

CHAPITRE 5

ÉVOLUTION DES FLIPS PLOPS

- Dans les circuits des chapitres 1 à 4 : l'information voyage de gauche à droite entrée → sortie. Il n'y a pas de retour ou "Feed-back".
- On va maintenant voir les circuits avec Feed-back qui permettent de réaliser des circuits de mémorisation.

MÉMOIRE i.e. éléments dont l'état présent dépend de l'histoire des entrées.

Ex : Flips Flops qui basculent d'un état à l'autre
(2 états stables, comme un interrupteur de maison).

{	<ul style="list-style-type: none"> - RS - JK - D ▲ - T 	<p>Plus utiles (plus faciles à comprendre, faciles à fabriquer en silicium (exige peu de place))</p>
---	---	--

- Pour qu'un FF change, il faut que les signaux d'entrées se soient stabilisés suffisamment pendant le "set-up" time avant le coup d'horloge ET juste après le coup d'horloge "hold time".
- FF ⇒ base des circuits synchrones. Ex : compteurs

5.1 Conséquence du Feed-back

- Définition de Feed-back : Retour de la sortie vers l'entrée pour influencer le comportement du système.

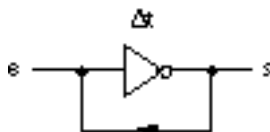
Feed-back positif ou négatif : pousse le système à ses limites

Ex : Des oscillations peuvent apparaître.

- Portes logiques : Elles ont un certain délai de propagation ΔT

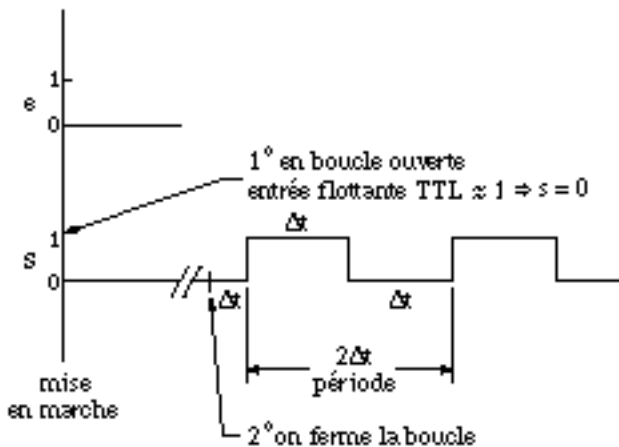
ΔT : essentiel ici

Ex : Soit :



Timing diagram

(chronogramme ou diagramme temporel)



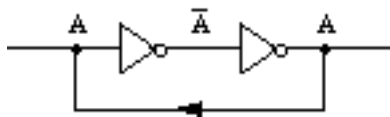
On observe 2 possibilités :

- oscillations 1,0,1,0
- la bascule s'arrête à un état mitoyen "métastable" entre 0 et 1

5.2 Circuits séquentiels

Comme on le voit, la sortie d'un circuit séquentiel dépend de l'entrée présente + des entrées précédentes. Il y a donc un effet de MÉMOIRE.

Reprenons le circuit précédent avec 2 inverseurs.



Cette fois le système est stable.

Sortie \Rightarrow A en entrée \Rightarrow A en sortie

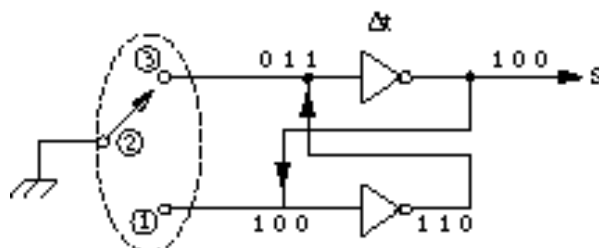
Avec 2 états stables $A = 1$ et $A = 0$, plus d'oscillations

Problème : Tellement stable qu'on ne peut plus modifier sa configuration initiale !

5.3 Suppressions des rebords pour SPDT

- Oscillations d'un interrupteur (switch ou SW) mécanique qui surviennent à chaque changement de position \Rightarrow problèmes, par exemple avec compteurs.

Interrupteur mécanique



- Chaque inverseur a 2 entrées
- SW + Feed-back
- Valeur à la masse l'emporte sur le Feed-back

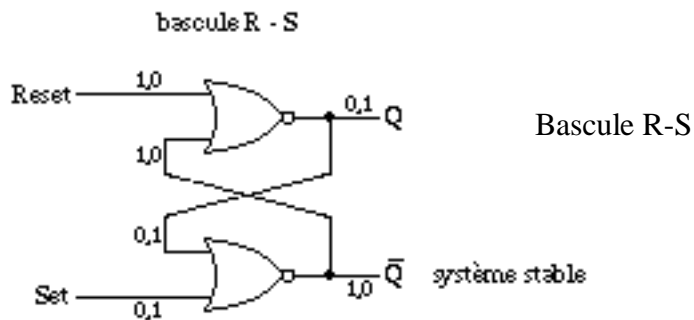
- 1) Supposons SW en bas $\Rightarrow S = 0$
- 2) On bascule le SW \Rightarrow pour un bref instant les 2 entrées des inverseurs sont déconnectées, la sortie reste stable toutefois à cause de l'effet "Mémoire".
- 3) Après un délai, la sortie passe à 1 (autre état stable de la paire d'inverseurs).

Lorsque le contact rebondit en haut, pas d'effet : le circuit a réagi au premier rebond.

Note : On ignore pour l'instant le court-circuit qui survient momentanément.

5.4 Bascules R-S ou Reset-Set

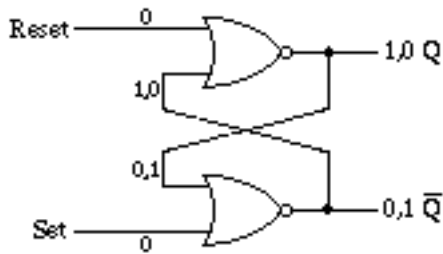
On améliore le schéma précédent comme ceci :



Le problème de $Q = 1$ pour toujours disparaît ici grâce à la combinaison $R = 1, S = 0$

- Cas
- $R = 1, S = 0 \Rightarrow Q \text{ à } 0$
 - $R = 0, S = 1 \Rightarrow Q \text{ à } 1$
- Rappel NOR = 0 si une des entrées est 1

Cas si $S = 0, R = 0$

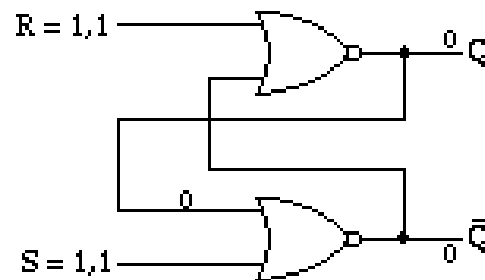


- Supposons $Q = 1 \rightarrow$ stable
- Supposons $Q = 0 \rightarrow$ stable

La bascule a mémorisé un bit !

La table de vérité est la suivante. Dans le cas de bascules on parle plutôt de "table de transition".

R	S	Q	Description
0	0	Q_0	Dépend de l'état précédent
0	1	1	Mise à 1
1	0	0	Mise à 0
1	1	?	Problème !



Oui, c'est stable dans ce cas, cependant essayons le cas $RS = 11$ ou le cas $RS = 00$. On va passer d'abord à 0,1 ou 1,0 sur une échelle en ns avant d'arriver à 00 (un signal est plus rapide que l'autre).

Donc 2 cas :

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

← Ou
← changement

Cas 1

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

← Cas
← intermédiaire

Cas 2

Le problème est le suivant :

"00" cas pas de changement donc le cas 1 est un désastre $RS = 11 \rightarrow Q = 0$ et $RS = 00 \rightarrow Q = 1$

On ne peut s'assurer que les 2 entrées R et S, changent en même temps.

Limitation : $\left\{ \begin{array}{l} \bullet \text{ cas } RS = 11 \text{ (pas permis, INTERDIT)} \\ \bullet \text{ seule une entrée change à la fois} \end{array} \right.$

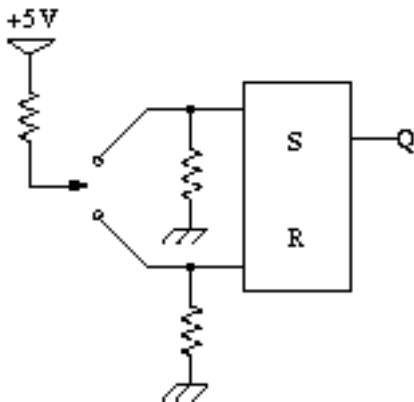
Avantages d'interdire $RS = 11$:

- Avec $RS = 11$ on viole la règle $S = 1 \Rightarrow Q = 1$
- Aussi en empêchant $RS = 11$, on a pour les autres cas que la sortie du second NOR est \bar{Q} (inverse de Q). Une sortie est l'inverse de l'autre.

On a ainsi la table de transition :

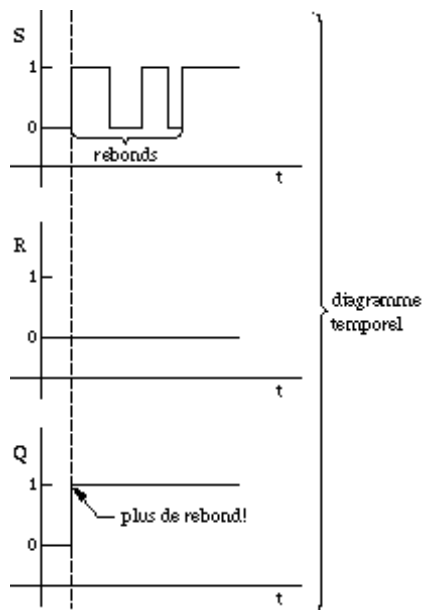
R	S	Q	\bar{Q}	Description
0	0	Q_0	\bar{Q}_0	État précédent, pas de changement
0	1	1	0	Mise à 1
1	0	0	1	Mise à 0
1	1	Interdit	Interdit	Interdit

Ex : Circuit anti-rebond avec bascule R-S



- R S
- Au repos : 0 0 $\Rightarrow Q_0$
 - Contact : 0 1 – en haut $\rightarrow Q = 1$
1 0 – en bas $\rightarrow Q = 0$

Valeur finale dès le 1^{er} rebond :



5.5 États

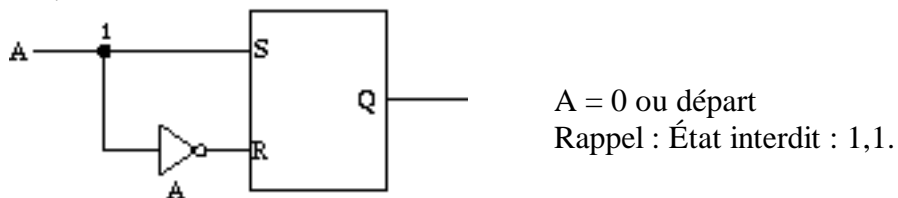
Définition : L'état d'une bascule correspond à une de ses conditions stables, $Q = 1$ (état à 1) ou $Q = 0$ (État à 0).

Ex : Pour une bascule SR on obtient un état instable si les entrées $RS = 11$ et des états stables si $RS = 10$ ou 01 .

5.6 Bascules R-S avec horloge

La bascule R-S est très sensible aux changements d'entrées. Essayons donc d'améliorer le design.

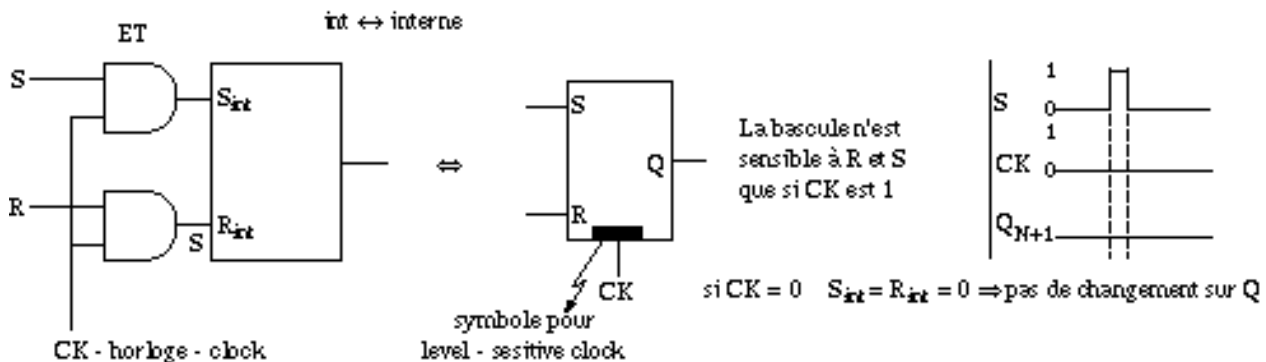
Ex :



- En principe l'inverseur devrait nous empêcher d'avoir la condition interdite $S = 1 = R$ sauf qu'à cause du délai Δ "glitch", "aléa" on l'aura pour un bref instant si $A : 0 \rightarrow 1$
- À noter : jamais d'état "00" \Rightarrow pas de mémorisation.

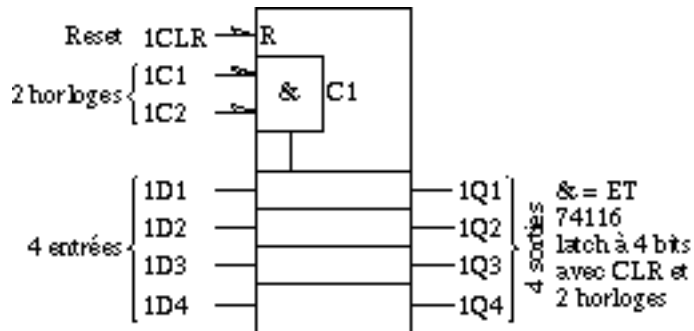
5.7 Horloge sensible aux transitions de niveaux logique (Level sensitive clock)

Une façon de rendre la bascule insensible aux aléas, c'est de lui ajouter une horloge. Lorsque l'horloge change \Rightarrow bascule change :



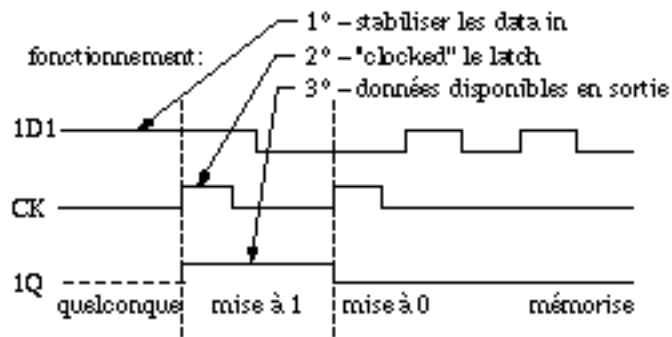
5.8 "D" LATCH

L'IEEE a standardisé les symboles des bascules

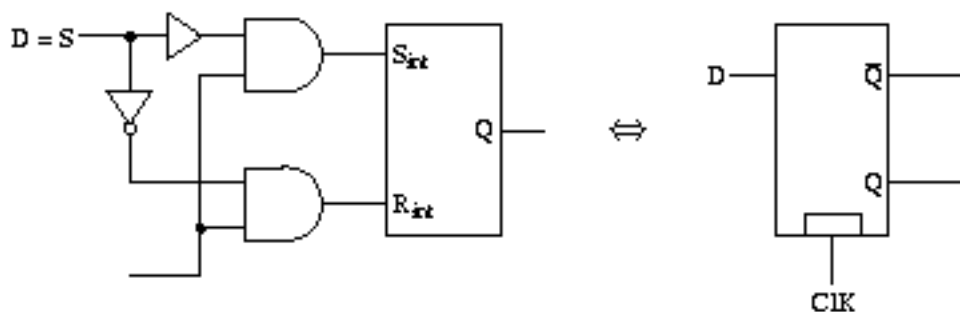


Définition :

Latch : type de bascule sans horloge ou avec une horloge de type "Level-sensitive" (sensible aux transitions).



c'est ce qu'on appelle un latch "D"



- Si $CIK = 1$ la bascule a un effet transparent sortie = entrée. Il faut rajouter un tampon (buffer) pour avoir un même délai dans les 2 branches.
- Si $CIK = 0$ la bascule mémorise son dernier état.

5.9 timing

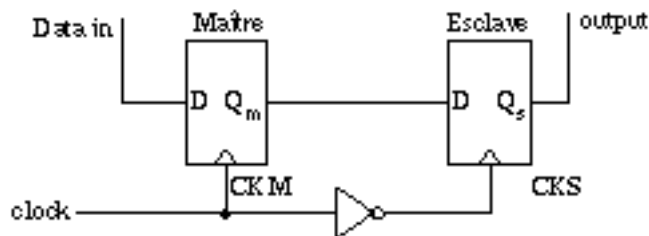
Il faut une durée d'impulsion minimum pour que la bascule réagisse.

Ex : < 25ns pour 7476 \Rightarrow pas de changement
74F112 \rightarrow 4.5ns seulement

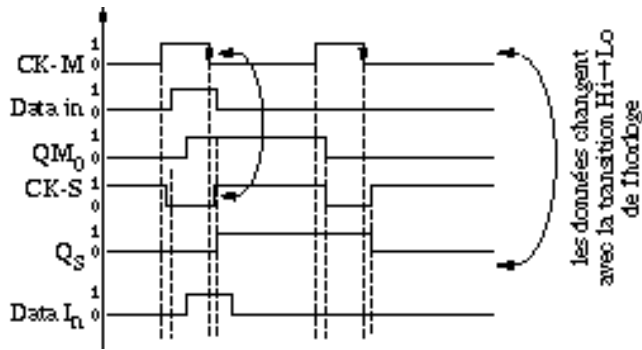
5.10 Flip Flop Maître-Esclave

On a besoin d'une bascule qui réagit uniquement aux coups d'horloge, cela évite les "courses" lorsque plusieurs bascules sont reliées les unes aux autres car actuellement avec une "level sensitive clock" les sorties peuvent changer si $CK = Hi$!

- Façon de faire pour corriger ceci : relier 2 bascules "D" ensemble. C'est ce qu'on appelle la configuration Maître-Esclave (master-slave).

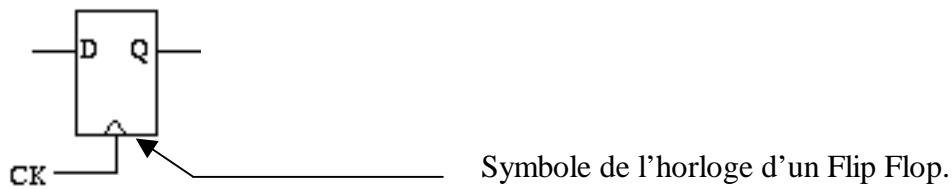


- 1- Data = Hi et clock = Hi \rightarrow latch maître transparent et $Q_M = Hi$
- 2- CKM = LO et maître mémorise la donnée
- 3- À cause du délai de propagation Δ , les 2 horloges restent à LO pendant Δ . CKM = CKS = 0. Il n'y a pas de "course".
- 4- CKS = Hi et sortie Q_s enregistre la donnée de Q_M
- 5- Latch maître s'ouvre à nouveau au coup d'horloge suivant et prend $Q_M = LO$



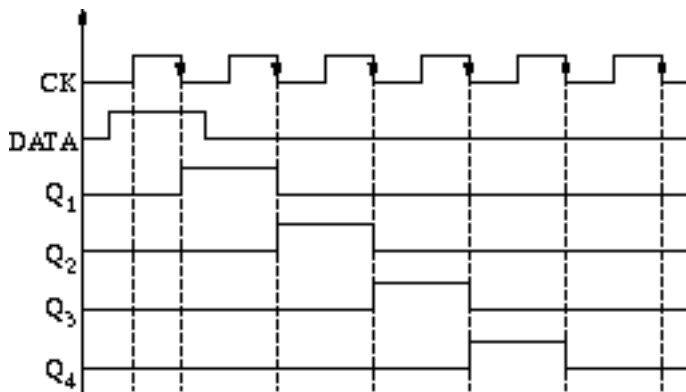
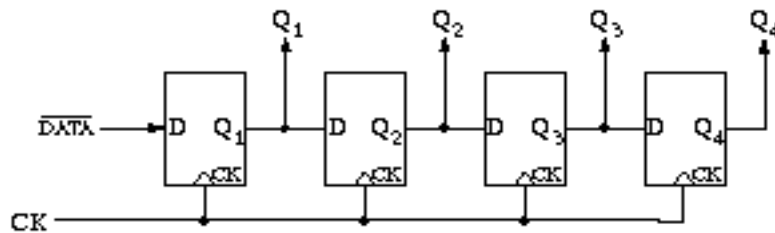
Note :

- Un latch → réagit aux transitions de niveaux.
- Un Flip Flop → réagit au coup d'horloge (symbole : triangle).



Les Flips Flops peuvent servir à faire des registres à décalage (shift register sans course).

Ex :

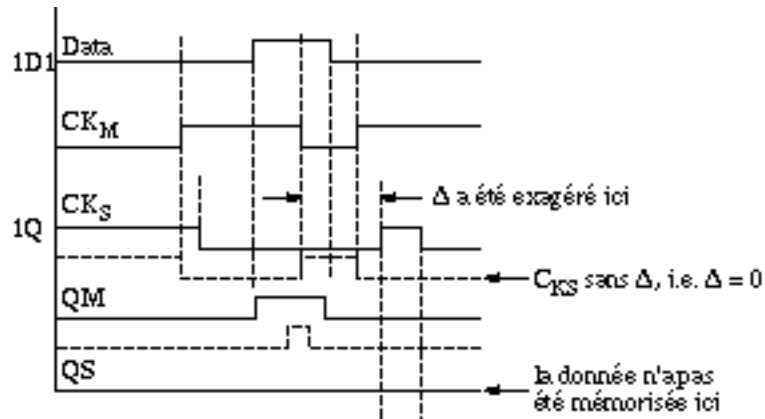


Note : La donnée rentre de façon asynchrone dans la bascule et en ressort synchronisée.

Désavantages : Chaque M/S (Master/Slave/Unit) nécessite 2 fois plus de hardware qu'un design "à transition" qui provoquerait ici des "courses" avec les "latches ouverts".

5.11 Vitesse d'horloge maximum dans Flip Flop Maître-Esclave

Pour que l'unité M/S fonctionne correctement ? il faut que la durée où l'horloge passe à "LO" soit supérieure au temps de propagation de l'inverseur.



Supposons un "duty cycle" de 50% \Rightarrow durée à LO au moins égale à $\Delta \Rightarrow$ période minimum $= 2\Delta$.

Donc :

$$\Rightarrow F_{MAX} = \frac{1}{2\Delta} \quad \text{ex : si } \Delta = 25 \text{ ns}$$

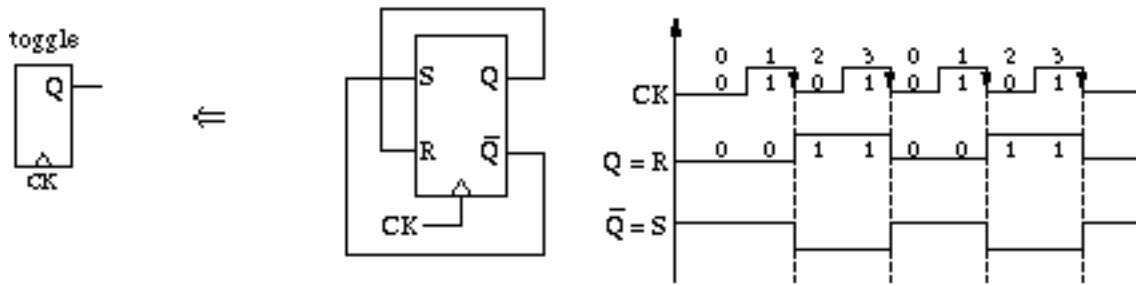
$$F_{MAX} = 25 \text{ MHz}$$

D'autres facteurs entrent en ligne de compte dans le calcul.

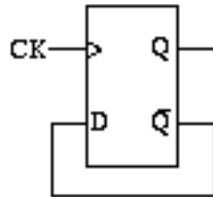
5.12 Flip-Flop Toggle

On peut employer la configuration M/S en reliant Q et \bar{Q} de façon croisée avec R et S, c'est le Flip Flop toggle qui change à chaque transition Hi-Lo-Hi-LoHi ...

"Toggle" signifie qui bascule d'un état à l'autre à chaque coup d'horloge.



ou mieux:

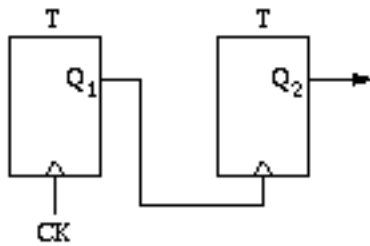


Notes :

? 1 seule entrée : ck peut-être vu comme un compteur à 2 bits

Q	Ck
MSB	LSB

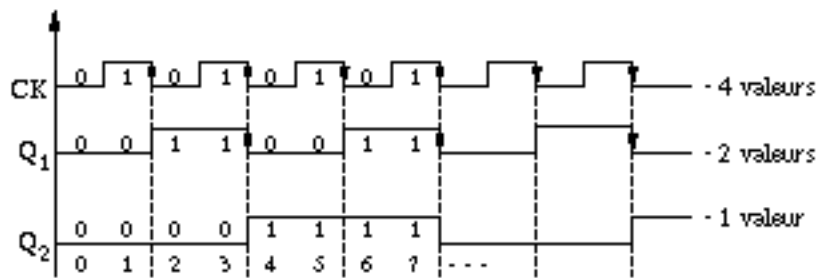
On peut créer un compteur à 3 bits de type "Ripple" (à débordement, l'horloge se propage) avec 2 bascules T en cascade :



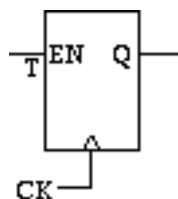
Compteur à débordement

Ripple compteur 3 bits

Ou "true" (vrai) compteur 2 bits en excluant l'horloge



- Flip Flop T : diviser fréquence de l'horloge en 2.
- Avec N Flip Flop en cascade \Rightarrow la fréquence de l'horloge est divisée par 2^N
- On a aussi la Bascule T avec un Enable :



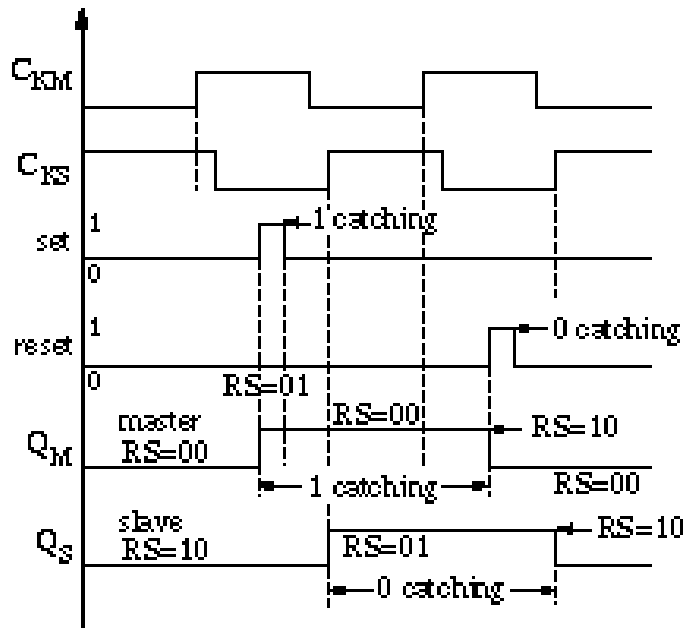
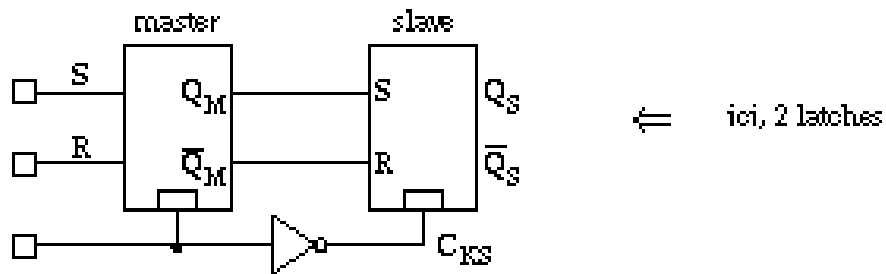
T=EN	Ck	Q_{N+1}
0	X	Q_N
1	actif	Q_N

Utile pour créer certains compteurs (voir plus loin).

5.13 Master-Slave R-S Flip flop et "1 catching" "0 catching"

"1 catching" et "0 catching" signifient que l'on attrape les 1 et les 0.

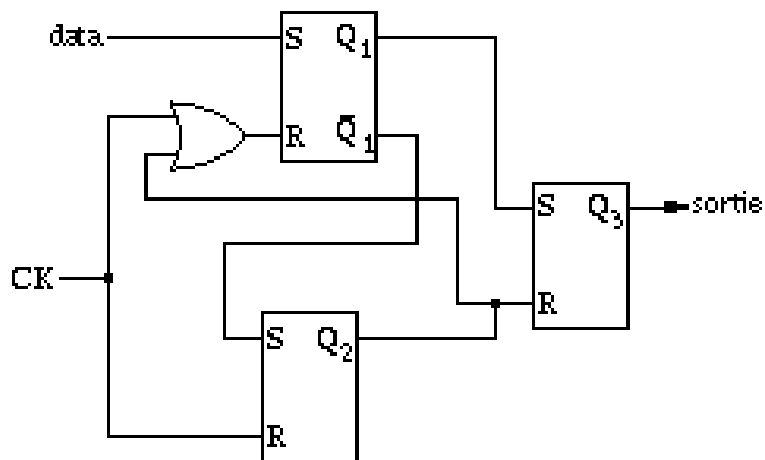
Les latches R-S Maître-Esclaves peuvent être sensibles aux "glitches", c'est ce qu'on appelle les 1 catching (pulse sur S) et 0 catching (pulse sur R).



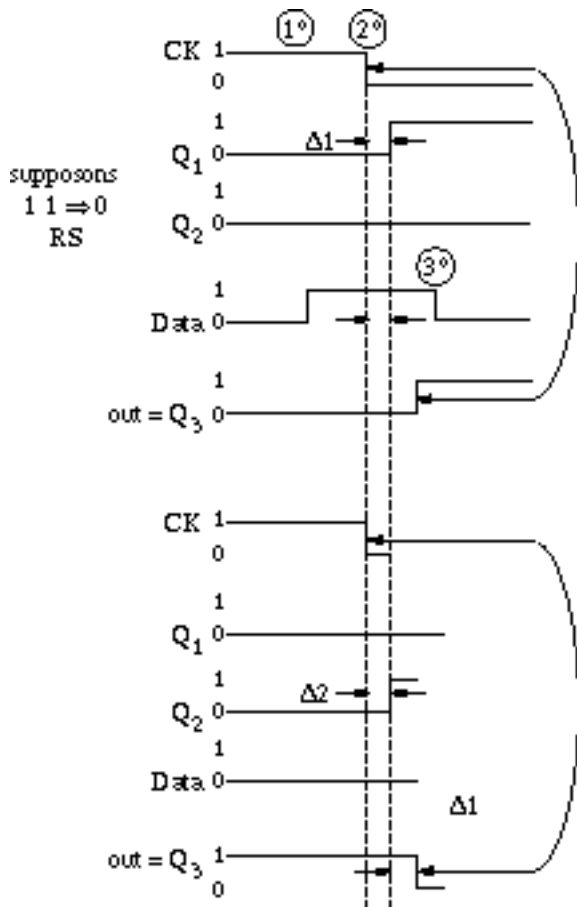
Ceci se passe à cause de la "transparence" de la bascule lorsque le signal d'horloge est à Hi.

5.14 Design pour Flip Flop "D" ("edge-triggered")

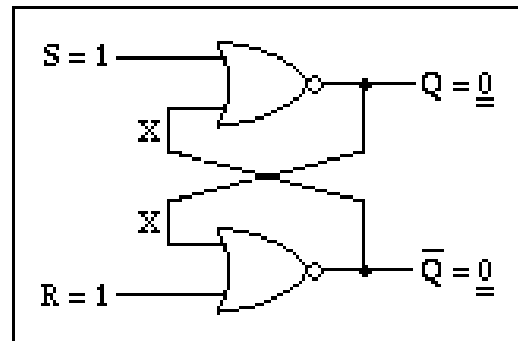
- Une paire Master/Slave n'est pas la seule façon de créer une bascule D "edge-triggered".
- Le design suivant prend avantage du délai des 3 latches SR, une porte OU et du Feed-Back.



- 1) Cas ck = hi $\rightarrow (Q_1, Q_2 = 0)$ Reset $\Rightarrow (S, R)$ de $Q_3 = 0 \rightarrow$ **SORTIE INCHANGÉE**
- 2) Supposons OU T = LO et Data = Hi et $Z = ck (Hi \rightarrow LO)$, puisque $Q_1 = Q_2 = 0 \Rightarrow [RS = 0, 1]$
 $\Rightarrow Q_1 = 1 \Rightarrow Q_3 = 1$ **MISE À 1**
 \Rightarrow état stable
- 3) CK = <0 et Data change de Hi \rightarrow LO ; (S, R) de $Q_1 = (0,0)$ pas de changement la pin S de Q_3 est inchangée.
- 4) Transition horloge (Hi \rightarrow LO) et Data = LO et sortie = Hi
 Cas 1 [$Q_1 = 1, Q_2 = 0$] impossible car CK (Hi \rightarrow LO (était à Hi) après transition
 Cas 2 [$Q_1 = 0, Q_2 = 0$] on avait [$SR = 0,1$ sur Q_1] \rightarrow sur Q_3 [$SR = 0, 0$] $\Rightarrow Q_1 = 0$ et $Q_2 = 1 \Rightarrow Q_3 = LO$



Rappel :



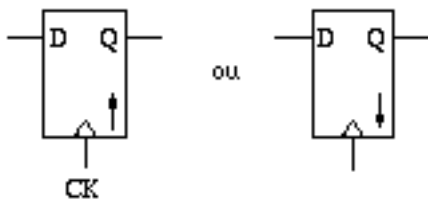
Note : Il faut ici que la bascule RS soit à 0 avec $R = S = 1$

Supposons :

- 1- $ck = Hi$
- 2- $Q_3 OUT=LO$, $Data = Hi$, $ck (Hi \rightarrow LO)$
- 3- $Ck = LO$, $Data = (Hi \rightarrow LO)$
- 4- $Ck = Hi \rightarrow LO$, $Data = LO$, $sortie = Hi$

5.15 Importance des Flip Flops D

schéma

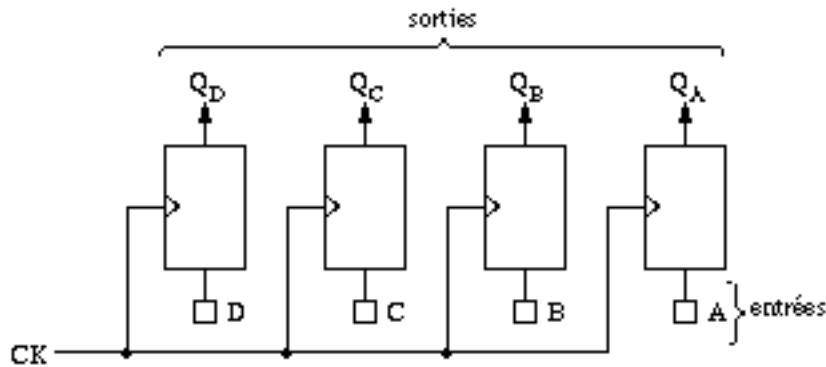


Rising or falling edge triggered selon le sens de la flèche.

Spécifications :

- durée minimum d'horloge 4.5 ns
 - temps maximum de propagation 6.5 ns
- Le Flip Flop "D" est le type de Flips Flops le plus important en circuit numérique :
- moins de place sur silicium (par rapport à un Flip Flop JK, voir plus loin)
 - 1 seule entrée ⇒ + facile à programmer
 - Flip Flop JK a souvent trop de capacité pour ce qui est requis (overdesigned)

5.16 Flips Flops D en parallèle = registre



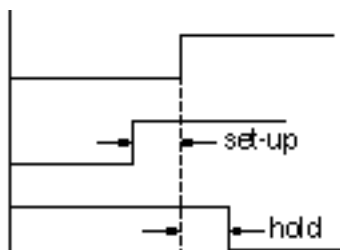
Registre PI PO
Parallèle In Parallèle Out

Mémoire la valeur présente en [D, C, B, A] au coup d'horloge et reste constant jusqu'au prochain coup d'horloge.

5.17 Timing des Flips Flops synchrones

Que se passe-t-il si les données changent au même moment que l'horloge ?

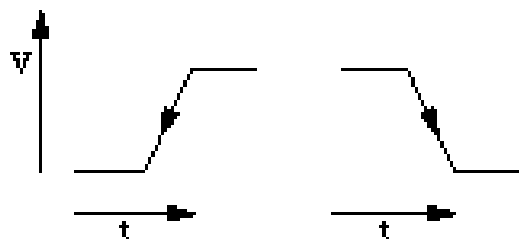
Rép. La donnée doit être présente un peu avant ET un peu après le coup d'horloge.



Autre paramètre d'importance : durée de l'impulsion d'horloge, c'est le paramètre t_w

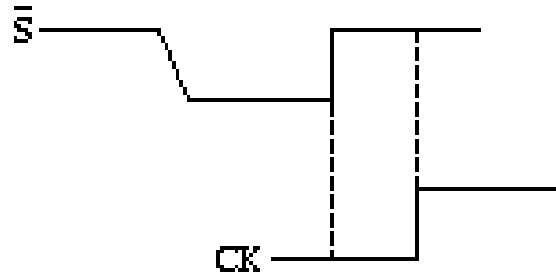
Ex : 74112, $t_w = 20$ ns minimum.

Slewrate : Pour le signal d'horloge, la transition arrive à un "slew Rate" (ou vitesse) minimum en $[V/s]$ de changement. Si l'entrée est trop lente l'horloge ne sera peut-être pas reconnue comme valable.



Autre paramètre "minimum recovery time" (temps minimum de rétablissement) t_{rec}

t_{rec} doit s'écouler après un changement d'entrée asynchrone pour que l'horloge change à nouveau.



74F74

Ex : $t_{setup} = 2ns$

$t_{hold} = 1ns$

$t_w = 4ns - 7 ns ?$ durée impulsion horloge

$t_{rec} = 2ns$

$$F_{max} = 125 \text{ MHz et } \Rightarrow \frac{1}{125 \text{ MHz}} \approx 8ns \approx t_{setup} + t_{hold} + t_p$$

$$2 + 1 + 5$$

5.18 Bascule JK

Un autre type de bascule est la bascule JK, symbole :

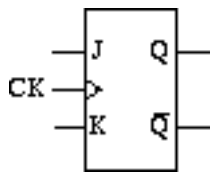


Table de transition

Q	J	K	Q _{futur}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

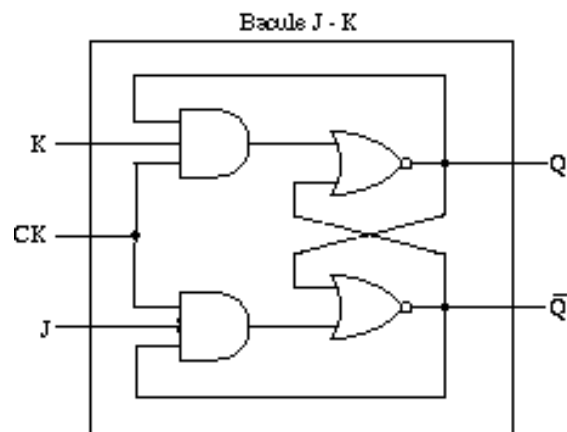


Table de transition

J	K	QN +1
0	0	Pas changement
0	1	0
1	0	1
1	1	Toggle ← On a réglé le cas "interdit" de la bascule R-S !