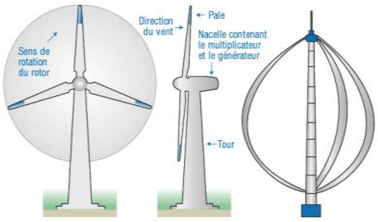
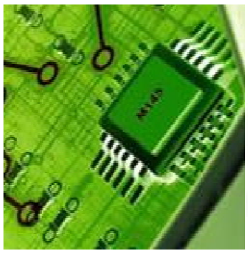
STI2D SIN





La logique combinatoire

I. Historique :

Les hommes ont voulu traduire leurs idées en langage mathématique de façon à permettre à une machine de reproduire des actions et réactions des hommes de façon toujours identique. Dans la logique combinatoire, l'évolution de l'état des sorties ne dépend que de l'état des entrées, indépendamment de l'état de la sortie au moment considéré. On ne tient donc pas compte du temps qui passe. L'électronique qui est associée à ces équations est la fonction logique. Dans un premier temps associée à un composant, la fonction logique se programme de nos jours dans des circuits spécialisés que nous verrons plus tard.

II. Vocabulaire :

1- Les états logiques :

Une variable ne peut prendre que deux états notés 0 ou 1 (le courant ne passe pas ou le courant passe). Il faut admettre également que tout élément « a » possède un inverse noté " ". Ainsi = comme indiqué dans la table ci-dessous :

0	1	0
1	0	1

On a généralement :
- a = 0 pour un contact ouvert ➔
- a = 1 pour un contact fermé ➔

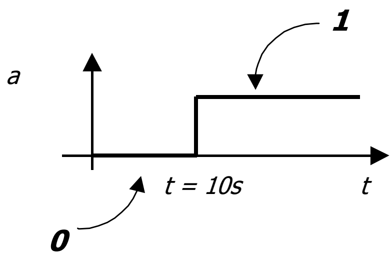
2- La table de vérité :

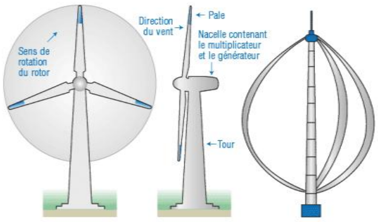
Une table de vérité est un tableau dans lequel on va renseigner les états logiques des variables d'entrée et de sortie. Les entrées sont regroupées à gauche et la ou les sorties à droite. On a par exemple :

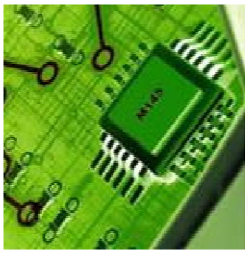
a	S
0	1
1	0

3- Les chronogrammes:

Les chronogrammes sont des représentations graphiques qui montrent l'évolution des entrées/sorties en fonction du temps. L'axe des ordonnés correspond à l'état logique (0 ou 1) et l'axe des abscisses correspond au temps. On a par exemple :

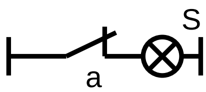
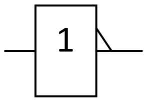
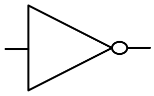
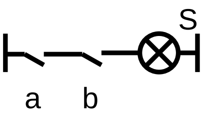
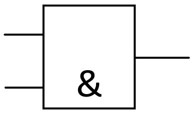
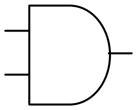
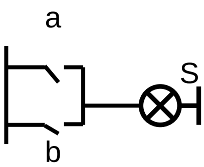
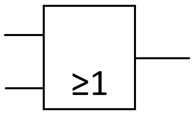
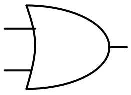
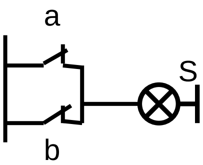
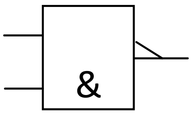
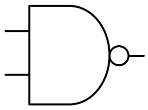
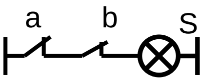
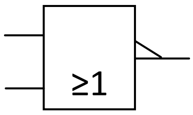
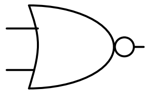
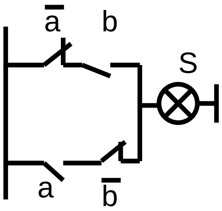
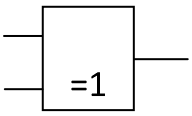
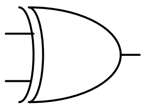


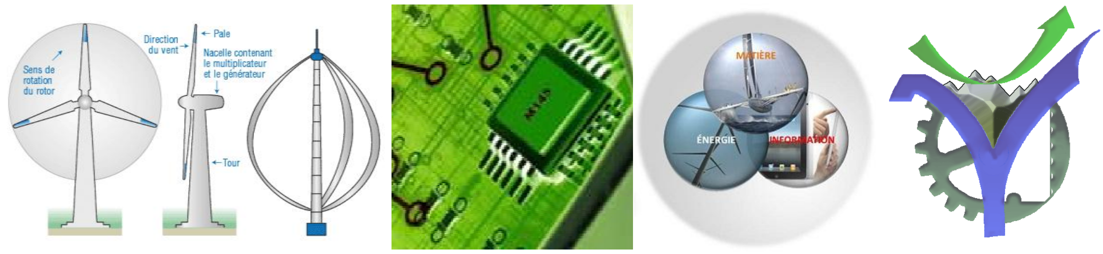




III. Synthèse des fonctions logiques :

Il existe plusieurs types de circuits intégrés conçus dans des technologies différentes et par plusieurs fabricants.

Nom de la fonction	Schéma électrique	Table de vérité	représentation -norme européenne-	représentation -norme américaine-	équation
NON		a / S: 0 → 1 ; 1 → 0			S =
ET		a b S: 0 0 0 ; 0 1 0 ; 1 0 0 ; 1 1 1			S = a . b
OU		a b S: 0 0 0 ; 0 1 1 ; 1 0 1 ; 1 1 1			S = a + b
NON ET		a b S: 0 0 1 ; 0 1 1 ; 1 0 1 ; 1 1 0			S = $\overline{a.b}$
NON OU		a b S: 0 0 1 ; 0 1 0 ; 1 0 0 ; 1 1 0			S = $\overline{a+b}$
OU Exclusif		a b S: 0 0 0 ; 0 1 1 ; 1 0 1 ; 1 1 0			S = a $\oplus$ b



IV. Le logigramme :

Le traitement logique des informations peut nécessiter la mise en œuvre d'un nombre important d'opérateurs binaires qui sont interconnectés.

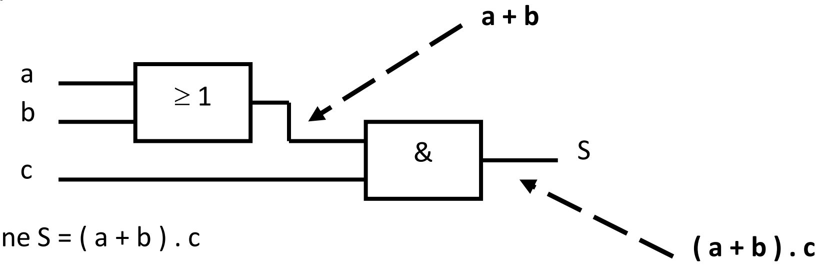
1- Association d'opérateurs logiques :

La représentation graphique de l'association de plusieurs opérateurs binaires est un logigramme.

2- Décodage d'un logigramme

Décoder un logigramme revient à rechercher la (ou les) combinaisons d'état des variables qui affecte(nt) à la sortie l'état logique 1.

Mise en équation successive :



Ce qui donne S = (a + b) . c

V. Résolution d'exercices :

1- Algèbre de Boole :

- *a + 0 = a*
- *a . 0 = 0*
- *a . (b + c) = a . b + a . c*
- $\overline{a+b} = \overline{a}.\overline{b}$
- *a + 1 = 1*
- *a . 1 = a*
- *a + (b . c) = (a + b).(a + c)*
- $\overline{a.b} = \overline{a} + \overline{b}$
- *a +   = 1*
- *a .   = 0*
- a +   .b = a + b
- *a + a = a*
- *a . a = a*
-   + a . b =   + b

2- A partir d'une table de vérité :

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

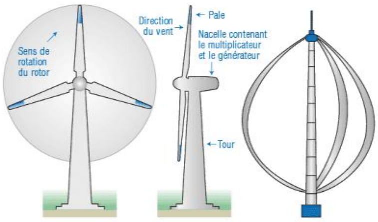
S = ($\overline{a}$. b . c) + (a . $\overline{b}$. c) + (a . b . c)

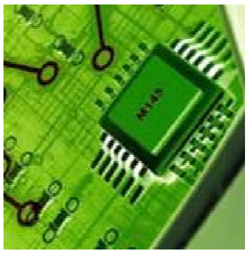
Après simplification, on obtient S = (a + b) . c

3- A partir d'un tableau de Karnaugh :

Afin de faciliter les simplifications algébriques il existe une méthode graphique qui facilite le travail. C'est la disposition sous forme de tableau de KARNAUGH.

La formation des ces tableaux respecte une règle principale : l'ordre d'évolution des variables est le code GRAY. La disposition en suite codée GRAY permet de repérer les variables absorbées.





Exemple :

cd / ab	00	01	11	10
00	0	1	1	0
01	0	0	0	1
11	0	0	0	1
10	0	0	0	0

Il faut former lors de la réunion des cases contiguës, des domaines les plus grands possibles. Ces domaines vont de 1 à 2^n cases, « n » étant le nombre de variables du tableau. Exemple de domaines réalisables :

- *Réunion de 2^1 cases (absorption d'une variable car a + =1)*

cd / ab	00	01	11	10
00	0	1	1	0
01	0	0	0	1
11	0	0	0	1
10	0	0	0	0

S = . .d + b.c.

cd / ab	00	01	11	10
00	0	1	0	0
01	0	0	0	0
11	1	0	0	1
10	0	1	0	0

S = a.b. + . . d

- *Réunion de 2^2 cases :*

cd / ab	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

S = b . d

cd / ab	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

S = .d + a .

cd / ab	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

S = .

cd / ab	00	01	11	10
00	0	1	0	0
01	0	1	0	0
11	1	1	1	1
10	0	1	0	0

S = . d

- *Réunion de 2^3 cases :*

cd / ab	00	01	11	10
00	0	1	1	0
01	1	1	1	1
11	1	1	1	1
10	0	1	1	0

S = b + d

cd / ab	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

S =

cd / ab	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

S =