

SYSTEME COMBINATOIRE

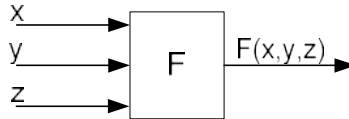
I. Définitions

I.1. Variable binaire

Une variable binaire ou (logique) est une variable ne pouvant prendre que deux états distincts et non simultanés qu'on notera 0 et 1. **Exemple** : un interrupteur est fermé (état 1) ou ouvert (état 0).

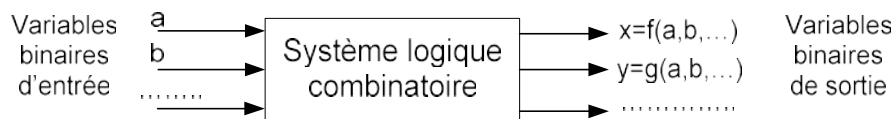
I.2. Fonction binaire

F est une fonction binaire si à une ou plusieurs variables x, elle associe une image f(x) qui ne peut prendre elle-même que deux états 0 et 1.



I.3. Système combinatoire

Un système combinatoire est un système qui à tout instant peut s'exprimer conformément au schéma suivant.



I.4. Table de vérité

La table de vérité représente l'état de la variable de sortie pour chacune des combinaisons des n variables d'entrée (2^n lignes).

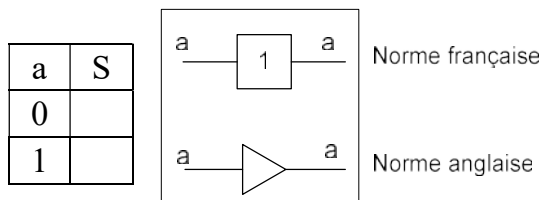
II. Fonctions binaires élémentaires

II.1. Fonction « OUI »

La fonction **OUI** est dite unaire (ne s'applique qu'à un seul opérande). Elle affecte à la variable de sortie l'état logique de la variable d'entrée.

Soient a l'entrée et S la sortie :

S = ...

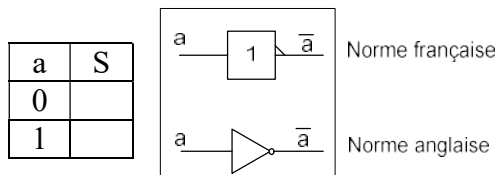


II.2. Fonction « NON »

La fonction **NON** est dite unaire (ne s'applique qu'à un seul opérande). Elle affecte à la variable de sortie l'état complémentaire de la variable d'entrée.

Soient a l'entrée et S la sortie :

S = ...



II.3. Fonction « ET »

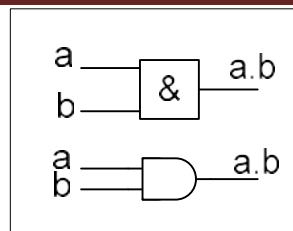
La fonction **ET** est le **produit logique**. Elle affecte à la variable de sortie l'état 1 si et seulement si les variables d'entrées sont à 1 simultanément.

Soient a et b les entrées et S la sortie :

S = ...

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

Table de vérité



II.4. Fonction « OU »

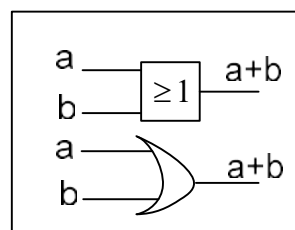
La fonction **OU** est le **somme logique**. Elle affecte à la variable de sortie l'état 1 si et seulement si une variable d'entrée est à 1.

Soient a et b les entrées et S la sortie :

S = ...

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

Table de vérité



III. Fonctions binaires induites :

III.1. Fonction « NAND »

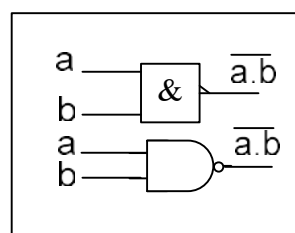
C'est l'association d'un NON et d'un ET. Elle affecte à la variable de sortie l'état 0 si et seulement si les variables d'entrées sont à 1 simultanément.

Soient a et b les entrées et S la sortie :

S = ...

a	b	a↑b
0	0	
0	1	
1	0	
1	1	

Table de vérité



III.2. Fonction « NOR »

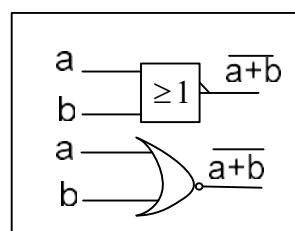
C'est l'association d'un NON et d'un OU. Elle affecte à la variable de sortie l'état 1 si et seulement si les variables d'entrée sont à 0 simultanément.

Soient a et b les entrées et S la sortie :

S = ...

a	b	a↓b
0	0	
0	1	
1	0	
1	1	

Table de vérité



III.3. Fonction « OU exclusif »

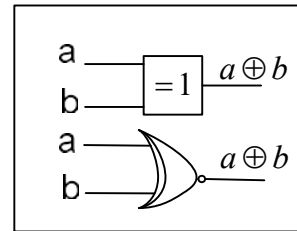
Elle ne prend l'état 1 que si une seule des entrées est à 1.

Soient a et b les entrées et S la sortie :

S = ...

a	b	$a \oplus b$
0	0	
0	1	
1	0	
1	1	

Table de vérité



IV. Les expressions logiques et leurs simplifications

IV.1. Propriétés des opérateurs logiques fondamentaux

Soient a, b et c trois variables logiques

✓ Commutativité :

$a+b = b+a$ (commutativité de l'opérateur OU)

$a.b = b.a$ (commutativité de l'opérateur ET)

✓ Associativité :

$(a+b) + c = a + (b+c) = a+b+c$ (distributivité du produit logique sur la somme logique)

$(a.b).c = a.(b.c) = a.b.c$ (distributivité de la somme logique sur le produit logique)

✓ Identité :

$a+0 = a$

$a.1 = a$

✓ Complémentaire :

$a + \bar{a} = 1$

$a \cdot \bar{a} = 0$

✓ Idempotence :

$a+a = a$

$a.a = a$

✓ Elément absorbant :

$a+1 = 1$

$a.0 = 0$

IV.2. Théorème de Morgan

Le but est d'exprimer les fonctions ET, OU et NON exclusivement à l'aide des fonctions NOR seuls ou NAND seule. On dit que les opérateurs NOR et NAND sont universels ou complets.

Premier théorème :

$\overline{a + b} = \bar{a} \cdot \bar{b} \Rightarrow$ Généralisation $\overline{\sum_{i=1}^n a_i} = \prod_{i=1}^n \bar{a}_i$ où les a_i sont les variables, $i \in [1, n]$

Second théorème :

$\overline{a \cdot b} = \bar{a} + \bar{b} \Rightarrow$ Généralisation $\overline{\prod_{i=1}^n a_i} = \sum_{i=1}^n \bar{a}_i$ où les a_i sont les variables, $i \in [1, n]$

IV.3. Expression des fonctions de base à l'aide des fonctions universelles

➤ $\bar{a} = \dots\dots\dots$

➤ $a.b = \dots\dots\dots$
 $a.b = \dots\dots\dots$

➤ $a + b = \dots\dots\dots$
 $a + b = \dots\dots\dots$

V. Ecriture des fonctions booléennes

V.1. Définitions

On appelle **minterme** de n variables, un produit logique de ces dernières (complémentées ou non). Avec n variables, on construit 2^n mintermes, c'est-à-dire autant que de combinaisons possibles de n éléments prenant deux états.

Exemple : pour deux variables a et b, voici les quatre mintermes :

On appelle **maxterme** de n variables, une somme logique de ces dernières (complémentées ou non). De la même manière que pour les mintermes, on construit 2^n maxtermes avec n variables.

Exemple : pour deux variables a et b, voici les quatre maxtermes :

V.2. Première forme canonique :

La première forme canonique d'une expression booléenne est composée d'une somme de mintermes exclusivement. Pour une expression donnée, cette forme est unique.

Remarque : La somme de tout les mintermes de n variables vaut toujours 1 puisqu'il existe toujours un minterme de n variables valant 1

Exemple :

Soient a et b les entrées et S la sortie :

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

Puisque la fonction n'est vrai que lorsqu'elle vaut 1 on ne s'occupe que des termes qui donnent $S = 1$
 $S = \dots\dots\dots$

V.3. Deuxième forme canonique :

La deuxième forme canonique d'une expression booléenne est composée d'un produit de maxtermes exclusivement. Pour une expression donnée, cette forme est unique.

Remarque : Le produit de tout les maxtermes de n variables vaut toujours 0 puisqu'il existe toujours un maxterme de n variables valant 0.

Exemple :

Soient a et b les entrées et S la sortie :

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

$S = \dots\dots\dots$

VI. Simplification des fonctions logiques

VI.1. Méthode algébrique

On utilise les relations fondamentales pour faire des simplifications.

Exemple :

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

.....

.....

.....

VI.2. Méthode graphique (Tableau de KARNAUGH)

Partant de la première forme canonique de la fonction on va construire la table de vérité.

A l'aide de la table de vérité on va diviser les variables et les mettre sous la forme d'un tableau à 2 dimensions, les variables seront écrit en code Gray.

Ce tableau ainsi construit s'appelle tableau de Karnaugh.

Exemples

• Cas 2 variables :

Soient a et b les variables d'entrées et S la variables de sortie

Tableau de Karnaugh

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

Table de vérité

S = ...

• Cas 3 variables

Soient a, b et c les variables d'entrées et S la variable de sortie

Tableau de Karnaugh

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

Table de vérité

S = ...

Afin de simplifier les expressions logiques, on regroupe les cases contenant des 1 lorsqu'elles sont adjacentes.

Règles de regroupement (voir les exemples de l'annexe) :

- On ne peut choisir que des groupements de type 2^n c à d : 1, 2, 4, 8,...
- Une case peut appartenir à un ou plusieurs groupements.
- Toutes les cases ayant le même chiffre doivent être utilisées.

Remarque : On commence par choisir le groupement qui peut avoir le plus de grand nombre de « 1 » .

Exemples :

	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	1	1
10	1	0	0	1

S = ...

	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	0	0
10	1	0	0	1

S = ...

	00	01	11	10
00	1	1	1	1
01	1	0	0	0
11	1	1	1	1
10	1	1	1	1

S = ...

VI.3. Simplification d'une fonction incomplètement définie**Exemple :**Soit $0 \leq N \leq 9$, N écrit en binaireSoit S la variable de sortie qui prend la valeur « 1 » si $2 \leq N \leq 5$

N	b ₃	b ₂	b ₁	b ₀	S
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					

	00	01	11	10
00				
01				
11				
10				

S = ...

VI.4. Logigramme

Appelé ainsi diagramme logique ou circuit logique, il illustre la façon de construction de la fonction binaire à partir des portes logiques.

Exemple : construire le logigramme de la fonction précédente**a/** en utilisant des fonctions logiques de base**b/** en utilisant les fonctions universelles.