LA LOGIQUE COMBINATOIRE

Compétences associées
A2 : Analyser et interpréter une information numérique

DEFINITION

De nombreux dispositifs électroniques, électromécanique, (mécanique, électrique, pneumatique, etc....) fonctionnement en TOUT ou RIEN.
Ceci sous-entend qu'ils peuvent prendre 2 états. Exemple :

Arrêt marche	- Ouvert fermé -
Enclenché déclenché	- Avant arrière
Vrai faux	- Conduction blocage

Un système présentera un fonctionnement logique <u>combinatoire</u> si l'état à un instant t des variables de sortie ne dépend que de l'état des variables d'entrée au même instant t.

VARIABLE LOGIQUE

Une variable logique ne peut prendre que 2 états:

- Etat vrai: oui; haut; 1; high.
- Etat faux: non; bas; 0; low.

Pour ces raisons, il est beaucoup plus avantageux d'employer un système mathématique n'utilisant que 2 valeurs numériques.

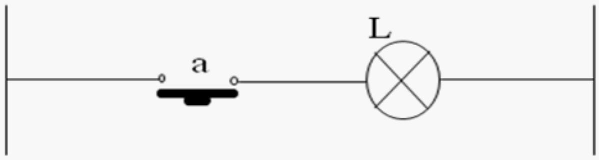
La variable binaire est aussi appelée variable **booléenne**. (De George Boole, mathématicien anglais 1815 -1864)

Du point de vue des contacts

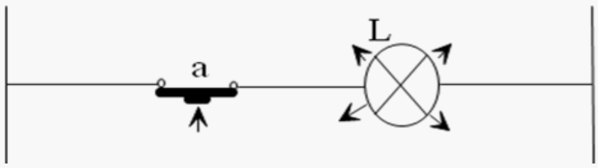
On choisit habituellement: *L'état « 1 » lorsqu'il y a action sur le contact*

L'état « 0 » lorsqu'il n'y a pas action sur le contact

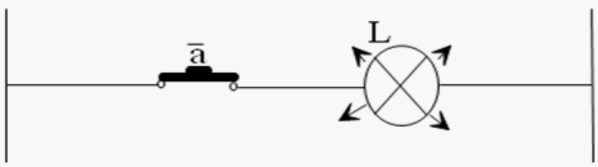
- *pas d'action sur a* $\Rightarrow a=0$
  a est au repos la lampe est éteinte $L=0$



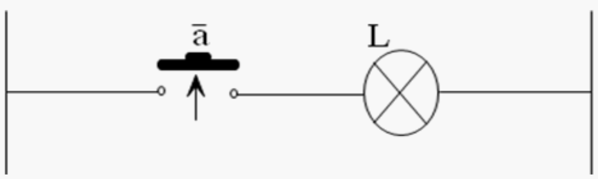
- *Action sur a* $\Rightarrow a=1$
  a est actionné la lampe s'allume $L=1$



- *Pas d'action sur* $\overline{a}$ $\Rightarrow \overline{a}=0$
- *au repos la lampe est allumée L=1*



- *Action sur* $\overline{a}$ $\Rightarrow \overline{a}=1$
  est actionné la lampe est éteinte $L=0$



REPRESENTATION D'UNE FONCTION

Une fonction **X** (exemple: allumer une lampe) peut comporter **n** variables logiques.
Nous obtenons $\mathbf{2^n}$ combinaisons pour ces **n** variables.
Pour chacune de ces combinaisons, la fonction peut prendre une valeur **0** ou **1**.
L'ensemble de ces $\mathbf{2^n}$ combinaisons des variables et la valeur associée de la fonction peut être représenté de différentes façons.

La table de vérité

Exemple:

X=1 si **a**=0 et **b**=1
a=0 et **b**=0
a=1 et **b**=0

a	b	X
0	1	1
0	0	1
1	1	0
1	0	1

La forme canonique

Pour écrire l'équation de **X** en fonction des 2 variables il faut dire autant de termes que de fois que la fonction est égale à 1.
Ce qui donne une écriture "algébrique" en notant :

La variable par sa lettre si elle vaut 1 (ex : **si a vaut 1** nous écrirons **a**)

La variable par sa lettre surlignée si elle vaut 0. (**si a vaut 0** nous écrirons $\bar{a}$ **,** nous lirons a barre)

Pour la table de vérité ci-dessus, cela nous donne

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

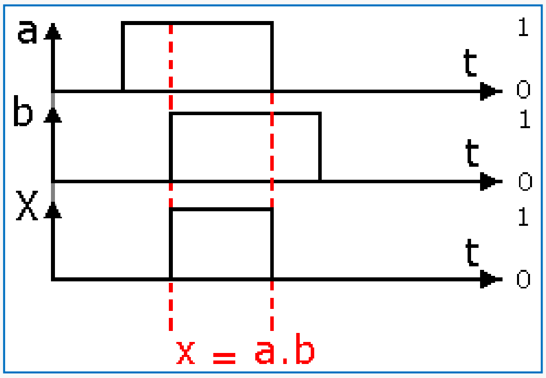
Cette forme d'écriture est appelée **FORME CANONIQUE**.

Chronogramme

Il existe une autre façon de représenter une fonction logique appelée **chronogramme** ou diagramme des temps.

Les variables binaires sont représentées par un **niveau (souvent de tension)** lorsqu'elles sont à **1**.

Elles évoluent **dans le temps** et nous représentons la fonction logique résultante de ces variables, également par un niveau de tension.



LES OPERATEURS LOGIQUES

La fonction NON ou NO

- *On associe à une variable binaire quelconque a son complément*

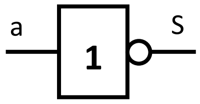
  S=1 si a = 0

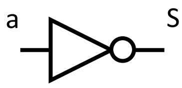
  $\bar{a} = 1$

a	S
0	1
1	0

- *Equation logique :* $S = \bar{a}$

Symbolisation	
Norme C.E.I	Norme A.N.S.I





La fonction ET ou AND

- *L'état 1 est obtenu lors de l'action simultanée sur les 2 variables a et b*

  S=1 si a=1 et b=1

a	b	S
0	0	0
0	1	0
1	0	0
1	1	1

- *Equation logique :* $S = a \,.\, b$

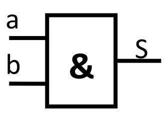
- *Propriétés :* $X \,.\, X = X$

  $X \,.1 = X$

  $X \,.\bar{X} = 0$

  $X \,.\, 0 = 0$

Symbolisation	
Norme C.E.I	Norme A.N.S.I



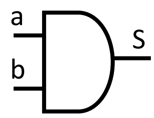
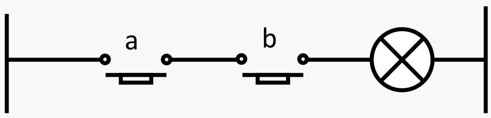


Schéma électrique:



La fonction OU ou OR

- *L'état 1 est obtenu lors de l'action sur la variable a ou sur la variable b ou des 2 .*

  S=1 si a=0 et b=1

  a=1 et b=0

  a=1 et b=1

a	b	S
0	0	0
0	1	1
1	0	1
1	1	1

- *Equation logique :* $S = a + b$

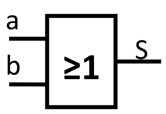
- *Propriétés :* $X + X = X$

  $X + 1 = 1$

  $X + \bar{X} = 1$

  $X + 0 = X$

Symbolisation	
Norme C.E.I	Norme A.N.S.I



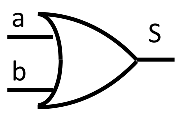
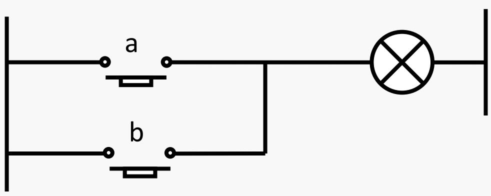


Schéma électrique:



La fonction OU exclusif ou XOR

- *L'état 1 est obtenu lors de l'action sur la variable a ou sur la variable b mais pas des 2 .*

S=1 si a=0 et b=1
a=1 et b=0

a	b	S
0	0	0
0	1	1
1	0	1
1	1	0

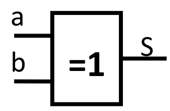
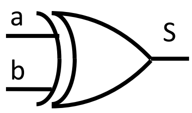
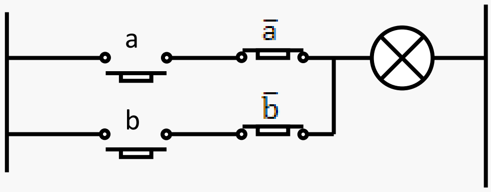
Symbolisation	
Norme C.E.I	**Norme A.N.S.I**
	

Schéma électrique:



- *Equation logique :* $S = a \oplus b$

La fonction NON-ET ou NAND

- *L'état 0 ($\overline{1}$) est obtenu lors de l'action sur la variable a ou sur la variable b mais pas des 2 .*

a	b	S
0	0	1
0	1	1
1	0	1
1	1	0

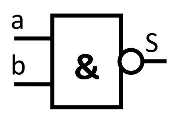
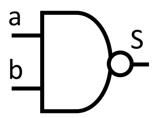
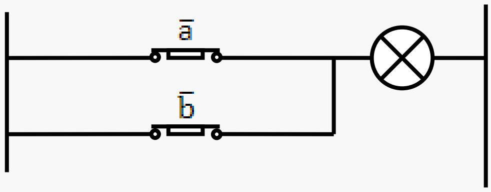
Symbolisation	
Norme C.E.I	**Norme A.N.S.I**
	

Schéma électrique:



- *Equation logique :* $S = \overline{a \, . \, b} = \bar{a} + \bar{b}$

La fonction NON-OU ou NOR

- *L'état 1 est obtenu s'il n'y aucune action, ni sur a ni sur b.*

a	b	S
0	0	1
0	1	0
1	0	0
1	1	0

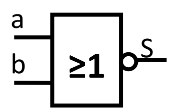
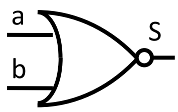
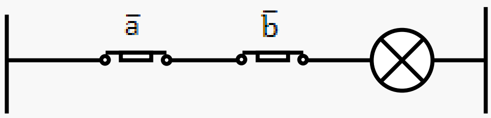
Symbolisation	
Norme C.E.I	**Norme A.N.S.I**
	

Schéma électrique:



- *Equation logique :* $S = \overline{a + b} = \bar{a} \, . \, \bar{b}$

ALGEBRE DE BOOLE

Redondance : $x + \bar{x} . y = x + y$

Distributivité : $a . (b + c) = a . b + a . c$

Relation de De Morgan : $\overline{a . b} = \bar{a} + \bar{b}$

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

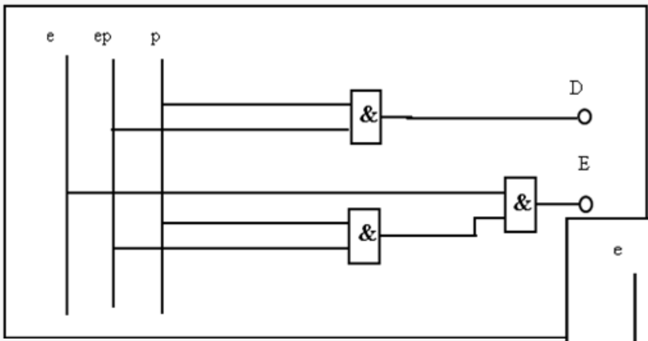
$S = \overline{\overline{\bar{a} . b} + b . \bar{c}} =$ $S = \overline{c . \bar{b} + a + \bar{c}} =$

SCHEMA LOGIQUE

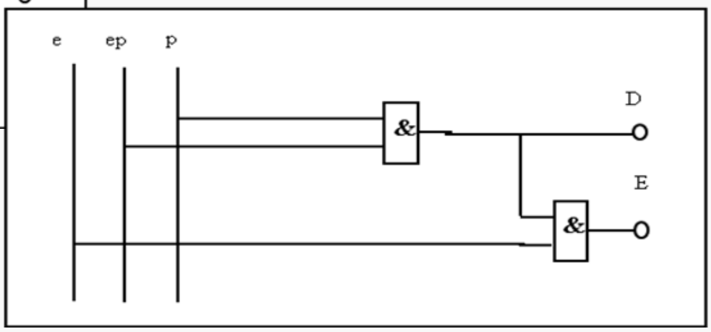
Un schéma logique est la représentation graphique de l'équation logique.

Exemple : D = ep . p

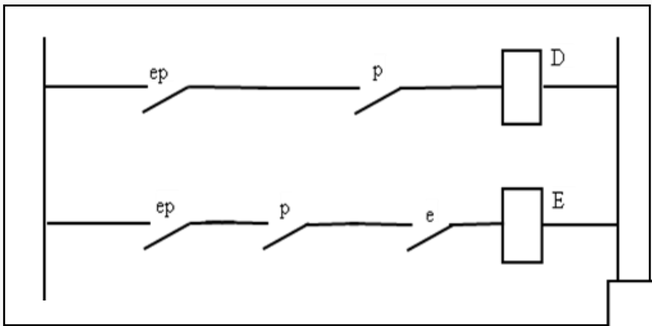
E = ep . e . p



*Ici, le **logigramme** laisse apparaître que l'on peut aussi écrire l'équation de E en fonction de D*



Le logigramme n'est pas le seul outil de schéma logique,



On peut dessiner un schéma à contacts (en échelle).

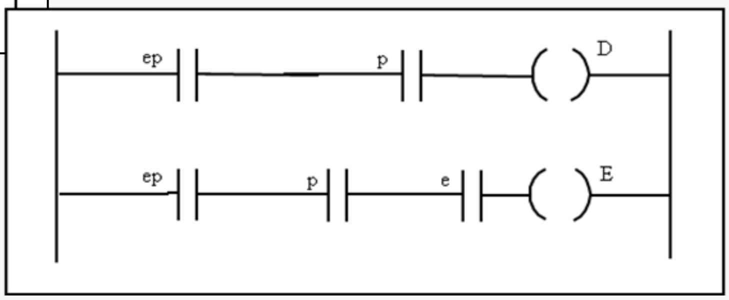


Schéma à contacts (Règles de schéma SCHNEIDER. PL7_1)

TABLEAU DE KARNAUGH

Ce tableau reprend les indications de la table de vérité pour les mettre sous une autre forme

- *Le nombre de cases est égal au nombre de lignes de la table de vérité*
- *Chaque ligne et chaque colonne correspond à un état d'une ou plusieurs variables d'entrées*

Exemples:

a \ b	0	1
0		
0		

Variables d'entrées a et b

a \ bc	00	01	11	10
0				
0				

Variables d'entrées a, b, c

- *Chaque ligne et chaque colonne est numérotée avec l'état que peuvent prendre les variables d'entrées*
- *Entre deux cases adjacentes, seule une variable d'entrée peut changer d'état*

Exemple :

Soit 4 variables a,b,c,d

La case x correspond à: a, b, c, d = 0

La case y correspond à: a, c = 0 b, d =1

La case z correspond à: a, b, c, d = 1

ab \ cd	00	01	11	10
00	x			
01		y		
11			z	
10				

TRANSPOSITION D'une table de vérité en un tableau de KARNAUGH

Exemple

	C	B	A	S
(a)	0	0	0	0
(b)	0	0	1	0
(c)	0	1	1	1
(d)	0	1	0	0
(e)	1	1	0	1
(f)	1	1	1	1
(g)	1	0	1	1
(h)	1	0	0	0

Devient :

C \ BA	00	01	11	10
0	0	0	1	0
1	0	1	1	1

SIMPLIFICATION DES FONCTIONS

Regroupement des cases

Pour simplifier l'équation, il suffit de regrouper les cases qui possèdent le même état de la variable de sortie dans les conditions suivantes

- *Les cases regroupées doivent être adjacentes*
- *Le regroupement des cases se fait par puissance de 2 (2,4,8,16,32....)*
- *Les cases possédant le même état de la variable de sortie doivent être utilisées*
- *Le regroupement doit être le plus grand possible*
- *Une case peut très bien appartenir à plusieurs regroupements.*

A partir de l'exemple précédent: 3 regroupements possibles

C \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

A.B

C \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

B.C

C \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

A.C

Equation de chaque regroupement.

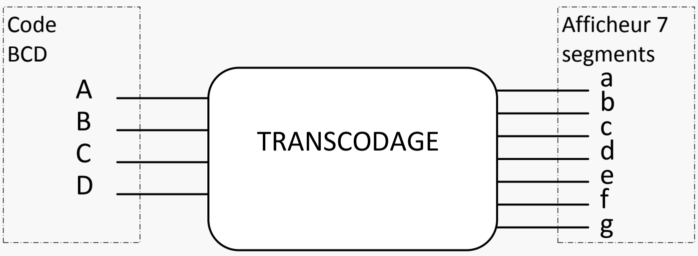
Chaque regroupement donne le produit logique des variables d'entrée qui n'ont pas changées d'état. L'ensemble de ces regroupements est une somme logique

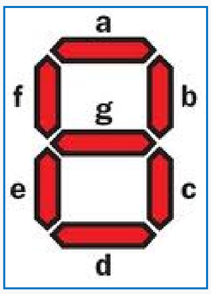
Dans notre cas : **S = A.B + B.C + A.C**

Cas particuliers

Lors d'un tableau à n variables, si les 2^n cas ne sont pas tous décrits, il subsistera alors des cas que l'on qualifiera d'indifférents. Ils seront symbolisés par la variable x dans le tableau de Karnaugh on pourra selon les besoins les remplacer individuellement par des 1 ou des 0

REALISATION D'UN DECODEUR BCD→7 SEGMENTS





- *Etablir la table de vérité du transcodeur.*
- *En utilisant KARNAUGH pour chaque segment à commander, trouver le logigramme correspondant pour chaque segment. Le réaliser à l'aide de portes NAND à 2 entrées*

- *Table de vérité du transcodeur*

Nb décimal	Codage BCD				a	b	c	d	e	f	g
	D	C	B	A							
0											
1											
2											
3											
4											
5											
6											
7											
8											
9											

- *Tableaux de KARNAUGH*

a

DC \ BA	00	01	11	10
00				
01				
11				
10				

b

DC \ BA	00	01	11	10
00				
01				
11				
10				

c

DC \ BA	00	01	11	10
00				
01				
11				
10				

d

DC \ BA	00	01	11	10
00				
01				
11				
10				

e

DC \ BA	00	01	11	10
00				
01				
11				
10				

f

DC \ BA	00	01	11	10
00				
01				
11				
10				

g

DC \ BA	00	01	11	10
00				
01				
11				
10				

- ***Equations:***

a =

b =

c =

d =

e =

f =

g =