

# Chap III - Logique séquentielle

## 1. Définitions

### 1.1. Logique séquentielle

La logique séquentielle, contrairement à la logique combinatoire fait intervenir le temps. La valeur d'une variable logique à l'instant  $t+1$  dépend de celles qu'avaient les variables logiques à l'instant  $t$ .

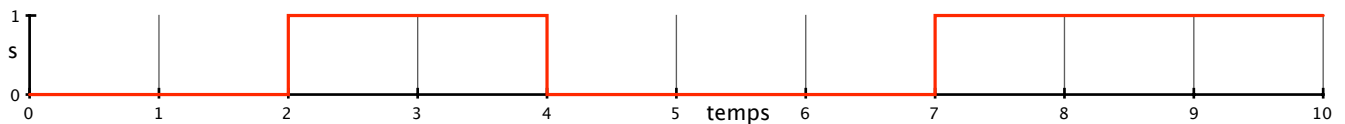
**Exemple :**

$$s_{t+1} = a_t + b_t \cdot s_t \text{ que l'on notera : } s = a + b \cdot s.$$

**Remarque :** On voit alors que  $s$  peut dépendre de  $s...$

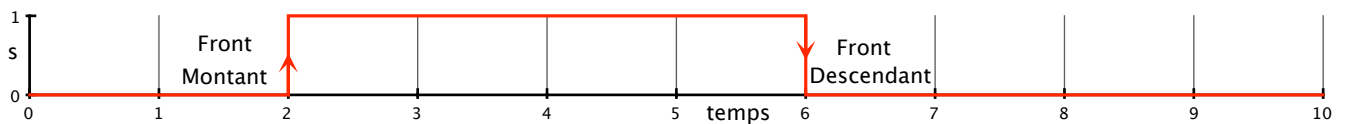
### 1.2. Chronogramme

Le chronogramme est une représentation graphique de l'évolution temporelle d'un signal électrique ou logique. On trouve sur l'axe des abscisses le temps, sur l'axe des ordonnées l'état (0 ou 1) des variables étudiées.



### 1.3. Niveau logique vs front

Une variable logique  $s$  peut avoir deux niveaux ; le niveau logique haut (vrai) et le niveau logique bas (faux). Quand elle passe du niveau bas vers le niveau haut, elle définit le front montant. Dans le cas contraire, elle définit le front descendant.



**Remarque :**

Le front montant de  $a$  correspond au front descendant de  $\bar{a}$ .

## 2. Les bascules RS (ou SR)

Une bascule est un circuit logique doté d'une sortie et d'une ou plusieurs entrées. Les changements d'état de la sortie sont déterminés par les signaux appliqués aux entrées. Ce qui différencie les bascules des circuits logiques combinatoires, c'est que la sortie maintient son état même après disparition du signal de commande, la bascule a une 'mémoire'.

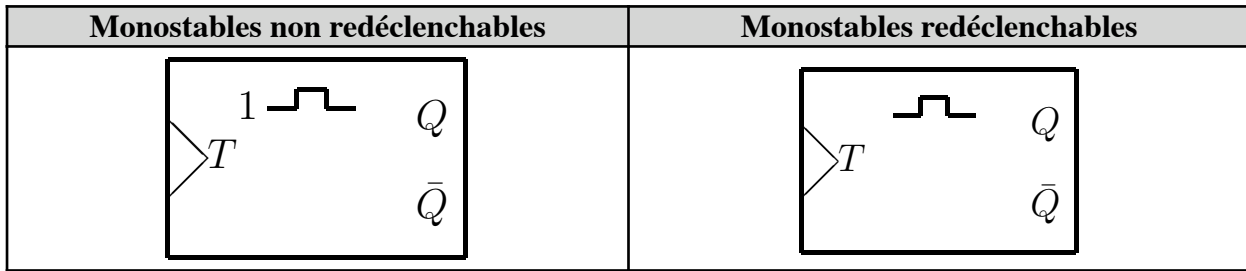
La bascule est l'élément de base de la logique séquentielle. En effet, en assemblant des bascules, on peut réaliser des compteurs, des registres, des registres à décalage, des mémoires.

La bascule RS à deux entrée, une pour mettre sa sortie  $Q$  à 1 ; S (set) et une pour mettre sa sortie  $Q$  à 0 ; R (reset).

Schématisation	Table de vérité															
	<table border="1"> <thead> <tr> <th>R</th> <th>S</th> <th><math>Q_{n+1}</math></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td><math>Q_n</math></td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>Interdit</td> </tr> </tbody> </table>	R	S	$Q_{n+1}$	0	0	$Q_n$	0	1	1	1	0	0	1	1	Interdit
R	S	$Q_{n+1}$														
0	0	$Q_n$														
0	1	1														
1	0	0														
1	1	Interdit														

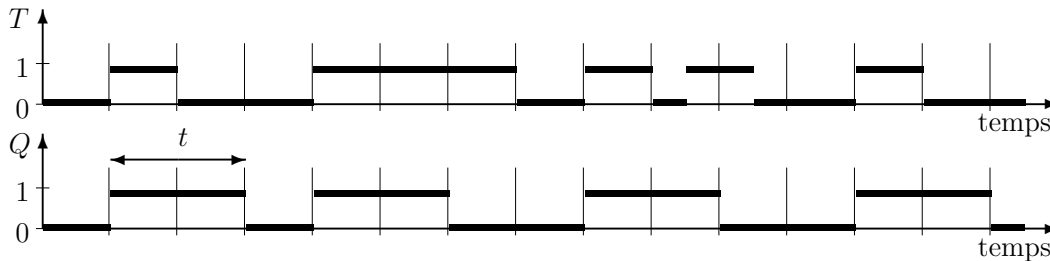
### 3. Monostables

Comme son nom l'indique ce composant ne connaît qu'un seul état stable. C'est l'état pour lequel sa sortie  $Q$  est à 0. Un front positif ou négatif sur son entrée  $T$  peut provoquer l'état instable qui dure un temps déterminé  $t_0$ . Il existe des monostables redéclenchables et des non redéclenchables.



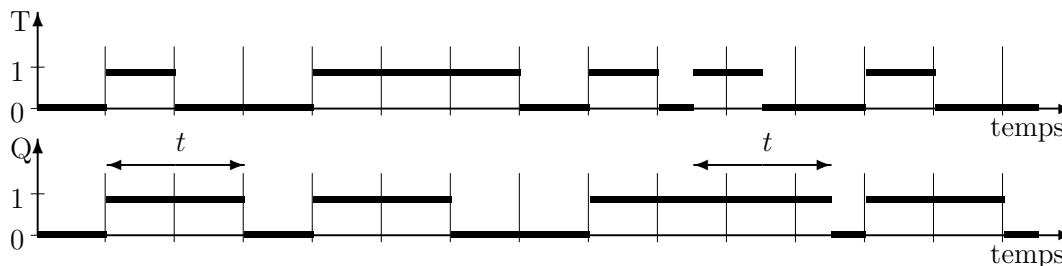
#### 3.1. Monostable non redéclenchable

Le front de  $T$  déclenche le monostable et la sortie  $Q$  passe à 1 pendant une durée  $t_0$  même si l'impulsion est plus longue. Les fronts de  $T$  n'ont aucune influence quand  $Q$  est à 1. Le monostable doit revenir dans son état stable pour être redéclenché.



#### 3.2. Monostable redéclenchable

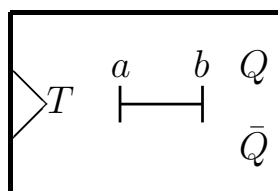
Le front de  $T$  déclenche le monostable quelque soit la valeur de la sortie  $Q$  qui passe ou reste à 1 pendant une durée  $t_0$  même si l'impulsion est plus longue.



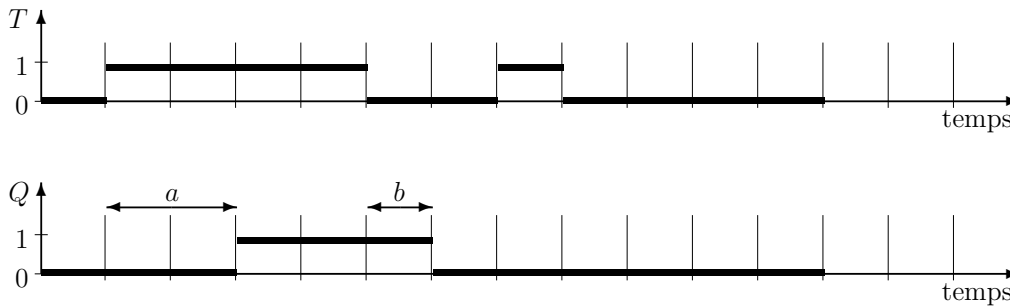
### 4. Temporisateur

Le temporisateur retarde le front montant d'un signal  $T$  d'un temps  $a$  et le front descendant du même signal d'un temps  $b$ .

Schématisation :



Le chronogramme sera alors le suivant :



## 5. Opérations numériques sur des mots

### 5.1. Définition

Un mot, en informatique, est l'unité de base manipulée par un microprocesseur. La taille d'un mot s'exprime en bits ou en octets, et est souvent utilisée pour classer les microprocesseurs (8 bits, 16 bits...). Toutes choses égales par ailleurs, un microprocesseur est d'autant plus rapide que ses mots sont longs, car les données qu'il traite à chaque opération sont plus longues. (d'après Wikipédia)

1 0 0 1 0 0 1 1

Mots de 8 bits

### 5.2. Masquage

Les opérateurs logiques (NON, ET, OU, OU exclusif) que nous avons vu précédemment, peuvent être appliqué sur un mot, bit par bit.

**Exemple :**

1 0 0 1 0 0 1 1

ET

0 0 0 0 1 1 1 1

=

0 0 0 0 0 0 1 1

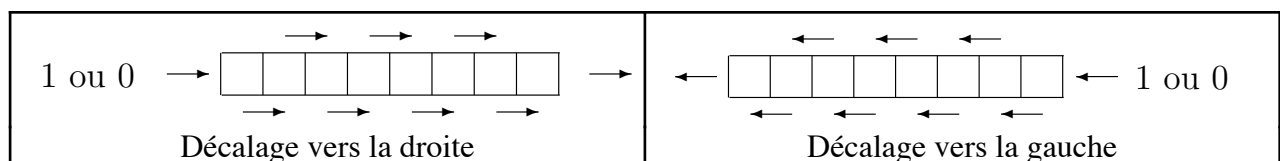
Ce type de fonction est très utilisé pour masquer une partie d'un mot. Dans l'exemple ci-dessus, les 4 bits de poids forts sont masqués.

**Remarque :**

$(M \oplus K) \oplus K = M$ , cette égalité permet de faire du codage de mot avec une clef privé K.

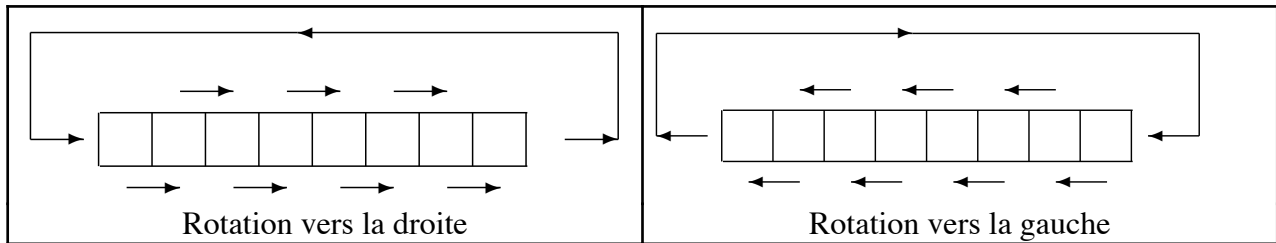
### 5.3. Décalage

Le décalage consiste à décaler les bits d'un mot soit vers la droite, soit vers la gauche et de remplacer la place vacante par un bit à 0 ou 1.



### 5.4. Rotation

La rotation est une permutation circulaire bit à bit d'un mot. Cette rotation se fait soit vers la droite, soit vers la gauche.



### 5.5. Compteur

Un compteur est un ensemble de  $n$  bascules interconnectées par des portes logiques. Il peut donc mémoriser des entiers codés en binaire naturel sur  $n$  bits. Il est généralement muni :

- d'une entrée **RAZ** pour le mettre à 0 ;
- d'une entrée **CU**, pour incrémenter le compteur ;
- d'une entrée **CD** pour le décrémenter ;
- d'une mémoire pour mémoriser la présélection **CP** ;
- d'une sortie débordement décomptage **E** ;
- d'une sortie présélection atteinte **D** ;
- d'une sortie débordement comptage **F**.

**Schématisation :**

