AUTOMATIQUE LOGIQUE COMBINATOIRE

1. Systèmes à logique combinatoire

Un système à logique combinatoire est un système dont l'état logique des sorties n'est décrit que par 2 valeurs (0 ou 1, vrai ou faux, éteint ou allumé,…) et dont les valeurs sont définies exclusivement à partir de l'état logique des entrées à l'instant considéré. On parle aussi de système à entrées-sorties Tout Ou Rien (TOR).

Les sorties peuvent alors être représentées comme une combinaison des entrées à partir des opérations logiques ET, OU et NON.

Exemples : Les systèmes électromécaniques type scie circulaire ou perceuse.

La rotation de la lame d'une scie circulaire ne s'effectue (sortie) que si la gâchette est enfoncée ET le bouton de sécurité est enfoncé (entrées).

Association : composant d'automatisme – variable binaire

De nombreux composants d'automatismes ne peuvent normalement prendre que 2 états différents :
- Une lampe est allumée ou éteinte
- Un détecteur ou un bouton-poussoir est actionné ou relâché
- Un moteur tourne ou est à l'arrêt
- …

A chacun de ces composants, on pourra associer une **variable binaire**, c'est à dire ne pouvant prendre que deux valeurs différentes, notées par convention **0** et **1**.
La convention habituellement utilisée est d'associer l'état « 1 » de la variable à la situation « actionnée », « activée », « travail », ou « en énergie » du composant.
L'état « 0 » est alors associé à la situation « relâchée », « désactivée », « repos », ou «hors énergie » du composant.

2. Opérations logiques, variables binaires

2.1. Structure de l'algèbre binaire

On considère l'ensemble {0,1} pour lequel on définit :
1. une relation d'équivalence notée **=**
2. deux lois de composition internes notées **+** et **.**
   (addition et multiplication logiques, OU et ET logiques)
3. une opération unaire : loi qui à tout élément a de E associe son complément $\overline{a}$
   (lu « $abarre$ », cette loi est appelée complémentation).

Ces opérations peuvent être représentées de manière exhaustive par les **tables de vérité** suivantes :

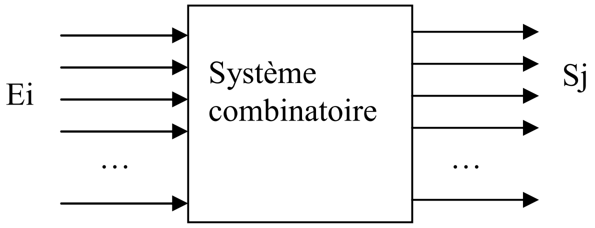
a	b	**a+b**
0	0	0
1	0	1
0	1	1
1	1	1

a	b	**a.b**
0	0	0
1	0	0
0	1	0
1	1	1

a	$\overline{a}$
0	1
1	0

2.2. Système combinatoire

Un système logique combinatoire, à tout instant, peut s'exprimer conformément au schéma suivant :



$$\forall\ t \quad Sj = f(Ei) \quad \begin{matrix} 1 \le i \le n \\ 1 \le j \le p \end{matrix}$$

Les fonctions de sortie Sj ne dépendent que des entrées Ei et à l'instant considéré.

On peut donc dire : « à un état des entrées correspond un unique état des sorties ».

Ei et Sj sont respectivement des **variables** et des **fonctions binaires** de ces variables, variables et fonctions ne pouvant prendre que deux valeurs **0** et **1** par convention.
Afin de représenter et de traiter de tels systèmes, il est nécessaire de s'appuyer sur un outil mathématique : **l'algèbre binaire**.

2.3. Opérateurs logiques élémentaires

Chacun des opérateurs logiques se présentera sous plusieurs représentations, plus ou moins proche de la réalisation matérielle de cette opération dans le système réel :

- Algébrique : équation (exemple : $S_3 = \overline{E_1} \,.\, \overline{E_2} + E_3.E_2$)
- Arithmétique : table de vérité
- Graphique : symbole logique

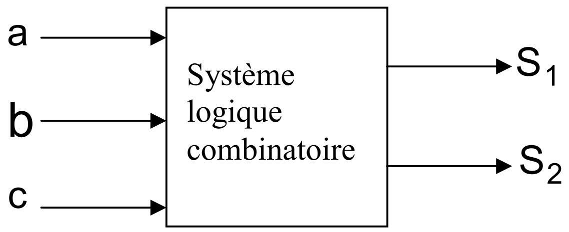
- Electronique : schéma développé (hors programme de PSI)
- Electrique : schéma développé (hors programme de PSI)
- Pneumatique : schéma développé (hors programme de PSI)

Chaque opérateur logique élémentaire OUI, NON, ET, OU permet l'opération logique correspondante.

2.1. Table de vérité

L'écriture d'une **table de vérité** est une méthode d'écriture systématique qui consiste à décrire toutes les combinaisons des variables figurant dans l'équation logique.

Ce tableau comporte autant de lignes qu'il existe de combinaisons possibles entre les **n** variables binaires, soit $\mathbf{2^n}$ combinaisons.



Ce système réalise 2 opérations binaires à à l'aide des 3 entrées : $2^3 = 8$ combinaisons

Table de vérité

a	b	c	S_1	S_2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

On peut alors déterminer les expressions algébriques de S_1, en considérant les différentes combinaisons des valeurs de a, b et c pour lesquelles S_1 est égale à 1.

Ainsi on peut écrire sous forme algébrique :

$$S_1 = \bar{a}.\bar{b}.c + \bar{a}.b.\bar{c} + \bar{a}.b.c + a.\bar{b}.\bar{c}$$

Dans cette expression, chaque terme de la somme logique correspond à une combinaison des 3 entrées pour laquelle S_1 est égale à 1.

Mais on peut aussi exprimer S_1 de manière plus dense en considérant le regroupement des 2 termes du milieu de l'expression en un seul et unique terme dans lequel la valeur logique de c n'apparait pas. En effet on peut dire que S_1 est égale à 1 dès que a vaut 0 et B vaut 1, quelle que soit la valeur logique de c :

$$S_1 = \bar{a}.\bar{b}.c + \bar{a}.b + a.\bar{b}.\bar{c}$$

Exercice : déterminer de même l'expression logique de S_2 sous la forme d'une somme logique de 5 termes puis simplifiée aves uniquement 2 termes.

On peut aussi réaliser le travail inverse et compléter la table de vérité d'une 3ème sortie

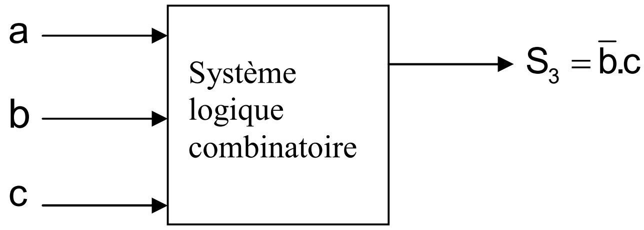


Table de vérité

a	b	c	S_3
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

2.2. Propriétés fondamentales de l'algèbre binaire

2.2.1. Commutativité

Le produit logique est commutatif, on peut écrire : $\boxed{a\,.\,b = b\,.\,a}$

De même, on peut commuter les termes d'une somme logique : $\boxed{a + b = b + a}$

2.2.2. Associativité

Les propriétés d'associativité sont applicables aux expressions logiques.

$$\boxed{a + (b + c) = (a + b) + c = a + b + c}$$

$$\boxed{a.(b.c) = (a\,.\,b)\,.\,c = a\,.\,b\,.\,c}$$

2.2.3. Distributivité

Distributivité de la fonction ET par rapport à la fonction OU :

$$\boxed{a\,.(b + c) = (a\,.\,b) + (a\,.\,c)}$$

Distributivité de la fonction OU par rapport à la fonction ET :

$$\boxed{a + (b\,.\,c) = (a + b)\,.\,(a + c)}$$

2.2.4. Absorption

$$\boxed{a + (a\,.\,b) = a.(1 + b) = a}$$

2.2.5. Théorèmes de DE MORGAN

Premier théorème de DE MORGAN

Le complément d'une somme logique est égal au produit des termes complémentés de cette somme.

$$\boxed{\overline{a + b} = \overline{a}\,.\,\overline{b}}$$

Le théorème s'applique quel que soit le nombre de termes de la somme et on peut alors généraliser la propriété :

$$\boxed{\overline{\sum_{i=1}^{n} a_i} = \prod_{i=1}^{n} \overline{a_i}}$$

exemple : $\overline{a + b + \overline{c}} = \overline{a}\,.\,\overline{b}\,.\,c$

Deuxième théorème de DE MORGAN

Le complément d'un produit logique est égal à la somme des termes complémentés de ce produit.

$$\boxed{\overline{a\,.\,b} = \overline{a} + \overline{b}}$$

Le théorème s'applique quel que soit le nombre de termes de la somme et on peut alors généraliser la propriété :

$$\boxed{\overline{\prod_{i=1}^{n} a_i} = \sum_{i=1}^{n} \overline{a_i}}$$

exemple : $\overline{a\,.\,b\,.\,\overline{c}} = \overline{a} + \overline{b} + c$

2.3. Opérateurs logiques supplémentaires

Opérateur NON –ET
C'est un opérateur ET dont la ,sortie est complémentée. Il se note NAND, contraction de NOT AND.

Equation : $\boxed{S = \overline{a\,.\,b} = \overline{a} + \overline{b}}$

Opérateur NON –OU
C'est un opérateur OU dont la ,sortie est complémentée. Il se note NOR.
La sortie de cette opérateur est dans l'état 1 si, et seulement si, toutes les entrées sont dans l'état 0. Dans les autres cas, la sortie est à 0.

Equation : $\boxed{S = \overline{a + b} = \overline{a}\,.\,\overline{b}}$

Opérateur OU exclusif

C'est un opérateur OU avec une particularité : la sortie est à 0 lorsque les deux entrées sont à 1.

La lampe s'allume si on appuie sur a seul ou sur b seul, mais elle est éteinte si on appuie sur les deux ou sur aucun des deux.

Equation : $\boxed{S = a \oplus b = a\,.\overline{b} + \overline{a}.b}$

Opérateur ET inclusif ou IDENTITE

C'est l'opérateur complémentaire du OU exclusif.

La lampe reste éteinte si on appuie sur a seul ou sur b seul, mais elle est allumée si on appuie sur les deux ou sur aucun des deux.

Equation : $\boxed{S = \overline{a}\,.\overline{b} + a.b}$

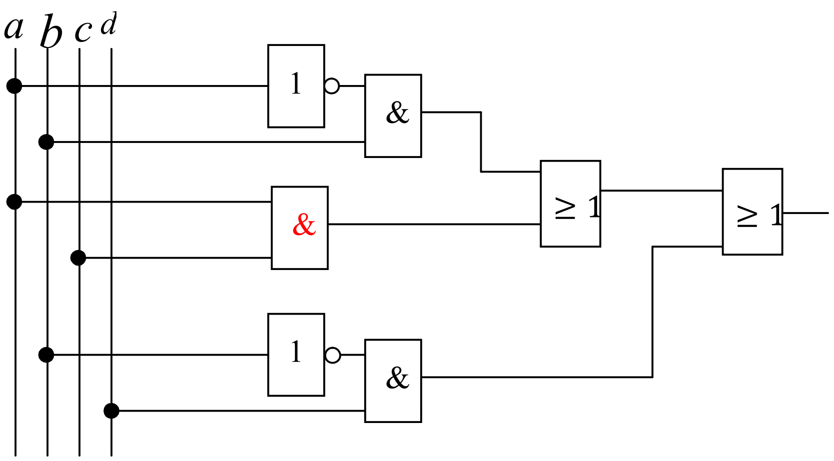
2.4. Représentation graphique des opérations logiques par logigramme

Ce tableau représente l'association des logigrammes élémentaires aux différentes opérations logiques élémentaires et suplémentaires

Opérateur	Fonction logique et notation algébrique	représentation norme IEC
OUI	$L = a$	a [1] L
NON	$L = \overline{a}$	a [1] L
ET	$L = a\,.\,b$ $L = \prod_{i=1}^{n} a_i$	a, b [&] L
OU	$L = a + b$ $L = \sum_{i=1}^{n} a_i$	a, b [≥ 1] L
NON-ET (NAND)	$L = \overline{a.b} = \overline{a} + \overline{b}$	a, b [&] L
NON-OU (NOR)	$L = \overline{a+b} = \overline{a}\,.\,\overline{b}$	a, b [≥1] L
OU exclusif	$L = a\,.\,\overline{b} + \overline{a}\,.\,b$ $L = a \oplus b$	a, b [=1] L
IDENTITE	$L = \overline{a}\,.\,\overline{b} + a\,.\,b$ $L = \overline{a \oplus b}$	a, b [=1] L

Un logigramme n'est pas unique pour une équation donnée.
C'est le schéma d'une association d'opérateurs logiques décrivant une équation logique combinatoire.

Exemple : $S = \overline{a}.b + c.a + d.\overline{b}$



Mais il y a d'autres représentations possibles

3. Les codages et transcodages

3.1 Définitions :

- **Codage** : suite de **caractères** formant un **mot** codant une information
- **Transcodage** : passage d'un code pour une information à un autre codage de cette même information.
- **Codage pondéré** : code pour lequel chaque caractère du mot a un poids (ordre).

3.2 Le codage binaire naturel

Correspond à l'expression d'un nombre en base 2.
Par exemple le nombre décimal 14 peut être représenté en binaire naturel par le mot binaire à 4 bits : 1110.
On peut aussi écrire : $(14)_{10}=(1110)_2$

D	B3	B2	B1	B0
0		**0**	**0**	**0**
1		**0**	**0**	**1**
2		**0**	**1**	**0**
3		**0**	**1**	**1**
4		**1**	**0**	**0**
5		**1**	**0**	**1**
6		**1**	**1**	**0**
7		**1**	**1**	**1**
8			**0**	**0**
9			**0**	**1**
10			**1**	**0**
11			**1**	**1**
12				**0**
13				**1**
14				**0**
15				**1**

3.3 Le codage binaire réfléchi (exemple de code Gray)

D	G3	G2	G1	G0
0		**0**	**0**	**0**
1		**0**	**0**	**1**
2		**0**	**1**	**1**
3		**0**	**1**	**0**
4		**1**	**1**	**0**
5		**1**	**1**	**1**
6		**1**	**0**	**1**
7		**1**	**0**	**0**
8				
9				
10				
11				
12				
13				
14				
15				

Un codage Gray correspond à une suite de mots binaires adjacents c'est-à-dire pour lesquels un seul bit change d'un mot à l'autre : 100110 110110
Le codage binaire réfléchi est un codage Gray particulier.

3.4 Le codage hexadécimal

Correspond à la base 16 et a été largement utilisé avec le développement de l'informatique, chaque caractère hexadécimal correspondant à un mot de 4 bits.
Les caractères hexadécimaux sont : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
Chaque caractère hexadécimal peut alors être codé en binaire à son tour, et on peut alors coder un nombre décimal de différentes manières :
$(14)_{10}=(E)_{16}$
$(163)_{10}=(A3)_{16}$

3.5 Le décimal codé binaire BCD (Binary Coded Decimal)

Correspond au codage des chacun des chiffres du nombre décimal sur un mot de 4 bit.
$(14)_{10}=(0001\ 0100)_{BCD}$

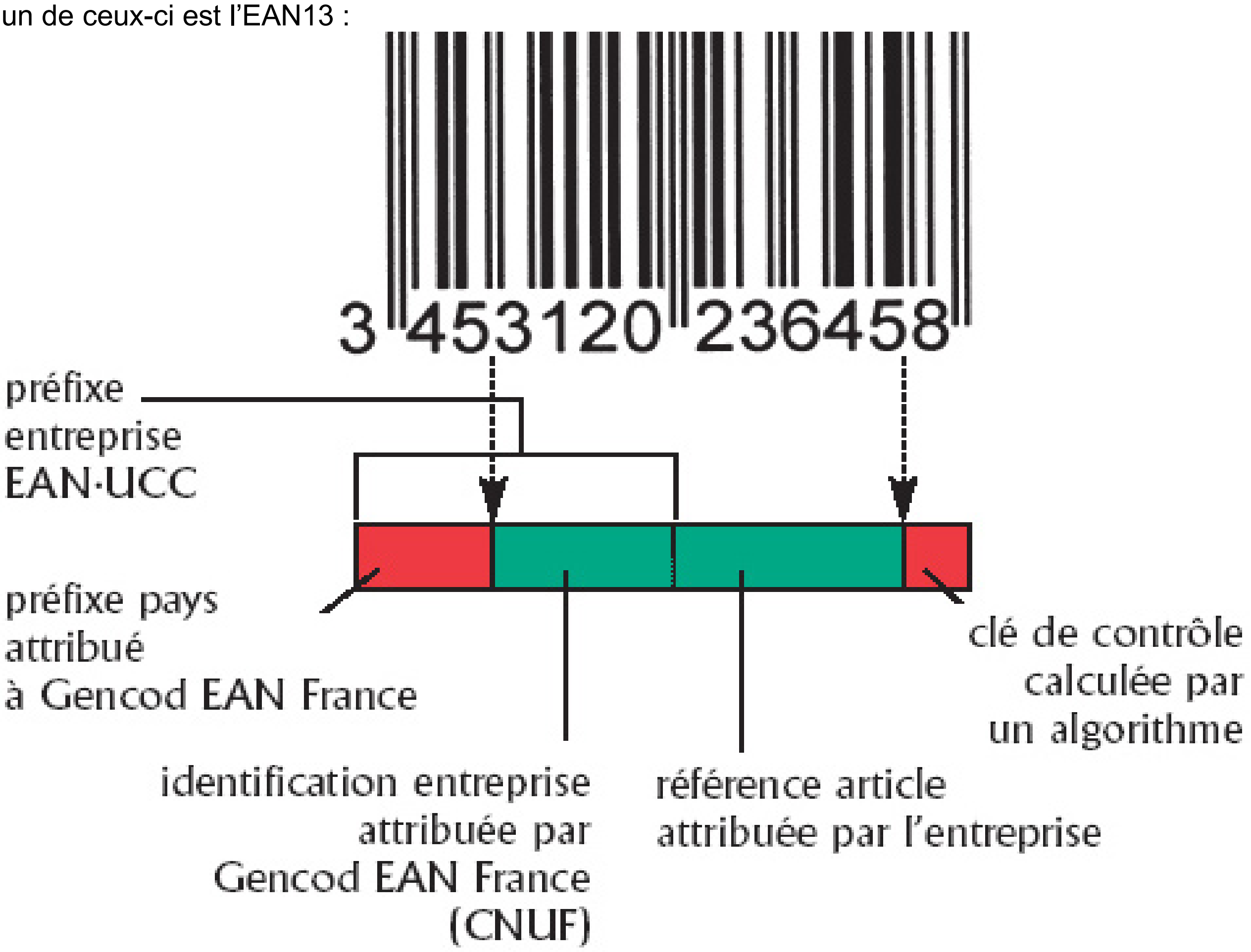
3.6 Codage d'un nombre décimal en binaire en hexadécimal ou hexadécimal codé binaire

On obtient le codage d'un nombre dans une base donnée par utilisation des puissances successives du nombre correspondant à la base ou par division euclidienne par ce nombre.

3.7 Les codes à barres

Ils sont nombreux et pratiques car lisibles à l'aide de capteurs optiques.

L'un de ceux-ci est l'EAN13 :



On a aussi les codes 2 parmi 5, ou 3 parmi 9 qui sont aussi des codes à barres détecteurs d'erreurs avec une clé comme pour l'EAN13.

Il y a aussi les désormais fameux QR code permettant de coder jusqu'à 4296 caractères numériques (contre 13 pour l'EAN13). De plus il est redondant et permet ainsi de détecter et corriger les erreurs de lecture et de transmission permettant une utilisation grand public de nos jours.

Différentes versions de Code QR

Version 1, 21×21, 10-25 caractères.	Version 2, 25×25, 20-47 caractères.	Version 3, 29×29, 35-77 caractères.	Version 4, 33×33, 67-114 caractères.	Version 10, 57×57, 174 à 395 caractère s.	Version 40, 177×177, 1 852 à 4 296 caract ères.

(source Wikipedia)

Les caractères sont des octets dont les bits sont représentés par des cases noircies ou non et agencés dans le tableau selon un ordre défini. Voici un exemple d'agencement à 8 caractères codés sur 8 bits

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8
2.6	2.7	2.8	5.3	5.4	5.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	1.3	1.4
1.8	6.3	6.4	6.5	8.1	8.2	1.6	1.7
7.2	6.6	6.7	6.8	8.3	8.4	8.5	7.1
7.4	7.5	3.1	3.2	8.6	8.7	8.8	7.2
7.7	7.8	3.3	3.4	3.5	4.1	4.2	7.3

https://barcode-coder.com/fr/specification-datamatrix-104.html