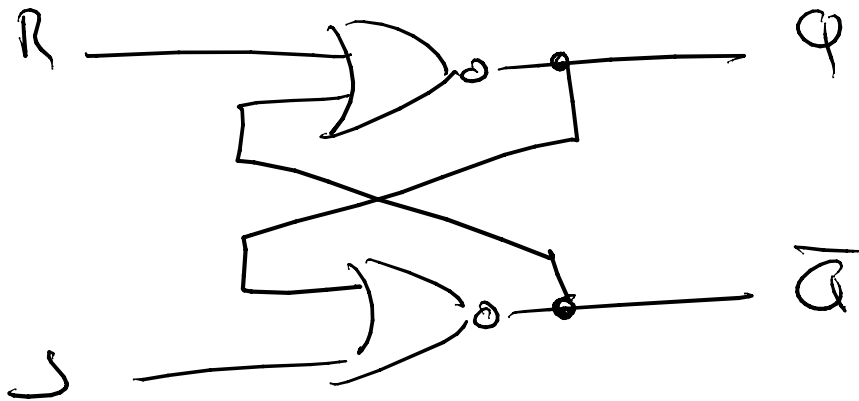


Les Bascules

1) Bascule RS :

a) A base de portes NOR



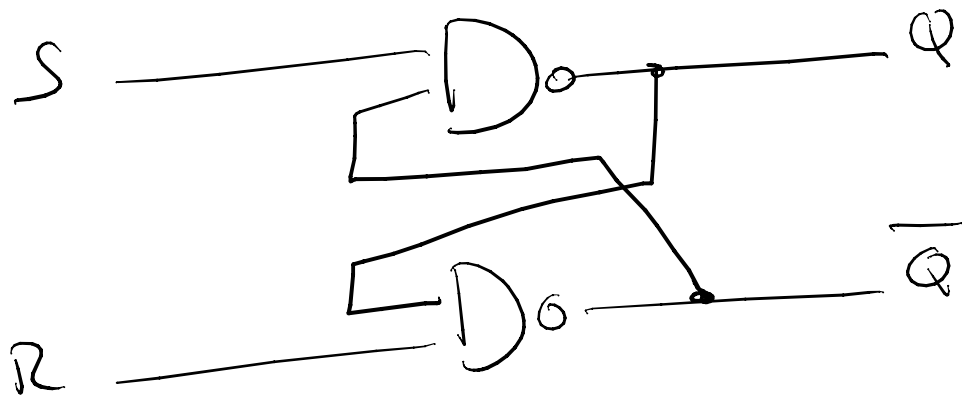
$$\begin{aligned} Q_{n+1} &= R_n + S_n + Q_n \\ &= \overline{R_n} \cdot (S_n + Q_n) \end{aligned}$$

$$Q_{n+1} = S_n \overline{R_n} + Q_n R_n$$

R	S	Q _{n+1}
0	0	Q _n
0	1	1
1	0	0
1	1	Interdit

Il est à noter que R et S sont actifs au niveau 1.

b) A base de NAND



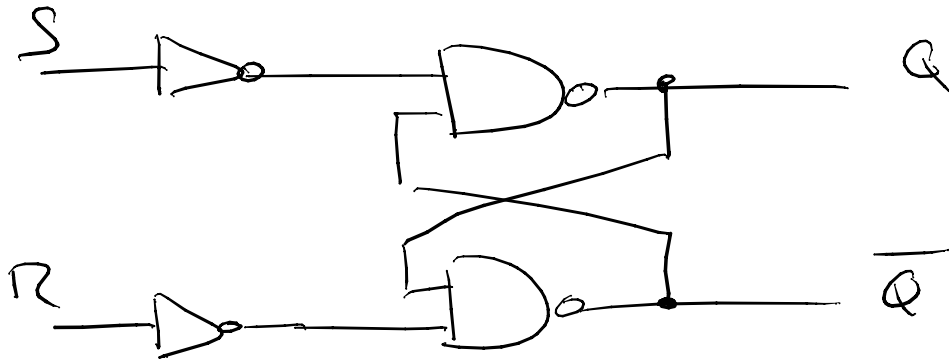
$$Q_{n+1} = S_n \cdot Q_n \cdot R_n$$

$$Q_{n+1} = \overline{S_n} + Q_n R_n$$

S	R	Q_{n+1}
0	0	Interdit
0	1	1
1	0	0
1	1	Q_n

Dans ce cas les signaux R et S sont actifs au niveau 0.

Afin de rendre les comportements des deux bascules RS similaires il faut rajouter un inverseur aux entrées



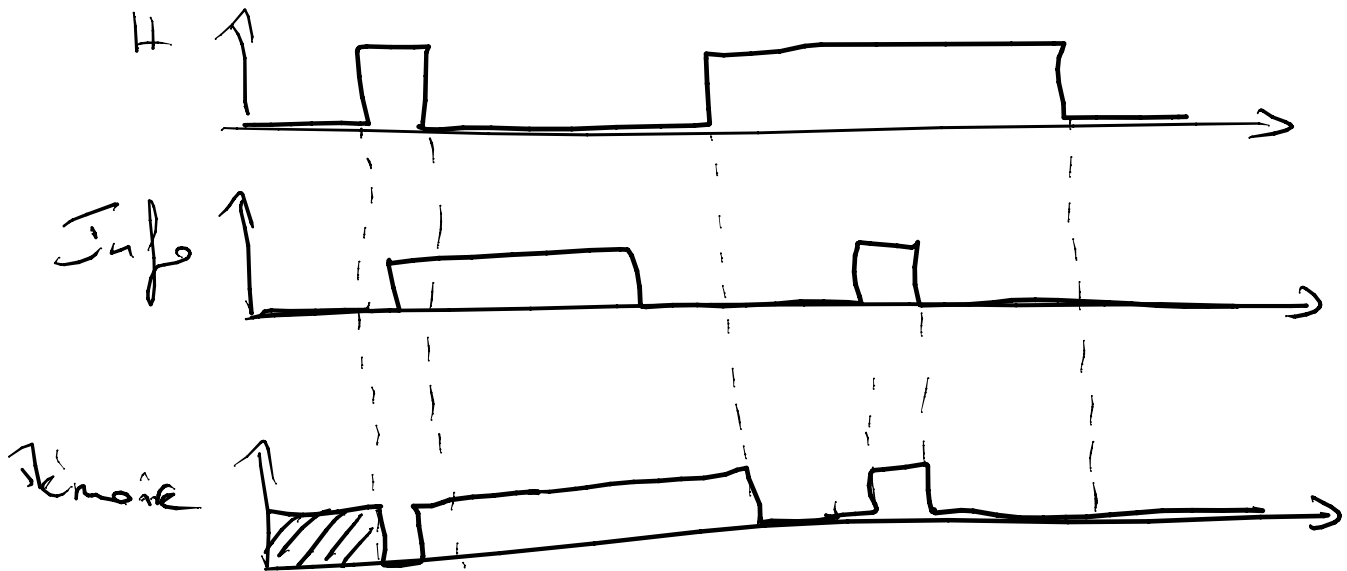
$$Q_{n+1} = \overline{\overline{S_n} \cdot \overline{R_n} \cdot Q_n}$$

$$Q_{n+1} = S_n + Q_n \overline{R_n}$$

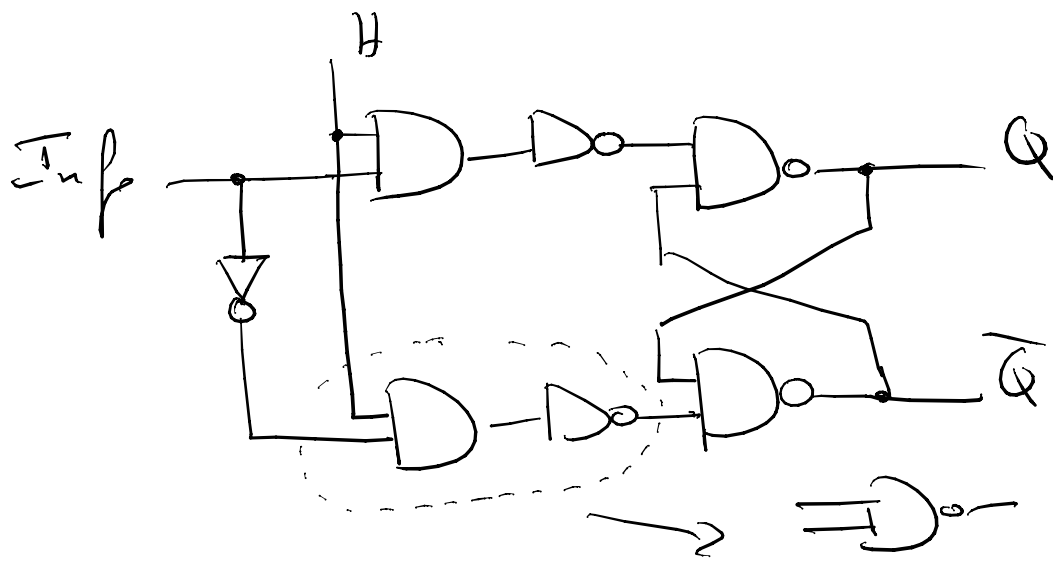
S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Interdit

2) Bascule D latch

Objectif: Rémémorer une information à l'aide d'une commande



L'info est mémorisée quand le signal H est à 1. Pour réaliser cela, on peut modifier une bascule RS. On envoie l'info sur S et son complément sur R à travers une porte de blocage (AND) commandée par H .



$$Q_{n+1} = \overline{H_n \cdot Inf_n} \cdot \overline{H_n \cdot Inf_n} \cdot Q_n$$

$$Q_{n+1} = H_n \cdot Inf_n + Q_n (\overline{H_n} + \overline{Inf_n})$$

Si $H = 1$

$$\hookrightarrow Q_{n+1} = Inf_n$$

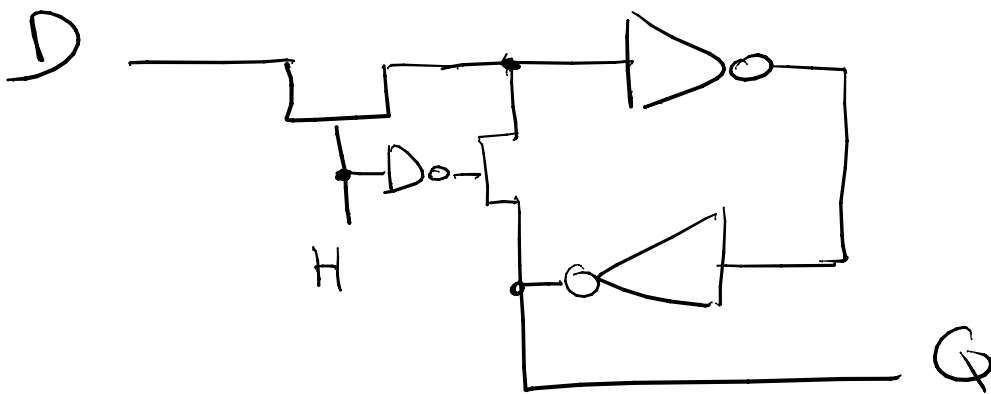
Si $H = 0$

$$\hookrightarrow Q_{n+1} = Q_n$$

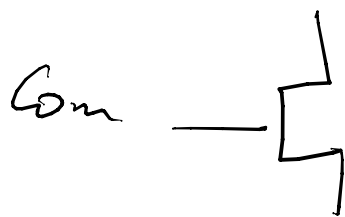
H	Inf	Q_{n+1}
0	0	Q_n
0	1	Q_n
1	0	0
1	1	1

C'est une bascule D latch, cad
 une bascule mémorisant une information
 sur niveau actif de l'horloge H.
 Dans notre exemple le niveau actif
 est le 1 logique.

Remarque: En réalité, afin de minimiser
 la surface occupée par la bascule, elle
 est réalisée de la manière suivante.



Fonctionnement:



si $G_m = 0$
 \hookrightarrow

si $G_m = 1$
 \hookrightarrow

capture de l'information en ouvrant la sonde

Q

Q

Mémorisation
La boucle est
fermée

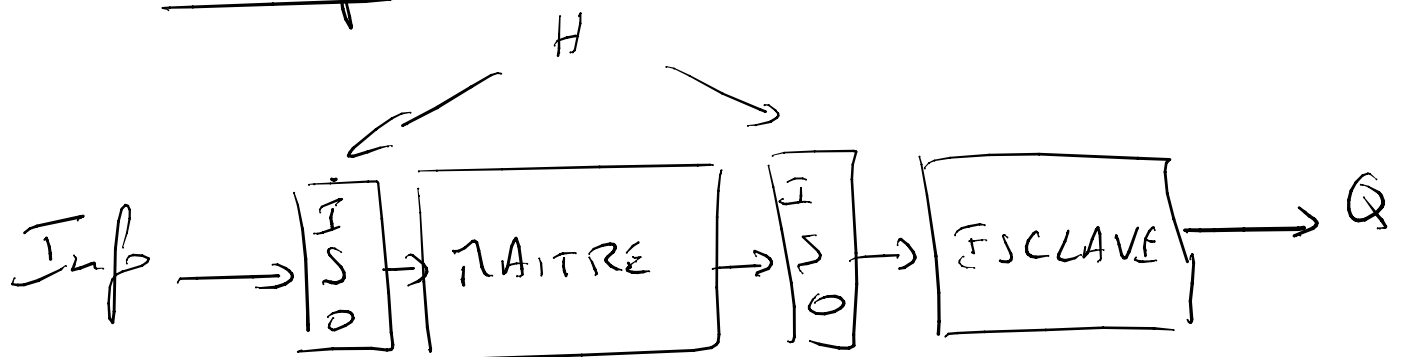
Avec une telle bascule l'information est mémorisée sur un niveau. Donc, si il y a changement de l'inf sur le niveau actif de H, il y a aussi changement de la valeur mémorisée.

3) Bascule D maître-esclave :

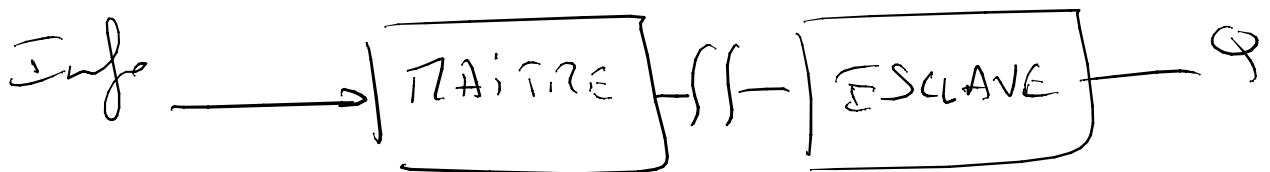
objectif: Mémoriser l'information à un instant donné.

↳ Échantillonnage

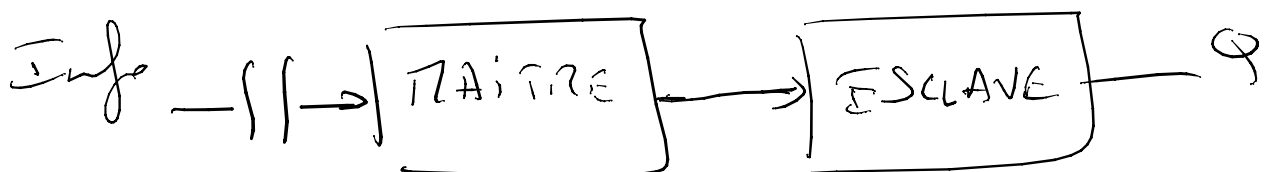
Principe:



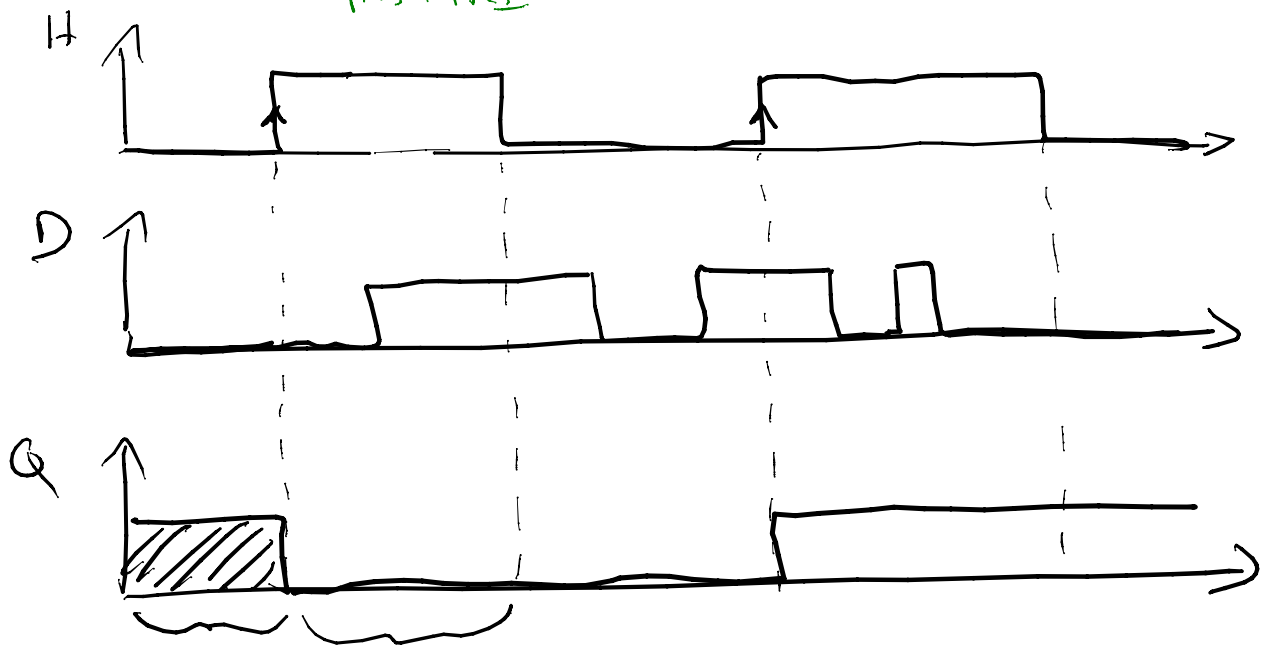
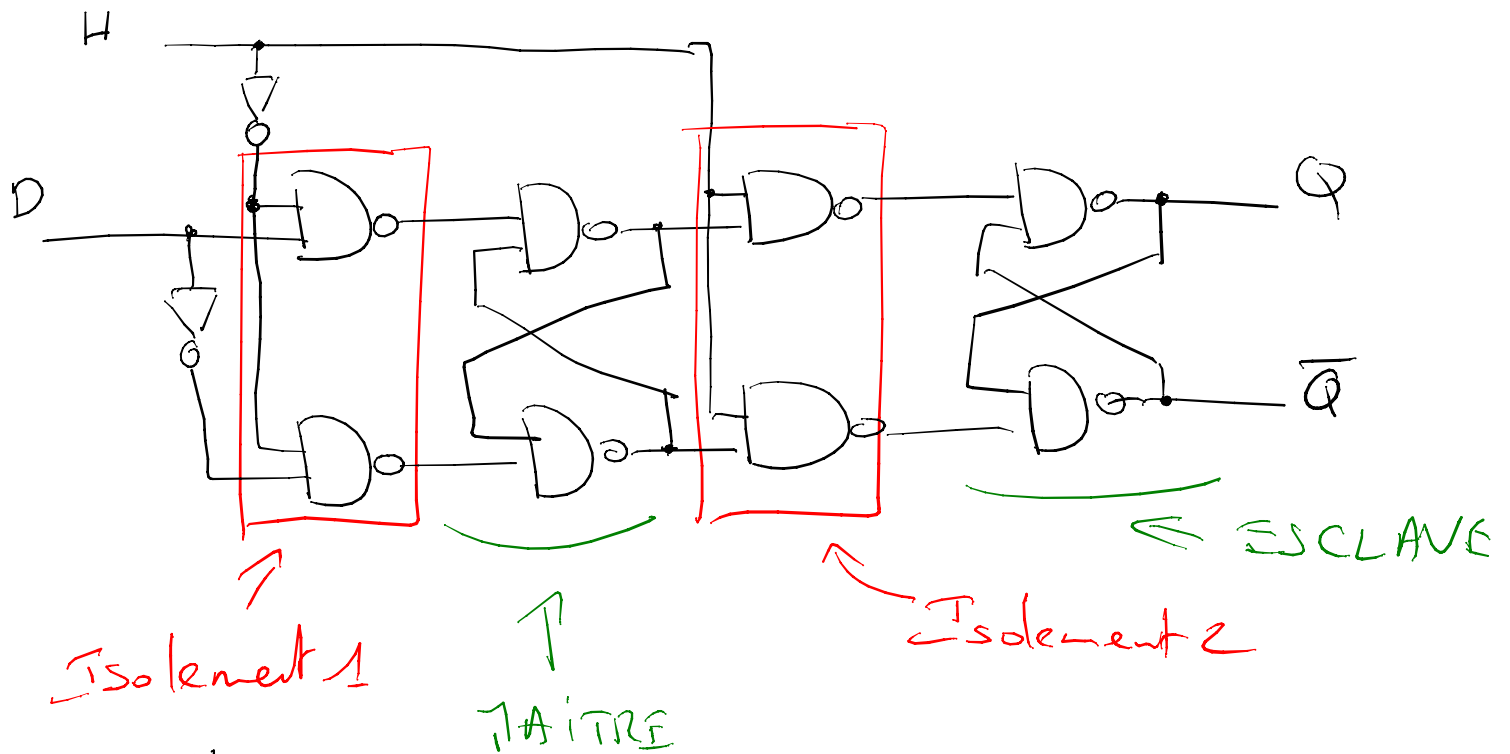
Si $H=0$



Si $H=1$



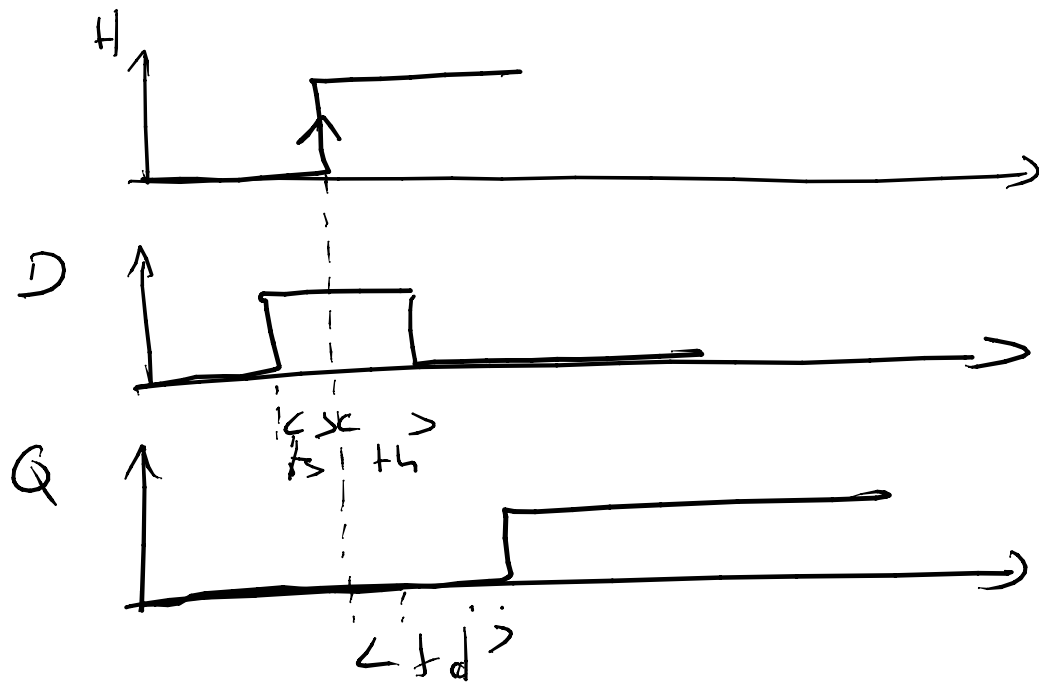
Réalisation: Mise en cascade de deux bascules D latch.



Le maître capture l'inf et tout changement d'inf tant que $H=0$

quand H passe à 1 (front montant \uparrow) le maître ne capture plus de valeur et l'esclave récupère la dernière valeur (stable) de maître.

Prise en compte du temps :



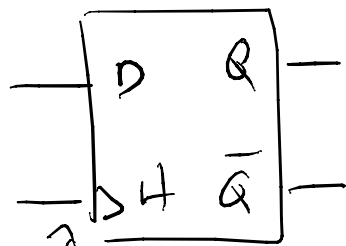
t_s : setup time

t_h : hold time

t_d : delay time

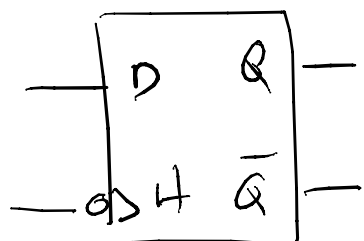
Conclusion : La donnée D à mémoriser doit être présente (et stable) à l'entrée D de la bascule un temps supérieur ou égal à t_s (setup time) avant le front d'horloge. Elle doit aussi rester stable un temps t_h après le front d'horloge.

Symbolle:



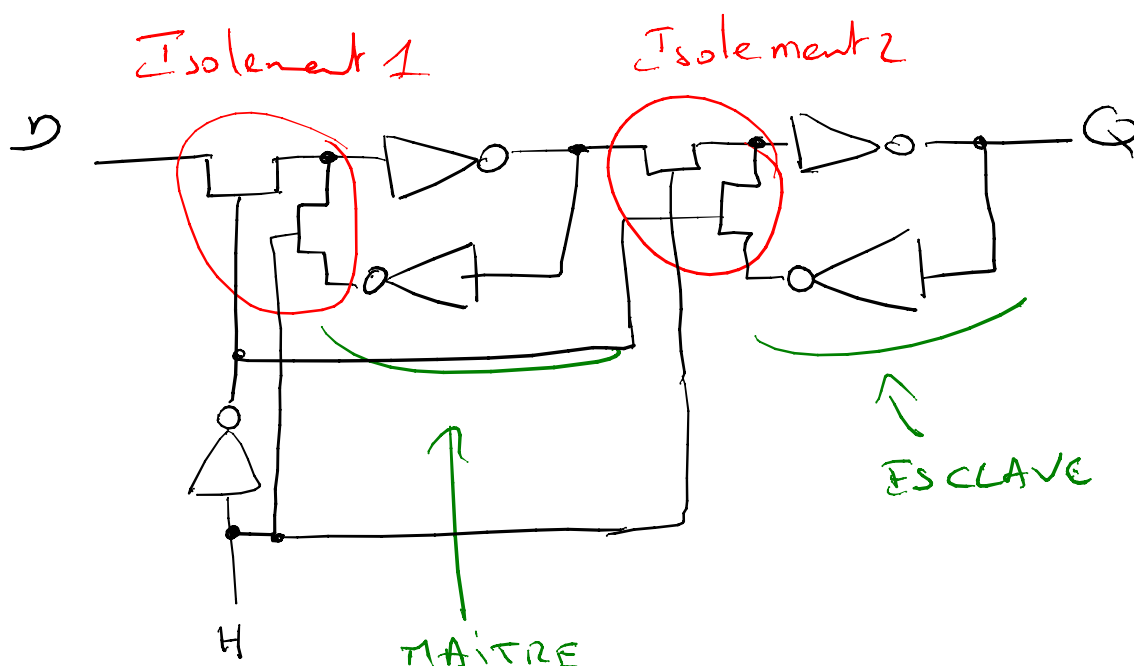
H	Q_{n+1}
0	Q_n
1	D_n

Horloge active sur front montant.

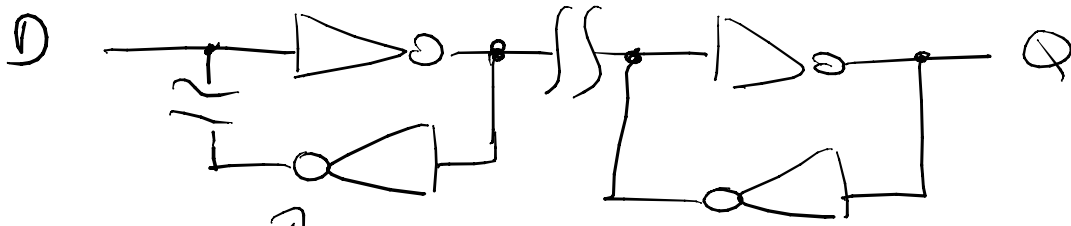


Idem mais Horloge active
sur front descendant

Remarque 1: Comme dans le cas de la bascule D latch, la réalisation d'un D maître-esclave est la suivante :



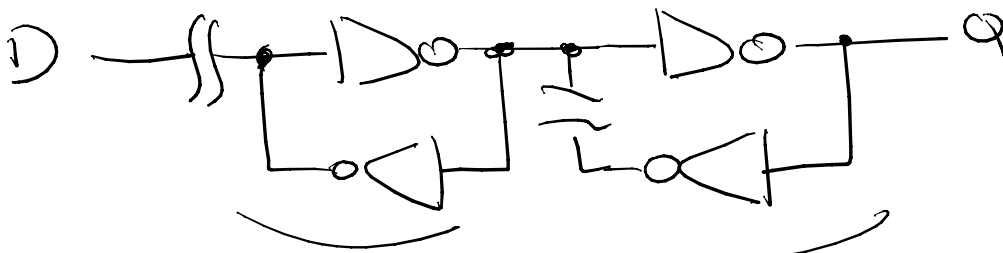
• $H=0$



le maître
capture la donnée
tant que $H=0$

l'ESCLAVE
conserve la valeur
précédemment mémorisée

• $H=1$



le MAÎTRE
a une valeur
stable

l'ESCLAVE
récupère la donnée
du MAÎTRE et présente
cette valeur sur la
sortie Q.