TD 3 – Électronique Numérique

Automates ou Machines d'états finis

Notes et Conseils

Remarque : Ce TD aborde plusieurs points relatifs aux *automates*, aussi appelés *machines d'états finis*[a] (*finite state machines*, FSM) :

- Diagrammes états-transitions : machines de Mealy et de Moore, et leur équivalence.
- Causalité des machines d'états finis.
- Conception d'un automate matériel à partir d'un diagramme états-transitions.

Les automates peuvent être vus comme l'outil conceptuel équivalent aux équations différentielles du monde analogique et continu : ces automates permettent de décrire l'évolution temporelle d'un système. Ils sont donc fondamentaux, non seulement pour la conception en Electronique, mais plus généralement pour toute l'Informatique.

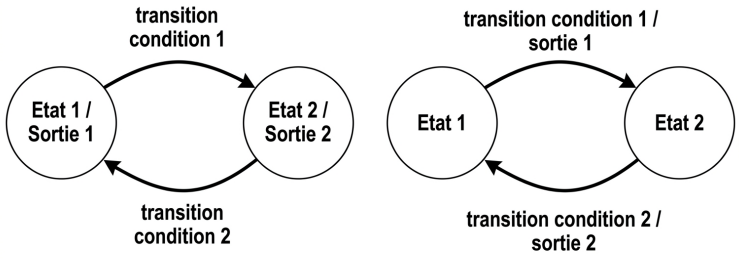
a. Ce nom quelque peu étrange signifie que le *nombre* d'états est *fini*. Il existe également des automates dont le nombre d'états est infini...

1 Diagramme états-transitions : Moore et Mealy

On rappelle que le diagramme *états-transitions* (ou *state transition diagram*, aussi appelé *diagramme à bulles*) est une représentation graphique abstraite du comportement d'un automate. Cette représentation consiste à décrire l'enchaînement des transitions d'un état à l'autre par une flèche, les états étant représentés par des cercles ou *bulles*.

On distingue deux grands types de FSM . Dans le cas des automates de Moore, les sorties ne dépendent que des états, alors que dans le cas des automates de Mealy, les sorties dépendent à la fois des états et des entrées.

Sur les transitions, on ajoute les *conditions* booléennes $c_i(s)$ qui autorisent à passer d'un état à un autre, ainsi que d'éventuelles sorties (généralement après une ou deux barres verticales) dans le cas des automates de Mealy. Lorsqu'on a affaire à un automate de Moore, les sorties sont annotées en conséquence dans les bulles (ou juste à côté).



(a) FSM de Moore et Mealy. Attention ! Les deux automates présentés ici ne sont ici pas équivalents en terme de comportement ! Voir exercice)

2 Causalité d'une FSM

Pour qu'une FSM soit considérée comme *causale*, correcte ou *consistante*, elle doit vérifier les deux propriétés suivantes :

— **Réactivité** (exhaustivité) : il doit exister une condition vraie sur au moins une des transitions sortant d'un état.

$$\forall s \in S : \sum_i c_i(s) = 1$$

— **Déterminisme** (exclusivité) : deux conditions sur les transitions sortant d'un état ne peuvent être vraies en même temps.

$$\forall s \in S, \forall (i,j) \text{ avec } i \neq j, \quad c_i(s) \cdot c_j(s) = 0$$

Une autre manière de résumer ces deux propriétés est qu'il existe – à tout instant discret – *un, et un seul, état suivant*. Notons qu'un état peut avoir comme état suivant lui-même : dans ce cas, le système décrit ne change pas d'état.

Questions

1. Observez la figure 1.Vérifier la consistance de la machine d'état suivante.
2. Est-ce un automate de Moore ou de Mealy ?

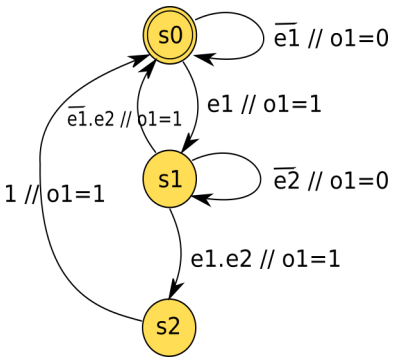


FIGURE 1 – FSM de l'exercice 1 et 2

3 Du diagramme états-transitions au circuit numérique.

Nous cherchons ici à réaliser l'électronique de l'automate de l'exercice précédent.

Notes et Conseils

La méthode manuelle consiste à décrire tout d'abord dans un tableau :

— L'évolution des états en fonction des entrées, en conservant leur nom *symbolique*.

— L'évolution des sorties en fonctions des états, et des entrées (cas de Mealy).

Après **choix d'un encodage binaire des états**, on connaît le nombre n de bascules D impliquées :

— On réécrit le tableau précédent pour chacune des bascules D_i.

— On déduit les équations $D_i = f(Q_0, Q_1, \ldots, Q_{n-1}, e_0, \ldots)$, où Q_j représente l'état courant de la bascule j et $\{e_k\}$ l'ensemble des entrées du circuit.

On rappelle qu'il existe un nombre infini de possibilité d'encodage des états. Cet encodage influe sur le nombre de bascules D nécessaires au stockage des états, mais également sur la

complexité de la logique combinatoire de la fonction de transitions et de la fonction de sortie (et donc de la rapidité du circuit).

On s'appuie sur les tableaux génériques suivants 1 et 2, qui permettent de procéder de manière systématique.

TABLE 1 – Encodage symbolique (nom des états préservés)

état courant	entrée 1	...	entrée n	état suivant	sortie 1	...	sortie n

TABLE 2 – Encodage binaire (états encodés –ici sur 2 bits)

Q1	Q0	entrée 1	...	entrée n	D1	D0	sortie 1	...	sortie n

Questions

— Etablir le tableau d'évolution des états symboliques et de la sortie.
— Choisir l'encodage *dense* le plus naturel.
— Etablir le tableau d'évolution des états encodés et de la sortie.
— Etablir les équations de chacun des bits d'états.
— Etablir les équations de chacun des bits de sortie.
— Dessiner le circuit à l'aide portes logiques élémentaires.

4 Automate *détecteur de séquence*

Il est fréquent de devoir détecter une séquence particulière dans un flux de données. Les automates se prêtent très bien à ce genre de détection. On peut citer :

— L'analyse de paquets réseaux, détection de paquets interdits (entrants ou sortants) dont on connaît une certaine signature numérique (travail de filtrage d'un proxy, etc.).
— L'analyse de flux multimédia : par exemple *start codes* de séquences, dans un film (stocké en numérique, dans un format de compression).
— L'analyse génomique : il existe des accélérateurs FPGA dédiés à la reconnaissance de séquence du génome, etc.

4.1 Détecteur de séquence `10` : Moore versus Mealy

Dans le cadre de la conception d'un sous-système de communication série, on souhaite réaliser un détecteur de séquence capable de repérer le motif binaire **"10"** sur un flux de données synchronisé par une horloge.

Le système possède :

— Une entrée binaire synchrone X (le bit courant),
— Une sortie binaire synchrone Z qui vaut **1** si et seulement si la séquence **"10"** vient d'être détectée sur l'entrée X, et **0** sinon.

On considère que les séquences peuvent se chevaucher (par exemple, dans la séquence `1 1 0`, le motif **"10"** est détecté sur les deux derniers bits).

1. **Automate de Mealy.** Concevoir un automate de type **Mealy** réalisant cette fonction. On donnera :

— la description des états et leur signification,
— la table des transitions (format : *état suivant / sortie*).

2. **Automate de Moore.** Concevoir un automate de type **Moore** ayant un comportement fonctionnel équivalent. On donnera :
   — la description des états, leur signification et la sortie associée,
   — la table des transitions.
3. **Comparaison temporelle.** Relever les valeurs de la sortie Z pour les deux automates sur la séquence d'entrée suivante, appliquée à chaque front montant de l'horloge :

   $$X = 0,\ 1,\ 0,\ 0,\ 1,\ 1,\ 0$$

   On supposera que les deux automates sont initialisés dans leur état de départ avant le premier front.
4. **Analyse.** Les deux automates sont-ils strictement équivalents ? Justifier en précisant la différence fondamentale entre les machines de Moore et de Mealy, et expliquer le phénomène observé à la question 3.

4.2 Détecteur de séquence `0110`

On considère le détecteur de séquence de la figure 2. Son rôle est de détecter les séquences *0-1-1-0* sur son unique entrée x. Lorsque cette séquence est détectée, il émet un *1* sur sa sortie y. Un chronogramme est fourni à titre d'exemple. Notez que dans le flux *0-1-1-0-1-1-0*, la séquence apparaît deux fois !

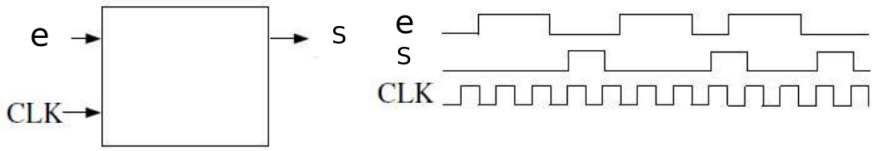


FIGURE 2 – Schéma de principe du détecteur de séquence `0110`. Chronogramme associé.

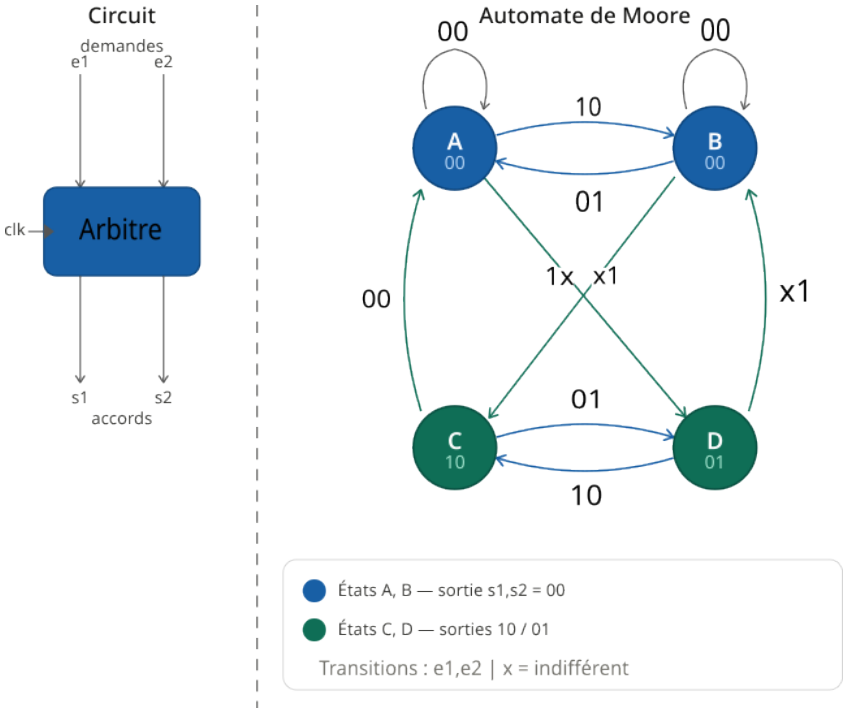
1. Proposer un diagramme états-transitions pour ce détecteur. On choisira une machine de Moore.
2. Proposer un encodage dense des états.
3. Réaliser le circuit.

5 Arbitre à priorité tournante (round-robin)

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 prioriatire.

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

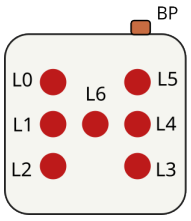
1. Proposer un encodage des états tel que chaque état soit codé sur 2 bits
2. Etablir la fonction de transition du système sous forme de table de vérité. Simplifier.
3. Etablir la fonction de sortie du système. Simplifier.
4. Dessiner le circuit correspondant. Combien de portes faut-il ?

6 Problème du dé électronique. Encodage des états spécifique.

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 sur 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



persistence rétinienne. Seule une illumination moyenne lui apparait.

On choisit d'utiliser un encodage des états particulier, *spécifique au problème* : la sortie de chaque bascule $Q_i, i \in 0..6$ pilote directement la led L_i correspondante. La fonction de sortie de l'automate est ainsi la *fonction identité*.

Question

1. Réaliser le tableau décrivant l'évolution de chaque bascule lorsque le dé est sollicité (bouton poussoir appuyé).
2. En déduire les équations constitutives du circuit correspondant.
3. Dessiner le circuit d'une ou deux de ces équations.
4. Rappeler succintement le schéma-blocs composé de 2 blocs : la logique d'état suivant et les bascules D (registres).
5. Choisir un autre encodage (par exemple one-hot) et réaliser à nouveau le circuit.

7 La magie de l'encodage "*one-hot*" ou "*un bit par état*"

Une machine distribue des canettes à 1€. Elle n'accepte que des pièces de 0,50€ et de 1€. Elle ne rend pas la monnaie.

On précise quelques éléments :
— **Entrées :** 'P50' (pièce de 50c) et 'P100' (pièce de 1€). Les deux entrées ne sont jamais actives en même temps.
— **Sorties :** 'DIST' (Distribuer la canette) et 'REND' (Rendre 0,50€ de monnaie - uniquement si le client a donné 1,50€).
— **Fonctionnement :**
  — Si somme totale atteint **1,00€** → 'DIST=1'.
  — Si somme totale atteint **1,50€** → 'DIST=1' ET 'REND=1' (car il a donné 1,50€ pour une canette à 1€).

Questions

1. Déssiner le diagramme états-transitions en conservant le nom des états symboliques.
2. Réaliser l'encodage *one-hot*.
3. Etablir les équations D_i de chacune des bascules mises en jeu.

8 Automate *serrure numérique*. Le problème des *spécifications*.

L'accès à un local est protégé par une serrure codée associée à un automatisme commandant la gâche électrique de la porte. La combinaison secrète est : **A, D, C**.

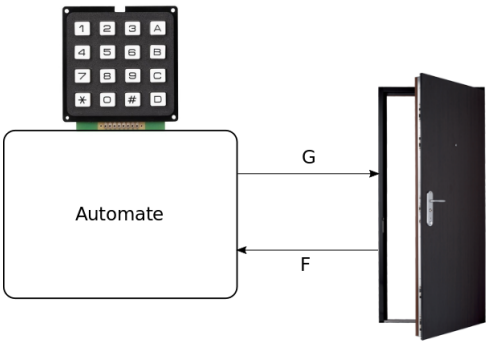


FIGURE 3 – Schéma de principe de la a serrure numérique.

Un capteur permet de connaître l'état de la porte : F indique son état. F vaut 1 si la porte est Fermée, et 0 si elle est ouverte.
G commande l'ouverture de la porte : si $G = 1$ la porte peut être ouverte. Lorsque la porte se referme, on a $G = 0$.
À l'initialisation, la porte est fermée, le contact de porte $F = 1$.
On précise qu'on ne peut appuyer que sur un seul bouton à la fois.

1. Proposer un automate de contrôle du dispositif.
2. Discuter de la robustesse de la solution, et de son réalisme. Proposer des alternatives.
3. Optionnel : construire le circuit complet.