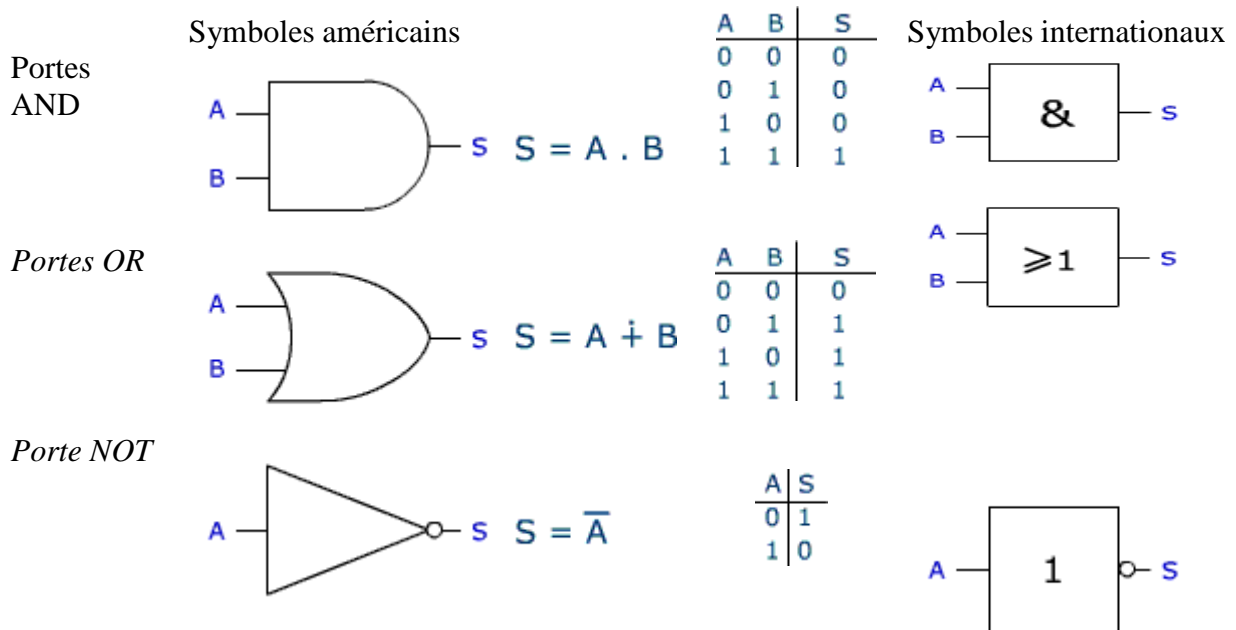


LA LOGIQUE COMBINATOIRE

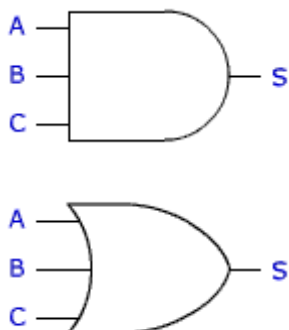
1- RAPPEIS SUR LES PORTES LOGIQUES

Nous avons jusqu'ici utilisé des boutons poussoirs et une lampe pour illustrer le fonctionnement des opérateurs logiques. En électronique digitale, les opérations logiques sont effectuées par des portes logiques. Ce sont des circuits qui combinent les signaux logiques présentés à leurs entrées sous forme de tensions. On aura par exemple 5V pour représenter l'état logique 1 et 0V pour représenter l'état 0.

1-1- Fonctions de base.



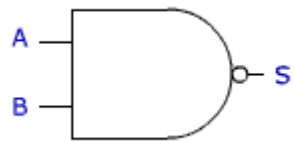
Le nombre d'entrées des fonctions AND et OR n'est pas limité. Voici par exemple une représentation de ces portes avec trois entrées :



1-2- Combinaisons de portes logiques.

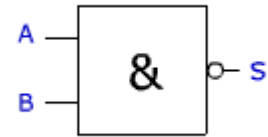
Ces trois fonctions logiques de base peuvent être combinées pour réaliser des opérations plus élaborées en interconnectant les entrées et les sorties des portes logiques.

La porte NAND (Non-ET)



$$S = \overline{A \cdot B}$$

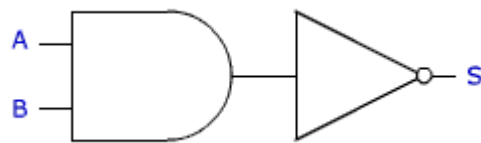
| A | B | S |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



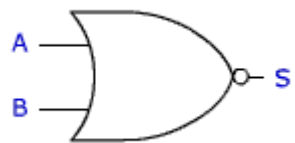
=NOT AND

(Symbole américain)
international)

(Symbole

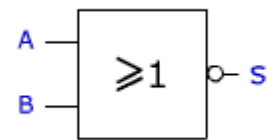


Porte NOR (Non-OU)



$$S = \overline{A + B}$$

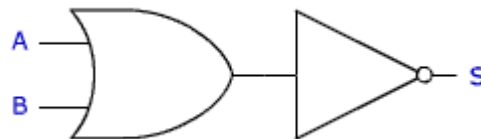
| A | B | S |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



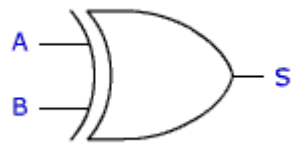
= NOT OR

(Symbole américain)
international)

(Symbole



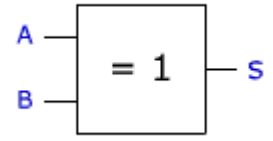
Porte XOR



(Symbole américain)
international)

$$S = A \oplus B$$

| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



(Symbole

Porte XOR à deux entrées

La fonction "OU Exclusif" est en principe d'une fonction de deux variables :

$$S = A \oplus B$$

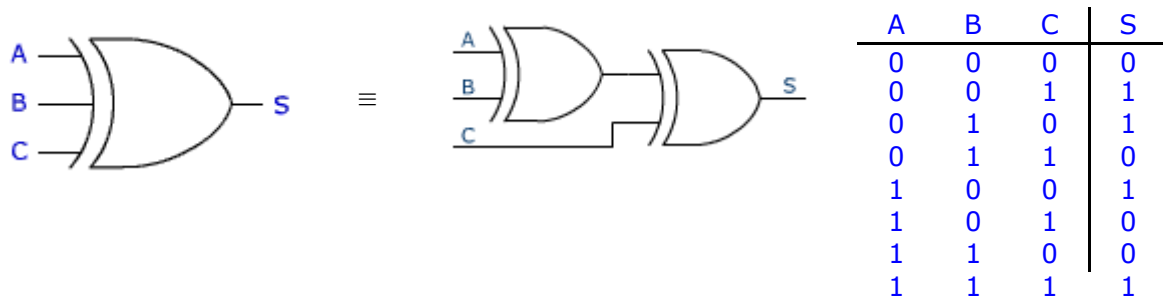
La sortie est à 1 si une seule des deux entrées vaut 1, d'où son appellation « Ou exclusif ».

Porte XOR à plusieurs entrées

Pour calculer le résultat de $S = A \oplus B \oplus C$, on doit pouvoir faire d'abord l'opération entre deux termes, puis refaire un ou exclusif entre le résultat obtenu et le troisième terme.

Ce qui se traduit par $S = (A \oplus B) \oplus C$ ou par $S = A \oplus (B \oplus C)$

On constate que l'appellation "Ou exclusif" n'est plus aussi ben à propos puisque avec trois variables, le résultat vaut 1 si une seule entrée ou toutes les trois valent 1.



Le résultat est en fin de compte un bit de parité. Il vaut 1 si le nombre d'entrées à 1 est impair.

L'inverse de la porte XOR à 2 entrées

Voyons ce que donne la table de vérité si on inverse la sortie d'une porte XOR :

| A | B | $S = \overline{A \oplus B}$ |
|---|---|-----------------------------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



Le résultat est vaut 1 si les deux entrées sont identiques.

Cette porte teste donc l'équivalence des deux entrées. Certains appellent cette fonction logique, "fonction équivalence", d'autres l'appelle "XNOR"

1-3- La fonction Booléenne F

C'est un ensemble de variables booléennes unies par des lois mathématiques

- Elle peut être définie sous forme conjonctive : lorsqu'elle consiste en produit de sommes
 $F = (a + b + c) (\overline{a} + b + \overline{c}) (a + b + c)$ etc.....
- Elle peut être définie sous la forme disjonctive : lorsqu'elle consiste en une somme de produits
 $F = a b c + \overline{a} b c + a b \overline{c} + \text{etc}.....$

Il existe trois façons de présenter la fonction booléenne :

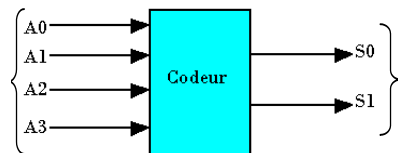
- Avec son expression algébrique
- Avec sa table de vérité
- Avec son logigramme

2- Circuits Combinatoires

La logique combinatoire concerne l'étude des fonctions dont la valeur de sortie ne dépend que de l'état logique des entrées se traduisant par une modification de la valeur des sorties et non pas non plus de ses états antérieurs : à chaque combinaison des variables d'entrée correspond toujours une seule combinaison des fonctions de sortie.

2-1- Les Codeurs

Le codeur (ou encodeur) est un circuit logique qui possède $2N$ voies entrées, dont une seule est activée et N voies de sorties. Il fournit en sortie le code binaire correspondant

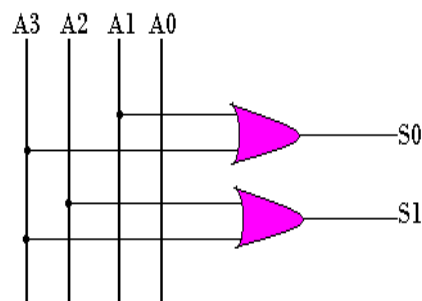


| Entées | | | | Sorties | |
|----------------------|-------|-------|-------|----------------------------|-------|
| Codage 1 parmi 2^n | | | | Nombre binaire de n bits | |
| A_3 | A_2 | A_1 | A_0 | S_1 | S_0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |

Equation des sorties

$$S_1 = 1 \text{ si } (A_2=1) \text{ ou } (A_3=1) ; S_1 = A_2 + A_3$$

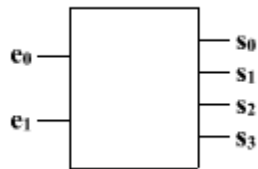
$$S_0 = 1 \text{ si } (A_1=1) \text{ ou } (A_3=1) ; S_0 = A_1 + A_3$$



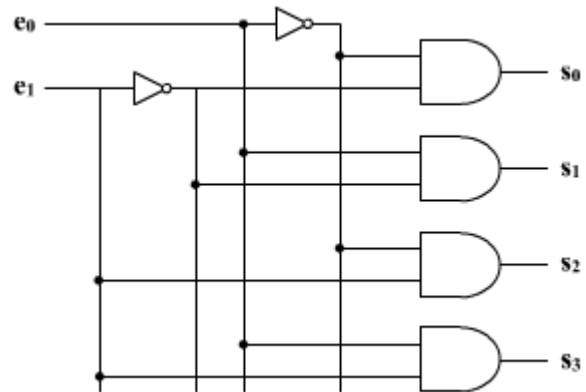
2-2- Décodeur

Le décodeur est un circuit qui possède n bits à d'entrées et au plus 2^n bits en sortie.
Parmi toutes ces sorties une seule est active, son numéro est formé par les n bits en entrée.

Exemple : Décodeur "1 parmi 4"



| e1 | e0 | s3 | s2 | s1 | s0 |
|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

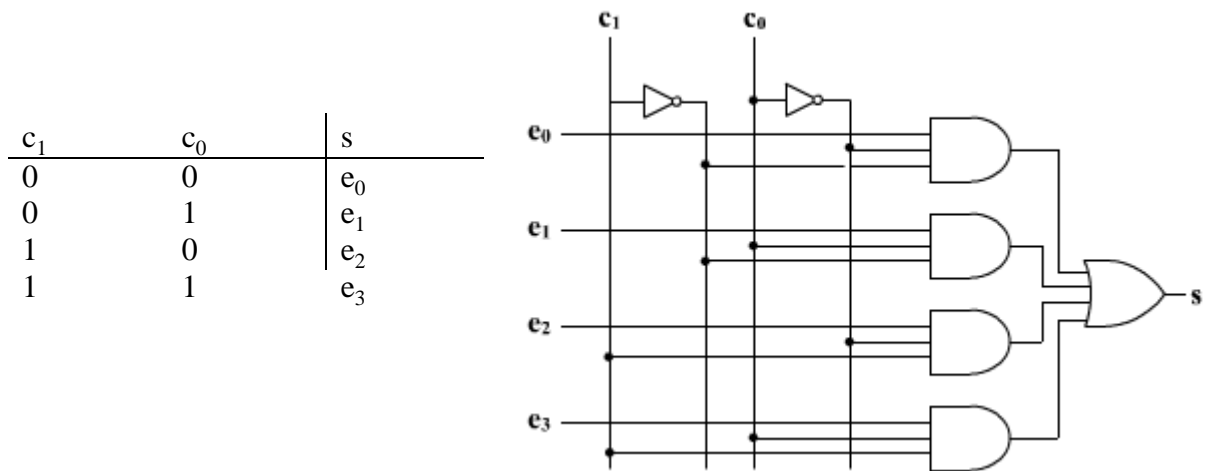


Les équations de sorties sont :

$$S_0 = \bar{e}_0 \bar{e}_1 ; S_1 = \bar{e}_0 e_1 ; S_2 = e_0 \bar{e}_1 ; S_3 = e_0 e_1$$

2-3- Multiplexeur

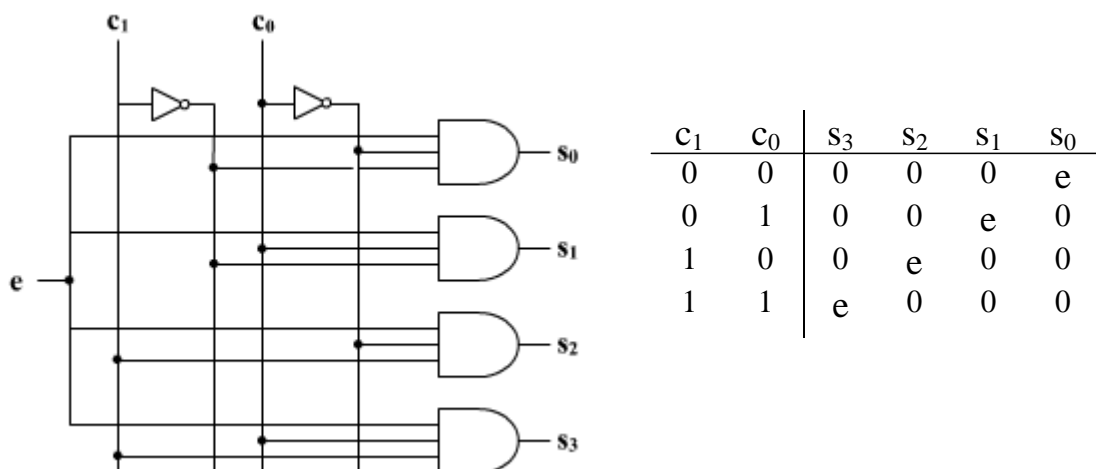
Le multiplexage est une opération qui consiste à utiliser un équipement unique pour traiter plusieurs signaux. Exemple : une ligne de transmission pour transmettre plusieurs signaux. On parle alors de multiplexage temporel : De mêmes intervalles de temps sont accordés successivement pour chacun des signaux à transmettre.



Le multiplexeur agit comme un "commutateur" qui transmet à la sortie le signal d'une entrée sélectionnée par un code binaire.

2-4- Démultiplexeur

Ce circuit réalise la fonction inverse du multiplexeur. Il possède plusieurs sorties (2^n), un signal en entrée et n bits pour désigner la sortie vers laquelle sera aiguillé le signal d'entrée.

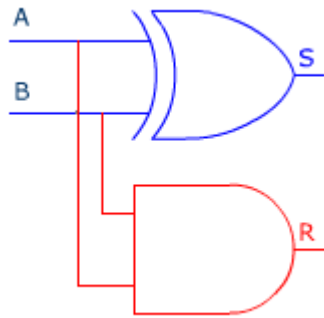


2-5- les additionneurs

Le demi additionneur

half adder

circuit à 2 entrées : 1 bit + 1 bit



$$S = A \oplus B$$

est la
somme

$R = A \cdot B$
est le
report

| A | B | S | R |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Le demi-additionneur effectue la somme de deux bits. S est la somme et R le report. (*carry*) Ce schéma ne convient cependant que pour additionner 2 nombres de 1 bit.

$$\begin{array}{r} 0 \\ + 0 \\ \hline = 00 \end{array}$$

$$\begin{array}{r} 0 \\ + 1 \\ \hline = 01 \end{array}$$

$$\begin{array}{r} 1 \\ + 0 \\ \hline = 01 \end{array}$$

$$\begin{array}{r} 1 \\ + 1 \\ \hline = 10 \end{array}$$

Le plein additionneur

full adder

Pour additionner de deux nombres de plusieurs bits il faut mettre en cascade des additionneurs qui additionnent les bits correspondant des deux nombres plus les reports R_{i-1} issus des additions des bits précédents.

Exemple : Calculons
 $1 + 3$

En binaire cela donne : $0001 + 0011$

$$\begin{array}{r} 0 \quad 1 \quad 1 \\ 0 \quad 0 \quad 0 \quad 1 \\ + 0 \quad 0 \quad 1 \quad 1 \\ \hline 0 \quad 1 \quad 0 \quad 0 \end{array}$$

L'addition des bits de droite est une addition de deux bits, elle peut être réalisée avec le demi-additionneur. Il faut tenir compte d'un éventuel report pour les bits suivants. Ainsi dès le deuxième bit de notre exemple (en comptant les bits de droite à gauche) il a fallu faire 2 additions ($1 + 0 + 1 = 10 \Rightarrow$ on pose 0 et on reporte 1)

Table de vérité du circuit plein additionneur

Cette table de vérité comporte 3 entrées : R_{n-1} (le report de l'addition précédente), A

et B Il lui faut deux sorties : S = la somme de 3 bits ($A + B + R_{n-1}$)

R = le nouveau report

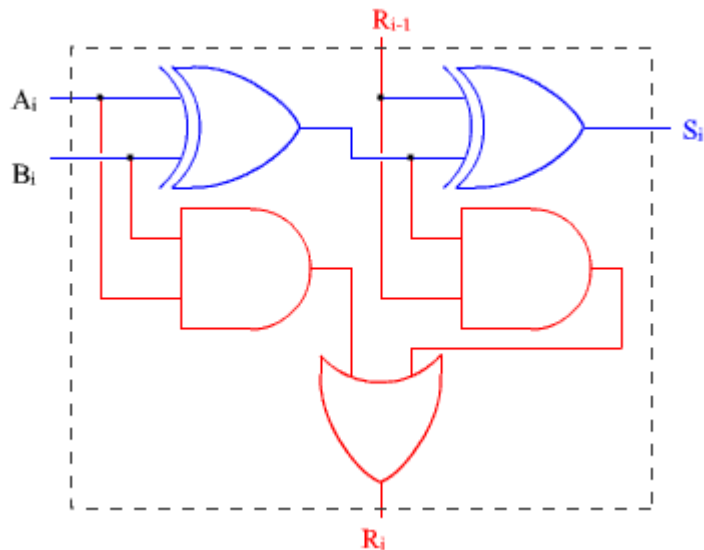
| A_i | B_i | R_{i-1} | R_i | S_i |
|-------|-------|-----------|-------|-------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Equations du circuit

$$S_i = A_i \oplus B_i \oplus R_{i-1}$$

$$R_i = \overline{A_i}B_iR_{i-1} + A_i\overline{B_i}R_{i-1} + A_iB_i\overline{R_{i-1}} + A_iB_iR_{i-1}$$

Schéma du circuit plein additionneur



$$S_i = A_i \oplus B_i \oplus R_{i-1}$$

$$R_i = \overline{A_i}B_iR_{i-1} + A_i\overline{B_i}R_{i-1} + A_iB_i\overline{R_{i-1}} + A_iB_iR_{i-1}$$

Le plein additionneur est un circuit à 3 entrées. Il se compose de 2 demi additionneurs et d'une porte OU qui génère le report quand la somme vaut 2 ou 3

Les circuits séquentiels : Les Bascules

I- Généralités

Les circuits auxquels nous nous sommes occupés jusqu'à présent appartiennent à la logique combinatoire ; c'est-à-dire des circuits qui produisent des résultats qui dépendent de la combinaison des variables appliquées à leurs entrées.

Exemples : « les portes OU ; ET ; NAND ; NOR ; etcetera ... »

La sortie d'un tel circuit est régie par équation du type $S=f(a, b, \dots, n)$

Les bascules appartiennent à la logique séquentielle ; Ces circuits produisent des résultats qui dépendent non seulement de la combinaison des variables appliquées à leurs entrées mais aussi à l'état antérieur de celles-ci .

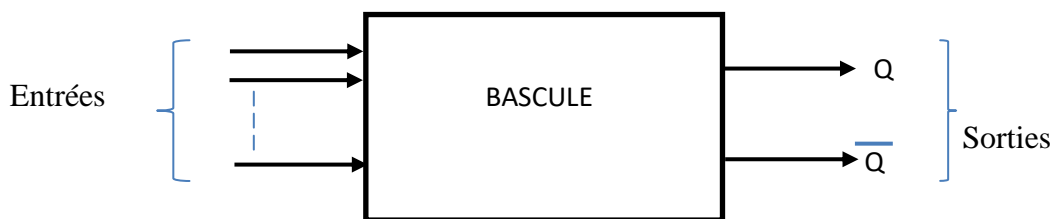
Il faut connaître l'état antérieur des sorties, combiné à l'état actuel des entrées pour le résultat final. La sortie d'un tel circuit est régie par une équation de type $S_{t+1}=f_t(a,b,\dots,n)$. Une bascule peut garder un état aussi longtemps que possible tant qu'on lui fasse changer celui-ci ; c'est la raison pour laquelle on les utilise comme cellule de mémoire.

La plupart des bascules possèdent une horloge, une sorte de gendarme qui organise la circulation des informations logiques.

Une bascule est caractérisée par plusieurs entrées mais avec seulement deux sorties complémentaires Q et \overline{Q} .

L'état de la bascule est déterminé par la sortie Q

- Si $Q=1$, on dit que la bascule est activée
- Si $Q=0$, on dit que la bascule est désactivée



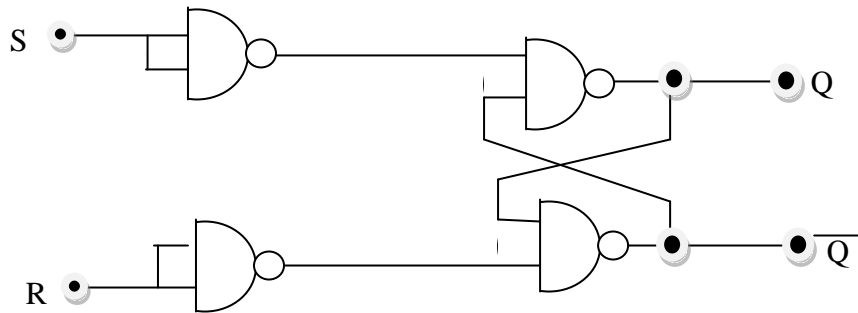
II- Bascule RS

C'est la première bascule qui a été conçue ; Elle possède deux entrées repérées par les lettres $S=SET$ (activer) et $R=RESET$ (désactiver) .

- L'application simultanée de deux niveaux bas (0) aux entrées S et R ne provoque aucune réaction de la bascule, qui se maintient à l'état initial.
- L'application simultanée de deux niveaux différents aux entrées S et R met la bascule à l'état de l'entrée qui a été activée
- L'application simultanée de deux niveaux hauts ($S=1$ et $R=1$) met la bascule dans un état imprévisible d'où cette combinaison est dite non recommandée

II-1- Bascule RS asynchrone (Pas de signal d'horloge)

II-1-1- schéma



II-1-2- Table de vérité

| Q_t | $\overline{Q_t}$ | S | R | Q_{t+1} | $\overline{Q_{t+1}}$ |
|----------|------------------|---|---|-----------|----------------------|
| 0 | 1 | 0 | 0 | 0 | 1 |
| | | 0 | 1 | 0 | 1 |
| | | 1 | 0 | 1 | 0 |
| | | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| | | 0 | 1 | 0 | 1 |
| | | 1 | 0 | 1 | 0 |
| | | 1 | 1 | 1 | 1 |

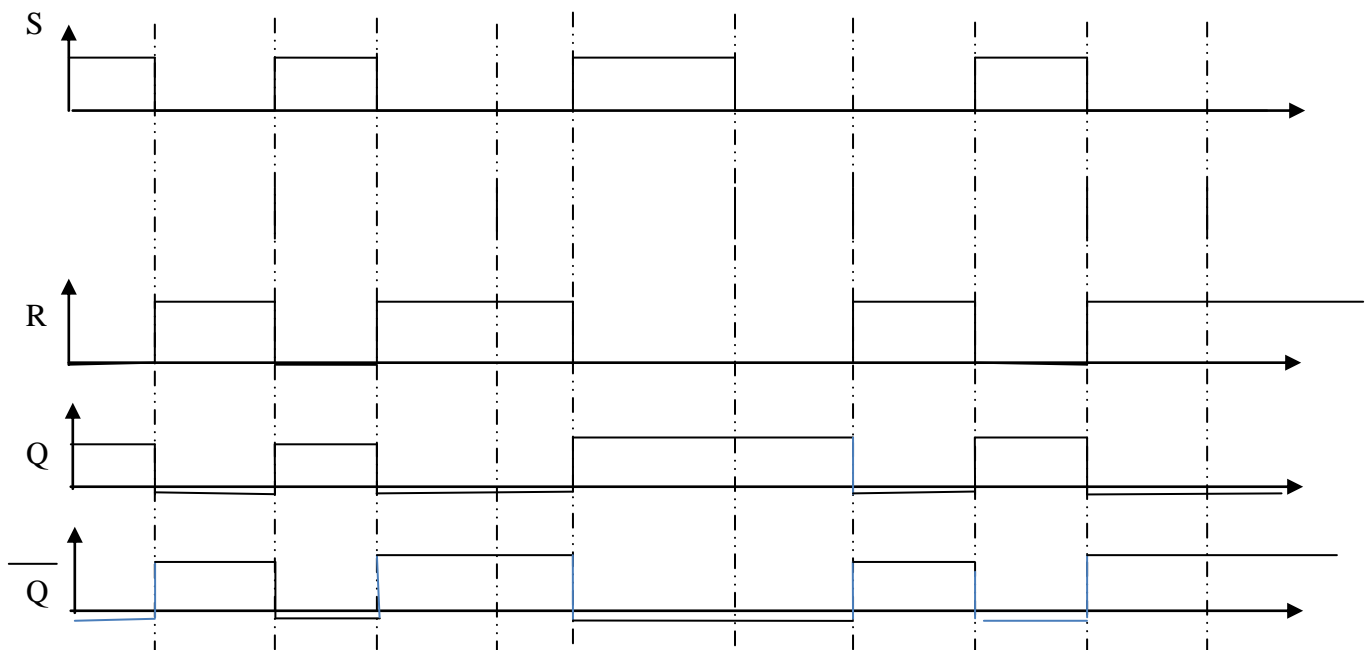
II-1-3 l'équation de la sortie Q_{t+1}

$$Q_{t+1} = S \overline{R} \overline{Q_t} + S R \overline{Q_t} + S \overline{R} Q_t + S R Q_t + S R Q_t$$

$$= S \overline{R} (Q_{t+} \overline{Q_t}) + S R (Q_{t+} \overline{Q_t}) + S \overline{R} Q_t$$

$$= S \overline{R} + S R + S \overline{R} Q_t = S + S \overline{R} Q_t = S + \overline{R} Q_t$$

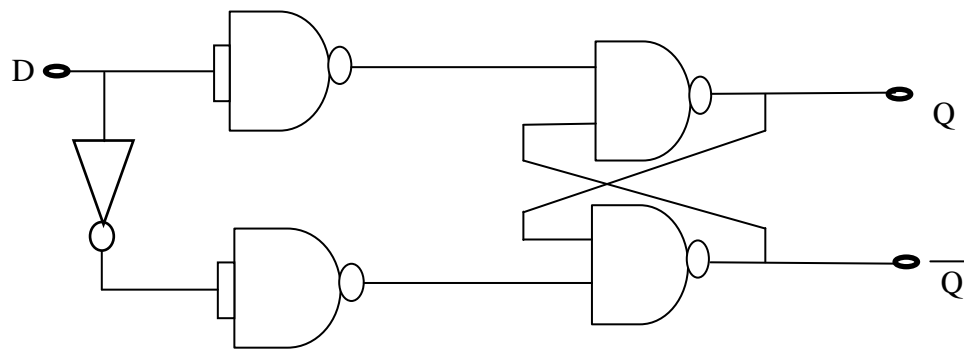
II-1-3- Les chronogrammes



La bascule D est une correction de la bascule RS ; Pour éviter que les combinaison non recommandée(S=1 et R=1) ne se produise on place un inverseur entre les entrées S et R et de commander la bascule à partir d'un seul endroit D. C'est la première bascule qui a été fabriquée ; elle participe à la fabrication de plusieurs circuits séquentiels.

III-1- Bascule D asynchrone

III-1-1-schéma



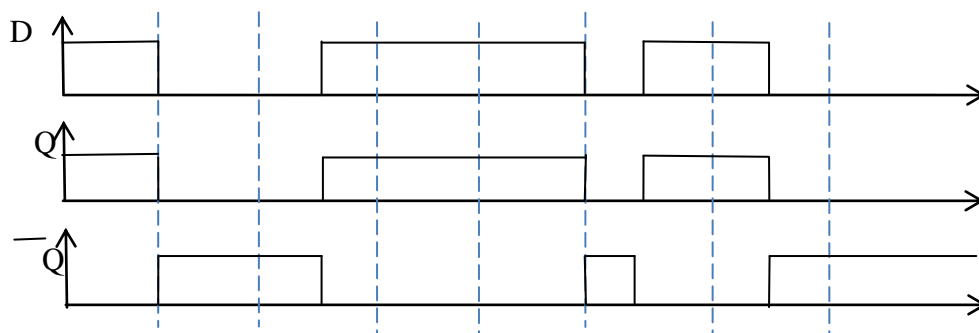
III-1-2- la table de vérité

| Q_t | \overline{Q}_t | D | Q_{t+1} | \overline{Q}_{t+1} |
|-------|------------------|---|-----------|----------------------|
| 1 | 0 | 0 | 0 | 1 |
| | | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| | | 1 | 1 | 0 |

III-1-3- Equation de la sortie Q_{t+1}

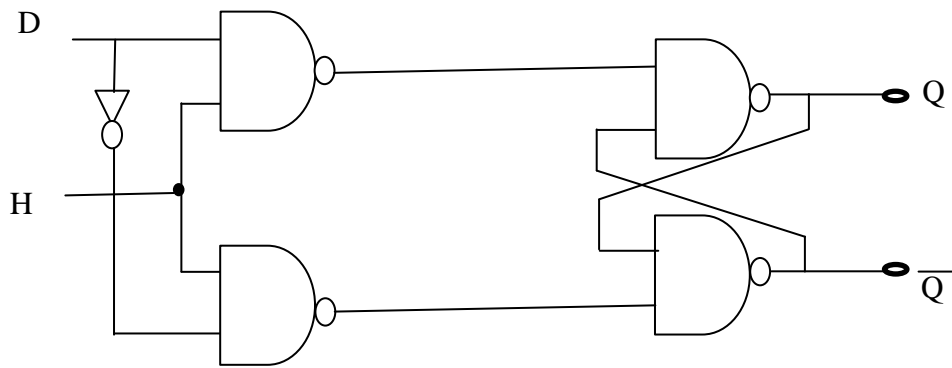
$$Q_{t+1} = DQ_t + D\overline{Q}_t = D(Q_t + \overline{Q}_t) = D$$

III-1-4- les chronogrammes



III-2- Bascule Dsynchrone (DH)

III-2-1- schéma



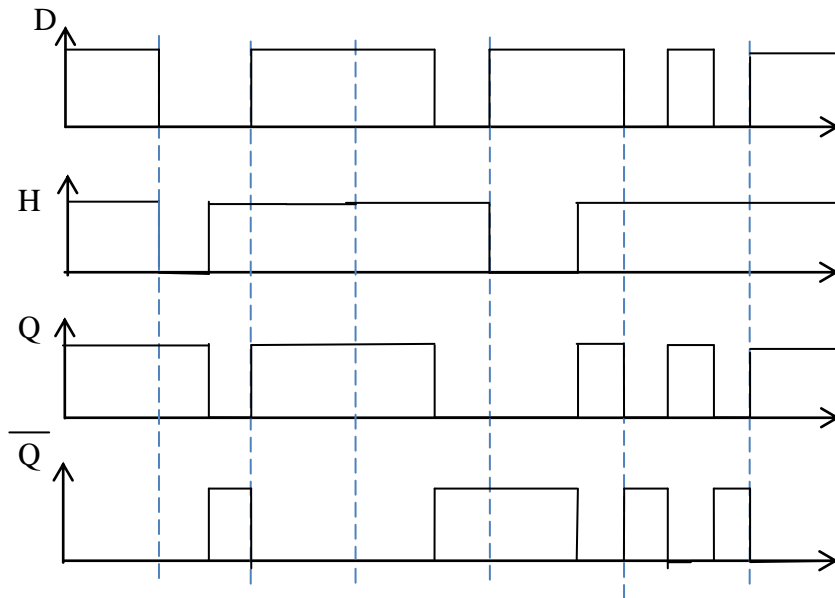
III-2-2-table de vérité

| Q_t | $\overline{Q_t}$ | H | D | Q_{t+1} | $\overline{Q_{t+1}}$ |
|-------|------------------|---|---|-----------|----------------------|
| 1 | 0 | 0 | 0 | 1 | 0 |
| | | 1 | 0 | 0 | 1 |
| | | 0 | 1 | 0 | 1 |
| | | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| | | 1 | 0 | 0 | 1 |
| | | 0 | 1 | 0 | 1 |
| | | 1 | 1 | 1 | 0 |

III-2-3- Equation de la sortie Q_{t+1}

$$Q_{t+1} = \overline{D}H\overline{Q_t} + D\overline{H}Q_t + D\overline{H}\overline{Q_t} = D\overline{H} + \overline{D}H\overline{Q_t}$$

III-2-4- Les chronogrammes



IV-Bascule JK

La bascule D n'a pu résoudre le problème car en évitant la combinaison (11) on a exclu (00) qui est valable.

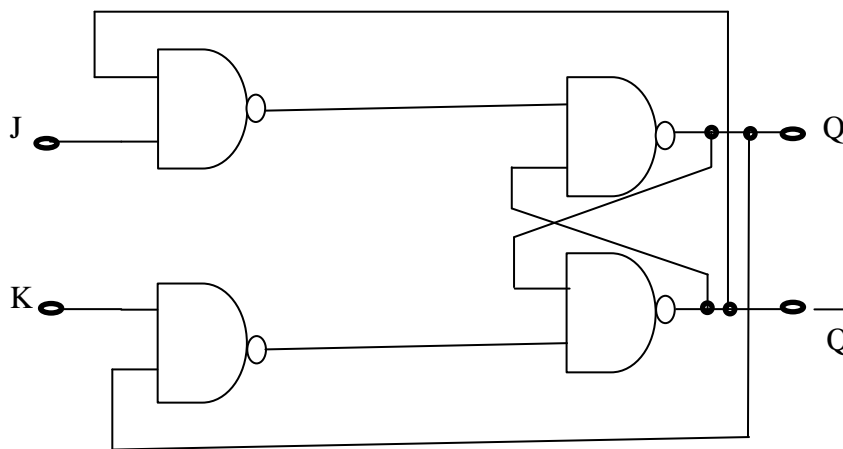
La bascule JK sera celle qui permettra l'utilisation de toutes les combinaisons ; ce qui fera d'elle la plus parfaite avec laquelle aussi on a réalisé la plupart des circuits séquentiels.

Son nom est inspiré des figurines d'un jeu anglo-saxon où J=Jack (officier) et K=king(Roi).

- L'application simultanée de d'un niveau bas (00) aux entrées J et K ne produit aucun effet au niveau de la bascule: c'est le maintien à l'état initial
- L'application de deux niveaux différents en J et K met la bascule à l'état de J ;
- L'application de deux niveaux hauts en J et K provoque un changement immédiat de l'état de la bascule.

1- Bascule JK asynchrone

a- schéma



b- La table de vérité

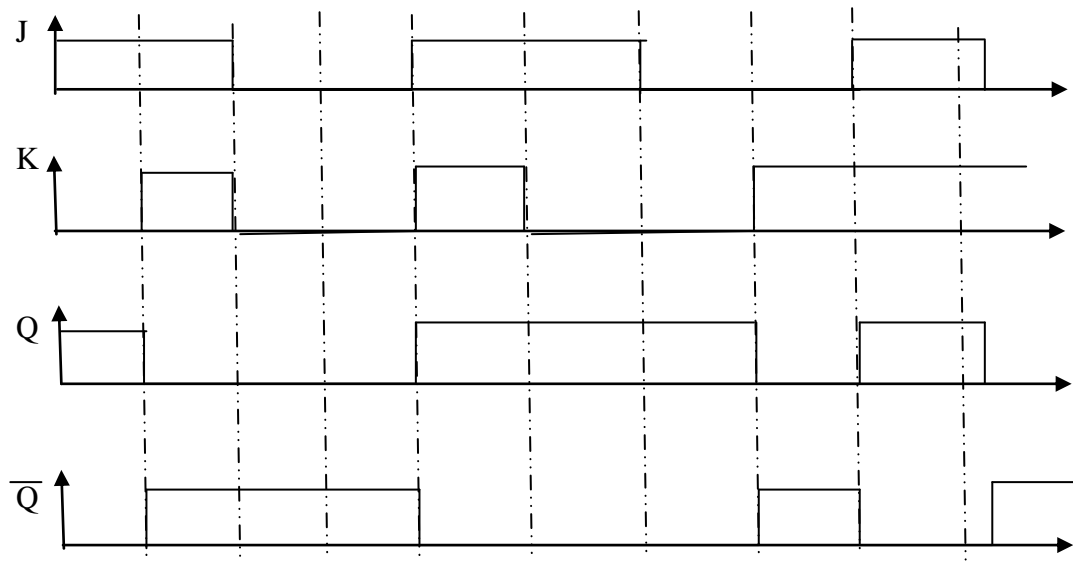
| Q_t | $\overline{Q_t}$ | J | K | Q_{t+1} | $\overline{Q_{t+1}}$ |
|-------|------------------|---|---|-----------|----------------------|
| 1 | 0 | 0 | 0 | 1 | 0 |
| | | 0 | 1 | 0 | 1 |
| | | 1 | 0 | 1 | 0 |
| | | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| | | 0 | 1 | 0 | 1 |
| | | 1 | 0 | 1 | 0 |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

c- Equation de la sortie Q_{t+1}

$$Q_{t+1} = \overline{J} \overline{K} Q_t + J \overline{K} Q_t + J \overline{K} \overline{Q_t} = \overline{J} \overline{K} Q_t + J \overline{K} (Q_t + \overline{Q_t}) = (\overline{J} Q_t + J) \overline{K}$$

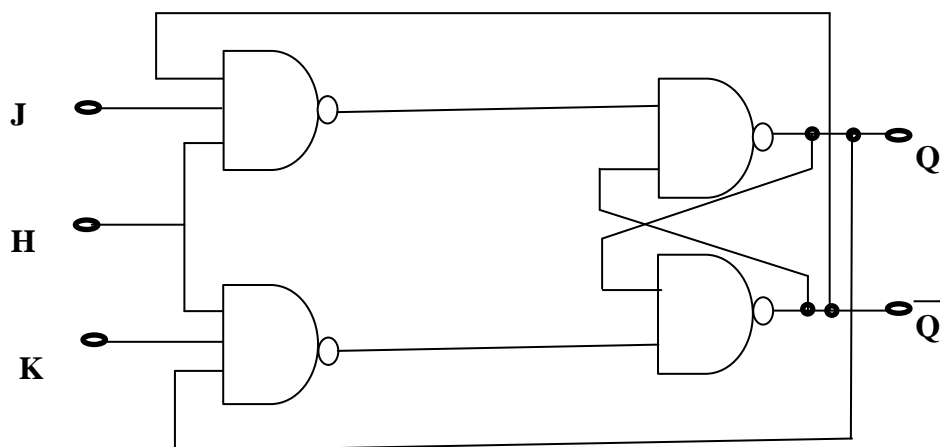
$$= \overline{K} (J + Q_t) = J \overline{K} + \overline{K} Q_t$$

d- Les Chronogrammes



2- Bascule JK synchrone

a- Schéma

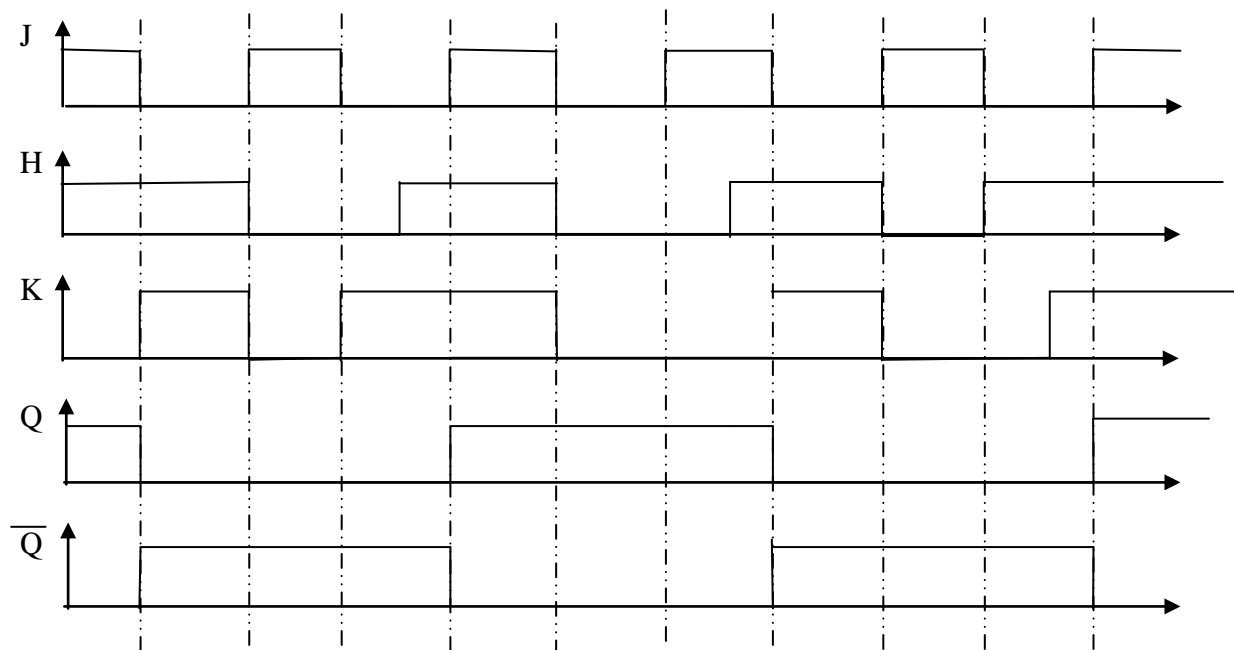


b- Table de vérité

| Q_t | \overline{Q}_t | H | J | K | Q_{t+1} | \overline{Q}_{t+1} |
|-------|------------------|---|---|---|-----------|----------------------|
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| | | 1 | 0 | 0 | 1 | 0 |
| | | 0 | 0 | 1 | 1 | 0 |
| | | 1 | 0 | 1 | 0 | 1 |
| | | 0 | 1 | 0 | 0 | 1 |
| | | 1 | 1 | 0 | 1 | 0 |
| | | 0 | 1 | 1 | 1 | 0 |
| | | 1 | 1 | 1 | 0 | 1 |

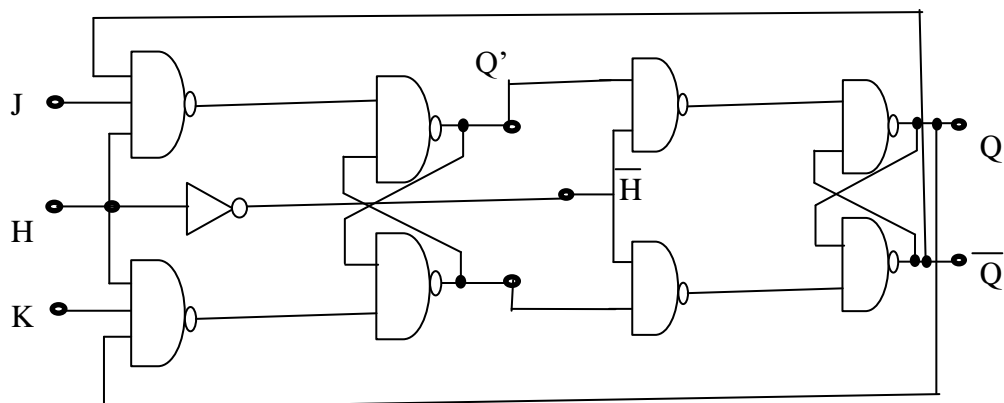
NB : L'équation d'une sortie sera écrite avoir complété le tableau

c- Les chronogrammes



3- Bascule JK maitre- esclave

a- Schéma



b- Table de vérité

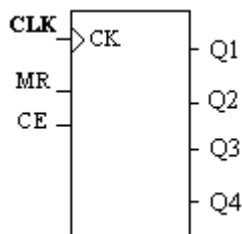
| Q_t | $\overline{Q_t}$ | H | J | K | Q' | \overline{H} | Q_{t+1} | $\overline{Q_{t+1}}$ |
|-------|------------------|---|---|---|------|----------------|-----------|----------------------|
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| | | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| | | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| | | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| | | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| | | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| | | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| | | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

Compteur-Décompteur

1- .Descriptions

Un compteur est un circuit séquentiel qui permet de dénombrer des impulsions appliquées sur son entrée d'horloge (clk : clock) et de restitué sur ces sorties des informations sous forme binaire .A chaque impulsion l' état du compteur est modifié et entre deux impulsions son état reste stable ,un compteur est donc aussi une sorte de registre mémoire .Un compteur doit démarrer vide ou plein suivant l' utilisation , l'entrée RAZ ou CLR ou MR ou R sert de remise à zéro .L' entrée CE (count enable) servant a valider le circuit ou interrompre le comptage .

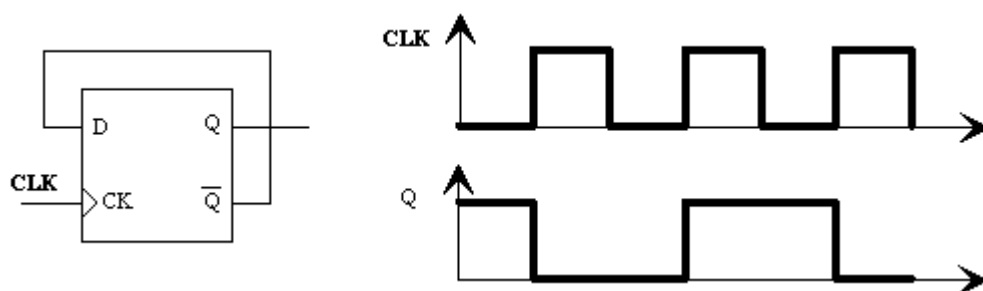
1-1- Symboles



.Unités ; Formules

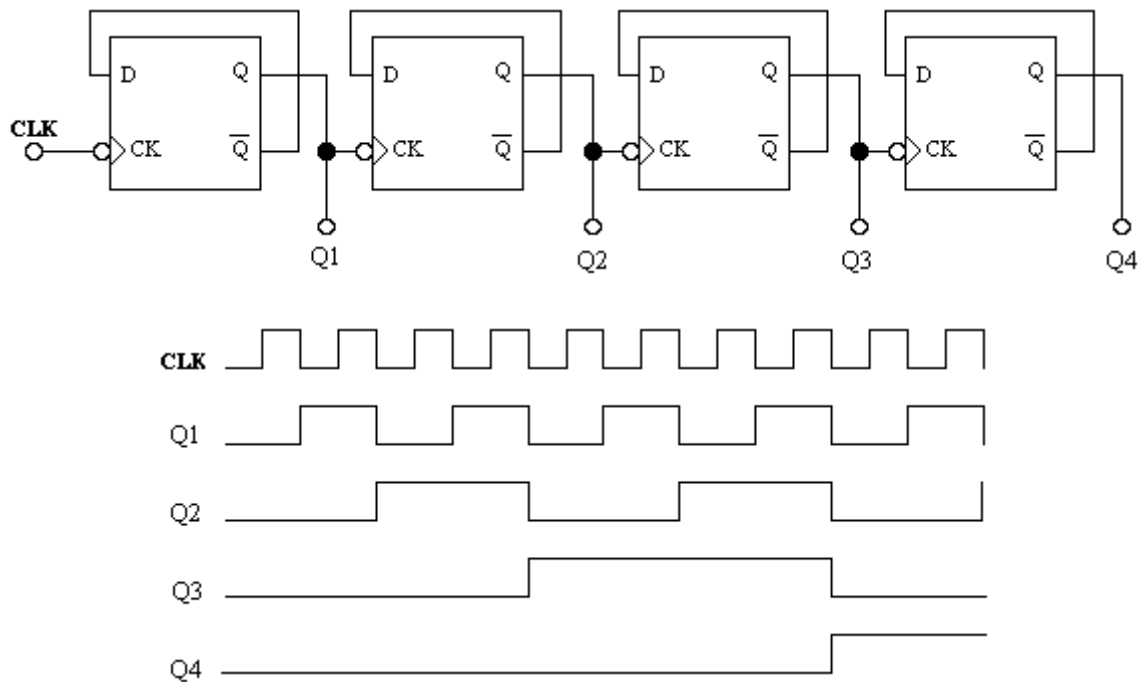
2- Compteur asynchrone

Le compteur asynchrone est basé sur le diviseur de fréquence. Il s'agit d'une simple bascule D dont la sortie inverseuse est envoyée sur son entrée D. Ainsi, à chaque front montant de l'entrée d'horloge, la donnée transférée sur la sortie est complémentée. La fréquence du signal est donc divisée par deux :



Un compteur asynchrone est simplement constitué de plusieurs diviseurs de fréquences (autant que de bits requis)

Si l'on désire un compteur, les entrées d'horloge doivent être actives sur front descendant. Cela conduit à ce résultat :



Comme vous pouvez le constater, Q1 Q2 Q3 Q4 sont successivement égales à 0000, 0001, 0010, ..., 1111, c'est-à-dire 0, 1, 2, ..., 15. Bien sûr ce cycle continue indéfiniment, aussi longtemps que le signal d'horloge est présent. C'est la façon la plus simple de réaliser un compteur, et bien sûr ce type de compteur est disponible sous forme de divers circuits intégrés prêts à l'emploi.

Exemple : Compteur asynchrone

M74HC393

DUAL BINARY COUNTER

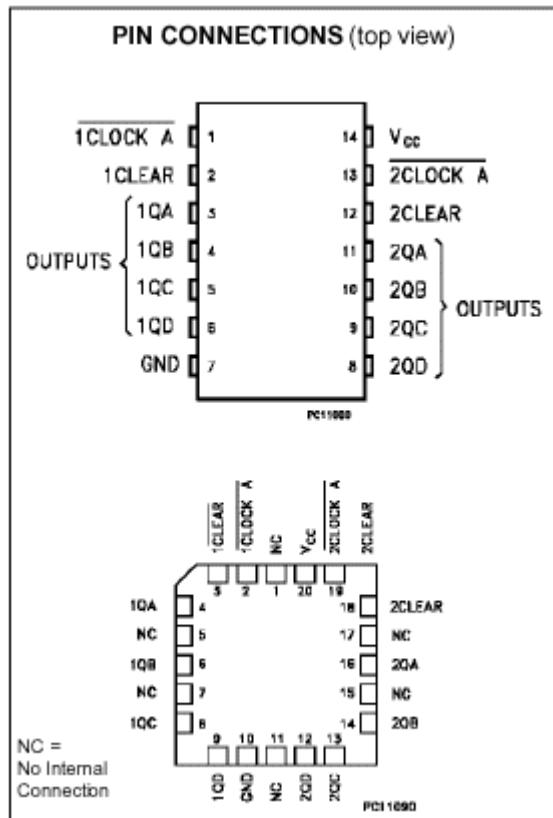
DESCRIPTION

The M54/74HC393 is a high speed CMOS DUAL BINARY COUNTER fabricated in silicon gate C2MOS technology. It has the same high speed performance of LSTTL combined with true COMS low power consumption.

This counter circuit contains independent ripple carry counters and two 4-bit ripple carry binary counters, which can be cascaded to create a single divide by 256 counter.

Each 4-bit counter is incremented on the high to low transition (negative edge) of the clock input, and each has an independent clear input. When clear is set to low all four bits of each counter are set to a low level. This enables count truncation and allows the implementation of divide by N counter configurations.

All inputs are equipped with protection circuits against static discharge and transient excess voltage.



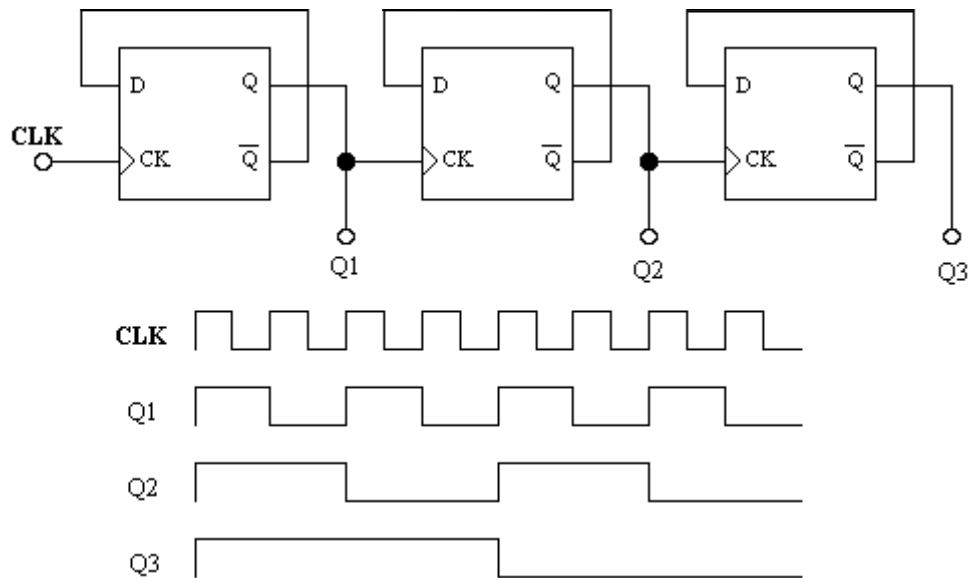
| INPUTS | | OUTPUTS | | | |
|--------|-------|-----------|----|----|----|
| CLOCK | CLEAR | QD | QC | QB | QA |
| X | H | L | L | L | L |
| | L | COUNT UP | | | |
| | L | NO CHANGE | | | |

X: Don't Care

| COUNT | OUTPUT | | | |
|-------|--------|----|----|----|
| | QD | QC | QB | QA |
| 0 | L | L | L | L |
| 1 | L | L | L | H |
| 2 | L | L | H | L |
| 3 | L | L | H | H |
| 4 | L | H | L | L |
| 5 | L | H | L | H |
| 6 | L | H | H | L |
| 7 | L | H | H | H |
| 8 | H | L | L | L |
| 9 | H | L | L | H |
| 10 | H | L | H | L |
| 11 | H | L | H | H |
| 12 | H | H | L | L |
| 13 | H | H | L | H |
| 14 | H | H | H | L |
| 15 | H | H | H | H |

3- Décompteur :

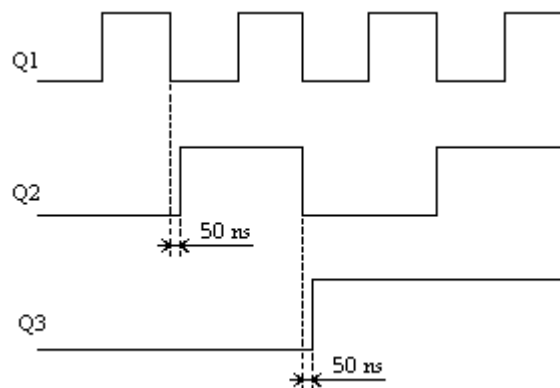
Pour réaliser un décompteur le signal d'horloge est actif sur front montant.
Exemple de décompteur 3 bits, qui décompte de 7 à 0 :



Asynchrone :

Pourquoi ce type de compteur est-il qualifié d'asynchrone ou à Propagation ?

Comme tout circuit logique, la bascule D ne réagit pas immédiatement à un changement sur son entrée d'horloge. Il existe un temps de transfert (quelques nanosecondes) entre Q0 et Clk. Le même écart existe entre Q1 et Q0, et entre Q2 et Q1. Pour n bits, le dernier signal est décalé de n fois cet écart élémentaire, ce qui peut devenir un décalage assez important. Les signaux ne sont plus synchrones, et si une combinaison logique de ces signaux doit être réalisée, des glitches apparaissent (transitions parasites 0-1-0 ou 1-0-1). Souvent cela ne pose pas de problème car seule la dernière sortie est utilisée. Mais dans les autres cas l'utilisation d'un compteur synchrone est préférable.



Modulo :

Si un compteur possède 16 états distincts (0000 à 1111), on dit alors que c'est un compteur Modulo 16