Université Sultan Moulay Slimane
Ecole Supérieure de Technologie
Département Mécatronique

Electronique Numérique

Chapitre 3 : Les portes logiques de base

Pr. ARSALANE

Plan du chapitre

I. Définitions

II. Les opérations logiques élémentaires

III. Les opérations logiques induites

IV. Les expressions logiques et leurs simplifications

V. Ecriture des fonctions booléennes

VI. Extraction d'une équation logique à partir d'une table de vérité

VII. Problème

VIII. Simplification des équations logiques : simplification par tableau de Karnaugh

I. Définitions

1. Algèbre de Boole : Système algébrique constitué de l'ensemble {0, 1}. Origine : Mathématicien anglais Georges Boole, 1815 – 1864

2. Variable binaire : On appelle une variable binaire, logique ou booléenne une variable prenons ces états de l'ensemble {0, 1}.

Exemple : L'état d'un interrupteur, un bouton poussoir, …

Soit **a** la variable associée à l'état d'un bouton poussoir, alors :

- a = 0 (faux, bas) signifie qu'il n'est pas actionné.
- a = 1 (vrai, haut) signifie qu'il est actionné.

3. Equation binaire :

On appelle une équation binaire, logique ou booléenne une combinaison de plusieurs variables logiques donnant l'état d'une variable dit de sortie associée. Cette combinaison est réalisée à l'aide des opérations logiques :

Soit X_i ($i \in [1, n]$) les variables d'entrées

L'équation $A = f(X_i)$ définit l'état de la variable de la sortie A.

3. Table de vérité :

La table de vérité représente l’état de la variable de sortie pour chacune des combinaisons des n variables d’entrée (2^n lignes).

1. Opérateur NON (NOT) :

L’opérateur NON est la fonction unitaire qui affecte à la variable de sortie l’état complémentaire de la variable d’entrée.

Equation : $s = \overline{a}$

Symbole

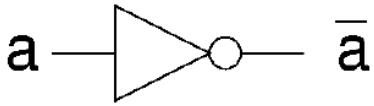
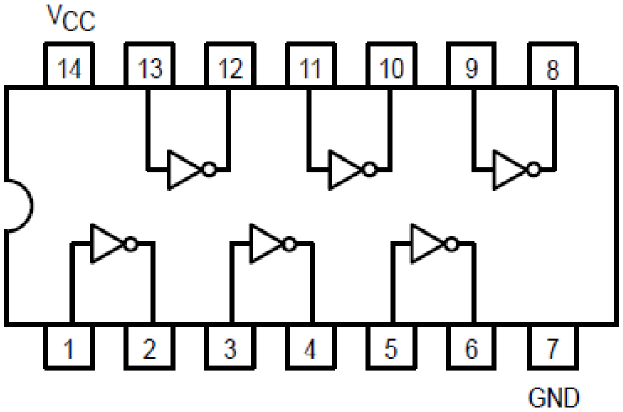


Table de vérité :

a	$s = \overline{a}$
0	1
1	0



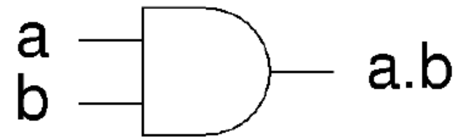
Circuit intégré : 74LS04

2. Opérateur ET (AND) :

L'opérateur ET est un produit logique. Le signe est celui de la multiplication (.), mais on lis ET. C'est un opérateur binaire qui affecte à la variable de sortie l'état '1' si et seulement si les variables d'entrée sont à '1' simultanément.

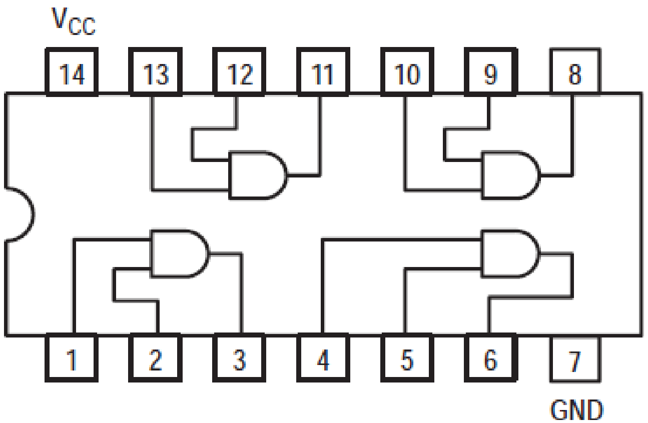
Equation : s = a.b

Symbole :



a	b	s = a.b
0	0	0
0	1	0
1	0	0
1	1	1

Table de vérité



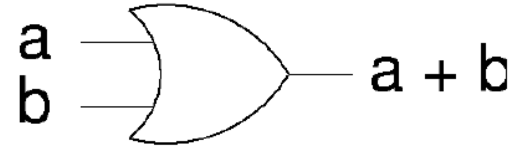
Circuit intégré : 74LS08

3. Opérateur OU (OR) :

L'opérateur OU est la somme logique. Le signe est celui de l'addition (+), mais on lis OU. C'est un opérateur binaire qui affecte à la variable de sortie l'état '1' si et seulement si une variable d'entrée est à '1'.

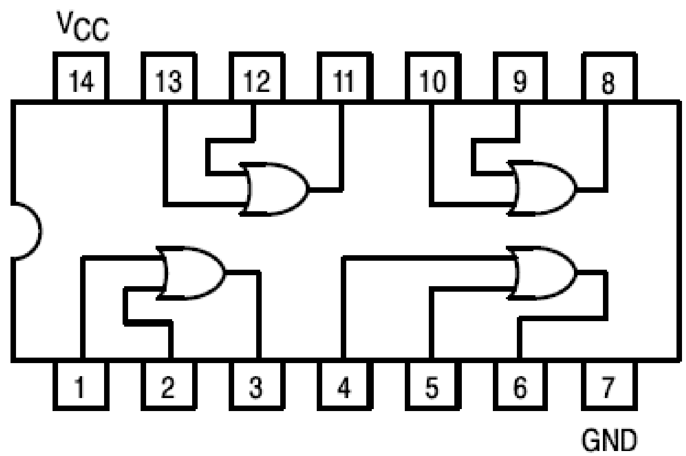
Equation : s = a + b

Symbole :



a	b	s = a+b
0	0	0
0	1	1
1	0	1
1	1	1

Table de vérité



Circuit intégré : 74LS32

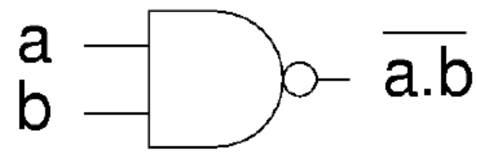
1. Opérateur NON-ET (NAND) :

Cette fonction logique est le résultat de l'association d'un NON et d'un ET.

C'est un opérateur binaire qui affecte à la variable de sortie l'état '0' si et seulement si les variables d'entrées sont à '1' simultanément.

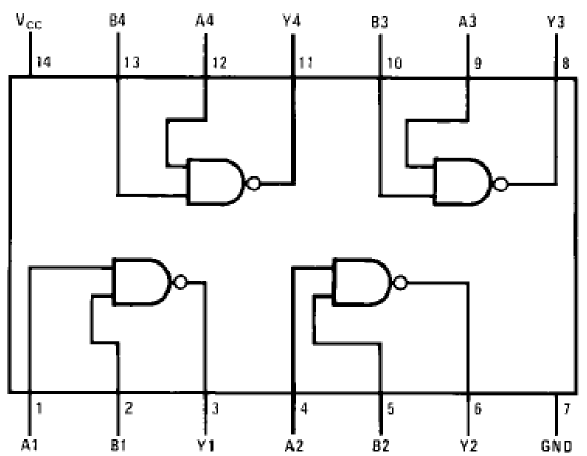
Equation : $s = \overline{a.b}$

Symbole



a	b	$s = \overline{a.b}$
0	0	1
0	1	1
1	0	1
1	1	0

Table de vérité



Circuit intégré : 74LS00

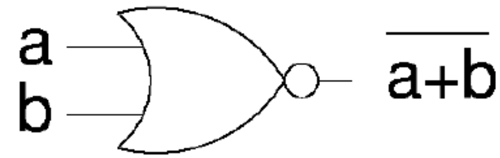
2. Opérateur NON-OU (NOR) :

Cette fonction logique est le résultat de l’association d’un NON et d’un OU.

C’est un opérateur binaire qui affecte à la variable de sortie l’état ‘1’ si et seulement si les variables d’entrées sont à ‘0’ simultanément.

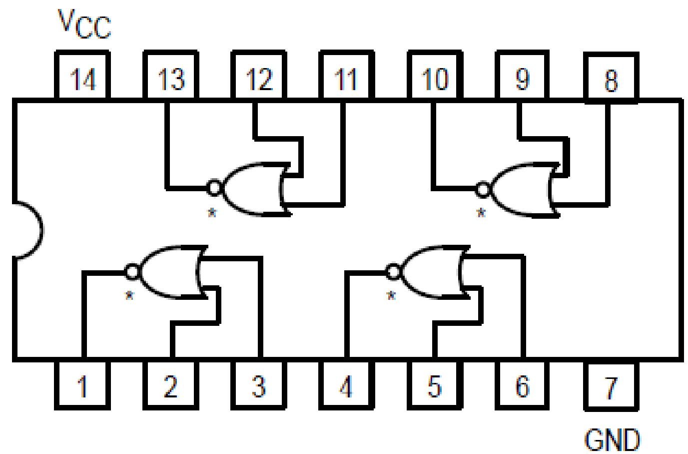
Equation : $s = \overline{a + b}$

Symbole :



a	b	$s = \overline{a + b}$
0	0	1
0	1	0
1	0	0
1	1	0

Table de vérité

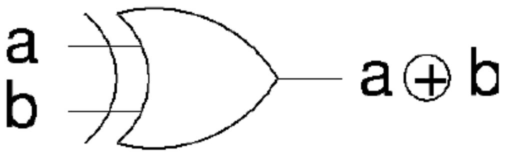


Circuit intégré : 74LS33

3. Opérateur OU exclusif (XOR) :

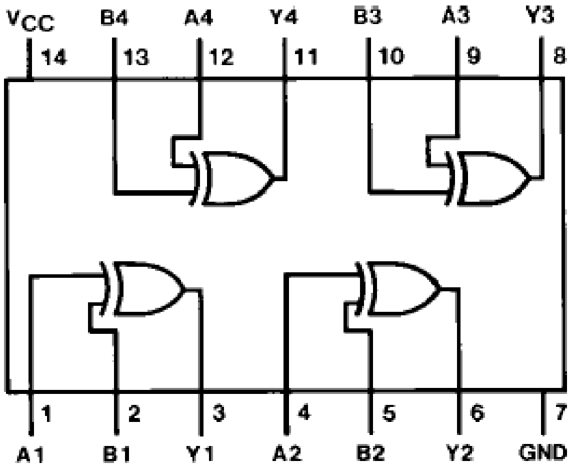
C’est un opérateur logique binaire qui ne prend la valeur ‘1’ que si une seule des entrées est à ‘1’.

Equation : $s = a \oplus b$ ou bien $s = a\bar{b} + \bar{a}b$ **Symbole** :



a	b	$s = a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Table de vérité



Circuit intégré : 74LS86

4. Autres opérateurs :

- Pour chaque opérateur à 2 entrées :
- Variantes à 3, 4, ... entrées (mais toujours une seule sortie)
- Généralisation de la fonction logique de base à plus de 2 variables en entrée
- Le symbole graphique utilisé est identique mais avec plus de 2 entrées

Exemples :

- Porte ET à 3 entrées a, b et c a pour expression logique : s = a.b.c
- Porte NOR à 4 entrées a, b, c et d a pour expression logique :

$$s = \overline{a + b + c + d}$$

Tous les opérateurs précédents permettent de combiner des variables pour en construire des nouveaux.

Exemple:

$c = a + b.d$

$d = e + f$

Donc $c = a + b.(e + f)$

1. Propriétés de base :

- **Involution** : $\overline{\overline{a}} = a$
- **Idempotence** : $a + a = a$ $\quad a \,.\, a = a$
- **Complémentarité** : $a \,.\, \overline{a} = 0$ $\quad a + \overline{a} = 1$
- **Elément neutre** : $a = a \,.\, 1 = 1 \,.\, a$

  $a = a + 0 = 0 + a$
- **Absorbants** : $a + 1 = 1$ $\quad a \,.\, 0 = 0$

1. Propriétés de base (suite) :

- **Associativité** : (a . b) . c = a . (b . c)

  (a + b) + c = a + (b + c)

- **Distributivité** : a . (b + c) = a . b + a . c

  a + (b . c) = (a + b) . (a + c)

- **Règles de Morgan** : $\overline{a+b} = \overline{a}\,.\,\overline{b}$ $\quad \overline{a\,.\,b} = \overline{a} + \overline{b}$

- **Optimisations** : $a + \overline{a}b = a + b$

  $a + bc = (a+b).(a+c)$

2. Exemple : $f(a,b,c) = abc + \overline{a}bc + a\overline{b}c + ab\overline{c}$

2. Exemple : $f(a,b,c) = abc + \overline{a}bc + a\overline{b}c + ab\overline{c}$

$$f(a,b,c) = bc(a + \overline{a}) + a\overline{b}c + ab\overline{c}$$

$$f(a,b,c) = bc + a\overline{b}c + ab\overline{c}$$

$$f(a,b,c) = c\left(b + a\overline{b}\right) + ab\overline{c}$$

$$f(a,b,c) = c(b + a) + ab\overline{c}$$

$$f(a,b,c) = ac + bc + ab\overline{c}$$

$$f(a,b,c) = bc + a(c + b\overline{c})$$

$$f(a,b,c) = bc + a(c + b)$$

$$f(a,b,c) = bc + ac + ab$$

1. Définition :

- On appelle Minterme de n variables un produit logique de ces dernières (complémentées ou non). Avec n variables, on construit 2^n Mintermes.

Exemple : pour 2 variables a et b, on trouve 4 Mintermes : $ab, \overline{a}b, a\overline{b}, \overline{a}\overline{b}$

- On appelle Maxterme de n variables une somme logique de ces dernières (complémentées ou non). Avec n variables, on construit 2^n Maxtermes .

Exemple : pour 2 variables a et b, on trouve 4 Maxtermes : $a + b, \overline{a} + b, a + \overline{b}, \overline{a} + \overline{b}$

2. Première forme canonique :

La première forme canonique d'une fonction logique est composée d'une somme de Mintermes exclusivement. Pour une fonction donnée, cette forme est unique.

Remarque : La somme de tous les Mintermes de n variables vaux toujours '1'.

3. Deuxième forme canonique :

La seconde forme canonique d'une fonction logique est composée d'un produit de Maxtermes exclusivement. Pour une fonction donnée, cette forme est unique.

Remarque : Le produit de tous les Maxtermes de n variables vaux toujours '0

4. Exemples de formes canoniques :

Fonction à 3 variables a, b et c, exemples :

- Mintermes : $abc, \overline{a}bc, a\overline{b}\overline{c}, \overline{a}b\overline{c}$, ...
- Maxtermes : $a + b + c, a + \overline{b} + c, a + \overline{b} + \overline{c}, \overline{a} + b + \overline{c}$, ...
- Première forme canonique : $abc + \overline{a}bc + a\overline{b}\overline{c} + \overline{a}b\overline{c}$
- Seconde forme canonique : $(a + b + c) . \left(a + \overline{b} + c\right) . \left(a + \overline{b} + \overline{c}\right) . (\overline{a} + b + \overline{c})$

5. Passage aux formes canoniques :

Pour changer de forme canonique, on effectue une double complémentation de la fonction logique suivant les propriété de l'algèbre de Boole :

- $\overline{\overline{a}} = a$
- $a.\overline{a} = 0$
- $a + \overline{a} = 1$
- ….

6. Exemple de passage à la première forme canonique :

Soit $f(a,b,c) = ab + \overline{b}c + a\overline{c}$

- Premier minterme : ab
  - Il manque la variable c
  - Transforme ab en $ab(c+\overline{c})$ car $c+\overline{c}=1$
- Même chose pour les 2 autres mintermes
- D'où : $f(a,b,c) = ab(c+\overline{c}) + \overline{b}c(a+\overline{a}) + a\overline{c}(b+\overline{b})$

$$= abc + ab\overline{c} + a\overline{b}c + \overline{a}\overline{b}c + ab\overline{c} + a\overline{b}\overline{c}$$

7. Exemple de passage à la seconde forme canonique :

Soit $f(a,b,c) = ab + \overline{b}c + a\overline{c}$

On passe par $\overline{\overline{a}} = a$

Après développement : $\overline{f(a,b,c)} = \overline{a}b + \overline{a}bc + \overline{a}\,\overline{c} + \overline{a}\overline{b}\overline{c}$

Reste à transformer les mintermes à 2 variables :

$\overline{a}b + \overline{a}\,\overline{c} = \overline{a}b(c+\overline{c}) + \overline{a}\,\overline{c}(b+\overline{b})$

Au final : $\overline{f(a,b,c)} = \overline{a}bc + \overline{a}\overline{b}\overline{c} + \overline{a}b\overline{c}$

Et $f(a,b,c) = (a+\overline{b}+\overline{c})(a+b+c)(a+\overline{b}+c)$

VI. Extraction d’une équation logique à partir d’une table de vérité

Une table de vérité recense l’ensemble des états d’une sortie pour toutes les combinaisons possibles des variables d’entrées.

Pour trouver une expression sous la première forme canonique, on applique la méthode suivante :

- On définit les mintermes de n variables qui sont les expressions logiques bâtis sur la combinaison des n variables.
- Chaque Minterme est associé à l’une des combinaison de la table de vérité (on conservons la correspondance ‘1’ pour la variable et ‘0’ pour la variable complémentée)
- Tous les mintermes valant 1 sont sommés logiquement
- Pour obtenir l’expression logique de la sortie, il faut effectuer les simplifications en utilisant les propriétés de l’algèbre de Boole.

Trois interrupteurs a, b et c commandent l'allumage de deux lampes R et S suivant les conditions suivantes :

- Dès qu'un ou plusieurs interrupteurs sont activés, la lampe R s'allume.
- La lampe S ne doit s'allumer que si au moins 2 interrupteurs sont activés.

Trouver les expression logiques de R et S et dessiner le logigramme (circuit électronique)

VIII. Simplification des équations logiques de circuit logique : simplification par tableau de Karnaugh

- Le tableau de Karnaugh permet la simplification des équations logiques.
- Elle comporte 2^n cases, n étant le nombre de variable d'entrée, organisées selon le code Gray (4 variables → 16 cases).
- Chaque case correspond à une combinaison possibles des variables d'entrées.
- Chaque combinaison exprimée de l'équation logique sera représentée par '1' dans la case correspondante.
- Il est ensuite possible de regrouper les cases par 2, 4, 8, …, 2^n afin d'éliminer les variables qui change d'état:
  - Un regroupement de 21 cases élimine une variables
  - Un regroupement de 2n cases élimine n variables

VIII. Simplification des équations logiques de circuit logique : simplification par tableau de Karnaugh

- Il faut regrouper les valeurs de S égales à 1.
- Le nombre de 1 dans chaque groupe doit être égal à une puissance de 2.
- Les groupes formés doivent être les moins nombreux possibles, mais ils doivent englober tous les 1.
- Un 1 peut être inclus dans plus d'un groupe, par contre aucun 0 ne doit être inclus.
- Les groupes sont composés d'une ou plusieurs colonnes et d'une ou plusieurs lignes. Si possible, assemblez-les par valeurs d'entrées communes.

VIII. Simplification des équations logiques de circuit logique : simplification par tableau de Karnaugh

Exemple 1 : $T_1 = xyz + xy\bar{z} + \bar{x}y\bar{z} + \bar{x}yz$

x \ yz	00	01	11	10
0			1	1
1			1	1

$T_1 = y$

Exemple 2 : $T_2 = x\bar{y}\,\bar{z} + xy\bar{z} + xyz$

x \ yz	00	01	11	10
0				
1	1		1	1

$T_2 = {\color{red}xy} + {\color{green}x\bar{z}}$

VIII. Simplification des équations logiques de circuit logique : simplification par tableau de Karnaugh

Exemple 3 : $T_3 = yw + zw + \bar{z}w + \bar{x}y\bar{z}\,\bar{w} + xy\bar{z}$

xy\zw	00	01	11	10
00		1	1	
01	1	1	1	
11	1	1	1	
10		1	1	

$T_3 = {\color{red}w} + {\color{green}y\bar{z}}$

VIII. Simplification des équations logiques de circuit logique : simplification par tableau de Karnaugh

Exemple 4 : $T_4 = \mathrm{xyz} + \mathrm{z}(\mathrm{x}\overline{y} + \overline{x}y)$

x \ yz	00	01	11	10
0			1	
1		1	1	

$T_4 = \color{red}{yz} + \color{green}{xz}$

VIII. Simplification des équations logiques de circuit logique : simplification par tableau de Karnaugh

Exemple 5 : T_5

xy\zw	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11		1	1	
10		1	1	

$$T_5 = {\color{red}w} + {\color{green}\overline{x}}$$