

Laboratoire de systèmes logiques semestre automne 2023 - 2024

Laboratoire CHENILLARD


Informations générales

Le rendu pour ce laboratoire se fera **par groupe de deux**, chaque groupe devra rendre son travail.

Ce laboratoire sera évalué de la façon suivante :

- Evaluation du rendu du laboratoire
- Evaluation du quiz

Ce quiz sera fait **individuellement** et évaluera votre compréhension du laboratoire. Il sera effectué au début de la prochaine séance de cours après la réception du feedback de votre rendu.

 **N'oubliez pas de sauvegarder et d'archiver votre projet à chaque séance de laboratoire**

NOTE 1 : Afin de ne pas avoir de pénalité pensez à respecter les points suivants

- Toutes les entrées d'un composant doivent être connectées. (-0.1 sur la note par entrée non-connectée)
- Lors de l'ouverture de Logisim, bien préciser votre nom en tant que User
- Ne pas modifier (enlever/ajouter/renommer) les entrées/sorties déjà placées
- Ne pas modifier le nom des composants déjà présents

NOTE 2 : Lors de la création de votre circuit, tenez compte des points suivants afin d'éviter des erreurs pendant la programmation de la carte FPGA :

- Nom d'un circuit \neq Label d'un circuit
- Nom d'un signal (Pin) \neq Label et/ou Nom d'un circuit, toutes les entrées/sorties doivent être nommées
- Les composants doivent avoir des labels différents

NOTE 3 : Nous vous rappelons que si vous utilisez les machines de laboratoire situées au niveau A, il ne faut pas considérer les données qui sont dessus comme sauvegardées. Si les machines ont un problème, nous les remettons dans leur état d'origine et toutes les données présentes sont effacées.

Outils

Pour ce laboratoire, vous devez utiliser les outils disponibles sur les machines de laboratoire (A07 / A09) ou votre ordinateur personnel avec Logisim installé.

 **La partie programmation d'une FPGA ne peut se faire que sur les ordinateurs présents dans les salles (A07/A09).**

Fichiers

Vous devez télécharger à partir du site Cyberlearn le projet Logisim dédié à ce laboratoire.

Logisim fourni

Vous allez recevoir un projet Logisim qui contient la plupart des entités que vous allez réaliser dans le cadre de ce laboratoire. Vous devrez compléter ces entités afin de réaliser les fonctions demandées. De plus, ne modifiez surtout pas les noms des entrées/sorties déjà placées dans ces entités et n'ajoutez pas d'entrée/sortie supplémentaires.

Conseil sur l'organisation du laboratoire

Pour permettre un suivi de ce laboratoire, vous devez remplir le fichier Excel mis à disposition sous Teams.

Ce laboratoire se déroule sur **4 séances**. vous pouvez suivre l'organisation suivante pour gérer votre travail sur ce laboratoire :

séance	Étape à terminer
1	Registre à décalage 4 bits et Registre à décalage 8 bits
2	Timer
3	Contrôle du chenillard
4	Simulation, intégration & validation

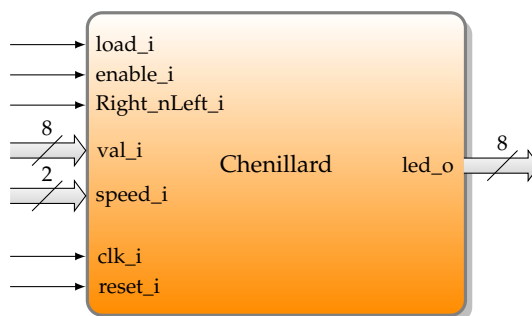
1 Objectifs du laboratoire

L'objectif principal de ce laboratoire est la réalisation d'un chenillard 8 bits.

1.1 Définition d'un chenillard

Un chenillard est un système de décalage de bits **circulaire** (ce qui sort est réinjecté au début) permettant d'afficher une animation sur des LEDs. Les leds allumées vont se déplacer de gauche à droite (Shift Right) ou de droite à gauche (Shift Left) en fonction du sens choisi par l'utilisateur. D'autre part, la vitesse du déplacement peut être choisie.

1.2 Composant



Nom I/O	Description
load_i	Activation du chargement sur la sortie led_o
enable_i	Activation du fonctionnement du chenillard
Right_nLeft_i	Sens du décalage ('1' pour un décalage à droite, '0' pour un décalage à gauche)
val_i	Données à charger si l'utilisateur désire faire un chargement
speed_i	Fréquence du décalage
clk_i	Horloge du système, fréquence de 30MHz
reset_i	Reset asynchrone du système
led_o	Affichage du chenillard sur les leds

Réalisation du laboratoire

Pour réaliser votre laboratoire, vous devrez d'abord créer un certain nombre de composants puis les utiliser afin de concevoir le chenillard. L'idée est de développer un système de A à Z afin que vous puissiez faire chaque étape vous-même et ainsi bien comprendre les concepts vus dans la théorie du cours afin de les appliquer dans un cas pratique.

Ce laboratoire est noté. Vous devez rendre le projet Logisim ainsi qu'un document pdf contenant les réponses aux questions demandées et répondre à un quiz. Le quiz sera orienté de façon à pouvoir évaluer votre compréhension du laboratoire.

2 CHENILLARD

Note : Dans ce laboratoire, vous tiendrez compte des points suivants :

- Vous réalisez un système séquentiel, ce qui implique que **chaque composant synchrone** de votre circuit comporte une entrée **clk_i**.
- Lorsque vous avez plusieurs composants, vous relierez toutes les entrées **clk_i** à une **même horloge (clk_i)**.
- Dans un circuit, on ne modifie **JAMAIS** le signal d'horloge (**clk_i**). En d'autres termes, on ne connecte jamais un signal d'horloge (**clk_i**) à une porte logique. ("Gated clock")

Travail à effectuer

Etape 1-a : Registre à décalage 4 bits

Dans le circuit **shift_reg_4b**, implémentez un registre à décalage 4 bits en utilisant des bascules Flip-Flop D ainsi que des multiplexeurs (voir cours).



Nom I/O	Description
ser_r_i	Entrée série MSB du registre à décalage
ser_l_i	Entrée série LSB du registre à décalage
value_load_i	Valeur à charger dans le registre à décalage
mode_i	Sélection du mode du registre à décalage
clk_i	Horloge du système
reset_i	Reset asynchrone du système
val_o	Valeur stockée dans le registre à décalage

Vous devez pouvoir changer le mode de fonctionnement du registre en fonction de la valeur mise sur l'entrée **mode_i** :

Opération	Fonctionnalité	Valeur sur mode_i
HOLD	Maintien de la valeur de la sortie val_o	00
LOAD	Chargement de la valeur de l'entrée value_load_i	01
SHIFT LEFT (SHL)	Décalage binaire sur la gauche	10
SHIFT RIGHT (SHR)	Décalage binaire sur la droite	11

QUESTION 1 : Que se passerait-il si l'entrée **clk_i** d'un des composants n'était pas reliée au signal d'horloge ?

QUESTION 2 : Si l'on charge la valeur **0b0110** dans le registre et que l'on effectue un décalage de 2 bits vers la droite, puis un décalage de 3 bits vers la gauche, quelle valeur va-t-on obtenir en sortie ? Démontrez les étapes pour obtenir votre résultat.

Etape 1-b : Registre à décalage 8 bits

Dans le circuit **shift_reg_8b**, implémentez un registre à décalage 8 bits en utilisant le circuit **shift_reg_4b** créé précédemment.



Nom I/O	Description
ser_r_i	Entrée sériele MSB du registre à décalage
ser_l_i	Entrée sériele LSB du registre à décalage
value_load_i	Valeur à charger dans le registre à décalage
mode_i	Sélection du mode du registre à décalage
clk_i	Horloge du système
reset_i	Reset asynchrone du système
val_o	Valeur stockée dans le registre à décalage

Validez le fonctionnement du circuit en extrayant le chronogramme de son comportement via le mode simulation.

Les étapes suivantes doivent être faites dans l'ordre :

1. Chargez la valeur **0x01** via l'entrée **value_load_i** dans votre circuit.
2. Faites décaler d'un tour complet à gauche (SHL) le LSB de la valeur chargée.
3. Chargez la valeur **0x80** via l'entrée **value_load_i** dans votre circuit.
4. Faites décaler d'un tour complet à droite (SHR) le MSB de la valeur chargée.
5. Chargez la valeur **0x88** via l'entrée **value_load_i** dans votre circuit.
6. Décalez cette valeur de 2 pas vers la droite (SHR) et démontrez que le mode HOLD est fonctionnel.
7. Faites un reset asynchrone sur votre système.

QUESTION 3 : Mettez en évidence les différentes étapes ci-dessus dans votre chronogramme.

QUESTION 4 : Votre circuit a-t-il le comportement attendu ? Argumentez et développez votre réponse.

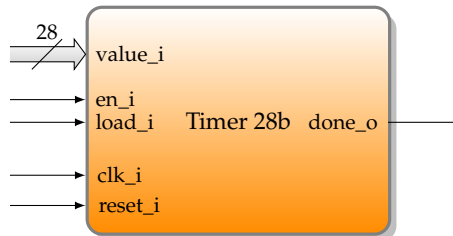
Etape 2 : Timer

Le timer va permettre de choisir la vitesse de décalage. Il est nécessaire de compter le nombre de cycle d'horloge dans le circuit correspondant à la fréquence visée avant d'appliquer un décalage sur les sorties LEDs.

Dans le circuit **timer_28b**, implémentez un compteur ou décompteur comme vu en cours.

⚠ NE PAS UTILISER le composant «compteur» de Logisim.

Ce circuit sera un timer 28 bits qui comprend :



Nom I/O	Description
value_i	Valeur du compteur
en_i	Enable synchrone du compteur
load_i	Chargement synchrone de la valeur
clk_i	Horloge du système
reset_i	Reset asynchrone du système
done_o	Flag indiquant que le compteur a fini

La sortie **done_o** sera en état "1" lorsque le nombre de cycles d'horloge est atteint. Le nombre de cycle visé en fonction de la fréquence d'horloge est chargé via l'entrée **value_i**.

QUESTION 5 : Avec une fréquence d'horloge (**clk_i**) de 30MHz, calculez le nombre de cycle à attendre pour qu'il corresponde à une fréquence de 4, 2, 1 et 0.5 Hz et complétez le tableau ci-dessous avec le nombre de cycles visé en fonction la fréquence et l'équivalence encodée au format hexadécimal sur 28 bits (0xffffffff).

Fréquence visée	Nombre de cycle	Equivalence 28bits hexadécimal
4.0 Hz		
2.0 Hz		
1.0 Hz		
0.5 Hz		

Pour la simulation, utilisez une valeur beaucoup plus petite (6 Hz, par exemple).

⚠ La partie permettant de choisir la valeur à atteindre en fonction de la fréquence ne doit pas être faite dans ce composant, mais dans la partie contrôle du chenillard.

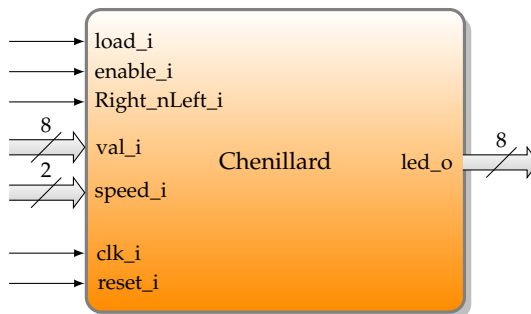
Etape 3 : Contrôle du chenillard

Complétez le composant **chenillard_TOP** avec les composants **timer_28b** et **shift_reg_8b** créés précédemment. Implémentez-y la partie permettant le contrôle du chenillard avec les composants nécessaires (portes logiques, multiplexeurs).

Les éléments suivants doivent pouvoir être réglés par l'utilisateur :

- Choix du sens de rotation
- Vitesse de rotation
- Valeur de départ

Composant Chenillard



Nom I/O	Description
load_i	Activation du chargement sur la sortie led_o (actif = état "1")
enable_i	Activation du fonctionnement du chenillard (actif = état "1")
Right_nLeft_i	Sens du décalage ('1' pour un décalage à droite, '0' pour un décalage à gauche)
val_i	Données à charger si l'utilisateur désire faire un chargement
speed_i	Fréquence du décalage
clk_i	Horloge du système, fréquence de 30MHz
reset_i	Reset asynchrone du système (actif = état "1")
led_o	Affichage du chenillard sur les leds

Comportement attendu

- Après le reset, la sortie led_o est à 0x00.
- Le reset prend effet de manière asynchrone.
- La fréquence du décalage peut être modifiée en cours de fonctionnement.
- La modification de la fréquence du décalage est asynchrone.
- Tant que l'entrée enable_i est à 0, aucun décalage n'est actif. Le circuit ne peut être qu'en mode chargement ou maintien.

Fonctionnement

1. L'utilisateur effectue un reset, la sortie led_o est à 0x00
2. L'utilisateur charge une donnée à la sortie led_o en introduisant une valeur dans val_i et en activant l'entrée load_i quelle que soit la valeur de l'entrée enable_i
3. L'utilisateur désactive le chargement puis active l'entrée enable_i. Le chenillard se lance, entraînant un décalage des leds dans le sens indiqué par l'entrée Right_nLeft_i à une fréquence indiquée par l'entrée speed_i

Table de sélection de la vitesse de défilement du pattern du chenillard en Hz

Valeur de speed_i	Fréquence de décalage du chenillard
00	4.0 Hz
01	2.0 Hz
10	1.0 Hz
11	0.5 Hz

Etape 4 : Simulation

Testez votre chenillard en simulation et relevez le chronogramme pour vérifier si le fonctionnement est conforme aux spécifications.

QUESTION 6 : Démontrez que le fonctionnement attendu est conforme en mettant en évidence les différents états sur votre chronogramme. Développez et argumentez votre réponse.

Etape 5 : Intégration/Validation

Avant d'alimenter la carte, il faut tout d'abord régler l'oscillateur qui fournira l'horloge à notre carte. Cela se fait en modifiant la configuration des curseurs de DS1. Ceux-ci doivent être réglé comme sur l'image, à savoir : ON ON OFF OFF ON. Cela permettra de régler la fréquence de l'oscillateur à 30MHz.



FIGURE 1 – configuration de l’horloge sur la carte MAX-V

Pour l'intégration sur carte, utilisez le composant **MAXV_CHENILLARD**.

Intégrez le projet **MAXV_CHENILLARD** avec pour but de programmer un circuit programmable et de tester votre Chenillard. Utilisez la console d'interrupteurs pour donner l'entrée **val_i**, les dip-switch de la carte Max V pour indiquer les autres entrées et les LEDs sur la console d'interrupteurs pour afficher les sorties du système. Lors de la programmation, dans le menu « FPGA commander », sélectionner la carte **MAX_V_CONSOLE** (Choose target board).

Faites valider le fonctionnement par l'assistant.

Rendu

Pour ce laboratoire, vous devez rendre :

- votre fichier *.circ*
- un rapport au format *pdf* contenant les réponses aux questions posées ainsi que les chronogrammes demandés.

Vous devez déposer les rendus sur Cyberlearn jusqu'à la date indiquée dans l'espace de rendu consacré à votre classe. Ainsi, vous recevrez un feedback dans le courant de semaine suivante.

CONSEIL : Faire une petite documentation sur cette partie vous préparerait directement pour le quiz et vous fera directement un résumé pour l'examen.
--