Université Internationale de Casablanca

Partie 2
Automatisme logique
La logique séquentielle

Pr. Khalid BENJELLOUN
bkhalid@emi.ac.ma

Département Electrique
Section Automatique et Informatique Industrielle

Université Mohammed V –Agdal

ECOLE MOHAMMADIA D'INGÉNIEURS

EMI

P L A N

1 : Notion de circuit séquentiel

2 : Bascule de base

3 : Réalisation par relais

4 : Exemples

1 : Introduction

- En logique combinatoire, aussi appelée logique booléenne, l'état des sorties est une fonction logique de l'état des entrées
- En logique séquentielle, l'état des sorties est fonction de l'état des entrées et de l'état du système, ce qui implique qu'une combinaison d'entrées ne génère pas toujours la même sortie.
- La logique combinatoire peut quand même être utilisée pour étudier les automatismes séquentiel simples.

Notion de circuit séquentiel

- Dans un circuit séquentiel, le temps intervient
- Le circuit se comporte en fonction du passé et du présent
- Pour tenir compte du passé, il doit y avoir un élément de mémoire
- L'entrée de la mémoire correspond à l'état futur (« next-state »)
- La sortie de la mémoire correspond à l'état présent (« present-state »)
- Les signaux de l'état futur sortent d'un circuit combinatoire et entrent dans l'élément de mémoire
- Les signaux de l'état présent sortent de l'élément de mémoire et entre dans un circuit combinatoire.
- Ces circuits ont la capacité de mémoriser des informations et par conséquent de traiter des séquences de données.

La logique séquentielle

- Tel que mentionné dans l'exemple du Moteur, le diagramme des phases montre 2 combinaisons de sorties pour une même combinaison de variables d'entrées !
- L'approche par la logique combinatoire ne peut résoudre un tel système séquentiel
- Les diagrammes des phases et des transitions peuvent être utile pour identifier ces situations

La logique séquentielle

- **Exemple:**
  - Le contacteur "C" d'un moteur est commandé par deux boutons
  - Le bouton "m" pour mettre le moteur en marche
  - Le bouton "a" pour arrêter le moteur
  - Le moteur tourne jusqu'à ce que l'opérateur appuis sur "a"

La logique séquentielle

- **Exemple:**
  - Moteur à l'arrêt:
    - entrées a=0, m=0 ; sortie C=0
  - Mise en marche:
    - entrées a=0, m=1 ; sortie C=1
  - Moteur en marche:
    - entrée a=0, m=0 ; sortie C=1
- **Problème ?**
  - Pour a=0, m=0 ; sortie C=0 ou C=1
- **Impossible à résoudre avec l'approche combinatoire**

La logique séquentielle

- **La méthode de Huffman utilise les diagrammes des phases et des transitions pour résoudre ces systèmes.**
  - Dénombrer tous les états possibles;
    - Établir un diagramme des phases
    - Établir un diagramme des transitions
  - Construire la table primitive des états;
  - Construire la table réduite des états;
    - Définir les variables secondaires.
  - Trouver les équations logiques des actionneurs et des variables secondaires.

La méthode de Huffman

- **Dénombrer tous les états possibles;**

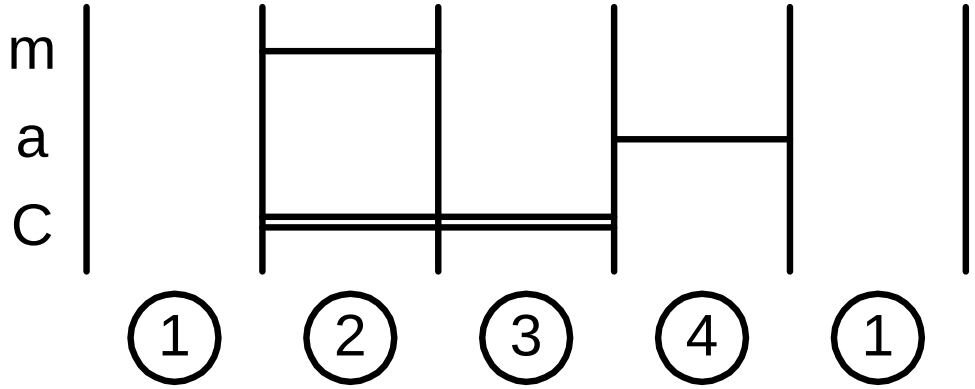


diagramme des phases

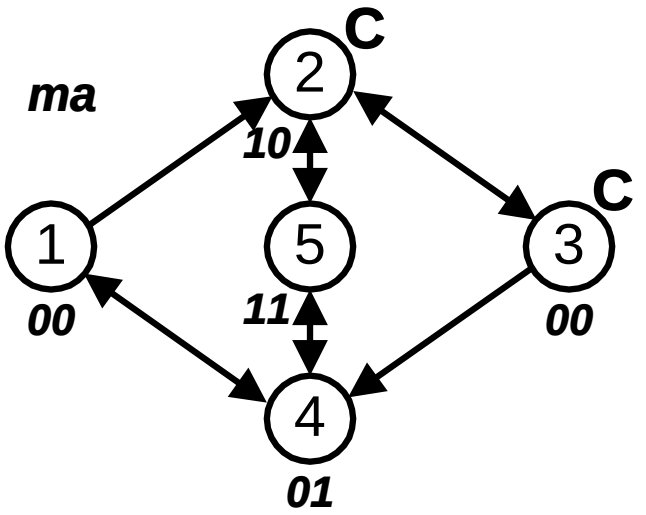


diagramme des transitions

La méthode de Huffman

- **Construire la table primitive des états;**
  - La matrice primitive des états est une transcription du diagramme des transitions.
  - Elle permet de représenter sous forme matricielle l'évolution du système.

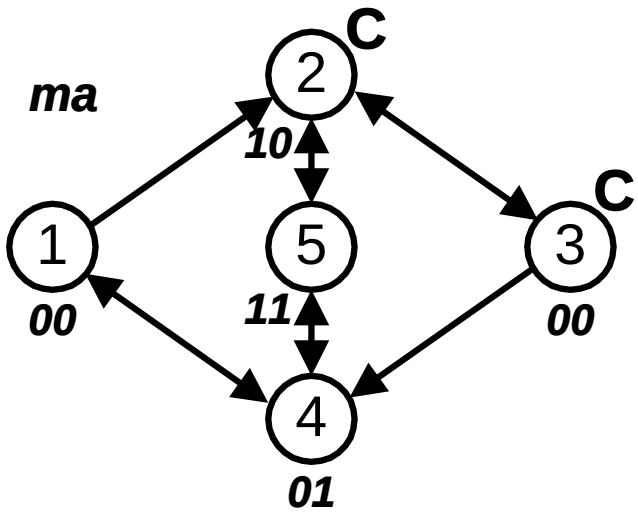


diagramme des transitions

La méthode de Huffman

- **Construire la table primitive des états;**
  - États stables = encerclés

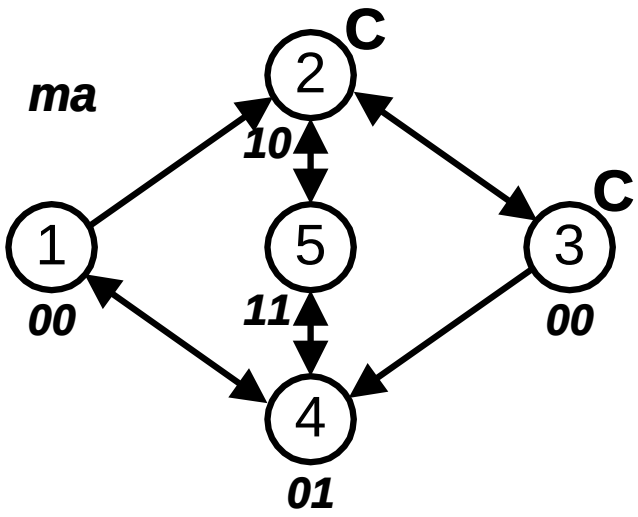


diagramme des transitions

5-arrêt prioritaire

ma \	00	01	11	10	C
	(1)				0
				(2)	1
	(3)				1
		(4)			0
			(5)		0

La méthode de Huffman

- **Construire la table primitive des états;**
  - Encerclés = stables, Non-encerclés = transitoires

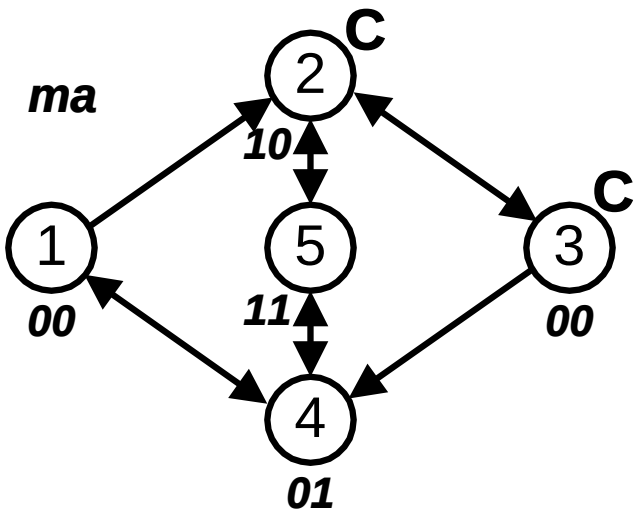


diagramme des transitions

5-arrêt prioritaire

m a				c
(1)	4	X	2	0
3	X	5	(2)	1
(3)	4	X	2	1
1	(4)	5	X	0
X	4	(5)	2	0

La méthode de Huffman

- **Construire la table réduite des états;**
  - Le regroupement de lignes de la matrice primitive doit obéir aux règles suivantes :
    - Les états sur chacune des lignes à regrouper doivent être les mêmes ou correspondre à un X;
    - Les niveaux logiques de la ou des sorties doivent être les mêmes sur les lignes à regrouper.

La méthode de Huffman

- Les états sur chacune des lignes à regrouper doivent être les mêmes ou correspondre à un X;
- Les niveaux logiques de la ou des sorties doivent être les mêmes sur les lignes à regrouper.

m / a				c
(1)	4	X	2	0
3	X	5	(2)	1
(3)	4	X	2	1
1	(4)	5	X	0
X	4	(5)	2	0

m / a				c
(1)	(4)	(5)	2	0
(3)	4	5	(2)	1

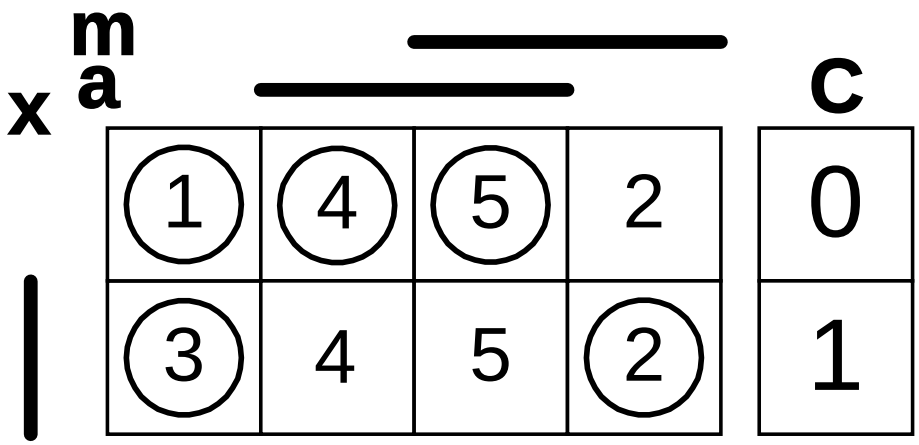
La méthode de Huffman

- **Définir les variables secondaires.**
  - Par exemple, si les seules informations disponibles sont que « m » et « a » sont tous deux à 0, il est impossible de savoir si la machine est dans l'état 1 ou dans l'état 3.
  - Avec une variable secondaire (x) nous saurons sur quelle ligne est l'état de la machine

x \ m a				c
	①	④	⑤	2	0
|	③	4	5	②	1

La méthode de Huffman

- **Définir les variables secondaires.**
  - Un code d'un bit permet de sélectionner une ligne parmi deux (2^1)
  - Un code de 2 bits une ligne parmi quatre (2^2)
  - Un code de 3 bits une ligne parmi huit (2^3).



La méthode de Huffman

- **Trouver les équations logiques - Karnaugh !**
  - Pour une variable secondaire, il faut mettre dans chaque case la valeur de la variable secondaire pour l'état stable correspondant au numéro d'état de la case correspondante de la matrice contractée

variable secondaire x

x \ m a					C
|	(1)	(4)	(5)	2	0
	(3)	4	5	(2)	1

x \ ma	00	01	11	10
0	0	0	0	1
1	1	0	0	1

$$x = m\bar{a} + x\bar{a} = (m + x)\bar{a}$$

La méthode de Huffman

- **Trouver les équations logiques - Karnaugh !**
  - Pour une variable de sortie, il faut mettre dans chaque case la valeur de la variable de sortie pour l'état stable correspondant au numéro d'état de la case correspondante de la matrice contractée

variable de sortie C

x \ ma	00	01	11	10
0	(1)	(4)	(5)	2
1	(3)	4	5	(2)

C
0
1

x \ ma	00	01	11	10
0	0	0	0	1
1	1	0	0	1

$$C = m\bar{a} + x\bar{a} = (m + x)\bar{a}$$

$$C = x$$

La méthode de Huffman

- **Trouver les équations logiques - Karnaugh !**

$$x = m\bar{a} + x\bar{a} = (m + x)\bar{a}$$

$$C = m\bar{a} + x\bar{a} = (m + x)\bar{a}$$

$$C = x$$

Les méthodes intuitives

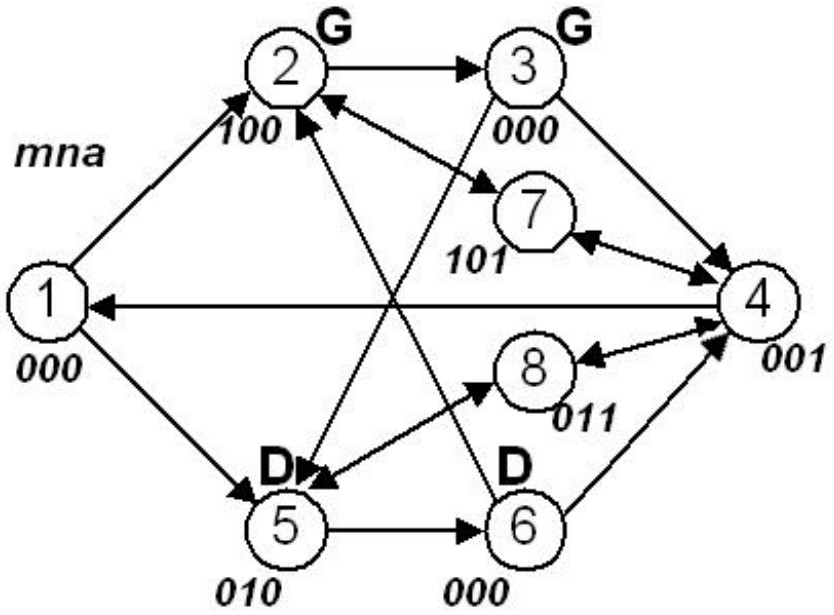
- **Basées sur la méthode de Huffman**
- **Façon de trouver les variables secondaires**

Les méthodes intuitives

- **Les variables secondaires identifiées aux sorties**
  - Pour certains automatismes, on peut simplement faire correspondre les variables secondaires aux sorties
- **Exemple:**
  - Un moteur peut tourner vers la gauche « G » ou vers la droite « D ». Commandé par 3 boutons :
    - « m » et « n » qui sont verrouillés (impossibles à actionner en même temps) et qui correspondent respectivement à une rotation à gauche et une rotation à droite;
    - « a » qui est le bouton d'arrêt (prioritaire)

Les méthodes intuitives

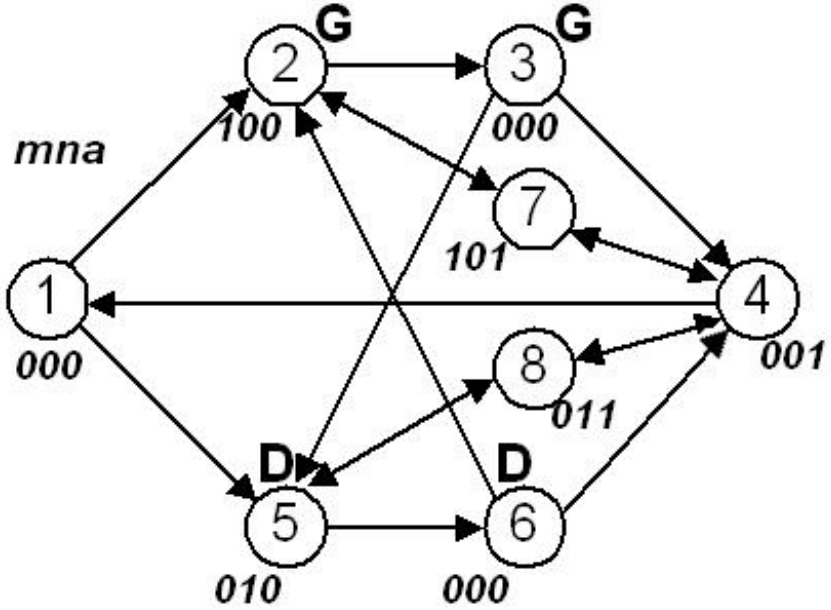
- **Pour la combinaison d'entrées 000, on retrouve trois états: 1, 3, 6**
- **Ce qui les distingue ?**
  - 1 -> G=0, D=0
  - 3 -> G=1, D=0
  - 6 -> G=0, D=1
- **Nous pouvons définir:**
  - x = G
  - y = D



Les méthodes intuitives

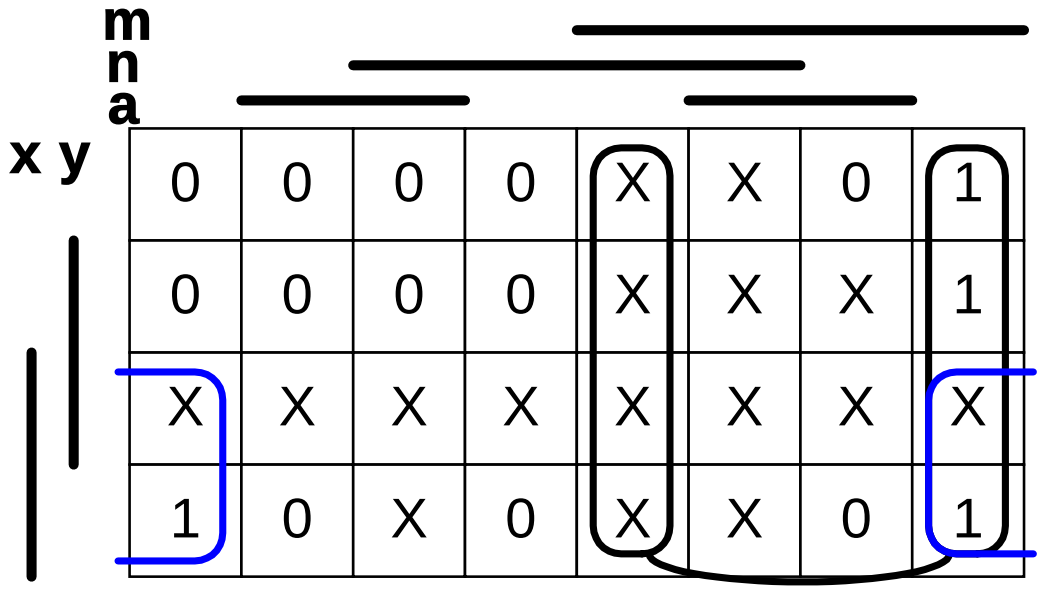
x = G

y = D



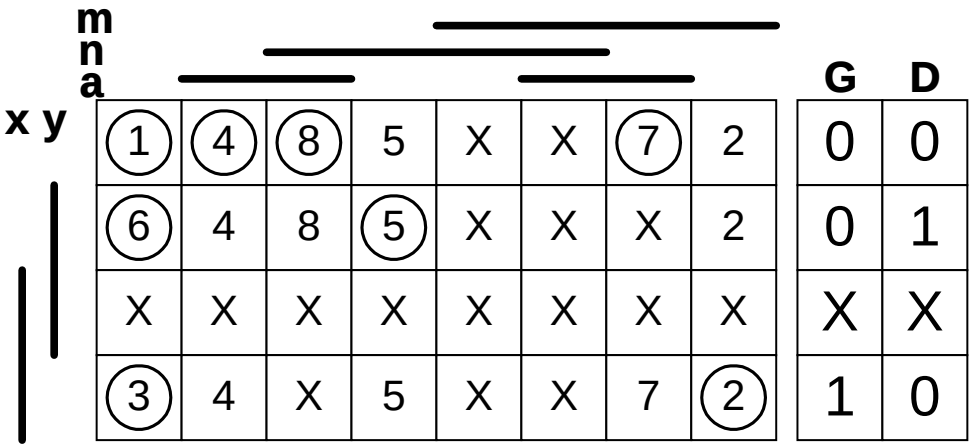
x y \ mna								G	D
	(1)	(4)	(8)	5	X	X	(7)	2	0	0
	(6)	4	8	(5)	X	X	X	2	0	1
	X	X	X	X	X	X	X	X	X	X
	(3)	4	X	5	X	X	7	(2)	1	0

Les méthodes intuitives



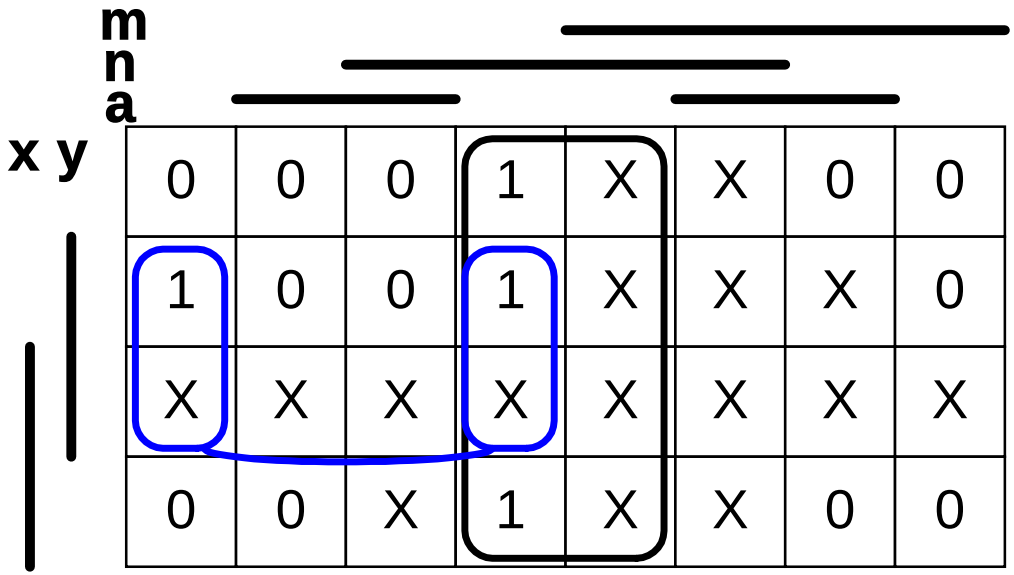
x y \ m n a	000	001	011	010	110	111	101	100
00	0	0	0	0	X	X	0	1
01	0	0	0	0	X	X	X	1
11	X	X	X	X	X	X	X	X
10	1	0	X	0	X	X	0	1

$$x = (m\bar{a} + x\bar{n}\,\bar{a})\,\bar{y} = (m + x\bar{n})\,\bar{a}\,\bar{y}$$



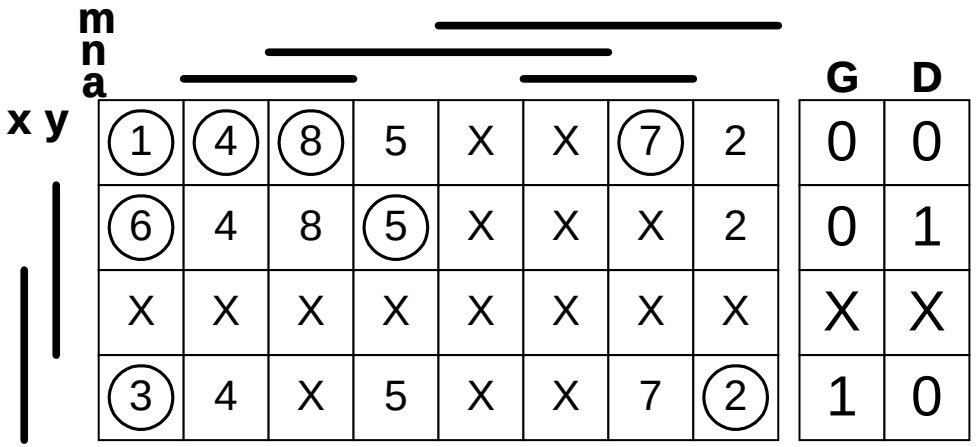
x y \ m n a	000	001	011	010	110	111	101	100	G	D
00	1	4	8	5	X	X	7	2	0	0
01	6	4	8	5	X	X	X	2	0	1
11	X	X	X	X	X	X	X	X	X	X
10	3	4	X	5	X	X	7	2	1	0

Les méthodes intuitives



x y \ m n a								
	0	0	0	1	X	X	0	0
	1	0	0	1	X	X	X	0
	X	X	X	X	X	X	X	X
	0	0	X	1	X	X	0	0

$$y = \left(n\bar{a} + y\overline{m}\,\bar{a}\right)\bar{x} = \left(n + y\overline{m}\right)\bar{a}\,\bar{x}$$



x y \ m n a									G	D
	(1)	(4)	(8)	5	X	X	(7)	2	0	0
	(6)	4	8	(5)	X	X	X	2	0	1
	X	X	X	X	X	X	X	X	X	X
	(3)	4	X	5	X	X	7	(2)	1	0

Les méthodes intuitives

- **Solution**

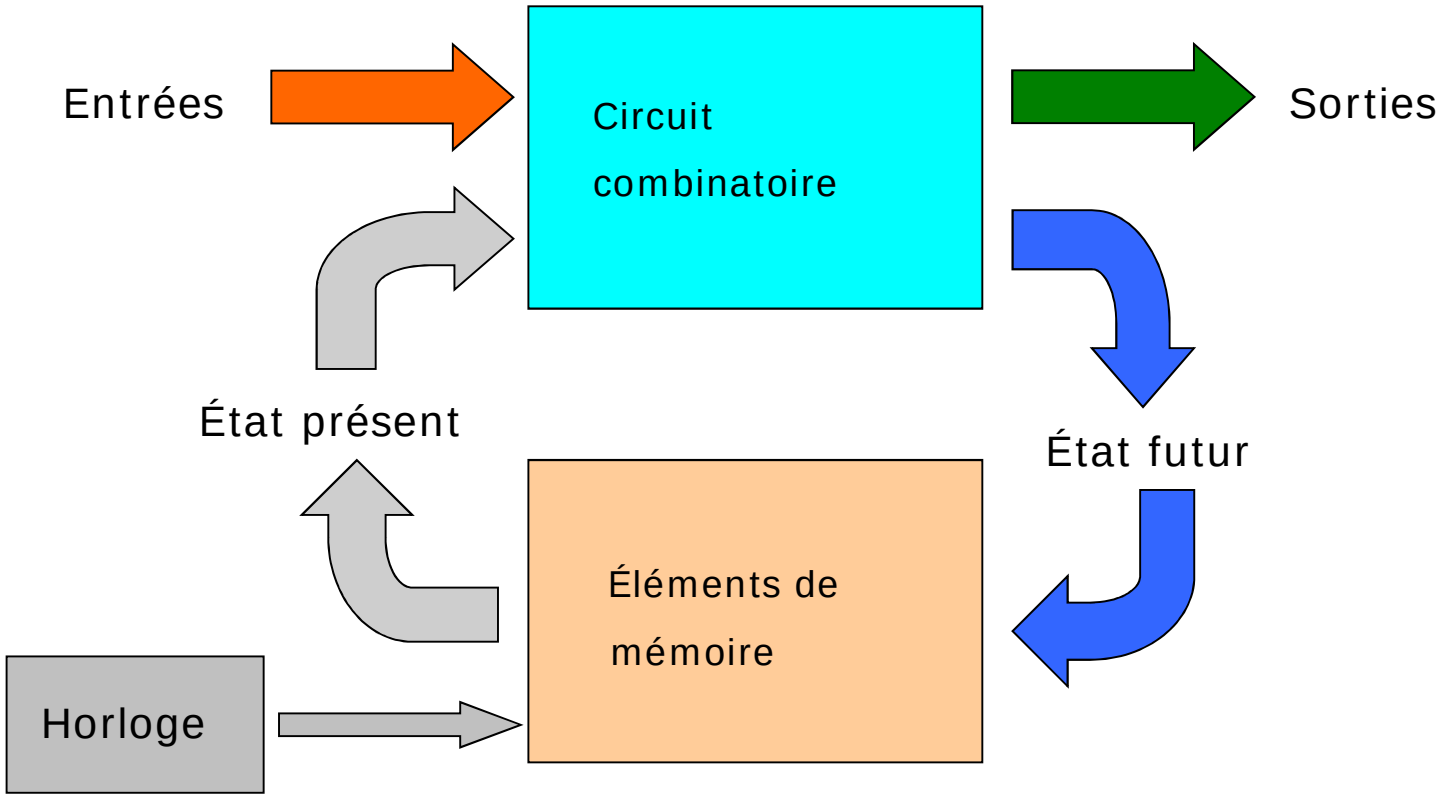
$$x = (m\bar{a} + x\bar{n}\ \bar{a})\bar{y} = (m + x\bar{n})\bar{a}\ \bar{y}$$

$$y = (n\bar{a} + y\bar{m}\ \bar{a})\bar{x} = (n + y\bar{m})\bar{a}\ \bar{x}$$

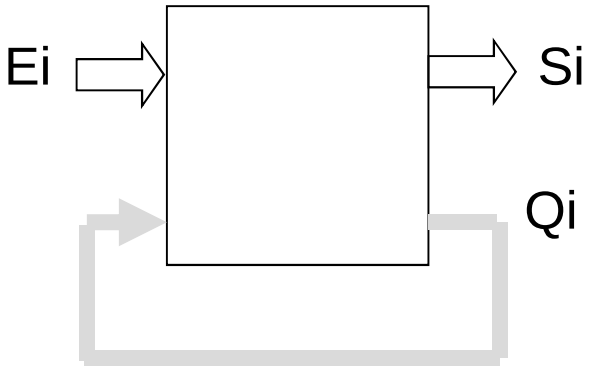
$$G = x \text{ et } D = y$$

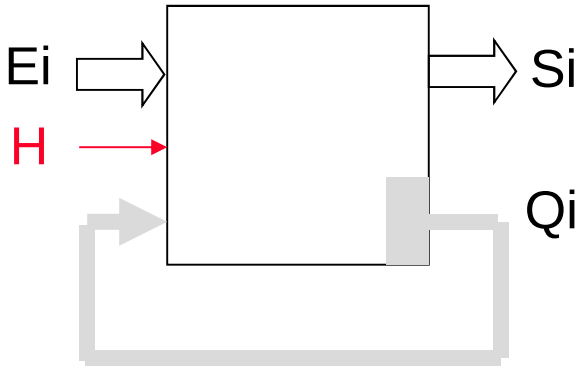
x y \ m n a								G	D
	(1)	(4)	(8)	5	X	X	(7)	2	0	0
	(6)	4	8	(5)	X	X	X	2	0	1
	X	X	X	X	X	X	X	X	X	X
	(3)	4	X	5	X	X	7	(2)	1	0

Notion de circuit séquentiel



Synchrone / Asynchrone





- **<u>Définition:</u> Un système séquentiel est *asynchrone* si à partir de l'instant ou on applique un vecteur d'entrée, son évolution est incontrôlable de l'extérieur.**

- **<u>Définition:</u> Un système séquentiel est *synchrone* si son évolution est contrôlable de l'extérieur par un signal d'horloge**

2 : Bascule

Bascules

- **Les bascules sont des éléments de mémoire simple**
- **Elles permettent de conserver l'état de la bascule**
- **Élément de base : 2 inverseurs en contre-réaction positive**

Bascules RS

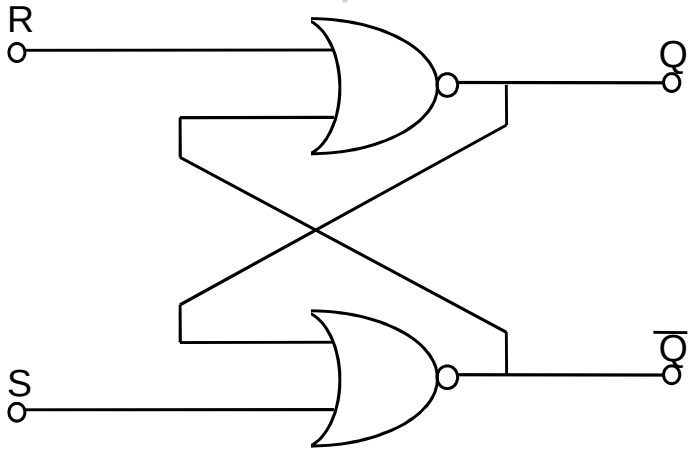
La bascule RS est la bascule de base.

Toutes les bascules ne sont en fait que des améliorations de cette bascule.

La bascule RS est un dispositif à deux entrées R et S et une sortie Q présentant la propriété suivante: une apparition (même fugitive) de S entraîne durablement Q=1, tandis qu'une apparition (même fugitive de R) entraîne durablement Q=0.

2 : Bascule

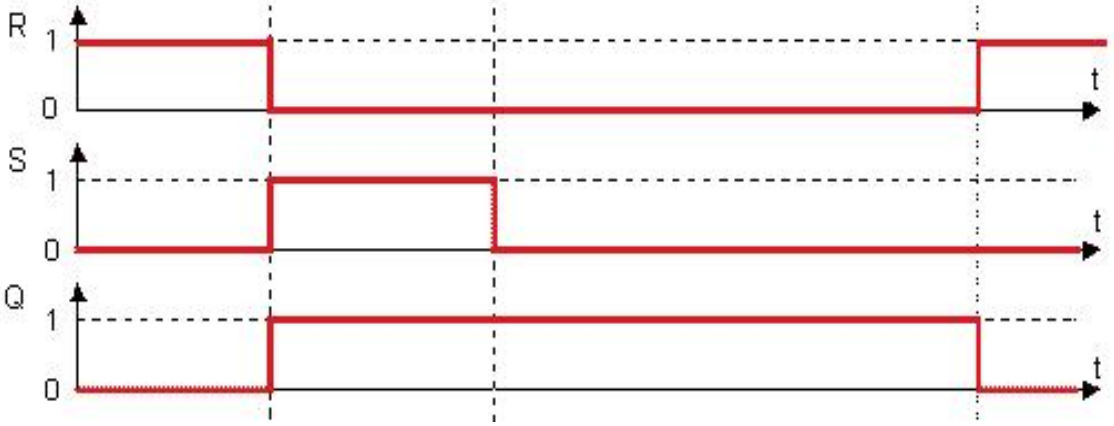
Bascule SR - portes NON-OU (NOR)



-RS flip-flop using NOR gates

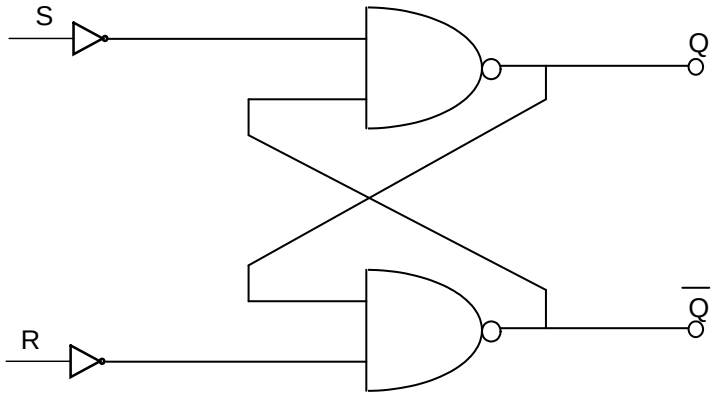
Bascule SR avec portes NON-OU			
R	S	Q(t+1)	/Q(t+1)
0	0	Q(t)	/Q(t)
0	1	1	0
1	0	0	1
1	1	0	0

État non permis



2 : Bascule

Bascule /S/R - portes NON-ET (NAND)



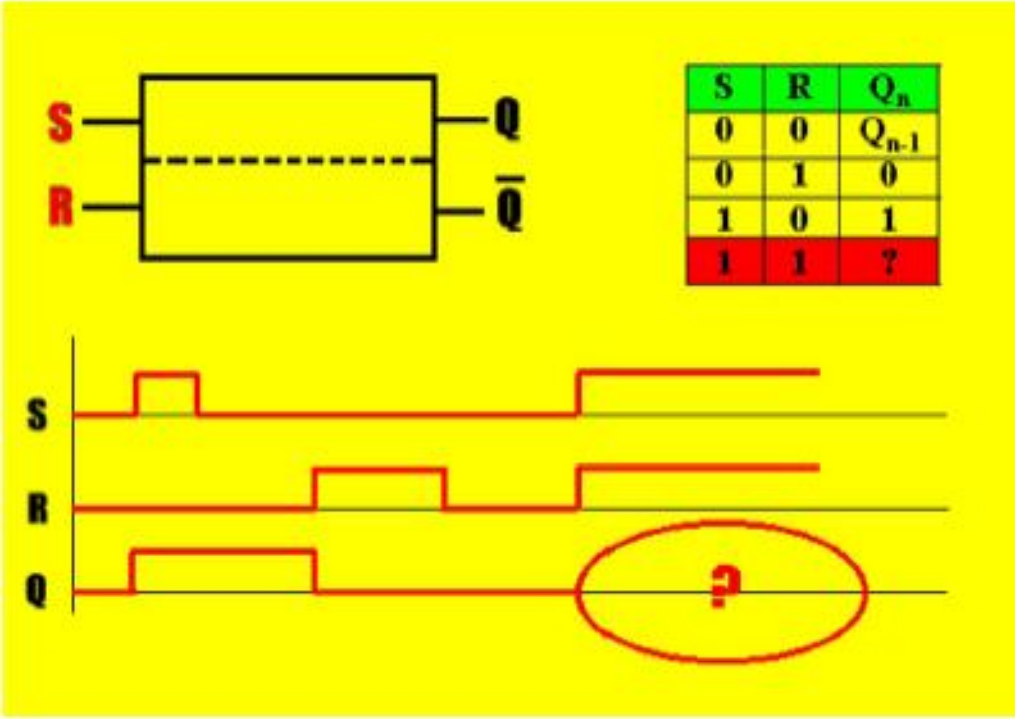
- A NAND $\overline{R}\overline{S}$ flip-flop

Bascule SR avec portes NON-ET			
$\overline{S}$	$\overline{R}$	Q(t+1)	/Q(t+1)
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q(t)	/Q(t)

État non permis

2 : Bascule

Une bascule est donc une mémoire, fonction logique dont les 2 états (0 et 1) restent maintenus même après disparition de la commande.



S	R	Q_n
0	0	Q_{n-1}
0	1	0
1	0	1
1	1	?

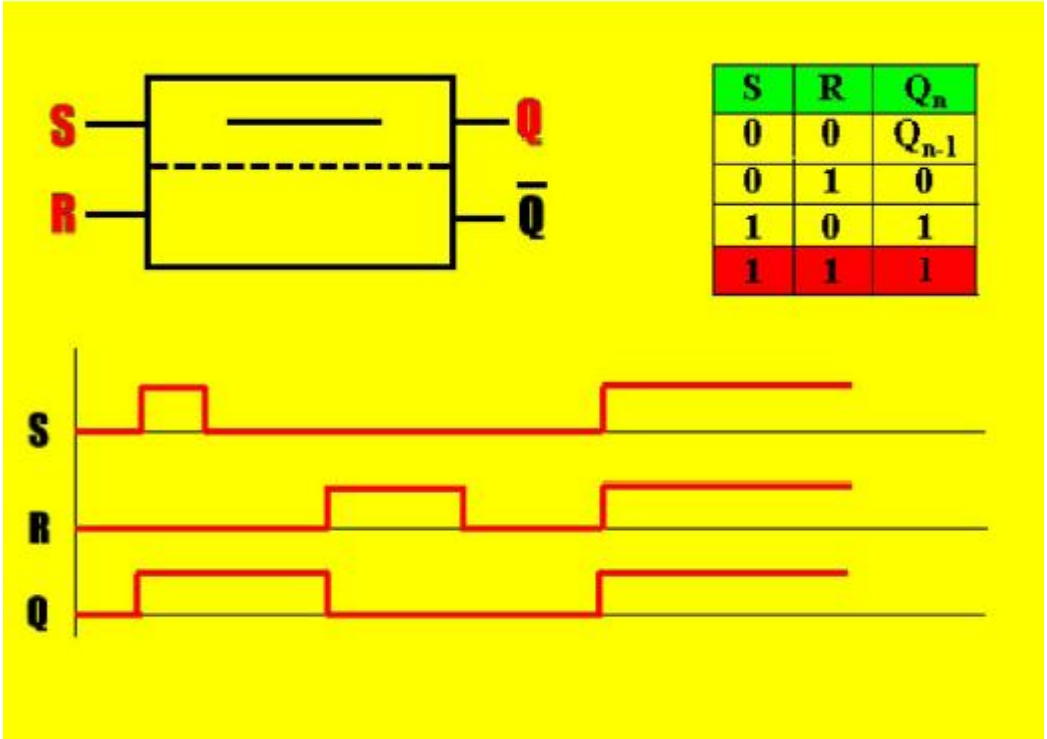
L'état où R=1 et S=1 doit être défini par un effacement prioritaire ou une écriture prioritaire

2 : Bascule

Écriture Prioritaire

S = 1 et R=1 donne Q=1 .

$$Q = S + \overline{R} \cdot Qn\text{-}1$$

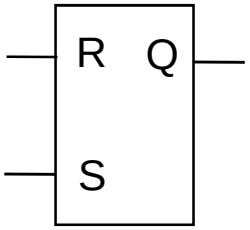


S	R	Q_n
0	0	Q_{n-1}
0	1	0
1	0	1
1	1	1

Bascule RS

La bascule RS est un dispositif à deux entrées R et S et une sortie Q présentant la propriété suivante:

- une apparition (même fugitive) de S entraîne durablement Q=1
- une apparition (même fugitive de R) entraîne durablement Q=0.



Qn	S	R	Qn+1
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

Qn \ SR	00	01	11	10
0	0	0	X	1
1	1	0	X	1

Qn+1

- **L'énoncé du problème est incomplet: les combinaisons (3) et (7) ne sont pas définies. Elles correspondent à des ordres d'enclenchement (SET) et de déclenchement (RESET) simultanés. En laissant le problème incomplètement spécifié, on peut obtenir plusieurs équations de la bascule**

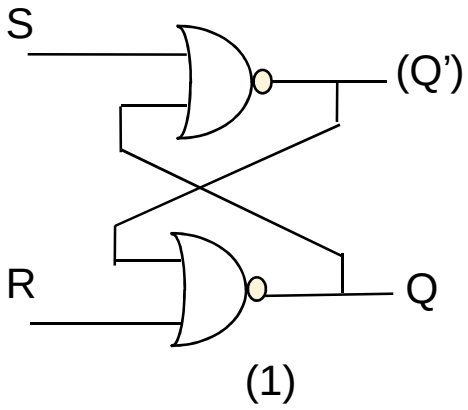
Bascule RS

$Q_{n+1} = S.R' + Q_n.R' = (S + Q_n)\ R' = ((S + Q_n)' + R)'$ (1)

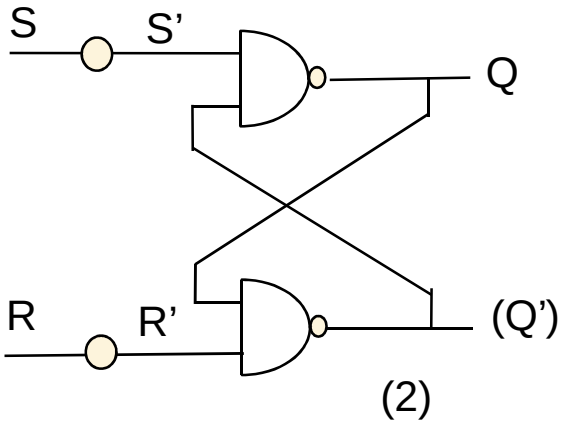
$Q_{n+1} = S + Q_n.R' = (S' . (Q_n.R)')'$ (2)

Qn \ SR	00	01	11	10
0	0	0	X	1
1	1	0	X	1

Qn+1



(1)



(2)

S	R	Qn+1
0	0	Qn
0	1	0
1	0	1
1	1	Interdit

- **(R,S)=(1,1) introduit une indétermination.**
  - En effet, le passage de la combinaison (R,S)=(1,1) à (R,S)=(0,0) entraîne deux valeurs possibles sur Q selon que R ou S commute en premier.
- **Si l'on interdit la combinaison (R,S)=(1,1) on remarque que sur les deux structures, la connexion symétrique de la sortie Q porte la valeur Q'.**

Bascule RS

- **Avantages:**
  - Simplicité

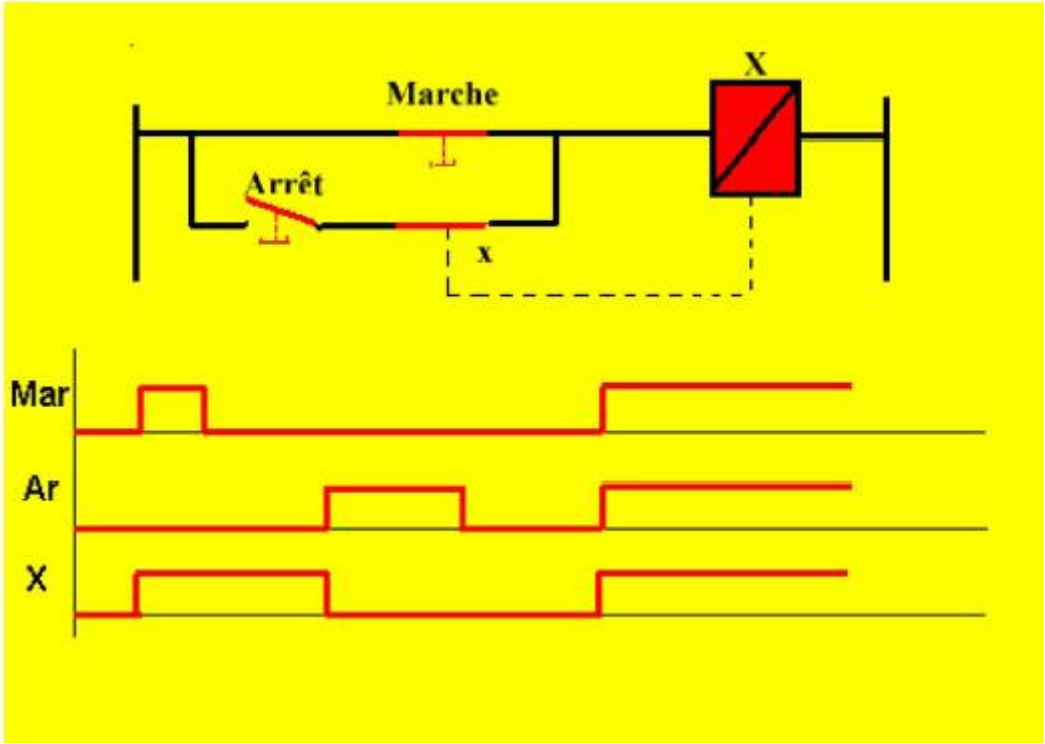
- **Inconvénients**
  - Dispositif asynchrone
  - Etat interdit
  - Sensibilité aux parasites (transitoires)

2 : Bascule

Écriture Prioritaire

$$X = Marche + \overline{Arrêt} \cdot x$$

Marche : mise à 1 (contact à E.C.)
Arrêt : mise à 0 (contact à R.C.)
X : Bobine relais
x : contact associé au relais
Le maintien est obtenu par auto alimentation.

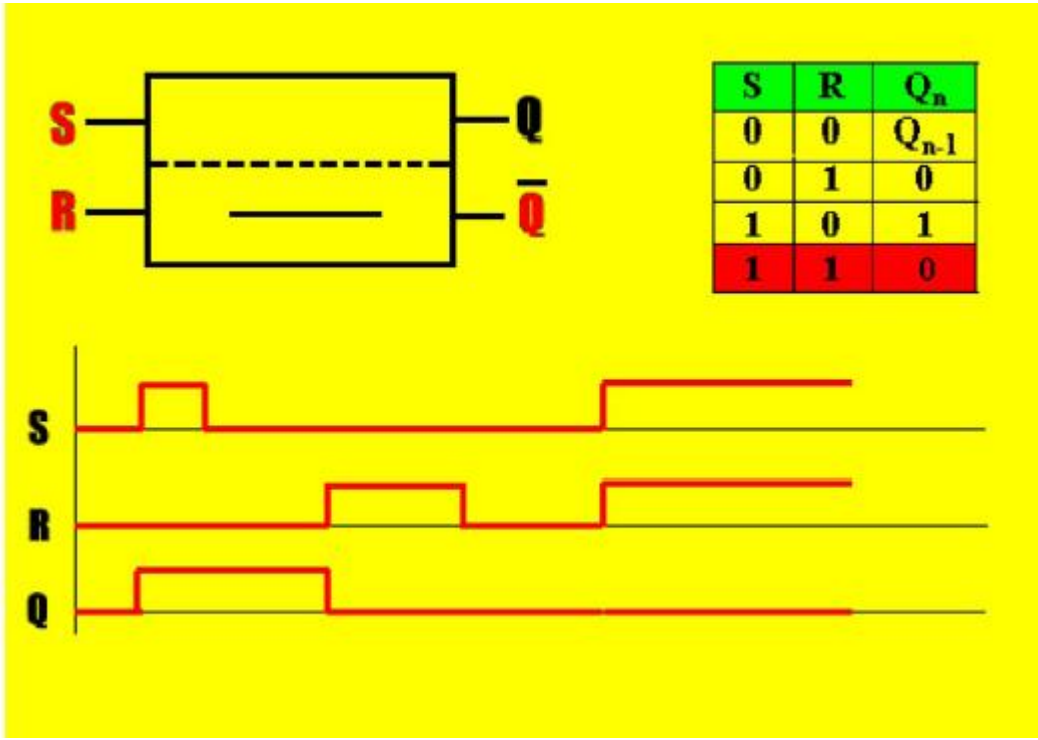


2 : Bascule

Effacement Prioritaire

S = 1 et R=1 donne Q=0

$$Q = \overline{R} . (S + Qn-1)$$

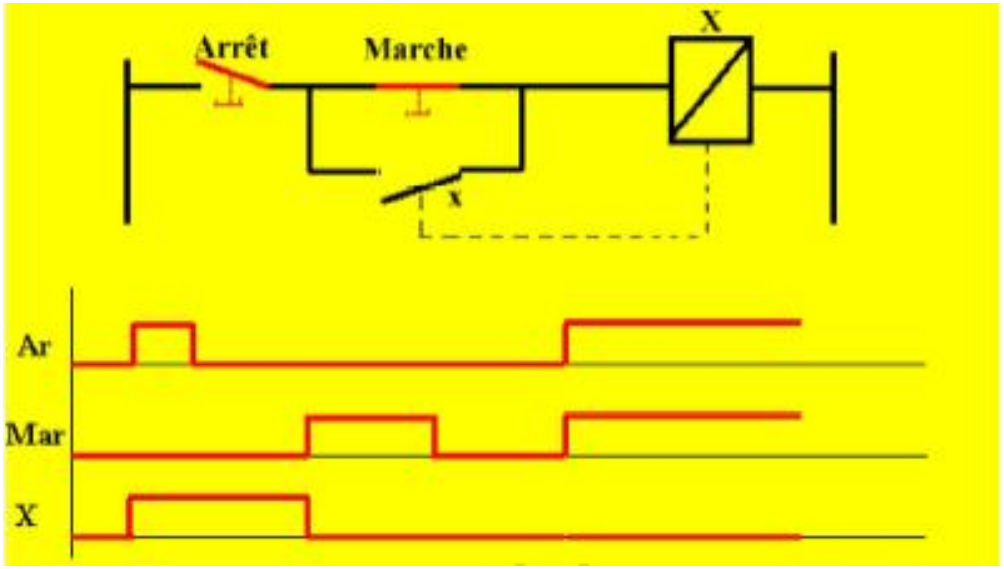


S	R	Q_n
0	0	Q_{n-1}
0	1	0
1	0	1
1	1	0

2 : Bascule

Effacement Prioritaire

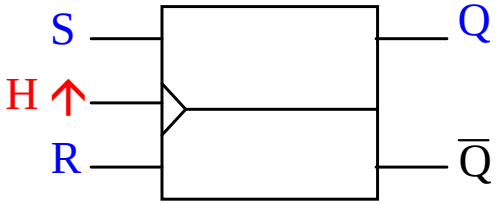
$$X = \overline{\text{Arrêt}}\,(\text{Marche} + x)$$

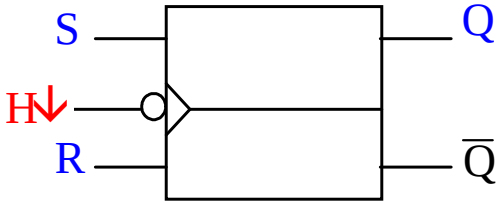


Marche : mise à 1 (contact à E.C.)
Arrêt : mise à 0 (contact à R.C.)
X : Bobine relais
x : contact associé au relais
Le maintien est obtenu par auto-alimentation

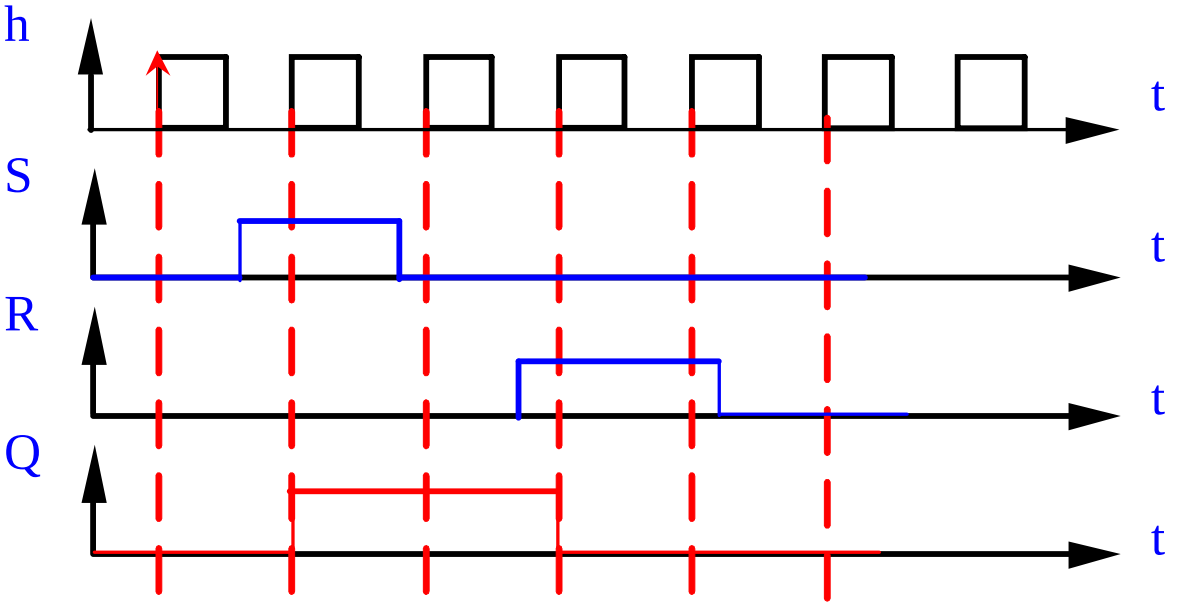
2 : Bascule

Bascule RS H synchrone



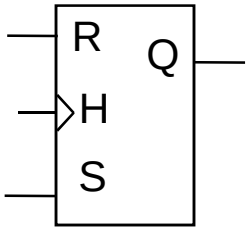


Bascule RSH				
H	S	R	Q(t+1)	/Q(t+1)
0	X	X	Q(t)	/Q(t)
1	0	0	Q(t)	/Q(t)
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1



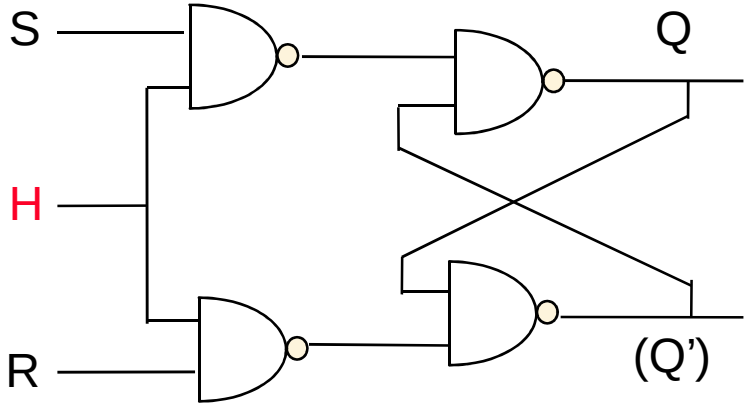
Bascule RSH

La bascule RSH est une bascule RS synchronisée par un signal d'horloge H. Lorsque H=0, la bascule est dans l'état mémoire. Lorsque H=1, la bascule fonctionne comme une bascule RS. Cette bascule a toujours un état interdit et fonctionne sur les niveaux d'horloge.



S	R	Qn+1
0	0	Qn
0	1	0
1	0	1
1	1	Interdit

H = 1



Bascule RSH

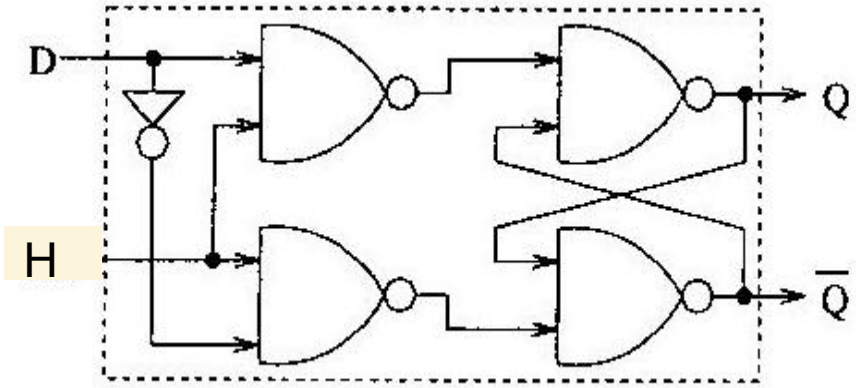
- **Fonctionnement sur niveau de l'horloge (H=1)**

- **Avantages:**
  - Insensibilité aux parasites (H=0)

- **Inconvénients**
  - Etat interdit
  - Sensibilité aux parasites (H=1)

2 : Bascule

Bascule D-latch

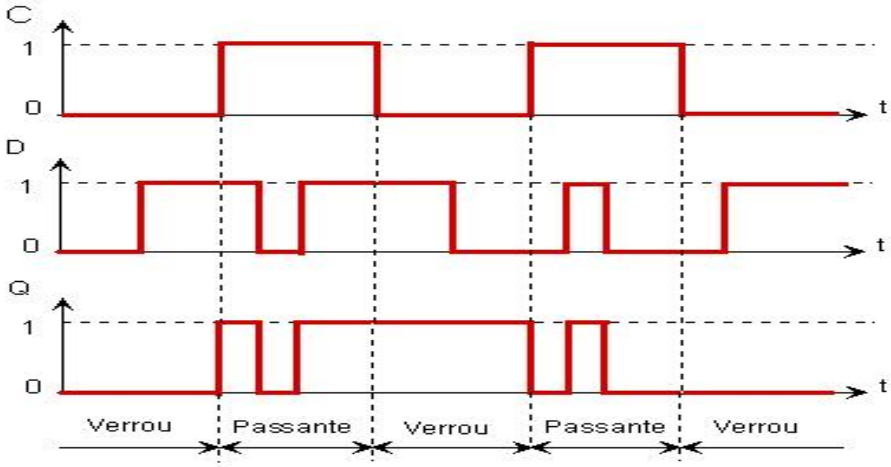


Bascule D-latch			
H	D	Q(t+1)	/Q(t+1)
0	X	Q(t)	/Q(t)
1	0	0	1
1	1	1	0

Q = D

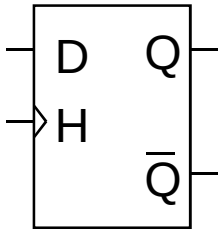
L'entrée D permet d'éliminer l'état non permis

La bascule D-Latch est une bascule conçue sur le même principe que la RSH. Elle est obtenue à partir d'une bascule RSH en ne considérant que les deux combinaisons (R,S) = (0,1) et (1,0). La D-Latch n'a qu'une seule entrée nommée D,



Bascule D-latch

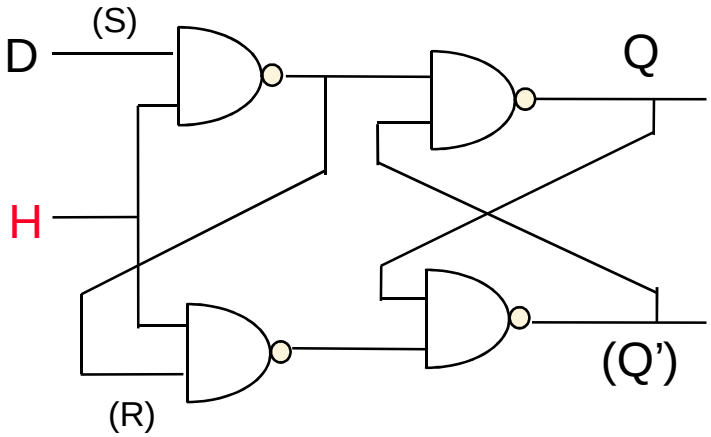
La bascule D-Latch est une bascule conçue sur le même principe que la RSH. Elle est obtenue à partir d'une bascule RSH en ne considérant que les deux combinaisons (R,S) = (0,1) et (1,0).

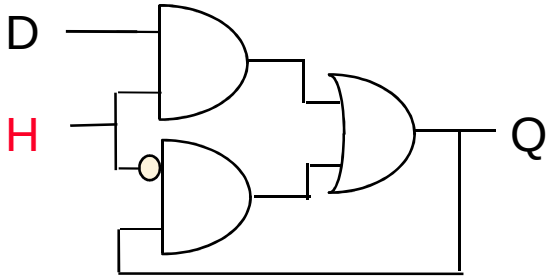


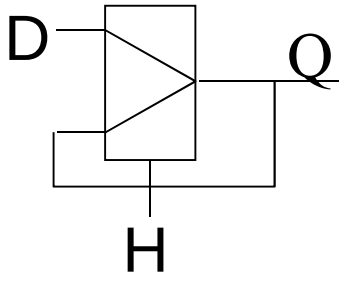
D	Q(n+1)
0	0
1	1

Qn+1 = Dn

H = 1







Bascule D-latch

- **Fonctionnement sur niveau de l'horloge (H=1)**

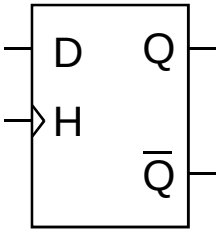
- **Avantages:**
  - Pas d'état interdit
  - Insensibilité aux parasites (H=0)

- **Inconvénients**
  - Sensibilité aux parasites (H=1)

Bascule D (Maître-Esclave)

Les bascules maître-esclaves permettent de diminuer la sensibilité aux parasites en minimisant la période de transparence. Elles fonctionnent sur le front d'horloge.

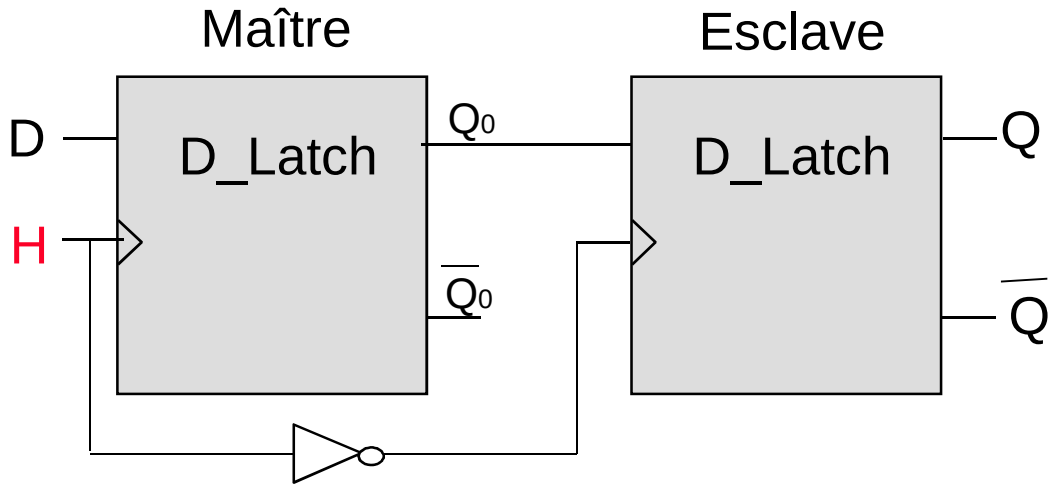
La bascule D maître-esclave est constituée de 2 D-Latch en cascade.



D	Q(n+1)
0	0
1	1

$Q_{n+1} = D_n$

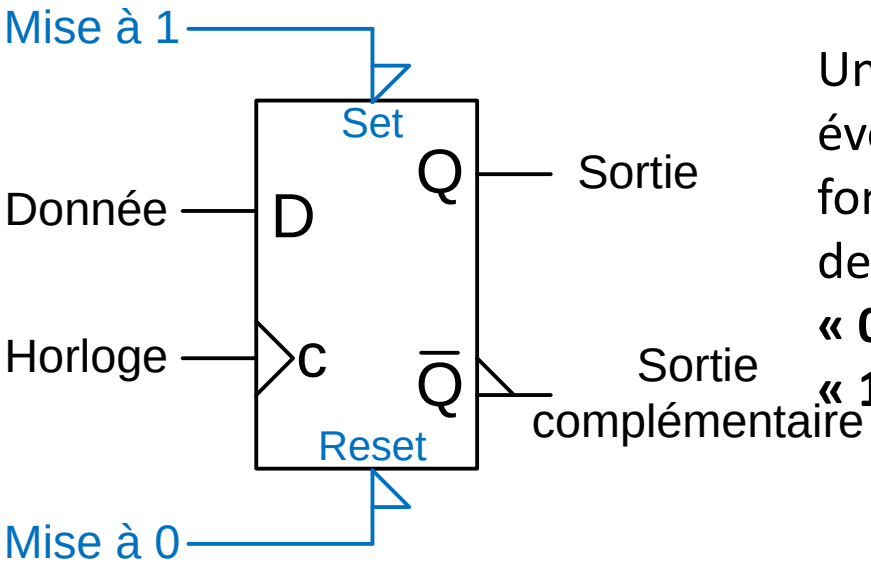
H = ↓



Bascule D

Symbole

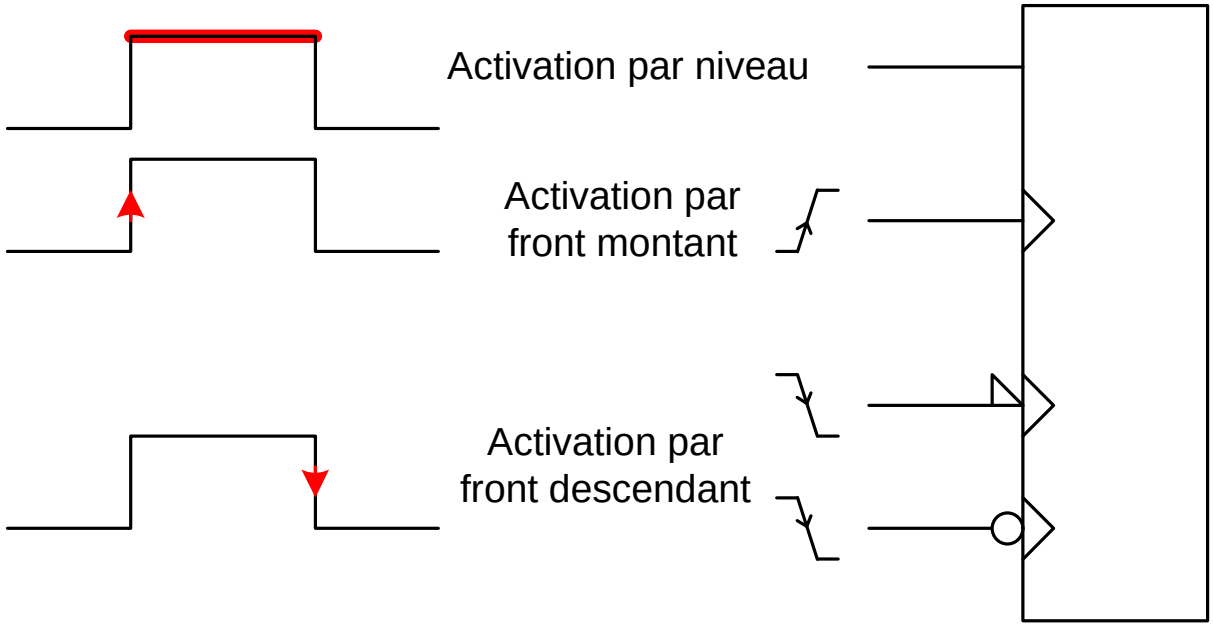
Options



Un front d'horloge est un événement unique qui active une fonction au moment du passage de

« 0 » vers « 1 » front montant

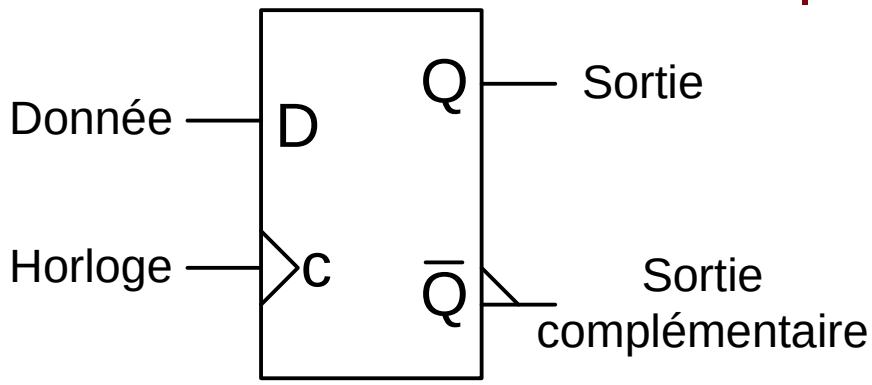
« 1 » vers « 0 » front descendant

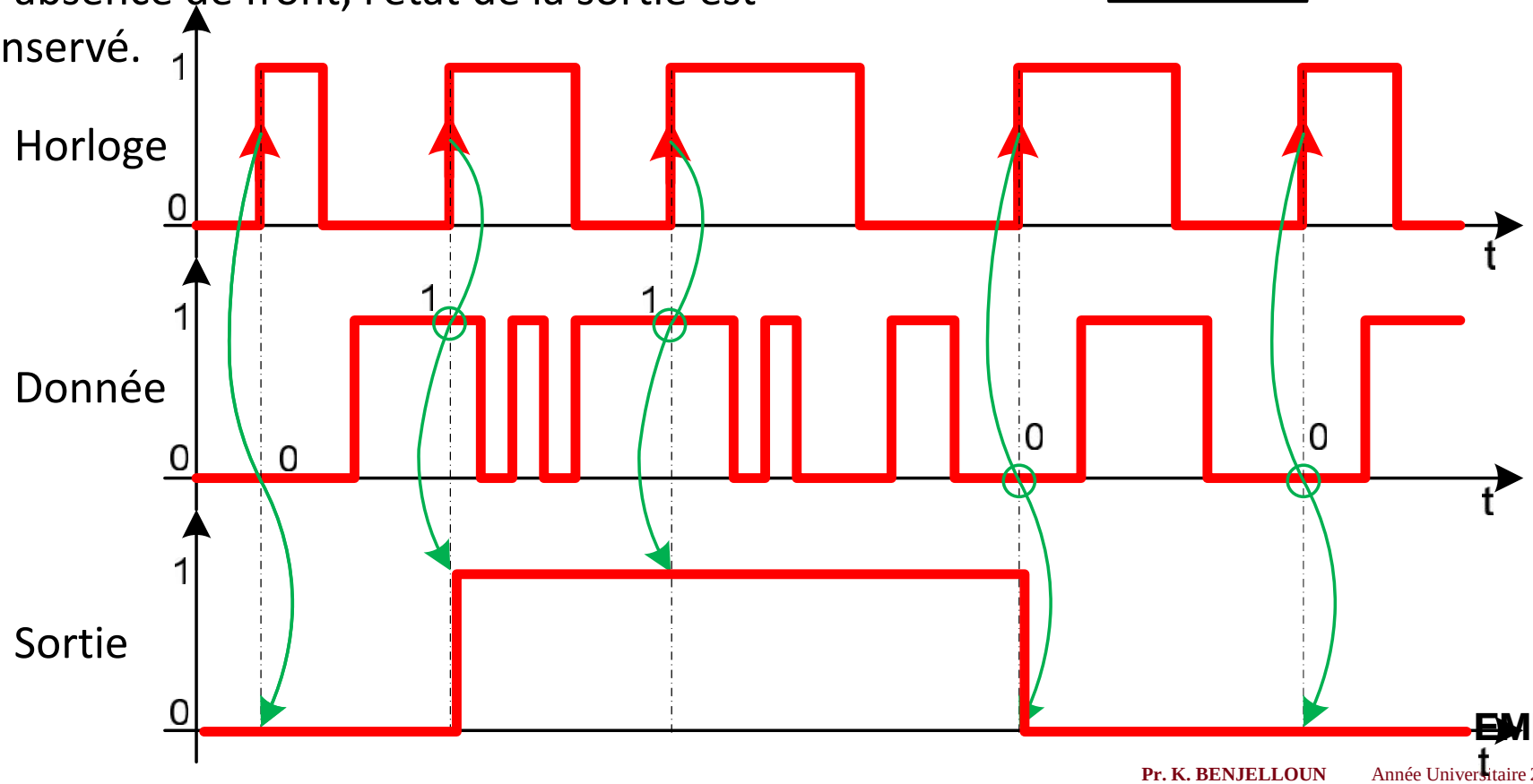


Fonctionnement

À chaque front (ici montant) d'horloge, le niveau logique en D est transféré en sortie Q.

En absence de front, l'état de la sortie est conservé.



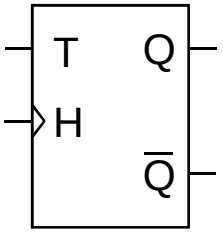


Bascule D (Maître-Esclave)

- **Fonctionnement sur fronts d'horloge**
- **Mémorisation (transfert) de données**
- **Avantages:**
  - Dispositif synchrone
  - Pas d'état interdit
  - Insensibilité aux parasites

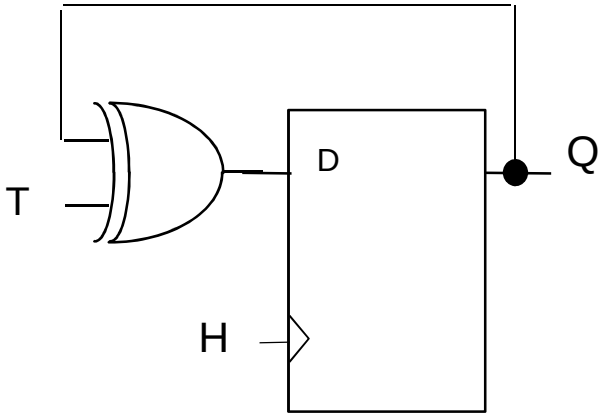
Bascule T (Toggle)

Comme la bascule D, la bascule T fonctionne sur front d'horloge. Elle permet de conserver la valeur de sortie précédente ou de l'inverser. Ce type de bascule est particulièrement intéressant pour la réalisation de compteurs. La bascule T peut être réalisée à partir d'une bascule D.



T	Q(n+1)
0	Q(n)
1	Q'(n)

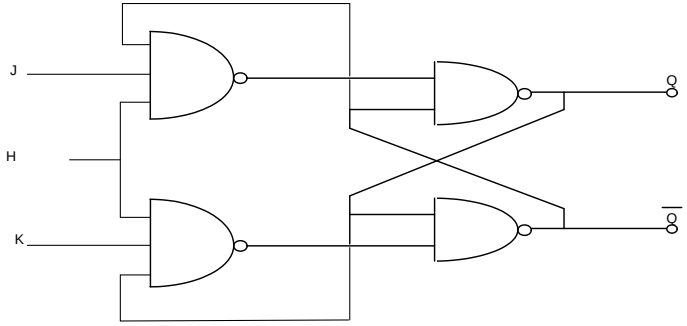
H = ↓



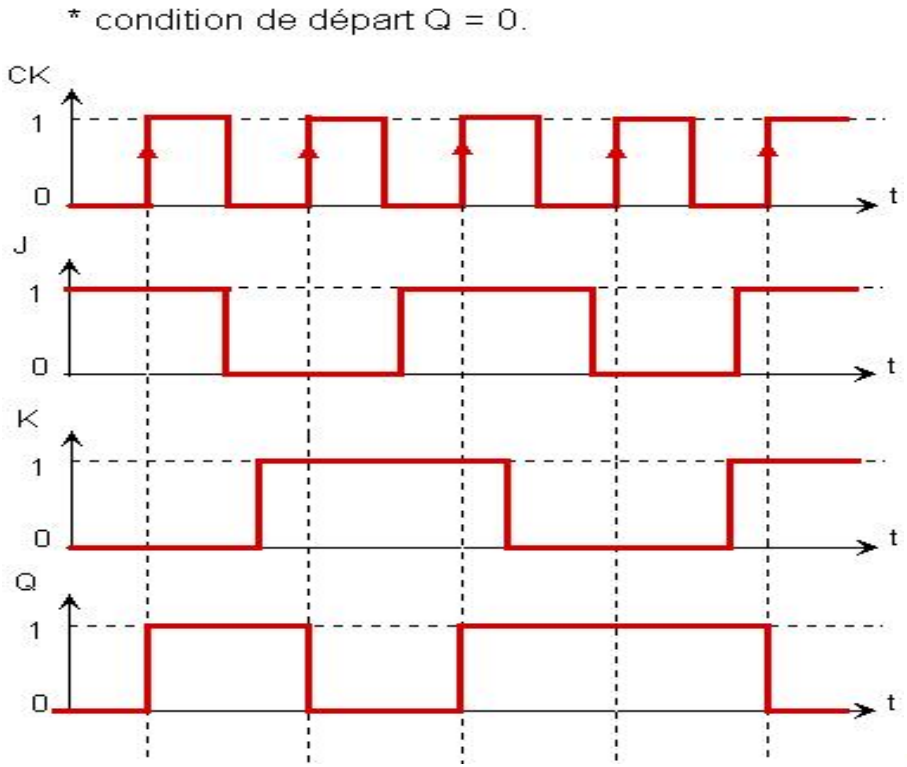
2 : Bascule

Bascule JKH

La bascule JKH (simple étage) est obtenue à partir d'une bascule R S H dont les sorties sont rebouclées sur les entrées. Ceci permet d'éliminer l'état indéterminé

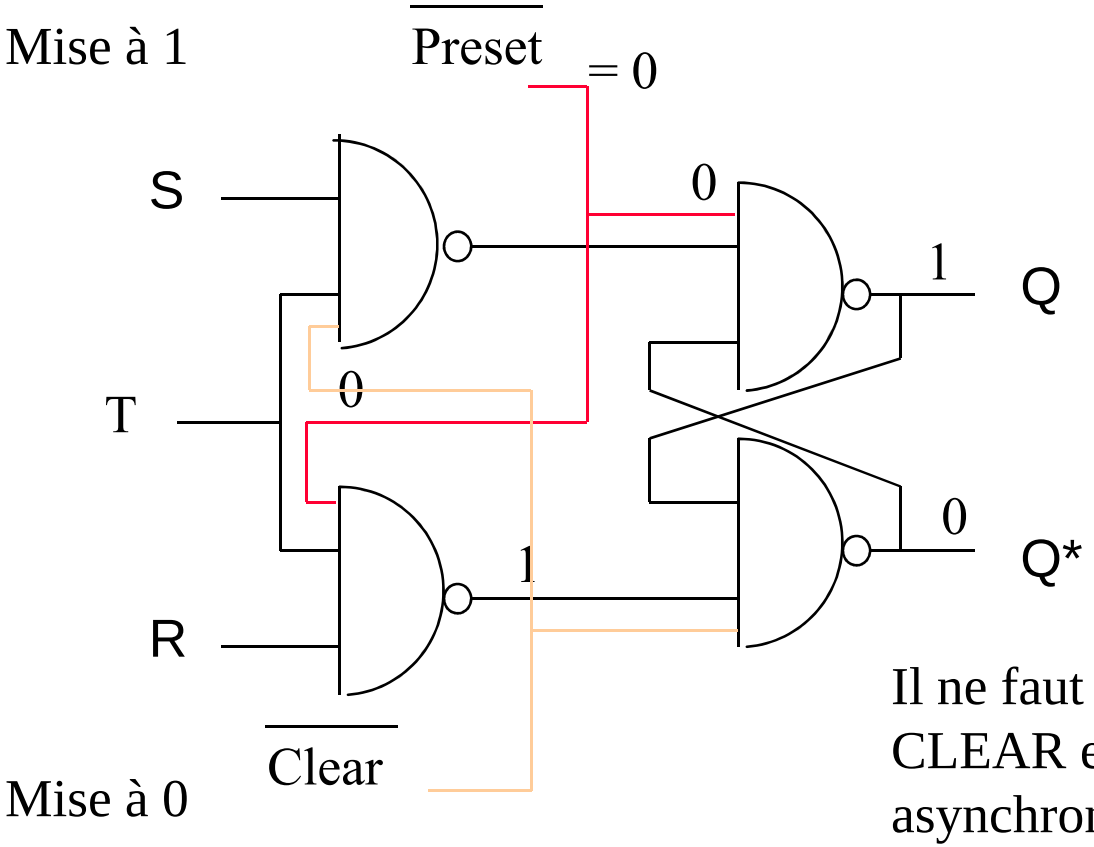


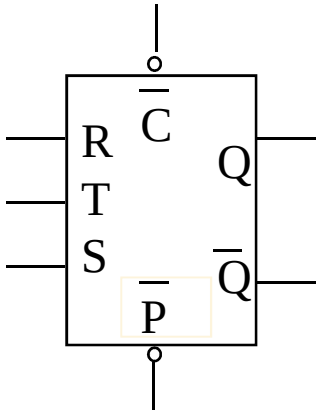
Entrées			Sorties	
CK	J	K	Qn+1	$\overline{Qn+1}$
0	X	X	Qn	$\overline{Qn}$
1	X	X	Qn	$\overline{Qn}$
↓	X	X	Qn	$\overline{Qn}$
↑	0	0	Qn	$\overline{Qn}$
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	$\overline{Qn}$	Qn



2 : Bascule

Bascule RST : initialisation





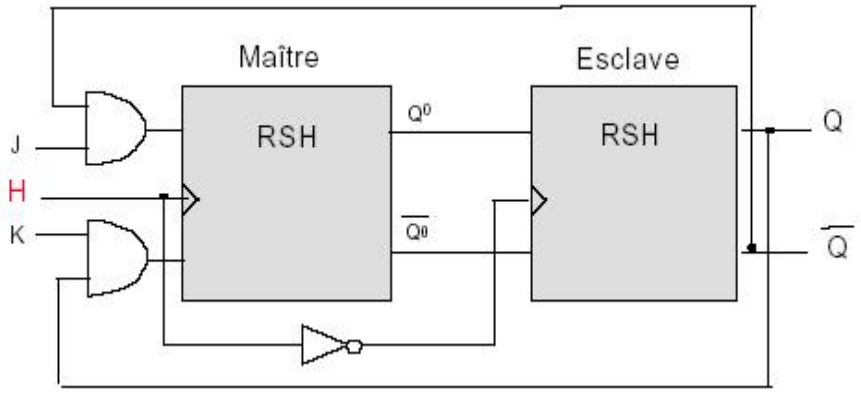
Il ne faut pas CLEAR=PRESET=0
CLEAR et PRESET commandes asynchrones

2 : Bascule

Bascule JKH Maitre-Esclave.

La bascule JKH est une bascule maître-esclave ne présentant plus d'état interdit. Sachant que les sorties sont toujours complémentaires, leur rebouclage sur les entrées élimine l'état interdit. Il n'y a pas d'inconvénient à ce rebouclage car les sorties de l'esclave ne change d'état que lorsque le maître est bloqué. Cette bascule fonctionne toujours sur les front descendant.

$$Qn+1 = J.Qn' + K'.Qn$$

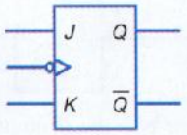


JK	Q(n+1)
00	Qn
01	0
10	1
11	Qn'

2 : Bascule

Bascule JKH Maitre-Esclave.

Table de vérité



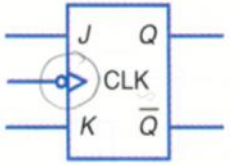
(a)

CLK	J	K	Q
0	X	X	NC
1	X	X	NC
↑	X	X	NC
X	0	0	NC
↓	0	1	0
↓	1	0	1
↓	1	1	Toggle

(b)

FIG. 31-46 Negative edge-triggered *JK* flip-flop. (*a*) Logic diagram. (*b*) Truth table.

Chronogramme



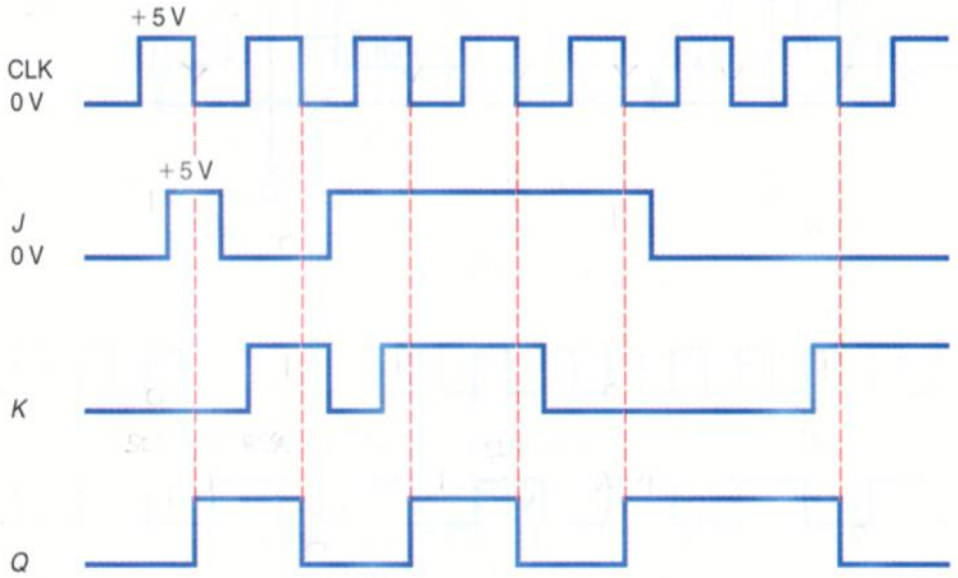
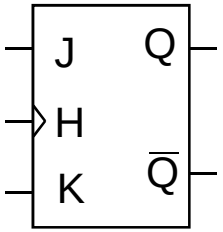


FIG. 31-47 Negative edge-triggered *JK* flip-flop with timing diagram to show how the *Q* output responds to *R*, *S*, and CLK inputs.

Bascule JK

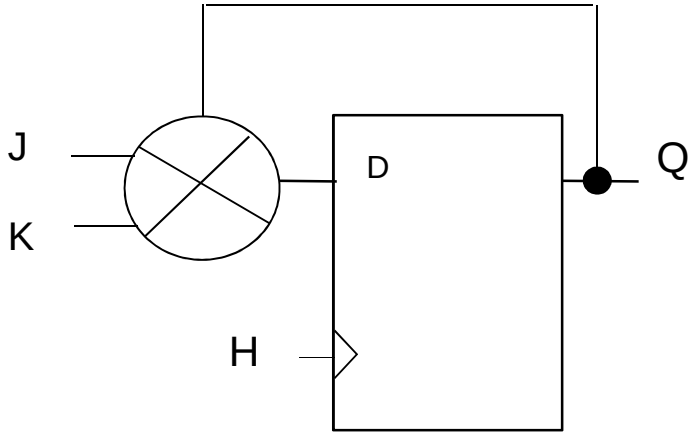
Comme la bascule D, la bascule JK est une bascule fonctionnant sur front. Elle dispose par contre de 2 entrées J et K. La bascule JK peut être réalisée à partir d'une bascule D.



JK	Q(n+1)
00	Qn
01	0
10	1
11	Qn '

$Qn+1 = J.Qn' + K'.Qn$

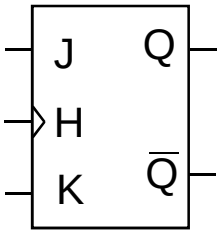
H = ↓



Avec D = J'K'Q + JK' + JKQ'

Bascule JK (autre réalisation)

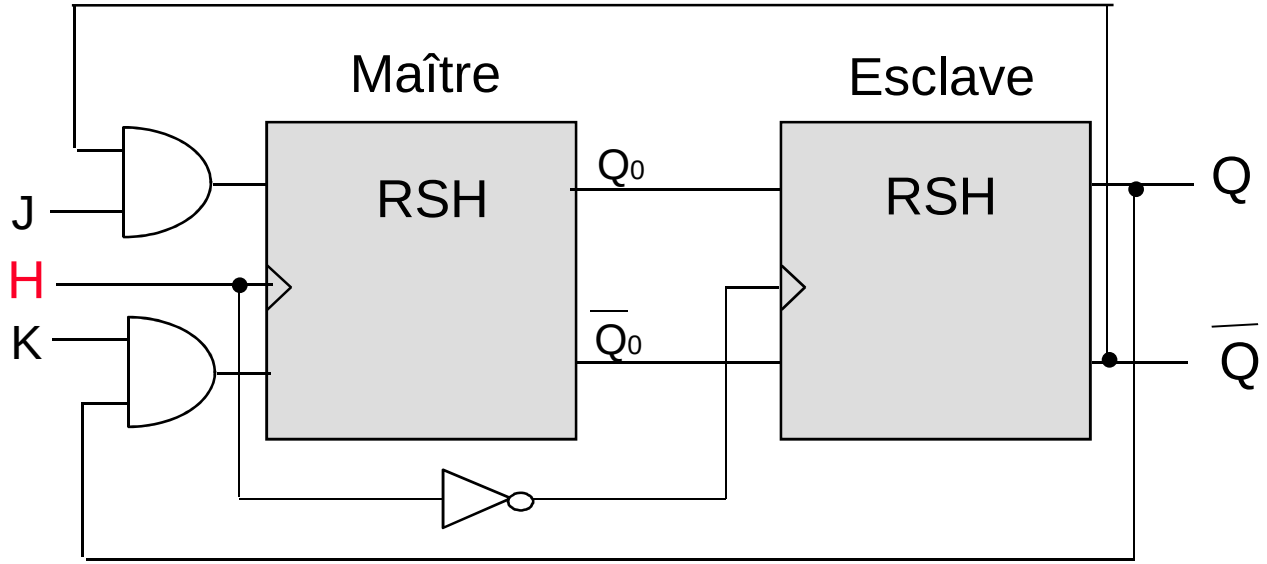
La bascule JK peut également être réalisée à partir de de 2 RSH en cascade et d ’un rebouclage permettant d’éliminer l’état interdit de la RSH.



JK	Q(n+1)
00	Qn
01	0
10	1
11	Qn ’

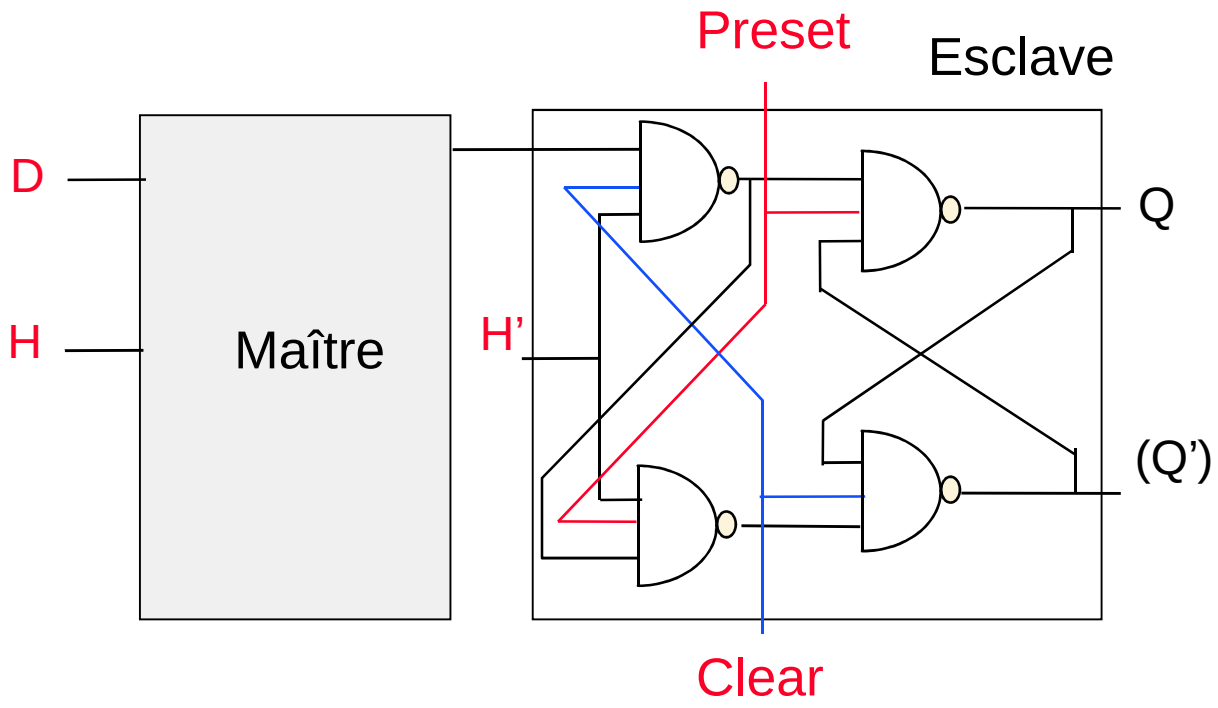
$Qn+1 = J.Qn' + K'.Qn$

H = ↓



Initialisation des bascules

Initialisation asynchrone par signaux de Preset (RAU) et Clear (RAZ).



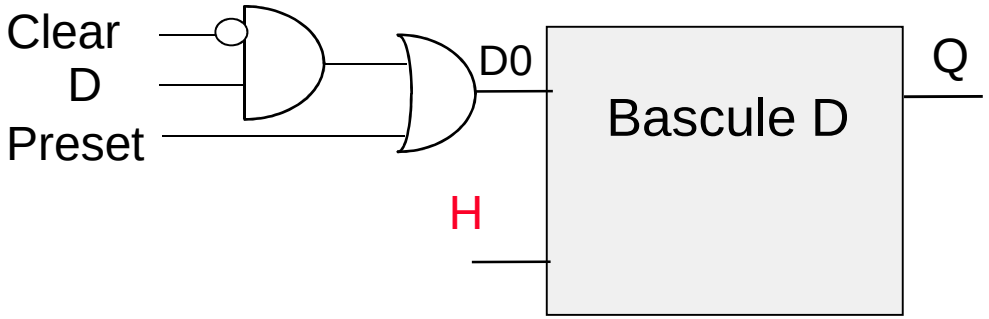
Preset et Clear actif sur niveau bas

Initialisation des bascules

Initialisation synchrone par signaux de Preset (RAU) et Clear (RAZ).

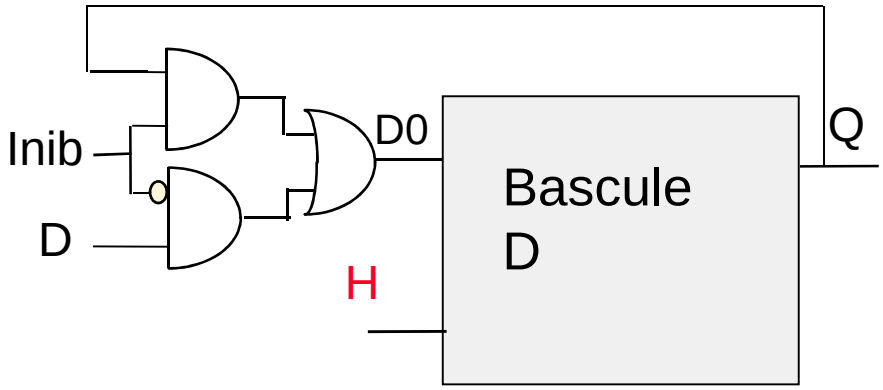
$D0 = \overline{Clear}.\overline{Preset}.D + Clear.0 + Preset.1$ (Preset Prioritaire)

$= Clear.D + Preset$

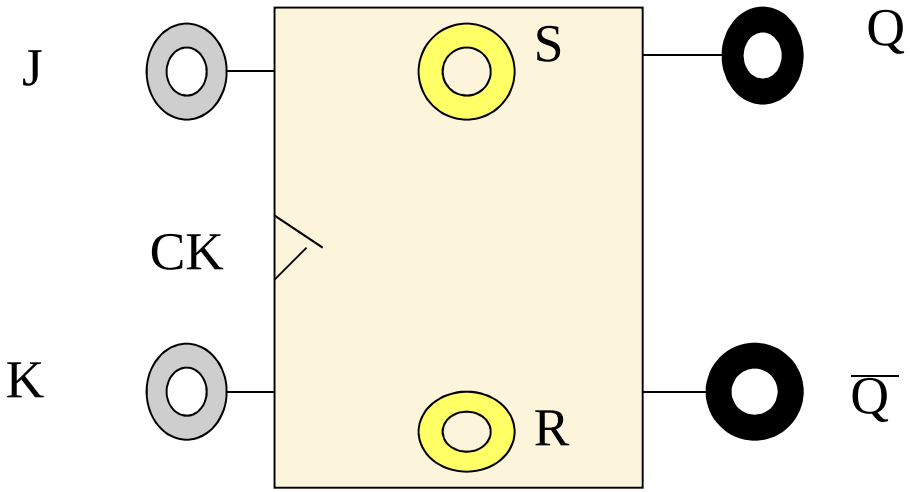


Inhibition des bascules

$D0 = \overline{Inib}.D + Inib.Q$



Bascule du laboratoire



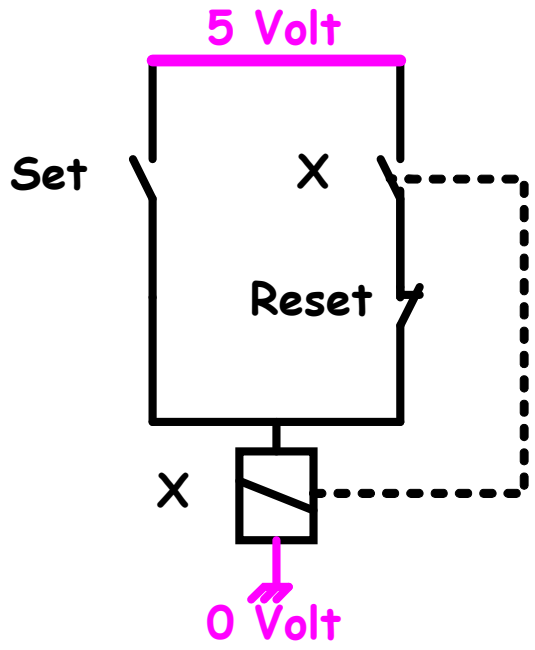
S=0 & R=1----> Q=1 (SET)

R=0 ---> maintien de Q à 1 (SET prioritaire)

R=0 & S=1 ----> Q=0 (RESET)

R=1 & S=1 ----> Q garde sa valeur précédente (mémoire)

3 : Réalisation par relais



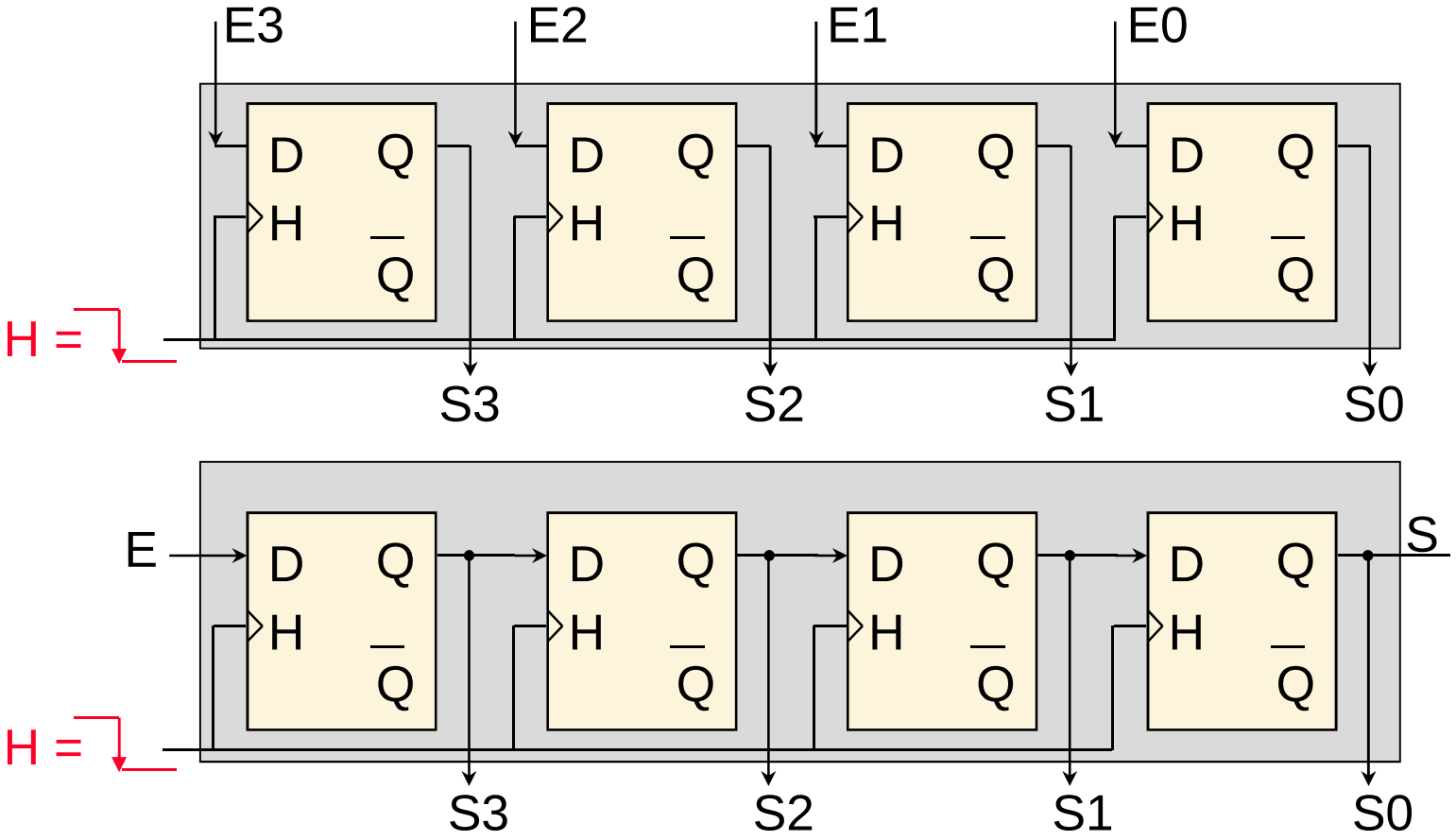
Ici la mémoire est réalisée par un relais et deux contacts

3 : Réalisation par relais

- **Les réalisations électriques câblées en logique séquentielle sont très courantes dans l'industrie. Elles sont basées sur des relais électriques; cependant il faut distinguer entre les relais de fonction et ceux d'interface.**
- **Généralement les relais de fonction peuvent être supprimés lors d'une commande programmée (ce qui génère une économie non négligeable) mais pas ceux destinés aux interfaces.**

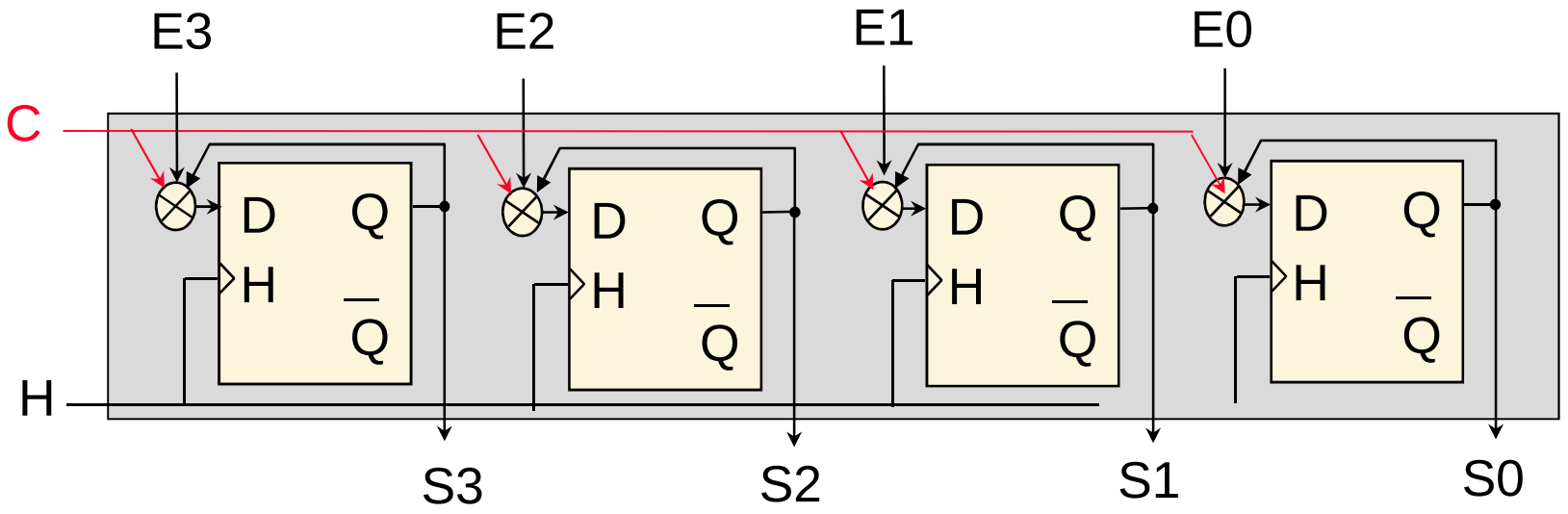
Registres

Les registres sont des associations de bascules permettant de mémoriser et de réaliser certaines opérations sur des mots logiques



Registres

Chargement // (C=1) et Inhibition (C=0)

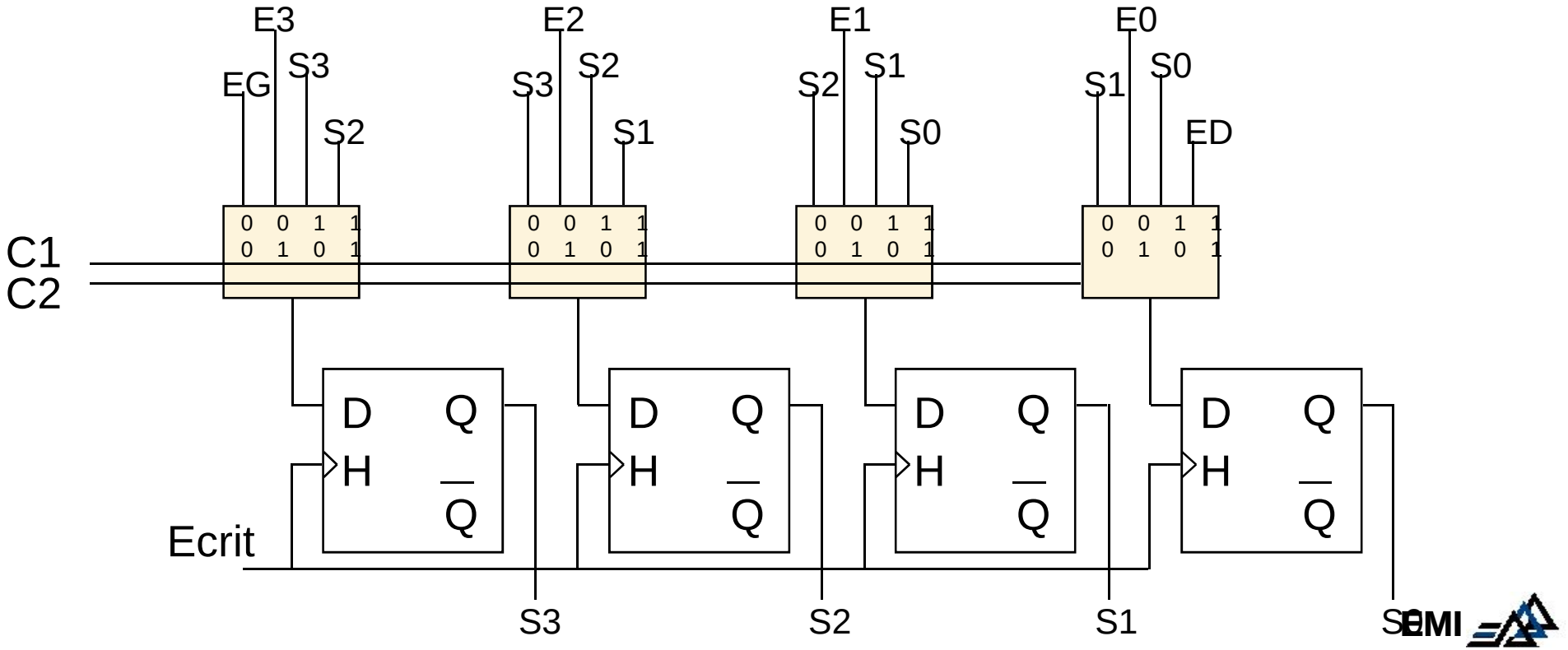


$$Di = C \cdot Ei + C' \cdot Si$$

Registre universel

C1C2 = 00 Chargement parallèle
C1C2 = 01 Décalage à droite
C1C2 = 10 Décalage à gauche
C1C2 = 11 Inhibition de l'horloge.

$$Di = C1'.C2'.ei + C1'.C2.Si-1 + C1.C2'.Si+1 + C1.C2.Si$$



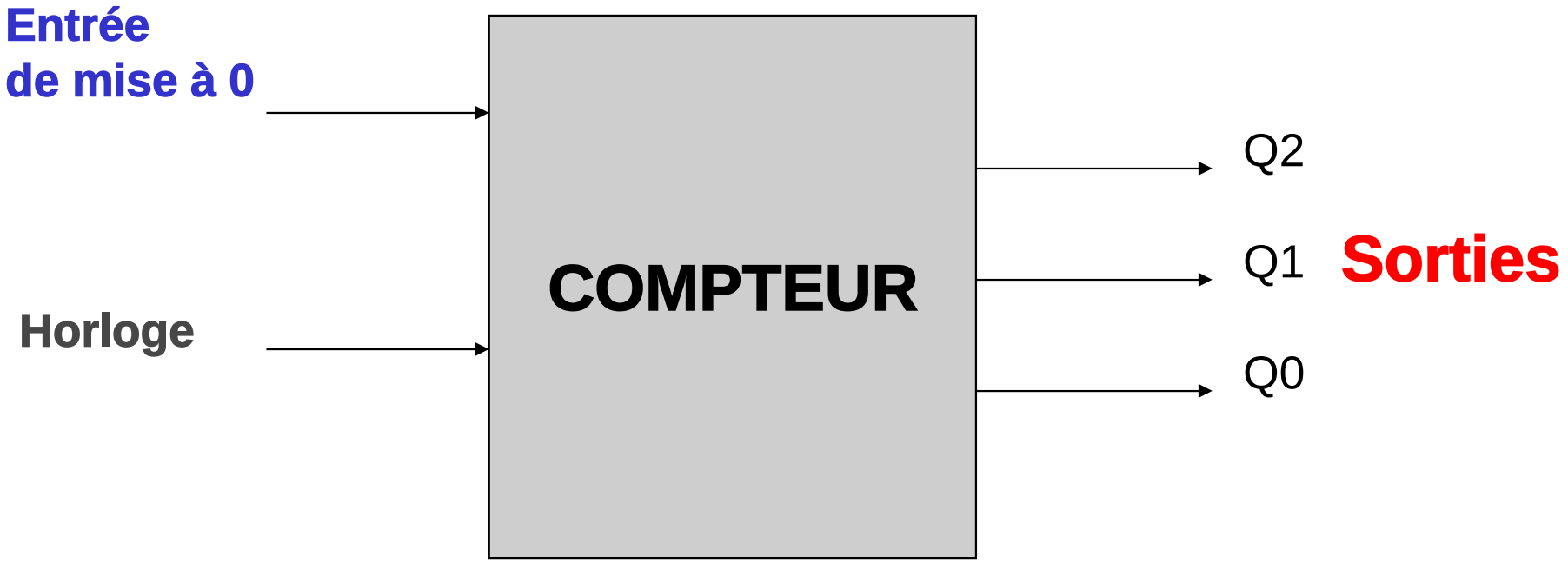
LES COMPTEURS

Les compteurs se présentent généralement sous la forme de circuits intégrés.
Ces derniers contiennent principalement des bascules.

Ils comptent, suivant le système de numération binaire, le nombre d'impulsions appliquées à son entrée.

Suivant qu'une nouvelle impulsion **incrémente** ou **décrémente** la valeur du mot binaire de sortie, le circuit fonctionne respectivement en **compteur** ou en **décompteur**.

Schéma d'un compteur 3 bits

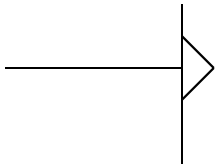
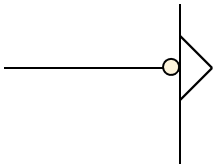


Description des entrées/sorties

Entrées :

- **Horloge (H, CLK, CP)**

Entrée permettant une évolution de la sortie.

Front montant actif :  Front descendant actif : 

- **Remise à zéro (Reset, CLR)**

Entrée permettant une mise à zéro des sorties.

Active sur niveau haut ou niveau bas.

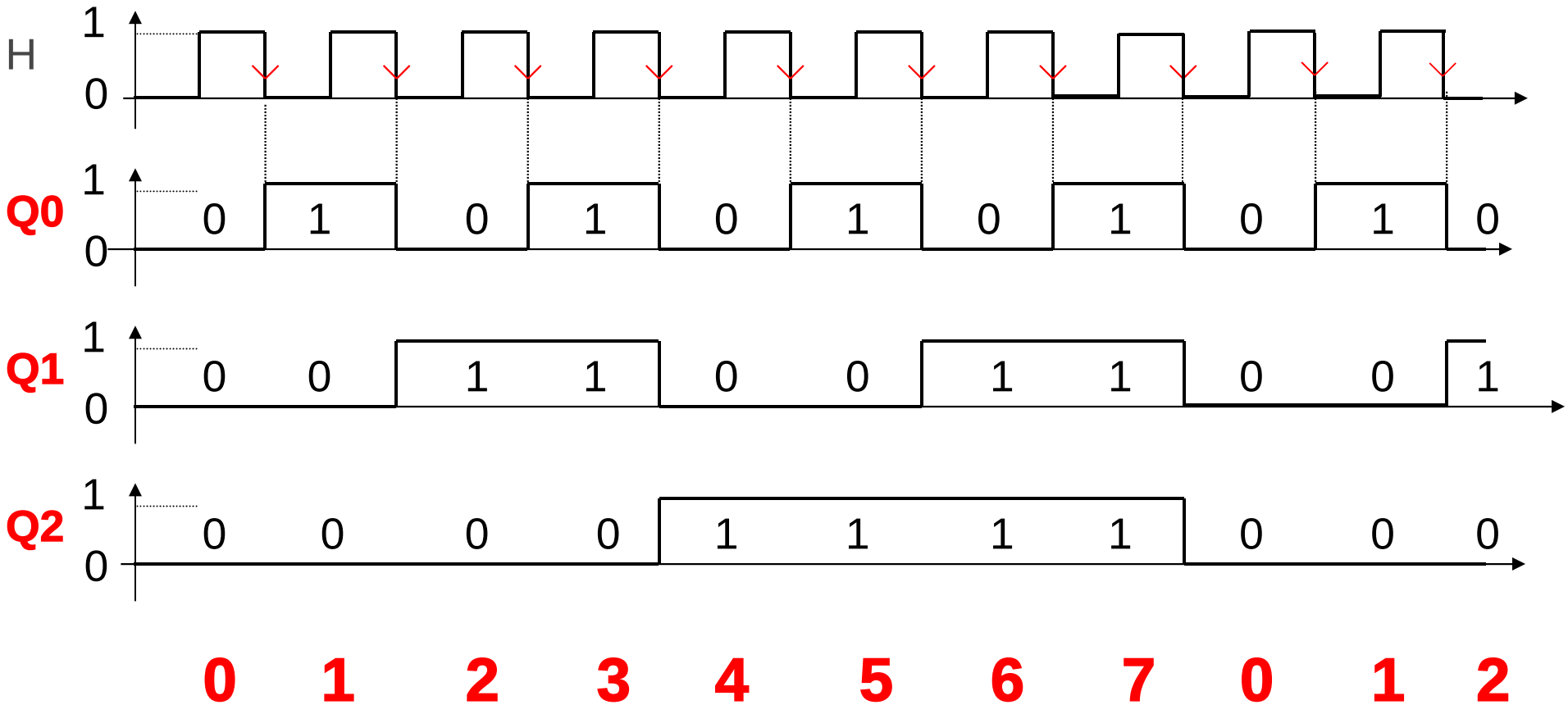
Sorties :

- **Q2, Q1, Q0**

Q2 : poids fort Q0 : poids faible

CHRONOGRAMMES

Horloge active sur front descendant



COMPTEUR 3 BITS

Le compteur précédent compte de **0 à 7**.
On dit que c'est un compteur **modulo 8.**

En observant les signaux on remarque que :

F0 = F/2 F : fréquence du signal H
F0 : fréquence du signal Q0

F1 = F/4 F1 : fréquence du signal Q1

F2 = F/8 F2 : fréquence du signal Q2

Un compteur peut servir de diviseur de fréquences

COMPTEUR SYNCHRONE
COMPTEUR ASYNCHRONE

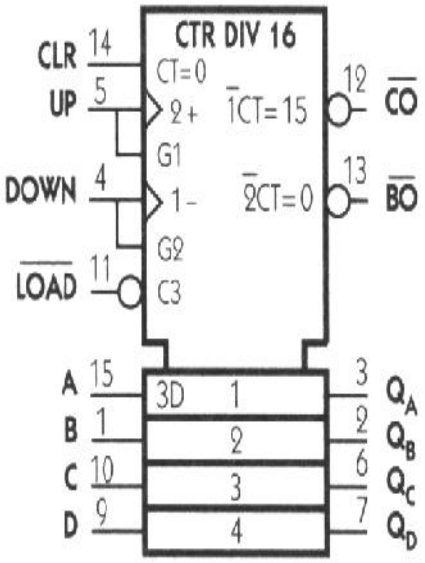
Dans la structure synchrone, l'horloge est la même pour tous les étages. Le basculement de toutes les sorties se fait en même temps.

Dans la structure asynchrone, l'impulsion de progression du compteur est appliquée sur l'entrée d'horloge du premier étage, les entrées d'horloge des autres bascules reçoivent le signal de sortie de l'étage précédent.

Exemple : le 74LS193

40193 **Compteur-décompteur binaire synchrone**, 4 bits *(compatible avec le circuit 193 de la série 74)*

Boîtier ***DIL 16***; V_{DD} *en 16*; V_{SS} *en 8*



CLR : remise à zéro
UP : entrée comptage
DOWN : entrée décomptage
$\overline{LOAD}$: validation de programmation
A, B, C et D : entrées de programmation
$\overline{CO}$: fin de cycle de comptage
$\overline{BO}$: fin de cycle de décomptage
Q_A, Q_B, Q_C et Q_D : sorties

Table des modes de fonctionnement

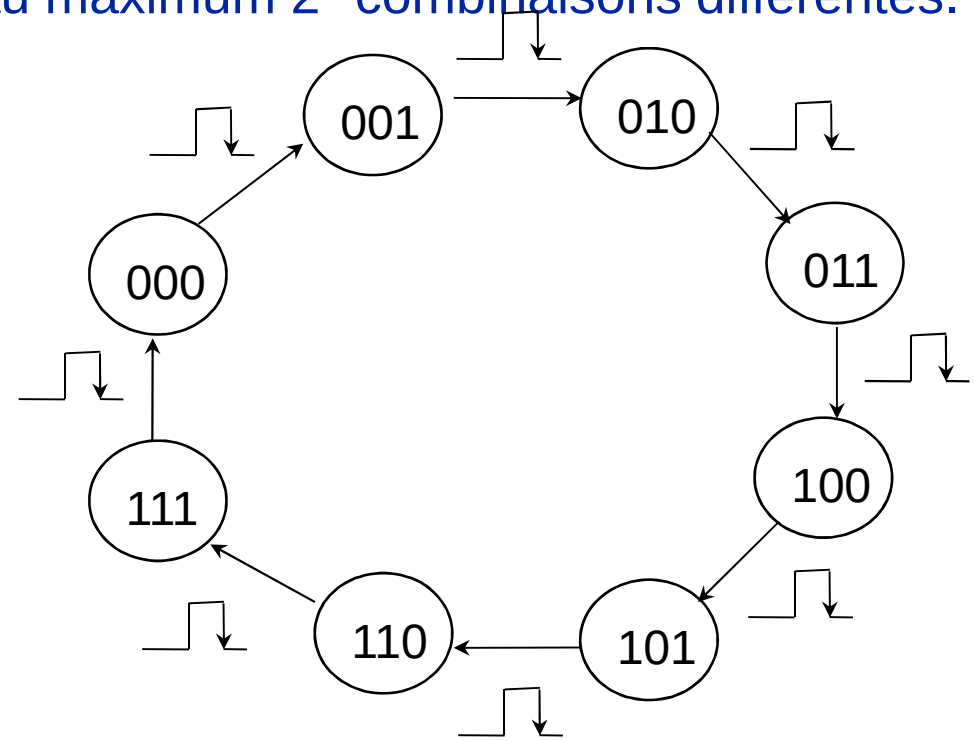
CLR	LOAD	UP	DOWN	MODE
L	L	X	X	Prépositionnement
L	H	↑	H	Comptage
L	H	H	↓	Décomptage
H	X	X	X	Toutes les sorties mises à 0

Les changements d'état se font sur le front montant du signal d'horloge.
La mise en cascade de plusieurs circuits revient à lier :

- $\overline{CO}$ de la première décade avec **UP** de la seconde; et $\overline{BO}$ de la première décade avec **DOWN** de la seconde.

Compteur

Un compteur est une association de n bascules permettant de décrire, au rythme d'une horloge, une séquence déterminée qui peut avoir au maximum 2^n combinaisons différentes.

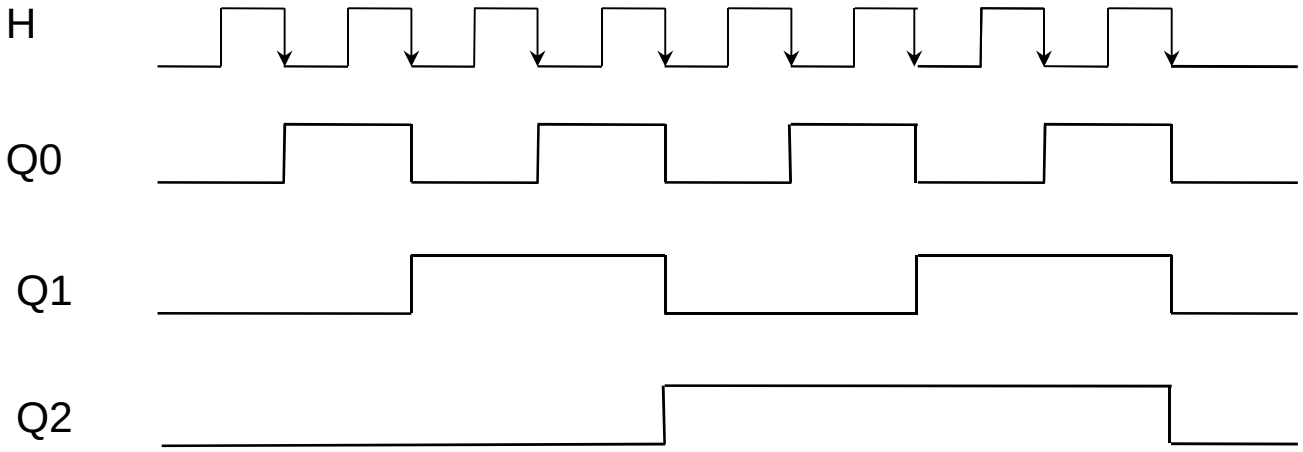


- **<u>Définition:</u> Une combinaison de sortie d'un compteur est appelée *état*. Le nombre d'états différents pour un compteur est appelé le *modulo ~* de ce compteur.**

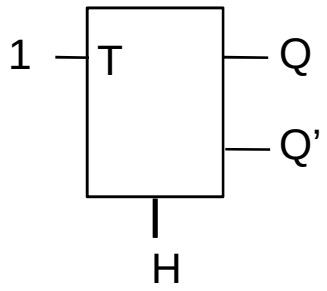
Compteur

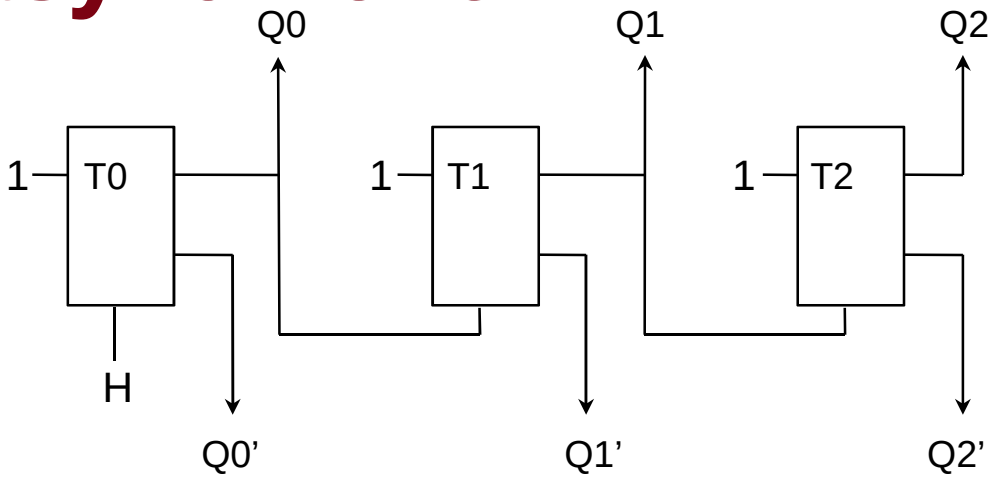
Un compteur est une association de n bascules permettant de décrire, au rythme d'une horloge, une séquence déterminée qui peut avoir au maximum 2^n combinaisons différentes.

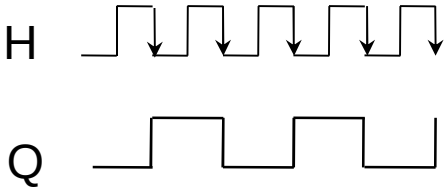
Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

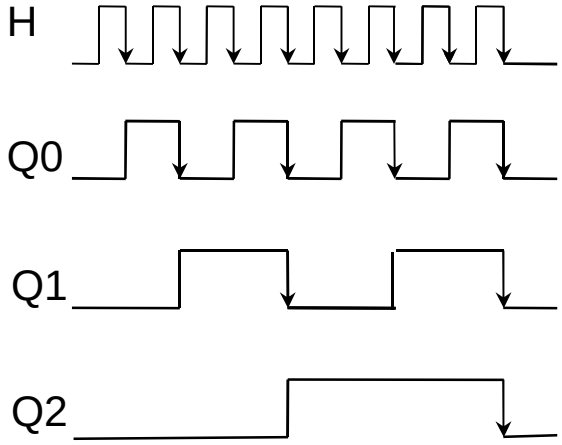


Compteur asynchrone





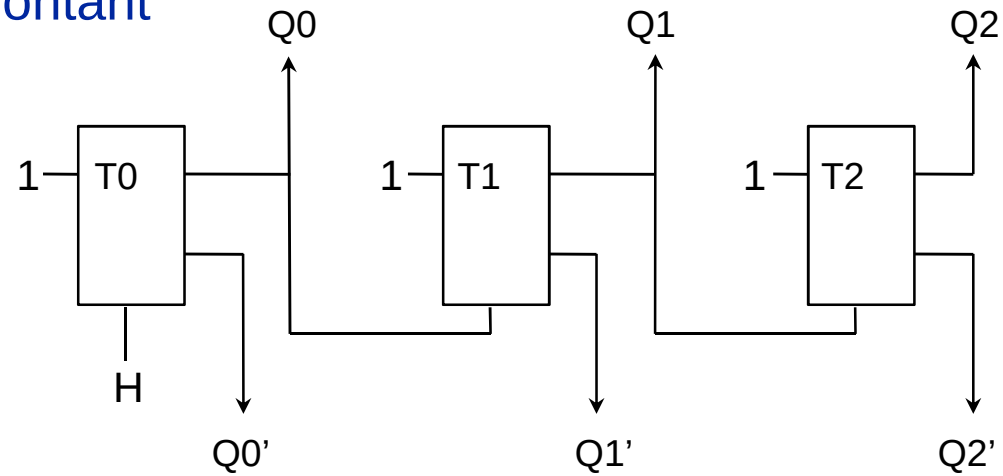


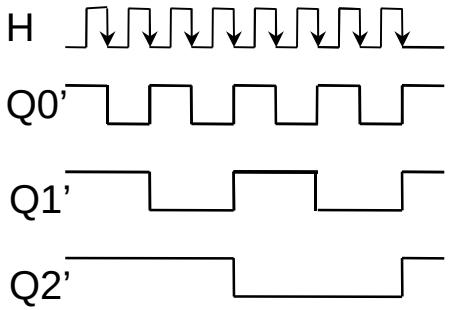


Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

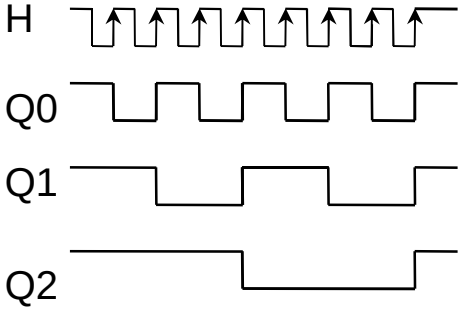
Décompteur asynchrone

Pour réaliser un décompteur il suffit de considérer sur les sorties Q' des bascules ou de réaliser le même montage avec des bascules fonctionnant sur front montant





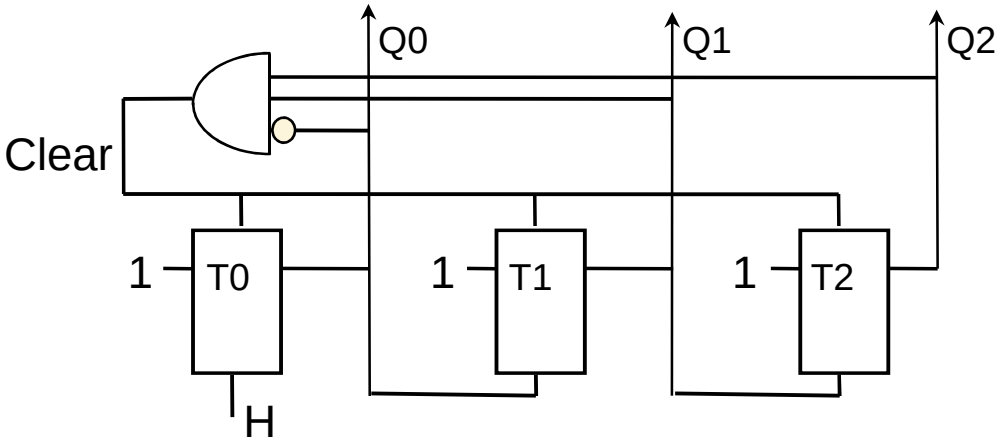
Q2'	Q1'	Q0
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1
0	0	0

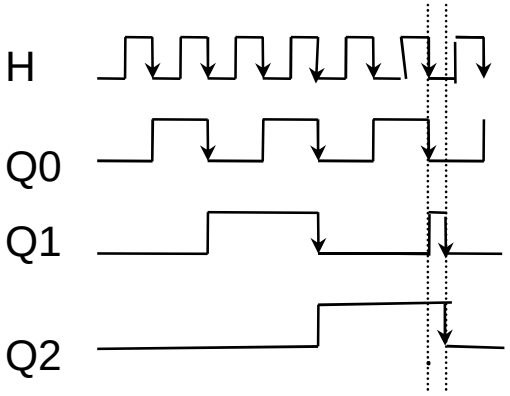


Q2	Q1	Q0
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1
0	0	0

Compteur asynchrone par 6

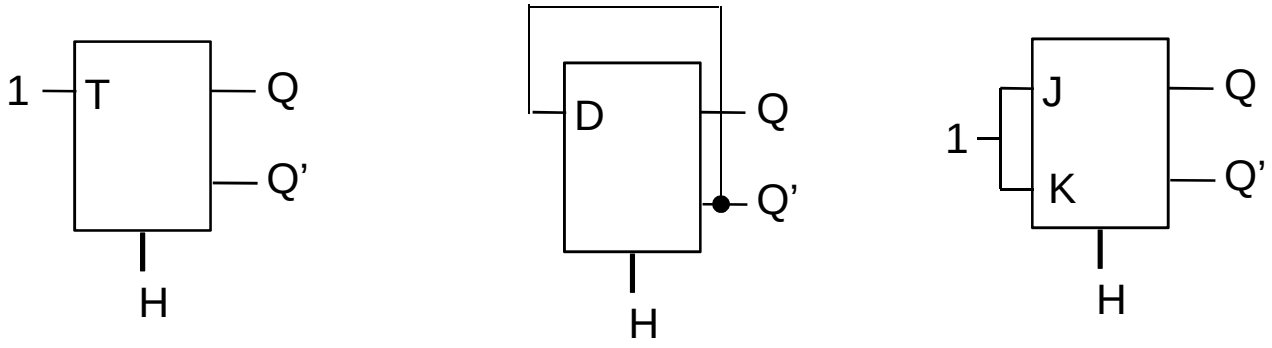
Pour réaliser un compteur ou un décompteur dont le cycle n'est pas une puissance de 2, la seule solution est d'agir sur l'entrée "Clear" lorsque la combinaison correspondant au modulo du compteur ce produit sur les sorties de celui ci.

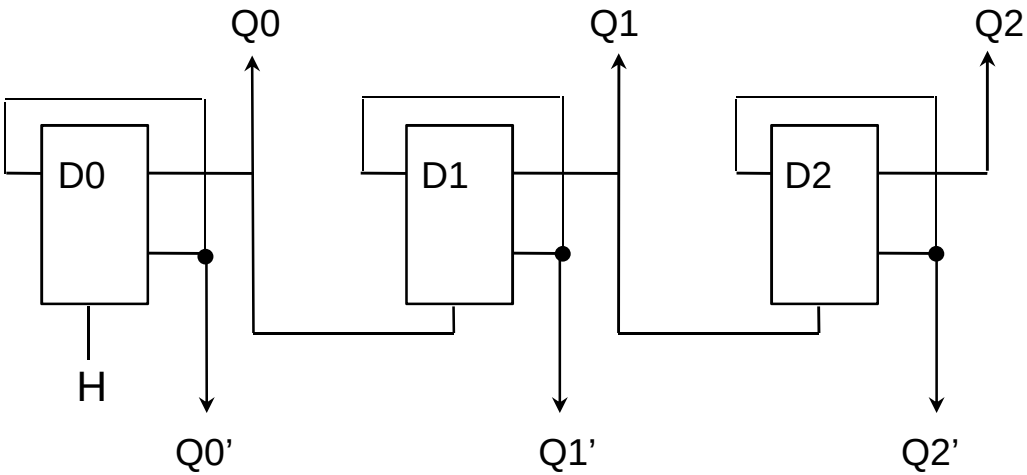




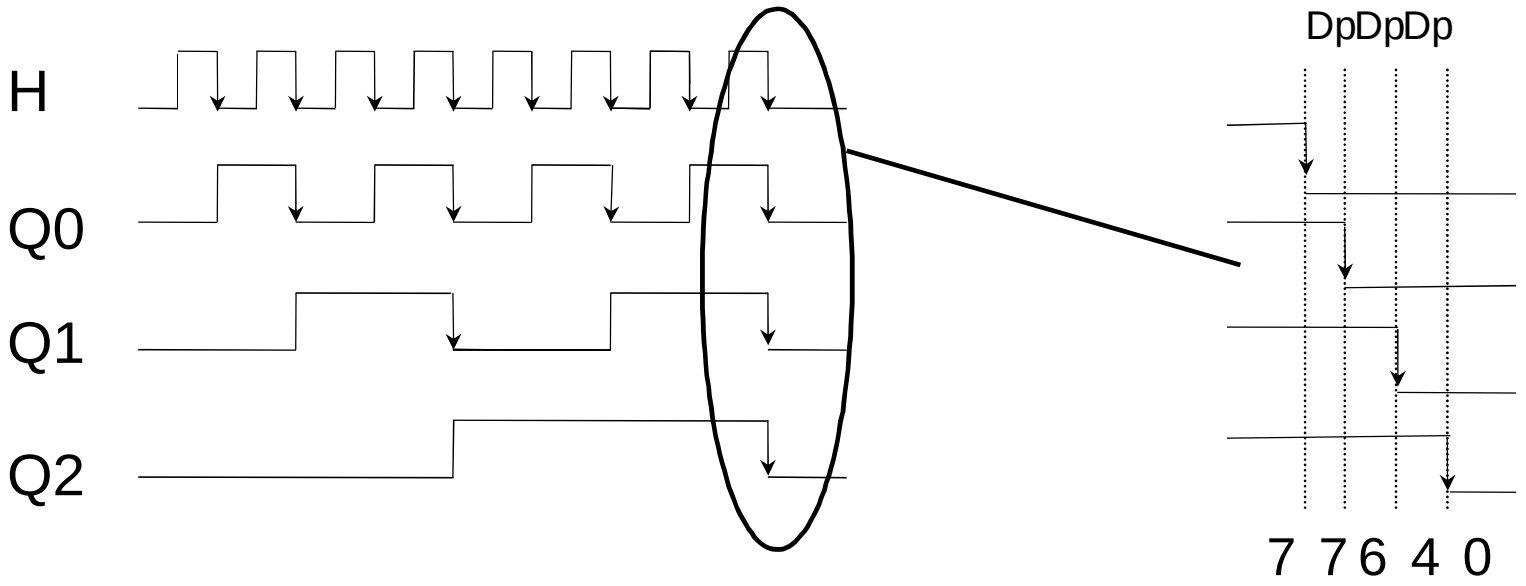
Q2	Q1	Q0	Clear
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1

Compteur asynchrone (D)





Inconvénients des compteurs asynchrones



Tm = Dp * n Délai de propagation du compteur

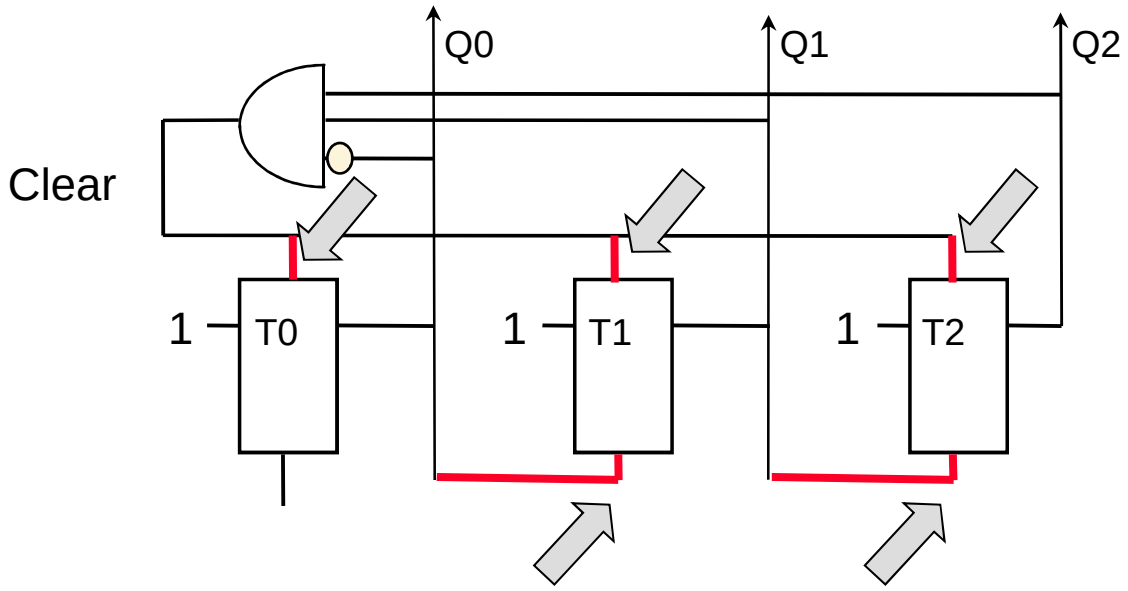
$T_H \geq Tm$ Période de l'horloge

$F_H \leq 1/(Tm) = 1 / (n*Dp)$ Fréquence de l'horloge

Inconvénients des compteurs asynchrones

Logique sur des signaux asynchrones (H, Clear)

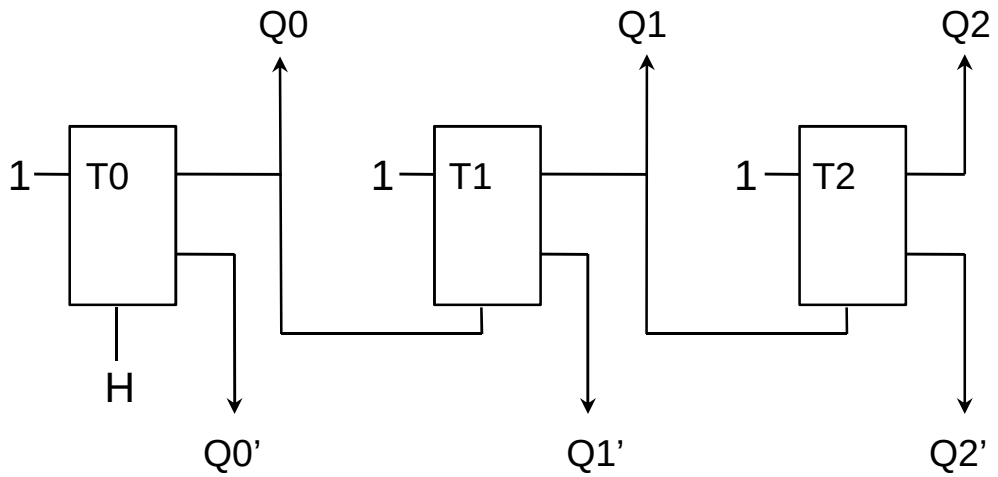
Risque de transitoires



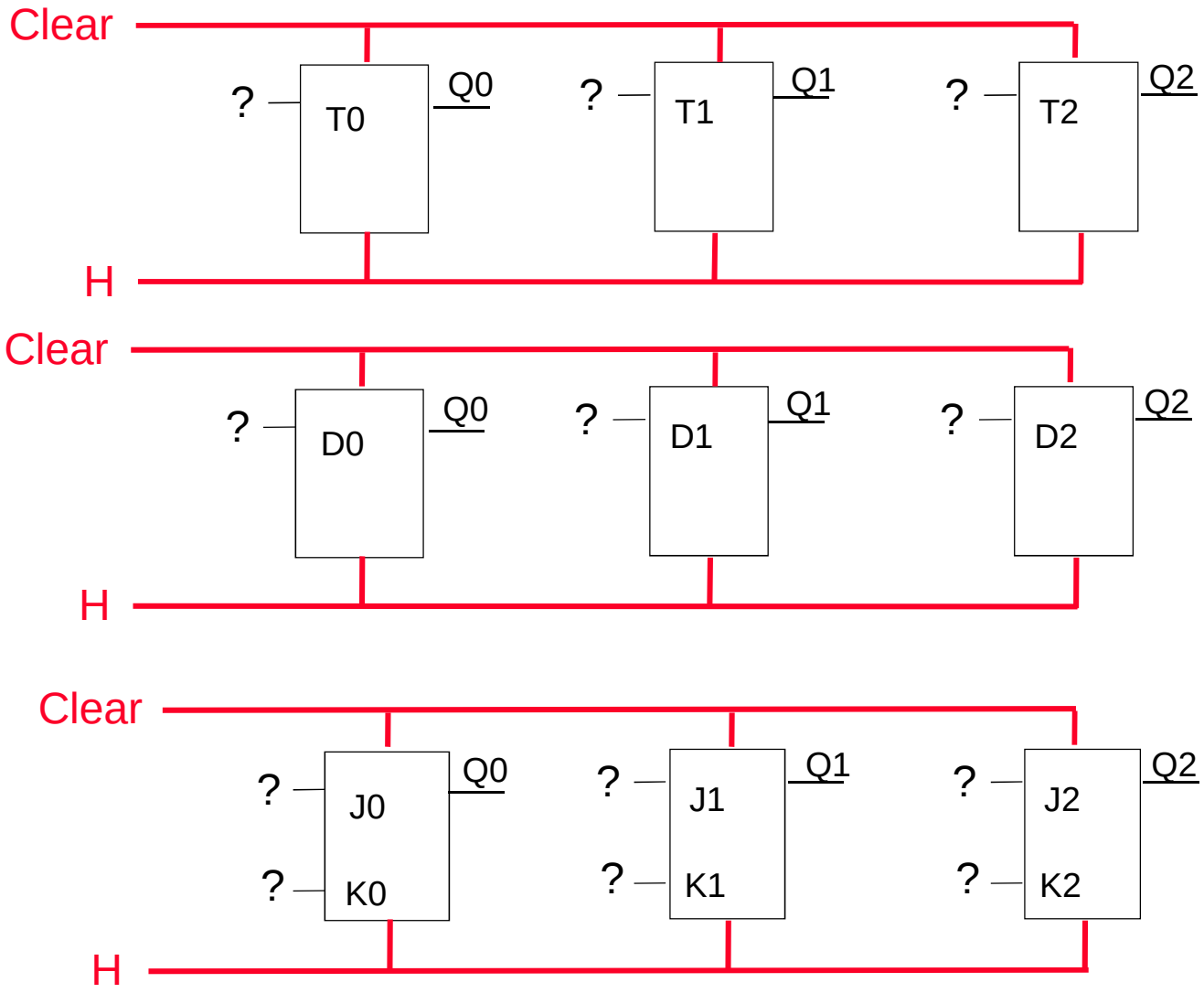
Inconvénients des compteurs asynchrones

Compteurs / Décompteur

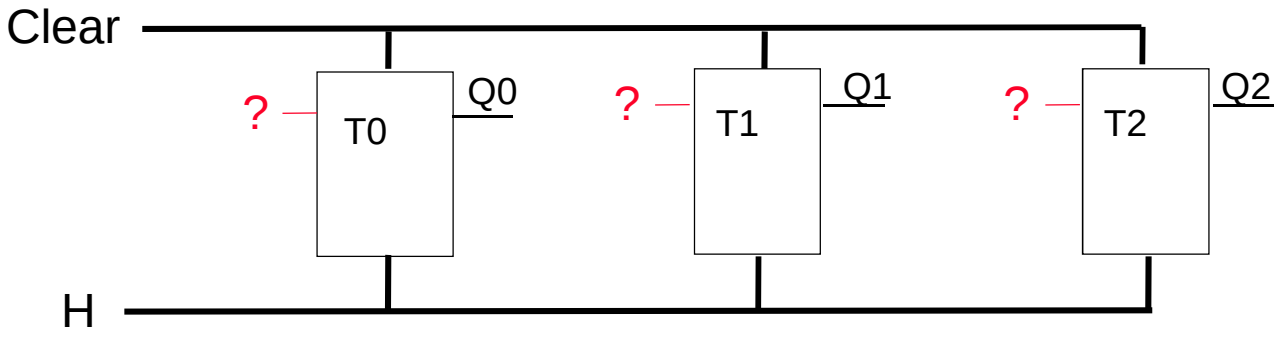
Modification de l'état



Compteur synchrone



Compteur synchrone

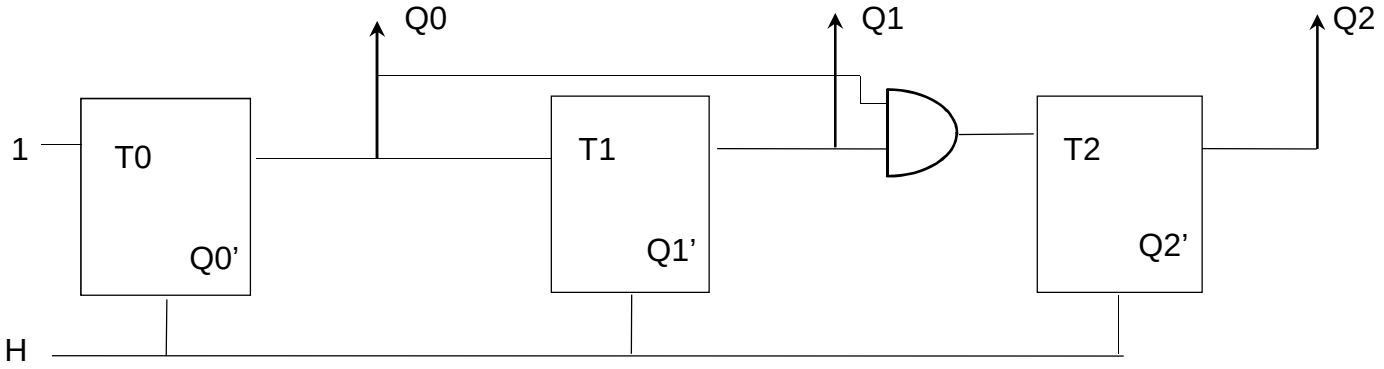


Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

T0 = 1
T1 = Q0
T2 = Q0.Q1

Tn = Q0.Q1....Qn-1

Compteur synchrone



Tm = Dp	Délai de propagation du compteur
$T_H \geq Tm$	Période de l'horloge
$F_H \leq 1/(Tm) = 1 / Dp$	Fréquence de l'horloge

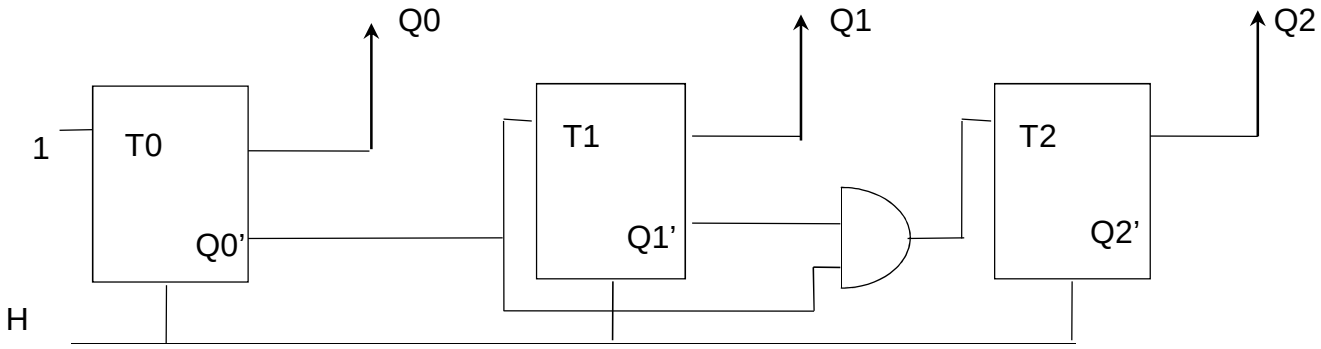
Décompteur synchrone

Un décompteur peut être obtenu en sortant sur les sortie Q' du compteur. On peut également réaliser un décompteur en remarquant sur la table de vérité que le bit de poids faible change à tous les coups d'horloge et qu'un bit quelconque change lorsque tous les bits de droite sont égaux à 0.

Q2	Q1	Q0
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1
0	0	0

T0 = 1
T1 = Q0'
T2 = Q0'.Q1'

Tn = Q0'.Q1'....Qn-1'



Compteur / décompteur synchrone

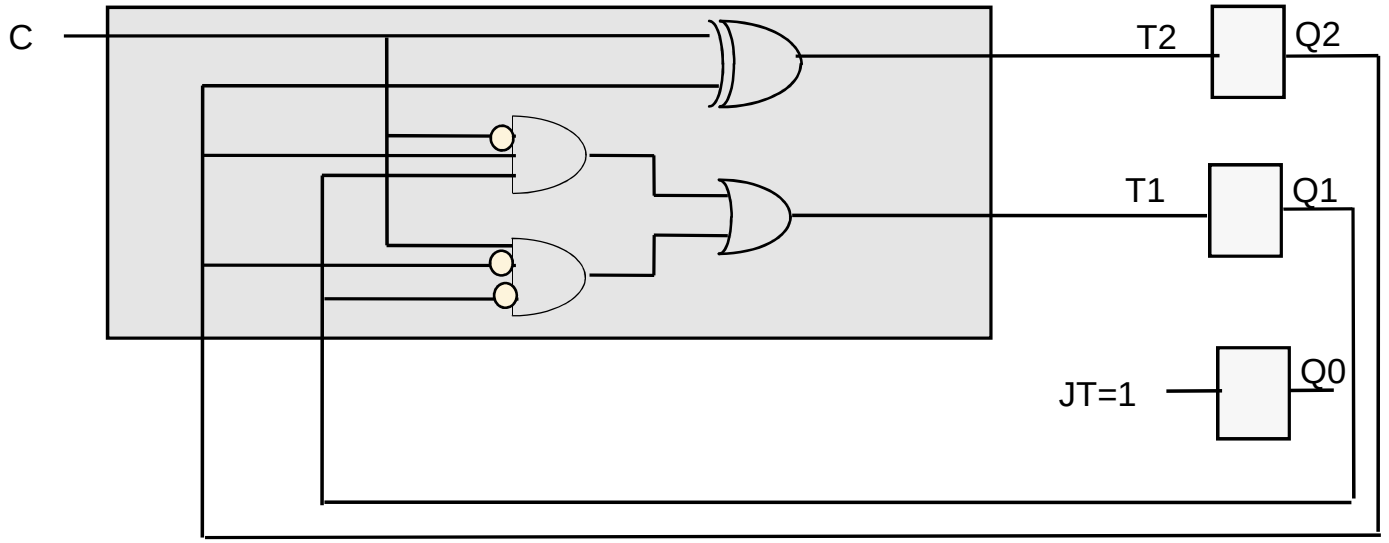
Par le même raisonnement, on peut déterminer la structure d'un compteur / décompteur synchrone dont le mode comptage ou décomptage est commandé par une commande C (C=0 => Comptage, C=1 => Décomptage).

$T0 = 1$

$T1 = C'.Q0 + C.Q0' = C \oplus Q0$

$T2 = C'.Q0.Q1 + C.Q0'.Q1'$

$Tn = C'.Q0.Q1...Qn-1 + C.Q0'.Q1'....Qn-1'$



Compteur synchrone par 6

Pour réaliser un compteur, un décompteur ou un compteur / décompteur dont le cycle n'est pas une puissance de 2, il faut recalculer les fonctions d'entrée des bascules.

Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1

Soit C5 un flag prévenant qu'on est sur la combinaison 5. C5 = Q2.Q1'.Q0

T0 = 1 (Même fonctionnement que C5 vaille 0 ou 1)

T1 = C5'.Q0 + C5.0 = C5'.Q0 (Conservation de la valeur de sortie lorsque C5=1)

T2 = C5'.Q0.Q1 + C5.1 = C5'.Q0.Q1 + C5 (Inv. de la valeur de sortie lorsque C5=1)

Compteur / Décompteur par 6 avec Inhibition

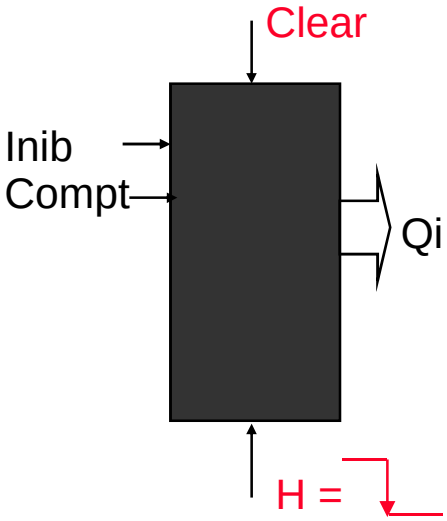
Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1

Inib : Signal d ’inhibition du compteur (actif sur niveau 1)
Compt : Signal de comptage (1) / Décomptage (0)
C0 : Détection de la combinaison 0
C5 : Détection de la combinaison 5

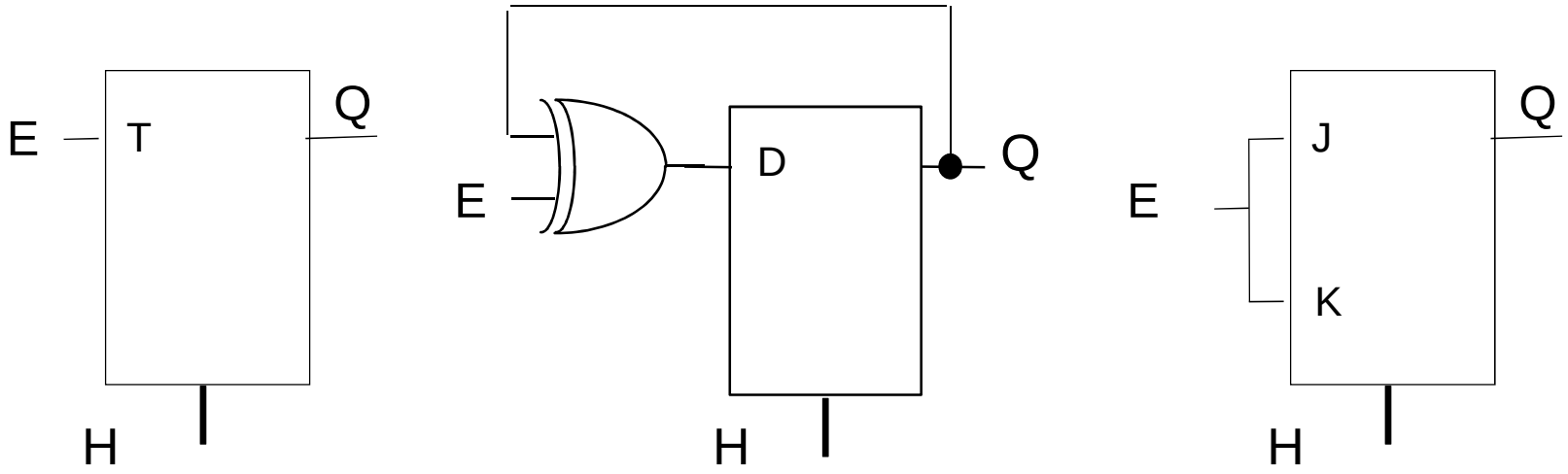
T0 = Inib’

T1 = Inib’[Compt{C5’.Q0)} + Compt’{C0’.Q0’}]

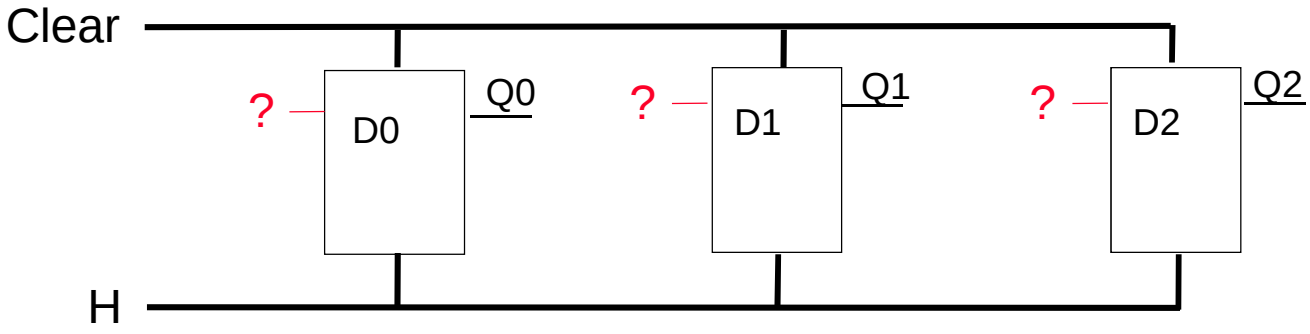
T2 = Inib’[Compt{C5’.Q0.Q1 + C5} + Compt’{C0’.Q0’.Q1’ + C0]



Bascules T, D et JK



Compteur synchrone (D)



Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

D	Q(n+1)
0	0
1	1

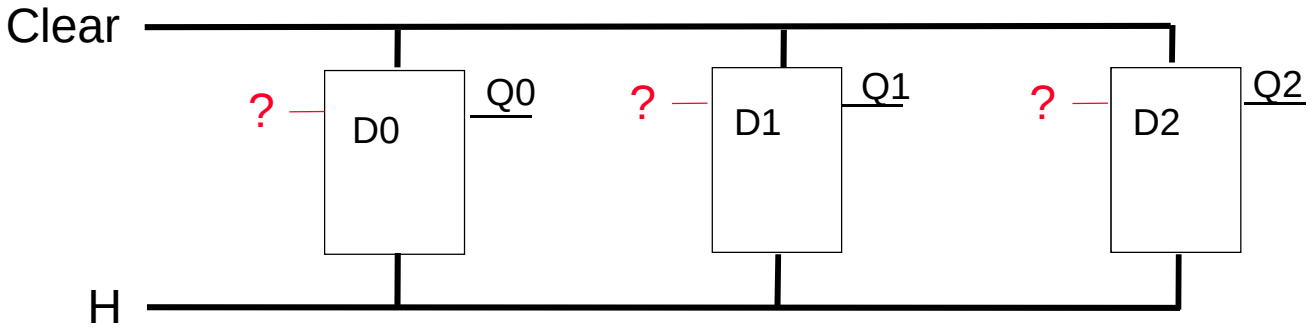
D0 = Q0’

D1 = Q0.Q1’ + Q0’.Q1 = Q0 ⊕ Q1

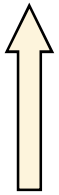
D2 = Q0.Q1. Q2’ + (Q0.Q1)’.Q2 = Q0.Q1 ⊕ Q2

Dn = ...

Décompteur synchrone (D)



Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



D	Q(n+1)
0	0
1	1

D0 = Q0’

D1 = Q0.Q1 + Q0’.Q1’ = (Q0 ⊕ Q1)’

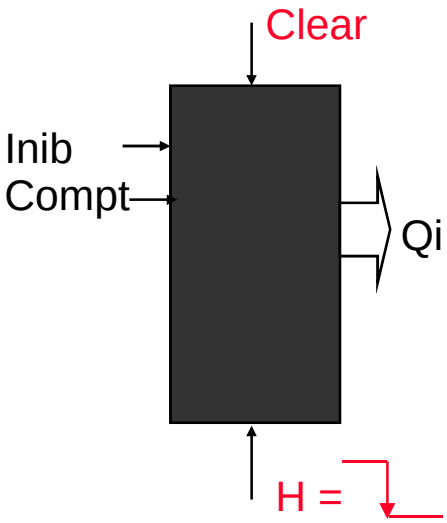
D2 = (Q0+Q1). Q2 + (Q0+Q1)’.Q2’ = (Q0+Q1) ⊕ Q2’

Dn = …

Compteur / Décompteur par 6 avec Inhibition

Inib : Signal d ’inhibition du compteur (actif sur niveau 1)
Compt : Signal de comptage (1) / Décomptage (0)
C0 : Détection de la combinaison 0
C5 : Détection de la combinaison 5

Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1

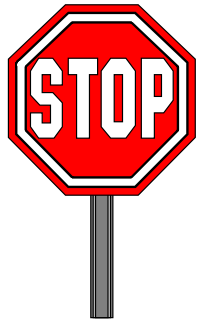


D0 = [Inib’[Compt{C5’.Q0’ + C5.Q0’} +
Compt’{C0’.Q0’ + C0.Q0’}] + Inib. Q0]
= [Inib’[Q0’] + Inib. Q0]
= Inib ⊕ Q0’

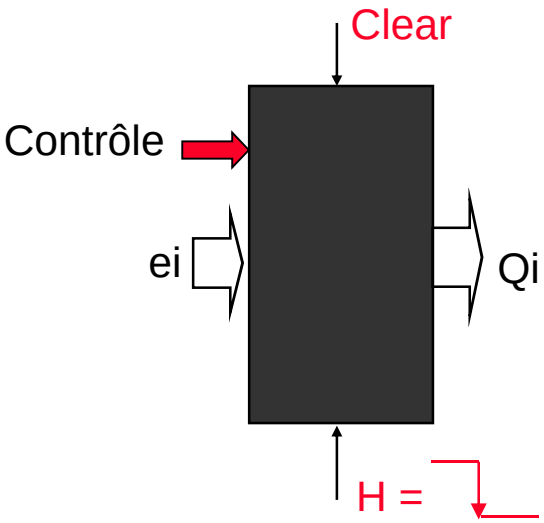
D1 = [Inib’[Compt{C5’.(Q0.Q1’+Q0’.Q1)+C5.0} +
Compt’{C0’.(Q0’.Q1’+Q0.Q1) + C0.0}] + Inib. Q0]
= [Inib’[Compt{C5’.(Q0 ⊕ Q1)} +
Compt’{C0’.(Q0 ⊕ Q1’) }] + Inib. Q0]

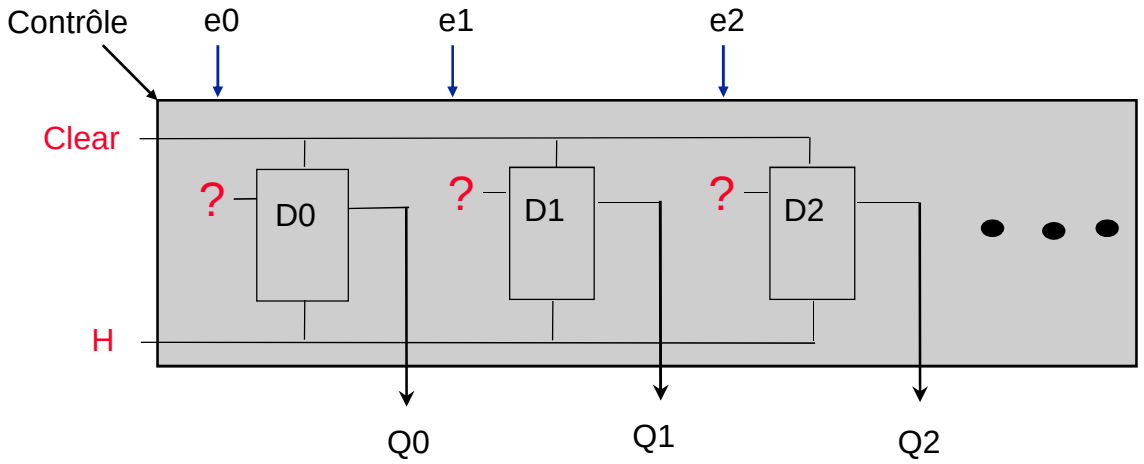
D2 = [Inib’[Compt{C5’.(Q0.Q1 ⊕ Q2)} +
Compt’{C0’.(Q0+Q1 ⊕ Q2’) }] + Inib. Q0]

Règles de conception (Registres, Compteurs, …)

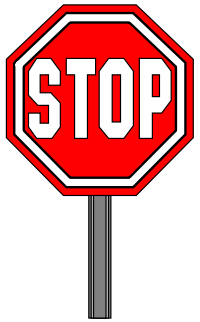


Pas de logique sur les signaux
- d 'horloge (H)
- de forçage (Clear, Preset)



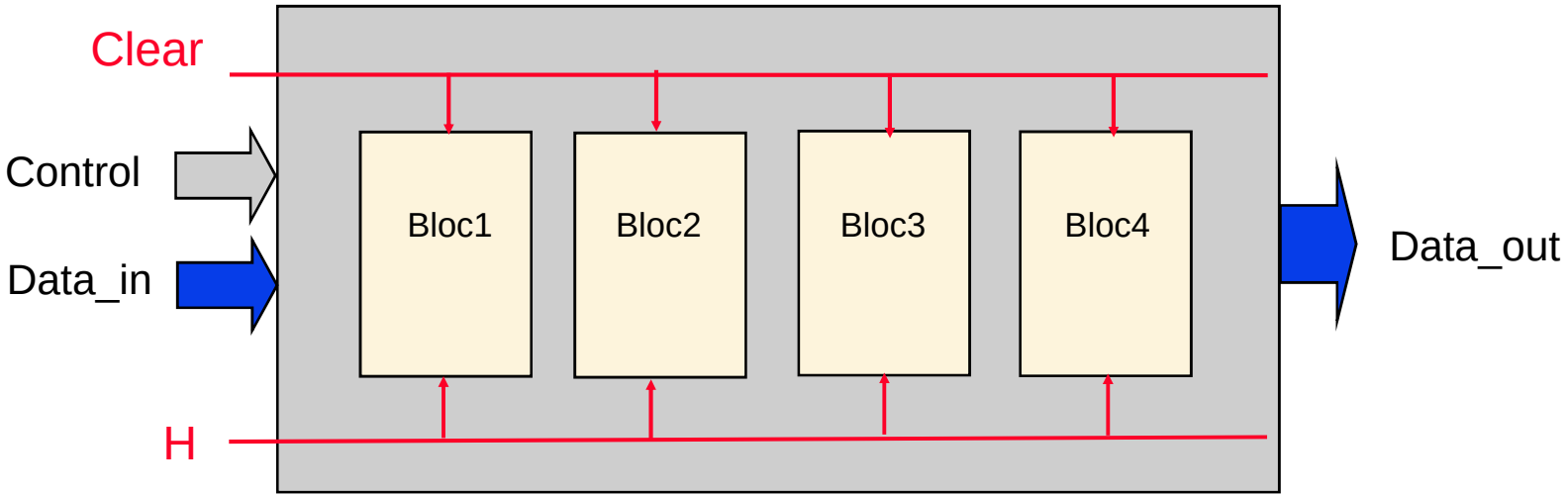


Règles de conception (Circuit)



Pas de logique sur les signaux

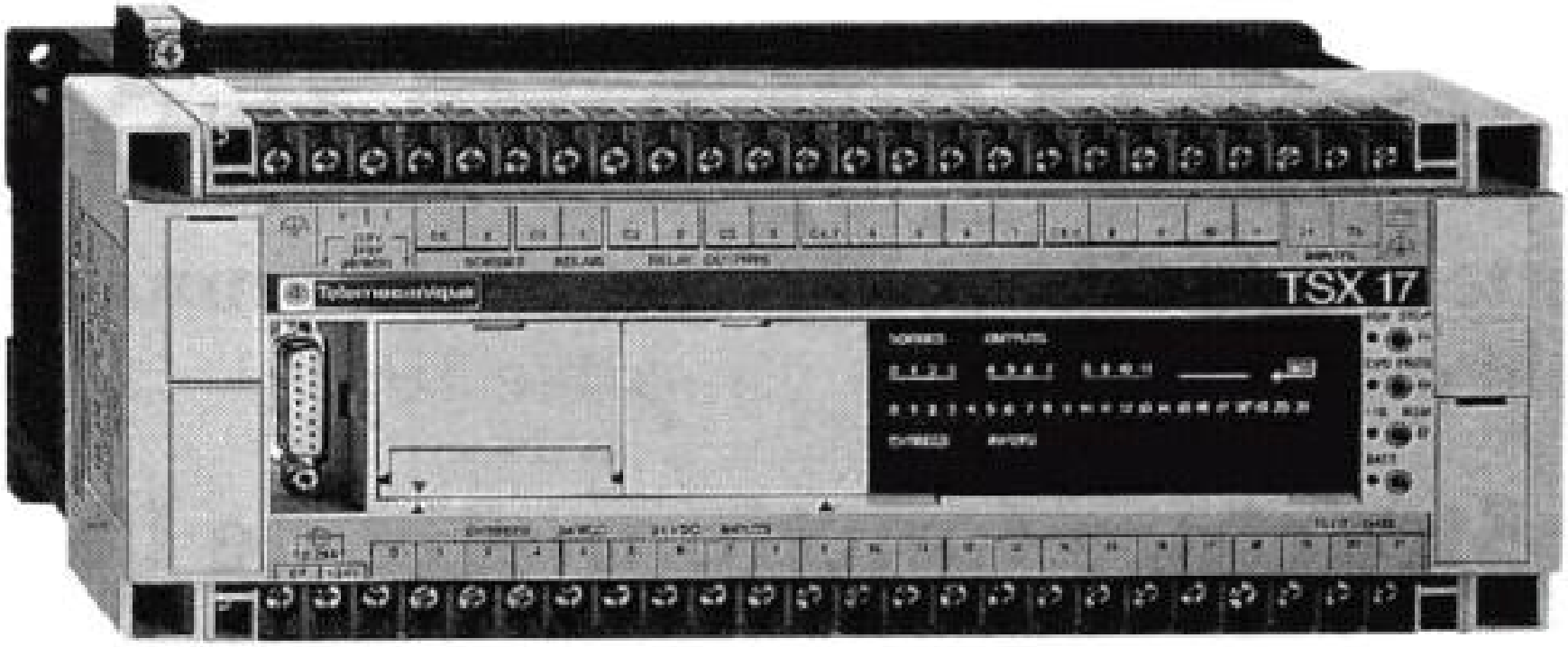
- d 'horloge (H)
- de forçage (Clear, Preset)



4 : Vulgarisation de l'automate programmable industriel

- **L'API est une machine électronique programmable par un personnel non informaticien et destinée à piloter en ambiance industrielle et en temps réel des procédés ou parties opératives.**
- **Un automate programmable industriel est adaptable à un maximum d'applications, d'un point de vue traitement, composants, langages.**
- **C'est pour cela qu'il est, souvent, de construction modulaire**

Automate monobloc TSX17



Fonctionnement d'un API

Le traitement est effectué en quatre phases :