

# Chapitre I

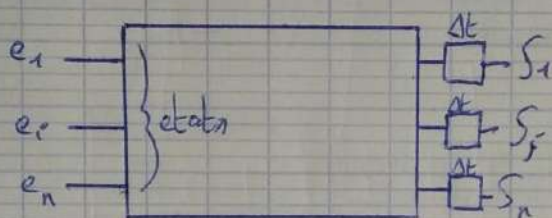
## Introduction à la logique séquentielle

### I) Généralités

#### 1) Logique combinatoire

$S_j$  en fonction des  $e_i$ :

$$S_j = f_j(e_1, \dots, e_i, \dots, e_n) \quad j \in \{1, \dots, m\}$$



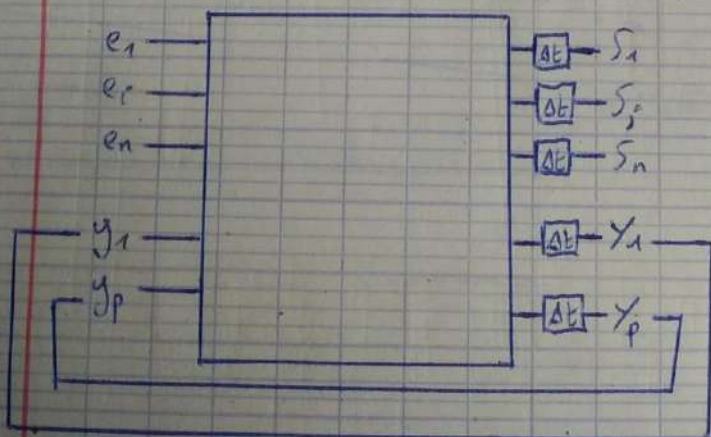
$n$ : nombre d'entrées

$m$ : nombre de sorties

#### 2) Logique séquentielle

Dans un système séquentiel, l'état ne dépend pas uniquement de la combinaison des entrées à un instant donné mais de la séquence précédente et de son état initial.

Un système séquentiel est un système combinatoire bouclé.



### 3) Systèmes séquentiels synchrones et asynchrones

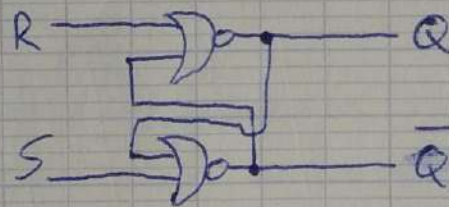
Asynchrone : L'état des sorties change en fonction d'un changement sur les entrées. La sortie stable ne dépend que de la succession des états transitoires.

Synchrone : Les sorties changent en fonction d'un signal de synchronisation appelé "horloge".

## II) Bascules

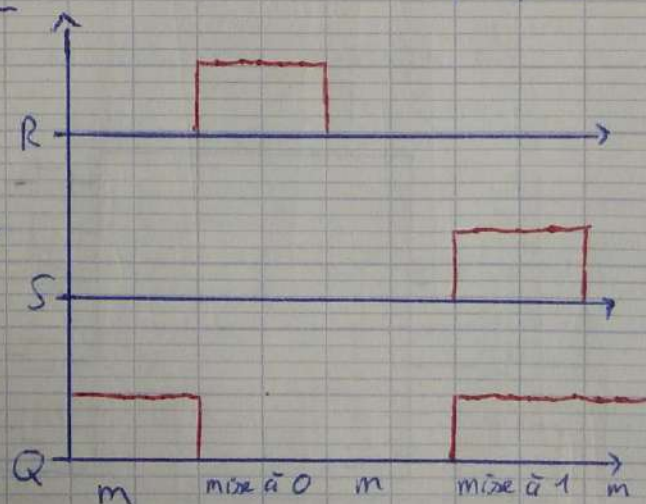
### 1) Bascule asynchrone

L> Bascule RS



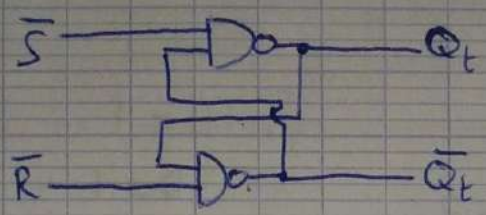
A	B	NOR $\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

R	S	$Q_{t+1}$	$\overline{Q}_{t+1}$
0	0	$Q_t$	$\overline{Q}_t$
0	1	0	1
1	0	0	1
1	1	interdit	



Au départ  $Q = 1$

↳ Bascule  $\bar{R}\bar{S}$



$\bar{R}$	$\bar{S}$	$Q_{\text{ent}}$	$\bar{Q}_{\text{ent}}$
0	0	interdit	
0	1	0	1
1	0	1	0
1	1	$Q_t$	$\bar{Q}_t$

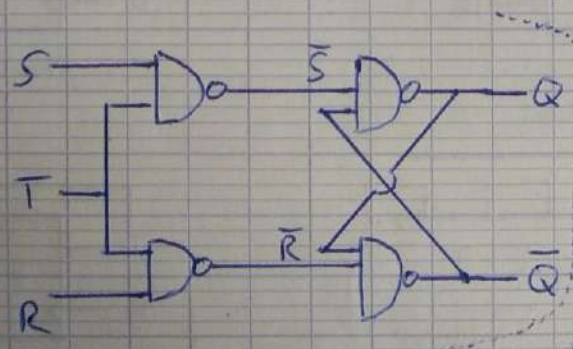
2) Bascule synchrone

Trois types de synchronisation :

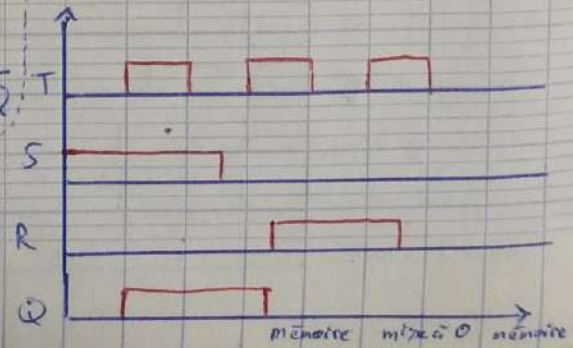
- Sur niveau  
niveau haut ou bas de l'horloge
- Sur front  
↑ ou ↓
- Sur impulsion  
↑↓ ou ↓↑

a) Bascules déclenchées sur niveau

↳ Bascule RST



$T=1 \rightarrow RS$   
 $T=0 \rightarrow$  Mémoire

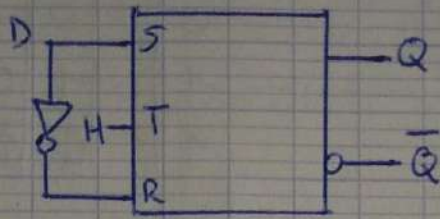


Au départ  $Q=0, t=0$

mémoire    mise à 0    mémoire

# L) Bascule D latch

D: Delay



Si  $H=1$ :

$$D=1 \rightarrow Q=1$$

$$D=0 \rightarrow Q=0$$

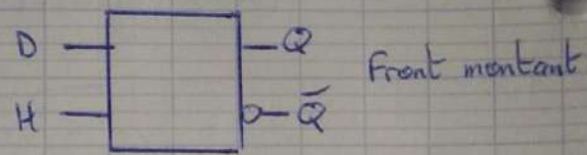
Si  $H=0$ :

Mémoire

D	H	$Q_{t+1}$
0	1	0
1	1	1
X	0	$Q_t$

recopie l'entrée

Mémoire

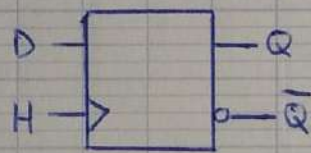


## b) Bascules déclenchées sur front

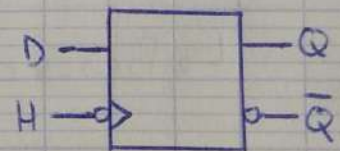
### L) Bascule D edge Triggered

↑ ou ↓

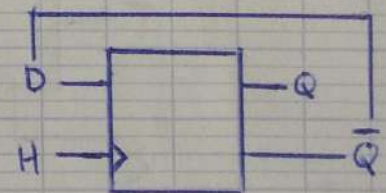
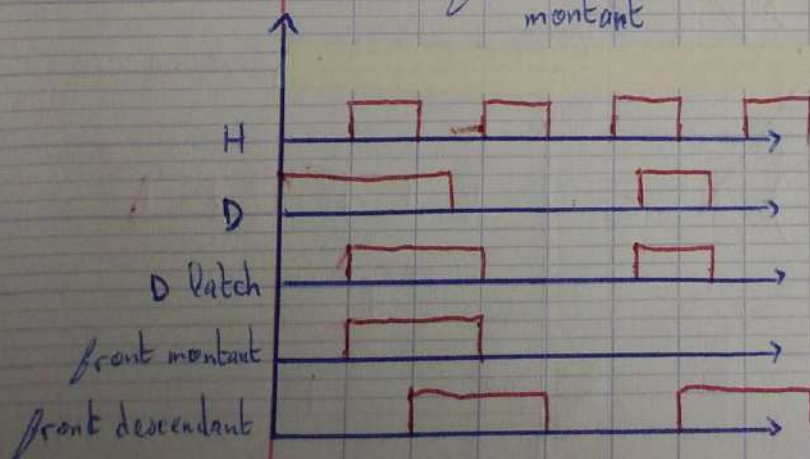
H	D	$Q_{t+1}$
↑ ou ↓	0	0
↑ ou ↓	1	1
X	X	$Q_t$



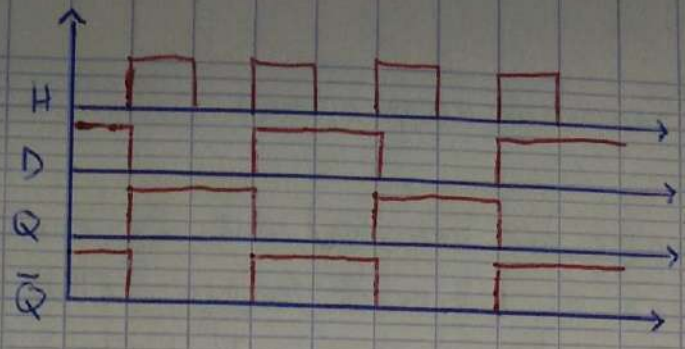
Front montant



Front descendant

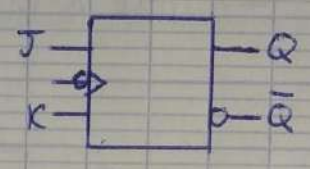
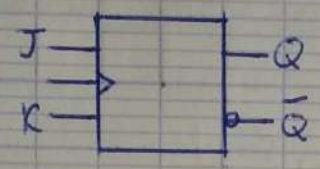


### 3 Systèmes numériques

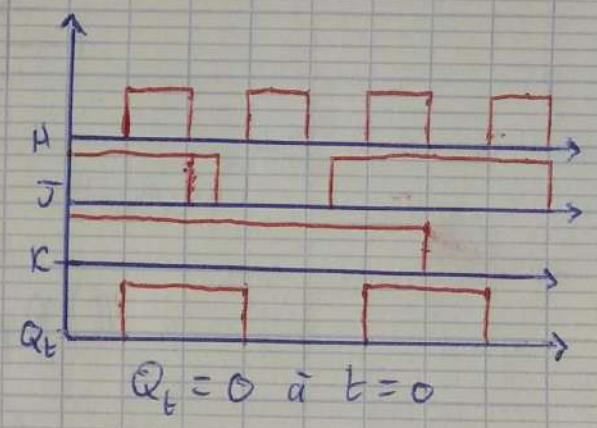


→ double la période et divise la fréquence par 2

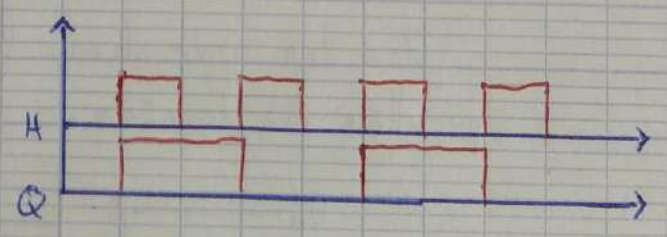
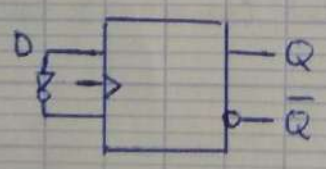
### ↳ Bascule JK



	J	K	$Q_{t+1}$
faux	0	0	$Q_t$ mémoire
"	0	1	0
"	1	0	1
"	1	1	$\bar{Q}_t$ toggle
X	X	X	mémoire



### Remarque :

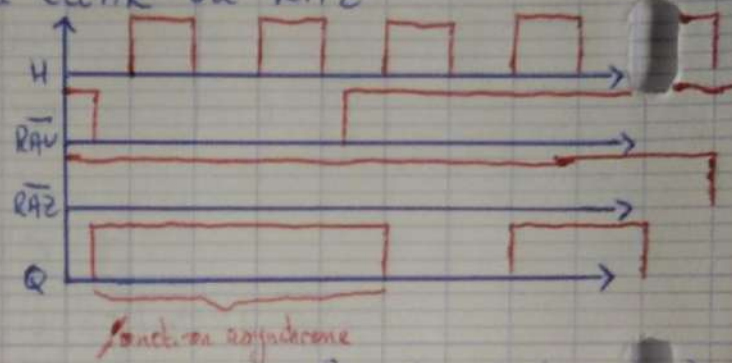
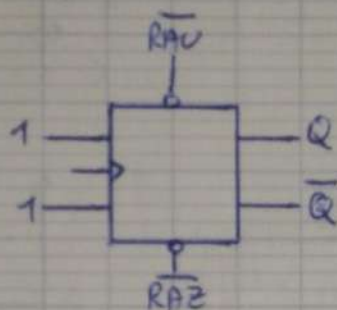


### 3) Bascules asynchrones

La plupart des bascules sont également munies d'entrées asynchrones ou entrées de forçage. Elles ne dépendent pas de l'horloge et sont prioritaires. Il existe 2 types d'entrées asynchrones :

L> Mise à 1 : SET ou PRESET ou RAV

L> Mise à 0 : RESET ou CLEAR ou RAZ

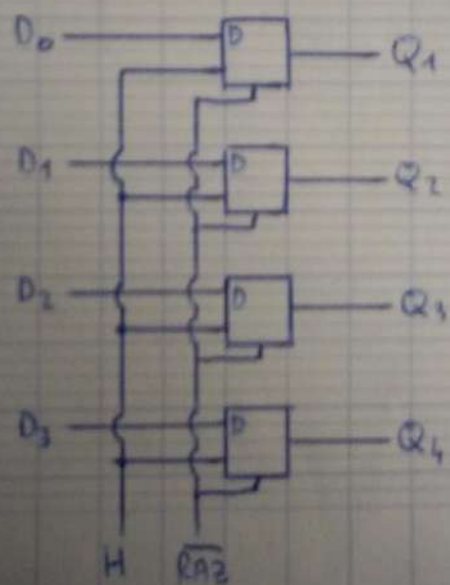


Ces entrées sont actives sur niveau bas (la plupart du temps).

### III) Application de bascule

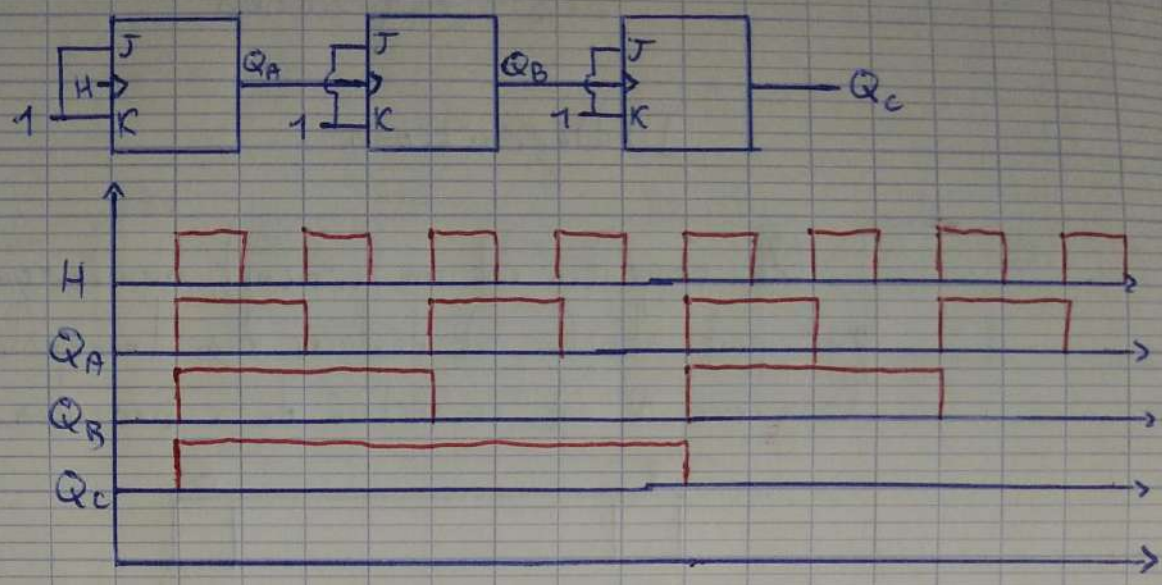
#### 1) Stockage des données

Une des applications les plus communes est le stockage parallèle des données.

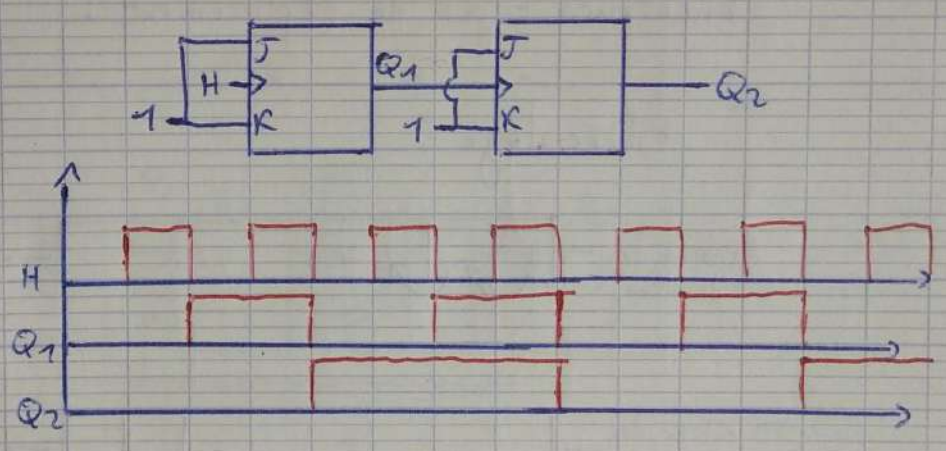


Sorties en parallèle

### 2) Division de fréquence



### 3) Comptage

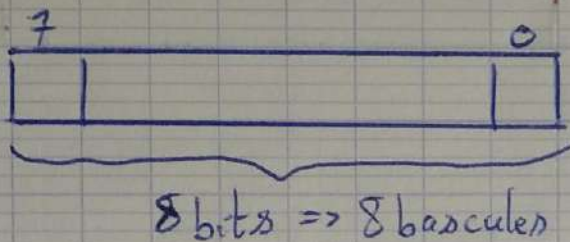


# Chapitre II

## Les registres

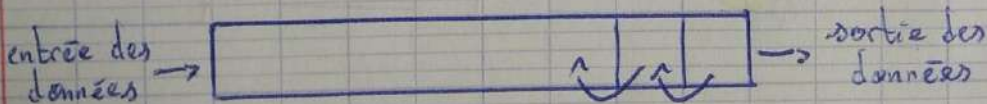
### I) Définitions

Un registre est un circuit numérique à deux fonctions qui peut stocker et déplacer des données. Un registre de longueur  $N$  est constitué de  $n$  bascules.



Plusieurs types de déplacement :

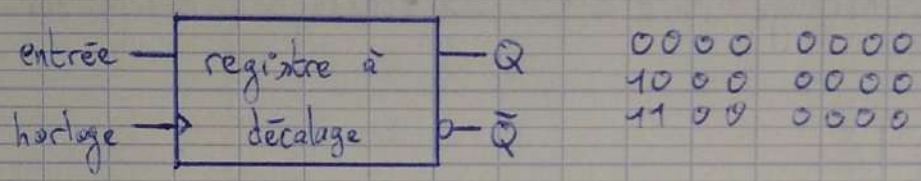
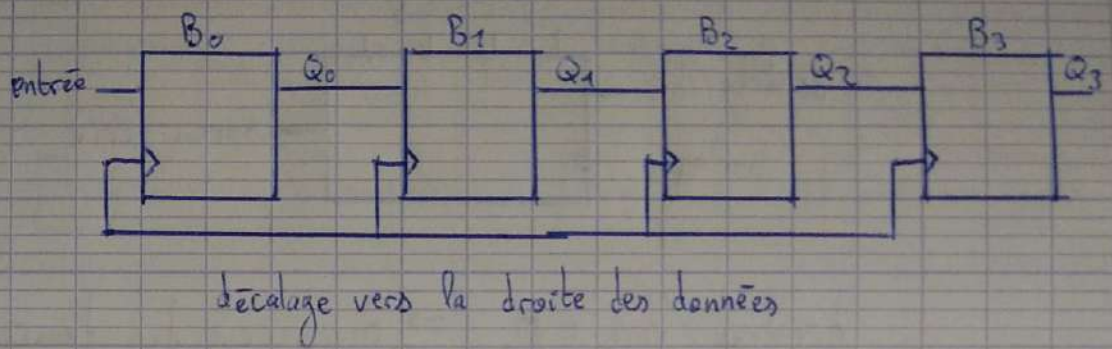
\* Décalage :



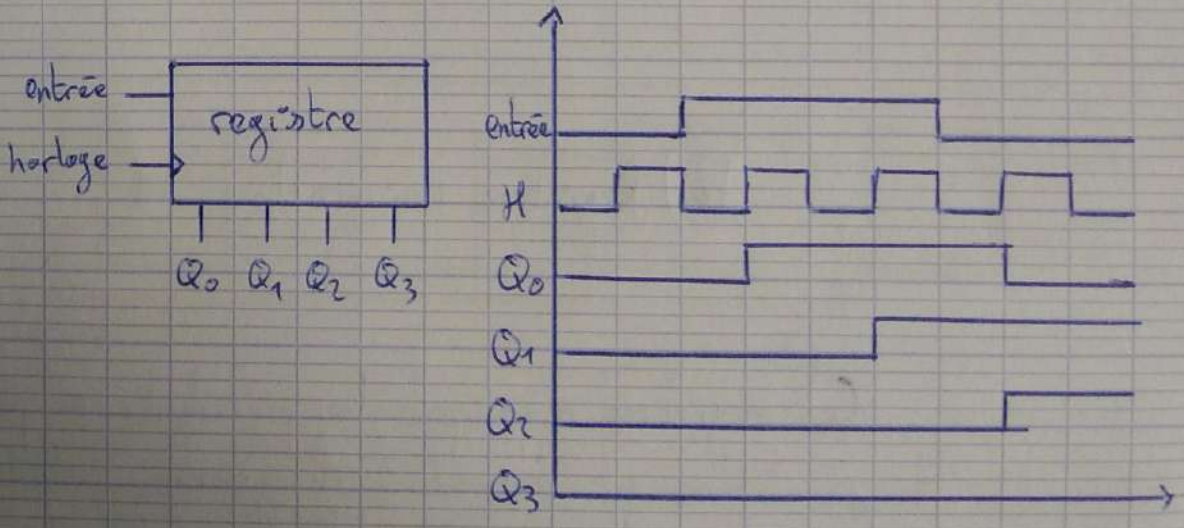
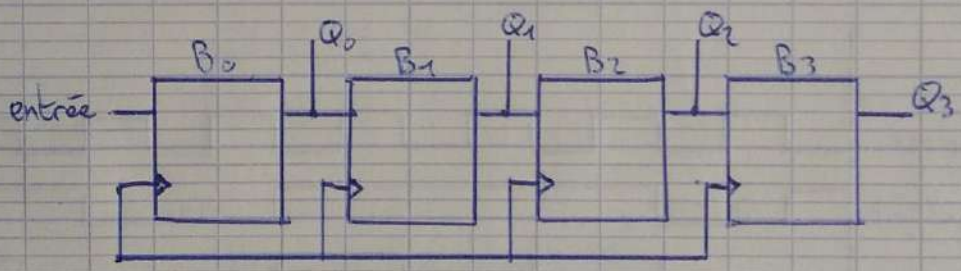


## II) Registre à décalage à entrée série

### 1) Sortie série

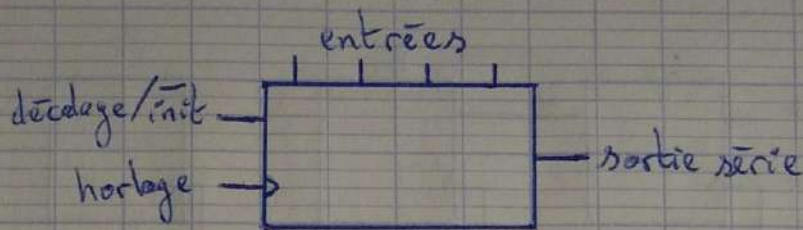


### 2) Sortie parallèle

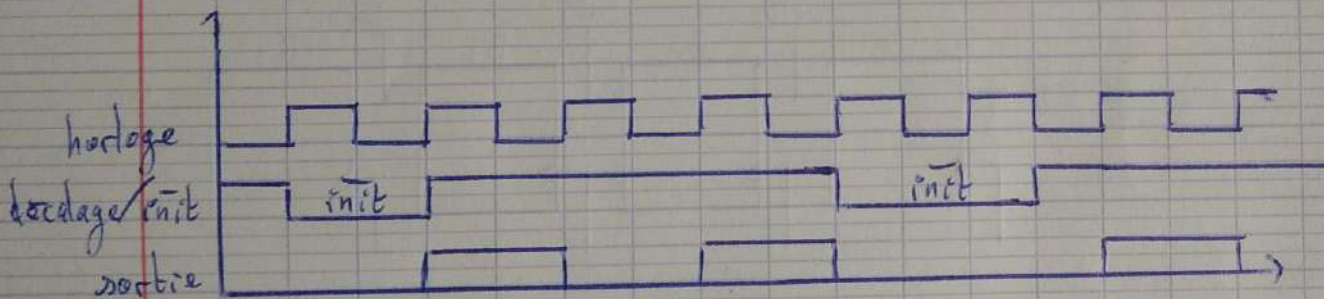
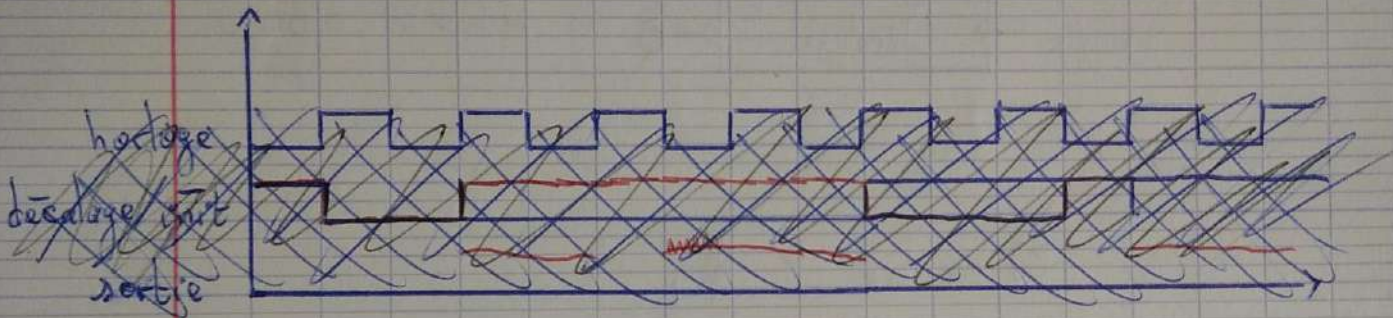


### III) Registre à décalage à entrées parallèles

#### 1) Sortie série



$\text{décalage/init} = 0 \Rightarrow$  chargement des données dans le registre  
 $\text{décalage/init} = 1 \Rightarrow$  décalage à droite des données

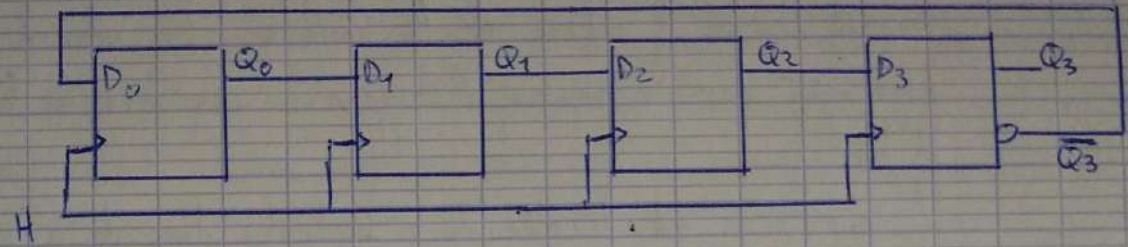


### IV) Registres universels

Ces registres regroupent les précédents en permettant les modes de fonctionnement suivants : chargement et lecture parallèles, chargement et lecture série, décalage à droite et à gauche.

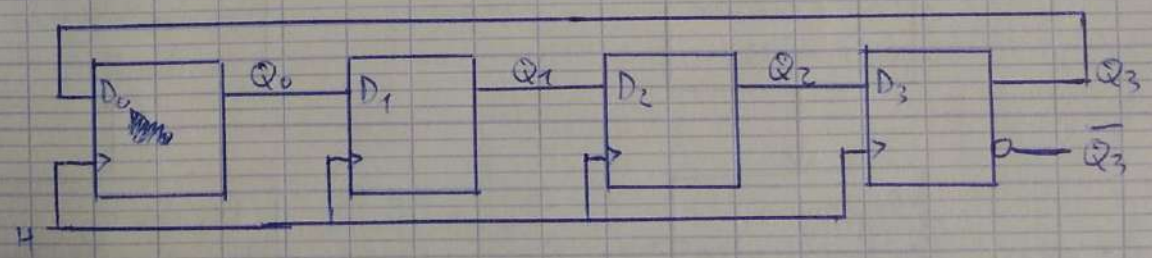
## V) Compteurs à registre à décalage

### 1) Compteur de Johnson



Transitions d'horloge	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

### 2) Compteur en anneau



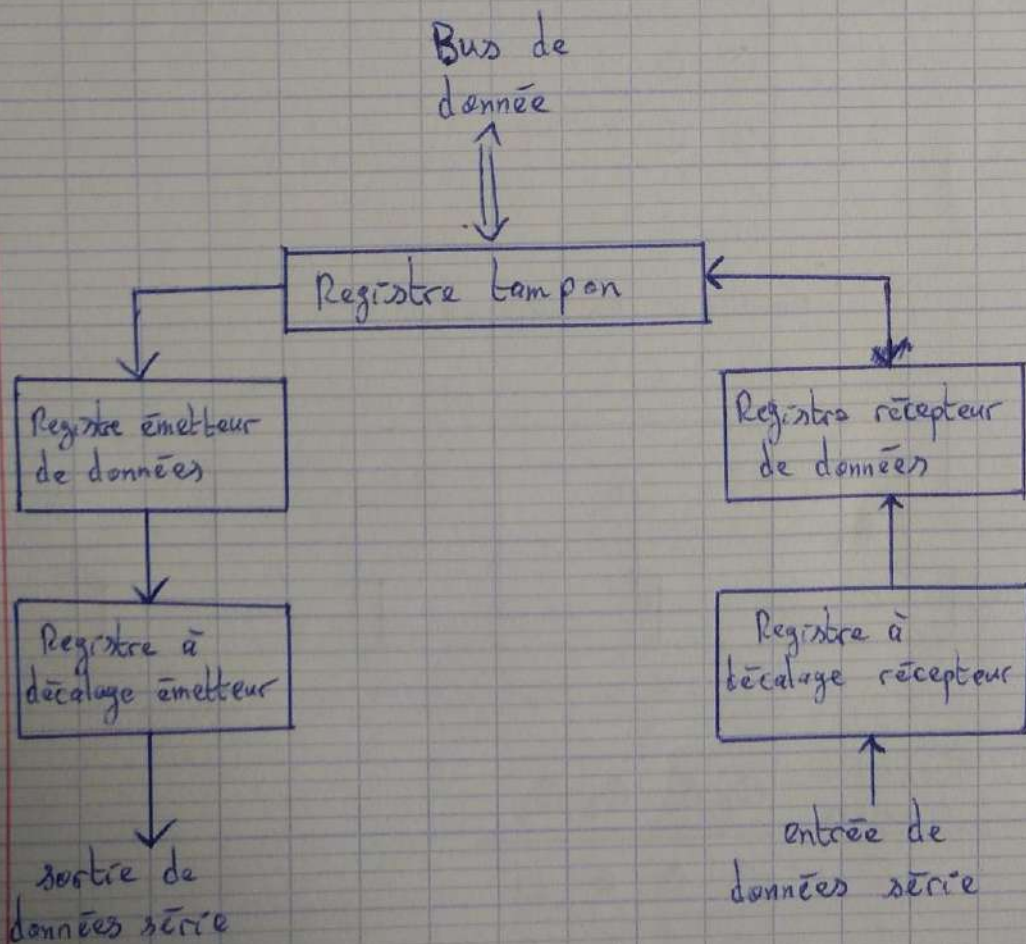
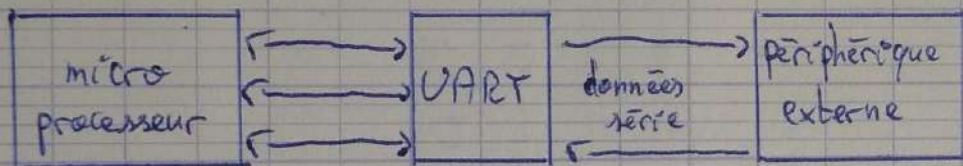
Transitions d'horloge	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0

## VI) Retard numérique

Un registre, entrée série / sortie série, peut être utilisé pour créer un retard entre l'entrée et la sortie. Ce retard numérique est déterminé par le nombre d'étage  $n$  du registre et la fréquence du signal d'horloge.

L'impulsion est décalée de  $n$  bascules donc de  $n$  étages.

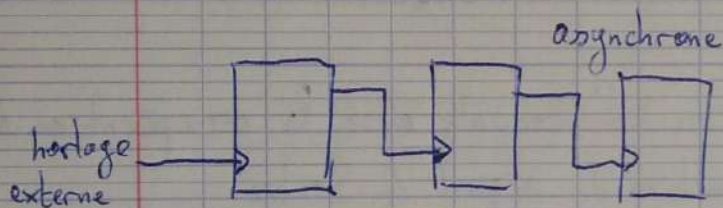
## VII) Émetteur - récepteur asynchrone universel (UART)



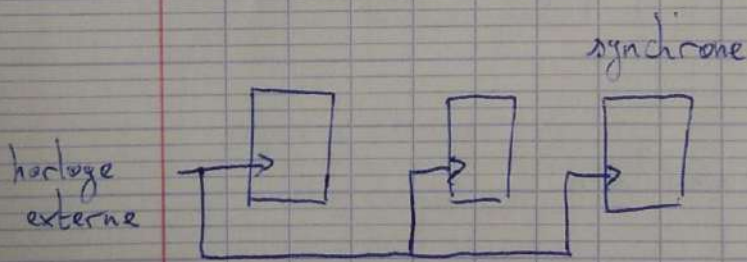
### Chapitre III Les compteurs

Les compteurs sont des éléments essentiels de la logique séquentielle car ils permettent d'établir une relation d'ordre de succession d'événements. Leur emploi ne se limite pas au système arithmétique, ils sont utiles partout où il est souhaitable de définir une suite d'états. L'état d'un compteur est défini par le nombre binaire formé avec l'ensemble de sorties des bascules.

Ils sont définis en deux catégories : les compteurs asynchrones (ou compteur à propagation ou compteur série) et les compteurs synchrones (ou parallèles).



La propagation de l'ordre des changements d'états se fait en cascade.

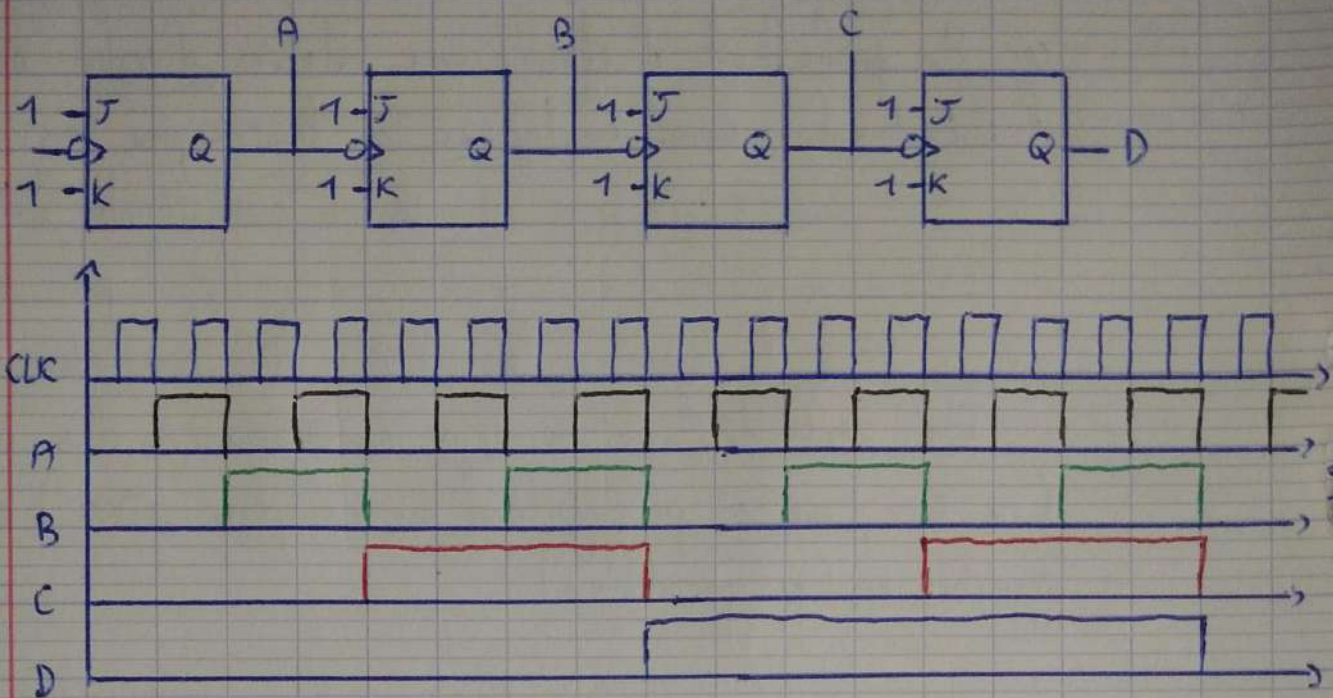


Le signal d'horloge est connecté à toutes les bascules.

Les compteurs sont définis par le nombre de bascules, le nombre d'états, et leur fréquence.

# I) Compteurs asynchrones

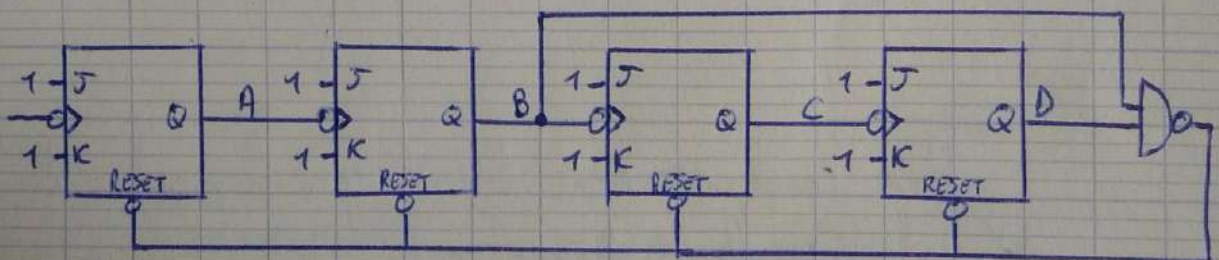
## 1) Le compteur binaire



Retard de propagation : L'intervalle entre la récupération du signal d'entrée et la production du signal de sortie.

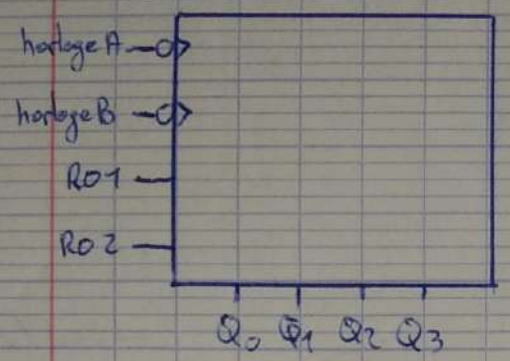
Bascule JK : retard de propagation de 10ns

## 2) Compteur modulo n



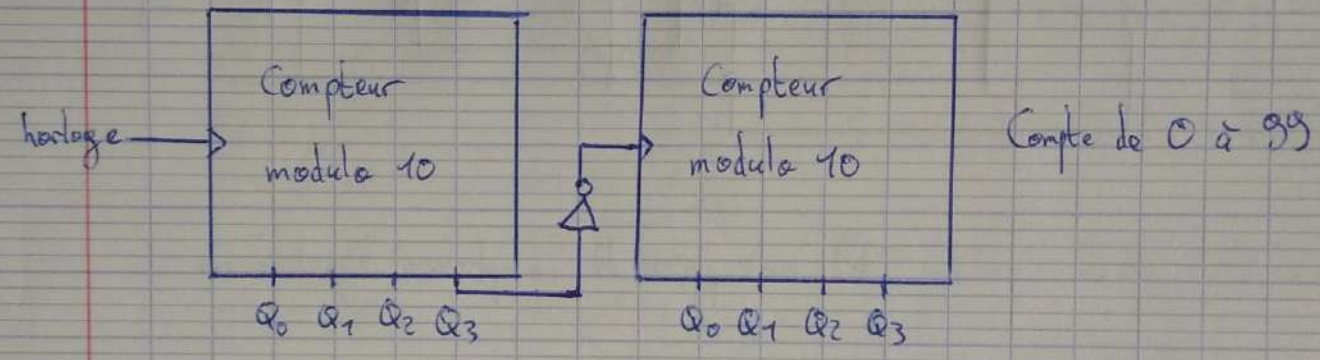
On utilise une porte logique pour reset.

### 3) Circuits intégrés



R01 et R02 sont des entrées de remise à 0.  
Remise à 0 si R01 = R02 = 1

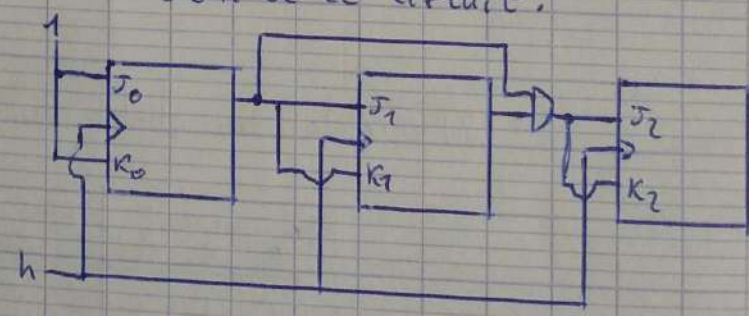
Exemple : Compteur décimal :



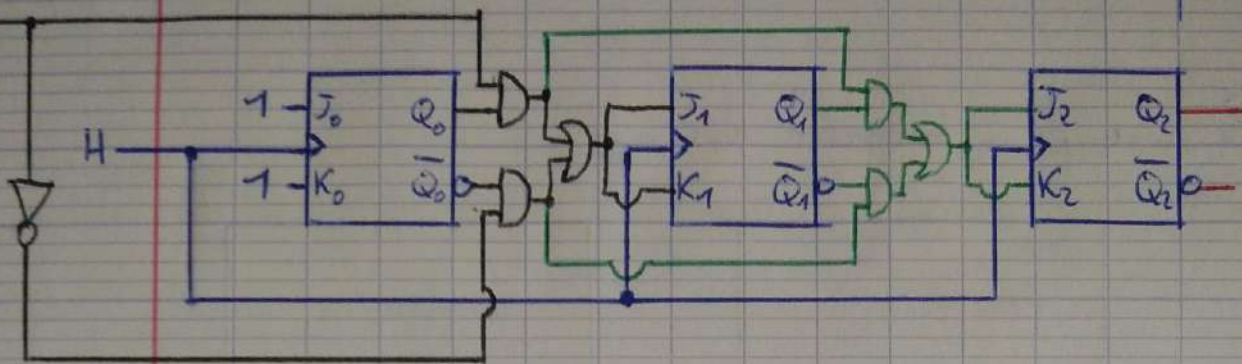
### II) Compteurs synchrones

impulsion d'horloge	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
état initial	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
0	0	0	0

Le schéma de ce circuit :

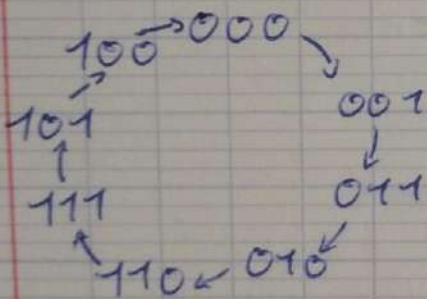


Compteur / décompteur



### 3) Conception de compteurs synchrones

A) Diagramme d'états :



Le diagramme des états illustre la séquence des états occupés après chaque impulsion d'horloge.

B) Table des états suivants :

Etat présent			Etat suivant		
$Q_2$	$Q_1$	$Q_0$	$Q_2$	$Q_1$	$Q_0$
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0



c) Table de transition de la bascule JK :

J	K	$Q_{t+1}$	$Q_t \rightarrow Q_{t+1}$	J	K
0	0	$Q_t$	0 $\rightarrow$ 0	0	$\emptyset$
0	1	0	0 $\rightarrow$ 1	1	$\emptyset$
1	0	1	1 $\rightarrow$ 0	$\emptyset$	1
1	1	$\overline{Q_t}$	1 $\rightarrow$ 1	$\emptyset$	0

d) Diagramme de Karnaugh et équation de sortie :

$\emptyset$  ou X  
idem

$\overline{J_0}$	00	01
00	1 $\emptyset$	$\emptyset$
01	0 $\emptyset$	$\emptyset$
11	1 $\emptyset$	$\emptyset$
10	0 $\emptyset$	$\emptyset$

$K_0$	00	01
00	$\emptyset$ 0	$\emptyset$ 1
01	$\emptyset$ 1	0 0
11	0 0	$\emptyset$ 1
10	$\emptyset$ 1	$\emptyset$ 1

$$\overline{J_0} = \overline{Q_2} \overline{Q_1} + Q_2 Q_1 = \overline{Q_1 \oplus Q_2}$$

$$K_0 = \overline{Q_2} Q_1 + Q_2 \overline{Q_1} = Q_1 \oplus Q_2$$

$\overline{J_1}$	0	1
00	0 1	$\emptyset$
01	$\emptyset$ $\emptyset$	$\emptyset$
11	$\emptyset$ $\emptyset$	$\emptyset$
10	0 0	$\emptyset$

$K_1$	0	1
00	$\emptyset$ $\emptyset$	$\emptyset$
01	0 0	$\emptyset$
11	0 1	$\emptyset$
10	$\emptyset$ $\emptyset$	$\emptyset$

$$\overline{J_1} = \overline{Q_2} Q_0$$

$$K_1 = Q_2 Q_0$$

$\overline{J_2}$	0	1
00	0 0	$\emptyset$
01	1 0	$\emptyset$
11	$\emptyset$ $\emptyset$	$\emptyset$
10	$\emptyset$ $\emptyset$	$\emptyset$

$K_2$	0	1
00	$\emptyset$ $\emptyset$	$\emptyset$
01	$\emptyset$ $\emptyset$	$\emptyset$
11	0 0	$\emptyset$
10	1 0	$\emptyset$

$$\overline{J_2} = Q_1 \overline{Q_0}$$

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

$$J_1 = K_1 = Q_0 \bar{Q}_1 + Q_3 Q_2$$

$Q_3 \backslash Q_2$	00	01	11	10
00	0	0	X	X
01	X	0	0	X
11	X	X	1	X
10	X	1	0	X

$$J_0 = K_0 = \bar{Q}_0 + Q_3 Q_2$$

$Q_3 \backslash Q_2$	00	01	11	10
00	1	0	X	X
01	X	0	0	X
11	X	X	1	X
10	X	0	0	X

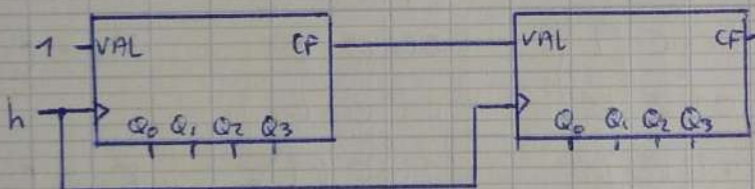
$$J_2 = K_2 = Q_0$$

$Q_3 \backslash Q_2$	00	01	11	10
00	0	1	X	X
01	X	1	1	X
11	X	X	1	X
10	X	1	1	X

$$J_3 = K_3 = Q_2 + Q_3 \bar{Q}_1$$

$Q_3 \backslash Q_2$	00	01	11	10
00	0	0	X	X
01	X	1	1	X
11	X	X	1	X
10	X	1	0	X

#### IV) Montage en cascade



CF: Compteur Final (retenue)

0000<sub>2</sub>  
 0001<sub>2</sub>  
 .....  
 1111<sub>2</sub>  
 0000<sub>2</sub>

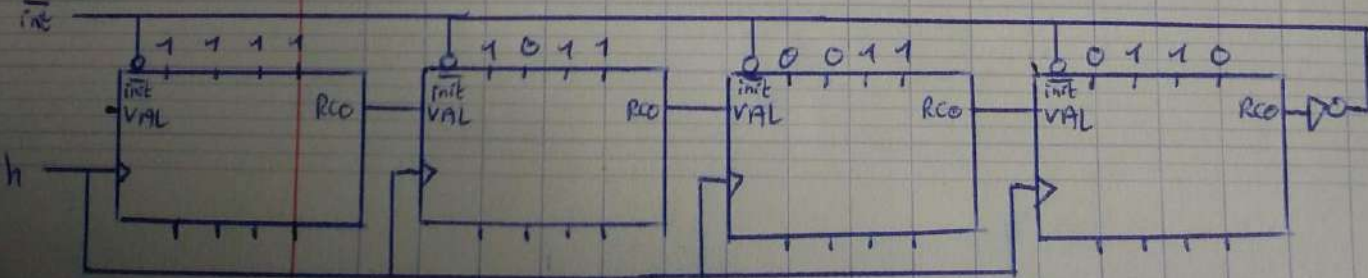
Mise en cascade avec des fréquences tronquées

On veut un compteur jusque 40 000.

4 compteurs binaires de 4 bits  $\rightarrow$  compte de 0 à  $2^{16} - 1$  soit 65 535.

Il faut reset à 40 000 :  $65\,535 - 40\,000 = 25\,535 = 63BF_{16}$ .

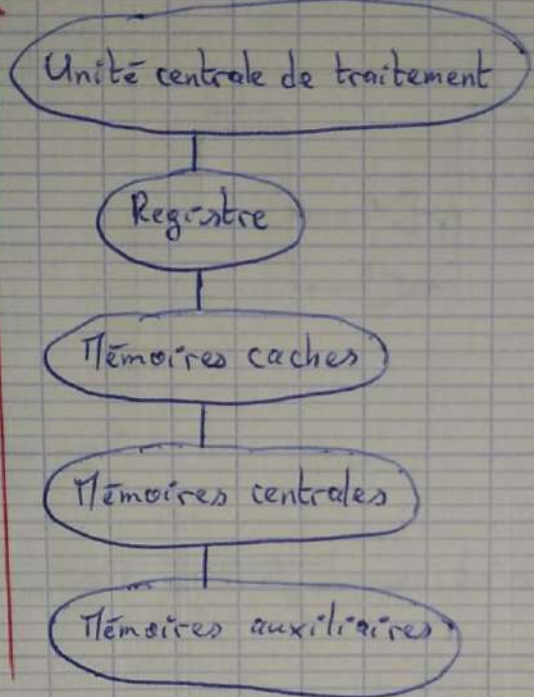
00000<sub>2</sub>  
 .....  
 25 535<sub>2</sub>  
 .....  
 65 535<sub>2</sub>



# Chapitre IV Mémoire

## I) Généralités

vitesse



quelques ns

quelques dizaine de ns

une centaine de ns

ms

Bande passante = largeur bus de données \* fréquence de la mémoire

Accès direct : accès direct via adresse mémoire

Accès séquentiel : parcourir toute la mémoire

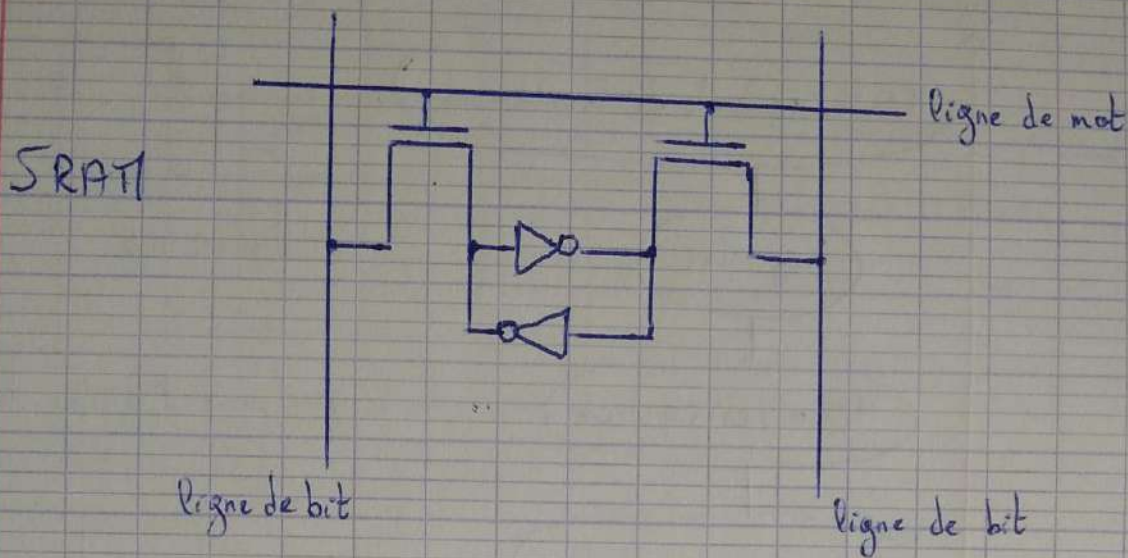
Accès semi-séquentiel : comme les HDD

types de mémoires

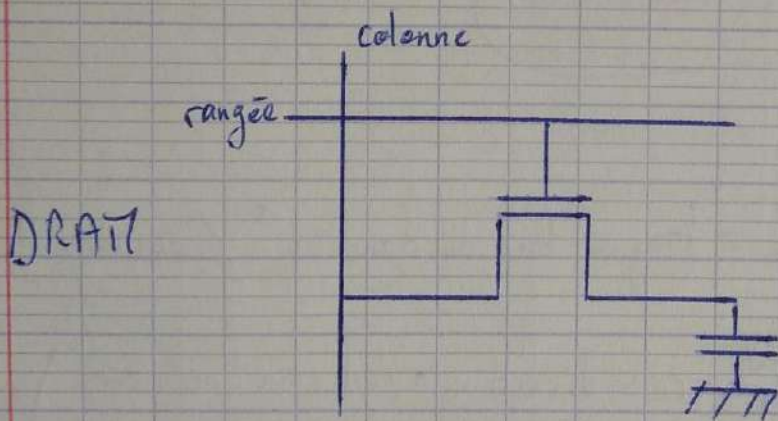
## II) Mémoire centrale

### 1) Mémoire RAM

Il existe 2 types de RAM: statiques et dynamiques.



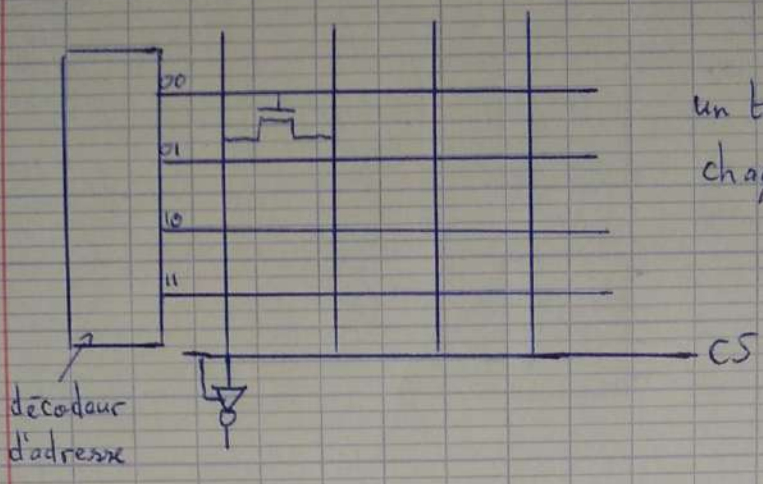
La mémoire cache est souvent faite de SRAM. (plus grande que la DRAM, mais pas besoin de rafraichissement)



DRAM stocke la valeur dans un condensateur. Moins cher et plus simple qu'une mémoire statique, mais un condensateur possède un courant de fuite et il faut régulièrement recharger la valeur (rafraichir).

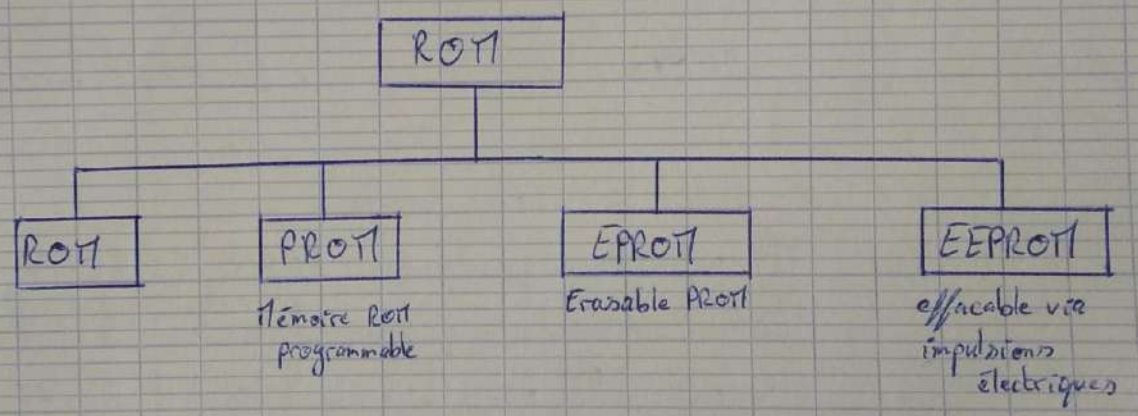
## 2) Mémoire morte

### A) ROM (Read Only Memory)



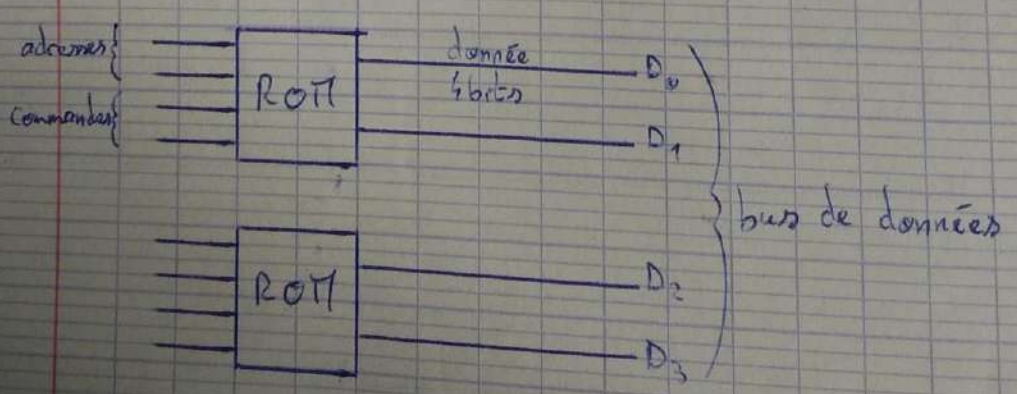
un transistor entre chaque ligne et chaque colonne

### B) Catalogue de mémoires ROM



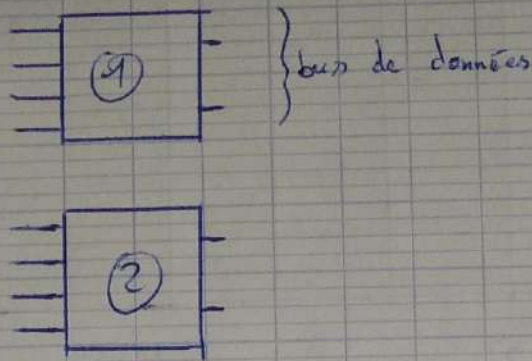
## 3) Constitution de blocs mémoires

### A) Extension d'un mot mémoire



## B) Extension de la capacité mémoire

On prend deux mémoires de 2ko et on cherche à réaliser une mémoire de 4ko.



## C) Décodage d'adresse

~~128ko~~ 64ko  $2^6 \times 2^{10} \Rightarrow 16$  bits d'entrée

Trois blocs de mémoire morte :

M1 est 4ko dans la zone \$0000 à \$0FFF

M2 est 8ko dans la zone \$1000 à \$2FFF

M3 est 16ko dans la zone \$3000 à \$6FFF

Chaque bloc mémoire a un bit de sélection  $\overline{CS}$ .

$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$		$A_0$	
0	0	0	0	---	0	\$0000
0	0	0	0	11	1	\$0FFF
0	0	0	1	00	0	\$1000
0	0	1	0	1	111	\$2FFF
0	0	1	1	0	0	\$3000
0	1	1	0	1	1	\$6FFF

$$\overline{CS}_2 = \overline{A_{15}} \overline{A_{14}} (\overline{A_{13}} A_{12} + A_{13} \overline{A_{12}})$$

$$\overline{CS}_3 = \overline{A_{15}} (\overline{A_{14}} A_{13} A_{12}) + A_{14} [(\overline{A_{13}} + \overline{A_{12}}) + \overline{A_{13}} A_{12} + A_{13} \overline{A_{12}}]$$