

Chapitre 2

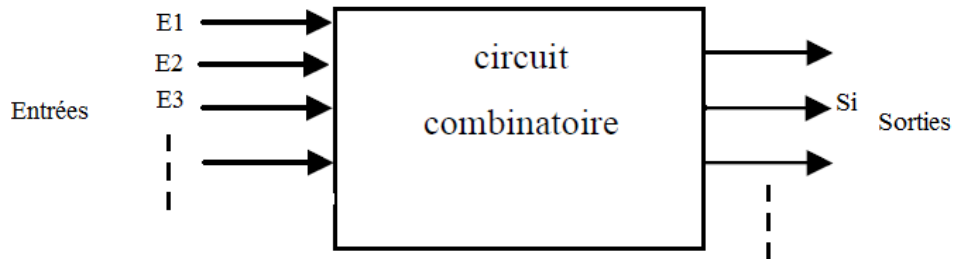
Logique séquentielle

2.1. Introduction aux circuits séquentiels

Dans les circuits combinatoires une sortie S_i est directement liée aux entrées à chaque instant.

$$S_i = F(E_1, E_2, \dots)$$

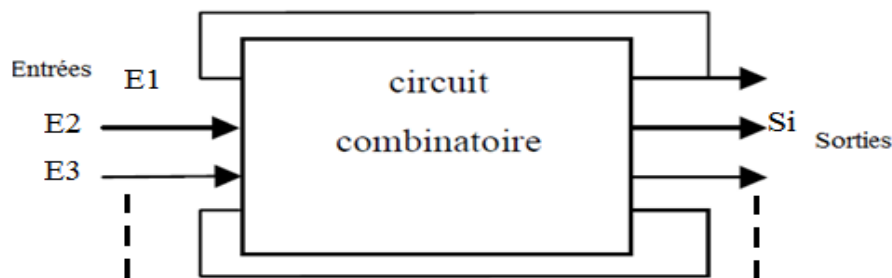
à chaque combinaison d'entrée correspond un seule état de sortie.



Ainsi, la sortie ne dépend pas des états antérieurs des entrées. Par conséquent, le temps n'intervient pas dans le fonctionnement de ces circuits.

Dans les circuits séquentiels la sortie S_i dépend des entrées et des états antérieurs des sorties (leurs passé). Ainsi, le temps constitue un des paramètres essentiels.

$$S_i^t = F(E_1, E_2, \dots, S_i^{t-1})$$



Exemple :

Etat n°1 : $E_1 = 0$ et $E_2 = 0 \Rightarrow S_i = 0$

Etat n°2 : $E_1 = 1$ et $E_2 = 0 \Rightarrow S_i = 1$

Etat n°3 : $E_1 = 0$ et $E_2 = 0 \Rightarrow S_i = 1$

Etat n°4 : $E_1 = 0$ et $E_2 = 1 \Rightarrow S_i = 0$

Etat n°5 : $E_1 = 0$ et $E_2 = 0 \Rightarrow S_i = 0$

Ainsi, un système logique séquentiel ne fournit pas la même sortie pour une même configuration d'entrées (état 3 et état 5). Sa fonctionnalité dépend donc de l'ordre des opérations (le déroulement des séquences).

Les circuits séquentiels fondamentaux sont :

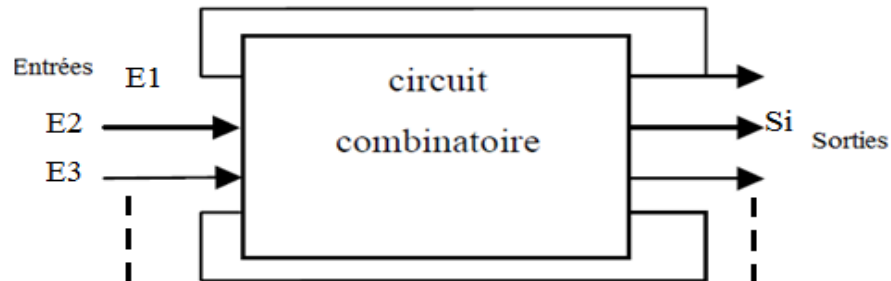
- Basculer,
- Registre,
- Compteur.

Les applications de base dans les systèmes informatiques sont :

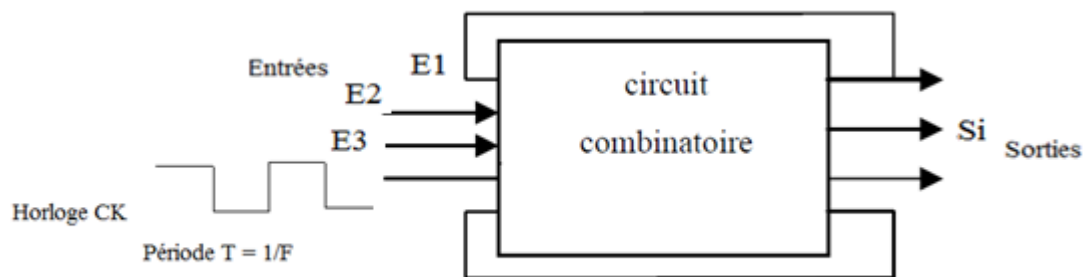
- Mémoire vive statique SRAM (Static Random Access Memory),
- Registres internes des microprocesseurs et microcontrôleurs

Différents types de circuit séquentiel

- Circuit séquentiel asynchrone: À tout moment Les sorties du montage peuvent changer dès qu'une ou plusieurs entrées changent après un temps de propagation (retard ou temps de réponse). Ils sont très difficiles à gérer.



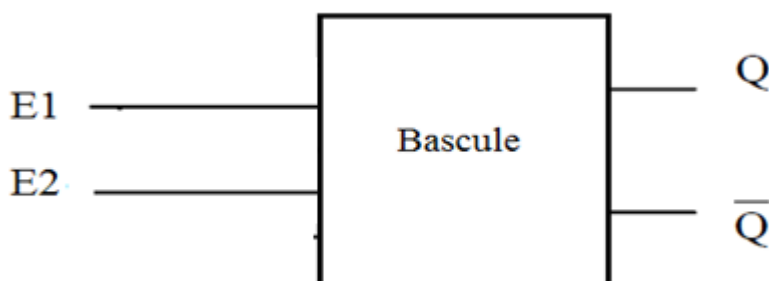
- Circuit séquentiel synchrone: Le changement des états de sorties est commandé, en plus des entrées, par un signal d'horloge CK. Il constitue une base de temps et représente un moyen de synchronisation des différentes opérations. Les changements d'état des sorties s'effectuent soit par **niveau logique** 1 ou 0 ou par une transition appelée « **front actif** » (montant ou descendant) du signal d'horloge. Ainsi, les entrées et les états antérieurs des sorties préparent les futurs changements d'états des sorties mais ne peuvent provoquer ces changements sans une synchronisation avec le signal d'horloge.



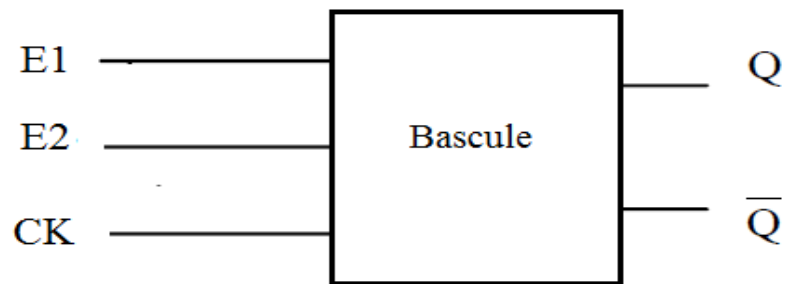
2.2. Bascules

Une bascule est un circuit logique séquentiel, appelé également éléments bistables, latch ou flip flop, qui possède deux entrées et deux sorties complémentaires Q et \bar{Q} .

Bascule asynchrone :

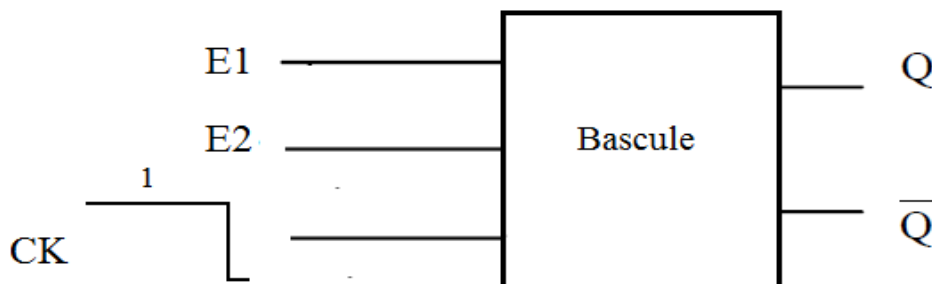


Bascule synchrone :

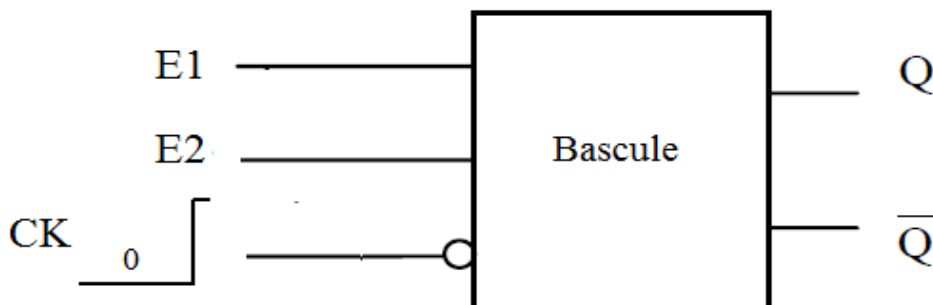


Différents types de bascules synchrones :

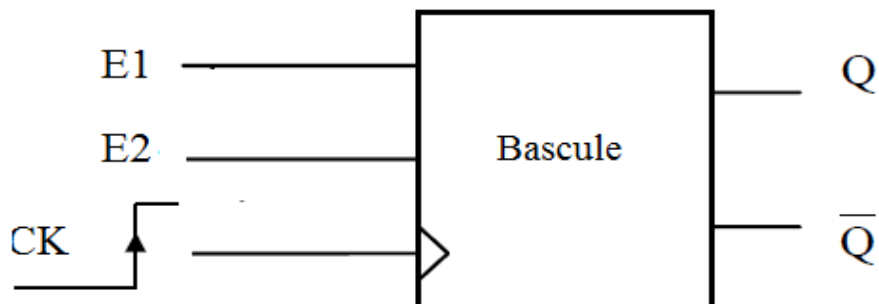
a. Bascule commandée par niveau haut du signal d'horloge (1 logique)



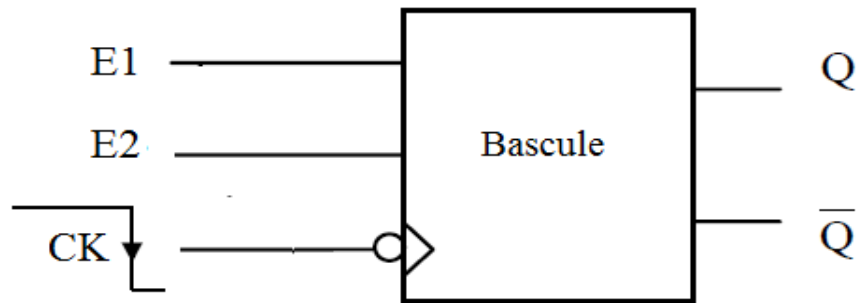
b. Bascule commandée par niveau bas du signal d'horloge (0 logique)



c. Bascule commandée par niveau front montant du signal d'horloge



d. Bascule commandée par niveau front descendant du signal d'horloge



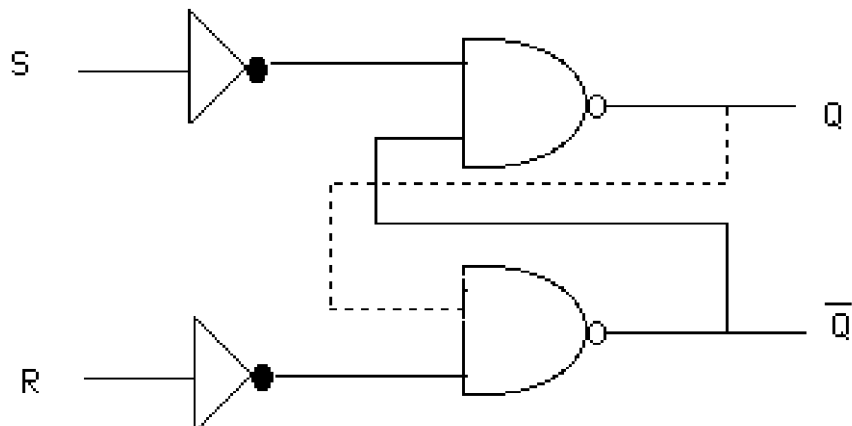
Il est à noter que la synthèse et la conception des circuits séquentiels sont effectuées par **la méthode de Huffman** (non traitée dans ce cours).

2.3. Bascule de type RS

a. Bascule RS asynchrone

La bascule RS est la base de toutes bascules standards (D, JK, ...). La sortie Q de la bascule est mise à 0 lorsque l'entrée RESET ($R = 1$) est activée et mise à 1 lorsque l'entrée SET ($S = 1$) est activée.

Circuit à base de portes NAND



Symbole

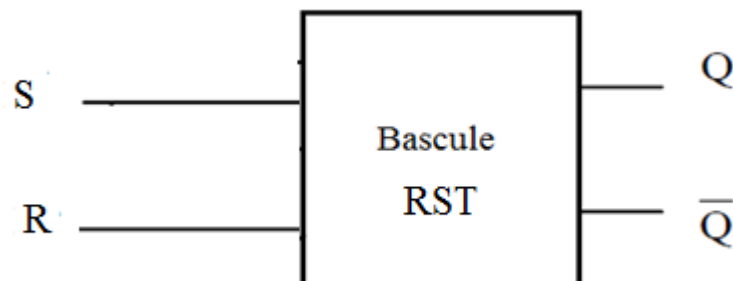


Table de vérité d'une bascule RS asynchrone

| Sorties à t_n | | Entrées | | Sorties à t_{n+1} | |
|-----------------|----------------|---------|---|---------------------|----------------|
| Q | \overline{Q} | S | R | Q | \overline{Q} |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 Reset |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 ind !!! | 1 ind !!! |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 Set |
| 0 | 1 | 1 | 1 | 1 ind !!! | 1 ind !!! |

Ind : indéterminé

On remarque que pour des états d'entrées identiques, les sorties sont différentes ($R=S=0 \Rightarrow Q=1$ ou $Q=0$ en fonction de l'état de sortie précédent).

Ainsi, les circuits séquentiels dépendent des états d'entrées et des états antérieurs des sorties.

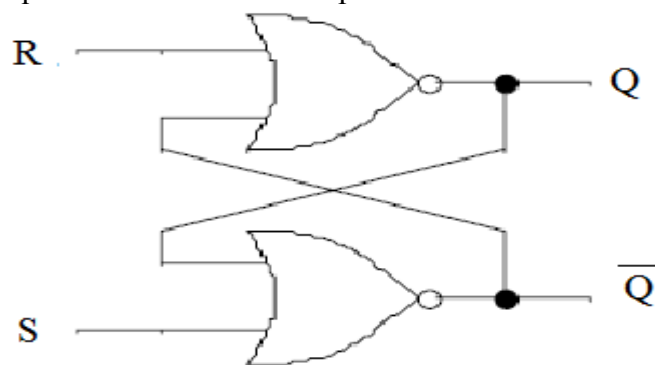
Table de vérité condensée

| S | R | Q |
|---|---|-------------|
| 0 | 0 | S.C |
| 0 | 1 | Etat Reset |
| 1 | 0 | Etat Set |
| 1 | 1 | indéterminé |

SC: Sans Changement, état de mémorisation

b. Bascule R-S à base des portes NOR

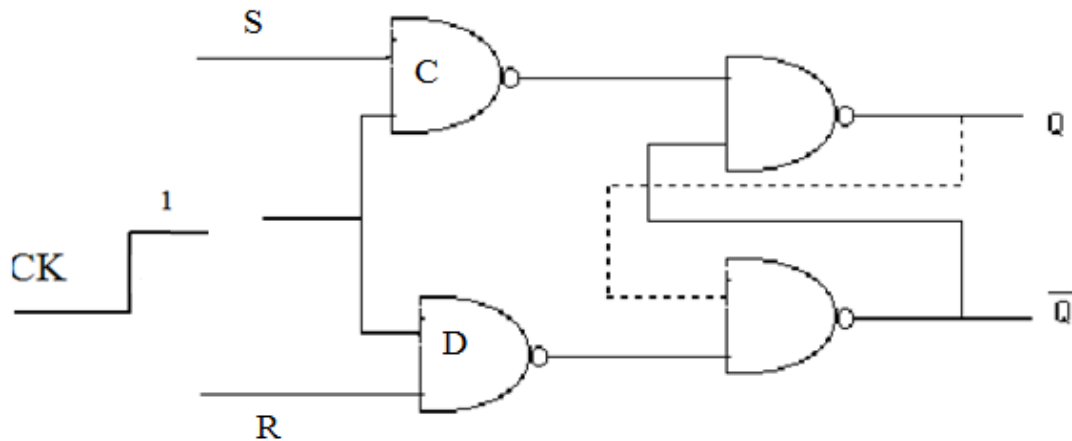
Même raisonnement que la bascule à base des portes NAND.



c. Bascule RS Synchrone

La bascule synchrone est pilotée par une horloge CK. Ce qui permet de changer d'état de la bascule à des moments précis. Elle est appelée **RST** ou **RSH**.

Circuit à base de portes NAND activé par une horloge par niveau haut



- Quand CK passe au niveau haut la bascule se trouvera sous le contrôle des entrées R et S.
- Lorsque l'entrée d'horloge CK est au niveau 0, les portes C et D sont à l'état 1 indépendamment des entrées R et S. La bascule se maintient ainsi dans un état pris antérieurement.

Table de vérité d'une bascule RS synchrone

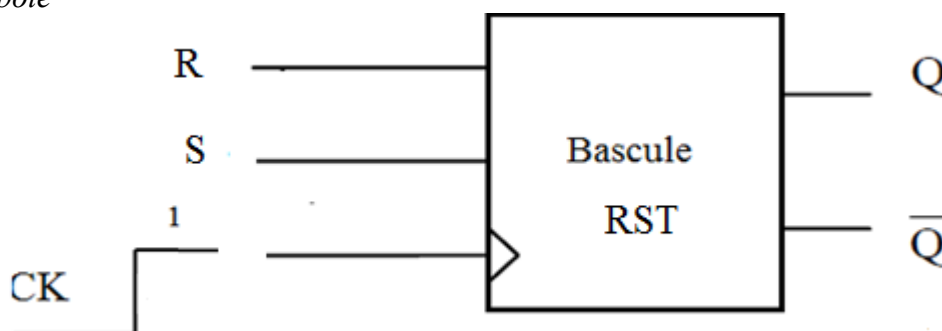
| S | R | CK | Q | \bar{Q} |
|---|---|----|-----|--------------|
| 0 | 0 | 0 | S.C | S.C |
| 0 | 0 | 1 | S.C | S.C |
| 0 | 1 | 0 | S.C | S.C |
| 0 | 1 | 1 | 0 | 1 Etat Reset |
| 1 | 0 | 0 | S.C | S.C |
| 1 | 0 | 1 | 1 | 0 Etat Set |
| 1 | 1 | 0 | S.C | S.C |
| 1 | 1 | 1 | Ind | ind |

SC : sans changement, état de mémorisation

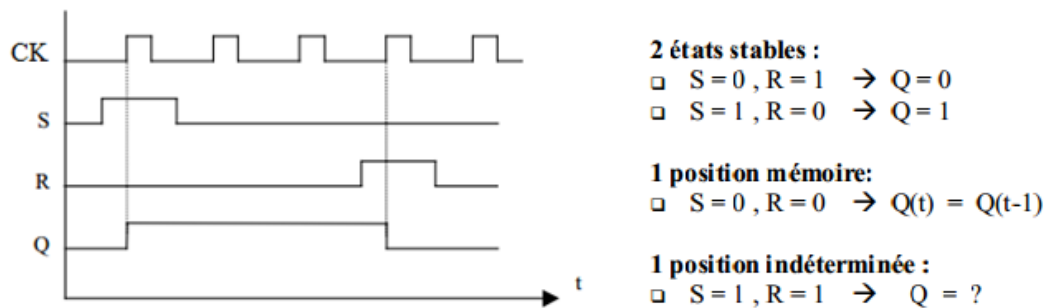
Ind: indéterminé.

L'état indéterminé constitue un inconvénient de la bascule. Depuis cet état, quand R ou S change d'état (de 1 à 0), nous serons dans l'impossibilité de déduire dans quels états vont basculer Q et \bar{Q} .

Symbole

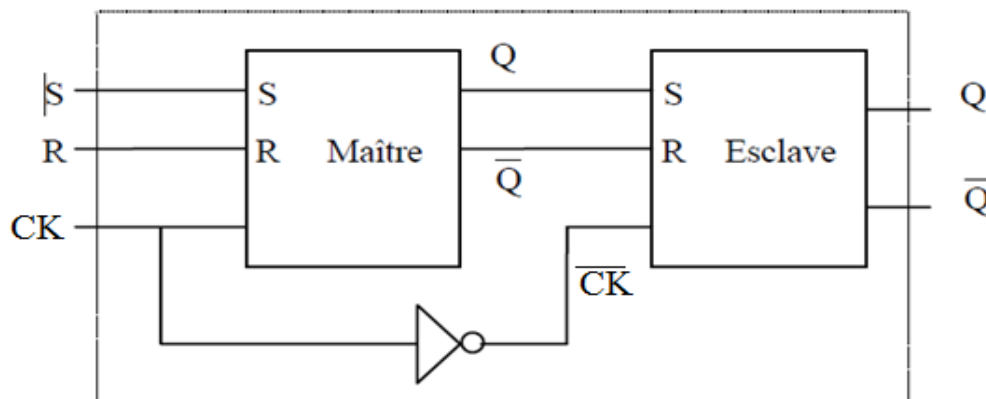


Chronogramme : La bascule RST est sensible au front montant du signal d'horloge



d. Bascules RST maître-esclave

Le circuit RST maître-esclave est composé deux bascules RST, elles montées en cascade et commandées par deux horloges en opposition de phase réalisent la même fonction qu'une seule bascule.



Les bascules maître-esclaves permettent de diminuer la sensibilité aux bruits et aux perturbations externes, en minimisant la durée d'activation de la bascule.

- sur le niveau bas de l'horloge, le maître est dans l'état « sans changement » (fermé), et l'esclave fonctionne normalement RS (ouvert) ;
- sur le niveau haut de l'horloge, le maître fonctionne normalement (ouvert), et l'esclave est dans l'état « sans changement » (fermé).

La période pendant laquelle la bascule est sensible aux bruits et aux perturbations externes se résume donc à la durée de commutation de l'horloge du niveau haut au niveau bas (front descendant).

e. Bascule D (Data)

La bascule de type D est une bascule RST pour laquelle on n'a conservé que les deux combinaisons $RS = (0,1)$ et $RS = (1,0)$. Ce qui permet éliminer l'état indéterminé et d'utiliser des entrées complémentaires. La bascule possède donc une seule entrée, nommée D '(Data).

$$D = R = \bar{S} \implies D$$

Circuit de la bascule D

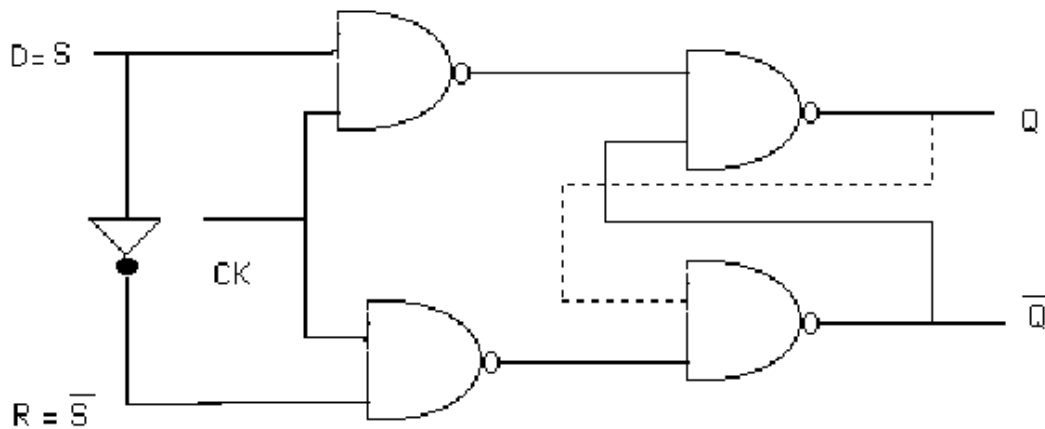
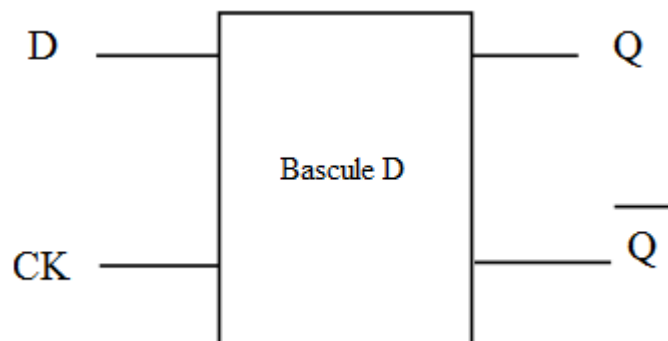
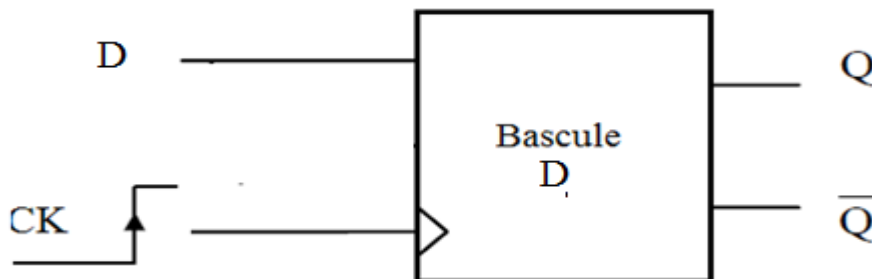


Table de vérité de la bascule D

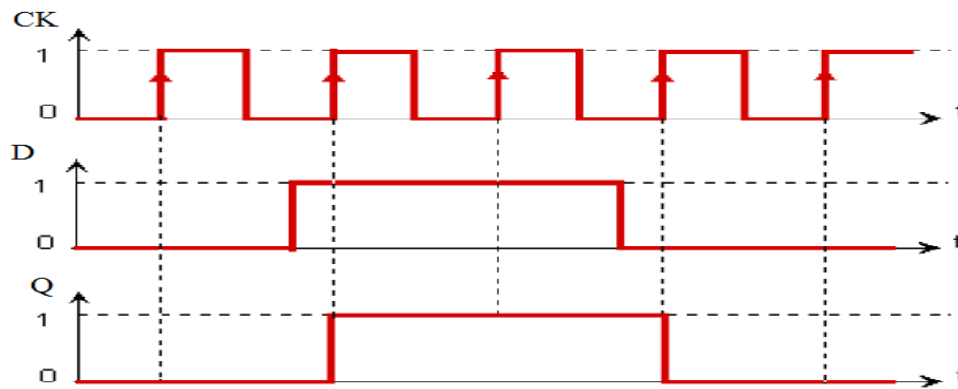
| D | CK | Q | \bar{Q} |
|---|----|--------------|--------------|
| 0 | 0 | S.C | S.C |
| 0 | 1 | 0 état reset | 1 état set |
| 1 | 0 | S.C | S.C |
| 1 | 1 | 1 état set | 0 état reset |

Symbole

La bascule D est appelée à verrouillage : détection par niveau de CK (appelée latch), dans ce cas l'entrée horloge CK est appelée entrée de validation. Si la bascule D est activée par détection de front de CK (elle est appelée flip flop).

Symbole

Chronogramme de la bascule D à front montant



Ainsi, quand CK passe de 0 \Rightarrow 1 : C'est la phase d'écriture dans la bascule.

Si D=0 \Rightarrow Q=0

Si D=1 \Rightarrow Q=1

Quand CK passe de 1 \Rightarrow 0, la bascule mémorise l'état logique enregistré. C'est la phase de mémorisation dans la bascule. Quelque soit l'état de D, la bascule mémorise l'état logique précédent de la sortie Q. Ceci montre que la bascule D matérialise le stockage d'un bit (0 ou 1). C'est pour cette raison qu'on appelle la bascule D **"un point mémoire"**. Pour mémoriser une information codée sur n bits, on utilise n bascules D. Il est à noter qu'on peut cascader deux bascules D en maître esclave pour réduire la sensibilité aux bruits et aux perturbations.

2.4. Bascule JK

La bascule JK est une variante de la bascule RS, le circuit de la bascule JK synchrone est comme suit :

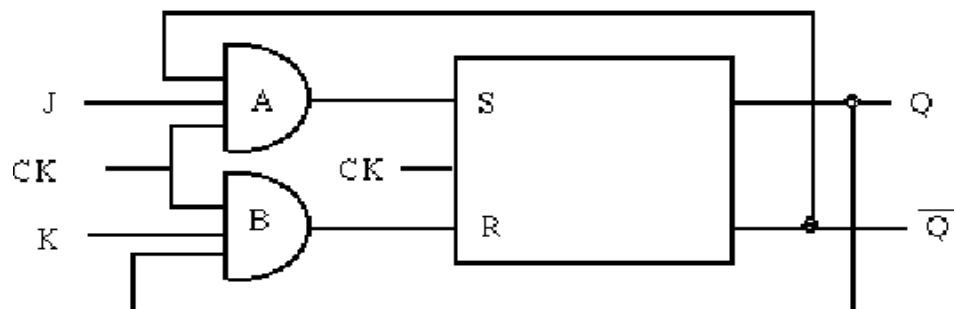


Table de vérité de la bascule JK synchrone

| J | K | CK | S | R | Q | \overline{Q} |
|---|---|----|---|---|-------------------|----------------|
| 0 | 0 | 0 | 0 | 0 | S.C | S.C |
| 0 | 0 | 1 | 0 | 0 | S.C | S.C |
| 0 | 1 | 0 | 0 | 0 | S.C | S.C |
| 0 | 1 | 1 | 0 | ? | 0 | 1 voir a |
| 1 | 0 | 0 | 0 | 0 | S.C | S.C |
| 1 | 0 | 1 | ? | 0 | 1 | 0 voir b |
| 1 | 1 | 0 | 0 | 0 | S.C | S.C |
| 1 | 1 | 1 | ? | ? | Complément voir c | |

a)

J=0, S=0, R?

- Si Q=1 alors $\bar{Q}=0 \implies R=1$ puisque CK=1, K=1Ainsi, S=0 et R=1 $\implies Q=0$ et $\bar{Q}=1$ - Si Q=0 alors $\bar{Q}=1 \implies R=0$ puisque Q = 0Ainsi, S=0 et R=0 \implies Il n'y a pas de changement des sorties Q=0 et $\bar{Q}=1$ \implies J=0 et K=1 **Etat reset****b)**

K=0, R=0, S?

- Si Q=1 alors $\bar{Q}=0 \implies S=0$ Ainsi, S=0 et R=0 \implies pas de changement des sorties. Q=1 et $\bar{Q}=0$ - Si Q=0 alors $\bar{Q}=1 \implies S=1$ puisque CK=1, J=1Ainsi, S=1 et R=0 $\implies Q=1$ et $\bar{Q}=0 \implies$ J=1 et K=0 **Etat set****c)**

J=1 et K=1, CK=1, S?, R?

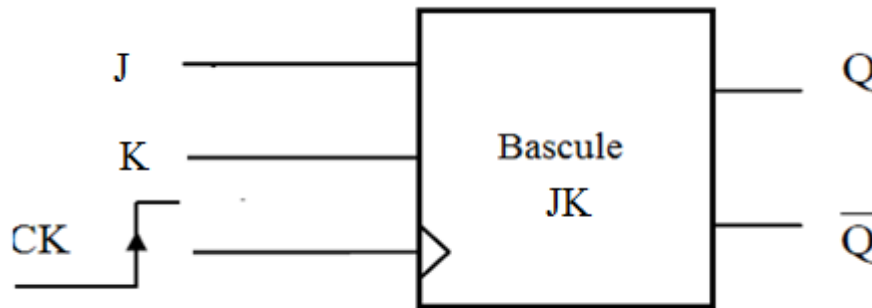
- Si Q=1 alors $\bar{Q}=0 \implies S=0$ et R=1 $\implies Q=0$ et $\bar{Q}=1$ - Si Q=0 alors $\bar{Q}=1 \implies S=1$ et R=0 $\implies Q=1$ et $\bar{Q}=0$

Pour J=1, K=1, quel que soit l'état de Q et \bar{Q} il y a une complémentation de leurs états (complément à 1 des deux sorties).

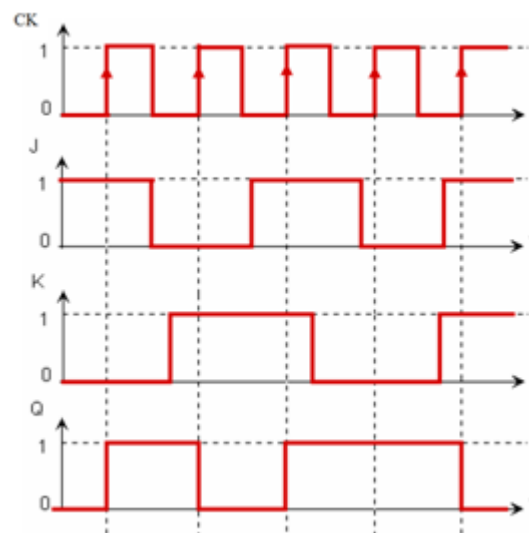
Table de vérité condensée de JK

| J | K | Q | \bar{Q} |
|---|---|--------------|-----------|
| 0 | 0 | S.C | SC |
| 1 | 0 | 1 état Set | 0 |
| 0 | 1 | 0 état Reset | 1 |
| 1 | 1 | C à 1. | C à 1 |

Symbole



Chronogramme de la bascule JK à front montant



Les états de J et K qui entraînent un changement de la sortie Q sur front montant :

$K = 1, J = 0$: mise à zéro de Q

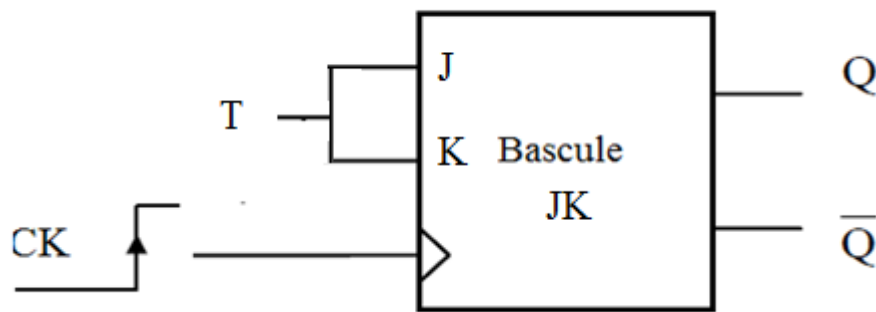
$K = 0, J = 1$: mise à un de Q

$K = J = 0$: mémorisation de Q

$K = J = 1$: complément à 1 diviseur par 2 " état bascule "

2.5. Bascule T

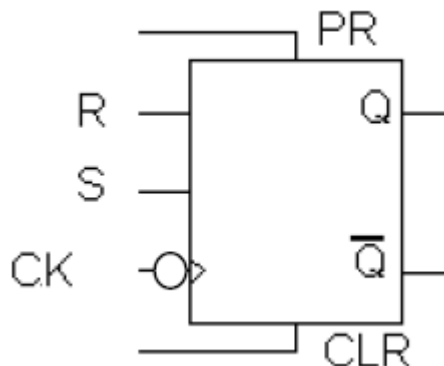
La bascule T (Toggle ou trigger) s'obtient par exemple à partir d'une bascule JK en reliant les entrées J et K entre elles. Elle est utilisable uniquement en mode synchrone.



2.6. Initialisation des bascules

On note la présence de deux entrées PR (Preset) et CLR (Clear) qui imposent respectivement $Q=1$ et $Q=0$ lorsqu'elles sont actives (positionnées à 1). Ces deux entrées sont prioritaires sur toutes les autres entrées. Selon les bascules, ces deux entrées peuvent de nature asynchrone, leur effet est "immédiat", ou synchrone, leur action sur les sorties est fonction d'une impulsion d'horloge.

Exemple :



2.7. Domaine d'application des bascules dans une machine informatique

Dans les microprocesseurs, interfaces et les mémoires vives statiques, on trouve des entités appelées **Registres et Compteurs dont la conception est fondée sur les bascules.**

a. Registre

Un registre est un circuit constitué de n bascules synchronisées permettant de stocker temporairement d'une information binaire codée sur n bits en vue de son transfert dans un autre circuit (pour traitement, affichage, mémorisation, etc.)

Le schéma d'un tel système comporte autant de bascules (de type D) que de bits binaires à mémoriser. Toutes les bascules sont commandées par le même signal d'horloge.

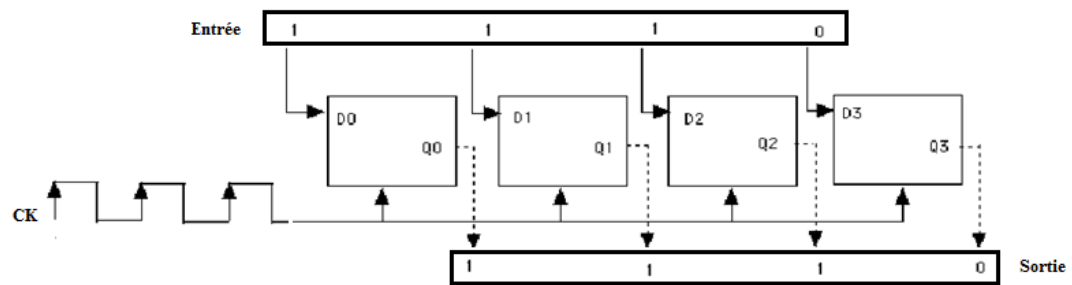
| | | | | | | | | | | |
|-------------|-------------|------|------|------|------|------|------|------|-----------|-----------|
| bn-1 | bn-2 | | | | | | | | b1 | b0 |
|-------------|-------------|------|------|------|------|------|------|------|-----------|-----------|

Chaque bit est matérialisé par une bascule,

En pratique, il existe plusieurs façons de réaliser un registre. Le mode d'utilisation d'un registre est très varié, il peut être utilisé comme un simple moyen de stockage, ou dans les calculs arithmétiques, ou bien dans les conversions parallèle-série ou série-parallèle des informations (registres à décalage), ...etc.

- **Registre à entrée parallèle sortie parallèle : Registre de stockage**

Exemple : Registre de stockage de 4 bits (4 Bascules D)



A $t = T$, $D0 = 1$, $D1 = 1$, $D2 = 1$, $D3 = 0 \Rightarrow Q0 = 1$, $Q1 = 1$, $Q2 = 1$, $Q3 = 0$

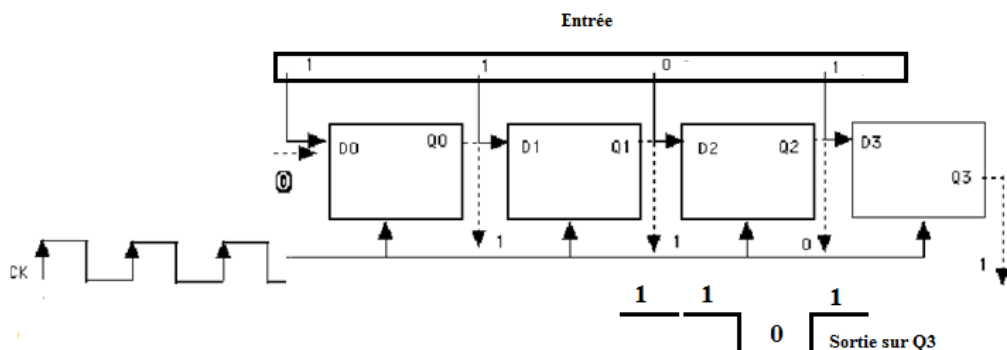
Fonctionnement.

Une information se présente aux niveaux des entrées $D0...D3$, à ce moment le signal d'horloge CK est positionné à 1 \Rightarrow mémorisation de l'information binaire $D0...D3$.

Quand CK passe à 0 \Rightarrow même si l'information disparaît au niveau des entrées, il restera mémoriser au niveau des bascules D.

- **Registre à entrée parallèle sortie série : Registre de transfert de données, utilisé comme convertisseur parallèle-série, il est nécessaire à l'émission lors d'une transmission série**

Exemple : Registre de stockage de 4 bits (4 Bascules D)



Ecriture de l'information en parallèle:

$D0 = 1$, $D1 = 1$, $D2 = 0$, $D3 = 1 \Rightarrow Q0 = 1$, $Q1 = 1$, $Q2 = 0$, $Q3 = 1$.

L'information restera mémorisée.

à $t = T$,

Avant l'activation de CK

$D0 = 0$ (forcée)

$D1 = Q0 = 1$

$D2 = Q1 = 1$

et $D3 = Q2 = 0$

Après l'activation de CK

$\Rightarrow Q0=0, Q1=1, Q2=1$ et $Q3=0 \Rightarrow$ Sortie du premier bit : "1"

à $t = 2T$,

Avant l'activation de CK

$D0 = 0$

$D1 = Q0 = 0$

$D2 = Q1 = 1$

et $D3 = Q2 = 1$

Après l'activation de CK

$\Rightarrow Q0 = 0, Q1 = 0, Q2 = 1$ et $Q3 = 1 \Rightarrow$ Sortie du deuxième bit: "0"

à $t = 3T$,

Avant l'activation de CK

$D0 = 0$

$D1 = Q0 = 0$

$D2 = Q1 = 0$

et $D3 = Q2 = 1$

Après l'activation de CK

$\Rightarrow Q0 = 0, Q1 = 0, Q2 = 0$ et $Q3 = 1 \Rightarrow$ Sortie du troisième bit: "1"

à $t = 4T$,

Avant l'activation de CK

$D0 = 0$

$D1 = Q0 = 0$

$D2 = Q1 = 0$

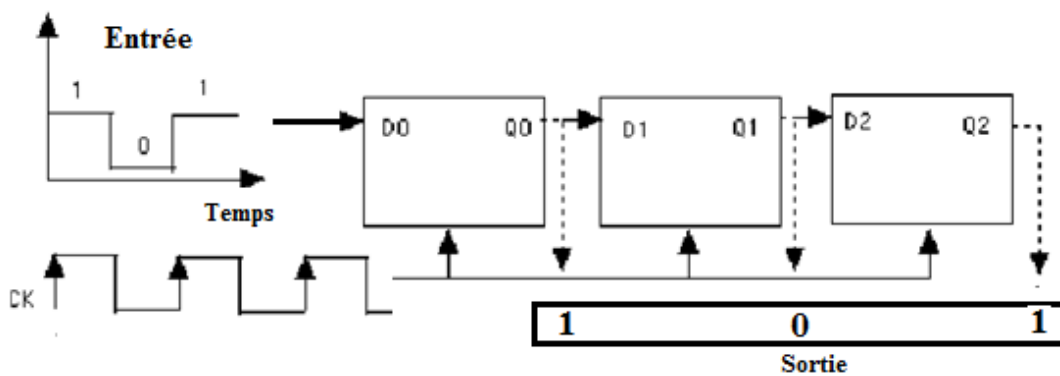
et $D3 = Q2 = 0$

Après l'activation de CK

$\Rightarrow Q0=0, Q1=0, Q2=0$ et $Q3=0 \Rightarrow$ Sortie du quatrième bit : "1".

- **Registre à entrée série sortie parallèle :** Registre de transfert de données, utilisé comme convertisseur parallèle-série, il est nécessaire à la réception de données lors d'une transmission série.

Exemple : Registre de stockage de 3 bits (3 Bascules D)



à $t = 0$, $Q0 = 0$, $Q1 = 0$, $Q2 = 0$

à $t = T$

$D0 = 1, \Rightarrow Q0 = 1$ et $Q1 = 0$, $Q2 = 0$

$D1 = Q0 = 0$

$D2 = Q1 = 0$

à $t = 2T$

$D0 = 0$,

$D1 = Q0 = 1 \Rightarrow Q0 = 0$ et $Q1 = 1$, $Q2 = 0$

$D2 = Q1 = 0$

à $t = 3T$

$D0 = 1$,

$D1 = Q0 = 0$

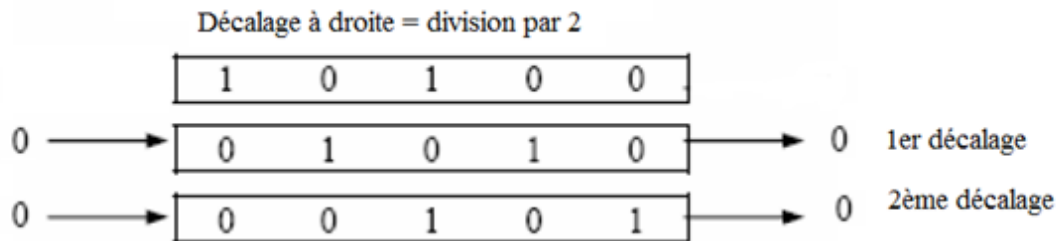
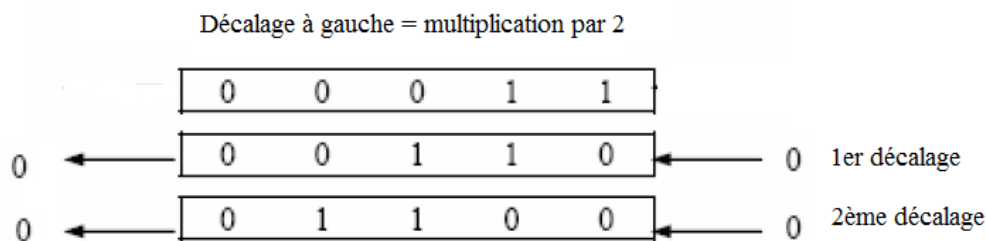
$D2 = Q1 = 1 \Rightarrow Q0 = 1$ et $Q1 = 0$, $Q2 = 1$

• Registres à décalage

Comme son nom l'indique, un registre à décalage consiste à décaler bit par bit une information binaire soit vers la gauche, soit vers la droite. Le registre à décalage peut être à écriture et à lecture série ou parallèle.

• Autre application : Multiplication / Division en binaire pur:

La fonction décalage par insertion de 0 peut être utilisée pour réaliser les multiplications ou des divisions par 2.

Exemple : Registre à décalage à droite de 5 bits**Exemple : Registre à décalage à gauche de 5 bits**

En général, x décalages à droite = à une division par 2^x , et x décalages à gauche = à une multiplication par 2^x .

- **Registres et mémoires**

Les registres sont utilisés pour concevoir des mémoires vives statiques SRAM (mémoire cache des microprocesseur, mémoire de données dans les microcontrôleurs)

- **Registres et microprocesseurs**

Les microprocesseurs utilisent les registres comme moyen de stockage d'informations (données, instructions) qui vont être traitées soit par l'UAL (Unité Arithmétique et Logique) ou décodées par la logique de commande et de contrôle, plusieurs types de registres existent à l'intérieur d'un microprocesseur:

Accumulateurs, registres opérationnel, registre d'état, registres d'adresses, registres d'indexation, registres d'instructions, pointeur de pile...etc.

- **Registres et interfaces d'Entrées/Sorties**

Les registres typiques sont:

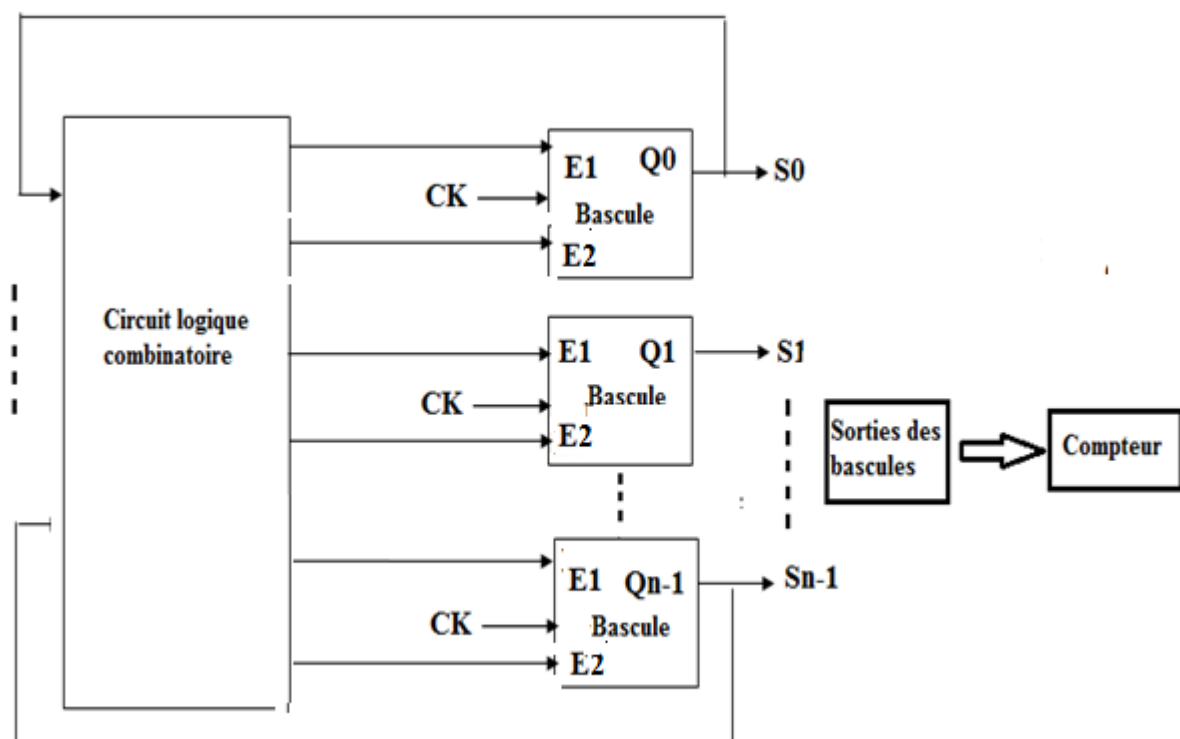
- registre d'état pour le fonctionnement de l'interface
- registre de contrôle pour la programmation de l'interface
- registre de direction pour la définition du sens de transfert de données.

b) Compteurs

Pour réaliser des opérations de comptage, en utilisant un code binaire donnée (code binaire pur, complément à deux, code réfléchi...etc), de temporisation, comptage d'impulsion d'horloge, d'itérations d'une boucle...etc. On utilise des circuits logiques appelés compteurs.

Un compteur est un circuit séquentiel comportant n bascules (D, JK, ..etc). Les sorties de ces bascules sont les entrées des circuits logiques combinatoires utilisés pour assurer les transitions (basculement) nécessaires au codage choisi pour le compteur. Il fonctionne au rythme d'une horloge et avec un cycle de comptage régulier ou quelconque d'un maximum de 2^n combinaisons.

Structure d'un compteur



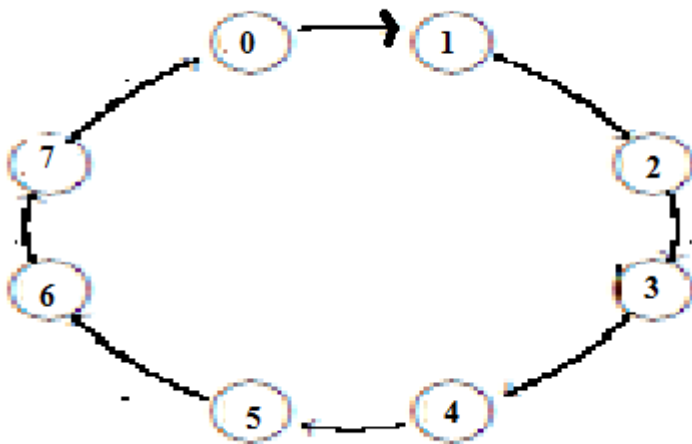
Définition: la combinaison des n sorties (n bascules) d'un compteur est appelée état, et le nombre d'états possible est appelé modulo.

Le Modulo est le nombre d'état N des combinaisons successives par le compteur pendant un cycle.

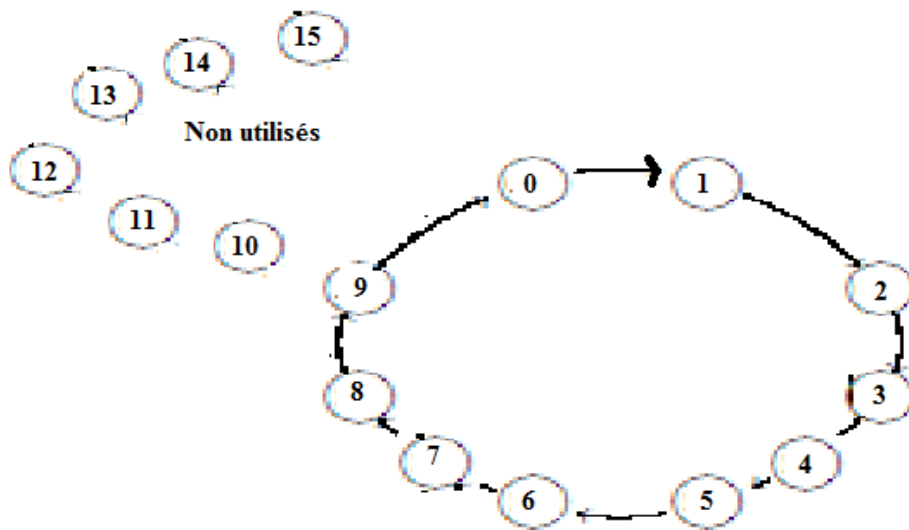
Le nombre d'état $N \leq 2^n$. Le modulo maximal d'un compteur à n bits (n bascules) est $N = 2^n$ est le cycle sera complet.. Si $N < 2^n$ un certain nombre d'états ne sont jamais utilisés et le cycle sera incomplet.

Par exemple, un compteur 3 bits possède 8 états distincts, c'est donc un compteur modulo 8.

Il effectue donc un cycle complet ($N = 2^n$).



Pour un compteur 4 bits, il possèdera 16 états distincts, c'est donc un compteur modulo 16. Si on fait fonctionner ce compteur comme modulo 10, il aura donc 10 états possibles et 6 états ne seront jamais utilisés. Il sera remis à zéro quand il atteindra l'état 9. Il effectuera donc un cycle incomplet ($N < 2^n$).



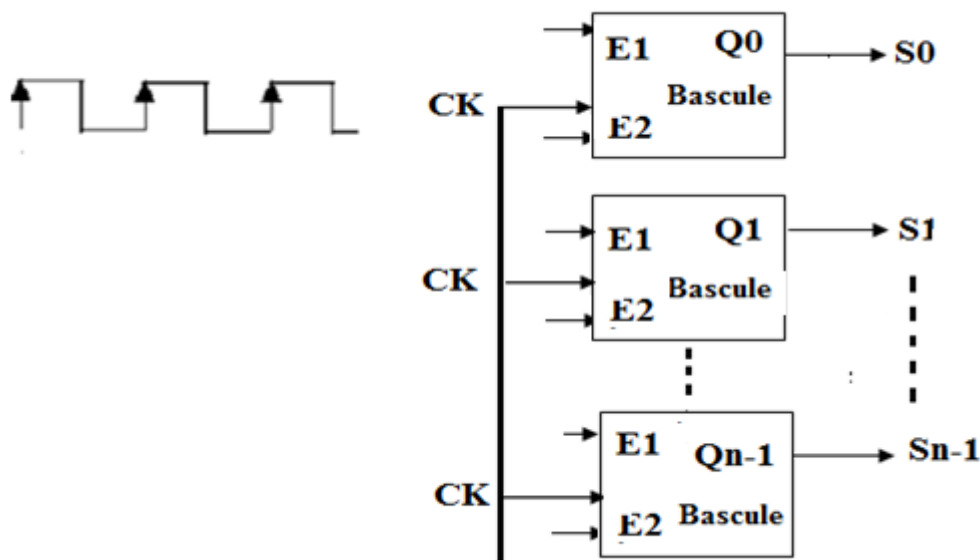
Types de compteurs:

Les compteurs binaires peuvent être classés en deux catégories :

- Compteurs asynchrones;
- Compteurs synchrones.

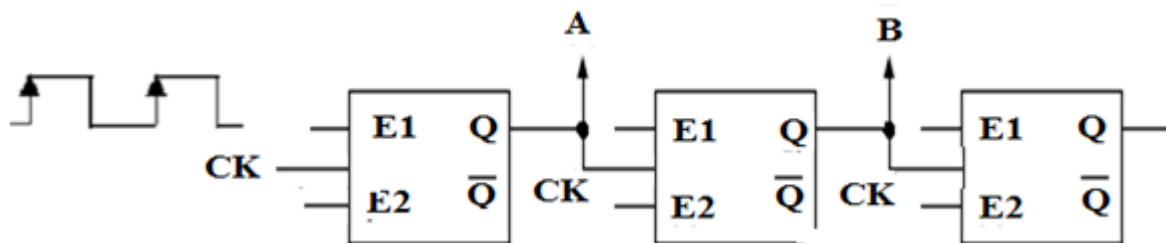
On distingue également les compteurs réversibles ou compteurs-décompteurs.

Compteurs Synchrones : Toutes les bascules sont attaquées simultanément par une seule Horloge et par conséquent, elles changent d'état presque en même temps.



Compteurs asynchrones : Dans ce type de compteur, le signal d'horloge d'une bascule sera issu d'une combinaison logique des sorties des bascules précédentes.

Exemple :



Dans cet exemple, la sortie de la première bascule est l'horloge de la deuxième, l'horloge de la troisième bascule est la sortie de la deuxième bascule etc.

Synthèse des compteurs

La réalisation est possible avec des bascules JK, D ou T. Pour faire cette synthèse, il faudra définir pour chaque bascule:

1^{er} point: Le signal d'horloge qui sera appliqué aux bascules.

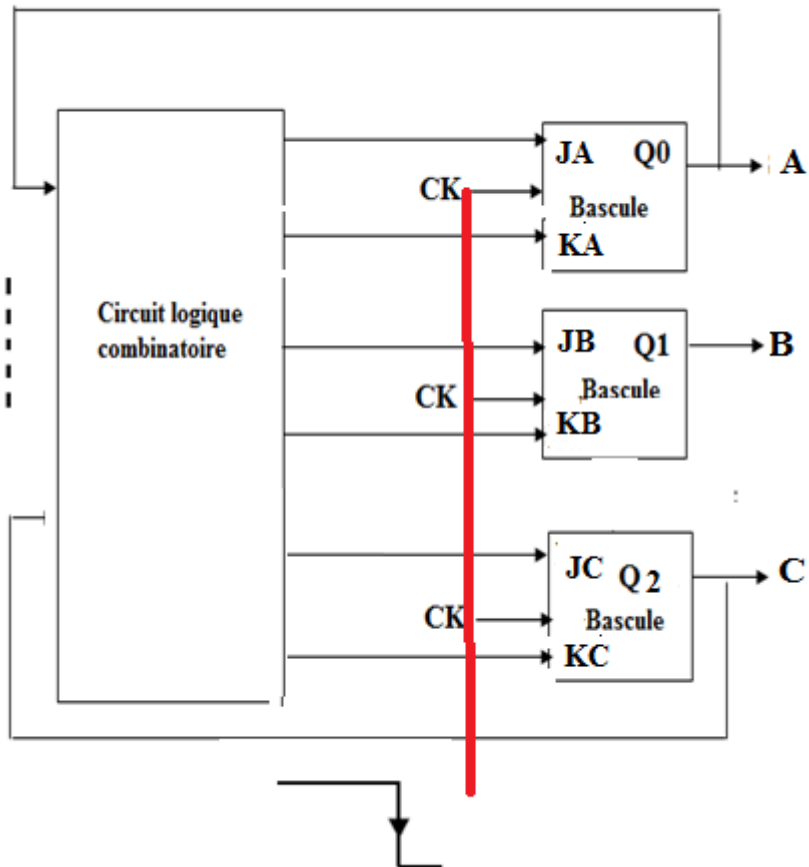
2^{ème} point: Les fonctions logiques à câbler pour réaliser les entrées des bascules.

a) Réalisation d'un compteur synchrone

Exemple 1: Compteur synchrone modulo $8 = 2^3$ 3 bits, donc 3 bascules (1 2 4), il effectuera un cycle complet de 8 états et il comptera en binaire pur

- Bascules choisies: **JK à front descendant**
- Code choisi: binaire pur
- Choix du signal d'horloge : Toutes les bascules reçoivent le même signal d'Horloge.

Il faudra calculer les états logiques des entrées JA, KA et JB, KB et JC, KC



Rappel

Table de vérité condensée de JK.

| J | K | Q | \bar{Q} |
|---|---|--------------|----------------|
| 0 | 0 | S.C | S.C |
| 1 | 0 | 1 | 0 Set |
| 0 | 1 | 0 | 1 Reset |
| 1 | 1 | C à 1 | C à 1 |

Table de vérité

| n | Sorties du compteur | | | Entrées des bascules | | | | | |
|---|---------------------|---|---|----------------------|----|----|----|----|----|
| | C | B | A | JA | KA | JB | KB | JC | KC |
| 0 | 0 | 0 | 0 | 1 | x | 0 | x | 0 | x |
| 1 | 0 | 0 | 1 | x | 1 | 1 | x | 0 | x |
| 2 | 0 | 1 | 0 | 1 | x | x | 0 | 0 | x |
| 3 | 0 | 1 | 1 | x | 1 | x | 1 | 1 | x |
| 4 | 1 | 0 | 0 | 1 | x | 0 | x | x | 0 |
| 5 | 1 | 0 | 1 | x | 1 | 1 | x | x | 0 |
| 6 | 1 | 1 | 0 | 1 | x | x | 0 | x | 0 |
| 7 | 1 | 1 | 1 | x | 1 | x | 1 | x | 1 |
| 0 | 0 | 0 | 0 | | | | | | |

Diagrammes de Karnaugh.

$JA = KA = 1$ (x remplacée par 1)

JB

| C/BA | 00 | 01 | 11 | 10 |
|------|----|----|-------|----|
| 0 | 0 | 1 | x = 1 | x |
| 1 | 0 | 1 | x = 1 | x |

=====> JB = A

KB

| C/BA | 00 | 01 | 11 | 10 |
|------|----|-------|----|----|
| 0 | x | x = 1 | 1 | 0 |
| 1 | 0 | x = 1 | 1 | 0 |

=====> KB = A

JC

| C/BA | 00 | 01 | 11 | 10 |
|------|----|----|-------|----|
| 0 | 0 | 0 | 1 | 0 |
| 1 | x | x | x = 1 | x |

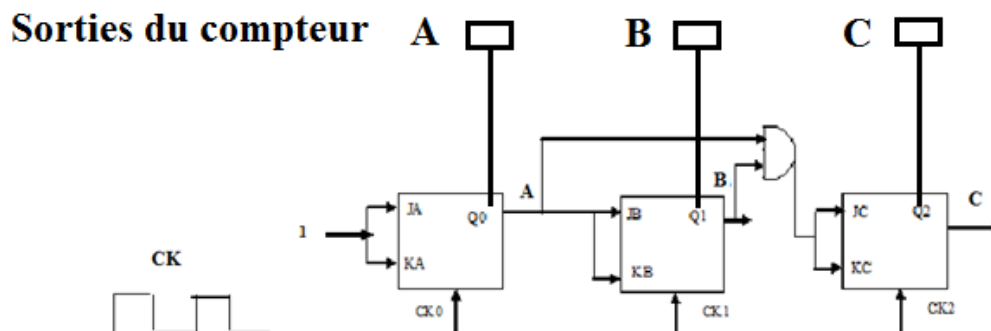
=====> JC = AB

KC

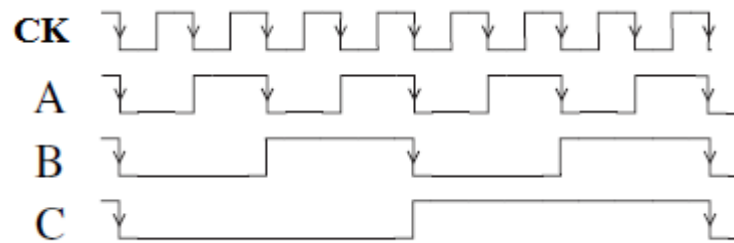
| C/BA | 00 | 01 | 11 | 10 |
|------|----|----|-------|----|
| 0 | x | x | x = 1 | x |
| 1 | 0 | 0 | 1 | 0 |

=====> KC = A.B

Circuit



Chronogramme



Exemple. 2: Synthèse d'un compteur synchrone modulo 10 (appelé également une décade), qui compte en binaire pur avec des bascules JK à front descendant, $2^3 < 10 < 2^4 \Rightarrow 4 \text{ bits} \Rightarrow 4$ bascules (1 2 4 8).

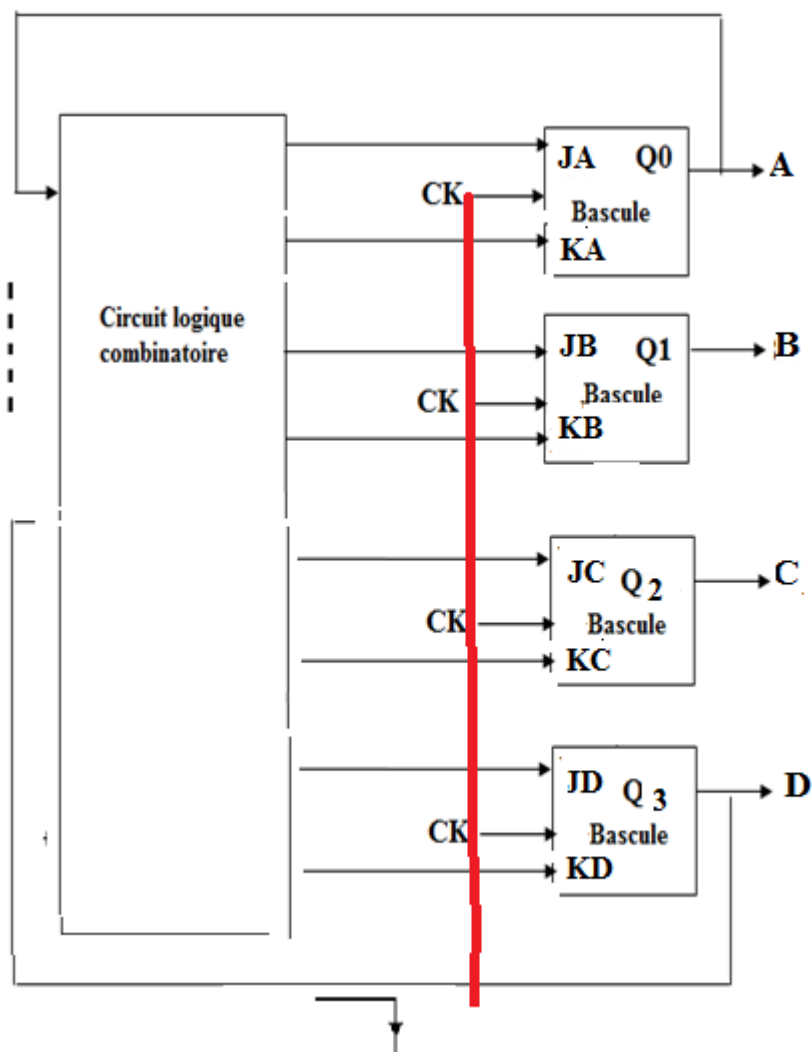


Table de vérité

| n | Sorties du compteur | | | | Entrées des bascules | | | | | | | |
|----|---------------------|---|---|---|----------------------|----|----|----|----|----|----|----|
| | D | C | B | A | JA | KA | JB | KB | JC | KC | JD | KD |
| 0 | 0 | 0 | 0 | 0 | 1 | x | 0 | x | 0 | x | 0 | x |
| 1 | 0 | 0 | 0 | 1 | x | 1 | 1 | x | 0 | x | 0 | x |
| 2 | 0 | 0 | 1 | 0 | 1 | x | x | 0 | 0 | x | 0 | x |
| 3 | 0 | 0 | 1 | 1 | x | 1 | x | 1 | 1 | x | 0 | x |
| 4 | 0 | 1 | 0 | 0 | 1 | x | 0 | x | x | 0 | 0 | x |
| 5 | 0 | 1 | 0 | 1 | x | 1 | 1 | x | x | 0 | 0 | x |
| 6 | 0 | 1 | 1 | 0 | 1 | x | x | 0 | x | 0 | 0 | x |
| 7 | 0 | 1 | 1 | 1 | x | 1 | x | 1 | x | 1 | 1 | x |
| 8 | 1 | 0 | 0 | 0 | 1 | x | 0 | x | 0 | x | x | 0 |
| 9 | 1 | 0 | 0 | 1 | x | 1 | 0 | x | 0 | x | x | 1 |
| 10 | 0 | 0 | 0 | 0 | | | | | | | | |

$$\implies \text{JA} = \text{KA} = 1 \quad (x = 1)$$

JB

| DC/BA | 00 | 01 | 11 | 10 |
|-------|----|----|-------|----|
| 00 | 0 | 1 | x = 1 | x |
| 01 | 0 | 1 | x = 1 | x |
| 11 | x | x | x | x |
| 10 | 0 | 0 | x | x |

$$\implies \text{JB} = \overline{\text{AD}}$$

KB

| DC/BA | 00 | 01 | 11 | 10 |
|-------|----|-------|-------|----|
| 00 | x | x = 1 | 1 | 0 |
| 01 | x | x = 1 | 1 | 0 |
| 11 | x | x = 1 | x = 1 | x |
| 10 | x | x = 1 | x = 1 | x |

$$\implies \text{KB} = \text{A}$$

JC

| DC/BA | 00 | 01 | 11 | 10 |
|-------|----|----|-------|----|
| 00 | 0 | 0 | 1 | 0 |
| 01 | x | x | x = 1 | x |
| 11 | x | x | x = 1 | x |
| 10 | 0 | 0 | x = 1 | x |

$$\implies \text{JC} = \text{AB}$$

KC

| DC/BA | 00 | 01 | 11 | 10 |
|-------|----|----|-------|----|
| 00 | x | x | x = 1 | x |
| 01 | 0 | 0 | 1 | 0 |

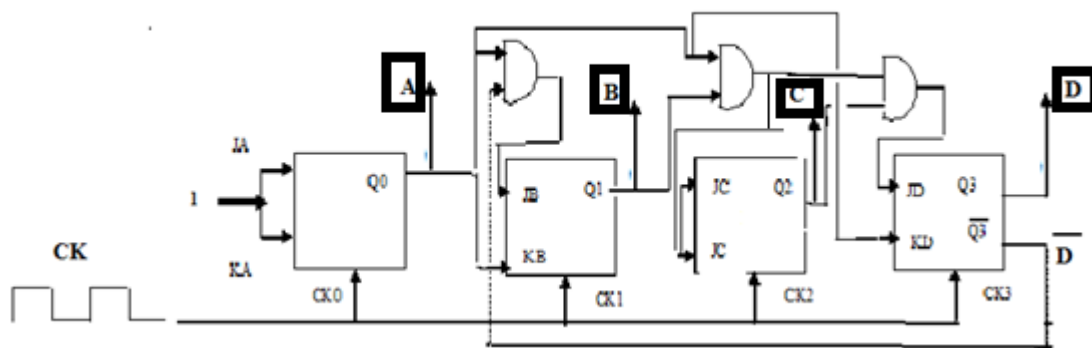
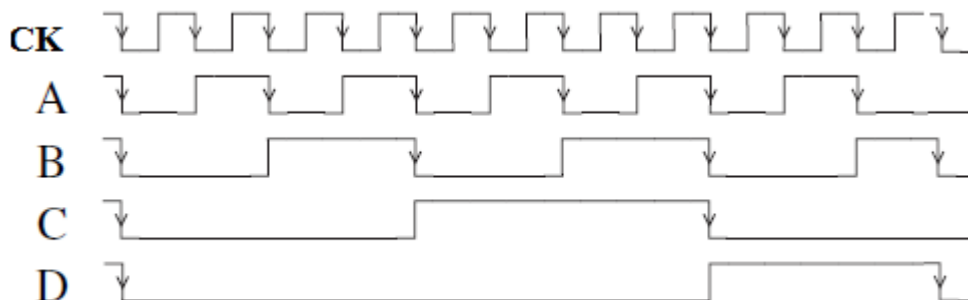
| | | | | |
|----|---|---|--------------|---|
| 11 | x | x | x = 1 | x |
| 10 | x | x | x = 1 | x |

=====> KC = AB

JD

| DC/BA | 00 | 01 | 11 | 10 |
|-------|----|----|--------------|----|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 0 |
| 11 | x | x | x = 1 | x |
| 10 | x | x | x | x |

=====> JD = ABC et KD = A

Circuit**Chronogramme****b) Réalisation d'un compteur asynchrone**

Exemple 1: Compteur asynchrone modulo $8 = 2^3$ 3 bits, donc 3 bascules (**1 2 4**), il effectuera un cycle complet de 8 états et il comptera en binaire pur

- Bascules choisies: **JK à front descendant**
- Code choisi: binaire pur
- Choix du signal d'horloge : Trouver le signal d'Horloge pour chaque bascule.

Il faudra également calculer les états logiques des entrées JA, KA et JB, KB

| n | Sorties du compteur | | |
|---|---------------------|---|---|
| | C | B | A |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |

Recherche des signaux d'horloge

- La bascule A change d'état cadencé par le signal d'Horloge CK0 = CK.
- Pour la bascule B change d'état chaque fois que la sortie de la bascule A passe de 1 à 0 descendant) CK1 = A.
- Pour la bascule C change d'état quand B passe de 1 à 0 \Rightarrow CK2 = B

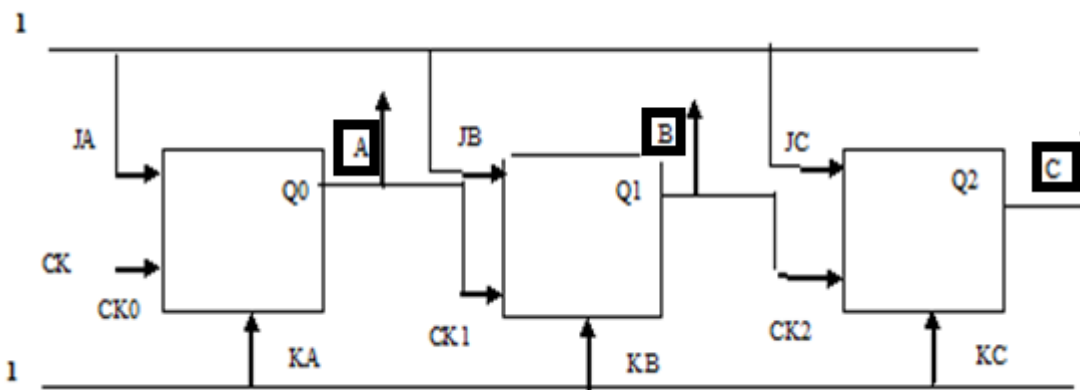
Table de vérité

| n | Sorties du compteur | | | Entrées des bascules | | | | | |
|---|---------------------|---|---|----------------------|----|----|----|----|----|
| | C | B | A | JA | KA | JB | KB | JC | KC |
| 0 | 0 | 0 | 0 | 1 | x | x | x | x | x |
| 1 | 0 | 0 | 1 | x | 1 | 1 | x | x | x |
| 2 | 0 | 1 | 0 | 1 | x | x | x | x | x |
| 3 | 0 | 1 | 1 | x | 1 | x | 1 | 1 | x |
| 4 | 1 | 0 | 0 | 1 | x | x | x | x | x |
| 5 | 1 | 0 | 1 | x | 1 | 1 | x | x | x |
| 6 | 1 | 1 | 0 | 1 | x | x | x | x | x |
| 7 | 1 | 1 | 1 | x | 1 | x | 1 | x | 1 |
| 0 | 0 | 0 | 0 | | | | | | |

Remarque : JB = KB = x, entrées quelconques puisque l'horloge est à front montant.

$$JA = KA = JB = KB = JC = KC = 1$$

Circuit



Exemple. 2: Synthèse d'un compteur asynchrone modulo 10 (appelé également une décade), qui compte en binaire pur avec des bascules JK à front descendant, $2^3 < 10 < 2^4 \Rightarrow 4 \text{ bits} \Rightarrow 4$ bascules (1 2 4 8).

| n | Sorties du compteur | | | |
|---|---------------------|---|---|---|
| | D | C | B | A |
| 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 |
| 0 | 0 | 0 | 0 | 0 |

Recherche des signaux d'horloge

- Bascule A CK0 = CK
- Bascule B change d'état à chaque front descendant de A \implies CK1 = A
- Bascule C change d'état à chaque front descendant de B \implies CK2 = B
- Bascule D la sortie A peut jouer le rôle d'un signal horloge avec précaution \implies CK3 = A

Table de vérité

| n | Sorties du compteur | | | | Entrées des bascules | | | | | | | |
|---|---------------------|---|---|---|----------------------|----|----|----|----|----|----|----|
| | D | C | B | A | JA | KA | JB | KB | JC | KC | JD | KD |
| 0 | 0 | 0 | 0 | 0 | 1 | x | x | x | x | x | x | x |
| 1 | 0 | 0 | 0 | 1 | x | 1 | 1 | x | x | x | 0 | x |
| 2 | 0 | 0 | 1 | 0 | 1 | x | x | x | x | x | x | x |
| 3 | 0 | 0 | 1 | 1 | x | 1 | x | 1 | 1 | x | 0 | x |
| 4 | 0 | 1 | 0 | 0 | 1 | x | x | x | x | x | x | x |
| 5 | 0 | 1 | 0 | 1 | x | 1 | 1 | x | x | x | 0 | x |
| 6 | 0 | 1 | 1 | 0 | 1 | x | x | x | x | x | x | x |
| 7 | 0 | 1 | 1 | 1 | x | 1 | x | 1 | x | 1 | 1 | x |
| 8 | 1 | 0 | 0 | 0 | 1 | x | x | x | x | x | x | x |
| 9 | 1 | 0 | 0 | 1 | x | 1 | 0 | x | x | x | x | 1 |
| 0 | 0 | 0 | 0 | 0 | | | | | | | | |

$$JA = KA = 1, KB = 1, JC = KC = 1, KD = 1$$

JB

| DC/BA | 00 | 01 | 11 | 10 |
|-------|-------|----|-------|-------|
| 00 | x = 1 | 1 | x = 1 | x = 1 |
| 01 | x = 1 | 1 | x = 1 | x = 1 |
| 11 | x | x | x | x |
| 10 | x | 0 | x | x |

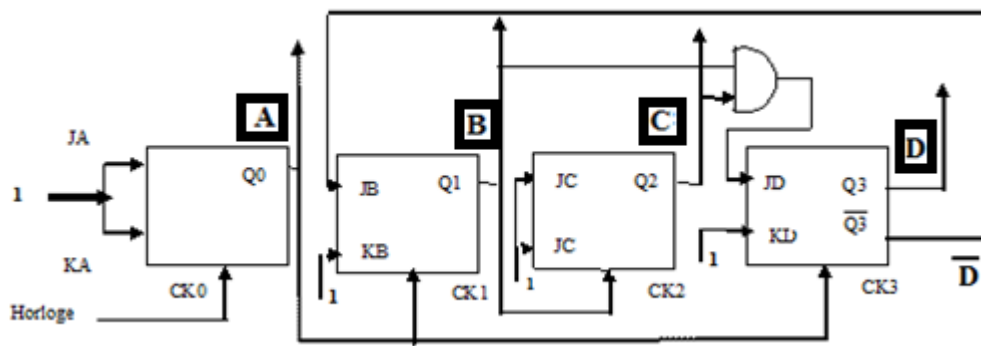
$$\implies JB = \overline{D}$$

JD

| DC/BA | 00 | 01 | 11 | 10 |
|-------|----|----|-------|-------|
| 00 | x | 0 | 0 | x |
| 01 | x | 0 | 1 | x = 1 |
| 11 | x | x | x = 1 | x = 1 |
| 10 | x | x | x | x |

$$\implies JD = BC.$$

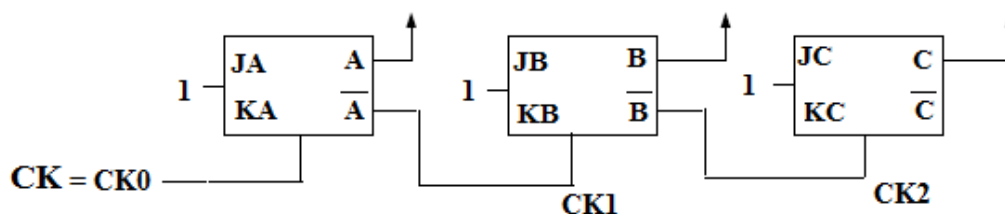
Circuit



c) Décompteurs

On peut réaliser des décompteurs Par exemple, dans le cas d'un décompteur modulo 8, Il suffit de piloter chaque entrée CK des bascules au moyen de la sortie complémentée de la bascule précédente.

Exemple : Décompteur asynchrone modulo 10



Remarque

D'autres compteurs sont également très utiles (Compteur en anneau, Compteur de Johnson,.....etc).

d) Cycle régulier et irrégulier

Cycle régulier: c'est un comptage dans l'ordre naturel par exemple 0, 1, 2, 3, 4, 5,....etc

Cycle irrégulier: c'est un comptage dans un ordre quelconque par exemple 2, 5, 6, 4, 10...etc

e) Compteurs et microprocesseur

Ils sont utilisés pour compter le nombre d'itérations d'une boucle, adresser les cases mémoire où sont stockées les instructions d'un programme) (compteur ordinal ou pointeur d'instructions)... etc.