

CIRCUITS LOGIQUES

1 - LOGIQUE COMBINATOIRE (Rappels 1^{ère})

1.1 Variable booléenne

On appelle variable **Booléenne** simple toute quantité susceptible de ne prendre que **2 valeurs**.
Pratiquement, on attribue les valeurs numériques 0 ou 1 à cette variable (*système binaire*)

1.2 Fonctions logiques de base (tables de vérité)

fonction NON (NOT) ou complément

a	\bar{a}
0	1
1	0

fonction ET (AND) ou produit logique

a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

fonction OU (OR) ou somme logique

a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

fonction OU exclusif (XOR)

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

1.2 Autres fonctions logiques

fonction ET-NON (NAND)

a	b	$\overline{a.b}$
0	0	1
0	1	1
1	0	1
1	1	0

fonction OU-NON (NOR)

a	b	$\overline{a+b}$
0	0	1
0	1	0
1	0	0
1	1	0

Théorème de DE MORGAN

$\overline{a+b} = \overline{a}.\overline{b}$ et $\overline{a.b} = \overline{a} + \overline{b}$
--

2 - CIRCUITS ELECTRONIQUES LOGIQUES

2.1 Familles logiques

Pour réaliser les fonctions logiques précédentes on utilise des circuits intégrés : **PORTES**. On distingue deux grandes familles de circuits logiques :

- Circuits **TTL** "Transistor Transistor Logic"
- Circuits **CMOS** "Complémentary Metal-Oxyde S.C"

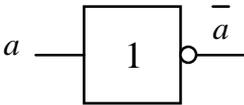
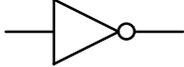
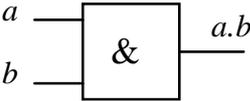
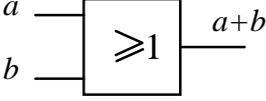
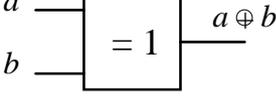
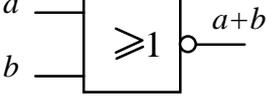
Quelques propriétés (voir TP)

- | | | |
|-------------------------|-------------------|-----------------------------------|
| • alimentation : | TTL $V_{cc} = 5V$ | CMOS $V_{DD} = 3 \text{ à } 15 V$ |
| • niveau bas d'entrée | TTL 0 à 0,8 V | CMOS 0 à 40% V_{DD} |
| • niveau haut d'entrée | TTL 2 à 5 V | CMOS 60% à 100% V_{DD} |
| • niveau bas de sortie | TTL 0 à 0,4 V | CMOS 0 à 10% V_{DD} |
| • niveau haut de sortie | TTL 2,4 à 5 V | CMOS 90% à 100% V_{DD} |

Les circuits de technologie TTL sont plus "rapides" que les circuits CMOS, par contre les circuits CMOS ont une "consommation" beaucoup plus faibles que les circuits TTL.

2.2 Portes logiques : symboles

Les portes logiques sont des circuits de **logique combinatoire**. Le signal de sortie ne dépend, à un instant donné, **que de la combinaison des entrées**.

NON		
ET		
OU		
OU exclusif		
ET-NON		
OU-NON-		

3- LOGIQUE SEQUENTIELLE

3.1 Fonction mémoire

Lorsque l'on réalise une chaîne logique, il est souvent nécessaire de conserver en *mémoire* des informations. Dans une structure *séquentielle*, pour une combinaison donnée des variables d'entrée, il peut y avoir *plusieurs situations des sorties* (Cf interrupteur "va et vient").

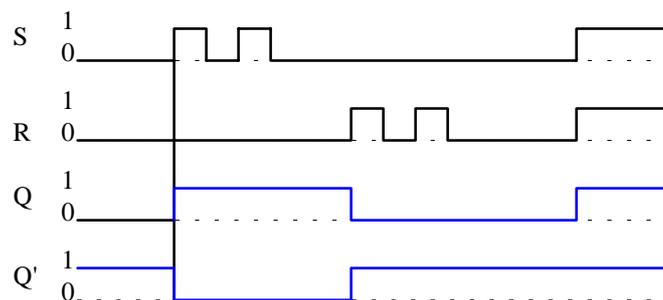
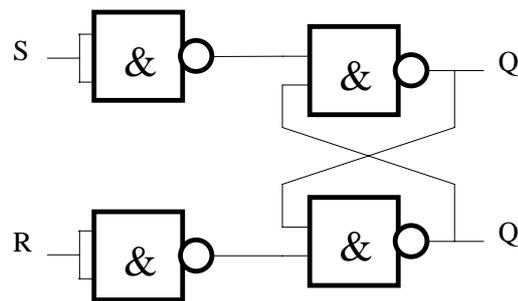
Dans un circuit de *logique séquentielle*, le signal de sortie dépend, à un instant donné, *de la combinaison des entrées ET de son état antérieur* (c'est-à-dire de la succession ou séquence des données).

Il faut donc une variable supplémentaire pour "retourner" l'information de sortie sur une entrée : fonction *MEMOIRE*.

3.2 Bascules (voir TP)

Le circuit de base de la logique séquentielle est une *BASCULE* caractérisée par une table de transition.

Bascule R S

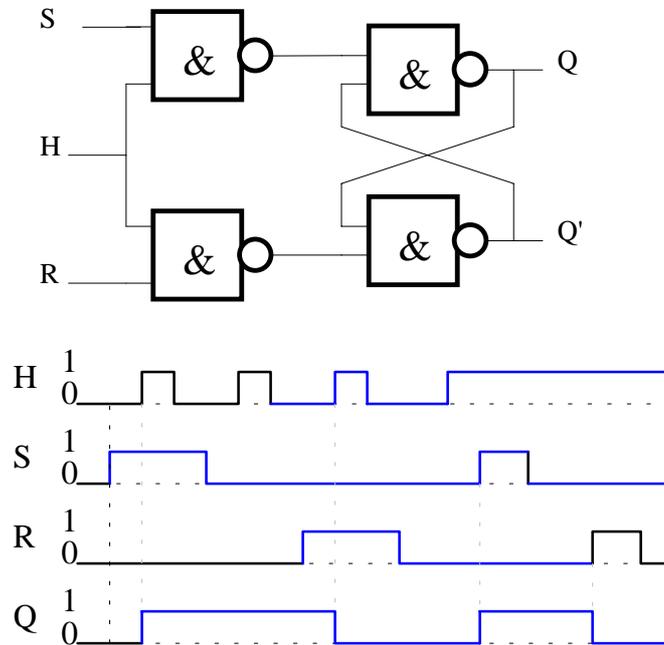


- $R = S = 0$ La bascule est au **repos**
- $S = 1 ; R = 0$ mise à 1 de Q S : **SET**
- $S = 0 ; R = 1$ mise à 0 de Q R : **RESET**
- $R = S = 1$ état "**interdit**"; le retour au repos n'est pas prévisible.

Une application intéressante de cette bascule est le circuit "*anti-rebonds*"

Bascule R S H

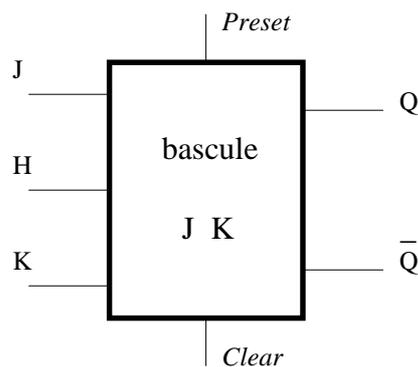
On améliore le fonctionnement de la bascule R-S avec une entrée supplémentaire appelée **HORLOGE** permettant de contrôler les instants de commutation des sorties (bascule *synchrone*).



- $H = 0$ la bascule est **bloquée**, les entrées R et S sont inactives
- $H = 1$ la bascule est **transparente**, elle fonctionne comme la bascule R-S

Remarques : La mise à 0 (ou à 1) peut donc être "*instantanée*" lorsque $H = 1$ ou "*différée*" lorsque $H = 0$. D'autre part l'état interdit ($Q = Q' = 1$) reste possible.

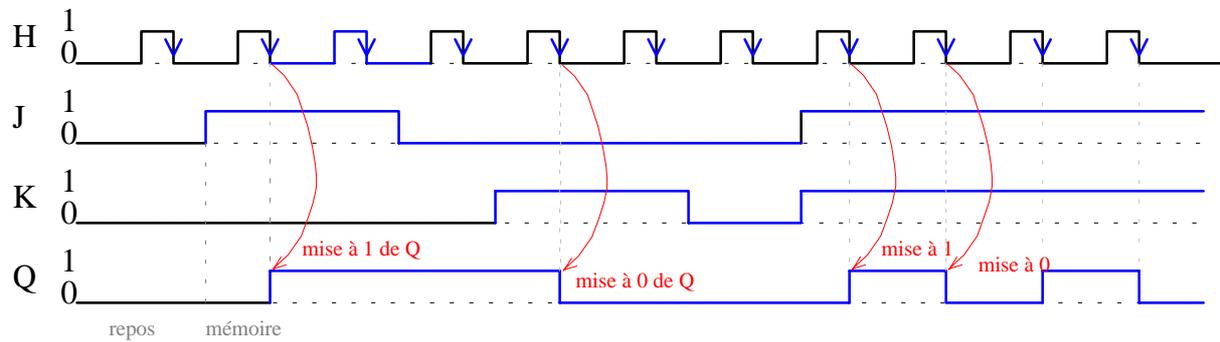
Bascule J K



Cette bascule très souple d'emploi est la plus utilisée :

- L'horloge commande les instants de commutation des sortie, en général, sur un "**front descendant**".
- Les sorties sont toujours complémentaires ($Q' = \bar{Q}$), il n'y a plus d'état interdit.
- Deux entrées supplémentaires permettent d'initialiser la bascule : $Q = 0$ avec l'entrée "**CLEAR**" ou $Q = 1$ avec l'entrée "**PRESET**".

Chronogramme et table de transition



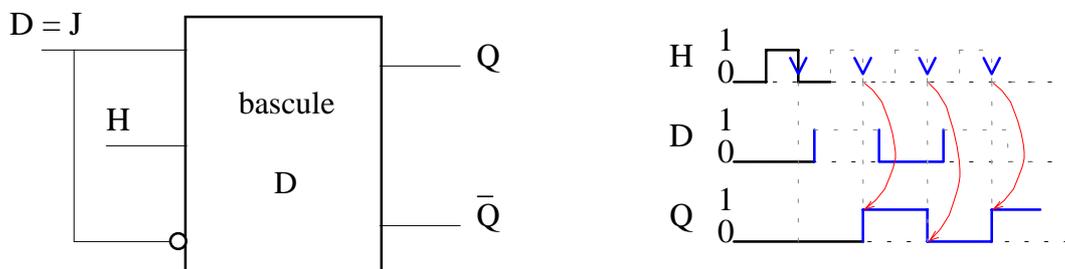
J	K	Q_n	Q_{n+1}	sur front descendant d'horloge
0	0	Q_n	Q_n	repos (mémoire)
0	1	Q_n	0	mise à zéro de Q (Reset)
1	0	Q_n	1	mise à 1 de Q (set)
1	1	Q_n	\bar{Q}_n	changement d'état à chaque front d'horloge

Remarques :

En logique TTL les entrées non connectées sont *au niveau haut* (1) et les entrées Clear et Preset sont actives à 0. Lorsque l'une des ces deux entrées est au niveau bas (0) la bascule est bloquée ($Q = 0$ ou 1 suivant l'entrée utilisée).

Bascule D

Cas particulier de la bascule J K, elle correspond à la combinaison $D = J = \bar{K}$.

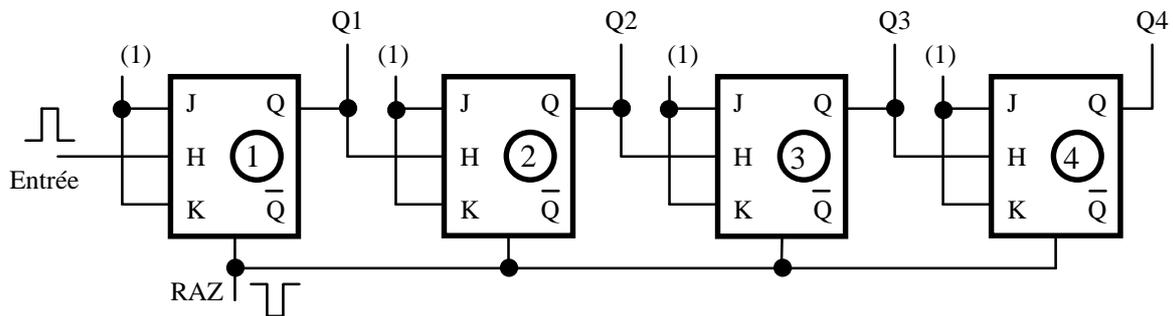


Bascule T

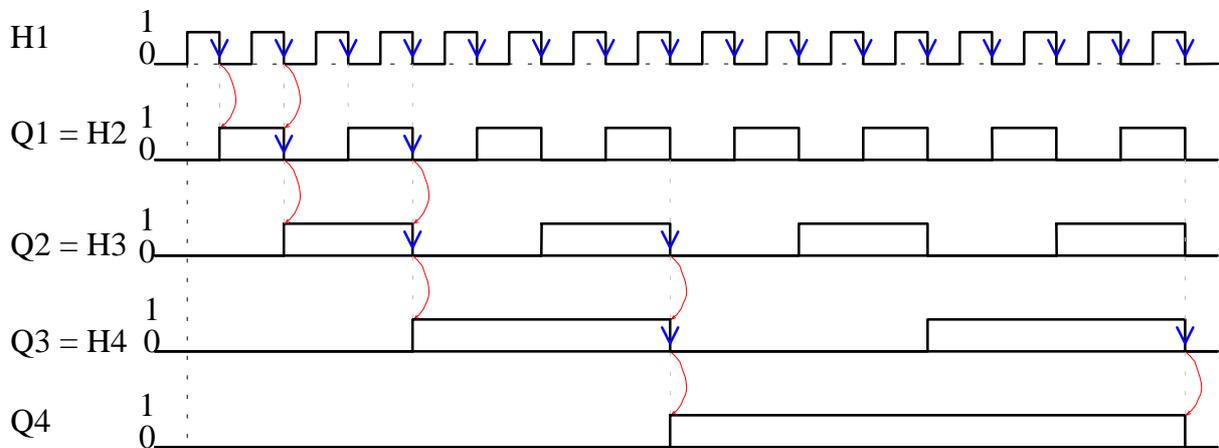
Autre cas particulier de la bascule J K ($J = K = 1$), elle est utilisée en *diviseur de fréquence par 2*.

4- APPLICATION : Compteur binaire asynchrone

Le compteur suivant est réalisé avec 4 bascules JKT ($J = K = 1$)



Chronogramme



impulsions d'horloge	Q4	Q3	Q2	Q1
	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

Remarques :

- Chaque bascule divise la fréquence d'horloge par deux
- Le compteur est recyclé en 16 impulsions d'horloge : *compteur hexadécimal*.

Compteur décimal

Au dixième coup d'horloge, on observe la combinaison suivante :

<i>impulsions d'horloge</i>	Q4	Q3	Q2	Q1
	2^3	2^2	2^1	2^0
n = 10	1	0	1	0

C'est la *première fois* que Q2 et Q4 sont à 1 *ensembles*. En connectant Q2 et Q4 à l'entrée d'une porte Et-Non dont la sortie est reliée à RAZ (Clear), le compteur sera recyclé à la dixième transition d'horloge ($\overline{1.1} = 0$), les entrées "Clear" de chaque bascule étant à 0.



N.B. Le retour à 0 des sorties Q2 et Q4 remet les entrées Clear à 1 ($\overline{0.0} = 1$), le compteur peut donc à nouveau fonctionner.