

Lycée Technique Mohammedia

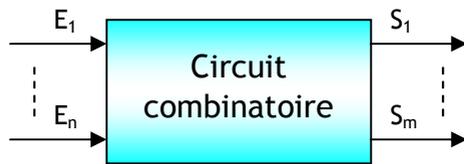
Logique Séquentielle

1^{ère} STE **Unité ATC**

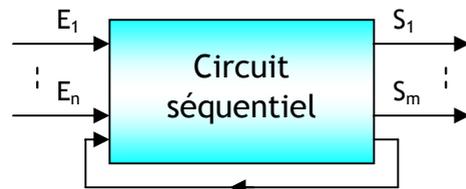
Professeur : **MAHBAB**

1. INTRODUCTION :

A la différence d'un circuit combinatoire, l'état d'un circuit séquentiel dépend de l'état de ses entrées et de l'état précédent de ses sorties ; il doit donc "se rappeler" ou avoir de la "mémoire". Par mémoire, on exprime le phénomène qui consiste à conserver l'effet d'un événement après sa disparition.



L'état des sorties ne dépend que de l'état des entrées



L'état des sorties dépend de l'état des entrées et de l'état précédent de ses sorties

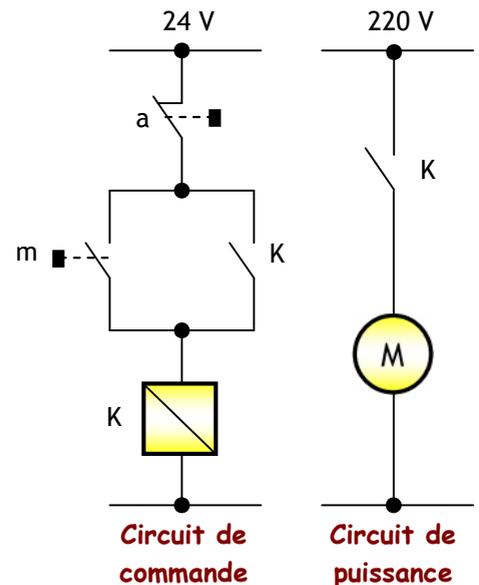
2. CIRCUIT MEMOIRE EN TECHNOLOGIE ELECTRIQUE :

Exemple : Circuit de la commande d'un moteur d'une perceuse

- ☞ Le bouton "m" permet de mettre en **marche** le moteur.
- ☞ Le bouton "a" permet de l'**arrêter**.
- ☞ Quand on appuie sur le bouton **m**, le moteur démarre ; quand on relâche le bouton, le moteur continue à tourner. L'ordre de mise en marche a donc été **mémorisé** ;
- ☞ Il en est de même pour le bouton **a** ;
- ☞ L'action arrêt est prioritaire : si **m** et **a** sont appuyés en même temps, on **arrête** le moteur.
- ☞ Quand l'utilisateur appuie sur **m**, la bobine du relais est alimentée. Les contacts **K** associés se ferment. Si l'utilisateur relâche **m**, le courant continue à circuler par **K** ; le relais est alors **auto-alimenté** et le moteur continue à tourner. L'équation du relais K et du moteur M est :

$K = \dots\dots\dots$

Circuit mémoire élémentaire en technologie électrique



3. CIRCUIT MEMOIRE EN TECHNOLOGIE ELECTRONIQUE :

Le circuit mémoire élémentaire en électronique est appelé **BASCULE RS**.

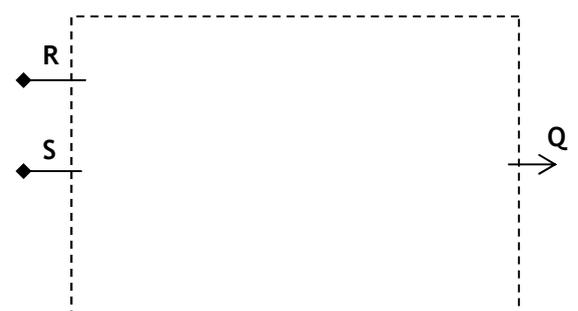
- ☞ **S** (Set) est l'entrée de mise à **1** de la sortie Q ;
- ☞ **R** (Reset) est l'entrée de mise à **0** de la sortie Q ;
- ☞ **Q** sortie de la bascule.



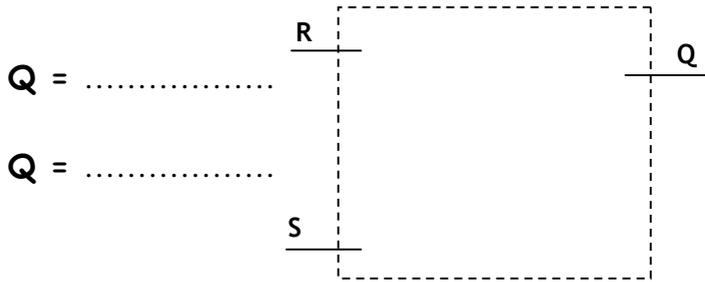
3.1. Approche 1 :

On part de l'équation d'auto-alimentation du relais et on fait la correspondance logique : **S** correspond à **m**, **R** correspond à **a** et **Q** correspond à **K**. On en déduit alors l'équation de la bascule SR avec Reset prioritaire, ainsi que son logigramme :

$Q = \dots\dots\dots$



Ce circuit est plus connu par sa réalisation simplifiée avec l'utilisation de portes NOR.



| S | R | Q | Fonction de la bascule |
|---|---|-----|------------------------|
| 0 | 0 | ... | |
| 0 | 1 | ... | |
| 1 | 0 | ... | |
| 1 | 1 | ... | |

q indique l'état précédent (0 ou 1) (*) Etat indéterminé si on passe à SR=00

3.2. Approche 2 :

Dans cette approche, on raisonne comme pour un circuit combinatoire. Il est donc nécessaire de connaître l'état de Q pour connaître l'état de la sortie lorsque les deux entrées sont à 0 (état de mémoire). On introduit alors une variable supplémentaire qui indique l'état précédent de Q. On note "q" cette variable.

| S | R | q | Q |
|---|---|---|-----|
| 0 | 0 | 0 | ... |
| 0 | 0 | 1 | ... |
| 0 | 1 | 0 | ... |
| 0 | 1 | 1 | ... |
| 1 | 0 | 0 | ... |
| 1 | 0 | 1 | ... |
| 1 | 1 | 0 | ... |
| 1 | 1 | 1 | ... |

| | | R.q | | | |
|---|---|-----|-----|-----|-----|
| | | 00 | 01 | 11 | 10 |
| S | 0 | ... | ... | ... | ... |
| | 1 | ... | ... | ... | ... |

Q =

Q =

Puisqu' on a (q = Q), alors :

Q =

1. INTRODUCTION :

Les Bascules, également connues sous le nom de flip-flop, sont des circuits logiques à deux états stables en sortie. Les bascules RS, D et JK sont décrites par des tables de vérité, dont la connaissance *doit devenir instinctive* si l'on veut pouvoir maîtriser des interfaces complexes.

A l'instar des opérateurs logiques élémentaires en logique combinatoire, les bascules (flip-flop) sont les éléments de base de la logique séquentielle.

2. LA BASCULE SR :

2.1. Bascule SR Asynchrone :

Les bascules RS sont à la base de tous les éléments de mémorisation. Il s'agit d'un montage utilisant deux portes NAND et capable de mémoriser un niveau logique choisi à l'aide de deux sorties complémentaires.

Symbole :

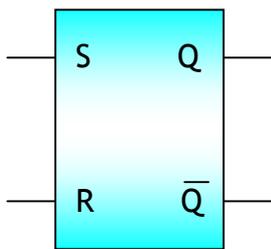
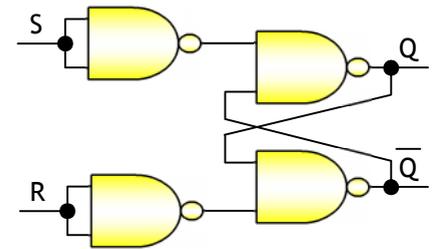


Table de vérité :

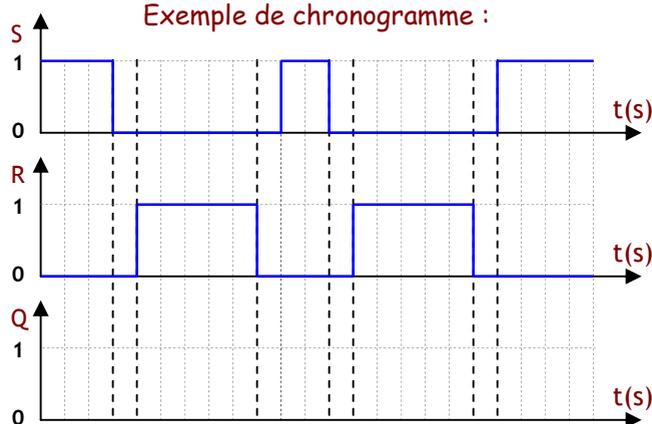
| S | R | Q | /Q | Commentaires |
|-----|-----|-----|-----|--------------|
| ... | ... | ... | ... | |
| ... | ... | ... | ... | |
| ... | ... | ... | ... | |
| ... | ... | ... | ... | |

Logigramme avec portes NAND :



q indique l'état précédent de Q

Exemple de chronogramme :



Remarque :

Cette bascule présente deux inconvénients majeurs :

- ☞ Sensibilité de la sortie Q aux changements indésirables (parasites) des entrées S et R ;
- ☞ La configuration S = R = 1 est à éviter parce qu'elle conduit à l'égalité entre les deux sorties et donc il n'y a plus complémentarité comme c'est indiqué dans la définition de la bascule ;

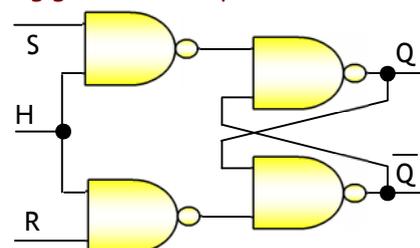
2.2. Bascule SR Synchrones statique :

Elle dispose d'une entrée de validation de la programmation en R et S. Ce signal appelé **HORLOGE** est actif sur *niveau haut* ou sur *niveau bas*. Donc, les ordres Set et Reset ne changent l'état de la sortie qu'après l'autorisation d'un signal de commande **H**.

Table de vérité (H active sur niveau haut) :

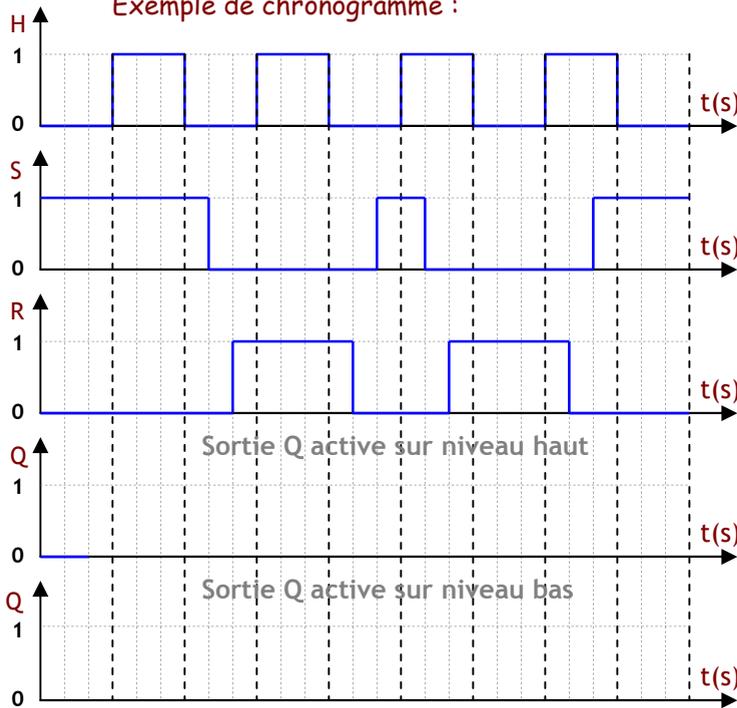
| H | S | R | Q | /Q | Commentaires |
|-----|-----|-----|-----|-----|--------------|
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |

Logigramme avec portes NAND :



q indique l'état précédent de Q

Exemple de chronogramme :



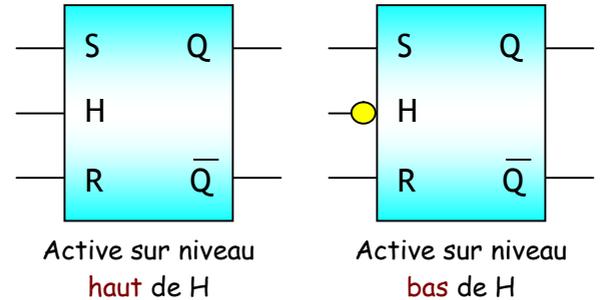
Remarque :

La bascule SR synchrone permet de résoudre le premier inconvénient de la bascule SR asynchrone.

Fonctionnement :

- ☞ Signal d'horloge actif → bascule RS
- ☞ Absence de signal actif d'horloge → mémoire.

Symbole :



2.3. Bascule SR Synchrone dynamique (maitre esclave) :

C'est une bascule R S dont la prise en compte de l'état des entrées est synchronisée par une d'horloge. L'horloge est active sur **front montant** ou sur **front descendant**.

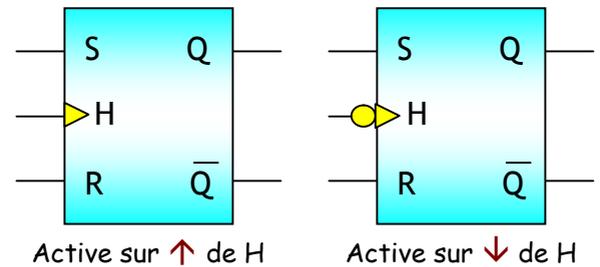
Table de vérité (H active sur front montant) :

| H | S | R | Q | /Q | Commentaires |
|-----|-----|-----|-----|-----|--------------|
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |

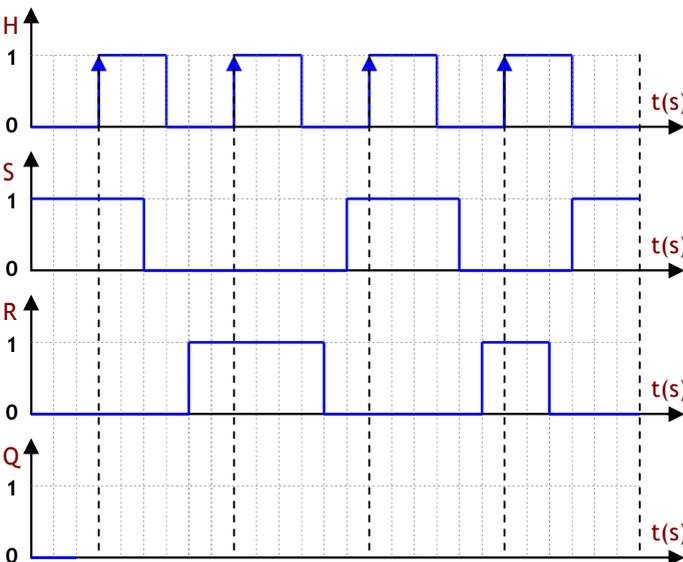
Fonctionnement :

- ☞ Signal d'horloge actif → bascule RS
- ☞ Absence de signal actif d'horloge → mémoire.

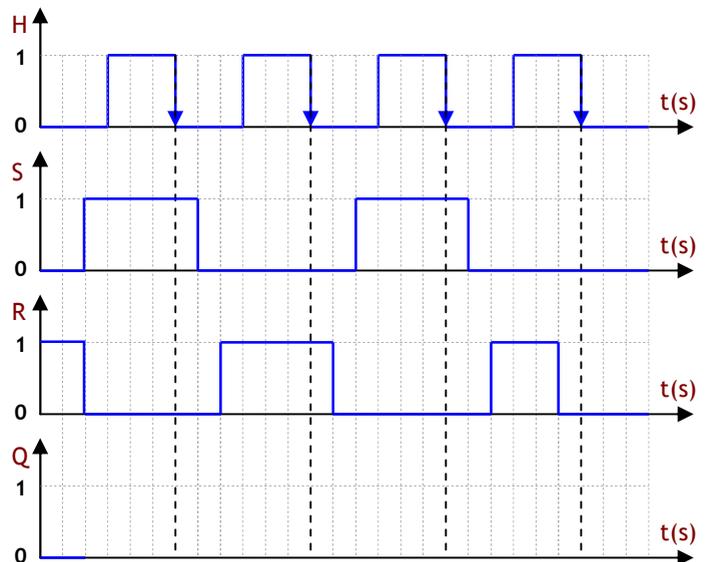
Symbole :

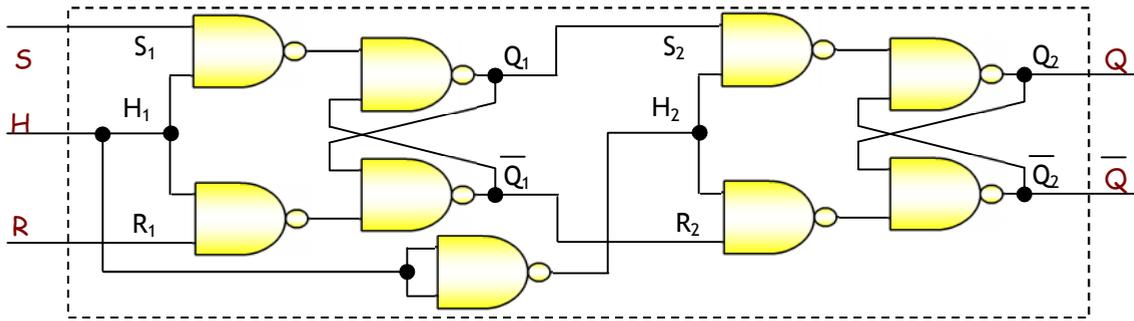


Chronogramme (H active sur front montant) :



Chronogramme (H active sur front descendant) :





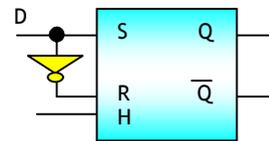
Logigramme avec portes NAND ; d'une bascule RS dynamique (maitre esclave) active sur front descendant

3. LA BASCULE D :

La bascule D est une bascule synchrone (avec une entrée d'horloge) à une seule entrée de donnée : l'entrée D ($D=Data=Donnée$). Elle supprime la combinaison interdite de la bascule RS, en ne gardant que les 3 fonctions utiles : (la mise à 0, la mise à 1 et la mémorisation).

Il existe 2 types de bascules D :

- ☞ la bascule D active sur niveau (statique) ;
- ☞ la bascule D active sur front (dynamique).



3.1. Bascule D statique :

Table de vérité (H active sur niveau haut) :

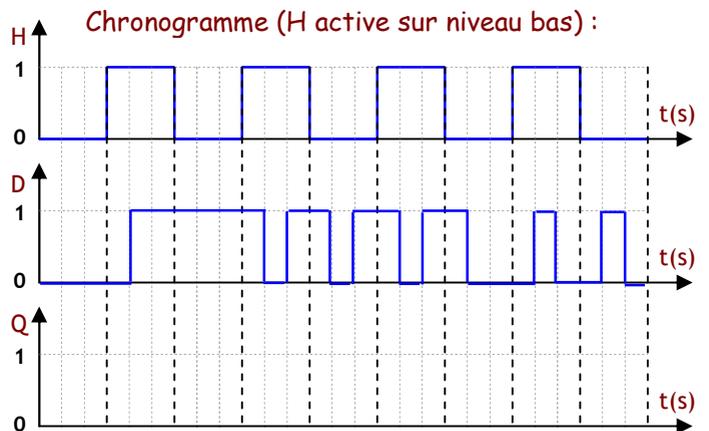
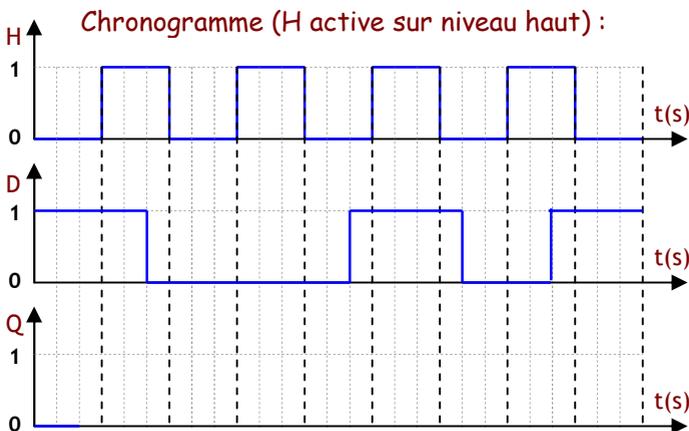
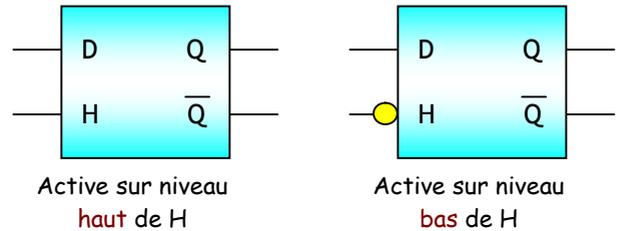
| H | D | Q | /Q | Commentaires |
|-----|-----|-----|-----|--------------|
| ... | ... | ... | ... | |
| ... | ... | ... | ... | |
| ... | ... | ... | ... | |

| H | D | Q | /Q | Commentaires |
|-----|-----|-----|-----|--------------|
| ... | ... | ... | ... | |
| ... | ... | ... | ... | |
| ... | ... | ... | ... | |

Fonctionnement :

- ☞ signal actif, la sortie recopie l'entrée $D \rightarrow$ *Fonction Ecriture* ;
- ☞ Signal inactif \rightarrow *Fonction Mémoire*.

Symbole :



3.2. Bascule D dynamique :

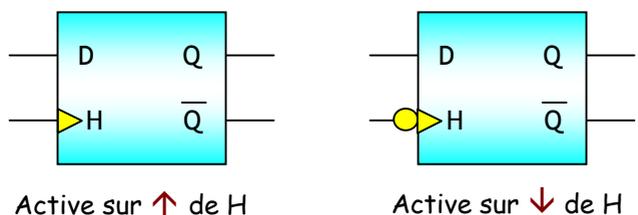
Table de vérité (H active sur front montant) :

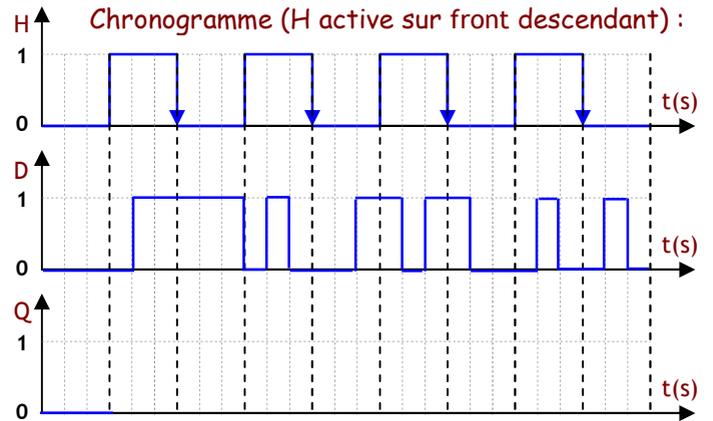
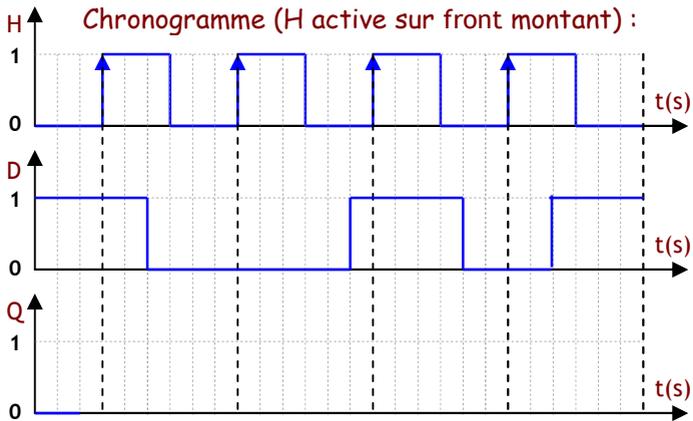
| H | D | Q | /Q | Commentaires |
|-----|-----|-----|-----|--------------|
| ... | ... | ... | ... | |
| ... | ... | ... | ... | |
| ... | ... | ... | ... | |
| ... | ... | ... | ... | |
| ... | ... | ... | ... | |

Fonctionnement :

- ☞ signal actif, la sortie recopie l'entrée $D \rightarrow$ *Fonction Ecriture* ;
- ☞ Signal inactif \rightarrow *Fonction Mémoire*.

Symbole :





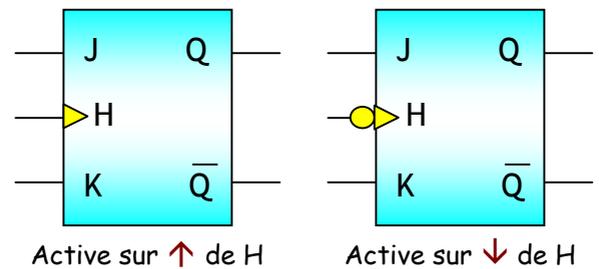
4. LA BASCULE JK :

C'est une bascule synchrone, offrant les fonctions *mémorisation*, *mise à 0*, *mise à 1* (les 3 fonctions de la bascule D), et assurant en plus la fonction **basculer**. Donc, la bascule JK permet en plus de lever l'ambiguïté qui existe pour l'état S=R=1 d'une bascule SR.

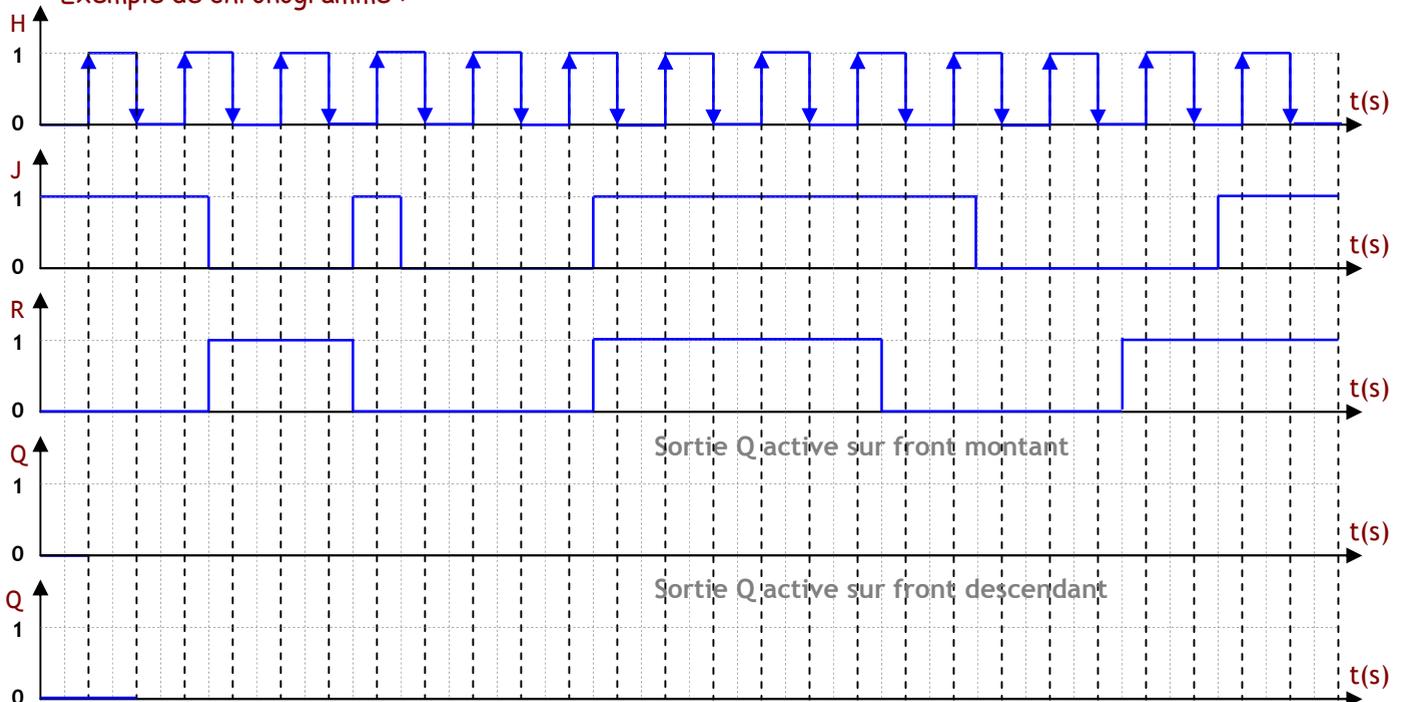
Table de vérité (H active sur front montant) :

| H | J | K | Q | /Q | Commentaires |
|-----|-----|-----|-----|-----|--------------|
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |
| ... | ... | ... | ... | ... | |

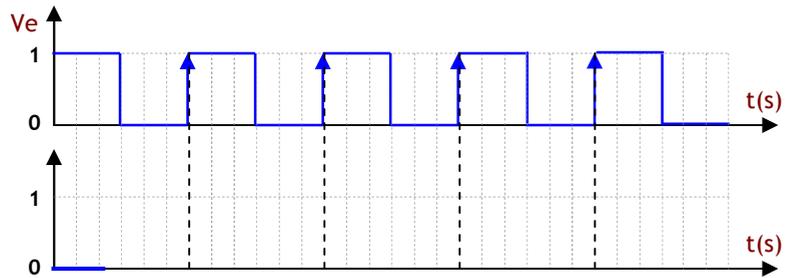
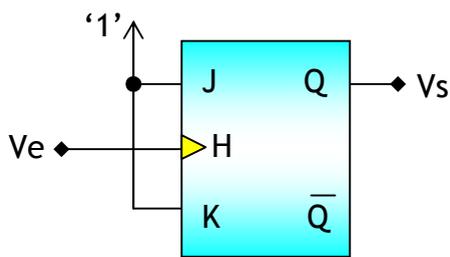
Symbole :



Exemple de chronogramme :



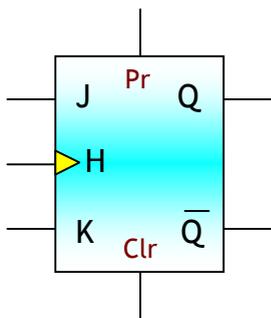
Si on utilise la bascule JK avec J = K = 1, on obtient l'une des principales applications de la bascule JK à savoir le diviseur de fréquence par 2.



5. Fonctionnement forcé des bascules :

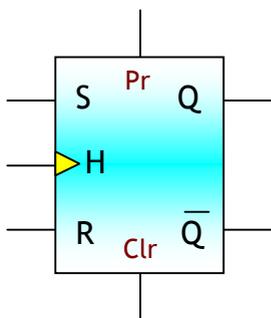
Il est parfois nécessaire d'affecter le niveau de sortie d'une bascule de manière non synchrone, c'est-à-dire indépendamment de l'horloge. C'est le rôle des entrées de forçage asynchrone Preset (Set) et Clear (Clr) qui permettent d'initialiser la bascule :

Bascule JK avec entrées de forçage :



| Pr | Clr | H | J | K | Q | /Q | Commentaires |
|-----|-----|-----|-----|-----|-----|-----|---|
| ... | ... | ... | ... | ... | ... | ... | Forçage de la sortie Q |
| ... | ... | ... | ... | ... | ... | ... | Signal d'horloge inactif → Mémorisation |
| ... | ... | ... | ... | ... | ... | ... | Fonctionnement normal de la bascule SR |

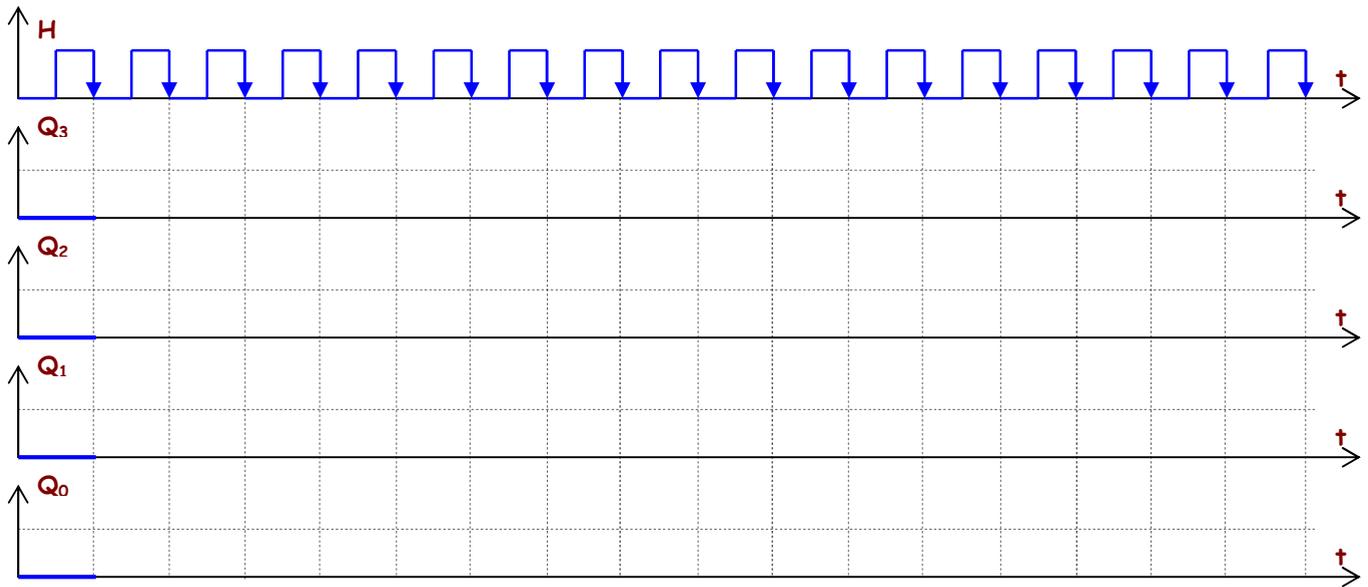
Bascule SR avec entrées de forçage :



| Pr | Clr | H | S | R | Q | /Q | Commentaires |
|-----|-----|-----|-----|-----|-----|-----|---|
| ... | ... | ... | ... | ... | ... | ... | Forçage de la sortie Q |
| ... | ... | ... | ... | ... | ... | ... | Signal d'horloge inactif → Mémorisation |
| ... | ... | ... | ... | ... | ... | ... | Fonctionnement normal de la bascule SR |
| 0 | 0 | ↑ | 1 | 1 | 1 | 1 | Etat indéfini |

Circuits intégrés à bascules (exercice):

- ☞ Brochage et table de vérité du **74 279**
- ☞ Brochage et table de vérité du **74 LS 75**
- ☞ Brochage et table de vérité du **74 LS 76**



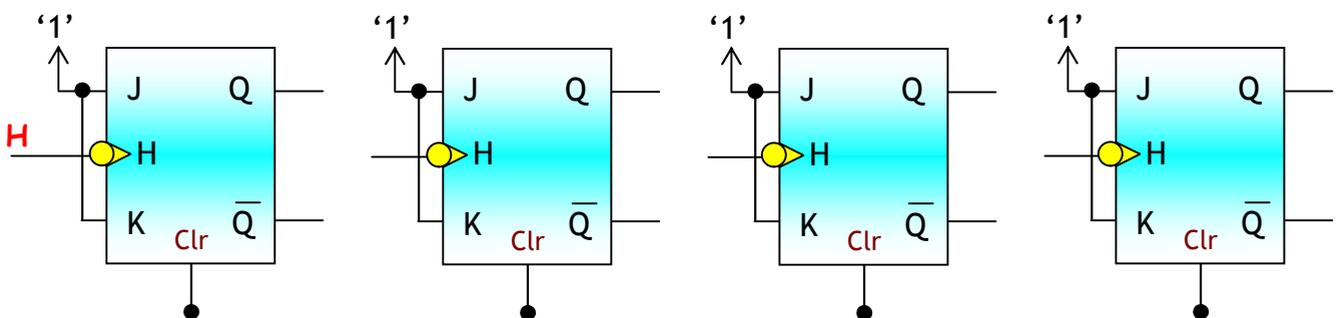
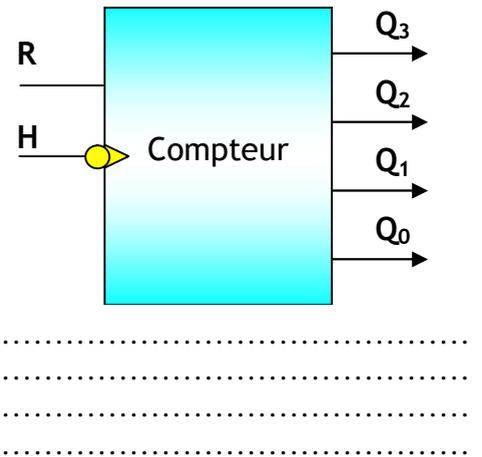
3. Compteur asynchrone modulo $N \neq 2^n$:

Exemple : Compteur asynchrone modulo 10 ($2^3 < 10 < 2^4$)

Pour le réaliser, il y a deux étapes :

- ☞ On cherche d'abord la puissance de 2 immédiatement supérieure à N.
- ☞ L'exposant de cette puissance de 2 donne le nombre de bascules JK à monter en cascade.
- ☞ On détecte ensuite l'état N qui remettra le compteur à 0.

| R | H | q ₃ | q ₂ | q ₁ | q ₀ | Q ₃ | Q ₂ | Q ₁ | Q ₀ |
|---|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1 | x | x | x | x | x | 0 | 0 | 0 | 0 |
| 0 | ↓ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | ↓ | | | | | | | | |
| 0 | ↓ | | | | | | | | |
| 0 | ↓ | | | | | | | | |
| 0 | ↓ | | | | | | | | |
| 0 | ↓ | | | | | | | | |
| 0 | ↓ | | | | | | | | |
| 0 | ↓ | | | | | | | | |
| 0 | ↓ | | | | | | | | |
| 0 | ↓ | | | | | | | | |



On relie les sorties Q₁ et Q₃ (=1 pour N=10) aux entrées d'une porte AND dont la sortie commandera l'entrée CLR de chaque bascule.

| | | | | |
|-----------|----|----|----|----|
| $Q_1.Q_0$ | 00 | 01 | 11 | 10 |
| Q_2 | | | | |
| 0 | | | | |
| 1 | | | | |

$J_0 = \dots\dots\dots$

| | | | | |
|-----------|----|----|----|----|
| $Q_1.Q_0$ | 00 | 01 | 11 | 10 |
| Q_2 | | | | |
| 0 | | | | |
| 1 | | | | |

$K_0 = \dots\dots\dots$

| | | | | |
|-----------|----|----|----|----|
| $Q_1.Q_0$ | 00 | 01 | 11 | 10 |
| Q_2 | | | | |
| 0 | | | | |
| 1 | | | | |

$J_1 = \dots\dots\dots$

| | | | | |
|-----------|----|----|----|----|
| $Q_1.Q_0$ | 00 | 01 | 11 | 10 |
| Q_2 | | | | |
| 0 | | | | |
| 1 | | | | |

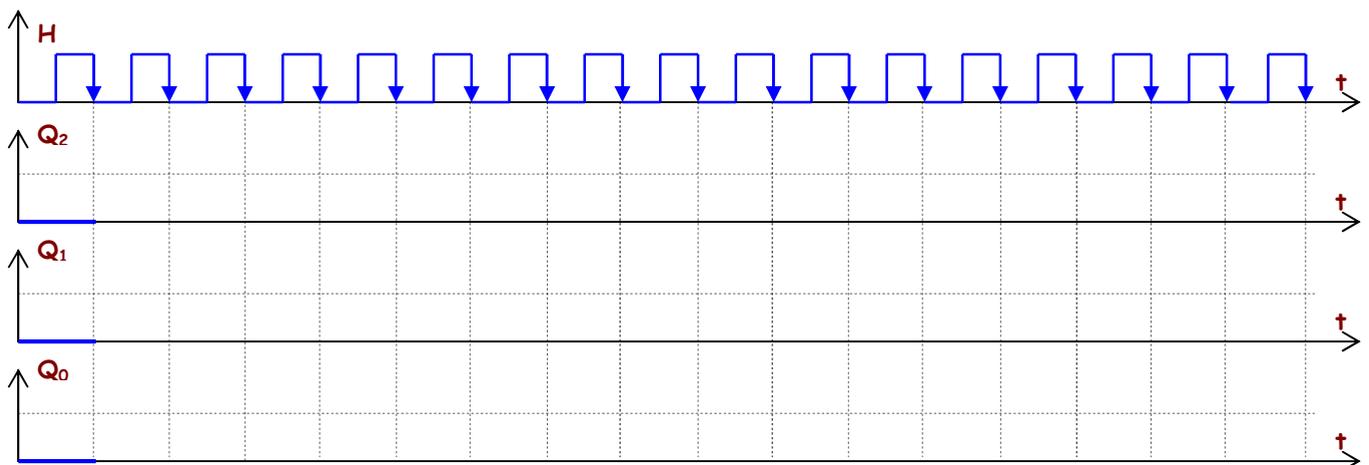
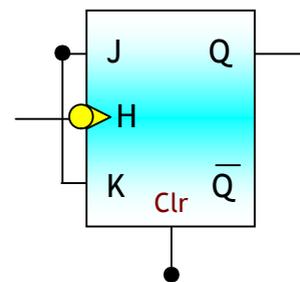
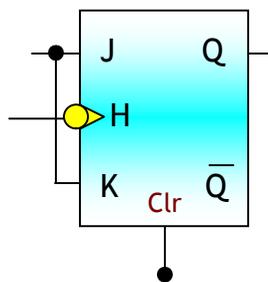
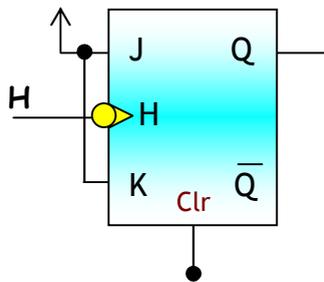
$K_1 = \dots\dots\dots$

| | | | | |
|-----------|----|----|----|----|
| $Q_1.Q_0$ | 00 | 01 | 11 | 10 |
| Q_2 | | | | |
| 0 | | | | |
| 1 | | | | |

$J_2 = \dots\dots\dots$

| | | | | |
|-----------|----|----|----|----|
| $Q_1.Q_0$ | 00 | 01 | 11 | 10 |
| Q_2 | | | | |
| 0 | | | | |
| 1 | | | | |

$K_2 = \dots\dots\dots$



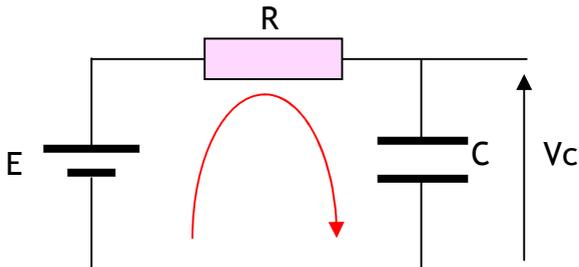
Compteurs intégrés (exercice):

- ☞ Brochage et table de vérité du **74 LS 90**
- ☞ Brochage et table de vérité du **74 LS 92**
- ☞ Brochage et table de vérité du **74 LS 93**

1. INTRODUCTION :

Dans les systèmes numériques, on a souvent besoin qu'une action soit effectuée pendant une durée déterminée ; on parle de temporisation. Aussi, on a besoin d'un signal périodique qui synchronise ou cadence les opérations d'un système séquentiel ; on parle de base de temps ou d'horloge (Clock).

2. LE CIRCUIT DE BASE-(Circuit RC) :

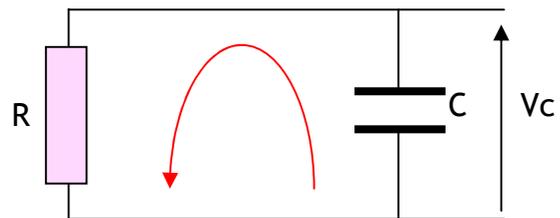
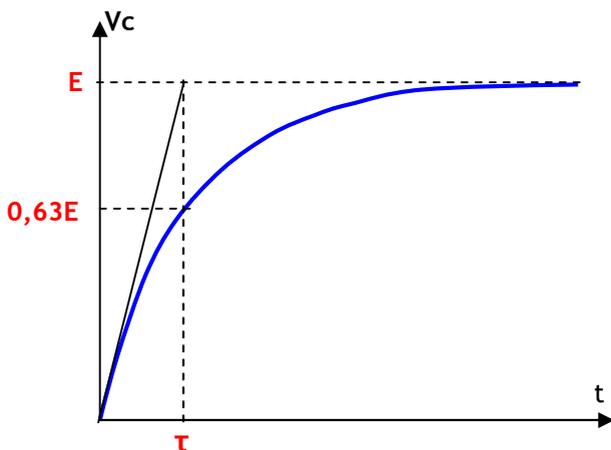


Dans un circuit RC, avec C déchargé ($V_c = 0V$), alimenté par une tension continue E, la tension V_c aux bornes de C augmente ; on dit que C se charge.

La loi de variation de V_c est de la forme :

$$V_c = E \cdot (1 - e^{-t/RC})$$

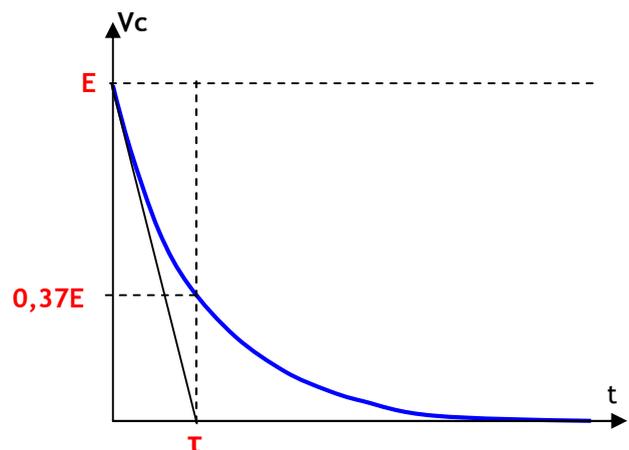
- ❖ Pour $t = 0 \rightarrow V_c = 0V$;
- ❖ Pour $t \rightarrow +\infty V_c \rightarrow E$;
- ❖ Pour $t = \tau V_c = 0.63E$ avec $\tau = RC$.



Dans un circuit RC, avec C déjà chargé ($V_c = E$), la tension aux bornes de C diminue ; on dit que C se décharge. la loi de variation de V_c est de la forme :

$$V_c = E \cdot (e^{-t/RC})$$

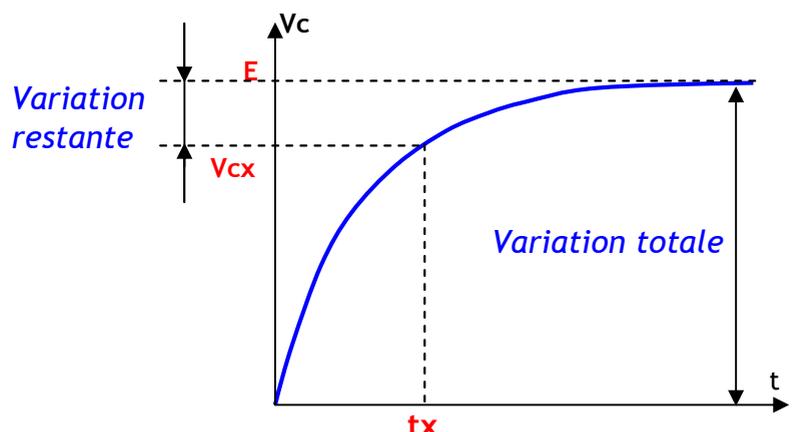
- ❖ Pour $t = 0 \rightarrow V_c = E$;
- ❖ Pour $t \rightarrow +\infty V_c \rightarrow 0$;
- ❖ Pour $t = \tau V_c = 0.37E$ avec $\tau = RC$.



On démontre que pour atteindre une certaine valeur V_{cx} , il faut un certain temps t_x , tel que :

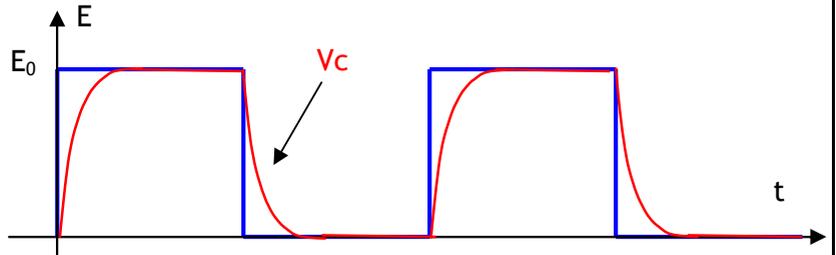
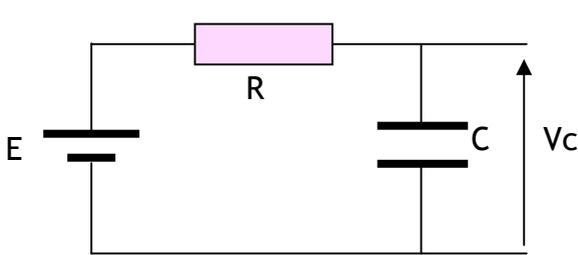
$$t_x = RC \cdot \ln \frac{\text{Variation totale}}{\text{Variation restante}}$$

$$t_x = RC \cdot \ln \frac{E}{E - V_{cx}}$$

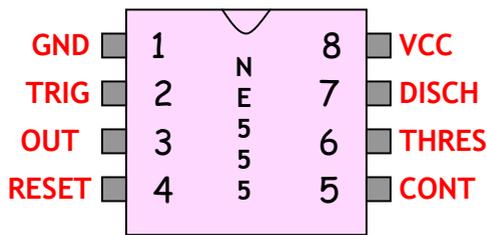


Où :

- ❖ ln est le logarithme népérien ;
- ❖ R est la valeur de la résistance ;
- ❖ C est la capacité du condensateur.

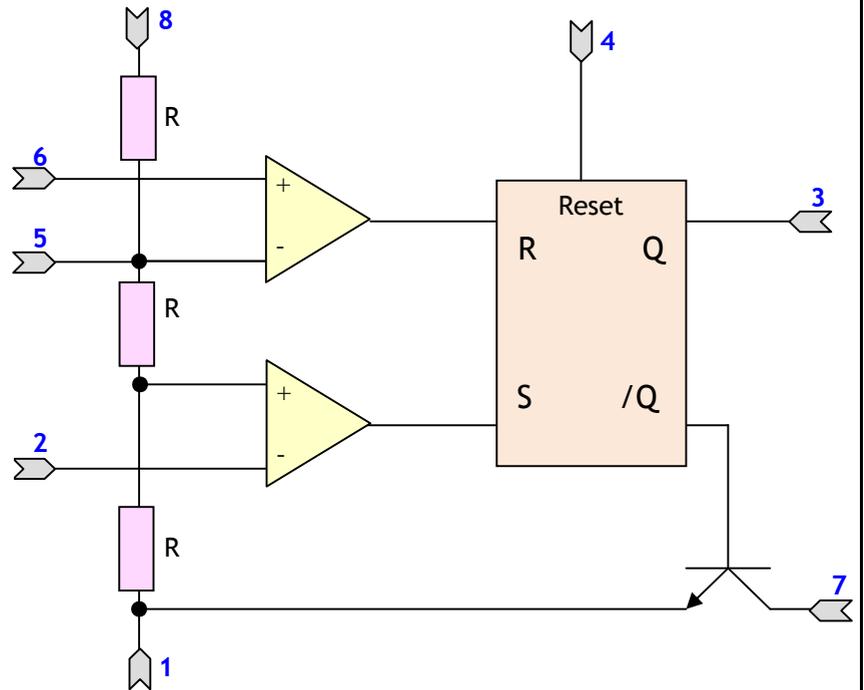


3. LE TEMPORISATEUR NE555 :

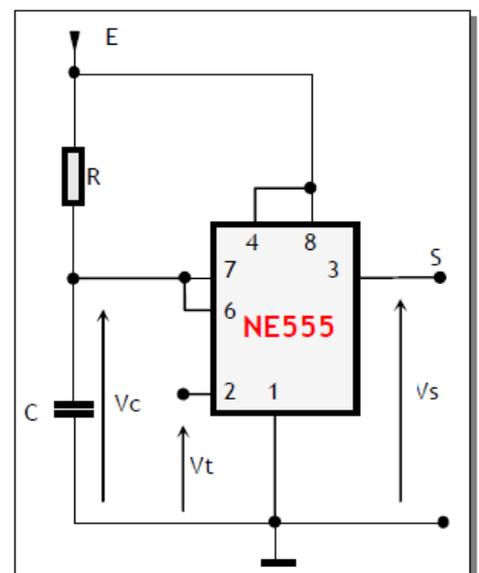
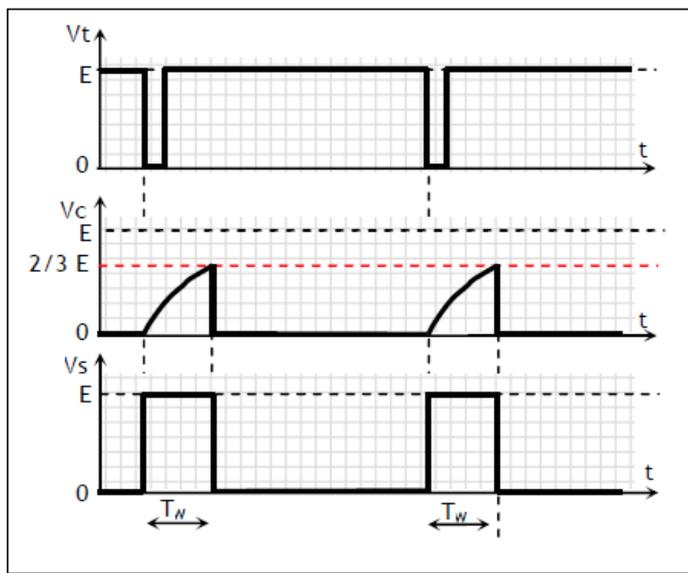


Comme l'indique son schéma interne, le NE555 est constitué de :

- ❖ 2 comparateurs dont les seuils sont fixés par le pont des 3 résistances R ;
- ❖ d'une bascule SR, avec une entrée de forçage à 0 (RESET) ;
- ❖ d'un transistor pour la décharge de condensateur externe.



3.1. FONCTIONNEMENT EN MONOSTABLE :

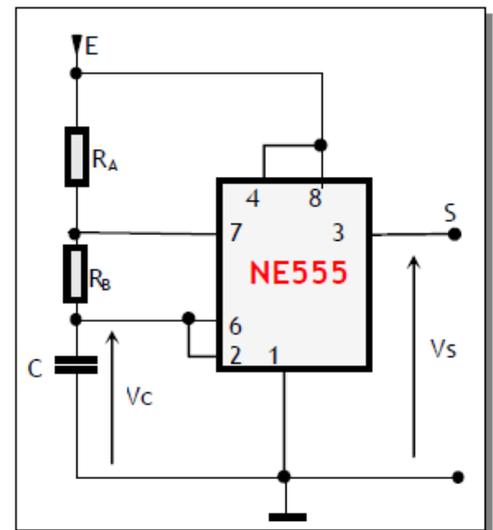
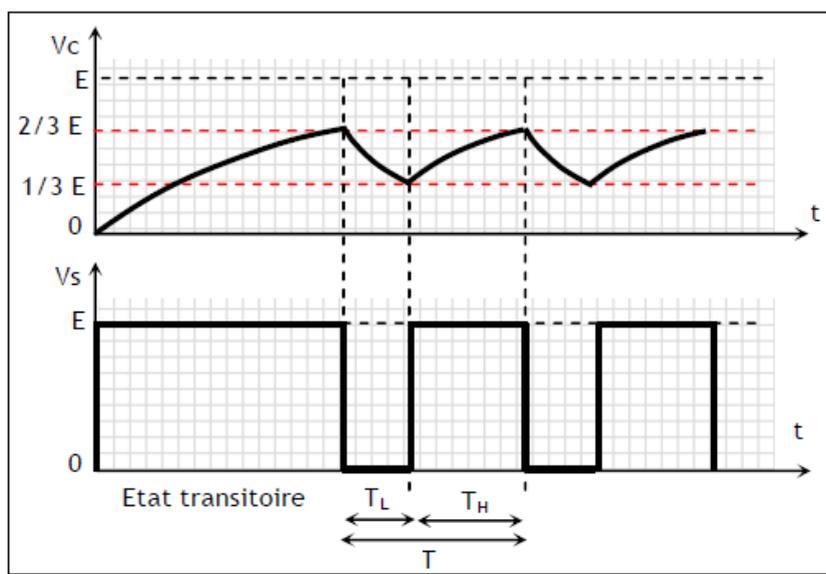


- ❖ On suppose qu'au démarrage, le condensateur C est déchargé et la sortie Vs à 0 ;
- ❖ L'entrée Vt est au repos au niveau logique 1 ; alors les variables du montage sont positionnées ainsi :
 - Les entrées de la bascule (S=0) et (R=0) → Etat mémoire de la bascule et Vs reste à 0 ;
 - Le transistor est saturé ; c'est l'état stable du monostable.

- ❖ Si V_t passe à 0 pendant une durée très courte par rapport à la durée qu'on veut du monostable, alors :
 - ($S=1$) et ($R=0$) → la sortie V_s est à 1 ;
 - Le transistor est bloqué, ce qui permet au condensateur C de se charger à travers R.
- ❖ Quand V_c , après un temps qui dépend de R et C, atteint $2/3$ de E, on a :
 - ($S=0$) et ($R=1$) → la sortie V_s est à 0 ;
 - Le transistor est saturé ; on revient à l'état de repos.
- ❖ L'expression de la durée de temporisation T_w est :

$$T_w = R.C.\ln 3 \quad \rightarrow \quad T_w = 1,1.R.C$$

3.2. FONCTIONNEMENT EN ASTABLE :



Il y a plusieurs variantes de cette fonction ; on va étudier la plus courante.

- ❖ On suppose qu'au démarrage, le condensateur C est déchargé et la sortie V_s à 1 ; alors les variables du montage sont positionnées ainsi :
 - Les entrées de la bascule ($S=0$) et ($R=0$) → Etat mémoire de la bascule et V_s reste à 1 ;
 - Le transistor est bloqué, ce qui permet au condensateur C de se charger à travers ($R_A + R_B$).
- ❖ Quand V_c , après un temps qui dépend de ($R_A + R_B$) et C, atteint $2/3$ de E, on a :
 - ($S=0$) et ($R=1$) → la sortie V_s est à 0 ;
 - Le transistor est saturé ; ce qui permet au condensateur C de se décharger à travers la résistance R_B . La tension V_c diminue alors.
- ❖ Quand V_c , après un temps qui dépend de R_B et C, atteint $1/3$ de E, on a :
 - ($S=1$) et ($R=0$) → la sortie V_s est à 1 ;
 - Le transistor est bloqué ; le cycle recommence. Il s'agit bien d'un oscillateur.
- ❖ L'expression de la période est $T = T_L + T_H$, avec :

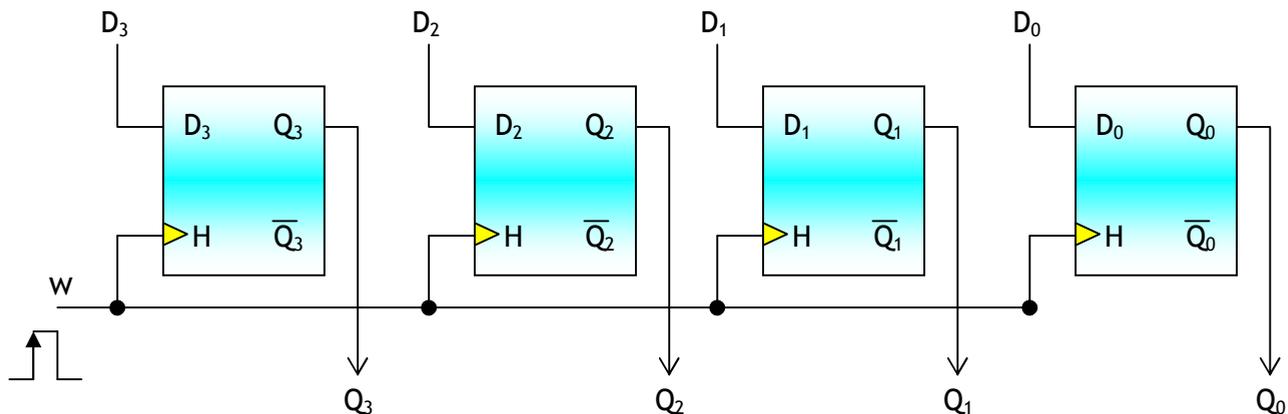
$$T_L = R_B.C.\ln 2 \quad \text{et} \quad T_H = (R_A + R_B).C.\ln 2 \quad \rightarrow \quad T = 0,7(R_A + 2.R_B).C$$

1. Présentation :

Un registre est constitué d'un assemblage de n bascules D permettant la mémorisation temporaire de n bits avec ou sans décalage. L'information est emmagasinée sur un signal de commande et ensuite conservée et disponible en lecture.

2. Le registre de mémorisation :

Les quatre bits D_0 , D_1 , D_2 et D_3 sont mémorisés dans les 4 bascules sur la même impulsion d'horloge, le mot binaire présenté en entrée est donc mémorisé.



Le signal d'horloge permet l'écriture (Write) des données D_0 , D_1 , D_2 et D_3 , dans les 4 bascules.

3. Les registres à décalage :

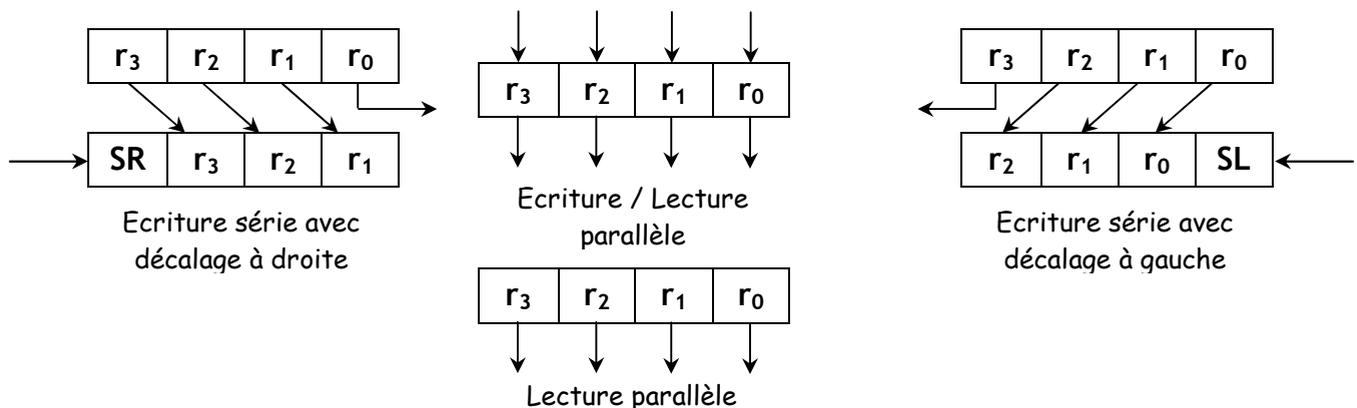
Dans un registre à décalage les bascules sont interconnectées de façon à ce que l'état logique de la bascule de rang i puisse être transmis à la bascule de rang $i+1$ (ou $i-1$) quand un signal d'horloge est appliqué à l'ensemble des bascules.

L'information peut être chargée de deux manières :

- ☞ Entrée parallèle : comme dans le cas d'un registre de mémorisation ;
- ☞ Entrée série : l'information est présentée séquentiellement bit après bit à l'entrée de la première bascule. Le décalage peut alors être vers la gauche ou la droite.

De même, l'information peut être lue en série ou en parallèle.

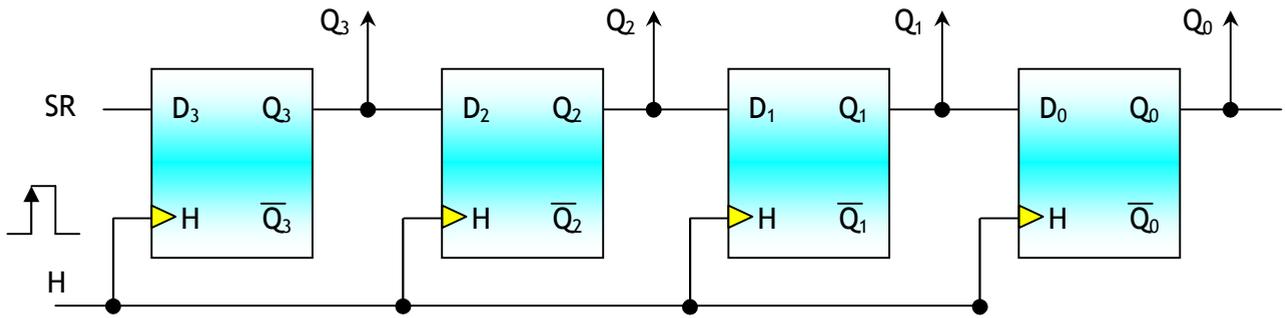
3.1. Les modes de fonctionnement d'un registre à décalage :



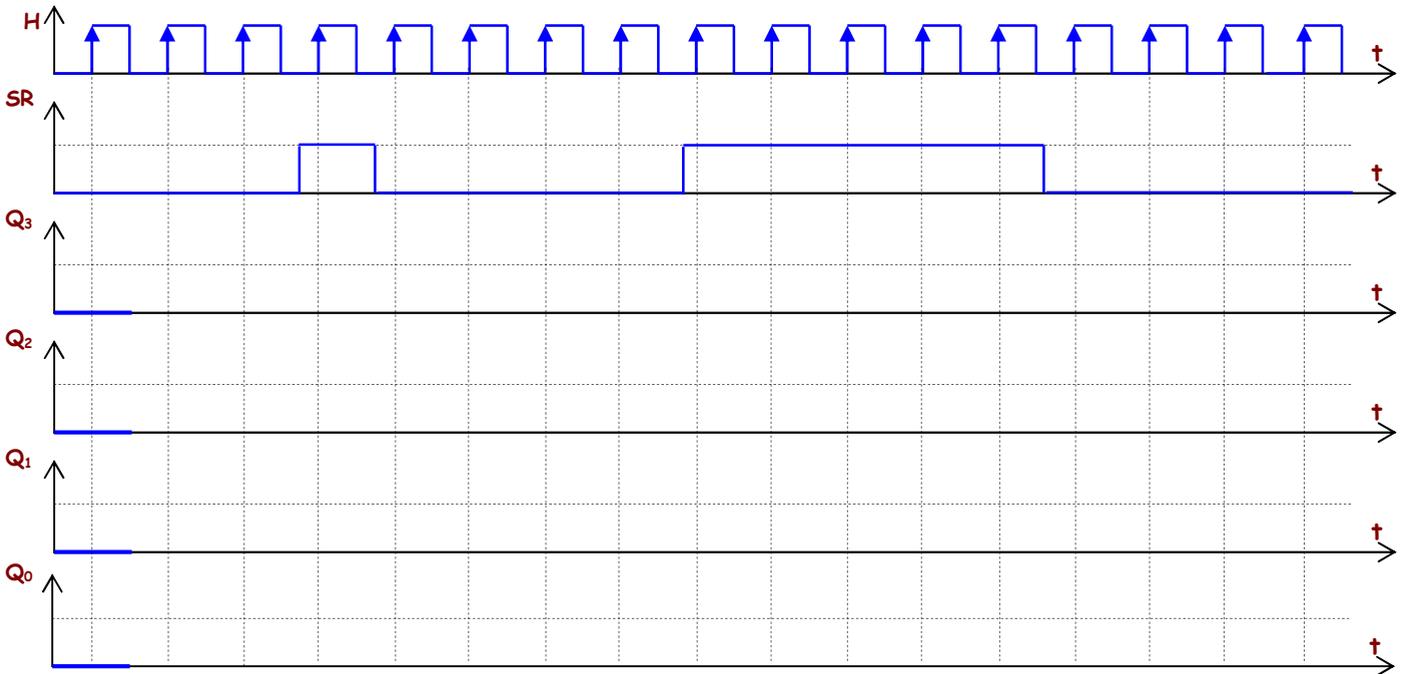
3.2. Exemple de registre à décalage :

L'information entrante qui est insérée bit par bit (en série) est reconstituée au rythme du signal d'horloge sur un format parallèle.

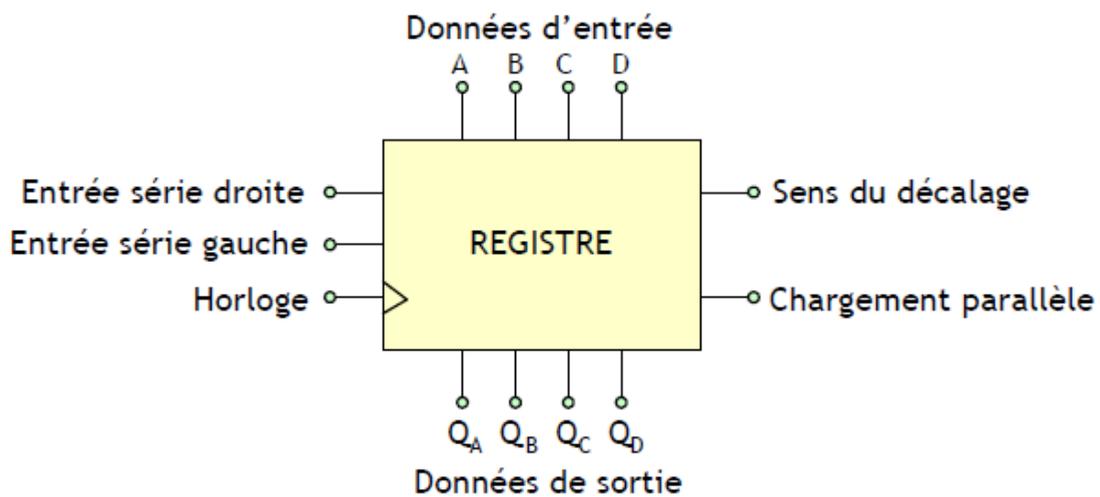
Exemple de registre à écriture série avec décalage à droite :



Exemple de chronogramme :



Un registre à décalage universel aura donc la structure suivante :



Parmi les registres universels, on trouve le 74194 qui est un registre à chargement parallèle ou série, avec la possibilité d'un déplacement de l'information vers la droite (Q_A vers Q_D) ou la gauche (Q_D vers Q_A).

Registres intégrés (exercice):

Brochage et table de vérité du **74 194**

1. Présentation :

Une cellule mémoire est un élément bistable capable d'emmagasiner puis de restituer un bit d'information ('0' ou '1'). EX : bascule, Disquette...

2. Mémoires électroniques :

2.1. Mémoire morte (ROM) :

C'est une mémoire à lecture seule, son contenu est non modifiable, elle reste inchangée même s'il y a coupure d'alimentation. On dit alors qu'elle est non volatile.

Les ROM sont utilisées pour stocker des informations figées telles que des programmes fixes dans des machines programmées ou les tables de conversion de données.

Le contenu est fixé à la construction ou par l'utilisateur et la disparition de l'alimentation électrique n'altère pas le contenu.

2.2. Mémoire vive (RAM) :

C'est une mémoire à accès aléatoire, on peut à chaque instant changer son contenu. Les RAM perdent leurs informations si on coupe l'alimentation, on dit qu'elles sont volatiles.

Dès qu'un système doit conserver temporairement des informations, la RAM trouve sa place. En informatique, elles sont largement mises en œuvre en quantités importantes (plus de 16 Mo en micro informatique et plusieurs centaines de méga octets en mini informatique).

2.3. Les mémoires programmables et effaçables par l'utilisateur :

Les mémoires programmables sont intermédiaires entre les RAM et les ROM. Leur contenu peut être défini par l'utilisateur et subsister sans alimentation électrique.

On en rencontre de différentes familles :

- ☞ Les **PROM** (Programmable ROM) : sont composées de fusibles que l'on peut détruire une seule fois ;
- ☞ Les **EPROM** (Erasable PROM) : ce sont des mémoires effaçables par ultraviolet et programmables électriquement ;
- ☞ Les **EEPROM** (Electrical Erasable PROM) : ce sont des mémoires effaçables et programmables électriquement.

3. Organisation interne :

3.1. Capacité :

C'est la quantité d'information qui peut être stockée dans la mémoire. Elle s'exprime en bits ou en mots de n bits. Par exemple :

$$64b, 4Kb, 8Ko \text{ (o : octet) avec } 1o = 8\text{bits} ; 1K = 2^{10} = 1024 ; 1M = 2^{20} = 1048576$$

3.2. Longueur de mot :

C'est la façon avec laquelle les bits sont organisés ou rangés, en général par mot de 8 bits ou de 4 bits.

- ❖ Un mot de 8bits : 1 Octet
- ❖ Un mot de 4bits : 1 Quartet

3.3. Adresse :

Pour identifier les mots on donne à chacun une adresse, on dit alors case mémoire d'adresse 40, case mémoire d'adresse FFetc.

3.4. Le temps d'accès :

C'est le temps qui s'écoule entre une demande d'information et le moment où elle est effectivement disponible.

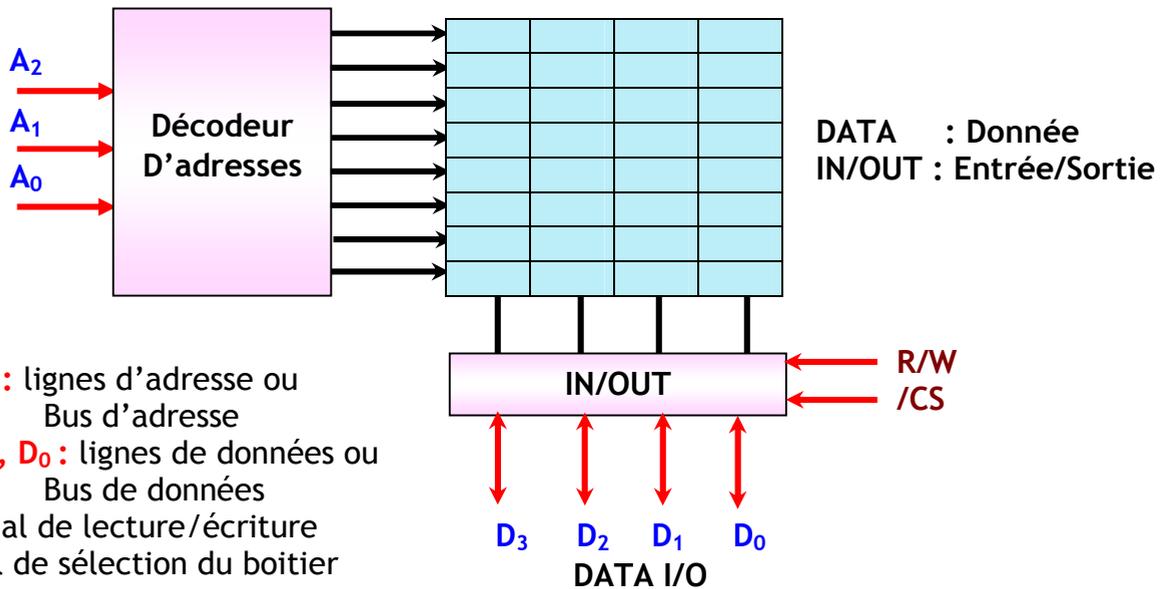
| | | | | |
|---|---|---|---|---|
| 7 | | | | |
| 6 | | | | |
| 5 | | | | |
| 4 | 1 | 0 | 1 | 0 |
| 3 | | | | |
| 2 | | | | |
| 1 | | | | |
| 0 | x | x | x | x |

8 mots de 4 bits ou 8 quartets
8 mots de 4 bits = 32 bits

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 3 | | | | | | | |
| 2 | | | | | | | |
| 1 | x | x | x | x | x | x | x |
| 0 | | | | | | | |

4 mots de 8 bits ou 4 octets
4 mots de 8 bits = 32 bits

3.5. Décodage d'adresse :



A_2, A_1, A_0 : lignes d'adresse ou Bus d'adresse
 D_3, D_2, D_1, D_0 : lignes de données ou Bus de données
 R/W : signal de lecture/écriture
 CS : signal de sélection du boîtier

Capacité = $2^{\text{nombre de lignes d'adresse}} \times \text{nombre de lignes de données}$

Pour l'exemple ci-dessus : Capacité = $2^3 \times 4 \text{ bits} = 8 \times 4 \text{ bits} = 32 \text{ bits}$
 Capacité = $8 \times 4 \text{ bits} = 8 \text{ q}$

On peut donc utiliser une mémoire soit en :

lecture :

- ✗ Appliquer le mot adresse sur le bus d'adresse ;
- ✗ Sélectionner le boîtier mémoire en appliquant un niveau logique bas sur la ligne CS ;
- ✗ Sélectionner le mode lecture en appliquant un niveau logique haut sur la ligne R/W ;

écriture :

- ✗ Appliquer le mot d'adresse sur le bus d'adresse ;
- ✗ Appliquer le mot de donnée sur le bus de données ;
- ✗ Sélectionner le boîtier mémoire en appliquant un niveau logique bas sur la ligne CS ;
- ✗ Sélectionner le mode écriture en appliquant un niveau logique bas sur la ligne R/W ;

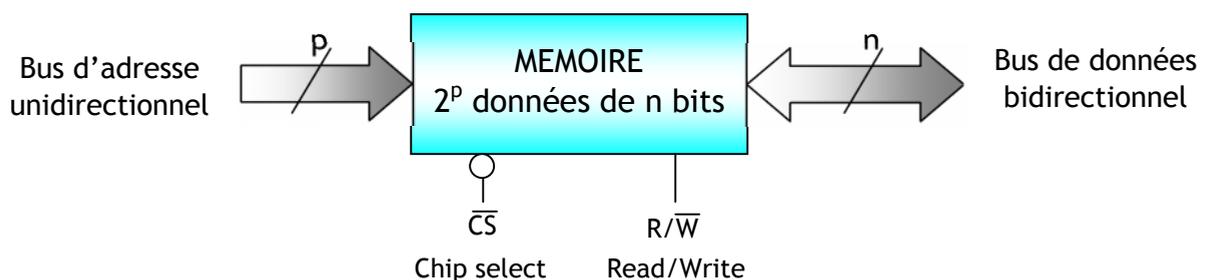


Schéma fonctionnel d'une mémoire

1. Exercice 'compteur synchrone modulo 16' :

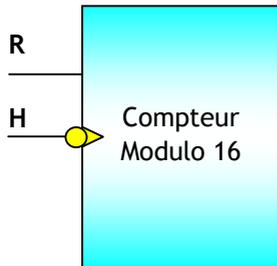
A- Table de vérité :

| q ₃ | q ₂ | q ₁ | q ₀ | Q ₃ | Q ₂ | Q ₁ | Q ₀ | J ₃ | K ₃ | J ₂ | K ₂ | J ₁ | K ₁ | J ₀ | K ₀ |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | | | | | | | | |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | | | | | | | | |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | | | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | | | | | | | | |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | | | | | | | | |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | | | | | | | | |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | | | | | | | | |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | | | | | | | | |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | | | | | | | | |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | | | | | | | | |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | | | | | | | | |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | | | | | | | | |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | | | | | | | | |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | | | | | |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | | | | | | | | |

| q | Q | J | K |
|---|---|---|---|
| 0 | 0 | | |
| 0 | 1 | | |
| 1 | 0 | | |
| 1 | 1 | | |

Table d'excitation de la bascule JK

B- Tableau de Karnaugh :



Il faut

| | | | | | |
|-----------------|----|----|----|----|----|
| $Q_1 \cdot Q_0$ | | 00 | 01 | 11 | 10 |
| $Q_3 \cdot Q_2$ | 00 | | | | |
| $Q_3 \cdot Q_2$ | 01 | | | | |
| $Q_3 \cdot Q_2$ | 11 | | | | |
| $Q_3 \cdot Q_2$ | 10 | | | | |

J₀ =

| | | | | | |
|-----------------|----|----|----|----|----|
| $Q_1 \cdot Q_0$ | | 00 | 01 | 11 | 10 |
| $Q_3 \cdot Q_2$ | 00 | | | | |
| $Q_3 \cdot Q_2$ | 01 | | | | |
| $Q_3 \cdot Q_2$ | 11 | | | | |
| $Q_3 \cdot Q_2$ | 10 | | | | |

K₀ =

| | | | | | |
|-----------------|----|----|----|----|----|
| $Q_1 \cdot Q_0$ | | 00 | 01 | 11 | 10 |
| $Q_3 \cdot Q_2$ | 00 | | | | |
| $Q_3 \cdot Q_2$ | 01 | | | | |
| $Q_3 \cdot Q_2$ | 11 | | | | |
| $Q_3 \cdot Q_2$ | 10 | | | | |

J₁ =

| | | | | | |
|-----------------|----|----|----|----|----|
| $Q_1 \cdot Q_0$ | | 00 | 01 | 11 | 10 |
| $Q_3 \cdot Q_2$ | 00 | | | | |
| $Q_3 \cdot Q_2$ | 01 | | | | |
| $Q_3 \cdot Q_2$ | 11 | | | | |
| $Q_3 \cdot Q_2$ | 10 | | | | |

K₁ =

| | | | | | |
|-----------------|----|----|----|----|----|
| $Q_1 \cdot Q_0$ | | 00 | 01 | 11 | 10 |
| $Q_3 \cdot Q_2$ | 00 | | | | |
| $Q_3 \cdot Q_2$ | 01 | | | | |
| $Q_3 \cdot Q_2$ | 11 | | | | |
| $Q_3 \cdot Q_2$ | 10 | | | | |

J₂ =

| | | | | | |
|-----------------|----|----|----|----|----|
| $Q_1 \cdot Q_0$ | | 00 | 01 | 11 | 10 |
| $Q_3 \cdot Q_2$ | 00 | | | | |
| $Q_3 \cdot Q_2$ | 01 | | | | |
| $Q_3 \cdot Q_2$ | 11 | | | | |
| $Q_3 \cdot Q_2$ | 10 | | | | |

K₂ =

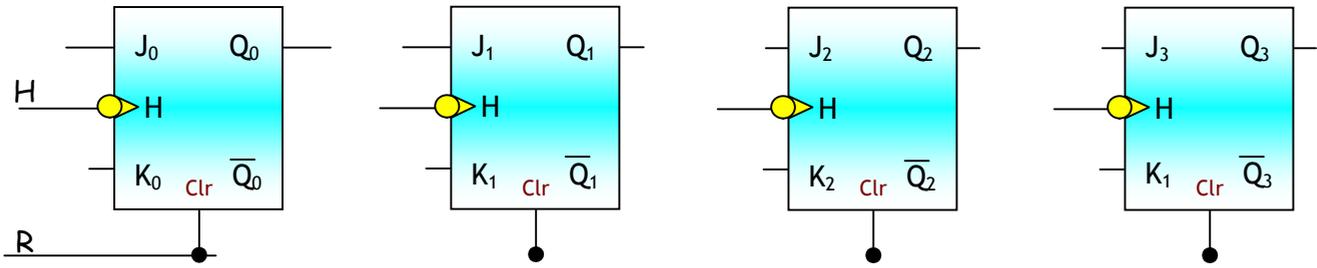
| | | | | | |
|-----------------|----|----|----|----|----|
| $Q_1 \cdot Q_0$ | | 00 | 01 | 11 | 10 |
| $Q_3 \cdot Q_2$ | 00 | | | | |
| $Q_3 \cdot Q_2$ | 01 | | | | |
| $Q_3 \cdot Q_2$ | 11 | | | | |
| $Q_3 \cdot Q_2$ | 10 | | | | |

J₃ =

| | | | | | |
|-----------------|----|----|----|----|----|
| $Q_1 \cdot Q_0$ | | 00 | 01 | 11 | 10 |
| $Q_3 \cdot Q_2$ | 00 | | | | |
| $Q_3 \cdot Q_2$ | 01 | | | | |
| $Q_3 \cdot Q_2$ | 11 | | | | |
| $Q_3 \cdot Q_2$ | 10 | | | | |

K₃ =

C- Logigramme :



2. Exercice 'compteur synchrone modulo 8 (0, 3, 6, 8, 10, 12, 13, 15)' :

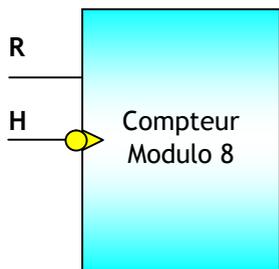
A- Table de vérité :

| q ₃ | q ₂ | q ₁ | q ₀ | Q ₃ | Q ₂ | Q ₁ | Q ₀ | J ₃ | K ₃ | J ₂ | K ₂ | J ₁ | K ₁ | J ₀ | K ₀ |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| q | Q | J | K |
|---|---|---|---|
| 0 | 0 | | |
| 0 | 1 | | |
| 1 | 0 | | |
| 1 | 1 | | |

Table d'excitation de la bascule JK

B- Tableau de Karnaugh :



Il faut

| | | | | | |
|-----------------|-----------------|----|----|----|----|
| $Q_1 \cdot Q_0$ | $Q_3 \cdot Q_2$ | 00 | 01 | 11 | 10 |
| 00 | | | | | |
| 01 | | | | | |
| 11 | | | | | |
| 10 | | | | | |

$J_0 = \dots\dots\dots$

| | | | | | |
|-----------------|-----------------|----|----|----|----|
| $Q_1 \cdot Q_0$ | $Q_3 \cdot Q_2$ | 00 | 01 | 11 | 10 |
| 00 | | | | | |
| 01 | | | | | |
| 11 | | | | | |
| 10 | | | | | |

$K_0 = \dots\dots\dots$

| | | | | | |
|-----------------|-----------------|----|----|----|----|
| $Q_1 \cdot Q_0$ | $Q_3 \cdot Q_2$ | 00 | 01 | 11 | 10 |
| 00 | | | | | |
| 01 | | | | | |
| 11 | | | | | |
| 10 | | | | | |

$J_1 = \dots\dots\dots$

| | | | | | |
|-----------------|-----------------|----|----|----|----|
| $Q_1 \cdot Q_0$ | $Q_3 \cdot Q_2$ | 00 | 01 | 11 | 10 |
| 00 | | | | | |
| 01 | | | | | |
| 11 | | | | | |
| 10 | | | | | |

$K_1 = \dots\dots\dots$

| | | | | | |
|-----------------|-----------------|----|----|----|----|
| $Q_1 \cdot Q_0$ | $Q_3 \cdot Q_2$ | 00 | 01 | 11 | 10 |
| 00 | | | | | |
| 01 | | | | | |
| 11 | | | | | |
| 10 | | | | | |

$J_2 = \dots\dots\dots$

| | | | | | |
|-----------------|-----------------|----|----|----|----|
| $Q_1 \cdot Q_0$ | $Q_3 \cdot Q_2$ | 00 | 01 | 11 | 10 |
| 00 | | | | | |
| 01 | | | | | |
| 11 | | | | | |
| 10 | | | | | |

$K_2 = \dots\dots\dots$

| | | | | | |
|-----------------|-----------------|----|----|----|----|
| $Q_1 \cdot Q_0$ | $Q_3 \cdot Q_2$ | 00 | 01 | 11 | 10 |
| 00 | | | | | |
| 01 | | | | | |
| 11 | | | | | |
| 10 | | | | | |

$J_3 = \dots\dots\dots$

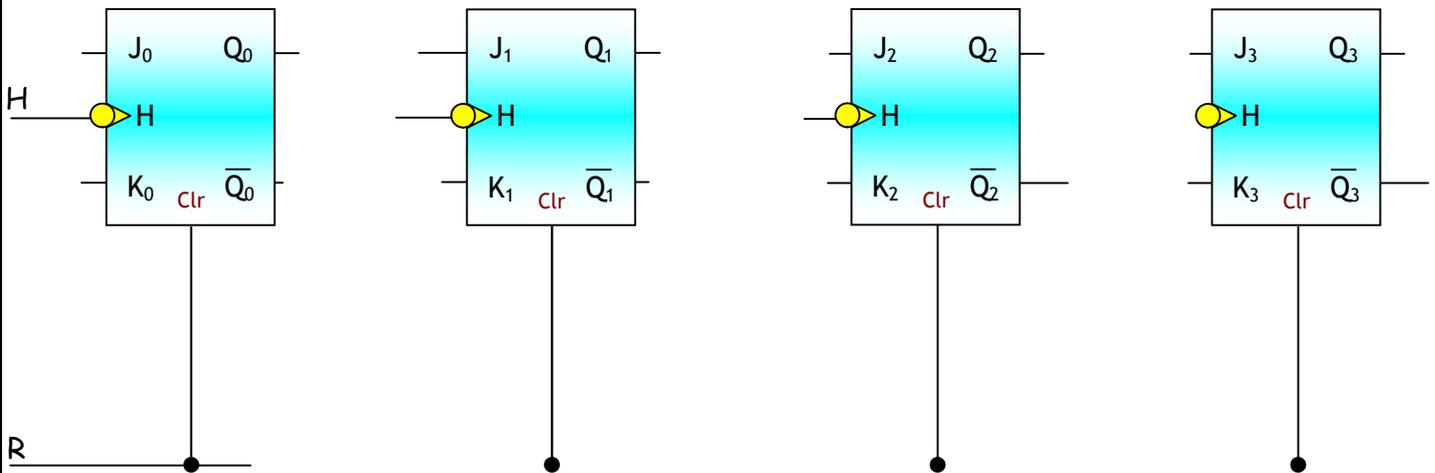
| | | | | | |
|-----------------|-----------------|----|----|----|----|
| $Q_1 \cdot Q_0$ | $Q_3 \cdot Q_2$ | 00 | 01 | 11 | 10 |
| 00 | | | | | |
| 01 | | | | | |
| 11 | | | | | |
| 10 | | | | | |

$K_3 = \dots\dots\dots$

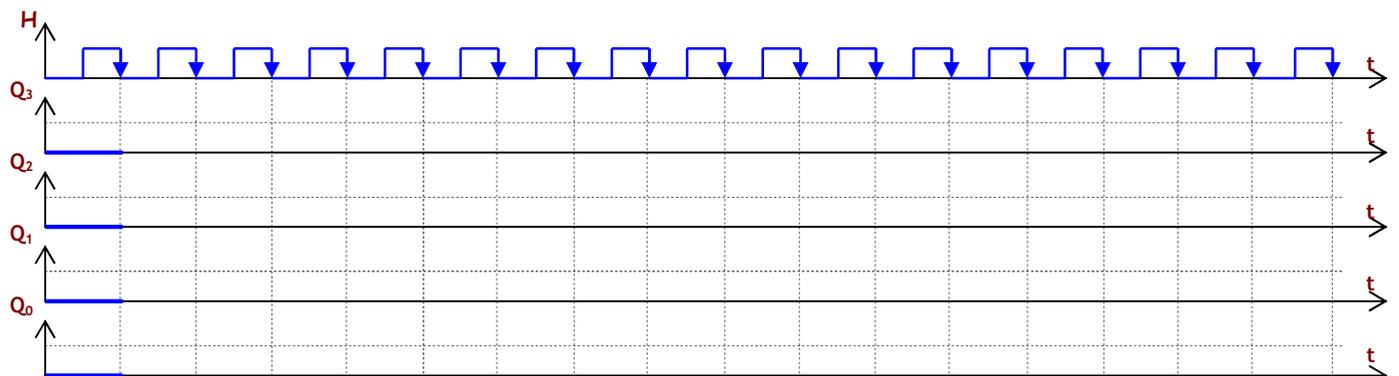
C- Compléter le tableau suivant :

| | | | | | | | | | | | | | | | | | |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Remise à 0 ou 8 ^{ème} impulsion | Q ₃ | | Q ₂ | | Q ₁ | | Q ₀ | | 1 ^{ère} impulsion | Q ₃ | | Q ₂ | | Q ₁ | | Q ₀ | |
| | 0 | | 0 | | 0 | | 0 | | | | | | | | | | |
| | J ₃ | K ₃ | J ₂ | K ₂ | J ₁ | K ₁ | J ₀ | K ₀ | | J ₃ | K ₃ | J ₂ | K ₂ | J ₁ | K ₁ | J ₀ | K ₀ |
| | | | | | | | | | | | | | | | | | |
| 2 ^{ème} impulsion | Q ₃ | | Q ₂ | | Q ₁ | | Q ₀ | | 3 ^{ème} impulsion | Q ₃ | | Q ₂ | | Q ₁ | | Q ₀ | |
| | | | | | | | | | | | | | | | | | |
| | J ₃ | K ₃ | J ₂ | K ₂ | J ₁ | K ₁ | J ₀ | K ₀ | | J ₃ | K ₃ | J ₂ | K ₂ | J ₁ | K ₁ | J ₀ | K ₀ |
| | | | | | | | | | | | | | | | | | |
| 4 ^{ème} impulsion | Q ₃ | | Q ₂ | | Q ₁ | | Q ₀ | | 5 ^{ème} impulsion | Q ₃ | | Q ₂ | | Q ₁ | | Q ₀ | |
| | | | | | | | | | | | | | | | | | |
| | J ₃ | K ₃ | J ₂ | K ₂ | J ₁ | K ₁ | J ₀ | K ₀ | | J ₃ | K ₃ | J ₂ | K ₂ | J ₁ | K ₁ | J ₀ | K ₀ |
| | | | | | | | | | | | | | | | | | |
| 6 ^{ème} impulsion | Q ₃ | | Q ₂ | | Q ₁ | | Q ₀ | | 7 ^{ème} impulsion | Q ₃ | | Q ₂ | | Q ₁ | | Q ₀ | |
| | | | | | | | | | | | | | | | | | |
| | J ₃ | K ₃ | J ₂ | K ₂ | J ₁ | K ₁ | J ₀ | K ₀ | | J ₃ | K ₃ | J ₂ | K ₂ | J ₁ | K ₁ | J ₀ | K ₀ |
| | | | | | | | | | | | | | | | | | |

D- Logigramme :

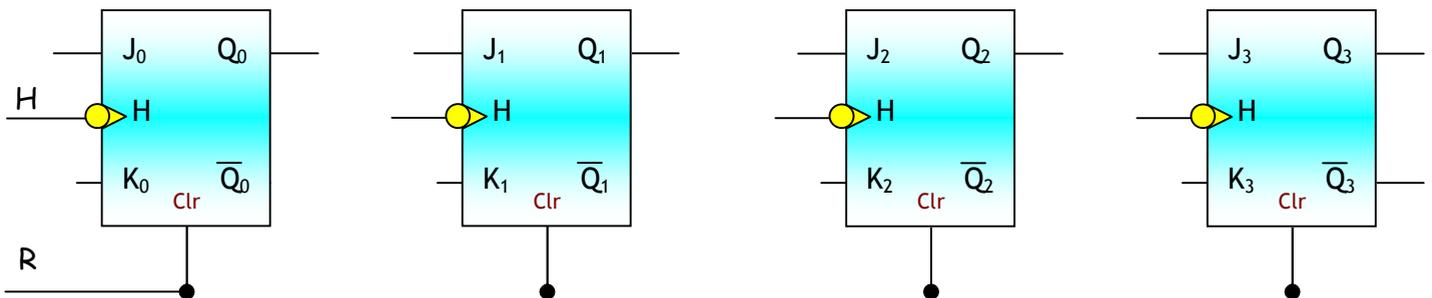


E- Chronogramme :



| | | | | | | | | | |
|----------------------------|----------------|----------------|----------------|----------------|----------------------------|----------------|----------------|----------------|----------------|
| 2 ^{ème} impulsion | Q ₃ | Q ₂ | Q ₁ | Q ₀ | 3 ^{ème} impulsion | Q ₃ | Q ₂ | Q ₁ | Q ₀ |
| | J ₃ | K ₃ | J ₂ | K ₂ | | J ₁ | K ₁ | J ₀ | K ₀ |
| | | | | | | | | | |
| 4 ^{ème} impulsion | Q ₃ | Q ₂ | Q ₁ | Q ₀ | 5 ^{ème} impulsion | Q ₃ | Q ₂ | Q ₁ | Q ₀ |
| | J ₃ | K ₃ | J ₂ | K ₂ | | J ₁ | K ₁ | J ₀ | K ₀ |
| | | | | | | | | | |
| 6 ^{ème} impulsion | Q ₃ | Q ₂ | Q ₁ | Q ₀ | 7 ^{ème} impulsion | Q ₃ | Q ₂ | Q ₁ | Q ₀ |
| | J ₃ | K ₃ | J ₂ | K ₂ | | J ₁ | K ₁ | J ₀ | K ₀ |
| | | | | | | | | | |
| 8 ^{ème} impulsion | Q ₃ | Q ₂ | Q ₁ | Q ₀ | 9 ^{ème} impulsion | Q ₃ | Q ₂ | Q ₁ | Q ₀ |
| | J ₃ | K ₃ | J ₂ | K ₂ | | J ₁ | K ₁ | J ₀ | K ₀ |
| | | | | | | | | | |

D- Logigramme :



E- Chronogramme :

