REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE

DEPARTEMENT D'ELECTRONIQUE

Mémoire De Fin D'étude

En vue d'obtention D'un Diplôme De Master En Electronique

Option

Réseaux Et Télécommunication

thème

ETUDE ET IMPLEMENTATION D'UN MODULATEUR FSK SUR UN CIRCUIT FPGA

Proposé Par :

Mr : K.BENNAMANE

Présenté par :

Melle : DJAMA SOUHILA

Melle : MANSEUR AMINA

Promotion

2010-2011

REMERCIEMENTS

Nous tenons avant tout de remercier le bon DIEU qui nous a donné la volonté et le courage pour la réalisation de ce modeste travail.

Nous remercions vivement Mr. k.bennamane notre promoteur pour ça précieuse assistance, ses aidées, ses conseils, ses critiques et son soutien qu'il nous accordé tout au long de ce projet.

Nos remerciements s'adressent également à monsieur le président de jury et les membres du jury pour l'honneur d'avoir assister à notre soutenance et juger ce travail.

Dédicaces

Je dédie ce travail

- *A la personne la plus importante dans ma vie , ma MAMAN chérie.*
- *A mon très chère frère AHMED et sa femme NAIMA.*
- *A la prunelle de mes yeux ma très chère sœur YASMINE.*
- *A mon très chère da OMAR et ses fille AMEL et DYHIA.*
- *A ma très chère tante HORIA et son petit garçon AHMED.*
- *A mon très chère beau frère FAOUZI.*
- *A toute ma famille sans exception, surtout mes très chers cousins, WIWI, HASSEN et AZOUAOU.*
- *A tout mes amis sans exception, AHMED, SABRINA ,CHABHA et surtout a ma très chère amie SAMIA.*
- *A mon très cher binôme SOUHILA.*
- *A toute ma promotion MASTER II.*

AMINA.M

Dédicaces

Je dédie ce travail

- *A la personne la plus chère au monde à mes yeux ma **MAMAN.***
- *A mes très chers frères, **RAFIK** et **NOUR.***
- *A mes très chères sœurs, **ZAHRA, LYNDA, HADJIRA.***
- *A mon très chère beau frère **DJAMEL.***
- *A toute ma famille sans exception.*
- *A tous mes amis sans exception, **lyes, KATIA, NASSIMA, NADIA, TITICH, RIMA, KARIMA, MONIA.***
- *A mon très cher binôme **AMINA.***
- *A toute ma promotion **MASTER II***

SOUHILA.DJ

SOMMAIRE

Chapitre II : Cartes et outils de développement des circuits FPGA

Chapitre III : Implémentation d'un Modulateur FSK sur FPGA

Liste des figures

• **PLD :** Programmable Logic Device - Dispositif logique programmable

• **CPLD :** Complex Programmable Logic Device - Réseau logique programmable complexe

• **EEPROM** Electrically Erasable PROM - Mémoire à lecture seule, électriquement effaçable

• **EPLD** Erasable Programmable Logic Device - Réseau logique programmable effaçable

• **EPROM** Erasable Programmable Read Only Memory - Mémoire à lecture seule effaçable

• **FPGA :** Field Programmable Gate Array - Réseau de portes programmables

• **FPROM:** Field Programmable Read Only Memory

• **GAL:** Generic Array Logic - PAL générique

• **ISP:** In-System (In Situ) Programmable - Composant programmable sur carte

• **LCA :** Logic Cell Array (Xilinx) - Réseau de cellules logiques

• **LUT :** Look-Up Table

• **PAL : Programmable** Array Logic - Réseau logique programmable

• **PROM : Programmable** Read Only Memory - Mémoire à lecture seule programmable

• **RAM : Random** Access Memory - Mémoire à accès aléatoire

• **ROM: Read** Only Memory - Mémoire à lecture seule

• **SRAM : Static** Random Access Memory - Mémoire statique à accès aléatoire

• **TTL :** Transistor Transistor Logic - Logique transistor-transistor

• **VHDL :** VHSIC Hardware Description Language - Langage de description matérielle VHSIC

• **CODEC:** Codeur-Décodeur

• **Cordon:** Câble relativement court équiper d'un connecteur à au moins l'une de ses deux extrémités

INTRODUCTION GENERALE

Introduction générale

L'électronique moderne se tourne de plus en plus vers le numérique qui présente de nombreux avantages sur l'analogique. Parmi ces avantages, nous citons entre autres la grande insensibilité aux parasites et aux dérives diverses, la modularité et la reconfigurabilité, et la facilité de stockage de l'information.

Cependant, les circuits numériques nécessitent une architecture plus lourde et leur mode de traitement de l'information met en œuvre plus de fonctions élémentaires que l'analogique d'où l'inconvénient majeur des temps de traitement. Pour y combler, les fabricants de circuits intégrés numériques s'attachent à fournir des circuits présentant des densités d'intégration toujours plus élevée, pour des vitesses de fonctionnement de plus en plus grandes.

D'abord réalisées avec des circuits SSI (**S**mall **S**cale **I**ntegration) les fonctions logiques intégrées se sont développées avec la mise au point du transistor MOS dont la facilité d'intégration a permis la réalisation de circuits MSI (**M**edium **S**cale **I**ntegration) puis LSI (**L**arge **S**cale **I**ntegratio**n**) puis VLSI (**V**ery **L**arge **S**cale **I**ntegration).

Au début des années 70 sont apparus les premiers composants (en technologie bipolaire) entièrement configurable par programmation. La nouveauté résidait dans le fait qu'il était maintenant possible d'implanter physiquement par simple programmation, au sein du circuit, n'importe quelle fonction logique. D'abord dédiés à des fonctions simples en combinatoire (décodage d'adresse par exemple), ces circuits laissent aujourd'hui au concepteur la possibilité d'implanter des composants.

La plupart de ces circuits sont maintenant programmés à partir d'un simple ordinateur (type PC) directement sur la carte où ils vont être utilisés. En cas d'erreur, ils sont reprogrammables électriquement sans avoir à extraire le composant de son environnement.

Les circuits FPGAs sont aujourd'hui les principaux circuits reconfigurables disponibles sur le marché. Ils sont aujourd'hui en mesure de fournir une solution efficace à la réalisation matérielle d'applications dans de nombreux domaines, les FPGAs offrent une surface homogène d'unités logiques universelles qui peuvent être reconfigurées à volonté de manière à implémenter n'importe quel circuit combinatoire ou séquentiel. Les circuits FPGA ont gagné énormément d'importance dans le domaine de l'informatique.

Grâce aux progrès technologiques, la capacité et la complexité de ces circuits reprogrammables sont en continuelle progression.

Dans ce mémoire nous nous intéresserons à l'étude, la conception, la simulation et la réalisation d'un modulateur FSK sur un circuit reconfigurable de type FPGA cyclone II (ALTERA).

Pour cela le plan suivant a été adopté :

Le premier chapitre est consacré à l'Etat de l'art des circuits logiques programmable.

Le second chapitre à la présentation des Cartes et outils de développement des circuits FPGA des familles XILINX et ALTERA.

Le troisième chapitre quant à lui il sera entièrement consacré à la présentation des résultats obtenus lors de l'étude et de l'implémentation d'un modulateur FSK sur carte DE1 de la famille ALTERA.

Enfin ,nous terminerons notre travail par une conclusion générale et quelques perspectives futures.

Chapitre I

Etat de l’art des circuits logique programmable

I Introduction

Il y a quelques années la réalisation d’un montage en électronique numérique impliquait l’utilisation d’un nombre important de circuits intégrés logiques. Ceci avait pour conséquences un prix de revient élevé, et un circuit imprimé de taille importante. Il est nécessaire de passer par un fondeur pour réaliser les circuits, ce qui introduit un délai de quelques mois dans le processus de conception. Cet inconvénient a conduit les fabricants à proposer des circuits programmables par l'utilisateur (sans passage par le fondeur) qui sont devenus au fil des années, de plus en plus évolués. Rassemblés sous le terme générique PLD.

Le développement des mémoires utilisées en informatique fut à l’origine des premiers circuits logiques programmables (PLD : programmable logic devices). Ce type de produit peut intégrer dans un seul circuit plusieurs fonctions logiques programmables par l’utilisateur. Sa mise en œuvre se fait très facilement à l’aide d’un programmateur, d’un micro-ordinateur et d’un logiciel adapté.

II Historique

Le principe de la logique programmable remonte au début des années 1960, le concept ayant été proposé par G. Estrin. Il a cependant fallu attendre les années 1980 pour que les premières réalisations matérielles apparaissent sur le marché. L’apparition de ce type de circuit s’est d’abord faite au travers de circuits logiques programmables simples de type PAL (Programmable Array Logic), qui se programment comme des mémoires non volatiles de type ROM et sont utilisés pour implémenter des fonctions combinatoires simples, telles des décodeurs d’adresse, ou des contrôleurs de bus.
Avec les évolutions en micro-électronique, différentes familles de circuits programmables ont commencé à apparaître : les CPLD (Complex Logic Programmable Devices), puis les FPGA (Field Programmable Gate Arrays), introduits par la société Xilinx en 1985.L’industrialisation de ce type de circuits s’est faite à grande échelle avec l’apparition de circuits de plus en plus performants et reprogrammables à volonté.

Les circuits de type FPGA les plus récents offrent désormais l’équivalent de dix millions de portes logiques programmables, à des fréquences de fonctionnement atteignant les 200 MHz.

À l'heure actuelle, on compte une dizaine de fabricants, le marché étant nettement dominé par les sociétés Xilinx, Altera (circuits reprogrammables) et Actel (circuits non reprogrammables).

III Présentation des circuits logique programmable

Les circuits logique programmable sont des circuits disposants d'entrées et de sorties dont l'utilisateur peut programmer le schéma logique d'après les besoins liées à la fonction souhaitée : Logique combinatoire et/ou séquentielle. Ces composants sont appelés PLDs (Programmable Logic Devices), Ce type de produit peut intégrer dans un seul circuit plusieurs fonctions logiques programmables par l'utilisateur. Sa mise en œuvre se fait très facilement à l'aide d'un programmateur, d'un micro-ordinateur et d'un logiciel adapté.

Les circuits logiques programmables sont des composants standards programmables électriquement une seule fois (fusible) ou reprogrammable.

IV Les différentes familles des PLDs

Le schéma suivant résume les différents circuits programmables disponibles sur le marché :

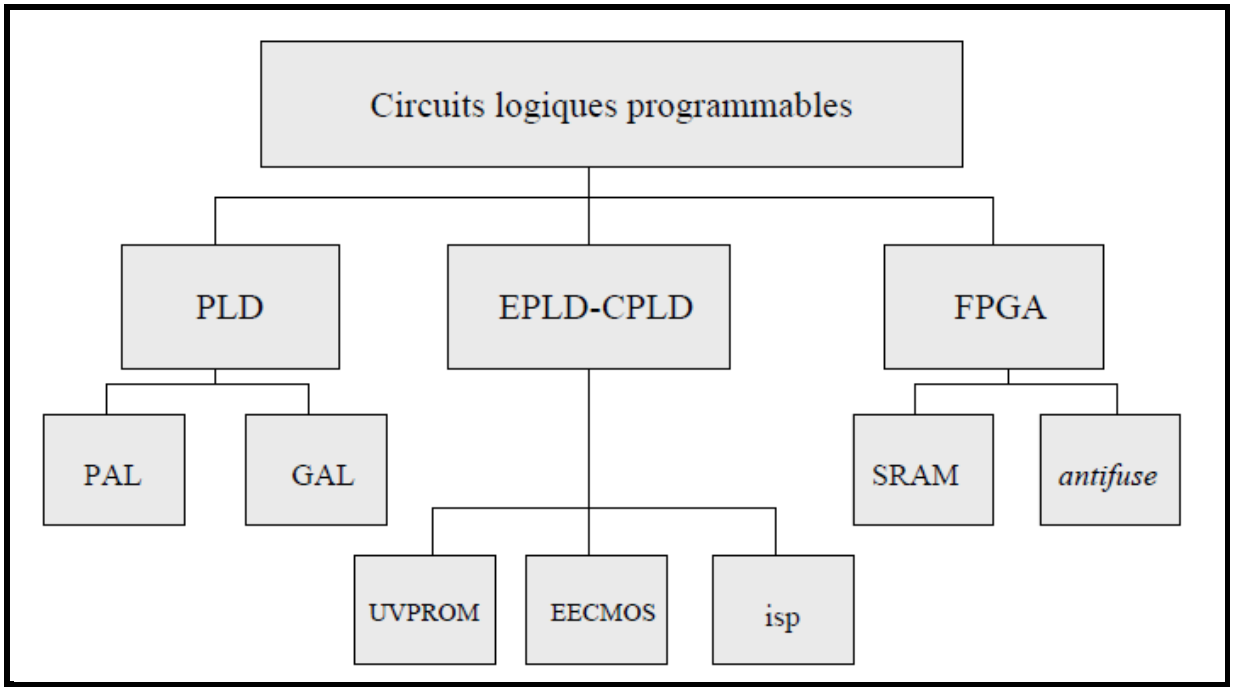


Figure I-1 : les différents circuits programmables

- PLD : Programmable Logic Devices composés de réseau de ET et de OU.
- EPLD : Erasable Programmable Logic Devices ou CPLD (Complex Programmable Logic Devices) constitué de plusieurs blocs de type PLD reliés par un réseau d'interconnexions.
- FPGA : Field Programmable Gate Array ou LCA (Logic Cell Array) (Xilinx), matrices de cellules simples identiques reliables par des interconnexions programmables.

IV-1 Les PLDs

IV-1-1 Structure de base d'un PLD

La plupart des PLDs suivent la structure suivante :

- Un ensemble d'opérateurs « ET » sur lesquels viennent se connecter les variables d'entrée et leurs compléments.
- Un ensemble d'opérateurs « OU » sur lesquels les sorties des opérateurs « ET » sont connectées.

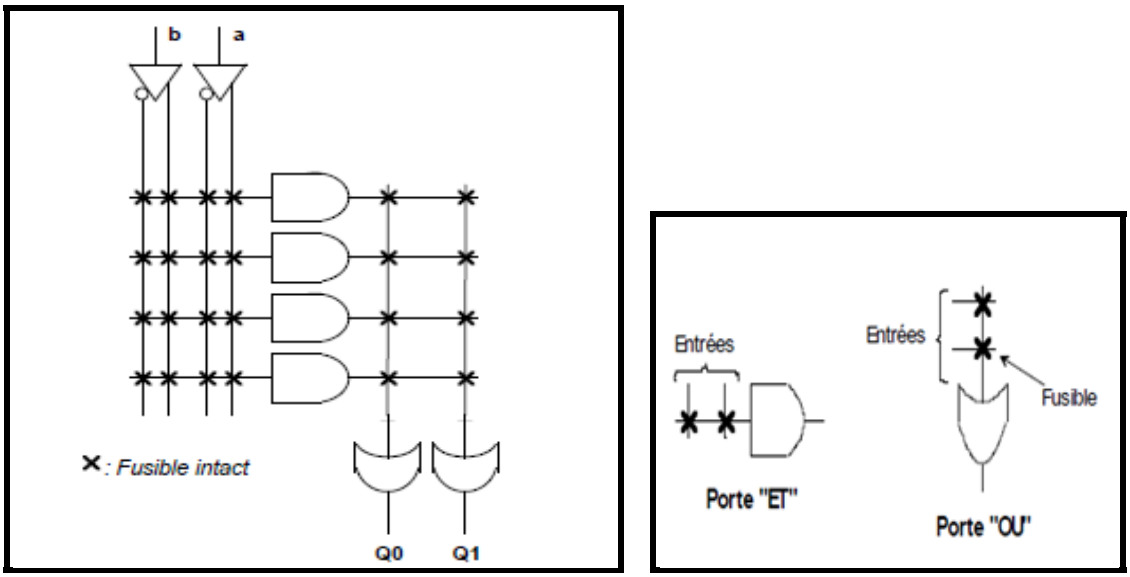


Figure I-2: Structure de base d'un PLD

Les deux ensembles forment chacun ce qu'on appelle une **matrice**. Les interconnexions de ces matrices doivent être programmables. C'est la raison pour laquelle elles sont assurées par des **fusibles** qui sont « grillés » lors de la programmation. Lorsqu'un PLD est vierge toutes les connexions sont assurées.

IV-1-2 Caractéristique des PLDs

Un circuit logique programmable PLD est caractérisé par

- La densité de quelques centaines de portes.
- L'architecture ET/OU programmable.
- Le nombre de terme produit par sortie
- Le retard de propagation (vitesse).
- La consommation de puissance.
- La conception sous forme de langages d'équations.

IV-1-3 Les PALs

La famille La plus ancienne et la plus connue PAL. (Programmable Array Logic), c'est à dire réseau logique programmable, Ils sont introduits sur le marché au début des années 70. Permettait uniquement la réalisation de fonctions combinatoires. Très rapidement, l'apparition de composants intégrant des bascules D (en plus du réseau combinatoire) a permis la réalisation de systèmes synchrones. Ce type de composants de faible densité est aujourd'hui communément désigné sous l'appellation PLD (Programmable Logic Device).La structure des PAL est constituée d'un réseau de ET programmables et d'un réseau de OU fixes. Sa structure logique est la suivante

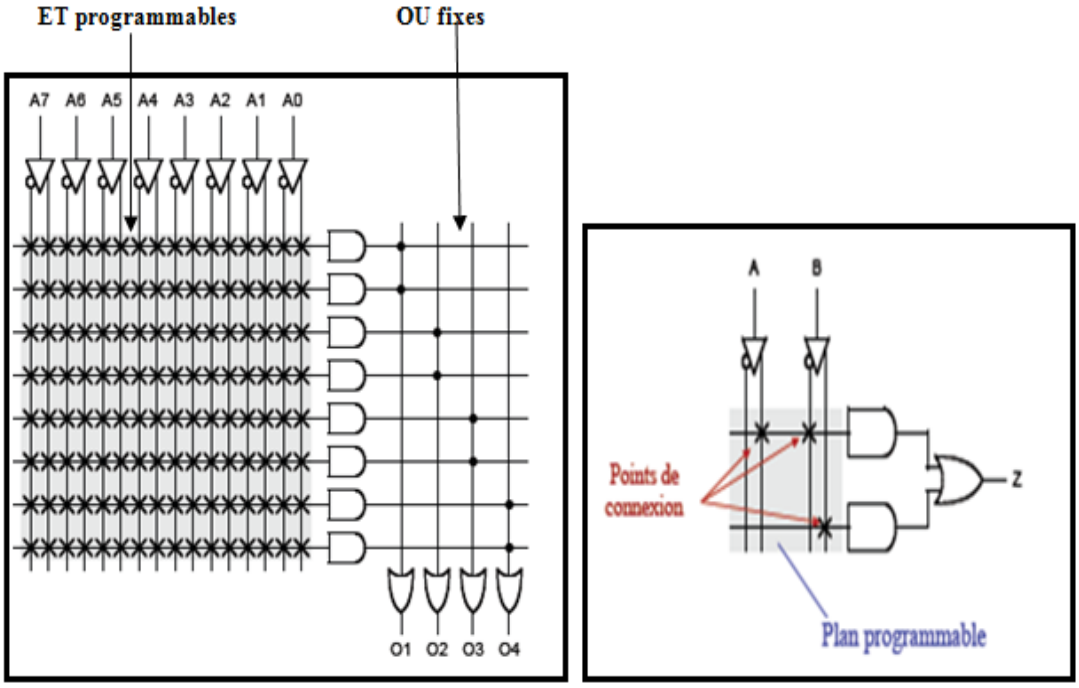


Figure I-3 : La structure logique de PAL

Exemple de programmation d'un PAL

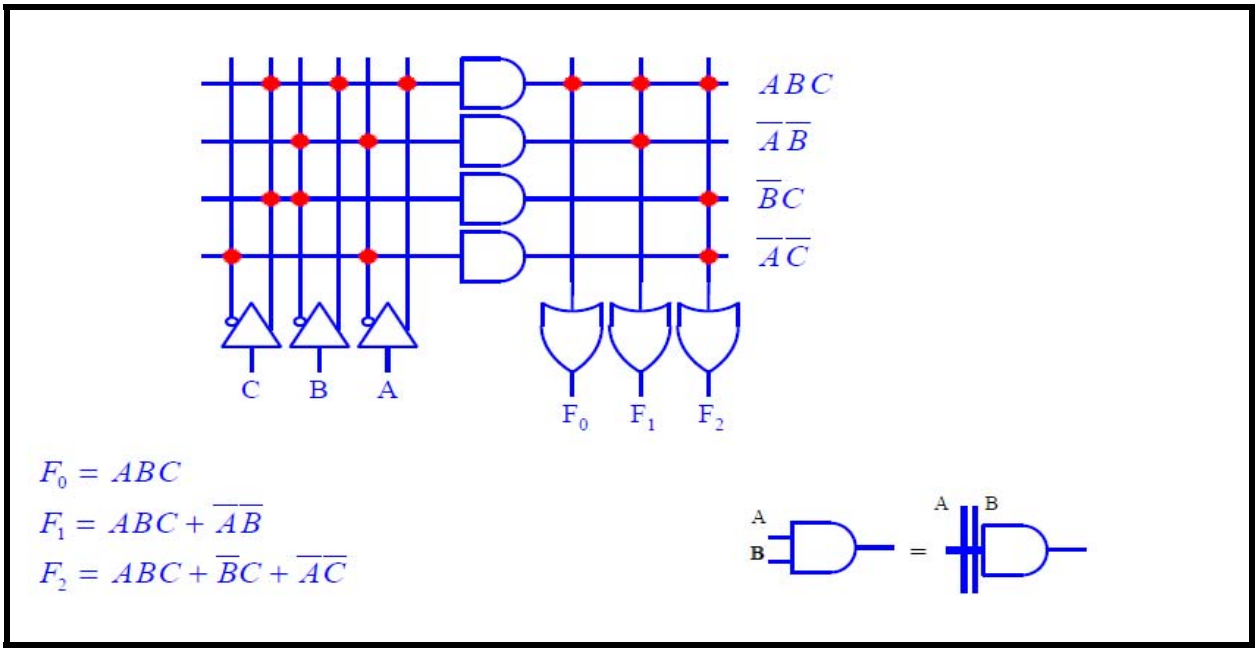


Figure I-4 : programmation d'un PAL

La programmation de ces circuits s'effectue par destruction de fusibles. Une fois programmée on ne peut plus les effacer. On distingue deux sous familles:

- Les P.A.L.s combinatoires ou P.A.L. simples sont constitués de fonctions de logique combinatoire.
- Les P.A.L.s à registres ou F.P.L.S. (Field Programmable Logic Séquencer) sont constitués de logique combinatoire et séquentielle (Registre).

IV-1-3-1 Caractéristique des PALs

Les circuits logique programmable PALs (Programmable Array Logic) sont caractérisé par :

- Le nombre d'entrées varie entre 10 et 22.
- Le nombre de sorties varie entre 1 et 10.
- La puissance est indiquée par une lettre code.
- La vitesse indique le temps de propagation en nS.

L'inconvénient majeur des PALs est qu'ils ne sont programmables qu'une seule fois. Ce qui impose un gaspillage important de ces circuits lorsqu'on veut développer un nouveau produit. Ceci a donc donné naissance aux GALs.

IV-1-4 Les GALs

Generic **A**rray **L**ogic que l'on pourrait traduire par « Réseau logique Générique ». Ces circuits peuvent donc être reprogrammés à volonté sans pour autant avoir une durée de vie restreinte. On peut aussi noter que dans leur structure interne les GALs sont constitués de transistor CMOS alors que les PALs classiques sont constitués de transistors bipolaires. La consommation des GALs est donc beaucoup plus faible. Depuis d'autres constructeurs fabriquent ce type de produit en les appelant « PAL CMOS ».

IV-1-4-1 Caractéristique des GALs

Les caractéristiques des GALs sont

- programmables et effaçables électriquement ; mêmes références que les PALs
- plus de souplesse d'utilisation
- architecture : macro cellule : bloc logique configurable

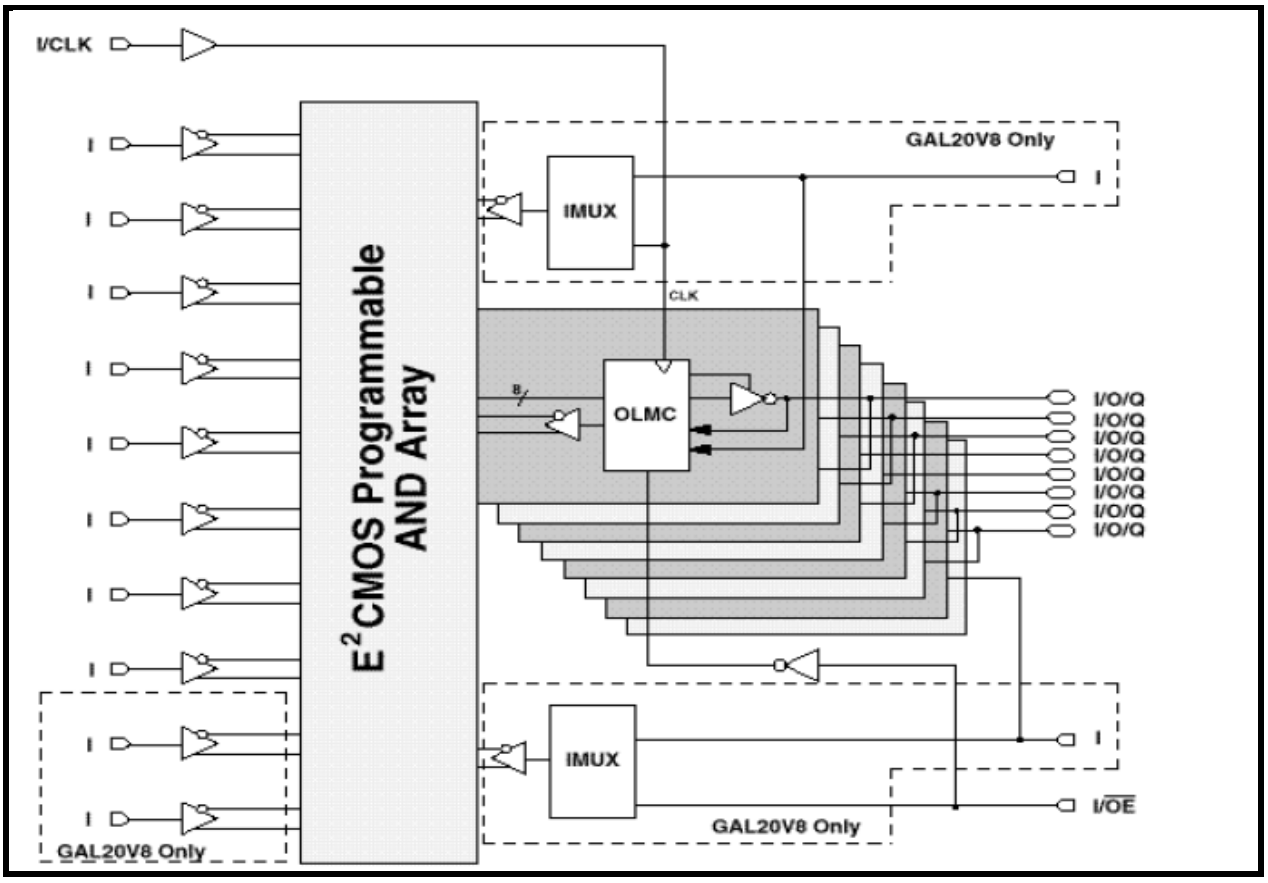


Figure I-5: Architecture de GAL

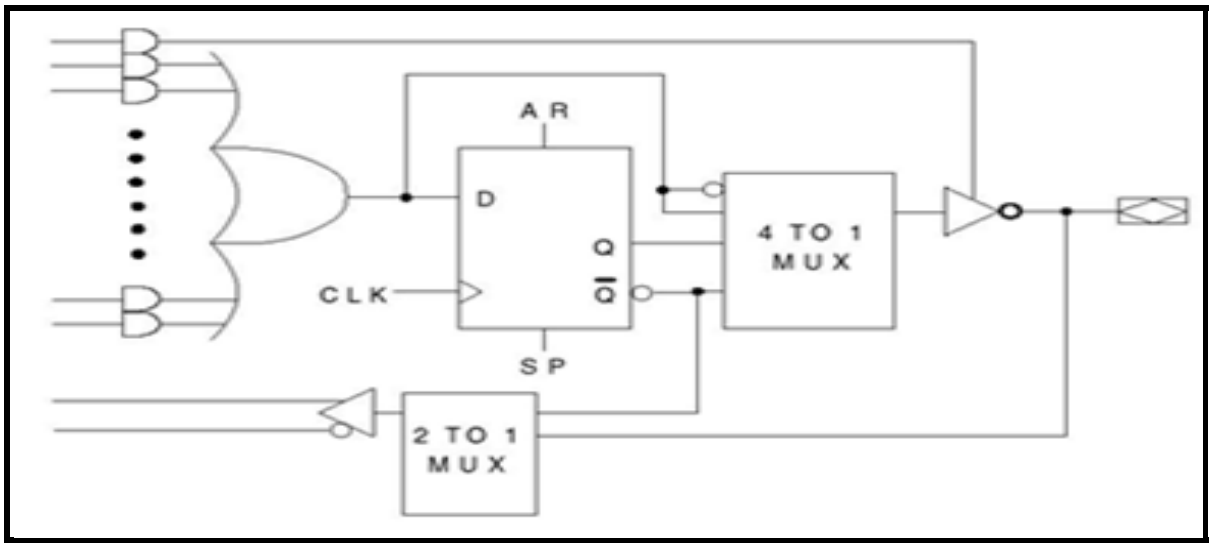


Figure I-6: Architecture d'Output logic macro cell **(OLMC ou GAL22V10)**

IV-1-5 Les avantages des PLDs

- Ils sont entièrement programmables par l'utilisateur,
- Ils sont généralement reprogrammables dans l'application, ce qui facilite la mise au point et garantit la possibilité d'évolution,
- les délais de conception sont réduits, il n'y a pas de passage chez le fondeur
- Composant de faible coût car produits en grand nombre.
- La fonction logique réalisée peut être modifiée par programmation, donc sans qu'il soit nécessaire de redessiner un nouveau circuit imprimé.

IV-1-6 Inconvénients des PLDs

- Impossibilité d'implémenter des fonctions multi niveaux,
- Impossibilité de partager des produits entre fonctions.
- les ressources d'interconnexion utilisent en général les 2/3 de la surface de silicium.

IV-2 Les CPLDs (**C**omplex **P**rogrammable **L**ogic **D**evices)

Les CPLDs sont aussi appelés EPLDs (Erasable Programmable logic Devices), se sont des circuits programmables électriquement et effaçables, soit par exposition aux UV pour les plus anciens, soit électriquement. Ces circuits, développés en premier par la firme ALTERA, sont arrivés sur le marché en 1985.

Les EPLDs sont une évolution importante des PALs CMOS. Ils sont basés sur le même principe pour la réalisation des fonctions logiques de base.

Les procédés physiques d'intégration permis par les EPLDs sont nettement plus importants que ceux autorisés par les PAL CMOS. En effet, les plus gros EPLD actuellement commercialisés intègrent plusieurs dizaines de milliers de portes utilisables par l'utilisateur. On peut ainsi loger dans un seul boîtier, l'équivalent d'un schéma logique utilisant jusqu'à 50 à 100 PAL classiques.

IV-2-1 Architecture des 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. Un certains nombre de macro cellules de base sont regroupés pour former des blocs logiques.

Les macro-cellules occupent une grande surface, mais elles présentent plusieurs intérêts :

- grand nombre de variables d'entrée possible
- grand nombre de termes de produits possibles

Voir les deux Figures suivantes :

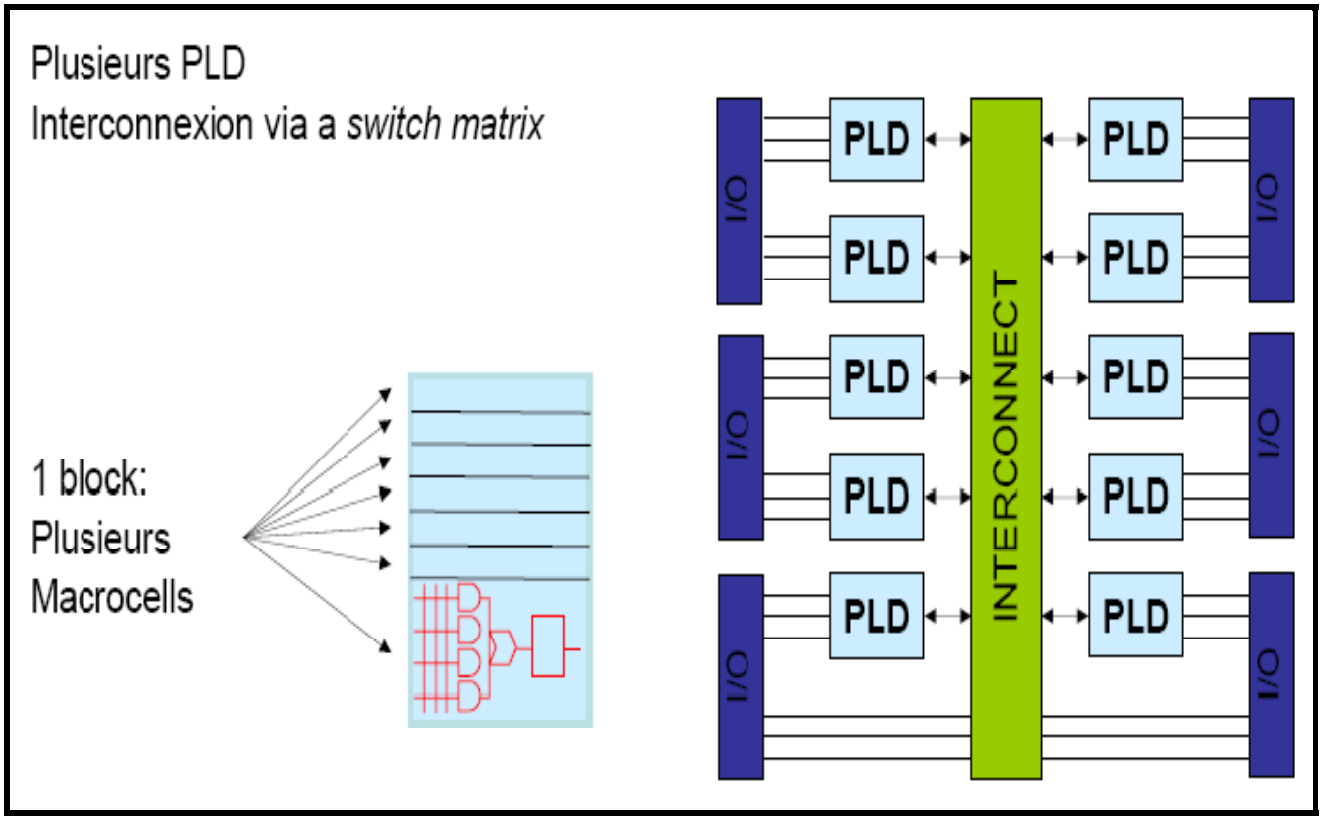


Figure I-7: Architecture générale d'un CPLD

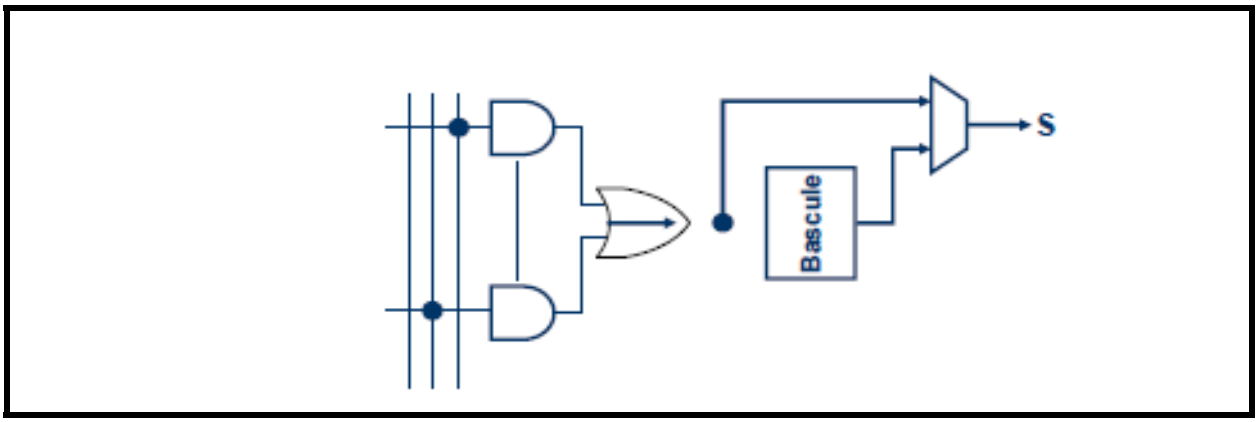


Figure I-8: Architecture des macro-cellules

IV-2-2 Les types des CPLDs

Il existe plusieurs types d'EPLD en technologie CMOS :

- Les circuits programmables électriquement et effaçables aux UV. UVPROM.
- Les circuits programmables électriquement et effaçables électriquement dans un programmateur. EECMOS.
- Les circuits programmables électriquement et effaçables électriquement sur la carte (ISP : In Situ Programmable), utilisant une tension unique de 5 V.

Les CPLDs sont des composants pour la plupart reprogrammables, peu chers et très rapides (fréquence de fonctionnement élevée) mais avec une capacité fonctionnelle moindre que les FPGA.Les FPGA permettent d'atteindre un niveau d'intégration plus élevé que celui des CPLD .

En première approximation, un FPGA est un CPLD avec un très grand nombre de macro-cellules et une grande souplesse d'interconnexion entre les macro-cellules.

En plus du niveau d'intégration, la différance entre CPLD et FPGA réside dans la maitrise du temps de propagation dans les couches logiques du circuit. Ce temps est prédictif dans les CPLD car les chemins parcourus par les signaux sont connus alors que dans les FPGAs ce temps dépend de l'organisation et de la distance entre les macro-cellules interconnectées.

IV-3 FPGA (**F**ield **P**rogrammable **G**ate **A**rays)

Lancée sur le marché en 1985 par la firme XILINX, le FPGA (**F**ield **P**rogrammable **G**ate **A**rays) est un circuit pré diffusé programmable par l'utilisateur et essentiellement constitués de trois parties :

- Une matrice de blocs logiques configurables CLB (configurable logic Bloc).
- Des blocs d'entrée/sortie configurables.
- Un Réseau d'interconnections programmable.

La figure I-9 suivante présente l'architecture générique d'un FPGA :

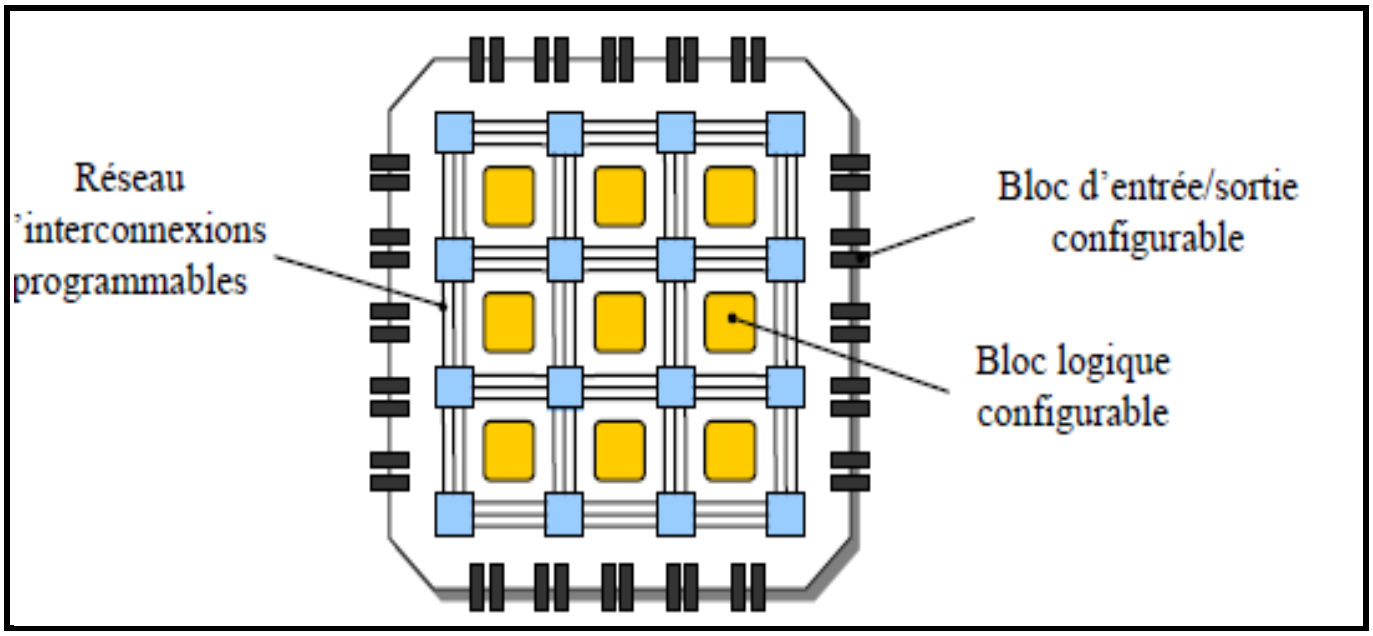


Figure I-9 : Architecture générique d'un FPGA

Les FPGA sont des ensembles de blocs logiques élémentaires que l'utilisateur peut interconnecter pour réaliser les fonctions logiques de son choix. La densité des portes est importante et sans cesse en évolution.

Le concept du FPGA est basé sur l'utilisation d'un multiplexeur comme élément combinatoire de la cellule de base. La figure I-10 suivante représente la cellule type de base d'un FPGA. Elle comprend un multiplexeur 8 vers 1 permettant de réaliser n'importe quelle fonction logique combinatoire de 4 variables (LUT : Look Up Table ou encore générateur de fonction). La bascule D permet la réalisation de fonctions logiques séquentielles.
La configuration du multiplexeur 2 vers 1 de sortie autorise la sélection des deux types de fonction.

Les cellules de base d'un FPGA sont disposées en rangées et en colonnes. Des lignes d'interconnexions programmables traversent le circuit, horizontalement et verticalement, entre les diverses cellules. Ces lignes d'interconnexions permettent de relier les cellules entre elles, et avec les plocs d'entrées/sorties.

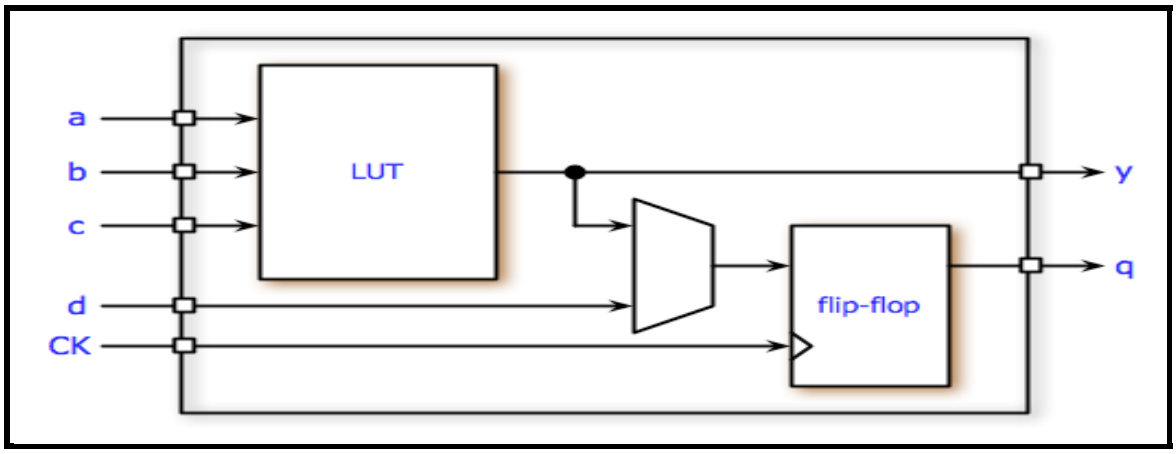


Figure I-10 : la cellule de base d'un FPGA.

La fonction de la LUT est de stocker la table de vérité de la fonction combinatoire à implémenter dans la cellule.

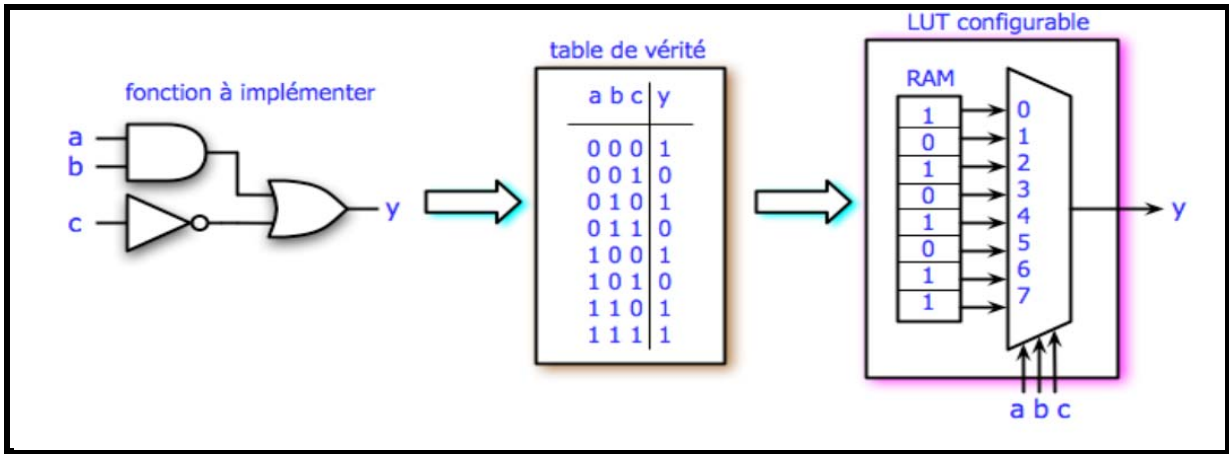


Figure I-11 : la cellule LUT

IV-3-1 L'interconnexion de FPGA

On distingue trois types de segments de lignes horizontales et verticales qui servent à interconnecter les différents blocs de FPGA

IV-3-1-1 L'interconnexion directe

Ces interconnexions permettent l'établissement de liaisons entre les CLB et les IOB avec un maximum d'efficacité en termes de vitesse et d'occupation du circuit. De plus, il est possible de connecter directement certaines entrées d'un CLB aux sorties d'un autre.

IV-3-1-2 Les longues lignes

Les longues lignes sont de longs segments métallisés parcourant toute la longueur et la largeur du composant, elles permettent éventuellement de transmettre avec un minimum de retard les signaux entre les différents éléments dans le but d'assurer un synchronisme. De plus, ces longues lignes permettent d'éviter la multiplicité des points d'interconnexion

IV-3-1-3 L'interconnexion simple

Ce système fonctionne en grille de segments métalliques verticaux et segments horizontaux positionnés entre les rangées et les colonnes de CLB et d'IOB. Ces interconnexions sont utilisées pour relier un CLB à n'importe quel autre.

Pour éviter que les signaux traversant les grandes lignes ne soient affaiblis, nous trouvons généralement des buffers implantés en haut et à droite de chaque matrice de commutation

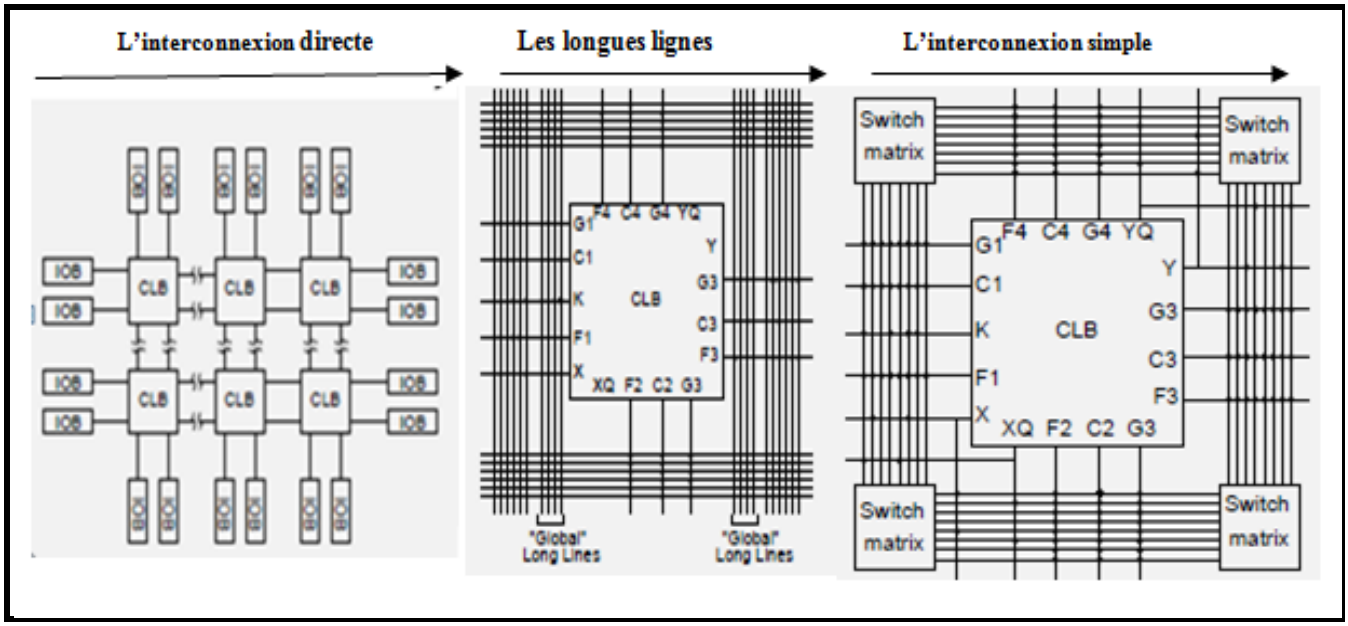


Figure I-12 : les différents 'interconnexion de FPGA

IV-3-2 Les types des FPGAs

Les deux types majeurs de systèmes de programmation pour les circuits FPGA sont les suivants:

- FPGA à SRAM ou LCA Logic Cell Array introduit en 1985 par Xilinx à base de SRAM pour configurer les connexions logique non dédiée avec des solutions d'interconnexions souples.

- FPGA à anti fusibles nés en 1990 par Actel, programmables électriquement par l'utilisateur non effaçable.

IV-3-2-1 Les FPGAs à SRAM (Static Random Access Memory)

Dans les FPGAs à type d'interconnexion SRAM, la structure d'un bloc est complexe. Les fonctions logiques sont réalisées à partir de LUT (Look Up Table) : ce sont des blocs de mémoire SRAM 16x1bit dans lesquels on peut stocker une table de transposition. La capacité d'un bloc logique est donc conditionnée par le nombre d'entrées et non par la complexité.

Les FPGAs à SRAM permettent de reconfigurer les circuits à volonté, dans des temps relativement courts. La logique de configuration peut prendre jusqu'à 20 % de la surface totale. La reprogrammation en temps réel des circuits, provoquant la reconfiguration du coprocesseur est un domaine de recherche de plus en plus important.

Les FPGAs standards à base de SRAM comme les Virtex et les Spartan de Xilinx se reconfigurent en quelques dizaines de millisecondes.

IV-3-2-1-1 Principe de l'architecteure d'un circuit FPGA-SRAM

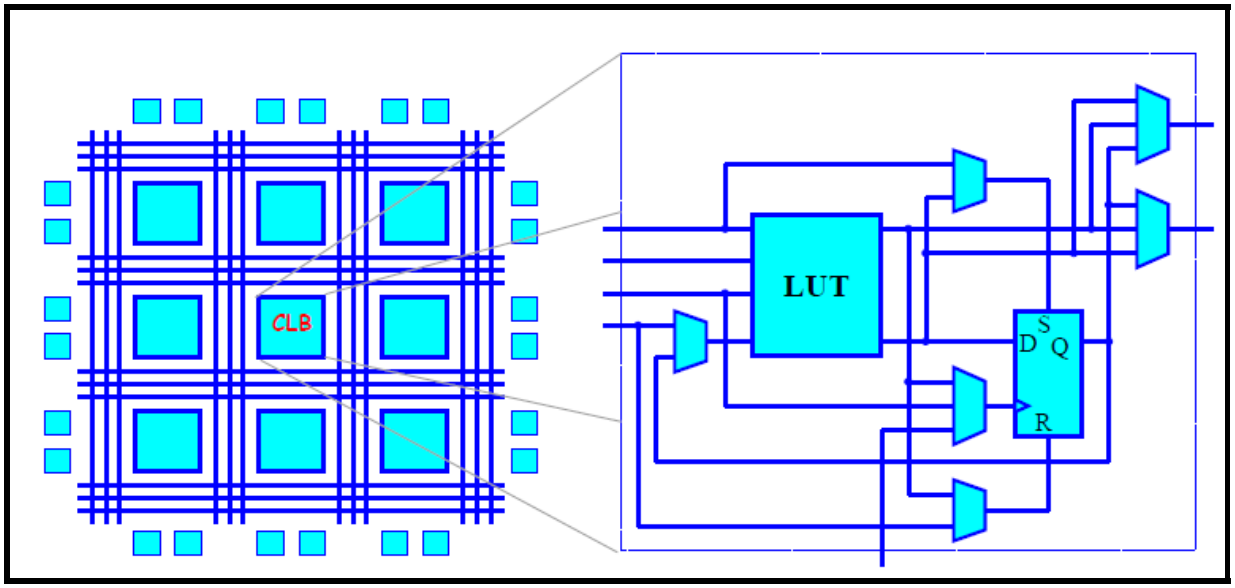


Figure I-13 : l'architecteur d'un circuit FPGA-SRAM

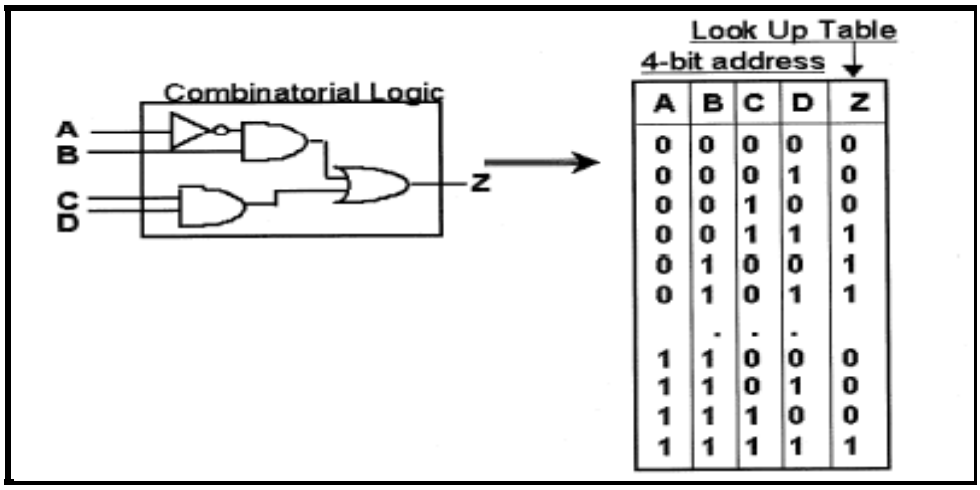


Figure I-14 : blocs de mémoire SRAM 16x1 bit

IV-3-2-2 Les FPGAs anti-fusible

Sont extrêmement petits, et leur résistance est très faible. Ils permettent d'obtenir les circuits les plus compacts et les plus rapides. Ils sont programmables une seule fois.

Le point de connexion est de type ROM, c'est-à-dire que la modification des points est irréversible. Dans l'état initial, le fusible est présent et il n'y a pas de contact.

Pour établir le contact, il faut détruire le fusible. Leur avantage est le nombre important des points de connexion à cause de la surface réduite des fusibles. Leur inconvénient est le non réversibilité de la liaison.

V) Conclusion

Dans ce chapitre, nous avons passé en revu les différentes familles de circuit logique programmable, Un état de l'art à été présenter. L'accent à été mis sur les familles les plus répondu a savoir les familles XILINX et ALTERA.

Chapitre II

Cartes et outils de développement des circuits FPGA

I Introduction

Les nouvelles générations de circuits logiques programmables (FPGA) offre l'équivalent de plusieurs dizaines de milliers de portes ainsi qu'un certain nombre de bancs mémoires. Elles ouvrent ainsi la possibilité d'intégrer un système complexe sur un unique FPGA, il a une organisation interne figée et surtout une quantité fixe de matériel. L'intégration efficace d'une application sur un FPGA doit donc tenir compte de ces contraintes à toutes les étapes de la conception, à savoir, le choix de l'algorithme, voire, leur reformulation pour obtenir une complexité compatible avec la cible technologique.

II Architecture générale des FPGAs

L'architecture d'un FPGA se décompose en deux types de ressources :

- les ressources de traitement (incluant les mémoires, la logique, les registres) regroupées en blocs logiques de différents types.
- les ressources d'interconnexions programmables qui relient les blocs logiques entre eux.

La programmation d'un circuit reconfigurable consiste donc à spécifier la fonctionnalité de chaque bloc logique et à organiser le réseau d'interconnexion afin de réaliser la fonction demandée.

II-1 Les différents éléments d'un circuit FPGA

Les éléments constitutifs d'un FPGA sont toujours à peu près les mêmes quelle que soit l'architecture choisie. Chaque fabricant ayant ses variantes par rapport à un autre.
Nous pouvons citer un certain nombre d'éléments. (Voir figure II-1)

- les éléments logiques
- Les éléments de mémorisation.
- Les éléments de routages.
- Les éléments d'entrées- sorties.
- Les éléments de contrôle et d'acheminement des horloges.

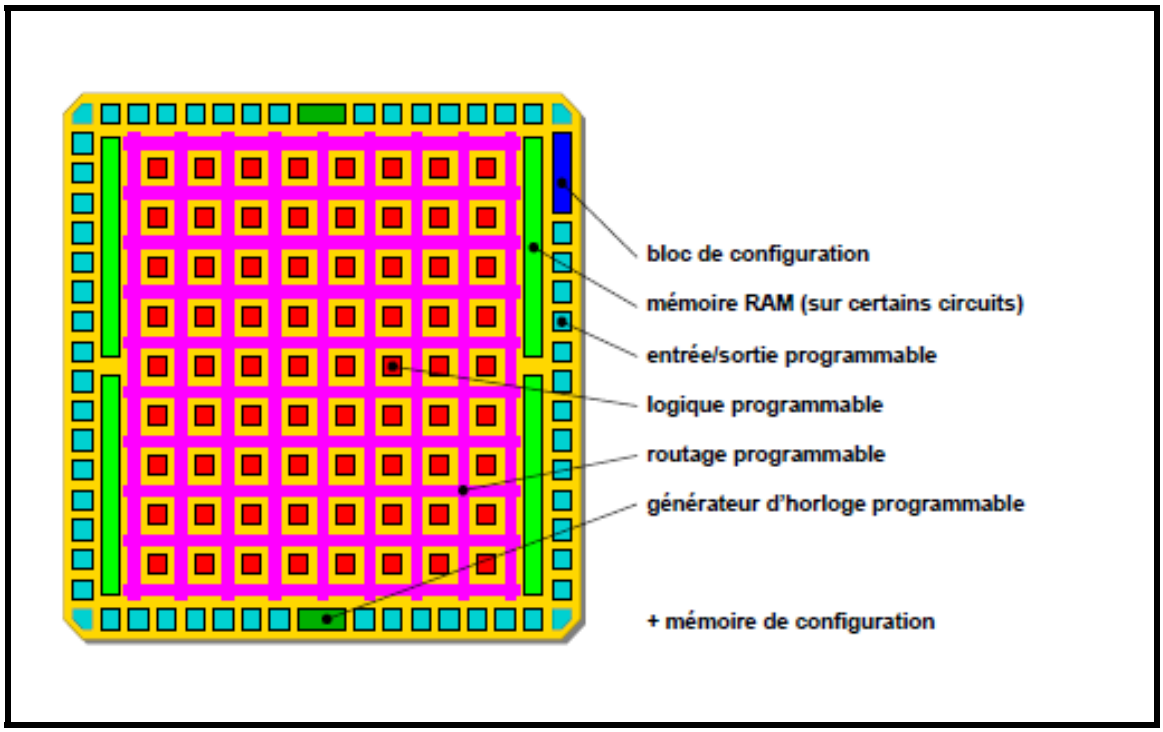


Figure II-1 : Architecture générale des FPGAs

II-1-1 Les éléments logiques

Ce sont les éléments de base de tout FPGA. Grâce à leur configuration on peut réaliser dans ces blocs toutes les opérations de logique combinatoire. Ces blocs ont souvent la même constitution et cela malgré la différence de fabricants et d'architectures. Ils sont généralement constitués d'une ou plusieurs LUTs (Look Up Table) qui contiennent, après configuration, la table de vérité de la fonction logique qu'elles doivent réaliser ou alors un ensemble de valeurs qui sont mémorisées comme dans une ROM. La taille des LUTs est généralement de 4 entrées. Les LUTs sont généralement suivis d'un registre de sortie, ce qui permet de synchroniser, si nécessaire, la sortie sur une horloge.

II-1-2 Les éléments de mémorisation

Pour des applications plus importantes, les FPGAs demandent souvent des capacités de stockage (comme en traitement d'images). La nécessité d'intégrer des blocs de mémoires directement dans l'architecture des FPGAs est vite devenue très important.

De cette façon les temps d'accès à la mémoire sont diminués puisqu'il n'est plus nécessaire de communiquer avec des éléments extérieurs au circuit.

II-1-3 Les éléments de routages

Les éléments de routages sont les plus importants dans un FPGA. En effet, les ressources de routages représentent la plus grosse partie de silicium consommée sur la puce réalisant le circuit. Ces ressources sont composées de segments (de longueurs différentes) qui permettent de relier entre eux les autres éléments via des matrices de connexions. Le routage de ces ressources est un point critique du développement d'une application sur un FPGA, ces éléments sont importants puisque ils déterminent la vitesse et la densité logique du système. Il existe plusieurs possibilités pour interconnecter deux blocs.

II-1-4 Les éléments d'entrées/sorties

Le circuit doit avoir un lien avec son environnement, c'est le but des éléments d'entrées/sorties. Ceux-ci peuvent bénéficier de protections, de buffer ou d'autres éléments permettant la gestion des entrées et des sorties. En particulier, il est à noter que les circuits actuels proposent différentes normes pour les niveaux d'entrées et de sorties qui par configuration peuvent êtres choisis afin de s'adapter à l'environnement.

II-1-5 Les éléments de contrôle et d'acheminement des horloges

Il paraît évident que dans tout système électronique relativement important, il faut disposer d'horloges et qu'elles sont souvent d'une importance capitale pour le bon fonctionnement du système. De ce fait, les FPGAs sont prévus pour recevoir une ou plusieurs horloges. Des entrées peuvent être spécialement réservées à ce type de signaux, ainsi que des ressources de routages spécialement adaptées au transport d'horloges sur de longues distances.

III Les différentes familles des FPGAs

Il ya plusieur constructeurs de composants FPGAs tels que Xilinx et Altera,Actel, Vantis,ces constructeurs utilisent différentes technologie pour la realisation de FPGAs.
parmi ces technologies, celles qui assurent une reprogrammation de FPGAs sont les plus interessantes etant donnée qu'elles permettent une grande flexibilite de conception . A l'heure actuelle, les deux monstres du secteur sont les Américains Altera et Xilinx.

Dans le cas des FPGAs développés par les sociétés Xilinx et Altera, et dans la plupart des cas, l'élément configurable de base se compose d'une LUT à 4 entrées, d'une chaîne de propagation rapide de la retenue et d'un registre de sortie afin d'assurer la synchronisation des signaux comme on le voit sur la figure II-2. Le nombre d'entrées des LUTs n'est pas dû au hasard, des études montrent qu'il s'agit là d'un bon compromis entre les performances du circuit et les contraintes des algorithmes de placement-routage.

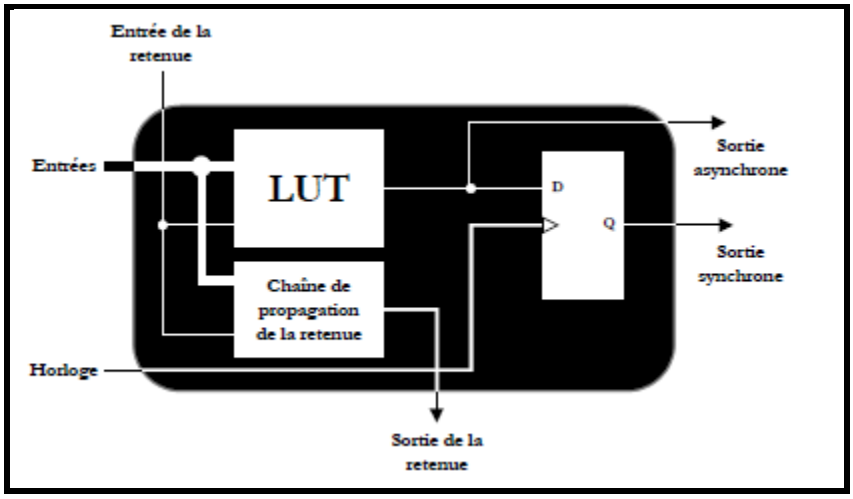


Figure II-2 : Elément configurable de base des FPGAs classique

III-1 Familles Xilinx

Xilinx est une entreprise américaine de semi-conducteurs. Inventeur du FPGA, fait Partie des plus grandes entreprises spécialisées dans le développement et la commercialisation de composants logiques programmables, et des services associés tels que les logiciels de CAO (Conception Assistée par Ordinateur) électroniques.

La première génération des FPGAs inventée par XILINX en 1985 est XC2000 elle comprend des produits d'une complexité allant de 600 à1500 portes logiques utilisables, Suivi par XC3000, XC4000, XC6000.Les nouveaux produits de Xilinx sont les VIRTEX et les SPARTAN. Ces deux composants offrent une très grande performance avec un nombre important de ressources disponibles et un grand nombre de bus de connexion avec le monde externe.

III-1-1 L'architecture d'FPGA Xilinx

L'architecture d'FPGA retenue par Xilinx se présente sous forme de deux blocs :

- Un bloc appelé circuit configurable.
- Un bloc appelé réseau mémoire SRAM

La couche dite "circuit configurable" est constituée d'une matrice de blocs logiques configurables (CLB) permettant de réaliser des fonctions combinatoires et des fonctions séquentielles. Tout autour de ces blocs logiques configurables, nous trouvons des blocs entrés/sorties (IOB) dont le rôle est de gérer les entrées-sorties réalisant l'interface avec les modules extérieurs. La programmation du circuit FPGA appelé aussi LCA (logic cells arrays) consistera par le biais de l'application d'un potentiel adéquat sur la grille de certains transistors à effet de champ à interconnecter les éléments des CLBs et des IOBs afin de réaliser les fonctions souhaitées et d'assurer la propagation des signaux. Ces potentiels sont tout simplement mémorisés dans le réseau mémoire SRAM.

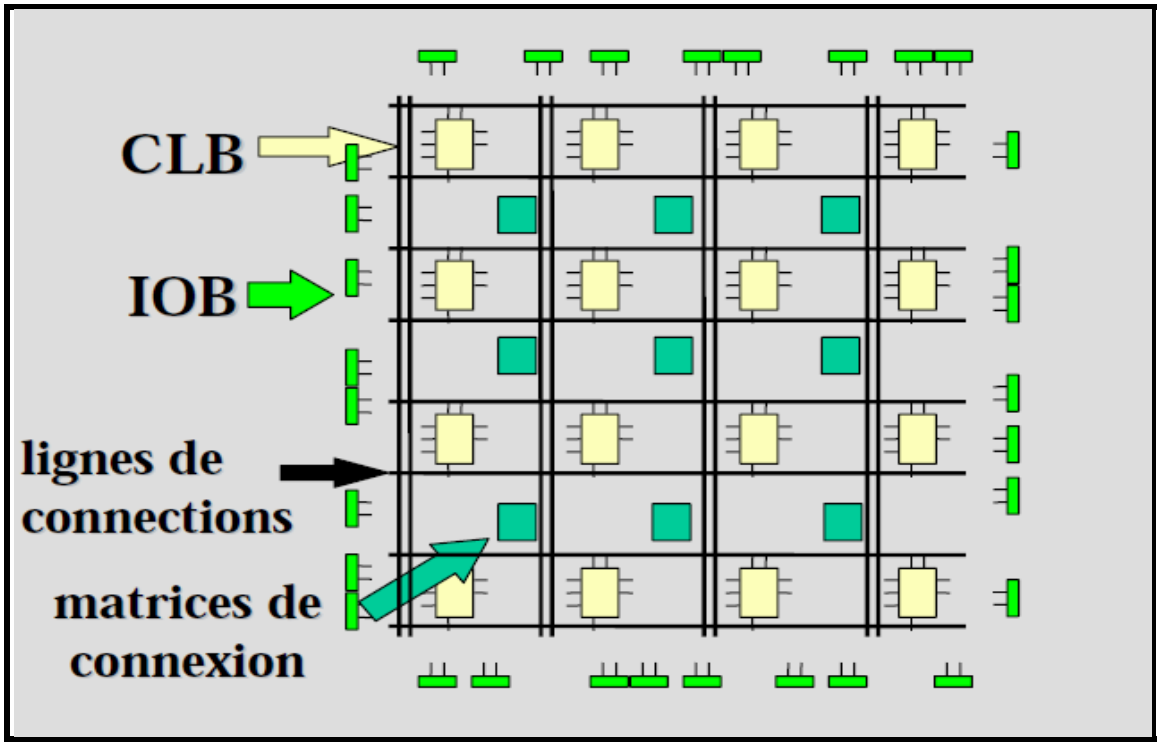


Figure II-3 : Architecture générale d'un FPGA

III-1-2 Un exemple de FPGA XILINX : La famille Virtex-II

La famille VIRTEX-II ouvre une gamme de complexité allant jusqu' à 10.000.000 portes logiques utilisables, Celle-ci fournit une architecteur régulière, flexible, et programmable de blocs logiques configurables(CLBs), entourée par un périmètre de blocs

Input /Output programmable (IOB) et possède des multiplieurs de 18 bits×18 bits, ainsi que des mémoires organisées en blocs.

La figure II-4 illustre l'architecture d'un circuit VIRTEX-II

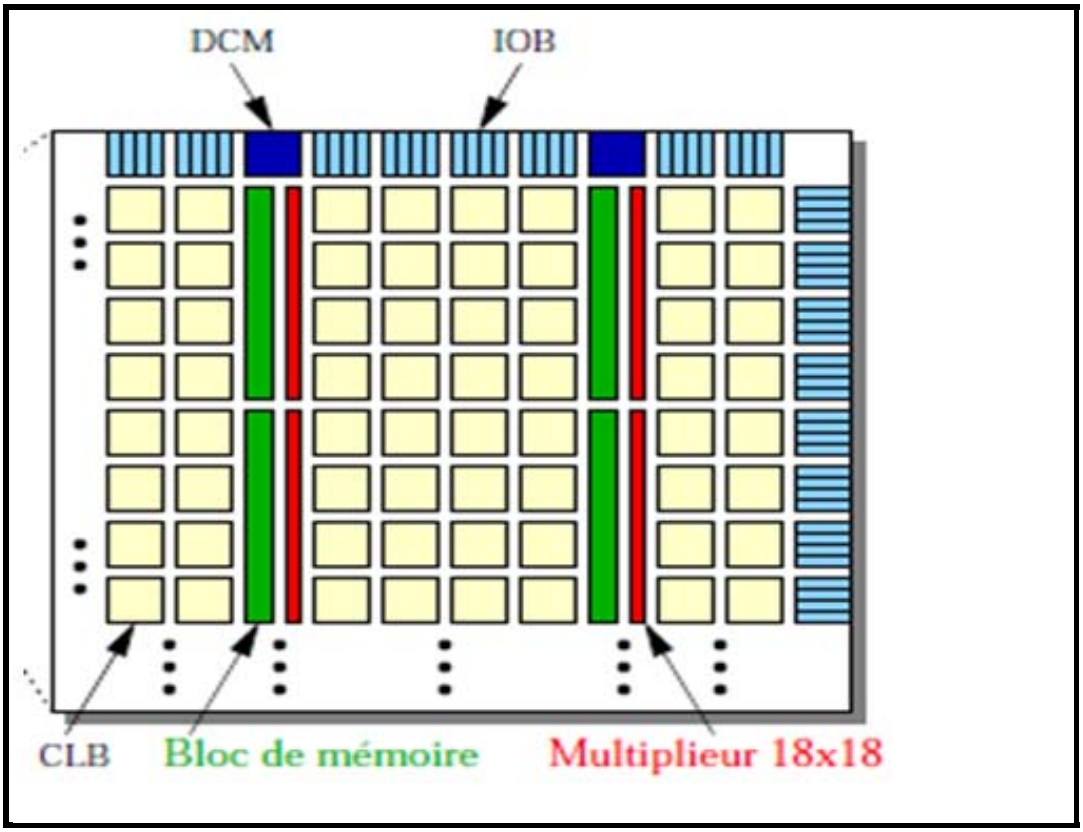


Figure II-4 : Architecture d'un circuit VIRTEX-II

Comme le montre la figure (II-4), le circuit VERTEX-II est composé de :

- blocs d'entrer /sorties (IOBs) programmable.
- blocs logiques configurables(CLBs).
- blocs mémoires SRAM (BLOCK Select RAM).
- multiplieurs de 18 bits* 18 bits.
- lignes d'horloges DCM et GCM (Globale Clock Mux).

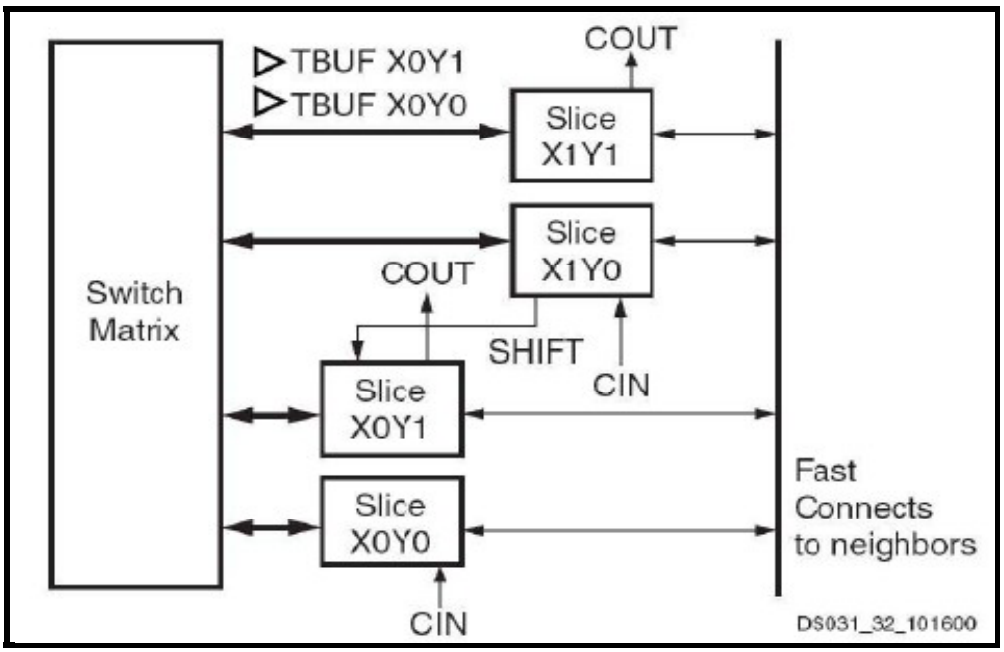
Chaque CLB est constitué de quatre cellules logiques (slices) où chaque slice contient un générateur de fonction, une logique de retenue (carry logic), des portes logiques arithmétiques, un multiplexeur et deux éléments RAM pour le stockage. Le nombre de blocs de RAM varie de 4 à 168 blocs pour le plus gros des Virtex-II.

Chaque bloc de RAM a une taille de 18Kbits avec une largeur de bus de données maximale de 36 bits (512 mots de 36 bits). Ces mémoires sont dotées de la technologie double port.

Le nombre maximum d'Entrées/Sorties utilisateur est de 1108. De plus, les composants Virtex-II possèdent aussi des multiplieurs 18x18 bits (de 4 à 168 multiplieurs), et jusqu'à 12 DCM (Digital Clock Manager) pour la gestion de l'horloge.

III-1-1-1 Architecture interne d'un circuit VIRTEX-II

III-1-1-1-1 structure d'un bloc CLB



FigureII-5 : Structure d'un bloc CLB

Chaque CLB est composé de quatre cellules logiques(Slices) réparties en deux tranches identiques de deux slices chacune avec deux retenues indépendantes et une chaine de décalage commun .Chaque Slice représenté sur la figure II-5 contient essentiellement :

Un générateur de fonction à quatre entrées : réaliser à l'aide d' une table associatives (LUT ou LOOK Up Table) de quatre entrées.les multiplexeurs F5,FX groupent les générateurs de fonctions d'une tranche d'un CLB afin de réaliser n'importe quelle fonction de cinq , six , sept ou huit variables booléennes. Chaque table permet la conception d'une mémoire synchrone 16×1 bits.

Une fois appariées, les deux LUT d'une tranche offrent une mémoire synchrone 16×2 bits, 32×1 bits ou 16×1 bits à double accès (Dual-Port Synchronous RAM).

L'élément de stockage : le circuit possède des signaux d'initialisation (set et reset) synchrones ou asynchrones.

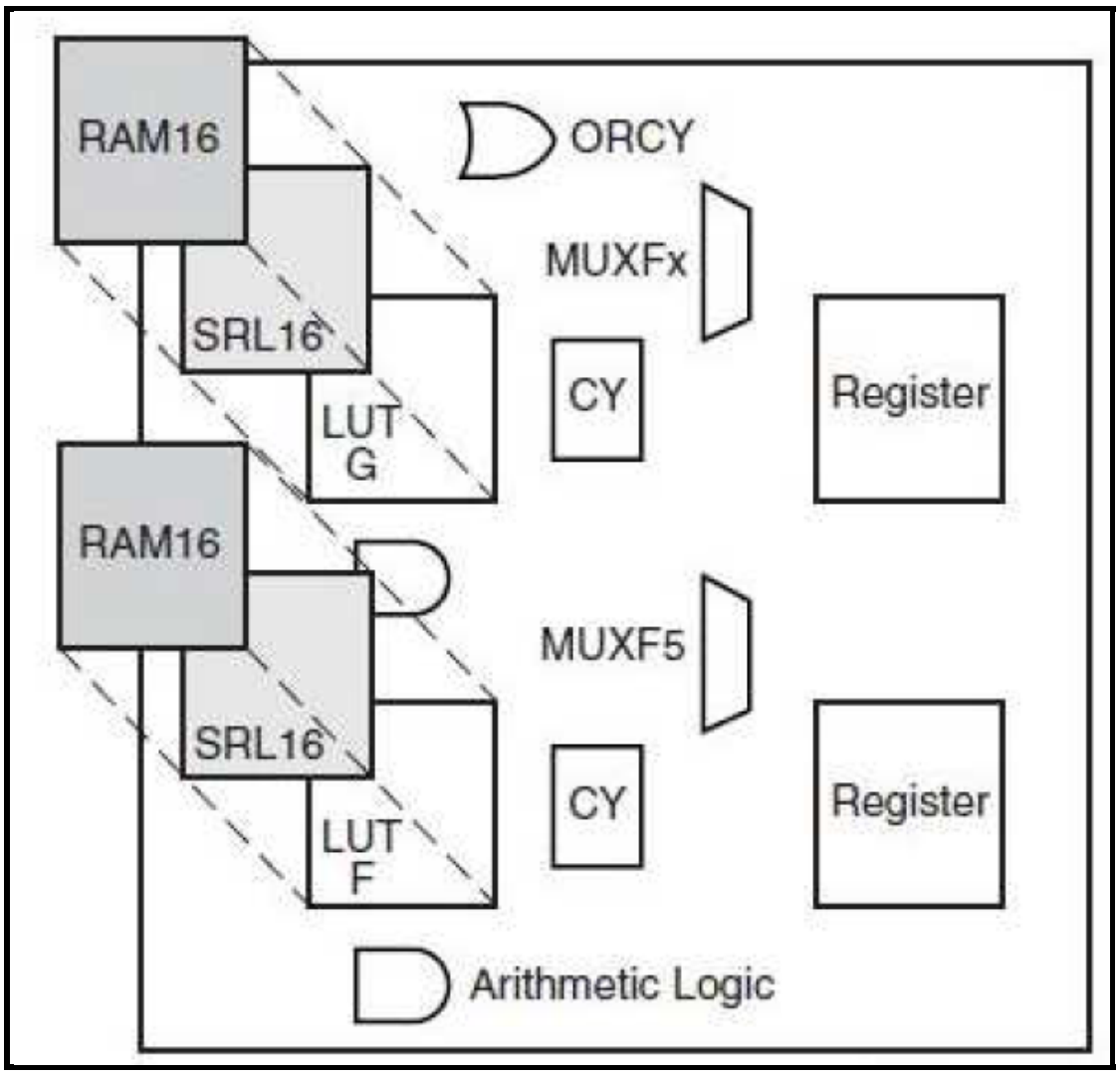
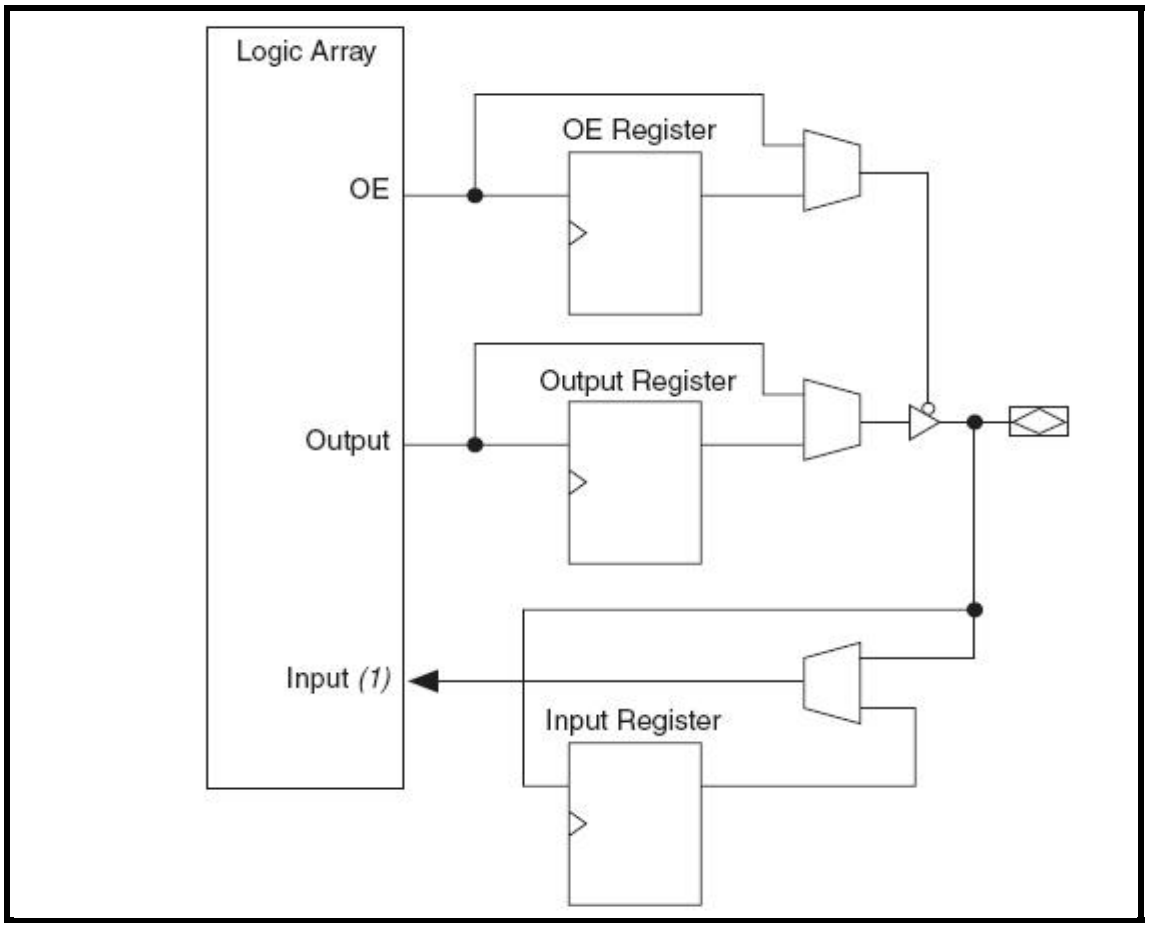


Figure II-6 : Structure d'un bloc slice

III-1-1-1-2 Les blocs d'entrée/sortie (IOB ou Input /Output Block)

Ils constituent l'interface entre les bornes du circuit et les CLBs. Nous pouvons ainsi modifier le système implanté sur le FPGA sans interférer avec l'attribution des bornes.Cette caractéristique s' avère importante si nous souhaitons développer des nouvelles versions d'un produit tout en conservant la compatibilité au niveau du boitier.



FigureII-7: Structure d'un IOB

Outre la possibilité de stocker de l'information dans les LUTs, le circuit VIRTEX-II offre des mémoires (SRAM ou Block Select RAM+), organisées en blocs, situées de part et d'autre de la matrice du CLB. Le circuit possède aussi des lignes d'horloge à l'intérieur du FPGA ou entre plusieurs circuits, ainsi que de déphaser, doubler ou diviser l'horloge.

III-1-1-1-3 structure d'un bloc mémoire (Select RAM)

Ils possèdent des signaux d'initialisation (set et reset) synchrones ou asynchrones. Le bloc Select RAM produit de large élément de stockage (18 Kbit), programmable de 16K x1bit bit à 512 x 36 bits et peut être en deux blocs RAM (dual-port RAM).

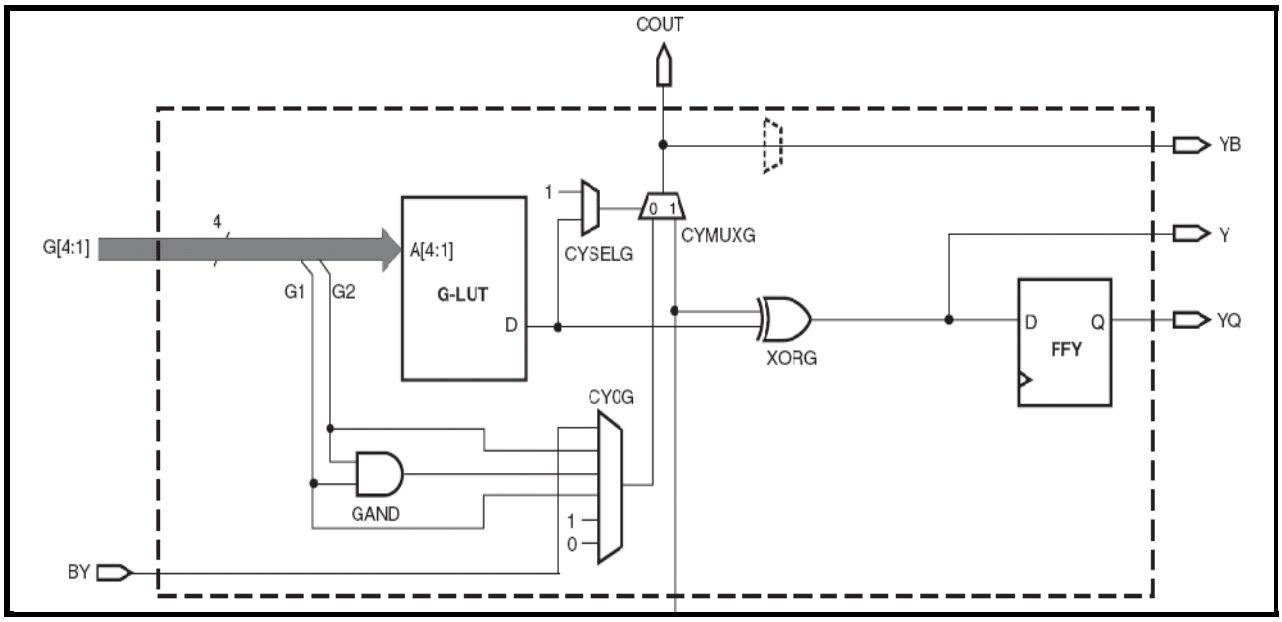


Figure II-8 : Structure d'un bloc mémoire

III-1-1-1-4 Multiplieur 18 bits × 18 bits

Le multiplieur 18 bits par 18 bits est un multiplieur à chiffre signé. Chaque entrée est sur 18 bits et la sortie est sur 36 bits. Ce multiplieur est généré par générateur de circuit VERTEX-II

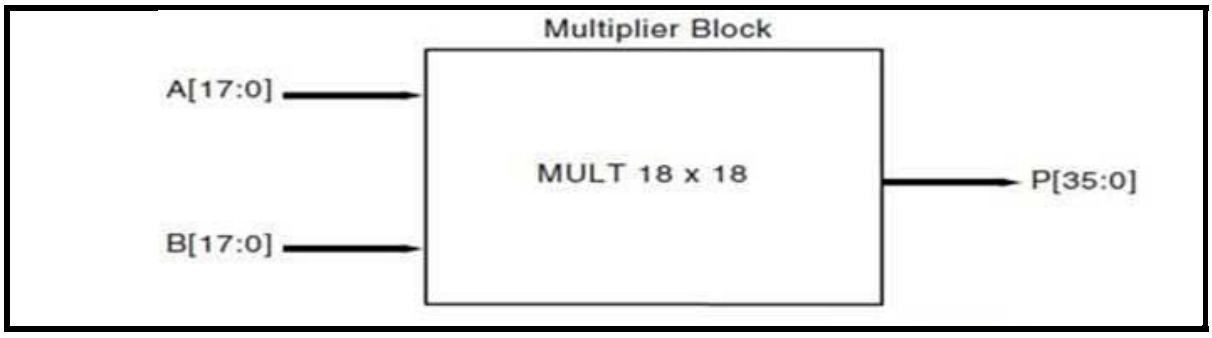


Figure II-9 : Multiplieur 18 bits × 18 bits

III-2 Familles ALTERA

La première génération des FPGAs ALTERA est Flex 8000 en 1992 Suivi par Flex 10K, Apex, Cyclone, Stratix. Les nouveaux produits de Altera sont les CYCLONE-IV en 2009 et les STRATIX –V en 2010.

III-2-1 L'architecture de FPGA retenue par ALTERA

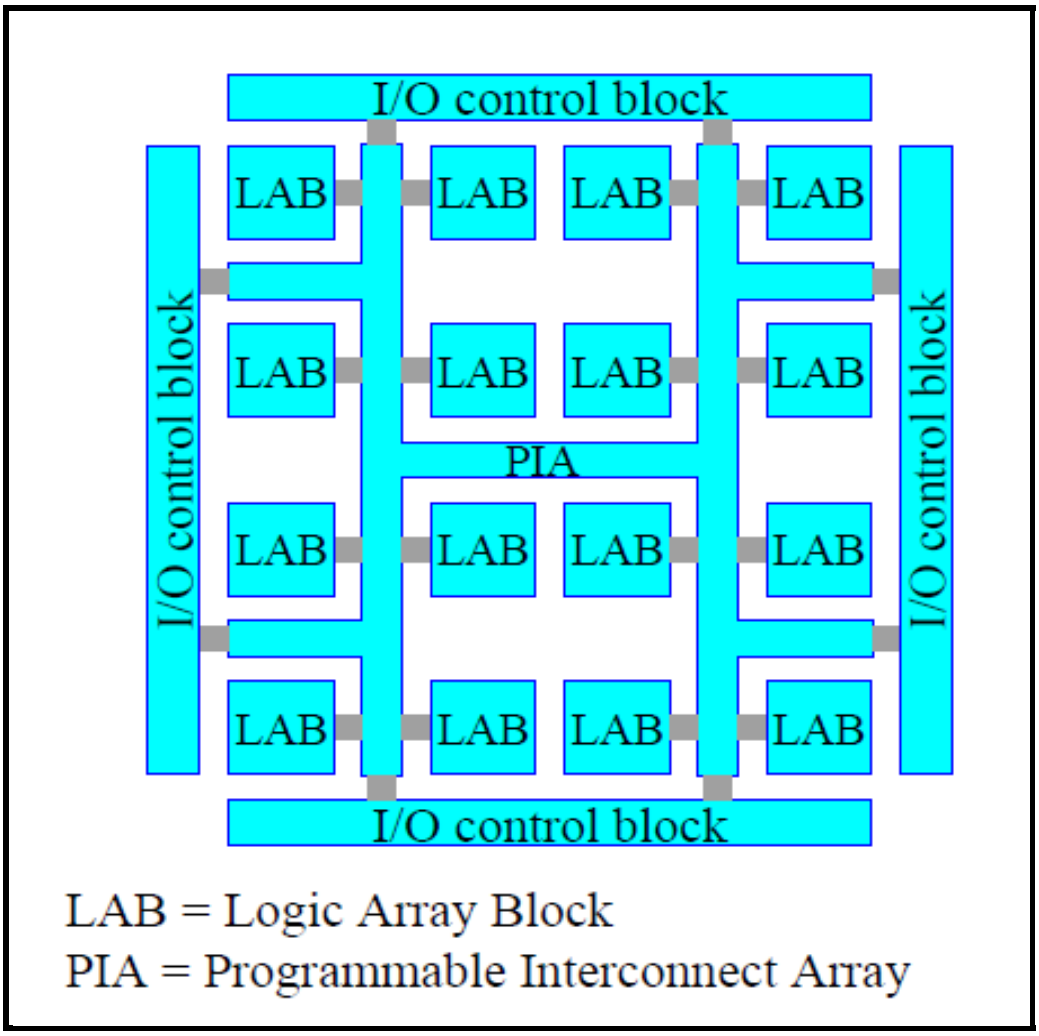


Figure II-10 : Architecture générale d'un FPGA Altera

III-2-1-1 Un exemple de FPGA Altera : Carte DE1

Notre bût est d'implanter un modulateur FSK sur la carte DE1 de la société ALTERA. Cette carte dispose d'un FPGA Cyclone II et de nombreux périphérique : différentes entrées/sorties (ports RS232, USB, Ethernet et VGA, emplacement pour une carte SD, des afficheurs 7 segments, des entrées/sorties audio, des boutons poussoirs, des LEDs…).

Dans cette partie nous présentons le circuit FPGA cyclone-II de la carte DE1 et le flot de développement que nous allons mettre en œuvre pour implanter notre modulateur.

III-3 Cible technologique et flot de développement

III-3-1 FPGA CYCLONE II (Carte DE1)

Introduite en 2004 par la société Altera elle peut avoir :

- De 4608 a 68416 éléments logiques (LE).
- Blocs de mémoire de 4kbits (4096 bits)
- Jusqu'a 150 multiplieurs dédies 18×18 Chacun est divisible en 2 multiplieurs 9× 9

Figure II-11 : Structure externe de FPGA cyclone-II

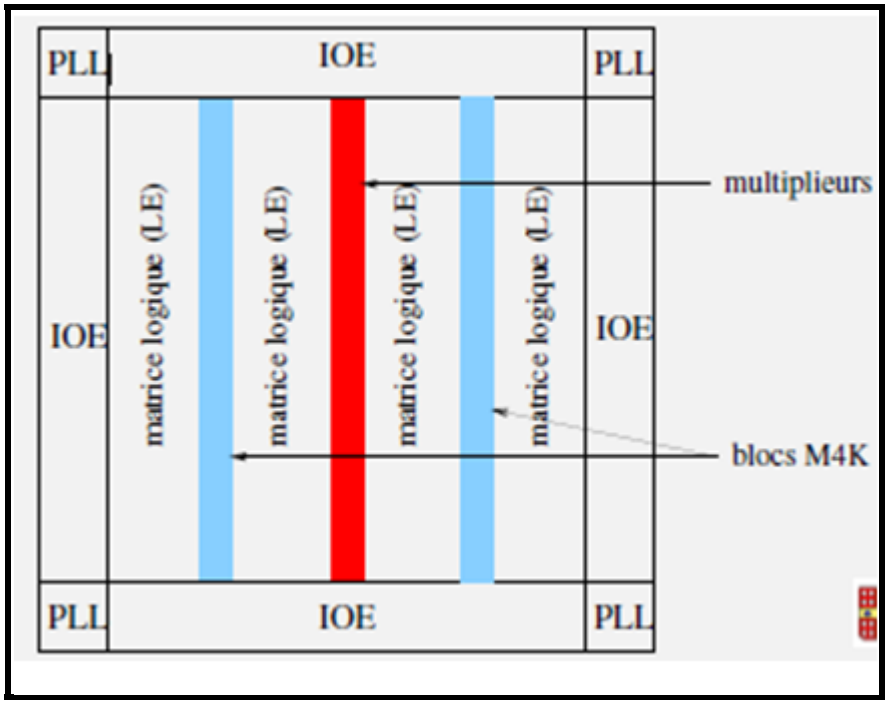


Figure II-12 : Architecture interne de FPGA cyclone-II

III-3-1-1 Les différents éléments du cyclone-II

III-3-1-1-1 Structure des LEs

La brique de base des FPGA de la société Altera est le Logic Elément (LE). Un LE est un élément configurable qui permet de réaliser des opérations de logique combinatoire et/ou séquentielle élémentaires. La Figure suivante présente l'architecture d'un LE du composant FPGA cyclone-II de la carte DE1.

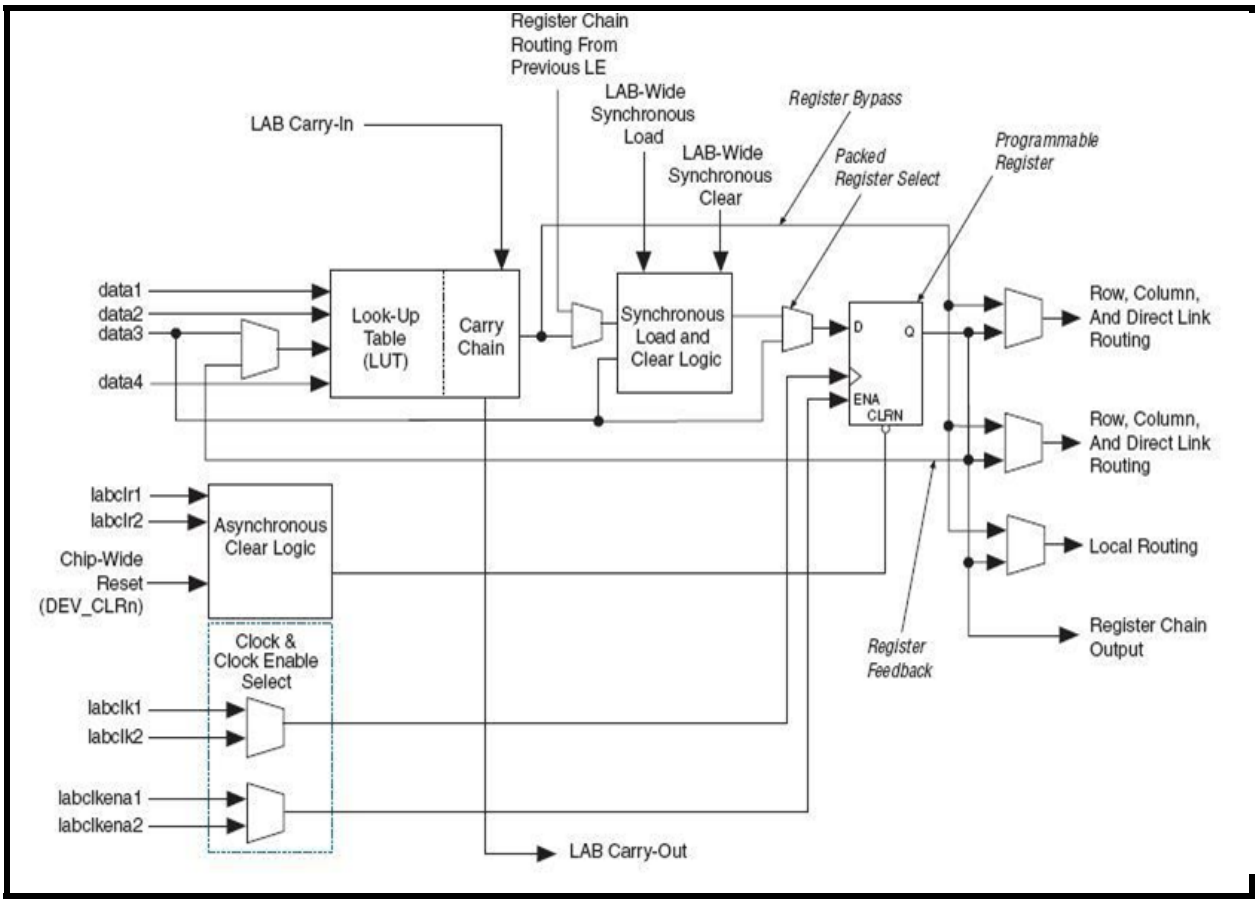


Figure II-13 : Architecture d'un LE du cyclone-II

III-1-1-1-2 Les différents éléments du LE (cyclone-II)

III-1-1-1-2-a LUT

Look **U**p **T**able **(LUT)** Permet d'implanter n'importe quelle fonction combinatoire à 4 entrées par programmation. Il s'agit d'une mémoire de 16 cases de 1 bit, l'adresse mémoire étant indiquée par les entrées.

III-1-1-1-2-b Des Multiplexeurs

C'est des opérateurs d'aiguillage. Il dispose d'entrées de données et de sélection et d'une sortie de données. L'entrée de sélection indique le numéro de l'entrée qui à diriger vers la sortie. La Figure suivante illustre le principe de fonctionnement d'un multiplexeur à 2 entrées.

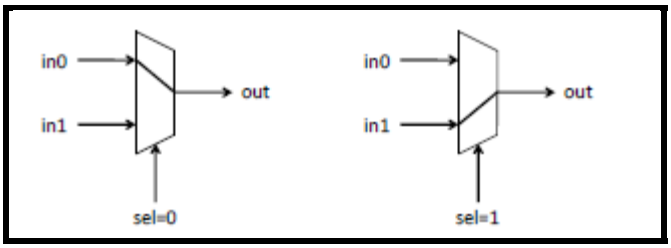


Figure II-14 : Multiplexeur à 2 entrées.

III-1-1-1-2-c Bascule D : C'est l'élément de base de toute synchronisation puisqu'elle fonctionne sur front et que sa sortie ne fait que recopier l'entrée

.

III-3-1-2 Structure de multiplieur 18×18

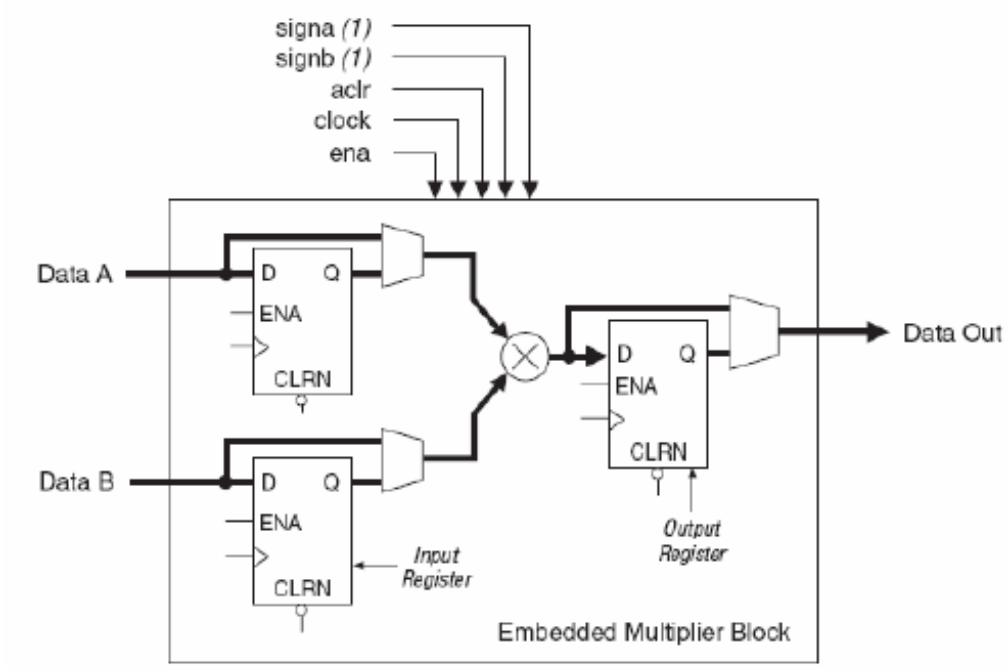


Figure II-15 : Structure de multiplieur 18×18

III-3-1-3 Structure des LABs

Les éléments logiques sont organises en groupes verticaux de16, appelés des LABs (Logic Array Block), qui disposent d'un réseau d'interconnexion configurable permettant d'interconnecter ses LE. Un autre réseau d'interconnexion configurable permet lui d'interconnecter les LE des différents LAB.

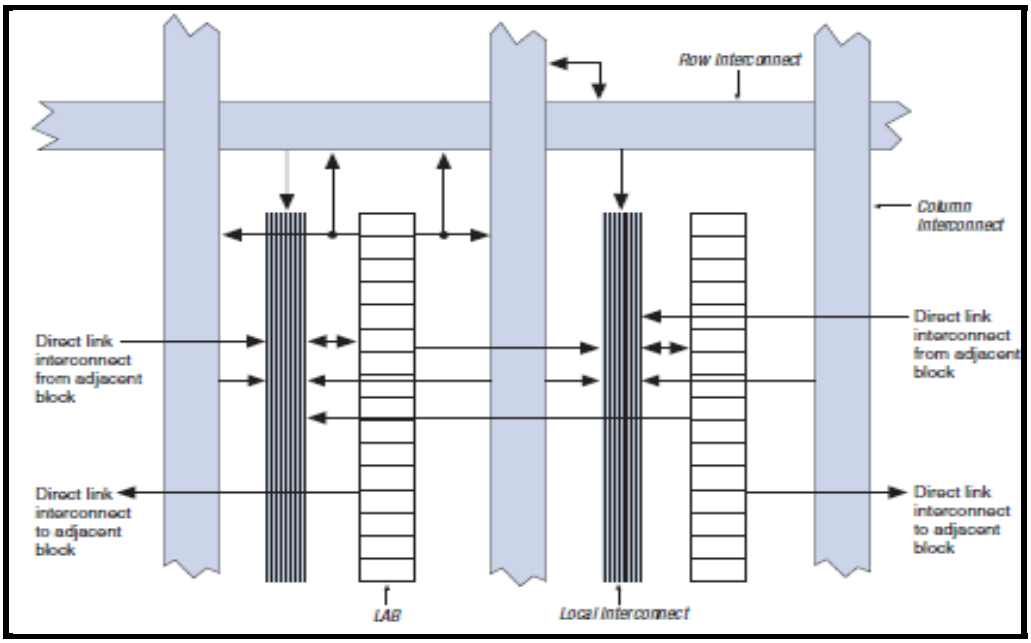


Figure II-16 : Structure des LABs

III-3-1-4 Structure des PLLs

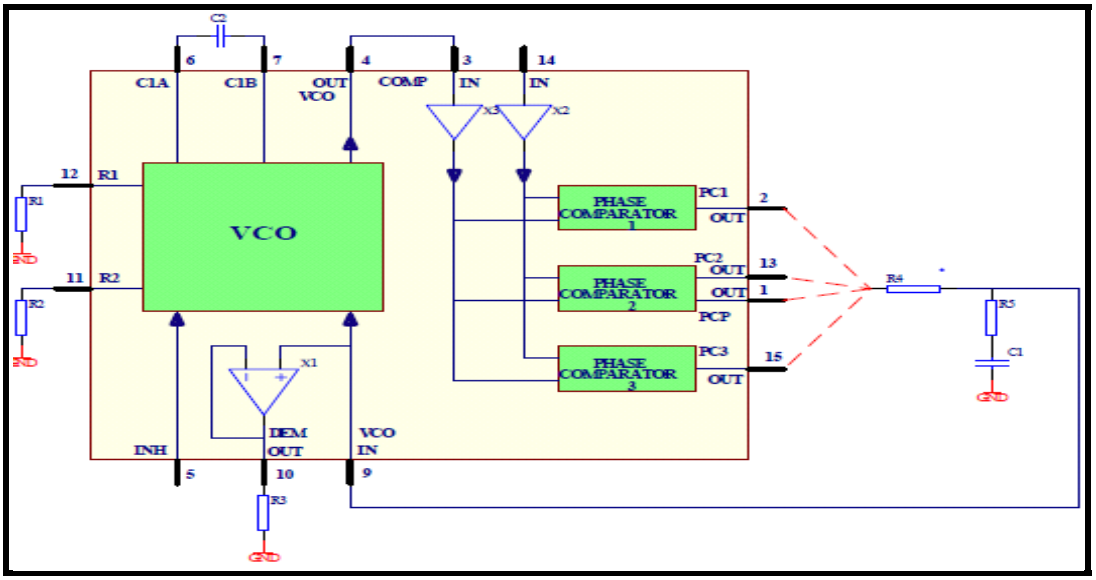


Figure II-17: Structure des PLLs

La PLL (phase locked loop) est un composent très utilisé pour les applications nécessitant des signaux stables en fréquence (modulation de fréquence, synthétiser ou resynchroniser les horloges, multiplication, division de fréquence, …..).
La boucle à verrouillage de phase se compose de plusieurs éléments simples : un oscillateur contrôlé en tension (VCO), trois comparateurs de phase, quelques suiveurs et amplificateurs pour la mise en forme des signaux et l'adaptation d'impédance.

III-3-2 Présentation de la carte DE1

Le principe de la carte de développement DE1 repose sur l'utilisation d'un circuit FPGA de la famille cyclone II 2C20.La carte de développement DE1comprend les fonctionnalités suivantes :

- Circuit FPGA cyclone II EP2C20.
- 512Kbit de SRAM 16 bits.
- 8 Mbit de SDRAM 32 bits.
- 4Mbit de mémoire Flash
- 4 afficheurs 7 segments
- un lecteur de carte SD
- 4 boutons poussoirs (KEY0 à KEY3)
- 10 interrupteurs (Sw0 à SW9)
- 10 led rouges (LEDR0 à LEDR9)
- 8 led vertes (LEDG0 à LEDG7)
- Un cordon de programmation USB Blaster
- Des oscillateurs internes RS: 50MHz, 27MHz et 24MHz
- une interface RS 232 avec connecteur 9 bronches
- une interface PS2 pour souris ou clavier
- une interface VGA avec convertisseurs numérique analogique 4bit
- Des CODEC audio (24bits) avec une entrée audio, une sortie audio et une entrée microphone (prises JACK).
- Deux connecteurs externes 40 bronches (GPIO_0 et GPIO_1)

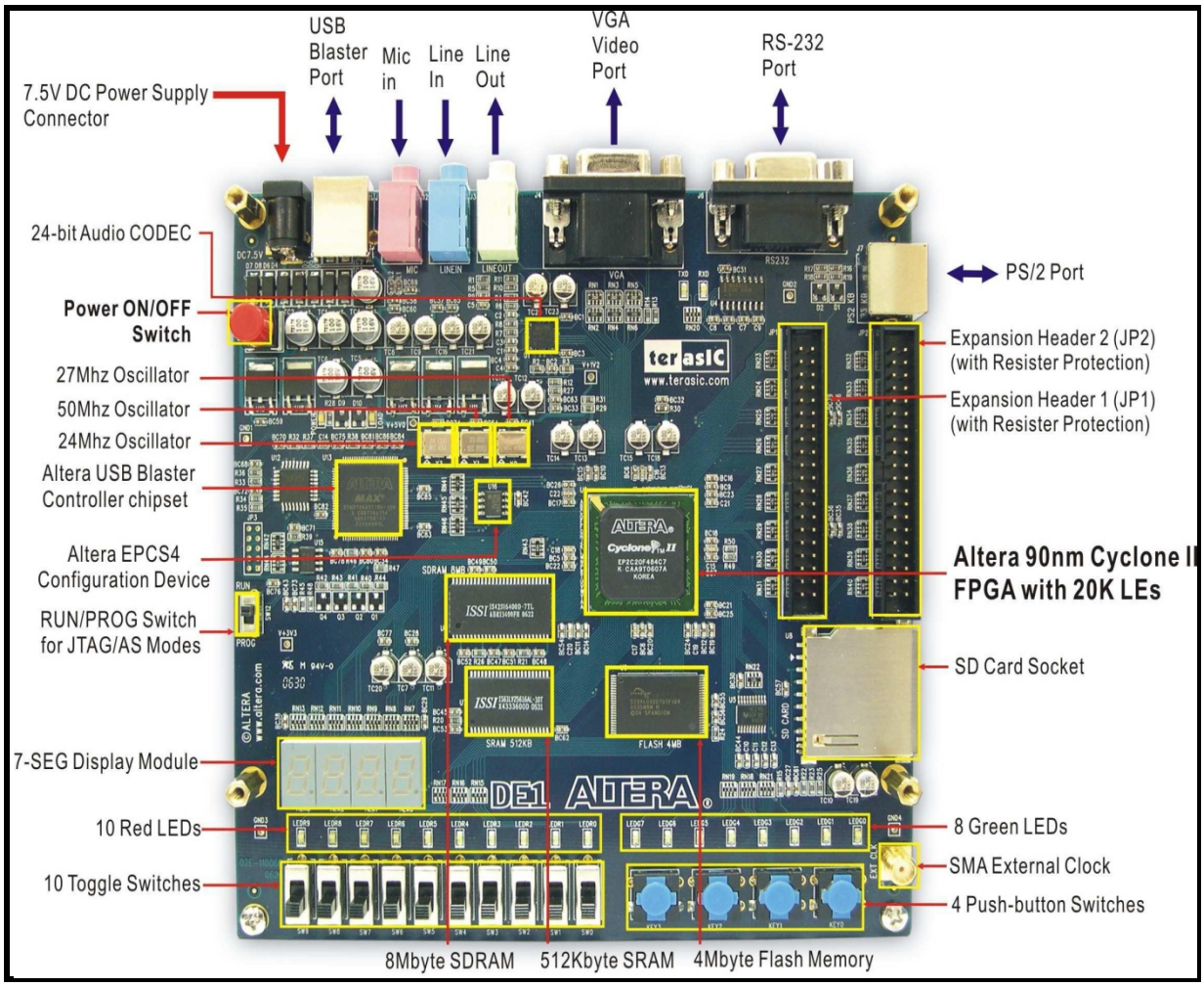


Figure II-18: Carte DE1

III-3-3 Le flot de développement QUARTUS

L'outil que nous utiliserons pour mettre en œuvre ce flot est Quartus. Il est développé par la société Altera. Permettant d'implanter des circuits numériques dans les FPGA qu'elle produit. Le flot de développement proposé dans cet outil comprend les étapes suivantes :

III-3-3- 1 Saisie schématique du circuit

Il faut d' abord commencer par décrire nous circuits en schématique. Altera fournit avec son logiciel Quartus des bibliothèques offrant de nombreuses fonctions.

La bibliothèque primitive que vous utiliserez comprend des opérateurs logiques élémentaires tels que des bascules, des portes logiques.

Quartus permet également de décrire les cricuits textuellement à l'aide de langages de description matérielle tels que VHDL, Verilog ou AHDL (langage spécifique Altera).

III-3-3-2 Conception d'un banc de test

Décrire la fonctionnalité d'un circuit n'est pas suffisant, Il faut s'assurer que son fonctionnement est correct. Pour cela il est nécessaire de simuler le circuit en lui appliquant des stimuli puis de vérifier qu'ils produisent les effets escomptés. Ces stimuli sont exprimés dans des bancs de tests (test bench) ou vecteurs de test décrits par le concepteur. Les stimuli permettant de valider le circuit seront exprimés à l'aide de chronogrammes.

III-3-3-3 Simulation fonctionnelle

Après avoir décrit le circuit en schématique et le banc de test associé, il est possible de réaliser une première simulation du circuit dite fonctionnelle. Celle-ci permet de vérifier le bon comportement du circuit sans prendre en compte les temps de propagation dans les portes logiques : la description utilisée par cette simulation est indépendante de la technologie

Si le comportement du circuit est satisfaisant, il est possible de procéder à sa synthèse. Sinon il est nécessaire de revoir sa saisie schématique.

III-3-3-4 Synthèse

L'étape de synthèse est constituée de deux sous-étapes :

la synthèse logique (logic synthesis) et la projection technologique (technology mapping).

- La synthèse logique consiste à analyser et transformer la description initiale du circuit en un assemblage minimal de ressources de logique combinatoire et séquentielle élémentaires, portes et, ou,… , bascules D,…
- La projection technologique consiste à projeter cette description sur les cellules de la technologie cible.

Une simulation peut éventuellement être faite après l'étape de synthèse.

III-3-3-5 Placement/routage

Le placement/ routage consiste à définir :

- L'emplacement sur le composant cible de chaque LE issue de la projection technologique
- le réseau d'interconnections entre les LEs et les broches d'entrées/sorties du FPGA. Le brochage du FPGA de la carte DE1 avec ses périphériques (leds, interrupteurs, . . .) étant figé, il n'est pas envisageable de laisser le routeur décider de leur routage.

Il est nécessaire de fixer la correspondance entre les broches d'entrées/sorties du FPGA et les signaux d'entrées/sorties du bloc de plus haut niveau hiérarchique, cette étape est appelée assignation. A l'issue du placement/routage, il est possible d'avoir un modèle précis du comportement temporel.

III-3-3- 6 Simulation temporelle

Une fois le placement/routage effectué, il est possible de réaliser une simulation annotée temporellement qui tient compte des temps de propagation dans les LEs et le réseau d'interconnexion. C'est la simulation la plus réaliste. Elle a toutefois l'inconvénient d'être la plus lente. Cette simulation se fait avec le même banc test que celui utilisé pour la validation fonctionnelle. Elle permet de vérifier le comportement temporel du circuit.

III-3-3-7 Programmation et vérification sur carte

Lorsque le comportement du circuit a été validé par simulation temporelle, il est possible de générer les fichiers de programmation du FPGA souvent appelés bitstream. Deux types de fichiers sont générés.

1- un fichier **.sof**, acronyme de SRAM **Object File**, qui permet de programmer la SRAM du FPGA, donc les LUT et multiplexeurs du FPGA et le réseau d'interconnexion. Ce type de programmation est volatile : Il est à refaire à chaque mise en route du FPGA.

2- un fichier **.pof**, acronyme de Programmer **Object Files**, qui est un fichier qui permet de stocker le fichier de configuration de la SRAM du FPGA dans une mémoire EEPROM.

Le contenu de cette mémoire (EEPROM). est lu à chaque démarrage de la carte pour programmer le FPGA, ce qui permet de le reprogrammer automatiquement à chaque démarrage.

Une fois les fichiers de programmation générés, il est possible de programmer la carte et vérifier le comportement du circuit dans son environnement réel. Dans notre cas, nous programmerons la carte avec le fichier sof.

IV Les étapes de création d'un projet

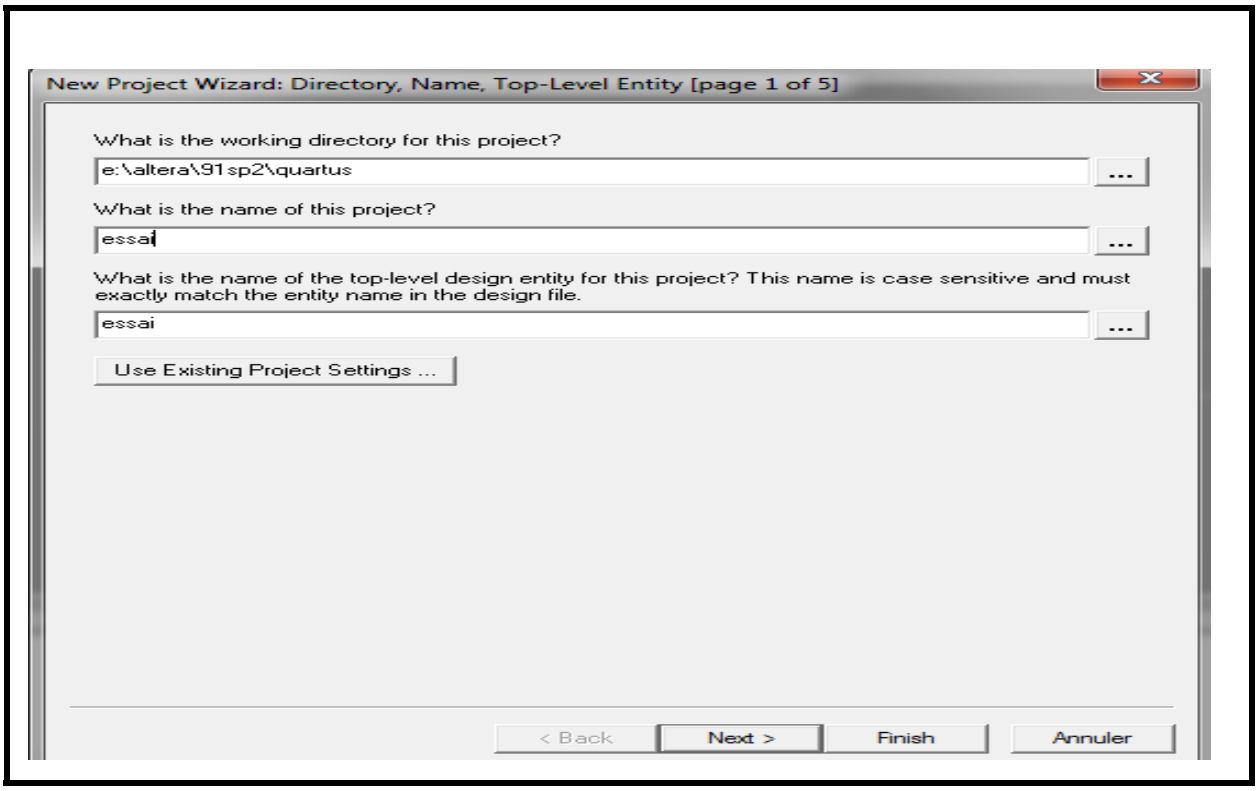
Pour le logiciel Quartus II, un projet consiste en un ensemble de fichiers de conception, de fichiers d'assignation, de fichiers de simulation, d'options de configuration et d'informations sur le projet. Le module **Création d'un Projet** nous guidera à travers les étapes qui sont nécessaires à la création de notre projet.

IV-1 Créer un nouveau projet

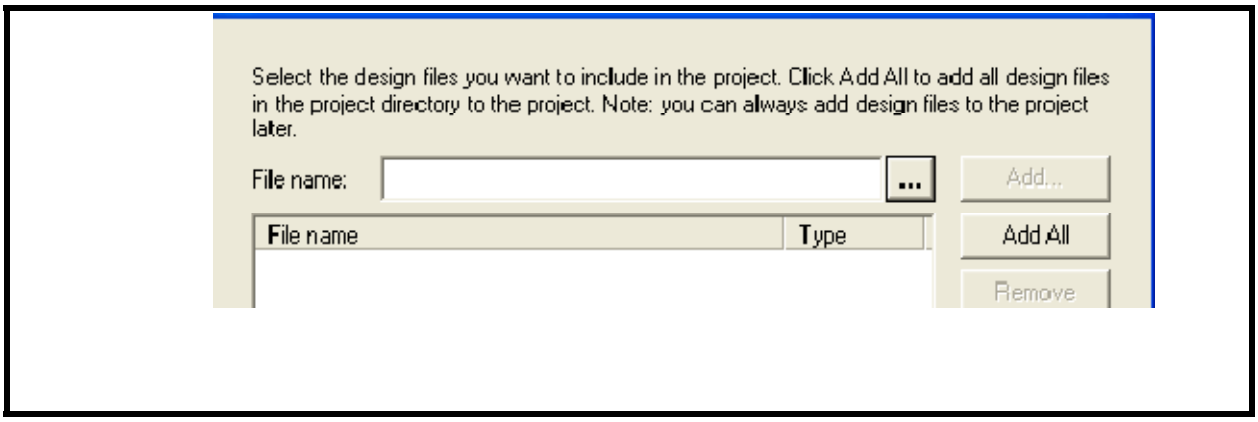
Pour créer un nouveau projet nous avons suive ces étapes :

- Sélectionner **New Project Wizard…** dans le menu **File**.
- Il se peut qu'une page d'introduction apparaisse la première fois que nous ouvrons le guide New Project Wizard; cliquer sur **Next** pour passer à la première page du guide.
- Dans la fenêtre suivante elle nous présente trois champs à remplir. Le premier demande le répertoire ou se trouvera le projet. Dans le deuxième, elle nous demande le nom du projet.

Nous pouvons choisir un nom arbitraire. Le troisième champ se remplit automatiquement alors que nous entrons le nom du projet dans le deuxième champ.

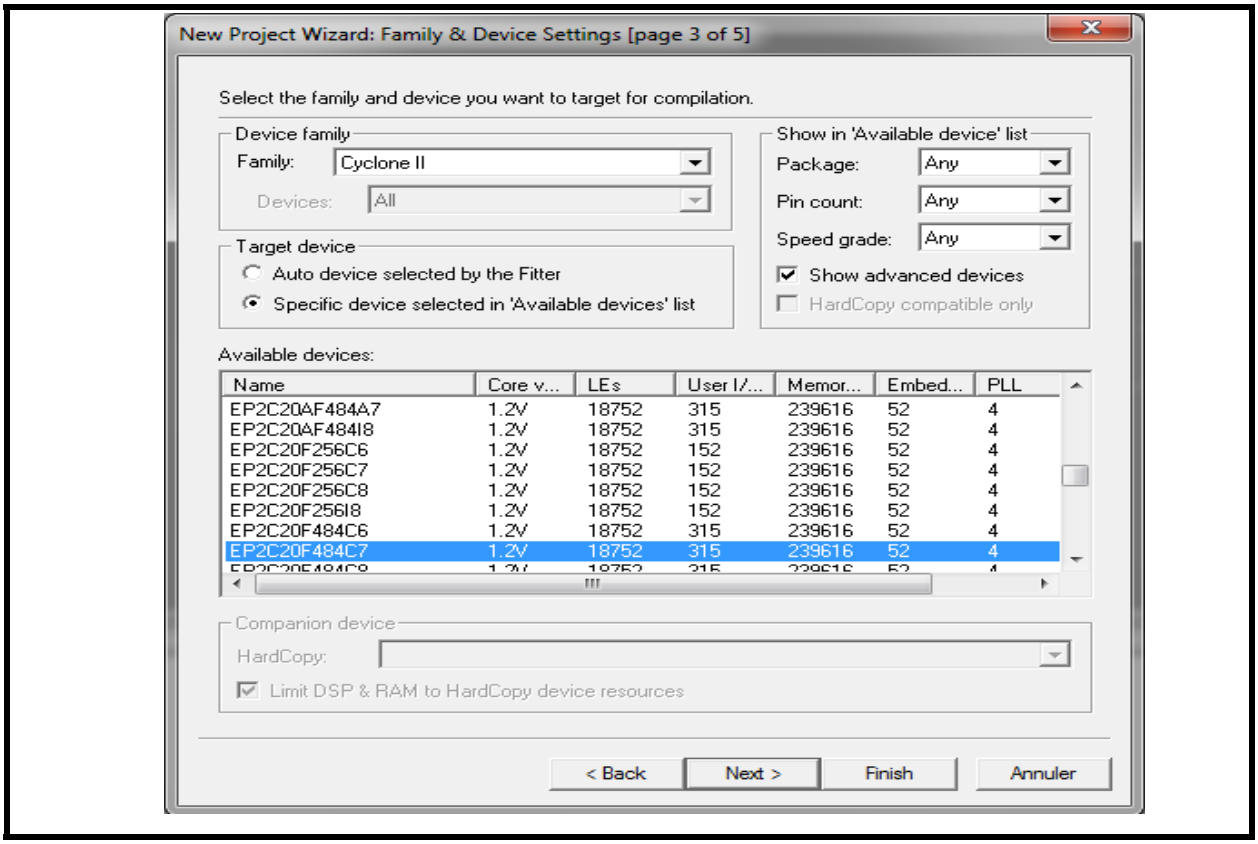


- On Clique sur **Next.**
- Dans la fenêtre suivante, on peut ensuite éventuellement ajouter des fichiers VHDL déjà écrits en les sélectionnant. C'est il n' ya pas on Clique sur **Next**.

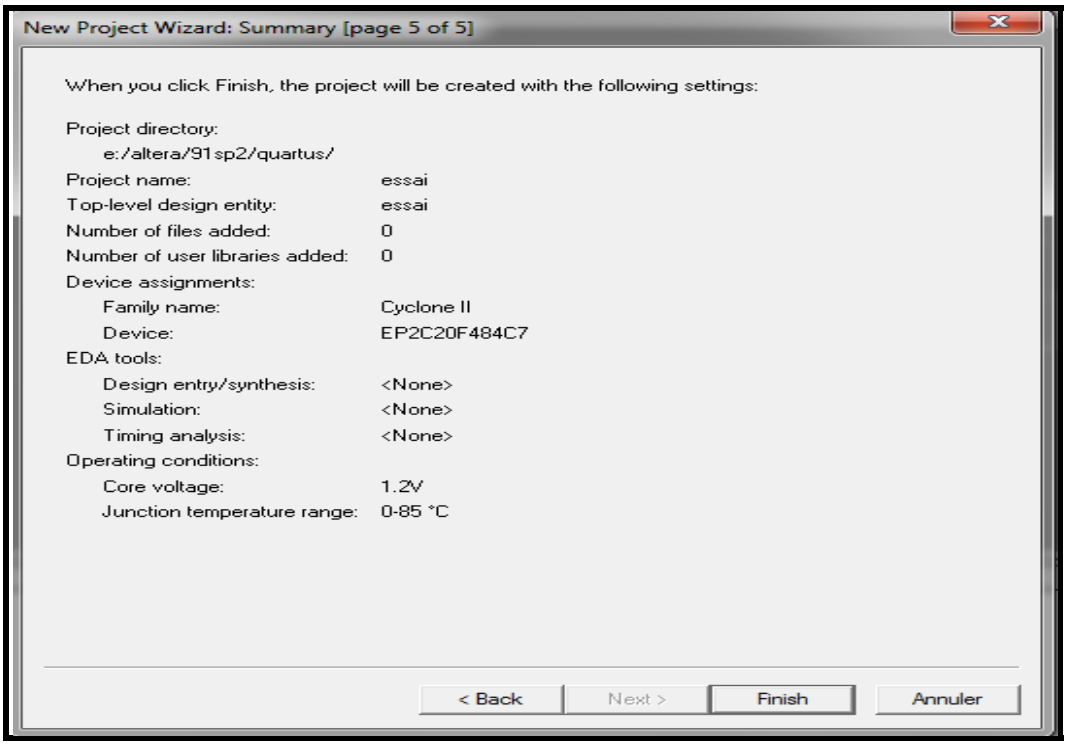


- Dans la fenêtre suivante on doit spécifier le composant dans lequel va être implémenté le circuit que nous aurons décrit. On choisit la famille puis le nom du FPGA utilisé dans la carte

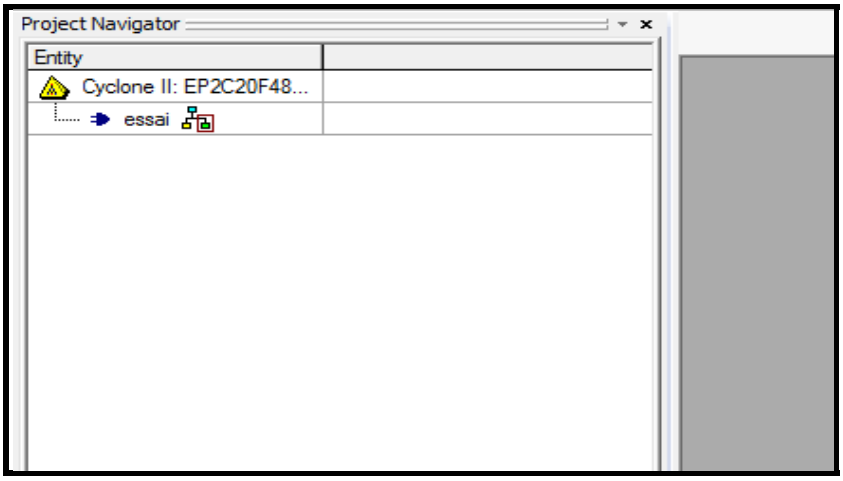
Par exemple : On choisit la famille Cyclone II puis le nom du FPGA utilisé dans la carte DE1 de Altera : EP2C20F484C7.



- On clique sur **Next**. une page **Summary** montre les options que nous avons choisies afin de configurer le projet.



- On clique sur **Finish**. Le projet est maintenant créé.



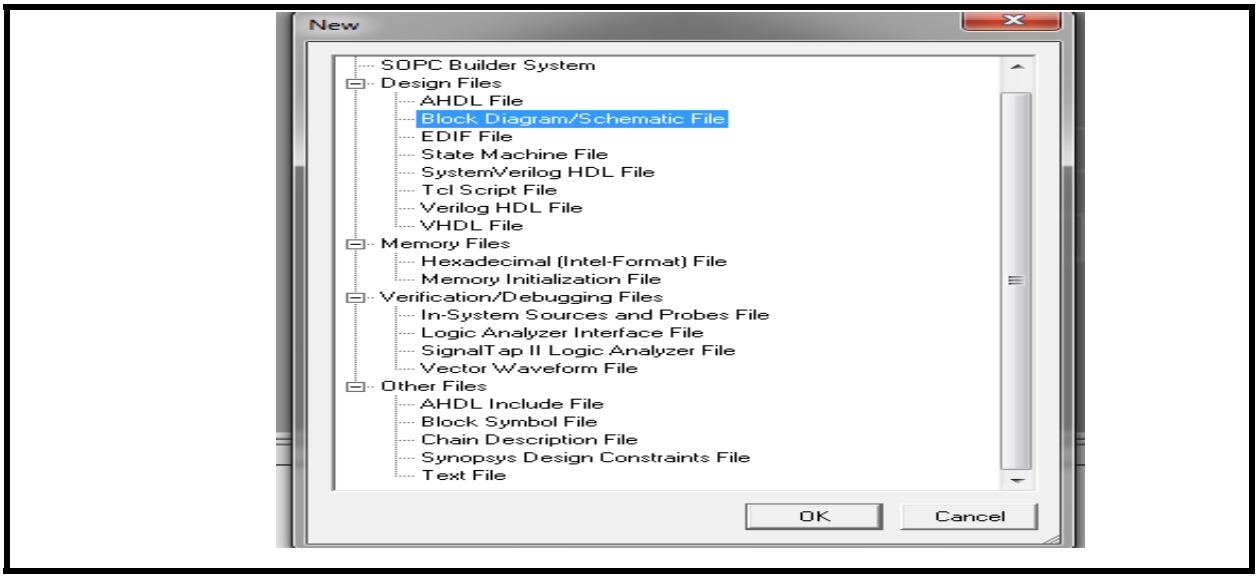
Pour créer maintenant le fichier principal du projet, il suffit d'aller dans **File** puis **New**. Plusieurs possibilités proposées par **Quartus** pour créer des fichiers.

Nous on va Choisir la description sous forme schématique.

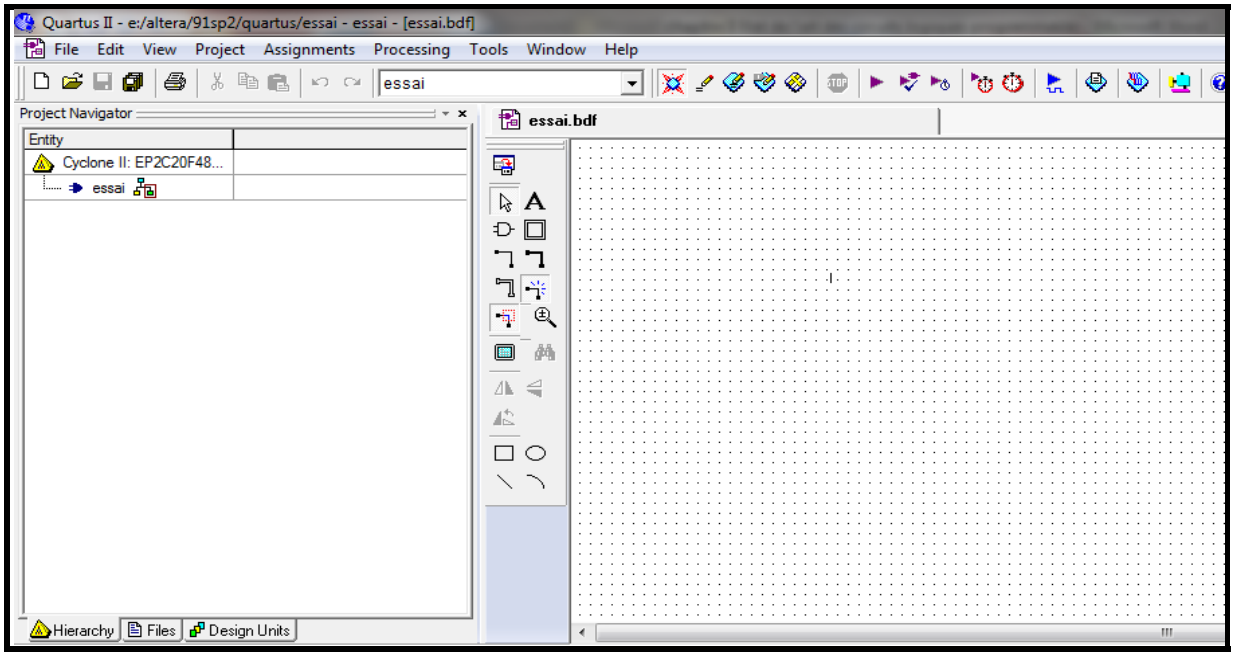
IV-2 Saisir un projet en mode graphique

- **File —>New**

La fenêtre suivante apparaît :

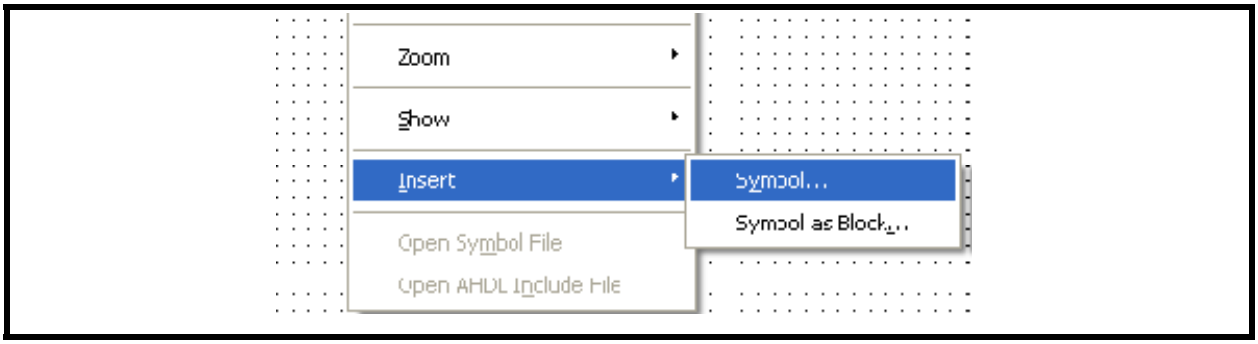


- On a choisi **Block Diagram/Schematic File** parmi la liste. Lorsque nous cliquons sur **OK,** un fichier dénommé par défaut Block1.bdf est crée. Puisque nous voulons que ce fichier soit le fichier principal, il faut le sauvegarder sous le même nom que nous avons indiqué au logiciel auparavant. donc on a fait, dans **File Save As...**, et entrez le nom du fichier principal.

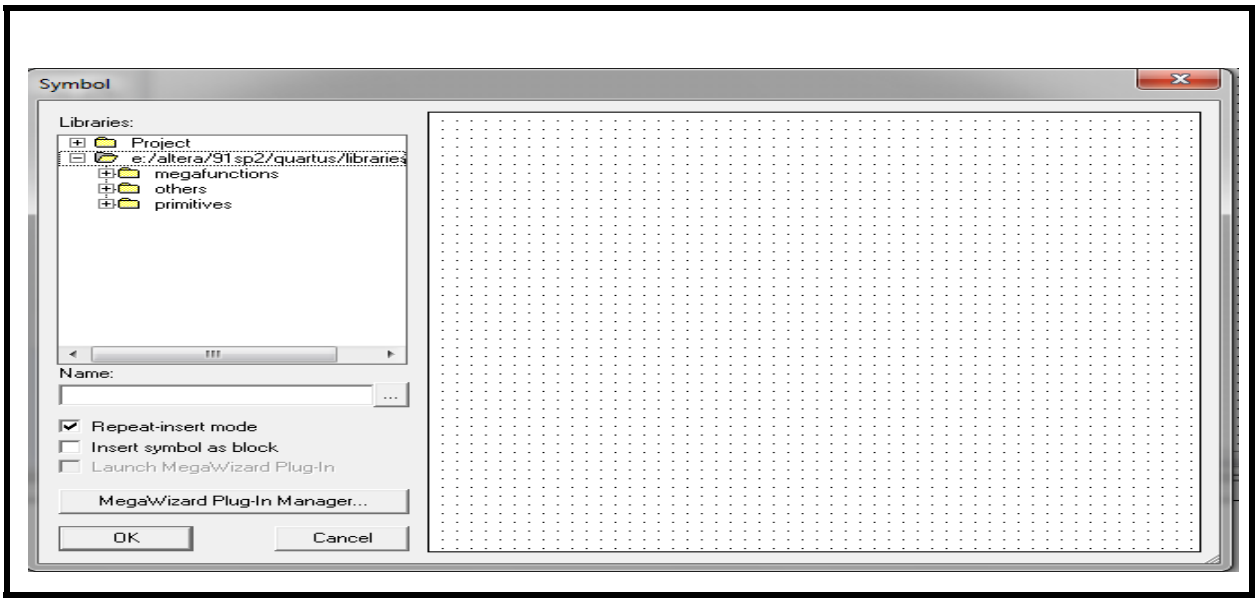


- **Insérer des symboles :**

Pour cela, nous pouvons choisir des composants de la librairie Altera, en cliquant avec le bouton droit sur la feuille et en allant dans **Insert —>Symbol**.



La fenêtre suivante s'ouvre.



Dans le champ **Name** qui apparaît, il faut entrer le nom de code du symbole dont nous avons besoin.

Exemple :

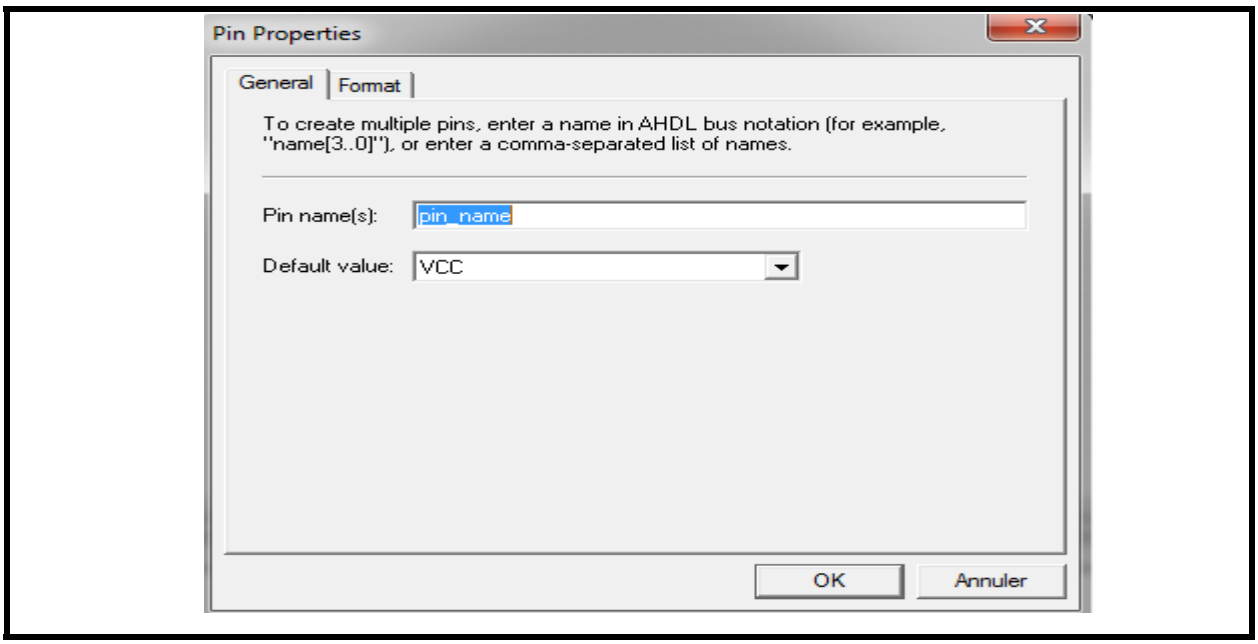
- Porte logique and : dans **Name**, mettre **and2** et cliquer sur **OK**. Placer ensuite avec un double clique, nous pouvons déplacer le symbole après facilement avec la souris.
- les entrées : Suiver la même procédure que pour le porte logique and mais cette fois-ci dans **Name**, metter **input**.
- les sorties : Suiver la même procédure que pour les entrées mais cette fois-ci dans **Name**, mettez **output.**
- **Liaison des symboles :**

Une fois tous les symboles placés, Pour tracer des connexions, on sélectionne l'outil  **(Orthogonal Node Tool)**, et on relie les deux terminaisons concernées entre les symboles.

Nous pouvons déplacer les files déjà existants, et aussi les effacer on les sélectionnant et bouton droit et on appuyant sur **Delete.**

- **Edition des noms :**

Il faut maintenant affecter un nom logique, double clique sur chaque symbole, La fenêtre qui apparaît contient un champ **Pin Name**, on donne le nom pour le symbole.



Une fois le schéma est fin on le sauvegarde et on passe a la troisième étape.

IV-3 Compilation

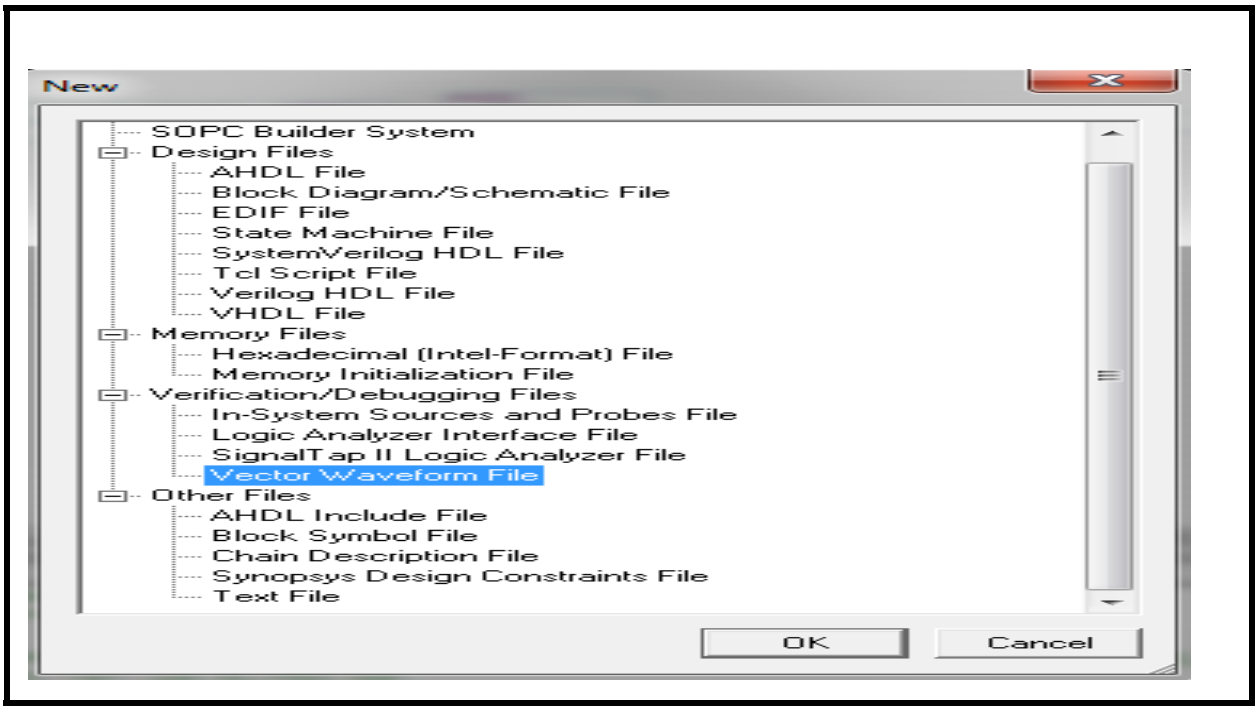
Le Compilateur de Quartus II est constitué d'une série de modules qui vérifient s'il n'y a pas d'erreurs

- On utilise le menu **Processing → Start compilatio**
- Ou bien on clique sur le bouton  dans la barre des icônes**.**

Si nous recevons un message indiquant que la compilation est réussie et qu'il n'y a pas eu d'erreurs, on clique sur **OK** sinon on corriger notre projet et on recommencer la compilation.

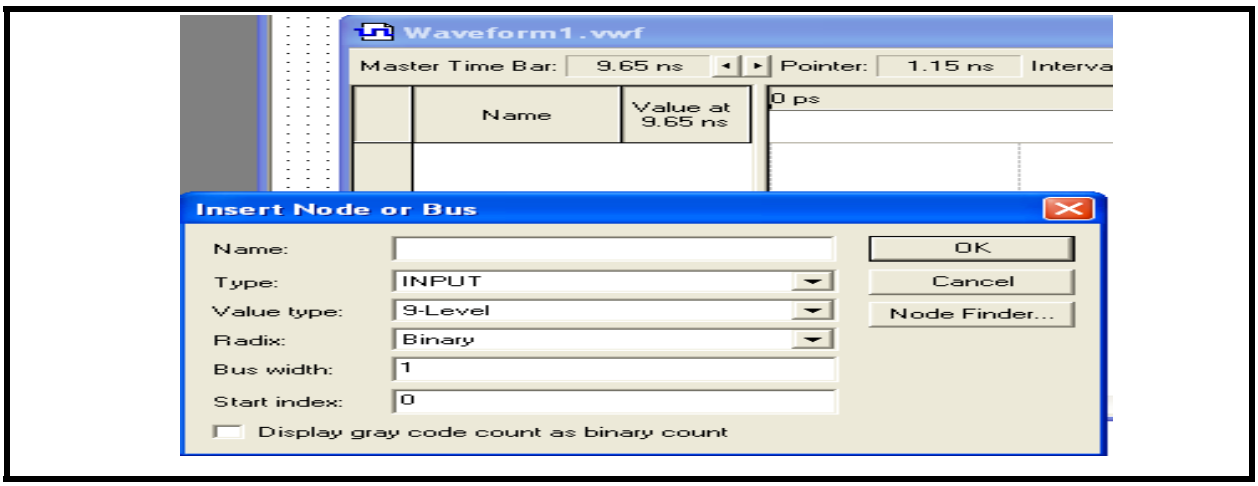
IV-4 Simulation du projet

- **Création d'un chronogramme** : Dans **File**, on choisi **New** puis, dans **Other files** on choisi **Vector waveform file**, puits **OK**



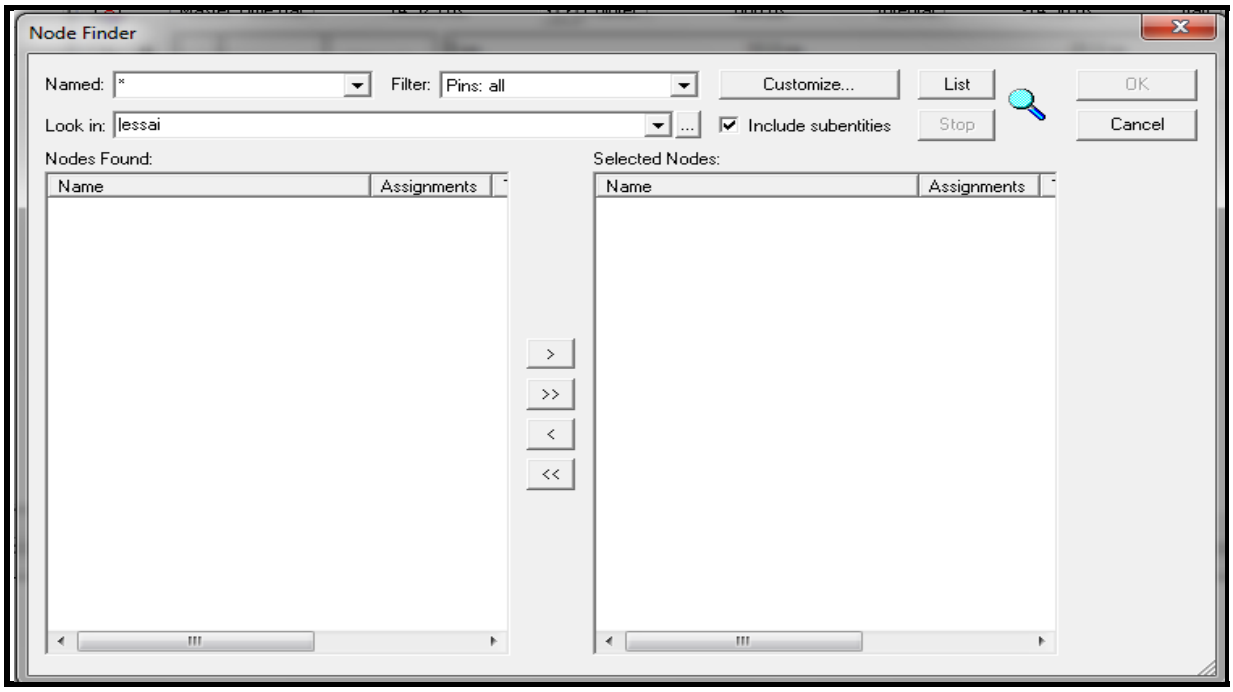
On commence par sauvegarder ce fichier et on choisi le nom proposé par défaut (le nom de notre fichier principal suivi de .wmf).

- **Ajout de signaux dans un chronogramme** : Dans la colonne **Name**, on clique avec le bouton droit Puis, on clique sur **Insert Node or Bus**



On clique sur **Node Finder** qui sert simplement à sélectionner des groupes de pins.

La fenêtre suivante s'ouvre.



Pour sélectionner toutes les pins, on laisse le symbole **"*"** dans le champ **Named**, et dans **Filter** on choisi **Pins: all**

On clique ensuite sur **List,** les pins trouvés dans notre projet sont affichées dans la liste à gauche.

On clique sur le symbole **"»"** pour envoyer la totalité des pins listées vers la colonne de droite, qui correspond aux pins sélectionnées. On clique enfin sur **OK.**

A notre retour dans la fenêtre **Insert Node or Bus**, on clique sur **OK**. Les noms des pins apparaissent maintenant sur le fichier .vwf.

Dans **Edit**, nous trouvons **Grid size** qui permet la résolution de la grille de temps et **End time** qui permet le temps de la fin de la simulation nous pouvons donnée les valeurs pour avoir une plage dons laquelle on travaille.

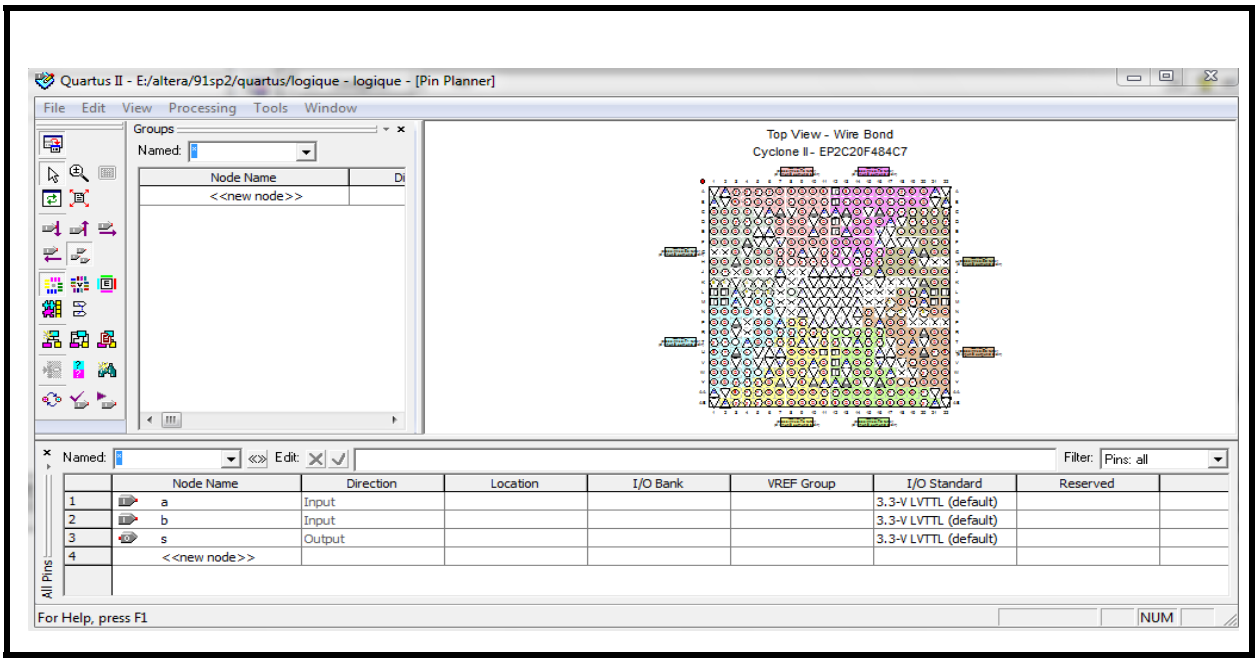
Il est important de savoir zoomer et dézoomer cette fenêtre, afin de voir toute la plage qui nous avons choisie.

Et pour lancer la simulation on clique sur **Processing / Start Simulation.**

On choisi **Functional ou Timing** selon le type de simulation souhaité.

IV-5 Définition du brochage du composant

Nous avons ouvert la fenêtre d'assignement avec le menu **Assignements→Pins**



On clique une fois sur la case **Location** correspondante à la première ligne de la liste, ensuite, nous pouvons remplir les cases très simplement en utilisant le clavier. Déplacer sur n'importe quelle case. On fait cela pour chaque pin. Une fois que tous les pins sont assignés, compiler et simuler le projet.

IV-6 Programmation du FPGA

Allons maintenant dans le menu de programmation **Programmer**

Le fichier **(nom de fichier.sof)** devrait être chargé. Nous Lançons la programmation du FPGA avec le bouton **Star**. Lorsque la diode **good** (sur la carte) se réactive, nous pouvons utiliser le circuit.

Manipuler et vérifier le bon fonctionnement de circuit.

V Conclusion

Ce chapitre à été dédié a la présentation détaillé des familles de circuit FPGA(XILINX ,ALTERA).Le flot de développement de la carte DE1 supportant un FPGA type cyclone II à été donné ,présenter et commenter.

Chapitre III

Implémentation d'un Modulateur FSK sur FPGA

I Introduction aux Techniques de modulation

I-1 Introduction

Les systèmes de transmission numérique véhiculent 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 l'information de la source vers le destinataire avec le plus de fiabilité possible.

Au cours du développement des dispositifs de télécommunication, il est rapidement apparu indispensable de coder l'information à transmettre, soit pour adapter l'information au canal de transmission (fibre optique, câble coaxial, faisceaux hertziens), soit pour transmettre simultanément plusieurs signaux informatifs sur un seul et même canal. De ce fait, le codage de l'information s'est révélé être un point-clef qui fait aujourd'hui encore l'objet de recherches et de normalisation. Il est donc nécessaire pour transmettre un signal de l'adapter au canal de transmission, pour cela deux solutions sont disponibles:

- La transmission en bande de base (il s'agit de l'adaptation par codage)
- La modulation (adaptation correspond alors à la transmission sur une fréquence porteuse).

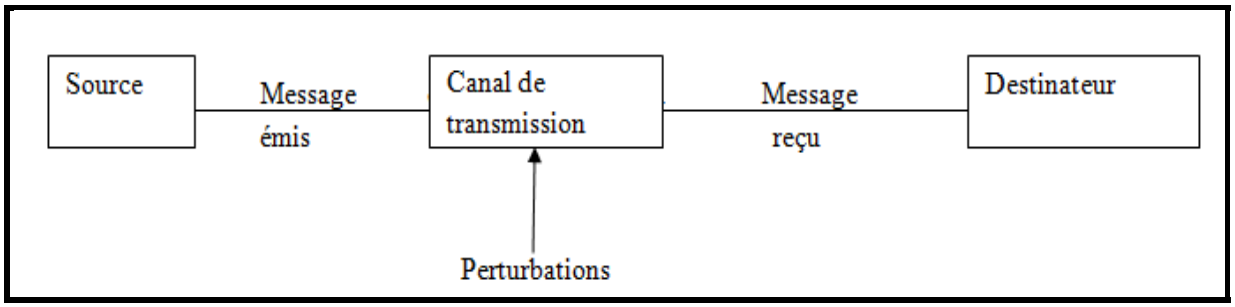


Figure III-1 : Transmission de l'information dans un canal

Dans ce chapitre nous allons nous intéresser à la deuxième solution, qui est la modulation.

I-2 Définition de la modulation

L'une des formes de codage d'information les plus simples et de transmettre à distance un signal basse fréquence porteur d'information (fréquences f : BF) on utilise une onde porteuse sinusoïdale de haute fréquence (fréquence f0: HF). La modulation est la transformation d'un message à transmettre en signal adapté à la transmission sur un support physique. Le signal porteur d'une information il s'appelle signal primaire ou modulant et autre signal appelé signal secondaire ou modulé.

Le but de la modulation est d'adapter le signal à transmettre au canal de communication. On introduit donc deux opérations supplémentaires à celle de la figure III-1; entre la source et le canal, une première opération appelée modulation et entre le canal et le destinataire, une seconde opération appelée démodulation. La chaîne de transmission globale est alors celle de la figure III-2

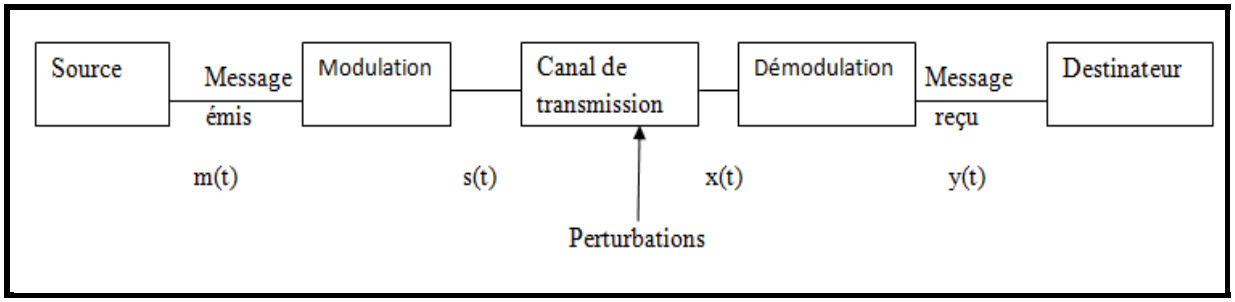


Figure III-2 : Chaîne globale de transmission

I-3 Les effets de la modulation

- transposition dans un domaine de fréquences adapté au support de transmission
- meilleure protection du signal contre le bruit
- transmission simultanée de messages dans des bandes de fréquences adjacentes, meilleure utilisation du support

I-4 Les techniques de modulation

Il existe deux types majeurs de modulation :

I-4-1 La modulation analogique

Un signal est principalement caractérisé par son amplitude, par sa phase et sa fréquence. La modulation consiste à faire varier une de ses trois caractéristiques de façon à ce quelles dépendent à chaque instant de la forme du signal modulant. Selon le paramètre du signal modulé, il y a modulation d'amplitude (AM), modulation de fréquence (FM) ou modulation de phase (PM).On appelle modulation analogique, la modulation d'une porteuse par un signal utile analogique.

I-4-1-1 Le signal analogique

L'amplitude du signal varie de façon continue, elle peut prendre une valeur quelconque (donc une infinité de valeurs), avec une durée infiniment petite entre 2 instants.

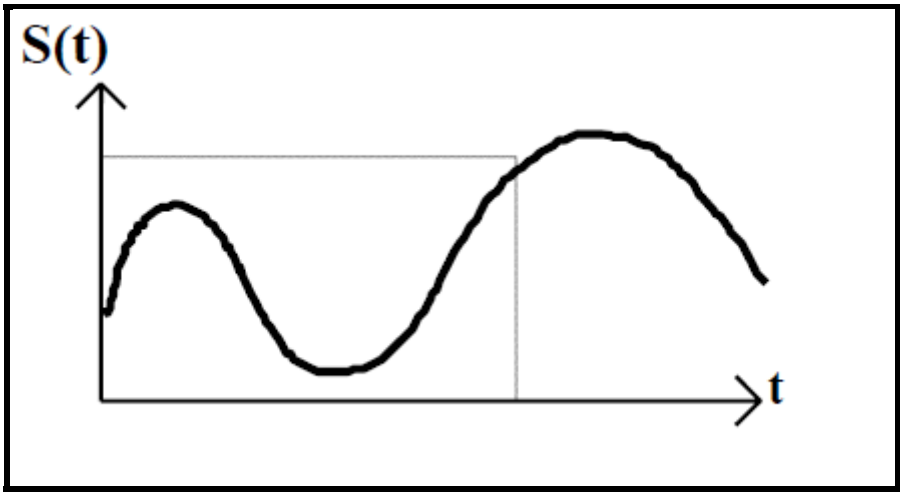


Figure III-3 : Signal analogique

La modulation analogique peut être en amplitude, en phase ou en fréquence.

I-4-1-2 Modulation d'amplitude (AM)

La modulation d'amplitude est le procédé le plus ancien, il consiste à faire varier linéairement l'amplitude du signal à moduler (généralement appelé porteuse) en fonction du signal modulant. Le principe repose sur la multiplication du signal porteur par le signal de basse fréquence (signal modulant). Il permet de transmettre des informations à longue distance grâce à la transposition du spectre du signal dans une bande de fréquence adaptée. Un signal modulé en amplitude varie au rythme du signal modulant.

Exemple

Le signal à transmettre (signal modulant) : $\mathbf{M(t)= A_m \cos (2\pi f_m t)}$

Le signal porteuse : $\mathbf{C(t)= A_c \cos (2\pi f_c t)}$

Le produit des deux signaux il va donnée le signal modulé :

$$\mathbf{S(t) = Ac\ A_m \cos (2\pi f_m t)\ A_c \cos (2\pi f_c t)}$$

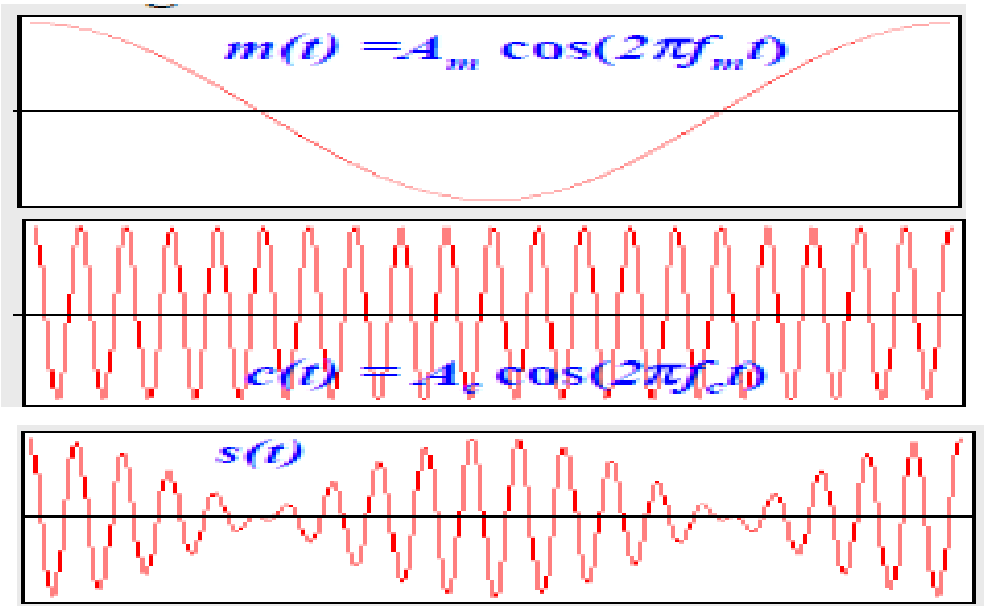


Figure III-4: Modulation d'amplitude

A_c : Amplitude de la porteuse.

A_m : Amplitude de signal utile.

$2\pi f_c$: Pulsation du signal porteur

$2\pi f_m$: Pulsation du signal utile,

M : taux de modulation, 0>M>1, M=Am/Ac

I-4-1-3 Modulation de fréquence (FM)

L'amplitude d'un signal modulé en amplitude est souvent modifiée par les parasites dus essentiellement aux interférences avec les autres stations émettrices. On a cherché alors à moduler la fréquence du signal en laissant son amplitude constante : c'est la modulation de fréquence.

La modulation de fréquence présente un autre avantage : sa puissance d'émission reste constante, une sélectivité accrue et elle est insensible aux parasites.

La modulation de fréquence (FM) est la technique de modulation analogique la plus utilisée dans les systèmes de transmissions mobiles.

Le signal modulé en fréquence garde l'amplitude constante, mais sa fréquence varie légèrement au cours du temps autour de la valeur f_p (fréquence de la porteuse). Les variations de fréquence reproduisent le signal modulant.

La fréquence du signal modulé n'est pas constante. Ses valeurs restent proches de la fréquence de la porteuse, mais elle varie au cours du temps en fonction du signal modulant.

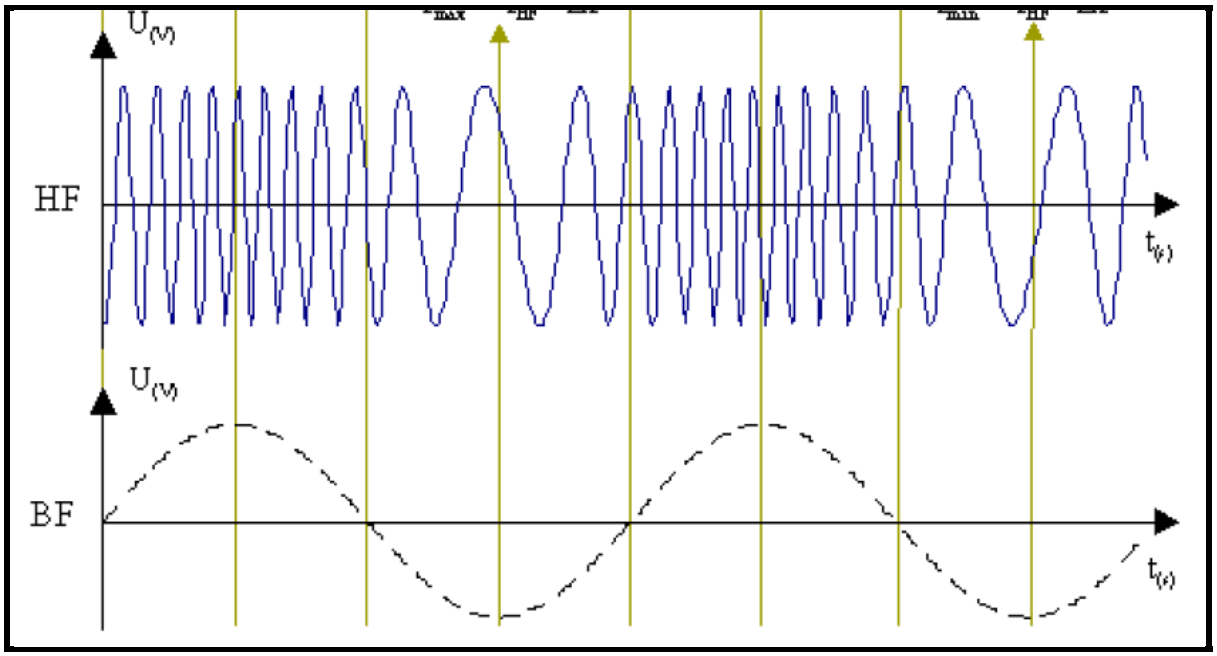


Figure III-5: Modulation de fréquence

I-4-1-4 La modulation de phase (PM)

La modulation par déplacement de phase PK (Phase Keying) c'est la phase instantanée qui varie linéairement en fonction du signal modulant, C'est-à-dire la phase de l'onde porteuse varie par rapport à un état de phase initial avec le signal modulant BF, la fréquence et l'amplitude restent constantes.

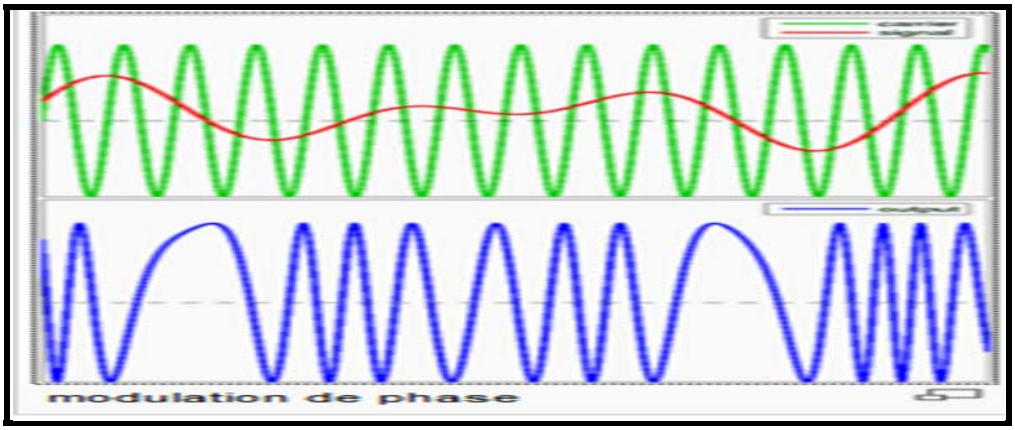


Figure III-6 : Modulation de phase

I-4-2 La modulation numérique

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 sont identiques dans leur principe à ceux vus auparavant pour les modulations analogiques, 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 ; on peut atteindre ainsi des taux d'erreurs (nombre de bits erronés divisé par le nombre de bits total) de 10^{13} .

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.

Le schéma de principe d'une chaîne de transmission numérique est représenté sur la figure III-7

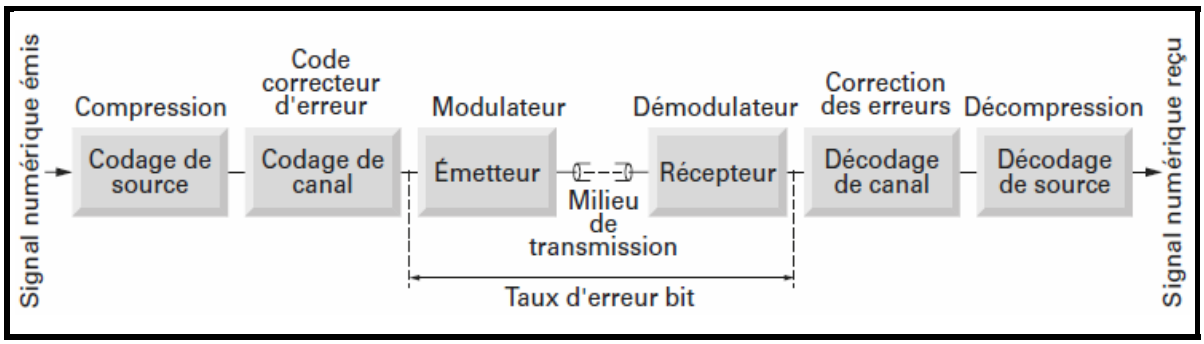


Figure III-7 : Synoptique d'une transmission numérique

I-4-2-1 Définition du signal numérique

Dans de nombreux cas, on ne souhaite pas ou on ne peut pas transmettre directement un signal analogique. On transmet alors après numérisation par exemple, le code binaire d'une grandeur. Les valeurs résultantes seront transmises en série et se présenteront alors comme une suite de 0 et de 1.

La figure III-8 représente un signal numérique dit NRZ pour Non Retour à Zéro

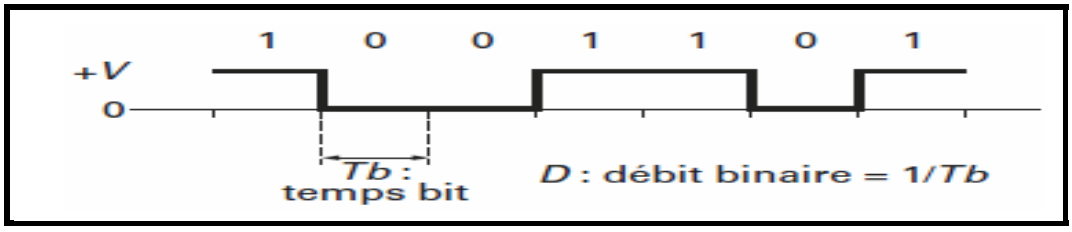


Figure III-8 : Représentation temporelle du signal NRZ

Tb : le temps pendant lequel un bit est transmis,

D : le débit binaire et vaut : **D= 1/ Tb**

Tb : exprimé en seconde, D est exprimé en bit par seconde ou baud.

I-4-2-2 Modulation d'amplitude numérique

La Modulation d'Amplitude Numérique ou modulation à saut d'amplitude (ASK, Amplitude Shift Keying) affecte à chaque état ou symbole numérique une valeur d'amplitude de la porteuse au rythme des données. Il s'agit comme en AM de faire varier l'amplitude Ap de la porteuse, de fréquence fixe fp "rapide",

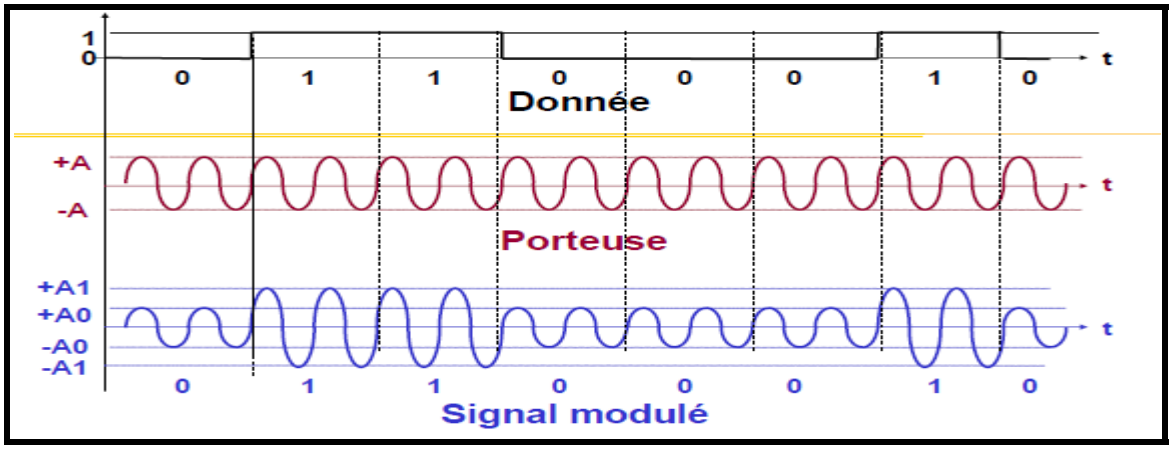


Figure III-9 : Modulation d'amplitude numérique

L'amplitude maximale du signal modulé vaut A0 quand le bit de donnée vaut 0, et A1 quand il vaut 1.

A0 = A – A

A1 = A + A

Où A est l'amplitude maximale de la porteuse.

On reconnaît le signal de donnée dans l'enveloppe du signal modulé

I-4-2-3 Modulation par déplacement de phase

La modulation par déplacement de phase, ou phase-shift keying (PSK), fait varier la phase de la porteuse entre la transmission d'un 0 et d'un 1.

Ce type de modulation, est utilisé en particulier pour les signaux GPS. La phase de la porteuse prend la valeur 0 ou π, selon la valeur du bit à transmettre.

L'expression du signal s(t) modulé est :

$$s(t) = P\sin\left(\omega_0 t + n(t)\frac{2\pi}{k}\right) \qquad n(t) = 0, 1, \ldots k-1$$

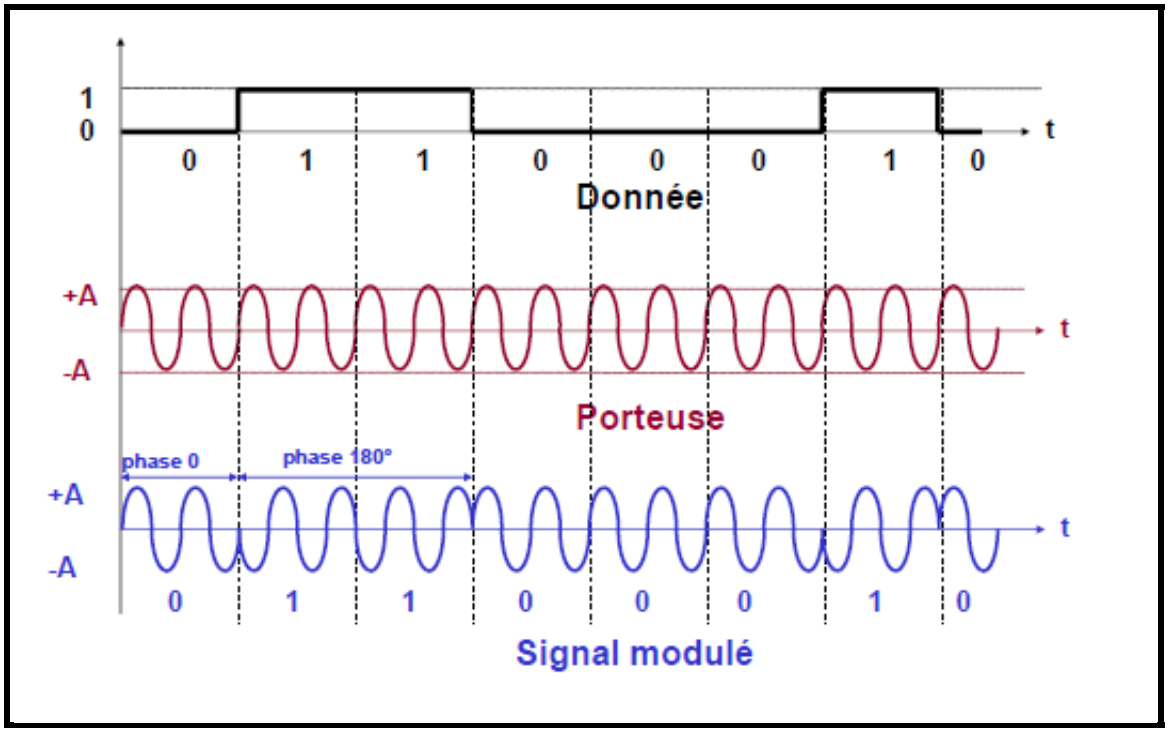


Figure III-10 : Modulation par déplacement de phase

La phase du signal modulé vaut 0 quand le bit de donnée vaut 0, et 180 ° quand il vaut 1.

I-4-2-4 Modulation par déplacement de fréquence

La modulation FSK peut être vu comme un cas particulier de la modulation de fréquence (FM), le signal modulant étant un signal binaire à 2 niveaux (par exemple NRZ), on associe à un niveau, une fréquence f1, et à l'autre, une fréquence f2. Il s'agit donc d'une modulation de fréquence à 2 fréquences discrètes. Mais, elle peut aussi se voir comme une double modulation d'amplitude ASK, le signal modulé peut alors être considéré comme la somme de deux signaux modulés en amplitude par le train binaire.

Le principe de cette modulation est relativement simple : le signal numérique comporte deux états, « 0 » et « 1 ». A chacun de ces états correspond une sinusoïde de fréquence déterminée.

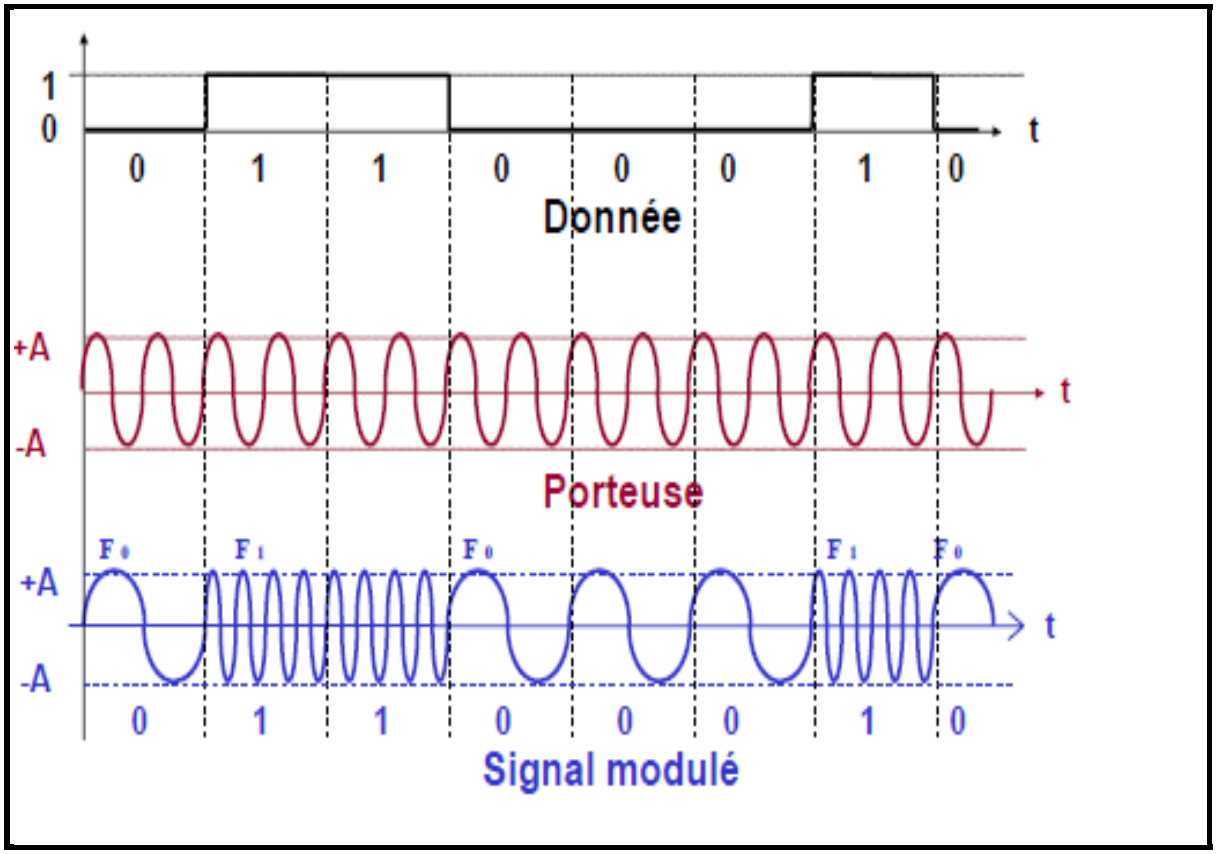


Figure III-11 : Modulation par déplacement de fréquence

I-4-3 Bilan

On peut résumer les avantages et les inconvénients des différents types de modulation numérique sous la forme d'un tableau :

Type de modulation	Avantage(s)	Inconvénient(s)
ASK	Modulation et démodulation facile à mettre en œuvre.	Faible résistance au bruit.
PSK	Bonne résistance au bruit.	Démodulation complexe (synchronisation difficile).
FSK **Discontinue** **Continue**	Simple à mettre en place. Bande passante plus étroite.	Nécessite une grande bande passante (faible débit). Plus complexe à réaliser.

Figure III-12 : Table de bilan de différents types de modulation

Dans ce projet, nous allons traiter l'implémentation d'un modulateur FSK (frequency shift keying) sur un circuit FPGA. La modulation FSK permet de transmettre l information numérique à l'aide d une porteuse qui prend un nombre discret de fréquence.
Dans notre cas 5 KHz pour transmettre « 1 » et 1 KHz pour transmettre « 0 ».

Nous allons pour cela utilisé un circuit FPGA et un convertisseur numérique analogique. L'outil que nous utiliserons pour mettre en œuvre notre réalisation est le logiciel QUARTUS II.

II Présentation générale de logiciel QUARTUS II

C'est un logiciel de CAO (société Altera) permettant la simulation, la synthèse et la programmation des circuits logiques programmables. QUARTUS II permet une conception d'un circuit numérique à partir d'une entrée sous forme :

- D'une schématique traditionnelle.
- D'une netlist standard de type EDIF (Electronic Design Interchange Format).
- Textuelle à l'aide d'un langage de description de matériel : VHDL, Verilog et A-HDL (Altera HDL).

L'architecteur de notre modulateur est présenté sur la figure ci-dessous,

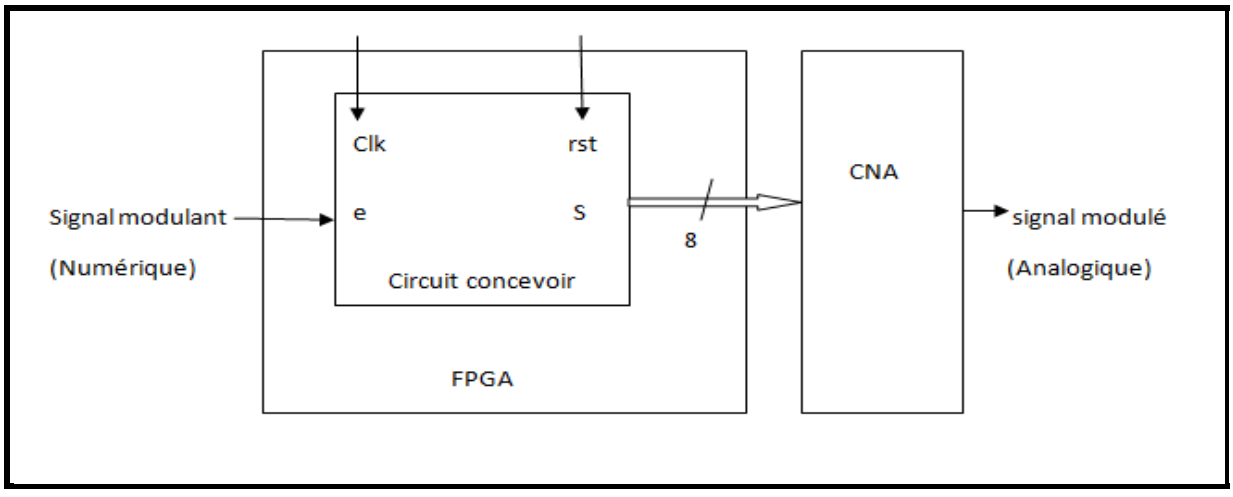


Figure III-13 : Schéma de principe du modulateur FSK

III Les étapes de conception

On a spécifié ce circuit avec une machine à états manipulant des variables comme le montre la figure suivante :

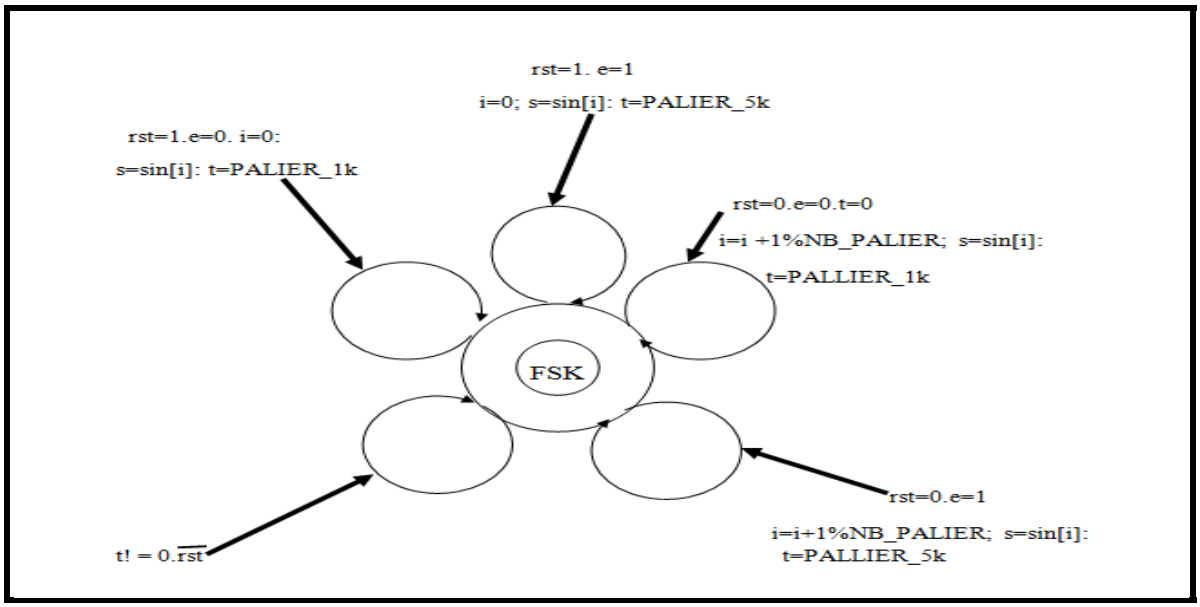


Figure III-14 : Machine à états manipulant des variables du modulateur FSK

Il n'y a qu'un seul état, le circuit se réduit à une partie opérative. Celle-ci est composée de quatre blocs principaux.

L'architecture des éléments de la partie opérative de notre modulateur FSK est présentée sur la figure suivante :

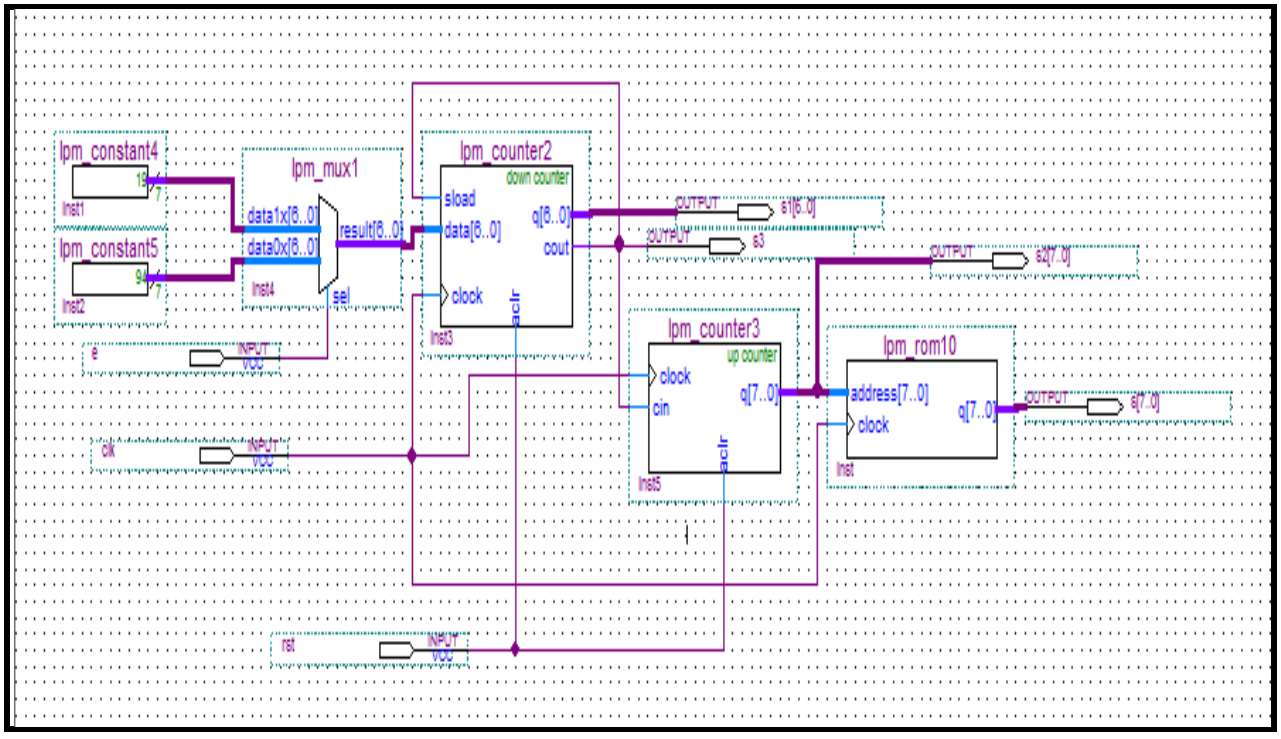


Figure III-15 : Partie opérative du modulateur FSK

Les éléments de la partie opérative sont les suivants :

- **un compteur de durée de palier**

La durée d'un palier du sinus à 1 KHz c'est 1 ms / nombre de paliers**.**

La durée d'un palier du sinus à 5 KHz c'est 0.2 ms / nombre de paliers

Nombre de paliers c'est le nombre d'échantillons du sinus, dans notre cas si 256 échantillons.

La fréquence d'horloge du FPGA est de 24 MHz

Le nombre des cycles d'horloge pour chaque palier sont :

PALIER_1k =$24.10^6/(1.10^3 \times 256)$ = 94 cycles d'horloge.

PALIER_5k = $24.10^6/(5.10^3 \times 256)$ = 19 cycles d'horloge.

Il faut donc un compteur à 7 bits, il doit également disposer d'une entrée de chargement et d'une sortie indiquant un passage à zéro. Voir la figure suivante :

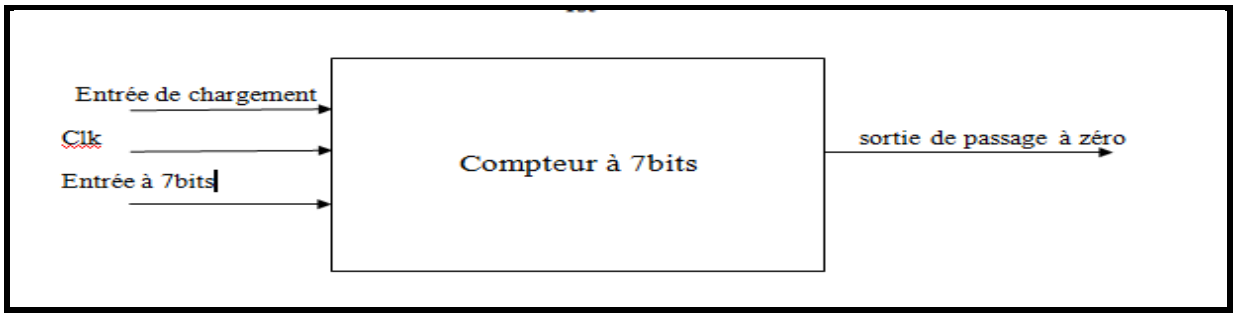


Figure III-16: Schéma d'un compteur à 7 bits

- **un multiplexeur 2 vers 1**

Il permet de sélectionner la valeur à charger, la sélection se fait directement par l'entrée « e » si e=0 il va chargée le signal sinus à 1 KHz, sinon si e=1 il va chargée le signal sinus à 5 KHZ.

Voir la figure suivante :

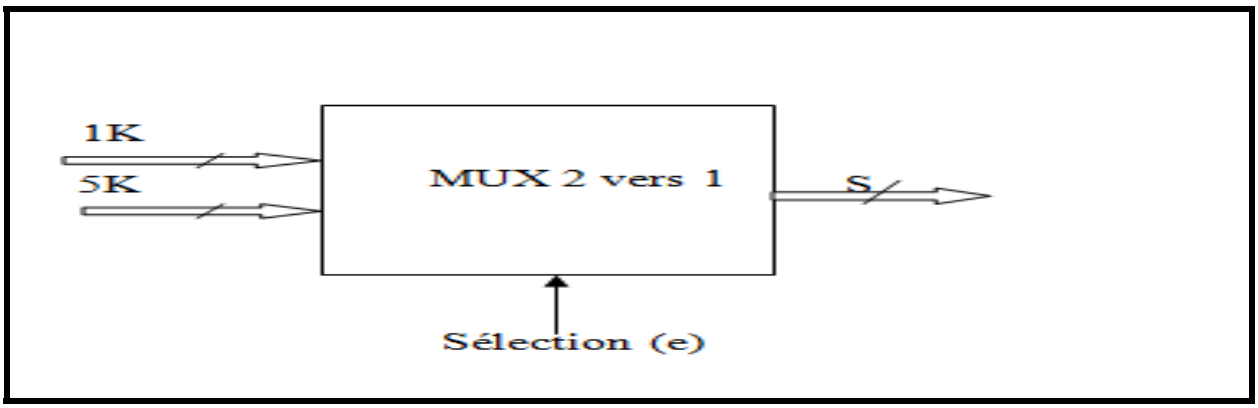


Figure III-17 : Schéma d'un multiplexeur 2 vers1

- **un compteur à 8 bits**

Permettant de faire varier l'adresse, il y a 256 adresses il faut donc un compteur à 8 bits, le compteur doit disposer d'une entrée de validation d'horloge, l'incrémentation de l'adresse ne se fait que lorsque le compteur de durée de palier passe à zéro. Voir la figure suivante :

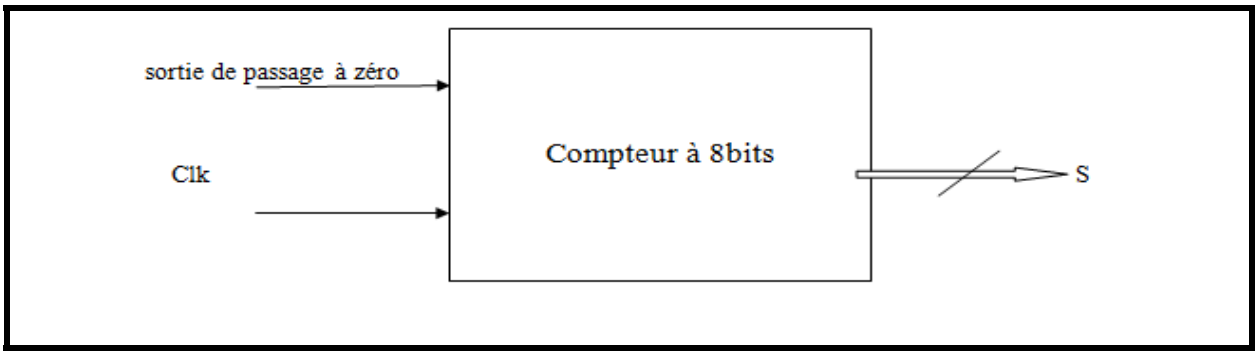


Figure III-18 : Schéma d'un compteur à 8 bit

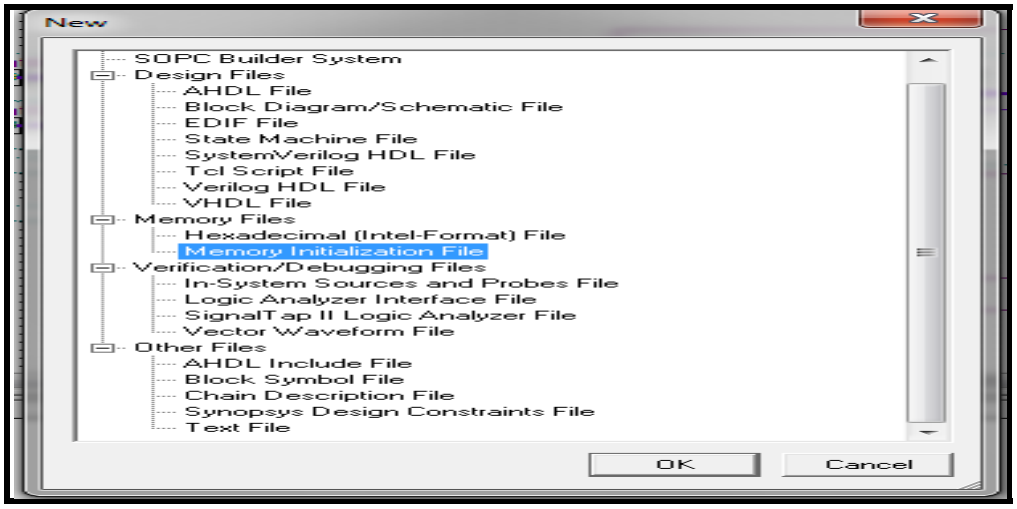
- **Une ROM**

Nous avons générer un signal sinusoïdale a laide du logiciel « C » puis nous avons stockées les valeurs de notre signal dans une ROM sous la forme de 256 valeurs de 8 bits. L'expression des coefficients placer dans la ROM est :

$$Sin[i] = 255/2\times [\sin(2\times pi\times i/256) + 1]$$

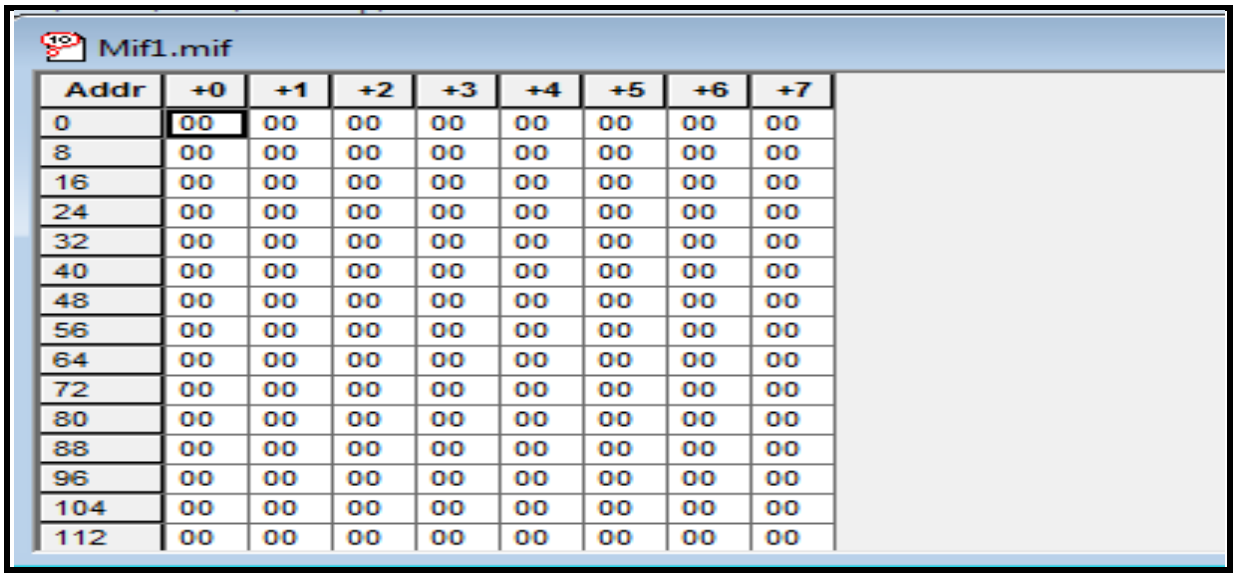
Pour stockées les valeurs de signale sinusoïdale dans la ROM nous avons suive les étapes suivante : **File → new → memory initialization file**

Voir la fenêtre suivante :



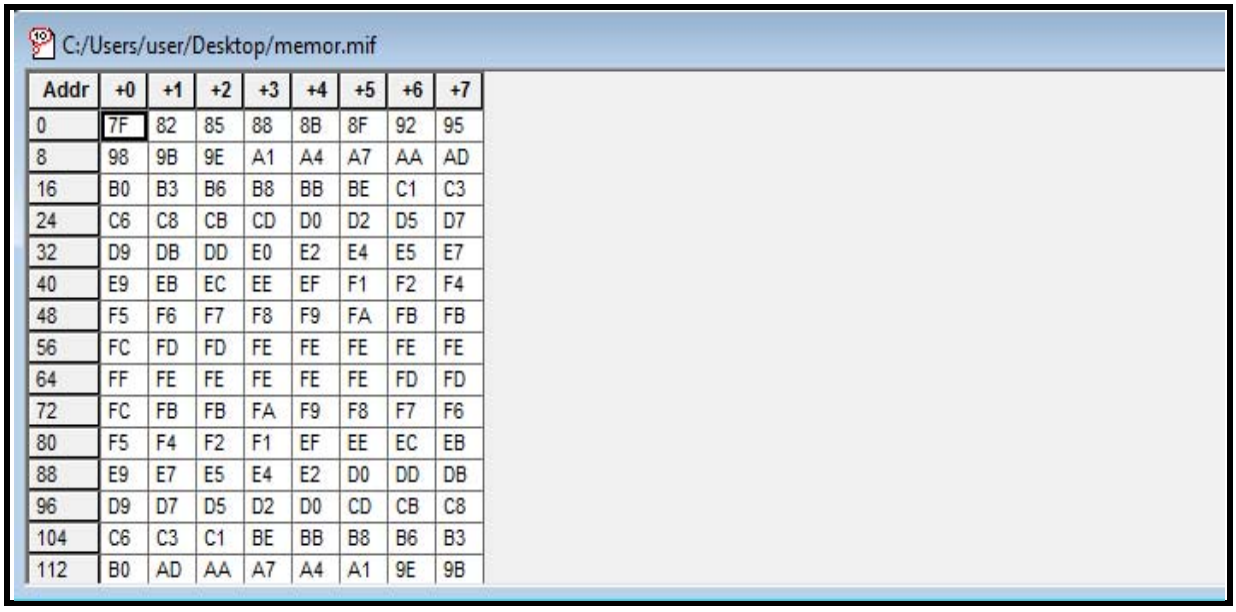
On clique sur OK une fenêtre apparait.

Voir la fenêtre suivante :



Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	00	00	00	00	00	00	00	00
8	00	00	00	00	00	00	00	00
16	00	00	00	00	00	00	00	00
24	00	00	00	00	00	00	00	00
32	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00
48	00	00	00	00	00	00	00	00
56	00	00	00	00	00	00	00	00
64	00	00	00	00	00	00	00	00
72	00	00	00	00	00	00	00	00
80	00	00	00	00	00	00	00	00
88	00	00	00	00	00	00	00	00
96	00	00	00	00	00	00	00	00
104	00	00	00	00	00	00	00	00
112	00	00	00	00	00	00	00	00

On remplie la fenêtre par les valeurs du signale sinusoïdale puis on le sauvegarde. Voir la figure suivante

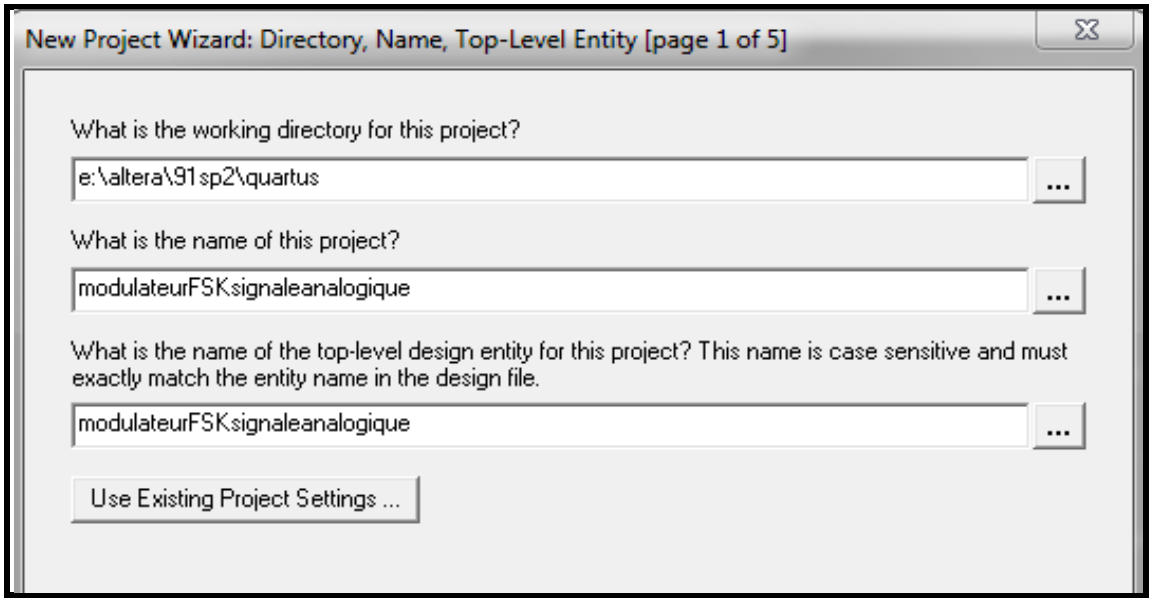


Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	7F	82	85	88	8B	8F	92	95
8	98	9B	9E	A1	A4	A7	AA	AD
16	B0	B3	B6	B8	BB	BE	C1	C3
24	C6	C8	CB	CD	D0	D2	D5	D7
32	D9	DB	DD	E0	E2	E4	E5	E7
40	E9	EB	EC	EE	EF	F1	F2	F4
48	F5	F6	F7	F8	F9	FA	FB	FB
56	FC	FD	FD	FE	FE	FE	FE	FE
64	FF	FE	FE	FE	FE	FE	FD	FD
72	FC	FB	FB	FA	F9	F8	F7	F6
80	F5	F4	F2	F1	EF	EE	EC	EB
88	E9	E7	E5	E4	E2	D0	DD	DB
96	D9	D7	D5	D2	D0	CD	CB	C8
104	C6	C3	C1	BE	BB	B8	B6	B3
112	B0	AD	AA	A7	A4	A1	9E	9B

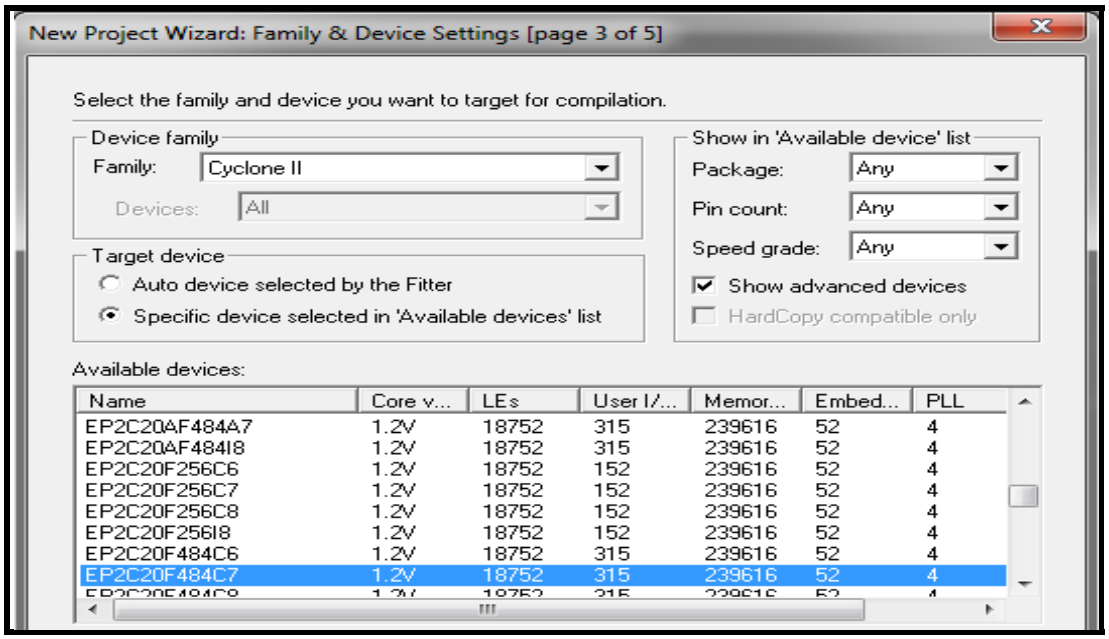
II-1 Les étapes de création de modulateur FSK avec le logiciel QUARTUS II

Voici les étapes que nous allons suivi pour réaliser notre modulateur :

- créer le **projet**



On clique sur **Next**



On a choisie dans les fenêtres **Family** et **Available Devices** le circuit logique programmable à utiliser, qui est un FPGA Cylone-II de référence EP2C20F484C7. L'interprétation de cette référence est la suivante :

EP : composant programmable électriquement.

2C : famille des cyclone-II.

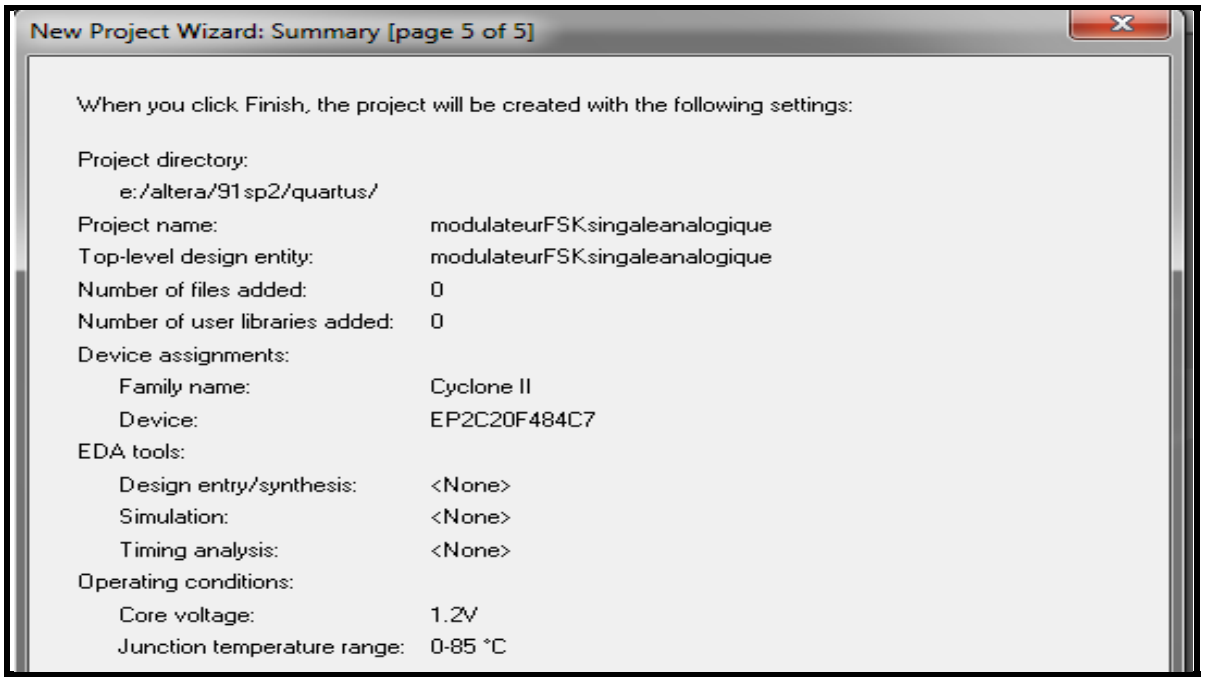
20 : nombre d'éléments logique (20000).

484 : nombre de bronches.

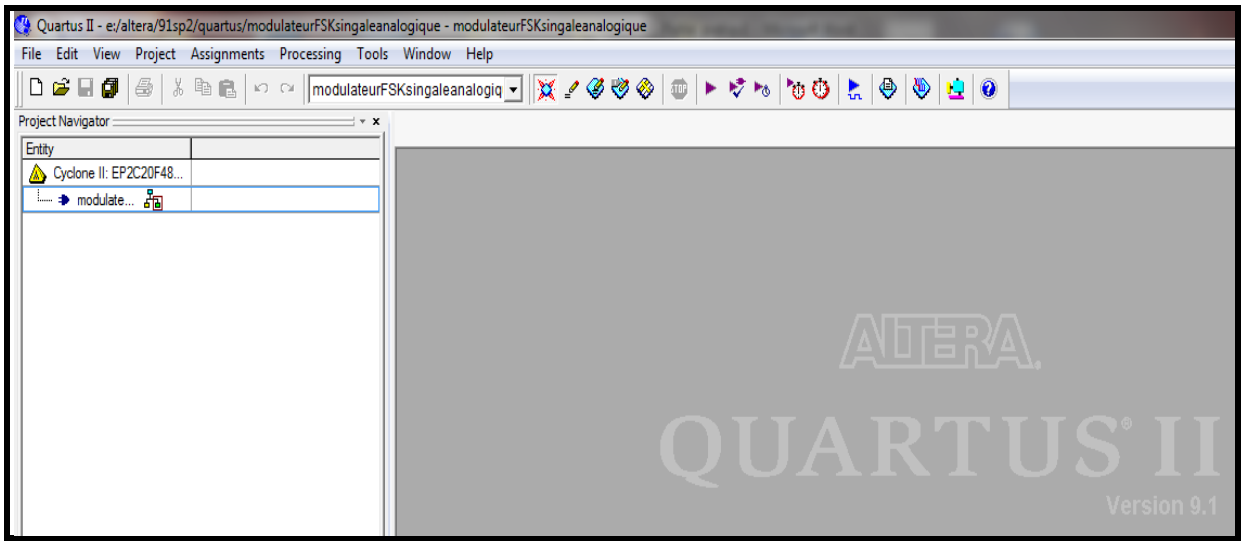
7 : vitesse (plus le chiffre est petit, plus le circuit est rapide).

La fenêtre nous renseigne également sur le nombre de blocs logique utilisable c'est 18752 blocs dans notre cas.

On passe à la fenêtre **EDA TOOL Setting**, une fenêtre récapitulative apparait. Voir la figure suivante



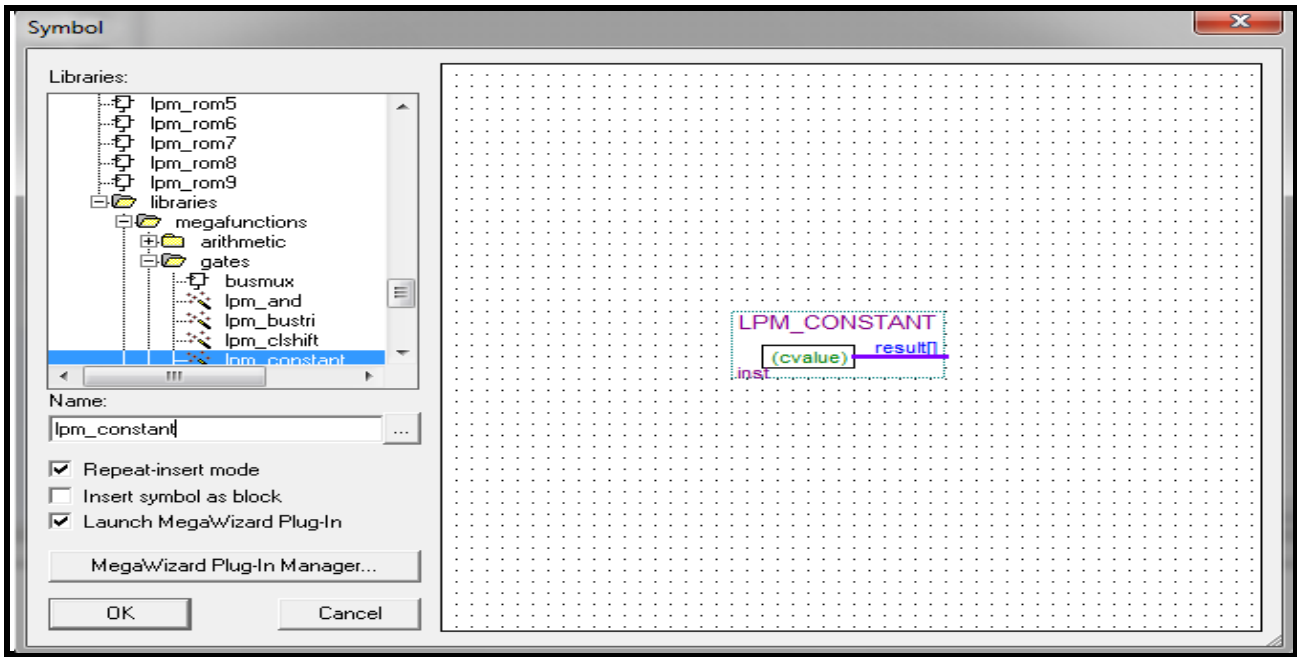
Dans le navigateur de projet un onglet avec le type de composant et l'entité maitre apparait :



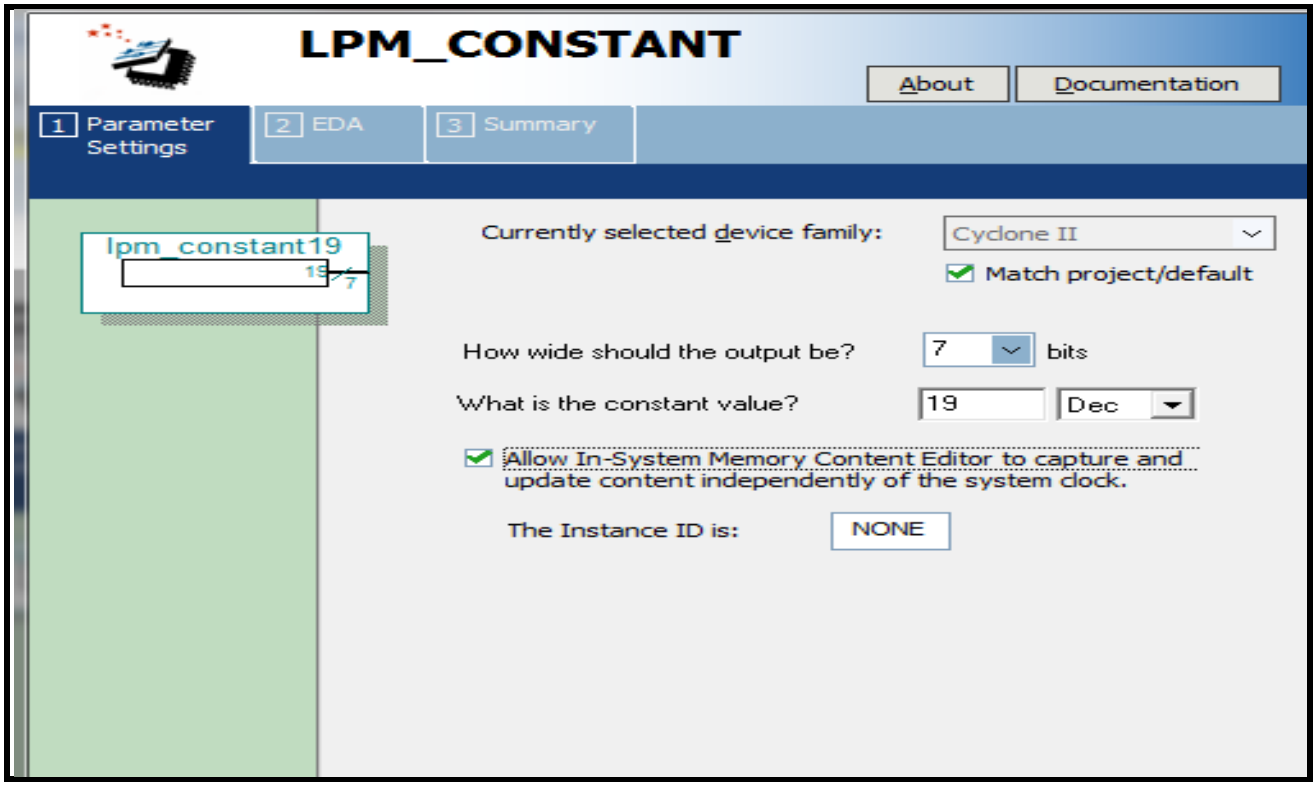
Pour faire la description complète du système on suit ces étapes :

- d' abord on retire tout les éléments constituant notre modulateur dans la bibliothèque de QUARTUS II.

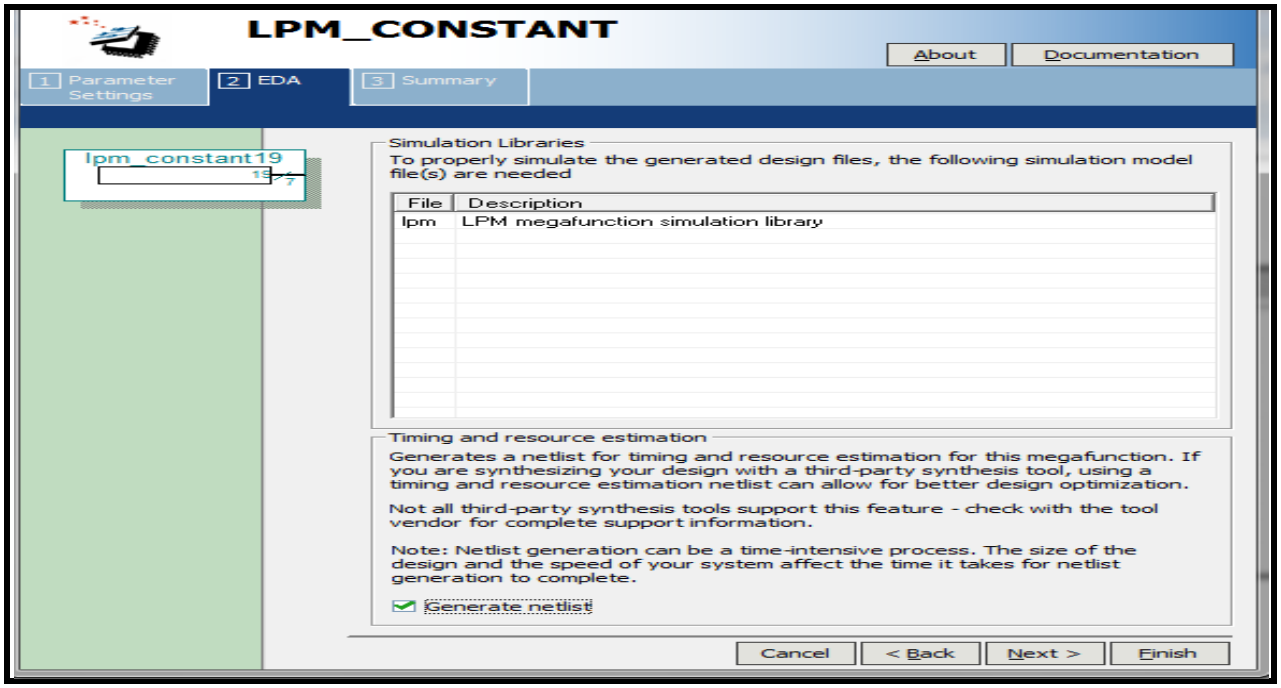
  Pour charger les entrer de chargement on suit les étapes suivante :



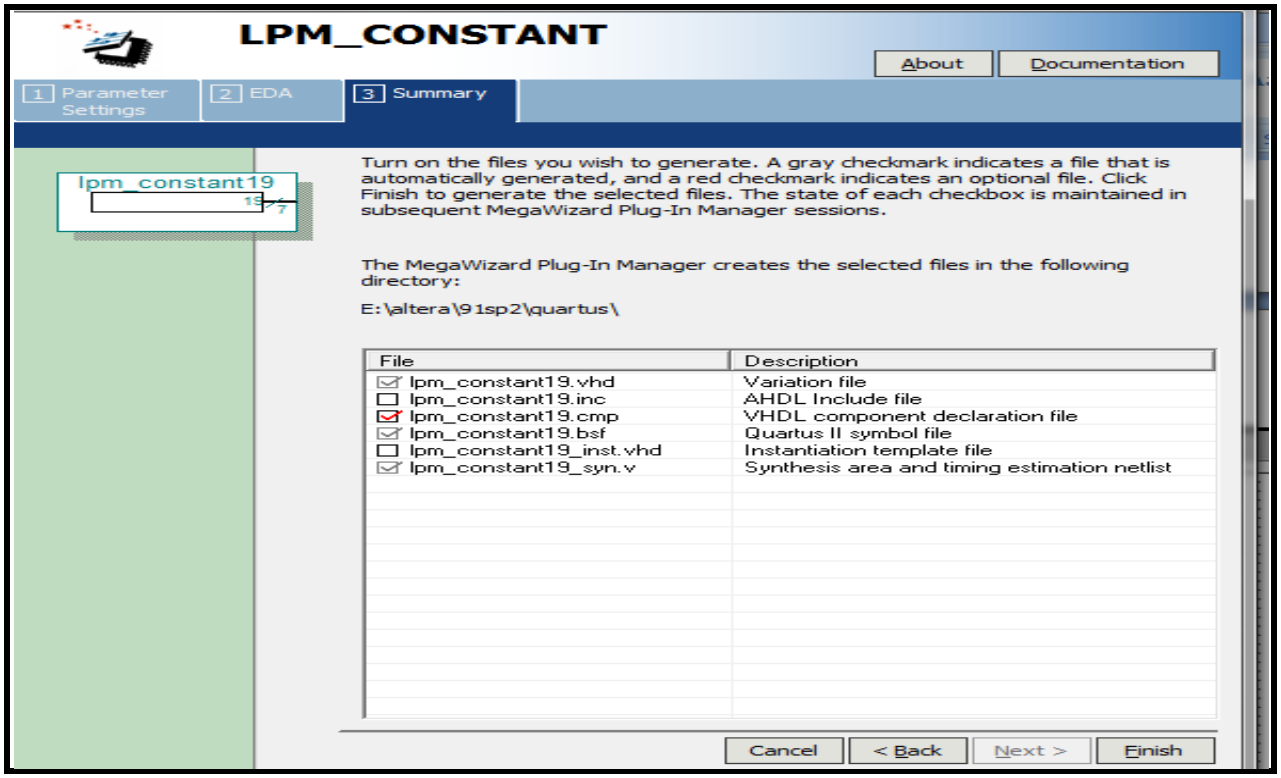
On clique sur **OK,**



On clique sur **Next**



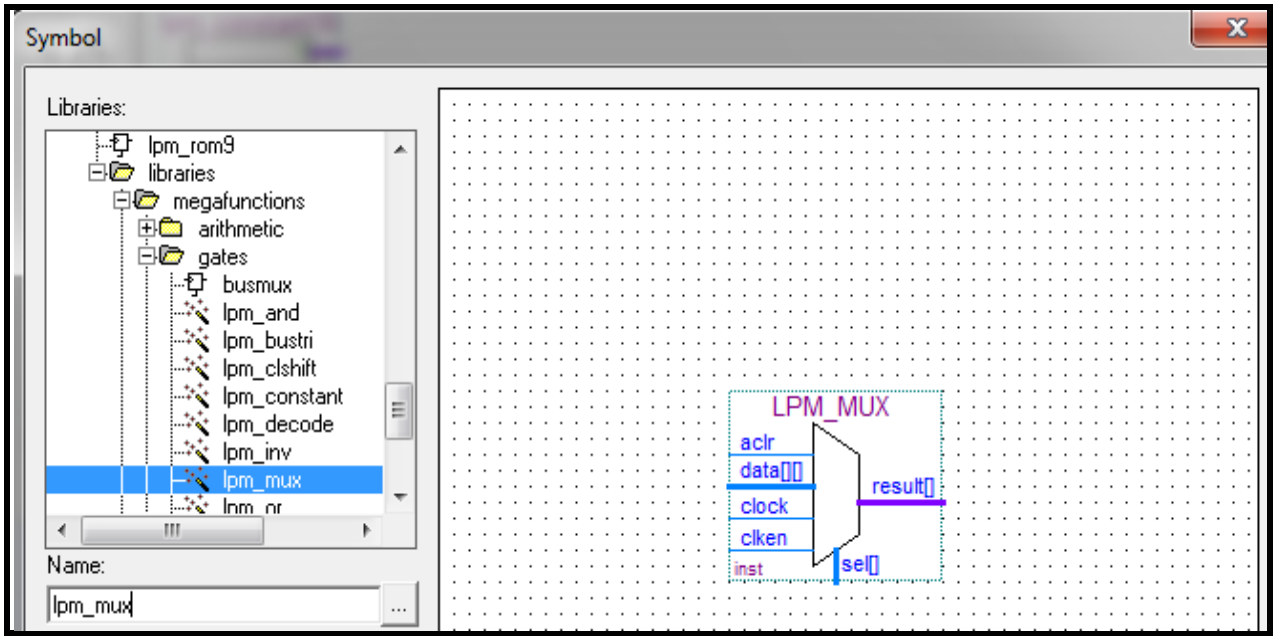
On clique sur **Next**



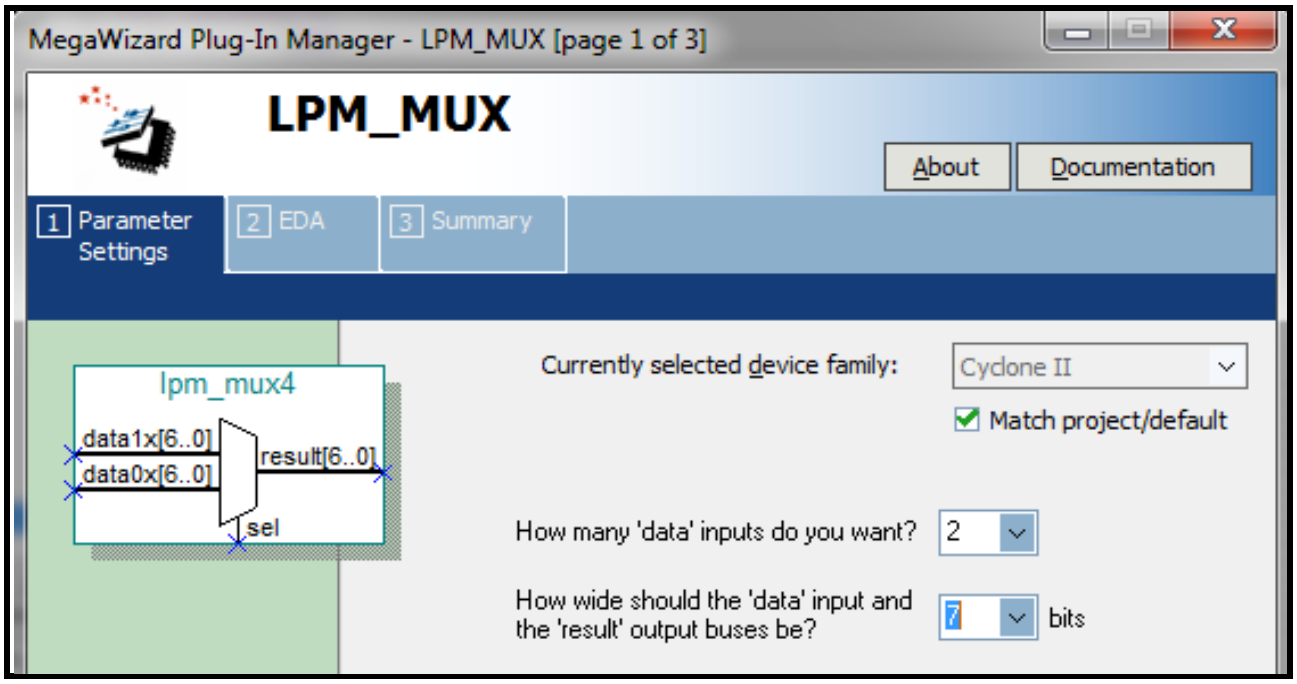
On clique sur **Finish**

Pour la deuxième entré de chargement on lui charge 94 cycles d'horloge on suivant les mêmes étapes.

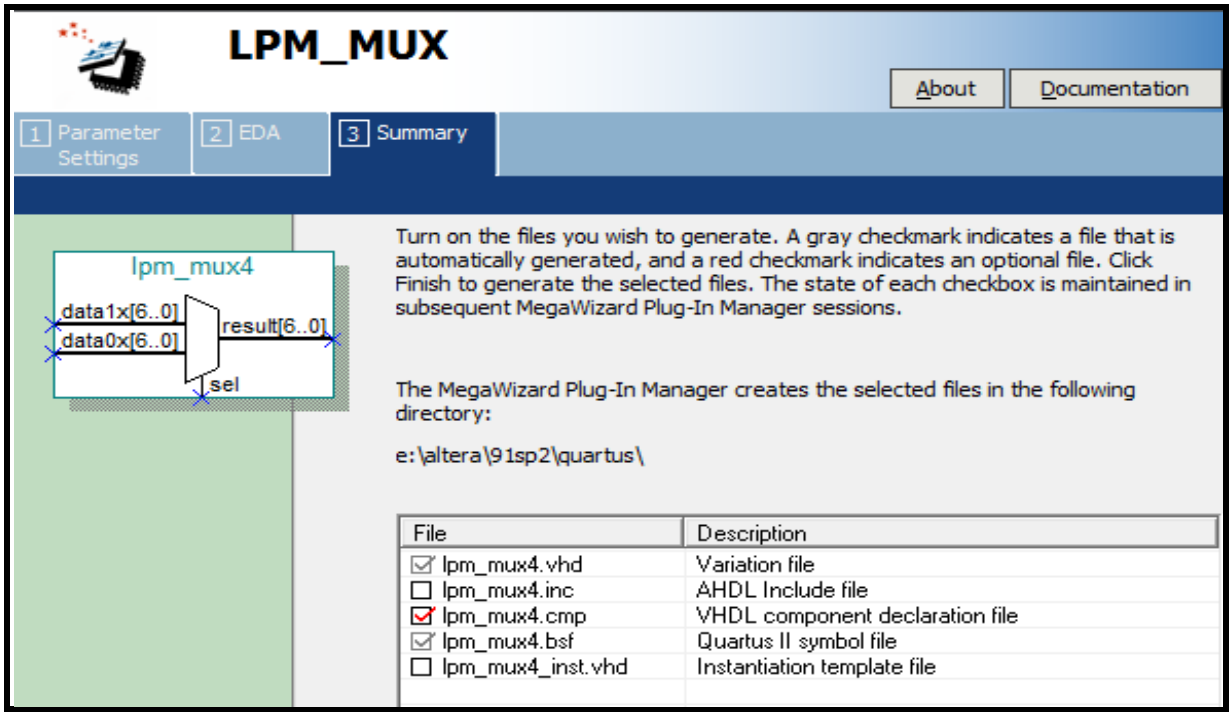
Concernant le multiplexeur on suit les étapes suivantes :



On clique sur **OK**

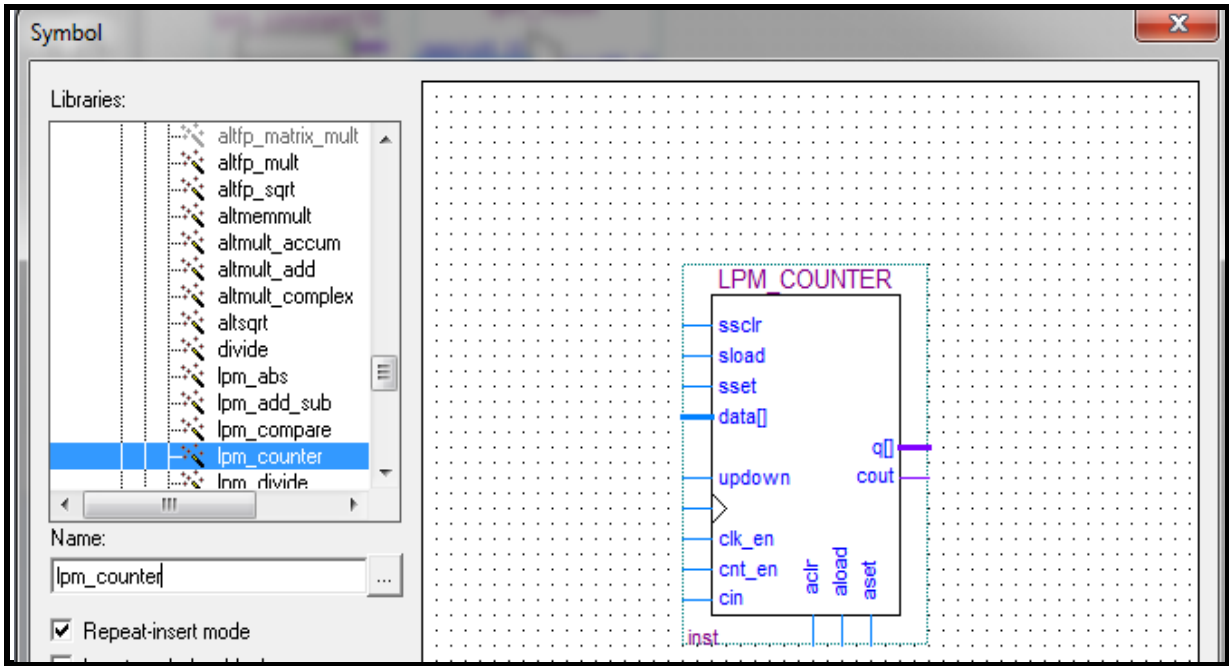


On clique sur **Next**

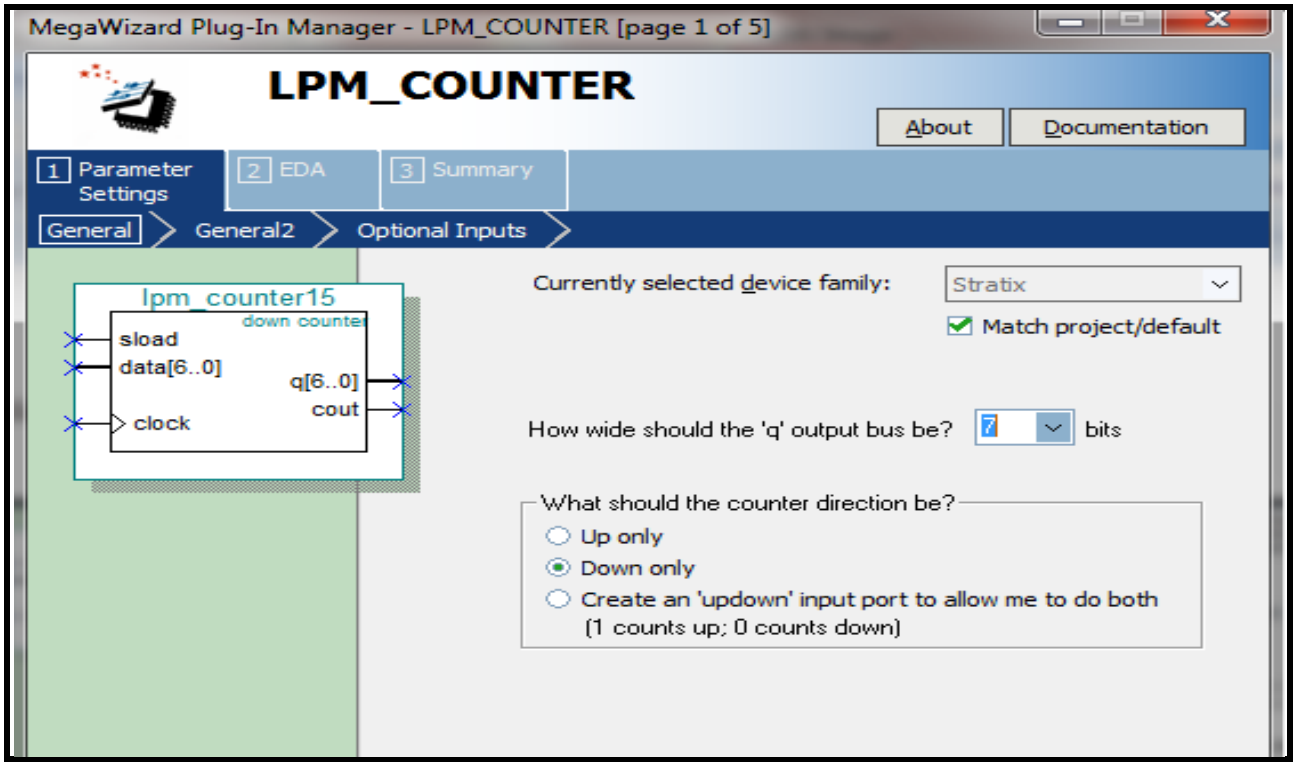


On clique sur **Finish**

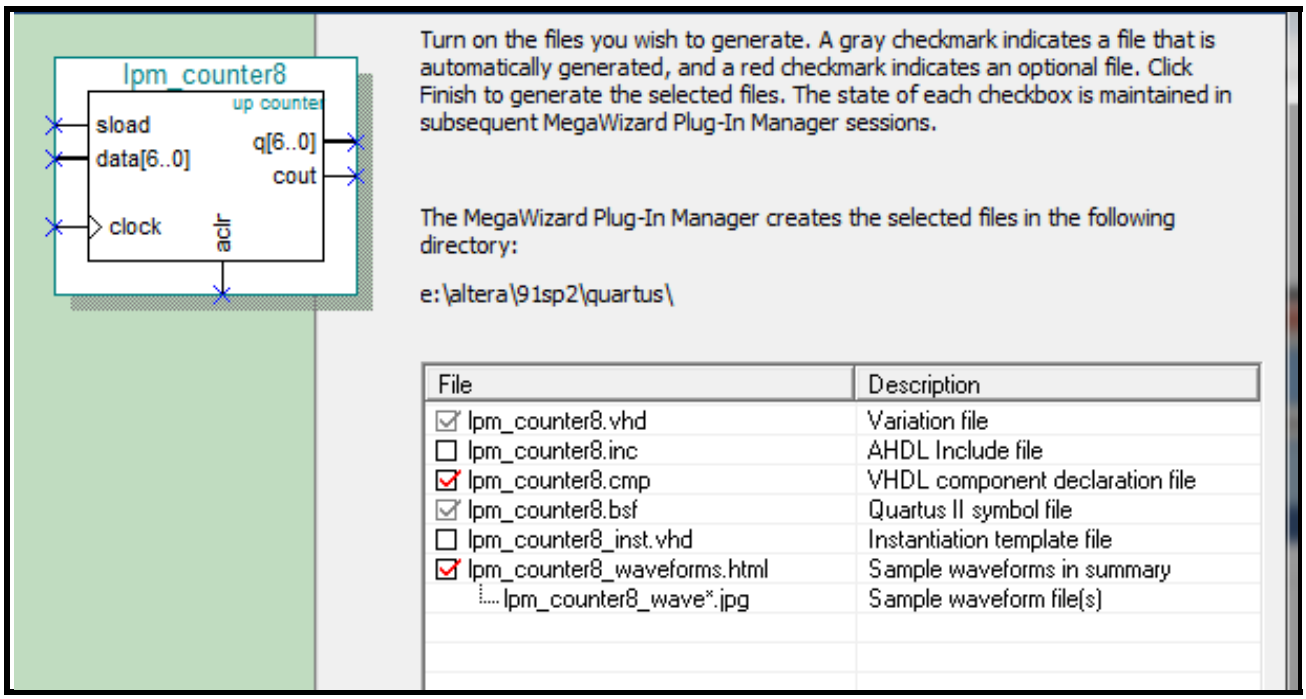
Par apport au compteur à 7 bits on suit les étapes suivantes



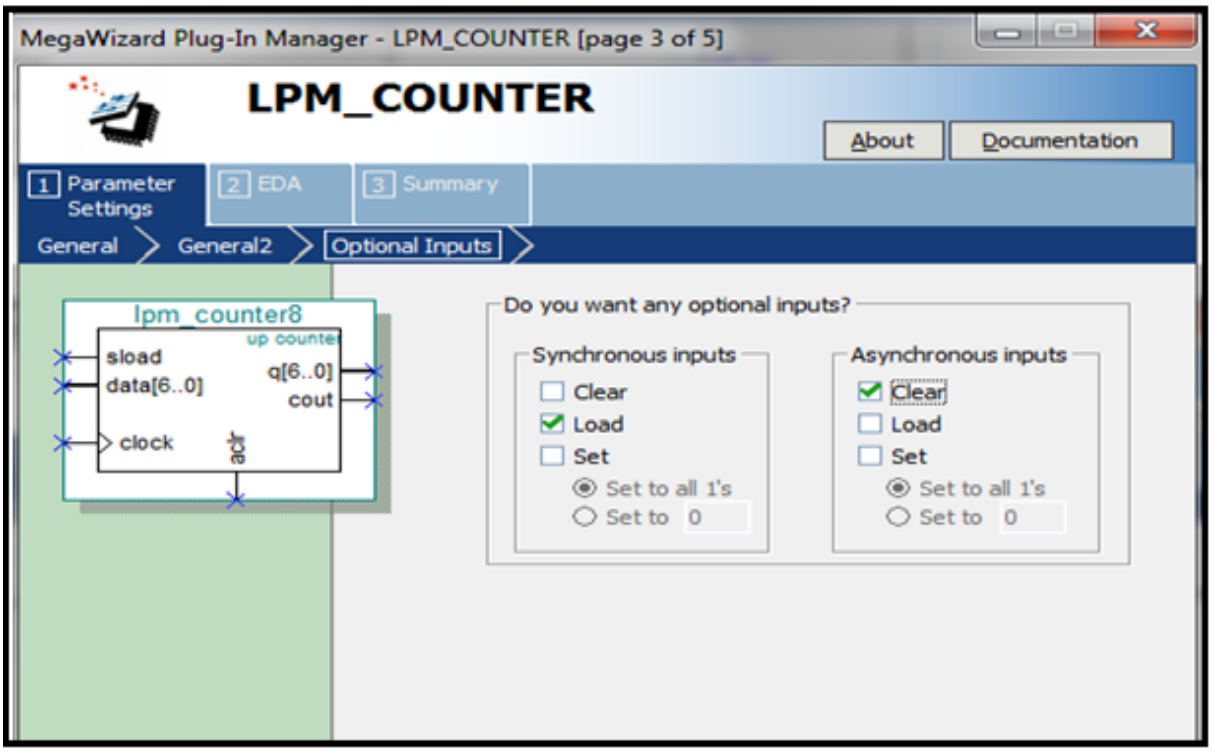
On clique sur **OK**



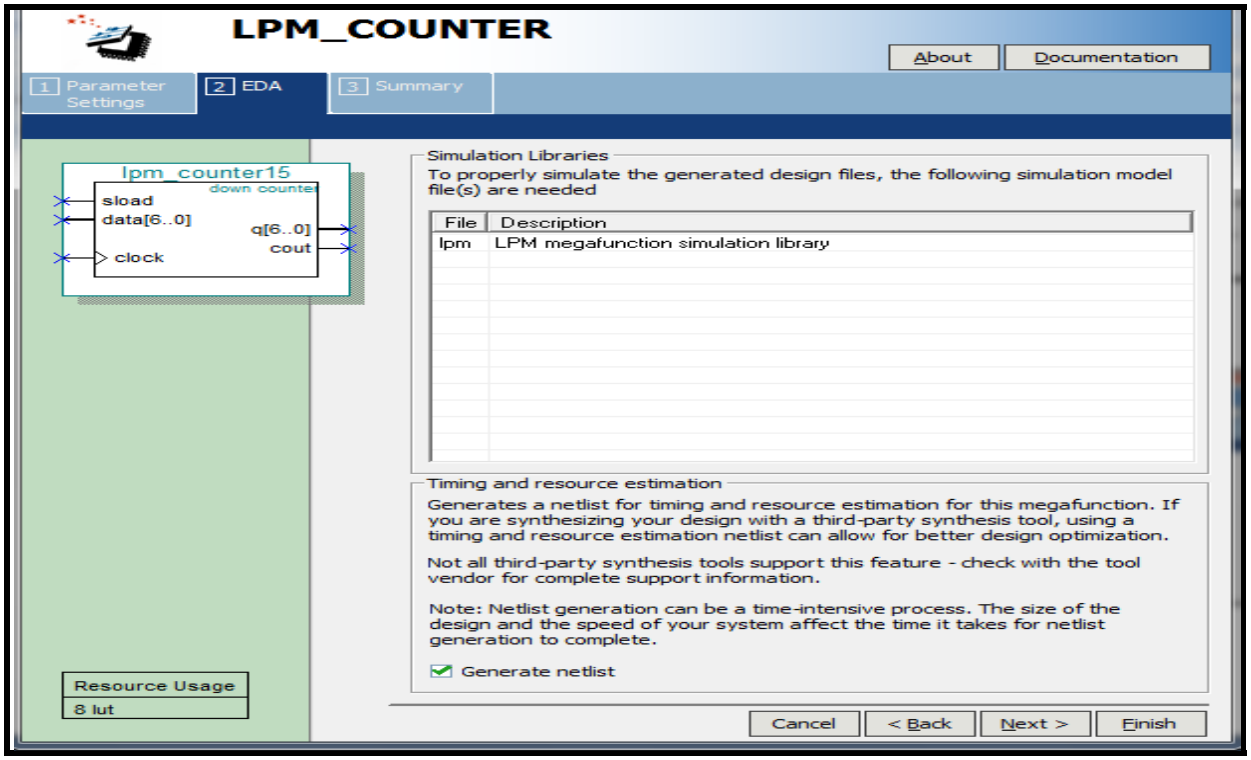
On clique sur **Next**



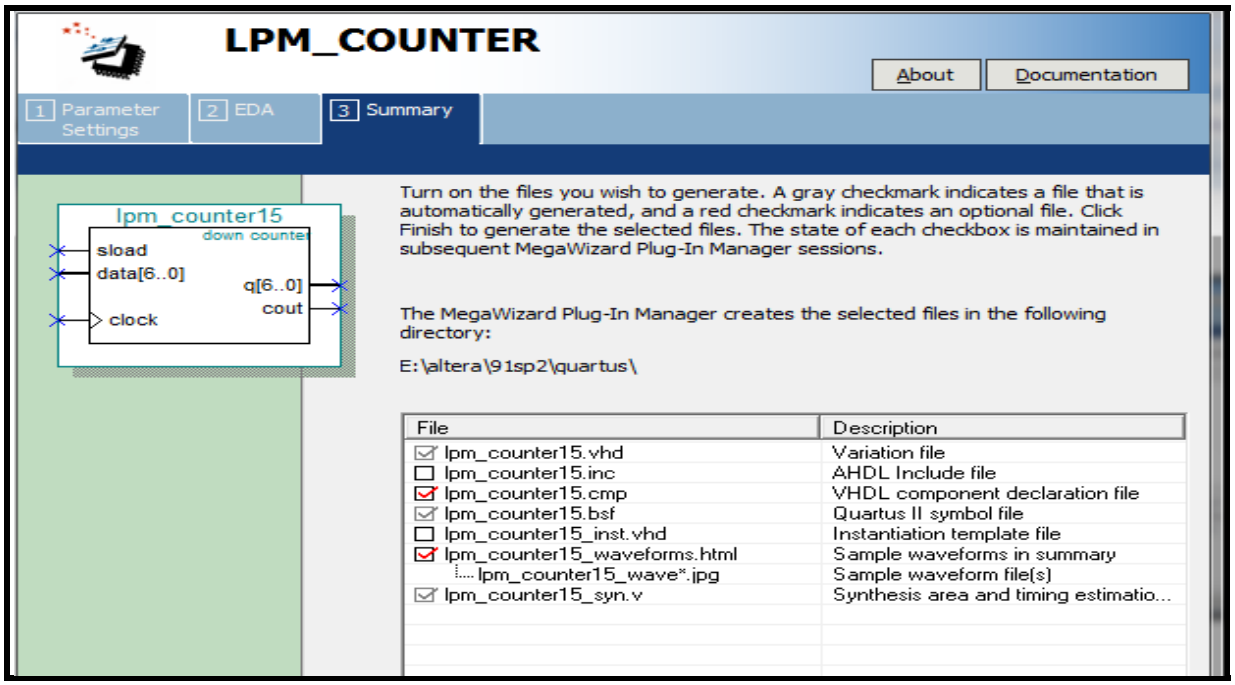
On clique sur **Next**



On clique sur **Next**

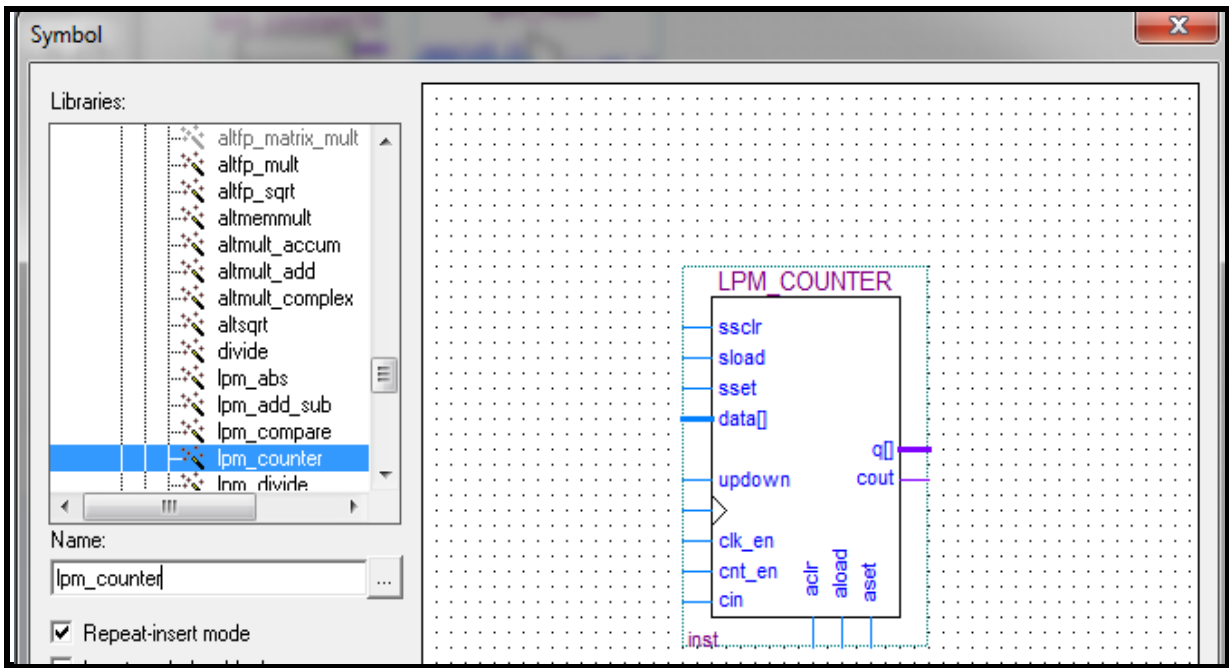


On clique sur **Next**

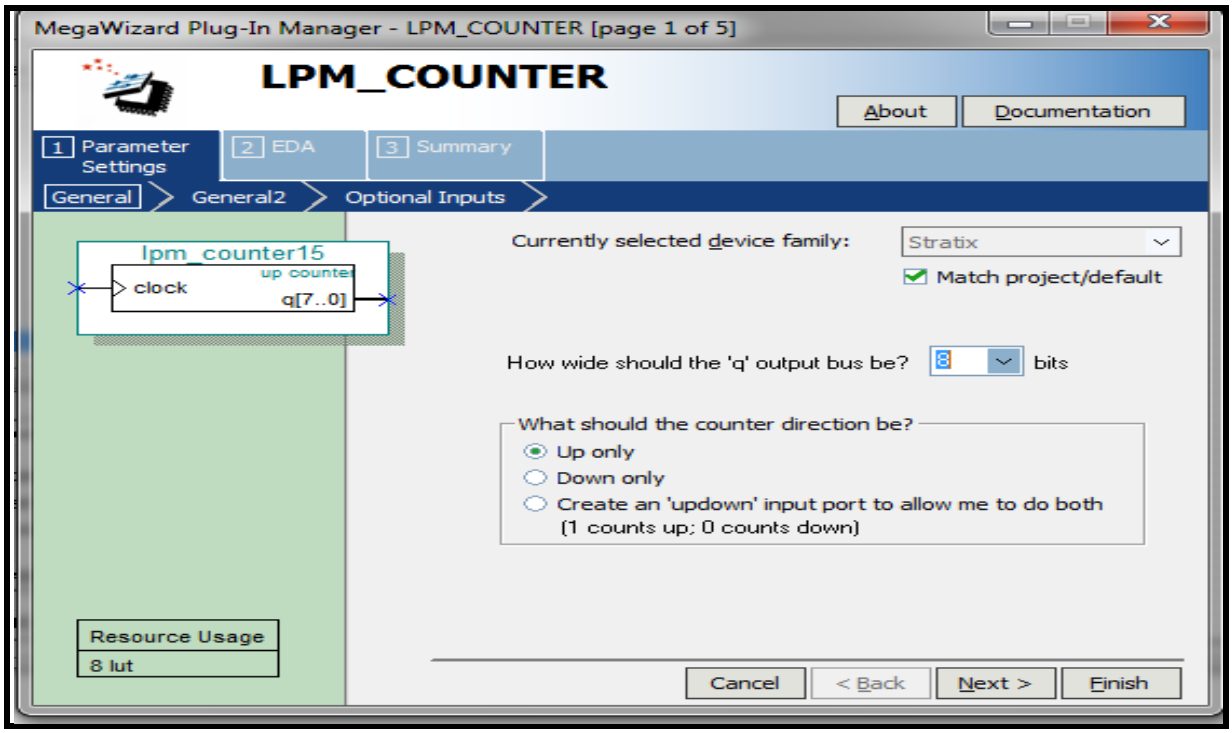


On clique sur **Finish**

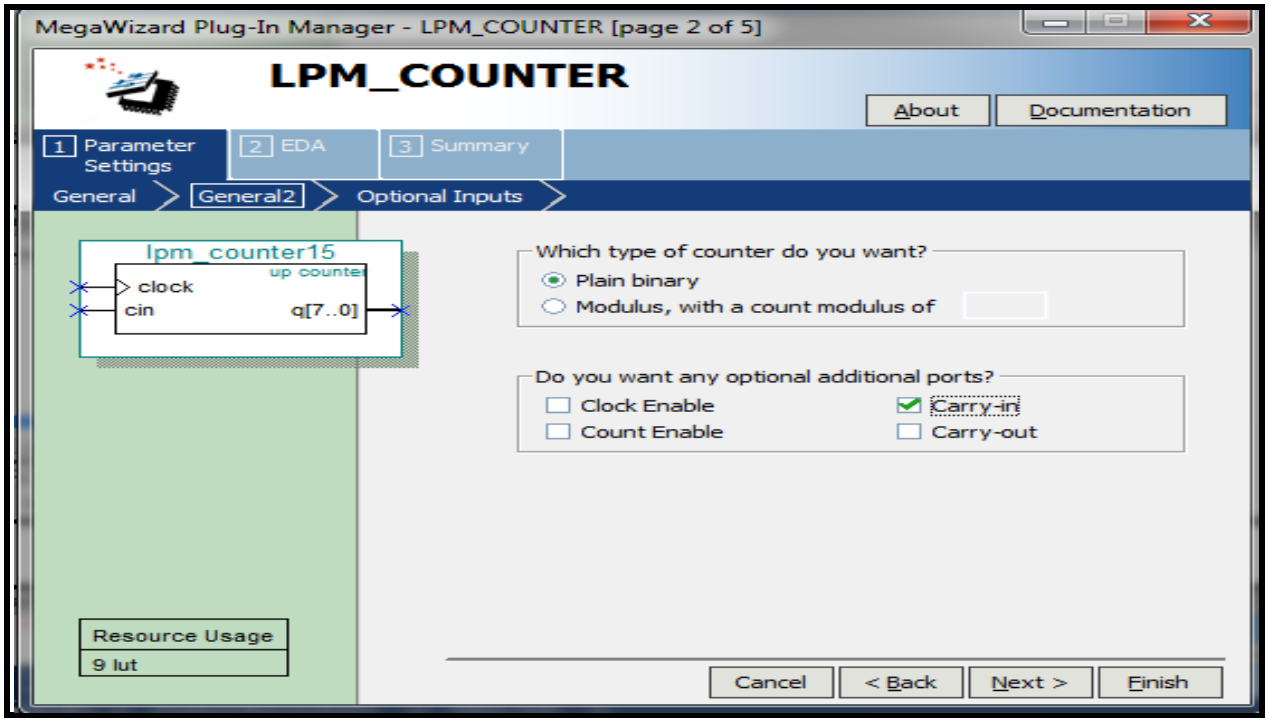
Par apport au compteur à 8 bits on suit les étapes suivantes



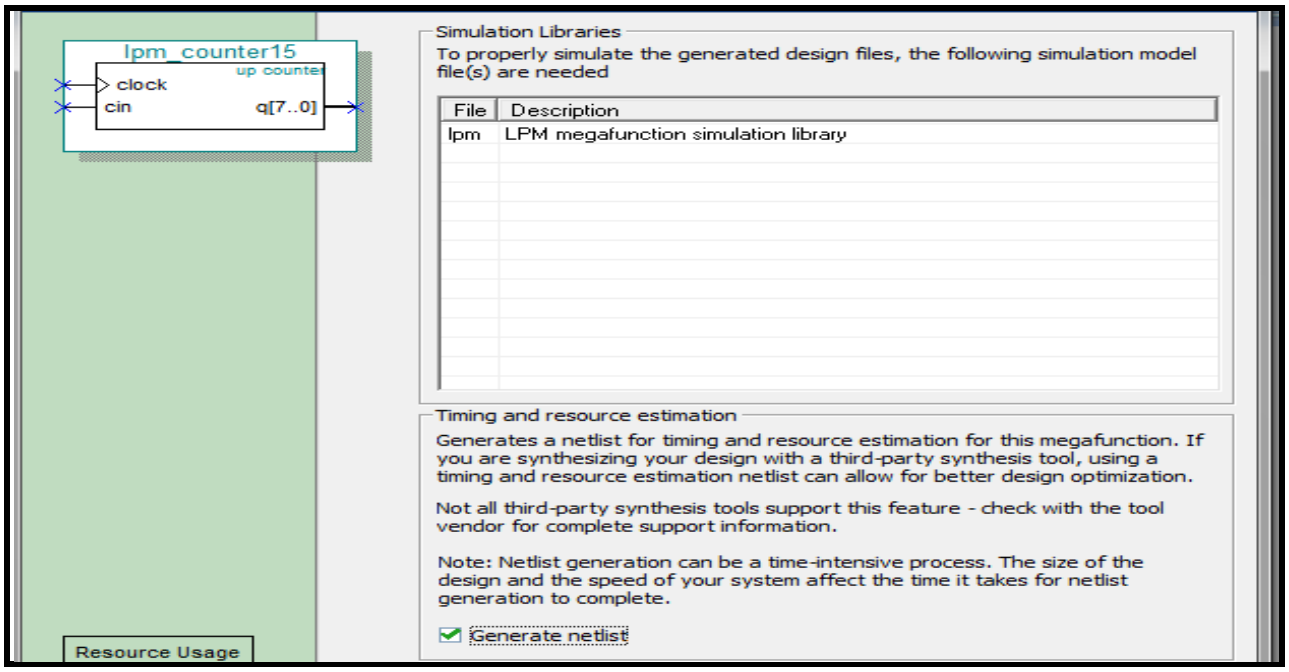
On clique sur **OK**



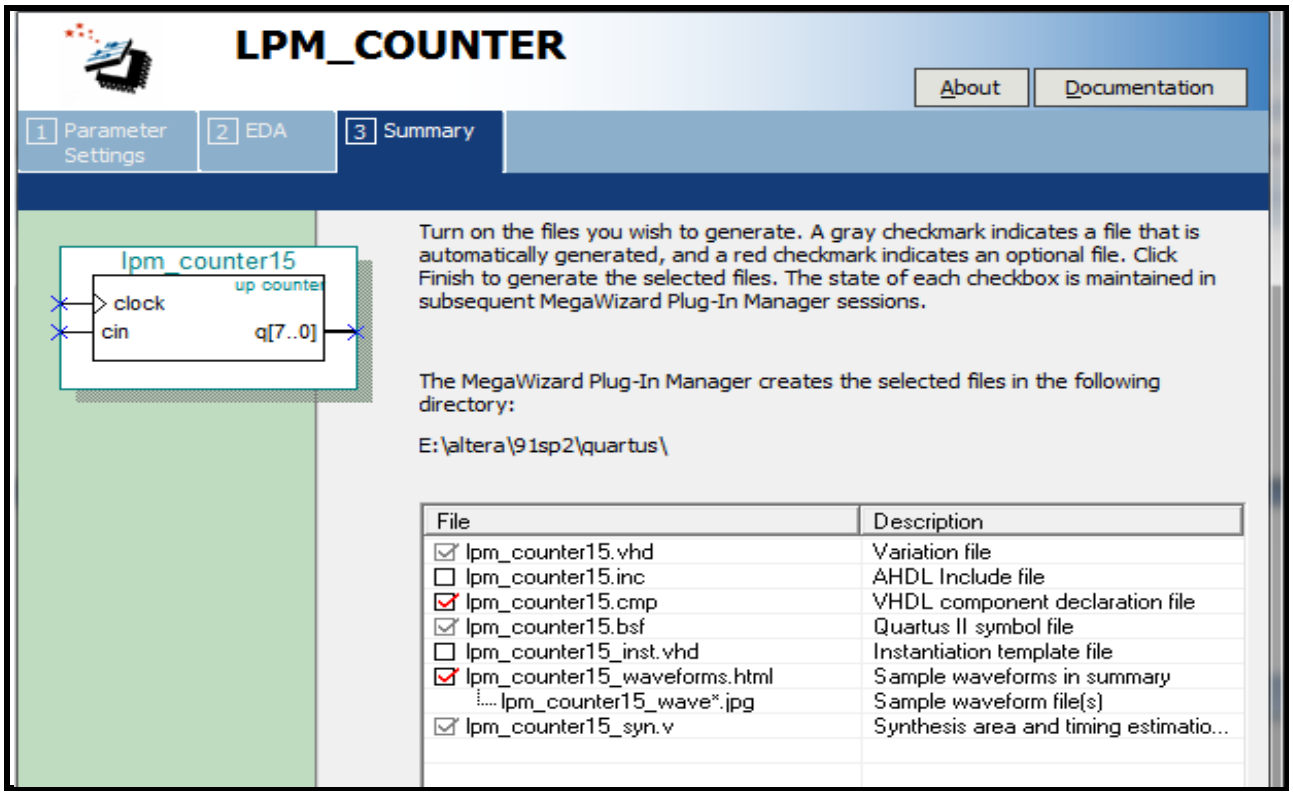
On clique sur **Next**



On clique sur **Next**

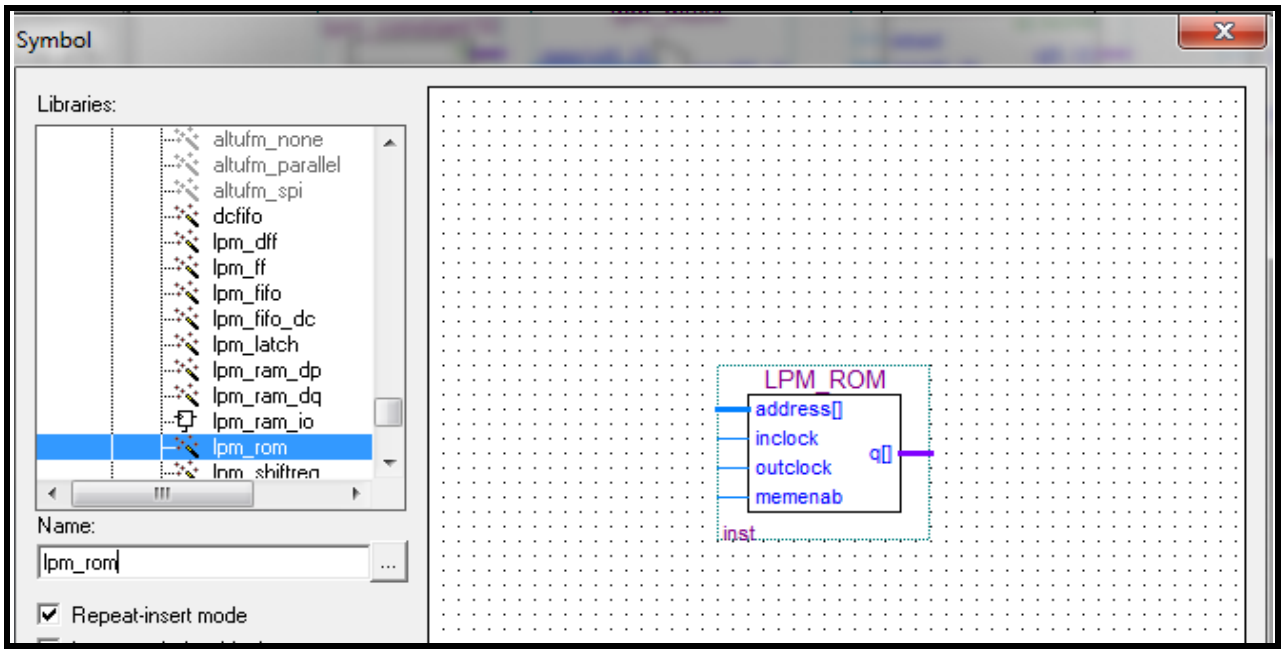


On clique sur **Next**

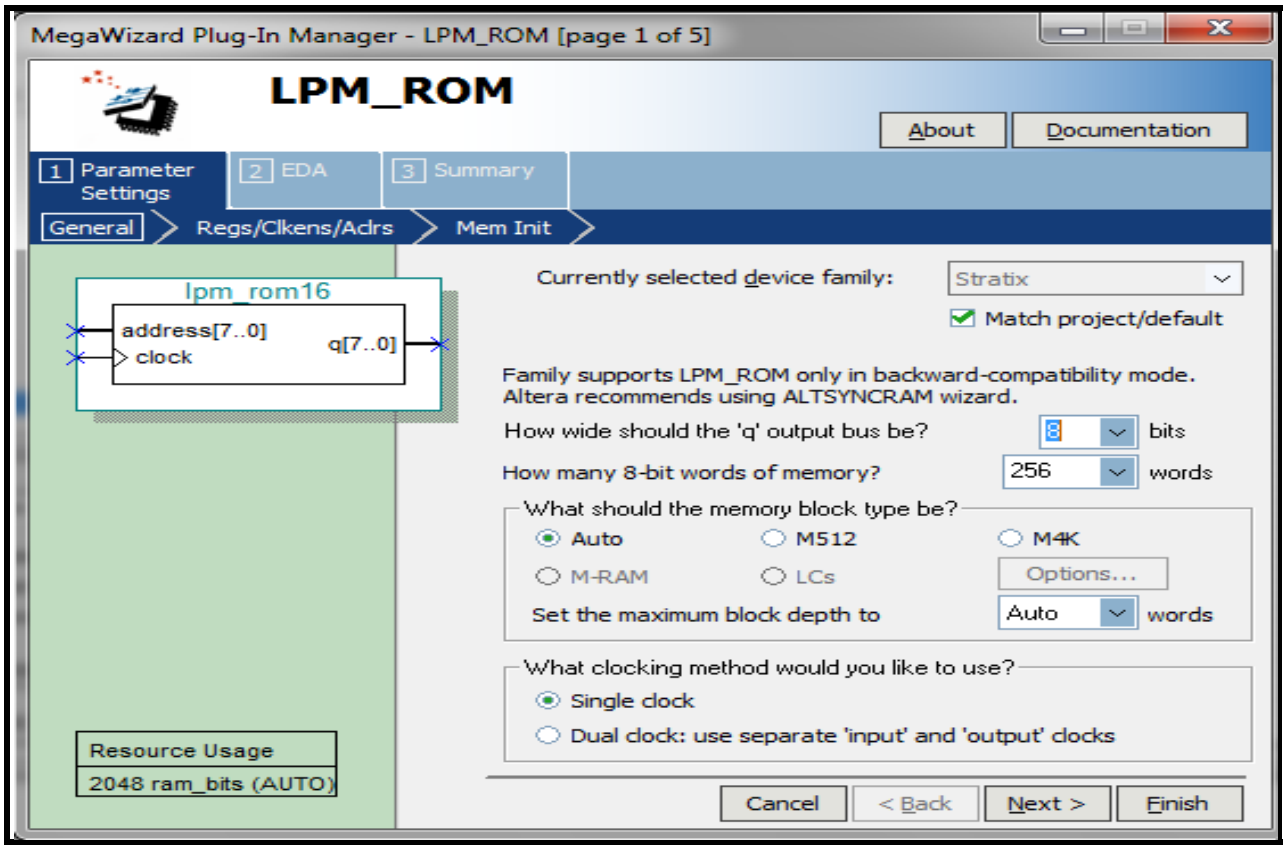


On clique sur **Finish**

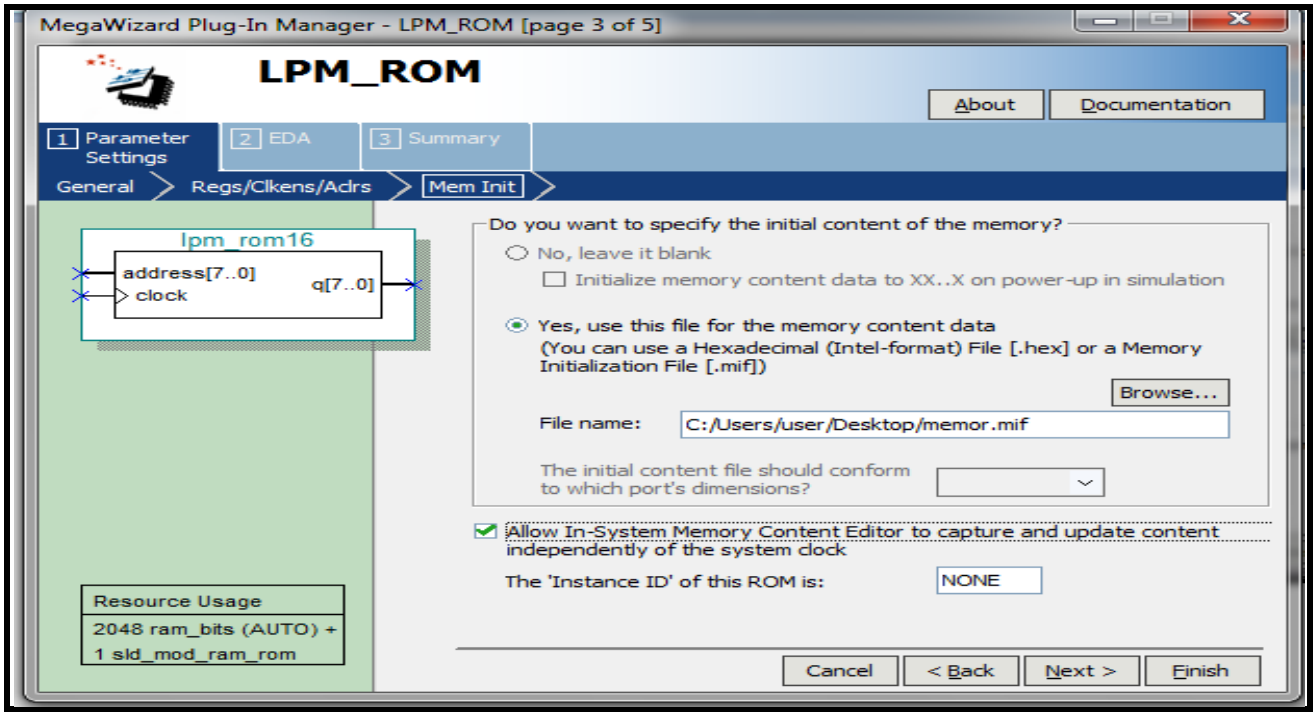
Pour charger la ROM on suit les étapes suivantes



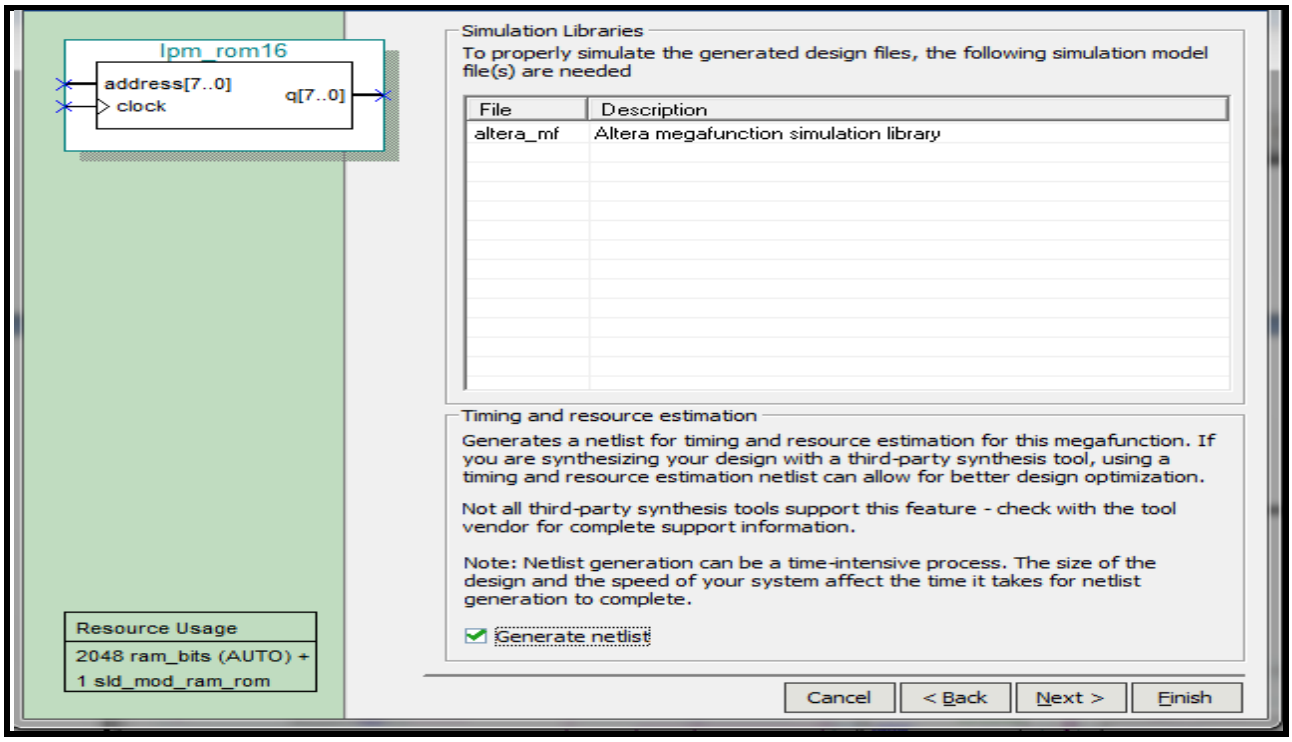
On clique sur **OK**



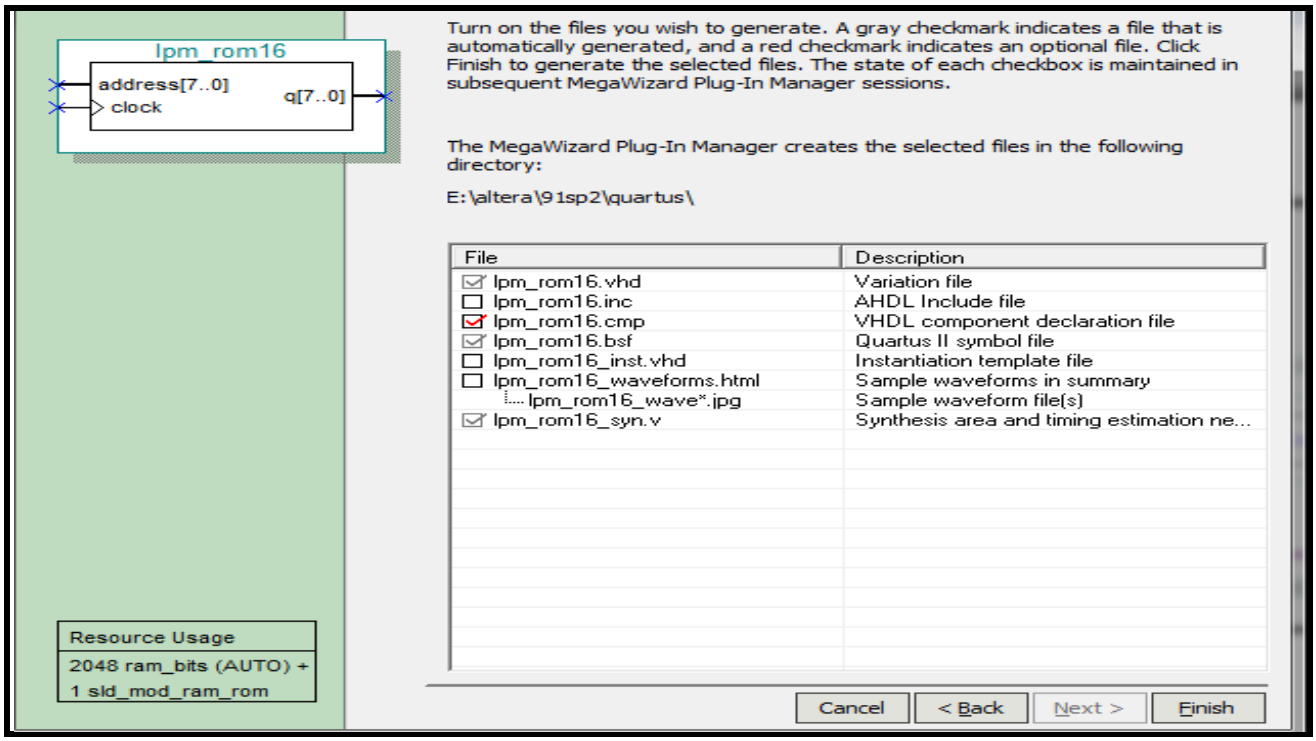
On clique sur **Next**



On clique sur **Next**

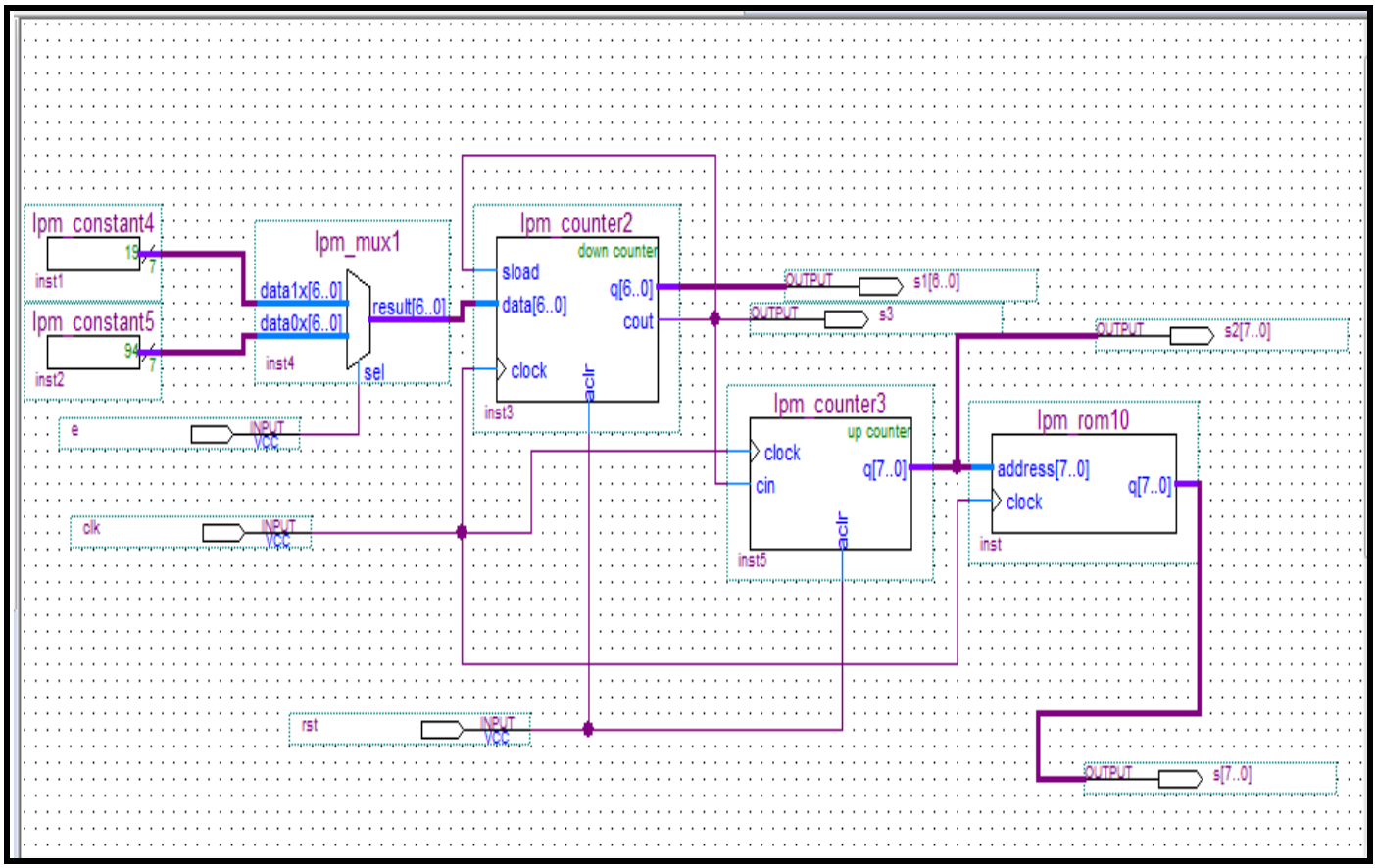


On clique sur **Next**

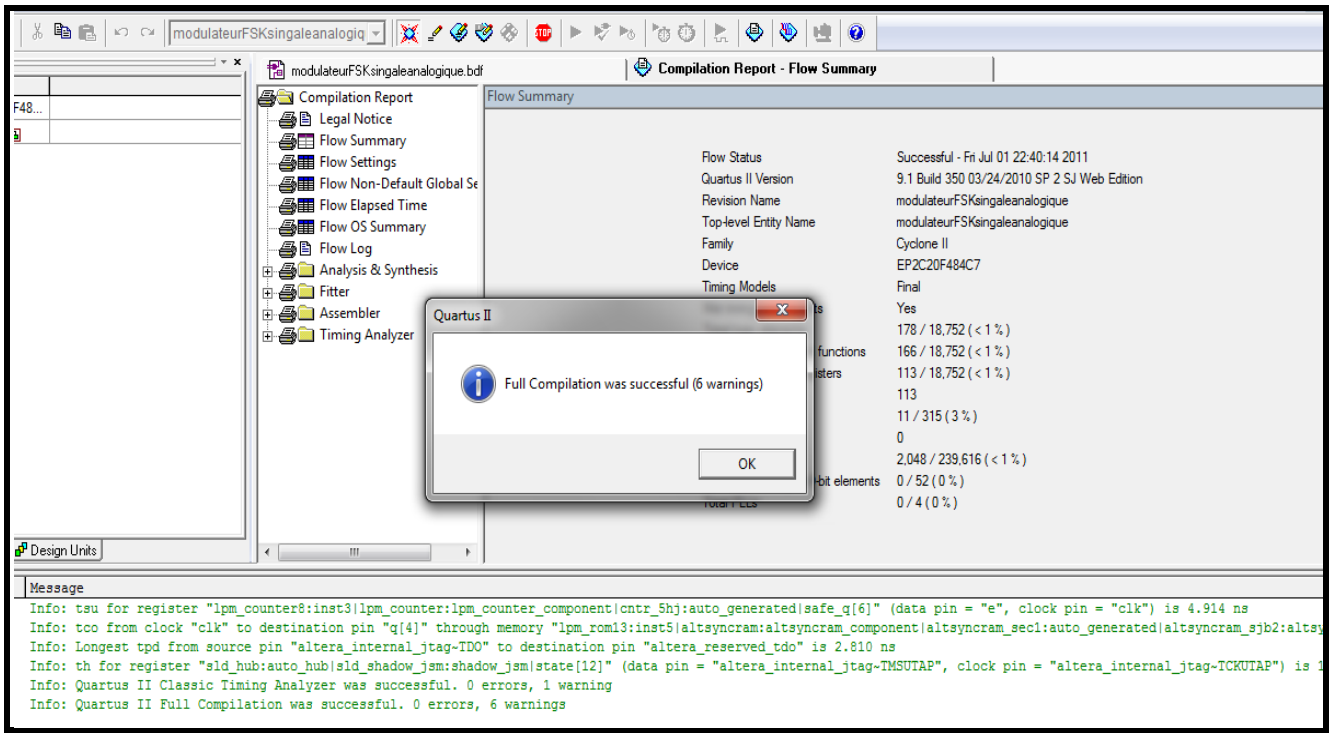


On clique sur **Finish**

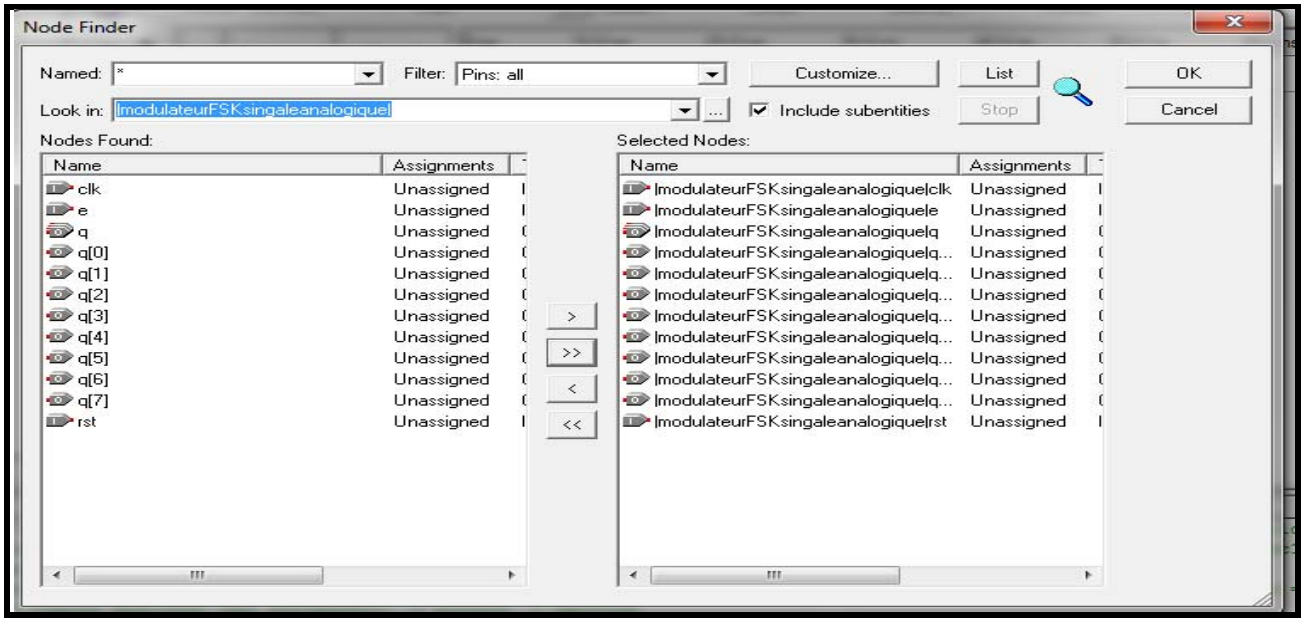
- enfin en relie les déférents blocs, et on obtient le schéma suivant :



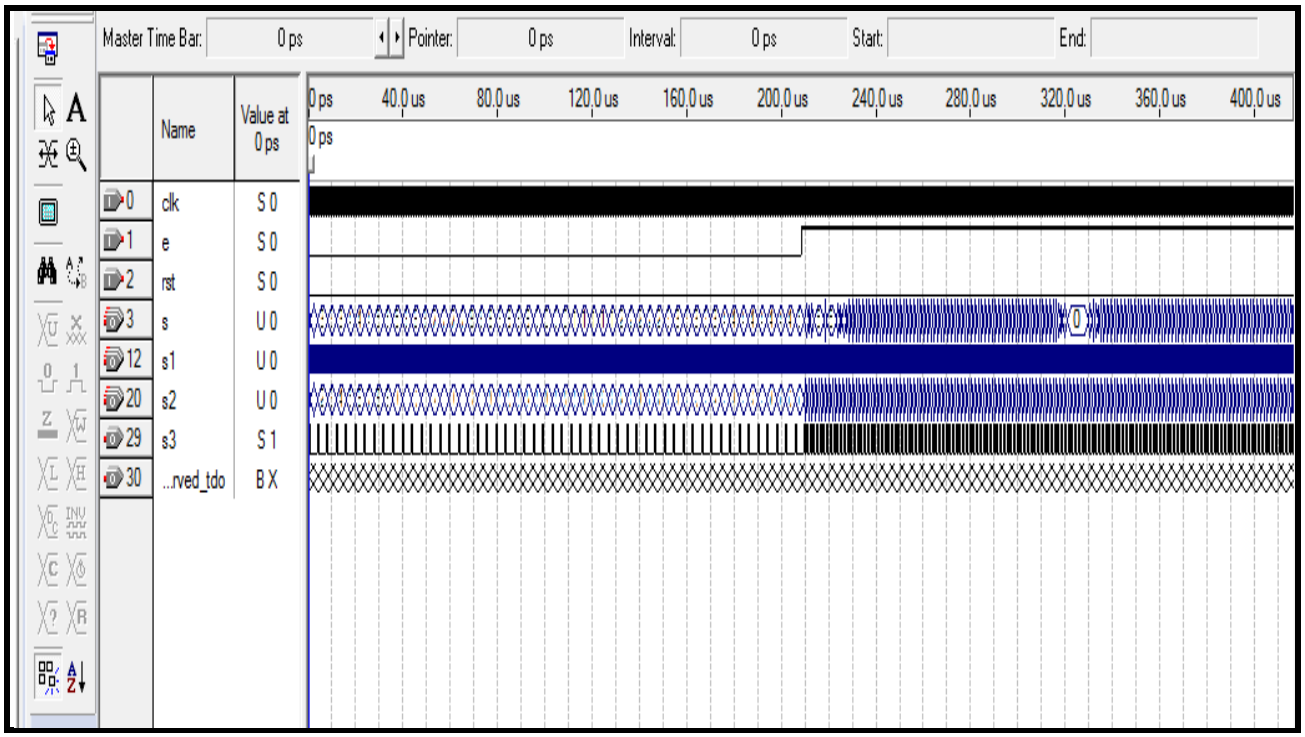
- Compilation de projet



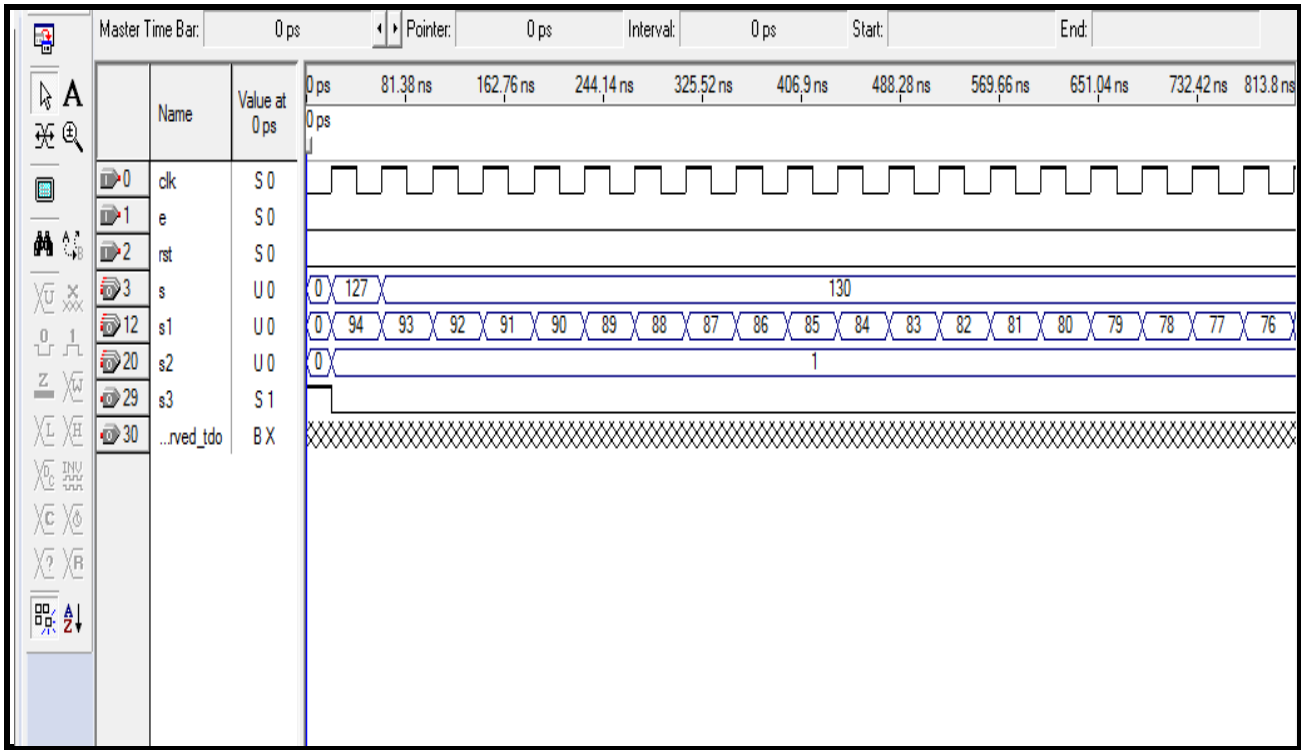
- La simulation est l'étape qui suit une compilation réussie. Elle consiste à construire des vecteurs de test (les entrées à appliquer au système), puis à lancer la simulation proprement dite. A la fin de celle-ci, le simulateur affiche la réponse du circuit aux vecteurs de test sous la forme de chronogrammes.



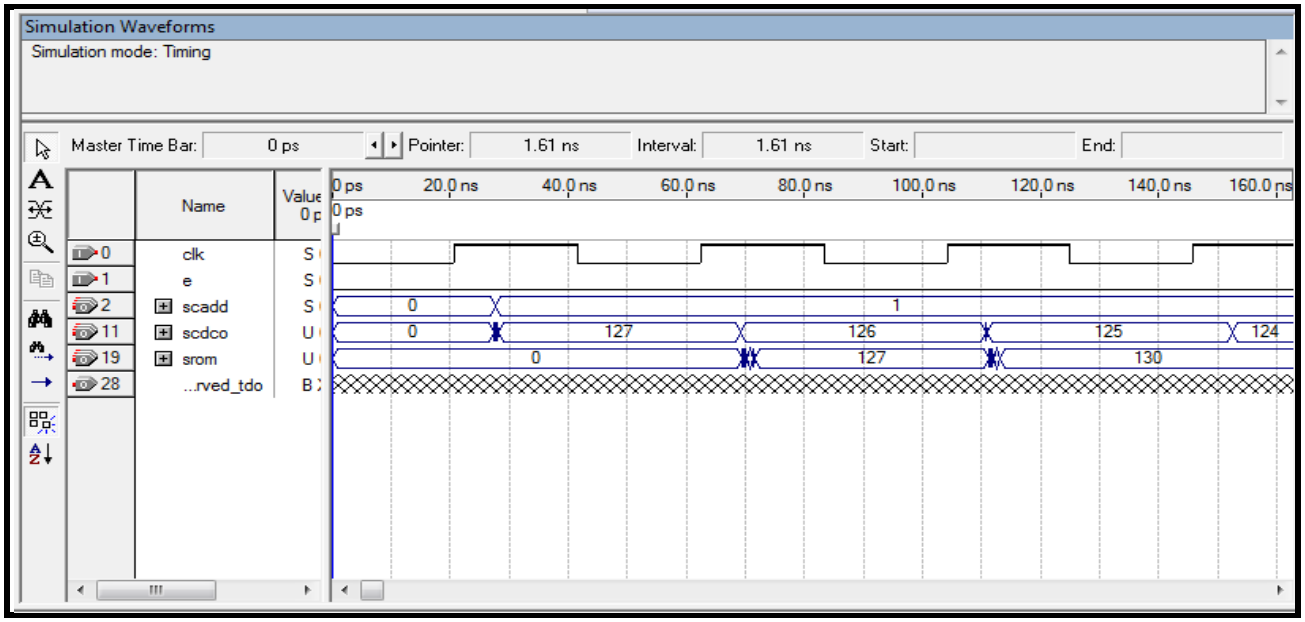
Nous donnons dans la figure suivante le résultat de la simulation fonctionnelle et le fichier test:



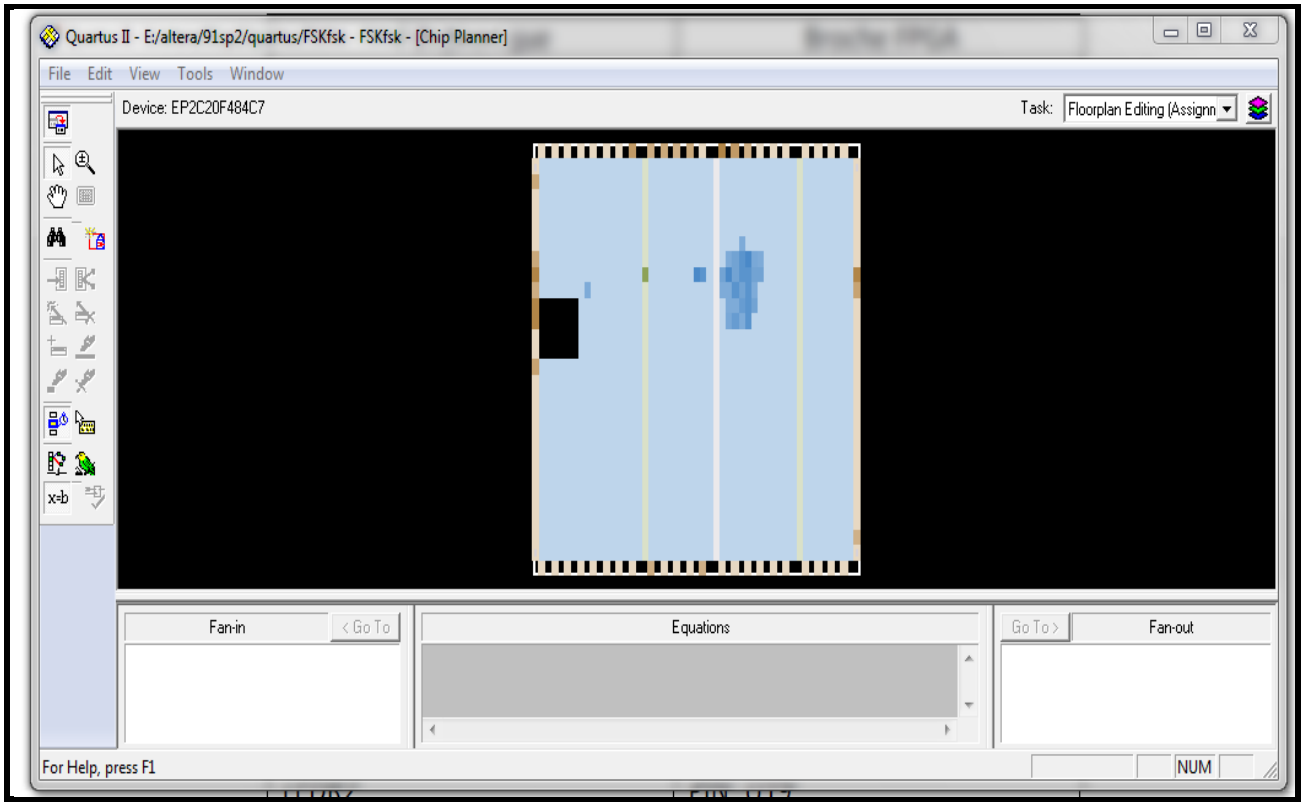
Le résultat de la simulation avec zoom.



Nous donnons dans la figure suivante le résultat de la simulation temporelle et le fichier test :



L'image suivante représente le schéma du modulateur FSK sur le circuit logique programmable cyclone II.



IV Conclusion

Dans ce chapitre nous avons adopté une approche expérimentale pour réaliser le modulateur FSK, les différents blocs on été conçu et valider par simulation sous QUARTUS. Une fois l'ensemble des blocs tester on a fait la connexion entre blocs, la réalisation proprement dit c'est fait sous environnement schématique vu la facilité offerte par le mode schématique. La modulation FSK à été simulé et les résultats de simulation sont conforme a nous prévision (modulation FSK).Tout les fichiers nécessaires au bon fonctionnement du modulateur on été générer a fin de l'implémenter sur la carte DE1.

CONCLUSION GENERALE

Conclusion générale

Au cours de ce projet, nous avons abordé et présenté les différents circuits logique programmable qui ont permis de renouveler les méthodes de conception des systèmes logique, on s'est intéressé particulièrement au circuit logique programmable type FPGAs qui est pour les concepteurs une évolution qui concilie désormais la possibilité d'intégration de fonctions complexes et performantes avec la souplesse et la flexibilité apportées par ce type de circuits.

Au cours de ce projet on a passé en revue les familles les plus répondues de circuit logique programmable (XILINX, ALTERA), puis on a implémenté un modulateur FSK simulé sous QUARTUS II sur la carte DE1 d'ALTERA dotée de l' FPGA Cyclone II.

Lors de la réalisation du projet, le mode schématique a été adopté vu les avantages qu'il procure au concepteur. Partant du modèle schématique des simulations comportementales, fonctionnelles et temporelles ont été réalisé avec succès. Les résultats en sortie sont en accord parfait avec la théorie, en effet les courbes en sortie sont bien celles d'une modulation FSK.

Au cours de ce projet nous avons fait une introduction aux dernières générations de circuits logique offerts par l'industrie micro-électronique et qui sont les FPGAs ainsi qu'aux outils de développement.

L'environnement de développement offert par la société ALTERA (QUARTUS II) été à la base de notre projet, la maitrise de cet outils nous a permis de mener à bien notre projet. En plus du côté théorique, qui rassemble le domaine des télécommunications et l'électronique numérique, on a acquis une assez bonne expérience dans le développement des circuits programmables dédies aux systèmes embarques.

Enfin, nous espérons que notre rapport de master servira de support didactique pour les générations futures voulant travailler sur les circuits logique programmable qui est un domaine en plein essor. Comme perspectives à notre travail, il serait très intéressant de réaliser un démodulateur et un système de transmission pour avoir un kit opérationnel.

BIBLIOGRAPHIE

[1] : Multiplieur en –ligne pour le calcule en longue précision sur un circuit FPGA : mémoire de fin d'étude du diplôme d'ingénieur en électronique, Tizi Ouzo (2002)

Mr D.AIT AZZI, Melle R .TIKOBAINI

[2] : logique programmable asynchrone pour systèmes embarqués sécurisés : thèse de doctorat l'institut polytechnique de Grenoble(2009)

Mr Taha Beyrouthy

[3] : Etude et Conception d'Architectures Haut-Débit pour la Modulation et la Démodulation Numériques : thèse de doctorat de L'Université Paul Verlaine – Metz (2006).

Mr Gérald ARNOULD

[4] : Implémentation d'un réseau de neurones d'un micro capteur sur un FPGA : mémoire de fin d'étude du diplôme de master en électronique (2010).

Mr MELLAL IDIR

[5] : Étude quantitative des techniques de partitionnement de réseaux de processeurs pour l'implantation sur circuits FPGA : thèse de doctorat DE L'UNIVERSITÉ DE RENNES(2002).

Mr STEVEN DERRIEN

[6] : La Boucle Locale Radio et la Démodulation directe de signaux larges bandes à 26GHz Thèse de Doctorat(2004)

M$^{\text{ême}}$ Sara ABOU CHAKRA (2004).

[7] : Etude et Implantation d'Algorithmes de Compression d'Images dans un Environnement Mixte Matériel et Logiciel : Thèse de Doctorat de L'UNIVERSITE BORDEAUX 1(2007)

Mr Ahmed BEN ATITALLAH.

[8] : Etude et intégration numérique d'un système multicapteurs amrc de télécommunication basé sur un prototype virtuel utilisant le langage de haut niveau vhdl-ams : Thèse de Doctorat de l'université de Toulouse II (2005).

M^{ême} Céline GUILLEMINOT

[9] : Commande numérique à base de composants FPGA d'une machine synchrone : Thèse de Doctorat de l'école nationale d'ingénieurs de Tunis et l université de Cergy pontoise (2007)

Mr MOHAMED WISSEM NAOUAR

[10] : Informatique industrielle : circuit logiques programmables. (IA 121)

NKETSA ALEXANDRE

[11] : logiques programmables : architecture des FPGA et CPLD. (1997) : IA17

DUTRIEUX LAURENT

[12]: Xilinx Company web: www.xilinx.com

[13]: Altera company web: www.altera.com

[14]: Modulation: www.dunod.com