

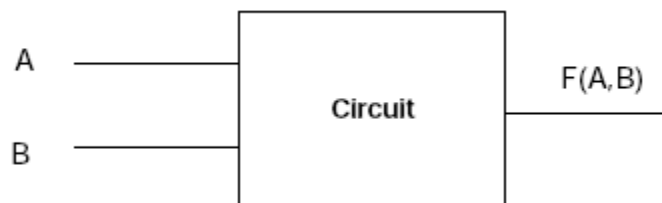
2. Algèbre binaire (algèbre de Boole)

2.1. Introduction

En algèbre de Boole le raisonnement se fait sous forme binaire, c-à-d la chose est ou pas (existe ou n'existe pas).

Cette façon de raisonnement est parfaitement adaptée pour l'étude des systèmes automatisés ou les appareils et les circuits ne peuvent prendre que deux états (le zéro (le faux), ou le 1 (le vrai)).

Les machines numériques sont constituées d'un ensemble de circuits électroniques. Chaque circuit fournit une fonction logique bien déterminée (addition, comparaison,...).



La fonction $F(A,B)$ peut être : la somme de A et B , ou le résultat de la comparaison de A et B ou une autre fonction. Pour concevoir et réaliser ce circuit on doit avoir un modèle mathématique de la fonction réalisée par ce circuit. Ce modèle doit prendre en considération le système binaire. Le modèle mathématique utilisé est celui de Boole.

2.2. Historique

L'algèbre binaire résulte des travaux de mathématicien George Boole qui a développé au 19^{ème} siècle une algèbre qui ne peuvent prendre qu'un nombre fini d'état.

Son application est limitée aux variables et fonctions à caractère binaire est attribuée à Claude Shannon.

2.3. Définition

C'est l'algèbre des grandeurs qui ne peuvent prendre que deux valeurs, 0 ou 1. L'algèbre de Boole est bien adaptée à l'étude des systèmes qui ne peuvent occuper que deux états. Plusieurs valeurs logiques peuvent être combinées pour donner un résultat qui est lui aussi une valeur logique. Exemple : (arrêt marche, ouvert fermé, enclenché déclenché, avant arrière, vrai faux, conduction blocage). Un interrupteur est ouvert ou fermé. Une lampe est allumée ou éteinte Une tension est élevée ou faible Une pression est présente ou pas.

2.4. Variables et fonctions logiques

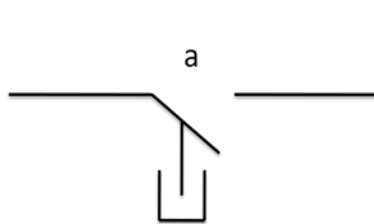
2.4.1. Variables logiques (variables booléennes)

Une variable logique est une grandeur qui ne peut prendre que deux valeurs 0 ou 1. Si cette dernière est vrai alors la variable prend la valeur 1, si non 0. Ils sont décomposés en deux types :

- **Variables d'entrées** : sont celles sur lesquelles on peut agir directement.
- **Variables de sorties** : contenant l'état de la fonction après l'évaluation des opérateurs logiques sur les variables d'entrée.

Exemple : état d'un interrupteur, d'un bouton poussoir, la présence d'une tension ...

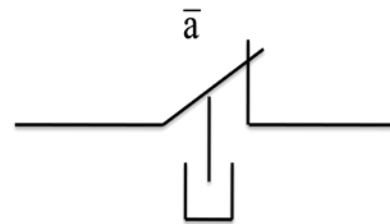
Soit e est la variable associée à l'état d'un interrupteur si $e=0$ cela signifie que le contact est ouvert et si $e=1$ cela signifie que le contact est fermé.



Contact ouvert

e	S
0	0
1	1

$$S=e$$



Contact fermé

e	S
0	1
1	0

$$S = \bar{e}$$

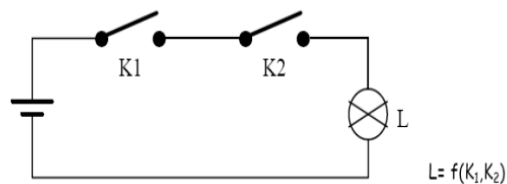
Remarque : une variable d'entrée doit être représentée en minuscule !

2.4.2. Fonction logique

Une fonction logique est le résultat de la combinaison d'une ou plusieurs variables logiques reliées entre elles par des opérations mathématiques BOOLEENNES bien définies: la valeur résultante de cette fonction dépend de la valeur des variables logiques, mais de toute façon cette résultante ne peut être que **0** ou **1**. Une fonction logique possède donc une ou des variables logiques d'entrée et une variable logique de sortie.

Exemple : la lampe L s'allume lorsque les interrupteurs k_1 et k_2 sont fermés tous les deux. Elle est éteinte dans tous les autres cas.

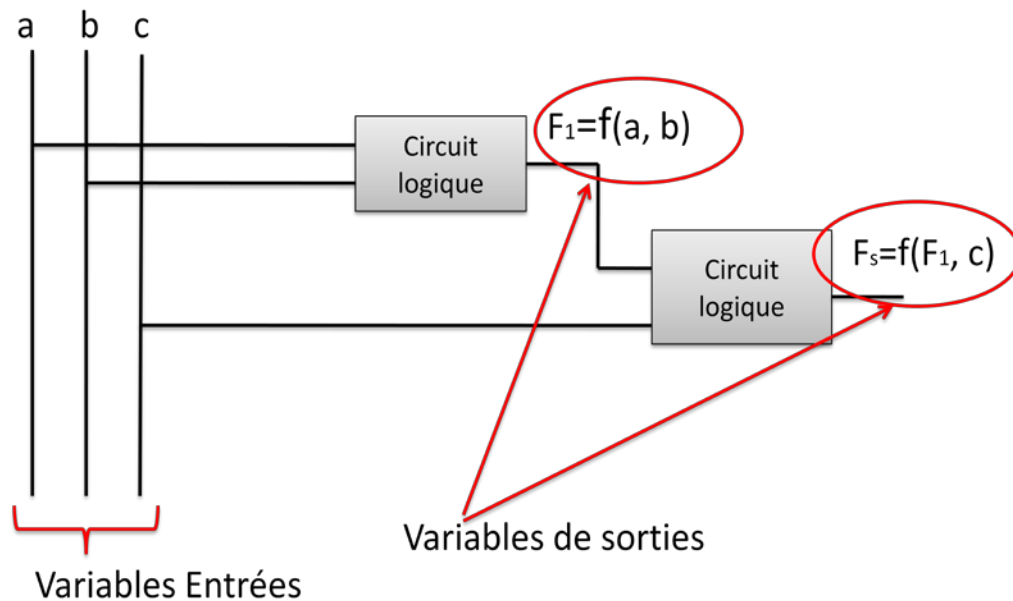
K1	K2	L
0	0	0
0	1	0
1	0	0
1	1	1



Fonction logique

Le doublet des deux variables (k_1, k_2) peut prendre 2^2 valeurs distinctes. Dans le cas général de n variables, il y aura 2^n configurations possibles. Les variables k_1 et k_2 sont dites les entrées du système. L est dit la sortie du système.

Le fonctionnement d'un système logique est décrit par une ou plusieurs propositions logiques qui présente un caractère binaire : vrai ou faux (0 ou 1). La relation entrées-sorties est décrit par une ou plusieurs fonctions logiques.



2.5.Code Gray (binaire réfléchi)

En code binaire pur, le passage d'un nombre au nombre suivant se traduit par un changement de un ou plusieurs bits. Le passage de 3 à 4 en décimal donne en binaire 011 à 100. Les trois bits changent simultanément. Cela présente parfois des inconvénients, notamment dans le cas des capteurs - codeurs (de position par exemple), où l'utilisation du code binaire pur peut amener la lecture de codes erronés au moment des transitions d'une position à la suivante. Pour éviter cet inconvénient, on utilise le code GRAY qui est un code binaire réfléchi dans lequel un seul bit change d'état au changement d'un nombre au nombre suivant.

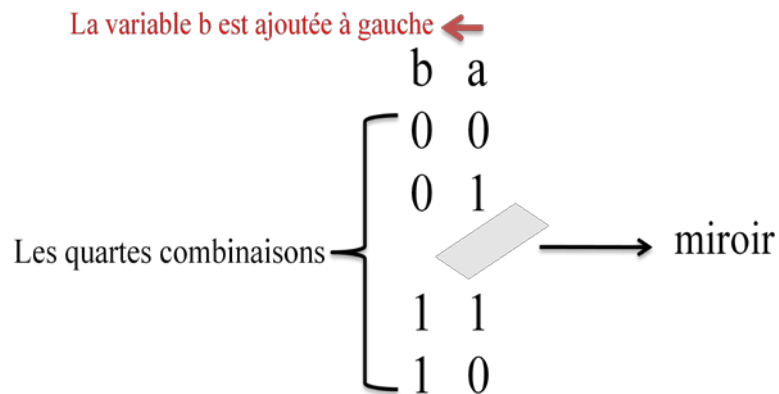
- **Cas d'une variable**

On considère la variable logique **a** d'après la loi pour trouver le nombre des combinaisons pour ce variable on trouve $2^1=2$, donc le variable **a** peut prendre que deux variable le 0 ou le 1.

a
0
1

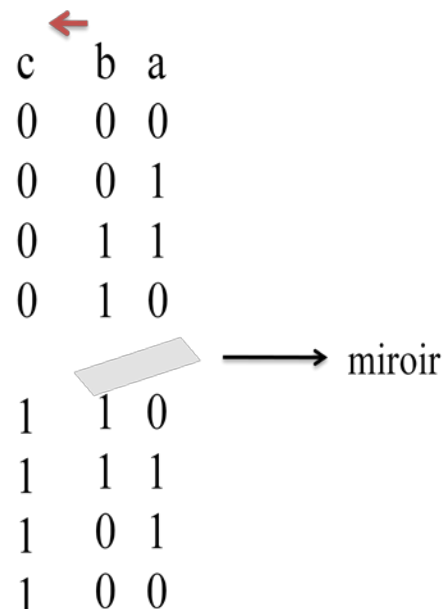
- **Cas de deux variables**

Lorsque on ajoute une variable **b**, on aura deux variable ce qui signifie $2^2=4$ combinaisons, dans ce cas la variable **b** doit être ajoutée dans la partie gauche et on ajoute un miroir au niveau de la variable **a**.



- **Cas de trois variables**

On fait la même chose avec trois variables, on ajoute une variable **c** à gauche après la variable **b** puis, on calcul le nombre de combinaisons possibles avec 3 variables, en appliquant la règle donc on doit y avoir 8 combinaisons 2^3 . Le miroir est appliqué dans ce cas pour les variables **a** et **b**.



On fera la même chose pour un nombre croissant de variables !

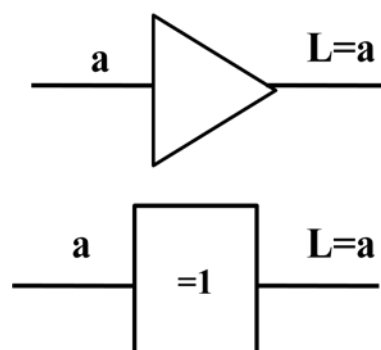
2.6. Les fonctions logiques (opérateurs logiques)

Les fonctions logiques sont conçues à partir d'un groupe d'opérateurs élémentaires appelés (portes). Chaque opérateur est représenté par un symbole et sa fonction est définie par une table de vérité.

- **La fonction OUI**

L'état de la sortie est égale à l'état de l'entrée, cette fonction ne présente par d'intérêt d'un point de vue logique mais peut être utile d'un point de vue technologique.

a	L
---	---

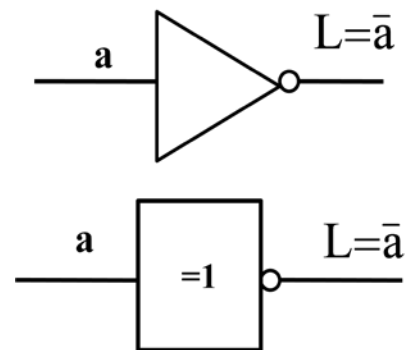


0	0
1	1

- **Porte NON (NOT)**

L'état logique de la sortie est le **complément** de celui de l'entrée

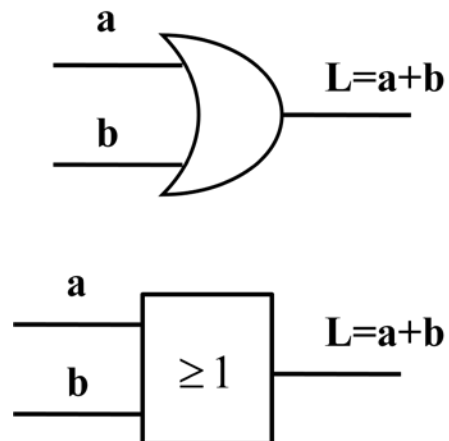
a	L
0	1
1	0



- **La fonction OU**

La sortie est à l'état 1 si au moins une des entrées est à l'état 1.

a	b	L
0	0	0
0	1	1
1	0	1
1	1	1

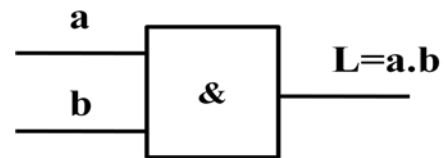
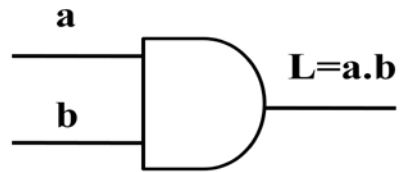


- **La fonction ET**

La sortie est à l'état 1 si les deux entrées sont simultanément à l'état 1.

a	B	L
0	0	0
0	1	0

1	0	0
1	1	1



2.6.Relation fondamentales

La somme	Le produit
$a+a=a$	$a.a=a$
$a+1=1$	$a.1=a$
$a+0=a$	$a.0=0$
$a+\bar{a}=1$	$a.\bar{a}=0$

2.7.Théorème de DeMorgan

Théorème 1

Le complément d'un produit est égal la somme des compléments de chaque variable.

$$\overline{a.b} = \bar{a} + \bar{b}$$

Théorème 2

Le complément d'une somme est égal le produit des compléments de chaque variable.

$$\overline{a + b} = \bar{a}.\bar{b}$$