Introduction à l'électronique Numérique

Licence Physique et Applications

Électronique combinatoire

Fabrice CAIGNET
LAAS - CNRS
fcaignet@laas.fr
http://www.laas.fr/~fcaignet

Plan du Cours

I. Les différents types de codage
II. La logique combinatoire
    A. Le système combinatoire
    B. La logique booléenne
    C. Les tableaux de Karnaugh

III. Synthèse de systèmes combinatoires

I. Les systèmes de codage binaires

Introduction

Le système de numération le mieux adapté pour effectuer des calculs est le système binaire, ou à base 2, qui ne comprend que deux caractères 0 et 1.

Dans un système numérique quelconque, les informations circulent sous la forme de mots binaires formés de suites de 1 et de 0. On fixe à l'avance le nombre d'élément de ces mots (un octet est un mot de huit éléments) et la manière de les écrire appelée code

Chaque élément de ces mots binaires est appelé élément binaire (eb) ou plus communément bit, contraction de l'expression "binary digit

Il existe plusieurs façons d'écrire les mots binaires car, selon les besoins, on a développé plusieurs codes

I. Les systèmes de codage binaires

Définition

Si l'on considère la base décimale (10) :

- **Les caractères définissant la base sont : 0, 1,2,3,4,5,6,7,8,9**
- **Un mot (ou chiffre) est une combinaison de ces 10 caractères**

Ex : 1664

Mot de 4 caractères

Poids fort — **Poids faible**

- **Le nombre se décompose sous la forme :**

$$1664 = 1 * 10^3 + 6 * 10^2 + 6 * 10^1 + 4 * \underline{10}^{0}$$

Poids

Base

I. Les systèmes de codage binaires

Le codage binaire naturel (base 2)

Comme précédemment, tout nombre binaire peut s'écrire comme un développement suivant les puissances de 2

$$1100101 = 1*2^6 + 1*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0$$

Poids

Base

1100101

Poids fort MSB

Poids faible LSB

I. Les systèmes de codage binaires

Le codage binaire naturel (base 2)

Code décimal	Code binaire naturel
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

I. Les systèmes de codage binaires

Le codage hexadécimal (base 16)

- **Les caractères définissant la base sont : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F**
- **Le nombre s'écrit :**

$$F2A_H = 15 * 16^2 + 2 * 16^1 + 10 * 16^0 = 8382_d$$

Le principal avantage de ce code est de pouvoir codé sur un mot court, un chiffre binaire important

$$1001\ 0101\ 1101_b = 95D_H$$

1 caractère hexa code 4 bits (ou quartet)

I. Les systèmes de codage binaires

Le passage d'un code à l'autre

Binaire vers hexadécimal

Un nombre hexadécimal est « découpable » en quartets facilement codables en binaire. Donc, pour convertir du binaire en hexadécimal, on divise le nombre binaire en « tranches de quatre » en partant de la droite. Chacun des « paquets » est ensuite converti en hexadécimal.

$$1001\ 0101\ 1101_b = 95D_H$$

1 caractère hexa code 4 bits (ou quartet)

I. Les systèmes de codage binaires

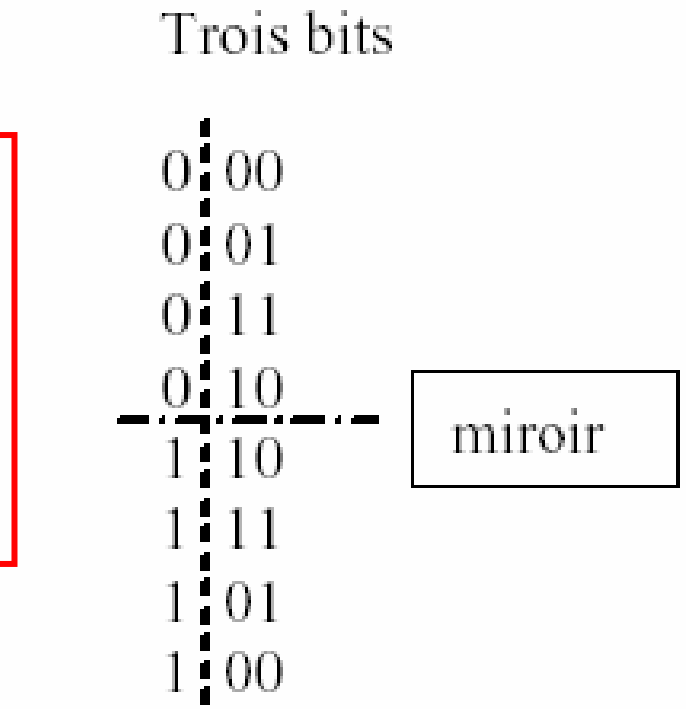
Le codage binaire réfléchit ou code Grey (base 2)

Dans ce code, deux nombres successifs ne diffèrent que d'un bit. Il a été élaboré pour éviter les risques d'erreur de détection d'une information dans les systèmes réels

Règle lorsque l'on compte :

(1) on change le bit de poids faible en premier. Si la combinaison existe, on passe au bit de rang supérieur

(2) On ne peut changer qu'un seul bit à la fois (d'une ligne sur l'autre)



Le code Grey

Code décimal	Code binaire naturel	Code binaire réfléchi
0	0	0000
1	1	0001
2	10	0011
3	11	0010
4	100	0110
5	101	0111
6	110	0101
7	111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

I. Les systèmes de codage binaires

Le codage des entiers négatifs : complément à 2

Règle : Le bit de poids fort est utiliser pour coder le signe :

- **0 : si l'entier est positif**
- **1 : si l'entier est négatif**

Le codage du nombre négatif s'effectue de la façon suivante :

$$-B = \left|\overline{B}\right| + 1$$

Exemple sur 4 bit : $-2 = \left|\overline{2_b}\right| + 1$

$$-2 = \overline{0010} + 1$$

$$-2 = 1101 + 1$$

$$-2 = 1110$$

2 écrit en binaire sur 4 bits

	c_3	c_2	c_1	c_0
7	0	1	1	1
6	0	1	1	0
5	0	1	0	1
4	0	1	0	0
3	0	0	1	1
2	0	0	1	0
1	0	0	0	1
0	0	0	0	0
-1	1	1	1	1
-2	1	1	1	0
-3	1	1	0	1

A. Définitions : Notions élémentaires en électroniques numérique

L'électronique combinatoire

1. Le système combinatoire

On appelle **système combinatoire** tout système numérique dont les sorties sont exclusivement définies partir des variables d'entrée (***Figure 1***).

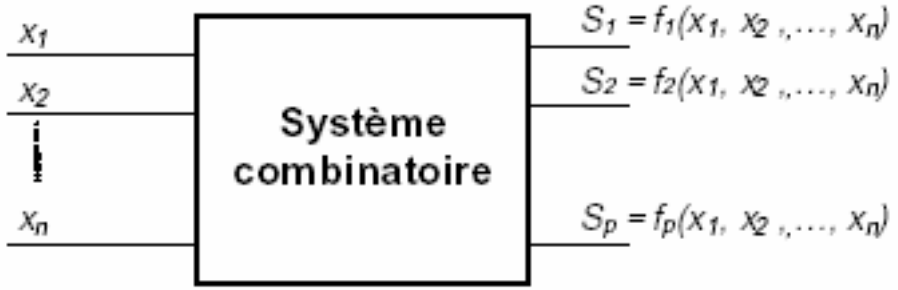


Figure 1 : système combinatoire générant p sorties S_i à partir des n variables d'entrée x_i.

Pour réaliser de tels systèmes, on fait appel à des portes dites « logiques » correspondantes aux fonctions autorisées par l'algèbre de Boole

B. La logique Booléenne

I.1. Variable binaire

On appelle **variable binaire** (ou logique), une variable prenant ses valeurs dans l'ensemble {0, 1}.

Exemple : état d'un interrupteur, d'un bouton poussoir, la présence d'une tension,...

Soit ***a*** la variable associée à l'état d'un bouton poussoir, alors $a = 0$ (faux ou bas) signifie qu'il n'est pas actionné, $a = 1$ (vrai ou haut) signifie qu'il est actionné.

I.2. Equation logique

On appelle **équation logique** une combinaison de plusieurs variables logiques donnant l'état d'une variable dite de sortie associée. Cette combinaison est réalisée à l'aide d'opérations logiques :

Soit xi ($i \in [1, n]$) les variables d'entrée. L'équation $A = f(xi)$ définit l'état de la variable de sortie A.

I.3. Table de vérité

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

B. La logique Booléenne

II. Les Opérateurs

II.1. Opérateur OUI

L'opération (ou opérateur) OUI est dite **unaire** (ne s'applique qu'à une seule opérande). Elle affecte à la variable de sortie l'état logique de la variable d'entrée.

x	S
0	0
1	1

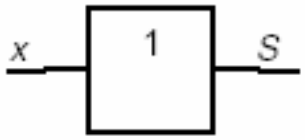
Table de vérité

Equation : x est la l'entrée, S la sortie : $S = x$.

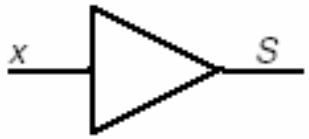
Diagramme de Venn
(représentation ensembliste)

Remarque : les anglo-américains notent H (*High*) le niveau haut et L (*Low*) le niveau bas.

Symbole (norme IEC[1])



Symbole : norme IEEE[2]



B. La logique Booléenne

II.2. Opérateur NON ou inverseur (not- INV)

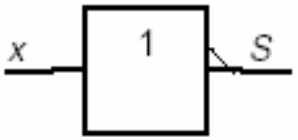
L'opération (ou opérateur) NON est la fonction unaire qui affecte à la variable de sortie l'état complémentaire de la variable d'entrée.

Equation : x est la l'entrée, S la sortie, $S = \overline{x}$ (prononcer « x barre »).

x	S
0	1
1	0

Table de vérité

Symbole (norme IEC)



Symbole (norme IEEE)

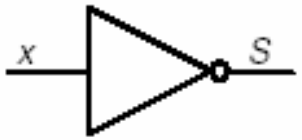


Diagramme de Venn

B. La logique Booléenne

II.3. Opérateur ET (AND)

L'opération ET est le **produit logique**. Le signe est celui de la multiplication (un point), mais on lit « et ». C'est un opérateur **binaire** qui affecte à la variable de sortie l'état 1 si et seulement si les variables d'entrée sont à 1 simultanément.

Equation : x et y les entrées, S la sortie, $S = x.y = xy$.

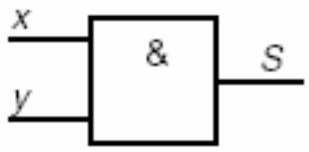
On note aussi l'opération ET par un V retourné :

$x.y = x \wedge y$ (penser à l'intersection d'ensembles).

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Table de vérité

Symbole (norme IEC)



Symbole (norme IEEE)

Diagramme de Venn

B. La logique Booléenne

II.4. Opérateur OU (OR)

L'opération OU est la **somme logique**. Le signe est celui de l'addition (+), mais on lit « ou ». C'est un opérateur binaire qui affecte à la variable de sortie l'état 1 si et seulement si une variable d'entrée est à 1. Cette définition induit directement le symbole ≥1.

Equation : x et y les entrées, S la sortie, $S = x.y = xy$.

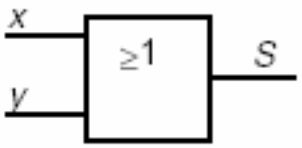
On note aussi l'opération OU par un V :

$x.y = x \vee y$ (penser à l'union d'ensembles).

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Table de vérité

Symbole (norme IEC)



Symbole (norme IEEE)

Diagramme de Venn

B. La logique Booléenne

II.5. Opérateur NON ET (NAND)

Cette fonction logique est le résultat de l'association un NON et d'un ET. C'est un opérateur binaire qui ffecte à la variable de sortie l'état 0 si et seulement si les ariables d'entrée sont à 1 simultanément

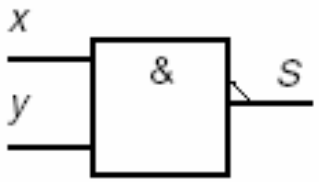
quation : x et y les entrées, S la sortie, $S = \overline{x.y}$.

n note aussi l'opération NAND par une flèche montante :
$= \overline{x.y} = x \uparrow y$ (penser $\wedge$).

x	y	$x \uparrow y$
0	0	1
0	1	1
1	0	1
1	1	0

Table de vérité

Symbole (norme IEC)



Symbole (norme IEEE)

Diagramme de Venn

B. La logique Booléenne

II.6. Opérateur NON OU (NOR)

Cette fonction logique est le résultat de l'association d'un NON et d'un OU. C'est un opérateur binaire qui affecte à la variable de sortie l'état 1 si et seulement si les variables d'entrée sont à 0 simultanément.

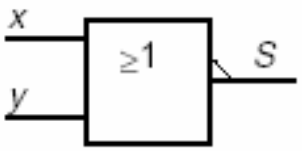
Equation : x et y les entrées, S la sortie, $S = \overline{x+y}$.

On note aussi l'opération NOR par une flèche descendante : $S = \overline{x+y} = x \downarrow y$ (penser $\vee$).

x	y	$x \downarrow y$
0	0	1
0	1	0
1	0	0
1	1	0

Table de vérité

Symbole (norme IEC)



Symbole (norme IEEE)

Diagramme de Venn

B. La logique Booléenne

II.7. Opérateur OU EXCLUSIF (XOR)

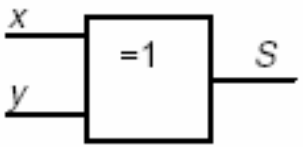
Cet opérateur logique binaire ne prend la valeur 1 que si une seule des entrées est à 1.

Equation : x et y les entrées, S la sortie, $S = x \oplus y$.

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Table de vérité

Symbole (norme IEC)

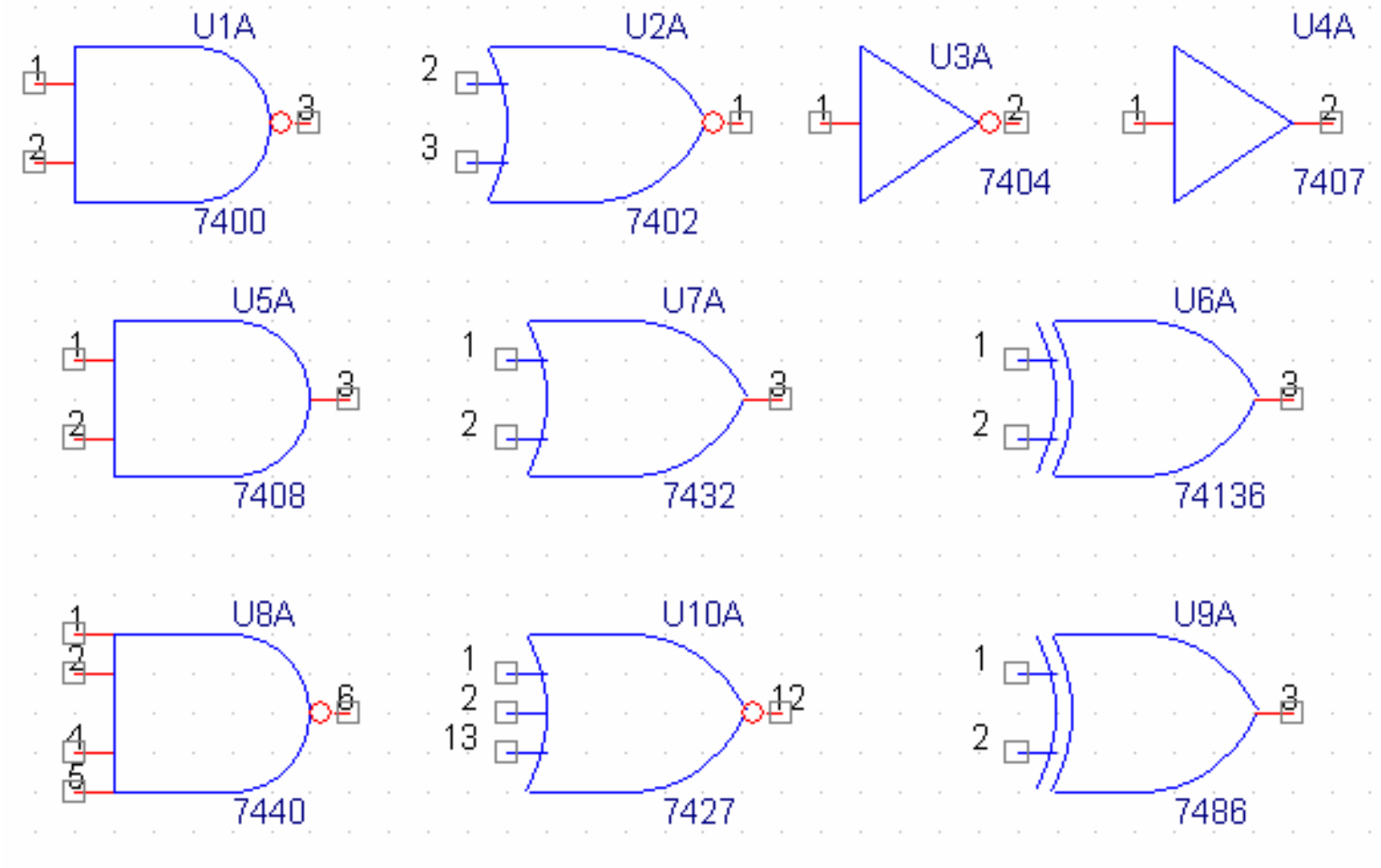


Symbole (norme IEEE)

Diagramme de Venn

B. La logique Booléenne

Toutes les portes élémentaires logiques sont associées à un composants codé



B. La logique Booléenne

III. Les expressions logiques et leurs simplifications

III.1. Propriétés

- Commutativité

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

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

- Associativité

$a + (b + c) = (a + b) + c = a + b + c$ (associativité de l'opération OU)

$(ab)c = a(bc) = abc$ (associativité de l'opération ET)

- Distributivité

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

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

Les parenthèses imposent une priorité supérieure.

B. La logique Booléenne

III.2. Les autres propriétés

$a + 0 = a$	Element neutre	$a \cdot 1 = a$
$a + 1 = 1$	Element absorbant	$a \cdot 0 = 0$
$a + a = a$	Idem potence (redondance)	$a \cdot a = a$
$a + \bar{a} = 1$	Propriété du complément	$a \cdot \bar{a} = 0$
$a + b = b + a$	Commutativité	$a \cdot b = b \cdot a$
$a + b + c = a + (b + c) = (a + b) + c$	Associativité	$a\,b\,c = a\,(b\,c) = (a\,b)\,c$

III.3. Les propriétés déduites

Propriété d'absorbtion : Lorsqu'une somme logique contient un terme et un de ses multiples n peut négliger le multiple. Exemple : $x + x\,y = x$

Règle du multiple du complément $a + \bar{a}\,b = (a + \bar{a})(a + b) = a + b$

B. La logique Booléenne

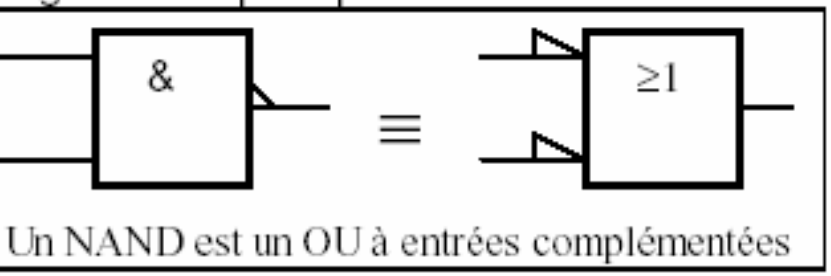
III.3. Propriétés de De Morgan.

But : exprimer les opérateurs ET, OU et NON exclusivement à l'aide d'opérateurs NOR seuls ou NAND seuls. On dit que les opérateurs NOR et NAND sont **universels** ou **complets**.

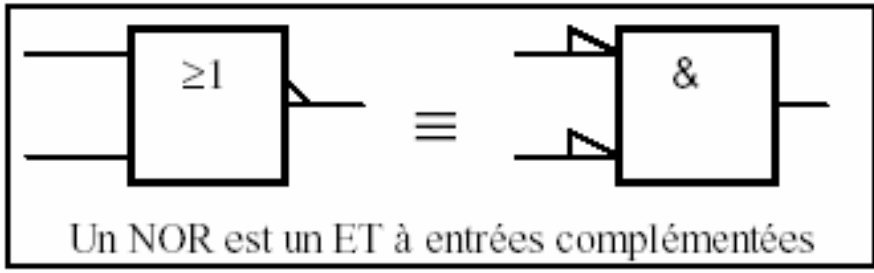
Premier théorème : $\overline{a+b} = \overline{a}\,\overline{b}$ → généralisation $\overline{\sum_{i=1}^{n} a_i} = \prod_{i=1}^{n} \overline{a_i}$ où les les a_i sont les variables, $i \in [1,n]$.

Second théorème : $\overline{ab} = \overline{a} + \overline{b}$ → généralisation $\overline{\prod_{i=1}^{n} a_i} = \sum_{i=1}^{n} \overline{a_i}$ où les les a_i sont les variables, $i \in [1,n]$.

Signification pratique



Un NAND est un OU à entrées complémentées



Un NOR est un ET à entrées complémentées

B. La logique Booléenne

III. Les expressions logiques et leurs simplifications

On considère l'exemple suivant :

$$l = \left(a + \bar{b} + c\right) \cdot \left(a + \bar{c}\right) \cdot \left(\bar{a} + \bar{b}\right)$$

Résoudre par l'algèbre en développant.

Résoudre par l'utilisation des propriétés de De Morgan

Autres exemples :

$P = (a + \bar{b})\ (b + \bar{c})\ (c + \bar{a})$

$Q = (a + b + c)\ (a + \bar{b} + c)\ (a + \bar{b} + \bar{c}$

$R = a\,b\,c + a\,\bar{b}\,\overline{(\bar{a}\ \ \bar{c})}$

$S = \bar{a}\ c\ \overline{(\bar{a}\ b\,d)} + \bar{a}\ b\ \bar{c}\ \bar{d} + a\,\bar{b}\ c$

$T = a\,b\,c + a\,b\,\bar{c} + a\,\bar{b}\ c$

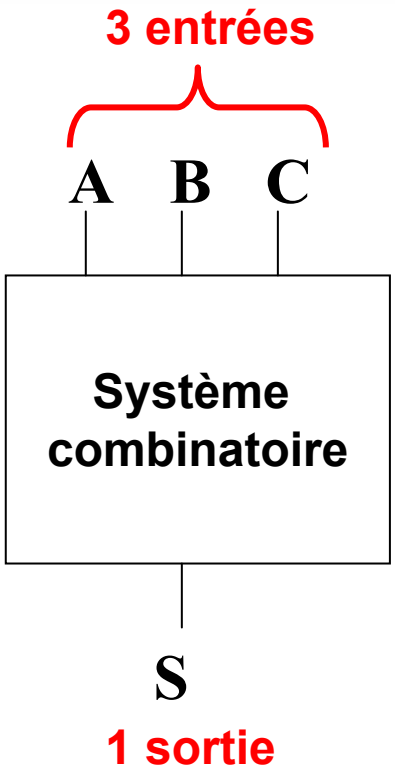
$U = (\bar{a} + b)(a + b + d)\,\bar{d}$

$V = (a + b)(a + c) + (b + c)\ (b + a) + (c + a)\ (c + b)$

B. La logique Booléenne

III. Les tables de vérités et leurs utilisations

Une table de vérité recense l'ensemble des états d'une sortie pour **toutes** les combinaisons possibles les variables d'entrée.



C	B	A	S
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

C. Les tableaux de Karnaugh

I. Définition

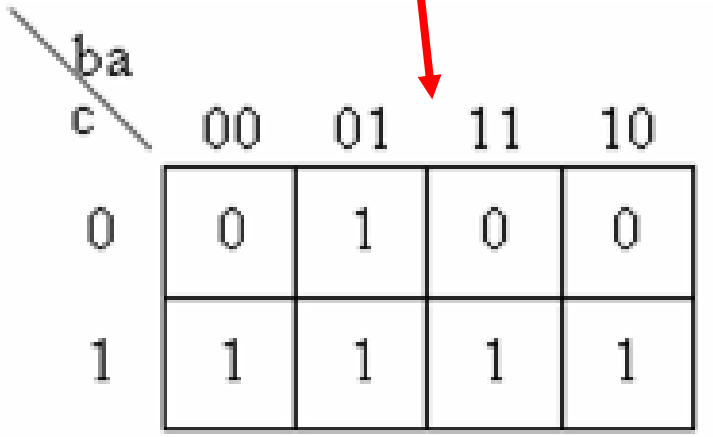
Le tableau de Karnaugh est une représentation différente de toutes les possibilités d'évolution d'un système, sous forme de matrice

- C'est un tableau de 2^n cases, n étant le nombre de variables.

- Dans chacune des cases, on place l'état de la sortie pour les combinaisons d'entrée correspondante.

C	B	A	S
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Attention au codage :
Code GRAY



C. Les tableaux de Karnaugh

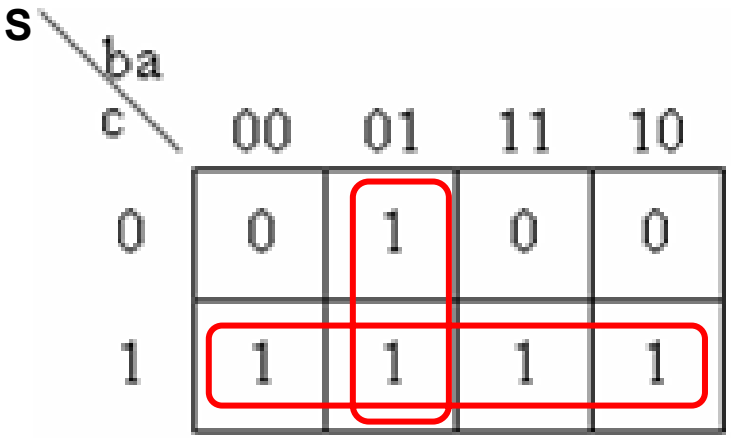
II. Utilisation

Le tableau de Karnaugh permet de faire des simplifications

Les tableaux de KARNAUGH permettent la simplification des équations logiques. Ils com ortent 2^n cases, n étant le nombres de variables d'entrée, organisés selon le code GRAY. (x : 4 variables donnent 16 cases).

Il est ensuite possible de regrouper les cases par 2, 4, 8, 2^n afin d'éliminer les variables ui change d'état dans le regroupement :

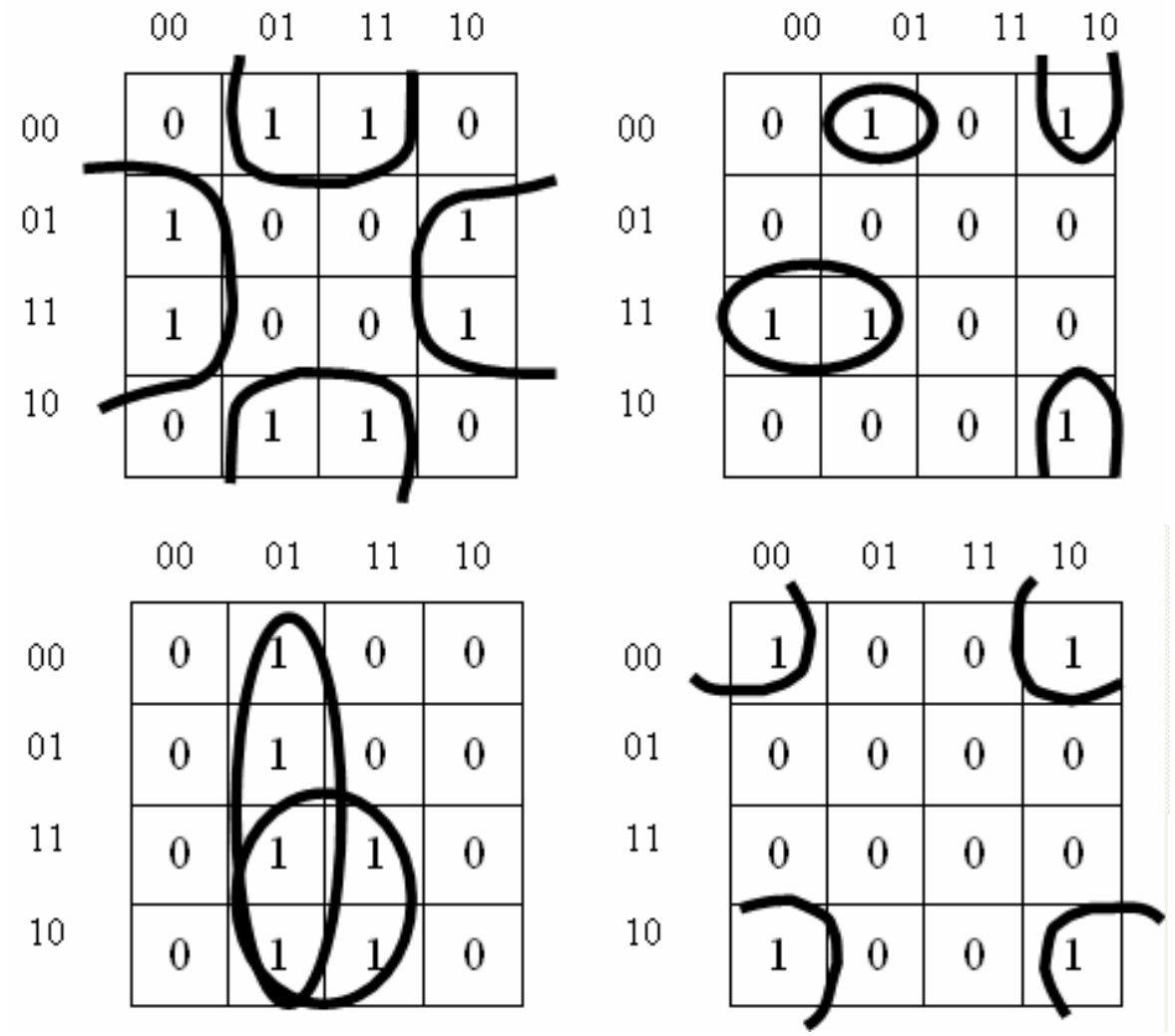
- **un regroupement de 2 cases élimine 1 variable;**
- **un regroupement de 2^x cases élimine x variables.**



$$S = \overline{B}A + C$$

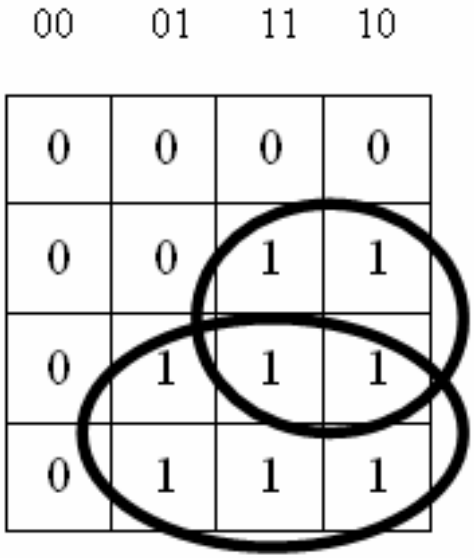
C. Les tableaux de Karnaugh

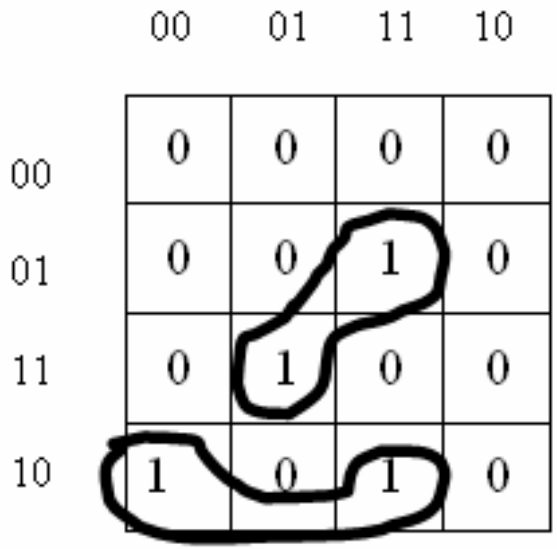
III.1. Des exemples de regroupements autorisés

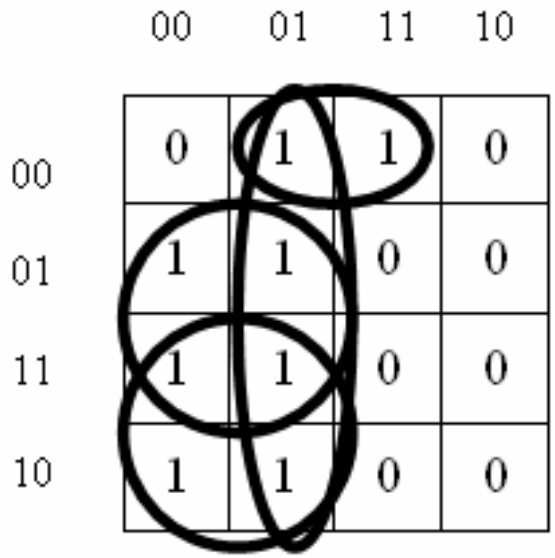


C. Les tableaux de Karnaugh

III.2. Des exemples de regroupements non autorisés ou redondants

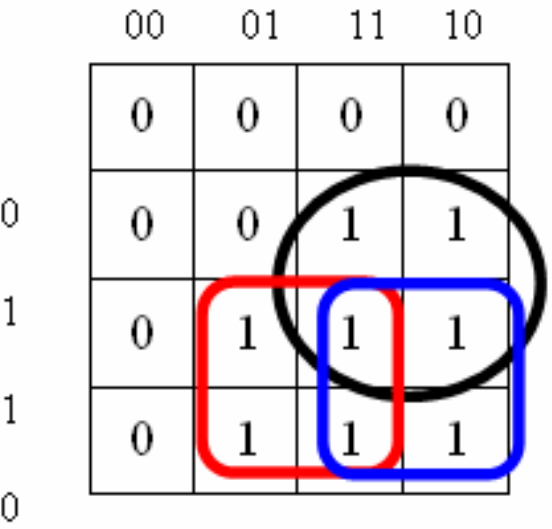


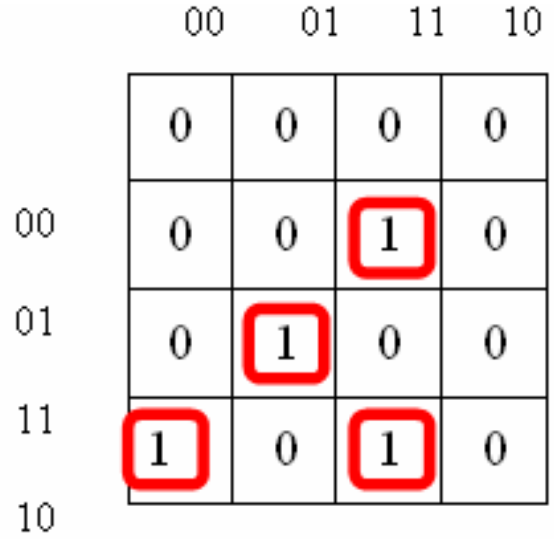


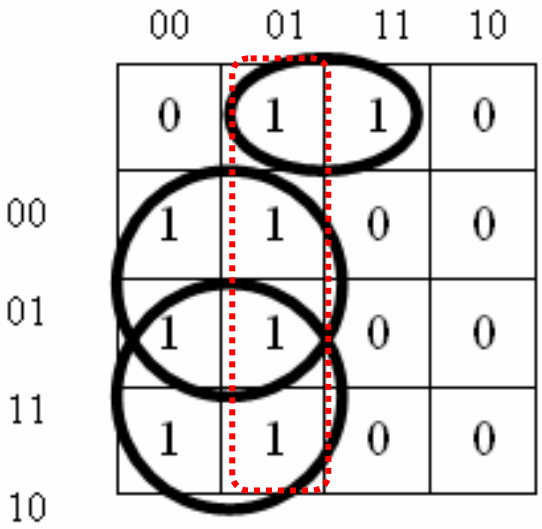


C. Les tableaux de Karnaugh

III.2. Des exemples de regroupements non autorisés ou redondants







IV. Méthode de synthèse des systèmes combinatoires

But : réaliser par un assemblage de portes logiques

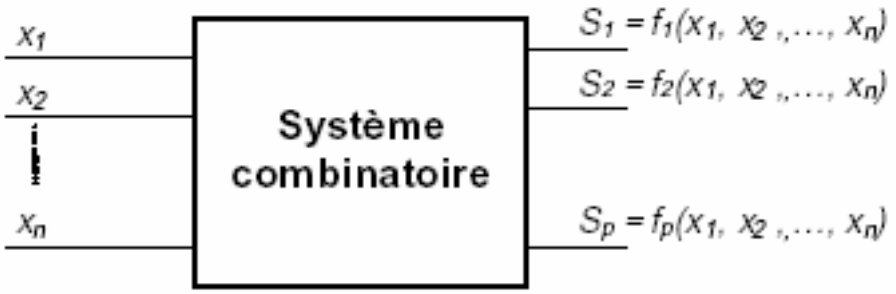


Figure 1 : système combinatoire générant p sorties S_i à partir des n variables d'entrée x_i.

1. **A partir du cahier des charges, identifier les entrées et sorties du système**
2. **Mettre en place la table de vérité décrivant le système**
3. **Trouver les équations simplifiées de chaque sortie en fonction des entrées.**
4. **Réaliser le schéma électrique par l'assemblage de portes en respectant les contraintes du cahier des charges**