



Faculté
Polydisciplinaire
de Khouribga

Université Sultan
Moulay Slimane
Béni Mellal



Cours d'électronique Numérique

Chapitre 5 : Circuits Séquentiels

FPK, Filière SMI S₃

Pr. A. LAHRECH

Année 2020-2021



Circuits Séquentiels



FPK

Filière SMI S3

Introduction

I. Les bascules

I.1. Bascules RS

I.2. Bascules JK

I.3. Bascules D

II. Les compteurs/décompteurs Asynchrone

III. Les compteurs/ décompteurs synchrone

IV. Les registres



Circuits Séquentiels



FPK

Filière SMI S3

Introduction

Un circuit séquentiel est un circuit dont l'état des sorties dépend non seulement des entrées à l'instant présent mais également de l'état des sorties à l'instant antérieur.

Les circuits séquentiels fondamentaux sont :

- Bascules ;
- Compteurs / décompteurs;
- Registres ;

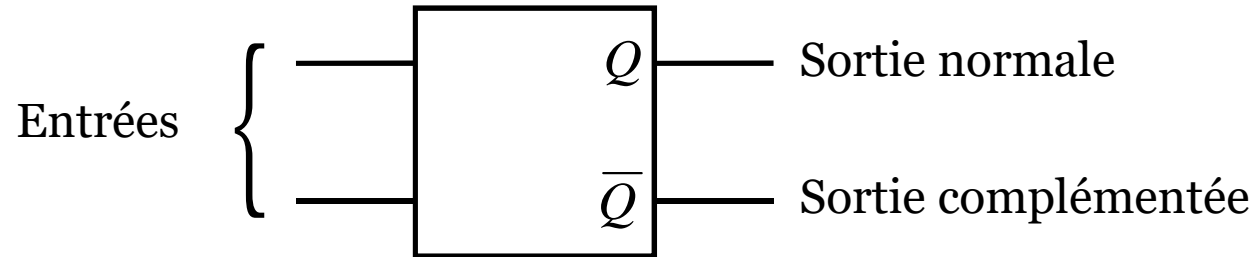
On distingue deux types de circuit séquentiel

- Circuit séquentiel asynchrone : l'état logique de la sortie varie immédiatement après le changement des entrées.
- Circuit séquentiel synchrone : l'évolution des sorties est cadencée par les variations du signal de commande, le signal d'horloge.

Elément de mémorisation : Bascules

La bascule constitue le système séquentiel de base et permet de mémoriser un élément d'information élémentaire appelé bit.

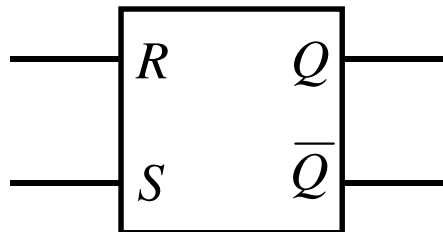
Symbole



La sortie normal Q détermine toujours l'état de la bascule. En pratique, la sortie complémentée \overline{Q} sert uniquement à simplifier le montage en évitant l'emploi des inverseurs.

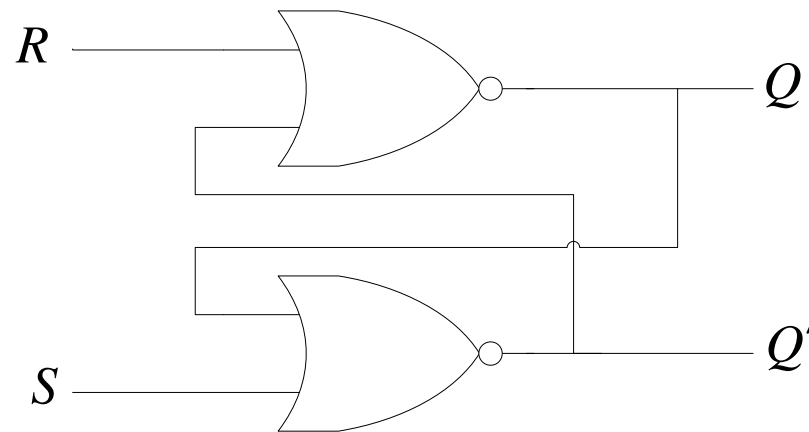
Basculé RS

La bascule RS est le circuit séquentiel le plus simple. C'est une bascule asynchrone.



⊕ Bascules RS avec des portes NOR

Présentation de l'état des sorties



Les états stables des sorties représentées par les variables Q et Q' sont données par :

$$Q = \overline{R + Q'} = \overline{R} \cdot \overline{Q'} \quad ; \quad Q' = \overline{S + Q} = \overline{S} \cdot \overline{Q}$$

Principe de fonctionnement :

- Si $R = 0$ et $S = 1 \Rightarrow Q = 1$ et $Q' = 0$



Circuits Séquentiels



Principe de fonctionnement de la bascule RS (suite) :

- Si $R = 0$ et $S = 0 \Rightarrow Q = \overline{Q'}$ et $Q' = \overline{Q}$

Ce qui donne deux solutions possibles :
$$\begin{cases} Q = 0 \text{ et } Q' = 1; \\ Q = 1 \text{ et } Q' = 0; \end{cases}$$

- Si $R = 1$ et $S = 0 \Rightarrow Q = 0$ et $Q' = 1$

- Si $R = 1$ et $S = 1 \Rightarrow Q = 0$ et $Q' = 0$

Afin de conserver une sortie complémentaire quelque soit la combinaison d'entrée, il convient d'interdire la combinaison $R = S = 1$

On peut définir la priorité d'une bascule comme l'état préférentiel affecté au cas $R=S=1$, ici on aurait donc une bascule RS à 1 prioritaire.

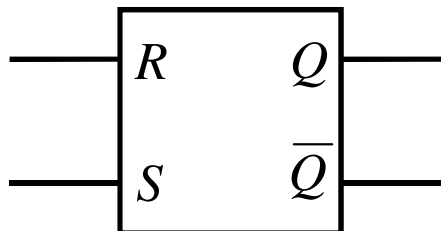
Bascules RS avec des portes NOR

Table de vérité d'une bascule RS NON-OU

| R | S | Q | Q' | |
|-----|-----|----------|------|------------------|
| 0 | 0 | inchangé | | Mémoire |
| 0 | 1 | 1 | 0 | Set (Mise à 1) |
| 1 | 0 | 0 | 1 | Reset (Mise à 0) |
| 1 | 1 | 0 | 0 | Interdit |

Si l'on interdit la combinaison $R = S = 1$ alors les sorties Q et Q' sont complémentaires : Les variables de sorties seront notées Q et \overline{Q}

Symbole :



⊗ Bascules RS avec des portes NOR

Notons Q_n l'état de la sortie Q avant la modification de l'une des variables d'entrée et Q_{n+1} l'état après cette modification.

La table de Karnaugh associée à la sortie Q_n qui permet la détermination de Q_{n+1} est donnée par :

| | | RS | | | |
|-------|---|------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_n | 0 | 0 | 1 | 0 | 0 |
| | 1 | 1 | 1 | 0 | 0 |

| R | S | Q_{n+1} |
|-----|-----|-----------|
| 0 | 0 | Q_n |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | interdit |

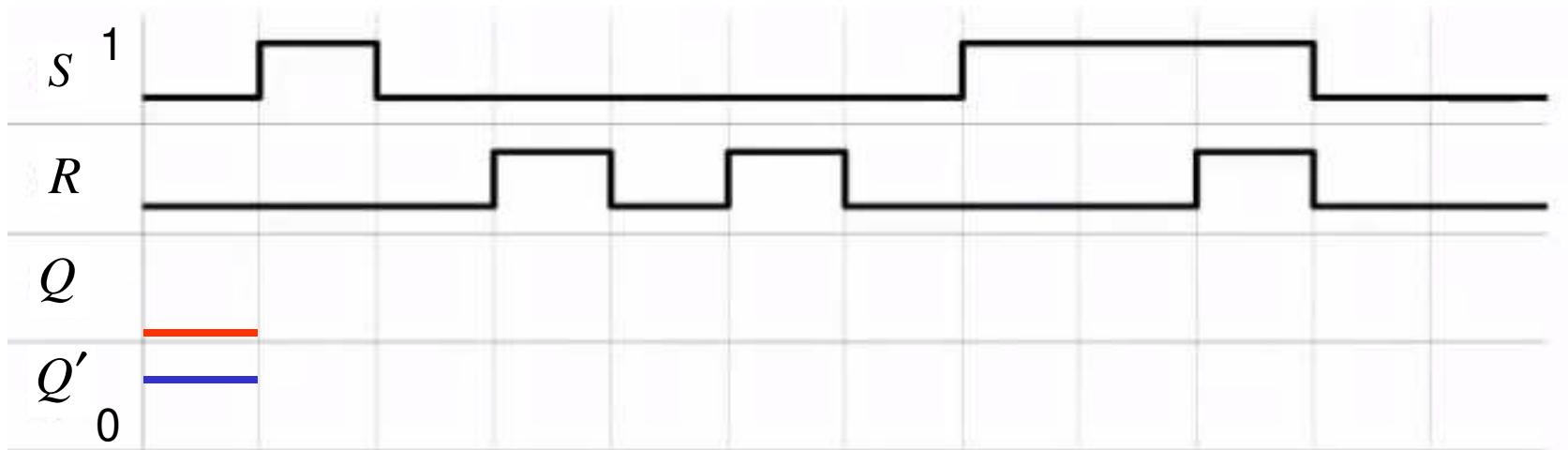
L'équation logique associée à la bascule RS :

$$Q_{n+1} = \overline{R}S + \overline{R}Q_n = \overline{R}(S + Q_n)$$

⊗ Bascules RS avec des portes NOR

Exemple de chronogrammes d'une bascule RS

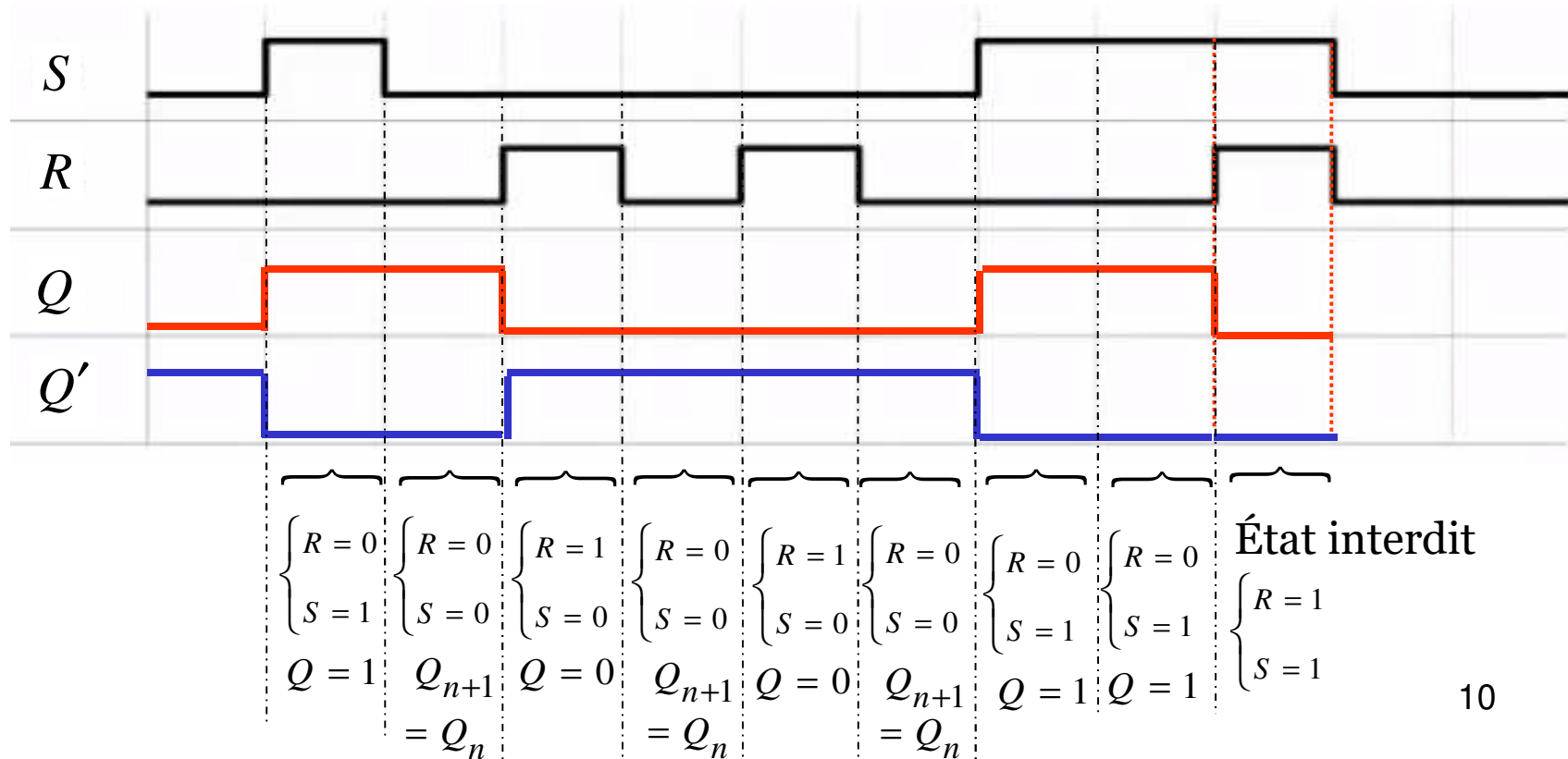
A $t=0$, Q est supposé être à 0, compléter les chronogrammes :



⊗ Bascules RS avec des portes NOR

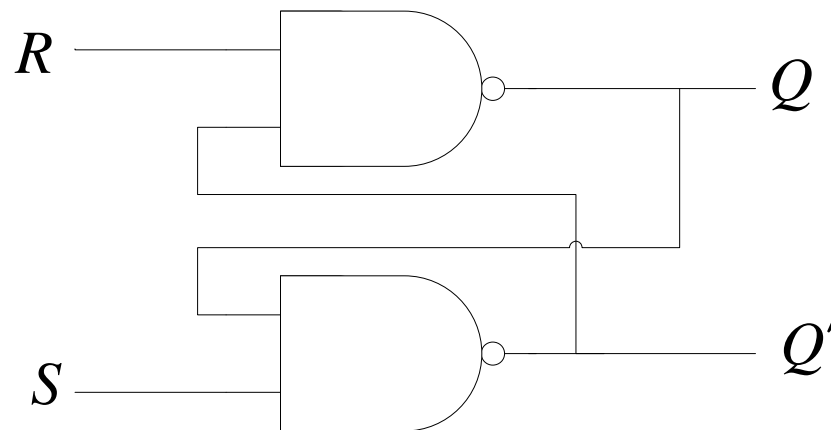
Exemple de chronogrammes d'une bascule RS

A $t = 0$, Q est supposé être à 0, compléter les chronogrammes :



🌐 Bascules RS avec des portes NAND

Présentation de l'états des sorties



Les états stables du circuit correspondent aux deux résultats des équations logiques suivantes :

$$Q = \overline{RQ'} = \overline{R} + \overline{Q'}$$

$$Q' = \overline{SQ} = \overline{S} + \overline{Q}$$



Circuits Séquentiels



FPK

Filière SMI S3

● **Bascules RS avec des portes NAND**

Principe de fonctionnement :

- Si $R = 0$ et $S = 0 \Rightarrow Q = 1$ et $Q' = 1$
- Si $R = 0$ et $S = 1 \Rightarrow Q = 1$ et $Q' = 0$
- Si $R = 1$ et $S = 0 \Rightarrow Q = 0$ et $Q' = 1$
- Si $R = 1$ et $S = 1 \Rightarrow Q = \overline{Q'}$ et $Q' = \overline{Q}$

Ce qui donne deux solutions possibles :
$$\begin{cases} Q = 0 \text{ et } Q' = 1; \\ Q = 1 \text{ et } Q' = 0; \end{cases}$$

Afin de conserver une sortie complémentaire quelque soit la combinaison d'entrée, il convient d'interdire la combinaison $R = S = 0$

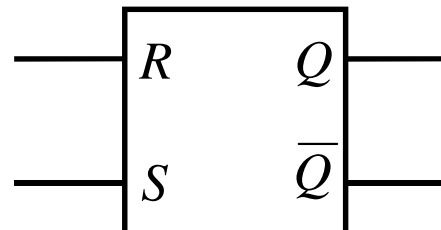
⊗ Bascules RS avec des portes NAND

Table de vérité d'une bascule RS NAND

| R | S | Q | Q' | |
|-----|-----|----------|------|----------|
| 0 | 0 | 1 | 1 | Interdit |
| 0 | 1 | 1 | 0 | Set |
| 1 | 0 | 0 | 1 | Reset |
| 1 | 1 | inchangé | | Mémoire |

Si l'on interdit la combinaison $R = S = 0$ alors les sorties Q et Q' sont complémentaires : Les variables de sorties seront notées Q et \overline{Q}

Symbole logique :



⊗ Bascules RS avec des portes NAND

Notons Q_n l'état de la sortie Q avant la modification de l'une des variables d'entrée et Q_{n+1} l'état après cette modification.

La table de Karnaugh associée à la sortie Q_n qui permet la détermination de Q_{n+1} est donnée par :

| | | RS | | | |
|-------|---|------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_n | 0 | 1 | 1 | 0 | 0 |
| | 1 | 1 | 1 | 1 | 0 |

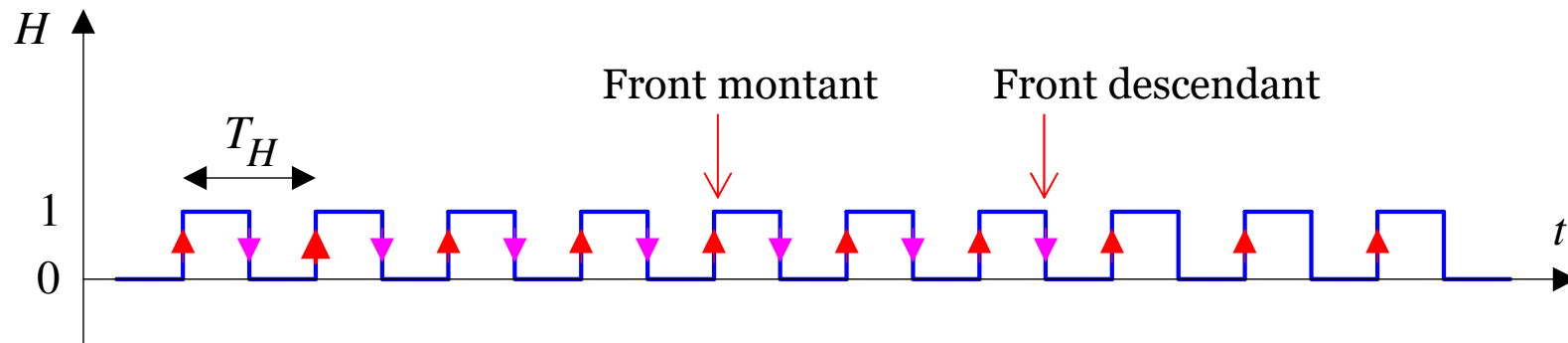
| R | S | Q_{n+1} |
|-----|-----|-----------|
| 0 | 0 | interdit |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | Q_n |

L'équation logique associée à la bascule RS :

$$Q_{n+1} = \bar{R} + SQ_n$$

Bascules synchrone

Toute bascule synchrone dispose d'une entrée d'horloge qui sert à commander le changement d'état de la sortie. Le changement d'état s'effectue pendant une transition appelée « front » montant ou descendant.



$H = 1$: Niveau haut de l'horloge

$H = 0$: Niveau bas de l'horloge

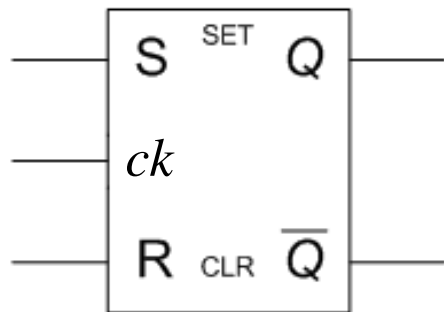
Front montant : l'instant où H bascule de 0 à 1

Front descendant : l'instant où H bascule de 1 à 0

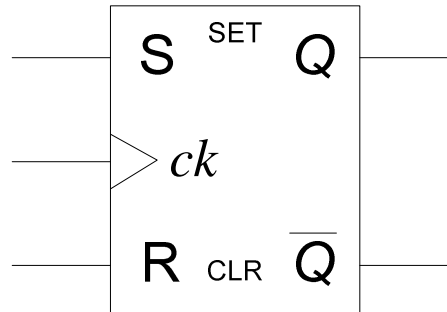
🌐 Bascule RS synchrone ou bascule RSH

La bascule RS synchrone dispose d'une entrée d'horloge CK

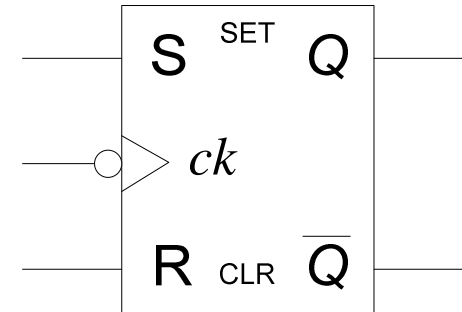
Les symboles des bascules RS à déclenchement sur niveau haut, sur front montant et sur front descendant de l'horloge :



Bascule RSH active sur niveau haut du signal d'horloge.

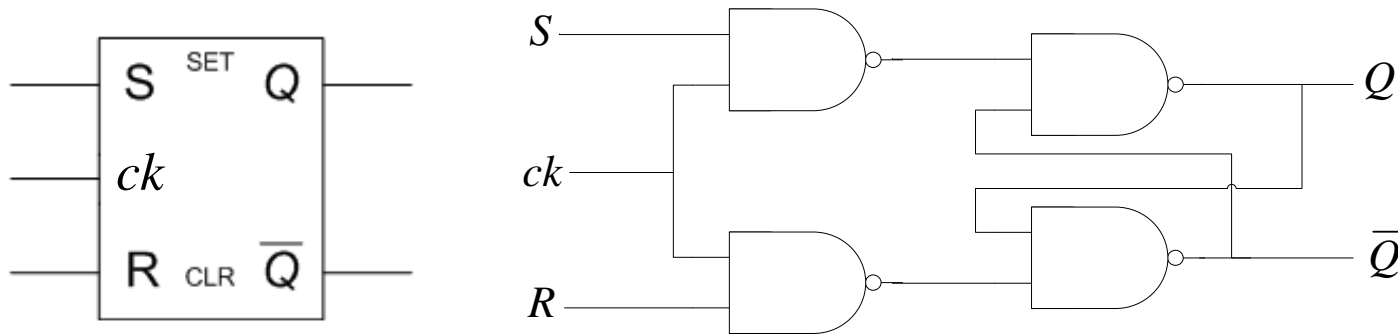


Bascule RSH à déclenchement sur front montant



Bascule RSH à déclenchement sur front descendant

⊕ Bascule RSH active sur niveau haut du signal d'horloge



- Quand le signal d'horloge ck est active sur niveau haut : la bascule RSH se comporte comme une bascule RS asynchrone.
- Quand ck est inactive (niveau 0) : verrouillage (fonction mémoire)

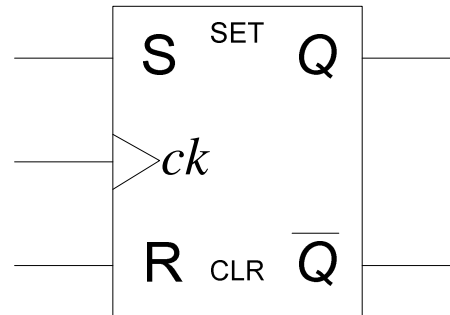
Table des états associée :

Bascule RSH active sur niveau haut :

$$\begin{cases} ck = 0 \Rightarrow \text{Etat mémoire (Bascule bloqué)} \\ ck = 1 \Rightarrow \text{Fonctionnement en bascule RS} \end{cases}$$

| ck | R | S | Q_{n+1} | |
|------|-----|-----|-----------|----------|
| 0 | × | × | Q_n | Mémoire |
| 1 | 0 | 0 | Q_n | Mémoire |
| 1 | 0 | 1 | 1 | Set |
| 1 | 1 | 0 | 0 | Reset |
| 1 | 1 | 1 | × | Interdit |

⊗ Bascule RSH à déclenchement sur front montant



Tables des états associées

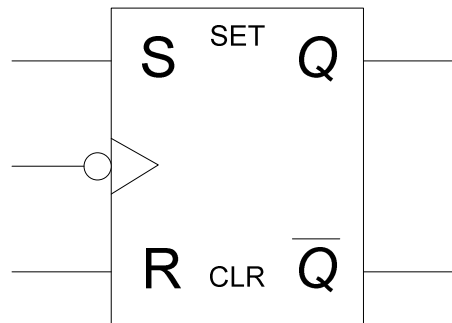
Bascule RSH synchronisé sur front montant

$$\begin{cases} ck = 0 \Rightarrow \text{Etat mémoire (Bascule bloqué)} \\ ck = 1 \Rightarrow \text{Etat mémoire (Bascule bloqué)} \end{cases}$$

Fonctionnement en bascule RS à chaque passage d'un front montant ↗

| ck | R | S | Q_{n+1} | |
|------|-----|-----|-----------|----------|
| 0 | × | × | Q_n | Mémoire |
| 1 | × | × | Q_n | Mémoire |
| ↗ | 0 | 0 | Q_n | Mémoire |
| ↗ | 0 | 1 | 1 | Set |
| ↗ | 1 | 0 | 0 | Reset |
| ↗ | 1 | 1 | × | Interdit |

⊗ Bascule RSH synchronisée sur front descendant



Tables des états associés

Bascule synchronisée sur front descendant

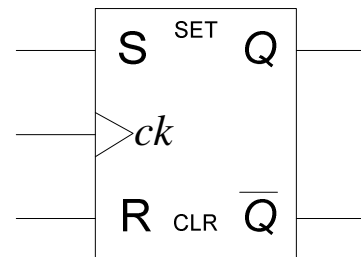
$\begin{cases} ck = 0 \Rightarrow \text{Etat mémoire (Bascule bloqué)} \\ ck = 1 \Rightarrow \text{Etat mémoire (Bascule bloqué)} \end{cases}$

Fonctionnement en bascule RS à chaque passage d'un front descendant \downarrow

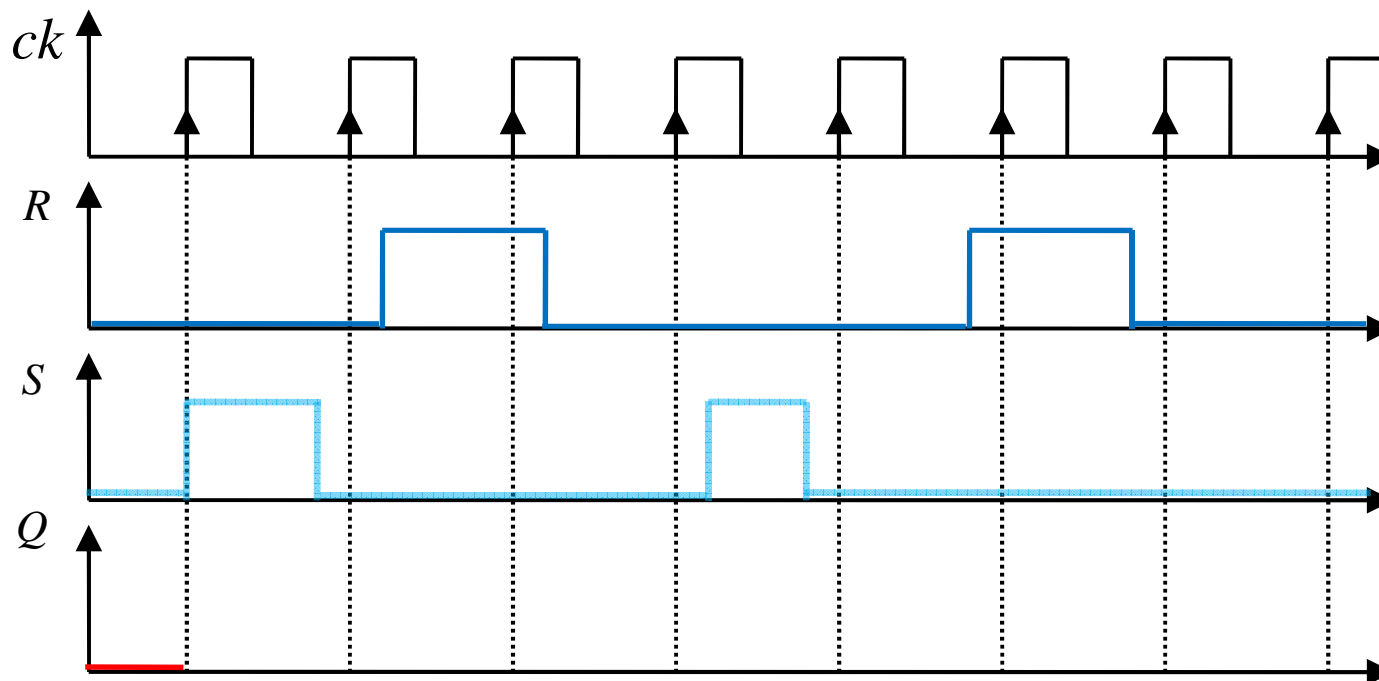
| ck | R | S | Q_{n+1} | |
|--------------|-----|-----|-----------|----------|
| 0 | × | × | Q_n | Mémoire |
| 1 | × | × | Q_n | Mémoire |
| \downarrow | 0 | 0 | Q_n | Mémoire |
| \downarrow | 0 | 1 | 1 | Set |
| \downarrow | 1 | 0 | 0 | Reset |
| \downarrow | 1 | 1 | × | Interdit |

Exemple de chronogrammes d'une bascule RSH

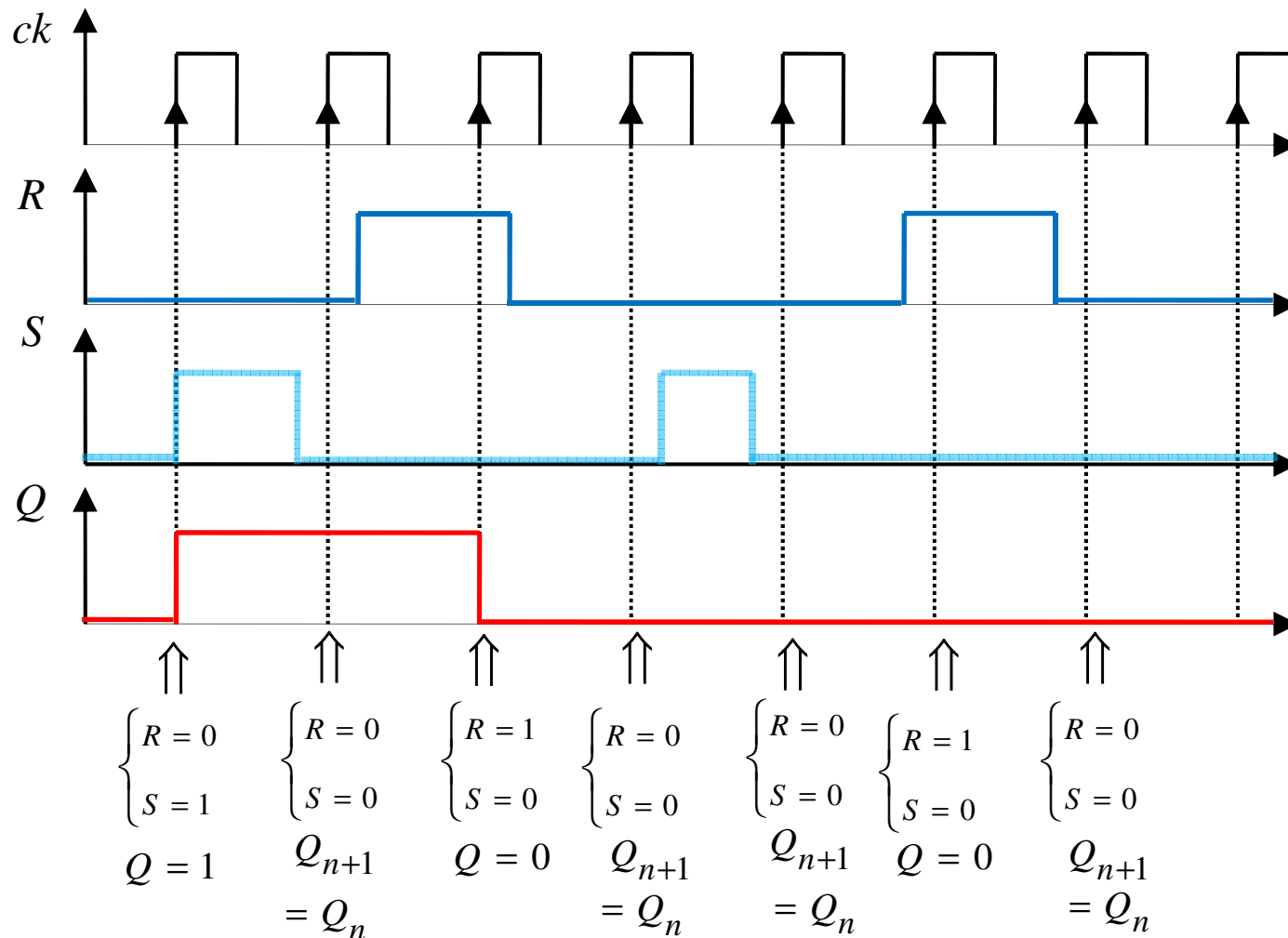
A $t = 0$, Q est supposé être à 0



Bascule RSH
synchronisée sur
front montant



Exemple de chronogrammes d'une bascule RSH





Circuits Séquentiels



FPK

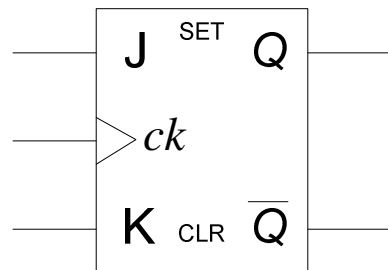
Filière SMI S3

Bascule JK synchrone

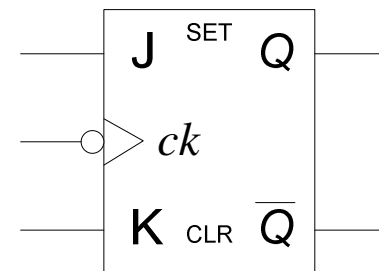
Le fonctionnement d'une bascule *JK* synchrone est analogue à celui d'une bascule RS, avec l'entrée *J* jouant le rôle de *S* et l'entrée *K* celui de *R* , mais aucun état n'est interdit.

La combinaison $(J, K) = (1, 1)$ est autorisé et la sortie $Q_{n+1} = \overline{Q}_n$ (inversion de l'état).

Symbole logique :



Bascule *JK* à déclenchement
sur front montant



Bascule *JK* à déclenchement
sur front descendant

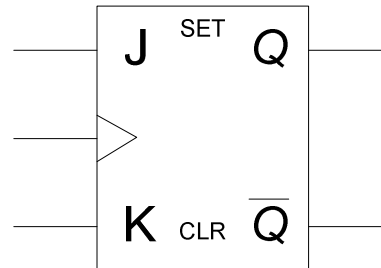
Principe de fonctionnement de la bascule synchrone :

- Si $K = 1$ et $J = 0 \Rightarrow$ Mise à zéro de la sortie Q
- Si $K = 0$ et $J = 1 \Rightarrow$ Mise à un de la sortie Q
- Si $K = 0$ et $J = 0 \Rightarrow$ La sortie Q conserve sa valeur (mémorisation de Q)
- Si $K = 1$ et $J = 1 \Rightarrow$ La sortie Q bascule en prenant l'état complémentaire

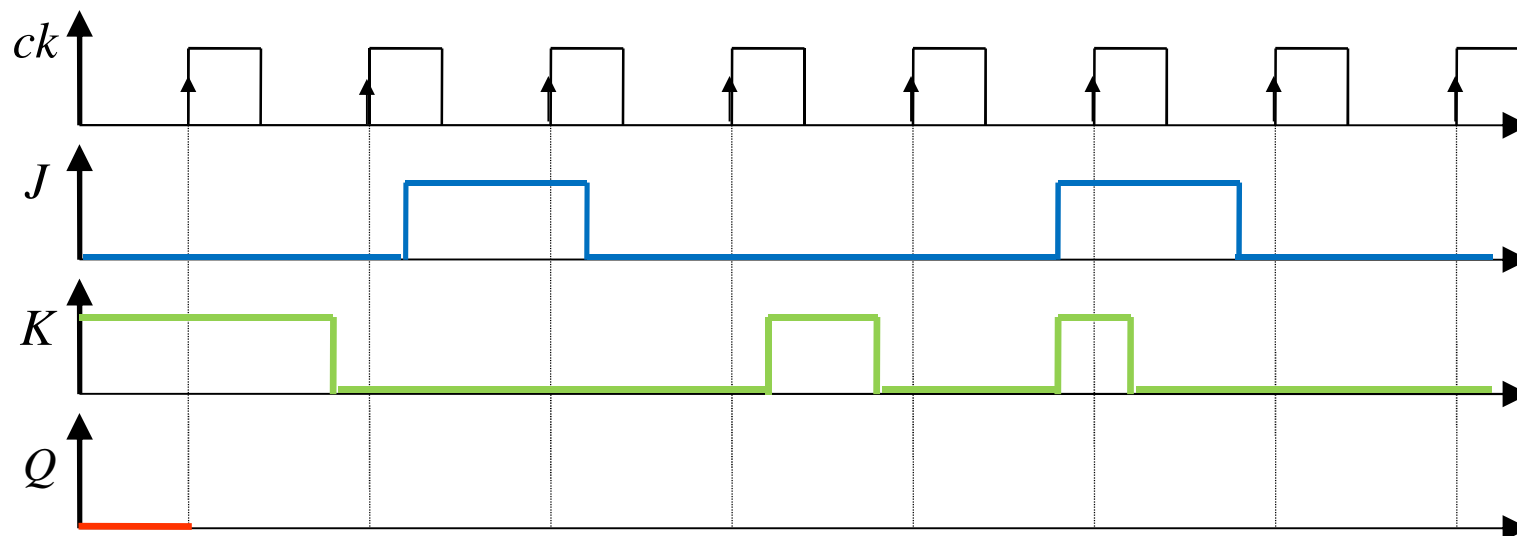
| J | K | Q_{n+1} | |
|-----|-----|------------------|----------|
| 0 | 0 | Q_n | Mémoire |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | $\overline{Q_n}$ | Interdit |

| ck | J | K | Q_{n+1} | |
|------------|----------|----------|------------------|-------------|
| 0 | \times | \times | Q_n | Mémoire |
| 1 | \times | \times | Q_n | Mémoire |
| \uparrow | 0 | 0 | Q_n | Mémoire |
| \uparrow | 0 | 1 | 0 | Rest |
| \uparrow | 1 | 0 | 1 | Set |
| \uparrow | 1 | 1 | $\overline{Q_n}$ | Basculement |

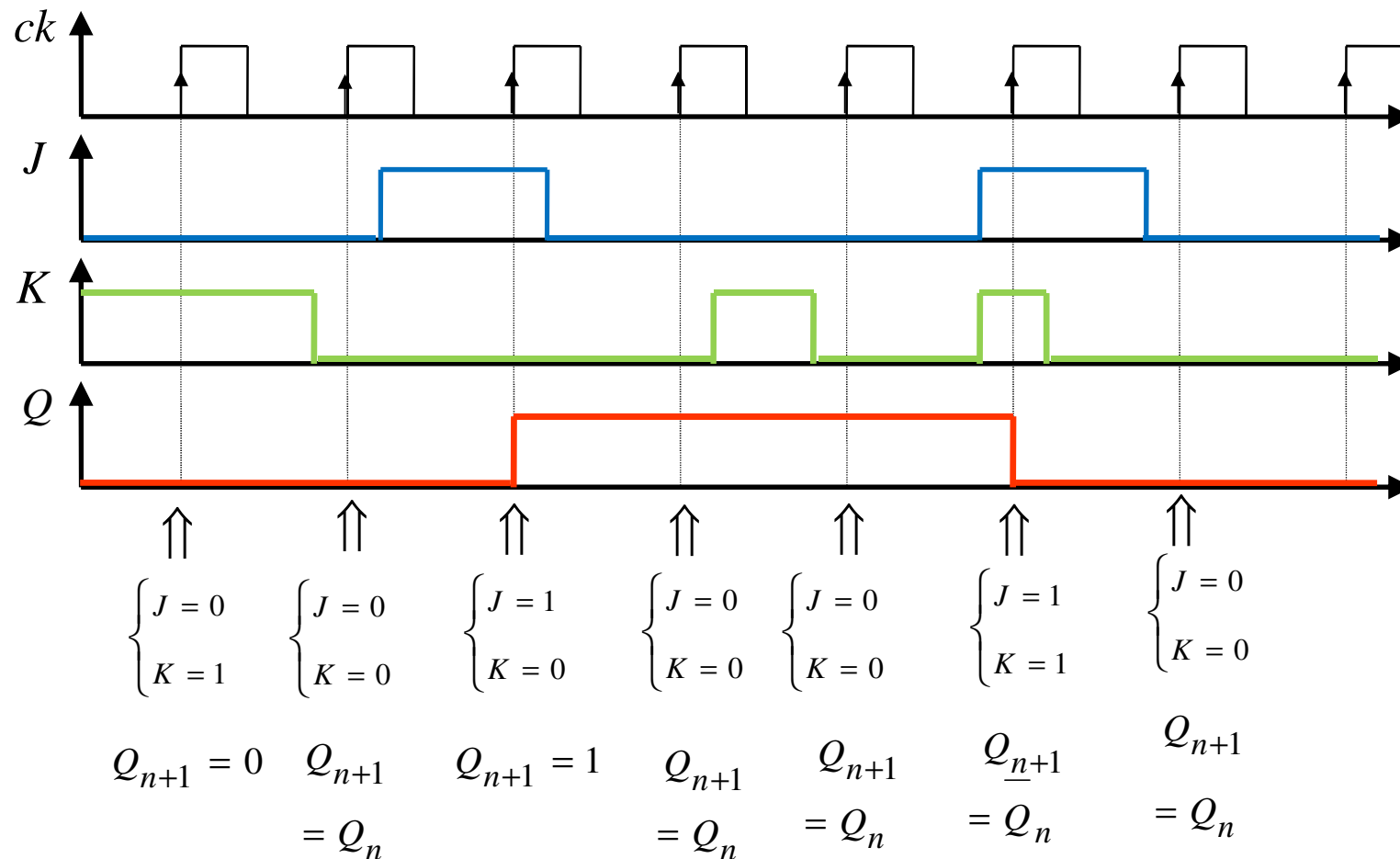
Exemple de chronogrammes d'une bascule JK



À $t = 0$, Q est supposé être à 0, compléter le chronogramme

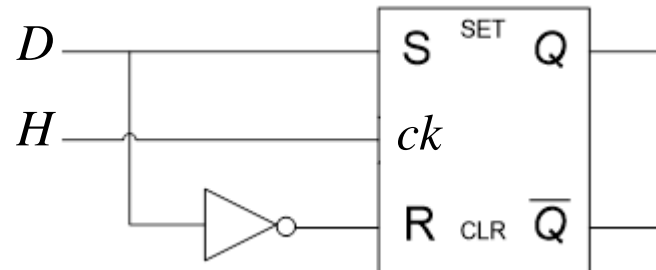
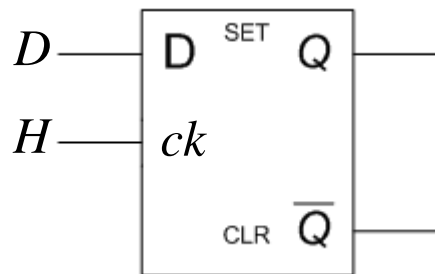


Exemple de chronogrammes d'une bascule JK



⊕ Bascule D synchronisé sur niveau ou D latch (verrou)

La bascule D recopie l'entrée D (Data) sur la sortie Q et cette sortie Q restera mémorisée tant que le signal d'horloge restera active.



| H | D | Q_n | Q_{n+1} | |
|-----|-----|-------|-----------|-----------------------|
| 0 | × | 0 | Q_n | Etat de Q verrouillée |
| | | 1 | | |
| 1 | 1 | 0 | 1 | Q recopie D |
| | | 1 | 1 | |
| | 0 | 0 | 0 | |
| | | 1 | 0 | |

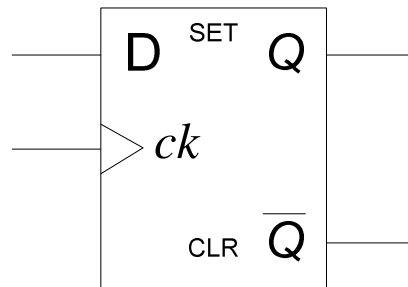
L'équation logique associée à la bascule D est :

$$Q_{n+1} = DH + \bar{H}Q_n$$

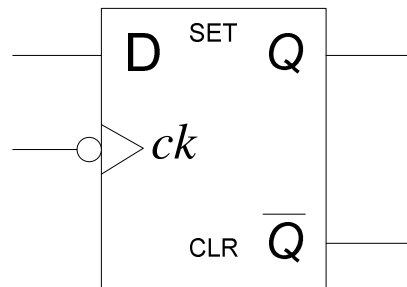
⊗ Bascule D synchronisé sur front ou «D flip-flop »

Bascule D synchronisée sur front montant ou descendant

Symbole logique :





Bascule D synchronisée
sur front montant



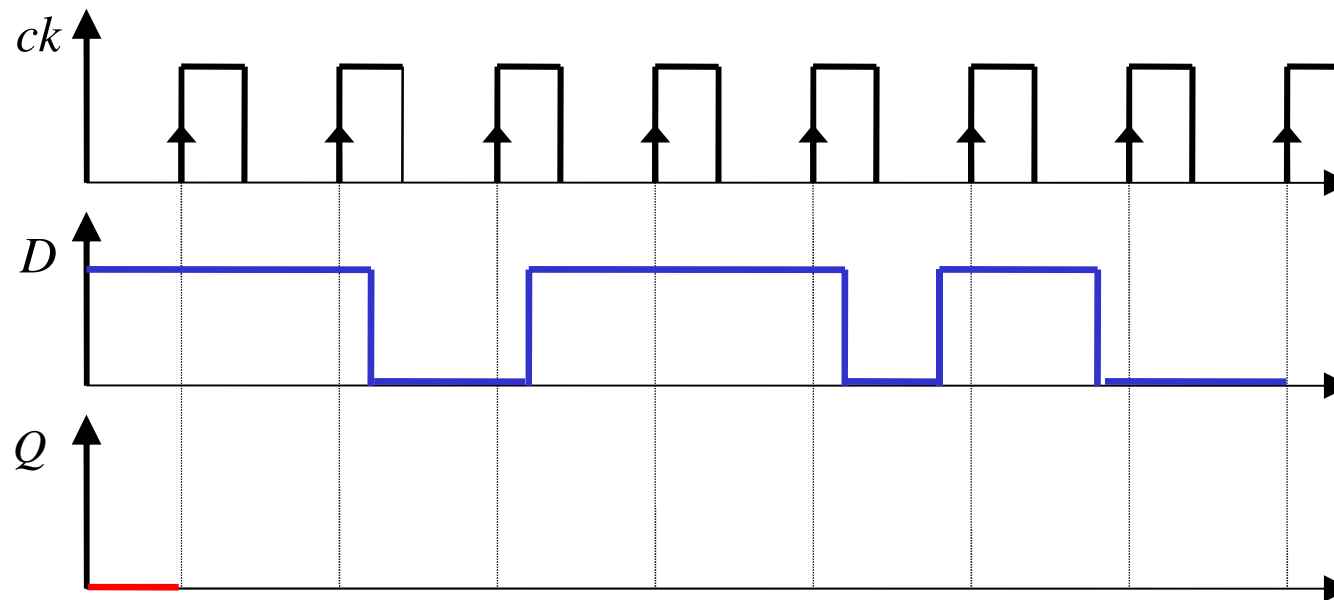
Bascule D synchronisée
sur front descendant

Modes de fonctionnement

| ck | D | Q_n | Q_{n+1} | |
|--|----------|-------|-----------|-----------------------|
| 0 | \times | 0 | Q_n | Etat de Q verrouillée |
| | | 1 | | |
| 1 | \times | 0 | Q_n | Etat de Q verrouillée |
| | | 1 | | |
|  ou  | 1 | 0 | 1 | Q recopie D |
| | | 1 | 1 | |
| | 0 | 0 | 0 | |
| | | 1 | 0 | |

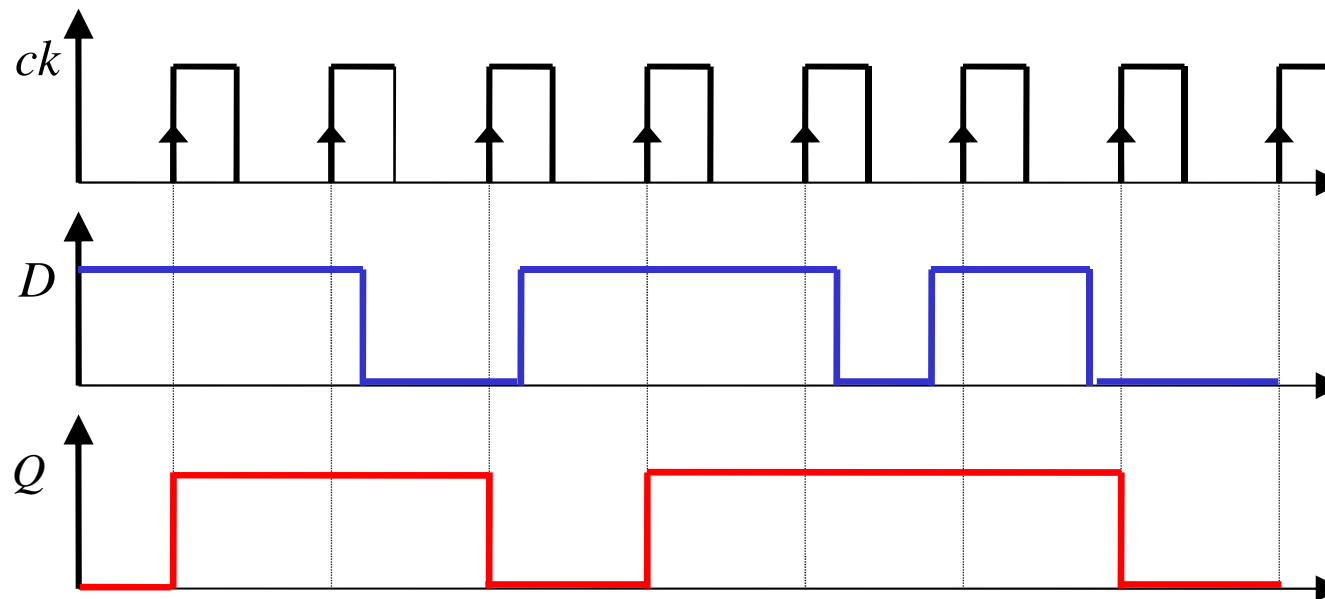
Exemple de chronogrammes d'une bascule D synchronisé sur front montant

À $t = 0$, Q est supposé être à 0, compléter le chronogramme



Exemple de chronogrammes d'une bascule D synchronisé sur front montant

A $t = 0$, Q est supposé être à 0, compléter le chronogramme

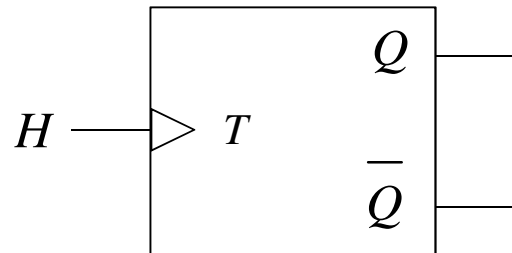


Sur chaque front montant de ck , la bascule mémorise l'état qu'elle voit en D à cet instant,


🌐 **Bascule T (Toggle) ou diviseur de fréquence par 2**

Elle possède une seule entrée T (Toggle) donnant l'ordre à la sortie de changer d'état sur chaque front montant du signal d'horloge appliqué en T .

Symbole logique :



Tables des états associées

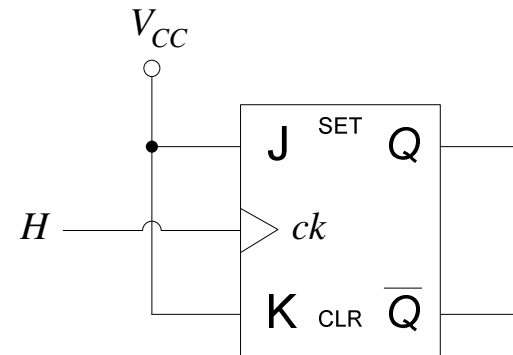
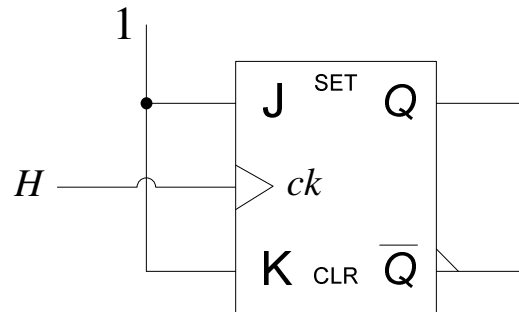
| H | Q_{n+1} |
|---|------------------|
| 0 | Q_n |
| 1 | Q_n |
|  | $\overline{Q_n}$ |

La bascule T n'existe pas en tant que composant discret. Elle est pourtant importante car elle est à la base de certains compteurs. Elle peut être obtenue à partir de bascules D ou JK

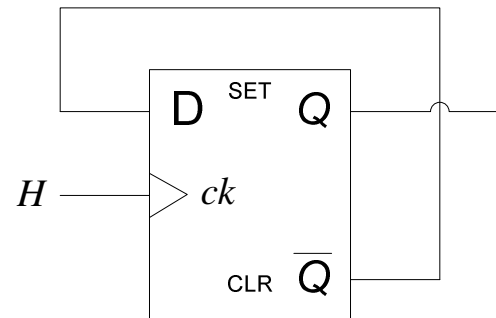
Exemple de réalisations d'une Bascule T (Toggle)

Il est possible de réaliser une bascule T à partir de bascules D ou JK

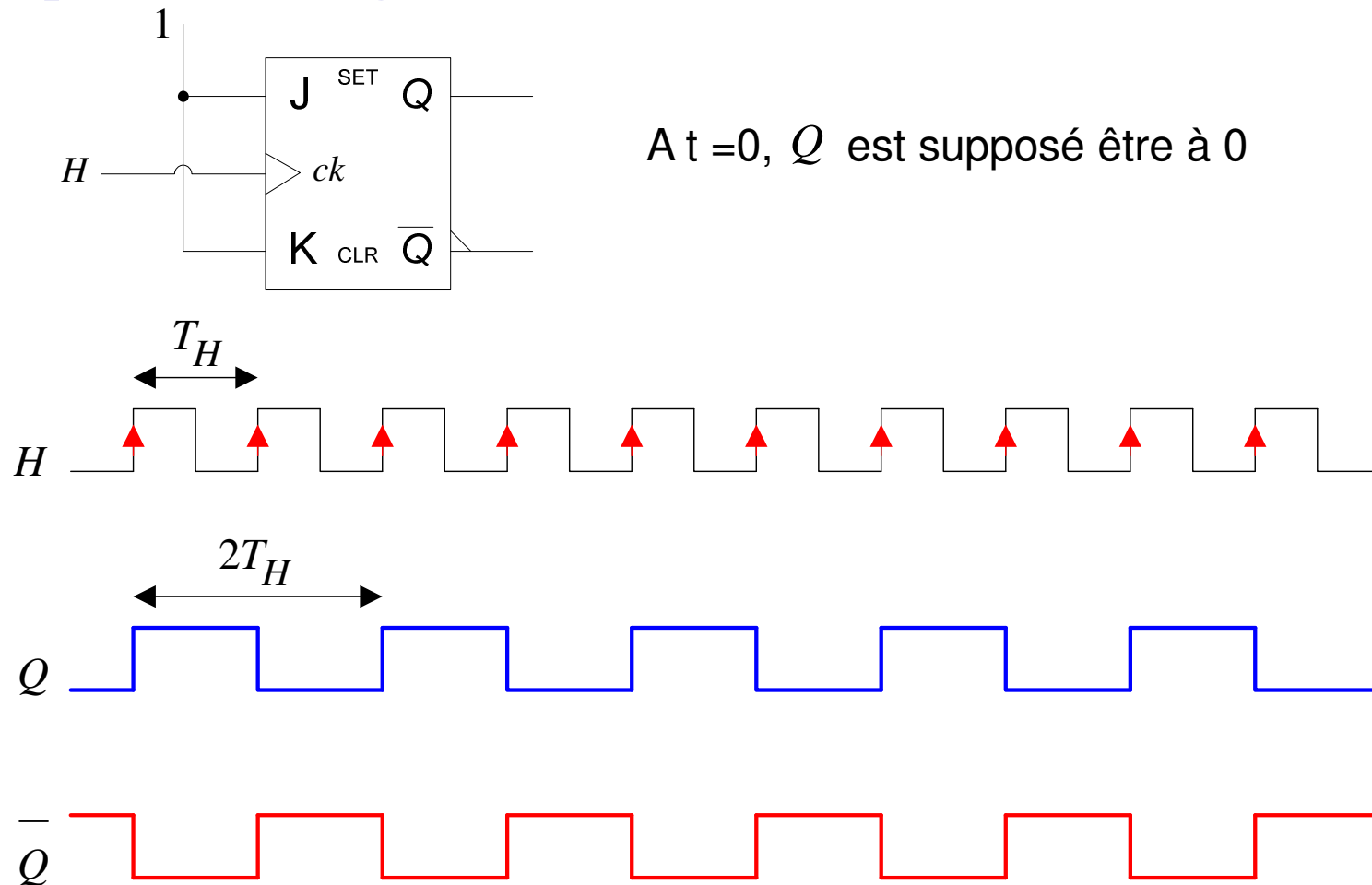
Synthèse avec une bascule JK



Synthèse avec une bascule D flip-flop



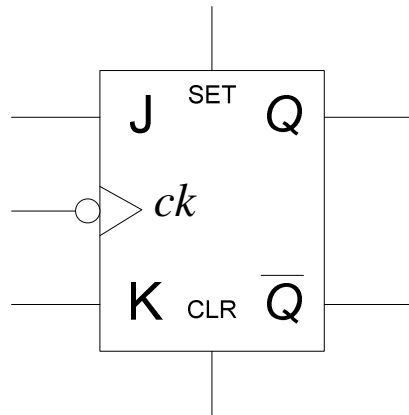
Exemple de chronogrammes d'une bascule T



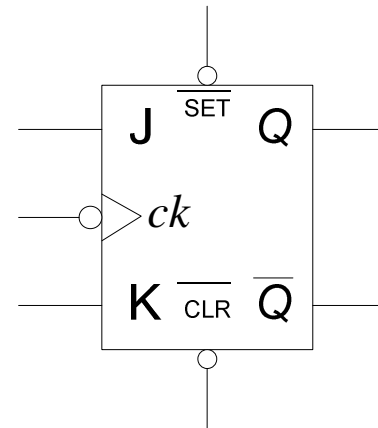
🌐 **Fonctionnement forcé des bascules : initialisation**

En général, les bascules synchrones possèdent deux entrées supplémentaires Set (ou Preset) et CLR (Reset).

Ces commandes asynchrone sont prioritaires car elles imposent au circuit un certain état quelque soit l'horloge : ces entrées sont dites de forçage ou de prépositionnement.

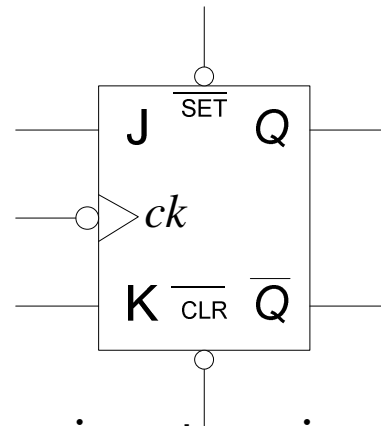


Les commandes asynchrones agissent au niveau haut ; il faut les porter au 1 logique pour qu'elles agissent



Les commandes asynchrones agissent au niveau bas; il faut les porter au 0 logique pour qu'elles agissent

⊗ Fonctionnement forcé des bascules : initialisation



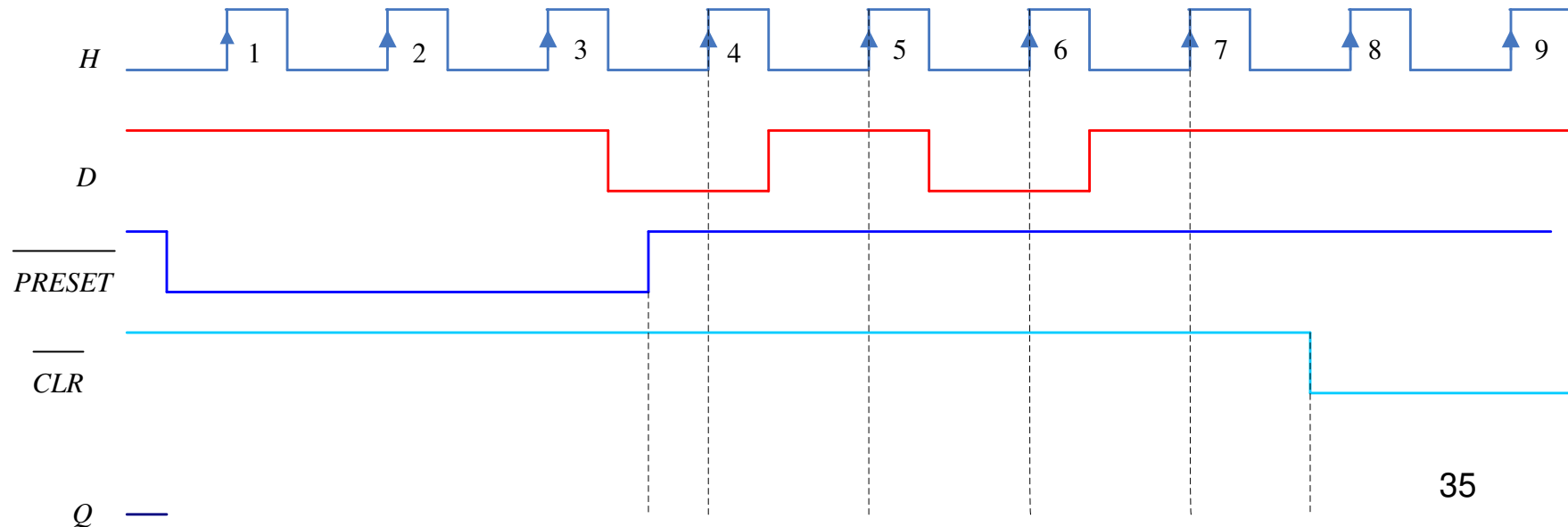
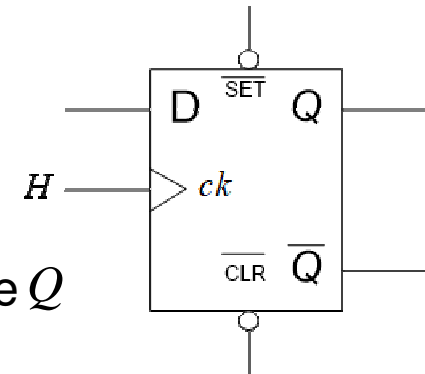
Les commandes asynchrones agissent au niveau bas; il faut les porter au 0 logique pour qu'elles agissent

Tables de vérité associée

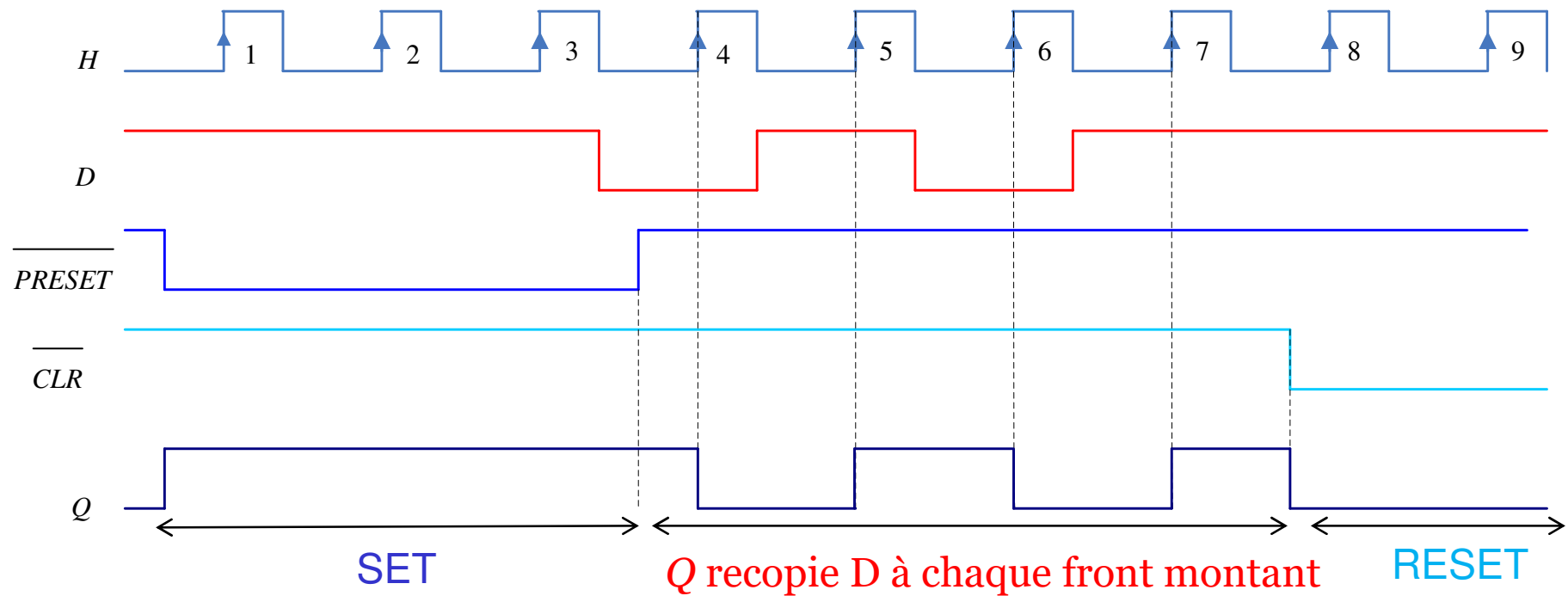
| SET | CLR | ck | J | K | Q_{n+1} | Remarques |
|-----|-----|----|---|---|-----------|-------------------|
| 0 | 1 | × | × | × | 1 | Mise à 1 |
| 1 | 0 | × | × | × | 0 | Mise à 0 |
| 0 | 0 | × | × | × | * | État imprévisible |

Exemple de chronogrammes d'une bascule D avec des entrées de forçage

At $t=0$, Q est supposé être à 0, compléter le chronogramme de Q

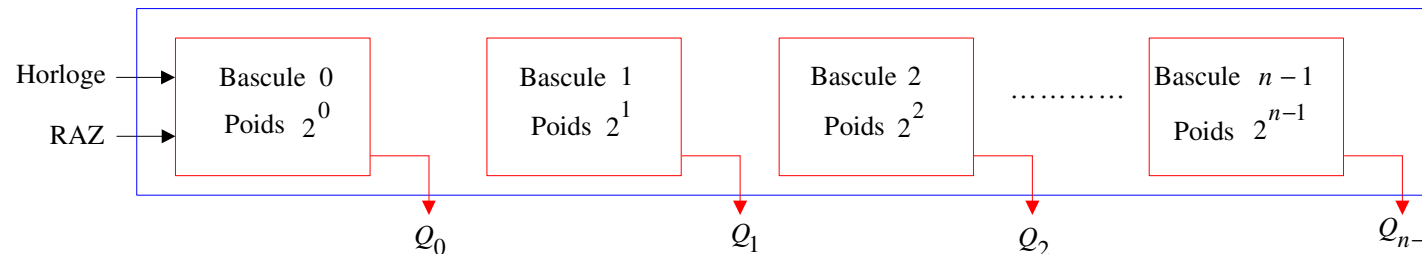


Exemple de chronogrammes d'une bascule D avec des entrées de forçage



Les compteurs

Un compteur numérique est un circuit séquentiel constitué de n bascules (D ou JK). Il a pour but de générer au rythme d'une horloge une séquence de nombres. Ce nombre est lu sous forme binaire sur les n sorties des bascules.



Après chaque impulsion d'horloge, le nombre binaire est augmenté de "1".

La valeur décimale du mot binaire $[Q_{n-1}, \dots, Q_2, Q_1, Q_0]$ de sortie égale à $Q_{n-1} 2^{n-1} + \dots + Q_2 2^2 + Q_1 2^1 + Q_0 2^0$

Le compteur possède en général une entrée RAZ (remise à zéro)

Le nombre maximum d'états possibles d'un compteur est appelé modulo

Le modulo maximal d'un compteur à n bits (n bascules) est 2^n (le nombre évolue entre 0 et $2^n - 1$)

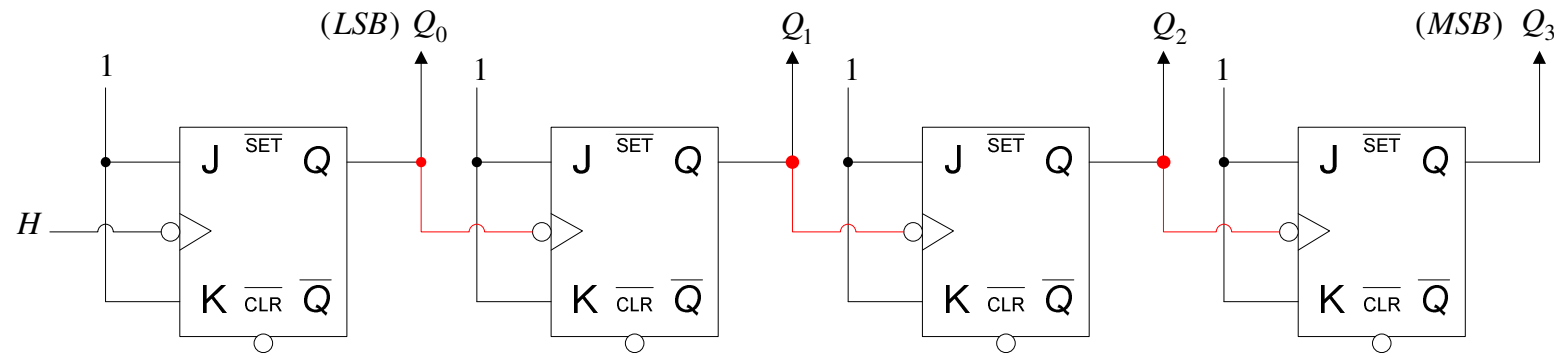


🌐 Différentes types de compteurs

- Les compteurs numériques peuvent être classifiés comme suit:
 - Selon leur fonction : compteurs synchrones, asynchrones.
 - Les compteurs asynchrones : chaque sortie de bascule est reliée à l'entrée d'horloge de la bascule suivante;
 - Les compteurs synchrones ; l'entrée d'horloge est commune à toutes les bascules
 - Selon leur direction de comptage : compteurs (incrémentaux), décompteurs, bidirectionnels
 - Selon leur codage du résultat : compteurs binaires, compteurs BCD.
 - Les compteurs se présentent généralement sous la forme de circuits intégrés

Les compteurs asynchrone modulo $N = 2^n$

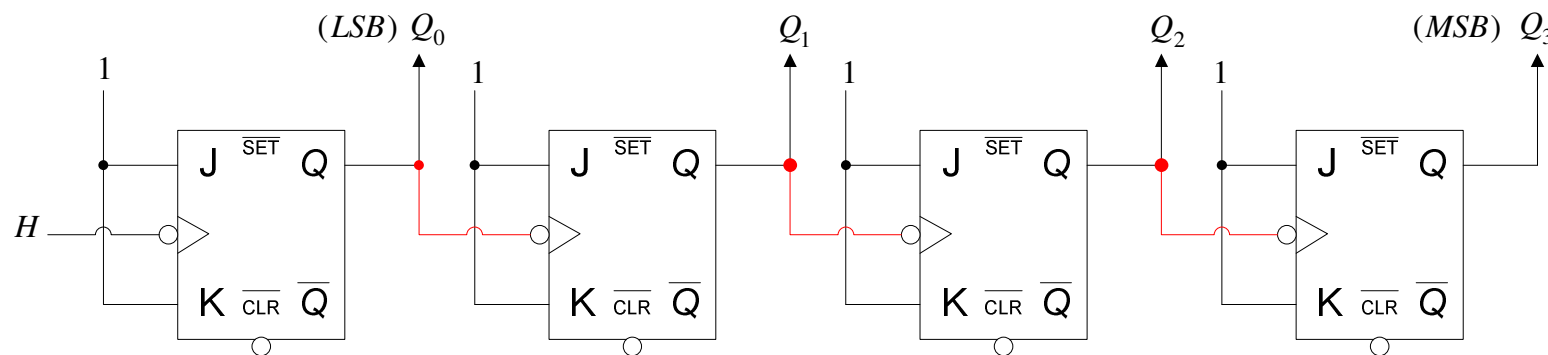
Exemple 1 : Compteur asynchrone modulo 16 en bascule JK



- Il est constitué de 4 bascules JK branchées en cascade. (car $2^4 = 16$)
- La sortie de chaque bascule commande le signal d'horloge de la suivante.
- Les bascule ne changent pas d'état simultanément mais les unes après les autres d'où le nom de compteur asynchrone ou à propagation.
- A chaque front descendant d'horloge, les sorties $[Q_3, Q_2, Q_1, Q_0]$ s'incrémentent cycliquement de 0000 à 1111.

Les compteurs asynchrone modulo $N = 2^n$

Fonctionnement du compteur asynchrone modulo 16 :



$J = K = 1$; toutes les bascules commutent sur des fronts descendants ;

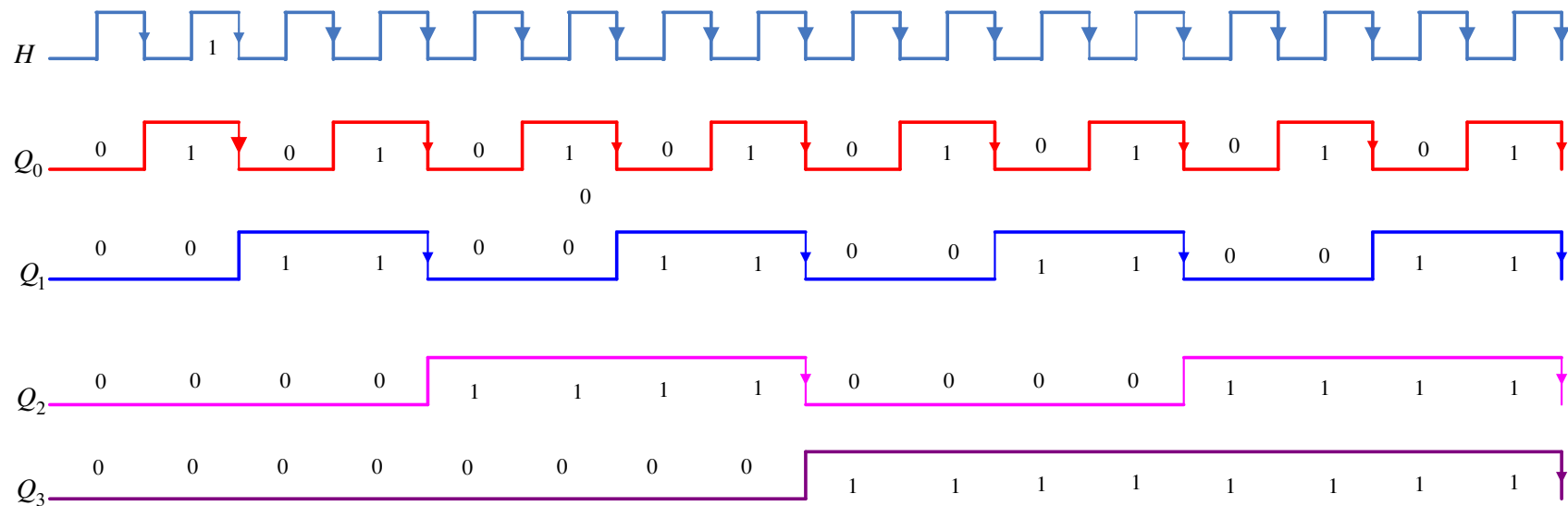
La bascule Q_0 commute à chaque front descendant du signal d'horloge ;

La sortie de la bascule Q_0 sert d'horloge pour la bascule $Q_1 \Rightarrow Q_1$ commute à chaque front descendant de Q_0 ;

de la même manière, Q_2 commute lorsque Q_1 passe de 1 à 0, et Q_3 commute lorsque Q_2 passe de 1 à 0.

Les compteurs asynchrone modulo $N = 2^n$

Chronogramme de fonctionnement du compteur modulo 16 :



Chaque bascule divise par deux la fréquence d'horloge qui alimente son entrée H :

$$f_{Q_0} = \frac{f_H}{2}; \quad f_{Q_1} = \frac{f_H}{4}; \quad f_{Q_2} = \frac{f_H}{8}; \quad f_{Q_3} = \frac{f_H}{16}; \quad f_{Q_i} = \frac{f_H}{2^{i+1}}; \quad 41$$



Circuits Séquentiels



FPK

Filière SMI S3

| Affichage | | | | Nombre d'impulsions comptées | |
|-----------|-------|-------|-------|---|--|
| Q_3 | Q_2 | Q_1 | Q_0 | Valeur décimale du mot binaire $[Q_3, Q_2, Q_1, Q_0]$ | |
| 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 1 | 1 | |
| 0 | 0 | 1 | 0 | 2 | |
| 0 | 0 | 1 | 1 | 3 | |
| 0 | 1 | 0 | 0 | 4 | |
| 0 | 1 | 0 | 1 | 5 | |
| 0 | 1 | 1 | 0 | 6 | |
| 0 | 1 | 1 | 1 | 7 | |
| 1 | 0 | 0 | 0 | 8 | |
| 1 | 0 | 0 | 1 | 9 | |
| 1 | 0 | 1 | 0 | 10 | |
| 1 | 0 | 1 | 1 | 11 | |
| 1 | 1 | 0 | 0 | 12 | |
| 1 | 1 | 0 | 1 | 13 | |
| 1 | 1 | 1 | 0 | 14 | |
| 1 | 1 | 1 | 1 | 15 | |

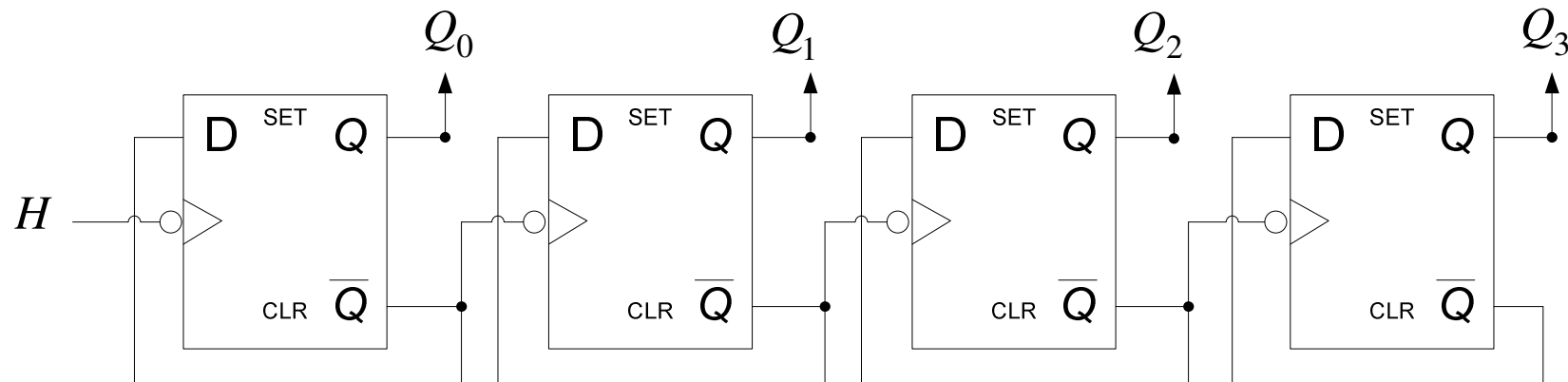
42

Les compteurs asynchrone modulo $N = 2^n$

Exemple 2 : Compteur asynchrone modulo 16 en bascule D

Câblage du compteur

Pour réaliser ce type de compteur en bascule D, il suffit de relier la sortie complémentée de la première bascule sur l'entrée d'horloge de la bascule suivante et l'entrée D à la sortie complémentée :

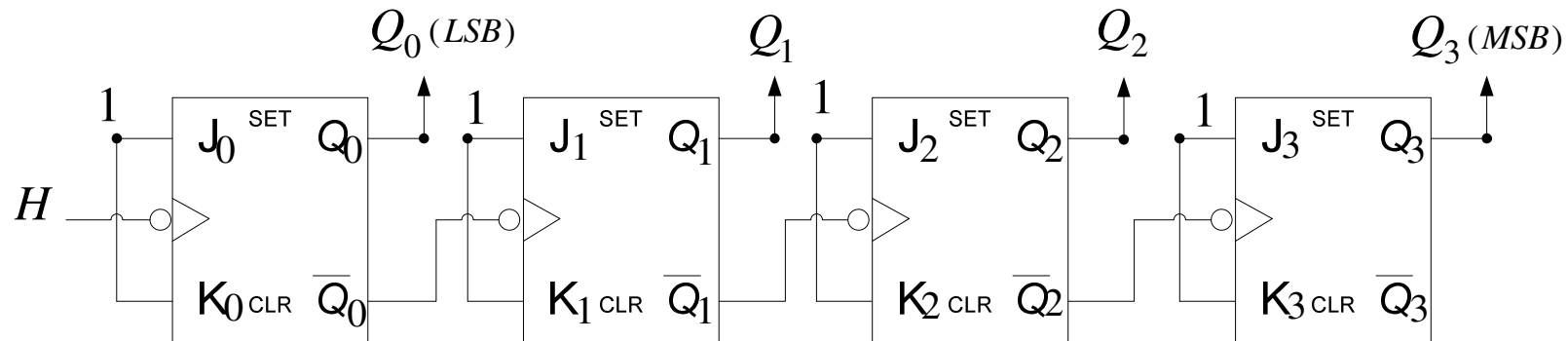


Réaliser ce montage sous ISIS à base du circuit 4013 puis tester le montage.

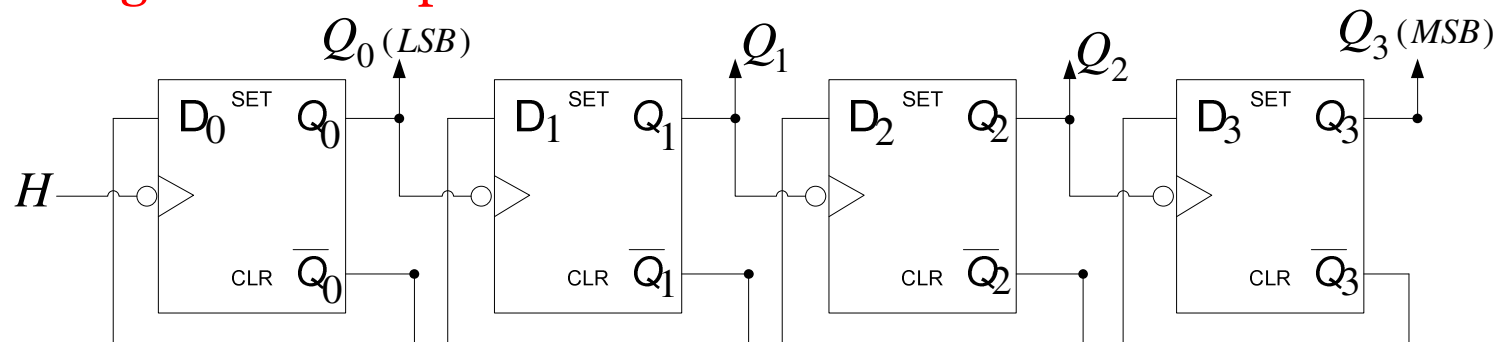
⊕ Décompteur asynchrone modulo-16

Un décompteur décrémente l'état logique en partant d'un nombre maximal jusqu'à l'obtention de la valeur 0.

Câblage du décompteur en bascule JK



Câblage du décompteur en bascule D





🌐 Compteur asynchrone modulo $N < 2^n$

La conception d'un compteur Modulo- N asynchrone se résume en trois points :

- 1) Trouver le nombre de bascules nécessaires à l'aide de la relation $2^{n-1} \leq N \leq 2^n$ dans laquelle n correspond au nombre de bascules et N , au nombre d'état.
- 2) Relier les bascules à la manière d'un compteur binaire asynchrone
- 3) Réaliser la fonction logique à l'aide de portes NAND pour remettre les bascules à zéro au compte de N .

Ainsi, le circuit effectue le compte de 0 jusqu'à $N-1$ pour produire un compteur Modulo- N

Exemple : Compteur asynchrone modulo-6

Il compte jusqu'à 5 (101 en binaire). La succession de nombre du compteur est donc :

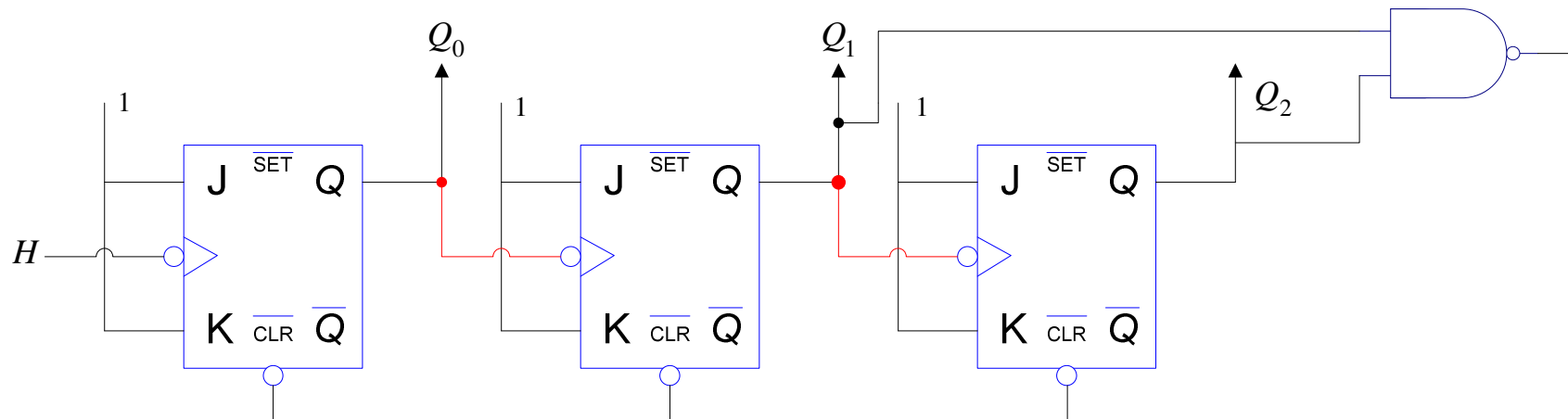
| Affichage | | | Nombre d'impulsions comptées |
|-----------|-------|-------|---|
| Q_2 | Q_1 | Q_0 | Valeur décimale du mot binaire |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 4 |
| 1 | 0 | 1 | 5 |
| 1 | 1 | 0 | État intermédiaire qui sert à remettre le compteur à zéro |

Pour que le compteur prenne uniquement les états de $(000)_2$ à $(101)_2$, il faut forcer la remise à zéro de toutes les bascules dès l'apparition du compte $(110)_2$.

La fonction logique qui permet de basculer le compte à zéro correspond aux condition $Q_1 = 1$ et $Q_2 = 1$. On réalise cette fonction à l'aide d'une porte NON-ET à deux entrée.

Exemple : Compteur asynchrone modulo-6 (suite)

Le câblage d'un tel compteur en bascule JK est donné par le schéma suivant :



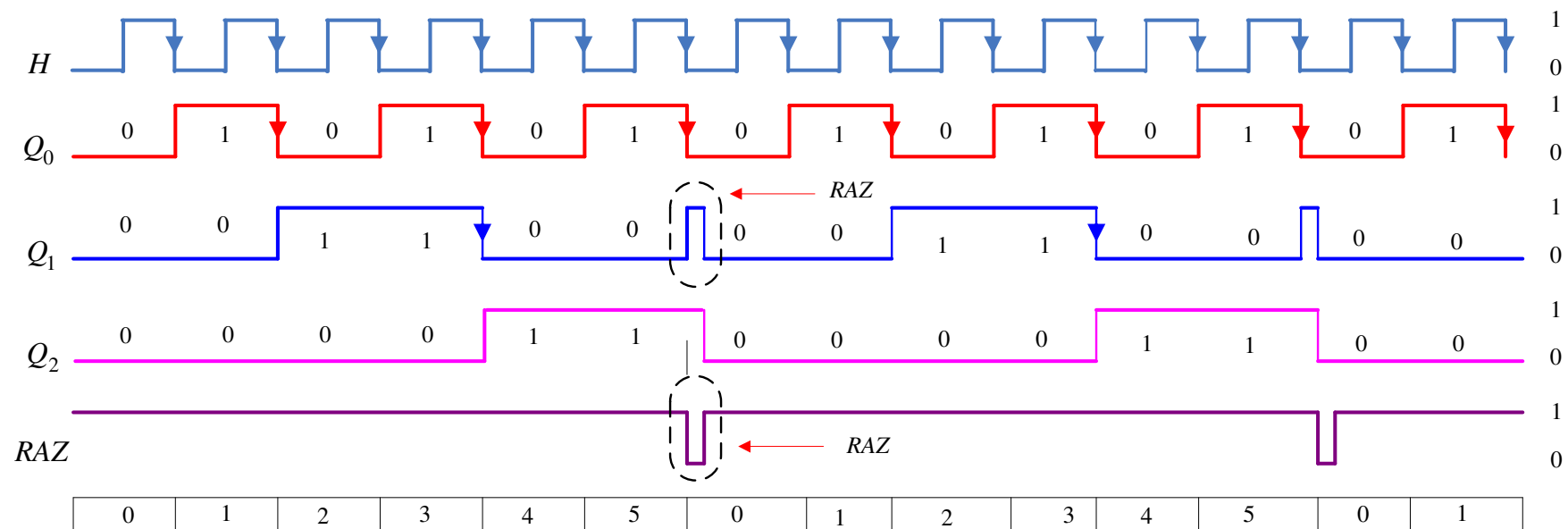
La bascule Q_0 commute à chaque front descendant du signal d'horloge ;

La bascule Q_1 commute à chaque front descendant sur Q_0 ;

La bascule Q_2 commute à chaque front descendant sur Q_1 ;

$(110)_2$ est la combinaison de remise à zéro

Chronogrammes de fonctionnement du compteur modulo 6



L'état intermédiaire $(Q_2Q_1Q_0) = (110)_2$ apparaît pendant une dure très courte (quelques nanosecondes), car l'impulsion produite à la sortie de la porte logique NAND force le compte à $(000)_2$. C'est un état indésirable.

Compteurs à décade (DCB) (ou Modulo-10)

Il compte jusqu'à 9 (1001 en binaire)

Il est composé de 4 bascules, il réalise une remise à zéro à toutes les bascules à la dixième impulsion d'horloge

$(1010)_2$ est la combinaison de remise à zéro

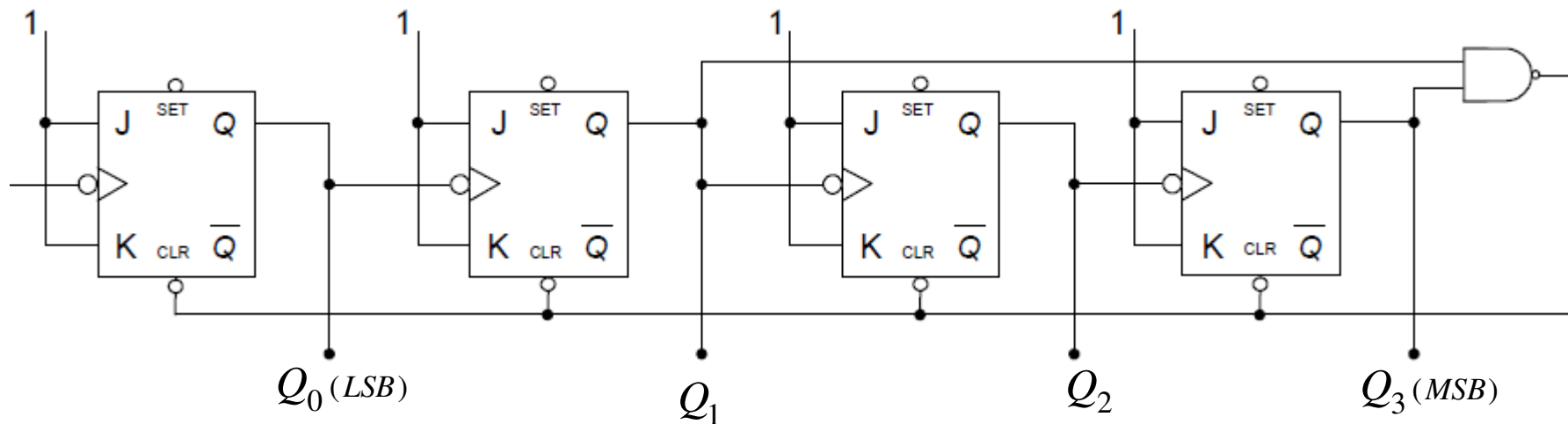
remise à zéro de toutes les bascules

| | | $Q_1 \cdot Q_0$ | | | |
|-----------------|---------------------------------------|---------------------------------------|----------------------------|----------------------------|-----------------|
| | | $\overline{Q_1} \cdot \overline{Q_0}$ | $\overline{Q_1} \cdot Q_0$ | $Q_1 \cdot \overline{Q_0}$ | $Q_1 \cdot Q_0$ |
| $Q_3 \cdot Q_2$ | $\overline{Q_3} \cdot \overline{Q_2}$ | 0 | 0 | 0 | 0 |
| | $\overline{Q_3} \cdot Q_2$ | 0 | 0 | 0 | 0 |
| | $Q_3 \cdot \overline{Q_2}$ | × | × | × | × |
| | $Q_3 \cdot Q_2$ | 0 | 0 | × | 1 |

$Q_1 \cdot Q_3$

combinaison de remise à zéro

Compteurs à décade (DCB) (ou Modulo-10)



Réaliser ce montage sous ISIS à base du circuit intégrés de la famille : 74SL76 , 7400 et 7405 puis tester le montage.



⊗ Inconvénients des compteurs asynchrones

- Les temps de propagation se cumulent, réduisant la fréquence maximale de comptage
- Présence d'état transitoires indésirables sur les sorties
- Inutilisable pour des comptages longs à cause de temps de propagation du compteur qui vaut n fois le temps de réponse d'une bascule ($n.t_p$).

Le temps de réponse d'une bascule ($t_p = 25ns$)

- Pour pallier ce problème, on réalise des compteurs synchrones, pour lesquels le signal d'horloge est commun à toutes les bascules.

⊗ Compteurs synchrone (parallèle)

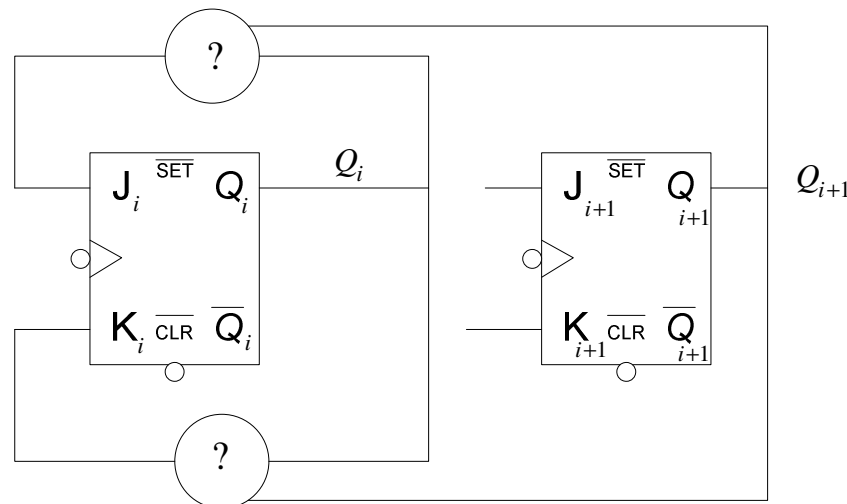
- Dans un compteur synchrone toutes les bascules sont commandées simultanément par le même signal d'horloge. Ceci évite le problème du retard de propagation

❶ Méthode de synthèse d'un compteurs synchrone

- 1) Déterminer le nombre de bascules nécessaires à l'aide de la relation $2^{n-1} \leq N \leq 2^n$ dans laquelle n correspond au nombre de bascules et N au nombre d'état.
- 2) Déterminer la table de transition du compteur (état suivant Q_{t+1} en fonction de l'état présent Q_t , supposé connu.
- 3) Déterminer l'expressions des entrée J_i et K_i des bascules en fonction des sorties (Q_0, \dots, Q_n)

$$J_i = f(Q_0, \dots, Q_n)$$

$$K_i = g(Q_0, \dots, Q_n)$$



⊗ Table d'excitation d'une bascule JK

Elle permet de trouver pour chaque transition possible de l'état Q_t à l'état Q_{t+1} les valeurs nécessaires des entrées J et K

Rappelons la table de vérité d'une bascule JK

| J | K | Q_{t+1} | |
|-----|-----|------------------|-------------|
| 0 | 0 | Q_t | Mémoire |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | $\overline{Q_t}$ | Basculement |

$$Q_{t+1} = J_n \overline{Q_n} + \overline{K_n} Q_n$$

| Sorties | | Entrées | |
|-------------|---------------|-------------|-----|
| Instant t | Instant $t+1$ | Instant t | |
| Q_t | Q_{t+1} | J | K |
| 0 | 0 | 0 | × |
| 0 | 1 | 1 | × |
| 1 | 0 | × | 1 |
| 1 | 1 | × | 0 |

× : état indifférent.

Exemple : compteur synchrone modulo-8 à bascules JK

Il y a trois bascules nécessaires, car on a $8 = 2^3$ états.

| | Etats présents | | | États suivants | | | Entrés à t | | | | | |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | Sorties à t | | | Sorties à t+1 | | | | | | | | |
| Etat | Q ₂ | Q ₁ | Q ₀ | Q ₂ | Q ₁ | Q ₀ | J ₂ | K ₂ | J ₁ | K ₁ | J ₀ | K ₀ |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | × | 0 | × | 1 | × |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | × | 1 | × | × | 1 |
| 2 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | × | × | 0 | 1 | × |
| 3 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | × | × | 1 | × | 1 |
| 4 | 1 | 0 | 0 | 1 | 0 | 1 | × | 0 | 0 | × | 1 | × |
| 5 | 1 | 0 | 1 | 1 | 1 | 0 | × | 0 | 1 | × | × | 1 |
| 6 | 1 | 1 | 0 | 1 | 1 | 1 | × | 0 | × | 0 | 1 | × |
| 7 | 1 | 1 | 1 | 0 | 0 | 0 | × | 1 | × | 1 | × | 1 |

Exemple : compteur synchrone modulo-8 à bascules JK

Ensuite, on écrit les tables de Karnaugh pour J_0, J_1, J_2, K_0, K_1 et K_2

J_0

| | | Q_1, Q_0 | | | |
|-------|---|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_2 | 0 | 1 | x | x | 1 |
| | 1 | 1 | x | x | 1 |

K_0

| | | Q_1, Q_0 | | | |
|-------|---|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_2 | 0 | x | 1 | 1 | x |
| | 1 | x | 1 | 1 | x |

J_1

| | | Q_1, Q_0 | | | |
|-------|---|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_2 | 0 | 0 | 1 | x | x |
| | 1 | 0 | 1 | x | x |

K_1

| | | Q_1, Q_0 | | | |
|-------|---|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_2 | 0 | x | x | 1 | 0 |
| | 1 | x | x | 1 | 0 |

J_2

| | | Q_1, Q_0 | | | |
|-------|---|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_2 | 0 | 0 | 0 | 1 | 0 |
| | 1 | x | x | x | x |

K_2

| | | Q_1, Q_0 | | | |
|-------|---|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_2 | 0 | x | x | x | x |
| | 1 | 0 | 0 | 1 | 0 |

On trouve alors :

$$J_0 = K_0 = 1$$



$$J_1 = K_1 = Q_0$$

$$J_2 = K_2 = Q_0 Q_1$$

🕒 Table d'excitation d'une bascule D

Elle permet de déterminer l'entrée D_i , à appliquer pour obtenir les sorties désirées du compteur, connaissant l'état des sorties avant l'impulsion d'horloge.

Rappelons la table de vérité d'une bascule D

| ck | D | Q_n | Q_{n+1} |
|---|-----|-------|-----------|
|  ou  | 0 | 0 | D |
| | | 1 | |
| | 1 | 1 | D |
| | | 0 | |

| Sorties | | Entrée |
|---------|-----------|--------|
| Q_t | Q_{t+1} | D |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$$D = Q_{t+1}$$

Exemple : compteur synchrone modulo-8 à bascules D

Il y a trois bascules nécessaires, car on a $8 = 2^3$ états.

| | Sorties à t | | | Sorties à $t+1$ | | | Entrées à t | | |
|------|---------------|-------|-------|-----------------|-------|-------|---------------|-------|-------|
| Etat | Q_2 | Q_1 | Q_0 | Q_2 | Q_1 | Q_0 | D_2 | D_1 | D_0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 3 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Exemple : compteur synchrone modulo-8 à bascules D

Ensuite, on écrit les tables de Karnaugh pour D_0 , D_1 et D_2

| | | Q_1, Q_0 | | | |
|-------|---|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_2 | 0 | 1 | 0 | 0 | 1 |
| | 1 | 1 | 0 | 0 | 1 |

| | | Q_1, Q_0 | | | |
|-------|---|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_2 | 0 | 0 | 1 | 0 | 1 |
| | 1 | 0 | 1 | 0 | 1 |

| | | Q_1, Q_0 | | | |
|-------|---|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_2 | 0 | 0 | 0 | 1 | 0 |
| | 1 | 1 | 1 | 0 | 1 |

$$D_0 = \overline{Q_0}$$

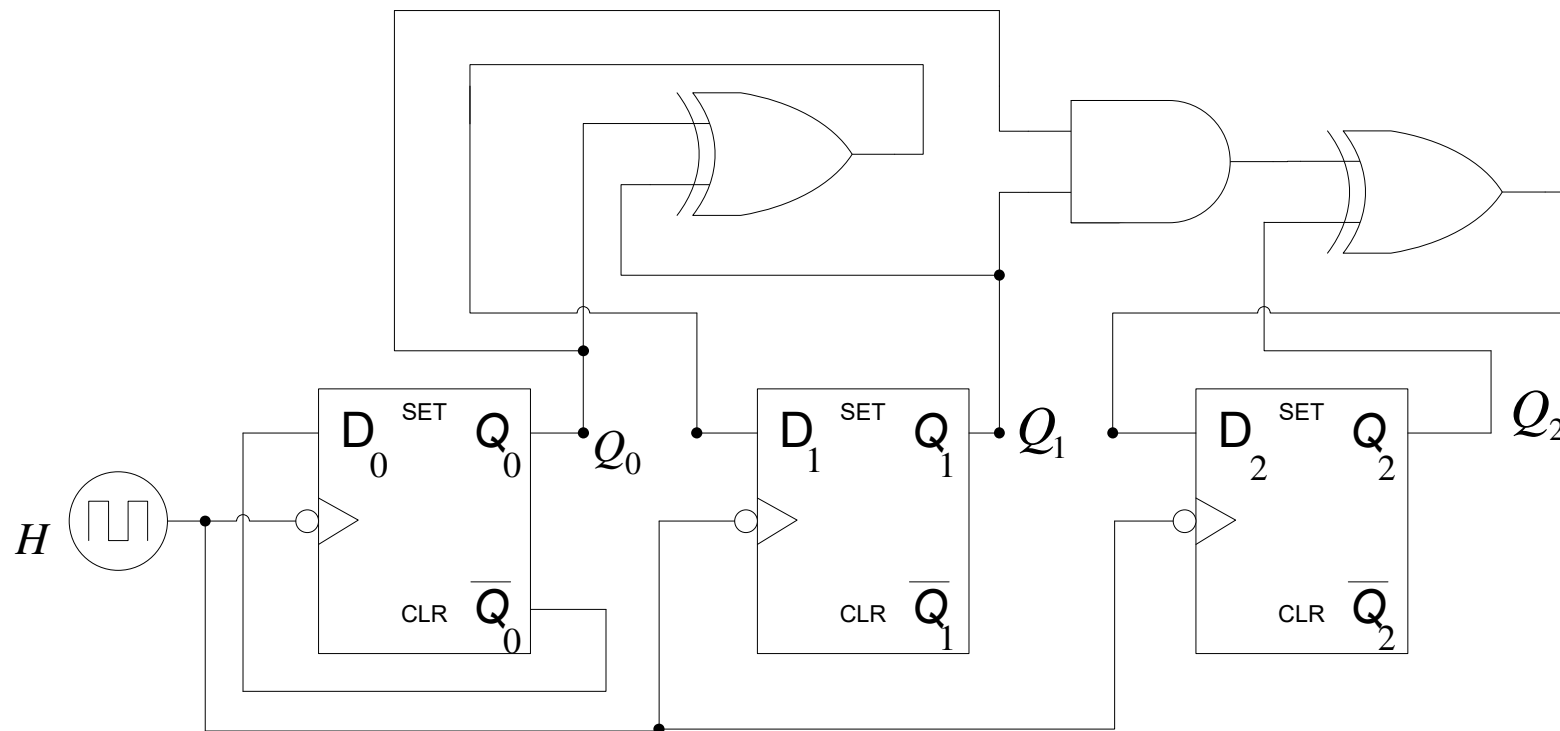
$$D_1 = Q_0 \overline{Q_1} + \overline{Q_0} Q_1 = Q_0 \oplus Q_1$$

$$D_2 = Q_0 Q_1 \overline{Q_2} + \overline{Q_1} Q_2 + \overline{Q_0} Q_2 = Q_0 Q_1 \oplus Q_2$$

Exemple : compteur synchrone modulo-8 à bascules D

Câblage du compteur

On a donc le schéma logique suivant pour le compteur modulo 8





● Les Registres

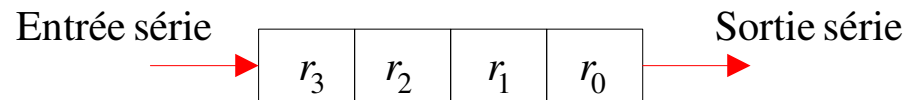
Un registre permet à la fois la mémorisation et le transfert d'un mot binaire de n bits. Il est donc constitué de n bascules qui peuvent être de type D ou JK, mémorisant chacune un bit. (toutes les bascules sont synchronisées par la même horloge) .

Différents types de registres

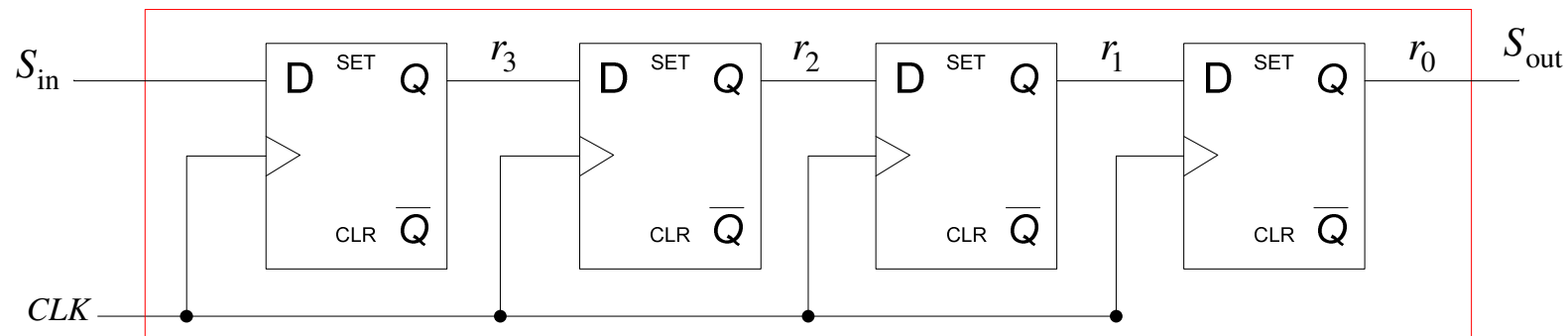
On distingue quatre types de registres selon la façon dont sont utilisées les entrées et les sorties : en parallèle ou en série

- Registre à entrées parallèles, sorties parallèles
- Registres à entrée série, sortie série
- Registres à entrée série, sorties parallèles
- Registres à entrées parallèles, sortie série

Registre à entrée série, sortie série



Considérons l'association série de bascules D



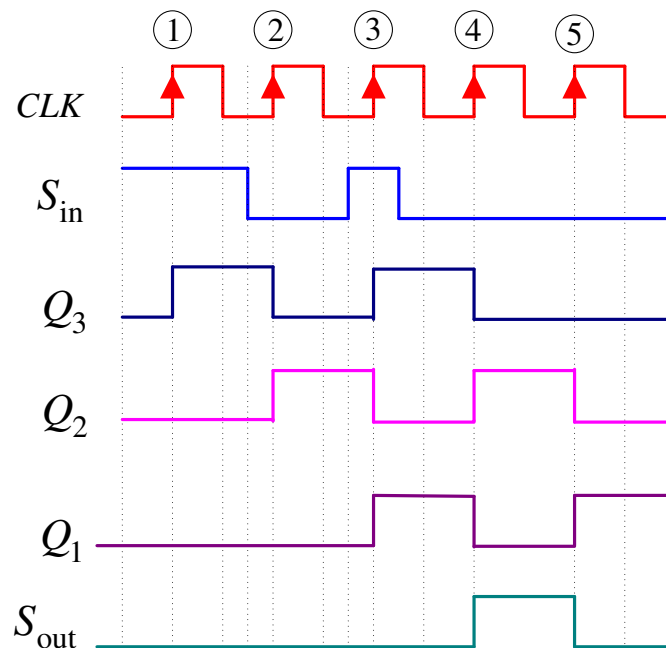
Ce registre possède une entrée série et une sortie Série. Les données binaires d'entrée sont introduites bit après bit. Elles sont également disponibles les unes après les autres au rythme de l'horloge en sortie. Ce type de registre est utilisé pour effectuer des décalages (registre à décalage à droite).

Ici 4 transitions d'horloge sont nécessaires pour charger le registre et 4 transitions sont aussi nécessaires pour lire les données par décalage.

Registre à entrée série, sortie série

Exemple de chronogrammes

Le registre est chargé initialement par la donnée (0000)



| CLK | Q_3 | Q_2 | Q_1 | Q_0 |
|-----|-------|-------|-------|-------|
| | 0 | 0 | 0 | 0 |
| ① | 1 | 0 | 0 | 0 |
| ② | 0 | 1 | 0 | 0 |
| ③ | 1 | 0 | 1 | 0 |
| ④ | 0 | 1 | 0 | 1 |
| ⑤ | 0 | 0 | 1 | 0 |

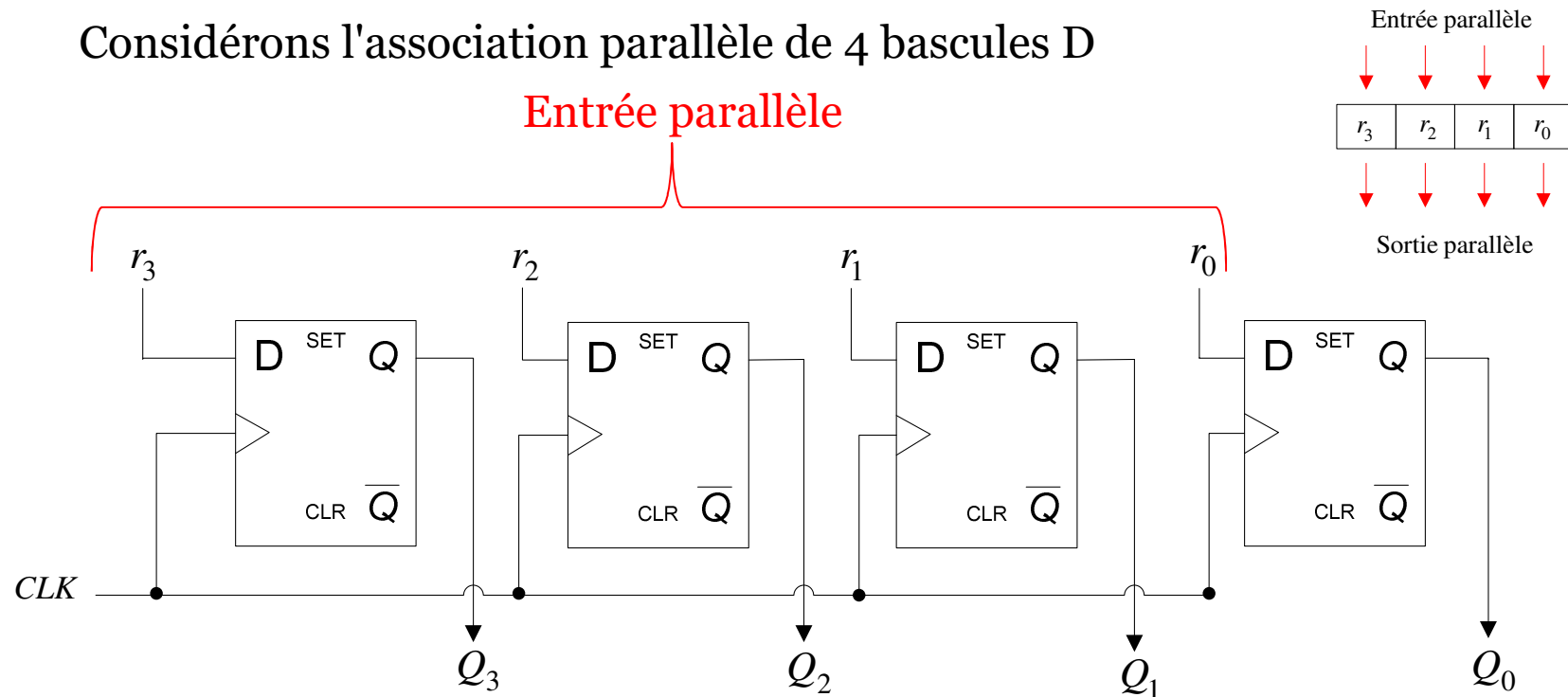
Etat initial

Etat du registre
juste après un
front montant
sur CLK

Tous les éléments binaires sont décalés d'un rang vers la droite.
Le registre est chargé par (0010) après 5 fronts montant de l'horloge

⊕ Registre à entrées parallèles et sorties parallèles

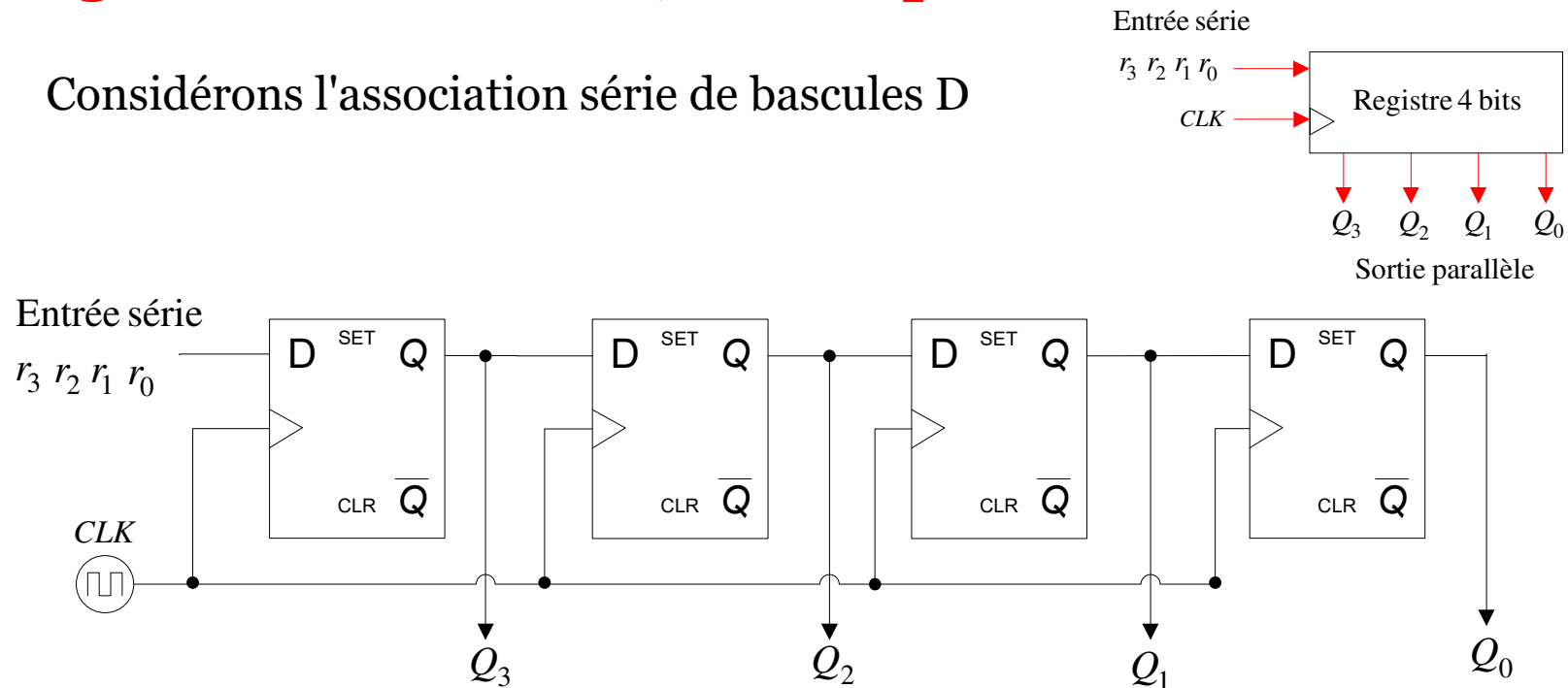
Considérons l'association parallèle de 4 bascules D



On obtient ici un registre de 4 bits, les 4 bascules sont chargées en parallèle par front montant du signal d'horloge. Les entrées véhiculent des informations de même nature : les lignes d'entrées constituent le BUS d'entrée. Les sorties Q constituent de même le bus de sortie

Registre à entrée série, sorties parallèles

Considérons l'association série de bascules D



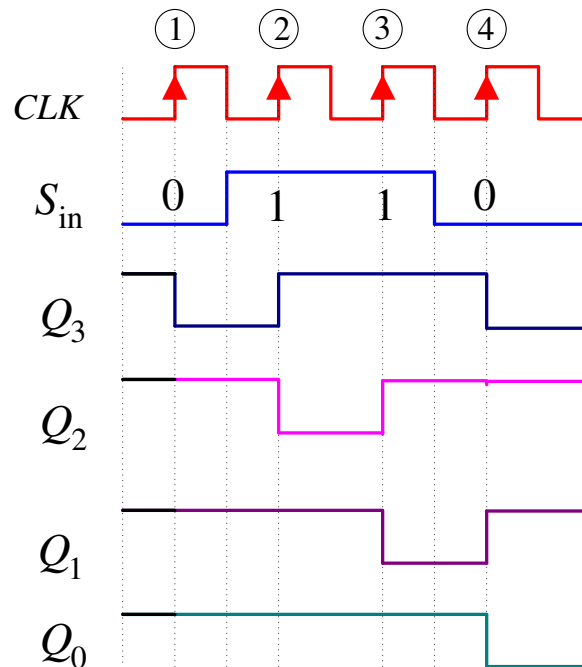
Ce registre possède une entrée série et plusieurs sorties (Q_3, Q_2, Q_1, Q_0). Les données binaires d'entrée sont introduites bit après bit. Les sorties sont toutes disponibles en même temps.

Le registre sera chargé par la donnée d'entrée après 4 coups d'horloge.

Registre à entrée série, sorties parallèles

Exemple de chronogrammes

Le registre est chargé initialement par la donnée (1111)



| CLK | Q ₃ | Q ₂ | Q ₁ | Q ₀ |
|-----|----------------|----------------|----------------|----------------|
| | 1 | 1 | 1 | 1 |
| ① ↑ | 0 | 1 | 1 | 1 |
| ② ↑ | 1 | 0 | 1 | 1 |
| ③ ↑ | 1 | 1 | 0 | 1 |
| ④ ↑ | 0 | 1 | 1 | 0 |

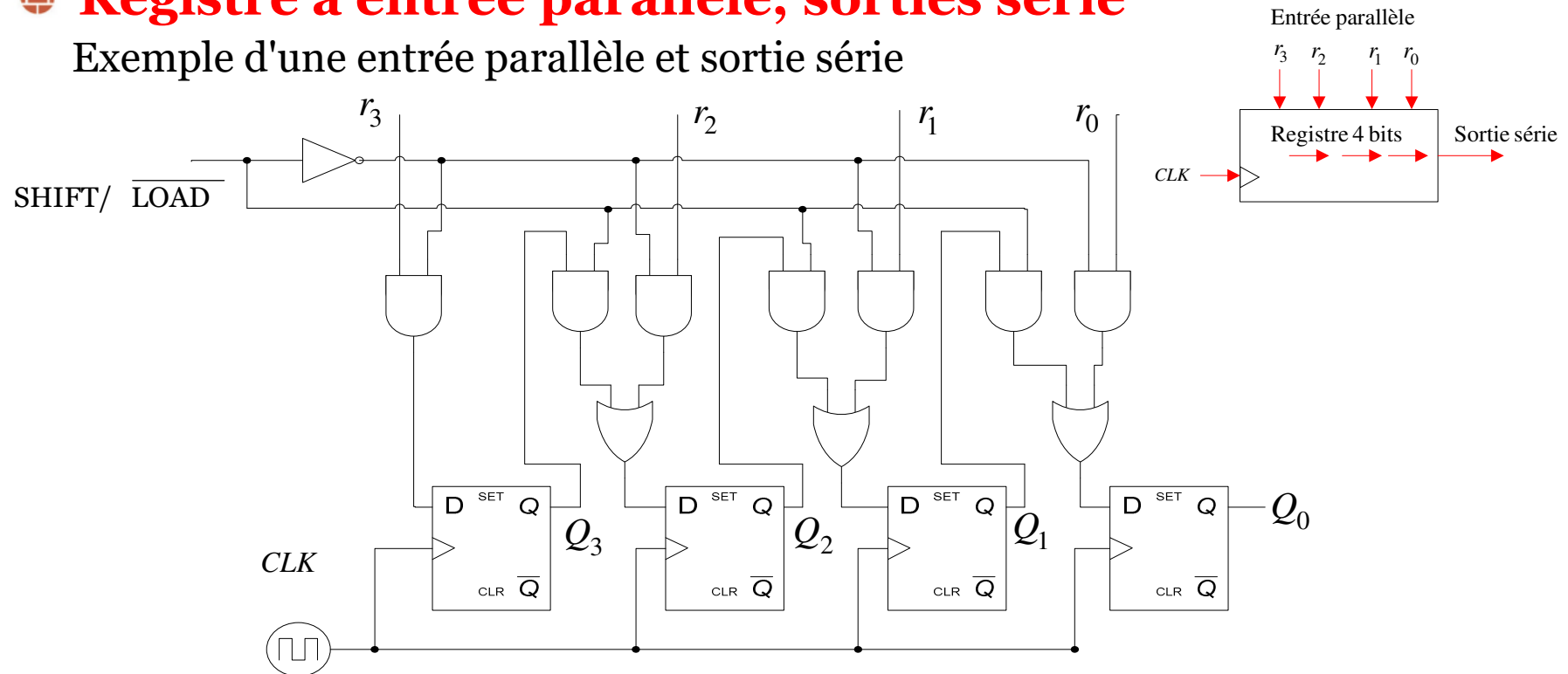
Etat initial

Etat du registre
juste après un
font montant
sur CLK

Le registre sera chargé par la donnée (0110) après 4 impulsions d'horloge.

Registre à entrée parallèle, sorties série

Exemple d'une entrée parallèle et sortie série



Si $\text{SHIFT/LOAD} = 0$, les entrées (r_3, r_2, r_1, r_0) sont introduites en même temps dans le registre. Les informations sont disponibles les unes après les autres au rythme de l'horloge.

Si $\text{SHIFT/LOAD} = 1$, l'entrée parallèle est inhibée et le décalage des sorties est validées.

Questions ?