

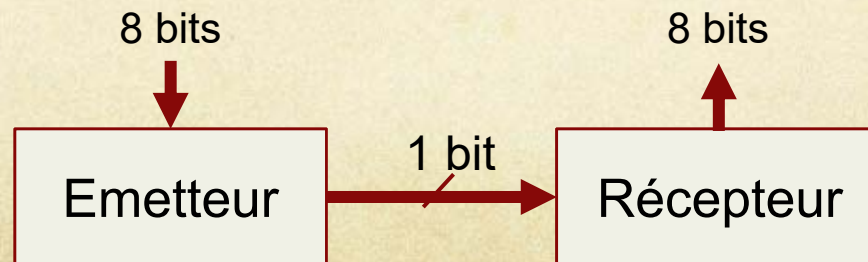
# Laboratoire UART

Universal Asynchronous Receiver Transmitter

Profs. Peña & Perez-Urbe & Mosqueron – V3

# UART

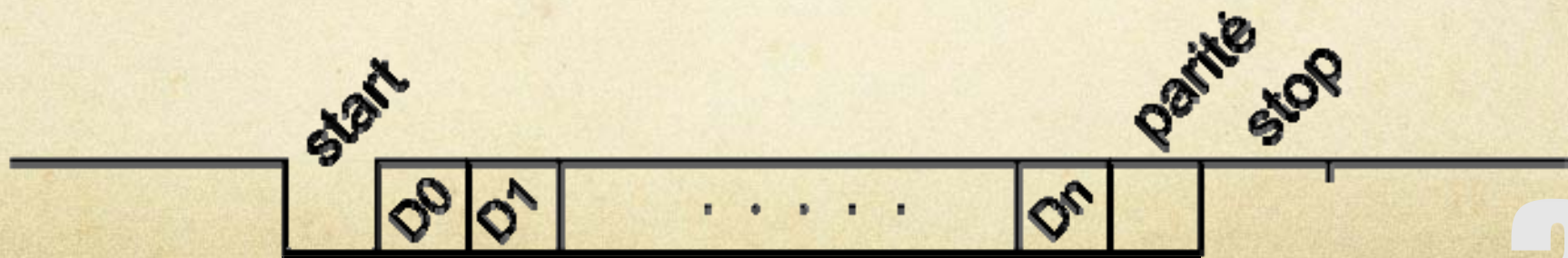
- Un UART est un émetteur /récepteur asynchrone pour transmission série  
(<https://fr.wikipedia.org/wiki/UART>)
- L'UART est dit « asynchrone » car l'horloge de réception peut être différente de l'horloge de transmission
- Les données rentrent en parallèle dans l'UART émetteur, sont transmises en série à l'UART récepteur et sont délivrées en parallèle





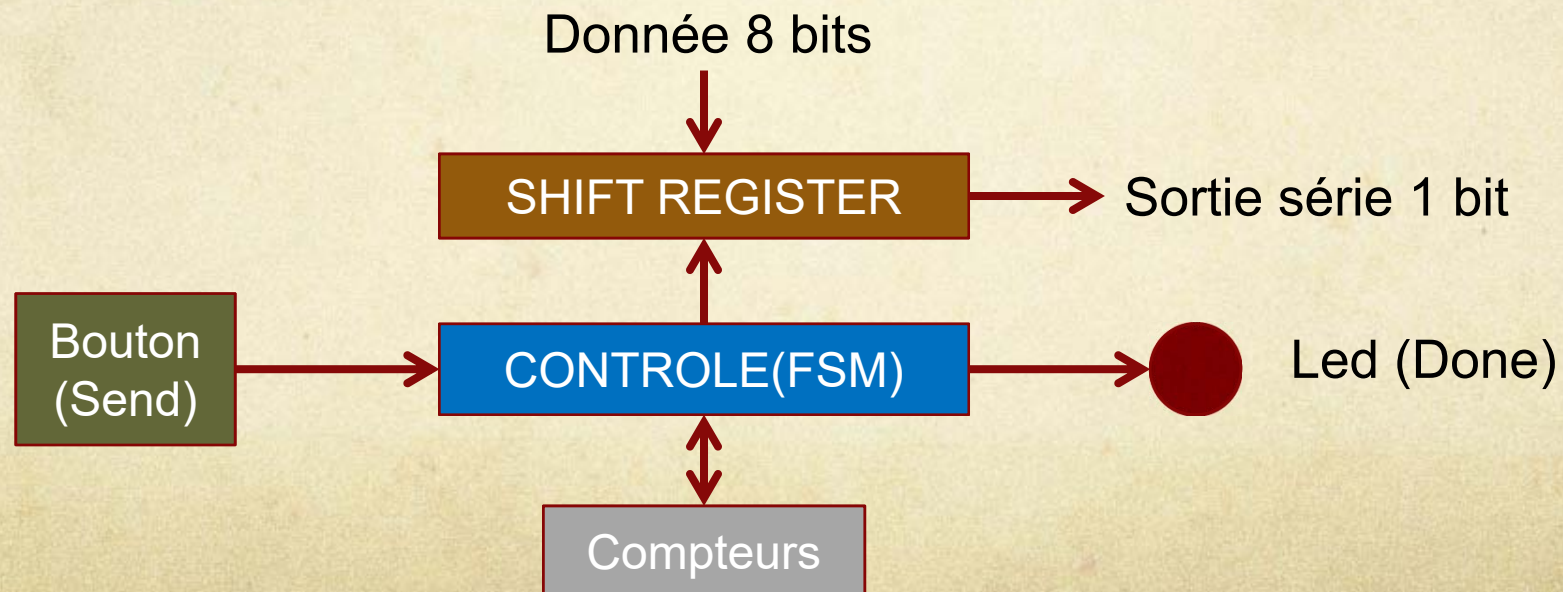
# Frame UART

- On appelle « **trame** » la suite de bits envoyés en série
- Le **start bit** est envoyé en premier. Il vaut toujours 0
- Les bits de la **donnée** à transmettre sont envoyés ensuite
- Le bit de **parité** est facultatif (non demandé dans ce labo)
- L'état de repos de la ligne série est 1



# Emetteur UART

- L'émetteur comprend un shift register, un compteur, et une partie contrôle (machine à états).
- 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



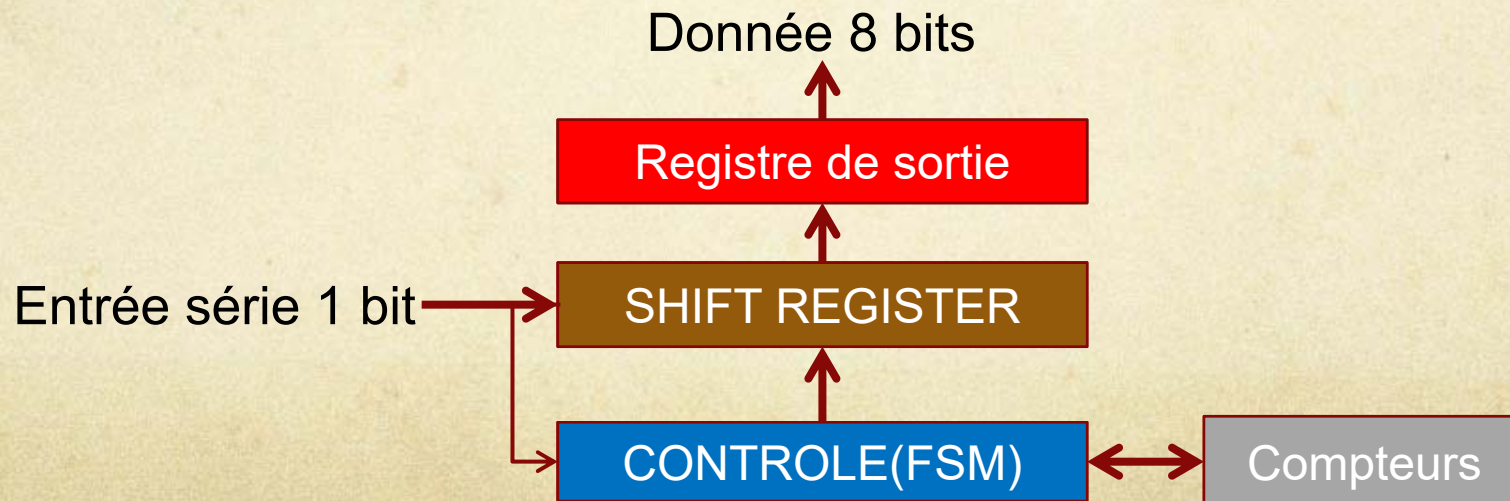


# Spécification de l'émetteur UART

- Après le reset, sortie série à l'état 1, led « Done » éteinte
- Chargement de la donnée 8 bits et début de la transmission sur appui du bouton « Send »
- Transmission des bits de poids faible en premier
- Durée du Start bit (état 0) : 2 cycles d'horloge
- Durée d'un bit de donnée : 4 cycles d'horloge
- A la fin de la transmission, la sortie série doit être à l'état 1 et la led « Done » allumée

# Récepteur UART

- Le récepteur comprend un shift register, un compteur, un registre de sortie et une partie contrôle (machine à états).
- Le début de la réception de la trame série est détectée par le contrôle. Les bits de la donnée reçus en série par le shift register sont chargés dans le registre de sortie dès que la transmission est terminée.



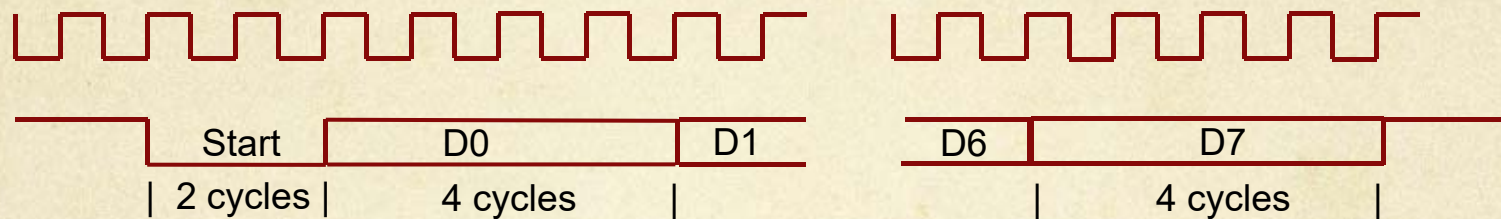


# Spécification du récepteur UART

- Après le reset, le registre de sortie contient la valeur 0
- Attente de la détection d'un front descendant (passage de 1 à 0) sur l'entrée série provoqué par l'envoi du start bit par l'émetteur
- Chargement du premier bit de donnée : 4 cycles d'horloge après le front descendant du Start bit et chargement des bits de donnée suivants tous les 4 cycles d'horloge
- A la fin de la transmission, la donnée reçue est chargée dans le registre de sortie

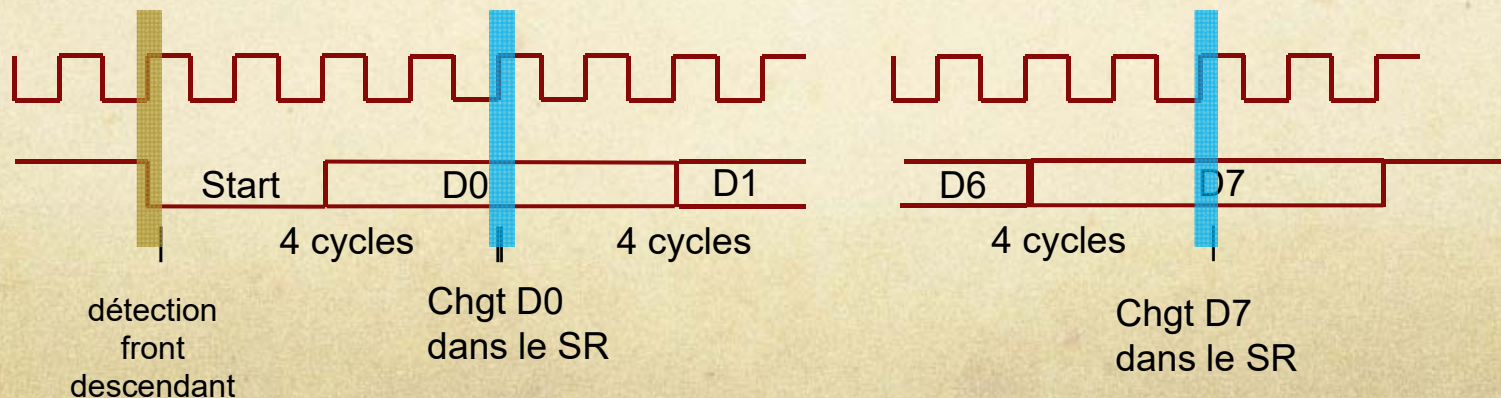
# Synchronisation

## ○ Génération de la trame par l'émetteur



## ○ Réception de la trame par le récepteur:

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





# Déroulement du labo (1)

- La moitié des groupes de 2 étudiants réalise un émetteur UART, l'autre moitié un récepteur UART.
- Vous devez utiliser votre shift register du labo précédent.
- Vous devez créer vos propres compteurs. N'utilisez pas les compteurs de la librairie Logisim.
- Vous devez utiliser un **codage 1 parmi M** pour réaliser la machine d'état.

# Déroulement du labo (2)

- Pour tester le récepteur en simulation sous Logisim, un générateur de test envoyant la donnée 0xA3 vous sera fourni. Vous devrez relever un chronogramme.
- Pour tester l'émetteur, vous devez relever le chronogramme et vérifier si la trame est conforme aux spécifications.
- Vous devez vous associer avec un autre groupe pour tester en simulation un émetteur connecté avec un récepteur
- Vous devez ensuite vous associer avec un autre groupe pour tester une transmission entre deux cartes



# Réalisation de l'émetteur

1. Utilisez le **shift register 8 bits** réalisé dans le labo précédent. Soyez certain qu'il fonctionne correctement
2. Créez un **compteur** comptant de 0 à 3 sans utiliser les composants «compteur» de Logisim. Ce compteur comprend une entrée remise à 0 synchrone et une sortie indiquant que le compteur a atteint la valeur max.
3. Créez un autre **compteur** comptant de 0 à 7 sans utiliser les composants «compteur» de Logisim. Ce compteur comprend une entrée remise à 0 synchrone, une entrée enable et une sortie indiquant que le compteur a atteint la valeur max
4. Créez une **machine d'état de type 1 parmi M** qui contrôle les deux compteurs et le shift register. Le nombre d'états doit être minimum.

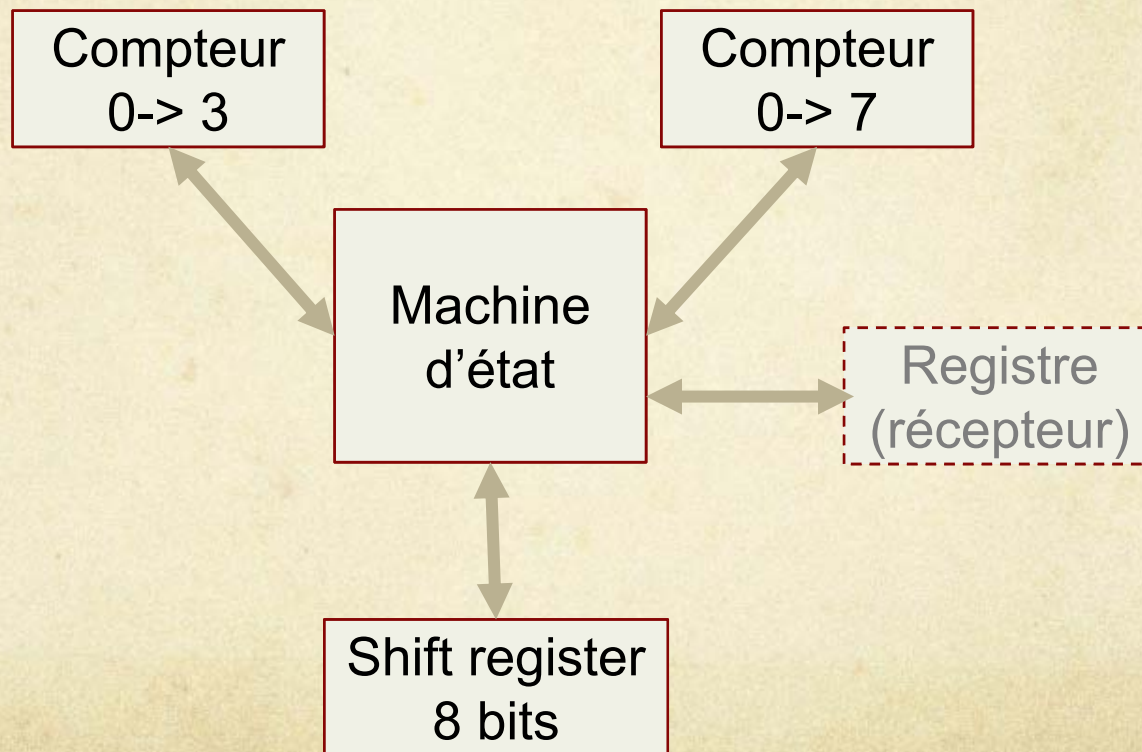
# Réalisation du récepteur

1. Utilisez le **shift register 8 bits** réalisé dans le labo précédent. Soyez certain qu'il fonctionne correctement
2. Créez un **compteur** comptant de 0 à 3 sans utiliser les composants «compteur» de Logisim. Ce compteur comprend une entrée remise à 0 synchrone et une sortie indiquant que le compteur a atteint la valeur max.
3. Créez un autre **compteur** comptant de 0 à 7 sans utiliser les composants «compteur» de Logisim. Ce compteur comprend une entrée remise à 0 synchrone, une entrée enable et une sortie indiquant que le compteur a atteint la valeur max
4. Créez une **machine d'état de type 1 parmi M** qui contrôle les deux compteurs, le registre pour stocker le résultat et le shift register. Le nombre d'états doit être minimum.



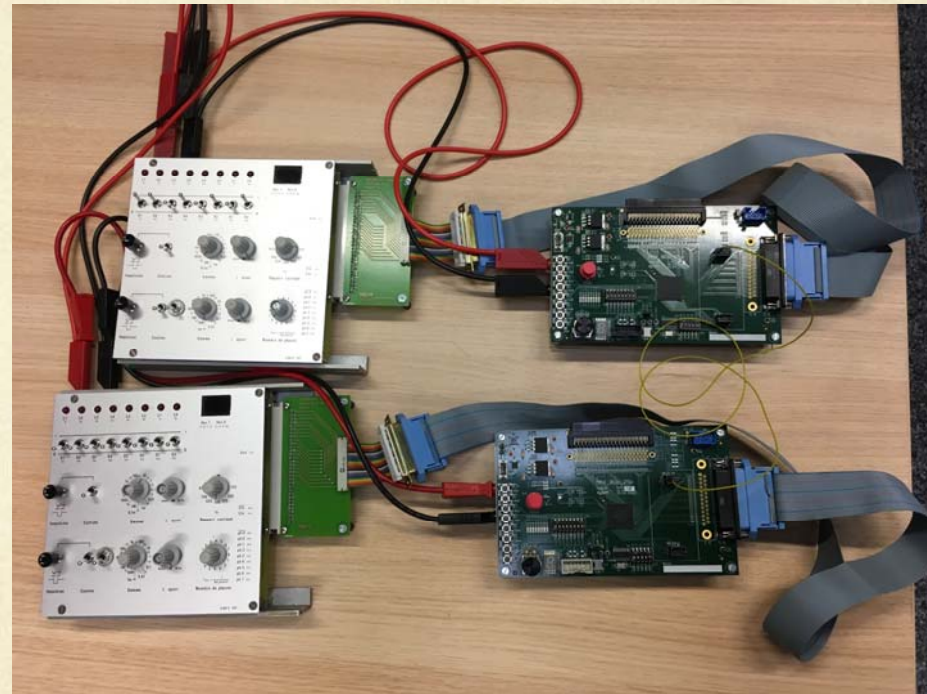
# Architecture émetteur ou récepteur

Attention: ce diagramme indique uniquement les liaisons de contrôle par la machine d'état. Il doit être complété par les chemins de données.



# Test sur cartes (1)

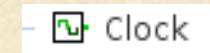
- Attention : les deux cartes doivent être connectées à la même alimentation.
- Le fil reliant les deux cartes (ligne série: câble jaune) est connecté sur la pin 9 du connecteur MEZ1, Attention au détrompeur sur le connecteur MEZ1 (coin en haut à droite).





# Test sur cartes (2)

- Dans votre circuit (au top) sous logisim, connectez le composant "Clock" (librairie Wiring) à l'entrée clk de votre design.
- Les autres entrées et sorties de votre design sont connectées à des pins inputs ou outputs, comme utilisé normalement.
- Ensuite ouvrir la fenêtre FPGA commander, et choisir les paramètres :
  - Choose target board : **MAX\_V\_CONSOLE**
  - Choose tick frequency : **32 Hz**
- Les deux cartes (émetteur et récepteur) doivent être programmer avec la même fréquence.



# Test sur cartes (3)

- **Emetteur**, placez les éléments de la manière suivante:
  - Les 8 bits de la donnée à transmettre sur les 8 switches (S7 à S0) de la console.
  - L'entrée **Reset** sur le bouton poussoir SW8 de la carte **MAX\_V**.
  - L'entrée **Send** sur le bouton poussoir SW1 de la carte **MAX\_V**.
  - La sortie **Done** sur la LED L0 de la console.
  - La sortie série 1 bit sur la pin 9 du connecteur MEZ1 de la carte **MAX\_V**.





# Test sur cartes (4)

- Récepteur, placez les éléments de la manière suivante:
  - Les 8 bits de la donnée reçue sur les 8 LEDs (L7 à L0) de la console. Affichage uniquement après avoir reçu tous les bits de la donnée
  - L'entrée Reset sur le bouton poussoir SW8 de la carte MAX\_V.
  - L'entrée série 1 bit sur la pin 9 du connecteur MEZ1 de la carte MAX\_V.



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