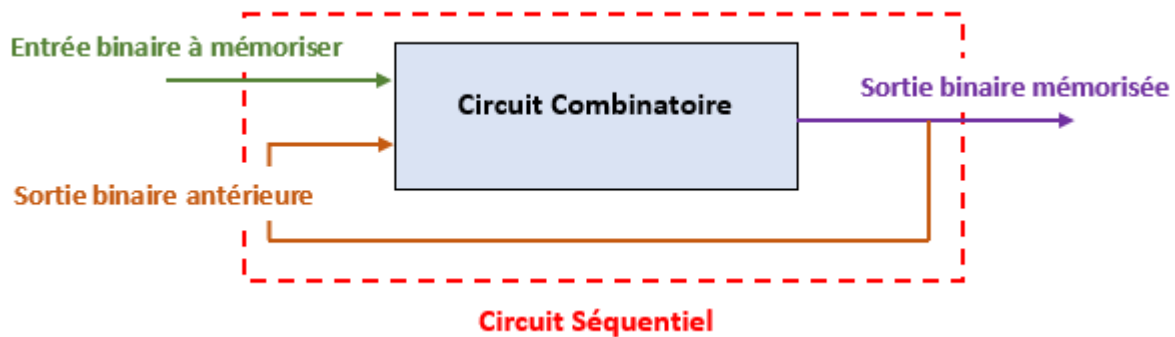


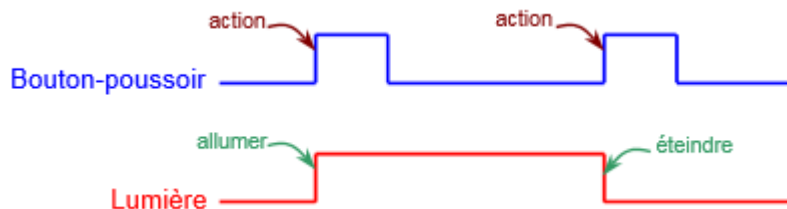
Résumé du cours

1) Introduction :

La différence essentielle entre les systèmes combinatoires que nous avons étudié dans le cours précédent et les systèmes séquentiels que nous allons aborder dans ce module, réside dans le fait que la fonction de sortie de ces derniers systèmes dépend à la fois des variables d'entrée et de l'état antérieur des sorties.



A titre d'illustration, prenons l'exemple très simple d'une lampe électrique commandée par un bouton poussoir de telle façon qu'une action sur le bouton allume la lampe, et qu'une action successive l'éteigne, selon l'exemple du chronogramme suivant :



On note bien que l'évolution du système dépend non seulement de la position du bouton poussoir à un instant donné, mais aussi du fait que la lampe soit allumée ou non. Le système dépend donc de l'état précédent car il conserve la mémoire de l'action précédente ; c'est la caractéristique essentielle d'un système séquentiel.

Dans la suite du cours, nous allons étudier le composant unitaire d'un système séquentiel qu'on appelle bascule.

Les bascules sont des circuits à deux états stables (ce qui les fait encore appeler bistables). Ce sont des circuits qui sont utilisés pour mémoriser l'état de signaux d'entrée, états qui sont alors disponibles en permanence à la sortie des bascules. C'est l'application d'un signal de commande qui permet de mémoriser l'état des signaux présents à l'entrée des bascules à un instant donné. Il existe deux types des bascules : les bascules asynchrones (bascule RS) et les bascules synchrones (bascules RSH, JK, D et T).

2) Les bascules asynchrones :

Ces bascules sont non asservies (ne sont pas contrôlées) par un signal d'horloge c'est-à-dire le changement d'état de la sortie se produit au moment où la combinaison des valeurs d'entrée est changée. Nous allons étudier l'exemple de la bascule RS.

La bascule RS :

Cette bascule, comme l'indique son nom, présente 2 entrées logiques (R pour RESET c'est-à-dire mise à 0 et S pour SET qui signifie mise à 1 et deux sorties logiques complémentaires (Q : sortie principale et sa complémentaire \bar{Q}).

Le fonctionnement de la bascule RS est décrit par la table de vérité réduite suivante :

R	S	Q_{n+1}	\bar{Q}_{n+1}	Signification
0	0	Q_n	\bar{Q}_n	Mémorisation
0	1	1	0	Mise à 1
1	0	0	1	Mise à 0
1	1	—	—	Etat interdit

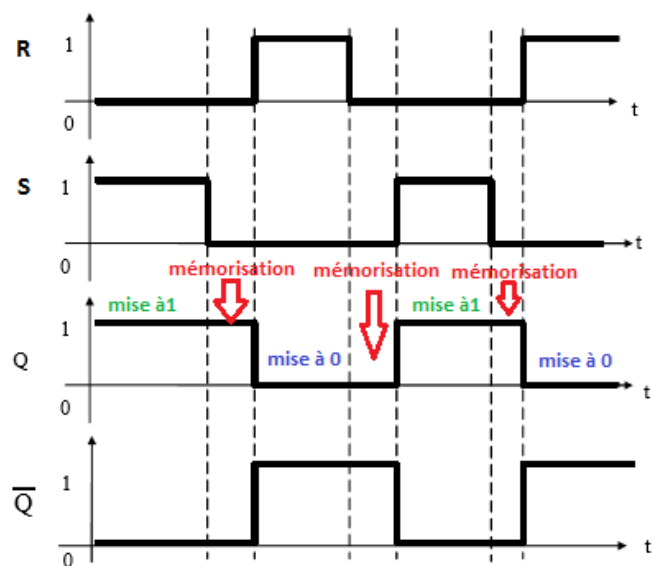
Equation logique :

$$Q = S + Q_n \cdot \bar{R} \rightarrow \bar{Q} = \overline{S + Q_n \cdot \bar{R}} = \bar{S} \cdot \overline{Q_n \cdot \bar{R}}$$

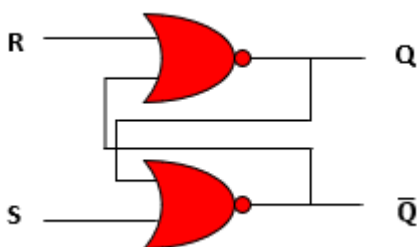
$$\bar{Q} = S \downarrow (\bar{Q}_n \downarrow R) : \text{technologie NOR}$$

$$Q = \bar{\bar{Q}} = \overline{\bar{S} \cdot \overline{Q_n \cdot \bar{R}}} = \bar{S} \cdot \overline{Q_n \cdot \bar{R}}$$

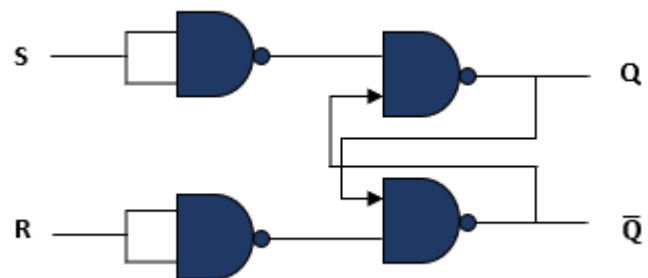
$$Q = (S|S)|((R|R)|Q) : \text{technologie NAND}$$



La bascule RS peut être représentée par deux technologies différentes : technologie NOR et technologie NAND.



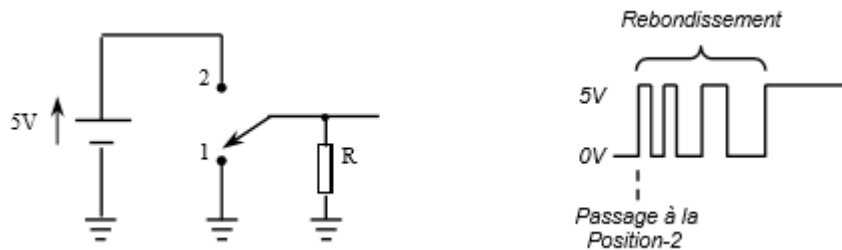
RS avec la technologie NOR



RS avec la technologie NAND

Exemple d'application d'une bascule RS : interrupteur anti-rebonds

Une application très utile de la bascule RS est l'interrupteur sans rebonds. Il est pratiquement impossible de réaliser un interrupteur mécanique dans lequel il n'y aurait qu'une seule transition entre deux tensions, à cause du phénomène de rebondissement des contacts. Ce phénomène est illustré par la figure suivante :



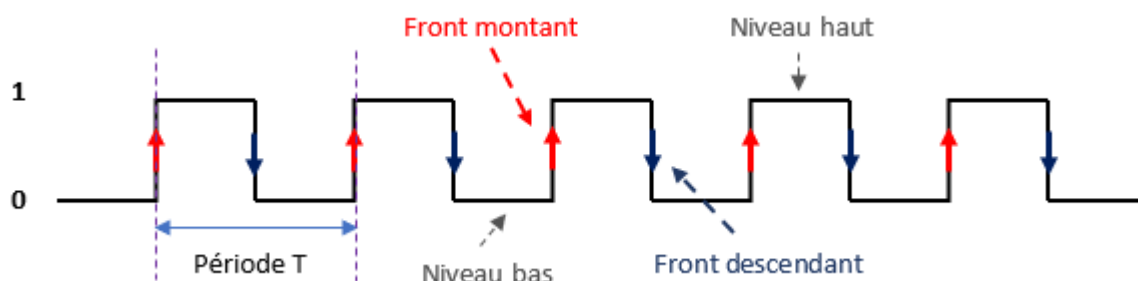
Les effets du rebondissement qui se produisent lors de la fermeture des contacts d'un interrupteur, ou d'un relais, peuvent être éliminés en utilisant une bascule RS, comme le montre le montage suivant :



3) Les bascules synchrones :

Ces sont des bascules pour lesquelles le changement d'état de la sortie, qui correspond à une nouvelle combinaison des valeurs d'entrées, ne peut s'effectuer que lorsqu'un signal de contrôle appelé signal d'horloge prend lui-même une valeur donnée.

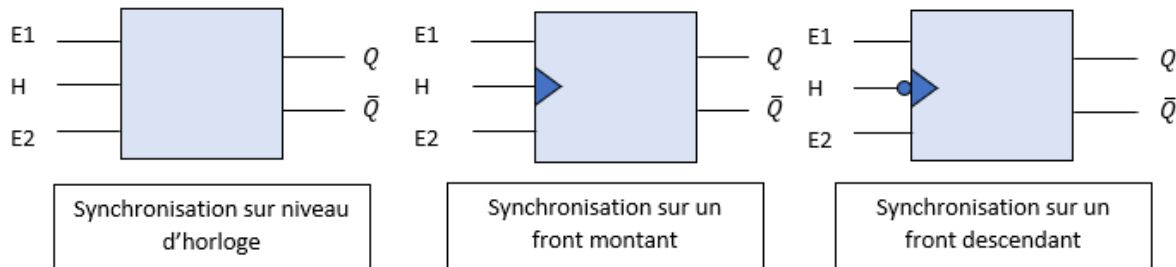
Horloge : composant électronique dont le signal logique passe indéfiniment et régulièrement d'un niveau haut à un niveau bas (succession de 1 et de 0), chaque transition s'appelle un top.



Fréquence : nombre des chargements par seconde qui s'exprime en Hertz : $f = \frac{1}{T}$

Les types de synchronisation : Les bascules synchrones fonctionnent selon l'un des deux modes de synchronisation suivants :

- Synchronisation sur un niveau d'horloge : la synchronisation sur un niveau du signal d'horloge est employée dans les bascules à verrouillage. Il suffit d'appliquer le niveau convenable (niveau haut ou bas), appelé niveau actif pour que la sortie de la bascule puisse changer d'état.
- Synchronisation sur un front d'horloge : La durée de synchronisation est réduite à son minimum, c'est à dire au temps pour que le signal d'horloge passe d'un niveau à un autre (front montant ou descendant). Cette synchronisation est employée dans les bascules à déclenchement sur fronts.



La bascule RSH :

La bascule RSH est une bascule synchrone dont les entrées n'influent pas sur l'état de sortie tant qu'une variable nommée horloge n'est pas activée. En fait une bascule RSH est une bascule RS à laquelle on a ajouté deux portes NAND, pour former la borne de contrôle d'horloge H. La figure ci-dessous présente le schéma interne de la bascule RSH.

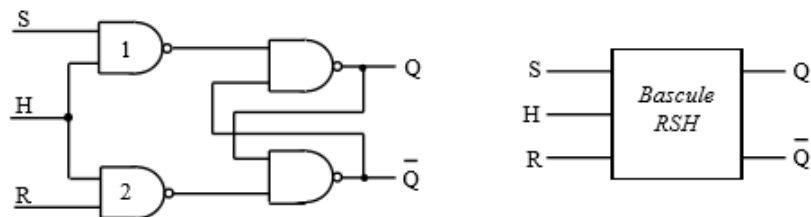
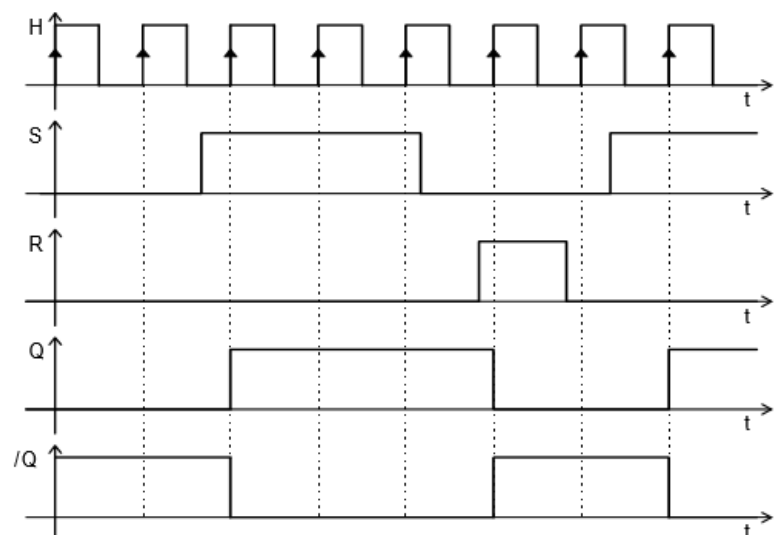


Table de vérité :

H	R	S	Q_{n+1}	\overline{Q}_{n+1}	Signification
0	X	X	Q_n	\overline{Q}_n	Mémorisation
1	0	0	Q_n	\overline{Q}_n	Mémorisation
1	0	1	1	0	Mise à 1
1	1	0	0	1	Mise à 0
1	1	1	—	—	Etat interdit



La bascule JK :

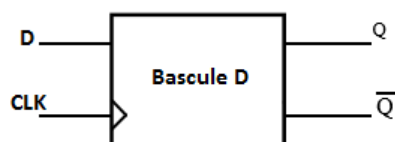
La bascule JK de type edge-triggered est une bascule qui se déclenche sur un front montant ou descendant d'horloge. Les entrées J et K commandent l'état de la bascule comme les entrées de la bascule R-S synchrone, à l'exception importante que $J = K = 1$ ne donne pas lieu à une situation ambiguë, mais plutôt à un basculement de la sortie. Le tableau ci-dessous montre la table de vérité de la bascule JK active sur un front montant du signal d'horloge CLK :

H	J	K	Q_{n+1}	\overline{Q}_{n+1}	Signification
0	X	X	Q_n	\overline{Q}_n	Mémorisation
↑	0	0	Q_n	\overline{Q}_n	Mémorisation
↑	0	1	1	0	Mise à 1
↑	1	0	0	1	Mise à 0
↑	1	1	\overline{Q}_n	Q_n	Etat interdit

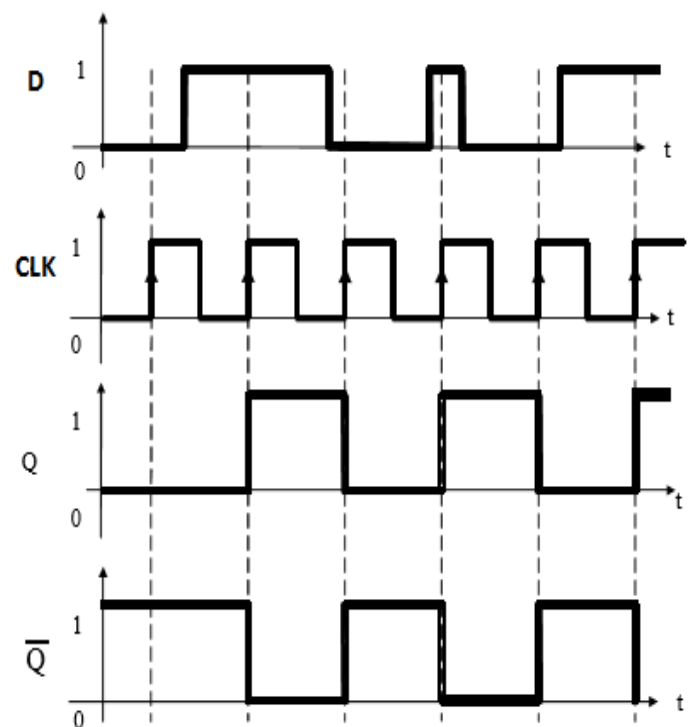
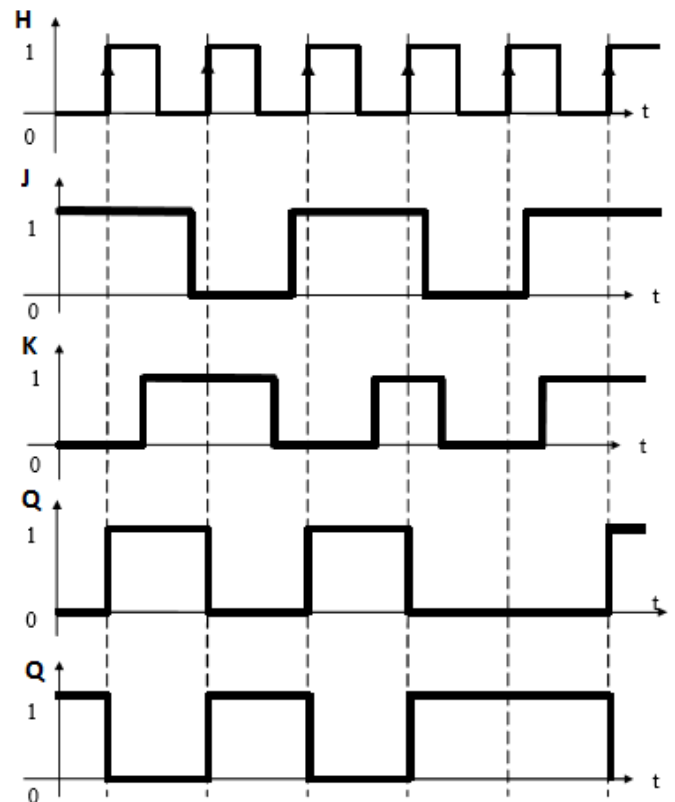
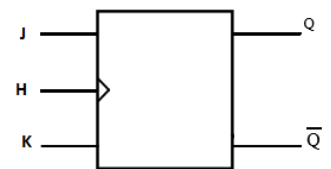
La bascule D :

Le fonctionnement de cette bascule est le suivant : la sortie recopie l'entrée au moment du front montant (ou descendant) du signal d'horloge, et non pas pendant sa mise à 1. On réalise ainsi une bascule D qui recopie son entrée uniquement au moment de la transition positive du signal d'horloge CLK.

Table de vérité :



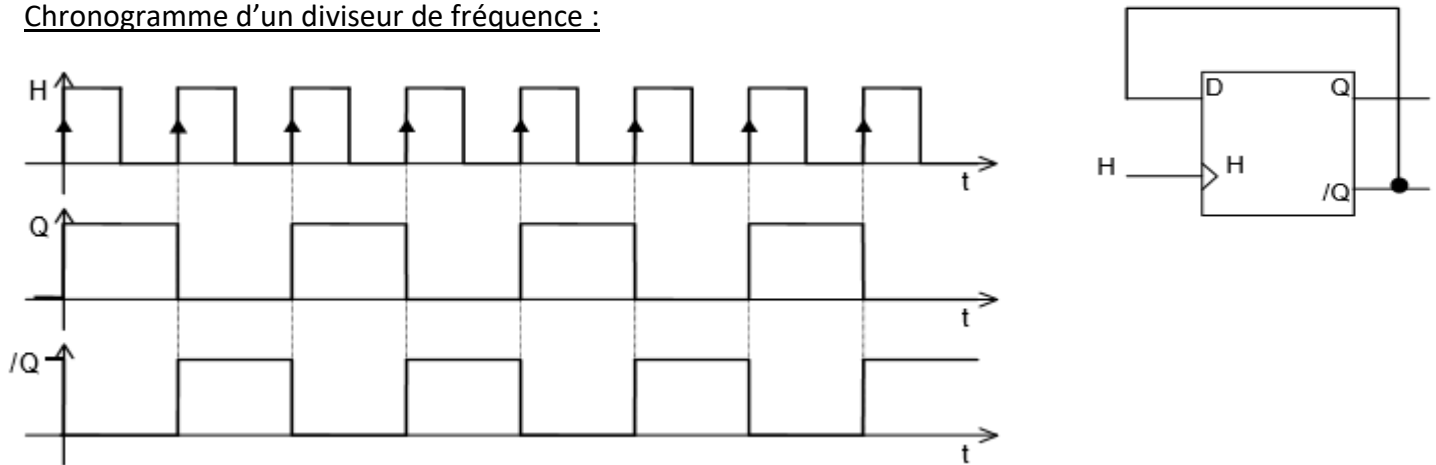
CLK	D	Q_{n+1}	\overline{Q}_{n+1}	Signification
0 ou 1	X	Q_n	\overline{Q}_n	Mémorisation
↑	0	0	1	Mise à 0
↑	1	1	0	Mise à 1



Exemple d'application de la bascule D : le diviseur de fréquence

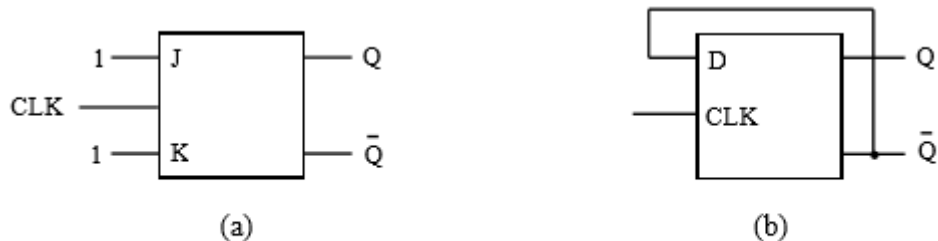
La bascule D change d'état à chaque top d'horloge : deux périodes du signal de l'horloge sont équivalentes à une période pour le signal de sortie : $T_Q = 2 T_{CLK} \rightarrow f_Q = \frac{f_{CLK}}{2}$

Chronogramme d'un diviseur de fréquence :



La bascule T (Toggle) :

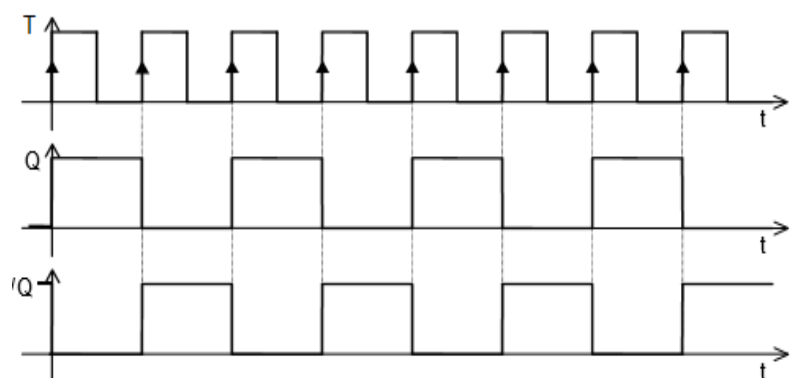
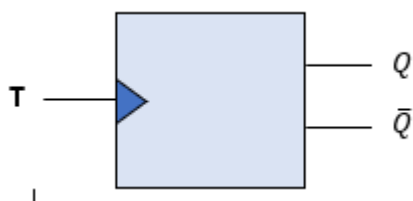
La bascule T est une bascule qui a une seule entrée T et deux sorties complémentaires Q et \bar{Q} . Les sorties de la bascule changent d'état à chaque front appliqué à son entrée T. La bascule T peut être réalisée à partir d'une bascule JK ou d'une bascule D, comme le montre la figure ci-dessous.



Réalisation d'une bascule T à partir d'une bascule JK (a), d'une bascule D (b)

La table de vérité de la bascule T est la suivante :

T	Q_{n+1}	\bar{Q}_{n+1}	Signification
0 ou 1	Q_n	\bar{Q}_n	Mémorisation
↑	\bar{Q}_n	Q_n	Basculement



4) Compteur asynchrone

Un compteur asynchrone est constitué de plusieurs bascules en cascade. La première bascule reçoit le signal d'horloge CLK, le deuxième reçoit comme signal d'horloge le signal de sortie de la bascule précédente et ainsi de suite.

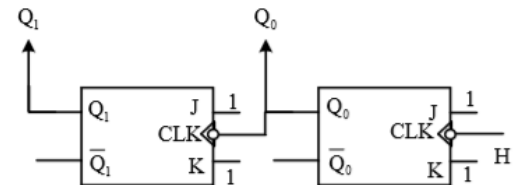
Compteur à cycle complet :

Exemple 1 : Compteur asynchrone modulo 4, avec des bascules JK à front descendant.

La capacité de comptage (N) fixe le nombre de bascules (n) :
 $2^{n-1} < N \leq 2^n$

$N = 4 = 2^2 \implies n = 2$ bascules.

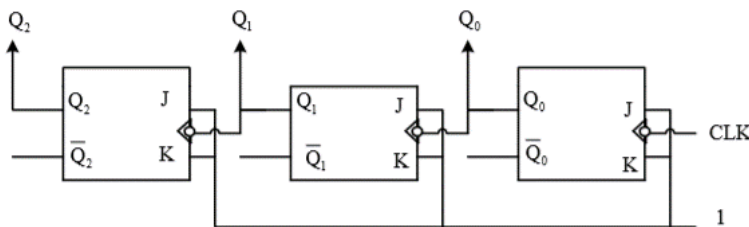
Bascules à front descendant donc : $H_0 = H$ et $H_i = Q_{i-1}$



Exemple 2 : Compteur asynchrone modulo 8, avec des bascules JK à front descendant.

Le nombre de bascules : $2^{n-1} < N \leq 2^n \rightarrow N = 8 = 2^3$
 $\implies n = 3$ bascules

Bascules à front descendant donc $H_0 = H$ et $H_i = Q_{i-1}$

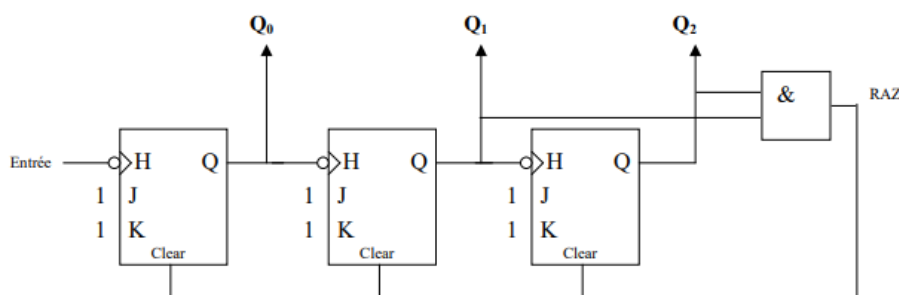


Remarque : pour des bascules à front montant :

$$H_0 = H \text{ et } H_i = \overline{Q_{i-1}}$$

Compteur à cycle incomplet : compteur modulo 6

On sait que $(6)_{10} = (110)_2$ Donc $Q_1 = Q_2 = 1$. Pour compter de 0 à 5, il faut forcer le compteur à s'arrêter à la valeur 6 c'est-à-dire forcer à 0 les sorties Q_1 et Q_2 . Ainsi l'entrée de forçage $RAZ = Q_1 \cdot Q_2$



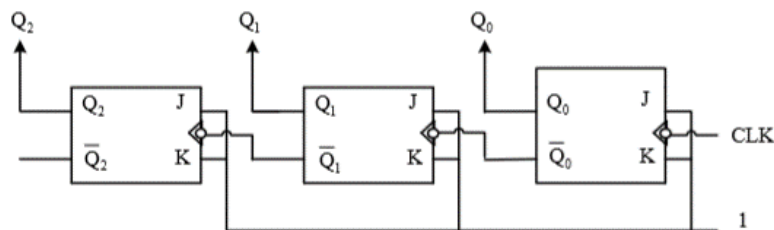
5) Décompteur asynchrone

Dans les décompteurs asynchrones l'information à décompter est reliée à l'entrée d'horloge de la première bascule.

- Si le front est descendant alors la sortie Q de la première bascule est reliée à l'entrée d'horloge de la bascule suivante est ainsi de suite ;
- Si le front est montant alors la sortie Q de la première bascule est reliée à l'entrée d'horloge de la bascule suivante est ainsi de suite.

Exemple : décompteur asynchrone modulo (N=8) à base de bascule JK

Bascule JK à front descendant donc $H_0 = H$ et $H_i = \overline{Q_{i-1}}$



6) **Compteur synchrone :** Pour ce type de compteur une horloge commune est appliquée simultanément à toutes les entrées des différentes bascules.

La capacité de comptage (N) fixe le nombre de bascules (n) : $2^{n-1} < N \leq 2^n$

La synthèse d'un compteur synchrone consiste à calculer chaque entrée des bascules constituant le compteur afin de conditionner leurs évolutions lors de la prochaine impulsion d'horloge.

Exemple 1 : Synthèse d'un compteur synchrone modulo 8 :

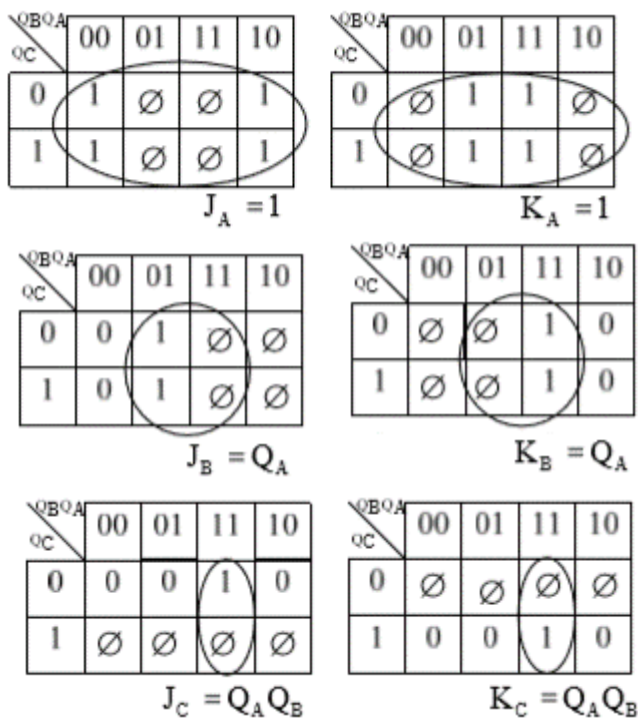
Etape 1 : table d'excitation de la bascule JK

Q_n	Q_{n+1}	J	K
0	0	0	Ø
0	1	1	Ø
1	0	Ø	1
1	1	Ø	0

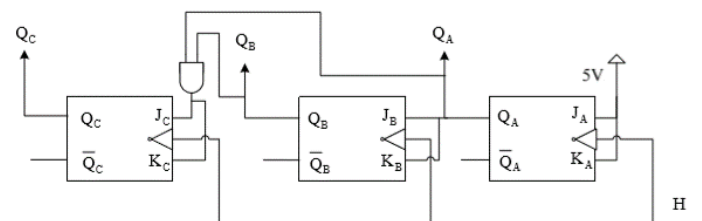
Etape 2 : table de transition du compteur

N°	n		n+1		Entrées: (Ø = 0 ou 1)							
	Q_C	Q_B	Q_A	Q_C'	Q_B'	Q_A'	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	1	Ø	0	Ø	Ø	0	Ø
1	0	0	1	0	1	0	Ø	1	1	Ø	0	Ø
2	0	1	0	0	1	1	Ø	Ø	Ø	0	0	Ø
3	0	1	1	1	0	0	Ø	1	Ø	1	1	Ø
4	1	0	0	1	0	1	Ø	0	Ø	Ø	Ø	0
5	1	0	1	1	1	0	Ø	1	1	Ø	Ø	0
6	1	1	0	1	1	1	Ø	Ø	Ø	0	Ø	0
7	1	1	1	0	0	0	Ø	1	Ø	1	Ø	1

Etape 3 : mise en équation par les tableaux de Karnaugh



Etape 4 : implémentation

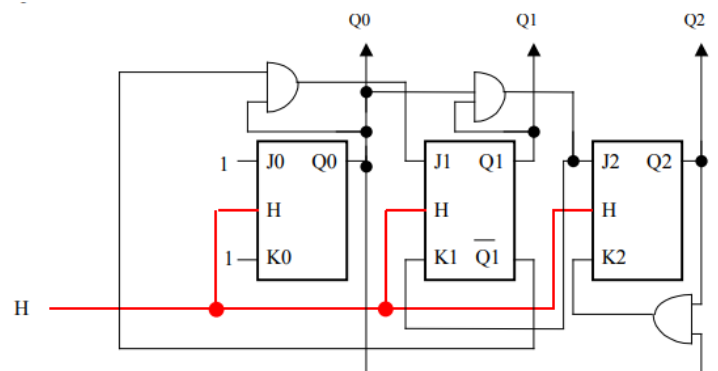


Exemple 2 : réalisation d'un compteur synchrone modulo 6

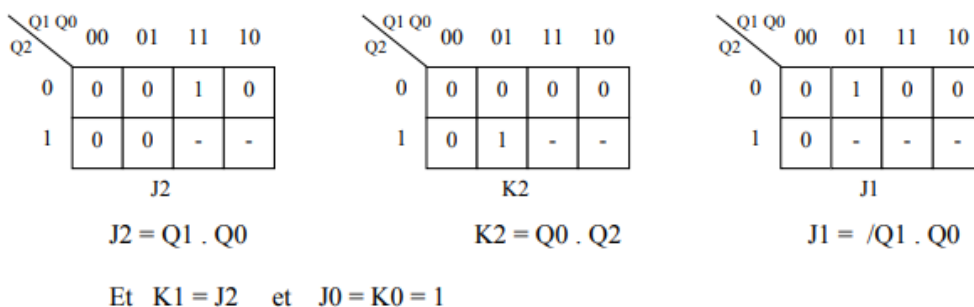
On a : $2^2 < 6 \leq 2^3$ donc ce compteur nécessite 3 bascules JK.

Table de vérité :

N	Q ₂	Q ₁	Q ₀	J ₂	K ₂	J ₁	K ₁	J ₀	K ₀
0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	1	0	0	1
2	0	1	0	0	0	0	0	1	0
3	0	1	1	1	0	0	1	0	1
4	1	0	0	0	0	0	0	1	0
5	1	0	1	0	1	0	0	0	1



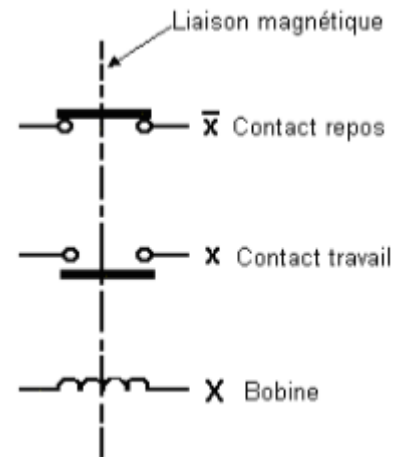
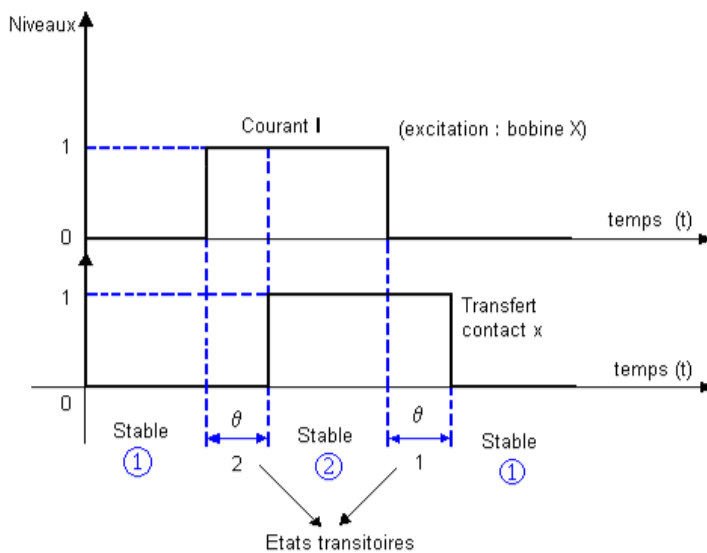
Tableaux de Karnaugh :



Implémentation

7) Synthèse d'un système séquentiel

Notion d'état stable et état de transition



Ce système permet un rétrocouplage : la bobine du relais communément appelée X (majuscule indiquant qu'il s'agit d'une variable de sortie) est une variable de sortie générant une nouvelle variable d'entrée liée : x , constituée par l'un des contacts de ce même relais.

On appelle états stables les états pendant lesquels la bobine d'un relais X , ou excitation, est dans le même état que son contact x , ou transfert.

Par convention, le nombre indiquant un état stable est toujours entouré d'un cercle.

Exemple : $X = 0$ $x = 0$ état stable repos (1)
 $X = 1$ $x = 1$ état stable travail (2)

Dans les deux états précédents, il y avait identité entre l'excitation et le transfert. Toutefois, le retard de x conditionne des états intermédiaires ou transitoires pour lesquels excitation et transfert sont dans des états complémentaires.

Exemple :

État transitoire 2 (par convention ne pas cercler le nombre), nous avons : $X = 1$ et $x = 0$.

Le transfert n'est pas instantané.

État transitoire 1 : nous avons $X = 0$ et $x = 1$

Les étapes de synthèse d'un circuit séquentiel par la méthode de Huffman

1. Description et analyse du cahier des charges
2. Modélisation du cahier des charges :
 - Par le graphe de fluence
 - Par la matrice primitive
3. Minimisation des états :
 - Par le polygone de fusion
 - Par la matrice contractée
4. Codage :
 - Des sorties d'excitation secondaire
 - Des sorties principales
5. Implémentation du circuit séquentiel

Résolution d'un problème séquentiel :

Pour illustrer cette méthode, reprenons un premier exemple.

Supposons que l'on dispose, pour commander l'allumage d'une lampe L, de deux boutons poussoirs, l'un appelé «m» ou «marche» et l'autre «a» ou «arrêt».

Le tableau suivant représente le fonctionnement du montage.

m	a	L	OPERATION EFFECTUEE
0	0	0	Aucun bouton enfoncé : la lampe L est éteinte
1	0	1	On appuie sur le bouton « m » : la lampe L s'allume
0	0	1	Le bouton « m » est relâché : la lampe L s'allume
0	1	0	Le bouton « a » enfoncé : la lampe L s'éteint
0	0	0	Le bouton « a » est relâché : La lampe L reste éteinte

Nous pouvons voir en examinant le tableau qu'il existe pour deux combinaisons identiques des variables d'entrées «a» et «m» deux états logiques différents pour L (combinaisons cerclées) ;

Ceci est nouveau. Nous ne pouvons donc pas résoudre le problème par la méthode combinatoire traditionnelle. En effet, le tableau de Karnaugh n'admet qu'une valeur 1 ou 0 par case ou combinaison des variables d'entrées.

Etape 1 : analyse du cahier des charges :

On dispose de deux variables d'entrée «m» et «a».

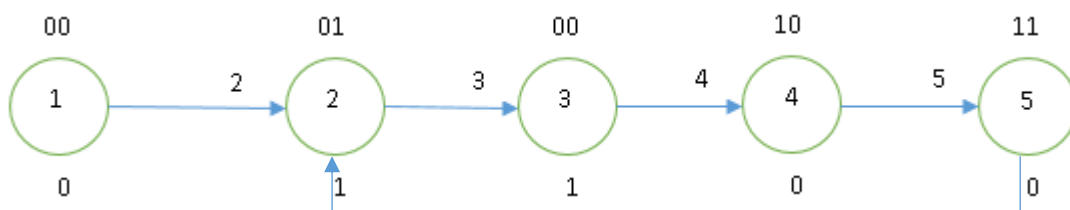
Ces deux variables permettent d'obtenir $2^2 = 4$ combinaisons.

Reprenons le fonctionnement du montage et définissons le nombre d'états stables :

- ① État initial $a = 0, m = 0$ avec $L = 0$
- ② Action sur m $a = 0, m = 1$ d'où $L = 1$
- ③ m est relâché $a = 0, m = 0$ et L toujours à 1
- ④ Action sur a $a = 1, m = 0$ et $L = 0$
- ⑤ Action simultanée sur a et m $a = 1, m = 1$ on a choisi de privilégier l'arrêt : $L = 0$

Etape 2 : modélisation

- **En utilisant le graphe d'états :** On appelle séquence une succession bouclée d'états stables séparés par des états transitoires. A chaque transition une seule entrée peut changer d'état binaire.



- **En utilisant la matrice primitive :** On appelle matrice primitive des états un tableau du modèle dans lequel il y a une ligne par état stable. on se sert d'un tableau avec quatre colonnes $2^{\text{nombre des entrées}}$ (car on a 2 entrées) correspondant aux combinaisons des entrées que l'on range selon le code binaire réfléchi. Dans une cinquième colonne, on inscrira la valeur binaire de l'état de sortie.

Les règles de remplissage de la matrice primitive sont comme suit :

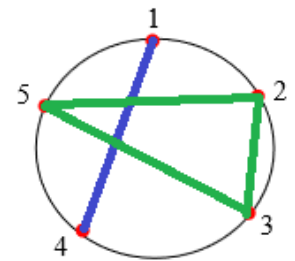
- On positionne chaque état stable dans la case pour laquelle les variables « m » et « a » sont aux valeurs ayant entraîné cet état stable.
- Les états transitoires sont situés à l'intersection de la ligne sur laquelle figure l'état stable initial et de la colonne dans laquelle figure l'état stable suivant.

- il convient de hachurer pour les éliminer, toutes les cases pour lesquelles sur une même ligne deux variables changent par rapport à la combinaison des variables qui a provoqué l'état stable inclus dans cette ligne.

	00	01	11	10	L
am	①	2	×	4	0
	3	②	5	×	1
	③	2	×	4	1
	1	×	5	④	0
	×	2	⑤	4	0

Etape 3 : minimisation des états

- **par le polygone de fusion** : Répartissons sur un cercle en comptant dans le sens des aiguilles d'une montre les cinq points matérialisant les cinq lignes de la matrice primitive des états.
Règles de contraction ou de fusion :
 - Il est possible de contracter la matrice primitive des états en superposant deux lignes si elles présentent verticalement les mêmes états.
 - On peut fusionner deux lignes en une seule lorsqu'il y a un état stable ou un transitoire de même numéro situé dans la même colonne ou un état stable ou transitoire avec une case hachurée située dans la même colonne, en effet, la case hachurée indique que, pour des raisons technologiques, ce cas ne peut se produire et est donc indifférent.
 - Les états de sortie n'interviennent pas dans les superpositions possibles mais si on a le choix on préférera des états de sortie identiques.
- **La matrice contractée** : c'est de fusionner les lignes équivalentes en une seule ligne en suivant les règles suivantes :
 - Les états stables emportent les états transitoires.
 - Les états hachurés emportent les états transitoires.



	00	01	11	10
Ligne 2	3	②	5	×
Ligne 3	③	2	×	4
Ligne 5	×	2	⑤	4

↓

	00	01	11	10
Lignes 2; 3; 5	③	②	⑤	4

	00	01	11	10
Ligne .1.	①	2	×	4
Ligne .4.	1	×	5	④

↓

	00	01	11	10
Lignes .1; .4.	①	2	5	④

Nous pouvons reconstituer une matrice dite contractée :

Etape 4 : codage des sorties secondaires et des sorties principales

Pour une matrice de $2^1 = 2$ lignes, on a 1 variable complémentaire, pour $2^2 = 4$ lignes, on a 2 variables complémentaires et pour 2^n lignes, on a n variables.

Nous voyons que la matrice contractée comprend deux lignes, ce qui signifie que les variables complémentaires ou transferts sont au nombre de un que l'on appelle x.

Remarque ; Si la matrice comprend $2^0 = 1$ ligne, le système examiné peut se résumer à un système combinatoire (aucune variable complémentaire).

Matrice des sorties secondaires :

- Pour chaque **état stable** X aura **la même valeur** que la variable d'entrée x que l'on voit sur la ligne de l'état stable (ex : pour l'état stable 1 {1ère ligne}, x vaut 0, donc X vaudra également 0).
- Pour **les états instables** (états transitoires), on prend comme valeur de X, la valeur de l'entrée x de l'état stable.
- Pour les cases hachurées, on met des valeurs indifférentes.

am	00	01	11	10
x				
0	0	0	0	1
1	1	0	0	1

$$X = a \cdot \bar{m} + x \cdot \bar{m}$$

Matrice des sorties principales :

Les matrices de sortie sont construites à partir de la matrice des phases et de la matrice réduite. On écrit une matrice (tableau de Karnaugh) pour chaque sortie.

- Pour **les états stables** il faut prendre l'état spécifié dans **la matrice des phases**.
- Pour **les états instables** (ils ne durent que pendant des temps très brefs = temps de propagations) **l'état peut en général être quelconque**, mais parfois pour éviter des problèmes d'aléas ou d'états transitoires parasites, on préfère imposer soit l'état de l'étape stable de départ, ou celui de l'étape stable d'arrivée.

L					
	am	00	01	11	10
x					
0		1	1	0	X
1		0	X	X	0

$$L = \bar{x} \cdot \bar{a}$$

Etape 5 : Logigramme

