13. Réalisation de la bascule DFF

Principes de fonctionnement des ordinateurs

Jonas Lätt

Centre Universitaire d'Informatique

Contenu du cours

Partie I: Introduction

Partie II: Codage de l'information

Partie III: Circuits logiques

Partie IV: Architecture des ordinateurs

1. Introduction
2. Histoire de l'informatique
3. Information digitale et codage de l'information
4. Codage des nombres entiers naturels
5. Codage des nombres entiers relatifs
6. Codage des nombres réels
7. Codage de contenu média
8. Portes logiques
9. Circuits logiques combinatoires et algèbre de Boole
10. Réalisation d'un circuit combinatoire
11. Circuits combinatoires importants
12. Principes de logique séquentielle
13. **Réalisation de la bascule DFF**
14. Architecture de von Neumann
15. Réalisation des composants
16. Code machine et langage assembleur
17. Architecture d'un processeur
18. Performance et micro-architecture
19. Du processeur au système

13. Réalisation de la bascule DFF

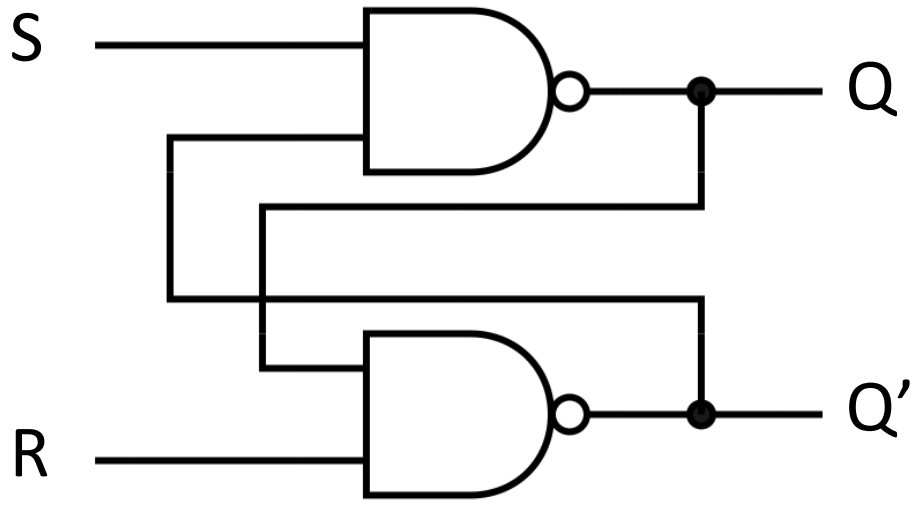
Le verrou S-R

Un circuit croisé

Jonas Lätt

Diagramme du verrou S-R

- Ce circuit qui mémorise un bit s'appelle un «verrou» (anglais: «latch») peut «verrouiller» la valeur à sa sortie.
- Réalisation possible: un circuit croisé avec deux portes NAND.



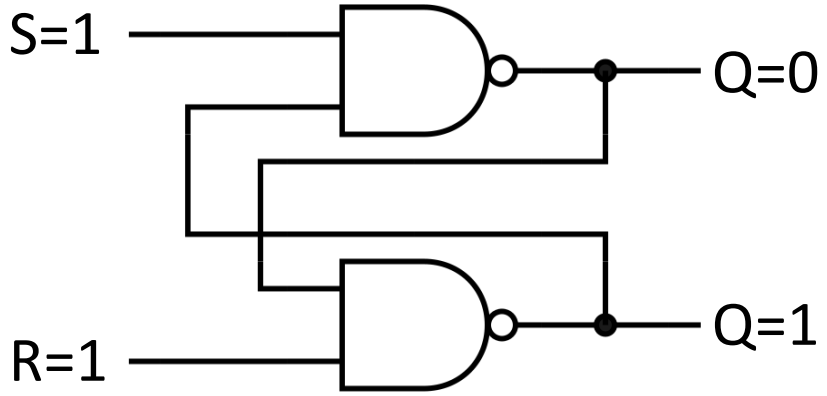
Q: l'état mémorisé dans le circuit.

Dans ce circuit, par construction,
Q' = !Q

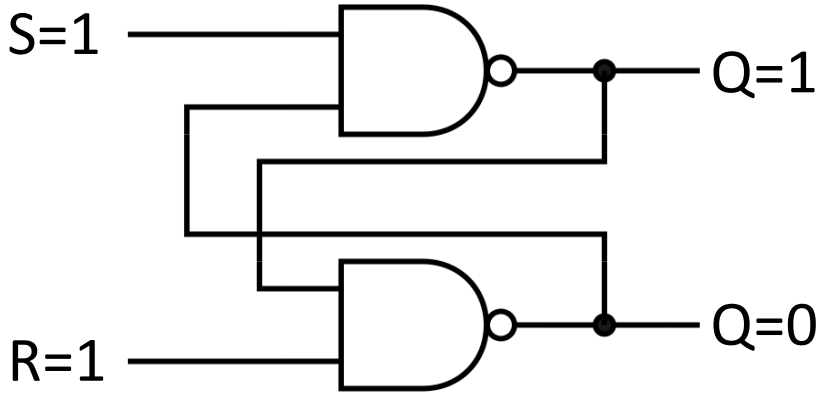
Verrou S-R: Mode *Verrouillé*

Le verrou est en mode *verrouillé* lorsque S=R=1. La sortie Q peut valoir 0 ou 1, et cette valeur est stable (elle est mémorisée).

Etat du circuit qui mémorise 0:



Etat du circuit qui mémorise 1:

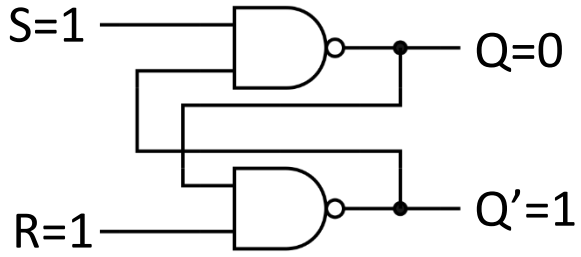


La sortie ne dépend pas que des entrées: il ne s'agit **pas** d'un circuit logique combinatoire.

Verrou S-R: Mode *Ouvert*

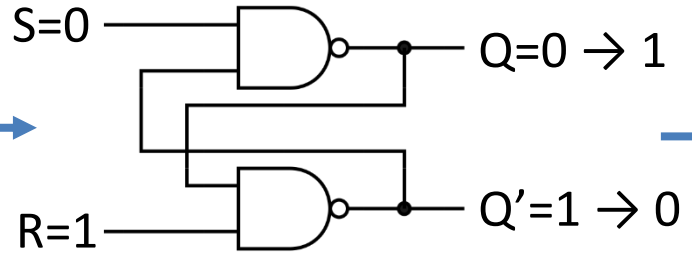
- La bascule est en mode *ouvert* lorsque S = !R. En ce mode elle fonctionne comme un circuit combinatoire. Elle bascule vers un nouvel état.
- Si S=0 (“Set”), le nouvel état est 1. Si R=0 (“Reset”), le nouvel état est 0.

Verrouillé



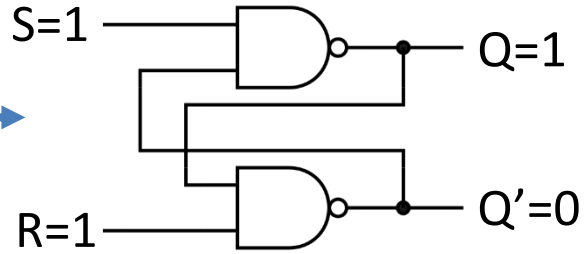
Dans cet exemple, on commence par mémoriser 0.

Ouvert



Pour changer l’état, il suffit de mettre momentanément S à 0.

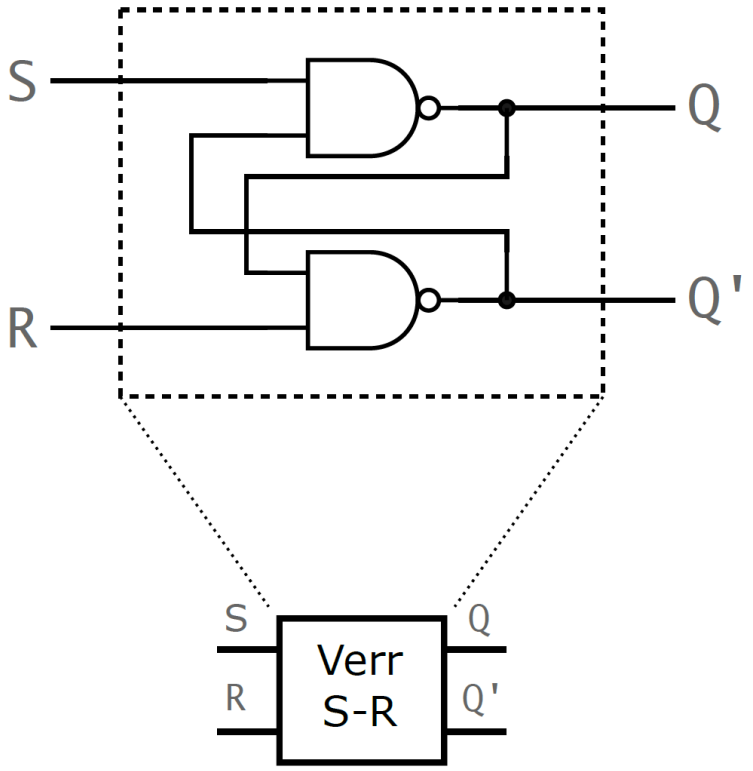
Verrouillé



Le nouvel état Q=1 persiste, même quand S revient à 1.

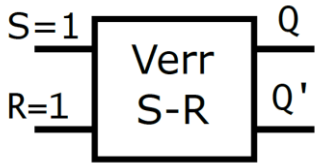
Verrou S-R: Résumé

Symbole de diagramme

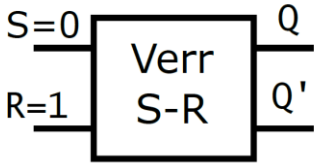


Modes de fonctionnement

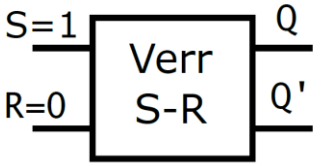
1) Verrouillé



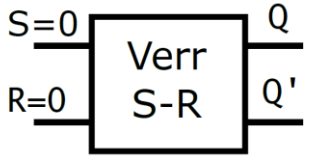
2) Ouvert



ou



3) Etat indéfini: *à éviter!*

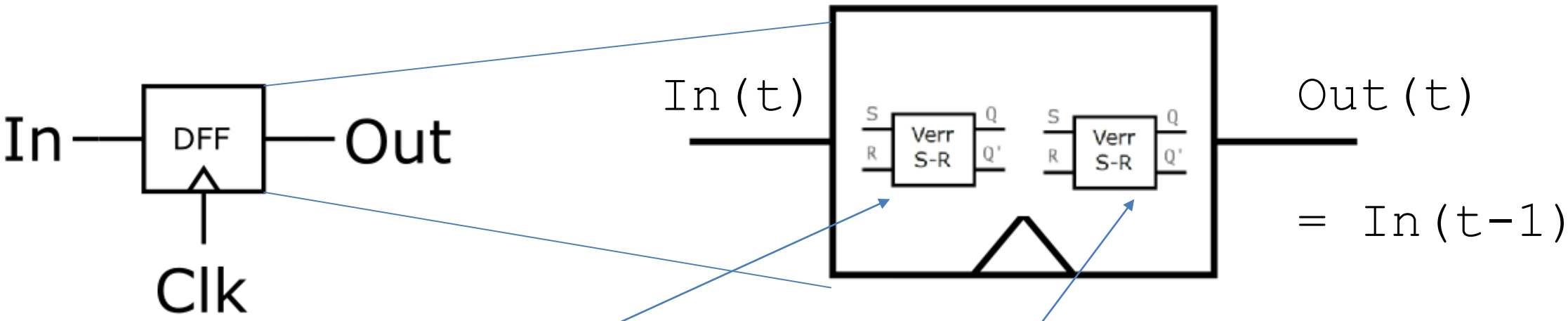


16. Réalisation de la bascule DFF

Construction du DFF à l’aide de deux verrous S-R

Jonas Lätt

Idée: Utilisation de deux verrous S-R



Verrou 1

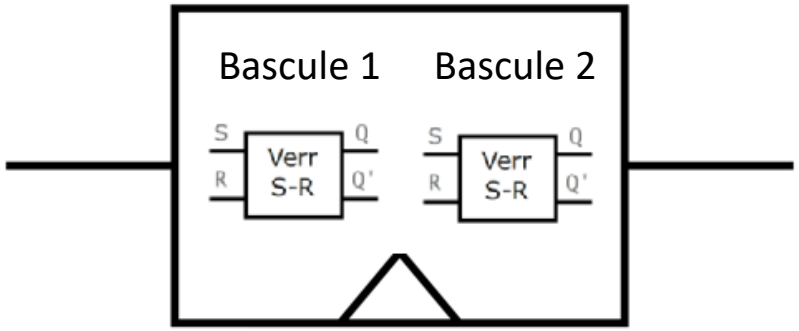
Ce verrou sert à accueillir la valeur en entrée, `In(t)`.

Verrou 2

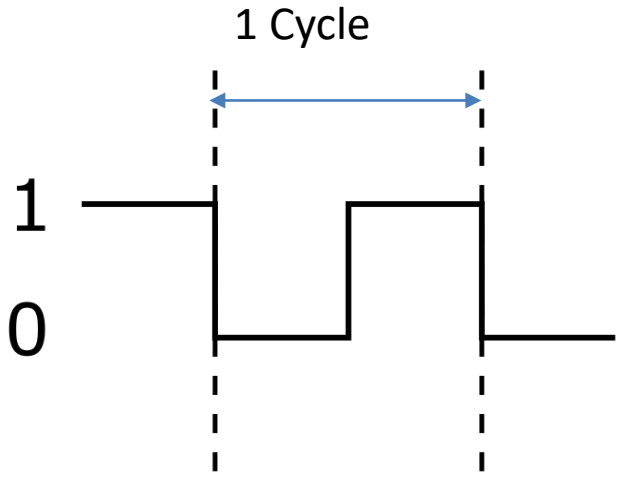
Ce verrou sert à maintenir la sortie à l'ancienne valeur, `In(t-1)`.

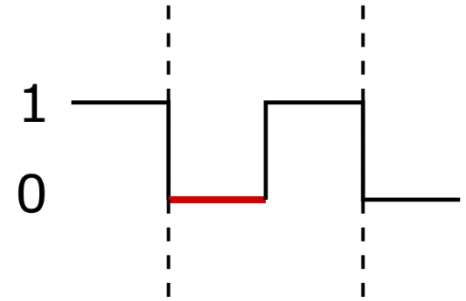
Division du cycle d'horloge en deux parties

DFF



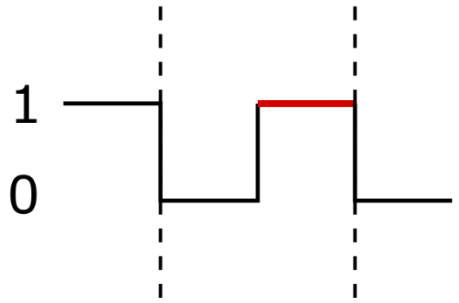
Rappel: le signal horloge (Clk)





Première partie du cycle (signal Clk faible):

- Le DFF n'accepte pas encore de nouvelle valeur en entrée.
- En interne, le DFF copie la donnée `In(t-1)` du verrou 1 vers le verrou 2.



Deuxième partie du cycle (signal Clk élevé):

- Le verrou 1 accueille la nouvelle valeur en entrée `In(t)`.
- Le verrou 2 maintient son état, la valeur `In(t-1)`.

Utilisation du Verrou 1

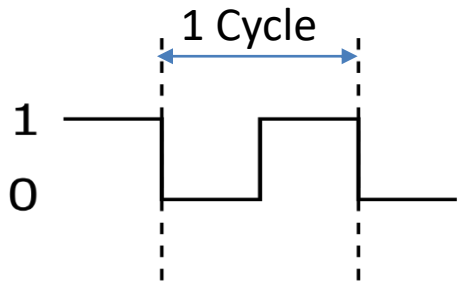
```
si Clk vaut 1
    si In(t)vaut 1
        S = 0
        R = 1
    autrement
        S = 1
        R = 0
autrement  // si !Clk
    S = 1
    R = 1
```

Mode *basculer*

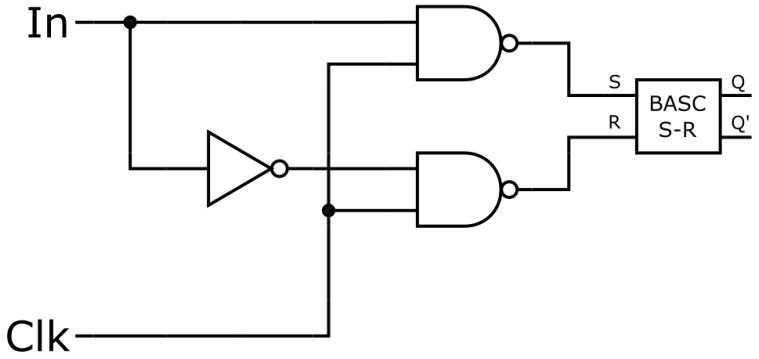
Opération *Set*

Opération *Reset*

Mode *mémoriser*



Clk	In(t)	S	R
0	0	1	1
0	1	1	1
1	0	1	0
1	1	0	1



Jonas Lätt

Circuit interne du Flip-Flop DFF

Le circuit pour le verrou 2 est identique à celui du verrou 1, sauf que

- Le signal en entrée `In(t)` est remplacé par la sortie `Q` du verrou 1.
- Le signal d'horloge `Clk` est inversé: le verrou 2 bascule lors d'un signal d'horloge faible.