

Conception d'une machine à état synchrone

SIN1 - Cours 3 - Partie 3

J. Villemejjane - julien.villemejjane@u-pec.fr

IUT Créteil-Vitry
Département GEII
Université Paris-Est Créteil

Année universitaire 2013-2014

Plan du cours

- 1 Machine de MOORE
 - Définition
 - Graphe d'états

- 2 Machine de MEALY
 - Définition
 - Graphe d'états

- 3 Conception et synthèse d'une machine à état
 - Graphe d'états / Codage des états
 - Table des transitions
 - Synthèse avec des bascules D
 - Description VHDL

Conception d'une machine à état synchrone

Les **machines à état** permettent de décrire des systèmes séquentiels dont l'évolution est plus complexe que les compteurs ou les registres.

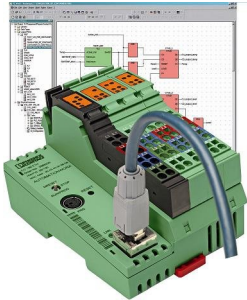
Il est remarquable de constater que le concept relatif aux automates (au sens machines à état) se retrouve désormais dans des applications diverses :

- circuits numériques
- automatismes industriels
- processeurs ou microcontrôleurs
- programmes informatiques

Pour représenter ces automates, qu'ils soient matériel ou logiciel, il existe deux architectures différentes : la **machine de MOORE** et la **machine de MEALY**.

Conception d'une machine à état synchrone

Les **machines à état** permettent de décrire des systèmes séquentiels dont l'évolution est plus complexe que les compteurs ou les registres.



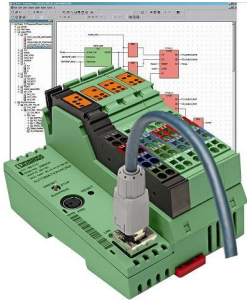
Il est remarquable de constater que le concept relatif aux automates (au sens machines à état) se retrouve désormais dans des applications diverses :

- circuits numériques
- automatismes industriels
- processeurs ou microcontrôleurs
- programmes informatiques

Pour représenter ces automates, qu'ils soient matériel ou logiciel, il existe deux architectures différentes : la **machine de MOORE** et la **machine de MEALY**.

Conception d'une machine à état synchrone

Les **machines à état** permettent de décrire des systèmes séquentiels dont l'évolution est plus complexe que les compteurs ou les registres.



Il est remarquable de constater que le concept relatif aux automates (au sens machines à état) se retrouve désormais dans des applications diverses :

- circuits numériques
- automatismes industriels
- processeurs ou microcontrôleurs
- programmes informatiques

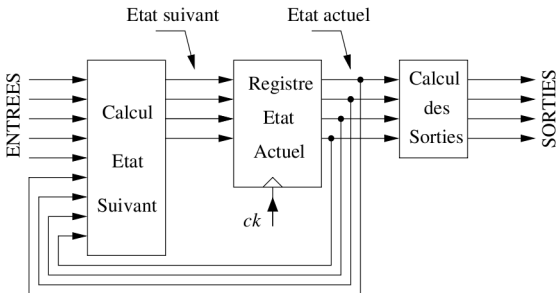
Pour représenter ces automates, qu'ils soient matériel ou logiciel, il existe deux architectures différentes : la **machine de MOORE** et la **machine de MEALY**.

Machine de MOORE

Définition

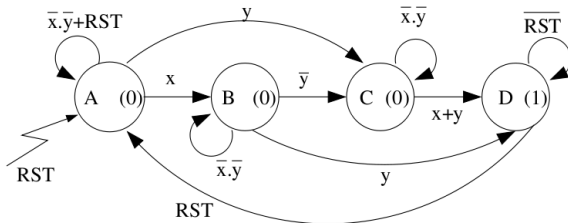
La machine de Moore - synchrone

- La **sortie** ne dépend que de l'état de la machine
- Les **sorties** sont **synchrone**s avec les transitions d'état et les fronts d'horloge



Machine de MOORE

Graphe d'états



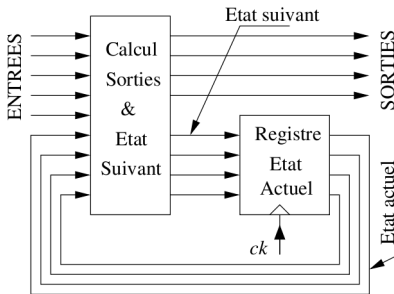
Dans une telle machine, les sorties étant fonction exclusivement de l'état, leurs valeurs sont indiquées dans les cercles.

Machine de MEALY

Définition

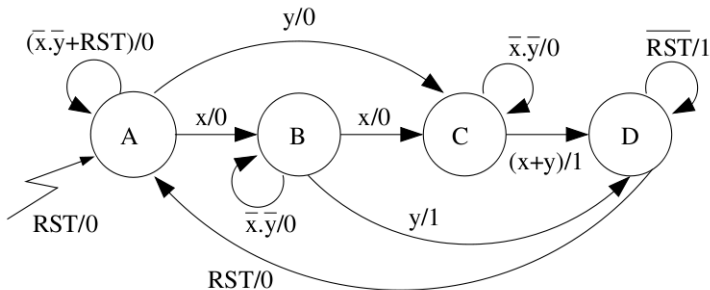
La machine de Mealy - asynchrone

- La sortie est calculée en fonction de l'état présent et de la valeur présente des entrées
- Les **sorties** sont **indépendantes** de l'horloge



Machine de MEALY

Grphe d'états



Dans une telle machine, les sorties étant fonction de l'état actuel et des entrées, leurs valeurs sont portées à côté des conditions (séparées par une barre oblique).

Conception et synthèse d'une machine à état

Détecteur de séquence

Le système à réaliser a une entrée E et une sortie S .

E reçoit des bits en série, cadencés par une horloge.

Chaque fois qu'une séquence 010 se présente en entrée, la sortie S devra passer à 1 dès le dernier bit détecté, puis retourner à 0 au bit suivant, quel qu'il soit.

Conception et synthèse d'une machine à état

Détecteur de séquence

Le système à réaliser a une entrée E et une sortie S .

E reçoit des bits en série, cadencés par une horloge.

Chaque fois qu'une séquence 010 se présente en entrée, la sortie S devra passer à 1 dès le dernier bit détecté, puis retourner à 0 au bit suivant, quel qu'il soit.



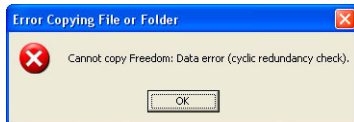
Conception et synthèse d'une machine à état

Détecteur de séquence

Le système à réaliser a une entrée E et une sortie S .

E reçoit des bits en série, cadencés par une horloge.

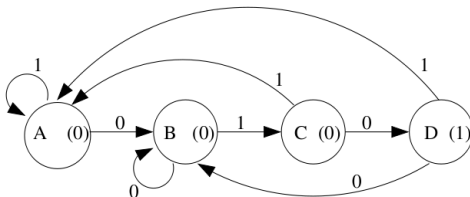
Chaque fois qu'une séquence 010 se présente en entrée, la sortie S devra passer à 1 dès le dernier bit détecté, puis retourner à 0 au bit suivant, quel qu'il soit.



Conception et synthèse d'une machine à état

Graphe d'états / Codage des états

Graphe d'états basé sur une machine de Moore.



Codage des états

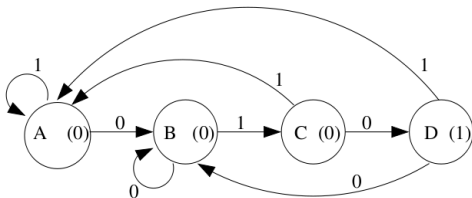
4 états nécessitent 2 bits d'état (donc 2 bascules).

On pourra utiliser un codage simple des états, par exemple le code de Gray : A=00, B=01, C=11 et D=10.

Conception et synthèse d'une machine à état

Graphe d'états / Codage des états

Graphe d'états basé sur une machine de Moore.



Codage des états

4 états nécessitent 2 bits d'état (donc 2 bascules).

On pourra utiliser un codage simple des états, par exemple le code de Gray : A=00, B=01, C=11 et D=10.

Conception et synthèse d'une machine à état

Table des transitions

Sous forme littérale

Actuel	Suivant		Sortie S
	0	1	
A	B	A	0
B	B	C	0
C	D	A	0
D	B	A	1

Sous forme codée

Q_1 et Q_2 représentent les sorties des bascules d'état, D_1 et D_2 les entrées correspondantes.

Conception et synthèse d'une machine à état

Table des transitions

Sous forme littérale

Actuel	Suivant		Sortie S
	0	1	
A	B	A	0
B	B	C	0
C	D	A	0
D	B	A	1

Sous forme codée

Actuel	Suivant		Sortie S
	0	1	
00	01	00	0
01	01	11	0
11	10	00	0
10	01	00	1

Q_1 et Q_2 représentent les sorties des bascules d'état, D_1 et D_2 les entrées correspondantes.

Conception et synthèse d'une machine à état

Synthèse avec des bascules D

Dans le cas de l'utilisation de bascules D, les valeurs des entrées D des bascules sont directement donnés par les codes de l'état suivant.

On en déduit alors (après simplification) les expressions de D_1 et de D_2 en fonction de Q_1 , Q_2 et E ainsi que l'expression de S .

$$D_1 = \overline{E} \cdot Q_1 \cdot Q_2 + E \cdot Q_1 \cdot \overline{Q_2}$$

$$D_2 = \overline{E} \cdot \overline{Q_1} + \overline{E} \cdot Q_2 + Q_1 \cdot \overline{Q_2}$$

$$S = Q_1 \cdot \overline{Q_2}$$

Conception et synthèse d'une machine à état

Description VHDL

```
library IEEE;
use IEEE.std_logic_1164.ALL;

entity detect_seq is
  port
  (
    E, CLK: in STD_LOGIC;
    s:      out STD_LOGIC
  );
end detect_seq;

architecture mach_etat of detect_seq is
  signal ETAT: STD_LOGIC_VECTOR(1 downto 0);

mach: process (CLK)
begin
  if (CLK'event and CLK='1') then
    case ETAT is
      when "00" =>
        if E='0' then ETAT <= "01";
        else ETAT <= "00";
        end if;
    end case;
  end if;
end process;
end mach_etat;
```

Conception et synthèse d'une machine à état

Description VHDL

```
when "01" =>
  if E='0' then ETAT <= "01";
  else ETAT <= "11";
  end if;
when "10" =>
  if E='0' then ETAT <= "01";
  else ETAT <= "00";
  end if;
when others =>
  if E='0' then ETAT <= "10";
  else ETAT <= "00";
  end if;
end case;
end if;
end process mach;

S <= '1' when ETAT = "10" else '0';

end mach_etat;
```