

N7_SdN_1A

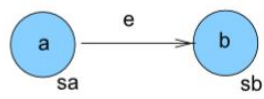
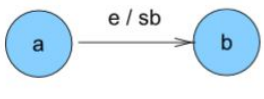
Architecture des ordinateurs

Semestre 2

TD2 : Circuits séquentiels

1- Rappel

- Circuit séquentiel = les sorties dépendent des entrées et de l'état courant
- Un circuit séquentiel peut être modélisé par un graphe d'états :
 - on passe d'un état « a » à un état « b » lorsque l'entrée « e » prend une certaine valeur
 - on distingue deux types de circuits

 <p>MOORE</p>	 <p>MEALY</p>
La sortie dépend de l'état et n'est valide que lorsqu'on atteint le nouvel état	La sortie est valide dès que l'entrée « e » change

- Lorsque les changements d'états sont cadencés par une horloge, le circuit est dit synchrone .
- Pour coder les différents états d'un circuit, on utilisera des bascules, qui sont des mémoires 1 bit. Il en existe 3 types classiques, dont voici les équations caractéristiques :

- Bascule D (Delay) : l'état X (=sortie) à l'instant n+1 et égal à l'entrée D. On notera :

$$X := D \text{ on CLK, reset when RST}$$

Bien noter le signe « := » qui indique une assignation séquentielle

- Bascule T (Trigger) : l'état X^{n+1} (à l'instant n+1) et égal à X^n si l'entrée T=0, et $\neg X^n$ si T=1 ; soit :

$$X := \neg T * X + T * \neg X \text{ on CLK, reset when RST}$$

- Bascule JK (2 entrées J et K) et dont l'équation est :

$$X := \neg K * X + J * \neg X \text{ on CLK, reset when RST}$$

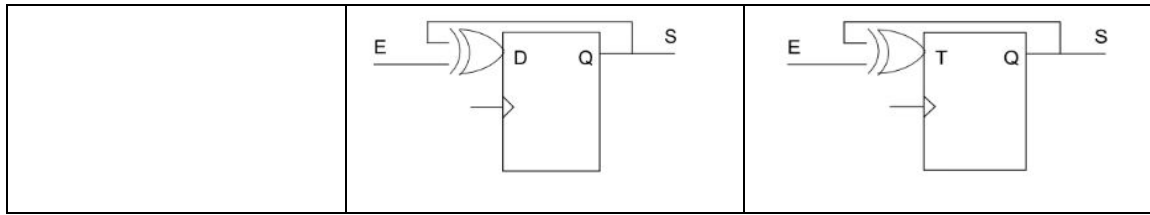
- On peut écrire de façon plus générale :

$$Q := \langle \text{expression} \rangle \text{ on CLK, reset when RST } [, \text{ enabled when EN}]$$

La partie « enabled when » est facultative.

Pour un circuit séquentiel synchrone ayant N états, on utilisera M bascules, pour représenter ces N états, avec $2^M \geq N$. Dans certains cas, on peut être amené à utiliser une bascule par état (one-shot encoding).

Exemple : indiquer pour les deux circuits suivants le vecteur des entrées, le vecteur d'état, et le vecteur des sorties ; et décrire leur fonctionnement.



Exercice 1 : On souhaite réaliser le circuit qui pilote le fonctionnement d'une chaudière, avec :

- Entrées :
 - t_basse : indique que la température est basse (par exemple, inférieure à 19°) et commande la mise en marche de la chaudière
 - t_haute : indique que la température est haute (par exemple, supérieure à 22°) et commande l'arrêt de la chaudière
 - Sortie : M indiquant si la chaudière est en marche (M=1) ou à l'arrêt.
- A- Dessiner le graphe d'états correspondant
 B- Dresser la table de transitions
 C- Effectuer une assignation des états, puis dresser la table de transitions instanciée
 D- Réaliser ce circuit en utilisant une bascule D. On doit retrouver l'équation de la bascule JK

Exercice 2 : On souhaite réaliser le circuit « count4 » qui compte de façon cyclique : 0, 1, 2, ..., 15, 0, ... lorsque l'entrée en = 1 count4 (rst, clk, en : s[3..0])

- A- Quel est le nombre d'états de ce circuit
 B- Quel est le nombre de bascules nécessaires
 C- Donner la description de ce circuit en shdl. Sans passer par la table de transitions, penser à la bascule T.

Exercice3 : Transformer le circuit « count4 » en count4Z(rst, clk, en, sclr : s[3..0]) qui compte comme précédemment lorsque en=1 et sclr=0, et se remet à 0 lorsque en=1 et sclr=1

Exercice4 : en utilisant le module count4z, réaliser un circuit qui compte de 0,1, ..., 9, 0 ...
 count0_9 (rst, clk, en : s[3..0])

Exercice5 : réaliser le circuit count_pair_impair (rst, clk, en, p : s[2..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

Deux possibilités : 1 bascule par état, et des sorties évaluées en fonction des états, mais c'est lourd.

Ou 3 bascules (table de transitions et Karnaugh) :

$$s[2] := s[2] * s[0] * p + /s[2] * s[1] * p + /s[2] * s[1] * s[0] + s[2] * /s[1]$$

on clk, reset when rst, enabled when en

$$s[1] := s[1] * s[0] * p + /s[1] * p + /s[1] * s[0] \text{ on clk, reset when rst, enabled when en;}$$

$$s[0] := /p \text{ on clk, reset when rst, enabled when en}$$