

Laboratoire #3 : Circuits séquentiels

INF3500 - Conception et réalisation de systèmes numériques
Automne 2018

Objectifs

Ce laboratoire a trois objectifs :

- vous familiariser avec les outils utilisés lors des laboratoires du cours INF3500 ;
- vous faire implémenter un circuit séquentiel ; et
- vous faire utiliser le concept de machine à état dans un circuit séquentiel.

Préparation au laboratoire

Avant d'arriver au laboratoire, suivre les étapes suivantes :

1. Revoir la matière des cours des semaines 5 et 6 ; et,
2. Lire le guide d'utilisation de Vivado.
3. Lire la FAQ du laboratoire

Familiarisation avec les outils

Suivre les instructions du guide pratique d'utilisation de Vivado. Ce tutoriel est indispensable pour faire la simulation, la synthèse, l'implémentation et la programmation du FPGA.

Contexte : circuits séquentiels

Les architectures utilisant des circuits séquentiels sont très répandues dans le domaine des circuits numérique. Par exemple, les circuits séquentiels sont utilisés dans les processeurs à usage général, les processeurs graphiques, les modules spécialisés dans le cryptominage, les unités de calculs dans un système de contrôle d'un drone, etc.

De tels circuits numériques sont organisés en blocs de logique combinatoire séparés par de la logique séquentielle.

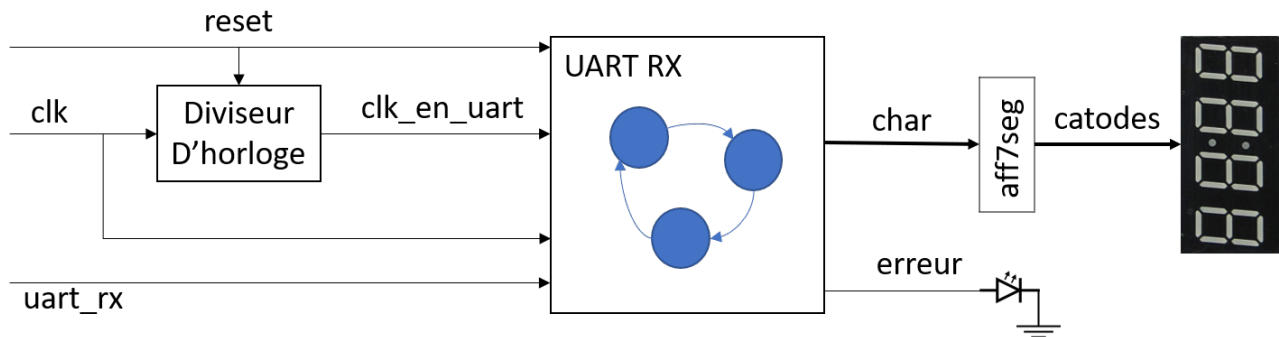
Par exemple, les machines à état, qui sont utilisées dans de nombreux circuits numériques, sont implémentées avec de la logique séquentielle, utilisées pour enregistrer l'état actuel dans une mémoire (une *bascule*), et avec de la logique combinatoire pour calculer la valeur de l'état futur.

Conception d'un récepteur UART

L'UART (en anglais Universal Asynchronous Receiver-Transmitter) est une interface de communication à basse vitesse utilisée dans les systèmes embarqués.

L'UART utilise un protocole de communication bidirectionnel et asynchrone entre un récepteur et un émetteur. Pour plus d'information à propos de l'UART, consultez l'article Wikipedia.

Dans ce TP, vous devez implémenter un **récepteur** UART comme celui présenté dans la figure suivant.



La conception du récepteur UART est décomposée en deux parties présentées ci-dessous.

Partie A : Diviseur d'horloge

Vous devez développer un diviseur d'horloge paramétrisable pour le récepteur UART. Le diviseur d'horloge doit générer deux signaux de sortie : une horloge dont la fréquence est divisée par un facteur k par rapport à l'horloge d'entrée, ainsi qu'un signal d'activation synchronisé sur le front montant de l'horloge de sortie, et valide pendant un cycle de l'horloge d'entrée.

Livrable : Remettre le code de votre module. Effectuez une démonstration du diviseur configuré pour 1Hz et utilisez le signal généré pour faire clignoter une DEL (à la fin du laboratoire).

Partie B : Récepteur UART

Vous devez développer un récepteur UART pour décoder un message envoyé par un module d'émission UART. Les requis pour le module de réception sont décrits ci-dessus :

- Développez une machine à état pour implémenter le récepteur.
- La vitesse d'opération doit être 57600 bauds.
- Votre module doit recevoir des trames UART contenant une donnée encodée sur 8 bits, avec parité impaire et stop bit simple.
- Utilisez le signal d'activation généré par le diviseur d'horloge comme marqueur de temps pour votre module.
- Échantillonnez au milieu de chaque bit transmis pour assurer un échantillage correct.
- Affichez la donnée décodée par votre récepteur UART sur un afficheur 7 segments. Le code de base de l'afficheur est disponible ici.
- Une DEL doit s'allumer en cas d'erreur de parité. La DEL doit rester allumée jusqu'à ce qu'une trame correcte soit reçue.

Livrable : Remettre le code de votre module, le diagramme des transitions d'états ainsi qu'un diagramme bloc de votre circuit.

Partie C : Synthèse et Implémentation

Synthétisez et implémentez votre module sur la carte Nexys4 DDR. Utilisez le fichier uart.xdc ci-dessous comme base pour votre implémentation.

```
## Clock signal
set_property -dict { PACKAGE_PIN E3 IOSTANDARD LVCMOS33 } [get_ports { CLK100MHZ }];
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK100MHZ}];

##7 segment display

set_property -dict { PACKAGE_PIN T10 IOSTANDARD LVCMOS33 } [get_ports { CA }];
set_property -dict { PACKAGE_PIN R10 IOSTANDARD LVCMOS33 } [get_ports { CB }];
set_property -dict { PACKAGE_PIN K16 IOSTANDARD LVCMOS33 } [get_ports { CC }];
set_property -dict { PACKAGE_PIN K13 IOSTANDARD LVCMOS33 } [get_ports { CD }];
set_property -dict { PACKAGE_PIN P15 IOSTANDARD LVCMOS33 } [get_ports { CE }];
set_property -dict { PACKAGE_PIN T11 IOSTANDARD LVCMOS33 } [get_ports { CF }];
set_property -dict { PACKAGE_PIN L18 IOSTANDARD LVCMOS33 } [get_ports { CG }];

set_property -dict { PACKAGE_PIN H15 IOSTANDARD LVCMOS33 } [get_ports { DP }];

set_property -dict { PACKAGE_PIN J17 IOSTANDARD LVCMOS33 } [get_ports { AN[0] }];
set_property -dict { PACKAGE_PIN J18 IOSTANDARD LVCMOS33 } [get_ports { AN[1] }];
set_property -dict { PACKAGE_PIN T9 IOSTANDARD LVCMOS33 } [get_ports { AN[2] }];
set_property -dict { PACKAGE_PIN J14 IOSTANDARD LVCMOS33 } [get_ports { AN[3] }];
set_property -dict { PACKAGE_PIN P14 IOSTANDARD LVCMOS33 } [get_ports { AN[4] }];
set_property -dict { PACKAGE_PIN T14 IOSTANDARD LVCMOS33 } [get_ports { AN[5] }];
set_property -dict { PACKAGE_PIN K2 IOSTANDARD LVCMOS33 } [get_ports { AN[6] }];
set_property -dict { PACKAGE_PIN U13 IOSTANDARD LVCMOS33 } [get_ports { AN[7] }];

##USB-RS232 Interface

set_property -dict { PACKAGE_PIN C4 IOSTANDARD LVCMOS33 } [get_ports { UART_TXD_IN }];
set_property -dict { PACKAGE_PIN D4 IOSTANDARD LVCMOS33 } [get_ports { UART_RXD_OUT }];

## LEDs

#set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { LED[0] }];

##Buttons

#set_property -dict { PACKAGE_PIN N17 IOSTANDARD LVCMOS33 } [get_ports { reset }];
```

Pour tester votre circuit, vous devez communiquer avec un ordinateur. Utilisez un terminal serial, comme PuTTY, pour envoyer des commandes par UART.

Rapport

Rédigez votre rapport selon les directives demandées ici.

Barème

Critère	points
Partie A	0.5
Partie B	2.5
Partie C	0.8
Rapport : présentation et qualité du français	0.2
Total	4.0