DUT-IDIA **Semestre 1**

Module Architecture des Ordinateurs

Chapitre II

ALGÈBRE DE BOOLE

Pr : Mustapha Johri

Année Universitaire: 2024 - 2025

I. Fonctions logiques

1. Définitions
2. Variables et Fonctions logiques
3. Opérateurs logiques

II. Lois fondamentales d'algèbre de Boole

1. Opérateurs: NON, ET, OU, XOR
2. Distributivité, Associativité, Dualité d'algèbre de Boole…
3. Théorèmes de De Morgan

III. Portes logiques

IV. Formes canoniques d'une fonction logique

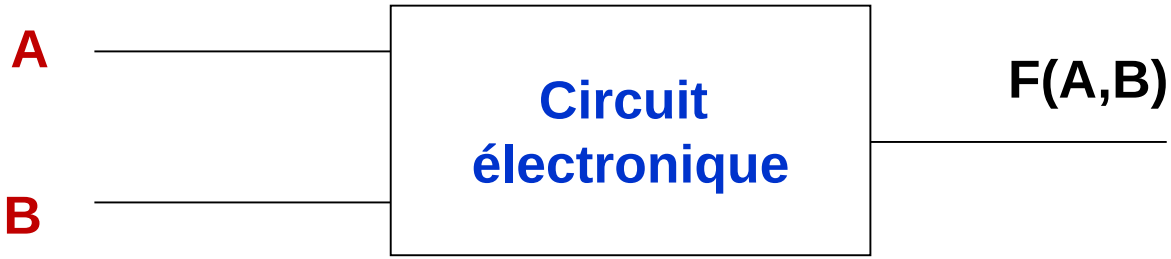
V. Simplification des fonctions logiques

1. Simplification algébrique
2. Tableaux de Karnaugh

FONCTIONS LOGIQUES

1. Définitions

- Les machines numériques sont constituées d'un ensemble de **circuits électroniques**.
- Chaque circuit fournit une **fonction logique** bien déterminée: **addition, comparaison** ,….



- La fonction **F(A,B)** est le **résultat d'un ensemble d'opération effectuées sur les données** A et B.

- Pour **concevoir** et **réaliser** un circuit, on doit avoir **un modèle mathématique** de la fonction à réaliser **par ce circuit**.
- Ce modèle doit prendre en considération le **système binaire**.

1. Définitions

- Le **modèle mathématique** utilisé est celui de **Boole (George Boole** est un **logicien**, **mathématicien** et **philosophe** britannique (**1815-1864**).

- Une **variable logique** (**booléenne**) est une variable qui peut prendre soit la valeur **0** ou **1**.

- Généralement, elle est exprimée par un **seul caractère alphabétique en majuscule ou minuscule** (A , B, s, …).

- Une fonction logique est une fonction qui **relie N variables logiques** à **une seule valeur logique**:

A,B,C,…. → F(A,B,C,….)

2. Variables et fonctions logiques

- Dans **l'Algèbre de Boole**, il existe trois opérateurs de base: **NON (négation), ET (conjonction), OU (disjonction).**

- Si une fonction logique possède **N variables logiques** → la fonction possède **2^n valeurs** (**2^n** combinaisons)**.**

Exemple : avec deux variables logiques A et B on a **2^2 valeurs** possibles qui sont **00, 01, 11, et 10**

- Les **2^n combinaisons** peuvent être représentées dans une table appelée: **table de vérité (TV).**

3. Opérateurs logiques

❑ Négation: NON

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

F(A)= **Non** A = $\overline{A}$ (**A barre**)

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

3. Opérateurs logiques

❑ Conjonction: ET (AND)

- **ET** est un opérateur binaire (deux variables), a pour rôle de réaliser un **produit logique** entre **deux variables booléennes.**
- **ET** fait la **conjonction** entre deux variables.

- **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

3. Opérateurs logiques

❑ Disjonction: OU (OR)

- **OU** est un opérateur binaire (deux variables) , a pour rôle de réaliser la **somme logique** entre **2 variables logiques.**
- **OU** fait la **disjonction** entre deux variables.
- **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

3. Opérateurs logiques

- Les opérateurs ET et OU peuvent réaliser le produit et la somme de **plusieurs variables** logiques.
- Dans une expression, on peut aussi utiliser les **parenthèses**.
- Pour évaluer une **expression logique** (**fonction logique**):
  - ❖ Evaluer les sous expressions entre les **parenthèses**.
  - ❖ Puis le **complément**: NON,
  - ❖ En suite le **produit** logique: ET,
  - ❖ Enfin la **somme** logique: OU.

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

si on veut calculer $F(0,1,1)$ alors :

$$F(0,1,1) = (\overline{0.1})(1+1) + 0.\overline{1}.1$$

$$= (\overline{0})\,(1) + 0.0.1$$

$$= 1.1 + 0.0.1$$

$$= 1 + 0$$

$$= 1$$

❑ **Exercice**

Trouver la table de vérité de cette fonction?

3. Opérateurs logiques

- Pour trouver la **table de vérité**, il faut trouver **la valeur de la fonction F** pour chaque combinaison des trois variables A, B , C.
- 3 variables → 2^3 = **8 combinaisons.**

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

$$F(0,0,0) = (\overline{0.0}).(0+0) + 0.\overline{0}.0 = 0$$
$$F(0,0,1) = (\overline{0.0}).(1+0) + 0.\overline{0}.1 = 1$$
$$F(0,1,0) = (\overline{0.1}).(0+1) + 0.\overline{1}.0 = 1$$
$$F(0,1,1) = (\overline{0.1}).(1+1) + 0.\overline{1}.1 = 1$$
$$F(1,0,0) = (\overline{1.0}).(0+0) + 1.\overline{0}.0 = 0$$
$$F(1,0,1) = (\overline{1.0}).(1+0) + 1.\overline{0}.1 = 1$$
$$F(1,1,0) = (\overline{1.1}).(0+1) + 1.\overline{1}.0 = 0$$
$$F(1,1,1) = (\overline{1.1}).(1+1) + 1.\overline{1}.1 = 0$$

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

LOIS FONDAMENTALES D'ALGÈBRE DE BOOLE

❑Opérateur NON

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

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

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

Théorème de DE MORGAN

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

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

❑ Opérateur OU

$$(A + B) + C = A + (B + C) = A + B + C \quad \text{Associativité}$$

$$A + B = B + A \quad \text{Commutativité}$$

$$A + A = A \quad \text{Idempotence}$$

$$A + 0 = A \quad \text{Elément neutre}$$

$$A + 1 = 1 \quad \text{Elément absorbant}$$

❑ Opérateur ET

$(A.B).C = A.(B.C) = A.B.C$	Associativité
$A \,.\, B = B \,.\, A$	Commutativité
$A \,.\, A = A$	Idempotence
$A \,.\, 1 = A$	Elément neutre
$A \,.\, 0 = 0$	Elément absorbant

❑ Exercice

Soit la fonction logique suivante :

$$F(a, b, c) = (a + b).(\bar{a} + b + c)$$

Calculer $\bar{F}$?

Développer F ?

❑ Autres opérateurs logiques

❑ OU exclusif XOR

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

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

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

❑ Exercice

Calculer

$$(A + B).(\bar{A} + \bar{B})$$

❑ NON ET (NAND)

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

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

❑ NON OU (NOR)

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

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

- En utilisant les **NAND** et les **NOR**, on peut **exprimer** n'importe quelle **expression** (fonction) logique.
- Pour cela, il suffit **d'exprimer les opérateurs de base** (**NON**, **ET**, **OU**) avec des **NAND** et des **NOR**.

❑ Distributivité

$$A.(B+C)=(A.B)+(A.C) \quad \text{Distributivité du ET sur le OU}$$

$$A+(B.C)=(A+B).(A+C) \quad \text{Distributivité du OU sur le ET}$$

❑ Autres relations utiles

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

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

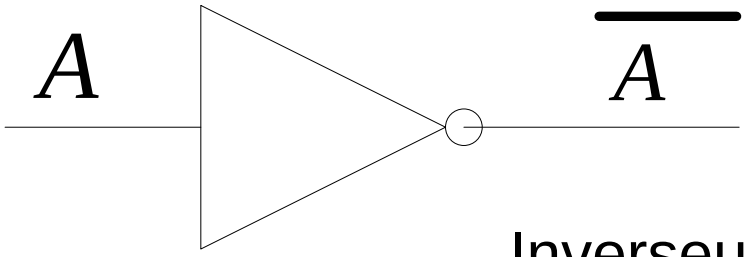
$$(A + B) \,.\, (A + \overline{B}) = A$$

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

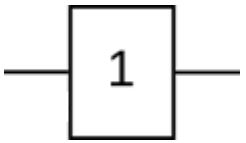
PORTES LOGIQUES

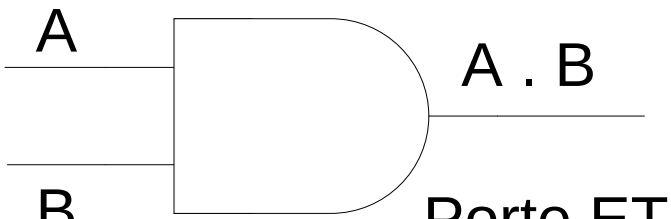
❑ Opérateurs logiques de base

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

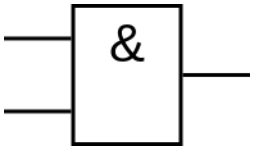


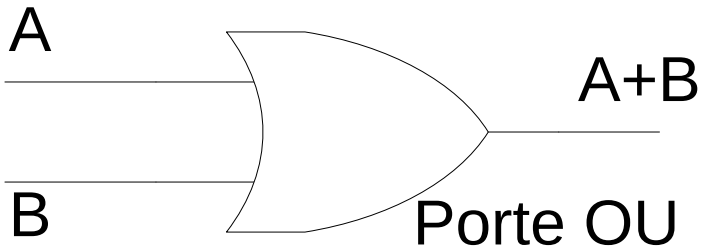
Inverseur



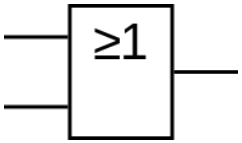


Porte ET

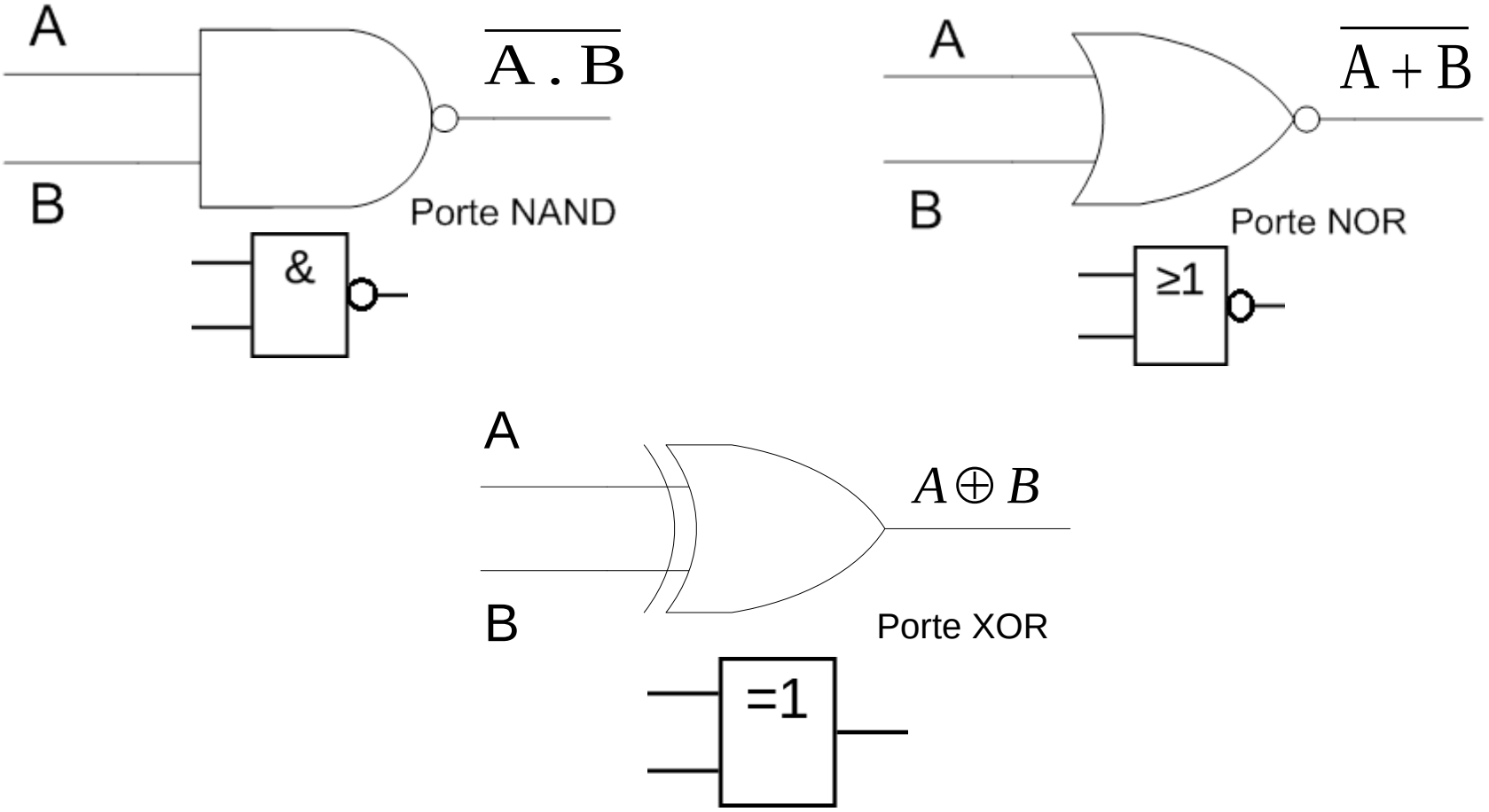




Porte OU



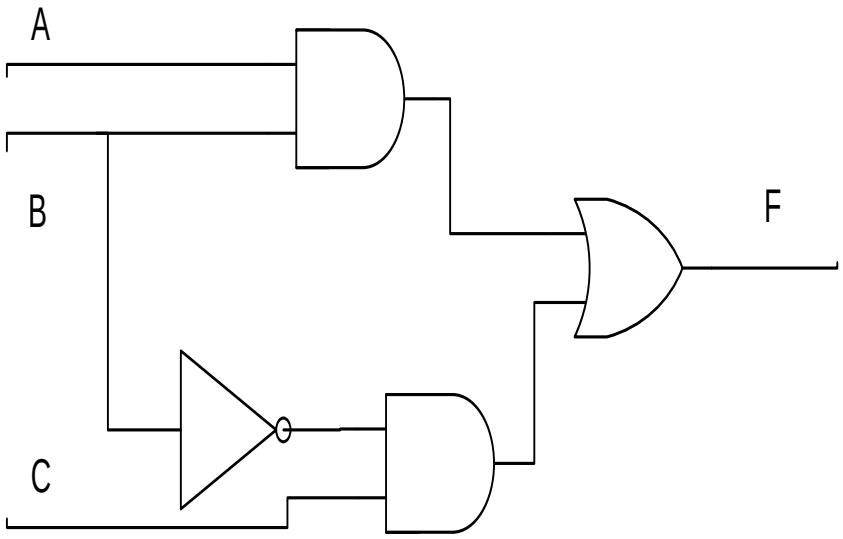
❑ Opérateurs logiques de base



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

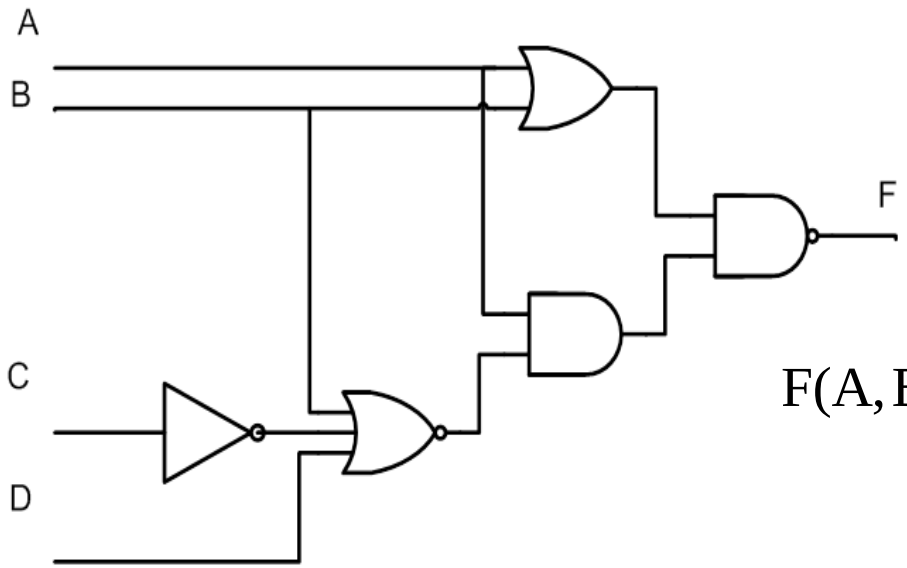
❑Logigramme

- C’est la traduction de la fonction logique en un **schéma électronique.**
- Le principe consiste à remplacer chaque **opérateur logique** par une **porte logique** correspondante.



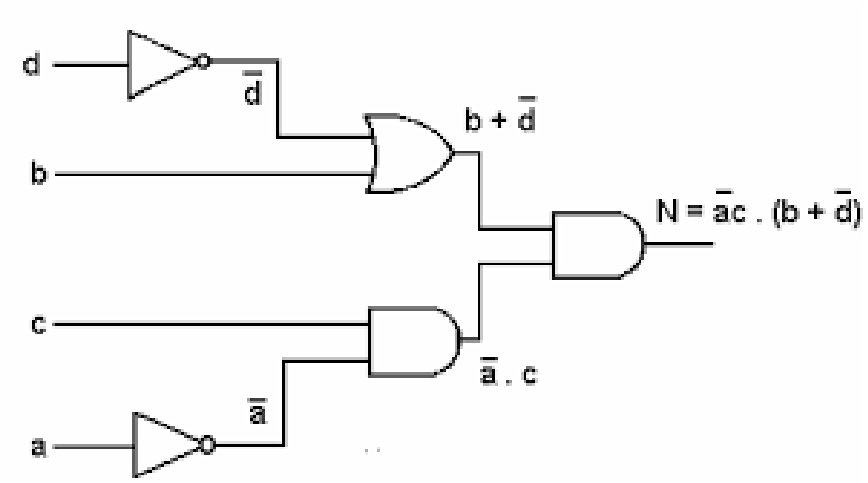
$$F(A,B,C) = A.B + \overline{B}.C$$

❑**Exercice :** Trouver la fonction logique F



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

❑**Exercice :** Tracer logigramme de la fonction $N = \bar{a}c.(b+\bar{d})$



FORMES CANONIQUES

❑ Forme canonique d'une fonction logique

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**

$\longrightarrow$ $A + B + C$: max terme

$\longrightarrow$ $A + B + \overline{C}$: max terme

$\longrightarrow$ $A + \overline{B} + C$: max terme

$\longrightarrow$ $\overline{A} . B . C$: min terme

$\longrightarrow$ $\overline{A} + B + C$: max terme

$\longrightarrow$ $A . \overline{B} . C$: min terme

$\longrightarrow$ $A . B . \overline{C}$: min terme

$\longrightarrow$ $A . B . C$: min terme

❑ Forme canonique d'une fonction logique

- **Somme des min termes**

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

- **Produit des max termes**

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

- Une **forme canonique** d'une fonction est la forme dont chaque **terme** comporte **toutes les variables.**

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

- On distingue entre **la première** et la **deuxième forme**.

❑ Forme canonique d'une fonction logique

- **La 1ère forme canonique** ou la **forme disjonctive** est une somme de produits **(somme des min termes)**. C'est la **disjonction de conjonctions**.

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

- **La 2ème forme canonique** ou la **forme conjonctive** est un produit de sommes (**produit des max termes). C'est la conjonction de disjonctions**.

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

❑ Forme canonique d'une fonction logique

1. $F(A, B) = A + B$

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

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

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

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

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

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

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

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

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

SIMPLIFICATION DES FONCTIONS LOGIQUES

1. Simplification algébrique

- L'objectif de la simplification des fonctions logiques est de:
  - ❖ Réduire le **nombre de termes** dans une fonction.
  - ❖ 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:
  - ❖ **Méthodes algébriques.**
  - ❖ **Méthodes graphiques: (Tableaux de karnaugh).**

2. Tableaux de karnaugh

❑ Termes adjacents

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

- Les deux termes possèdent les mêmes 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**.

2. Tableaux de karnaugh

❑ Termes adjacents

❖ **Ces termes sont adjacents.**

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

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

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

❖ **Ces termes ne sont pas adjacents.**

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

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

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

2. Tableaux de karnaugh

❑ Description

- La **méthode de Karnaugh** se base sur la **règle précédente (termes adjacents).**

- 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).

2. Tableaux de karnaugh

❑ Description

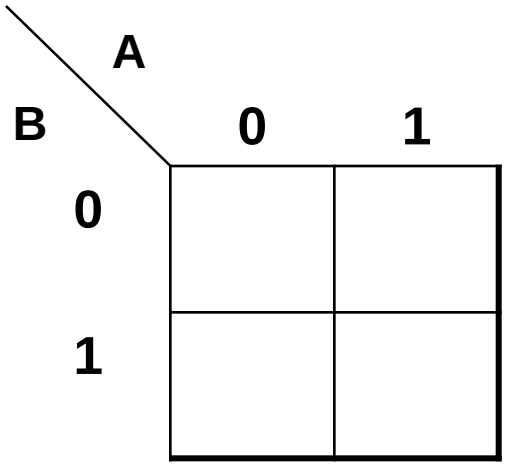
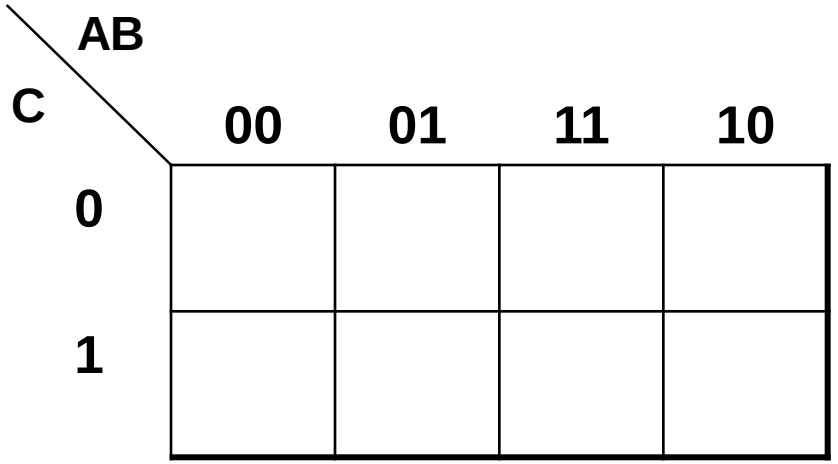


Tableau à 2 variables



Tableaux à 3 variables

2. Tableaux de karnaugh

❑ Description

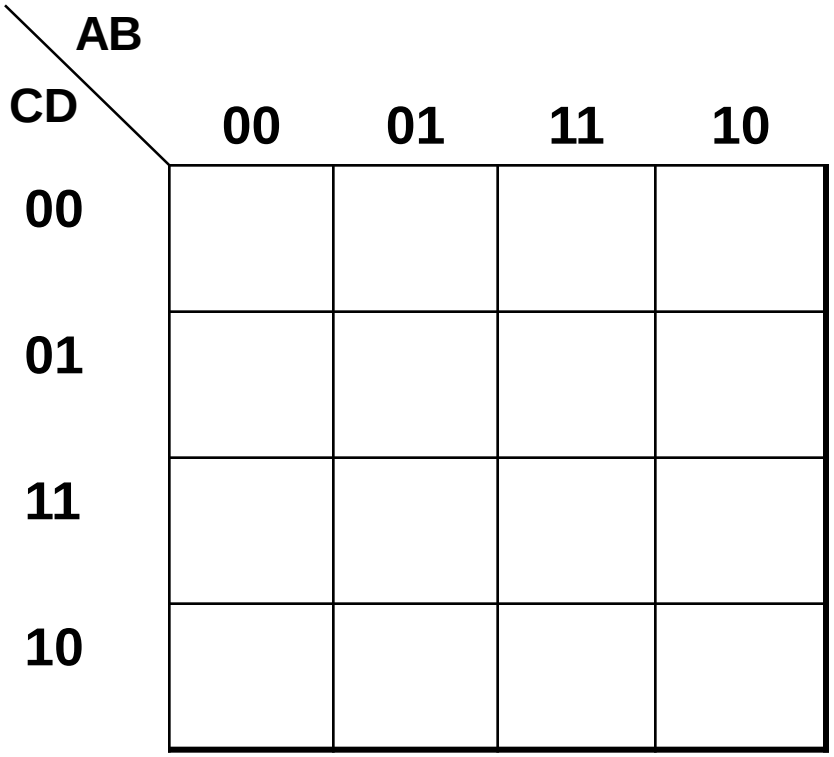
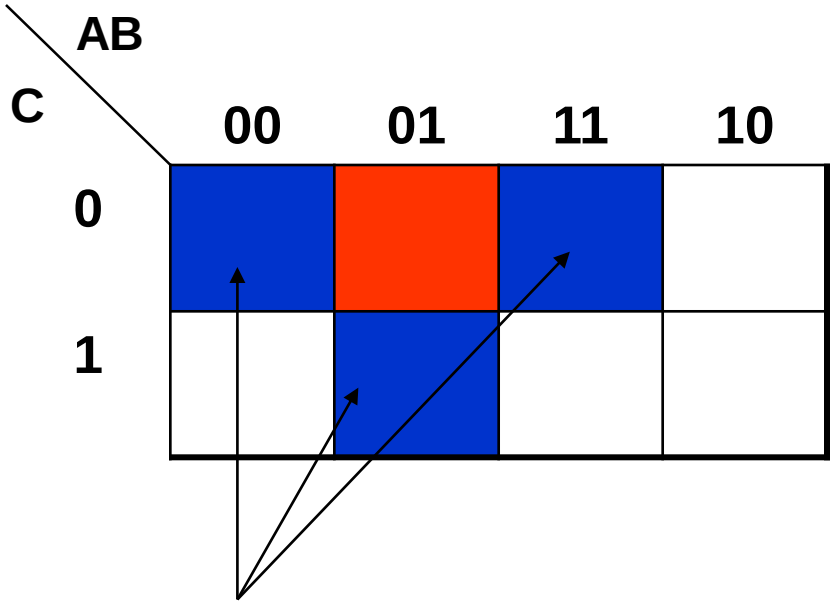


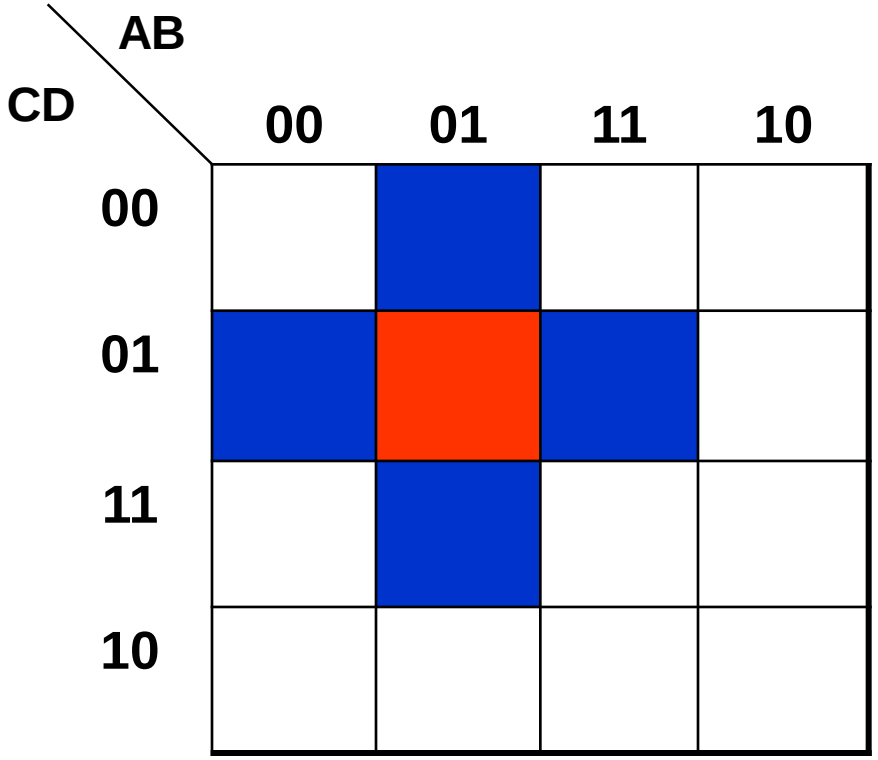
Tableau à 4 variables

2. Tableaux de karnaugh

❑ Description

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



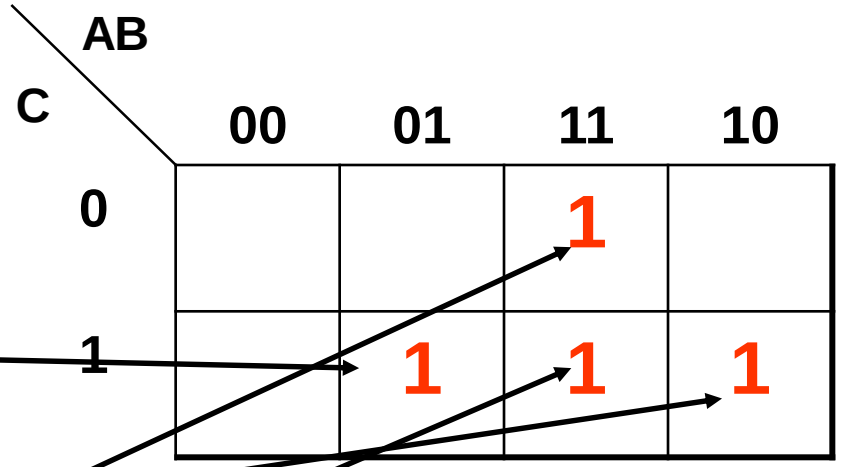


- Les trois cases **bleues** sont des **cases adjacentes** à la case **rouge.**

2. Tableaux de karnaugh

❑ Passage de la TV au tableau de Karnaugh

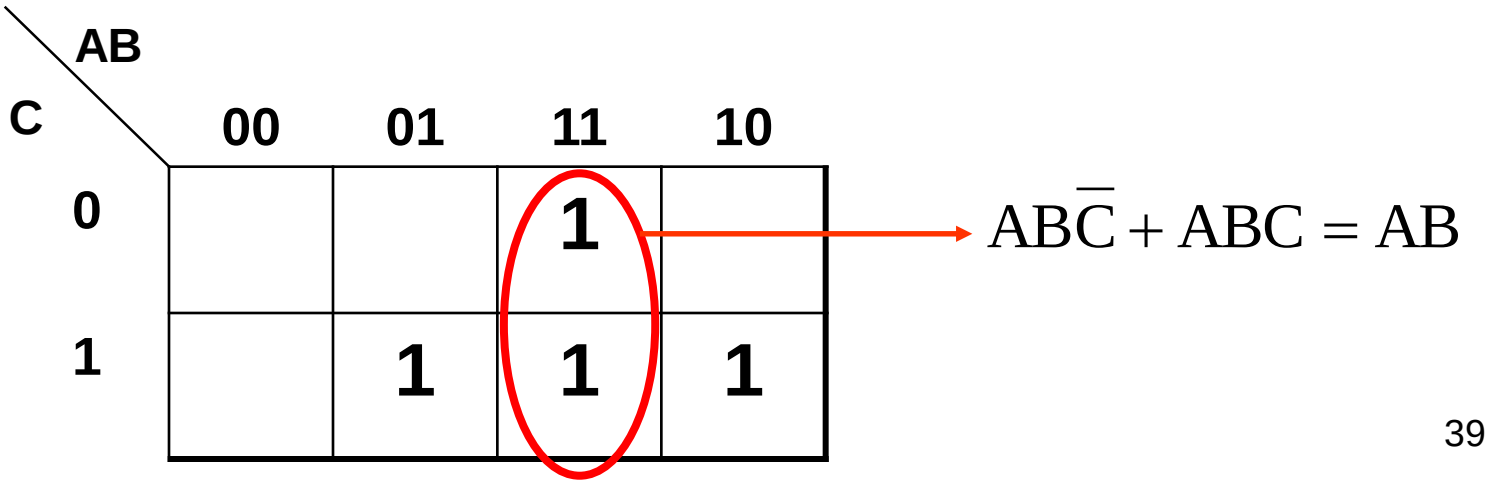
A	B	C		S
0	0	0		**0**
0	0	1		**0**
0	1	0		**0**
0	1	1		**1**
1	0	0		**0**
1	0	1		**1**
1	1	0		**1**
1	1	1		**1**



2. Tableaux de karnaugh

❑ Méthode de simplification: 3 variables

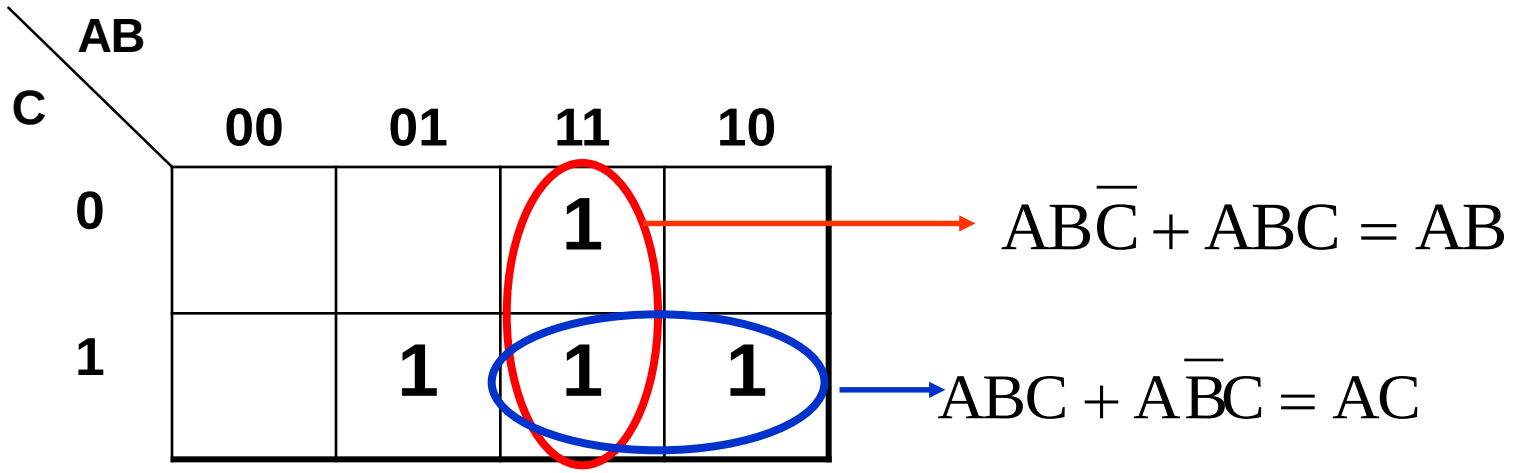
- L'idée de base est d'essayer de regrouper (**regroupements**) les **cases adjacentes** qui comportent des **1** (**rassembler les termes adjacents**).
- Faire des regroupements avec le **maximum** de cases (**16,8,4 ou 2**).
- Dans notre exemple, on peut faire uniquement des regroupements de 2 cases .



2. Tableaux de karnaugh

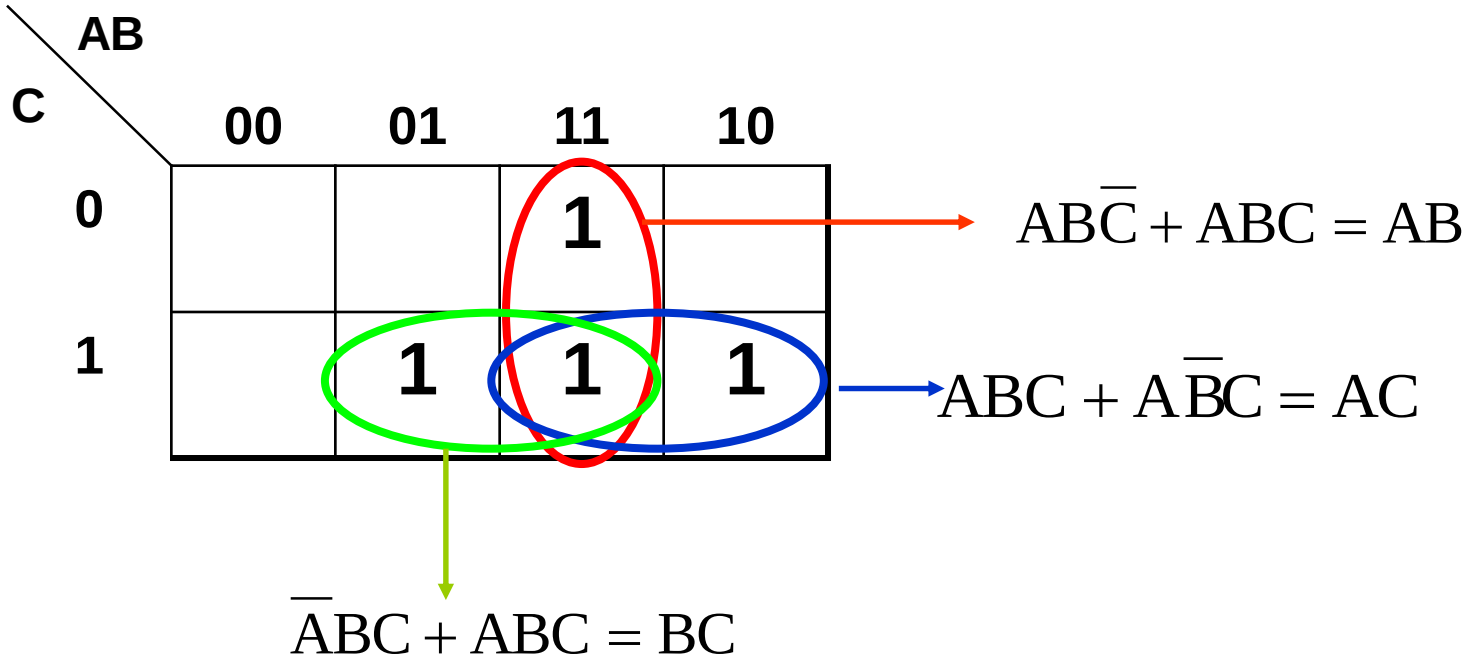
❑ Méthode de simplification: 3 variables

- Puisqu'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**.



2. Tableaux de karnaugh

❑ Méthode de simplification: 3 variables

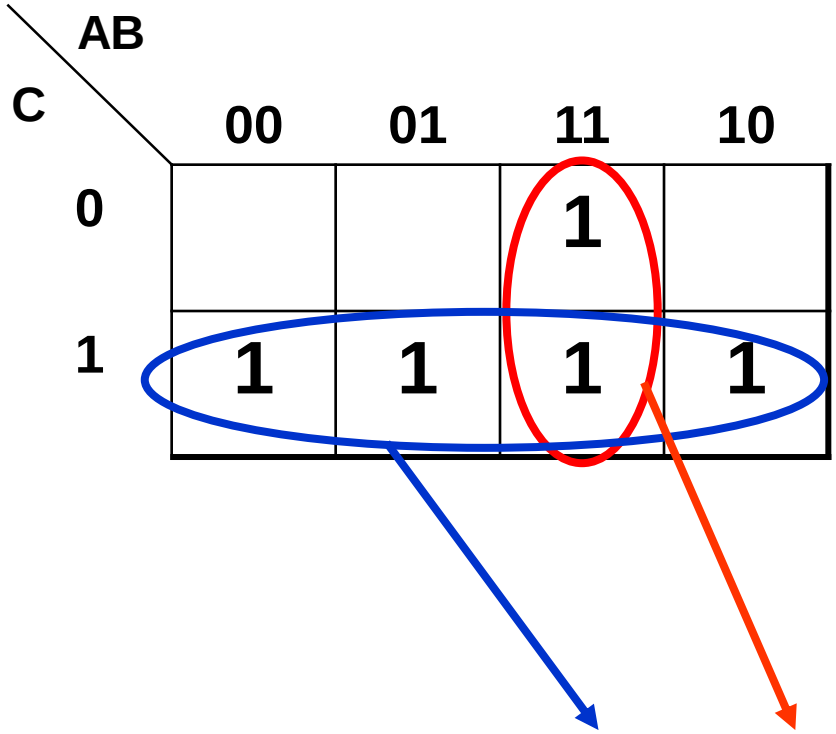


$$F(A, B, C) = AB + AC + BC$$

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

2. Tableaux de karnaugh

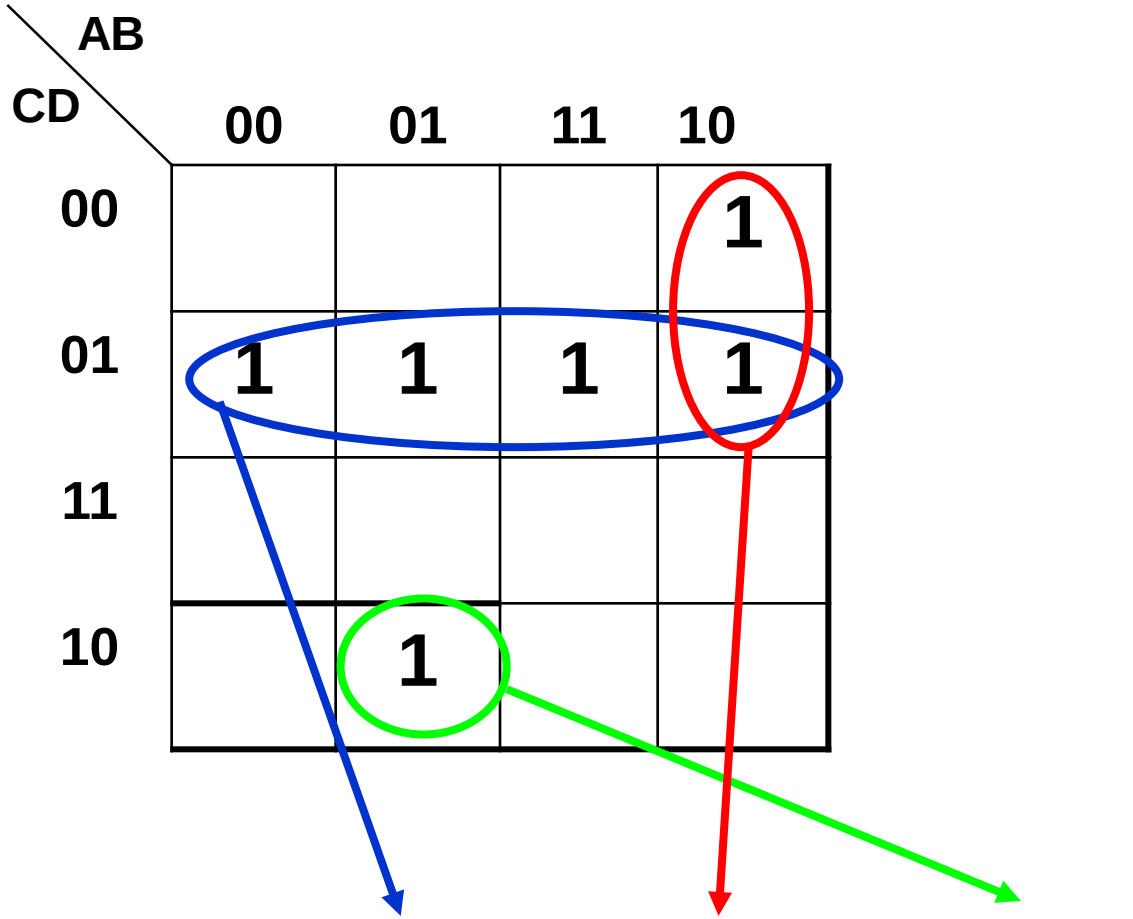
- **Méthode de simplification: 3 variables**



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

2. Tableaux de karnaugh

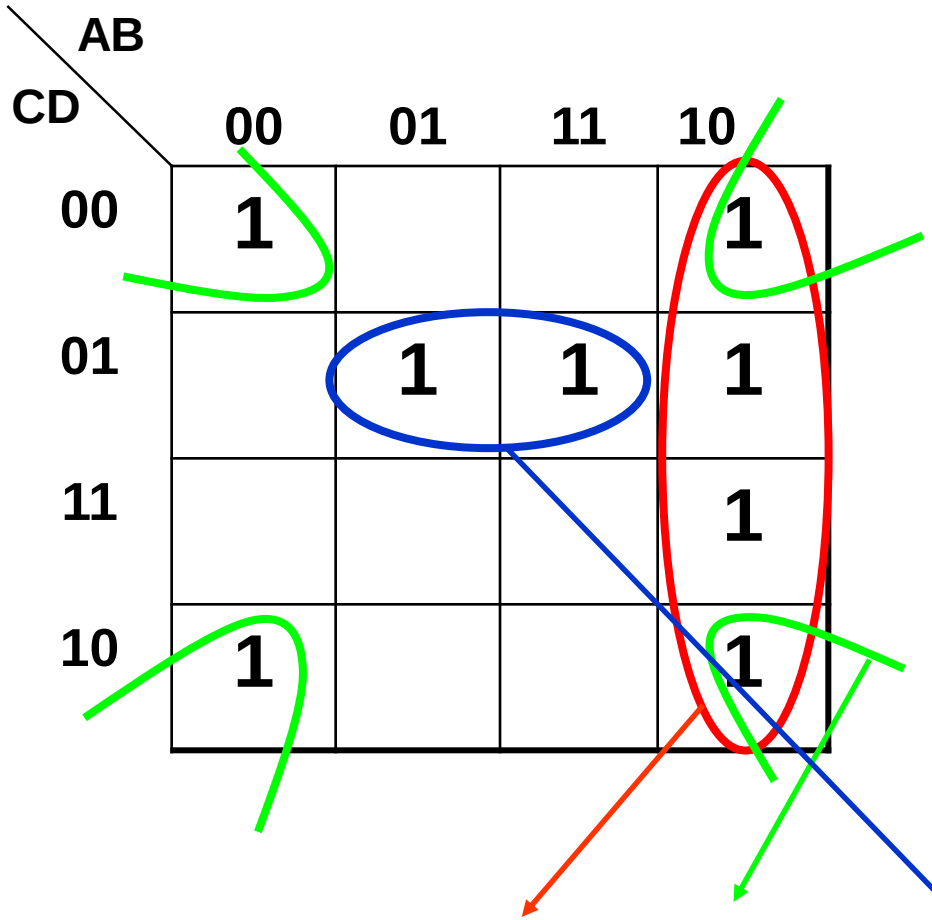
❑ Méthode de simplification: 4 variables



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

2. Tableaux de karnaugh

- **Méthode de simplification: 4 variables**



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

2. Tableaux de karnaugh

❑ Méthode de simplification

Trouver la forme simplifiée des fonctions à partir des deux tableaux ?

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

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

2. Tableaux de karnaugh

1. **Remplir** le tableau à partir de la **table de vérité** ou à partir de la **forme canonique**.
2. Faire des **regroupements**: des regroupements de **16,8,4,2,1** cases. Les **même termes** peuvent participer à plusieurs **regroupements**.
3. Dans un **regroupement**:
   - Qui contient **un seule terme**, on peut pas éliminer de variables.
   - Qui contient **deux termes,** on peut éliminer **une variable**.
   - Qui contient **4 termes** on peut éliminer **2 variables**.
   - Qui contient **8 termes** on peut éliminer **3 variables**.
   - Qui contient **16 termes** on peut éliminer **4 variables**.
5. L'expression **logique finale** est la réunion (la somme) des groupements après simplification et élimination des variables qui changent d'état.

2. Tableaux de karnaugh

❑ Cas d'une fonction non totalement définie

❑Exemple

- 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 le fonctionnement de cette serrure dans la table de vérité ?

2. Tableaux de karnaugh

- Pour les cas impossibles ou interdites, il faut mettre un **X** dans la T.V.
- Les cas impossibles sont représentés aussi par des **X** dans la table de karnaugh.

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

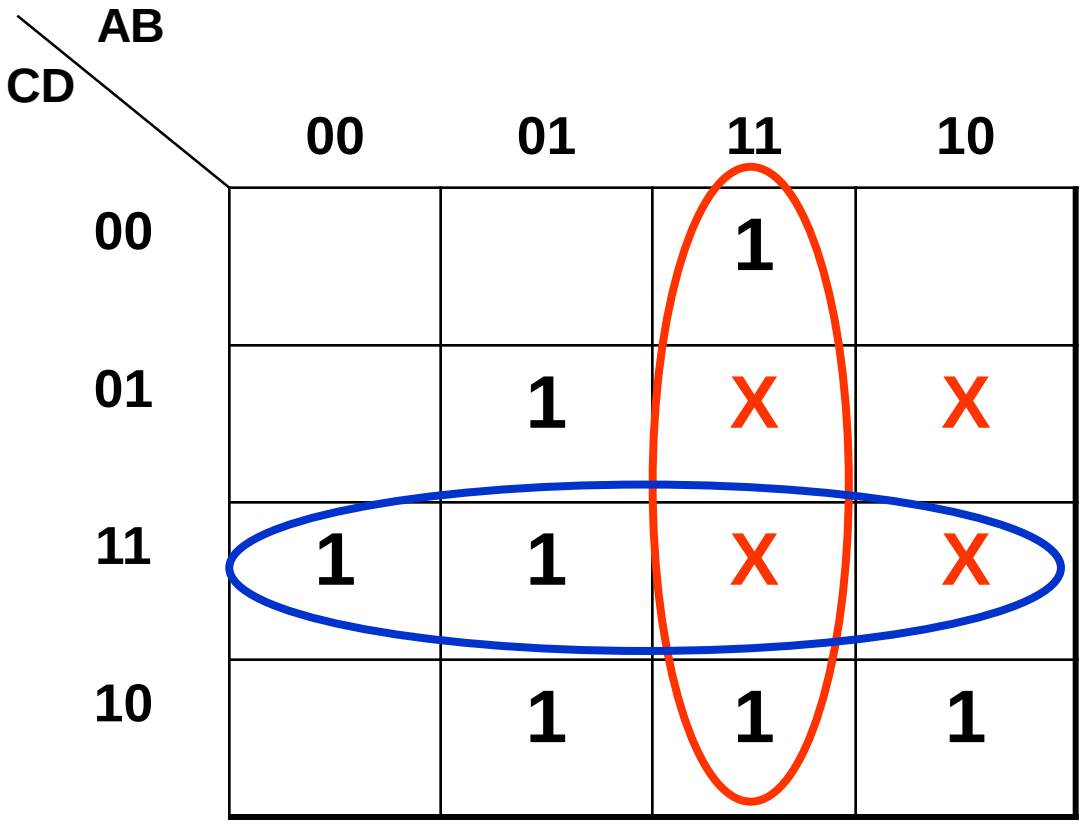
A	B	C	D		S
0	0	0	0		0
0	0	0	1		0
0	0	1	0		0
0	0	1	1		1
0	1	0	0		0
0	1	0	1		1
0	1	1	0		1
0	1	1	1		1
1	0	0	0		0
1	0	0	1		X
1	0	1	0		1
1	0	1	1		X
1	1	0	0		1
1	1	0	1		X
1	1	1	0		1
1	1	1	1		X

2. Tableaux de karnaugh

- Il est possible d'utiliser les **X** dans des regroupements:
  - ❖Soit les prendre comme étant des **1**.
  - ❖Ou les prendre comme étant des **0**.

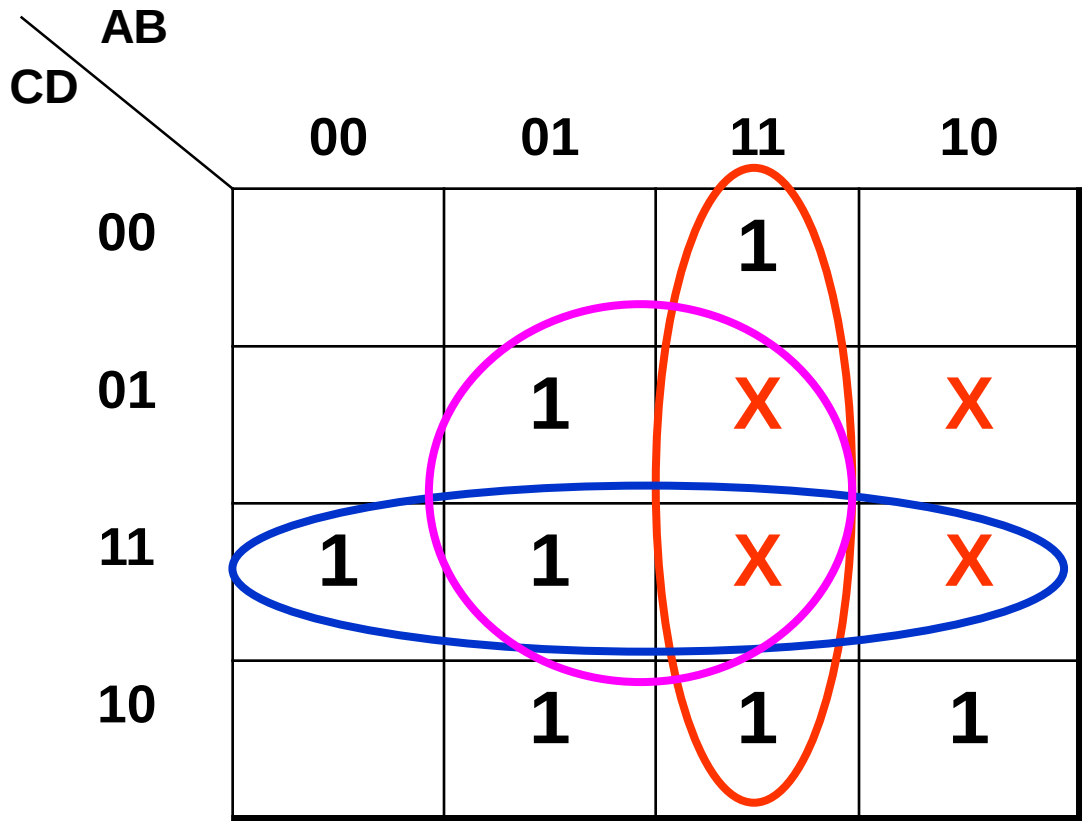
- Il ne faut pas former des regroupement qui contient uniquement des **X**.

2. Tableaux de karnaugh



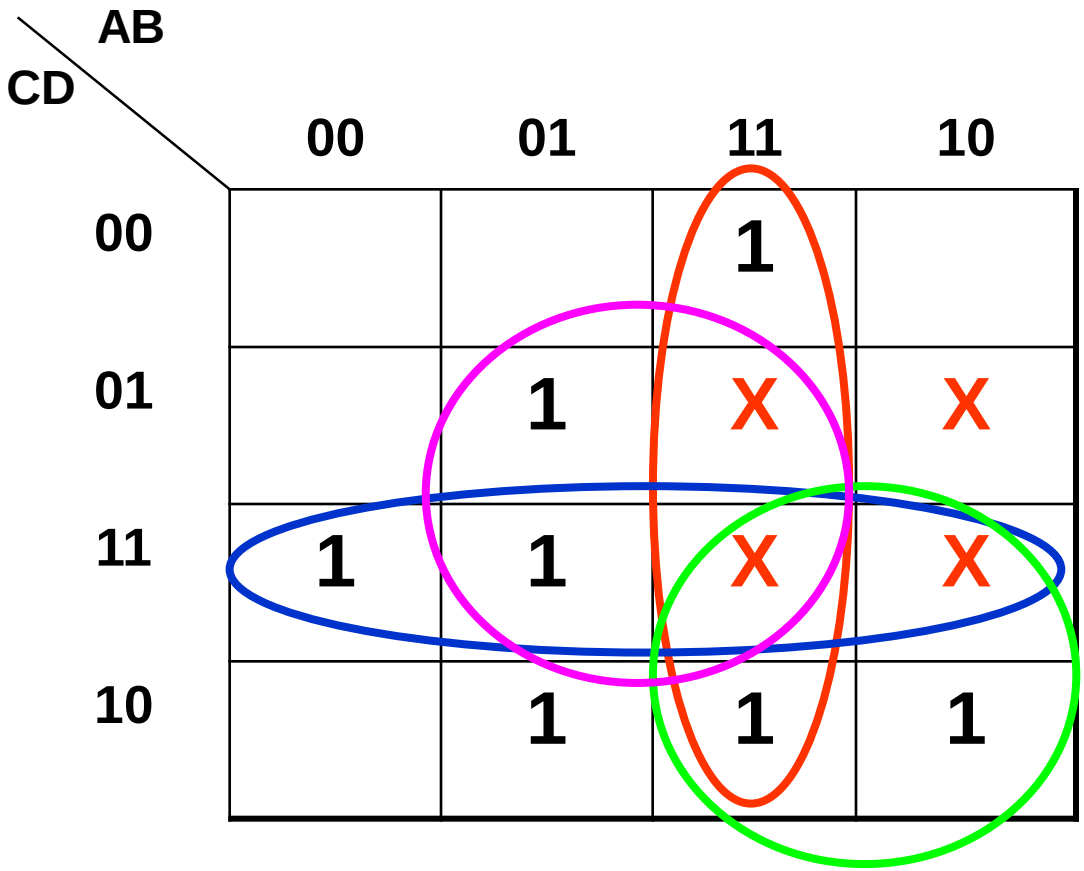
$$AB + CD$$

2. Tableaux de karnaugh



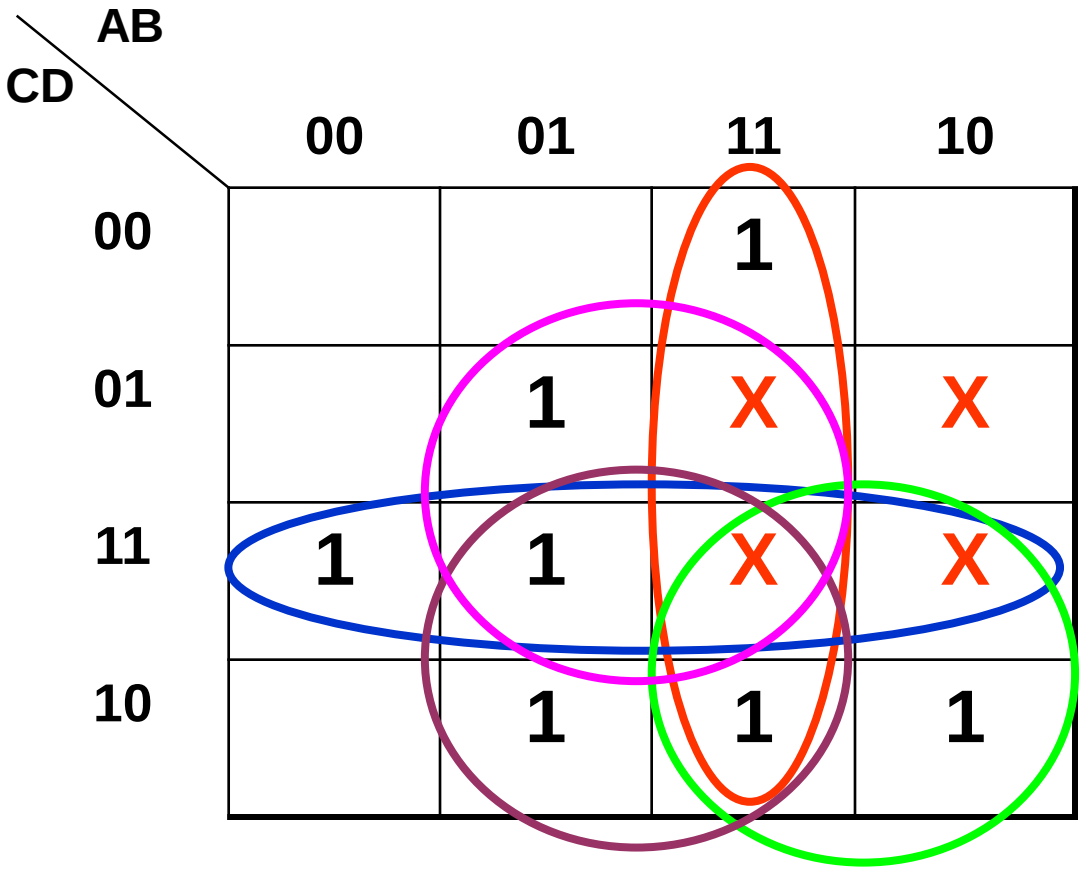
$$AB + CD + BD$$

2. Tableaux de karnaugh



$$AB + CD + BD + AC$$

2. Tableaux de karnaugh



$$AB + CD + BD + AC + BC$$

FIN