

Electronique numérique

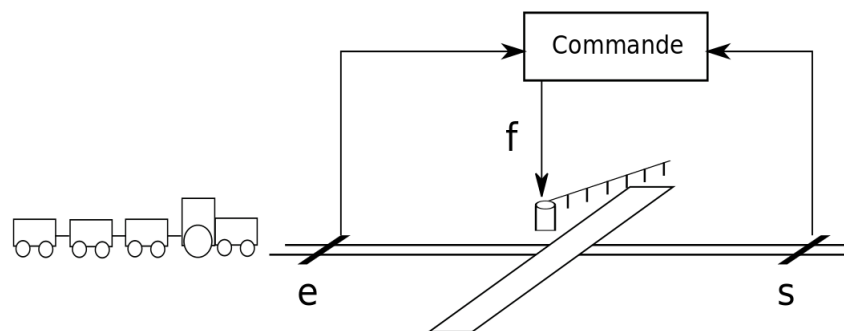
Machines d'états finis (2)

CORRECTIONS

Tous les exercices ne seront pas forcément résolus en TE.

1 Problème de modélisation séquentielle

Le circuit suivant doit commander un passage à niveau.

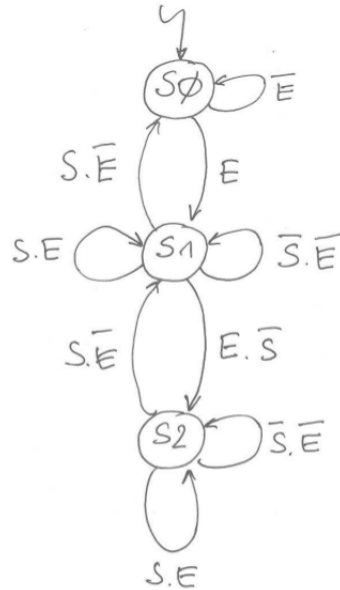


Deux capteurs e (entrée) et s (sortie) signalent l'entrée et la sortie d'un train ; les signaux e et s durent exactement 1 cycle pour le passage d'un train ; il peut y avoir au maximum 2 trains entre les points e et s ; un train peut entrer alors que simultanément un autre sort ; le passage à niveau doit être fermé s'il y a au moins un train entre e et s .

Questions

1. Donner un diagramme d'état ("à bulles") du circuit de commande.
2. Réaliser ce circuit à l'aide de bascules D.

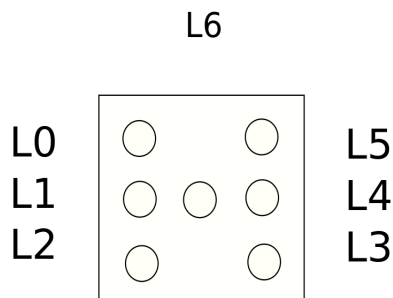
solution (partielle)



Note : ce sujet simple prête toutefois à des interrogations tout à fait légitimes pour l'ingénieur : dans le dernier état, que se passe-t'il si le signal d'entrée s'active alors que S ne s'active pas ? La spécification du système dit " *il peut y avoir au maximum 2 trains entre les points e et s* ", mais est-ce à l'automate de s'en assurer ou est-ce une hypothèse d'entrée parfaitement vérifiée par l'environnement ? Il y a là une ambiguïté qui devrait être discutée. L'ingénieur responsable peut aussi proposer une solution : la plus raisonnable semble être de rajouter un état d'erreur vers lequel transiter. Dans ce cas, une fois cet état atteint, il faudrait une autre entrée (reset) pour sortir l'automate de cet état...Mais cela peut engendrer un ensemble de conséquences sur le réseau amont et aval...Un tel *système électronique* est bel et bien plongé dans un environnement complexe : le "Système".

2 Problème du dé électronique

On doit réaliser une boîte équipée d'un bouton poussoir, de 7 diodes lumineuses (LEDs) d'un circuit logique (à définir) et d'une pile. Ce dispositif doit simuler un dé, lancé aléatoirement : il affiche entre 1 et 6.



Chaque fois que l'utilisateur appuie sur le bouton-poussoir toutes les leds s'allument puis, dès que l'opérateur relâche le bouton, les LEDs se fixent à l'une des configurations ci-dessous.

A l'appui sur le bouton poussoir toutes les configurations sont affichées en séquence à la cadence d'une horloge rapide. L'utilisateur ne peut pas distinguer les combinaisons, du fait de la persistance rétinienne. Seule une illumination moyenne lui apparaît.

Question

- Réaliser ce système, de manière à ce que la fonction de sortie de l'automate soit l'identité. Pour cela choisir un encodage des états approprié.
- Dessiner un schéma-blocs composé de 2 blocs : la logique d'état suivant et les bascules D (registres).
- Choisir un autre encodage (par exemple one-hot) et réaliser à nouveau le circuit.

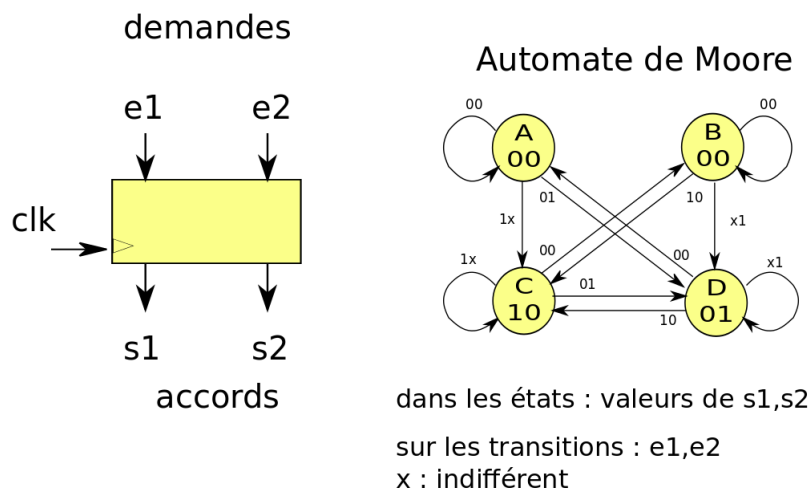
3 Du diagramme à bulles au circuit. Exemple de l'arbitre à priorité tournante (round-robin)

On rappelle que le diagramme à bulle est une représentation, qui se veut intuitive, du comportement séquentiel d'un automate d'états finis (FSM) :

chaque état est représenté par un cercle (bulle) duquel peuvent partir différentes transitions (ou arcs) vers d'autres états possibles. Les transitions sont étiquetées par la condition booléenne amenant à l'état suivant.

On désire réaliser un circuit d'arbitrage entre deux clients susceptibles d'utiliser une ressource qui n'admet qu'un seul occupant à la fois (par exemple deux enfants qui veulent jouer au vélo, et il n'y a qu'un vélo. Le cas se présente également dans les ordinateurs, où différents périphériques peuvent vouloir accéder à un ensemble de fils –bus central– en même temps). Le but du circuit est d'accorder correctement la ressource à un seul client à la fois.

Les clients manifestent respectivement sur e_1 et e_2 leurs désirs d'utiliser la ressource : $e_i = 1$ si le client i désire la ressource, $e_i = 0$ s'il ne la désire pas.



Le circuit génère sur s_1 et s_2 l'attribution de la ressource à chaque client : $s_i = 1$ si la ressource est attribuée au client i , $s_i = 0$ sinon.

Lorsque la ressource est attribuée à un client, elle le demeure tant que le client la désire.

Lorsque la ressource est libre, elle est attribuée au premier qui la demande ; en cas de demandes simultanées, elle est attribuée selon une certaine priorité, mais pour éviter les injustices, la priorité entre les clients change à chaque attribution : le client qui obtient la ressource devient le moins prio-

riatire.

L'analyse du problème conduit à l'automate suivant : il y a quatre états, deux états A et B dans lesquels la ressource est libre, et deux états C et D dans lesquels la ressource est occupée.

Questions

1. Proposer un encodage des états tel que chaque état soit codé sur 2 bits
2. Etablir la fonction de transition du système sous forme de table de vérité. Simplifier.
3. Etablir la fonction de sortie du système. Simplifier.
4. Dessiner le circuit correspondant. Combien de portes faut-il ?

Etat	Encodage entier	Encodage binaire	Q_1	Q_0
A	0	00	0	0
B	1	01	0	1
C	2	10	1	0
D	3	11	1	1

Etat courant			Entrées		Etat futur		
symbole	Q_1	Q_0	e_1	e_2	D_1	D_0	symbole
A	0	0	0	0	0	0	A
A	0	0	0	1	1	1	D
A	0	0	1	0	1	0	C
A	0	0	1	1	1	0	C
B	0	1	0	0	0	1	B
B	0	1	0	1	1	1	D
B	0	1	1	0	1	0	C
B	0	1	1	1	1	1	D
C	1	0	0	0	0	1	B
C	1	0	0	1	1	1	D
C	1	0	1	0	1	0	C
C	1	0	1	1	1	0	C
D	1	1	0	0	0	0	A
D	1	1	0	1	1	1	D
D	1	1	1	0	1	0	C
D	1	1	1	1	1	1	D

solutions

Cet encodage nous indique qu'il faut 2 bascules D pour stocker l'état du système. Nommons par exemple ces deux bascules Q_1 et Q_0 .

Ecrivons la table qui décrit l'évolution des entrées des bascules en fonction des entrées et de l'état courant. Dans le tableau, on a rappelé les états symboliques.

partir de ce tableau, on cherche les équations de D_1 et D_0 . D_1 se trouve facilement, par une observation attentive du tableau : on peut voir une factorisation aisée :

$$D_1 = e_1 + e_2$$

Pour D_0 , passons par un tableau de Karnaugh.

Les regroupements amènent à :

$$D_0 = \overline{e_1} \cdot e_2 + e_2 \cdot Q_0 + \overline{e_1} \cdot \overline{Q_1} \cdot Q_0 + Q_1 \cdot \overline{Q_0} \cdot \overline{e_1}$$

Calculons les sorties, de la même manière. On trouve : $S_1 = Q_1 \cdot \overline{Q_0}$ et $S_2 = Q_1 \cdot Q_0$

On peut désormais dessiner le circuit complet :

