Electronique numérique
Automates (1)

Tous les exercices ne seront pas forcément résolus en TE.

Ce TD aborde plusieurs points relatifs aux *automates*, aussi appelés *machines d'états finis*[1]*finite state machines* (FSM).

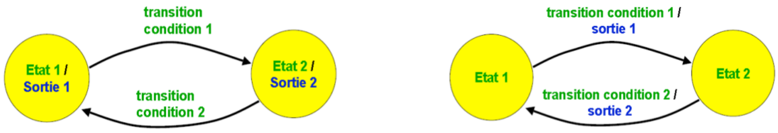
- Diagrammes états transitions : machines de Mealy et de Moore.
- Causalité des machines d'états finis.
- Conception d'un automate matériel à partir d'un diagramme états-transitions.

1 Diagramme états-transitions. Moore et Mealy

On rappelle que le diagramme à bulle (ou "state transition diagram" ou diagramme "états-transistions") est une représentation graphique pratique des automates. Elle 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 (voir figure 1)[2]. 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 !).

1. 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...
2. Attention ! Les deux automates représentés l'un à côté de l'autre ne sont pas équivalents en terme de fonctionnement. Il existe des algorithmes pour transformer Mealy en Moore (et réciproquement), de manière à obtenir des comportements identiques.... Voir par exemple `https://fr.wikipedia.org/wiki/Machine_de_Mealy`)



(a) label 1 (b) label 2

FIGURE 1 – FSM de Moore et Mealy

2 Causalité d'une FSM

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

— **Réactivité** (exhausitivité) : 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.

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

Une autre manière de résumer ces deux propriété 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.

Observez la figure 2.

— Vérifier la consistance de la machine d'état suivante.

— Est-ce un automate de Moore ou de Mealy ?

$\boxed{\textit{corrections}}$ Il y a plusieurs manières de procéder. Par exemple, on calcule les formules :

— exhausitivité :
  — Pour S0 : $\overline{e_1} + e_1 = ... = 1$
  — Pour S1 : $\overline{e_2} + \overline{e_1}.e_2 + e_1.e_2 = ... = 1$
  — Pour S2 : $1 = \cdots = 1$

— exclusivité :
  — Pour S0 : $\overline{e_1}.e_1 = ... = 0$
  — Pour S1 : $\overline{e_2}.\overline{e_1}.e_2 + \overline{e_2}.e_1.e_2 + \overline{e_1}.e_2.e_1.e_2 = 0 + 0 + 0 = 0$
  — Pour S2 : une seule transition. Il n'y a pas de problème d'exclusivité.

C'est un automate de Mealy, car les sorties dépendent des transitions, et non pas seulement des états.

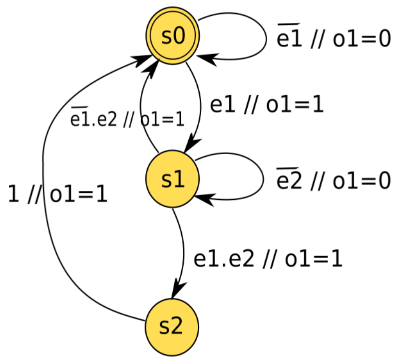


FIGURE 2 – FSM de l'exercice 1 et 2

2.1 Equations logiques de l'automate

On cherche les équations logiques de la fonction "état suivant", ainsi que de la fonction de sortie de l'automate précédent. Ces deux ensembles nous permettrons de réaliser concrètement le circuit. On suggère ici de procéder en 2 étapes. La première consiste à conserver le nom *symbolique* des états tandis que la seconde fait apparaître l'*encodage* de ces états. Dans la FSM à réaliser, en choisissant un encodage *one-hot*, trois bits seraient nécessaires pour encoder les états. On va se limiter toutefois ici à un encodage *dense* où 2 bits seuls sont nécessaires. On s'appuie sur les tableaux génériques suivant 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)

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

Solution

Puis encodons les états symboliques S0,S1 et S2 par 00,01,10 : Attention : ne pas chercher les équations directement à partir de ce tableau ! Il manque des combinaisons !

A partir de ce dernier tableau, on peut déterminer les équations de D1,D0 et o1 :

état courant	e1	e2	état suivant	o1
S0	0	X	S0	0
S0	1	X	S1	1
S1	X	0	S1	0
S1	0	1	S0	1
S1	1	1	S2	1
S2	X	X	S0	1

Q1	Q0	e1	e2	D1	D0	o1
0	0	0	X	0	0	0
0	0	1	X	0	1	1
0	1	X	0	0	1	0
0	1	0	1	0	0	1
0	1	1	1	1	0	1
1	0	X	X	0	0	1

$$D_1 = Q_0.e_1.e_2$$

$$D_0 = \overline{Q_1}.\overline{Q_0}.e1 + Q_0.\overline{e_2}$$

$$o_1 = \overline{Q_1}.\overline{Q_0}.e_1 + Q_0.e_2 + Q_1$$

Puis le circuit numérique correspondant...

3 Application

Soit le diagramme états-transitions de la figure 3. On suppose que les entrées A et B proviennent de deux capteurs. Les sorties sont S1,S2 et S3.

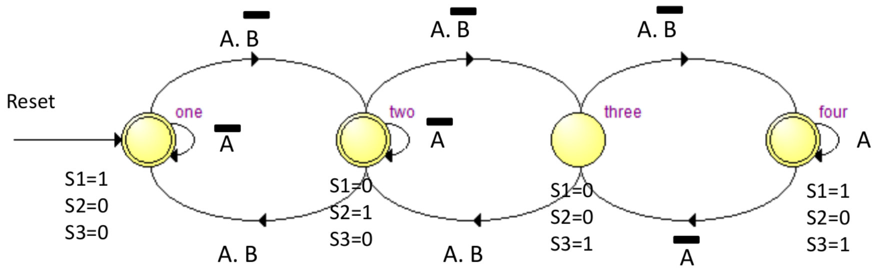


Figure 3 – Diagramme états-transitions (à modifier).

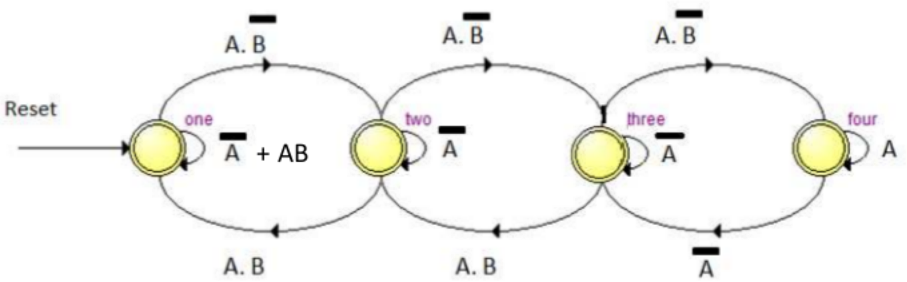
1. Cette machine est-elle causale ? Si non, proposer une modification afin de la rendre causale.

	/e1./e2	/e1.e2	e1.e2	e1./e2
/Q1./Q0	00	00	01	01
/Q1.Q0	01	00	10	01
Q1.Q0	xx	xx	xx	xx
Q1./Q0	00	00	00	00

2. Proposer un encodage "one-hot" pour cette FSM, et un encodage dense.
3. Etablir les équations de la fonction de transition. On rappelle que ces équations déterminent les entrées D des bascules d'état de l'automate.
4. Etablir les équations de sortie.
5. Dessiner la structure globale du circuit, en faisant apparaître, sous forme de nuage logique, les fonctions précédentes.
6. Tenter de dessiner le circuit complet.
7. Dessiner à part la connectique qui permet d'initaliser le système dans l'état

$\boxed{\textit{Solution}}$ C'est une solution proposée. Il en existe plusieurs.

1. La machine n'est pas causale, car les conditions sont "incomplètes". On peut proposer de compléter ces conditions de manière totalement "ad hoc", pour que la FSM devienne causale. **Attention !** : ce "patch" sauvage est dangereux ! Cela signifie probablement que l'ingénieur qui a écrit la FSM initiale n'a pas une formulation robuste de son problème ! Dans un cas réel, le "patch" doit être soigneusement discuté afin de vérifier qu'il respecte le comportement attendu du système visé.



2. Encodage one-hot :
3. Appelons Q_3,Q_2,Q_1,Q_0 les (sorties des) bascules qui stockent chacun des bits d'états dans le cas de l'encodage "one-hot". On conserve l'ordre indiqué dans les codes précédents : on a ainsi $(Q_3, Q_2, Q_1, Q_0 = (0, 0, 0, 1)$ pour l'état "one".

htp

Table 3 – Encodage one-hot

état	code
one	0001
two	0010
three	0100
four	1000

Table 4 – Encodage dense (exemple)

état	code
one	00
two	01
three	10
four	11

On trouve les équations de transitions, à la simple lecture du diagramme, du fait de l'encodage one-hot :

$$\begin{aligned} D_0 &= Q_0.(\overline{A} + A.B) + Q_1.A.B \\ D_1 &= Q_1.\overline{A} + Q_0.A.\overline{B} + Q_2.A.B \\ D_2 &= Q_2.\overline{A} + Q_1.A.\overline{B} + Q_3.\overline{A} \\ D_3 &= Q_3.A + Q_2.A.\overline{B} \end{aligned}$$

Les équations des sorties sont :

$$\begin{aligned} S_1 &= Q_0 + Q_3 \\ S_2 &= Q_1 \\ S_3 &= Q_2 + Q_3 \end{aligned}$$

Le circuit ressemble à la figure 4.

Le dessin à la main d'un tel circuit est très fastidieux...Il faut passer par des logiciels de synthèse automatique pour obtenir un tel schéma.

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 connait une certaine signature numérique. C'est le travail de filtrage d'un "proxy", etc
- L'analyse de flux multimedia : par exemple "start codes" de séquences, dans un film (stocké en numérique, dans un format de compression).

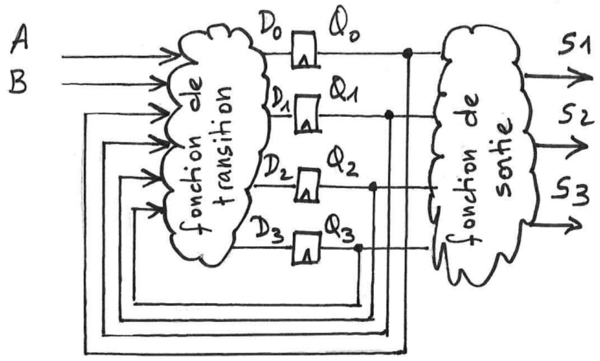


FIGURE 4 – Schéma du circuit présentant les fonctions "état suivant" et "sortie" sous forme de nuage combinatoire.

— L'analyse génomique : il existe des accélérateurs FPGA dédiés à la reconnaisance de séquence du génôme, etc.
— etc.

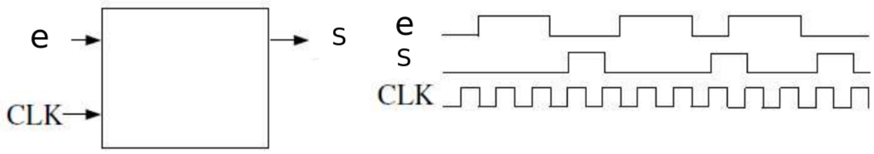


FIGURE 5 – Schéma de principe du détecteur de séquence. Chronogramme associé.

On considère le détecteur de séquence de la figure 7. 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 !

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.

Solution

L'automate solution est décrit sur la figure 6. On choisit l'encodage dense suivant :

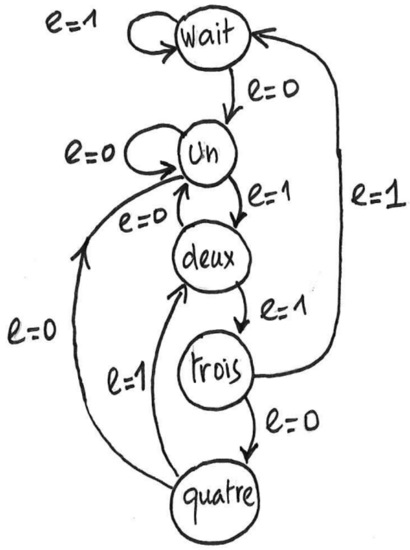


FIGURE 6 – Diagramme états-transitions pour le détecteur de séquences.

état	code
wait	000
un	001
deux	010
trois	011
quatre	100

On a alors la table de séquencement suivante :

Q_2	Q_1	Q_0	e	D_2	D_1	D_0	s
0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	1	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0
0	1	0	1	0	1	1	0
0	1	1	0	1	0	0	1
0	1	1	1	0	0	0	0
1	0	0	0	0	0	1	0
1	0	0	1	0	1	0	0

Cherchons maintenant les équations, en passant par des tableaux de Karnaugh.

Q_2Q_1

Q_0E	00	01	11	10
00	1	1	X	1
01	0	1	X	0
11	0	0	X	X
10	1	0	X	X

On obtient la formule $\boxed{D_0 = Q_1.\overline{Q_0} + \overline{Q_1}.\overline{E}}$

Q_2Q_1

Q_0E	00	01	11	10
00	0	0	X	0
01	0	1	X	1
11	1	0	X	X
10	1	0	X	X

On obtient la formule $\boxed{D_1 = Q_1.\overline{Q_0}.E + \overline{Q_1}.Q_0.E + Q_2.E}$.

Q_2Q_1

Q_0E	00	01	11	10
00	0	0	X	0
01	0	0	X	0
11	0	0	X	X
10	0	1	X	X

On obtient la formule $\boxed{D_2 = Q_1.Q_0.\overline{E}}$

Q_0E \ Q_2Q_1	00	01	11	10
00	0	0	X	1
01	0	0	X	1
11	0	0	X	X
10	0	0	X	X

On obtient la formule $\boxed{D_3 = Q_2}$

5 Automate "serrure numérique"

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.

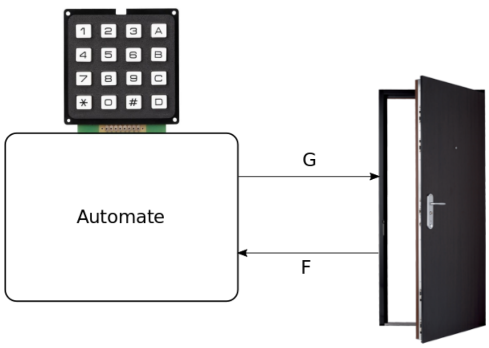


FIGURE 7 – Schéma de principe de la serrure numérique.

Un capteur permet de connaître l'état de la porte : F indique son état. F vaut à 1 si la porte est Fermée à 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. A 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 dispostif.

2. Discuter de la robustesse de la solution, et de son réalisme. Proposer des alternatives.
3. Optionel : construire le circuit complet.

$\boxed{\textit{Solution}}$ Dans ce type de dispositifs, les *spécifications* du dispostif sont ambigües et sujettes à interprétations. Cela est une constante dans ce type de dispostifs !

— Par exemple, est-il réaliste de ne pas prendre en compte un temps limite d'attente entre deux actions d'appui sur un bouton ? On pourrait donc, dans un état donné, déclancher un *autre automate* dédié au décomptage du temps (timer).
— Ce timer est-il à déclancher *entre* deux appuis, ou dès qu'on commence la séquence ?
— De même, le sujet stipule que la porte "peut être ouverte" : cela ne signifie pas que le retrait de la gâche entraîne forcément l'ouverture de la porte : on devra donc là aussi attendre une action de l'utilisateur, probablement sur la poignée de la porte.
— S'il y a une poignée d'un côté de la porte, peut-on supposer qu'un *autre* utilisateur puisse ouvrir la porte, de manière inopinée pendant la tentative du premier utilisateur ? Etc...

Nous proposons ici une solution à peu près réaliste : dès que la lettre A est correctement entrée, un timer se déclanche. Si la porte est par ailleurs ouverte (de l'intérieur), le dispositif se remet à l'état d'attente initial.

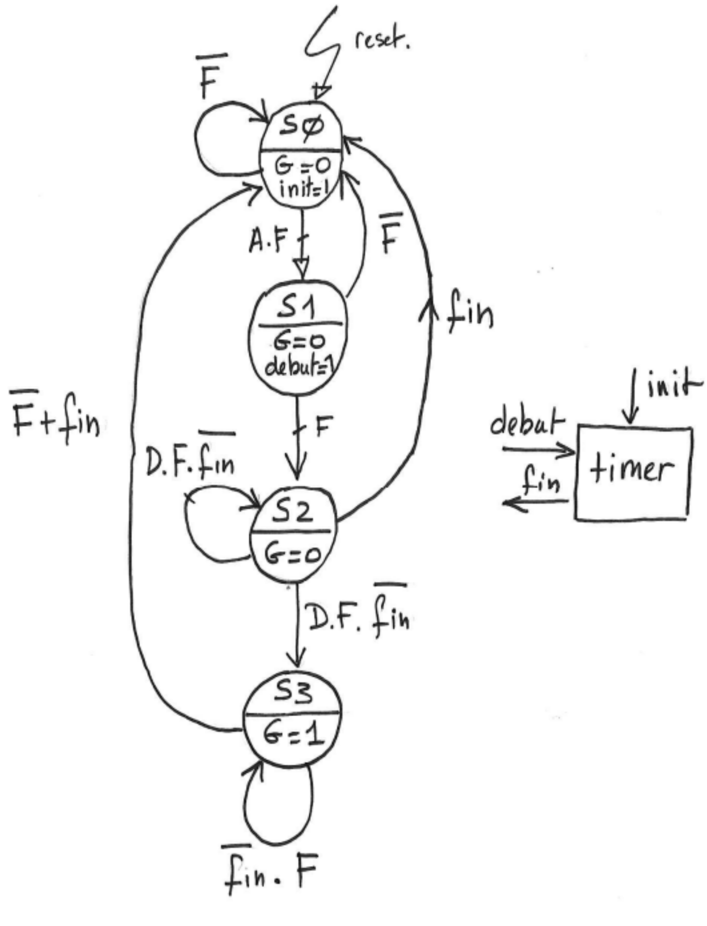


Figure 8 – Diagramme état-transition pour la serrure numérique. On fait appel à un autre dispositif numérique : un timer.