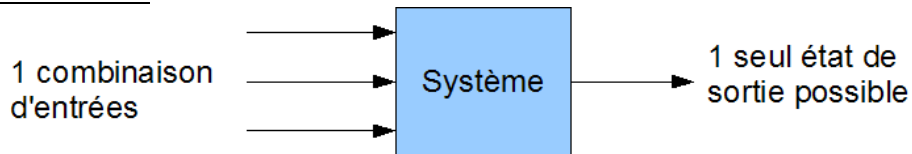


## Logique séquentielle

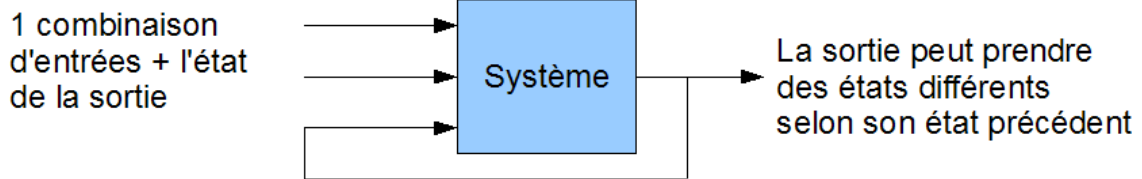
### Fonction Mémoire

### 1 – Introduction

Logique combinatoire :



Logique séquentielle :



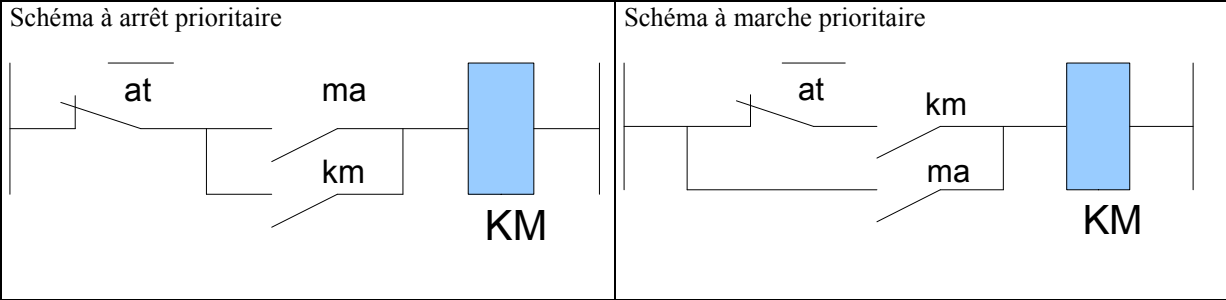
Exemple : Télérupteur, commande TOR de machine outil, GRAFCET ...

### 2 – Fonction mémoire électromécanique

Exemple : Commande d'un moteur par un contacteur.

ma : Bouton poussoir d'ordre de marche  
 at : Bouton poussoir d'arrêt  
 km : Contact associé au contacteur  
 KM : Contacteur

at	km	ma	KM	
0	0	0	0	Arrêt
0	0	1	1	Mise en mémoire
0	1	0	1	Mémoire
0	1	1	1	Marche
1	0	0	0	Mise à zéro
1	0	1	0 / 1	L'état de la priorité est à définir.
1	1	0	0	
1	1	1	0 / 1	



L'effet mémoire peut être réalisé par deux moyens :

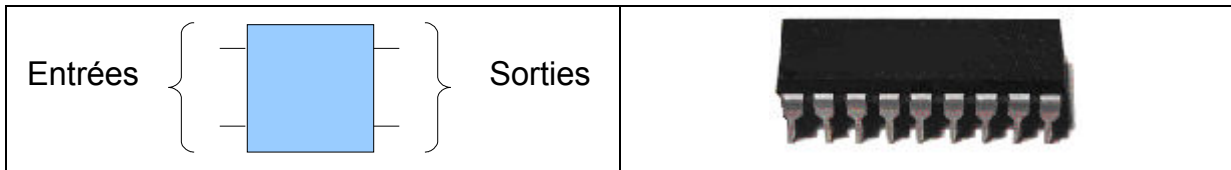
- ⇒ Par maintien physique de l'état (came)
- ⇒ Par boucle interne (mise en parallèle de contacts)

### 3 – Fonction mémoire par bascules

#### 3 - 1 Vocabulaire

$Q_n$  : sortie à l'état n  
 $Q_{n-1}$  : sortie à l'état précédent  
 Bascules : système électronique de mémorisation

#### 3 – 2 Représentation



#### 3 – 3 Horloges

Certaines bascules sont synchronisées avec des horloges. Une horloge est un signal périodique carré précis produit par des circuits RC ou quartz

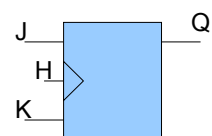
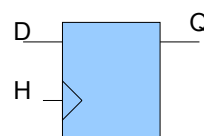
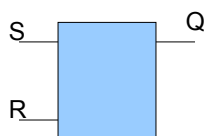
		Synchronisation sur le front montant de l'horloge
		Synchronisation sur le front descendant de l'horloge
		Synchronisation sur le niveau bas de l'horloge
		Synchronisation sur le niveau haut de l'horloge

#### 3 – 4 Principe

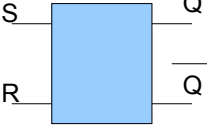
Des entrées mettent à 1 (Set) ou à 0 (Reset) des sorties. Il faut en plus tenir compte de l'état précédent de la sortie. Les sorties sont électroniquement mémorisées et conservées jusqu'à un autre ordre.

Il existe principalement les bascules :

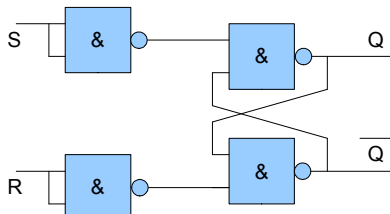
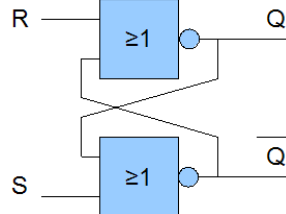
- RS
- D
- JK




### 4 - Bascule RS

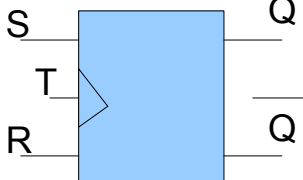
	<p>S : SET mise à 1 de la sortie                  R : RESET mise à zéro de la sortie                  Q : Sortie  <math>\bar{Q}</math> : Sortie complémentée</p>	<p>Equation de la fonction mémoire</p> $Q = S + \bar{R} \cdot Q$
---	--	--

On peut réaliser l'équation précédente avec des portes NAND ou des NOR.

<p>Schéma en NAND</p> 	<p>Schéma en NOR</p> 
---	--

<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>R</th> <th>S</th> <th>Q</th> <th>état initial : Qn = 0</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Etat mémoire</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Mise à 1</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Etat mémoire</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Mise à 0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0/1</td> <td>indéterminé</td> </tr> </tbody> </table>	R	S	Q	état initial : Qn = 0	0	0	0	Etat mémoire	0	1	1	Mise à 1	0	0	1	Etat mémoire	1	0	0	Mise à 0	1	1	0/1	indéterminé		<p><u>Inconvénient de la bascule RS</u></p> <p>Lorsque R = S = 1 , Q et Q se retrouvent dans un état indéterminé. On a donc construit d'autres types de bascules pour palier à cela.</p>
R	S	Q	état initial : Qn = 0																							
0	0	0	Etat mémoire																							
0	1	1	Mise à 1																							
0	0	1	Etat mémoire																							
1	0	0	Mise à 0																							
1	1	0/1	indéterminé																							

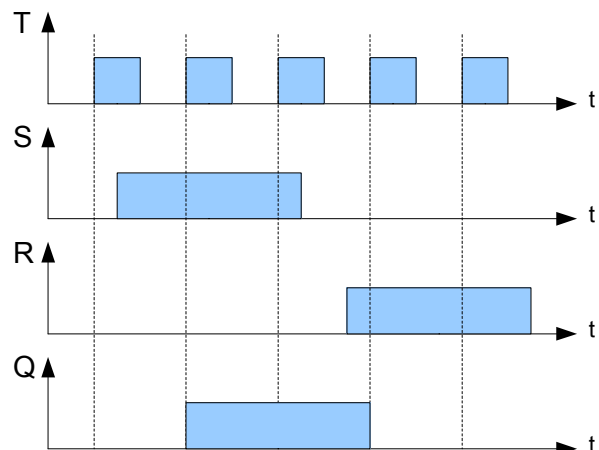
**5 – Bascule RST synchrone** : On synchronise l'évolution de la sortie sur une horloge.

	<p>T = X ⇒ pas d'évolution de la sortie, mémorisation</p> <p>T = ↑ ⇒ Evolution de la sortie en fonction des entrées</p>
---	---

Ici l'horloge H s'appelle T, on pourra trouver CLK (clock)

A chaque front montant, on regarde les niveaux des entrées et on fait évoluer la sortie.

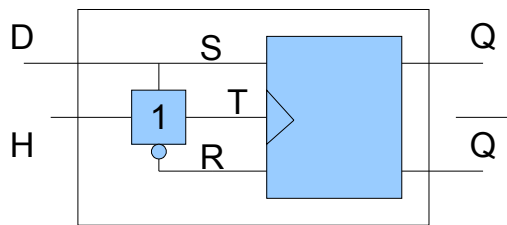
T	R	S	Q
0	X	X	μ
↑	0	0	0
↑	0	1	1
↑	1	0	0
↑	1	1	0/1



### 6 - Bascule D synchrone

La bascule D permet de mémoriser un ou des bits en attendant leur traitement.

Il s'agit d'une bascule RS dont on a modifié le comportement en ajoutant une fonction NON entre S et R.

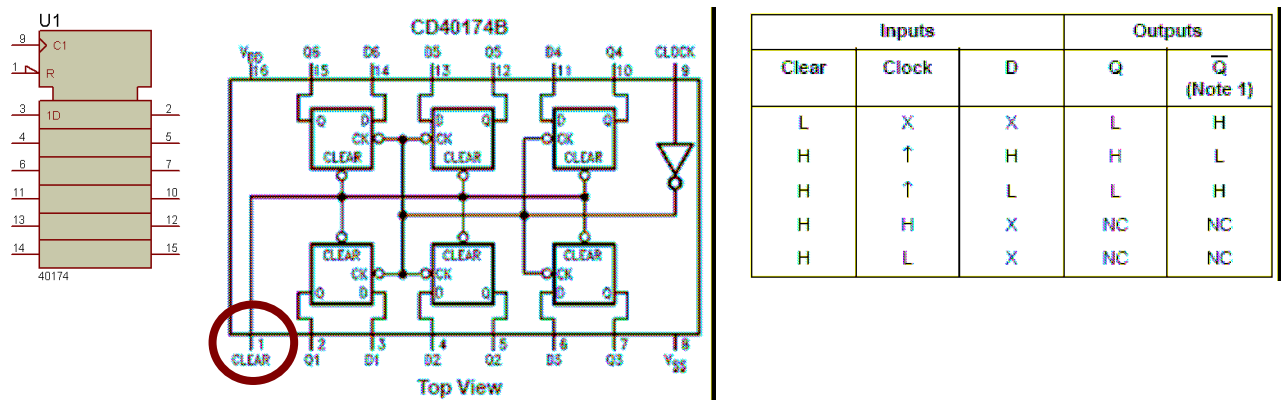


H	D	Q
X	0/1	$\mu$
↑	1	1
↑	0	0

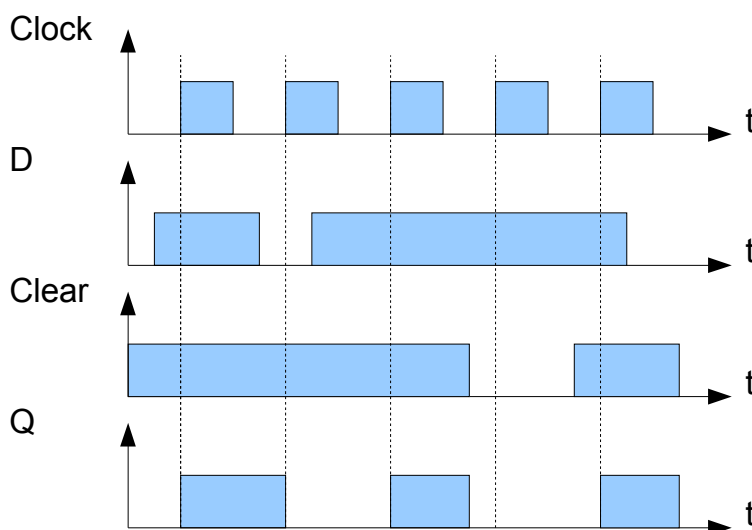
Les bascules possèdent 1 ou 2 entrées supplémentaires pour permettre le forçage à 0 ou à 1.

- Une action sur PRE force à 1 la sortie
- Une action sur CLR force à 0 la sortie

#### Exemple 40174



Ici H s'appelle CLOCK et la fonction CLEAR est active au niveau bas quelles que soient les entrées.



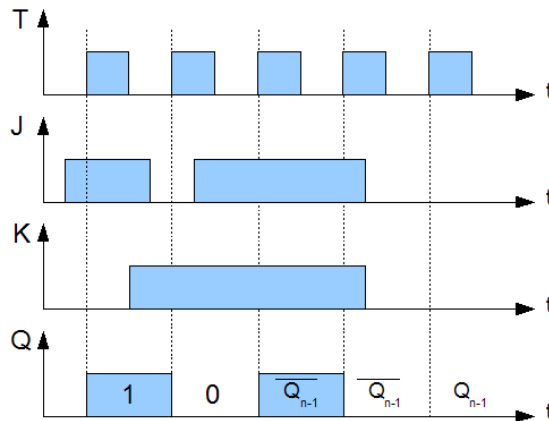
### 7 – Bascule JK

Cette bascule permet de prendre en compte les états indéterminés de la bascule RS.

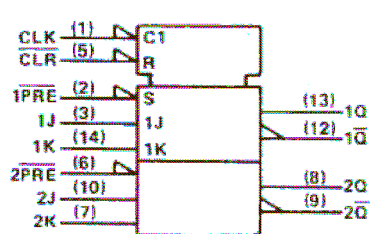
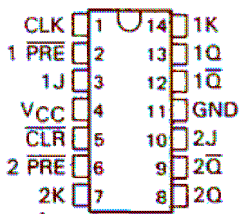
⇒ lorsque les deux entrées sont à 1, la sortie bascule (toggle) vers le complément de l'état précédent.

$$J = K = 1 \Rightarrow Q_n = \overline{Q_{n-1}}$$

	H	J	K	$Q_n$
	0	X	X	$Q_{n-1}$
	↑	0	0	$Q_{n-1}$
	↑	0	1	0
	↑	1	0	1
↑	1	1	1	$\overline{Q_{n-1}}$



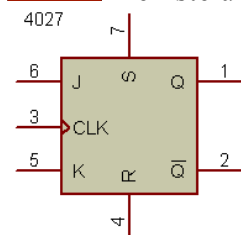
Exemple 74LS78a



INPUTS					OUTPUTS	
PRE	CLR	CLK	J	K	Q	$\overline{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H <sup>†</sup>	H <sup>†</sup>
H	H	↓	L	L	$Q_0$	$\overline{Q}_0$
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	TOGGLE	
H	H	H	X	X	$Q_0$	$\overline{Q}_0$



Il existe une bascule proche d'aspect MASTER SLAVE FLIP FLOP,



ici si une entrée est à 1, on ignore l'action sur la seconde.

Inputs $t_{n-1}$ (Note 1)						Outputs $t_n$ (Note 2)	
CL (Note 3)	J	K	S	R	Q	Q	$\overline{Q}$
✓	I	X	0	0	0	I	O
✓	X	0	0	0	I	I	O
✓	0	X	0	0	0	O	I
✓	X	I	0	0	I	O	I
✓	X	X	0	0	X	(No Change)	
X	X	X	I	0	X	I	O
X	X	X	0	I	X	O	I
X	X	X	I	I	X	I	I

## Fonction comptage : Compteurs asynchrone et synchrone

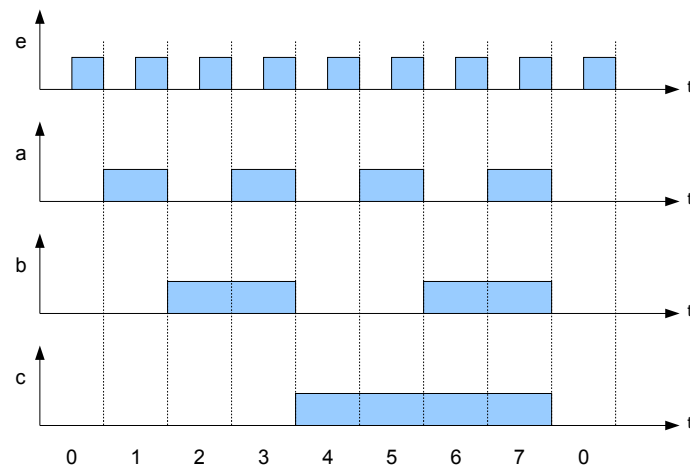
### 1 – Mise en situation

mise en paquet de 6 œufs après le marquage

### 2 – Analyse d'un compteur 3 bits

Complétez le chronogramme de a, b et c d'après la table de vérité :

c	b	a	Valeur décimale
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

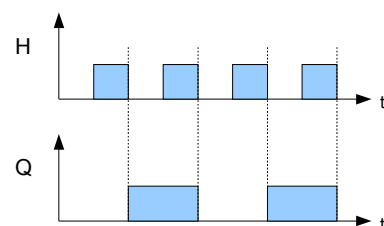
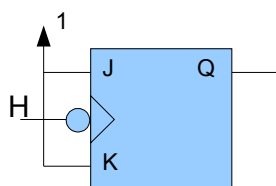


### 3 – Observation de la table de vérité

- Que pouvons nous dire de la fréquence du signal a par rapport ?  
⇒ La période est double , la fréquence a été divisée par deux.
- Que pouvons nous dire de la fréquence du signal b par rapport a ?  
⇒ La période est double , la fréquence a été divisée par deux.
- A quel instant par rapport au signal a le signal b change-t-il ?  
⇒ Sur le front descendant
- De même pour le signal c par rapport au signal b ?  
⇒ Sur le front descendant
- Quelle est la valeur binaire (0 ou 1) de a et de b quand c passe à 1 ?  
⇒ a et b sont à 1.

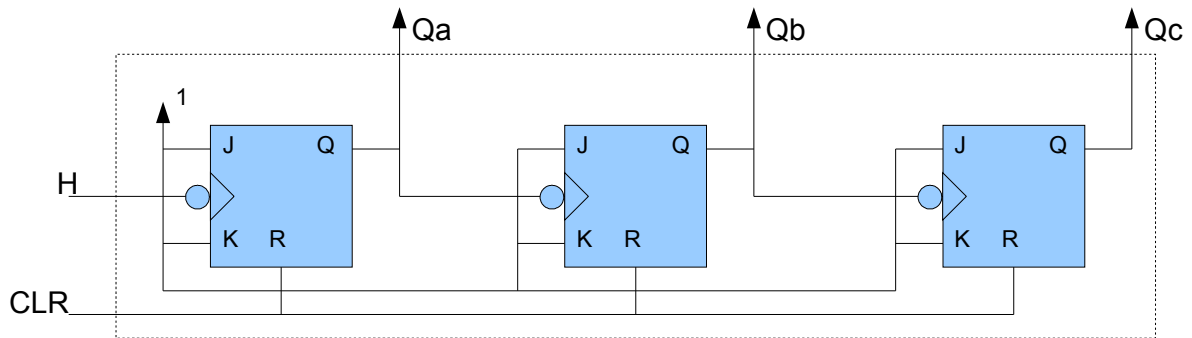
### 4 – Diviseur de fréquence

On branche en permanence J et K à 1, la sortie passera au complément de l'état précédent à chaque front ici descendant.



## 5 – Compteur asynchrone

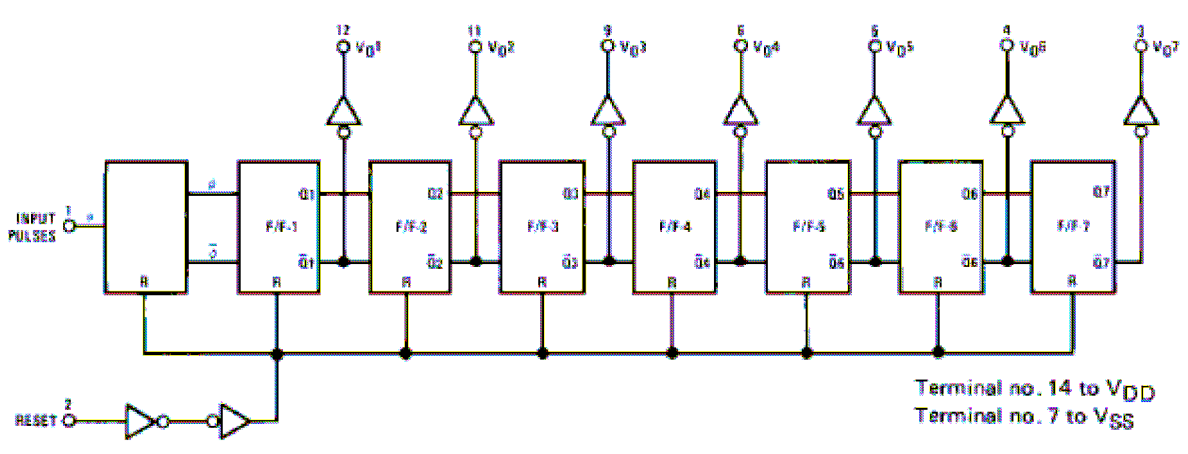
Le compteur asynchrone est constitué de **diviseurs de fréquence** branchés en cascade.



Inconvénient : Les temps de commutation s'additionnent.

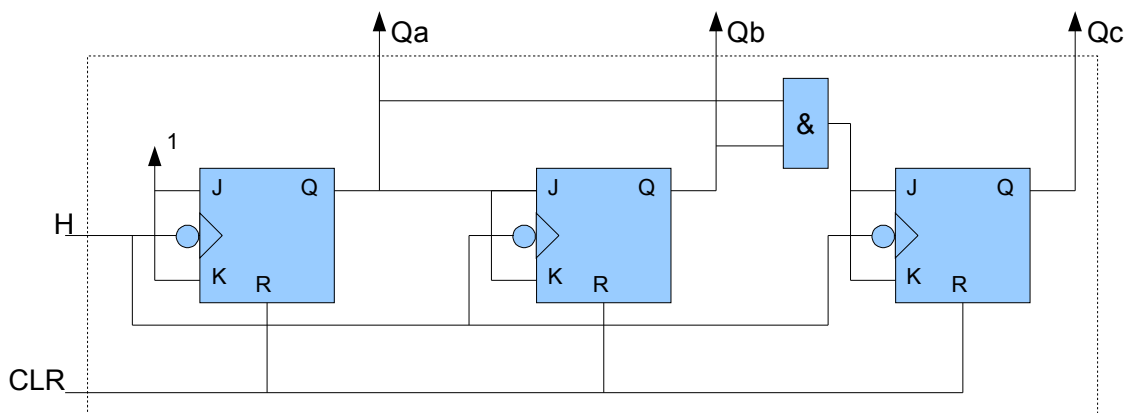
Avantage : Simplicité

Exemple : CD4024BC 7-Stage Ripple Carry Binary Counter



## 6 – Compteur synchrone

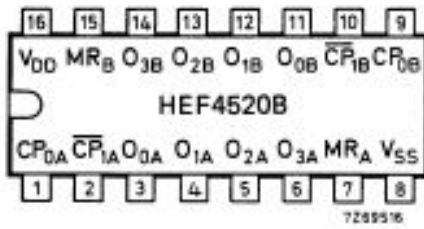
On utilise la particularité du changement d'état de c quand a et b ont simultanément un front descendant.



Inconvénient : Ajout d'une fonction logique ET

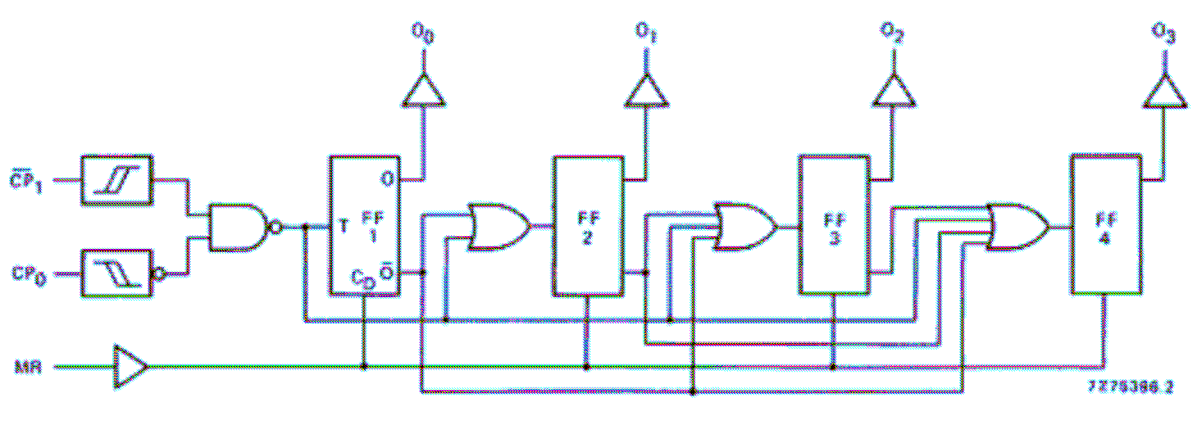
Avantage : Pas d'erreur de commutation

Exemple : HEF4520b : DUAL BINARY COUNTER



CP <sub>0</sub>	CP <sub>1</sub>	MR	MODE
↗	H	L	counter advances
L	↘	L	counter advances
↘	X	L	no change
X	↗	L	no change
↗	L	L	no change
H	↘	L	no change
X	X	H	O <sub>0</sub> to O <sub>3</sub> = LOW

Schéma structurel



**7 – Blocage d'un compteur**

On recherche dans la table de vérité la valeur à laquelle on désire bloquer le compteur. On la transforme en produit pour obtenir une combinaison originale et on l'injecte sur le reset des bascules.

Exemple : on désire compter 6 œufs.

On laisse passer 6 états et on bloque avec la valeur décimale 6.

La combinaison originale est donc 110.

c	b	a	X	
0	0	0	0	1 <sup>er</sup> état
0	0	1	1	2 <sup>ème</sup> état
0	1	0	2	3 <sup>ème</sup> état
0	1	1	3	4 <sup>ème</sup> état
1	0	0	4	5 <sup>ème</sup> état
1	0	1	5	6 <sup>ème</sup> état
1	1	0	6	blocage

