
RAPPORT DE PROJET

Projet d'électronique n°1 : Le contrôleur de feux tricolores

Auteurs:

Charlotte TA CASALI
Axelle ROUZIER
Noé FOUASSIER
Adame DROUIN

Enseignant :

M. Mokhber Arash

Nous attestons que ce travail est original, qu'il est le fruit d'un travail commun au groupe et qu'il a été rédigé de manière autonome.

Paris, le 23/10/2022

Table des matières

- I. Objectif**
- II. Glossaire**
 - A. Termes**
 - B. Acronymes**
- III. L'équipe 4**
 - A. Présentation de l'équipe**
 - B. Organisation de l'équipe**
 - C. Diagramme de GANTT**
- IV. Contexte et problématique**
 - A. Contexte**
 - B. Problématique**
 - C. Spécifications techniques**
- V. Conception**
 - A. Architecture fonctionnelle**
 - B. Architecture matérielle**
 - C. Architecture logicielle**
- VI. Développement**
 - A. Module 1**
 - B. Module 2**
 - C. Module 3**
- VII. Tests et validation**
 - A. Module 1**
 - B. Module 2**
 - C. Module 3**
- VIII. Bilan**
 - A. État d'avancement**
 - B. Pertinence de la solution technique**
 - C. Bilan sur le travail d'équipe**
- IX. Sources**

I. Objectif

Quel est l'objectif de ce document ?

Le rapport est un élément essentiel d'un projet. Il permet de faire part de notre avancée, de la méthode que nous avons appliquée et des notions que nous avons apprises tout au long du projet. Nous allons vous expliquer comment nous avons réalisé un contrôleur de feux tricolores.

Tout d'abord, il est nécessaire d'introduire la notion de feu tricolore: le feu tricolore a été inventé en 1920 par Léon Fœnquinos. C'est un dispositif qui permet de réguler le trafic routier entre les différents usagers de la route tels que les véhicules, piétons, vélos, etc. Il a pour but d'assurer la sécurité de tous les usagers de la voirie.

Pour notre projet, nous avons traité d'un feu de croisement composé de deux feux tricolores et de deux feux piétons.

Nos feux devaient respecter les contraintes suivantes: le feu piéton devait être vert si le feu voiture était rouge, un décompte du temps restant jusqu'au prochain passage du feu piéton au feu vert devait être affiché pour les piétons. Nous devions également émettre un signal sonore indiquant le temps restant pour traverser ainsi qu'ajouter un mode de maintenance ou les feux clignotent au orange. Enfin, nos feux devaient être réglés par un potentiomètre afin de régler la durée du feu vert en fonction de la circulation routière.

Que va y trouver le lecteur ?

Le lecteur y trouvera une présentation de l'équipe, une brève définition de l'utilité des différents matériaux utilisés, une explication détaillée des différentes étapes de conception et de validation, des graphiques(RTL) et une présentation du résultat final.

II. Glossaire

A. Termes

VHDL	VHDL ou de son nom complet "VHSIC Hardware" est un langage de description matériel qui est utilisé en électronique et qui représente les différents comportement des circuits informatiques.
Machines à état	La machine à état est un concept utilisé dans les programmes informatiques qui va

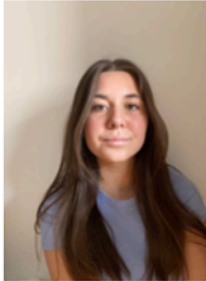
	lire un ensemble d'entrées et passe à un état différent en fonction de ces entrées. Il existe deux types de machine à état : finis et infinis.
Registres à décalage	Un registre à décalage est composé d'un ensemble de bascules reliées entre elles. Ce registre permet de stocker une donnée sur une certaine durée
Afficheurs 7-SEG	Un afficheur 7-SEG est un composant permettant d'afficher un numéro, une lettre ou autre grâce à 7 segments qui s'allument en fonction de la forme désirée

B. Acronymes

VGA	VGA (Video Graphics Array) désigne une interface de connexion permettant de transmettre des signaux vidéo analogique.
ADC	Un ADC (Analog to Digital Converters) est un périphérique permettant de convertir une tension d'entrée en une valeur numérique. Cette image est donc quantifiée, c'est-à-dire qu'elle ne peut prendre qu'un ensemble limité de valeurs.
FPGA	Les FPGA (Field Programmable Gate Arrays) sont des composants reconfigurables et reprogrammables et permet d'exécuter plusieurs opérations en parallèle ce qui permet d'accélérer les calculs

III. L'équipe

A. Présentation de l'équipe



Charlotte se présente comme un point fort dans l'équipe. Elle apporte des solutions aux problèmes pointus, se montre en tant que leader dans le groupe et reste focalisée sur nos objectifs.



Axelle est enthousiaste, entreprenante et organisée, elle trouve des ressources nécessaires et aide à résoudre les problèmes rencontrés.



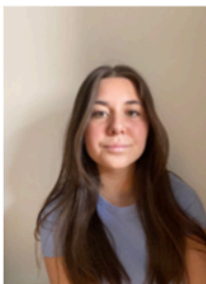
Noé est extrêmement polyvalent et pointilleux. Il possède de bonnes compétences pratiques et informatiques.



Adame est très créatif et possède une grande logique organisationnelle.

B. Organisation de l'équipe

Ce projet a été réalisé grâce à un travail organisé après les cours et par appel à distance.



- Conception de la machine à état
- Diviseur de fréquence
- Édition du RTL sur quartus
- Buzzer
- Rapport de projet



- Diviseur de fréquence
- Édition du RTL sur quartus
- Branchements
- Décompteur
- Rapport de projet



- Diviseur de fréquence
- Édition du RTL sur quartus
- Décompteur
- Rapport de projet



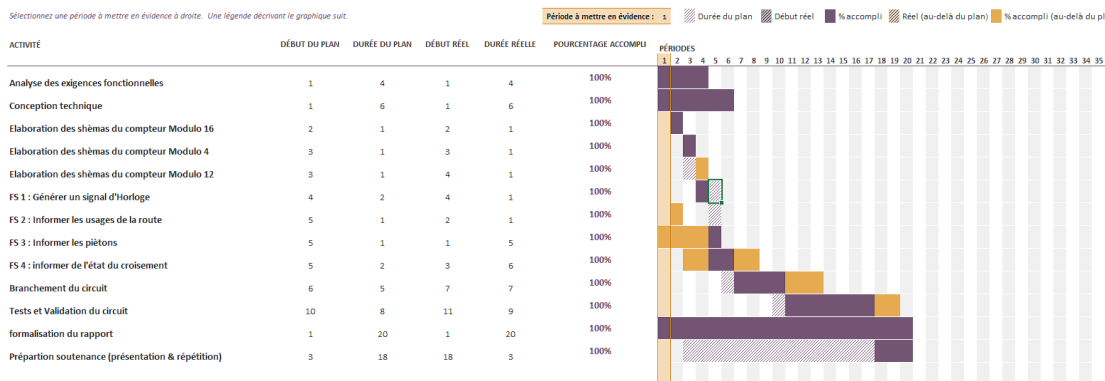
- Diviseur de fréquence
- Édition du RTL sur quartus
- Décompteur
- Rapport de projet

C. Diagramme de GANTT

Pour représenter l'allocation du temps à notre projet, nous avons décidé d'utiliser le diagramme de Gantt suivant :

Contrôleur de feux tricolores - Charlotte - Axelle - Adame et Noé

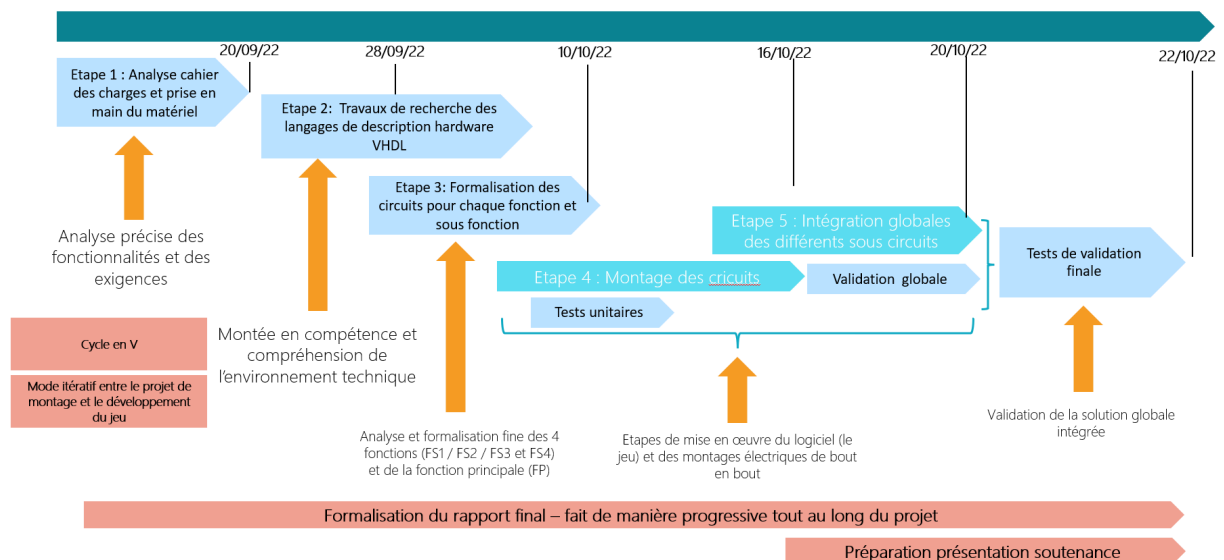
Sélectionnez une période à mettre en évidence à droite. Une légende décrivant le graphique suit.



Nous avons cherché à nous structurer et nous organiser pour assurer un suivi et répondre aux échéances du projet. Nous avons eu deux adaptations majeures par rapport au planning initial :

- Familiarization du logiciel Quartus
- Comprendre le fonctionnement d'un FPGA : De10-lite

. Méthodologie & Planning projet .



IV. Contexte et problématique

A. Contexte

Dans quel contexte s'est déroulé le projet ? Étions-nous capables de le mener à bien ? Avions-nous les compétences, le moyen, le temps ? L'idée du projet était-elle adaptée à notre situation ?

Le projet a été réalisé dans le cadre d'une nouvelle matière qui est dans la continuité du cours d'électronique que nous avons eu l'année précédente. Nous donnant ainsi une base de connaissances que nous avons pu exploiter. Cependant, certaines notions telles que la machine à état ou l'ADC étaient nouvelles pour nous. L'utilisation d'un FPGA tel que le DE10-lite étaient également quelque chose de nouveau que nous avons dû maîtriser en seulement deux TP. En termes de gestion du temps, nous étions déstabilisés par les différents TP et examens qui se préparaient en parallèle. Nous aurions ainsi dû mieux nous organiser afin de gérer cette charge de travail.

B. Problématique

A quelle problématique répond le projet ?

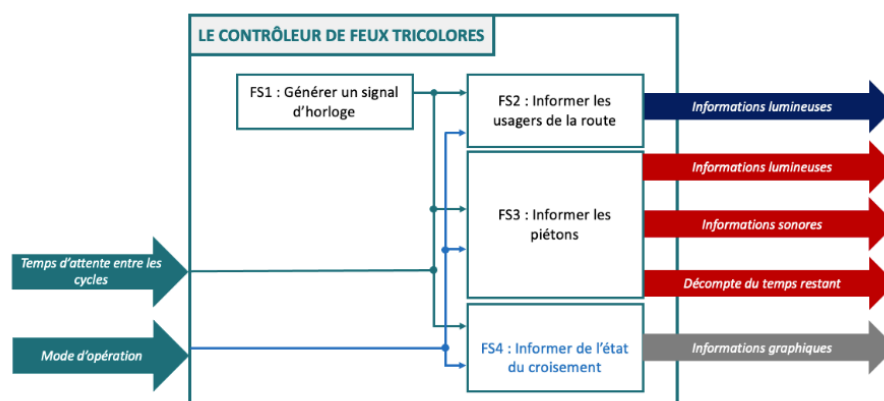
Comment créer un contrôleur de feu tricolore fonctionnel capable d'assurer la sécurité des différents usagers d'une intersection ?

Les problématiques de ce projet sont de différents ordres :

- Appréhender le fonctionnement d'un nouveau logiciel : Quartus.
- Comprendre et maîtriser un nouveau langage, le VHDL.
- Maîtriser et utiliser un nouveau composant : le De10-lite
- Réaliser un montage électronique et finaliser l'intégration des feux tricolores
- La mise en place de potentiomètre ainsi que du buzzer.
- La rédaction d'un rapport pour documenter notre réalisation.

V. Conception

A. Architecture fonctionnelle

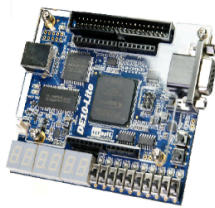


Voici l'ensemble des fonctionnalités auxquelles nos feux répondent:

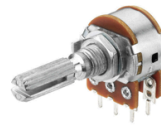
- Le feu vert des voitures reste allumé 5 secondes, le feu orange 3 secondes et le feu rouge 8 secondes.
- L'afficheur 7 segments le plus à gauche affiche l'état des feux tricolores.
- Le feu vert des piétons reste allumé 5 secondes puis rouge 11 secondes.
- Les deux afficheurs 7 segments les plus à droite permettent d'afficher le nombre de secondes avant le prochain feu vert des piétons.
- Un buzzer émet un son de plus en plus rapidement afin de signaler aux piétons malvoyant de la durée restante pour traverser.

B. Architecture matérielle

Quel matériel est utilisé et pourquoi ? Voici la liste des principaux composants:



Le DE10-lite
FPGA



Potentiomètre



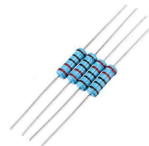
Les leds



Breadboard



Buzzer



Résistances 220R

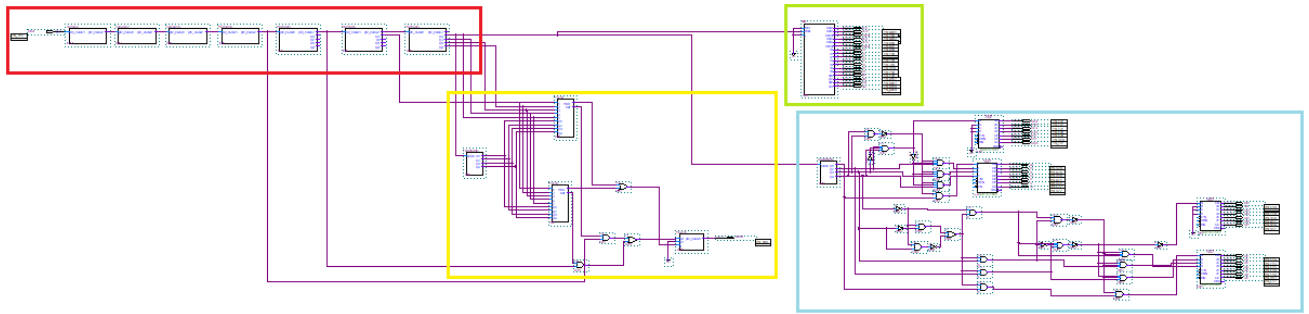
Focus sur le DE10-lite :

Le DE10-lite est un FPGA. Comme expliqué dans la partie II, un FPGA sert à implémenter un système numérique. C'est un circuit intégré conçu pour être reprogrammé après sa fabrication en utilisant un langage spécifique, donc sans modifier le matériel. Contrairement aux microprocesseurs traditionnels, un FPGA peut exécuter plusieurs opérations en parallèle.

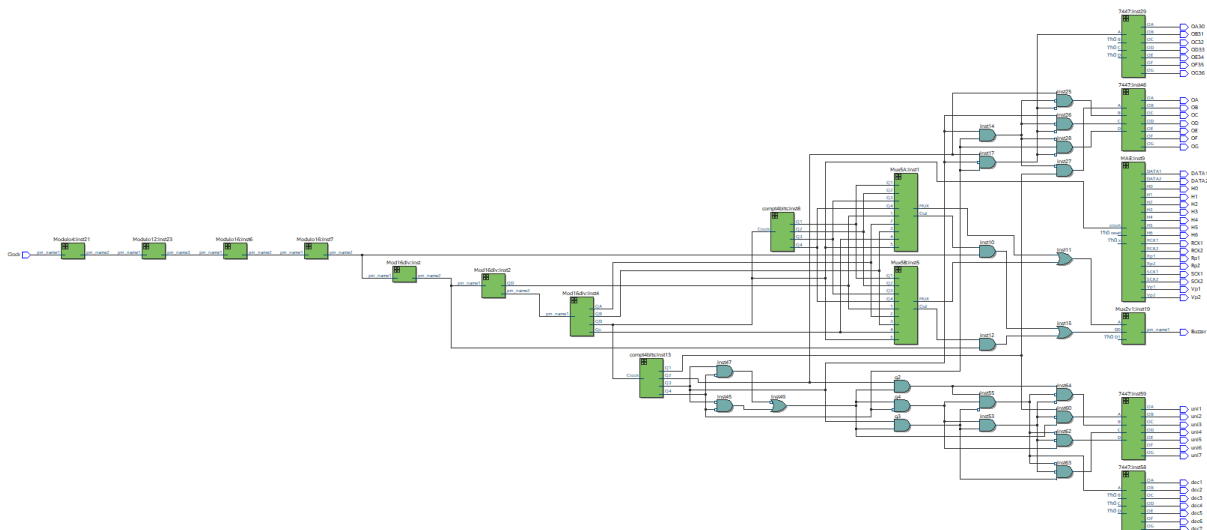
Le plus important à comprendre pour le projet sur le fonctionnement de la DE10-lite:

- La DE10-lite possède une mémoire de seulement 64 mo SDRAM avec un bus de données x16 bits.

Comment les différentes briques techniques sont connectées entre elles ?



Sur la photo ci-dessus, nous pouvons voir les différentes parties de notre montage. En rouge, la partie concernant le diviseur d'horloge. En jaune, le montage concernant l'allumage du buzzer. La partie correspondant à la couleur verte est celle de notre machine à état. Elle permet d'allumer les deux feux tricolores et notre afficheur 7-SEG des différents états. Enfin, la partie bleue correspond à l'affichage sur le 7-SEG du décompte du temps restant entre le feu rouge et le prochain feu vert de chaque feu.



Ci-dessus, une photo du RTL de notre montage complet.

C. Architecture logicielle

Comment fonctionne le programme ?

Pour faire fonctionner notre projet, nous n'avons pas eu à beaucoup coder en langage VHDL. En effet, l'allumage de nos LEDs et d'une partie de nos afficheurs 7-Segments est réalisé grâce à notre graphe d'état. A partir de notre graphe d'état, Quartus est capable de créer un code VHDL. Ce code nous servira ensuite à créer un composant que l'on ajoutera à notre schéma fonctionnelle pour réaliser les tâches souhaitées.

Durant ce projet nous avons utilisé plusieurs logiciels de conception telle que :

- Proteus pour tester la plupart de nos montages

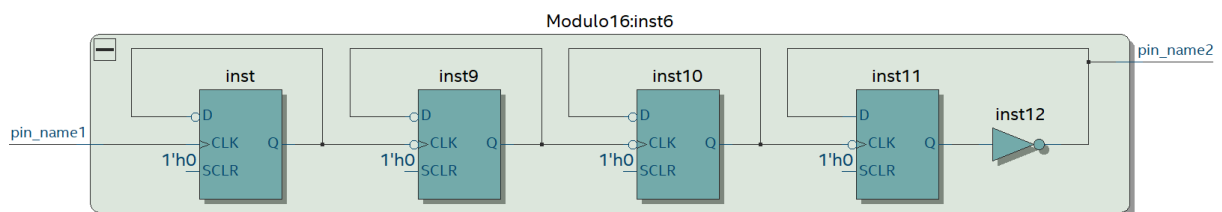
- Quartus pour communiquer avec la DE10-lite

VI. Développement

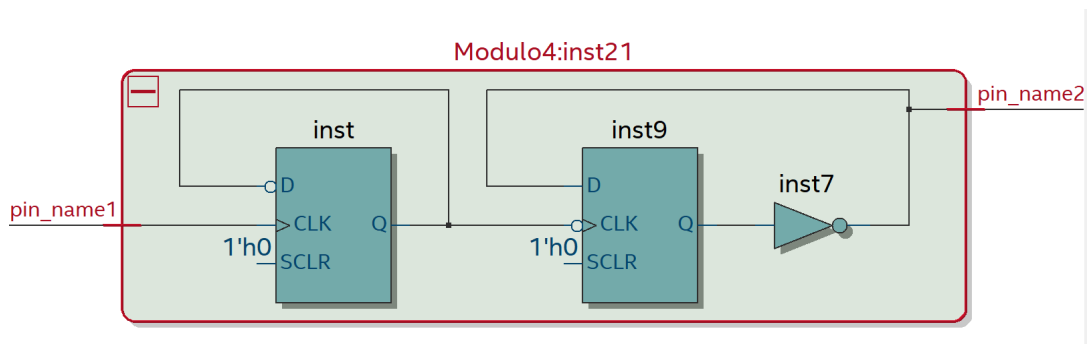
A. Module 1

La première étape de la construction de nos feux a été de transformer un signal d'horloge de fréquence 50MHz en un signal de 1 Hz avec au maximum plus ou moins 40% d'erreur. Afin de ralentir ce signal d'horloge, notre montage devait être composé de bascules D. Pour obtenir le signal de 1 Hz, nous avons eu l'idée de réaliser une succession de compteurs nous permettant d'obtenir la fréquence souhaitée.

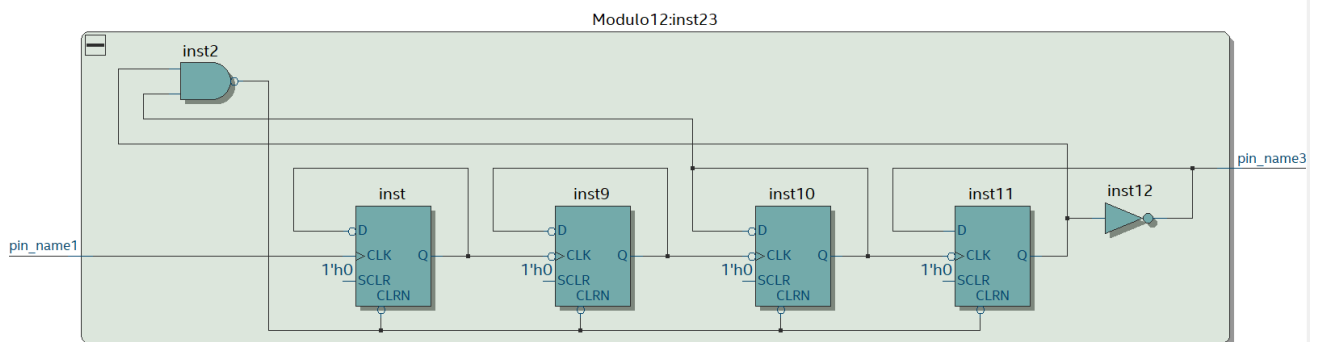
Nous avons réalisé une succession de 5 compteurs modulo 16:



Suivi d'un compteur modulo 4:



Suivi d'un compteur modulo 12:



Cette succession de compteur nous a permis d'obtenir un signal proche de 1Hz. En effet:

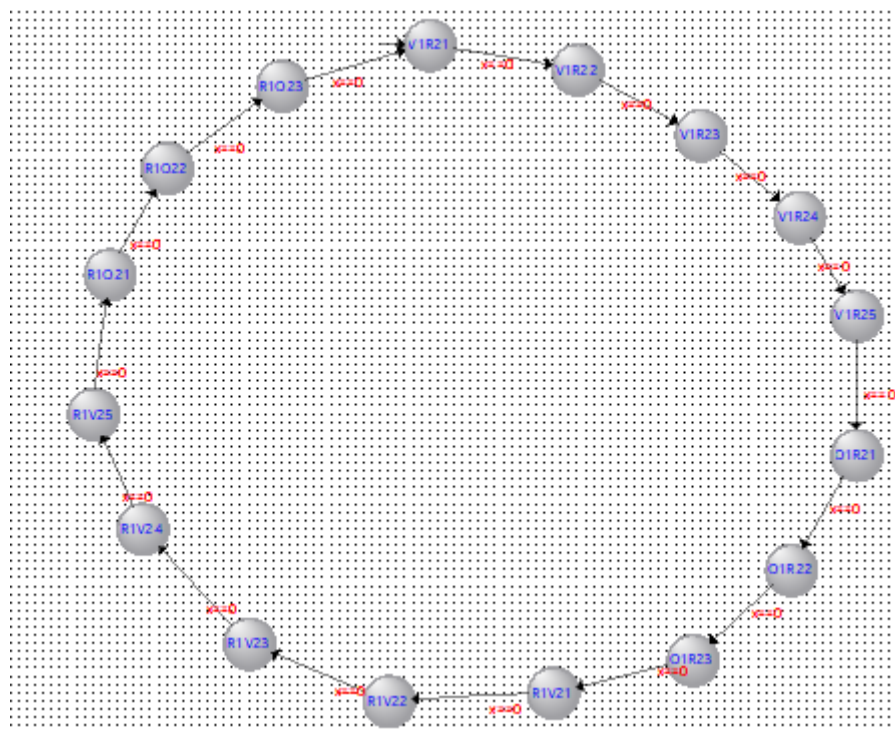
$$\frac{50 \times 10^6}{16^5} \approx 47.68 \text{ Hz}$$

$$\frac{47.68}{4} \approx 11.92 \text{ Hz}$$

$$\frac{11.92}{12} \approx 0.993 \text{ Hz soit environ 1 Hz}$$

B. Module 2

Le module 2 consiste à informer les automobilistes de s'ils peuvent traverser ou non l'intersection en toute sécurité à l'aide de deux feux tricolores. Pour cela, nous devons réaliser une machine à état générée par l'outil graphique de machine à état de Quartus. Nous avons réalisé une machine à état composée de 16 états (un pour chaque seconde du cycle routier). En réalité, les deux feux tricolores n'ont que quatre états différents : Vert-Rouge, Orange-Rouge, Rouge-Vert, Rouge-Orange. Cependant ces états n'ont pas la même durée. Le premier et le troisième état doivent durer respectivement cinq secondes, tandis que le deuxième et le quatrième état doivent durer respectivement 3 secondes, nous donnant un total de seize secondes.

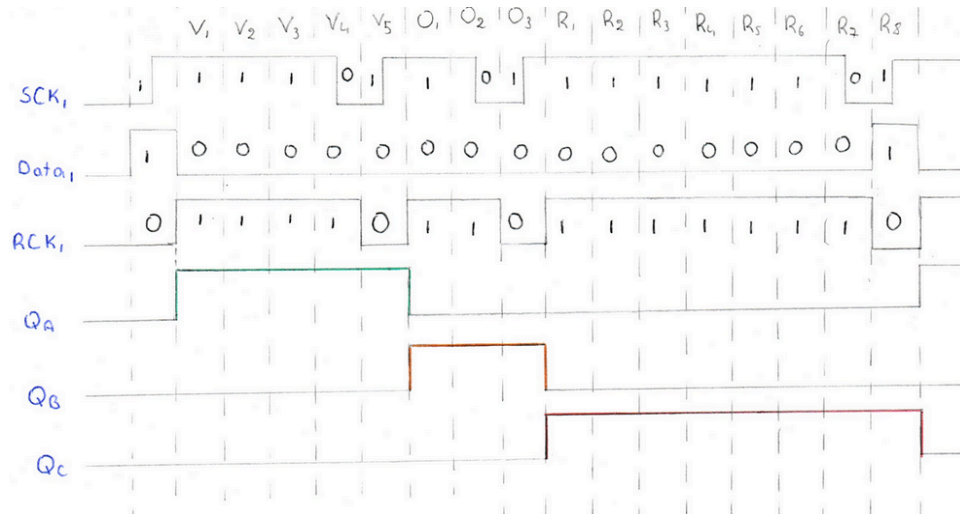


Graphe d'état obtenu

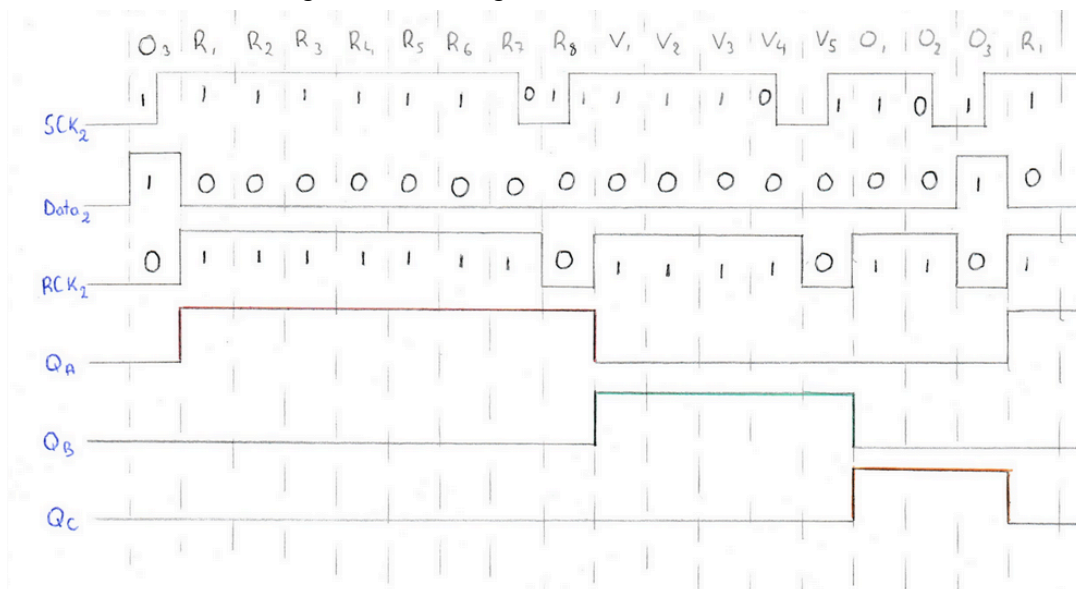
Après avoir créé nos seize états, nous nous sommes intéressés à la façon dont nous allions allumer nos LEDs. Il était demandé dans l'énoncé de piloter nos LEDs à l'aide de deux registres à décalage (un par feu tricolore).

Nous avons donc essayé de comprendre le fonctionnement des registres à décalage nous permettant de tracer les signaux suivant:

Registre à décalage du premier feu tricolore



Registre à décalage du second feu tricolore



A partir des registres à décalage que nous avons obtenus, nous avons créé les sorties SCK_1 , SCK_2 , $Data_1$, $Data_2$, RCK_1 et RCK_2 dans notre graphe d'état. Nous avons rempli leur valeur pour chaque état.

Par exemple, pour le premier état, soit sur les registres à décalage V_1 pour le premier feu et R_1 pour le second, nous avons rempli :

Output Port	Output Value	Additional Conditions
SCK1	1	
RCK1	1	
DATA1	0	
SCK2	1	
RCK2	1	
DATA2	0	

Dans ce second module, il était également question d'afficher l'état de la machine sur la première des trois paires de l'afficheur 7-SEG. Nous avons décidé d'afficher les 4 états différents. Pour cela, nous avons créé de nouvelles sorties (H0,H1,H2,H3,H4,H5,H6) correspondant chacune à un des segments de l'afficheur:

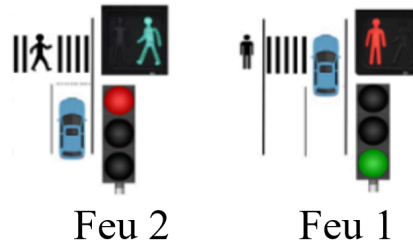
Output Port	Output Value	Additional Conditions
H0	1	
H1	0	
H2	0	
H3	1	
H4	1	
H5	1	
H6	1	

C. Module 3

Le module 3 a été le plus dur et celui qui nous a demandé le plus de temps à réaliser. Il consistait à informer les piétons de l'état du feu à l'aide de deux feux bicolores. Il était également demandé d'afficher un décompte du temps restant avant le prochain feu vert ainsi qu'ajouter un bip sonore de son différent pour chacun des feux, signalant aux personnes malvoyantes du temps restant pour traverser.

L'allumage des feux piétons fut la partie la plus simple de ce module. Nous avons modifié notre graphe d'état en ajoutant de nouvelles sorties correspondant aux différentes LEDs. Nous avons rajouté les sorties Rp1, Vp1, Rp2 et Vp2 que nous avons mises à 1 lorsque nous voulions que les LEDs soient allumées et à 0 lorsqu'elles devaient être éteintes.

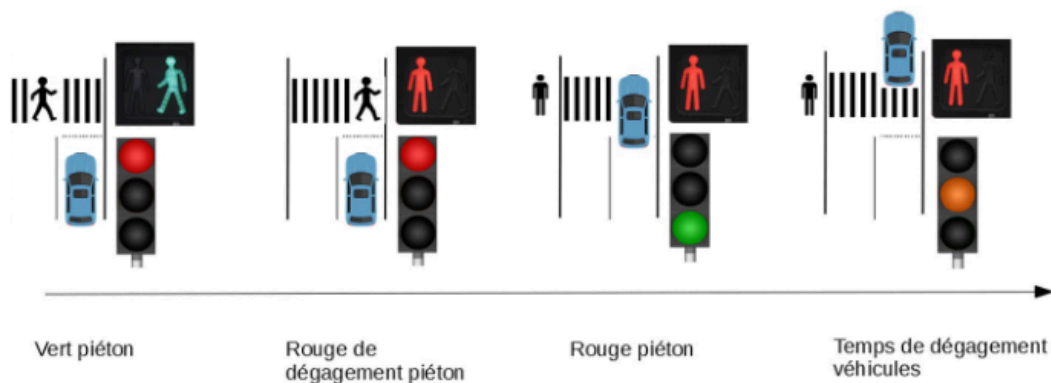
Par exemple, notre premier état correspond à celui-ci :



Les sorties des feux bicolores de notre premier état correspond ainsi à ceux de la photo ci-dessous:

Vp1	0
Rp1	1
Vp2	1
Rp2	0

Nous avons reproduit ce processus pour les 16 états de notre machine à état. Cependant, une mauvaise lecture du sujet, nous a fait perdre beaucoup de temps et recommencer du début le module 3. En effet, dans le sujet, les feux doivent suivre cet ordre ci:



En implémentant les différents feux bicolores, nous avons oublié le temps de dégagement piéton. Nous nous retrouvions alors avec un feu piéton vert durant huit secondes suivi d'un feu piéton rouge pendant les huit prochaines secondes. Le décompteur que nous avons créé décomptait le temps de 8 à 1 seconde. Pour cela, nous avons réalisé les diagrammes de Karnaugh de notre décompteur ci-dessous:

Nombres	QD	QC	QB	QA	DD	DC	DB	DA
8	1	0	0	0	0	1	1	1
7	0	1	1	1	0	1	1	0
6	0	1	1	0	0	1	0	1
5	0	1	0	1	0	1	0	0
4	0	1	0	0	0	0	1	1
3	0	0	1	1	0	0	1	0
2	0	0	1	0	0	0	0	1
1	0	0	0	1	1	0	0	0

DD :

QD QC \ QB QA	0 0	0 1	1 1	1 0
0 0	X	0	X	0
0 1	1	0	X	X
1 1	0	0	X	X
1 0	0	0	X	X

$$DD = \overline{Q_B} \cdot \overline{Q_D} \cdot \overline{Q_C}$$

DC :

QD QC \ QB QA	0 0	0 1	1 1	1 0
0 0	X	0	X	1
0 1	0	1	X	X
1 1	0	1	X	X
1 0	0	1	X	X

$$DC = \overline{Q_D} + Q_A Q_C + Q_B Q_C$$

DB :

QD QC \ QB QA	0 0	0 1	1 1	1 0
0 0	X	1	X	1
0 1	0	0	X	X
1 1	1	1	X	X
1 0	0	0	X	X

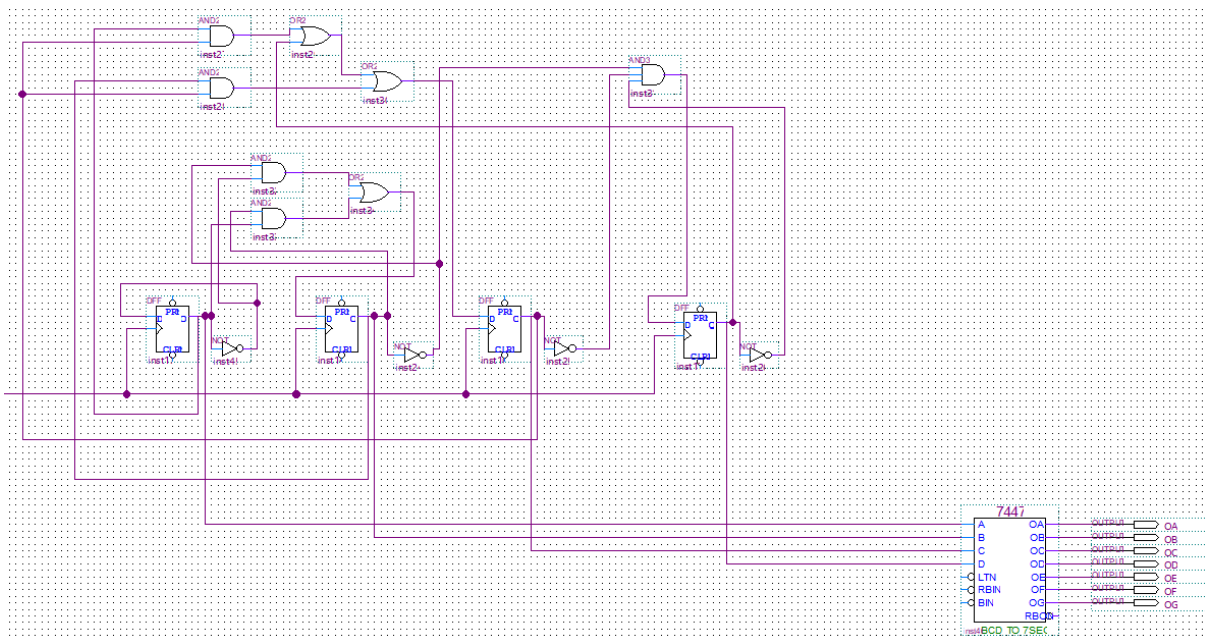
$$DB = \overline{Q_B} \cdot \overline{Q_A} + Q_B Q_A$$

DA :

QB QA \ QD QC	0 0	0 1	1 1	1 0
0 0	X	1	X	1
0 1	0	0	X	X
1 1	0	0	X	X
1 0	1	1	X	X

$$DA = \overline{Q_A}$$

Une fois ces quatre expressions obtenues, nous avons monté le décompteur ci-dessous:

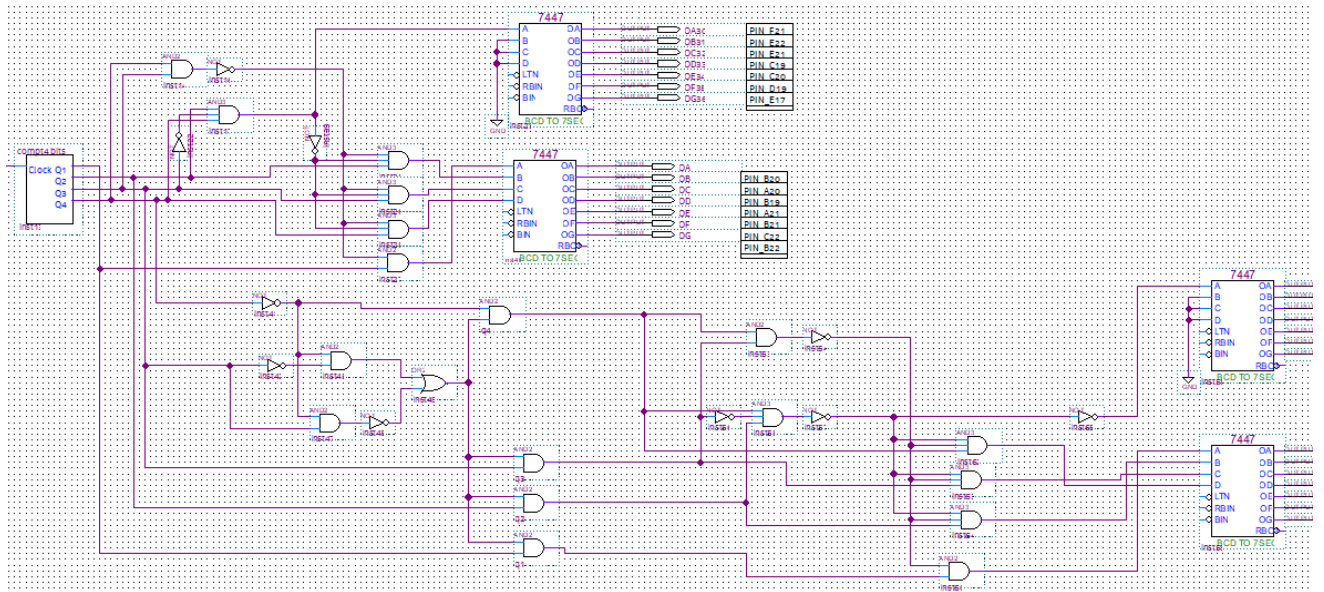


Une fois que nous nous sommes rendu compte de l'absence du dégagement piéton, nous avons dû trouver une nouvelle méthode pour décompter le temps restant avant le prochain feu piéton vert. Nous avons dû modifier la durée d'allumage des feux bicolores: cinq secondes pour le feu vert, et onze secondes pour le feu rouge.

Nous avons décidé de partir d'un décompteur modulo 16, qui grâce à des portes and et des portes or affiche 0 au lieu d'afficher les secondes 15, 14, 13 et 12 pour le premier décompteur. Le second décompteur, lui, va afficher 0 au lieu des secondes 8, 7, 6, 5 et 4. Cependant, contrairement au décompteur précédent, les secondes où le 0 n'est pas affiché ne correspondent pas à celles que nous souhaitons afficher. Par exemple : le décompte commence à la seconde numéro 15, mais le nombre de secondes qu'il nous reste avant le prochain feu vert est de 7 secondes.

En binaire, le chiffre 15 correspond à 1111 et le chiffre 7 correspond à 0111. Nous pouvons observer que la seule différence entre ces deux chiffres binaires est la bit de point fort qui est égal à 1 pour 15 et à 0 pour 7. A l'aide de portes and et or, nous avons pu transformer l'état 15 en 7 et faire de même pour les autres secondes du décompteur.

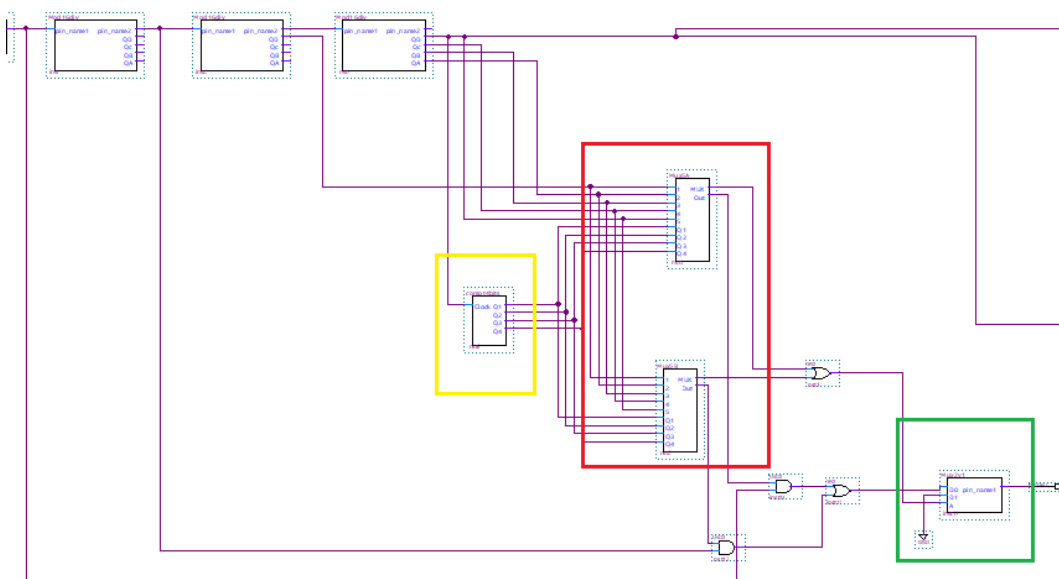
Nous obtenons le montage ci-dessous:



La dernière partie de ce module consiste à implémenter un buzzer en langage VHDL qui bip à une fréquence de son constante et dissociable selon le feu (un son différent par feu), informant les personnes mal voyantes du temps restant pour traverser. La cadence des bips doit s'accélérer à force que l'on s'approche de la fin du décomptage.

Nous avons essayé de coder le buzzer en VHDL, mais le son du buzzer restant constant, nous avons décidé de l'implémenter en BDF. Pour cela, nous avons réalisé deux multiplexeurs 5 vers 1 avec comme sélecteur un compteur sur 4 bits et comme entrées 5 fréquences différentes. A l'aide de portes and et or, nous avons relié nos deux multiplexeurs 5 vers 1 à un multiplexeurs 2 vers 1. Ce schéma permettra ainsi de créer un son d'une certaine fréquence, qui s'allumera et s'éteindra à une vitesse de plus en plus vite grâce aux 5 fréquences d'entrées.

Nous obtenons le montage ci-dessous avec en jaune notre compteur 4 bits, en rouge nos deux multiplexeurs 5 vers 1 (un pour chaque feu) et en vert notre multiplexeur 2 vers 1 permettant d'alterner le son:

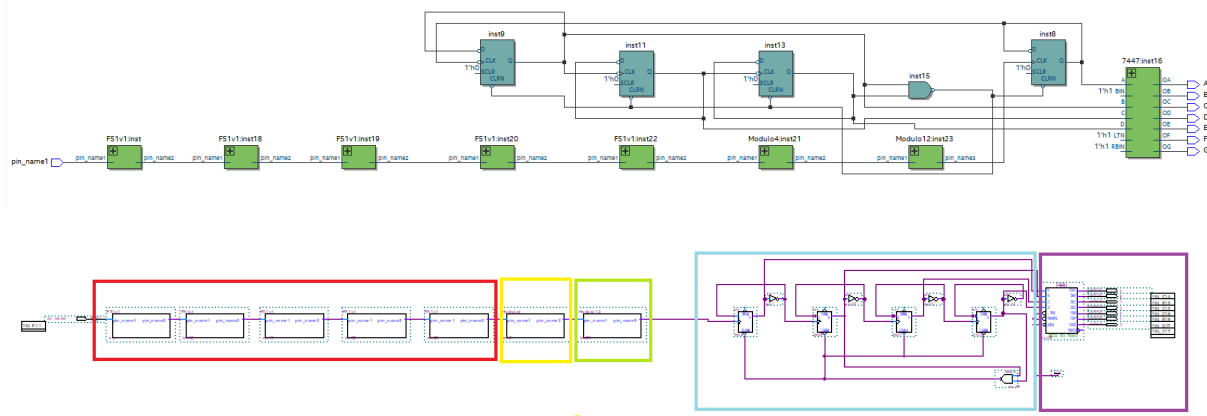


VII. Tests et validation

A. Module 1

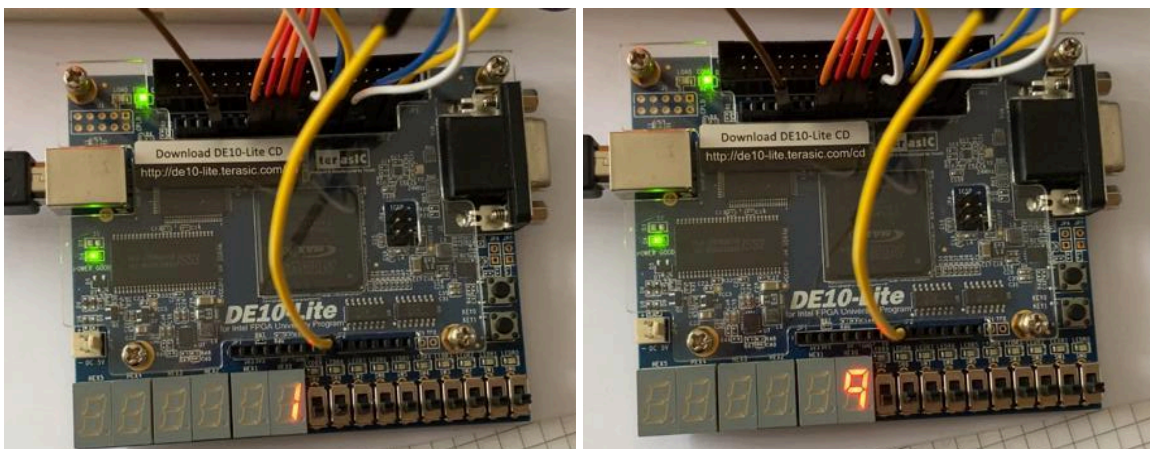
Afin de vérifier notre résultat et le bon fonctionnement de notre montage, il était demandé d'afficher la fréquence d'horloge obtenue sur un oscilloscope. Ceux-ci ne fonctionnant pas même après avoir demandé de l'aide à notre professeur, nous avons décidé d'afficher sur l'afficheur 7-Segments avec comme horloge le signal obtenu.

Pour cela, nous avons rajouté à la suite de notre montage un compteur modulo 10 qui une fois relié à un 7447, nous permettra d'afficher sur notre 7-SEG les chiffres compris entre 0 et 9.



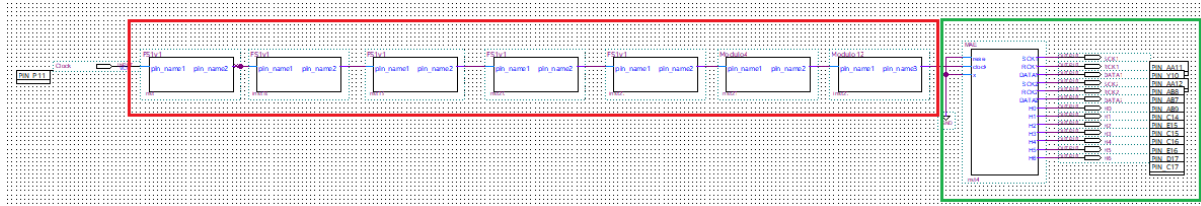
Schémas montrant la configuration de notre montage

En rouge, nous avons la succession des cinq compteur modulo 16, en jaune, le compteur modulo 4 et en vert le compteur modulo 12. La partie bleue, elle, correspond au compteur de 0 à 9. Et enfin la partie violette correspond à l'affichage du compteur sur le 7-SEG. En chronométrant le temps d'affichage de ce compteur, nous pouvons valider que notre signal d'horloge est à une fréquence de 1 Hz.



B. Module 2

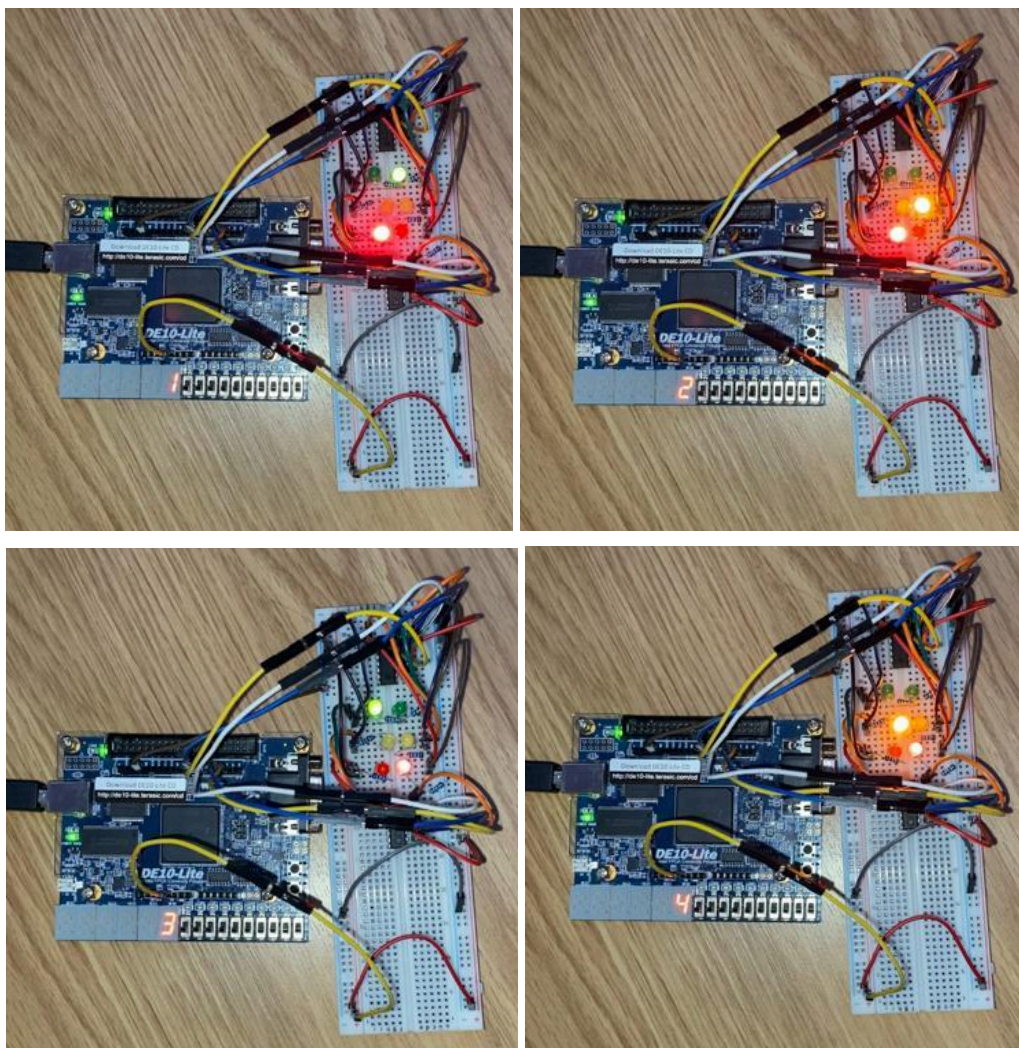
Après avoir vérifié que nous avons bien un signal proche de 1 Hz (module 1), nous avons pu tester notre second module. Pour cela, nous avons branché notre machine à état à notre signal d'horloge sur Quartus. Nous obtenons le montage ci-dessous, avec en rouge notre signal d'horloge de 1 Hz et en vert notre machine à état:



Nous avons ensuite branché nos deux feux tricolores ainsi que nos deux registres à décalage sur la breadboard.

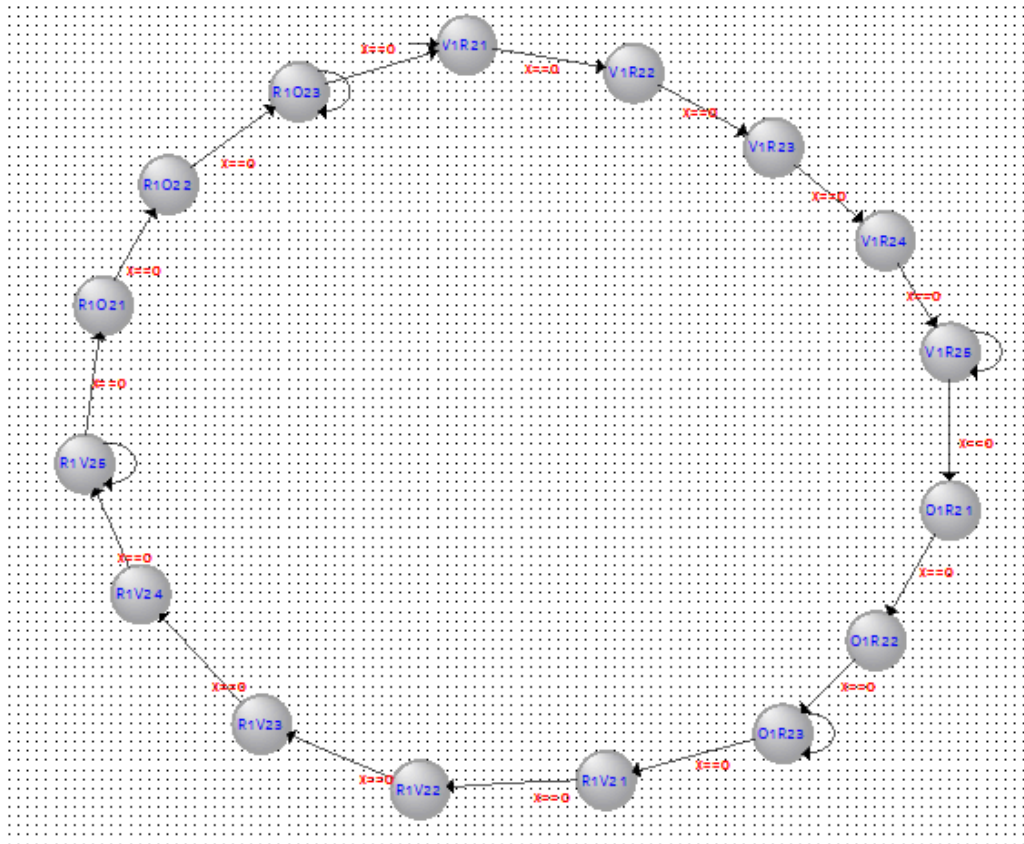
Une fois le pin planner réalisé et le programme compilé, nous avons réalisé en analysant le résultat obtenu que les afficheurs 7-SEG faisaient l'inverse de ce que nous souhaitions. En effet, les segments que nous souhaitions allumés étaient éteints et inversement. Nous avons alors remplacé les 1 par des 0 dans le graphe d'état et inversement. Une fois inversés, les afficheurs 7-SEG affichaient correctement les différents états de nos feux tricolores. Les feux tricolores étaient bien synchronisés et les LEDs souhaitées étaient allumées.

Nous avons obtenu les résultats ci-dessous :



Une partie du Module 2 consistait à ajouter un potentiomètre afin de faire varier le temps du feu vert en fonction de la circulation routière. Plus il y a d'automobilistes (valeur retournée

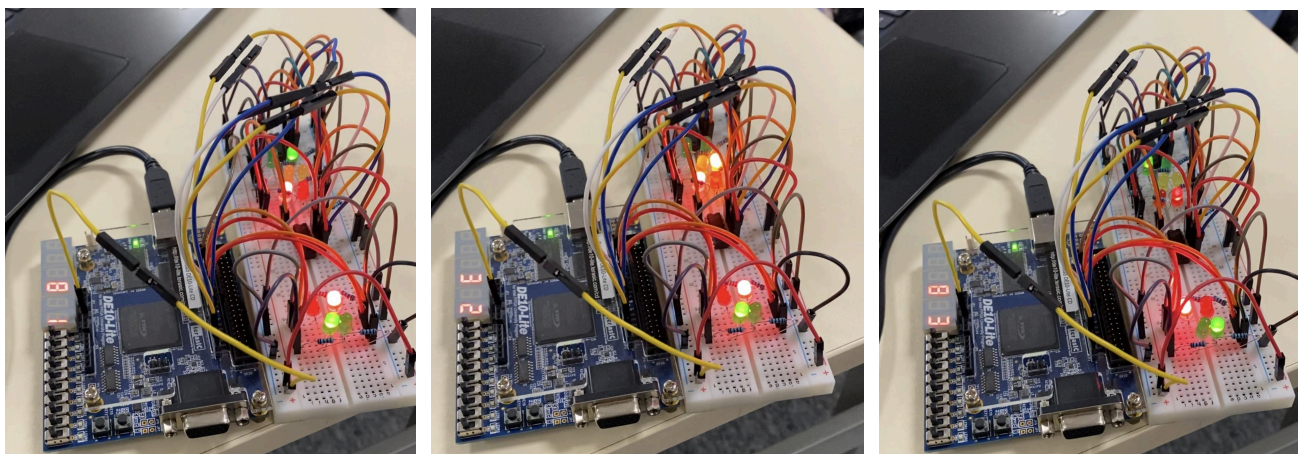
par l'ADC élevée), et plus le feu vert reste allumé longtemps. Pour cette fonctionnalité, nous avons une idée, mais celle-ci n'a pas abouti. Nous avons pour idée de rajouter une transition qui reboucle sur le dernier état du feu. Comme ci-dessous :



Avec cette configuration, nous pensions lire la valeur retournée par l'ADC. Plus elle était élevée, et plus le dernier état se répèterait. Cependant, nous n'avons pas réussi à trouver la condition de la transition qui permettrait de à l'état de se répéter.

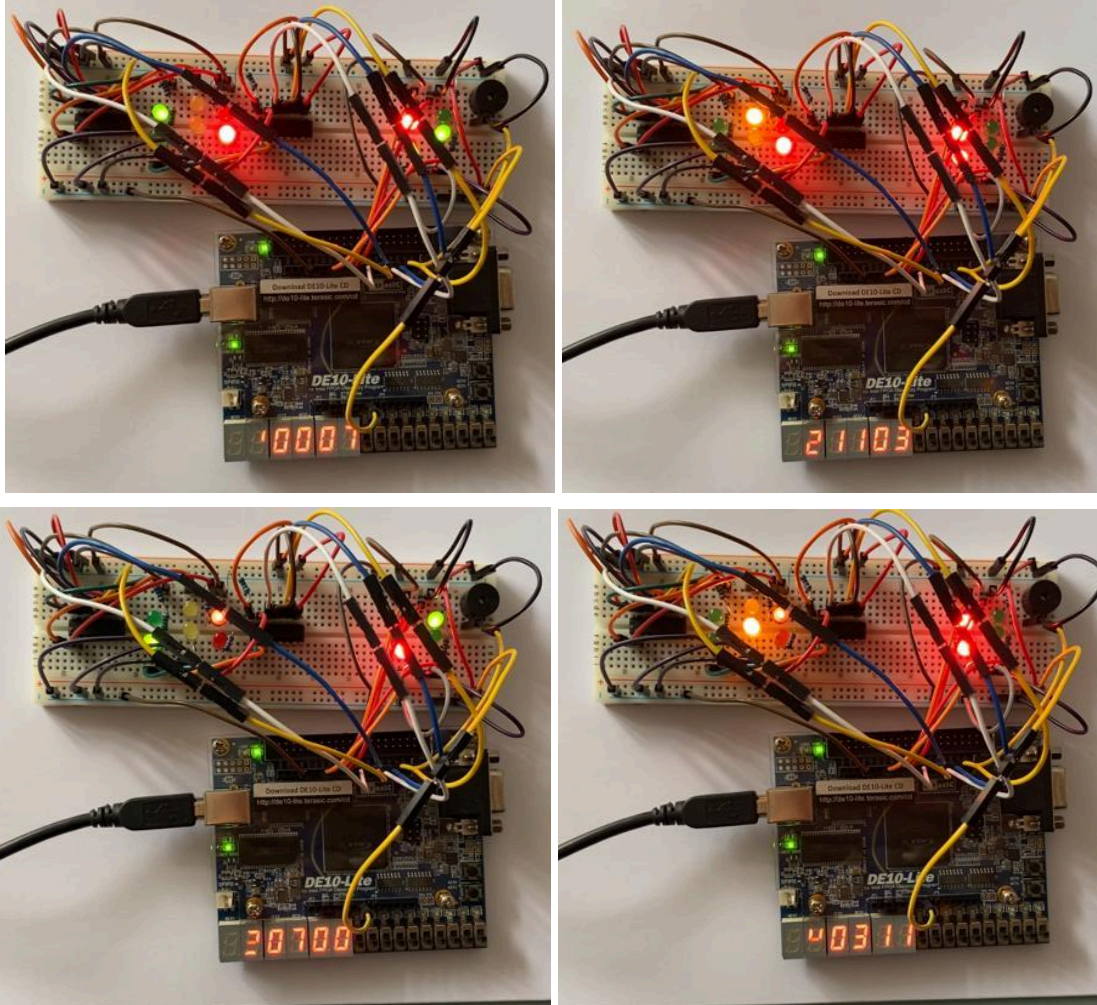
C. Module 3

Comme dit dans la partie développement, nous avons commencé par faire un décompte de 8 à 1 grâce aux diagrammes de Karnaugh suite à une mauvaise compréhension du sujet. En testant cette partie du module 3, nous avons obtenu le résultat ci-dessous:



Comme nous pouvons voir à l'état 1, le décompteur commence à 8 puis diminue et reprend à 8 à l'état 3. Nous pouvons valider le fonctionnement des feux bicolores qui changent au moment souhaité soit lorsque le décompteur reprend de 8.

Une fois que nous nous sommes rendu compte de notre erreur, nous avons pu tester notre décompteur modulo 16 que nous avons modifié. Nous obtenons les résultats suivant:



Nous pouvons observer que le compteur commence à 00 pour le premier feu piéton puisqu'il est vert (feu vert à droite sur la photo). Le décompteur du second feu piéton est à 07 puisqu'il reste sept secondes avant que le feu rouge passe au vert. Sur la seconde photo, le feu piéton qui était vert est passé au rouge et le décompteur est passé à 11 affichant le temps de durée du feu rouge. Il se produit la même chose inversement à partir de l'état 3. Comme on peut le voir sur les photos, l'afficheur 7-SEG le plus à gauche possède le segment numéro 2 qui est grillé ne nous permettant pas d'afficher l'état en entier. Nous avons essayé de le mettre sur le dernier afficheur 7-SEG libre, mais celui-ci avait les segments 2 et 3 grillés.

Pour le buzzer, nous avons pu effectuer nos tests après l'avoir ajouté à notre breadboard. Au début, nous obtenions un son qui allait vite au départ puis lentement par la suite. Nous nous sommes rendu compte de notre erreur qui était d'avoir créé un décompteur 4 bits au lieu d'un compteur. En effet, nous avons relié notre compteur à un signal d'horloge inversé, soit actif sur front montant. Nous avons désormais un buzzer fonctionnel qui sonne à une fréquence de 4 069 Hz pour le premier et à une fréquence de 254 Hz.

VIII. Bilan

A. État d'avancement

Nous avons désormais atteint la fin de ce rapport. Actuellement, le projet est abouti et nous avons réalisé ce que nous avons pu. Les objectifs sont atteints et nous pouvons affirmer que le défi de créer notre propre système de feu a été relevé malgré certains éléments fonctionnels manquants.

B. Pertinence de la solution technique

Quelles sont les limites techniques de la solution développée ?

Malgré quelques éléments manquants, nous obtenons un contrôleur de feux qui est composé de deux feux tricolores et de deux feux bicolores. Sur les afficheurs 7-SEG, les états sont affichés ainsi que les décomptes du temps restant avant le prochain feu vert pour les deux feux piétons. Nous avons également un buzzer qui bip à une fréquence de son constante et dissociable selon le feu, informant les personnes mal voyantes du temps restant pour traverser. Nous n'avons pas pu fournir de solutions fonctionnelles pour l'état de maintenance et pour régler la durée du feu vert grâce au potentiomètre.

C. Bilan sur le travail d'équipe

Nous avons travaillé ensemble de manière très concertée pour appréhender aussi bien le fonctionnement de Quartus que la partie développement et les complexités liées à la tâche demandée.

Il nous semble que cette volonté de travailler en même temps sur une grande partie du projet a pu poser un problème d'optimisation de notre temps, surtout dans un contexte de forte charge tant sur les cours que sur les révisions des examens.

Néanmoins ceci nous a permis de progresser individuellement et d'apprendre de nouvelles choses en travaillant ensemble. La forte charge de travail en fin de projet, liée notamment aux difficultés rencontrées avec le buzzer et le potentiomètre, nous a forcés à nous investir très intensément dans la réalisation et nous en tirons d'excellents enseignements, aussi bien sur la communication du groupe ainsi que sur nos capacités à nous investir périodiquement de manière très soutenue sur une tâche précise.

En conclusion, nous tirons un bilan positif de ce projet, avec la satisfaction d'avoir abouti à un bon résultat, certes incomplet, mais néanmoins fonctionnel. Cependant, nous avons conscience que notre projet mérite d'être retravaillé pour être intégralement abouti.

Ce projet nous permet aussi de tirer des enseignements en rapport avec l'avenir de notre scolarité à l'ECE Paris, ainsi qu'à notre future vie professionnelle. En effet, nous sommes arrivés à la conclusion qu'un approfondissement de la compréhension du sujet, des recherches et des expérimentations en amont du développement permettrait d'éviter

quelques surprises, ce qui s'avèrera très utile pour les prochains projets. De plus, nous pourrons mettre en avant de nombreuses compétences pour une future recherche de stage, notamment en mettant en avant notre bonne gestion de la pression ainsi que notre très bonne communication. Ces éléments s'avèrent très utiles dans le milieu professionnel.

IX. Sources

<https://tcuvelier.developpez.com/tutoriels/vhdl/introduction-langage/>

https://fr.wikipedia.org/wiki/Registre_%C3%A0_d%C3%A9calage

<https://www.lcd-compare.com/definition-de-vga.htm>

<https://openclassrooms.com/fr/courses/4117396-developpez-en-c-pour-lembarque/4630046-domptez-votre-convertisseur-analogique-numerique>

<https://www.youtube.com/watch?v=s90UAHl1xYA&t=374s>