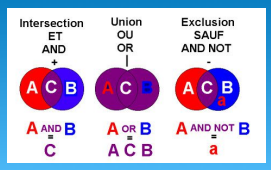
Circuits et logique booléenne



Transistors, portes logiques, tables de vérité, circuits combinatoires

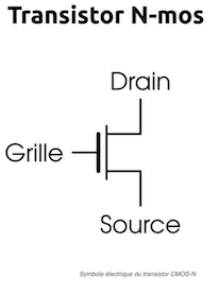
Présentation

Pour simplifier, on dit souvent que les circuits d'un ordinateur (mémoire, microprocesseur) manipulent uniquement des chiffres binaires 0 et 1. Cette affirmation mérite d'être précisée...

À la base de la plupart des composants d'un ordinateur, on retrouve le transistor. Précisons que l'invention du transistor a été un immense progrès, mais les premiers ordinateurs (comme l'ENIAC) sont antérieurs à cette invention : on utilisait alors des tubes à vide.

tube à vide

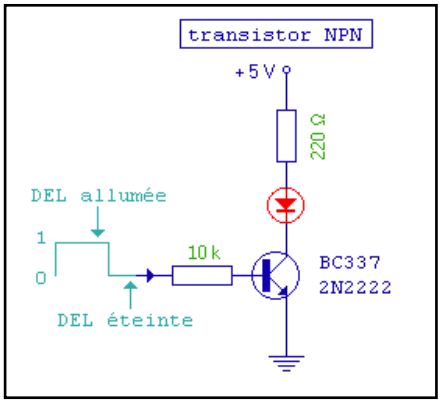
transistor



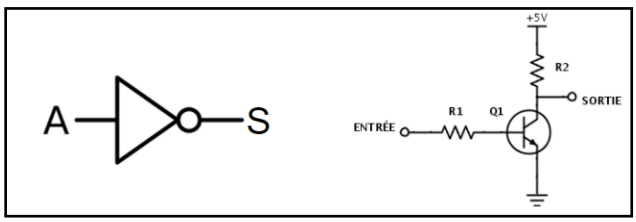
Actuellement on ne trouve plus de transistors en tant que **composant électronique** seul (comme sur la photo ci-dessus). Dans un ordinateur, les transistors sont regroupés au sein de ce que l'on appelle des **circuits intégrés**. Dans un circuit intégré, les transistors sont désormais gravés directement sur des plaques de silicium.

Fonctionnement

Les transistors que l'on trouve dans les circuits des ordinateurs se comportent comme des interrupteurs qui laissent ou non passer un courant électrique, selon le mode du tout ou rien : soit le courant passe, soit le courant ne passe pas. Il faut bien avoir conscience qu'il n'y a pas dans un ordinateur des petits « 1 » ou des petits « 0 » qui circulent... En réalité, les chiffres binaires 0 et 1 représentent simplement des tensions électriques. Ainsi le chiffre binaire 0 représente une tension électrique basse (état « bas ») et le chiffre binaire 1 représente une tension électrique haute (état « haut »).



Le transistor est l'élément de base des circuits logiques. Un circuit logique permet de réaliser une opération booléenne. Ces opérations booléennes sont directement liées à l'algèbre de Boole (que nous n'étudierons pas ici...). En 1938, Claude Shannon fait le lien entre l'algèbre de Boole et des composants électriques. C'est le début de l'ère de l'électronique numérique. Un circuit logique prend en entrée un ou des signaux électriques et renvoie en sortie un ou des signaux électriques ; les entrées et sorties sont dans un état haut (« 1 ») ou à un état bas (« 0 »).



Il existe deux catégories de circuits logiques :

- les circuits combinatoires (les états en sortie dépendent uniquement des états en entrée)
- les circuits séquentiels (les états en sortie dépendent des états en entrée ainsi que du temps et des états antérieurs)

Les variables booléennes

- En programmation, plutôt que d'utiliser 0 et 1, on dit qu'une variable booléenne peut prendre deux valeurs : FALSE ou TRUE. Ces variables sont du type Bool (ou Boolean) en informatique.

```
def check(x):
    if x+1 is 1+x:
        return False
    if x+2 is not 2+x:
        return False
    return True
```

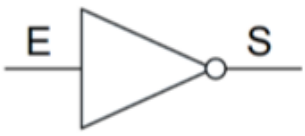
- 1) **En langage Python, quelle différence il y-a-il entre les instructions** a=0 et b==0 ?
- 2) **En langage Python, pourquoi l'instruction if total=0 ne fonctionne pas ?**

Les portes logiques et tables de vérité

Une porte logique est une fonction qui prend en entrée un ou plusieurs bits et qui renvoie un bit en sortie. Les différents opérateurs logiques sont décrits par des tables de vérité et sont représentés par des symboles dans les circuits électroniques.

La porte logique NON

Un simple transistor permet de réaliser la porte logique « porte NON ». Cette porte logique est la plus simple : elle n'a qu'un seul bit en entrée et sa sortie vaut 0 quand l'entrée vaut 1 ; et inversement la sortie vaut 1 quand l'entrée vaut 0.

Porte NON (NOT Gate)	Table de vérité
	E (Entrée) | S (Sortie) 1 | 0 |
	Notations : $\bar{E}$ $(\neg E)$

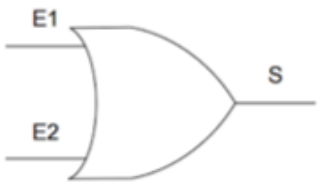
La porte logique ET

On peut fabriquer d'autres portes logiques en combinant plusieurs transistors: par exemple la porte logique ET. La porte logique ET a deux entrées E1 et E2, et une sortie S.

Porte ET (AND Gate)	Table de vérité
	E1 E2 S 0 0 0 1 1 0 1 1 Notations : $E_1 \cdot E_2$ $(E_1 \wedge E_2)$

La porte logique OU

La porte logique OU a deux entrées E1 et E2, et une sortie S.

Porte OU (OR Gate)	Table de vérité
	E1 E2 S 0 0 0 1 1 0 1 1 Notations : $E_1 + E_2$ $(E_1 \vee E_2)$

Les fonctions booléennes OU, ET, NON forment une base complète et permettent de construire toutes les autres fonctions

booléennes. Par exemple, les portes logiques XOR, NAND, NOR,...

La porte logique OU EXCLUSIF (XOR)

La porte logique XOR a deux entrées E1 et E2, et une sortie S. Elle retourne 1 en sortie si une seule des entrées est à 1.

Porte OU EXCLUSIF (XOR Gate)	Table de vérité
	E1 E2 S 0 0 0 1 1 0 1 1 Notation : $E_1 \oplus E_2$