



# Machines d'états

## Exercices Conception numérique

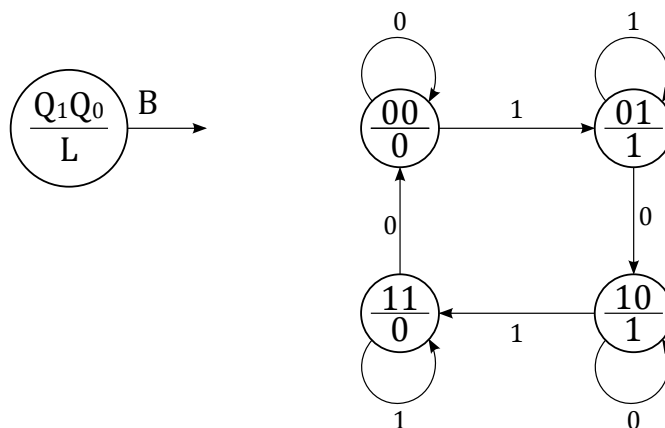


### Solution vs. Hints:

Toutes les réponses fournies ici ne sont pas des solutions complètes. Certaines ne sont que des indices pour vous aider à trouver la solution vous-même. Dans d'autres cas, seule une partie de la solution est fournie.

## 1 | FSM - Machines de Moore

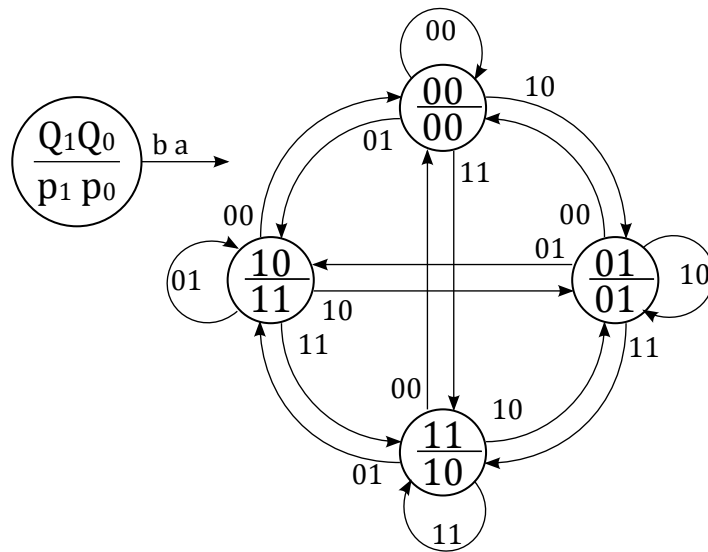
### 1.1 Graphe d'une machine d'états



*fsm/moore-01*



## 1.2 Graphe d'une machine d'états



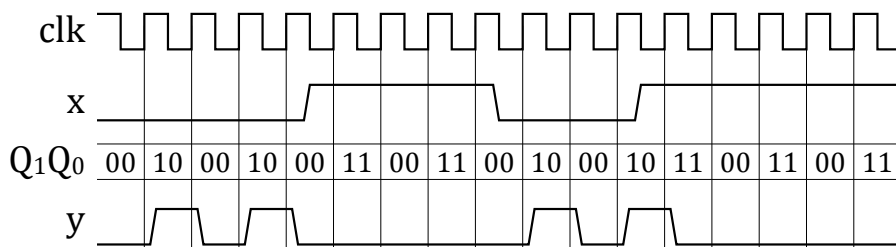
fsm/moore-02

## 1.3 Séquence d'un compteur

...  $\Rightarrow 0 \Rightarrow 1 \Rightarrow 3 \Rightarrow 2 \Rightarrow 6 \Rightarrow 7 \Rightarrow 5 \Rightarrow 4 \Rightarrow 0 \Rightarrow \dots$  (1)

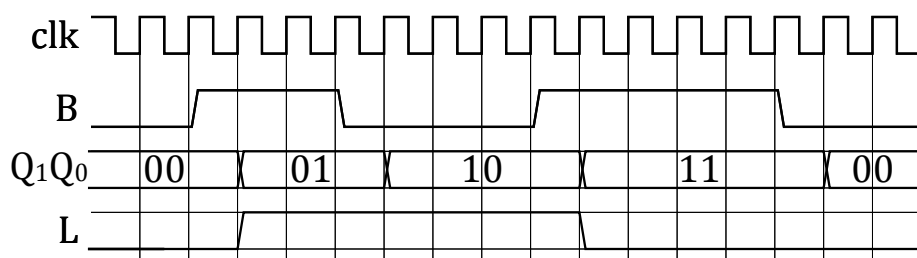
fsm/moore-03

## 1.4 Comportement temporel d'une machine d'états



fsm/moore-04

## 1.5 Comportement temporel d'une machine d'états

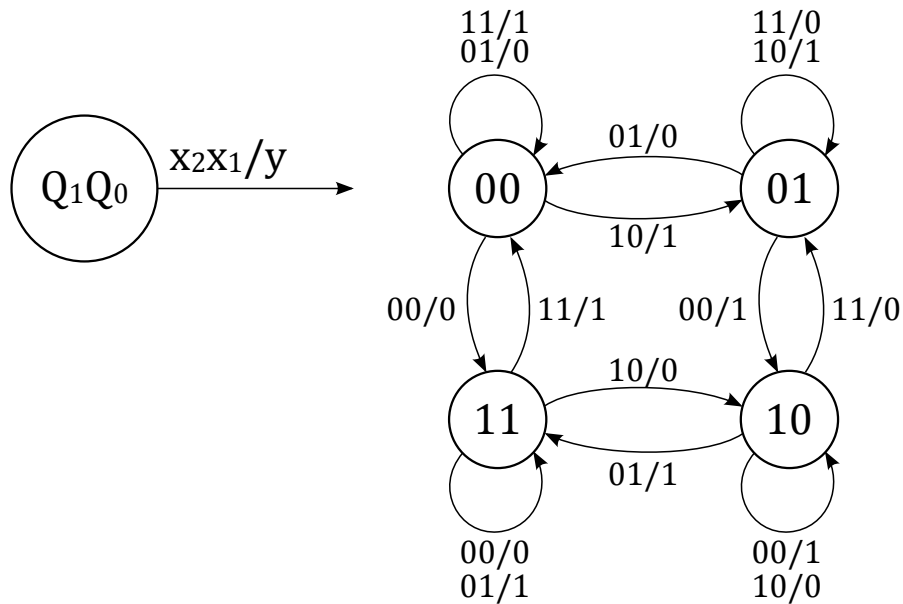


fsm/moore-05



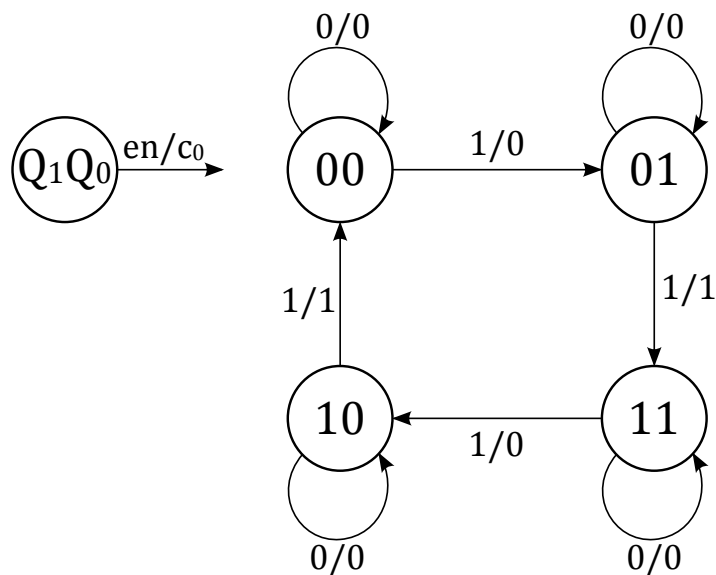
## 2 | FSM - Machines de Mealy

### 2.1 Graphe d'une machine d'états



*fsm/mealy-01*

### 2.2 Graphe d'une machine d'états



*fsm/mealy-02*



## 2.3 Comportement temporel d'une machine d'états

### 2.3.0.1 Initial State

$$x = 0 \Rightarrow Q = "00"$$

### 2.3.0.2 Outputs

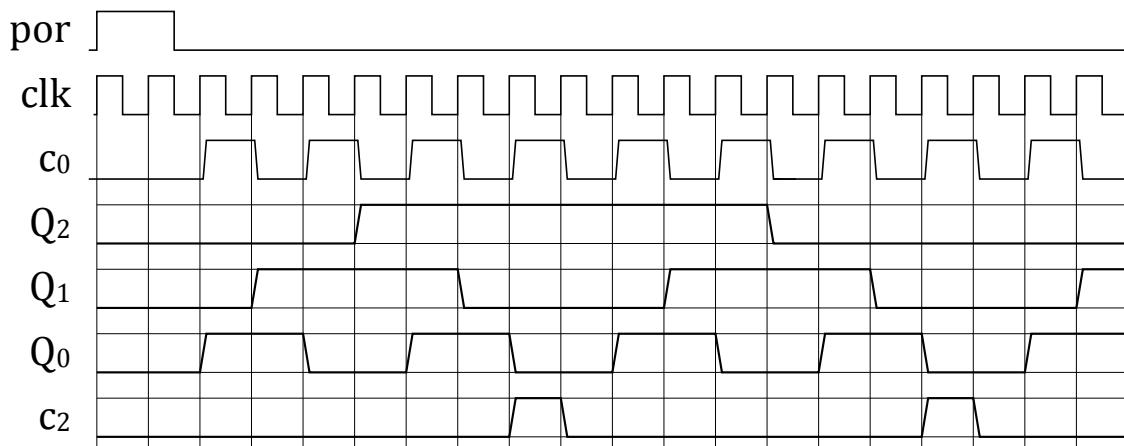
$$y_1 = 1 \Rightarrow \begin{cases} Q = "10" & \& x = 1 \\ Q = "11" & \& x = 1 \mid x = 0 \end{cases} \quad (2)$$

$$y_0 = 1 \Rightarrow \begin{cases} Q = "01" & \& x = 1 \\ Q = "11" & \\ Q = "10" & \& x = 0 \end{cases}$$

*fsm/mealy-03*

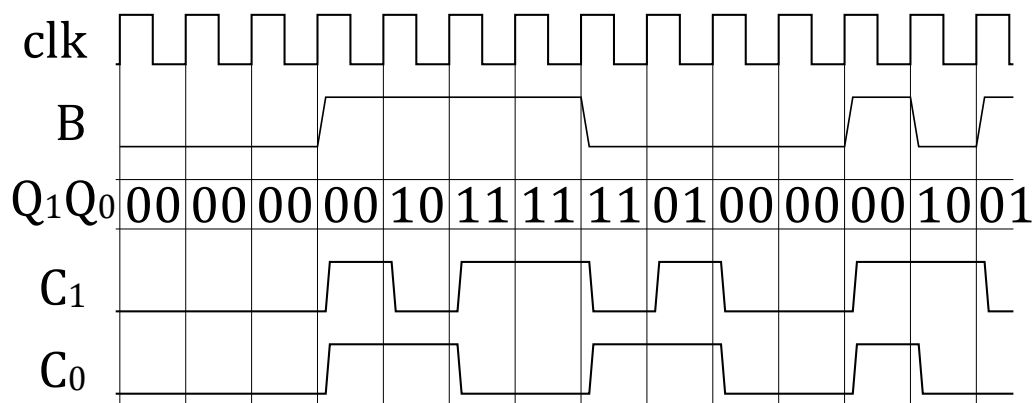
## 2.4 Compteur itératif

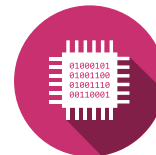
Mealy-Machine since  $c_2$  depends on  $c_0$  &  $Q_0$  &  $Q_1$ .



*fsm/mealy-04*

## 2.5 Comportement temporel d'une machine d'états



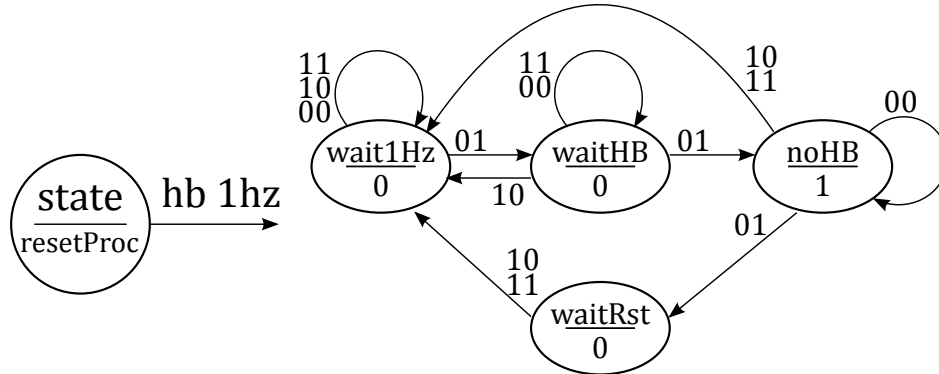


*fsm/mealy-05*



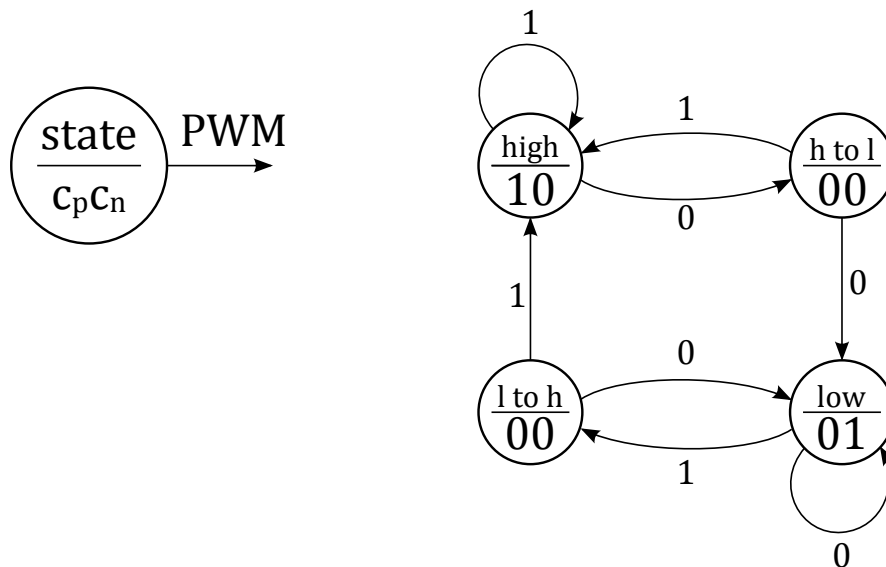
### 3 | FSM - Établissement du graphe des états

#### 3.1 Superviseur de fonctionnement



fsm/fsm-01

#### 3.2 Générateur de commandes non-recouvrantes



fsm/fsm-02

#### 3.3 Commande de distributeur automatique

FSM-Type = Moore. There is no realtime action needed

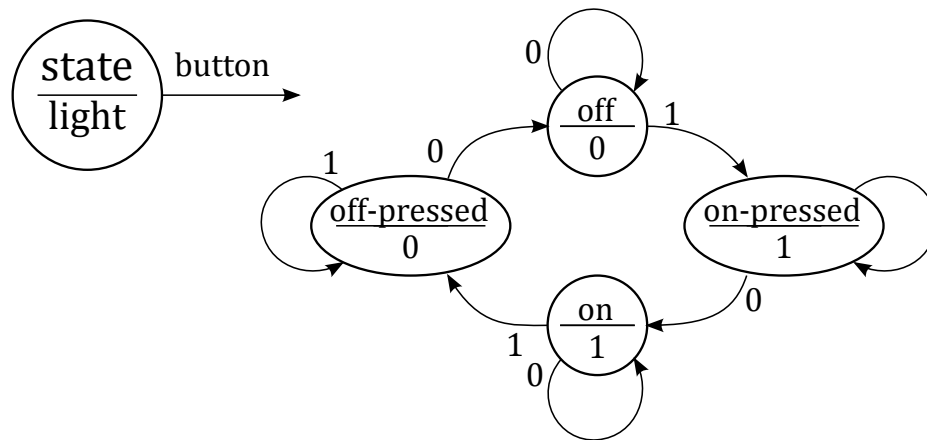
$c_1c_2 = "11" \Rightarrow$  impossible

fsm/fsm-03



### 3.4 Commande de lumières

FSM Type = Moore. There is no realtime action needed.

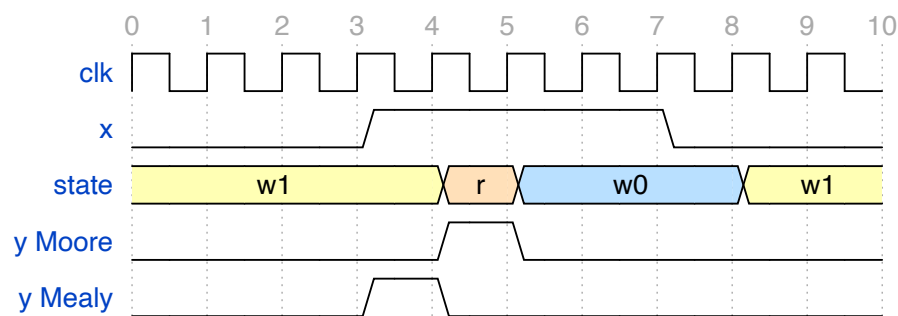


*fsm/fsm-04*

### 3.5 Détecteur de flanc montant

FSM Type = Moore and Mealy possible.

#### 3.5.0.1 Timing Diagram



#### 3.5.0.2 Grap

Moore FSM can be done with 3 states. Mealy FSM can be done with 2 states.

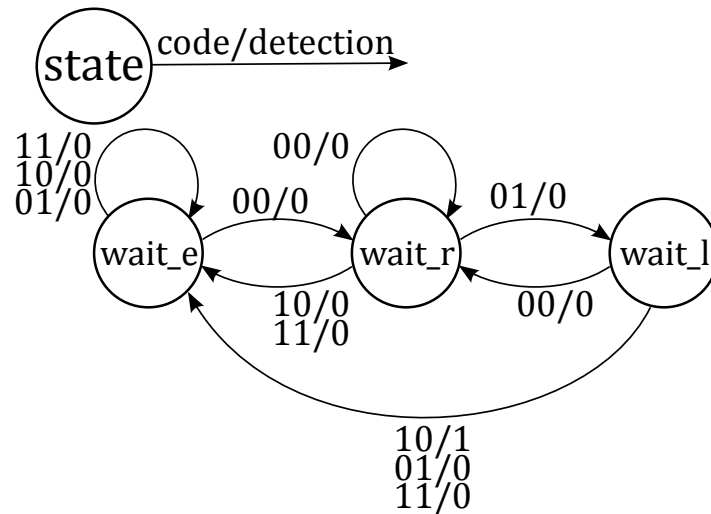
*fsm/fsm-05*



### 3.6 Détection de chaîne de caractères

FSM-Type = Mealy since an immediate response is needed.

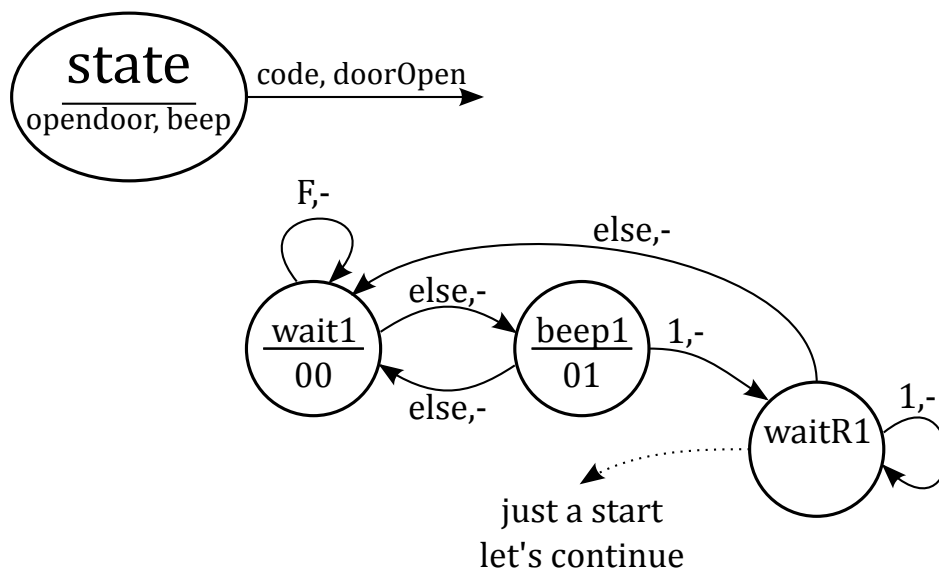
#### 3.6.0.1 Graph



fsm/fsm-06

### 3.7 Serrure électronique

FSM-Type = Moore. The output signal is during one clock period.



fsm/fsm-07





## 4 | FSM - Réduction de graphes

### 4.1 Réduction de graphe

#### 4.1.0.1 Truth Table

state \ x	0	1
st0	st0,0	st1,0
st1	st3,0	st2,0
st2	st3,0	st4,1
st3	st0,0	st1,0
st4	st5,1	st7,1
st5	st6,1	st7,1
st6	st0,0	st7,1
st7	st5,1	st4,1

The **blue** and **green** states can be combined to new states e.g. **st03** and **st47**. Draw also the new graph.

*fsm/reduction-01*

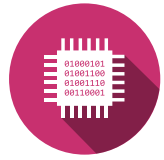
### 4.2 Réduction de graphe

#### 4.2.0.1 Truth Table

state \ $x_1x_2$	00	01	10	11
st0	st0,0	st2,0	st1,0	st0,0
st1	st1,0	st2,0	st1,0	st3,0
st2	st2,0	st2,0	st1,0	st3,0
st3	st5,1	st4,1	st3,0	st3,0
st4	st4,1	st4,1	st0,0	st3,0
st5	st5,1	st5,1	st0,0	st3,0

The **blue** and **green** states can be combined to new states e.g. **st12** and **st45**. Draw also the new graph.

*fsm/reduction-02*



## 5 | FSM - Codage des états

### 5.1 Circuit logique

$$\begin{aligned}
 Q_2^+ &= D_2 = \overline{x \oplus Q_2} \\
 Q_1^+ &= D_1 = \overline{x} \overline{Q_2} \overline{Q_1} Q_0 + \overline{x} Q_2 Q_1 Q_0 + x \overline{Q_2} Q_1 \overline{Q_0} + x Q_2 \overline{Q_1} \overline{Q_0} \\
 Q_0^+ &= D_0 = \overline{x} \overline{Q_2} \overline{Q_1} \overline{Q_0} + \overline{x} Q_2 \overline{Q_1} Q_0 + x \overline{Q_2} \overline{Q_1} Q_0 + x Q_2 Q_1 \overline{Q_0} \\
 y_1 &= Q_2 \\
 y_2 &= Q_2 \overline{Q_1} \overline{Q_0}
 \end{aligned} \tag{3}$$

*fsm/coding-01*

### 5.2 Circuit logique

$$\begin{aligned}
 Q_1^+ &= x(Q_1 + Q_0) \\
 Q_0^+ &= xQ_1 + x\overline{Q_0} \\
 y_1 &= Q_1 Q_0 + xQ_1 \\
 y_0 &= \overline{x}Q_1 + xQ_0
 \end{aligned} \tag{4}$$

*fsm/coding-02*

### 5.3 Circuit logique

One-Hot Encoding Scheme was used.

$$\text{all arrows to a state} \left\{ \begin{aligned} D_0 &= Q_0 \overline{\text{step}} + Q_7 \text{step} \text{ cw} + Q_1 \text{step} \overline{\text{cw}} \\ D_1 &= Q_1 \overline{\text{step}} + Q_0 \text{step} \text{ cw} + Q_2 \text{step} \overline{\text{cw}} \\ D_2 &= Q_2 \overline{\text{step}} + Q_1 \text{step} \text{ cw} + Q_3 \text{step} \overline{\text{cw}} \\ D_3 &= Q_3 \overline{\text{step}} + Q_2 \text{step} \text{ cw} + Q_4 \text{step} \overline{\text{cw}} \\ D_4 &= Q_4 \overline{\text{step}} + Q_3 \text{step} \text{ cw} + Q_5 \text{step} \overline{\text{cw}} \\ D_5 &= Q_5 \overline{\text{step}} + Q_4 \text{step} \text{ cw} + Q_6 \text{step} \overline{\text{cw}} \\ D_6 &= Q_6 \overline{\text{step}} + Q_5 \text{step} \text{ cw} + Q_7 \text{step} \overline{\text{cw}} \\ D_7 &= Q_7 \overline{\text{step}} + Q_6 \text{step} \text{ cw} + Q_0 \text{step} \overline{\text{cw}} \end{aligned} \right. \tag{5}$$

$$\text{states were the output is set} \left\{ \begin{aligned} c_1 &= Q_0 + Q_1 + Q_7 \\ c_2 &= Q_1 + Q_2 + Q_3 \\ c_3 &= Q_3 + Q_4 + Q_5 \\ c_4 &= Q_5 + Q_6 + Q_7 \end{aligned} \right.$$

*fsm/coding-03*



## 5.4 Circuit logique

### Additional signal

The states  $Q_1$  and  $Q_0$  can distinguish 4 different clock periods. But the signal as 8 clockperiods repeating as a mirror.

⇒ An additional signal is needed, to differentiate.

#### 5.4.0.1 Truth table

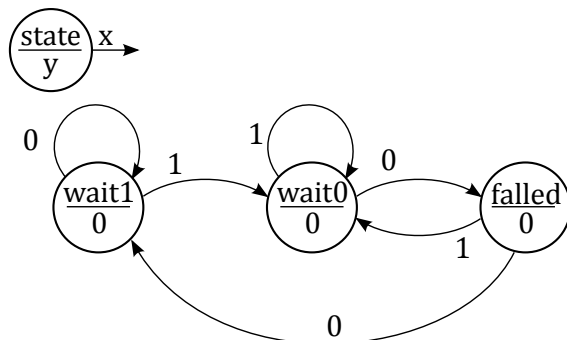
$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$c_1$
0	0	0	0	0	1	0
0	0	1	0	1	1	0
0	1	0	1	1	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	1
1	0	1	1	0	0	0
1	1	0	1	1	1	0
1	1	1	1	0	1	0

#### 5.4.0.2 Equations

$$\begin{aligned}
 D_2 &= Q_0 Q_2 + \overline{Q_0} Q_1 \\
 D_1 &= Q_0 \overline{Q_2} + \overline{Q_0} Q_1 \\
 D_0 &= Q_1 \oplus \overline{Q_2} \\
 c_1 &= Q_2 \overline{Q_1} \overline{Q_0}
 \end{aligned}
 \quad (6)$$

fsm/coding-04

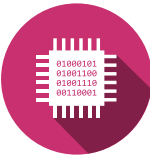
## 5.5 Détecteur de flanc descendant



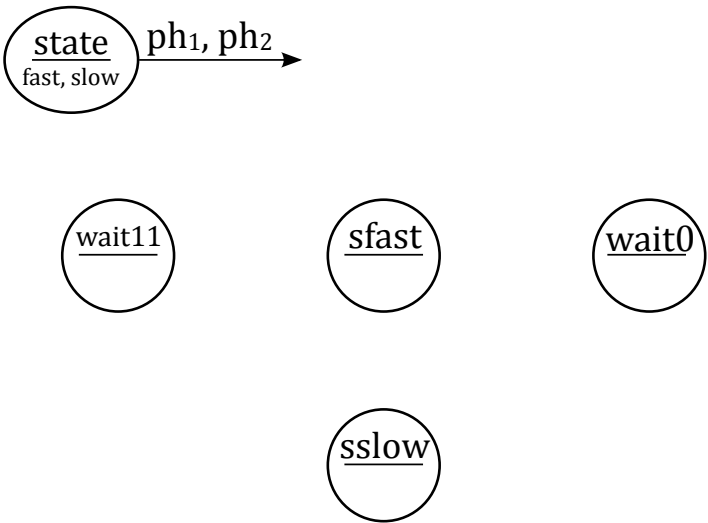
state	Q Encoding
wait1	10
wait0	00
failed	11

Next steps is to create the truth table and the Equations in order to draw the circuit.

fsm/coding-05



## 5.6 Détecteur de phase



### 5.6.0.1 State encoding (One-Hot)

state	Q Encoding
wait11	0001
sfast	0010
sslow	0100
wait0	1000

### 5.6.0.2 Equations

$$\begin{aligned} D_0 &= ph_1 ph_2 \\ D_1 &= (Q_0 + Q_1) \overline{ph_1} ph_2 \\ D_2 &= (Q_0 + Q_2) ph_1 \overline{ph_2} \\ D_3 &= \overline{ph_1} \overline{ph_2} + (Q_1) ph_1 \overline{ph_2} + (Q_2) \overline{ph_1} ph_2 + Q_3 (ph_1 \oplus ph_2) \\ fast &= Q_1 \\ slow &= Q_2 \end{aligned}$$

*fsm/coding-06*