heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Analyse et synthèse des systèmes séquentiels (partie 1)

Profs. Peña & Perez-Uribe & Mosqueron
Basé sur le cours du Prof. E. Sanchez

Machine à états finie

- Un système séquentiel est aussi appelé *machine à états finie* ou *Finite State Machine* (FSM) en anglais. Dans une telle machine, une mémoire est nécessaire pour stocker une indication sur l'historique des entrées: c'est l'état du système. Etant donné le caractère fini de la mémoire, la séquence doit être également de taille finie: le nombre d'états d'un système séquentiel est donc fini
- Une machine à états finie asynchrone peut changer ses valeurs de sortie et d'état à chaque fois qu'il y a un changement des entrées
- Une machine à états finie synchrone change d'état à des moments fixes, généralement définis par le flanc montant ou descendant d'un signal d'horloge

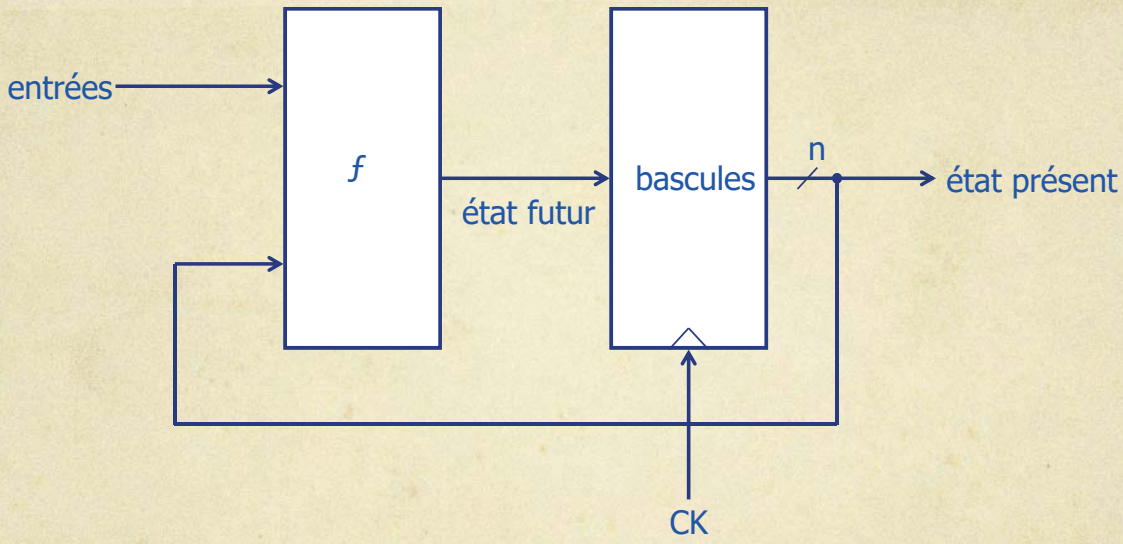
Etat d'un système séquentiel

- L'état d'un système séquentiel est donné par l'ensemble de variables d'état du système, dont les valeurs à un moment donné contiennent toutes les informations sur le passé nécessaires au calcul du comportement futur

état futur = f(état présent, entrées)

- Dans les systèmes séquentiels synchrones le changement d'état est synchronisé avec un signal d'horloge
- Le nombre d'états d'un système séquentiel est une valeur finie, égale à 2^n, où n est le nombre de variables d'état

Etat d'un système séquentiel



Chaque variable d'état peut être stockée dans une bascule D.
f est une fonction combinatoire.

Machines séquentielles synchrones MSS

- Afin de connaitre l'évolution du système une mémorisation est nécessaire.
- L'état du système est stocké dans les variables (bits) d'état du système.
- Ces variables, à un instant donné, contiennent toutes les informations nécessaires au calcul du comportement futur du système.
- L'état futur du système est fonction de son état courant et des entrées
  - état futur = F(état présent,entrées).
- Un système séquentiel synchrone voit son état synchronisé par un signal dit d' «horloge».
- Une MSS est aussi appelée machine d'état.

Décomposition d'une MSS
(Machine Séquentielle Synchrone)

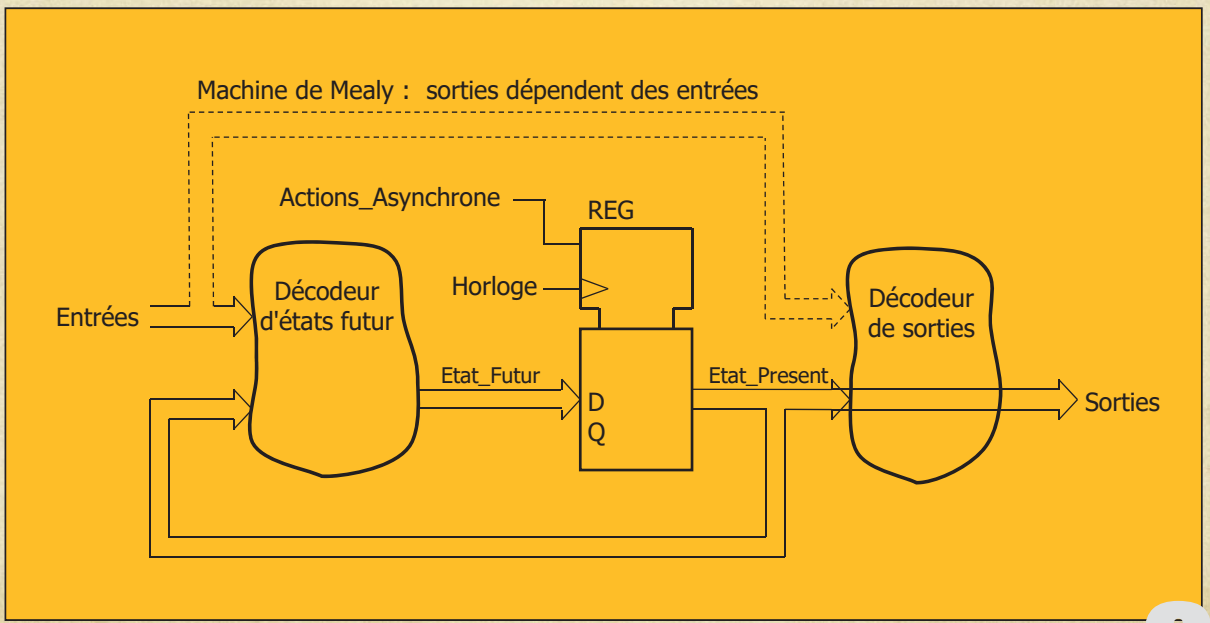
- Ramener les MSS (machines séquentiels synchrones) à :
  des systèmes combinatoires
  \+ des variables internes (mémoires)
- Eléments mémoires constitués par des flip-flops, nommés :
  Bits d'état
- Les flip-flops répondent à la définition énoncée au chapitre précédent

Types de machine d'état (MSS simples)

Trois types de MSS simples

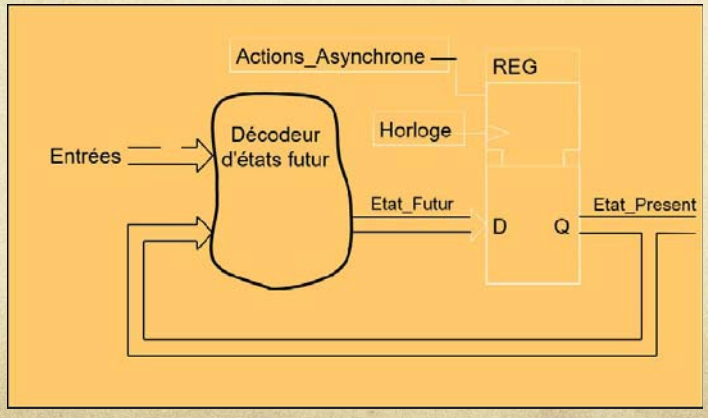
- MEDVEDEV
  - Sorties correspond directement aux bits d'états
- MOORE
  - Sorties dépendent uniquement de l'état présent
- MEALY
  - Sorties dépendent de l'état présent **et** des entrées

Schéma bloc



Décodeur d'état futur :

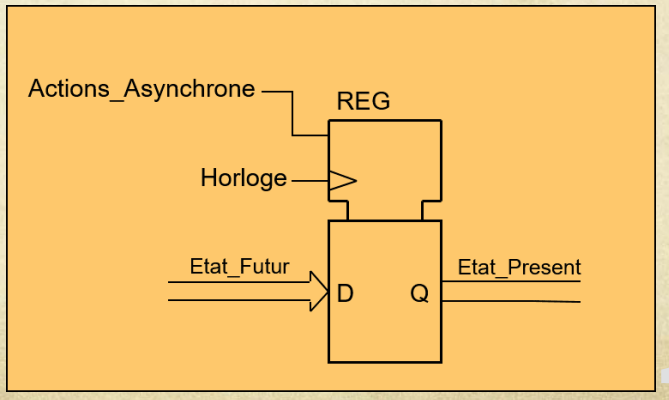
- Système logique combinatoire (SLC)
- Détermine le prochain état en fonction de :
  - état actuel de la machine =>
    état mémorisé dans les flip-flops (bits d'états)
  - valeur actuelle des signaux d'entrées



Etat de la MSS / Bits d'état

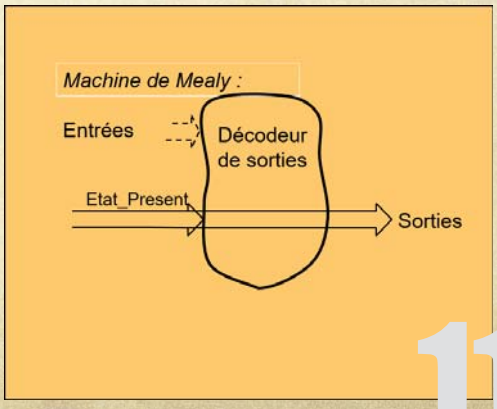
Bits d'état:

- Mémorisation de l'état présent
- Registre synchrone constitué de flip-flops D



Décodeur de sorties

- Système logique combinatoire
- Détermine l'état des sorties en fonction de :
  - état actuel de la machine =>
    état mémorisé dans les flip-flops (bits d'états)
  - valeur actuelle des entrées (MSS Mealy)

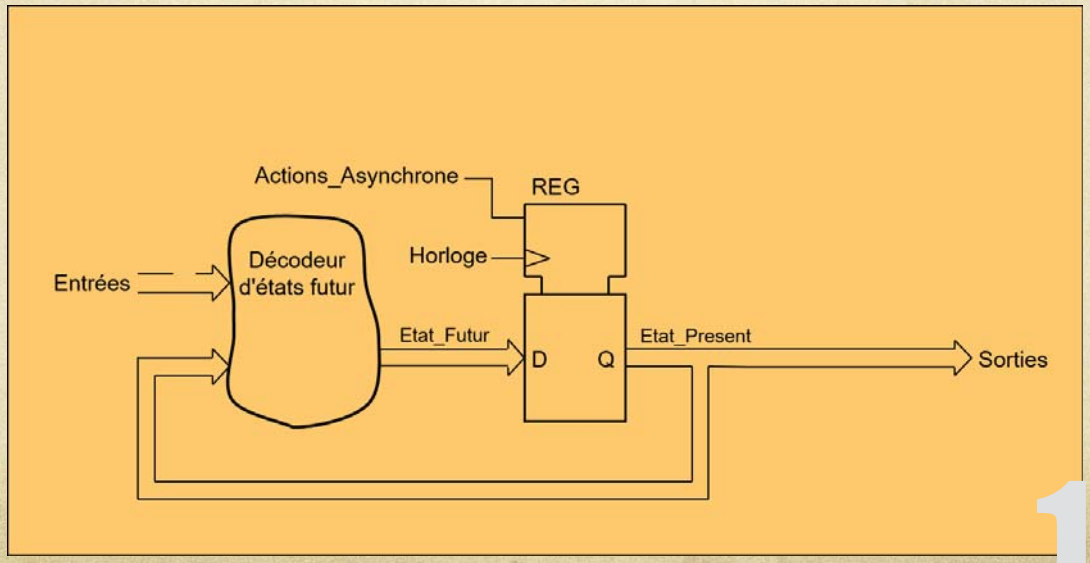


MSS simple: principe

- Principe de fonctionnement général:
  - Le système fait évoluer les entrées
  - L'état interne évolue selon le changement des entrées
  - Les sorties dépendent de l'état interne
    - Sauf Mealy: les sorties dépendent aussi des entrées (changement immédiat)

MSS simple de MEDVEDEV

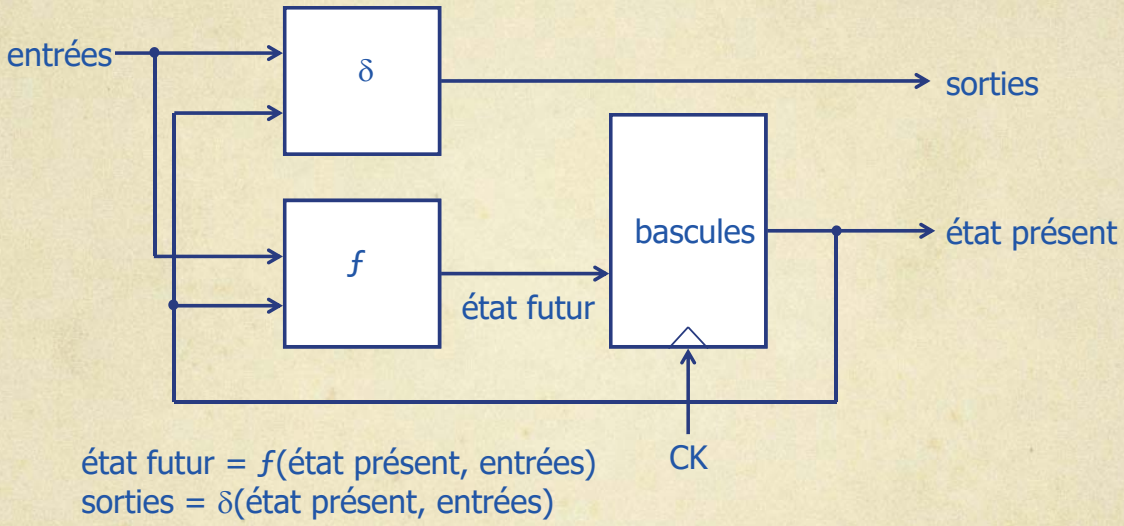
- Les sorties sont égales aux bits d'états
  - Sorties inconditionnelles



Machine de Mealy

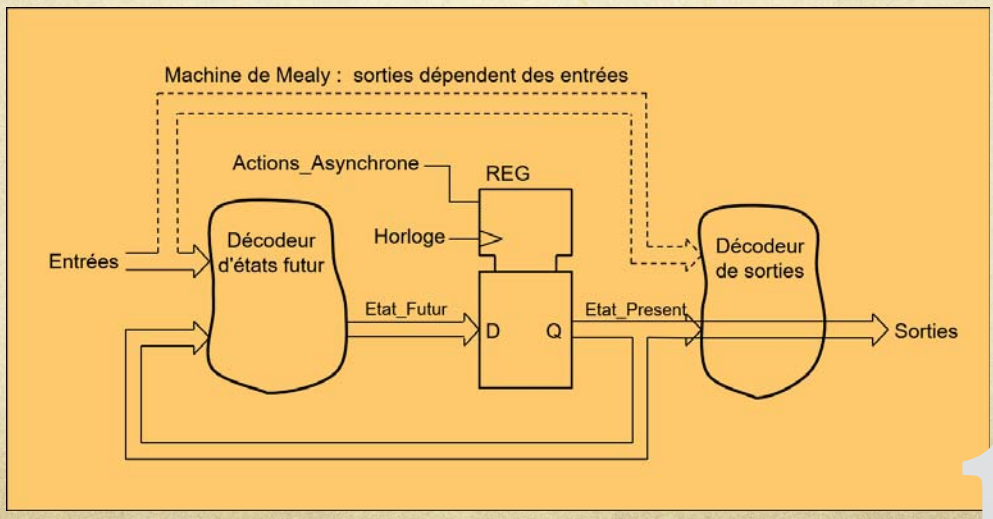
- Si les valeurs des variables de sortie dépendent de l'état présent et des variables d'entrée, le système séquentiel est appelé une machine de Mealy
- Dans ce cas, le changement des sorties n'est pas synchrone: il se fait avec le changement des entrées

MSS simple de MEALY



MSS simple de MEALY

- Les sorties changent immédiatement avec les entrées et après la mise à jours de l'état interne (bits d'état)
  - sorties conditionnelles

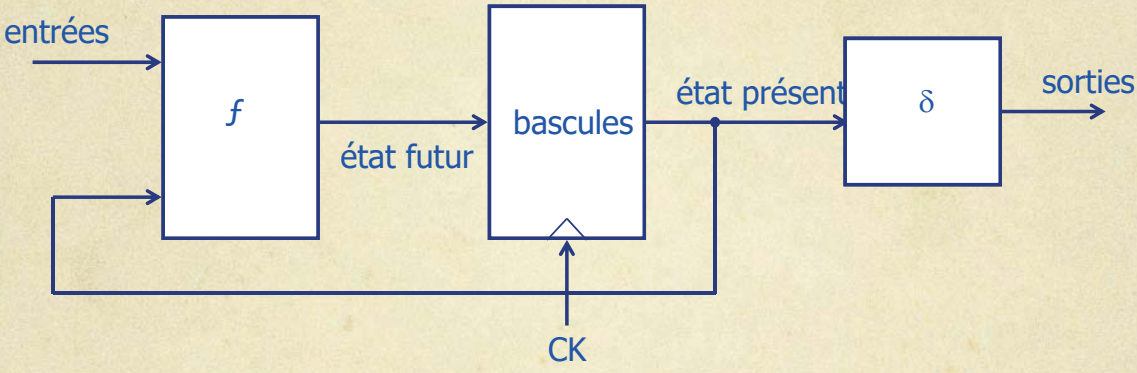


Machine de Mealy

- Exemple de code VHDL:

Machine de Moore

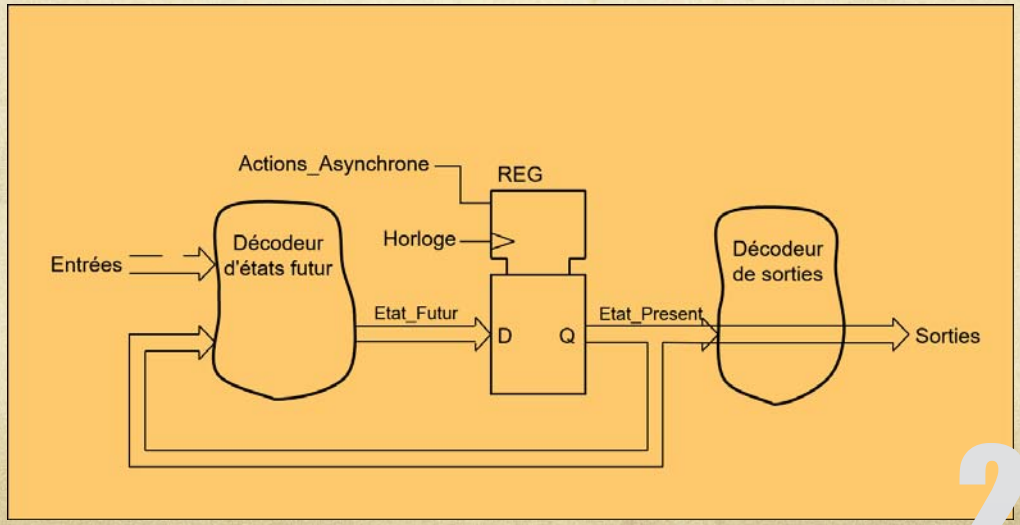
- Si les valeurs des variables de sortie dépendent uniquement de l'état présent, le système séquentiel est appelé une machine de Moore
- Dans ce cas, le changement des sorties est synchrone: il se fait avec le changement des états



état futur = ƒ(état présent, entrées)
sorties = δ(état présent)

MSS simple de MOORE

- Les sorties changent uniquement après la mise à jour de l'état interne (bits d'état)
  - Sorties inconditionnelles



Machine de Moore

- Exemple de code VHDL:

- Sauf contraintes de type physique spécifiées dans le cahier des charges, tout problème peut être résolu par une machine de type Mealy ou Moore, indifféremment
- Normalement, pour un même problème, une machine de Moore demande plus d'états que la machine de Mealy équivalente
- **Toutefois, assez souvent on préfère la solution de type machine de Moore**. En effet, les sorties d'une machine Mealy ne sont pas synchronisées avec l'horloge, pouvant changer en même temps qu'une entrée externe. Cet asynchronisme amène comme conséquence des durées quelconques pour les sorties
- Comme les sorties des machines de type Moore sont synchronisées, leur durée est toujours un multiple de la période du signal d'horloge

Description du fonctionnement

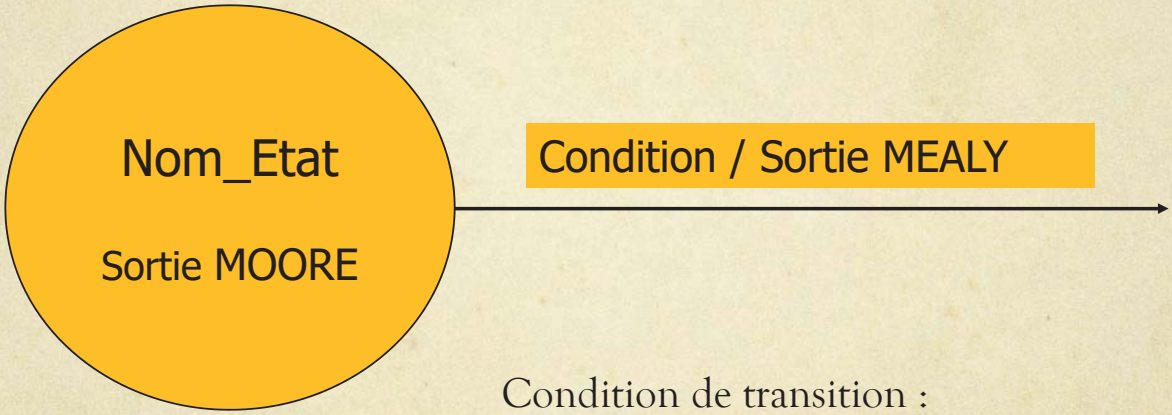
- Chronogramme :
  - présente une seule séquence
  - vue partielle
- Graphe des états :
  - méthode graphique
  - permet de représenter tous les "chemins" possibles
  - pratique pour des MSS simples

Graphe des états ...

- Constitué d'un série de bulles reliées par des flèches (transitions):
  - une bulle représente un état distinct du système séquentiel. Elle correspond à un code des bits d'état.
  - une flèche représente une transition entre deux états. La condition, fonction des entrées, sera écrite sur chaque transition.

... graphe des états

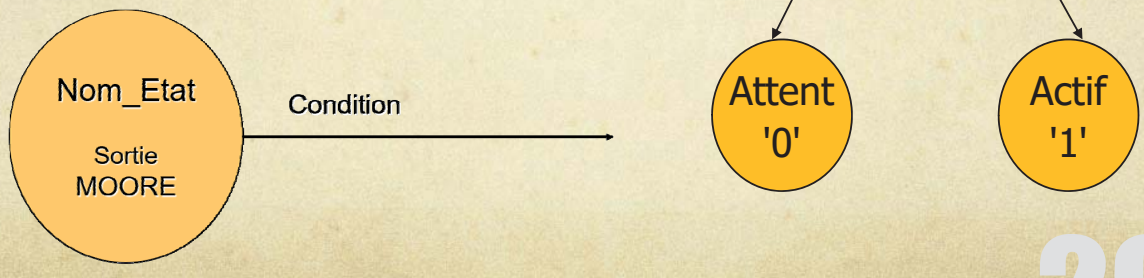
Convention de dessin :

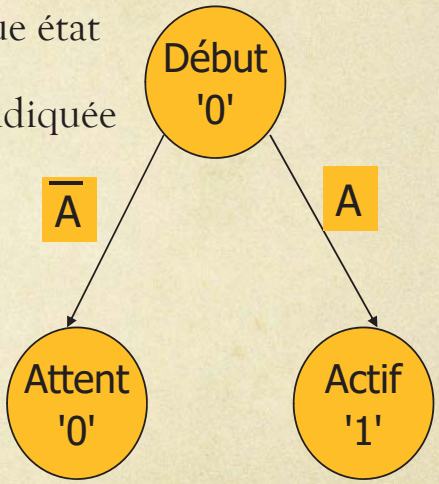


Condition de transition :
équation logique fonction des entrées

Graphe des états pour MOORE

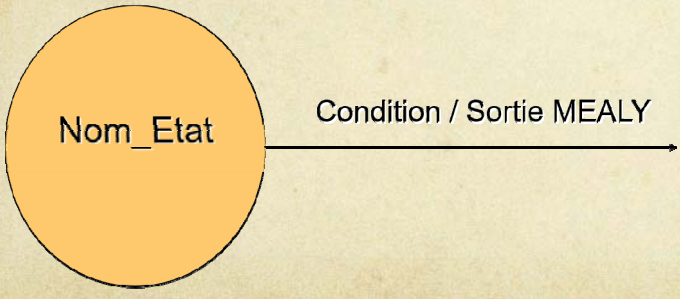
- Chaque état est nommé
- L'état de la sortie est indiqué dans chaque état
- Sur chaque transition la condition est indiquée

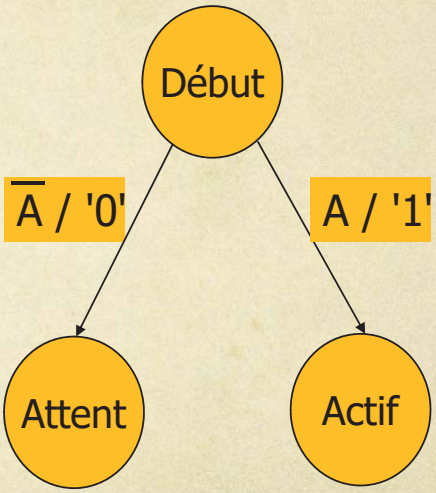




Graphe des états pour MEALY

- Chaque état est nommé
- Sur chaque transition :
  - la condition est indiquée
  - l'état de la sortie est indiqué





Conception d'un graphe des états

<u>Méthode de conception :</u>

- Génération de proche en proche
- Génération depuis un chronogramme
- Génération par l'identification des états internes

- Dans tous les cas :
  Compléter le graphe en respectant les règles de construction

Règles construction graphe des états

1. De chaque bulle d'état :
   - Autant de flèches que de combinaisons valides des entrées
2. Chaque changement des entrées pertinentes (nouvelle combinaison) provoque un changement d'état
   - Génère souvent trop d'états => seront simplifiés après
3. Chaque changement sur les sorties, avec les mêmes valeurs des entrées, implique un changement d'état
4. Maintien des sorties pendant un certains temps (multiple Thorloge) est réalisé par une succession d'état interne (1 état = maintien de 1 $T_{horloge}$)

Table des états ...

- Représentation sous la forme d'un tableau du graphe des états
- C'est la liste des actions synchrones de la MSS, équivalent à la liste des actions synchrones d'un registre ou compteur
- Utilisée pour déterminer les équations logiques du décodeur d'état futur et du décodeur de sortie

Structure pour une MSS de Moore

Toutes les combinaisons des entrées

Sorties
Dépendent uniquement état présent

Etat présent	Etat futur f(A,B) 00	01	11	10	Sorties X Y
E0	(E0)	(E0)	E1	E2	0 0
E1	E2	E0	(E1)	E0	1 0
E2	E0	E3	E1	(E2)	0 1
…	…	….			…

Liste de tous les états présents

(…) Indique un état stable

Structure pour une MSS de Mealy

Toutes les combinaisons des entrées

Sorties
Dépendent de l'état présent et des entrées

Etat présent	Etat futur f(A,B) / Sorties X Y 00	01	11	10
E0	(E0) / 00	(E0) / 00	E1 / 10	E2 / 01
E1	E2 / 01	E0 / 00	(E1) / 10	E0 / 00
E2	E0 / 00	E3 / 11	E1 / 10	(E2) / 01
…	…	….		

Liste de tous les états

(…) Indique un état stable

Réduction MSS

- Méthode pour réduire le nombre des états
- Recherche des états identiques
  - même états futurs et mêmes sorties } pour chaque combinaison des entrées
- Recherche des états compatibles
  - états futurs différents mais compatible après groupement des états compatibles;
    les sorties doivent être identique

Codage des états d'une MSS

- Jusqu'ici :
  - Etats internes désignés par un nom symbolique
- Réaliser un circuit :
  - Assigner un code binaire distinct à chaque état
- MSS comportant M états, il faudra N bits avec :

$$2^N \geq M$$

Choix du code des états

- Il y a des règles obligatoires à respecter pour garantir le bon fonctionnement de la MSS.
- D'autre part il faut optimiser le codage ?
  - Dans le cas d'une machine à 9 états (m=9), avec laquelle nous utilisons 4 bits pour le codage (n=4), il y a 10,8 millions de possibilités de codage distincts !
  - Nous verrons des pistes pour optimiser le codage.

Contraintes pour fonctionnement correct

Le codage doit garantir un fonctionnement correct de la MSS

- Entrées asynchrones :
  - **Obligatoire** de synchroniser les entrées correspondantes avec un flip-flop
- Sorties sans aléas (transitoires):
  - **Obligatoire** de générer les sorties correspondantes directement par un flip-flop,
    => la sortie doit correspondre à un bit d'état de la MSS!

Choix du code des états

- Déterminer le nombre de bits d'états N :
  - $2^N \geq$ nombres d'états
- puis
  - Essayer de placer les états dans une table de Karnaugh en respectant les sorties sans aléas et en simplifiant au mieux les décodeurs
- Remarques :
  - Code zéro ("00...0") pour l'état initial, facile à initialiser
  - Eventuellement augmenter le nombre de bits d'état

Exemple de codage des états

- Soit une MSS avec 6 états (E0 à E5)
  - L'état E0 est l'état initial de la MSS
  - Les entrées sont synchrones
  - Les sorties sont utilisées pour des commandes synchrones (aléas acceptables)
  - aucunes optimisations des décodeurs n'est appliquées

exemple codage binaire

- Voici un codage des états :
  - Nous avons 6 états => il faut 3 bits ($2^3 > 6$)
  - Nous utilisons l'état "000" pour E0 (état initial)

  Choix arbitraire pour les autres états :

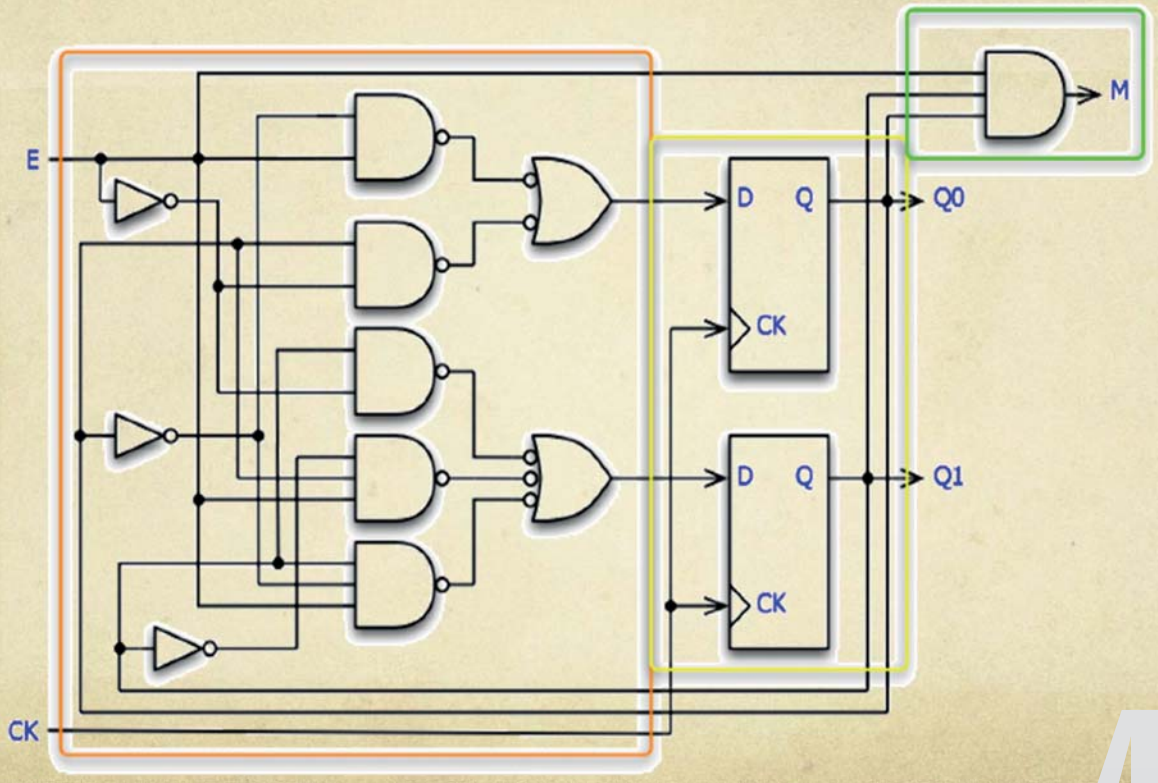
E0 = "000"	E3 = "011"
E1 = "001"	E4 = "100"
E2 = "010"	E5 = "101"

  Les codes "110" et "111" sont non utilisés

Analyse des machines séquentielles

- Déterminer les fonctions d'état futur et de sortie, de façon à pouvoir prédire le comportement de la machine
- La spécification du comportement est faite à l'aide de deux types de représentation:
  - la table d'états
  - le graphe des états

Exemple: machine de Mealy



Exemple: machine de Mealy

- Equations du système:

Les équations des entrées des bascules sont:

$$D0 = Q0 \bullet \overline{E} + \overline{Q0} \bullet E$$

$$D1 = Q1 \bullet \overline{E} + \overline{Q1} \bullet Q0 \bullet E + Q1 \bullet \overline{Q0} \bullet E$$

En remplaçant l'équation caractéristique des bascules D ($Q^+ = D$), on obtient les équations des variables d'état futur:

$$Q0^+ = Q0 \bullet \overline{E} + \overline{Q0} \bullet E$$

$$Q1^+ = Q1 \bullet \overline{E} + \overline{Q1} \bullet Q0 \bullet E + Q1 \bullet \overline{Q0} \bullet E$$

L'équation de la sortie est:

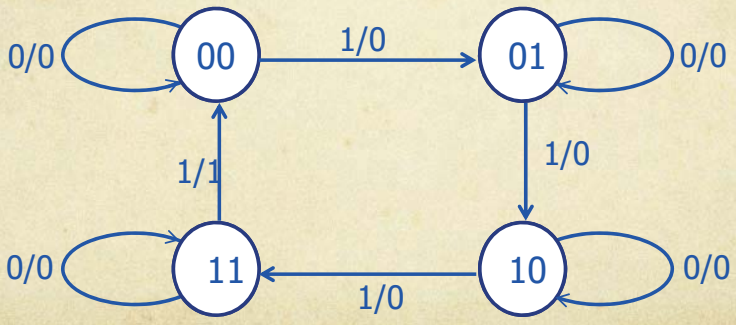
$$M = Q1 \bullet Q0 \bullet E$$

- Mode de représentation: la table d'états
  - chaque ligne de la table représente un état présent
  - chaque colonne de la table représente une combinaison des entrées
  - a chaque case de la table on introduit l'état futur et la sortie correspondante
- Pour notre exemple, on obtient:

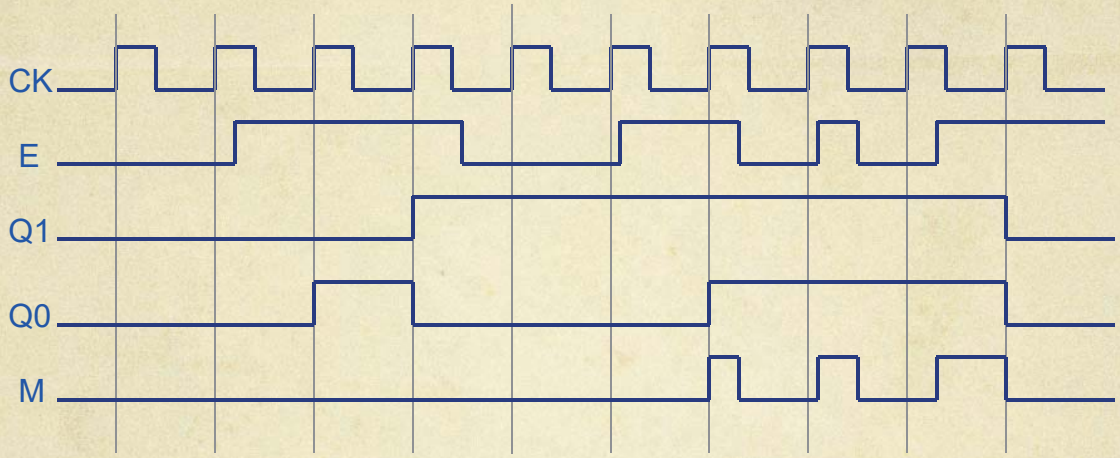
	E	
	0	1
00	00,0	01,0
01	01,0	10,0
10	10,0	11,0
11	11,0	00,1
Q1Q0		

- Pour une machine de Moore, la table est plus simple: il y a une seule valeur de sortie par état, c'est-à-dire par ligne

- Mode de représentation: le graphe des états
  - chaque état est représenté par le sommet d'un graphe orienté
  - les changements d'état sont représentés par des flèches
  - sur les flèches on indique les valeurs des entrées qui produisent le changement d'état et les valeurs des sorties
- Pour notre exemple, on obtient:



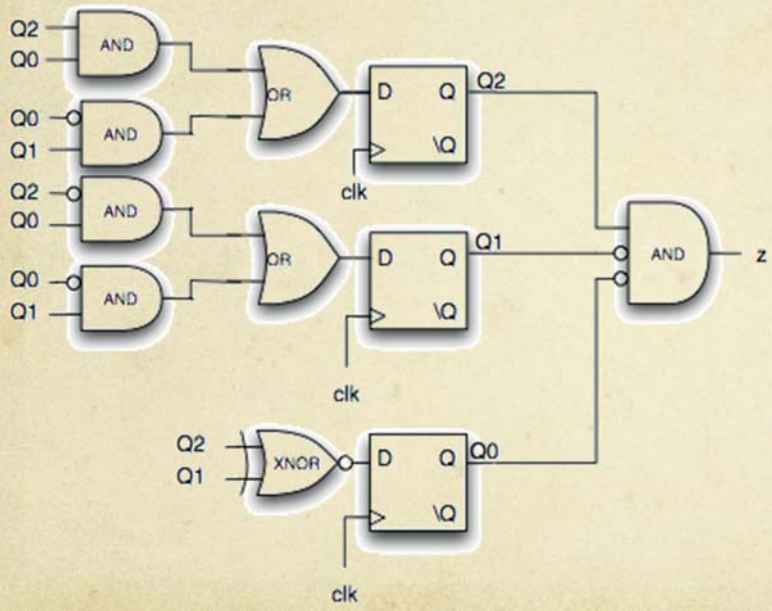
- Mode de représentation: le chronogramme



  - les états changent seulement avec le flanc montant de l'horloge
  - les sorties peuvent changer à tout moment, avec le changement d'un état ou d'une entrée (dans une machine de Moore, les sorties changent uniquement avec le changement d'un état)

- Dans cet exemple on peut voir le désavantage d'une machine de Mealy par rapport à une machine de Moore: la durée de la sortie M est toujours inférieure à une période d'horloge et son changement (passage à 0 ou passage à 1) peut survenir à n'importe quel moment, en fonction des changements de l'entrée E
- Si la sortie d'une machine séquentielle sert à contrôler une action externe, la durée du signal est un facteur très important

Exemple: machine de Moore



Equations du système

$Q_2^+ = Q2Q0 + Q0' Q1$

$Q_1^+ = Q2' Q0 + Q0' Q1$

$Q_0^+ = Q2Q1 + Q2' Q1'$

$Z = Q2Q1' Q0'$

Exemple: machine de Moore (2)

Equations du système

$Q_2^+ = Q2Q0 + Q0' Q1$

$Q_1^+ = Q2' Q0 + Q0' Q1$

$Q_0^+ = Q2Q1 + Q2' Q1'$

$Z = Q2Q1' Q0'$

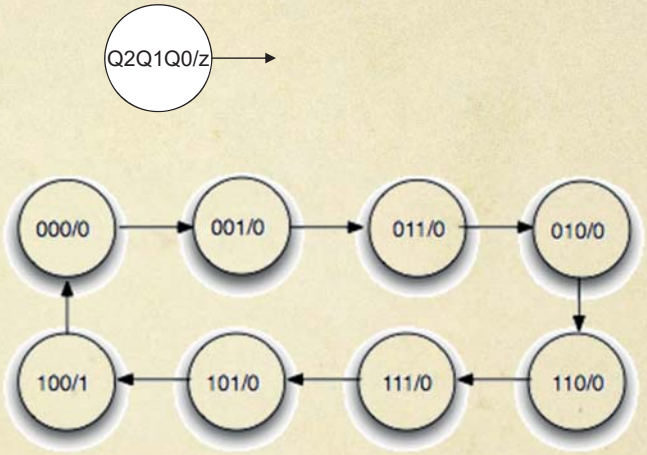
Table d' états

	$Q_2^+ Q_1^+ Q_0^+ / z$
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
Q2Q1Q0	

Exemple: machine de Moore (3)

Table d' états

	$Q_2^+Q_1^+Q_0^+/z$
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
Q2Q1Q0	



Compteur Gray 3 bit