

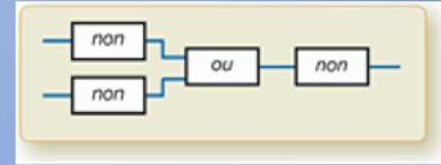
Le temps est ce qui permet d'éviter de tout faire en même temps.

Dans ce chapitre, nous voyons comment les circuits électroniques prennent le temps en compte.

Nous voyons d'abord comment fabriquer un circuit mémoire.

Puis, comment un circuit particulier, l'horloge, permet de synchroniser tous les autres.

Otto Schmitt (1913-1998) est un pionnier du *génie biomédical*. En 1934, en étudiant la propagation de l'influx nerveux dans les nerfs des calmars, il a compris qu'un circuit en boucle fermée positive – c'est-à-dire dans lequel la sortie est connectée à l'entrée, sans inversion de valeur – avait deux états stables et pouvait donc être utilisé pour mémoriser une grandeur. En électronique, une *bas-cule de Schmitt* est une forme de circuit bistable, qui utilise cette idée de boucle fermée positive



Circuit combinatoire et circuit séquentiel

Les circuits que nous avons vus dans le chapitre précédent, par exemple le circuit illustré ci-dessus, ont des entrées (deux à gauche sur la figure) et des sorties (une à droite sur la figure) et l'état des sorties est déterminé par celui des entrées.

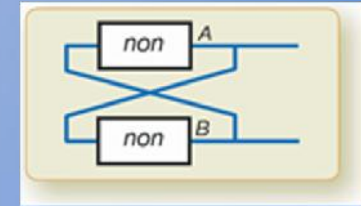
Dans cet exemple, la sortie est dans l'état 1 quand les deux entrées sont dans l'état 1 et elle est dans l'état 0 quand au moins l'une des entrées est dans l'état 0.

L'état des sorties à un instant donné dépend de l'état des entrées à ce même instant, mais pas de l'état des entrées une seconde ou une minute plus tôt.

Un tel circuit, qui **ignore le temps**, s'appelle un circuit **combinatoire**.

Il y a, autour de nous, beaucoup de circuits combinatoires.

Par exemple, une lampe est allumée quand son interrupteur est fermé et elle est éteinte quand cet interrupteur est ouvert ; l'état de la lampe dépend de la position de l'interrupteur, mais pas de la position de l'interrupteur une seconde ou une minute plus tôt.



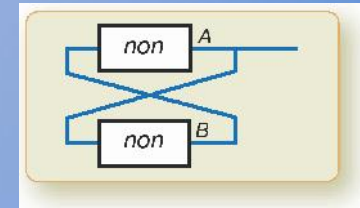
Le circuit séquentiel le plus simple est le circuit mémoire un bit qui permet de mémoriser un 0 ou un 1.

Construire un tel circuit n'est pas difficile, mais il faut procéder en plusieurs étapes.

La première est de construire un circuit qui a deux états stables, par exemple celui de la figure ci-dessus.

Ce circuit a deux états stables car :

- Si la sortie *A* de la première porte *non* est dans l'état 0, alors l'entrée de la seconde porte *non*, qui est *A* également, est aussi dans l'état 0 ; par conséquent, sa sortie *B* est dans l'état 1, donc l'entrée de la première porte, qui est *B* également, est dans l'état 1, ce qui participe à perpétuer le fait que sa sortie *A* soit dans l'état 0.
- Si, en revanche, la sortie *A* de la première porte *non* est dans l'état 1, alors l'entrée de la seconde porte *non*, qui est *A* également, est aussi dans l'état 1 ; par conséquent, sa sortie *B* est dans l'état 0, donc l'entrée de la première porte, qui est *B* également, est dans l'état 0, ce qui participe à perpétuer le fait que sa sortie *A* soit dans l'état 1.



Autrement dit, les deux états stables de ce circuit sont :

- $A = 0$ et $B = 1$,
- $A = 1$ et $B = 0$.

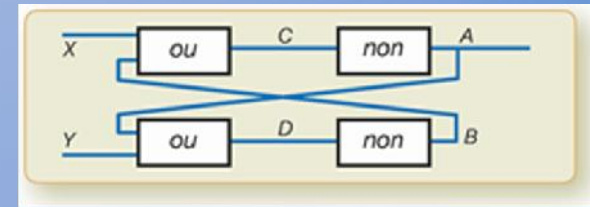
En supprimant la sortie B et en ne gardant que la sortie A on obtient un circuit qui a deux états stables.

La sortie A vaut 0 dans le premier et 1 dans le second. On peut donc dire que ce circuit *mémorise* la valeur 0 dans le premier cas et la valeur 1 dans le second.

Ce circuit est donc un *circuit mémoire*.

Toutefois, ce circuit ayant une sortie, mais pas d'entrée, il n'est pas possible de changer son état et donc la valeur mémorisée.

Pour ce faire, il faut construire un circuit un peu plus complexe, en ajoutant deux portes **ou**.



Tant que les entrées X et Y sont dans l'état 0, tout se passe comme dans le circuit précédent. En effet, si la sortie A de la première porte **non** est dans l'état 0, alors le point D à l'entrée de la seconde est dans l'état 0 également, car $0 \text{ ou } 0 = 0$. Et si la sortie A est dans l'état 1, le point D est dans l'état 1 également, car $1 \text{ ou } 0 = 1$.

Le point D est donc dans le même état que la sortie A dans les deux cas. De même, le point C à l'entrée de la première porte **non** est dans le même état que B.

Tout se passe donc comme si les deux portes **ou** n'étaient pas là.

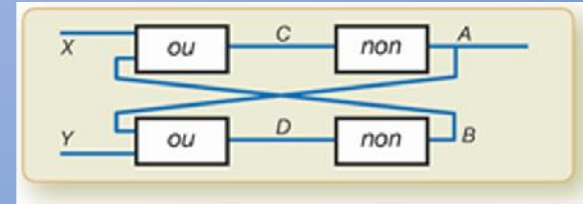
En revanche, si pendant une courte durée on met l'entrée X dans l'état 1 tout en laissant l'entrée Y dans l'état 0, alors le point C passe dans l'état 1 quelle que soit la valeur de B car $1 \text{ ou } 0 = 1$ et $1 \text{ ou } 1 = 1$; la sortie A passe donc dans l'état 0, le point D également et le point B passe dans l'état 1.

On force donc le circuit à se mettre dans l'état $A = 0$ et $B = 1$, c'est-à-dire à mémoriser la valeur 0.

Et quand l'entrée X sera revenue à la valeur 0, le circuit restera dans cet état.

Au cours d'étapes qui s'enchaînent très vite, l'état des points *C*, *D*, *A* et *B* devient :

X	Y	C	D	A	B
1	0				
1	0	1			
1	0	1		0	
1	0	1	0	0	
1	0	1	0	0	1



De même, si pendant une courte durée, on met l'entrée *Y* dans l'état 1 tout en laissant l'entrée *X* dans l'état 0, on force le circuit à mémoriser la valeur 1. Ce circuit mémorise donc une valeur 0 ou 1 et, en stimulant l'entrée *X* ou l'entrée *Y*, on peut changer la valeur mémorisée.

Ce circuit s'appelle une **bascule RS (Reset-Set)** et on peut le représenter comme ci-dessous.



Table de vérité d'une bascule RS :

R	S	Q
0	0	état précédent
0	1	1
1	0	0
1	1	état interdit

Exercice 1

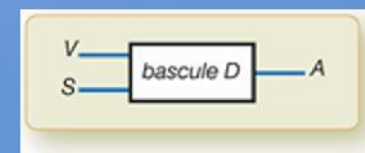
- a) Construire un circuit combinatoire à deux entrées V et S et à deux sorties X et Y tel que, si l'entrée S est dans l'état 0, alors X et Y sont dans l'état 0 et si S est dans l'état 1, alors X est dans l'état non V et Y est dans l'état V .
- b) Connecter ce circuit combinatoire avec une bascule RS pour obtenir un circuit qui :
- quand S est dans l'état 0, ne change pas d'état et produit la valeur mémorisée sur la sortie A ,
 - quand S est dans l'état 1 et V dans l'état 0, mémorise la valeur 0 et met la sortie A dans l'état 0,
 - quand S est dans l'état 1 et V dans l'état 1, mémorise la valeur 1 et met la sortie A dans l'état 1.

Exercice 2

Quand on utilise plusieurs composants identiques dans un même circuit, on peut différencier leurs entrées et sorties en leur donnant un numéro : ainsi, si l'on a plusieurs verrous D, on appellera V_1 l'entrée V du premier, A_2 la sortie du deuxième, etc.

On définit un circuit séquentiel appelé **bascule D** en reliant la sortie A_1 d'un premier verrou D avec l'entrée V_2 d'un second verrou D, et en reliant l'entrée S_1 du premier à une porte **non** dont la sortie est raccordée à l'entrée S_2 du second. Si l'on considère ce circuit dans son intégralité, ses entrées sont donc l'entrée V_1 du premier verrou D et une entrée S unique qui alimente à la fois S_1 directement et S_2 via la porte non. Sa sortie est la sortie A_2 du second verrou D.

- Dessiner ce circuit.
- Montrer que quand S est dans l'état 0, le second verrou met sa valeur à jour, mais pas le premier, et quand S est dans l'état 1, le premier verrou met sa valeur à jour, mais pas le second.



Circuit asynchrone et circuit synchrone

Quand on assemble des circuits séquentiels qui interagissent les uns avec les autres, on obtient un circuit **asynchrone** : l'état de chacun des circuits évolue dans le temps, en fonction de l'état des circuits auxquels il est connecté, mais de manière relativement désordonnée.

Beaucoup d'interactions que nous avons avec les autres sont asynchrones : dans un grand magasin, par exemple, chaque client fait ses courses relativement indépendamment des autres.

Et quand plusieurs clients veulent payer leurs courses en même temps, il se forme une file d'attente. De ce fait, chaque client est à peu près sûr qu'il finira par payer et sortir, mais il n'a pas de garantie *a priori* sur le temps que cela prendra.

Circuit asynchrone et circuit synchrone

À l'inverse, un petit nombre de nos interactions doivent être **synchrones** : jouer dans un orchestre demande non seulement de jouer toutes les notes de la partition dans un certain ordre, mais également de les jouer à un moment donné.

Les clients d'un magasin peuvent faire leurs courses chacun à son rythme, mais les musiciens d'un orchestre, les danseurs d'une chorégraphie ou les soldats qui marchent au pas, doivent agir dans une même temporalité.

Une manière d'obtenir cette synchronie est de confier à l'un des musiciens, le chef d'orchestre, le rôle de battre la mesure.

L'horloge

En informatique, les machines de grande taille, par exemple les réseaux, sont des machines asynchrones.

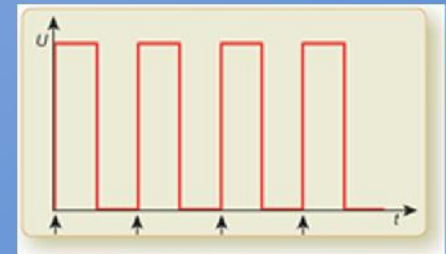
Chaque utilisateur va à son rythme et le réseau finit par répondre à tout le monde, mais sans garantie du temps que cela prendra.

En revanche, les machines de petite taille, telles que les processeurs, sont des machines synchrones.

C'est pour cela qu'il y a dans les ordinateurs un circuit, l'**horloge**, dont le rôle est de battre la mesure pour les autres circuits.

Une horloge est simplement un circuit qui émet sur sa sortie un signal périodique, par exemple le signal suivant.

Chaque flèche sur la figure marque le début d'un cycle. Avec ce type d'horloge, la sortie est à 1 pendant la première moitié du cycle et à 0 pendant la seconde.



Chacun des circuits, en particulier les circuits mémoires, se synchronise sur un signal d'horloge.

Par exemple, si on connecte la sortie de l'horloge sur l'entrée S d'une bascule D, on obtient un circuit qui enregistre la valeur de l'entrée V à chaque cycle.

Fréquence d'horloge

La fréquence d'horloge, qui est souvent indiquée dans les caractéristiques techniques des ordinateurs, est le nombre de cycles par seconde.

Par exemple, quand la fréquence d'horloge est 1 GHz, la période de l'horloge, c'est-à-dire la longueur d'un cycle, est de 1 nanoseconde, un milliardième de seconde.

Exercice 3

Lorsque l'on relie l'entrée S d'une bascule D à une horloge, montrer que la sortie A se comporte comme l'entrée V , mais avec au maximum un cycle d'horloge de retard.