Systèmes Digitaux

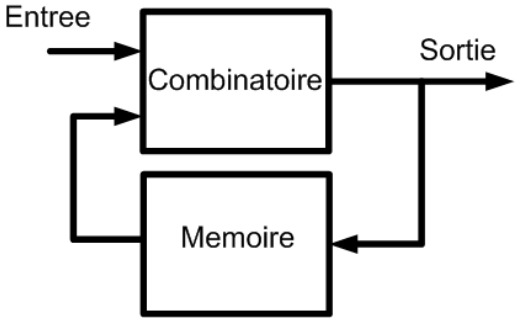
Cours 5

Logique séquentielle

- Jusqu'à présent, on n'a parlé que des circuits combinatoires:
  - Les sorties ne dépendent que des entrées
- Aujourd'hui, on va parler de la logique séquentielle:
  - Les sorties dépendent des entrées et du passé
  - Ça implique qu'il doit avoir une certaine mémoire...

Combinatoire

Séquentiel



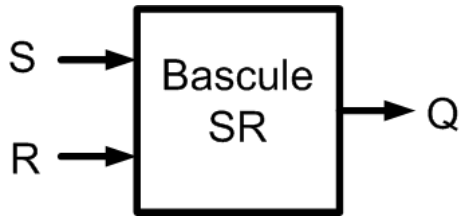
Logique séquentielle

- On sépare les éléments mémoire en catégories:
  - Bascule (“latch”)
  - Flip flop
- Aujourd’hui, on va parler des éléments suivants
  - Bascule SR avec et sans autorisation
  - Bascule D
  - Flip flop D

Commençons avec la bascule SR

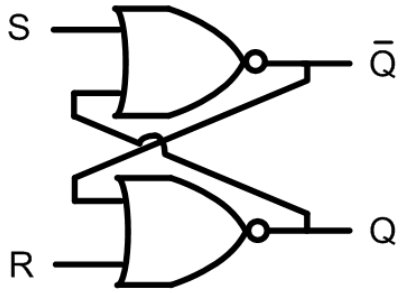
Bascule SR

- Le nom SR vient des termes anglais “set-reset”
- Ce sont ses 2 entrées:
  - Une entrée “SET” qui met la sortie à ‘1’
  - Et une entrée “RESET” qui met la sortie à ‘0’
  - Si aucune des entrées n’est active, elle garde sa valeur précédente
- La bascule SR peut être faite avec NOR ou avec NAND



Bascule SR

- En utilisant 2 NOR, on obtient ceci:



- Comment se rappeler du schéma?
  - On a 2 entrées S et R: une dans chaque NOR
  - La sortie l'un entre dans l'entrée de l'autre
  - S est devant Q' et R est devant Q

Pour activer SET, ou RESET on met '1'...

Comment analyser?

- Pour analyser le comportement d'un système séquentiel, l'approche est un peu différente
- Pourquoi?
  - La sortie dépend aussi de ce qu'il y a dans la mémoire. Quelle est la valeur de l'état?
  - Il y a plus d'information à gérer
  - Un changement peut en provoquer un autre qui peut en provoquer un autre...
  - Ce n'est pas évident de savoir quand on a terminé

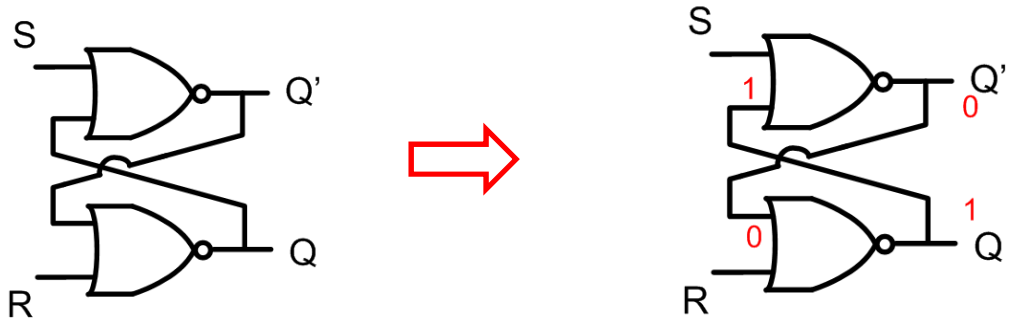
Méthode

- On se donne un état initial
  - Sans état de départ, on ne peut rien faire..
- On essaie toutes les entrées possibles
- Pour chaque cas:
  - On propage les signaux partout dans le circuit
  - Si quelque chose change, on poursuit l'analyse
  - On continue jusqu'à ce que plus rien ne change

Allons essayer ça...

Comment analyser?

- On se donne un état de départ Q=1 et Q'=0



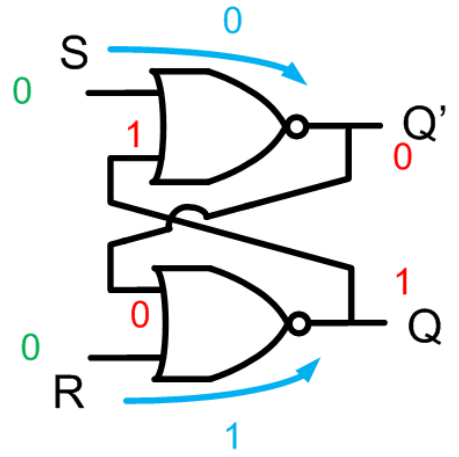
- Avant de passer à l'autre diapo, rappelons-nous de ce que fait un NON-OU

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

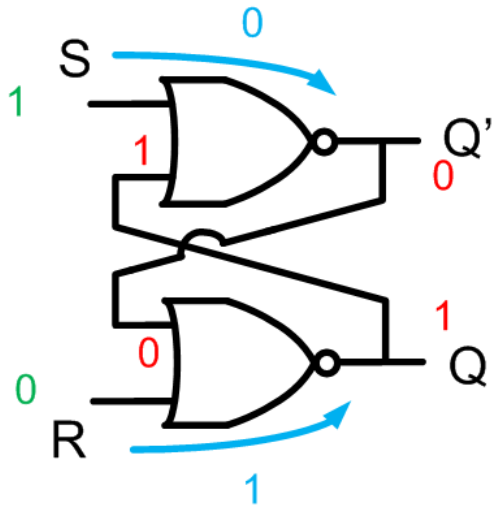
Essayons maintenant
toutes les entrées possibles

Comment analyser?

- Si l'entrée devenait S=0, R=0
  - Le NOR du haut donnerait '0'
  - Le NOR du bas donnerait '1'
  - Rien n'a changé... on a fini



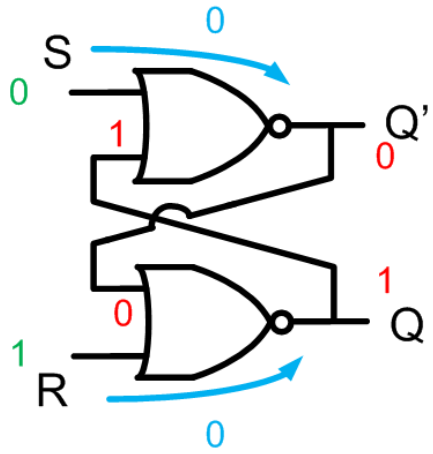
- Si l'entrée devenait S=1, R=0
  - Le NOR du haut donnerait '0'
  - Le NOR du bas donnerait '1'
  - Rien n'a changé... on a fini



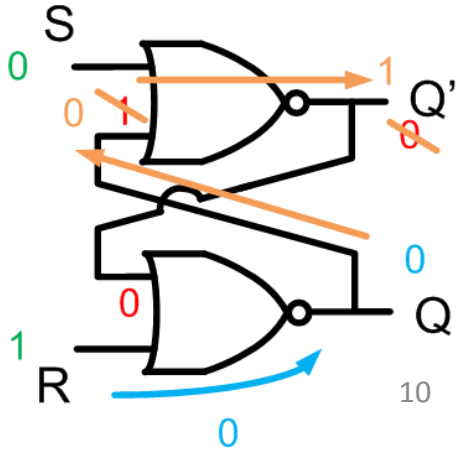
Quand les sorties restent les mêmes, l'analyse est terminée...

Comment analyser?

- Si l'entrée était S=0, R=1
  - Le NOR du haut donnerait '0'
  - Le NOR du bas donnerait '0'
- Ici, la sortie du bas a changé...
  - Donc, Q n'est plus '1' mais est rendu '0'
- Est-ce que la sortie du haut est encore bonne?

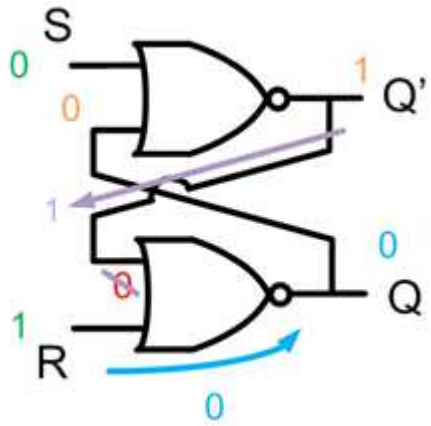


Maintenant la sortie du haut a changé!

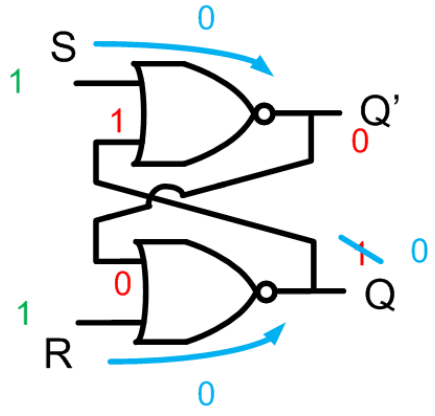


Comment analyser?

- La sortie du bas ne change plus
  - Le système s'est stabilisé
  - L'analyse est complète

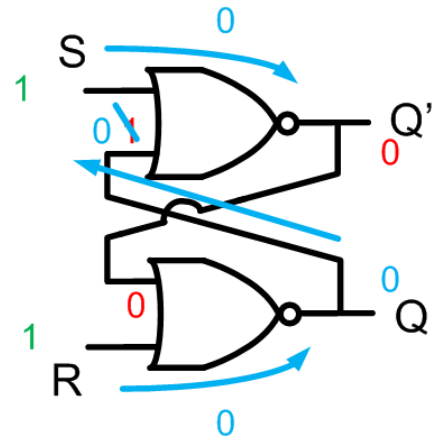


- Si l'entrée etait S=1, R=1
  - Le NOR du haut donnerait '0'
  - Le NOR du bas donnerait '0'
  - La sortie du bas a changé
  - Est-ce que celui du haut est encore bon?



Comment analyser?

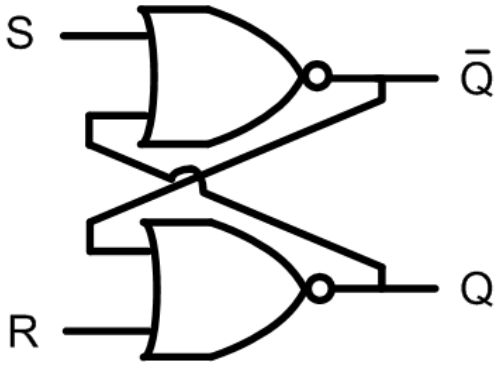
- On continue la propagation des signaux et la sortie est encore ‘0’:
  - L’analyse est terminée



- Note:
  - Les sorties Q et Q’ devraient être complémentaires
  - Puisque les 2 sont égaux, il y a contradiction
  - Pour cette raison, on dit que l’entrée “11” est interdite

Bascule SR

- On peut reprendre l'analyse avec d'autres valeurs et états initiaux...
  - La table de vérité de la bascule SR avec NOR nous donnera toujours ceci...

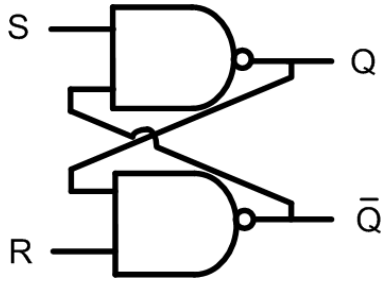


S	R	Q	Q'
0	0	Valeur précédente	Valeur précédente
0	1	0	1
1	0	1	0
1	1	0 (interdit)	0 (interdit)

On peut aussi faire une bascule SR avec des portes NAND

Bascule SR

- En utilisant 2 NAND, on obtient ceci:



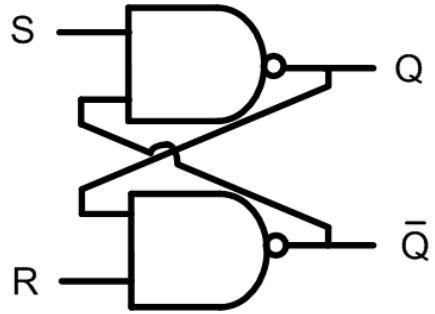
- Comment se rappeler du schema?
  - On a 2 entrées S et R: une dans chaque NAND
  - La sortie de l'un va dans l' entrée de l'autre
  - S est devant Q et R est devant Q' (pas comme avec NOR)

Pour activer SET, ou RESET on met '0'

On peut dire que c'est "actif BAS" ou "actif à '0'"

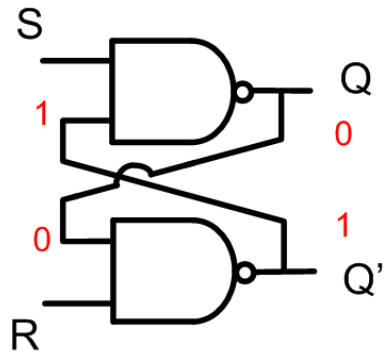
Exemple (seul)

- Supposez que Q=0 et Q'=1 au tout début
  - Trouvez les valeurs de Q et Q' pour toutes les 4 combinaisons d'entrées possibles

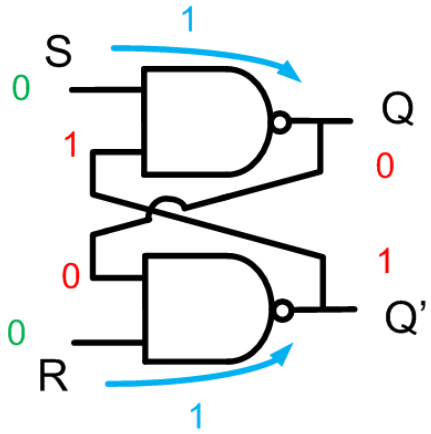


Exemple (seul)

- On commence avec l'état de départ:

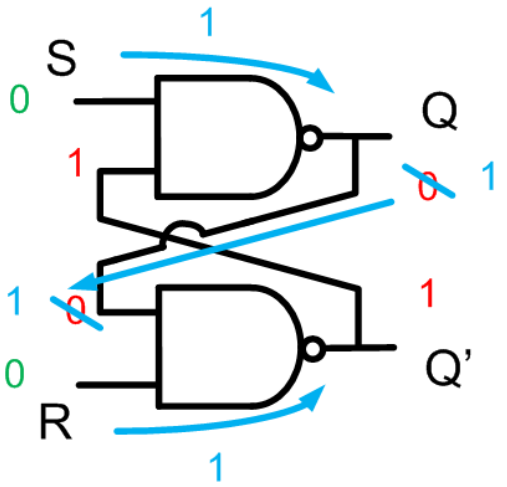


- On met les entrées et on détermine les signaux de sortie:



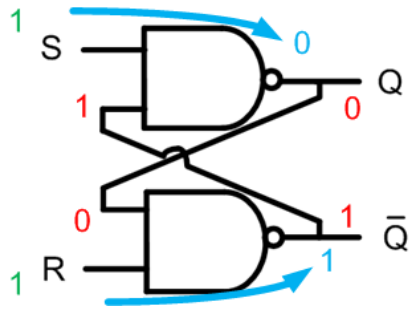
Exemple (seul)

- Le bas ne change pas la sortie
- Le haut change la sortie a ‘1’
  - Il faut continuer l’analyse
- Le ‘1’ se propage en bas
  - Mais le ‘1’ ne change pas la sortie
  - Le système est stable et plus rien de change... C’est fini
- Quand l’entrée est “00”, la sortie est “11”
  - Ce n’est plus Q et Q’... Cette entrée est donc interdite

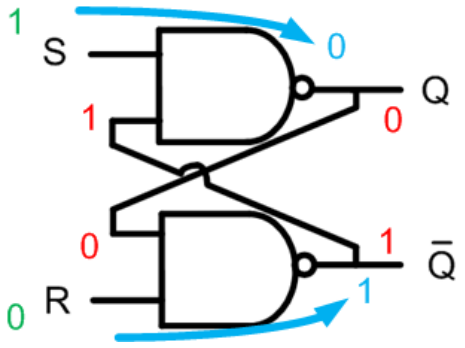


Exemple (seul)

- Avec SR=11 en entrée:
  - Les sorties ne changent pas

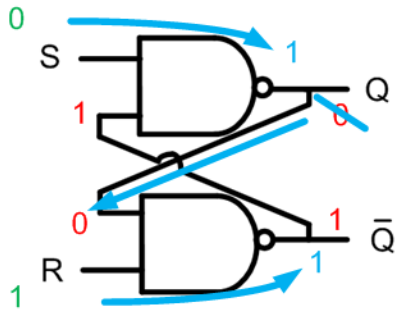


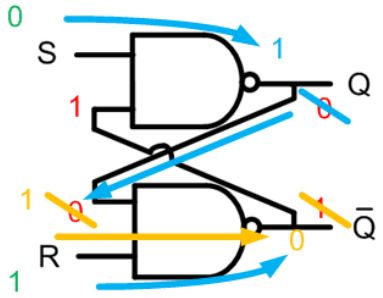
- Avec SR=10 en entrée:
  - Les sorties ne changent pas



Exemple (seul)

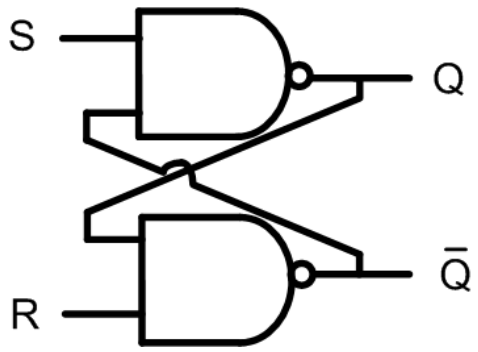
- Finalement, avec SR=01 en entrée:
  - Le bas ne change pas
  - La sortie du haut change a 1
  - Ça se propage vers la section du bas
  - La propagation change la sortie du bas
  - Cette nouvelle sortie change la section du haut
  - Cependant, ce changement ne change pas la sortie du haut
  - Notre analyse est terminée...





Bascule SR

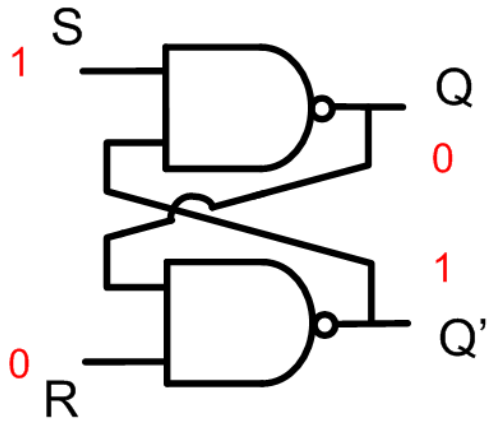
- On peut faire l'analyse avec toutes les entrées et condition de départ possibles
- On se retrouverait avec la table de vérité suivante:

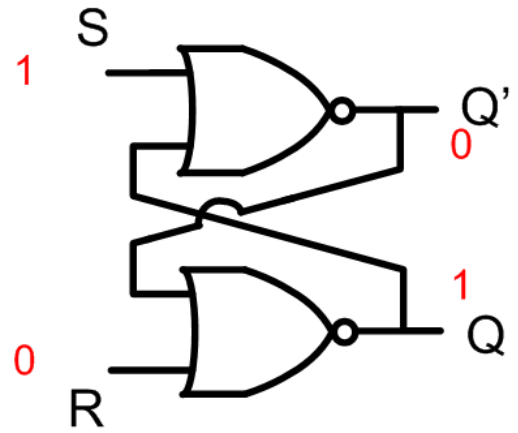


S	R	Q	Q'
0	0	1 (interdit)	1 (interdit)
0	1	1	0
1	0	0	1
1	1	Valeur précédente	Valeur précédente

Parenthèse: Rétroaction

- Plusieurs types d'éléments memoire utilisent une rétroaction positive
  - "Feedback positif": quand la sortie revient à l'entrée et RENFORCE la réponse
  - C'est ce qu'on a avec la bascule SR

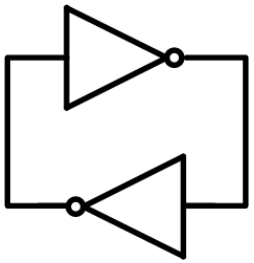




On peut aussi faire la même chose avec des inverseurs...

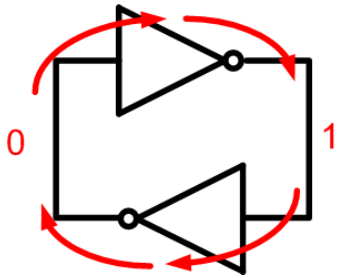
Parenthèse: Rétroaction

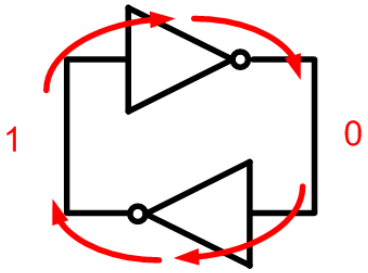
- En connectant la sortie à l'entrée, on obtient un feedback positif:



Comme les SRAM

- Ça nous donne aussi un élément "memoire"

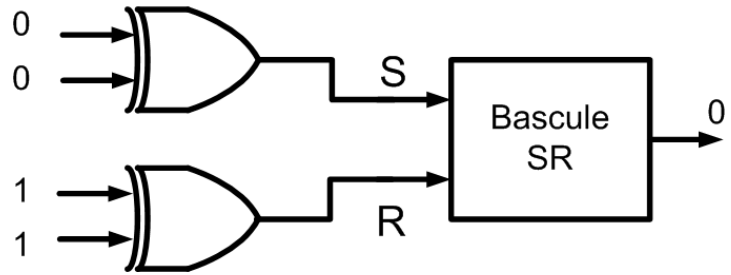




On ferme la parenthèse

Un problème...

- Les bascules SR peuvent avoir un problème
- Pensons par exemple à un circuit comme ceci
  - La sortie est 0

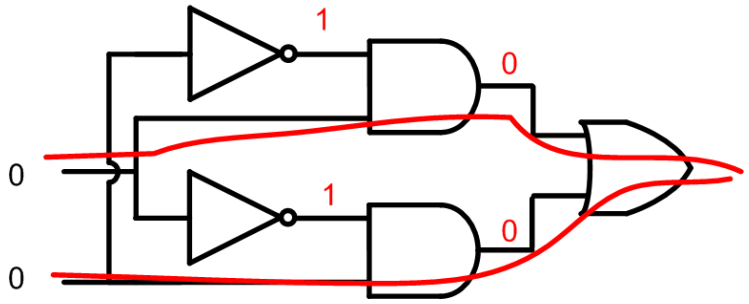


  - Qu'arrive-t-il si les DEUX entrées du haut changeaient de "00" à "11"?
  - Idéalement, rien ne devrait changer...

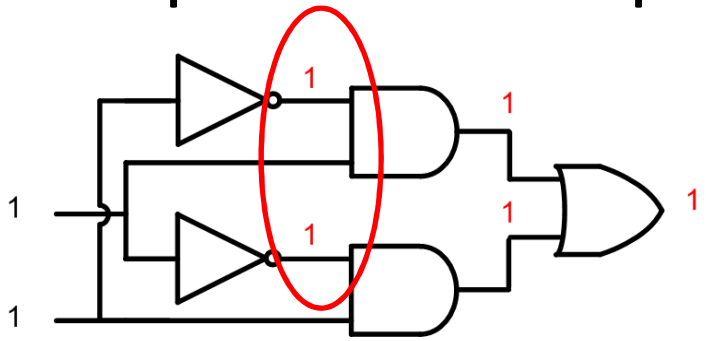
Mais ça c'est dans un monde idéal...

Un problème...

- Une porte XOR ressemble à ceci:



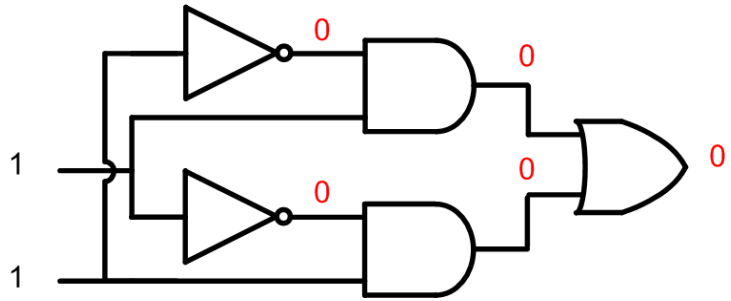
- En changeant les entrées à 1, les sorties des inverseurs n'ont pas eu le temps de changer...



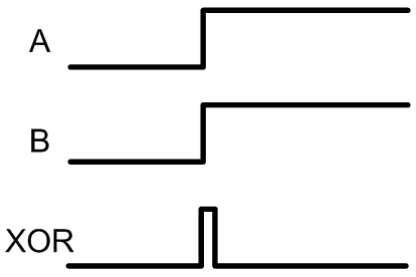
La sortie sera '1' momentanement

Un problème...

- Quand les ‘1’ de l’entrée auront passé par l’inverseur, la sortie deviendra ‘0’

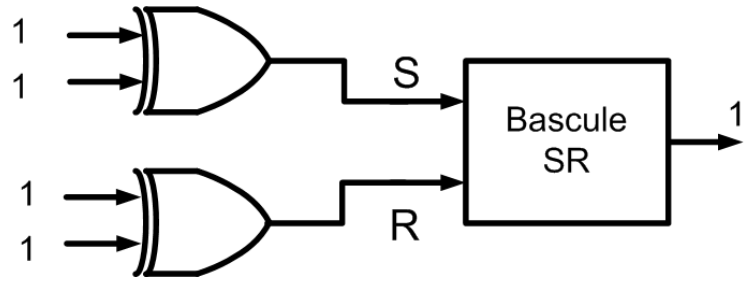


- Cette sortie ramènera la sortie à 0...



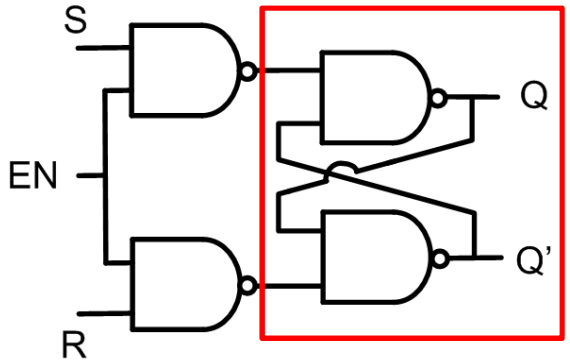
Un problème...

- En changeant de 00 a 11, rien n'aurait dû changer...
  - À cause des délais, S est monté à '1' momentanément
  - Ça fait changer la sortie à 1
  - Par la suite, S retombe à 0
- Ça aurait du rester a 0...
  - Comment fait-on pour régler le problème?



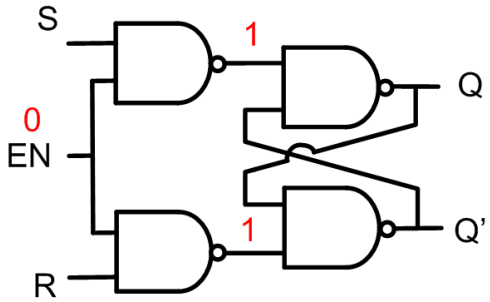
Bascule SR avec autorisation

- Ce serait bien d'avoir une facon de bloquer les entrées SAUF quand on le permet
  - Comme ça, les changements momentanés n'auraient pas d'effet
- Pour ce faire, on peut utiliser la bascule avec autorisation ("enable")



Bascule SR avec autorisation

- L'analyse est assez simple:
  - Si EN=0, la sortie des 2 premiers NANDs sera 1
  - La bascule SR garde sa valeur précédente



- Si EN=1, la sortie des NAND dépendent de S et de R...
  - Dans ce circuit S et R sont actifs haut (S=1 ou R=1)...

NOTE: c'est facile d'oublier lequel est actif '1' ou '0'... Et où les Q et Q' se trouvent!

Exemple (seul)

- Faites la “table de vérité” d’une bascule SR avec autorisation:

S	R	EN	Q	Q’
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

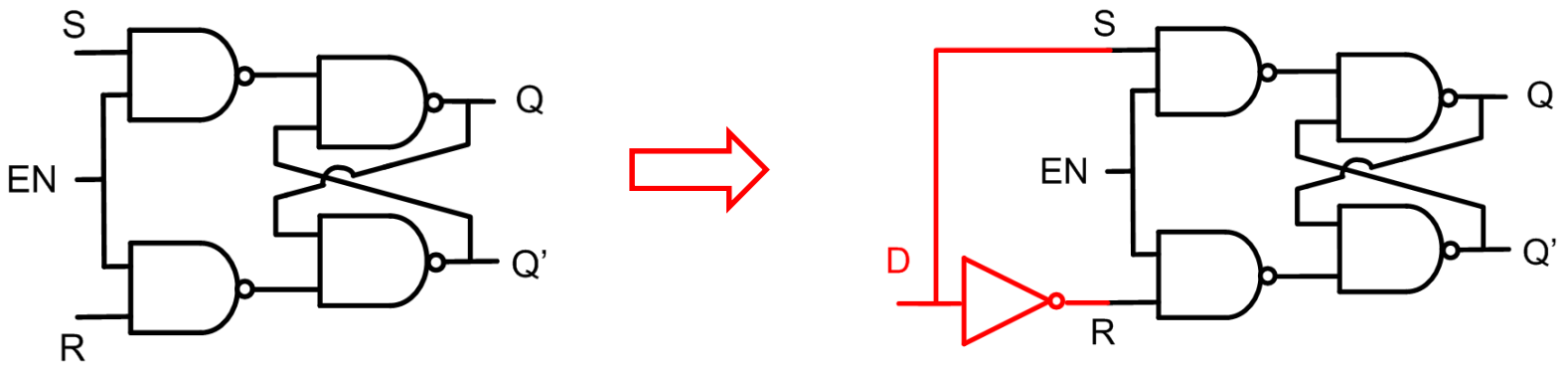
Exemple (seul)

- On remplit:
  - Quand EN=0, ca garde sa valeur précédente

S	R	EN	Q	Q’
0	0	0	V. Prec.	V. Prec.
0	0	1	V. Prec.	V. Prec.
0	1	0	V. Prec.	V. Prec.
0	1	1	0	1
1	0	0	V. Prec.	V. Prec.
1	0	1	1	0
1	1	0	V. Prec.	V. Prec.
1	1	1	1 (Interdit)	1 (Interdit)

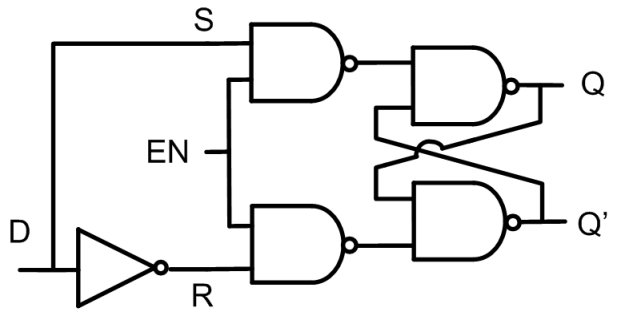
Bascule D

- À la place de contrôler une mémoire avec S et R, il est possible d'utiliser **un** signal de donnée
  - C'est ce qu'on appelle une bascule D
- On reprend la bascule SR avec autorisation
  - On ajoute un inverseur
  - Et on connecte les entrées ensemble…



Bascule D

- Quand le enable est 0, les 2 entrées du 2e SR sont ‘1’:
  - La sortie conserve sa valeur précédente
- En mettant EN=1, la valeur de D peut changer la sortie
  - Avec D=1, on a S=1 et R=0: la sortie sera 1
  - Avec D=0, on a S=0 et R=1: la sortie sera 0

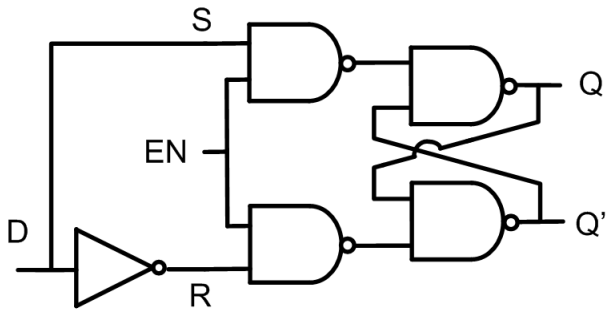


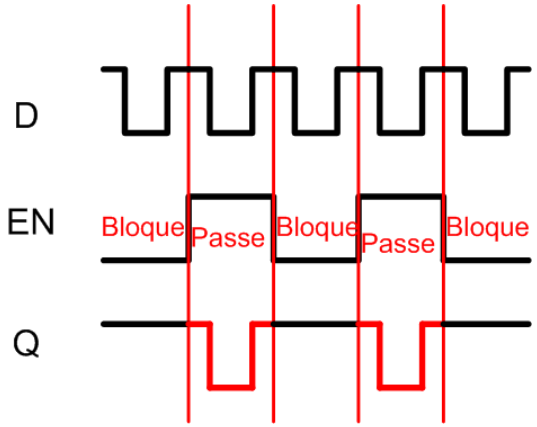
Technique d'analyse

- Identifier clairement les zones EN=0 et EN=1
  - EN=0 c'est quand ça garde sa valeur précédente
  - L'entrée est donc BLOQUÉE
  - EN=1 c'est quand la sortie est égale à l'entrée
  - Dans ce cas-ci, l'entrée PASSE
- Quand ça passe, on copie exactement l'entrée vers la sortie
- Quand c'est bloque, la valeur juste avant est conservée et la sortie ne change pas

Technique d'analyse

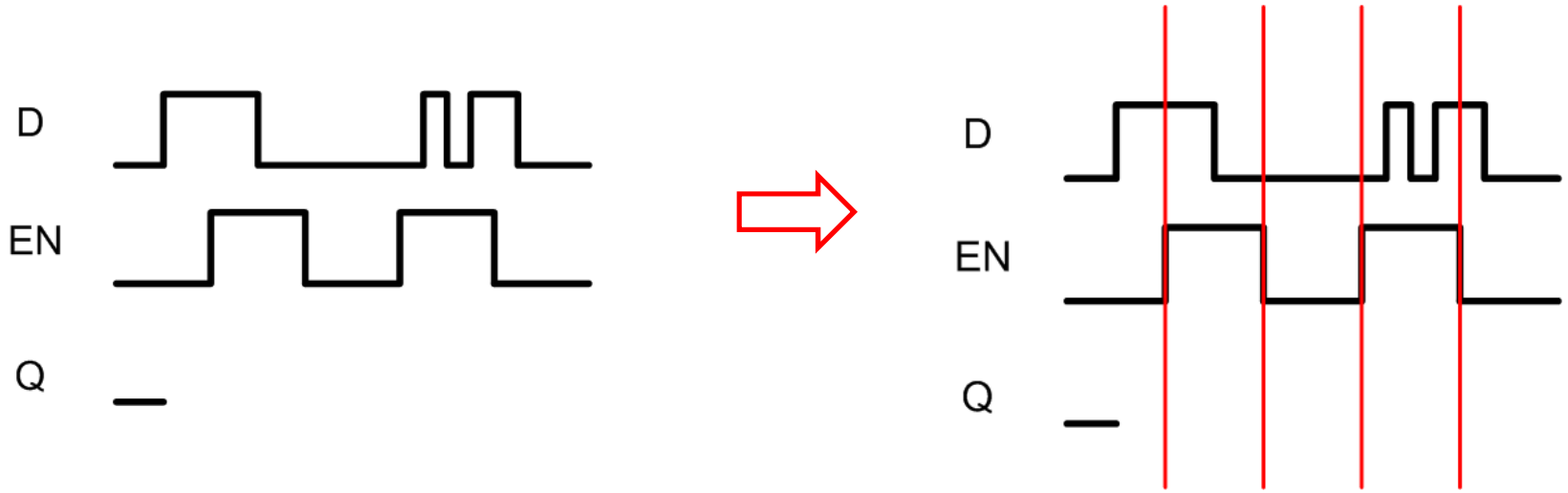
- On prend un exemple pour illustrer...
- On trace les lignes pour délimiter EN=0 et EN=1
  - La valeur ne change pas quand EN=0
  - La valeur D passe directement à la sortie quand EN=1





Exemple (seul)

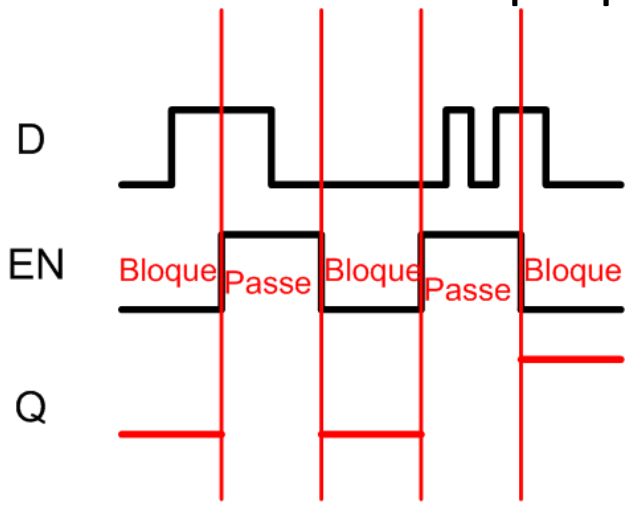
- Déterminez la valeur de Q
  - On voit que la valeur initiale de Q est 0...



Suggestion

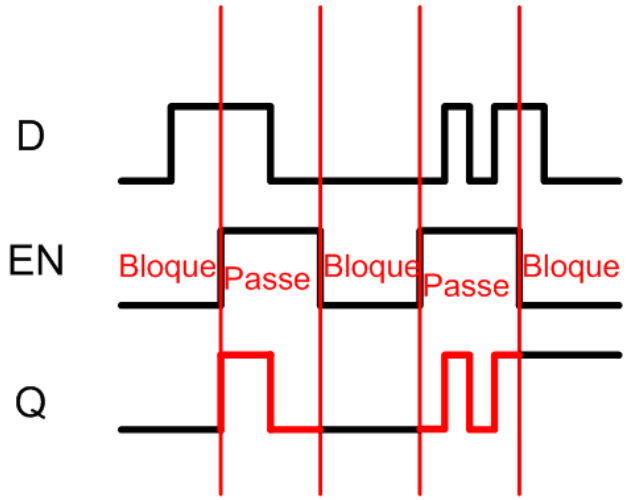
Exemple (seul)

- En séparant le diagramme temporel et tranches, on distingue les zones d'opération
  - Quand EN=0, on conserva le valeur précédente
- Quand EN tombe a 0, on regarde D:
  - Si D=0, Q=0 durant tout le temps que EN=0
  - Si D=1, Q=1 durant tout le temps que EN=0



Exemple (seul)

- Quand EN monte à 1, la sortie est déterminée par D
  - Quand EN=1, la sortie est égale à l'entrée
  - Quand D=1, Q devient 1
  - Quand D=0, Q devient 0
  - Le latch devient "transparent" quand EN=1...



Flip flop D

- Les bascules D sont des éléments qui sont sensibles aux niveaux:
  - Si EN etait a un NIVEAU haut, c'est transparent
  - Si EN etait a un NIVEAU bas, c'est bloqué
- Il existe aussi d'autres éléments de mémoire
  - Ceux-ci sont sensibles aux TRANSITIONS (fronts)
  - Ce sont les **FLIP FLOPs**

Flip flop D

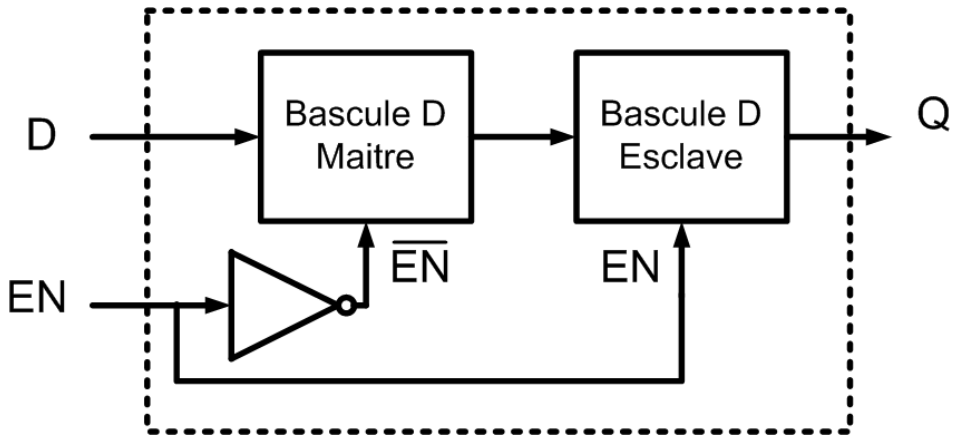
- Les flip flops peuvent être sensibles aux front montants ou descendants:

- On lit la donnée au front montant ou au front descendant
  - On ne peut JAMAIS être sensible aux 2 fronts…
  - Certains chercheurs le font, mais ce n'est pas pour nous

Comment fait-on pour être sensible aux transitions?
Il existe plusieurs structures pour l'être…

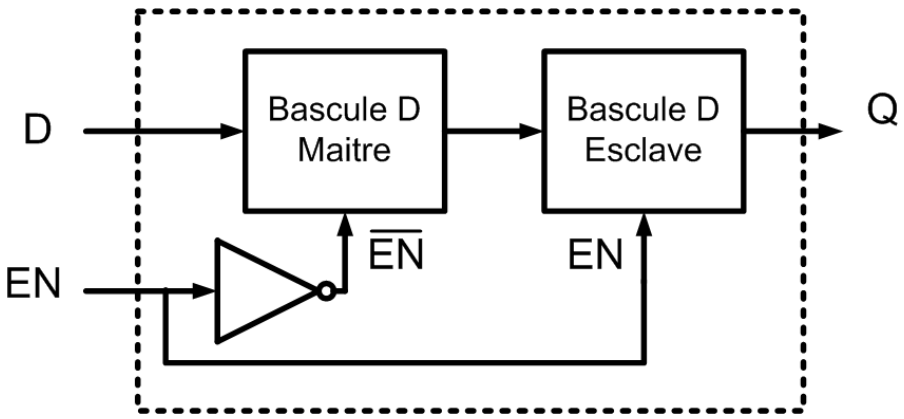
Flip flop D

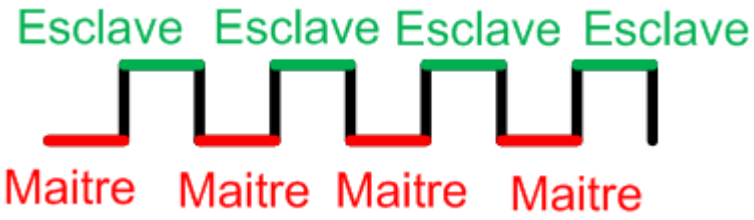
- La structure classique s'appelle la structure maître-esclave
  - Il y a une bascule D maître
  - Il y a une bascule D esclave



Flip flop D

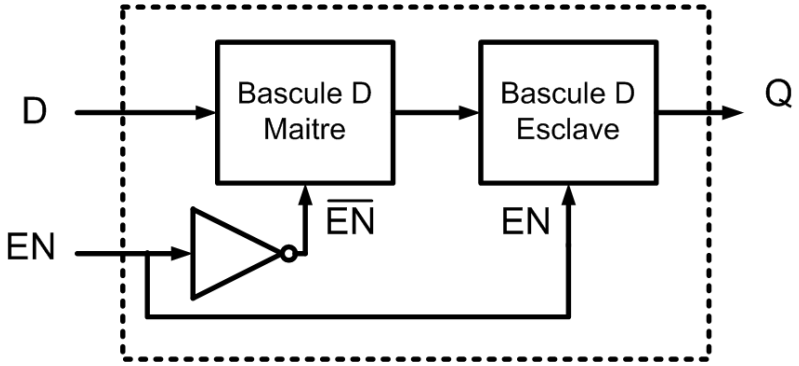
- Quand une bascule fonctionne, l'autre ne fonctionne pas
  - Quand le maître est transparent, l'esclave est bloqué
  - Quand l'esclave est transparent, le maître est bloqué





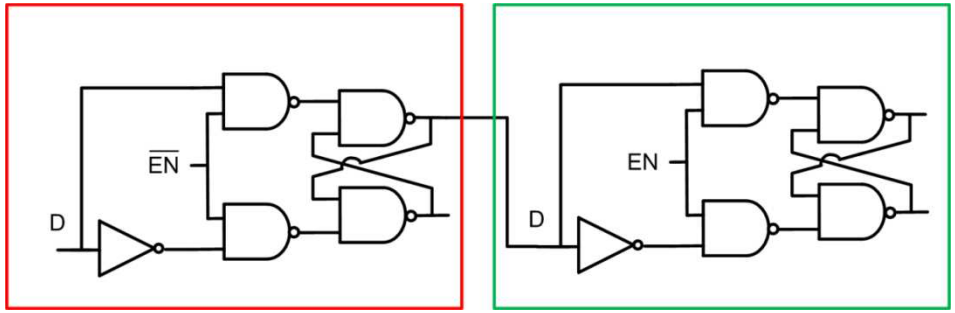
Flip flop D

- Quand EN=0, la sortie ne change pas
  - L'esclave est bloqué
  - L'entrée modifie le maître avec une donnée *D*
- Quand EN=1, la sortie devient immédiatement *D* et ne peut plus changer
  - Quand EN=1, le maître est bloqué
  - La sortie ne change donc QUE durant la transition 0 à 1



Flip flop D

- Le circuit d'une flip flop en détail ressemble à ceci:
  - C'est réellement juste 2 bascules D l'un après l'autre avec un EN qui est inversé...

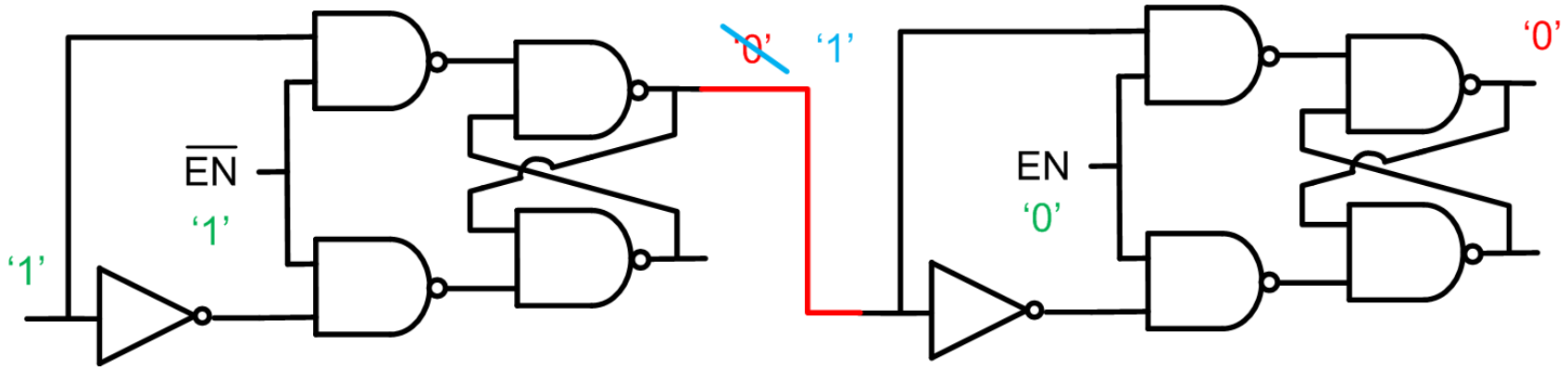


  - Voyons comment ça se comporte...

NOTE: La condition initiale est que les 2 bascules sont à '0'

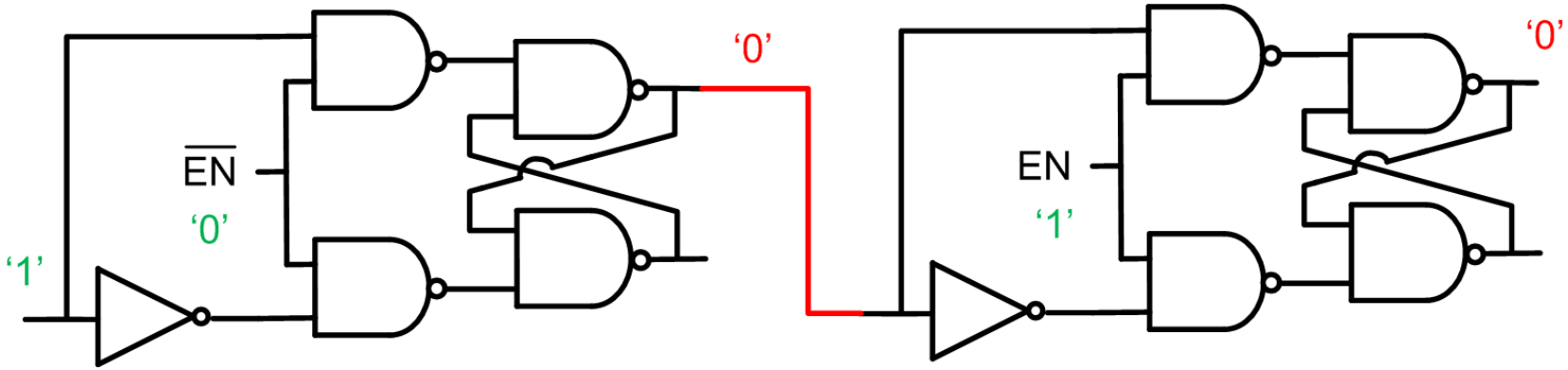
Flip flop D

- Cas #1: Quand D=1 et EN=0:
  - La premiere bascule est transparente
  - Sa sortie devient 1
  - La deuxieme bascule est bloquee (rien ne passe)
  - Sa sortie reste donc a 0



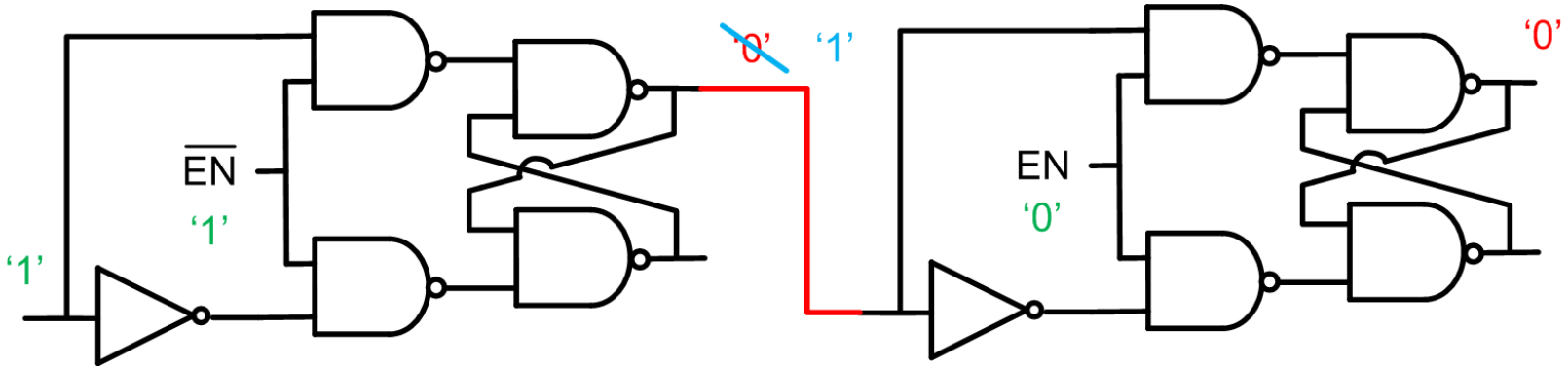
Flip flop D

- On remet tout a ‘0’ et on recommence...
- Cas #2: Quand D=1 et EN=1:
  - La premiere bascule est bloquee (rien ne passe)
  - Sa sortie reste a 0
  - La deuxieme bascule est transparente
  - L’entree est 0 et donc, la sortie est 0



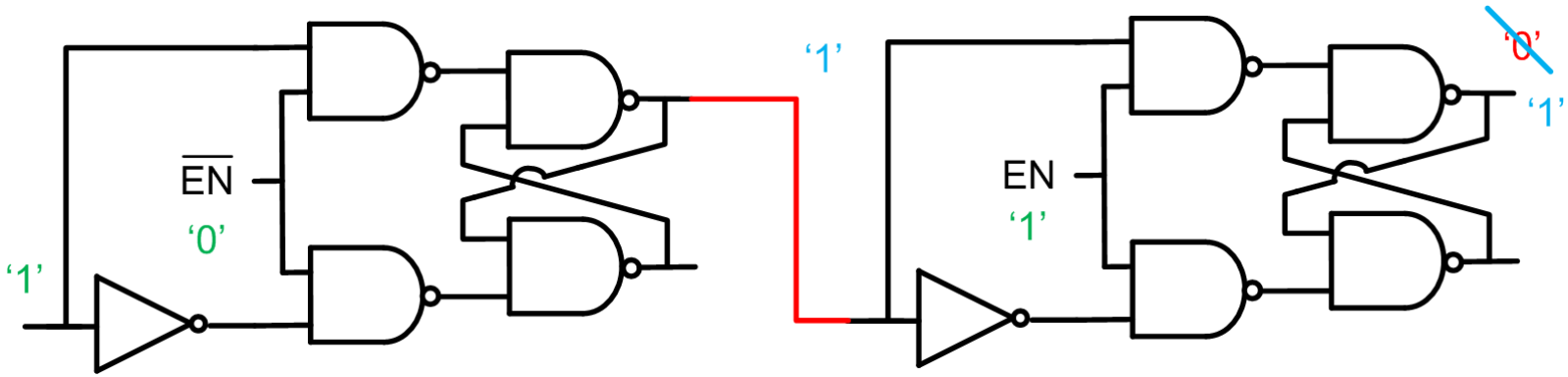
Flip flop D

- On remet tout a ‘0’ et on recommence...
- Cas #3: Quand D=1 et EN passe de 0 a 1
- Au debut, EN est a ‘0’
  - La sortie de la premiere bascule est 1
  - Mais la sortie ne change pas parce que la deuxieme bascule a son EN=0



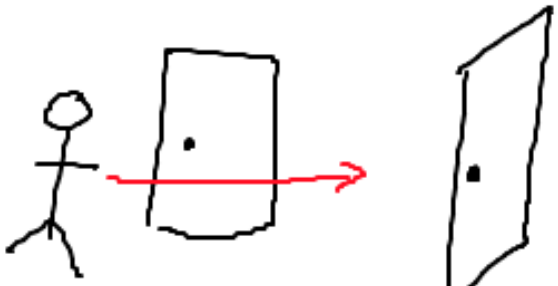
Flip flop D

- Ensuite, EN devient ‘1’
  - La premiere bascule est bloquee et sa sortie reste a 1
  - La deuxieme bascule est maintenant transparente
  - Et la sortie change a 1



Flip flop D: Resume

- Quand EN=0, la donnee passe par le maitre
- Quand EN=1, la sortie du maitre passe par l'esclave
  - Une fois EN=1, la sortie ne change plus...



Maitre laisse passer

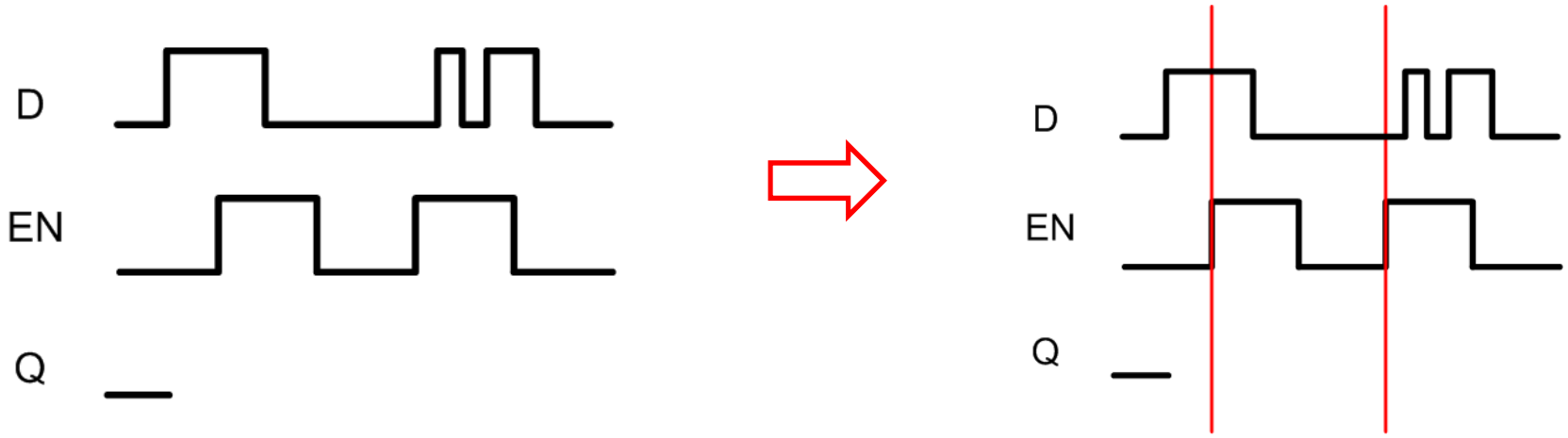
Esclave laisse passer

Technique d’analyse

- Identifier les fronts actifs (montants ou descendants) en tracant une ligne
    - Cette information est normalement connue
- A chaque ligne, la sortie peut changer
    - La valeur en entree devient la valeur a la sortie a cet instant
    - Le reste du temps, la sortie ne peut pas changer

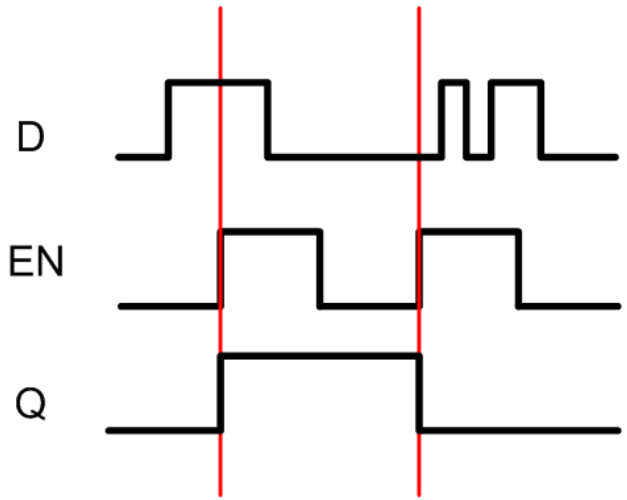
Exemple (seul)

- Determinez la valeur de Q
  - On voit que la valeur initiale de Q est 0...



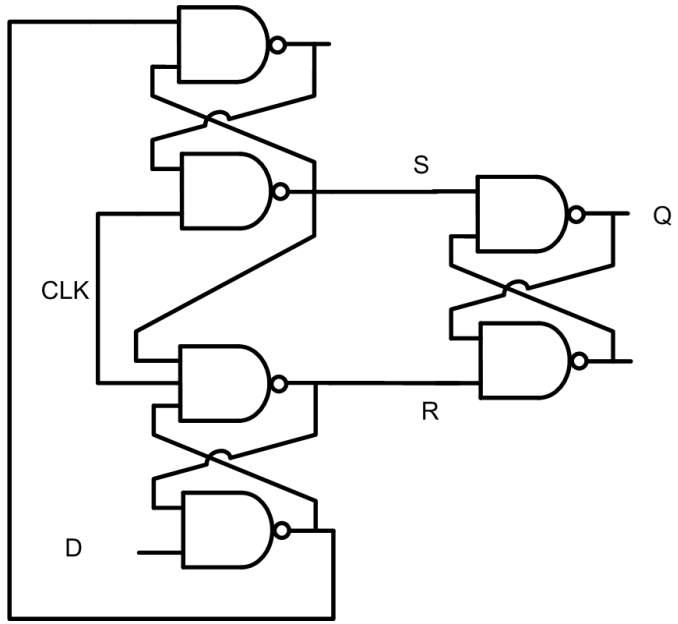
Exemple (seul)

- A chaque transition montante, on regarde la valeur de D:
  - Si c'est 1, la sortie sera 1 jusqu'a la prochaine transition
  - Si c'est 0, la sortie sera 0 jusqu'a la prochaine transition



Implementation #2

- La structure maitre-esclave est la classique
- Il en existe d'autres... Comme celle-ci
  - Son fonctionnement demande du temps de reflexion

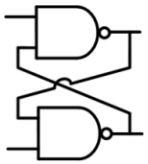


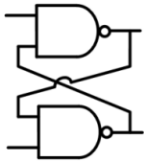
Premierement, decomposons sa structure...

Implementation #2

- On commence avec 3 bascule SR

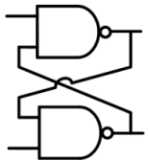
Cette partie controle le S



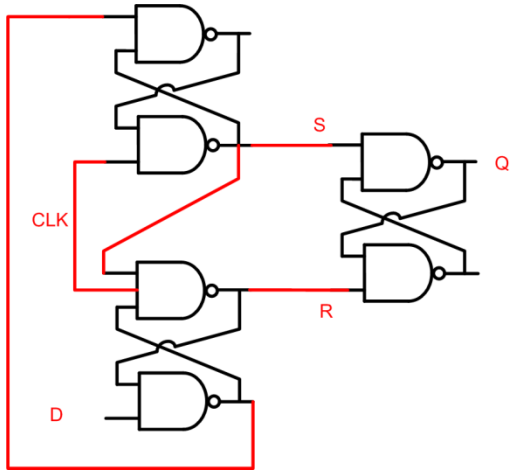


SR de sortie

Cette partie controle le R



- Et on les connecte ensemble



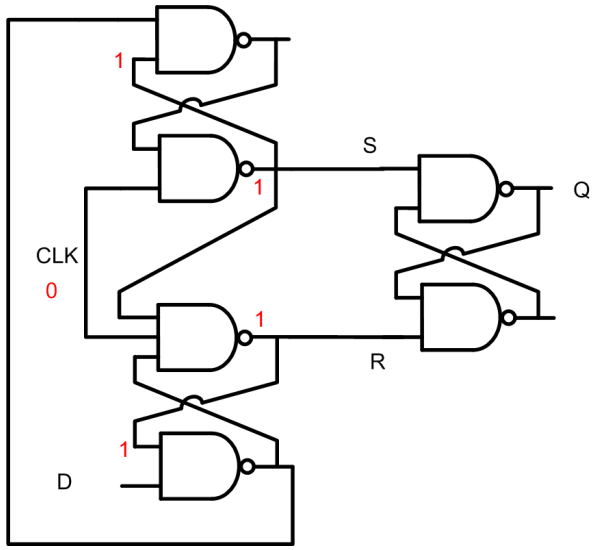
Comment est-ce que ca fonctionne?

On veut entrer 0

- Ex: on veut entrer 0 dans la flip flop
- Il faut faire 2 choses
  - Mettre D=0
  - Attendre la transition de CLK de 0 a 1
- L'analyse commence avec CLK=0

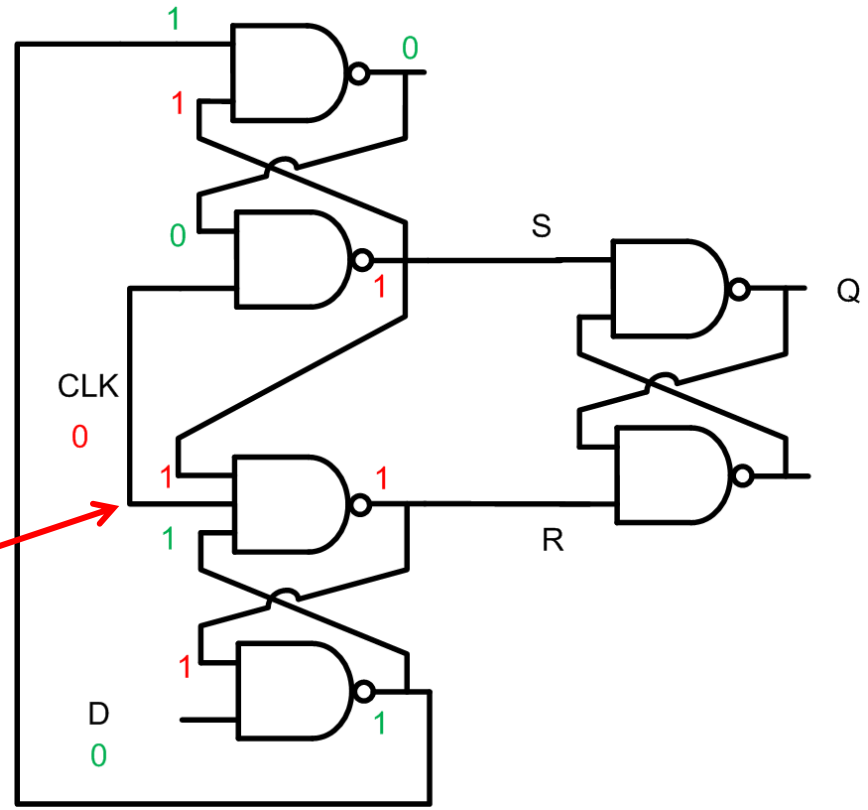
1. On commence par analyser le 0 de l'horloge
2. Ca genere des 1 aux valeurs de S et R
3. Les valeurs de sortie resteront les memes tant que CLK=0

Parlons maintenant de la donnee D...



On veut entrer 0

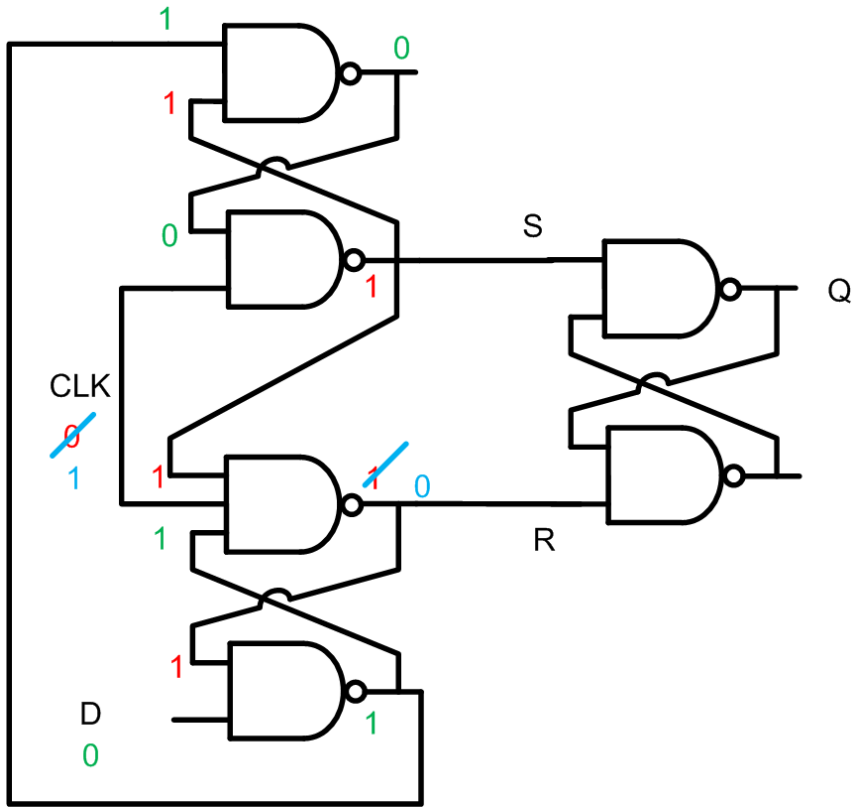
- Avec 0 a l'entree D, ca ne change pas la sortie tant que l'horloge reste a 0



Quand CLK monte a 1
R deviendra 0...

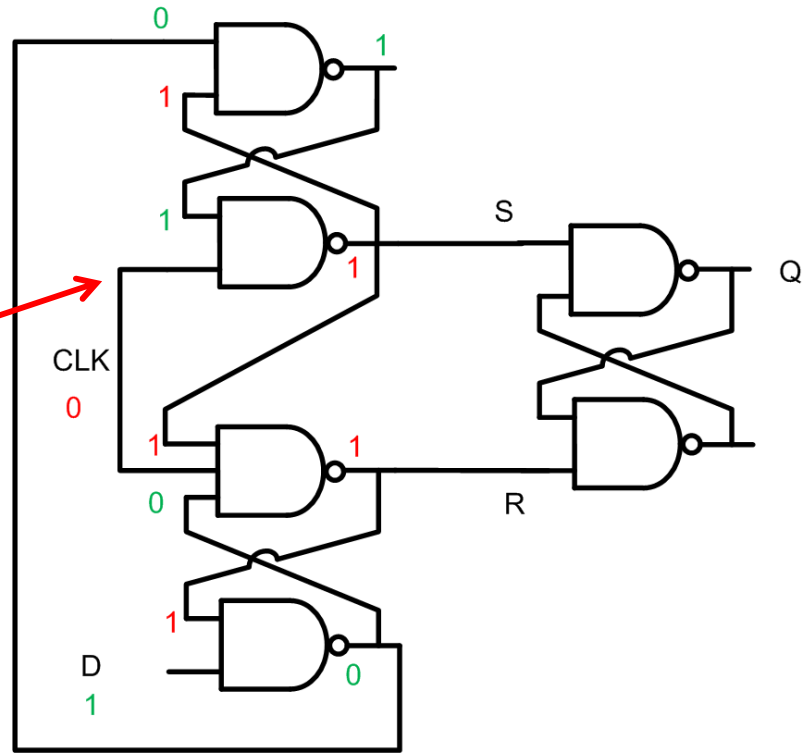
On veut entrer 0

- Quand l'horloge passe a 1, le R devient 0:
  - La sortie sera donc 0



On veut entrer 1

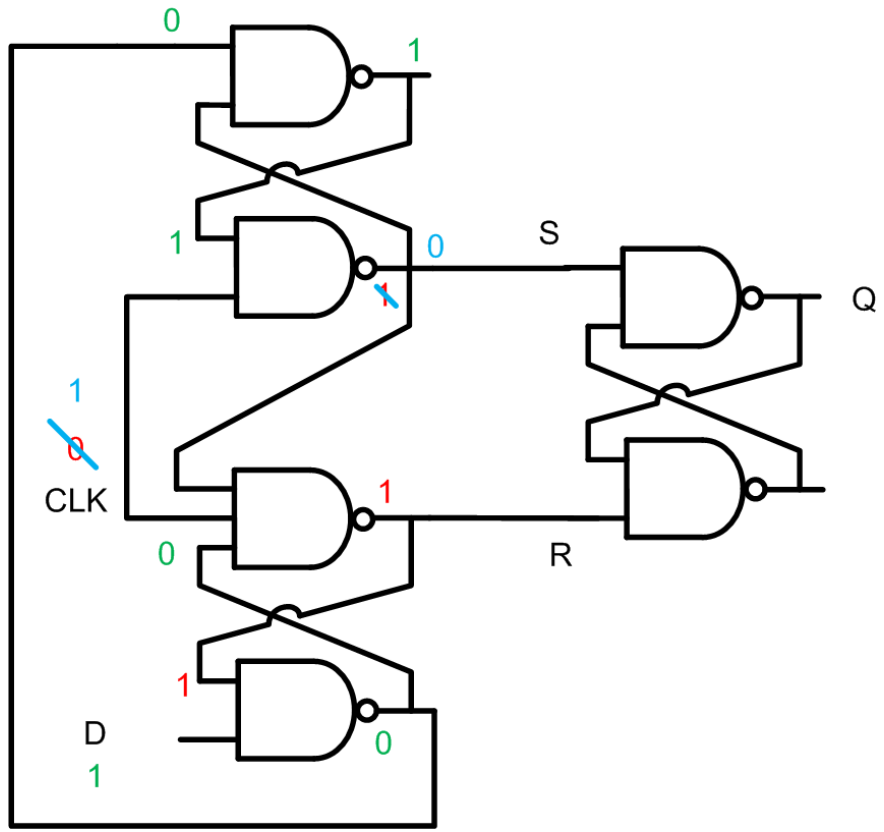
- Si on voulait entrer 1, on recommence avec CLK=0...
  - Avec S et R egal a 1, la sortie ne change pas



Quand CLK monte a 1
S deviendra 0...

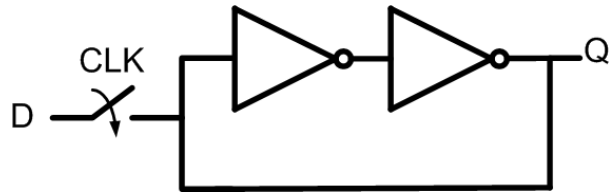
On veut entrer 1

- Quand l'horloge passe de 0 a 1 on aurait S=0
  - La sortie prendra la valeur de 1



Implementation #3

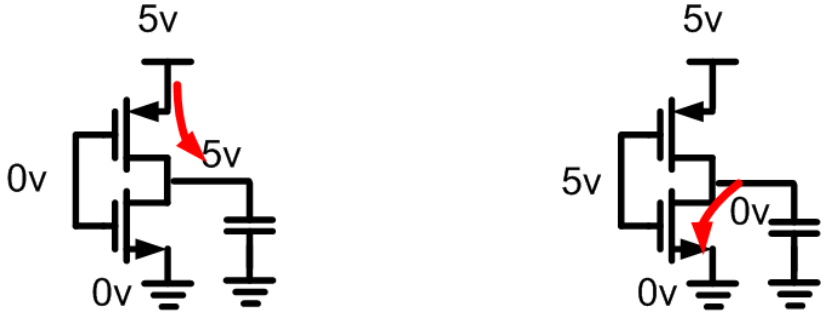
- La derniere implementation est celle qui est la plus utilisee en industrie:
  - Elle est basee sur la structure avec 2 bascules D maitre-esclave
  - Chaque bascule ressemble a ceci:



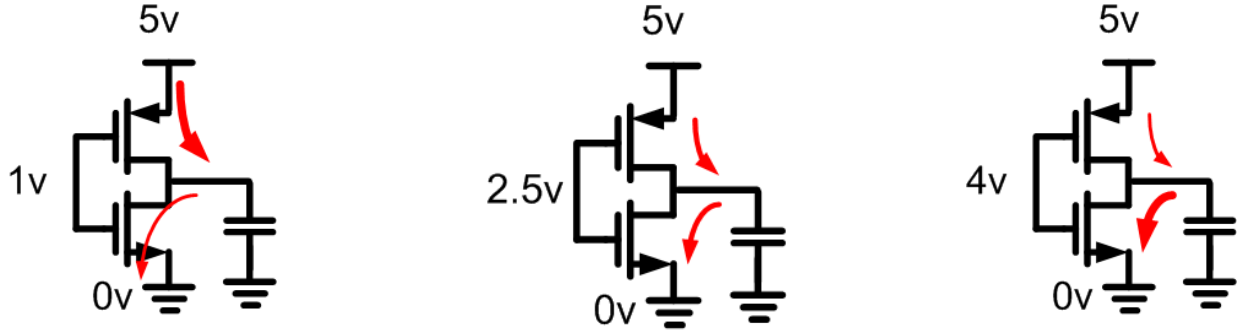
- Ca utilise une retroaction positive

Implementation #3

- On sait qu'un inverseur ressemble a ceci:

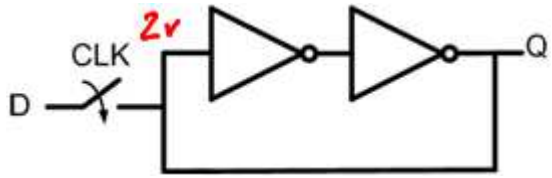


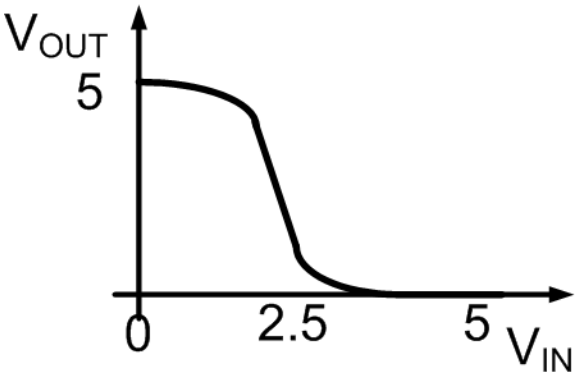
- Si les entrees n'etaient pas 0v ou 5v...
  - Plus l'entree est petite, plus la sortie ressemble a 5v
  - Plus l'entree est grande, plus la sortie ressemble a 0v



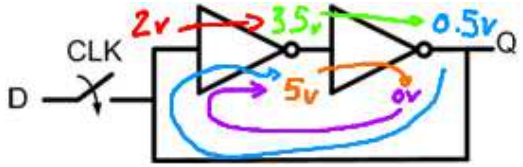
Implementation #3

- Imaginons que la valeur apres le commutateur etait 2v quand ca arrete de conduire:



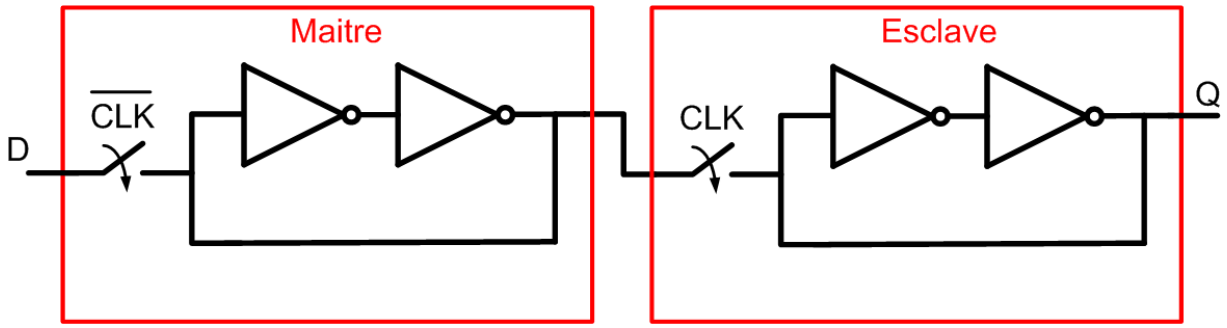


- Par exemple:
  - Le 2v donnerait 3.5v a l'entree de l'autre inverseur
  - L'autre inverseur donnerait 0.5v a l'entree du premier
  - A sa sortie, j'aurais presque 5v
  - A sa sortie, j'aurais 0... Qui donnera 5v...



Implementation #3

- On connecte les bascules en maitre-esclave
  - On doit avoir un etage avec CLK' et l'autre avec CLK
  - Ca nous fera un structure qui est sensible aux transition d'horloge ("front d'horloge")



Temps de setup et de hold

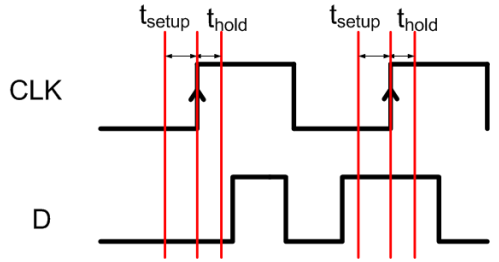
- On peut voir le fonctionnement d'une flip flop de la facon suivante:
  - Quand un front montant d'horloge arrive, je regarde la valeur a l'entree D
  - Cette valeur se retrouvera alors a la sortie Q
  - Sinon, rien ne change
- On pourrait se poser la question suivante:
  - Que se passerait-il si D changeait en MEME temps que le front montant?

Temps de setup et de hold

- Si D changeait en meme temps que CLK, ca risque de mal fonctionner:
  - Soit que ca prend la bonne valeur (valeur voulue)
  - Soit que ca prend la mauvause valeur
  - Soit que ca donne une valeur indeterminee
- Pour s'assurer que ca fonctionne, il faut respecter 2 choses:
  - Le temps de "setup" (preparation)
  - Le temps de "hold" (maintient)

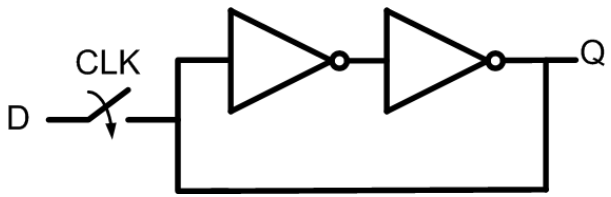
Temps de setup et de hold

- Temps de setup (t_{setup})
  - Temps AVANT le front d'horloge durant lequel D n'a pas le droit de changer
  - D doit avoir une valeur stable t_{setup} avant le front d'horloge
- Temps de hold:
  - Temps APRES le front d'horloge durant lequel D n'a pas le droit de changer
  - D doit garder la valeur t_{hold} apres le front d'horloge



Temps de setup et de hold

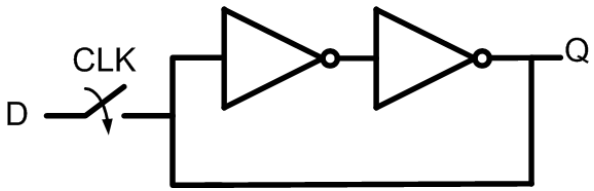
- Si D changeait trop proche de la transition active d'horloge, il y aurait probleme
  - La reponse peut soit etre 0, soit etre 1 ou pire encore, etre indetermine
  - Cette reponse indeterminee s'appelle "metastabilite"
- Pour comprendre ce que ca veut dire, retournons voir l'implementation #3...
  - Ca utilise 2 inverseurs connectes comme ceci:

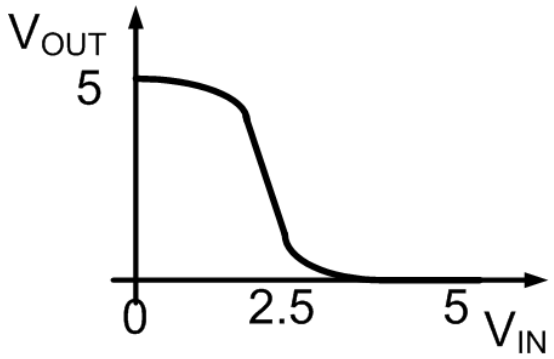


Ca c'est juste une bascule D

Temps de setup et de hold

- On sait qu'un inverseur donne 1 quand l'entree est 0 et donne 0 quand l'entree est 1
  - Que se passe-t-il quand l'entree est 2.5v?
  - Reponse: la sortie sera 2.5v
  - Ca va rester comme ca jusqu'a ce qu'il y ait du bruit
  - Ce 2.5v correspond a la metastabilite





2.5v ne veut pas dire 0 ni 1

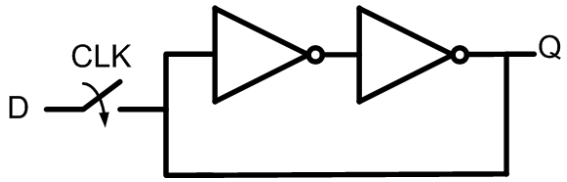
Et ca risque de rester 2.5v pendant un certain temps...

Temps de setup et de hold

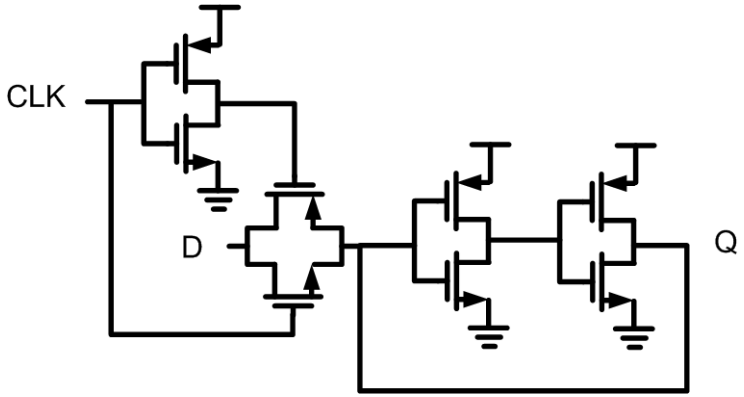
- Alors la metastabilite n'est pas bonne...
- C'est problematique parce que differents circuits intepretent ca de differentes manieres
  - 2.5v pourrait etre vu comme '0' pour certains circuits et vu comme '1' pour d'autres
  - On risque d'avoir des comportements bizarres si ca se propageait
- On doit donc respecter les temps de setup et de hold

Temps de setup et de hold

- D'ou viennent ces temps de setup et hold?
- Pour le savoir, il faut examiner la bascule D...

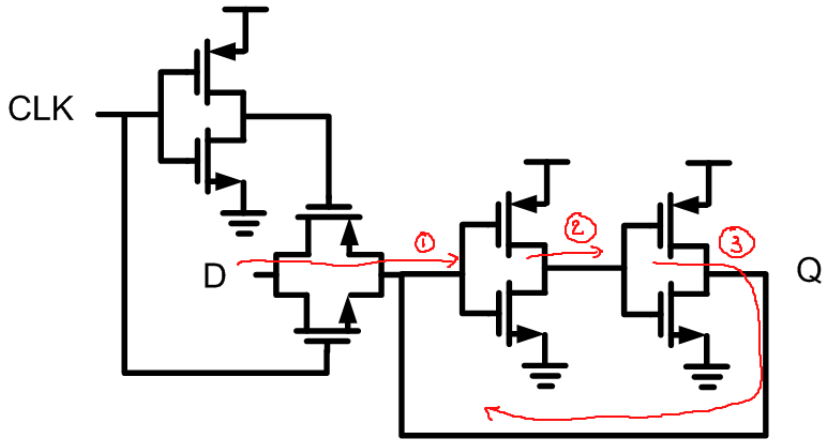


- Mais il faut aller au niveau transistor...



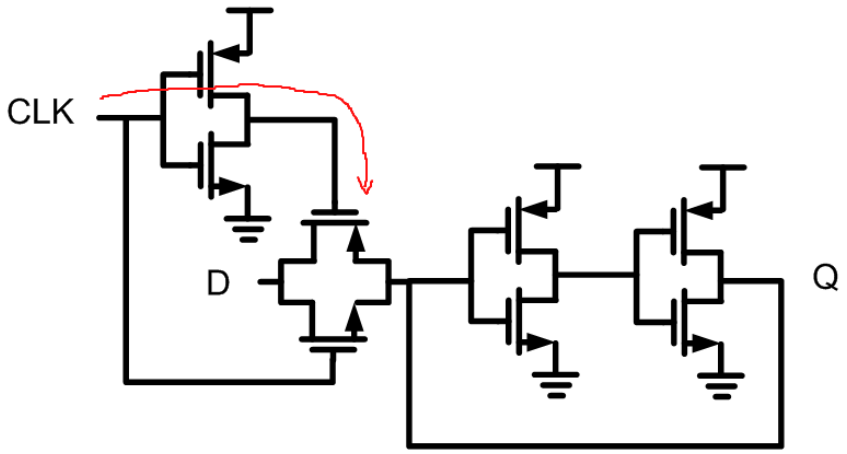
Temps de setup

- Quand je bloque le commutateur, D doit:
  - Avoir passe par la porte de transmission
  - Avoir change la tension a l'entree du premier inverseur
  - Avoir change la tension a l'entree du deuxieme inverseur
- D doit donc arriver un certain temps avant CLK



Temps de hold

- Quand je decide de mettre CLK a 0, ce 0 doit passer par l'inverseur pour desactiver le PMOS
  - Durant ce temps, D ne devrait pas changer parce que ca risque de deranger les tensions de l'autre cote
- D doit donc rester stable un certain temps avant CLK



Resume

- Il existe 2 elements memoires:
  - Les bascules et les flip flops
- La grosse difference:
  - Bascules: sensibles aux niveaux
  - Flip-flops: sensibles aux fronts
- Les bascules differentes:
  - SR: S met la sortie a 1, R met la sortie a 0
  - Il y a parfois un enable pour permettre le changement
  - D: Il y a 1 seule entree. Sortie=entree quand enable=1

Resume

- Flip flop D: Semblable a la bascule D mais est sensible au front
  - La sortie change de valeur SEULEMENT au front actif de l'horloge
  - Par exemple, quand ca change de 0 a 1 (ou vice versa)
- La bascule D permet le changement de valeur n'importe quand si le enable est a 1