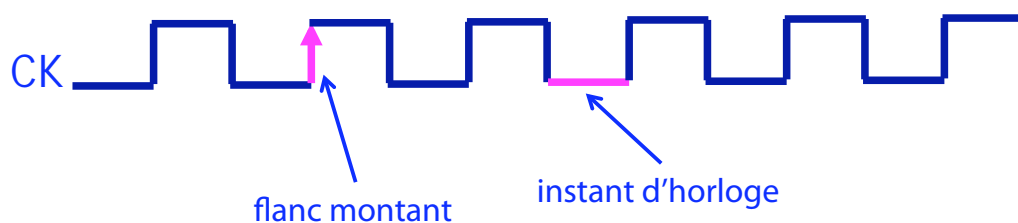
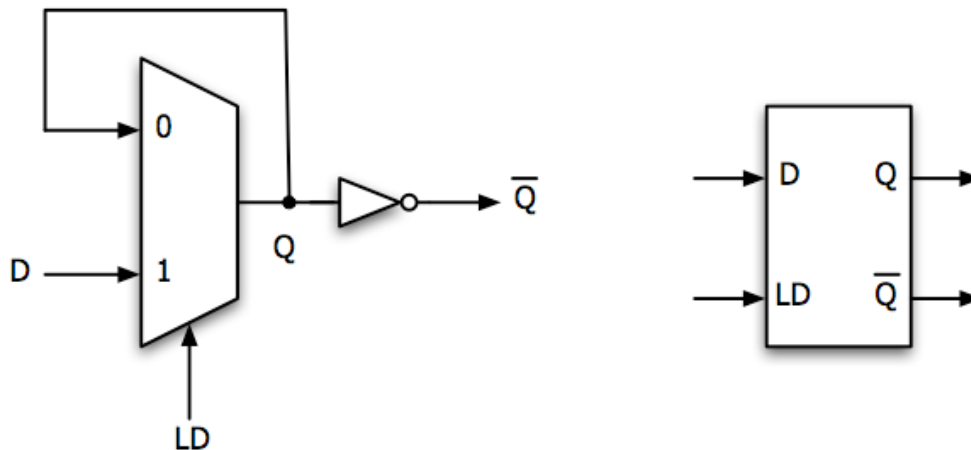


Systemes séquentiels

- ◆ La valeur des sorties dans un système séquentiel ne dépend pas uniquement des valeurs des entrées à l'instant présent, mais également de la valeur des entrées dans les instants précédents
- ◆ Les systèmes séquentiels introduisent donc la notion de mémoire (pour stocker l'état du système) et d'horloge (pour déterminer les instants de mesure)



Elément de mémoire D (*latch*)



D: entrée d'excitation
LD: entrée de contrôle
Q: sortie ou état du latch

D	LD	Q
0	0	Q
0	1	0
1	0	Q
1	1	1

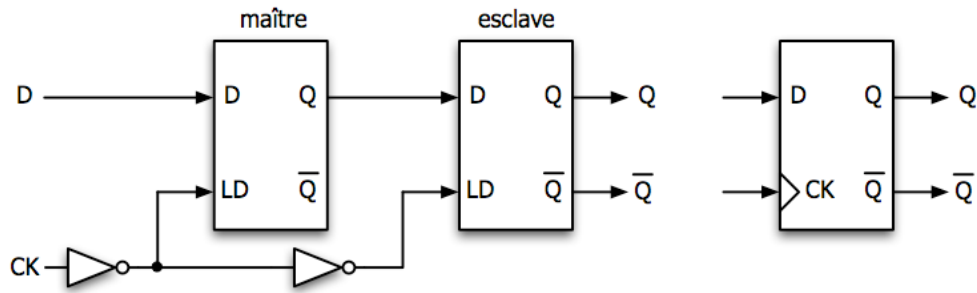
Si l'entrée de contrôle est inactive (**LD=0**), la sortie conserve sa valeur: fonction de mémorisation (**HOLD**).

Si l'entrée de contrôle est active (**LD=1**), la sortie prend la valeur de son entrée d'excitation (**LOAD**).

La table d'opérations du latch est donc:

opération	description	LD
HOLD	$Q \leftarrow Q$	0
LOAD	$Q \leftarrow D$	1

Bascule bistable D (flip-flop)



- ◆ Si $CK=0$, l'élément maître laisse passer l'entrée D à sa sortie, mais la sortie de l'élément esclave reste inchangée. Lorsque $CK=1$, la sortie de l'élément maître est conservée, et passe à la sortie de l'élément esclave
- ◆ La sortie de la bascule peut changer seulement pendant la montée du signal d'horloge CK
- ◆ La sortie de la bascule prend la valeur mesurée à l'entrée D au moment de la montée du signal d'horloge

- ◆ La table d'états de la bascule est donc:

D	Q	Q ⁺
0	0	0
0	1	0
1	0	1
1	1	1

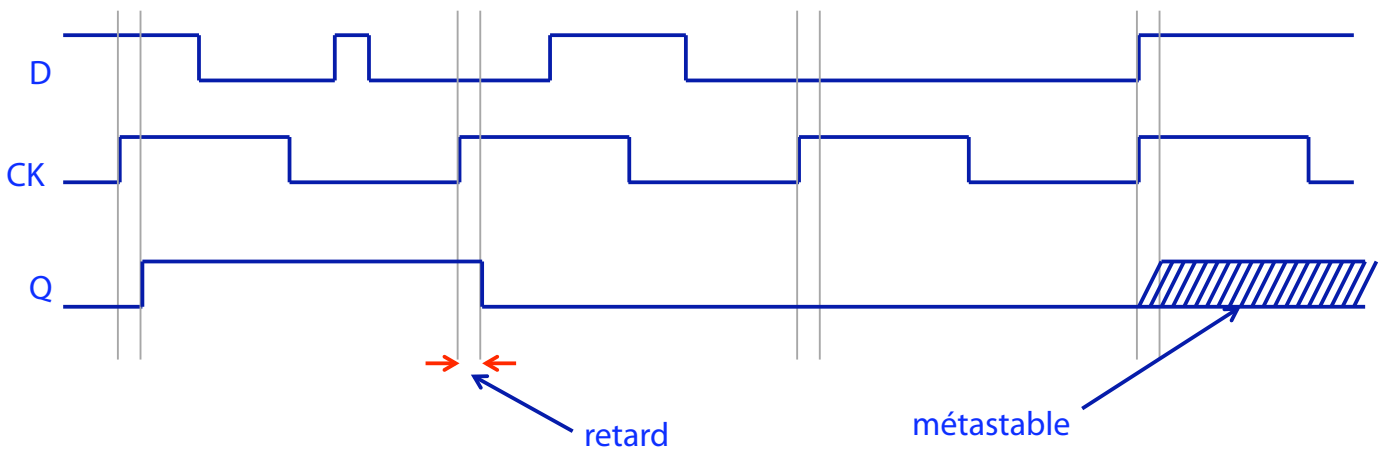
Cette table montre la valeur de la sortie future en fonction de l'entrée et de l'état présent

- ◆ Le comportement de la bascule peut être décrit également par son équation caractéristique:

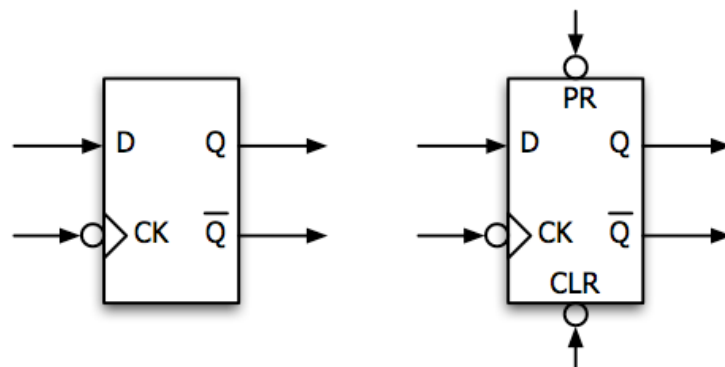
$$Q^+ = D$$

$$Q \leftarrow D$$

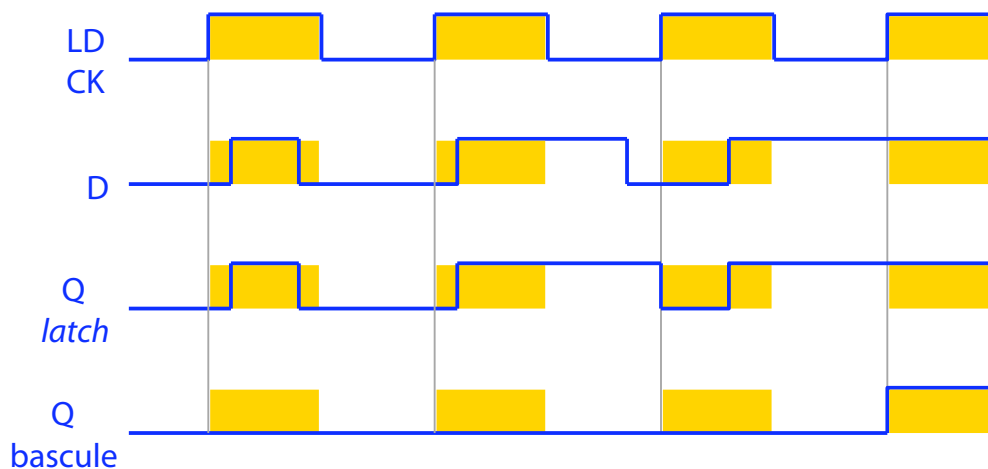
(la sortie Q prend la valeur de l'entrée D au prochain flanc montant de l'horloge)



- ◆ Il existe également des bascules **D** sensibles au flanc descendant de l'horloge
 - ◆ La bascule **D** peut avoir des entrées asynchrones, qui forcent la sortie à **1** ou **0**, indépendamment des entrées **D** et **CK**:
 - PR** (*preset*) met la sortie à **1**
 - CLR** (*clear*) met la sortie à **0**
- Dans la mesure du possible, ces entrées asynchrones sont utilisées uniquement pour l'initialisation ou le test des bascules

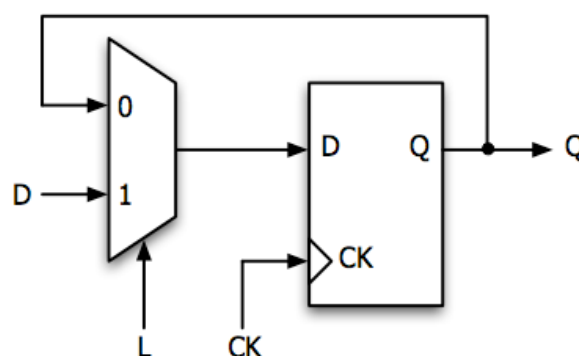


Différence de comportement entre le latch et la bascule:

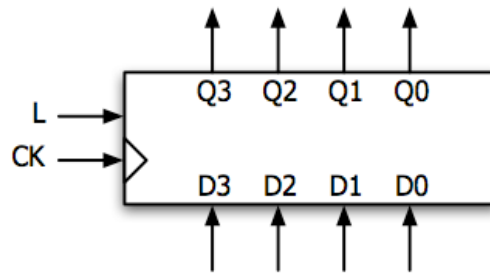


Le registre

- ◆ Un registre est un ensemble de bascules qui partagent le même signal d'horloge, ce qui permet le stockage simultané de plusieurs bits d'information
- ◆ L'existence d'un signal de contrôle **LOAD (L)** permet d'inhiber ou pas le chargement de l'information: à la montée du signal d'horloge, si **L=1**, les signaux d'entrée sont stockés dans le registre
- ◆ Chaque bit d'un registre possède la structure suivante:



◆ Exemple: registre à 4 bits

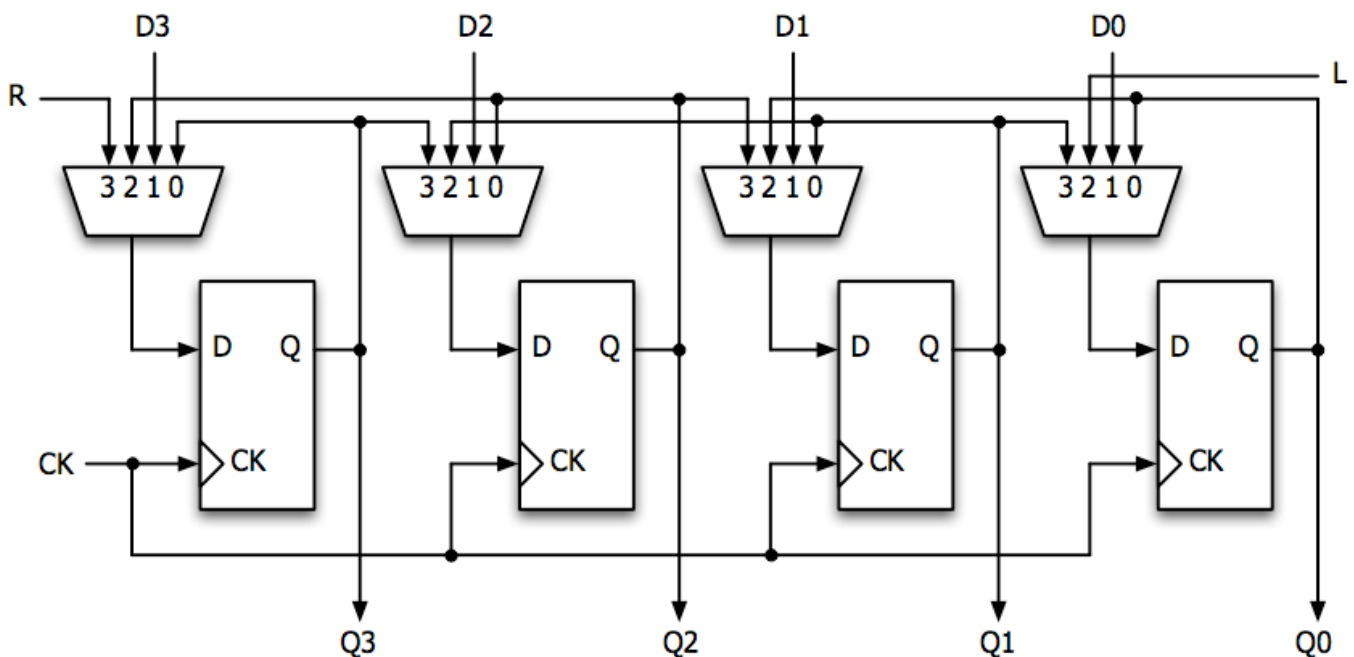


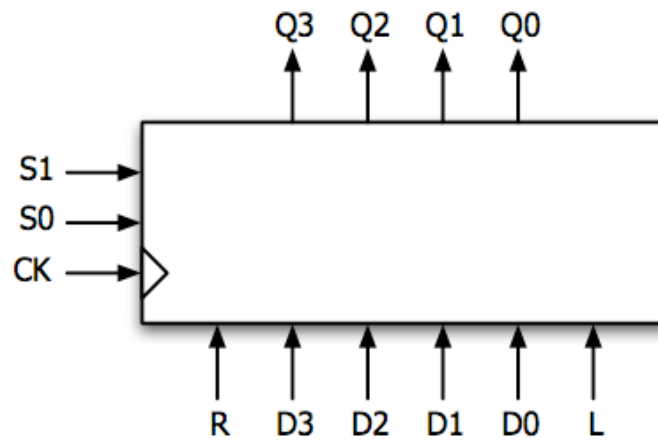
Sa table d'opérations est:

opération	description	L
HOLD	$Q \leftarrow Q$	0
LOAD	$Q \leftarrow D$	1

Le registre à décalage (*shift register*)

◆ Exemple d'un registre à décalage à 4 bits:





S1	S0		
0	0	hold	$Q^+ = Q$
0	1	load	$Q^+ = D$
1	0	shift left	$Q^+ = \ll Q$
1	1	shift right	$Q^+ = \gg Q$

A lire dans Wakerly

- ◆ Chapitre 7
 - 7.2: Latch et bascule
- ◆ Chapitre 8
 - 8.2.5: Registres
 - 8.5.1: Registres à décalage