Institut Superieur d'Enseignement Professionnel de Diamniadio (ISEP)

Promo 3

Circuits combinatoires

M. Mansour KHOUMA

April 25, 2022

Notions Fondamentales

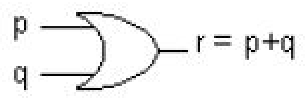
Les fonctions logiques combinatoires directement issues des mathématiques (**algèbre de Boole**) sont les outils de base de l'électronique numérique, donc de l'automatisme et de l'informatique. Elles sont utilisées en électronique sous forme de **portes logiques** pour de nombreuses applications en informatique et dans la conception des circuits électroniques.

Addition Logique ou OU Logique

Le comportement de la fonction logique OU a été reproduit à travers des circuits électroniques appelés aussi **portes OU**.

p	q	p+q
0	0	0
0	1	1
1	0	1
1	1	1

$\Longrightarrow$

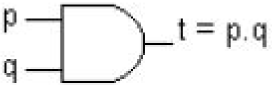


Multiplication Logique ou ET Logique

Le comportement de la fonction logique ET a été reproduit à travers des circuits électroniques appelés aussi **portes ET**..

p	q	p.q
0	0	0
0	1	0
1	0	0
1	1	1

$\Longrightarrow$

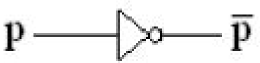


Inversion Logique ou NON Logique

Le comportement d'opération NON a été reproduit à travers des circuits électroniques symbolisé par :

p	$\overline{p}$
0	1
1	0

$\Longrightarrow$

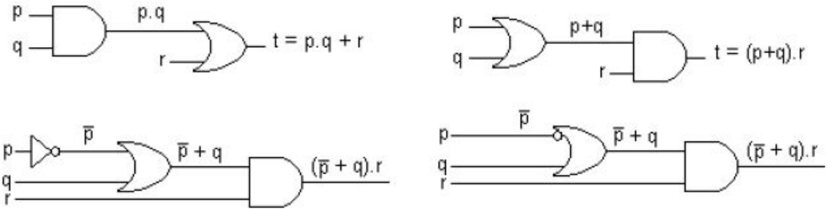


Remarque : Le petit rond, indique toujours une inversion.

Ecriture des circuits logiques sous forme algébrique

Tout circuit logique quelque soit son degré de complexité peut être décrit au moyen des opérations booléennes ET, NON, OU. Ces trois circuits constituent les éléments fondamentaux. On peut ainsi associer à tout circuit une équation booléenne, et réciproquement.

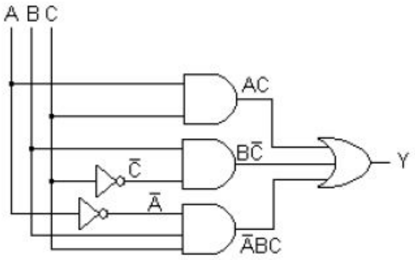
Exemples :



Matérialisation d'un circuit à partir d'une expression booléenne

La connaissance de l'expression booléenne permet de déterminer le circuit permettant de réaliser la fonction en question.

Exemples : Soit $Y = AC + B\overline{C} + \overline{A}BC$

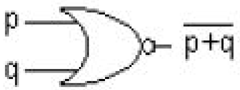


Porte NI(NOR)

Par combinaison avec le NON, l'addition logique OU donne la porte de type **NI** (**NOR** en anglais).

p	q	p+q	$\overline{p+q}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

$\Longrightarrow$

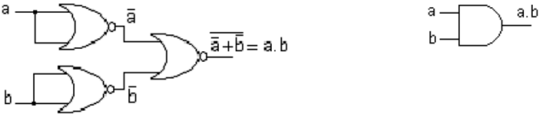


Réalisatio porte NI(NOR)

Inversion

OU

ET

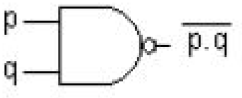


Porte NON-ET(NAND)

Par combinaison avec le NON, la multiplication logique ET donne la porte de type **NON-ET** (**NAND** en anglais).

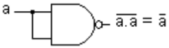
p	q	p.q	$\overline{p.q}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

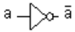
$\Longrightarrow$



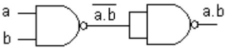
Réalisatio porte NON-ET (NAND)

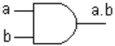
Inversion



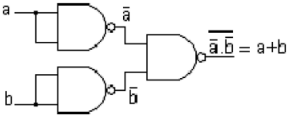


ET





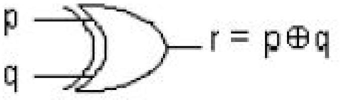
OU



La fonction OU EXCLUSIF, souvent appelée **XOR** (**eXclusive OR**), est un opérateur qui est très utilisé en électronique, en informatique, et aussi en cryptographie du fait de ses propriétés intéressantes. Son symbole est traditionnellement un signe plus inscrit dans un cercle $\bigoplus$.

p	q	p $\bigoplus q$
0	0	0
0	1	1
1	0	1
1	1	0

$\Longrightarrow$



NI Exclusive

La fonction **NI EXCLUSIF**, souvent appelée **NXOR** (Not eXclusive OR), est un opérateur qui est en fait la négation de la fonction XOR.

p	q	$p \bigoplus q$	$\overline{p \bigoplus q}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Propriétés :

$$p \bigoplus q = \overline{p}q + p\overline{q}$$

$$\overline{p \bigoplus q} = pq + \overline{pq}$$

$$p \bigoplus 1 = \overline{p}$$

$$p \bigoplus 0 = p$$

$$p \bigoplus p = 0$$

$$p \bigoplus \overline{p} = 1$$

La manipulation des fonctions logiques nécessite très souvent une étape de simplification pour plusieurs raisons. De manière générale, la simplification est faite pour baisser la consommation électrique du circuit, le coût du production mais aussi l'encombrement du dispositif. Les contraintes techniques ont essentiellement pour origine une limitation du nombre de portes ou du nombre d'entrées possibles ou bien encore le temps de propagation ou même enfin le nombre de connexion acceptables.

Complément d’une fonction

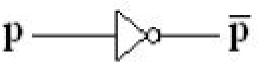
Pour avoir le complément une fonction, il faut remplacer chaque variable de la fonction par son complément, chaque OU par un ET et chaque ET par un OU.

Exemple :

$$p = A + \overline{A}B + (C + D)$$

$$\longrightarrow \overline{p} = \overline{A + \overline{A}B + (C + D)}$$

$$\longrightarrow \overline{p} = \overline{A}.(A + \overline{B}).\overline{C}.D$$



Première forme canonique

La première forme canonique procède à la lecture sur les états logiques 1; c'est à dire que si une variable est égale à 1, on l'écrit sous sa forme directe, par contre si elle est égale à 0, on l'écrit sous sa forme complémentée.

A	B	C	p
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$$p = \overline{A}.B.\overline{C} + \overline{A}.B.C + A.\overline{B}.C$$

Deuxième forme canonique

La deuxième forme canonique procède à la lecture sur les états logiques 0; c'est à dire que si une variable est égale à 0, on l'écrit sous sa forme directe, par contre si elle est égale à 1, on l'écrit sous sa forme complémentée.

A	B	C	p
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

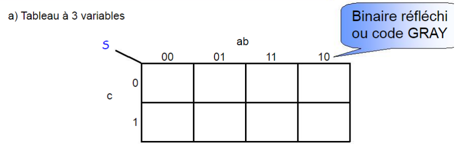
$$p = (A+B+C).(A+B+\overline{C}).(\overline{A}+B+C).(\overline{A}+\overline{B}+C).(\overline{A}+\overline{B}+\overline{C})$$

Diagramme de Karnaugh

Il peut arriver que la simplification des expressions logiques soit difficile. Il devient alors préférable de passer par une méthode graphique de simplification dite **méthode de Karnaugh**. Cette dernière repose sur une table de vérité où 2 termes logiques **adjacents** sont disposés de manière à être géométriquement **adjacents**. On utilise ainsi pour ce faire la séquence des nombres de Gray.

Diagramme de Karnaugh

Dans le diagramme de Karnaugh, n variables logiques correspondent à 2^n cases.



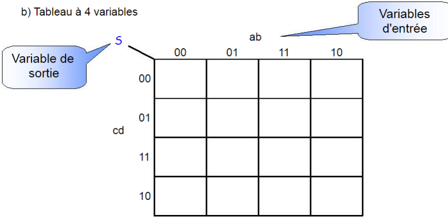
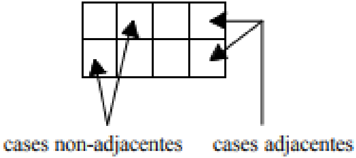


Diagramme de Karnaugh

Simplification

Pour simplifier à partir du diagramme de Karnaugh, on réalise des groupements de cases adjacentes. Ces groupements de cases doivent être de taille maximale (nombre de cases max.) et égale à un multiple de 2^n, avec n étant un entier quelconque. On cesse d'effectuer les groupements lorsque tous les « 1 » appartiennent au moins à l'un d'eux.



4	2	2	4
3	1	1	3
3	1	1	3
4	2	2	4

Les cases portant le même chiffre sont des exemples de cases adjacentes
→ regroupement possible.

Exemples de Simplification

Les groupements de cases étant égale à un multiple de 2^n, la **réunion de doublet** est favorable à la simplification.

cd \ ab	$\bar{a}\bar{b}$ 00	$\bar{a}b$ 01	ab 11	$a\bar{b}$ 10
$\bar{c}\bar{d}$ 00	1	1	1	1
$\bar{c}d$ 01	1	1	1	
cd 11	1	1	1	1
$c\bar{d}$ 10	1		1	1

Ce chapitre est l'objet de voir les circuits logiques combinatoires par :

- les portes logiques et leur caracteristique
- ainsi les simplifications.