



CIRCUITS LOGIQUES COMBINATOIRES

MAMADOU LAMINE NDIAYE, ESP DAKAR

Document en cours de relecture, version du 2016/01/25

ÉLÉMENTS DE BASE DES SYSTÈMES LOGIQUES

Dans les circuits logiques, le codage des informations utilise deux niveaux de tension et un état logique est associé à chaque niveau de tension. Les systèmes numériques (logiques) fonctionnent en binaire c'est à dire que les variables discrètes (entrées ou sorties) du système ne prennent que deux valeurs *0* ou *1*. L'état d'une variable logique peut être *faux* ou *vrai*. On représente l'état faux par *0* et l'état vrai par *1*.

Par exemple, pour la technologie électrique TTL, le *0* logique correspond à une tension entre 0 et 0,8V, alors que le *1* logique correspond à une tension entre 2,4 et 5V.

Si nous prenons l'exemple d'un moteur électrique, le *0* logique peut correspondre à l'état *Arrêt*, alors que le *1* logique peut correspondre à l'état *Marche*.

ALGÈBRE DE BOOLE

La logique binaire peut être représentée par l'Algèbre de BOOLE, qui permet de décrire dans un modèle mathématique le traitement et le fonctionnement des systèmes binaires. L'Algèbre de BOOLE est conçue autour d'opérations logiques de base :

- le complément logique : NON, représenté par un surligné
- l'addition logique : OU, représenté par le signe $+$
- produit logique : ET, représenté par le signe \bullet
- et leurs dérivés (XOR, NAND, NOR, etc)

DRAFT

PORTES LOGIQUES

Les éléments de base des systèmes logiques sont des portes logiques que l'on peut assembler pour réaliser des fonctions logiques qui peuvent à leur tour être assemblées pour construire des machines numériques.

La seule porte intéressante à une entrée est l'inverseur :

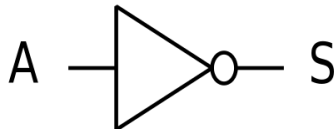
Fonction	Symbole	Equation	Table de vérité						
NON		$S = \overline{A}$	<table><tr><th>A</th><th>S</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	S	0	1	1	0
A	S								
0	1								
1	0								

Figure : Inverseur

La table de vérité qui est un tableau qui indique la valeur de la sortie pour toutes les combinaisons des entrées. Chaque ligne correspond à une combinaison des variables. Ici, elle n'a que deux lignes.

Les deux portes de base à 2 entrées sont la porte OU et la porte ET :

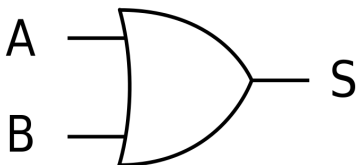
Fonction	Symbole	Equation	Table de vérité															
OU		$S = A + B$	<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1
A	B	S																
0	0	0																
0	1	1																
1	0	1																
1	1	1																

Figure : Porte OU

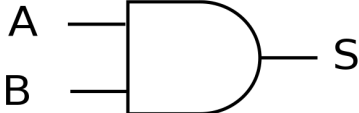
Fonction	Symbole	Equation	Table de vérité															
ET		$S = A \cdot B$	<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1
A	B	S																
0	0	0																
0	1	0																
1	0	0																
1	1	1																

Figure : Porte ET

Les tables de vérité ont alors 4 lignes.

Les autres portes logiques à deux entrées sont des dérivées des éléments de base troisième élément de base :

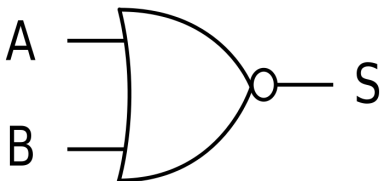
Fonction	Symbole	Equation	Table de vérité															
NOR		$S = \overline{A + B}$	<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	0
A	B	S																
0	0	1																
0	1	0																
1	0	0																
1	1	0																

Figure : Porte NOR

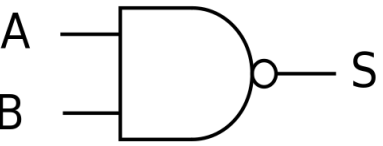
Fonction	Symbole	Equation	Table de vérité															
NAND		$S = \overline{A \cdot B}$	<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	1	0	1	1	1	0	1	1	1	0
A	B	S																
0	0	1																
0	1	1																
1	0	1																
1	1	0																

Figure : Porte NAND

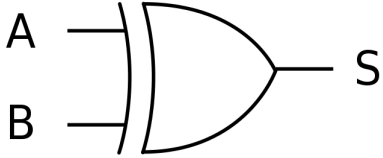
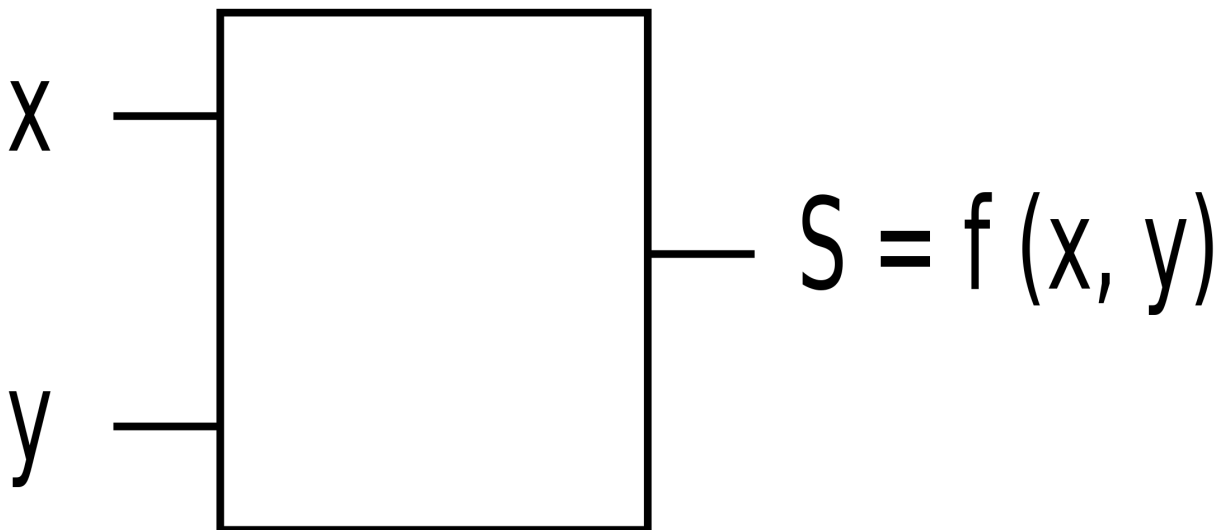
Fonction	Symbole	Equation	Table de vérité															
XOR		$S = A \oplus B$	<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	0
A	B	S																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

Figure : Porte Ou exclusif

EXPRESSION MATHÉMATIQUE D'UNE FONCTION LOGIQUE

Tout système logique peut être défini à l'aide d'une fonction logique (ou expression logique) qui représente les relations entre les variables de sortie en fonction des variables d'entrée.

*Figure : Fonction logique*

La représentation des fonctions logiques se fait:

- de manière algébrique (représentation mathématique), déjà utilisé la représentation algébrique avec les équations logiques. Exemple :

- de manière tabulaire (table de vérité), une table de vérité est un tableau à 2^N lignes et $N+1$ colonnes. Chaque ligne correspond à une combinaison des N variables. Les N premières colonnes contiennent les variables et la colonne $N+1$ contient la valeur de la fonction
- de manière graphique avec un logigramme qui est un schéma illustrant l'expression d'une fonction logique avec les portes logiques (Exemple du schéma de la porte XOR).
- avec des diagrammes, plusieurs diagrammes existent, nous présenterons celui de Karnaugh lors de la leçon sur la synthèse des circuits logiques combinatoires.

L'expression en algèbre de Boole d'une fonction logique peut être faite sous deux formes : la forme *somme de produits* et la forme *produit de sommes*. L'expression sous la forme somme de produit est obtenue à partir de la somme de tous les lignes de la table de vérité où la fonction vaut 1. On appelle ces lignes *mintermes*.

A	B	C	S	Minterme	Maxterme	
0	0	0	0	$\bar{A} \cdot \bar{B} \cdot \bar{C} = 0$	$A+B+C = 0$	<i>Somme de produits :</i>
0	0	1	0	$\bar{A} \cdot \bar{B} \cdot C = 0$	$A+B+\bar{C} = 0$	
0	1	0	1	$\bar{A} \cdot B \cdot \bar{C} = 1$	$A+\bar{B}+C = 1$	
0	1	1	1	$\bar{A} \cdot B \cdot C = 1$	$A+\bar{B}+\bar{C} = 1$	
1	0	0	0	$A \cdot \bar{B} \cdot \bar{C} = 0$	$\bar{A}+B+C = 0$	<i>Produit de somme :</i>
1	0	1	1	$A \cdot \bar{B} \cdot C = 1$	$\bar{A}+B+\bar{C} = 1$	
1	1	0	1	$A \cdot B \cdot \bar{C} = 1$	$\bar{A}+\bar{B}+C = 1$	
1	1	1	0	$A \cdot B \cdot C = 0$	$\bar{A}+\bar{B}+\bar{C} = 0$	

$$S = \bar{A} \cdot B \cdot \bar{C} + \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C}$$

$$S = (A+B+C) \cdot (A+B+\bar{C}) \cdot (\bar{A}+B+C) \cdot (\bar{A}+\bar{B}+\bar{C})$$

Figure : Fonction à 3 entrées

L'expression sous la forme *produit de sommes* est obtenue en multipliant tous les *maxtermes* pour lesquels la fonction vaut 0.

PROPRIÉTÉS DE L'ALGÈBRE DE BOOLE

Il est souvent utile de simplifier ces expressions pour réduire le nombre de portes logiques et par conséquent le coût du matériel ou le temps de câblage. On utilise alors les propriétés de l'Algèbre de BOOLE.

Commutativité :

$$\backslash(\backslashmathrm{A}\bullet B=B\bullet A)\backslash)$$

$$\backslash(\backslashmathrm{A+B=B+A})\backslash)$$

Idempotence :

$$\backslash(\backslashmathrm{A}\bullet A=A)\backslash)$$

$$\backslash(\backslashmathrm{A+A=A})\backslash)$$

Constantes :

$$\backslash(\backslashmathrm{A}\bullet 0=0)\backslash)$$

$$\backslash(\backslashmathrm{A}\bullet 1=A)\backslash)$$

$$\backslash(\backslashmathrm{A+0=A})\backslash)$$

$$\backslash(\backslashmathrm{A+1=1})\backslash)$$

Complémentation :

$$\backslash(\backslashmathrm{A}\bullet \overline{A}=0)\backslash)$$

$$\backslash(\backslashmathrm{A+\overline{A}=1})\backslash)$$

Distributivité :

$$\backslash(\backslashmathrm{A}\bullet (B+C)=(A\bullet B)+(A\bullet C))\backslash)$$

$$\backslash(\backslashmathrm{A+(B\bullet C)=(A+B)\bullet (A+C)})\backslash)$$

Associativité :

$$\backslash(\backslashmathrm{A}\bullet (B\bullet C)=(A\bullet B)\bullet C=A\bullet B\bullet C)\backslash)$$

$$\backslash(\backslashmathrm{A+(B+C)=(A+B)+C=A+B+C})\backslash)$$

De Morgan :

$$\backslash(\backslashmathrm{\overline{A}\bullet B}=\overline{A}+\overline{B})\backslash)$$

$$\backslash(\backslashmathrm{\overline{A+B}}=\overline{A}\bullet \overline{B})\backslash)$$

Ces propriétés sont utilisées pour la simplification des fonctions logiques. Elles permettent d'obtenir une expression logique comportant un nombre minimal de termes, ainsi qu'un nombre minimal de variables dans chaque terme dans le but de simplifier la réalisation matérielle.