HEIG-VD | REDS Institut Reconfigurable and Embedded Digital Systems

Laboratoire de systèmes logiques
semestre automne 2026 - 2027

Laboratoire Guirlande de Noël

Prof. Romuald Mosqueron

Prof. Olivier Auberson

Prof. Anthony Convers

⚠ **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
- Validation lors de la dernière séance

⚠ **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
- Merci de ne pas utiliser des portes XOR sur plus de deux entrées.

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 **3 séances**. vous pouvez suivre l'organisation suivante pour gérer votre travail sur ce laboratoire :

Séance	**Étape à terminer**
1	Timer
2	Contrôle de la guirlande
3	Simulation, intégration & validation

Contexte du laboratoire

L'objectif principal de ce laboratoire est la réalisation d'un chenillard, basé sur un registre à décalage 8 bits, qui sera utilisé pour illustrer le fonctionnement d'une guirlande de Noël.

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.

Réalisation du laboratoire

Pour réaliser votre laboratoire, vous devrez d'abord mettre en place un timer qui permettra de cadencer le fonctionnement de la guirlande. Vous mettrez ensuite en place la gestion de la logique de commande de la guirlande. L'idée est de développer un système de A à Z afin que vous puissiez réaliser 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 de la guirlande

Dans le cadre de ce laboratoire pour le chenillard, vous allez implémenter un registre à décalage sur 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 vers la gauche.
— SHIFT RIGHT : Décaler vers la droite.
— BLINK : Faire clignoter toutes les LEDs (par rapport à leur valeur initiale).

Structure de la guirlande

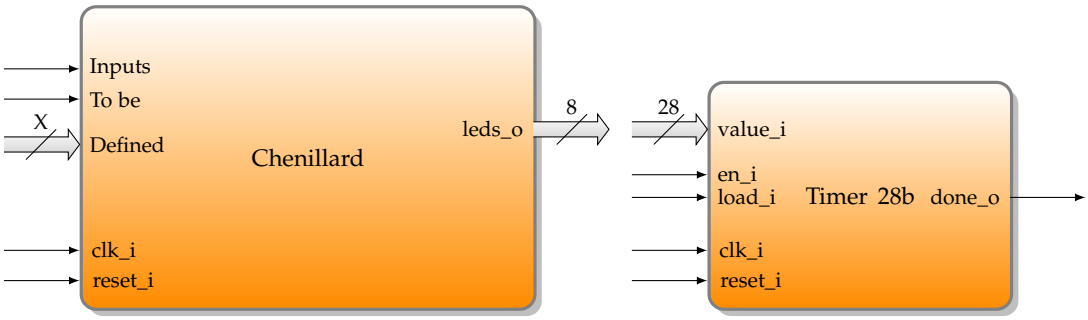
Ce composant se structure comme suit :



Nom I/O	Description
blink_i	Passage en mode clignotement
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
ledX_o	Affichage sur la led X

Le circuit `guirlande_TOP` est composé de deux sous-blocs, chacun réalisant une fonctionnalité particulière :

— **le Timer** : qui permet de contrôler la vitesse de fonctionnement de la guirlande
— **le Chenillard** qui contiendra un registre pour l'état de la guirlande et sa gestion



1 Travail à effectuer

1.1 Timer

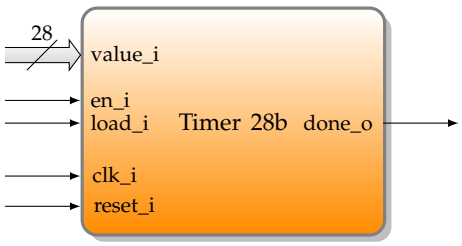
Etape 1 : Réalisation du Timer

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) puis indique à l'aide d'un flag que le compteur à atteint la valeur souhaitée. Dans notre application, ce flag est transmis au reste du système pour appliquer le décalage des LEDs à la fréquence souhaitée.

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 (Utilisée 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 à charger dans le 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 cycle d'horloge lorsque le nombre de cycle sera atteint. **value_i** est chargée de manière synchrone dans le compteur lorsque le signal **load_i** est actif.

QUESTION 1 : Quel est l'intérêt d'un décompteur face à un compteur dans ce cas ?

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.

QUESTION 2 : Complétez le tableau ci-dessous.

Fréquence visée	Nombre de cycle en décimale	Nombre de cycles sur 28bits en hexadécimal	Value_i sur 28 bits en haxadécimal
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 plus tard dans le circuit Guirlande_TOP.**

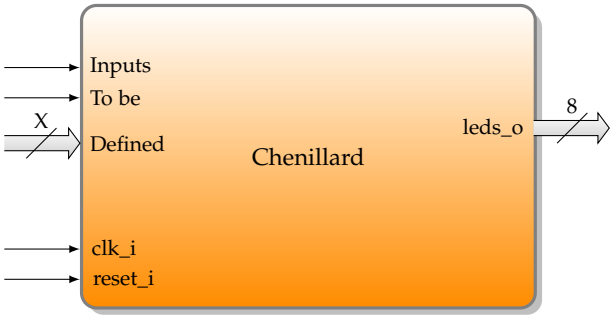
1.2 Chenillard

Etape 1 : Réalisation du chenillard

Le chenillard est le composant permettant de gérer l'évolution de l'affichage sur les LEDs.

Pour gérer la mémorisation de l'état des LEDs, vous allez mettre en place un regsitre à décalage. Vous utiliserez un `registre` de Logisim avec une taille appropriée.

Le circuit à réaliser est le suivant :



Nom I/O	Description
..._i	A vous de le définir
clk_i	Horloge du système
reset_i	Reset asynchrone du système
leds_o	Valeur sur 8 bits à afficher sur les LEDs

QUESTION 3 : Dressez la liste des entrées dont le circuit aura besoin, donnez le nom de chaque entrée, sur combien de bits, et sa description. Vous n'avez pas besoin de mentionner les entrées clk_i et reset_i qui feront forcément partie de votre circuit.

Ajoutez ensuite ces entrées au circuit Chenillard dans Logisim.

Une fois vos entrées définies, vous devrez implémenter le fonctionnement du chenillard. Pour rappel, les différentes fonctions sont les 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.
- BLINK : Faire clignoter toutes les LEDs (par rapport à leur valeur initiale).

Concernant le décalage, dans le cas d'un décalage sur la gauche, la séquence à suivre est la suivante (exemple sur 4 bits) :

0000 -> 0001 -> 0011 -> 0111 -> 1111 -> 1110 -> 1100 -> 1000 -> 0000

Le fonctionnement vers la droite est le même, mais inversé.

QUESTION 4 : En vous basant sur les différentes entrées du circuit, dressez la table des fonctions synchrones du chenillard en mentionnant l'ordre de priorité choisi pour les fonctions.

Implémentez ensuite le comportement correspondant dans le circuit Chenillard sur Losigim.

1.3 Guirlande TOP

Etape 3 : Contrôle de la guirlande

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

L'utilisateur devra pouvoir choisir la vitesse de fonctionnement de la guirlande à l'aide de l'entrée `speed_i` selon la table suivante :

Valeur de speed_i	Fréquence de fonctionnement de la Guirlande
00	10 MHz
01	2.0 Hz
10	1.0 Hz
11	0.5 Hz

Implémentez le circuit `Guirlande_TOP` avec la solution trouvée.

Comportment attendu

— Après le reset, toutes les sorties sont à 0.
— 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 Simulation

Etape 4 : Simulation

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

La liste des signaux d'entrée et de sortie à observer dans le chronogramme est la suivante :

— clk_i
— reset_i
— guitlande_TOP/timer_28b/done_o
— enable_i
— speed_i
— Right_nLeft_i
— blink_i
— load_i
— val_i
— led0_o
  .
  .
  .
— led7_o

Note : Initialement toutes les entrées sont à 0.

Note 2 : Pour faire fonctionner votre chronogramme, configurez l'horloge de la même manière que dans le tuto Logisim (semaine 1). Les 30MHz du signal d'horloge n'ont pas besoin d'être respecter. **N'oubliez pas qu'il est possible d'avancer manuellement la simulation d'une période de clock à la fois.**

Fonctionnement

1. Effectuer un reset, les sorties sont toutes à 0.
2. Configurer speed_i = "00".
3. Charger la donnée `0xF0` à la sortie led_o en introduisant une valeur dans val_i et en activant l'entrée load_i pendant 1 cycle de clock.
4. Désactiver l'entrée load_i.
5. Activer l'entrée enable_i, configurer Right_nLeft_i = '1' et effectuer 2 périodes de Timer.
6. Passer Right_nLeft_i à '0' et effectuer 2 périodes de Timer.
7. Activer l'entrée blink_i et effectuer 2 périodes de Timer.
8. Désactiver l'entrée blink_i et effectuer 2 périodes de Timer.
9. Désactiver l'entrée enable_i et attendre 3 cycles de clock. Aucun décalage ne doit avoir lieu.
10. Charger la donnée `0xA5` à la sortie led_o en introduisant une valeur dans val_i et en activant l'entrée load_i pendant 1 cycle de clock (enable_i reste désactivée).
11. Désactiver l'entrée load_i.
12. Activer l'entrée enable_i, configurer Right_nLeft_i = '1' et effectuer 2 périodes de Timer.
13. Pendant que enable_i est à 1, charger la donnée `0x1F` à la sortie led_o en introduisant une valeur dans val_i et en activant l'entrée load_i pendant 1 cycle de clock.
14. Désactiver l'entrée load_i et effectuer 2 périodes de Timer.
15. Activer l'entrée reset_i au milieu d'un cycle d'horloge (sans front montant), puis la désactiver. Les sorties doivent passer à 0 immédiatement.
16. Attendre 2 cycles de clk (clock).

QUESTION 5 : Démontrez que le fonctionnement attendu est conforme en mettant en évidence les différents états sur votre chronogramme (avec des rectangles de couleur, par exemple). Développez et argumentez votre réponse.

QUESTION 6 : Que pouvez vous constater quant au fonctionnement de la guirlande lorsqu'une valeur ne faisant pas partie de la séquence normale a été chargée par load_i (typiquement 0xA5) ?.

1.5 Intégration sur carte

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.

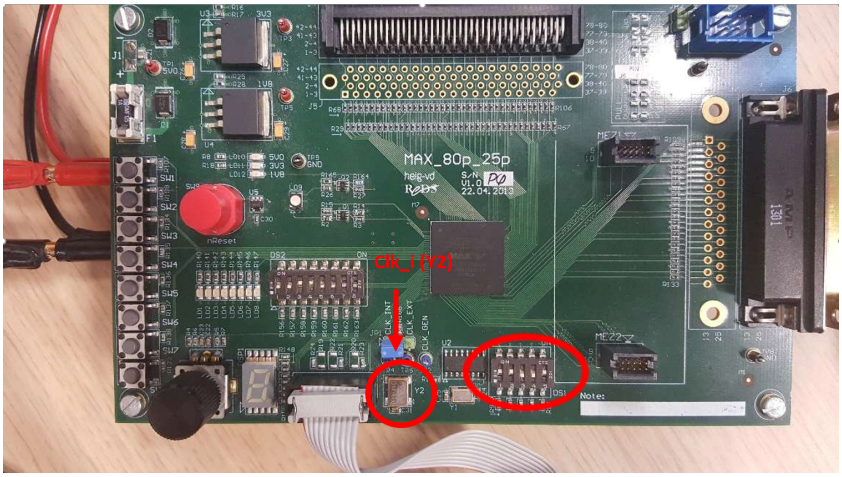


FIGURE 1 – configuration de l'horloge sur la carte MAX-V

Lors de la programmation, dans le menu « Synthesize and Download», n'oubliez pas de sélectionner la carte `MAX_V_CONSOLE` (Target board).

Intégrez le projet `MAXV_GUIRLANDE` en respectant le mapping suivant :

Nom I/O	Description
switch_i	S7 :S0
entree_i	DS2
load_i	SW8
clk_i	Y2
nReset_i	nReset
led7_o : led0_o	L7 :L0

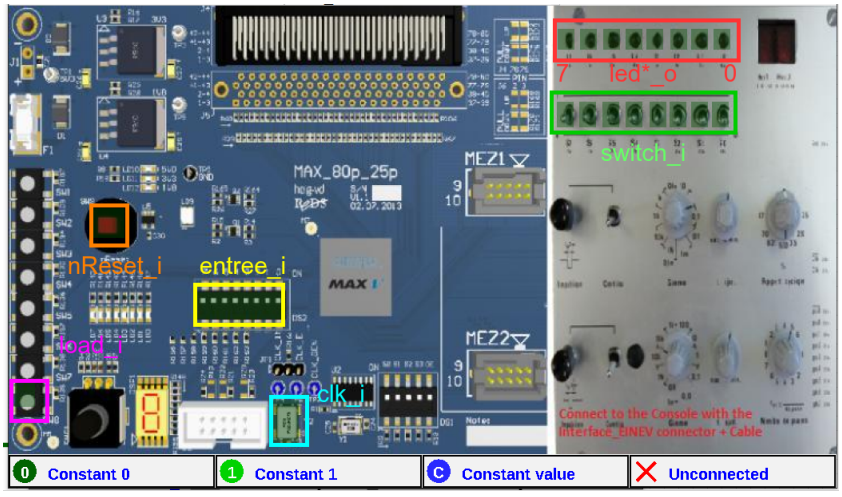


FIGURE 2 – Position des signaux sur la carte MAX-V

Faites valider le fonctionnement par l'assistant.

2 Rendu

Ce laboratoire est réalisé par groupe de deux.

Pour ce laboratoire, chaque groupe devra rendre une archive qui comporte les noms de votre binôme "rendu_NomEtudiant1_NomEtudiant2" et qui contient :

— Votre projet Logisim : **labo_girlande_etu.circ**
— Un fichier **PDF** contenant les réponses aux différentes questions théoriques.

Vous devez déposer les rendus sur Cyberlearn jusqu'à la date indiquée dans l'espace de rendu consacré à votre classe.