INF1500 : Logique des systèmes numériques

- Cours 9: Machines à états Moore et Mealy

Procédure d'analyse d'un circuit séquentiel

- La procédure pour analyser un circuit séquentiel synchrone à partir d'un diagramme donné consiste à :
  - Identifier les variables d'état;
  - Écrire les équations d'état et les équations de sortie;
  - Dresser le tableau d'états; et,
  - Dessiner le diagramme d'état.

Variables et équations d'état

- L'état d'un circuit séquentiel est la valeur de tous ses éléments à mémoire à un moment donné.

- Dans un circuit séquentiel, chaque signal de sortie d'un élément à mémoire est une variable d'état du circuit.

- Les équations d'état d'un circuit séquentiel déterminent la valeur des variables d'état du circuit en fonction de leurs valeurs présentes ainsi que des entrées du système. Les équations d'état sont aussi appelées équations de transition.

Machines d'états finis (Finite State Machine)

- Modèle de base pour décrire le comportement d'un système :
  - Selon son état actuel, selon les évènements extérieurs,
    - envoie des messages à l'extérieur
    - change d'état
- Exemples:
  - Distributeur automatique de monnaie
  - Feu de circulation pour piéton
  - La partie contrôle d'un processeur
  - ...

Tableau d'état

- Un tableau d'état (aussi appelé tableau de transitions d'état) est similaire à une table de vérité. Il comporte quatre sections : les états présents, les entrées, les états prochains, et les sorties.

- Si on a *m* bascules et *n* entrées, le tableau a 2^{m+n} rangées en forme générale.

- En forme compacte, le tableau n'a que 2^{m} rangées. On forme alors des colonnes pour couvrir les différents cas des variables d'entrée.

Diagramme d'état

- **Toute l'information présente dans un tableau d'état peut être représentée sous forme graphique par un diagramme d'état, et vice versa. Un diagramme d'état ne contient pas plus d'information qu'un tableau d'état mais facilite la compréhension du comportement du circuit. Dans un diagramme d'état :**
  - **Les états sont identifiés par des cercles, avec leur nom et/ou leur code binaire associé écrit dans le cercle;**
  - **Les transitions entre les états sont identifiées par des flèches entre les cercles;**
  - **Les conditions pour toute transition (i.e. les valeurs nécessaires de variables d'entrée) sont placées à côté des flèches de transition;**
  - **Pour les machines de Moore (i.e. si les sorties ne dépendent que de l'état présent), la valeur des signaux de sortie est placée à l'intérieur des cercles; et,**
  - **Pour les machines de Mealy (i.e. si les sorties dépendent de l'état présent et des entrées), la valeur des signaux de sortie est placée à côté des flèches de transition - on les sépare des conditions de transition par une barre oblique.**

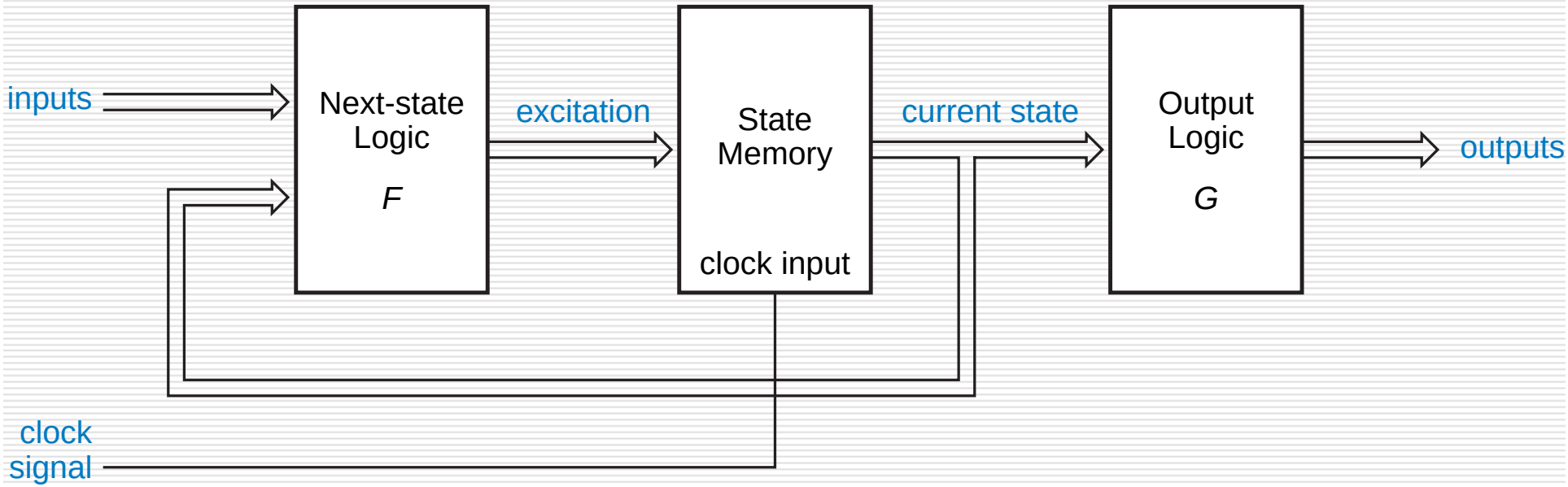
Circuits séquentiels de Moore et de Mealy

- Un circuit séquentiel a toujours des éléments à mémoire et une fonction combinatoire pour calculer le prochain état. Il peut aussi avoir une ou des fonctions combinatoires de Mealy ou de Moore ou des deux.

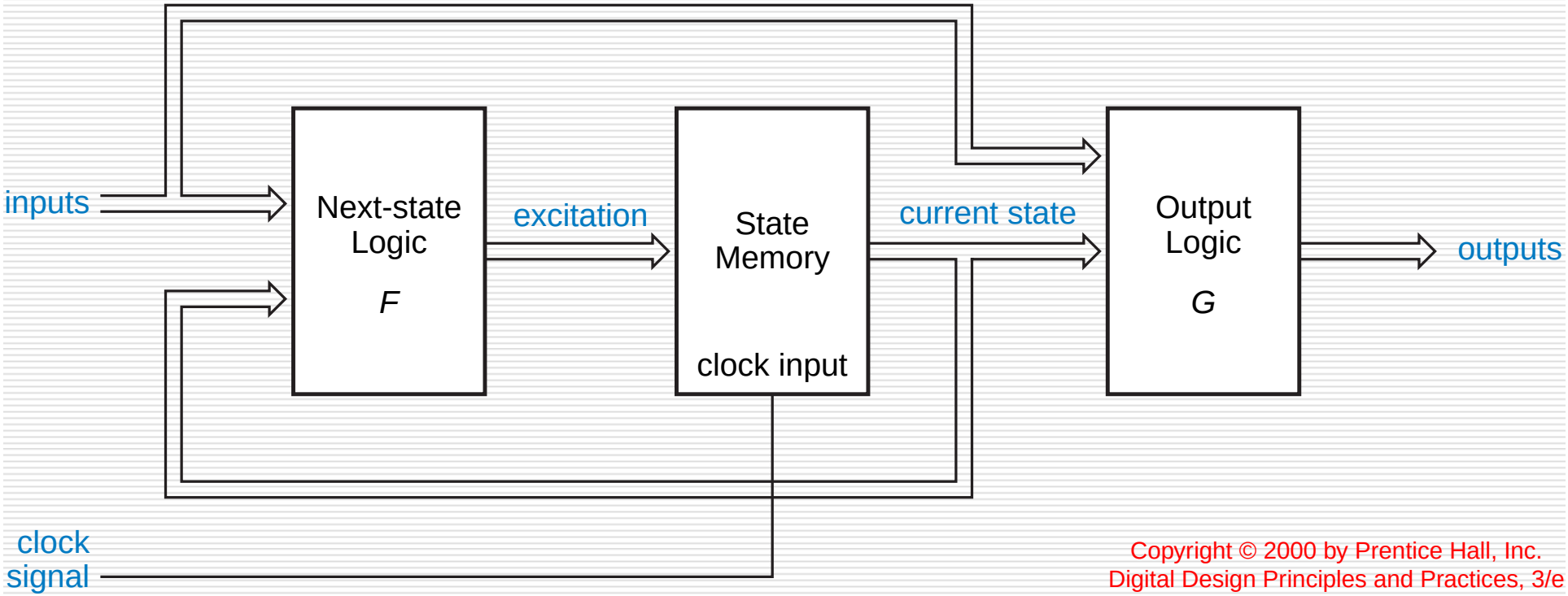
- Dans une machine de Moore, les sorties ne sont fonctions que de l'état présent.

- Dans une machine de Mealy, les sorties sont fonctions de l'état présent ainsi que des entrées.

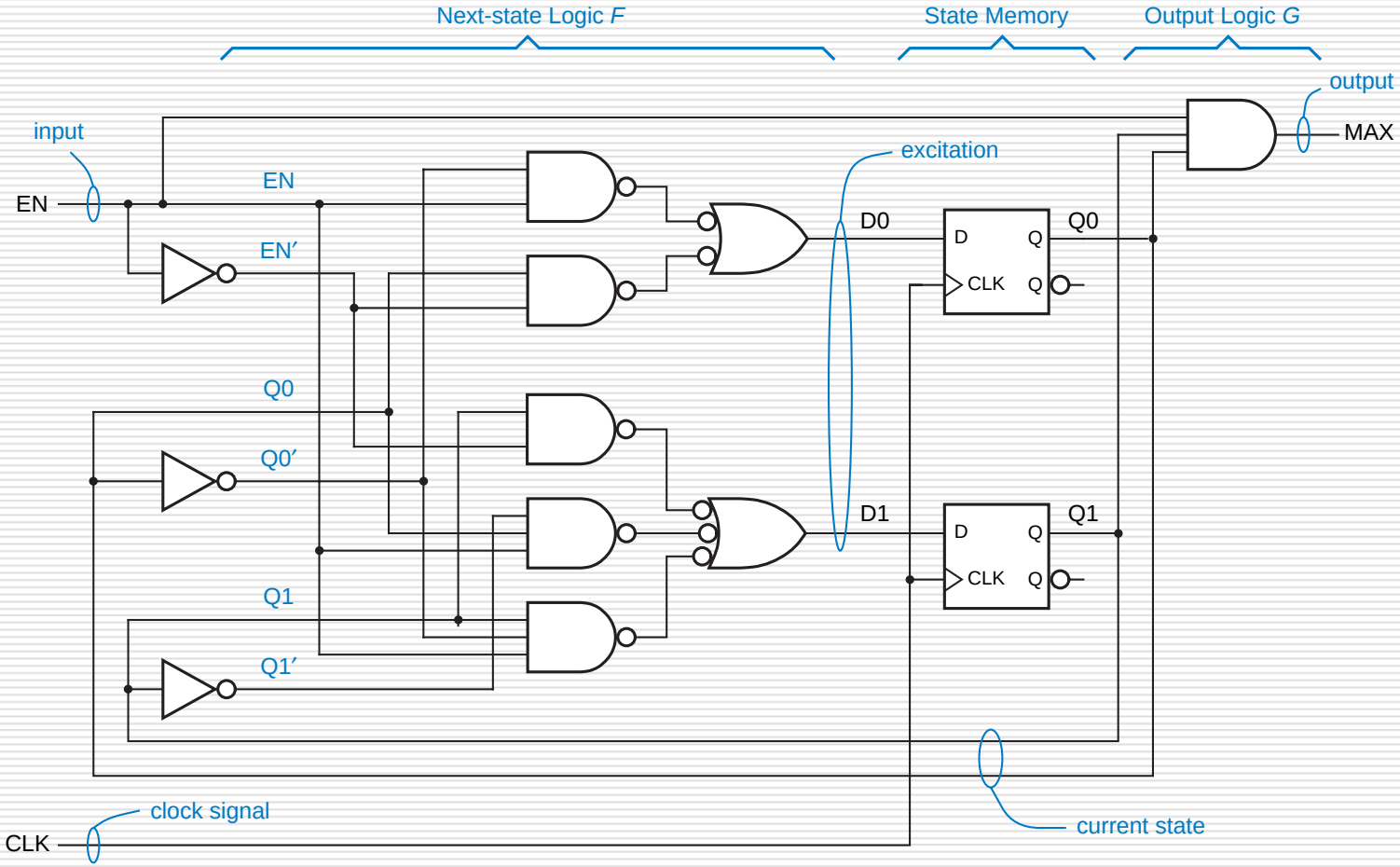
Machine de Moore



Machine de Mealy



Exemple: compteur 2 bits avec la sortie MAX



Exemple 1

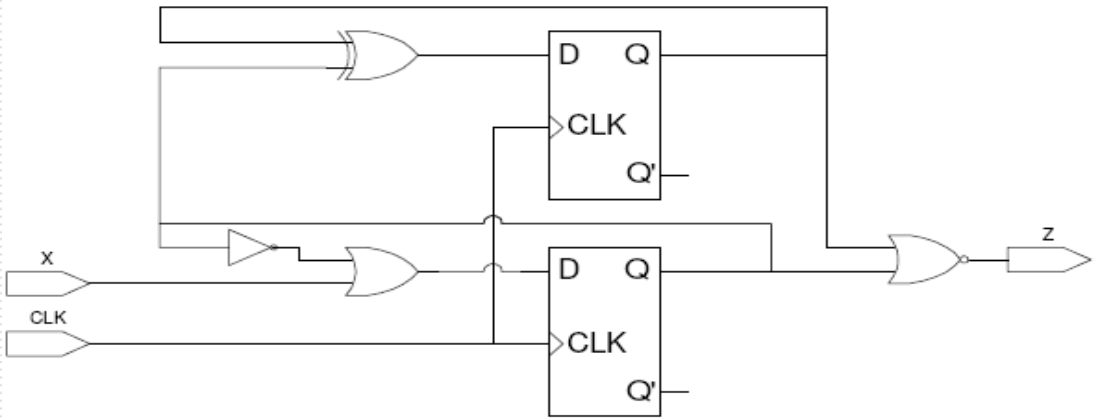
variables d'état, équations d'état et équation de sortie

A+ = **Z =**

B+ =

Tableau d'état

état présent		*entrée*	*état prochain*		*sortie*
A	*B*	*X*	*A+*	*B+*	*Z*
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



Exemple 1

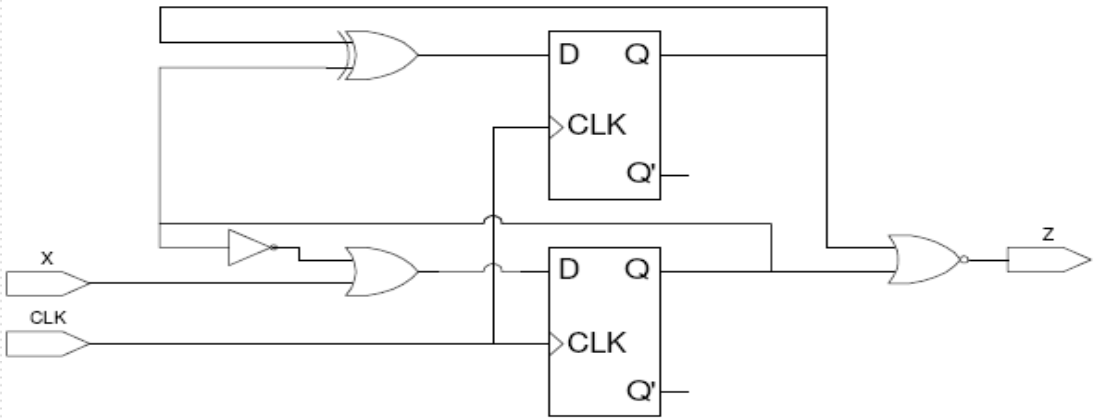
variables d'état, équations d'état et équation de sortie

$A+ = A \oplus B$ $\quad\quad$ $Z = (A + B)'$

$B+ = B' + X$

Tableau d'état $\quad\quad$ **Diagramme d'état ?**

état présent		*entrée*	*état prochain*		*sortie*
A	*B*	*X*	*A+*	*B+*	*Z*
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	1	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	0	1	0



Exemple 2

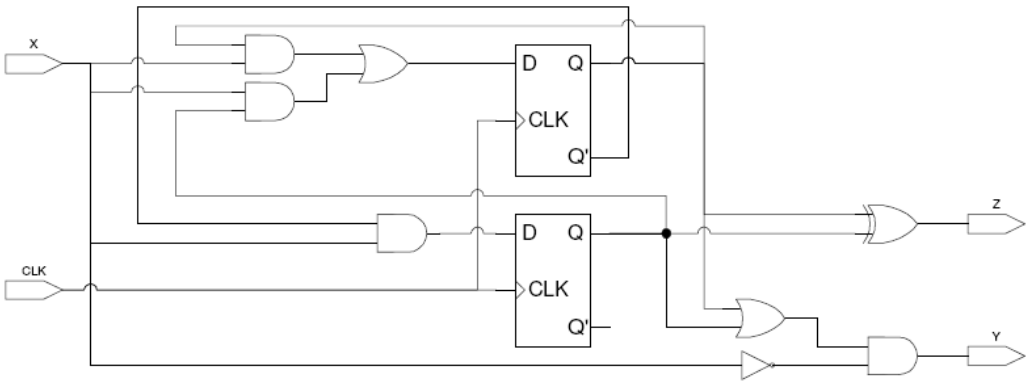
variables d'état, équations d'état et équations de sortie

A+ = **Y =**

B+ = **Z =**

Tableau d'état

état présent		*entrée*	*état prochain*		*sorties*	
A	*B*	*X*	*A+*	*B+*	*Y*	*Z*
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				



Exemple 2

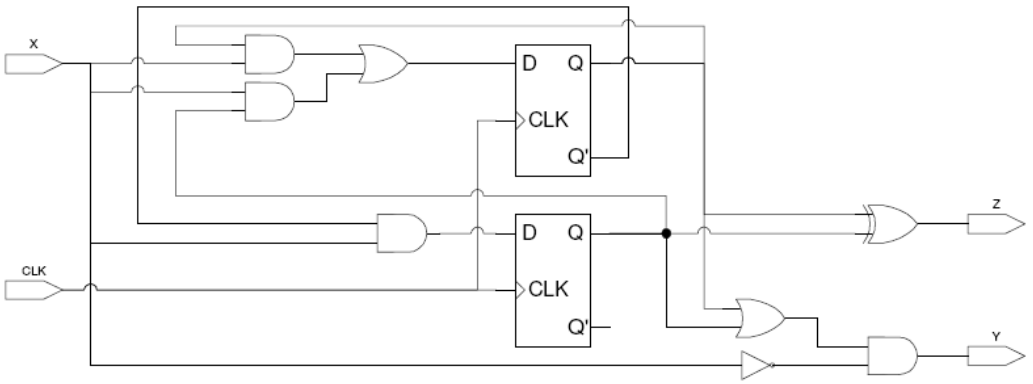
variables d'état, équations d'état et équations de sortie

A+ = XA + XB **Y = (A+B)X'**

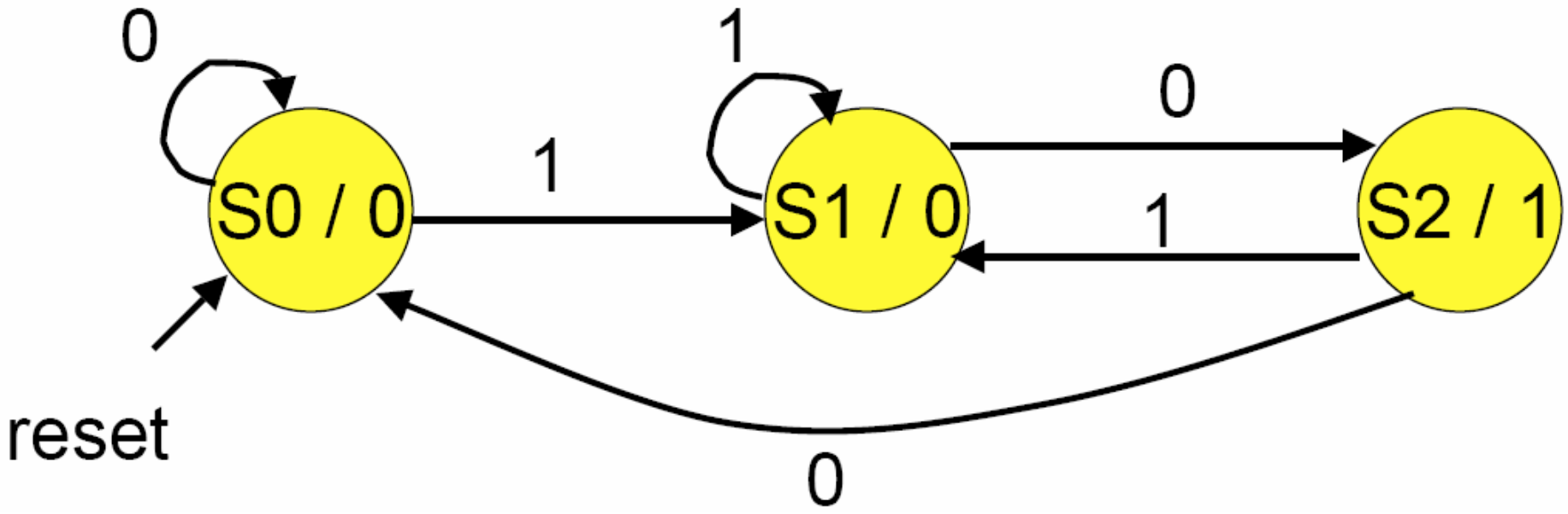
B+ = XA' **Z = A ⊕ B**

Tableau d'état **Diagramme d'état ?**

état présent		*entrée*	*état prochain*		*sorties*	
A	*B*	*X*	*A+*	*B+*	*Y*	*Z*
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	1
0	1	1	1	1	0	1
1	0	0	0	0	1	1
1	0	1	0	0	0	1
1	1	0	0	0	1	0
1	1	1	0	0	0	0

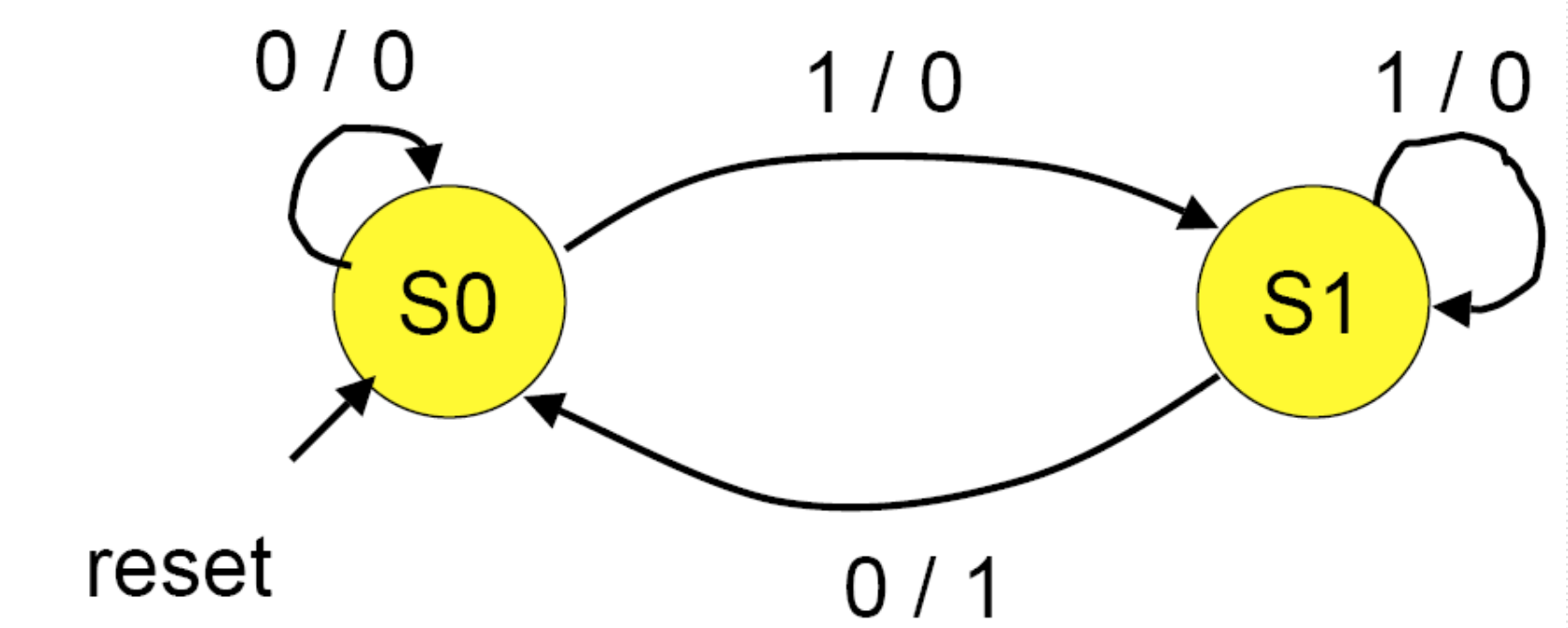


Exemple de FSMs reconnaissant la séquence 10 - En Machine de Moore



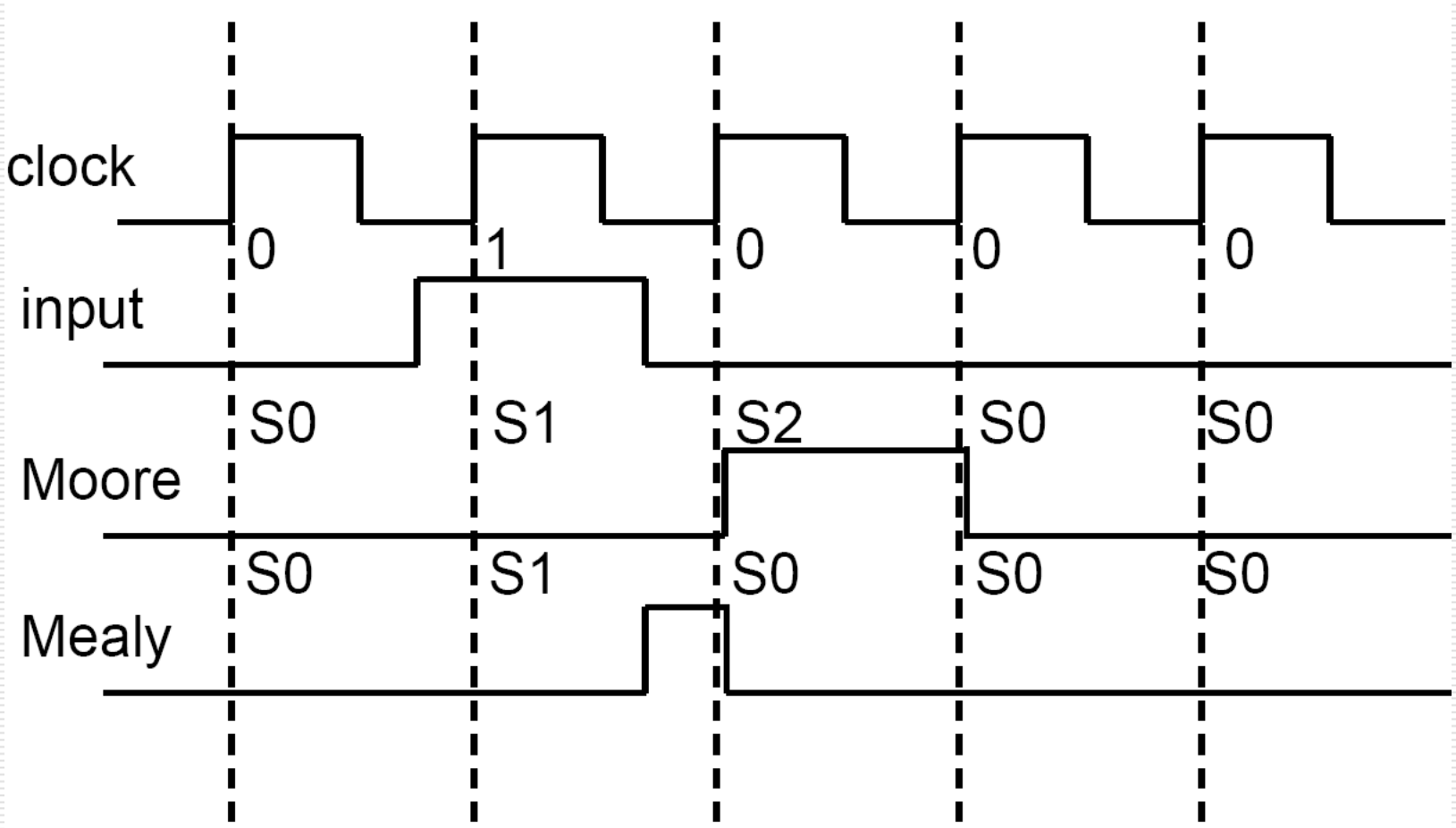
S0: Pas d'élément de la séquence
S1: «1» observé
S2: «10» observé

Exemple de FSMs reconnaissant la séquence 10 - En Machine de Mealy

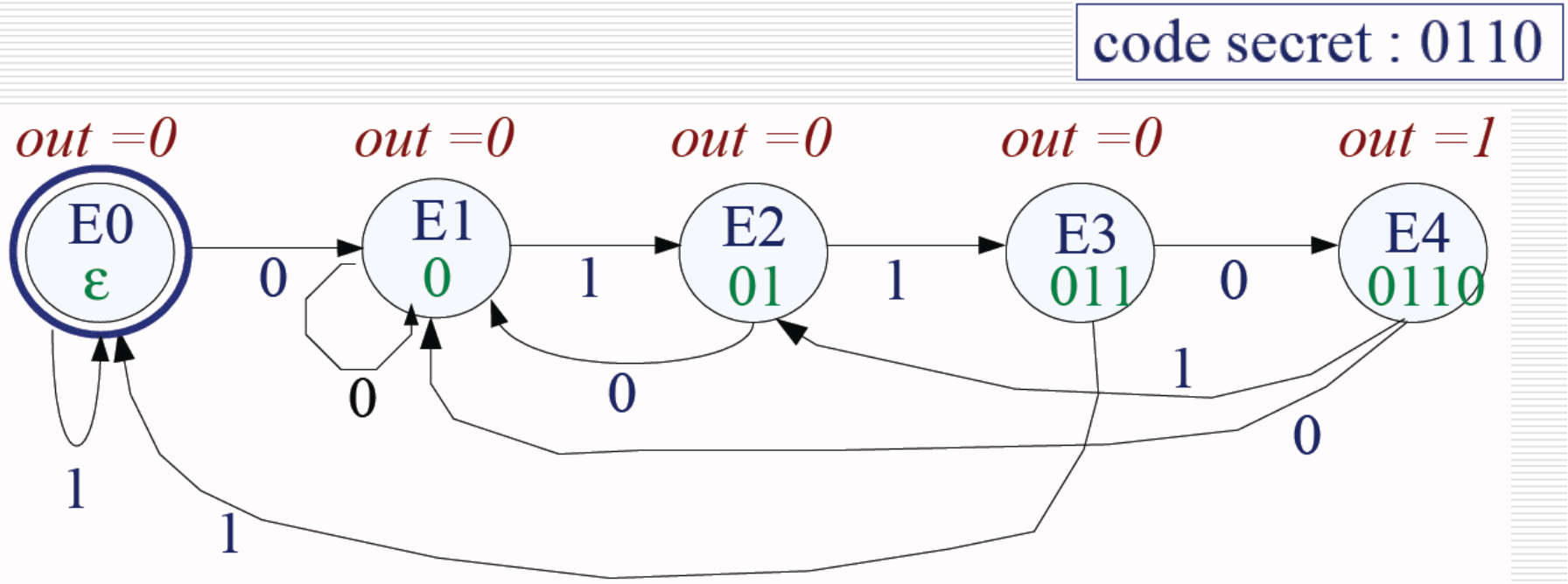


S0: Pas d'élément de la séquence
S1: «1» observé

Exemple de FSMs reconnaissant la séquence 10 – Diagramme temporel



Exemple: Concevoir une machine qui gère un code secret



La sortie «Out» vaut 1 ssi les 4 dernières entrées étaient 0110

Moore et Mealy en VHDL

```
entity FSM is
port ( clk, reset : in std_logic;
       in1, in2   : in std_logic;
       out1       : out
std_logic);
end FSM;
```

```
architecture archi of FSM is
  type STATE_TYPE is (state_0,
state_1, state_2, state_3, state_4);
signal state, next_state :
STATE_TYPE;
```

```
process (clk, reset)
  begin
    if (clk='1' and clk'event) then
       if (reset='0') then  state <=
state_0;
       else  state <= next_state;
       end if;
    end if;
 end process;
```

-Partie séquentielle
→ Implémentation du registre « state »

```
process (state, in1, in2)
  begin
  case state is
      when state_0 =>
        if (in1='0') then
           next_state <=
state_0;
           out1 <= '0';
        else
           next_state <=
state_2;
           out1 <= '1';
        end if;
      when state_1 =>
        …
   end case;
end process;

end archi
```

-Partie combinatoire
→ Implémentation de « Mealy »

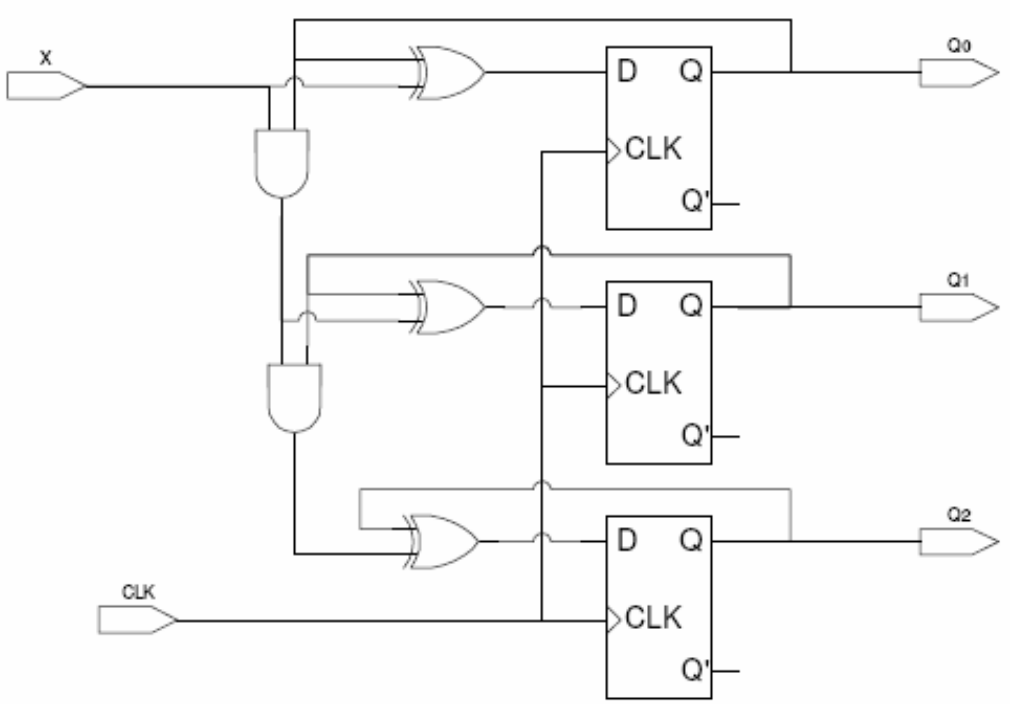
```
process (state , in1, in2)
  begin
    case state is
      when state_0 =>
        out1 <= '0';
        if (in1='0') then
           next_state <=
state_0;
        else
           next_state <=
state_2;
        end if;
      when state_1 =>
        …
    end case;
  end process;

end archi;
```

-Partie combinatoire
→ Implémentation de « Moore »

Exercices

tableau d'état

état présent			*entrée*	*état prochain*		
Q_2	Q_1	Q_0	X	Q_2	Q_1	Q_0
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			



variables d'état, équations d'état et équations de sortie

Q0+ =

Q1+ =

Q2+ =

Exercice: Concevoir un système de feux de circulation à une intersection