

TP 3 : Circuits séquentiels (suite)

Rappels

R	S	Q_{n+1}	\overline{Q}_{n+1}	J	K	Q_{n+1}	\overline{Q}_{n+1}
0	0	Q_n	\overline{Q}_n	0	0	Q_n	\overline{Q}_n
1	0	0	1	0	1	0	1
0	1	1	0	1	0	1	0
1	1	x	x	1	1	\overline{Q}_n	Q_n

TABLE 1 – Table d'excitation des bascules RS (gauche) et JK (droite)

Équation de fonctionnement d'une bascule D : $Q_{n+1} = D_n$

Équation de fonctionnement d'une bascule JK : $Q_{n+1} = Q_n \overline{K}_n + \overline{Q}_n J_n$

Méthodologie

Tout circuit séquentiel peut être modélisé formellement par un automate. Un automate est un opérateur séquentiel dont l'état et les sorties futures sont fonction des entrées et de l'état présent de l'automate. Un automate utilise des bascules non transparentes (souvent des bascules D) pour mémoriser l'état présent et des opérateurs combinatoires pour générer les sorties et l'état futur. L'utilisation de bascules non transparentes est nécessaire à cause du rebouclage (indirecte) des sorties des bascules sur les entrées. Plus formellement, on considère ici un automate – déterministe à états finis – défini par un 6-uplet (Q, q_0, E, S, T, O) avec :

- Q l'ensemble des états
- $q_0 \in Q$ l'état initial
- E l'alphabet d'entrée
- S l'alphabet de sortie
- T la fonction de transition ($T : Q \times E \rightarrow Q$)
- O la fonction de sortie
 - Si automate de Moore, $O : Q \rightarrow S$
 - Si automate de Mealy, $O : Q \times E \rightarrow S$

On ne s'intéressera dans un premier temps qu'à l'automate de Moore donné ci-dessous :

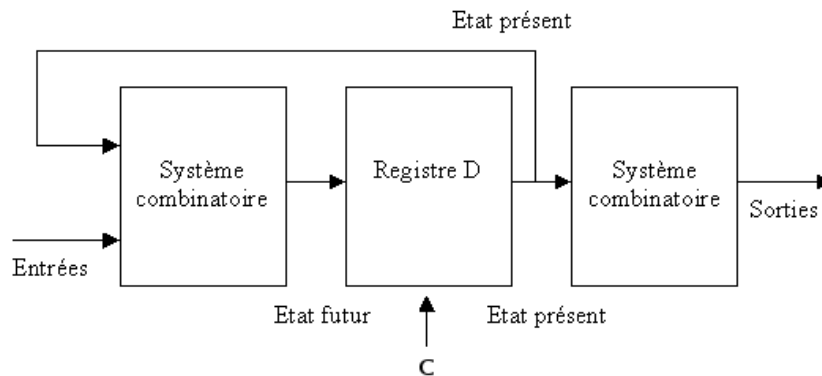


FIGURE 1 – Automate de Moore

Pour réaliser le circuit séquentiel répondant à un problème, la méthodologie est la suivante :

1. Représentation de l'automate (de Moore/Mealy) par un graphe d'état.
2. Choisir un codage pour les états de l'automate, les entrées et les sorties.
3. Établir le tableau de Karnaugh pour la fonction de transitions d'états (selon l'état courant et l'entrée)
4. Dédire pour chaque bit du codage des états sa fonction booléenne en fonction des bits des états précédents et des entrées.
5. Établir le tableau de Karnaugh pour la fonction de sortie (selon l'état courant si Moore, l'état courant et l'entrée si Mealy)
6. Dédire pour chaque bit du codage des sorties sa fonction booléenne en fonction des bits des états précédents (et des entrées si Mealy).
7. Choisir un type de bascule.
8. Adapter les fonctions booléennes au fonctionnement de la bascule choisie.

Les opérations booléennes revisited

On sait faire ça, non ?

On souhaite effectuer la somme logique (OU) du contenu de deux registres A et B de 4 bits chacun et obtenir le résultat dans le registre B , soit

$$\text{contenu}(A) + \text{contenu}(B) \rightarrow B$$

Le circuit doit effectuer cette somme en parallèle sur tous les bits. Soit $\text{contenu}(A_i)$ le i -ème bit ($i = 0, 1, 2, 3$) du registre A (idem pour B). En n'utilisant que des bascules RS avec des NOR :

1. Donnez la table d'états correspondant au sous-circuit réalisant la somme logique au niveau du bit (elle comprendra $\text{contenu}(A_i)$ et $\text{contenu}(B_i)$ au temps t (avant le transfert) $\text{contenu}(B_i)$ après le transfert et les entrées de la bascule RS) ;
2. Donnez l'expression logique simplifiée au maximum exprimant chaque entrée de la bascule en fonction des entrées du sous-circuit ;
3. Donnez le schéma du sous-circuit obtenu et en déduire le schéma du circuit complet en supposant que les registres contiennent initialement les valeurs A_i et B_i .
4. Généralisez ce circuit avec une horloge afin que
 - les registres se mettent à jour sur son front montant,
 - la somme logique soit calculée sur son front descendant.

Le Digicode

Open Sesame !

On souhaite réaliser un digicode ouvrant une porte lorsque la séquence "1664" est saisie.

L'alphabet d'entrée est $\{0,1,2,3,4,5,6,R\}$ où R est le signal de reset du digicode quand la porte se referme.

L'alphabet de sortie est donc $\{\text{Opened}, \text{Closed}\}$.

Les séquences suivantes mènent à l'ouverture de la porte : "1664", "1111664", "1431664", "1661661664".

1. Représentez le fonctionnement du digicode avec un automate de Moore et donnez les fonctions de transitions/sortie avec des bascules *flip-flop* D.
2. Réalisez le circuit sur TkGate.
3. (*Bonus*) Idem avec des bascules *flip-flop* JK.

Feux rouges

`tkgate2.0 just crashed : 5 victims.`

On considère un automate contrôlant les feux à un carrefour entre une grande route et un chemin. Sur le chemin, des détecteurs repèrent la présence éventuelle d'une voiture en attente. Le feu ne passe au rouge sur la route qu'après une durée appelée "temps long" pendant laquelle un véhicule attend sur le chemin, et le feu ne peut pas rester rouge sur la route plus longtemps que cette durée "temps long", même si des véhicules sont présents sur le chemin. Une durée appelée "temps court" correspond à la durée des feux oranges. On appelle TC (*resp.* TL) la variable associée au temps court (*resp.* temps long), telle que $TC = 1$ (*resp.* $TL = 1$) si le temps écoulé depuis la dernière remise à zéro du compteur est supérieur à "temps court" (*resp.* "temps long") et $TC = 0$ (*resp.* $TL = 0$) sinon.

On souhaite réaliser la version Moore de l'automate. Les états internes sont :

- RV (Feux Route Verts et Feux Chemin Rouges)
 - RO (Feux Route Oranges et Feux Chemin Rouges)
 - CV (Feux Route Rouges et Feux Chemin Verts)
 - CO (Feux Route Rouges et Feux Chemin Oranges)
1. Explicitiez l'automate contrôlant les feux du carrefour ; donnez ses fonctions de transitions/sortie avec bascules *flip-flop* D. Dans un premier temps on peut considérer que les variables TC et TL sont données en entrées (ce n'est pas à vous de les calculer).
 2. Réalisez le circuit sur **TkGate**.