

# N7\_SN\_1A

## Architecture des ordinateurs - Semestre 5

### TP2

#### Exercice 1

Pour coder les différents états d'un circuit séquentiel, on utilisera principalement des bascules D et des bascules T, dont le fonctionnement est régi par les équations suivantes :

- Bascule D (Delay) : l'état X (=sortie) à l'instant  $n+1$  et égal à l'entrée D à l'instant  $n$ .  
 $X := D$  on clk reset when rst enabled when en // en est facultatif, égal à 1 par défaut
- Bascule T (Trigger) : l'état  $X_{n+1}$  est égal à  $X_n$  si l'entrée  $T=0$ , et  $= /X_n$  si  $T=1$   
 $X := /T * X + T * /X$  on clk reset when rst enabled when en

Dans un module bascules, décrire et tester le fonctionnement de la bascule D, et de la bascule T.

#### Exercice 2

Un registre est un ensemble de cases mémoire binaires (8, 16, 32, ...) qui sert à mémoriser des informations utiles à l'exécution des instructions dans un processeur : opérandes, adresses, etc. L'entrée du registre y est mémorisée sur le top d'horloge.

Ecrire et tester le module reg8\_D (rst, clk, en, e[7..0] : sr[7..0]) en utilisant des bascules D

Ecrire et tester le module reg8\_T (rst, clk, en, e[7..0] : sr[7..0]) en utilisant des bascules T

#### Exercice 3

Ecrire et tester le module count2 (rst, clk, en : c[1..0]) qui compte de façon cyclique : 0, 1, 2, 3, 0, ... lorsque l'entrée en = 1

#### Exercice 4

Réaliser le module count4 en utilisant le module count2. Tester.

#### Exercice 5

On souhaite réaliser un compteur 4 bits initialisable :

count\_init4 (rst, clk, en, init, e[3..0] : c[3..0]) qui est initialisé avec la valeur e[3..0] lorsque inti=1, et compte de façon cyclique 0, 1, 2, ..., 15, 0, ... lorsque init=0.

Réaliser et tester ce module.

#### Exercice 6

En utilisant le module count\_init4, réaliser et tester le circuit count\_012789 (rst, clk, en : c[3..0]) qui compte de la façon suivante : 0, 1, 2, 7, 8, 9, 0, ...

#### Exercice 7

Réaliser le module count4\_b1\_b2 (rst, clk, count, init, b1[3..0], b2[3..0] : c[3..0]), qui fonctionne de la façon suivante (sans utiliser l'entrée en des bascules) :

- Est initialisé avec b1 quand init=1 et count=0
- S'incrémente quand init=0 et count=1, en s'arrêtant quand il atteint la valeur b2
- N'évolue pas dans les autres cas

### **Exercice optionnel**

Réaliser et tester le circuit count\_pair\_impair (rst, clk, en, p : c[3..0]) qui fonctionne de la façon suivante (lorsque en=1) :

P = 1 : passe de la valeur courante à la première valeur paire supérieure

P = 0 : passe de la valeur courante à la première valeur impaire supérieure