Chapitre 04 :
Partie 1: Algèbre de Boole

Introduction: Algèbre de Boole

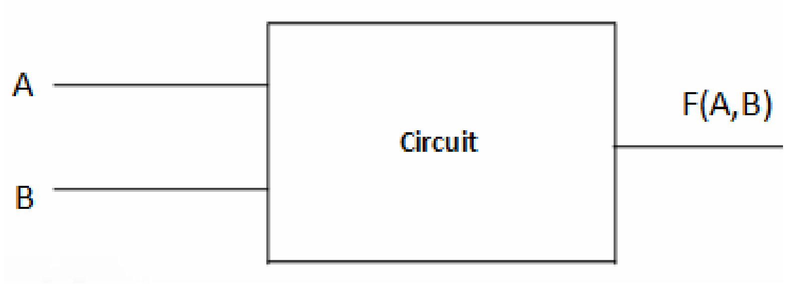
- L'algèbre de Boole (du nom Georges Boole 1915 - 1864) C’est une théorie mathématique proposant de traduire des signaux électriques (à deux états) en expressions mathématiques. C’est un moyen permettant de concevoir des circuits électroniques réalisant des opérations complexes. Dont les signaux élémentaires sont définit par des variables logiques et leur traitement par des fonctions logiques. Des méthodes (table de vérité) permettent de définir les opérations que l'on désire réaliser, et de transcrire le résultat en une expression algébrique.

Introduction: Algèbre de Boole

- **Introduction**
- Aujourd'hui, l'algèbre de Boole trouve de nombreuses applications en informatique et dans la conception des circuits électroniques numériques comme: Les mémoires ,Les circuits de calcul, Les microprocesseurs, Etc…
- Les machines numériques sont constituées d'un ensemble de circuits électroniques.
- Chaque circuit fournit une fonction logique bien déterminée (addition, comparaison ,….).

Algèbre de Boole

- On appelle B l'ensemble constitué de deux éléments appelés valeurs de vérité ***VRAI***, ***FAUX***. *Cet* ensemble est aussi noté *B = {1, 0}* .Sur cet ensemble on peut définir deux lois : ***ET*** *et* ***OU*** *et une* transformation appelée complémentaire, **inversion** ou contraire.



- La fonction F(A,B) peut être : la somme de A et B , ou le résultat de la comparaison de A et B ou une autre fonction

Algèbre de Boole

- Pour concevoir et réaliser ce circuit on doit avoir un modèle mathématique de la fonction réalisée par ce circuit .

- Ce modèle doit prendre en considération le système binaire.

- Le modèle mathématique utilisé est celui de Boole.

Algèbre de Boole

- **Exemple de systèmes à deux états**

Un interrupteur est ouvert ou non ouvert (fermé) Une lampe est allumée ou non allumée (éteinte) Une porte est ouverte ou non ouverte (fermée)

- **Remarque :**

On peut utiliser les conventions suivantes :

OUI → VRAI (true)

NON → FAUX (false)

OUI → 1 (Niveau Haut)

NON → 0 (Niveau Bas)

Algèbre de Boole

- **Définitions et conventions**

Niveau logique : Lorsque on fait l'étude d'un système logique il faut bien préciser le niveau du travail.

Niveau	**Logique positive**	**Logique négative**
H (Hight) haut	**1**	**0**
L (Low) bas	**0**	**1**

Exemple : Logique positive : lampe allumée : 1   (L.Nég:0)

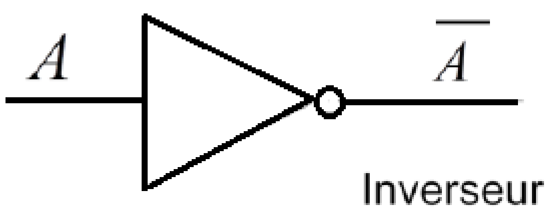
lampe éteinte : 0   (L.Posi:1)

Operateurs logiques de l’algèbre de Boole

- **Définition:**
- On appelle B l'ensemble constitué de deux éléments appelés **valeurs de vérité** **{VRAI, FAUX}.**
- Cet ensemble est aussi noté B = {1, 0}.
- Sur cet ensemble on peut définir deux lois (opérations ou fonctions) : **ET** et **OU** et une transformation appelée **NON**, complémentaire, inversion ou contraire.
- Ces opérations on les appelle les opérations logiques de base.

Opérateurs logiques de base

- **<u>NON</u>** : (Négation) : est un opérateur unaire (une seule variable) qui a pour rôle d'inverser la valeur d'une variable. Tel que Le contraire d'une variable "A" est VRAI si et seulement si *A est FAUX.*
  Le contraire de A est noté $\bar{A}$
- *(lire : A barre)*
- **F(A)= Non A =** $\bar{A}$

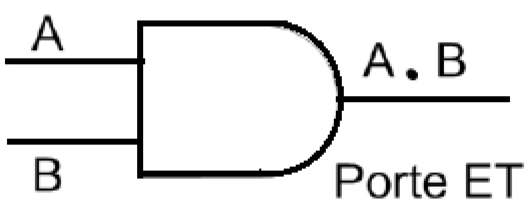


A	$\bar{A}$
0	1
1	0

Opérateurs logiques de base

- **ET (AND)**
- Le **ET** est un opérateur binaire (deux variables) , à pour rôle de réaliser le Produit logique entre deux variables booléennes.

Le **ET** fait la conjonction entre deux variables. Le **ET** Est définipar : **F(A,B)= A . B**



A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

Opérateurs logiques de base

- **OU (OR)**
- Le **OU** est un opérateur binaire (deux variables) , à pour rôle de réaliser la somme logique entre deux variables logiques.
- Le **OU** fait la disjonction

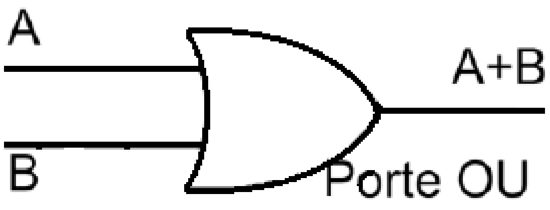
entre deux variables.

- Le **OU** est défini par:
- **F(A,B)= A + B**

(il ne faut pas confondre

avec la somme arithmétique)

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1



Opérateurs logiques de base

- **Remarques**

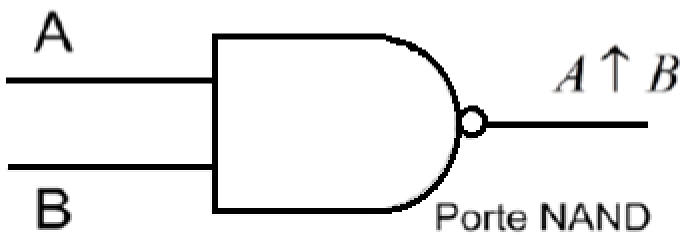
- Dans la définition des opérateurs ET , OU , nous avons juste donné la définition de base avec deux variables logiques.
- L'opérateur ET peut réaliser le produit de plusieurs variables logique (ex : A . B . C . D).
- L'opérateur OU peut aussi réaliser la somme logique de plusieurs variables logiques (ex : A + B + C +D).
- Dans une expression on peut aussi utiliser les parenthèses.

Autres opérateurs logiques

•L'opérateur **NON-ET** (**NAND** en anglais) associe un résultat qui a lui-même la valeur VRAI seulement si au moins l'un des deux opérandes a la valeur FAUX. Il est défini comme suit :

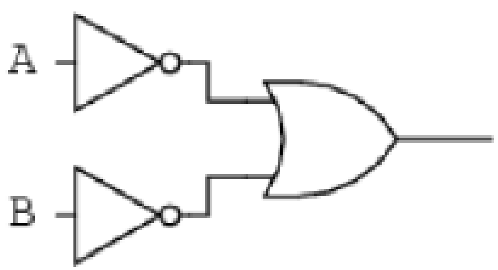
$$F(A,B)=\overline{A\,.\,B}$$

$$F(A,B)=A\uparrow B$$



A	B	$\overline{A\bullet B}$
0	0	1
0	1	1
1	0	1
1	1	0

Or:



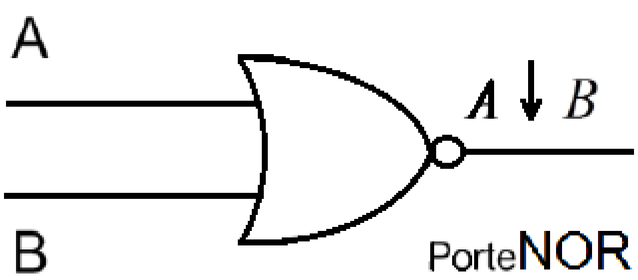
Autres opérateurs logiques

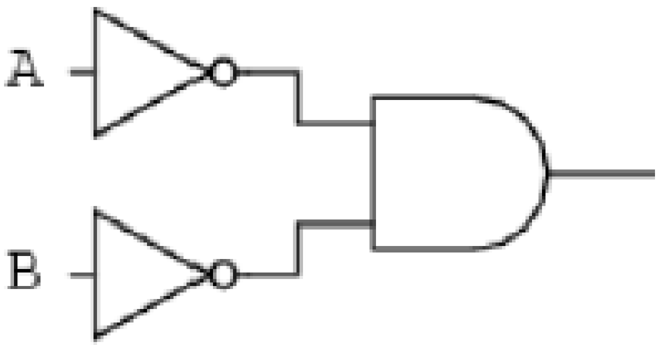
•L'opérateur **NON-OU** (NOR en anglais) associe un résultat qui a lui-même la valeur VRAI seulement si les deux opérandes ont la valeur FAUX. Il est défini comme suit :

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

$$F(A,B) = \overline{A+B}$$

$$F(A,B) = A \downarrow B$$



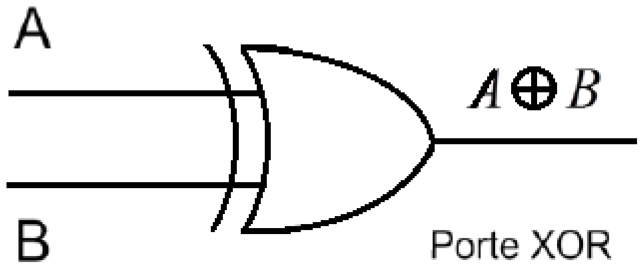


Autres opérateurs logiques

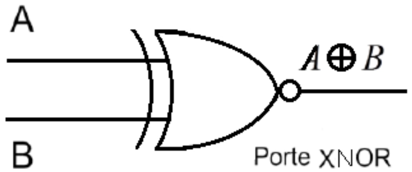
- L'opérateur **OU exclusif**, souvent appelée **XOR** (eXclusive OR), associe un résultat qui a lui-même la valeur VRAI seulement si les deux opérandes ont des valeurs distinctes. Il est défini comme suit : $F(A,B)=A \oplus B$

$$A \oplus B=\bar{A}.B+A.\bar{B}$$

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0



Autres opérateurs logiques

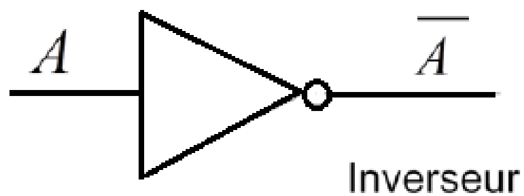


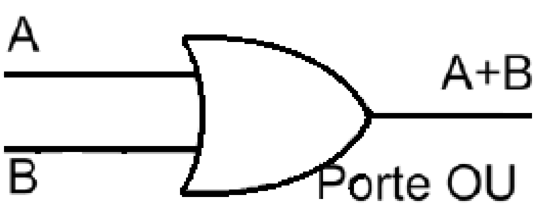
- **Remarque :**
- On peut noter :

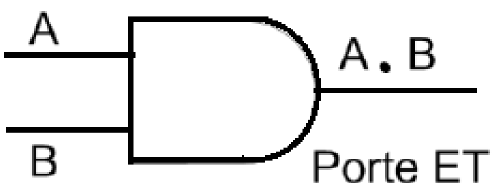
  A non XOR B par **A ⊗ B** et on lit **A XNOR B**.
- L'opérateur ET peut réaliser le produit de plusieurs variables logique (ex : A . B . C . D).
- L'opérateur OU peut aussi réaliser la somme logique de plusieurs variables logiques (ex : A + B + C +D).
- Dans une expression on peut aussi utiliser les parenthèses.
- Les portes ET, OU, NAND, NOR peuvent avoir plus que deux entrées
- Il n'existe pas de OU exclusif à plus de deux entrées

Portes logiques

- Une porte logique est un circuit électronique élémentaire qui Permet de réaliser la fonction d'un opérateur logique de base .



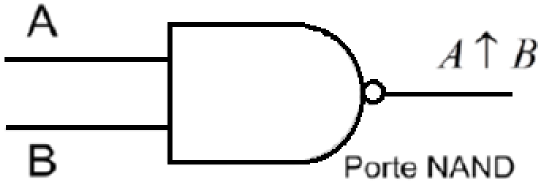


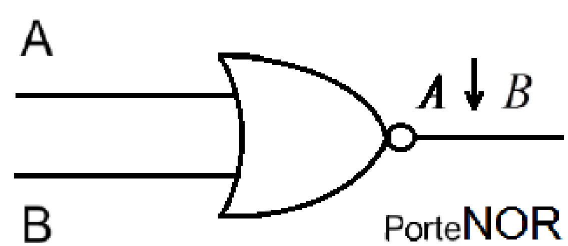


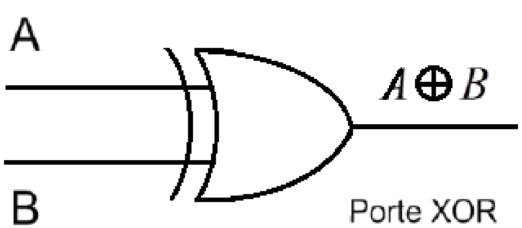
Portes logiques

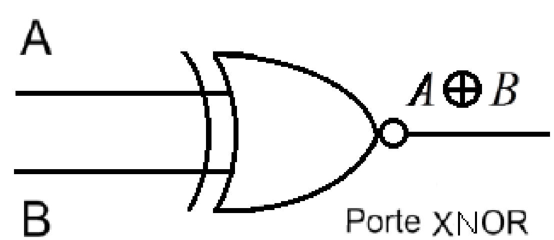
Remarque :

- Les portes ET , OU , NAND , NOR peuvent avoir plus que deux entrées
- Il n’existe pas de OU exclusif à plus de deux entrées









Lois fondamentales de l’algèbre de Boole

PROPRIETES DE L’OPERATEUR « OU »

- ✓ $a + 1 = 1$ élément absorbant
- ✓ $a + 0 = a$ élément neutre
- ✓ $a + a = a$ idempotence
- ✓ $a + \bar{a} = 1$ complémentarité
- ✓ $a + b = b + a$ commutativité
- ✓ $a + b + c = (a + b) + c = a + (b + c)$ associativité

Lois fondamentales de l’algèbre de Boole

PROPRIETES DE L’OPERATEUR « ET »

- ✓ $a \cdot 1 = a$ élément neutre
- ✓ $a \cdot 0 = 0$ élément absorbant
- ✓ $a \cdot a = a$ idempotence
- ✓ $a \cdot \overline{a} = 0$ compléméntarité
- ✓ $a \cdot b = b \cdot a$ commutativité
- ✓ $a \cdot b \cdot c = (a \cdot b) \cdot c = a \cdot (b \cdot c)$ associativité

Lois fondamentales de l’algèbre de Boole

PROPRIETES DE L’OPERATEUR « NON »

$$\overline{\overline{A}} = A$$

$$\overline{A} + A = 1$$

$$\overline{A}.A = 0$$

PROPRIETE DE LA DISTRIBUTIVITE

- ✓ a . (b + c) = a.b + a.c
- ✓ (a + b) . (c + d) = a.c + a.d + b.c + b.d
- ✓ a + (b . c) = (a+b) . (a+c)

Lois fondamentales de l'algèbre de Boole

PROPRIÉTÉS DES OPÉRATEURS «NAND»

$$A \uparrow 0 = 1$$

$$A \uparrow 1 = \overline{A}$$

$$A \uparrow B = B \uparrow A$$

$$(A \uparrow B) \uparrow C \neq A \uparrow (B \uparrow C)$$

PROPRIÉTÉS DES OPÉRATEURS «NOR»

$$A \downarrow 0 = \overline{A}$$

$$A \downarrow 1 = 0$$

$$A \downarrow B = B \downarrow A$$

$$(A \downarrow B) \downarrow C \neq A \downarrow (B \downarrow C)$$

Lois fondamentales de l'algèbre de Boole

- **Remarque 1 :**

Toutes ces formules permettent de simplifier des fonctions logiques, c'est à dire d'éliminer au maximum les opérateurs logiques (sans bien sur modifier la fonction initiale !)

- **Remarque 2:**

NAND et NOR sont des opérateurs universels (complets) c.-à-d. en les utilisant, on peut exprimer n'importe quelle fonction logique, et Pour cela, Il suffit d'exprimer les opérateurs de base (NON, ET, OU) avec des NAND et des NOR.

Lois fondamentales de l'algèbre de Boole

- **Exemple :**

Réalisation des opérateurs de base avec des NOR

$$\overline{A} = \overline{A + A} = A \downarrow A$$

$$A + B = \overline{\overline{A + B}} = \overline{A \downarrow B} = (A \downarrow B) \downarrow (A \downarrow B)$$

$$A.B = \overline{\overline{A.B}} = \overline{\overline{A} + \overline{B}} = \overline{A} \downarrow \overline{B} = (A \downarrow A) \downarrow (B \downarrow B)$$

Lois fondamentales de l’algèbre de Boole

- **Dualité de l’algèbre de Boole**
- Toute expression logique reste vraie si on remplace:
  le ET par le OU, le OU par le ET, le 1 par 0, le 0 par 1.
- **Exemple :**

$$A + 1 = 1 \Rightarrow A\,.\,0 = 0$$

$$A + \bar{A} = 1 \Rightarrow A.\,\overline{A} = 0$$

Lois fondamentales de l'algèbre de Boole

- **Théorème de de-morgane**
- La somme logique complimentée de deux variables est égale au produit des compléments des deux variables.

$$\overline{A + B} = \overline{A} \,.\, \overline{B}$$

- Le produit logique complimenté de deux variables est égal au somme logique des compléments des deux variables.

$$\overline{A . B} = \overline{A} + \overline{B}$$

Lois fondamentales de l'algèbre de Boole

- **Généralisation du théorème de morgane a n variables**

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

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

Lois fondamentales de l'algèbre de Boole

- **Exemple :**
- Soit S à simplifier

$$S = (X + \overline{Y}).(X + Y) + Z.(\overline{X} + Y)$$

Par distributivité	$S = (X + \overline{Y}).X + (X + \overline{Y}).Y + Z.(\overline{X} + Y)$
Par distributivité	$S = X.X + \overline{Y}.X + X.Y + \overline{Y}.Y + Z.\overline{X} + Z.Y$
Par idempotence (X.X = X)	$S = X + \overline{Y}.X + X.Y + \overline{Y}.Y + Z.\overline{X} + Z.Y$
Par complémentarité $Y.\overline{Y} = 0$	$S = X + \overline{Y}.X + X.Y + Z.\overline{X} + Z.Y$
Par identité remarquable (1.X = X)	$S = 1.X + \overline{Y}.X + X.Y + Z.\overline{X} + Z.Y$
Par distributivité	$S = X.(1 + \overline{Y} + Y) + Z.\overline{X} + Z.Y$
Par identité remarquable (sur + puis .)	$S = X + Z.\overline{X} + Z.Y$

Partie 2: Circuit logique

Circuit logique (logigramme)

•Notion d’un circuit logique (logigramme)

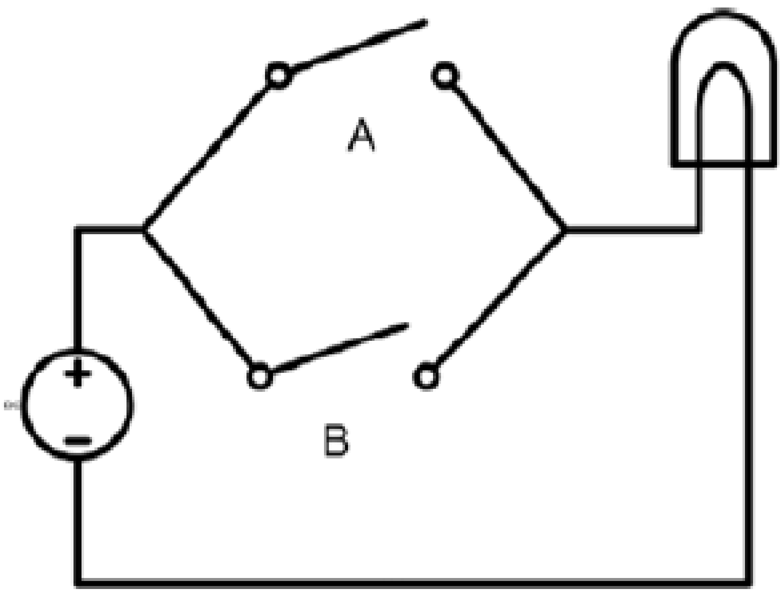
Le lien établi entre l’algèbre de Boole et les circuits logiques date du début du XXe siècle. Ce fut une véritable révolution dont nous connaissons largement les conséquences aujourd’hui.

Il s’agissait initialement d’une application aux circuits à relais (sorte d’interrupteurs).

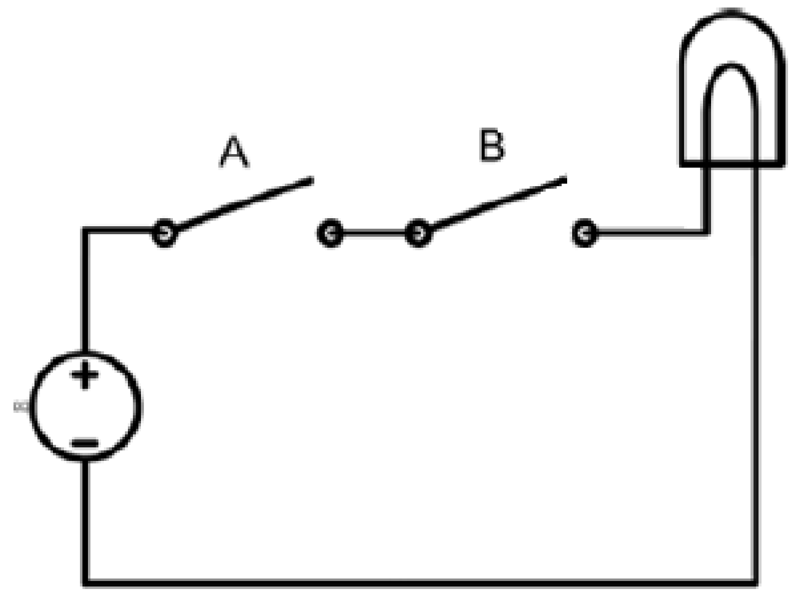
Si des relais obéissaient à la même commande (variable), alors une fonction logique pouvait exprimer son fonctionnement général :

Circuit logique (logigramme)

- **Exemples d’application aux circuits à relais:**

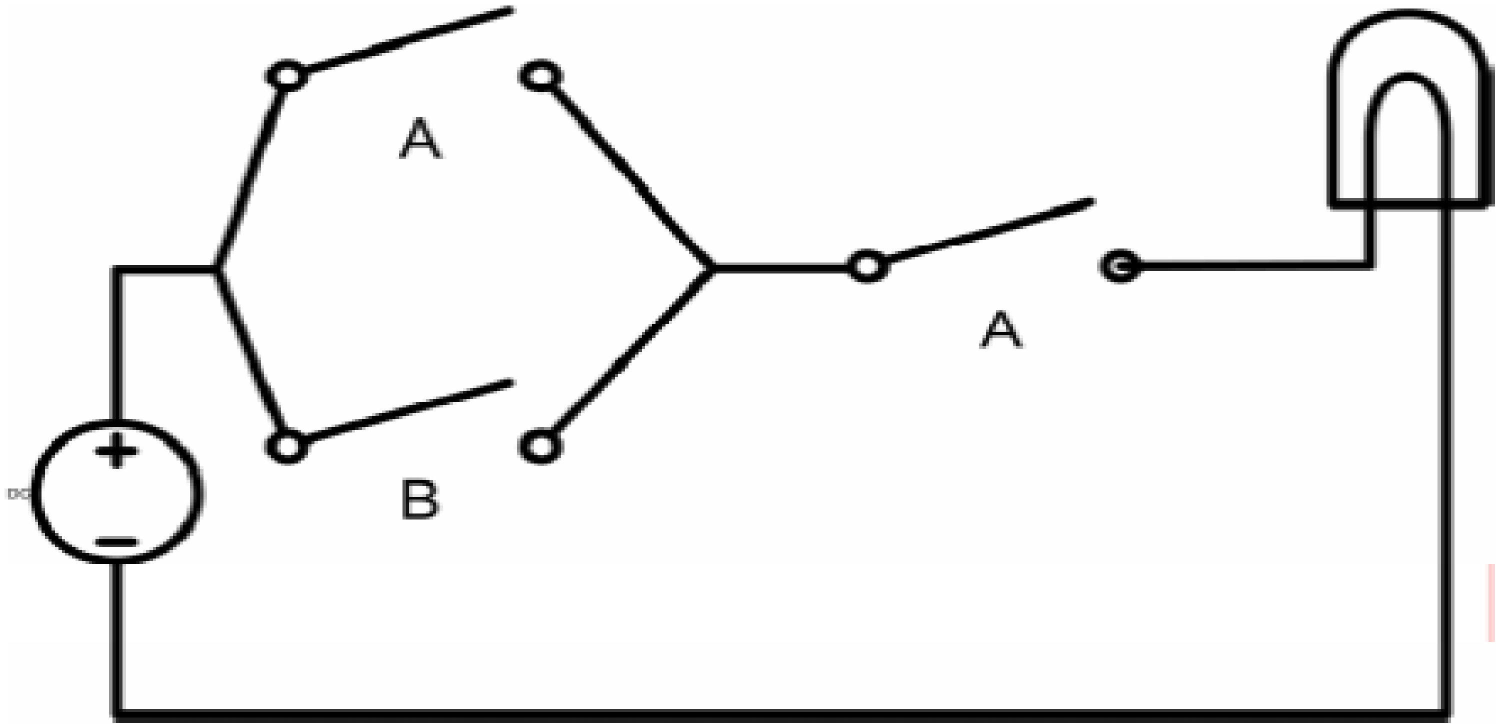


La lampe s’allume si A OU B est fermé



La lampe s’allume si A ET B sont fermé

Circuit logique (logigramme)



La lampe s’allume si (A OU B) ET A est fermé;
Donc la lampe s’allume si A est fermé puisque
$A(A+B)=A$

Circuit logique

Un circuit logique est la traduction de la fonction logique en un schéma électronique.

Le principe consiste à remplacer chaque opérateur logique par la porte logique qui lui correspond.

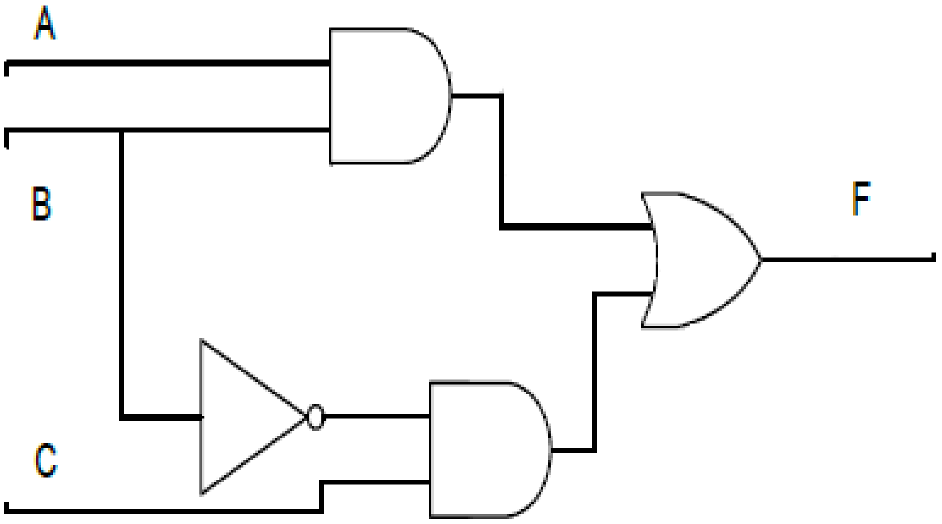
Les portes logiques sont les éléments de base à l'aide desquels nous pouvons exprimer toute fonction logique.

Les variables d'une fonction logique deviennent les **entrées** du circuit, ce circuit donnant en **sortie** la valeur de la fonction logique suivant la valeur des entrées.

Circuit logique

- On peut ainsi connecter des portes logiques les unes aux autres pour réaliser une fonction logique.
  À l'inverse, trouver la fonction logique réalisée par un circuit nous permettra de le manipuler en vue de simplifications éventuelles.
- **Exemple :**

$$F(A,B,C) = A.B + \overline{B}.C$$



Fonction logique

- **Fonction** qui relie **N** variables logiques avec un ensemble d'opérateurs logiques de base.
- Il existe trois **opérateurs** de base : NON , ET , OU.
- La valeur d'une fonction logique est égale à 1 ou 0 selon les valeurs des variables logiques.
- Si une fonction logique possède N variables logiques → 2^n combinaisons → la fonction possède 2^n valeurs.
- Les 2^n combinaisons sont représentées dans une table qui s'appelle table de vérité (TV).

Fonction logique

- **Exemple d'une fonction logique**

$$F(A,B,C)=\overline{A}.\overline{B}.C+\overline{A}.B.C+A.\overline{B}.C+A.B.C$$

La fonction possède :

3 variables 2^3 combinaisons

Une table de vérité

Tableau représentant les valeurs que prend une expression booléenne pour chaque combinaison possible de ses entrées.

A	B	C		F
0	0	0		0
0	0	1		1
0	1	0		0
0	1	1		1
1	0	0		0
1	0	1		1
1	1	0		0
1	1	1		1

Fonction logique

- **Définition textuelle d'une fonction logique**

- Généralement la définition du fonctionnement d'un système est donnée sous un format textuel.
- Ainsi, pour faire l'étude et la réalisation d'un tel système on doit avoir son modèle mathématique (fonction logique)
- → il faut tirer (déduire) la fonction logique à partir de la description textuelle.

Fonction logique

- **Exemple :**

- Une serrure de sécurité s’ouvre en fonction de trois clés.
- Le fonctionnement de la serrure est défini comme suite :
- La serrure est ouverte si au moins deux clés sont utilisées.
- La serrure reste fermée dans les autres cas.

- Donner le schéma du circuit qui permet de contrôler l’ouverture de la serrure ?

Fonction logique

- Le système possède trois entrées : chaque entrée représente une clé.
- On va correspondre à chaque clé une variable logique : la clé1 → A, la clé2 → B, la clé3 → C

- Si la clé1 est utilisée alors la variable A=1 sinon A =0
- Si la clé2 est utilisée alors la variable B=1 sinon B =0
- Si la clé3 est utilisée alors la variable C=1 sinon C =0

Fonction logique

- Le système possède une seule sortie qui correspond à l’état de la serrure (ouverte ou fermé).
- On va correspondre une variable **S** pour designer la **sortie**
- **S=1** si la serrure est **ouverte**,
- **S=0** si elle est **fermée** ,

$$S=F(A,B,C)=\begin{cases} F(A,B,C)=1 \text{ si au moins deux clés sont introduites} \\ F(A,B,C)=0 \text{ sinon .} \end{cases}$$

Fonction logique

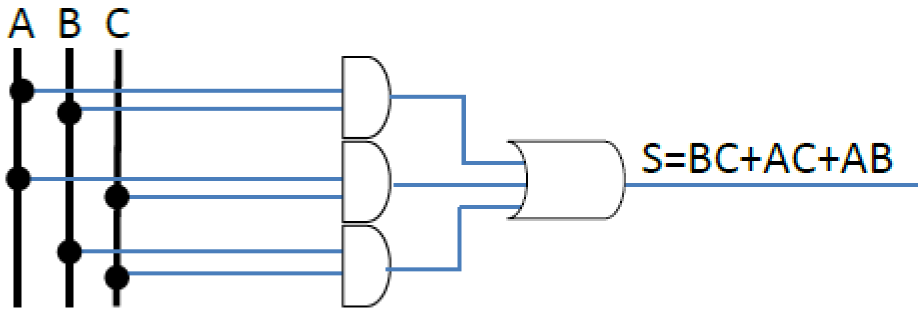
- Ainsi, la table de vérité du système sera comme suit :
- D'après la table de vérité on a :

S=F(A, B, C)= $\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

S=BC+AC+AB

- Alors le circuit Correspondant

est le suivant :



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	**1**
1	1	1	**1**

Fonction logique

Forme canonique d’une fonction logique

- On appel forme canonique d’une fonction:

  La forme où chaque terme de la fonction comporte toutes les variables.

  On l’appelle canonique car elle est unique pour chaque fonction (Une expression canonique n’est cependant pas optimale).

Fonction logique

On identifie, selon le principe de dualité propre à l'algèbre de Boole, deux formes canoniques :

1- La somme de produits (appelé forme canonique disjonctive ou la somme des minterms). Une somme de produits est sous forme canonique si toutes les variables apparaissent dans tous les termes des produits qui la composent.

2- Le produit de sommes (appelé forme canonique conjonctive ou produit de maxterms). Un produit de sommes est sous forme canonique si toutes les variables apparaissent dans tous les termes de sommes qui le composent.

Fonction logique

Ligne	A	B	C	F	*Minterms* m_i	*Maxterms* M_i
0	0	0	0	0	$\bar{A}\bar{B}\bar{C}$	$A+B+C$
1	0	0	1	0	$\bar{A}\bar{B}C$	$A+B+\bar{C}$
2	0	1	0	0	$\bar{A}B\bar{C}$	$A+\bar{B}+C$
3	0	1	1	1	$\bar{A}BC$	$A+\bar{B}+\bar{C}$
4	1	0	0	0	$A\bar{B}\bar{C}$	$\bar{A}+B+C$
5	1	0	1	1	$A\bar{B}C$	$\bar{A}+B+\bar{C}$
6	1	1	0	1	$AB\bar{C}$	$\bar{A}+\bar{B}+C$
7	1	1	1	1	ABC	$\bar{A}+\bar{B}+\bar{C}$

Fonction logique

- On en déduit l'expression de F sous forme canonique disjonctive (somme des minterms) et ses écritures condensées :

$$F(A,B,C) = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$F(A, B, C) = m_3 + m_5 + m_6 + m_7$$

- On en déduit l'expression de F sous forme canonique conjonctive (produit de maxterms) et ses écritures condensées :

F	*Minterms* m_i
0	$\bar{A}\bar{B}\bar{C}$
0	$\bar{A}\bar{B}C$
0	$\bar{A}B\bar{C}$
1	$\bar{A}BC$
0	$A\bar{B}\bar{C}$
1	$A\bar{B}C$
1	$AB\bar{C}$
1	ABC

$$F(A,B,C) = (A+B+C)(A+B+\bar{C})(A+B+C)(\bar{A}+B+C)$$

$$F(A, B, C) = M_0.M_1.M_2.M_4$$

Fonction logique

- On en déduit l'expression de F sous forme canonique disjonctive (somme des minterms) et ses écritures condensées :

$$F(A,B,C) = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$F(A, B, C) = m_3 + m_5 + m_6 + m_7$$

F	Maxterms M_i
0	$A + B + C$
0	$A + B + \bar{C}$
0	$A + \bar{B} + C$
1	$A + \bar{B} + \bar{C}$
0	$\bar{A} + B + C$
1	$\bar{A} + B + \bar{C}$
1	$\bar{A} + \bar{B} + C$
1	$\bar{A} + \bar{B} + \bar{C}$

- On en déduit l'expression de F sous forme canonique conjonctive (produit de maxterms) et ses écritures condensées :

$$F(A,B,C) = (A + B + C)(A + B + \bar{C})(A + B + C)(\bar{A} + B + C)$$

$$F(A, B, C) = M_0.M_1.M_2.M_4$$

Fonction logique

Il existe une autre représentation des formes canoniques d'une fonction logique, cette représentation est appelée forme numérique.

R ou Σ : pour indiquer la forme disjonctive.

P ou Π : pour indiquer la forme conjonctive.

Si on reprend la fonction de l'exemple précédent :

$$F(A,B,C) = \sum(3,5,6,7) = \mathrm{R}(011,101,110,111)$$
$$= \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC.$$

$$F(A,B,C) = \prod(0,1,2,4) = \mathrm{P}(000,001,010,100)$$
$$= (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+B+C)$$

Fonction logique

Remarque :

- On peut toujours ramener n'importe qu'elle fonction logique à l'une des formes canoniques.
- Cela revient à rajouter les variables manquantes dans les termes qui ne contiennent pas toutes les variables (les termes non canoniques), en utilisant les règles de l'algèbre de Boole:
  - Multiplier un terme avec une expression qui vaut 1.
  - Additionner à un terme avec une expression qui vaut 0.
  - Par la suite faire la distribution

Fonction logique

- La première et la deuxième forme canonique sont équivalentes.
- Il est possible de passer d'une expression canonique disjonctive à une expression canonique conjonctive (et vice et versa) en considérant les termes manquants dans la duale.

$$A\overline{C} + \overline{B}C + \overline{A}B = A\underbrace{(B+\overline{B})}\overline{C} + \underbrace{(A+\overline{A})}\overline{B}C + \overline{A}B\underbrace{(C+\overline{C})}$$

Fonction logique

- **Simplification des fonctions logiques**
- **Pourquoi ?**
- Utiliser le moins de composants possibles ;
- Simplifier au maximum le schéma de câblage en réduisant le nombre de portes logiques utilisées → réduction du coût du circuit.
- Il faut donc trouver la forme minimale de l’expression logique considérée, et pour cela on doit :
- Réduire le nombre de terme dans une fonction ;
- Réduire le nombre de variable dans un terme.

Fonction logique

Trois méthodes

- Algébrique (en utilisant des propriétés et des théorèmes)
- Graphique (tableaux de Karnaught; ...).
- Programmable (méthode de quine-mac cluskey)

Simplification algébrique

- Cette méthode n’a pas une démarche bien spécifique, son principe consiste à appliquer les règles de l’algèbre de Boole afin d’éliminer des variables ou des termes.

Fonction logique

Ainsi, Cette technique de simplification repose sur l’utilisation des théorèmes fondamentaux et des propriétés de l’algèbre de Boole.

Après la recherche de l’expression algébrique de la fonction, l’étape suivante consiste à minimiser le nombre de termes dans une fonction afin d’obtenir un circuit plus petit donc plus facile à construire avec un coût plus réduit.

La simplification algébrique repose sur beaucoup d'astuce, quand la fonction est plus complexe (au-delà de trois variables) cette méthode de simplification devient peu évidente.

Fonction logique

- Supprimer les associations de termes multiples.
- Mettre en facteur des variables pour éliminer plusieurs termes.
- Mettre en facteur des variables pour faire apparaître des termes inclus.
- Ajouter un terme qui existe déjà à une expression logique.

Fonction logique

- Quelques règles fondamentales :
- Règle 1 : Dans une somme on peut éliminer tous les multiples d'un terme fondamental. X+XY= X .
- Règle 2 : (Absorption) : Dans la somme d'un terme et d'un multiple de son complément, on peut éliminer le complément. $X + \bar{X}Y = X + Y$
- Règle 3 : Regrouper des termes à l'aide des règles de l'algèbre de Boole.

$$ABC + AB\bar{C} + A\bar{B}CD = AB(C + \bar{C}) + A\bar{B}CD$$

$$= AB + A\bar{B}CD = A(B + \bar{B}(CD))$$

$$= A(B + CD) = AB + ACD$$

Fonction logique

- Règle 4 : Rajouter un terme déjà existant à une expression

$$A\,B\,C + \overline{A}BC + A\overline{B}C + AB\overline{C} =$$
$$= ABC + \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C}$$
$$= BC + AC + AB$$

- Règle 5 : Il est possible de supprimer un terme superflu (un terme en plus), c’est-à-dire déjà inclus dans la réunion des autres termes.

$$
\begin{aligned}
F(A,B,C) &= A\,B + \overline{B}C + AC \\
&= AB + \overline{B}C + AC\,(B + \overline{B}) \\
&= AB + \overline{B}C + ACB + A\overline{B}C \\
&= AB\,(1 + C) + \overline{B}C\,(1 + A) \\
&= AB + \overline{B}C
\end{aligned}
$$

Fonction logique

- Règle 6 : Il est préférable de simplifier la forme canonique ayant le nombre de termes minimum

$$F(A,B,C) = R(2,3,4,5,6,7)$$

$$\overline{F(A,B,C)} = R(0,1) = \overline{A}.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C$$

$$= \overline{A}.\overline{B}(\overline{C} + C)$$

$$= \overline{A}.\overline{B} = \overline{A + B}$$

$$F(A,B,C) = \overline{\overline{F(A,B,C)}} = \overline{\overline{A + B}} = A + B$$

Fonction logique

- **Exemple 1:** Simplifier l'expression suivantes en utilisant les règles de l'algèbre de Boole.

$$
\begin{aligned}
F(A,B,C) &= (A+B+C).(\overline{A}+B+C)+A.B+B.C \\
&= [A.\overline{A}+(B+C)]+A.B+B.C \\
&= B+C+A.B+B.C \\
&= B.(1+A)+C.(1+B) \\
&= B+C
\end{aligned}
$$

Fonction logique

- **Exemple 2:** Simplifier l'expression suivantes en utilisant les règles de l'algèbre de Boole.

$$F(A,B,C) = \overline{\overline{A+B}+\overline{\overline{\overline{A}+\overline{C}}}} + \overline{A+C}$$

$$= (A+B).(\overline{A}+\overline{C}) + \overline{A}.\overline{C}$$

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

$$= \overline{C}.(A+\overline{A}+B) + \overline{A}.B \quad \text{Car } A+\overline{A}=1 \text{ et } 1+B=1$$

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

Fonction logique

- **Exemple 3:** Simplifier l'expression suivantes en utilisant les règles de l'algèbre de Boole.

$$F(A,B,C,D) = (\overline{A}.B + A.B + A.\overline{B}).(C.\overline{D} + \overline{C}.\overline{D}) + \overline{C}.D.(\overline{A}.B + A.B)$$

$$= [B.(A + \overline{A}) + A.(B + \overline{B})].[\overline{D}.(C + \overline{C})] + \overline{C}.D.[B.(A + \overline{A})]$$

$$= (A + B).\overline{D} + B.\overline{C}.D$$

$$= A.\overline{D} + B.(\overline{D} + D.\overline{C}) \qquad X + \bar{X}Y = X + Y$$

$$= A.\overline{D} + B.\overline{D} + B.\overline{C}$$

Simplification par:

la table de Karnaugh

Simplification par table de **Karnaugh**

Méthode de simplification de Karnaugh

Cette méthode est souvent utilisée pour remédier aux difficultés que l'on rencontre dans la méthode algébrique. Elle est très intéressante lorsque le nombre de variables de la fonction ne dépasse pas 6.

Au-delà de six variables, elle est difficile à utiliser.

Elle est basée sur **l'inspection visuelle** de tableaux disposés de façon que deux cases adjacentes en ligne et en colonne ne diffèrent que par l'état d'une variable et une seule.

Simplification par table de **Karnaugh**

- **Les termes adjacents**

Examinons l'expression suivante : $A \,.\, B + A \,.\, \overline{B}$

- Les deux termes possèdent les même s variables.
- La seule différence est l'état de la variable B qui change.
- Si on applique les règles de simplification on obtient :

$$AB + A\overline{B} = A(B + \overline{B}) = A$$

- Ces termes sont dites adjacents.

Simplification par table de **Karnaugh**

- **Exemple de termes adjacents**
- Les termes suivant sont adjacents:

$$\overline{A}.B + A.B = B$$

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

$$A.B.C.D + A.B.\overline{C}.D = A.B.D$$

- Les termes suivant ne sont pas adjacents:

$$A.B + \overline{A}.\overline{B}$$

$$A.B.C + A.\overline{B}.\overline{C}$$

$$A.B.C.D + \overline{A}.\overline{B}.C.D$$

Simplification par table de **Karnaugh**

- **Description de la table de karnaugh**

- La méthode de Karnaugh se base sur la règle précédente.
- La méthode consiste à mettre en évidence par une méthode graphique (un tableau) tous les termes qui sont adjacents (qui ne différent que par l'état d'une seule variable).
- La méthode peut s'appliquer aux fonctions logiques de 2,3,4,5 et 6 variables.
- Un tableau de Karnaugh comportent 2^n cases (N est le nombre de variables).

Simplification par table de **Karnaugh**

A \ B	0	1
0		
1		

Tableau à 2 variables

C \ AB	00	01	11	10
0				
1				

Tableaux à 3 variables

CD \ AB	00	01	11	10
00				
01				
11				
10				

Tableau à 4 variables

Simplification par table de **Karnaugh**

Par exemple, la case foncée dans le tableau suivant correspond au minterme m1 représentant:

(x,y,z,t)=(0,0,0,1)

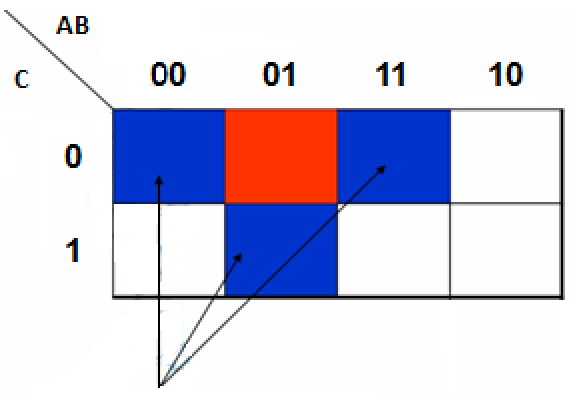
ce qui donne :

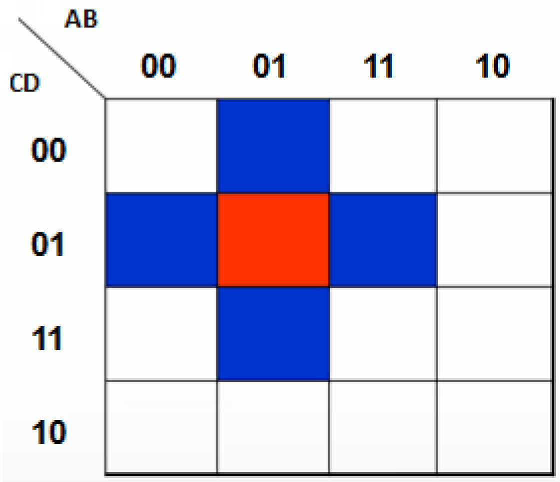
$\boldsymbol{m1 = \overline{}\,t}$

zt \ xy	00	01	11	10
00	m_0	m_4	m_{12}	m_8
01	m_1	m_5	m_{13}	m_9
11	m_3	m_7	m_{15}	m_{11}
10	m_2	m_6	m_{14}	m_{10}

Simplification par table de **Karnaugh**

- Dans un tableau de karnaugh , chaque case possède un certain nombre de cases adjacentes.





- Les trois cases bleues sont des cases adjacentes à la case rouge

Simplification par table de **Karnaugh**

Passage de la table de vérité à la table de Karnaugh

- Pour chaque combinaison qui représente un min terme lui correspond une case dans le tableau qui doit être mise à 1 .

- Pour chaque combinaison qui représente un max terme lui correspond une case dans le tableau qui doit être mise à 0

- Lorsque on remplit le tableau , on doit soit prendre les mintermes ou les maxtermes

Simplification par table de **Karnaugh**

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	1
1	1	1	1

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

Simplification par table de **Karnaugh**

Passage de la forme canonique à la table de Karnaugh

- Si la fonction logique est donnée sous la première forme canonique (disjonctive), alors sa représentation est directe : pour chaque terme lui correspond une seule case qui doit être mise à 1.

- Si la fonction logique est donnée sous la deuxième forme canonique (conjonctive), alors sa représentation est directe : pour chaque terme lui correspond une seule case qui doit être mise à 0 .

Simplification par table de **Karnaugh**

- **Exemple**

- F1(A, B, C) = ∑(1,2,5,7)

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

- F2(A, B, C) = ∏ (0,2,3,6)

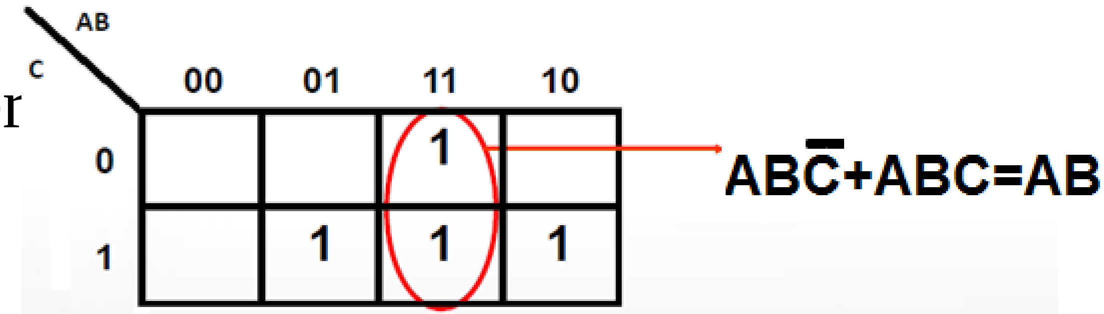
C \ AB	00	01	11	10
0	0	0	0	
1		0		

Simplification par table de **Karnaugh**

- **Méthode de simplification (Exemple : 3 variables)**
- L'idée de base:

est d'essayer de regrouper les cases adjacentes qui comportent des 1

(rassembler les termes adjacents).



- Essayer de faire des regroupements avec le maximum de cases (16,8,4 ou 2)
- Dans notre exemple on peut faire uniquement des regroupements de 2 cases .

Simplification par table de **Karnaugh**

Puisque il existe encore des cases qui sont en dehors d'un regroupement on refait la même procédure :

former des regroupements.

- Une case peut appartenir à plusieurs regroupements

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

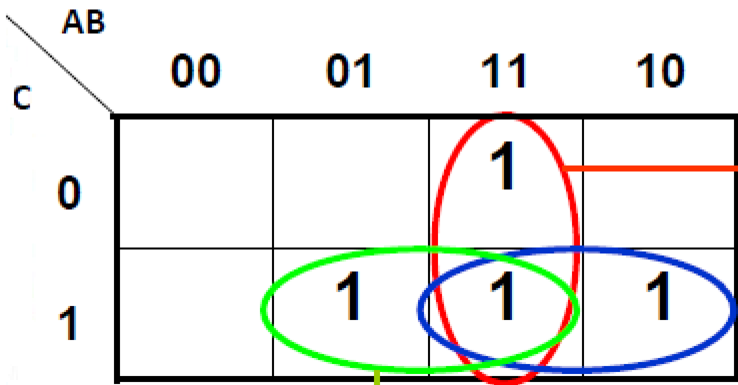
$$AB\overline{C} + ABC = AB$$

$$ABC + A\overline{B}C = AC$$

Simplification par table de **Karnaugh**

- On s'arrête lorsque **il y a plus de 1** en dehors des regroupements
- La fonction final est égale à la réunion (somme) des termes après simplification.

$$F(A, B,C) = AB + AC + BC$$



$$AB\overline{C} + ABC = AB$$

$$ABC + A\overline{B}C = AC$$

$$\overline{A}BC + ABC = BC$$

Simplification par table de **Karnaugh**

Résumé pour simplifier une fonction par la table de karnaugh il faut suivre les étapes suivantes :

1. Remplir le tableau à partir de la table de vérité ou à partir de la forme canonique.
2. Faire des regroupements de 16,8,4,2,1 cases (une puissance de 2) on se basant sur le regroupement des bits a 1 adjacents en blocs rectangulaires ou carres et le regroupement doit contenir le maximum possible de 1, tel que les même termes peuvent participer à plusieurs regroupements, L'intersection entre les regroupent est autorisée, par contre L'inclusion est interdite) .

Simplification par table de **Karnaugh**

4. Dans un regroupement : Qui contient un seul terme on ne peut pas éliminer de variables.

- Qui contient deux termes on peut éliminer **<u>une variable</u>** (celle qui change d'état).
- Qui contient 4 termes on peut éliminer **<u>2 variables</u>**.
- Qui contient 8 termes on peut éliminer **<u>3 variables etc….</u>**

5. Une case a 1 doit appartenir a <u>au moins</u> un groupement

6. L'expression logique finale est la réunion (la somme) des groupements correspondant aux blocs obtenus après simplification et élimination des variables qui changent d'état a l'intérieur du bloc.

Simplification par table de **Karnaugh**

- **Exemple 1 :**
- 3 variables

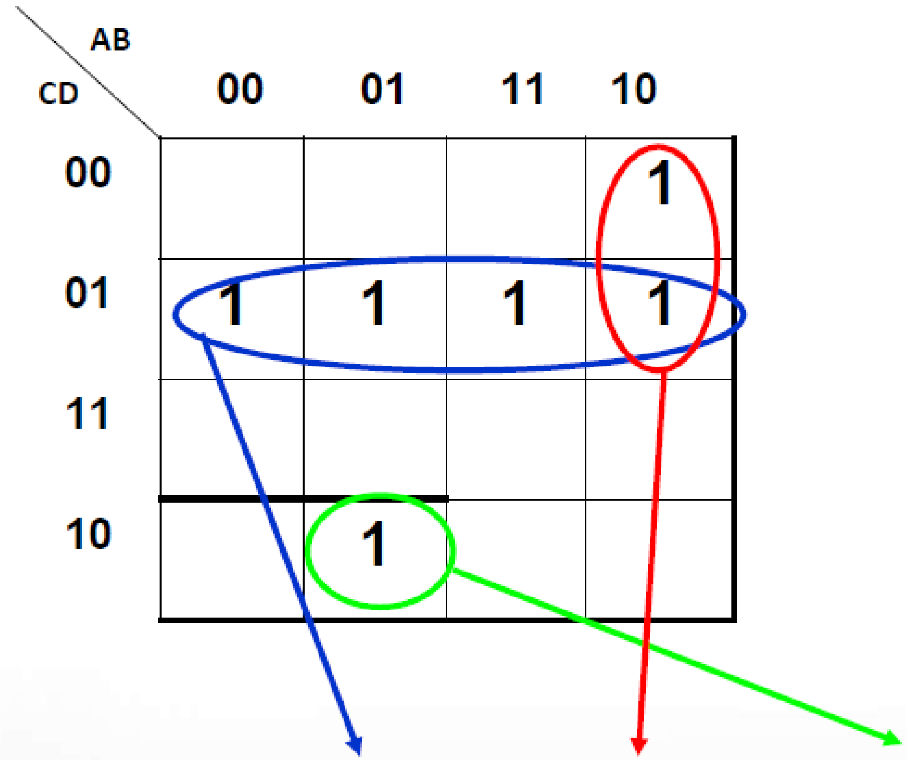
C \ AB	00	01	11	10
0			1	
1	1	1	1	1

$$F(A,B,C) = C + AB$$

Simplification par table de **Karnaugh**

- **Exemple 2:**

  4 variables

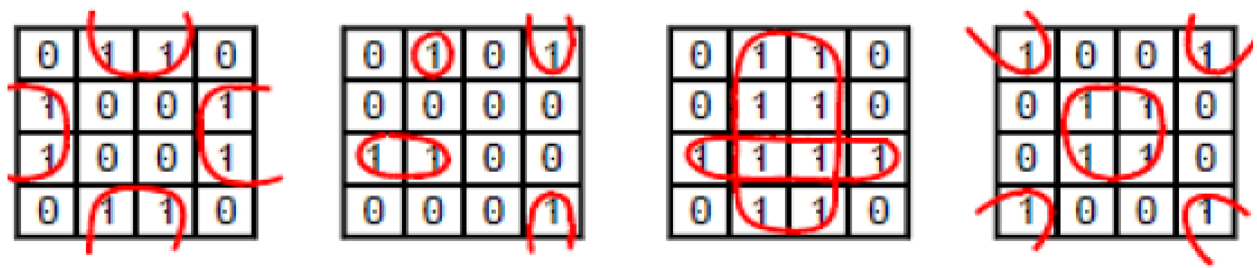


$$F(A,B,C,D) = \overline{C}.D + A.\overline{B}.\overline{C} + \overline{A}.B.C.\overline{D}$$

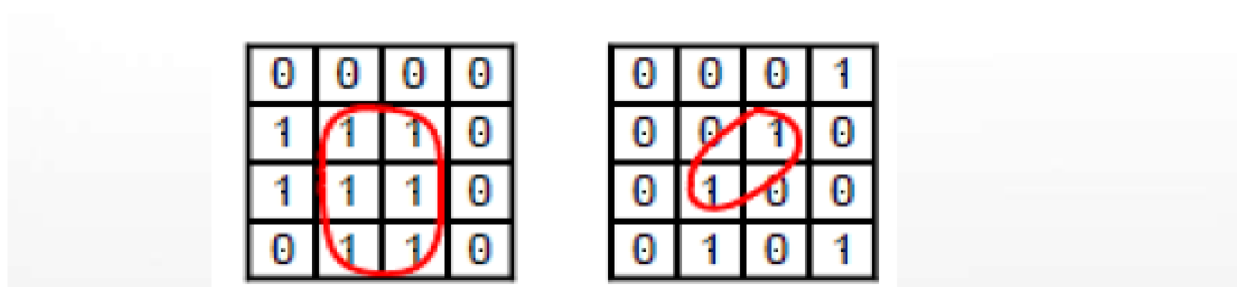
Simplification par table de **Karnaugh**

- **Remarques:** Avec la méthode de Karnaugh on essaye de faire le minimum des regroupements qui contient le maximum des cases Les cases des coins sont des cases adjacentes
- **Exemples**

POSSIBLES



IMPOSSIBLES



Simplification par table de **Karnaugh**

- **Exemple 3 :**

4 variables

CD \ AB	00	01	11	10
00	1			1
01		1	1	1
11				1
10	1			1

$$F(A,B,C,D) = A\overline{B} + \overline{B}\,\overline{D} + B\overline{C}D$$

Simplification par table de **Karnaugh**

- Simplification des expressions suivantes:
  Exemple1: $F(A, B,C,D)=\Sigma(3,5,6,7,8,9,10,11,12,13,14,15)$

$$F(A,B,C,D) = A$$

C.D \ A.B	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

Simplification par table de **Karnaugh**

- Simplification des expressions suivantes:
  Exemple1: $F(A, B,C,D)=\Sigma(3,5,6,7,8,9,10,11,12,13,14,15)$

$$F(A,B,C,D) = A + C.D$$

C.D \ A.B	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

Simplification par table de **Karnaugh**

- Simplification des expressions suivantes:
  Exemple1: $F(A, B,C,D)=\Sigma(3,5,6,7,8,9,10,11,12,13,14,15)$

$$F(A.B.C.D) = A + C.D + B.C$$

C.D \ A.B	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

Simplification par table de **Karnaugh**

- Simplification des expressions suivantes:
  Exemple1: *F(A, B,C,D)=Σ(3,5,6,7,8,9,10,11,12,13,14,15)*

$$F(A,B,C,D) = A + C.D + B.C + B.D$$

$C.D$ \ $A.B$	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

Simplification par table de **Karnaugh**

- Simplification des tableaux de karnaugh à 5 variables:

Exemple1:

$e = 0$

cd \ ab	00	01	11	10
00	0 $_{0}$	1 $_{2}$	1 $_{3}$	0 $_{1}$
01	0 $_{8}$	1 $_{10}$	1 $_{11}$	1 $_{9}$
11	0 $_{12}$	0 $_{14}$	0 $_{15}$	1 $_{13}$
10	1 $_{4}$	1 $_{6}$	1 $_{7}$	1 $_{5}$

$e = 1$

cd \ ab	00	01	11	10
00	0 $_{16}$	1 $_{18}$	0 $_{19}$	0 $_{17}$
01	0 $_{24}$	1 $_{26}$	0 $_{27}$	1 $_{25}$
11	0 $_{28}$	0 $_{30}$	1 $_{31}$	1 $_{29}$
10	0 $_{20}$	1 $_{22}$	1 $_{23}$	0 $_{21}$

$F = c\bar{d}\bar{e} + b\bar{c}\bar{e} +$

Simplification par table de **Karnaugh**

- Simplification des tableaux de karnaugh à 5 variables:
  <u>Exemple1:</u>

$e = 0$

cd \ ab	00	b 01	11	10
$\bar{c}$ 00	0_{0}	1_{2}	1_{3}	0_{1}
01	0_{8}	1_{10}	1_{11}	1_{9}
11	0_{12}	0_{14}	0_{15}	1_{13}
10	1_{4}	1_{6}	1_{7}	1_{5}

$e = 1$

cd \ ab	00	01	11	10
00	0_{16}	1_{18}	0_{19}	0_{17}
01	0_{24}	1_{26}	0_{27}	1_{25}
11	0_{28}	0_{30}	1_{31}	1_{29}
10	0_{20}	1_{22}	1_{23}	0_{21}

$F = c\bar{d}\bar{e} + b\bar{c}\bar{e} + acde$

Simplification par table de **Karnaugh**

- Simplification des tableaux de karnaugh à 5 variables:
  Exemple1:

$e = 0$

cd \ ab	00	01	11	10
00	0_{0}	1_{2}	1_{3}	0_{1}
01	0_{8}	1_{10}	1_{11}	1_{9}
11	0_{12}	0_{14}	0_{15}	1_{13}
10	1_{4}	1_{6}	1_{7}	1_{5}

$e = 1$

cd \ ab	00	01	11	10
00	0_{16}	1_{18}	0_{19}	0_{17}
01	0_{24}	1_{26}	0_{27}	1_{25}
11	0_{28}	0_{30}	1_{31}	1_{29}
10	0_{20}	1_{22}	1_{23}	0_{21}

$$F = c\bar{d}\bar{e} + b\bar{c}\bar{e} + acde + a\bar{b}d$$

même position Alors: pas de e

Simplification par table de **Karnaugh**

- Simplification des tableaux de karnaugh à 5 variables:

Exemple1:

$e = 0$

cd \ ab	00	01	11	10
00	0_{0}	1_{2}	1_{3}	0_{1}
01	0_{8}	1_{10}	1_{11}	1_{9}
11	0_{12}	0_{14}	0_{15}	1_{13}
10	1_{4}	1_{6}	1_{7}	1_{5}

$e = 1$

cd \ ab	00	01	11	10
00	0_{16}	1_{18}	0_{19}	0_{17}
01	0_{24}	1_{26}	0_{27}	1_{25}
11	0_{28}	0_{30}	1_{31}	1_{29}
10	0_{20}	1_{22}	1_{23}	0_{21}

$$F = c\bar{d}\bar{e} + b\bar{c}\bar{e} + acde + a\bar{b}d + bc\bar{d}$$

Simplification par table de **Karnaugh**

- Simplification des tableaux de karnaugh à 5 variables:

Exemple1:

$e = 0$

cd \ ab	00	01	11	10
00	0_{0}	1_{2}	1_{3}	0_{1}
01	0_{8}	1_{10}	1_{11}	1_{9}
11	0_{12}	0_{14}	0_{15}	1_{13}
10	1_{4}	1_{6}	1_{7}	1_{5}

$e = 1$

cd \ ab	00	01	11	10
00	0_{16}	1_{18}	0_{19}	0_{17}
01	0_{24}	1_{26}	0_{27}	1_{25}
11	0_{28}	0_{30}	1_{31}	1_{29}
10	0_{20}	1_{22}	1_{23}	0_{21}

$$F = c\bar{d}\bar{e} + b\bar{c}\bar{e} + acde + a\bar{b}d + bc\bar{d} + \bar{a}b\bar{c}$$

Simplification par table de **Karnaugh**

- Simplification des tableaux de karnaugh à 5 variables:
  Exemple1:

$e = 0$

cd \ ab	00	01	11	10
00	0 $_{0}$	1 $_{2}$	1 $_{3}$	0 $_{1}$
01	0 $_{8}$	1 $_{10}$	1 $_{11}$	1 $_{9}$
11	0 $_{12}$	0 $_{14}$	0 $_{15}$	1 $_{13}$
10	1 $_{4}$	1 $_{6}$	1 $_{7}$	1 $_{5}$

$e = 1$

cd \ ab	00	01	11	10
00	0 $_{16}$	1 $_{18}$	0 $_{19}$	0 $_{17}$
01	0 $_{24}$	1 $_{26}$	0 $_{27}$	1 $_{25}$
11	0 $_{28}$	0 $_{30}$	1 $_{31}$	1 $_{29}$
10	0 $_{20}$	1 $_{22}$	1 $_{23}$	0 $_{21}$

$$F = c\bar{d}\bar{e} + b\bar{c}\bar{e} + acde + a\bar{b}d + bc\bar{d} + \bar{a}b\bar{c}$$

Simplification par table de **Karnaugh**

- Simplification des expressions suivantes:

  Exemple2:

  F(A, B,C,D, E) =Σ(0,1,2,4,6,8,9,10,14,16,17,18,19,22,24,25,26,27,30)

D.E \ B.C	00	01	11	10
00	1	1	0	1
01	1	0	0	1
11	0	0	0	0
10	1	1	1	1

$A = 0$

D.E \ B.C	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

$A = 1$

$$F(A,B,C,D,E) = D.\overline{E}$$

Simplification par table de **Karnaugh**

- Simplification des expressions suivantes:

Exemple2:

F(A, B,C,D, E) =Σ(0,1,2,4,6,8,9,10,14,16,17,18,19,22,24,25,26,27,30)

D.E \ B.C	00	01	11	10
00	1	1	0	1
01	1	0	0	1
11	0	0	0	0
10	1	1	1	1

$A = 0$

D.E \ B.C	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

$A = 1$

$$F(A,B,C,D,E) = D.\overline{E} + \overline{C}.\overline{D}$$

Simplification par table de **Karnaugh**

- Simplification des expressions suivantes:

  Exemple2:

  F(A, B,C,D, E) =Σ(0,1,2,4,6,8,9,10,14,16,17,18,19,22,24,25,26,27,30)

D.E \ B.C	00	01	11	10
00	1	1	0	1
01	1	0	0	1
11	0	0	0	0
10	1	1	1	1

$A=0$

D.E \ B.C	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

$A=1$

$$F(A,B,C,D,E) = D.\overline{E} + \overline{C}.\overline{D} + A.\overline{C}$$

Simplification par table de **Karnaugh**

- Simplification des expressions suivantes:

Exemple2:

F(A, B,C,D, E) =Σ(0,1,2,4,6,8,9,10,14,16,17,18,19,22,24,25,26,27,30)

D.E \ B.C	00	01	11	10
00	1	1	0	1
01	1	0	0	1
11	0	0	0	0
10	1	1	1	1

$A = 0$

D.E \ B.C	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

$A = 1$

$$F(A,B,C,D,E) = D.\overline{E} + \overline{C}.\overline{D} + A.\overline{C} + \overline{A}.\overline{B}.\overline{E}$$

Exemple : $F(A,B,C,D,E,F) = \sum(1,5,8,9,12,13,16,17,18,21,24,25,26,29,$
$33,35,37,39,40,41,44,45,48,49,50,51,53,55,56,57,58,61).$

$E.F \backslash C.D$	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

$A.B = 00$

$E.F \backslash C.D$	00	01	11	10
00	16	20	28	24
01	17	21	29	25
11	19	23	31	27
10	18	22	30	26

$A.B = 01$

$E.F \backslash C.D$	00	01	11	10
00	32	36	44	40
01	33	37	45	41
11	35	39	47	43
10	34	38	46	42

$A.B = 10$

$E.F \backslash C.D$	00	01	11	10
00	48	52	60	56
01	49	53	61	57
11	51	55	63	59
10	50	54	62	58

$A.B = 11$

Exemple : $F(A, B, C, D, E, F) = \sum(1, 5, 8, 9, 12, 13, 16, 17, 18, 21, 24, 25, 26, 29,$
$33, 35, 37, 39, 40, 41, 44, 45, 48, 49, 50, 51, 53, 55, 56, 57, 58, 61).$

E.F \ C.D	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

$A.B = 00$

E.F \ C.D	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	0	0	0	0
10	1	0	0	1

$A.B = 01$

E.F \ C.D	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	1	1	0	0
10	0	0	0	0

$A.B = 10$

E.F \ C.D	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	0	0
10	1	0	0	1

$A.B = 11$

$$F(A, B, C, D, E, F) = \overline{E}.F$$

Exemple : $F(A,B,C,D,E,F) = \sum(1,5,8,9,12,13,16,17,18,21,24,25,26,29,$
$33,35,37,39,40,41,44,45,48,49,50,51,53,55,56,57,58,61).$

E.F \ C.D	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

$A.B = 00$

E.F \ C.D	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	0	0	0	0
10	1	0	0	1

$A.B = 01$

E.F \ C.D	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	1	1	0	0
10	0	0	0	0

$A.B = 10$

E.F \ C.D	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	0	0
10	1	0	0	1

$A.B = 11$

$$F(A,B,C,D,E,F) = \overline{E}.F + \overline{B}.C.\overline{E}$$

Exemple : $F(A,B,C,D,E,F) = \sum(1,5,8,9,12,13,16,17,18,21,24,25,26,29,$
$33,35,37,39,40,41,44,45,48,49,50,51,53,55,56,57,58,61).$

E.F \ C.D	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

$A.B = 00$

E.F \ C.D	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	0	0	0	0
10	1	0	0	1

$A.B = 01$

E.F \ C.D	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	1	1	0	0
10	0	0	0	0

$A.B = 10$

E.F \ C.D	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	0	0
10	1	0	0	1

$A.B = 11$

$$F(A,B,C,D,E,F) = \overline{E}.F + \overline{B}.C.\overline{E} + B.\overline{D}.\overline{F}$$

Exemple : $F(A,B,C,D,E,F) = \sum(1,5,8,9,12,13,16,17,18,21,24,25,26,29,$
$33,35,37,39,40,41,44,45,48,49,50,51,53,55,56,57,58,61).$

$E.F \backslash C.D$	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

$A.B = 00$

$E.F \backslash C.D$	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	0	0	0	0
10	1	0	0	1

$A.B = 01$

$E.F \backslash C.D$	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	1	1	0	0
10	0	0	0	0

$A.B = 10$

$E.F \backslash C.D$	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	0	0
10	1	0	0	1

$A.B = 11$

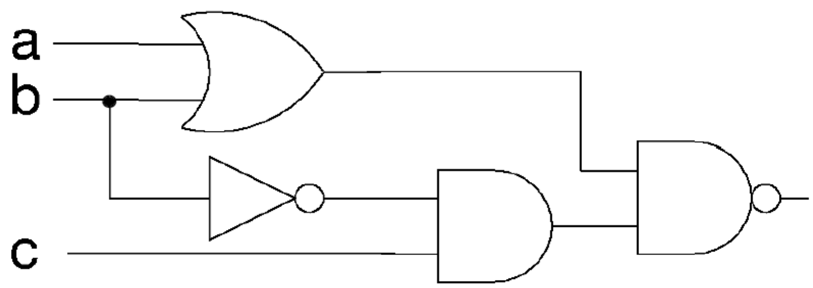
$$F(A,B,C,D,E,F) = \overline{E}.F + \overline{B}.C.\overline{E} + B.\overline{D}.\overline{F} + A.\overline{C}.F$$

Synthèse d'un circuit logique

- Il faut bien comprendre le fonctionnement du système.
- Il faut définir les variables d'entrée.
- Il faut définir les variables de sortie.
- Etablir la table de vérité.
- Ecrire les équations algébriques des sorties (à partir de la table de vérité).
- Effectuer des simplifications (algébrique ou par tableaux de Karnaugh…..).
- Faire le schéma avec un minimum de portes logiques.

Analyse de circuit logique

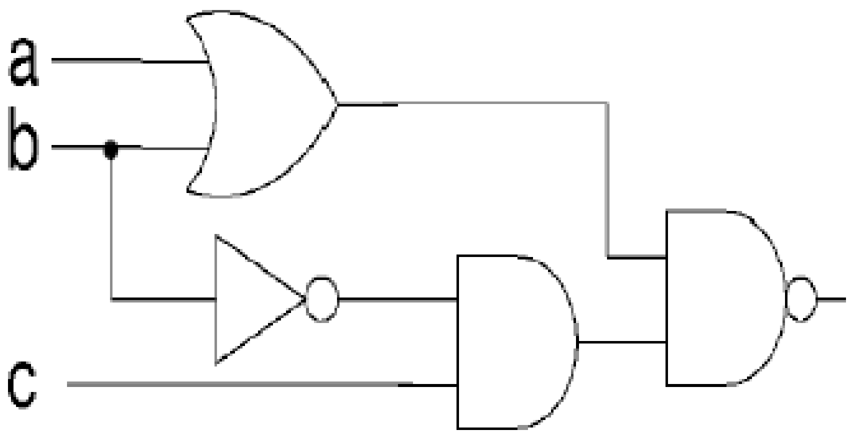
- Analyse de circuit logique
- 3 entrées, 1 sortie
- Composé uniquement de portes logiques OU, ET et NON



- A partir de son logigramme $f(a,b,c) = \overline{(a+b) \bullet (\bar{b} \bullet c)}$

Analyse de circuit logique

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



a	b	c	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Méthode de quine-mac cluskey

- **Introduction :**
- Bien que peu employée, nous devons quand même citer une méthode différente de celle des tableaux de Karnaugh pour la simplification des fonctions de plusieurs variables. Il s'agit d'une méthode imaginée par le mathématicien américain W. V. Quine et remodelée par son compatriote le Docteur Edwards J. Mac Cluskey Junior dans une thèse de doctorat qu'il présenta au M. I. T. (Massachusetts Institute of Technology) en juin 1956.

Méthode de quine-mac cluskey

- Description de la méthode
- Pour examiner cette méthode, nous prendrons un exemple à quatre variables. Partons de la table de vérité de la fonction **f (a, b, c, d)**.
- Nous pouvons constater que la fonction f est **"vraie"** (égale à 1) pour onze combinaisons des variables:
  a, b, c et d.
- Donnons comme nom à chacune de ces combinaisons la valeur décimale qui correspond au code binaire formé par les variables a, b, c, d en considérant que a est le bit de poids fort et d le bit de poids faible.

La méthode de Quine-Mc Cluskey est utilisée lorsque la fonction est représentée par:
première forme canonique
(ou deuxième forme canonique)
ou chaque intersection contient
toutes les variables de la fonction.

a	b	c	d	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Méthode de quine-mac cluskey

Elle consiste tout d'abord à classer les intersections canoniques monômes selon leurs poids.

On appelle **poids** d'un monôme complet, le poids de son équivalent binaire.

On appelle **l'équivalent binaire** le nombre binaire obtenu en remplaçant toutes les variables en négation (variables barrées) par 0 et autres (variables non barrées) par des 1.

On appelle poids d'un nombre binaire le nombre de 1 contenu dans ce nombre où le nombre de lettre non barrée dans ce monôme.

Exemple : monôme complet : $ab\bar{c}\bar{d}$

Équivalent binaire 1100;   Poids : 2

Méthode de quine-mac cluskey

- **Remarque :**
  le poids maximum = nombre de variable de la fonction.
- Une fois que le classement fait, on compare chaque terme de chaque groupe avec chaque terme de groupe de poids immédiatement supérieur. En utilisant l'identité suivante :

$$Ax + A\bar{x} = A$$

- A : représente l'ensemble des variables qui ne changent pas d'état.
- x : la variable qui change d'état. On obtiendra ainsi une 1ère forme réduite. Que l'on réduira de nouveau jusqu'à que ne soit plus possible de faire des réductions.

Méthode de quine-mac cluskey

- Exemple : $ab\bar{c}\bar{d}=1100$
- nous l'appellerons la combinaison 12 puisque 1100 en binaire équivaut à 12 en décimal. Si nous récapitulons, la fonction f est donc la somme des combinaisons résumées tableau suivant. Cette liste peut être récapitulée de la façon suivante :

$$\boxed{f \quad = \quad \Sigma \quad (0, 1, 2, 4, 6, 8, 9, 12, 13, 14, 15)}$$

Fonction «somme de» Liste des combinaisons

- A partir de cette liste de combinaisons commence réellement la méthode de Quine-Mac Cluskey.

Méthode de quine-mac cluskey

a	b	c	d	Combinaisons
0	0	0	0	Combinaison 0
0	0	0	1	Combinaison 1
0	0	1	0	Combinaison 2
0	1	0	0	Combinaison 4
0	1	1	0	Combinaison 6
1	0	0	0	Combinaison 8
1	0	0	1	Combinaison 9
1	1	0	0	Combinaison 12
1	1	0	1	Combinaison 13
1	1	1	0	Combinaison 14
1	1	1	1	Combinaison 15

Combinaisons pour les quelles f = 1

Méthode de quine-mac cluskey

- La première chose à faire est de classer ces combinaisons en ensembles successifs en fonction du nombre de zéros (ou nombre de 1) de la forme binaire de chaque combinaison en commençant par celle qui en compte le plus (qui compte le moins).
- Le premier ensemble est donc formé par la combinaison 0 dont la forme binaire 0000 comporte quatre zéros. Le second ensemble regroupe les combinaisons comportant trois zéros (un seul 1) ... jusqu'au cinquième qui est formé de la combinaison 15 dont la forme binaire 1111 ne comporte pas de zéro du tout (tous des uns). Les cinq ensembles sont représentés dans la figure suivante:.

Méthode de quine-mac cluskey

	a	b	c	d	
Combinaison 0	0	0	0	0	ensemble 1 (4 zéros)
Combinaison 1	0	0	0	1	ensemble 2 (3 zéros)
Combinaison 2	0	0	1	0	
Combianison 4	0	1	0	0	
Combinaison 8	1	0	0	0	
Combinaison 6	0	1	1	0	ensemble 3 (2 zéros)
Combinaison 9	1	0	0	1	
Combinaison 12	1	1	0	0	
Combinaison 13	1	1	0	1	ensemble 4 (1 zéro)
Combinaison 14	1	1	1	0	
Combinaison 15	1	1	1	1	ensemble 5 (0 zéro)

Méthode de quine-mac cluskey

- La seconde phase de la méthode consiste à comparer les termes de chaque ensemble avec les termes de l'ensemble qui suit immédiatement de façon à éliminer une variable si cela est possible.
- La combinaison 0000 de l'ensemble 1 doit donc être comparée aux quatre combinaisons de l'ensemble 2. Puis chacune des quatre combinaisons de cet ensemble 2 sera comparée à chacune des trois combinaisons de l'ensemble 3 et ainsi de suite jusqu'à l'ensemble 5...

Méthode de quine-mac cluskey

Lorsque deux combinaisons ne diffèrent que d'une variable, celles-ci peuvent être associées pour n'en former plus qu'une dans laquelle la variable qui diffère peut être éliminée (l'adjacence). Quand une variable s'élimine dans une telle association, on signale ce fait en remplaçant cette variable par une croix (ou tout autre signe à votre convenance).

Dans l'exemple que nous avons choisi, les combinaisons 0 et 1 peuvent, par exemple, être associées puisque seule la variable d est différente.

Combinaison 0 :	0 0 0 0
Combinaison 1 :	0 0 0 1
Combinaison 0 - 1:	0 0 0 X

Méthode de quine-mac cluskey

- Procédons ainsi en comparant chaque terme de chaque ensemble avec chaque terme de l'ensemble suivant et nous obtenons les combinaisons décrites dans la le tableau suivant.
- Nous obtenons de nouveaux membres regroupés à présent dans quatre ensembles I, II, III et IV.
- Recommençons la même opération avec ces nouveaux ensembles. L'autre tableau donne les nouvelles combinaisons obtenues.

	a	b	c	d	
0 - 1	0	0	0	X	
0 - 2	0	0	X	0	ensembles 1 et 2
0 - 4	0	X	0	0	I
0 - 8	X	0	0	0	
1 - 9	X	0	0	1	
2 - 6	0	X	1	0	ensembles 2 et 3
4 - 6	0	1	X	0	
4 - 12	X	1	0	0	II
8 - 9	1	0	0	X	
8 - 12	1	X	0	0	
6 - 14	X	1	1	0	
9 - 13	1	X	0	1	ensembles 3 et 4
12 - 13	1	1	0	X	III
12 - 14	1	1	X	0	
13 - 15	1	1	X	1	ensembles 4 et 5
14 - 15	1	1	1	X	IV

Combinaisons des ensembles 1, 2, 3, 4 et 5

Méthode de quine-mac cluskey

			a	b	c	d	
ensembles I et II	0 - 1	8 - 9	X	0	0	X	A
	0 - 2	4 - 6	0	X	X	0	B
	0 - 4	2 - 6	0	X	X	0	B
	0 - 4	8 - 12	X	X	0	0	C
	0 - 8	1 - 9	X	0	0	X	A
	0 - 8	4 - 12	X	X	0	0	C
ensembles II et III	4 - 6	12 - 14	X	1	X	0	D
	4 - 12	6 - 14	X	1	X	0	D
	8 - 9	12 - 13	1	X	0	X	E
	8 - 12	9 - 13	1	X	0	X	E
ensembles III et IV	12 - 13	14 - 15	1	1	X	X	F
	12 - 14	13 - 15	1	1	X	X	F

Combinaisons des ensembles I, II, III, IV

Méthode de quine-mac cluskey

- Dans les principes généraux de la méthode de Quine-Mac Cluskey, il convient de continuer ce processus jusqu'à ce qu'il n'y ait plus de combinaison possible entre un de ces ensembles et le suivant.
- Dans l'exemple choisi, nous en sommes arrivés à ce point. De plus, nous constatons dans cette figure que plusieurs combinaisons sont identiques.
- Nous ne garderons bien sûr que les groupements différents que nous appellerons les **impliquants premiers** : **A, B, C, D, E et F**

Méthode de quine-mac cluskey

- La fonction f prend alors la forme :
- **f = A + B + C + D + E + F**

  $= \overline{b}\,\overline{c} + \overline{a}\,\overline{d} + \overline{c}\,\overline{d} + b\overline{d} + a\overline{c} + ab$

- **Remarque**

  Dans tous les regroupements que nous avons effectués, il aurait pu arriver que certains termes d'un ensemble ne se combinent avec aucun terme de l'ensemble suivant. Ces termes sont alors des **termes premiers** et il convient de ne pas les oublier dans l'équation finale.

Méthode de quine-mac cluskey

- Arrivé à ce stade, il convient de vérifier si l'équation obtenue ne peut pas encore être simplifiée. Pour éliminer les éventuels termes superflus, formons une grille dite **"diagramme des termes premiers"**.
- Ce diagramme représenté comporte onze colonnes correspondant aux combinaisons initiales 0, 1, 2, 4, 6, 8, 9, 12, 13, 14 et 15. Il comprend d'autre part six lignes horizontales correspondant aux six groupements A, B, C, D, E et F obtenus après les différents processus de simplification. Sur chaque ligne, marquons d'une croix les combinaisons ayant servies à former le groupement considéré.

Méthode de quine-mac cluskey

- Par exemple pour le groupement A qui fait intervenir les combinaisons 0, 1, 8 et 9, nous avons coché sur la ligne A les colonnes 0, 1, 8 et 9.
- Après avoir fait de même pour les lignes B, C, D, E et F, nous constatons que certaines colonnes (1, 2 et 15) ne comportent qu'une seule croix que nous entourons alors d'un cercle.
- Ces combinaisons sont situées sur les lignes A, B, et F qui sont appelées **"lignes de base"**.

Méthode de quine-mac cluskey

- Nous constatons que d'autre part les combinaisons incluses dans les lignes C, D, E se retrouvent au moins une fois dans les lignes A, B et F (les impliquants premiers essentiels).
- Les lignes C, D et E peuvent donc être supprimées et la fonction f se simplifie donc encore pour devenir :

- **f = A + B + F** $= \bar{b}\,\bar{c} + \bar{a}\,\bar{d} + ab$

Méthode de quine-mac cluskey

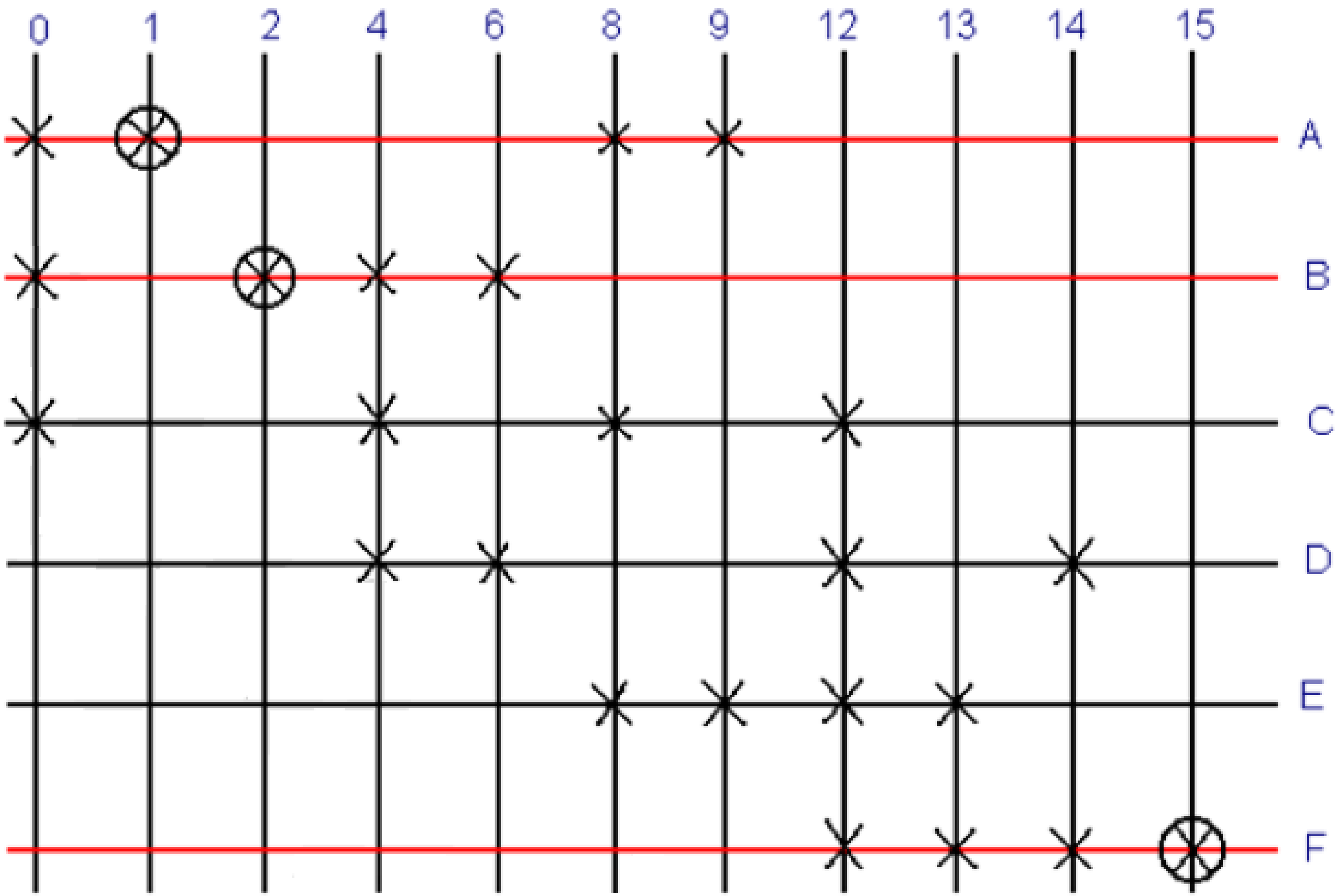


Diagramme des termespremiers ou grille de MAC Cluskey

Méthode de quine-mac cluskey

- **Comparaison avec la méthode de karnaugh**
- A titre de comparaison, effectuons le même exercice par la méthode des tableaux de Karnaugh. A l'aide de la table de vérité, établissons le tableau de Karnaugh correspondant .
- Les regroupements effectués dans ce tableau (cerclés de rouge) donnent immédiatement la solution :
- **f = 1 + 2 + 3** $= \overline{b}\,\overline{c} + \overline{a}\,\overline{d} + ab$
- On retrouve le même résultat que précédemment.

a	b	c	d	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

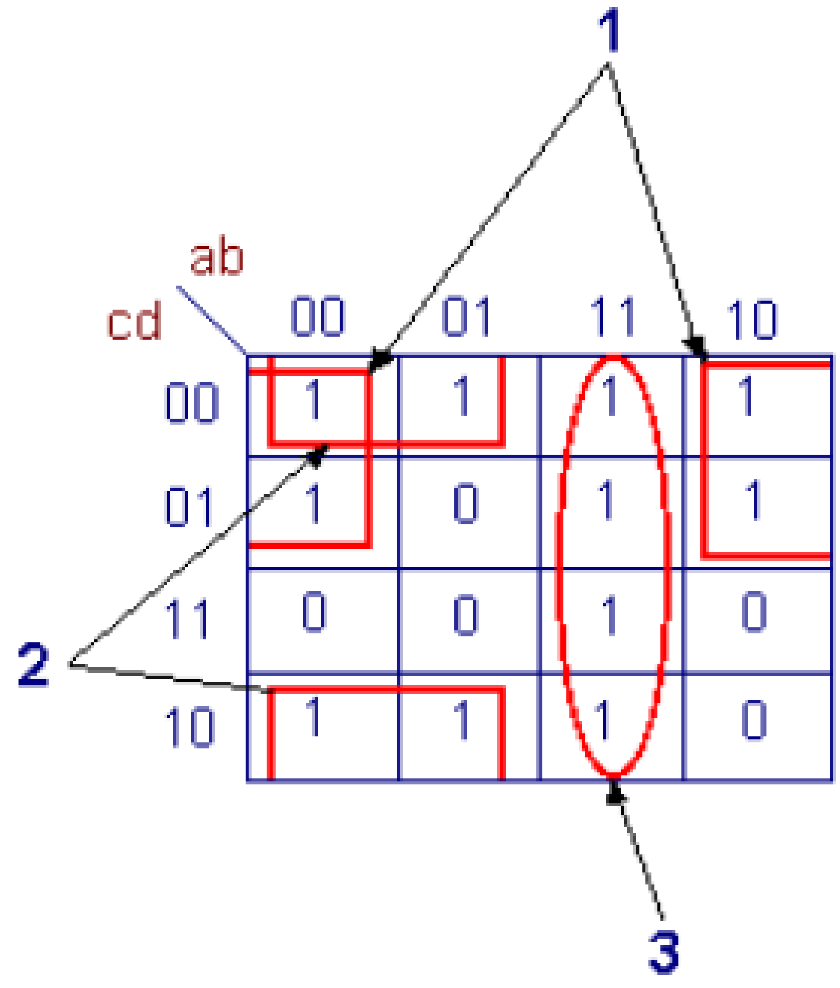


Tableau de Karnaugh

Méthode de quine-mac cluskey

Conclusion : a la lumière de cet exemple, résolu d'une part avec la méthode de Quine Mac Cluskey, d'autre part grâce à un tableau de Karnaugh, il apparaît que la seconde est nettement plus simple, rapide et séduisante que la première. Ceci explique que cette première méthode soit quasiment inemployée. Nous ne nous étendrons pas davantage sur elle.

Précisons toutefois à sa décharge que cette méthode de Quine Mac Cluskey s'applique quel que soit le nombre de variables alors que la méthode des tableaux de Karnaugh se complique notablement lorsque le nombre de variables augmentent et devient supérieur à 6.

Fin du chapitre