INF1500 – logique des systèmes numériques

Laboratoire 5

Introduction au langage VHDL

Simulation et implémentation d'un circuit en logique séquentiel en VHDL

Révisions :
Alexy Torres Aurora Dugo - V1.0

Date de rendu pour le groupe 1 : 19/11/2018
Date de rendu pour le groupe 2 : 26/11/2018
Date de rendu pour le groupe 3 : 20/11/2018
Date de rendu pour le groupe 4 : 27/11/2018
Date de rendu pour le groupe 5 : 19/11/2018
Date de rendu pour le groupe 6 : 26/11/2018

Automne 2019

Département de génie informatique et de génie logiciel
École Polytechnique de Montréal

1 A LIRE IMPORTANT !

Vous n'avez qu'une semaine pour faire ce TP car la semaine d'examen approche. C'est pour cela qu'il est plus court.

Les date de rendu sont données sur la page de garde de l'énoncé. Les date d'évaluation (ou vous devez venir au lab pour vous faire évaluer) sont les suivantes :

— Groupe 1 : 26 Novembre 2019 — 8h30 - 10h00
— Groupe 2 : 26 Novembre 2019 — 10h00 - 11h30
— Groupe 3 : 27 Novembre 2019 — 12h45 - 14h15
— Groupe 4 : 27 Novembre 2019 — 14h15 - 15h45
— Groupe 5 : 26 Novembre 2019 — 12h45 - 14h15
— Groupe 6 : 26 Novembre 2019 — 14h15 - 15h45

2 Objectifs

L'objectif de ce laboratoire est de se familiariser au concept de base d'une machine à états en implémentant une solution d'un problème bien connu qui intègre le concept en utilisant le langage VHDL. Pour ce dernier laboratoire, vous allez faire une implémentation entièrement en VHDL. La validation des différentes composantes sera effectuée à travers la simulation et en dernière étape l'implémentation du circuit complet sur la carte FPGA.

3 Système à réaliser

Dans ce laboratoire, vous allez réaliser un système de feux de circulation (vert/orange/rouge) utilisé dans la gestion de la circulation d'un croisement. Le système est composé de deux feux de circulation. Un pour gérer la circulation dans la direction Ouest-Est et un pour gérer la circulation dans la direction Sud-Nord. La figure ci-dessous montre comment les deux feux sont positionnés.

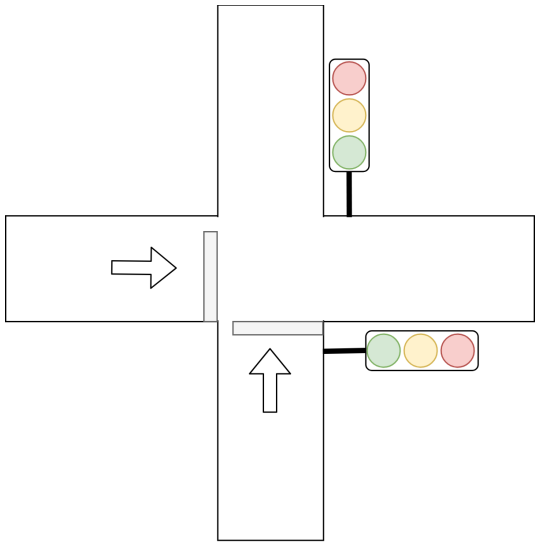

FIGURE 1 – Système de feux de croisement

Le système que vous devez concevoir comporte trois modules :

— Le système de feux, permettant la gestion des transition Vert/Orange/Rouge.
— Un générateur de pulsion, permettant de ne comptabiliser qu'une seule pulsion lors d'un appui sur un bouton.
— Un filtre anti-rebond permettant de "debounce" les appuis sur les boutons.

Le système commence à un état de base ou chaque feux est à l'état "Init". Une fois que l'utilisateur appuie sur le bouton poussoir "Start", les feux commencent leur transition, et ce, en boucle jusqu'à l'appui sur le bouton "Reset" par l'utilisateur (attention, le bouton rouge "Reset" présent sur la carte n'a rien avoir et ne doit pas être utilisé à cette fin).

Un squelette pour le module de feux vous est donné dans les ressources supplémentaires pour ce laboratoire. Le generateur de pulsion ainsi que le filtre anti-rebond vous sont fournis au complet.

3.1 Transitions des feux

Les deux feux fonctionnent en alternance une fois le bouton "Start" appuyé. Lorsqu'un feux est vert ou orange, l'autre est obligatoirement rouge. Les temps pour chaque état des feux sont donnés par le tableau suivant :

État	Vert	Orange	Rouge
Temps	4s	1s	5s

FIGURE 2 – Répartition des temps pour chaque état des feux

Pour calculer les temps d'attente en VHDL, vous devez vous baser sur la fréquence de l'horloge du circuit : 100MHz. Vous avez un exemple VHDL d'un circuit qui fait clignoter

une LED sur moodle pour vous aider.

Ainsi les transitions des feux sont les suivantes :

Feu 1	Vert	Orange	Rouge	Rouge
Feu 2	Rouge	Rouge	Vert	Orange

Vous devrez concevoir la machine à états correspondant a ce circuit et la fournir dans votre rapport.

FIGURE 3 – Séquence de transitions

3.2 Ré-initialisation du système

Le système doit être réinitialisable à tout moment grâce à un bouton poussoir “Reset”. Lors d’un Reset, l’état du système doit être remis à “Init”.

3.3 Filtre anti-rebond et pulse gen

Attention la carte FPGA ne contient pas de filtre anti-rebond, nous vous fournissons une module DEBOUNCE permettant d’effectuer cette fonction.

Jusqu’ici, nous avons assumé que lorsque nous appuyons sur un bouton, une seule pulsion sera générée qui est synchronisée avec l’horloge. Cela n’est pas le cas en réalité. Nous distinguons deux problèmes :

— Synchronisation : l’une des complications est que le signal généré par l’appui d’un bouton n’est pas forcément synchrone avec l’horloge. Cela est un problème que nous cherchons toujours à éviter pour ne pas tomber dans des ambiguïtés que nous aurons du mal à expliquer.

— Génération de pulsion unique : le deuxième problème se manifeste dans l’exemple de la figure 1. Lorsque nous appuyons sur un bouton, le temps du pressage est généralement plus long qu’une période d’horloge (puisque nous travaillons avec une horloge de large fréquence (100MHz)). Le système de feux que nous allons implémenter va voir en entrée une séquence de ‘1’ et puisqu’il est sensible à l’horloge, il va agir à chaque activation de l’horloge.

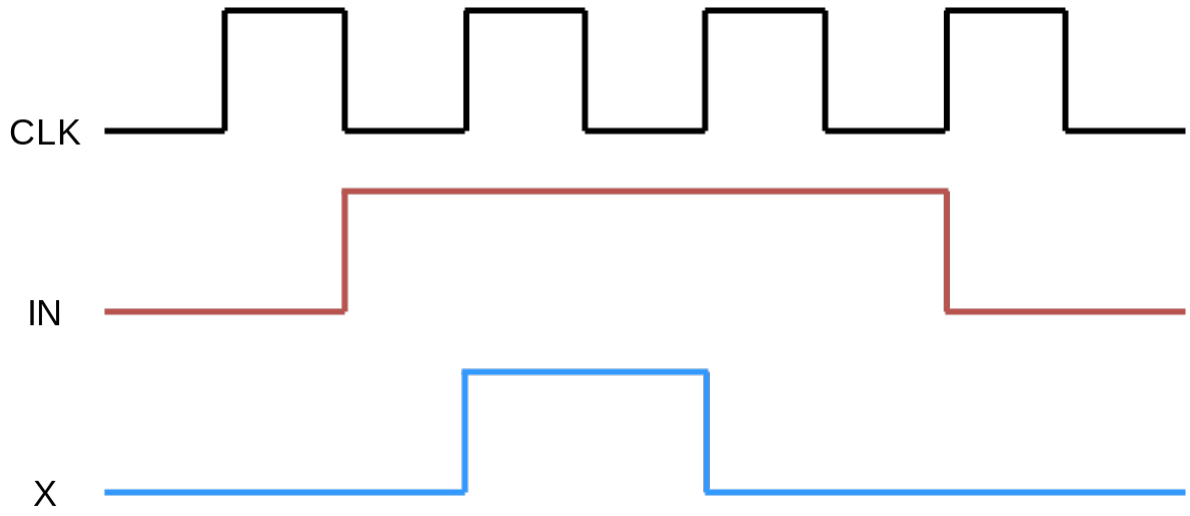

FIGURE 4 – Générateur de pulsion à partir d'un signal de bouton

Pour ces raisons, nous vous fournisson un generateur de pulsion : PUSE_GEN.

3.4 Affichage

Nous voulons être capables de suivre le comportement des feux. Pour cela, nous allons utiliser les huit 7-segments pour afficher la succession des opérations ainsi que le statut du système. Nous voulons voir l'affichage suivant :
Dans l'état initiale, un affichage de " - - - - " dois être implémenté. Une fois le système démarré, les quatre 7-segments de gauche représenterons le feu 1 et les quatre 7-segments de droite le feu 2.
Pour chaque état, voici l'affichage attendu :

État	Vert	Orange	Rouge	Init
Affichage	- - - u	- - o -	- r - -	- - - -

FIGURE 5 – Affichage par état

Le tableau suivant résume les codes déjà implémentés dans le fichier fourni *DISP_7SEG_LAB5.vhd* et qui sont nécessaires pour l'affichage des symboles correspondants :

Valeur	0	1	2	autre
Symbole	r	u	o	-

FIGURE 6 – Affichage par état

3.5 Top Level Design

Le design à implémenter dans le top level est le suivant :

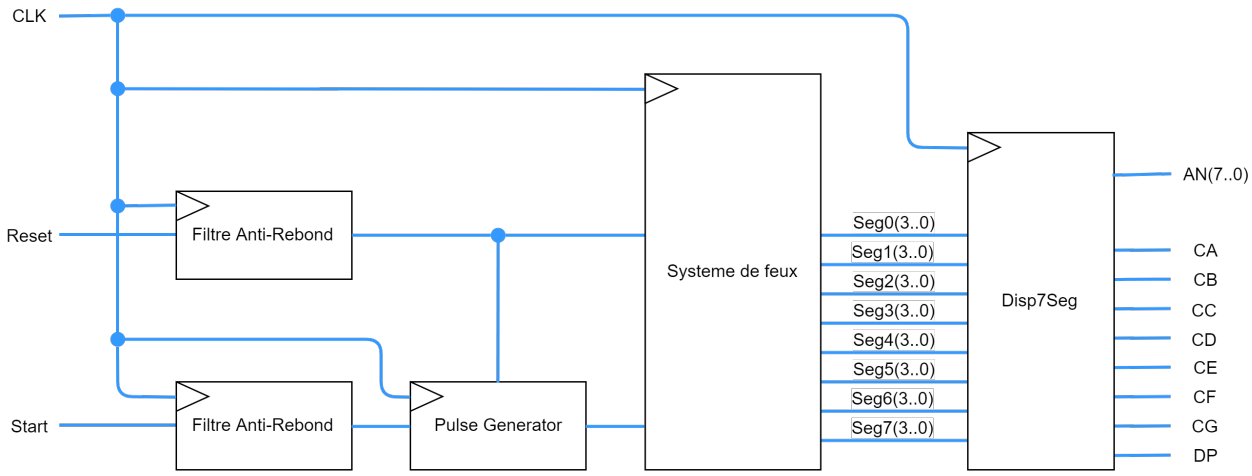

FIGURE 7 – Circuit du système complet

Après avoir confirmé que votre design fonctionne avec la simulation, vous devez implémenter votre réalisation sur la carte FPGA. Vous avez le choix pour la méthode de création du fichier de contraintes. Cependant, vous devez suivre l'assignation des pins comme décris ci-dessous :

Pin	Location
Start	P17
Reset	M17
CLK	E3
7-segment	Voir vi dessous

```
set_property -dict { PACKAGE_PIN T10   IOSTANDARD LVCMOS33 } [get_ports { CA }];
set_property -dict { PACKAGE_PIN R10   IOSTANDARD LVCMOS33 } [get_ports { CB }];
set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 } [get_ports { CC }];
set_property -dict { PACKAGE_PIN K13   IOSTANDARD LVCMOS33 } [get_ports { CD }];
set_property -dict { PACKAGE_PIN P15   IOSTANDARD LVCMOS33 } [get_ports { CE }];
set_property -dict { PACKAGE_PIN T11   IOSTANDARD LVCMOS33 } [get_ports { CF }];
set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports { CG }];

set_property -dict { PACKAGE_PIN H15   IOSTANDARD LVCMOS33 } [get_ports { DP }];

set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports { AN[0] }];
```

```
set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports { AN[1] }];
set_property -dict { PACKAGE_PIN T9    IOSTANDARD LVCMOS33 } [get_ports { AN[2] }];
set_property -dict { PACKAGE_PIN J14   IOSTANDARD LVCMOS33 } [get_ports { AN[3] }];
set_property -dict { PACKAGE_PIN P14   IOSTANDARD LVCMOS33 } [get_ports { AN[4] }];
set_property -dict { PACKAGE_PIN T14   IOSTANDARD LVCMOS33 } [get_ports { AN[5] }];
set_property -dict { PACKAGE_PIN K2    IOSTANDARD LVCMOS33 } [get_ports { AN[6] }];
set_property -dict { PACKAGE_PIN U13   IOSTANDARD LVCMOS33 } [get_ports { AN[7] }];
```

4 Travail à effectuer

Vous êtes en charge de concevoir :
— Le système de feux.
— Le top level design, permettant de mettre en place tout le système.

Pour chaque étape, il est conseillé de valider le fonctionnement de son circuit par simulation. N'oubliez pas de décrire cette étape dans votre rapport !

La dernière étape consiste à implémenter et valider le fonctionnement du circuit sur la carte FPGA.

5 Ressources supplémentaires

Pour ce laboratoire, nous mettons a disposition deux fichiers :
— Un exemple de processus VHDL qui attends une seconde, allume une LED, attends deux secondes, éteints la LED, et se répète. Ce fichier a pour nom WAIT_PROC.vhd.
— Le squelette d'une FSM a utiliser pour le système de feux de circulation. Ce fichier a pour nom TRAFFIC_LIGHT.vhd.
— Les fichiers DEBOUNCE.vhd et PULSE_GEN.vhd à utiliser.
— Le fichier driver pour l'afficheur 7-segments. Il contient le module DISP_7_SEG. Ce fichier a pour nom DISP_7_SEG_LAB5.vhd.

6 Livrables attendus

Les livrables suivants sont attendus :

- Un rapport pour le laboratoire.
- Le dossier du projet (un dossier contenant **l'intégralité de vos fichiers**).

Le tout à remettre dans une seule archive **zip** avec pour nom matricule1_matricule2_lab3.zip à téléverser sur Moodle.

Le rapport doit contenir :

- Une introduction
- Une description pour chacun des modules
- Les machines à états pour le générateur de pulsion et le système de feux
- Le code VHDL de chaque module et du système complet
- Une description de la stratégie de test pour chacun des modules et des images de vos simulations
- Une conclusion

Si vous désirez mettre du code VHDL ou autre dans votre rapport, ne faites pas de capture d'écran de l'éditeur. Il est préférable de créer un fichier séparé et d'y faire référence dans votre rapport.

Consultez le site Moodle du cours pour la date et l'heure limites de remise des fichiers.

7 Barème

La pondération sera donnée de la façon suivante sur 7 :

0/7 : l'étudiant n'a rien ou presque rien fait ;
1/7 : l'étudiant a réussi à faire approximativement 25% du laboratoire ;
2/7 : l'étudiant a réussi à faire un peu moins que la moitié du laboratoire ;
3/7 : l'étudiant a réussi à faire un peu plus que la moitié du laboratoire ;
4/7 : l'étudiant a réussi à faire presque tout le laboratoire et a démontré une compréhension comportant des faiblesses ;
5/7 : l'étudiant a réussi à faire tout le laboratoire et a démontré une compréhension comportant des faiblesses ;
6/7 : l'étudiant a réussi à faire presque tout le laboratoire et a démontré une excellente compréhension ;
7/7 : l'étudiant a réussi à faire tout le laboratoire et a démontré une excellente compréhension ;

25% des points sont retranchés par jour de retard !!!