

La logique combinatoire - algèbre de Boole

Date de dernière mise à jour : 27/06/2007 à 19:36

Source : <http://www.vulgarisation-informatique.com/logique-combinatoire.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

L'algèbre de Boole qui tire son nom du mathématicien anglais Georges Boole a pour fonction de traduire des signaux électriques en expressions mathématiques. Ainsi, à chaque signal correspond une variable logique qui sera traitée par une fonction logique. Ces fonctions peuvent être simplifiées grâce à l'algèbre de Boole et à des identités remarquables de base.

On représente ainsi des circuits logiques grâce à des symboles avant de les réaliser physiquement.

Les expressions logiques sont utilisées en électronique, et donc dans nos PC. Un processeur par exemple est composé de transistors assemblés permettant de réaliser des fonctions très simples (addition, soustraction, etc).

Variable logique :

Elle se note avec un caractère. Elle peut prendre deux valeurs logiques : 0 ou 1 (langage binaire, le seul compréhensible par l'ordinateur).

Fonction logique :

C'est une fonction permettant d'avoir plusieurs états de sortie en fonction des entrées.

Ex : $S = a + b + cd$

Portes logiques :

Les opérations simples faites par les transistors s'établissent à l'aide de portes logiques. Voici la liste des portes logiques :

-La fonction **ET (AND)**- : la sortie est égale à 1 si les deux variables sont égales à 1, sinon elle est égale à 0. Elle se représente par un point (dans les fonctions logiques)

-La fonction **OU (OR)**- : la sortie est égale à 0 si les deux variables sont à 0, sinon elle est égale à 1. Elle se représente par un signe + dans les fonctions logiques

-La fonction **NON (NO)**- : elle est égale au contraire de la variable. Si la variable est à 0, la sortie sera à 1, et inversement.

-La fonction **OU EXCLUSIF (XOR)**- : elle est égale à 1 si une des deux entrées sont égales à 1 mais pas les deux.

Les portes logiques sont aussi regroupées en deux portes principales utilisées en électronique :

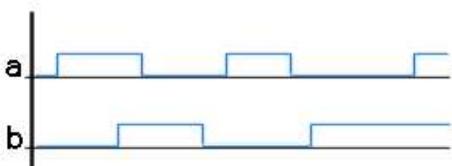
-Les portes **NAND (NON ET)** : -elles sont le regroupement d'une fonction **NON**- et d'une fonction **ET**-)

-Les portes **NOR (NON OU)** : -elles sont le regroupement d'une fonction **NON**- et d'une fonction **OU**-)

Chronogramme :

L'état d'un signal se représente graphiquement à l'aide de chronogrammes, qui comme leur nom le stipule indiquent la valeur du signal en fonction du temps.

Exemple d'un chronogramme avec la fonction **OU** :



Fonctions logiques booléennes :

Les opérateurs des fonctions logiques sont représentés par des signes :

-L'opérateur **ET** -se représente par un point

-L'opérateur **OU**- se représente par un signe +

-L'opérateur **NON**- se représente par un slash (/) devant la variable ou par une barre au dessus de la variable :

-L'opérateur **OU exclusif (XOR)** -se représente par un signe + encerclé

Table de vérité :

La table de vérité est un tableau rassemblant toutes les entrées avec toutes les possibilités de sorties (2^n variables pour le binaire)

Exemple de la table de vérité de la fonction **OU** :

A

B

S

1

1

1

1

1

1

Les lois de composition :

Les lois de composition sont des règles simple d'écriture permettant de simplifier les expressions algébriques booléennes.

- L'associativité : $A.(B.C)$ - est équivalent à $B.(C.A)$ - ou $C.(B.A)$
- L'absorption : $A.(A.B)$ - est équivalent à $A.B$ -, de même que $A+/A.B$ - est équivalent à $A+B$
- La commutativité : $A.B$ - est équivalent à $B.A$ - . $A+B$ - est équivalent à $B+A$
- La distributivité-: $A.(B+C)$ - est équivalent à $A.B+ A.C$
- L'idempotence : $A.A$ - est équivalent à A - et $A+A$ - est équivalent à A
- L'inversion : $A./A$ - est équivalent à - . $A+/A$ - est équivalent à 1
- L'identité : $A.1$ - est équivalent à A -. $A+0$ - est équivalent à A -. $A.0$ - est équivalent à

Théorème de DE MORGAN :

$$A.B = /A + /B$$

Exemple de simplification d'équation :

$$S = /a + /b + a.b.c + a.b./c + /a.b.c$$

$$s = /a.(1+bc) + ab.(c+/c) + /b$$

$$s = /a + ab + /b$$

$$s = /a + b + /b$$

$$s = 1$$

Source : <http://www.vulgarisation-informatique.com/logique-combinatoire.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)