

# FONCTIONS SEQUENTIELLES

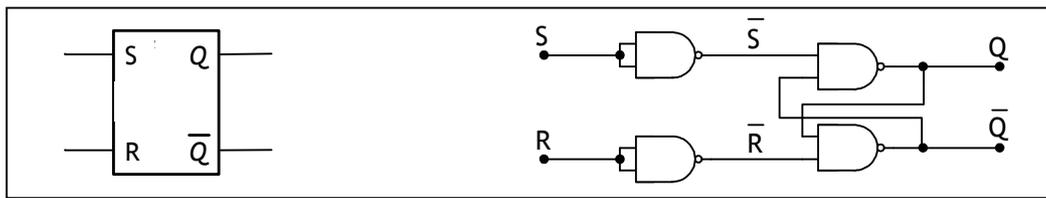
## 1. LES BASCULES :

A l'instar des opérateurs logiques élémentaires en logique combinatoire, les bascules (appelées aussi flip-flop) sont les éléments de base de la logique séquentielle

### 1.1. Bascule SR Asynchrone :

Les bascules RS sont à la base de tous les éléments de mémorisation. Il s'agit d'un montage utilisant deux portes NAND et capable de mémoriser un niveau logique choisi à l'aide de deux sorties complémentaires. Son schéma est donné à la figure 1 :

Fig. 1 : Bascule SR asynchrone

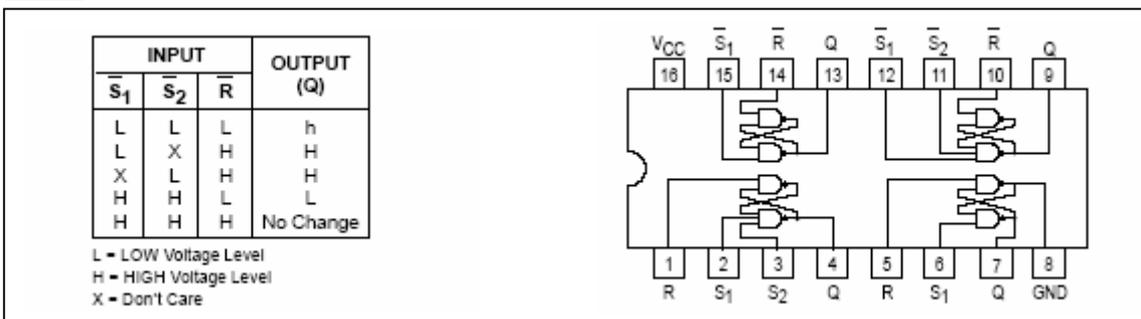


La table de vérité d'une telle bascule est la suivante :

Entrées		Etat précédent	Sorties		Commentaires
S	R	$Q_{n-1}$	$Q_n$	$\bar{Q}_n$	
0	0	0	0	1	Mémorisation de l'état précédent
0	0	1	1	0	
0	1	X	0	1	Remise à zéro de la sortie $Q_n$
1	0	X	1	0	Remise à un de la sortie $Q_n$
1	1	X	1	1	Etat indéfini

Parmi les circuits intégrés à bascules  $\bar{S}\bar{R}$ , on trouve le 74279 dont la table de fonction et le schéma de brochage sont donnés à la figure 2 :

Fig. 2 : Brochage et table de vérité du 74279



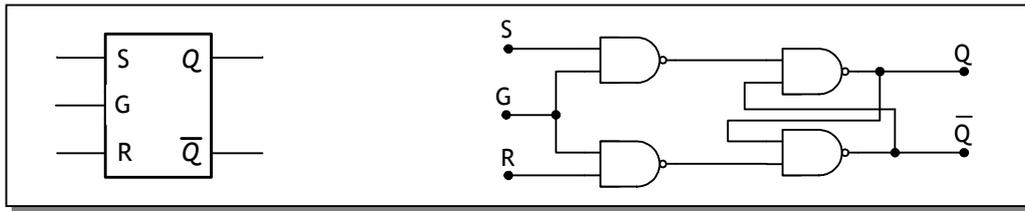
Cette bascule présente deux inconvénients majeurs :

- Sensibilité de la sortie Q aux changements indésirables (parasites) des entrées S et R ;
- La configuration  $S = R = 1$  est à éviter parce qu'elle conduit à l'égalité entre les deux sorties  $Q$  et  $\bar{Q}$  et donc il n'y a plus complémentarité comme c'est indiqué dans la définition de la bascule ;

## 1.2. Bascule SR Synchronne :

La bascule SR synchronne (figure 3) permet de résoudre le premier inconvénient de la bascule SR asynchrone. Les ordres Set et Reset ne changent l'état de la sortie qu'après l'autorisation d'un signal de commande G (Gate) :

Fig. 3 : Bascule SR synchronne

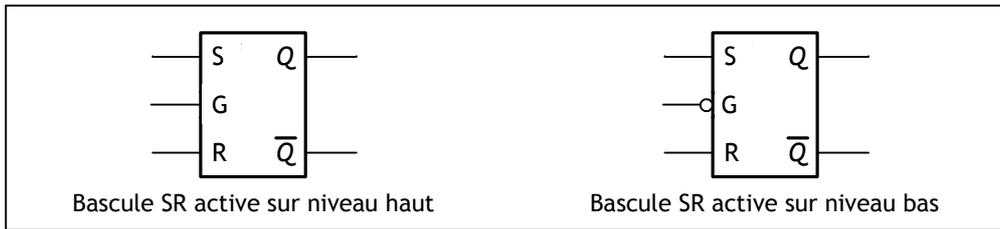


La table de vérité d'une telle bascule est la suivante :

G	S	R	$Q_n$	Commentaires
0	X	X	$Q_{n-1}$	Mémorisation de l'état de la sortie
1	0	0	$Q_{n-1}$	Fonctionnement normal de la bascule SR
1	0	1	0	
1	1	0	1	

Le changement d'état de cette bascule est autorisé sur niveau logique 1 du signal de commande G. On dit que c'est une bascule commandée de manière statique (sur niveau logique 0 ou 1) et sa représentation est donnée à la figure 4 :

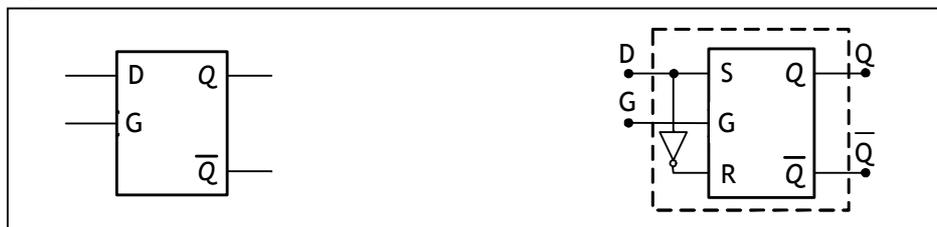
Fig. 4: Bascule SR statique



## 1.3. Bascule D :

La bascule D est dérivée de la bascule SR en ajoutant un inverseur entre Set et Reset pour n'avoir plus qu'une seule entrée pour fixer le niveau à mémoriser. Avec un tel montage, il n'y a plus de combinaisons d'entrées invalides ( $S=R=1$ ) :

Fig. 5: Bascule D



La table de vérité d'une telle bascule est la suivante :

G	D	$Q_n$	Commentaires
0	X	$Q_{n-1}$	Mémorisation de l'état de la sortie
1	0	0	Mise à zéro de la sortie Q
1	1	1	Mise à un de la sortie Q

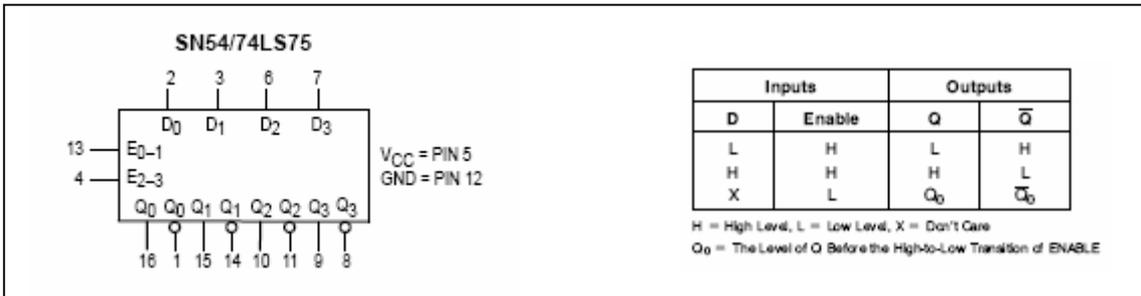
Le changement d'état d'une bascule D peut aussi être autorisé par le front montant ou le front descendant du signal de commande G. On parle alors de bascule D dynamique (figure 6) :

**Fig. 6:** Bascule D dynamique



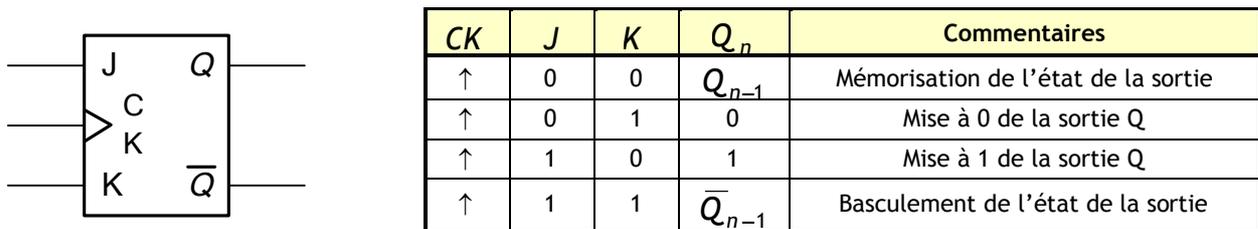
Parmi les circuits intégrés à bascules D, on trouve le 7475 dont la table de fonction et le schéma de brochage sont donnés par la figure 7 :

**Fig. 7:** Brochage et table de vérité du 74LS75



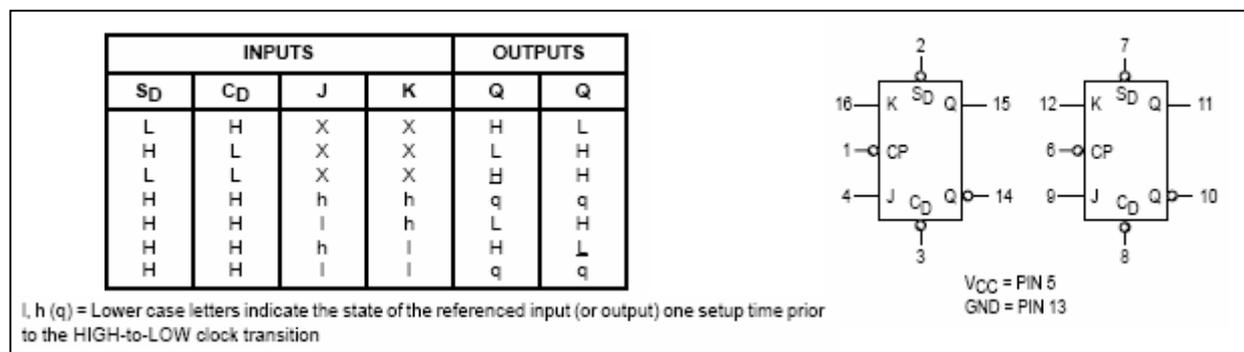
### 1.4. Bascule JK :

Le principe des bascules dynamiques permet de mieux protéger la bascule contre les changements indésirables des entrées. La bascule JK permet en plus de lever l'ambiguïté qui existe pour l'état S=R=1 d'une bascule SR. Son symbole et sa table de vérité sont les suivants :



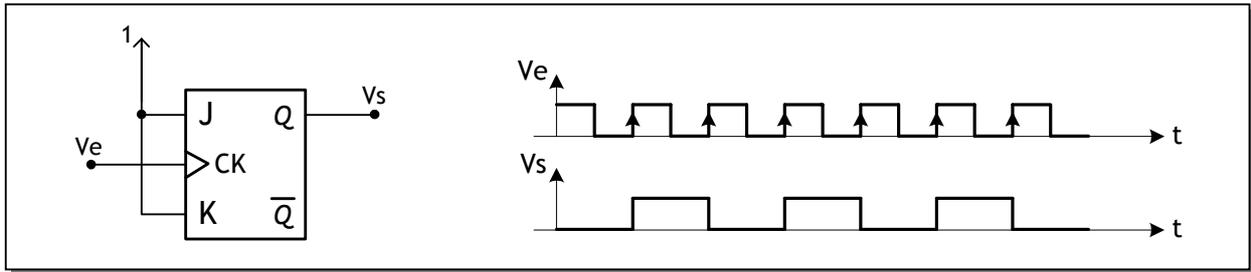
Parmi les circuits intégrés à bascules JK, on trouve le 7476 dont la table de fonction et le schéma de brochage sont donnés à la figure 8 :

**Fig. 8:** Brochage et table de vérité du 74LS76



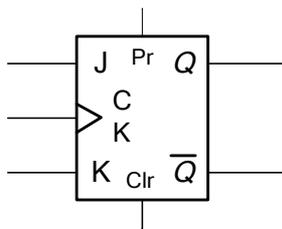
Si on utilise la bascule JK avec  $J = K = 1$ , on obtient l'une des principales applications de la bascule JK à savoir le diviseur de fréquence par 2 (figure 9) :

**Fig. 9:** Diviseur de fréquence par 2



### 1.5. Fonctionnement forcé des bascules :

Il est parfois nécessaire d'affecter le niveau de sortie d'une bascule de manière non synchrone c'est-à-dire indépendamment de l'horloge. C'est le rôle des entrées de forçage asynchrone Preset (Set) et Clear (Clr) qui permettent d'initialiser la bascule :



PR	Clr	$Q_n$	Commentaires
0	0	$Q_n$	Fonctionnement normal de la bascule
0	1	0	Forçage à 0 de la sortie Q
1	0	1	Forçage à 1 de la sortie Q

## 2. LES COMPTEURS :

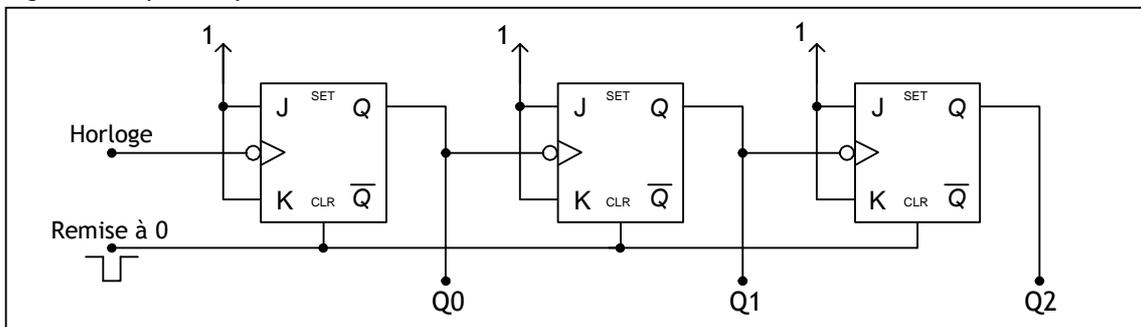
Un compteur est un ensemble de  $n$  bascules interconnectées par des portes logiques. Ils peuvent décrire, au rythme d'un signal de commande appelé horloge, une suite d'états binaires. Il ne peut y avoir au maximum que  $2^n$  combinaisons et le nombre total  $N$  des combinaisons successives est appelé le modulo du compteur. Les compteurs binaires peuvent être classés en deux catégories :

- Les compteurs asynchrones ;
- Les compteurs synchrones ;

### 2.1. Compteur asynchrone modulo $N = 2^n$ :

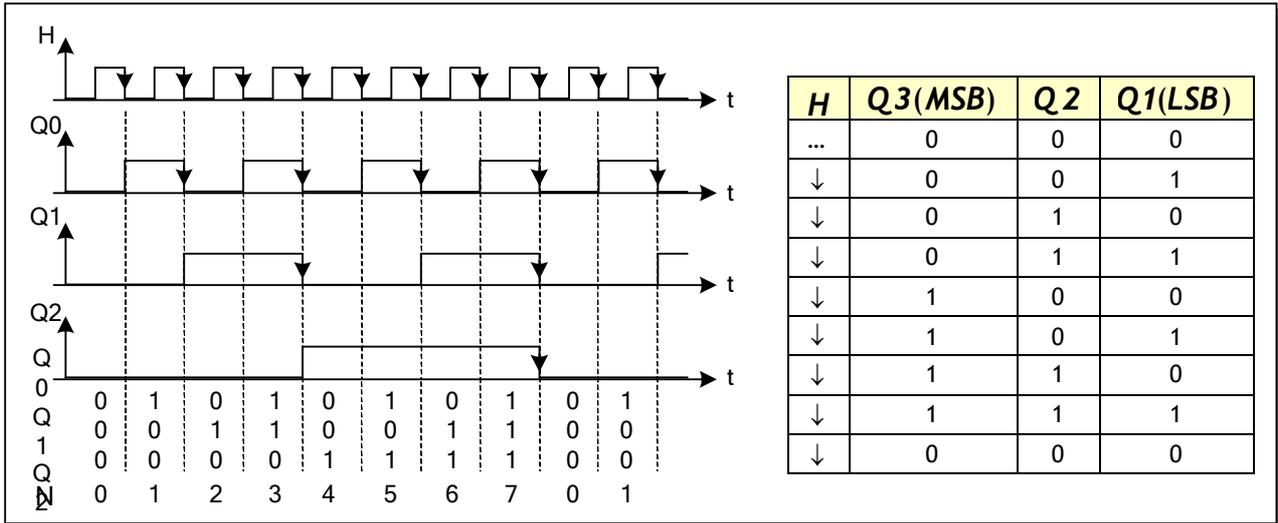
Ce type de compteur est constitué de  $n$  bascules JK fonctionnant en mode T (Toggle) :  $J=K=1$ . Ces bascules sont montées en cascade c'est-à-dire le signal d'horloge commande uniquement la première bascule tandis que pour chacune des autres bascules le signal d'horloge est fourni par la sortie de la bascule de rang immédiatement inférieur. Pour bien comprendre le principe, réalisons un compteur modulo 8 permettant de compter de 0 à 7 comme le montre la figure 10 :

**Fig. 10:** Compteur asynchrone module 8



Les chronogrammes et al table de vérité d'un tel compteur sont donnés à la figure 11 :

**Fig. 11:** Chronogrammes et table de vérité d'un compteur asynchrone module 8

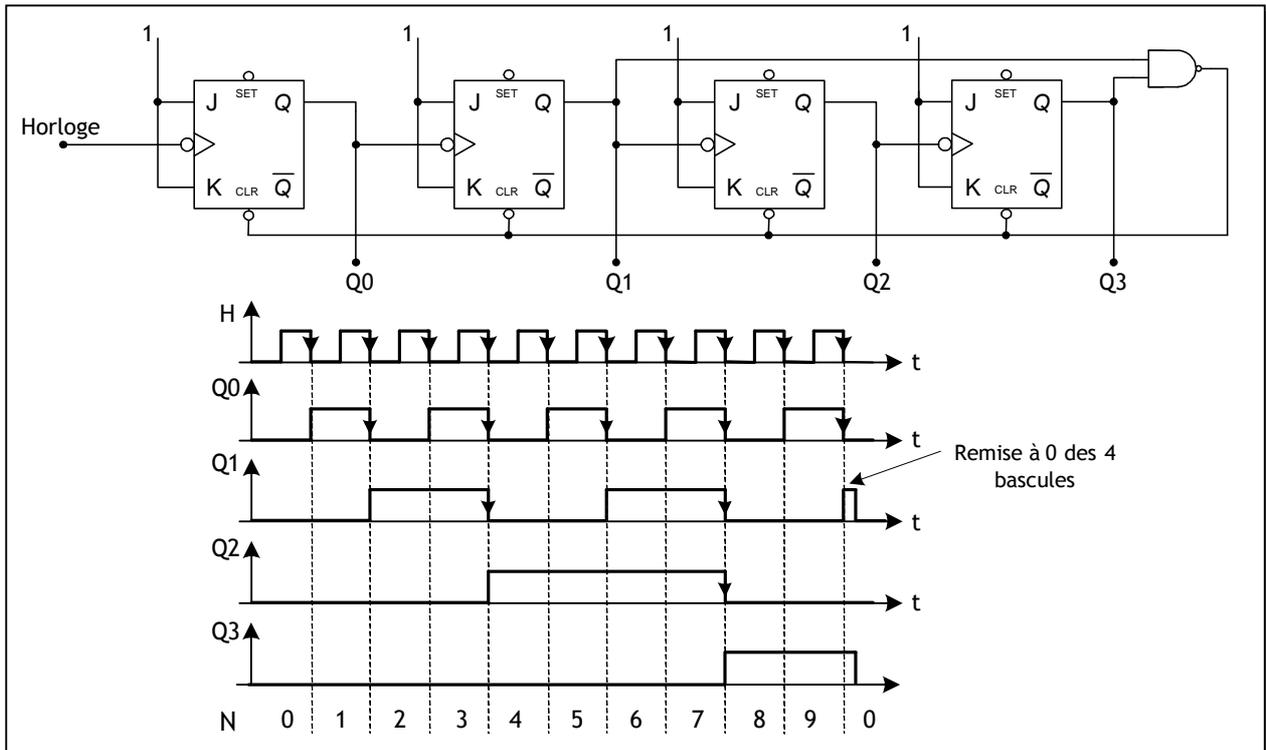


## 2.2. Compteur asynchrone modulo $N \neq 2n$ :

Pour ce type général de compteur qui compte de 0 à N-1, on va étudier l'exemple d'un compteur asynchrone modulo 10 (0 à 9). Pour le réaliser, il y a deux étapes :

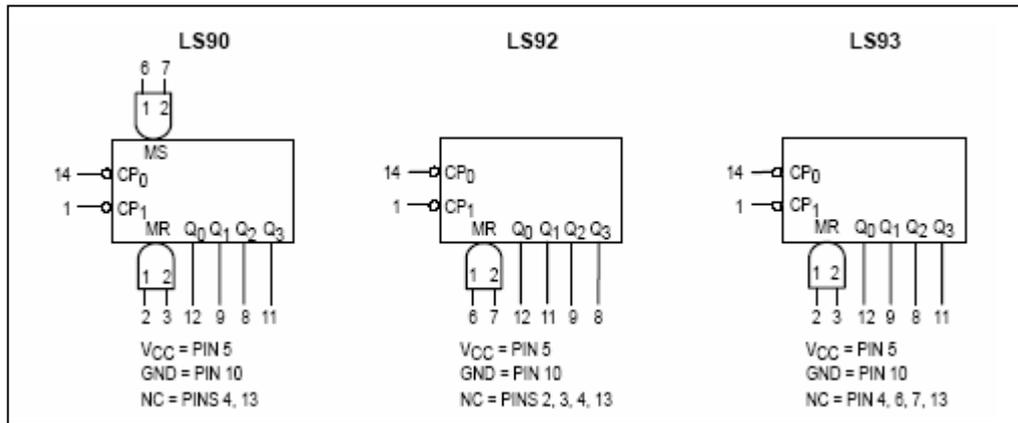
- On cherche d'abord la puissance de 2 immédiatement supérieure à N et qui est pour notre compteur  $2^4 = 16$ . L'exposant de cette puissance de 2 donne le nombre de bascules JK à monter en cascade, 4 pour notre exemple ;
- On détecte ensuite l'état N qui remettra le compteur à 0 et qui est pour notre compteur  $10 = (1010)_2$ . On relie les sorties  $Q = 1$  ( $Q_1$  et  $Q_3$ ) pour N aux entrées d'une porte NAND dont la sortie commandera l'entrée CLR de chaque bascule (figure 12).

**Fig. 12:** Compteur asynchrone modulo 10



Les compteurs 7490 (modulo 10), 7492 (modulo 12) et 7493 (modulo 16) sont des compteurs asynchrones (figure 13) composés de 4 bascules dont les connexions internes varient suivant le type du compteur :

**Fig. 13:** Exemples de compteur asynchrones



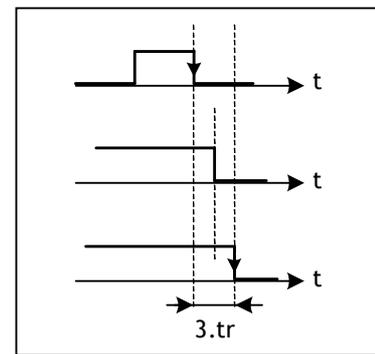
### 2.3. Compteur synchrone modulo N :

L'inconvénient du compteur asynchrone c'est le temps de réponse de chaque bascule. Ainsi, le signal d'horloge ne parvient pas simultanément sur toutes les bascules. Ceci a pour conséquence de provoquer des états transitoires qui peuvent être indésirables.

Supposons un temps de réponse  $t_r$  identique pour toutes les bascules et considérons la chronologie du passage d'un compteur asynchrone 3 bits de 011 à 100 dans la figure ci-contre. Nous constatons que le compteur passe par les états transitoires 011 et 001 qui sont faux en plus d'un temps de propagation qui a triplé.

Dans un compteur synchrone toutes les bascules reçoivent en parallèle le même signal d'horloge. Pour faire décrire au compteur une séquence déterminée il faut à chaque impulsion d'horloge définir les entrées synchrones J et K.

**Fig. 14:** Temps de réponse



Pour cela on utilise la table de transition de la bascule J-K ainsi que la table de vérité décrivant la séquence du compteur. Prenons l'exemple d'un compteur modulo 8 :

Q3	Q2	Q1
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

$Q_n$	$Q_{n+1}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

**Table d'excitation**

La résolution du problème consiste à chercher les équations des entrées J et K de chaque bascule à l'aide de la table des états recherchés :

**$J_0$**

		<b><math>Q_1 Q_0</math></b>			
		00	01	11	10
<b><math>Q_2</math></b>	0	1	x	X	1
	1	1	X	X	1

$J_0 = K_0 = 1$

**$K_0$**

		<b><math>Q_1 Q_0</math></b>			
		00	01	11	10
<b><math>Q_2</math></b>	0	X	1	1	X
	1	X	1	1	X

**$J_1$**

		<b><math>Q_1 Q_0</math></b>			
		00	01	11	10
<b><math>Q_2</math></b>	0	0	1	X	X
	1	0	1	X	X

$J_1 = K_1 = Q_0$

**$K_1$**

		<b><math>Q_1 Q_0</math></b>			
		00	01	11	10
<b><math>Q_2</math></b>	0	X	X	1	0
	1	X	X	1	0

**$J_2$**

		<b><math>Q_1 Q_0</math></b>			
		00	01	11	10
<b><math>Q_2</math></b>	0	0	0	1	0
	1	X	X	X	X

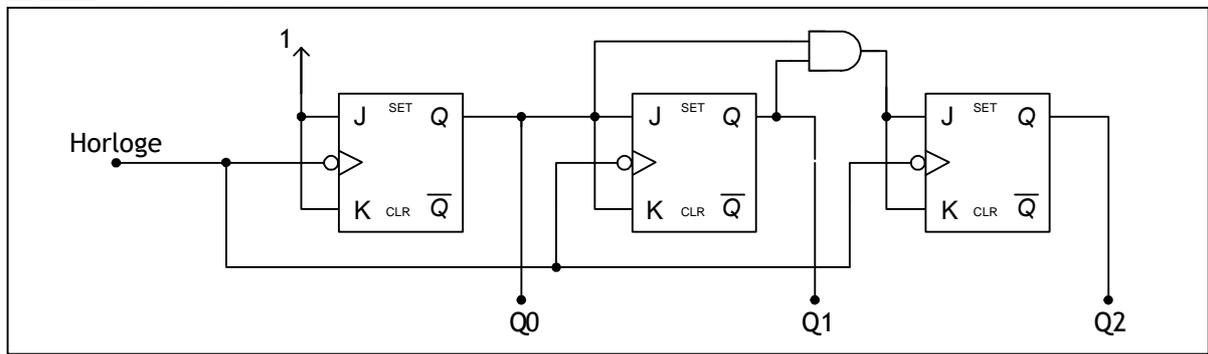
$J_2 = K_2 = Q_0.Q_1$

**$K_2$**

		<b><math>Q_1 Q_0</math></b>			
		00	01	11	10
<b><math>Q_2</math></b>	0	X	X	X	X
	1	0	0	1	0

Le montage du compteur synchrone 3 bits est donné à la figure 15 :

Fig. 15: Compteur synchrone modulo 8

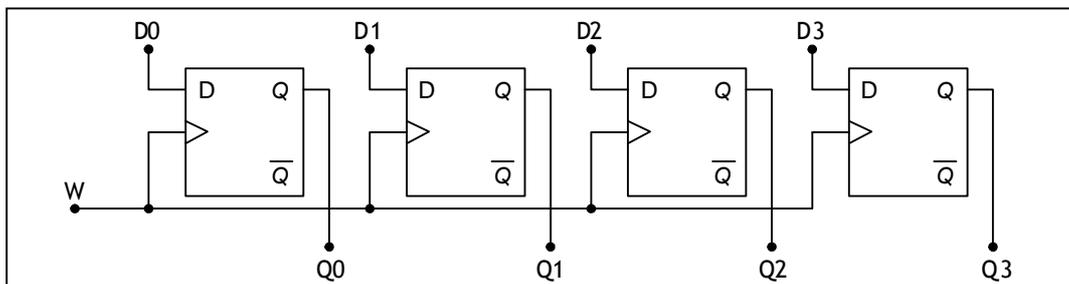


### 3. LES REGISTRES :

#### 3.1. Présentation :

Un registre est constitué d'un assemblage de n bascules D permettant la mémorisation temporaire de n bits avec ou sans décalage. L'information est emmagasinée sur un signal de commande et ensuite conservée et disponible en lecture. La figure 16 donne un exemple de registre de mémorisation 4 bits avec le signal d'écriture W (Write) qui commande la mémorisation des données D0, D1, D2 et D3:

Fig. 16: Registre de mémorisation 4bits à lecture / écriture parallèles



### 3.2. Les registres à décalage :

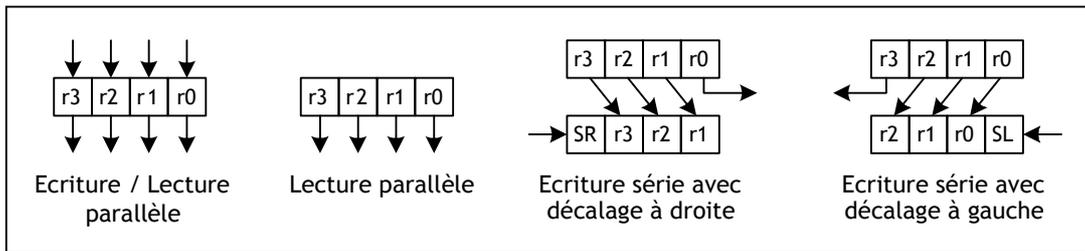
Dans un registre à décalage les bascules sont interconnectées de façon à ce que l'état logique de la bascule de rang  $i$  puisse être transmis à la bascule de rang  $i+1$  (ou  $i-1$ ) quand un signal d'horloge est appliqué à l'ensemble des bascules.

L'information peut être chargée de deux manières :

- Entrée parallèle : comme dans le cas d'un registre de mémorisation ;
- Entrée série : l'information est présentée séquentiellement bit après bit à l'entrée de la première bascule. Le décalage peut alors être vers la gauche ou la droite.

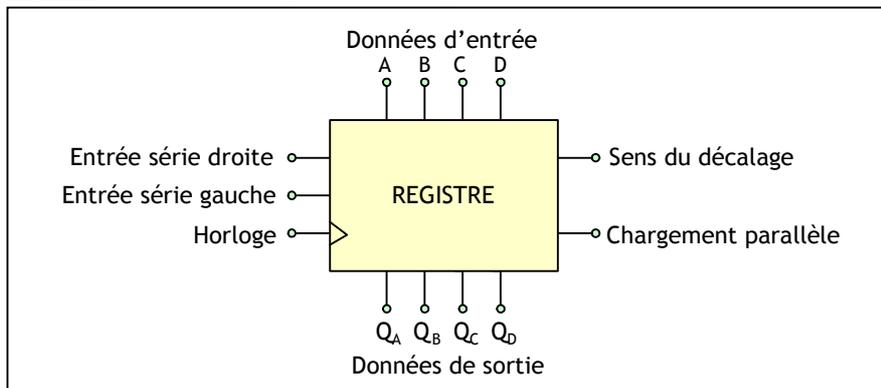
De même, l'information peut être lue en série ou en parallèle. La figure 17 résume les modes de fonctionnement d'un registre à décalage :

**Fig. 17:** Modes de fonctionnement d'un registre à décalage



Un registre à décalage universel aura donc la structure de la figure 18 :

**Fig. 18:** Structure d'un registre à décalage universel



Parmi les registres universels, on trouve le 74194 qui est un registre à chargement parallèle ou série, avec la possibilité d'un déplacement de l'information vers la droite (QA vers QD) ou la gauche (QD vers QA). La description et le schéma de brochage sont donnés à la figure 19 :

**Fig. 19:** Brochage et table de fonctionnement du registre universel 74LS194

CONNECTION DIAGRAM DIP (TOP VIEW)										MODE SELECT — TRUTH TABLE																																																																																																				
VCC	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	CP	S <sub>1</sub>	S <sub>0</sub>																																																																																																							
16	15	14	13	12	11	10	9																																																																																																							
										<table border="1"> <thead> <tr> <th rowspan="2">OPERATING MODE</th> <th colspan="6">INPUTS</th> <th colspan="4">OUTPUTS</th> </tr> <tr> <th>MR</th> <th>S<sub>1</sub></th> <th>S<sub>0</sub></th> <th>D<sub>SR</sub></th> <th>D<sub>SL</sub></th> <th>P<sub>n</sub></th> <th>Q<sub>0</sub></th> <th>Q<sub>1</sub></th> <th>Q<sub>2</sub></th> <th>Q<sub>3</sub></th> </tr> </thead> <tbody> <tr> <td>Reset</td> <td>L</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>L</td> <td>L</td> <td>L</td> <td>L</td> </tr> <tr> <td>Hold</td> <td>H</td> <td>I</td> <td>I</td> <td>X</td> <td>X</td> <td>X</td> <td>Q<sub>0</sub></td> <td>Q<sub>1</sub></td> <td>Q<sub>2</sub></td> <td>Q<sub>3</sub></td> </tr> <tr> <td rowspan="2">Shift Left</td> <td>H</td> <td>h</td> <td>I</td> <td>X</td> <td>I</td> <td>X</td> <td>Q<sub>1</sub></td> <td>Q<sub>2</sub></td> <td>Q<sub>3</sub></td> <td>L</td> </tr> <tr> <td>H</td> <td>h</td> <td>I</td> <td>X</td> <td>h</td> <td>X</td> <td>Q<sub>1</sub></td> <td>Q<sub>2</sub></td> <td>Q<sub>3</sub></td> <td>H</td> </tr> <tr> <td rowspan="2">Shift Right</td> <td>H</td> <td>I</td> <td>h</td> <td>I</td> <td>X</td> <td>X</td> <td>L</td> <td>Q<sub>0</sub></td> <td>Q<sub>1</sub></td> <td>Q<sub>2</sub></td> </tr> <tr> <td>H</td> <td>I</td> <td>h</td> <td>h</td> <td>X</td> <td>X</td> <td>H</td> <td>Q<sub>0</sub></td> <td>Q<sub>1</sub></td> <td>Q<sub>2</sub></td> </tr> <tr> <td>Parallel Load</td> <td>H</td> <td>h</td> <td>h</td> <td>X</td> <td>X</td> <td>P<sub>n</sub></td> <td>P<sub>0</sub></td> <td>P<sub>1</sub></td> <td>P<sub>2</sub></td> <td>P<sub>3</sub></td> </tr> </tbody> </table>					OPERATING MODE	INPUTS						OUTPUTS				MR	S <sub>1</sub>	S <sub>0</sub>	D <sub>SR</sub>	D <sub>SL</sub>	P <sub>n</sub>	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Reset	L	X	X	X	X	X	L	L	L	L	Hold	H	I	I	X	X	X	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Shift Left	H	h	I	X	I	X	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	L	H	h	I	X	h	X	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	H	Shift Right	H	I	h	I	X	X	L	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	H	I	h	h	X	X	H	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Parallel Load	H	h	h	X	X	P <sub>n</sub>	P <sub>0</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>
OPERATING MODE	INPUTS						OUTPUTS																																																																																																							
	MR	S <sub>1</sub>	S <sub>0</sub>	D <sub>SR</sub>	D <sub>SL</sub>	P <sub>n</sub>	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>																																																																																																				
Reset	L	X	X	X	X	X	L	L	L	L																																																																																																				
Hold	H	I	I	X	X	X	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>																																																																																																				
Shift Left	H	h	I	X	I	X	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	L																																																																																																				
	H	h	I	X	h	X	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	H																																																																																																				
Shift Right	H	I	h	I	X	X	L	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>																																																																																																				
	H	I	h	h	X	X	H	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>																																																																																																				
Parallel Load	H	h	h	X	X	P <sub>n</sub>	P <sub>0</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>																																																																																																				

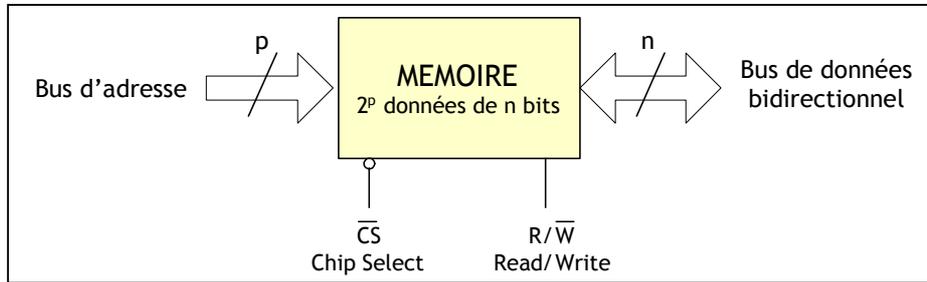
## 4. LES MEMOIRES :

### 4.1. Introduction :

Les systèmes modernes sont souvent à microprocesseur de tels systèmes exigent une capacité mémoire importante pour le stockage des données. Les mémoires électroniques ou à semi-conducteurs sont les éléments qui répondent à ce besoin.

Une mémoire est un dispositif capable d'emmagasiner puis de restituer une information. L'unité d'information (bit, octet, etc.) s'appelle «point mémoire » ou «cellule».

Fig. 19: Schéma fonctionnel d'une mémoire



- L'adresse fournie par le bus d'adresses est le mot binaire de  $p$  bits qui permet de localiser la donnée ;
- La donnée de  $n$  bits entre (écriture) et sort (lecture) par le bus de données qui est bidirectionnel : deux sens possibles, en liaison avec le signal  $R/W$  ;
- La mémoire peut stocker  $2^p$  données de  $n$  bits chacune ;
- Le signal  $CS$  permet la sélection du circuit ou le mettre en haute impédance ; cette possibilité permet, comme on le verra, l'extension de la capacité mémoire d'un système.

On peut donc utiliser une mémoire soit en :

- **lecture :**
  - ✓ Appliquer le mot adresse sur le bus d'adresse ;
  - ✓ Sélectionner le boîtier mémoire en appliquant un niveau logique bas sur la ligne  $CS$  ;
  - ✓ Sélectionner le mode lecture en appliquant un niveau logique haut sur la ligne  $R/W$  ;
- **écriture :**
  - ✓ Appliquer le mot d'adresse sur le bus d'adresse ;
  - ✓ Le mot de donnée sur le bus de données ;
  - ✓ Sélectionner le boîtier mémoire en appliquant un niveau logique bas sur la ligne  $CS$  ;
  - ✓ Sélectionner le mode écriture en appliquant un niveau logique bas sur la ligne  $R/W$  ;

### 4.2. Caractéristiques des mémoires :

- **La capacité :** c'est la quantité d'information qui peut être stockée dans la mémoire. Elle s'exprime en bits ou en mots de  $n$  bits. Par exemple :  
 $64b, 4Kb, 8Ko$  ( $o$  : octet ou byte ) avec  $1o = 8b$  ;  $1K = 2^{10} = 1024$  ;  $1M = 2^{20} = 1048576$
- **L'organisation :** elle définit le nombre de mots et la longueur de chaque mot. Par exemple :
  - ✓ Une mémoire de  $64K \times 1$  est constituée de 65536 mots de 1 bit. Sa capacité est donc de 64Kb (8Ko) ;
  - ✓ Une mémoire de  $8K \times 8$  contient 8192 mots de 8 bits. Sa capacité est de 64Kb (8Ko) ;
- **Le temps d'accès :** c'est le temps qui s'écoule entre une demande d'information et le moment où elle est effectivement disponible.

### 4.3. Les différents types de mémoires

Les mémoires sont classées suivant deux familles :

- Mémoires mortes (**ROM** pour Read Only Memory) : mémoire à lecture seule ;
- Mémoires vives (**RAM** pour Random Access Memory) : mémoire à lecture et écriture.

### 4.3.1. Les mémoires mortes :

Les ROM sont utilisées pour stocker des informations figées telles que des programmes fixes dans des machines programmées ou les tables de conversion de données.

Le contenu est fixé à la construction ou par l'utilisateur et la disparition de l'alimentation électrique n'altère pas le contenu.

### 4.3.2. Les mémoires vives

Dès qu'un système doit conserver temporairement des informations, la RAM trouve sa place. En informatique, elles sont largement mises en oeuvre en quantités importantes (plus de 16 Mo en microinformatique et plusieurs centaines de méga octets en mini-informatique).

### 4.3.3. Les mémoires programmables et effaçables par l'utilisateur

Les mémoires programmables sont intermédiaires entre les RAM et les ROM. Leur contenu peut être défini par l'utilisateur et subsister sans alimentation électrique.

On en rencontre de différentes familles :

- ✓ Les **PROM** (Programmable ROM) : sont composées de fusibles que l'on peut détruire une seule fois ;
- ✓ Les **EPROM** (Erasable PROM) : ce sont des mémoires effaçables par ultraviolet et programmables électriquement ;
- ✓ Les **EEPROM** (Electrical Erasable PROM) : ce sont des mémoires effaçables et programmables électriquement .

## 4.4. Extension de capacité :

Il est courant dans un système microinformatique de grouper plusieurs circuits pour augmenter la capacité (nombre des mots et/ou longueur des mots). Par exemple, à l'aide de 4 boîtiers mémoires de 1Kx4bits, on peut réaliser les mémoires suivantes : 1Kx16bits, 4Kx4bits, 2Kx8bits.

Le schéma de la figure 20 réalise une mémoire de 1Kx16bits à partir d'une mémoire élémentaire de 1Kx4bits :

- Nécessité de 4 boîtiers ;
- Nécessité de 10 bits d'adresses  $A_0$  à  $A_9$  ; (nombre de mots =  $2^{\text{nombre de bits d'adresse}}$ )
- Nécessité de 16 bits de données  $D_0$  à  $D_{15}$  ;

**Fig. 20:** Extension de format d'une mémoire

