

Chapitre III : conception des Circuits Séquentiels

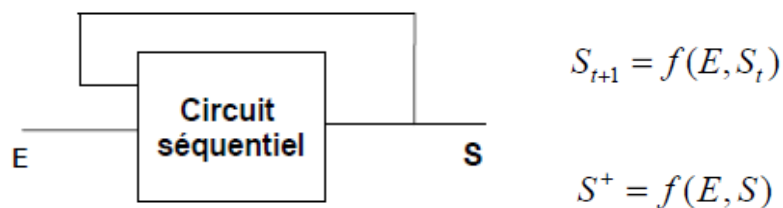
1. Introduction :

Dans la logique combinatoire nous les signaux de sortie ne dépendaient que des états des variables d'entrée. Pour les circuits de logique séquentielle nous devons tenir compte de l'état du système. Ainsi les sorties dépendent des entrées mais également de l'état du système. Celui-ci dépend aussi des entrées.

Les systèmes séquentiels sont des systèmes dont le fonctionnement dépend d'une part de la valeur des entrées et d'autre part par l'état du système

La logique séquentielle a pour élément de base « la bascule » contrairement à la logique combinatoire qui avait pour élément de base la porte logique. Les circuits séquentiels présentent une caractéristique de mémoire. La différence entre la logique combinatoire et la logique séquentielle est que :

- logique combinatoire : les états de sortie dépendent uniquement de la combinaison des variables d'entrées.
- logique séquentielle : l'état de la sortie dépend à la fois de la combinaison des variables d'entrée et de l'état antérieur de la sortie (temporelle).



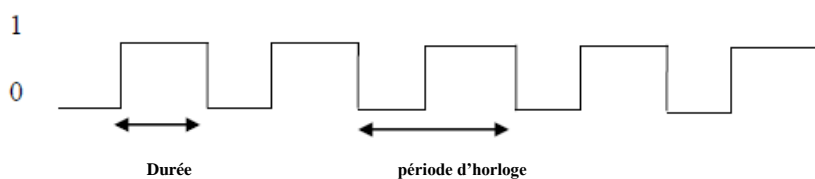
2. Système séquentiels synchrone et asynchrones :

On classe les systèmes séquentiels en deux grandes catégories :

a- Systèmes séquentiels asynchrones: dit ainsi s'il peut évoluer seul sans ordre extérieur. Les sorties sont définie par les entrées et par le fonctionnement de circuit.

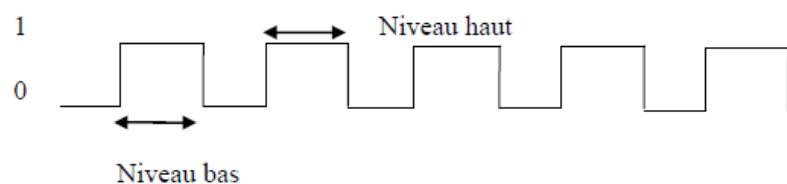
b- Systèmes séquentiels synchrones: dit ainsi s'il ne peut évoluer que sur un ordre extérieur à lui-même (en absence d'ordre le système reste figé dans l'état ou il se trouve) on appelle cette entrée de commande horloge.

Le signal horloge : est un signal de synchronisation périodique logique qui passe de l'état 1 à l'état 0 et de 0 à 1 d'une façon périodique dans le temps.



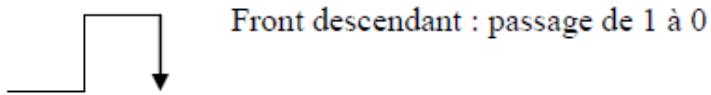
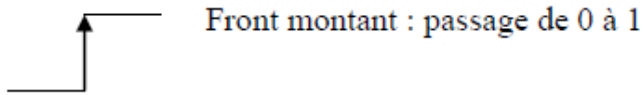
Forme des signaux de commandes :

a- Signal à niveau :



b- Signal impulsionnel



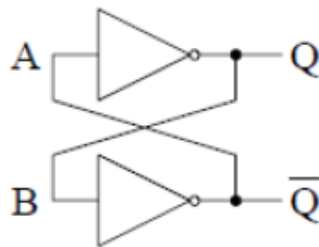


Les entrées ne sont validées que au moment ou les impulsions d’horloge sont produites, certain sont sensible à des fronts montants ou descendants.

3. Les bascules :

3.1. Définition d’une bascule :

Une **bascule (flip-flop)** a pour rôle de mémoriser une information élémentaire. C'est une mémoire à 1 bit. Une bascule possède deux sorties complémentaires Q et \bar{Q} . La mémorisation fait appel à un verrou (latch) ou système de blocage, dont le principe de rétro-action peut être représenté de la façon suivante :



Les bistables (Flip Flop) et les Bascules (MASTER SLAVE Flip Flop ou LATCH) représentent la base de la logique séquentielle dont la fonction essentielle est la fonction de mémorisation. La bascule est un circuit qui comporte une ou plusieurs entrées et deux sorties complémentaires Q et \bar{Q} .

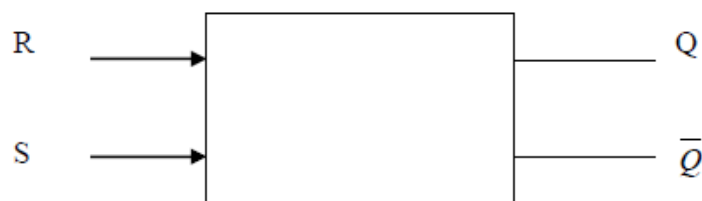
3.2. Les types des bascules :

Les bistables les plus utilisées sont :

Bistable RS, Bistable JK ; bistable D et bistable T.

3.2.1. Bascule RS :(Reset set)

Elle comporte deux entrées R et S ; deux sorties Q et \bar{Q} .



Le nouvel état de la bascule Q^+ dépend de son état antérieur Q et de l’état des entrées R et S.

$$Q^+ = f(Q, R, S)$$

Fonctionnement statique:

R : Reset : entrée de mise à 0 de la bascule (si R=1 ; la sortie Q=0).

S : set : entrée de mise à « 1 » de la bascule (si S=1 ; la sortie Q=1).

La condition R=S=1 est interdite. Elle donne un état indéterminé de la sortie.

Table de vérité:

R	S	Q	Q ⁺
0	0	0	0 mémorisation de l'information
0	0	1	1 mémorisation de l'information
0	1	0	1 mise à 1 de la sortie
0	1	1	1 mise à 1 de la sortie
1	0	0	0 mise à 0 de la sortie
1	0	1	0 mise à 0 de la sortie
1	1	0	X état indéterminé
1	1	1	X état indéterminé

La table de vérité condensée

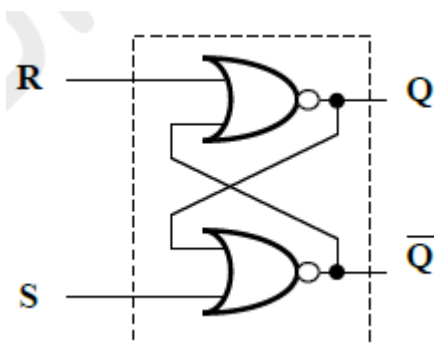
R	S	Q ⁺
0	0	Q conservation de l'état interne
0	1	1 mise à 1 ($\forall Q$)
1	0	0 mise à 0 ($\forall Q$)
1	1	X état indéterminé

Les équations d'état

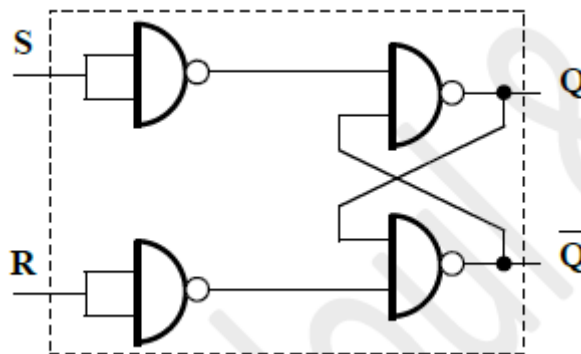
$$\begin{aligned}
 Q^+ &= \bar{R}\bar{S}Q + \bar{R}SQ + \bar{R}S\bar{Q} \\
 &= \bar{R}\bar{S}Q + \bar{R}S(Q + \bar{Q}) \\
 &= \bar{R}\bar{S}Q + \bar{R}S \\
 &= \bar{R}(\bar{S}Q + S) \\
 &= \bar{R}((\bar{S} + S)(Q + S)) \\
 &= \bar{R}(Q + S)
 \end{aligned}$$

$$Q^+ = \bar{Q}^+ = \overline{\overline{\bar{R}(Q + S)}} = R + \overline{(Q + S)}$$

Réalisation à l'aide de portes NOR



Réalisation à l'aide de portes NAND



Diagrammes d'états ou de transitions

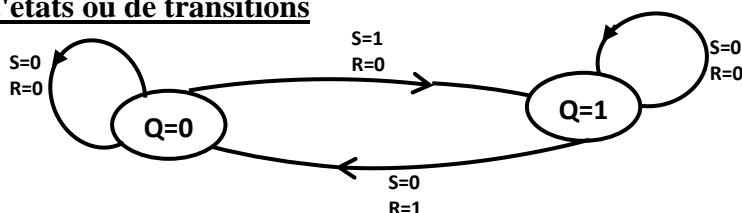
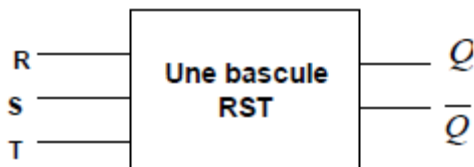


Table de transition d'une bascule RS

Q	Q ⁺	R	S
0	0	X	0
0	1	0	1
1	0	1	0
1	1	0	X

3.2.2. Bascules RST:



Fonctionnement :

Les entrées R et S sont prises en compte que si elles sont en coïncider avec un signal de commande T(horloge)

- Si T=0 la bascule conserve son état.

- Si T=1

$$\left\{ \begin{array}{ll} R = S = 0 & \text{conservation} \\ R = 1, S = 0 & \text{mise à 0} \\ R = 0, S = 1 & \text{mise à 1} \\ R = S = 1 & \text{indéter min é} \end{array} \right.$$

Table de vérité:

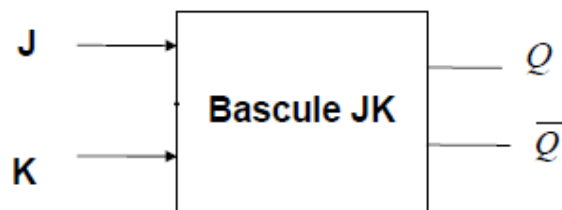
T	R	S	Q	Q ⁺
0	0	0	0	0 conservation d'état
0	0	0	1	1 conservation d'état
0	0	1	0	0 conservation d'état
0	0	1	1	1 conservation d'état
0	1	0	0	0 conservation d'état
0	1	0	1	1 conservation d'état
0	1	1	0	0 conservation d'état
0	1	1	1	1 conservation d'état
1	0	0	0	0 conservation d'état
1	0	0	1	1 conservation d'état
1	0	1	0	1 mise à 1 de la sortie
1	0	1	1	1 mise à 1 de la sortie
1	1	0	0	0 mise à 0 de la sortie
1	1	0	1	0 mise à 0 de la sortie
1	1	1	0	X état indéterminé
1	1	1	1	X état indéterminé

La table de vérité condensée

T	R	S	Q ⁺
0	X	X	Q mémorisation de l'information (conservation d'état)
1	0	0	Q mémorisation de l'information (conservation d'état)
1	0	1	1 mise à 1 de la sortie
1	1	0	0 mise à 0 de la sortie
1	1	1	X état indéterminé

3.2.3. Bascule JK:

Les bascules JK, identiques aux bascules RS pour les entrées autres que 11 (l'entrée J correspond à S et l'entrée K à R). Pour les entrées 11 ces bascules fonctionnent en mode Toggle à savoir que leur sortie Q change de valeur à chaque impulsion d'horloge. Par rapport aux bascules RS, les bascules JK permettent d'utiliser toutes les combinaisons des entrées.



Fonctionnement statique:

Contrairement à la bascule RS, la condition J=K=1, ne donne pas lieu à une condition indéterminée, mais par contre la bascule passe à l'état opposé.

Table de vérité :

J	K	Q	Q ⁺
0	0	0	0 mémorisation de l'information (conservation d'état)
0	0	1	1 mémorisation de l'information (conservation d'état)
0	1	0	0 mise à 0 de la sortie
0	1	1	0 mise à 0 de la sortie
1	0	0	1 mise à 1 de la sortie
1	0	1	1 mise à 1 de la sortie
1	1	0	1 basculement(inverse d'état)
1	1	1	0 basculement(inverse d'état)

La table de vérité condensée

J	K	Q ⁺
0	0	Q conservation de l'état (état mémoire)
0	1	0 mise à 0 ($\forall Q$)
1	0	1 mise à 1 ($\forall Q$)
1	1	\bar{Q} basculement (inverse d'état)

Les équations d'état

$$\begin{aligned}
 Q^+ &= \overline{JK}Q + \overline{JK}\overline{Q} + J\overline{K}Q + JK\overline{Q} \\
 &= \overline{K}Q(J + \overline{J}) + J\overline{Q}(K + \overline{K}) \\
 &= \overline{K}Q + J\overline{Q}
 \end{aligned}$$

Diagrammes d'états ou de transitions

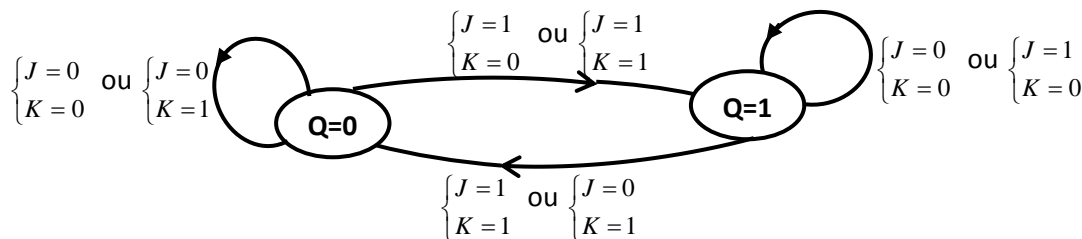
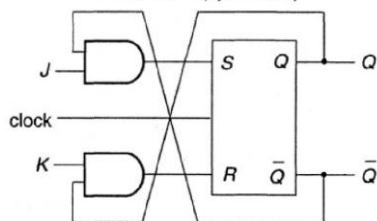


Table de transition d'une bascule JK

On connaît les valeurs des sorties, comment déterminer les valeurs des entrées JK ?

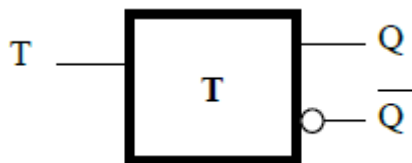
Q	Q+	J	K	
0	0	0	X	mise à 0 ou état mémoire
0	1	1	X	mise à 1 ou basculement
1	0	X	1	mise à 0 ou basculement
1	1	X	0	mise à 1 ou état mémoire

Bistable JK (synchrone)



3.2.4. Bascule T (Trigger flip-flop):

Bascule à déclenchement



Fonctionnement :

si T=1, inversion des états;

si T=0, Q ne change pas.

Table de vérité :

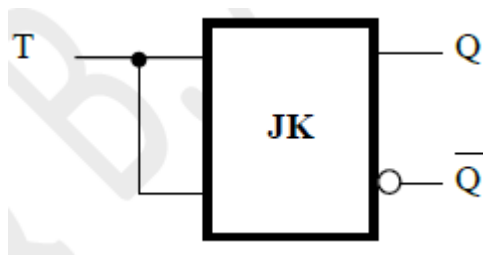
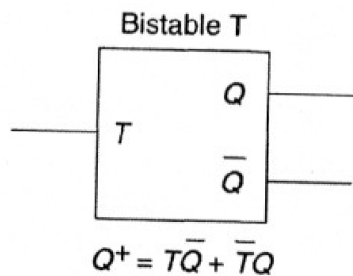
T	Q	Q+
0	0	0 conservation de l'état (état mémoire)
0	1	1 conservation de l'état (état mémoire)
1	0	1 basculement (inverse d'état)
1	1	0 basculement (inverse d'état)

Les équations d'état

$$Q^+ = T\bar{Q} + \bar{T}Q$$

$$= T \oplus Q$$

Réalisation



Remarque : En remplaçant J et K par T dans l'équation de la bascule JK on aura $Q^+ = TQ + \bar{T}\bar{Q} = T \oplus \bar{Q}$

3.2.5. Bascule D (Delay):

Recopie, sur sa sortie Q, le signal d'entrée D

Fonctionnement :

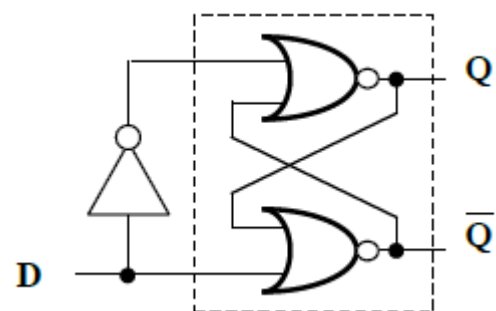
Un appui sur D → Mise à 1 de Q

Un relâchement de D → Mise à 0 de Q si C=1, $Q^+ = Q = D$;

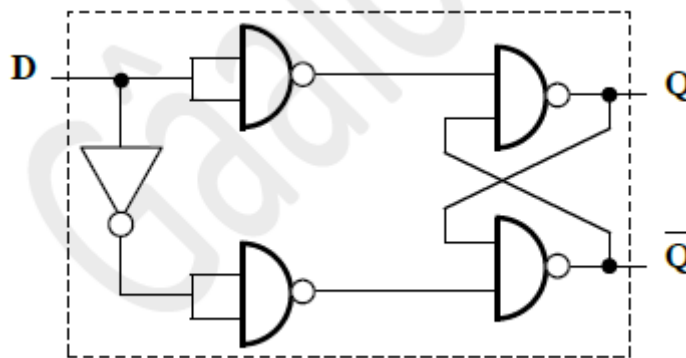
Table de vérité :

D	Q	Q+
0	0	0 mise à 0
0	1	0 mise à 0
1	0	1 mise à 1
1	1	1 mise à 1

Réalisation à l'aide de portes NOR



Réalisation à l'aide de portes NAND



Remarque : En mettant $S = D$ et $R = \bar{D}$ dans l'équation de la bascule RS on aura $Q^+ = D\bar{D}Q + \bar{D}D = D$

Ainsi on obtient une bascule D en rajoutant un inverseur entre S et R.

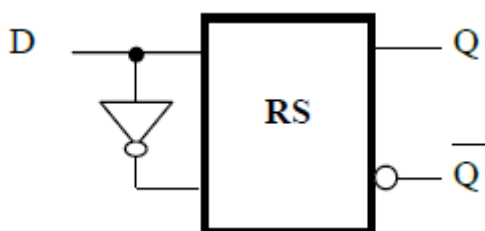


Table de transition d'une bascule D

Q	Q+	D	
0	0	0	mise à 0
0	1	1	mise à 1
1	0	0	mise à 0
1	1	1	mise à 1

4. Utilisation des bascules :

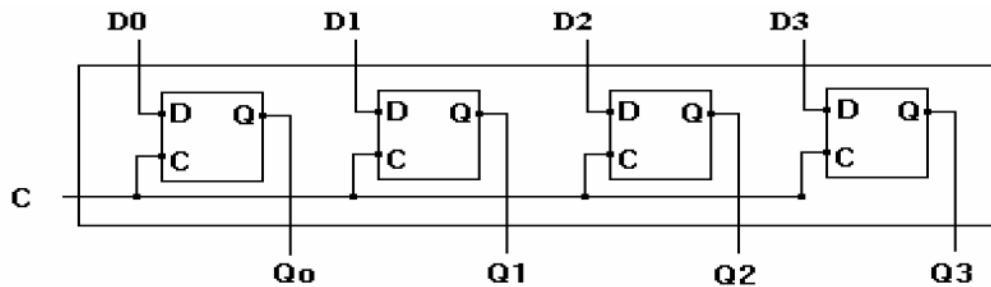
4.1. Utilisation des bascules pour réaliser un registre

a-définition d'un registre:

Une bascule est l'élément de base de la logique séquentielle.

- Une bascule permet de mémoriser un seul bit.
- Un registre est ensemble un ordonné de **n** bascules.
- Un registre permet de mémoriser (sauvegarder) une information sur n bits.

Exemple : Registre à 4 bits

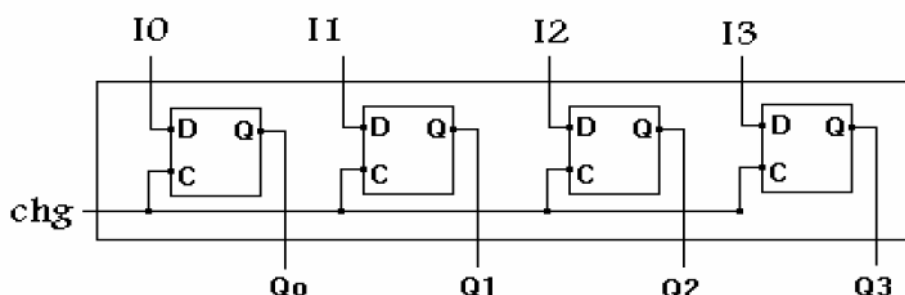


b-Type des registres

- Il existe plusieurs types de registres :
 - Registre à entrées parallèles et sorties parallèles (Registre à chargement parallèle).
 - Registre à entrée série et sortie série
 - Registre à entrée série et sortie parallèle.
 - Registre à entrée parallèle et sortie série.
 - Registre à décalage circulaire.

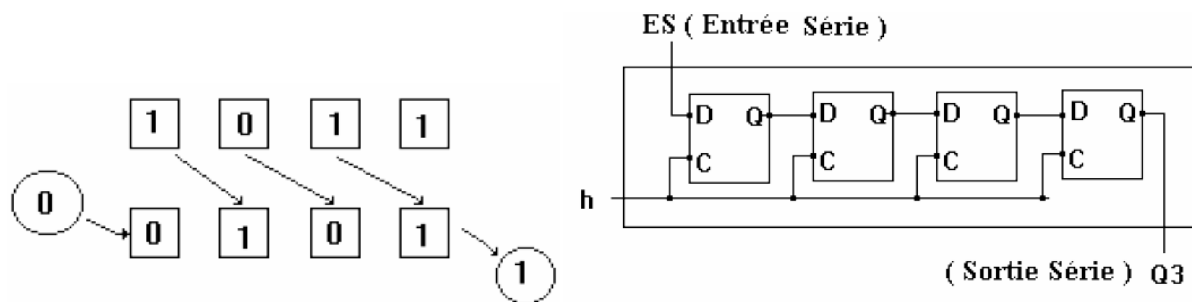
1- Registre à entrées parallèles et sorties parallèles (Registre à chargement parallèle):

- Il peut charger une information sur **N bits** en même temps.
- Les **n** bascules changent d'états en même temps.
- Chaque bascule Bi prend la valeur de l'information i.
- Il possède une entrée de chargement chg (chg=0 état mémoire, chg=1 chargement)

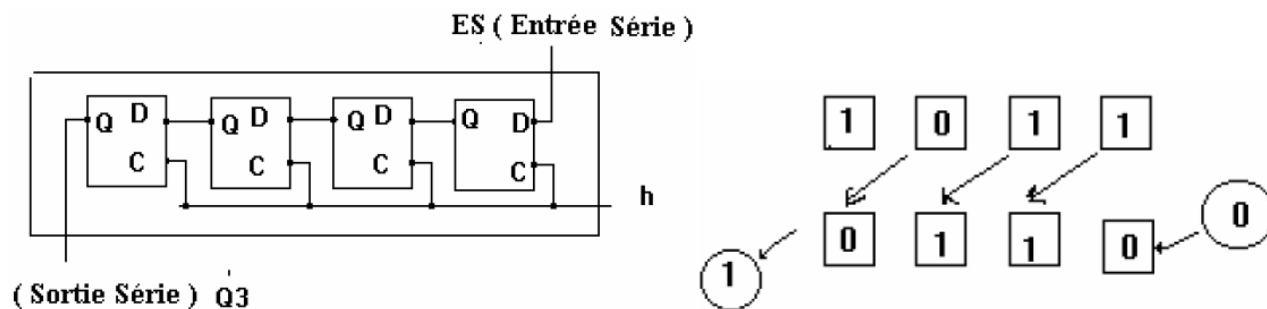


2- Registre à entrée série et sortie série:

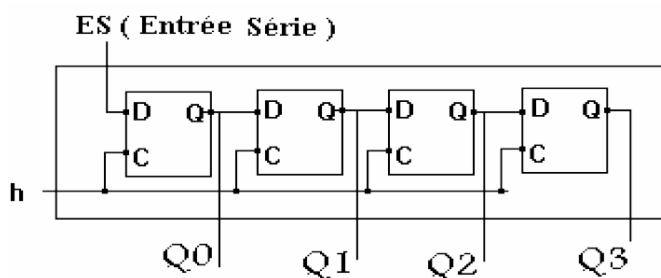
- L'information est introduite bit par bit (en série).
- L'ensemble du registre est décalé d'une position (B_i, B_{i+1}) et la bascule B_0 reçoit une nouvelle entrée E_s .
- Un tel registre est appelé registre à entrée série à gauche et à sortie série à droite.



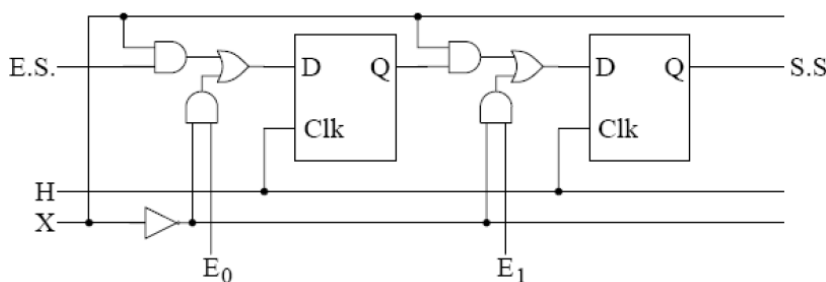
3- registre à entrée série à droite et à sortie série à gauche:



4- Registre à entrée série et sortie parallèle:

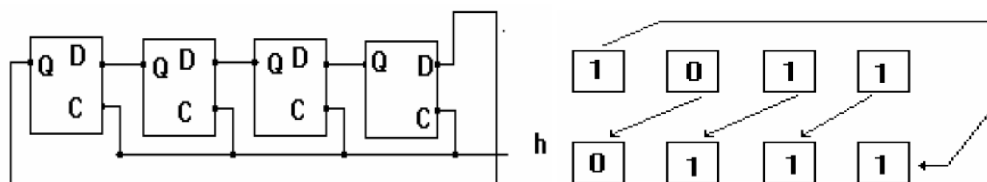


5- Registre à entrée parallèle et sortie série:

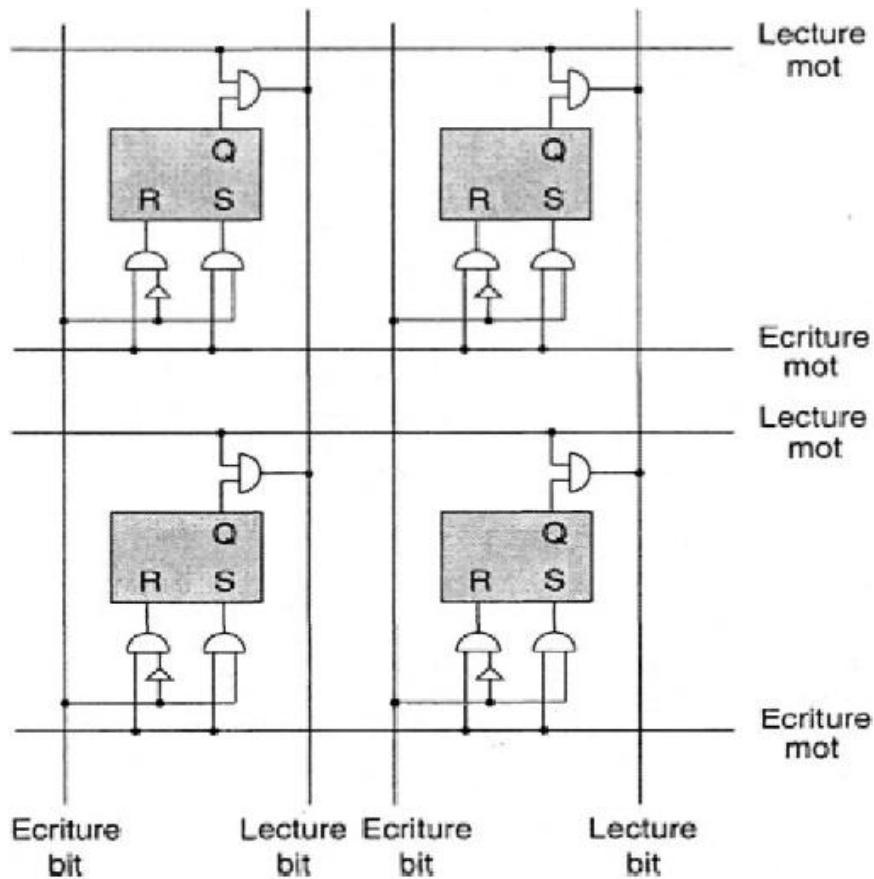


6- Registre à décalage circulaire:

- C'est un registre qui effectue un décalage vers la gauche en répercutant la sortie de la dernière bascule vers l'entrée de la dernière bascule.
- Le décalage peut être un décalage droite (circulaire droite) ou gauche (circulaire gauche)



4.2. Utilisation des bascules pour la mémoire centrale



4.3. Utilisation des bascules pour réaliser des compteurs

4.3.1. C'est quoi un compteur ?

- Un compteur est un circuit séquentiel qui possède **N états** (E_0, E_1, \dots, E_{n-1}).
- À chaque top d'horloge, il passe de l'état E_i à l'état E_{i+1} .
- Il revient toujours à l'état initiale E_0 : Un compteur possède un **cycle (une séquence d'états)**.
- Un compteur est constitué de **n** bascules.
- Le nombre d'états d'un compteur est inférieur ou égale à 2^n .
- Il existe deux types de compteurs : les compteurs synchrones et asynchrones.
 - les bascules qui constituent un compteur synchrone possèdent la **même horloge**.
 - les bascules qui constituent un compteur asynchrone possèdent des **horloges différentes**.

Q2	Q1	Q0	Q2+	Q1+	Q0+
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

4.3.2. Types de compteurs

• Les compteurs modulo 2^n (cycle complet)

- n=2 : 0,1,2,3,0 → modulo 4
- n=3 : 0,1,2,3,4,5,6,7,0 → modulo 8
- n=4 : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,0 → modulo 16

• Les compteurs modulo N (cycle incomplet)

- Pour N=5 : 0,1,2,3,4,0 → modulo 5
- Pour N= 10 : 0,1,2,3,4,5,6,7,8,9,0 → modulo 10

• Les compteurs à cycle quelconque : → 0,2,5,6,7,8,10,0

4.3.3. Étude des compteurs modulo 2^n :

Exemple Modulo $2^3=8$ (bascules JK)

- On remarque que Q_0 passe à chaque fois de 0 à 1 et de 1 à 0 : **un basculement** → $J_0=K_0=1$.
- On remarque que Q_1 passe de 1 à 0 et passe de 1 à 0 à chaque fois que Q_0 passe de 1 à 0 → Q_0 est utilisée comme une **horloge** pour Q_1 avec $J_1=K_1=1$ pour avoir un basculement.
- On remarque que Q_2 passe de 1 à 0 et passe de 1 à 0 à chaque fois que Q_1 passe de 1 à 0 → Q_1 est utilisée comme une **horloge** pour Q_2 avec $J_2=K_2=1$ pour avoir un basculement.

Ce compteur est appelé un compteur **ASYNCHRONE** : chaque bascule possède sa propre horloge.

Diagramme d'état :

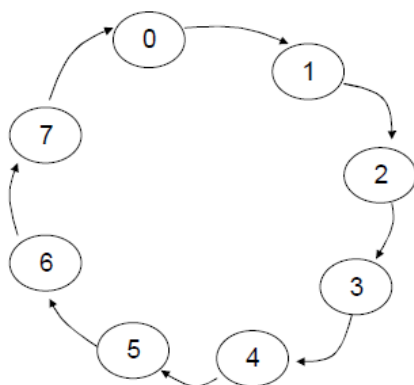


Table d'état

Q2	Q1	Q0	Q2 ⁺	Q1 ⁺	Q0 ⁺
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Chronogramme d'un compteur asynchrone modulo 8

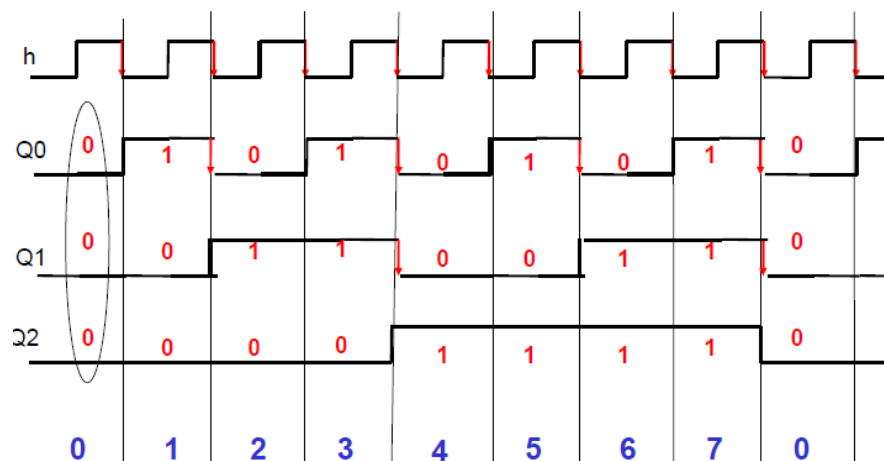
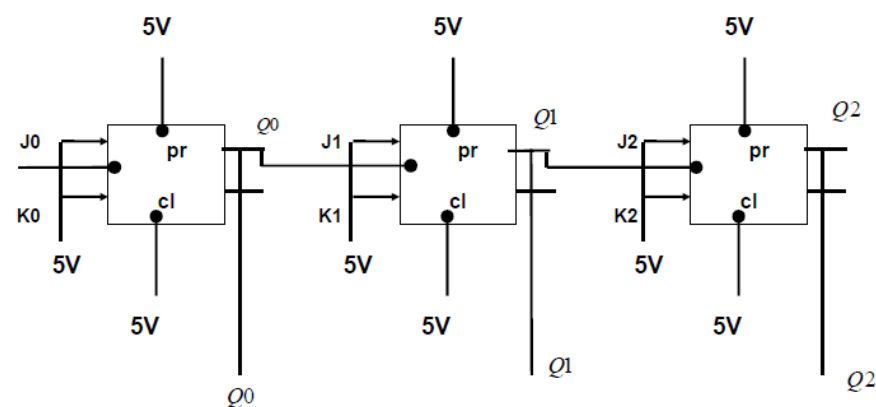
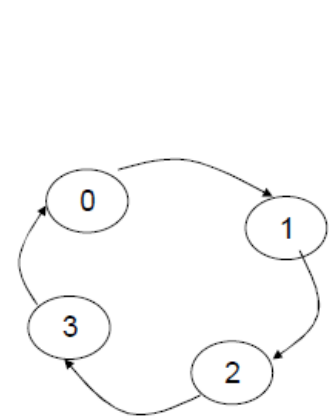


Schéma d'un compteur asynchrone modulo 8



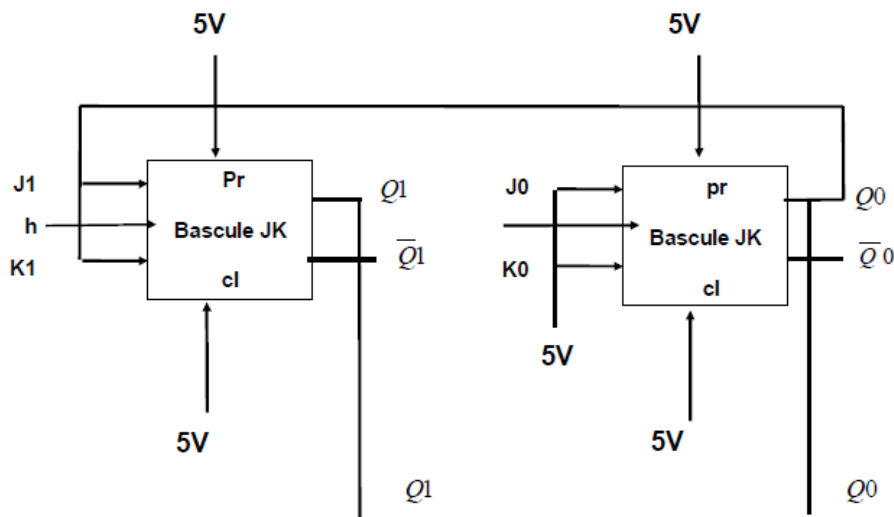
Utilisation de bascules pour créer un compteur à 3 bits

exemple : Réaliser le circuit qui permet de réaliser le cycle suivant 0,1,2,3 à l'aide de bascules JK?



Q1	Q0	J1	K1	J0	K0	Q1+	Q0+
0	0	0	X	1	X	0	1
0	1	1	X	X	1	1	0
1	0	X	0	1	X	1	1
1	1	X	1	X	1	0	0

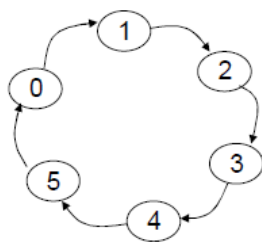
J0=K0=1
J1=K1=Q0



4.3.4. Études des Compteurs modulo N:

Exemple : compteur modulo 6

Q2	Q1	Q0	Q2 ⁺	Q1 ₊	Q0 ₊	Pri	Cli
0	0	0	0	0	1	1	1
0	0	1	0	1	0	1	1
0	1	0	0	1	1	1	1
0	1	1	1	0	0	1	1
1	0	0	1	0	1	1	1
1	0	1	1	1	0	1	1
1	1	0	0	0	0	1	0
1	1	1	0	0	0	1	1



$$\overline{Cli} = Q2.Q1.\overline{Q0}$$

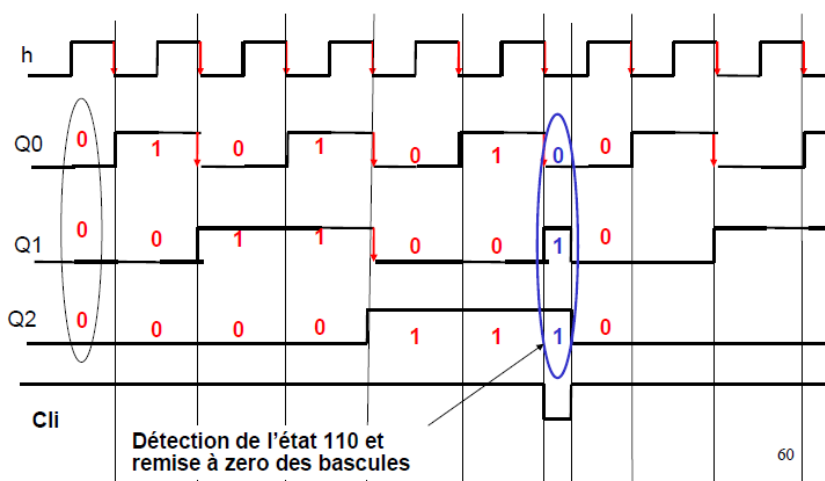
$$Cli = \overline{Q2.Q1.Q0}$$

État inaccessible

Détection de l'état 110 et remise à zero asynchrone : Cli=0 et Pri=1

59

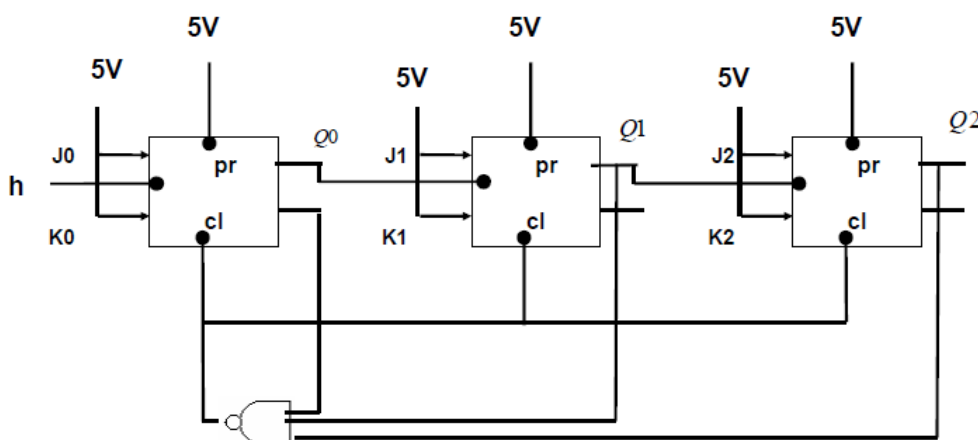
Chronogramme d'un compteur modulo 6 asynchrone



Détection de l'état 110 et remise à zero des bascules

60

Schéma d'un compteur modulo 6 asynchrone



4.3.5. Étude des compteurs synchrones modulo 2ⁿ :

Exemple : compteur Modulo 8 (bascule JK)

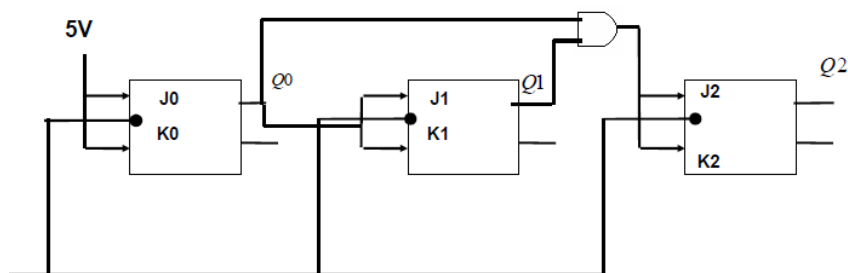
Toutes les bascules possèdent la même horloge.

Pour réaliser le compteur il faut **déterminer** les équations des Ji et Ki ?

Q2	Q1	Q0	Q2+	Q1+	Q0+	J2	K2	J1	K1	J0	K0
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1

Schéma d'un compteur modulo 8 synchrone

D'après la table de vérité on trouve : $J_0=K_0=1$, $J_1=K_1=Q_0$, $J_2=K_2=Q_0.Q_1$



h

Exemple : compteur synchrone modulo 2ⁿ exemple Modulo 8 (bascule T)

Une bascule T possède deux états : mémoire si T=0 et basculement si T=1.

Q2	Q1	Q0	Q2+	Q1+	Q0+	T2	T1	T0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

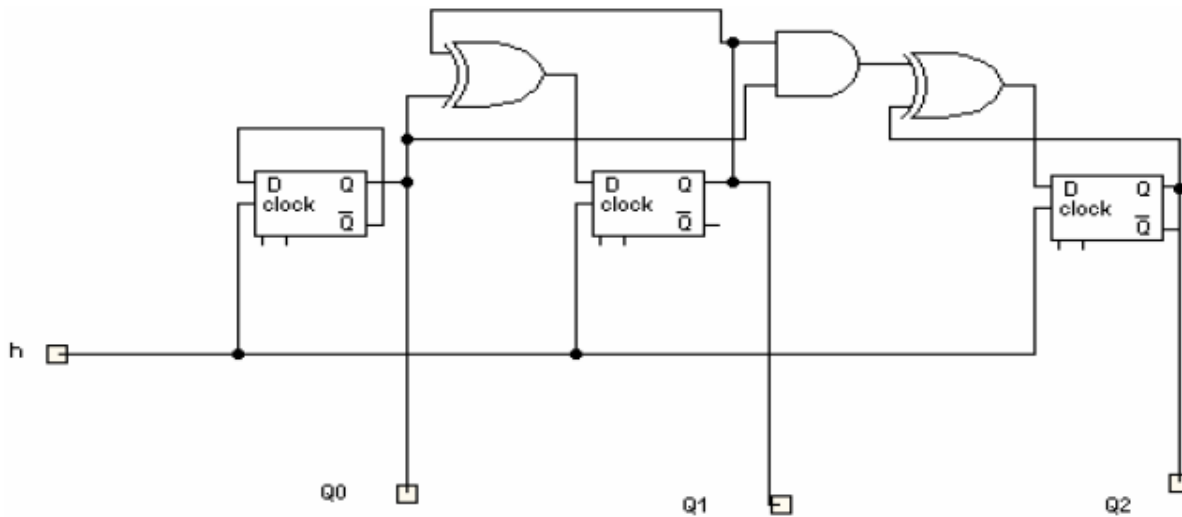
$T0=1$
 $T1=Q0$
 $T2=Q0.Q1$

Exemple : compteur synchrone modulo 2ⁿ exemple Modulo 8 (bascule D)

Q2	Q1	Q0	Q2+	Q1+	Q0+	D2	D1	D0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

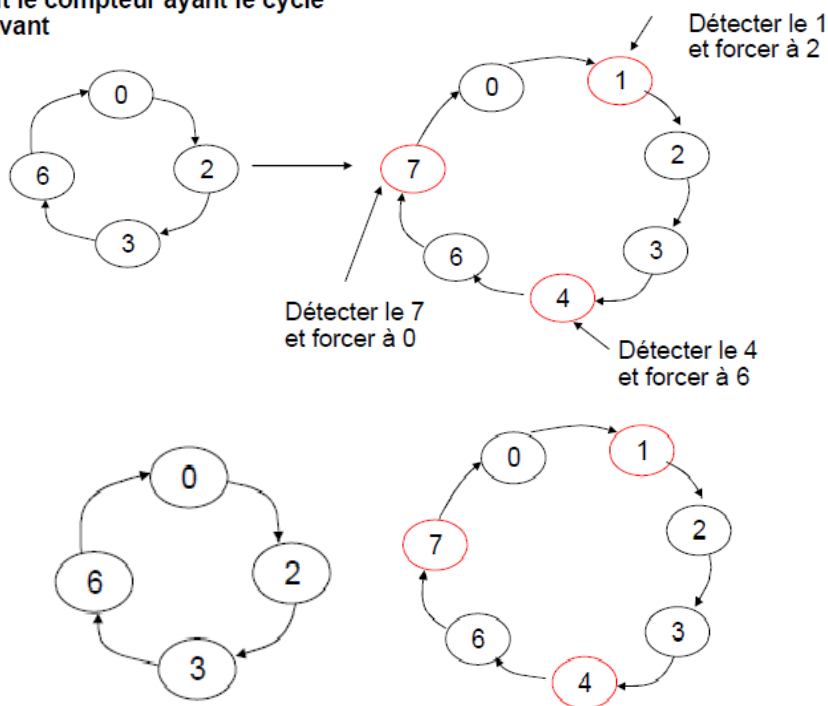
$D0 = \overline{Q0}$
 $D1 = Q1 \oplus Q0$
 $D2 = Q2 \oplus (Q1.Q0)$

Schéma d'un compteur modulo 8 synchrone avec des bascules D



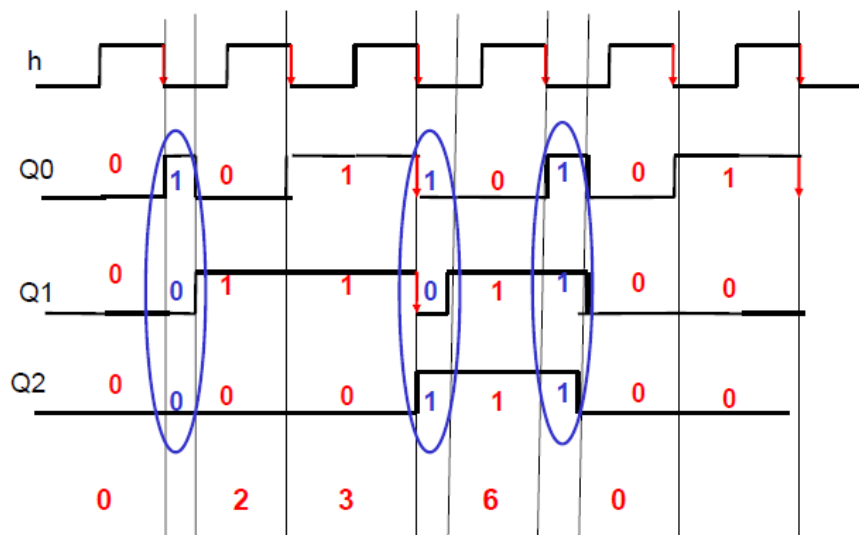
4.3.6. Les compteurs asynchrones à cycle quelconque

Soit le compteur ayant le cycle suivant



Pour forcer le compteur d'un état à un autre il faut agir sur les entrées asynchrones Cli et Pri des bascules.

Chronogramme d'un compteur asynchrone à cycle quelconque



Exemple d'un compteur asynchrone à cycle quelconque

Q2	Q1	Q0	Q2*	Q1*	Q0*	Pr2	Cl2	Pr1	Cl1	Pr0	Cl0
0	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	0
0	1	0	0	1	1	1	1	1	1	1	1
0	1	1	1	0	0	1	1	1	1	1	1
1	0	0	1	1	0	1	1	0	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	1	0	1	0	1	0

$Pr\ 2 = 1$

$Pr\ 1 = \overline{\overline{Q2.Q1.Q0}} + \overline{Q2.Q1.Q0}$

$Pr\ 0 = 1$

$Cl\ 2 = \overline{Q2.Q1.Q0}$

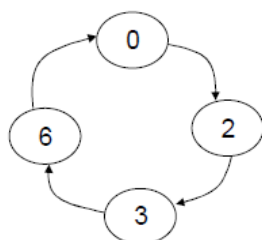
$Cl\ 1 = \overline{Q2.Q1.Q0}$

$Cl\ 0 = \overline{\overline{Q2.Q1.Q0}} + \overline{Q2.Q1.Q0}$

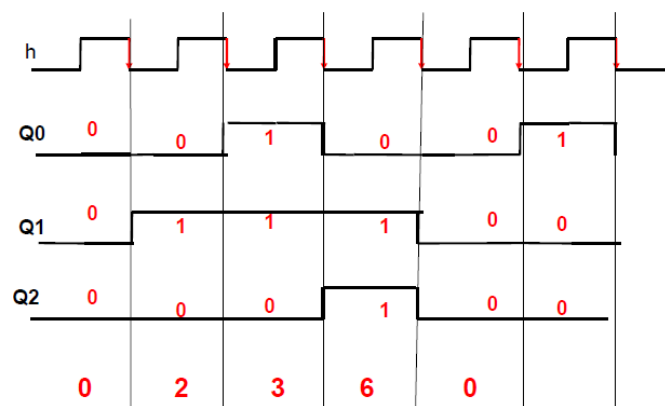
4.3.7. Les compteurs synchrones à cycle quelconque

Soit le compteur ayant le cycle suivant

- Pour forcer le compteur d'un état à un autre il faut agir sur les entrées synchrones Ji et Ki
- Pour les états qui n'appartiennent pas au cycle du compteur il faut les considérer comme étant des états indéterminés (Ji=X et Ki=X)



Chronogramme d'un compteur synchrone à cycle quelconque



Réalisation d'un compteur à cycle quelconque cas synchrone avec des bascules J.K

Q2	Q1	Q0	Q2*	Q1*	Q0*	J2	K2	J1	K1	J0	K0
0	0	0	0	1	0	0	X	1	X	0	X
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	1	0	1	X	X	0	X	1
1	1	0	0	0	0	X	1	X	1	0	X
0	0	1	X	X	X	X	X	X	X	X	X
1	0	0	X	X	X	X	X	X	X	X	X
1	0	1	X	X	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X	X	X

$J2 = K2 = Q1.(Q2 \oplus Q0)$

$J1 = K1 = \overline{Q0}.(Q2 \oplus Q1)$

$J0 = K0 = \overline{Q2}.Q1$

Exemple : Réalisation d'un compteur synchrone à cycle quelconque avec des bascules T

Q2	Q1	Q0	Q2 ⁺	Q1 ⁺	Q0 ⁺	T2	T1	T0
0	0	0	0	1	0	0	1	0
0	1	0	0	1	1	0	0	1
0	1	1	1	1	0	1	0	1
1	1	0	0	0	0	1	1	0
0	0	1	X	X	X	X	X	X
1	0	0	X	X	X	X	X	X
1	0	1	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X

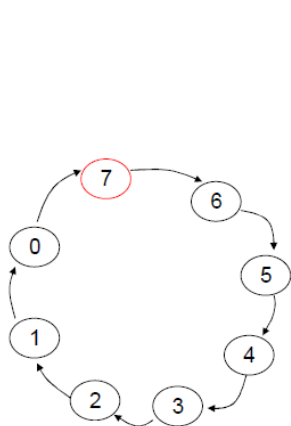
$T2 = Q1.(Q2 \oplus Q0)$

$T1 = \overline{Q0}.(Q2 \oplus Q1)$

$T0 = \overline{Q2}.Q1$

4.4. Utilisation des bascules pour réaliser Les décompteurs :

L'étude des décompteurs se fait exactement de la même manière que l'étude des compteurs. exemple :



Q2	Q1	Q0	Q2 ⁺	Q1 ⁺	Q0 ⁺	Q2	Q1	Q0	T2	T1	T0
1	1	1	1	1	0	1	1	1	0	0	1
1	1	0	1	0	1	1	1	0	0	1	1
1	0	1	1	0	0	1	0	1	0	0	1
1	0	0	0	1	1	1	0	0	1	1	1
0	1	1	0	1	0	0	1	1	0	0	1
0	1	0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	0	0	0	1	0	0	1
0	0	0	1	1	1	0	0	0	1	1	1

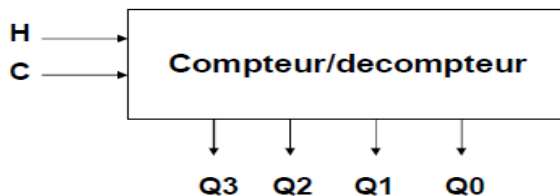
$T0 = J0 = K0 = 1$

$T1 = J1 = K1 = \overline{Q0}$

$T2 = J2 = K2 = \overline{Q0}.Q1$

4.5. Utilisation des bascules pour réaliser Les compteurs/décompteurs:

- Le même circuit peut offrir l'opération de comptage et décomptage
- Rajouter une entrée C, si C=0 alors comptage, si C=1 alors décomptage



Exemple d'un compteur/décompteur :

C	Q2	Q1	Q0	T2	T1	T0
0	0	0	0	0	0	1
0	0	0	1	0	1	1
0	0	1	0	0	0	1
0	0	1	1	1	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	1
0	1	1	0	0	0	1
0	1	1	1	1	1	1
1	1	1	1	0	0	1
1	1	1	0	0	1	1
1	1	0	1	0	0	1
1	1	0	0	1	1	1
1	0	1	1	0	0	1
1	0	1	0	0	1	1
1	0	0	1	0	0	1
1	0	0	0	1	1	1

$T0 = J0 = K0 = 1$

$T1 = J1 = K1 = \overline{C}.Q0 + C.\overline{Q0}$

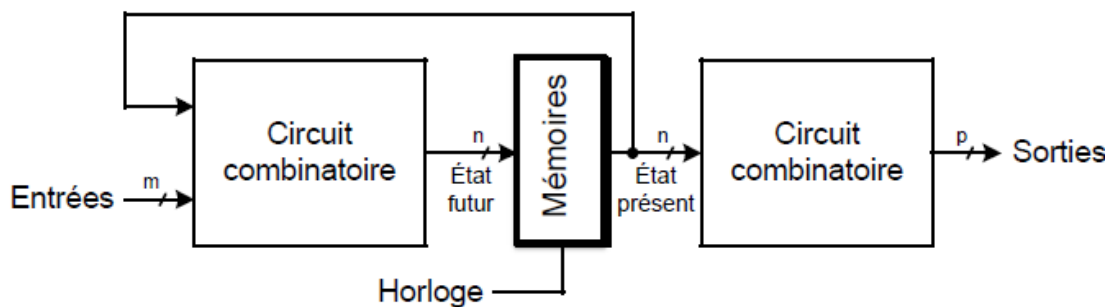
$T2 = J2 = K2 = \overline{C}.Q0.Q1 + C.\overline{Q0}.Q1$

5. MSA (machine séquentielle algorithmique)

5.1. Définition de la machine séquentielle algorithmique

Circuit séquentiel qui peut se retrouver dans un nombre fini d'états. Une fois dans un état, le système doit demeurer dans cet état pour une durée déterminée.

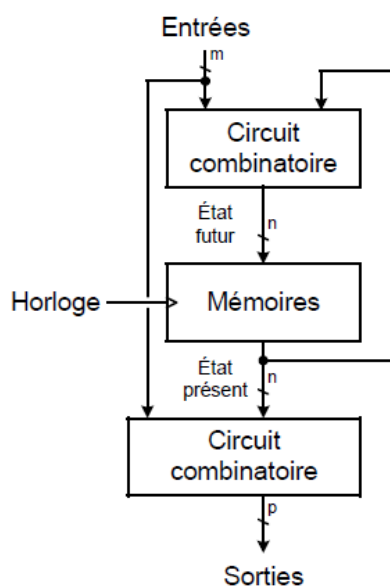
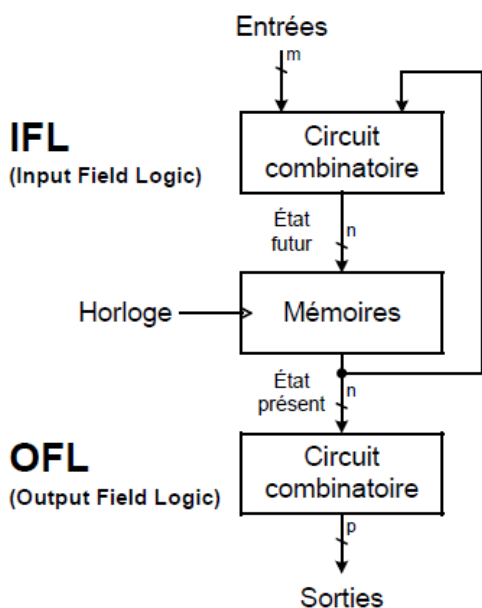
- **Autres appellations:**
 - ✓ Machine à états
 - ✓ Machine à états finis
 - ✓ Séquenceur
 - ✓ Contrôleur
 - ✓ Automate
 - ✓ Machine séquentielle
 - ✓ Machine séquentielle algorithmique.
- **Catégories de MSA**
 - ✓ Synchrone
 - ✓ Principalement synchrone
 - ✓ Asynchrone
- **Structure des machines à états finis**



- **Classes de MSA**

Moore

Mealy



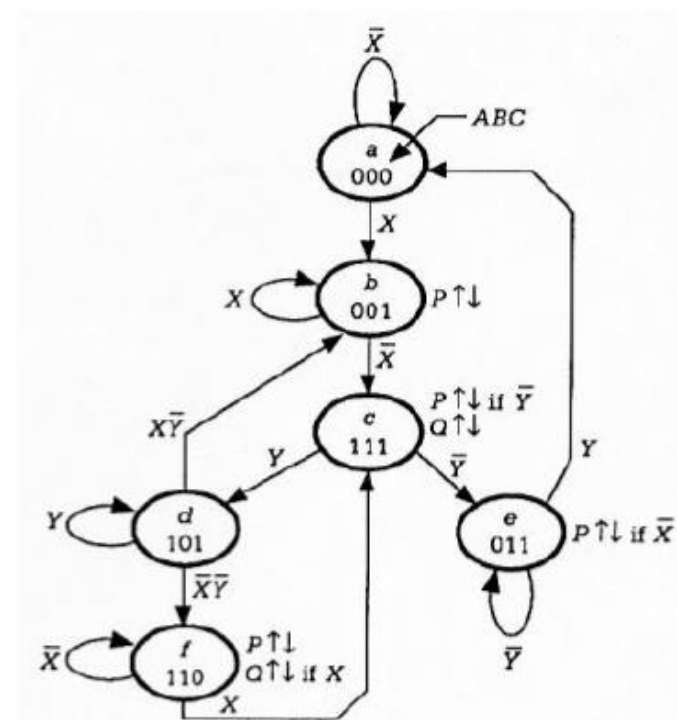
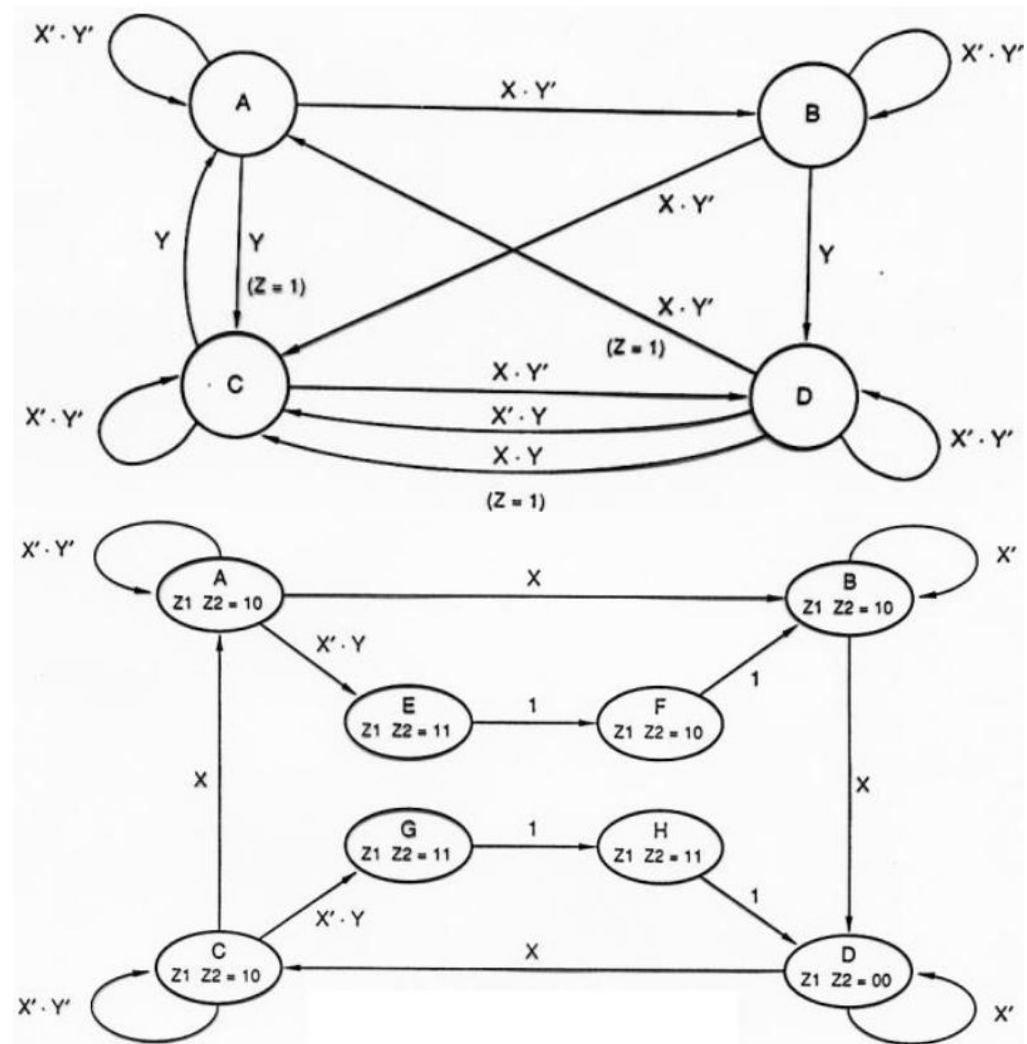
- **Nombre d'états**

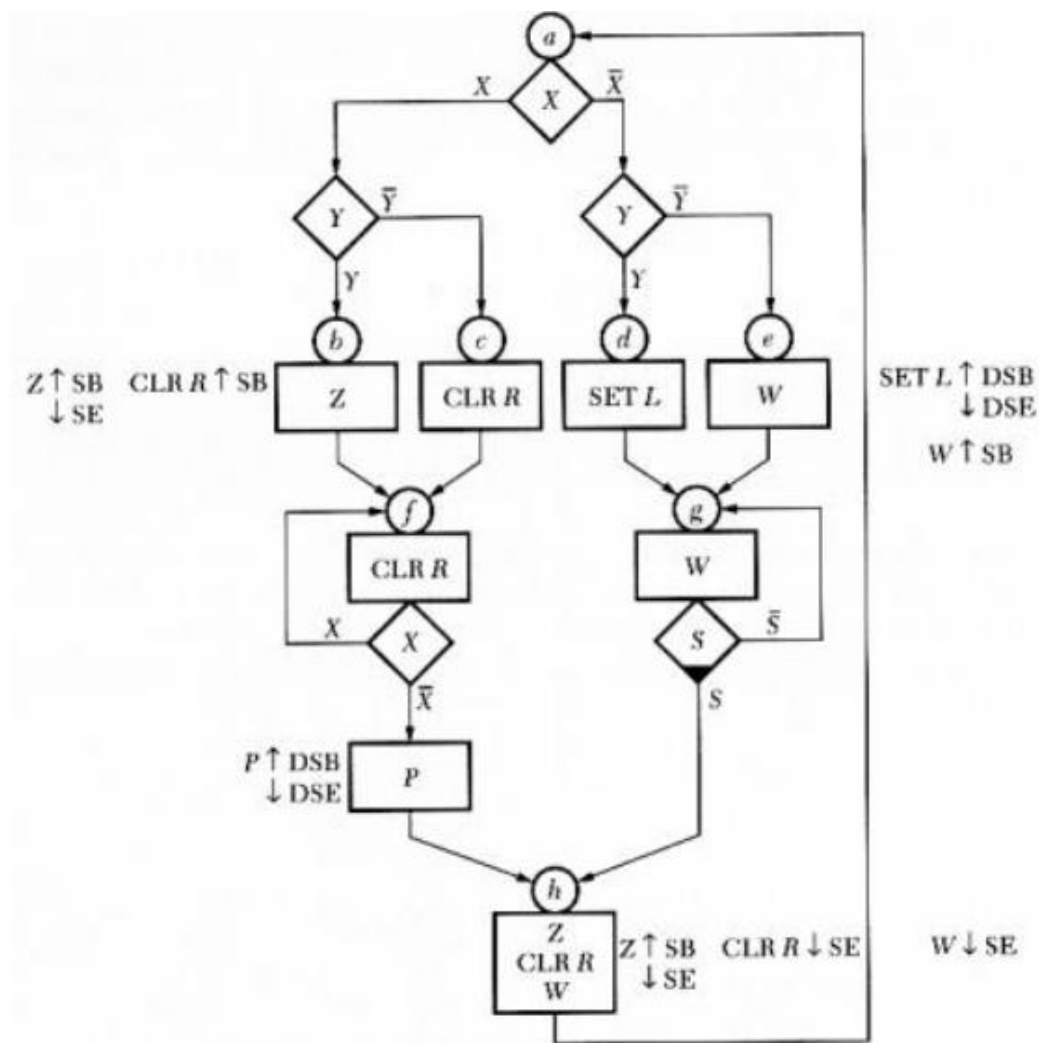
Le nombre d'états d'une MSA est limité par le nombre de bits de mémoire.

Nombre maximal d'états = 2^n où n est le nombre de bits de mémoire, Pour les machines ayant de multiples interactions entre les états, il est conseillé de limiter à 8 états (Loi de Miller).

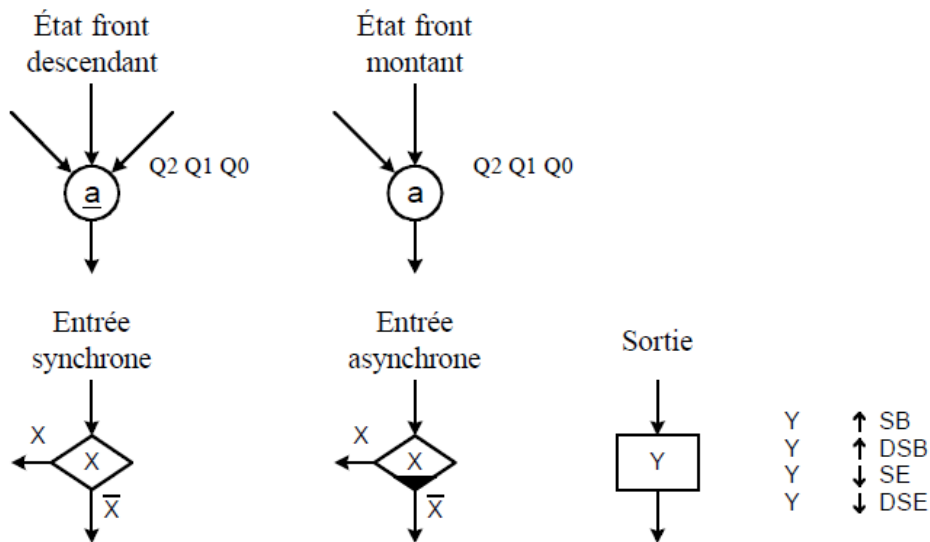
▪ Représentation de diagramme d'états

Exemples de diagramme d'états





Symboles utilisés dans les diagrammes d'états (Représentation Comer)



▪ Types de bascules

✓ Propriétés des bascules

D	Q⁺	S	R	Q⁺	T	Q⁺	J	K	Q⁺
0	0	0	0	Q	0	Q	0	0	Q
0	1	0	1	0	0	Q	0	1	0
1	1	1	0	1	1	\bar{Q}	1	0	1
		1	1	?			1	1	\bar{Q}

✓ **Équations caractéristiques des bascules**

- Bascule RS $Q^+ = S + RQ$
- Bascule JK $Q^+ = J\bar{Q} + \bar{K}Q$
- Bascule T $Q^+ = T \oplus Q$
- Bascule D $Q^+ = D$

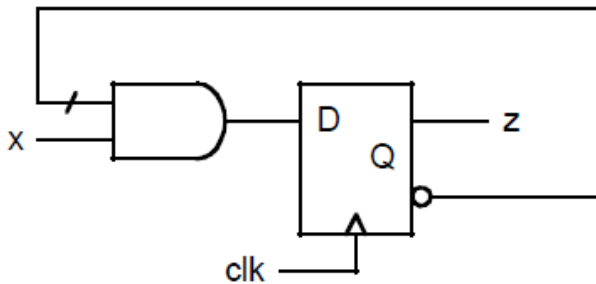
5.2. Analyse de machines à états finis

L'analyse et la qualification de machines séquentielles algorithmiques se fait en 8 étapes:

1. Identifier et évaluer les équations d'excitation (sorties de l'étage IFL);
2. Identifier et évaluer les fonctions de sortie (équations de l'étage OFL);
3. Construire la table d'excitation (sorties de l'IFL);
4. Compléter la table de transition (état suivant);
5. Compléter la table des sorties;
6. Réaliser une table d'états et de sorties;
7. Construire le diagramme d'états;

Représenter la MSA à l'aide de son diagramme temporel (chronogramme).

Exemple 1: MSA à deux états



1. Équations de la logique d'entrée (étage IFL) :

$$D = \bar{Q}X$$

2. Équations de la logique de sortie (étage OFL)

$$Z = Q$$

3. Table d'excitation (sorties de l'IFL)

Q	X	D
0	0	0
0	1	1
1	0	0
1	1	0

$$D = \bar{Q}X$$

4. Table de transition (prochain état)

Q	X	Q ⁺
0	0	0
0	1	1
1	0	0
1	1	0

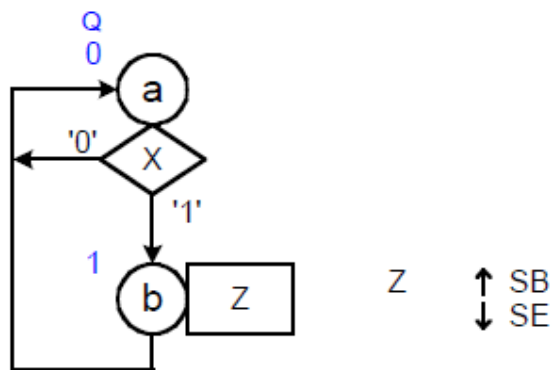
5. Table des sorties

Q	X	Z
0	0	0
0	1	0
1	0	1
1	1	1

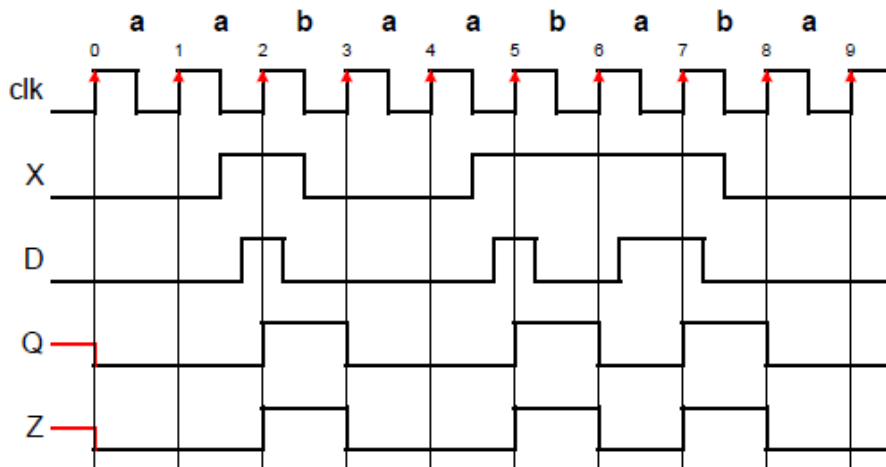
6. Table des états et sorties

Q	0	1
a		b*

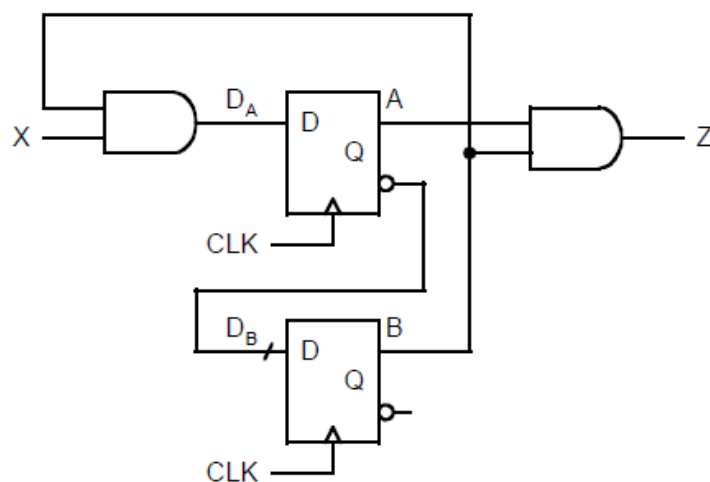
7. Diagramme d'états



8. Diagramme temporel (chronogramme)



Exemple 2



1. Équations de la logique d'entrée (étage IFL) :

$$D_A = BX$$

2. Équations de la logique de sortie (étage OFL) : $D_B = \bar{A}$

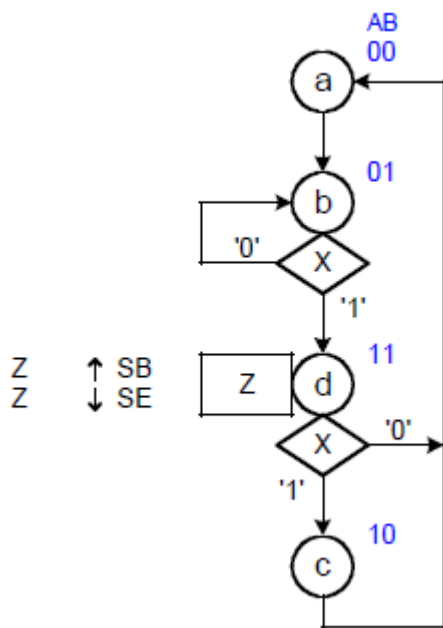
3. et 4. et 5. Table d'excitation, de transition et de sortie

A	B	X	D _A	D _B	A*	B*	Z
0	0	0	0	1	0	1	0
0	0	1	0	1	0	1	0
0	1	0	0	1	0	1	0
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	1
1	1	1	1	0	1	0	1

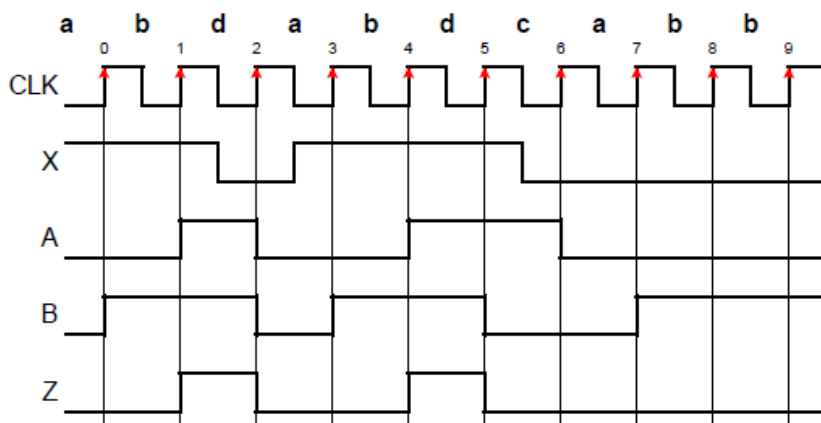
6. Table des états et sorties

A \ B	0	1
0	a	c
1	b	d*

7. Diagramme d'états



8. Diagramme de phase (chronogramme)



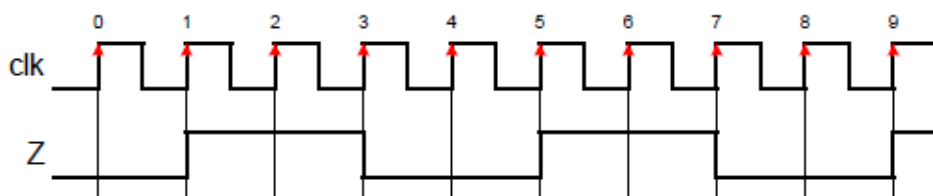
5.3. Synthèse de machines séquentielles

La synthèse de MSA utilise principalement les mêmes étapes que pour l'analyse mais dans l'ordre inverse:

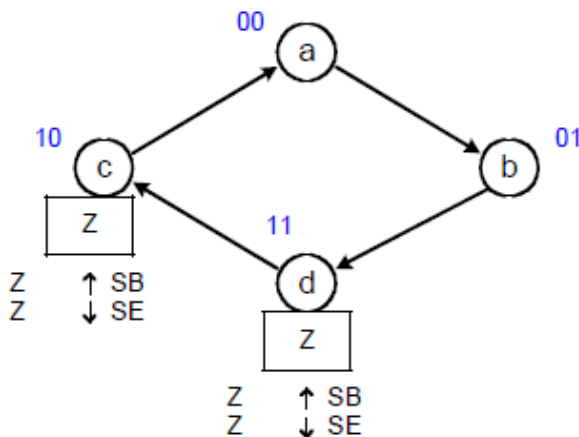
1. Préparer le diagramme de phase (chronogramme);
2. Construire le diagramme d'états;
3. Réaliser une table d'états et de sorties ;
4. Compléter la table de transition (état suivant);
5. Construire la table d'excitation (sorties de l'IFL);
6. Compléter la table des sorties;
7. Extraire les équations d'excitation (sorties de l'étage IFL);
8. Évaluer les fonctions de sortie (équations de l'étage OFL);
9. Dessiner le schéma logique (logigramme);
10. Vérifier le problème d'impasse.

Exemple 1: Diviseur par quatre synchrone (CLK/4)

1. Diagramme de phase (chronogramme)



2. Diagramme d'états



3. Table d'états et de sorties

	Q_1	0	1
Q_0	0	a	c*
	1	b	d*

4. & 5. & 6. Table d'excitation, de transition et de sorties

	Q_1	Q_0	Q_1^+	Q_0^+	D_1	D_0	Z
a	0	0	0	1	0	1	0
b	0	1	1	1	1	1	0
c	1	0	0	0	0	0	1
d	1	1	1	0	1	0	1

7. Équations d'excitation (étage IFL)

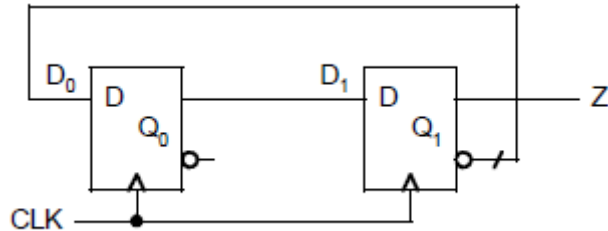
$$D_1 = Q_0$$

$$D_0 = \overline{Q_1}$$

8. Fonctions de sortie (étage OFL)

$$Z = Q_1$$

9. Schéma logique



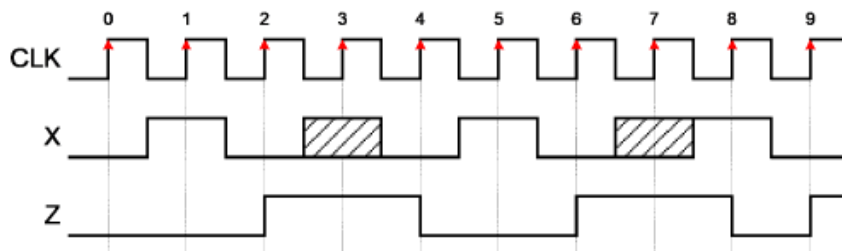
10. Vérification des impasses

Aucune impasse: 2 bits d'états et 4 états utilisés

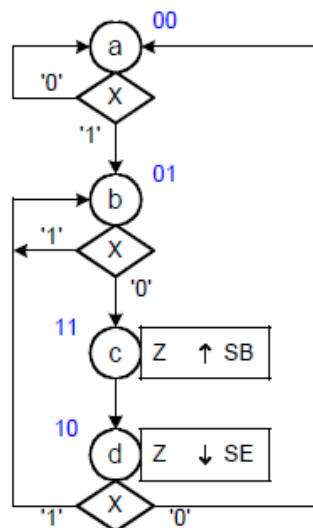
Exemple 2: Détection de front descendant d'un signal

Nous voulons concevoir un système logique permettant de détecter le front descendant sur une entrée X. Une sortie Z sera générée pour deux périodes d'horloge lorsque la transition sera détectée.

1. Diagramme de phase (chronogramme):



2. Diagramme d'états



3. Table d'états et de sorties

$Q_1 \backslash Q_0$	0	1
0	a	d*
1	b	c*

4. & 5. & 6. Table d'excitation, de transition et de sorties (Bascules D)

$Q_1 Q_0$	X	$Q_1^+ Q_0^+$ $D_1 D_0$	Z
a $\begin{cases} 0 0 \\ 0 0 \end{cases}$	0	a 0 0	0
	1	b 0 1	0
b $\begin{cases} 0 1 \\ 0 1 \end{cases}$	0	c 1 1	0
	1	b 0 1	0
d $\begin{cases} 1 0 \\ 1 0 \end{cases}$	0	a 0 0	1
	1	b 0 1	1
c $\begin{cases} 1 1 \\ 1 1 \end{cases}$	0	d 1 0	1
	1	d 1 0	1

4. & 5. & 6. Table d'excitation, de transition et de sorties (Bascules J-K)

J	K	Q^+
0	0	Q
0	1	0
1	0	1
1	1	\bar{Q}

$$Q^+ = J\bar{Q} + \bar{K}Q$$

Q	Q^+	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

$Q_1 Q_0$	X	$Q_1^+ Q_0^+$	$J_1 K_1$	$J_0 K_0$	Z
a $\begin{cases} 0 0 \\ 0 0 \end{cases}$	0	a 0 0	0 -	0 -	0
	1	b 0 1	0 -	1 -	0
b $\begin{cases} 0 1 \\ 0 1 \end{cases}$	0	c 1 1	1 -	- 0	0
	1	b 0 1	0 -	- 0	0
d $\begin{cases} 1 0 \\ 1 0 \end{cases}$	0	a 0 0	- 1	0 -	1
	1	b 0 1	- 1	1 -	1
c $\begin{cases} 1 1 \\ 1 1 \end{cases}$	0	d 1 0	- 0	- 1	1
	1	d 1 0	- 0	- 1	1

J	K	Q^+
0	0	Q
0	1	0
1	0	1
1	1	\bar{Q}

$$Q^+ = J\bar{Q} + \bar{K}Q$$

7. Équations d'excitation (étage IFL)

Basculés D

$$D_1 = Q_0 \bar{X} + Q_1 Q_0$$

$$D_0 = \bar{Q}_0 X + \bar{Q}_1 Q_0$$

Basculés J-K

$$J_1 = Q_0 \bar{X}$$

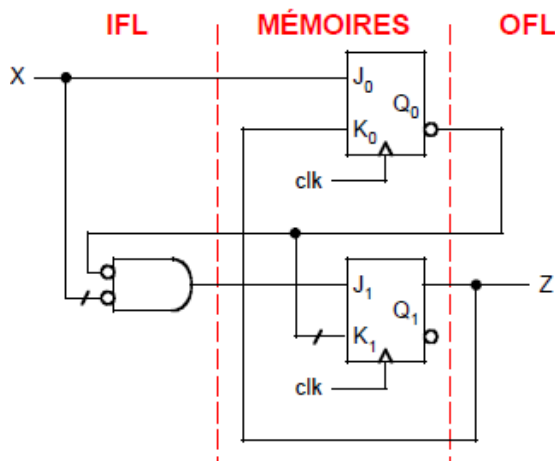
$$K_1 = \bar{Q}_0$$

$$J_0 = X$$

$$K_0 = Q_1$$

$$Z = Q_1$$

8. Fonctions de sortie (étage OFL)



9. Schéma logique

10. Vérification des impasses : Aucune impasse: 2 bits d'états et 4 états utilisés.