

N7_SdN_1A

Architecture des ordinateurs

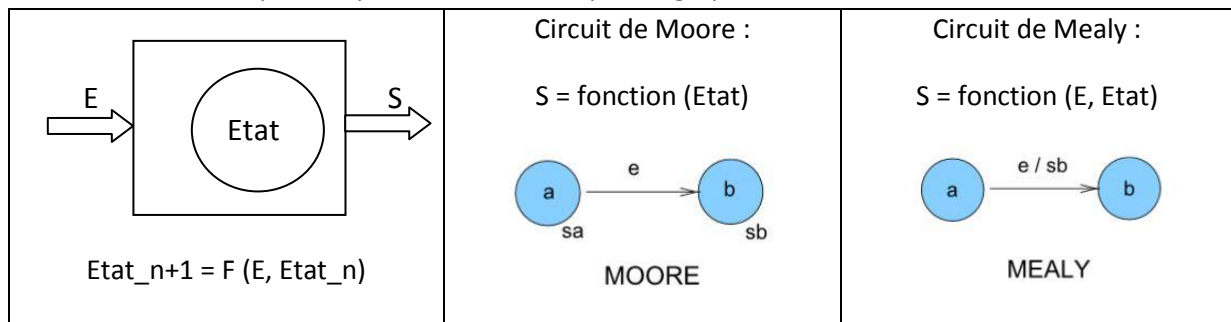
Semestre 2

TD2 : Circuits séquentiels

1- Rappel (lecture, questions, réponses et explications : 20 mn)

Circuit séquentiel = circuit qui conserve un état qui évolue en fonction des entrées et de lui-même.

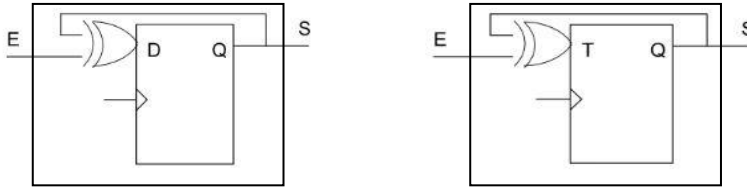
- On notera $Etat_n$ l'état à l'instant n (courant) et $Etat_n+1$ l'état à l'instant $n+1$ (suivant)
- Un circuit séquentiel peut être modélisé par un graphe d'états



- Lorsque les changements d'états sont cadencés par une horloge, le circuit est dit synchrone. On travaillera exclusivement avec des circuits synchrones.
- Pour coder les différents états d'un circuit, on utilisera des bascules, qui sont des mémoires 1 bit. Historiquement, il en existe 3 sortes, 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 :
 $X := D$ // bien noter l'affectation « $:=$ » qui indique une équation 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 $T=1$. Ce qui donne l'équation caractéristique : $X := \neg T * X + T * \neg X$
 - Une troisième bascule moins utilisée, nommée JK (2 entrées J et K) et dont l'équation est : $X := \neg K * X + J * \neg X$
- On utilisera principalement des bascules de type D, simples à décrire, et dont la syntaxe shdl est la suivante : $Q := D$ on clk reset when rst enabled when en
 - « D » pouvant être une expression logique , sous la forme de somme de min-termes
 - « clk » est le signal d'horloge qui fait passer la bascule de l'état n à l'état $n+1$
 - « rst » est le signal qui effectue une remise à zero asynchrone (forçage) de la bascule
 - « en » est le signal qui permet d'activer la bascule. L'expression « enabled when en » est facultative. Par défaut, « en » est à 1.
- Dans certaines situations, le fonctionnement du circuit ressemble à celui d'une bascule T (changements d'états bine identifiés), et il peut s'avérer intéressant de le réaliser en utilisant des bascules T :
 - $X := \neg T * X + T * \neg X$ on clk reset when rst ...
 - $T = \text{fonction} (E, X)$
- 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 pour faciliter la description du circuit.

Exemple : (5 + 5 mn)

Indiquer pour les deux circuits suivants le vecteur des entrées (entrées), le vecteur d'état (états), et le vecteur des sorties (sorties) ; et décrire ce qu'ils réalisent.



Exercice 1 : (15 + 10 mn)

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 (M=0).
- A- Dessiner le graphe d'états correspondant
B- Dresser la table de transitions (états)
C- Effectuer une assignation des états, puis dresser la table de transitions instanciée (bascules)
D- Réaliser ce circuit en utilisant une bascule D

Exercice 2 : (10 + 5 mn)

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- Identifier les cas où S[0] change (0 vers 1 et inversement), où S[1] change, etc.
D- Donner la description de ce circuit en shdl sans passer par la table de transitions

Exercice3 : (10 + 5 mn)

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 de façon synchrone lorsque en=1 et sclr=1

Exercice4 : (5 + 5)

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

Exercice5 : (en fonction du temps qui reste)

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

Pour simplifier le travail, on utilisera une bascule par sortie, donc 3 bascules.