CACHAN
Département d'électronique

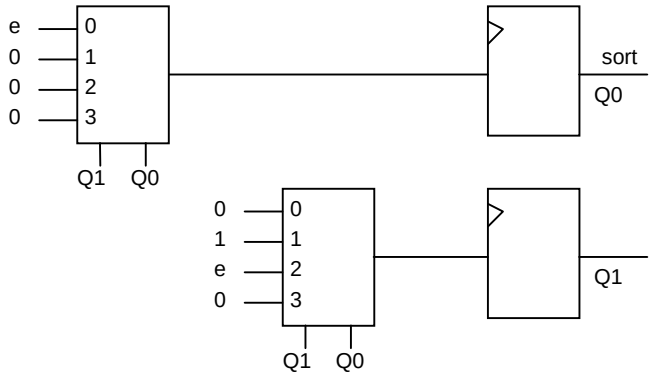
Cachan le 13 avril 2000

Travaux dirigés d'informatique industrielle : bascules synchrones <u>corrigé</u>

« Rappel » : commenter les diagrammes, bien éclaircir les notions de « avant » et « après » le front d'horloge.
Montrer, sur la JK, par exemple, le passage du diagramme de transitions aux équations de commande d'une bascule D.
Noter que dans la synthèse avec bascule D on écrira les valeurs de l'état d'arrivée, dans les autres cas on s'intéresse aux changements de valeurs (mémoire implicite) pour en déduire les commandes T ou J-K.

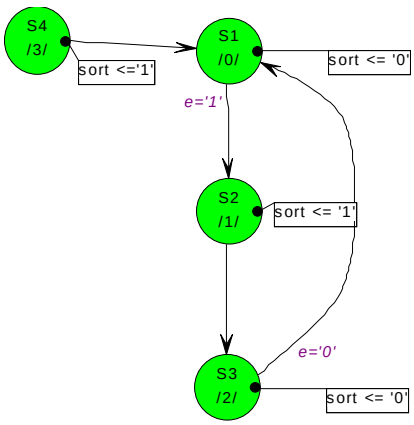
Deux petits automates

- Équations de commande : (sort = bascule0)
  sort.D = e * /sort.Q * /basc1.Q
  basc1.D = e * /sort.Q * basc1.Q + sort.Q * /basc1.Q
  Dans la syntaxe palasm : / pour non, * pour et, + pour ou
- Multiplexeurs 4 vers 1 :



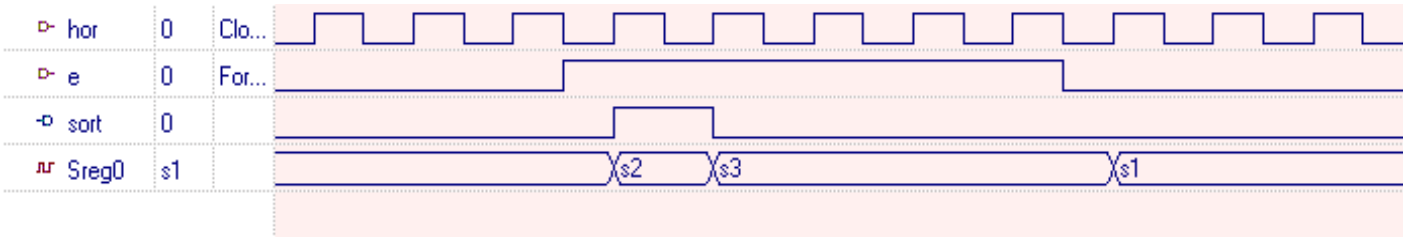
Deux objectifs dans cette partie : « rappel » concernant les multiplexeurs, fonction combinatoire importante, utilisée dans beaucoup de circuits programmables pour générer des blocs de calcul combinatoires. Lié directement aux instructions de contrôle comme l'équivalent VHDL du switch (case … when).

- En déduire un diagramme de transitions qui décrit le fonctionnement du montage.



(on peut passer par une table de transitions intermédiaire)

- Quelle est l'allure du signal « sort » si le signal « e », initialement à '0' prend la valeur '1' pendant cinq périodes de l'horloge puis revient à '0' ?



- Proposer un schéma qui réalise la même fonction réalisé avec des bascules T puis avec des bascules JK.

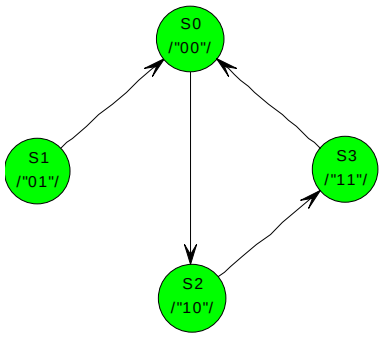
basc1.T = /e * basc1.Q + sort.Q
sort.T = e * /basc1.Q + sort.Q

Pour une bascule JK on peut utiliser l'équation de transition : T = J * /Q + K * Q

basc1.J = sort.Q basc1.K = /e + sort.Q
sort.J = e * /basc1.Q sort.K = 1

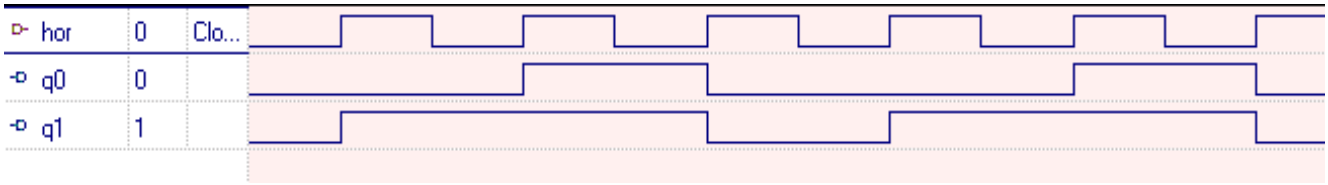
On considère le schéma suivant :

- Établir un diagramme de transitions qui décrit le fonctionnement du montage.



(on peut passer par une table de transitions intermédiaire)

- En admettant que les deux bascules sont initialement à 0, établir un chronogramme qui fait apparaître l'horloge et les deux sorties Q1 et Q0.



Ou comment réaliser un diviseur de fréquence par trois.

- L'état initial des bascules a-t-il une importance ?

Non, passé, éventuellement, la première période d'horloge.

Equivalence des différents types de bascules synchrones.

Les trois catégories de bascules synchrones élémentaires sont la bascule D, la bascule T et la bascule J-K.

- Etablir les logigrammes qui permettent de passer d'un type à l'autre (réaliser une J-K avec une D, et réciproquement, une T avec une D et réciproquement etc.)

D = J * /Q + /K * Q
D = T ⊕ Q
Etc …

Petites synthèses

1. **Génération de séquences**

On souhaite réaliser une fonction logique synchrone qui fournit en sortie les signaux suivants :

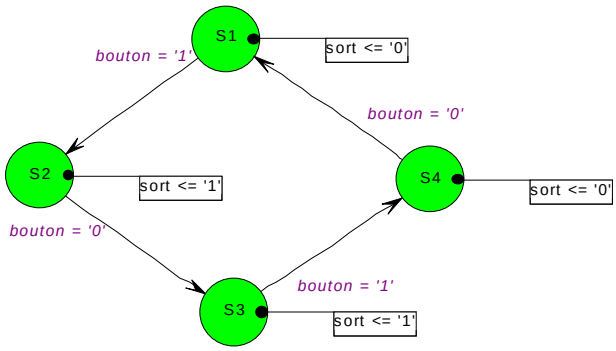
- Etablir un diagramme de transition qui permet de répondre au problème.
  Voir le diviseur par trois précédent.
- Proposer une solution qui fait appel à des bascules D, puis à des bascules T puis J-K.
  Idem, je n'avais pas fait attention à l'identité des schémas.
- Préciser quelle doit être la fréquence de l'entrée d'horloge.
  1/(3µs)

Les chronogrammes précédents sont modifiés :

- Montrer qu'il faut rajouter à la solution précédente une bascule.
  La périodicité est de 6, il faut rajouter un signal d'état interne : 3 bits.
  La suite est un exercice de numérotation.
- Proposer une solution qui fait appel à des bascules D.
  On espère qu'à la fin de ces exercices ils sauront passer d'un STD à un schéma de commande.

2. Un bouton poussoir

- Analyser les faces cachées du problème.
  Cela redevient plus drôle. Le piège est que le système change d'état au rythme de l'horloge tant que le bouton est appuyé ; ce qui n'est pas la fonction demandée. Il faut donc synchroniser l'évolution de l'automate sur les changements de valeur de l'entrée. On retrouve un principe du même style que le premier exo :
- Montrer que l'on a besoin de prévoir un système à quatre états, établir son diagramme de transitions.

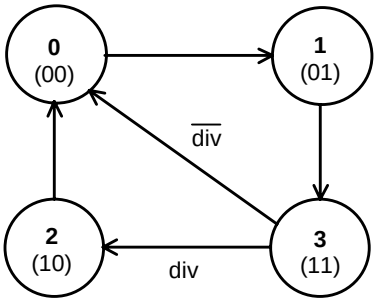


  Ici on pourrait évidemment commencer à discuter MOORE ou MEALY, mais on reviendra sur le sujet un peu plus tard.
  Autre remarque : il est possible de donner une version asynchrone du problème, mais cela nécessite de surveiller l'adjacence des états successifs. Ce n'est peut-être pas trop notre propos que de discuter ce genre de choses.
- Proposer une réalisation de l'automate avec quelques bascules et quelques portes.
  basc1.D = bouton
  basc0.D = /basc0.Q * basc1.Q * /bouton + basc0.Q * /basc1.Q + basc0.Q * bouton
  sort = basc0.Q * /basc1.Q + /basc0.Q * basc1.Q

3. Un diviseur de fréquence programmable

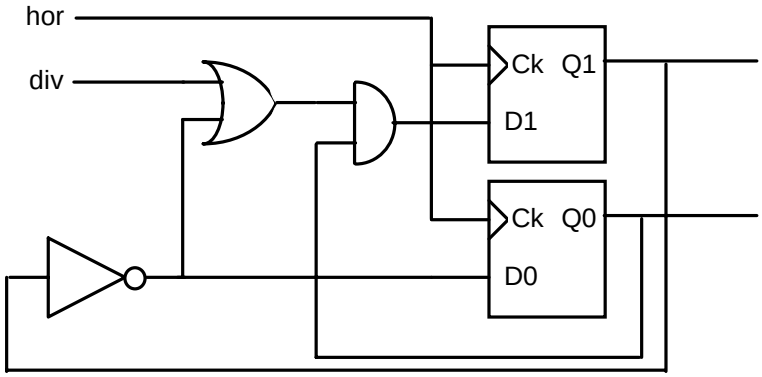
L'exercice est traité dans le bouquin :

- Établir un diagramme de transitions qui satisfasse au problème.



Etat : (Q1 Q0)

- Proposer une réalisation avec quelques bascules D, puis T et des portes logiques.



Tous ces exos tournent autour des mêmes choses et sont assez classiques, il est important que les étudiants fassent eux-mêmes, le prof étant là pour discuter les solutions. La démarche automate prend du temps, les années passées nous nous sommes rendus compte que l'on passait trop vite dessus, d'où des difficultés dès le premier problème nouveau rencontré. Il faut également faire comprendre aux étudiants que les méthodes qu'ils acquièrent ici leur permettent de traiter n'importe quel problème de logique classique, synthèses de compteurs divers, utilisation de compteurs etc.

A mon sens il faut accepter de passer dans un premier temps par des tables de transitions, et pousser les étudiants à raisonner progressivement sur les diagrammes eux-mêmes, sans passer par les tables. Autre chose : faire comprendre qu'avec un peu d'habitude on fait le lien direct entre STD et chronogrammes. Dans une deuxième étape on associera le 4ème larron : le programme. Le but est de créer des automatismes de visualisation

- Structure du schéma
- STD
- Programme
- Évolution temporelle

4. Addition série.

Cet exercice peut être vu avant les autres exercices de synthèse, ce n'est pas à proprement parlé un exercice « automate », mais une réflexion sur ce qui doit être réalisé par des opérateurs séquentiels (bascule de mémorisation de la retenue) et ce qui peut être réalisé en combinatoire, parce que le calcul est « instantanné » (calcul de la somme de deux chiffres et de la retenue précédente, calcul de la retenue suivante).

- Analyser le problème à résoudre, en français.
  Faire une addition à la main, on voit que une colonne correspond à un temps (de droite à gauche !), qu'il faut savoir additionner trois chiffres, et mémoriser le report pour la colonne suivante (temps suivant).
  Le résultat peut être plus long d'une période d'horloge que les opérandes, d'où la nécessité de prolonger « validout » si un report final est généré.
- Proposer une réalisation avec des bascules D et quelques portes.
  Le problème est alors assez simple à résoudre : un additionneur comlet un bit, une bascule D pour mémoriser la retenue, penser à initialiser cette bascule à 0 entre deux opérations (signal valid). Je dois avoir au bureau un schéma complet si vous le souhaitez, je ne l'ai pas sous la main et j'en ai marre de dessiner.
- Quel est l'intérêt de ce type de fonction (de tels circuits existent effectivement) ?
  Cela peut aller loin : compromis vitesse surface, traitement des données qui arrivent en flots, pipe line etc. On trouve souvent des structures hybrides pour le traitement du signal : arithmétique sur des tranches de 4 ou 8 bits, sérialisation entre les tranches. Le problème reste le même.