

## 1. NOTION LOGIQUE

### PROBLEMATIQUE AU NIVEAU DU TRAITEMENT DE L'INFORMATION :

Une des fonctions principales de la chaîne d'informations d'un système technique est de « **traiter les informations** »

Cette fonction « traiter les informations » est souvent réalisée par un circuit électronique dont les informations traitées sont du type **logique ou binaire ou TOR (Tout ou rien)**.

Le problème consiste :

- Soit à concevoir ce circuit à partir d'un cahier des charges de fonctionnement,
- Soit à analyser ce circuit si le système est déjà existant.

Dans les deux cas, il sera nécessaire :

- D'élaborer les équations des variables de sorties en fonction des variables d'entrée  $\Rightarrow$  **règles de base de la logique**
- D'établir le schéma complet du circuit de commande à partir de ces équations  $\Rightarrow$  **opérateurs logiques, logigramme**
- De réaliser (si besoin est) le circuit :

Nous allons ainsi nous intéresser à l'application des règles de l'algèbre de Boole. C'est l'algèbre utilisée par les calculateurs, les ordinateurs, les automates programmables industriels (API), pour « **traiter les informations** »

### VARIABLES LOGIQUES (OU BINAIRES)

#### Définition :

Cette logique repose sur une variable logique ou binaire (bit, de la contraction des mots anglais Binary digIT, élément binaire), ne pouvant prendre que deux valeurs : 2 états différents et distincts. Exemple :

- Ouvert / Fermé
- Tout / Rien
- Actionné / Non actionné.

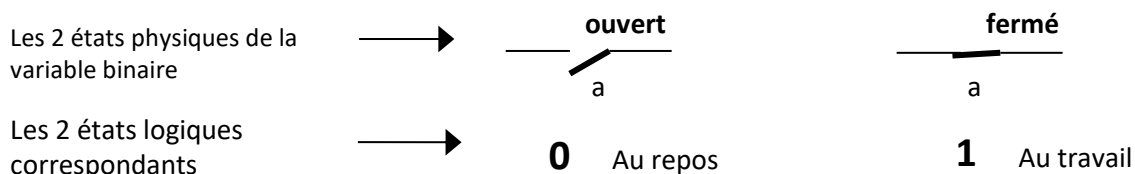
Par convention ces 2 états sont notés : **0 et 1**

#### Variable d'entrée :

##### Exemple 1 : le Contact à fermeture ou **Contact ouvert au repos (NO)**

C'est un contact qui est normalement ouvert (Normally Open) au repos. Il se ferme lorsqu'il est actionné. On désigne ce type de contact par des lettres minuscules a, b, c...

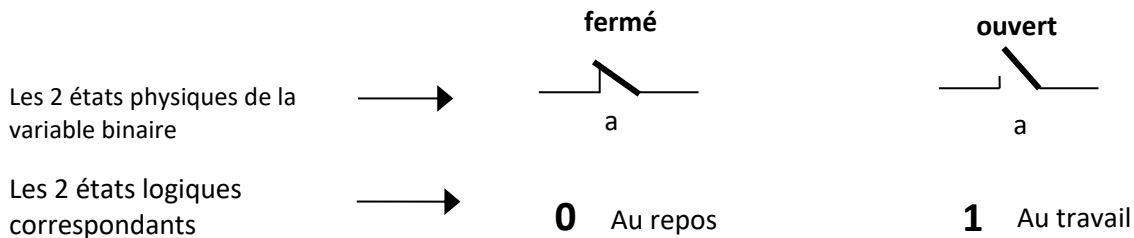
Il ne peut présenter que 2 états physiques différents : ouvert ou fermé. A chacun de ces 2 états distincts, on fait correspondre les 2 valeurs logiques : 0 et 1



### Exemple 2 : le contact à ouverture ou **Contact fermé au repos (NF ou NC)**

C'est un contact qui est normalement fermé (Normally Closed) au repos et qui s'ouvre lorsqu'il est actionné. On désigne ce type de contact par des lettres  $\bar{a}$ ,  $\bar{u}$ ,  $\bar{i}$ , ( $\bar{a}$  se lit a barre).

Il ne peut présenter que 2 états physiques différents : ouvert ou fermé. A chacun de ces 2 états distincts, on fait correspondre les 2 valeurs logiques : 0 et 1



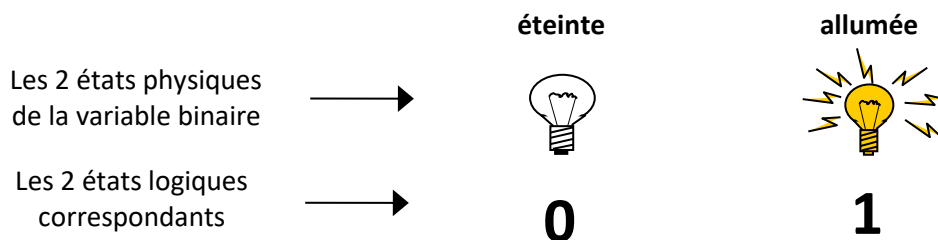
### Conclusion :

Pour un contact :

- **L'absence** d'action intervenant sur lui représente l'état logique **0**.
- **La présence** d'une action intervenant sur lui représente l'état logique **1**.

### Variable de sortie :

**Exemple d'une lampe :** Celle-ci ne peut présenter que 2 états : allumé ou éteinte. A chacun de ces 2 états distincts on fait correspondre les 2 valeurs logiques : **0** et **1**



### Conséquence :

Les actionneurs tels que : moteurs, vérins sont à l'état 0 lorsqu'ils ne sont pas alimentés. Le circuit est alors ouvert.

- Pour un circuit pneumatique, l'état 0 correspond à une absence de pression.
- Pour un circuit électrique, cela correspond à une absence de différence de potentiel entre les bornes du circuit.

Les actionneurs sont à l'état 1 lorsqu'ils sont alimentés.

- Pour un circuit pneumatique ou hydraulique ceci correspond à une pression d'air ou d'huile dans le circuit.
- Pour un circuit électrique cela correspond à une différence de potentiel entre les bornes du circuit.

**Un circuit électrique, pneumatique, hydraulique peut ainsi prendre 2 états logiques « 1 » ou « 0 ». Ces états dépendent de l'état des composants en série dans le circuit.**

## DIFFERENTES REPRESENTATIONS EN LOGIQUE BINAIRE

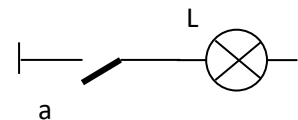
Ainsi, l'état du récepteur en logique binaire peut se représenter de plusieurs manières différentes par :

- Représentation électrique (schéma électrique),
- Représentation algébrique (équation logique ou équation booléenne),
- Représentation arithmétique (table de vérité),
- Représentation temporelle (chronogramme),
- Représentation logique (logigramme avec opérateurs logiques élémentaires),
- Représentation en langage à contact.

### Représentation par schéma électrique :

Soit un circuit possédant un contact « a » associé en série avec une lampe L.

- Lorsque le contact « a » est à l'état 0, la lampe n'est pas allumée.
- Lorsque le contact « a » est à l'état 1, la lampe est allumée.



### Représentation par équation logique ou équation booléenne :

On dit que l'état du récepteur est fonction de l'état 0 ou 1 du contact « a », ce qui peut s'écrire sous forme d'équation booléenne (Georges Boole logicien et mathématicien anglais) par :

$$L = a$$

### Représentation par table de vérité :

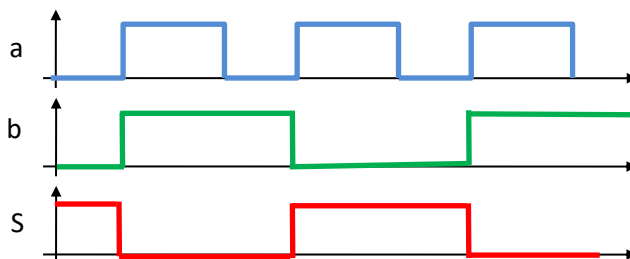
Une table de vérité est la représentation de l'évolution du comportement d'un système automatisé en fonction des variations de ses entrées. Chacune des variables est représentée sous une écriture binaire. Une table de vérité s'utilise principalement en logique combinatoire.

Elle est représentée sous la forme suivante :

Variables d'entrée (a et b)				Variable de sortie (L)
	a	b	L	
Evolution des variables d'entrée	0	0	0	Evolution de la Variable de sortie en fonction des états des entrées
	0	1	0	
	1	0	0	
	1	1	1	

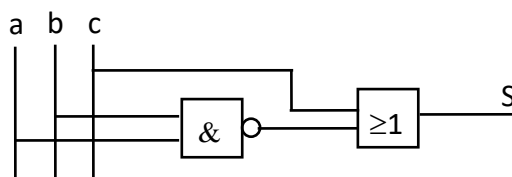
### Représentation par chronogramme :

Un chronogramme est une représentation schématique temporelle de l'évolution d'un système automatisé en fonction des variations d'état d'une ou plusieurs entrées. Cette évolution est représentée sous la forme suivante



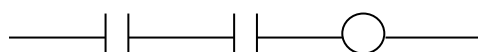
### Représentation en logigramme :

Un logigramme est un schéma représentant une succession de symboles logiques permettant d'obtenir par combinaison de variables d'entrées la sortie recherchée. Attention, les fonctions logiques sont des opérateurs logiques et non des opérateurs mathématiques. Le résultat obtenu sera un résultat logique et non un résultat mathématique.



**Remarque :** La fonction logique est représentée par un rectangle, à l'intérieur duquel on indique le type de fonction, avec à gauche les entrées et à droite les sorties.

### Représentation en langage à contact ou LADDER (exemple langage Automgen) :



## LE CODE BINAIRE PUR :

Le code binaire pur est une représentation numérique en base deux. Cette représentation permet de représenter des nombres sous forme de 1 et de 0, ou de décrire l'évolution des variables vraies ou non vraies d'un système automatisé, c'est ce code que nous allons utiliser pour établir une table de vérité. Le nombre de combinaisons possibles des variables se calcule de la façon suivante :

- 1 variable d'entrée  $2^1 = 2$  combinaisons de sortie
- 2 variables d'entrée  $2^2 = 4$  combinaisons de sortie
- 3 variables d'entrée  $2^3 = 8$  combinaisons de sortie
- 4 variables d'entrée  $2^4 = 16$  combinaisons de sortie
- n variables d'entrée  $2^n$  combinaisons de sortie

S.A à 1 variable

0
1

S.A à 2 variables

0	0
0	1
1	0
1	1

On retrouve la structure précédente en haut à droite de ce tableau, puis en dessous. La 2<sup>ème</sup> variable prend 2 fois la valeur 0 puis 2 fois la valeur 1

S.A à 3 variables

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

On retrouve la structure précédente en haut à droite de ce tableau puis en dessous. La 3<sup>ème</sup> variable prend 4 fois la valeur 0 puis 4 fois la valeur 1

S.A à 4 variables

0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

On retrouve la structure précédente en haut à droite de ce tableau puis en dessous, suivi de 8 x 0 et 8 x 1 pour la 4<sup>ème</sup> variable.