

Chapitre 5: Les bascules

Licence en Ingénierie Informatique

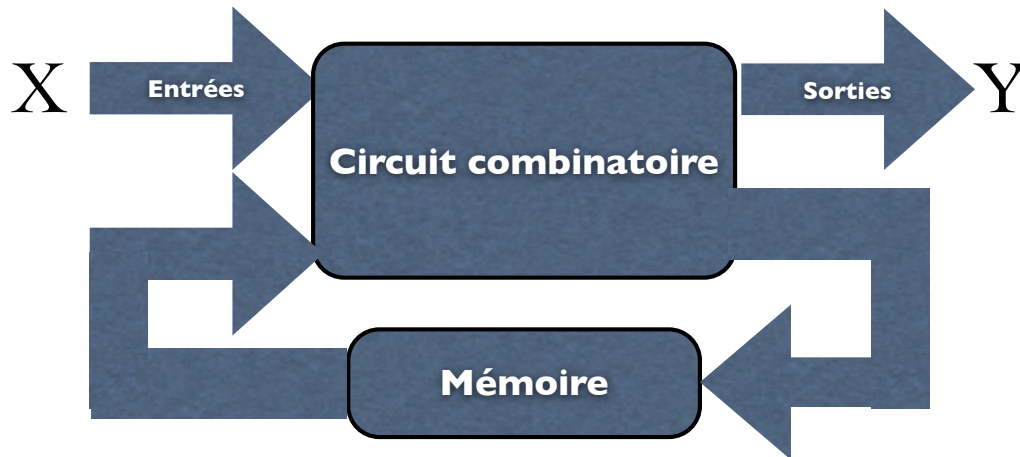
Pr Youssou FAYE

Année 2020-2021

Les circuits séquentiels

Pour un circuit séquentiel, les sorties Y dépendent des entrées X mais également de l'état Q du système. Celui-ci dépend aussi des entrées X .

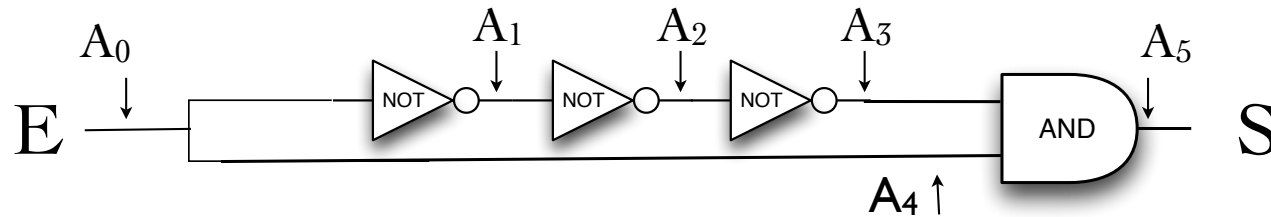
- ▶ $Y = f(X, Q_t)$
- ▶ $Q_t = f(X, Q_{t-1})$
- ▶ Une rétroaction des sorties sur les entrées
- ▶ Un nouveau paramètre temporel entre en jeu (état actuel, état antérieur, délai de propagation)



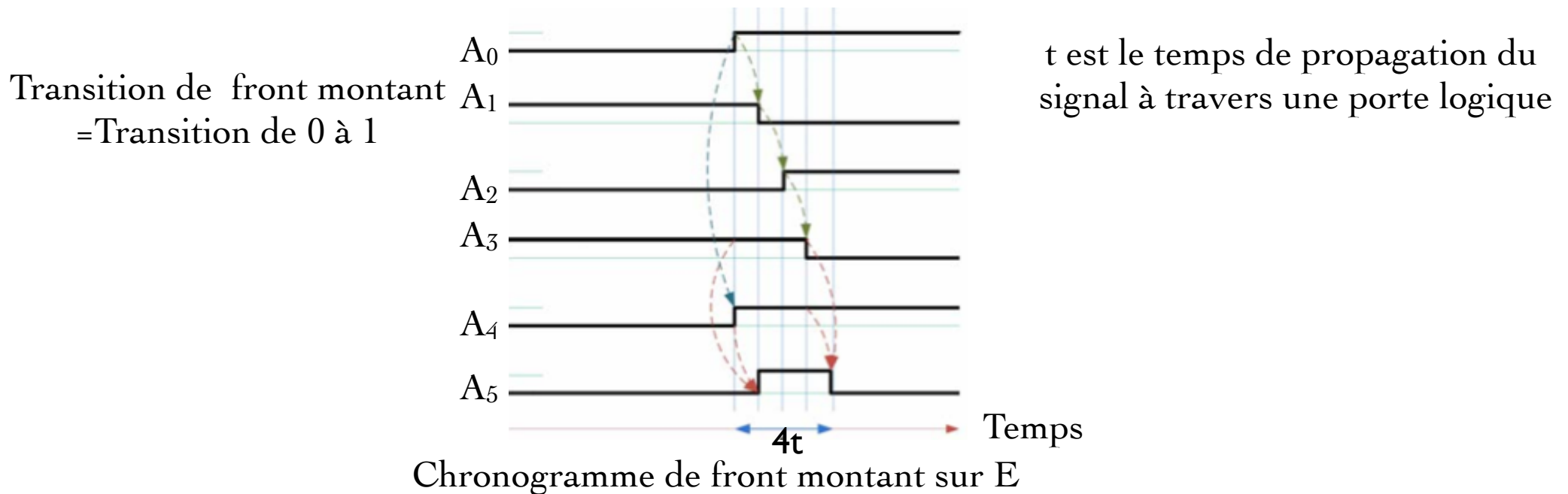
- **Exemples**
 - Bascule
 - Registre
 - compteur

Délai de propagation des portes logiques

- Temps mis par les signaux logiques entre l'entrée et la sortie d'une porte logique

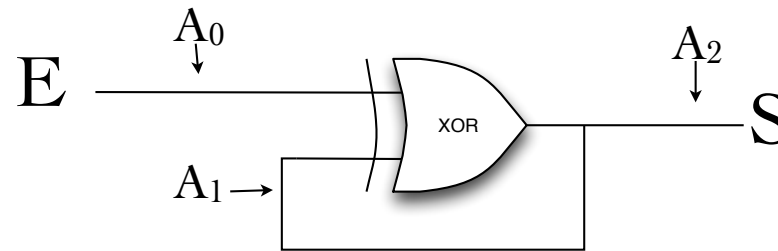


- L'équation booléenne de la sortie S en fonction de l'entrée E est: $S = \bar{\bar{\bar{E}}} \cdot E = \bar{\bar{E}} \cdot E = 0$

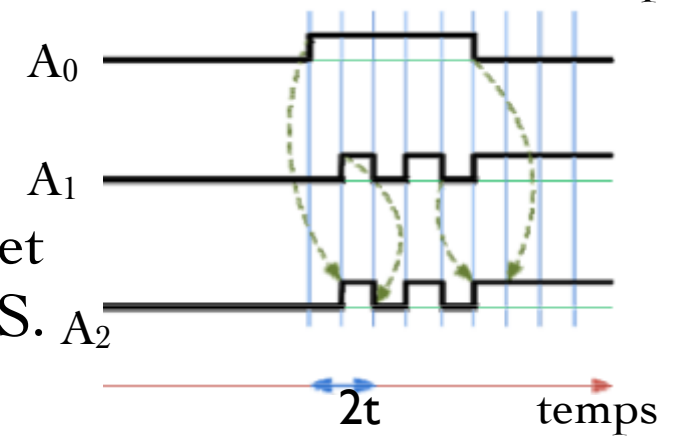
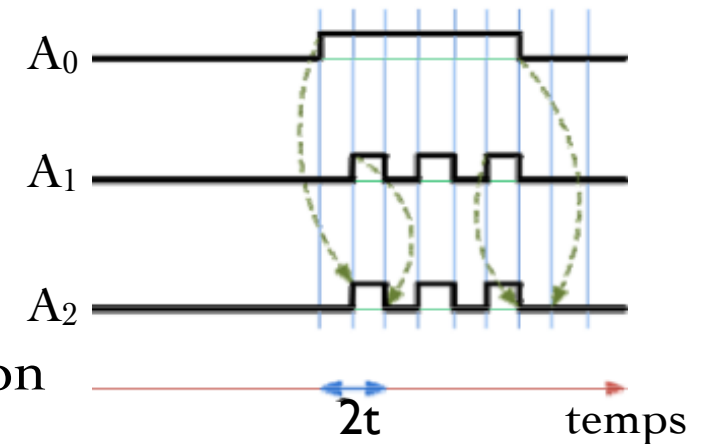


Phénomène de mémorisation et de stabilisation

- Créer une rétroaction entre les sorties et les entrées du circuit combinatoire
- Exemple: avec un XOR,



- Entrée unique (E), sortie unique (S)
- Sondons le circuit en A_0, A_1, A_2
- Rétroaction de A_2 à A_1 sans délai de propagation
- Supposons un état initial $E=0$ et $S=0$
- Si le signal E passe à 1 au point A_0 , la sortie S oscille entre 0 et 1 jusqu'à ce que E repasse à 0 et S peut valoir 0 ou 1 selon la dernière valeur de S .



Bascule

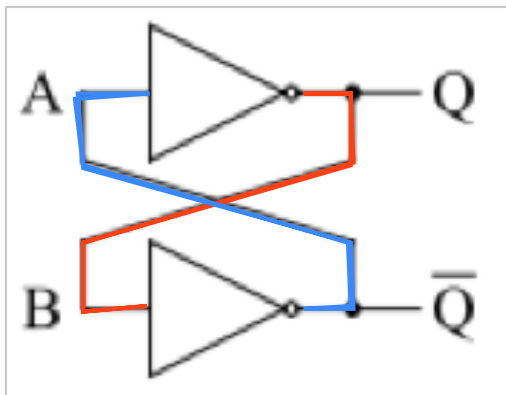
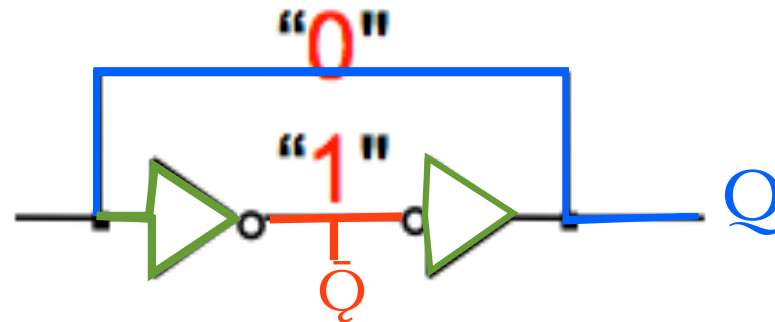
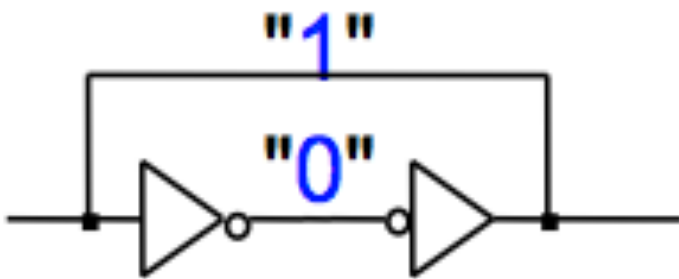
- ▶ Bascule: permet la mémorisation d'un bit
- ▶ Portes logiques: brique de base des circuits combinatoires: décodeur, multiplexeur.....
- ▶ Bascules: Eléments de base des circuits séquentiels:
- ▶ Circuits séquentiels fondamentaux
 - ▶ Bascule
 - ▶ registre,
 - ▶ compteur
 - ▶ RAM

Bascule

- ▶ On trouve deux grandes familles de bascules :
- ▶ bascules de mémorisation: elles possèdent les commandes de mise à zéro, mise à un, mémorisation ;
- ▶ bascules de comptage: elles possèdent en outre une commande de changement d'état.

Le bistable

- ▶ Deux inverseurs en série
- ▶ Principe de rétroaction
- ▶ Deux états possibles
- ▶ Pour "écrire", on modifie le bistable



$$(Q \text{ ou } Q_{(t-1)}=1) \Rightarrow (B=1) \Rightarrow (\bar{Q}=0) \Rightarrow (A=0) \Rightarrow (Q \text{ ou } Q_{(t)}=1)$$

$$(Q \text{ ou } Q_{(t-1)}=0) \Rightarrow (B=0) \Rightarrow (\bar{Q}=1) \Rightarrow (A=1) \Rightarrow (Q \text{ ou } Q_{(t)}=0)$$

On note deux états de la bascule

▶ "1" ($Q = 1, \bar{Q} = 0$)

▶ "0" ($Q = 0, \bar{Q} = 1$)

▶ Circuit à deux états stables possibles: appelé circuit **bistable**

Bascule RS (R=Reset, S=Set)

- ▶ La bascule RS (faite avec des portes NOR) a 2 entrées: R et S et 2 sorties ($Q_{(t)}$ et $\bar{Q}_{(t)}$)
- ▶ Fonctionnement
 - * Si $S=0$ et $R=0 \Rightarrow Q_{(t)} = Q_{(t-1)}$
 - * Si $S=1$ et $R=0 \Rightarrow Q_{(t)} = 1$ (état SET)
 - * Si $S=0$ et $R=1 \Rightarrow Q_{(t)} = 0$ (état RESET)
 - * Si $S=1$ et $R=1 \Rightarrow Q_{(t)} = \bar{Q}_{(t)} = 0 \Rightarrow ?$ (état indéterminé à interdire)

Entrées			Sorties		
S	R	$Q_{(t-1)}$	$Q_{(t)}$	$\bar{Q}_{(t)}$	Etats
0	0	0	0	1	$Q_{(t)} = Q_{(t-1)}$: Mémoire
0	0	1	1	0	$Q_{(t)} = Q_{(t-1)}$: Mémoire
0	1	0	0	1	Reset: Remise à 0
0	1	1	0	1	Reset: Remise à 0
1	0	0	1	0	Set: Remise à 1
1	0	1	1	0	Set: Remise à 1
1	1	0	0	0	Indéterminé(interdit)
1	1	1	0	0	Indéterminé(interdit)

Table de Vérité Résumée

S	R	$Q_{(t)}$	
0	0	$Q_{(t-1)}$	Mémorisation)
0	1	0	Reset
1	0	1	Set
1	1	X	Interdit

Bascule RS de base (R=Reset, S=Set)

► Résumé

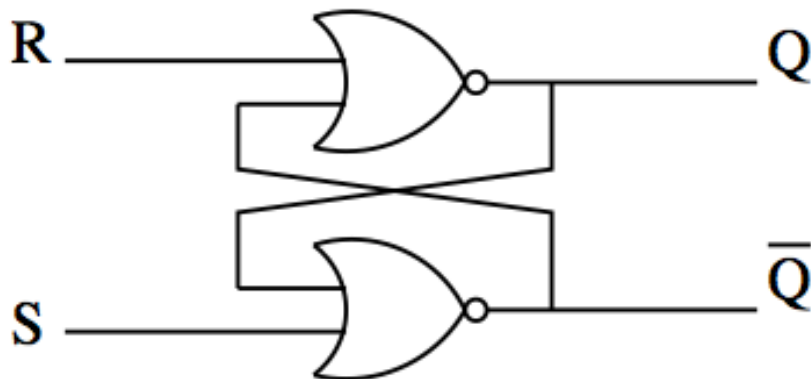
► Fonctionnement NAND

► Si $R=S=1$, Etat INTERDIT

► Si $R \neq S$, alors écriture $Q=S$

► $R=S=0$, Etat mémoire

S	R	$Q(t)$
0	0	$Q_{(t)}=Q_{(t-1)}$:Mémoire
0	1	Reset: Remise à 0
1	0	Set: Remise à 1
1	1	Indéterminé(interdit)



Bascule RS (R=Reset, S=Set)

- ▶ La bascule RS (avec des portes NAND) a 2 entrées: R et S et 2 sorties $Q_{(t)}$ et $\bar{Q}_{(t)}$
- ▶ Fonctionnement
 - * Si $S=0$ et $R=0 \Rightarrow Q_{(t)} = \bar{Q}_{(t)} = 1 \Rightarrow ?$ (état indéterminé à interdire)
 - * Si $S=0$ et $R=1 \Rightarrow Q_{(t)} = 1$ (état SET)
 - * Si $S=1$ et $R=0 \Rightarrow Q_{(t)} = 0$ (état RESET)
 - * Si $S=1$ et $R=1 \Rightarrow Q_{(t)} = Q_{(t-1)}$

Table de Vérité

Entrées			Sorties		
S	R	$Q_{(t-1)}$	$Q_{(t)}$	$\bar{Q}_{(t)}$	Etats
0	0	0	1	1	Indéterminé(interdit)
0	0	1	1	1	Indéterminé(interdit)
0	1	0	1	0	Set: Remise à 1
0	1	1	1	0	Set: Remise à 1
1	0	0	0	1	Reset: Remise à 0
1	0	1	0	1	Reset: Remise à 0
1	1	0	0	1	$Q_{(t)} = Q_{(t-1)}$:Mémoire
1	1	1	1	0	$Q_{(t)} = Q_{(t-1)}$:Mémoire



R	S	$Q_{(t)}$	
1	1	$Q_{(t-1)}$	Mémorisation)
0	1	0	Reset
1	0	1	Set
0	0	X	Indéterminé

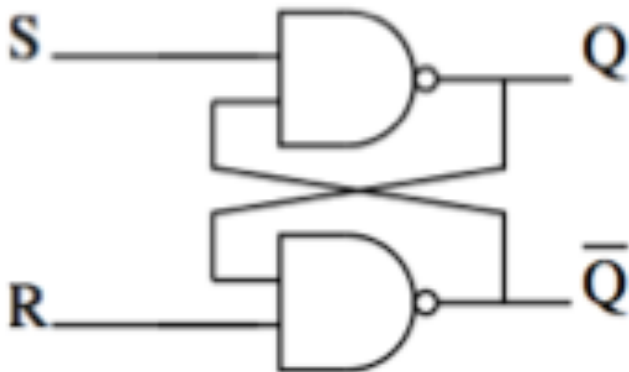
Bascule RS de base (R=Reset, S=Set)

► Résumé

► Fonctionnement NAND

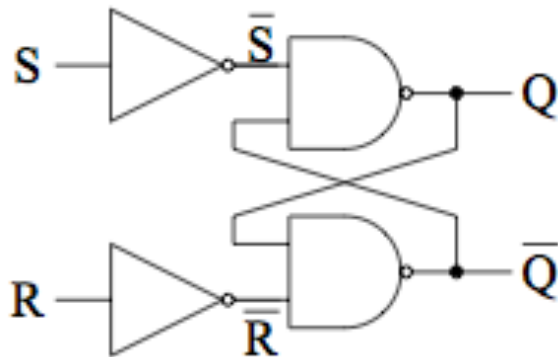
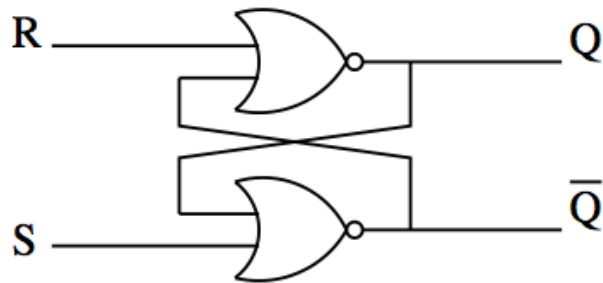
- Si $R=S=1$, Etat mémoire
- Si $R \neq S$, alors écriture $Q=R$
- $R=S=0$, INTERDIT

R	S	$Q_{(t)}$
1	1	$Q_{(t-1)}$
0	1	0
1	0	1
0	0	X



Bascule RS de base (R=Reset, S=Set)

- Inverser les entrées de RS NAND pour avoir la même table de vérité RS NOR



R	S	Q (t)
0	0	Q (t-1)
1	0	0
0	1	1
1	1	X

Sorties inchangées (mémoire)

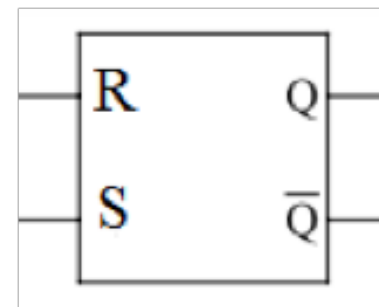
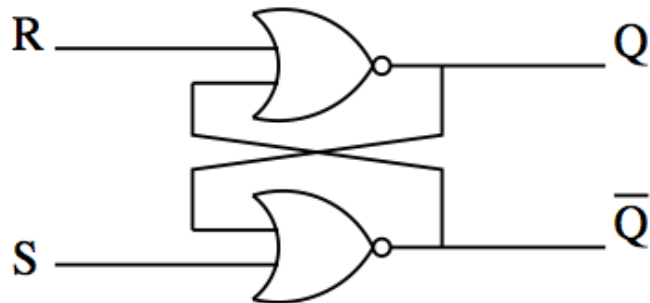
Reset : Remise à 0

Set : Remise à 1

Interdit

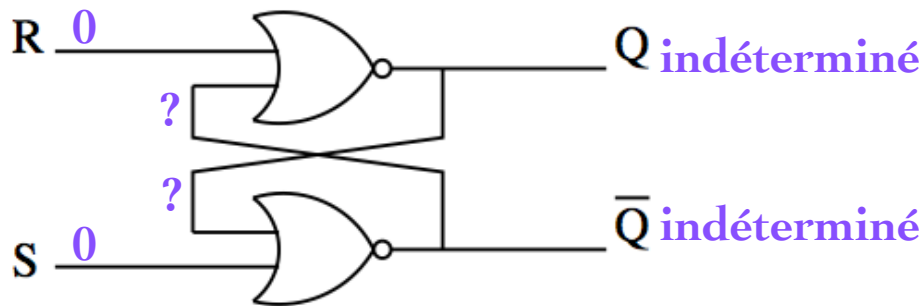
- Ce qu'on peut retrouver en utilisant les conséquences du théorème de Morgan: une porte NON-OU est une porte ET avec ses entrées inversées.

- Représentation simplifiée



Bascule RS de base (R=Reset, S=Set) (1)

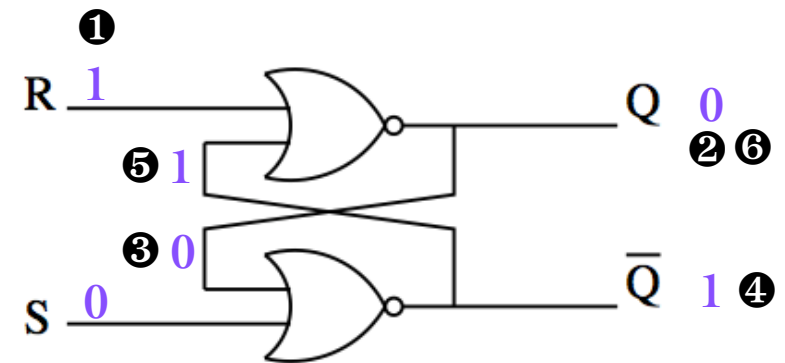
- ▶ Deux variables d'entrées
 - ▶ S pour la mise à l'état 1 de la bascule
 - ▶ R pour la mise à l'état 0
- ▶ Deux variables de sorties Q et \bar{Q}
 - ▶ Q est toujours le complément de \bar{Q} .
 - ▶ Quand $Q=\bar{Q}$ on parle d'état interdit
- ▶ **Fonctionnement avec NOR**



Quand la bascule est mise sous tension, on ne peut déterminer avec certitude l'état des sorties:

indétermination

1



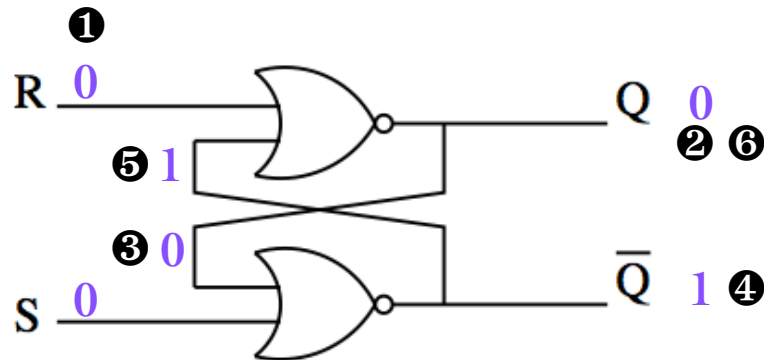
(R passe à 1) \Rightarrow (Q passe à 0) \Rightarrow (\bar{Q} passe à 1) \Rightarrow (Q reste toujours à 0)

L'état de la bascule est dit: **RESET**

2

Bascule RS de base (R=Reset, S=Set) (2)

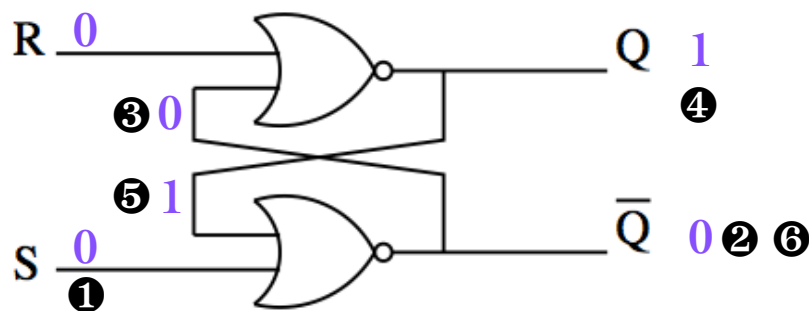
► Fonctionnement avec NOR



(R revient à 0) \Rightarrow (Q reste à 0) \Rightarrow (\bar{Q} reste à 1) \Rightarrow (Q reste toujours à 0)

L'état de la bascule est dit: **MEMOIRE**

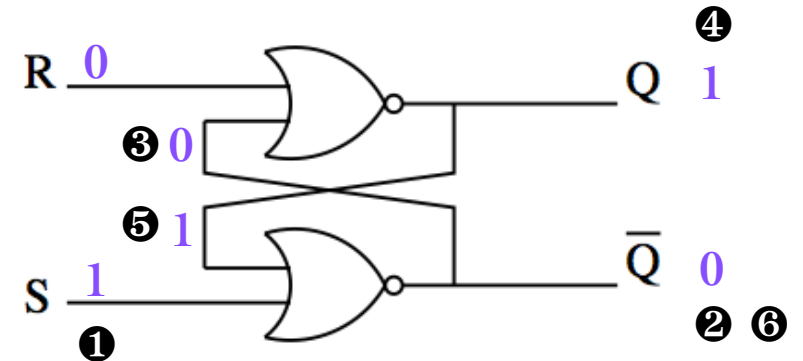
3



(S revient à 0) \Rightarrow (\bar{Q} reste à 0) \Rightarrow (Q reste à 1) \Rightarrow (\bar{Q} reste toujours à 0)

L'état de la bascule est dit: **MEMOIRE**

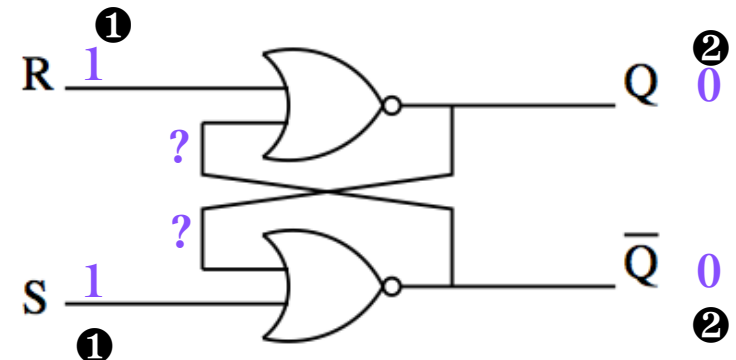
5



(S passe à 1) \Rightarrow (\bar{Q} passe à 0) \Rightarrow (Q passe à 1) \Rightarrow (\bar{Q} reste toujours à 0)

L'état de la bascule est dit: **SET**

4



(S et R sont à 1) \Rightarrow (Q et \bar{Q} sont à 0)

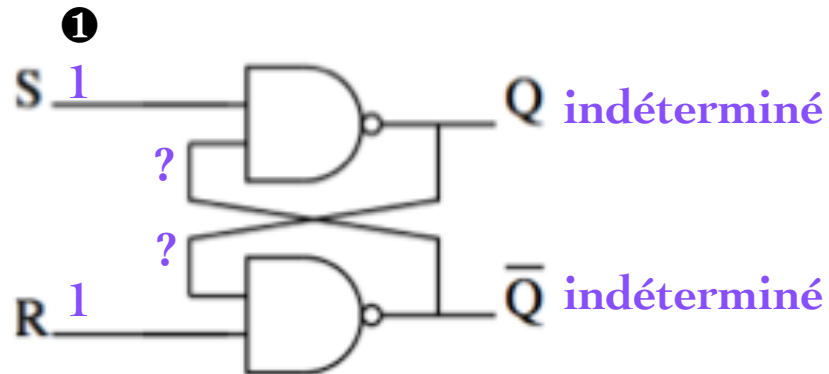
$Q = \bar{Q}$ (anormal)

L'état de la bascule est dit: **INTERDIT**

6

Bascule RS de base (R=Reset, S=Set) (3)

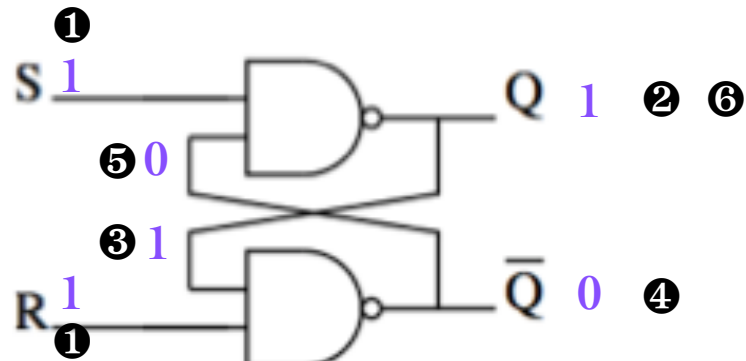
► Fonctionnement avec NAND



Quand la bascule est mise sous tension, on ne peut déterminer avec certitude l'état des sorties:

indétermination

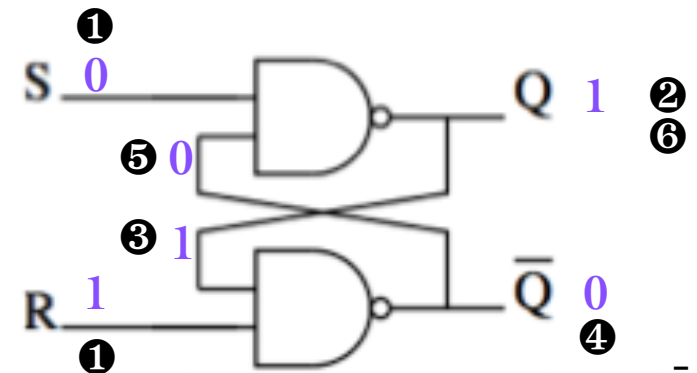
①



(S revient à 1) \Rightarrow (Q reste à 1) \Rightarrow (\bar{Q} reste à 0) \Rightarrow (Q reste toujours à 1)

L'état de la bascule est dit: MEMOIRE

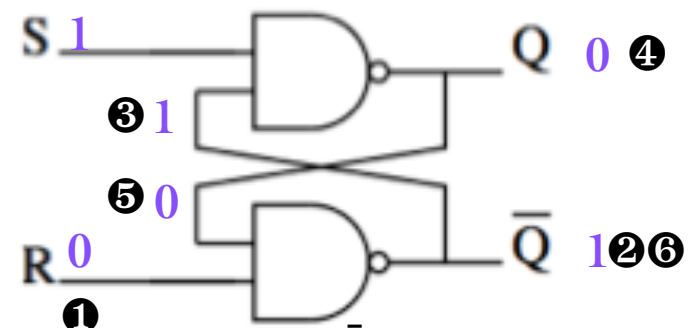
③



(S passe à 0) \Rightarrow (Q passe à 1) \Rightarrow (\bar{Q} passe à 0) \Rightarrow (Q reste toujours à 1)

L'état de la bascule est dit: SET

②



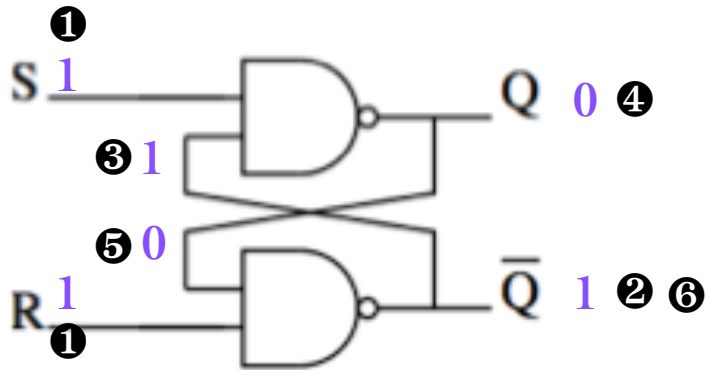
(R passe à 0) \Rightarrow (\bar{Q} passe à 1) \Rightarrow (Q passe à 0) \Rightarrow (\bar{Q} reste toujours à 1)

L'état de la bascule est dit: RESET

④

Bascule RS de base (R=Reset, S=Set) (4)

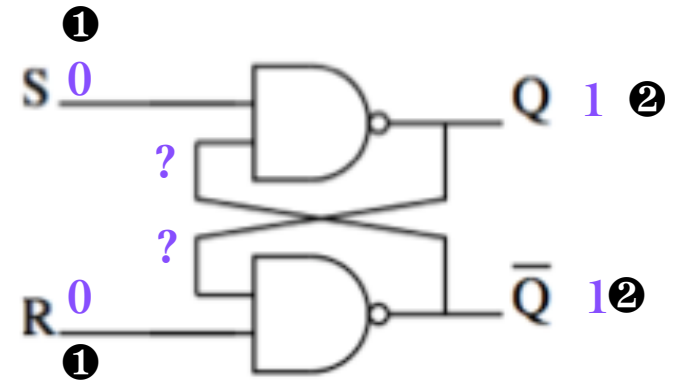
► Fonctionnement avec NAND



(R revient à 1) \Rightarrow (\bar{Q} passe à 1) \Rightarrow (Q reste à 0) \Rightarrow (\bar{Q} reste toujours à 1)

L'état de la bascule est dit: **MEMOIRE**

5



(S et R sont à 0) \Rightarrow (Q et \bar{Q} sont à 1)

$Q = \bar{Q}$ (anormal)

L'état de la bascule est dit: **INTERDIT**

6

NB1: L'état $S=R=0 \rightarrow Q = \bar{Q}=1$, est aussi appelé **indéterminé** car si R et S repasse à 1 simultanément, l'une des deux sorties doit prendre l'état 0: il est impossible de prédire la quelle

NB2: Les états Set et Reset relatifs aux entrées S et R dépendent de la position de S et R par rapport à Q et \bar{Q} .

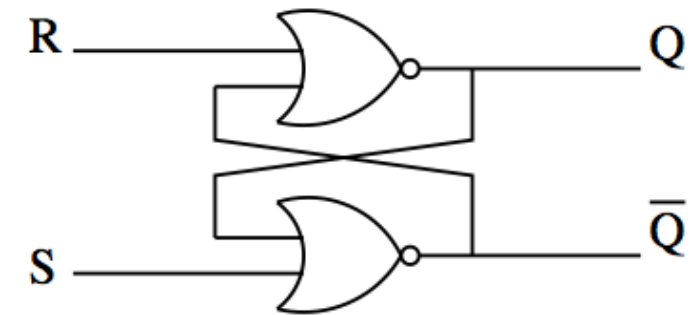
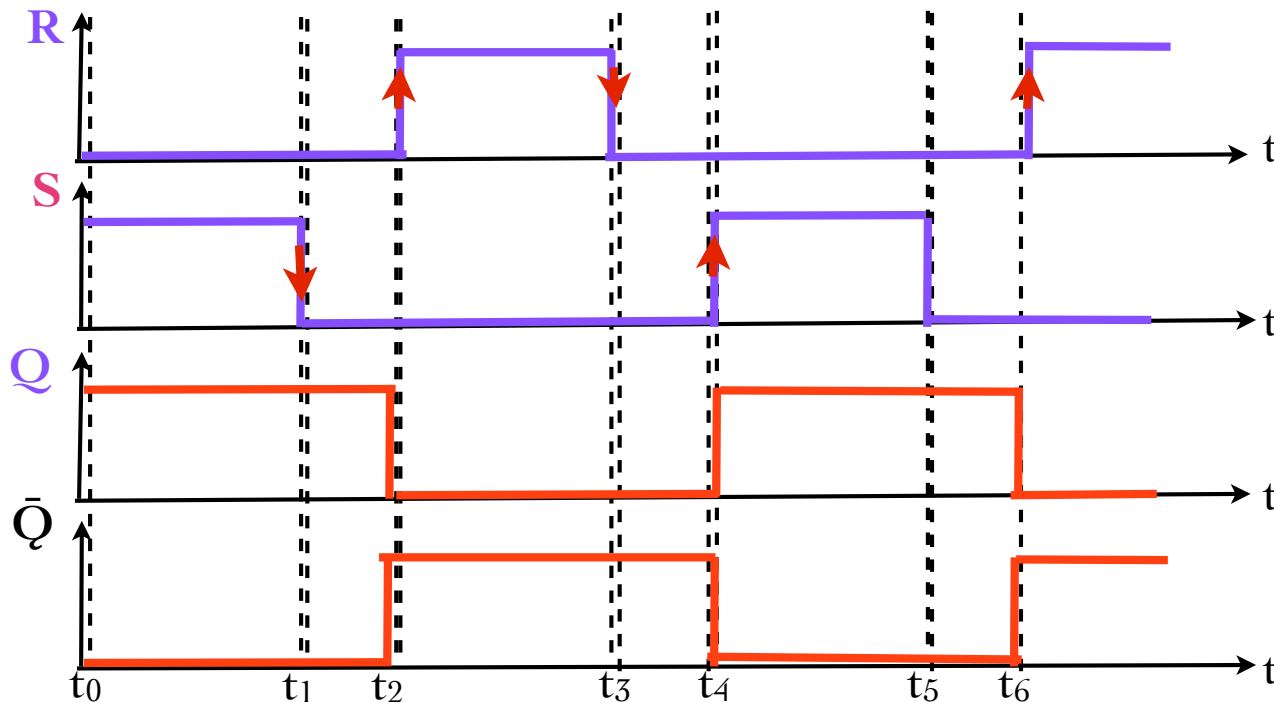
Bascule RS: Chronogramme

► Résumé

► Fonctionnement NOR

- Si $R=S=0$, Etat mémoire
- Si $R \neq S$, alors écriture $Q=S$
- $R=S=1$, INTERDIT

R	S	Q (t)
0	0	Q (t-1)
1	0	0
0	1	1
1	1	X



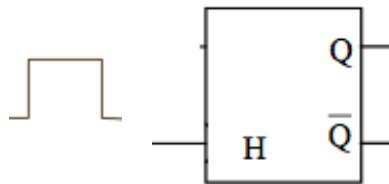
- Si on suppose au départ à t_0 $S=1$, $R=0$
- A t_1 S passe à 0 $\Rightarrow Q$ reste à 1
- A t_2 R passe à 1 $\Rightarrow Q$ passe à 0
- A t_3 R passe à 0 $\Rightarrow Q$ reste à 0
- A t_4 S passe à 1 $\Rightarrow Q$ passe à 1

- Etat mémoire
- Etat Set
- Etat Reset

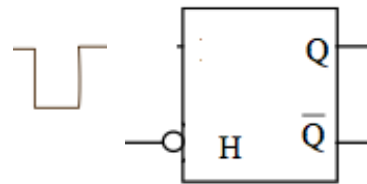
Bascule Asynchrone/Synchrone

- ▶ Bascule Asynchrone,
 - * La sortie évolue dès lors qu'un changement a lieu sur l'une des entrées
 - * Exemple: la Bascule RS déjà vu à la page précédente
- ▶ Bascule Synchrone
 - * La sortie évolue quand le signal d'horloge est actif
 - * Actif par **niveau**: niveau haut ou bas, le circuit est dit à excitation statique
 - * Actif par **front**: avant (montant) ou arrière (descendant), le circuit est dit à excitation dynamique

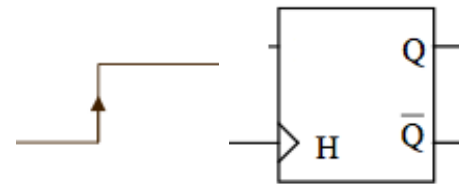
Actif par niveau haut



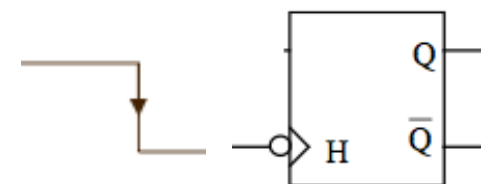
Actif par niveau bas



Actif par front avant

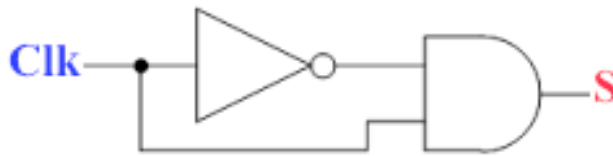


Actif par front arrière

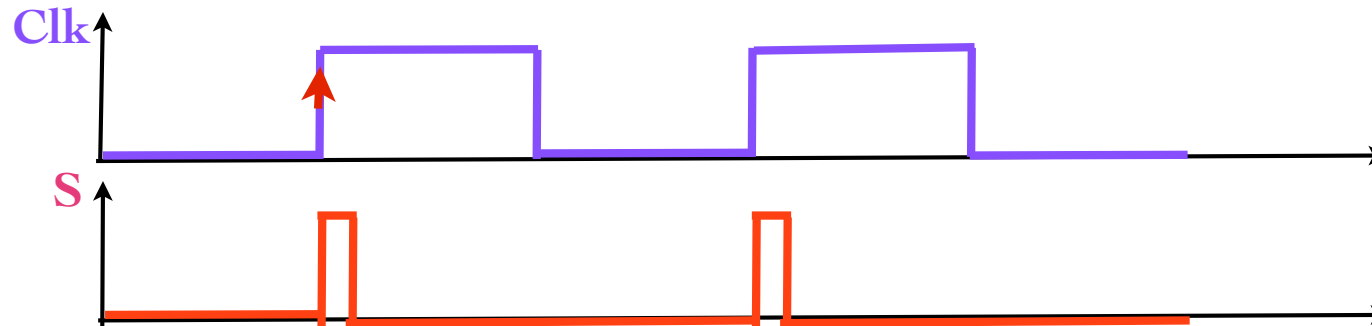
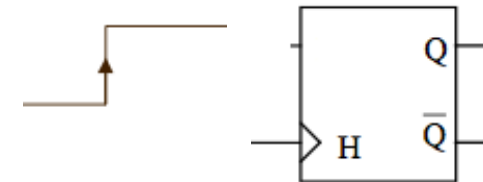


Synchrone sur front montant ou descendant

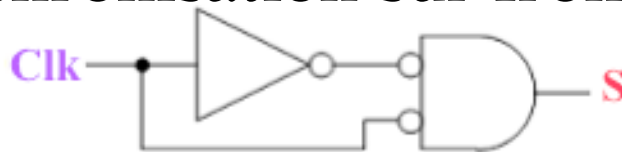
► Synchronisation sur front montant



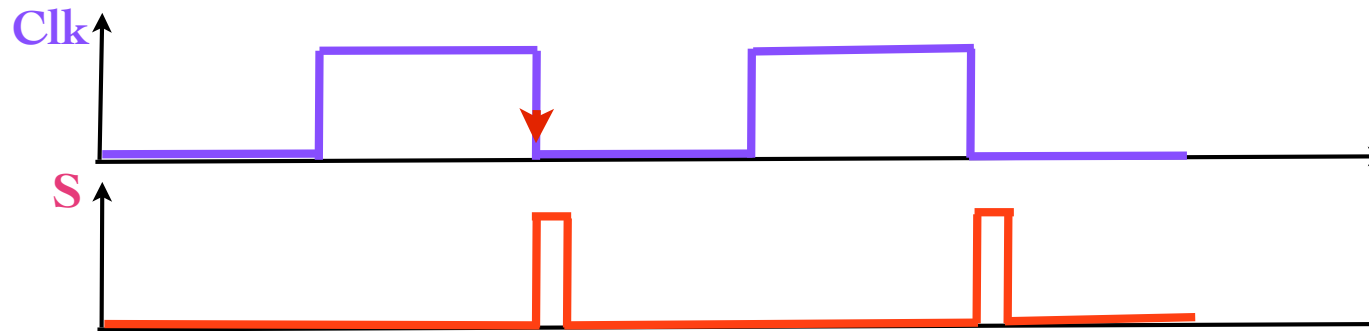
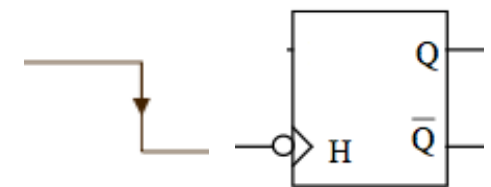
Actif par front avant



► Synchronisation sur front descendant



Actif par front arrière



Quand Clk passe de 0 à 1, un pic de durée de porte NAND ressemble à un front montant

22 Quand Clk passe de 1 à 0, un pic de durée de porte NAND ressemble à un front descendant

Bascule RS avec signal d'activation: RSH (1)

- ▶ Bascule commandée par un signal d'activation périodique Clk (appelé horloge)
- ▶ Trois entrées (S,R, Clk) deux sorties (Q et \bar{Q})
- ▶ La bascule fonctionne normalement quand Clk est actif (Clk=1)
- ▶ La bascule mémorise la dernière valeur lorsque Clk est inactif (Clk=0)
- ▶ Introduction de portes ET aux entrées

Ambiguïté de la RSH:

Si $\text{Clk}=\text{R}=\text{S}=1 \rightarrow \text{Q}=\bar{\text{Q}}=0$: Interdit, **Indéterminé** car Clk repasse à 0 , l'une des deux sorties doit prendre l'état 1: il est impossible de prédire la quelle.

Interdit \neq Indéterminé

Quand Clk=1, le circuit fonctionne, et quand Clk=0 le circuit mémorise la dernière valeur

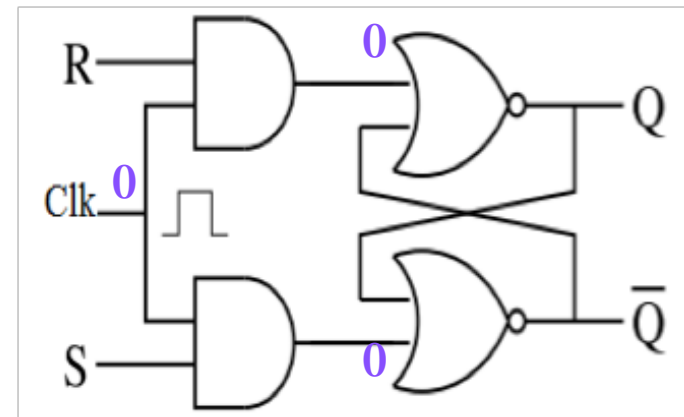
Table de Vérité

Entrées			Sorties
Clk	R	S	$\text{Q}_{(t)}$
0	-	-	$\text{Q}_{(t-1)}$
1	0	0	$\text{Q}_{(t-1)}$
1	0	1	1
1	1	0	0
1	1	1	X

mémorise la dernière valeur

Sorties inchangées

Indéterminé

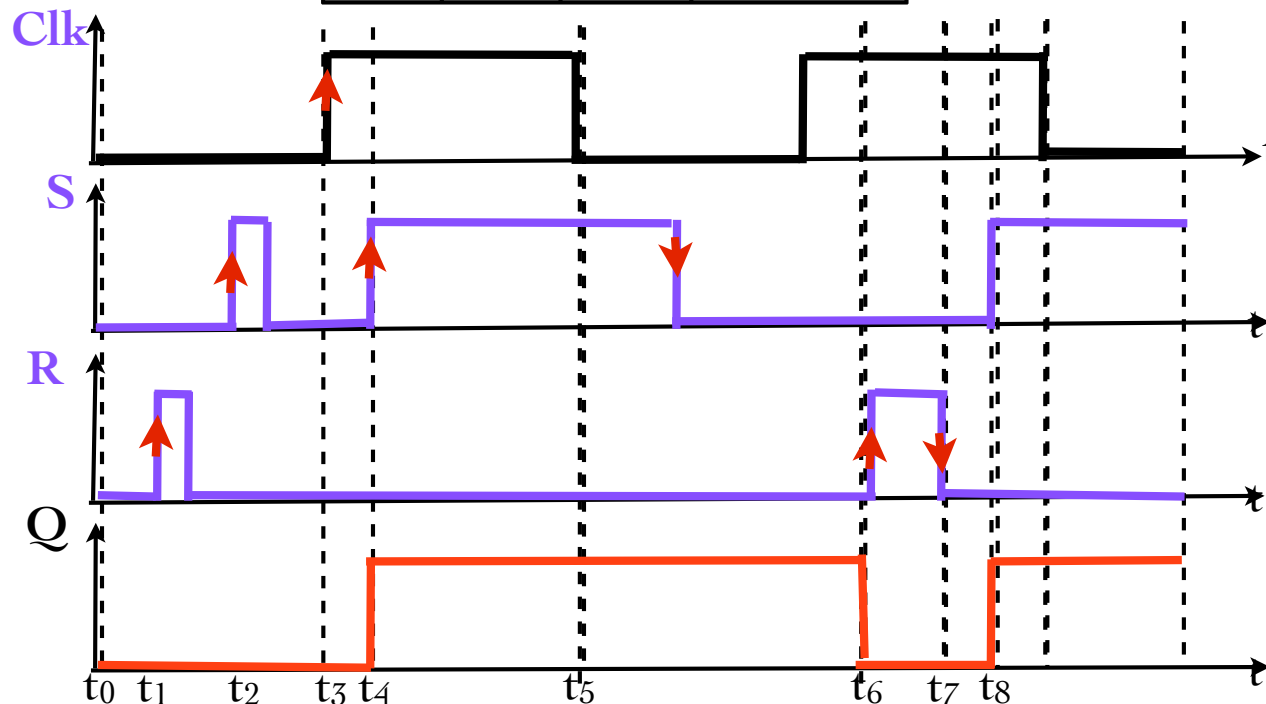
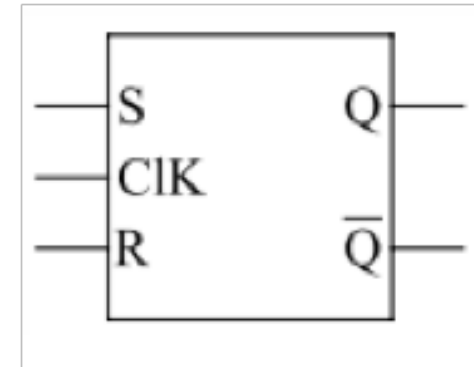


Basculer RS synchrone avec l'horloge à l'état **bloqué**

Bascule RS avec signal d'activation: RSH (2)

Table de Vérité

Entrées			Sorties
Clk	R	S	$Q_{(t)}$
0	-	-	$Q_{(t-1)}$
1	0	0	$Q_{(t-1)}$
1	0	1	1
1	1	0	0
1	1	1	X



Chronogramme

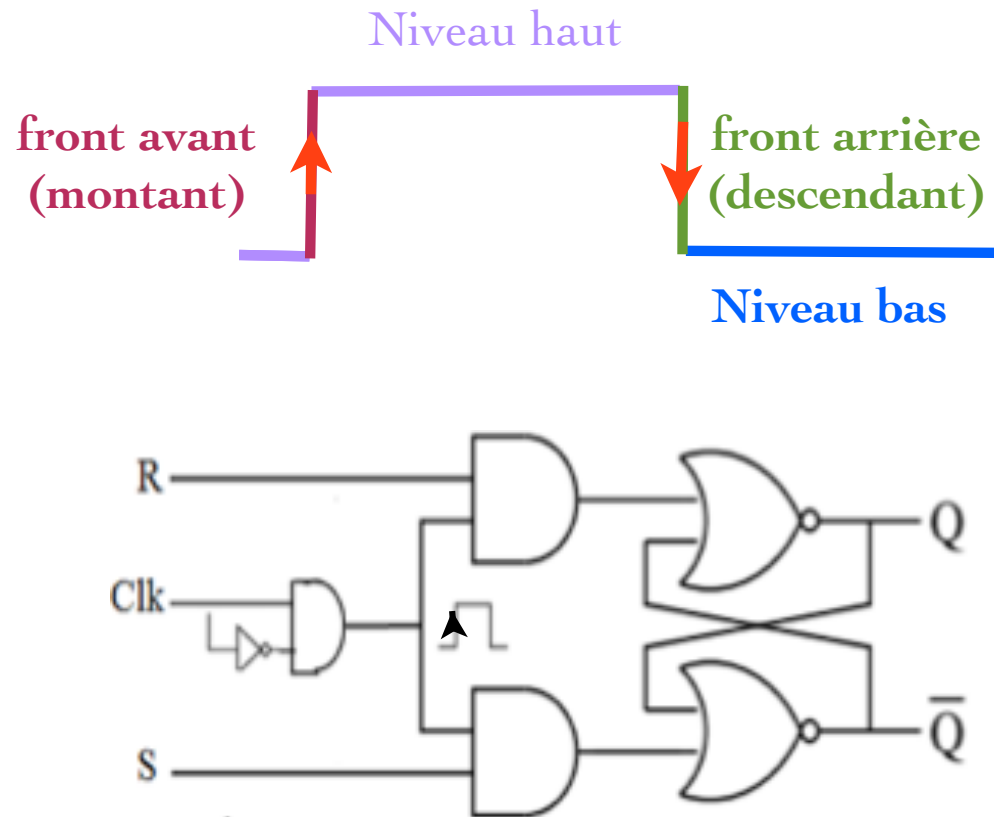
- ▶ Si on suppose au départ à t_0 la bascule est Reset $Q=0$
- ▶ A t_1 R passe à 1, comme $Clk=0 \Rightarrow Q_{(t)} = Q_{(t-1)}$ (état mémoire)
- ▶ Idem à t_2 quand S passe à 1 Q passe à 1
- ▶ A t_3 Clk passe à 1 $\Rightarrow Q_{(t)} = Q_{(t-1)}$
- ▶ A t_4 S passe à 1 $\Rightarrow Q_{(t)}$ passe à 1
- ▶ A t_5 Clk passe à 0 $\Rightarrow Q_{(t)}$ reste à 1
- ▶ A t_6 R passe à 1 $\Rightarrow Q_{(t)}$ passe à 0



- ▶ Etat mémoire
- ▶ Etat Set
- ▶ Etat Reset

Bascule RS avec signal d'activation: RSH (3)

► Bascule activée par front avant



Entrées			Sorties
Clk	R	S	$Q_{(t)}$
0	-	-	$Q_{(t-1)}$
1	-	-	$Q_{(t-1)}$
↑	0	0	$Q_{(t-1)}$
↑	0	1	1
↑	1	0	0
↑	1	1	X

Bascule RS synchrone avec front montant

Quand Clk passe de 0 à 1, un pic de durée de porte NAND

Bascule D (Delay)

- ▶ La bascule D recopie, sur sa sortie Q, l'unique signal D appliqué à son entrée avec une période d'horloge Clk.
- ▶ Le bistable D dérive de la SRH
 - ▶ Si $\text{Clk} = 1$ et $D=1 \Rightarrow Q_{(t)}=1= D$
 - ▶ Si $\text{Clk} = 1$ et $D=0 \Rightarrow Q_{(t)}=0= D$
 - ▶ Si $\text{Clk}=0$ $Q_{(t)}$ ne change pas (mémoire la dernière valeur)

$$Q=D$$

Table de Vérité

Entrées			Sorties
D	Clk	$Q_{(t-1)}$	$Q_{(t)}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Entrées		Sorties
Clk	D	$Q_{(t)}$
0	-	$Q_{(t-1)}$
1	0	0
1	1	1



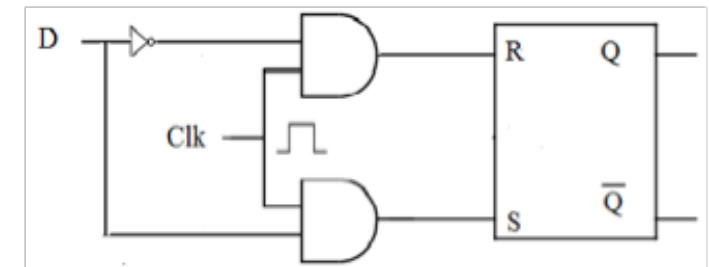
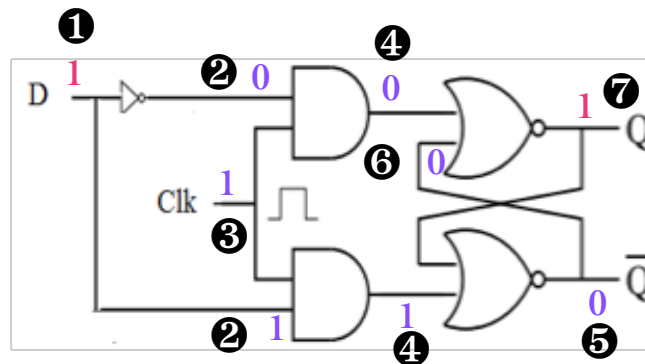
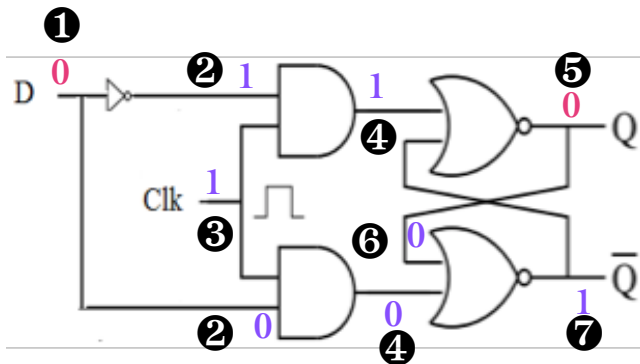
Table de Vérité
(version condensée)

Entrée	Sortie
D	$Q_{(t)}$
0	0
1	1

- Equation booléenne après simplification: $Q_{(t)} = D\text{Clk} + Q_{(t-1)}\overline{\text{Clk}}$

Bascule D (Delay)

- ▶ Résous l'ambiguïté de la bascule RS, RSH
- ▶ Principe: faire en sorte que l'état $R=S=1$ ne soit jamais présent à l'entrée de la bascule
- ▶ Réunir les signaux SR par un seul signal D qui correspond à la donnée que l'on veut écrire dans la mémoire de la bascule
 - ▶ Si $\text{Clk} = 1$ et $D=1 \Rightarrow Q=1$
 - ▶ Si $\text{Clk} = 1$ et $D=0 \Rightarrow Q=0$
 - ▶ Si $\text{Clk}=0$ $Q_{(t)}$ ne change pas



Bascule D synchrone

Bascule T (Toggle ou Trigger)

- ▶ La bascule T a une seule entrée T.
- ▶ A l'entrée d'une impulsion, la sortie Q est inversée
 - ▶ Si $T=0$, la valeur $Q_{(t)}=Q_{(t-1)}$ est maintenue
 - ▶ Si $T=1$, la valeur de $Q_{(t)} = \bar{Q}_{(t-1)}$ est inversée

Table de Vérité

Entrées			Sorties	
T	Clk	$Q_{(t-1)}$	$Q_{(t)}$	
0	0	0	0	$Q_{(t)}=Q_{(t-1)}$
0	0	1	1	$Q_{(t)}=Q_{(t-1)}$
0	1	0	0	$Q_{(t)}=Q_{(t-1)}$
0	1	1	1	$Q_{(t)}=Q_{(t-1)}$
1	0	0	0	$Q_{(t)}=Q_{(t-1)}$
1	0	1	1	$Q_{(t)}=Q_{(t-1)}$
1	1	0	1	$Q_{(t)} = \bar{Q}_{(t-1)}$
1	1	1	0	$Q_{(t)} = \bar{Q}_{(t-1)}$

Synchrone quand Clk=1



Entrées		Sorties
T	$Q_{(t-1)}$	$Q_{(t)}$
0	0	0
0	1	1
1	0	1
1	1	0



Table de Vérité
(version condensée)

Entrée	Sortie
T	$Q_{(t)}$
0	$Q_{(t-1)}$
1	$\bar{Q}_{(t-1)}$

$$Q_{(t)} = \bar{T}Q_{(t-1)} + T\bar{Q}_{(t-1)}$$

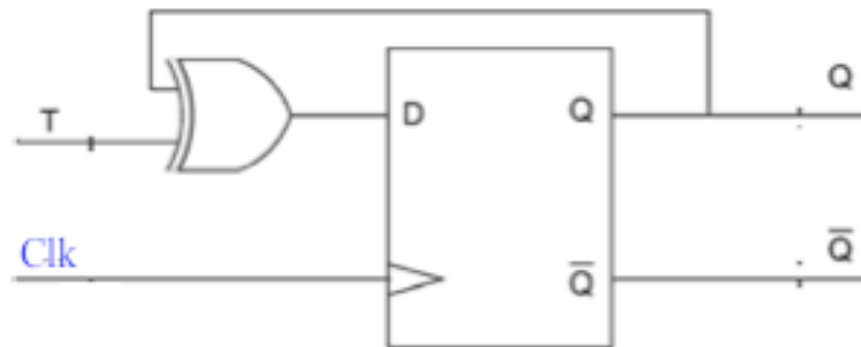
$$Q_{(t)} = T \oplus Q_{(t-1)}$$

Bascule T (Toggle ou Trigger)

- ▶ La bascule T est une bascule D à laquelle on ajoute une rétroaction depuis la sortie Q vers un XOR
- ▶ Une seule entrée T
 - ▶ Si $T=0$, la valeur $Q=Q_{(t-1)}$ est maintenue au front montant
 - ▶ Si $T=1$, la valeur de $Q= \bar{Q}_{(t-1)}$ est inversée au front montant

Table de Vérité

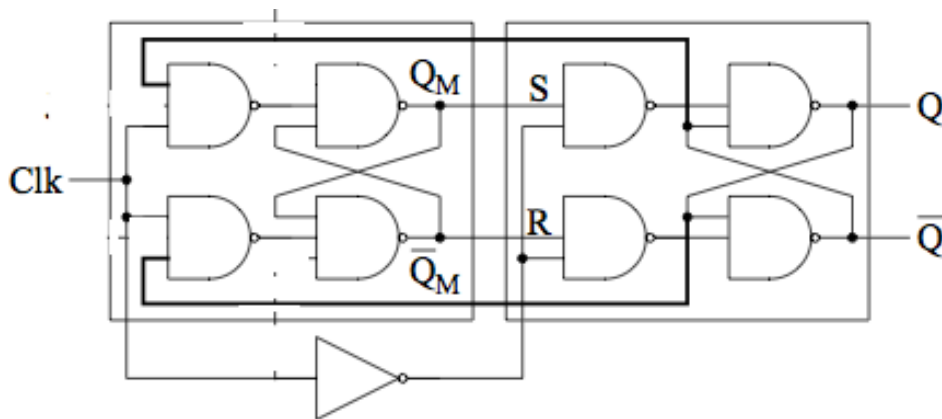
Clk	T	$Q(t)$
0	-	$Q_{(t-1)}$
↑	0	$Q_{(t-1)}$
↑	1	$\bar{Q}_{(t-1)}$



Bascule T

Bascule Maître-Esclave

- ▶ Objectif: éviter l'état interdit $S=R=1$
- ▶ Deux Bascules RSH connectées en cascade, les connexions entre les entrées et sorties sont croisées
- ▶ RSH de gauche est le maître, celle de droite l'esclave
- ▶ Les deux horloges sont reliées entre elles
- ▶ Lorsque $Clk_M=1$ pour le maître $\Rightarrow Clk_E=0$ pour l'esclave, Q_M et \bar{Q}_M vont être recopiés dans les entrées de l'esclave qui les bloque car $Clk_E=0$
- ▶ Quand Clk_M passe à 0 $\Rightarrow Clk_E=1$, l'esclave envoie Q_M et \bar{Q}_M comme entrées, ils sortent respectivement Q et \bar{Q} et sont re-bouclées aux entrées du maître qui les bloque aussitôt (si $Clk_M=0$)
- ▶ Lorsque Clk_M reste = 0, le système est bloqué
- ▶ Pas de possibilité pour l'utilisateur d'imposer une valeur initiale
- ▶ Il n'y a pas d'état mémoire



- ▶ Si $Q_M=1, \bar{Q}_M=0 \Rightarrow (S=1 \text{ et } R=0) \Rightarrow Q=1, \bar{Q}=0$
- ▶ Si $Q_M=0, \bar{Q}_M=1 \Rightarrow (S=0 \text{ et } R=1) \Rightarrow \bar{Q}=1, Q=0$

Bascule JK (1)

- ▶ Elle possède deux entrées J et K et lève l'ambiguïté (l'indétermination quand $S=R=1$) qui existe dans la bascule RS en asservissant les entrées R et S aux sorties Q et \bar{Q}
- ▶ Règle le problème de la bascule maître-esclave
 - ▶ Permet l'état mémoire ($S=R=0 \Rightarrow J=K=0$)
 - ▶ Permet à l'utilisateur d'imposer une valeur initiale
- ▶ Fonctionne à mi chemin entre la bascule RS et la bascule T
 - ▶ Comportement proche de la bascule T
 - ▶ Conserve la valeur de Q quand $J=K=0$
 - ▶ Inverse la valeur de Q quand $J=K=1$
 - ▶ Comportement proche de la bascule RS
 - ▶ Si $JK=10$, $Q=1$
 - ▶ Si $JK=01$, $Q=0$

Bascule JK (2)

- ▶ Si $JK=11$, $Q_{(t)} = \bar{Q}_{(t-1)}$
- ▶ Si $JK=00$, $Q_{(t)} = Q_{(t-1)}$
- ▶ Si $JK=10$, $Q = 1$
- ▶ Si $JK=01$, $Q = 0$

▶ Table de vérité

Entrées				Sorti	
J	K	$Q_{(t-1)}$	$\bar{Q}_{(t-1)}$	$Q_{(t)}$	
0	0	0	1	0	$Q_{(t)}=Q_{(t-1)}$
0	0	1	0	1	
0	1	0	1	0	$Q_{(t)}=0$
0	1	1	0	0	
1	0	0	1	1	$Q_{(t)}=1$
1	0	1	0	1	
1	1	0	1	1	$Q_{(t)}=\bar{Q}_{(t-1)}$
1	1	1	0	0	

On a éliminé 8/16 combinaisons, là où $Q = \bar{Q}$



Entrées		Sorties
J	K	$Q_{(t)}$
0	0	$Q_{(t-1)}$
0	1	0
1	0	1
1	1	$\bar{Q}_{(t-1)}$

Table Karnaugh

$\begin{smallmatrix} J \\ K \end{smallmatrix}$	0	1
0	$Q_{(t-1)}$	1
1		$\bar{Q}_{(t-1)}$

- Equation booléenne après simplification (Karnaugh)
- $Q_{(t)} = J\bar{Q}_{(t-1)} + \bar{K}Q_{(t-1)}$

Bascule JK (3)

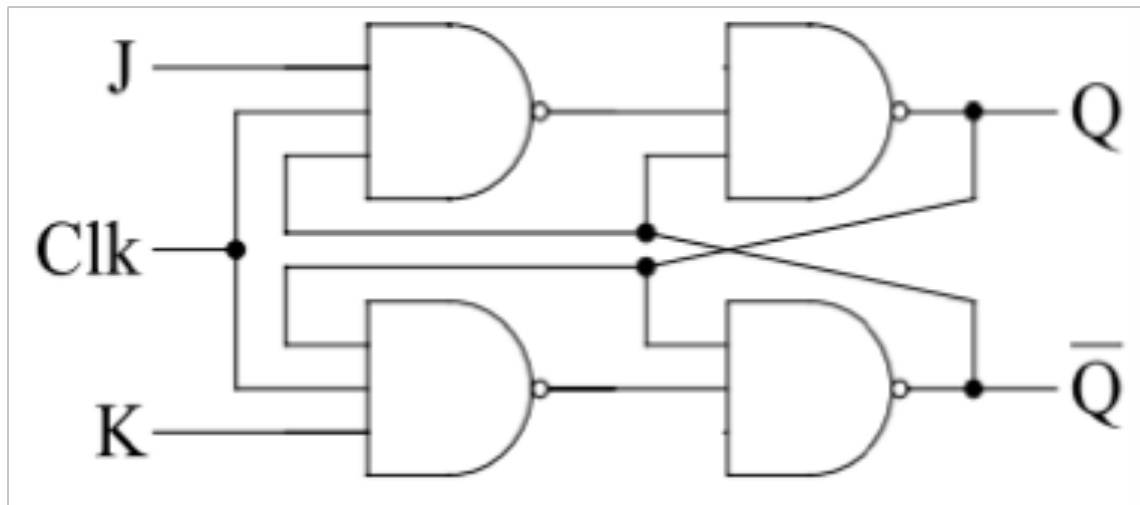


Schéma détaillé

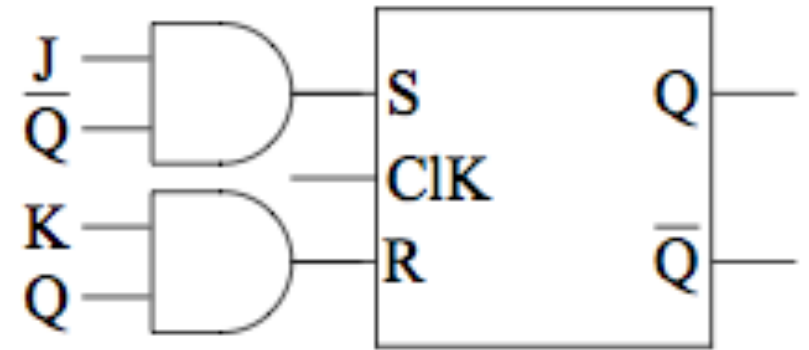
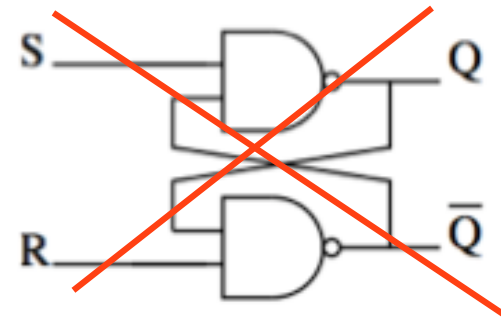
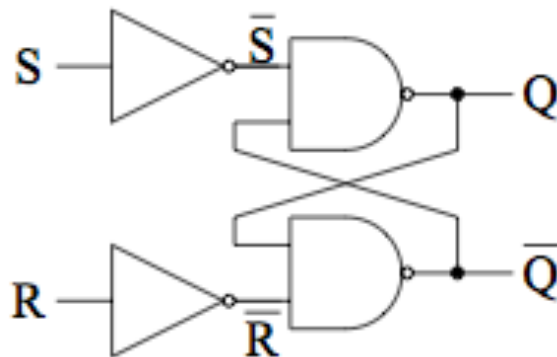


Schéma simplifié

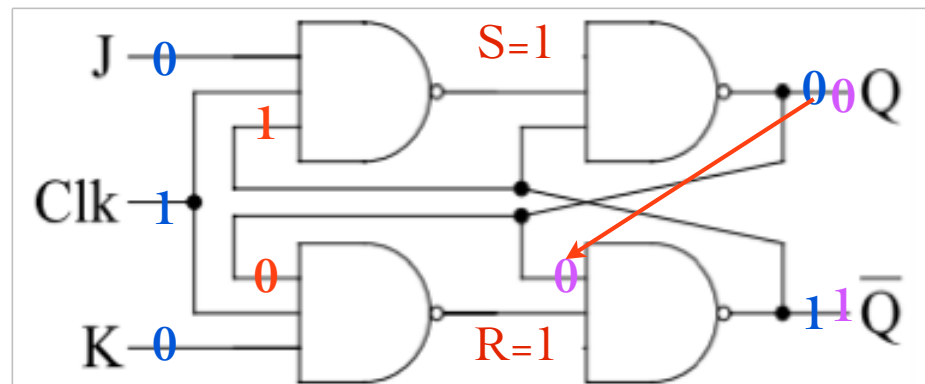
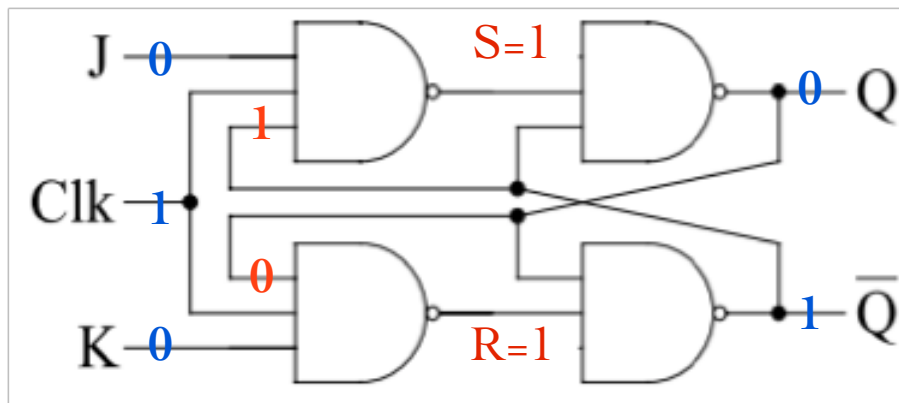
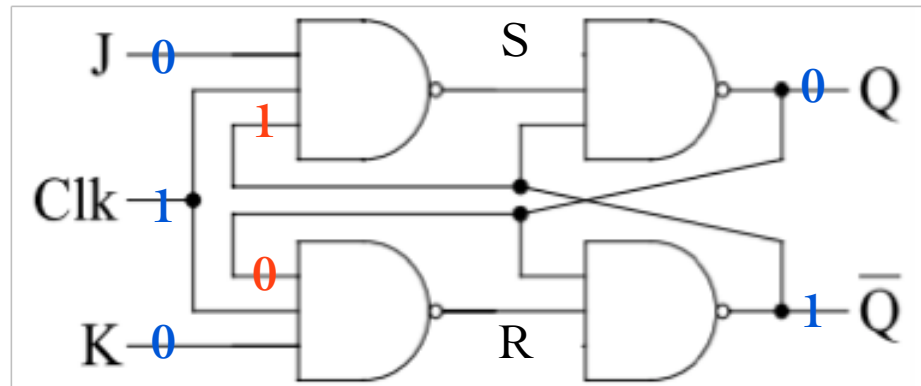
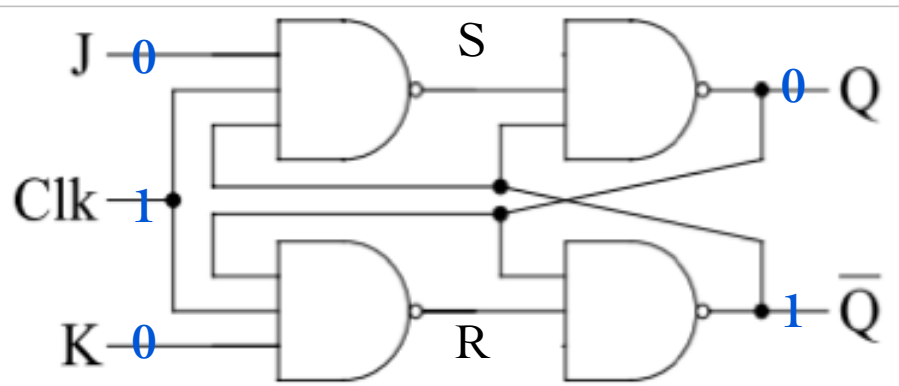
► $S = J\bar{Q}$

► $R = KQ$

► La bascule SR dans le schéma simplifié

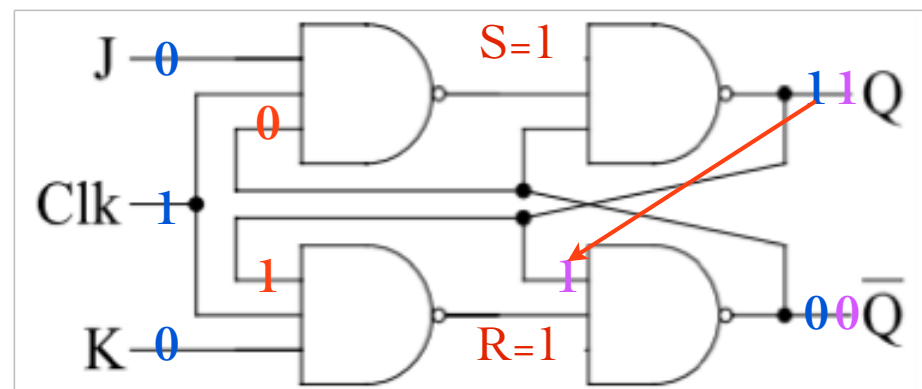
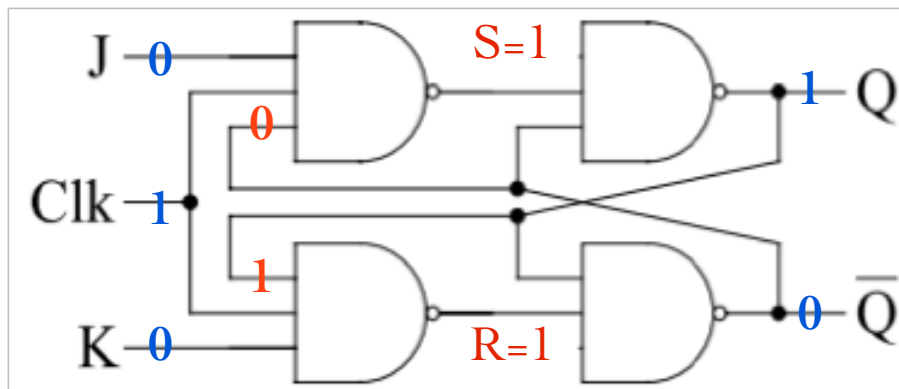
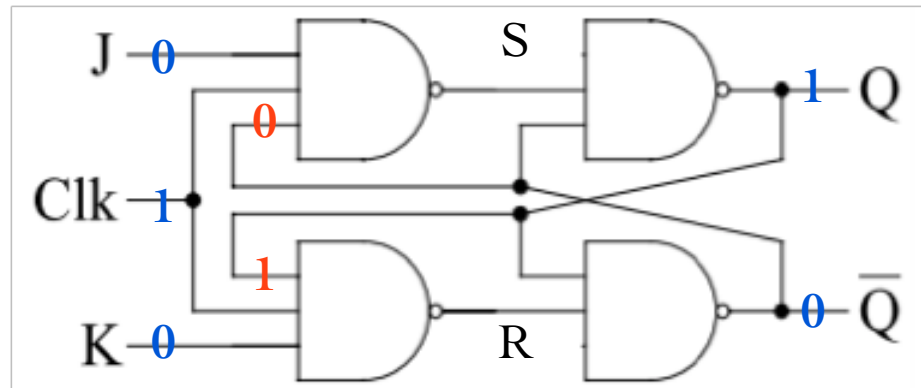
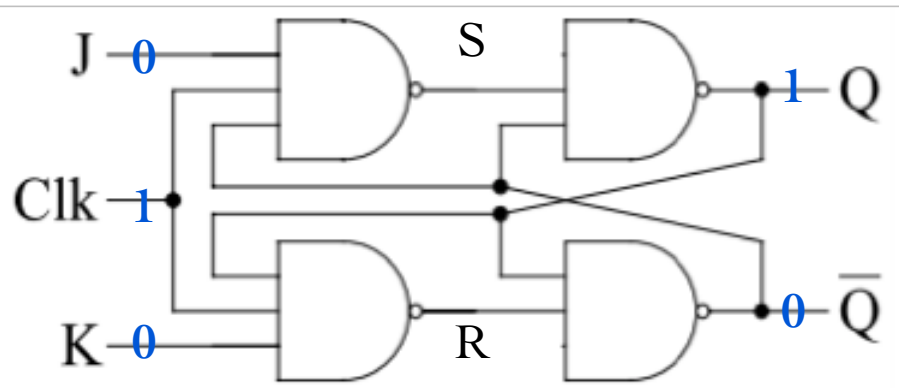


Bascule JK ()



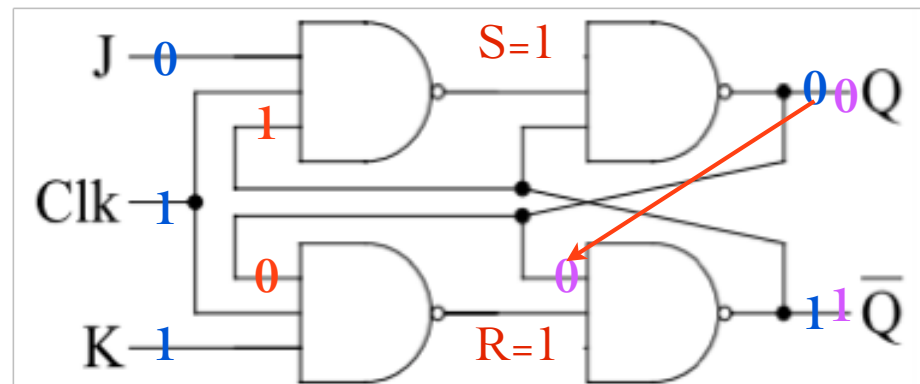
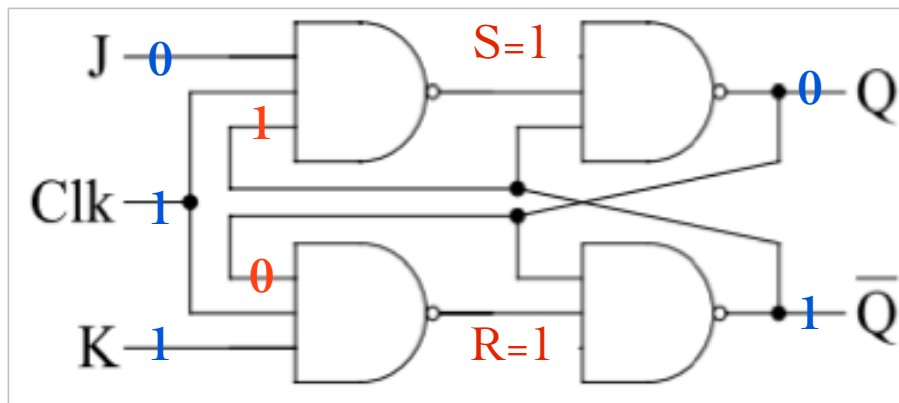
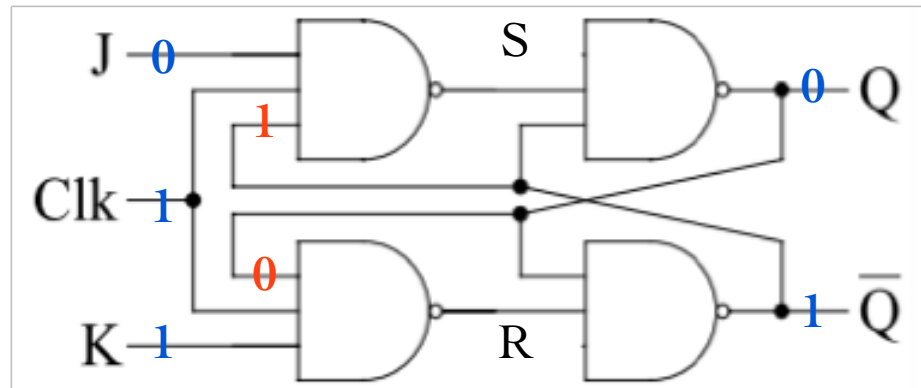
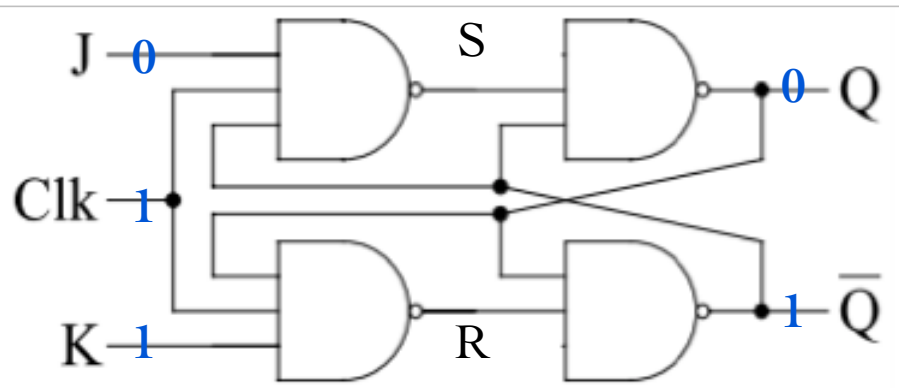
- Pour $JK=00$ et $Q_{(t-1)}=0$,
- $\Rightarrow SR=11$, à l'entrée d'une bascule RS (NAND),
- Donc $Q_{(t)}=0=Q_{(t-1)}$: Etat mémoire (à prouver avec le cas suivant $Q_{(t-1)}=1$)

Bascule JK ()



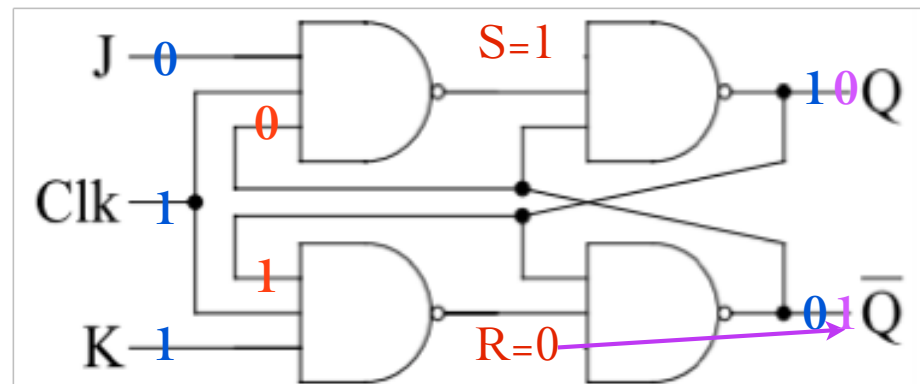
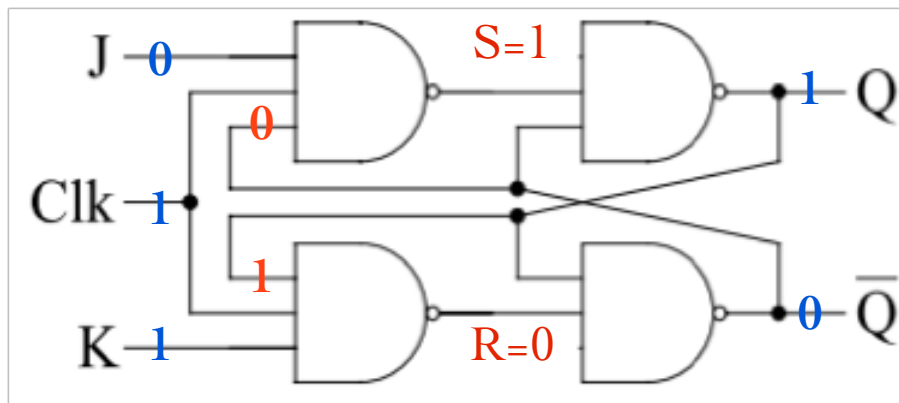
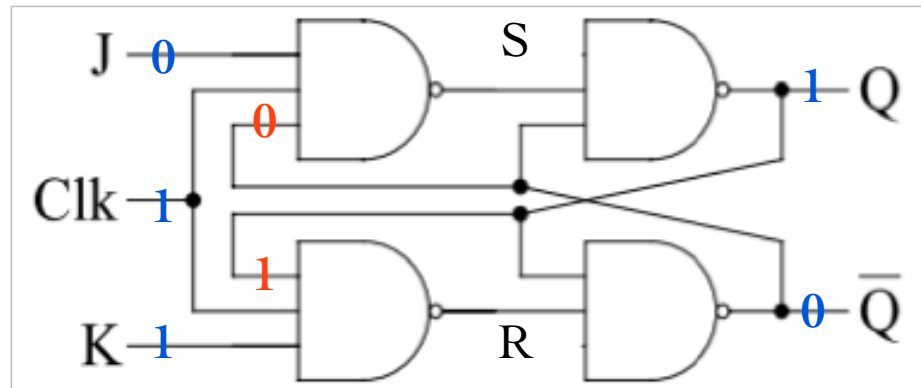
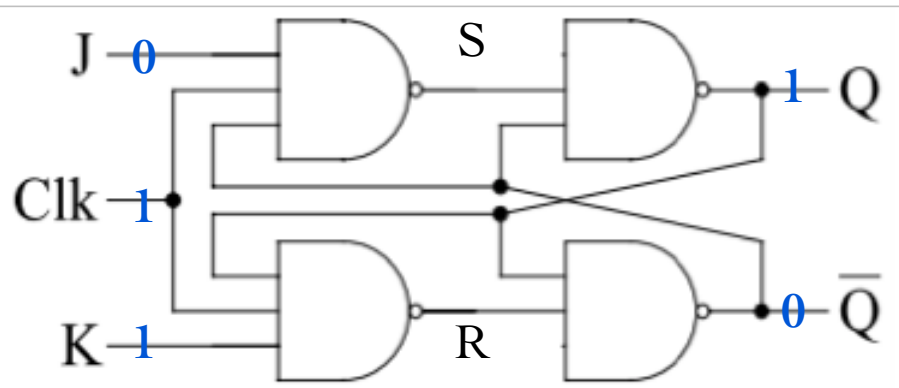
- Pour $JK=00$ et $Q_{(t-1)}=1$,
- $\Rightarrow SR=11$, à l'entrée d'une bascule RS (NAND),
- Donc $Q_{(t)}=1 = Q_{(t-1)}$: Etat mémoire confirmée

Bascule JK ()



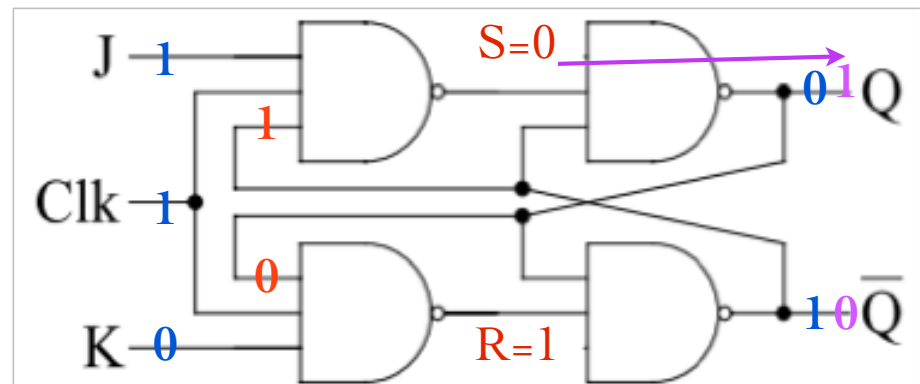
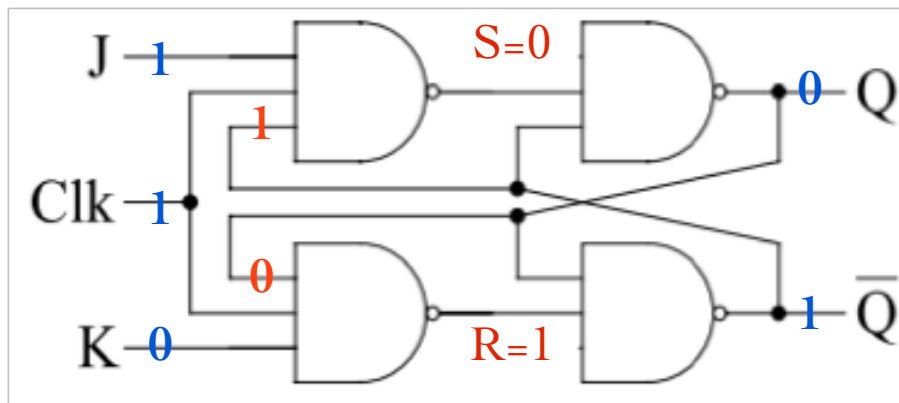
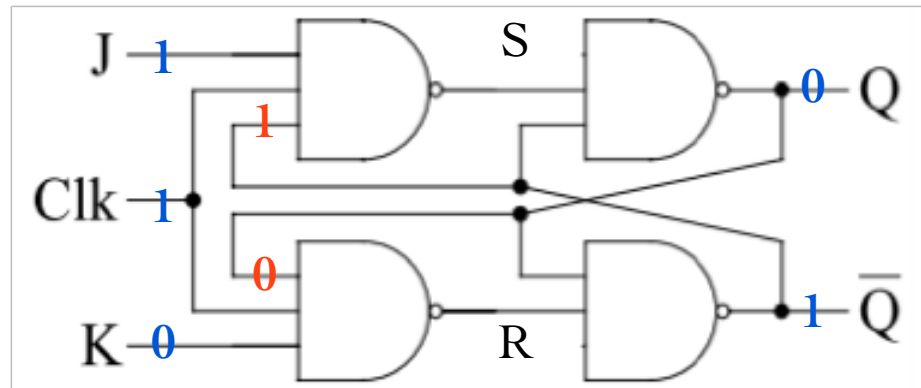
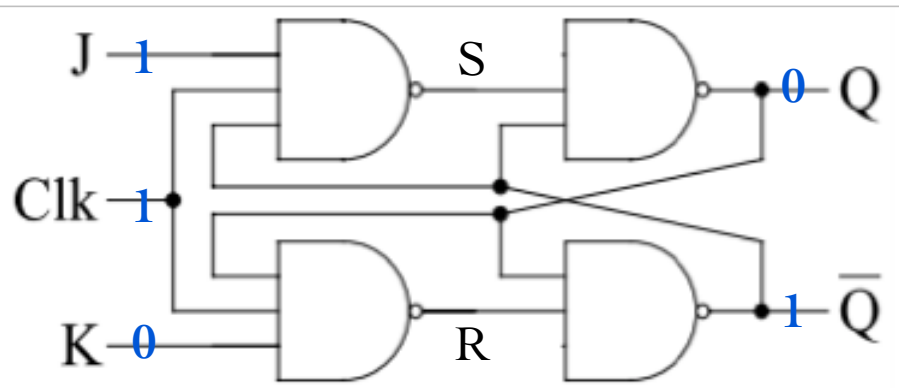
- Pour $JK=01$ et $Q_{(t-1)}=0$,
- $\Rightarrow SR=11$, à l'entrée d'une bascule RS (NAND),
- Donc $Q_{(t)}=0=Q_{(t-1)}$: (à prouver par la suite si c'est un état mémoire)

Bascule JK ()



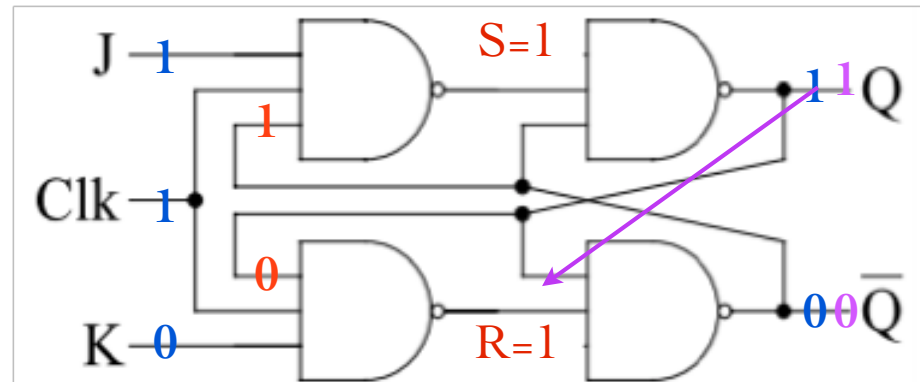
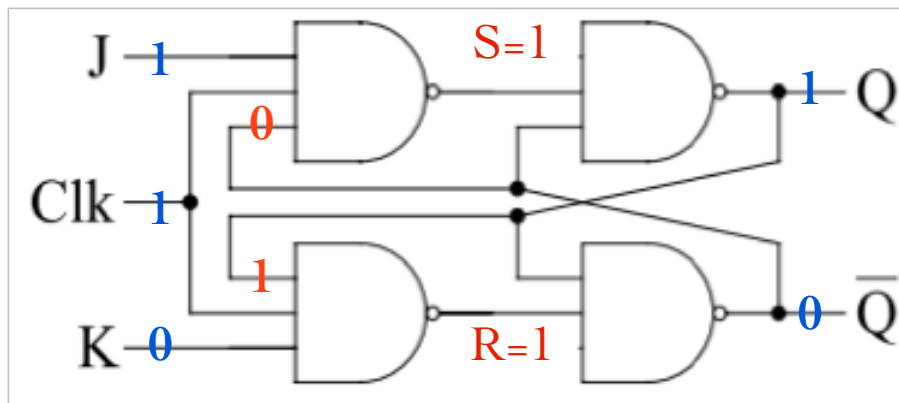
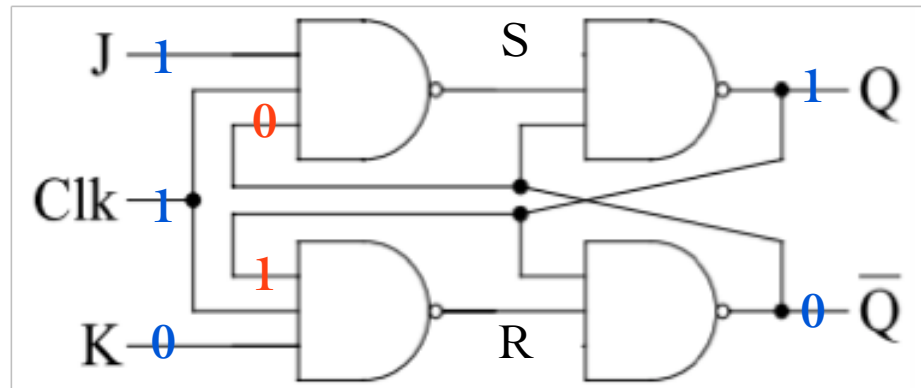
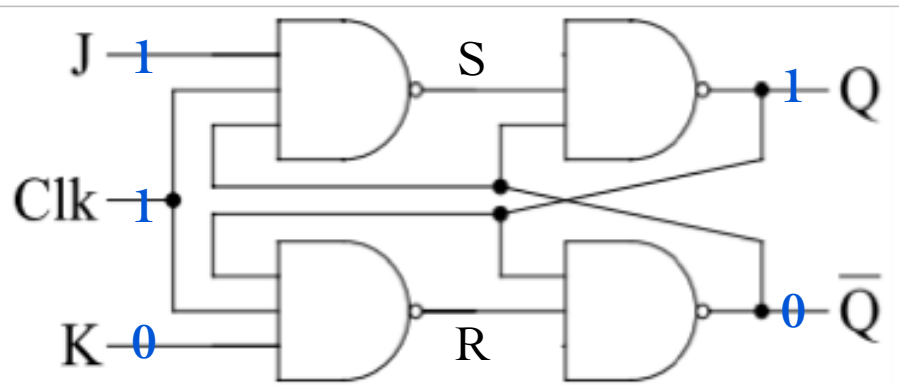
- Pour $JK=01$ et $Q_{(t-1)}=1$,
- $\Rightarrow SR=10$, à l'entrée d'une bascule RS (NAND),
- Donc $Q_{(t)}=0 \neq Q_{(t-1)}$: $Q_{(t)}$ est forcé à 0 quelque soit $Q_{(t-1)}$

Bascule JK ()



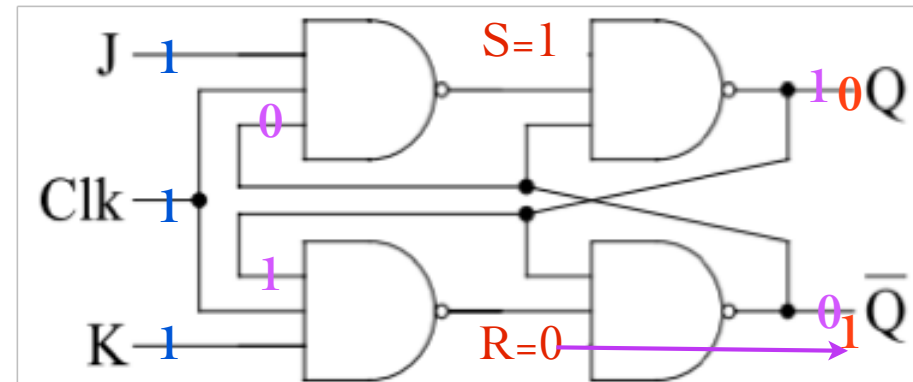
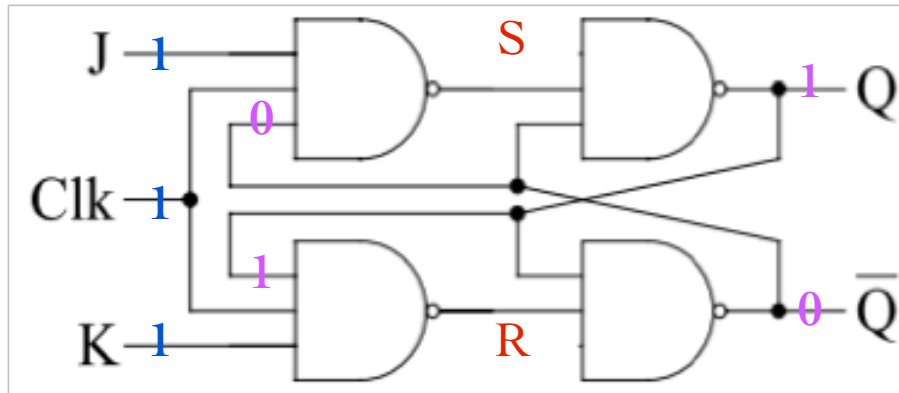
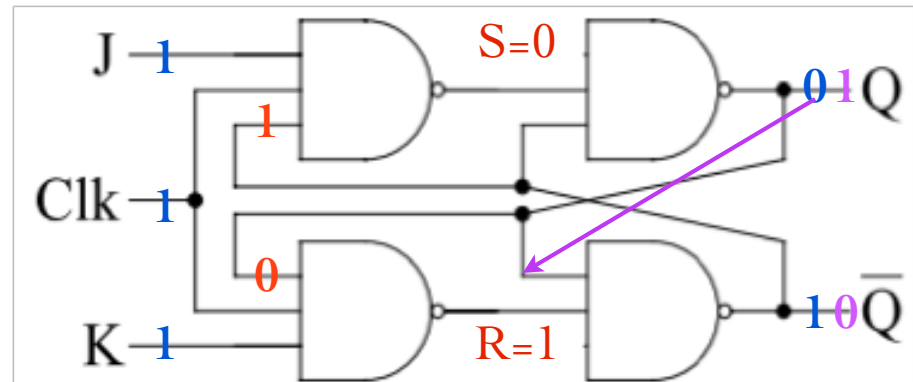
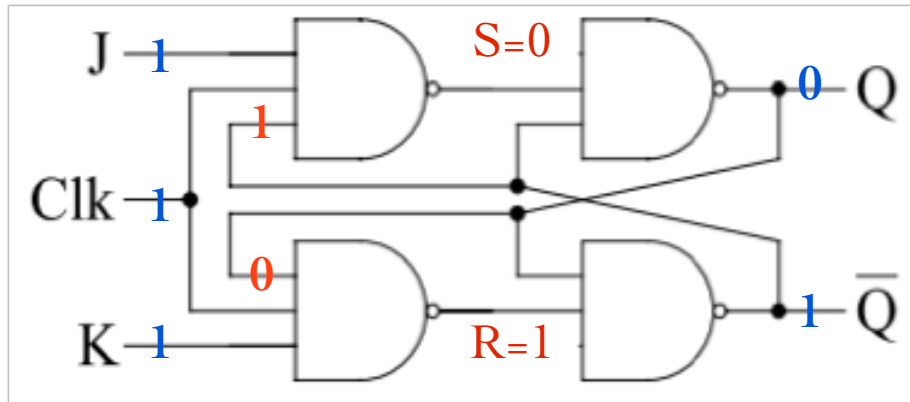
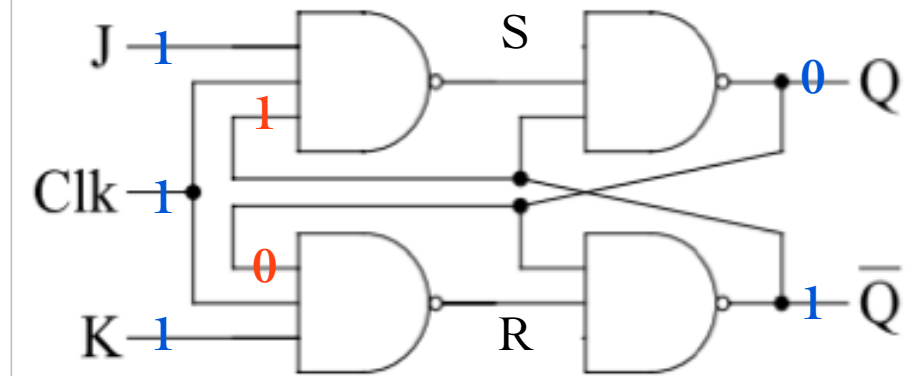
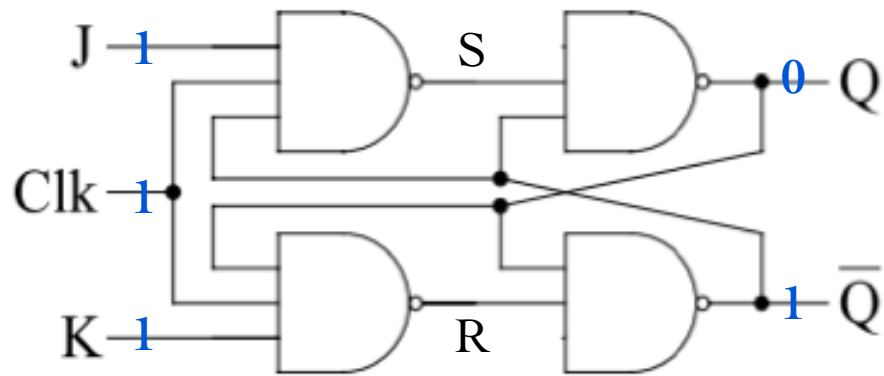
- Pour $JK=10$ et $Q_{(t-1)}=0$,
- $\Rightarrow SR=01$, à l'entrée d'une bascule RS (NAND),
- Donc $Q_{(t)}=1 \neq Q_{(t-1)}$: $Q_{(t)}$ est forcé à 1, à prouver par la suite

Bascule JK ()



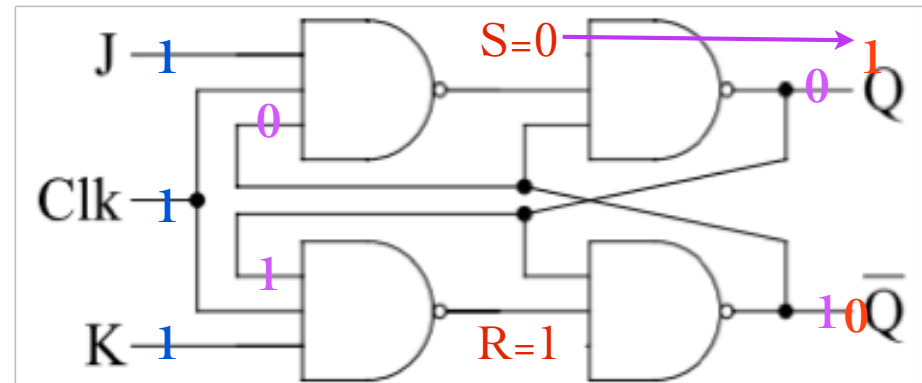
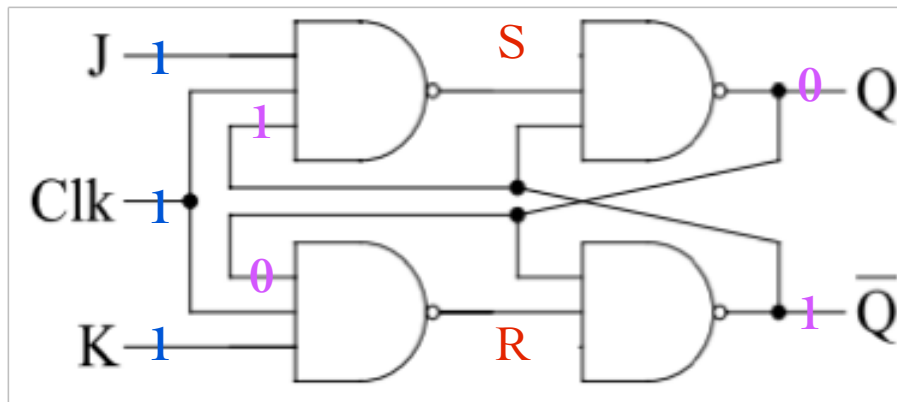
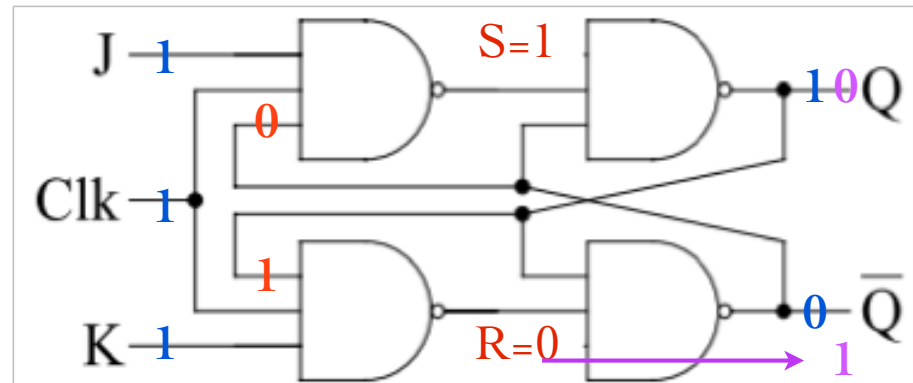
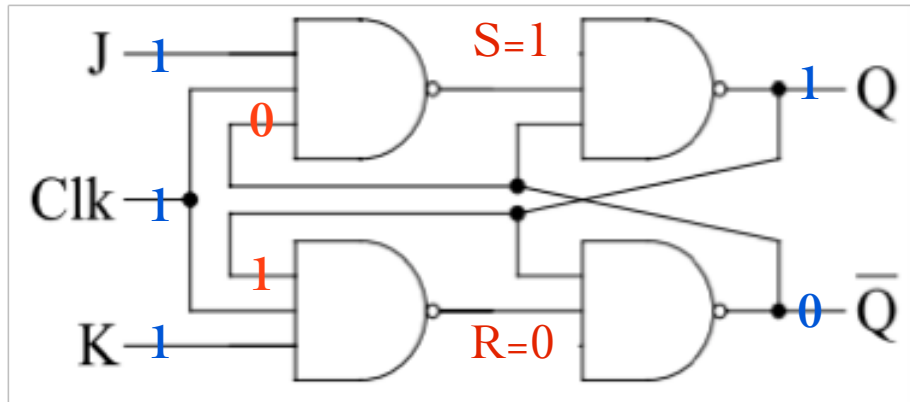
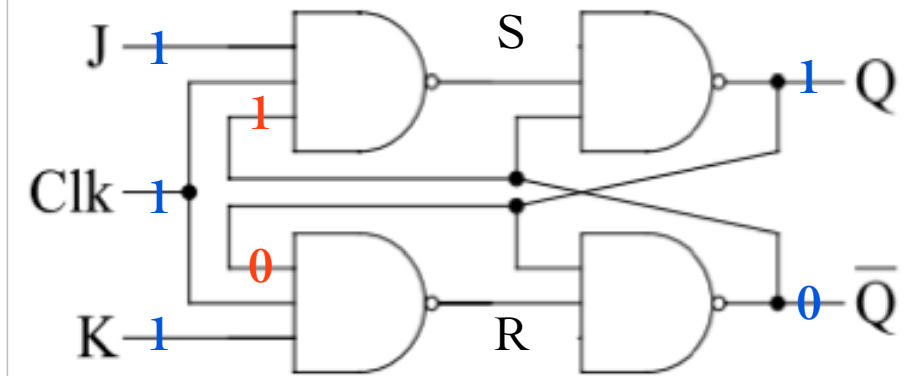
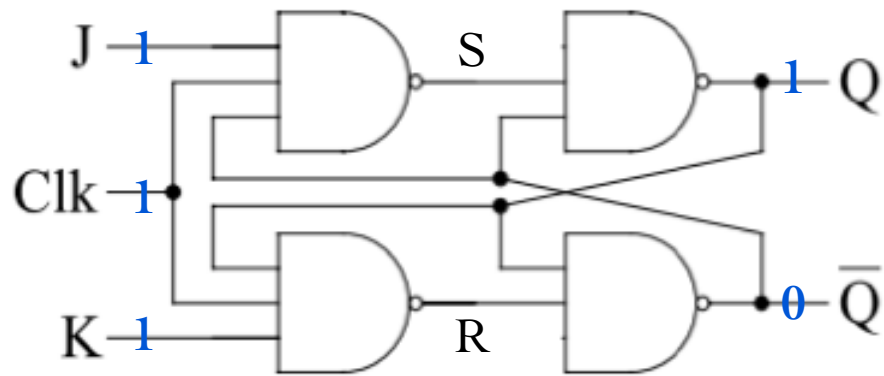
- Pour $JK=10$ et $Q_{(t-1)}=1$,
- $\Rightarrow SR=11$, à l'entrée d'une bascule RS (NAND),
- Donc $Q_{(t)}=1=Q_{(t-1)}$: $Q_{(t)}$ est forcé à 1, quelque soit $Q_{(t-1)}$

Bascule JK ()



- Pour $JK=11$ et $Q_{(t-1)}=0$,
- $\Rightarrow SR=01$, à l'entrée d'une bascule RS (NAND),
- Donc $Q_{(t)}=1 \neq Q_{(t-1)} \Rightarrow SR=10 \Rightarrow Q_{(t)}=0 \neq Q_{(t-1)}$ La sortie Q va osciller entre 0 et 1 pendant toute la durée du signal d'horloge rendant le résultat ambigu

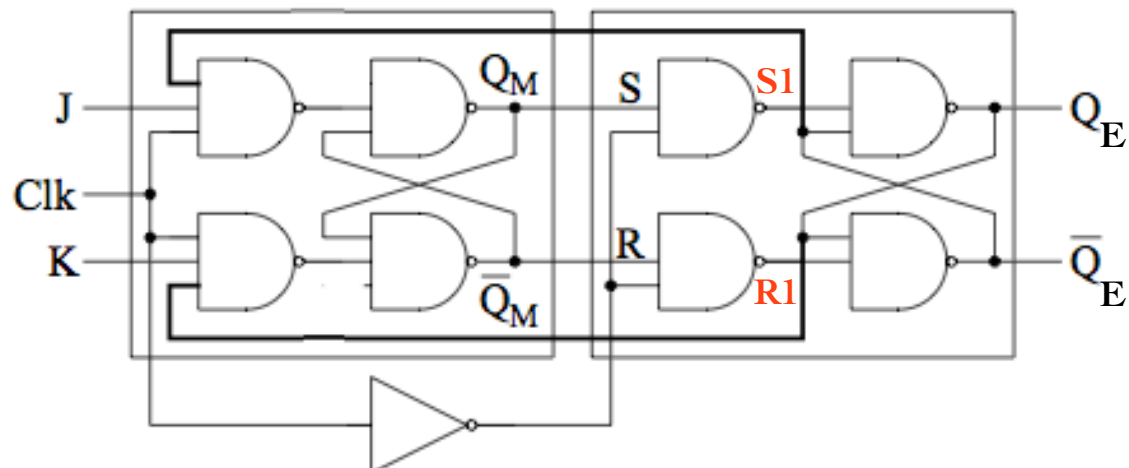
Bascule JK ()



- Pour $JK=11$ et $Q_{(t-1)}=1$,
- $\Rightarrow SR=10$, à l'entrée d'une bascule RS (NAND),
- Donc $Q_{(t)}=0 \neq Q_{(t-1)} \Rightarrow SR=01 \Rightarrow Q_{(t)}=1 \neq Q_{(t-1)}$ La sortie Q va osciller entre 0 et 1 pendant toute la durée du signal d'horloge rendant le résultat ambigu

Bascule JK (Maître-Esclave)

► Pour une bascule JK, quand $J=K=1=Clk$ (qui synchronise avec le niveau haut) la sortie Q va osciller entre 0 et 1 pendant toute la durée du signal d'horloge rendant le résultat ambigu. Pour éviter ce problème on monte deux bascules R-S en cascade



► Quand le signal d'horloge à 1 pour le maître et 0 pour l'esclave. L'état $Q_{(t)}$ est donc invariant car l'horloge de l'esclave multiplie par zéro les entrées S et R.

► Si $Q_M=S$, et $\bar{Q}_M=R \Rightarrow S1=R1=1$, $Q_{E(t)}=Q_{E(t-1)} \Rightarrow$ un état mémoire

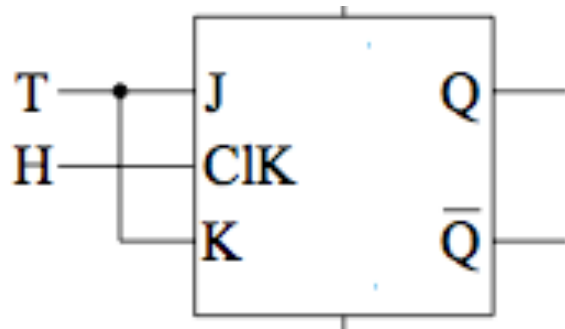
► Lorsque le signal d'horloge passe de 1 à 0 (front descendant), l'état de la bascule maître est transféré à la bascule esclave: le maître est bloqué et l'esclave libéré

► Si $Q_M=1$, $\bar{Q}_M=0 \Rightarrow (S=1 \text{ et } R=0) \Rightarrow Q=1$, $\bar{Q}=0$

► Si $Q_M=0$, $\bar{Q}_M=1 \Rightarrow (S=0 \text{ et } R=1) \Rightarrow \bar{Q}=1$, $Q=0$

Bascule T à partir de la JK

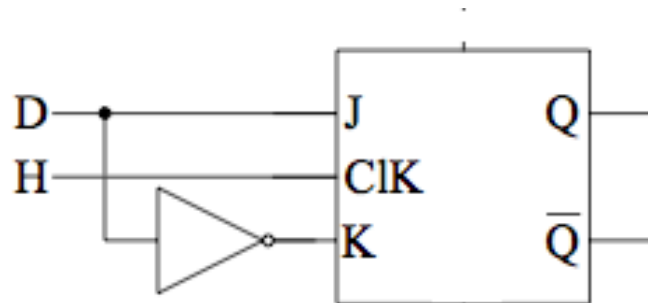
- La bascule T peut également être obtenue à partir d'une bascule JK, on injecte la même entrée sur J et K.



J	K	Q (t)
0	0	Q(t-1)
0	1	0
1	0	1
1	1	$\bar{Q}(t-1)$

Bascule D à partir de la JK

- ▶ La bascule D peut également être obtenue à partir d'une bascule JK, on envoie simultanément:
 - ▶ La donnée sur l'entrée J
 - ▶ Et son inverse sur l'entrée K



J	K	Q (t)
0	0	$Q(t-1)$
0	1	0
1	0	1
1	1	$\bar{Q}(t-1)$