Chapitre 2 : Algèbre de Boole

BOULAICHE Ammar

Université de Jijel

Cours Structure Machine 1 - Première Année Informatique

Septembre 2023

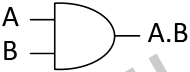
Dernière modification : 24 septembre 2023

Introduction

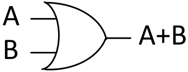
- L'algèbre de Boole est une algèbre définie sur l'ensemble des valeurs binaires (0 (faux) et 1 (vrai)).
- Sur cet ensemble binaire, on peut définir trois opérations de base : ET logique (notée $\bullet$), OU logique (notée $+$) et complémentation logique (notée^{-}).
- La combinaison de ces opérations avec des variables booléennes (variable qui prend une valeur 0 (faux) ou 1 (vrai)) permet de construire une expression appelée ≪fonction logique≫ ou ≪expression logique≫.
- Une fonction logique peut être définie par une expression algèbrique, une table de vérité ou par un schéma logique appelé ≪logigramme≫.

Fonctions logiques de base

- **La fonction ET (AND)**

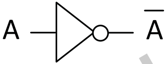
<table>
<tr><th>Expression logique</th><th>Porte logique</th><th>Table de vérité</th></tr>
<tr><td>$F(A,B) = A \cdot B$</td><td>
</td><td>
<table>
<tr><th>A</th><th>B</th><th>A.B</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
</table>

- **La fonction OU (OR)**

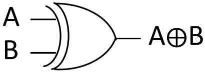
<table>
<tr><th>Expression logique</th><th>Porte logique</th><th>Table de vérité</th></tr>
<tr><td>$F(A,B) = A + B$</td><td>
</td><td>
<table>
<tr><th>A</th><th>B</th><th>A+B</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
</table>

Fonctions logiques de base

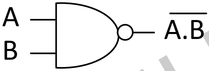
- **La fonction NON (NOT)**

<table>
<tr><th>Expression logique</th><th>Porte logique</th><th>Table de vérité</th></tr>
<tr><td>$F(A) = \overline{A}$</td><td>
</td><td><table><tr><th>A</th><th>$\overline{A}$</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
</table>

- **La fonction OU-Exclusif (XOR)**

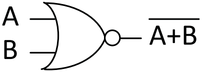
<table>
<tr><th>Expression logique</th><th>Porte logique</th><th>Table de vérité</th></tr>
<tr><td>$F(A, B) = A \oplus B$</td><td>
</td><td><table><tr><th>A</th><th>B</th><th>A⊕B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
</table>

- **La fonction NON-ET (NAND)**

Expression logique	Porte logique	Table de vérité
$F(A,B) = \overline{A \cdot B}$		(voir ci-dessous)

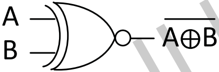
A	B	$\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

- **La fonction NON-OU (NOR)**

Expression logique	Porte logique	Table de vérité
$F(A,B) = \overline{A + B}$		(voir ci-dessous)

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

- **La fonction NON-OU-Exclusif (XNOR)**

<table>
<tr><th>Expression logique</th><th>Porte logique</th><th>Table de vérité</th></tr>
<tr>
<td>$F(A,B) = \overline{A \oplus B}$</td>
<td>
</td>
<td>
<table>
<tr><th>A</th><th>B</th><th>$\overline{A \oplus B}$</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
</table>

- Le OU-Exclusif (XOR) et le NON-OU-Exclusif (XNOR) peuvent être écrits en fonction des 3 premières opérations comme suit :

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

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

Théorèmes fondamentaux de l'algèbre de Boole

Commutativité :	$A+B=B+A$	$A.B=B.A$
Distributivité :	$A+(B.C)=(A+B).(A+C)$	$A.(B+C)=A.B+A.C$
Associativité :	$(A+B)+C=A+(B+C)$	$(A.B).C=A.(B.C)$
Idempotence :	$A+A=A$	$A.A=A$
Complémentation :	$A+\overline{A}=1$	$A.\overline{A}=0$
Involution :	$\overline{\overline{A}}=A$	
Elément neutre :	$A+0=A$	$A.1=A$
Elément dominant :	$A+1=1$	$A.0=0$

Théorèmes fondamentaux

- Absorption : $A + A.B = A$ $\quad$ $A.(A + B) = A$
- Allègement : $A + \overline{A}.B = A + B$ $\quad$ $A.(\overline{A} + B) = A.B$
- Inclusion : $(A + B).(A + \overline{B}) = A$ $\quad$ $A.B + A.\overline{B} = A$
- De Morgan : $\overline{A + B} = \overline{A}.\overline{B}$ $\quad$ $\overline{A.B} = \overline{A} + \overline{B}$

Table de vérité d'une fonction logique

- La table de vérité d'une fonction logique est un tableau décrivant les valeurs retournées par cette fonction pour toutes les combinaisons possibles de ses valeurs d'entrée. Pour n variables d'entrée, la table va donc avoir 2^n lignes (combinaisons).

Exemple : La T.V de la fonction : $F(A,B,C) = A.B + \overline{B}.C + A.\overline{C}$, est :

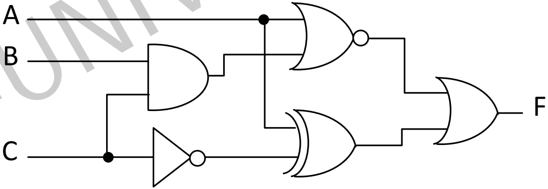
A	B	C	$\overline{B}$	$\overline{C}$	A·B	$\overline{B}$·C	A·$\overline{C}$	F(A,B,C)
0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	1	0	1
0	1	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	1	0	1
1	1	0	0	1	1	0	1	1
1	1	1	0	0	1	0	0	1

Schéma logique d'une fonction logique

- Le schéma logique (appelé aussi logigramme) d'une fonction logique est un assemblage de portes logiques qui sont reliées entre elles pour représenter graphiquement cette fonction.

Exemple : Le schéma logique de la fonction :

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



Formes canoniques d'une fonction logique

- La forme canonique d'une fonction logique est écrite en n'utilisant que des mintermes ou des maxtermes.
- Un **minterme** de *n* variables est un produit de ces *n* variables ou de leurs complémentaires.

  Exemple : Pour 4 variables (A, B, C et D) :

  $A.\overline{B}.C.D$ et $\overline{A}.\overline{B}.C.\overline{D}$ sont des mintermes.

  $\overline{A}.C.D$ et $B.\overline{C}.\overline{D}$ ne sont pas des mintermes.

- Un **maxterme** de *n* variables est un somme de ces *n* variables ou de leurs complémentaires.

  Exemple : Pour 4 variables (A, B, C et D) :

  $A+B+\overline{C}+D$ et $\overline{A}+B+\overline{C}+\overline{D}$ sont des maxtermes.

  $B+C+\overline{D}$ et $A+\overline{C}+D$ ne sont pas des maxtermes.

Formes canoniques d'une fonction logique

- La première forme canonique (1^{ere} F.C) d'une fonction logique est la forme où la fonction est exprimée en somme de mintermes. Elle est obtenue algébriquement en ajoutant les variables manquantes aux termes du somme de produits de la fonction ou à partir de la table de vérité en prenant les termes pour lesquels la fonction vaut 1.

  Exemple : La 1^{ere} F.C de la fonction : $F(A, B, C) = A.\overline{B} + \overline{B}.C$ est :

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

- La deuxième forme canonique (2^{eme} F.C) d'une fonction logique est la forme où la fonction est exprimée en produit de maxtermes. Elle est obtenue algébriquement en ajoutant les variables manquantes aux termes du produit de sommes de la fonction ou à partir de la table de vérité en prenant les termes pour lesquels la fonction vaut 0.

  Exemple : La 2^{eme} F.C de la fonction : $F(A, B, C) = A.\overline{B} + \overline{C}$ est :

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

Représentation décimale des deux formes canoniques

- La représentation décimale de la première forme canonique est obtenue en remplaçant chaque minterme de la fonction par la valeur décimale correspondant à la combinaison binaire de ses variables.

  Exemple : La représentation décimale de la fonction :

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

  $$F(A,B,C) = 101 + 100 + 001 = \sum(1,4,5).$$

- La représentation décimale de la deuxième forme canonique est obtenue en remplaçant chaque maxterme de la fonction par la valeur décimale correspondant à la combinaison binaire de ses variables.

  Exemple : La représentation décimale de la fonction :

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

  $$F(A,B,C) = (001).(011).(111) = \prod(1,3,7).$$

Simplification des fonctions logiques

- Il est toujours possible de minimiser une fonction logique afin d'avoir une fonction équivalente avec le moins de termes et le moins de variables par terme.
- La simplification d'une fonction logique permet de réduire le nombre de portes logiques et de raccordements, ce qui permet d'avoir un circuit plus petit et plus rapide avec un coût de fabrication beaucoup plus inférieur à celui du circuit correspondant à la fonction originale.
- Plusieurs méthodes peuvent être utilisées pour simplifier une fonction logique, les plus utilisées sont :
  - **La méthode algébrique :** elle est basée sur les propriétés de l'algèbre de Boole.
  - **La méthode de karnaugh :** il s'agit d'une méthode graphique basée sur des tables appelées tables de karnaugh.

Simplification par la méthode algébrique

- Le principe de la méthode algébrique consiste à appliquer les règles et les théorèmes de l'algèbre de boole afin d'éliminer des variables ou des termes de la fonction.
- Malheureusement, il n'est pas toujours facile de savoir quelle règle doit-on appliquer pour réaliser la simplification, ni de savoir dans certains cas si l'on a obtenu la forme la plus réduite ou non.
- Après chaque simplification, on cherche à déduire d'autres simplifications, jusqu'à ce que cette opération ne soit plus possible.

Exemple :
$$
\begin{aligned}
F(A,B,C,D) &= A.B.C + A.B.\overline{C} + A.\overline{B}.C.D \\
&= A.B.(C+\overline{C}) + A.\overline{B}.C \cdot D \\
&= A.B + A.\overline{B}.C.D \\
&= A.(B + \overline{B}.(C.D)) \\
&= A.(B + C.D) \\
&= A.B + A.C.D
\end{aligned}
$$

Simplification par la méthode de karnaugh

- La méthode de karnaugh est une méthode de simplification graphique basée sur une table à 2 dimensions, dont chaque dimension regroupe une ou 2 variables.
- Les cases de la table de karnaugh contiennent les valeurs retournées par la fonction logique pour toutes les combinaisons possibles de ses valeurs d'entrée.
- Pour n variables d'entrée, la table de karnaugh va avoir 2^n cases.

Exemple :

$B \backslash A$	0	1
0		
1		

2 variables

$C \backslash A.B$	00	01	11	10
0				
1				

3 variables

$C.D \backslash A.B$	00	01	11	10
00				
01				
11				
10				

4 variables

Simplification par la méthode de karnaugh

- Les lignes et les colonnes sont codées en code gray. Par conséquent, le passage d'une ligne à une ligne adjacente ou d'une colonne à une colonne adjacente change la valeur d'une seule variable.
- Les lignes extrêmes (la plus haute et la plus basse) sont également adjacentes, idem pour les colonnes extrêmes (la plus à droite et la plus à gauche).
- Les cases de la table de karnaugh sont remplies à partir de la table de vérité ou bien immédiatement à partir de la fonction logique.

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

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

Simplification par la méthode de karnaugh

- La simplification par la méthode de karnaugh se base sur le regroupement des bits à 1 adjacents en blocs rectangulaires ou carrés, tout en respectant les règles suivantes :
    1. Le nombre de 1 par groupement doit être une puissance de 2.
    2. Chaque groupement doit contenir le maximum possible de 1.
    3. L'intersection entre deux ou plusieurs groupement est autorisée.
    4. L'inclusion d'un groupement dans d'autres groupements est interdite.
    5. Une case à 1 doit appartenir à au moins un groupement.
- La fonction simplifiée est égale à la somme des termes correspondant aux blocs obtenus. Le terme d'un bloc est égal au produit des variables qui ne changent pas d'état à l'intérieur du bloc.

Exemple :

$C \backslash A.B$	00	01	11	10
0	1	0	1	1
1		1	1	1

$\overline{B}.\overline{C}$

A

$B.C$

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

Simplification par la méthode de karnaugh

- Si la table de karnaugh contient des valeurs inconues (des x), ces dernières sont considérées comme des 1 quand on fait le regroupement, sauf qu'il ne faut pas avoir des groupements qui contiennent uniquement des x ou qui contiennent des x avec des 1 déjà regroupés dans d'autres groupes.

Exemple :

$C.D$ \ $A.B$	00	01	11	10
00	0	x	0	x
01	1	1	x	1
11	1	x	x	x
10	x	1	0	x

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

Simplification par la méthode de karnaugh

- Pour une fonction à cinq variables, la table de karnaugh va se diviser en deux tables de 4 variables chacune.

$D.E \backslash B.C$	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

$A = 0$

$D.E \backslash B.C$	00	01	11	10
00	16	20	28	24
01	17	21	29	25
11	19	23	31	27
10	18	22	30	26

$A = 1$

- Deux cases occupant la même position dans les deux tables sont adjacentes (la case 5 est adjacente avec la case 21).

- Pour simplifier une fonction à cinq variables, on respecte les mêmes règles, tout en prenant en compte l'adjacence entre les deux tables.

  Exemple : la simplification de la fonction ci-dessous sera comme suit :

$$F(A,B,C,D,E) = \sum(1,2,3,6,8,9,10,11,12,14,17,18,19,\\ 20,21,22,24,25,26,27,28,29,30).$$

$A = 0$

$D.E \backslash B.C$	00	01	11	10
00	0	0	1	1
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

$A = 1$

$D.E \backslash B.C$	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	1	0	0	1
10	1	1	1	1

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

Simplification par la méthode de karnaugh

- Pour une fonction à six variables, la table de karnaugh va se diviser en 4 tables de 4 variables chacune. La case 5 est adjacente avec les cases 21 et 37, mais elle n'est pas adjacente avec la case 53 (croisement).

$E.F \backslash C.D$	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

$A.B = 00$

$E.F \backslash C.D$	00	01	11	10
00	16	20	28	24
01	17	21	29	25
11	19	23	31	27
10	18	22	30	26

$A.B = 01$

$E.F \backslash C.D$	00	01	11	10
00	32	36	44	40
01	33	37	45	41
11	35	39	47	43
10	34	38	46	42

$A.B = 10$

$E.F \backslash C.D$	00	01	11	10
00	48	52	60	56
01	49	53	61	57
11	51	55	63	59
10	50	54	62	58

$A.B = 11$

Simplification par la méthode de karnaugh

Exemple : $F(A,B,C,D,E,F) = \sum(1,5,8,9,12,13,16,17,18,21,24,25,26,29,$
$33,35,37,39,40,41,44,45,48,49,50,51,53,55,56,57,58,61).$

E.F \ C.D	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

$A.B = 00$

E.F \ C.D	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	0	0	0	0
10	1	0	0	1

$A.B = 01$

E.F \ C.D	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	1	1	0	0
10	0	0	0	0

$A.B = 10$

E.F \ C.D	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	0	0
10	1	0	0	1

$A.B = 11$

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

Références

Wladimir Mercouroff.
Architecture matérielle et logicielle des ordinateurs et des microprocesseurs.
Armand Colin, 1990. ISBN : 2-200-42007-2.
Bibliothèque centrale, Université de Jijel, Cote :002/07.

Joel Ristori, Lucien Ungaro.
Cours d'architecture des ordinateurs.
Eyrolles, 1991.
Bibliothèque centrale, Université de Jijel, Cote :002/17.

Pierre-Alain Goupille.
Technologie des ordinateurs pour les I.U.T et B.T.S informatique avec exercices.
Masson, 1993.
Bibliothèque centrale, Université de Jijel, Cote :002/01.

Références

André Poinsot.
Problèmes d'électronique logique.
Masson , 1994. ISSN : 2225844518.
Bibliothèque centrale, Université de Jijel, Cote :621/270.

Paolo Zanella, Yves Ligier.
Architecture et technologie des ordinateurs.
Dunod, 1998. ISBN : 210003801X.
Bibliothèque centrale, Université de Jijel, Cote :004/258.

Habiba Drias-Zerkaoui.
Introduction à l'architecture des ordinateurs.
Office des publications universitaires, 2006.
Bibliothèque centrale, Université de Jijel, Cote :002/08.