

# Systèmes numériques

I/ Les opérateurs séquentiels

II/ Registres

III/ Les compteurs

IV/ Les mémoires

V/ Introduction au micro-contrôleur

Chapitre I : Les opérateurs séquentiels

I/ Généralités

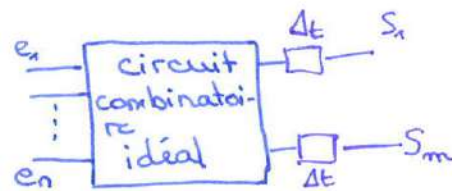
1.1) Circuits combinatoires

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

n: nb d'entrées

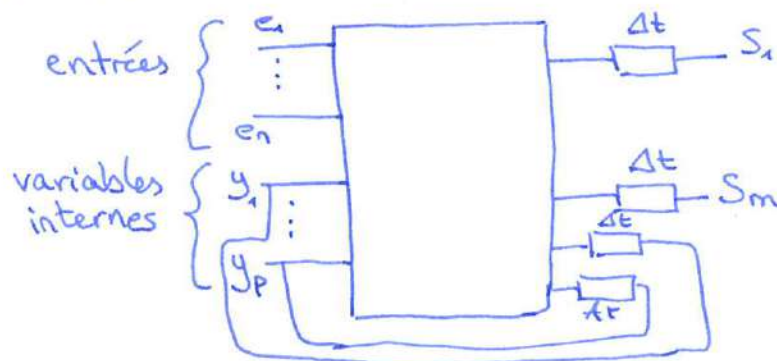
m: nb de sorties

Un circuit combinatoire à n entrées est un système à  $2^n$  états.



1.2) Circuits séquentiels

Dans un circuit séquentiel l'état dépend de l'état des entrées mais aussi de la séquence précédant ces entrées et aussi de l'état initial.



Ces variables internes vont dépendre des sorties précédentes.

Un circuit combinatoire est bouclé.

### 1.3) Logique séquentielle synchrone et asynchrone

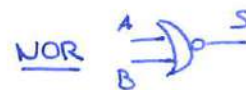
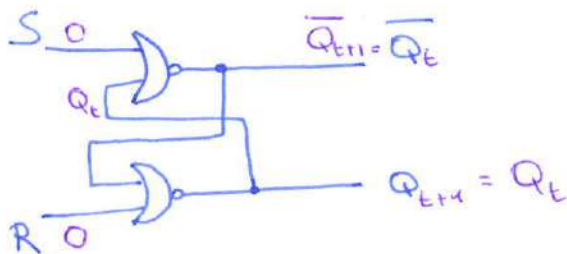
Dans un circuit séquentielle synchrone les sorties changent en fonction d'un signal de synchronisation appelé horloge.

Dans un circuit séquentielle asynchrone l'état des sorties changent uniquement en fonction des entrées et des variables internes. Ces circuits sont moins fiables et plus sensible aux parasites que les circuits synchrones. Ils sont donc moins utilisés.

## II/ des bascules

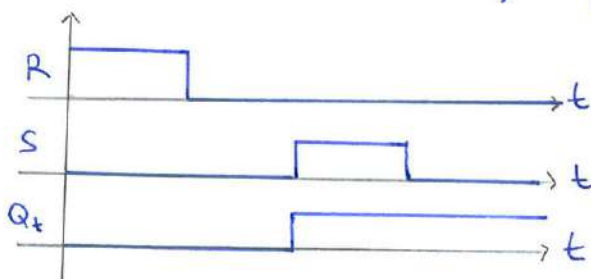
### 2.1) Les bascules asynchrones

#### \* Bascule RS



A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

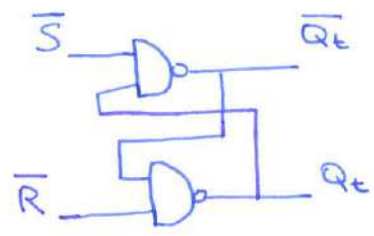
S	R	$Q_{t+1}$	$\overline{Q_{t+1}}$	
0	0	$Q_t$	$\overline{Q_t}$	mémoire
0	1	0	1	mise à 0
1	0	1	0	mise à 1
1	1	état	interdit	parce que $Q_t = \overline{Q_t}$



\* Bascule  $\overline{R}\overline{S}$

NAND

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



$\overline{S}$	$\overline{R}$	$Q_{t+1}$	$\overline{Q_{t+1}}$
0	0	état	interdit
0	1	1	0
1	0	0	1
1	1	$Q_t$	$\overline{Q_t}$ mémoire

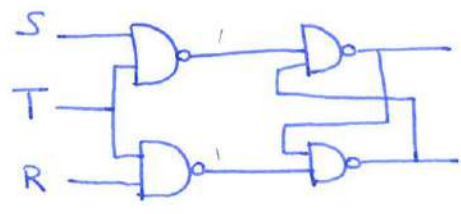
2.2) Les bascules synchrones

On a trois types de synchronisation :

- synchronisation sur niveau, le changement d'état se fera en fonction du niveau 0 ou 1 du signal d'horloge
- synchronisation sur fond d'horloge, les bascules changent d'état uniquement sur fond montant  $\uparrow$  ou descendant  $\downarrow$  de l'horloge
- synchronisation par impulsion, les bascules changent d'état après deux fonds d'horloge

2.2.1) Synchronisation sur niveau

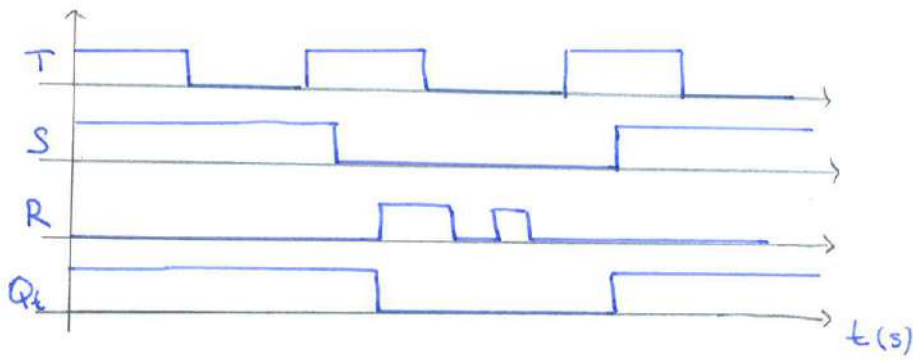
\* Bascule RST



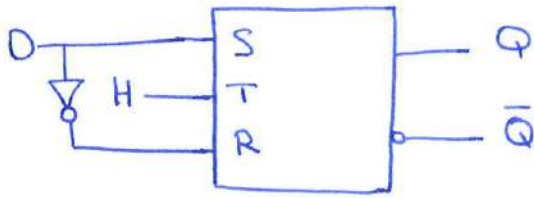
$T=0$ , la bascule fonctionne en mémoire  
 $T=1$ , la bascule est une bascule RS.

T	S	R	$Q_{t+1}$
0	X	X	$Q_t$
1	0	0	mémoire
1	0	1	0
1	1	0	1
1	1	1	interdit





\* Bascule D latch



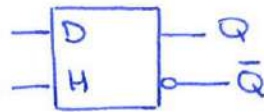
H = 0 mémoire

H = 1  $\begin{cases} Q = 1 (D=1) \\ Q = 0 (D=0) \end{cases}$

D	T	Q
X	0	mémoire
0	1	0
1	1	1

} recopie l'entrée

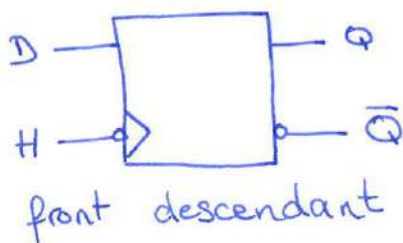
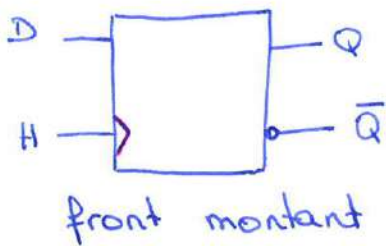
représentation :



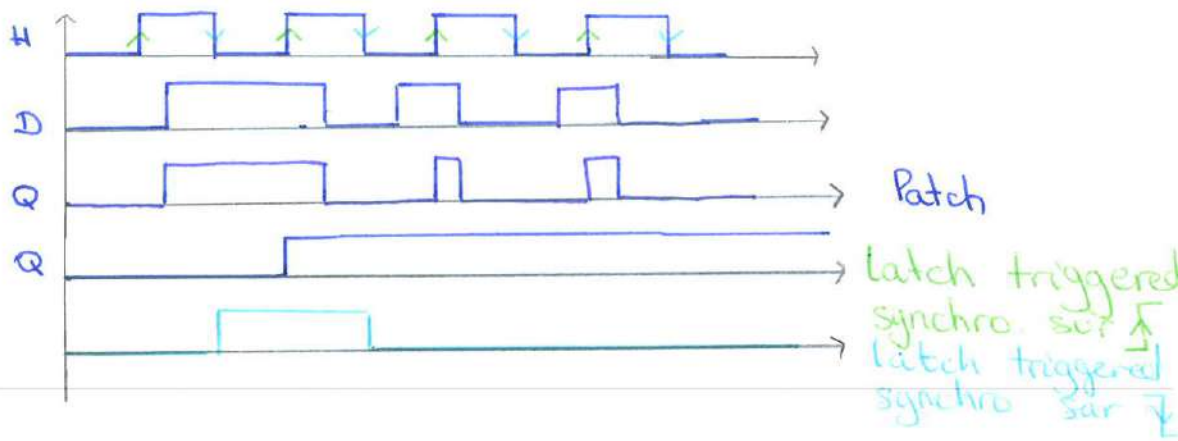
2.2.2) Bascules déclenchées sur front

montant ou descendant

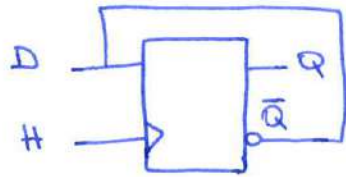
\* Bascule D latch triggered



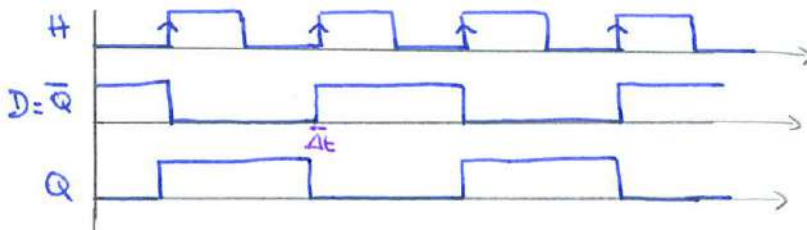
⚠ la seule façon de différencier la bascule D latch de la bascule D latch triggered est le Δ.



ex 2 D Patch triggered  $\uparrow$



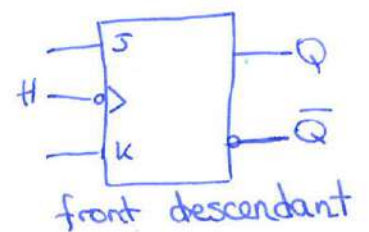
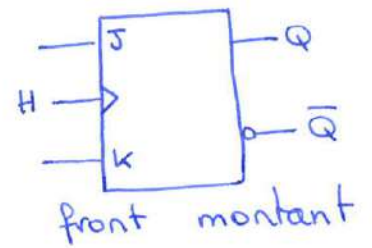
Au départ  $D = 1$

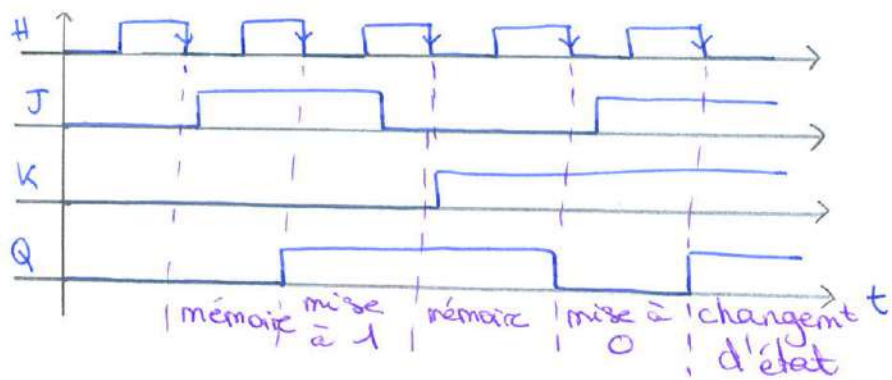


C'est un diviseur de fréquence par deux

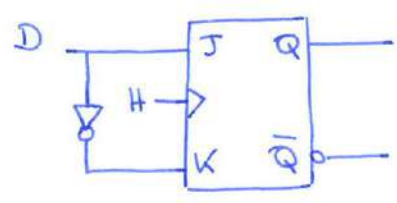
\* Bascule JK

H	J	K	$Q_{t+1}$
↑ ou ↓	0	0	$Q_t$ mémoire
"	1	0	1
"	0	1	0
"	1	1	$\overline{Q}_t$ basculement

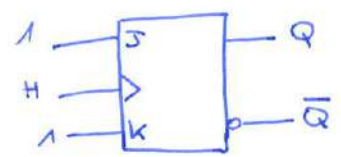




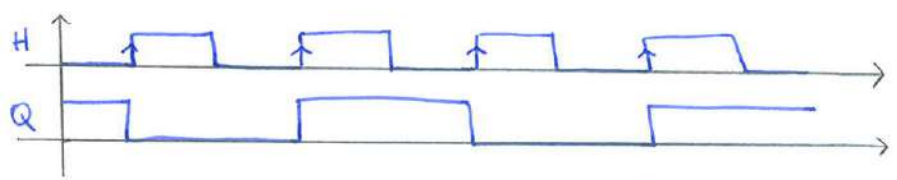
On peut fabriquer un bascule D avec une JK: latch triggered



ex 3

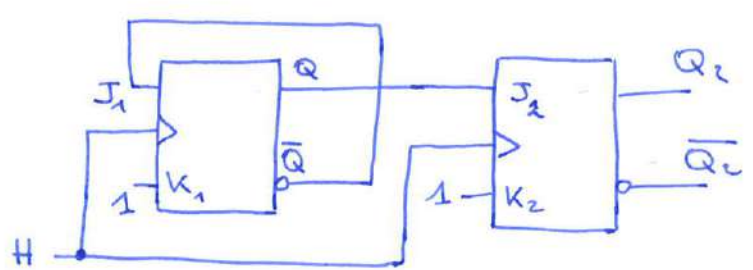


Au départ  $Q = 1$

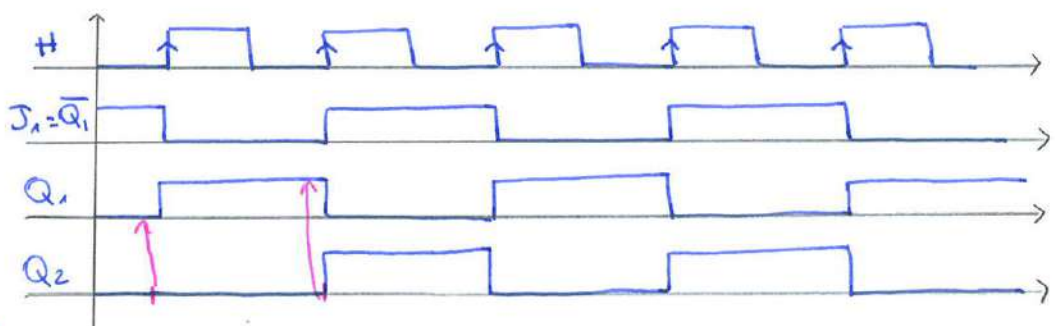


Ce circuit divise par 2 la fréquence

ex 4



Au départ  $Q_1 = 0$  et  $Q_2 = 0$



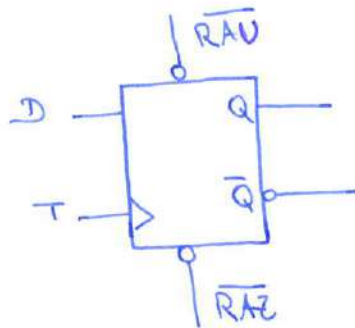
⚠ Pour  $Q_2$ , on regarde juste avant le front d'horloge

## 2.2.3) Entrées asynchrones

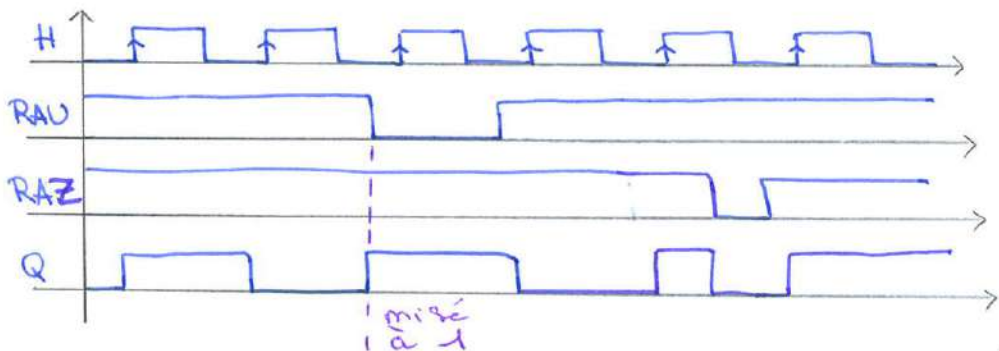
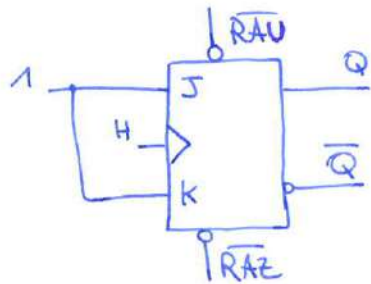
La plupart des bascules sont également munies d'entrées asynchrones ou de forçages. Elles ne dépendent pas de l'horloge et sont prioritaires. On a deux commandes :

- la commande de mise à 1  $\Rightarrow$  SET  
PRESET  
RAU

- la commande de mise à 0  $\Rightarrow$  RESET  
CLEAR  
RAZ



ex





# Chapitre II: Les registres

## I / Définition

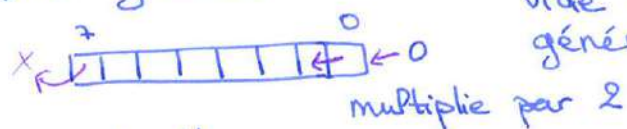
Un registre est un circuit numérique à deux fonctions. Il peut stocker des données et déplacer des données. Un registre de longueur  $N$  est constitué de  $N$  bascules.

Un registre stocke les données provisoirement en attendant son traitement ultérieur.

## II / Fonctions de décalages ou de rotations

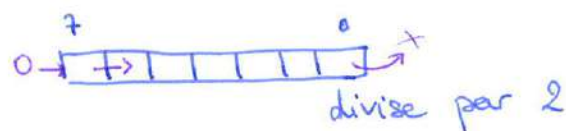
⇒ Décalage

→ à gauche

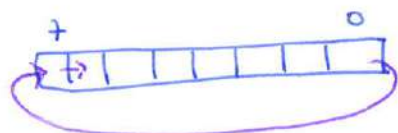
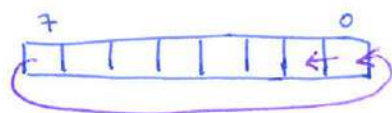


Comme la case 0 est vide on ajoute généralement un 0.

→ à droite



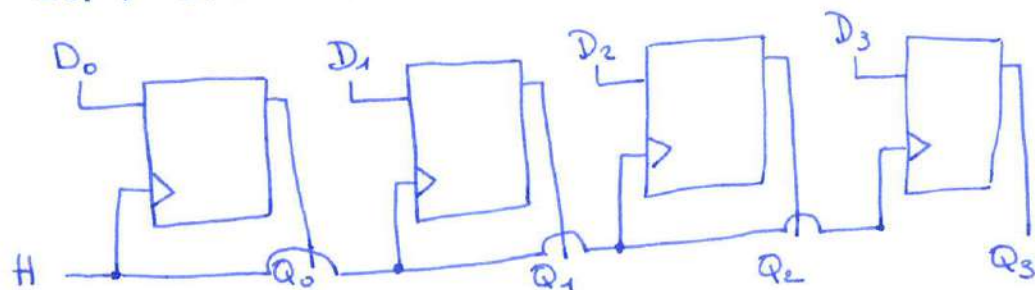
⇒ Rotation



## III / Différents types de registre

### 3.1) Entrées //, sorties //

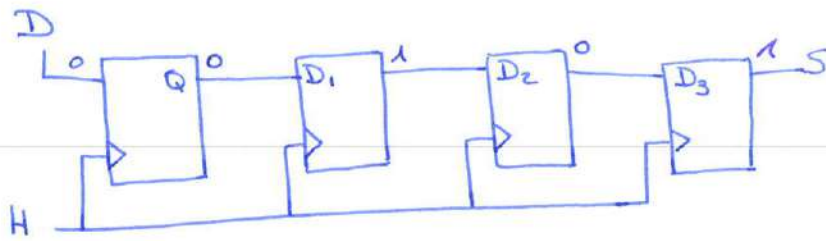
ex: 4 éléments binaires





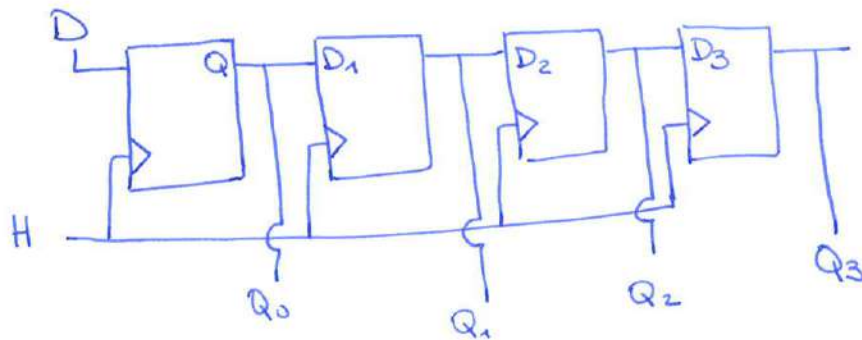
Agis en mémoire tant que P'horloge n'est pas active puis sur front montant copie P'entrée.

### 3.2) Entrée série, sortie série

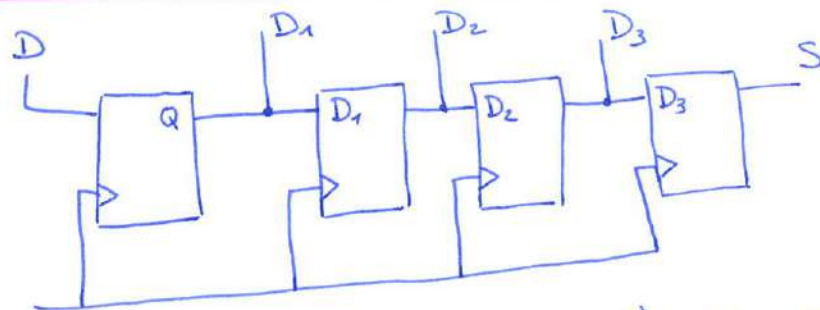


Ici le point faible est à gauche donc 1010.  
On obtient 0100 donc un décalage vers la gauche.

### 3.3) Entrée série, sorties //



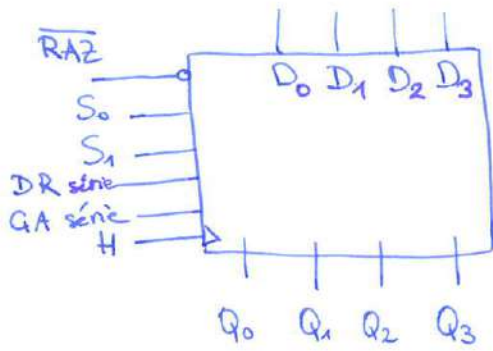
### 3.4) Entrées //, sortie série



Ce registre fonctionnera soit en entrées //, soit en décalage.

## IV / Registre universel P

C'est un registre qui va regrouper les  $\neq$  modes de fonctionnement précédent : chargement et lecture //, chargement et lecture série, décalage à droite et à gauche. (ex TD : 74 HC 194)

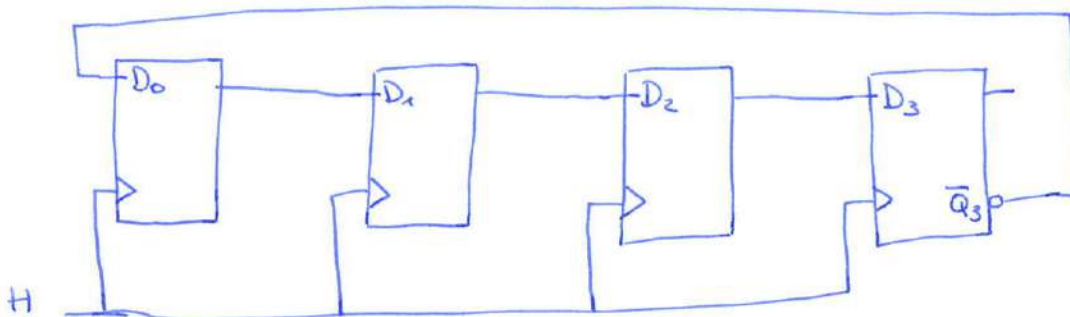


S <sub>0</sub>	S <sub>1</sub>	
0	0	stockage des données mémoire
0	1	décalage à droite
1	0	décalage à gauche
1	1	chargement

## IV/ Compteur à registre à décalage

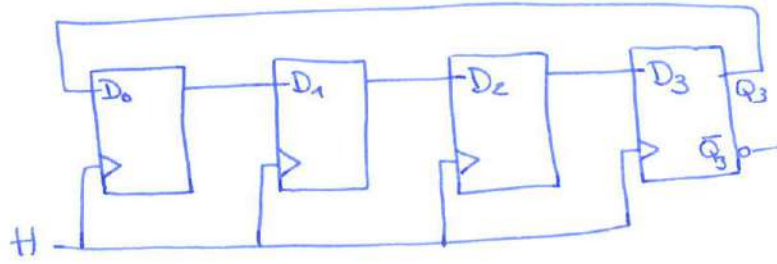
Certains registres sont classés comme compteurs car ils produisent des séquences d'états spécifiques.

### 5.1) Compteur de Johnson



H ↑	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

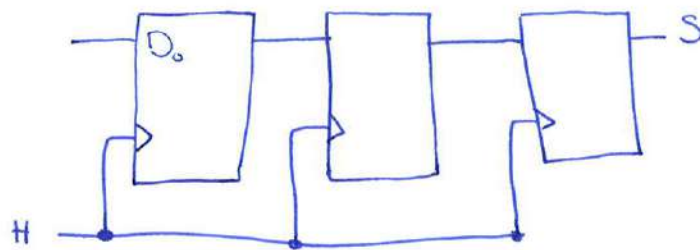
## 5.2) Compteur en anneau



H	$\downarrow$	$Q_0$	$Q_1$	$Q_2$	$Q_3$
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 / Application de registre à décalage

Un registre à entrée série et sortie série peut être utilisé pour créer un retard numérique entre l'entrée et la sortie.



Registre à 8 éléments binaires :  $f = 1 \text{ MHz} \Rightarrow p = 1 \mu\text{s}$   
↳ retard de  $8 \mu\text{s}$ .  
Si 3 bits  $\Rightarrow 3 \mu\text{s}$

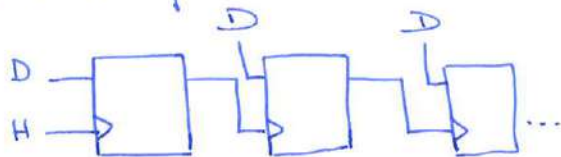


## Chapitre III : Les compteurs

Le nb de bascules utilisées et la façon de les interconnecter détermine le nb d'état du compteur. L'état est défini par le nb binaire formé avec l'ensemble des sorties des bascules.

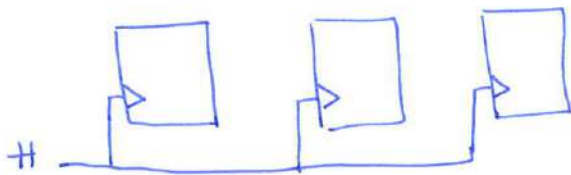
Ils sont classés en deux catégories :

\* les compteurs asynchrones ou compteurs à propagation série  
⇒ La 1<sup>o</sup> bascule est connectée sur l'horloge externe et les autres seront synchronisées sur les bascules précédentes.



\* les compteurs synchrones ou parallèles

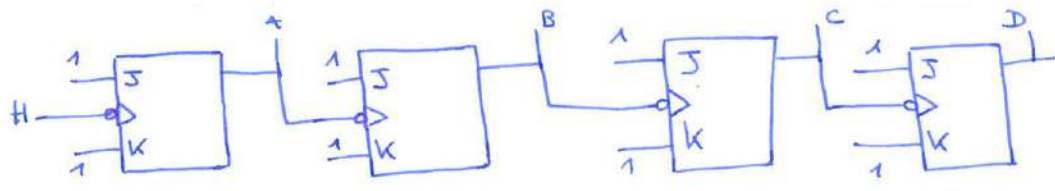
⇒ Le signal d'horloge est connecté à toutes les bascules et permet de les déclencher toutes en même temps



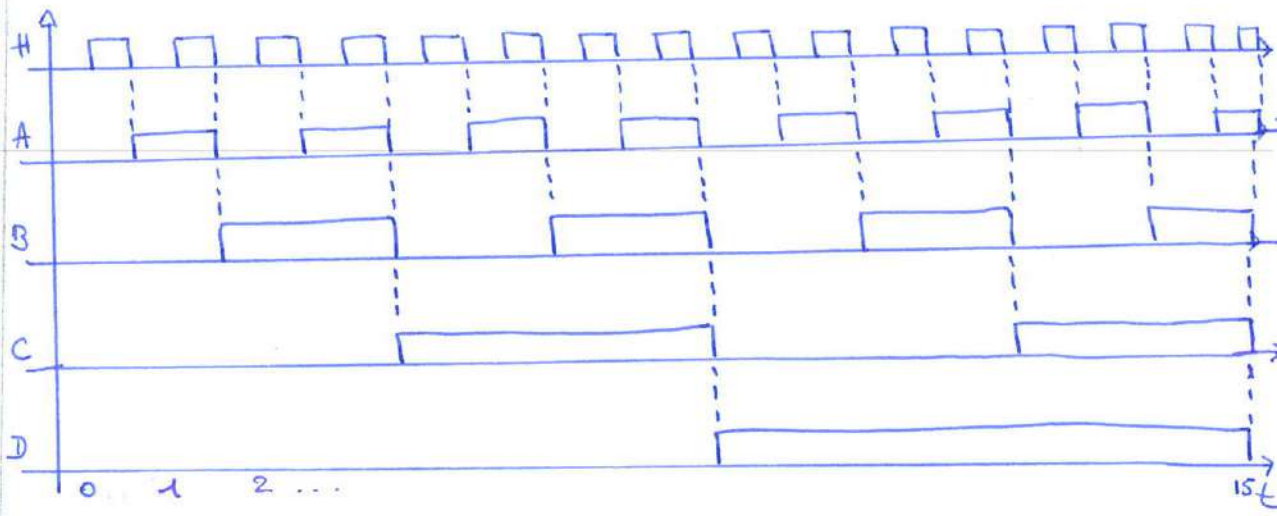
Pour chacune de ces catégories, les compteurs sont classés selon le type de leurs séquences, le nb d'états et le nb de bascules qu'ils comportent.

### I / Compteurs asynchrones

#### 1.1) Compteur binaire



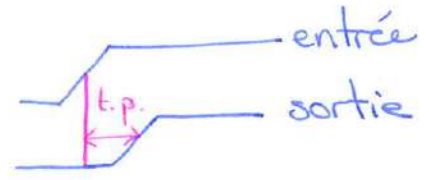
(DCBA)<sub>2</sub>



C'est un compteur modulo 16.

retard de propagation

Il existe deux types de retard de propagation de temps de propagation c'est le temps entre les 50% de l'entrée et les 50% de la sortie.



ex du dessus: Il y aura un retard entre A et B, B et C, C et D. A chaque fois, il faut rajouter le tps de propagation.

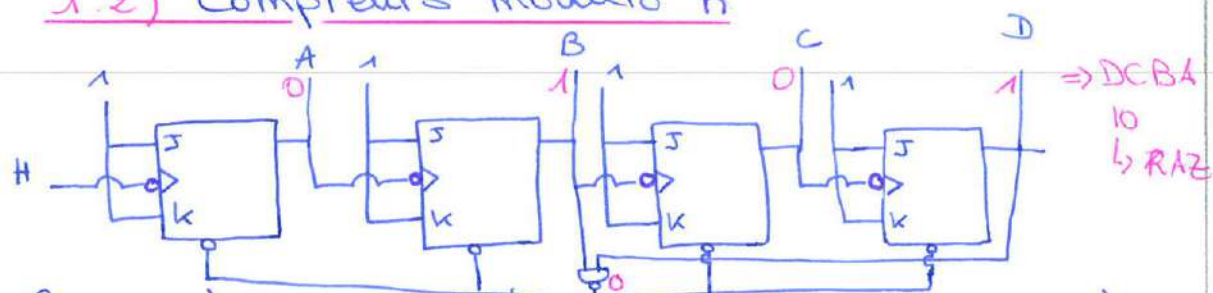
On peut voir apparaître des états transitoires. Ce retard cumulé est un désavantage majeur car il limite la fréquence de synchronisation de l'horloge et crée des problèmes de décodage. Le retard cumulé maximal doit toujours être inférieur à la période du signal d'horloge.



ex tps de retards de propagat° de 10 ns  
 tps de propagat° totale = 40 ns

$$f_{max} = \frac{1}{40 \cdot 10^{-9}} = 25 \text{ MHz}$$

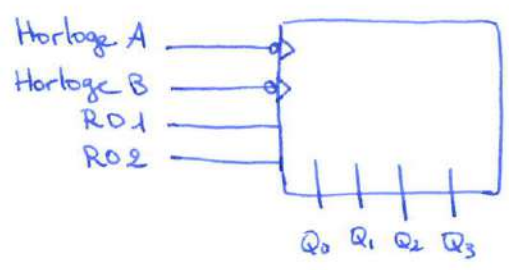
1.2) Compteurs modulo n



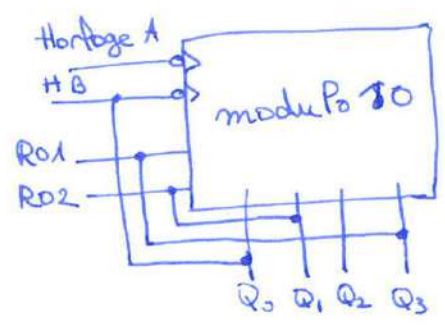
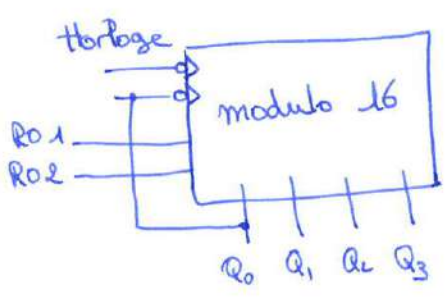
On veut un compteur modulo 10 (de 0 à 9)  
 Il faut donc le remettre à 0. Il faut le faire sur le 10 car si on le fait sur le 9, il n'apparaîtra que quelques nanosec. et on ne le verra pas.

1.3) Les circuits intégrés

exemple 74LS93



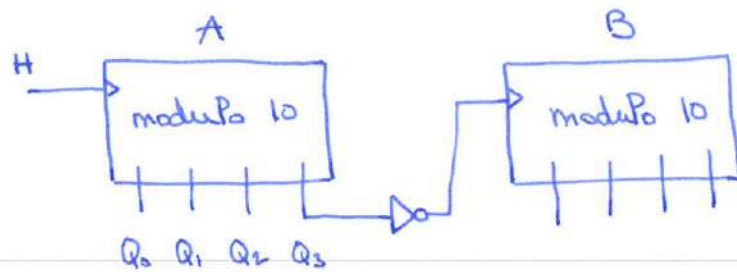
R01 et R02 remise à 0  
 $R01 = R02 = 1$   
 Horloge A : modulo 8  
 Horloge B : modulo 16





## exemple d'application

↳ compteur décimale



B  
0000  
⋮  
0001  
⋮

A  
0000  
⋮  
1001  
0000  
⋮

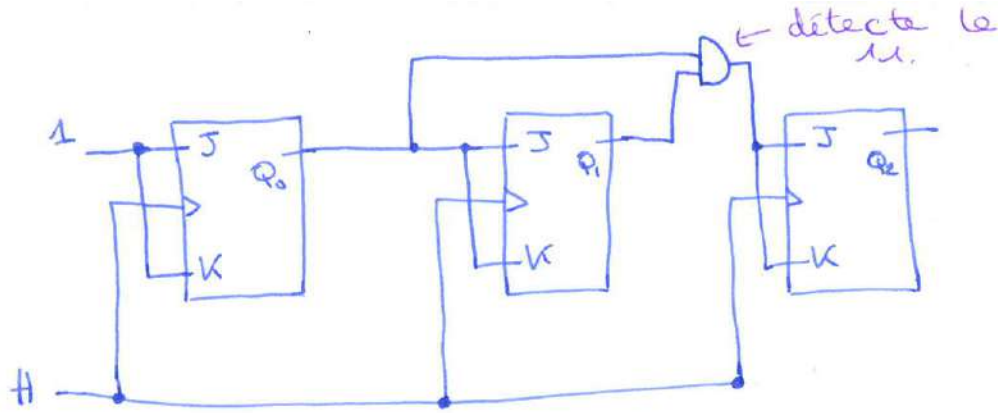
C'est ce qu'on utilise dans les chronomètres avec un compteur modulo 10 et un modulo 6

## II / Compteur synchrone

Les compteurs synchrones permettent d'une part de déterminer les états transitoires et d'autre part de rendre possible l'exécution d'un cycle quelconque. Même horloge pr ttes les bascules

2.1) Exemple d'un compteur binaire synchrone de 3 bits

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	<del>1</del>	0	0
5	1	0	1
6	1	1	0
7	1	1	1



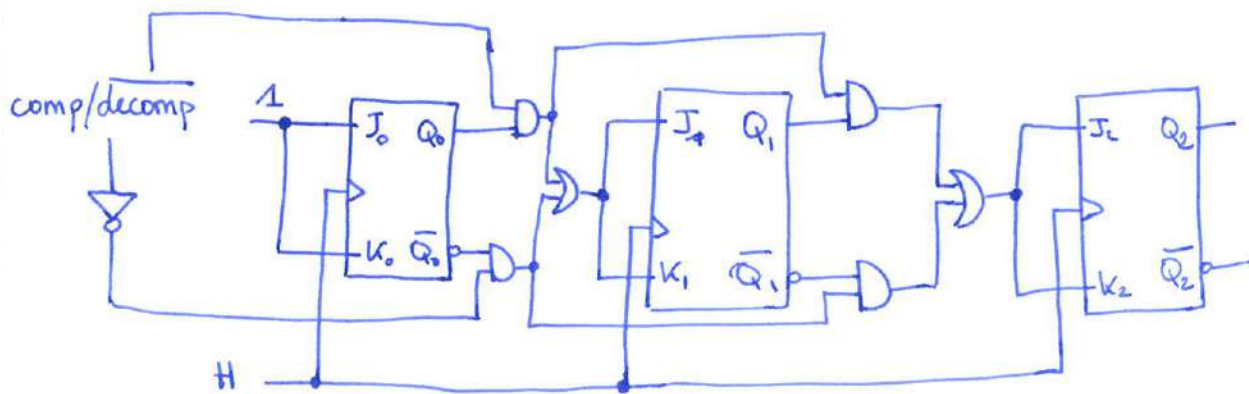
$Q_1$  change d'état à chaque front descendant de  $Q_0$

## 2.2) Compteur réversible

C'est un compteur décompteur ( $0 \rightarrow 7$  puis  $7 \rightarrow 0$ )

En comptage ou en décomptage,  $Q_0$  change à chaque front d'horloge.

comp /  $\overline{\text{décomp}}$

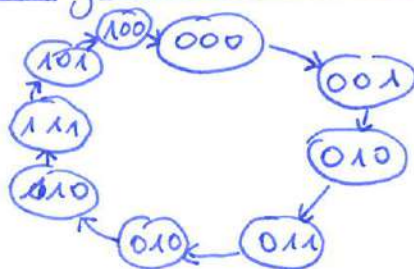


$$Q_1: J_1 = K_1 = Q_0 \text{ comp} + \overline{Q_0} \overline{\text{décomp}}$$

$$Q_2: J_2 = K_2 = Q_0 Q_1 \text{ comp} + \overline{Q_0} \overline{Q_1} \overline{\text{décomp}}$$

## 2.3) Conception de compteur synchrone

### 3.3.1) Diagramme des états



### 2.3.2) Table des états suivants

états présents			états suivants		
$Q_2$	$Q_1$	$Q_0$	$Q_2$	$Q_1$	$Q_0$
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

### 2.3.3) Table de transition des bascules

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

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

*indifférent*

### 2.3.4) Equations Logiques

$Q_2 Q_1$	$J_0$	
	0	1
00	1	$\phi$
01	0	$\phi$
11	1	$\phi$
10	0	$\phi$

$Q_2 Q_1$	$K_0$	
	0	1
00	$\phi$	0
01	$\phi$	1
11	$\phi$	0
10	$\phi$	1

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

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

$J_1$	0	1
	00	0
01	$\phi$	$\phi$
11	$\phi$	$\phi$
10	0	0

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

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

$$K_1 = Q_2 Q_0$$

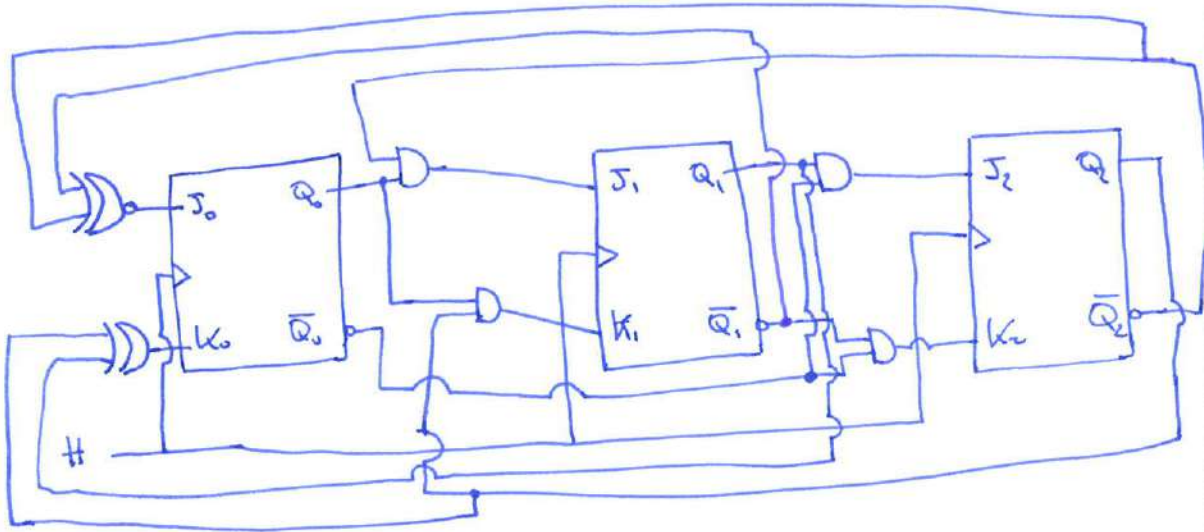


$J_2$	0	1
00	0	0
01	1	0
11	$\phi$	$\phi$
10	$\phi$	$\phi$

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

$$J_2 = Q_1 \bar{Q}_0$$

$$K_2 = \bar{Q}_1 \bar{Q}_0$$



exemple 2 On prend comme contrainte  $J = K$

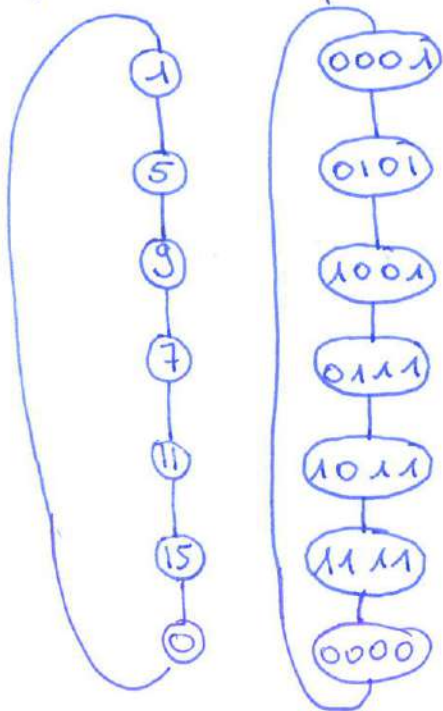


table des états suivants

$Q_3 Q_2$	00	01	11	10
00	0001	0101	$\phi$	$\phi$
01	$\phi$	1001	1011	$\phi$
11	$\phi$	$\phi$	0000	$\phi$
10	$\phi$	0111	1111	$\phi$

$$J_0 = K_0 = \overline{Q_0} + Q_2 Q_3$$

$Q_1 Q_0$ / $Q_3 Q_2$	00	01	10	11
00	1	0	$\phi$	$\phi$
01	$\phi$	0	0	$\phi$
11	$\phi$	$\phi$	1	$\phi$
10	$\phi$	0	0	$\phi$

$$J_1 = K_1 = Q_2 Q_3 + Q_3 \overline{Q_1}$$

$Q_1 Q_0$ / $Q_3 Q_2$	00	01	10	11
00	0	0	$\phi$	$\phi$
01	$\phi$	0	0	$\phi$
11	$\phi$	$\phi$	1	$\phi$
10	$\phi$	1	0	$\phi$

$$J_2 = K_2 = Q_0$$

$Q_1 Q_0$ / $Q_3 Q_2$	00	01	11	10
00	0	1	$\phi$	$\phi$
01	$\phi$	1	1	$\phi$
11	$\phi$	$\phi$	1	$\phi$
10	$\phi$	1	1	$\phi$

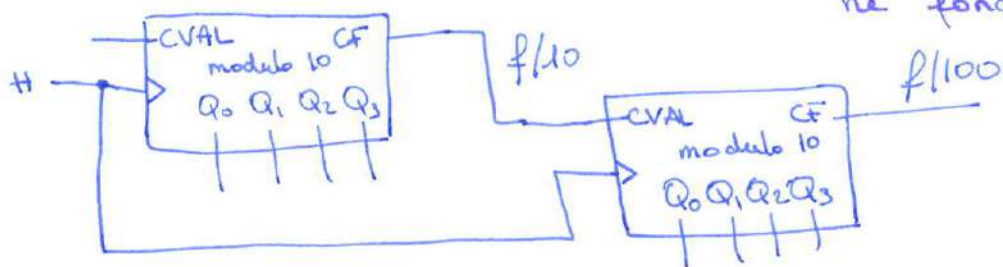
$$J_3 = K_3 = Q_2 + Q_3 \overline{Q_1}$$

$Q_1 Q_0$ / $Q_3 Q_2$	00	01	11	10
00	0	0	$\phi$	$\phi$
01	$\phi$	1	1	$\phi$
11	$\phi$	$\phi$	1	$\phi$
10	$\phi$	1	0	$\phi$

### III / Compteurs montés en cascade

CVAL = 1 le compteur fonctionne (entrée de validation)

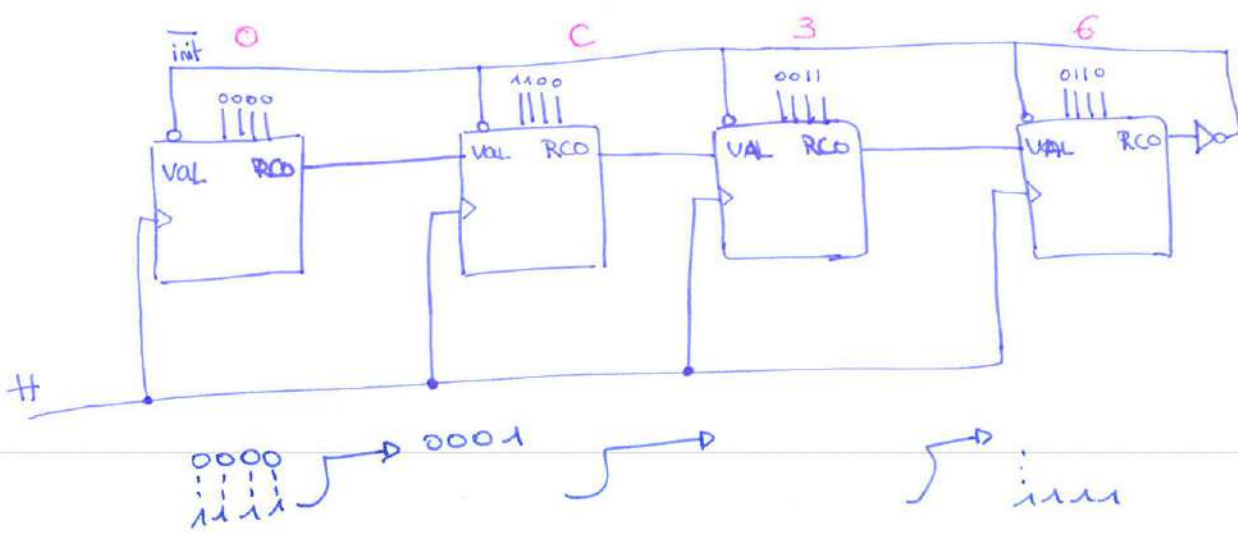
CVAL = 0 le compteur ne fonctionne pas



exemple 2

On conçoit un compteur avec une séquence tronquée

$$\begin{array}{r}
 2^{16} \text{ sorties (4 compteurs)} = 65\ 536 \\
 \text{modulo } 40\ 000 \quad \underline{-40\ 000} \\
 \hline
 25\ 536 = 63\text{C0}]_{16}
 \end{array}$$



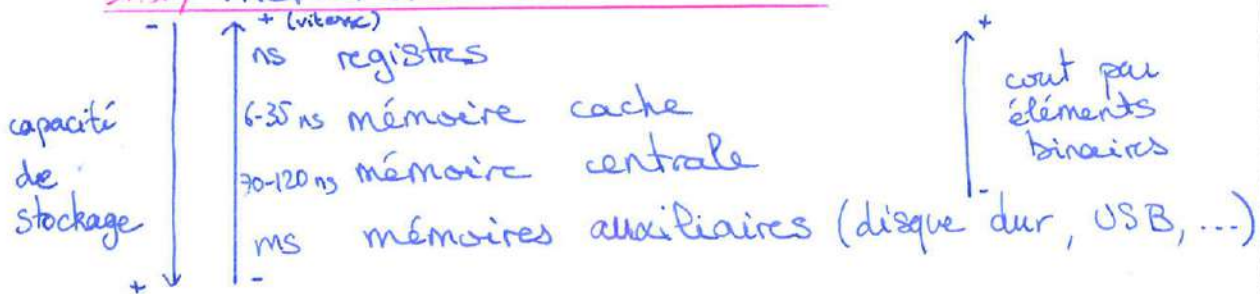
## Chapitre IV : des mémoires

### I / Généralités et définitions

Un ordinateur a deux caractéristiques essentielles :

- vitesse d'exécution
- capacité de mémoire

#### 1.1) Hiérarchie des mémoires



#### 1.2) Organisation des informations

bits      octets (multiple de 8 bits)  
mots (multiple d'octets)

↳ adresse

emplacement d'une unité de donnée

↳ capacité mémoire

correspond au nb d'info que peut contenir la mémoire

kilo octets : 1024 octets =  $2^{10}$  octets  $K_0$

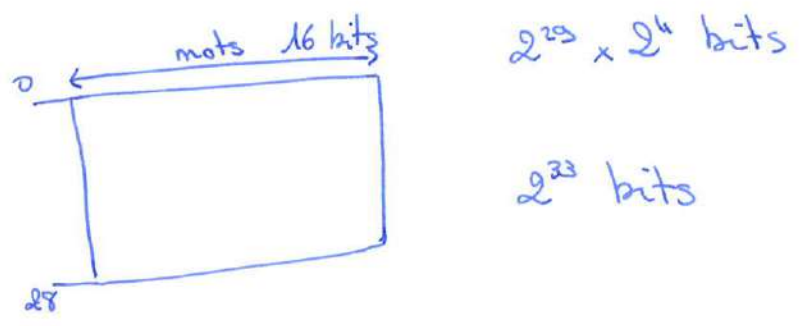


méga octets :  $2^{20}$  octets Mo  
 giga octets :  $2^{30}$  octets Go  
 téra octets :  $2^{40}$  octets To

512 Mega mots de 16 bits

$2^9 \times 2^{20}$  mots de 16 bits

$2^{29}$  mots de 16 bits



↳ temps d'accès

tps qui s'écoule entre le lancement d'un opérat° d'accès (lecture ou écriture) et son accomplissement

mémoires RAM, ROM, registres ns  
 et fréquence =  $\frac{1}{\text{tps d'accès}}$

↳ bande passante de la mémoire

$$B_p = n \times f$$

↑  
nb bit de données

↳ volatilité

caractérise la permanence d'une info de la mémoire.

Une mémoire volatile a besoin d'un apport extérieur pr conserver la mémoire.

## ↳ différents types d'accès mémoire

accès séquentiel  $\Rightarrow$  le  $\oplus$  (ont)  
(bande magnétique)

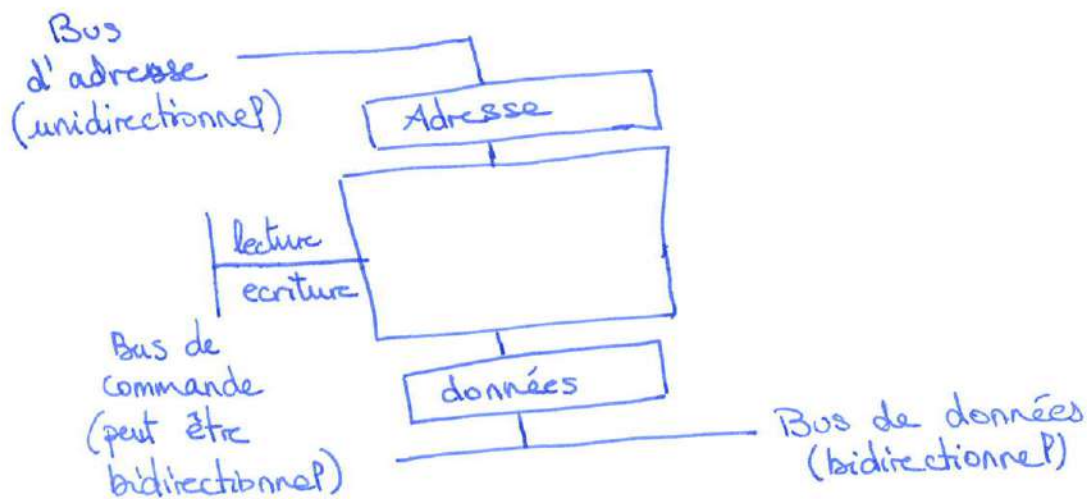
accès direct  $\Rightarrow$  par @

accès semi-séquentiel  $\Rightarrow$  combinaison  
des deux accès précédents  
(disque magnétiques)

accès par contenu  $\Rightarrow$  mémoire  
cache

identifié par clé

## II / Mémoire centrale : mémoire à semi-conducteurs



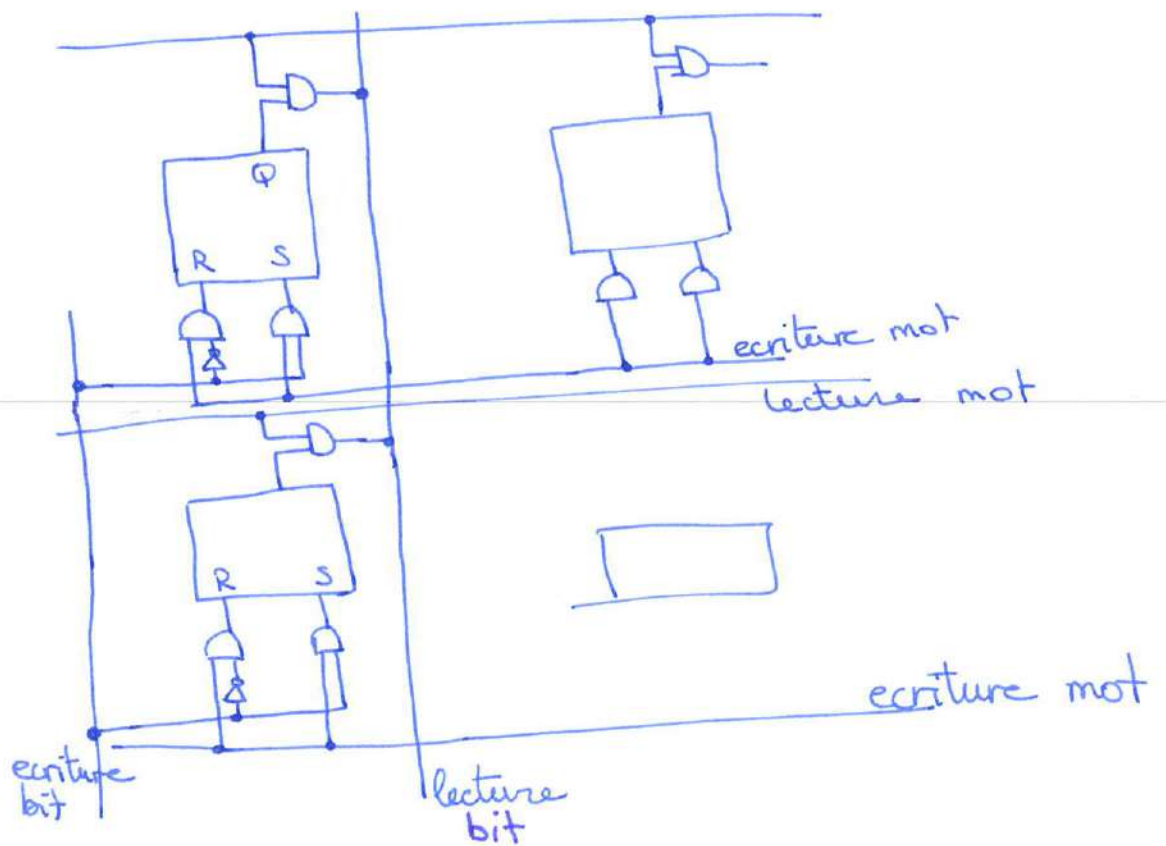
mémoire vive = RAM  
Random Access Memory

mémoire morte = ROM  
Read Only Memory

## III / des mémoires vives (RAM)

RAM statique et RAM dynamique  
(cache) (centrale)

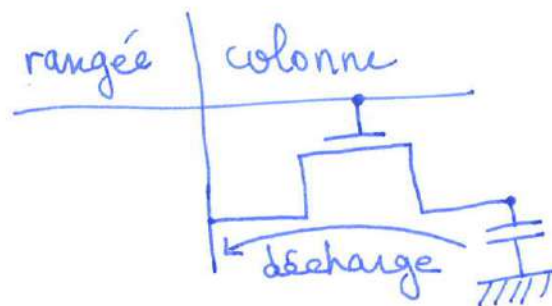
### 3.1) SRAM



### 3.2) Mémoire vive dynamique DRAM

stocke données ds condensateur  
 avantage : simplicité, permet de construire  
 de gd tableau à cout faible

inconvenient : on doit rafraîchir les données  
 régulièrement



\*rangée niveau haut  
 transistor conduit  
 condensateur décharge  
 \*rangée 0 condensateur  
 garde la valeur  
 (interrupteur ouvert)

### IV/ Mémoires mortes

↳ ROM : mémoire programmée par le fournisseur

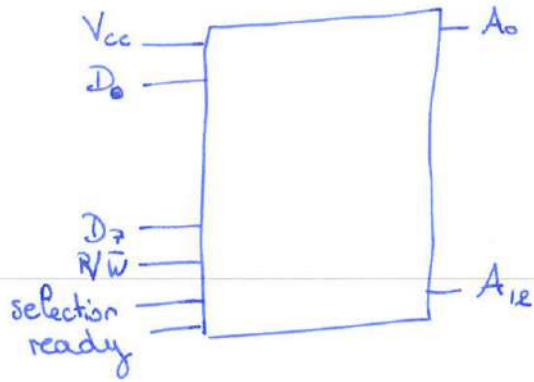
↳ PROM : programmable qu'une seule fois  
 par l'utilisateur

↳ EPROM : mémoire programmable par l'utilisateur



exemple

# Brochage et fonctionnement

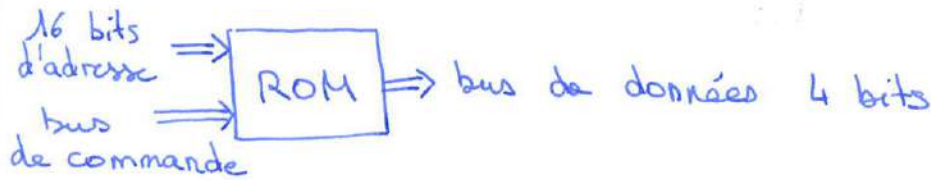


$2^{13}$  addresses et  
8 bits de données

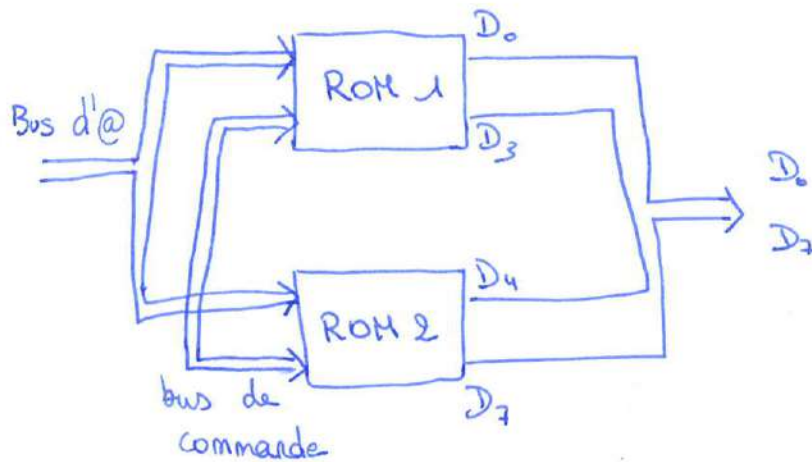
$$2^{13} \times 2^3 = 2^{16} \text{ bits}$$

## V / Extension de la mémoire

### 5.1) Extension de la longueur du mot

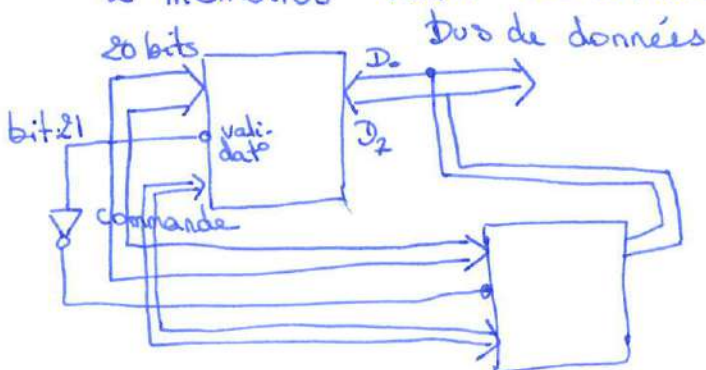


En cherhe à obtenir une longueur de 8 bits



### 5.2) Extension de la capacité de mots

2 mémoires RAM de  $1M \times 8$  a obtenir  $2M \times 8$



## Chapitre V: Introduction aux micro-contrôleurs

Utilisé dans l'électroménager, ds l'automobile  
informatique industrielle  $\Rightarrow$  milieu entre informatique  
et électronique.

d'architecture interne de ce micro-contrôleur (68HC11)

↳ Unité centrale

↳ Mémoire

- RAM

- ROM

- EEPROM

↳ Compteur/timer de 16 bits

↳ 38 entrées/sorties logiques

↳ Une liaison série asynchrone SCI

↳ Une liaison série synchrone SPI

↳ Un convertisseur analogique/numérique

### II/Modèle de programmation

#### 1.1) Registres de l'unité centrale

↳ 2 accumulateurs A et B de 8 bits  
construct<sup>os</sup> logiques et arithmétiques

ou ↳ 1 accumulateur  $D = A + B$   
↳ 16 bits

exemple    A B A    Accum A + Accum B  $\Rightarrow$  Accum A

↳ 2 registres d'index X et Y de 16 bits  
utilisés comme pointeur

"    en adressage indexé

↳ Compteur ordinal : PC  
(compteur qui pointe sur chaque instruct<sup>o</sup>)

↳ Pointeur de pile : SP  
(pointe sur la 1<sup>o</sup> @ libre de la pile)

Remarque: - interruption matérielle ou logicielle

interruption matérielle  $\Rightarrow$  vient d'un périphérique qui demande la main à l'ordi  
l'ordi va contrôler si interruption  
⊕ prioritaire  
si non traite interruption puis revient au programme initial.

timer programmer en interruption

2 broches externes d'interruption

### 1.2) des signaux disponibles

↳ les ports et autres sorties (7)

→ ports unidirectionnels

Port B en sortie

Port E en entrée

→ ports bidirectionnels

programmation à l'aide d'un

registre DDR<sub>L</sub>

7	6	5	4	3	2	1	0
0	1	0	0	0	0	1	0

8 bits

DDR<sub>D</sub> \$42

PD6 = PD1 = 1

// hexa-décimale

à nous de programmer si  
Sortie ou entrée