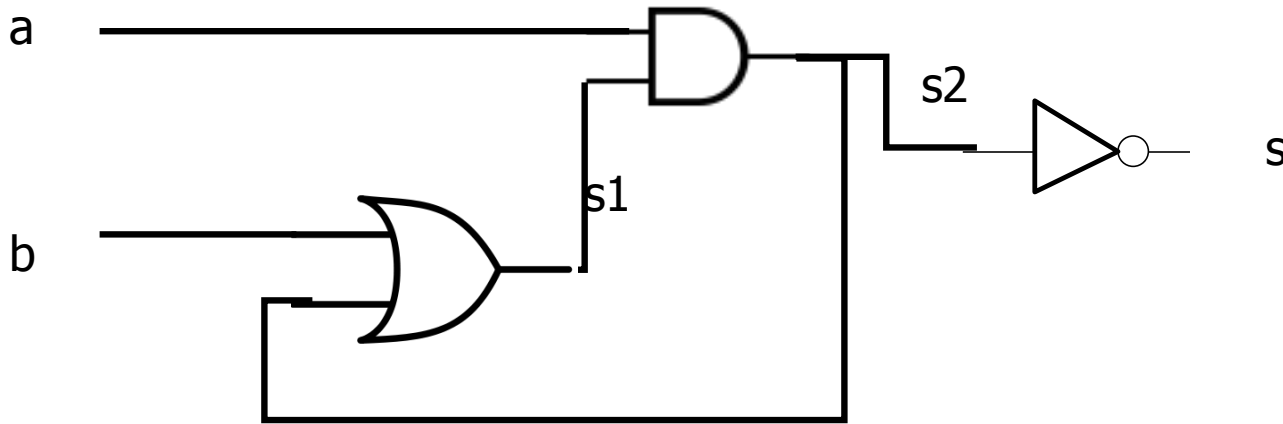


Bascules et éléments mémorisants

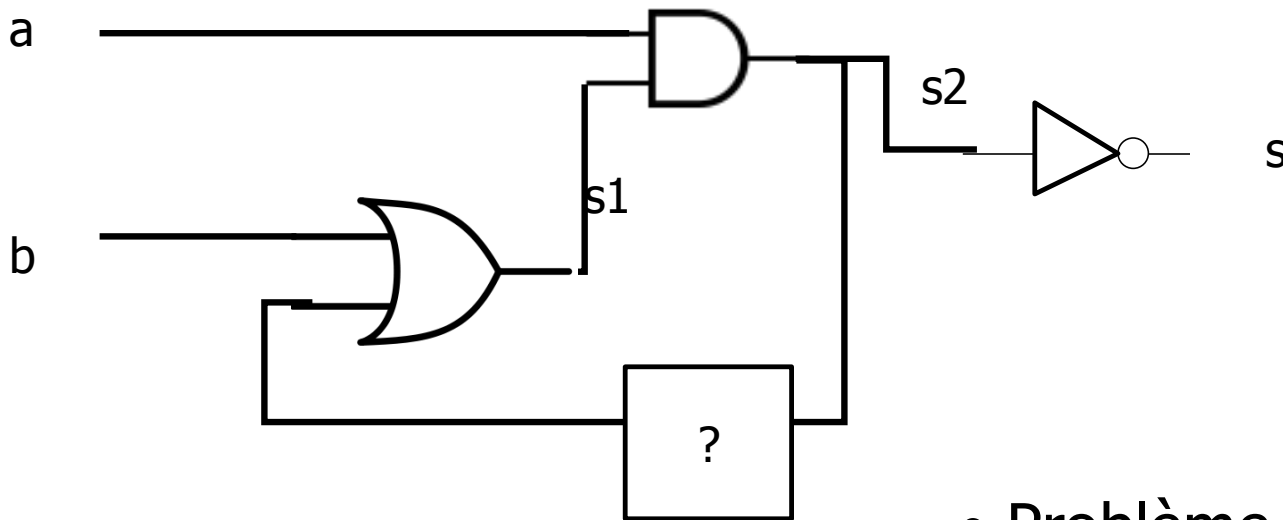
B. Miramond

Exemple de circuit avec *rétroaction*



- Problème de stabilité dans les signaux électriques

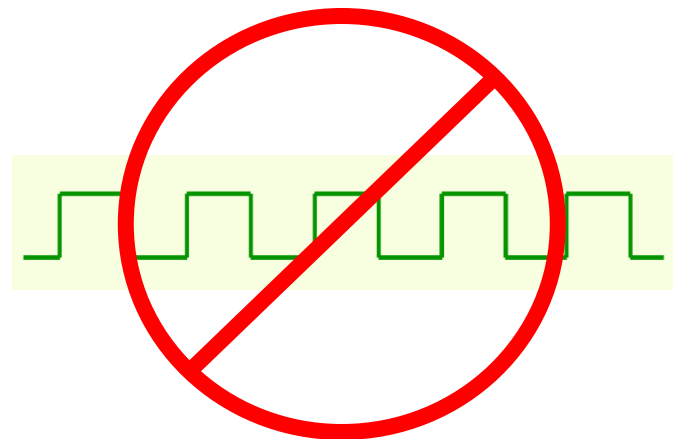
Exemple de circuit avec *rétroaction*



- Problème de stabilité dans les signaux électriques
- Introduction d'éléments mémorisant

Types de circuits séquentiels

- Synchrones – les changements sur les sorties ne sont autorisés qu'à des instants déterminés par une horloge globale.
- Asynchrones – les changements arrivent à chaque changement d'évènement sur les entrées.



Types de circuits séquentiels

- Circuits séquentiels synchrones
 - Tous les signaux sont synchronisés sur l'horloge
 - Les mémoires répondent sur requêtes uniquement sur les fronts d'activation de l'horloge
 - L'unité de mémorisation élémentaire est la bascule synchrone ou **flip-flop**
 - Les circuits peuvent être conçus en utilisant des méthodes de synthèse systématiques

Types de circuits séquentiels

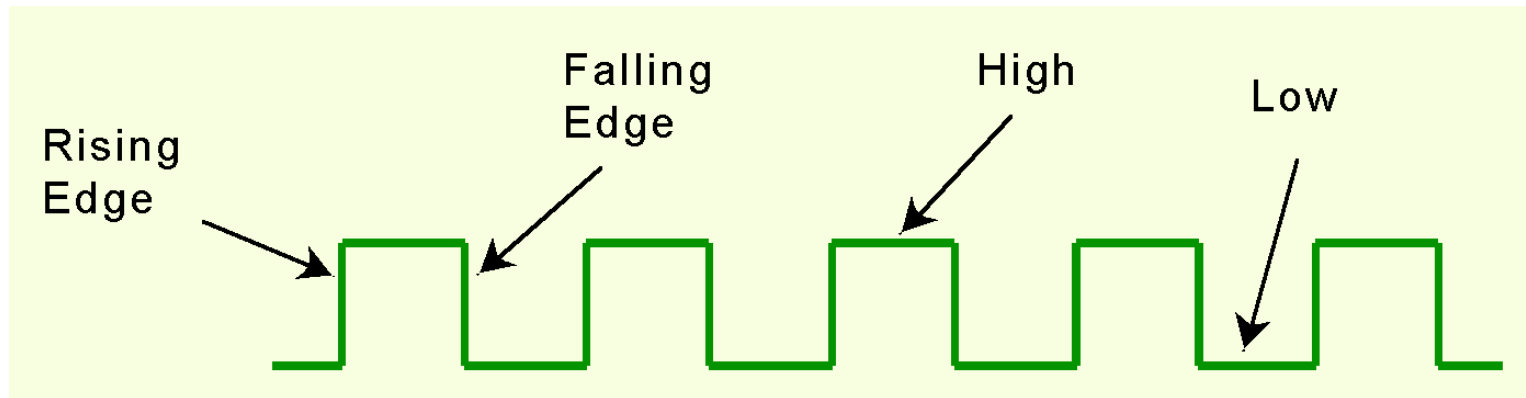
- Circuits séquentiels asynchrones
 - Les sorties dépendent uniquement de l'ordre de changements des entrées
 - Ils sont basés sur les délais de propagation des parties combinatoires
 - L'unité de mémorisation élémentaire est la bascule asynchrone ou **latch**
- Les méthodes de conception synchrones ne s'appliquent pas
- Ces circuits consomment moins mais sont moins systématiques

Stabilité

- Ces éléments de mémorisation (flip-flop et latches) sont des éléments oscillatoires :
 - Astable : non stable, pas d'états stables
 - Monostable : un seul état stable
 - Bistable : deux états stables

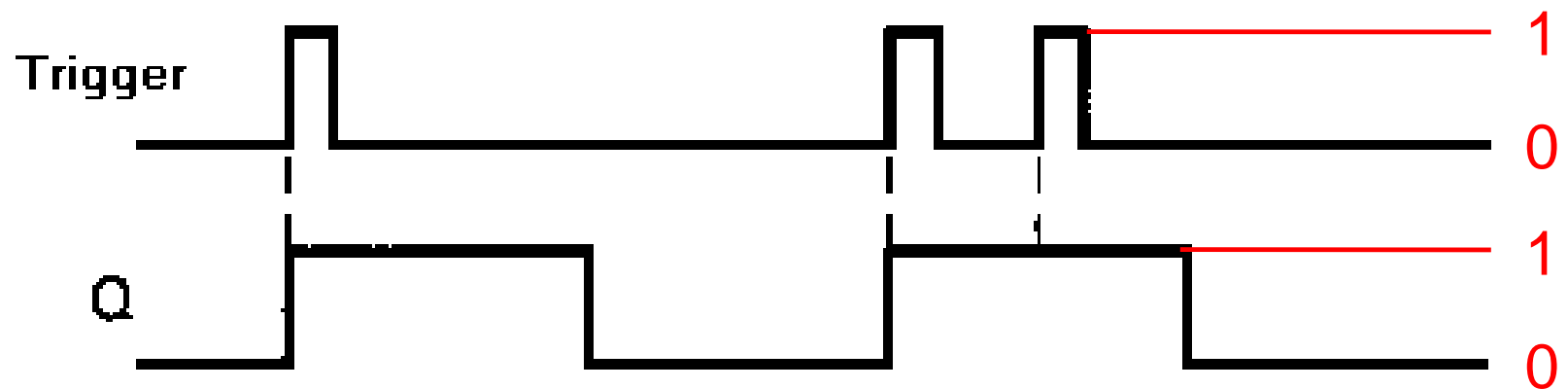
Éléments Astable

- Il s'agit d'oscillateurs qui n'ont pas besoin de signaux externes pour changer d'état
- Comme le signal d'horloge généré par le quartz du circuit



Éléments mono-stable

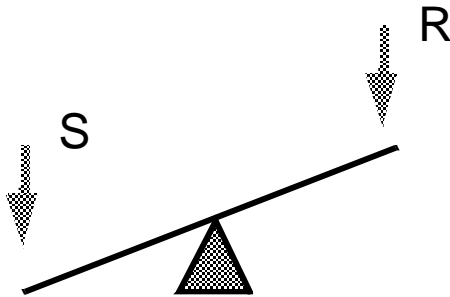
- Un élément mon-stable est déclenché par un signal externe qui le fait changer d'état pendant une durée déterminée avant de revenir naturellement dans son état initial



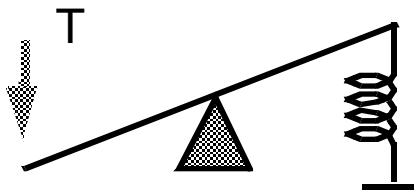
Éléments Bi-stable

- Un élément bi-stable change d'état sur un premier signal de déclenchement
- Et nécessite un second signal de déclenchement pour changer à nouveau d'état

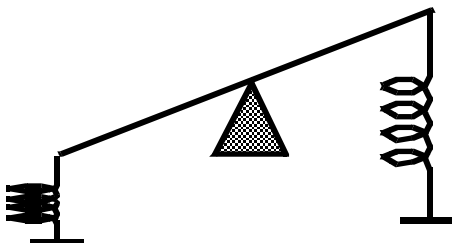
Analogie mécanique



Bistable Multivibrator
flip-flop, Schmitt Trigger



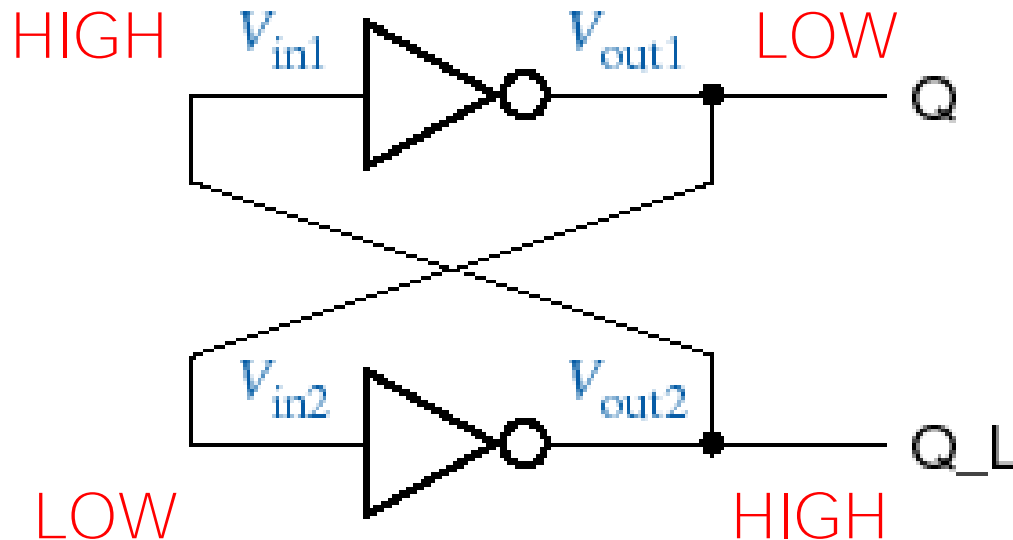
Monostable Multivibrator
one-shot



Astable Multivibrator
oscillator

Élément Bistable

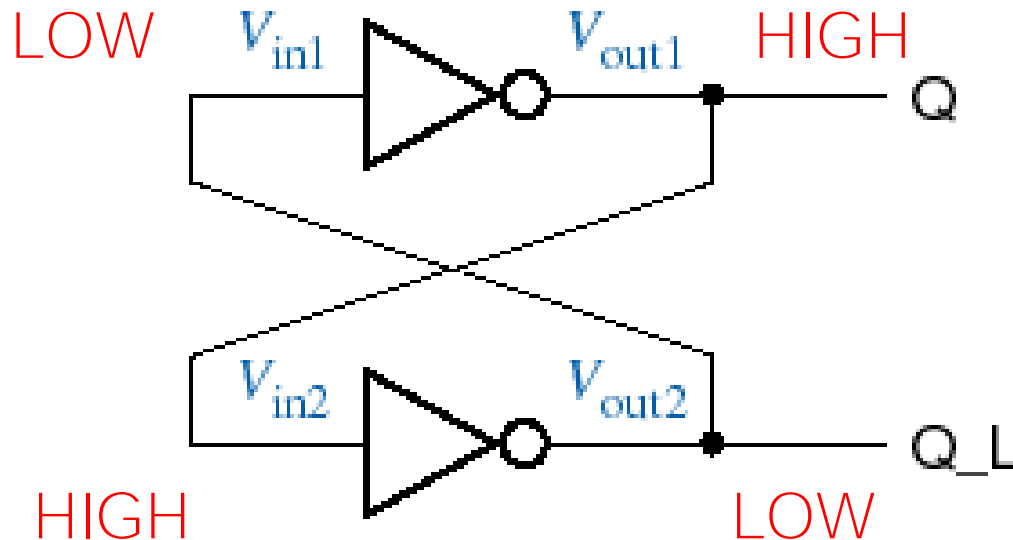
- Exemple avec le circuit le plus simple
- À deux états
 - La variable d'état est notée Q



Q Stable à 0

Élément Bistable

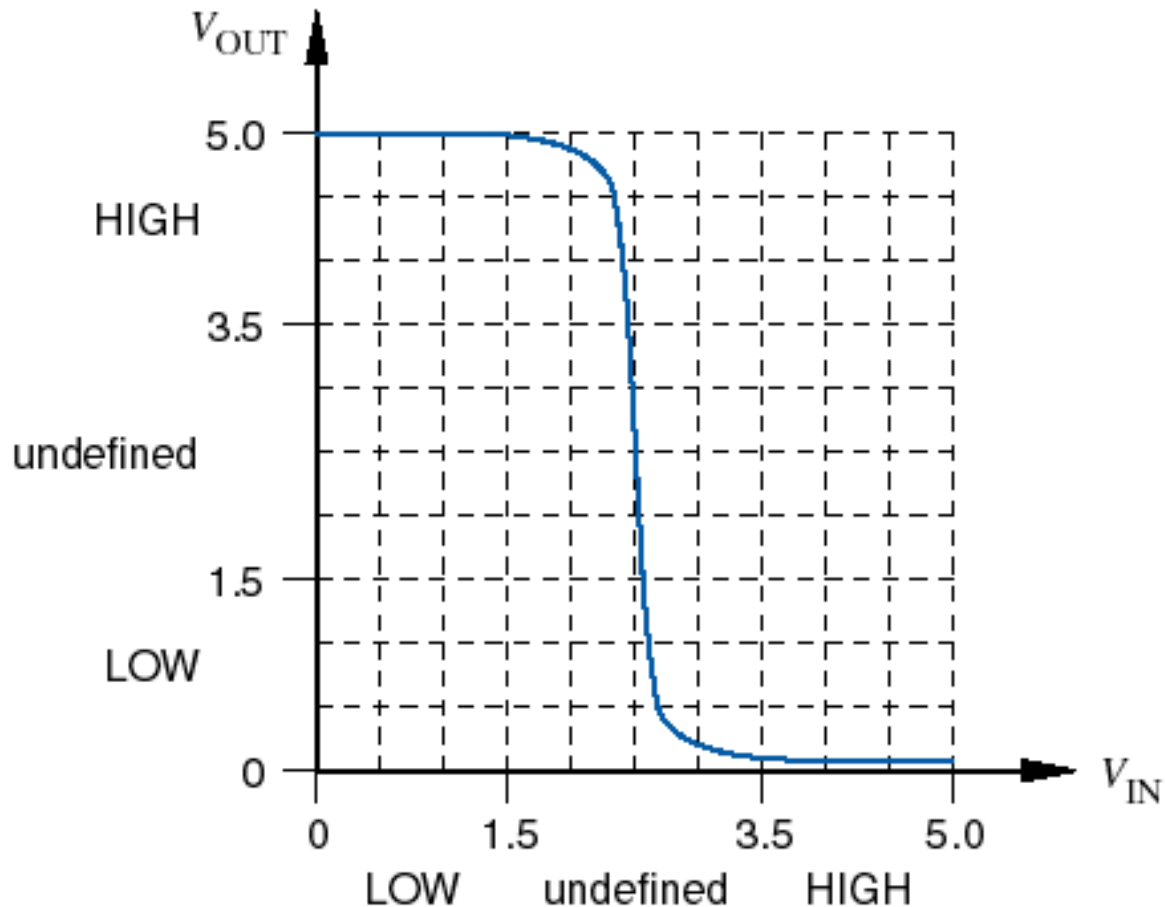
- Exemple avec le circuit le plus simple
- À deux états
 - La variable d'état est notée Q



Q Stable à 1

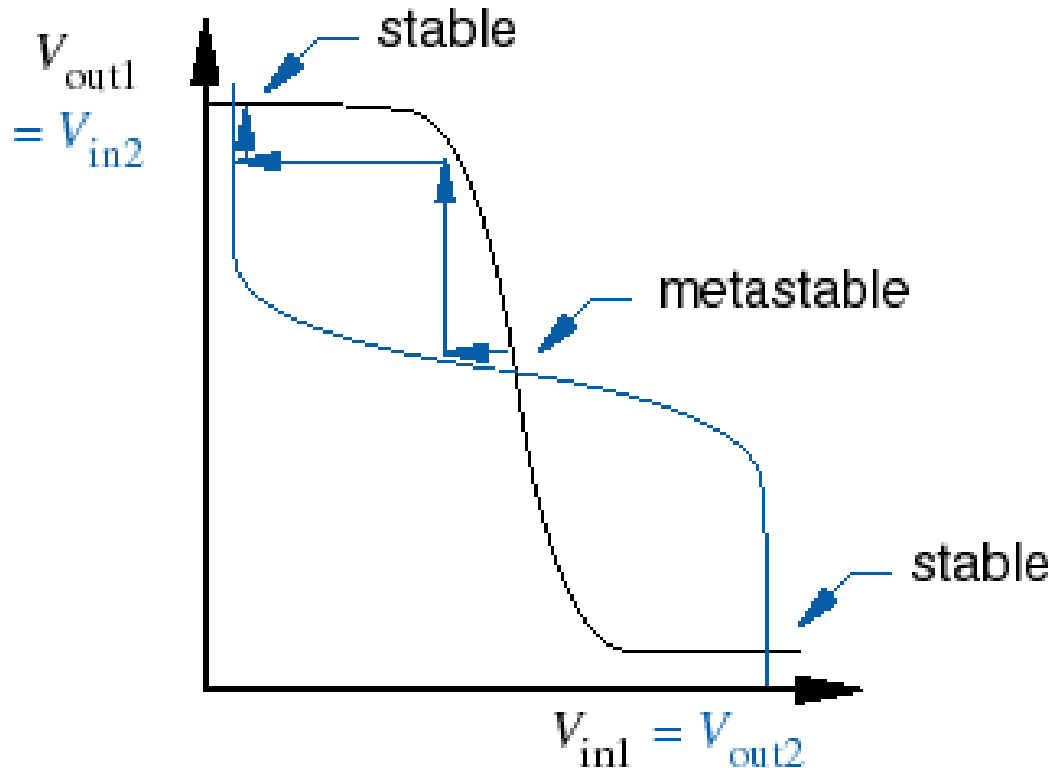
Analyse analogique d'un inverseur

- En supposant un seuil CMoS à **5.0 V**
- Et un seuil de basculement à **2.5 V**



Méta-stabilité des 2 inverseurs

- La méta-stabilité est inhérente à tout circuit bistable



Transfer function:

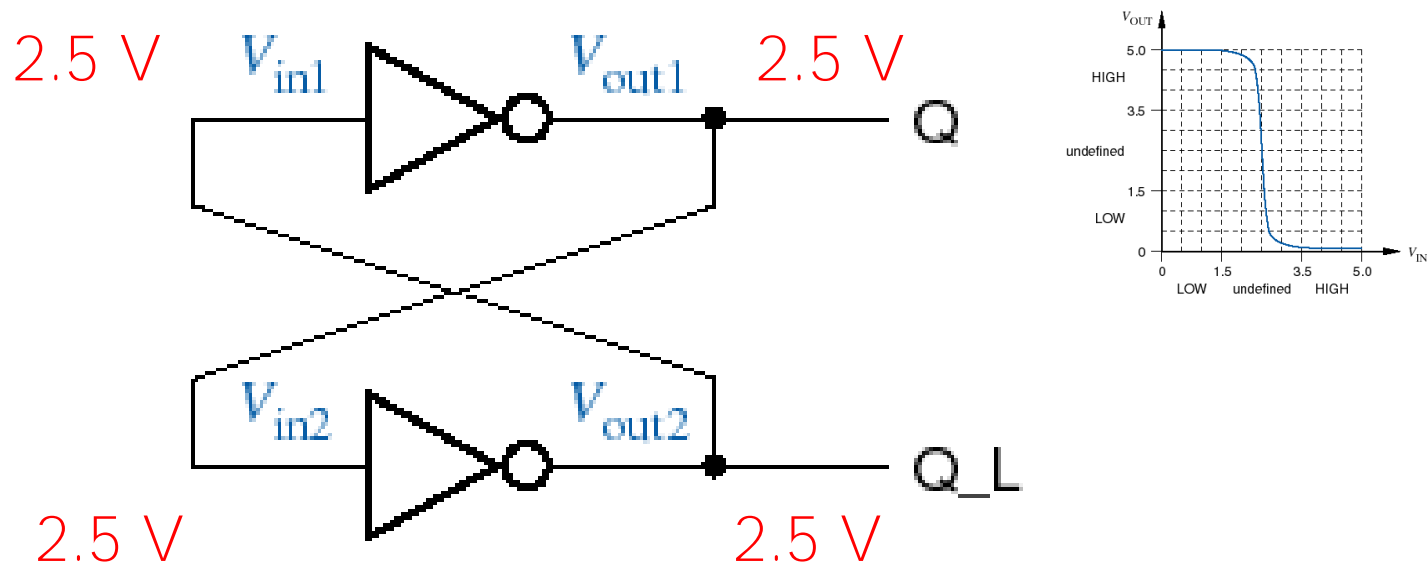
$$V_{out1} = T(V_{in1})$$

$$V_{out2} = T(V_{in2})$$

- Il y a 2 états stables et un état métastable

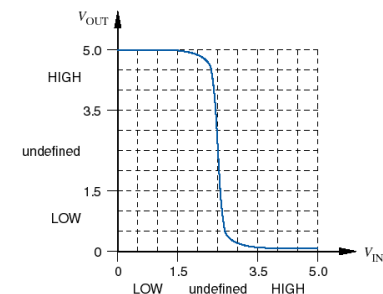
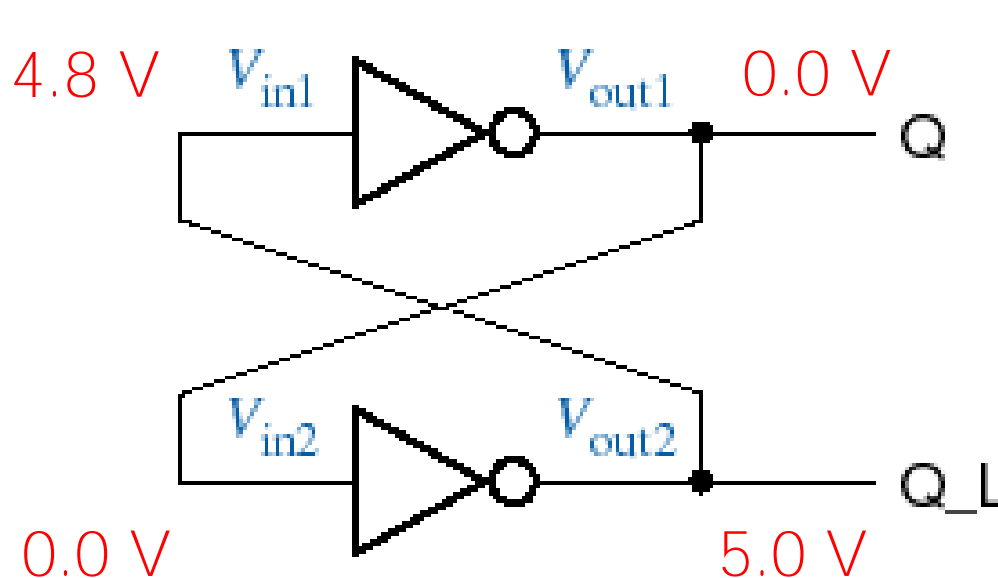
Analyse analogique

- En supposant un seuil CMoS à **5.0 V**
- Et un seuil de basculement à **2.5 V**



Analyse analogique

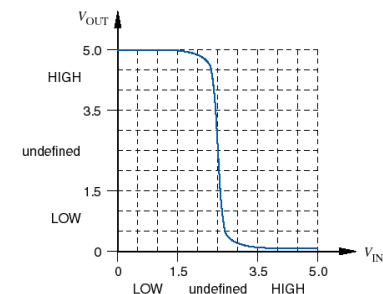
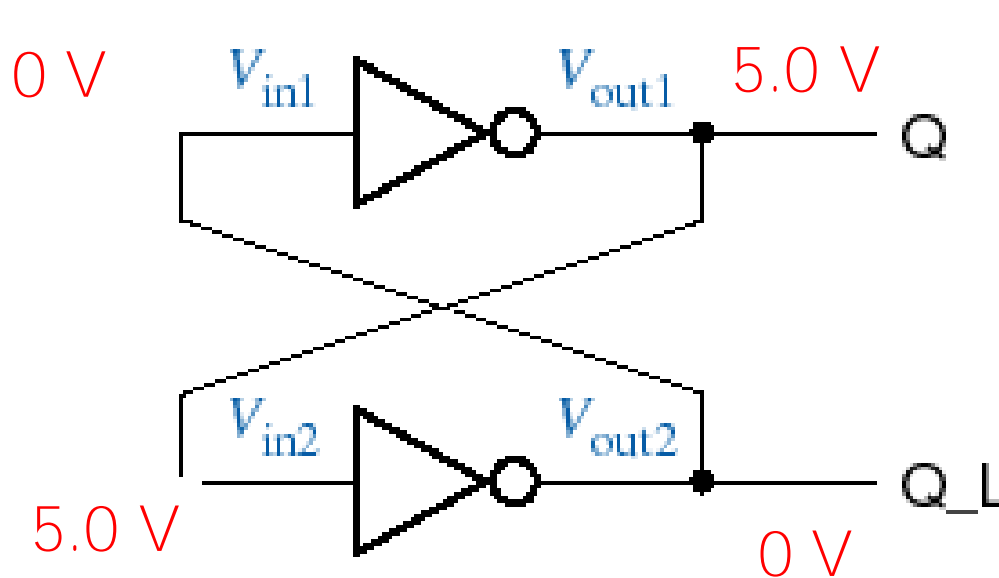
- En supposant un seuil CMoS à **5.0 V**
- Et un seuil de basculement à **2.5 V**



Q Stable à 0

Analyse analogique

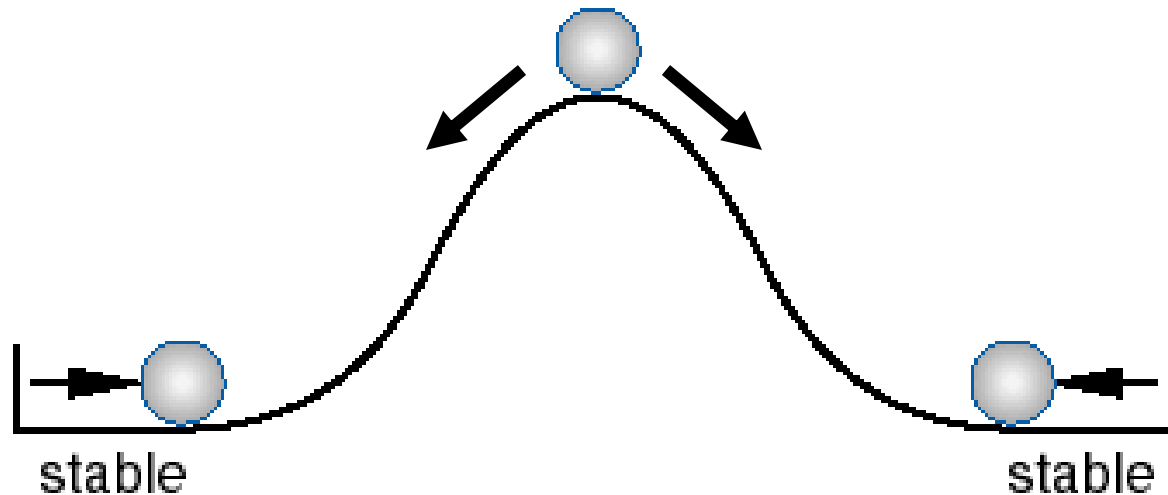
- En supposant un seuil CMoS à **5.0 V**
- Et un seuil de basculement à **2.5 V**



Mais on peut
aussi bien
passer à
Q Stable à 1

Méta-stabilité

Une légère variation conduit à passer vers un état stable ou un autre



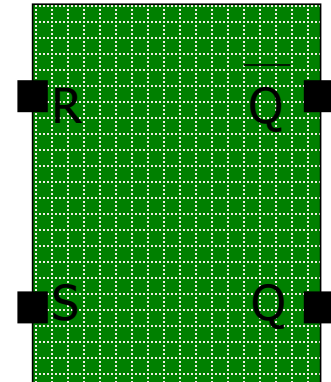
Le problème est alors que dans le mode asynchrone il est très difficile de garantir la stabilité de l'état lorsque les entrées changent de manière non-synchronisée. Il faut donc ajouter des signaux de contrôle !

Bascules à changement contrôlé

- Nous appellerons ces éléments de mémorisation contrôlés des bascules.
- Il en existe plusieurs sortes :
 - Bascules RS
 - Bascule D
 - Bascule D à clock
 - Bascule JK

Principe de la bascule RS

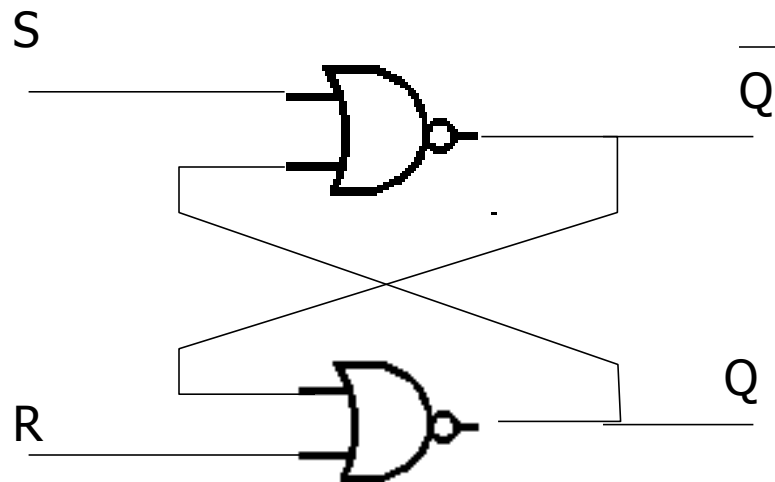
- Deux entrées :
 - R pour reset
 - S pour set
- Deux sorties
 - Q et \overline{Q}



Bascule RS

- La sortie ne dépend pas que de la valeur des entrées
- Elle dépend également de l'état mémorisé dans le circuit :

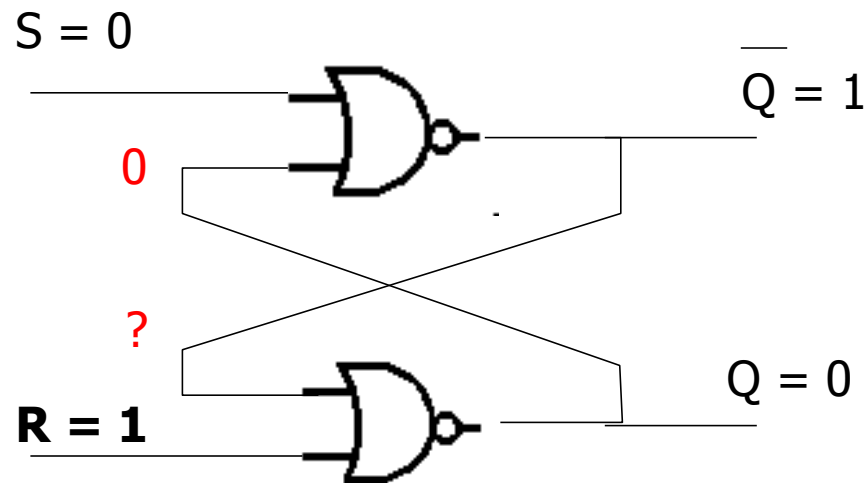
a	b	a nor b
0	0	1
0	1	0
1	0	0
1	1	0



Bascule RS

- La sortie ne dépend pas que de la valeur des entrées
- Elle dépend également de l'état mémorisé dans le circuit :

a	b	a nor b
0	0	1
0	1	0
1	0	0
1	1	0



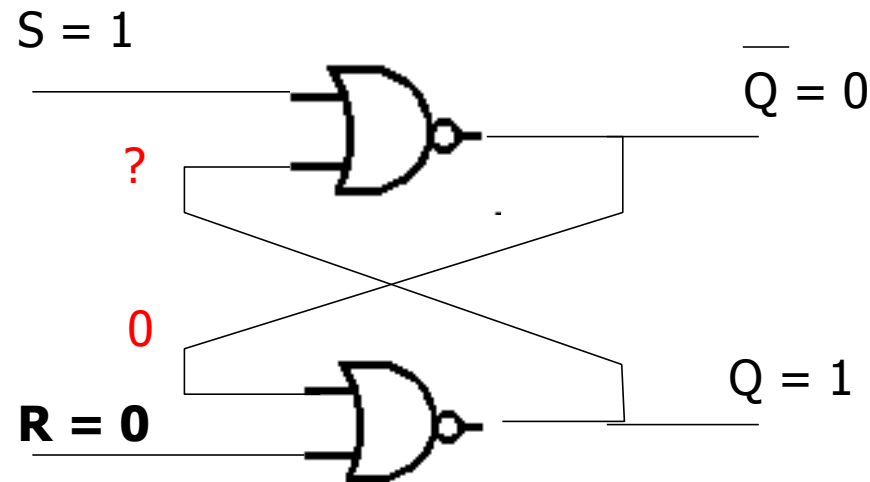
R	S	Q
0	0	
0	1	
1	0	0
1	1	

Combinatoire : Reset

Bascule RS

- La sortie ne dépend pas que de la valeur des entrées
- Elle dépend également de l'état mémorisé dans le circuit :

a	b	a nor b
0	0	1
0	1	0
1	0	0
1	1	0



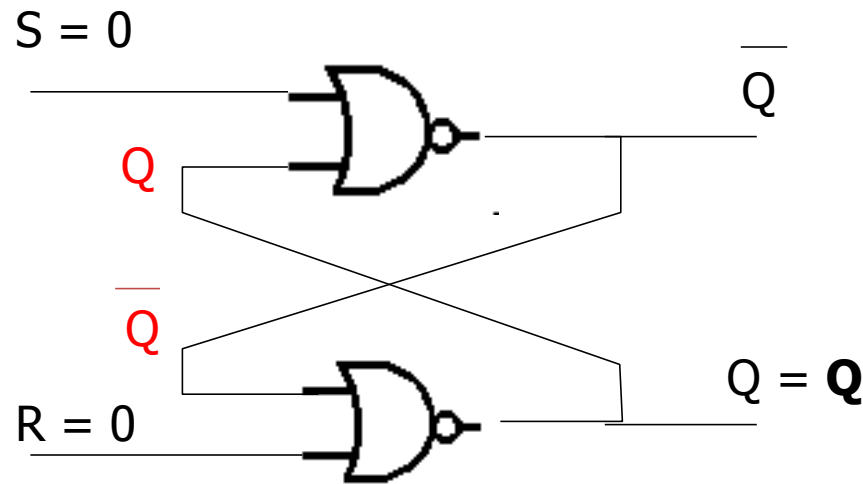
R	S	Q
0	0	
0	1	1
1	0	0
1	1	

Combinatoire : Set

Bascule RS

- La sortie ne dépend pas que de la valeur des entrées
- Elle dépend également de l'état mémorisé dans le circuit :

a	b	a nor b
0	0	1
0	1	0
1	0	0
1	1	0



R	S	Q
0	0	Q
0	1	1
1	0	0
1	1	?

Séquentiel : Memory

Bascule RS

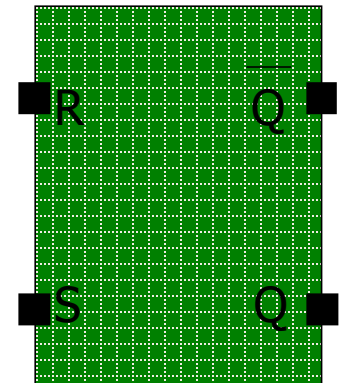
R	S	Q	Q+	notQ+
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	?	?
1	1	1	?	?

$Q^+ = Q$

Set

Reset

Méta-stable, Non utilisé



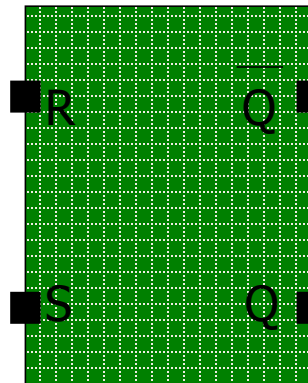
Bascule RS

- BasculeRS_1bit

R	S	Q	Q+	notQ+
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	?	?
1	1	1	?	?

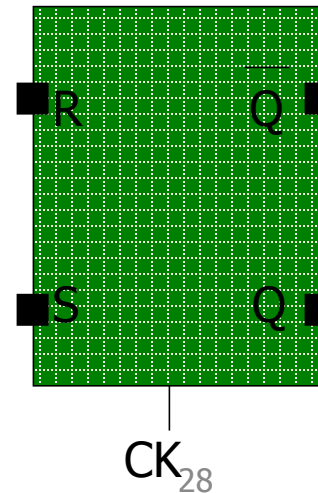
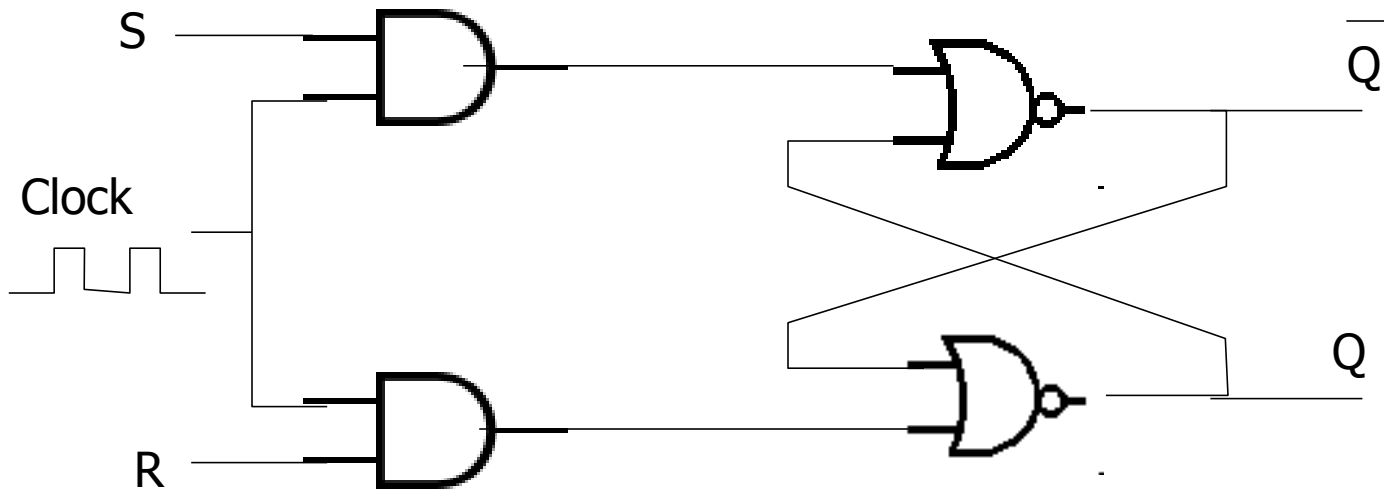
Q\RS	00	01	11	10
0		x		
1	x	x		

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



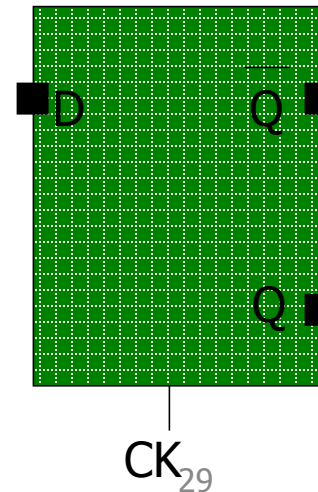
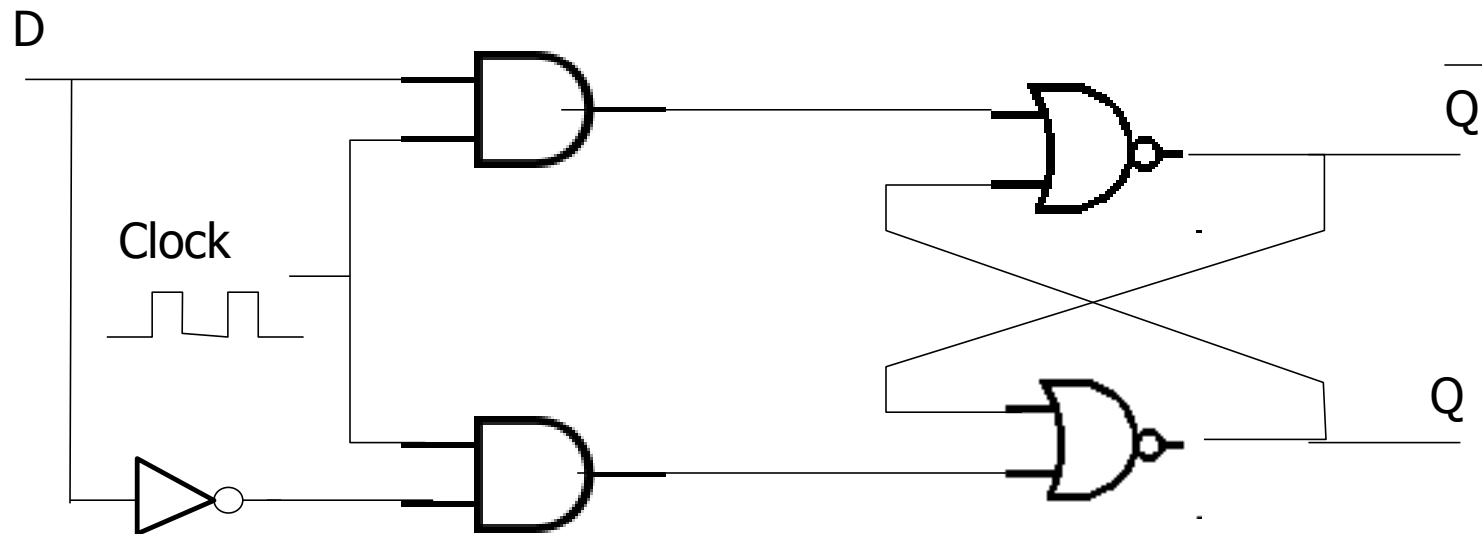
Bascule RS à *Clock*

- L'enregistrement ne se fait qu'à des moments bien précis dans le temps symbolisé par un signal en créneaux : *Clock*



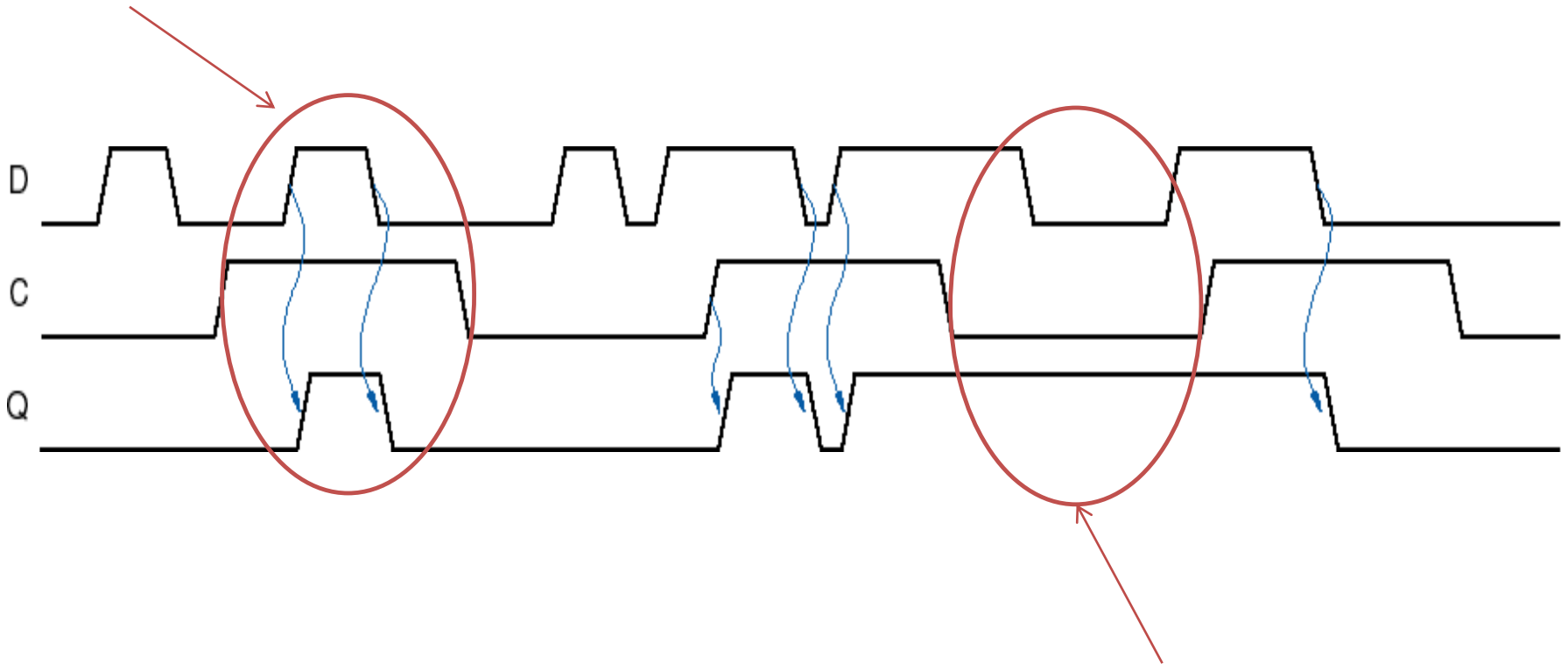
Bascule D

- L'état change sur toute la durée de l'état haut de l'horloge
- BasculeD_4bits



Comportement temporel

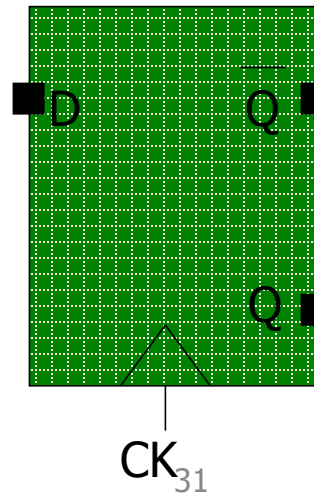
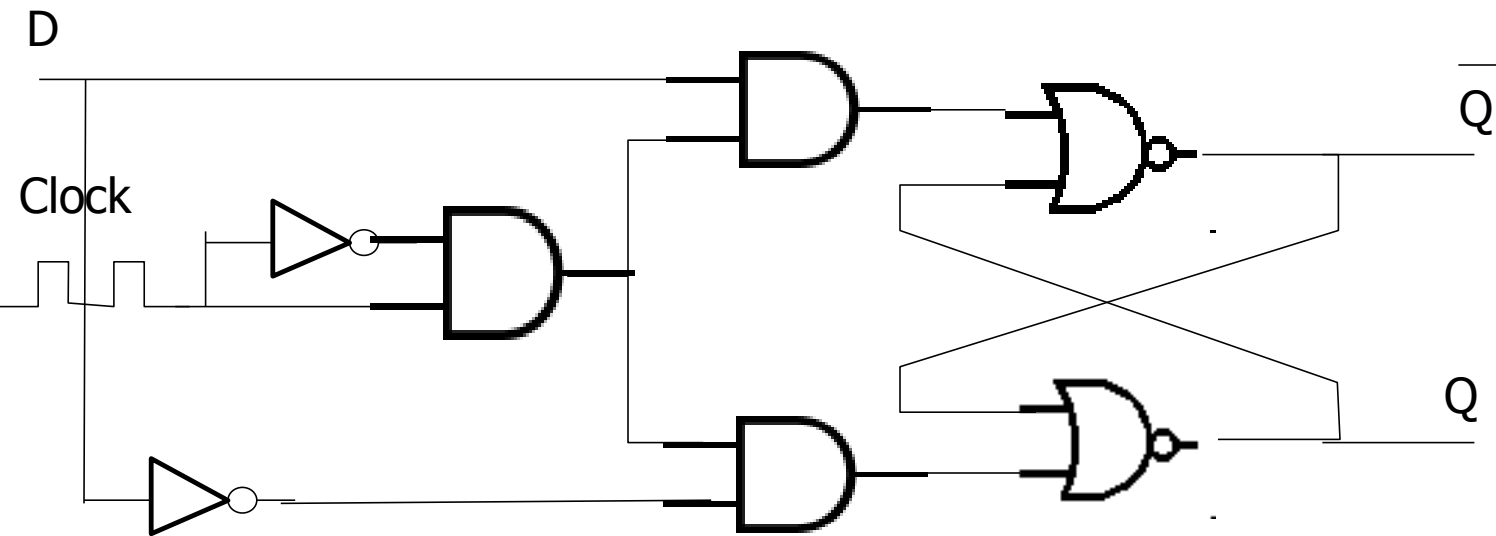
Sur état haut, la valeur de D est recopiée sur Q



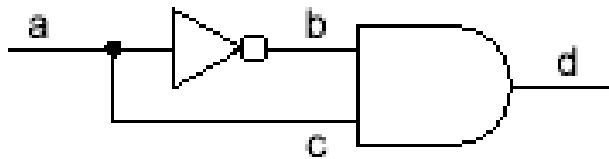
Sur état bas de Clk, Q maintient sa valeur

Flip-flop (bascule D sur front)

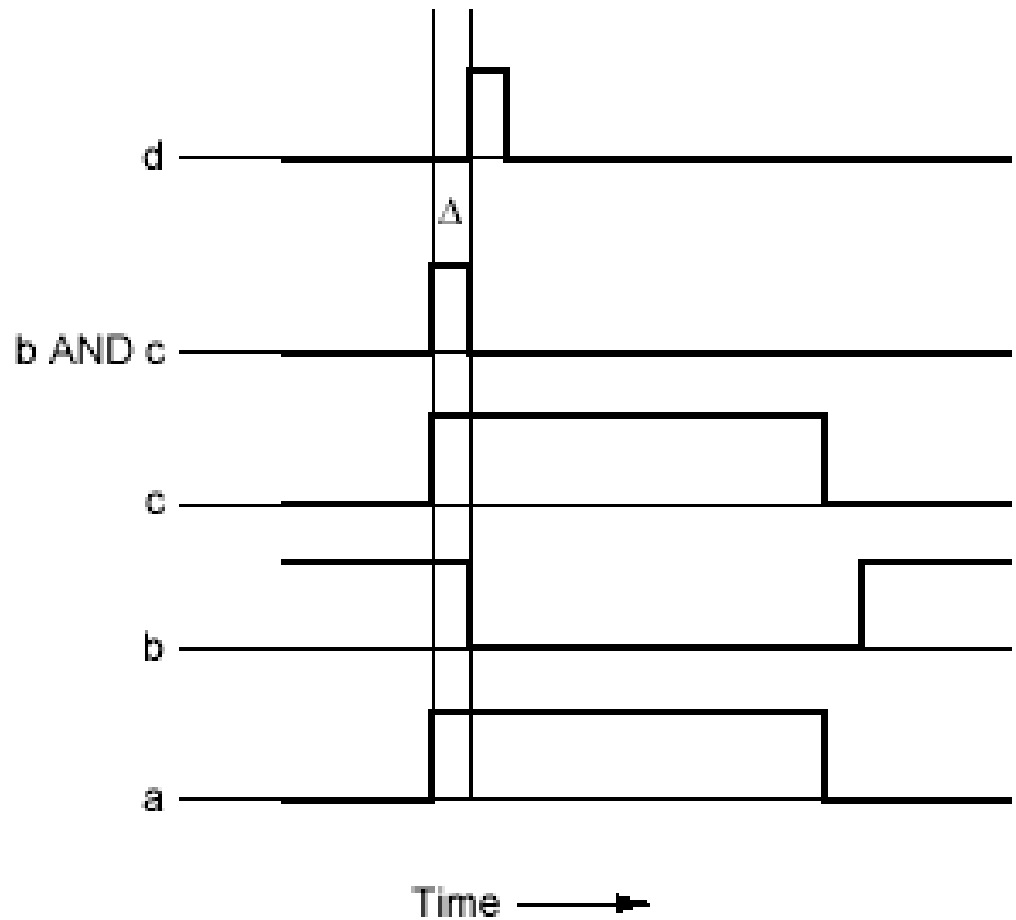
- Changement d'état au front montant de l'horloge



Activation sur front montant

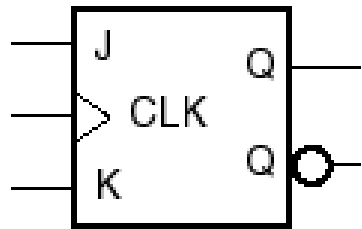
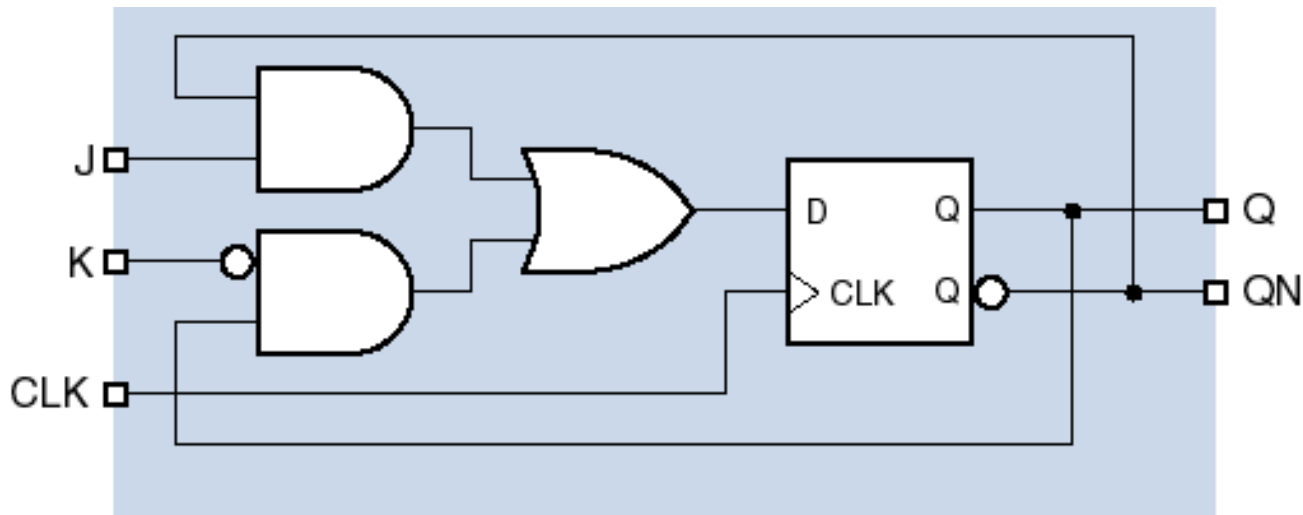


(a)



(b)

J-K Flip-Flops



J	K	CLK	Q	QN
x	x	0	last Q	last QN
x	x	1	last Q	last QN
0	0		last Q	last QN
0	1		0	1
1	0		1	0
1	1		last QN	last Q

Si $J = K = 0$, il y a conservation du dernier état logique

Si J différent de K , la sortie Q recopie l'entrée J et la sortie QN recopie l'entrée K

Si $J = K = 1$, le système bascule à chaque front d'horloge $Q = !Q$

Table de transition

J	K	Q	Q+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

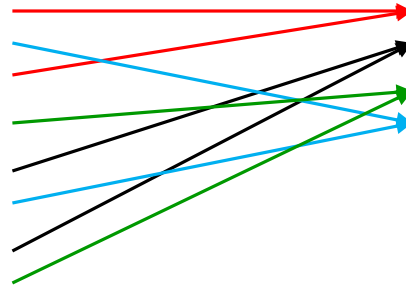


Table JK

Q	Q+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

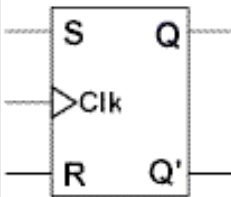
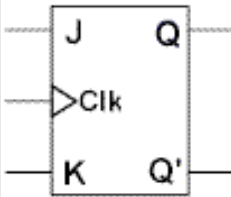
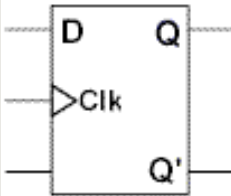
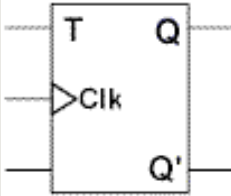
K-Map for Q+

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

Equation de JK

$$Q^+ = J \cdot \bar{Q} + \bar{K} Q$$

Résumé sur les Flip-Flop

FLIP-FLOP NAME	FLIP-FLOP SYMBOL	CHARACTERISTIC TABLE	CHARACTERISTIC EQUATION	EXCITATION TABLE																																			
SR		<table><tr><th>S</th><th>R</th><th>Q_(next)</th></tr><tr><td>0</td><td>0</td><td>Q</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>?</td></tr></table>	S	R	Q _(next)	0	0	Q	0	1	0	1	0	1	1	1	?	$Q_{(next)} = S + R'Q$ $SR = 0$	<table><tr><th>Q</th><th>Q_(next)</th><th>S</th><th>R</th></tr><tr><td>0</td><td>0</td><td>0</td><td>X</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>X</td><td>0</td></tr></table>	Q	Q _(next)	S	R	0	0	0	X	0	1	1	0	1	0	0	1	1	1	X	0
S	R	Q _(next)																																					
0	0	Q																																					
0	1	0																																					
1	0	1																																					
1	1	?																																					
Q	Q _(next)	S	R																																				
0	0	0	X																																				
0	1	1	0																																				
1	0	0	1																																				
1	1	X	0																																				
JK		<table><tr><th>J</th><th>K</th><th>Q_(next)</th></tr><tr><td>0</td><td>0</td><td>Q</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>Q'</td></tr></table>	J	K	Q _(next)	0	0	Q	0	1	0	1	0	1	1	1	Q'	$Q_{(next)} = JQ' + K'Q$	<table><tr><th>Q</th><th>Q_(next)</th><th>J</th><th>K</th></tr><tr><td>0</td><td>0</td><td>0</td><td>X</td></tr><tr><td>0</td><td>1</td><td>1</td><td>X</td></tr><tr><td>1</td><td>0</td><td>X</td><td>1</td></tr><tr><td>1</td><td>1</td><td>X</td><td>0</td></tr></table>	Q	Q _(next)	J	K	0	0	0	X	0	1	1	X	1	0	X	1	1	1	X	0
J	K	Q _(next)																																					
0	0	Q																																					
0	1	0																																					
1	0	1																																					
1	1	Q'																																					
Q	Q _(next)	J	K																																				
0	0	0	X																																				
0	1	1	X																																				
1	0	X	1																																				
1	1	X	0																																				
D		<table><tr><th>D</th><th>Q_(next)</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	D	Q _(next)	0	0	1	1	$Q_{(next)} = D$	<table><tr><th>Q</th><th>Q_(next)</th><th>D</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	Q	Q _(next)	D	0	0	0	0	1	1	1	0	0	1	1	1														
D	Q _(next)																																						
0	0																																						
1	1																																						
Q	Q _(next)	D																																					
0	0	0																																					
0	1	1																																					
1	0	0																																					
1	1	1																																					
T		<table><tr><th>T</th><th>Q_(next)</th></tr><tr><td>0</td><td>Q</td></tr><tr><td>1</td><td>Q'</td></tr></table>	T	Q _(next)	0	Q	1	Q'	$Q_{(next)} = TQ' + T'Q$	<table><tr><th>Q</th><th>Q_(next)</th><th>T</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	Q	Q _(next)	T	0	0	0	0	1	1	1	0	1	1	1	0														
T	Q _(next)																																						
0	Q																																						
1	Q'																																						
Q	Q _(next)	T																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	0																																					

Registre et Banc de registres

- Un registre est un assemblage de bascules mémorisant un mot de données
 - Un octet, 8 bits
 - Un demi-mot, 16 bits
 - Un mot, 32 bits
 - ...
- Un banc de registres est un ensemble de plusieurs registres permettant de stocker une donnée temporaire dans le processeur avant d'être rangée en mémoire.

Banc et mémoire

- Le banc doit être en accès lecture et/ou écriture
- Les données à écrire proviennent soit de l'ALU soit de la mémoire de données
- Les données à lire sont à destination soit de l'ALU soit de la mémoire.

Architecture générale

2. Contrôleur

3. Chemin de données

