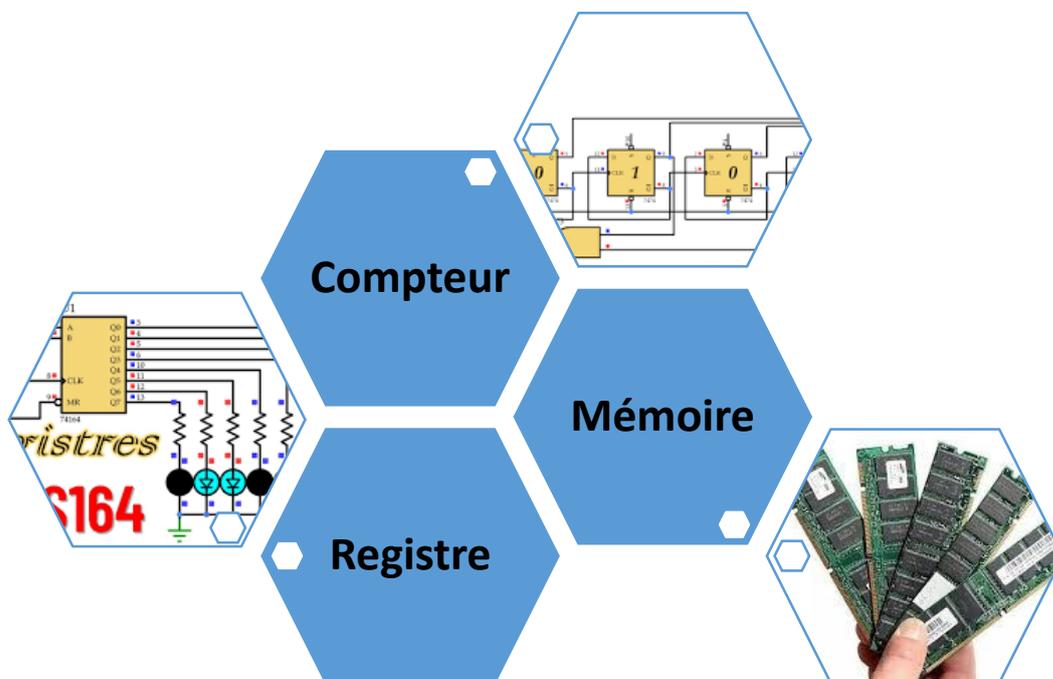


REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique  
Université Djilali Bounaâma de Khemis Miliana  
Faculté des sciences et techniques  
Département de maths et informatique  
Niveau : Licence MI



Rédigé par : Dr. MAHROUG RABIAA  
E-mail : r.mahroug@univ-dbkm.dz

# Chapitre III : Conception des Circuits Séquentiels



Année universitaire 2022-2023

# Table des matières

Table des matières.....	i
Abréviation.....	iii
Chapitre III : conception des Circuits Séquentiels .....	1
3.1. Introduction.....	1
3.2. Système séquentiels synchrone et asynchrones.....	1
3.3. Les bascule .....	3
3.3.1. Définition d'une bascule .....	3
3.3.2. Les types des bascules .....	3
3.3.2.1. Bascule RS :( Reset_set) .....	3
3.3.2.2. Bascules RST .....	4
3.3.2.3. Bascule JK.....	6
3.3.2.4. Bascule T (Trigger flip-flop) .....	7
3.3.2.5. Bascule D (Delay).....	8
3.4. Utilisation des bascules.....	9
3.4.1. Utilisation des bascules pour réaliser un registre .....	9
3.4.1.1. Définition d'un registre.....	9
3.4.1.2. Fonctionnement d'un registre .....	10
3.4.1.3. Type des registres .....	10
3.4.2. Utilisation des bascules pour la mémoire centrale .....	14
3.4.2.1. Définition d'une mémoire.....	14
3.4.2.2. Différents types de la mémoire.....	14
3.4.2.3. Caractéristiques d'une mémoire.....	18
3.4.3. Utilisation des bascules pour réaliser des compteurs.....	18
3.4.3.1. C'est quoi un compteur ?.....	18
3.4.3.2. Types de compteur .....	19
3.4.4. Utilisation des bascules pour réaliser Les décompteurs.....	31
3.4.5. Utilisation des bascules pour réaliser Les compteurs/décompteurs.....	32
3.5. Synthèse de circuits séquentiels.....	35
3.5.1. Définition d'une machine à états finis.....	35
3.5.1.1. Tables de transitions .....	36
3.5.1.2. Diagramme d'état .....	36
3.5.2. Classes de MSA.....	38
3.5.2.1. Machine de Moore.....	38
3.5.2.2. Machine de Mealy .....	38

3.5.3. Analyse d'un circuit séquentiel.....	39
3.5.4. Synthèse d'un circuit séquentiel.....	39

## Abréviation

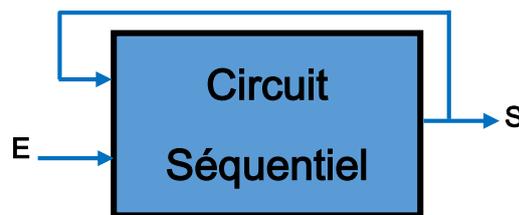
DA	: Demi-Additionneur
ADD	: Additionneur complet
MUX	: Multiplexeur
DEMUX	: Démultiplexeur
CLK / H ou h	: Clock / Horloge
RAM	: Random Access Memory (mémoires vives)
ROM	: Read Only Memory (mémoires mortes)
FSM	: Finit State Machine
MEF	: Machine à Etats Finis
CI	: Circuit Intégré
CMOS	: Complementary Metal Oxide Semiconductor
TTL	: Transistor Transistor Logic

## Chapitre III : conception des Circuits Séquentiels

### 3.1. 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).



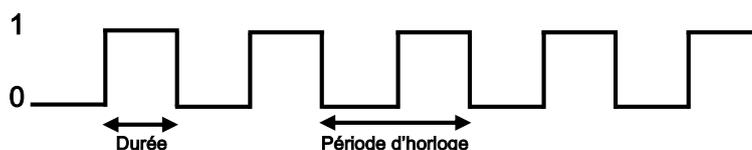
$$S_{t+1} = f(E, S_t) \text{ ou } S^+ = f(E, S)$$

### 3.2. Système séquentiels synchrone et asynchrones

On classe les systèmes séquentiels en deux grandes catégories :

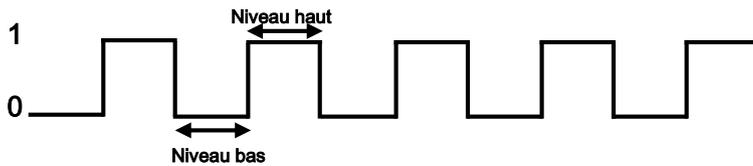
**a-Systèmes séquentiels asynchrones** : dit ainsi s'il peut évoluer seul sans ordre extérieur. Les sorties sont définies par les entrées et par le fonctionnement de circuit.

**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 où il se trouve) on appelle cette entrée de commande horloge. Sa fréquence est l'inverse de sa période (ou temps de cycle) [5]. 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 :**



⇒ Synchronisation sur niveau haut

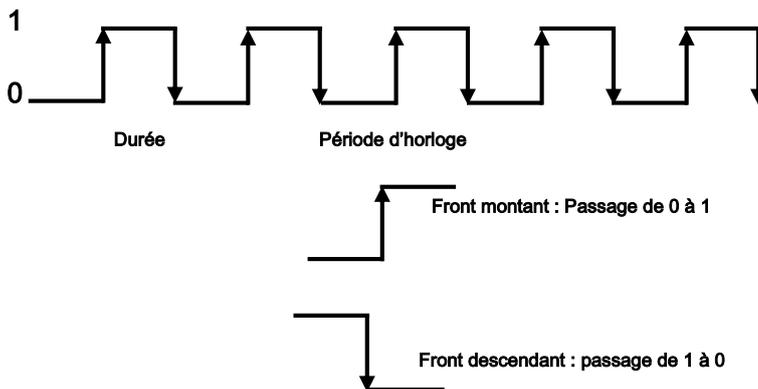
- Si H=0 : la sortie maintient son état, quelles que soient les valeurs des entrées.
- Si H=1 : la bascule fonctionne en mode normale, les sorties obéissent aux entrées.

Donc la bascule ne fonctionne normalement que si H=1(Niveau haut).

⇒ Synchronisation sur niveau bas

- Si H=1 : la sortie maintient son état, quelles que soient les valeurs des entrées.
- Si H=0 : la bascule fonctionne en mode normale.

**b. Signal impulsionnel**



⇒ Synchronisation sur front

Les variables logiques ont deux niveaux : le niveau logique bas « 0 » et le niveau logique haut « 1 ».

- Le passage du niveau bas vers le niveau haut s'appel front montant.
- Le passage du niveau haut vers le niveau bas s'appel front descendant.

Symbole	Fonctionnement	Forme du signal
	Horloge fonctionnant sur niveau haut (H)	
	Horloge fonctionnant sur niveau bas (B)	
	Horloge fonctionnant sur passage du niveau bas au niveau haut	
	Horloge fonctionnant sur passage du niveau haut au niveau bas	

**Table 2.1.** Types d'horloges et leurs symboles.

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

### 3.3. Les bascule

#### 3.3.1. Définition d'une bascule

Une bascule (flip-flop) a pour rôle de mémoriser une information élémentaire. C'est une mémoire à 1 bit. La mémorisation fait appel à un verrou (latch) ou système de blocage. 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  $\bar{Q}$ .

#### 3.3.2. Les types des bascules

Les bistables les plus utilisées sont : Bistable RS, Bistable JK ; bistable D et bistable T.

##### 3.3.2.1. Bascule RS :( Reset\_set)

Est constituée par deux entrées ( S mise à 1 (Set)) et ( R mise à 0 (Rset)) et de deux sorties  $Q$  et  $\bar{Q}$  [6].



Le nouvel état de la bascule  $Q^+$  dépend de son état antérieur  $Q$  et de l'état des entrées R et S.  $Q^+=f(Q,R,S)$

Sorties inchangées :  $\Rightarrow$ Set : remise à 1  $\Rightarrow$ Reset : remise à 0 à proscrire

#### 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

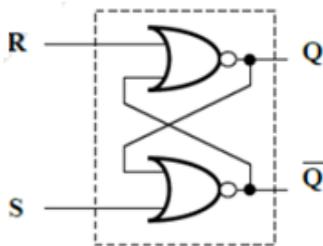
**La table de vérité condensée**

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

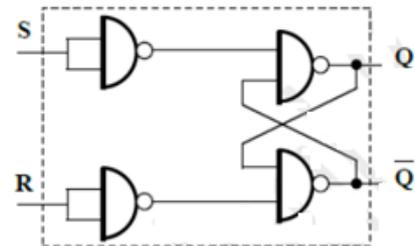
**Les équations d'état**

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

**Représentation à l'aide de portes NOR**



**Représentation à l'aide de portes NAND**



**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

**3.3.2.2. Bascules RST**



**Fonctionnement :**

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

- Si  $T=0$  la bascule conserve son état.
- Si  $T=1$ 

$$\left\{ \begin{array}{l} R = S = 0 \quad \text{conservation d'état} \\ R = 1, S = 0 \quad \text{mise à 0} \\ R = 0, S = 1 \quad \text{mise à 1} \\ R = S = 1 \quad \text{indéterminé} \end{array} \right.$$

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

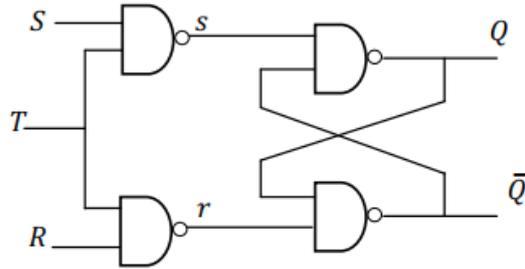
T	R	S	$Q^+$
0	X	X	$Q$ conservation de l'état (mémorisation de l'information)
1	0	0	$Q$ conservation de l'état (mémorisation de l'information)
1	0	1	1 mise à 1 ( $\forall Q$ )
1	1	0	0 mise à 0 ( $\forall Q$ )
1	1	1	X état indéterminé

Les équations d'état

$$Q^+ = \overline{S}R\overline{T} + \overline{T}Q + \overline{S}R\overline{T}Q$$

$$\left\{ \begin{array}{l} s = \overline{TS} = \overline{T} + \overline{S} \\ r = \overline{TR} = \overline{T} + \overline{R} \end{array} \right.$$

### Représentation par les portes NAND



### 3.3.2.3. 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 <sup>+</sup>
0	0	0	0 conservation d'état (mémorisation de l'information)
0	0	1	1 conservation d'état (mémorisation de l'information)
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 inverse d'état (basculément)
1	1	1	0 inverse d'état (basculément)

#### La table de vérité condensée

J	K	Q <sup>+</sup>
0	0	Q conservation de l'état interne
0	1	0 mise à 0 ( $\forall Q$ )
1	0	1 mise à 1 ( $\forall Q$ )
1	1	$\bar{Q}$ inverse d'état (basculément)

**Les équations d'état**

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

**Table de transition d'une bascule JK**

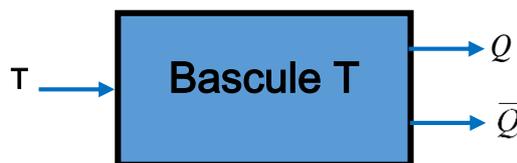
On connaît les valeurs des sorties, comment déterminer les valeurs des entrées JK ? La table des transitions d'une bascule JK se remplit à l'aide de sa table de vérité.

- ligne 1  $\Rightarrow$  Q passe de 0 à 0  $\Rightarrow$   $\begin{cases} \text{état mémoire (J = 0, K = 0)} \\ \text{mise à 0 (J = 0, K = 1)} \end{cases} \Rightarrow (J = 0, K = X)$
- ligne 2  $\Rightarrow$  Q passe de 0 à 1  $\Rightarrow$   $\begin{cases} \text{état inverseur (J = 1, K = 1)} \\ \text{mise à 1 (J = 1, K = 0)} \end{cases} \Rightarrow (J = 1, K = X)$
- ligne 3  $\Rightarrow$  Q passe de 1 à 0  $\Rightarrow$   $\begin{cases} \text{état inverseur (J = 1, K = 1)} \\ \text{mise à 0 (J = 0, K = 1)} \end{cases} \Rightarrow (J = X, K = 1)$
- ligne 4  $\Rightarrow$  Q passe de 1 à 1  $\Rightarrow$   $\begin{cases} \text{état mémoire (J = 0, K = 0)} \\ \text{mise à 1 (J = 1, K = 0)} \end{cases} \Rightarrow (J = X, K = 0)$

Q	Q <sup>+</sup>	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

**3.3.2.4. 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 :**

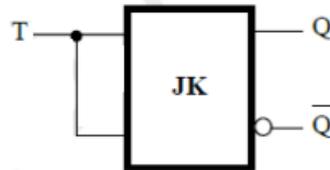
T	Q	Q <sup>+</sup>	
0	0	0	conservation de l'état (état mémoire)
0	1	1	conservation de l'état (état mémoire)
1	0	1	inverse d'état (basculement)
1	1	0	inverse d'état (basculement)

**Les équations d'état**

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

$$= T \oplus Q$$

**Réalisation**



**Remarque :** En remplaçant J et K par T dans l'équation de la bascule JK on aura

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

$$= T \oplus \bar{Q}$$

**3.3.2.5. Bascule D (Delay)**

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

**Fonctionnement :**

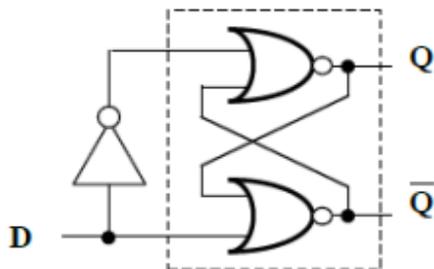
Un appui sur D → Mise à 1 de Q (si D=1, Q<sup>+</sup> = D).

Un relâchement de D → Mise à 0 de Q.

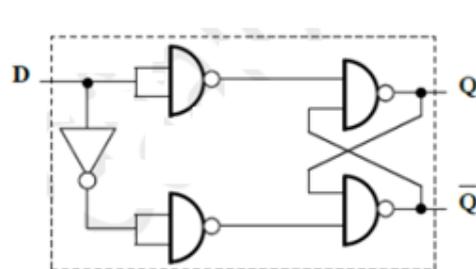
**Table de vérité :**

D	Q	Q <sup>+</sup>
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**

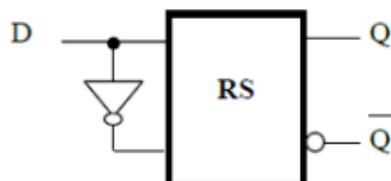


**Remarque :** En mettant S=D et R=  $\bar{D}$  dans l'équation de la bascule RS on aura

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

$$= D$$

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

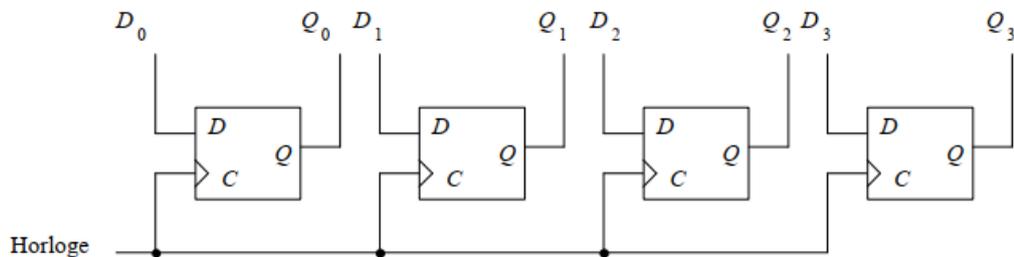
**3.4. Utilisation des bascules**

**3.4.1. Utilisation des bascules pour réaliser un registre**

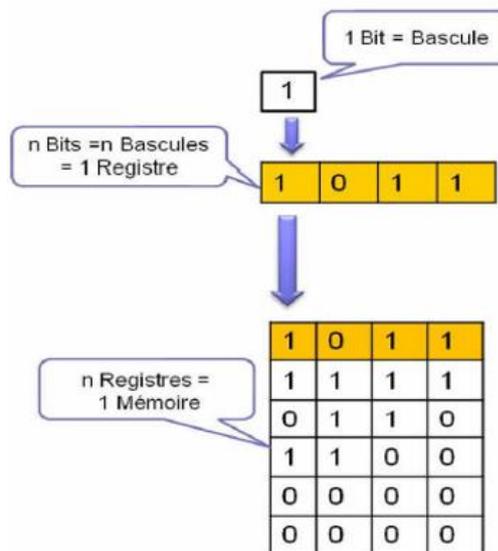
**3.4.1.1. Définition d'un registre**

Un registre est d'abord un ensemble de cases ou cellules mémoires capables de stocker une information (un mot ou un nombre binaire). Un registre est constitué d'une collection de bascules de même type, actionnées par la même impulsion d'horloge. Le nombre de bascules constituant un registre détermine sa capacité [7]. Dans le système binaire, une case mémoire est définie à l'aide d'une bascule. Un registre est donc un ensemble ordonné de bascules.

- Une case mémoire est définie à l'aide d'une bascule
- Une bascule est l'élément de base de la logique séquentielle.
- Une bascule permet de mémoriser un seul bit.
- Un registre est un ensemble ordonné de n bascules.
- Un registre permet de mémoriser (sauvegarder) une information sur n bits.



**Figure 3. 1.** Exemple d'un registre à 4 bits à base de bascules D.



**Figure 3. 2.** Illustration de la relation bit, registre et mémoire.

### 3.4.1.2. Fonctionnement d'un registre

Un registre sert à mémoriser un mot ou un nombre binaire. Le schéma d'un tel système comporte autant de bascules type D que l'élément binaires à mémoriser. Toutes les bascules sont commandées par le même signal d'horloge.

Les principales fonctions d'un registre sont :

➔ La mémorisation

Mémoriser l'information telle qu'elle reste sans aucun changement

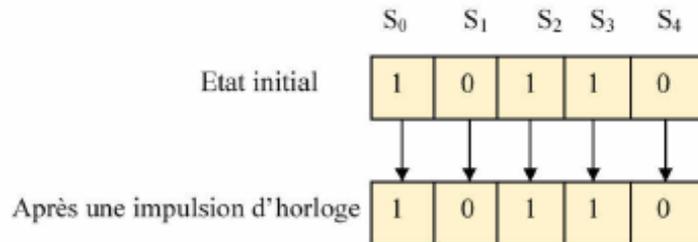


Figure 3. 3. Illustration de la mémorisation.

➔ Le décalage à droite ou à gauche

Décaler l'information soit de la gauche vers la droite ou de la droite vers la gauche

- Décalage à droite : pour chaque impulsion d'horloge le contenu de la bascule de rang  $i$  est transmis à celle de rang  $i+1$ .

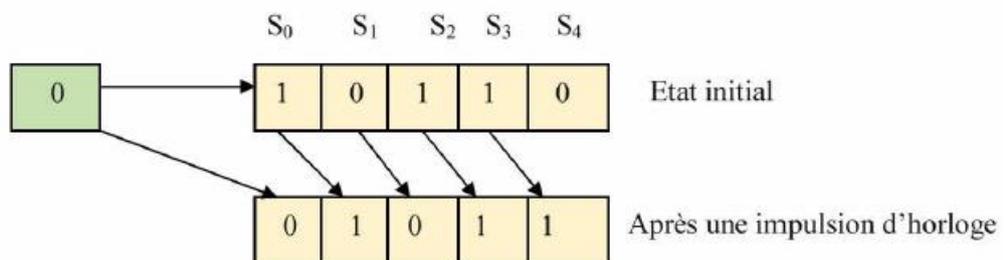


Figure 3. 4. Illustration du décalage à droite.

- Décalage à gauche : pour chaque impulsion d'horloge la bascule de rang  $i$  prend le contenu de la bascule de rang  $i-1$ .

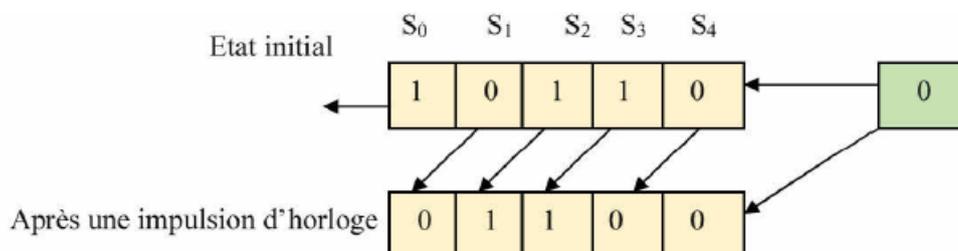
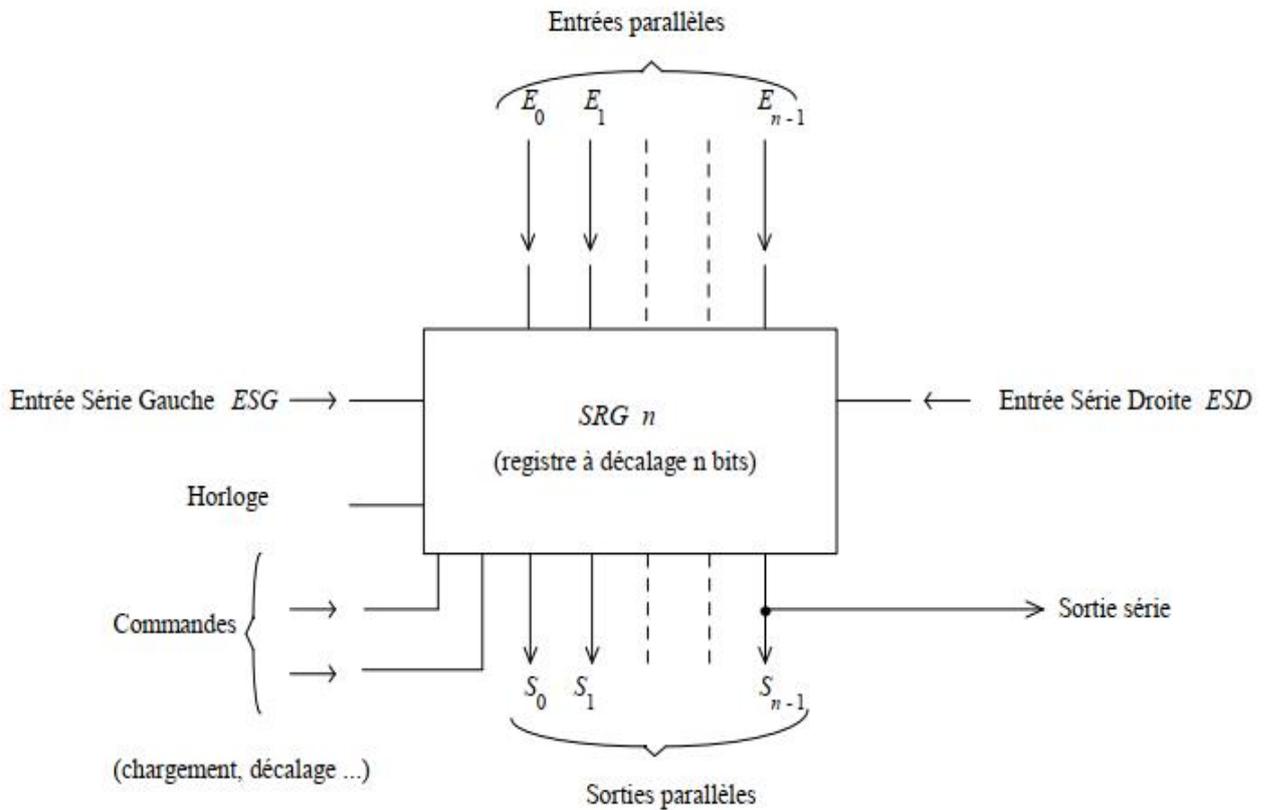


Figure 3. 5. Illustration du décalage à gauche.

### 3.4.1.3. Type des registres

Plus généralement un registre peut se représenter par le schéma suivant [8]:



Il existe plusieurs types de registres :

- Registre de mémorisation
- 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 à droite.
- Registre à décalage à gauche.
- Registre à décalage circulaire.

### 3.4.1.3.1. Registre de mémorisation

Registre de mémorisation peut se réaliser sous la forme représentée par l'exemple au-dessous et qui consiste à interdire l'action de l'horloge en intercalant une porte ET (and) en série.

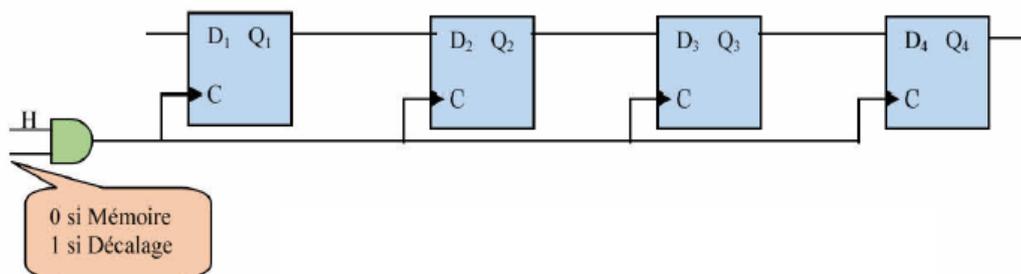


Figure 3. 6. Exemple de registre de mémorisation à base de bascules D.

### 3.4.1.3.2. 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 chargement d'états en même temps.
- Chaque bascule de rang i prend la valeur de l'information i.
- Il possède une entrée de chargement chg ( chg=0 état mémoire, chg=1 chargement )

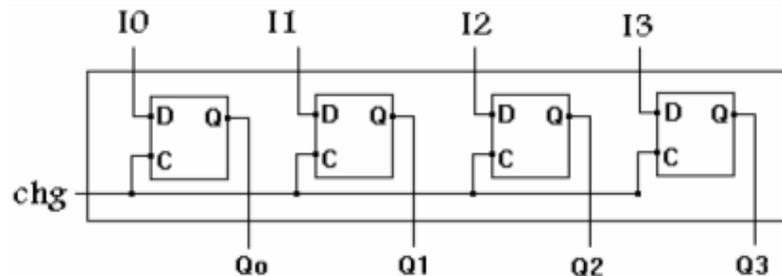


Figure 3. 7. Exemple de registre à chargement parallèle à base de bascules D.

### 3.4.1.3.3. Registre à entrée série et sortie série

- L'information est introduite bit par bit (en série).
- L'ensemble du registre est décalé d'une bascule de rang i vers la bascule de rang i+1 et la première bascule reçoit une nouvelle entrée  $E_s$ .
- Un tel registre est appelé soit :

➔ Registre à entrée série à gauche et à sortie série à droite.

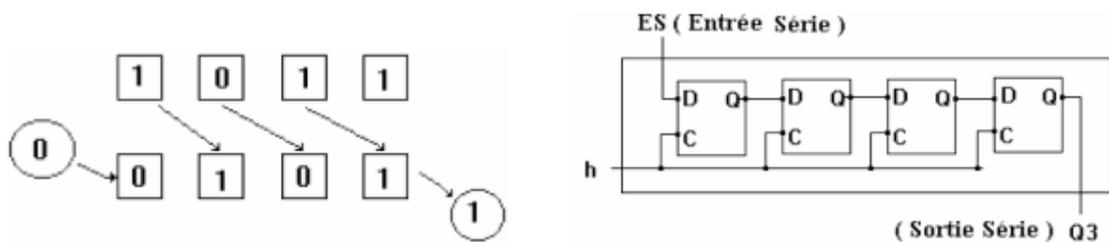


Figure 3. 8. Exemple de registre à entrée série à gauche et à sortie série à droite à base de bascules D.

➔ Registre à entrée série à droite et à sortie série à gauche

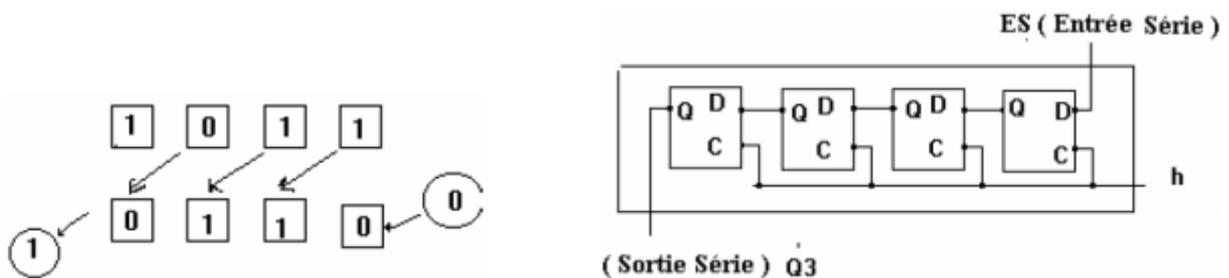


Figure 3. 9. Exemple de registre à entrée série à droite et à sortie série à gauche à base de bascules D.

### 3.4.1.3.4. Registre à entrée série et sortie parallèle

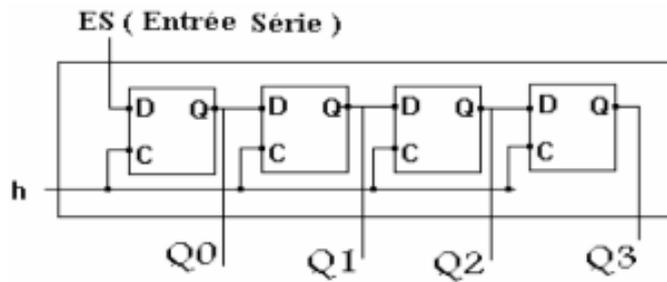


Figure 3. 10. Exemple de registre à entrée série et à sortie parallèle à base de bascules D.

### 3.4.1.3.5. Registre à entrée parallèle et sortie série

Un registre à décalage à entrée parallèle et sortie série transforme un codage spatial en codage temporel. Dans cet exemple Si  $X=1$  l'entrée parallèle est inhibée et l'entrée série est validée. Si  $X = 0$  l'entrée série est bloquée par contre le chargement par l'entrée parallèle est autorisé.

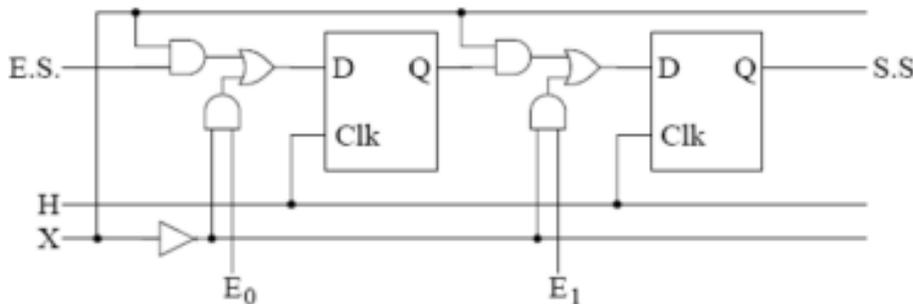


Figure 3. 11. Exemple de registre à entrée parallèle et à sortie série à base de bascules D.

### 3.4.1.3.6. Registre à décalage à droite

C'est un registre qui effectue un décalage vers la droite tel que l'entrée de la première bascule reçoit une nouvelle entrée  $E_s$ .

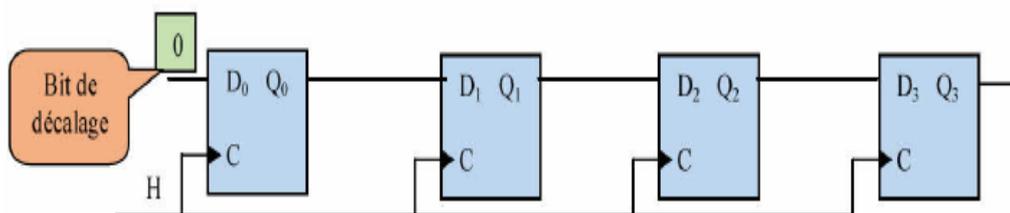


Figure 3. 12. Exemple de registre à décalage à droite à base de bascules D.

### 3.4.1.3.7. Registre à décalage à gauche

C'est un registre qui effectue un décalage vers la gauche tel que l'entrée de la première bascule reçoit une nouvelle entrée  $E_s$ .

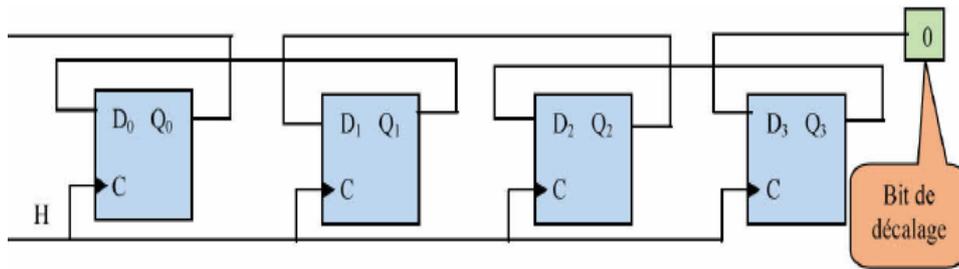


Figure 3. 13. Exemple de registre à décalage à gauche à base de bascules D.

### 3.4.1.3.6. 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)

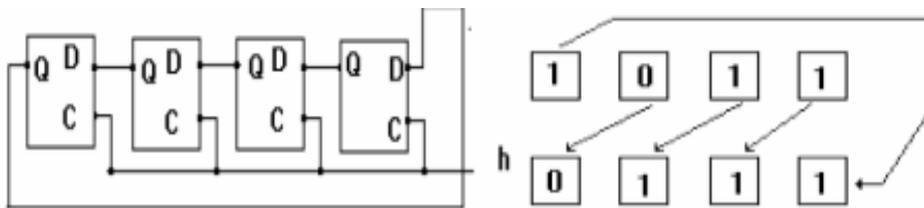


Figure 3. 14. Exemple de registre à décalage circulaire à base de bascules D.

## 3.4.2. Utilisation des bascules pour la mémoire centrale

### 3.4.2.1. Définition d'une mémoire

Une mémoire est un dispositif capable d'enregistrer, de conserver et de restituer des informations. Les éléments de mémoire d'un ordinateur se répartissent en plusieurs niveaux caractérisés par leur capacité et leur temps d'accès.

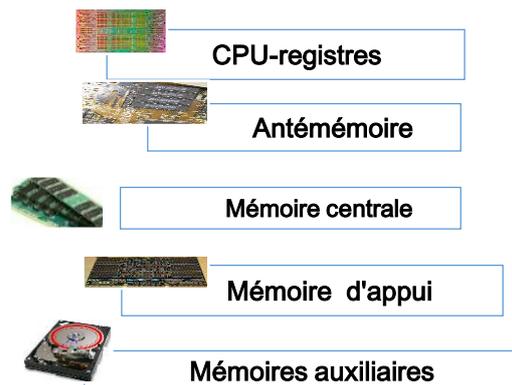


Figure 3. 15. Principaux niveaux de mémoire d'un ordinateur.

### 3.4.2.2. Différents types de la mémoire

Notons que plus on s'éloigne de la CPU vers les mémoires auxiliaires plus le temps d'accès et la capacité des mémoires augmentent, mais que le coût par bit diminue.

1. **les registres** : sont les éléments de mémoire situés dans l'unité centrale de traitement (CPU), élément de stockage des opérandes et des résultats intermédiaires.

2. **L'antémémoire** ou **mémoire cache** Introduit en milieu des années 60 par M. V. Wilkes sous le nom de mémoire esclave (Slave Memory), l'antémémoire ou mémoire cache, ou plus simplement cache, est une mémoire qui sert de tampon entre le processeur et la mémoire [9]. Est une mémoire rapide de faible capacité. Cette mémoire permet au CPU de faire moins d'accès à la mémoire centrale et ainsi de gagner de temps.

3. **La mémoire centrale** est l'organe principal de rangement des informations utilisées par la CPU. La mémoire centrale est un organe qui permet d'enregistrer, de stocker et de restituer les informations [10]. On distingue deux types :

### 3.1. Mémoires vives

- Les mémoires vives ou RAM (signifie en anglais : Random Access Memory) sont des mémoires à lecture et écriture qui permettent d'enregistrer des informations, de les conserver et de les restituer.



Figure 3. 16. RAM.

- La mémoire centrale est composée d'un ensemble ordonné de  $2^m$  cellules (point mémoire), chaque cellule contenant un mot de  $n$  bits, c'est-à-dire que les  $n$  bits sont traités (écrits ou lus) simultanément. La cellule mémoire est la plus petite subdivision (entité atomique) de la mémoire dans laquelle il est possible de lire ou d'écrire une information [11].

- Une mémoire peut être représentée comme une armoire de rangement constituée de différents tiroirs. Chaque tiroir représente alors une case mémoire qui peut contenir un seul élément : des données. Le nombre de cases mémoires pouvant être très élevé, il est alors nécessaire de pouvoir les identifier par un numéro. Ce numéro est appelé adresse. Chaque donnée devient alors accessible grâce à son adresse [12].

Pour pouvoir identifier individuellement chaque mot on utilise  $m$  lignes d'adresse (signal adr). La taille d'un bloc mémoire est donc  $2^m$ , le premier mot se situant à l'adresse 0 et le dernier à l'adresse  $2^m-1$ .

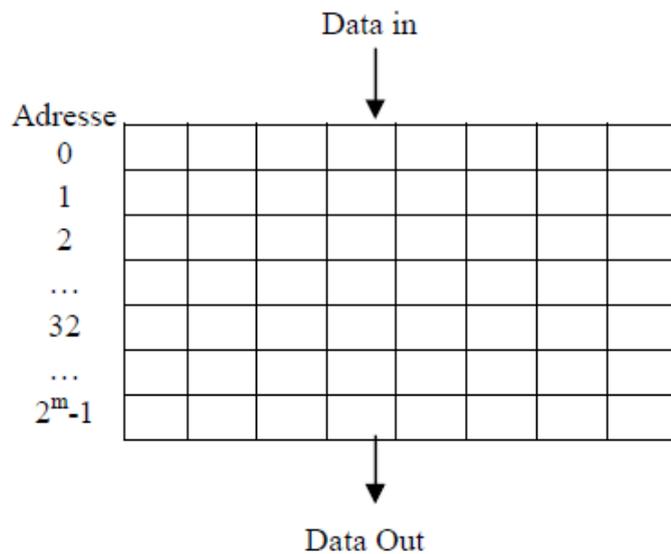


Figure 3. 17. Exemple d'une cellule d'un octet.

Parmi les caractéristiques d'une mémoire nous trouvons la capacité et le format. La capacité représente le nombre total de bits et le format correspond à la longueur des mots. Si  $m$  est le nombre de bits d'adresse et  $n$  est le nombre de bits par mot, la capacité de la mémoire est donnée par :

$$\text{Capacité} = 2^m \text{ mots} = 2^m n \text{ bits}$$

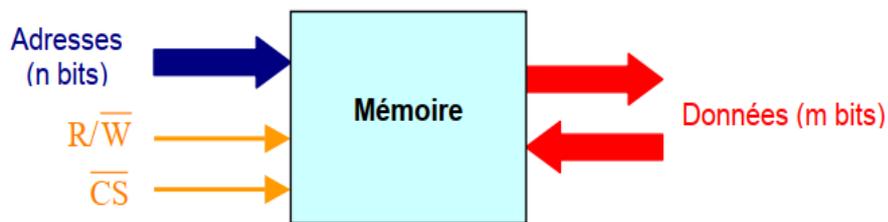


Figure 3. 18. Présentation d'un circuit mémoire.

Sur un circuit mémoire (Figure 3.18) on peut distinguer :

- Les entrées d'adresses
- Les entrées de données
- Les sorties de données
- Les entrées de commandes :
  - Une entrée de sélection de lecture ou d'écriture ( $R/\overline{W}$ ).
  - Une entrée de selection du circuit  $\overline{CS}$ .

Une opération de lecture ou d'écriture de la mémoire suit toujours le mêmes cycle :

1. Sélection de l'adresse
2. Choix de l'opération à effectuer ( $R/\overline{W}$ ).
3. Sélection de la mémoire ( $\overline{CS} = 0$ )
4. Lecture ou écriture de la donnée.

Il existe deux grandes familles de mémoires RAM : les RAM statiques (SRAM) ou les RAM dynamiques (DRAM).

- Dans le cas des RAM statique, le point mémoire élémentaire est une bascule.
- Dans le cas des mémoires dynamiques (DRAM), l'élément de mémorisation est un condensateur (capacité) commandée par un transistor. Ce type de mémoire est très utilisé car peu couteux [13].

### 3.2. Mémoires mortes

La mémoire morte permet de stocker des données nécessaires au démarrage de l'ordinateur:

- ➔ Appelée ROM (Read Only Memory, c'est donc une mémoire en lecture seule),
- ➔ Ne s'efface pas lors de la mise hors tension du système,
- ➔ Contient les éléments essentiels au démarrage de l'ordinateur.

Les mémoires mortes sont utilisées, entre autres, pour stocker :

- ➔ Les informations nécessaires au démarrage d'un ordinateur (BIOS, instructions de démarrage, microcode)
- ➔ Des tables de constantes ou des tables de facteurs de conversion



Figure 3. 19. ROM.

**4. La mémoire d'appui** est une mémoire intermédiaire entre la mémoire centrale et les mémoires auxiliaires.

**5. Les mémoires auxiliaires** (appelées aussi mémoires périphériques ou mémoires de masse ou alors secondaire) sont des mémoires de grande capacité et de coût relativement faible ; qui permettent de stocker les informations pour une plus longue période que ne le fait la mémoire principale de capacité plus limitée. Ce sont par exemple les disques et disquettes, ou les bandes magnétiques.

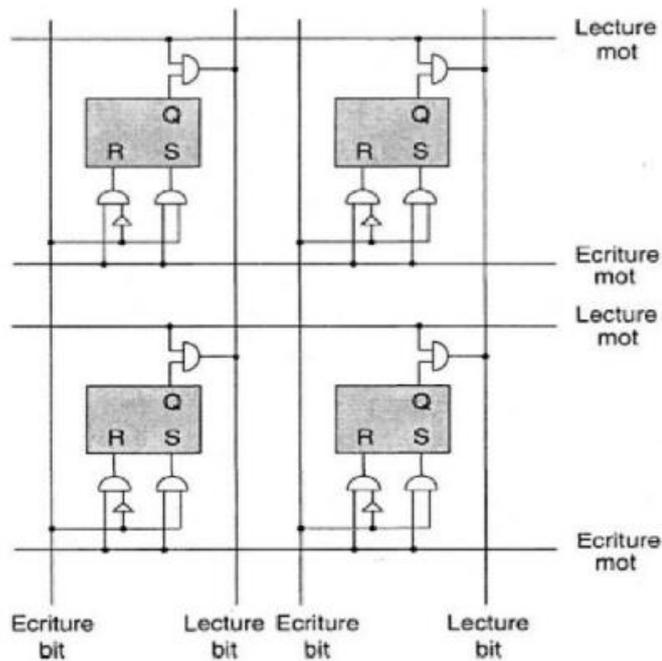


Figure 3. 20. Présentation d'une mémoire.

### 3.4.2.3. Caractéristiques d'une mémoire

- **La capacité** : c'est le nombre total de bits que contient la mémoire. Elle s'exprime aussi souvent en octet.
- **Le format des données** : c'est le nombre de bits que l'on peut mémoriser par case mémoire. On dit aussi que c'est la largeur du mot mémorisable.
- **Le temps d'accès** : c'est le temps qui s'écoule entre l'instant où a été lancée une opération de lecture/écriture en mémoire et l'instant où la première information est disponible sur le bus de données.
- **Le temps de cycle** : il représente l'intervalle minimum qui doit séparer deux demandes successives de lecture ou écriture.
- **Le débit** : c'est le nombre maximum d'informations lues ou écrites par seconde.
- **Volatilité** : elle caractérise la permanence des informations dans la mémoire. L'information stockée est volatile si elle risque d'être altérée par un défaut d'alimentation électrique et non volatile dans le cas contraire.

### 3.4.3. Utilisation des bascules pour réaliser des compteurs

#### 3.4.3.1. 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  $2^n$ .

Caractéristiques générales des compteurs [14]:

- commande d'horloge (synchrone ou asynchrone)
- capacité de comptage
- code de comptage
- vitesse de comptage
- comptage/décomptage
- possibilités de présélection

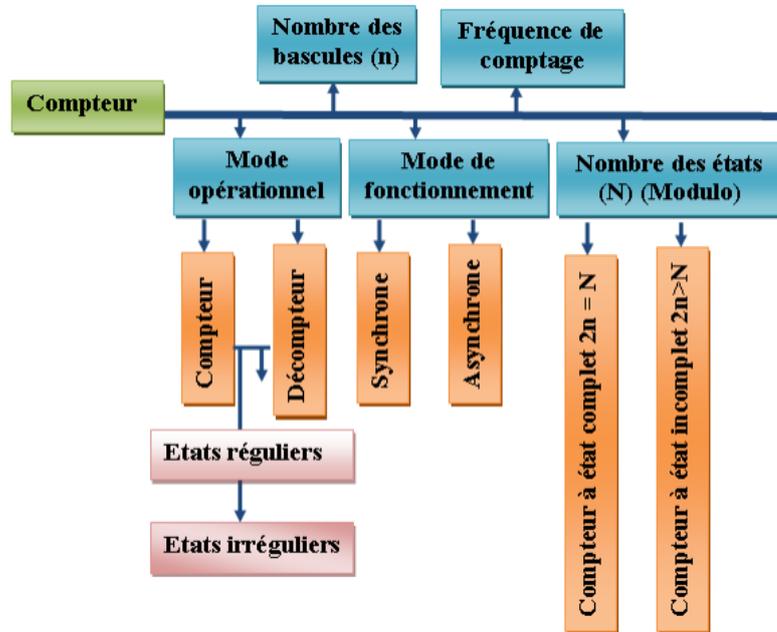


Figure 3. 21. Caractéristiques principales des compteurs [15].

### 3.4.3.2. Types de compteur

Suivant le mode de fonctionnement il existe deux types de compteurs :

- Les compteurs synchrones ou parallèles.
- Les compteurs asynchrones ou séries.

#### 3.4.3.2.1. Compteurs synchrones

Dans un compteur synchrone, toutes les bascules reçoivent, en parallèle, le même signal d'horloge. C'est-à-dire le signal d'horloge synchronise toutes les bascules simultanément.

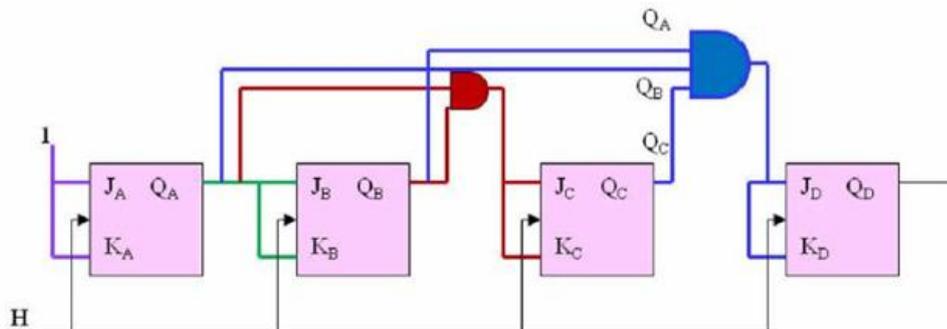


Figure 3. 22. Exemple d'un compteur synchrone à base de bascules JK.

Suivant le cycle réalisé, il existe trois types de compteurs :

**a. Les compteurs synchrones modulo  $2^n$  (cycle complet)**

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

**b. Les compteurs synchrones modulo N (cycle incomplet)**

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

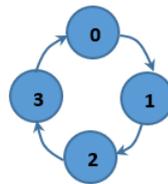
**c. Les compteurs synchrones à cycle quelconque**

- Exemple :  $0 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 8 \rightarrow 10 \rightarrow 0$

**3.4.3.2.1.1. Étude des compteurs synchrones modulo  $2^n$**

**Exemple1** : réalisation d'un compteur synchrone modulo  $2^2 = 4$  à l'aide des bascules JK (le cycle : 0,1,2,3)

**Diagramme d'état**



**Table de vérité**

Pour faire décrire au compteur une séquence déterminée, il faut, à chaque impulsion d'horloge, définir les entrées synchrones J et K. Pour cela, on utilise la table de transition de la bascule J-K (Tableau ci-dessous).

$Q_1$	$Q_0$	$Q_1^+$	$Q_0^+$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
1	1	0	0	X	1	X	1

**Les équations des entrées des bascules**

$Q_1 \backslash Q_0$	0	1
0	0	X
1	1	X

$J_1 = Q_0$

$Q_1 \backslash Q_0$	0	1
0	X	0
1	X	1

$K_1 = Q_0$

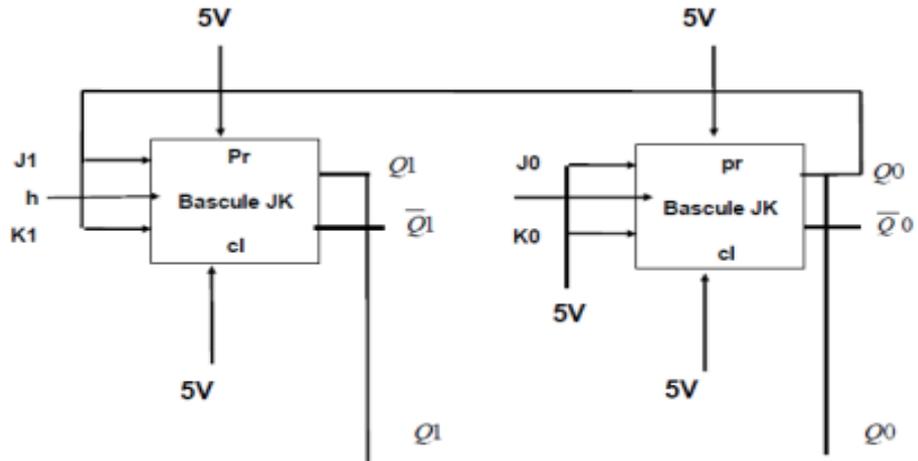
$Q_1 \backslash Q_0$	0	1
0	1	1
1	X	X

$J_0 = 1$

$Q_1 \backslash Q_0$	0	1
0	1	1
1	X	X

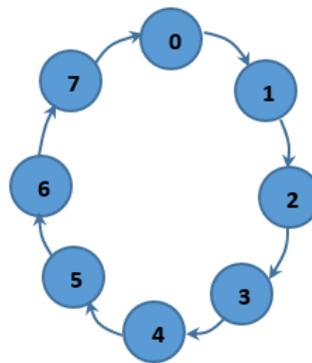
$K_0 = 1$

**Schéma d'un compteur synchrone modulo 4**



**Exemple2** : réalisation d'un compteur synchrone modulo  $2^3 = 8$  à l'aide des bascules JK  
Toutes les bascules possèdent la même horloge. Le cycle est : 0,1,2,3,4,5,6,7.

**Diagramme d'état**



**Table de vérité**

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
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

**Les équations des entrées des bascules**

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	0	0	X	X
$Q_0 \backslash 1$	0	1	X	X

$$J_2 = Q_1 Q_0$$

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	X	X	0	0
$Q_0 \backslash 1$	X	X	0	1

$$K_2 = \overline{Q_1} Q_0$$

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	0	X	X	0
$Q_0 \backslash 1$	1	X	X	1

$$J_1 = Q_0$$

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	X	0	0	X
$Q_0 \backslash 1$	X	1	1	X

$$K_1 = Q_0$$

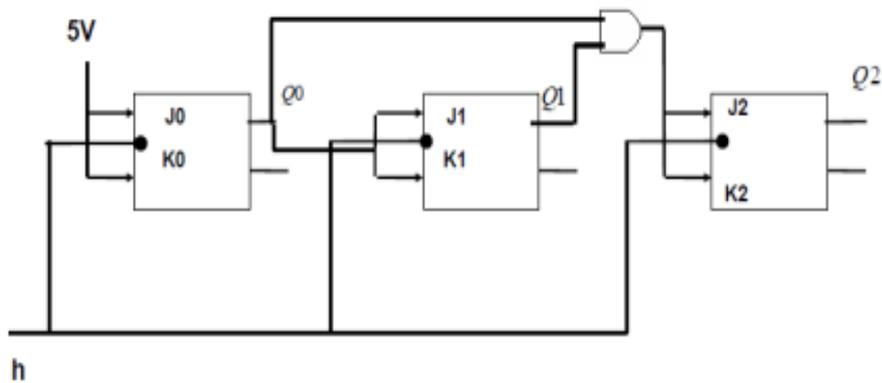
$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	1	1	1	1
$Q_0 \backslash 1$	X	X	X	X

$$J_0 = 1$$

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	X	X	X	X
$Q_0 \backslash 1$	1	1	1	1

$$K_0 = 1$$

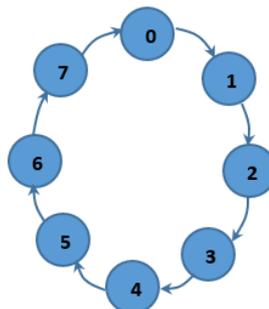
**Schéma d'un compteur modulo 8 synchrone**



**Exemple 3 :** réalisation d'un compteur synchrone modulo  $2^3 = 8$  à l'aide des bascules T (le cycle : 0,1,2,3,4,5,6,7)

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

**Diagramme d'état**



**Table de vérité**

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$T_2$	$T_1$	$T_0$
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

**Les équations des entrées des bascules**

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	0	0	0	0
1	0	1	1	0

$$T_2 = Q_1 Q_0$$

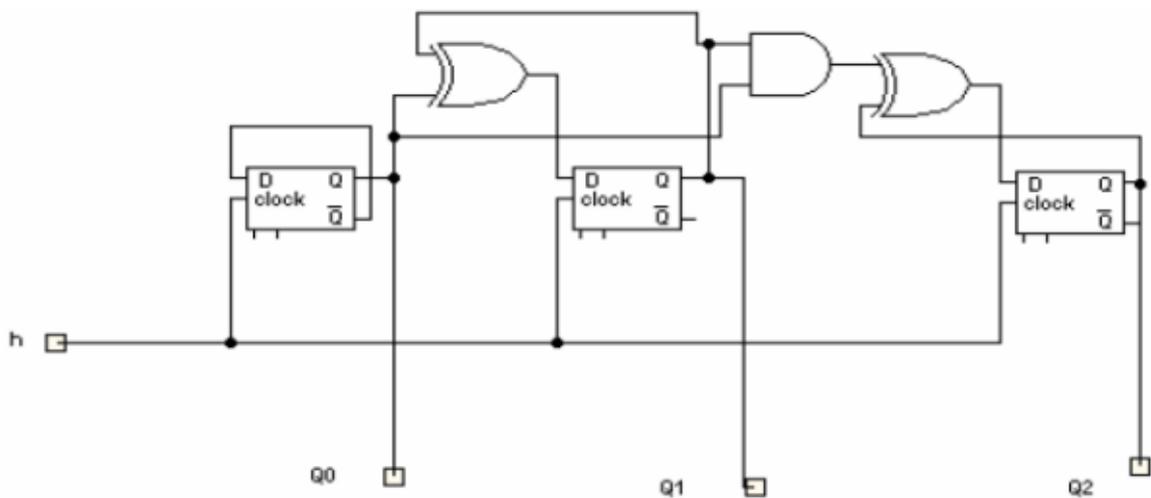
$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	0	0	0	0
1	1	1	1	1

$$T_1 = Q_0$$

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	1	1	1	1
1	1	1	1	1

$$T_0 = 1$$

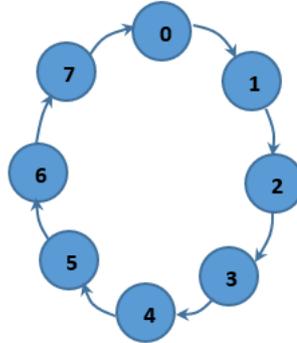
**Schéma d'un compteur modulo 8 synchrone avec des bascules T**



**Exemple 4** : réalisation d'un compteur synchrone modulo  $2^3 = 8$  à l'aide des bascules D (le cycle : 0,1,2,3,4,5,6,7).

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

**Diagramme d'état**



**Table de vérité**

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$D_2$	$D_1$	$D_0$
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

**Les équations des entrées des bascules**

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0$				
0	0	0	1	1
1	0	1	0	1

$$\begin{aligned}
 D_2 &= \bar{Q}_2 \bar{Q}_1 Q_0 + Q_2 \bar{Q}_1 + Q_2 \bar{Q}_0 \\
 &= \bar{Q}_2 \bar{Q}_1 Q_0 + Q_2 (\bar{Q}_1 + \bar{Q}_0) \\
 &= \bar{Q}_2 \bar{Q}_1 Q_0 + Q_2 (\overline{Q_1 Q_0}) \\
 &= Q_2 \oplus (Q_1 Q_0)
 \end{aligned}$$

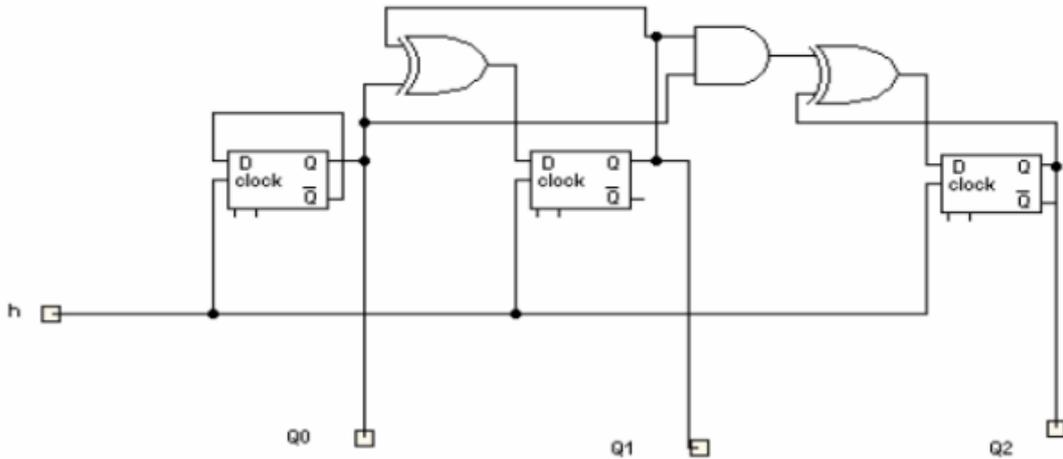
$Q_2 \backslash Q_1$	00	01	11	10
$Q_0$				
0	0	1	1	0
1	1	0	0	1

$$\begin{aligned}
 D_1 &= Q_1 \bar{Q}_0 + \bar{Q}_1 Q_0 \\
 &= Q_1 \oplus Q_0
 \end{aligned}$$

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0$				
0	1	1	1	1
1	0	0	0	0

$$D_0 = \bar{Q}_0$$

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



3.4.3.2.1.2. Étude des Compteurs synchrones modulo N

Exemple 1 : réalisation d'un compteur Modulo 6 à l'aide des bascules D (le cycle :0,1,2,3,4,5)

Diagramme d'état

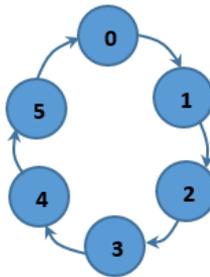


Table de vérité

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$D_2$	$D_1$	$D_0$
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	0	0	0	0	0	0

Les équations des entrées des bascules

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0$				
0	0	0	X	1
1	0	1	X	0

$$D_2 = Q_2 \bar{Q}_0 + Q_1 Q_0$$

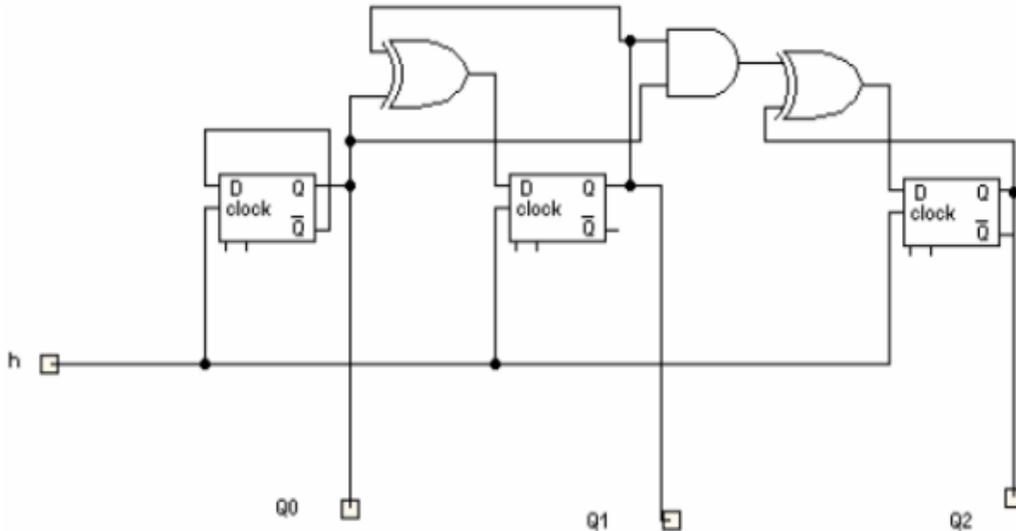
$Q_2 \backslash Q_1$	00	01	11	10
$Q_0$				
0	0	1	X	0
1	1	0	X	0

$$D_1 = \bar{Q}_2 \bar{Q}_1 Q_0 + Q_1 \bar{Q}_0$$

$Q_2 \backslash Q_1$	00	01	11	10
0	1	1	X	1
1	0	0	X	0

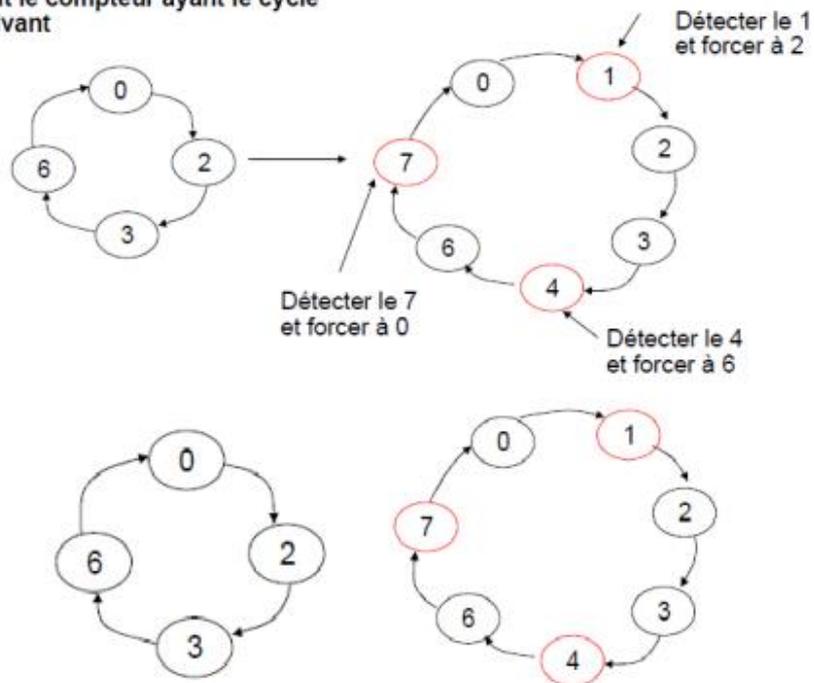
$$D_0 = \bar{Q}_0$$

Schéma d'un compteur modulo 6 synchrone avec des bascules D



3.4.3.2.1.3. Étude des 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 des bascules.
- Pour les états qui n'appartiennent pas au cycle du compteur il faut les considérer comme étant des états indéterminés

**Exemple :** réalisation d'un compteur synchrone à cycle quelconque à l'aide des bascules JK  
Toutes les bascules possèdent la même horloge. Le cycle est : 0,2,3,6.

Diagramme d'état

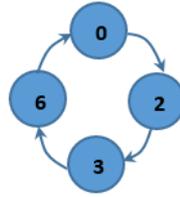


Table de vérité

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	1	0	0	X	0	X	1	X
0	1	0	0	1	1	0	X	1	X	X	1
0	1	1	1	1	0	0	X	X	0	1	X
1	1	0	0	0	0	1	X	X	1	X	1

Les équations des entrées des bascule

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	0	0	X	X
1	0	1	X	X

$$J_2 = Q_1 Q_0$$

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	X	X	0	0
1	X	X	0	1

$$K_2 = \bar{Q}_1 Q_0$$

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	0	X	X	0
1	1	X	X	1

$$J_1 = Q_0$$

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	X	0	0	X
1	X	1	1	X

$$K_1 = Q_0$$

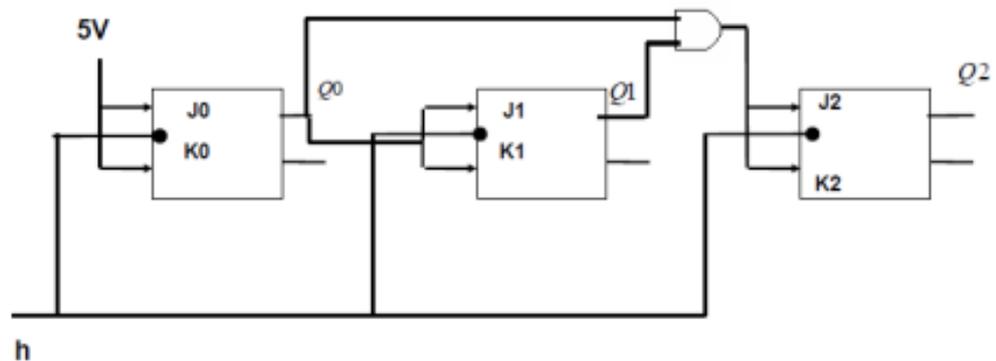
$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	1	1	1	1
1	X	X	X	X

$$J_0 = 1$$

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	X	X	X	X
1	1	1	1	1

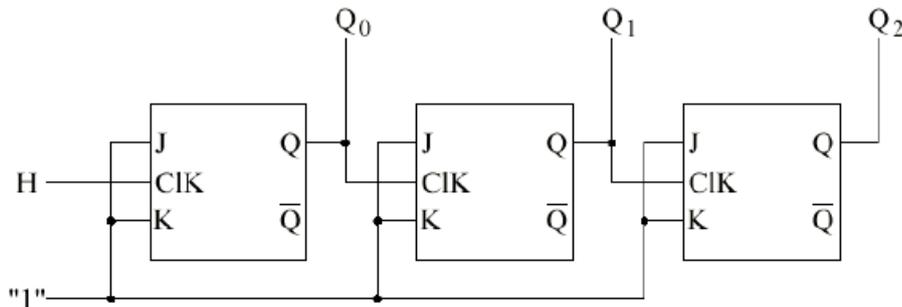
$$K_0 = 1$$

Schéma d'un compteur synchrone modulo 8

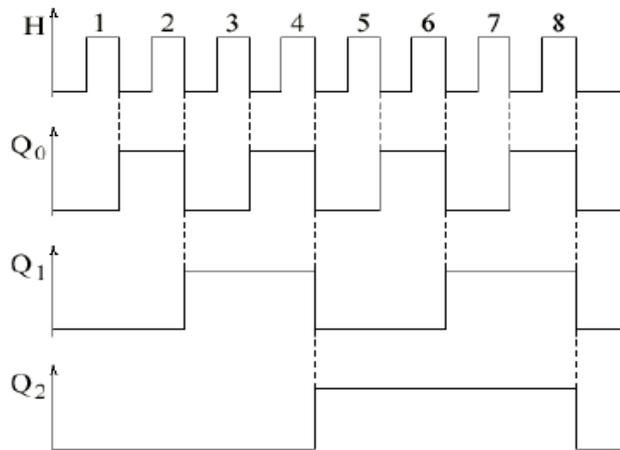


### 3.4.3.2.2. Compteurs asynchrones ou séries

Un compteur asynchrone est un système logique composé de bascules dans lesquels les impulsions que l'on applique à l'entrée doivent traverser la première bascule avant de pouvoir commander la seconde et ainsi de suite jusqu'à la dernière bascule [16]. C'est-à-dire le signal d'horloge n'est reçu que par le premier étage (bascule LSB : *Least Significant Bit*). Pour chacune des autres bascules le signal d'horloge est fourni par une sortie de la bascule de rang immédiatement inférieur. Exemple d'un compteur asynchrone à base de bascules JK.



Dans cet exemple la sortie  $Q_0$  bascule sur chaque front descendant du signal d'horloge. La sortie  $Q_1$  change d'état à chaque transition 1 vers 0 de la sortie  $Q_0$ . De même le basculement de la sortie  $Q_2$  est déclenché par une transition 1 vers 0 de la sortie  $Q_1$ . Comme le montre les chronogrammes suivants.



Suivant le cycle réalisé, il existe trois types de compteurs asynchrones :

#### a. Les compteurs asynchrones modulo $2^n$ (cycle complet)

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

#### b. Les compteurs asynchrones modulo N (cycle incomplet)

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

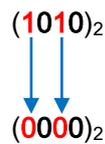
#### c. Les compteurs asynchrones à cycle quelconque

- Exemple :  $0 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 8 \rightarrow 10 \rightarrow 0$

**Exemple 1** : réalisation d'un compteur asynchrone modulo 10 avec des bascules D.

	$Q_3$	$Q_2$	$Q_1$	$Q_0$
	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
Remise à zéro	1	0	1	0

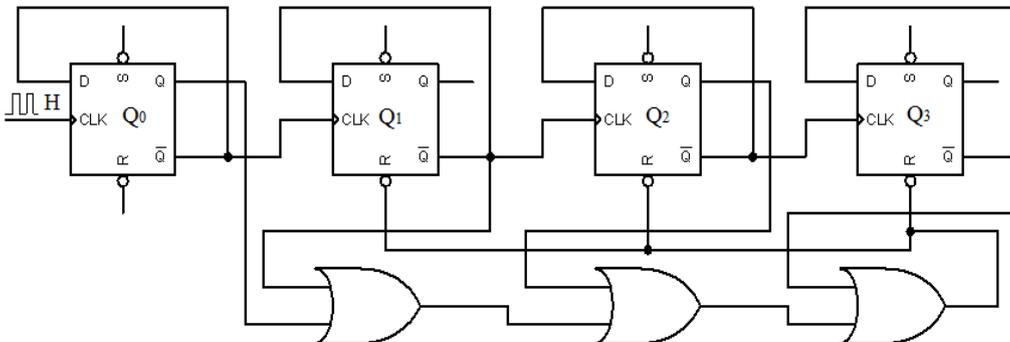
On veut que le compteur passe à  $(0000)_2 = (0)_{10}$  lorsqu'il passe de  $(1001)_2$  à  $(1010)_2$ .



On veut que le compteur passe à 0 lorsqu'il atteint :  $1010 = (1010)_2$ . Pour cela on peut écrire l'expression logique :

$$R = \overline{Q_3} \overline{Q_2} Q_1 \overline{Q_0} = \overline{Q_3} + Q_2 + \overline{Q_1} + Q_0$$

Ce n'est pas suffisant de remettre à zéro  $Q_3$  et  $Q_1$  ( $(1010)_2$  c'est-à-dire  $Q_3 = 1$ ,  $Q_2 = 0$ ,  $Q_1 = 1$  et  $Q_0 = 0$ ) ; il faut mettre  $Q_2$  aussi à zéro. Sinon, lorsque  $Q_1$  passe de 1 à 0,  $Q_2$  passe de 0 à 1 ( $Q_1$  horloge de la bascule  $Q_2$ ).

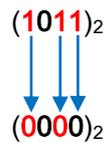


**Exemple 2 :** réalisation d'un compteur asynchrone modulo 11 avec des bascules D.

$Q_3$	$Q_2$	$Q_1$	$Q_0$
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

**Remise à zéro**

On veut que le compteur passe à  $(0000)_2 = (0)_{10}$  (l'entrée R (RESET) soit à 0) lorsqu'il passe de  $(1010)_2$  à  $(1011)_2$ .

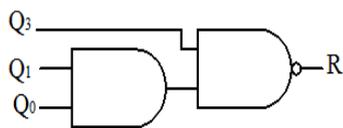


On veut que le compteur passe à 0 lorsqu'il atteint :  $(11)_{10} = (1011)_2$ . Pour cela on peut écrire l'expression logique :

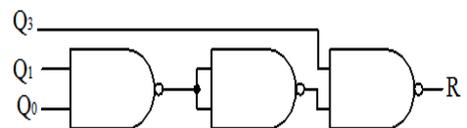
$$R = \overline{Q_3} \overline{Q_2} Q_1 Q_0 = \overline{Q_3} + Q_2 + \overline{Q_1} + \overline{Q_0}$$

Ce n'est pas suffisant de remettre à zéro  $Q_3$ ,  $Q_1$  et  $Q_0$  ( $(1011)_2$  c'est-à-dire  $Q_3 = 1$ ,  $Q_2 = 0$ ,  $Q_1 = 1$  et  $Q_0 = 1$ ) ; On peut simplifier cette relation logique en ne tenant compte que des sorties à 1. En effet c'est la 1<sup>ère</sup> fois que  $Q_3$ ,  $Q_1$  et  $Q_0$  soient à 1. On peut donc utiliser :

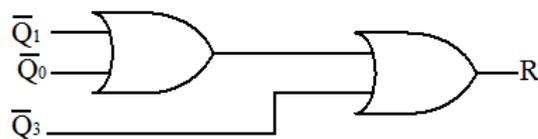
$$R = \overline{Q_3} \overline{Q_1} \overline{Q_0} = \overline{Q_3} + \overline{Q_1} + \overline{Q_0}$$

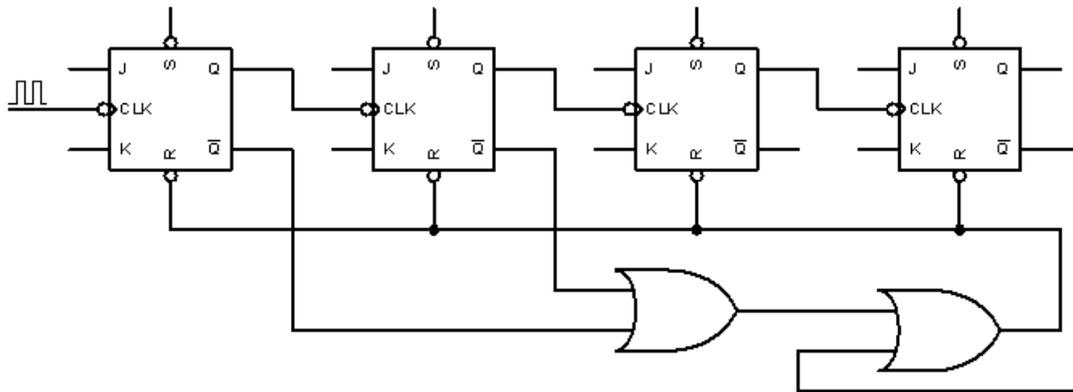


Ou



Ou





Quand des bascules D, ou JK câblées en diviseur par deux ( $J=K=1$ , ces entrées étant reliées au plus de l'alimentation), sont disposées en cascade, la première impulsion d'horloge change l'état de la première bascule, la deuxième impulsion la fait changer à nouveau d'état et ceci provoque le changement de la deuxième bascule, etc [17].

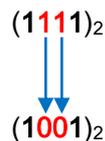
### 3.4.4. Utilisation des bascules pour réaliser Les décompteurs

**Exemple 1 :** réalisation d'un décompteur asynchrone modulo 10 à l'aide des bascules JK

$Q_3$	$Q_2$	$Q_1$	$Q_0$
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1
0	0	1	0
0	0	0	1
0	0	0	0
1	1	1	1

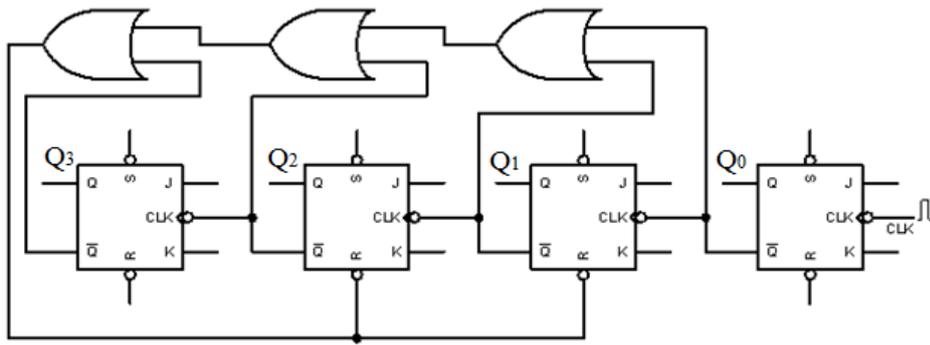
Remise à neuf

On veut que le décompteur passe à  $(1001)_2 = (9)_{10}$  lorsqu'il passe de  $(0000)_2$  à  $(1111)_2$



Pour cela, il suffit de remettre à zéro  $Q_2$  et  $Q_1$ . On veut que le décompteur passe à 9 lorsqu'il atteint :  $(15)_{10} = (1111)_2$ . Pour cela on peut écrire l'expression logique :

$$R = \overline{Q_3 Q_2 Q_1 Q_0} = \overline{Q_3} + \overline{Q_2} + \overline{Q_1} + \overline{Q_0}$$

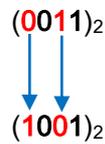


**Exemple 2 :** réalisation d'un décompteur asynchrone de cycle suivant :  $9 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 9$  à l'aide des bascules JK.

$Q_3$	$Q_2$	$Q_1$	$Q_0$
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>

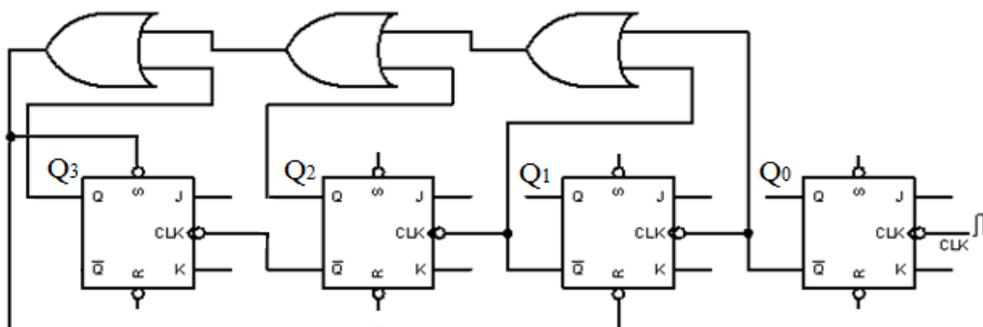
Remise à neuf

On veut que le décompteur passe à  $(1001)_2 = (9)_{10}$  lorsqu'il passe de  $(0100)_2$  à  $(0011)_2$



Pour cela, il suffit de remettre  $Q_1$  à zéro ( $R_1$ ) et  $Q_3$  à 1 ( $S_3$ ).

$$R_1 = S_3 = \overline{Q_3} \overline{Q_2} Q_1 Q_0 = Q_3 + Q_2 + \overline{Q_1} + \overline{Q_0}$$



### 3.4.5. 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 1** : un compteur/décompteur synchrone modulo 8 à l'aide de bascules T

**Table de vérité**

C	$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$T_2$	$T_1$	$T_0$
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	0	1	1	0	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	0	1	1	0	1	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	0	1	0	0	0	0	0	1
1	0	0	0	1	1	1	1	1	1

**Les équations des entrées des bascules**

$C Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	1	0	0
10	0	0	0	0

$$T_2 = \overline{C}Q_1Q_0 + C\overline{Q_1}\overline{Q_0}$$

$C Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$$T_1 = \overline{C}Q_0 + C\overline{Q_0}$$

$C Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$T_0 = 1$$

**Exemple 2** : un compteur/décompteur asynchrone modulo 8 à l'aide de bascules T

**Table de vérité**

C	$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$T_2$	$T_1$	$T_0$
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	0	1	1	0	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	0	1	1	0	1	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	0	1	0	0	0	0	0	1
1	0	0	0	1	1	1	1	1	1

**Les équations des entrées des bascules**

$Q_1 Q_0 \backslash C Q_2$	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	1	0	0
10	0	0	0	0

$$T_2 = \bar{C}Q_1Q_0 + C\bar{Q}_1\bar{Q}_0$$

$Q_1 Q_0 \backslash C Q_2$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$$T_1 = \bar{C}Q_0 + C\bar{Q}_0$$

$Q_1 Q_0 \backslash C Q_2$	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$T_0 = 1$$

### 3.5. Synthèse de circuits séquentiels

La théorie d'automates finis représente un modèle théorique très utile pour la synthèse des circuits séquentiels trop complexes.

#### 3.5.1. Définition d'une machine à états finis

Un automate fini (MEF : Machine à Etats Finis ou en anglais FSM : *Finite State Machine*) ou automate fini est un système séquentiel qui peut se trouver dans un nombre fini d'états et notamment des mémoires. Une machine d'états finis comporte une partie combinatoire et les bascules d'états. La partie combinatoire contient la logique de calcul des sorties et des états suivants [18]. On peut les utiliser pour mémoriser des informations. Autres appellations : (Machine à états, Machine à états finis, Séquenceur, Contrôleur, Automate, Machine séquentielle ou Machine séquentielle algorithmique).

Un automate ne peut prendre qu'un nombre fini d'état (ou noeuds). Il est caractérisé par :

- Sa sortie  $s$ .
- Sa entrée  $e$ .
- Son état  $q$ .

D'où, formellement une FSM est un sextuplet  $M = (Q, U, Y, Init, R, S)$ , où

$Q$  : un ensemble fini d'états de la machine.

$U$  : un ensemble fini de signaux d'entrée, l'alphabet d'entrée.

$Y$  : un ensemble fini de signaux de sortie, l'alphabet de sortie.

$Init \subseteq Q$  : Un ensemble d'états initiaux.

$R : Q \times U \rightarrow Q$ , une fonction appelée fonction de transition.

$S : Q \times U \rightarrow Y$ , une fonction appelée fonction de sortie.

**Remarque** : On peut avoir une FSM où tout état peut être un état initial donc, inutile de spécifier l'ensemble des états initiaux ou sans sortie donc, la fonction  $Y$  n'a aucun sens, ...etc. Les automates sont dits synchrones lorsque le passage d'un état (état présent) à l'état suivant (état futur) a lieu sur une transition d'un signal appelé horloge commun à toutes les bascules de l'automate [19].

On peut représenter une machine par :

- Un chronogramme.
- Une équation logique.
- Une table de transition.
- Un diagramme d'état.
- Une structure logique.

Si les concepts de chronogramme, équation logique et structure logique sont connus les notions de tables de transitions et diagramme d'état restent à définir.

### 3.5.1.1. Tables de transitions

Les diagrammes de transitions ou tables de transitions ou matrices de transitions décrivent sous forme matricielle les fonctions R et S. Elles doivent contenir les états  $q_i$  à l'instant t, les entrées  $e_i$ , les états futurs  $q_i^+$  et les sorties  $s_i$ .

#### Exemple

Soit un automate  $M = (Q, U, Y, R, S)$  définie par :  $Q = U = Y = \{0,1\}$  et dont les fonctions de transition et de sortie sont définies par les tables suivantes :

R	0	1
0	0	1
1	0	1

S	0	1
0	0	0
1	1	1

-Catégories de MSA

#### ⇒ Synchrones

Un automate est dit synchrone lorsque le passage d'un état (état présent) à l'état suivant (état future) a lieu sur une transition d'un signal appelé horloge commun à toutes les bascules de l'automate.

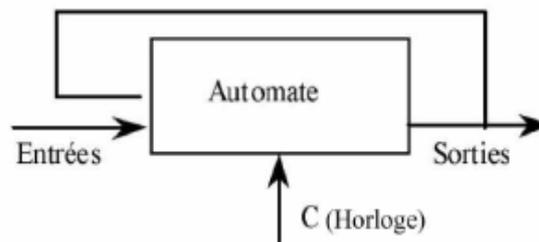


Figure 3. 23. Automate synchrone

#### ⇒ Principalement synchrone

#### ⇒ Asynchrone

### 3.5.1.2. Diagramme d'état

Une autre façon de représenter une machine est l'utilisation d'un graphe de transition, soit une description équivalente des tables de transitions,

- les états (contenus possibles du registre d'état) sont représentés par des cercles.
- les transitions (possibilités de passage d'un état à l'autre) par des arcs orientés, allant de l'état initial à l'état final.
- les états initiaux par deux cercles.



Figure 3. 24. Exemples de diagramme d'états.

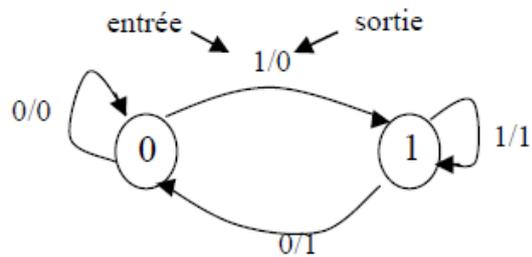
### Remarques

- Une transition peut être inconditionnelle, c'est-à-dire que si le système est dans l'état source considéré, la transition se produit lors du front actif d'horloge suivant.
- Le plus souvent, une transition est conditionnelle, c'est-à-dire que quand le système est dans l'état source, la transition se produit lors du front actif d'horloge suivant si une condition sur les entrées est vérifiée.
- Il peut y avoir maintien d'un état pour certaines valeurs d'entrée. Dans ce cas, l'arc orienté qui représente la transition se referme sur le même cercle.

### Exemples

1. Diagramme d'état de l'exemple précédent.

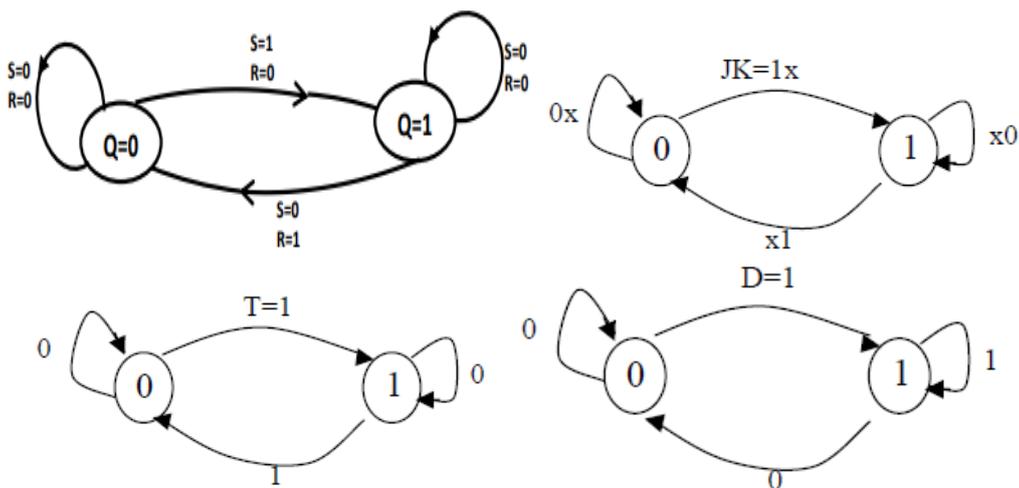
Il suffit de traduire les données des tables de transitions en diagrammes d'états.



2. Diagrammes d'état des bascules RS, JK, T et D

Rappelons les tables caractéristiques réduites des bascules JK, T et D

$Q$	$Q^+$	$J$	$K$	$T$	$D$
0	0	0	X	0	0
0	1	1	X	1	1
1	0	X	1	1	0
1	1	X	0	0	1



### Structure des machines à états finis

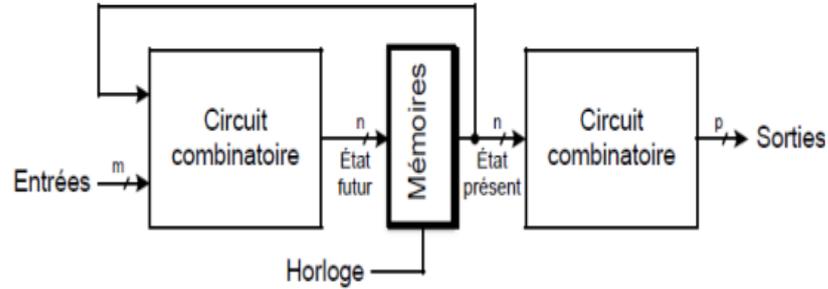


Figure 3. 25. Exemple d'une machine à états finis.

### 3.5.2. Classes de MSA

Souvent, on distingue entre deux modèles de machines d'états finis à savoir le modèle de Moore et modèle de Mealy [18]. Ils diffèrent seulement par la manière dont la sortie est générée.

#### 3.5.2.1. Machine de Moore

Dans une machine de Moore l'état des sorties est fonction seulement de l'état actuel de la machine, résultant de l'état actuel des entrées et de l'état antérieur de la machine. C'est le vecteur d'état, qui permet de prendre en compte l'état antérieur de la machine.

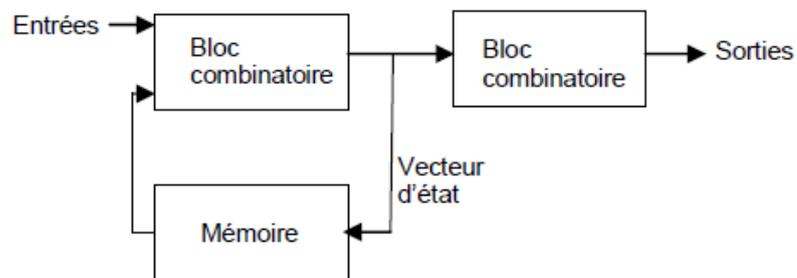


Figure 3. 26. Structure d'une machine de Moore.

#### 3.5.2.2. Machine de Mealy

Dans une machine de Mealy, l'état des sorties est déterminé non seulement à partir de l'état actuel de la machine, matérialisé par le vecteur d'état, mais aussi de l'état actuel des entrées.

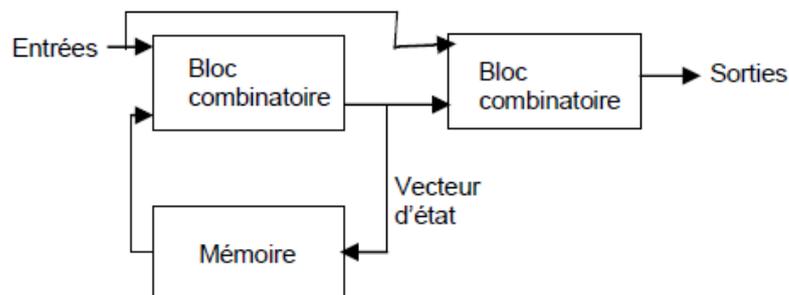


Figure 3. 27. Structure d'une machine de Mealy.

### 3.5.3. Analyse d'un circuit séquentiel

Analyser un circuit séquentiel c'est déterminer son rôle. Pour analyser un circuit séquentiel, on peut suivre les étapes suivantes :

1. Déterminer les fonctions des variables d'entrée.
2. Dresser la table caractéristique du circuit. De cette table, déduire les variables de sortie en se basant sur les expressions logiques des variables d'entrée. Elle a la forme suivante :

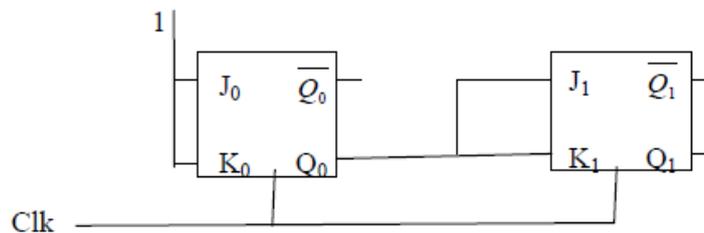
Variables d'entrée	$Q$	$Q^+$
connue	connue	A déterminer

Pour déterminer les états de sortie  $Q^+$ , il faut utiliser la table caractéristique dont il est question dans le circuit à analyser.

3. Déduire le rôle du circuit analysé.

#### Exemple

Analysons le circuit suivant :



Détermination des fonctions d'entrée de chaque bascule.

$$\text{bascule 0} \begin{cases} J_0 = 1 \\ K_0 = Q_1 \end{cases} \quad \text{bascule 1} \begin{cases} J_1 = Q_0 \\ K_1 = Q_0 \end{cases}$$

Table caractéristique.

$Q_1$	$Q_0$	$J_1$	$K_1$	$J_0$	$K_0$	$Q_1^+$	$Q_0^+$
0	0	0	0	1	1	0	1
0	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0

Conclusion : C'est un compteur binaire comptant de 0 jusqu'à 3.

### 3.5.4. Synthèse d'un circuit séquentiel

La synthèse d'un circuit consiste à élaborer le circuit logique à partir du cahier de charge.

Pour faire la synthèse d'un circuit on peut suivre les étapes suivantes :

1. Etablir la table d'excitation correspondant au circuit à réaliser. Elle est de la forme :

$Q$	$Q^+$	Variables d'entrée
connue	connue	A déterminer

2. Déduire l'expression de chaque variable d'entrée.
3. Réaliser le circuit en utilisant les bascules et les portes logiques requises.