CENTRE UNIVERSITAIRE DE MILA
INSTITUT DES SCIENCES ET DE LA TECHNOLOGIE
1ere année MI

Structure Machine 02

Chapitre 04

Les circuits séquentiels

Plan du Chapitre

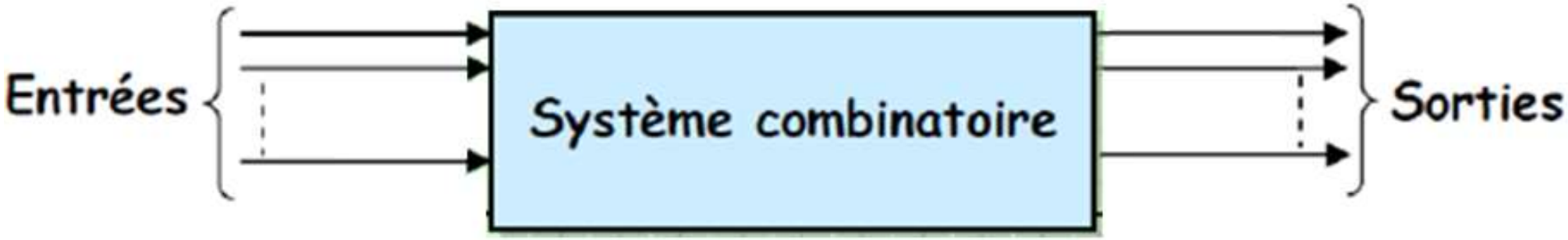
- Introduction
- Bascules
- Registres
- Compteurs

Introduction:

Plusieurs circuits utilisés dans la vie courante ont besoin de mémoire.
Ainsi, Un système séquentiel est un système logique dont l'état des variables de sortie dépend de l'état des variables d'entrée **+** de **l'état précédent des variables de sortie**.
Le système se souvient du passé en enregistrant les états précédents de ses sorties, faisant appel pour cela, à des **variables internes, ou mémoires.**

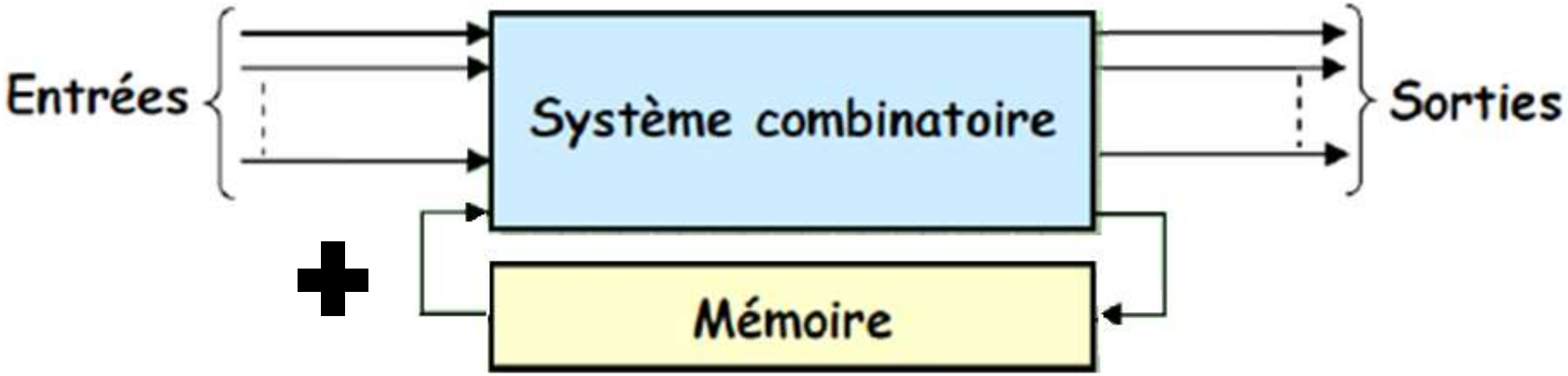
Introduction:

(Un circuit séquentiel est un circuit numérique (logique) dont l'état à **l'instant t+1** est une fonction des entrées en même **instant t+1** et de **l'état** précédente du système (**l'instant t**)).



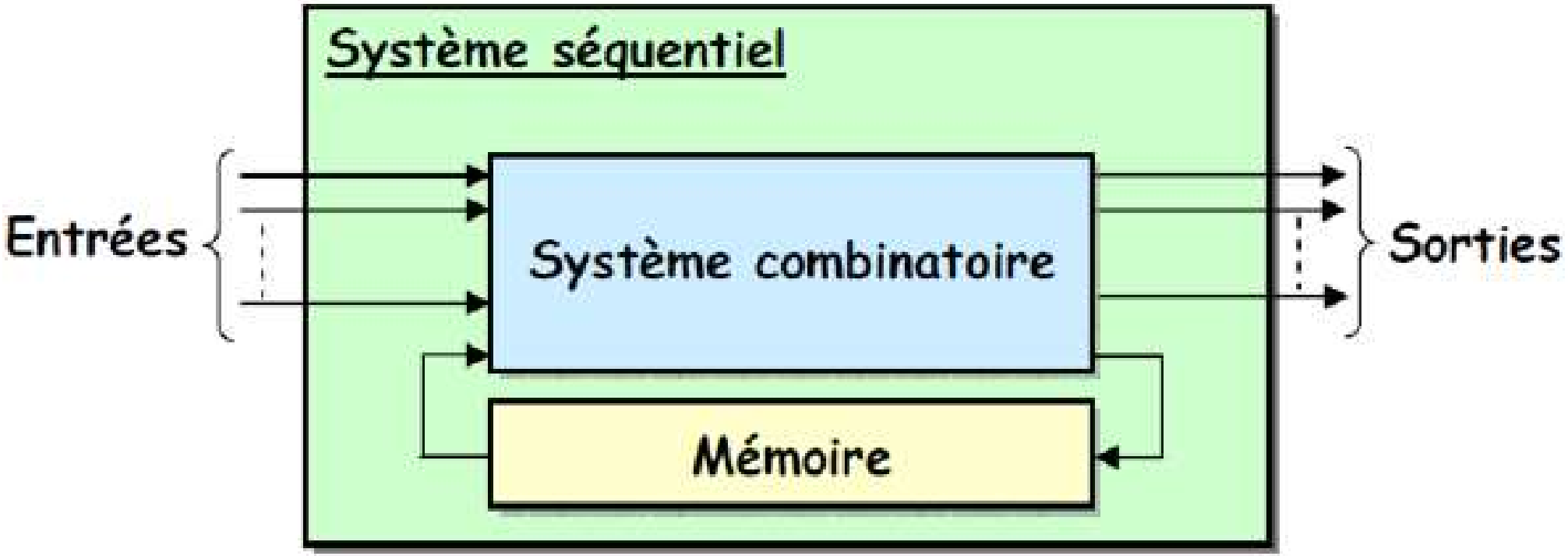
Introduction:

(Un circuit séquentiel est un circuit numérique (logique) dont l'état à **l'instant t+1** est une fonction des entrées en même **instant t+1** et de **l'état** précédente du système (**l'instant t**)).



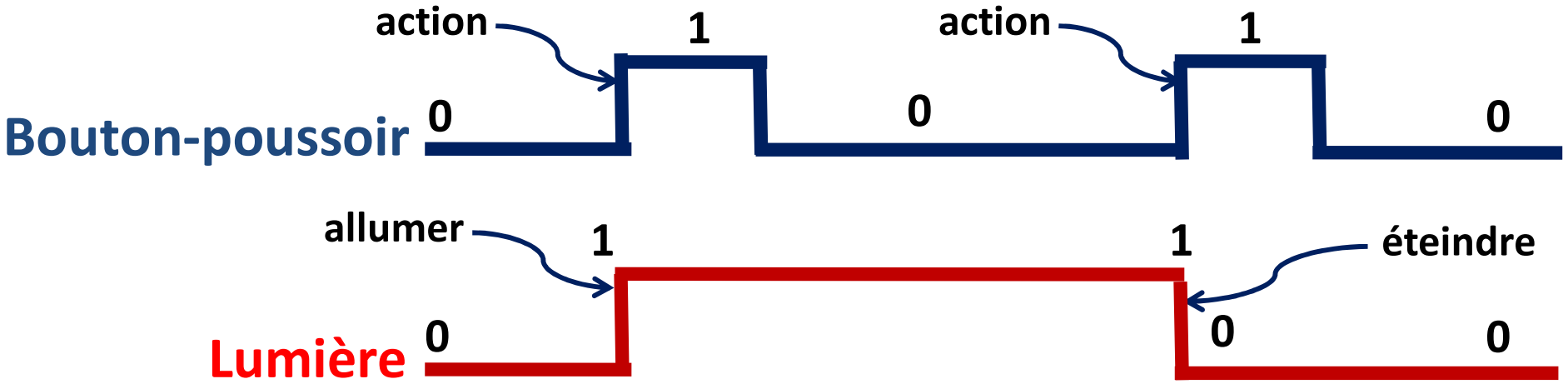
Introduction:

(Un circuit séquentiel est un circuit numérique (logique) dont l'état à **l'instant t+1** est une fonction des entrées en même **instant t+1** et de **l'état** précédente du système (**l'instant t**)).



Concept d'état

A titre d'illustration, prenons l'exemple 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 :



Concept d’état

La position (actionné ou non), n'est pas facilement identifiable par l'opérateur ; on leur adjoint souvent un voyant, indiquant l'état du circuit.
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.
Alors: 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.

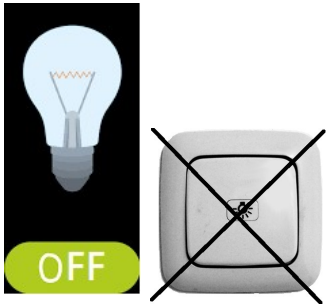
Concept d'état

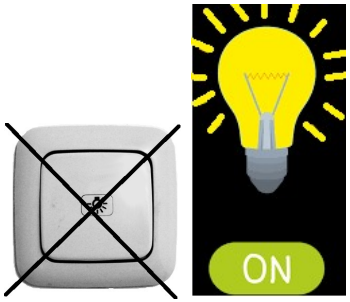
Graphe des phases

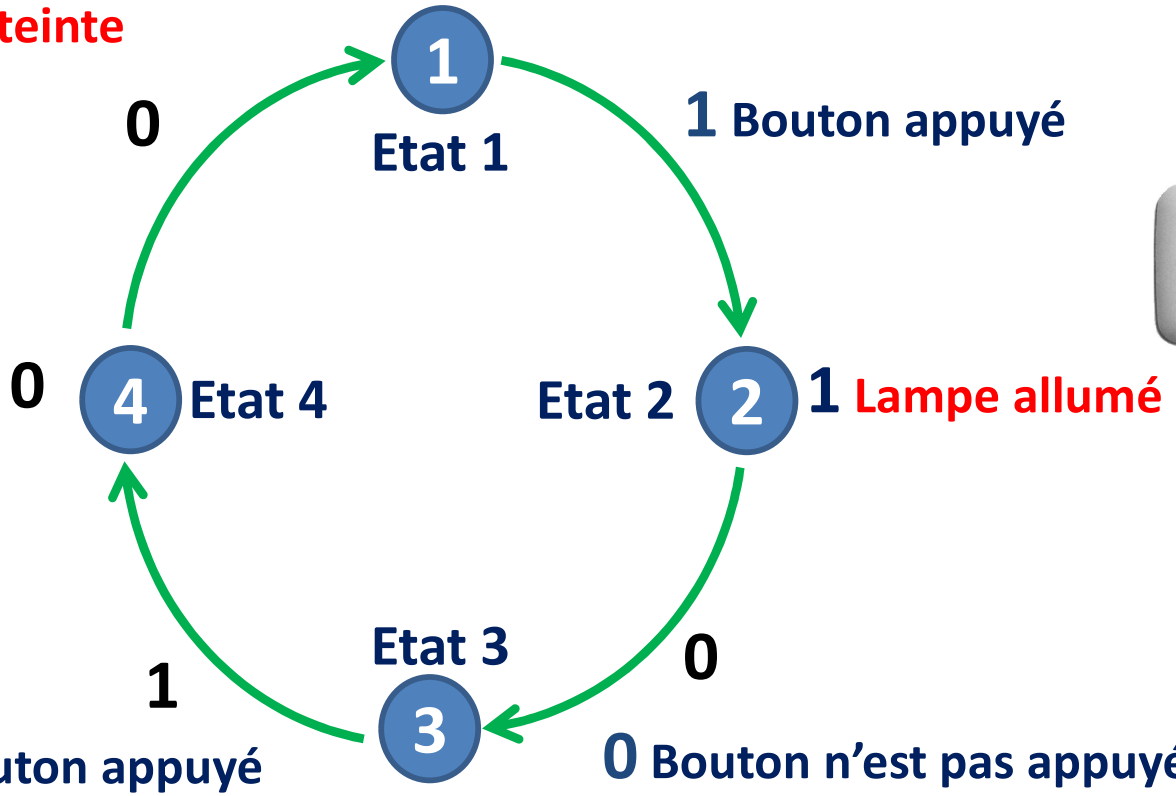
0 Lampe éteinte

0 Bouton n'est pas appuyé

1 Lampe toujours éteinte



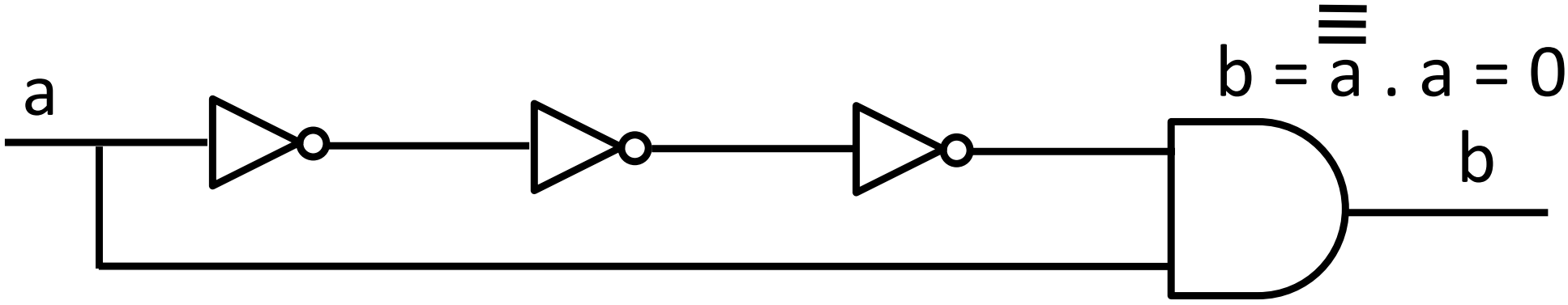




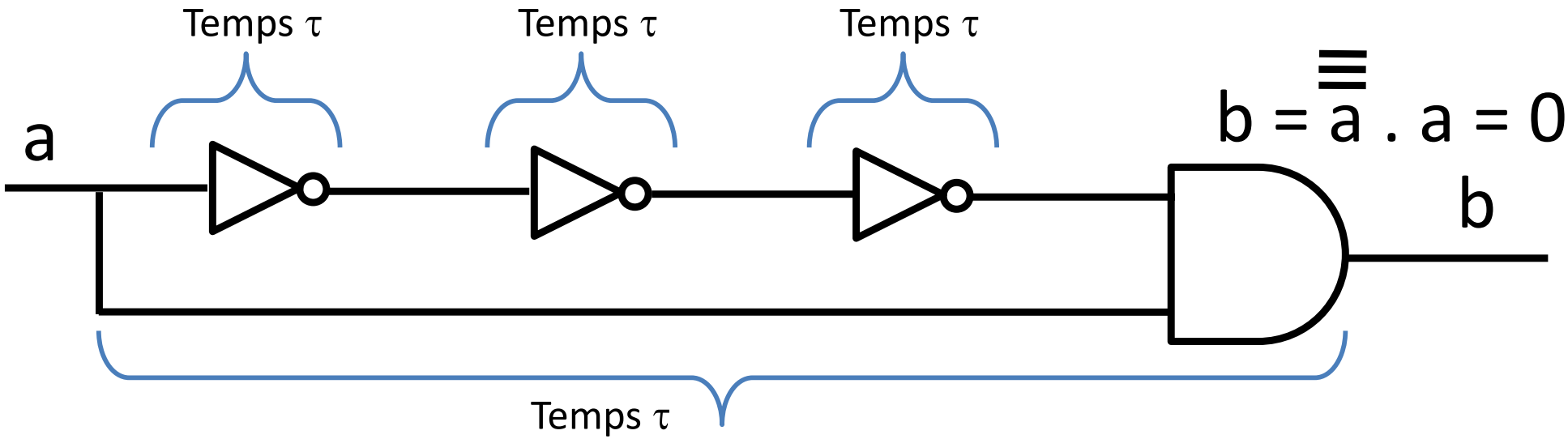
Délai de propagation des signaux :

La conception des circuits logiques séquentiels prend en compte une propriété que nous avons ignorée dans les circuits combinatoires. Il s'agit du délai de propagation des signaux électriques à travers les portes logiques.

Exemple :

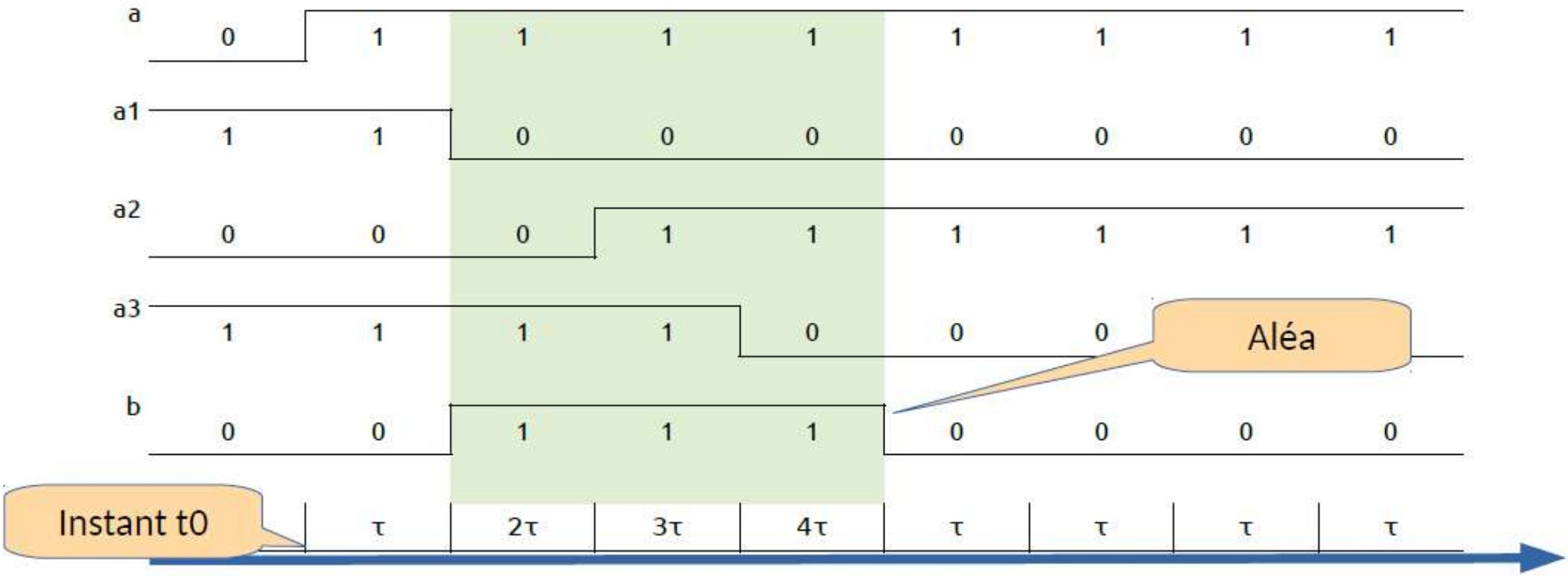


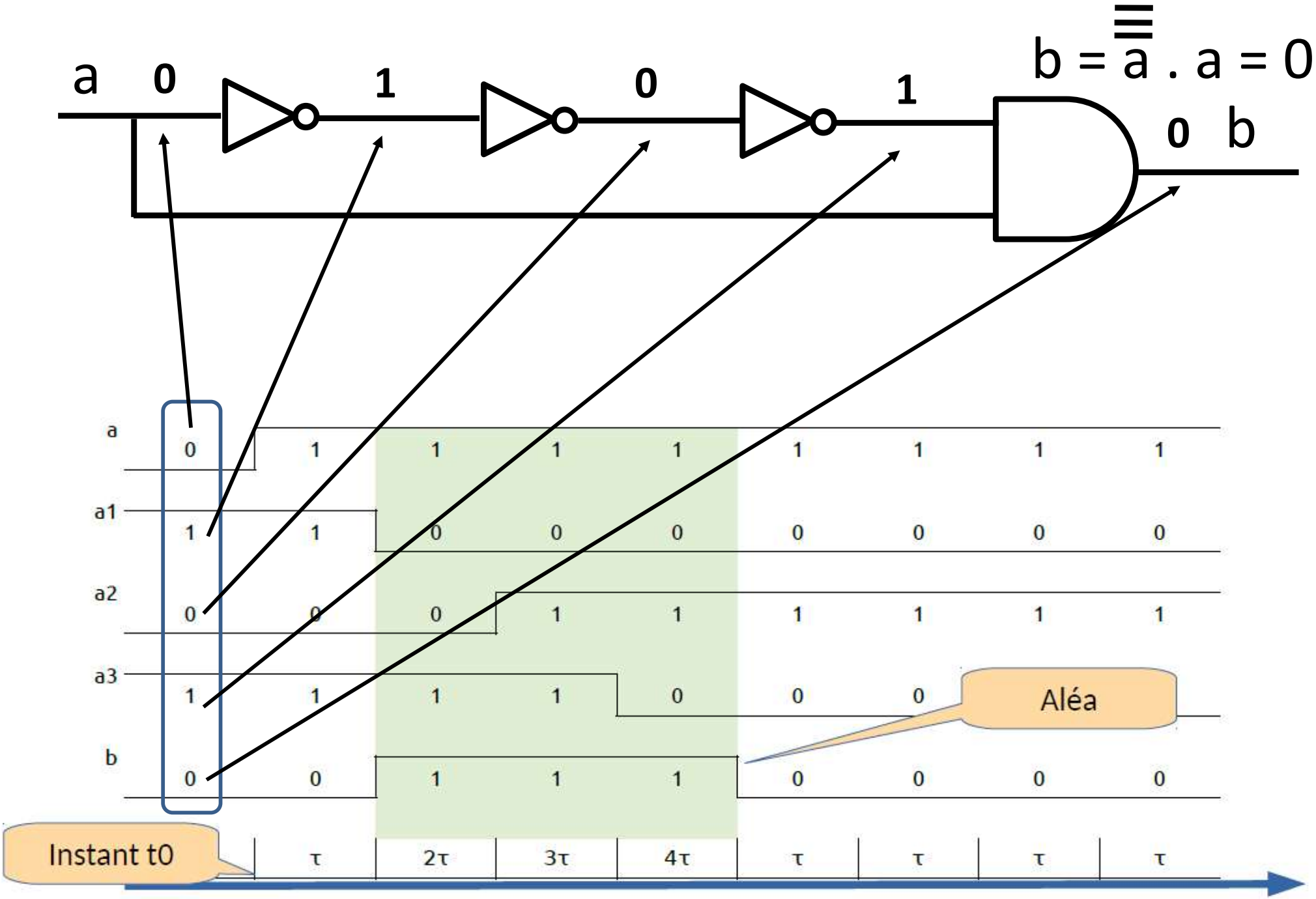
En effet, nous avons supposé que dès qu'un changement se produits dans les entrées d'un circuit combinatoire, immédiatement les sorties sont affectées. En réalité, chaque porte logiques mettra toujours un certain temps (dit temps de propagation = Temps τ) pour fait passer un signal électrique depuis son entrée vers sa sortie.

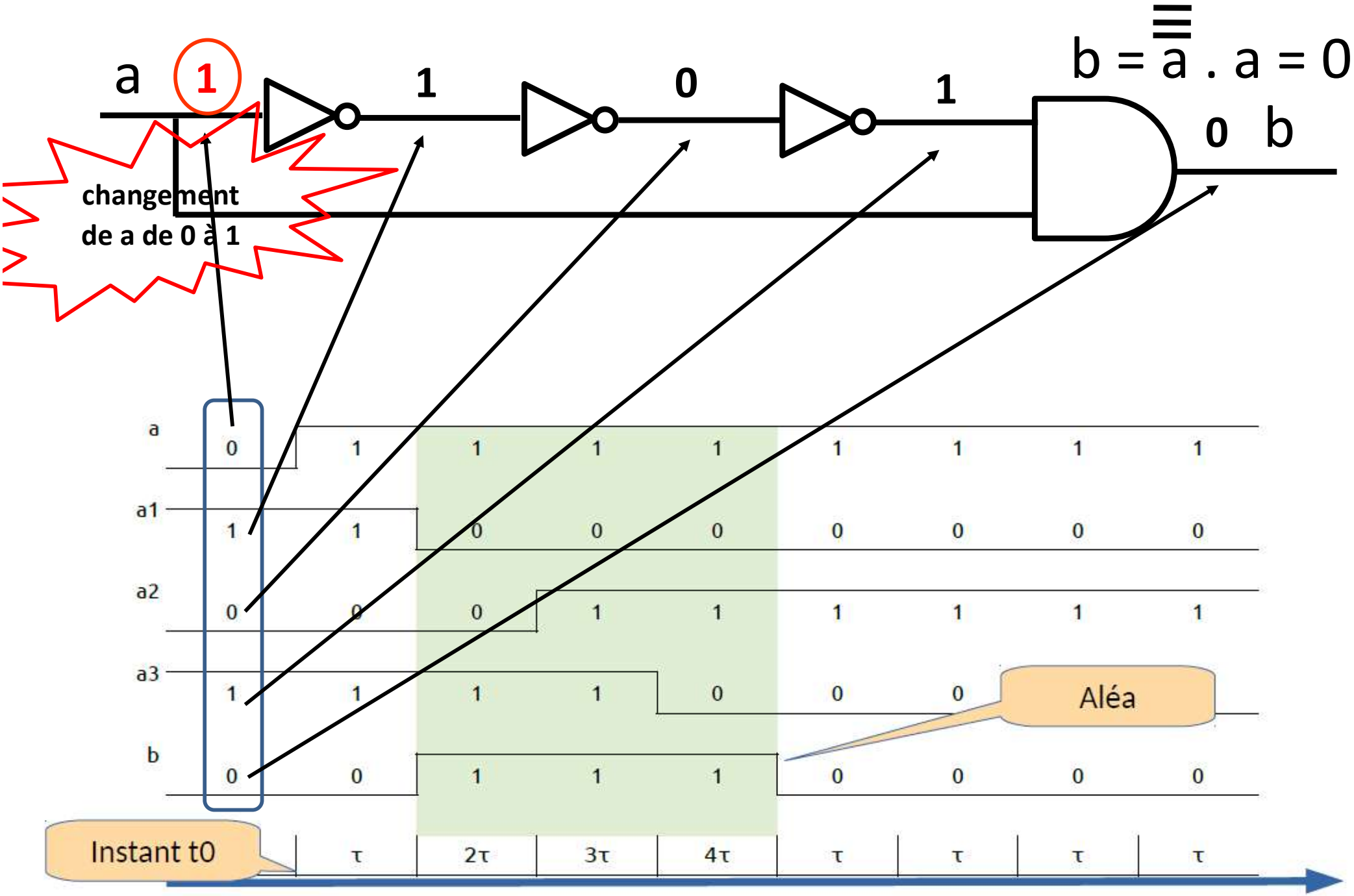


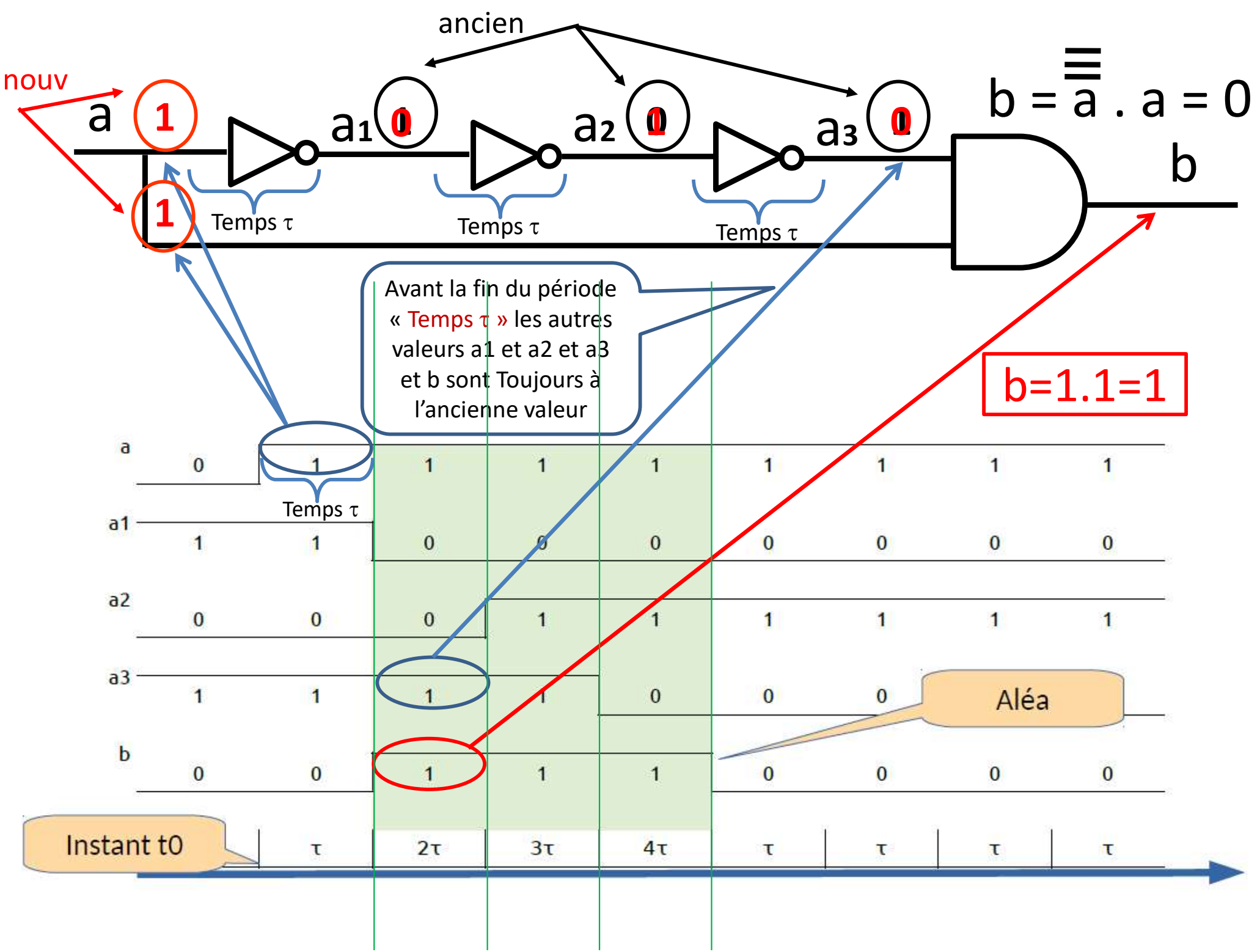
En principe, en se basant sur les circuits logiques combinatoires, la sortie de ce circuit devrait être toujours égale à **0**.
Vous allez voir, en réalité qu'il y a un petit aléa (problème) qui se produit lorsque qu'on change la valeur de l'entrée « a ».
Supposons que chacune des portes (ici «**NOT**» et «**AND**») mette un temps égale à τ.
Traçons dans un graphique temporelle (que l'on appel chronogramme) l'évolution de nos signaux: (a, a1, a2, a3 et b).

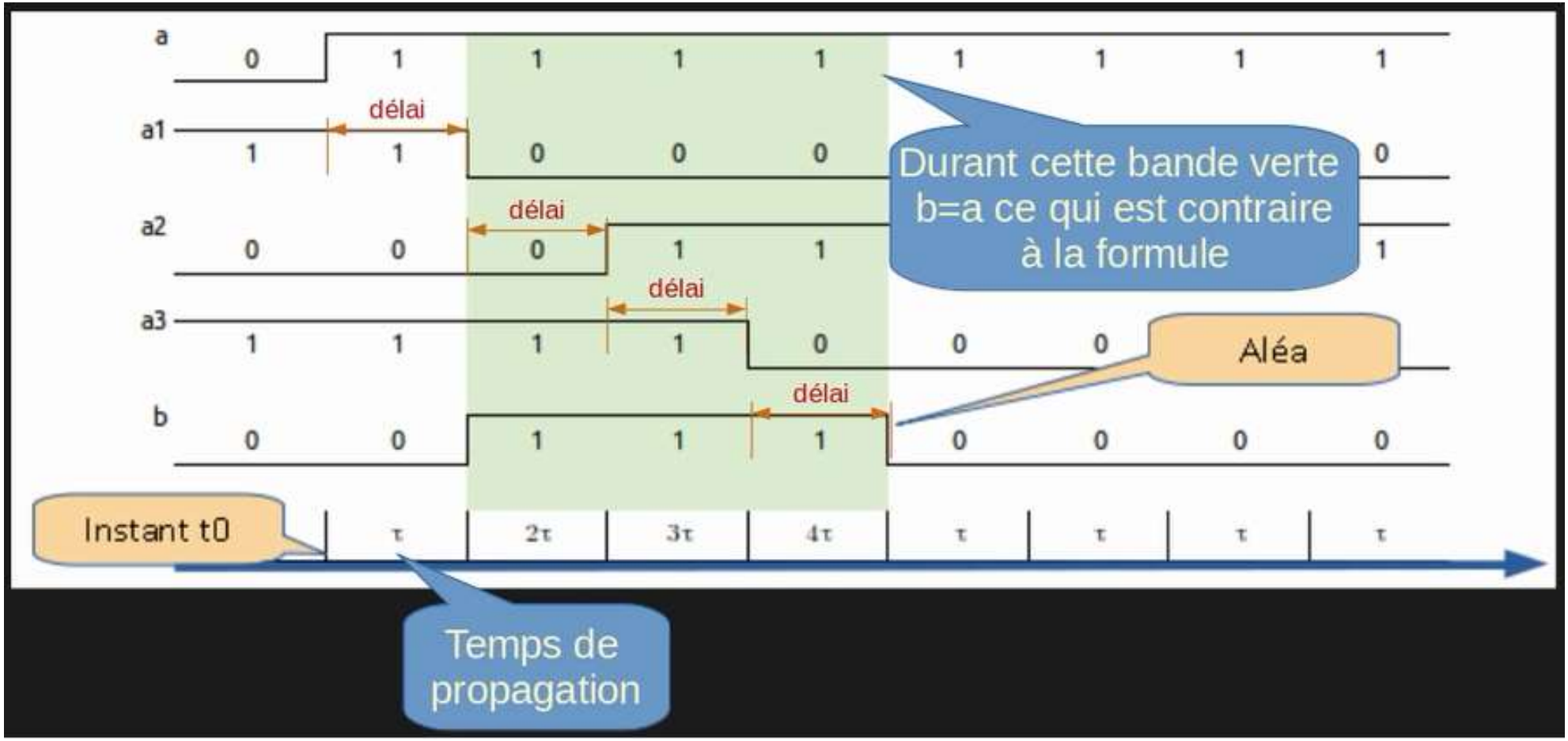
Remarque: Le schéma temporelle indiqué ci-dessous est appelée « **chronogramme ».** Il est souvent utilisé pour montrer les fonctions des circuits logiques séquentiels.




a
0
1
0
1
b = $\overline{a}$. a = 0
0 b
a
a1
a2
a3
b
0 1 1 1 1 1 1 1 1
1 1 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1
1 1 1 1 0 0 0
0 0 1 1 1 0 0 0 0
Aléa
Instant t0
τ 2τ 3τ 4τ τ τ τ τ


a
1
1
0
1
b = $\overline{a}$. a = 0
0 b
changement
de a de 0 à 1
a
a1
a2
a3
b
0 1 1 1 1 1 1 1 1
1 1 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1
1 1 1 1 0 0 0 0 0
0 0 1 1 1 0 0 0 0
Aléa
Instant t0
τ 2τ 3τ 4τ τ τ τ τ


ancien
nouv
a
1
1
a1
0
a2
0
a3
0
b = $\overline{a}$. a = 0
b
Temps τ
Temps τ
Temps τ
Avant la fin du période « Temps τ » les autres valeurs a1 et a2 et a3 et b sont Toujours à l'ancienne valeur
b=1.1=1
Temps τ
a
a1
a2
a3
b
Aléa
Instant t0
τ
2τ
3τ
4τ



On suppose que « **a** » passe de **« 0 » vers « 1 »** à l'instant t_0.

- A l'instant ($t_0+\tau$) que «a_1» passe à 0 (puis «a_1» est l'inverse de « a »).

• à l’instant ($t_{0+2\tau}$) que « a_2 » passe à 1 (puis « a_2 » est l’inverse de « a_1 »).

• à l’instant ($t_{0+3\tau}$) que « a_3 » passe à 0 (puis « a_3 » est l’inverse de « a_2 »).

• Il faut remarquer qu’entre (t_0) et ($t_{0+3\tau}$) « a » et « a_3 » sont tous les 2 à « 1 » au même temps ce qui donne une situations particulière pour « b » égale à « 1 » .

types de circuits séquentiels :

Il y a deux types de circuits séquentiels :

- ***synchrone****, où le comportement est déterminé par les signaux à* des instants discrets (mesurés, modérés) de temps (cadencé au rythme d'un signal appelé horloge "clock" ou bien CLK) ;
- ***asynchrone****, où le comportement est déterminé par les signaux à n'importe quel instant, et* l'ordre avec lequel les entrées varient.

Ainsi, Dans les circuits asynchrones, la sortie est modifiée dès qu'il y a un changement de l'état des entrées. Dans les circuits synchrones, la sortie ne change qu'après un signal d'horloge.

Système Synchrone (Notion de l'horloge)

La synchronisation est obtenue à l'aide d'une horloge, qui fournit un signal périodique.

Ce signal périodique est distribué à l'ensemble du système de sorte que les éléments de stockage sont seulement affectés avec l'arrivée d'un pulse d'horloge.

L'horloge détermine *quand* il y a activité dans le circuit, et les autres signaux déterminent *quoi (ce qui change) dans le circuit.*

Les éléments de stockage qui sont contrôlés par de la transition de l'horloge sont appelés des ***bascules.***

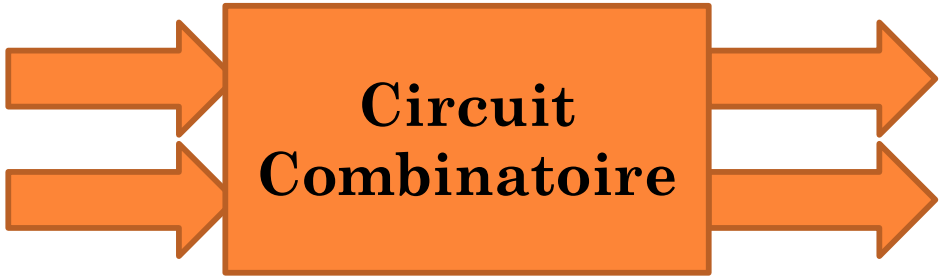
La bascule est l'élément de base permettant la construction de circuits séquentiels:
(**Mémoire**, **Registre**, **Compteur**, etc.).

Une horloge est une variable logique qui passe successivement de 0 à 1 et de 1 à 0 d'une façon périodique.
Cette variable est utilisée souvent comme une entrée des circuits séquentiels (l'horloge est notée par h ou ck (clock)).

Rappel: Circuit Combinatoire/séquentiel

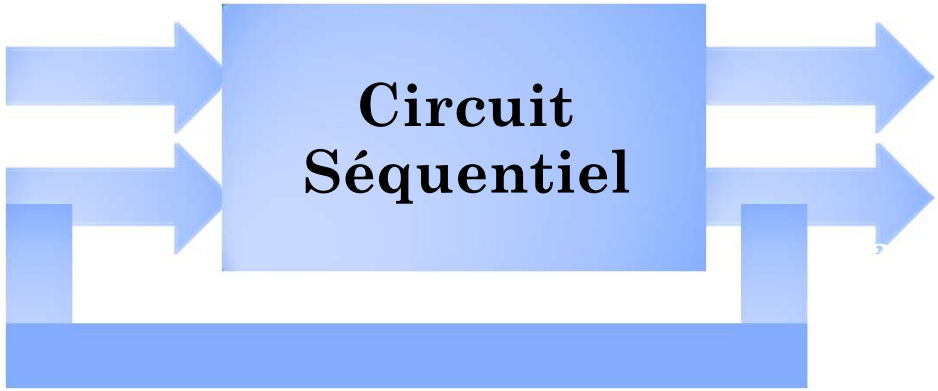
Circuits Combinatoires

- Les fonctions de sortie s'expriment selon des expressions logiques des **seules** variables d'entrée.



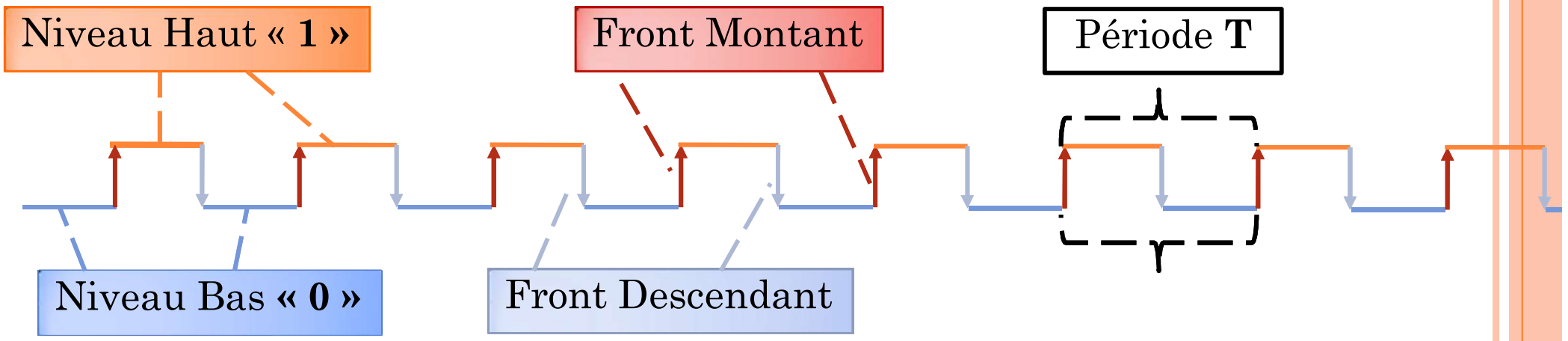
Circuits Séquentiels

- Les fonctions de sortie dépendent non seulement de l'état des variables d'entrée mais également de l'état antérieur (passé) de certaines variables de sortie (propriétés de mémorisation).



NOTION D'HORLOGE

- Une horloge, noté par **H** ou **ck** (clock), est une variable logique qui passe successivement de 0 à 1 et de 1 à 0 d'une façon périodique.



Fréquence = 1/T = nombre de changement par seconde en hertz (Hz)

Horloge	Période
1 Hz	1 seconde
1 Méga Hz	1 milliseconde
1 Giga Hz	1 nanoseconde

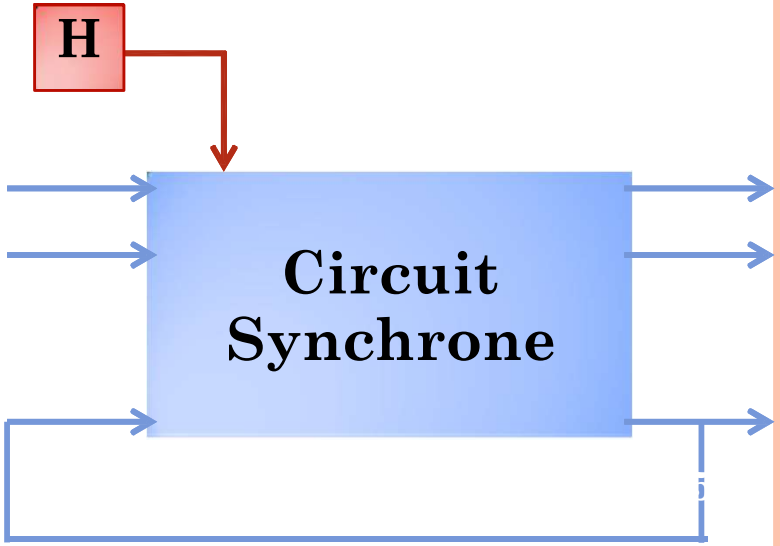
Circuit Asynchrones et Synchrones

Circuits Asynchrones

- Les variables du système évoluent librement au cours du temps.

Circuits Synchrones

- L'évolution des variables dépend **d'une impulsion d'horloge** comme un des signaux d'entrée.



Partie 1:

Bascules

Les bascules

Une bascule est un circuit logique capable, dans certaines circonstances, de maintenir les valeurs de ses sorties malgré les changements de valeurs d'entrées, c'est-à-dire de mémoriser sont état « mémoire ». Il s'agit de l'élément de mémorisation de la logique séquentielle.

On distingue deux catégories principales de bascules :

- les ***bascules asynchrones*** *que l'on nomme verrous (ou latch en anglais)*
- et les ***bascules synchrones*** *(dépendant d'un signal d'horloge) que l'on nomme simplement* bascules (ou *flip-flop en anglais).*

BASCULES

Une bascule :

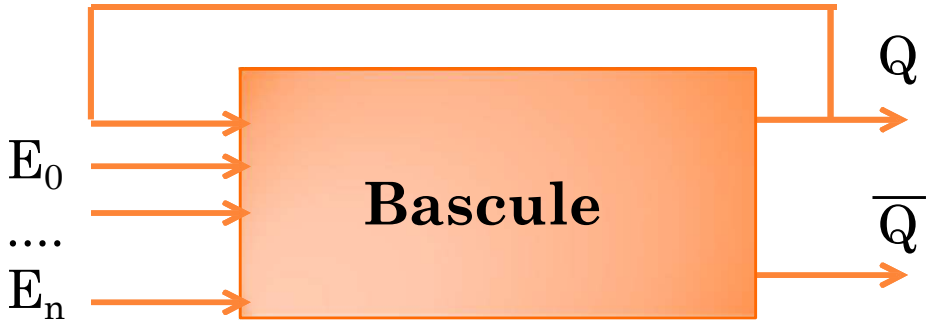
- Est un circuit séquentiel élémentaire permettant de mémoriser une information binaire (bit).
- Peut être synchrone ou asynchrone.
- Possède deux sorties complémentaires Q et $\overline{Q}$.

Exemples:

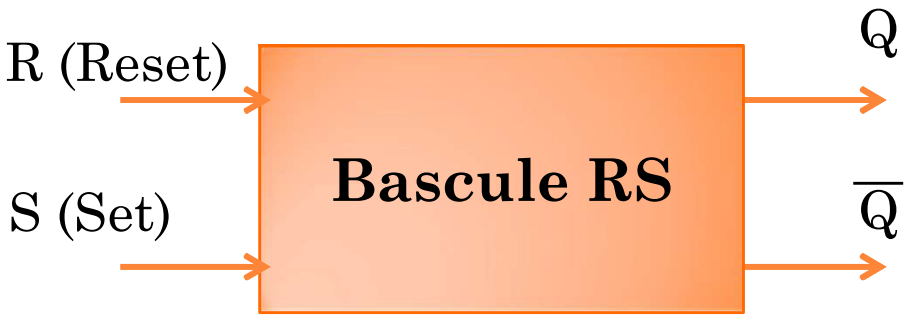
- ✓ Bascule **RS**,
- ✓ Bascule **D** ,
- ✓ Bascule **JK**,
- ✓ Bascule **T** ,..... etc.

$Q^- = Q(t)$

$Q^+ = Q(t+1) = F(Ei, Q^-)$



Bascule RS



R	S	Q+	
0	0	Q-	État mémoire
0	1	1	Mise (Set) à 1
1	0	0	Remise (Reset) à 0
1	1	X	État interdit

Bascule RS définition:

Dans le nom de cette bascules vous remarquez 2 lettres : **« R » et « S ».**

Ce sont les entrées de cette bascule:

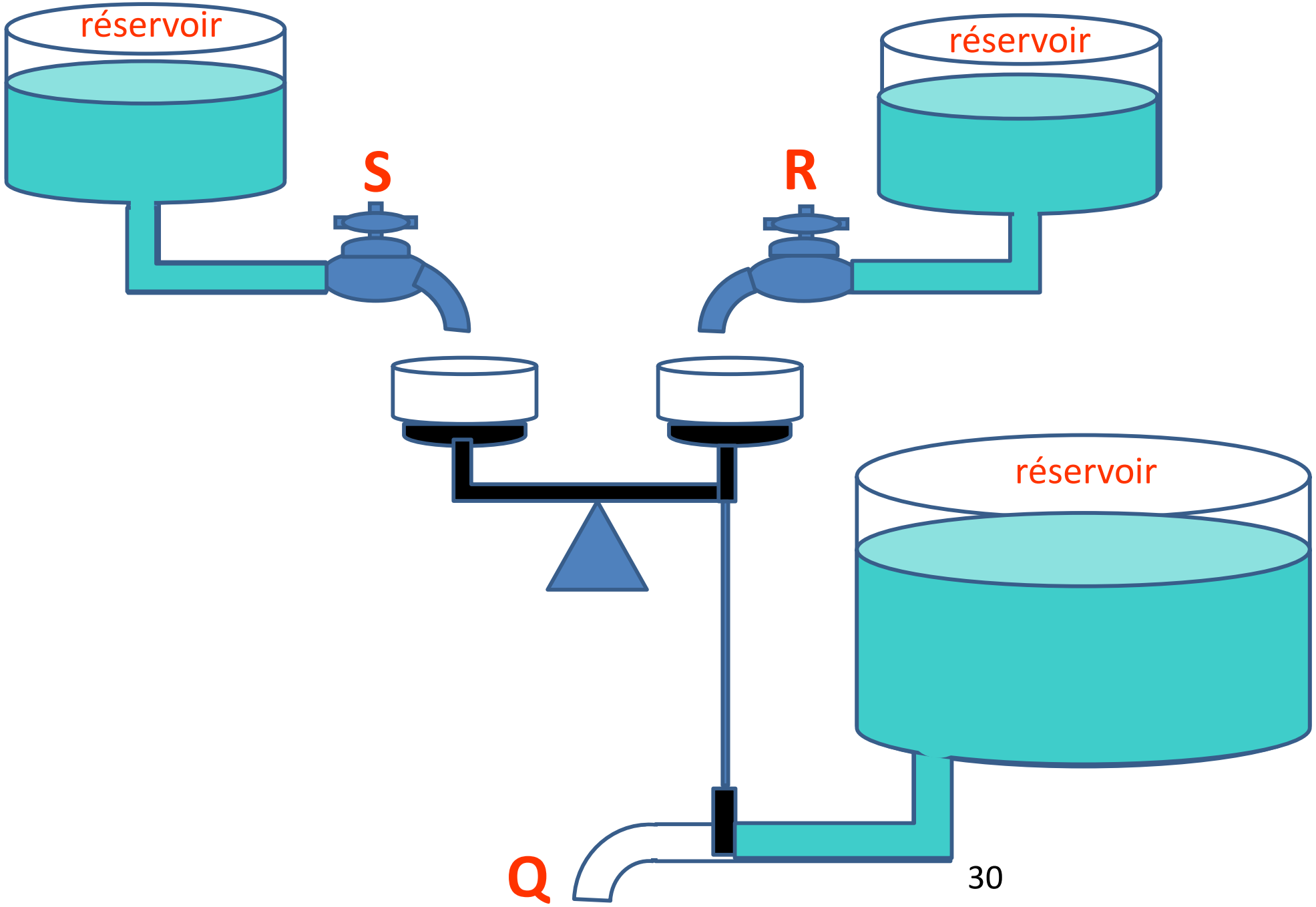
✓ **« R »** pour *reset ou mise à* ***zéro*** ***« 0 »*** *et*

✓ **« S »** pour *set ou mise à* ***un*** ***« 1 »***.

Bien évidement, cette bascule va avoir une sortie que l'on appellera **Q**.

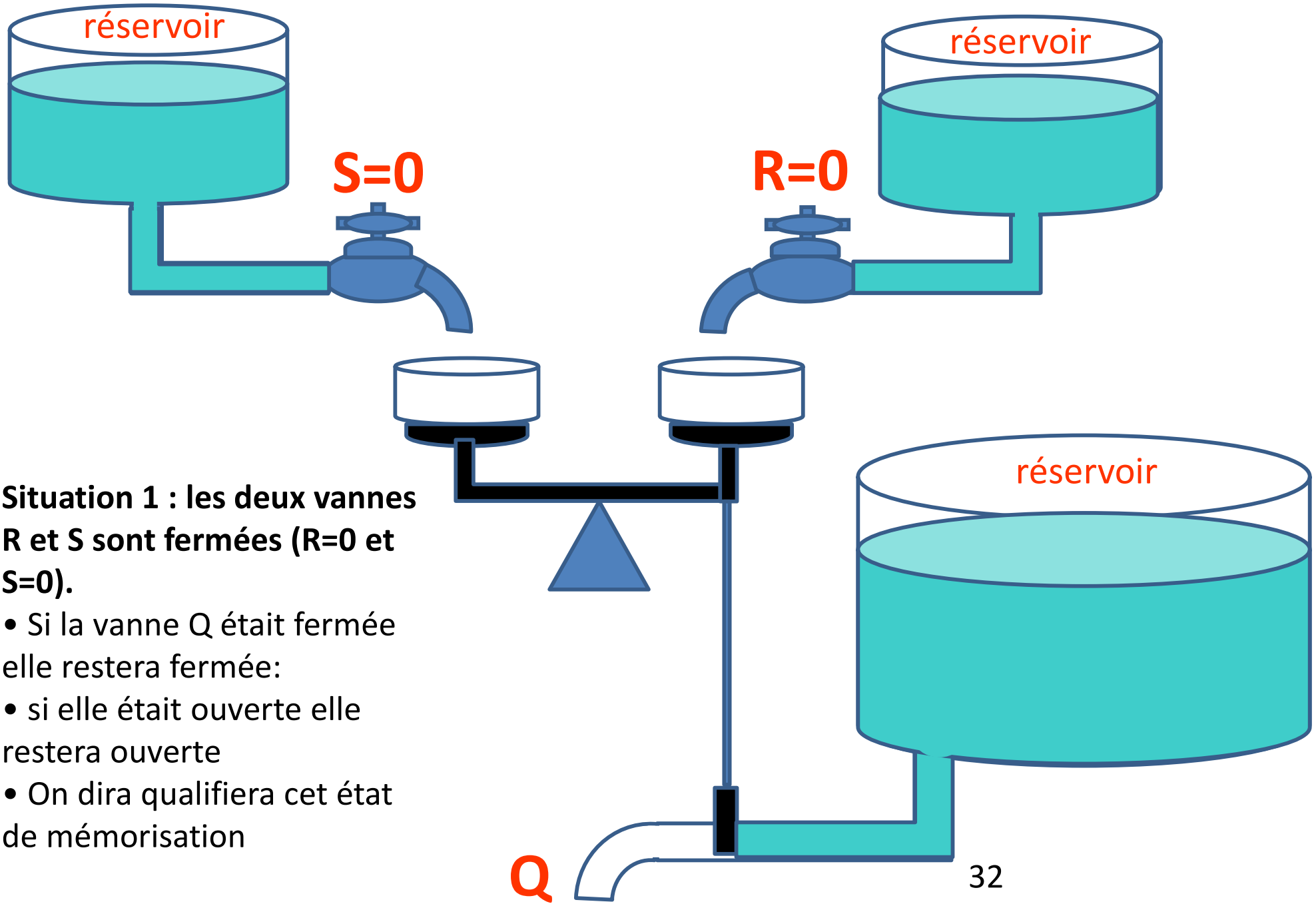
En réalité, nous aurons toujours 2 sorties : **Q et** $\overline{Q}$ **.**

Exemple de fonctionnement d'une bascule RS :



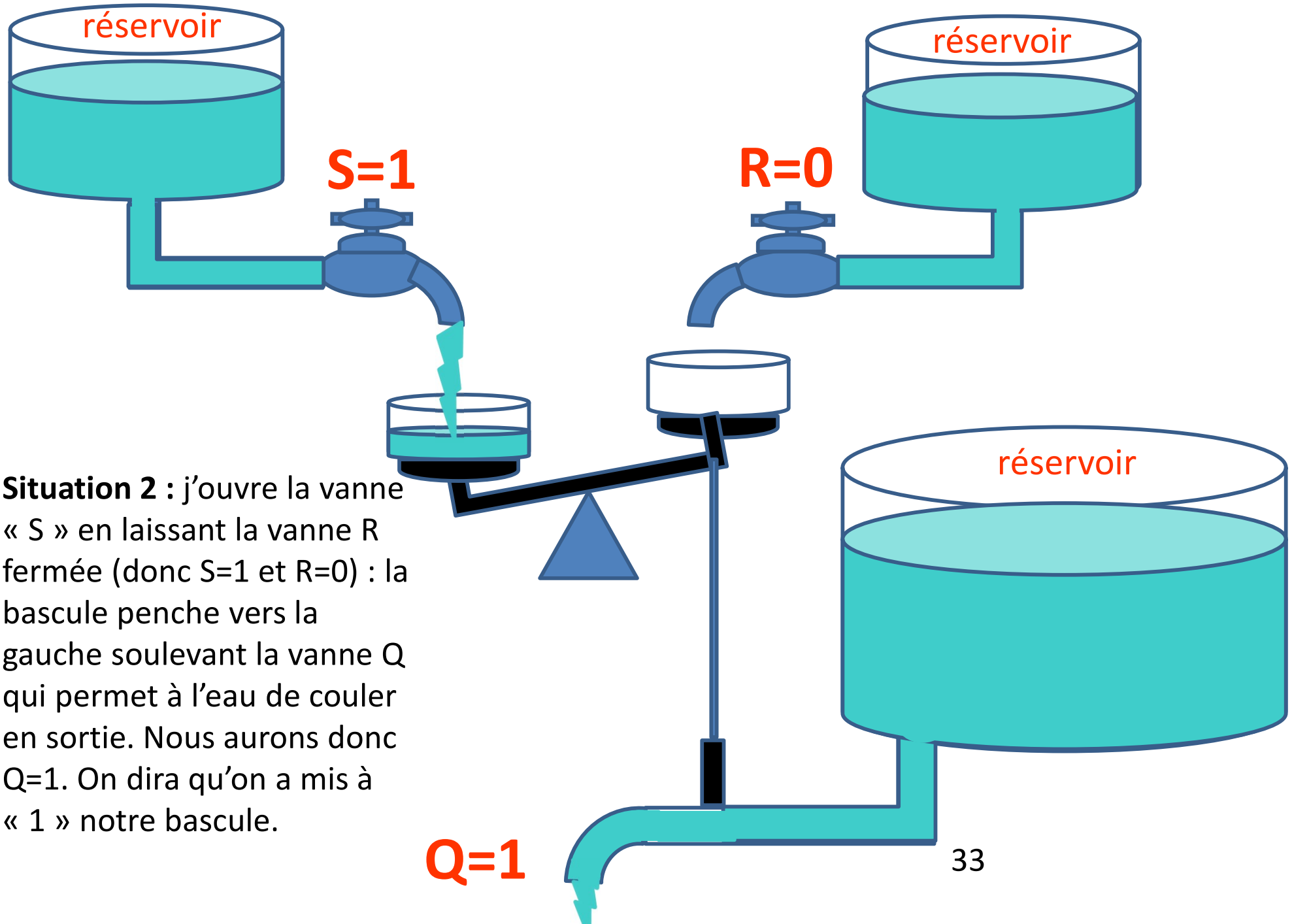
Exemple de fonctionnement d'une bascule RS :

- deux vannes R et S
- une bascules
- une sortie Q qui est un tuyaux
- R = 1 signifie que le robinet R est ouvert (idem pour S)
- R = 0 signifie que le robinet R est fermé (idem pour S)
- la sortie Q (tuyau de sortie) est à « 1 » veut dire qu'il fait sortir de l'eau
- la sortie Q (tuyau de sortie) est à « 0 » veut dire qu'il ne fait pas sortir de l'eau
- la partie droite de la bascule est liée à une tige qui actionne une vanne pour fermer ou ouvrir le tuyau de sortie Q.

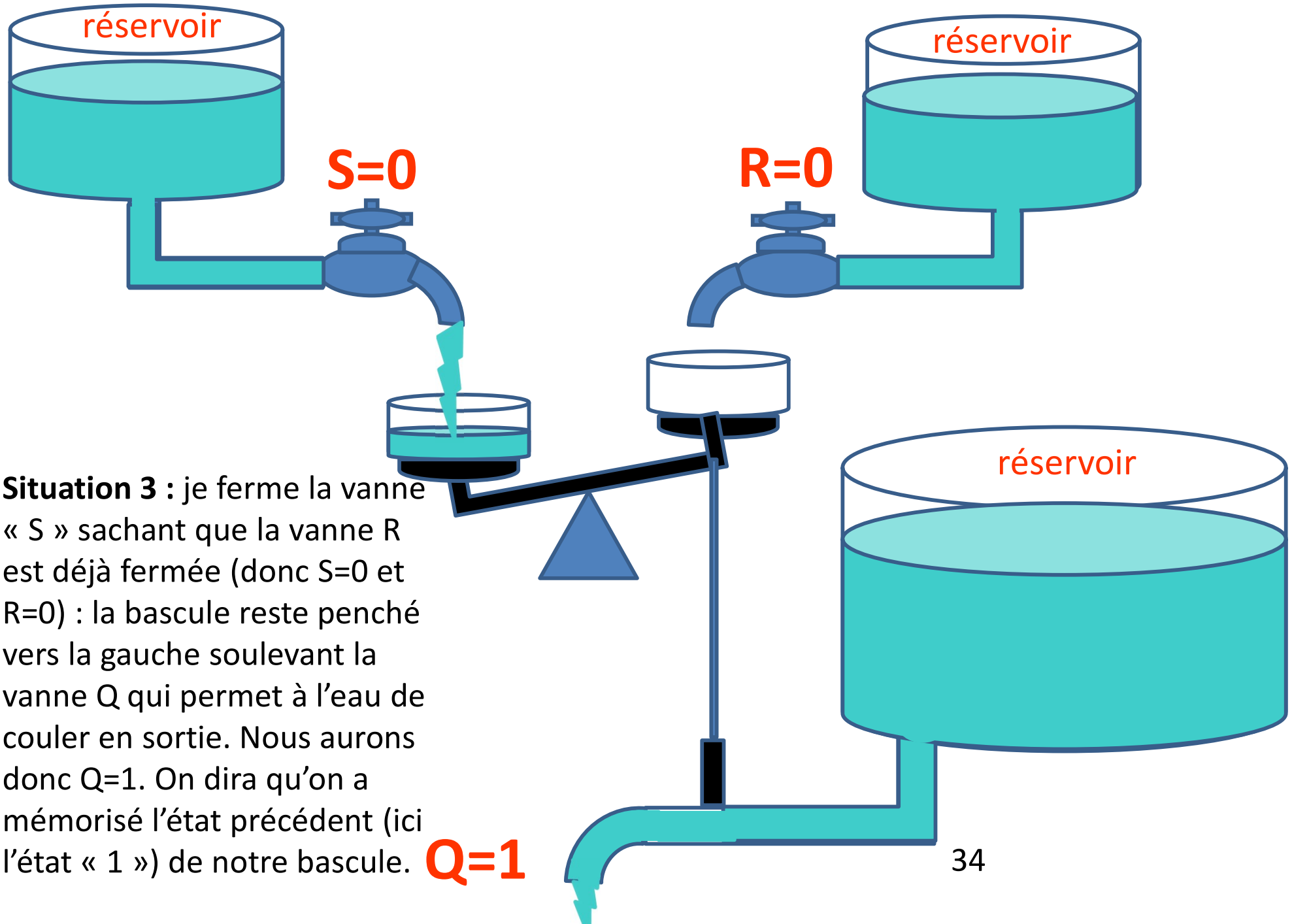


Situation 1 : les deux vannes R et S sont fermées (R=0 et S=0).

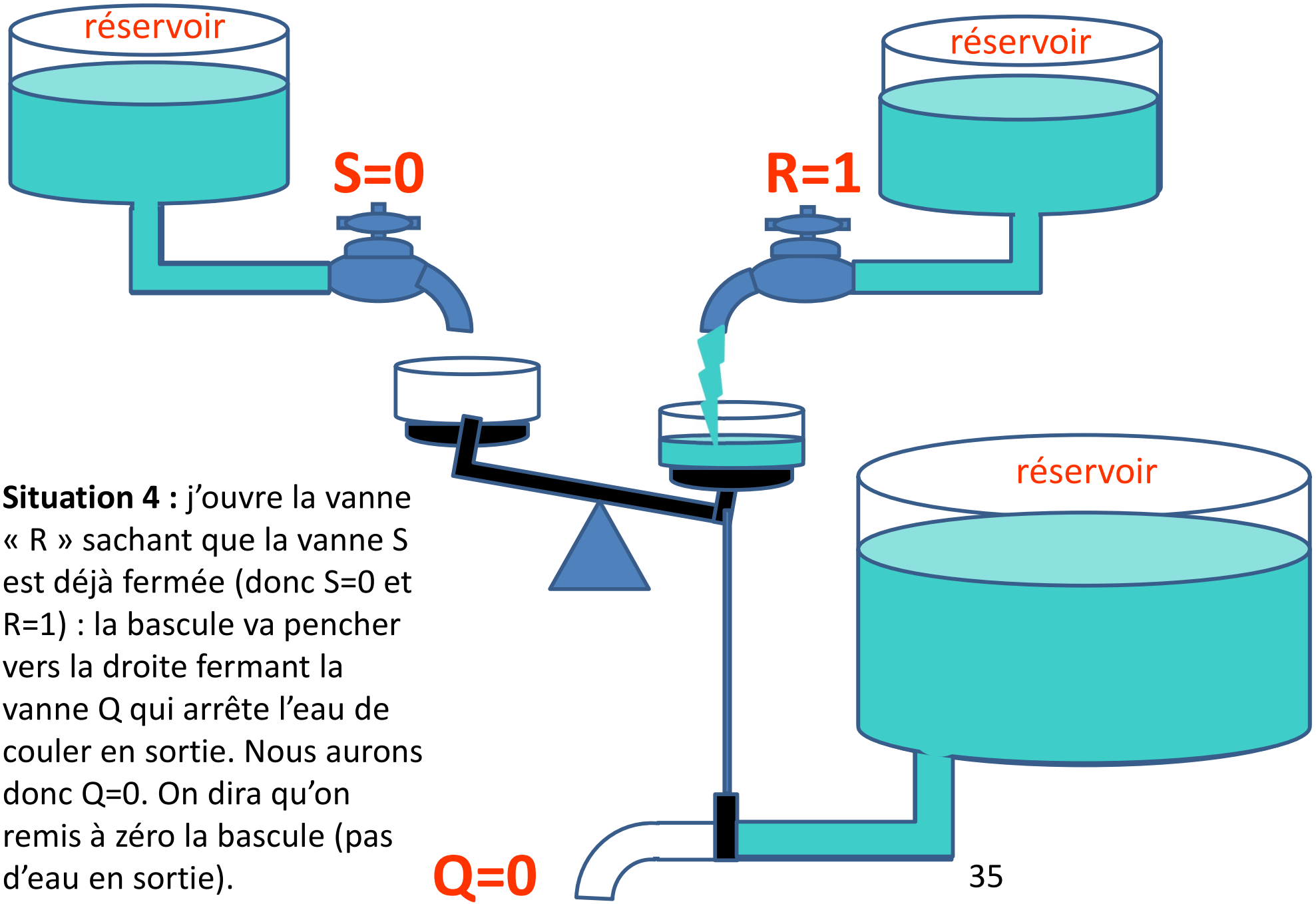
- Si la vanne Q était fermée elle restera fermée:
- si elle était ouverte elle restera ouverte
- On dira qualifiera cet état de mémorisation



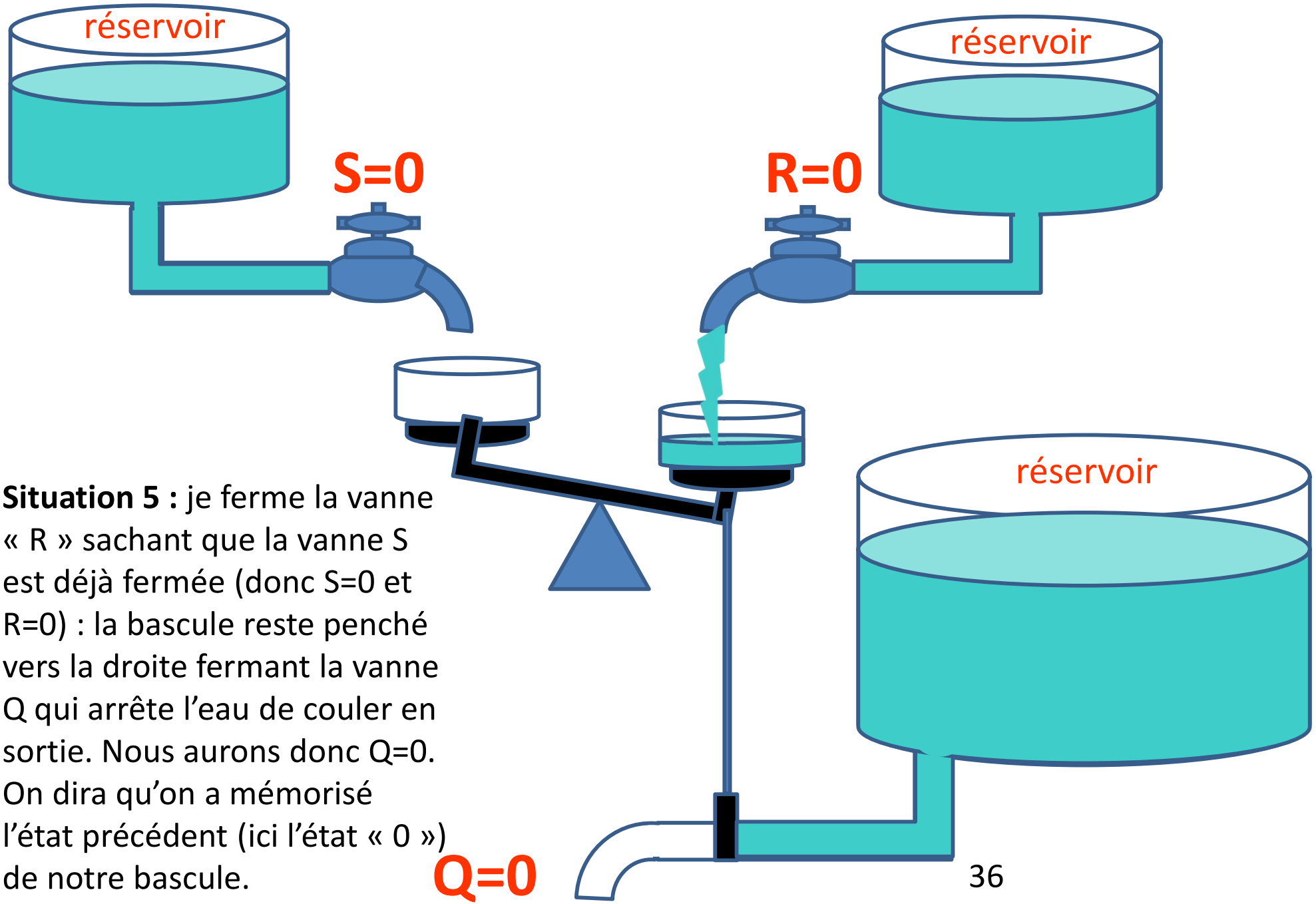
Situation 2 : j'ouvre la vanne « S » en laissant la vanne R fermée (donc S=1 et R=0) : la bascule penche vers la gauche soulevant la vanne Q qui permet à l'eau de couler en sortie. Nous aurons donc Q=1. On dira qu'on a mis à « 1 » notre bascule.



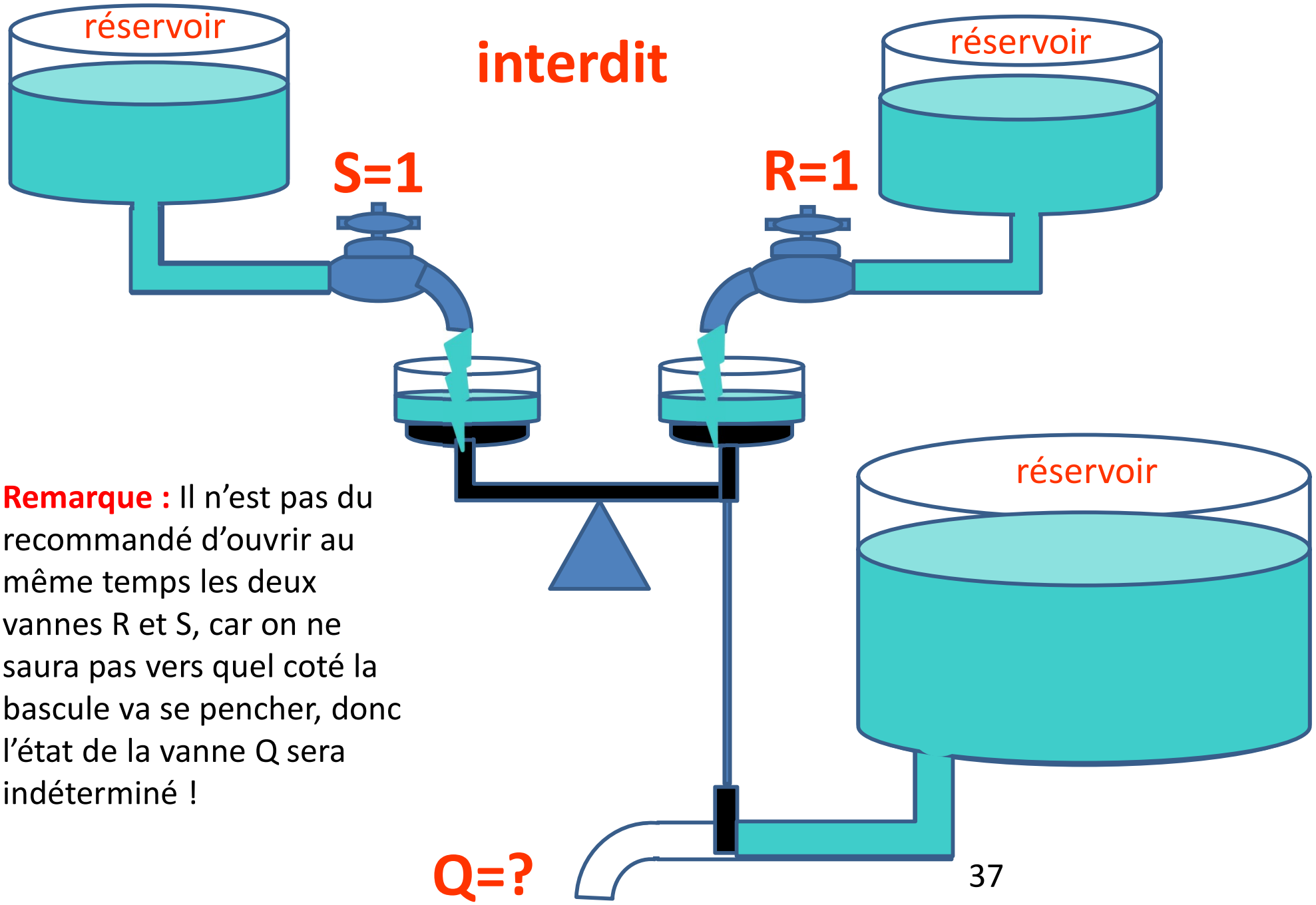
Situation 3 : je ferme la vanne « S » sachant que la vanne R est déjà fermée (donc S=0 et R=0) : la bascule reste penché vers la gauche soulevant la vanne Q qui permet à l'eau de couler en sortie. Nous aurons donc Q=1. On dira qu'on a mémorisé l'état précédent (ici l'état « 1 ») de notre bascule.



Situation 4 : j'ouvre la vanne « R » sachant que la vanne S est déjà fermée (donc S=0 et R=1) : la bascule va pencher vers la droite fermant la vanne Q qui arrête l'eau de couler en sortie. Nous aurons donc Q=0. On dira qu'on remis à zéro la bascule (pas d'eau en sortie).



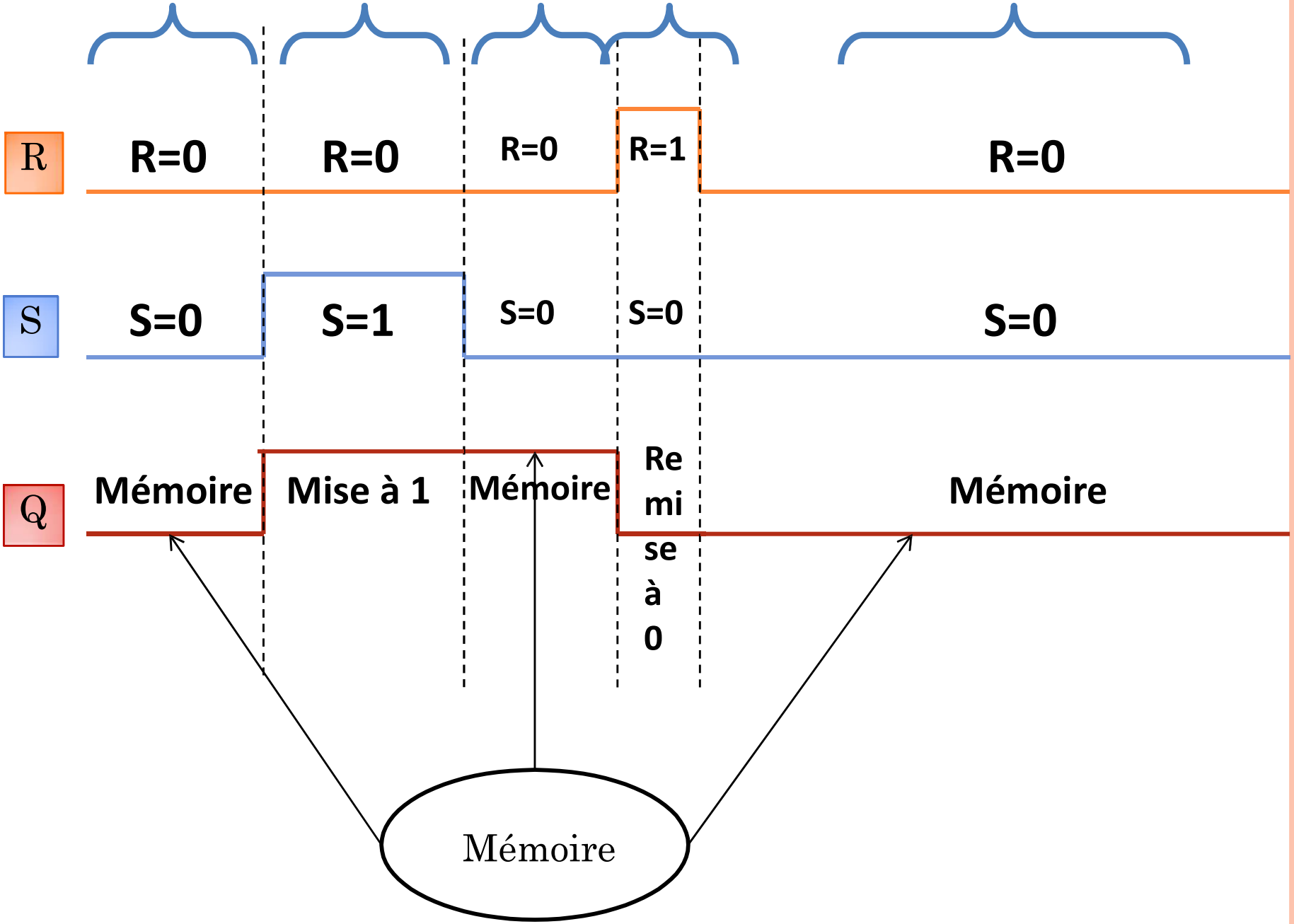
Situation 5 : je ferme la vanne « R » sachant que la vanne S est déjà fermée (donc S=0 et R=0) : la bascule reste penché vers la droite fermant la vanne Q qui arrête l'eau de couler en sortie. Nous aurons donc Q=0. On dira qu'on a mémorisé l'état précédent (ici l'état « 0 ») de notre bascule.



Remarque : Il n'est pas du recommandé d'ouvrir au même temps les deux vannes R et S, car on ne saura pas vers quel coté la bascule va se pencher, donc l'état de la vanne Q sera indéterminé !

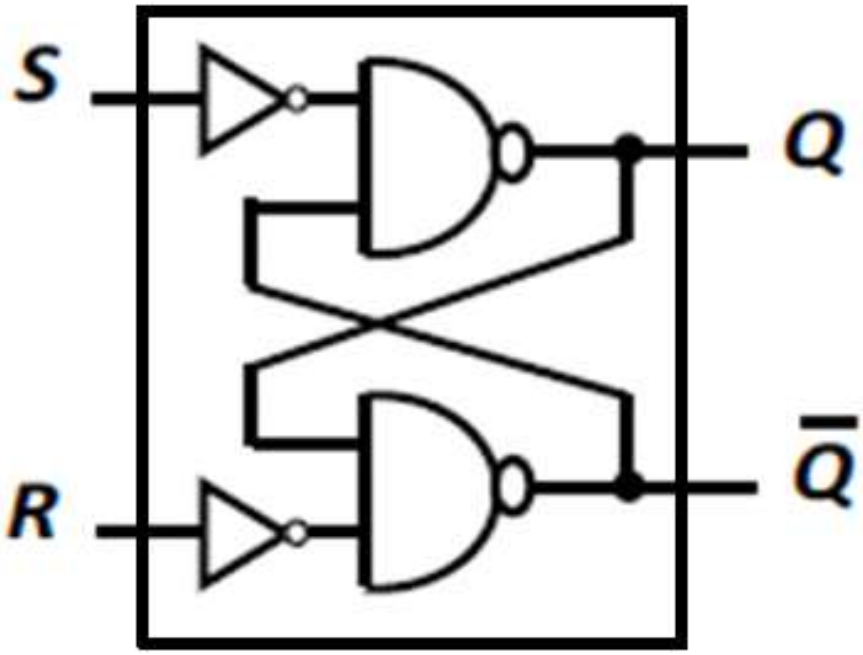
Bascule RS

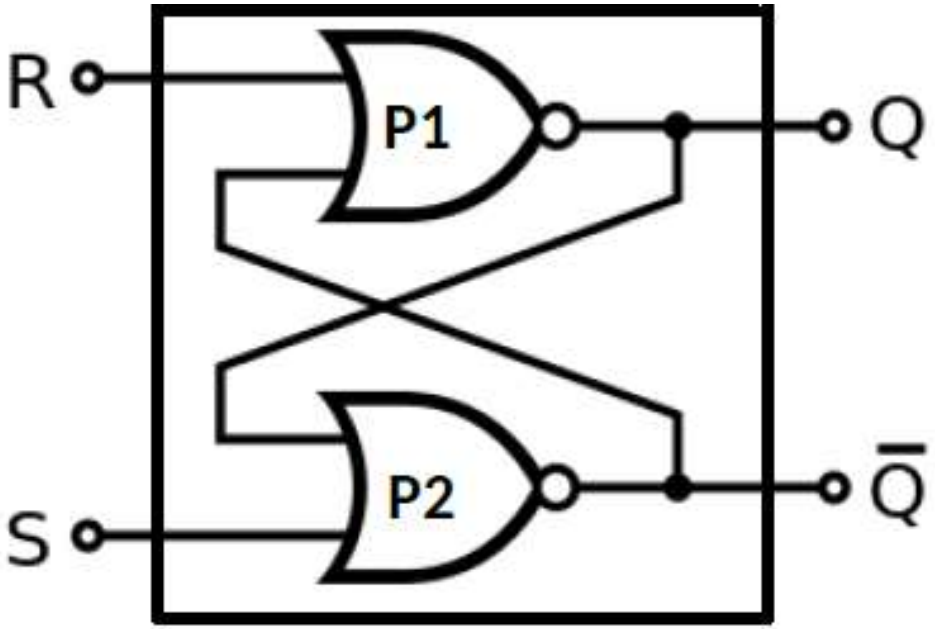
Chronogramme



Bascule RS

Structure Interne





Bascule RS

Structure Interne

Rappelez-vous que le but d'une bascule est de remplir la fonction de mémorisation.
Nous utiliserons les 2 entrées R et S pour mettre à « 1 » ou mettre à « 0 » cette bascule.
Lorsque R et S sont à « 0 » toutes les deux, la bascule doit mémoriser sont état (précédent).
Vous allez voir qu'on va éviter de mettre R et S au même temps à « 1 ».

Bascule RS

Structure Interne

Vérifions ce fonctionnement en analysant le circuit de la base RS à base des portes NOR :

On peut représenter la fonction Q par :
deux états Q+ et Q.
Q+ indiquant l'état future de la bascule et
Q indiquant son état présent.
Ainsi, nous pourrons écrire Q+ =*f(R, S, Q)*.

Bascule RS

Structure Interne

R	S	Q+	
0	0	Q-	État mémoire
0	1	1	Mise (Set) à 1
1	0	0	Remise (Reset) à 0
1	1	X	État interdit

R	S	Q-	Q+
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	X
1	1	1	X

Bascule RS

Structure Interne

R	S	Q-	$\overline{Q}$-	Q+	$\overline{Q}$+
0	0	0	1	0	1
0	0	1	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	X	X
1	1	1	0	X	X

RS / Q-	00	01	11	10
0	0	1	X	0
1	1	1	X	0

$Q+ = S + \overline{R}\ Q-$

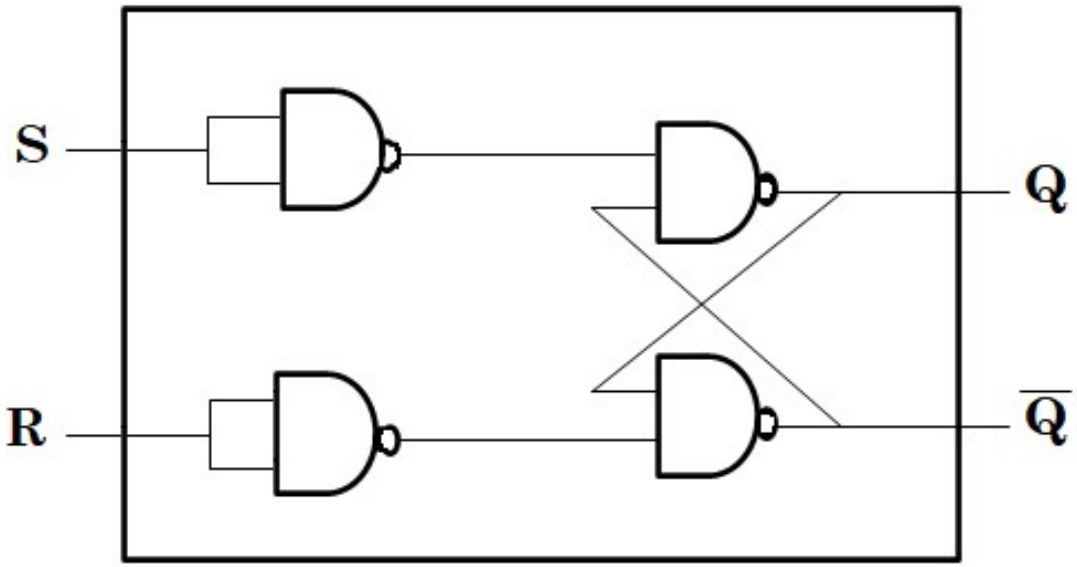
RS / $\overline{Q}$-	00	01	11	10
0	0	0	X	1
1	1	0	X	1

$\overline{Q}+ = R + \overline{S}\ \overline{Q}-$

Bascule RS

Structure Interne

- **Exercice 1:** Réaliser la bascule RS en utilisant seulement des portes NAND.

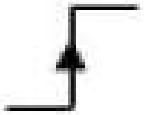
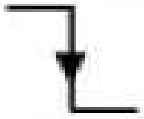


BASCULES SYNCHRONES

Le changement d'état d'une sortie d'un système séquentiel synchrone dépend de **l'état des entrées de commande** et de celui du **signal actif** de synchronisation appelé **signal d'horloge.**

➢ Modes d'action du signal d'horloge:

Il existe quatre modes d'actions d'horloge connus par les symboles suivants (voir figure) :

Symbole	Désignation
	Niveau Haut
	Niveau Bas
	Front montant
	Front descendant

BASCULES SYNCHRONES

Suivant le mode d'action d'horloge, on distingue deux familles de bascules :

- Bascules à commande par **niveau d'horloge** **(niveau haut ou niveau bas) :**

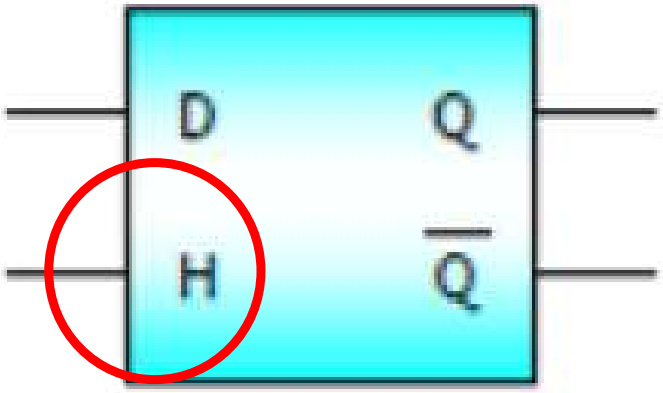
On dit que la bascule est commandée de manière *statique* *(active sur niveau haut ou active sur niveau bas).*

- Bascules à commande par **front d'horloge** **(front montant ou front descendant) :**

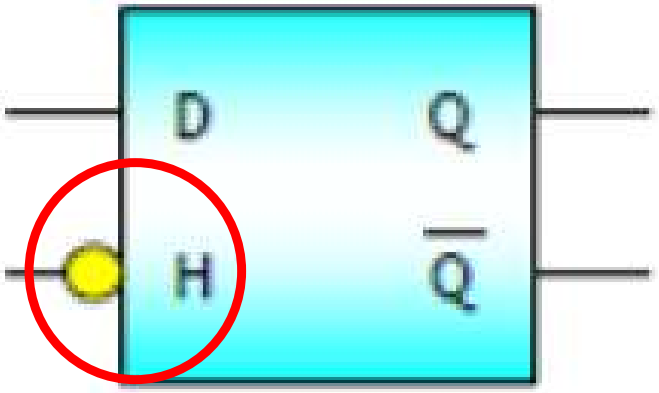
On dit que la bascule est commandée de manière *dynamique* *(active sur front* montant ou active sur front descendant).

BASCULES SYNCHRONES

Symbole :

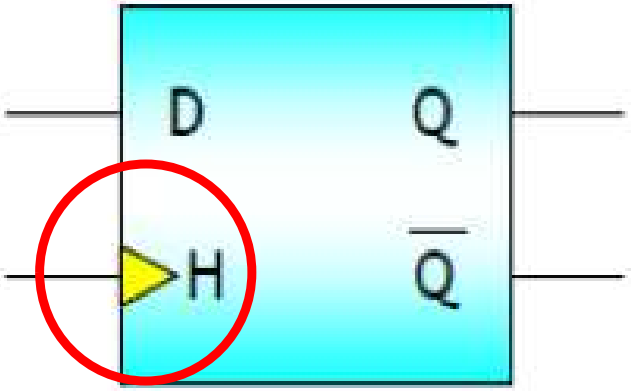


Active sur niveau haut de H

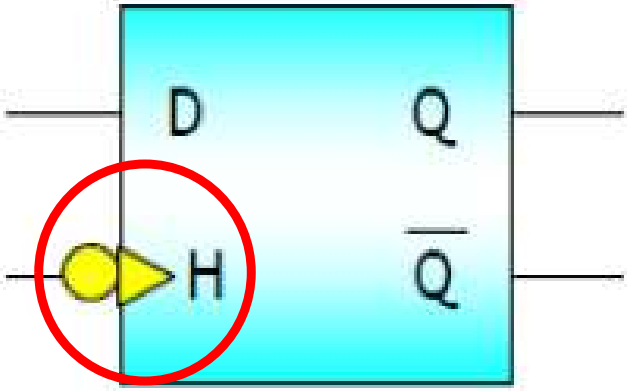


Active sur niveau bas de H

Symbole :



Active sur ↑ de H



Active sur ↓ de H

Bascules Synchrones RSH

Bascule Synchrone RSH ou RST (RS avec activation)

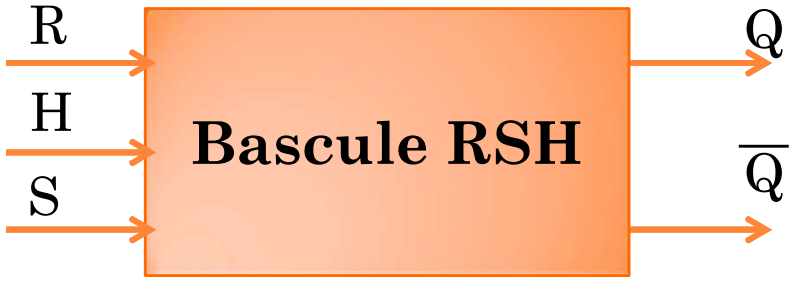
On peut facilement modifier la bascule RS pour la rendre synchrone, c'est à dire de ne permettre de changement que lorsque le signal d'horloge vaut 1(ou 0).

La bascule R.S.H est une bascule pour laquelle les entrées S et R ne sont prises en compte qu'en coïncidence avec un signal de commande.

Ce signal peut être fourni par une horloge, nous avons alors une bascule synchrone.

Alors, la bascule RSH est une bascule **RS** dont les ordres **Set et Reset** ne changent l'état de la sortie **Q** qu'après l'autorisation d'un signal d'horloge **H (Clock CK)**

Bascule RSH: bascule RS avec un signal d'horloge



La bascule RSH est une : bascule RS synchronisée avec un signal d'horloge H

H	R	S	Q+	
0	**X**	**X**	**Q-**	**Mémorisation**
1	0	0	Q-	**Bascule RS**
1	0	1	1	
1	1	0	0	
1	1	1	X	

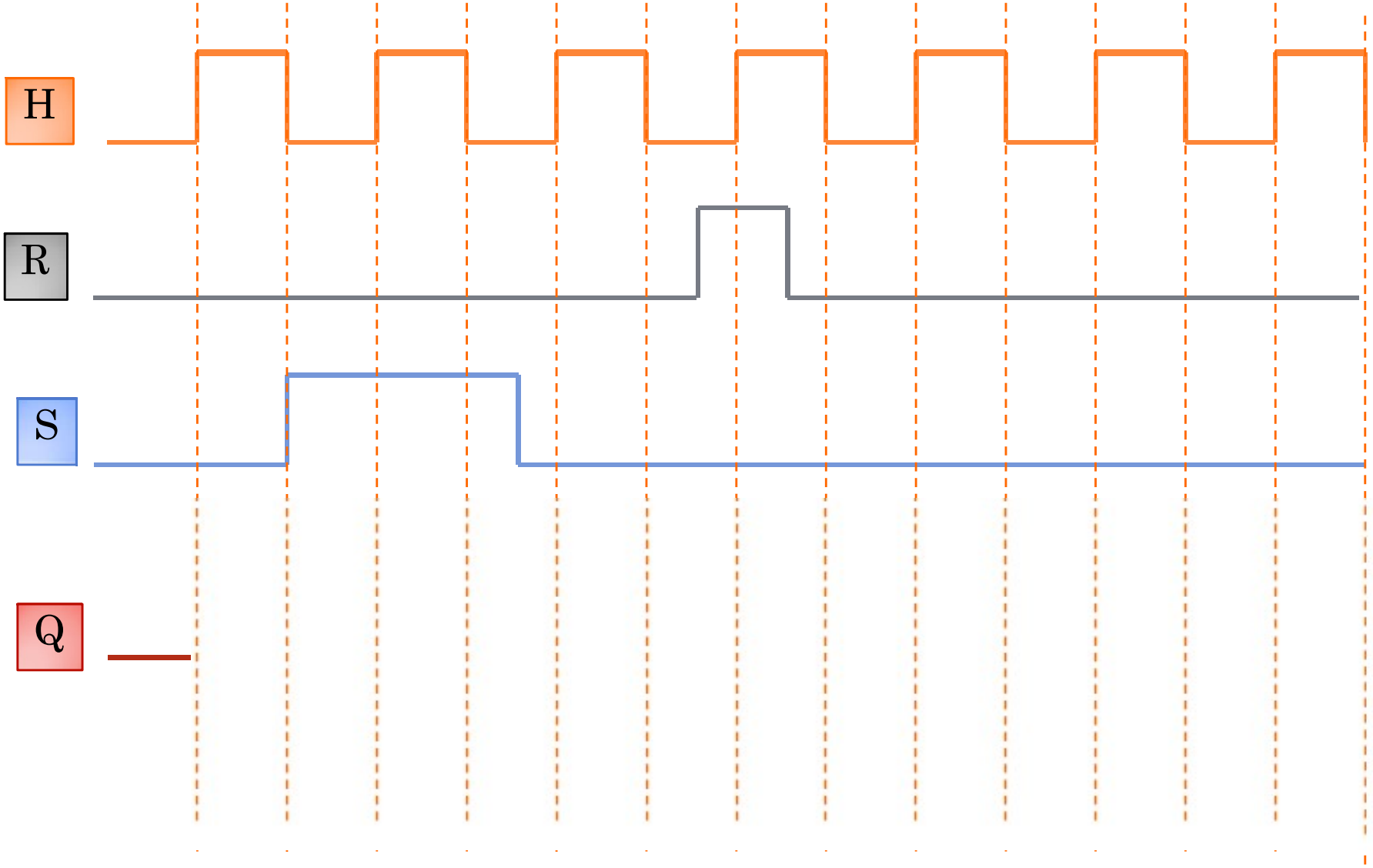
Types de synchronisations

Plusieurs façons de prendre en compte les changements de l'état des bascules:

- Niveau d'horloge haut : on prend en compte les entrées de bascule lorsque le niveau de l'horloge est haut (à « 1 »).
- Niveau d'horloge bas: en prend en compte les entrées de la bascule lorsque le niveau de l'horloge est bas (à « 0 »).
- Front d'horloge mentant.
- Front d'horloge descendant.

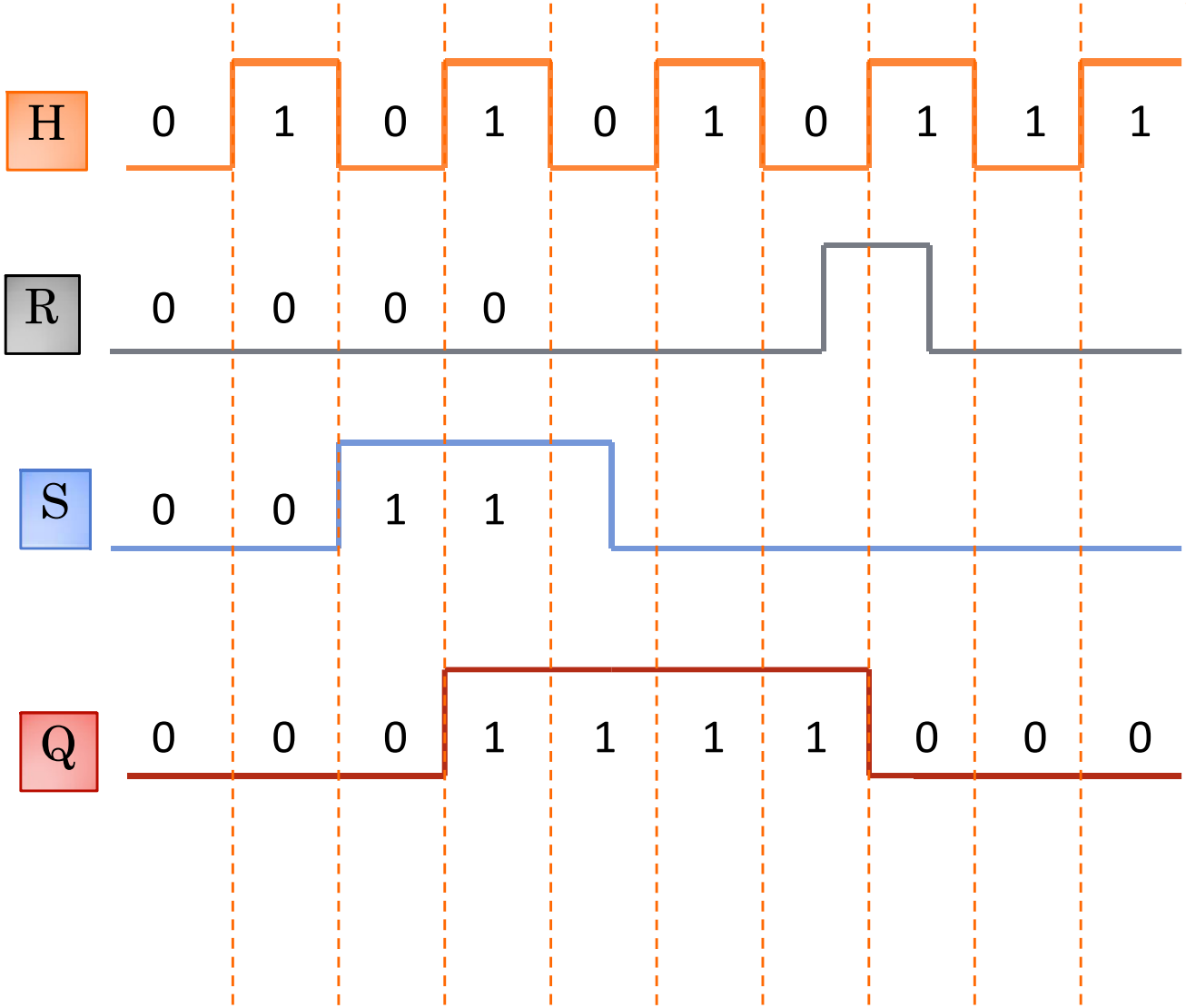
Bascule RSH

Chronogramme



Bascule RSH

Chronogramme



Bascule RSH

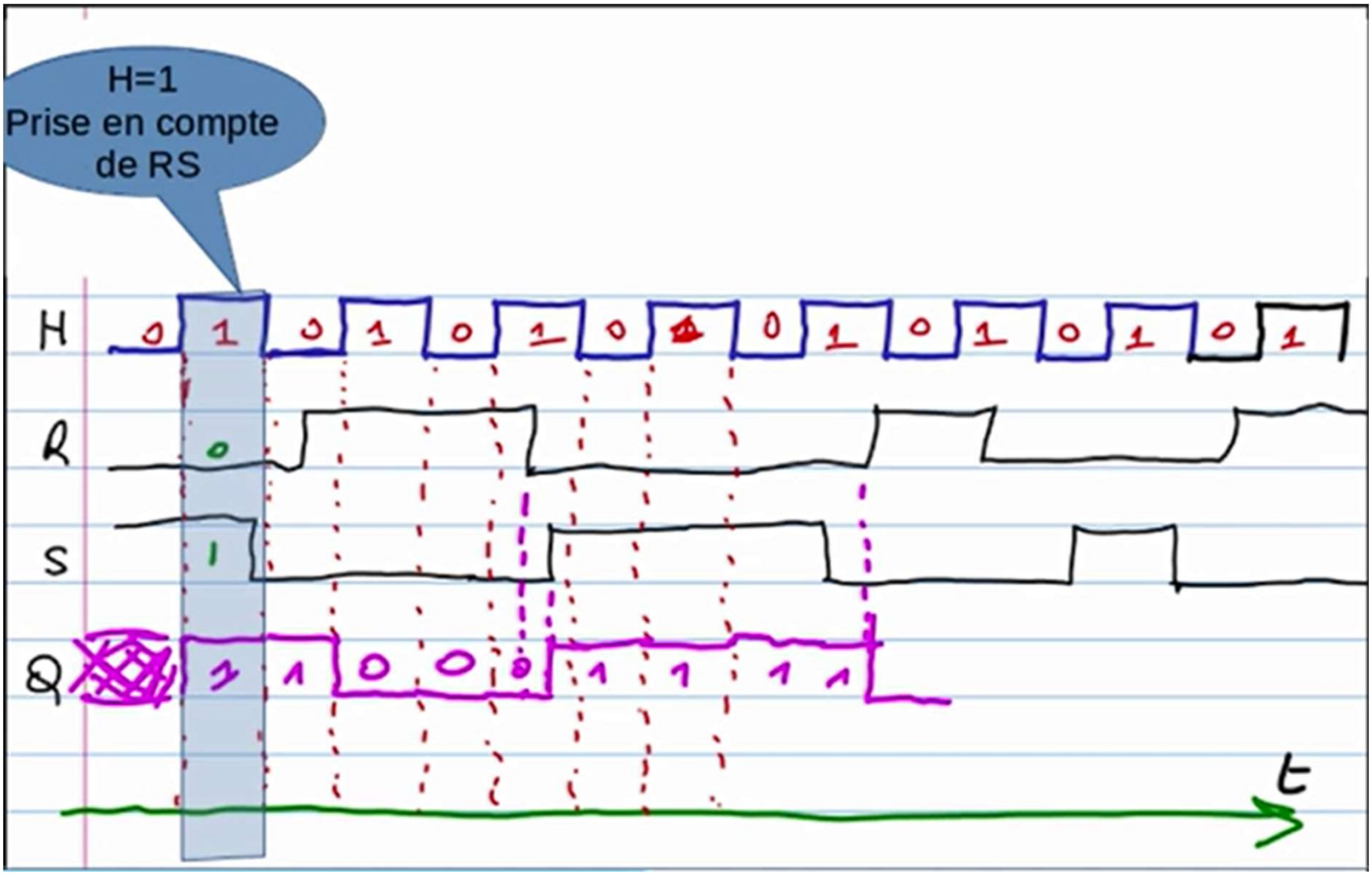
Exemple :Chronogramme

Bascule RSH

Exemple :Chronogramme

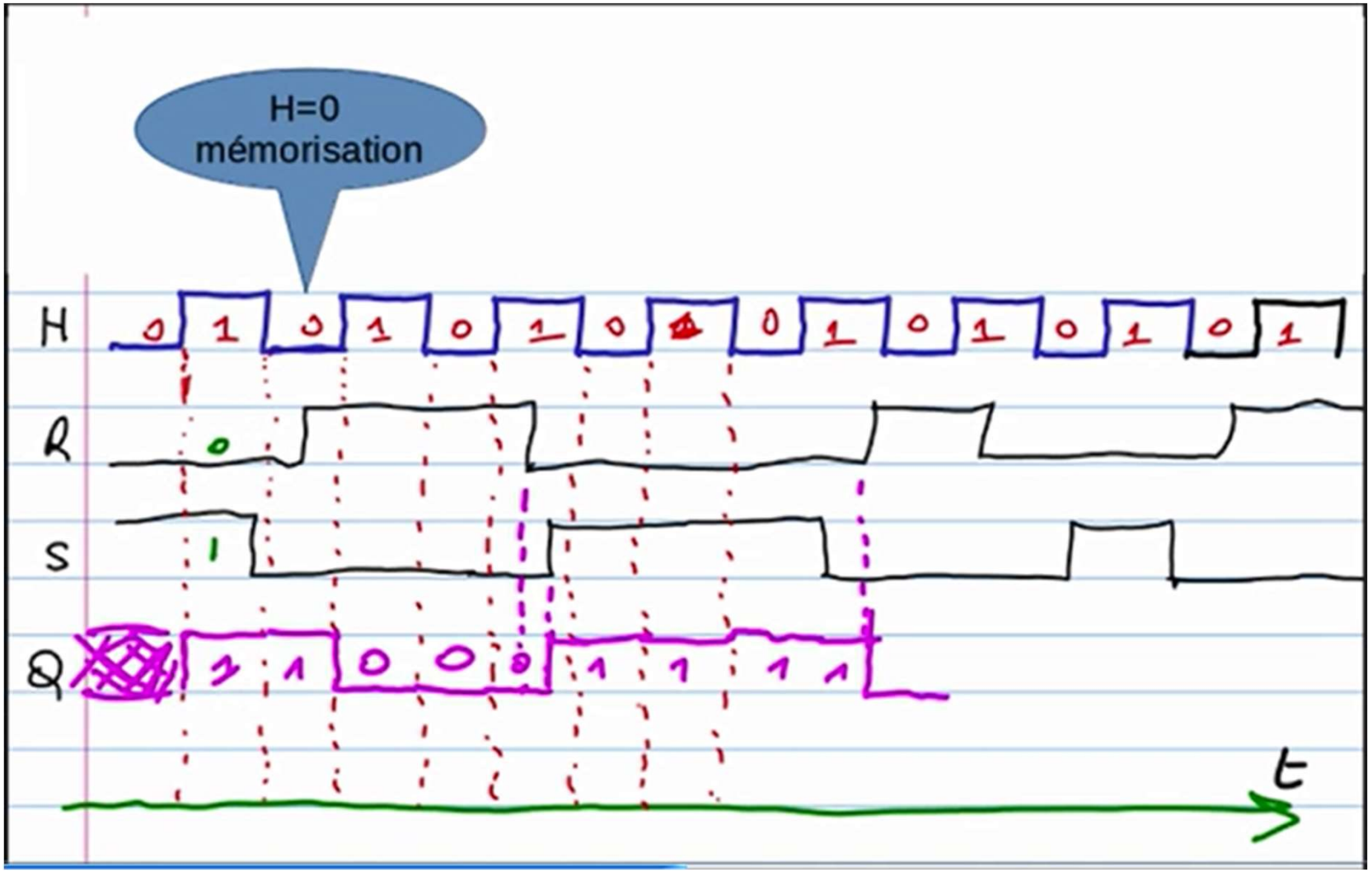
BASCULE RSH

Exemple :CHRONOGRAMME



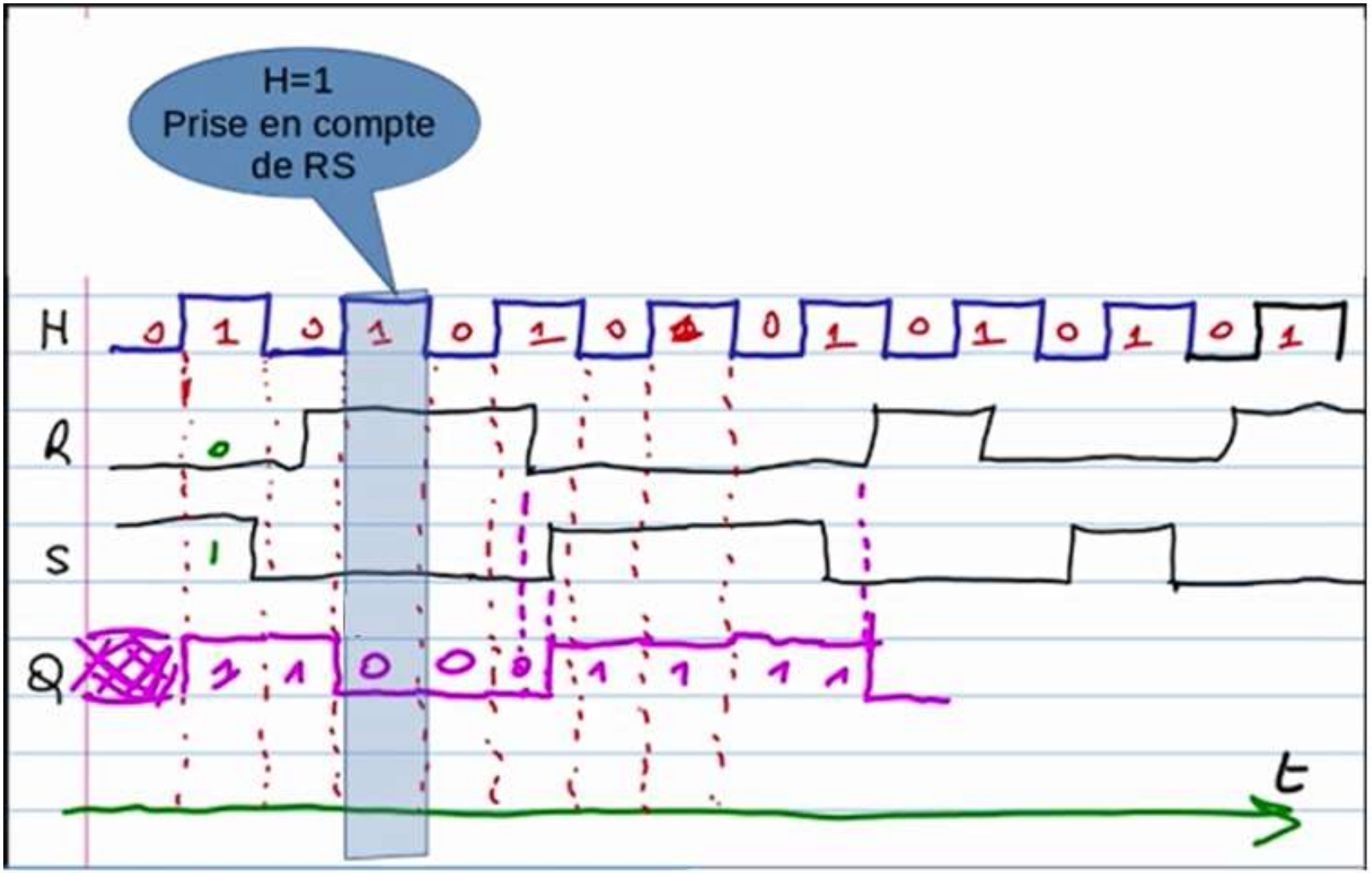
Bascule RSH

Exemple :Chronogramme



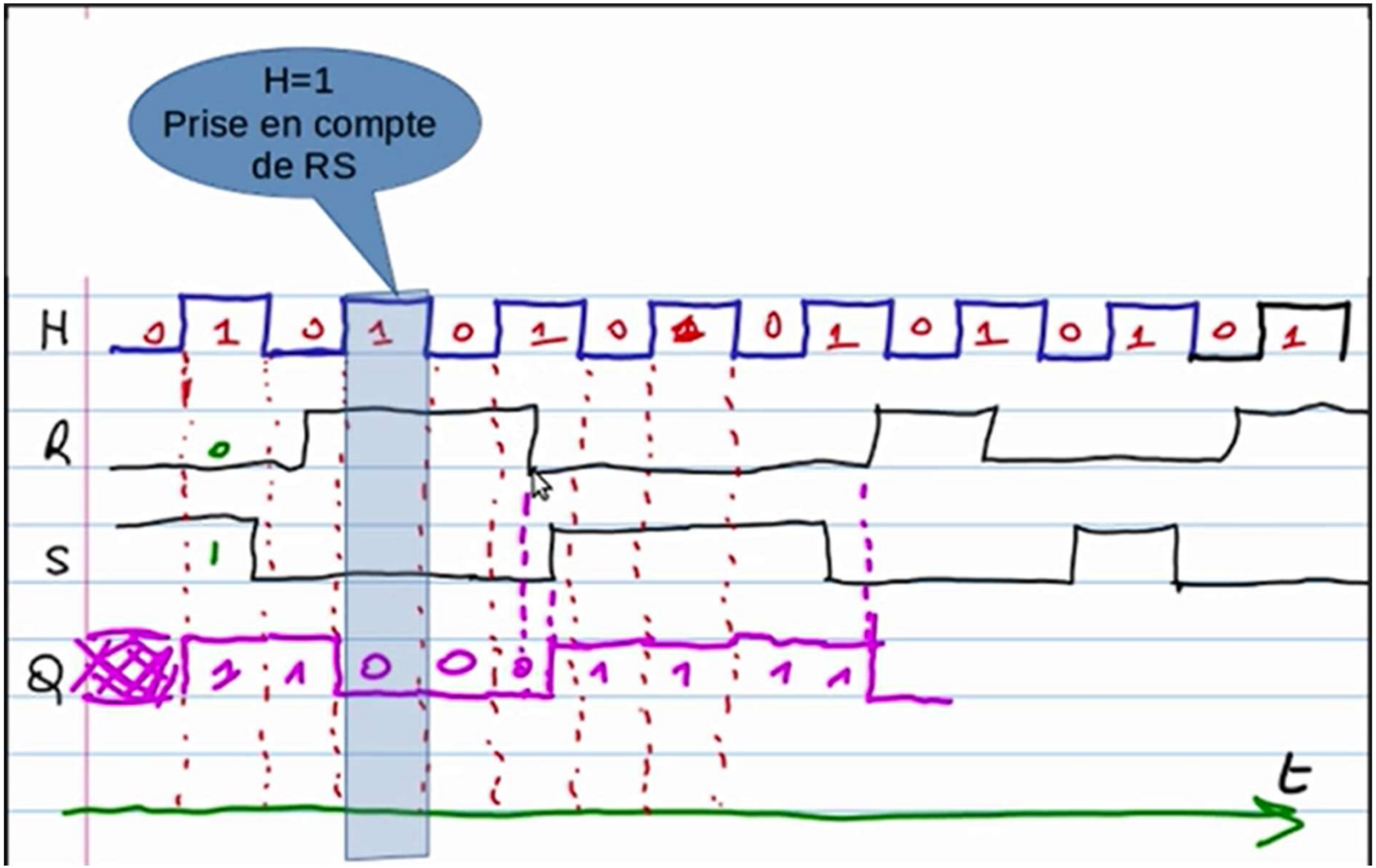
BASCULE RSH

Exemple :CHRONOGRAMME



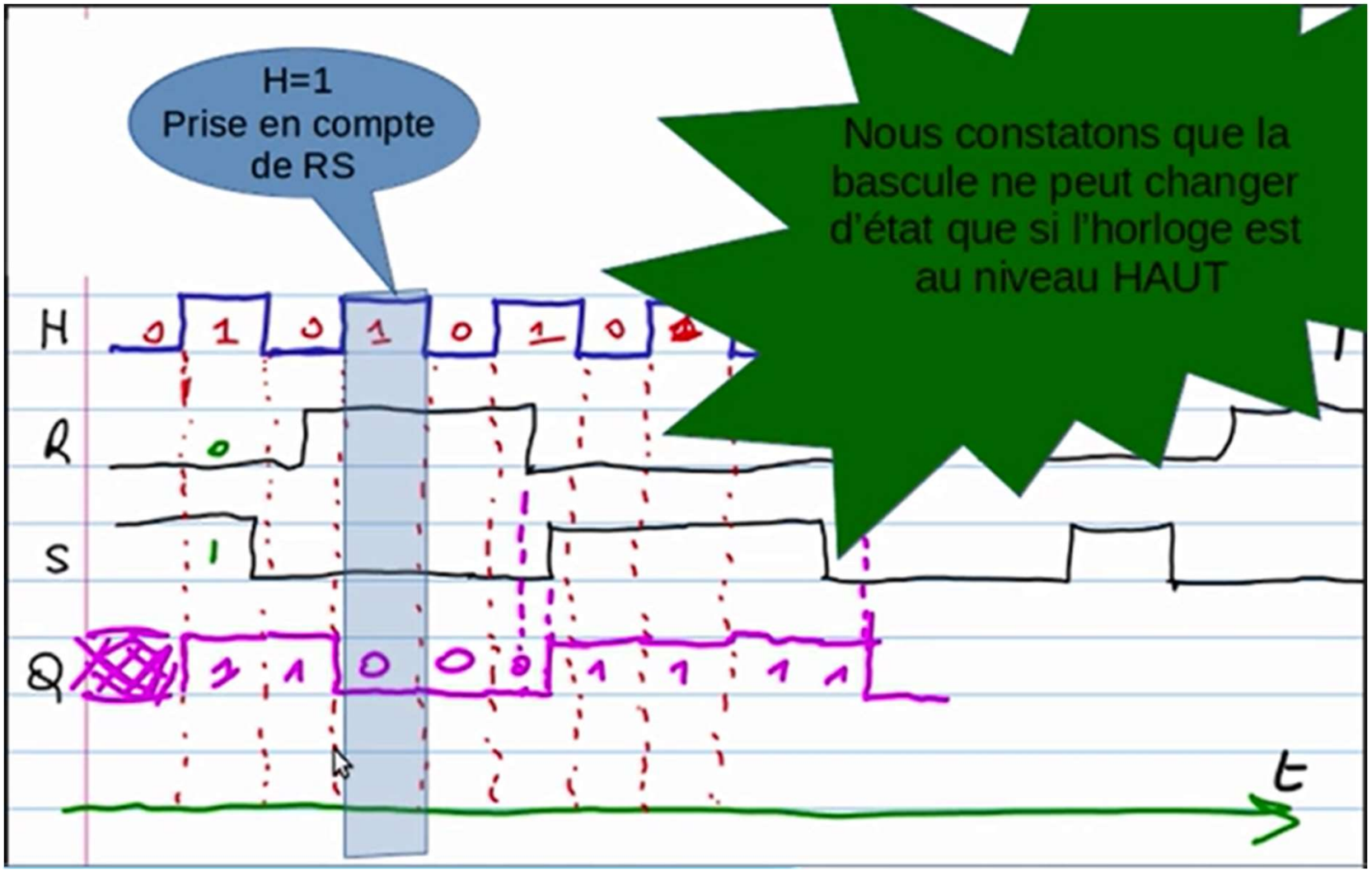
Bascule RSH

Exemple :Chronogramme



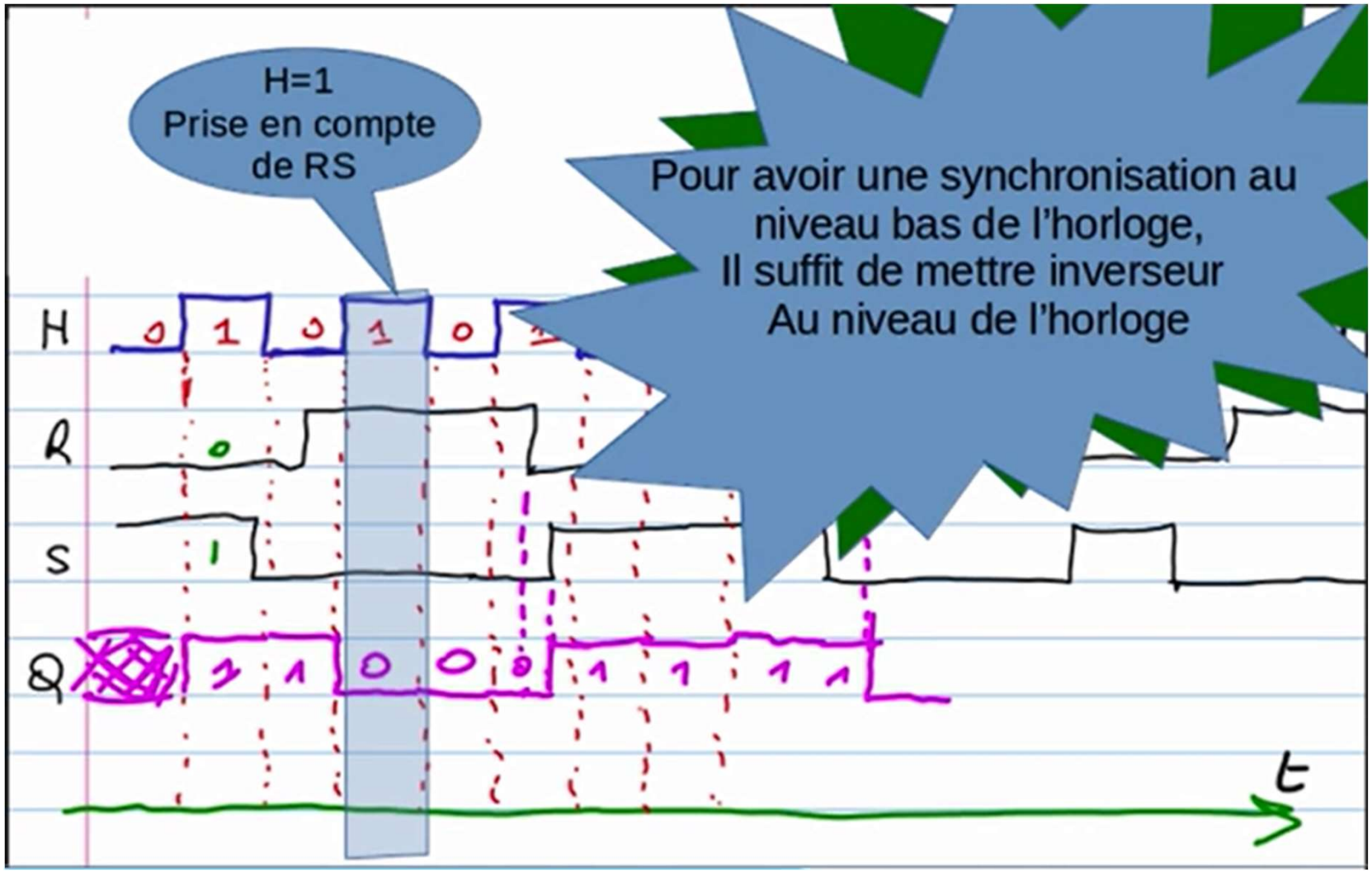
Bascule RSH

Exemple :Chronogramme

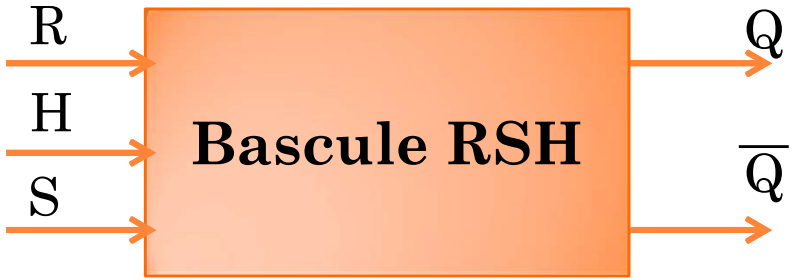


BASCULE RSH

Exemple :CHRONOGRAMME

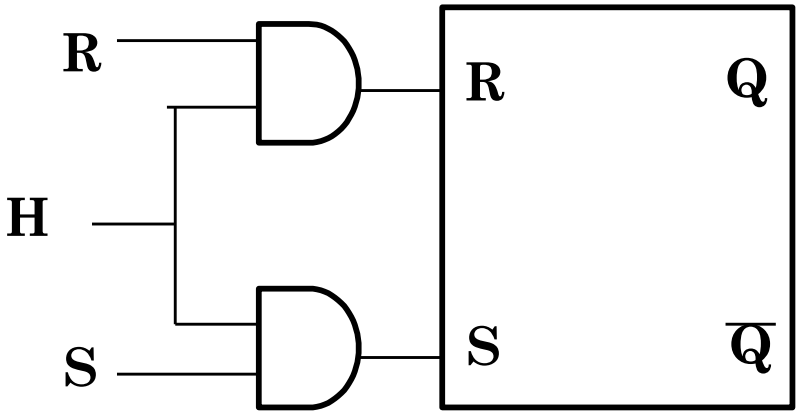


BASCULE RSH



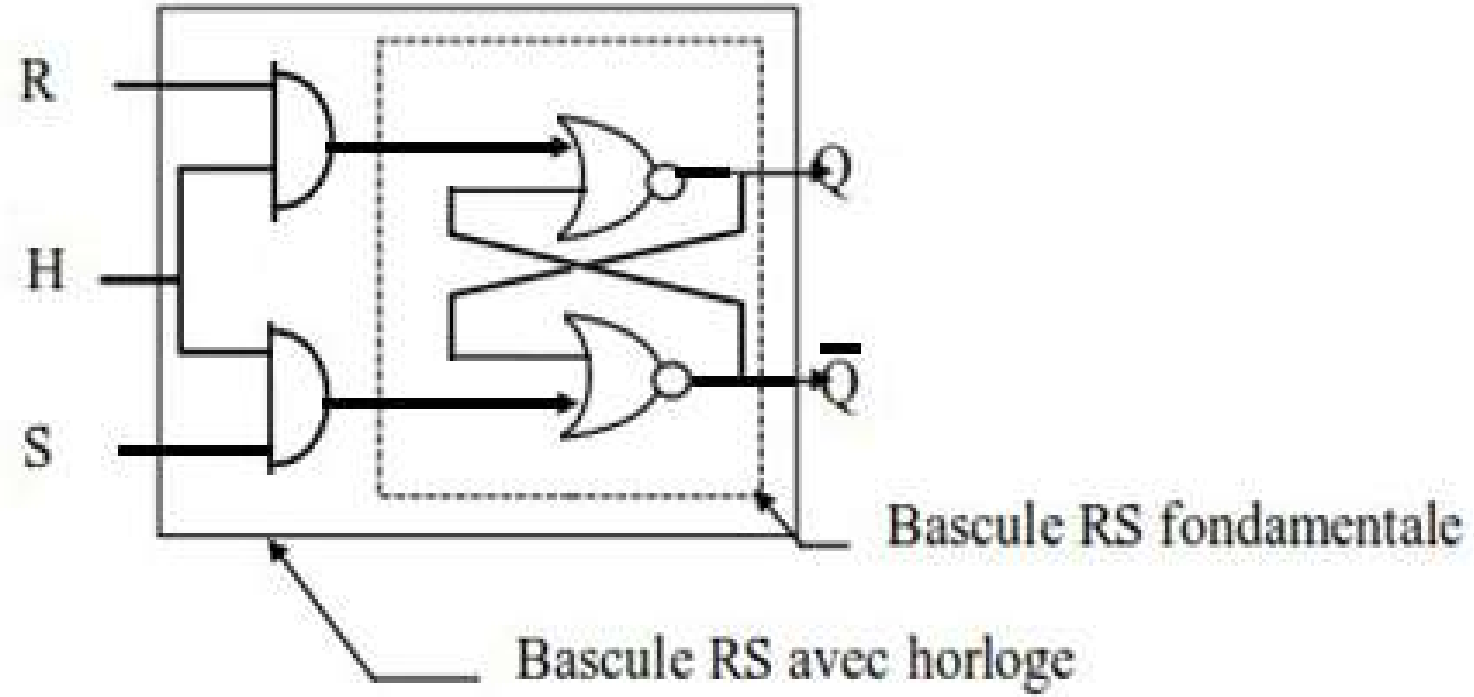
Exercice 2: Donner le circuit de cette bascule RSH en utilisant la bascule RS.

Solution: il suffit de faire un ET logique entre ses entrées et le signal d'horloge, pour imposer à ce que la prise en considération de ses entrées soit rythmée par le signal d'horloge.

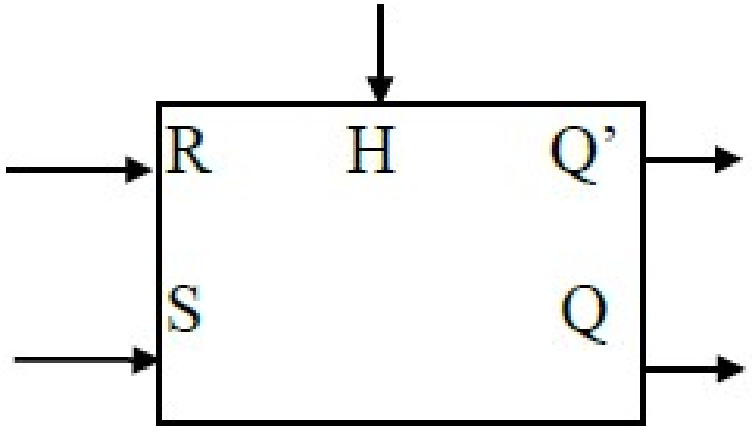


BASCULE RSH

- **Exercice 2:** Donner le circuit de cette bascule en utilisant la bascule RS.



Schématiquement,
on représente la bascule RS
comme suit:



Bascule D

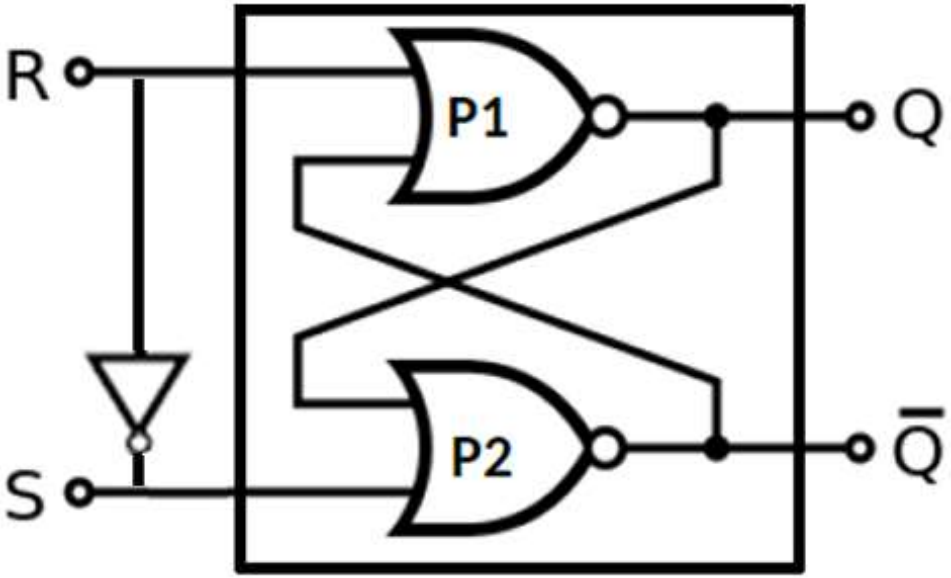
Bascule D

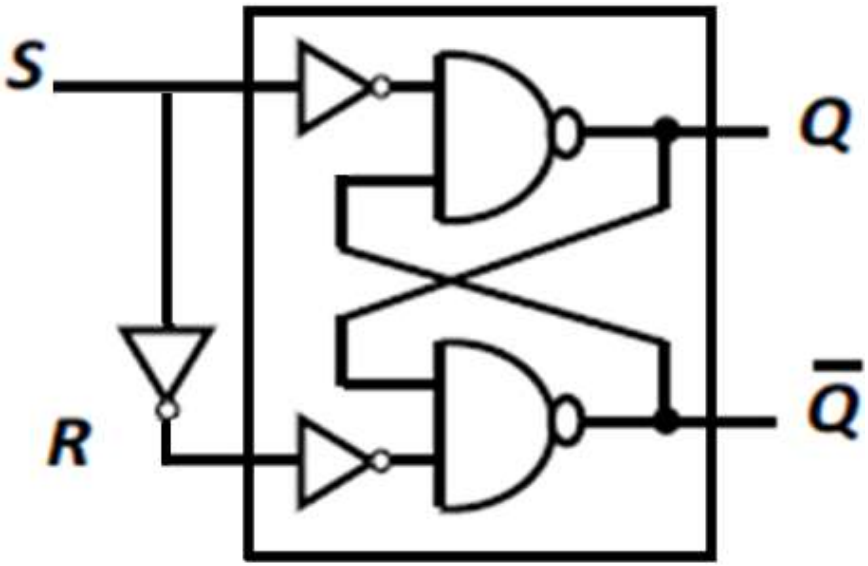
La bascule **D (Data** ou **Donnée)** est dérivée de la bascule **RSH** en ajoutant une porte inverseuse entre les entrées **Set** et **Reset** pour n'avoir qu'une seule entrée pour fixer le niveau logique à mémoriser. Avec cette bascule, il n'y a *plus de combinaison invalide* **S=R=1 (**Dans ces conditions l'état R=S=1qui donnait lieu à un fonctionnement ambigu ne peut plus exister). On distingue deux types de bascules **D :**

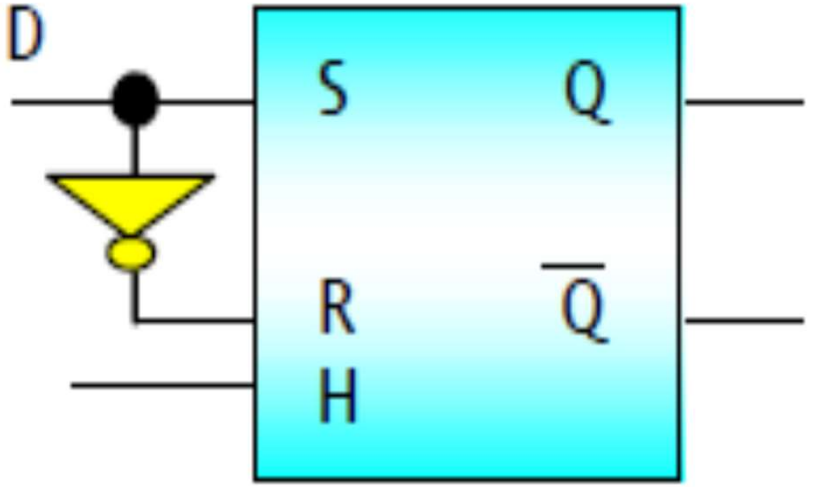
- Bascule D active sur niveau (Statique).
- Bascule D active sur front (Dynamique).

Bascule D

Bascule D







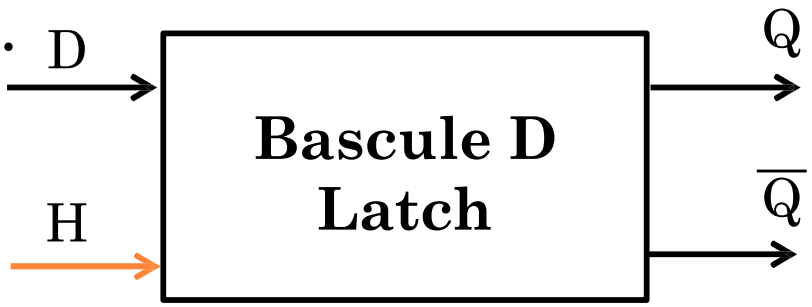
Bascule D Latch ou Bascule D à Verrouillage (Latch : Verrou)

La bascule D est une bascule RSH dont les entrées R et S sont complémentées afin d'éliminer l'état interdit (R=S=1) et de minimiser le nombre d'états de mémorisation.

Lorsque H est active et D=1, la sortie Q sera mise a 1 (set).
Lorsque H est active et D=0, la sortie Q sera remise a 0 (reset).
Lorsque H n'est pas active, les sorties restent dans l'état précèdent (mémorisation).

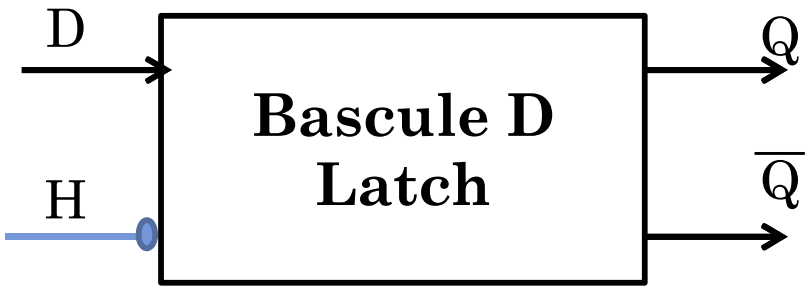
Bascule D Latch

- C'est une bascule synchrone sur niveau Haut ou niveau Bas.



Sur niveau Haut « 1 »

Si H = 1 alors Q+ = D



Sur niveau Bas « 0 »

Si H = 0 alors Q+ = D

H/$\overline{H}$	Q+
0	Q-
1	D

H/$\overline{H}$	D	Q+
0	0	Q-
0	1	Q-
1	0	0
1	1	1

Bascule D Latch ou Bascule D à Verrouillage (Latch : Verrou)

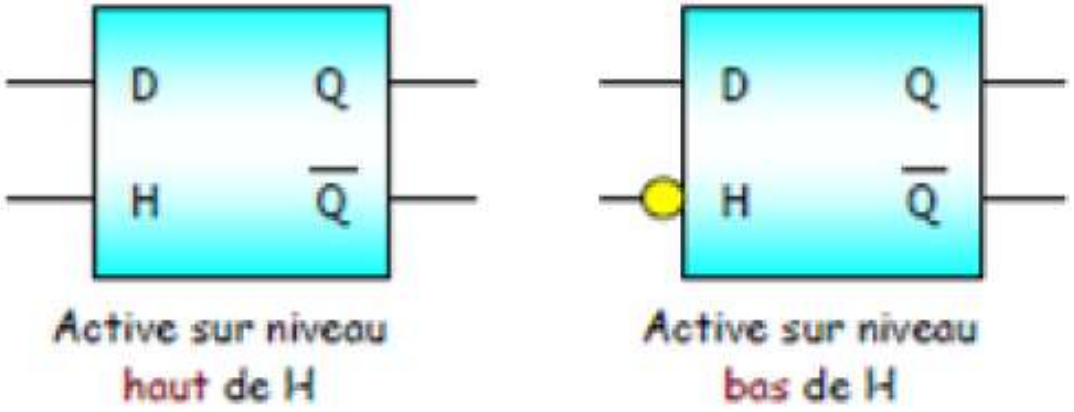
C'est une bascule **statique** synchrone sur le **niveau** d'horloge dont le fonctionnement est le suivant :

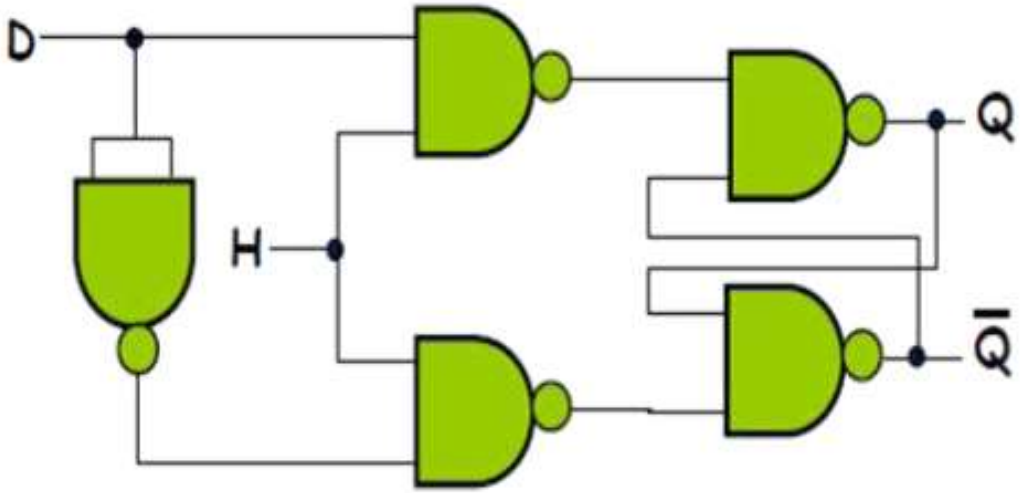
- La bascule est transparente tant que le signal d'horloge est au **niveau haut (ou niveau bas).**
  La sortie **Q** suit toutes les variations de l'entrée **D.**
  Le **verrou** est dit **transparent.**

- L'état de la sortie **Q** est verrouillé (mémoire) tant que le signal d'horloge est **au niveau bas (ou niveau haut).** La sortie **Q** conserve son état logique. Le **verrou** est dit **bloqué.**

Bascule D Latch ou Bascule D à Verrouillage (Latch : Verrou)

Symbole :





Bascule D Latch ou Bascule D à Verrouillage (Latch : Verrou)

- Lorsque H = 1, la sortie Q est identique à D (Q=D) ; on dit que la bascule est *transparente*.
- Lorsque H = 0, la sortie Q demeure à la dernière valeur de D qu'elle avait avant que H passe au niveau 0.

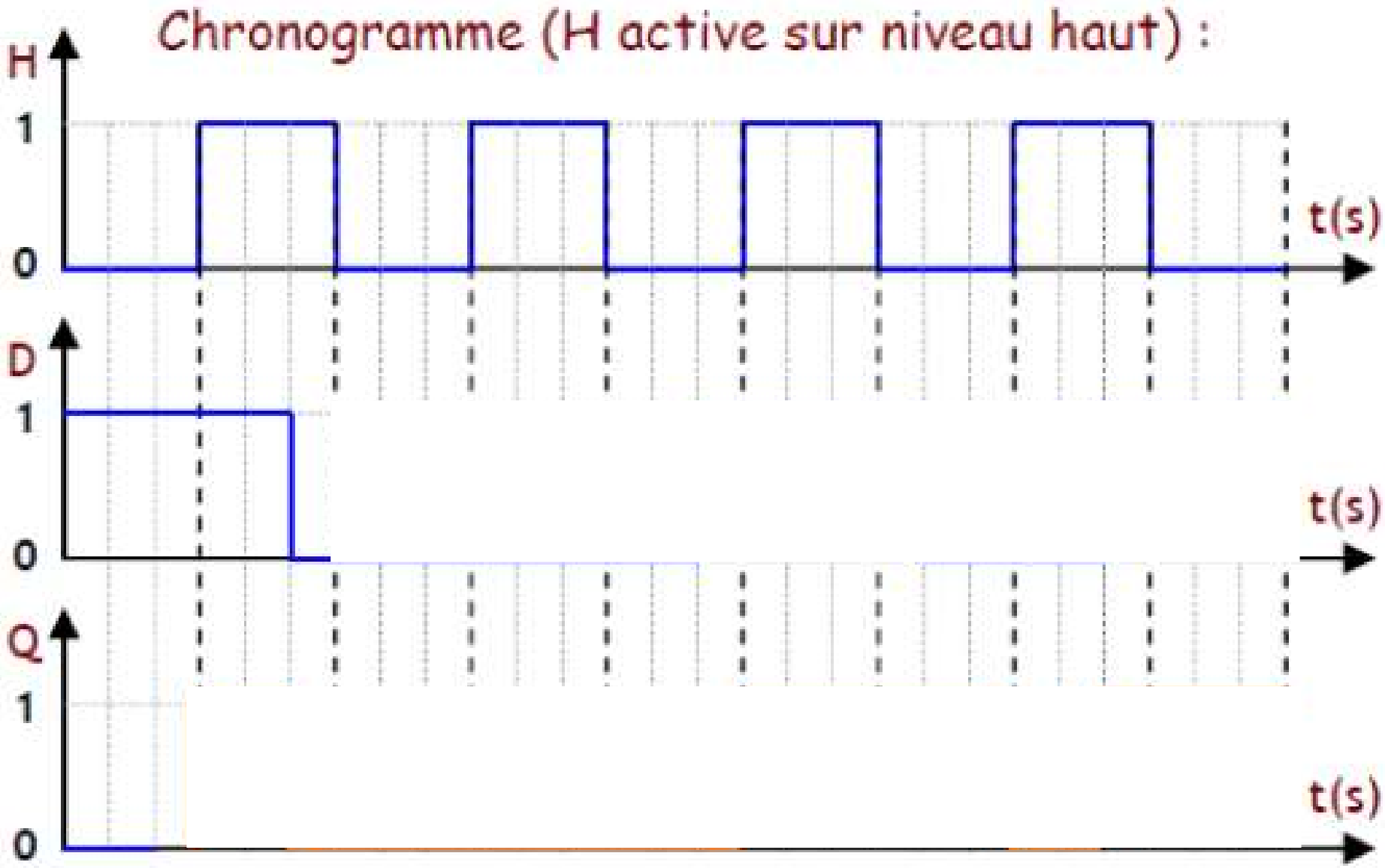
En d'autre terme, la sortie est *verrouillée "latched" sur D et ne change pas tant que* CLK reste au niveau bas, même si D change de valeur.

Le fonctionnement de la bascule D "latch" est résumé dans la table de vérité suivante:

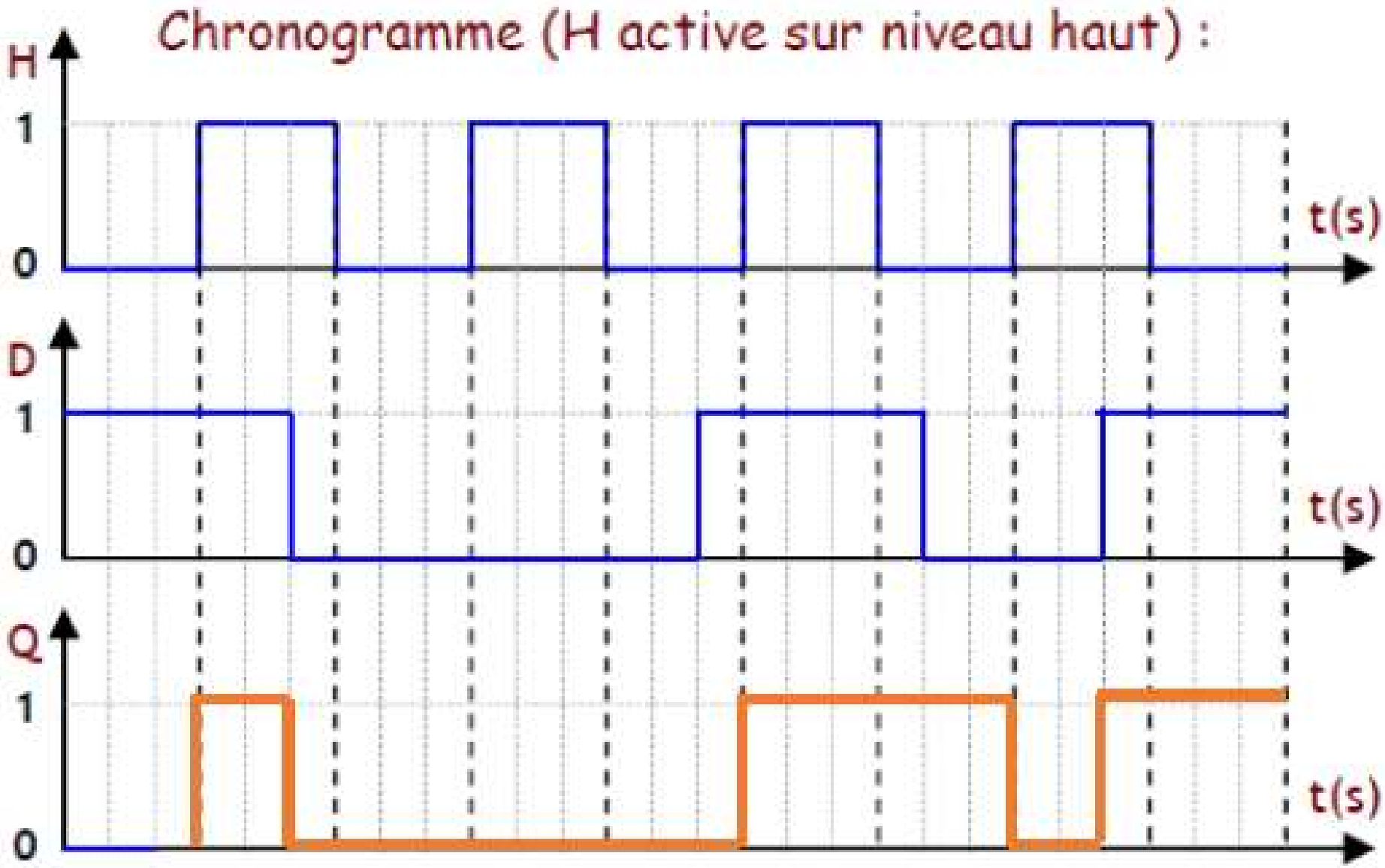
Bascule D Latch ou Bascule D à Verrouillage (Latch : Verrou)

Table de vérité			
H	D	Q	Commentaire
0	x	q	Mémorisation
1	0	0	La bascule recopie la valeur de D sur Q
1	1	1	

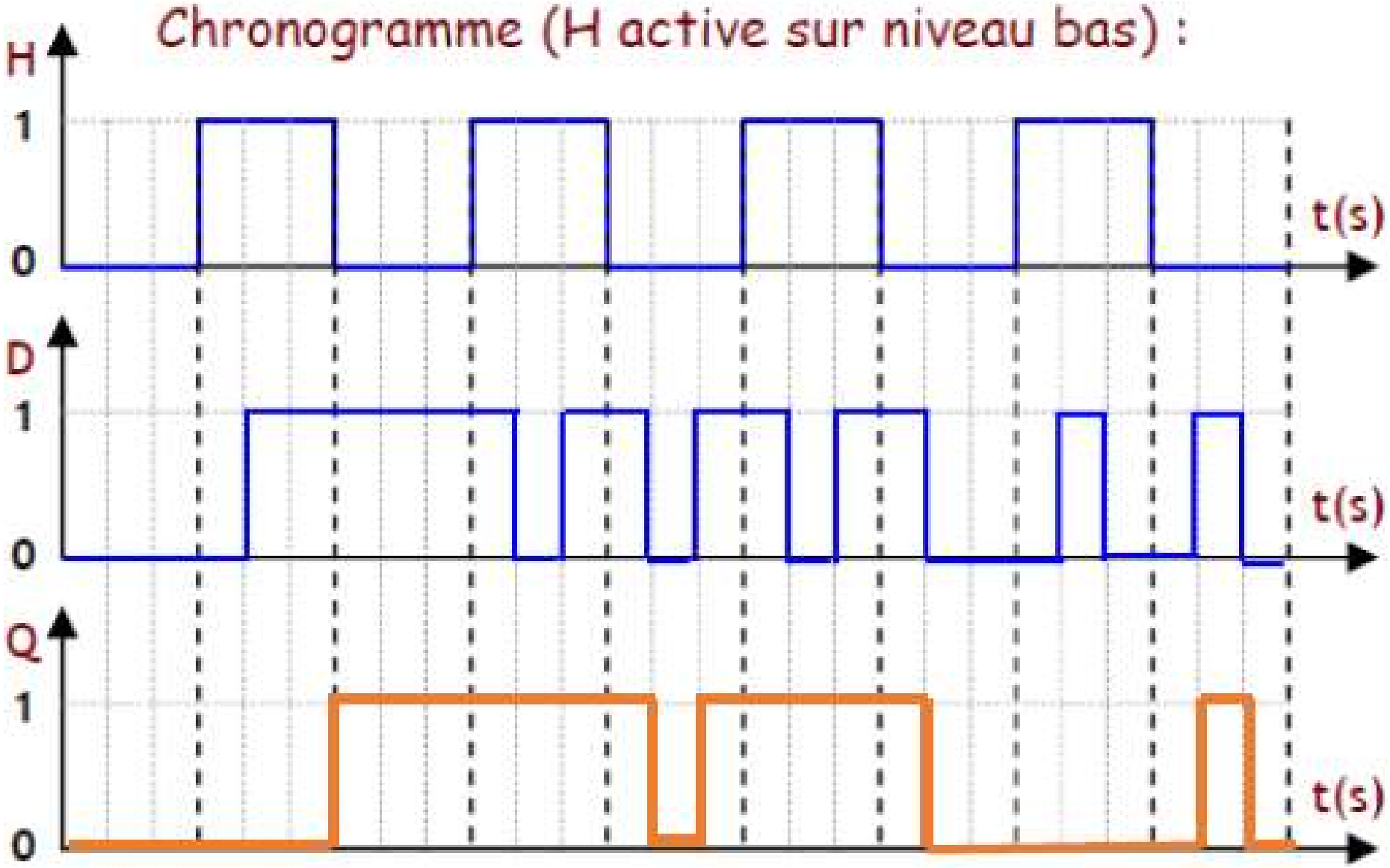
Bascule D Latch Chronogramme (H Active sur niveau haut)



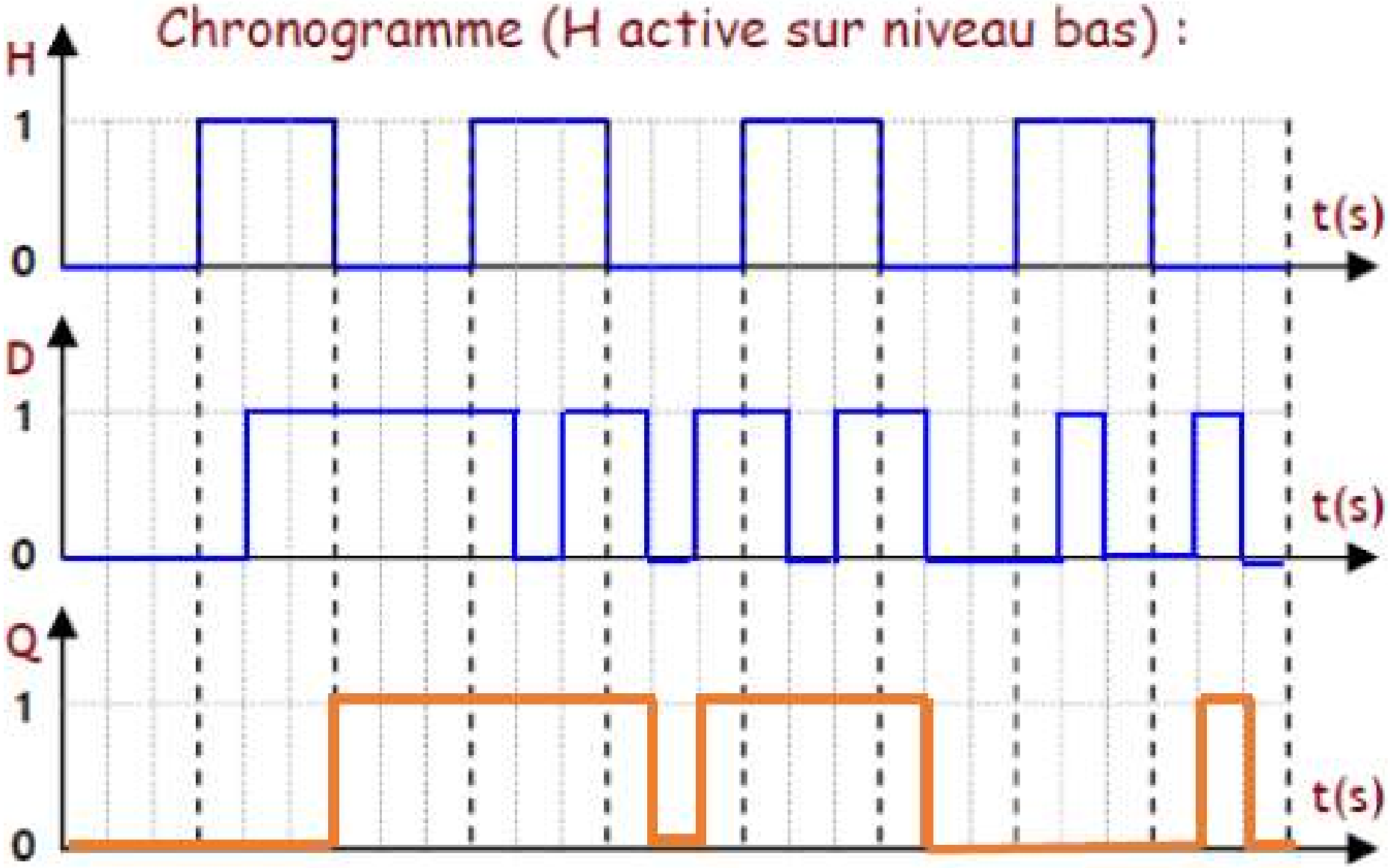
Bascule D Latch Chronogramme (H Active sur niveau haut)



Bascule D Latch Chronogramme
(H Active sur niveau bas)

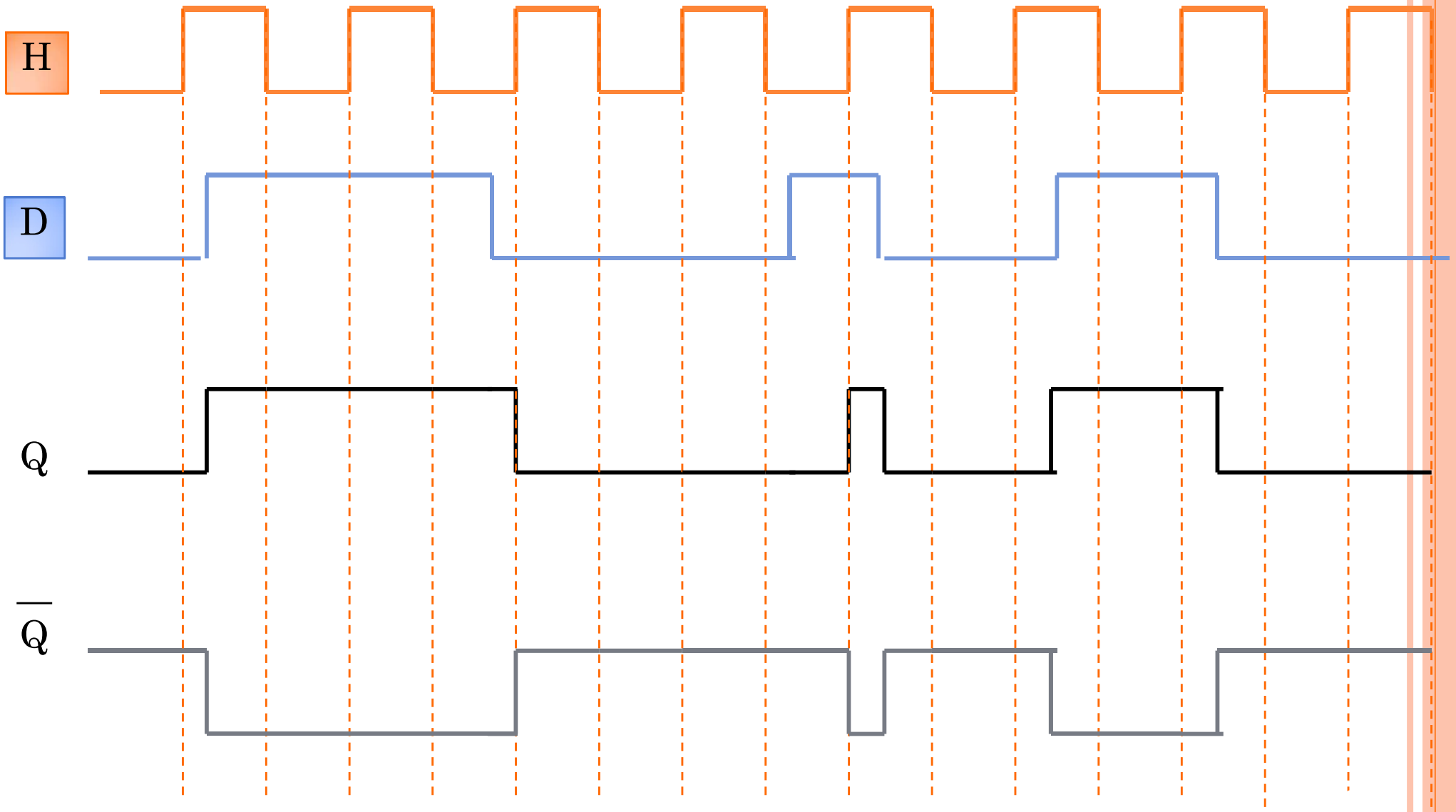


Bascule D Latch Chronogramme
(H Active sur niveau bas)



Bascule D Latch

Chronogramme (niveau haut)



BASCULE D LATCH

- **Exercice 3:** Transformer une bascule RSH pour qu'elle agisse comme une bascule D Latch (niveau haut).

H	D	Q+
0	0	Q-
0	1	Q-
1	0	0
1	1	1

H	R	S	Q+
0	**X**	**X**	**Q-**
1	0	0	Q-
1	0	1	1
1	1	0	0
1	1	1	X

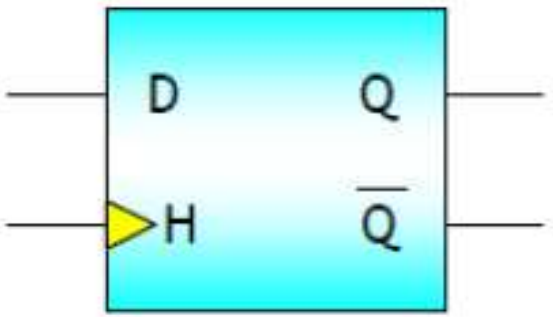
$$H_D = H_{RSH},\ R = \overline{D}\ ;\ S = D$$

Une Bascule D flip-flop ou à Commande par front.

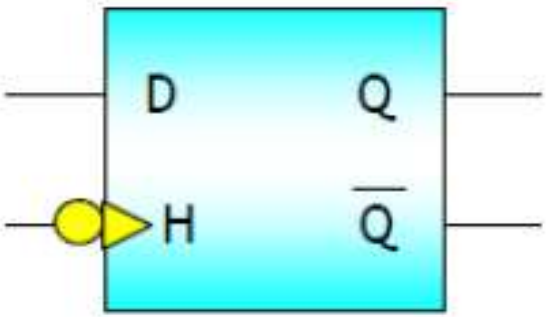
C’est une bascule **dynamique** synchrone sur le **front** d’horloge dont le fonctionnement est le suivant :

- En présence du front actif de l’horloge, la bascule recopie l’état logique de l’entrée **D** sur la sortie **Q**.
- En absence du front actif de l’horloge, la bascule mémorise son état logique de la sortie **Q**.

Symbole :



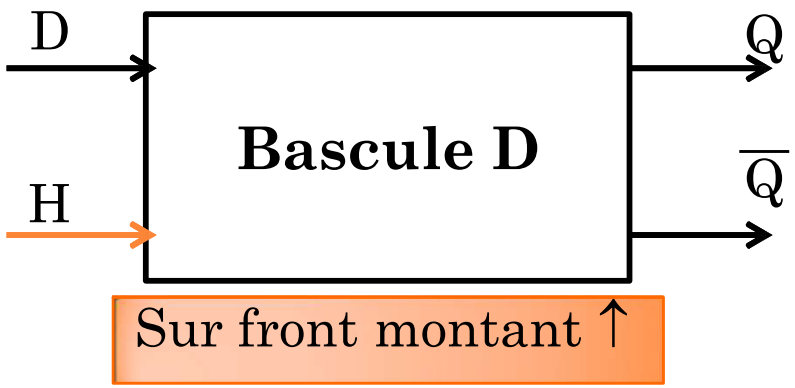
Active sur ↑ de H

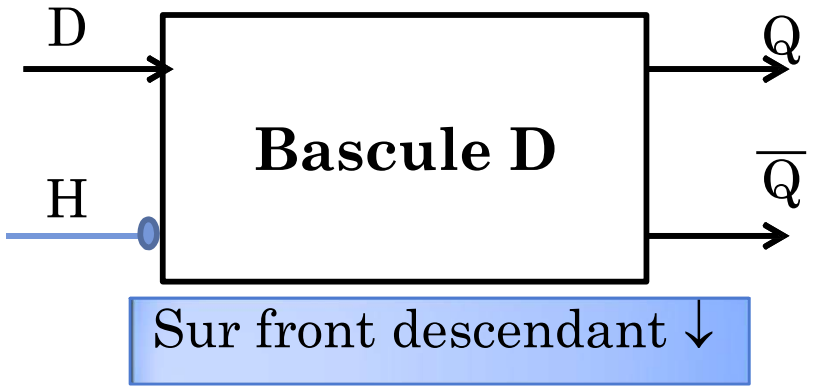


Active sur ↓ de H

BASCULE D

- C'est une bascule synchronisée sur front montant ou descendant

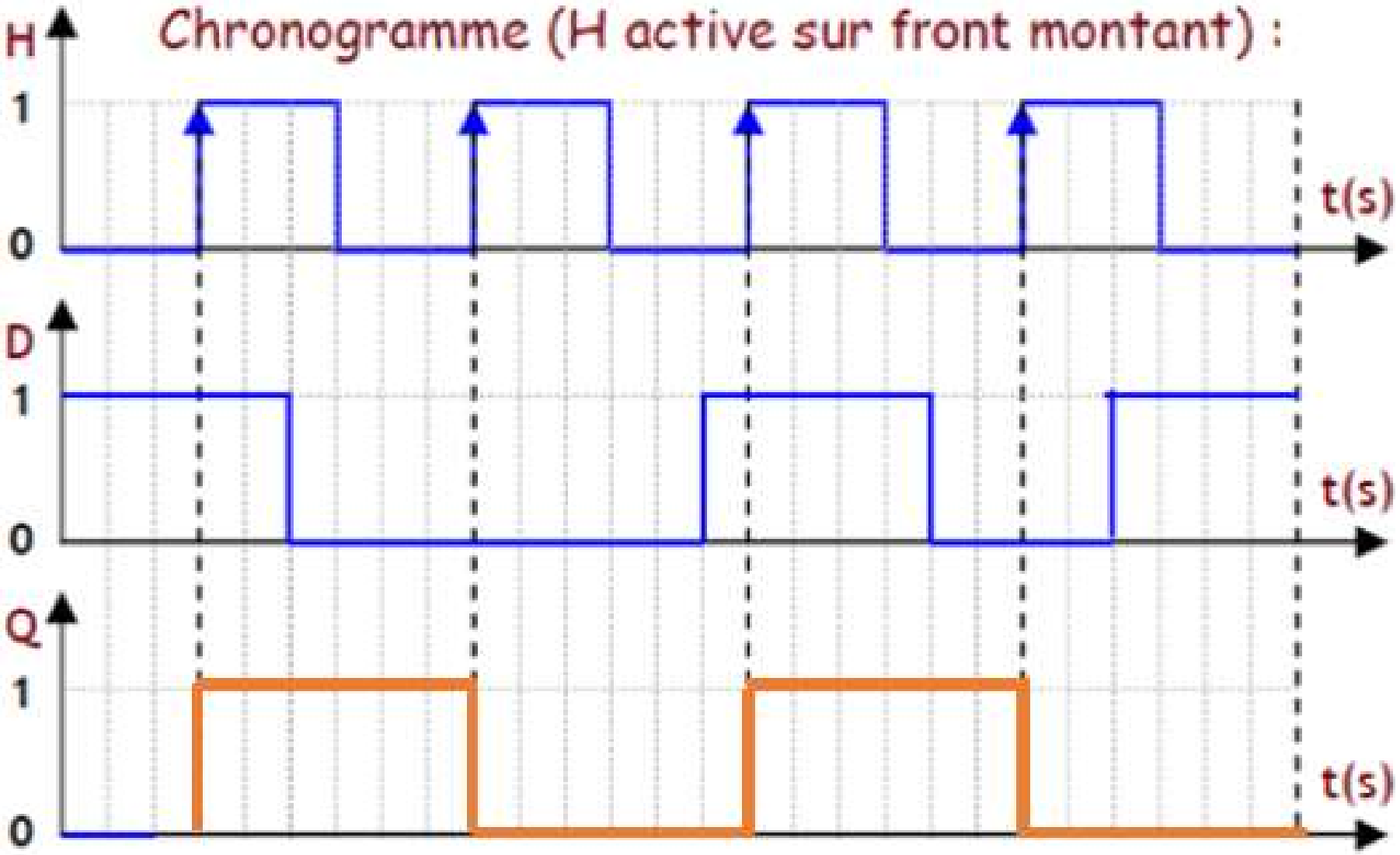




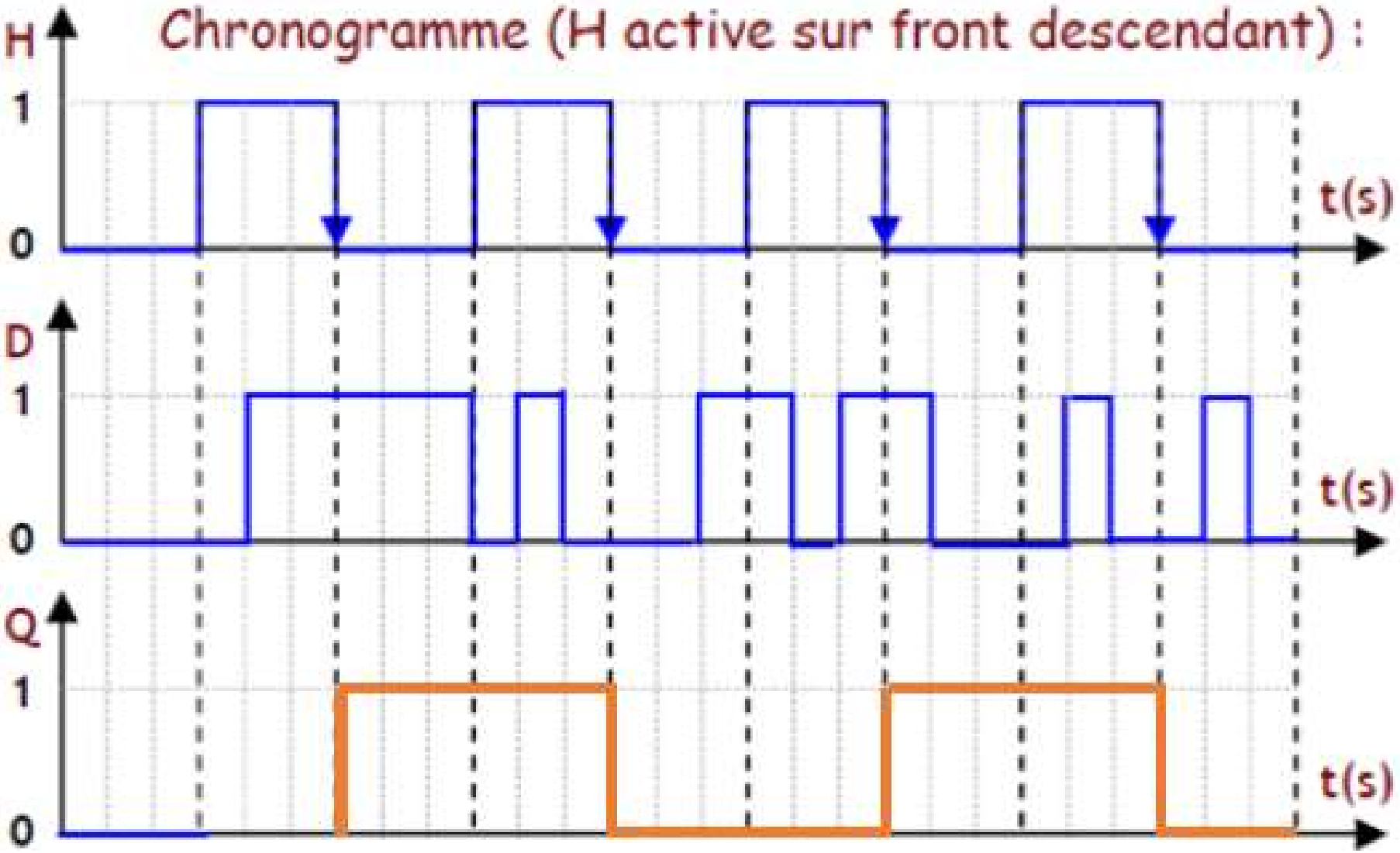
H	Q+
0/1/↓	Q-
↑	D

H	D	Q+
0/1/↓	0	Q-
0/1/↓	1	Q-
↑	0	0
↑	1	1

Une Bascule D flip-flop ou à Commande par front montant.

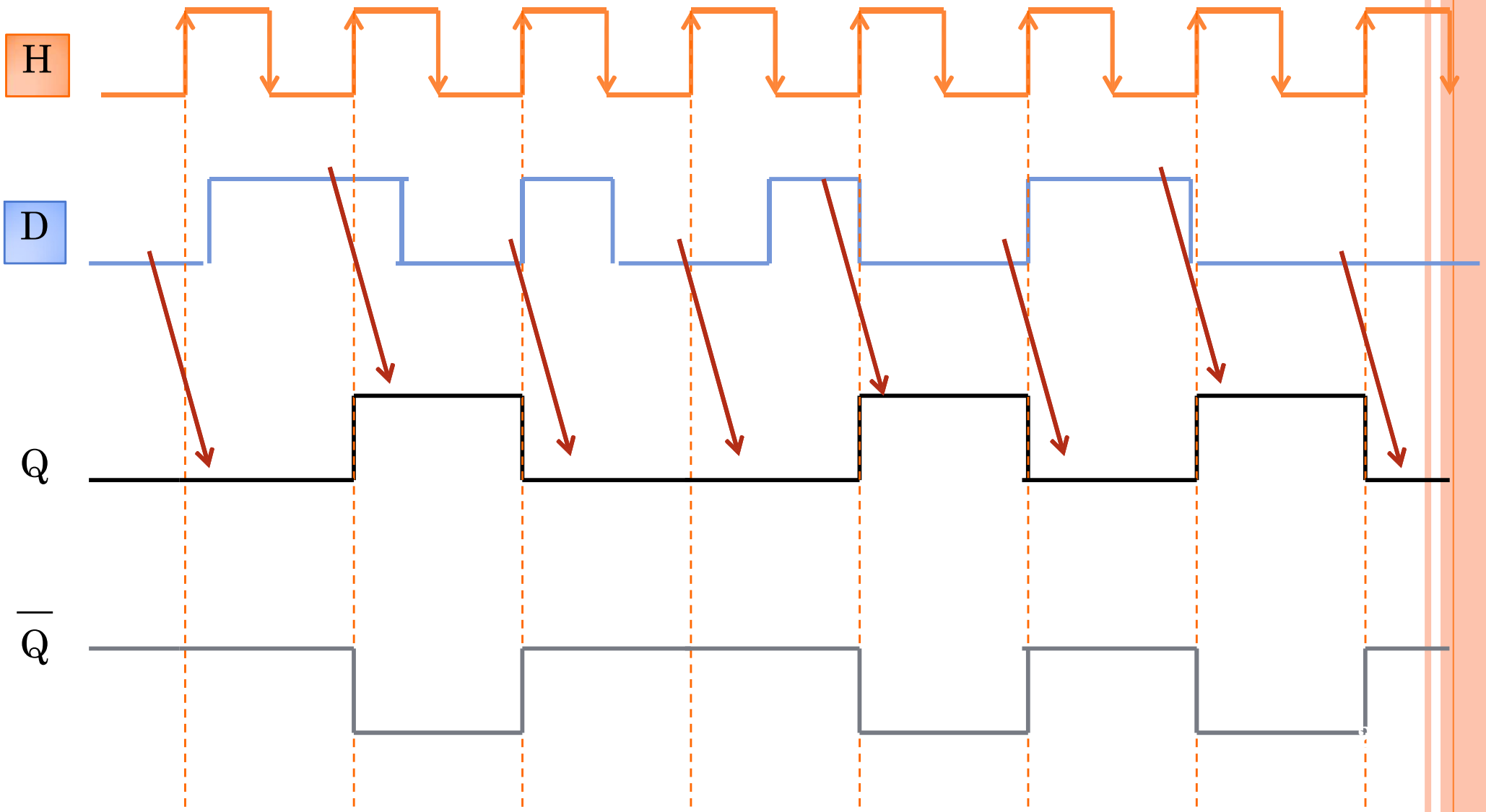


Une Bascule D flip-flop ou à Commande par front déscendant.



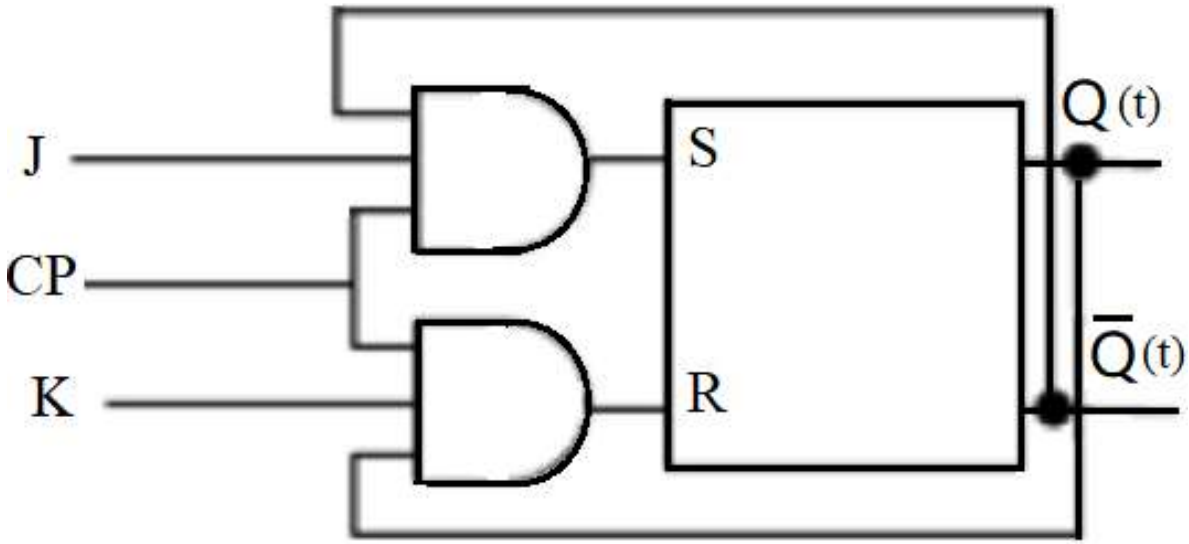
Bascule D

Chronogramme (Front Montant)



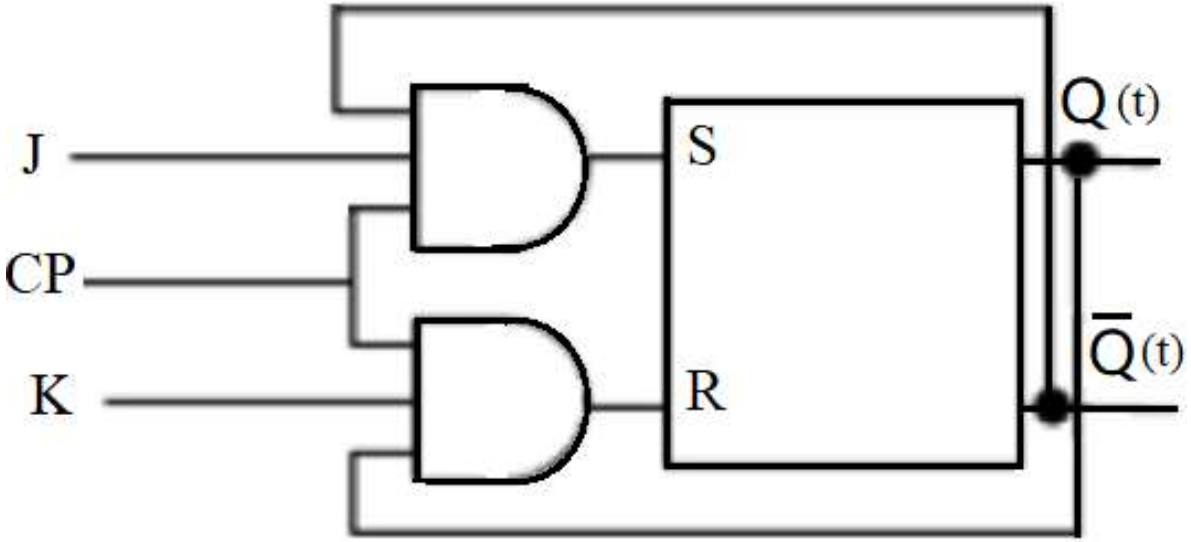
Bascule JK

La bascule **JK** est un perfectionnement de la bascule RS. Le cas indéterminé de cette dernière (S et R égaux à 1) *est défini dans la bascule JK, par une inversion des* états des sorties en asservissant (attachant) les entrées R et S aux sorties : $S=J\bar{q}$ et $R=Kq$



Bascule JK

La bascule **JK** permet donc *de lever* ***l'ambiguïté*** *qui existe pour* la combinaison **S=R=1** de la bascule **RSH.**



Equation des sorties

Q \ JK	00	01	11	10
0	0	0	1	1
1	1	0	0	1

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

Bascule JK

La bascule **JK** permet donc *de lever* ***l'ambiguïté*** *qui existe pour* la combinaison **S=R=1** de la bascule **RSH.**

J	K	Q(t)	S	R	Q(t+1)	Comment	Commentaire
0	0	0	0	0	Q(t)	Q(t)	Pas de changement
0	0	1	0	0	Q(t)		
0	1	0	0	0	Q(t)=0	0	Mise à zéro
0	1	1	0	1	0		
1	0	0	1	0	1	1	Mise à 1
1	0	1	0	0	Q(t)=1		
1	1	0	1	0	1	$\overline{Q}$ (t)	État inversé
1	1	1	0	1	0		

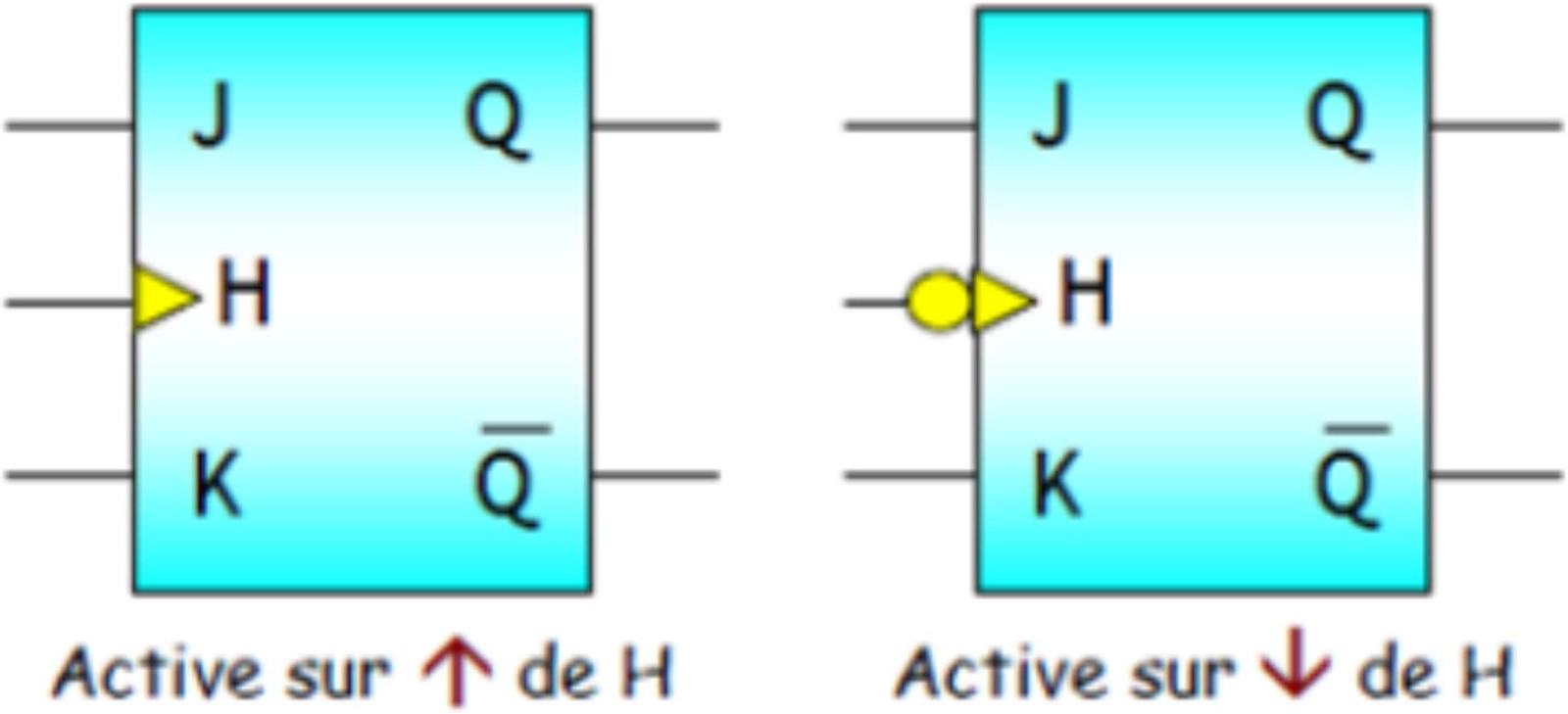
Bascule JK

La bascule **JK** est une bascule synchrone **dynamique** possédant deux entrées commandant l'état de la bascule, et une entrée H de synchronisation :

- L'entrée de l'enclenchement **J (Jump, saut à un)** joue le rôle de l'entrée **S** de la bascule **RSH.**
- L'entrée de déclenchement **K (Kill, mise à 0)** joue le rôle de l'entrée **R** de la bascule **RSH.**
- En absence du signal d'horloge, la bascule conserve l'état précédant de la sortie **Q (mémorisation).**
- Pour la combinaison **J=K=0,** la bascule mémorise l'état de la sortie **Q** à chaque front actif d'horloge (**front** montant ou descendant).

Bascule JK

Symbole :



Bascule JK

- Lorsque $J = \overline{K}$, la sortie **Q** recopie l'état de l'entrée **J** à chaque front actif d'horloge.

- Pour la combinaison **JK=10 (J=1 et K=0)**, La sortie **Q** est mise à **1** à chaque front actif d'horloge.

- Pour la combinaison **JK=01 (J=0 et K=1)**, La sortie **Q** est mise à **0** à chaque front actif d'horloge.

- A l'action simultanée sur **J** et **K (J=K=1)**, la bascule change d'état à chaque front actif d'horloge c'est le mode de basculement.

Bascule JK synchrone à front montant d'horloge

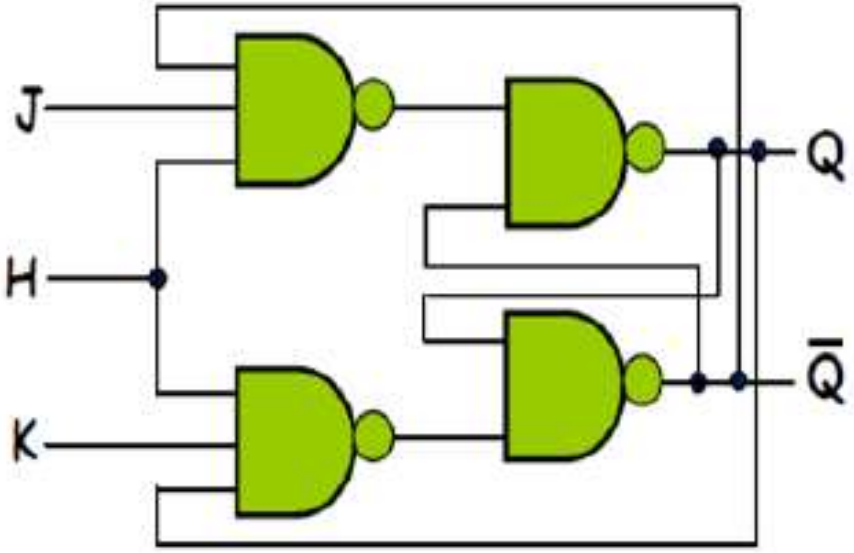
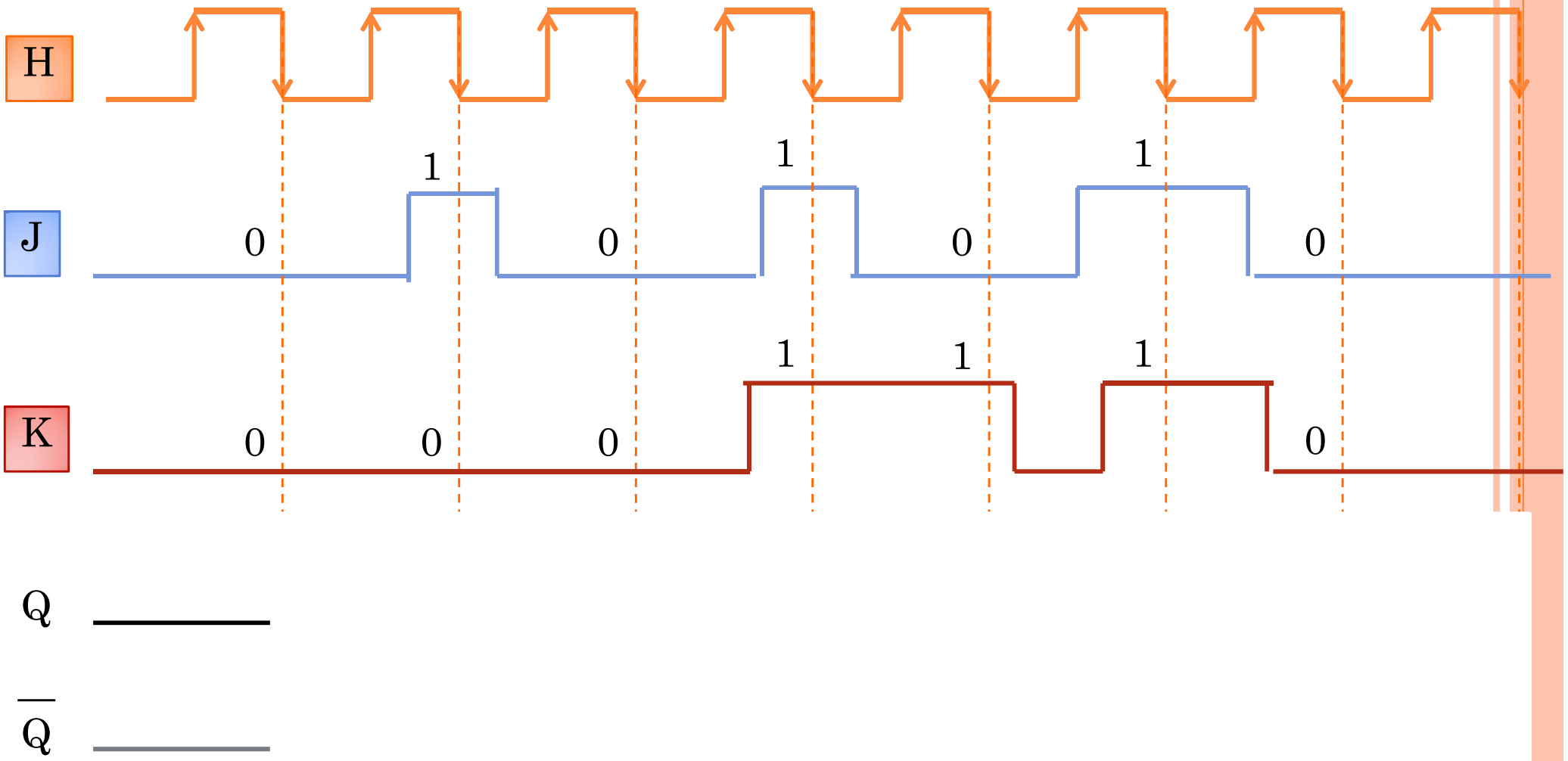


Table de vérité				
H	J	K	Q	Commentaire
↑	0	0	q	Mémorisation
↑	0	1	0	Mise à 0 de la sortie Q
↑	1	0	1	Mise à 1 de la sortie Q
↑	1	1	$\bar{q}$	Basculement de l'état de sortie

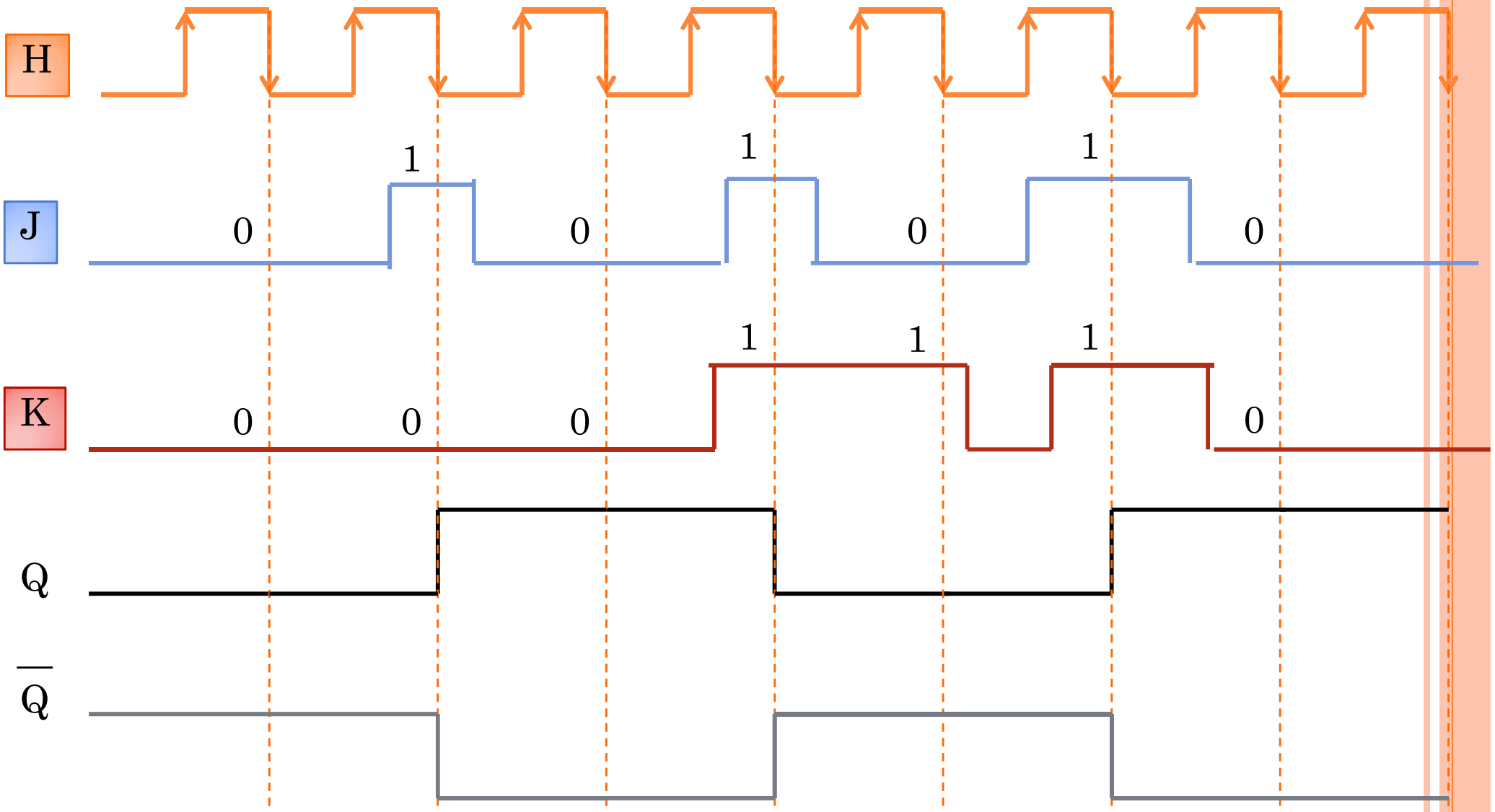
Bascule JK

Chronogramme (Front Descendant)



Bascule JK

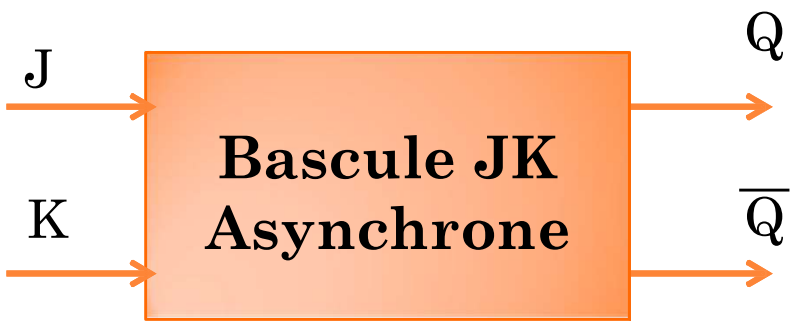
Chronogramme (Front Descendant)



BASCULE JK

ASYNCHRONE

- C'est une bascule variante de RS où on prend en compte le cas ou R=S=1

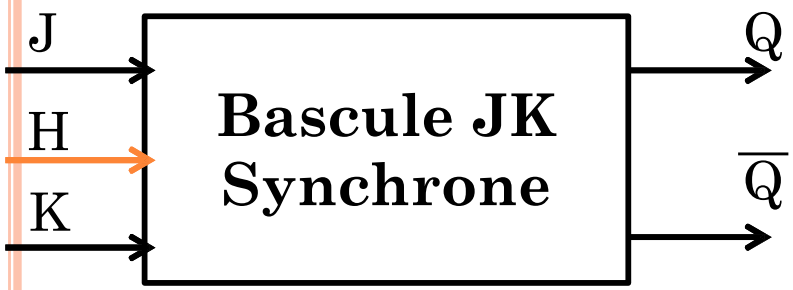


J	K	Q+	
0	0	Q-	État mémoire
0	1	**0**	Remise à 0
1	0	**1**	Remise à 1
1	1	$\overline{Q}$-	Basculement

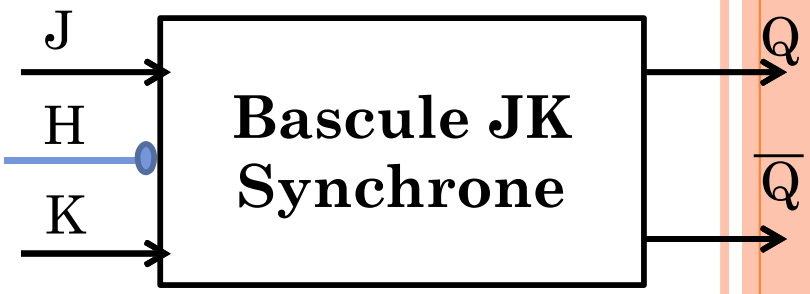
BASCULE JK

SYNCHRONE

- C'est une bascule avec deux entrées J et K et une horloge (front montant ou descendant)



Sur front montant ↑

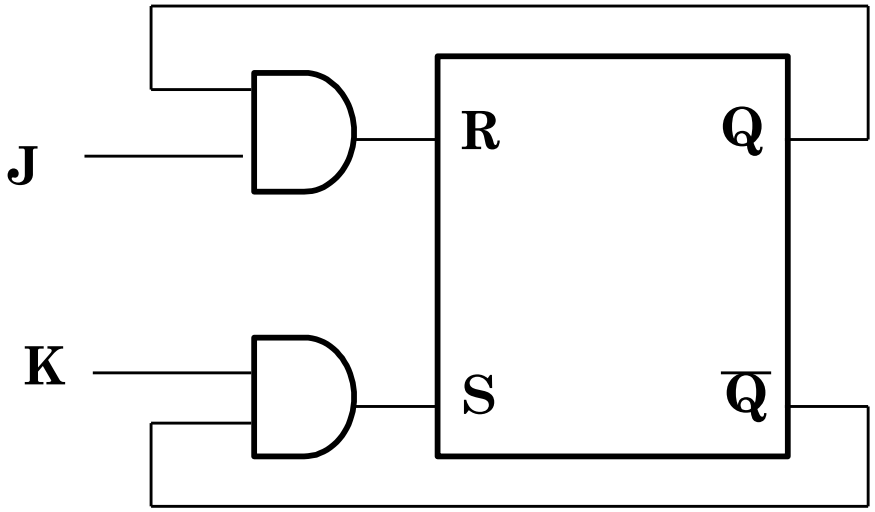


Sur front descendant ↓

H	J	K	Q+
0/1, ↑	X	X	Q-
↓	0	0	Q-
↓	0	1	0
↓	1	0	1
↓	1	1	$\overline{Q}$-

BASCULE JK

- **Exercice 5:** Réaliser une bascule JK asynchrone en utilisant une bascule RS.



BASCULE JK

- **Exercice 5:** Réaliser une bascule JK asynchrone en utilisant une bascule RS.

R	S	Q-	Q+
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	**1**	**0**	**X**
1	**1**	**1**	**X**

J	K	Q-	Q+
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	**1**	**0**	**1**
1	**1**	**1**	**0**

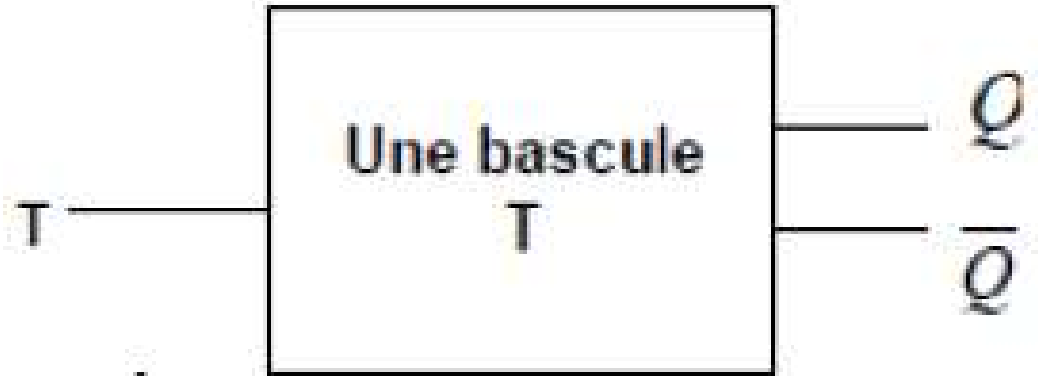
Bascule T

La bascule T (*toggle, **Trigger)** fonctionne sur front d'horloge.*

Elle permet de conserver la valeur de sortie précédente ou de l'inverser (faire le complément ou non de l'état actuel).

Ce type de bascule est particulièrement intéressant pour la réalisation de compteurs.

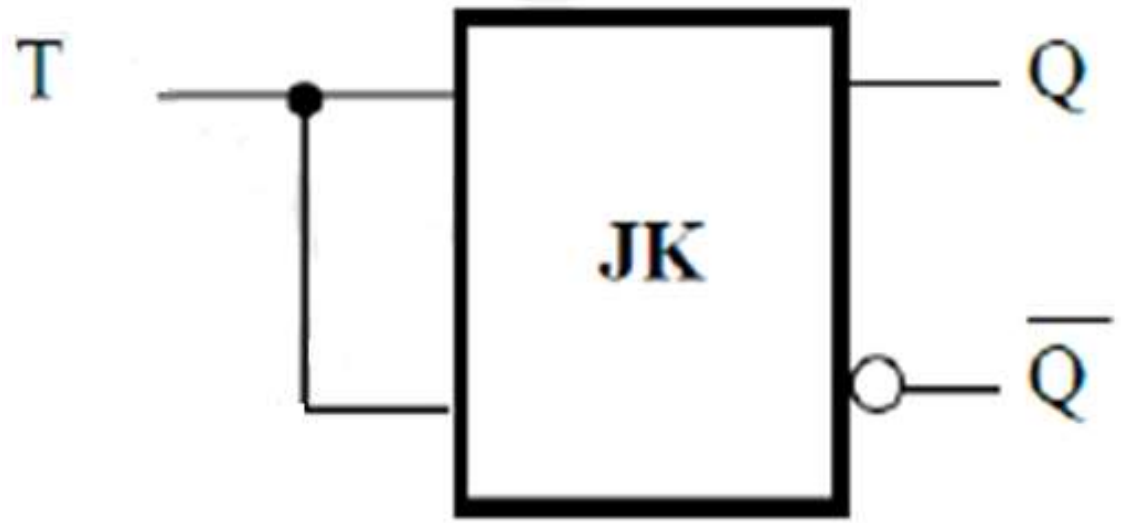
La bascule T peut être réalisée à partir d'une bascule D.



T		Q+
0		q
1		$\bar{q}$

Bascule T

Pour une bascule **J-K**, nous constatons que si **J=K=1**, l'état de la sortie est inversé à chaque cycle d'horloge.
Ainsi, La bascule T a une seule entrée appelée «**Trigger »** (mot anglais pour dire déclenchement) peut être obtenue par connexion de entrées **J** et **K** à une même source.
Il est pour cela parfois dit la bascule **complément.**

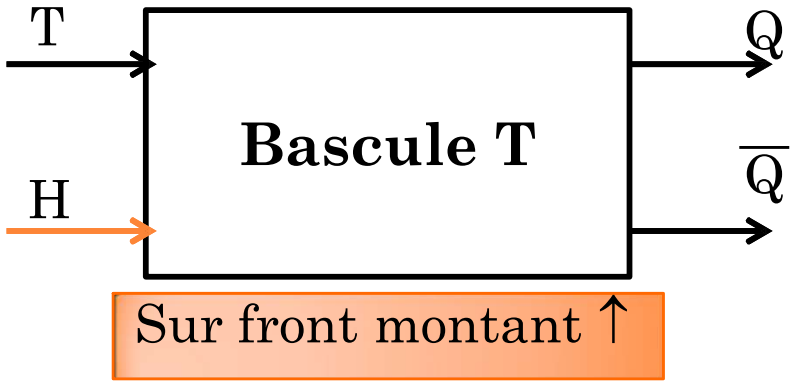


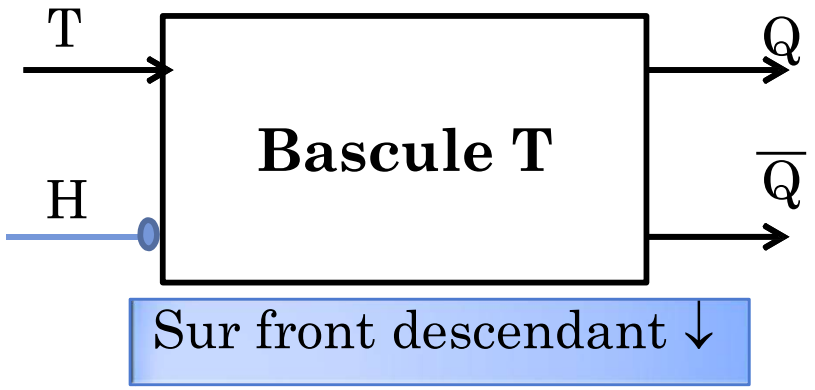
Bascule T

La bascule **T** synchrone est déclenchée par le signal d'horloge **H**.
L'unique entrée **T** (Trigger) commande l'état de la bascule.
La sortie **Q** change d'état chaque fois que l'entrée **T** passe à l'état logique **1** et conserve son état le reste du temps.

BASCULE T

- La bascule T (Toggle) bascule à chaque impulsion d'horloge (front montant ou descendant) lorsque son entrée T est active.

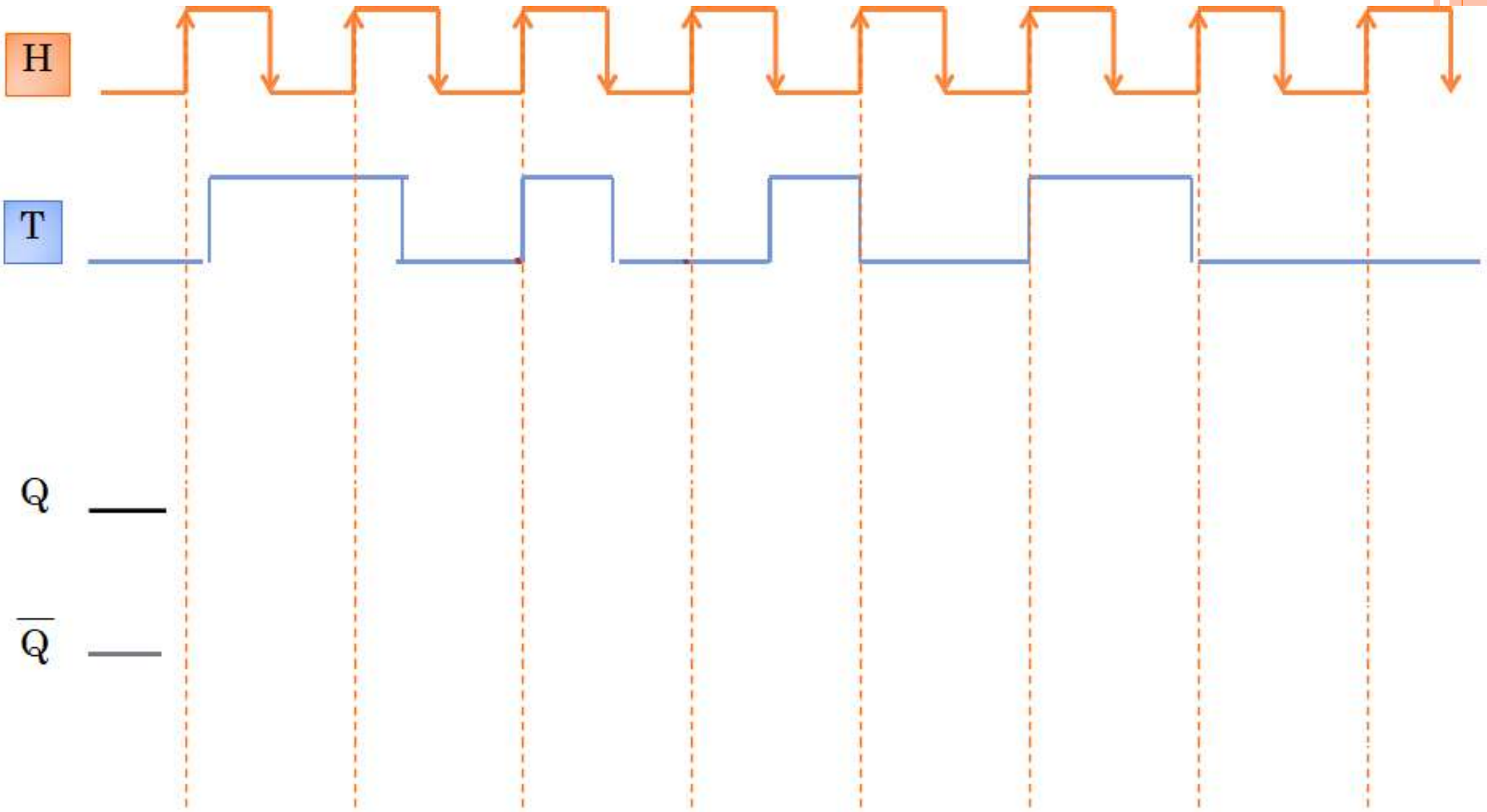




T	H	Q+
0	X	Q-
1	0/1, ↓	Q-
1	↑	$\overline{\mathbf{Q}}$**-**

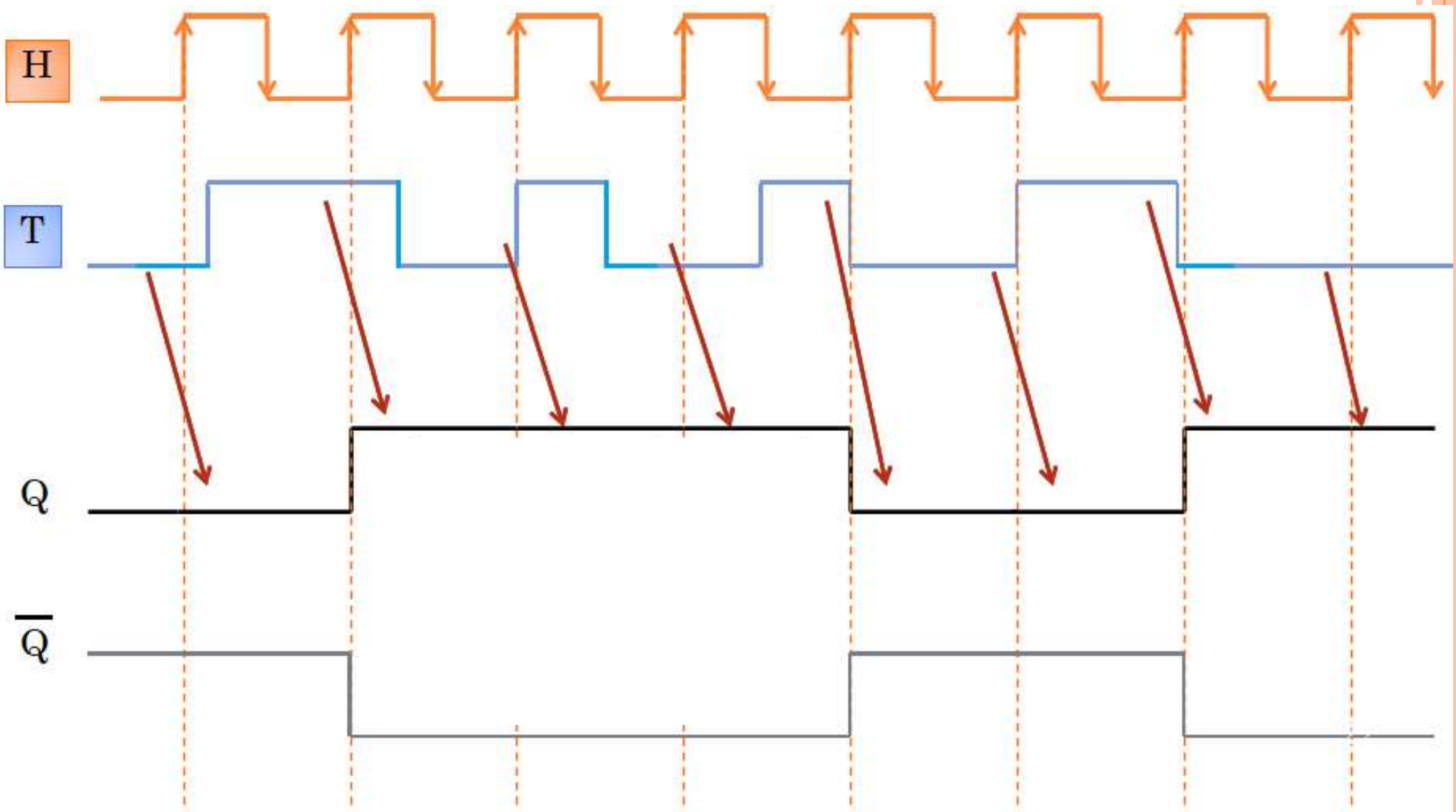
Bascule T

Chronogramme (Front Montant)



Bascule T

Chronogramme (Front Montant)

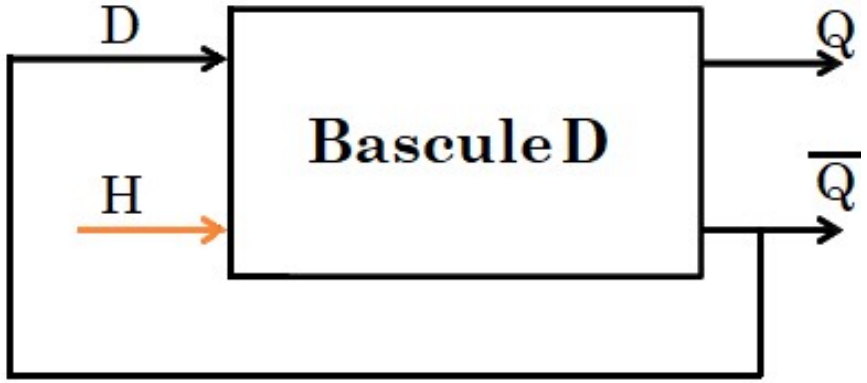


BASCULE T

- **Exercice 4:** Transformer une bascule D pour qu'elle agisse comme une bascule T (front montant).

H	Q+
0/1/↓	Q-
↑	D

T	H	Q+
0	X	Q-
1	0/1, ↓	Q-
1	↑	$\overline{Q}$-



Exercice

- **Exercice 6:** Transformer une bascule JK synchrone en une bascule D.

H	D	Q+
0/1	0	Q-
0/1	1	Q-
↑	0	0
↑	1	1

H	J	K	Q+
0/1	X	X	Q-
↓	0	0	Q-
↓	0	1	0
↓	1	0	1
↓	1	1	$\overline{Q}$-

$$H_{JK} = \overline{H}_D,\ J = D\ ;\ K = \overline{D}$$

Exercice

- **Exercice 7:** Transformer une bascule JK pour qu'elle agisse comme une bascule T (front descendant).

T	H	Q+
0	**X**	**Q-**
1	0/1, ↑	Q-
1	↓	$\overline{\mathbf{Q}}$**-**

H	J	K	Q+
0/1, ↑	**X**	**X**	**Q-**
↓	0	0	Q-
↓	0	1	0
↓	1	0	1
↓	**1**	**1**	$\overline{\mathbf{Q}}$**-**

$$H_T = H_{JK},\ J = K = T$$