

Opérateurs logiques

1 Les portes logiques

Dans le processeur et d'autres composants électroniques, on a des **transistors** qui sont des **semi-conducteurs**. Ils ont la particularité de laisser passer ou non le courant électrique.

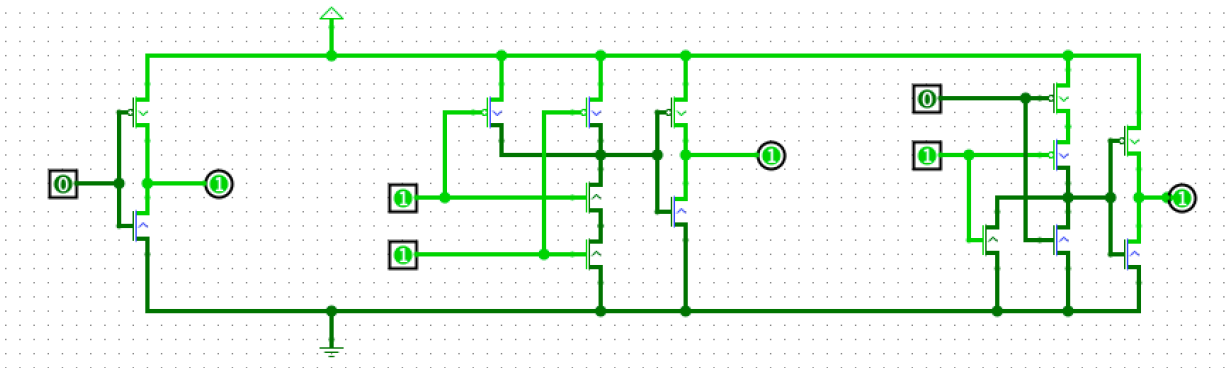
Il existe deux types de transistor : les PNP et les NPN.

En les associant, ils vont constituer des **portes logiques** qui modifient les courants électriques et donc les valeurs des bits 0 et 1.

Exemple

Sur la figure ci-dessous, avec l'application java *logissim*, on a reproduit:

- à gauche la porte logique **not**;
- au centre la porte logique **et**;
- à droite la porte logique **ou**.



1.1 La porte logique NOT

La porte NOT a un seul bit d'entrée et un seul bit de sortie.

- Si le bit d'entrée vaut 1, alors il vaut 0 en sortie.
- Si le bit d'entrée vaut 0, alors il vaut 1 en sortie.

On donne les symbolisations de la porte NOT et la table logique:

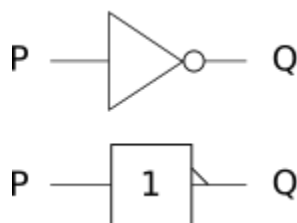


Table de vérité

P	Q
0	1
1	0

1.2 La porte logique ET

La porte ET a 2 bits en entrée et un seul bit de sortie.

- Si les 2 bits d'entrée valent 1, alors le bit de sortie vaut 1.
- Si un bit d'entrée ou les 2 valent 0, alors le bit de sortie vaut 0.

On donne les symbolisations de la porte ET et la table logique:

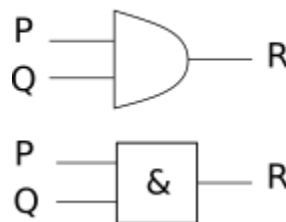


Table de vérité

P	Q	P ET Q
0	0	0
1	0	0
0	1	0
1	1	1

1.3 La porte logique OU

La porte OU a 2 bits en entrée et un seul bit de sortie.

- Si les 2 bits d'entrée valent 0, alors le bit de sortie vaut 0.
- Si un bit d'entrée ou les 2 valent 1, alors le bit de sortie vaut 1.

On donne les symbolisations de la porte OU et la table logique:

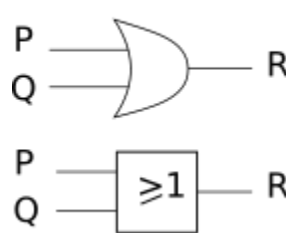


Table de vérité

P	Q	P OU Q
0	0	0
1	0	1
0	1	1
1	1	1

2 Algèbre de Boole

George Boole est un mathématicien britannique qui s'est intéressé à l'algèbre des opérateurs logiques.

Les 3 **opérateurs** logiques de base appelés aussi **opérateurs booléens** sont NOT, AND et OR.

Les valeurs appliquées aux opérateurs sont les **opérandes**. Ils ont pour valeur soit **True (1)**, soit **False (0)**.

Le résultat d'une expression logique est soit **True**, soit **False**.

- L'opérateur NOT s'applique à un seul opérande x ; il se note $\text{NOT}(x)$ ou $\neg(x)$ ou \bar{x} .

A	NOT(A)
True	False
False	True

- L'opérateur AND s'applique à deux opérandes x et y ; il se note x AND y ou $x \wedge y$ ou $x.y$.

A	B	A AND B
False	False	False
False	True	False
True	False	False
True	True	True

- L'opérateur OR s'applique à deux opérandes x et y ; il se note x OR y ou $x \vee y$ ou $x + y$.

A	B	A OR B
False	False	False
False	True	True
True	False	True
True	True	True

Pour évaluer une expression booléenne, on rassemble les différents cas possibles, selon les valeurs des opérandes, dans une table appelée **table de vérité**.

2.1 Expressions booléennes

Les expressions booléennes respectent des règles de priorité pour être évaluées convenablement comme en mathématiques:

1. Les parties entre parenthèses sont d'abord évaluées;
2. L'opérateur NOT est le premier opérateur évalué;
3. L'opérateur AND est le second opérateur évalué;
4. L'opérateur OR est le dernier opérateur évalué.

Exemple

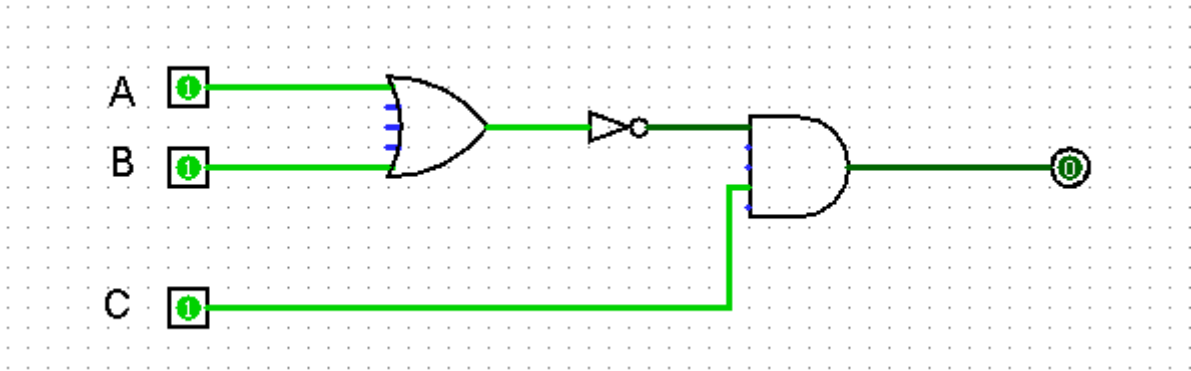
La table de vérité de l'expression booléenne A AND B OR C .

On écrit la table de vérité de chaque variable puis de l'expression:

A	B	C	A AND B	A AND B OR C
False	False	False	False	False
False	False	True	False	True
False	True	False	False	False
False	True	True	False	False
True	False	False	False	False
True	False	True	False	True
True	True	False	True	True
True	True	True	True	True

Exemple

Le circuit logique de l'activité peut s'évaluer comme une expression booléenne:



L'expression booléenne de ce circuit est : $\text{NOT}(A \text{ OR } B) \text{ AND } C$

On écrit la table de vérité de chaque variable puis de l'expression:

A	B	C	A AND B	NOT(A OR B)	NOT(A OR B) AND C
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	0	0

Remarque

Si une expression booléenne contient N variables, la table de vérité contient 2^N lignes.