

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد حطاب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Projet de Fin d'Études

présenté par

HaoucheFethi

&

Hamäidi Mohamed

pour l'obtention du diplôme master2 en Électronique option microélectronique microsystemes
et conception de circuits.

Thème

Implémentation sur FPGA d'un feu tricolore avec commande à distance

Proposé par :MmeI.KAOULA&Mme H.BOUGHERIRA

Année Universitaire 2012-2013

Dédicaces

Je dédie ce modeste travail à :

*La lumière de ma vie, le soleil de mon ciel, celle qui m'éclaire mère,
que dieu la garde et la protège.*

*Le pilier de ma vie, la lune de mes nuits, mon père, que dieu me le
garde et le protège.*

*Mon soutient dans la vie, mes chères frères : Ishak, yacoub, Youcef,
Brahim.*

Mes amies : Abdelmoumen, Abdelhak, ALI, Ahmed.

Et mes toutes camarades de 5ème année électronique.

A tous mes oncles, cousines et cousins.

A toute ma belle-famille sans exceptionnelle.

Tous ceux qui me sont chères.

Fethi.

Dédicaces

Mohamed.

Chapitre 4 Conception et implémentation

4.1 Introduction

Considéré comme l'un des meilleurs outils offerts sur le marché, le circuit FPGA (**Field Programmable Gate Array**) de Xilinx (voir annexe) offre une panoplie de composants, ainsi qu'un système de développement ISE (*Integrated Software Engineering*) *Fondation* offrant d'énormes possibilités de développement (synthèse, simulation et implémentation automatique).

Dans ce chapitre, il sera question de simulation et implémentation de l'architecture proposée.

La méthodologie de base pour la conception sur circuit FPGA de Xilinx se décompose en quatre étapes:

1. Description de l'architecture.
2. Simulation fonctionnelle.
3. Synthèse et implémentation.
4. Simulation temporelle et vérification.

L'architecture que nous avons réalisée a été conçue dans l'environnement ISE 9.1 de Xilinx. Le processus d'implémentation convoite certaines phases de vérification. En premier lieu, l'architecture doit être décrite en langage VHDL (Very High Description Language). Après la description VHDL de l'architecture et sa compilation, une simulation fonctionnelle est introduite par le simulateur (ISE simulator), dont les résultats figurent sous forme de chronogramme. Une fois que la simulation fonctionnelle est validée, une phase de synthèse va permettre de donner un rapport bien détaillé sur les ressources du circuit FPGA consommées, et sur la fréquence maximale de fonctionnement. Finalement, une phase de placement et routage éclate

schématiquement la surface consommée par l'architecture globale et configure les routages d'interconnexion entre les différents blocs des ressources du circuit FPGA. Dans la simulation temporelle, on vérifie si le circuit obtenu respecte les contraintes temporelles et utilise les délais des portes et les délais dans les interconnexions pour calculer la vitesse maximale. La figure 4.1 regroupe toutes les étapes d'une conception sur circuit FPGA.

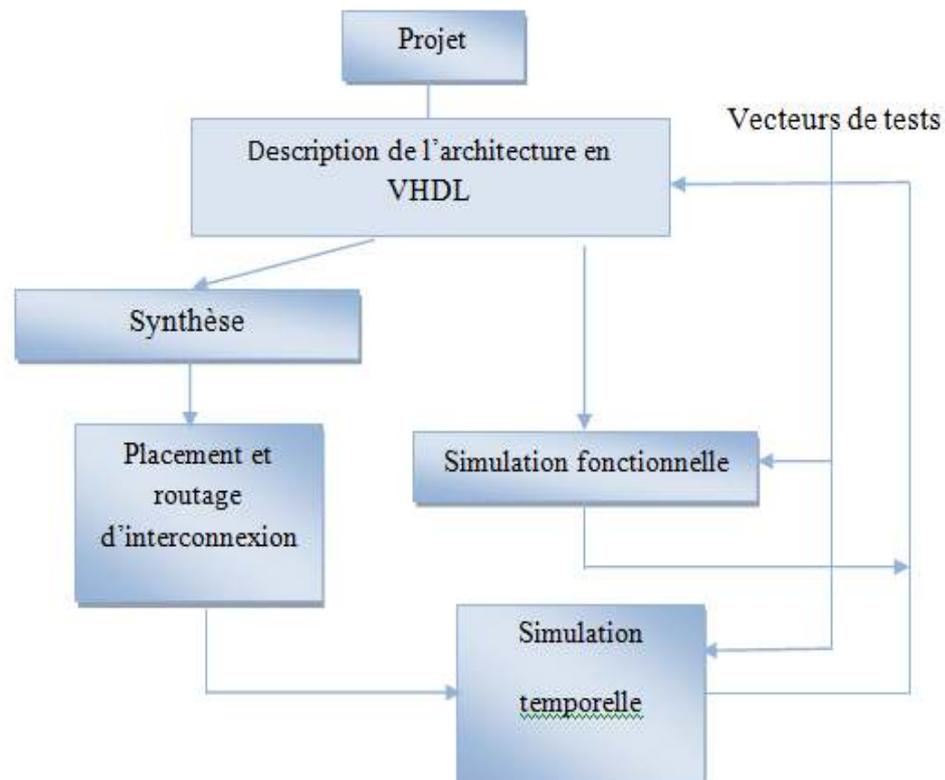


Figure 4.1. Schéma des étapes de conception sur un circuit FPGA.

4.2 Description de l'architecture globale

L'architecture que nous avons proposée est composée de trois blocs définis comme suit:

1. Bloc générateur d'horloge (diviseur).
2. Maintien du signal de télécommande.
3. Bloc commande logique (Le séquenceur).

L'architecture globale est montrée par la figure 4.2 :

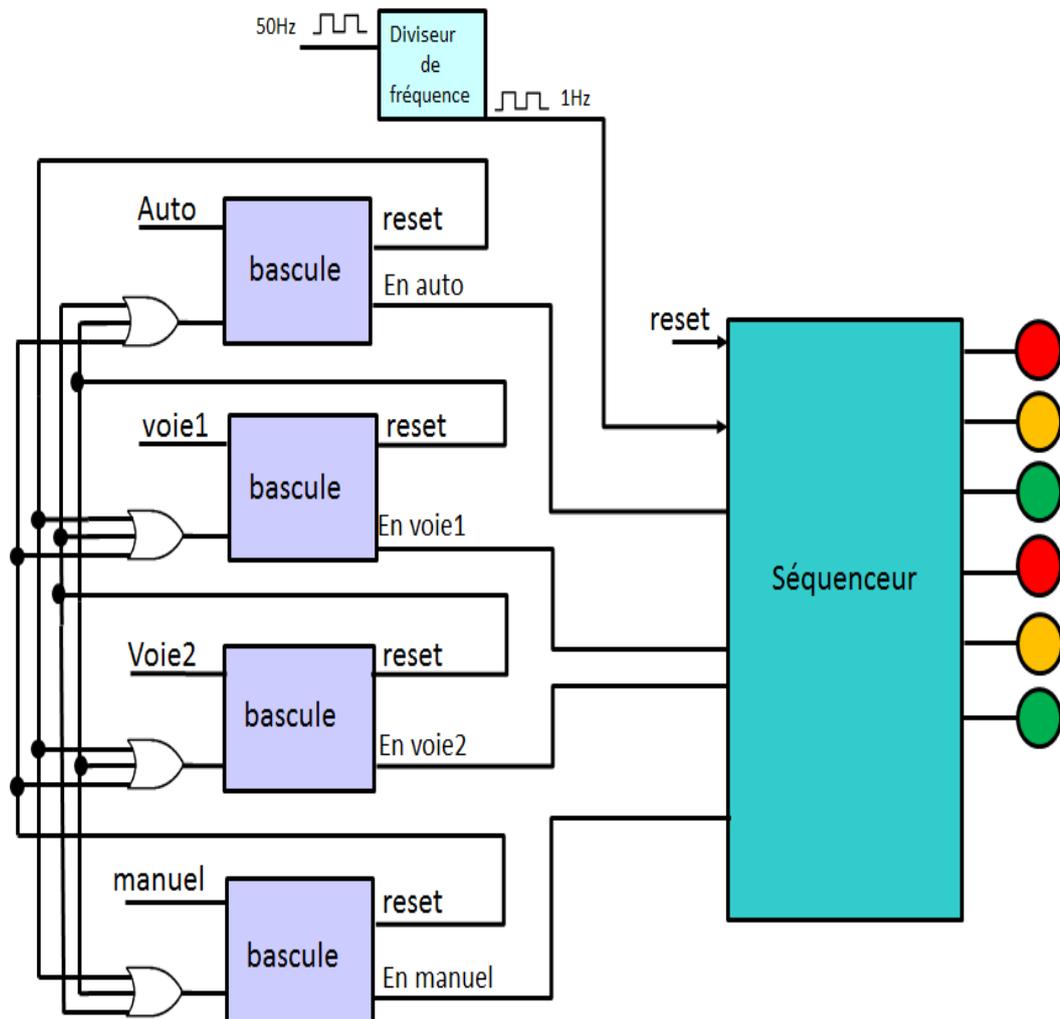


Figure 4.2. Architecture globale.

Le schéma de fonctionnement traduit de manière graphique la fonction d'usage de l'objet technique en tenant compte des solutions techniques et technologiques retenues pour réaliser l'objet.

4.2.1 Bloc générateur d'horloge

Le bloc générateur d'horloge est le premier bloc de notre conception sous FPGA. C'est un bloc qui reçoit à son entrée le signal *reset* et l'horloge principal de la carte. En sortie, il génère un signal *freq* qui sera utilisé dans le bloc séquenceur.

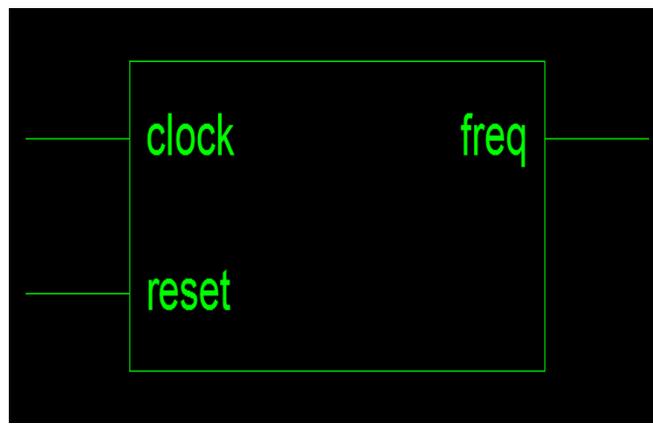


Figure 4.3. Architecture du Bloc Diviseur de fréquence.

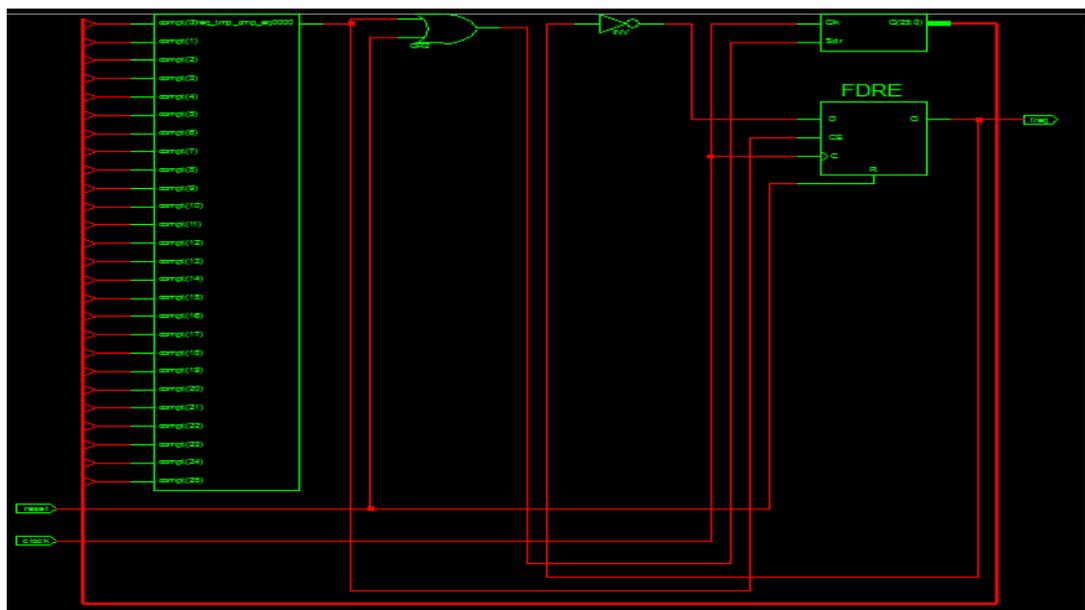


Figure 4.4. Architecture interne du Bloc Diviseur de fréquence.

4.2.2 Sauvegarde du signal de télécommande

C'est le bloc de la sélection des modes (auto, voie1, voie2 ou manuel) qui nécessite une intervention externe à travers un bouton poussoir via une télécommande RF.

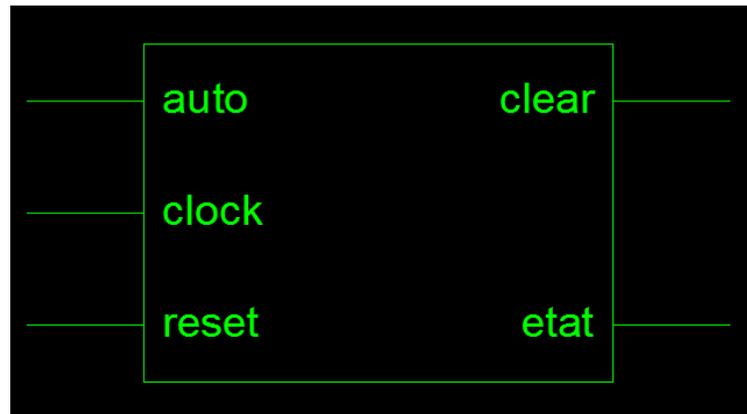


Figure 4.5. Architecture du Bloc Sauvegarde.

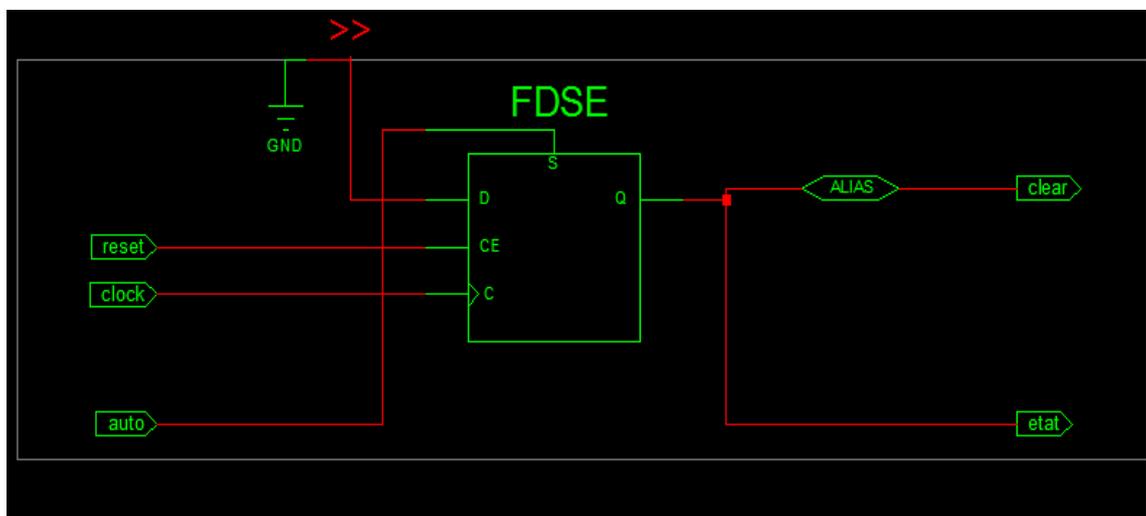


Figure 4.6. Architecture interne du Bloc Sauvegarde.

4.2.3 Bloc commande logique (Le séquenceur)

Ce bloc est le séquenceur qui consiste à générer les différents signaux de contrôle des feux (orange, rouge et vert) des deux voies 1 et 2.

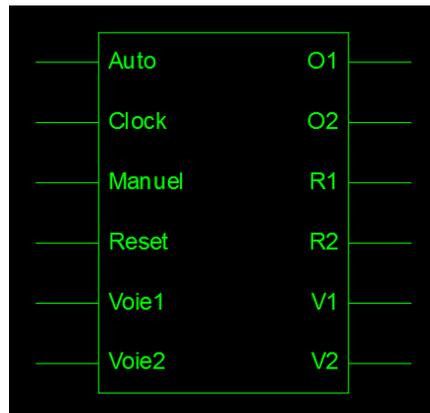


Figure 4.7. Architecteur du bloc commande logique (Le séquenceur)

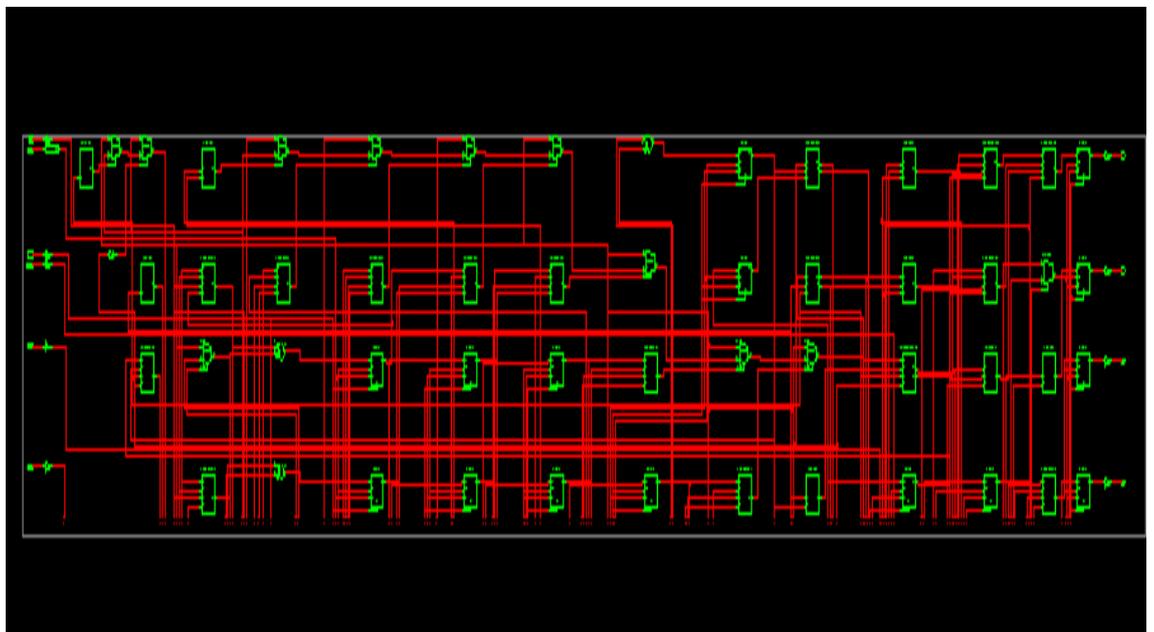


Figure 4.8. Architecteur interne du bloc commande logique (Le séquenceur).

4.3 Simulations et Résultats

Les étapes suivies pour la simulation de l'implémentation de feux tricolores se résument comme suit:

- Description de l'architecture en langage VHDL sous l'outil ISE 9.1 et la simulation par l'outil ISE simulator.
- Synthèse de l'architecture.
- Implémentation de l'architecture sur le circuit XC3S250E_4FT256 Cde la famille Spartan3E.

Cette partie nous permet de vérifier le bon déroulement de notre système avant de le charger dans la carte FPGA. Nous pouvons détecter, de cette manière-là toutes les erreurs qui auraient pu se glisser soit dans notre code soit dans notre saisie de schémas.

Une deuxième étape sera de simuler notre système dans le domaine temporel pour tenir compte de la réalité des composants.

4.3.1 Résultat de la simulation du Blocs générateur d'horloge (diviseur).

Le résultat de la simulation fonctionnelle du Bloc générateur d'horloge est représenté sur la figure 4.9

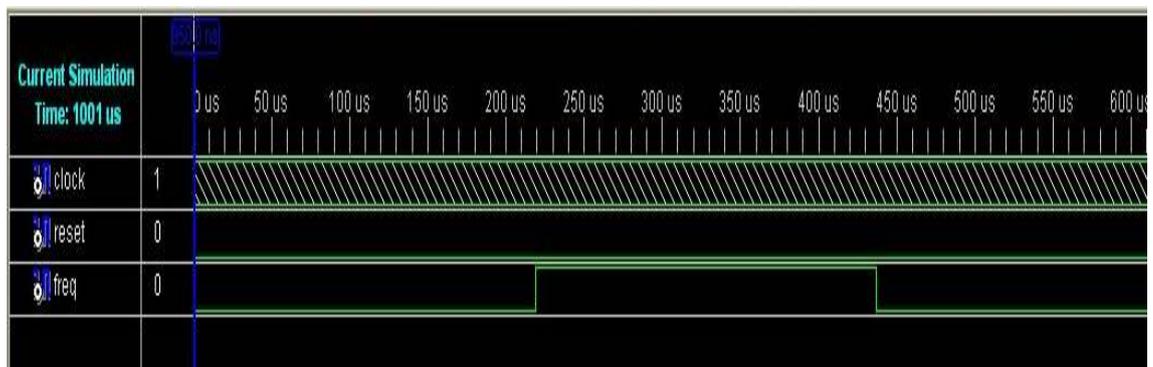


Figure 4.9: Chronogramme de bloc générateur d'horloge .

4.3.2 Résultats de la simulation du Bloc Maintient du signal de télécommande.

Le résultat de la simulation fonctionnelle du Bloc Maintientdu signal de télécommandeest représenté sur la figure 4.10.

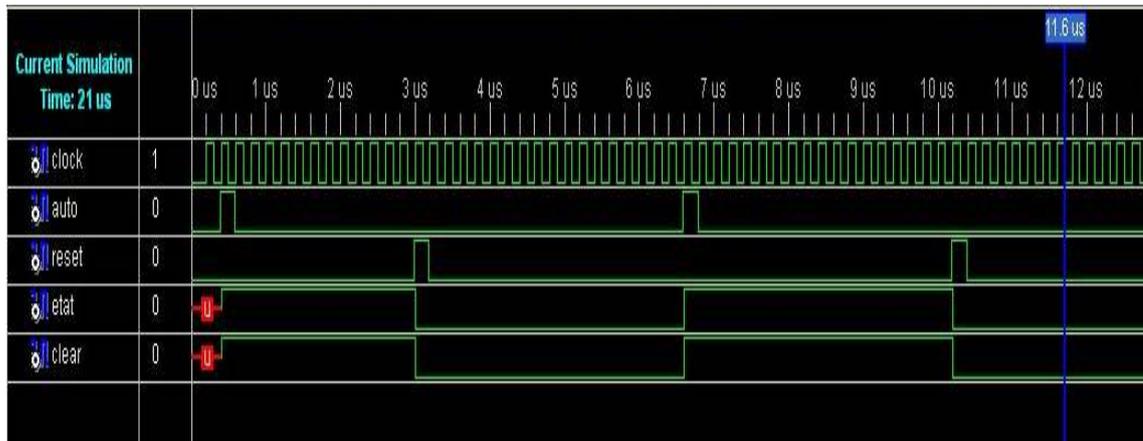


Figure 4.10.Chronogramme du Bloc Maintient du signal.

4.3.3 Résultats de la Simulation du séquenceur

Le résultat de la simulation fonctionnelle du Bloc séquenceur est représenté sur la figure 4.11.

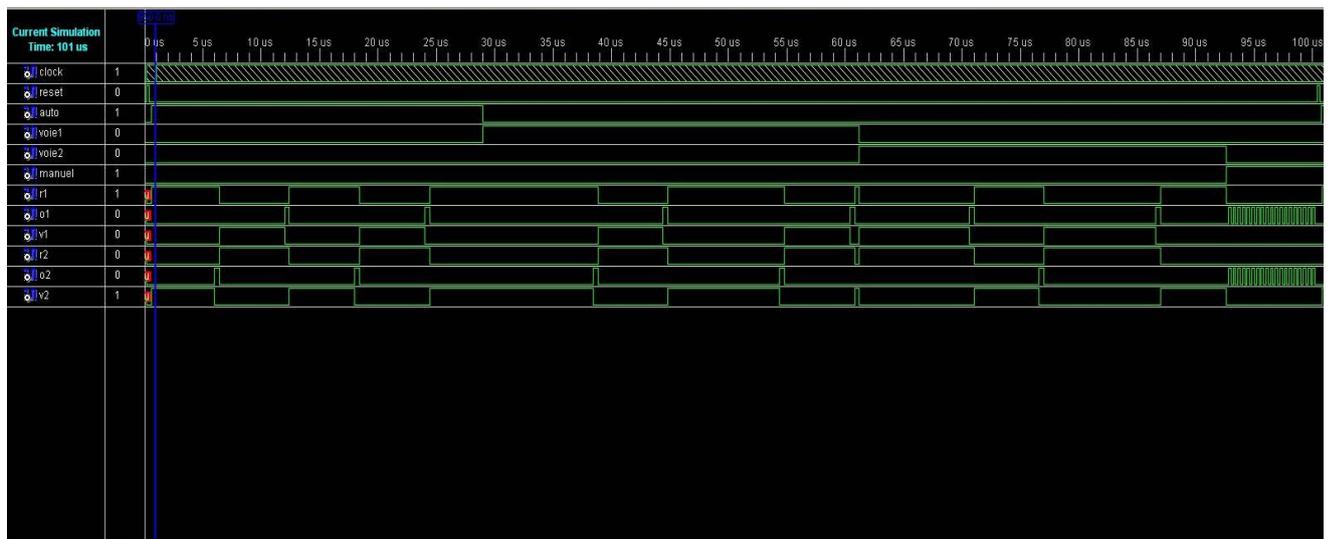


Figure 4.11Chronogramme du séquenceur.

4.3.3 Résultats de la Simulation finale

Le résultat de la simulation fonctionnelle du Bloc finale est représenté sur la figure 4.12 :

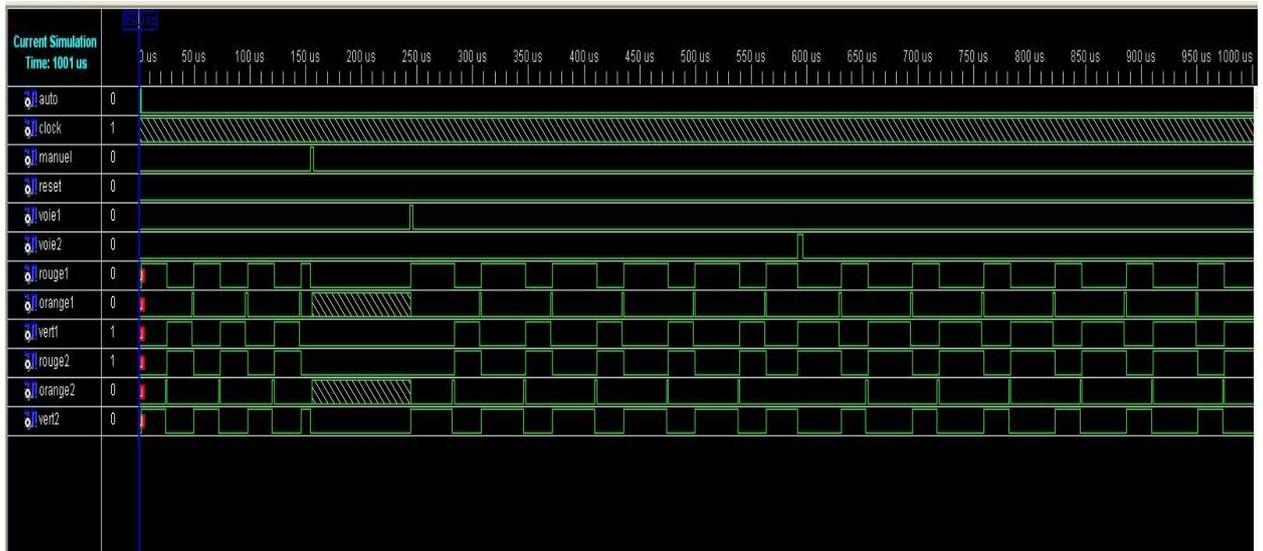


Figure 4.12 Chronogramme finale.

4.4 Simulation sous TINA

Après avoir simulé notre programme sous ISE, on a vérifié le bon fonctionnement de notre système en simulant ce dernier sous TINA. La figure suivante représente le résultat de cette simulation.

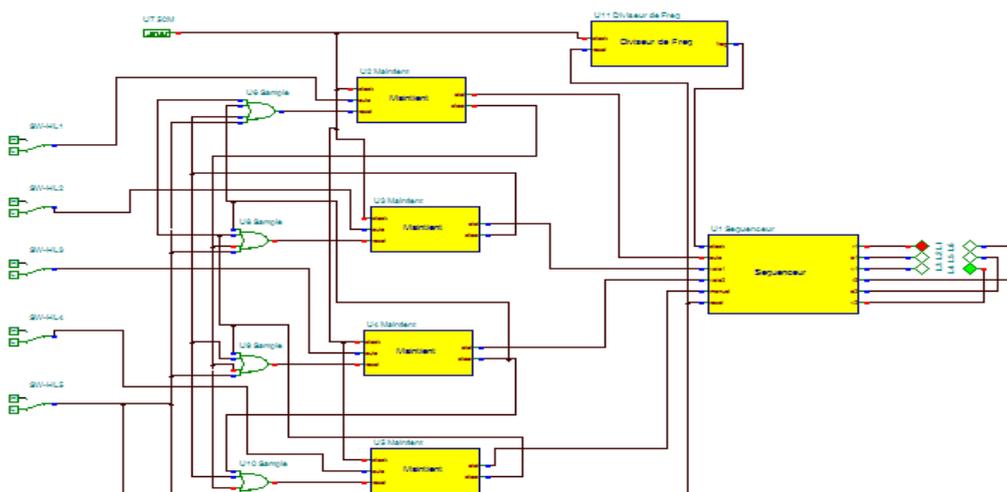


Figure 4.13 Simulation sous TINA

4.5 Synthèse et implémentation

✓ Synthèse

Cette étape est une synthèse des différents blocs de l'architecture globale. Ils sont exécutés par l'outil de synthèse XST et permet d'avoir une estimation du taux d'occupation et du temps d'exécution. Ensuite, nous sommes passés à l'implémentation qui est la dernière étape dans la conception et qui permet le placement et le routage.

La figure suivante présente le Layout de l'architecture du système.

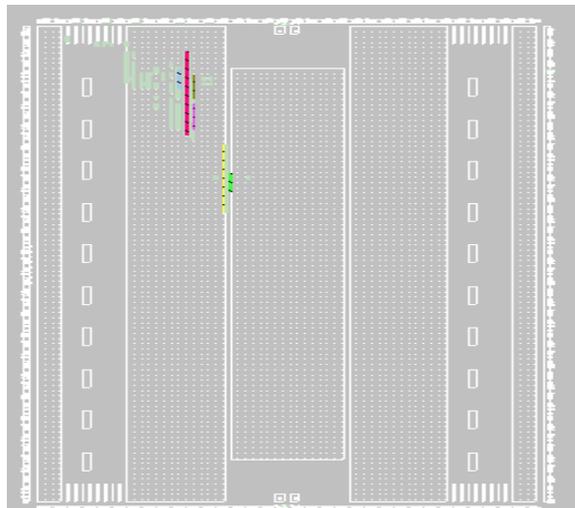


Figure 4.14 Layout de l'architecture du système

✓ Implémentation

Pour implémenter un circuit sur un FPGA, il y a trois éléments indispensables pour parvenir :

- Disposer de la description logique du circuit (schématique, diagrammes d'états ou code tel VHDL).
- L'environnement de développement est choisi en fonction du composant sur lequel le circuit sera implémenté spécialement le logiciel du fabricant du composant (ISE :XILINX Integrated Software Environment).

- Une carte de développement pour implémenter le circuit et vérifier son comportement.

Le produit final est représenté dans la figure 4.14, on a la carte spartan3e, la télécommande RF « pour choisir les modes », La carte de réception, et les panneaux des feux.



Figure 4.15 système de contrôle des feux à distance

L'architecture proposée a été implémentée avec succès, ainsi que les blocs d'émission, réception, comme le montre la figure suivante.

4.6 Conclusion

Dans ce chapitre, nous avons présenté la méthodologie de conception de notre architecture. Nous avons également présenté, les résultats obtenus de la simulation et de l'implémentation sur le circuit FPGA XC3S250E_4FT256C de la familleSpartan3E. Pour cela, nous sommes passés par plusieurs étapes, chacune d'entre elles utilise un outil bien spécifique.

Résumé

Ce travail a pour but d'implémenter sur un circuit FPGA sous l'environnement Xilinx ISE 10.1, un système de contrôle de feux tricolores commandé à distance par une télécommande via radio fréquence, cette dernière nous permet d'avoir quatre modes différents d'utilisation (mode automatique, mode manuel, priorité pour la voie1, priorité pour la voie2).

Mots clés : FPGA, feux tricolores, commande à distance via RF.

Abstrcat

The purpose of this work is to implement on FPGA in the Xilinx ISE 10.1 environment, a control system of traffic lights controlled by a remote control via radio frequency, it allows us to have four different modes of operation (automatic, manual mode, priority for channel 1, channel 2 for priority).

Keys words: FPGA, traffic light, RF remote control.

ملخص

الهدف الرئيسي من هذا العمل هو هو زرع في الدارة المنطقية المبرمجة (FPGA Field Programmable Gate Array) على Xilinx ISE 10.1, نظام التحكم عن بعد في إشارة المرور عن طريق الترددات ألاسلكية الذي يمكننا من الحصول على اربعة انماط مختلفة (اوتوماتيكي, يدوي, طريق1, طريق2).

كلمات المفاتيح, FPGA, أضواء الالوان الثلاثة, الترددات الراديوية

Introduction générale

Circuler entre les endroits de la ville est devenu une tâche difficile en raison des embouteillages. Le trafic congestionné est un problème majeur dans de nombreuses villes modernes dans le monde entier. Car l'augmentation de voitures est proportionnelle avec l'augmentation des populations. Par conséquent, le trafic congestionné se développe. Les embouteillages entraînent une perte de temps et d'énergie pour les gens, et provoquent la tension nerveuse.

La plupart des contrôleurs de feux de circulation sont à cycle fixe, dans lequel tous les feux de signalisation alternatifs obtiennent un intervalle de temps particulier pour être vert. [1]

Donc un système de commande à distance de feux de signalisation de temps fixe, peut être la solution optimale pour le trafic urbain.

La plupart des systèmes de feux de circulation fonctionnent en fonction du temps propre à chaque signalisation lumineuse. Un système détecte la présence ou l'absence de véhicules et réagit en conséquence.

La mise en place de stratégies de commande des feux de signalisation constitue l'épine dorsale de la gestion du trafic urbain. Ces stratégies reposent principalement sur l'estimation des temps d'attente aux intersections. Ce paramètre permet d'évaluer la qualité de service des intersections à feux.

Un carrefour routier, devenu dangereux par l'accroissement constant du trafic urbain, doit être équipé d'un système à fonctionnement automatique ou manuel permettant de réguler le débit de la circulation à distance par radio fréquence. Cette évolution de

l'automatisation est due à une large part ,au séquenceur ,ce dispositif technologique automatise rationnellement le fonctionnement des machines, limitant l' intervention humaine à des tâches de surveillance ou de maintenance du processus.

La partie essentielle dans tout les automatismes est bien la partie commande qui synchronise en quelque sorte, les mouvements à réaliser en établissant le cycle de fonctionnement. C'est un système qui élabore en sortie des ordres destinés au processus et des signaux de visualisation en fonction du contrôle du processus. Notre projet de fin d'étude, consiste à étudier et réaliser un système automatique permettant la régulation de la circulation des véhicules sur un croisement de deux voix. Le système de régulation que nous vous proposant possède une philosophie de fonctionnement totalement différente des systèmes électroniques. C'est une commande de feux de croisement ou feux routiers par radio fréquence et implémentation sur FPGA.

Ce mémoire intitulé «Implémentation sur FPGA d'un feu tricolore avec commande à distance » se divise en quatre chapitres organisés de la manière qui suit:

Dans le premier chapitre, nous commençons par une étude des principales techniques de gestion de feux tricolores ainsi que les différentes parties du projet et leurs fonctionnalités puis le fonctionnement du schéma global qui constituent un système de gestion de feux tricolores commandé à distance par radio fréquence et implémentation sur FPGA.

Dans le deuxième chapitre, nous présentons l'étude et la conception RF (radio fréquence) qui permet de gérer les feux tricolore a distance, les techniques radio fréquence paramétrique et leur avantages, par exemple, leur insensibilité à la position, ainsi que leur facilité de commande automatiquement.

Le troisième chapitre, se divise en deux parties, la première traite des circuits logique programmables selon leur ordre chronologique d'évènement jusqu'à l'invention des FPGA. la deuxième partie, traite l'outil indispensable pour la programmation d'un FPGA, les langages HDL (Hard Ware Description Langages) puis le travail est orienté en particulier vers le langage VHDL (Very High Speed Integrated Circuits Hardware Description Language).

Dans le dernier chapitre, nous commençons par la présentation de l'outil ISE 10.1 de Xilinx utilisé pour l'implémentation de notre architecture. Les résultats de simulation et de synthèse des différents blocs de notre architecture ainsi que les résultats de routage sur le circuit FPGA sont donnés à la fin du chapitre.

Nous terminons ce mémoire par une conclusion illustrant les perspectives de recherche et développement de systèmes en instrumentation ayant pour but l'amélioration des performances et l'augmentation de l'efficacité d'un système de gestion d'un feu routier. Des perspectives qui sont le fruit de la recherche bibliographique et de l'application des connaissances acquises dans le domaine de développement des systèmes électroniques.

Table des matières

Introduction Générale

Chapitre 1 Généralité de feux tricolore

1.1 Introduction.....	Erreur ! Signet non défini.
1.2 Les feux tricolores.....	1
1.3 Configuration topographique.....	2
1.3.2 Types de contrôleurs.....	4
a Les contrôleurs à logique programmable.....	4
b Les contrôleurs électromécaniques	Erreur ! Signet non défini.
c Les contrôleurs à logique câblée	Erreur ! Signet non défini.
d Les contrôleurs à microprocesseur	5
1.3.3 Modes de fonctionnement des contrôleurs	5
a Mode manuel.....	5
b Mode automatique.....	Erreur ! Signet non défini.
c Mode adaptatif.....	Erreur ! Signet non défini.
d Mode coordonné	6
1.4 Machine d'états	Erreur ! Signet non défini.
1.5 Conclusion	Erreur ! Signet non défini.2

Chapitre 2 Présentation globale des système de commande du feux tricolore

2.1 Introduction.....	13
2.2 Généralité sur la radio fréquence.....	14
2.3 Structureur d'émetteur et récepteur	Erreur ! Signet non défini.
2.3.1 Carte del'émetteur.....	15
2.3.2 Le récepteur	Erreur ! Signet non défini.
2.3.3 Circuit du système de commande.....	17
2.3.4 Carte de puissance.....	19
2.4 Conclusion.....	21

Chapitre 3 Technologie des circuits FPGA

3.1 Introduction.....	22
3.2 Description d'un FPGA (Field Programmable Gate Array)	23
3.2.1 PLD.....	24
3.2.2 CPLD.....	25

3.2.3	FPGAs.....	26
3.2.4	Familles des FPGA de Xilinx.....	26
3.2.5	Nomenclature des circuits FPGA.....	27
3.3	Langage de description matériel VHDL	28
3.3.1	VHDL.....	28
3.3.2	Structure d'une description VHDL simple	29
3.4	Etapes nécessaires au développement d'un projet sur FPGA.....	30
3.4.1	Saisie du texte VHDL.....	Erreur ! Signet non défini.
3.4.2	Vérification des erreurs.....	Erreur ! Signet non défini.
3.4.3	Simulation.....	Erreur ! Signet non défini.
3.4.4	Synthèse	Erreur ! Signet non défini.
3.4.5	Optimisation, placement et routage.....	Erreur ! Signet non défini.
3.4.6	Programmation du composant et test.....	33
3.5	Les étapes de conception d'application sur FPGA	34
3.6	Conclusion	36

Chapitre 4 Conception et implémentation

4.1	Introduction.....	37
4.2	Description de l'architecture globale	38
4.2.1	Bloc générateur d'horloge	41
4.2.2	Bloc Maintient	Erreur ! Signet non défini.
4.2.3	Bloc commande logique (Le séquenceur).....	Erreur ! Signet non défini.
4.3	Résultats de la simulation.....	Erreur ! Signet non défini.
4.3.1	Résultat de la simulation du Blocs générateur d'horloge...	Erreur ! Signet non défini.
4.3.2	Résultats de la simulation du Bloc Maintient	Erreur ! Signet non défini.
4.3.3	Résultats de la Simulation du séquenceur	47
4.3.4	Synthèse et implémentation.....	48
4.4	Conclusion.....	49

Conclusion général

Annexe

Bibliographie

Liste des figures

Chapitre1 : Généralité de feux tricolore

Figure 1.1. Les 16 mouvements possibles sur un carrefour à quatre branches

Figure 1.2. Bloc machine états

Figure 1.3. Machine d'états du mode 1

Figure 1.4. Machine d'états du mode 2

Figure 1.5. Machine d'états du mode 3

Figure 1.6. Machine d'états du mode 4

Chapitre 2 : Présentation globale du système de commande du feu tricolore

Figure 2.1. Récepteur et émetteur RF

Figure 2.2. Schéma synoptique du système de commande d'un feu tricolore

Figure 2.3. Dispositif Émetteur

Figure 2.4. Récepteur du feu tricolore

Figure 2.5. Feux tricolores de croisement

Figure 2.6. Schéma synoptique de diviseur

Figure 2.7. Schéma synoptique de Sauvegarde

Figure 2.8. Représentation temporelle

Figure 2.9. Amplificateur de tension

Figure 2.10. Ligne du circuit de la carte de puissance

Chapitre 3 : Technologie des circuits FPGA.

Figure 3.1. Classification des circuits numériques

Figure 3.2. Architecture des CPLD

Figure 3.3. Architecture générale d'un FPGA

Figure3.4. Nomenclature d'un FPGA de XILINX

Figure3.5. Organisation fonctionnelle de développement d'un projet sur circuit FPGA

Figure3.6. Etapes de développement d'un projet sur circuit FPGA

Chapitre4 : Conception et implémentation

Figure 4.1. Schéma des étapes de conception

Figure 4.2. Architecture globale

Figure 4.3. Architecture du Bloc Diviseur de fréquence

Figure 4.4. Architecture interne du Bloc Diviseur de fréquence

Figure 4.5. Architecture du Bloc Maintient

Figure 4.6: Architecture interne du BlocMaintient

Figure 4.7. Architecteur du bloc commande logique (Le séquenceur)

Figure 4.8. Architecteur interne du bloc commande logique (Le séquenceur)

Figure 4.9. Chronogramme du bloc Diviseur de fréquence

Figure 4.10. Chronogramme du bloc Maintient

Figure 4.11. Chronogramme du bloc commande logique (Le séquenceur)

Figure 4.12. Chronogramme finale

Figure 4.13 Simulation sous TINA

Figure 4.14 Layout de l'architecture du système

Figure 4.15 système de contrôle des feux à distance

Chapitre 1 Généralité de feux tricolore

1.1 Introduction

Les carrefours à feux constituent l'un des moyens les plus importants qu'utilise l'homme pour réguler la circulation des véhicules sur un croisement de deux voies.

Aujourd'hui l'organisation de la circulation des véhicules dans un carrefour est assurée par l'homme ainsi que les systèmes de commande pour gérer tous les cas qui existent dans la route. Le carrefour à contrôler possède deux voies (voie1, voie2) protégées par des feux tricolores.

L'objectif de ce chapitre est de faire connaissance des principes généraux de conception d'un carrefour sur lequel est installée une signalisation lumineuse tricolores, où on présentera quelques principes élémentaires de conception, de configuration topographiques, et de quelques types de contrôleurs ainsi que la machine d'états pour la gestion des feux.

1.2 Les feux tricolores

Un feu de circulation est un dispositif permettant la régulation entre les usagers de la rue ou de la route, utilisé pour la gestion de la circulation de véhicules par les feux (rouge, vert, orange).

Le feu autorise ou non le passage des véhicules sur les voies en fonction de sa couleur. Il est important de bien respecter les arrêts imposés par le feu tricolore sous peine d'être en situation d'infraction routière [1].

Le feu tricolore règle la circulation selon trois couleurs:

- ✓ Le feu rouge, indiquant l'arrêt obligatoire.
- ✓ Le feu vert, indiquant le passage libre et autorise le conducteur à s'engager dans une intersection.
- ✓ Le feu orange, appelé feu jaune dans le Code de la route, impose un arrêt s'il est fixe, sauf si lorsque le feu passe à l'orange l'arrêt du véhicule peut présenter un danger. Un feu orange clignotant a pour fonction d'attirer l'attention des conducteurs sur un éventuel danger. Il autorise ainsi le passage du véhicule, à condition de respecter les dispositions relatives aux règles de priorité.

L'équipement d'un carrefour avec une signalisation lumineuse tricolore a pour objectif de supprimer certains conflits, en séparant dans le temps l'admission de flux d'utilisateurs incompatibles.

Cette gestion des conflits dans le temps impose le respect d'un certain nombre de règles de conception et de dimensionnement. Même avec la signalisation lumineuse la plus sophistiquée, un carrefour à feux ne peut avoir un bon fonctionnement sans un aménagement géométrique adapté. A ce titre, conception et fonctionnement doivent être étudiés ensemble.

1.3 Configuration topographique

La configuration matérielle permet de délimiter les frontières du système où évolue l'objet "Feux tricolores de croisement".

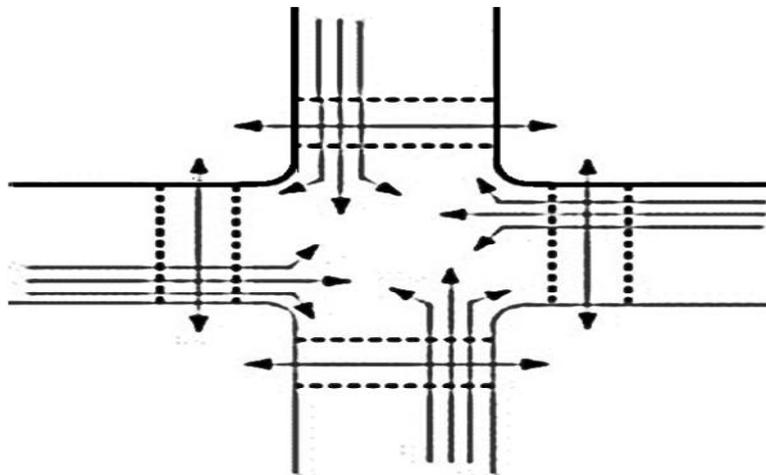


Figure 1.1. Les 16 mouvements possibles sur un carrefour à quatre branches

Notre projet a pour objectif la gestion des cycles de feux permettant d'assurer la sécurité des automobilistes et des piétons tout en offrant un débit maximal de véhicule afin de fluidifier et d'organiser la circulation de piétons et des automobiles au niveau d'un carrefour, pour lequel un certain nombre de contraintes doivent être respectées.

En effet, ces contraintes sont au moins de quatre types :

1. Contrainte de **Conflit** : consistant à l'évitement de croisement de deux *mouvements simultanés*, de véhicules ou de piétons.
2. Contrainte de **Courant** qui doit rejeter la réunion de l'ensemble des *mouvements* sur une même voie (piétons et véhicules utilisant la traversée de la voie simultanément).
3. Contrainte du **Cycle des feux** devant tenir compte d'une programmation (intelligente) correcte de l'enchaînement des différentes *phases des feux*.

Contrainte de **Durée de vert minimum** qui doit imposer une durée minimale réglementaire de vert nécessaire pour écouler le stock d'usagers constitué pendant la durée de rouge.

Le caractère conventionnel du feu tricolore concerne donc les couleurs à utiliser, mais également leur ordre de succession ou d'apparition : en effet, l'allumage des feux est séquentiel, c'est-à-dire astreint à une procédure ordonnée. De plus, des durées minimales (et souvent maximales) d'exposition de chaque couleur ont été définies par les administrations nationales compétentes, et font aujourd'hui l'objet d'une standardisation largement partagée. Un vocabulaire technique commun désignant le fonctionnement de la signalisation tricolore est également utilisé par l'ensemble de la communauté des ingénieurs [1]:

- un état (ou intervalle) : correspond à l'allumage d'une couleur sur le feu : il y a donc trois états possibles : vert, jaune et rouge.
- une phase : durée pendant laquelle un feu passe par les trois états, c'est-à-dire effectue une séquence d'allumages.
- un cycle : période pendant laquelle l'ensemble des feux effectuent leur phase, permettant successivement l'admission de tous les courants de véhicules dans le carrefour.

- un plan de feux : durée et ordre de déroulement de l'allumage complet, pendant un cycle, de l'ensemble des groupes de feux (on verra qu'un même carrefour peut suivre alternativement plusieurs plans de feux).
- un rouge barrage (ou intégral) : durée pendant laquelle aucun véhicule n'est admis dans le carrefour, c'est-à-dire lorsque l'état est rouge sur chaque feu.

Dans un premier temps, le fonctionnement des feux tricolores ne tiendra compte ni du trafic ni des piétons. Soit manuellement soit automatiquement à certaines heures du jour, il sera possible de substituer au fonctionnement habituel un fonctionnement pour le trafic à faible densité.

Il existe deux modes de fonctionnement, selon l'importance du trafic routier :

- Mode « feux tricolores », pour un trafic normal ou intensif,
- Mode « feux oranges clignotants », pour un faible trafic.

L'étude suivante portera sur une maquette reproduisant quelques-unes des fonctionnalités liées à la gestion d'un croisement par des feux tricolores [2].

1.3.1 Types de contrôleurs

Ce qu'il faut de comprendre en fait, est la manière dont s'opère le contrôle de ces opérations d'affichage du signal et de leur succession. Originellement, le « contrôleur » du feu est un être humain, utilisant son intelligence, c'est-à-dire sa perception du trafic, pour activer le sémaphore, tel qu'il soit (agiter un membre, émettre un son, allumer une lampe, etc.). Suite aux critiques d'inefficacité mais aussi de coûts liés à cette technique, des contrôleurs automatiques (c'est-à-dire non humains) furent mis au point. On en distingue trois types ou générations :

a Les contrôleurs à logique programmable

Dans ce type, il nous faut tout d'abord une programmation en langage **VHDL**. Cette programmation consiste à gérer et synchroniser les états des deux feux selon les conditions qui sont présentes. Pour cela, il faut réaliser une machine d'état de la gestion des feux tricolores et le schéma synoptique du montage [3].

b Les contrôleurs électromécaniques

Introduits pour la première fois aux Etats- Unis en 1922, ils fonctionnent à l'aide d'un arbre à cames animant les interrupteurs contrôlant les lampes, entraîné par un moteur pas à pas ou un disque [3].

c Les contrôleurs à logique câblée

Ici, le dispositif électromécanique est remplacé par un compteur (ou horloge) numérique de faible voltage. On y relie alors les circuits chargés d'établir les aiguillages correspondant aux divers plans de feux. En fait, chaque plan possède un circuit distinct, toute modification de plan exigeant alors une nouvelle connexion des circuits au contrôleur électronique [3].

d Les contrôleurs à microprocesseur

L'insertion d'un microprocesseur à l'intérieur du contrôleur a pour double avantage de réduire le nombre et le volume des connexions exigées par la solution câblée, et de transformer la programmation des plans de feux et leurs modifications en transmission d'instructions informatiques (ce qui n'exige plus la commutation physique des circuits). Les plans de feux sont en fait stockés dans une mémoire électronique [3].

1.3.2 Modes de fonctionnement des contrôleurs

Les différents types de contrôleurs peuvent fonctionner sous divers modes. Outre le contrôle manuel, on distingue généralement le contrôle automatique, le contrôle adaptatif et le contrôle coordonné. Dans les faits, ces modes ne sont pas toujours distincts et peuvent se combiner à plusieurs titres, voire même se substituer l'un à l'autre [4].

a Mode manuel

Dans ce cas, c'est un individu qui actionne le changement d'état du carrefour. C'était le cas des premiers feux de circulation, comme on l'a vu ; mais aujourd'hui encore, les

feux peuvent fonctionner en mode manuel si la situation l'exige, ou également en mode semi-manuel (ou semi-automatique), lorsque l'opérateur humain actionne en fait le déclenchement d'une phase et non de chaque intervalle [4].

b Mode automatique

Ce mode n'exige aucune intervention extérieure : les états successifs d'un cycle se déroulent séquentiellement suivant les plans de feux en place dans le contrôleur ; en conséquence, on parle aussi de fonctionnement en cycles fixes [4].

c Mode adaptatif

Contrairement au mode automatique, le mode adaptatif permet l'adaptation de la durée de l'intervalle vert, et par conséquent la modulation de l'ensemble des phases d'un cycle, en fonction de la demande, c'est-à-dire en fonction des variations du trafic. Ainsi, un vert peut être allongé pour écouler le flux d'une voie où s'écoule un trafic plus important que dans les voies adjacentes. Un autre exemple d'adaptation est la priorité donnée aux véhicules de transport public : leur arrivée dans un carrefour déclenche soit l'allongement de l'état vert, soit la réduction de l'intervalle rouge sur la voie qu'ils empruntent [4].

d Mode coordonné

Ce mode signifie que plusieurs contrôleurs, donc plusieurs carrefours, sont soumis à une même stratégie de régulation. Il existe deux grandes applications du mode coordonné. La première est la régulation d'axes de circulation, plus connue sous l'expression d'onde verte. Elle consiste à coordonner les phases des différents carrefours d'un même axe routier afin de rendre plus confortable et plus fluide la progression des véhicules, c'est-à-dire d'éliminer leurs arrêts aux carrefours (du moins s'ils respectent une vitesse moyenne, définie par le gestionnaire des feux, qui permet alors de profiter de l'onde verte). La seconde application de la coordination est la régulation de zones. Dans ce cas, ce sont les feux de l'ensemble d'un périmètre géographique déterminé (un quartier urbain par exemple) qui sont coordonnés entre eux, afin d'optimiser les déplacements des usagers, c'est-à-dire de minimiser le temps qu'ils passent dans cette zone [4].

Nous avons adoptés dans ce travail les deux premiers modes parce que ce sont les plus courants.

Les deux derniers modes qu'on vient d'évoquer soulèvent chacun des problèmes techniques que ne posent pas les deux premiers modes. Le mode adaptatif exige de disposer de moyens de détecter les variations du trafic, c'est-à-dire de recueillir des données caractérisant la circulation. Le mode coordonné, outre le fait qu'il peut être lui aussi adaptatif et donc requérir la présence de capteurs, exige que s'opère une transmission d'informations entre les différents contrôleurs des carrefours d'une même zone.

1.4 Machine d'états

La machine d'état permettant de contrôler notre carrefour constitué de deux voies, la voie 1 et la voie 2 (cf. figure, possède six états qui correspondent aux différentes situations possibles du Problème. Dans chaque état, on gère ensemble l'allumage des deux feux tricolores. Cette machine d'état est implémentée dans le bloc **machine_etat** (figure 1.2).

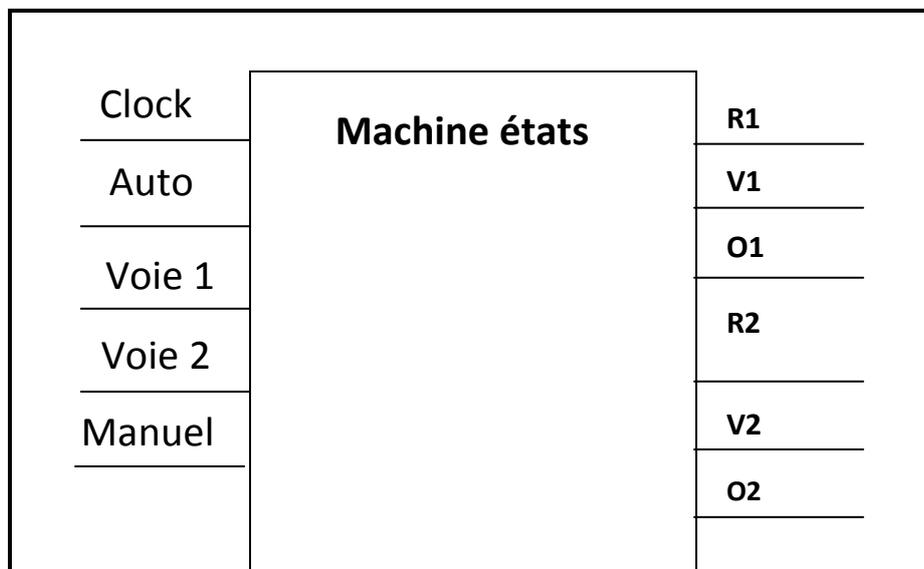


Figure 1.2. Bloc machine états.

Les conditions de transition (entrées) utilisées sont :

- *Clock*, pour générer le signal d'horloge
- *Auto*, (mode 1)
- *Voie 1*, (mode 2)
- *Voie 2*, (mode 3)
- *Manuel*, (mode 3)

Les sorties R1, O1, V1, R2, O2 et V2 de la **machine états** permettent d'allumer ou d'éteindre les différents feux selon l'état dans lequel est le système [6].

1.4.1 Convention sur le nom des signaux :

Par la suite de notre projet, afin de simplifier le travail et ainsi de pouvoir coder cette programmation de gestion plus tard, on donne les noms ou les abréviations à chaque état des feux, comme indiqué ci-dessous :

- R1, O1, V1 : Rouge, Orange, Vert, **voie1**:

Ces abréviations correspondront à l'état du feu de première voie.

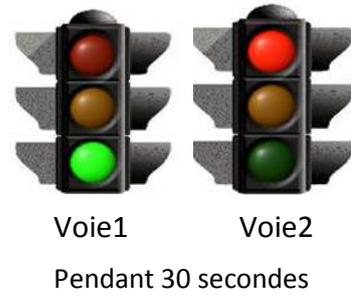
- R2, O2, V2 : Rouge, Orange, Vert, **voie2** :

Celles-ci correspondent à l'état du feu de la deuxième voie.

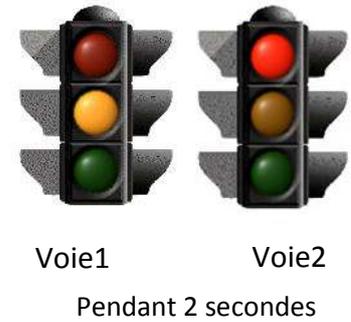
Dans ce travail nous avons adopté quatre boutons poussoirs appelés *BP*.

- **Mode1 : Auto(automatique) :**

A l'état **V1_R2_wait_30s**, le feu de la voie1 est au vert et le feu de la Voie2 au rouge ou Allumage de feux vert de la voie1, et des feux rouges de la voie 2 durant 30 secondes : $V1=1$ et $R2=1$.



A l'état **O1_R2_wait_2s**, le feu de la voie1 est orange et celui de la voie2 est rouge : $O1=1$ et $R2=1$, et garde cette configuration pendant 2 secondes



A l'état **V2_R1_wait_30s**, le feu de la voie1 est au rouge et le feu de la voie2 au vert ou Allumage de feux rouge de la voie 1, et des feux verts de la voie2 durant 30 secondes : $V2=1$ et $R1=1$.

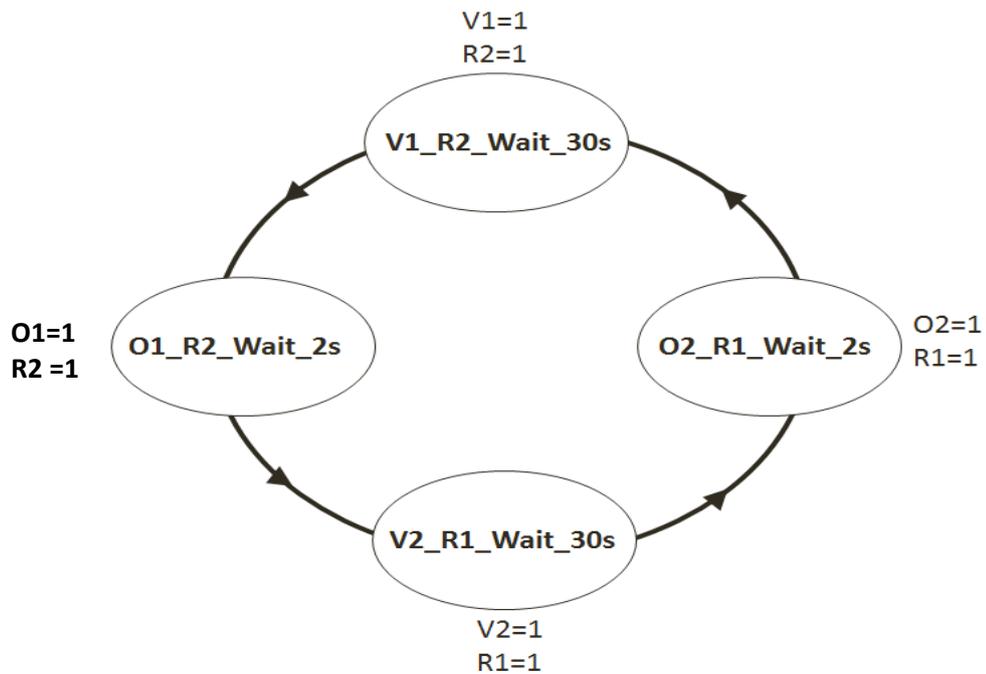
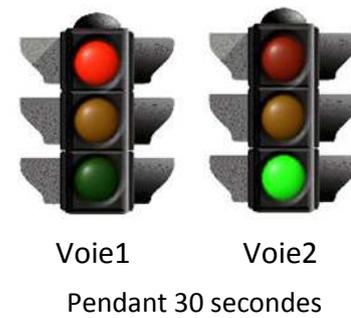
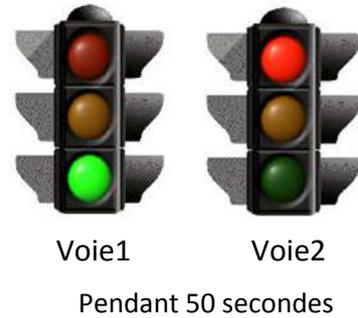


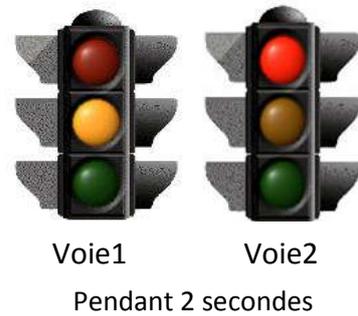
Figure 1.3. Machine d'états du mode 1

- **Mode 2 : Voie1**

A l'état V1_R2_wait_50s, le feu de la voie1 est au vert et le feu dans la voie2 au rouge ou Allumage des feux verts de la voie 1, et des feux rouges de la voie 2 durant 50 secondes : V1=1 et R2 =1.



A l'état O1_R2_wait_2s, le feu de la voie1 est orange et celui de la voie2 est rouge : O1=1 et R2=1, et garde cette configuration pendant 2 secondes



A l'état V2_R1_wait_30s, le feu de la voie1 est au vert et le feu Voie2 au rouge: V2=1 et R1 =1.

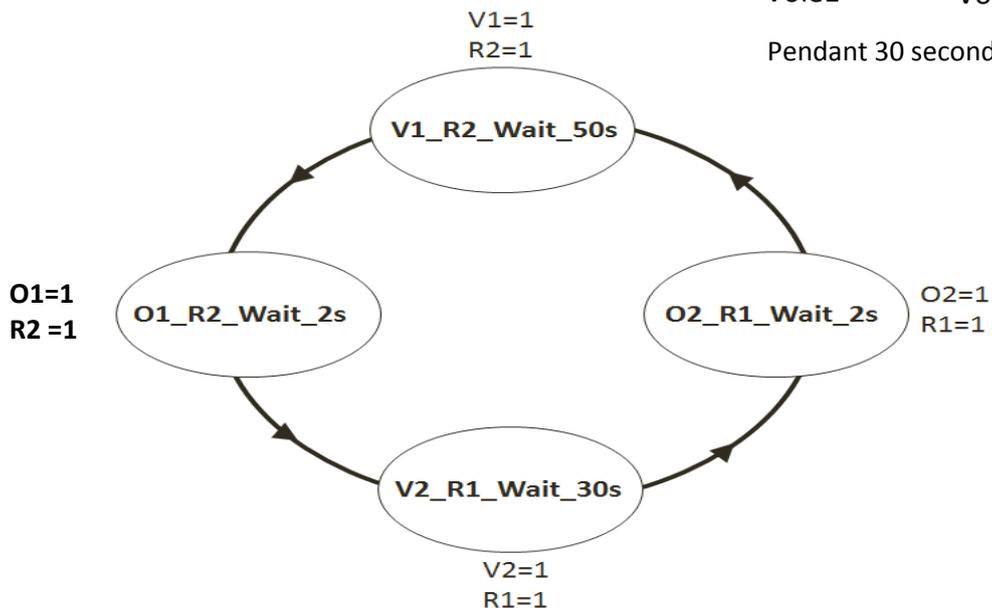
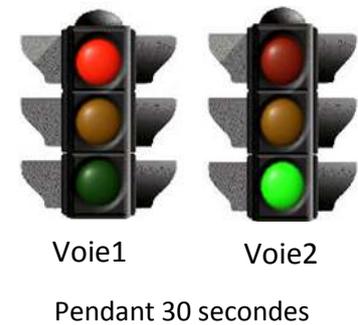
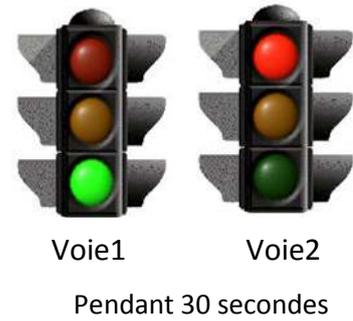


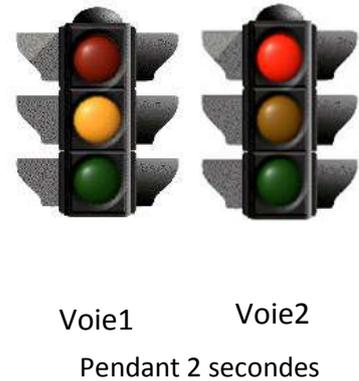
Figure 1.4. Machine d'états de mode 2

- **Mode 3 : Voie 2**

A l'état V1_R2_wait_30s, le feu de la voie1 est au vert et le feu de la Voie2 au rouge ou Allumage des feux verts de la voie 1, et des feux rouges de la voie 2 durant 30 secondes : V1=1 et R2 =1.



A l'état O1_R2_wait_2s, le feu de la voie1 est orange et celui de la voie2 est rouge : O1=1 et R2=1, et garde cette configuration pendant 2 secondes.



A l'état V2_R1_wait_50s, le feu de la voie1 est au vert et le feu de la voie2 au rouge: V2=1 et R1 =1.

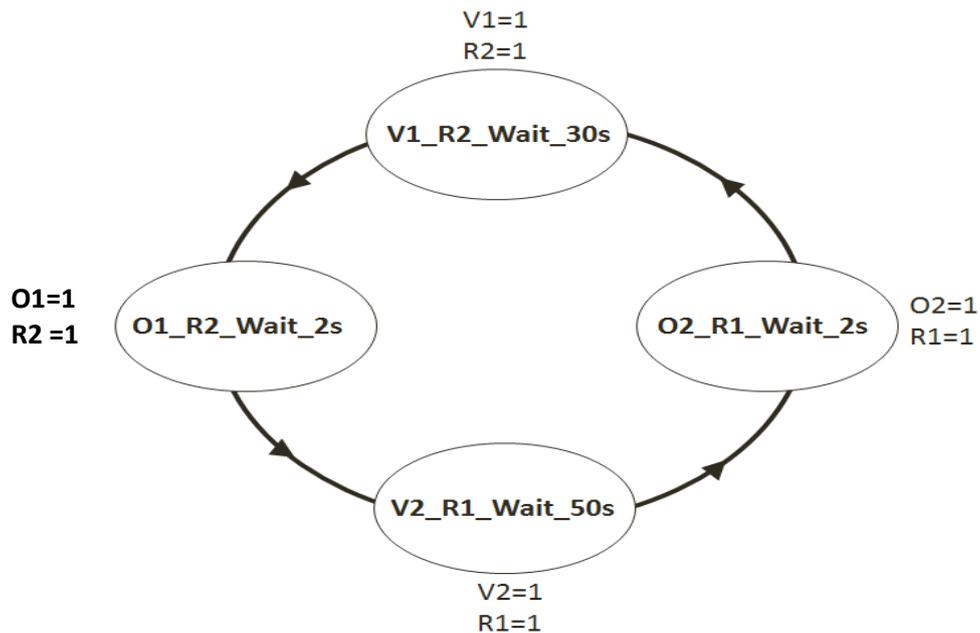
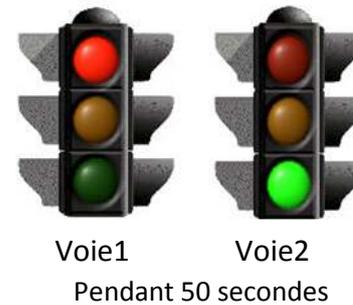


Figure 1.5.Machine d'états de mode 3.

- **Mode 4 : Manuel**

A l'état O1_O2, le feu de la voie1 est celui de la voie2 est orange: O1=1 et O2 =1. C'est le mode manuel, où intervient l'utilisateur (policier), et pour cela les feux orange restent clignotants dans cette configuration afin d'attirer l'attention des conducteurs.

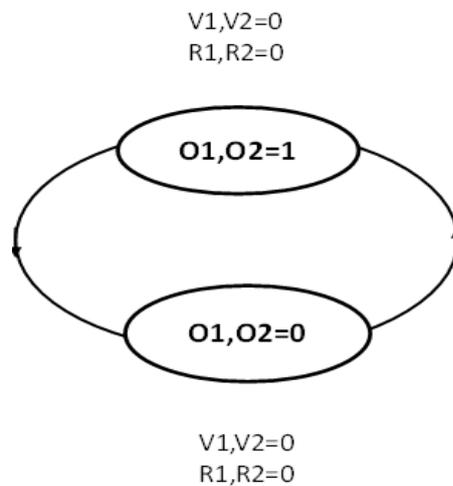
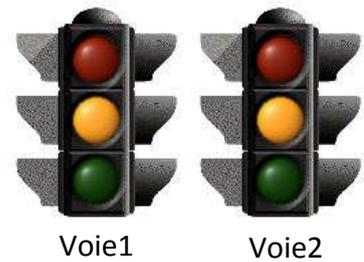


Figure 1.6. Machine d'états de mode 4.

1.5 Conclusion

Nous avons présenté dans ce chapitre, des généralités sur les feux tricolores ainsi que les différents types de contrôleurs qui constituent un système de feux ainsi que la configuration topographique des feux. Quatre modes ont été abordés concernant la machine d'états pour la gestion des feux : mode automatique (voie 1 et voie 2 ont la même priorité), mode voie1 (priorité à la voie 1), mode voie2 (priorité à la voie 2) et mode manuel (l'homme gère le carrefour).

Chapitre 2 Présentation globale du système de commande du feu tricolore

2.1 Introduction

Ce chapitre cible les communications sans fils orienté vers les systèmes télécommandés. Il existe plusieurs techniques de commande à distance. La télécommande d'un téléviseur (infrarouge), le téléphone cellulaire, Bluetooth ou le radio fréquence sont des exemples de transferts d'informations sans fils (Wireless) et sur des distances très différentes. Pour les courtes distances et lorsque l'émetteur et le récepteur peuvent avoir un contact visuel (line of sight), l'émission d'onde lumineuse infrarouge est souvent employée. Pour les communications sur de plus grandes distances impliquant des obstacles, c'est la communication par ondes radios (RF) qui est souvent privilégiée.

Notre carte utilise un module radio fréquence (RF) faisant office de télécommande sans fil, qui peut être utilisé par un agent de la circulation. Ce module RF, comme son nom l'indique, utilise la radiofréquence pour transmettre des signaux HF vers un récepteur configuré de manière à réagir automatiquement aux signaux accordé à ladite fréquence.

L'objectif de notre chapitre étant l'utilisation d'un module radiofréquence très faible consommation adapté aux applications de feu trafic. Son étude sommaire est nécessaire afin de déterminer les caractéristiques de l'émetteur et du récepteur que nous désirons atteindre en fonction des spécifications du cahier de charge. Dans ce

travail, nous avons utilisé un module télécommandé commercial pour gérer les feux routier.

Dans ce qui suit, nous présentons le système de commande feu tricolore associé à quelques notions sur les communications sans fil.

2.2 Généralité sur la radio fréquence

Ces détecteurs permettent la réception, l'amplification, la démodulation et la mise en forme des signaux RF issus de télécommandes (le module ressort directement les créneaux du codage). Idéal pour la réalisation de récepteur de télécommandes. Excellente sensibilité, grande fiabilité et immunité aux perturbations parasites.

Dans notre cas, la radiocommande utilisée consiste à contrôler les feux routiers, par radio fréquence ou l'agent de la circulation dispose d'une télécommande «intégrées dans un boîtier» programmée de manière à transmettre des « consignes d'allumage » par l'intermédiaire de quatre boutons poussoirs (figure 2.1).

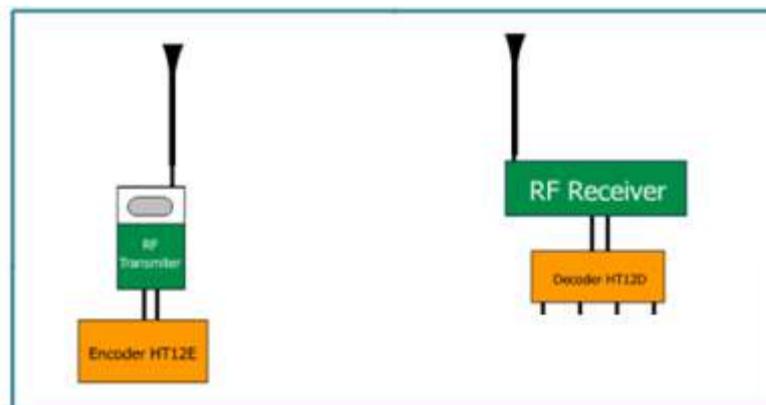


Figure 2.1. Récepteur et émetteur RF.

2.3 Structure d'émetteur et récepteur

Le système de commande du feu tricolore est composé de plusieurs étages permis lesquels on distingue : un émetteur, un récepteur, la carte FPGA, la carte de puissance et les panneaux des feux tricolores.

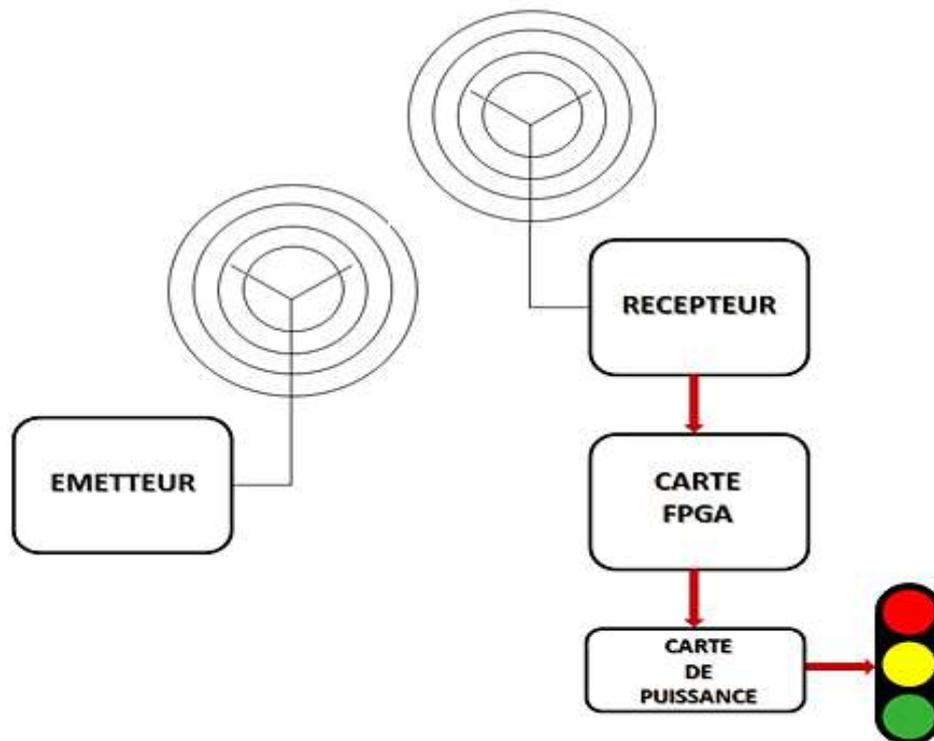


Figure 2.2. Schéma synoptique du système de commande d'un feu tricolore.

2.3.1 Carte de l'émetteur

L'émetteur génère une onde porteuse de fréquence 40 MHz dans la bande nécessaire qui tient de support à la transmission des informations commandant le dispositif. Cet émetteur utilise quatre boutons poussoirs (figure 2.3) générant chacun une consigne spécifique.

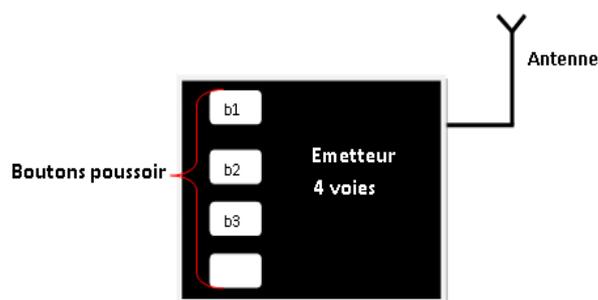


Figure 2.3. Dispositif Émetteur.

Chaque bouton est associé à une fonction particulière :

- **Bouton B1** : activation du mode automatique (les deux voies ont la même priorité).
- **Boutons (B2,B3)** : on fait intervenir ces boutons lorsque l'une des voies perpendiculaire à un flux de trafic plus important que l'autre, le tableau suivant présente la relation qui existe entre les boutons B2 et B3 et le trafic des voies 1 et Voie 2.

B3 inactif; B2 actif	B3 actif; B2 inactif
La Voie 1 a un trafic plus dense que la voie 2.	La Voie 2 a un trafic plus important que la voie 1.

Tableau 2.1. Comparaison de deux boutons B2 et B3.

- **Le bouton B4** : activation du mode manuel (l'utilisateur prend le contrôle).

L'émetteur alimenté sous 3 volts délivre, une consigne qui exprime l'activation de l'un des quatre boutons. Cette activation va être prise en compte au niveau du récepteur.

2.3.2 Récepteur

C'est un dispositif qui assure la réception du signal venant de l'émetteur. Il admet en entrée une antenne et en sortie quatre voies dont chacune a des 4 boutons de commande de l'émetteur.

Le récepteur est souvent plus complexe que l'émetteur. La structure finale d'un récepteur découle d'une suite de compromis entre les différents paramètres influant sur les performances. Tous ces paramètres sont étroitement imbriqués, il n'y a ni solution idéale ni solution universelle.

Le récepteur est la partie électronique se situant à bord de la maquette. Il doit être léger et faible. Il reçoit et décode les signaux envoyés par l'émetteur et transmet les ordres aux servocommandes.

Le récepteur se présente sous la forme d'un petit boîtier de la taille d'une boîte d'allumettes. Il y sort une antenne filaire. Un petit groupe de connecteurs permet d'y raccorder le pack d'accus (ou les piles) de réception, variateurs.

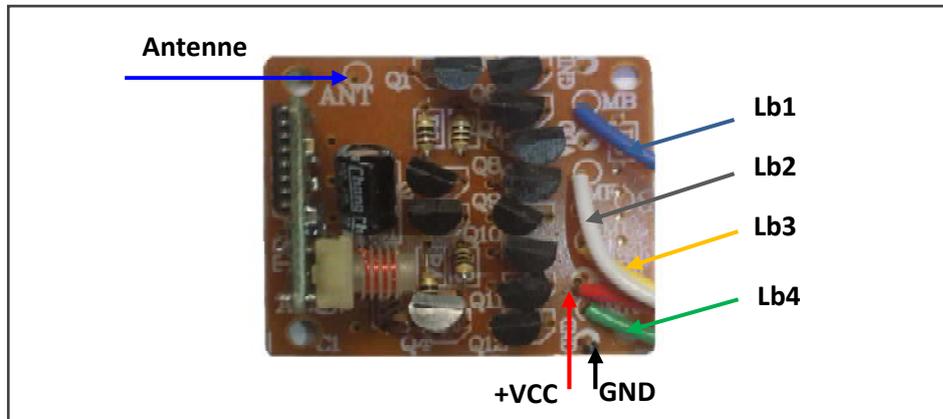


Figure 2.4. Récepteur du feu tricolore.

Chaque fois qu'un bouton b_i ($i = 1;4$) de l'émetteur est actif, le récepteur met la ligne de sortie L_{b_i} à 5 volts et les autres à 0 volts. Ces les L_{b_i} constituent une entrée du circuit du système de commande implémenté dans un FPGA.

Le récepteur reçoit une fraction de la porteuse modulée émise en présence de bruit et de multiples autres signaux de puissance et de fréquences diverses et inconnues. Ceci se conçoit particulièrement bien en examinant le cas de la bande de fréquence.

2.3.3 Circuit du système de commande

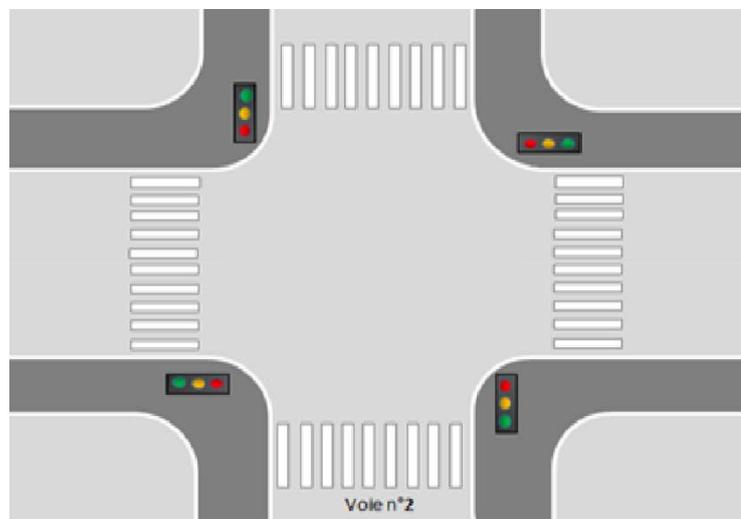


Figure 2.5. Feux tricolores de croisement.

Sachant que les cartes FPGA disposent de fréquence d'horloge très important (50Mhz pour la carte utilisée dans notre cas la spartan3E), nous devons la réduire, pour ce la nous avons conçu un diviseur de fréquence qui nous génère un signal d'une fréquence 1Hz.

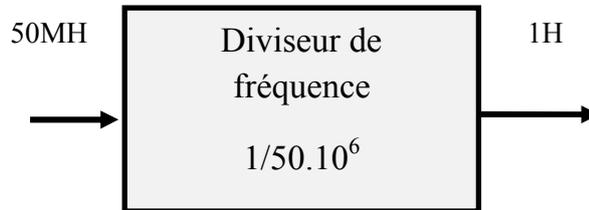


Figure2.6. Schéma synoptique de diviseur

Pour pouvoir garder le signal émit par la télécommande, on a eu recourt à un module de Maintien des entrées, ces dernières vont être injectées dans la carte FPGA qui contient le programme principal de commande du système du feu tricolore

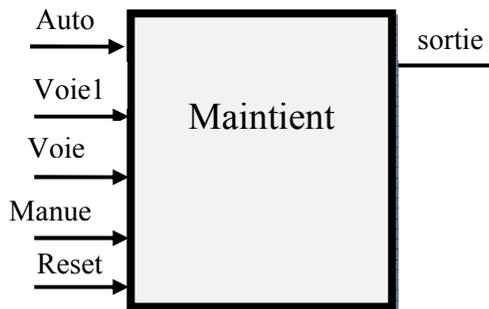


Figure2.7. Schéma synoptique duMaintien.

Le programme de commande est un séquenceur dont l'entrée est composée de l'horloge du FPGA, la commande qui arrive de la télécommande, et la sortie est sous forme de six lignes qui vont attaquer la carte de puissance pour activer les feux tricolores.

Le chronogramme ci-dessous montre le séquencèrent des activations des feux:

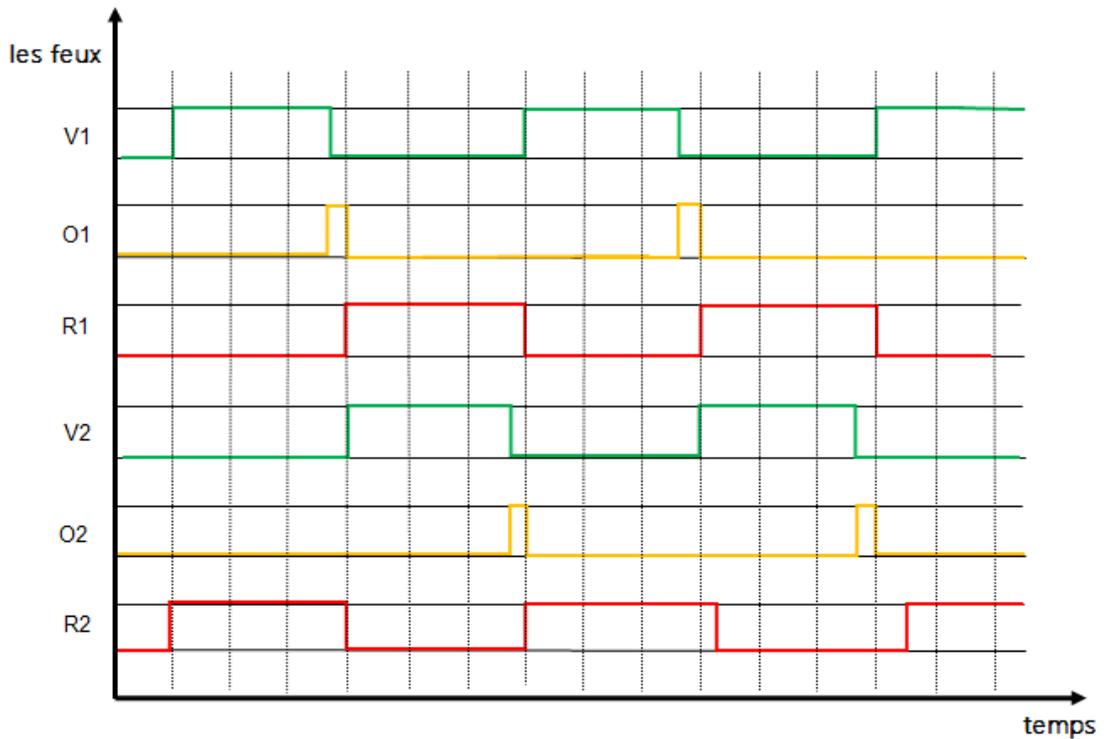


Figure 2.8. Représentation temporelle.

2.3.4 La carte de puissance

Comme son nom L'indique elle sert à amplifier les tensions qui arrivent du circuit du système de commande implanté dans le FPGA.

Elle comporte six (6) lignes d'entrées convient également des sources trio d'alimentation en tension.

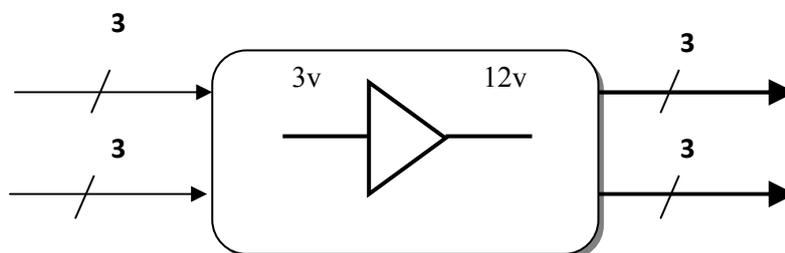


Figure 2.9. Amplificateur de tension

La carte se compose d'un ensemble de transistor monté ou applicateur de tension et travaillant en mode saturation blocage pour actives ou désactives des relais (figure 2.10)

C'est pour cette raison que l'on procédera au calcul des composait pour une ligne et on dupliquera pour le reste des autres lignes.

On a mit un transistor bipolaire avant les relais pour deux raisons :

- Protection pour la partie commande en cas de court-circuit
- Amplification du courant pour pouvoir actionner les relais ($I_C = \beta I_B$ Avec :
 I_B : courant délivré pour carte de commande.
 β : gain des transistors.)

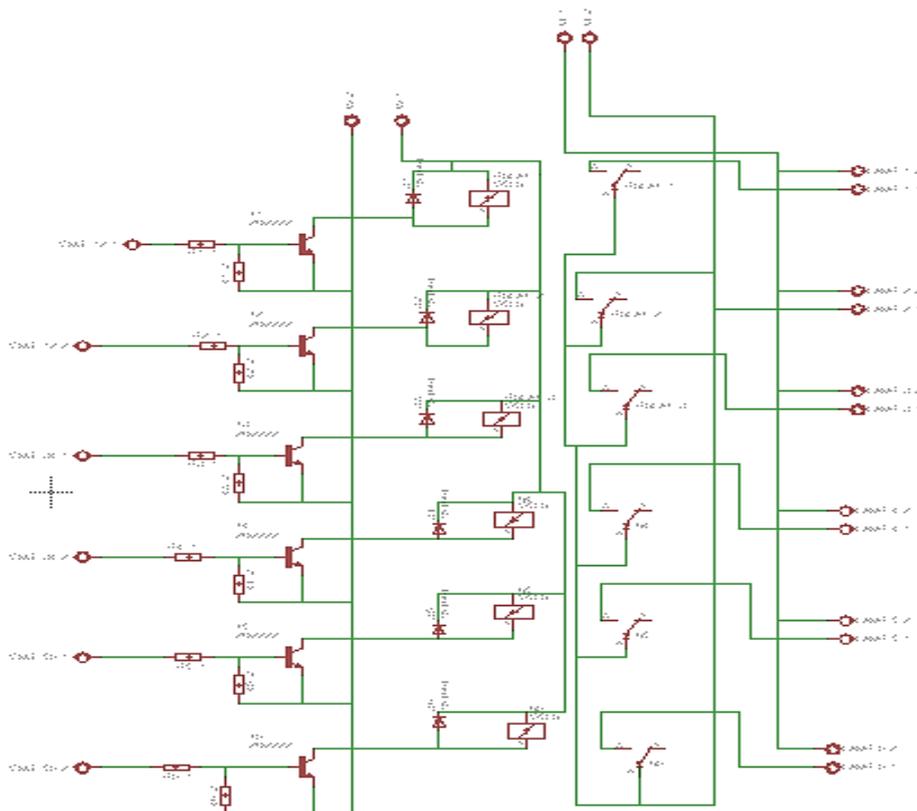


Figure 2.10.schéma électrique de tous les ligne amplifiées.

2 .4 Conclusion

Dans ce chapitre, nous avons présenté des généralités sur la conception des feux tricolores, ainsi les différents modes de contrôle, et les modes accessibles par notre système.

Dans ce chapitre résumera Une présentation globale du système de commande du feu tricolore, ce système est assuré par un émetteur récepteur RF. L'émetteur permet d'avoir 4 modes # via 4 B P. se décompose en un système de commande, une carte de puissance

Chapitre 3 Technologie des circuits FPGA

3.1 Introduction

Au début des années 80, les développeurs disposaient de circuits de type PLD, facilement programmables mais ne pouvant contenir des designs trop complexes. Les ASICS quant à eux supportaient des systèmes de grande complexité mais n'avaient pas les propriétés de configuration des PLD. Il manquait donc un type de circuits permettant la réalisation des systèmes complexes, tout en offrant une configuration rapide et peu onéreuse. C'est pourquoi en 1984, Ross Freeman, Bernie Vonderschmit, et Jim Barnett fondent la compagnie Xilinx. En 1985, ils introduisaient sur le marché le premier FPGA, le XC2064, et offrent ainsi une alternative aux précédentes approches.

Depuis la commercialisation du premier pré diffusé programmable (FPGA) en 1985, l'utilisation de ces circuits ne cesse de s'étendre à des domaines et applications variés. Les circuits FPGA, permettent d'implémenter des systèmes numériques aussi complexes que ceux réalisés jusqu'alors grâce aux ASICS, tout en ayant le grand avantage de pouvoir être programmés électriquement [7].

L'intérêt suscité par les FPGA est dû essentiellement à leurs prix abordables, facilité de mise en œuvre et flexibilité. En outre, les coûts fixes et délais de fabrications, en comparaison avec les circuits spécifiques (ASIC), sont totalement éliminés. Cependant, ils présentent une faible densité d'intégration de portes logiques et atteignent des fréquences de travail relativement faibles devant les ASIC.

Les circuits FPGA sont principalement composés d'un tableau d'éléments plus ou moins complexes pouvant être configurés, ainsi que d'un réseau complexe de connexions également configurables.

Un FPGA est un composant électronique constitué de millions voir milliers de transistors connectés ensembles pour réaliser des fonctions logiques. Des fonctions logiques simples peuvent être réalisées telles que des additions ou soustractions tout comme des fonctions complexes peuvent être réalisées telles que le filtrage numérique du signal ou détection d'erreurs et correction. Aérospatial, aviation, automobile, radar, missiles, ordinateur... ne sont que quelques exemples des domaines ayant recours aux FPGAs.

Dans ce chapitre nous allons décrire d'une manière générale l'architecture des circuits FPGA ainsi que les étapes nécessaires au développement d'un projet sur le circuit FPGA Spartan 3E, disponible dans notre université, en partant de la programmation en utilisant le langage de description matériel VHDL, jusqu'au chargement sur la carte FPGA, ce qui permet d'utiliser par la suite les ressources de cette carte pour implémenter notre système de commande de feu tricolore discuté dans le chapitre précédent.

3.2 Description d'un FPGA (Field Programmable Gate Array)

Les FPGA (Field Programmable GateArray) sont des circuits à architecture programmable qui ont été inventés par la société XILINX en 1985. Ils sont entièrement reconfigurables et ne demandent donc pas de fabrication spéciale en usine, ni de systèmes de développement coûteux; ceci permet de les reprogrammer à volonté afin d'accélérer notablement certaines phases de calculs. Un autre avantage de ces circuits est leur grande souplesse qui permet de les réutiliser à volonté dans des algorithmes différents en un temps très court (quelques millisecondes).

De nombreuses familles de circuits programmables et reprogrammables sont apparues Depuis les années 70 avec des noms très divers suivant les constructeurs. Une classification possible des circuits numériques en précisant où se situent les circuits FPGA dans cette classification. Les FPGA sont utilisés dans de nombreuses applications, on en cite dans ce qui suit quelques unes:

- Prototypage de nouveaux circuits;
- Fabrication de composants spéciaux en petite série;
- Adaptation aux besoins rencontrés lors de l'utilisation;
- Systèmes de commande à temps réel;
- DSP (Digital Signal Processor);
- Imagerie médicale.

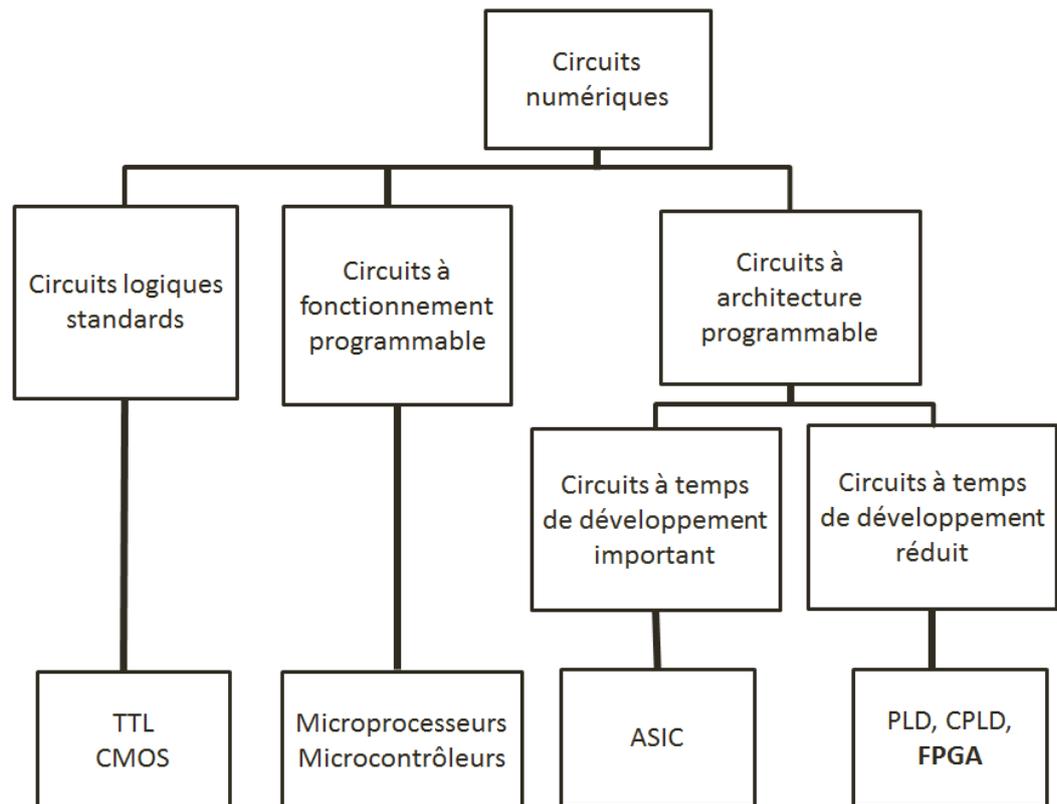


Figure 3.1. Classification des circuits numérique.

Nous allons décrire les différents types de circuit programmable, dans l'ordre chronologique de leur apparition, correspondant également à leur complexité, en commençant par les PLD, les CPLDs, puis les FPGAs .

3.2.1 PLD

Les SPLDs (Simple Programmable LogicDevice), dans une description haut niveau, sont composés d'une grille de portes ET et d'une grille de portes OU, les de deux étant reliées. Les entrées du système peuvent être connectées aux portes ET, et le résultat

des portes OU correspond à la sortie du système. Dans ces circuits les connexions sont préexistantes, les différentes lignes étant reliées par des fusibles ou des transistors. En brûlant certaines de ces fusibles, ou en programmant les transistors, il est alors possible de réaliser différentes fonctions logiques. Il existe dans cette famille les PROMs, les PLAs ainsi que les PALs [8].

3.2.2 CPLD

Les CPLDs (Complex Programmable Logic Device), apparus au début des années 80, sont le résultat de l'évolution des PLDs. Ils permettent l'implémentation de systèmes nettement plus complexes, et sont composés d'éléments de base programmables, connectés entre eux par un réseau d'interconnexions relativement simple. Ces éléments de base sont du type SPLD [8]. La technologie de programmation des CPLDs dépend évidemment du constructeur, et peut être de type EPROM, EEPROM, FLASH ou SRAM (figure 3.2.).

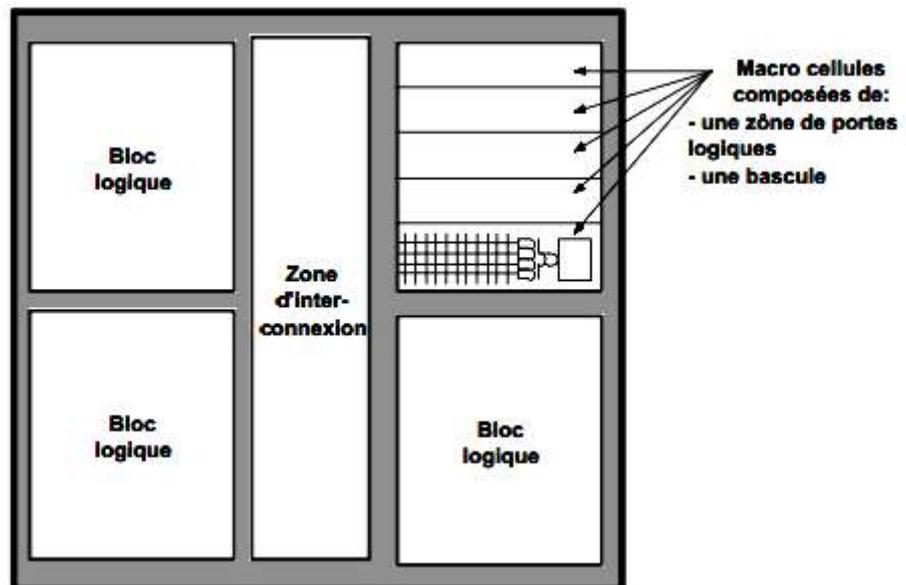


Figure 3.2. Architecture des CPLD

3.2.3 FPGAs

Un FPGA ou (réseaux logiques programmables) consiste en une matrice carrée de cellules configurables CLB (Configurables Logique Blocs) permettant de réaliser des fonctions combinatoires et des fonctions séquentielles. Tout autour de ces blocs logiques configurables, nous trouvons des blocs d'entrées /sorties IOB (Input Output Blocs) dont le rôle est de gérer les entrées-sorties réalisant l'interface avec les modules extérieurs. La figure 3.3 présente l'architecture générale d'un FPGA[10, 8,].

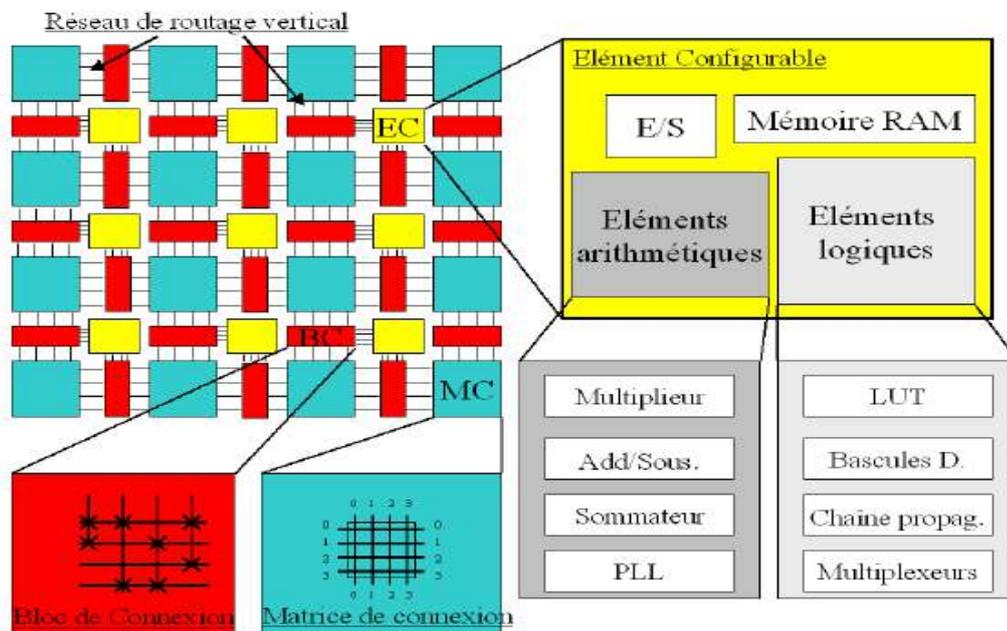


Figure 3.3. Architecture générale d'un FPGA.

Il existe plusieurs fabricants des circuits FPGA. Mais ceux de la société Xilinx occupent à peu près la moitié du marché à cause de leur diversité ainsi que leurs performances.

3.2.4 Familles des FPGA de Xilinx

La tendance des dernières générations est de cibler certains créneaux porteurs du marché comme la taille, le type et le nombre de cellules qui varient suivant les familles de composants [9].

Les FPGA du constructeur Xilinx sont divisés en deux gammes :

- FPGA hautes performances : gamme **Virtex** : VirtexM, VirtexEM, Virtex2, Virtex2Pro, Virtex4, Virtex5...
- FPGA pour la fabrication en grande série : gamme **Spartan** :Spartan2, Spartan2E, Spartan3, Spartan3A, Spartan3AN, Spartan3DSP, Spartan 3E...

3.2.5 Nomenclature des circuits FPGA

Les circuits FPGA suivent la nomenclature suivante, selon un exemple donné [10]:

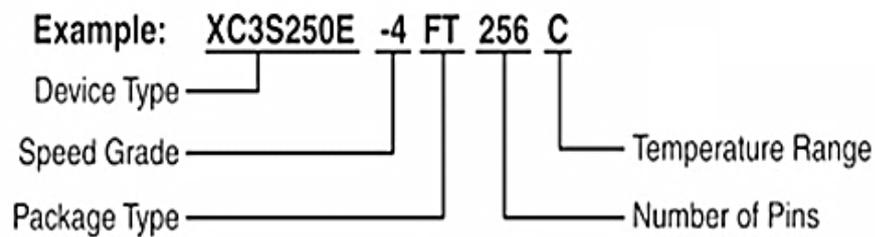


Figure 3.4. Nomenclature d'un FPGA de XILINX.

Device type: il montre le type de la famille qui est dans notre exemple Spartan 3E.

X: est la lettre initiale de Xilinx.

Speed grade:représente la vitesse du composant qui dépend de la technologie du circuit.

256: "Number of pins "représente le nombre de pins.

F: "Package type" représente le type de boîtier.

C: "Temperature Range" représente la gamme de température tolérée.

Terminons ce point par un bref aperçu des avantages et désavantages [7] de l'approche FPGA en commençant par les désavantages:

- Une fréquence d'horloge moins élevée, pour une même application, en comparaison avec les ASICs.

- Une plus grande place nécessaire sur le silicium, beaucoup de logique additionnelle étant nécessaire au bon fonctionnement de la programmation du FPGA.
- Dans la même lignée, l'ensemble des éléments programmables ne sont jamais entièrement utilisés. Leur nombre dépend de la taille du design, mais ne peut pas atteindre 100%, le routage devenant quasiment impossible lorsque le circuit est presque plein.

Et enfin les avantages des FPGAs:

- Un faible coût de développement, le prototype ne nécessitant pas de réalisation matériel, mais seulement des tests successifs sur un FPGA.
- Peu de risques, dans le sens où une erreur de design est très vite corrigée et n'implique pas la création d'un nouveau circuit.
- Une grande rapidité lors de la réalisation d'un prototype.

3.4 Langage de description matériel VHDL

Le réseau de connexions des FPGAs est des plus complexes, de par le nombre d'éléments à connecter (Jusqu'à plus de 150'000) Le placement-routage d'un design pour un circuit donné est donc une tâche laborieuse. Un développeur commence par décrire son système, soit dans un langage de description matériel de type VHDL, Verilog, soit grâce à un éditeur de schéma, tel ViewLogic ou HDL Designer. Le logiciel s'occupe ensuite de la synthèse, transformant la création du développeur en une liste de composants de base et de leurs interconnexions. L'étape suivante consiste à placer ces composants dans les éléments de base du FPGA, puis de créer le routage nécessaire au transfert des signaux entre les composants [7].

3.4.1 VHDL

Le VHDL est un langage de description de matériel électronique. Ce terme est l'acronyme du mot anglais (VHSIC HDL : Very High Speed Integrated circuits Hardware DescriptionLanguage). Ce langage de programmation est conçu en vue de la

modélisation et de la simulation de circuit ASIC, le VHDL traite des circuits intégrés à très grande spécification, la simulation et la synthèse d'une électronique, VHDL est l'un des plus réponsus de nos jours.

L'intérêt d'une telle description réside dans son caractère exécutable: une spécification décrite en VHDL peut être vérifiée par simulation, avant que la conception détaillée ne soit terminée. En outre, les outils de conception assistée par ordinateur permettant de passer directement d'une description fonctionnelle en VHDL à un schéma en portes logiques ont révolutionné les méthodes de conceptions des circuits numériques, FPGA[11].

Le langage VHDL a été commandé par le Département de la défense des Etats Unis. Sa version initiale standard IEEE 1076-1987, incluait un large éventail de types de données, numériques (entiers, réels), logiques (bits, booléens), caractères, temps, plus les tableaux de bits et chaînes de caractères. L'un des principaux problèmes concernait le type bit. Celui-ci ne pouvant prendre que 2 valeurs (0, 1), il était impossible de représenter les signaux de valeur inconnue ou encore les signaux en haute impédance. La norme IEEE 1164 définit le type `std_logic` avec neuf états possibles. Ceci a été adopté dans le VHDL-93 (seconde version de la norme IEEE 1076).

Afin de répondre aux différents problèmes de l'électronique, la norme VHDL a dû évoluer. L'IEEE Design Automation Standards Committee (DASC) a créé la norme IEEE 1076.1 (1999), ou VHDL-AMS (VHDL-Analog& Mixed Systems). Cette nouvelle norme est une extension de la norme IEEE 1076-1987 déjà existante. Elle permet la description et la simulation de circuits analogiques, numériques, et mixtes (analogique et numérique) [12].

3.4.2 Structure d'une description VHDL simple

La description d'un objet en VHDL se compose d'une description de l'interface externe et d'une description de la structure interne ou du fonctionnement interne. On parle d'entité pour la partie externe: Entity. L'entité comprend principalement le

port, contenant lui-même la liste des signaux de l'interface et leur type ainsi que leur mode (in, out, in out, buffer).

La description interne s'appelle une architecture. Le langage VHDL prévoit plusieurs possibilités de décrire une même entité avec différentes architectures.

L'architecture comprend deux parties: une partie déclarative et une description VHDL structurelle ou comportementale. Les déclarations concernent surtout les signaux internes non visibles de l'extérieur mais utiles pour la connexion des différents éléments composant l'architecture. Enfin, l'architecture en elle-même est constituée de trois types d'éléments: des instructions concurrentes, des instances des composants, et des descriptions algorithmiques appelées processus [13].

3.4 Etapes nécessaires au développement d'un projet sur FPGA

Le but d'un langage de description matériel tel que le VHDL est de faciliter le développement d'un circuit numérique en fournissant une méthode rigoureuse de description du fonctionnement et de l'architecture du circuit désirée. L'idée est de ne pas avoir à réaliser (fondre) un composant réel, en utilisant à la place des outils de développement permettant de vérifier le fonctionnement attendu. Ce langage permet en effet d'utiliser des simulateurs dont le rôle est de tester le fonctionnement décrit par le concepteur et de synthétiser cette description matérielle pour obtenir un composant réalisant les fonctions désirées, à l'aide d'éléments logiques concrets (portes logiques, bascules ou registres). Ceux-ci seront implémentés dans les éléments programmables des FPGA. Après la synthèse viennent les phases de: placement, routage Figure 3.5 [10,14].

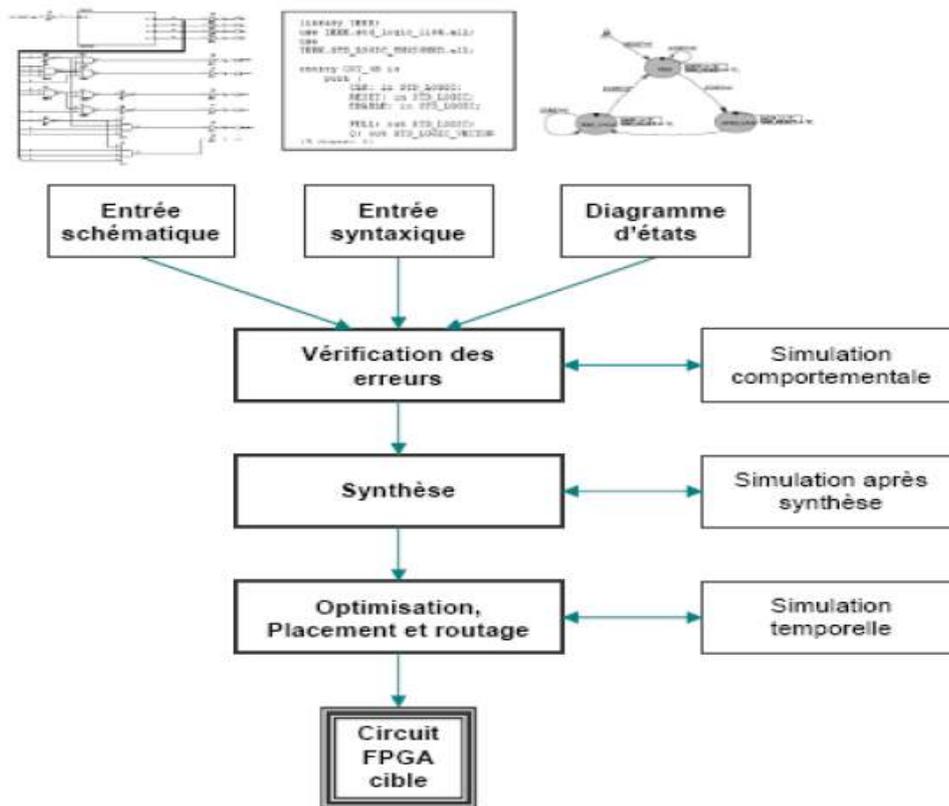


Figure 3.5. Organisation fonctionnelle de développement d'un projet sur circuit FPGA

3.4.1 Saisie du texte VHDL

La saisie du texte VHDL se fait sur le logiciel «ISE Xilinx Project Navigator». Ce logiciel propose une palette d'outils permettant d'effectuer toutes les étapes nécessaires au développement d'un projet sur circuit FPGA. Il possède également des outils permettant de mettre au point une entrée schématique ou de créer des diagrammes d'état, qui peuvent être utilisés comme entrée au lieu du texte VHDL.

La Figure 3.6.a montré comment se présente le logiciel «ISE Xilinx Project Navigator».

On commence par créer un projet, ensuite inclure des fichiers sources dans lesquels il faut saisir le texte VHDL désiré. On peut inclure autant de sources qu'on veut dans un projet.

3.4.2Vérification des erreurs

Une fois qu'un composant a été décrit, il doit être compilé avant de le simuler. Comme pour tous les langages de programmation, le processus de compilation consiste à passer d'une représentation textuelle, compréhensible par l'homme et l'ordinateur à une représentation interne, uniquement utilisable par le simulateur. En VHDL, le résultat d'une compilation vient peupler une bibliothèque, éventuellement organisée en plusieurs compartiments appelés «packages». A chaque bibliothèque est associé un répertoire (dossier) dans le système de fichier de l'ordinateur. Par défaut le résultat de la compilation sera mis dans une bibliothèque de travail «work».

Cette étape est effectuée en appuyant sur le bouton « check syntax ». Elle permet de vérifier les erreurs (errors) de syntaxe du texte VHDL et d'afficher les différentes alarmes (warnings) liées au programme. Cette étape permet donc de valider la syntaxe du programme et de générer la « netlist », qui est un fichier contenant la description de l'application sous forme d'équations logiques.

3.4.3Simulation

Le simulateur utilisé est le « ISim Simulator », Figure 3.6.b. La simulation permet de vérifier le comportement d'un design avant ou après implémentation dans le composant cible. Lors de l'étape de simulation comportementale, on valide l'application indépendamment de l'architecture et des temps de propagation du futur circuit cible. La phase de simulation après synthèse valide l'application sur l'architecture du circuit cible FPGA.

3.4.4Synthèse

La synthèse permet de réaliser l'implémentation physique d'un projet. Le synthétiseur a pour rôle de convertir le projet, en fonction du type du circuit FPGA cible utilisé, en portes logiques et bascules de base. L'outil « View RTL Schematic » permet de visualiser les schémas électroniques équivalents générés par le synthétiseur Figure 3.6.c.

3.5 Les étapes de conception d'application sur FPGA

La conception, la description et la simulation des composants et systèmes en langage VHDL (Very High Description Language) présentent deux aspects distincts pour un modèle.

- Une vue externe du circuit: appelée « Spécification ».
- Une vue interne du circuit: appelée « Architecture ».

Chacune de ces deux parties sera une unité de conception et ce sont ces unités qui constituent la bibliothèque VHDL. Une bibliothèque peut compter des centaines d'unités de conception séparées qui permettent de modifier facilement le fonctionnement interne.

Le VHDL est un langage très modulaire, on peut ne pas écrire de longues descriptions, mais des unités plus petites et hiérarchisées; certaines parties de ces descriptions peuvent être compilées séparément; elles sont suffisantes en elles-mêmes pour être comprises. Ce sont les unités de conception. On peut dire que:

- Le circuit est considéré comme une entité (modèle).
- Cette entité peut être composée par d'autres entités.
- Chaque entité est composée de deux parties, interface et corps. Elle permet d'avoir plusieurs corps pour une même interface.

Parmi les avantages du VHDL nous insistons sur les aspects suivants:

- Le langage permet une description hiérarchique: un système peut être modélisé par un ensemble de composants, lequel, à son tour, peut être modélisé par l'intermédiaire d'un ensemble de sous composants.
- Le langage n'est pas spécifique à une technologie particulière mais peut quand même supporter des caractéristiques spécifiques à une technologie cible.

- Le VHDL permet des descriptions génériques ou paramétrées. Ces caractéristiques seront largement exploitées dans la description du réseau de neurone.

L'outil de ISE Simulator utilisé est l'un des logiciels de CAO les plus répandus, sur les plateformes de type PC pour la mise en œuvre des circuits programmables FPGA et qui réalisent la simulation.

La conception se passe principalement en trois phases:

- La saisie du circuit.
- L'implémentation.
- La configuration du composant.

Auxquelles il faut ajouter les phases de vérification:

- La simulation fonctionnelle: permet de vérifier la validité du circuit par rapport au cahier des charges, celle-ci ne tient pas compte des capacités de liaison dues au routage entre les différentes cellules.

La simulation temporelle: vérifie la fonctionnalité du circuit en prenant en compte les longueurs d'interconnexion et les retards apportés par les capacités liées au routage.

3.6 Conclusion:

Le présent chapitre a été consacré à l'étude de l'architecture des circuits logiques programmables FPGAs en particulier celle retenue par Xilinx. Nous avons également présenté les étapes nécessaires au développement d'un projet sur un circuit FPGA.

Finalement, nous avons présenté les différents modes de simulation d'une description VHDL, les étapes principales de l'implémentation du circuit dans une cible FPGA ainsi que l'apport des FPGA aux problèmes de contrôle.

Le prochain chapitre contient la description hardware du feu tricolore, les différentes techniques utilisées pour implémenter ce feu ainsi que les résultats obtenus.

Les résultats obtenue au niveau de la simulation sous démontre l'efficacité de notre algorithme, ceci nous permet donc de passé à la description des différent parties de notre architecture pour une implémentation Hardware sur un circuit FPGA de Xili

Chapitre 4 Conception et implémentation

4.1 Introduction

Considéré comme l'un des meilleurs outils offerts sur le marché, le circuit FPGA (**Field Programmable GateArray**) de Xilinx (voir annexe) offre une panoplie de composants, ainsi qu'un système de développement ISE (*IntegratedSoftware Engineering*) *Fondation* offrant d'énormes possibilités de développement (synthèse, simulation et implémentation automatique).

Dans ce chapitre, il sera question de simulation et implémentation de l'architecture proposée.

La méthodologie de base pour la conception sur circuit FPGA de Xilinx se décompose en quatre étapes:

1. Description de l'architecture.
2. Simulation fonctionnelle.
3. Synthèse et implémentation.
4. Simulation temporelle et vérification.

L'architecture que nous avons réalisée a été conçue dans l'environnement ISE 9.1 de Xilinx. Le processus d'implémentation convoite certaines phases de vérification. En premier lieu, l'architecture doit être décrite en langage VHDL (Very High Description Langage). Après la description VHDL de l'architecture et sa compilation, une simulation fonctionnelle est introduite par le simulateur (ISE simulator), dont les résultats figurent sous forme de chronogramme. Une fois que la simulation fonctionnelle est validée, une phase de synthèse va permettre de donner un rapport bien détaillé sur les ressources du circuit FPGA consommées, et sur la fréquence maximale de fonctionnement. Finalement, une phase de placement et routage éclate

schématiquement la surface consommée par l'architecture globale et configure les routages d'interconnexion entre les différents blocs des ressources du circuit FPGA. Dans la simulation temporelle, on vérifie si le circuit obtenu respecte les contraintes temporelles et utilise les délais des portes et les délais dans les interconnexions pour calculer la vitesse maximale. La figure 4.1 regroupe toutes les étapes d'une conception sur circuit FPGA.

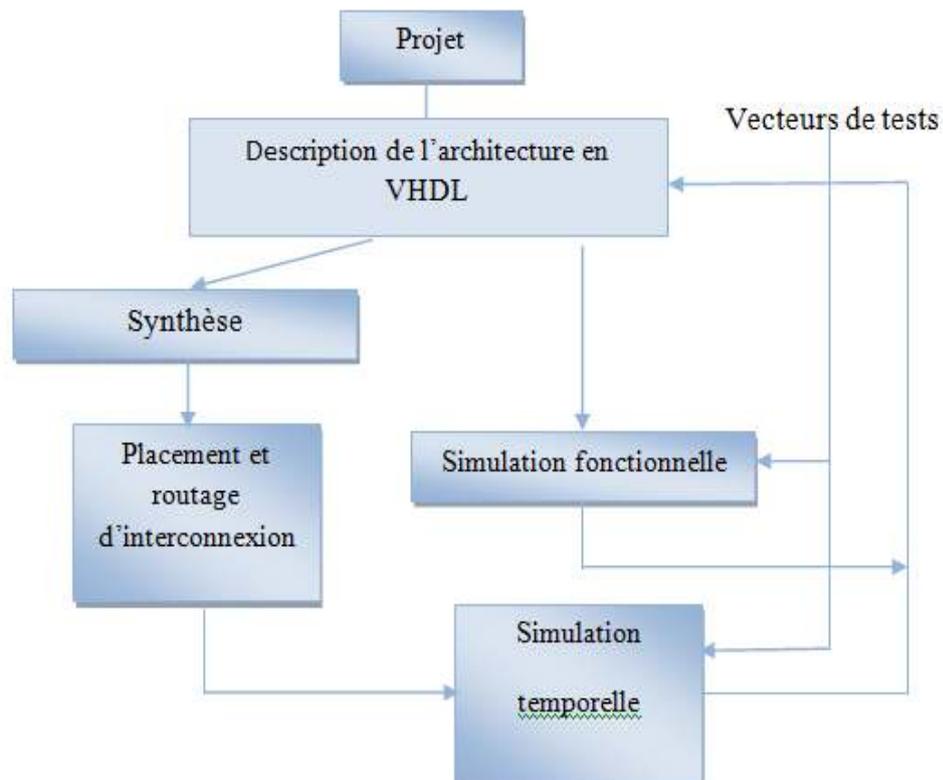


Figure 4.1. Schéma des étapes de conception sur un circuit FPGA.

4.2 Description de l'architecture globale

L'architecture que nous avons proposée est composée de trois blocs définis comme suit:

1. Bloc générateur d'horloge (diviseur).
2. Maintient du signal de télécommande.
3. Bloc commande logique (Le séquenceur).

L'architecture globale est montrée par la figure 4.2 :

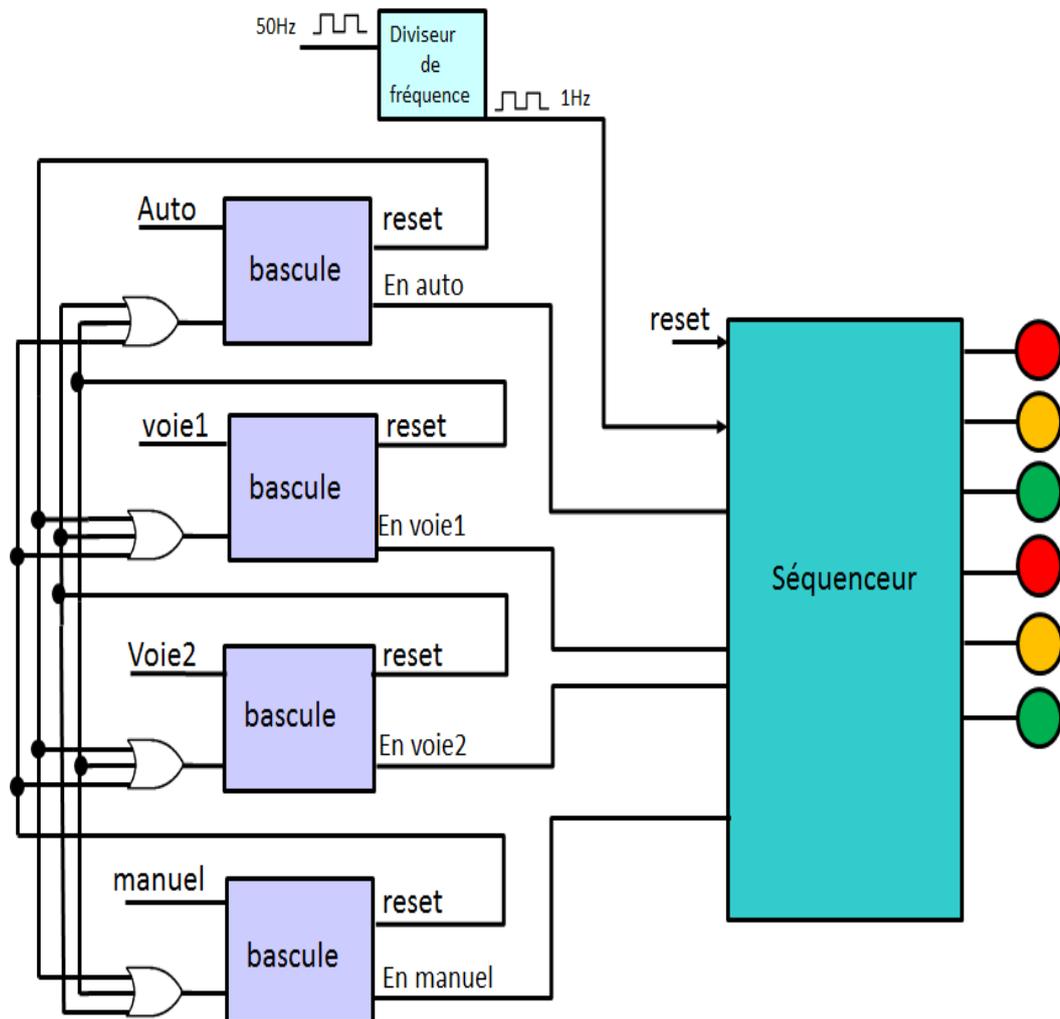


Figure 4.2. Architecture globale.

Le schéma de fonctionnement traduit de manière graphique la fonction d'usage de l'objet technique en tenant compte des solutions techniques et technologiques retenues pour réaliser l'objet.

4.2.1 Bloc générateur d'horloge

Le bloc générateur d'horloge est le premier bloc de notre conception sous FPGA. C'est un bloc qui reçoit à son entrée le signal *reset* et l'horloge principal de la carte. En sortie, il génère un signal *freq* qui sera utilisé dans le bloc séquenceur.

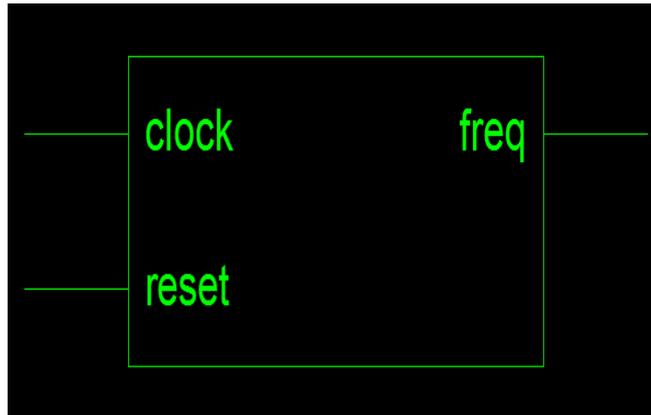


Figure 4.3. Architecture du Bloc Diviseur de fréquence.

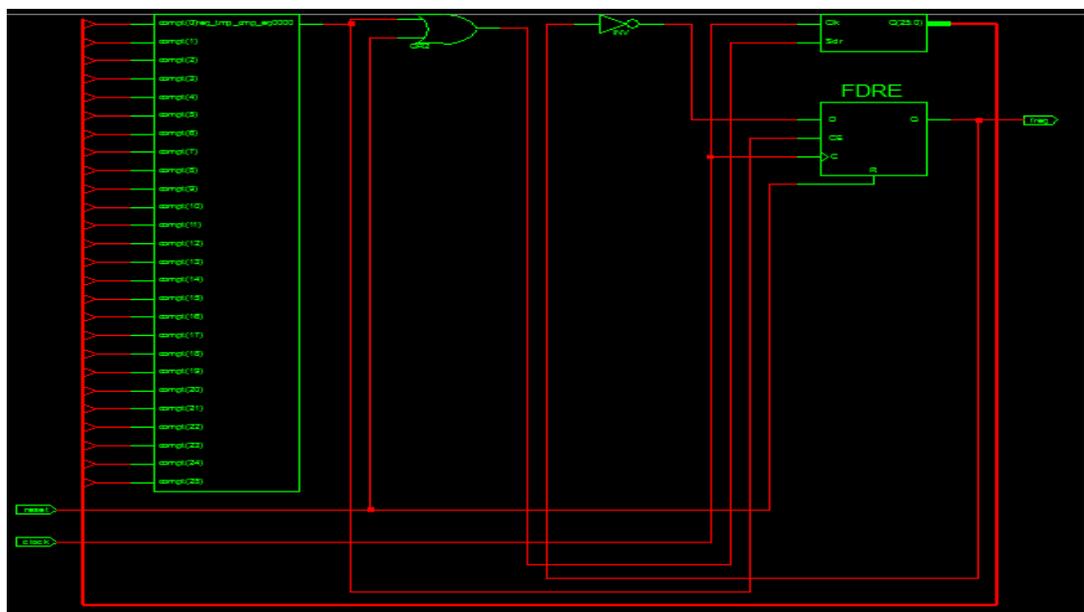


Figure 4.4. Architecture interne du Bloc Diviseur de fréquence.

4.2.2 Sauvegarde du signal de télécommande

C'est le bloc de la sélection des modes (auto, voie1, voie2 ou manuel) qui nécessite une intervention externe à travers un bouton poussoir via une télécommande RF.

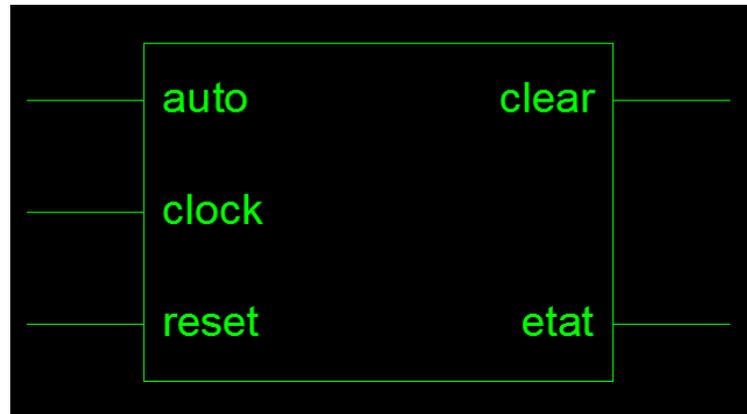


Figure 4.5. Architecture du Bloc Sauvegarde.

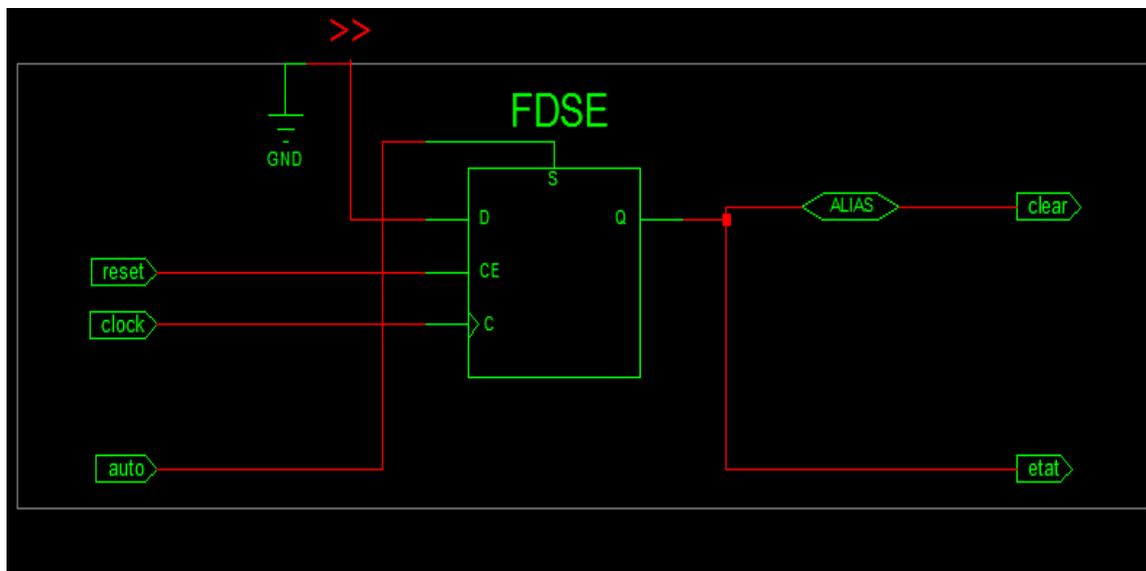


Figure 4.6. Architecture interne du Bloc Sauvegarde.

4.2.3 Bloc commande logique (Le séquenceur)

Ce bloc est le séquenceur qui consiste à générer les différents signaux de contrôle des feux (orange, rouge et vert) des deux voies 1 et 2.

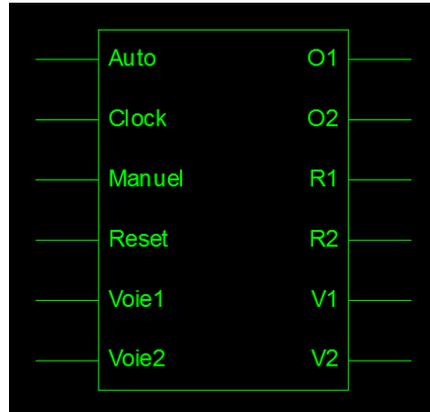


Figure 4.7. Architecteur du bloc commande logique (Le séquenceur)

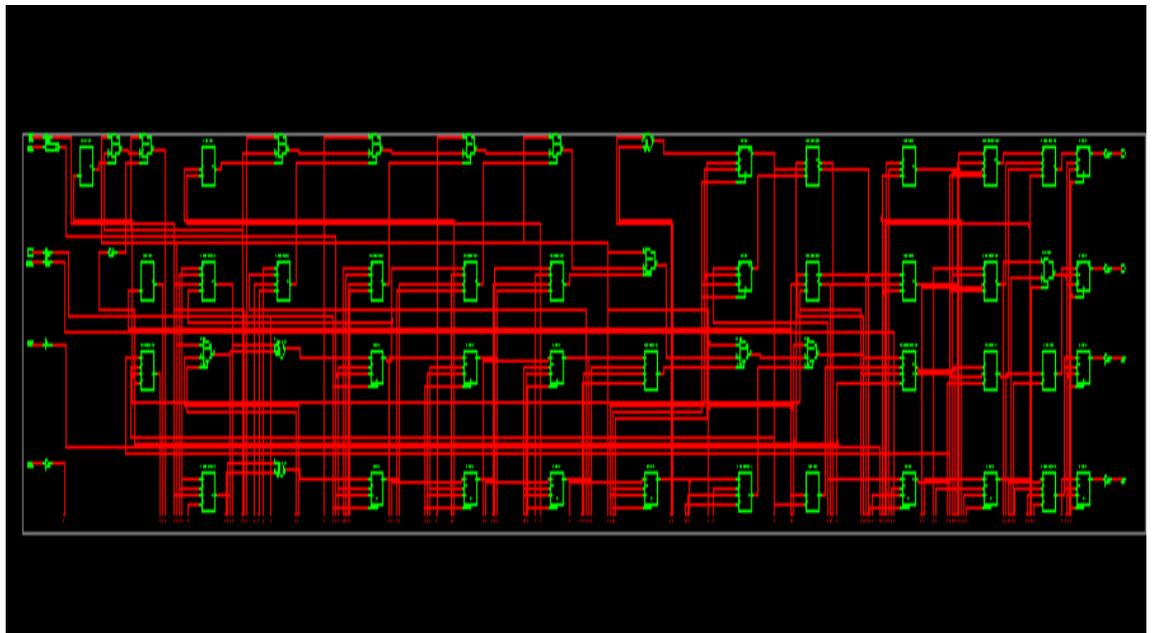


Figure 4.8. Architecteur interne du bloc commande logique (Le séquenceur).

4.3 Simulations et Résultats

Les étapes suivies pour la simulation de l'implémentation de feux tricolores se résumant comme suit:

- Description de l'architecture en langage VHDL sous l'outil ISE 9.1 et la simulation par l'outil ISE simulator.
- Synthèse de l'architecture.
- Implémentation de l'architecture sur le circuit XC3S250E_4FT256 Cde la famille Spartan3E.

Cette partie nous permet de vérifier le bon déroulement de notre système avant de le charger dans la carte FPGA. Nous pouvons détecter, de cette manière-là toutes les erreurs qui auraient pu se glisser soit dans notre code soit dans notre saisie de schémas.

Une deuxième étape sera de simuler notre système dans le domaine temporel pour tenir compte de la réalité des composants.

4.3.1 Résultat de la simulation du Blocs générateur d'horloge (diviseur).

Le résultat de la simulation fonctionnelle du Bloc générateurd'horloge est représenté sur la figure 4.9

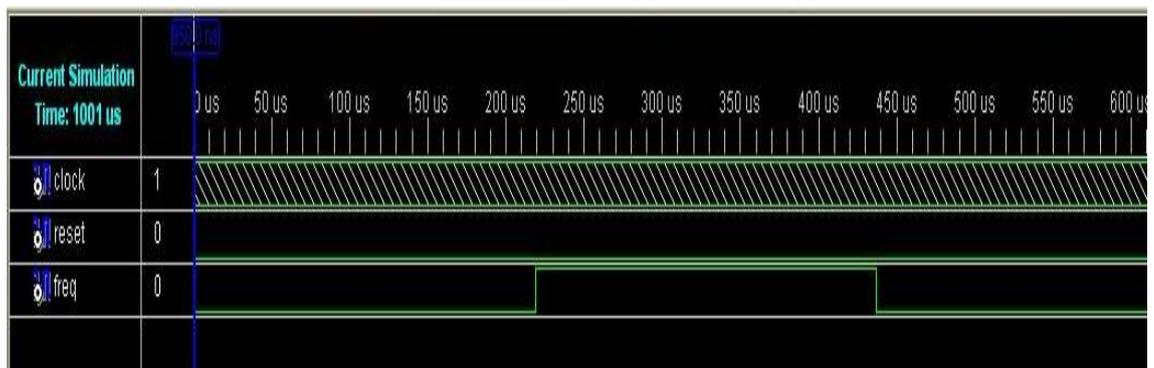


Figure 4.9: Chronogramme de bloc générateur d'horloge .

4.3.2 Résultats de la simulation du Bloc Maintient du signal de télécommande.

Le résultat de la simulation fonctionnelle du Bloc Maintientdu signal de télécommandeest représenté sur la figure 4.10.

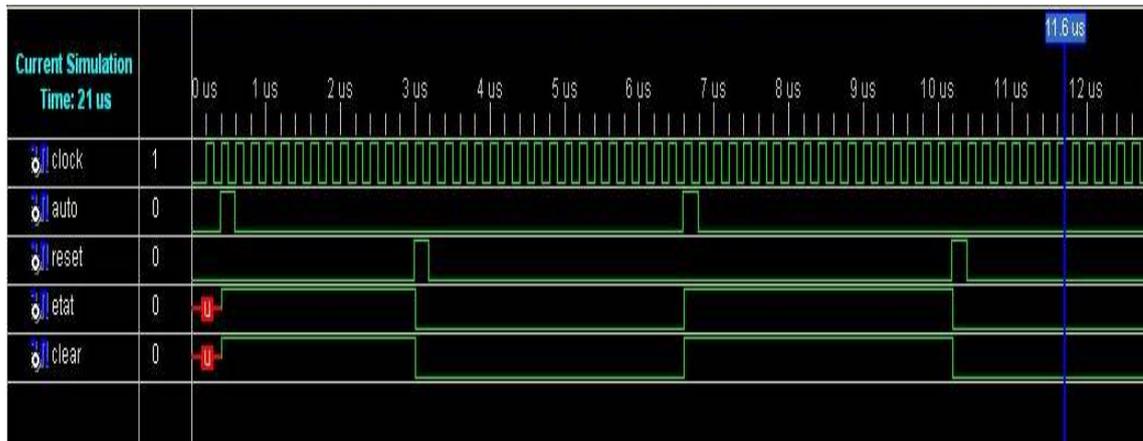


Figure 4.10. Chronogramme du Bloc Maintient du signal.

4.3.3 Résultats de la Simulation du séquenceur

Le résultat de la simulation fonctionnelle du Bloc séquenceur est représenté sur la figure 4.11.

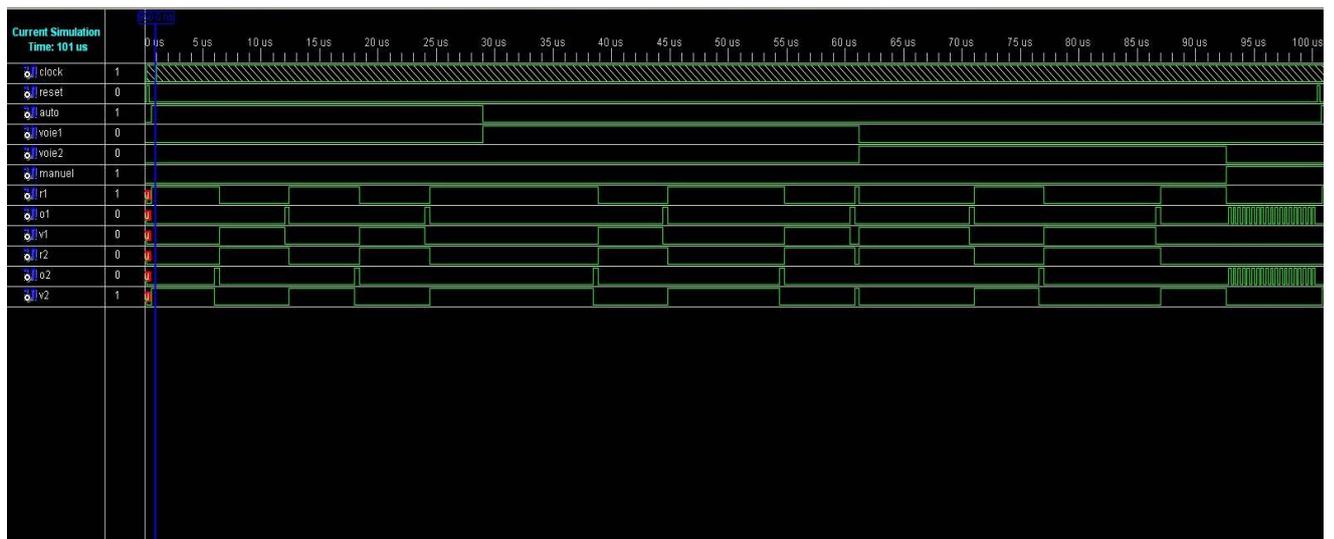


Figure 4.11 Chronogramme du séquenceur.

4.3.3 Résultats de la Simulation finale

Le résultat de la simulation fonctionnelle du Bloc finale est représenté sur la figure 4.12 :

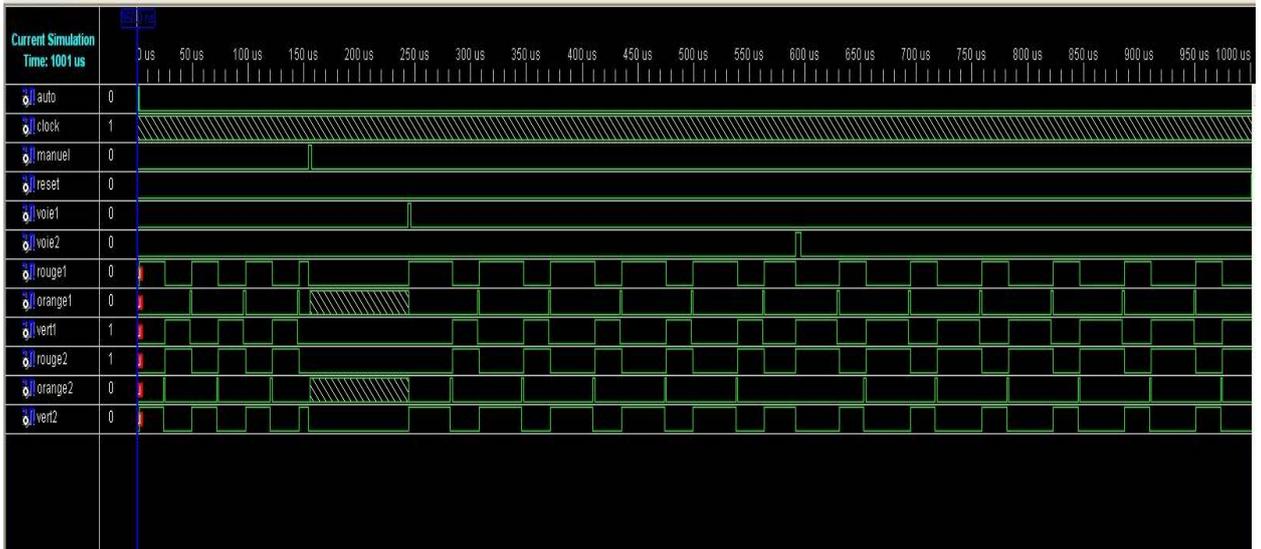


Figure 4.12 Chronogramme finale.

4.4 Simulation sous TINA

Après avoir simulé notre programme sous ISE, on a vérifié le bon fonctionnement de notre système en simulant ce dernier sous TINA. La figure suivante représente le résultat de cette simulation.

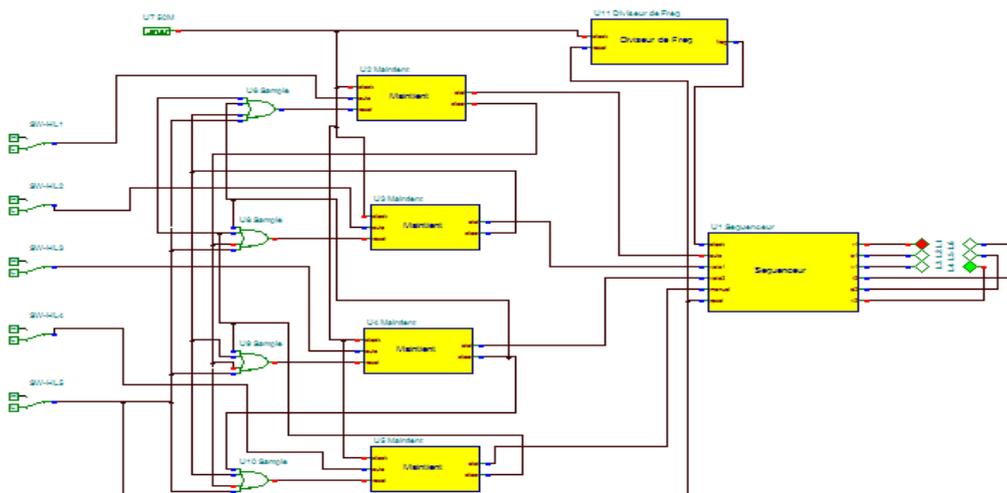


Figure 4.13 Simulation sous TINA.

4.5 Synthèse et implémentation

✓ Synthèse

Cette étape est une synthèse des différents blocs de l'architecture globale. Ils sont exécutés par l'outil de synthèse XST et permet d'avoir une estimation du taux d'occupation et du temps d'exécution. Ensuite, nous sommes passés à l'implémentation qui est la dernière étape dans la conception et qui permet le placement et le routage.

La figure suivante présente le Layout de l'architecture du système.

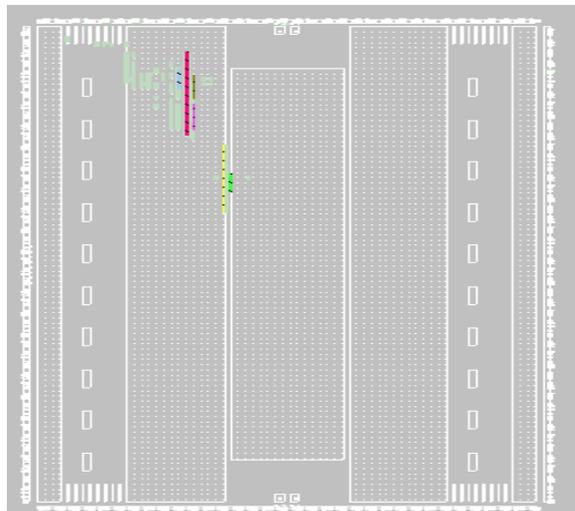


Figure 4.14 Layout de l'architecture du système

✓ Implémentation

Pour implémenter un circuit sur un FPGA, il y a trois éléments indispensables pour parvenir :

- Disposer de la description logique du circuit (schématique, diagrammes d'états ou code tel VHDL).
- L'environnement de développement est choisi en fonction du composant sur lequel le circuit sera implémenté spécialement le logiciel du fabricant du composant (ISE :XILINX Integrated Software Environment).
- Une carte de développement pour implémenter le circuit et vérifier son comportement.

Le produit final est représenté dans la figure 4.14, on a la carte spartan3e, la télécommande RF « pour choisir les modes », La carte de réception, et les panneaux des feux.



Figure 4.15 système de contrôle des feux à distance

L'architecture proposée a été implémenté avec succès, ainsi que les blocs d'émission, réception, comme le montre la figure suivante.

4.6 Conclusion

Dans ce chapitre, nous avons présenté la méthodologie de conception de notre architecture. Nous avons également présenté, les résultats obtenus de la simulation et de l'implémentation sur le circuit FPGA XC3S250E_4FT256C de la familleSpartan3E. Pour cela, nous sommes passés par plusieurs étapes, chacune d'entre elles utilise un outil bien spécifique.

Conclusion générale

La réalisation de ce mémoire nous a permis de faire des recherches approfondies dans le domaine de l'implémentation sous FPGA, en implémentant sous XILINX ISE 9.1 un système de contrôle de feux tricolores commandé à distance par une télécommande via Radio fréquence, cette dernière permet d'avoir quatre mode de fonctionnement différents,(mode automatique, mode manuel, la priorité à la voie 1, la priorité à la voie 2).

A cet effet, nous avons commencé par une étude des principales techniques de feux tricolores ainsi que les différents blocs qui constituent ce système. Après avoir introduit les notions générales des principales techniques du feu tricolore, nous avons présenté les différents blocs qui constituent un système de commande en occurrence le bloc de feu.

Ce travail à deux axes principaux, la partie hard consiste à réaliser les blocs d'émission et de réception, qui a été effectué avec succès. La seconde étant la partie soft, dont l'implémentation sous FPGA, accomplie avec succès où le programme a été enregistré d'abord dans l'EPROM de la carte FPGA.

Le système de contrôle des feux tricolores implémenté sous FPGA a été réalisé avec succès. Ce travail peut être amélioré par la réalisation des feux tricolores en mode coordonné.

Conclusion générale

Bibliographie

- [1] <http://www.controletrafic.com>-Contrôle du trafic routier urbain par un réseau sans fil.
- [2] Salmi Ahmed: 'Etude et implémentation de feu tricolore', Ecole Nationale Polytechnique Département d'Electronique Alger, Septembre 2004.
- [3] Hanane Salim : ' gestion de la circulation routière', faculté des sciences de l'ingénieur département d'informatique université El hadj lakhder-Batna, 2010.
- [4] www.pdfactory.comtraffiac // trafic contrôler fpga 2012//.
- [5] René Husson, commande des feux de trafic: ' Université de Belfort-Montbéliard'.
- [6] les feux tricolores de croisement 'Groupe académique STI électronique'. Académie de CAEN.
- [7] De la configuration des circuits électroniques.
[Islwww.epfl.ch/~thoma/research/thèse](http://www.epfl.ch/~thoma/research/thèse).
- [8] Introduction au microprocesseur et circuits assimilés cinquième partie (5/5) :
Les composants FPGA :
http://pagesperso-orange.fr/michel.hubin/physique/microp/chap_mp5.htm
<http://pagesperso-orange.fr/jacques.weber/>
http://uuu.enseirb.fr/~renaud/files/cours_intro_FPGA_2006.pdf
- [9] E. Sanchez. Les circuits FPGA. Laboratoire de Systèmes Logiques, Ecole

Polytechnique Fédérale de Lausanne.

http://islwww.epfl.ch/pages/teaching/cours_isl/cp_es/fpga.pdf

[10] F.CHEKIRED. Etude et implémentation d'une commande MPPT neuro flou sur FPGA. Mémoire de Magister. ENP, Algérie 2008.

[11] Introduction à la Synthèse logique V.H.D.L.

<ftp://ftp.discip.crdp.accaen.fr/discip/crgelec/Cours/vhdl.pdf>

[12] MEMEC Design. Spartan3E V2MB1000 Development Bord user's Guide.

<http://legacy.memec.com/solutions/refernce/xilinx>, Décembre 2002.

[13] J. O. Clein, J. Cachan. L'Essentiel du langage VHDL.

www.ief.u-psud.fr/~jok/iut/VHDL/COURS_VHDL.PDF

[14] H. Mekki. Modélisation d'Un Système commande en Utilisant Le Langage VHDL en vue de l'Implémentation Sur Un Circuit FPGA. Mémoire de Magister,

Université de Blida, Algérie, 2009.

1. Architecture interne de spartan3E xilinx.

Les circuits FPGA est un réseau de blocs logique programmables (CLB) organisés dans une structure matricielle et de blocs d'entrée / sortie (IOB) placés sur la périphérie du circuit FPGA, le tout est interconnectées par un réseau d'interconnexions programmables. La figure suivante la représentation de l'architecture interne d'un FPGA.

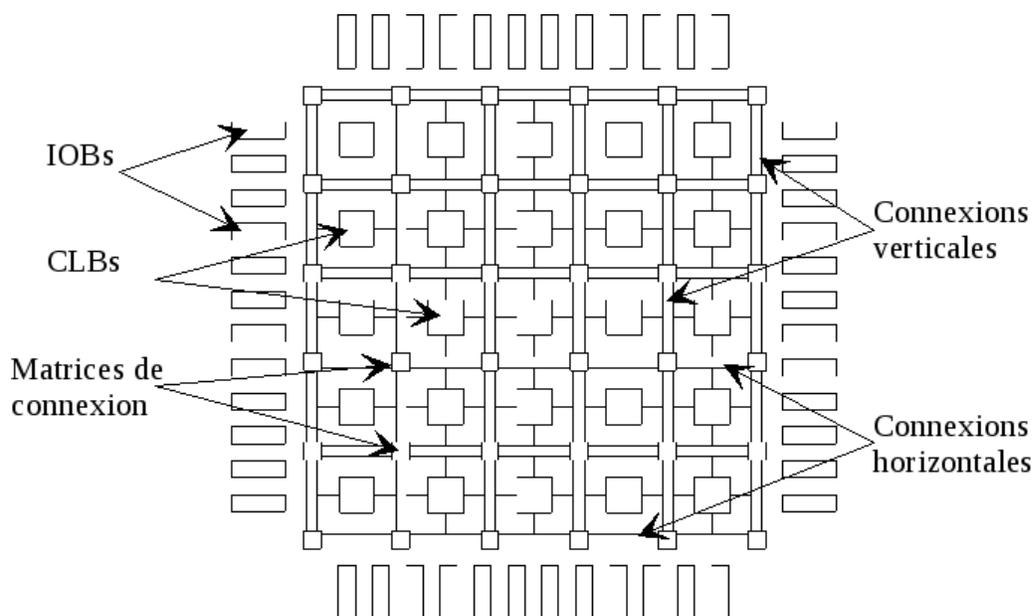


Figure 1: Architecture interne d'un FPGA.

2. La famille Spartan3E

La famille Spartan3E a été conçue pour réaliser des conceptions à faible ou grande densité d'intégration et qui exigent des performances élevées (elle peut atteindre jusqu'à 10 millions de portes logiques). La fréquence de son utilisation peut être portée à 420 Mhz. Actuellement, cette famille est utilisée dans plusieurs applications tels que: les réseaux, la télécommunication, la vidéo et les applications DSP (Digital Signal Processing).

L'architecture d'un circuit Spartan3E est montrée sur la figure 3. Celle-ci se présente comme un réseau de bloc logique configurable. Ces derniers sont des blocs Input/Output (IOB) et des blocs logiques configurables (CLB).

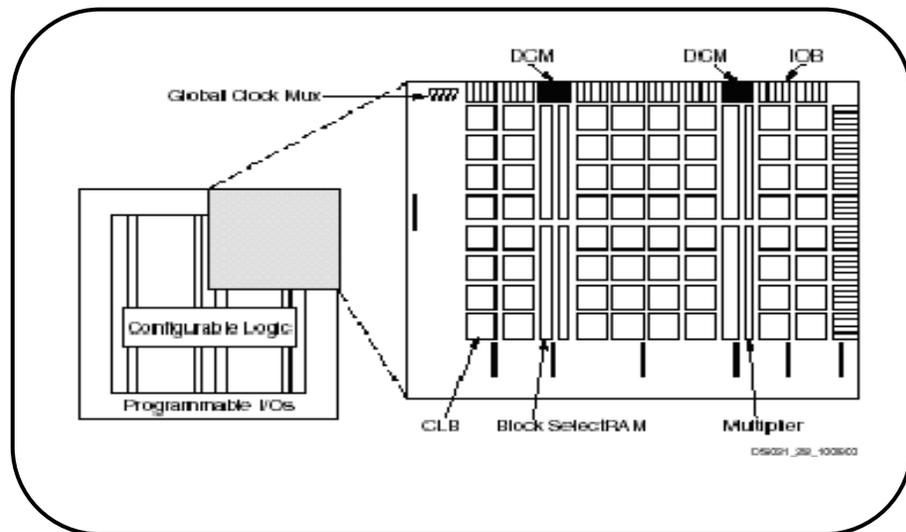


Figure 2: Architecture interne d'un Spartan3E.

2.1 Le bloc logique interne configurable

La logique interne configurable inclut quatre éléments majeurs organisés sous une forme régulière et un chemin dédié à la propagation de la retenue. Les éléments en question sont:

- Bloc Logique Configurable (CLB).

- Bloc Select RAM 18 K bits à doubles ports.
- Bloc multiplie.
- DCM (Digital Clock Manager).

La connexion entre tout ces blocs est assurée d'une manière programmable grâce à des ressources d'interconnexions configurables GRM " General Routing Matrix " .

2.2 Bloc d'entrées/sorties (IOB)

Les blocs d'entrée/sortie fournissent une interface entre les broches externes du circuit et la logique interne. Deux IOBs peuvent être utilisés comme une paire différentielle. Celle-ci est souvent connectée à une même matrice d'interconnexion .Comme cela est illustrée sur la figure suivante.

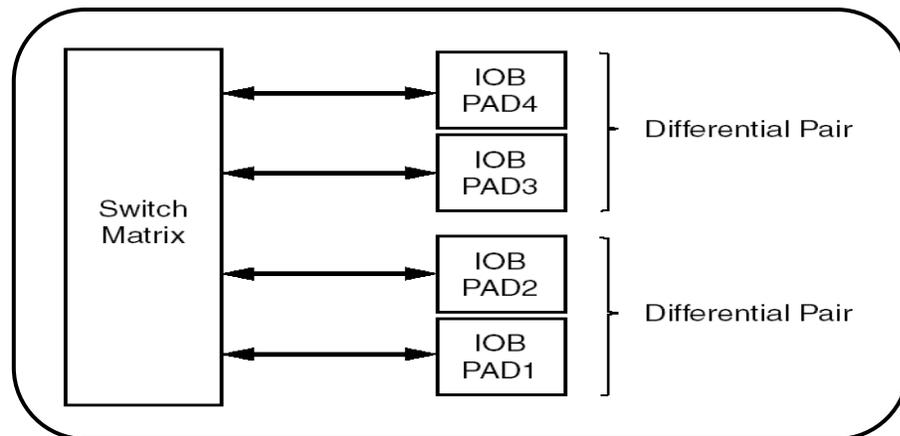


Figure 3: Connexion des IOBs à une matrice d'interconnexion.

Les IOBs sont configurés soit en : entrée, sortie, bidirectionnels ou en haut impédance. Ils incluent six bascules D, leur utilisation est souvent recommandée pour la synchronisation des signaux de communication. Chaque bascule peut être configurée comme étant:

- Sensible au front montant ou descendant d'une horloge.
- Sensible au niveau d'une horloge.

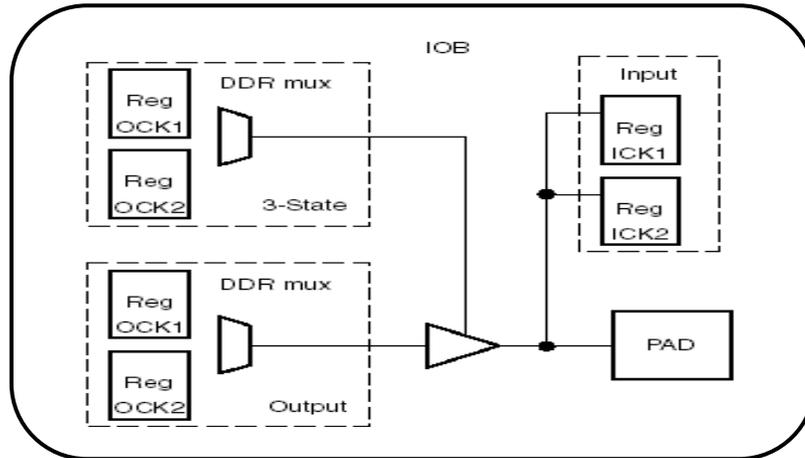


Figure 3: Bloc IOB de Spartan3E.

Leur utilisation suivant les trois états, l'une ou les deux bascules peuvent fournir un chemin à double débit. Ceci est réalisé par l'utilisation de deux horloges déphasées de 180° et générées par une DCM. Cela est montré dans la figure.

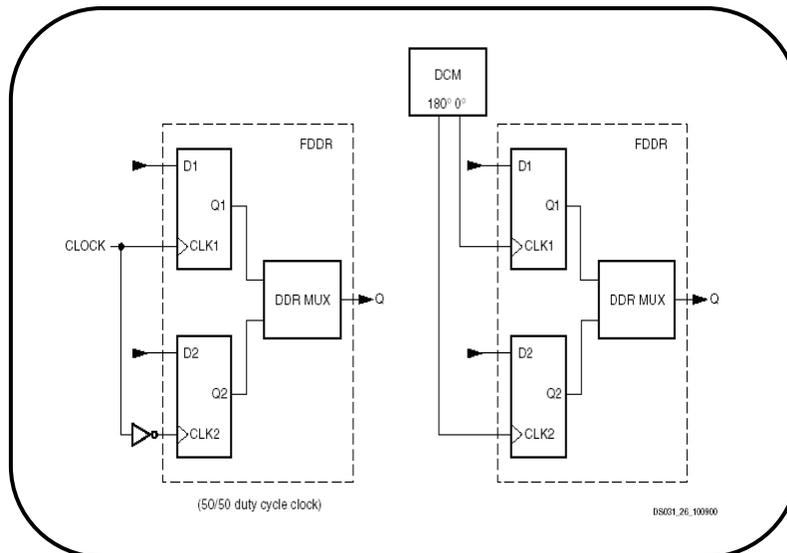


Figure 4: Registre à double débit.

2.3 Bloc logique configurable (CLB)

Un bloc logique programmable est constitué principalement de cellules logiques (LC : Logic Cell) qui représentent les éléments de base constituant un CLB. La Figure 5. Représente un schéma simplifié d'un CLB de la carte FPGA Spartan. La combinaison de deux LCs forme ce qu'on appelle un slice et la combinaison de quatre slices nous donne un CLB.

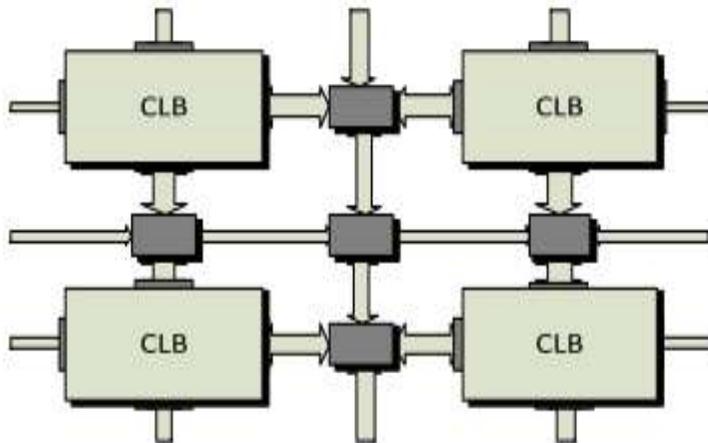


Figure 5: CLB de Spartan3E.

2.3.1. Description d'un Slice

L'architecture d'un Slice se présente comme une structure symétrique. Cette caractéristique rend ce dernier flexible à l'optimisation des applications données. Chaque Slice est composé de:

- Deux générateurs de fonction F et G à quatre entrées et une sortie.
- Deux éléments de stockage (bascule D).
- Des portes logiques (XOR) dédiées à l'implémentation des Full Adders.
- Une logique pour la propagation de la retenue.

➤ Des multiplexeurs.

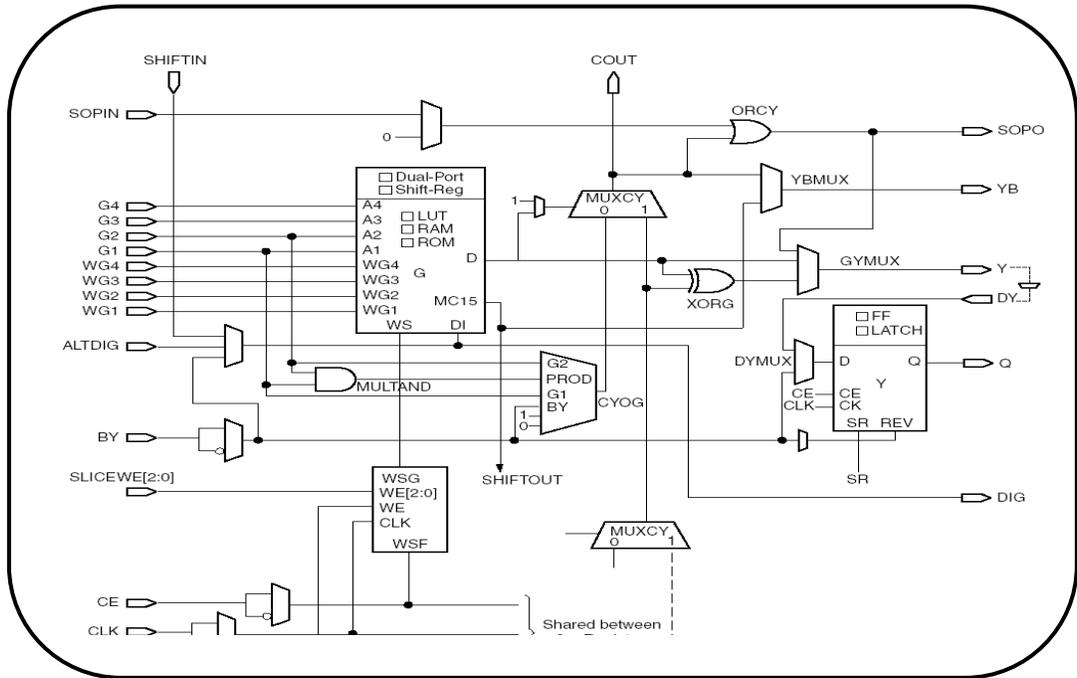


Figure 6: Schéma détaillé de la moitié supérieure d'un Slice.

Les générateurs de fonction peuvent être configurés comme: des Look Up Table (LUT) à quatre entrées, des RAMs à 16 bits ou des registres à décalage de 16 bits.

Les sorties des générateurs de fonction sont considérées comme étant les sorties du Slice utilisé ou Les entrées des deux bascules (D).

2.3.2 Routage global et interconnexion

Les blocs logiques configurables (CLB) et les blocs d'entrée/sortie (IOB) sont connectés entre eux par des dispositifs variés qui constituent un réseau d'interconnexions programmable. Les connexions internes dans les circuits FPGA sont composées de segments métallisés. Parallèlement à ces lignes, nous trouvons des matrices programmables réparties sur la totalité du circuit, horizontalement et verticalement

entre les divers CLB. Elles permettent les connexions entre les diverses lignes, celles-ci sont assurées par des transistors MOS dont l'état est contrôlé par des cellules de mémoire vive ou RAM (Random Access Memory). Le rôle de ces interconnexions est de relier avec un maximum d'efficacité les blocs logiques et les blocs d'entrées/sorties afin que le taux d'utilisation dans un circuit donné soit le plus élevé possible. Pour parvenir à cet objectif, Xilinx propose trois sortes d'interconnexions selon la longueur et la destination des liaisons. Nous disposons:

- d'interconnexions à usage général (matrices),
- d'interconnexions directes,
- de longues lignes.

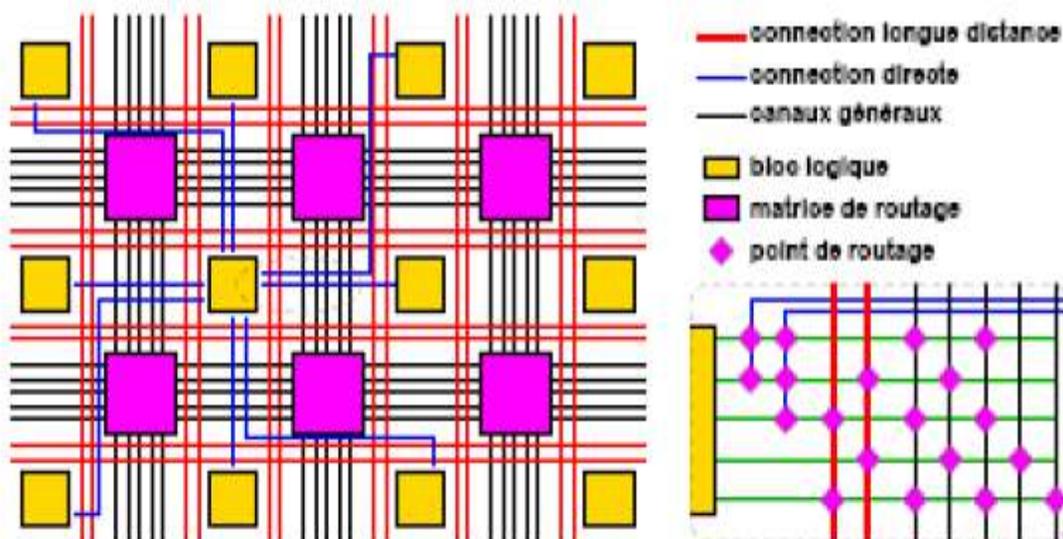


Figure 7 : Réseau d'interconnexions dans un circuit FPGA

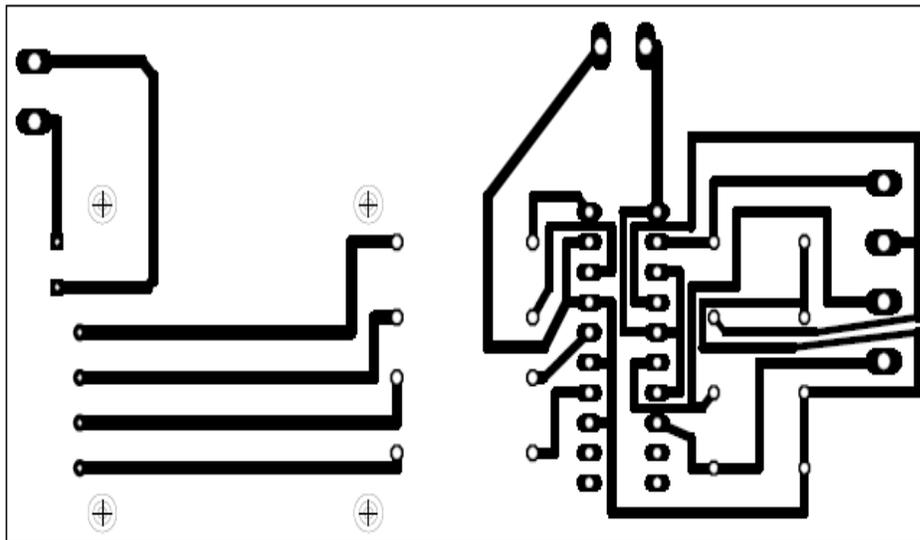


Figure 8 : Carte protection du FPGA

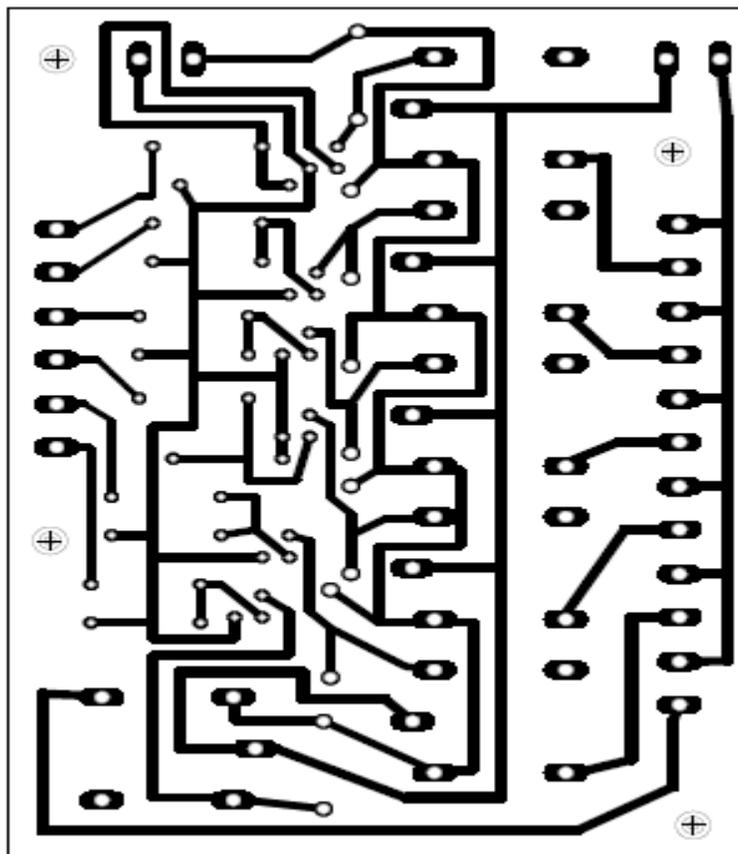


Figure 9 : Carte de puissance