Chapitre 2 : Algèbre de Boole & Fonctions logiques

Plan du cours :

I. Algèbre de Boole

1. Introduction

- George Boole est un mathématicien anglais (1815-1864).
- Il a fait des travaux dont les quels les fonctions (expressions) sont constitués par des variables qui ne peuvent prendre que les valeurs 'OUI' ou 'NON'.
- Ces travaux ont été utilisés pour faire l'étude des systèmes qui possèdent deux états qui s'excluent mutuellement :
  - Le système peut être uniquement dans deux états E1 et E2 tel qu'E1 est l'opposé d'E2.
  - Le système ne peut pas être dans l'état E1 et E2 en même temps
- Ces travaux sont bien adaptés au Système binaire (0 et 1).

Exemple de systèmes à deux états

- Un interrupteur est ouvert ou non ouvert (fermé)

- Une lampe est allumée ou non allumée (éteinte)
- Une porte est ouverte ou non ouverte (fermée)

Remarque :
On peut utiliser les conventions suivantes :

OUI → VRAI (true) OUI → 1 (Niveau Haut)
NON → FAUX (false) NON → 0 (Niveau Bas)

2. Définitions et conventions

2.1 Niveau logique : Lorsqu'on fait l'étude d'un système logique il faut bien préciser le niveau du travail.

Niveau	Logique positive	Logique négative
H (Hight) haut	**1**	**0**
L (Low) bas	**0**	**1**

Exemple :

Logique positive : lampe allumée : **1**, lampe éteinte : **0**
Logique négative : lampe allumée : **0**, lampe éteinte : **1**

2.2 Variable logique (Booléenne)

- Une variable logique (booléenne) est une variable qui peut prendre soit la valeur ***0*** ou ***1.***
- Généralement elle est exprimée par un seul caractère alphabétique en majuscule (A , B, S , …)

Exemple :

- Une lampe : allumée : ***L = 1,*** éteinte ***L = 0***
- Premier interrupteur ouvert : ***I1 =1,*** fermé : ***I1 =0***
- 2éme interrupteur ouvert : ***I2=1,*** fermé : ***I2=0***

3. Opérateurs logiques

3.1 Opérateurs logiques de base: NON, ET, OU

☞ NON :

Inverse la valeur d'une variable .

$F(A) = \text{Non } A = \overline{A}$ (A barre)

NOT
$\overline{0} = 1$
$\overline{1} = 0$

☞ ET (AND) :

Réalise le Produit logique entre deux variables binaires:

$F(A,B) = A \bullet B$

AND
$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

☞ OU (OR :

Réalise la Somme logique entre deux variables binaires:

$F(A,B) = A + B$

(Ne pas confondre avec la somme arithmétique)

OR
$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 1$

Remarque

- L'opérateur ET peut réaliser le produit de plusieurs variables logiques (ex : A . B . C . D).
- L'opérateur OU peut aussi réaliser la somme logique de plusieurs variables (ex : A + B + C +D).
- Dans une expression on peut aussi utiliser les parenthèses.

3.2 Priorité des opérateurs

Pour évaluer une fonction logique :

- Tout d'abord, effectuer toutes les inversions de termes simples
- évaluer les sous expressions entre les parenthèses.
- en suite le produit logique (ET)
- enfin la somme logique (OU)

Exemple :

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

si on veut calculer F(0,1,1) alors :

$F(0,1,1) = (\overline{0.1})(1\ +1)\ +\ 0.\overline{1}.1$

$F(0,1,1) = (\overline{0}\ )\ (1\ )\ +\ 0.0.1$

$F(0,1,1) = 1.1\ +\ 0.0.1$

$F(0,1,1) = 1\ +\ 0$

$F(0,1,1) = 1$

3.3 Autres opérateurs logiques

☞ **NAND (NON ET)**

$$F(A,\ B) = \overline{A\ .B}$$

La fonction NAND est fausse si les 2 variables sont vraies

A	B	$\overline{A \bullet B}$
0	0	1
0	1	1
1	0	1
1	1	0

☞ **NOR(NON OU)**

$$F(A,\ B) = \overline{A + B}$$

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

☞ **OU exclusif (XOR)**

$$F(A,\ B) = A \oplus B = \overline{A}.B + A.\overline{B}$$

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Remarque :

NAND et NOR sont des **opérateurs universels**

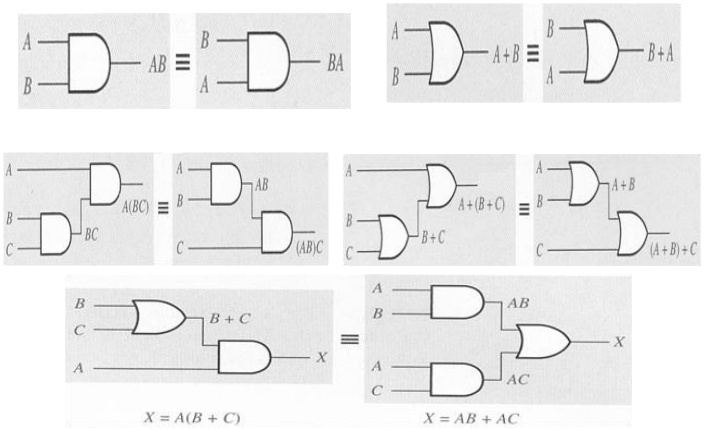
- on peut exprimer n'importe qu'elle fonction logique en utilisant uniquement l'opérateur NAND ou NOR.
- Pour cela, Il suffit d'exprimer les opérateurs de base (NON , ET , OU) avec l'opérateur NAND ou l'opérateur NOR.

4 Règles et théorèmes de l'algèbre de Boole

4.1 *Les lois fondamentales*

Les lois fondamentales de l'algèbre de BOOLE se vérifient en écrivant les tables de vérités et en testant tous les cas possibles. Ces lois sont les suivantes :

Commutativité	A.B = B.A A+B = B+A
Associativité	A.(B.C) = (A.B).C A+(B+C) = (A+B)+C
Distributivité	ET sur OU : A.(B+C) = (A.B) + (A.C) OU sur ET : A+(B.C) = (A+B) . (A+C)



4.2 Règle l'algèbre de BOOLE

Idempotence	A.A = A A+A = A
Complémentarité	$A.\overline{A} = 0$ $A+\overline{A} = 1$
Identités remarquables	1.A = A 1+A=1 0.A = 0 0+A = A

A partir de ces propriétés, on démontre les relations de base suivantes :

$A.B+A.\overline{B} = A$ $(A+B).(A+\overline{B}) = A$	Théorème d'inclusion
A + A.B = A A.(A+B) = A	Théorème d'absorption
$A+\overline{A}.B = A+B$ $A.(\overline{A}+B) = A.B$	Théorème d'allégement

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

$$A + AB = A(B+\overline{B}) + AB = AB + A\overline{B} = A(B+\overline{B}) = A$$
$$A(A+B) = A + AB = A$$

Remarque :

- Le OU exclusif possède les propriétés de commutativité et d'associativité, mais n'est pas distributif par rapport au ET ou au OU.
- NAND ET NOR possèdent la propriété de commutativité mais pas d'associativité

4.3 Théorèmes de DeMorgan

Le théorème de DE MORGAN complète les propriétés de l'algèbre de BOOLE. Il est indépendant du nombre de variables. Il s'énonce des deux manières suivantes :

a. La négation d'un produit de variables est égale à la somme des négations des variables.

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

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

b. La négation d'une somme de variables est égale au produit des négations des variables.

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

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

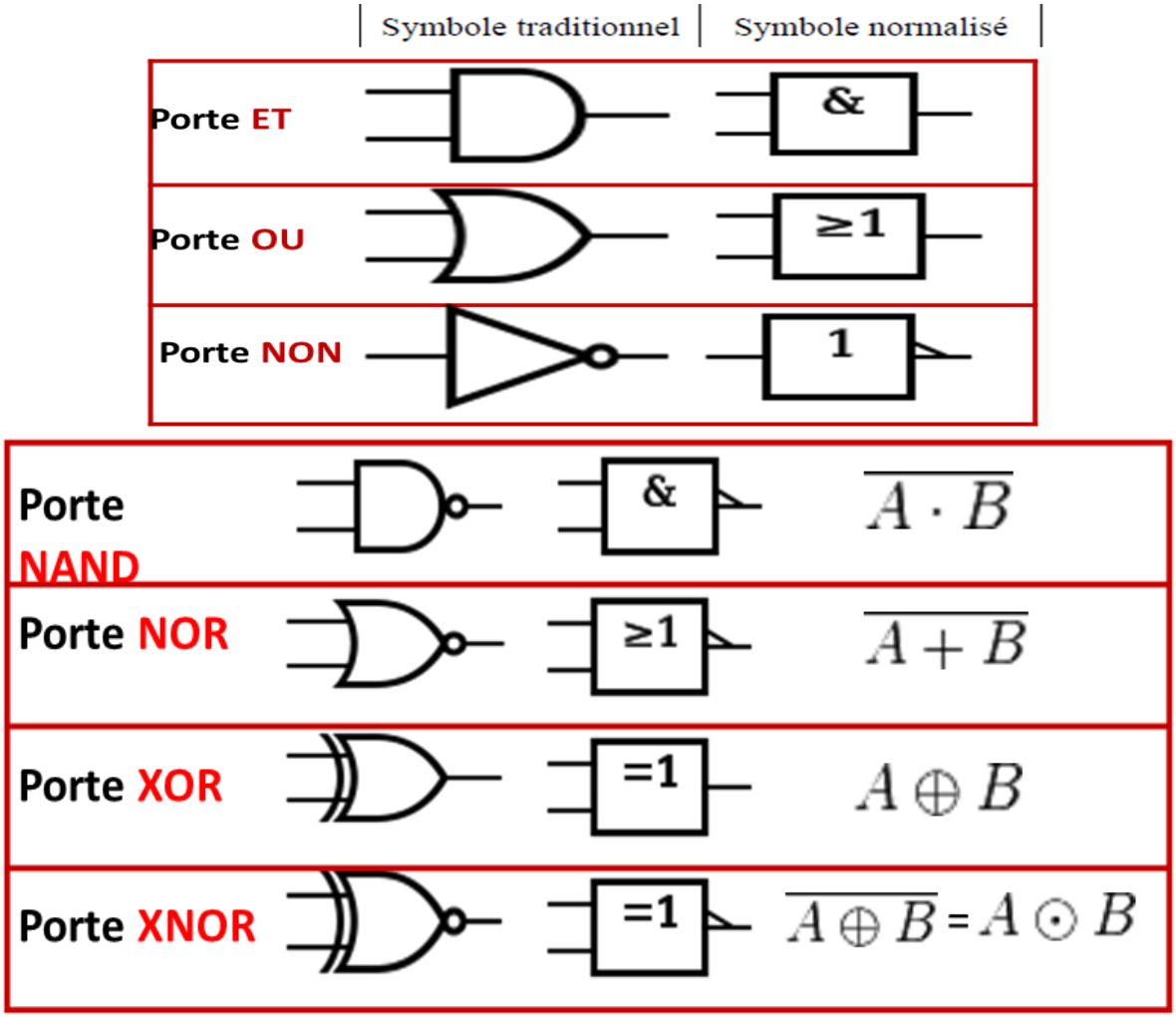
Généralisation du Théorème DE-MORGANE à N variables :

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

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

5 Portes logiques

Pour matérialiser une fonction logique on utilise des portes logiques qui sont des briques de base pour la construction des circuits numériques.



Remarque :

- Les portes ET, OU, NAND, NOR peuvent avoir plus que deux entrées
- Il n'existe pas de OU exclusif à plus de deux entrées

II. Fonctions logiques

1. Définition

Fonction logique : équation qui relie N variables logiques (ne pouvant prendre que les valeurs 0 ou 1), avec un ensemble d'opérateurs logiques (ET, OU, NON,...)

Exemple : $F(A,B,C) = (A+B).\left(\overline{B+\bar{C}+D}\right).A$

La valeur d'une fonction logique est égale $\mathbf{1}$ ***ou*** $\mathbf{0}$ selon la valeur des variables d'entrées

2. Formes d'une expression Booléenne

Une fonction logique, peut être écrite sous différentes formes :

- ***Forme disjonctive*** *:* une somme de produits
  Exemple : $\boldsymbol{F(a,b,c,d) = abc + bcd + acd}$
- ***Forme conjonctive*** *:* un produit de sommes
  Exemple : $F(\boldsymbol{a,b,c,d}) = (\boldsymbol{a+d+c}).(\boldsymbol{b+c+d}).(\boldsymbol{a+b+d})$
- ***Forme Canonique*** chaque terme contient l'ensemble des variables,
  Exemple1 :
  $\boldsymbol{F(a,b,c,d) = a\bar{b}cd + ab\bar{c}d + abcd}$ $\rightarrow$ ***Première forme canonique*** **;** ou bien ***forme canonique disjonctive*** ; Les termes de la somme sont appelés les ***mintermes***

  Exemple2 :
  $\boldsymbol{F(a,b,c) = (a+b+c).(a+\bar{b}+c).(\bar{a}+\bar{b}+c)}\rightarrow$ ***Seconde forme canonique*** ou bien ***forme canonique conjonctive,*** Les termes du produits sont appelés les ***maxtermes.***
- On peut passer d'une forme à l'autre $\rightarrow$ elles sont toutes équivalentes (voir TD)

3. Table de vérité (TV)

3.1 Définition

La table de vérité est un tableau qui indique la valeur de la fonction logique en fonction de toutes les combinaisons possibles des variables binaires d'entrée.
Si une fonction logique possède $\boldsymbol{N}$ variables binaires, alors il existe $\mathbf{2}^{\boldsymbol{N}}$ combinaisons possibles d'entrée.

Exemple 1

Soit F est une fonction logique à 2 variables, il y aura 2^2 soit 4 combinaisons possibles. La table de vérité de F est la suivante :

A	B	$F(A,B)$
0	0	$F(0,0)$
0	1	$F(0,1)$
1	0	$F(1,0)$
1	1	$F(1,1)$

Exemple 2

Soit la fonction logique : $F(A,B,C) = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$. Donnez la table de vérité de $F(A,B,C) = \bar{A}$

La fonction possede 3 variables d'entrées $\rightarrow 2^3 = 8$ combinaisons possibles.

$F(0,0,0) = 110 + 100 + 010 + 000 = 0$
$F(0,0,1) = 111 + 101 + 011 + 001 = 1$
$F(0,1,0) = 100 + 110 + 000 + 010 = 0$
$F(0,1,1) = 101 + 111 + 001 + 011 = 1$
$F(1,0,0) = 010 + 000 + 110 + 100 = 0$
$F(1,0,1) = 011 + 001 + 111 + 101 = 1$
$F(1,1,0) = 000 + 010 + 100 + 110 = 0$
$F(1,1,1) = 001 + 011 + 101 + 111 = 1$

A	B	C	(A,B,C)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

3.2 Extraction de la fonction logique à partir d'une TV

Exemple 3: A partir de la table de vérité suivante, donner l'expression de la fonction à trois variables $F(A,B,C) = ??$

Réponse 3 :

A	B	C	$F(A,B,C)$	Mintermes	Maxtermes
0	0	0	**0**	←	$\boldsymbol{A+B+C}$
0	0	1	**1**	← $\boldsymbol{\bar{A}.\bar{B}.C}$	
0	1	0	**0**	←	$\boldsymbol{A+\bar{B}+C}$
0	1	1	**1**	← $\boldsymbol{\bar{A}.B.C}$	
1	0	0	**0**	←	$\boldsymbol{\bar{A}+B+C}$
1	0	1	**1**	← $\boldsymbol{A.\bar{B}.C}$	
1	1	0	**0**	←	$\boldsymbol{\bar{A}+\bar{B}+C}$
1	1	1	**1**	← $\boldsymbol{A.B.C}$	

Minterme: Produit qui est vrai pour les entrées de la ligne.

Maxterme: Somme qui est fausse pour les entrées de la ligne.

On peut déduire à partir de cette table plusieurs expressions de F qui sont toutes équivalentes

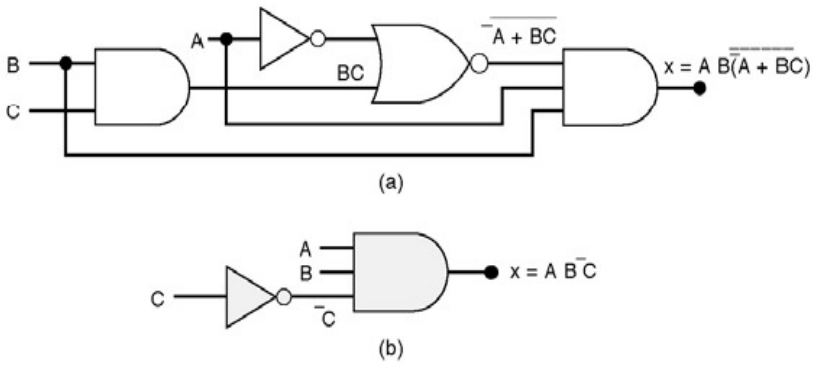
- Forme numérique : $\boldsymbol{F(A,B,C) = \sum(1,3,5,7)}$
- 1ere forme canonique :$\boldsymbol{F(A,B,C) = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC}$ (somme des mintermes pour lesquels $F(A,B,C) = 1$)
- 2ème forme canonique : $\boldsymbol{F(A,B,C) = (A+B+C)(A+\bar{B}+C)(\bar{A}+B+C)(\bar{A}+\bar{B}+C)}$ (produit des maxtermes pour lesquels$F(A,B,C) = 0$)

Remarque : la fonction F n'est pas simplifiée

4. Simplification d'une fonction logique

Avant de matérialiser une fonction booléenne à l'aide des portes logiques, il est nécessaire de chercher l'expression la plus simple de cette fonction.

Objectif : trouver la forme minimale (minimum de termes, minimum de variables dans chaque terme) ce qui permet une réalisation simple, rapide, à moindre coût.



Cette simplification peut être faite soit par :

- des méthodes purement algébriques.
- Ou des méthodes graphiques

4.1 Méthode algébrique.

Utilise les lois et les théorèmes de l'algebre de Boole

Exemple1 : $A + AB + BC = A(1+ B) + BC = A + BC$

Exemple2 :
$$\begin{aligned} AB + A\overline{C} + BC &= AB(C + \overline{C}) + A\overline{C} + BC \\ &= ABC + AB\overline{C} + A\overline{C} + BC \\ &= BC(1 + A) + A\overline{C}(1 + B) \\ &= BC + A\overline{C} \end{aligned}$$

Exemple3 :
$$\begin{aligned} \overline{A\overline{B}\overline{C} + \overline{A}CD + B\overline{C}} &= \overline{(\overline{A} + B + \overline{C}) + (\overline{A} + \overline{C} + \overline{D}) + B\overline{C}} \\ &= \overline{(\overline{A} + B + \overline{C} + \overline{D}) + B\overline{C}} \\ &= \overline{(\overline{A} + B + \overline{C} + \overline{D})}.\overline{(B\overline{C})} \\ &= A\overline{B}CD.(\overline{B} + C) \\ &= A\overline{B}CD \end{aligned}$$

Remarque :

Cette méthode est rapide pour des problèmes simples, efficacité selon l'expérience. . .
Mais, il est facile de rater une simplification, notamment quand la fonction est compliquée.

4.2 Méthode Graphique – Table de Karnaugh (TK)

a. Définition

- Méthode graphiques de simplification
- consiste à mettre en évidence par un tableau tous les termes qui sont adjacents (**code gray**). Deux termes sont dits adjacents, s'ils différent dans une seule variable logique.

  Exemple: ABC *et* $A\overline{B}C$ sont adjacents, seule la variable B qui diffère.

  ABC *et* $\overline{A}B\overline{C}$ ne sont pas adjacents, les variables A *et* C diffèrent
- La table de Karnaugh est applicable aux équations logiques qui sont constituées en particulier avec moins de 6 variable. Le nombre de cases du tableau dépend des nombres de variables, et égal à 2^n (***n*** étant le nombre de variables).

AB	00	01	11	10

AB	0	1
0		
1		

ABC	00	01	11	10
0				
1				

CD				
00				
01		?		
11				
10			??	

Tables de Karnaugh à 2, 3 et 4 variables

- Chaque case correspond donc à un minterme ; par exemple la case**?** Correspond à $\bar{A}B\bar{C}D$ **, et** la case**??** Correspond à $ABC\bar{D}$
- Les cases sont placées d'une façon telle que les mintermes adjacents ont une frontière commune sur une ligne ou sur une colonne, ou bien se trouvent aux extrémités d'une ligne ou d'une colonne

b. Etapes de simplification par la table de Karnaugh

- Remplissage de ta table de karnaugh à partir de la table de verité ou de la forme canonique de la fonction à simplifier; (écrire dans chaque case la valeur de la fonction)
- Faire des groupements de cases adjacentes contenants des 1 (de 1, 2, 4, 8 cases, toujours une puissance de 2);
  - Toutes les cases à 1 du tableau doivent être incluses dans au moins un regroupement
  - On doit créer des regroupements les plus gros possibles
  - Une case à 1 peut appartenir à plusieurs regroupements si cela permet de créer des blocs plus gros
- A chaque regroupement correspond un terme formé comme suit
  - on élimine les variables qui changent d'état et on conserve le produit des variables qui n'ont pas changé d'état dans le regroupement;
- La fonction logique simplifiée est le OU de tous les termes des regroupements trouvés

Exemple1 :

Table pour 2 variables

a	b	f(a,b)
0	0	0
0	1	1
1	0	1
1	1	1

table de karnaugh →

b \ a	0	1
0	0	1
1	1	1

- 2 regroupements de 2 cases adjacentes :
- Pour le vertical : on a toujours a = 1 donc cela donne le terme a
- Le regroupement horizontal donne le terme b
- $\underline{f(a,b) = a + b}$

Exemple2 :

Table pour 3 variables

a	b	c	g
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

table de Karnaugh →

c \ ab	00	01	11	10
0	0	0	1	1
1	1	0	1	1

Pour le regroupement le plus petit :

- **a** passe de **0** à **1**, on ne la prendra pas en compte
- **b** reste à **0** et **c** reste à **1**
- Il est donc représenté par le terme $\bar{b}\,c$

Pour le regroupement le plus gros :

- a **reste** à $1, b$ **passe de** 0 à 1 **et** c **passe de** 0 à 1
- On ne conserve que les variables qui ne changent pas, on a donc le terme a

Par conséquent : $g(a, b, c) = a + \bar{b}\,c$

Exemple3 :

$$F(a,b,c,d) = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + ab\bar{c}d + a\bar{b}\bar{c}d + a\bar{b}\bar{c}\bar{d} + \bar{a}bcd + abcd + a\bar{b}cd + \bar{a}bc\bar{d}$$

cd \ ab	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	1	1
10	0	1	0	0

3 regroupements :

- 8 cases : d
- 4 cases : $b\,c$
- 2 cases : abc

Au final :

$$F(a,b,c,d) = d + \bar{b}\,\bar{c} + \bar{a}bc$$

c. *Fonctions incomplétement définies*

Il se peut qu'une fonction logique ne soit pas definie pour certaines combinaisons d'entrées (qui ne peuvent jamais se produire, ou n »ont pas d'effet sur le résultat), se sont des états indetérminés qu'on notera **Ø ou X.** Ces états peuvent prendre 0 ou 1 et peuvent contribuer aux simplifications dans les tables de Karnaugh.

Exemple

A B C	F(A,B,C)
0 0 0	∅
0 0 1	0
0 1 0	1
0 1 1	∅
1 0 0	0
1 0 1	0
1 1 0	∅
1 1 1	1

➔

AB \ C	0	1
00	∅	0
01	1	∅
11	∅	1
10	0	0

➔ F(A,B,C) = B

Le symbole Ø peut prendre indifféremment la valeur 0 ou 1; on remplace donc par 1 uniquement ceux qui permettent de simplifier une expression par regroupement.

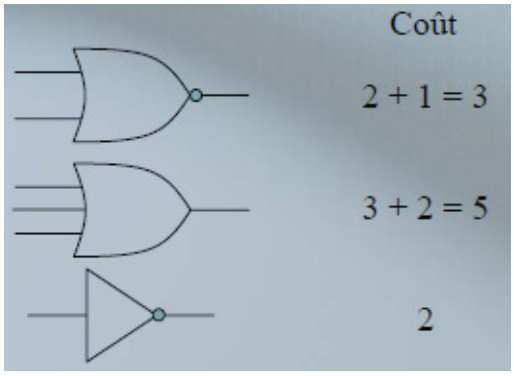
5. Materialisation d'une fonction logique

5.1 Cout d'un circuit logique

Lors de la simplification d'une équation logique, il est souhaitable d'obtenir la forme d'équation équivalente qui produira un circuit de coût minimum

Une métrique permet d'évaluer le coût d'un circuit :

- Porte logique avec sortie inversée avec N entrées : coût de $\boldsymbol{N+1}$
- Porte logique avec sortie non inversée avec N entrées : coût $de\ \boldsymbol{N+2}$
- XOR, XNOR à N entrées : coût de $\boldsymbol{2N}$



Note : Cette métrique représente un estimé de coût du circuit qui est plus ou moins représentatif de la réalité en fonction de la technologie utilisée pour implémenter le circuit.

Exemples de calcul de coût:

On considère que l'on a accès à l'inverse des signaux d'entrée

Exemple1

$$\boldsymbol{S = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C}$$

✓ Si on utilise les portes ET et OU

$$\left.\begin{matrix}4\ portes\ ET\ \text{à}\ 3\ entrées\ \rightarrow 4(3+2)\\ +\\ une\ porte\ ou\ \text{à}\ 4\ entrées \rightarrow (4+2)\end{matrix}\right\} \rightarrow \boldsymbol{cout = 20+6=26}$$

✓ Si on utilise des portes NAND

$$\boldsymbol{S = \overline{A}\overline{B}C + \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C}$$

$$\boldsymbol{= \overline{\overline{\overline{A}\overline{B}C}\,.\,\overline{\overline{A}BC}\,.\,\overline{A\overline{B}\overline{C}}\,.\,\overline{A\overline{B}C}}}$$

$$\left.\begin{matrix}4\ portes\ NAND\ \text{à}\ 3\ entrées\ \rightarrow 4(3+1)\\ +\\ une\ porte\ NAND\ \text{à}\ 4\ entrées \rightarrow (4+1)\end{matrix}\right\} \rightarrow \boldsymbol{cout = 16+5=21}$$

Il vaut mieux donc d'implémenter la seconde expression de S

5.2 Caracteristiques des portes logiques

Nous allons voir quelques caractéristiques électriques des portes logiques

a. Familles logiques

Le comportement des porteslogiques dépend de la technologie utilisée ; Il existe principalement 2 grandes familles de technologies :

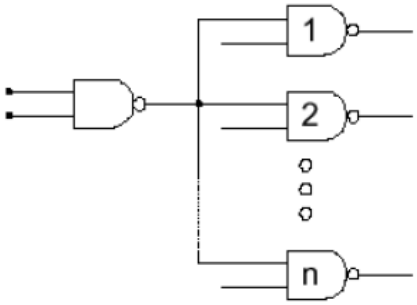
✓ Famille **TTL (Transistor Transistor Logic)** , réalisés à partir de transistors bipolaires NPN ou PNP. Il existe 7 sous familles logiques en technologie TTL :
- **TTL Standard : 74xx ,** Famille logique qui n'est plus couramment utilisée maintenant.
- **TTL Low Power : 74Lxx,** Circuits logiques TTL à faible consommation de courant.
- **TTL Schottky : 74Sxx,** Circuits logiques rapide à base de diodes Schottky.
- **TTL Fast : 74Fxx** Circuit logique à temps de propagation très faible, augmentant donc la rapidité.
- **TTL Low Power Schottky : 74LSxx,** Circuit TTL associant l'avantage des circuits 74Lxx et des 74Sxx, c'est à dire rapide et à faible consommation de courant.
- **TTL Advanced Schottky : 74ASxx,** Circuits logiques très rapides, pouvant travailler à de grandes fréquences d'horloges.
- **TTL Advanced Low Power Schottky : 74ALSxx,** Ces circuits associes les avantages des 74ASxx et 74Lxx : faible consommation et très rapides

✓ Famille **CMOS - Complementary Metal Oxide Semiconductor,** réalisée à partir de transistors MOSFET → **40xxx**

Les principaux Paramètres des Familles Logiques sont :
- Sortance
- retards de propagation,
- Alimentation
- Marge de bruit,

b. Sortance (Fan out)

La ***sortance*** (appelée aussi facteur de charge) est, définie comme le nombre maximal d'entrées logiques standard qui peuvent être commandées par la sortie d'une porte logique. Par exemple, quand il est indiqué qu'une porte logique a une sortance de ***n***, cela signifie qu'elle peut

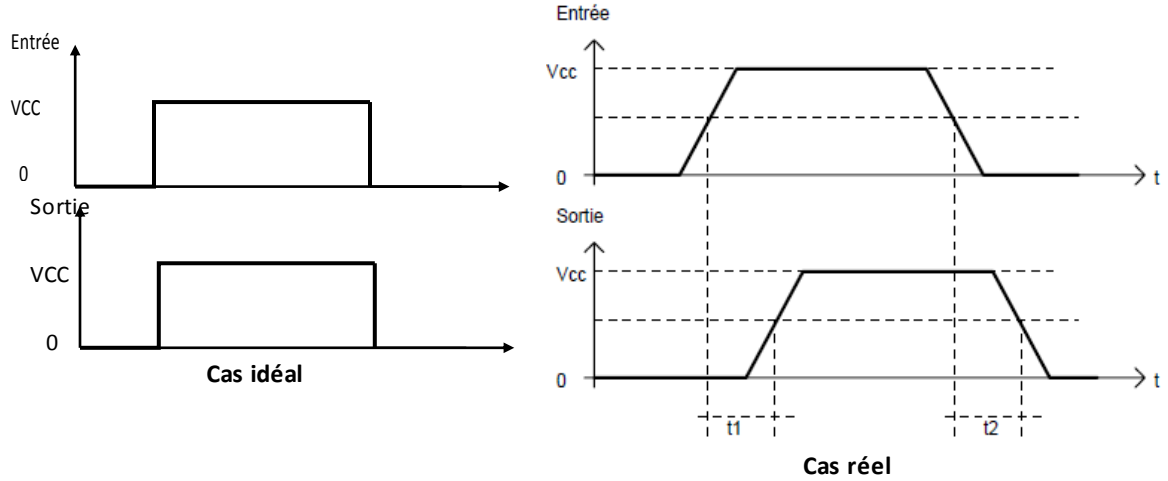


piloter ***n*** entrées logiques standard sans problème. Si on dépasse ce nombre, le fonctionnement normal de la porte n'est pas assuré. Sortance d'une porte TTL-standard = 10

c. Retards de propagation

Nous avons jusqu'à présent considéré les circuits logiques comme étant parfaits ; en particulier nous avons considéré que la réponse d'une porte logique à un signal d'entrée est instantanée ; En réalité ce n'est pas le cas, car un signal logique qui traverse une porte logique subit toujours un retard. Deux retards de propagation sont définis (ça se mesure en nanosecondes):

- $\boldsymbol{t_1 = tp_{LH}}$: retard pour passer du niveau logique 0 au niveau logique 1.
- $\boldsymbol{t_2 = tp_{HL}}$: retard pour passer du niveau logique 1 au niveau logique 0.
- Le temps de propagation est défini par : $\boldsymbol{tp = \frac{tp_{HL}+tp_{LH}}{2}}$



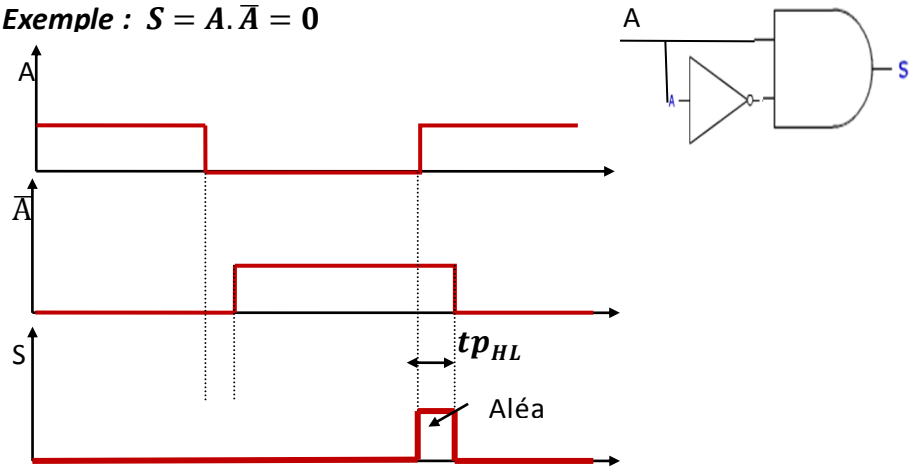
$\frac{\mathbf{1}}{\boldsymbol{tp}}$: Est appelée **fréquence de commutation,** elle correspond à la vitesse maximale d'une porte, c.a.d. le nombre de fois maximal qu'elle peut changer d'état par secondes.

Ce temps ainsi que la fréquence de commutation permettent de juger de la rapidité des circuits logiques.

Remarque :

Ces retards de propagation peuvent induire ***des Aléas ;*** un aléa est une impulsion parasite qui apparait à la sortie d'une porte lors des commutations.

Exemple : $\boldsymbol{S = A.\overline{A} = 0}$



d. Niveaux logiques et immunité au bruit

- $\boldsymbol{VIH}\ (\boldsymbol{min})$: tension d'entrée niveau HAUT: niveau de tension nécessaire pour avoir un logique en entrée.
- $\boldsymbol{VIL}\ (\boldsymbol{max})$: tension d'entrée niveau BAS: niveau de tension nécessaire pour avoir un 0 logique en entrée.
- $\boldsymbol{VOH}\ (\boldsymbol{min})$: tension de sortie niveau HAUT: niveau de tension de la sortie d'un circuit logique correspondant à l'état logique 1.
- $\boldsymbol{VOL}\ (\boldsymbol{max})$: tension de sortie niveau BAS: niveau de tension de la sortie d'un circuit logique correspondant à l'état logique 0.

Exemple de technologie TTL

- $\boldsymbol{VIH}\ (\boldsymbol{min}) = 2{,}0V$
- $\boldsymbol{VIL}\ (\boldsymbol{max}) = 0{,}8V$
- $\boldsymbol{VOH}\ (\boldsymbol{min}) = 2{,}4V$
- $\boldsymbol{VOL}\ (\boldsymbol{max}) = 0{,}5V$