

# Logique séquentielle : verrous et bascules

Philippe Notez (philippe.notez@inmc.fr)



## Sommaire

- [Logique séquentielle](#)
- [Compteur binaire / diviseur de fréquence par 2](#)

## Introduction

L'électronique et l'informatique ont profondément modifié notre société. C'est certainement la révolution industrielle la plus rapide de l'histoire de l'humanité. Aujourd'hui, les systèmes embarqués sont omniprésents dans notre vie quotidienne et nous emmènent vers un monde de plus en plus connecté, avec ses avantages et ses inconvénients...

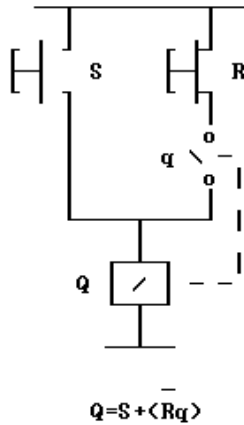
L'auteur ne pourra en aucun cas être tenu responsable des dommages qui résulteraient de l'utilisation des informations publiées sur ce site, sous [licence Creative Commons BY-NC-SA](#). Toute reproduction ou modification d'un document, même partielle, est autorisée à condition que son origine et le nom de l'auteur soient clairement indiqués (BY), qu'il soit utilisé à des fins non commerciales (NC), que son mode de diffusion soit identique au document initial (SA), et que cela ne porte pas atteinte à l'auteur.

Ce document présente la logique séquentielle qui constitue le second pilier de l'électronique numérique (le premier pilier étant la [logique combinatoire](#)), en espérant toujours être le plus clair et précis possible. Malgré tout le soin apporté à la rédaction, l'auteur vous remercie de bien vouloir le contacter si vous relevez la moindre erreur ou omission, et vous souhaite une agréable lecture.

## Logique séquentielle

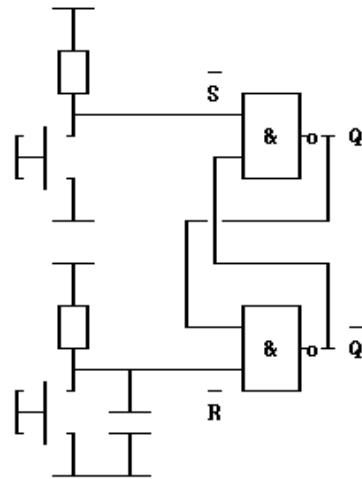
A la différence de la logique combinatoire où la sortie ne dépend que de l'état des entrées, la logique séquentielle introduit la notion de mémoire : la sortie dépend de l'état des entrées ET de son état précédent. Prenons l'exemple suivant :

**Système à marche prioritaire**



Quand on appuie simultanément sur les deux interrupteurs, la sortie est à 1 (marche prioritaire).

**Verrou RS à portes NAND**



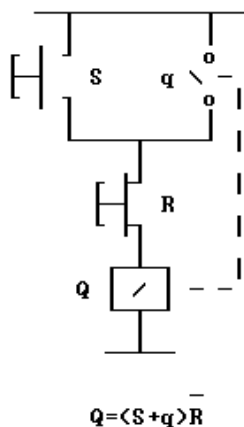
| S | R | Q | Q̄ |               |
|---|---|---|----|---------------|
| 0 | 0 | 1 | 1  | état interdit |
| 0 | 1 | 1 | 0  |               |
| 1 | 0 | 0 | 1  |               |
| 1 | 1 | n | n  | état mémoire  |

On note S barre et R barre car les entrées Set et Reset présentent un niveau logique haut (valeur 1) quand les interrupteurs sont au repos. Elles sont actives (elles modifient les sorties) quand on appuie sur les interrupteurs, c'est à dire au niveau logique bas (valeur 0).

Le condensateur positionné sur l'entrée R barre permet d'initialiser le verrou lors de la mise sous tension car, le condensateur étant déchargé, on a S barre = 1 et R barre = 0, donc Q = 0 et Q barre = 1.

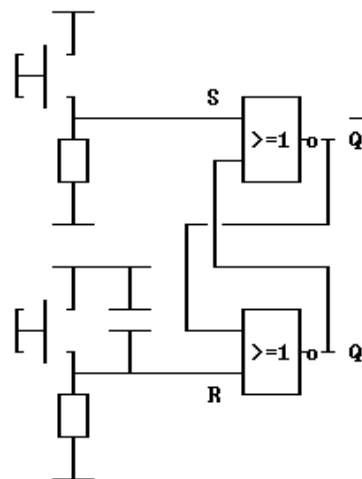
Lorsque les deux interrupteurs sont ouverts, le verrou mémorise l'état précédent (l'état juste avant le relâchement de l'un ou de l'autre). L'appui sur les deux interrupteurs simultanément est interdit car nous aurions dans ce cas Q = Q barre.

**Système à arrêt prioritaire**



Quand on appuie simultanément sur les deux interrupteurs, la sortie est à 0 (arrêt prioritaire).

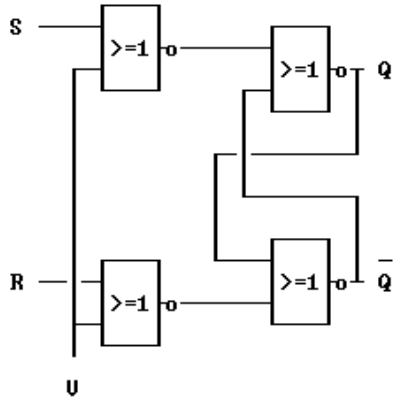
**Verrou RS à portes NOR**



| S | R | Q | Q̄ |               |
|---|---|---|----|---------------|
| 0 | 0 | n | n  | état mémoire  |
| 0 | 1 | 0 | 1  |               |
| 1 | 0 | 1 | 0  |               |
| 1 | 1 | 0 | 0  | état interdit |

Contrairement aux verrous, qui sont des bascules asynchrones, les bascules synchrones (ou simplement bascules) possèdent une entrée de validation.

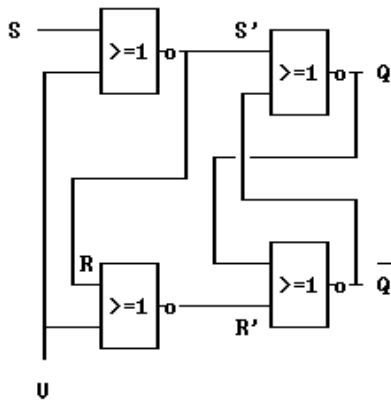
### Bascule RS (verrou RS avec entrée de validation)



| V | S | R | Q | $\bar{Q}$ |                 |
|---|---|---|---|-----------|-----------------|
| 0 | 0 | 0 | 0 | 0         | état interdit   |
| 0 | 0 | 1 | 0 | 1         |                 |
| 0 | 1 | 0 | 1 | 0         |                 |
| 0 | 1 | 1 | m | m         | état mémoire    |
| 1 | X | X | m | m         | état verrouillé |

X = 0 ou 1, peu importe

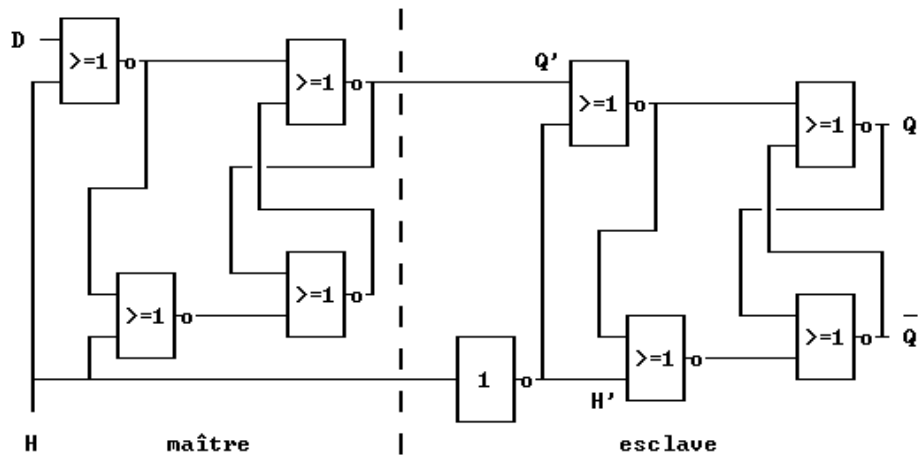
Quand  $V = 0$ , on retrouve le fonctionnement d'un verrou RS à portes NOR. Quand  $V = 1$ , les sorties des deux premières portes NOR passent à 0, on obtient la mémorisation de l'état précédent (l'état juste avant que  $V$  ne passe à 1). Les sorties sont verrouillées, elles ne changent pas d'état tant que  $V = 1$ . Pour éviter l'état interdit ( $S = R = 0$ ), on peut connecter l'entrée  $R$  à la sortie de la porte NOR correspondant à l'entrée  $S$ , comme indiqué dans le schéma suivant :



| V | S | R | S' | R' | Q | $\bar{Q}$ |                 |
|---|---|---|----|----|---|-----------|-----------------|
| 0 | 0 | 1 | 1  | 0  | 0 | 1         |                 |
| 0 | 1 | 0 | 0  | 1  | 1 | 0         |                 |
| 1 | X | 0 | 0  | 0  | m | m         | état verrouillé |

Il reste néanmoins un inconvénient : si l'entrée  $S$  change d'état plusieurs fois quand  $V = 0$ , la sortie  $Q$  fera de même puisqu'elle reflète l'état de l'entrée  $S$ .

Pour éviter ce problème, nous devons également mémoriser l'état de l'entrée  $S$  quand  $V = 0$ . En d'autres termes, nous devons mémoriser l'état de l'entrée  $S$  au moment du changement d'état de l'entrée  $V$ , c'est à dire sur un front et non plus sur un niveau. L'astuce consiste à mettre deux verrous en cascade, le premier verrou est appelé *maître*, le second verrou est appelé *esclave*. Si on renomme l'entrée  $S$  en  $D$  (*Donnée*) et l'entrée  $V$  en  $H$  (*Horloge*), on obtient une bascule  $D$  :



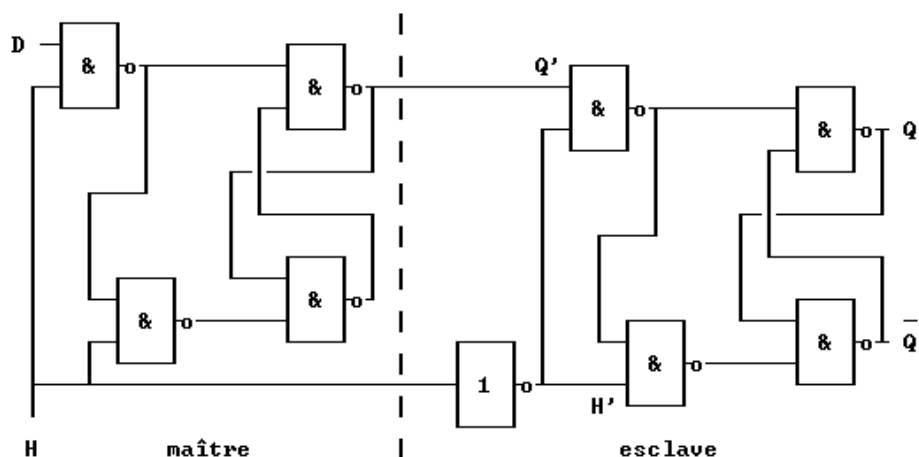
| D | H | Q' | H' | Q  |
|---|---|----|----|----|
| 0 | 1 | m  | 0  | Q' |
| 0 | 0 | 0  | 1  | m  |
| 1 | 1 | m  | 0  | Q' |
| 1 | 0 | 1  | 1  | m  |

Quand  $H = 1$ , le verrou maître est en mode mémorisation (verrou opaque) et le verrou esclave en mode copie (verrou transparent,  $Q = Q'$ ). Lors du front descendant de  $H$ , les modes des verrous s'inversent (verrou maître transparent et verrou esclave opaque). On obtient  $Q' = D$ . La sortie  $Q$  ne change pas, son état est égal à l'état de  $Q'$  juste avant le front descendant de  $H$  puisque le verrou esclave est passé en mode mémorisation. Lors du front montant de  $H$ , les modes des verrous s'inversent à nouveau. On obtient  $Q = Q'$ . Le verrou maître étant redevenu opaque, tout changement sur l'entrée  $D$  n'est pas recopié sur la sortie  $Q'$ . Le verrou esclave est en mode copie ( $Q = Q'$ ), mais comme  $Q'$  ne change pas (verrou maître opaque),  $Q$  ne change pas.

On constate que la sortie  $Q$  prend l'état de l'entrée  $D$  lors du front montant de l'entrée  $H$ , c'est donc une bascule sur front montant : toute modification de l'état de l'entrée  $D$  ne sera prise en compte que lors du prochain front montant sur l'entrée  $H$ .

L'entrée  $D$  est transférée dans le verrou maître sur le front descendant, le transfert du verrou maître vers le verrou esclave se fait sur le front montant. On dit que l'entrée est validée sur le front descendant, la sortie est validée sur le front montant.

Il est possible d'utiliser des portes NAND pour obtenir une bascule  $D$  sur front descendant :

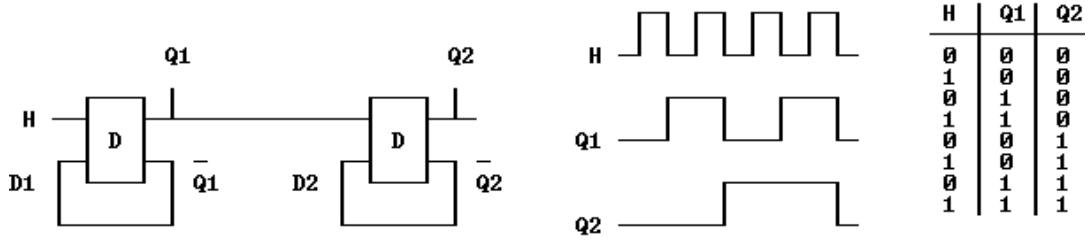


- quand  $H = 1$ , le verrou maître est transparent, le verrou esclave est opaque
- quand  $H = 0$ , le verrou maître est opaque, le verrou esclave est transparent

Nous avons négligé le temps de propagation des portes. Or, même si ce temps est très faible, il n'est pas nul. Cela pose un problème au niveau de la porte qui inverse l'état de l'horloge entre les deux verrous. En effet, il est possible que pendant un temps très bref, les horloges  $H$  et  $H'$  soient toutes les deux à 0 ou à 1. On aurait alors les deux verrous opaques ou transparents simultanément. Pour éviter ce dysfonctionnement, il faut supprimer cette porte inverseuse et utiliser, soit un verrou maître à portes NOR et un verrou esclave à portes

NAND (bascule sur front montant), soit un verrou maître à portes NAND et un verrou esclave à portes NOR (bascule sur front descendant).

**Compteur binaire / diviseur de fréquence par 2**



Dans cet exemple, on utilise un maître NAND et un esclave NOR car on bascule sur le front descendant.