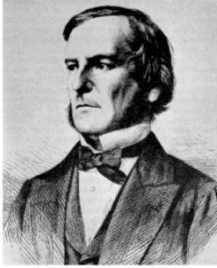


Chapitre E3. Logique combinatoire – Portes logiques



Georges Boole (1815 - 1864) est un logicien, mathématicien et philosophe britannique. En 1847 et en 1854, il publie deux travaux où il y développe une nouvelle forme de logique, à la fois symbolique et mathématique. Il crée une algèbre binaire n'acceptant que deux valeurs numériques : 0 et 1. Les travaux théoriques de Boole trouveront des applications primordiales dans des divers domaines (informatique, probabilités, circuits électriques et téléphoniques...) grâce à d'autres scientifiques (Russell, Turing, Shannon...).

INTRO :

En MP2I et dans le ChE1, l'information à transmettre et à traiter est représentée par une tension variable au cours du temps, qui peut prendre toute valeur dans une plage considérée : c'est le domaine de l'électronique analogique.

L'électronique logique consiste à coder l'information en n'utilisant que **deux états électriques nettement séparables** : **Haut** (par exemple une tension supérieure à 5 V) et **Bas** (par exemple une tension inférieure à 0,5 V), qui **représentent les chiffres 1 et 0**, correspondant à un **bit** (Binary digiT : chiffre binaire).

Cette manière de transmettre l'information présente une **grande immunité** aux parasites (bruit, déformation du signal) puisqu'il suffit de reconnaître, non pas la forme du signal mais des niveaux électriques nettement séparés.

Dans ce chapitre, on s'intéresse à la **LOGIQUE COMBINATOIRE** : il s'agit de l'ensemble des **opérations logiques pour lesquelles l'état de sortie ne dépend que de l'état des variables d'entrée**. Ces opérations (ou fonctions) logiques vérifient les relations de l'**algèbre de Boole** et sont réalisées en pratique par des **PORTES LOGIQUES**.

Buts de ce chapitre : Présenter les portes logiques usuelles ; Construire et/ou exploiter des tables de vérité.

Plan du chapitre :

A) Signal logique.....	1
B) Les portes logiques fondamentales	2
1) Description générale des portes logiques.....	2
2) Opérateur à une seule entrée : porte NOT	3
3) Opérateurs à plusieurs entrées : porte AND et OR.....	3
4) Opérateurs inverseurs à plusieurs entrées : portes NAND et NOR.....	3
C) Propriétés générales des opérateurs	4
1) L'algèbre de Boole	4
2) Porte XOR	4
3) Lois de de Morgan	4

A) Signal logique

Un **signal logique** Y ne peut prendre que deux valeurs, traditionnellement représentées par 0 (niveau bas) et 1 (niveau haut). Concrètement, un signal logique Y est associé à une **tension** u_Y entre un nœud d'un circuit et la masse.

♦ **Idéalement**, $u_Y = 0 \Leftrightarrow Y = 0$ et $u_Y = V_{dd} > 0 \Leftrightarrow Y = 1$.

La valeur de V_{dd} dépend du montage et de la technologie, on peut avoir $V_{dd} = 5\text{ V}$ par exemple.

♦ **En pratique**, il existe deux **tensions de seuil** V_{sb} et V_{sh} , avec $0 < V_{sb} < V_{sh} < V_{dd}$, telles que $u_Y < V_{sb}$ correspond à $Y = 0$ et $u_Y > V_{sh}$ correspond à $Y = 1$.

On supposera que $V_{sb} = V_{sh} = \frac{V_{dd}}{2}$ (hypothèse proche de la réalité, cf TP) ainsi :

en pratique, $u_Y < \frac{V_{dd}}{2} \Leftrightarrow Y = 0$ et $u_Y > \frac{V_{dd}}{2} \Leftrightarrow Y = 1$.

Dans ce chapitre, on considèrera que les tensions ne prennent que les deux valeurs 0 ou V_{dd} et on négligera le temps de commutation* des composants électroniques. On tiendra compte de la tension seuil et du temps de commutation au ChE4.

* temps de propagation des éventuels changements des entrées vers les sorties.

B) Les portes logiques fondamentales

1) Description générale des portes logiques

Les portes logiques électroniques sont mises en œuvre à l'aide de blocs semi-conducteurs (diodes ou transistors) qui fonctionnent comme des **interrupteurs commandés en tension**.

Les portes logiques sont représentées par des **symboles**. Deux normes sont en vigueur :

- la norme ANSI (American National Standards Institute), utilisée en Amérique du Nord ;
- la norme IEC (International Electrotechnical Commission), utilisée en Europe.

Porte logique	Symbole IEC	Symbole ANSI	Fonction logique	Table de vérité															
NOT			$S = \bar{a}$	<table border="1"> <thead> <tr> <th>a</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td></td> </tr> </tbody> </table>	a	S	0		1										
a	S																		
0																			
1																			
AND			$S = a \cdot b$	<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table>	a	b	S	0	0		0	1		1	0		1	1	
a	b	S																	
0	0																		
0	1																		
1	0																		
1	1																		
OR			$S = a + b$	<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table>	a	b	S	0	0		0	1		1	0		1	1	
a	b	S																	
0	0																		
0	1																		
1	0																		
1	1																		
NAND			$S = \overline{a \cdot b} =$	<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table>	a	b	S	0	0		0	1		1	0		1	1	
a	b	S																	
0	0																		
0	1																		
1	0																		
1	1																		
NOR			$S = \overline{a + b} =$	<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table>	a	b	S	0	0		0	1		1	0		1	1	
a	b	S																	
0	0																		
0	1																		
1	0																		
1	1																		
XOR			$S = a \oplus b$	<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table>	a	b	S	0	0		0	1		1	0		1	1	
a	b	S																	
0	0																		
0	1																		
1	0																		
1	1																		

Rq : Pour la norme IEC, la négation peut également être notée $\bar{\ }_P$ au lieu de \bar{P} .

On caractérise l'opération (ou la fonction) logique réalisée par une porte logique via sa **TABLE DE VERITE**. Il s'agit d'un **tableau** comportant plusieurs **colonnes** : celles des **variables d'entrée** et celle de la **variable de sortie**.

Les valeurs des cellules de ce tableau sont **1** (ou V pour vrai / activé) ou **0** (ou F pour faux / désactivé).

La table de vérité donne **l'état de la variable de sortie pour chacune des combinaisons des variables d'entrée**. Ainsi, s'il y a n variables d'entrée, la table comporte 2^n combinaisons différentes d'entrées.

2) Opérateur à une seule entrée : porte NOT

L'opérateur NOT (NON en français) agit sur une seule variable d'entrée a , il est aussi appelé inverseur logique. **La porte renvoie « l'inverse » de la variable d'entrée, noté \bar{a} .**

3) Opérateurs à plusieurs entrées : porte AND et OR

◆ L'opérateur AND (ET en français) agit sur au moins deux variables d'entrée.

Une porte AND à deux entrées a et b renvoie 1 uniquement si les deux entrées a et b valent 1 sinon elle renvoie 0. On note sa sortie $a.b$.

Une porte AND à plus que deux entrées renvoie 1 uniquement si toutes les entrées valent 1.

◆ L'opérateur OR (OU en français) agit sur au moins deux variables d'entrée.

Une porte OR à deux entrées a et b renvoie 0 uniquement si les deux entrées a et b valent 0 sinon elle renvoie 1. On note sa sortie $a+b$.

Une porte OR à plus que deux entrées renvoie 0 uniquement si toutes les entrées valent 0.

4) Opérateurs inverseurs à plusieurs entrées : portes NAND et NOR

L'opérateur inverseur permet de combiner directement une porte avec la fonction NOT. La représentation est celle de la porte originale à laquelle on rajoute le symbole du NOT.

◆ L'opérateur NAND (NON ET en français) agit sur au moins deux variables d'entrée.

Une porte NAND à deux entrées a et b renvoie 0 uniquement si les deux entrées a et b valent 1 sinon elle renvoie 1. On note sa sortie $\overline{a.b}$.

◆ L'opérateur NOR (NON OU en français) agit sur au moins deux variables d'entrée.

Une porte NOR à deux entrées a et b renvoie 1 uniquement si les deux entrées a et b valent 0 sinon elle renvoie 0. On note sa sortie $\overline{a + b}$.

Rq : Lors du développement de l'électronique logique, il s'avérait plus facile et plus économique de fabriquer des portes inverseuses que des portes non inverseuses. De plus, le fait de travailler essentiellement avec des portes inverseuses permettait aux industriels d'avoir un stock moins important de composants différents. En effet, les portes NAND et NOR sont des fonctions complètes (ou universelles) i.e. qu'il est possible de réaliser toutes les fonctions logiques seulement avec des portes NAND (ou seulement avec des portes NOR), cf § B.3.

C) Propriétés générales des opérateurs

1) L'algèbre de Boole

L'algèbre de Boole est constituée de deux éléments : 0 et 1.

On peut utiliser trois opérations : la négation, l'addition et la multiplication telles que définies par les opérateurs NOT, OR et AND.

Pour montrer une égalité dans l'algèbre de Boole, il suffit de construire les tables de vérité des deux termes et de constater l'égalité.

⇒ Justifier les égalités suivantes $a + 0 = a$, $a + 1 = 1$, $a \cdot 0 = 0$ et $a \cdot 1 = a$.

Rq : 1 est neutre pour \cdot et 0 est neutre pour $+$.

On peut montrer que les opérations \cdot et $+$ sont commutatives (i), associatives (ii) et distributives (iii).

- (i) $a + b = b + a$ et $a \cdot b = b \cdot a$
(ii) $a + (b + c) = (a + b) + c$ et $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
(iii) $a + (b \cdot c) = (a + b) \cdot (a + c)$ et $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

2) Porte XOR

L'opérateur XOR (OU EXCLUSIF en français) à deux entrées a et b renvoie 1 si l'une ou l'autre entrée (mais pas les deux) vaut 1 sinon elle renvoie 0. On note sa sortie $a \oplus b$.

L'opération XOR n'est pas une opération fondamentale : il est possible de la construire par combinaison des autres opérations.

⇒ Démonstration : Justifier l'égalité suivante $a \oplus b = \bar{b} \cdot a + b \cdot \bar{a}$.

Rq : on définit aussi l'opérateur XNOR comme l'opérateur inverseur de XOR, sa sortie est notée $\overline{a \oplus b}$.

3) Lois de de Morgan

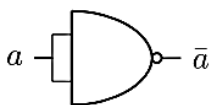
Les lois de de Morgan établissent le lien entre les trois opérateurs fondamentaux :

$$\overline{a+b} = \bar{a} \cdot \bar{b} \quad \text{et} \quad \overline{a \cdot b} = \bar{a} + \bar{b}$$

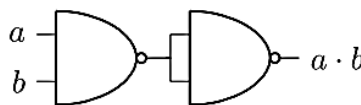
⇒ Démonstration : Justifier les lois de de Morgan.

À partir des lois de de Morgan, on montre que **les fonctions des portes NAND et NOR sont des fonctions complètes (ou universelles) i.e. qu'il est possible de réaliser toutes les fonctions logiques seulement avec des portes NAND (ou seulement avec des portes NOR).**

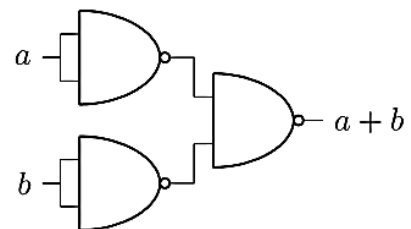
⇒ Montrer que les associations de portes NAND ci-dessous permettent d'obtenir les opérations NOT, AND et OR.



(a) L'opérateur NOT



(b) L'opérateur AND



(c) L'opérateur OR

Cf <https://ressources.univ-lemans.fr/AccesLibre/UM/Pedago/physique/02/electro/nands.html>