

Architecture des ordinateurs

Chapitre 2 : D. Circuits Séquentiels

L2 Génie Industriel

Motivations

Dans un circuit combinatoire, **les valeurs de sorties**, à un instant donné, sont **imposées par celles des entrées**.

- ⇒ Traitement des données uniquement accessibles immédiatement
- ⇒ La valeur de la sortie ne dépend que de l'entrée et pas de ce qui s'est passé auparavant
- ⇒ Uniquement applicable aux problèmes sans besoin de mémorisation
- ⇒ On sait traiter et manipuler l'information, comment la mémoriser ?



Circuits séquentiels (= circuits logiques à mémoire)

Circuit séquentiel

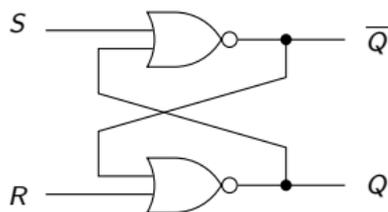
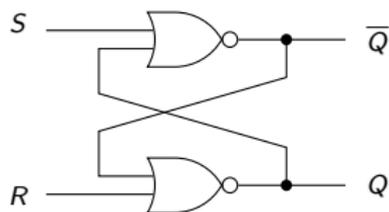
- Circuit logique capable de **mémoriser des informations**
- **Variables internes** en plus des variables d'entrée
- Ensemble des informations mémorisées = **état** du circuit
- Mémoires = circuits spéciaux pouvant stocker un ou plusieurs bits
 - ▶ Les bascules
 - ▶ Les bascules *latch*
 - ▶ Les bascules *flip-flop*
 - ▶ Les registres
- Modification des informations mémorisées \Rightarrow modification de l'état du circuit

- Pour la mémorisation d'un bit
- "Se souviennent" de la valeur que le circuit a enregistrée
- Construction avec une ou deux portes logiques NON-OU (ou NON-ET)
 - ▶ Une ou plusieurs entrées
 - ▶ Une ou deux sorties
- La sortie maintient son état même après disparition du signal de commande
⇒ **logique séquentielle**
- Remarque : on suppose que le temps est *discret* (on peut le découper)

Bascule RS

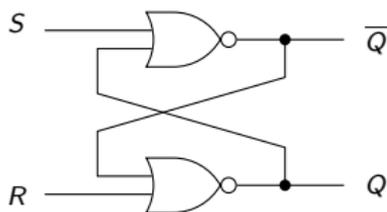
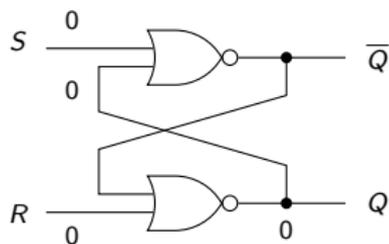
- Deux entrées :
 - ▶ S (*Set*) pour la mise à l'état 1 de la bascule
 - ▶ R (*Reset*) pour la mise à l'état 0 de la bascule
- Deux variables de sortie : Q et \overline{Q}
- La valeur de sortie Q_n à l'instant $t = n$ dépend :
 - ▶ des variables d'entrées
 - ▶ de la valeur antérieure de la sortie (Q_{n-1})

Bascule RS : états stables



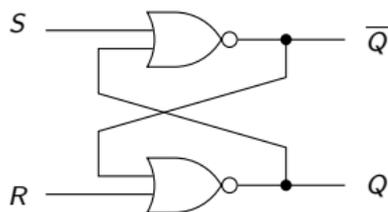
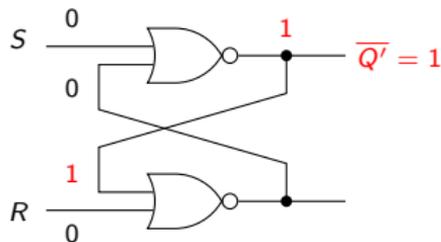
- Cas 1 : On suppose que $S = R = Q = 0 \Rightarrow \bar{Q}' = ?$ et $Q' = ?$
- Cas 2 : On suppose que $S = R = 0$ et $Q = 1 \Rightarrow \bar{Q}' = ?$ et $Q' = ?$

Bascule RS : états stables



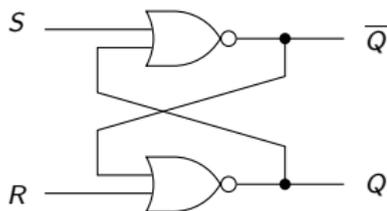
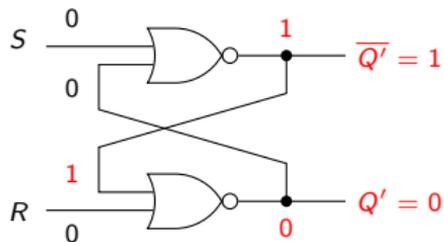
- Cas 1 : On suppose que $S = R = Q = 0 \Rightarrow \bar{Q}' = ?$ et $Q' = ?$
- Cas 2 : On suppose que $S = R = 0$ et $Q = 1 \Rightarrow \bar{Q}' = ?$ et $Q' = ?$

Bascule RS : états stables



- Cas 1 : On suppose que $S = R = Q = 0 \Rightarrow \overline{Q}' = ?$ et $Q' = ?$
- Cas 2 : On suppose que $S = R = 0$ et $Q = 1 \Rightarrow \overline{Q}' = ?$ et $Q' = ?$

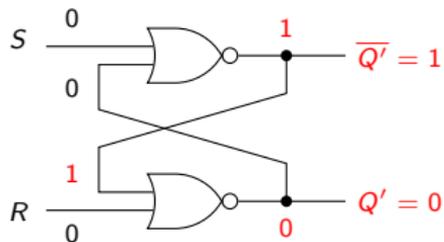
Bascule RS : états stables



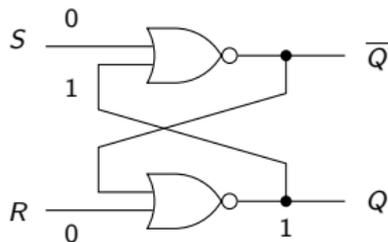
Bascule RS à l'état 0

- Cas 1 : On suppose que $S = R = Q = 0 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
- Cas 2 : On suppose que $S = R = 0$ et $Q = 1 \Rightarrow \overline{Q'} = ?$ et $Q' = ?$

Bascule RS : états stables

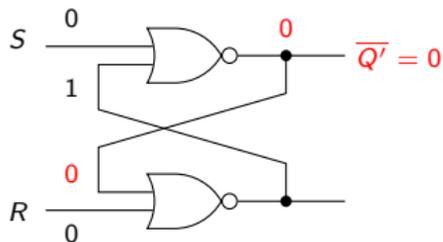
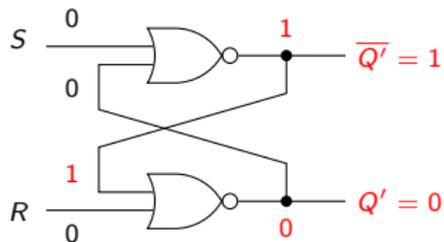


Bascule RS à l'état 0



- Cas 1 : On suppose que $S = R = Q = 0 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
- Cas 2 : On suppose que $S = R = 0$ et $Q = 1 \Rightarrow \overline{Q'} = ?$ et $Q' = ?$

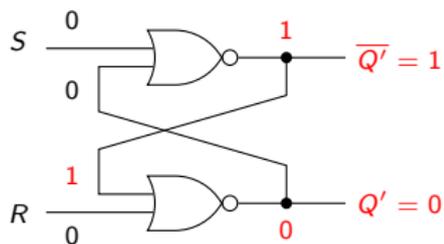
Bascule RS : états stables



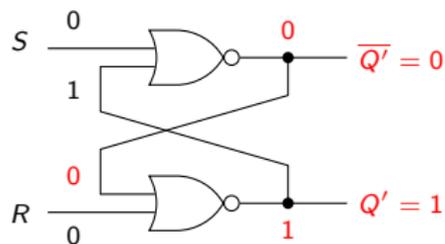
Bascule RS à l'état 0

- Cas 1 : On suppose que $S = R = Q = 0 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
- Cas 2 : On suppose que $S = R = 0$ et $Q = 1 \Rightarrow \overline{Q'} = ?$ et $Q' = ?$

Bascule RS : états stables



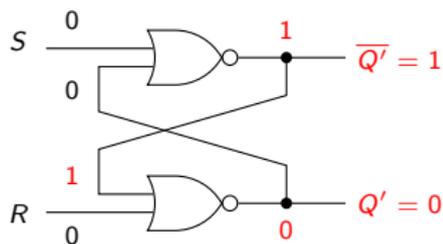
Bascule RS à l'état 0



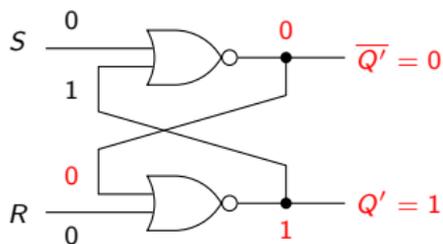
Bascule RS à l'état 1

- Cas 1 : On suppose que $S = R = Q = 0 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
- Cas 2 : On suppose que $S = R = 0$ et $Q = 1 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$

Bascule RS : états stables



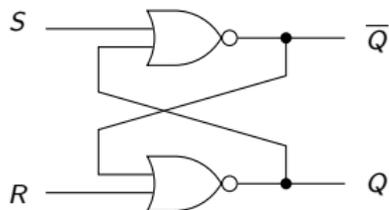
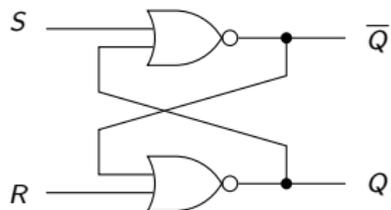
Bascule RS à l'état 0



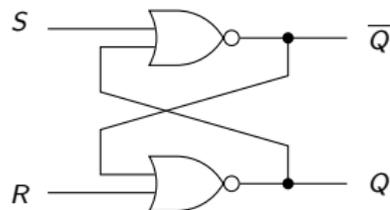
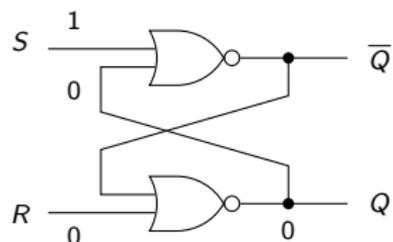
Bascule RS à l'état 1

- Cas 1 : On suppose que $S = R = Q = 0 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
 - Cas 2 : On suppose que $S = R = 0$ et $Q = 1 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- ⇒ Les deux sorties Q' et $\overline{Q'}$ ne peuvent pas être simultanément à 0
- ⇒ Les deux sorties Q' et $\overline{Q'}$ ne peuvent pas être simultanément à 1
- ⇒ Pour $S = R = 0$, la bascule offre **deux états stables qui dépendent de Q**

Bascule RS : activation

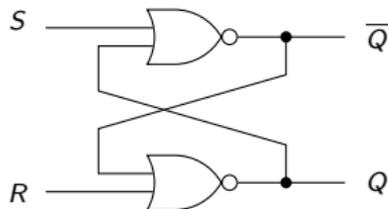
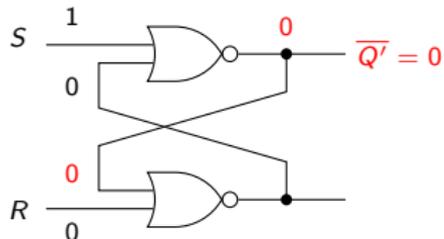


Bascule RS : activation



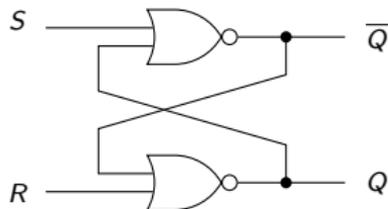
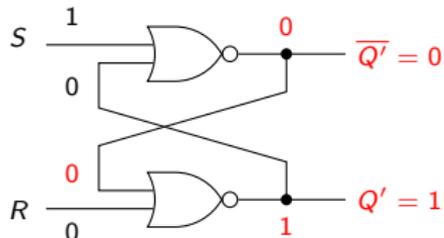
- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \bar{Q}' = ?$ et $Q' = ?$

Bascule RS : activation



- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = ?$ et $Q' = ?$

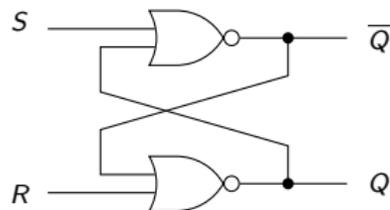
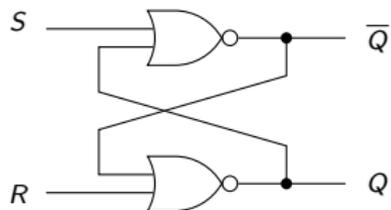
Bascule RS : activation



Bascule RS à l'état 1

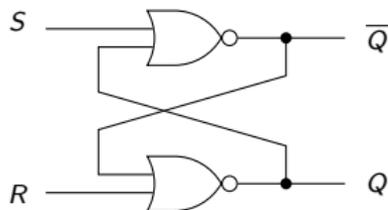
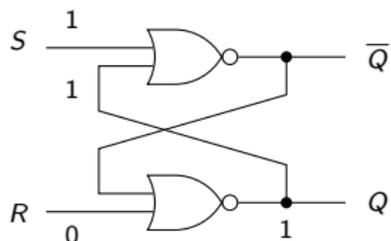
- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$

Bascule RS : activation



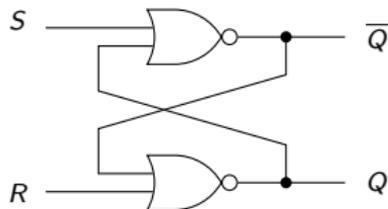
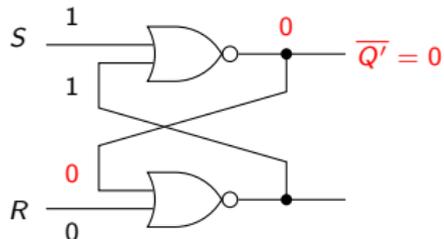
- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \bar{Q}' = 0$ et $Q' = 1$

Bascule RS : activation



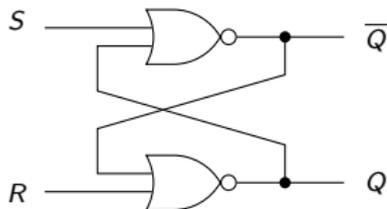
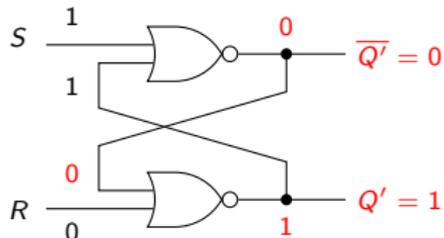
- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \bar{Q}' = 0$ et $Q' = 1$
- Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \bar{Q}' = ?$ et $Q' = ?$

Bascule RS : activation



- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \overline{Q'} = ?$ et $Q' = ?$

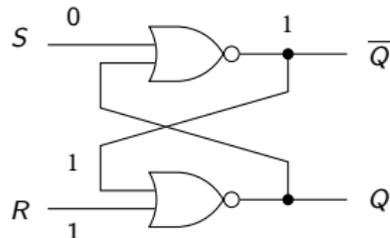
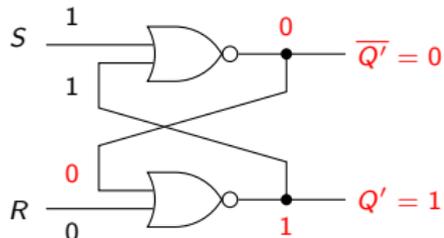
Bascule RS : activation



Bascule RS à l'état 1

- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$

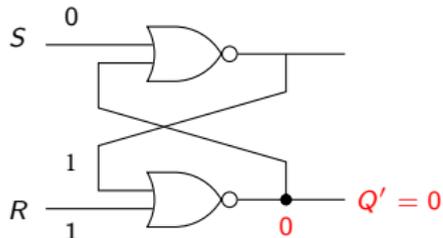
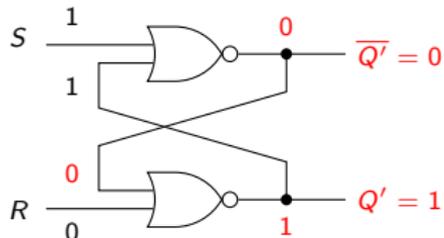
Bascule RS : activation



Bascule RS à l'état 1

- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 4.1 : On suppose que $S = Q = 0$ et $R = 1 \Rightarrow \overline{Q'} = ?$ et $Q' = ?$

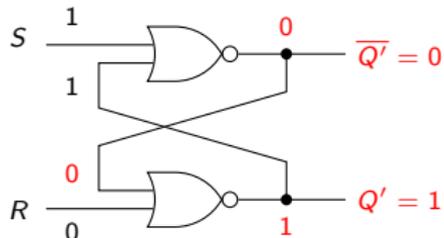
Bascule RS : activation



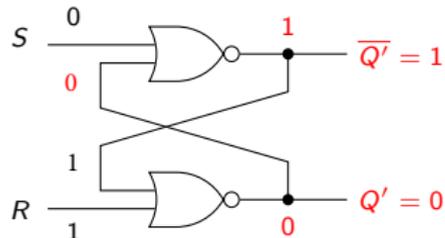
Bascule RS à l'état 1

- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 4.1 : On suppose que $S = Q = 0$ et $R = 1 \Rightarrow \overline{Q'} = ?$ et $Q' = ?$

Bascule RS : activation



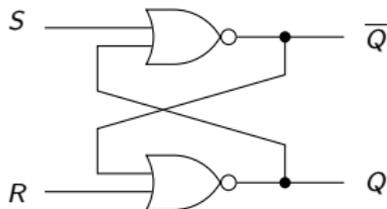
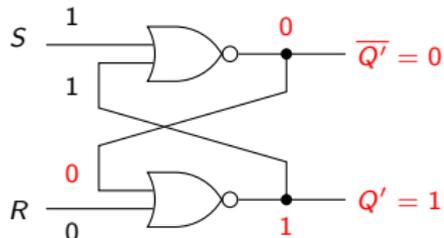
Bascule RS à l'état 1



Bascule RS à l'état 0

- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 4.1 : On suppose que $S = Q = 0$ et $R = 1 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$

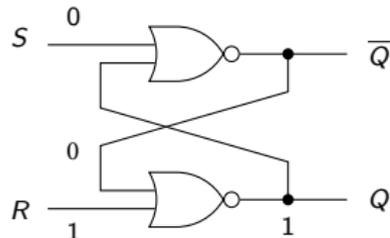
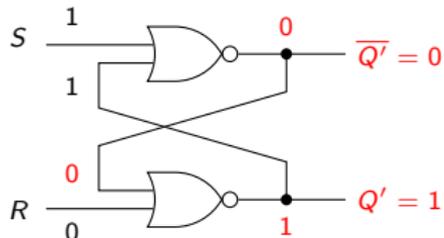
Bascule RS : activation



Bascule RS à l'état 1

- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 4.1 : On suppose que $S = Q = 0$ et $R = 1 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$

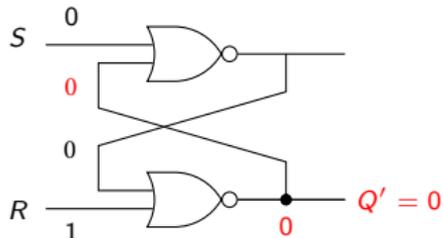
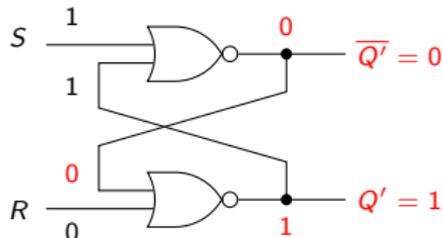
Bascule RS : activation



Bascule RS à l'état 1

- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 4.1 : On suppose que $S = Q = 0$ et $R = 1 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
- Cas 4.2 : On suppose que $S = 0$ et $R = Q = 1 \Rightarrow \overline{Q'} = ?$ et $Q' = ?$

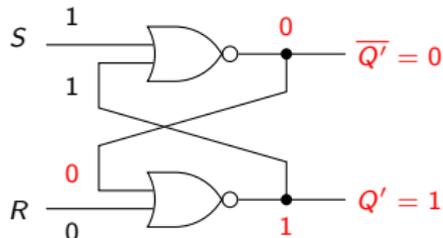
Bascule RS : activation



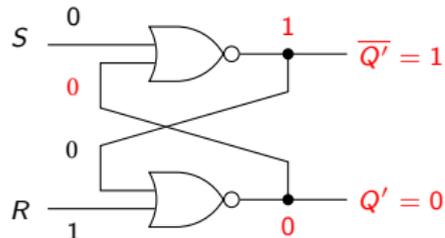
Bascule RS à l'état 1

- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 4.1 : On suppose que $S = Q = 0$ et $R = 1 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
- Cas 4.2 : On suppose que $S = 0$ et $R = Q = 1 \Rightarrow \overline{Q'} = ?$ et $Q' = ?$

Bascule RS : activation



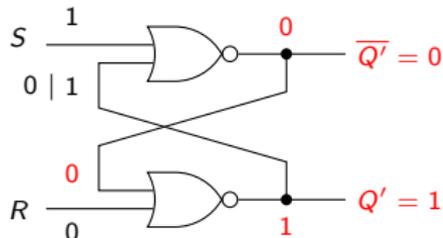
Bascule RS à l'état 1



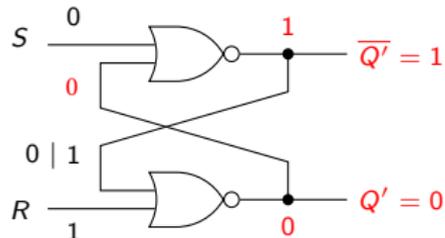
Bascule RS à l'état 0

- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
- Cas 4.1 : On suppose que $S = Q = 0$ et $R = 1 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
- Cas 4.2 : On suppose que $S = 0$ et $R = Q = 1 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$

Bascule RS : activation



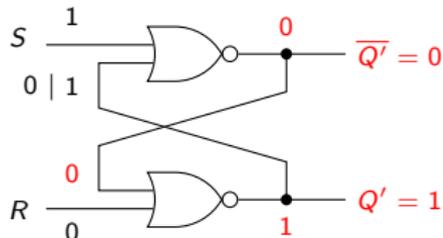
Bascule RS à l'état 1



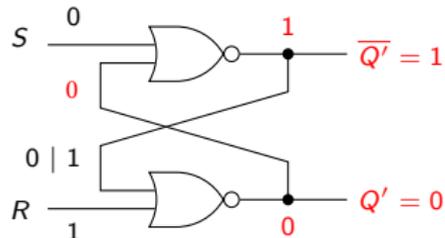
Bascule RS à l'état 0

- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
 - Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
 - Cas 4.1 : On suppose que $S = Q = 0$ et $R = 1 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
 - Cas 4.2 : On suppose que $S = 0$ et $R = Q = 1 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
- ⇒ Si $S = 1$, la bascule RS passe (ou se maintient) à la valeur $Q' = 1$
- ⇒ Si $R = 1$, la bascule RS passe (ou se maintient) à la valeur $Q' = 0$

Bascule RS : activation



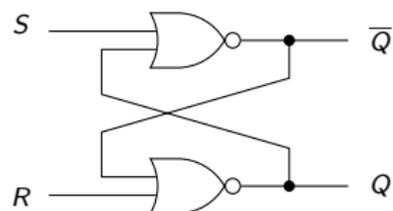
Bascule RS à l'état 1



Bascule RS à l'état 0

- Cas 3.1 : On suppose que $S = 1$ et $R = Q = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
 - Cas 3.2 : On suppose que $S = Q = 1$ et $R = 0 \Rightarrow \overline{Q'} = 0$ et $Q' = 1$
 - Cas 4.1 : On suppose que $S = Q = 0$ et $R = 1 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
 - Cas 4.2 : On suppose que $S = 0$ et $R = Q = 1 \Rightarrow \overline{Q'} = 1$ et $Q' = 0$
- ⇒ Si $S = 1$, la bascule RS passe (ou se maintient) à la valeur $Q' = 1$
- ⇒ Si $R = 1$, la bascule RS passe (ou se maintient) à la valeur $Q' = 0$
- ⇒ **Une bascule RS “se souvient” de l'action antérieure de R ou S**

Bascule RS : Table de vérité



S	R	Q	\bar{Q}	Q'	\bar{Q}'
0	0	0	0	x	x
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	x	x
0	1	0	0	x	x
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	x	x
1	0	0	0	x	x
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	x	x
1	1	0	0	x	x
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	x	x

Bascule RS : Résumé

S	R	Q	\overline{Q}	État de la bascule
0	0	Q	\overline{Q}	<i>Sorties inchangées</i>
0	1	0	1	<i>RESET : remise à 0</i>
1	0	1	0	<i>SET : mise à 1</i>
1	1	0	0	<i>Non utilisé (état instable)</i>

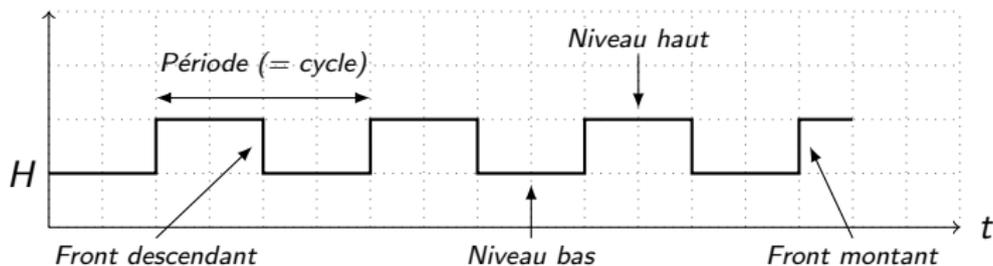
La bascule RS mémorise la valeur des entrées : sa sortie dépend de la dernière entrée mise à 1 (R ou S)

Horloge (1/2)

- L'ordre d'apparition des variables revêt une importance souvent cruciale.
 - La conception des systèmes logiques dépend si une variable arrive avant l'autre ou bien si elles arrivent en même temps.
- ⇒ Besoin de respecter des relations de séquentialité contraignantes
- ⇒ Utilisation **d'horloge** (*base de temps* ou *système de cadencement*)

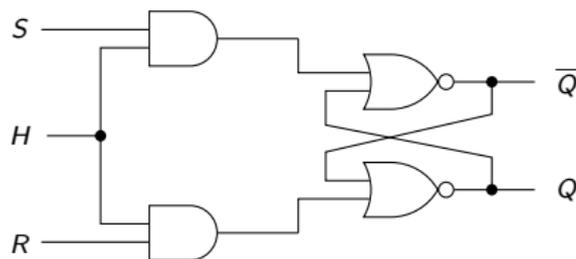
Horloge (2/2)

- Système logique qui émet **régulièrement** une suite d'impulsions calibrées
 - Intervalle de temps entre deux impulsions = *temps de cycle* ou *période* de l'horloge
 - Fréquence des impulsions comprise entre 1 et 100 MHz
- ⇒ Temps de cycle compris entre 10 ns à 10 μ s



Bascule RSH (*latch*)

- Bascule RS + Horloge
- Permet de faire **changer d'état à la bascule à un instant t précis**
- S_n et R_n : états des entrées à l'instant $t = n$
- Q_{n+1} : sortie au prochain cycle d'horloge (instant $t = n + 1$)

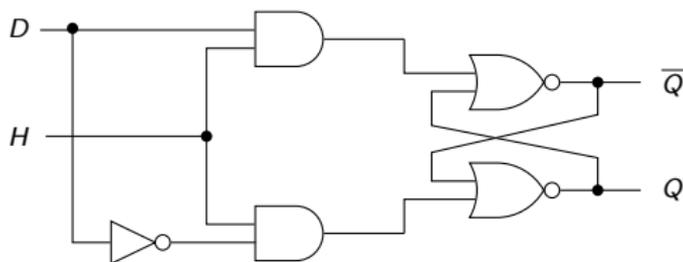


S_n	R_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?

$$Q_{n+1} = S + \bar{R}Q_n$$

Bascule D (*latch*)

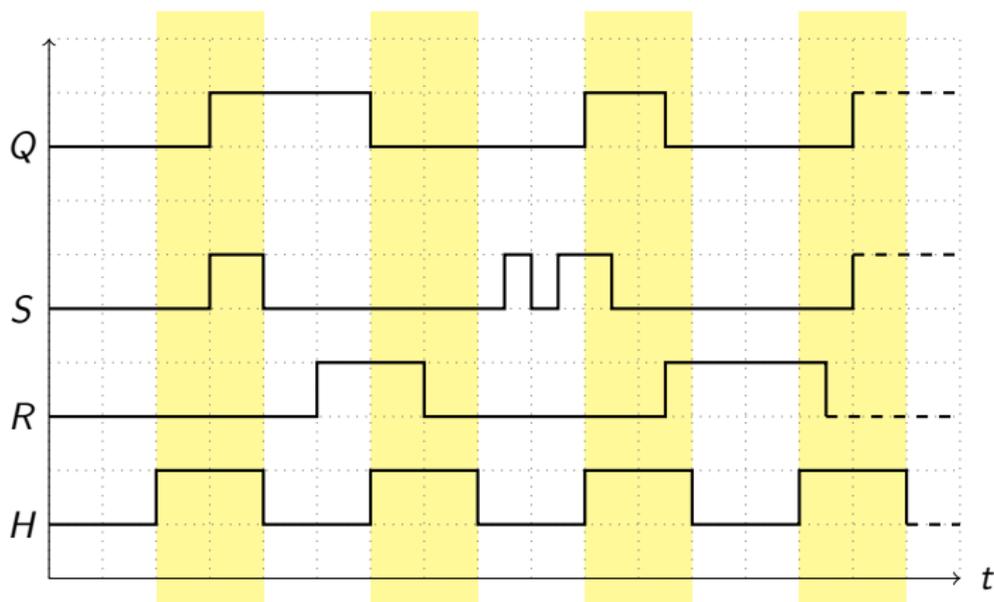
- Pour résoudre l'ambiguïté propre à la bascule RS (quand $S = R = 1$)
- Fait en sorte que l'état correspondant à $S = R = 1$ ne soit jamais en entrée
- Une seule entrée externe D



Bascule *latch* vs. Bascule *flip-flop* (1/2)

Bascule *latch* :

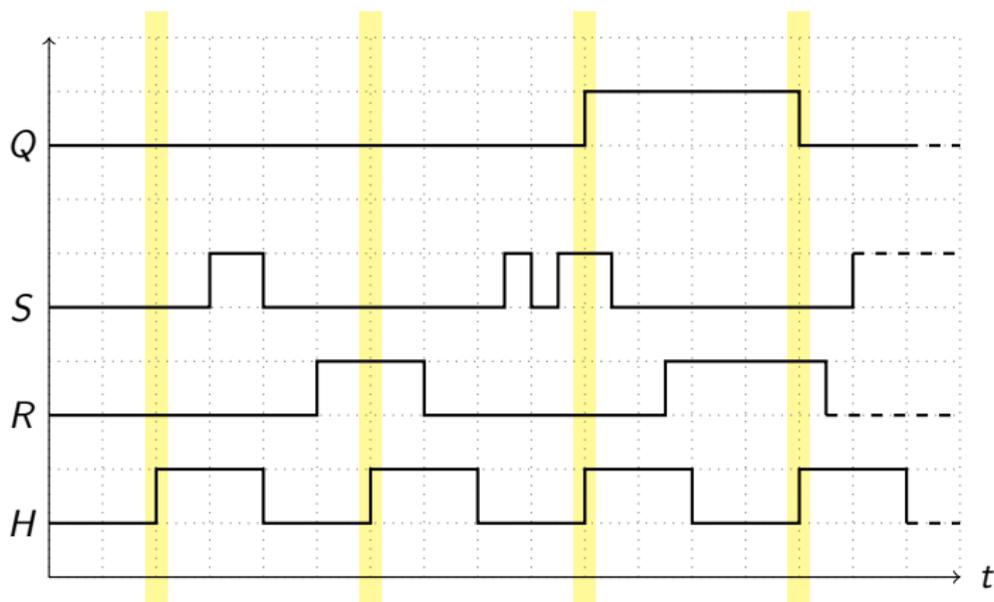
- Bascule **asynchrone**
- Change d'état lorsque **l'horloge est au niveau 1** (= niveau haut)



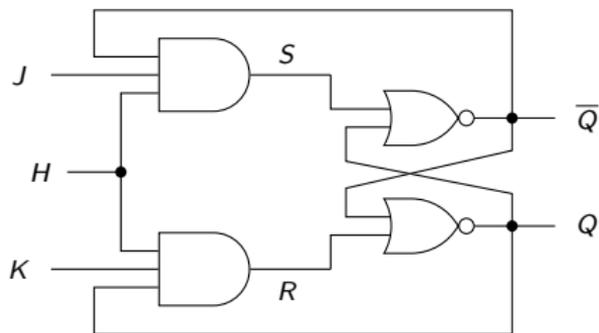
Bascule *latch* vs. Bascule *flip-flop* (2/2)

Bascule *flip-flop* :

- Bascule **synchrone**
- Change d'état lorsque **l'horloge est en front montant**

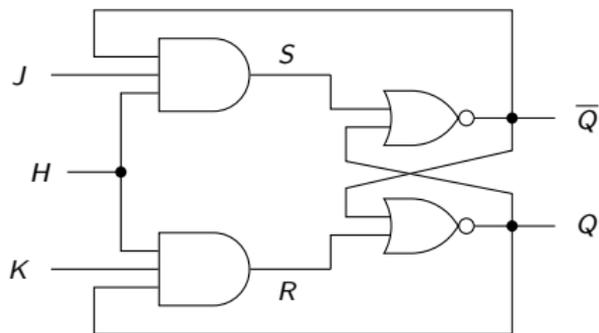


Bascule JK (*flip-flop*)



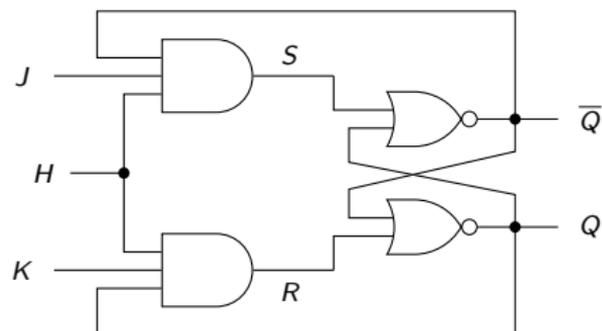
J_n	K_n	Q_n	$\overline{Q_n}$	S	R	Q_{n+1}
0	0	0	1			
0	0	1	0			
0	1	0	1			
0	1	1	0			
1	0	0	1			
1	0	1	0			
1	1	0	1			
1	1	1	0			

Bascule JK (*flip-flop*)



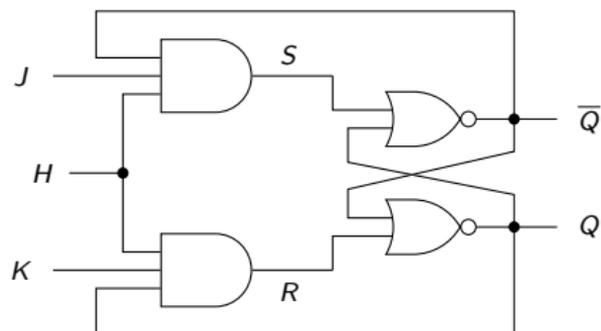
J_n	K_n	Q_n	\bar{Q}_n	S	R	Q_{n+1}
0	0	0	1	0	0	0
0	0	1	0			
0	1	0	1			
0	1	1	0			
1	0	0	1			
1	0	1	0			
1	1	0	1			
1	1	1	0			

Bascule JK (*flip-flop*)



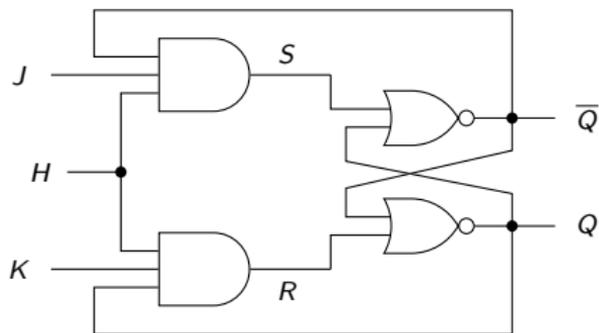
J_n	K_n	Q_n	$\overline{Q_n}$	S	R	Q_{n+1}
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1			
0	1	1	0			
1	0	0	1			
1	0	1	0			
1	1	0	1			
1	1	1	0			

Bascule JK (*flip-flop*)



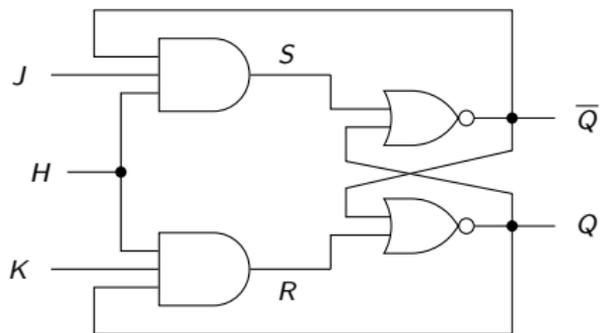
J_n	K_n	Q_n	$\overline{Q_n}$	S	R	Q_{n+1}
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0			
1	0	0	1			
1	0	1	0			
1	1	0	1			
1	1	1	0			

Bascule JK (*flip-flop*)



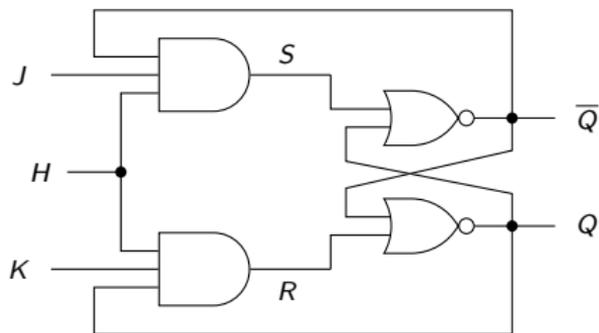
J_n	K_n	Q_n	$\overline{Q_n}$	S	R	Q_{n+1}
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	1	0
1	0	0	1			
1	0	1	0			
1	1	0	1			
1	1	1	0			

Bascule JK (*flip-flop*)



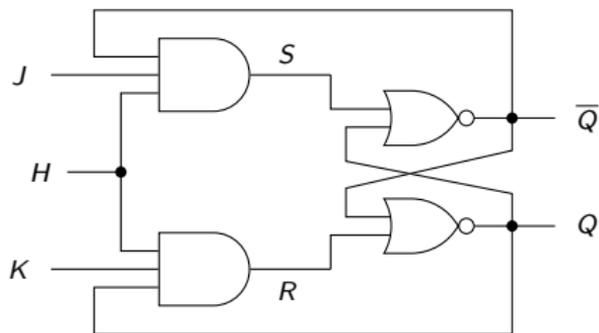
J_n	K_n	Q_n	$\overline{Q_n}$	S	R	Q_{n+1}
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	1	0
1	0	0	1	1	0	1
1	0	1	0			
1	1	0	1			
1	1	1	0			

Bascule JK (*flip-flop*)



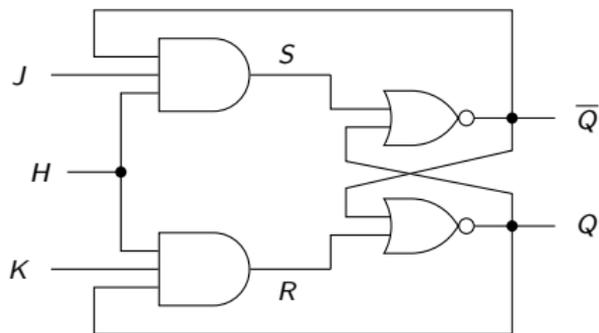
J_n	K_n	Q_n	$\overline{Q_n}$	S	R	Q_{n+1}
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	1	0
1	0	0	1	1	0	1
1	0	1	0	0	0	1
1	1	0	1			
1	1	1	0			

Bascule JK (*flip-flop*)



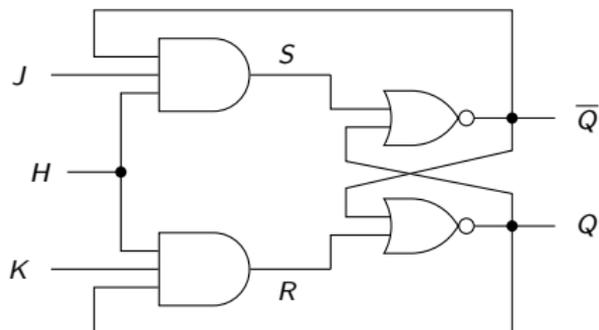
J_n	K_n	Q_n	\bar{Q}_n	S	R	Q_{n+1}
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	1	0
1	0	0	1	1	0	1
1	0	1	0	0	0	1
1	1	0	1	1	0	1
1	1	1	0			

Bascule JK (*flip-flop*)



J_n	K_n	Q_n	\bar{Q}_n	S	R	Q_{n+1}
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	1	0
1	0	0	1	1	0	1
1	0	1	0	0	0	1
1	1	0	1	1	0	1
1	1	1	0	0	1	0

Bascule JK (*flip-flop*)

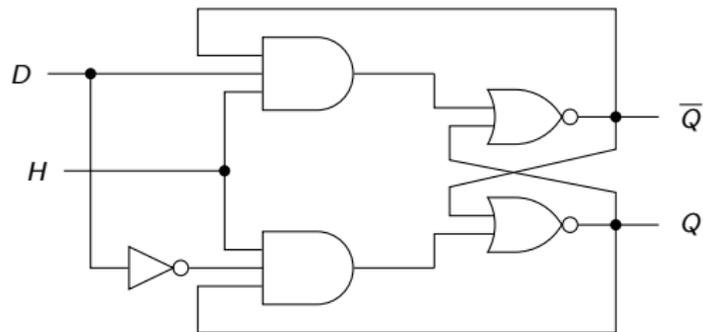


J_n	K_n	Q_n	$\overline{Q_n}$	S	R	Q_{n+1}
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	1	0
1	0	0	1	1	0	1
1	0	1	0	0	0	1
1	1	0	1	1	0	1
1	1	1	0	0	1	0

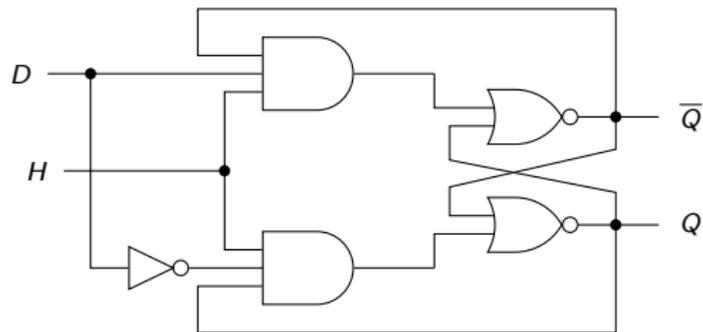
J_n	K_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

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

Bascule D (*flip-flop*)



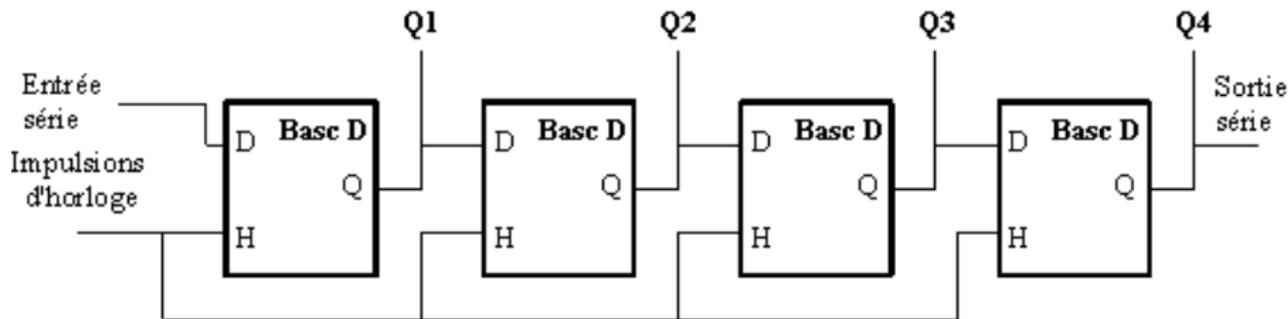
Bascule D (*flip-flop*)



$$Q_{n+1} = D_n$$

Les registres

- Une bascule est l'élément de base de la logique séquentielle
- **Une bascule permet de mémoriser un seul bit**
- Un registre est un ensemble ordonné de n bascules
- **Un registre permet de mémoriser une information sur n bits**



Les types de registres

- Registres à chargement parallèle
- Registres à entrée/sortie série
- Registres à entrée série et sortie parallèle
- Registres à entrée parallèle et sortie série
- Registres à décalage circulaire

Entrées asynchrones

Toutes les bascules précédentes peuvent se voir dotées d'entrées asynchrones permettant de forcer la valeur de la sortie:

- *Preset* : pour mettre la sortie Q à 1.
- *Clear* : pour mettre la sortie Q à 0.
- *Enable* : pour désactiver le signal d'horloge.
- etc...