

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Saad DAHLAB de Blida
Institut d'aéronautique et des études spatiales

Projet de Fin d'Études

En vue d'obtention du Diplôme Master Académique en Aéronautique

Option: CNS/ATM

Communication Navigation Surveillance/Air Traffic Management

Thème :

Implémentation d'un modulateur ASK2 dans une carte FPGA

Réalisé par:

KENZA BELKAIS

Encadré et dirigé par:

Madame H. BOUGHRIRA

Promotion 2015/2016

Remerciement

En préambule à ce mémoire nous remerciant ALLAH qui nous aide et nous donne la patience et le courage durant ces longues années d'étude.

Nous souhaitant adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Ces remerciements vont tout d'abord au corps professoral et administratif de l'institut d'aéronautique, Juridiques et sociales, pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

Nous tenant à remercier sincèrement Madame H. BOUGHERIRA, qui, en tant que promotrice de mémoire, a toujours montrés à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'ils ont bien voulu nous consacrer et sans qui ce mémoire n'aurait jamais vu le jour.

On n'oublie pas nos parents pour leur contribution, leur soutien et leur patience.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragés au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Résumé :

La nouvelle génération de circuits logiques programmables permet l'intégration de systèmes complexes et hétérogènes. Mais, la structure figée de leur architecture modifie l'expertise de l'architecte de circuit. Le problème l'interaction entre algorithme et architecture se traduit différemment dans les domaines de ces circuits, l'objectif de ce travail est de présenter la méthode d'implémentation dans une carte FPGA en utilisant le logiciel Simulink/Matlab et Xilinx Design System, pour faire cette implémentation on a choisi le modulateur numérique d'amplitude OOK. La technologie des circuits reprogrammable FPGA est une alternative innovante par rapport de contrôle conventionnels (ASIC, DSP, microprocesseur ...) c'est pour cette raison, les FPGA font l'objet de recherches active.

تلخيص:

الجيل الجديد للدارات القابلة للبرمجة، يسمح لنا بدمج الأنظمة المعقدة و الغير متجانسة، لكن المهندس المتخصص في هذه الدارات يقوم بتغييرات على مستوى بنيتها الثابتة و تركيبها المعقد، و المشكل المتواجد في هذه الدارات هو انعكاس الخوارزمية على الهندسة الخاصة بها. و الهدف من عملنا هذا هو طريقة التطبيق شرائح مصفوفة البوابات المنطقية القابلة للبرمجة باستخدام برنامج مصفوفة وبرنامج توليد البنية المتكاملة من اجل هذا التطبيق اخترنا المغير الرقمي من نوع ايقاف تشغيل. هذه التكنولوجيا لشرائح مصفوفة البوابات المنطقية القابلة للبرمجة هي بديل المبتكر بالنسبة إلى معالج المعلومات و غيره من الشرائح و هذا ما جعلها نشطة في مجال البحث العلمي

Abstract:

The new generation of programmable logic allows permits of complex heterogeneous systems. But the fixed structure of their architecture changes the expertise of the circuit designer. The problem the interaction between algorithm and architecture is reflected differently in the fields of these circuits, the objective of this work is to present the implementation method in an FPGA using the software Simulink / Matlab and Xilinx ISE Design System for make this implementation we have chosen digital OOK modulator. The technology of reprogrammable FPGA circuits is an innovative alternative to conventional control (ASIC, DSP, microprocessor ...) for that reason, FPGAs are being actively searches.

Sigle et abréviation utilisé

FPGA: Feild programmable Gate Arrays

ASK: Amplitude shift Keying

FSK: Frequency shift keying

PSK: Phase Shift Keying

OOK: On Off Keying

TEB: Taux d'Erreur par Bit

MDA: Modulation par Déplacement d'Amplitude

MDP: Modulation par Déplacement de Phase

MAQ : Modulation d'Amplitude Quadrature

MDP : Modulation par Déplacement de Phase

VCO: voltage controlled oscillator

BPSK: Binary Phase Shift Keying

QPSK: Quadrature Phase Shift Keying

VHDL: Very high speed Hardware Description Langage

PLD: Programmable Logic Devices

EDA: Electronic Design Automatique

RAM: Random Access Memory

CLB: Configurable Logic Blocs

LUT: Look Up Table

HLS: High Level Synthesis

NI labVIEW: National Instrument Laboratory Virtual Instrument Engineering Workbench

ISE: Integrated Synthesis Environment

PAL: Programmable Array Logic

PROM: Programmable Read Only Mémory

GAL: Generic Array Logic

EPLD: Erasable Programmable Logic Device

CPLD : Complexe Programmable Logic Device

La liste de figure

Chapitre 1

Figure 1.1 : la chaine de transmission

Figure 1.2 : la forme générale du modulateur

Figure 1.3 : position d'un symbole dans le plan fresnel

Figure 1.4 : densité spectrale de puissance

Figure 1.5 : la forme polaire de la projection

Figure 1.6 : constellation

Figure 1.7 : modulation FSK et leurs spectre

Figure 1.8 : la bande passante dans le spectre

Figure 1.9 : modulateur FSK

Figure 1.10 : un oscillateur VCO (volage controlled oscillator)

Figure 1.11 : démodulateur FSK

Figure 1.12 : modulateur BPSQ

Figure 1.13 : démodulateur BPSK

Figure 1.14 : modulateur QPSK

Figure 1.15 : constellation

Figure 1.16 : modulateur ASK

Figure 1.17 : démodulateur ASK

Figure 1.18 : signal modulé par un modulateur ASK

Figure 1.19 : le modulateur OOK et le démodulateur

Chapitre 2

Figure 2.1 : les familles PLD

Figure 2.2 : les déférentes parties de FPGA

Figure 2.3 : bloc logique configurable

Figure 2.4: Look Up Table

Figure 2.5: flot de conception pour la génération de code VHDL par system generator

Figure 2.6 : les blocs Xilinx dans le system generator

Figure 2.7 : étapes de configuration de Matlab pour utiliser le system generator

Figure 2.8 : la fenêtre de la configuration matlab

Chapitre 3

Figure 3.1 : organigramme de notre projet

Figure 3.2 : modulateur OOK

Figure 3.3 : le programme CORDIC d'un signal cosinus dans le system generator

Figure 3.4 : étape de conversion en virgule fixe dans un modulateur OOK

Figure 3.5 : représentation virgule fixe binaire des nombres réels

Figure 3.6 : étape Matlabcoder

Figure 3.7 : le schéma bloc d'un modulateur

Figure 3.8 : les signaux dans le scope

Figure 3.9 : le bloc HDL coder du system generator

Figure 3.10 : le system generator

Figure 3.11 : les fichiers générer

Figure 3.12 : étape de simulation du code

Figure 3.13 : xilinx ISE system design

Figure 3.14 : le schéma RTL montre les entrées et les sorties

Figure 3.15 : le schéma RTL détaillé

Figure 3.16 : la première étape pour faire le test bench

Figure 3.17 : création d'une nouvelle source

Figure 3.18 : ossature du test bench VHDL

Figure 3.19 : les test bench avec stimuli prêt à la simulation

Figure 3.20 : les signaux de test bench

Figure 3.21 : la programmation des courbes

Figure 3.22 : les courbes (message, porteuse, le signal modulé)

Annexes :

Figure A.1 : la structure d'un PLD

Figure B.1 : cellule de base d'un PAL (architecture combinatoire)

Figure E.1 : architecture global d'un CPLD

Sommaire

Remerciement

Liste des abréviations

La liste de figure

Introduction générale.....1

Chapitre 1 : la modulation numérique :

1.1 Introduction2

1.2 La chaîne de transmission2

1.3 La modulation numérique3

1.3.1 L'analogique et le numérique.....3

1.3.2 Définition3

1.3.3 L'objectif de la modulation4

1.3.4 Les avantages de la modulation.....4

1.3.5 Les critères de choix d'une modulation5

1.3.6 Les applications des modulations numériques.....5

1.3.7 Principes de la modulation numérique.....5

1.3.8 Le concept de la voix I et la voix Q.....7

1.3.9 La constellation.....8

1.3.10 Le diagramme de l'œil10

1.4 Les types de la modulation numérique.....10

1.4.1 Modulation en fréquence FSK.....11

1.4.1.1 modulateur FSK.....12

1.4.1.2 Démodulateurs FSK.....13

1.4.2 Modulation en phase PSK.....14

1.4.2.1 La modulation BPSK (binary phase shift keying)14

1.4.2.1.1 Modulateur BPSK14

1.4.2.1.2 Démodulateur BPSK.....14

1.4.2.2 La modulation QPSK (Quadrature phase shift keying)15

1.4.2.2.1 Modulateur QPSK.....	15
1.4.2.2.2 Constellation.....	16
1.4.3 Modulation d'amplitude ASK.....	16
1.4.3.1 Modulateur ASK.....	17
1.4.3.2 Démodulateur ASK.....	17
1.4.4 Modulation OOK.....	17
1.4.4.1 modulateurs et le démodulateur OOK.....	18
1.5 Conclusion.....	18
 Chapitre 2: FPGA (feilds programmable gate arrays)	
2.1 Introductions.....	19
2.2 Définition des PLD.....	19
2.3 Les FPGAs (Feilds programmable Gate Arrays)	20
2.3.1 Définition des partie d'un FPGA.....	20
2.3.2 les cinqs principaux de la technologie FPGA.....	21
2.3.3 le choix de FPGA.....	21
2.4 les outils de conception des circuits FPGA(design tools)	21
2.4.1 System generator.....	22
2.4.2 Les étapes de la configuration.....	23
2.5 La conclusion.....	25
 Chapitre 3 : simulation et interprétation des résultats	
3.1 Introduction	26
3.2 Simulation d'un modulateur ASK2	27
3.2.1 Implémentation de la porteuse.....	27
3.2.2 Représentation des nombres en virgule fixe.....	28
3.2.3 Matlab coder : implémentation du modulateur sous matlab.....	29
3.2.4 HDL coder : génération du code VHDL du modulateur sous matlab.....	30
3.3 Simulation de modulateur ASK sous Xilinx ISE design system.....	32

3.3.1 Ouverture de Projet sous ISE.....	32
3.3.2 Création de test bench	34
3.3.4 Simulation et résultat.....	36
3.3.5 Validation des résultats.....	37
3.4 Conclusion.....	38
Conclusion Générale.....	39
Annexes	

Introduction

Le contrôle aérien fait partie de la navigation aérienne qui se présente comme le nerf le plus sensible du domaine aéronautique, cette dernière porte sur la sécurité des voyageurs ainsi que la sécurité des équipages (ou personnel navigant)

Le trafic aérien devient de plus en plus intense et crée de nombreux problèmes qui menacent la sécurité et le contrôle à la fois. A titre d'exemple l'encombrement et la saturation des systèmes de communication nécessite des équipements très complexes, en effet les protocoles de communication consiste à manipuler et traiter les informations transmises et reçus (codage, modulation, démodulation, décodage, affichage) selon des normes internationales standards préétablies. Ces différentes étapes de traitement de l'information (ou protocoles) sont programmées et exécutées par des processus puissants au niveau des tours de contrôle, et moins puissant au niveau des avions.

Le temps pris pour traitement, et donc de communication et alors trop long, et une perte d'information en de coule entraînant un risque de sécurité..... une des solutions pour pallier à ce problème de puissance (rapidité surtout) des processeurs, et d'implanter les protocoles de communication sur circuit et non de les programmer, un algorithme implanter en matériel étant des milliers, voir de millions de fois plus rapide que le même algorithme implémente en logiciel, c'est dans ce cadre que s'inscrit notre projet, qui consiste à implanter une des étapes du protocole de communication, la modulation sur circuit FPGA.

Un circuit FPGA et programmable peut être utilisé pour le prototypage, ou embarqué sur un aéronef.

Dans ce contexte, notre travail consiste à réaliser un modulateur de type ASK, le tester puis l'implémenter sur la carte FPGA. La simulation se fait à l'aide du system generator et ISE Design system.

Pour ce faire notre projet de fin d'étude sera organisé en 3 chapitre, le premier chapitre présente des généralités sur les différents types de modulations, le deuxième chapitre sera dédié à des notions sur la technologie FPGA ainsi que quelque circuit logique programmable. Puis un troisième chapitre qui décrit l'implémentation du modulateur ASK sous matlab et sa conversion en code VHDL en utilisant le « system generator » et sera consacré aux simulations et présentation des différentes étapes suivies pour la réalisation du modulateur ainsi que son implémentation sur FPGA (spartan de xilinx) suivies de discussion des résultats

Chapitre 1

La modulation numérique

Chapitre 1 la modulation numérique

1.1 Introduction

Dans un système de communication numérique, l'information se transporte d'une source vers un destinataire grâce à un support physique tel qu'un câble, une fibre optique ou encore la propagation d'onde. Les signaux transmis peuvent être analogiques ou numériques.

Donc le système de transmission doit acheminer le signal informatif vers le destinataire en conservant le maximum ou la totalité des données qu'on veut transmettre, c'est pour cette raison qu'on fait appel à la modulation.

Dans le cas de transmission numériques par ondes radio, on utilise une porteuse sinusoïdale en modifiant l'un ou plusieurs de ses paramètres amplitude, phase ou fréquence. On parle alors de modulations numériques avec porteuse. Puisque le but de notre projet est l'implémentation sur circuit FPGA de la modulation ASK2, nous positionnons, dans ce chapitre, la modulation dans la chaîne de transmission numérique (voir figure 1.1), puis nous présentons différents types, définitions, et principes respectifs.

1.2 La chaîne de transmission

La chaîne de transmission numérique véhicule de l'information entre une source et un destinataire en utilisant un support physique comme le câble, la fibre optique ou, encore la propagation sur un canal radioélectrique. Les signaux transportés peuvent être soit directement d'origine numérique comme dans les réseaux de données, soit d'origine analogique (parole, image...) mais convertis sous une forme numérique. La tâche du système de transmission est d'acheminer le signal de la source vers le destinataire avec le plus de fiabilité possible.

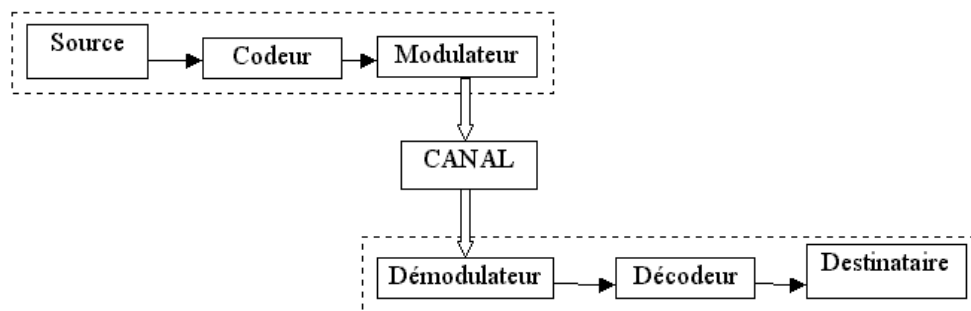


Figure 1.1 : la chaîne de transmission

1.3 La modulation numérique

La modulation numérique est une technique de facilitation de transfert des informations dans un milieu. La voix par exemple ne peut pas être envoyée à très grande distance pour cela on a besoin de la transmettre dans un autre milieu que l'air comme les lignes téléphoniques ou la radio. Le processus de conversion de l'information dans ce cas est la voix, qui peut être envoyée dans un milieu choisi avec succès est appelé Modulation.

1.3.1 L'analogique et le numérique

Nous débuterons notre discussion sur la modulation numérique par la mise en évidence de la différence entre l'analogique et le numérique :

L'information peut être représentée sous deux formes, analogique ou numérique. Le signal analogique est considéré continu, son amplitude peut prendre n'importe quelle valeur entre le maximum et le minimum du signal. Nous restons toujours avec la voix qui est un signal analogique et qui peut prendre aussi n'importe quel nombre des niveaux de volume dans ce qu'on appelle « la distance dynamique » qui est la distance du volume que peuvent produire nos cordes vocales. La voix analogique est convertie en signal numérique par le processus de l'échantillonnage dans un premier temps et la quantification dans un deuxième temps, le signal est quantifié en niveaux et chaque niveau est converti en nombre binaire.

1.3.2 Définitions:

- Symbole

Comme le bit est l'unité d'information, le symbole est l'unité de l'énergie de transmission, on peut dire aussi que c'est la représentation des bits qu'on veut transmettre pour acheminer l'information. C'est un élément d'un alphabet. Si M est la taille de l'alphabet, le symbole est alors dit M-aire. Lorsque M=2, le symbole est dit binaire. En groupant, sous forme d'un bloc, n symboles binaires indépendants, on obtient un alphabet de $M = 2^n$ symboles M-aires. Ainsi un symbole M-aire véhicule l'équivalent de n digits suivant la relation :

$$n = \log_2 M \text{ Bits} \dots \dots \dots (1.1)$$

- La rapidité de modulation R

C'est le nombre de changements d'états par seconde d'un ou de plusieurs paramètres modifiés simultanément. Un changement de phase de la porteuse, une excursion de fréquence ou une variation d'amplitude sont par définition des changements d'états.

$$R = 1/T \text{ s'exprime en "bauds"} \dots \dots \dots (1.2)$$

- Le débit binaire D

Il représente le nombre de bits transmis par seconde.

$$D = 1/T_b \text{ s'exprime en bits par seconde} \dots \dots \dots (1.3)$$

Dans le cas où $M=2$, on a une égalité entre le débit binaire et la rapidité de modulation.

La relation entre le débit et la rapidité est :

$$T = nT_b \text{ d'où : } D = nR \dots \dots \dots (1.4)$$

- Taux d'erreur par bit

Ce facteur caractérise la qualité d'une liaison.

$$TEB = \frac{\text{nombre de bits faux}}{\text{nombre de bits transmis}} \dots \dots \dots (1.5)$$

- L'efficacité spectrale de la modulation

Elle se définit par le paramètre $\eta = D/B$ et s'exprime en "bit/seconde/Hz".

B est la largeur de la bande occupée par le signal modulé. Pour un signal utilisant des symboles M-aires, on a :

$$\eta = \frac{\log_2 M}{T_s B} \dots \dots \dots (1.6)$$

L'efficacité spectrale augmente avec le nombre de bit/symbole.

1.3.3 L'objectif de la modulation numérique

L'intérêt de la modulation est l'adaptation du signal à transmettre aux conditions particulières du milieu de transmission et aussi l'utilisation simultanée de même canal de transmission par plusieurs communications c'est-à-dire le multiplexage.

1.3.4 Les avantages de la modulation

Les modulations numériques consistent à utiliser une porteuse sinusoïdale haute fréquence modulée par un signal informatif numérique. Les techniques de modulation diffèrent mais les

circuits modulateurs et démodulateurs sont identiques, dans leur principe, à ceux des modulations analogiques mais ils ont les avantages suivants :

1-Tout d'abord, le rapport signal sur bruit est meilleur avec un système numérique car, même si un signal numérique est bruité, distordu ou parasité, il est facile de le reconstruire en comparant ce signal déformé à un seuil ce qui définit une bonne qualité de transmission.

2-ensuite, les densités spectrales des signaux modulés numériquement ont des largeurs moindres qu'en analogique, ce qui permet d'augmenter le nombre de canaux utilisables par Hz pour les transmissions d'informations.

3- La sensibilité aux perturbations est moindre.

1.3.5 Les critères de choix d'une modulation

Le choix d'un tel type de modulation dépend des facteurs ci-dessous :

- Le débit binaire.
- Le taux d'erreur sur les bits TEB.
- L'efficacité d'occupation spectrale.
- La puissance émise.
- La simplicité de réalisation.

1.3.6 Les applications des modulations numériques :

- Les modems téléphoniques :
- Les faisceaux hertziens.
- Les transmissions par satellite.
- Les radiocommunications avec les mobiles.

1.3.7 Principes des modulations numériques

Le message à transmettre est issu d'une source binaire.

Le signal modulant, obtenu après codage, est un signal en bande de base, éventuellement complexe, qui s'écrit sous la forme :

$$c(t) = \sum_k c_k \cdot g(t - kT) \dots\dots\dots(1.7)$$

$$\text{avec } c_k(t) = a_k(t) + jb_k(t) \dots\dots\dots(1.8)$$

La fonction $g(t)$ est une forme d'onde qui est prise en considération dans l'intervalle $[0, T[$ puisque t doit vérifier la relation : $kT \leq t < (k+1)T$.

Dans les modulations MDA, MDP et MAQ (modulation d'amplitude quadrature), la modulation transforme ce signal $c(t)$ en un signal modulé $m(t)$ tel que :

$$m(t) = \text{Re} \left[\sum_k c_k(t) \cdot e^{j(\omega_0 t + \varphi_0)} \right] \dots\dots\dots(1.9)$$

La fréquence $f_0 = \frac{\omega_0}{2\pi}$ et la phase φ_0 caractérisent la sinusoïde porteuse utilisée pour la modulation.

Si les $c_k(t) = a_k(t) + jb_k(t)$ sont réels ($b_k(t) = 0$), la modulation est dite unidimensionnelle, et s'ils sont complexes la modulation est dite bidimensionnelle.

Le signal modulé s'écrit aussi plus simplement :

$$m(t) = \sum_k a_k(t) \cdot \cos(\omega_0 t + \varphi_0) - \sum_k b_k(t) \cdot \sin(\omega_0 t + \varphi_0) \dots \dots \dots (1.10)$$

Ou encore : $m(t) = a(t) \cdot \cos(\omega_0 t + \varphi_0) - b(t) \cdot \sin(\omega_0 t + \varphi_0) \dots \dots \dots (1.11)$

En posant : $a(t) = \sum_k a_k(t)$ et $b(t) = \sum_k b_k(t) \dots \dots \dots (1.12)$

Le signal $a(t) = \sum_k a_k(t)$ module en amplitude la porteuse en phase $\cos(\omega_0 t + \varphi_0)$ et le signal $b(t) = \sum_k b_k(t)$ module en amplitude la porteuse en quadrature $\sin(\omega_0 t + \varphi_0)$.

Dans la plupart des cas les signaux élémentaires $a_k(t)$ et $b_k(t)$ sont identiques à un coefficient près et ils utilisent la même forme d'impulsion $g(t)$ appelée aussi "formant".

$$a_k(t) = a_k \cdot g(t - kT) \text{ et } b_k(t) = b_k \cdot g(t - kT) \dots \dots \dots (1.13)$$

Les deux signaux $a(t)$ et $b(t)$ sont aussi appelés "trains modulants" et s'écrivent :

$$a(t) = \sum_k a_k \cdot g(t - kT) \text{ et } b(t) = \sum_k b_k \cdot g(t - kT) \dots \dots \dots (1.14)$$

Les symboles a_k et b_k prennent respectivement leurs valeurs dans l'alphabet (A_1, A_2, \dots, A_M) et dans l'alphabet (B_1, B_2, \dots, B_M) .

Le schéma théorique du modulateur est représenté figure 1.2

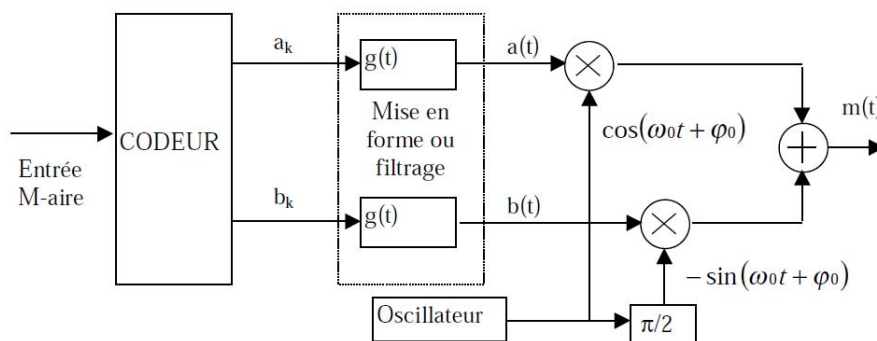


Figure 1.2 : forme générale du modulateur

Les différents types de modulations sont définis par les alphabets décrits ci-dessous et par la fonction $g(t)$.

A chaque symbole émis correspond un signal élémentaire de la forme (equ. 1.15):

$$m_k(t) = a_k \cdot g(t - kT) \cdot \cos(\omega_0 t + \varphi_0) - b_k \cdot g(t - kT) \cdot \sin(\omega_0 t + \varphi_0) \dots \dots \dots (1.15)$$

qui peut être représenté dans un espace à deux dimensions dont les vecteurs de base sont :

$g(t - kT) \cdot \cos(\omega_0 t + \varphi_0)$ et $-g(t - kT) \cdot \sin(\omega_0 t + \varphi_0)$ c'est la décomposition de Fresnel comme est montré dans la figure (X)

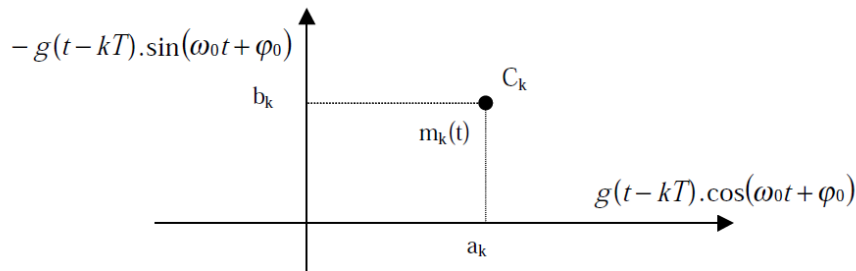


Figure 1.3 : position d'un symbole dans le plan de Fresnel

Le signal modulé $m(t)$ véhicule des informations distinctes à travers $a_k(t)$ et $b_k(t)$ qui sont deux signaux en bande de base appelés respectivement *composante en phase* (I en anglais) et *composante en quadrature* (Q en anglais). La récupération de $a_k(t)$ et $b_k(t)$ sera possible uniquement si ces deux signaux sont de bande limitée à l'intervalle $[-B, B]$ avec $B < f_0$ (Condition de Rayleigh).

Concernant la Densité Spectrale de Puissance (DSP) du signal modulé $m(t)$, nous rappelons que si $\alpha_m(t) = x_c(t) + jx_s(t)$ représente le signal en bande de base de $m(t) = \text{Re}[\alpha_m(t) \cdot e^{j(\omega_0 t + \varphi_0)}]$

et si $\gamma_{\alpha_m}(f)$ est la densité spectrale de puissance de $\alpha_m(t)$, alors la DSP du signal modulé $m(t)$ sera : $\gamma_m(f) = \frac{1}{4} [\gamma_{\alpha_m}(f - f_0) + \gamma_{\alpha_m}(-f - f_0)]$

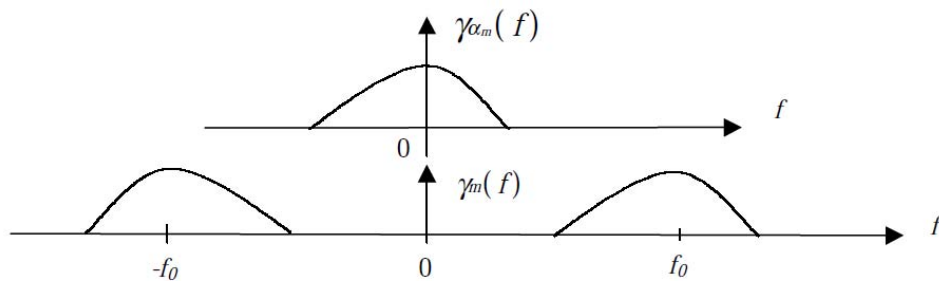


Figure 1.4 : densité spectrale de puissance.

Lorsque la modulation est linéaire, l'opération de modulation a pour effet, dans la plupart des cas, de translater la densité spectrale de puissance (DSP) du signal modulant.

La DSP du signal modulé $m(t)$ est aussi liée à la forme de l'onde $g(t)$, (qui sera souvent rectangulaire) par sa transformée de Fourier $G(f)$.

1.3.8 Le concept de la voie I et la voie Q :

On peut écrire des signaux réels comme une somme de deux vecteurs de deux signaux en quadrature appelé I et Q, on peut aussi les assimiler à une projection du signal sur le plan (x, y) .

$S = [I \ Q]$.

Dans la figure 1.5 qui suit nous voyons deux représentations d'un signal, l'une nous montre la projection I et Q et l'autre nous montre la forme polaire

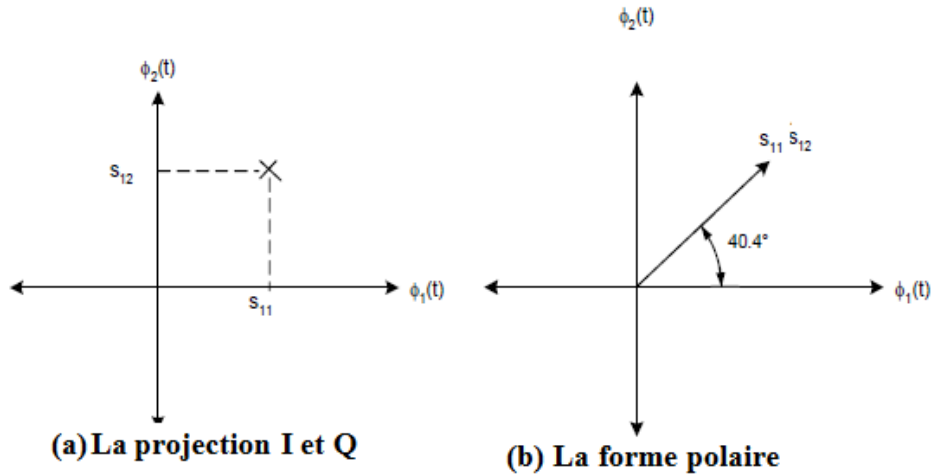


Figure 1.5 : la forme polaire et la projection

Dans la figure (1.5.a), les axes x et y sont la projection I et Q du signal respectivement.

S^{11} est la projection du canal I et S^{12} est la projection du canal Q.

La figure (1.5.b) montre le même signal mais en forme polaire avec :

Sa longueur égale à son amplitude et son angle égale à sa phase.

Quand on joint les coefficients S^{11} et S^{12} dans le plan (x,y), ces deux amplitudes nous donnent un vecteur. L'angle formé entre ce vecteur et l'axe des x est la phase du signal.

L'amplitude du signal : $S = \sqrt{I^2 + Q^2}$ (1.16)

La phase du signal : $\varphi = \tan^{-1} \left(\frac{I}{Q} \right)$ (1.17)

1.3.9 La constellation

On appelle une constellation un graphique qui nous montre l'amplitude et la phase c'est la représentation dans le plan complexe des points associés aux symboles, ces points nous permettent de différencier chaque type de modulation.

On obtient le diagramme de constellation par l'échantillonnage des deux voies I et Q au même instant, après nous implantons la valeur de I et Q dans le plan (x,y) (fig.1.6).

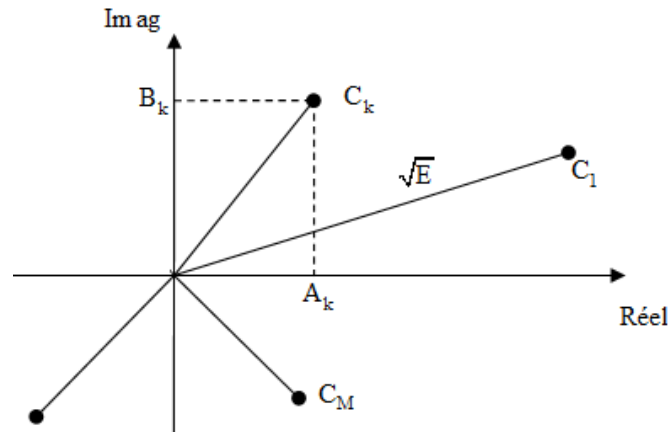


Figure1.6 : constellation

- La longueur du vecteur représente l'énergie de ce signal.
- Les points qui sont loin de l'origine ont une énergie plus grande que ceux qui sont près.
- L'angle entre les deux vecteurs est le produit de corrélation des deux signaux.

-La répartition des points dans le plan dépend de :

- **La distance minimale D_{min} entre deux symboles:** plus cette distance est grande plus la probabilité d'erreur sera faible.

La distance minimale entre tous les symboles est :

$$D_{min} = \text{Max}_{i \neq j} (D_{ij}) \text{ avec : } D_{ij} = |C_i - C_j|^2$$

- **La puissance d'émission des symboles :** à chaque symbole émis correspond un signal élémentaire $m(t)$ et une énergie nécessaire à la transmission de ce symbole.

L'énergie de ce signal est proportionnelle à la distance entre le point et l'origine.

La puissance moyenne d'émission des symboles est assimilable à $\sum_i |C_i|^2$ et la puissance crête à $\text{Max}_i |C_i|^2$.

Ces deux critères sont antagonistes puisque l'on serait tenté d'une part d'éloigner les symboles au maximum pour diminuer la probabilité d'erreur et d'autre part, de les rapprocher de l'origine pour minimiser l'énergie nécessaire à la transmission.

1.3.10 Le diagramme de l'œil

Le diagramme de l'œil est très connu dans la transmission numérique, c'est un oscillogramme qui représente les données numériques échantillonnées de manière répétitives et sont appliquées à l'entrée de déviation verticale, alors que la déviation horizontale, qu'on appelle le déclenchement, est synchronisée avec le débit du signal. Le nom de ce diagramme vient du fait que pour nombre de codage, le motif obtenu ressemble à une suite d'yeux encadrés par deux rails horizontaux.

De nombreux critères de performance peuvent être déduits de cette analyse. Si les signaux sont trop longs, trop courts, mal synchronisés par rapport à l'horloge du système, de niveau trop important ou trop faible, trop entachés de bruit, trop lents lors des changements d'état, ou comportant trop de dépassements ou d'inertie, le diagramme de l'œil les mettra en évidence. Un œil ouvert correspondra à un signal comportant un minimum de distorsion ce qui nous informe que le signal est de bonne qualité. La distorsion de la forme d'onde du signal, pouvant être attribuée à de l'interférence inter-symbole ou à du bruit, se traduit par une fermeture de l'œil (dégradation du signal).

Le tableau suivant nous interprète quelques mesures :

Caractéristique de l'œil	Phénomène mesuré
Ouverture de l'œil (amplitude, valeur crête à crête)	mesure le bruit ajouté au signal
Dépassement de la valeur nominale, inertie du signal	mesure la distorsion subie par les fronts du signal
Largeur de l'œil	mesure l'efficacité de la synchronisation et l'effet de la gigue (décalage).

1.4 Les types de la modulation numériques

Vu que le domaine de transmission évolue rapidement ces dernier temps, plusieurs techniques de modulation sont apparues :

*Modulation par Déplacement d'Amplitude MDA (Amplitude Shift Keying ASK).

*Modulation par Déplacement de Fréquence MDF (Frequency Shift Keying FSK).

* Modulation par Déplacement de Phase MDP (Phase Shift Keying PSK).

Ces trois types sont la base de la modulation numérique. Pour chaque type on a recourt à la variation d'un paramètre du signal sinusoïdal qui représente l'information qu'on veut envoyer. Les trois paramètres du signal sinusoïdal qui peuvent être changés sont soit l'amplitude, soit la fréquence soit la phase.

Dans les procédés de modulation **binaire**, l'information est transmise à l'aide d'un paramètre qui ne prend que deux valeurs possibles.

Dans les procédés de modulation **M-aire**, l'information est transmise à l'aide d'un paramètre qui prend M valeurs. Ceci permet d'associer à un état de modulation un mot de n digits binaires. Le nombre d'états est donc $M=2^n$. Ces n digits proviennent du découpage en paquets de n digits du train binaire.

1.4.1 Modulation en fréquence (FSK: Frequency Shift Keying)

Lorsque la valeur binaire est 0, on envoie une fréquence f_1 et lorsque la valeur binaire est 1, on envoie une autre fréquence f_2 .

T est la durée d'un symbole. Il n'y a que 2 symboles possible, soit une fréquence f_1 , soit une fréquence f_2 .

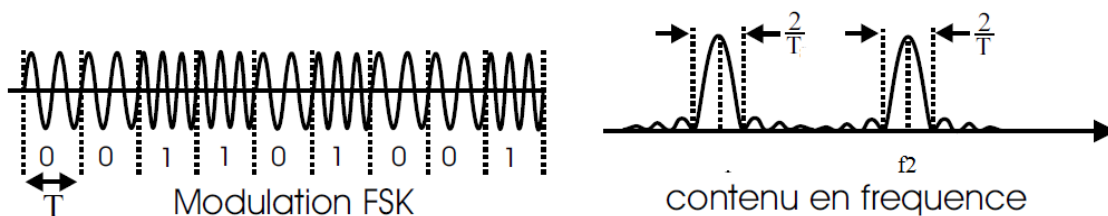


Figure 1.7 : modulation FSK et leur spectre

Le spectre en fréquence d'un signal FSK est constitué de 2 raies, une à la fréquence f_1 et l'autre à la fréquence f_2 . La largeur des raies dépend de la durée des symboles

La bande passante minimum théorique de la ligne de transmission pour passer correctement un signal FSK est de (f_2-f_1+2/T) . pour une application réelle, il faut garder une bande passante un peu plus grande pour tenir compte du fait que les filtres ne sont pas parfaits.

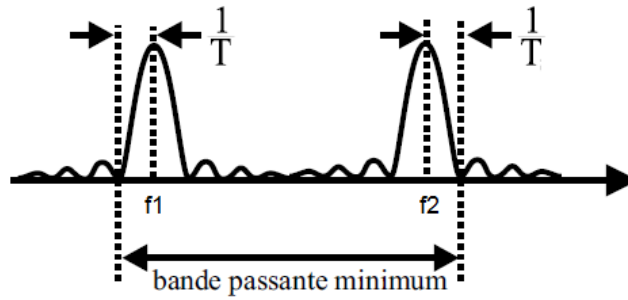


Figure 1.8 : la bande passante dans le spectre

1.4.1.1 Modulateur FSK

On peut utiliser 2 porteuses et un commutateur pour sélectionner la fréquence de la porteuse selon la valeur du binaire. Cette méthode est simple mais il y a des discontinuités de la porteuse au moment du changement de fréquence.

Le modulateur MSK (Minimum Shift Keying) élimine cette discontinuité en choisissant des fréquences de porteuse qui sont des multiples impairs de la demi fréquence binaire (f_1 et $f_2 = n \cdot (f_b/2)$ où n est un entier impair et f_b est la fréquence du binaire en bit par seconde).

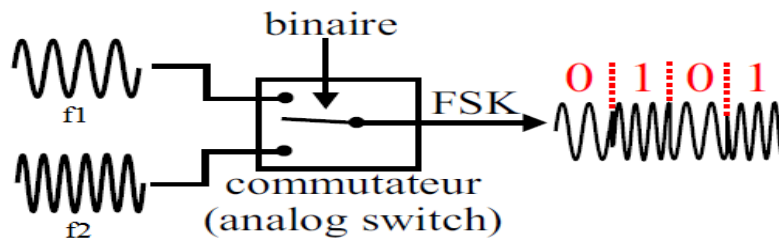


Figure 1.9 : modulateur FSK

Une autre façon de moduler en FSK est d'utiliser la tension du signal binaire pour changer la fréquence d'un oscillateur VCO (Voltage Controlled Oscillator). La transition est progressive et se fait en douceur.

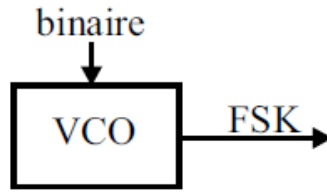


Figure1.10 :un oscillateur VCO(voltage controlled oscillator)

1.4.1.2 Démodulateur FSK

On peut utiliser 2 filtres comme le montre la figure1.11 ci après.

La tension crête de la sortie des deux filtres est entrée dans le comparateur qui décide laquelle des deux fréquences est la plus importante. Si l'amplitude de la fréquence 2 est la plus grande, alors c'est un 1.

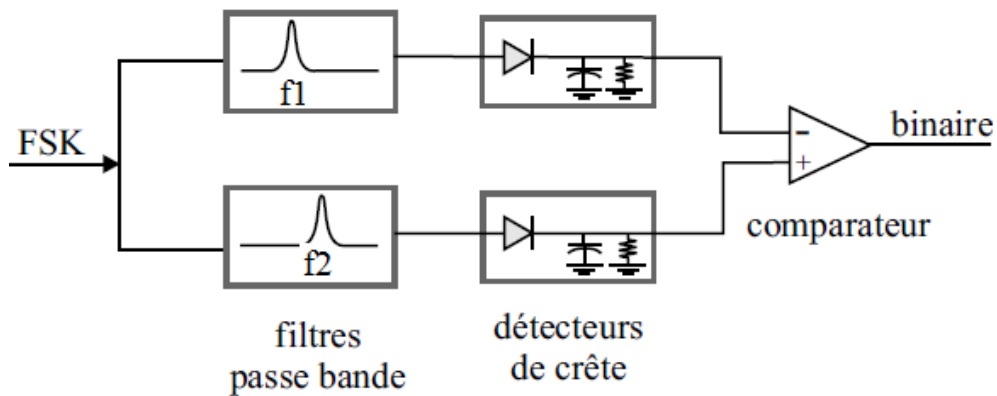


Figure1.11 : démodulateur FSK

Une autre façon de démoduler un signal FSK est d'utiliser la PLL (Phase LockLoop). Un excréteur élimine les trop hautes tensions qui pourraient endommager le circuit. Le comparateur de phase génère une tension d'erreur qui est proportionnelle à la différence de fréquences entre ces deux entrées. Cette tension d'erreur contrôle la fréquence d'un VCO. Si la fréquence du VCO est inférieure à celle du signal FSK, alors le comparateur de phase génère une tension qui fait augmenter la fréquence du VCO. Si la fréquence du VCO est supérieure à celle du signal FSK, alors le comparateur de phase génère une tension qui fait diminuer la fréquence du VCO. Il s'en suit que la tension d'erreur s'ajuste pour que la fréquence du VCO devienne égale à celle du signal FSK. Lorsque la fréquence du FSK change (en fonction du binaire), la tension d'erreur change aussi (il y a une tension qui permet

de reproduire la fréquence f_1 et une autre tension qui permet au VCO de générer la fréquence f_2) reproduisant ainsi le binaire. Enfin un Schmitt trigger change les niveaux de tension pour les rendre conformes.

La modulation FSK est simple et résiste très bien au bruit mais demande une grande bande passante car il faut au moins passer les deux fréquences. Dans le but d'augmenter l'efficacité spectrale, on utilise d'autres types de modulations qui n'utilisent qu'une seule fréquence.

1.4.2 Modulation en phase PSK (Phase Shift Keying)

Il s'agit d'une modulation par saut de phase. Ici on parle de deux types qui sont :

1.4.2.1 La BPSK (Binary Phase Shift Keying):

Dans ce type de modulation, la phase de la porteuse, de fréquence et amplitude constantes, saute brusquement d'une phase ϕ_1 à une autre phase ϕ_2 donc il n'y a qu'une seule fréquence de porteuse. Lorsque le binaire est 1, on envoie la porteuse et lorsque le binaire est 0, on envoie l'inverse de la porteuse, soit un déphasage de 180° .

1.4.2.1.1 Modulateur BPSK

Lorsque le niveau binaire est 1, on multiplie la porteuse par $+V$, ce qui l'amplifie mais ne l'inverse pas. Lorsque le niveau binaire est 0, on multiplie la porteuse par $-V$, ce qui l'inverse, créant un déphasage de 180° .

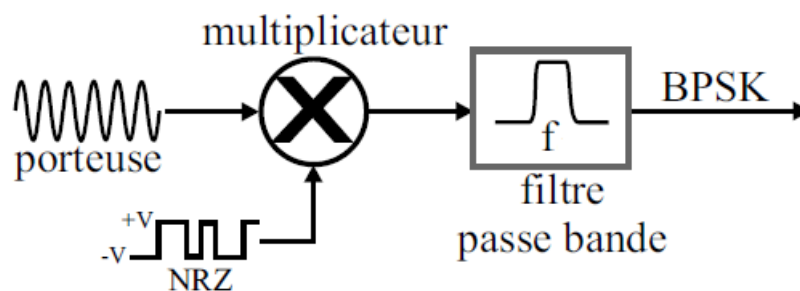


Figure 1.12 : modulateur BPSK

1.4.2.1.2 Démodulateur BPSK :

Le signal BPSK reçu est soit $+A \cos(2\pi f_0 t)$ lorsque le binaire est au niveau logique 1, soit $-A \cos(2\pi f_0 t)$ lorsque le niveau binaire est 0.

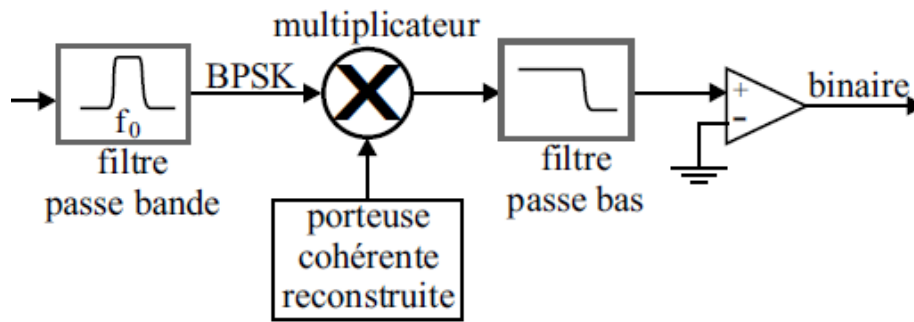


Figure 1.13:démodulateur BPSK

1.4.2.1 Modulation QPSK (Quadrature Phase Shift Keying)

La quadrature consiste à moduler indépendamment deux porteuses déphasées de 90° et la somme de ces deux porteuses donne une onde sinusoïdale de même fréquence que les porteuses mais déphasée. La porteuse de référence est appelée I pour "In phase" et celle qui est déphasée de 90° , est appelée Q pour "In Quadrature".

En QPSK, on code indépendamment chacune des porteuses en BPSK. Les deux signaux BPSK sont additionnés pour former le signal QPSK. À la réception, on peut séparer ce qui a été codé sur I de ce qui a été codé sur Q. Pour chaque symbole de la porteuse, nous récupérons 2 bits, un sur I et l'autre sur Q. puisqu'il y a 2 bits par symbole, le débit binaire est le double de celui des symboles par seconde.

1.4.2.2.1 Modulateur QPSK :

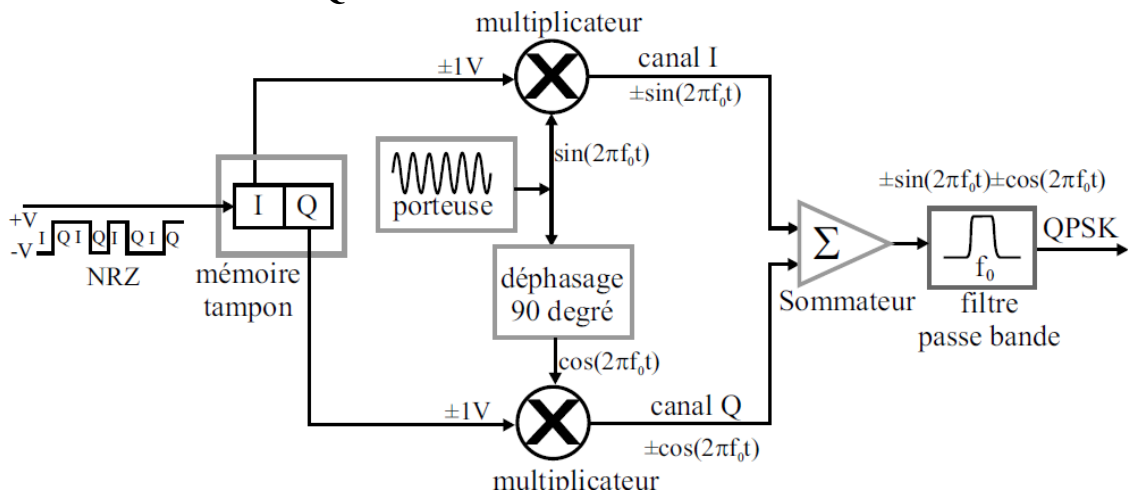


Figure 1.14 : modulateur QPSK

Les bits arrivent en série dans la mémoire pour ressortir en parallèle deux par deux. Le bit I module en BPSK une porteuse en phase tandis que le bit Q module en BPSK une porteuse déphasée de 90°. Les deux modulations se font simultanément. Les signaux BPSK sont additionnés pour donner : $\pm \sin(2\pi f_0 t) \pm \cos(2\pi f_0 t)$. Il y a quatre combinaisons possibles, soit :

- 00, signifiant : 0 sur le canal I et 0 sur le canal Q $\{-\sin(2\pi f_0 t) - \cos(2\pi f_0 t)\}$.
- 01, signifiant : 0 sur le canal I et 1 sur le canal Q $\{-\sin(2\pi f_0 t) + \cos(2\pi f_0 t)\}$.
- 10, signifiant : 1 sur le canal I et 0 sur le canal Q $\{+\sin(2\pi f_0 t) - \cos(2\pi f_0 t)\}$.
- 11, signifiant : 1 sur le canal I et 1 sur le canal Q $\{+\sin(2\pi f_0 t) + \cos(2\pi f_0 t)\}$.

1.4.2.2 Constellation :

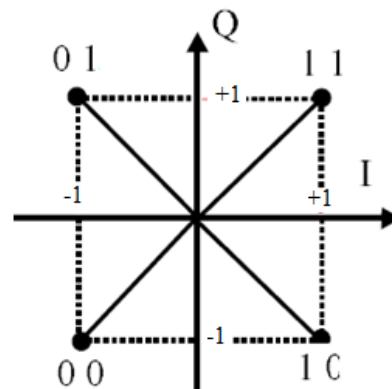


Figure 1.15 : constellation

Cette constellation nous montre l'amplitude et la phase de chacune des 4 possibilités. Les amplitudes ± 1 des sinus (codés sur le canal I) sont représentées sur l'axe I à déphasage nul. Les amplitudes ± 1 des cosinus (codés sur le canal Q) sont représentées sur l'axe I à déphasage de 90°. Chaque point représente la somme des deux canaux. On constate que l'amplitude est la même pour tous les points et que c'est le déphasage qui les différencie. Les déphasages sont :

- 00 déphasage de 225°.
- 01 déphasage de 135°.
- 10 déphasage de 315°.
- 11 déphasage de 45°.

1.4.3 La modulation d'amplitude ASK (Amplitude Shift Keying)

En modulation ASK, l'amplitude du signal est le seul paramètre qui change, les autres sont fixes. Le bit "1" est transmis par un signal avec une amplitude particulière. Pour transmettre le "0", nous devons changer l'amplitude tout en conservant la fréquence constante.

Ce type de modulation est généralement utilisé en (wire-based radio) avec ou sans porteuse.

On-Off Keying (OOK) est un cas particulier de la modulation d'amplitude ASK, c'est un cas où l'une des deux amplitudes est nulle et c'est la technique que nous avons choisie pour notre étude.

1.4.3.1 Modulateur ASK

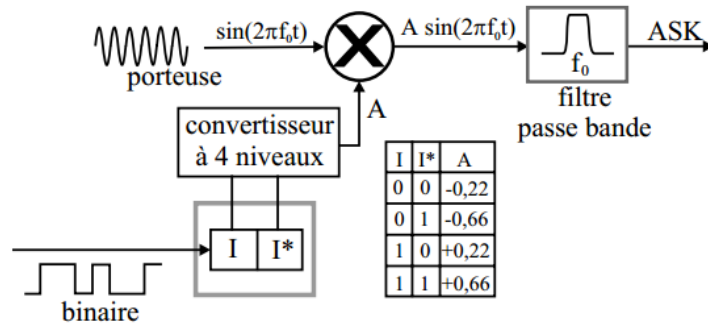


Figure 1.16 : modulateur ASK

1.4.3.2 Démodulateur ASK

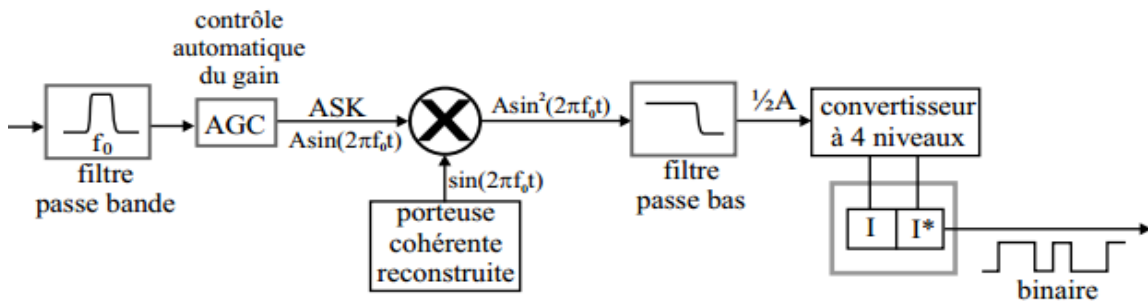


Figure 1.17 : démodulateur ASK

1.4.4 Modulation par tout ou rien (OOK : On Off Keying)

C'est la modulation la plus simple, elle consiste à moduler directement la porteuse par le signal binaire 0 ou 1 (0 est l'absence de porteuse et 1 la présence de la porteuse).

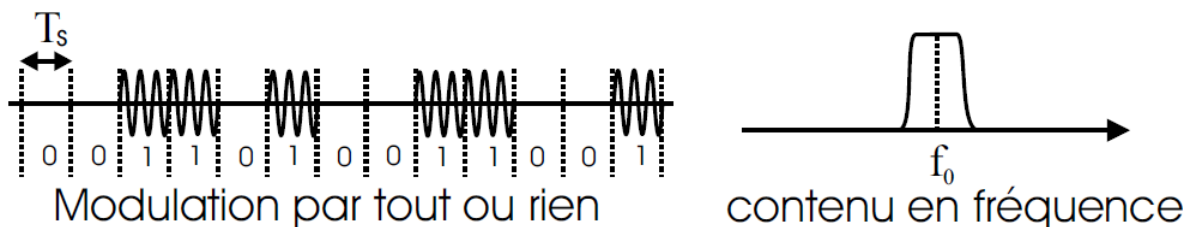


Figure 1.18 : signal modulé par un modulateur OOK

Le spectre (ou contenu en fréquence) donne une seule raie à la fréquence de la porteuse (f_0). La largeur de raie dépend de la durée des symboles. Comme il n'y a qu'une seule raie à passer, la bande passante minimum requise est plus petite.

1.4.4.1 Modulateur et démodulateur OOK

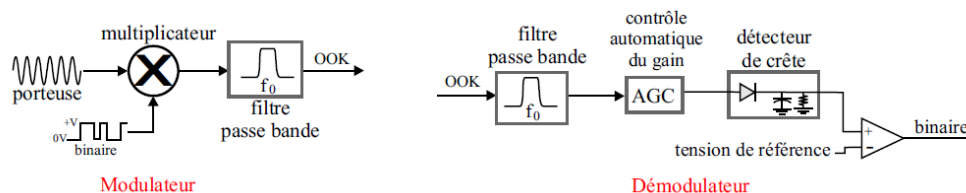


Figure 1.19 : le modulateur OOK et le démodulateur

1.5 Conclusion

La diversité des applications que nous venons d'exposer met en évidence l'importance des différentes techniques de modulation dans la transmission numérique sur porteuse.

Un intérêt majeur des transmissions numériques réside dans la possibilité de leur insertion harmonieuse dans les réseaux intégrés numériques qui se développent de jour en jour. Un autre avantage réside dans la possibilité de conserver l'intégrité de l'information à transmettre, ce qui est tout à fait impossible avec une transmission analogique. Cependant, la simplicité d'utilisation des modulations analogiques traditionnelles fait qu'elles ne sont pas encore reléguées au musée des techniques désuètes.

Les systèmes modernes de communication numérique sont complexes et requièrent des circuits de modulation et de démodulation de plus en plus sophistiqués. Nous avons examiné les trois types de base de modulations qui sont aujourd'hui utilisées. Il s'avère que le choix d'un type de modulation est toujours déterminé par les contraintes de l'application. Dans notre cas nous avons choisi d'implémenter la modulation OOK sur circuit FPGA. Nous consacrons le chapitre suivant à une description succincte des circuits FPGA, et des outils de conception que nous utiliserons pour l'implémentation matérielle de notre application.

Chapitre 2

FPGA (Feild Programmable Gate Array)

Chapitre 2 : FPGA (Feild Programmable Gate Array)

2.1. Introduction

Le but de notre projet étant l'implantation de la modulation ASK de type OOK sur circuit FPGA, nous présentons dans ce chapitre, les circuits programmable FPGA (appartenant à la famille des circuits logiques programmables PLD); nous introduisons également la méthodologie utilisée pour la création du code VHDL (Very high speed Hardware Description Langage) d'un algorithme mathématique complexe,utilisant l'outil de conception assistée par ordinateur (CAO) et EDA(Electronic Design Automation), le System Generator de MATLAB.

2.2. Définition des PLDs

Un circuit programmable est un assemblage d'opérateurs combinatoires (les opérateurs combinatoires génériques qui interviennent dans les circuits programmables proviennent soit des mémoires (réseaux logiques) soit des fonctions standard (multiplexeurs et XOR)) et de bascules dans lequel la fonction réalisée n'est pas fixée lors de la fabrication. Il contient potentiellement la possibilité de réaliser toute une classe de fonctions, plus ou moins large suivant son architecture. La programmation du circuit consiste à définir une fonction parmi toutes celles qui sont potentiellement réalisables. Comme dans toute réalisation en logique câblée, une fonction logique est définie par les interconnexions entre des opérateurs combinatoires et des bascules et par les équations des opérateurs combinatoires. Ce qui est programmable dans un circuit concerne donc les interconnexions et les opérateurs combinatoires.

Les circuits FPGA que nous utilisons dans notre projet appartiennent à la famille des PLD (voir annexe)

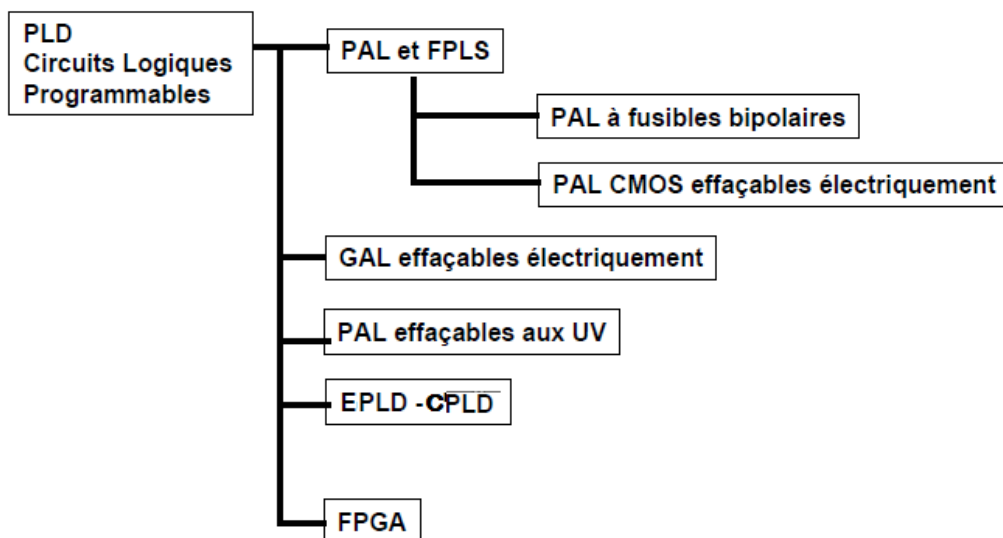


Figure 2.1 : les familles PLD

2.3. Les circuits FPGA:

Au niveau le plus élevé, un FPGA est un circuit en silicium reprogrammable. À l'aide de blocs logiques préconstruits et de ressources de routage programmables, vous pouvez configurer ce circuit afin de mettre en œuvre des fonctionnalités matérielles personnalisées, sans avoir jamais besoin d'utiliser une maquette ou un fer à souder. Il vous suffit de développer des tâches de traitement numérique par logiciel et de les compiler sous forme de fichier de configuration ou de flux de bits contenant des informations sur la manière dont les composants doivent être reliés. En outre, les FPGA sont totalement reconfigurables et peuvent adopter instantanément une nouvelle « personnalité » si vous recompilez une nouvelle configuration de circuits. Jusqu'à présent, seuls des ingénieurs particulièrement expérimentés en matière de conception de matériel numérique pouvaient utiliser la technologie FPGA. Toutefois, la généralisation des outils, de conception de haut niveau et d'automatisation de la conception, est en train de modifier les règles de la programmation de FPGA, grâce à de nouvelles technologies permettant de convertir des diagrammes graphiques ou même du code C ANSI en circuits matériels numériques.

2.3.1 Définition des parties d'un FPGA

Chaque puce FPGA est constituée d'un nombre fini de ressources prédéfinies avec des connexions programmables pour mettre en œuvre un circuit numérique reconfigurable et des blocs d'E / S pour permettre au circuit d'accéder au monde extérieur.

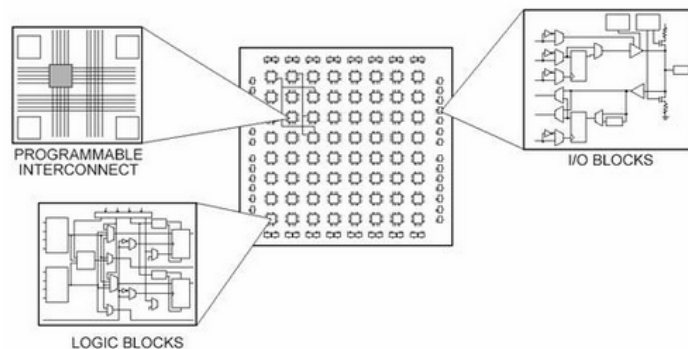


Figure 2.2 : Les différentes parties de FPGA

Spécifications de ressources FPGA incluent souvent le nombre de blocs logiques configurables, le nombre de fonctions fixes blocs logiques tels que des multiplicateurs, et la taille des ressources de mémoire comme bloc intégré RAM. Parmi les nombreuses parties de la puce FPGA, ce sont généralement les plus importants lors de la sélection et la comparaison des FPGA pour une application particulière.

Les blocs logiques configurables (CLB) est l'unité logique de base d'un FPGA. Parfois appelé tranches ou cellules logiques, CLB sont constitués de deux éléments de base: flip-flopset Look UP Tables (LUT) voir la figure 2.12. Plusieurs familles de FPGA diffèrent dans la manière des bascules et LUT sont emballés ensemble, il est donc important de comprendre bascules et LUT.

La fonction de la LUT est de stocker la table de vérité de la fonction combinatoire à implémenter dans la cellule (voir la figure 2.3)

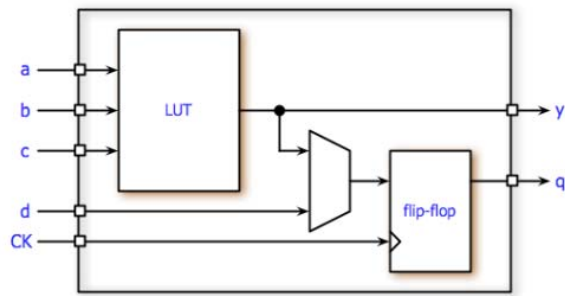


Figure 2.3 : bloc logique configurable

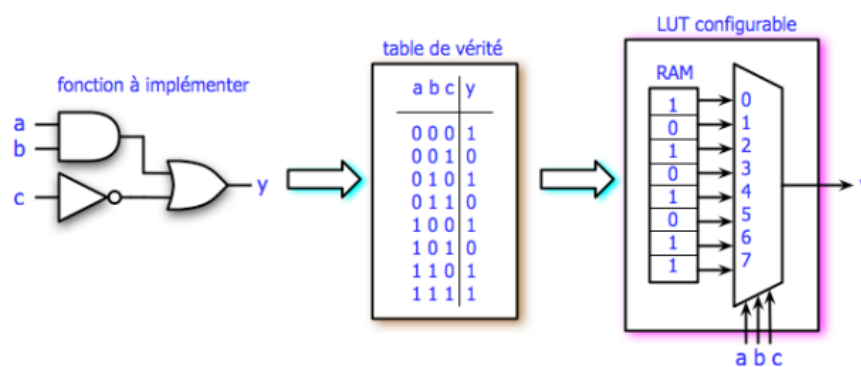


Figure 2.4 : Look Up Table

2.3.2 Les cinq principaux avantages de la technologie FPGA

1. Performances
2. Temps de mise sur le marché
3. Coût
4. Fiabilité
5. Maintenance à long terme

2.3.3 Le choix du FPGA

Lorsque vous étudiez les spécifications d'un circuit FPGA, souvenez-vous qu'ils sont souvent constitués de différents éléments : des blocs logiques configurables, comme des slices (tranches) ou des cellules logiques ; de la logique fixe, comme des multiplicateurs ; des ressources de mémoire, comme des blocs de RAM intégrés. Bien que les FPGA contiennent bien d'autres composants, ceux mentionnés sont généralement les plus déterminants lorsqu'il s'agit de choisir et de comparer des FPGA pour une application particulière.

2.4 Outils de conception des circuits FPGA (Design Tools)

Maintenant que vous connaissez les blocs de construction d'une puce FPGA, vous pouvez demander, "Comment configurer tous ces millions de composants pour construire la logique que vous devez exécuter?"

La réponse est que vous définissez les tâches de calcul numérique en logiciel en utilisant des outils de développement, puis compilez-les dans un fichier de configuration ou bitstream qui contient des informations sur la façon dont les composants doivent être reliés ensemble. Le défi dans le passé avec la technologie FPGA est que les outils de conception FPGA bas niveau ne peuvent être utilisés que par des ingénieurs avec une compréhension profonde de la conception de matériel numérique. Toutefois, l'augmentation de la synthèse de haut niveau (HLS) des outils de conception, tels que les logiciels de conception de systèmes NI LabVIEW, ou les toolbox de matlab comme system generator(section 2.4.1) qui modifient les règles de la programmation FPGA et fournissent de nouvelles technologies qui convertissent des diagrammes graphiques dans les circuits de matériel numérique ou des algorithmes soft en circuits numériques....

2.4.1 Matlab System generator

L'utilisation de system generator de Matlab permet de générer automatiquement le VHDL pour notre modulateur ASK2. La figure2.4 montre le flot de conception de la génération d'un code HDL pour implémentation sur curcuit FPGA

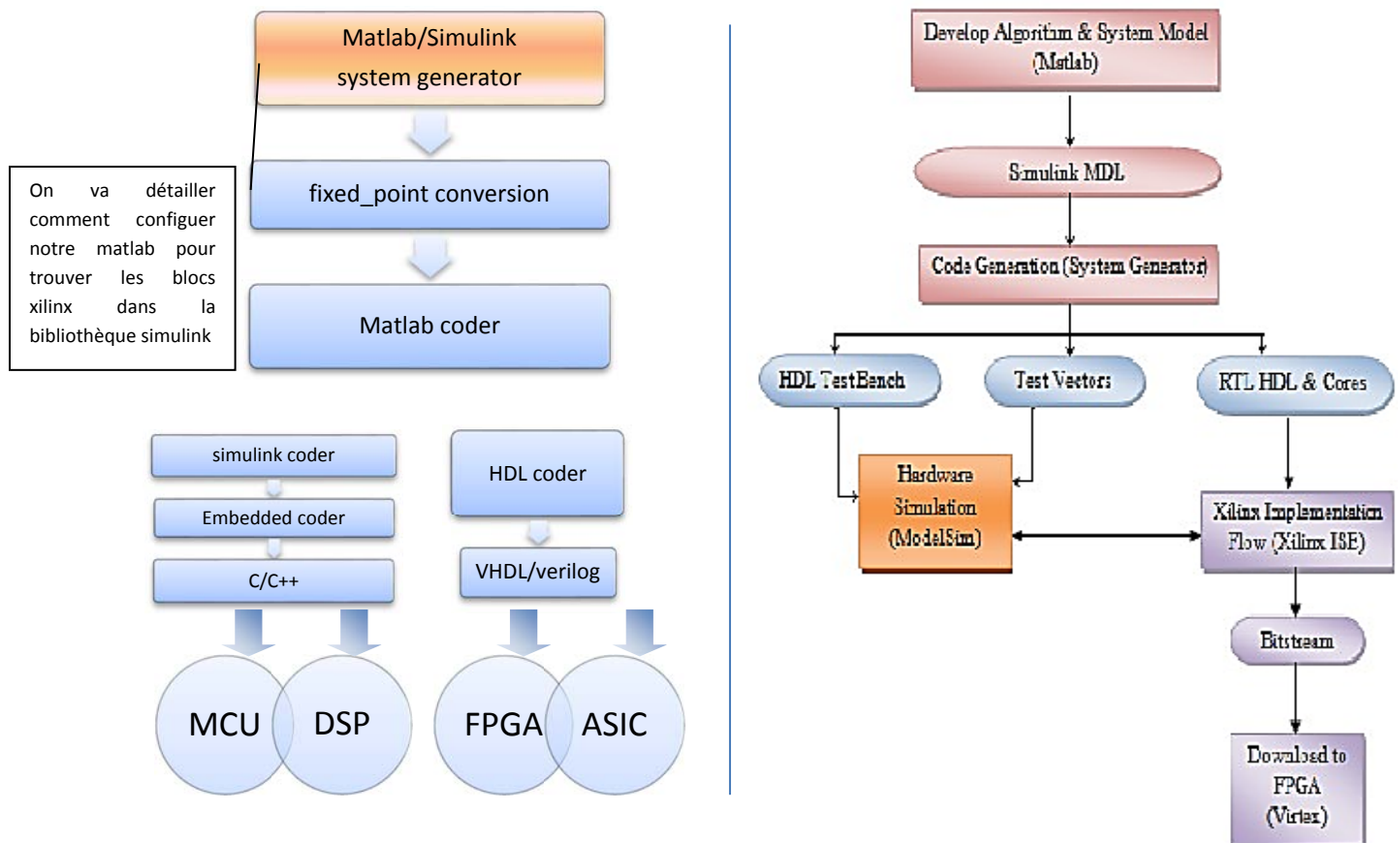


Figure 2.5 : Flot de conception pour la génération du code VHDL par System Generator
 a) flot global ; b) les étapes de conception

2.4.2 les étapes de configuration

Pour utiliser system generator il faut installer le logiciel matlab et ISE design system compatible puis les configurer en suivant les étapes de configuration montrées ci-dessous.

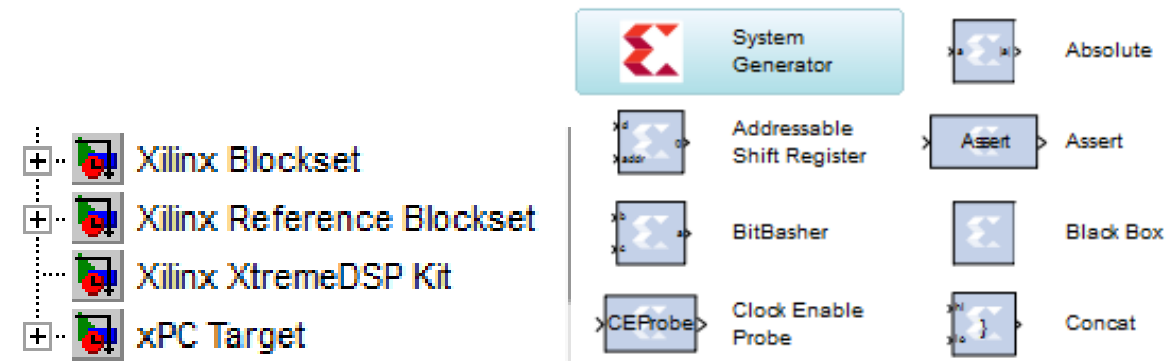


Figure 2.6 : les blocs xilinx dans system generator

Pour trouver ces blocs on va configurer notre matlab à l'aide de ISE Design Suite (figure 2.7) retraçant les étapes suivantes :

- ouvrir le menu démarrer
- choisir tous les programmes
- Xilinx Design Tools
- clic sur ISE design suite
- system generator
- choisir "system generatorMatlabconfigurator "
- on va sélectionner la version de Matlab choisie
- appuyer sur le bouton APPLY

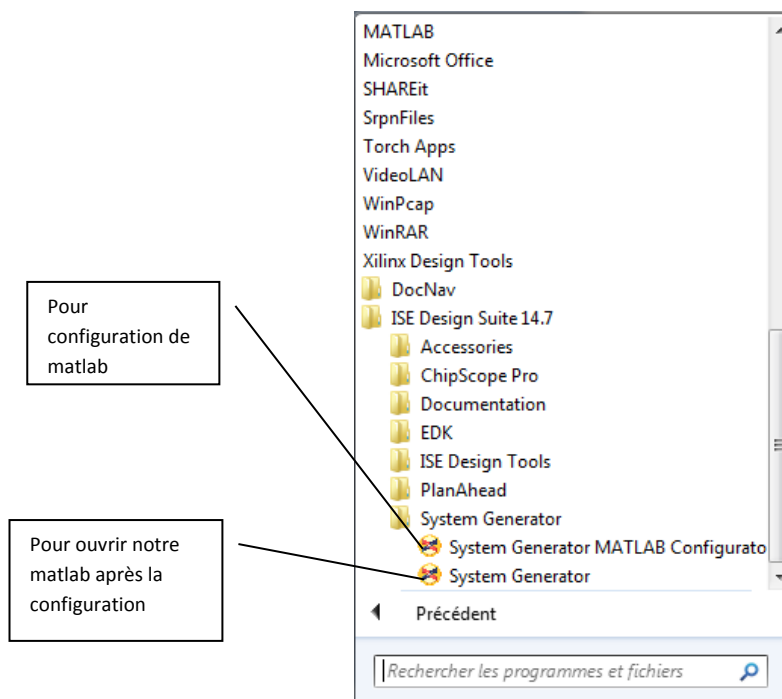


Figure2.7: étape de configuration de Matlab pour utiliser le system generator

La figure 2.7 montre la fenêtre de configuration

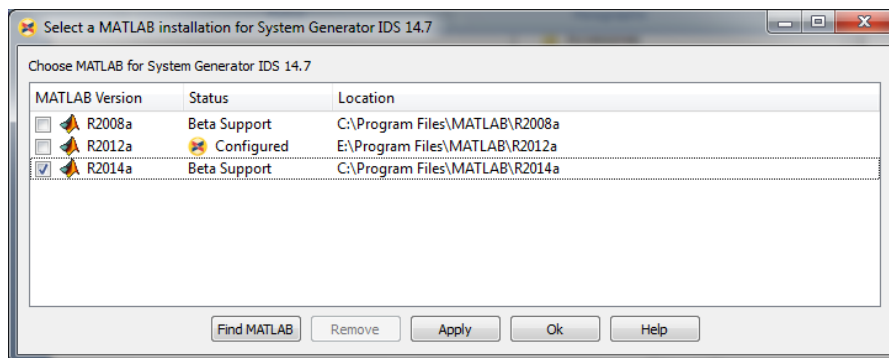
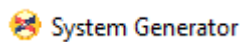


Figure 2.8 : la fenêtre de configuration matlab

Après la configuration on utilise le System generator pour ouvrir notre matlab configuré



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

Afin de pouvoir finaliser un FPGA, il est nécessaire d'utiliser un langage de description matériel ou bien un outil de saisie graphique. Après compilation de cette description, on obtient un fichier de configuration pour le FPGA choisi. VHDL et Verilog sont les deux langages de description les plus répandus, pour cela on a traité dans ce chapitre le System Generator qui nous aide à générer les fichiers VHDL d'une manière simple et facile sans avoir à maîtriser le langage de description VHDL, on va décrire cette méthode en détaille dans le chapitre suivant.

2.5 Conclusion

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

Afin de pouvoir finaliser un FPGA, il est nécessaire d'utiliser un langage de description matériel ou bien un outil de saisie graphique. Après compilation de cette description, on obtient un fichier de configuration pour le FPGA choisi. VHDL et Verilog sont les deux langages de description les plus répandus, pour cela on a traité dans ce chapitre le System

Generator qui nous aide à générer les fichiers VHDL d'une manière simple et facile sans avoir à maîtriser le langage de description VHDL.

Dans notre cas on va implémenter un modulateur ASK OOK sur carte FPGA à l'aide de system generator et on va voir les étapes en détail dans le chapitre suivant.

Chapitre 3

Implémentation d'un modulateur ASK2

Chapitre 3 Implémentation d'un modulateur ASK2

3.1.Introduction

Dans ce chapitre nous allons présenter notre travail, à savoir l'implémentation d'un modulateur ASK2 d'abord sous Matlab, puis sur une carte FPGA. Les blocs fonctionnels du modulateur seront implémentés et simulés sous Matlab en utilisant les logiciels Simulink, System Generator et ISE Design Suite. Une fois les résultats théoriques confirmés, le code VHDL sera testé sous XILINX ISE Fondation ; les résultats numériques de simulation des circuits seront utilisés pour tracer les courbes et vérifier si elles correspondent à celles obtenues sous Matlab . Des interprétations des résultats seront proposées, la figure 3.1 illustre les étapes de notre projet.

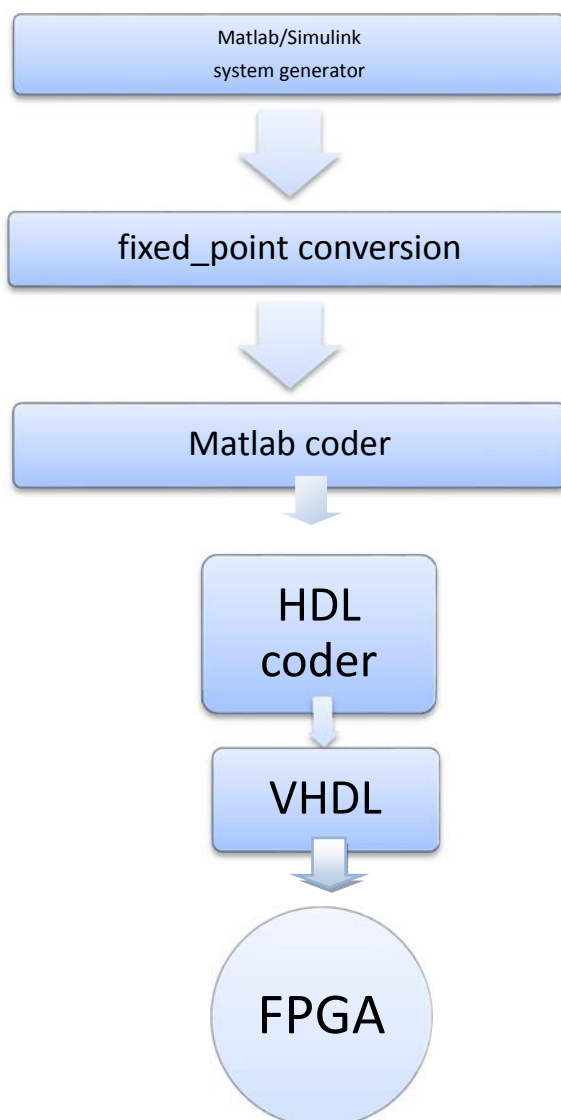


Figure 3.1 : organigramme de notre projet

3.2. Simulation d'un modulateur ASK2 sous Matlab

3.3.

Nous avons configuré Matlab, comme nous l'avons indiqué dans le chapitre précédent, de manière à générer le code VHDL de notre en utilisant le 'system generator', une fois que notre simulation aurait réussi.

Nous présentons dans cette section la simulation du modulateur OOK sous Matlab.

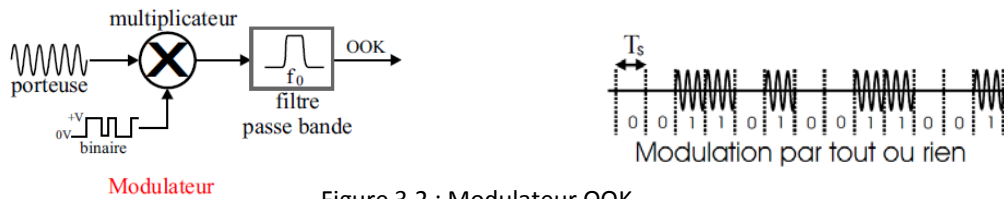


Figure 3.2 : Modulateur OOK

3.2.1. Implémentation de la porteuse

Les blocs de sinus et cosinus disponibles au niveau de Simulink sous Matlab, sont directement utilisables pour la simulation, mais leur code VHDL n'est pas implémentable. Nous avons utilisé un bloc CORDIC des fonctions trigonométriques (algorithme itératif consistant en une succession d'additions et de décalages), et pris une porteuse cosinus (fig.3.3).

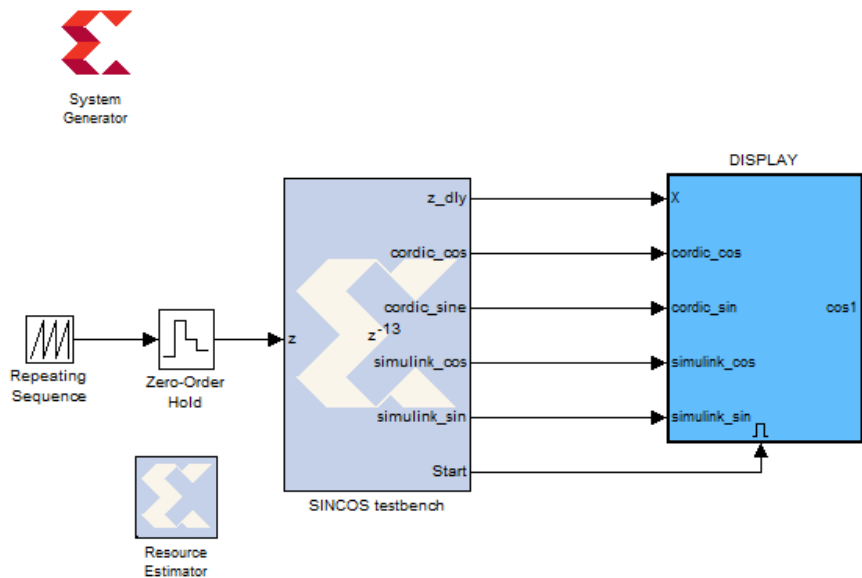


Figure 3.3 : le programme CORDIC d'un signal cosinus dans le system generator

3.2.2. Représentation des nombres en virgule fixe :

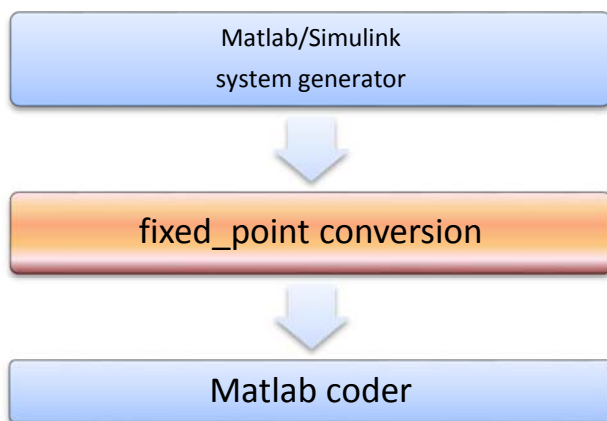


Figure 3.4 : étape conversion en virgule fixe dans un Modulateur OOK

Dans le système de modulation à simuler puis à réaliser matériellement sur FPGA, les données circulant entre les différents blocs, doivent être représentées en virgule fixe, ou en virgule flottante ; cette dernière nécessitant un nombre de portes logiques important, nous utiliserons la représentation des nombres en virgule fixe pour implémenter notre modulateur.

Un type de données à virgule fixe est caractérisé par la longueur de mot en bits, la position de la virgule binaire, et si elle est signée ou non signée. La position de la virgule binaire est le moyen par lequel les valeurs à virgule fixe sont mis à l'échelle et interprétées.

Une représentation commune d'un nombre à virgule fixe binaire, signé ou non, est montrée dans la figure 3.5.

Des modules de Xilinx de conversion des nombres en virgule fixe sont utilisés au niveau de chaque bus de données du modulateur en utilisant des blocs gateway_in (en entrée) et gateway_out(en sortie)

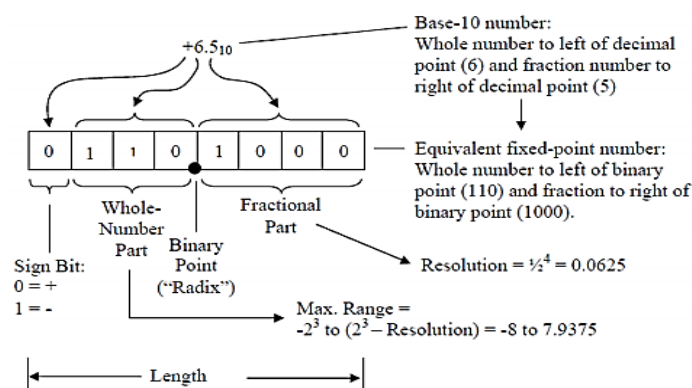


Figure 3.5 : représentation virgule fixe binaire des nombres réels

3.2.3. Matlabcoder : implémentation du modulateur sous Matlab

Dans la troisième étape on va présenter notre modulateur ASK2 programmé dans le system generator « Matlab coder » (fig.3.6)

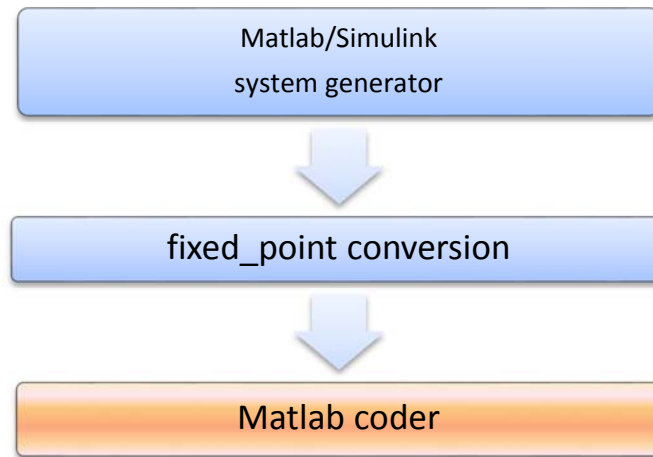


figure3.6 :étape Matlabcoder

Afin de créer notre modulateur (fig.3.7), nous multiplions notre message binaire (0,1) par la porteuse (sortie du bloc cosinus). A la sortie du modulateur, le bit '1' est une onde cos, et le bit '0' est un zéro. Un scope permet de visualiser le message binaire, la porteuse, et le signal modulé.

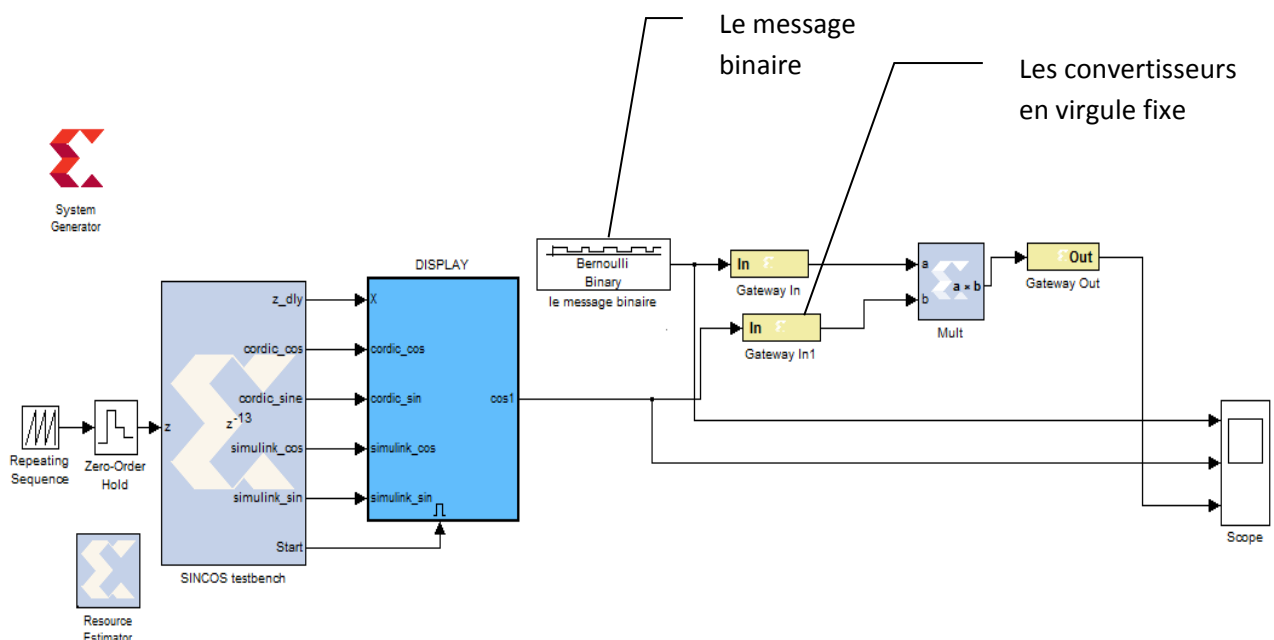


Figure3.7 : le schéma bloc d'un modulateur ASK2

le résultat de simulation de ce modulateur apparaît dans le scope (fig. 3.8) qui montre le signal message, le signal sinusoïdal et le signal modulé.



Figure 3.8 : les signaux dans le scope

3.2.4. HDL coder : Génération du code VHDL du modulateur sous Matlab

La simulation du modulateur sous Matlab ayant été validée, dans cette partie, on va générer le code VHDL du modulateur à l'aide du bloc system generator, pour pouvoir ouvrir notre modulateur à l'aide de ISE design system

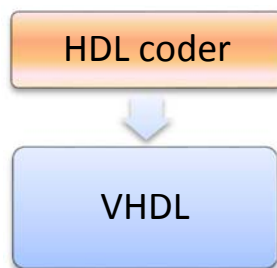


Figure 3.9 : le bloc HDL coder du system generator

La fenêtre de system generator permet de fixer les paramètres à utiliser pour la génération du code HDL, tels que le dossier de stockage des fichiers créés, le choix du langage HDL, le choix du circuit FPGA ciblé... Une fois ces paramètres fixés, l'activation du bouton 'generate' (fig. 3.10) entrainera la création du code VHDL, et son stockage dans fichier de type '.vhd.'

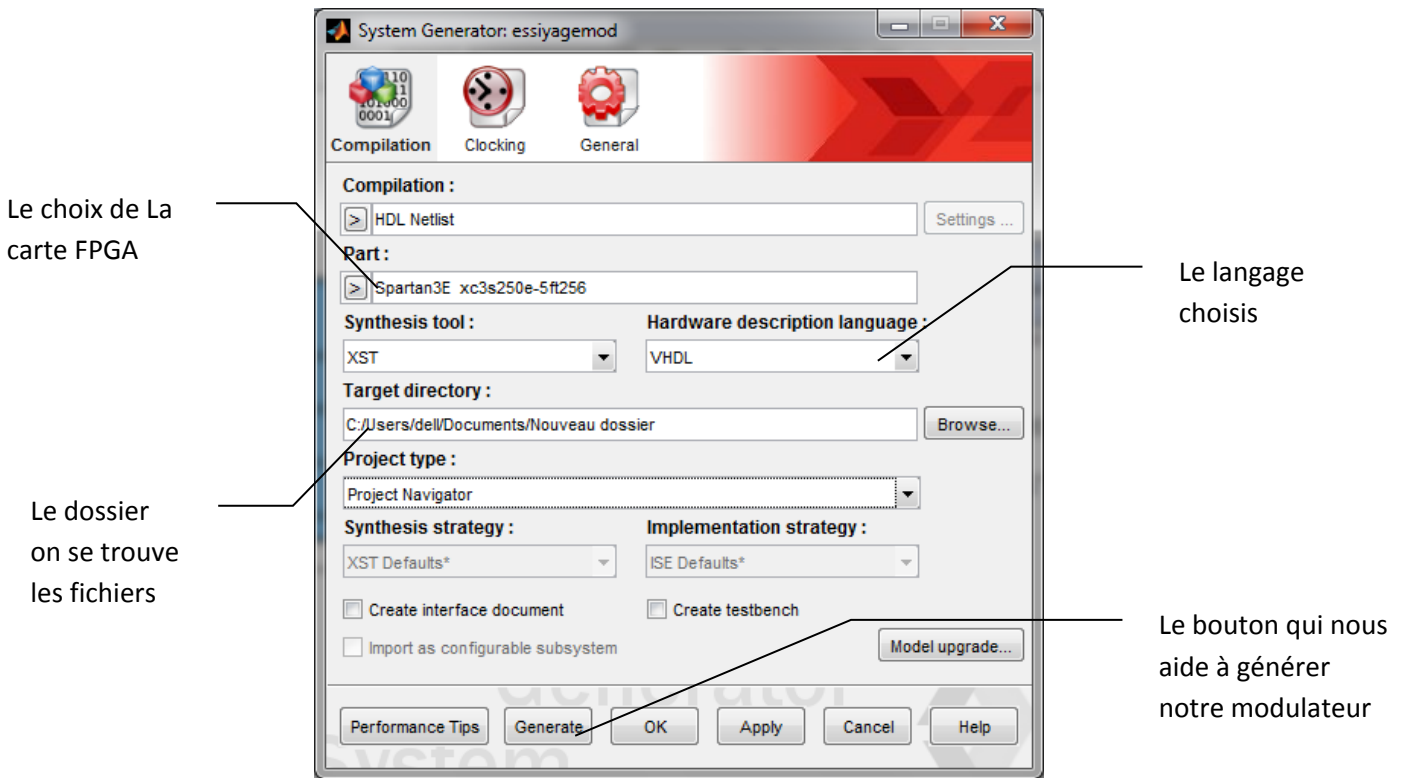


Figure3.10 : system generator

Après avoir généré notre modulateur par le bloc System generator on se trouve les fichiers dans le dossier choisi, la figure 3.11 montre les fichiers.

_xmmsgs	23/10/2016 21:38	Dossier de fichiers	
documentation	23/10/2016 21:38	Dossier de fichiers	
sysgen	23/10/2016 21:38	Dossier de fichiers	
addsb_11_0_55ba67abb0801391.ngc	19/08/2016 20:13	Fichier NGC	16 Ko
blockCoreMap	23/10/2016 21:38	Fichier	1 Ko
commandLines	23/10/2016 21:38	Fichier	1 Ko
globals	23/10/2016 21:38	Fichier	2 Ko
hdlFiles	23/10/2016 21:38	Fichier	1 Ko
isim_modulateur.prj	23/10/2016 21:38	Fichier PRJ	1 Ko
isim_modulateur_run.tcl	23/10/2016 21:38	Fichier TCL	1 Ko
modulateur.vhd	23/10/2016 21:38	Fichier VHD	181 Ko
modulateur_cw.gise	23/10/2016 21:38	Fichier GISE	2 Ko
modulateur_cw.sdc	23/10/2016 21:38	Fichier SDC	1 Ko
modulateur_cw	23/10/2016 21:38	Xilinx System Gen...	0 Ko
modulateur_cw.ucf	23/10/2016 21:38	Fichier UCF	1 Ko
modulateur_cw.vhd	23/10/2016 21:38	Fichier VHD	16 Ko
modulateur_cw.xcf	23/10/2016 21:38	Fichier XCF	1 Ko
modulateur_cw.xdc	23/10/2016 21:38	Fichier XDC	1 Ko
modulateur_cw	23/10/2016 21:38	Xilinx ISE Project	44 Ko
modulateur_cw_import	23/10/2016 21:38	Document texte	3 Ko
modulateur_gateway_in.dat	23/10/2016 21:38	Fichier DAT	9 Ko
modulateur_gateway_in1.dat	23/10/2016 21:38	Fichier DAT	9 Ko
modulateur_gateway_out.dat	23/10/2016 21:38	Fichier DAT	9 Ko
modulateur sincos testbench cos.dat	23/10/2016 21:38	Fichier DAT	9 Ko

Figure3.11 : les fichiers générés : le projet ISE et le top module 'vhd'

3.4. Simulation du modulateur ASK2 sous XilinxISE design system(integrate synthesis Environment)

C'est un outil créé par Xilinx pour faciliter la programmation en langage VHDL, la conception des schémas équivalents et la réalisation de testsbench pour la simulation, avant de mettre les informations sur la carte FPGA

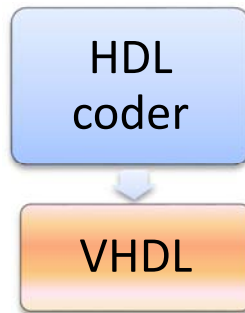
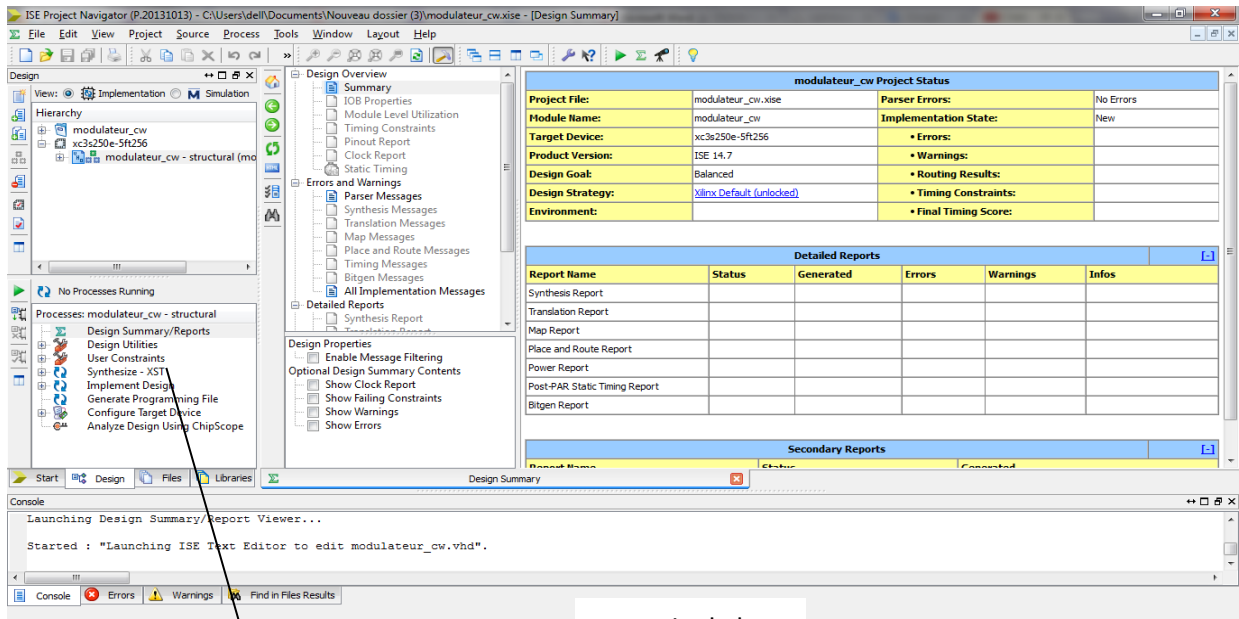


Figure 3.12 : étape simulation du code VHDL:

3.4.1. Ouverture du projet sous ISE

Et on va voir dans cette partie les figures des schémas et langage de VHDL et comment faire la simulation. La commande File ->open Project permet d'ouvrir le projet, puis d'utiliser les différentes commandes disponibles au niveau d'ISE pour synthétiser et simuler le circuit.



partie de la simulation

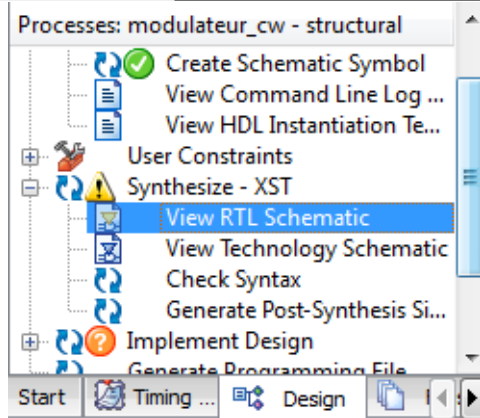


Figure 3.13: xilinx ISE system design

Les commandes de ISE permettent de vérifier la syntaxe vhd. Un fichier Test Bench est ensuite créé pour stimuler le circuit. Différentes commandes permettent de visualiser le schéma RTL (fig.3.14), et le schéma RTL détaillé (fig.3.14)

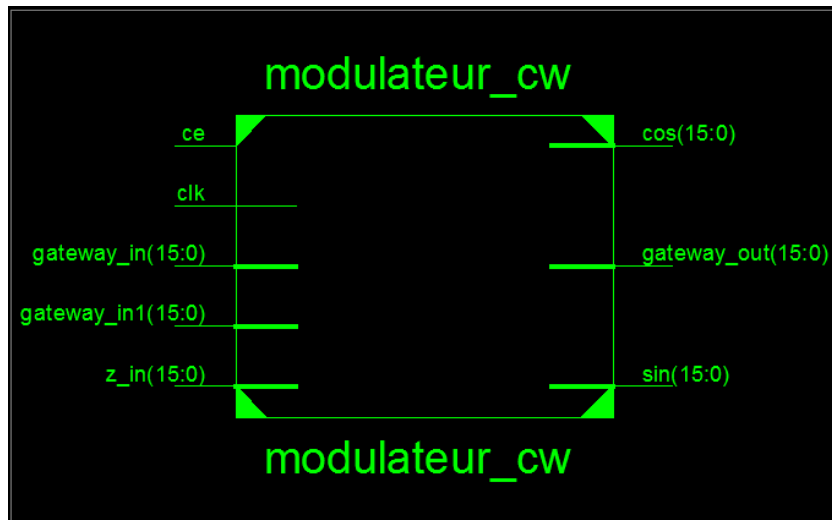


Figure 3.14 : le schéma RTL montre les entrées et sorties du circuit

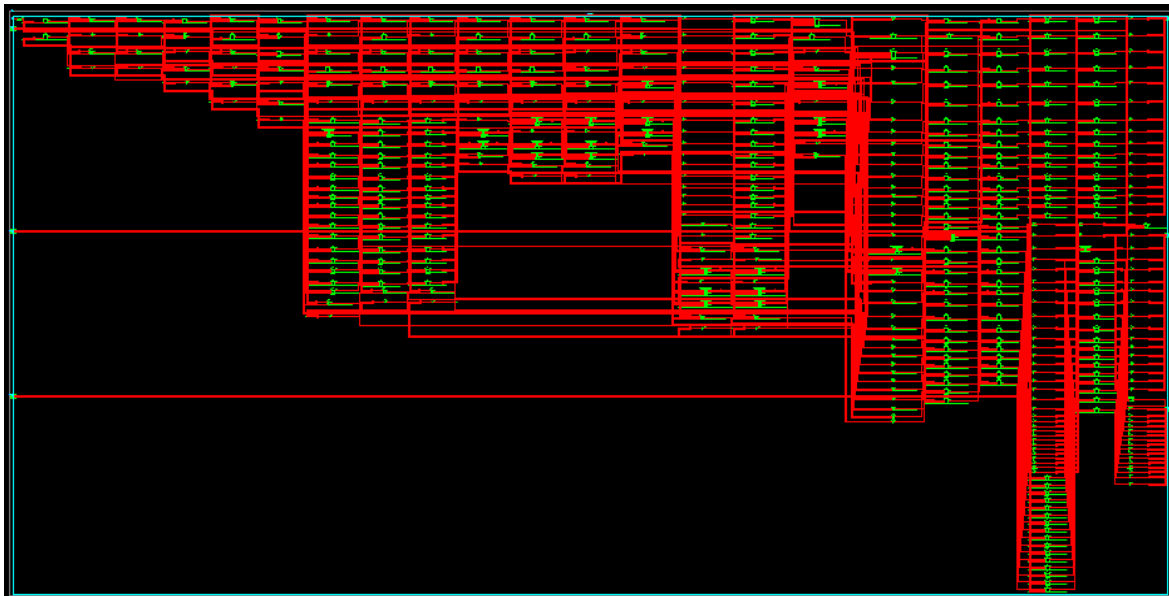


Figure 3.15 : le schéma RTL détaillé

3.4.2. Création du test bench

On fait le test bench pour tester les résultats de notre modulateur dans l'ISE, on va créer une nouvelle source et on va choisir le test bench pour notre modulateur, les figures suivantes illustrent comment va faire le test bench

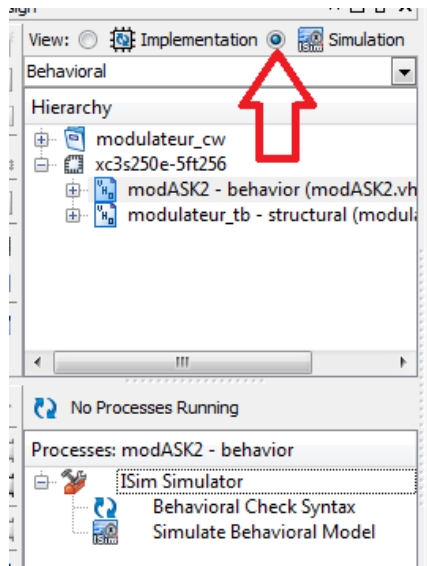


Figure 3.16 : la première étape pour faire le test bench

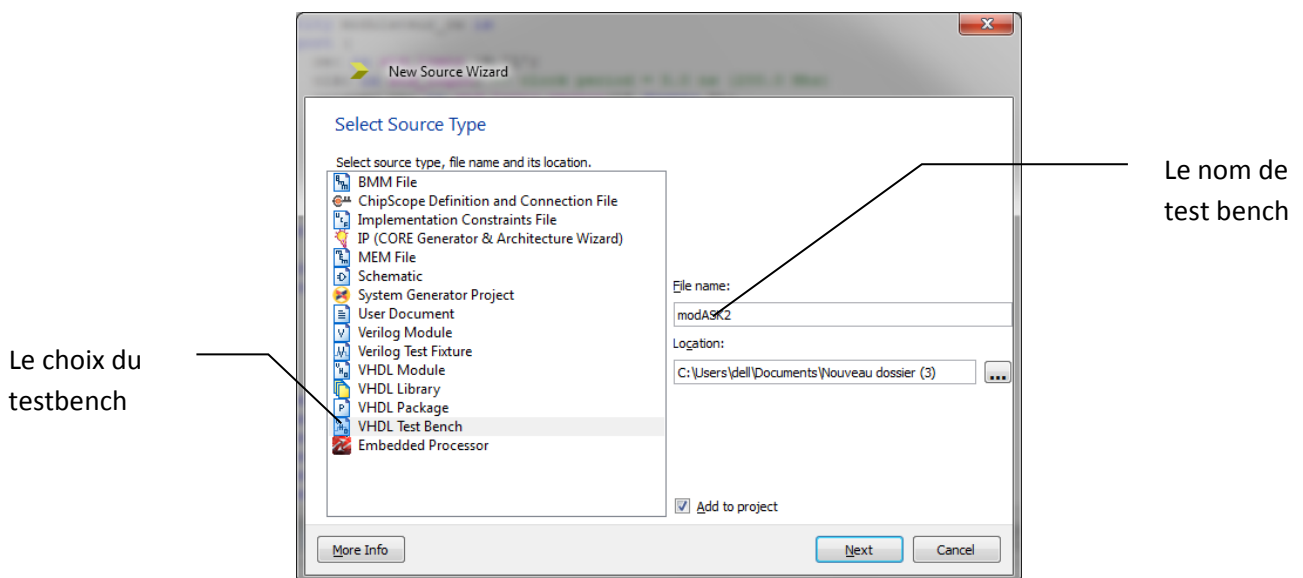


Figure 3.17 : création d'une nouvelle source

Après la création du fichier test bench(en utilisant 'create new source')on va trouver le fichier écrit par le VHDL comme illustre la figure 3.18 et on va faire des modifications décrites par la figure 3.19

```

88     clk <= '0';
89     wait for clk_period/2;
90     clk <= '1';
91     wait for clk_period/2;
92     end process;
93
94
95     -- Stimulus process
96     stim_proc: process
97     begin
98         -- hold reset state for 100 ns.
99         wait for 100 ns;
100
101         wait for clk_period*10;
102
103         -- insert stimulus here
104
105         wait;
106     end process;
107
108 END;
109

```

Figure 3.18 : ossature du test bench VHDL

```

85     -- Clock process definitions
86     --clk_process :process
87     --begin
88         --clk <= '0';
89         --wait for clk_period/2;
90         --clk <= '1';
91         --wait for clk_period/2;
92     --end process;
93
94
95     -- Stimulus process
96     stim_proc: process
97     begin
98         wait for 100 ns;
99         z_in <= not z_in
100     end process;
101
102     stim_proc1: process
103     begin
104         wait for 100 ns;
105         gateway_in <= not z_in
106     end process;
107
108     stim_proc2: process
109     begin
110         wait for 100 ns;
111         gateway_in <= not gateway_in
112     end process;

```

On peut annuler le clock

Figure 3.19 : le test bench avec stimuli prêt pour la simulation

3.4.3. Simulation et résultats

Après la simulation la figure 3.17 montre les signaux résultant de notre modulateur, le signal magenta montre le cosinus, le rouge aussi le cosinus dans les points fixes, le bleu c'est notre message binaire et le signal jaune décrit le signal modulé

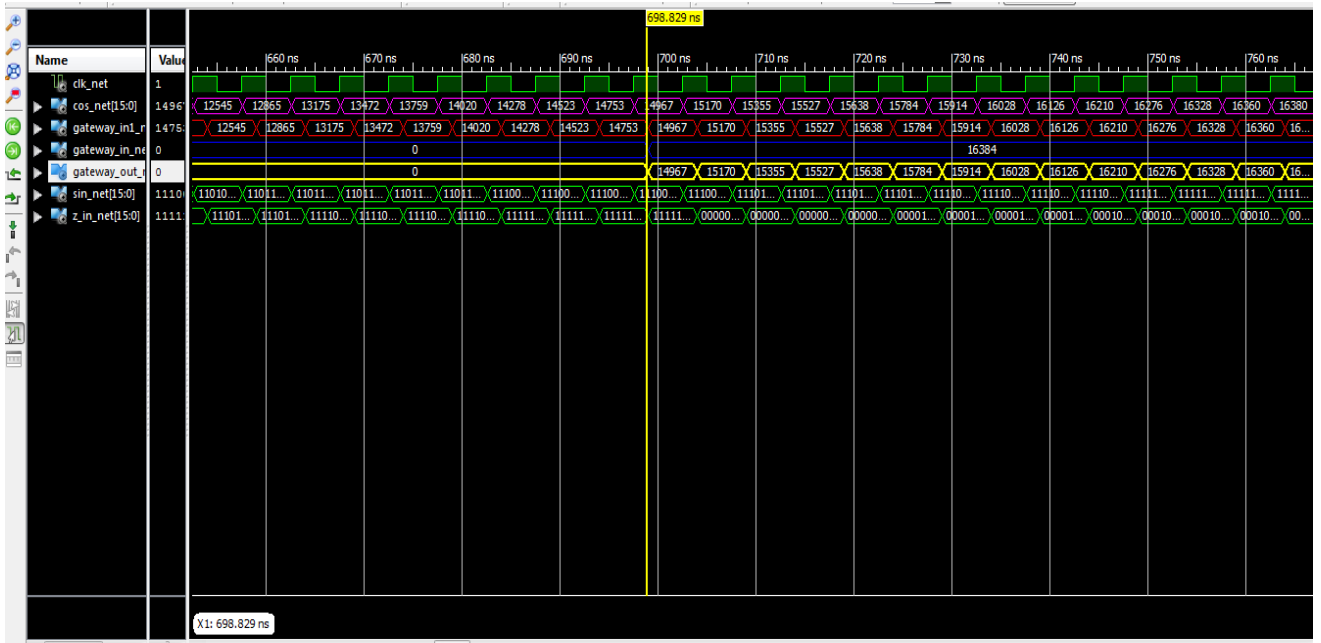


Figure 3.20 : les signaux de test bench

3.4.4. Validation des résultats

Après le test bench on va prendre les données extraites des résultats obtenus sous ISE, puis on les réinjecte dans Matlab pour tracer les courbes équivalentes la figure 3.19 montre le programme de ces courbes et la figure 3.20 illustre les courbes qui valide les résultats

```
File Edit Text Go Cell Tools Debug Desktop Window Help
+ - 1.0 + ÷ 1.1 x % %
1 - clear all
2 - gateway_in1=[-8192 -8186 -8174 -8151 -8123 -81
3 - subplot(3,1,1);
4 - plot(gateway_in1)
5 - gateway_in=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 - subplot(3,1,2);
7 - plot(gateway_in)
8 - gateway_out=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9 - subplot(3,1,3);
10 - plot(gateway_out)
```

Figure3.21 : la programmation des courbes

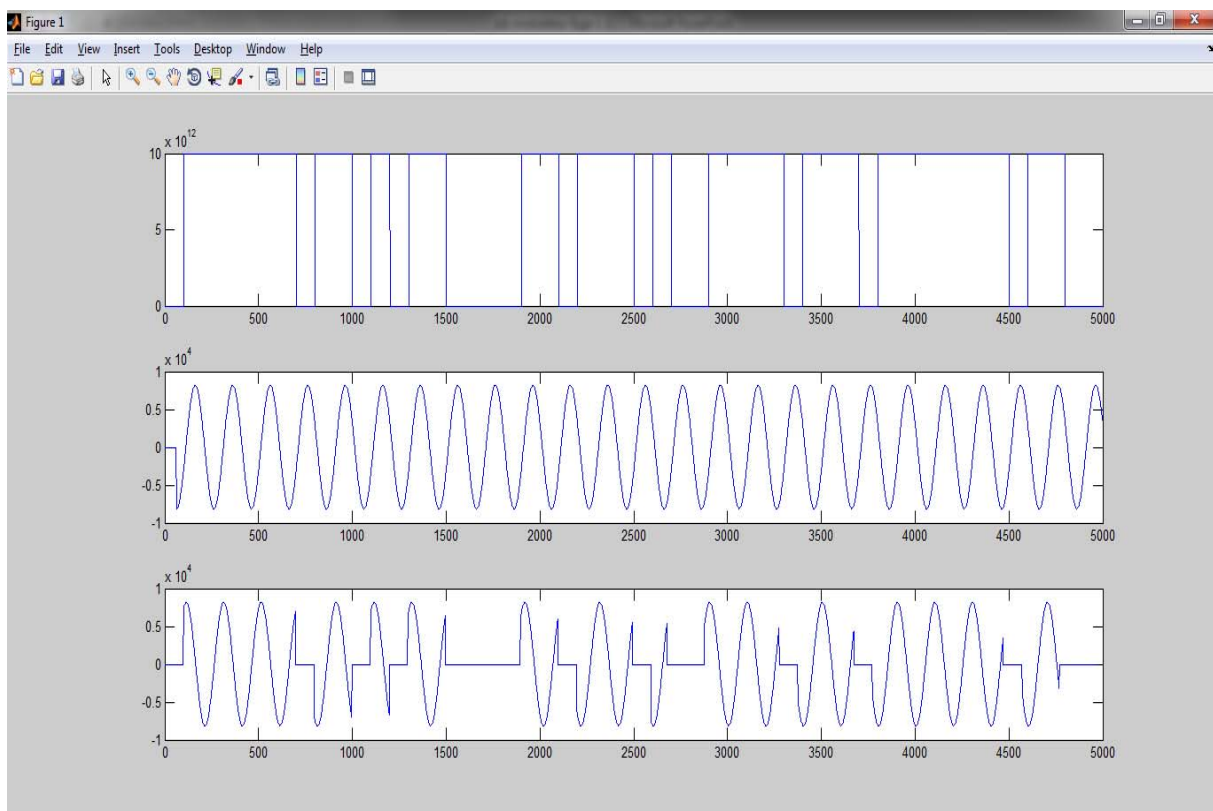


Figure 3.22 : les courbes de haut en bas : la porteuse, le message binaire, le message modulé

3.5. Conclusion

Le modulateur ASK2 a été conçu et implémenté avec succès sous le système generator. le code VHDL généré par matlab et simulé sous ISE design system, a été validé grâce aux résultats de modulation corrects obtenus pour les stimuli présentés au circuit. Notre objectif est donc atteint.

On a choisi le circuit FPGA Spartan3E, disponible au niveau du département d'Electronique de l'USDB afin d'augmenter nos chances d'implantation du circuit du modulateur sur le circuit FPGA.

Conclusion générale

Les systèmes numériques présentent de nombreux avantages dans le domaine de la transmission de l'information. Plus particulièrement, les circuits reconfigurables de type FPGA sont séduisants, par leur faible coût, leurs possibilités d'évolution importantes et leur intérêt économique pour les productions en petite série. Ils permettent de plus de passer directement de l'étape de prototypage à l'étape de produit fini. Mais le domaine des transmissions est dominé par les architectures analogiques, en particulier en ce qui concerne les couches les plus basses des protocoles, qui sont chargées de produire le signal à transmettre sur le médium. Car ces couches, dont les plus critiques sont celles chargées de la modulation ou de la démodulation.

La nouvelle méthodologie de conception que j'ai décrite dans ce mémoire se facilite l'implantation dans la carte FPGA sans faire l'algorithme en langage VHDL. Elle propose de nouvelles architectures traduisant directement les fonctionnalités nécessaires à la modulation de signaux en algorithmes adaptés aux circuits numériques, et aux FPGA en particulier. Notre modulateur ASK2, utilisant le logiciel system generator qui nous aide à décrire notre algorithme en VHDL et implanté sur un circuit FPGA en utilisant cette nouvelle méthodologie de conception. Ce système a été conçu de façon à tirer au mieux parti de la structure interne spécifique des circuits FPGA de type SPARTAN 3^E (ALTERA) sur lesquels ils sont implantés.

Primordiaux pour pouvoir intégrer ces algorithmes sur des systèmes embarqués. En- fin, nous nous sommes focalisés plus sur le type de modulation ASK2 et sur les algorithmes numériques capables de résoudre les problèmes d'implantation d'un modulateur sur FPGA que sur les protocoles réseaux existants. Une prochaine étape de cette étude doit prendre en compte plus précisément les spécifications des protocoles courants, comme par exemple la démodulation numérique ou bien les autres types de modulation. Il est déjà possible d'agir sur ces paramètres, en changeant la profondeur des pipelines utilisés dans le schéma bloc de system generator du modulateur. Le projet peut-être avantageusement étendu pour permettre la gestion en ligne, sans nécessiter de reprogrammation, de protocoles utilisant plusieurs types de modulation pour s'adapter par exemple à la qualité de la ligne de transmission surtout dans notre domaine aéronautique. En conclusion, nous avons démontré dans ce mémoire la faisabilité d'implémentation de la modulation numérique d'amplitude dans la carte FPGA, en nous focalisant plus particulièrement sur les étapes de faire se passage sans écrire l'algorithme en langage VHDL, nous avons développé une méthode de conception différente, qui permet d'utiliser le matlab et Xilinx ISE Design. Nous avons proposé des pistes d'amélioration pour permettre d'étendre cette méthodologie à des protocoles plus complexes, et d'intégrer les blocs fonctionnels conçus ici dans le cadre plus global d'un processeur réseau.

Annexes

Annexe A :

Les PLDs

A.1 Structure d'un PLD :

Un circuit PLD contient les éléments suivants :

- Un bloc d'entrées : permet de fournir l'état de chaque entrée et de son complément au bloc combinatoire.
- Un ensemble d'opérateurs « ET » : permet de connecter les variables d'entrées et leurs compléments.
- Un ensemble d'opérateurs « OU » : permet de connecter les sorties des opérateurs « ET ».
- Un bloc de sorties.
- Un bloc d'entrées/sorties: il comporte une porte à 3 états et une broche d'entrée/sortie. Le bloc combinatoire programmable est composé de matrices « ET » et de matrices « OU », les interconnexions de ces matrices doivent être programmables, ceci est réalisé par des fusibles qui sont grillés lors de la programmation. L'autre approche, radicalement opposée, « cellule universelle interconnectée » est de renoncer à la réduction en première forme normale des équations logiques. On divise le circuit en blocs logiques indépendants, interconnectés par des chemins de routage. Une fonction logique est récursivement décomposée en opérateurs simples, jusqu'à ce que les opérations élémentaires rentrent dans une cellule.

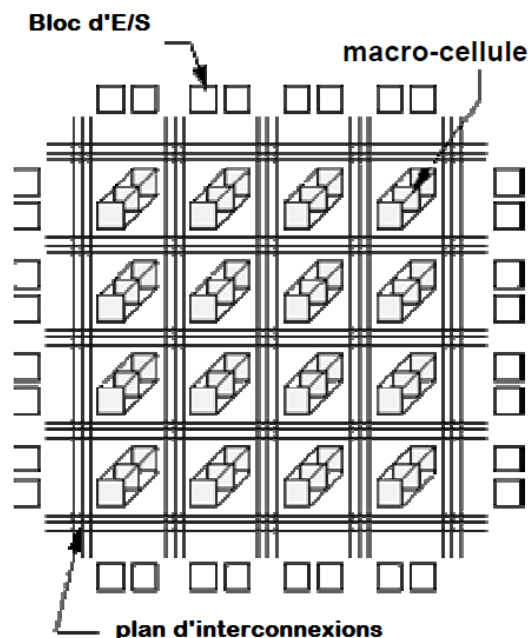


Figure A.1 : la structure d'un PLD

Le bloc de sortie est souvent appelé macro-cellule,OLMC (Output Logic macro Cell, macro-cellule logique de sortie). Elle fut l'élément clé dans le développement des circuits logiques

programmables. En effet, la macro-cellule procure au composant la flexibilité de configuration : entrée, sortie, entrée/sortie ou haute impédance. Elle comporte :

- Une porte « XOR », une bascule D.
- Des multiplexeurs qui permettent de définir les différentes configurations et un dispositif de rebouclage sur la matrice « ET ».
- Des fusibles de configuration.

Placement et routage :

Consiste à lier les blocs de calcul aux opérateurs logiques d'une fonction et à choisir les broches d'E/S. le routage consiste à créer les interconnexions nécessaires. Pour des PLD simples, le placement est relativement trivial et le routage inexistant. Les compilateurs génériques (indépendants du fondeur) effectuent très bien ces deux opérations. Pour les CPLDs, et plus encore les FPGA, ces deux opérations deviennent plus complexes et nécessitent un outil spécifique du fondeur.

A.2 Critère de choix des PLDs :

Les principaux critères de choix des P.L.D. sont:

- ✓ La densité d'intégration.
- ✓ La rapidité de fonctionnement (paramètre directement lié à l'impédance intrinsèque des éléments programmables.
- ✓ La facilité de mise en œuvre (programmation, reprogrammation...).
- ✓ La possibilité de maintien de l'information.
- ✓ Le nombre de portes.
- ✓ La consommation.
- ✓ La technologie et le prix.

A.3 Avantages des PLD :

- Le coût et temps de développement.
- Le prototypage et temps de simulation accéléré.
- La prise de risque est réduite. (Modifications possibles, risque d'inventaire)
- Des supports logiciels peu chers.
- Des produits éprouvés. (Pas de vecteurs de test structurel)

A.4 Inconvénients des PLD :

- Un niveau d'intégration moindre aux ASIC.
- Une consommation plus élevée.
- Des prix importants et une faible disponibilité pour de grandes séries.

Annexe B :

Les PALs (Programmable Array Logic) :

Généralité :

Les PALs ont eu un grand succès dès leur parution dans le marché, ce fut les premiers circuits logiques programmables.

Un PAL est un composant relativement simple, dérivé des PROM (Programmable Read Only Memory, mémoire morte à lecture seule programmable une fois). Les ingénieurs de MMI ont combiné la technologie à fusibles (utilisée pour les mémoires PROM) avec des portes ET, OU pour réaliser des fonctions logiques. La compréhension de la cellule de base d'un PAL suffit car c'est la même qui se répète sur l'étendue de la capacité de celui-ci voir la figure X. la cellule de base se compose d'un buffer d'entrée qui dispose de l'information et de son complément (tous les PALs sans exception disposent d'un certain nombre d'entrées qui aboutissent toutes, sous forme directe et inversé, sur la matrice de fusible de programmation), suivit de la matrice à fusibles puis des portes ET (considérées en entrée) puis suivit de portes OU en sortie.

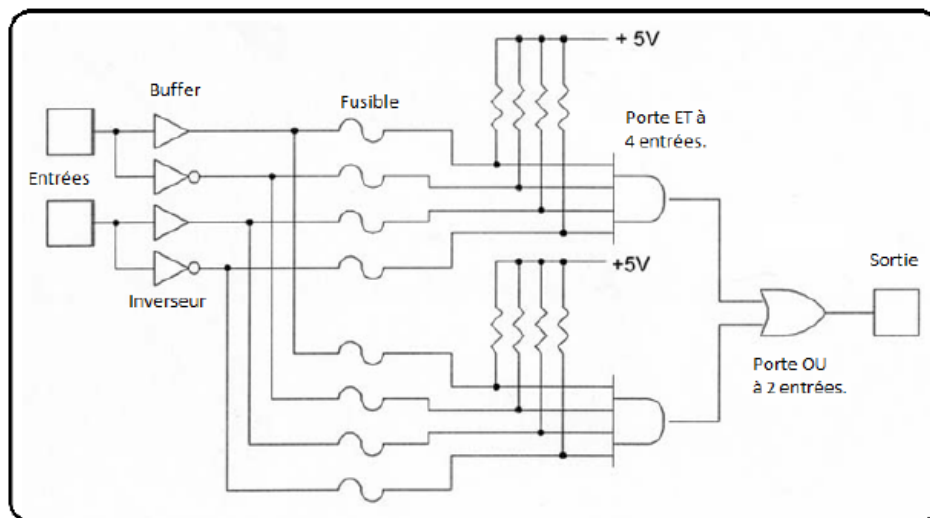


Figure B.1 : Cellule de base d'un PAL (architecture combinatoire)

La programmation d'un PAL s'effectue par la destruction de fusibles, à l'aide d'un programmeur dédié en appliquant des tensions de programmation requises.

L'inconvénient majeur des PALs, c'est qu'une fois programmé, ils ne sont plus effaçables car le fusible est détruit c'est contraignant en cas d'erreur de programmation ou de mise à jour.

Annexe C :

Les GALs (Generic Array Logic) :

Les GALs ne sont rien d'autre (d'un point de vue architectural) que des PALs reprogrammables. D'un point de vue technologique au lieu d'utiliser des transistors bipolaires, ils ont utilisé des transistors MOSFET pouvant être régénérés. Cette possibilité de régénération des fusibles sans réduire la durée de vie du composant.

Après étude des besoins du marché LATTICE SEMICONDUCTOR se fixe 4 objectifs pour la mise au point des GALs :

1. Offrir des produits ayant des vitesses de travail comparable à celle des PALs bipolaires, tout en étant testable à 100%.
2. Permettre un remplacement, au moins fonctionnel, mais idéalement broche pour broche, des PALs bipolaires dans n'importe quelle application.
3. Offrir une consommation beaucoup plus faible que les PALs bipolaire d'une complexité équivalente.
4. Proposer une plus grande souplesse de configuration des entrées/sorties que les PALs bipolaire.

C'est ainsi que LATTICE à palier aux inconvénients majeurs des PALs pour donner naissance à un composant lui imposant une forte concurrence.

Annexe D :

Les EPLDs (Erasable Programmable Logic Device) :

Les EPLDs sont des circuits réalisés en technologie CMOS, présentant l'avantage de faible consommation électrique, mais qui augmente en fonction de la fréquence. Ils disposent d'une macro-cellule plus évolué que celle des PALs, en plus ils sont effaçables.

L'introduction des EPLDs telle que l'a voulu ALTERA visait deux buts distincts :

1. Permettre une densité d'intégration nettement supérieur à celle offerte par les PALs et aussi proche que possible que celle permise par les réseaux de portes programmables.
2. Fonctionner à une vitesse égale à celle des PALs bipolaires et en tout cas nettement supérieur à celle des portes traditionnels.

Annexe E :

Les CPLDs

L'architecture typique d'un CPLD se présente comme un ensemble de fonctions de type PAL pouvant être interconnectées à l'aide d'une matrice. La physionomie est généralement très structurée. Un certain nombre de macros cellules de base sont regroupées pour former des blocs logiques.

La complexité, le nombre de macros cellules dans un bloc ainsi le nombre de blocs varie d'un composant à l'autre. On peut considérer deux niveaux d'interconnexion : une matrice globale et un système de distribution des signaux intégrés à chaque bloc logique.

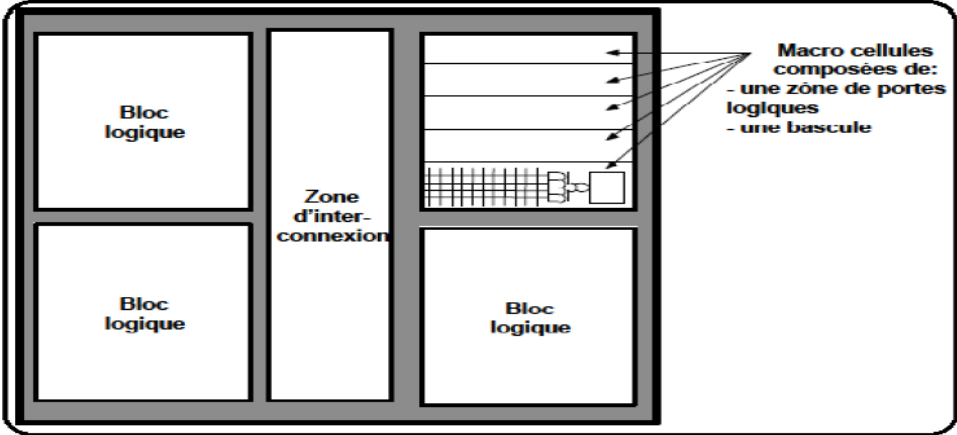


Figure E.1 : Architecture globale d'un CPLD.

Références bibliographique

- [1] Rajib Das, "FPGA implementation of digital modulation schemes : BPSK and QPSK using VHDL", these, année 2014
- [2] Virendrakumar V. Raut, "IMPLEMENTATION OF DIGITAL MODULATION TECHNIQUES IN MATLAB", these, July 2014
- [3] Thierry ROCACHER, "Modulateurs et démodulateurs", these, année 2010/2011
- [4] Emmanuel Boutillon, Jean-Luc Danger, "Implantation d'un démodulateur numérique sur FPGA", these, septembre 1997
- [5] Zartash Afzal Uzmi, "Digital Phase Modulation and Démodulation", these aout 2007
- [6] Mr.Pankaj Taklikar, Dr. Sanjay Badjate, Prof. Madhav Deshpande "Implementation of ASK ,PSK and FSK on FPGA", cours, l'année 2014
- [7] Steven DERRIEN, " Étude quantitative des techniques de partitionnement de réseaux de processeurs pour l'implantation sur circuits FPGA", these, 02 décembre 2002
- [8] Eduardo Sanchez, "Circuits reconfigurables: Les FPGAs", document pdf
- [9] Sparsh Mittal, Saket Gupta, and S. Dasgupta " System Generator: The State-of-art FPGA Design tool for DSP Applications ", these, année 2013
- [10] Eleena Mohapatra, "A Novel Design of QPSK Modulator for High Data Rate Transmission" these, année 2014
- [11] <https://www.altera.com/support/literature/lit-index.html>
- [12] www.xilinx.com
- [13] www.mathworks.com
- [14] www.technologuepro.com
- [15] <http://www.ebooks-gratuit.org/documents/pdf>