


## Laboratoire de systèmes logiques semestre automne 2024 - 2025

### Laboratoire CHENILLARD

 **Validation obligatoire pendant la dernière séance**

## 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 circuit rendu
- Evaluation des réponses aux questions

 **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
- Contrairement à ce que vous avez pu voir en cours, merci de ne pas utiliser des portes XOR sur plus d'un bit

**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 Logisim fourni

---

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

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 et en créer de nouvelles 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

---

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

## Contexte du laboratoire

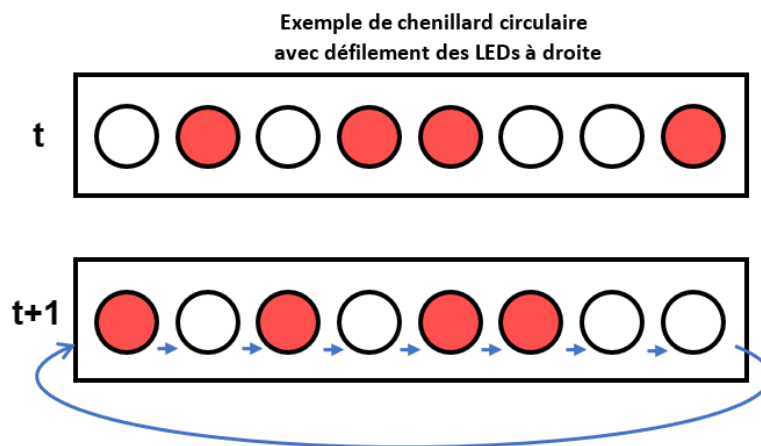
---

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

### Définition d'un chenillard

Un *chenillard* est un système qui permet d'allumer une série de voyants lumineux (LEDs) les uns après les autres. 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. Il est également possible de choisir la vitesse de déplacement des LEDs.

Un chenillard **circulaire** est un chenillard dont le défilement est en boucle : ce qui sort d'un côté est réinjecté de l'autre côté.



## 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.

**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 "**signal d'horloge**" (**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")

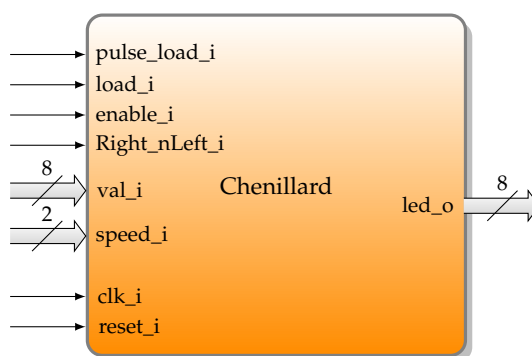
## Réalisation du chenillard

Dans le cadre de ce laboratoire, vous allez implémenter un chenillard circulaire 8 bits capable de réaliser les fonctions suivantes :

- **LOAD** : Chargement parallèle d'une valeur **val\_i** dans le circuit.
- **HOLD** : Maintenir l'état de la sortie du chenillard dans la position actuelle.
- **SHIFT LEFT** : Décaler la sortie vers la gauche.
- **SHIFT RIGHT** : Décaler la sortie vers la droite.
- **PULSE LOAD** : Injecter une valeur '1' dans le chenillard lors de son fonctionnement (par la gauche ou la droite).

### Structure du Chenillard

Ce composant se structure comme suit :

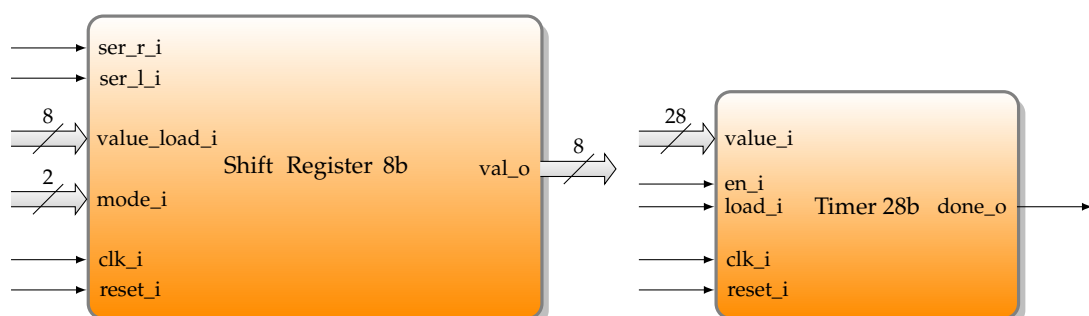


Nom I/O	Description
pulse_load_i	Injection de la valeur '1' par la gauche ou la droite (selon le sens du décalage)
load_i	Activation du chargement de val_i sur la sortie led_o
enable_i	Activation du fonctionnement du chenillard
Right_nLeft_i	Sens du décalage ('1' pour un décalage à droite, '0' pour un 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

Le chenillard est composé de deux sous-blocs , chacun réalisant une fonctionnalité particulière :

- **le Timer** : qui permet de contrôler la vitesse de décalage
- **le registre à décalage 8 bits** : qui permet de contrôler l'état des LEDs

Vous allez mettre en place ces composants afin de réaliser le Chenillard.



# 1 Travail à effectuer

## 1.1 Registre a décalage

### 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 sérielle MSB du registre à décalage
ser_l_i	Entrée sérielle 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 saisie sur l'entrée **mode\_i** :

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

**QUESTION 1** : Si l'on charge la valeur **0b0110** dans le registre, que les entrées **ser\_r\_i** et **ser\_l\_i** sont à **0** 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. Faites un chronogramme.

### 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 **réalisant le chronogramme** de son comportement via le mode simulation.

Les entrées `ser_r_i` et `ser_l_i` sont à 0.

Les étapes suivantes doivent être faites dans l'ordre :

1. Faites un reset asynchrone sur votre système.
2. Chargez la valeur `0x01` via l'entrée `value_load_i` dans votre circuit.
3. Décalez de 8 à gauche (SHL).
4. Chargez la valeur `0x80` via l'entrée `value_load_i` dans votre circuit.
5. Décalez de 8 à droite (SHR).
6. Chargez la valeur `0x88` via l'entrée `value_load_i` dans votre circuit.
7. Décalez cette valeur de 2 vers la droite (SHR) et démontrez que le mode HOLD est fonctionnel.

**QUESTION 2 :** Mettez en évidence les différentes étapes ci-dessus dans votre chronogramme. Expliquez le résultat obtenu.

## 1.2 Timer

### Etape 2 : Réalisation du Timer

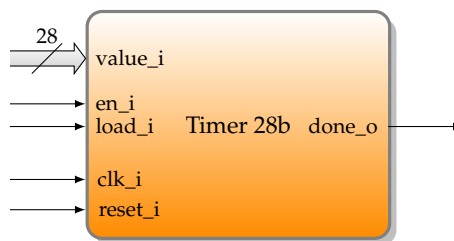
Le Timer est le composant qui permet de choisir la vitesse de décalage. Pour y parvenir, le timer tient compte de la fréquence d'horloge (ici **clk\_i**) et compte un certain nombre de cycle d'horloge (correspondant à la fréquence visée) avant d'appliquer un décalage sur les sorties LEDs.

Dans le circuit `timer_28b`, implémentez un **décompteur**.

Ici, la fréquence d'horloge est de 30MHz et le Timer doit permettre d'atteindre les fréquences de 2Hz, 1Hz, 0.5Hz et 10 MHz pour la simulation.

**⚠ 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 active pendant 1 coup d'horloge lorsque le nombre de cycle sera atteint. Le nombre de cycle visé en fonction de la fréquence d'horloge est chargé via l'entrée **value\_i**.

**QUESTION 3 :** Quel est l'intérêt d'un décompteur face à un compteur dans ce cas ?

**QUESTION 4 :** Avec une fréquence d'horloge (**clk\_i**) de 30MHz, calculez le nombre de cycle à attendre pour qu'il corresponde à une fréquence de 2Hz, 1Hz et 0.5Hz pour la mise sur carte et 10 MHz 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 (0x f f f f f f f).

Fréquence visée	Nombre de cycle	Equivalence 28bits en hexadécimal	Value_i en haxadéci-mal
10 MHz			
2.0 Hz			
1.0 Hz			
0.5 Hz			

**⚠ 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.**

## 1.3 Chenillard

### Etape 3 : Contrôle du chenillard

Après avoir développé les composants `timer_28b` et `shift_reg_8b`, vous allez maintenant compléter le chenillard dans le composant `chenillard_TOP`.

Dans ce composant, nous vous mettons déjà à disposition la gestion du `PULSE_LOAD`.

**QUESTION 5 :** Comment fonctionne la partie du `PULSE_LOAD`? décrivez le fonctionnement du `PULSE_LOAD`.

**QUESTION 6 :** Expliquez le fonctionnement de la Bascule RS, pourquoi utiliser une bascule RS dans ce cas? Justifiez votre réponse.

Complétez le composant `chenillard_TOP` avec les composants `timer_28b` et `shift_reg_8b` créés précédemment. Implémentez la partie permettant de contrôler le chenillard avec les composants nécessaires (portes logiques, multiplexeurs).

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

- Choix du mode de fonctionnement du chenillard
- Vitesse de rotation
- Valeur de départ
- Injection d'une valeur '1' pendant le défilement

#### Etape 3.1 : Choix du mode de défilement du chenillard

**QUESTION 7 :** Quelles sont les priorités de fonctionnement du chenillard? Expliquez.

**QUESTION 8 :** Faites la table de vérité de `mode_i` qui doit gérer le composant registre à décalage 8 bits. Quelles sont les entrées à utiliser?

**QUESTION 9 :** Donnez les équations simplifiées de `mode_i(1)` et de `mode_i(0)`.

Implémentez le circuit `chenillard_TOP` avec la solution trouvée.

#### Etape 3.2 : Table de sélection de la vitesse de défilement du pattern du chenillard en Hz

Valeur de <code>speed_i</code>	Fréquence de décalage du chenillard
00	10 MHz
01	2.0 Hz
10	1.0 Hz
11	0.5 Hz

Implémentez le circuit `chenillard_TOP` avec la solution trouvée.

#### Comportement attendu

- Après le reset, la sortie `led_o` est à 0x00.
- Le reset prend effet de manière asynchrone.
- La fréquence du décalage est définie par `speed_i`.
- Tant que l'entrée `enable_i` est à 0, aucun décalage n'est actif.



## 1.4 Intégration sur carte

### Etape 4 : 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 réglés comme sur l'image, à savoir : ON ON OFF OFF ON. Cela permettra de régler la fréquence de l'oscillateur à 30MHz.

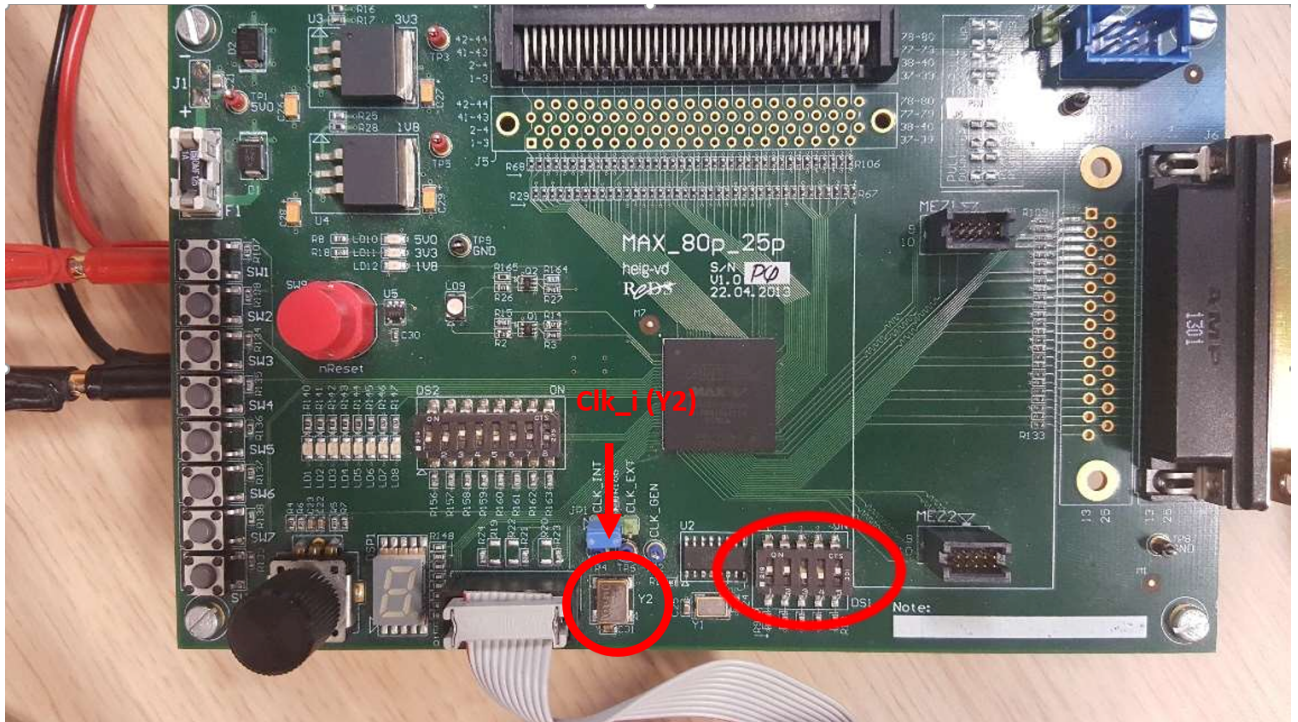


FIGURE 1 – configuration de l'horloge sur la carte MAX-V

Pour l'intégration sur carte, utilisez le composant MAXV\_CHENILLARD.

Intégrez le projet MAXV\_CHENILLARD en respectant le mapping suivant :

Nom I/O	Description
switch_i	S7 :S0
entree_i	DS2
Pulse_load_i	SW1
nReset_i	nReset
load_i	SW8
clk_i	Y2
led_o	L7 :L0

Lors de la programmation, dans le menu « FPGA commander », n'oubliez pas de sélectionner la carte MAX\_V\_CONSOLE (Choose target board).

Faites valider le fonctionnement par l'assistant.

## 1.5 Simulation

### Etape 5 : Simulation

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

#### Fonctionnement

1. Effectuer un reset, la sortie led\_o est à 0x00
2. Configurer speed\_i = "00".
3. Charger la donnée 0x12 à la sortie led\_o en introduisant une valeur dans val\_i et en activant l'entrée load\_i pendant 1 coup de clock.
4. Désactiver le chargement puis activer l'entrée enable\_i.
5. Effectuer 3 périodes de Timer avec Right\_nLeft\_i = '1'.
6. Appuyer sur Pulse\_load\_i.
7. Après 5 décalages du chenillard,
8. Désactiver l'entrée enable\_i.
9. Changer l'état de Right\_nLeft\_i pendant 1 coups de clock.
10. Réactiver le chenillard
11. Effectuer 3 périodes de Timer.
12. Appuyer sur Pulse\_load\_i.
13. Après 5 décalages du chenillard, désactiver l'entrée enable\_i.

**QUESTION 10 :** Démontrez que le fonctionnement attendu est conforme en mettant en évidence les différents états sur votre chronogramme. Développez et argumentez votre réponse.

## 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.