

## Laboratoire UART

---

### Informations générales

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

Le laboratoire sera évalué sous la forme d'un quiz à la fin de la dernière séance du laboratoire. La note sera ainsi pondérée par rapport à la qualité de votre circuit et surtout sur votre compréhension sur ce dernier qui sera évalué grâce au quiz, ciblé sur ce laboratoire.

**⚠ 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 ≠ Label d'un circuit
- Nom d'un signal (Pin) ≠ 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.

---

### Objectifs du laboratoire

L'objectif principal de ce laboratoire est la réalisation d'un émetteur UART. Pour cela, vous devrez d'abord créer un certain nombre de composants puis les utiliser afin de concevoir l'émetteur.

Ce laboratoire est noté. Vous devez rendre le projet Logisim et répondre à un quiz. Le quiz sera orienté de façon à pouvoir évaluer votre compréhension du laboratoire.

## Outils

---

Pour ce labo, 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.

# 1 UART

---

## Description

---

Un UART (Universal Asynchronous Receiver Transmitter) est un émetteur/récepteur asynchrone universel pour transmission série. L'UART est dit « asynchrone » car l'horloge de réception peut être différente de celle de transmission. Les données rentrent en parallèle dans l'émetteur, sont transmises en série au récepteur qui nous fourni la valeur lue en parallèle. Aujourd'hui, les UART sont généralement intégrés dans des composants comme des microcontrôleurs. Ils ne sont dans ce cas plus un composant à proprement parler, mais une fonction périphérique du composant.

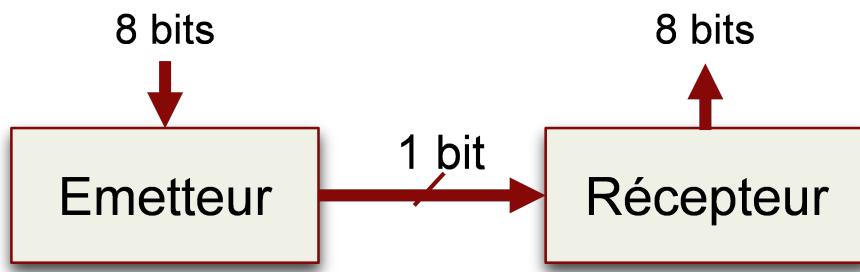


FIGURE 1 – Schéma de base d'une ligne UART

## Trame UART

---

On appelle « trame » la suite de bits envoyés en série :

- Le start bit est envoyé en premier. Il vaut toujours 0 et sert à la synchronisation du récepteur
- Les bits de la donnée à transmettre sont envoyés ensuite. Dans cet exemple on commence par le MSB
- Le bit de parité (paire ou impaire) est non-demandé dans ce laboratoire
- 1, 1.5 ou 2 bits de stop. Ils valent toujours 1
- L'état de repos de la ligne série est 1

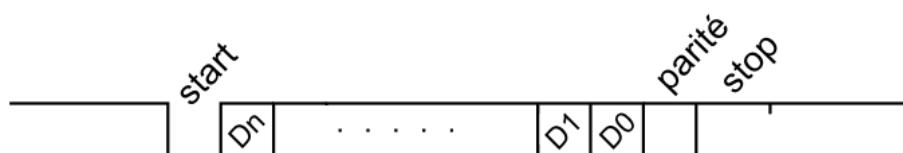


FIGURE 2 – Composition d'une trame UART

## Emetteur UART

---

L'émetteur comprend :

- Un shift register (registre à décalage)
- Un timer
- Une partie contrôle

L'appui sur le bouton « Send » charge la donnée dans le shift register et déclenche la transmission. La led « Done » s'allume quand la transmission est terminée.

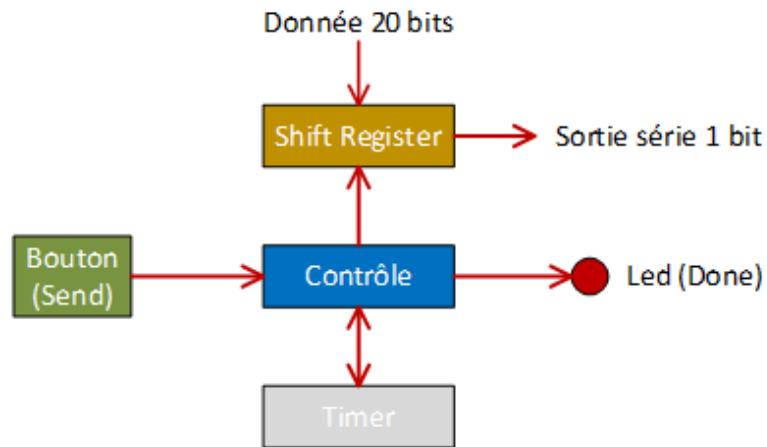
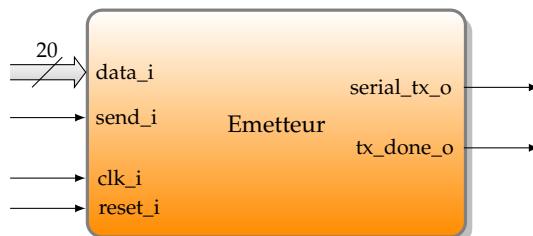


FIGURE 3 – Schéma-bloc d'un émetteur UART

### Spécifications de l'émetteur UART

1. Après le reset, la sortie série à la valeur 1, led « Done » allumée
2. Chargement de la donnée 20 bits et début de la transmission sur appui du bouton « Send », la led « Done » doit s'éteindre
3. Transmission du bit de start
4. Transmission des bits de donnée, poids fort (MSB) en premier
5. Transmission de deux bits de stop
6. A la fin de la transmission, la sortie série doit être à la valeur 1 et la led « Done » allumée
7. A l'état de repos, la ligne série est toujours à la valeur 1
8. L'émetteur doit être prêt à envoyer une nouvelle donnée

### Composant émetteur



Nom I/O	Description
data_i	Données à envoyer
send_i	Bouton permettant de démarrer un envoi
clk_i	Horloge du système, fréquence de 30MHz
reset_i	Reset asynchrone du système
serial_tx_o	Sortie sérielle de la ligne UART
tx_done_o	Sortie permettant de savoir lorsqu'une émission est terminée

**Synchronisation** Génération de la trame par l'émetteur :

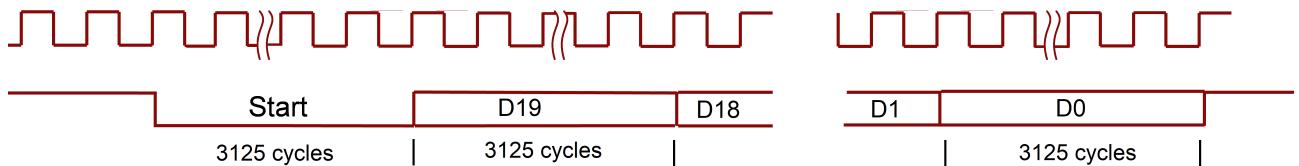


FIGURE 4 – Génération d'une trame par l'émetteur

Réception de la trame par le récepteur :

1. Le début de la trame est identifié par le front descendant du Start.
2. Le chargement de l'état du bit de donnée se fait au milieu de la plage de temps où l'état est stable.

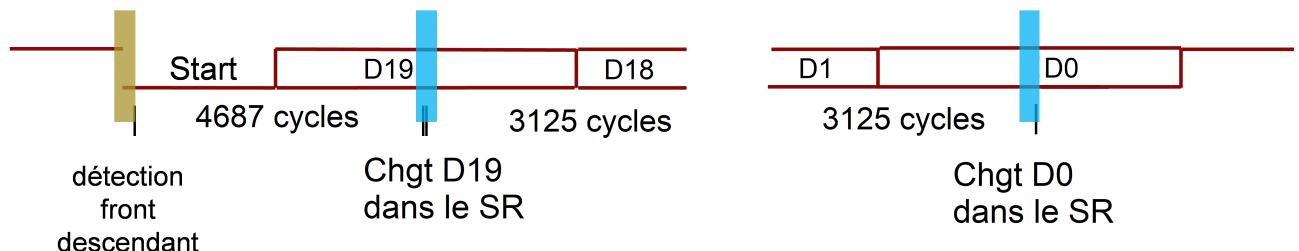


FIGURE 5 – Réception d'une trame sur le récepteur

## Travail à effectuer

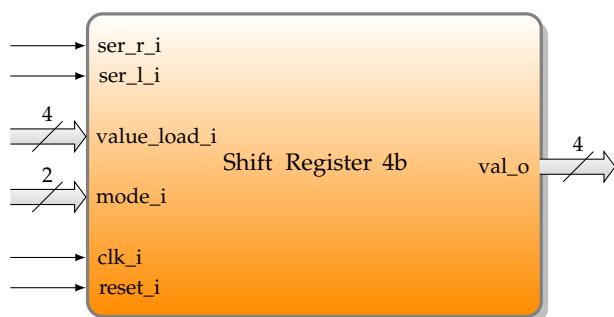
---

**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")

### Etape 1-a : Registre à décalage 4 bits

Complétez le circuit `shift_reg_4b` que vous trouvez dans le projet Logisim.



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

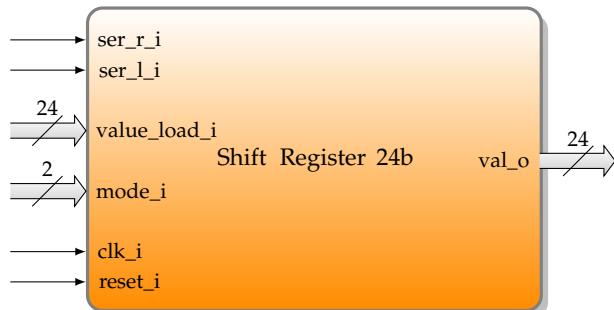
Ce registre à décalage de 4 bits doit pouvoir en fonction de la valeur sur l'entrée mode\_i :

- mode\_i = 00 : garder son contenu (HOLD)
- mode\_i = 01 : charger une donnée (LOAD)
- mode\_i = 10 : décaler à gauche (SHL)
- mode\_i = 11 : décaler à droite (SHR)

Le registre à décalage doit être réalisé à l'aide de flip-flops D et de multiplexeurs, comme vu dans le cours.

#### Etape 1-b : Registre à décalage 24 bits

Complétez le circuit shift\_reg\_24b que vous trouvez dans le projet Logisim en utilisant des registres à décalage de 4 bits (point 1-a) et en créant un ou des circuits intermédiaires si besoin.



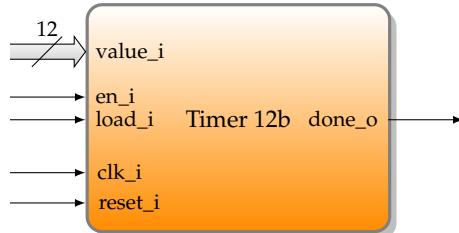
Nom I/O	Description
ser_r_i	Entrée sérielle MSB du registre à décalage
ser_l_i	Entrée sérielle LSB du registre à décalage
value_load_i	Valeur à charger dans le registre à décalage
mode_i	Sélection du mode du registre à décalage
clk_i	Horloge du système
reset_i	Reset asynchrone du système
val_o	valeur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

Utilisez le mode simulation pour valider son fonctionnement.

## **Etape 2 : Timer**

Compléter le circuit timer\_12b que vous trouvez dans le projet Logisim sans utiliser le composant «compteur» de Logisim.

Ce circuit sera un timer 12 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

Pour le fonctionnement normal du compteur, vous utiliserez la valeur 3124. Avec une fréquence d'horloge (**clk\_i**) de 30MHz, cette valeur permettra de faire fonctionner notre UART à 9600 bauds.

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

## **Etape 3 : Emetteur**

Complétez votre composant émetteur avec les composants registre à décalage 24bits et timer créés précédemment et en y ajoutant le nécessaire (portes logiques, bascules).

## **Etape 4 : Simulation**

Testez votre émetteur en simulation et relevez le chronogramme pour vérifier si la trame est conforme aux spécifications.

Vous disposez également d'un testbench en ligne (<http://reds-calculator/logisim-validator/>) qui vous permet de valider le fonctionnement de votre UART. Référez-vous au labo d'introduction pour son utilisation.

Le simulateur qui est mis à disposition n'est actuellement pas capable de vous retourner une erreur si vous ne respectez pas les points recommandés pour créer un circuit (NOTE 1 et 2). Donc si la simulation de votre circuit sur la page web ne s'arrête pas (dépasse 40s), vérifiez et respectez bien les points suivants :

- Nom du fichier labo\_uart\_etu.circ
- Nom du composant uart\_tx
- Respecter bien les informations données dans les NOTE 1 et 2 au début de la donnée.

## **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 6 – configuration de l'horloge sur la carte MAX-V

**Attention : les deux cartes et la console doivent être connectées à la même alimentation.**

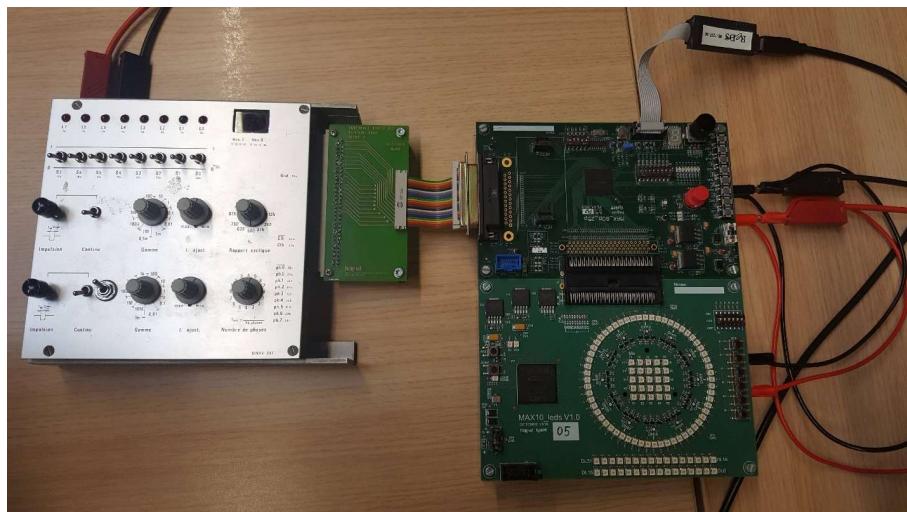


FIGURE 7 – Montage des cartes

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

Intégrez le projet « MAXV\_UART » dans le but de programmer un circuit programmable et tester votre UART. Dans le menu « FPGA commander », sélectionner la carte « MAX\_V\_CONSOLE » (Choose target board).

Pour placer les éléments, utilisez le fichier MAXV\_UART-MAX\_V\_CONSOLE-MAP.xml fourni avec le projet en cliquant sur le bouton « Load Map » :

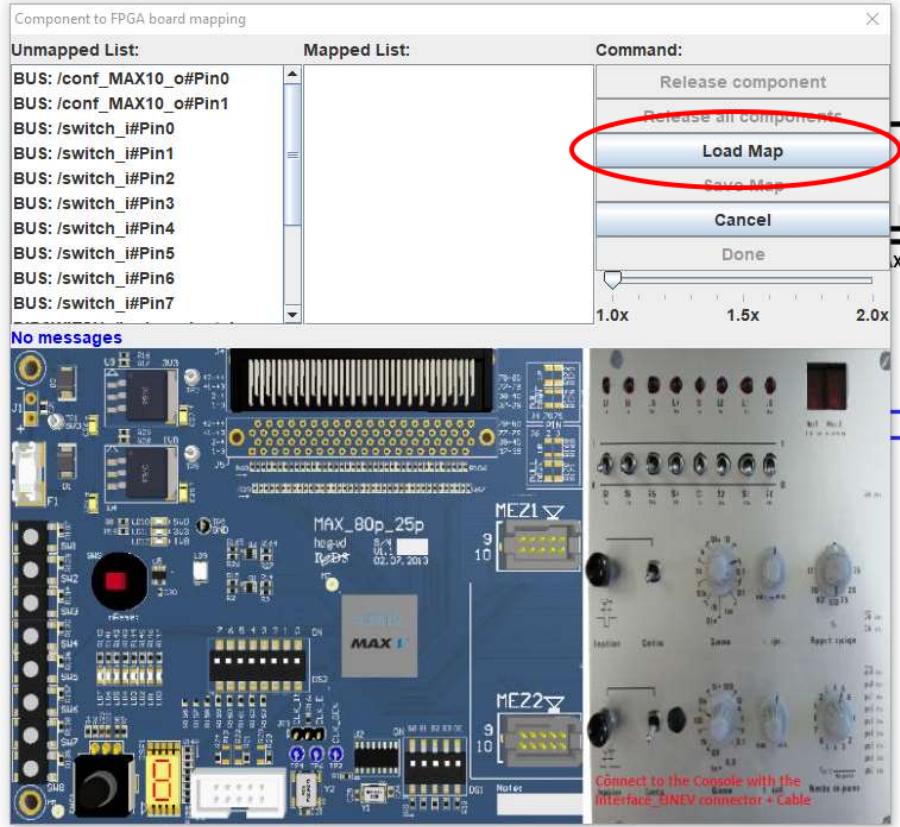


FIGURE 8 – Placement des différents éléments

Avec ce placement, les différents éléments seront disponibles :

- Switch\_i est placé sur les interrupteurs S0 à S7 de la console
- Leds\_select\_i est placé sur le dip\_switch DS2, curseurs 4 à 1
- Send\_i est placé sur le bouton SW7
- nReset\_i est placé sur le bouton SW9
- clk\_i est placé sur l'oscillateur réglable 30MHz
- tx\_done\_o est placé sur une Led et permet de savoir lorsqu'un envoi est terminé
- conf\_MAX10\_o est placé sur des Leds et permet de savoir si la MAX10 est configurée correctement
- serial\_tx\_o est la ligne série de l'UART et est connectée au connecteur noir 80p.

**Avant de commencer vos tests, appuyez sur le bouton reset des 2 cartes pour initialiser les UART (émetteur et récepteur).**

**Pour vos tests, vous devez vous aider du tableau sur la page suivante afin de déterminer quelles LEDs vous commandez par la ligne UART**

Leds_select_i [3..0]	Data [15..0]
0000	Data[15] not used, Leds secondes DS15..1
0001	Data[15] not used, Leds secondes DS30..16
0010	Data[15] not used, Leds secondes DS45..31
0011	Data[15] not used, Leds secondes DS60..46
0100	Lignes Leds DL15..0
0101	Lignes Leds DL31..16
0110	Data[15] not used, Carré Leds DM11-15 / 21-25 / 31-35
0111	Data[15..10] not used, Carré Leds DM41-45 / 51-55
1000	Leds heures DH04..01, Int[3..0], 4b intensité par LED
1001	Leds heures DH08..05, Int[3..0], 4b intensité par LED
1010	Leds heures DH12..09, Int[3..0], 4b intensité par LED
1011 - 1111	Reserved

Les Leds des heures sont des leds RGB. Chaque Led-heure dispose de 3 leds de couleur, soit : R=rouge, G=vert, B=bleu. Chaque Led-heure est pilotée avec une commande de 4bits nommée Int[3..0]. Les détails de cette commande sont disponibles dans les tableaux suivants :

Int [3..2]	Color	Couleur sélectionnée
00	Red	Couleur rouge
01	Green	Couleur verte
10	Blue	Couleur bleue
11	White	Couleur blanche, soit les 3 Leds RGB

Int [1..0]	Intensity	Intensité sélectionnée
00	Strong	Intensité forte
01	Medium	Intensité moyenne
10	Weak	Intensité faible
11	Off	Eteint

Pour valider le laboratoire, vous devez allumer la première et l'avant-dernière ligne de la matrice de LEDs centrale comme suit :



FIGURE 9 – LEDs à allumer pour la validation

Faites valider le fonctionnement par l'assistant ou le professeur.

## Rendu

---

Pour ce laboratoire, vous devez rendre :

- votre fichier *.circ*

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.**