LPI	Low probability of intercept
RAM	Random access memory
RIF	Réponse à impulsion finie
RTL	Register transfer level
SCD	Spectre de corrélation de densité
SQNR	Rapport de signal à bruit de quantification
SNR	Rapport signal à bruit
TB	Trace back

INTRODUCTION

Dans un environnement militaire, il existe des situations dans lesquelles il est important d'utiliser des méthodes de communications avec faible probabilité d'interception ("Low probability of intercept", LPI) et de détection ("Low probability of detection", LPD). Les signaux modulés chaotiquement sont non linéaires, apériodiques, non corrélés et possèdent une apparence similaire au bruit. Ces caractéristiques en font des candidats idéaux pour des communications LPI et LPD. Malheureusement, les signaux générés à partir de systèmes numériques basés sur la modulation chaotique sont cyclostationnaire [1]. Cette caractéristique cyclostationnaire des signaux chaotiques peut facilement être exploitée lorsqu'ils sont assujettis à une transformation non-linéaire de degré supérieur à un. En effet, ce type de transformation génère des composantes périodiques qui peuvent facilement être détectées lors d'analyse spectrale.

L'objectif de ce mémoire est de proposer un système de communication LPI basée sur la modulation chaotique. Le modulateur chaotique est basé sur la symbolique dynamique. Une technique d'introduction de phase aléatoire appliquée au signal chaotique du système proposé est évaluée afin de rendre le signal chaotique LPI. Il est démontré que cette technique permet d'atténuer le degré de cyclostationnarité du signal modulé chaotiquement. Au récepteur, un algorithme à maximum de vraisemblance est proposé afin d'estimer la séquence transmise. Cet algorithme est un décodeur de Viterbi adapté pour la modulation chaotique basée sur la symbolique dynamique et à la technique d'introduction de phase aléatoire proposée. Ainsi, dans ce mémoire, deux contributions y sont faites. La première contribution porte sur une technique de phase aléatoire proposée dans le but de rendre le signal chaotique LPI. La deuxième contribution porte sur la modification du décodeur de Viterbi employé par [2] pour démoduler un signal chaotique sur lequel une phase aléatoire est appliquée et dont le modulateur est basé sur la symbolique dynamique.

Afin de faire la démonstration de la faisabilité du système proposé. Le démodulateur chaotique, le module le plus complexe du système de communication, est implémenté dans un ("Field Prommable Gate Array", FPGA) de type Virtex-4. Enfin, les performances en terme de ("Bit Error Rate", BER) entre le modèle du système de communication réalisé sous le logiciel Matlab et de son implémentation matérielle sont évaluées.

Le premier chapitre passe en revue les caractéristiques nécessaires pour que des communications soient LPI/LPD en plus de définir la problématique de l'utilisation de signaux modulés chaotiquement dans un contexte LPI/LPD. Le deuxième chapitre expose la théorie du fonctionnement de la modulation chaotique. Le troisième chapitre propose un système de communication sans-fil LPI basé sur la modulation chaotique. Une analyse du degré de cyclostationarité du signal de transmission est réalisée en plus d'une évaluation des performances du système. Le quatrième chapitre présente l'architecture matérielle du démodulateur chaotique. Enfin, le cinquième et dernier chapitre évalue les performances du démodulateur chaotique implémenté dans un FPGA.

CHAPITRE 1

COMMUNICATION LPI/LPD

Ce chapitre passe en revue les caractéristiques nécessaires pour que des communications soient de faible probabilité d'interception ("Low probability of intercept", LPI) et de faible probabilité de détection ("Low probability of detection", LPD). La problématique de l'utilisation des signaux chaotiques dans un contexte LPI/LPD y est définie en plus des objectifs et hypothèses de ce projet.

1.1 Communications LPI/LPD et signaux chaotiques

Dans un environnement militaire, il existe des situations pour lesquelles il est important d'utiliser des méthodes de communications avec faible probabilité d'interception (LPI) et de détection (LPD). Par exemple, il serait néfaste qu'un receveur ennemi puisse détecter ou caractériser le signal de transmission afin de localiser le transmetteur ou d'en extraire de l'information.

Afin de bien définir la problématique, il est important de définir les termes LPI et LPD. Le terme LPD désigne un signal dont la présence est difficilement discernable de l'environnement qu'il partage avec un observateur ennemi qui possède des moyens de détection sophistiqués. Contrairement au signal LPD, un signal défini LPI est détectable. Cependant, un observateur ne peut extraire aucune information pouvant caractériser le signal transmis en termes de type, modulation, paramètres, fréquence de porteuse, fréquence de symbole, etc [3]. Ainsi, un signal LPI est un signal ne possédant aucun paramètre exploitable ("featureless waveforms", FW).

Généralement les signaux classés LPI/LPD tentent d'imiter des processus présents dans la nature comme le bruit thermique. Les caractéristiques les plus importantes sont, mais ne se limitent pas à : une distribution gaussienne avec une moyenne nulle, une auto-corrélation

impulsive, un spectre de puissance faible et uniformément distribué, une phase aléatoire, une amplitude aléatoire, et une nature apériodique.

Les signaux modulés chaotiquement sont non linéaires, apériodiques, non corrélés et possèdent une apparence similaire au bruit. Ces caractéristiques en font des candidats idéaux pour les communications LPI et LPD [4, 5]. Malheureusement, la majorité des signaux créés par l'homme possèdent des caractéristiques exploitables qui sont induites à partir de transformations périodiques comme l'échantillonnage, la modulation, le multiplexage et le codage, pour en nommer quelques-unes [6]. Ces signaux peuvent être modélisés comme étant cyclostationnaires. Bien qu'un signal cyclostationnaire ne soit pas périodique, quand il est assujéti à une transformation non linéaire de degré supérieur à un, il génère des composantes périodiques. Ces composantes périodiques résultant de transformations non linéaires peuvent être représentées par des raies spectrales et exploitées lors d'une analyse spectrale. Les signaux générés à partir de systèmes numériques basés sur la modulation chaotique sont cyclostationnaires [1]. Cette cyclostationnarité est induite par un taux de symbole constant du modulateur.

Ceci encapsule donc la problématique et justifie notre intérêt de créer une forme d'onde LPI basée sur la modulation chaotique. De plus, dans le cadre de ce projet, l'aspect LPD ne sera pas abordé étant donné qu'aucune technique d'étalement de spectre et de codage n'est utilisée afin de transmettre le signal de transmission sous le seuil du bruit thermique pour le rendre difficilement discernable de son environnement.

1.1.1 Résumé des caractéristiques d'un signal chaotique LPI/LPD

Cette section présente de manière plus détaillée les caractéristiques LPI/LPD des signaux modulés chaotiquement.

Non-linéarité

La nature non linéaire des systèmes chaotiques réduit leur vulnérabilité. Lors de l'analyse d'une séquence chaotique par un récepteur ennemi, il devient plus difficile de déduire la fonction qui permet de générer cette séquence. Par exemple, afin de rendre moins vulnérables les séquences pseudo aléatoires générées de façon linéaire, les sorties de différents étages de registre à décalage sont combinées de façon non linéaire. Ceci permet d'obtenir une séquence non linéaire dont il est plus difficile de déduire les interconnexions de rétroaction des registres[7].

Auto corrélation impulsive et faible inter corrélation

L'intercorrélation entre deux séquences tend à être faible due à la propriété de sensibilité aux conditions initiales intrinsèques aux systèmes chaotiques. Les séquences générées à partir de deux conditions initiales différentes divergent exponentiellement. De plus, grâce à cette même propriété la fonction d'autocorrélation diminue rapidement. Le spectre de fréquence associé avec cette fonction d'autocorrélation est un signal à large bande avec un spectre de puissance uniformément distribué [8].

Nature apériodique

Les conditions de sensibilité aux conditions initiales, de transitivité et de point périodique dense que doivent respecter les systèmes chaotiques les forcent à constamment visiter de nouvelles régions de leurs domaines générant ainsi de nouvelles séquences [9]. Ainsi, une séquence d'information transmise à répétition est représentée à chaque fois par une séquence chaotique différente. Les séquences générées sont apériodiques et généralement caractérisées comme ayant une apparence de bruit blanc ce qui les rend difficilement détectables par autocorrélation.

Cyclostationnarité

Tel que mentionné précédemment, les systèmes numériques basés sur la modulation chaotique sont cyclostationnaires dus à leur taux de symbole constant. Évidemment, cette caractéristique est indésirable puisqu'elle peut être exploitée par un récepteur ennemi. La suppression ou l'atténuation de cette caractéristique devient incontournable et, en fait, le but de ce projet afin d'obtenir un signal chaotique LPI. Un processus ne possédant aucune caractéristique cyclostationnaire est dit stationnaire. Le bruit thermique est un exemple de processus stationnaire.

1.2 Méthodes de détection de la cyclostationnarité

La théorie de la cyclostationnarité d'ordre deux forme la base à partir de laquelle la théorie de la cyclostationnarité d'ordre supérieur ("Higher-Order Cyclostationarity Statistic", HOCS) fut développée. Dans le cadre de ce projet, seules les propriétés cyclostationnaires de deuxième ordre sont abordées étant donné que les algorithmes utilisés par la théorie des HOCS demandent une puissance de calcul beaucoup plus importante [10].

Deux transformations quadratiques sont utilisées pour détecter et caractériser les propriétés cyclostationnaires d'ordre deux d'un signal: la fonction de densité de corrélation spectrale ("Spectral Correlation Density", SCD) et le spectre de densité de puissance de la transformation au carré du signal.

1.2.1 Fonction de densité de corrélation spectrale

La fonction SCD (Voir Équation (1.6)) est présentée utilisant une approche non probabiliste basée sur la moyenne dans le temps [10]. Un processus $x(t)$ à temps discret est dit cyclostationnaire si sa fonction d'autocorrélation est périodique (Voir Équation (1.1)).

$$R_x(t, \tau) = \langle x(t)x^*(t - \tau) \rangle \quad (1.1)$$

$$\langle \cdot \rangle \stackrel{\Delta}{=} \lim_{T \rightarrow \infty} \frac{1}{2T + 1} \sum_{t=-T}^T (\cdot) \quad (1.2)$$

Où * représente le conjugué complexe. Étant donné que la fonction d'autocorrélation est périodique, elle peut être représentée par ses coefficients de Fourier (*Voir Équation (1.3)*). Le terme $\exp[i\pi\alpha t]$ est introduit dans l'équation (1.3) afin que la théorie dans le domaine discret corresponde à la théorie dans le domaine continu [10]. L'équation (1.3) peut être réexprimée en factorisant le terme $\exp[i\pi\alpha t]$. On obtient alors l'équation (1.4) que l'on nomme fonction d'autocorrélation cyclique.

$$R_x^\alpha(\tau) = \langle x(t)x^*(t - \tau)e^{-i2\pi\alpha t} \rangle e^{i\pi\alpha t} \quad (1.3)$$

$$R_x^\alpha(\tau) = \left\langle \left[x(t)e^{-i\pi\alpha t} \right] \cdot \left[x(t - \tau)e^{+i\pi\alpha(t - \tau)} \right] \right\rangle \quad (1.4)$$

La fonction d'autocorrélation cyclique est une transformation quadratique et représente, pour toute fréquence cyclique $\alpha \neq 0$, l'intensité des composantes sinusoïdales de fréquence α résultant de l'autocorrélation. Un processus est dit stationnaire si $R_x^\alpha(\tau) = 0$ pour toute fréquence cyclique $\alpha \neq 0$. Un processus est purement cyclostationnaire d'ordre deux si la fonction d'autocorrélation cyclique est différente non-nulle, $R_x^\alpha(\tau) \neq 0$, pour tout α multiple entier d'une seule fréquence fondamentale $1/T$ et il est dit cyclostationnaire d'ordre deux dans le sens général si $R_x^\alpha(\tau) \neq 0$ pour certains α multiples entiers d'une seule fréquence fondamentale $1/T$. De plus, un processus est polycyclostationnaire d'ordre deux s'il possède plusieurs fréquences fondamentales. Ainsi, la période de cyclostationnarité peut correspondre, pour un signal modulé, au taux de symbole, à la fréquence de la porteuse, au taux de code d'étalement, etc [6].

En utilisant la relation cyclique de Wiener, on obtient la fonction de densité spectrale (Voir équation (1.5)) et l'estimation de cette fonction représentée par l'équation (1.6) où N représente le nombre d'échantillons. Cette dernière exprime la fonction SCD comme étant la corrélation entre les composantes spectrales d'un signal $x(t)$ translaté en fréquence de $f + \alpha/2$ et $f - \alpha/2$ pour $\alpha = K/T$ (Voir Figure 1.1). De plus, pour $\alpha \neq 0$ la SCD équivaut au spectre de densité spectral.

$$S_x^\alpha(f) = \sum_{\tau=-\infty}^{\infty} R_x^\alpha(\tau) e^{-i2\pi f\tau} \quad (1.5)$$

$$\tilde{S}_x^\alpha(f) = \frac{1}{N} \frac{1}{T} \sum_{n=-N/2}^{N/2} X_T(n, f + \frac{\alpha}{2}) X_T^*(n, f - \frac{\alpha}{2}) \quad (1.6)$$

$$X_T(n, f) = \int_{n-T/2}^{n+T/2} x(u) e^{-i2\pi fu} du \quad (1.7)$$

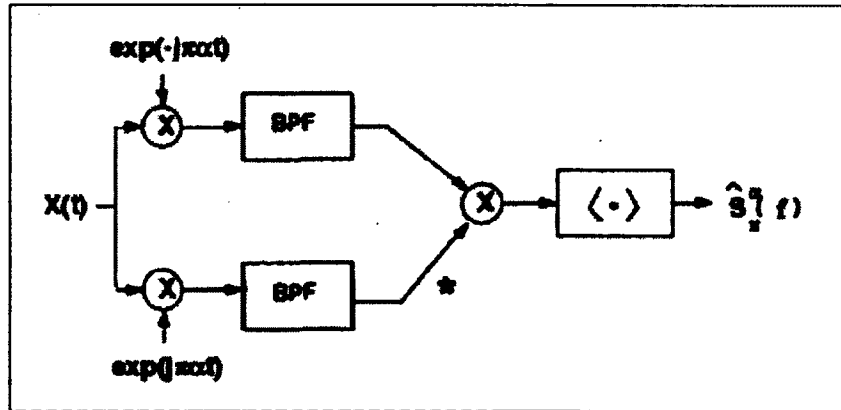


Figure 1.1 Implémentation de la SCD.
Tirée de Gardner (1994, p. 103)

1.2.2 Spectre de densité de puissance de la transformation au carré du signal

Cette transformation d'ordre deux est un cas particulier de la SCD [11]. Elle utilise les propriétés cyclostationnaires du signal reçu afin d'en extraire le taux de symbole à partir des raies spectrales générées par la transformation au carré du signal reçu (*Voir Figure 1.2*). De plus, cette transformation est utilisée par certains algorithmes de synchronisation de symbole, par exemple celui de Oerder et Meyr [12].

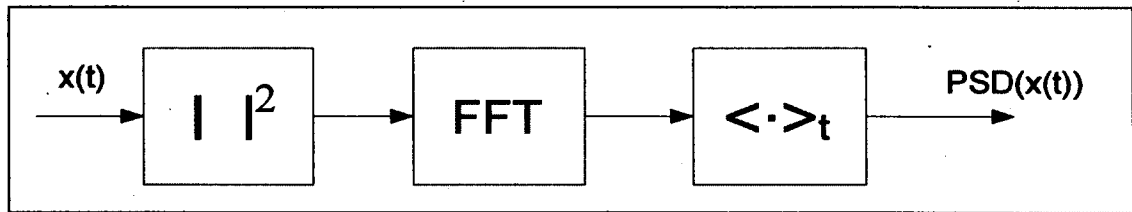


Figure 1.2 Spectre de densité de puissance de la transformation au carré du signal.

1.3 Revue de la littérature des méthodes de suppression de la cyclostationnarité.

La cyclostationnarité est une caractéristique indésirable pour les systèmes de transmissions LPI/LPD. Dû à la nature militaire des communications LPI/LPD, la documentation qui porte sur le sujet est souvent confidentielle. Ceci limite grandement l'information disponible dans ce domaine. Il suffit de penser aux techniques d'étalement de spectre qui, connues depuis les années 1940 par l'armée américaine, ont pris 60 ans avant d'être déclassifiées et utilisées pour usage commercial dans les années quatre-vingt.

La cyclostationnarité d'un signal modulé chaotiquement en bande de base est due au taux de symbole constant du modulateur [1]. Trois méthodes permettant de supprimer la cyclostationnarité fonction du taux de symbole sont présentées.

1.3.1 Filtre passe bas avec largeur de bande $1/2T$

Limiter la largeur de bande d'un signal à $1/2T$ élimine la cyclostationnarité d'ordre deux qui est fonction du taux de symbole $1/T$. La Figure 1.3 représente un signal modulé en amplitude $x(t)$ (Voir Équation (1.8)) avec une enveloppe rectangulaire de durée inférieure à T . La Figure 1.4 représente le spectre de densité de puissance $S_x(f)$ (Voir Équation (1.9)) de la transformation au carré du signal $x(t)$. Des raies spectrales à la fréquence de la fondamental et de ses harmoniques sont présentes. En utilisant un filtre rectangulaire idéal avec une largeur de bande limitée à $1/2T$ (Voir Équation (1.10)), aucun dépassement spectral n'est possible, éliminant ainsi toutes raies spectrales. Cette méthode est impossible à implémenter étant donné qu'elle se traduit par un filtre passe-bas idéal. Cependant, l'utilisation d'un filtre à cosinus surélevé qui permet de limiter la largeur de bande du signal entre $1/T \leq \beta \leq 1/2T$ en faisant varier le facteur d'arrondissement β entre $0 \leq \beta \leq 1$, permet de réduire considérablement les raies spectrales en choisissant le plus petit β permettant une implémentation physique du filtre.

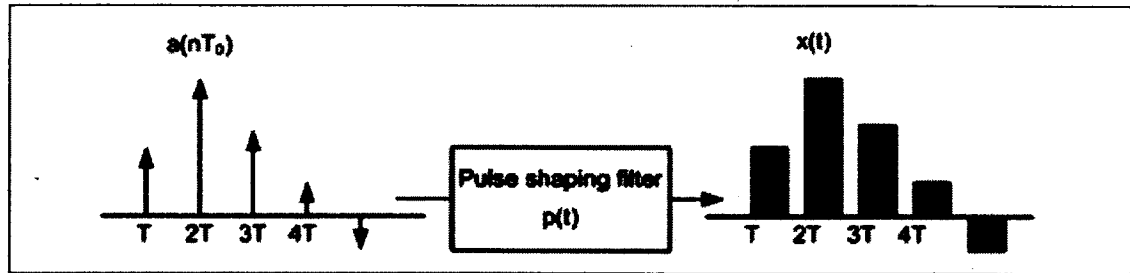


Figure 1.3 Modulation en amplitude avec enveloppe de durée inférieure à T .

$$x(t) = \sum_n a(nT) p(t - nT) \quad (1.8)$$

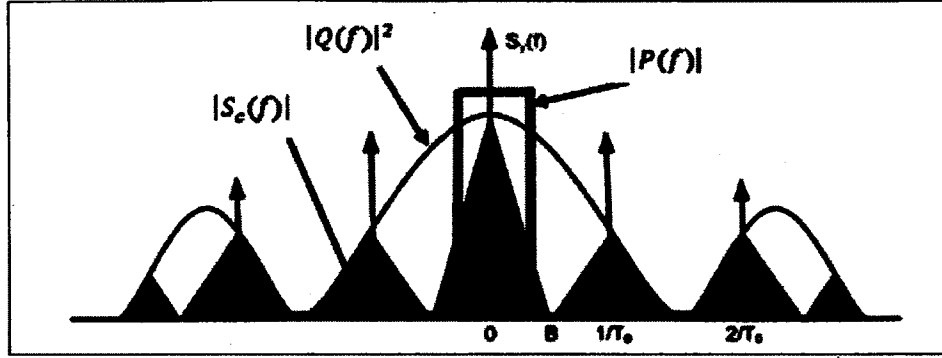


Figure 1.4 Spectre de densité de puissance $S_x(f)$.

$$S_x(f) = \frac{1}{T} |Q(f)|^2 \sum_m \left\{ K \delta \left(f - \frac{m}{T} \right) + S_c \left(f - \frac{m}{T} \right) \right\} \quad (1.9)$$

$$P(f) = \begin{cases} 1 & \text{pour } -\frac{1}{2T} \leq f \leq \frac{1}{2T} \\ 0 & \text{autre} \end{cases} \quad (1.10)$$

1.3.2 Signal avec enveloppe temporelle rectangulaire de durée T

Un signal avec une enveloppe temporelle rectangulaire de durée T , force son spectre de densité de puissance à zéro à la fréquence de symbole $1/T$ et de ses harmoniques. La Figure 1.5 représente un signal modulé en amplitude $x(t)$ avec une enveloppe rectangulaire de durée T . La Figure 1.6 représente le spectre de densité de puissance de la transformation au carré du signal $x(t)$ avec un taux de symbole de $1/T=100\text{Hz}$. Aucune raie spectrale à la fréquence de la fondamentale et de ses harmoniques n'est présente. Du point de vue de son implémentation, cette méthode est impossible à réaliser étant donné que cette forme d'onde demande une largeur de bande infinie et par le fait même cause beaucoup d'interférence entre symboles dans un canal limité en fréquence. De plus, l'interférence entre symboles dégrade les performances du système de transmission.

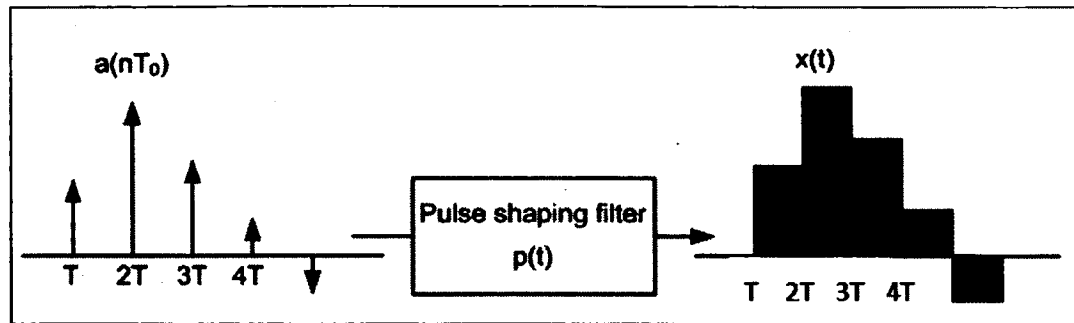


Figure 1.5 Modulation en amplitude avec enveloppe de durée T .

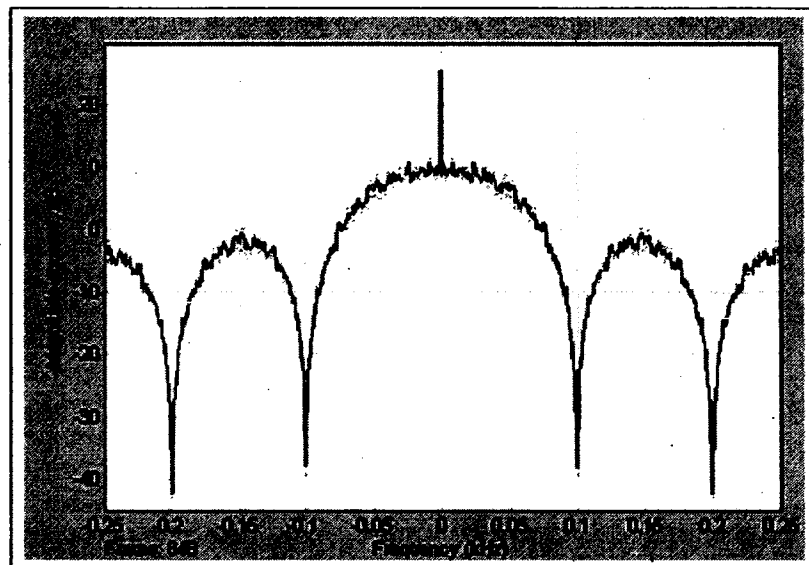


Figure 1.6 Spectre de densité de puissance de $x(t)^2$.

1.3.3 Phase aléatoire

Selon [13], la "phase randomization" est une technique utilisée afin de rendre stationnaire un processus cyclostationnaire. Ceci est accompli en introduisant une phase aléatoire θ statistiquement indépendante et uniformément distribuée sur chaque période T d'un processus $x(t)$ (Voir Équation (1.11)). La période de symbole n'étant plus constante, la cyclostationnarité fonction du taux de symbole ne peut être détectée. Cette méthode est réalisable du point de vue de son implémentation et est donc retenue dans le cadre de ce projet.

$$y(t) = x(t + \Theta) \quad (1.11)$$

1.4 Conclusion

Ce chapitre a, dans un premier temps, exposé les caractéristiques d'un système LPI et LPD nécessaire aux systèmes de communication sécurisée. Ceci a permis de définir la problématique de cyclostationarité des signaux chaotiques. Dans un deuxième temps, les techniques de détection de la cyclostationarité d'ordre deux, soit le spectre de densité de puissance de la transformation au carré du signal et la SCD ont été définies. Finalement, une revue des méthodes de suppression de la cyclostationnarité a été présentée. La méthode de phase aléatoire est retenue étant donné qu'elle est la seule à être réalisable du point de vue implémentation.

CHAPITRE 2

MODULATION CHAOTIQUE

Ce chapitre expose la théorie du fonctionnement de la modulation chaotique. Une revue des différentes méthodes de modulation chaotique les plus rencontrées dans la littérature est présentée. Le concept de synchronisation dans le domaine du chaos est abordé. Finalement, une configuration de modulateur chaotique est proposée.

2.1 Théorie du fonctionnement de la modulation chaotique

La modulation chaotique consiste à assigner les bits d'informations aux états d'un signal qui varie chaotiquement (*Voir Figure 2.1*).

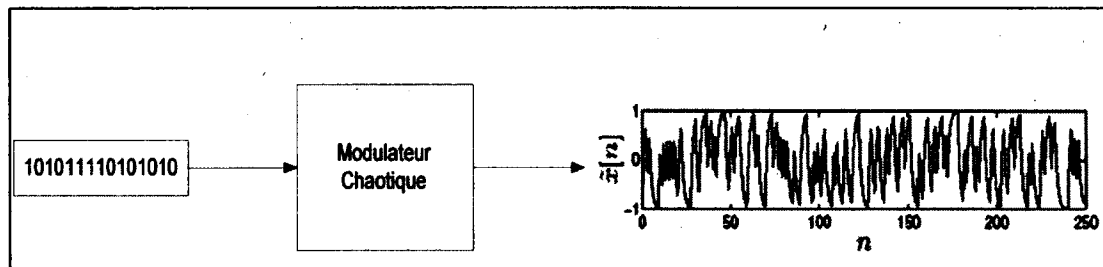


Figure 2.1 Modulation chaotique.

Une séquence chaotique est générée à partir de systèmes dynamiques non linéaires possédant un comportement pouvant sembler complètement imprévisible, même aléatoire. Ce régime chaotique est dû à un phénomène d'instabilité numérique appelé sensibilité aux conditions initiales. Un système dynamique est un système qui évolue de façon causale et déterministe [14], c'est-à-dire qu'une évolution dans le temps à partir d'une condition initiale définie une seule et unique séquence d'états futurs. L'évolution déterministe de ces systèmes peut être représentée par une équation différentielle dans le domaine continu (*Voir Équation (2.1)*) ou par une fonction de récurrence dans le domaine discret (*Voir Équation (2.2)*).

$$\frac{dx(t)}{dt} = f(x(t)) \quad x(t) \in R^m \quad (2.1)$$

$$x[n] = f(x[n-1]) \quad x[n] \in R^m \quad (2.2)$$

où x est le vecteur d'état, R^m représente l'espace des phases, soit l'ensemble de tous les états pouvant être pris par le système, et m est l'ordre du système. L'ensemble des points résultants de l'évolution d'une fonction $f(x)$ à partir d'une condition initiale pour un système dynamique discret est appelé orbite de $x[0]$ et dénoté $O^+(x[0])$ (Voir Équation (2.3)).

$$O^+(x[0]) = x[1] = f(x[0]), x[2] = f(x[1]) \dots x[n] = f(x[n-1]) \quad x[n] \in R \quad (2.3)$$

Dans le cadre de ce projet, seuls les systèmes discrets sont considérés. Les systèmes dynamiques discrets non linéaires sont représentés par des fonctions continues ou continues par morceau (Voir Figure 2.2) aussi appelé fonctions chaotiques. Le comportement de l'orbite $O^+(x[0])$ qui décrit la dynamique du système peut être représenté géométriquement sur une fonction chaotique en affichant les valeurs d'entrées-sorties en fonction de chaque itération. Cette méthode de présentation géométrique de la dynamique du système est appelée diagramme de Cobweb. Le diagramme de Cobweb de l'orbite $O^+(0.5)$ de $f(x)$ est illustré à la Figure 2.3 et le signal chaotique décrit par ce diagramme de Cobweb est illustré à la Figure 2.4.

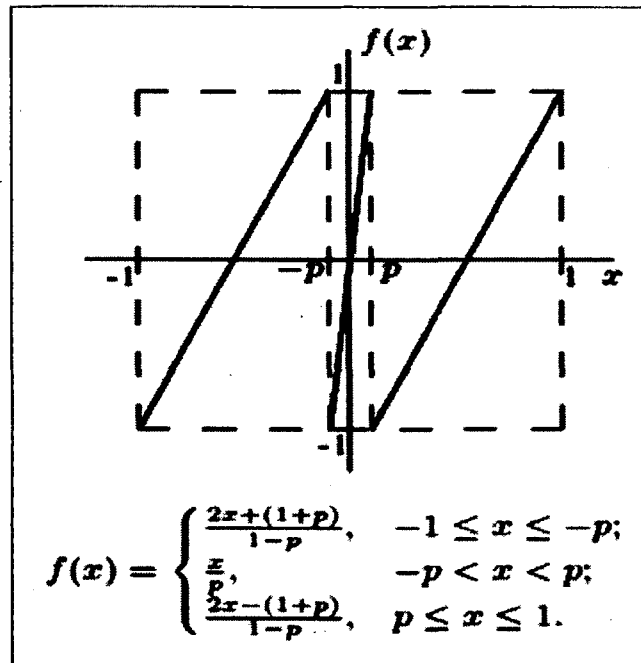


Figure 2.2 Exemple de fonction continue par morceau.

Tirée de Luengo (2005, p. 2)

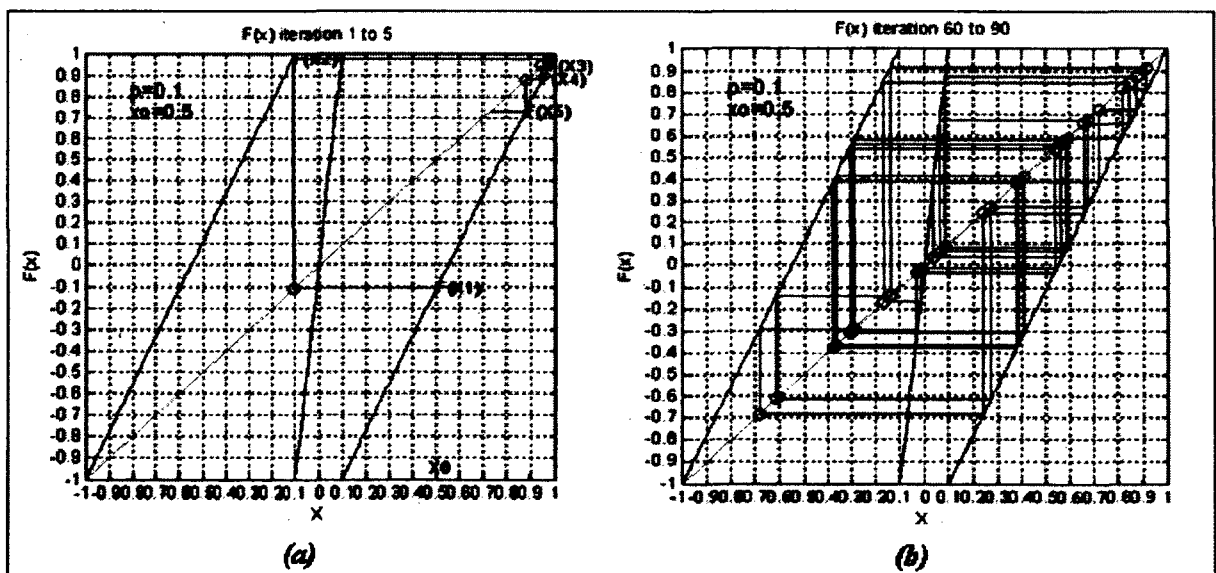


Figure 2.3 Diagramme de Cobweb a) orbite $0^+(0.5)$ de $f(x)$ pour les cinq premières itérations b) orbite $0^+(0.5)$ de $f(x)$ pour les itérations 60 à 90.

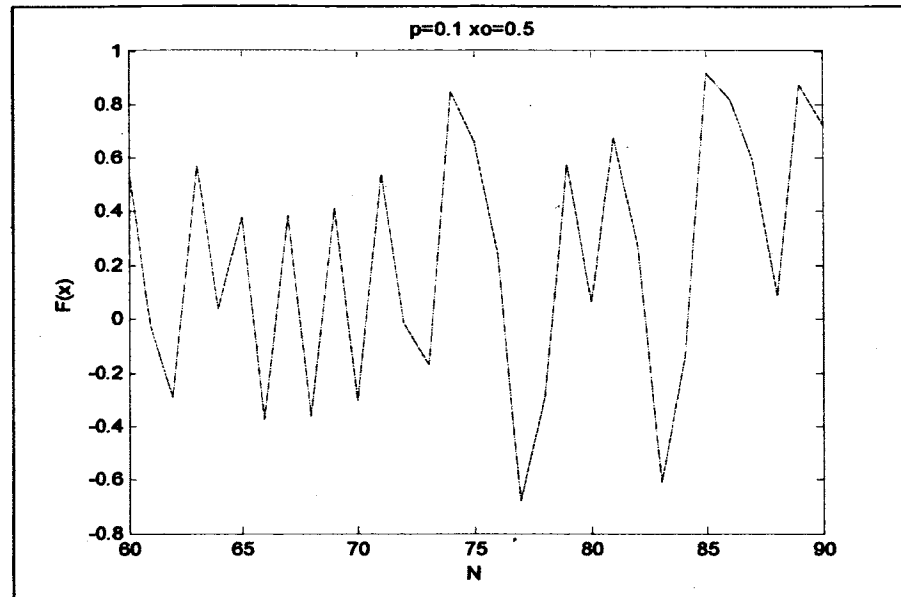


Figure 2.4 Signal chaotique de l'orbite $\theta^+(0.5)$ de $f(x)$ pour les itérations 60 à 90.

2.1.1 Définition du chaos

Les systèmes dynamiques non linéaires possèdent une variété de régimes permanents tels que stationnaire, périodique, quasi périodique et chaotique. Certaines conditions doivent être respectées afin qu'un système dynamique non linéaire soit en régime chaotique. Plusieurs définitions du chaos pour les systèmes dynamiques existent. La définition de Devaney [15] pour les systèmes discrets est l'une des plus populaires puisqu'elle s'applique à de nombreux exemples importants.

Théorème 2.1

Pour un ensemble R , $f: R \rightarrow R$ est dit chaotique sur R si

1. f est sensible aux conditions initiales, c.à.d, qu'il existe $\delta > 0$ de telle sorte que, pour n'importe lequel $x \in R$ et n'importe quel voisinage U de x , il existe $y \in U$ et $n \geq 0$ de sorte que $|f^n(x) - f^n(y)| > \delta$.

Ceci implique que deux orbites initialisées avec deux conditions initiales légèrement différentes divergent de façon exponentielle l'une de l'autre.

2. f est typologiquement transitive, c.à.d que pour n'importe quelle paire d'ensembles $V, W \subset R$ il existe $k > 0$ de telle sorte que $f^k(V) \cap W \neq \emptyset$.

Une fonction typologiquement transitive possède des points qui se déplacent d'une région de l'espace des phases vers n'importe quelle autre région de l'espace des phases.

3. les points périodiques de f sont denses en R , c.à.d que pour n'importe quel $x \in R$ et n'importe quel voisinage U de x , il existe $y \in U$ et $n \geq 0$ de telle sorte que $f^n(x) = y$.

Ceci implique que le chaos est caractérisé par de fortes instabilités localement mais qu'il reste borné globalement.

2.1.2 Symbolique dynamique

La symbolique dynamique est un outil qui permet de caractériser la dynamique des systèmes chaotiques par une séquence de symboles. L'espace des phases R est d'abord partitionné en I intervalles disjoints (*Voir Équation (2.4)*).

$$R = R_1 \cup R_2 \cup \dots \cup R_{I-1} \quad (2.4)$$

Où chaque intervalle est un homéomorphisme soit $f|_{R_i} : R_i \rightarrow f(R_i)$. En associant à chaque intervalle R_i un symbole v_i , une séquence symbolique peut être définie comme étant une séquence de régions que le signal chaotique visite durant son évolution dans le temps (*Voir*

Équation (2.5)). Ainsi, l'orbite de $x[0]$, normalement représentée par une infinité d'états possibles, peut aussi être représentée par une séquence contenant un nombre fini de symboles (Voir l'équation (2.6)).

$$S(x[n]) = v_i \leftrightarrow x[n] \in R_i \quad (2.5)$$

$$S(x[0]) = S(f(x[0])), S(f(x[1])), \dots S(f(x[n-1])) \quad (2.6)$$

2.2 Survol des méthodes de modulation et synchronisation chaotique

Dans cette section, les méthodes de modulation et de synchronisation chaotiques les plus communément retrouvées dans la littérature sont présentées. Les techniques de modulation par masquage chaotique [16], chaos shift keying (CSK) [17] et les modulations basées sur la symbolique dynamique [18] sont présentées dans un premier temps. Deuxièmement, les techniques de synchronisation chaotique les plus utilisées sont énumérées et un exemple de synchronisation par couplage est présenté afin d'illustrer le concept de la synchronisation chaotique.

2.2.1 Méthodes de modulation chaotique

Masquage chaotique

La modulation par masquage chaotique (Voir Figure 2.5) est l'une des premières méthodes appliquées aux communications chaotiques [19]. Cette technique est basée sur le principe de synchronisation par couplage entre deux générateurs chaotiques. Le transmetteur additionne le signal chaotique $c(t)$ avec un signal d'information $m(t)$, possédant une amplitude beaucoup plus petite, cachant ainsi le signal d'information.

Au récepteur, le signal reçu $r(t)$ sert à synchroniser un générateur chaotique avec celui du transmetteur. Le message d'information reçue $m(t)$ est décodé par la différence entre le signal

$r(t)$ reçu et une copie du signal chaotique $c(t)$ généré localement. L'inconvénient de ce système est la grande sensibilité au bruit de l'unité de synchronisation et de son effet sur la faible amplitude du signal d'information [20].

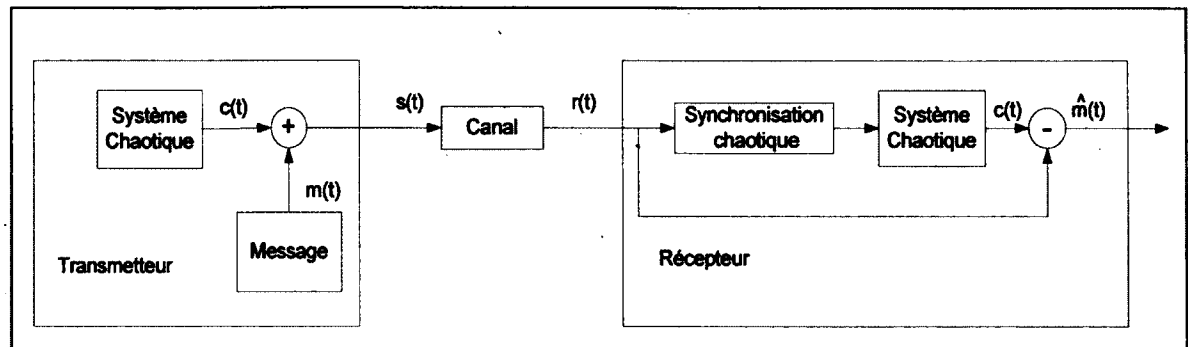


Figure 2.5 Système de communication par masquage chaotique.

Chaos shift keying

La modulation de type chaos shift keying (CSK) est une autre technique basée sur la synchronisation par couplage [21]. La Figure 2.6 montre le schéma bloc d'un système de communication CSK cohérent. Un signal d'information binaire encode les bits "0" et "1" sur deux signaux chaotiques statistiquement similaires $c1(t)$ et $c2(t)$ respectivement. Les signaux $c1(t)$ et $c2(t)$ peuvent être générés par une fonction chaotique définie par des paramètres différents résultant ainsi en deux générateurs chaotiques $f1$ et $f2$.

Le récepteur cohérent contient une copie des deux générateurs du transmetteur. Le signal reçu $r(t)$ sert à synchroniser un des deux générateurs chaotiques afin de générer les signaux chaotiques $c1(t)$ et $c2(t)$ en fonction de la transmission d'un bit "0" ou "1". Le message original est décodé soit en calculant l'erreur ou par corrélation entre le signal reçu et les signaux $\hat{c}1(t)$ et $\hat{c}2(t)$ générés localement.

Contrairement au système CSK cohérent, un système de communication CSK non cohérent n'a pas besoin de générer les séquences chaotiques localement pour démoduler l'information.

Il se base plutôt sur la différence en énergie des séquences chaotiques reçues pour distinguer les différents symboles transmis [22]. À chaque bit d'information "0" ou "1" est associée une séquence chaotique d'énergie différente $c1(t)$ et $c2(t)$. Les séquences $c1(t)$ et $c2(t)$ sont générées à partir de deux générateurs chaotiques ayant des énergies différentes ou par un générateur avec un gain d'énergie différent pour les bits "0" ou "1". L'énergie du signal reçu est estimée par un processus de corrélation intégration et le message original est décodé en comparant l'énergie de chaque symbole avec un seuil.

Bien que beaucoup plus robuste au bruit que la modulation par masquage chaotique, la modulation CSK a comme principal inconvénient sa faible sécurité étant donné qu'il serait facile de distinguer la différence en distribution de $f1$ et $f2$ pour le système cohérent et la différence en énergie pour le système non cohérent [20].

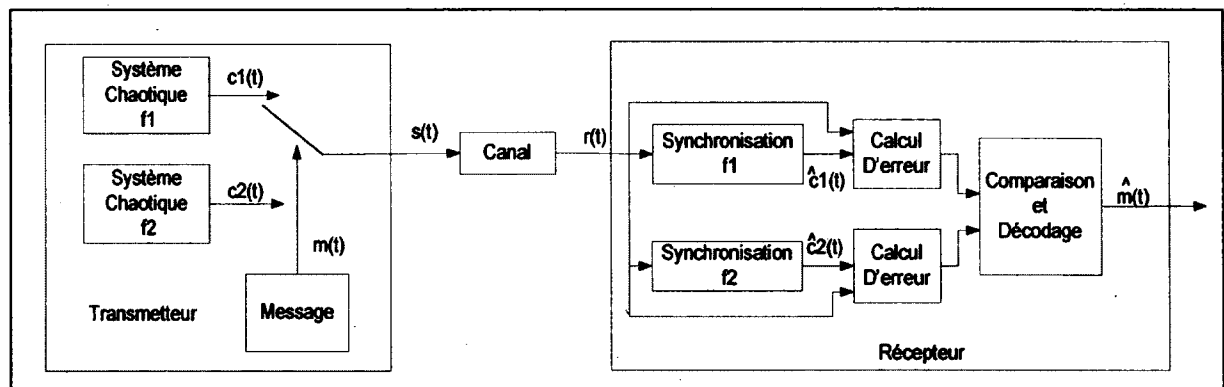


Figure 2.6 Système de communication CSK cohérent.

Modulation chaotique basée sur la dynamique symbolique

Cette modulation encode les bits d'informations sur les états d'un signal qui varie chaotiquement en utilisant la dynamique symbolique. La Figure 2.7 montre un exemple d'architecture de modulation chaotique basée sur la dynamique symbolique. Le signal d'information $s[n]$ forme une séquence symbolique. Une fonction de mappage $B(\cdot)$ établie la relation entre chaque symbole ou un regroupement de symboles et un état $x[n]$ correspondant d'une fonction chaotique.

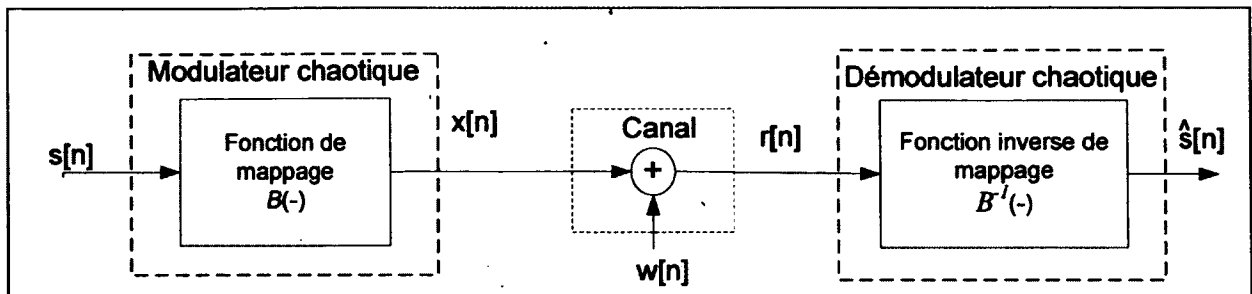


Figure 2.7 Système de communication de modulation chaotique basée sur la dynamique symbolique.

Le récepteur utilise une fonction inverse de mappage $B^{-1}(\cdot)$ afin d'associer un symbole à chaque échantillon $r[n]$ reçu. La symbolique dynamique étant une représentation plus générale de la dynamique d'un système chaotique, ceci permet au récepteur d'utiliser un algorithme à maximum de vraisemblance afin de décoder la séquence symbolique la plus vraisemblable transmise par le transmetteur [2].

2.2.2 Méthodes de synchronisation des systèmes chaotiques

Dans les systèmes de communication numérique conventionnelle, la synchronisation de porteuse sert à retrouver la phase et la fréquence de la porteuse de transmission qui est un signal périodique. Par contre, une porteuse chaotique n'est pas périodique. De plus, deux systèmes chaotiques identiques initialisés avec des conditions légèrement différentes produisent des trajectoires chaotiques qui divergent rapidement dues à leurs sensibilités aux conditions initiales. La synchronisation chaotique a pour but de synchroniser deux systèmes chaotiques afin que leurs orbites convergent asymptotiquement l'une avec l'autre lorsque t tend vers l'infini. Une fois le transmetteur et le récepteur synchronisés, la démodulation du signal chaotique est possible. Toutefois, il est à noter que la synchronisation chaotique n'est pas nécessaire pour les systèmes de communication chaotique dont le modulateur est basé sur la dynamique symbolique et utilisant une fonction chaotique inverse. Un exemple d'un tel système est proposé à la prochaine section.

Plusieurs méthodes de synchronisation chaotique existent. La plus utilisée est la méthode de synchronisation identique [23] aussi appelée synchronisation par couplage. Cette méthode a été réalisée par Pecora et Carroll [24, 25] afin de démontrer pour la première fois que deux systèmes chaotiques peuvent se synchroniser. Par la suite, plusieurs méthodes dérivées de cette dernière ont été proposées, par exemple la synchronisation généralisée [26, 27], synchronisation de phase [28, 29], synchronisation par rétroaction d'erreur [30], etc.

Toutes ces méthodes de synchronisation ont été réalisées pour des systèmes chaotiques dans le domaine continu qui, rappelons-le, sont des systèmes basés sur des équations différentielles. Afin de réaliser ces systèmes numériquement, deux méthodes d'intégration numérique soit Runge-Kutta et Euler sont les plus fréquemment utilisées. De plus, dans le domaine discret ces techniques de synchronisation peuvent être appliquées aux fonctions chaotiques.

Synchronisation identique

La technique de synchronisation identique telle que proposée par Pecora et Carroll consiste à diviser un système chaotique de n dimensions en deux sous-systèmes.

$$\dot{U}(t) = F(U(t)) \Leftrightarrow \begin{cases} \dot{X} = G(X, y_1) & (\text{dimension } m) \\ \dot{Y} = H(x_1, Y) & (\text{dimension } k) \end{cases} \quad (2.7)$$

où $U = [u_1, \dots, u_m]$

avec $X = [u_1, \dots, u_m] = [x_1, \dots, x_m]$

et $Y = [u_{m+1}, \dots, u_n] = [y_1, \dots, y_p]$

tel que $n = m + k$

Ainsi, X et Y sont les variables d'états des sous-systèmes G et H respectivement et $x_1(t)$ et $y_1(t)$ sont les variables servant à relier les deux sous-systèmes en boucle fermée. La synchronisation par couplage consiste donc à utiliser les deux sous-systèmes G et H en boucle fermée comme système maître et une copie des ces derniers en boucle ouverte comme système esclave à la réception (Voir Figure 2.8). Le signal $y_1(t)$ du système maître sert à piloter le système esclave et les trajectoires convergentes asymptotiquement (Voir Équation (2.8)).

(2.8)

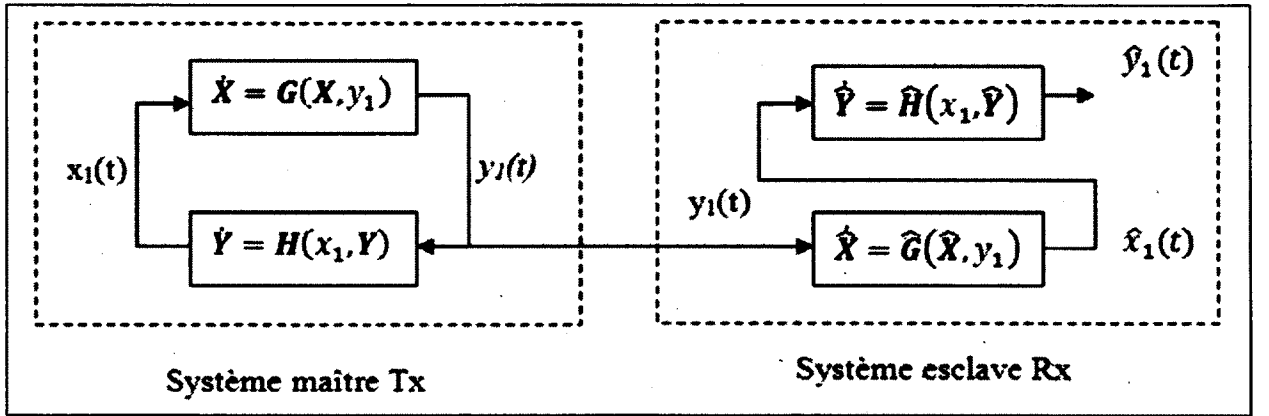


Figure 2.8 Principe de synchronisation identique.

2.3 Choix du modulateur chaotique proposé

La structure du modulateur chaotique retenue dans le cadre de ce projet est basée sur la symbolique dynamique. Plus précisément, le modulateur chaotique retenu est celui de [2] dont le fonctionnement est détaillé au chapitre 3. Ce modulateur a pour avantage d'éviter l'utilisation de techniques de synchronisation chaotiques. Ceci est possible en itérant sur la fonction chaotique inverse, itération guidée par la symbolique dynamique (Voir Équation (2.9)).

$$x[n] = f_{s[n]}^{-1}(x[n-1]) = \dots = f_{s[n], \dots, s[N-1]}^{-(N-n)}(x[N]) \quad (2.9)$$

où $f^{-1}()$ dénote la fonction inverse et $s[n]$ l'intervalle de la fonction de la prochaine itération. Pour une séquence symbolique de longueur N , la condition initiale $x[0]$ est contenue dans un intervalle défini par l'équation (2.10). Plus la longueur de la séquence symbolique augmente, plus l'intervalle qui contient la condition initiale diminue. Si N tend vers l'infini, l'intervalle contient un unique point initial [31]. De plus, il peut être démontré que pour chaque point de l'espace des phases est associée une séquence symbolique de longueur N . Ceci démontre qu'une séquence chaotique peut-être générée en partant d'un point $x[N]$ quelconque et en itérant sur l'inverse de la fonction chaotique, en utilisant la symbolique dynamique [2].

$$I = \cap_{n=0}^{N-1} f^{-n}(R_{I_n}) \quad (2.10)$$

Le fait d'itérer sur la fonction chaotique inverse (Voir Figure 2.9 et Figure 2.10) force chaque intervalle $I1$, $I2$ et $I3$ à mapper leur ensemble de départ x vers un ensemble d'arrivé y inférieur. Étant donné que la fonction inverse se contracte à chaque itération, le problème de sensibilité aux conditions initiales est éliminé.

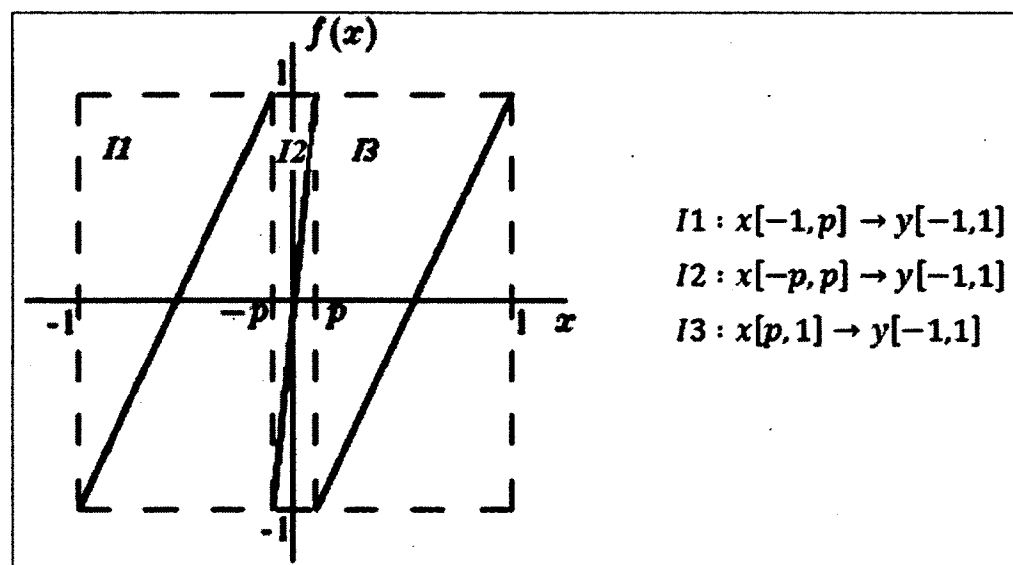


Figure 2.9 Fonction de $f(x)$.
Adaptée de Luango (2005, p. 2)

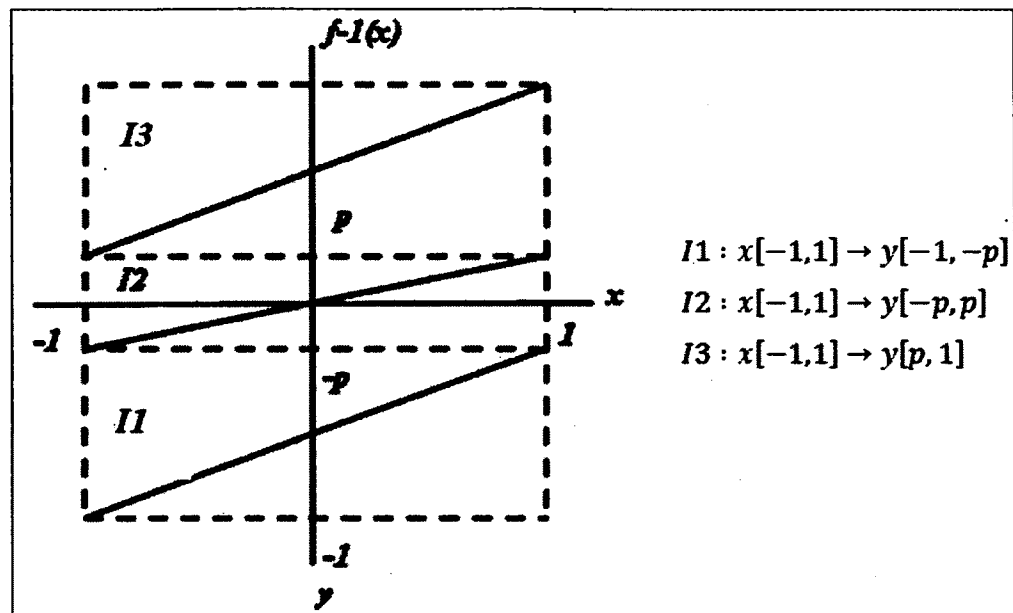


Figure 2.10 Fonction inverse de $f(x)$.

2.4 Conclusion

Dans ce chapitre, la théorie de fonctionnement de la modulation chaotique a été abordée. Parmi les différentes méthodes de modulation chaotique les plus populaires, une configuration de modulation basée sur la symbolique dynamique est retenue puisqu'elle permet d'utiliser un algorithme à maximum de vraisemblance au récepteur en plus d'éviter l'utilisation d'un module de synchronisation chaotique.

CHAPITRE 3

SYSTÈME CHAOTIQUE PROPOSÉ

Dans ce chapitre, nous proposons un système de communication sans fil LPI basé sur la modulation chaotique. Chaque module du système y est présenté en détail. Une analyse de la cyclostationarité d'ordre deux du signal de transmission est présentée. Cette analyse permet d'évaluer la méthode de phase aléatoire retenue au chapitre 1, méthode ayant pour but d'éliminer la caractéristique cyclostationnaire du signal de transmission induite par le taux de symbole constant du modulateur chaotique. Enfin, les performances du système de communication proposé sont présentées et comparées avec un système de communication BPSK à titre de référence.

3.1 Chaîne de transmission du système chaotique

Dans le présent chapitre, un système de communication chaotique est proposé tel que présenté à la Figure 3.1. Un train d'information binaire $b[n]$ est modulé sur une séquence chaotique à l'aide d'un modulateur (chaotique). Une phase aléatoire est introduite pour chaque symbole de la séquence chaotique $x[n]$ afin d'éliminer la périodicité introduite par un taux de symbole constant. Avant d'être transmis, le signal $x_p[n]$ est filtré par un filtre de mise en forme afin de limiter la bande passante du signal chaotique. Le canal de transmission ajoute un bruit blanc gaussien $w[n]$ au signal $y[n]$ transmis. À la réception, le démodulateur utilise un estimateur à maximum de vraisemblance pour décoder le train d'information original.

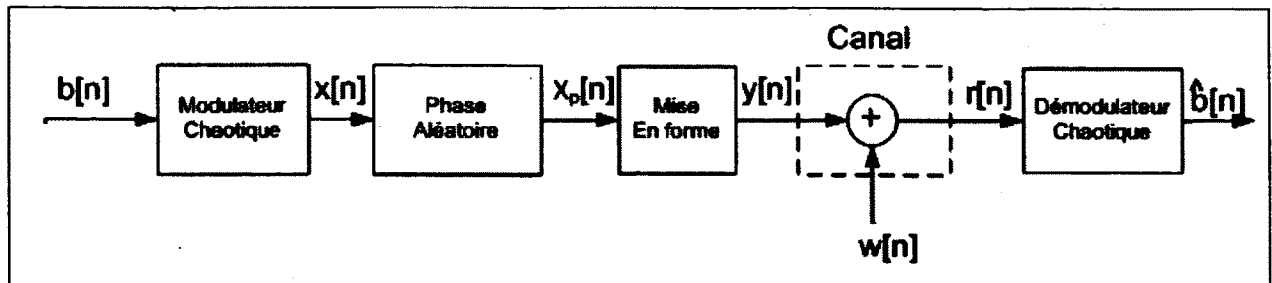


Figure 3.1 Système de communication chaotique.

3.1.1 Modulateur chaotique

Le modulateur chaotique employé dans ce système est celui de [2]. Le modulateur chaotique est basé sur la symbolique dynamique et utilise une fonction chaotique continue par morceau (Voir Figure 3.2). Un paramètre p permet de contrôler une zone tampon définie pour l'intervalle $-p \leq x \leq p$ dans laquelle aucune itération n'est réalisée.

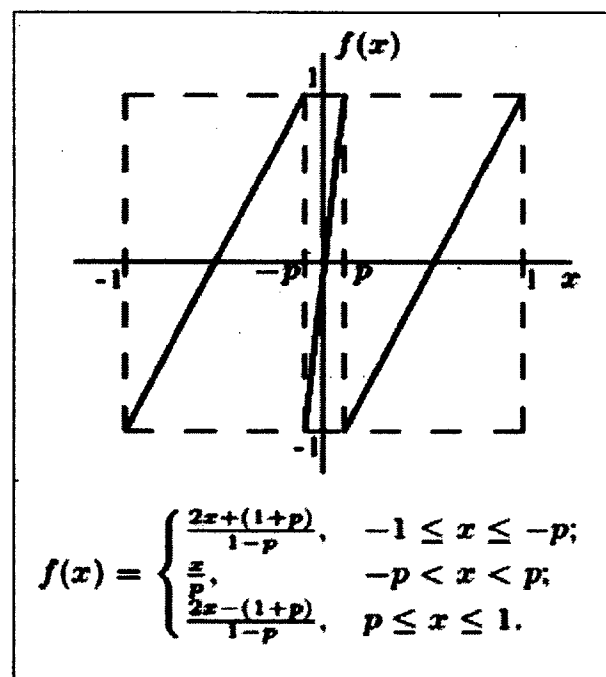


Figure 3.2 Fonction chaotique.

La zone tampon permet dans un premier temps d'assurer une distance minimum entre les symboles des deux régions I_1 et I_3 en plus de modifier le comportement de la séquence chaotique générée. Plus le paramètre p augmente, plus la distance euclidienne entre les symboles chaotiques de la séquence générée augmente, entraînant ainsi de meilleures performances du système. Il est à noter que ce gain en performance est fait au détriment d'une moins bonne immunité à l'identification du signal. En effet, plus le paramètre p augmente, plus il est facile de distinguer dans quelle région de $f(x)$ les symboles de la séquence chaotique proviennent. De plus, le vide dans la distribution d'amplitude pourrait être détecté. La Figure 3.3 illustre deux séquences chaotiques pour deux valeurs p différentes.

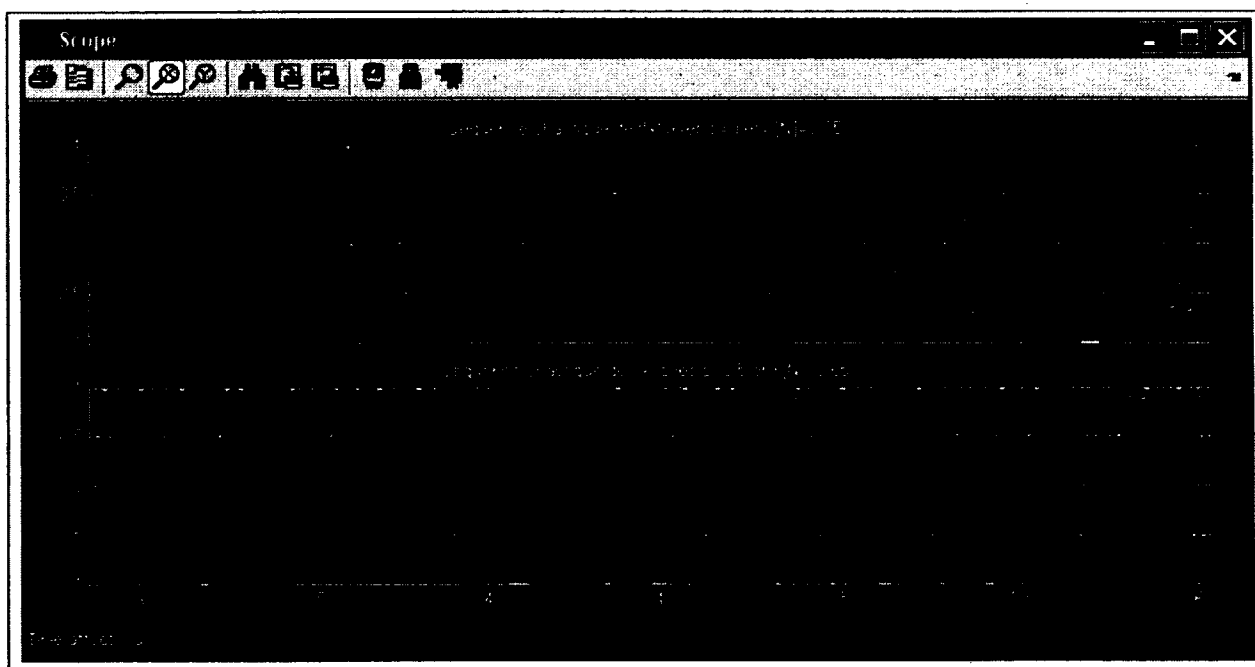


Figure 3.3 Séquence chaotique pour $p=0$, $p=0.5$ et $X[N]=0.65$.

La structure du modulateur chaotique est présentée à la Figure 3.4. Le train d'information binaire $b[n]$ forme une séquence symbolique $s[n]$ contrainte aux zones I_1 et I_3 de la fonction chaotique. La séquence chaotique $x[n]$ est générée à partir d'une condition initiale $X[N]$ en itérant sur la fonction inverse de $f(x)$ (Voir Figure 3.5) et en utilisant la séquence symbolique $s[n]$ comme itinéraire.

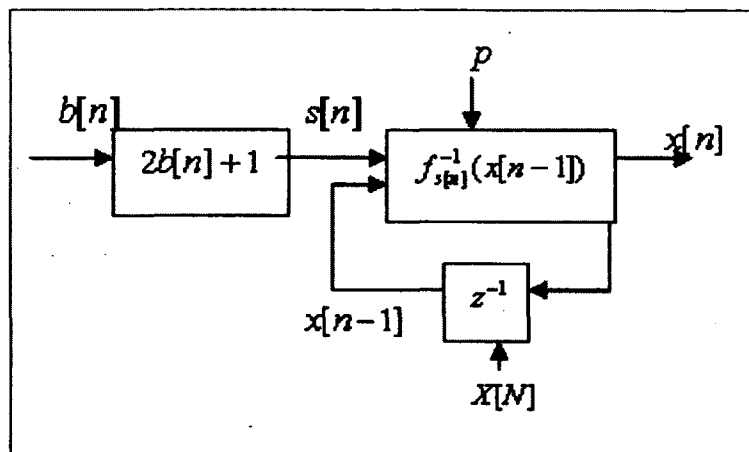


Figure 3.4 Structure du modulateur chaotique.
Tirée de Luango (2005, p. 2)

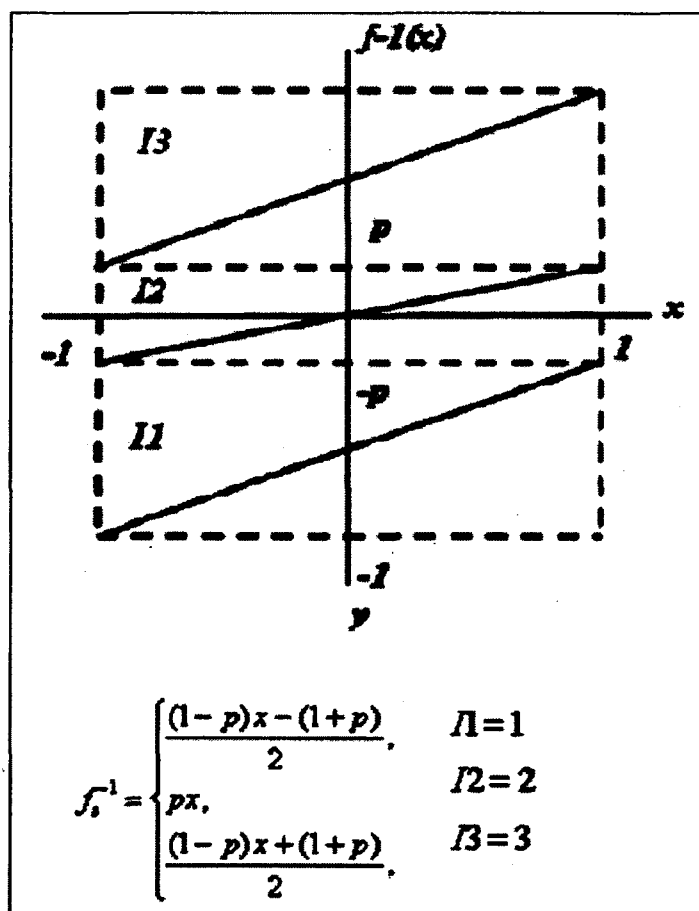


Figure 3.5 Fonction chaotique inverse.

3.1.2 Introduction de phase aléatoire

Le signal à la sortie du modulateur chaotique est généré avec une période de symbole constante le rendant ainsi cyclostationnaire. Dans cette section, une méthode de phase aléatoire est proposée.

Sachant qu'un suréchantillonnage doit être accompli afin que le signal $x(t)$ puisse être mis en forme pour la transmission, une phase aléatoire Θ est introduite en effectuant un échantillonnage à des instants choisis aléatoirement à l'intérieur d'une période de symboles (Voir Équation (3.1)) et (3.2). Le signal peut alors être modélisé par l'équation (3.3) qui représente un signal $x(t)$ échantillonné par un train d'impulsion auquel une phase aléatoire est introduite à chaque période de symbole T_0 .

$$x_p(t) = x(t)p(t) \quad (3.1)$$

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_0 + \Theta(n)) \quad (3.2)$$

$$x_p(t) = \sum_{n=-\infty}^{\infty} x(nT_0 + \Theta(n))\delta(t - nT_0 + \Theta(n)) \quad \text{où } \Theta < T_0 \quad (3.3)$$

La structure du module d'introduction de phase aléatoire proposé est représentée à la Figure 3.6. Le signal est suréchantillonné par un facteur M au minimum deux fois supérieur au taux de symbole du signal d'entrée. Un générateur de nombres aléatoires permet d'introduire un délai de Θ échantillons, où $0 < \Theta < M$, permettant ainsi de sélectionner l'une des M phases possibles d'un symbole suréchantillonné (Voir Figure 3.7).

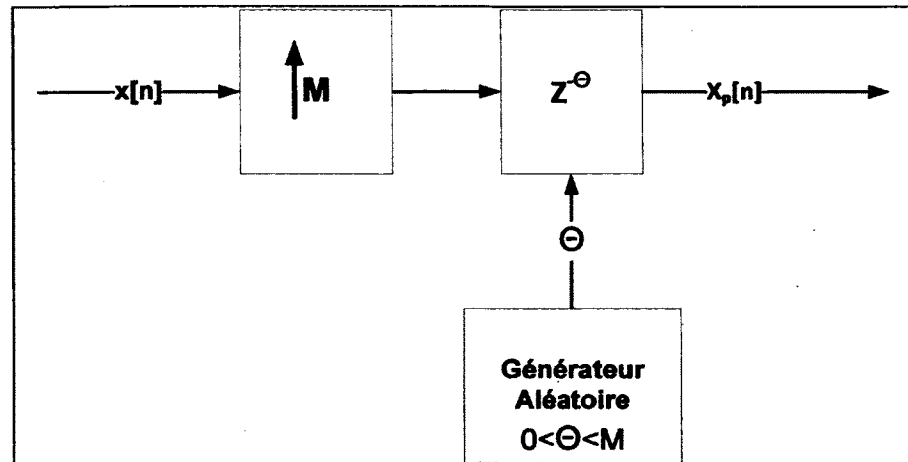


Figure 3.6 Module de phase aléatoire.

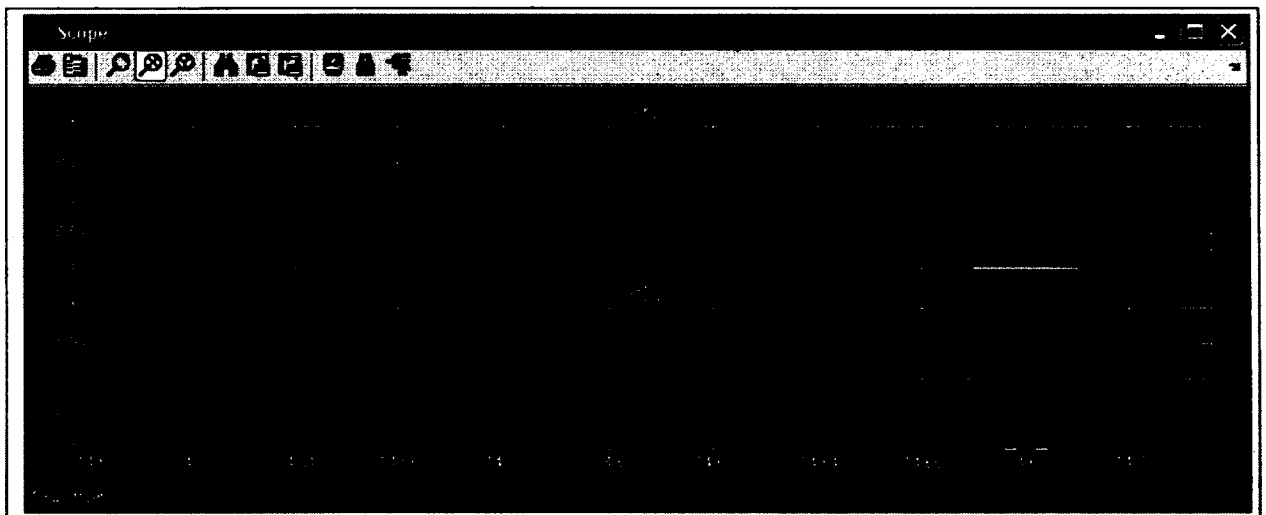


Figure 3.7 Signal $x[n]$ échantillonné par un facteur de $M=5$.

3.1.3 Filtre de mise en forme

Généralement, les systèmes de transmission sans fil doivent opérer dans des bandes de fréquences qui leur sont allouées. Ces contraintes sont souvent imposées par des règles gouvernementales ou de designs. Il est donc important de limiter la bande passante du signal transmis. Le filtre de mise en forme le plus souvent utilisé est le filtre en cosinus surélevé

(Voir Figure 3.8). La réponse fréquentielle et la bande passante (BP) du filtre en cosinus surélevé sont définies par les équations (3.4) et (3.5).

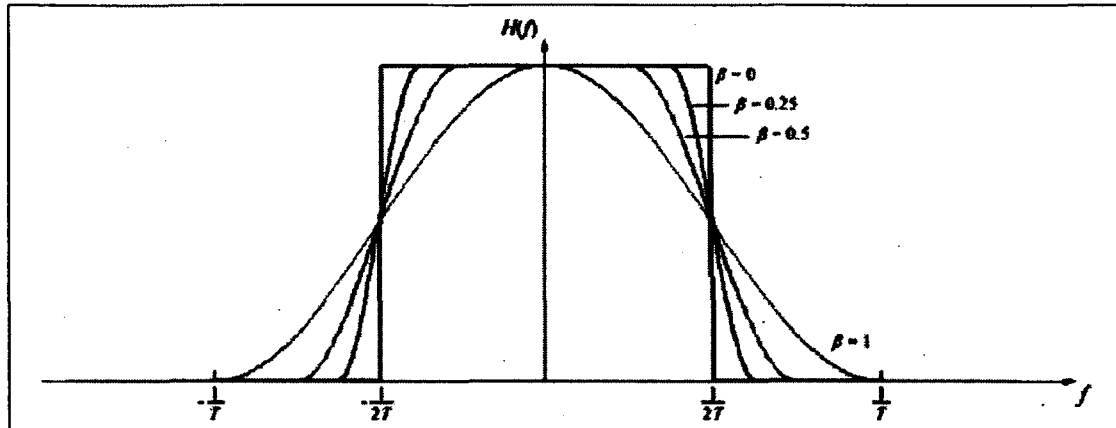


Figure 3.8 Réponse fréquentielle du filtre cosinus surélevé.
Tirée de Wikipedia (2010)

$$H(f) = \begin{cases} T_0, & |f| \leq \frac{1-\beta}{2T_0} \\ \frac{T_0}{2} \left[1 + \cos \left(\frac{\pi T_0}{\beta} \left[|f| - \frac{1-\beta}{2T_0} \right] \right) \right], & \frac{1-\beta}{2T_0} < |f| \leq \frac{1+\beta}{2T_0} \\ 0, & \text{autre} \end{cases} \quad (3.4)$$

$$BP = \frac{1}{2T_0} (1 + \beta) \quad (3.5)$$

où T_0 représente la période de symbole et β représente le facteur d'arrondissement compris entre $0 \leq \beta \leq 1$ permettant de limiter la BP entre la fréquence de Nyquist $1/(2T_0)$ et le taux de symbole $1/T_0$.

Pour le système proposé, le filtre de mise en forme est implémenté à l'aide d'un filtre numérique à réponse à l'impulsion finie (RIF). Les coefficients sont définis par la réponse à l'impulsion du cosinus surélevé représenté par l'équation (3.6). Comme l'illustre la Figure 3.9, la réponse à l'impulsion $h(t)$ s'étend sur D symboles et passe à zéro à chaque période de

symbole T_0 . Cette dernière caractéristique permet d'éliminer l'interférence inter-symbole à chaque période T_0 (Voir Figure 3.10).

$$h(t) = \text{sinc}\left(\frac{t}{T_0}\right) \frac{\cos\left(\frac{\pi\beta t}{T_0}\right)}{1 - \frac{4\beta^2 t^2}{T_0^2}} \quad (3.6)$$

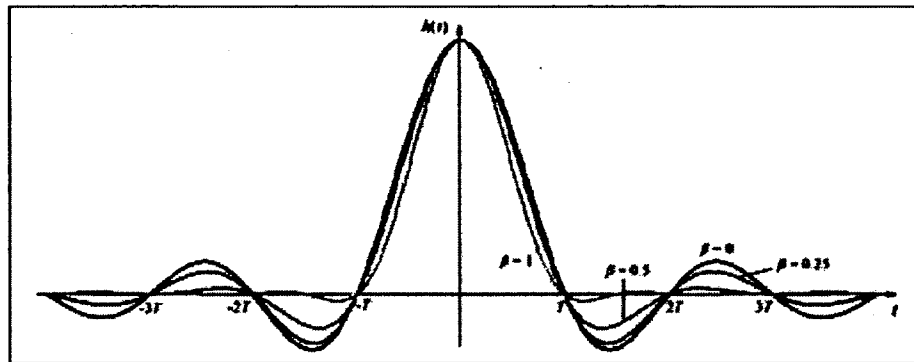


Figure 3.9 Réponse à l'impulsion du filtre cosinus surélevé.
Tirée de Wikipedia (2010)

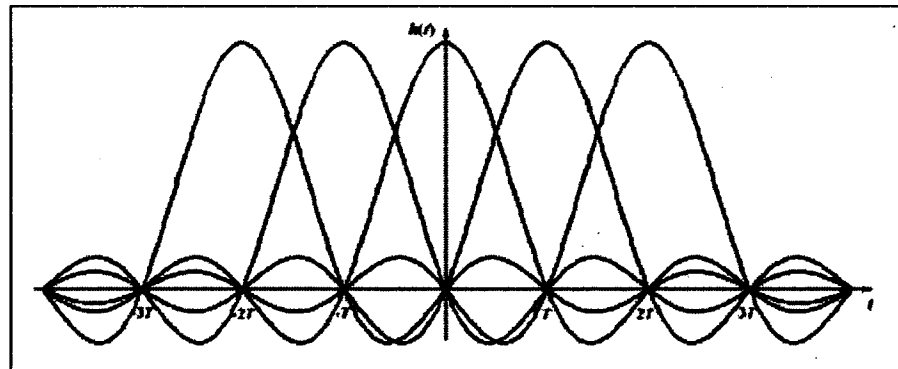


Figure 3.10 Impulsions cosinus surélevé consécutives.
Tirée de Wikipedia (2010)

Le degré avec lequel la réponse à l'impulsion $h(t)$ est approximée est fonction du nombre N de coefficients utilisés. Généralement N est un multiple entier du facteur M de suréchantillonnage (Voir Équation (3.7)). Ceci permet d'obtenir un nombre N qui couvre un nombre entier de symboles D .

$$N = D * M \quad (3.7)$$

Évidemment, plus D et M sont grand, meilleure est l'approximation de la réponse idéale. Cependant, D est proportionnel à la complexité du filtre qui souvent est un facteur que l'on veut minimiser. En résumé, la réalisation d'un filtre numérique en cosinus surélevé pour une réponse fréquentielle donnée dépend du facteur d'arrondissement β , du nombre de symboles D que couvre la réponse à l'impulsion, du facteur de suréchantillonnage M et de l'ordre du filtre soit le nombre de coefficients N . Cependant, il est à noter que le facteur de suréchantillonnage a peu d'influence sur les performances du filtre [33].

La réponse à l'impulsion étant définie, le signal chaotique à la sortie du filtre en cosinus surélevé est modélisé par les équations (3.8) et (3.9). Tel qu'illustré à la Figure 3.11, la phase aléatoire Θ introduite par $p(t)$, induit une mémoire sur $D+1$ symboles. Chaque symbole de M échantillons subi l'interférence de $D/2$ symboles passés et $D/2$ symboles futurs.

$$y(t) = [x(t)p(t)] \otimes h(t) \quad (3.8)$$

$$y(n) = \sum_{n=-\infty}^{\infty} x(nT_0 + \Theta(n)) h(t - nT_0 + \Theta(n)) \quad (3.9)$$

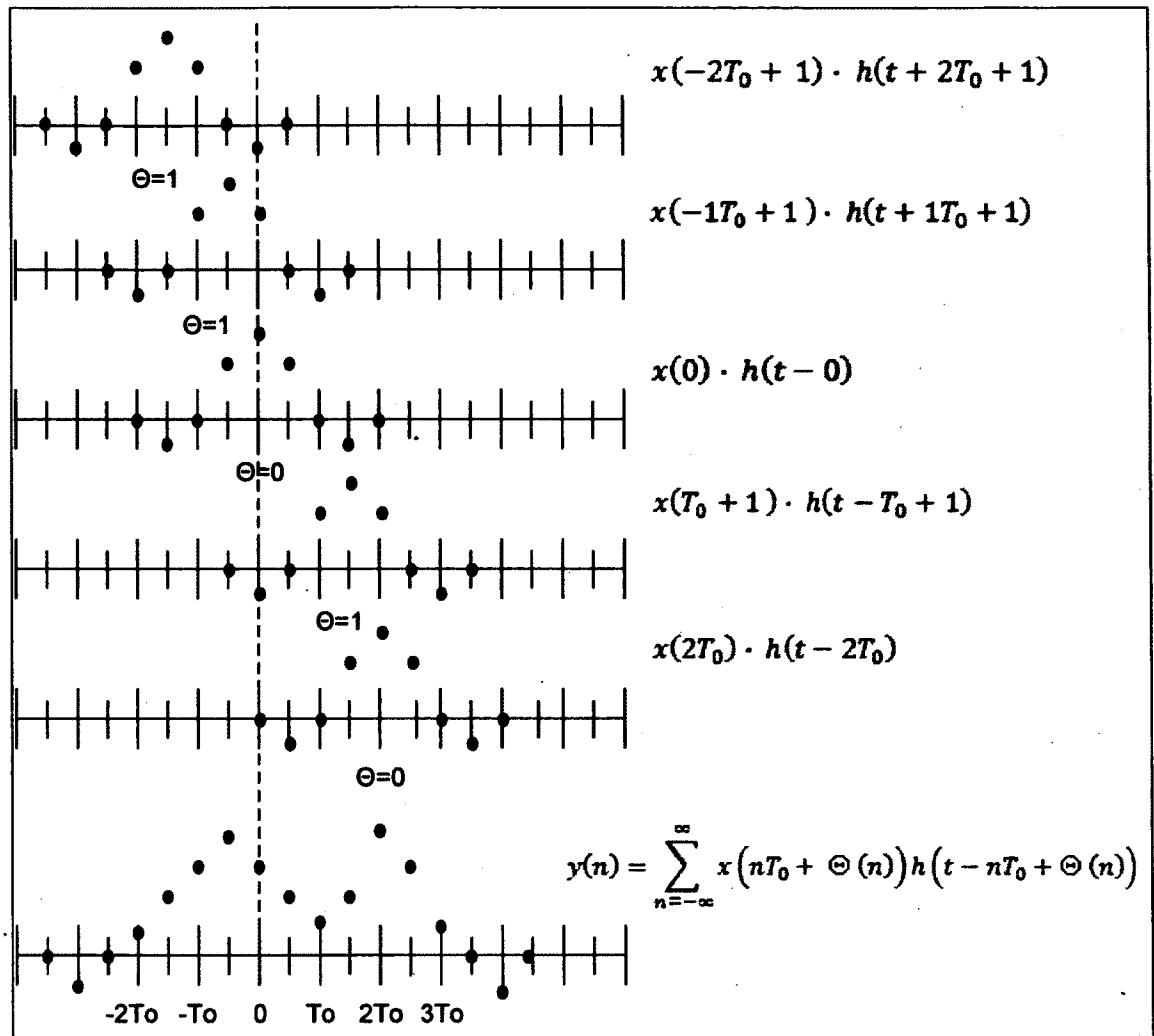


Figure 3.11 Mémoire induite par la phase aléatoire, $M=2$, $D=4$.

3.1.4 Démodulateur chaotique

Représentation par diagramme de treillis

Le concept de base du démodulateur chaotique repose sur la représentation en diagramme de treillis de la trajectoire de la séquence chaotique. Selon [2], un treillis avec deux états peut être utilisé. Soit un état pour chaque région I_1 et I_3 de la fonction chaotique utilisée par le modulateur présenté à la section précédente. Ce concept est justifié par le fait que la fonction chaotique $f^{-1}(x)$ employée se contracte à chaque itération. Ainsi, les bits précédents d'une

séquence symbolique ont peu d'importance sur l'estimation des bits futurs. Ceci est démontré par l'exemple suivant.

Considérons deux échantillons différents, $x_1[n]$ et $x_2[n]$ où $0 \leq n \leq N$, qui partagent un même itinéraire $s_1[n] = s_2[n] = s[n]$ sur la fonction chaotique $f^{-1}(x)$. Les échantillons futurs $x_1[n+1]$ et $x_2[n+1]$ sont donnés par les équations (3.10) et (3.11) respectivement.

$$x_1[n+1] = \frac{(1-p)x_1[n] + (s[n+1]-2)(1+p)}{2} \quad (3.10)$$

$$x_2[n+1] = \frac{(1-p)x_2[n] + (s[n+1]-2)(1+p)}{2} \quad (3.11)$$

La distance entre deux échantillons après une itération peut alors être calculée avec l'équation (3.12). Le résultat indique qu'indépendamment de leur position, la distance entre deux échantillons après une itération diminue par un facteur de $(1-p)/2$.

$$d[n+1] = |x_2[n+1] - x_1[n+1]| = \frac{1-p}{2} d[n] \quad \text{où } d[n] = |x_2[n] - x_1[n]| \quad (3.12)$$

Ainsi, il est démontré que la distance entre les deux séquences diminue de façon exponentielle après k itérations (Voir équation (3.13)). Ceci prouve donc que les bits passés d'une séquence symbolique ont peu d'influence sur les échantillons futurs d'une séquence chaotique et justifie l'utilisation d'un treillis à deux états pour représenter l'itinéraire du système chaotique.

$$d[n+k] = \left(\frac{1-p}{2} \right)^k d[n] \quad (3.13)$$

La Figure 3.12 illustre le treillis à deux états auquel l'algorithme de Viterbi est appliqué. La métrique de branche $c_{ij}[n]$ pour une branche j partant d'un état i est représentée par l'équation (3.14). La métrique d'état C_i est représentée par l'équation (3.15). Le terme $r[n]$

représente l'échantillon reçu et $\hat{x}_i[n]$ représente l'échantillon chaotique de l'itération précédente généré par la métrique de branche $c_{ij}[n]$ ayant le plus petit cumul avec la métrique d'état correspondante. L'échantillon $\hat{x}_i[n + 1]$ représente donc l'échantillon chaotique le plus vraisemblable entre deux branches du treillis menant au même état.

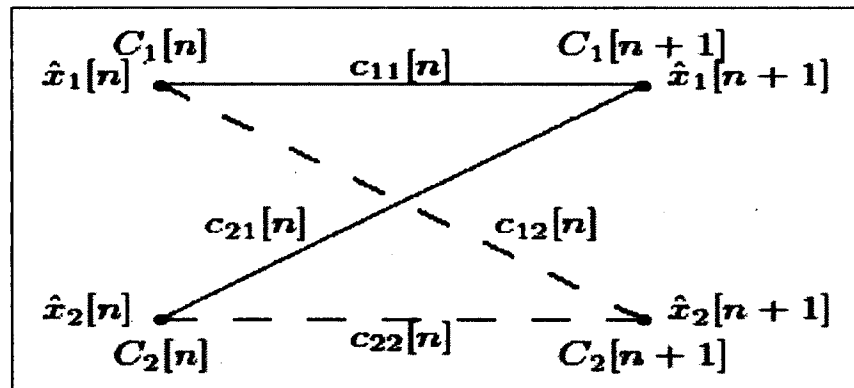


Figure 3.12 Treillis utilisant deux états par itération de séquence chaotique.

Tirée de Luango (2005, p. 3)

$$c_{ij}[n] = |r[n] - f_j^{-1}(\hat{x}_i[n])| \quad (3.14)$$

$$C_i[n + 1] = \min_{j=1,2} \{C_j[n] + c_{ji}[n]\} \quad (3.15)$$

Expansion du diagramme de treillis

Afin d'estimer le signal $y(t)$ à la sortie du transmetteur, le diagramme de treillis est modifié pour accommoder la mémoire induite par le filtre de mise en forme ainsi que la séquence de gigue de phase $\theta(t)$ du transmetteur. Le nombre d'états du treillis résultant, pour un filtre de mise en forme pour lequel la réponse à l'impulsion couvre D symboles, augmente à 2^D états (Voir Figure 3.13). Les D bits de chaque état représentent un segment possible de la séquence symbolique $S[k]$ transmise. Étant donné que la séquence symbolique est formée de valeurs binaires assignées aux régions $I1$ et $I3$ de la fonction chaotique, seulement deux branches de transition entrent et quittent chaque état i . Chaque branche j du treillis représente le prochain

symbole possible de la séquence symbolique. La transition entre l'état présent et le prochain est donc déterminée par la dynamique symbolique. Le diagramme du treillis permet ainsi de représenter n'importe quelle séquence symbolique par un chemin unique constitué de branches successives.

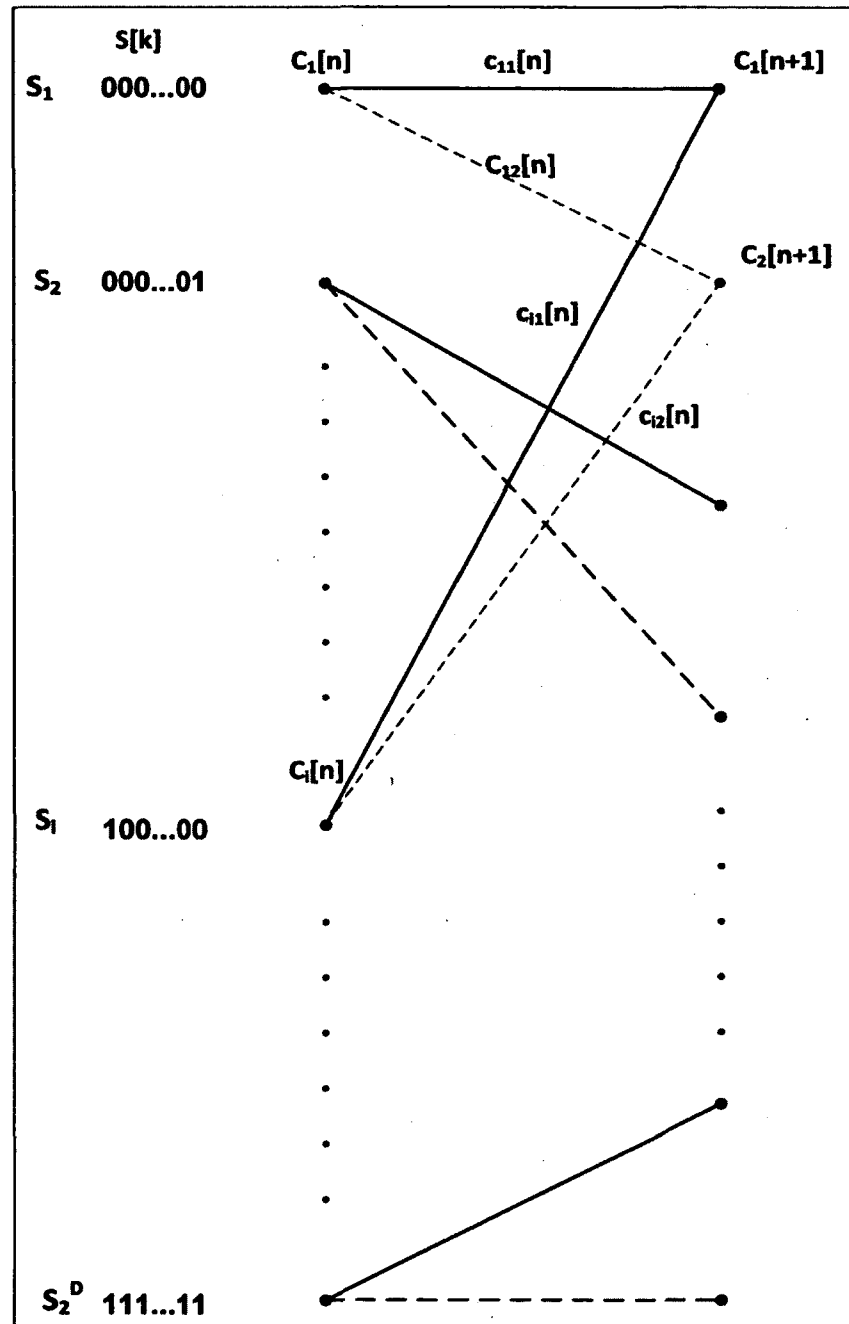


Figure 3.13 Diagramme de treillis adapté pour $h(t)$ et $\Theta(t)$.

Ainsi pour chacune des branches du treillis, une séquence chaotique $x_{ij}[m]$ (Voir Équation (3.16)) est générée à l'aide de la séquence symbolique $sd[m]$ (Voir Équation (3.17)). Connaissant la fonction $f^{-1}(x)$ du modulateur, $h(t)$ du filtre de mise en forme et $\Theta(t)$ de la séquence de phase aléatoire du transmetteur, chaque branche du treillis permet de générer un symbole $y_{ij}[n]$ (Voir Équation (3.18)) de M échantillons ayant subi l'interférence de $D/2$ symboles passés et futurs afin d'estimer l'échantillon reçu $r[n]$. La métrique de branche est maintenant exprimée par l'équation (3.19) et la métrique d'état reste telle que présentée par l'équation (3.15).

$$x_{ij}[m] = f_{sd[m]}^{-1}(x_{ij}[m-1]) \quad (3.16)$$

$$sd[m] = [s[k], j] \quad (3.17)$$

$$y_{ij}[n] = \sum_{m=1}^{D+1} \left(x_{ij}(mT_0 + \Theta(m)) h(t - mT_0 + \Theta(m)) \right) \quad (3.18)$$

$$c_{ij}[n] = \sum_M |r[n] - y_{ij}[n]| \quad (3.19)$$

L'algorithme de Viterbi

L'algorithme de Viterbi, un algorithme à maximum de vraisemblance, est utilisé afin de sélectionner la séquence la plus vraisemblable du treillis. Soit Y une séquence de M échantillons d'un symbole transmis dans un canal gaussien de variance σ^2 , et X une séquence de M échantillons d'une des branches du treillis. De plus, on dénote les échantillons de Y par r_m et X par c_{jm} où l'index m indique le m^{th} échantillons et j indique la branche j du treillis.

Selon [7], une métrique de branche pour une branche j d'un chemin U du treillis est définie par la fonction log-vraisemblance (Voir Équation (3.20)).

$$c_j^u = \log P(Y/X_j^u) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{[r_m - c_{jm}^u]^2}{2\sigma^2} \right\} \quad (3.20)$$

Si on simplifie et on néglige tout les termes communs aux métriques de branche du treillis on obtient l'équation (3.21).

$$c_j^u = \log P(Y/X_j^u) = \sum_{m=1}^M [r_m - c_{jm}^u]^2 \quad (3.21)$$

Ainsi, pour un signal à une dimension, la métrique de branche est la distance absolue au carré entre le symbole reçu et celui correspondant à la branche j d'un chemin U du treillis. De plus, maximiser la probabilité d'une décision correcte revient à minimiser la distance absolue entre un symbole reçu Y et un symbole Z de la branche donnée.

Une métrique d'état pour un état i , au temps n d'une séquence de N symboles, est définie par le cumul le plus faible des métriques de branche d'un chemin U menant à cet état. La métrique d'état est calculée récursivement à partir de la métrique d'état de l'itération précédente (*Voir Équation (3.22)*).

$$C_i[n] = \min_j \{C_j[n-1] + c_{ji}[n]\} \quad (3.22)$$

où $C_j[n-1]$ est la métrique minimum de l'état i à l'instant $n-1$ et $c_{ji}[n]$ est la branche j menant à l'état i . Cette opération est appelée "Add-Compare-Select" (ACS) (*Voir Figure 3.14*).

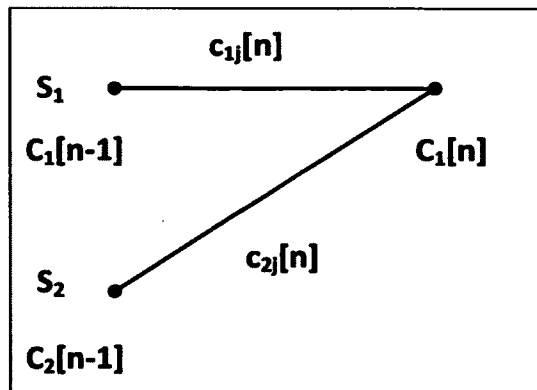


Figure 3.14 ACS.

La métrique d'état finale $C_i[N]$, aussi appelée métrique de chemin, est la plus petite métrique de tous les chemins U qui entrent dans cet état. Ainsi, la séquence la plus vraisemblable du treillis est la séquence avec la plus petite métrique de chemin.

3.2 Évaluation de la cyclostationnarité du signal de transmission

L'évaluation de la cyclostationnarité du signal de transmission est effectuée à l'aide de deux transformations quadratiques soit la fonction de densité de corrélation spectrale ("Spectral Correlation Density", SCD) et la transformation au carré du signal de transmission telle que vue au chapitre un. L'évaluation est réalisée dans l'environnement *Simulink* de *Matlab* (Voir Figure 3.15). La Figure 3.16 résume les paramètres de simulation. L'analyse est réalisée pour des facteurs M de suréchantillonnage de trois et cinq choisis arbitrairement.

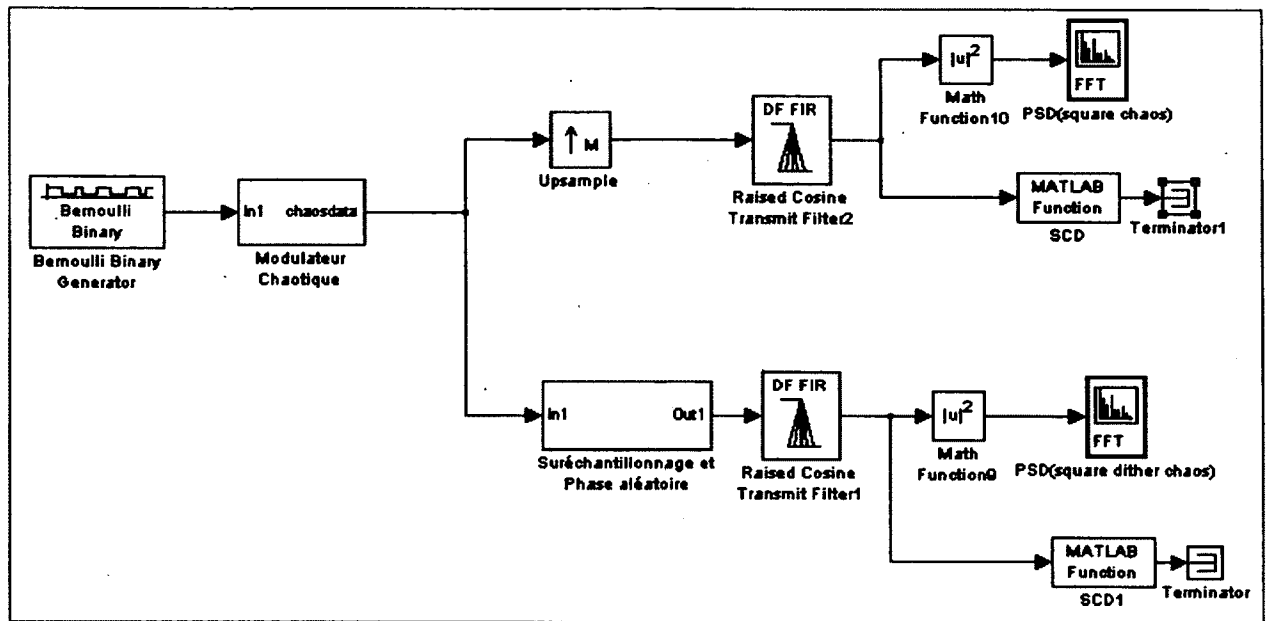


Figure 3.15 Test de la cyclostationnarité du signal.

Function Block Parameters: Paramètres Système

Subsystem (mask):

Parameters:

Fréquence Tx (Hz):

Modulateur chaotique (Paramètre p):

Facteur de suréchantillonnage (M):

Facteur D'arrondissement (B):

Durée de l'impulsion (D symboles):

Longueur FFT:

Nombre de moyenne spectral:

OK Cancel Help Apply

Figure 3.16 Paramètres du système.

3.2.1 Résultats et analyse de la transformation au carré du signal de transmission

Pour les facteurs de suréchantillonnage de trois et cinq, avec et sans phase aléatoire, les résultats sont présentés aux Figure 3.17, Figure 3.18, Figure 3.19 et Figure 3.20 respectivement. Dans les deux cas, une faible atténuation des raies spectrales à la fréquence de symbole du modulateur chaotique est observée.

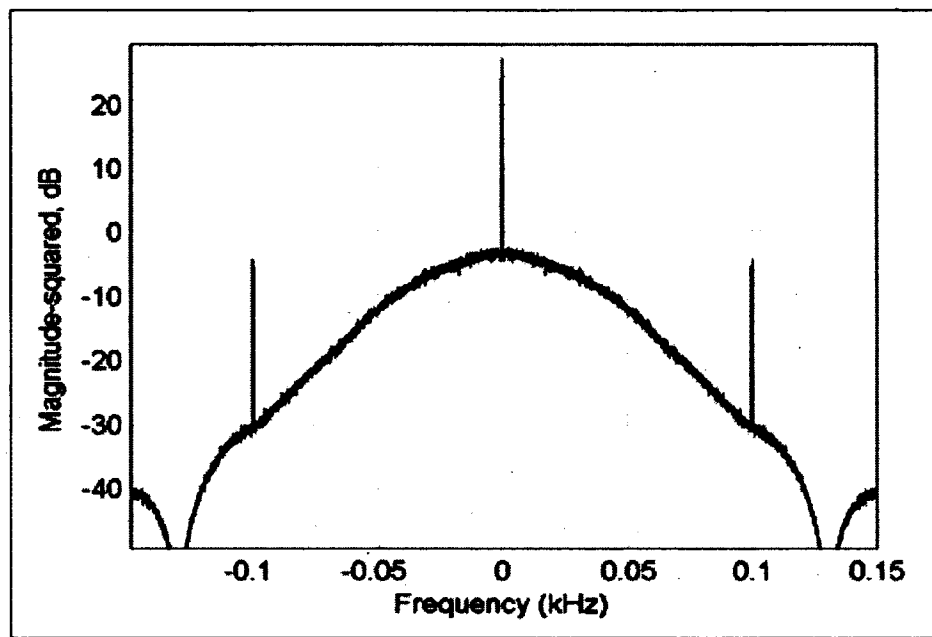


Figure 3.17 PSD de la transformation au carré du signal de transmission sans phase aléatoire, $M=3$.

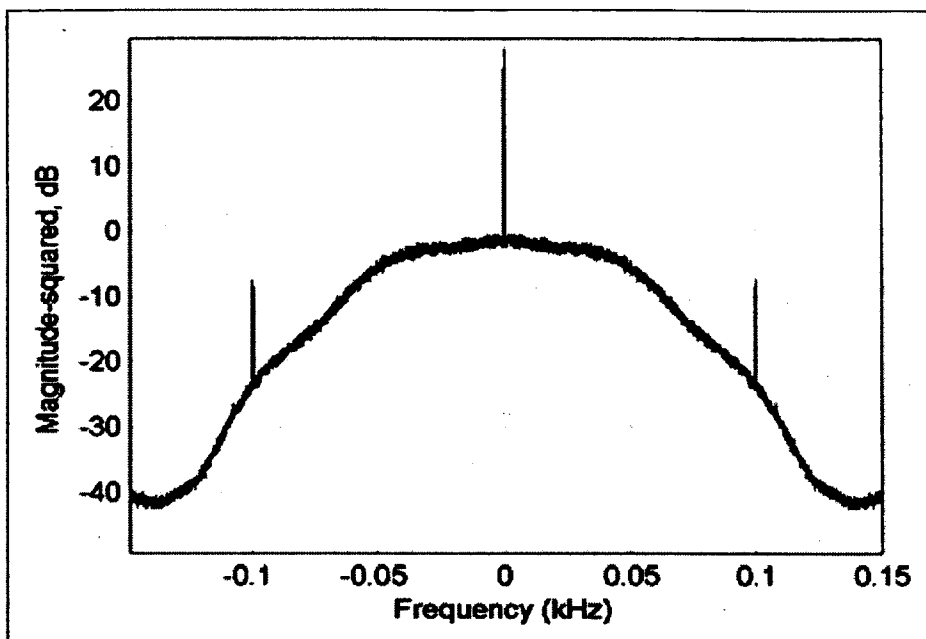


Figure 3.18 PSD de la transformation au carré du signal de transmission avec phase aléatoire, $M=3$.

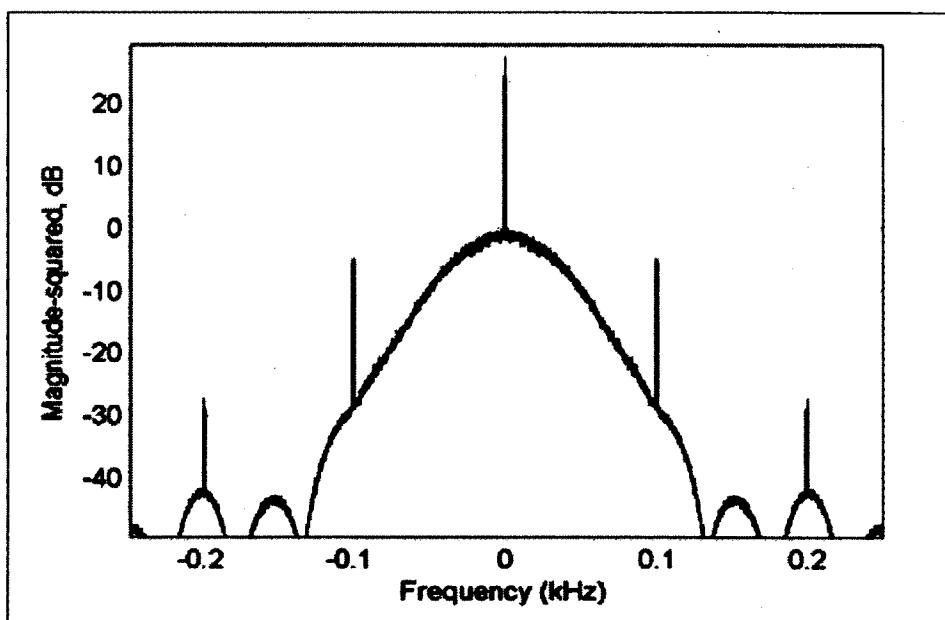


Figure 3.19 PSD de la transformation au carré du signal de transmission sans phase aléatoire, $M=5$.

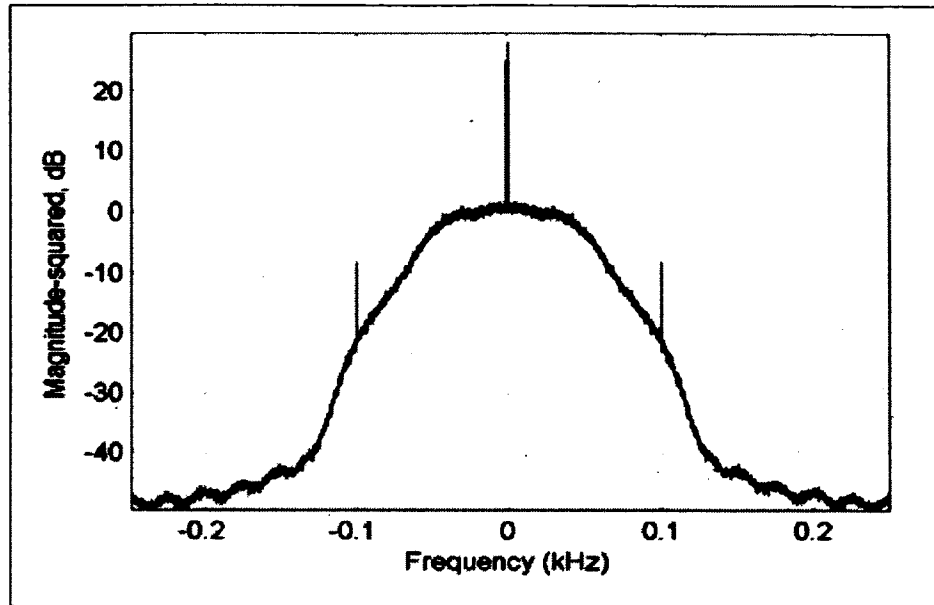


Figure 3.20 PSD de la transformation au carré du signal de transmission avec phase aléatoire, $M=5$.

Après analyse des résultats, une hypothèse suggère que l'impossibilité d'atténuer complètement les raies spectrales à la fréquence de symbole vient du fait que la fonction continue par morceau $f^{-1}()$ employée induit une caractéristique cyclostationnaire fonction du taux de symbole. Cette caractéristique cyclostationnaire est introduite par un taux de variation en amplitude constant. Cette variation en amplitude constante est due aux pentes linéaires de la fonction $f^{-1}()$. Afin d'éliminer cette périodicité, le même principe employé pour éliminer la cyclostationnarité induite par un taux de symbole constant est appliqué en amplitude. Comme illustrée pour la Figure 3.21, une séquence aléatoire permet de varier aléatoirement la polarité de la séquence chaotique $x[n]$ générée par le modulateur chaotique. Le module de phase aléatoire permet ensuite d'éliminer la cyclostationnarité induite par le taux de symbole constant.

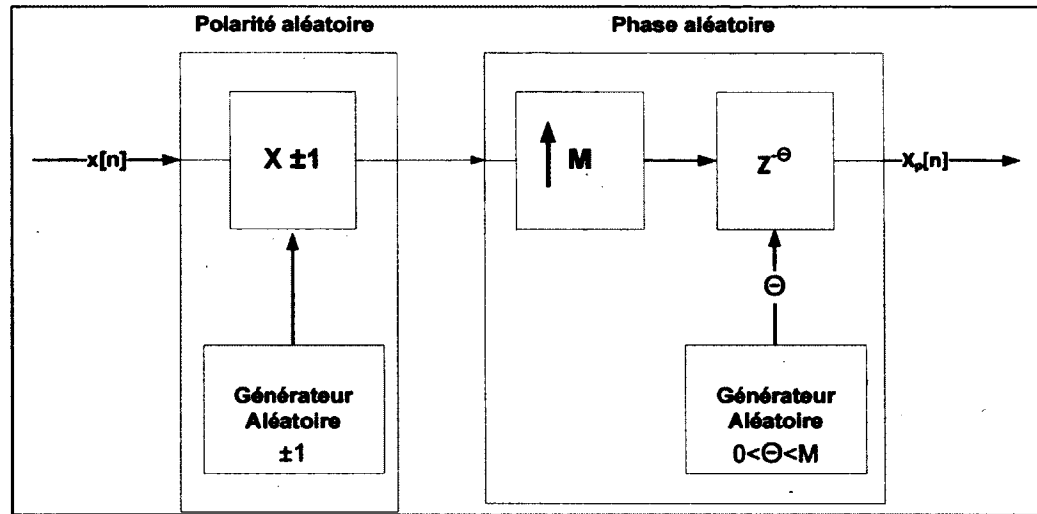


Figure 3.21 Module de polarité et phase aléatoire.

Les Figure 3.22 et Figure 3.23, représentent le spectre de densité spectrale de la transformation au carré du signal de transmission pour lequel une polarité et phase aléatoire sont appliquées. Pour les facteurs de suréchantillonnage de trois et cinq respectivement, il est possible de constater que les caractéristiques cyclostationnaires qui sont fonction du taux de symbole sont complètement éliminées.

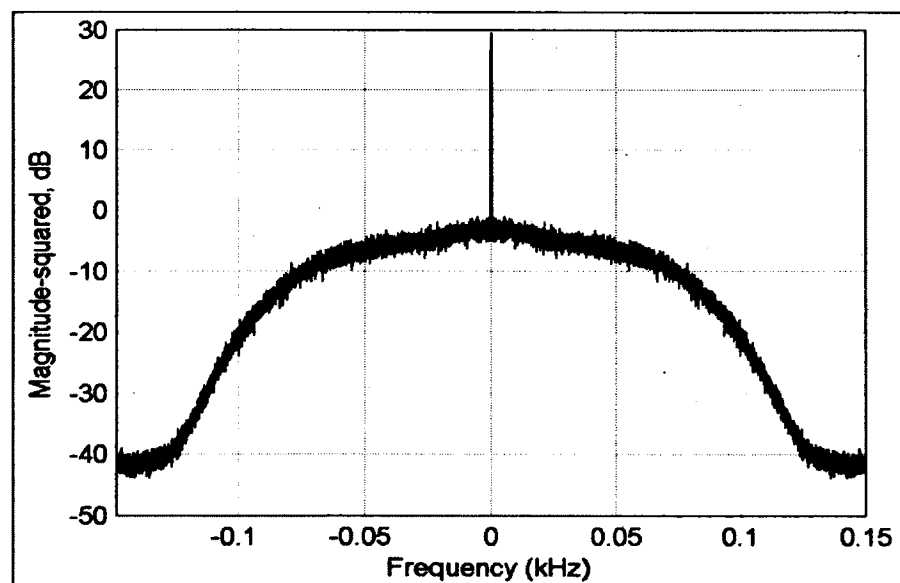


Figure 3.22 PSD de la transformation au carré du signal de transmission avec polarité et phase aléatoire, $M=3$.

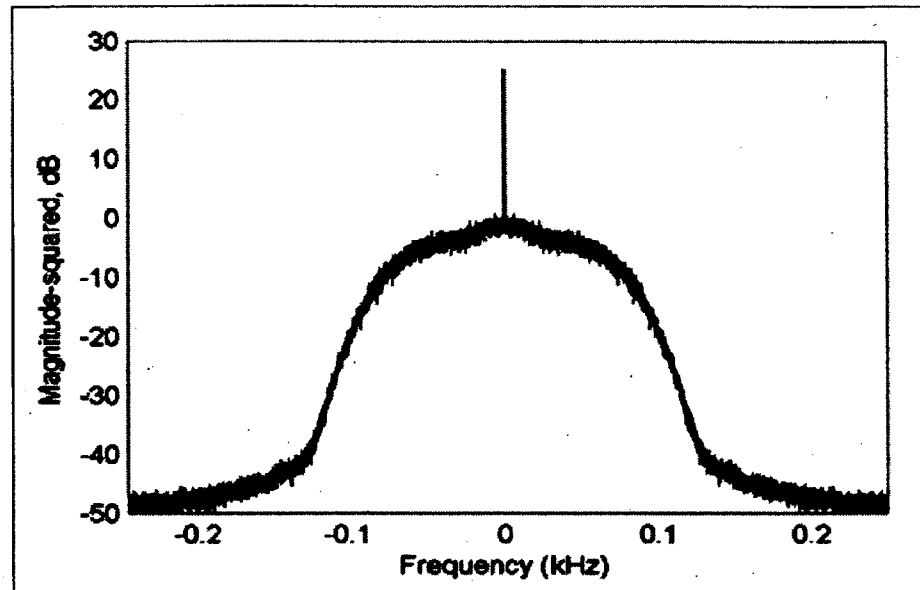


Figure 3.23 PSD de la transformation au carré du signal de transmission avec polarité et phase aléatoire, $M=5$.

3.2.2 Résultats et analyse de la SCD du signal de transmission

Les résultats de la SCD du signal de transmission, pour un facteur de suréchantillonnage de cinq avec et sans polarité aléatoire et phase aléatoire, sont présentés aux Figure 3.24, Figure 3.25, 3.26 et Figure 3.26. Il est possible d'observer que les composantes spectrales restent faiblement corrélées à la fréquence de symbole après application d'une polarité et phase aléatoire. Étant donné que la phase aléatoire introduite au signal de transmission est limitée à l'intérieur d'une période T_0 pour chaque symbole, il est possible de penser qu'une moyenne de la SCD sur plusieurs échantillons permet de corréler certaines composantes spectrales à la fréquence de symbole.

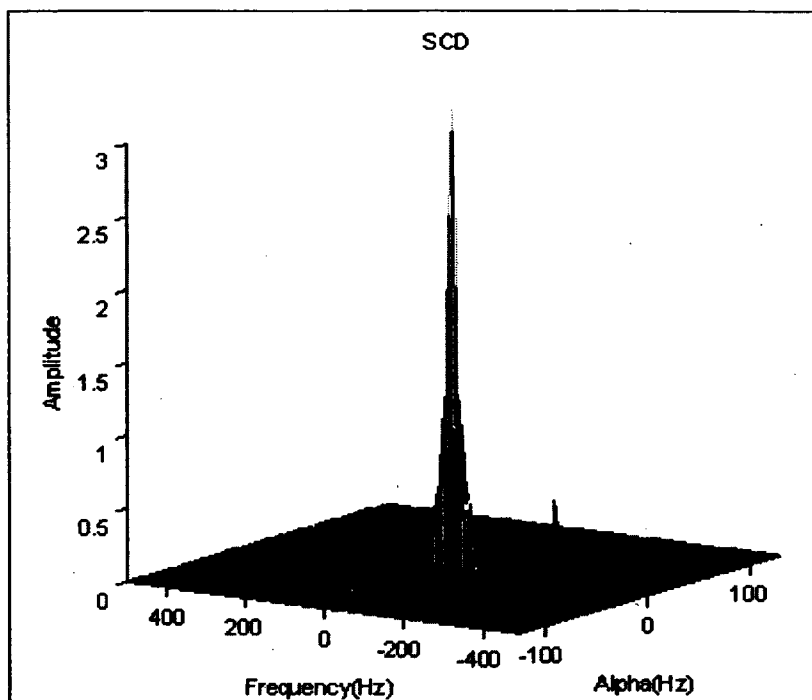


Figure 3.24 SCD sans polarité et phase aléatoire, $M=5$.

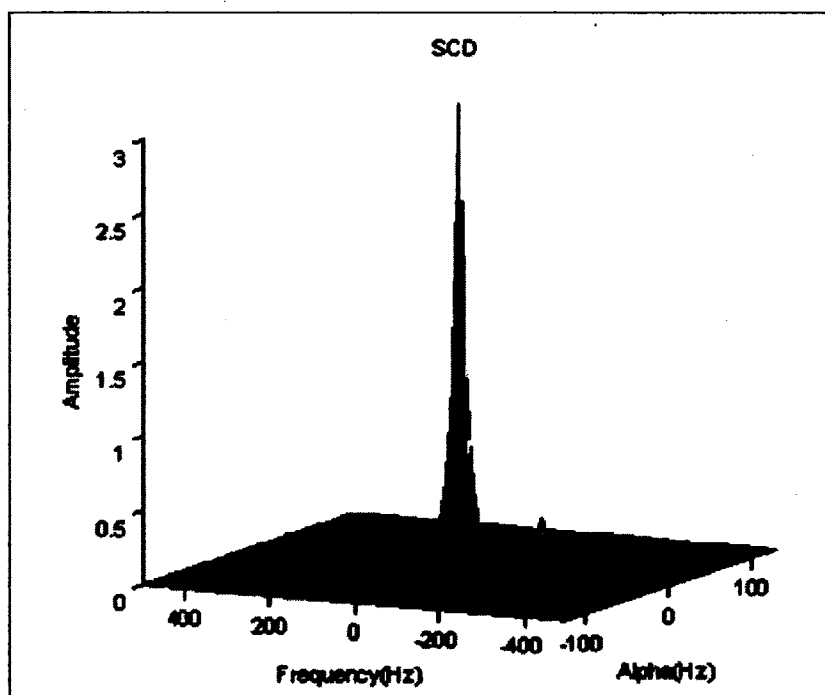


Figure 3.25 SCD avec polarité et phase aléatoire, $M=5$.

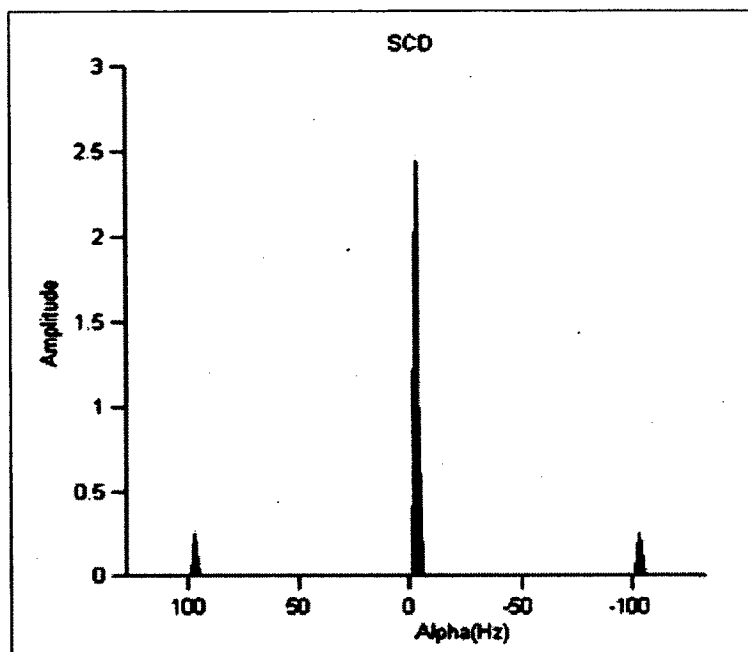


Figure 3.26 SCD vue alpha sans polarité phase aléatoire, M= 5.

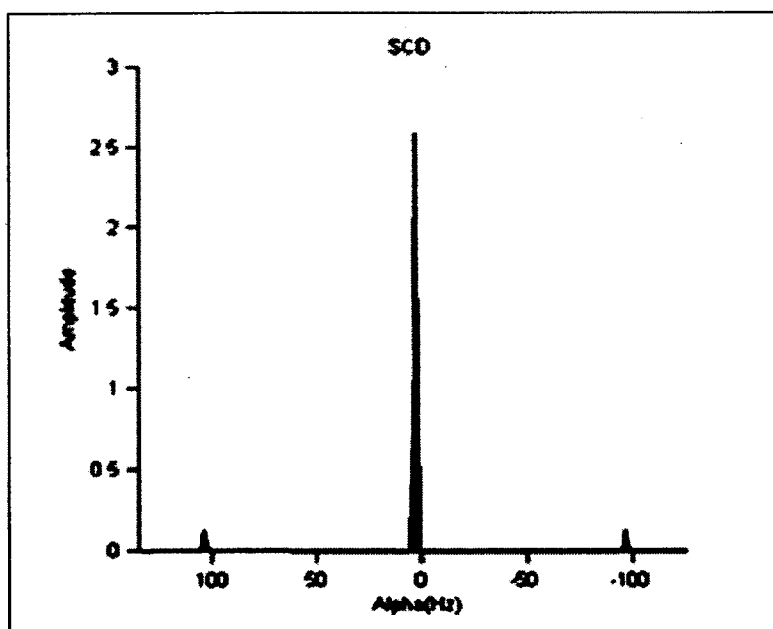


Figure 3.27 SCD vue alpha avec polarité et phase aléatoire, M= 5.

3.3 Évaluation des performances du système de communication chaotique

L'évaluation des performances du système de communication chaotique est faite en termes de taux d'erreur binaire ("Bit Error Rate", BER) fonction du rapport signal à bruit. L'évaluation est réalisée dans l'environnement *Simulink* de *Matlab* (Voir Figure 3.28). La Figure 3.29 résume les paramètres du montage. Seul le module d'introduction de phase aléatoire est implémenté.

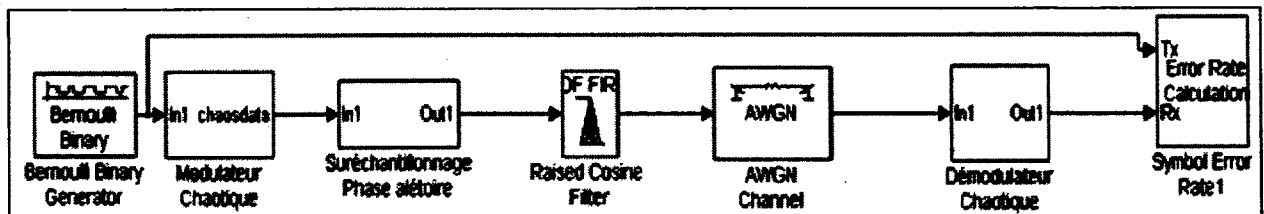


Figure 3.28 Environnement de test du système de communication chaotique.

Function Block Parameters: Paramètres Système

Subsystem (mask)

Parameters

Fréquence Tx (Hz)
100

Modulateur chaotique (Paramètres p)
0.1, 0.3 et 0.5

Facteur de suréchantillonnage (M)
2

Facteur D'arondissement (B)
0.5

Durée de l'impulsion (D symboles)
4

Longueur de trame (N symboles)
500

OK Cancel Help Apply

Figure 3.29 Paramètres du système.

3.3.1 Résultats et analyse des performances

Les performances du système sont comparées avec celle d'un système de modulation BPSK (Voir Figure 3.30). Dans les deux cas, un bit par symbole est transmis. Il est possible d'observer que plus le paramètre p du modulateur chaotique s'approche de un, plus les performances du système s'approchent de celle de la modulation BPSK. Ceci s'explique par le fait que plus le paramètre p augmente, plus la distance euclidienne entre les symboles chaotiques générés augmente, ce qui entraîne de meilleures performances du système.

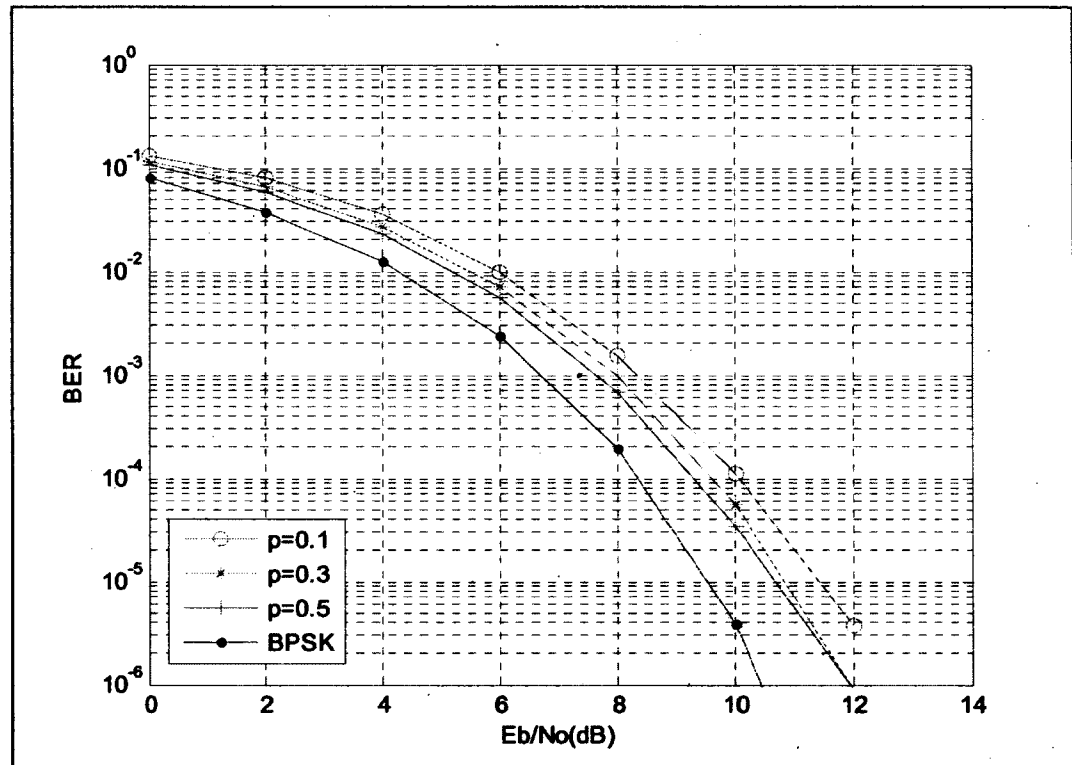


Figure 3.30 E_b/N_0 (dB).

3.4 Conclusion

Dans ce chapitre, il est montré qu'en plus du taux de symbole constant, la pente linéaire de la fonction chaotique utilisée par le modulateur induit une caractéristique cyclostationnaire qui est fonction du taux de symbole. Ceci a été un élément inattendu. Afin d'éliminer

complètement les raies spectrales à la fréquence de symbole suite à une transformation au carré du signal de transmission, il est nécessaire en plus d'une phase aléatoire de varier aléatoirement la polarité du signal de transmission. Ainsi, le module de phase aléatoire permet d'éliminer la caractéristique cyclostationnaire induite par un taux de symbole constant et le module de polarité aléatoire permet d'éliminer la caractéristique cyclostationnaire induite par une variation constante en amplitude. Pour la SCD du signal de transmission, les composants spectrales à la fréquence de symbole restent faiblement corrélés suite à l'introduction d'une polarité et phase aléatoire. Il est possible de penser qu'une moyenne de la SCD sur plusieurs échantillons permet de corrélérer certaines composantes spectrales à la fréquence de symbole lorsque la phase aléatoire est limitée à l'intérieur d'une période de symbole T_o .

La démodulation du signal chaotique avec phase aléatoire est rendue possible à l'aide de la mémoire induite par la phase aléatoire en combinaison avec la réponse à l'impulsion du filtre de mise en forme. L'utilisation d'un treillis au récepteur permet de représenter n'importe quelle séquence transmise par le transmetteur. L'algorithme de Viterbi est utilisé afin de sélectionner la séquence transmise la plus vraisemblable du treillis. Les performances du système dans un canal à bruit blanc gaussien sont comparées avec celles d'une modulation BPSK. Les résultats montrent que plus les paramètres p de la fonction chaotique utilisée augmentent, plus les performances du système s'approche des performances de la modulation BPSK. Pour terminer, le système proposé permet d'atteindre nos objectifs. Dans le prochain chapitre, nous aborderons l'implémentation matérielle de la partie la plus complexe du système soit le démodulateur chaotique. Il est à noter que l'implémentation matérielle du démodulateur chaotique sera faite pour un système utilisant seulement un module d'introduction de phase aléatoire afin de comparer les performances avec celle du modèle réalisé avec *Matlab/Simulink*.

CHAPITRE 4

ARCHITECTURE MATÉRIELLE DU DÉMODULATEUR CHAOTIQUE

Dans ce chapitre, nous présentons l'architecture du démodulateur chaotique qui sera implémenté dans un circuit intégré programmable par l'utilisateur ("Field Programmable Gate Array", FPGA). Le schéma bloc du démodulateur est illustré à la Figure 4.1. Il est formé d'un module série-parallèle et d'un décodeur de Viterbi. Le module série-parallèle permet au décodeur de Viterbi de traiter en parallèle les M échantillons formant un symbole. Ceci allège le contrôle du décodeur en plus d'augmenter sa vitesse au détriment d'une augmentation de la surface utilisée du FPGA. Le décodeur de Viterbi est séquencé en utilisant une structure conventionnelle pour son implémentation [34]. Cette structure comprend: un module de calcul de métrique de branche ("Branch Metric Unit", BMU), un module d'addition comparaison et sélection ("Add-Compare-Select", ACS), un module de gestion de l'historique, et un module dernier arrivé premier sorti ("Last In First Out", LIFO). Les modules de gestion de l'historique et LIFO, représentés en gris sur la Figure 4.1, sont adaptés pour la technique radix-2 à partir de l'implémentation décrite dans [34]. Le module BMU traite en parallèle les échantillons reçus et génère la métrique de branche de chaque transition. Cette métrique de branche représente la distance euclidienne entre le symbole reçu et le symbole qui correspond à la branche donnée du treillis. Le module ACS sélectionne la plus petite métrique d'état après avoir comparé le résultat de l'addition des métriques de branche aux métriques d'état correspondant. Cette sélection se réduit à un bit appelé bit de décision. Ce bit de décision est utilisé par le module BMU afin de réinitialiser chaque état du treillis avec l'échantillon chaotique le plus vraisemblable de l'une des deux branches de l'itération précédente correspondant à cet état. Le module de gestion de l'historique permet de décoder la séquence transmise. Cette opération consiste à retracer les bits de la séquence transmise à l'aide des bits de décision enregistrés en mémoire. La séquence décodée se trouve dans l'ordre inverse, le module LIFO permet de la réorganiser.

La séquence décodée représente donc la séquence avec la métrique cumulée la plus faible aussi appelée la plus vraisemblable.

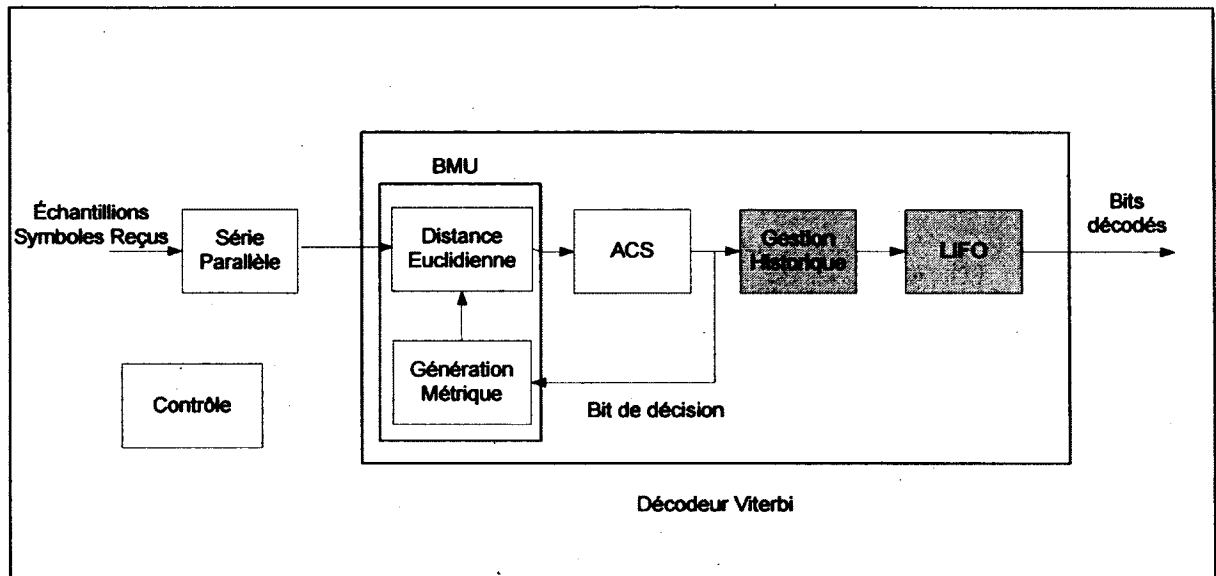


Figure 4.1 Schéma bloc du démodulateur.

4.1 Module série-parallèle

Chaque série de M échantillons formant un symbole est mise en parallèle à l'aide de simples registres à décalage. La Figure 4.2 représente le schéma bloc au niveau ("Register Transfer Level", RTL) modélisé avec l'outil de conception System Generator. Cet outil sera abordé plus en détail au prochain chapitre.

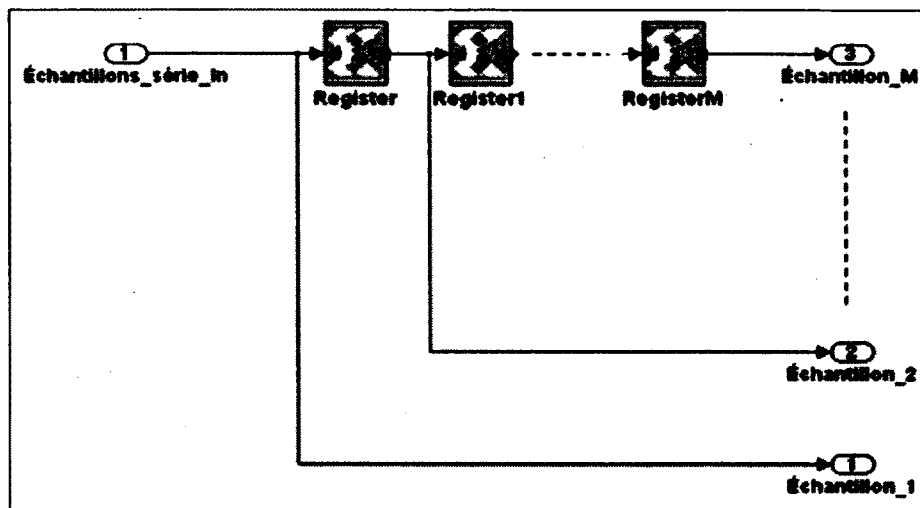


Figure 4.2 Schéma bloc(RTL) du module série-parallèle.

4.2 Module de calcul des métriques de branche BMU

Pour chaque itération du treillis, le module BMU reçoit un symbole de M échantillons et produit 2^D métriques de branche correspondant à la distance euclidienne entre le symbole reçu et les symboles de chaque branche du treillis. Comme décrit au chapitre 3, D représente le nombre de bits des états du treillis qui est fonction de la mémoire induite par le filtre de mise en forme ainsi que la séquence de gigue de phase $\theta(t)$ du transmetteur. Cette mémoire induit de l'interférence sur chaque symbole transmis étant donné qu'elle s'étend sur $D+1$ symboles. Afin d'estimer le symbole reçu, chaque branche du treillis doit générer un symbole chaotique ayant subi l'interférence de $D/2$ symboles passés et futurs. Pour ce faire, chaque branche j quittant chaque état i du treillis constitue une séquence symbolique $s[m]$ composée de $D+1$ bits. Rappelons qu'une séquence symbolique représente un train d'information binaire encodée sur la fonction chaotique du transmetteur selon $x[n] = f_{s[m]}^{-1}(x[n-1])$. Conséquemment, chaque branche du treillis permet, à l'aide d'un générateur chaotique, de générer $D+1$ symboles chaotiques. La Figure 4.3 représente un exemple d'interrelation entre trois états du treillis pour $s[m]$ et $x[n]$ de chaque branche.

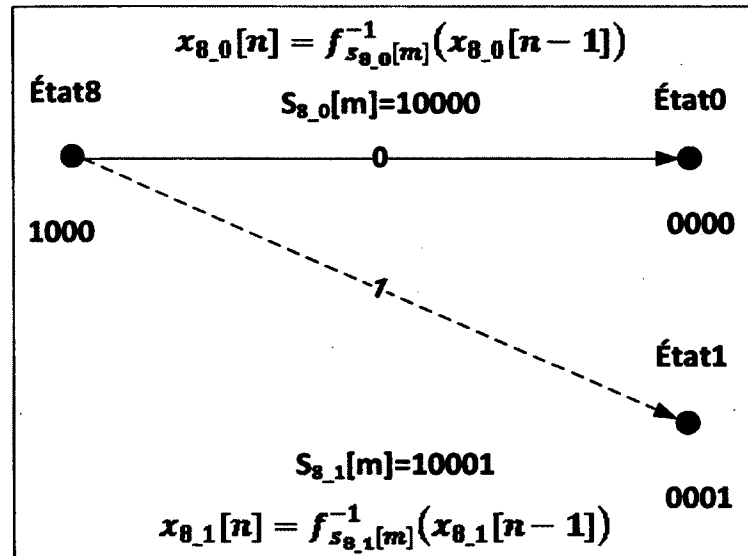


Figure 4.3 Exemple d'interrelation treillis (D=4).

La séquence de symboles chaotiques $x[n]$ générée par chaque branche du treillis représente donc les $D/2$ symboles passés et futurs en plus du symbole présent. En toute logique, le symbole présent est le $x[D/2+1]$ symbole de la séquence. Connaissant la réponse à l'impulsion du filtre de mise en forme $h(t)$ ainsi que la séquence de gigue de phase, il est maintenant possible d'estimer le symbole reçu en sélectionnant le symbole de M échantillons correspondant à la position $D/2+1$ de la séquence de symbole chaotique après mise en forme et insertion de la gigue de phase. Ainsi, n'importe quelle séquence de symbole chaotique transmise peut être représentée par un chemin unique constitué de branches successives dans le diagramme de treillis.

Pour chaque nouvelle itération du treillis, les générateurs chaotiques de chaque branche doivent être réinitialisés avec une condition initiale $x[0]$. Le passage d'un état à l'autre se traduit par un décalage d'un symbole chaotique vers la gauche. La condition initiale correspond donc au deuxième symbole de la séquence de $D+1$ symboles chaotiques générés à l'itération précédente par l'une des deux branches menant à un état donné du treillis. La condition initiale la plus vraisemblable entre l'une des deux branches menant à un état donné est sélectionnée par le bit de décision du module ACS. Ainsi, une continuité est préservée entre chaque séquence de $D+1$ symboles chaotiques de chaque itération du treillis. La Figure

4.4 illustre un exemple d'initialisation en utilisant le bit de décision du module ACS afin de sélectionner le symbole d'initialisation le plus vraisemblable qui initialisera les générateurs chaotiques des deux branches sortant de chaque état du treillis.

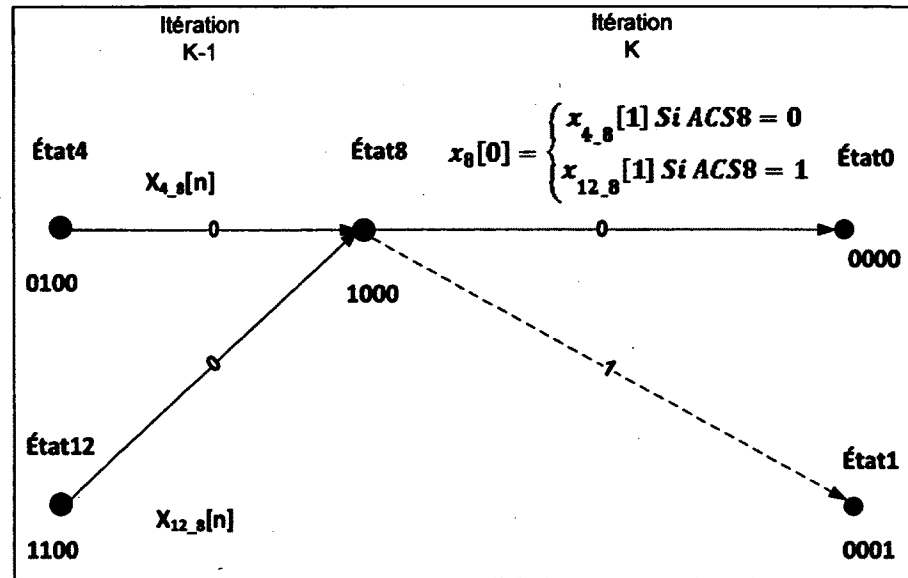


Figure 4.4 Exemple d'initialisation treillis.

La Figure 4.5 illustre le diagramme bloc du module BMU. Le module BMU est composé de 2×2^D modules de branche, d'un module de filtre de mise en forme et d'un générateur pseudo-aléatoire. Les modules de branche représentent chacune des branches du treillis et calculent la distance euclidienne entre chaque symbole qu'ils génèrent et le symbole reçu. De plus, chaque module de branche génère un symbole chaotique d'initialisation. Les bits de décision du module ACS sont utilisés afin de sélectionner à l'aide d'un multiplexeur le symbole chaotique d'initialisation le plus vraisemblable de l'un des deux modules de branches se connectant à chaque module de branche. Le générateur pseudo-aléatoire génère une séquence de gigue de phase identique à celle du transmetteur. Cette séquence permet de sélectionner la phase des coefficients du module de filtre de mise en forme. Ces coefficients sont utilisés par les modules de branche afin de mettre en forme et d'introduire la gigue de phase sur la

séquence de $D+1$ symbole chaotique que génèrent chaque module de branche afin d'estimer le symbole reçu à chaque itération du treillis.

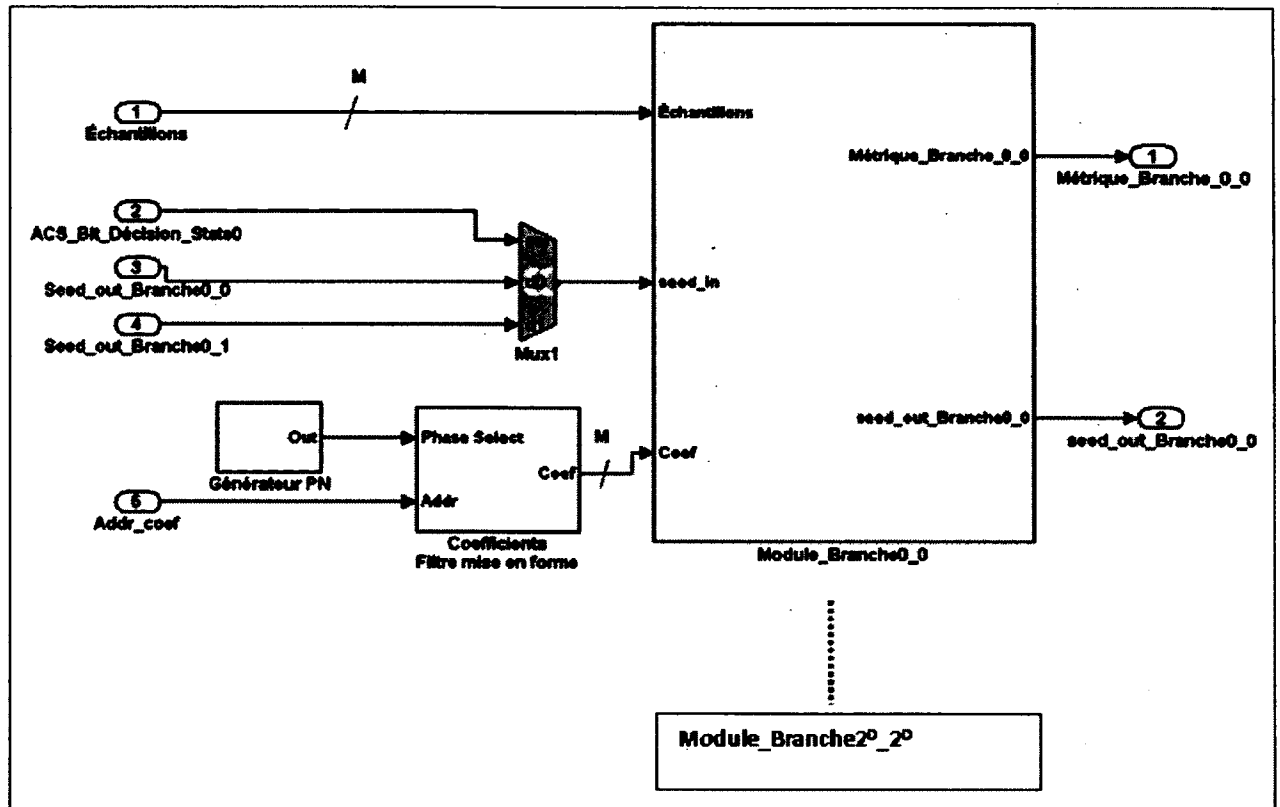


Figure 4.5 Diagramme bloc BMU.

La Figure 4.6 illustre le diagramme bloc du module de branche. Il est formé d'un module de génération des métriques, d'un module de calcul de la distance euclidienne et d'un registre à décalage qui représente les D bits de l'état de départ en plus du bit de transition vers l'état d'arrivée auquel le module branche correspond. Le module de génération des métriques est formé d'un module de générateur chaotique et d'un module de multiplication et accumulation. Le module de générateur chaotique utilise la séquence symbolique que forme le registre à décalage afin de générer une séquence de $D+1$ symboles chaotiques. De plus, ce module génère le symbole chaotique d'initialisation. Le module multiplication et accumulation accomplit, dans un premier temps, une mise en forme et insertion de la gigue

de phase en multipliant la séquence de symboles chaotique du module de générateur chaotique avec les coefficients du filtre de mise en forme. Deuxièmement, une opération addition et accumulation est accomplie sur les $D+1$ symboles chaotiques mis en forme afin d'obtenir un symbole de M échantillons ayant subi l'interférence de $D/2$ symboles passés et futurs. Finalement, la distance euclidienne est calculée entre le symbole reçu et le symbole généré par chacun des modules de branche.

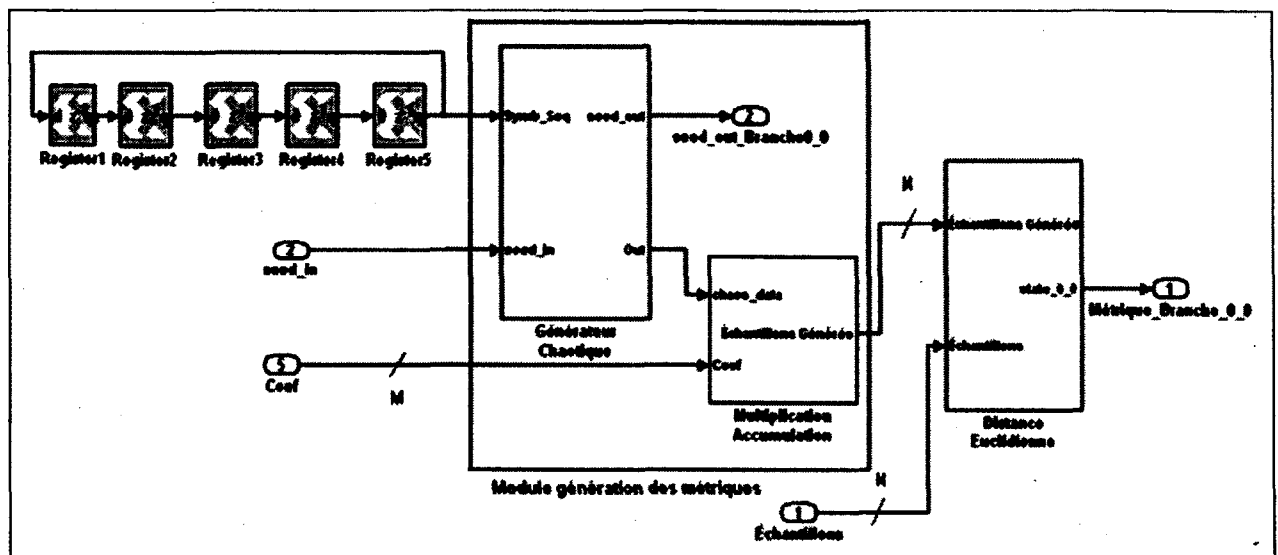


Figure 4.6 Diagramme bloc Branche0_0.

4.2.1 Générateur pseudo-aléatoire

Le module du générateur de nombre pseudo-aléatoire est implémenté à l'aide d'un registre à décalage avec rétroaction linéaire ("Linear Feedback Shift Register", LFSR) et illustré à la Figure 4.7. Le LFSR utilise le même polynôme générateur ainsi que la même semence que celui du transmetteur. Pour chaque nouveau symbole reçu, le LFSR génère une séquence de $D+1$ bits, soit un bit pour chaque échantillon chaotique généré par le module de générateur chaotique. Ainsi, deux phases sont possibles pour chaque bit soit un déphasage de zéro ou 180 degrés. Étant donné que le passage d'un état à l'autre du treillis équivaut à un décalage de un bit à gauche d'une séquence de $D+1$ bits. Le LFSR est réinitialisé à l'aide du signal

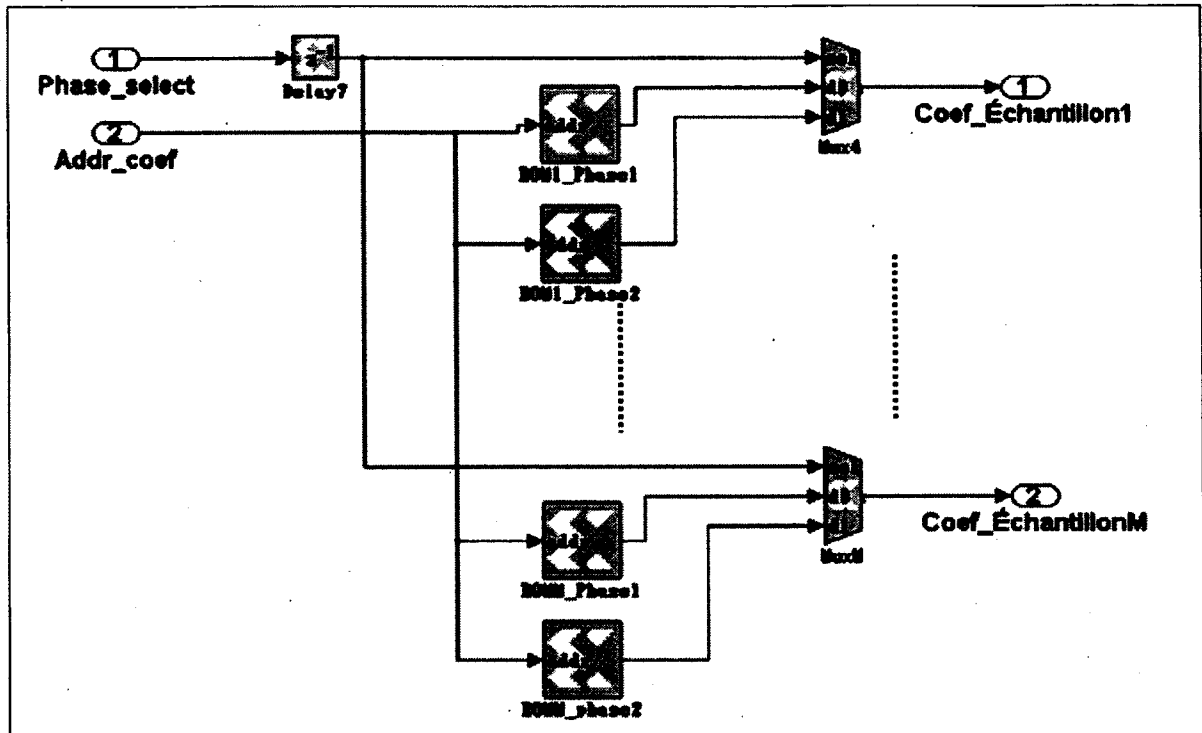


Figure 4.8 Schéma RTL du module de filtre de mise en forme.

La Figure 4.9 représente la réponse à l'impulsion $h(t)$ qui s'étale sur quatre symboles et possède deux échantillons par symbole ($M=2$ et $D=4$). Le Tableau 4.1 représente les vecteurs d'initialisations des mémoires ROM en fonction de la réponse à l'impulsion $h(t)$. Deux phases pour $h(t)$ sont possibles. Pour la première phase, les blocs ROM *ROM1_Phase1* et *ROM2_Phase1* sont initialisés avec les premiers et deuxièmes échantillons de chaque symbole respectivement. Pour la deuxième phase, les blocs ROM *ROM1_Phase2* et *ROM2_Phase2* sont initialisés de la même façon à l'exception que $h(t)$ est décalé d'un échantillon vers la droite.

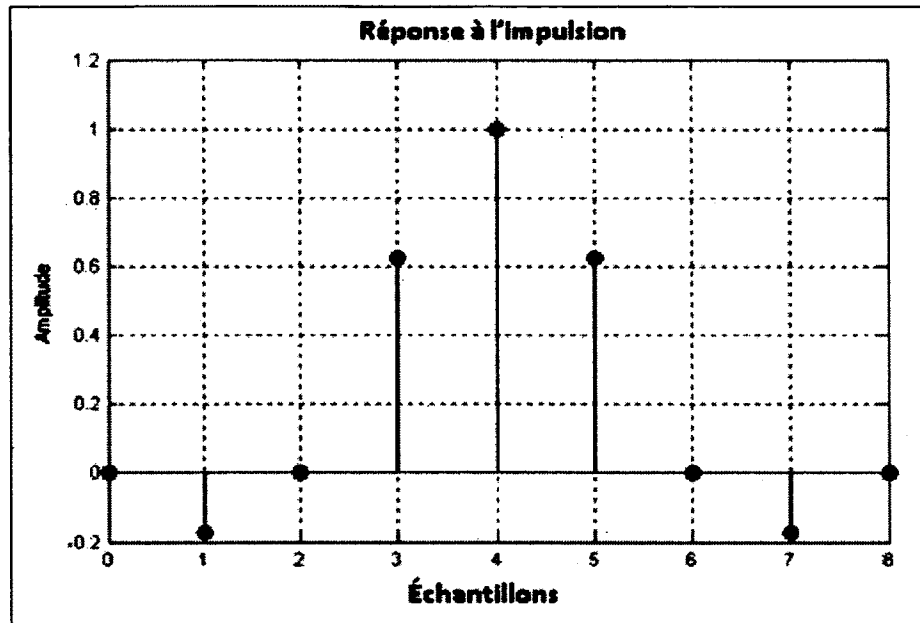


Figure 4.9 Réponse à l'impulsion $h(t)$ ($M=2$ $D=4$).

Tableau 4.1 Initialisation des mémoires ROM

ROM	Vecteur d'initialisation
ROM1_Phase1	[Échantillon0, Échantillon2, Échantillon4, Échantillon6, Échantillon8]
ROM1_Phase2	[0, Échantillon1, Échantillon3, Échantillon5, Échantillon7]
ROM2_Phase1	[Échantillon1, Échantillon3, Échantillon5, Échantillon7, 0]
ROM2_Phase2	[Échantillon0, Échantillon2, Échantillon4, Échantillon6, Échantillon8]

4.2.3 Générateur chaotique

Le module du générateur chaotique est illustré à la Figure 4.10. Les équations (4.1) et (4.2) représentent les équations de la fonction chaotique $f^{-1}(x)$. Leur implémentation est faite à

partir de multiplicateurs et d'additionneurs sous forme $ax + b$. Le signal *Symb_seq* provient du registre à décalage et sélectionne dans quelle zone la prochaine itération de la fonction chaotique aura lieu. Pour chaque nouvelle itération du treillis, le générateur chaotique est initialisé en chargeant le signal *seed_in* avec le signal *Rst_symb*. Le signal *seed_in* représente l'échantillon chaotique d'initialisation le plus vraisemblable sélectionné par le module ACS de l'itération précédente du treillis. Un registre à décalage permet de garder en mémoire la valeur de l'échantillon chaotique qui permettra d'initialiser, au début de la prochaine itération du treillis, un des états auquel le module de branche correspond. Étant donné qu'une transition entre chaque état équivaut à un décalage à gauche de un bit, le registre à décalage implémente un délai de D registres.

$$I1 = (1 - p)x/2 - (1 + p)/2 \quad (4.1)$$

$$I3 = (1 - p)x/2 + (1 + p)/2 \quad (4.2)$$

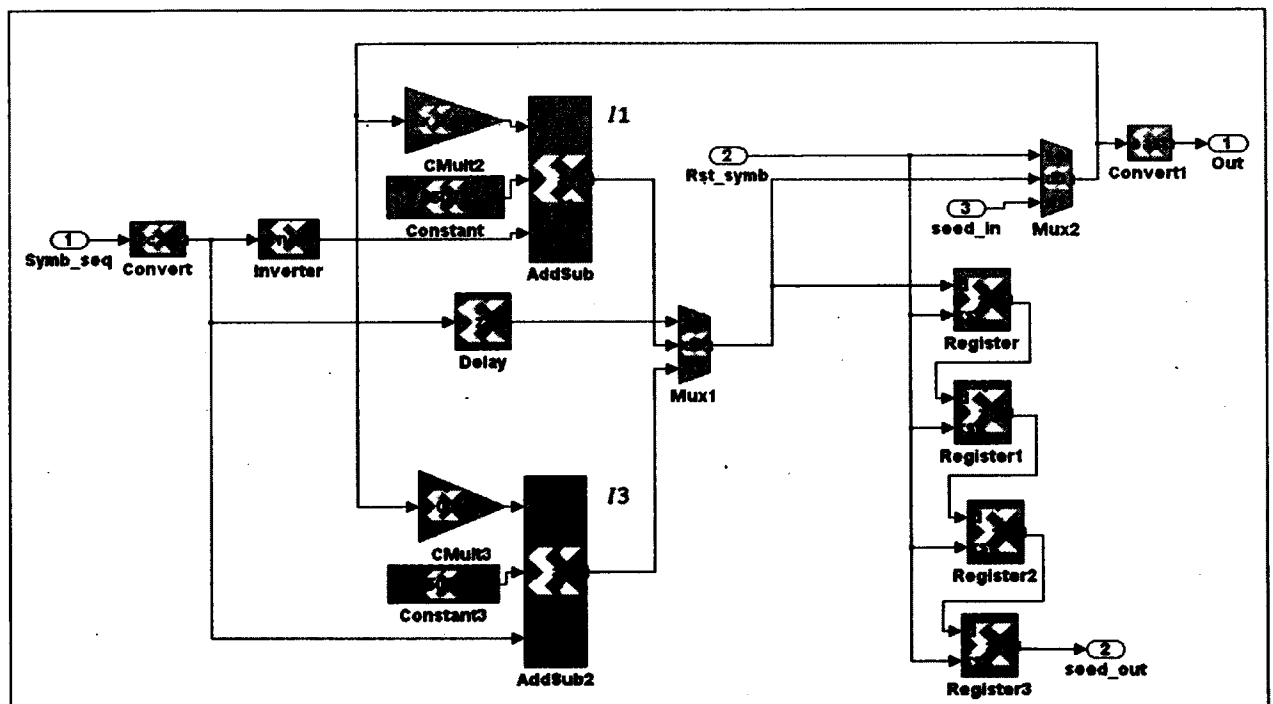


Figure 4.10 Schéma RTL du modulateur chaotique, $p=0.1$.

4.2.4 Module multiplication accumulation

Le module multiplication et accumulation génère un symbole de M échantillons. Pour chaque échantillon, un multiplicateur effectue la multiplication des $D+1$ symboles du module de générateur chaotique avec les $D+1$ coefficients du module de filtre de mise en forme. La sortie du multiplicateur est accumulée $D+1$ fois tel qu'illustré à la Figure 4.11. Ainsi, les M échantillons générés par ce module représentent le symbole estimé ayant subi l'interférence des $D/2$ symboles passés et futurs.

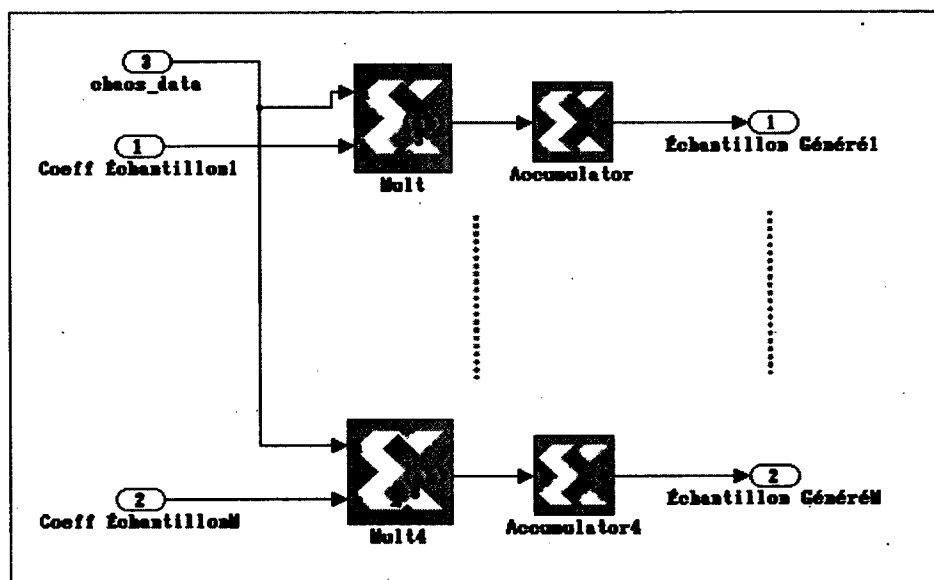


Figure 4.11 Schéma RTL du module multiplication accumulation.

4.2.5 Calcul de la métrique euclidienne

La métrique euclidienne pour chacun des modules de branche du treillis est obtenue en additionnant la valeur absolue de la soustraction en complément à deux des M échantillons du symbole reçu avec ceux générés par chaque module de branche du treillis tel qu'illustré à la Figure 4.12.

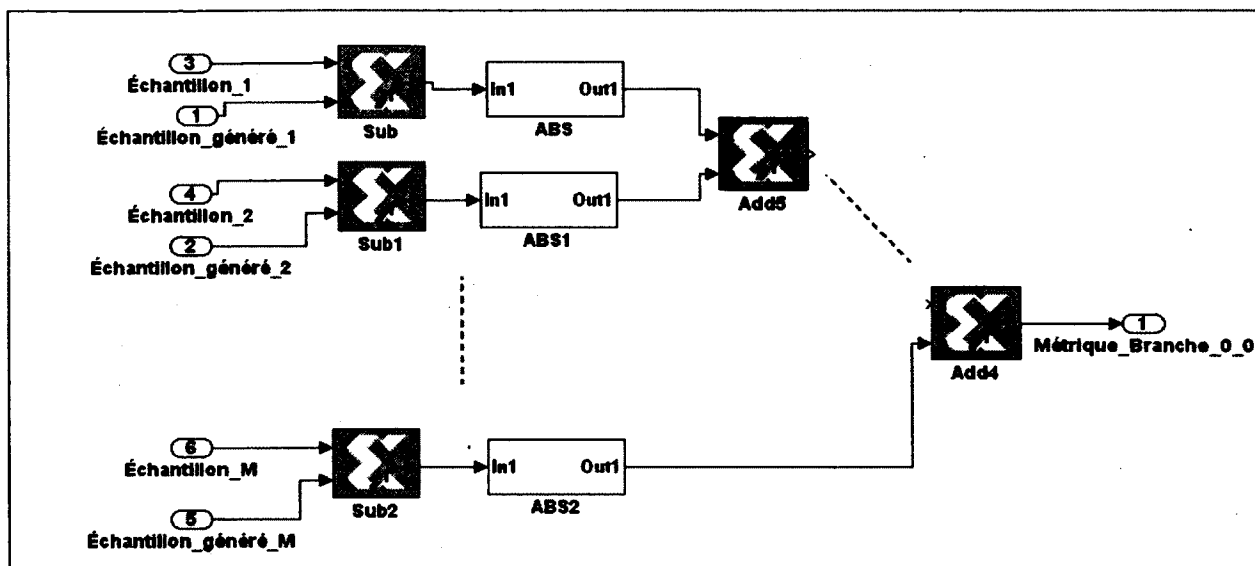


Figure 4.12 Schéma RTL du module distance euclidienne branche0_0.

4.3 Module de l'Addition-Comparaison-Sélection ACS

Le module ACS sélectionne le chemin avec la plus petite métrique cumulée des deux chemins entrants de chaque état. Il fournit aussi, aux modules de génération des métriques et de gestion de l'historique, un bit de décision indiquant quel chemin le plus vraisemblable a été choisi. Le module ACS est implémenté avec une configuration ACS à deux chemins qui représente chaque état du treillis. Deux branches entrent dans chaque état. À chaque itération du treillis, les métriques d'états associées à ces branches sont additionnées aux métriques de branche correspondantes. Ces métriques sont comparées afin de sélectionner la plus petite pour mettre à jour la métrique d'état de l'ACS. Le bit de décision est généré par le comparateur. Les modules ACS à deux chemins qui partagent les mêmes états sont regroupés ensemble et devient donc la cellule de base pour le module ACS (*Voir Figure 4.13*).

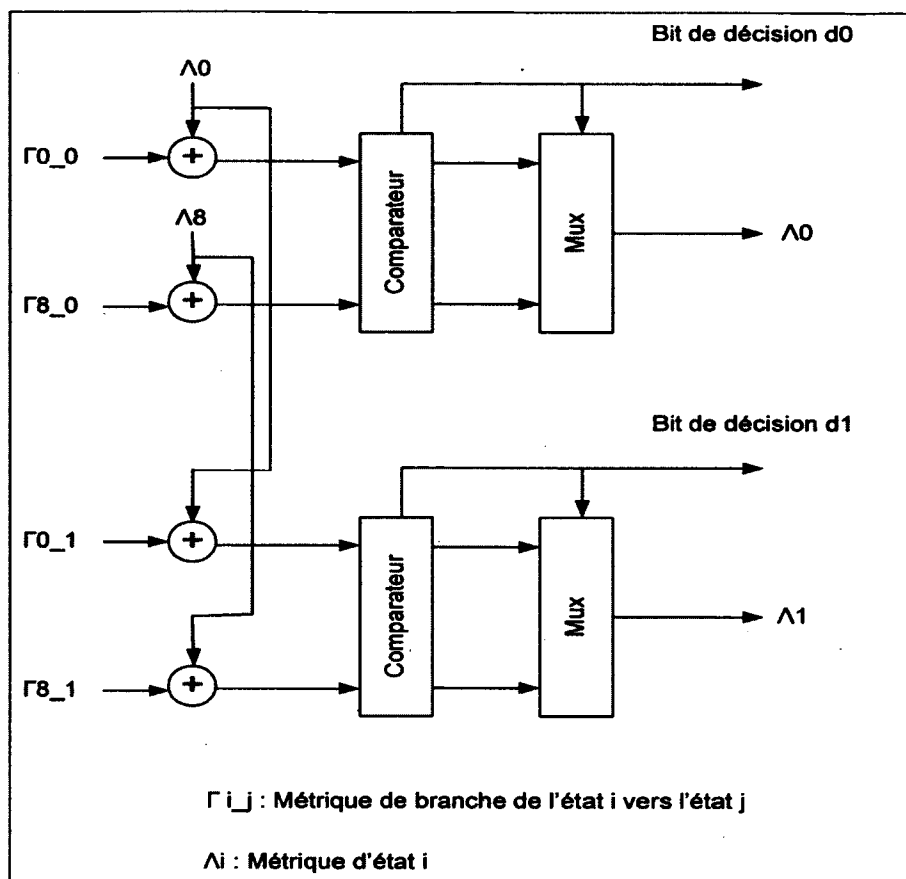


Figure 4.13 Module ACS.

La description RTL du module ACS est illustrée à la Figure 4.14. L'addition est non signée puisque les valeurs des métriques de branches et d'états correspondent à une mesure de distance soit une mesure toujours positive. La comparaison est réalisée par une soustraction complément à deux. Le bit de signe du vecteur résultant est utilisé pour sélectionner la plus petite métrique.

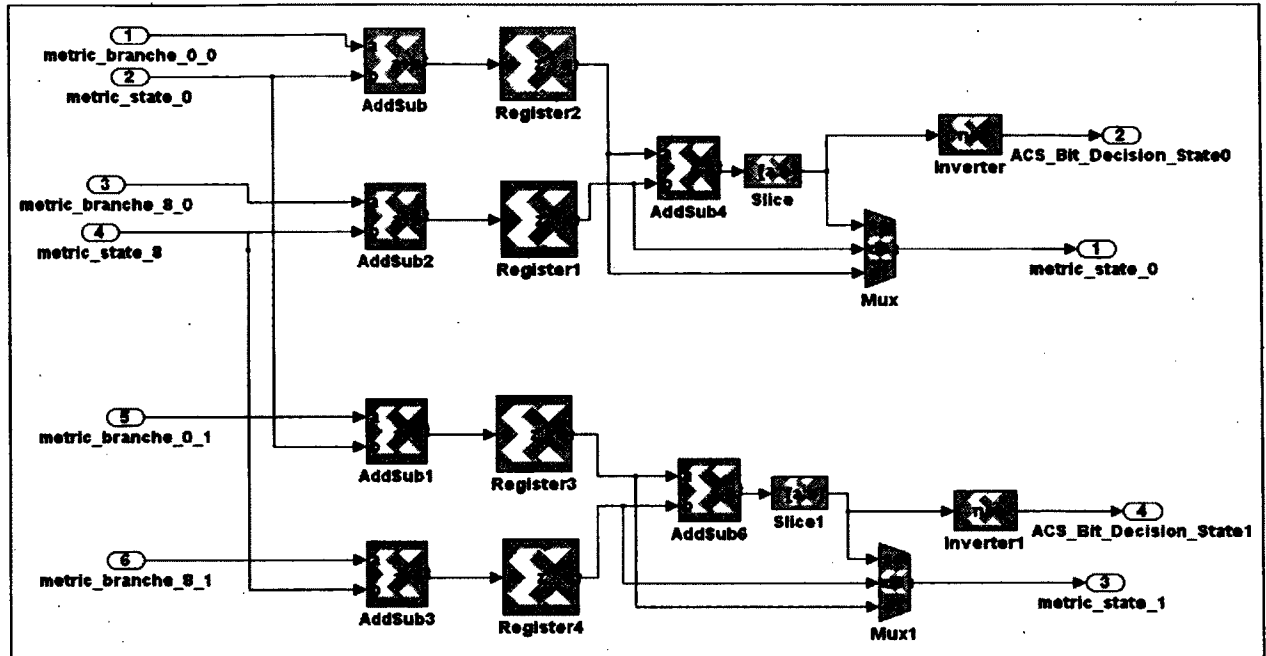


Figure 4.14 Schéma RTL du module ACS.

4.4 Unité de gestion de l'historique

Le module de gestion de l'historique utilise pour le décodage de la séquence estimée une approche par "Trace Back" (TB) [34]. L'implémentation de cette approche se base presque intégralement sur l'architecture retrouvée dans [34]. À l'exception que dans le cadre de ce projet, seulement un bit de décision est traité à la fois. Ainsi, la description de ce module est faite de manière très générale.

L'approche TB repose sur une propriété de l'algorithme de Viterbi selon laquelle tous les chemins survivants convergent après L itérations. Pour les codes convolutionnels, L doit être au moins cinq fois la mémoire du code où L est appelé la longueur de TB [7]. Nous considérons donc dans le cadre de ce projet une longueur de TB à au moins $L = 5D$ où D est la mémoire du signal. L'approche TB est divisée en deux processus, soit un processus de convergence et un processus de décodage chacun de longueur L . Le processus de convergence, qui dure L itérations, remonte à rebours à partir d'un état choisi au hasard jusqu'à l'état où tous les chemins survivants convergent. Cet état initialise le processus de

décodage qui décode le bloc de L bits de décisions suivant. La latence du module de gestion de l'historique est de L itérations.

4.4.1 Gestion de la mémoire

La mémoire du TB est partitionnée en quatre blocs RAM de longueur L pour gérer un flot continu de bits de décisions fourni par le module ACS. La séquence d'utilisation des RAM est présentée à la Figure 4.15. Un cycle de décodage se fait en quatre périodes. Le diagramme bloc du TB est illustré à Figure 4.16.

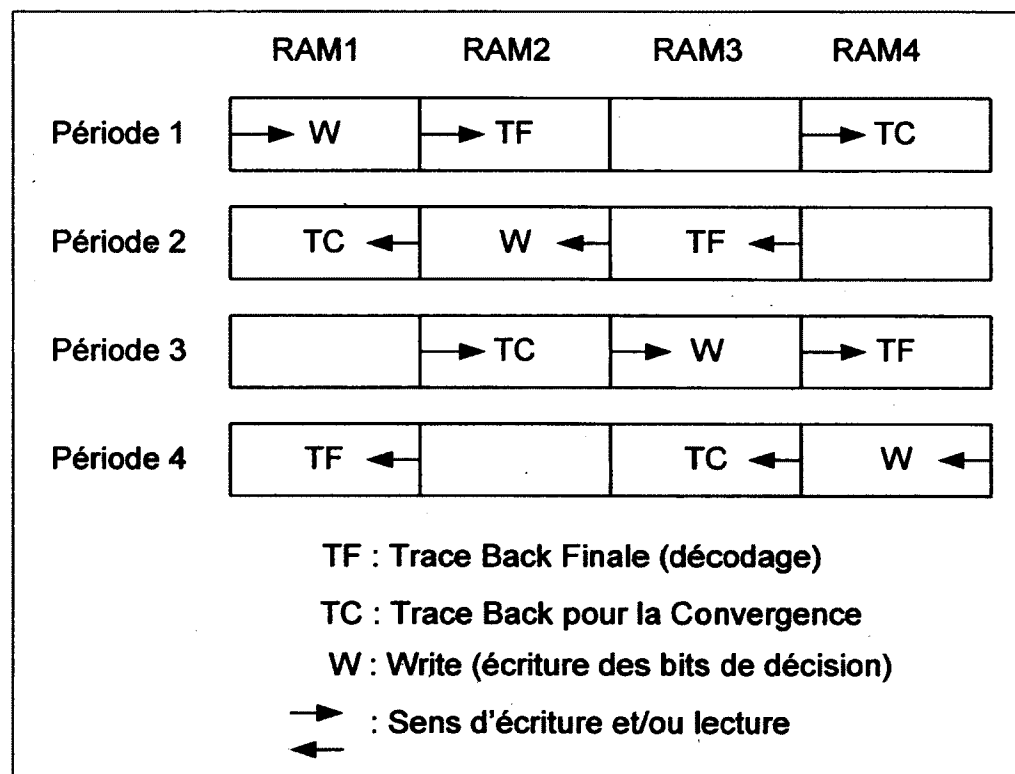


Figure 4.15 Flot de la mémoire du TB à quatre RAM.
 Tirée de Bensedek (2004, p. 40)

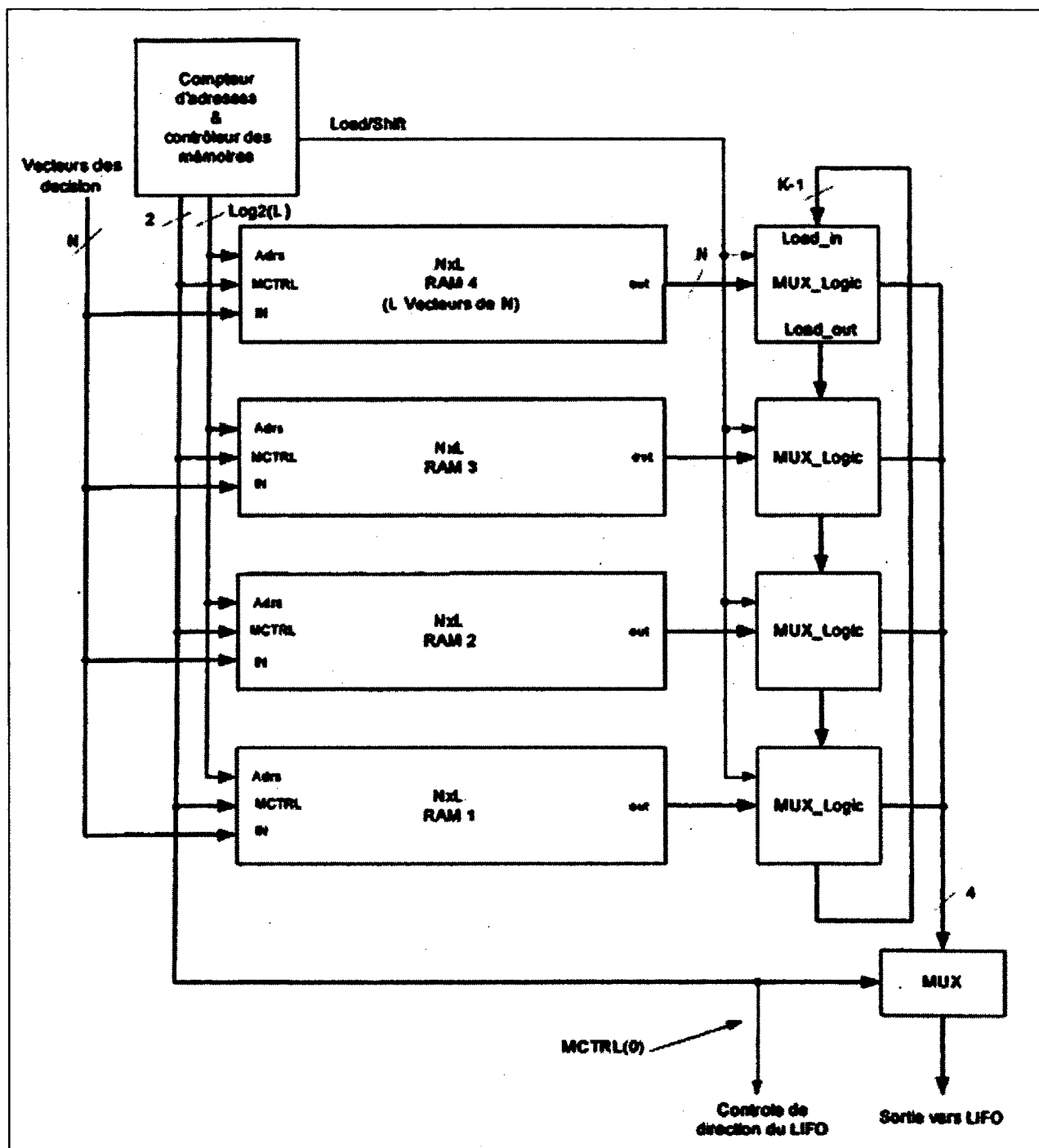


Figure 4.16 Diagramme bloc du TB.
Modifiée de Bensedek (2004, p. 42)

4.4.2 Principe de décodage

Le principe de décodage pour les processus de convergence et décodage est basé sur un registre à décalage à droite chargeable pour chacune des RAM qui fournit à ces registres un vecteur de N bits décisions (voir Figure 4.17). Pour le processus de convergence, un registre à décalage est initialisé avec un état arbitraire. L'état du registre sélectionne le bit du vecteur de décisions qui correspond à la branche qui mène à cet état. Le bit de décision sélectionné décale à droite le contenu du registre. Le nouvel état contenu dans le registre représente l'état précédent et le bit à la sortie du registre est ignoré. Ce processus est répété L fois. L'état final du processus de convergence est chargé dans le registre qui servira au processus de décodage. Le processus de décodage est identique au processus de convergence à l'exception du bit à la sortie du registre après chaque décalage représente le bit décodé.

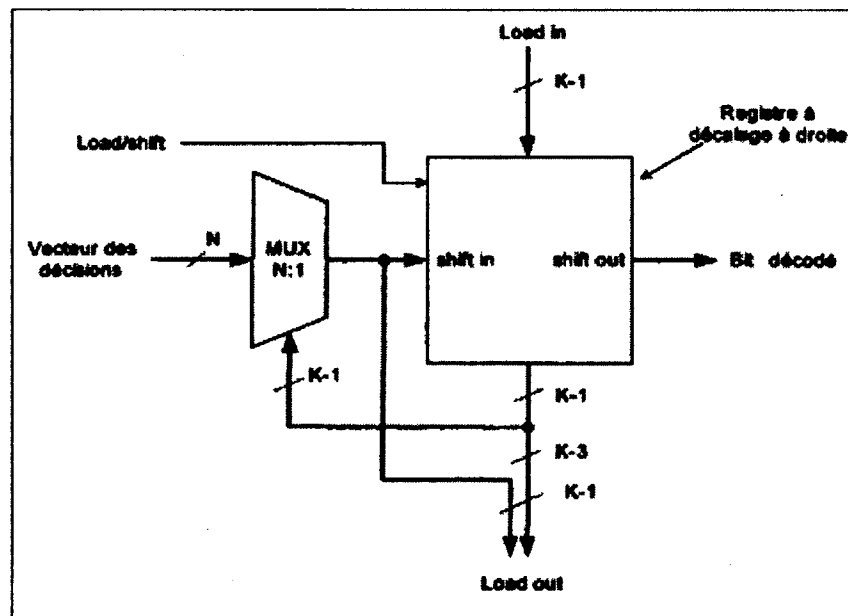


Figure 4.17 Bloc Mux_Logic.
Modifiée de Bensedek (2004, p. 43)

4.5 Unité LIFO

Les séquences de L bits fournies par le module de gestion de l'historique sont continues et délivrées dans l'ordre inverse. Le module LIFO réorganise les séquences à l'aide de deux mémoires RAM (Voir Figure 4.18). Pendant qu'une séquence de L bit est écrite dans une RAM la deuxième RAM est lue à l'inverse afin de délivrer une séquence organisée de L bits.

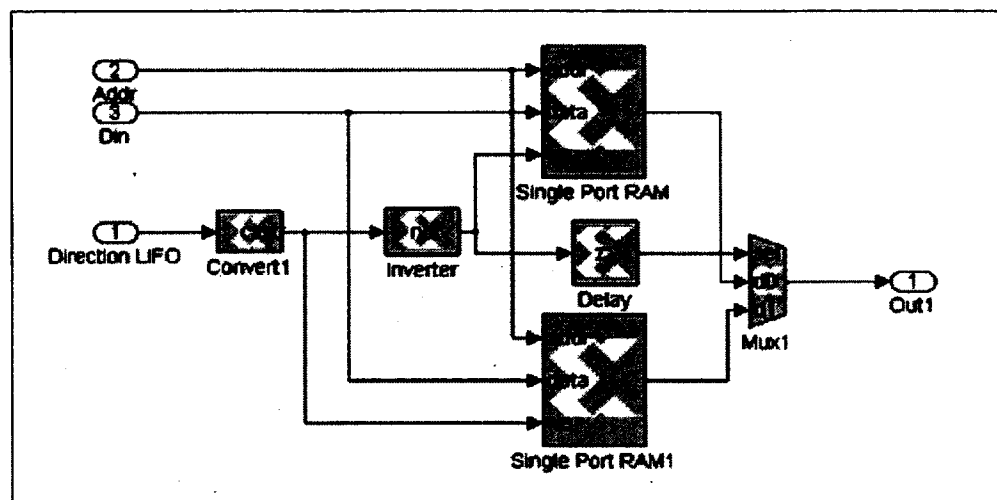


Figure 4.18 Schéma RTL du module LIFO.

4.6 Contrôle

Le schéma bloc du module de contrôle est illustré à la Figure 4.19. Il permet d'orchestrer la séquence du lancement des opérations des modules ACS (en_acs), de la génération des métriques (en_metric_gen), du calcul de la distance euclidienne (en_dist_eucl), de la gestion de l'historique (en_gen_hist) et de la réorganisation par le LIFO (en_LIFO) à l'aide d'un compteur de lancement. Ce compteur possède un nombre de comptes égaux à la latence totale des modules ACS, de la génération des métriques et de la distance euclidienne. De plus, le module de contrôle génère les $D+1$ adresses des coefficients ($addr_rom_coef$) du filtre de mise en forme du module de génération des métriques. Le compteur d'adresses des

coefficients est réinitialisé à chaque itération du treillis et mis en arrêt une fois rendu au compte final.

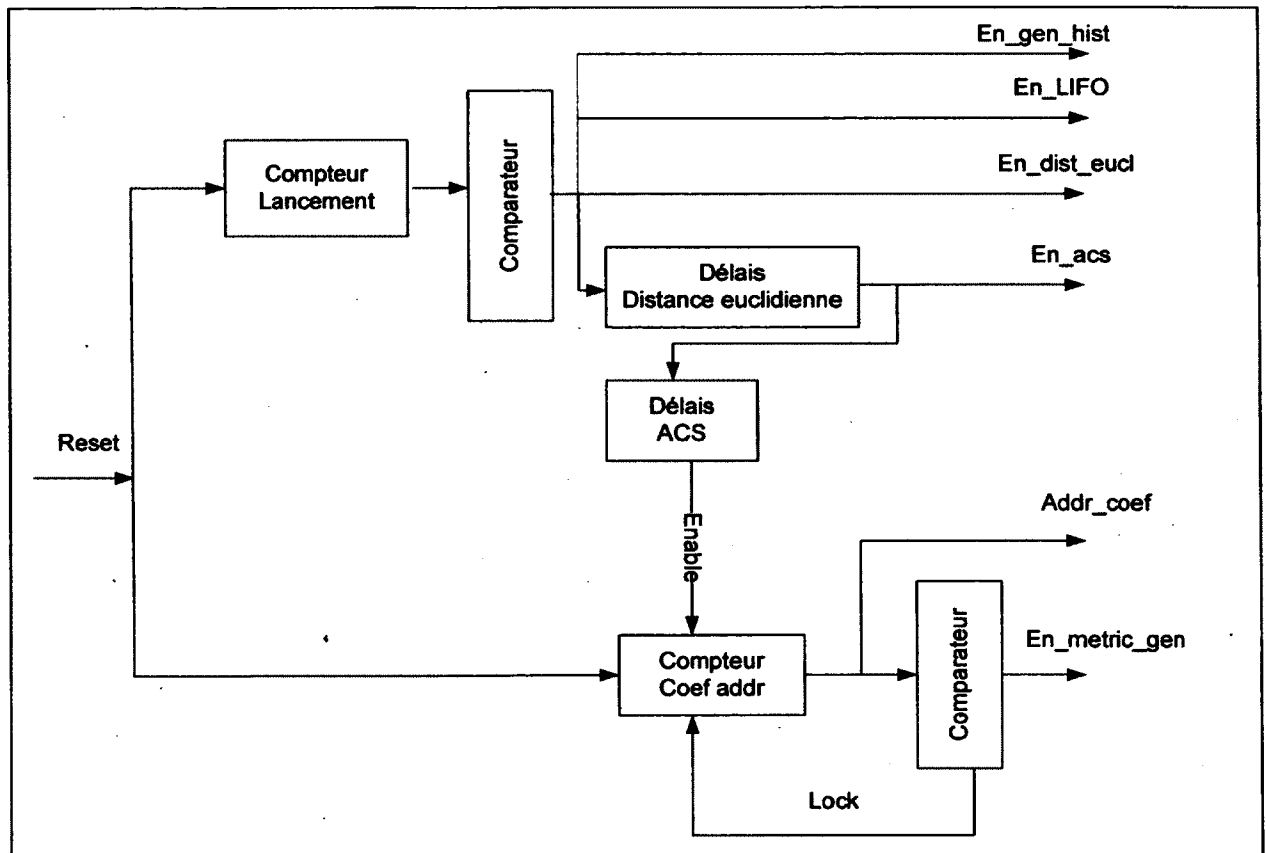


Figure 4.19 Schéma bloc du module de contrôle.

La séquence des opérations des différents modules est illustrée à la Figure 4.20. Pour chaque nouvelle itération du treillis, les modules de gestion de l'historique et LIFO décodent les bits de décisions du module ACS calculés à l'itération précédente. De plus, afin d'initialiser les générateurs chaotiques du module de génération des métriques avec le symbole le plus vraisemblable et d'estimer le nouveau symbole reçu, la distance euclidienne est calculée entre chaque symbole estimé par les modules de génération des métriques et le symbole reçu de l'itération précédente. Les bits de décisions du module ACS sont ensuite calculés. Ils serviront à l'initialisation des générateurs chaotiques du module de génération des métriques et seront décodés par les modules de gestion de l'historique et LIFO à la prochaine itération

du treillis. Finalement, après initialisation des générateurs chaotiques, le module de génération des métriques génère un symbole de M échantillons pour chaque module de branche afin d'estimer le nouveau symbole reçu.

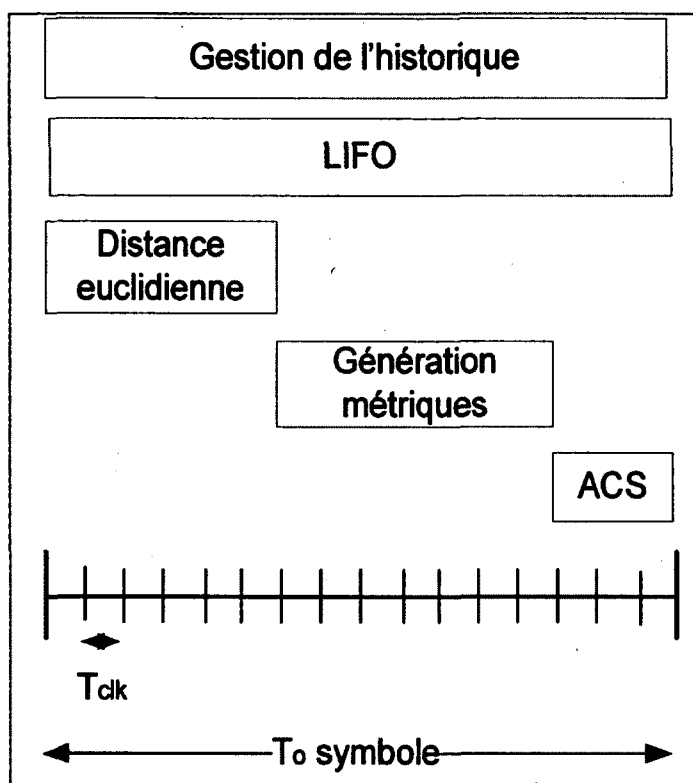


Figure 4.20 Séquence des opérations des modules du décodeur.

Chaque itération du treillis est accomplie en une période de symbole T_o . Les modules de gestion de l'historique et LIFO décodent les bits de décision du module ACS à la vitesse de symbole T_o . La vitesse de fonctionnement minimal de l'horloge T_{clk} est déterminée par la latence totale des modules ACS, génération des métriques et distance euclidienne étant donné que ces trois opérations doivent s'effectuer à l'intérieur d'une période T_o .

4.7 Conclusion

Dans ce chapitre, l'architecture d'un décodeur de Viterbi a été modifiée afin de démoduler les séquences générées par un modulateur chaotique basé sur la symbolique dynamique. Le traitement de chaque échantillon formant un symbole est fait en parallèle afin d'augmenter la vitesse de fonctionnement du décodeur et d'en simplifier le contrôle. Le module BMU est formé de modules de branche représentant chacune des branches du treillis. Chaque module de branche génère l'estimé du symbole reçu et calcule la distance euclidienne. Étant donné que chaque échantillon d'une séquence chaotique est fonction de l'échantillon au temps $n-1$, un lancement séquentiel des modules "Génération des métriques", "Distance euclidienne" et "ACS" est nécessaire. Enfin, les modules de gestion de l'historique et LIFO possèdent la même architecture qu'un décodeur de Viterbi générique.

CHAPITRE 5

IMPLÉMENTATION, TEST ET RÉSULTATS

Dans ce chapitre, nous présentons les résultats d'implémentation du démodulateur chaotique dans un FPGA Virtex-4 de Xilinx, le XC4VSX35 avec un grade de vitesse de -10. Les performances du démodulateur sont évaluées en termes de taux d'erreur binaire (Bit Error Rate, BER) dans une chaîne de communication numérique opérant en bande de base. Les performances sont ensuite comparées avec celles obtenues avec le modèle Matlab du chapitre précédent.

5.1 Stratégie d'implémentation

Ce projet étant une preuve de concept et afin d'accélérer la conception, l'implémentation du démodulateur chaotique est réalisée à l'aide de System Generator. System Generator est une librairie du logiciel Simulink de Matlab développée par Xilinx. Cette librairie est constituée de fonctions arithmétiques, logiques, mémoires et de traitement de signaux permettant de modéliser et simuler des designs basés sur les FPGA de Xilinx en utilisant un haut niveau d'abstraction similaire aux librairies de Simulink. System Generator s'interface avec les autres librairies de Simulink à l'aide de ports entrée/sortie. Ceci permet de tirer avantage du riche environnement de simulation de Simulink afin de stimuler, analyser et résoudre des problèmes de conception [35] (*Voir Figure 5.1 et Figure 5.2*). Ainsi, le démodulateur chaotique peut être facilement testé à chaque étape de son implémentation. Enfin, System Generator permet de générer notre design en langage de description matérielle tel que le VHDL ou le Verilog synthétisable afin d'être implémenté dans un FPGA. Les outils d'implémentation de Xilinx ISE, par exemple XST, NGDBuild, MAP, PAR et BitGen sont utilisés pour générer le fichier de configuration binaire.

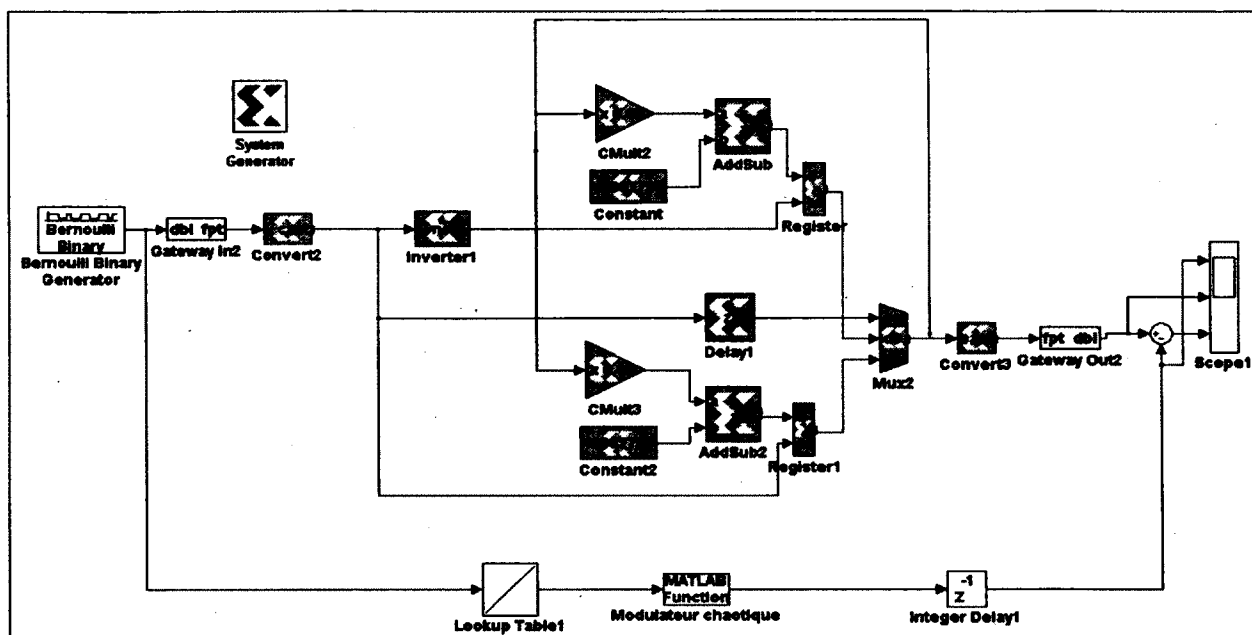


Figure 5.1 Environnement de simulation Simulink/System Generator. Test du générateur chaotique, $p=0.1$.

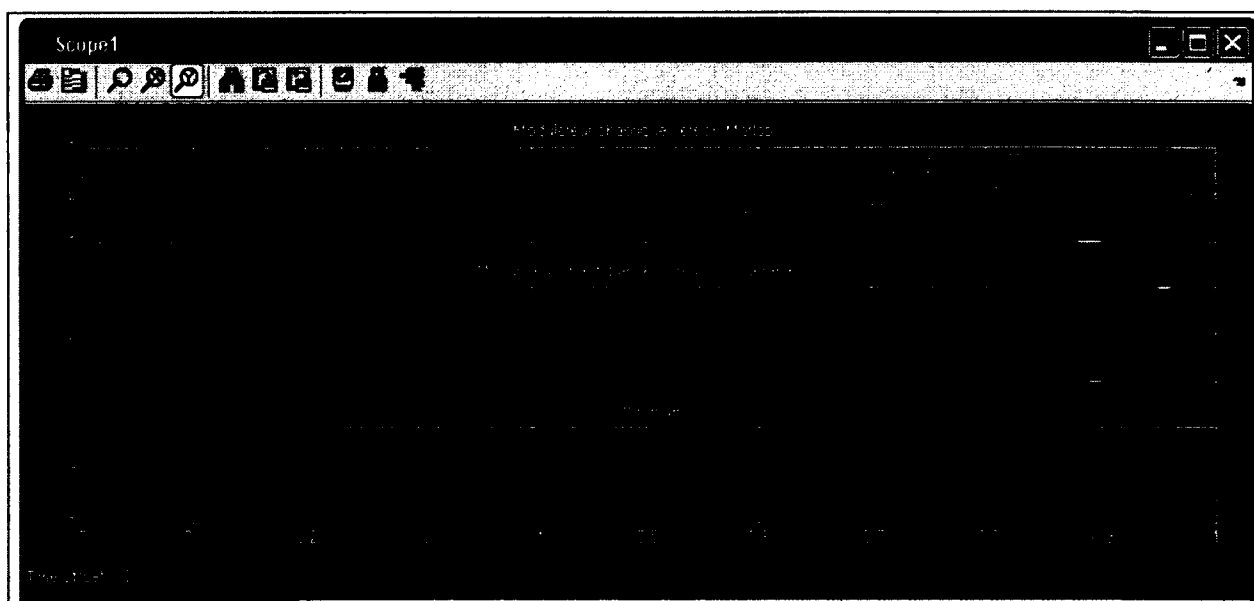


Figure 5.2 Résultat de comparaison du modulateur chaotique version Matlab (haut) et System Generator (milieu).

5.1.1 Quantification

Les signaux du démodulateur chaotique sont représentés en format point fixe. L'équation (5.1) permet de calculer le rapport signal à bruit de quantification (SQNR) pour une sinusoïdale de pleine plage quantifiée sur b bits.

$$SQNR = 1.7 + 20 \log(2^b) \approx 1.7 + 6 * b \text{ (dB)} \quad (5.1)$$

Pour chaque bit additionnel, le SQNR augmente de 6 dB. Une quantification sur plus de 8 bits n'apporte qu'une augmentation négligeable du bruit de quantification [36]. De plus, afin d'aider au choix de la quantification point fixe nécessaire pour chacun des signaux du design, la plupart des blocs de la librairie System Generator offrent la possibilité de calculer la valeur de sortie en double précision. Ainsi, lors d'une simulation et en affichant les signaux à quantifier, il est possible de déterminer quelle précision en point fixe il est nécessaire afin de minimiser l'erreur de quantification et d'éviter tout débordement. De plus, la librairie System Generator offre une fonction qui calcule l'erreur de quantification. Lorsque cette fonction possède un signal point fixe comme entrée, elle calcule l'erreur de quantification entre la valeur du signal point fixe et la valeur de ce même signal si l'opération en amont qui a servi à générer ce signal avait été faite en double précision.

5.1.2 Plateforme de développement

La plateforme de développement sur laquelle s'interface le Virtex-4 est la "XtremeDSP Development Kit-IV " (*Voir Figure 5.3*). Cette plateforme est utilisée lors de la vérification fonctionnelle du démodulateur chaotique dans son environnement de test. Cette méthode de vérification fonctionnelle se nomme ("Hardware-in-the-loop", HIL) et permet de tester le démodulateur chaotique dans le circuit Virtex-4. De plus, Cette plateforme permet une connexion USB, PCI ou JTAG entre le logiciel Matlab et le Virtex-4. Le type de connexion n'a aucun effet sur l'environnement de test. Pour ce projet, une connexion JTAG est utilisée.

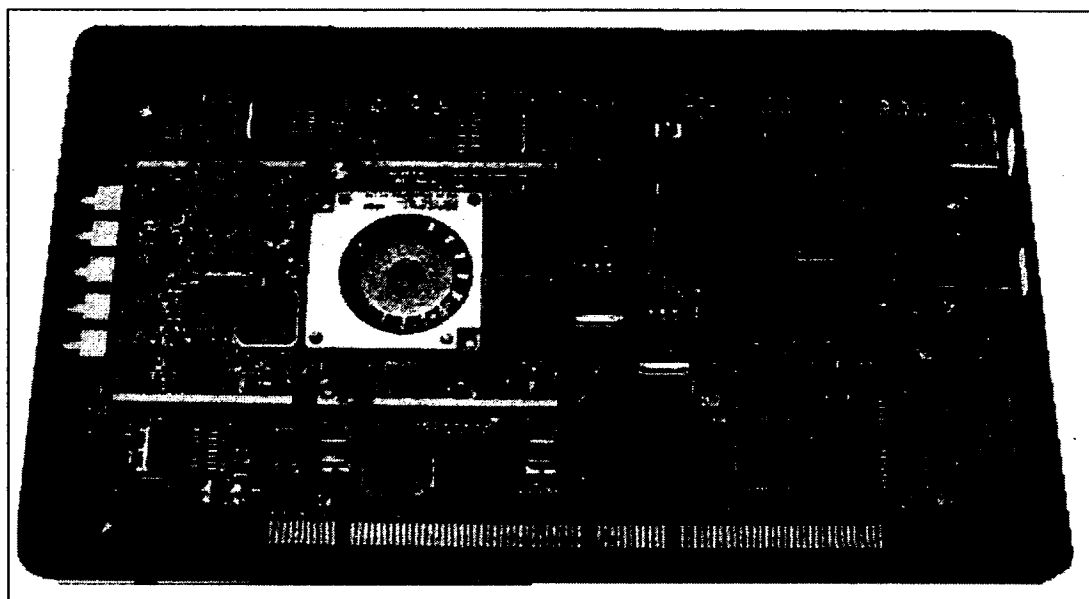


Figure 5.3 XtremeDSP Development Kit-IV.

5.2 Test et résultat

Une fois l'implémentation terminée et le démodulateur chaotique fonctionnel, un environnement de test est conçu afin de tester l'implémentation du démodulateur chaotique sur le Virtex-4. Le but de ce test est de vérifier le fonctionnement réel du démodulateur chaotique.

5.2.1 Environnement de test

La Figure 5.4 illustre l'environnement de test réalisé dans Matlab/Simulink et utilisé pour tester le démodulateur chaotique afin de tracer la courbe de BER. La chaîne de transmission est la même que celle utilisée au chapitre 3. Les paramètres sont illustrés à la Figure 5.5. Pour $D = 4$, le démodulateur chaotique est représenté par un treillis de 16 états. Ceci représente un module BMU de 32 modules de branche. Chacun de ces modules traite $M=2$ échantillons

par symbole afin de générer les métriques de branche. Le module du générateur chaotique de chaque module de branche est configuré avec le même paramètre $p=0.1$ du transmetteur.

System Generator fournit une interface de co-simulation matériel qui permet d'incorporer le FPGA directement dans l'environnement de test Matlab/Simulink [35]. Le démodulateur chaotique est représenté par un bloc de co-simulation de Xilinx. Ce dernier est généré par System Generator afin d'interfacer l'environnement de test Matlab/Simulink et le démodulateur chaotique implémenté sur le FPGA de la plateforme de développement. De plus, les tâches de configuration, transfert de données et contrôle d'horloge sont automatisées par System Generator. Ainsi, lors de la simulation de l'environnement de test, les vecteurs de test fournis au bloc de co-simulation du démodulateur chaotique sont traités par le FPGA.

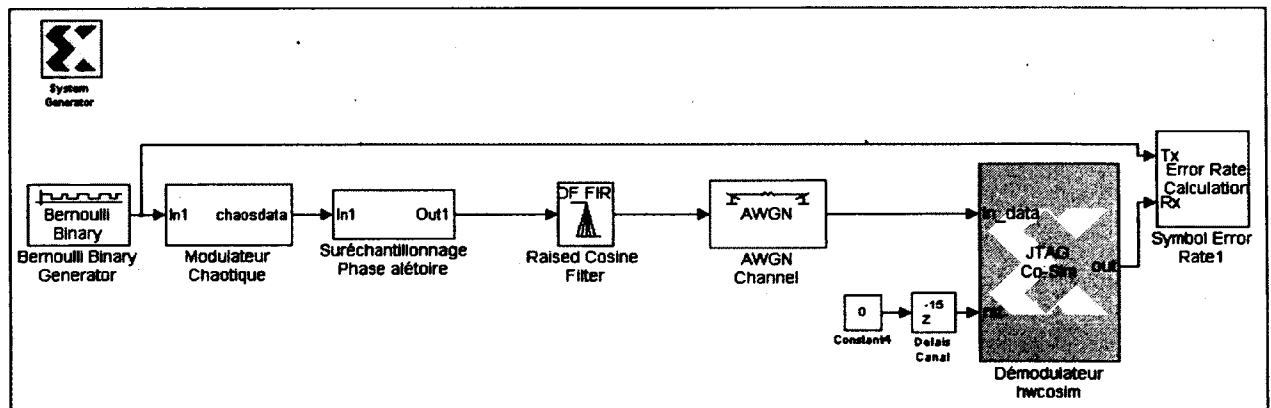


Figure 5.4 Environnement de test du démodulateur chaotique.

Function Block Parameters: Paramètres Syst...

Subsystem (mask)

Parameters

Fréquence Tx (Hz)
100

Modulateur chaotique (Paramètre p)
0.1

Facteur de suréchantillonnage (M)
2

Facteur D'arrondissement (B)
0.5

Durée de l'impulsion (D symboles)
4

Longueur de trame (# symboles)
500

OK Cancel Help Apply

Figure 5.5 Paramètres de simulation.

5.2.2 Procédure de test

La Figure 5.6 illustre la procédure de test permettant de tracer la courbe de BER en fonction du rapport signal à bruit (SNR). Le nombre d'erreurs est calculé à partir des bits du message original et du message décodé par le démodulateur chaotique. Si le nombre de bits du message original atteint 10^6 ou si le nombre d'erreurs atteint 100, la simulation est arrêtée pour un SNR donné et le BER est calculé selon l'équation (5.2). Ceci permet d'obtenir une bonne estimation de la probabilité d'erreur binaire pour un $BER=10^{-5}$. Le SNR est ensuite incrémenté d'un pas de 2 dB. Lorsque le SNR atteint 14 dB, la simulation est terminée et la courbe de BER est tracée en fonction du SNR.

$$BER = \frac{\text{nombre d'erreurs}}{\text{nombre total des bits envoyés}} \quad (5.2)$$

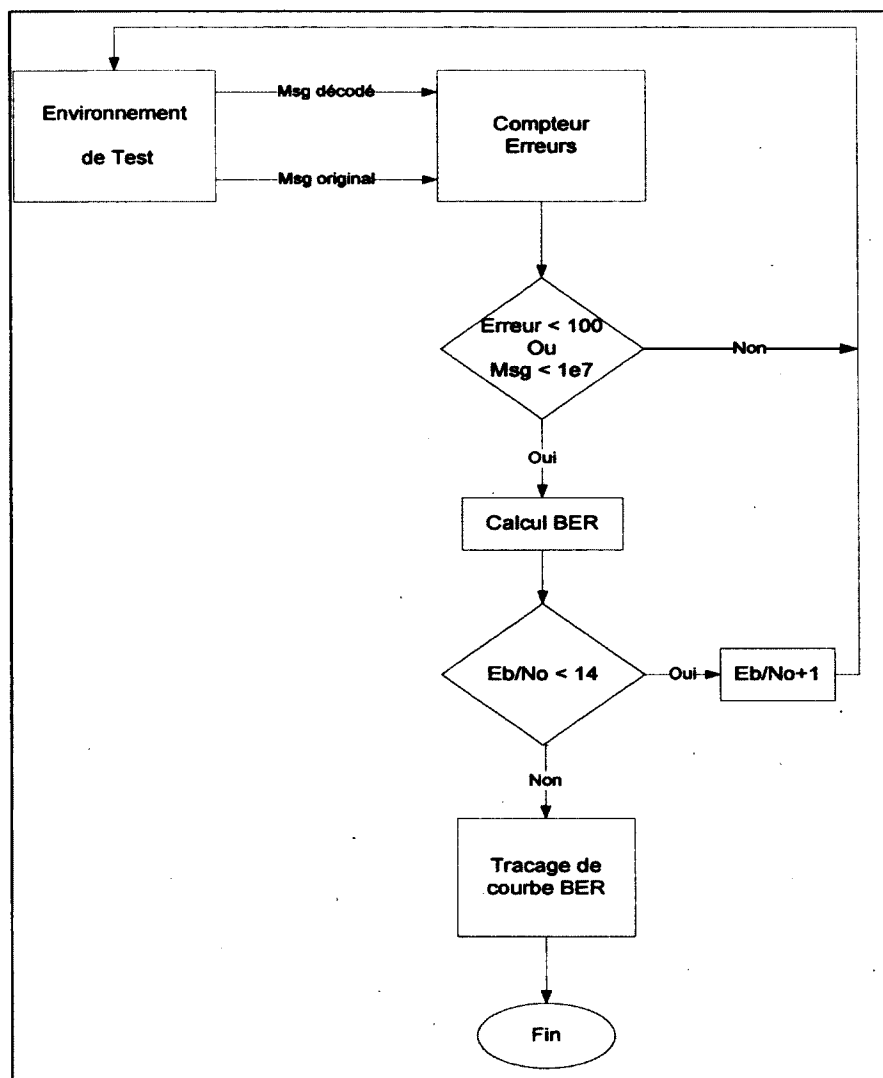


Figure 5.6 Procédure de test.

5.2.3 Résultat de co-simulation et comparaison avec modèle Matlab

La Figure 5.7 illustre les courbes de BER pour la co-simulation System Generator et le modèle Matlab du chapitre 3 pour les mêmes paramètres de simulation. Ces courbes de performance d'erreur coïncident quasi parfaitement. Ceci confirme donc le fonctionnement réel du design sur un circuit FPGA.

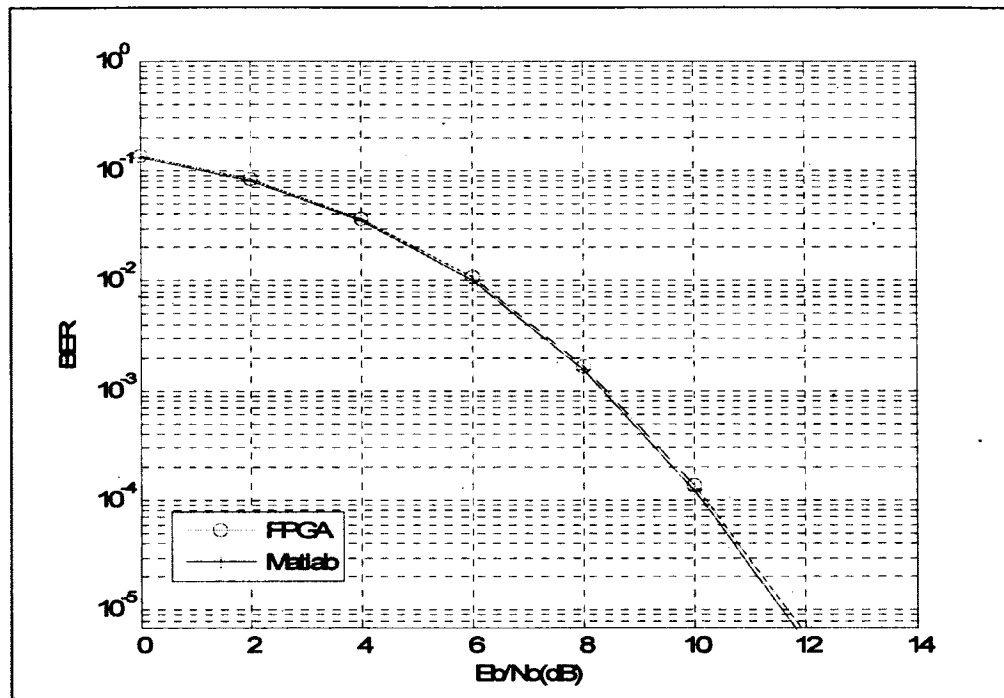


Figure 5.7 Courbes BER pour simulation Matlab et co-simulation sur circuit FPGA, ($D=4$, $M=2$, $p=0.1$).

5.2.4 Rapport d'implémentation

L'analyse de l'occupation en surface de l'implémentation du démodulateur chaotique se divise parmi les trois principaux types d'éléments logiques qui constituent l'architecture du Virtex-4. Ces éléments sont les tranches logiques, les tranches de DSP48 et les blocs RAM. Chaque tranche logique offre de la logique combinatoire et synchrone, de la mémoire distribuée en plus de registres à décalage. Chaque tranche de DSP48 contient un multiplicateur dédié de 18bits x 18bits, un additionneur intégré et un accumulateur de 48bits. Enfin, chaque bloc RAM est constitué de 18Kbits de RAM type double-port. Une description complète de chacun de ces types d'élément logique se trouve dans les références [37] et [38].

Un résumé des résultats d'occupation en surface et en vitesse du démodulateur chaotique en entier en plus de chacune de ses unités est présenté au Tableau 5.1. Le pourcentage d'occupation en surface du FPGA est présenté entre parenthèses. Ces résultats sont obtenus après le placement et routage afin d'obtenir l'occupation en surface et la fréquence maximale réelle du démodulateur chaotique et de ses unités. Les rapports correspondants sont placés à l'annexe 1. Les résultats indiquent que le module BMU est le plus critique en occupation de surface et en vitesse. Ceci devient évident lorsque l'on analyse l'architecture de chaque module de branche que forme le module BMU présenté au chapitre précédent. Ce sont ces modules qui accomplissent la majorité des calculs du démodulateur chaotique en générant les métriques de chacune des branches du treillis.

Tableau 5.1 Résumé des résultats après le placement et routage du démodulateur chaotique et de ses modules, ($D=4$, $M=2$, $p=0.1$)

Module	S/P	BMU	ACS	TBU	LIFO	Démodulateur Complet
Nombre de tranches	6 (1%)	9907 (64%)	502 (3%)	53 (1%)	9 (1%)	10503 (68%)
Nombre de blocs RAM	0	2 (1%)	0	4 (2%)	2 (1%)	8 (4%)
Fréquence Maximale (MHz)	144.78	72.33	128.27	183.01	323.62	71.65

Afin d'analyser l'impact de l'utilisation des tranches de DSP48 offertes par le Virtex-4 sur l'occupation en surface et en vitesse, les multiplicateurs du module multiplication-accumulation de chaque module de branche du BMU ont été implémentés spécifiquement en ciblant les tranches de DSP48. Pour $D = 4$, $M=2$ et selon l'architecture présentée au chapitre précédent, ceci implique $2^D \times 2 = 32$ modules de branche ayant chacun deux multiplicateurs par module multiplication-accumulation pour un total de $32 \times M = 64$ multiplicateurs. Le

Tableau 5.2 compare les résultats du module BMU avec et sans utilisation de tranches DSP48. Il est possible d'observer une diminution d'occupation en surface de 30% en plus d'une légère augmentation de la fréquence maximale du module BMU. Chaque multiplicateur occupe une tranche de DSP48. Ces résultats démontrent bien que les opérations mathématiques du module BMU occupent un grand pourcentage de la surface utilisée par ce module. De plus, aucune technique de pipeline et de balancement des registres n'a été utilisée, ce qui empêche de tirer avantage du gain en performance que peuvent offrir les multiplicateurs dédiés des tranches de DSP48. Ceci explique donc la légère augmentation en vitesse du module BMU utilisant les tranches de DSP48.

Tableau 5.2 Résumé des résultats après le placement et routage du module BMU avec et sans DSP48, (D=4, M=2, p=0.1)

BMU	Nombre de tranches	Nombre de blocs RAM	Nombre de DSP48	Fréquence Maximale (MHz)
Sans DSP48	9907 (64%)	2 (1%)	0	72.33
Avec DSP48	5265 (34%)	2 (1%)	64 (33%)	73.90

Tel qu'il le fut montrée au chapitre 4, le débit du démodulateur chaotique dépend directement de la latence totale des modules ACS, de la génération des métriques et de la distance euclidienne. Cette latence détermine à quel débit chaque bit de décision du module ACS est transmis aux modules de gestion de l'historique et LIFO afin de décoder la séquence transmise. La période du signal de contrôle des modules de gestion de l'historique et LIFO doit être un multiple entier du débit d'entrée du démodulateur chaotique qui correspond à un facteur de $M=2$. Étant donné qu'un seul bit est décodé à la fois, le débit net est $1/(14 \times T_{clk})$ selon le Tableau 5.3. En utilisant la fréquence la plus critique du démodulateur soit $1/T_{clk}=72.33\text{Mhz}$, le débit net est d'environ 5.16Mbit/s.

Tableau 5.3 Calcul du débit net du démodulateur chaotique, ($D=4$, $M=2$, $p=0.1$)

Module	Latence
Génération Métriques	10
Distance Euclidienne	2
ACS	1
Latence Totale	13
Période du signal de contrôle des modules gestion de l'historique et LIFO	$13+(13 \bmod(2))=14 \times T_{clk}$
Débit net des données	$1(14 \times T_{clk})$

5.3 Conclusion

Le test de l'implémentation du démodulateur chaotique sur le Virtex-4 de la plateforme de développement "XtremeDSP Development Kit-IV " a été décrit dans ce chapitre. Afin d'accélérer l'implémentation, la librairie System Generator de Matlab/Simulink est utilisée. De plus, System Generator a permis d'intégrer directement le Virtex-4 dans l'environnement de test Matlab/Simulink. Ainsi, lors d'une co-simulation Simulink/System Generator, les vecteurs d'informations reçus par le démodulateur chaotique sont traités directement par le FPGA. Les courbes de performance de BER obtenue entre la simulation du modèle Matlab et celle de la co-simulation Simulink/System Generator confirment le fonctionnement réel de l'implémentation du démodulateur chaotique sur un circuit FPGA.

CONCLUSION

Dans ce mémoire, on y propose un système de communication LPI basé sur la modulation chaotique. Le modulateur chaotique est basé sur la symbolique dynamique et utilise une fonction chaotique inverse. La symbolique dynamique permet l'utilisation d'un algorithme à maximum de vraisemblance au récepteur et la fonction chaotique inverse évite le besoin d'un module de synchronisation chaotique. Deux contributions y sont faites. La première contribution porte sur une technique de phase aléatoire proposée dans le but de rendre le signal chaotique LPI. La deuxième contribution porte sur la modification du décodeur de Viterbi employé par [2] pour démoduler un signal chaotique dont le modulateur est basé sur la symbolique dynamique. Cette modification permet d'adapter le décodeur de Viterbi à la mémoire du signal reçu induite par la technique de phase aléatoire en combinaison avec la réponse à l'impulsion du filtre de mise en forme.

La technique de phase aléatoire proposée atténue le degré de cyclostationnarité d'ordre deux du signal modulé chaotiquement fonction du taux de symbole. L'impossibilité d'éliminer complètement les raies spectrales à la fréquence de symbole vient du fait que le paramètre T_0 qui sert à définir la réponse à l'impulsion du filtre de mise en forme reste constant.

Le démodulateur chaotique proposé est modélisé dans l'environnement de simulation Matlab et son implémentation matérielle est réalisée dans un FPGA Xilinx de type Virtex-4. Les performances du système de communication évaluées en termes de BER pour la fonction chaotique du modulateur démontrent de bonnes performances considérant la nature LPI du système. De plus, il est possible de s'approcher d'une modulation BPSK ainsi que de sa performance en modifiant le paramètre p de la fonction chaotique du modulateur. Plus le paramètre p augmente, plus la distance euclidienne entre les symboles chaotiques de la séquence générée augmente, entraînant ainsi de meilleures performances du système. Évidemment, ce gain en performance est fait au détriment d'une moins bonne robustesse du signal en termes de sécurité.

L'architecture de l'implémentation matérielle démontre que le module BMU du décodeur de Viterbi est le plus critique en occupation de surface et en vitesse. Ceci vient du fait que la majorité des calculs du démodulateur chaotique sont effectués par ce module. Enfin, une co-simulation Matlab/System Generator vient confirmer le fonctionnement réel de l'implémentation du démodulateur chaotique sur un circuit FPGA.

RECOMMANDATIONS

Les futurs travaux pourraient porter dans un premier temps sur l'exploration de nouvelles méthodes d'introduction de phase aléatoire afin d'éliminer complètement la cyclostationnarité du signal. Dans un deuxième temps, des tests de cyclostationnarité d'ordre supérieur ("Higher-Order Cyclostationarity Statistic", HOCS) pourraient être effectuées afin de mieux caractériser le signal chaotique. Finalement, l'utilisation de techniques d'étalement de spectre et de codage pourrait être envisagée afin de réduire la détectabilité du signal chaotique.

ANNEXE I

RAPPORT APRÈS IMPLÉMENTATION ET ROUTAGE DU DÉMODULATEUR CHAOTIQUE ET DE SES MODULES

Rapport 1 Démodulateur chaotique

Release 7.1i par H.38

Copyright (c) 1995-2005 Xilinx, Inc. All rights reserved.

LACIME0355278:: Sat Feb 13 17:48:02 2010

```
par -w -intstyle ise -ol std -t 1
new_mod_2latence14clean_original_clk_wrapper_map.ncd
new_mod_2latence14clean_original_clk_wrapper.ncd
new_mod_2latence14clean_original_clk_wrapper.pcf
```

Constraints file: new_mod_2latence14clean_original_clk_wrapper.pcf.
Loading device for application Rf_Device from file '4vsx35.nph' in environment
C:/Xilinx.

"new_mod_2latence14clean_original_clk_wrapper" is an NCD, version 3.1, device
xc4vsx35, package ff668, speed -10

This design is using the default stepping level (major silicon revision) for
this device (0). Unless your design is targeted at devices of this stepping
level, it is recommended that you explicitly specify the stepping level of the
parts you will be using. This will allow the tools to take advantage of any
available performance and functional enhancements for this device. The latest
stepping level for this device is '1'. Additional information on "stepping
level" is available at support.xilinx.com.

Initializing temperature to 85.000 Celsius. (default - Range: 0.000 to 85.000
Celsius)

Initializing voltage to 1.425 Volts. (default - Range: 1.425 to 1.575 Volts)

WARNING:Timing:2666 - Constraint ignored:

```
TS_ce_7_fd49bc4c_group_to_ce_7_fd49bc4c_group = MAXDELAY FROM TIMEGRP
"ce_7_fd49bc4c_group" TO TIMEGRP "ce_7_fd49bc4c_group" 98 ns;
```

WARNING:Timing:2666 - Constraint ignored:

```
TS_ce_14_fd49bc4c_group_to_ce_7_fd49bc4c_group = MAXDELAY FROM TIMEGRP
"ce_14_fd49bc4c_group" TO TIMEGRP "ce_7_fd49bc4c_group" 98 ns;
```

Device speed data version: "ADVANCED 1.52 2005-01-22".

Device Utilization Summary:

Number of BUFGs	1 out of 32	3%
Number of DCM_ADVs	8 out of 8	100%
Number of ILOGICs	13 out of 448	2%
Number of External IOBs	17 out of 448	3%
Number of LOCed IOBs	1 out of 17	5%
Number of PMVs	1 out of 1	100%
Number of RAMB16s	8 out of 192	4%
Number of Slices	10503 out of 15360	68%
Number of SLICEMs	84 out of 7680	1%

Overall effort level (-ol): Standard (set by user)

Placer effort level (-pl): Standard (set by user)

Placer cost table entry (-t): 1

Router effort level (-rl): Standard (set by user)

Starting initial Timing Analysis. REAL time: 18 secs

Finished initial Timing Analysis. REAL time: 19 secs

WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_1 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_1 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_2 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_2 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_3 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_3 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_4 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_4 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_5 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_5 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_6 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_6 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_7 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_7 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_8 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_8 has no load

Starting Placer

Phase 1.1

Phase 1.1 (Checksum:99bc1b) REAL time: 20 secs

Phase 2.31

Phase 2.31 (Checksum:1312cfe) REAL time: 20 secs

Phase 3.2

Phase 3.2 (Checksum:1c9c37d) REAL time: 44 secs

Phase 4.30

Phase 4.30 (Checksum:26259fc) REAL time: 44 secs

Phase 5.3

Phase 5.3 (Checksum:2faf07b) REAL time: 49 secs

Phase 6.5

Phase 6.5 (Checksum:39386fa) REAL time: 49 secs

Phase 7.8

.....
Phase 7.8 (Checksum:1852441) REAL time: 1 mins 26 secs

Phase 8.5

Phase 8.5 (Checksum:4c4b3f8) REAL time: 1 mins 26 secs

Phase 9.18

Phase 9.18 (Checksum:55d4a77) REAL time: 1 mins 33 secs

Phase 10.27

Phase 10.27 (Checksum:5f5e0f6) REAL time: 1 mins 39 secs

Phase 11.5

Phase 11.5 (Checksum:68e7775) REAL time: 1 mins 39 secs

Writing design to file new_mod_2latence14clean_original_clk_wrapper.ncd

Total REAL time to Placer completion: 1 mins 50 secs

Total CPU time to Placer completion: 1 mins 45 secs

Starting Router

Phase 1: 73613 unrouted; REAL time: 1 mins 51 secs

Phase 2: 63880 unrouted; REAL time: 1 mins 58 secs

Phase 3: 18299 unrouted; REAL time: 2 mins 26 secs

Phase 4: 18299 unrouted; (2121798) REAL time: 2 mins 28 secs

Phase 5: 19318 unrouted; (1680) REAL time: 7 mins 43 secs

Phase 6: 19316 unrouted; (0) REAL time: 7 mins 46 secs

Phase 7: 0 unrouted; (0) REAL time: 8 mins 25 secs

Phase 8: 0 unrouted; (0) REAL time: 8 mins 32 secs

Phase 9: 0 unrouted; (0) REAL time: 9 mins 29 secs

Total REAL time to Router completion: 9 mins 31 secs

Total CPU time to Router completion: 9 mins 26 secs

Generating "PAR" statistics.

Generating Clock Report

Clock Net	Resource	Locked	Fanout	Net Skew(ns)	Max Delay(ns)
clk_BUFGP	BUFGCTRL_X0Y20	No	7209	0.387	3.721

The Delay Summary Report

The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0

The AVERAGE CONNECTION DELAY for this design is: 1.653

The MAXIMUM PIN DELAY IS: 9.788

The AVERAGE CONNECTION DELAY on the 10 WORST NETS is: 9.148

Listing Pin Delays by value: (nsec)

d < 2.00	< d < 4.00	< d < 6.00	< d < 8.00	< d < 10.00	d >= 10.00
51077	14246	4244	1368	147	0

Timing Score: 0

Asterisk (*) preceding a constraint indicates it was not met.
This may be due to a setup or hold violation.

Constraint	Requested	Actual	Logic Levels
TS_clk_fd49bc4c = PERIOD TIMEGRP "clk_fd49bc4c" 14 ns HIGH 50%	14.000ns	13.956ns	22
TS_ce_14_fd49bc4c_group_to_ce_14_fd49bc4c_group = MAXDELAY FROM TIMEGRP "ce_14_fd49bc4c_group" TO TIMEGRP "ce_14_fd49bc4c_group" 196 ns	196.000ns	6.137ns	3
TS_ce_7_fd49bc4c_group_to_ce_7_fd49bc4c_group = MAXDELAY FROM TIMEGRP "ce_7_fd49bc4c_group" TO TIMEGRP "ce_7_fd49bc4c_group" 98 ns	N/A	N/A	N/A
TS_ce_14_fd49bc4c_group_to_ce_7_fd49bc4c_group = MAXDELAY FROM TIMEGRP "ce_14_fd49bc4c_group" TO TIMEGRP "ce_7_fd49bc4c_group" 98 ns	N/A	N/A	N/A
TS_ce_7_fd49bc4c_group_to_ce_14_fd49bc4c_group = MAXDELAY FROM TIMEGRP "ce_7_fd49bc4c_group" TO TIMEGRP "ce_14_fd49bc4c_group" 98 ns	98.000ns	5.952ns	2

All constraints were met.

INFO:Timing:2761 - N/A entries in the Constraints list may indicate that the constraint does not cover any paths or that it has no requested value.
Generating Pad Report.

All signals are completely routed.

Total REAL time to PAR completion: 9 mins 55 secs

Total CPU time to PAR completion: 9 mins 36 secs

Peak Memory Usage: 599 MB

Placement: Completed - No errors found.

Routing: Completed - No errors found.

Timing: Completed - No errors found.

Number of error messages: 0

Number of warning messages: 18

Number of info messages: 0

Writing design to file new_mod_2latence14clean_original_clk_wrapper.ncd

PAR done!

Rapport 2 Module S/P

Release 7.1i par H.38

Copyright (c) 1995-2005 Xilinx, Inc. All rights reserved.

LACIME0355278:: Mon Feb 15 13:53:42 2010

```
par -w -intstyle ise -ol std -t 1
new_mod_2latence14clean_serialpara2_clk_wrapper_map.ncd
new_mod_2latence14clean_serialpara2_clk_wrapper.ncd
new_mod_2latence14clean_serialpara2_clk_wrapper.pcf
```

Constraints file: new_mod_2latence14clean_serialpara2_clk_wrapper.pcf.

Loading device for application Rf_Device from file '4vsx35.nph' in environment C:/Xilinx.

"new_mod_2latence14clean_serialpara2_clk_wrapper" is an NCD, version 3.1, device xc4vsx35, package ff668, speed -10

This design is using the default stepping level (major silicon revision) for this device (0). Unless your design is targeted at devices of this stepping level, it is recommended that you explicitly specify the stepping level of the parts you will be using. This will allow the tools to take advantage of any available performance and functional enhancements for this device. The latest stepping level for this device is '1'. Additional information on "stepping level" is available at support.xilinx.com.

Initializing temperature to 85.000 Celsius. (default - Range: 0.000 to 85.000 Celsius)

Initializing voltage to 1.425 Volts. (default - Range: 1.425 to 1.575 Volts)

WARNING:Timing:2666 - Constraint ignored:

TS_ce_7_fafbb562_group_to_ce_7_fafbb562_group = MAXDELAY FROM TIMEGRP
 "ce_7_fafbb562_group" TO TIMEGRP "ce_7_fafbb562_group" 98 ns;

Device speed data version: "ADVANCED 1.52 2005-01-22".

Device Utilization Summary:

Number of BUFGs	1 out of 32	3%
Number of DCM_ADVs	8 out of 8	100%
Number of ILOGICs	13 out of 448	2%
Number of External IOBs	41 out of 448	9%
Number of LOCed IOBs	1 out of 41	2%
Number of PMVs	1 out of 1	100%
Number of Slices	6 out of 15360	1%
Number of SLICEMs	0 out of 7680	0%

Overall effort level (-ol): Standard (set by user)

Placer effort level (-pl): Standard (set by user)

Placer cost table entry (-t): 1

Router effort level (-rl): Standard (set by user)

Starting initial Timing Analysis. REAL time: 2 secs

Finished initial Timing Analysis. REAL time: 2 secs

WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_1 has no load

WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_1 has no load

WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_2 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_2 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_3 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_3 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_4 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_4 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_5 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_5 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_6 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_6 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_7 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_7 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_8 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_8 has no load

Starting Placer

Phase 1.1

Phase 1.1 (Checksum:989797) REAL time: 2 secs

Phase 2.31

Phase 2.31 (Checksum:1312cfe) REAL time: 2 secs

Phase 3.2

Phase 3.2 (Checksum:1c9c37d) REAL time: 18 secs

Phase 4.30

Phase 4.30 (Checksum:26259fc) REAL time: 18 secs

Phase 5.3

Phase 5.3 (Checksum:2faf07b) REAL time: 18 secs

Phase 6.5

Phase 6.5 (Checksum:39386fa) REAL time: 18 secs

Phase 7.8

Phase 7.8 (Checksum:9a8e94) REAL time: 18 secs

Phase 8.5

Phase 8.5 (Checksum:4c4b3f8) REAL time: 18 secs

Phase 9.18

Phase 9.18 (Checksum:55d4a77) REAL time: 18 secs

Phase 10.27

Phase 10.27 (Checksum:5f5e0f6) REAL time: 18 secs

Phase 11.5

Phase 11.5 (Checksum:68e7775) REAL time: 18 secs

Writing design to file new_mod_2latence14clean_serialpara2_clk_wrapper.ncd

Total REAL time to Placer completion: 18 secs

Total CPU time to Placer completion: 18 secs

Starting Router

Phase 1: 131 unrouted; REAL time: 19 secs

Phase 2: 96 unrouted; REAL time: 25 secs

Phase 3: 0 unrouted; REAL time: 25 secs

Phase 4: 0 unrouted; (0) REAL time: 25 secs

Phase 5: 0 unrouted; (0) REAL time: 25 secs

Phase 6: 0 unrouted; (0) REAL time: 25 secs

Phase 7: 0 unrouted; (0) REAL time: 25 secs

Phase 8: 0 unrouted; (0) REAL time: 25 secs

Total REAL time to Router completion: 25 secs

Total CPU time to Router completion: 25 secs

Generating "PAR" statistics.

Generating Clock Report

Clock Net	Resource	[Locked]	Fanout	Net Skew(ns)	Max Delay(ns)
clk_BUFGP	BUFGCTRL_X0Y20	No	19	0.393	3.720

The Delay Summary Report

The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0

The AVERAGE CONNECTION DELAY for this design is: 1.768

The MAXIMUM PIN DELAY IS: 5.769

The AVERAGE CONNECTION DELAY on the 10 WORST NETS is: 2.504

Listing Pin Delays by value: (nsec)

d < 1.00	< d < 2.00	< d < 3.00	< d < 4.00	< d < 6.00	d >= 6.00
34	40	12	28	2	0

Timing Score: 0

Asterisk (*) preceding a constraint indicates it was not met.

This may be due to a setup or hold violation.

Constraint	Requested.	Actual	Logic
			Levels
TS_clk_fafbb562 = PERIOD TIMEGRP "clk_fafbb562" 14 ns HIGH 50%	14.000ns	6.907ns	0
TS_ce_7_fafbb562_group_to_ce_7_fafbb562_group = MAXDELAY FROM TIMEGRP "ce_7_fafbb562_group" TO TIMEGRP "ce_7_fafbb562_group" 98 ns	N/A	N/A	N/A

All constraints were met.

INFO:Timing:2761 - N/A entries in the Constraints list may indicate that the constraint does not cover any paths or that it has no requested value.

Generating Pad Report.

All signals are completely routed.

Total REAL time to PAR completion: 40 secs

Total CPU time to PAR completion: 26 secs

Peak Memory Usage: 226 MB

Placement: Completed - No errors found.

Routing: Completed - No errors found.

Timing: Completed - No errors found.

Number of error messages: 0

Number of warning messages: 17

Number of info messages: 0

Writing design to file new_mod_2latence14clean_serialpara2_clk_wrapper.ncd

PAR done!

Rapport 3 Module BMU

Release 7.1i par H.38

Copyright (c) 1995-2005 Xilinx, Inc. All rights reserved.

LACIME0355278:: Mon Feb 15 14:55:55 2010

par -w -intstyle ise -ol std -t 1

new_mod_2latence14clean_original_genseq_serial_clk_wrapper_map.ncd

new_mod_2latence14clean_original_genseq_serial_clk_wrapper.ncd

new_mod_2latence14clean_original_genseq_serial_clk_wrapper.pcf

Constraints file:

new_mod_2latence14clean_original_genseq_serial_clk_wrapper.pcf.

Loading device for application Rf_Device from file '4vsx35.nph' in environment C:/Xilinx.

"new_mod_2latence14clean_original_genseq_serial_clk_wrapper" is an NCD, version 3.1, device xc4vsx35, package ff668, speed -10

This design is using the default stepping level (major silicon revision) for this device (0). Unless your design is targeted at devices of this stepping level, it is recommended that you explicitly specify the stepping level of the parts you will be using. This will allow the tools to take advantage of any available performance and functional enhancements for this device. The latest stepping level for this device is '1'. Additional information on "stepping level" is available at support.xilinx.com.

Initializing temperature to 85.000 Celsius. (default - Range: 0.000 to 85.000 Celsius)

Initializing voltage to 1.425 Volts. (default - Range: 1.425 to 1.575 Volts)

Device speed data version: "ADVANCED 1.52 2005-01-22".

Device Utilization Summary:

Number of BUFGs	1 out of 32	3%
Number of DCM_ADVs	8 out of 8	100%
Number of External IOBs	431 out of 448	96%
Number of LOCed IOBs	1 out of 431	1%
Number of OLOGICs	1 out of 448	1%
Number of PMVs	1 out of 1	100%
Number of RAMB16s	2 out of 192	1%
Number of Slices	9907 out of 15360	64%
Number of SLICEMs	67 out of 7680	1%

Overall effort level (-ol): Standard (set by user)

Placer effort level (-pl): Standard (set by user)

Placer cost table entry (-t): 1

Router effort level (-rl): Standard (set by user)

Starting initial Timing Analysis. REAL time: 17 secs

Finished initial Timing Analysis. REAL time: 17 secs

WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_1 has no load

WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_1 has no load

WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_2 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_2 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_3 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_3 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_4 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_4 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_5 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_5 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_6 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_6 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_7 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_7 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_8 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_8 has no load

Starting Placer

Phase 1.1

Phase 1.1 (Checksum:997e0b) REAL time: 18 secs

Phase 2.31

Phase 2.31 (Checksum:1312cfe) REAL time: 18 secs

Phase 3.2

Phase 3.2 (Checksum:1c9c37d) REAL time: 38 secs

Phase 4.30

Phase 4.30 (Checksum:26259fc) REAL time: 38 secs

Phase 5.3

Phase 5.3 (Checksum:2faf07b) REAL time: 40 secs

Phase 6.5

Phase 6.5 (Checksum:39386fa) REAL time: 40 secs

Phase 7.8

.....

Phase 7.8 (Checksum:1243ebe) REAL time: 1 mins 9 secs

Phase 8.5

Phase 8.5 (Checksum:4c4b3f8) REAL time: 1 mins 10 secs

Phase 9.18

Phase 9.18 (Checksum:55d4a77) REAL time: 1 mins 17 secs

Phase 10.27

Phase 10.27 (Checksum:5f5e0f6) REAL time: 1 mins 22 secs

Phase 11.5

Phase 11.5 (Checksum:68e7775) REAL time: 1 mins 22 secs

Writing design to file

new_mod_2latence14clean_original_genseq_serial_clk_wrapper.ncd

Total REAL time to Placer completion: 1 mins 31 secs

Total CPU time to Placer completion: 1 mins 27 secs

Starting Router

Phase 1: 68878 unrouted; REAL time: 1 mins 32 secs

Phase 2: 59898 unrouted; REAL time: 1 mins 39 secs

Phase 3: 16997 unrouted; REAL time: 2 mins 4 secs

Phase 4: 16997 unrouted; (772084) REAL time: 2 mins 5 secs

Phase 5: 17098 unrouted; (16240) REAL time: 3 mins 13 secs

Phase 6: 17138 unrouted; (0) REAL time: 3 mins 18 secs

Phase 7: 0 unrouted; (0) REAL time: 3 mins 37 secs

Phase 8: 0 unrouted; (0) REAL time: 3 mins 43 secs

Phase 9: 0 unrouted; (0) REAL time: 4 mins 27 secs

Total REAL time to Router completion: 4 mins 28 secs

Total CPU time to Router completion: 4 mins 24 secs

Generating "PAR" statistics.

Generating Clock Report

Clock Net	Resource	Locked	Fanout	Net Skew(ns)	Max Delay(ns)
clk_BUFGP	BUFGCTRL_X0Y20	No	6797	0.408	3.729

The Delay Summary Report

The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0

The AVERAGE CONNECTION DELAY for this design is: 1.707

The MAXIMUM PIN DELAY IS: 8.955

The AVERAGE CONNECTION DELAY on the 10 WORST NETS is: 8.360

Listing Pin Delays by value: (nsec)

d < 2.00 < d < 4.00 < d < 6.00 < d < 8.00 < d < 9.00 d >= 9.00

44891 16597 4238 939 29 0

Timing Score: 0

Asterisk (*) preceding a constraint indicates it was not met.

This may be due to a setup or hold violation.

Constraint	Requested	Actual	Logic
			Levels
TS_clk_9917fbbf= PERIOD TIMEGRP "clk_991	14.000ns	13.825ns	12
7fbbf" 14 ns HIGH 50%			

All constraints were met.

Generating Pad Report.

All signals are completely routed.

Total REAL time to PAR completion: 4 mins 48 secs

Total CPU time to PAR completion: 4 mins 30 secs

Peak Memory Usage: 567 MB

Placement: Completed - No errors found.

Routing: Completed - No errors found.

Timing: Completed - No errors found.

Number of error messages: 0

Number of warning messages: 16

Number of info messages: 0

Writing design to file

new_mod_2latence14clean_original_genseq_serial_clk_wrapper.ncd

PAR done!

Rapport 4 Module ACS

Release 7.1i par H.38

Copyright (c) 1995-2005 Xilinx, Inc. All rights reserved.

LACIME029650:: Thu Sep 09 00:05:24 2010

par -w -intstyle ise -ol std -t 1

new_mod_2latence14clean_original2_subsystem1_clk_wrapper_map.ncd

new_mod_2latence14clean_original2_subsystem1_clk_wrapper.ncd

new_mod_2latence14clean_original2_subsystem1_clk_wrapper.pcf

Constraints file: new_mod_2latence14clean_original2_subsystem1_clk_wrapper.pcf.

Loading device for application Rf_Device from file '4vsx35.nph' in environment C:/Xilinx71.

"new_mod_2latence14clean_original2_subsystem1_clk_wrapper" is an NCD, version 3.1, device xc4vsx35, package ff668, speed -10

This design is using the default stepping level (major silicon revision) for this device (0). Unless your design is targeted at devices of this stepping level, it is recommended that you explicitly specify the stepping level of the parts you will be using. This will allow the tools to take advantage of any available performance and functional enhancements for this device. The latest stepping level for this device is '1'. Additional information on "stepping level" is available at support.xilinx.com.

Initializing temperature to 85.000 Celsius. (default - Range: 0.000 to 85.000 Celsius)

Initializing voltage to 1.425 Volts. (default - Range: 1.425 to 1.575 Volts)

Device speed data version: "ADVANCED 1.52 2005-01-22".

Device Utilization Summary:

Number of BUFGs	1 out of 32	3%
Number of DCM_ADVs	8 out of 8	100%
Number of ILOGICs	1 out of 448	1%
Number of External IOBs	403 out of 448	89%
Number of LOCed IOBs	1 out of 403	1%
Number of OLOGICs	16 out of 448	3%
Number of PMVs	1 out of 1	100%
Number of Slices	502 out of 15360	3%
Number of SLICEMs	1 out of 7680	1%

Overall effort level (-ol): Standard (set by user)
 Placer effort level (-pl): Standard (set by user)
 Placer cost table entry (-t): 1
 Router effort level (-rl): Standard (set by user)

Starting initial Timing Analysis. REAL time: 7 secs
 Finished initial Timing Analysis. REAL time: 7 secs

WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_1 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_1 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_2 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_2 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_3 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_3 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_4 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_4 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_5 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_5 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_6 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_6 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_7 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_7 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_8 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_8 has no load

Starting Placer

Phase 1.1
 Phase 1.1 (Checksum:98a6e1) REAL time: 9 secs

Phase 2.31
 Phase 2.31 (Checksum:1312cfe) REAL time: 9 secs

Phase 3.2

Phase 3.2 (Checksum:1c9c37d) REAL time: 52 secs

Phase 4.30
 Phase 4.30 (Checksum:26259fc) REAL time: 52 secs

Phase 5.3

Phase 5.3 (Checksum:2faf07b) REAL time: 55 secs

Phase 6.5

Phase 6.5 (Checksum:39386fa) REAL time: 55 secs

Phase 7.8

.

..

Phase 7.8 (Checksum:ea913c) REAL time: 56 secs

Phase 8.5

Phase 8.5 (Checksum:4c4b3f8) REAL time: 56 secs

Phase 9.18

Phase 9.18 (Checksum:55d4a77) REAL time: 57 secs

Phase 10.27

Phase 10.27 (Checksum:5f5e0f6) REAL time: 57 secs

Phase 11.5

Phase 11.5 (Checksum:68e7775) REAL time: 57 secs

Writing design to file

new_mod_2latence14clean_original2_subsystem1_clk_wrapper.ncd

Total REAL time to Placer completion: 58 secs

Total CPU time to Placer completion: 57 secs

Starting Router

Phase 1: 4101 unrouted; REAL time: 1 mins

Phase 2: 3707 unrouted; REAL time: 1 mins 14 secs

Phase 3: 949 unrouted; REAL time: 1 mins 15 secs

Phase 4: 949 unrouted; (0) REAL time: 1 mins 15 secs

Phase 5: 949 unrouted; (0) REAL time: 1 mins 15 secs

Phase 6: 949 unrouted; (0) REAL time: 1 mins 16 secs

Phase 7: 0 unrouted; (0) REAL time: 1 mins 17 secs

Phase 8: 0 unrouted; (0) REAL time: 1 mins 19 secs

Phase 9: 0 unrouted; (0) REAL time: 1 mins 21 secs

Total REAL time to Router completion: 1 mins 21 secs

Total CPU time to Router completion: 1 mins 19 secs

Generating "PAR" statistics.

Generating Clock Report

Clock Net	Resource	Locked	Fanout	Net Skew(ns)	Max Delay(ns)
clk_BUFGP	BUFGCTRL_X0Y20	No	324	0.417	3.744

The Delay Summary Report

The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0

The AVERAGE CONNECTION DELAY for this design is: 1.751

The MAXIMUM PIN DELAY IS: 7.200

The AVERAGE CONNECTION DELAY on the 10 WORST NETS is: 4.589

Listing Pin Delays by value: (nsec)

d < 1.00	< d < 2.00	< d < 3.00	< d < 4.00	< d < 8.00	d >= 8.00
1316	1479	493	421	323	0

Timing Score: 0

Asterisk (*) preceding a constraint indicates it was not met.
This may be due to a setup or hold violation.

Constraint	Requested	Actual	Logic Levels
TS_clk_61cec82e = PERIOD TIMEGRP "clk_61c ec82e" 14 ns HIGH 50%	14.000ns	7.796ns	19

All constraints were met.
Generating Pad Report.

All signals are completely routed.

Total REAL time to PAR completion: 1 mins 25 secs
Total CPU time to PAR completion: 1 mins 23 secs

Peak Memory Usage: 207 MB

Placement: Completed - No errors found.
Routing: Completed - No errors found.
Timing: Completed - No errors found.

Number of error messages: 0
Number of warning messages: 16
Number of info messages: 0

Writing design to file
new_mod_2latence14clean_original2_subsystem1_clk_wrapper.ncd

PAR done!

Rapport 5 Module TBU

Release 7.1i par H.38

Copyright (c) 1995-2005 Xilinx, Inc. All rights reserved.

LACIME0355278:: Mon Feb 15 13:00:48 2010

```
par -w -intstyle ise -ol std -t 1
new_mod_2latence14clean_subsystem_clk_wrapper_map.ncd
new_mod_2latence14clean_subsystem_clk_wrapper.ncd
new_mod_2latence14clean_subsystem_clk_wrapper.pcf
```

Constraints file: new_mod_2latence14clean_subsystem_clk_wrapper.pcf.
Loading device for application Rf_Device from file '4vsx35.nph' in environment C:/Xilinx.

"new_mod_2latence14clean_subsystem_clk_wrapper" is an NCD, version 3.1, device xc4vsx35, package ff668, speed -10

This design is using the default stepping level (major silicon revision) for this device (0). Unless your design is targeted at devices of this stepping level, it is recommended that you explicitly specify the stepping level of the parts you will be using. This will allow the tools to take advantage of any available performance and functional enhancements for this device. The latest stepping level for this device is '1'. Additional information on "stepping level" is available at support.xilinx.com.

Initializing temperature to 85.000 Celsius. (default - Range: 0.000 to 85.000 Celsius)

Initializing voltage to 1.425 Volts. (default - Range: 1.425 to 1.575 Volts)

Device speed data version: "ADVANCED 1.52 2005-01-22".

Device Utilization Summary:

Number of BUFGs	1 out of 32	3%
Number of DCM_ADVs	8 out of 8	100%
Number of External IOBs	27 out of 448	6%
Number of LOCed IOBs	1 out of 27	3%
Number of OLOGICs	6 out of 448	1%
Number of PMVs	1 out of 1	100%
Number of RAMB16s	4 out of 192	2%
Number of Slices	53 out of 15360	1%
Number of SLICEMs	8 out of 7680	1%

Overall effort level (-ol): Standard (set by user)
 Placer effort level (-pl): Standard (set by user)
 Placer cost table entry (-t): 1
 Router effort level (-rl): Standard (set by user)

Starting initial Timing Analysis. REAL time: 2 secs
 Finished initial Timing Analysis. REAL time: 2 secs

WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_1 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_1 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_2 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_2 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_3 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_3 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_4 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_4 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_5 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_5 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_6 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_6 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_7 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_7 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_8 has no load
 WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_8 has no load

Starting Placer

Phase 1.1
 Phase 1.1 (Checksum:9899d1) REAL time: 2 secs

Phase 2.31
 Phase 2.31 (Checksum:1312cfe) REAL time: 2 secs

Phase 3.2

Phase 3.2 (Checksum:1c9c37d) REAL time: 19 secs

Phase 4.30
 Phase 4.30 (Checksum:26259fc) REAL time: 19 secs

Phase 5.3

Phase 5.3 (Checksum:2faf07b) REAL time: 19 secs

Phase 6.5

Phase 6.5 (Checksum:39386fa) REAL time: 19 secs

Phase 7.8

.

.....

Phase 7.8 (Checksum:9f07bc) REAL time: 19 secs

Phase 8.5

Phase 8.5 (Checksum:4c4b3f8) REAL time: 19 secs

Phase 9.18

Phase 9.18 (Checksum:55d4a77) REAL time: 19 secs

Phase 10.27

Phase 10.27 (Checksum:5f5e0f6) REAL time: 19 secs

Phase 11.5

Phase 11.5 (Checksum:68e7775) REAL time: 19 secs

Writing design to file new_mod_2latence14clean_subsystem_clk_wrapper.ncd

Total REAL time to Placer completion: 19 secs

Total CPU time to Placer completion: 19 secs

Starting Router

Phase 1: 708 unrouted; REAL time: 20 secs

Phase 2: 488 unrouted; REAL time: 26 secs

Phase 3: 147 unrouted; REAL time: 26 secs

Phase 4: 147 unrouted; (0) REAL time: 26 secs

Phase 5: 147 unrouted; (0) REAL time: 26 secs

Phase 6: 147 unrouted; (0) REAL time: 26 secs

Phase 7: 0 unrouted; (0) REAL time: 26 secs

Phase 8: 0 unrouted; (0) REAL time: 26 secs

Phase 9: 0 unrouted; (0) REAL time: 26 secs

Total REAL time to Router completion: 26 secs

Total CPU time to Router completion: 26 secs

Generating "PAR" statistics.

Generating Clock Report

Clock Net	Resource	Locked	Fanout	Net Skew(ns)	Max Delay(ns)
clk_BUFGP	BUFGCTRL_X0Y20	No	33	0.198	3.721

The Delay Summary Report

The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0

The AVERAGE CONNECTION DELAY for this design is: 1.183

The MAXIMUM PIN DELAY IS: 3.721

The AVERAGE CONNECTION DELAY on the 10 WORST NETS is: 2.808

Listing Pin Delays by value: (nsec)

d < 1.00 < d < 2.00 < d < 3.00 < d < 4.00 < d < 5.00 d >= 5.00

282	162	40	34	0	0
-----	-----	----	----	---	---

Timing Score: 0

Asterisk (*) preceding a constraint indicates it was not met.
This may be due to a setup or hold violation.

Constraint	Requested	Actual	Logic Levels
TS_clk_43e5adaf = PERIOD TIMEGRP "clk_43e5adaf" 14 ns HIGH 50%	14.000ns	4.322ns	1
TS_ce_14_43e5adaf_group_to_ce_14_43e5adaf_group = MAXDELAY FROM TIMEGRP "ce_14_43e5adaf_group" TO TIMEGRP "ce_14_43e5adaf_group" 196 ns	196.000ns	5.464ns	4

All constraints were met.
Generating Pad Report.

All signals are completely routed.

Total REAL time to PAR completion: 43 secs
Total CPU time to PAR completion: 27 secs

Peak Memory Usage: 188 MB

Placement: Completed - No errors found.
Routing: Completed - No errors found.
Timing: Completed - No errors found.

Number of error messages: 0
Number of warning messages: 16
Number of info messages: 0

Writing design to file new_mod_2latence14clean_subsystem_clk_wrapper.ncd

PAR done!

Rapport 6 Module LIFO

Release 7.1i par H.38

Copyright (c) 1995-2005 Xilinx, Inc. All rights reserved.

LACIME0355278:: Mon Feb 15 13:06:10 2010

```
par -w -intstyle ise -ol std -t 1
new_mod_2latence14clean_lifo1_clk_wrapper_map.ncd
new_mod_2latence14clean_lifo1_clk_wrapper.ncd
new_mod_2latence14clean_lifo1_clk_wrapper.pcf
```

Constraints file: new_mod_2latence14clean_lifo1_clk_wrapper.pcf.

Loading device for application Rf_Device from file '4vsx35.nph' in environment C:/Xilinx.

"new_mod_2latence14clean_lifo1_clk_wrapper" is an NCD, version 3.1, device xc4vsx35, package ff668, speed -10

This design is using the default stepping level (major silicon revision) for this device (0). Unless your design is targeted at devices of this stepping level, it is recommended that you explicitly specify the stepping level of the parts you will be using. This will allow the tools to take advantage of any available performance and functional enhancements for this device. The latest stepping level for this device is '1'. Additional information on "stepping level" is available at support.xilinx.com.

Initializing temperature to 85.000 Celsius. (default - Range: 0.000 to 85.000 Celsius)

Initializing voltage to 1.425 Volts. (default - Range: 1.425 to 1.575 Volts)

WARNING:Timing:2666 - Constraint ignored:

TS_ce_14_c472b1dd_group_to_ce_14_c472b1dd_group = MAXDELAY FROM
TIMEGRP

"ce_14_c472b1dd_group" TO TIMEGRP "ce_14_c472b1dd_group" 196 ns;

Device speed data version: "ADVANCED 1.52 2005-01-22".

Device Utilization Summary:

Number of BUFGs	1 out of 32	3%
Number of DCM_ADVs	8 out of 8	100%
Number of External IOBs	11 out of 448	2%
Number of LOCed IOBs	1 out of 11	9%
Number of PMVs	1 out of 1	100%
Number of RAMB16s	2 out of 192	1%
Number of Slices	9 out of 15360	1%
Number of SLICEMs	0 out of 7680	0%

Overall effort level (-ol): Standard (set by user)

Placer effort level (-pl): Standard (set by user)

Placer cost table entry (-t): 1

Router effort level (-rl): Standard (set by user)

Starting initial Timing Analysis. REAL time: 2 secs

Finished initial Timing Analysis. REAL time: 2 secs

WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_1 has no load

WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_1 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_2 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_2 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_3 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_3 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_4 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_4 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_5 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_5 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_6 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_6 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_7 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_7 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKFX_8 has no load
WARNING:Par:276 - The signal XIL_ML_UNUSED_DCM_CLKOUT_8 has no load

Starting Placer

Phase 1.1

Phase 1.1 (Checksum:98971a) REAL time: 2 secs

Phase 2.31

Phase 2.31 (Checksum:1312cfe) REAL time: 2 secs

Phase 3.2

Phase 3.2 (Checksum:1c9c37d) REAL time: 18 secs

Phase 4.30

Phase 4.30 (Checksum:26259fc) REAL time: 18 secs

Phase 5.3

Phase 5.3 (Checksum:2faf07b) REAL time: 18 secs

Phase 6.5

Phase 6.5 (Checksum:39386fa) REAL time: 18 secs

Phase 7.8

Phase 7.8 (Checksum:9a2fd5) REAL time: 18 secs

Phase 8.5

Phase 8.5 (Checksum:4c4b3f8) REAL time: 18 secs

Phase 9.18

Phase 9.18 (Checksum:55d4a77) REAL time: 18 secs

Phase 10.27

Phase 10.27 (Checksum:5f5e0f6) REAL time: 18 secs

Phase 11.5

Phase 11.5 (Checksum:68e7775) REAL time: 18 secs

Writing design to file new_mod_2latence14clean_lifo1_clk_wrapper.ncd

Total REAL time to Placer completion: 18 secs

Total CPU time to Placer completion: 18 secs

Starting Router

Phase 1: 176 unrouted; REAL time: 19 secs

Phase 2: 88 unrouted; REAL time: 25 secs

Phase 3: 11 unrouted; REAL time: 25 secs

Phase 4: 11 unrouted; (0) REAL time: 25 secs

Phase 5: 11 unrouted; (0) REAL time: 25 secs

Phase 6: 11 unrouted; (0) REAL time: 25 secs

Phase 7: 0 unrouted; (0) REAL time: 25 secs

Phase 8: 0 unrouted; (0) REAL time: 25 secs

Phase 9: 0 unrouted; (0) REAL time: 25 secs

Total REAL time to Router completion: 25 secs

Total CPU time to Router completion: 25 secs

Generating "PAR" statistics.

Generating Clock Report

Clock Net	Resource	[Locked]	Fanout	Net Skew(ns)	Max Delay(ns)
clk_BUFGP	BUFGCTRL_X0Y20	No	10	0.086	3.505

The Delay Summary Report

The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0

The AVERAGE CONNECTION DELAY for this design is: 1.579

The MAXIMUM PIN DELAY IS: 4.020

The AVERAGE CONNECTION DELAY on the 10 WORST NETS is: 2.414

Listing Pin Delays by value: (nsec)

d < 1.00 < d < 2.00 < d < 3.00 < d < 4.00 < d < 5.00 d >= 5.00

-----	-----	-----	-----	-----	-----
32	37	17	10	1	0

Timing Score: 0

Asterisk (*) preceding a constraint indicates it was not met.

This may be due to a setup or hold violation.

Constraint	Requested	Actual	Logic Levels
TS_clk_c472b1dd = PERIOD TIMEGRP "clk_c472b1dd" 14 ns HIGH 50%	14.000ns	3.090ns	0
TS_ce_14_c472b1dd_group_to_ce_14_c472b1dd_group = MAXDELAY FROM TIMEGRP "ce_14_c472b1dd_group" TO TIMEGRP "ce_14_c472b1dd_group" 196 ns	N/A	N/A	N/A

All constraints were met.

INFO:Timing:2761 - N/A entries in the Constraints list may indicate that the constraint does not cover any paths or that it has no requested value.

Generating Pad Report.

All signals are completely routed.

Total REAL time to PAR completion: 40 secs

Total CPU time to PAR completion: 27 secs

Peak Memory Usage: 185 MB

Placement: Completed - No errors found.

Routing: Completed - No errors found.

Timing: Completed - No errors found.

Number of error messages: 0

Number of warning messages: 17

Number of info messages: 0

Writing design to file new_mod_2latence14clean_lifo1_clk_wrapper.ncd

PAR done!

BIBLIOGRAPHIE

- [1] Atwal, S., F. Gagnon, and C. Thibeault. *An LPI Wireless Communication System Based On Chaotic Modulation* in MILCOM 2009. 2009. Boston, MA.
- [2] Luengo, D. and I. Santamaria. *Secure communications using OFDM with chaotic modulation in the subcarriers*. in *Vehicular Technology Conference, 2005. VTC 2005-Spring*. 2005 IEEE 61st. 2005.
- [3] Orr¹, R., T. Farrell, and G. Prescott, *transform-based low probability of intercept communications*.
- [4] Lau, F. and C. Tse, *Chaos-based digital communication systems: operating principles, analysis methods, and performance evaluation*. 2003: Springer Verlag.
- [5] Silva, C. and A. Young. *Introduction to chaos-based communications and signal processing*. 2000.
- [6] Gardner, W., *Spectral Correlation of Modulated Signals: Part I--Analog Modulation*. Communications, IEEE Transactions on, 1987. 35(6): p. 584-594.
- [7] Proakis, J. and M. Salehi, *Digital communications*. 1995: McGraw-hill New York.
- [8] Eisencraft, M., D. Kato, and L. Monteiro, *Spectral properties of chaotic signals generated by the skew tent map*. Signal Processing. 90(1): p. 385-390.
- [9] Garcia, P. and J. Jimenez, *Communication through chaotic map systems*. Physics Letters A, 2002. 298(1): p. 35-40.
- [10] Gardner, W. and D. Cochran, *Cyclostationarity in communications and signal processing*. 1994: IEEE press Piscataway, NJ.
- [11] Lopez-Salcedo, J. and G. Vazquez. *Second-order cyclostationary approach to NDA ML square timing estimation with frequency uncertainty*.
- [12] Oerder, M. and H. Meyr, *Digital filter and square timing recovery*. IEEE transactions on communications, 1988. 36(5): p. 605-612.
- [13] Gardner, W., *Stationarizable random processes*. IEEE Transactions on Information Theory, 1978. 24(1): p. 8-22.
- [14] Ott, E., *Chaos in dynamical systems*. 2002: Cambridge Univ Pr.

- [15] Banks, J., et al., *On Devaney's definition of chaos*. The American mathematical monthly, 1992. 99(4): p. 332-334.
- [16] Morgül, Ö. and M. Feki, *A chaotic masking scheme by using synchronized chaotic systems*. Physics Letters A, 1999. 251(3): p. 169-176.
- [17] Parlitz, U., et al., *Transmission of digital signals by chaotic synchronization*. Int. J. Bifurcation and Chaos, 1992. 2(4): p. 973-977.
- [18] Ciftci, M. and D.B. Williams. *Optimal estimation for chaotic sequences using the Viterbi algorithm*. in *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*. 2001.
- [19] Oppenheim, A., et al. *Signal processing in the context of chaotic signals*. 1992.
- [20] Yang, T., *A survey of chaotic secure communication systems*. International Journal of Computational Cognition (<http://www.YangSky.com/yangijcc.htm>), 2004. 2(2): p. 81-130.
- [21] Pecora, L. and T. Carroll, *Driving systems with chaotic signals*. Physical Review A, 1991. 44(4): p. 2374-2383.
- [22] Kolumbán, G., M. Kennedy, and L. Chua, *The role of synchronization in digital communications using chaos—Part II: Chaotic modulation and chaotic synchronization*. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl, 1998. 45(11): p. 1129-1140.
- [23] Kennedy, M. and I. Motivation, *Three steps to chaos—Part I: Evolution*. chaos. 3: p. 4.
- [24] Pecora, L. and T. Carroll, *Synchronization in chaotic systems*. Physical review letters, 1990. 64(8): p. 821-824.
- [25] Pecora, L., et al., *Fundamentals of synchronization in chaotic systems, concepts, and applications*. Chaos: An Interdisciplinary Journal of Nonlinear Science, 1997. 7: p. 520.
- [26] Rulkov, N., et al., *Generalized synchronization of chaos in directionally coupled chaotic systems*. Physical Review E, 1995. 51(2): p. 980-994.
- [27] Kocarev, L., U. Parlitz, and T. Stojanovski. *Generalized synchronization of chaotic signals*. 1995.

- [28] Rosenblum, M., A. Pikovsky, and J. Kurths, *Phase synchronization of chaotic oscillators*. Physical review letters, 1996. 76(11): p. 1804-1807.
- [29] Parlitz, U., et al., *Experimental observation of phase synchronization*. Physical Review E, 1996. 54(2): p. 2115-2117.
- [30] Liu, F., et al., *A linear feedback synchronization theorem for a class of chaotic systems*. Chaos, solitons and Fractals, 2002. 13(4): p. 723-730.
- [31] Devaney, R., *An introduction to chaotic dynamical systems*. 2003: Westview Press.
- [32] Wikipedia. *Raised-cosine filter* 31 January 2010 16:27 UTC [cited 2010 2 February 18:46 UTC]; Available from: http://en.wikipedia.org/w/index.php?title=Raised-cosine_filter&oldid=341105579.
- [33] Gentile, K., *Digital Pulse-Shaping Filter Basics*. Analog Devices, AN-922 Application Note.
- [34] Bensadek, K., *Developpement d'un modele VHDL synthetisable d'un decodeur de Viterbi*. 2004, Ecole de Technologie Superieure (Canada): Canada. p. 113.
- [35] Xilinx, I., *Xilinx System Generator v7. 1 User Guide*.
- [36] Ifeachor, E. and B. Jervis, *Digital signal processing: a practical approach*. 2002: Pearson Education.
- [37] Inc, X., *Virtex-4 user guide*. UG070 (v2. 3), August, 2007.
- [38] Inc, X., *XtremeDSP for Virtex-4 FPGAs User Guide*. UG073 (v2. 1), December, 2005.