REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

Université Larbi Ben M'hidi Oum el Bouaghi
ISTA Ain M'lila
Département Réseaux et Télécommunications

1ère Année Licence
Module : Informatique 1

Algèbre de Boole

© Dr BENACER Imad Jan 2021

Année Universitaire : 2020 / 2021

PLAN D'EXPOSE

Définition des variables et fonctions logiques

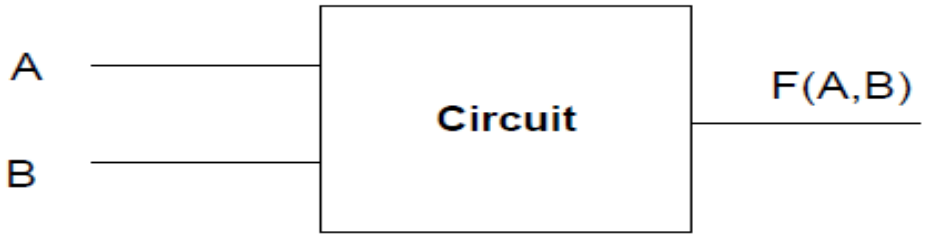
Les opérateurs de base et les portes logiques

Les lois fondamentales de l'algèbre de Boole

Introduction

- Les machines numériques (ordinateur, tablette, téléphone…) sont constituées d'un ensemble de circuits électroniques.

- Chaque circuit fournit une fonction logique bien déterminée; opérations logiques ou arithmétiques (addition, soustraction, comparaison ,….).

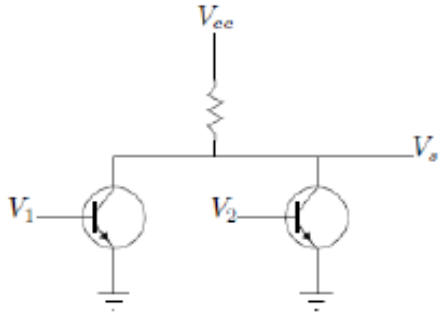


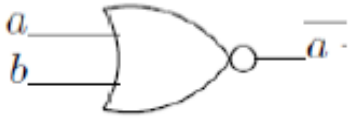
La fonction F(A,B) peut être : la somme de A et B , ou le résultat de la comparaison de A et B ou une autre fonction

- Une fonction logique de base est réalisée à l'aide des portes logiques qui permettent d'effectuer des opérations élémentaires.

Introduction

- Ces portes logiques sont aujourd'hui réalisées à l'aide de transistors.





- 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.

2. Algèbre de Boole

- George Boole est un mathématicien anglais (1815-1864).
- Il a fait des travaux dont les quels les fonctions (expressions) sont constitués par des variables qui peuvent prendre les valeurs 'OUI' ou 'NON' .
- Ces travaux ont été utilisés pour faire l'étude des systèmes qui possèdent deux états s'exclus mutuellement :
  - Le système peut être uniquement dans deux états E1 et E2 tel que E1 est l'opposé de E2.
  - Le système ne peut pas être dans l'état E1 et E2 en même temps
- Ces travaux sont bien adaptés au Système binaire (0 et 1).

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)

3. Définitions et conventions

3.1. **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

lampe éteinte : 0

Logique négative

lampe allumée : 0

lampe éteinte : 1

3.3. Fonction logique

- C'est une fonction qui relie N variables logiques avec un ensemble d'opérateurs logiques de base.
- Dans l'Algèbre de Boole 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).

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

$$F(0,0,0) = \overline{0}.\overline{0}.0 + \overline{0}.0.0 + 0.\overline{0}.0 + 0.0.0 = 0$$
$$F(0,0,1) = \overline{0}.\overline{0}.1 + \overline{0}.0.1 + 0.\overline{0}.1 + 0.0.1 = 1$$
$$F(0,1,0) = \overline{0}.\overline{1}.0 + \overline{0}.1.0 + 0.\overline{1}.0 + 0.1.0 = 0$$
$$F(0,1,1) = \overline{0}.\overline{1}.1 + \overline{0}.1.1 + 0.\overline{1}.1 + 0.1.1 = 1$$
$$F(1,0,0) = \overline{1}.\overline{0}.0 + \overline{1}.0.0 + 1.\overline{0}.0 + 1.0.0 = 0$$
$$F(1,0,1) = \overline{1}.\overline{0}.1 + \overline{1}.0.1 + 1.\overline{0}.1 + 1.0.1 = 1$$
$$F(1,1,0) = \overline{1}.\overline{1}.0 + \overline{1}.1.0 + 1.\overline{1}.0 + 1.1.0 = 0$$
$$F(1,1,1) = \overline{1}.\overline{1}.1 + \overline{1}.1.1 + 1.\overline{1}.1 + 1.1.1 = 1$$

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

Une table de vérité

4. Opérateurs logiques de base
4.1 NON (négation)

- NON : est un opérateur unaire (une seule variable) qui à pour rôle d'inverser la valeur d'une variable .

$$F(A) = \text{Non } A = \overline{A}$$

(lire : A barre)

A	
0	1
1	0

4.2 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 défini par : F(A,B)= A . B

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

4.3 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

4.5 Lois fondamentales de l'Algèbre de Boole

- **Commutativité**

  $a+b = b+a$

  $a.b = b.a$

- **Associativité**

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

  $a.(b.c) = (a.b).c$

- **Distributivité**

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

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

- **Idempotence**

  $a+a = a$

  $a.a = a$

- **Absorption**

  $a+a.b = a$

  $a.(a+b) = a$

- **Involution**

  $\overline{\overline{a}} = a$

- **Elément neutre**

$$a+0 = a$$

$$a.1 = a$$

- **Elément absorbant**

$$a+1 = 1$$

$$a.0 = 0$$

- **Inverse**

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

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

- **Autres relations utiles**

$$A + (A . B) = A$$

$$A.(A + B) = A$$

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

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

6. Théorème de DE-MORGANE

•La somme logique complimentée de deux variables est égale au produit des compléments des deux variables.

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

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

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

6.1 Généralisation du Théorème DE-MORGANE à N variables

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

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

7. Autres opérateurs logiques
7.1 OU exclusif (XOR)

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

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

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

7.2 NAND (NON ET)

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

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

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

7.3 NOR (NON OU)

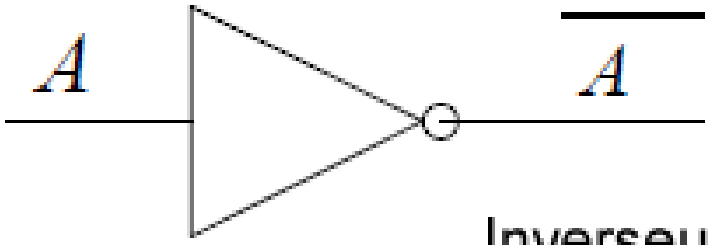
$$F(A,B) = \overline{A+B}$$

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

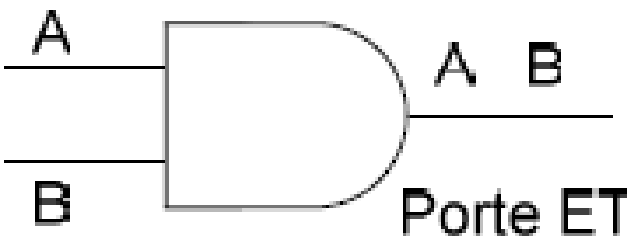
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

8. Portes logiques

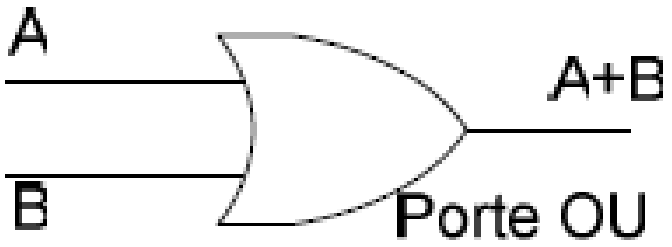
Une porte logique est un circuit électronique élémentaire qui Permet de réaliser la fonction d'un opérateur logique de base



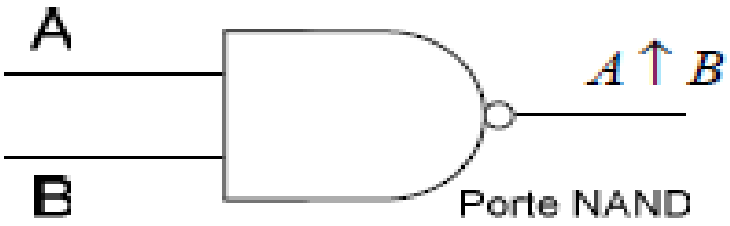
Inverseur

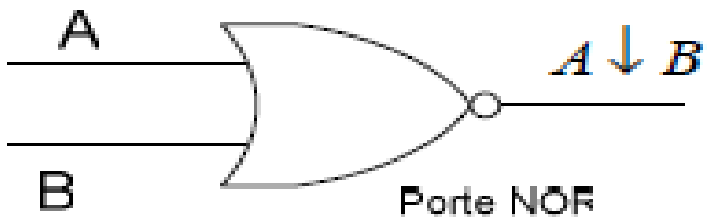


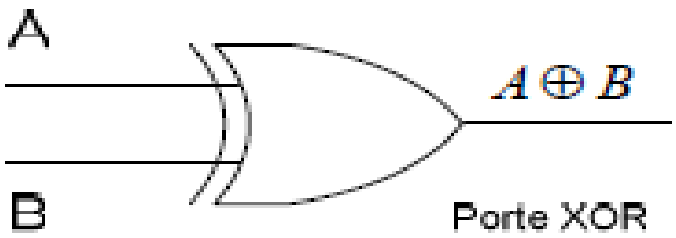
Porte ET



Porte OU







Remarque :

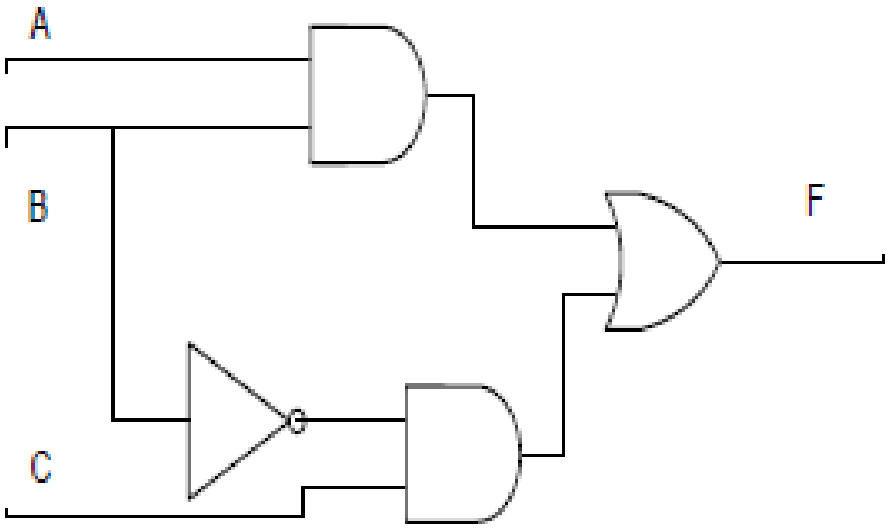
- Les portes ET , OU , NAND , NOR peuvent avoir plus que deux entrées
- Il n'existe pas de OU exclusif à plus de deux entrées

8.1 Schéma d'un circuit logique (Logigramme)

•C'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.

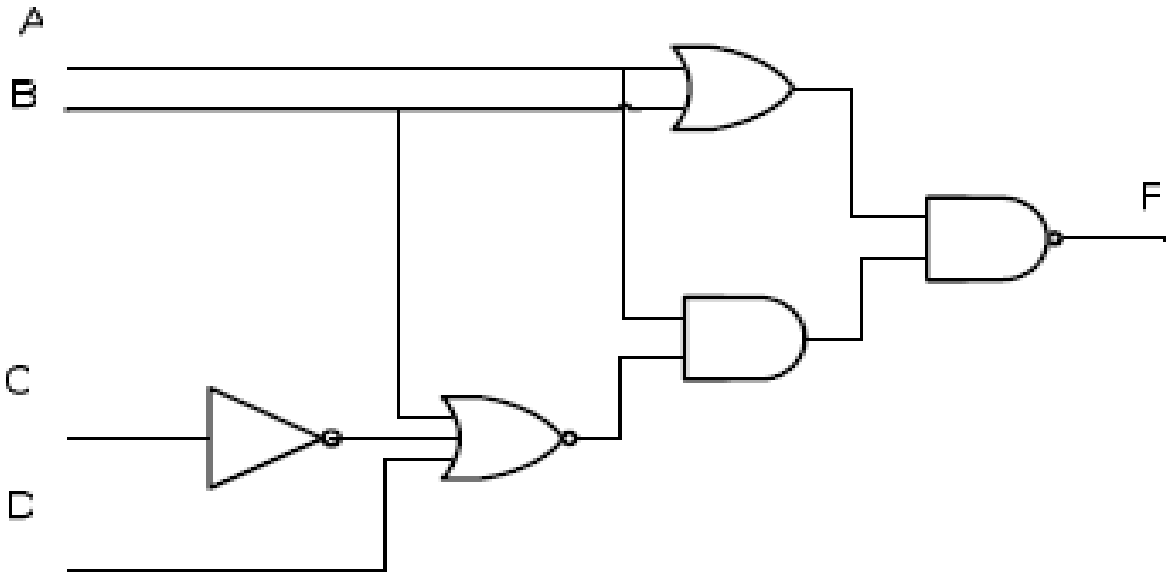
Exemple1

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



Exemple 2

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



Définition textuelle d'une fonction logique , table de vérité , formes algébriques , simplification algébrique, table de Karnaugh

Définition textuelle d'une fonction logique

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

Exemple : définition textuelle du fonctionnement d'un système

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

Donner la schéma du circuit qui permet de contrôler l'ouverture de la serrure ?

Étapes de conception et de réalisation d'un circuit numérique

- Pour faire l'étude et la réalisation d'un circuit il faut suivre le étapes suivantes :

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

Si on reprend l'exemple de la serrure :

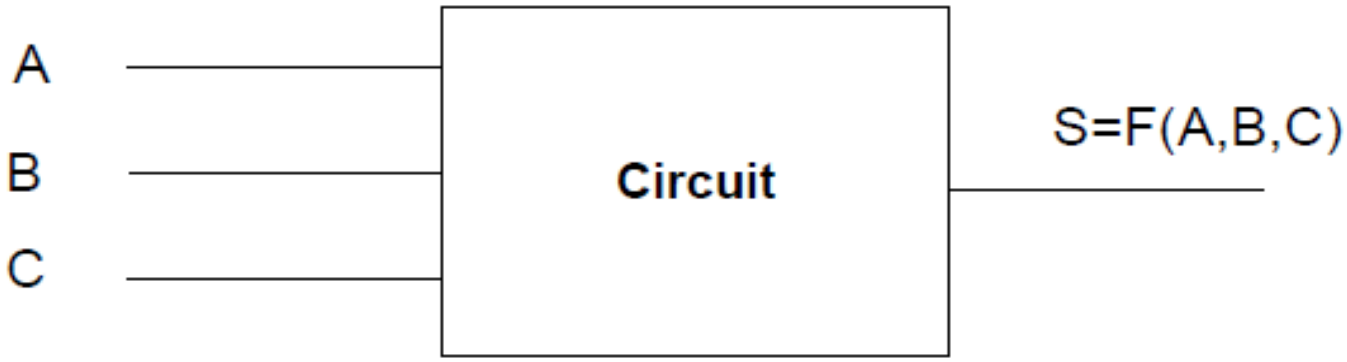
- Le système possède trois entrées : chaque entrée représente une clé.
- On va correspondre à chaque clé une variable logique: clé 1 → A , la clé 2 → B , la clé 3 → C
  - Si la clé 1 est utilisée alors la variable A=1 sinon A =0
  - Si la clé 2 est utilisée alors la variable B=1 sinon B =0
  - Si la clé 3 est utilisée alors la variable C=1 sinon C =0

- 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)

F(A,B,C)= 1 si au mois deux clés sont introduites

F(A,B,C)=0 si non .



Remarque :

Il est important de préciser aussi le niveau logique avec lequel on travail (logique positive ou négative).

9 Table de vérité (Rappel)

- 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.

9 Table de vérité (Exemple)

A	B	C		S		
0	0	0		**0**	→	$A+B+C$: max terme
0	0	1		**0**	→	$A+B+\overline{C}$: max terme
0	1	0		**0**	→	$A+\overline{B}+C$: max terme
0	1	1		**1**	→	$\overline{A}.B.C$: min terme
1	0	0		**0**	→	$\overline{A}+B+C$: max terme
1	0	1		**1**	→	$A.\overline{B}.C$: min terme
1	1	0		**1**	→	$A.B.\overline{C}$: min terme
1	1	1		**1**	→	$A.B.C$: min terme

Extraction de la fonction logique à partir de la T.V

- F = somme min termes

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

- F = produit des max termes

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

Forme canonique d'une fonction logique

- On appel forme canonique d'une fonction la forme ou chaque terme de la fonction comportent toutes les variables.

- Exemple :

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

Il existent plusieurs formes canoniques : les plus utilisées sont la première et la deuxième forme .

Première forme canonique

- Première forme canonique (forme disjonctive) : somme de produits
- C'est la somme des min termes.
- Une disjonction de conjonctions.

- Exemple :

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

- Cette forme est la forme la plus utilisée.

Deuxième forme canonique

- Deuxième forme canonique (conjonctive): produit de sommes
- Le produit des max termes
- Conjonction de disjonctions
- Exemple :

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

La première et la deuxième forme canonique sont équivalentes .

Simplification des fonctions logiques

10. Simplification des fonctions logiques

- L'objectif de la simplification des fonctions logiques est de :
  - réduire le nombre de termes dans une fonction
  - et de réduire le nombre de variables dans un terme

- Cela afin de réduire le nombre de portes logiques utilisées
  → réduire le coût du circuit

- Plusieurs méthodes existent pour la simplification :
  - La Méthode algébrique
  - Les Méthodes graphiques : (ex : table de karnaugh)
  - Les méthodes programmables

A Méthode algébrique

- Le principe consiste à appliquer les règles de l'algèbre de Boole afin d'éliminer des variables ou des termes.
- Mais il n'y a pas une démarche bien spécifique.
- Voici quelques règles les plus utilisées :

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

$$A + A \cdot B = A$$

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

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

$$A \cdot (A + B) = A$$

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

Règles de simplification

- Règles 1 : regrouper des termes à l'aide des règles précédentes

- Exemple

$$ABC + AB\overline{C} + A\overline{B}CD = AB\,(C + \overline{C}) + A\overline{B}CD$$
$$= AB + A\overline{B}CD$$
$$= A\,(B + \overline{B}\,(CD))$$
$$= A\,(B + CD)$$
$$= AB + ACD$$

- Règles 2 : Rajouter un terme déjà existant à une expression

- Exemple :

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

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

$$BC + \qquad AC \qquad + \quad AB$$

- **Règles 3 :** il est possible de supprimer un terme superflu (un terme en plus), c'est-à-dire déjà inclus dans la réunion des autres termes.

- Exemple 1 :

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

$$= AB + \overline{B}C + ACB + A\overline{B}C$$

$$= AB\,(1 + C) + \overline{B}C\,(1 + A)$$

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

B Simplification par la table de Karnaugh

Les termes adjacents

- Examinons l'expression suivante :

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

- Les deux termes possèdent les même 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.

Exemple de termes adjacents

Ces termes sont adjacents

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

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

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

Description de la table de karnaugh

- La méthode de Karnaugh se base sur la règle précédente.
- La méthode consiste a mettre en évidence par une méthode graphique (un tableaux) 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).

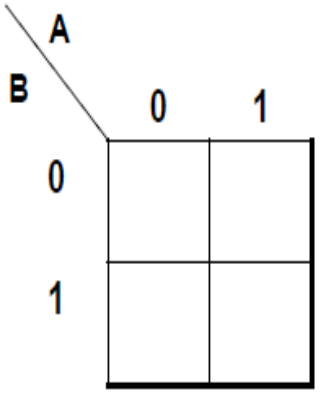
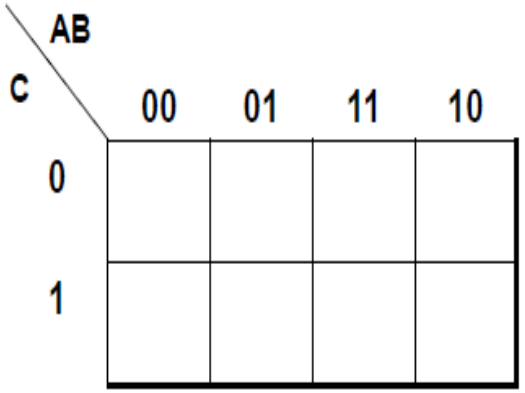


Tableau à 2 variables



Tableaux à 3 variables

Tableau à 4 variables

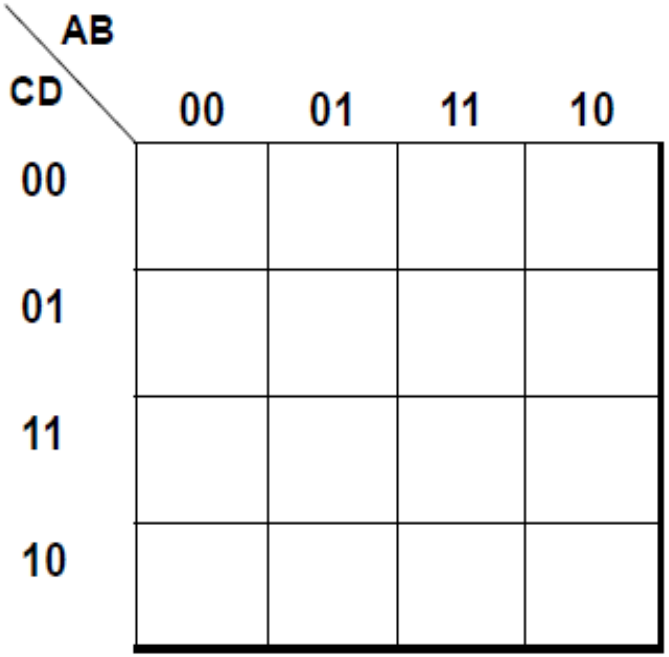


Tableau à 5 variables

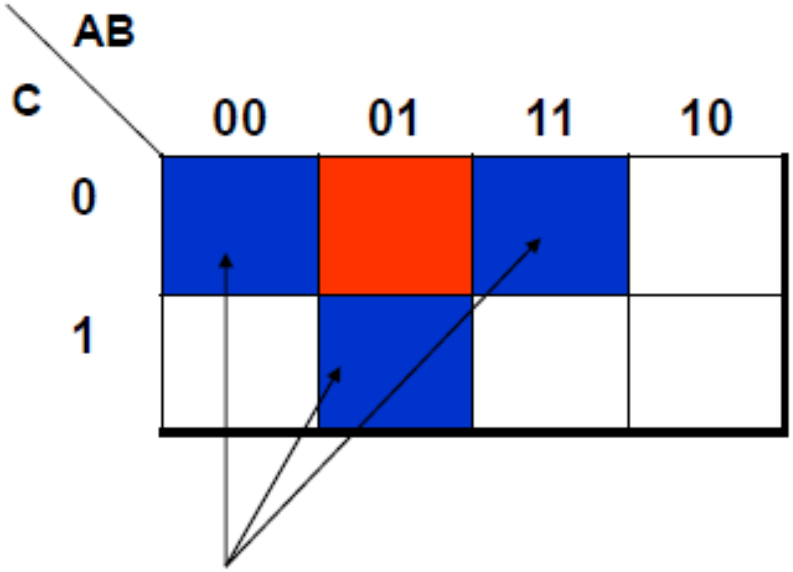
CD \ AB	00	01	11	10
00				
01				
11				
10				

U = 0

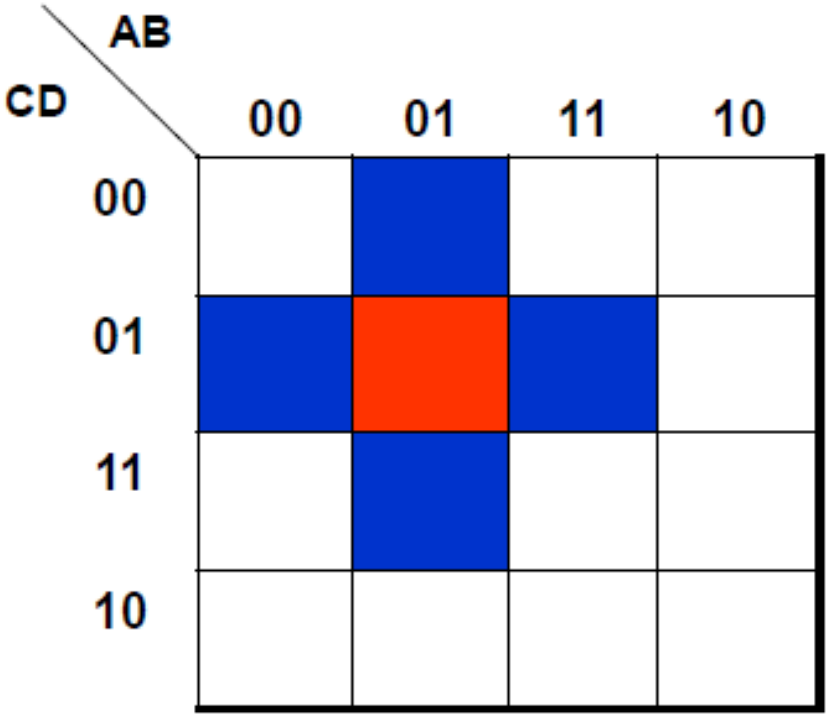
CD \ AB	00	01	11	10
00				
01				
11				
10				

U= 1

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



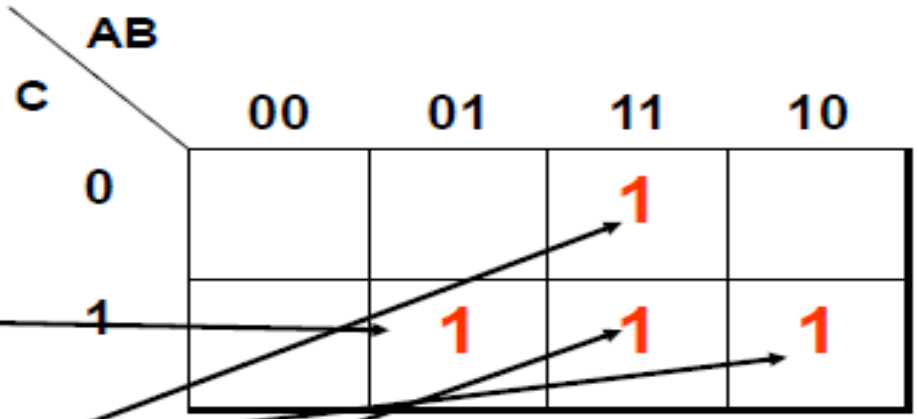
Les trois cases bleues sont des cases adjacentes à la case rouge



Passage de la table de vérité à la table de Karnaugh

Exemple :

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

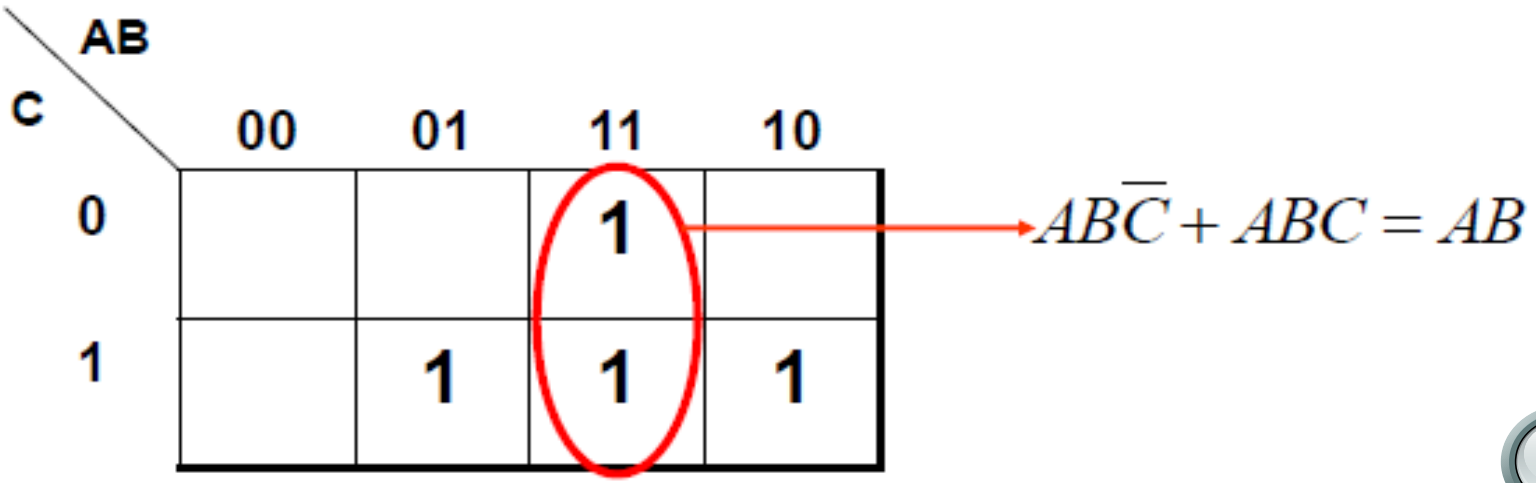


Méthode de simplification (Exemple : 3 variables)

•L'idée de base est d'essayer de regrouper (faire des regroupements) 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)

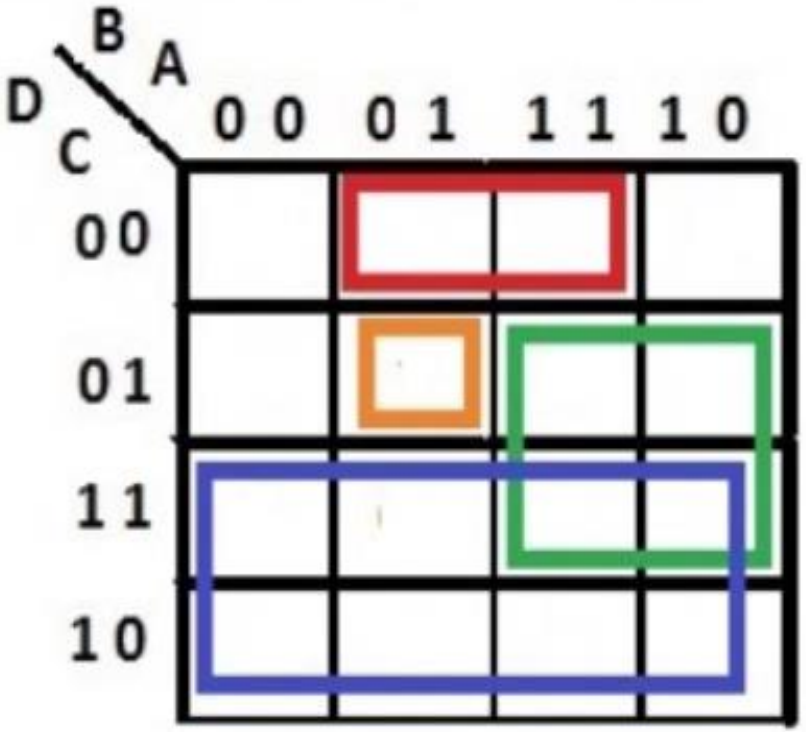
•Dans notre exemple on peut faire uniquement des regroupements de 2 cases .



Les cases regroupées doivent être adjacentes c'est à dire qu'une seule variable change d'une case à l'autre .

Grâce au binaire réfléchi c'est le cas pour les cases voisines !

... mais pas seulement !!!!



1 case

2 cases

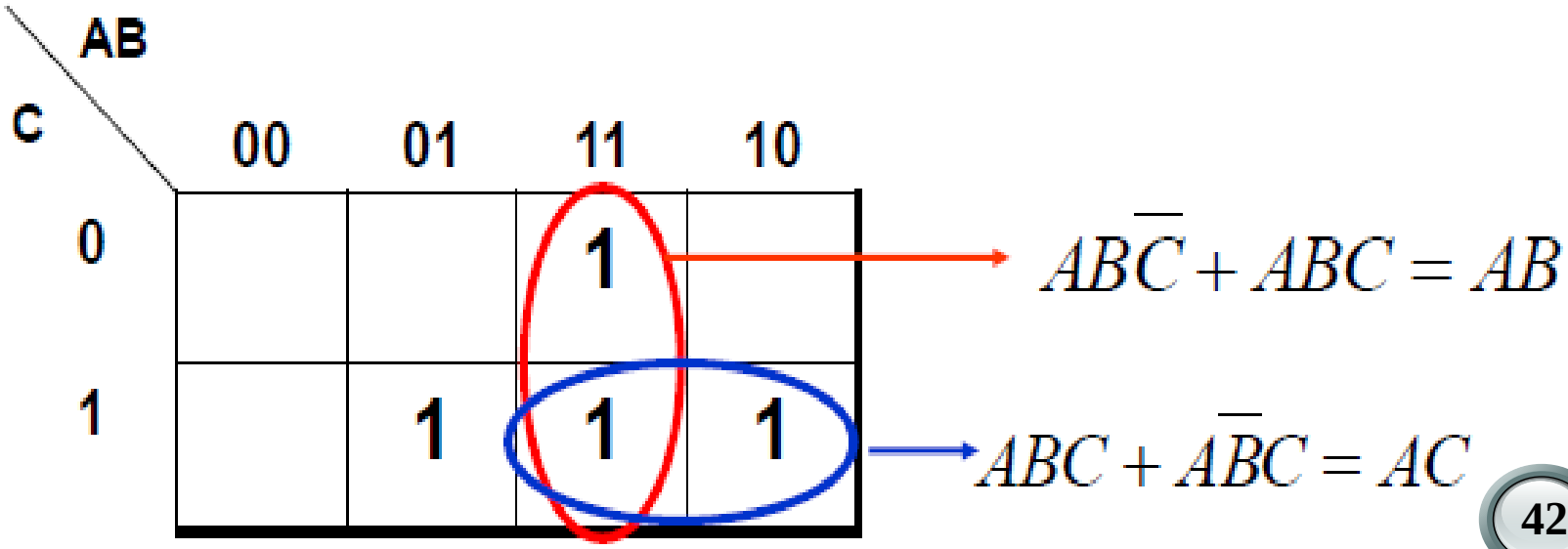
4 cases

8 cases

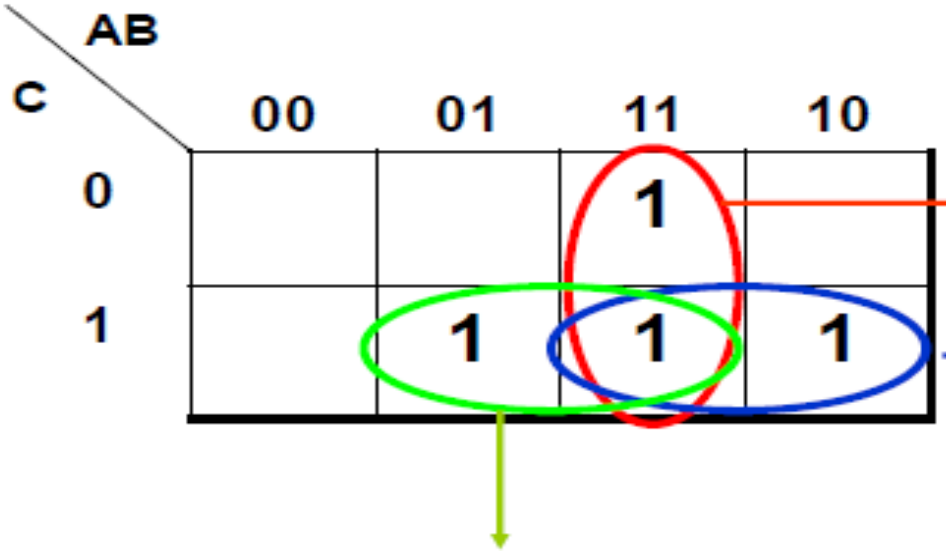
TOUJOURS 2^n CASES

•Puisque il existent 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



•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.



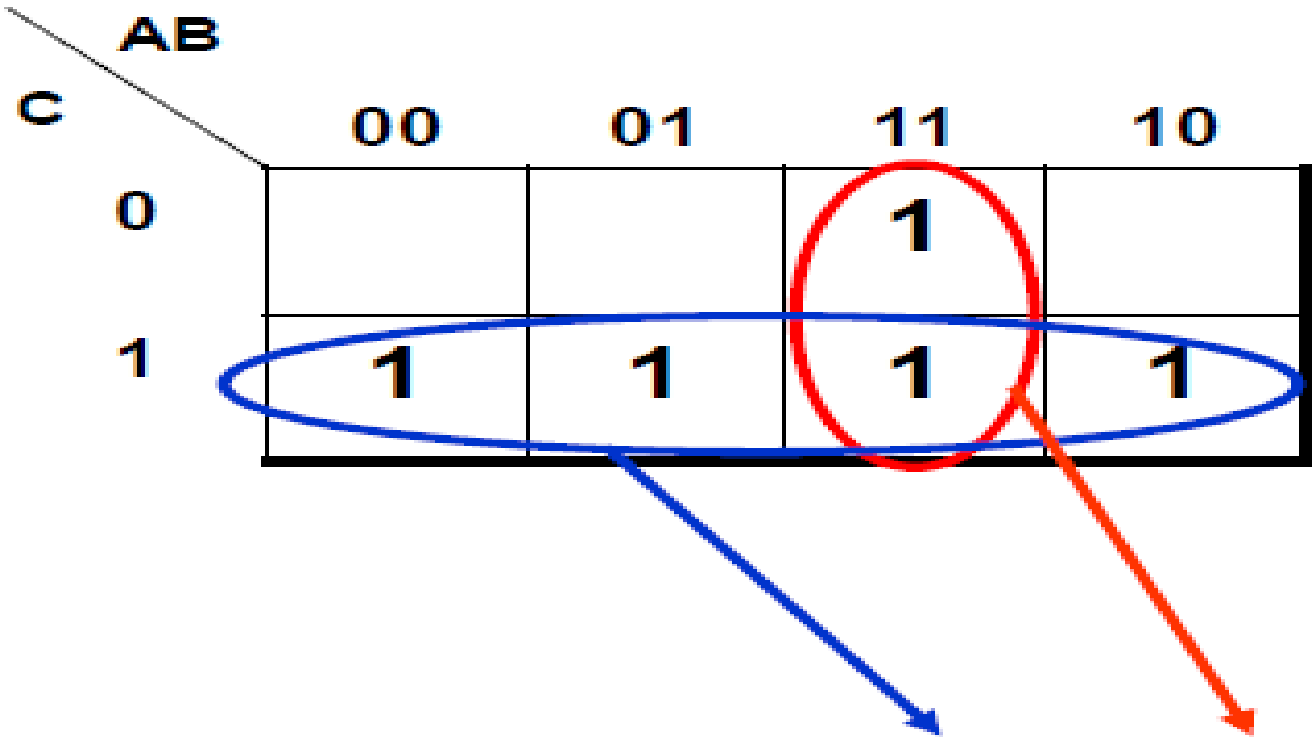
$$AB\overline{C} + ABC = AB$$

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

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

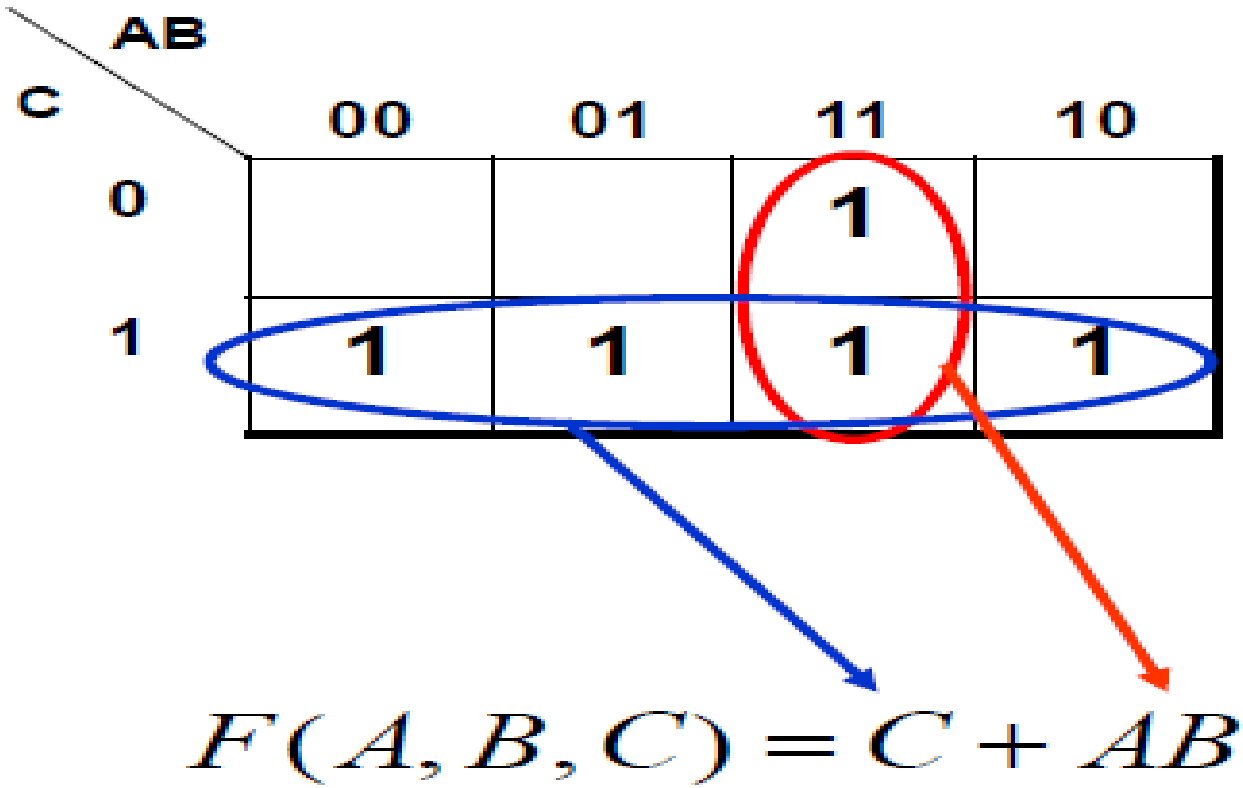
$$F(A,B,C) = AB + AC + BC$$

Exemple 1 : 3 variables



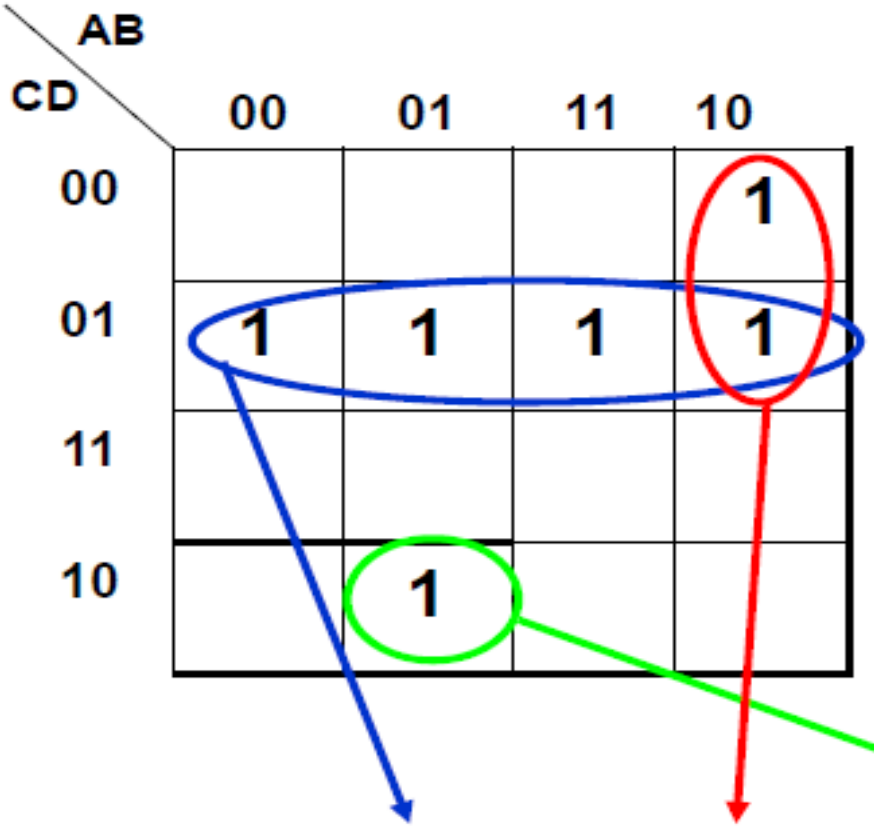
$$F(A,B,C) = C + AB$$

Exemple 1 : 3 variables



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

Exemple 2 : 4 variables



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

Cas d'une fonction non totalement définie

• Examinons l'exemple suivant :

Une serrure de sécurité s'ouvre en fonction de quatre clés A, B, C D. Le fonctionnement de la serrure est définie comme suite :

S(A,B,C,D)= 1 si au moins deux clés sont utilisées

S(A,B,C,D)= 0 sinon

Les clés A et D ne peuvent pas être utilisées en même temps.

•On remarque que si la clé A et D sont utilisées en même temps l'état du système n'est pas déterminé.

•Ces cas sont appelés cas impossibles ou interdites → comment représenter ces cas dans la table de vérité ?.

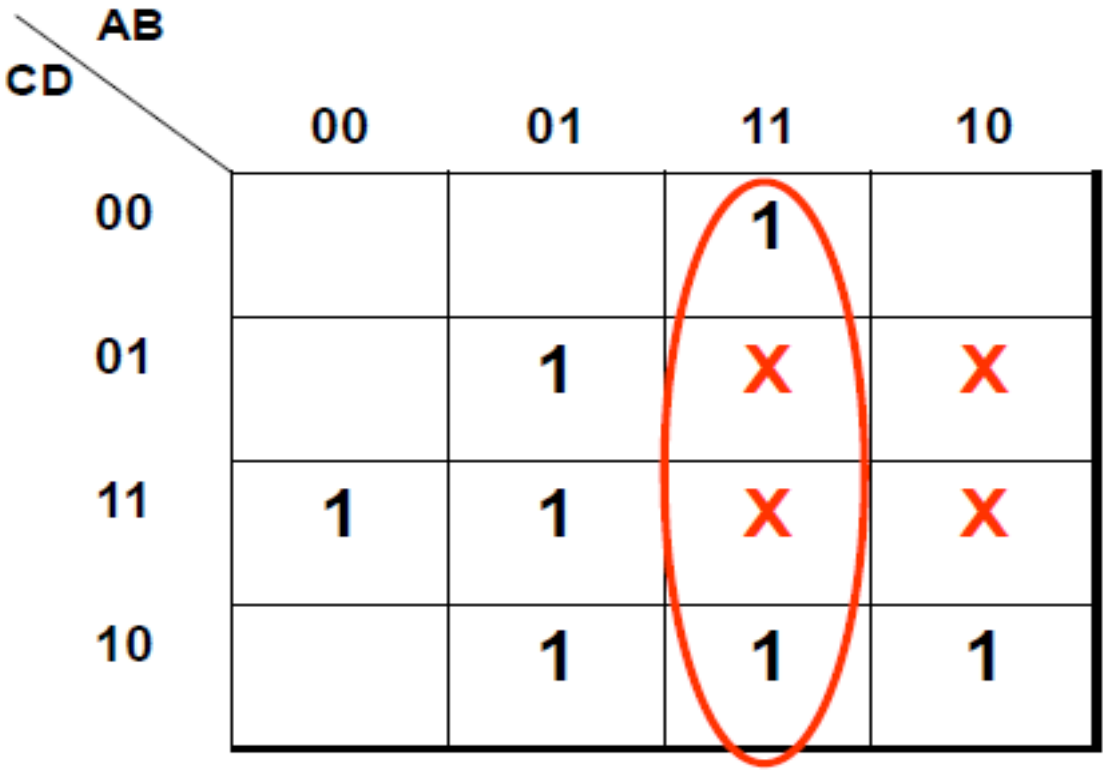
•Pour les cas impossibles ou interdites il faut mettre un X dans la T.V .

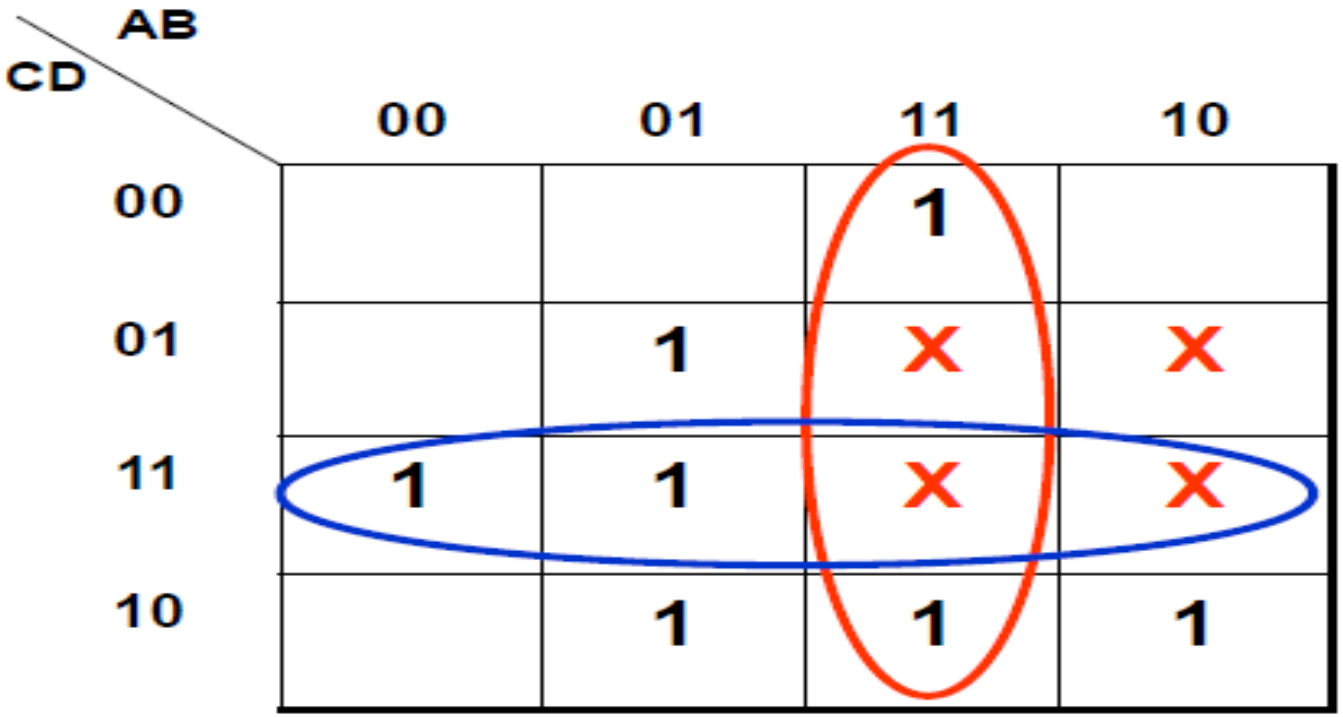
•Les cas impossibles sont représentées aussi par des X dans la table de karnaugh

A	B	C	D		S
0	0	0	0		0
0	0	0	1		0
0	0	1	0		0
0	0	1	1		1
0	1	0	0		0
0	1	0	1		1
0	1	1	0		1
0	1	1	1		1
1	0	0	0		0
1	0	0	1		X
1	0	1	0		1
1	0	1	1		X
1	1	0	0		1
1	1	0	1		X
1	1	1	0		1
1	1	1	1		X

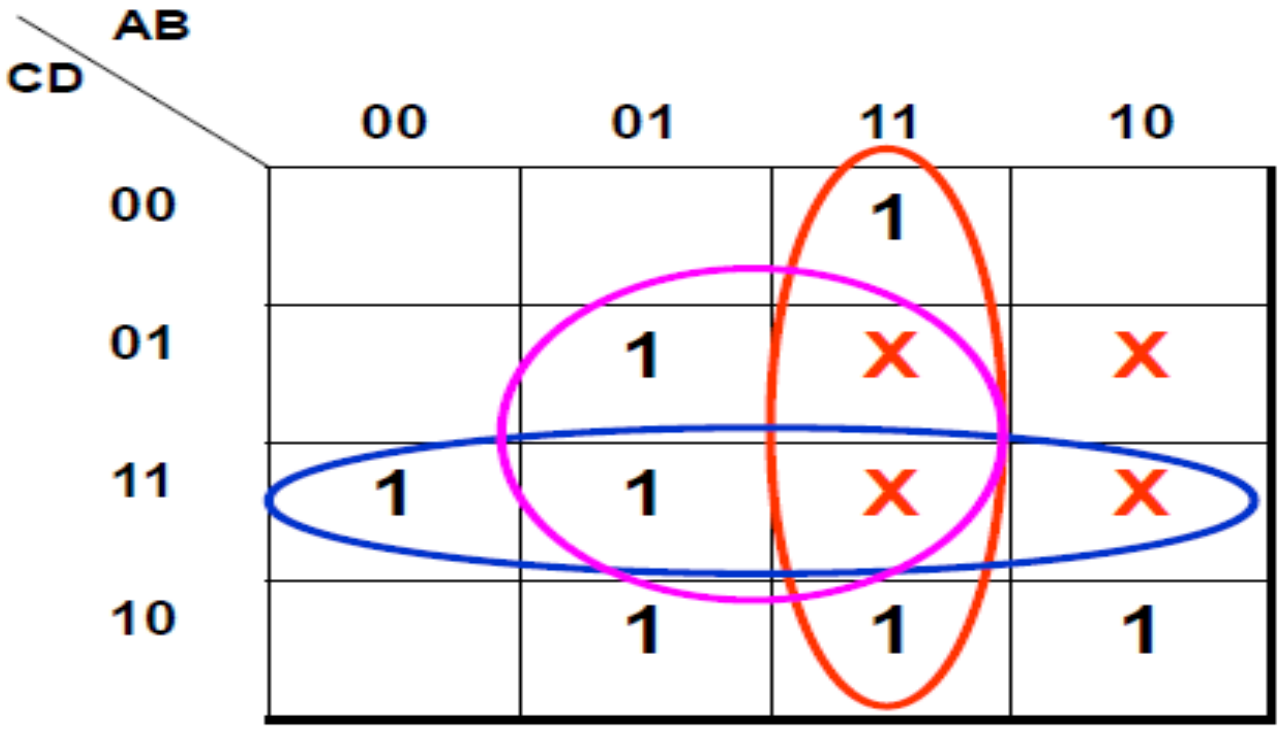
CD \ AB	00	01	11	10
00			1	
01		1	X	X
11	1	1	X	X
10		1	1	1

- Il est possible d'utiliser les X dans des regroupements :
  - Soit les prendre comme étant des 1
  - Ou les prendre comme étant des 0
- Il ne faut pas former des regroupement qui contient uniquement des X





$$AB + CD$$



$$AB + CD + BD$$

Merci pour votre aimable attention