CENTRE UNIVERSITAIRE ABDELHAFID BOUSSOUF-MILA

MODULE : STRUCTURE MACHINE 1

CHAPITRE 4

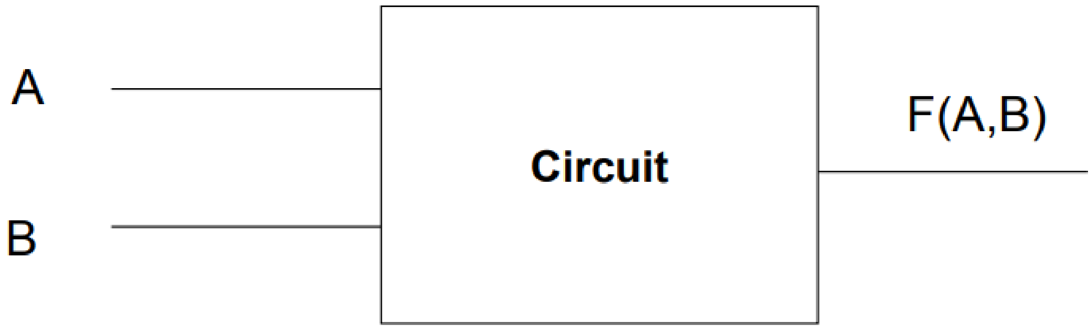
L'ALGÈBRE DE BOOLE BINAIRE

ENSEIGNÉ PAR : BOUMASSATA MERIEM

2022/2023

1. Introduction

- Les machines numériques sont constituées d'un ensemble de circuits électroniques.
- Chaque circuit fournit une fonction logique bien déterminée; opérations logiques ou arithmétiques (addition, soustraction, comparaison, ...).



- 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 E1et 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).

2. Algèbre de Boole (Suite)

- 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 (High) 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. Définitions et conventions (Suite)

3.2. Variable logique (booléenne)

- 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 (A, B, S, …).
- Exemple :

➢ Une lampe : allumée : L = 1

éteinte : L = 0

➢ Premier interrupteur : ouvert : I1 = 1

fermé : I1 = 0

➢ 2éme interrupteur : ouvert : I2 = 1

fermé : I2 = 0

3. Définitions et conventions (Suite)

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).

3. Définitions et conventions (Suite)

3.3. Fonction logique (Suite)

- Exemple d'une fonction logique :

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

La fonction possède 3 variables $\rightarrow 2^3$ combinaisons

$F(0,0,0) = \bar{0}.\bar{0}.0 + \bar{0}.0.0 + 0.\bar{0}.0 + 0.0.0 = 0$

$F(0,0,1) = \bar{0}.\bar{0}.1 + \bar{0}.0.1 + 0.\bar{0}.1 + 0.0.1 = 1$

$F(0,1,0) = \bar{0}.\bar{1}.0 + \bar{0}.1.0 + 0.\bar{1}.0 + 0.1.0 = 0$

$F(0,1,1) = \bar{0}.\bar{1}.1 + \bar{0}.1.1 + 0.\bar{1}.1 + 0.1.1 = 1$

$F(1,0,0) = \bar{1}.\bar{0}.0 + \bar{1}.0.0 + 1.\bar{0}.0 + 1.0.0 = 0$

$F(1,0,1) = \bar{1}.\bar{0}.1 + \bar{1}.0.1 + 1.\bar{0}.1 + 1.0.1 = 1$

$F(1,1,0) = \bar{1}.\bar{1}.0 + \bar{1}.1.0 + 1.\bar{1}.0 + 1.1.0 = 0$

$F(1,1,1) = \bar{1}.\bar{1}.1 + \bar{1}.1.1 + 1.\bar{1}.1 + 1.1.1 = 1$

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)

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

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

A	$\overline{A}$
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. Opérateurs logiques de base (Suite)

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. Opérateurs logiques de base (Suite)

- Précédence des opérateurs (priorité des opérateurs) :

- Pour évaluer une expression logique (fonction logique) :
  - on commence par évaluer les sous expressions entre les parenthèses.
  - puis le complément (NON) ,
  - ensuite le produit logique (ET)
  - enfin la somme logique (OU)
- Exemple : $F(A, B, C) = \overline{(A \,.\, B)} \,.\, (C + B) + A \,.\, \overline{B} \,.\, C$

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

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

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

$F(0,1,1) = 1 \,.\, 1 + 0$

$F(0,1,1) = 1 + 0$

$F(0,1,1) = 1$

5. Autres opérateurs logiques

5.1. NAND (Not-AND (NON ET))

- $F(A, B) = \overline{A \cdot B} = A \uparrow B$

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

5.2. NOR (Not-OR (NON OU))

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

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

5. Autres opérateurs logiques (Suite)

- Remarque : NAND et NOR sont des opérateurs universels.

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

5. Autres opérateurs logiques (Suite)

5.3. OU exclusif (XOR)

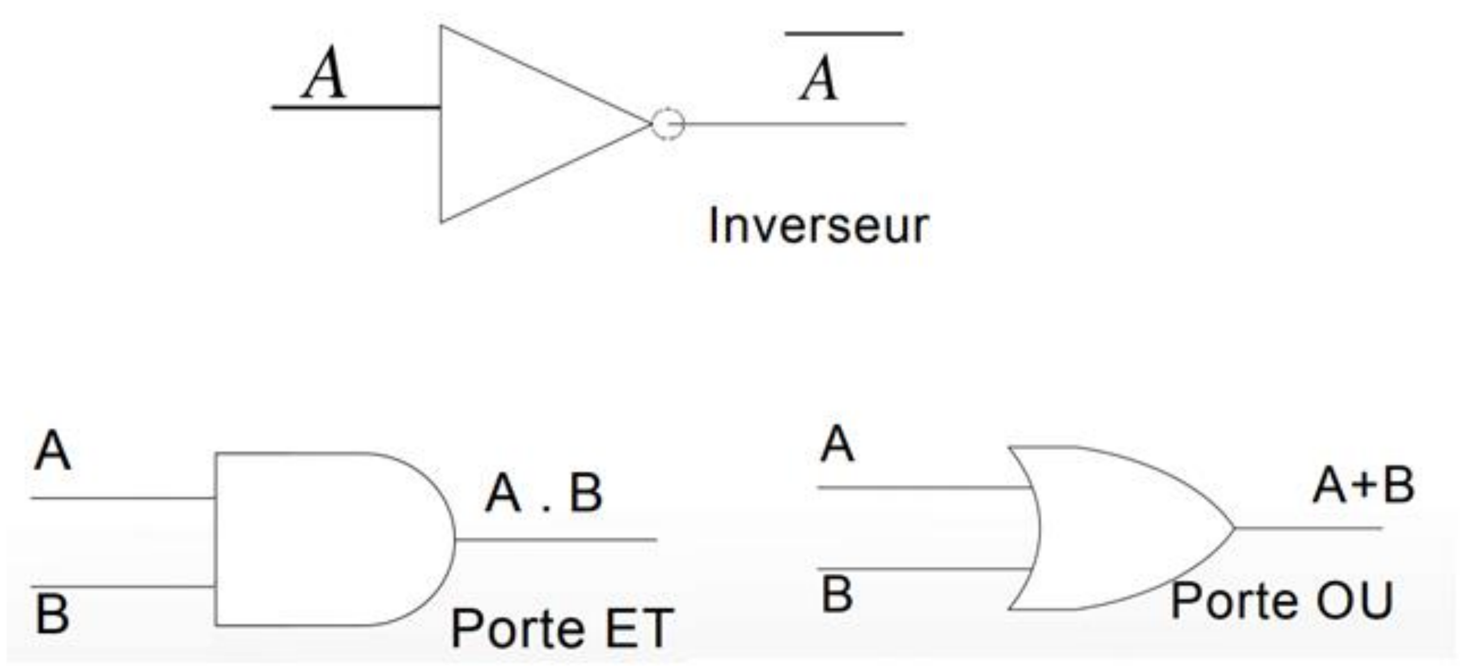
- $F(A, B) = A \oplus B$
- $A \oplus B = \overline{A} . B + A . \overline{B}$

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

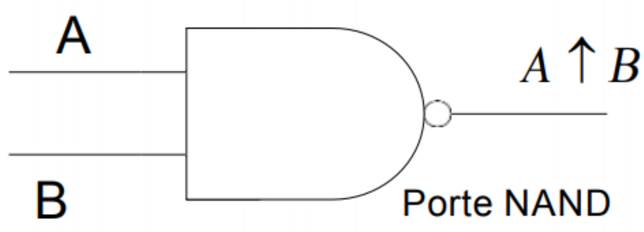
- F est vrai si A OU B est vrai mais pas les deux.
- XOR (Ou-Exclusif) vaut 1 si A est différent de B.

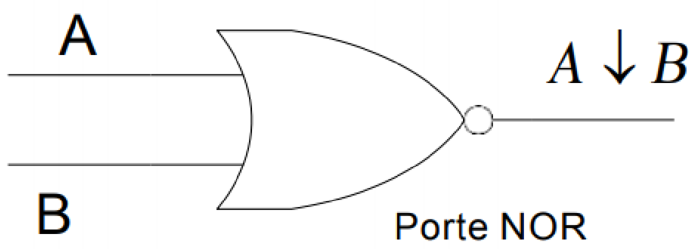
6. Portes logiques

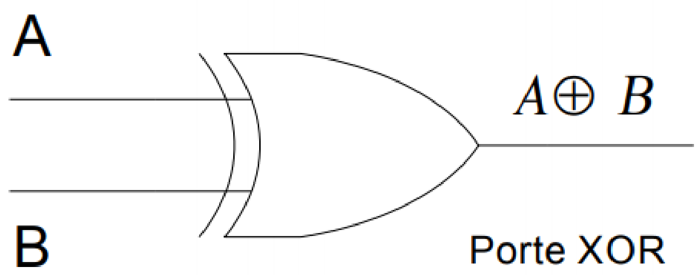
- Une porte logique est un circuit électronique élémentaire qui permet de réaliser la fonction d'un opérateur logique de base.



6. Portes logiques (Suite)







- Remarque :

➢ Les portes ET , OU , NAND , NOR peuvent avoir plus que deux entrées.

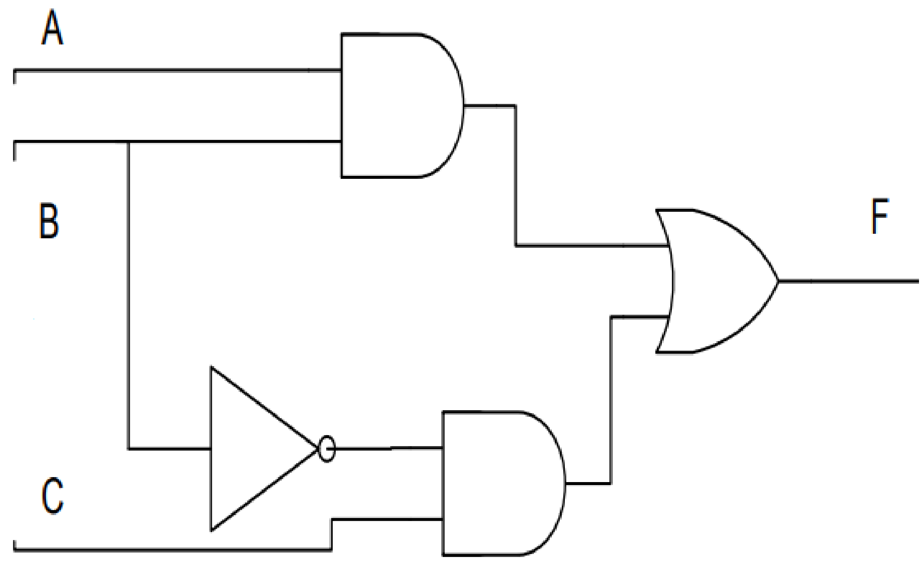
➢ Il n'existe pas de OU exclusif à plus de deux entrées.

6. Portes logiques (Suite)

- 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.
- Exemple 1 :

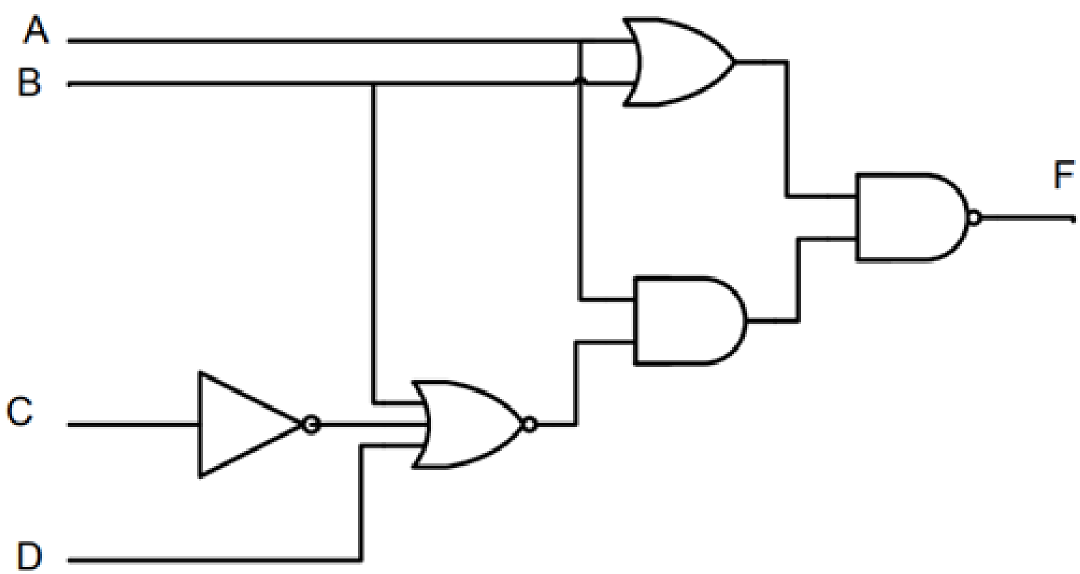
$F(A, B, C) = A \cdot B + \overline{B} \cdot C$



6. Portes logiques (Suite)

- Schéma d'un circuit logique (Suite)
- Exemple 2 :

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



7. Extraction d'une fonction logique à partir d'une table de vérité

- Forme canonique d'une fonction logique :

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

- Exemple :

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

- Remarque : Il existe plusieurs formes canoniques : les plus utilisées sont la première et la deuxième forme.

7. Extraction d'une fonction logique à partir d'une table de vérité (Suite)

- Soit une table de vérité avec N variables et une sortie F, on défini les notions suivantes :
- **Min-terme** : est le produit logique des N variables sous leurs forme vraie ou complémentée.
- **Max-terme** : est la somme logique des N variables sous leurs forme vraie ou complémentée.

- Ecriture des min-termes et des max-termes:
- Si F = 1 on défini un min-terme avec les variables à 0 sont complémentées et les variables à 1 restent non complémentées.
- Si F = 0 on défini un max-terme avec les variables à 1 sont complémentées et les variables à 0 restent non complémentées.

7. Extraction d'une fonction logique à partir d'une table de vérité (Suite)

- Exemple :

- Table de vérité d'une fonction F:

A	B	C		F		
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

7. Extraction d'une fonction logique à partir d'une table de vérité (Suite)

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

F = somme des mintermes

- Exemple :

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

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

F = produit des maxtermes

- 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.

7. Extraction d'une fonction logique à partir d'une table de vérité (Suite)

- Remarque :

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

7. Extraction d'une fonction logique à partir d'une table de vérité (Suite)

- Exemple : Ecrire les fonctions suivantes dans des formes canoniques :

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

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

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

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

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

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

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

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

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

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

8. 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.
  - Tableau de karnaugh.
  - Méthode de quine-mc cluskey.
  - …

8. Simplification des fonctions logiques (Suite)

8.1. Méthode algébrique

- Le principe consiste à appliquer les règles de l'algèbre de Boole afin d'éliminer des variables ou des termes.

- 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

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

8. Simplification des fonctions logiques (Suite)

8.1. Méthode algébrique (Suite)

- Lois fondamentales de l'algèbre de Boole (Suite) :

Elément neutre

$A + 0 = A$

$A \cdot 1 = A$

Elément absorbant

$A + 1 = 1$

$A \cdot 0 = 0$

Inverse

$A + \overline{A} = 1$

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

Simplification

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

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

Théorème de DE-MORGAN

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

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

$$\overline{\sum_i x_i} = \prod_i \overline{x_i}$$

$$\overline{\prod_i x_i} = \sum_i \overline{x_i}$$

8. Simplification des fonctions logiques (Suite)

8.1. Méthode algébrique (Suite) :

- Exemple 1 :

Démontrer la proposition suivante :

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

Correction :

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

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

$$= 1.B.C + 1.A.C + 1.A.B$$

$$= B.C + A.C + A.B$$

8. Simplification des fonctions logiques (Suite)

8.1. Méthode algébrique (Suite) :

- Exemple 2 :

Donner la forme simplifiée de la fonction suivante :

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

Correction :

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

- Remarque : En examinant la méthode de simplification algébrique on remarque que cette dernière devient très difficile si le nombre des variables est grand.

- La méthode de Karnaugh est une technique de simplification rapide.

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh

- Les termes adjacents :

- Examinons l'expression suivante :

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

- 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 :

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

- Ces termes sont dites adjacents.

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Exemple de termes adjacents :

- Ces termes sont adjacents :

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

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

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

- Ces termes ne sont pas adjacents :

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

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

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

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Description de la table de Karnaugh :

- La méthode de Karnaugh 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).

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Description de la table de Karnaugh (Suite) :

- Exemple 1 :

a b c	f(a,b,c)
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1

a \ bc	00	01	11	10
0	0	1	0	0
1	1	0	1	0

f(a,b,c)

1 seul changement entre 2 codes successifs

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Description de la table de Karnaugh (Suite) :

- Autres exemples :

A \ B	0	1
0		
1		

Tableau à 2 variables

C \ AB	00	01	11	10
0				
1				

Tableau à 3 variables

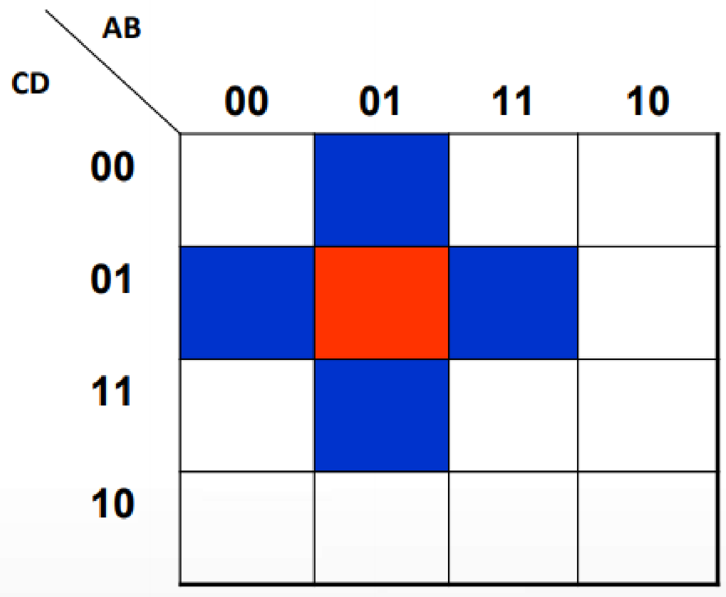
CD \ AB	00	01	11	10
00				
01				
11				
10				

Tableau à 4 variables

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Description de la table de Karnaugh (Suite) :
- Remarque :
- Dans un tableau de karnaugh , chaque case possède un certain nombre de cases adjacentes.
- Dans l'exemple suivant, les cases bleues sont des cases adjacentes à la case rouge :



8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- 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 min termes ou les max termes.

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Passage de la table de vérité à la table de Karnaugh (Suite) :
- 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**

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

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- 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.

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Passage de la forme canonique à la table de Karnaugh (Suite) :

- Exemple :

La fonction logique suivante est donnée sous la première forme canonique:

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

La table de Karnaugh de la fonction F est :

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

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Méthode de simplification :

- 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).
- On s'arrête lorsqu'il y a plus de 1 en dehors des regroupements.
- On élimine les variables qui changent d'états.
- La fonction finale est égale à la somme des termes dont l'état des variables ne change pas à l'intérieur d'un regroupement.
- Une ou plusieurs cases peuvent être communes à plusieurs regroupements.

8. Simplification des fonctions logiques (Suite)

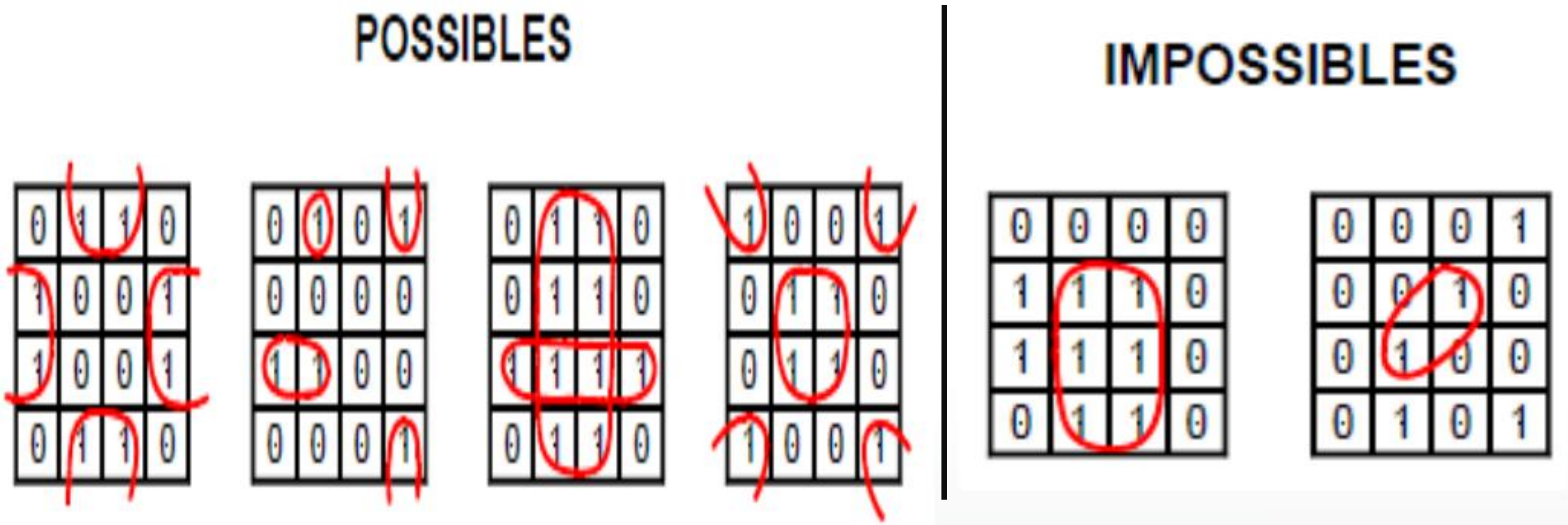
8.2. Simplification par la table de Karnaugh (Suite)

- Méthode de simplification (Suite) :
- Règles de regroupement :
  - Groupes de 2^n cases : 1, 2, 4, 8, …
  - En ligne, colonne, rectangle, carré, mais pas diagonale.
  - Tous les 1, mais pas les 0 au moins une fois dans les groupements.
- Règles de minimisation de la fonction :
  - Rechercher les groupements en commençant par les cases qui n'ont qu'une seule façon de se grouper.
  - Rechercher les groupements les plus grands.
  - Les groupements doivent contenir au moins un 1 non utilisé par les autres groupements.
  - L'expression logique finale est la réunion (la somme) des groupements après simplification et élimination des variables qui changent d'état.

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Méthode de simplification (Suite) :
- 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.



8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Méthode de simplification (Suite) :
- Exemple 1 : 3 variables

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

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

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Méthode de simplification (Suite) :

- Exemple 2 : 4 variables

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

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

8. Simplification des fonctions logiques (Suite)

8.2. Simplification par la table de Karnaugh (Suite)

- Méthode de simplification (Suite) :

- Exemple 3 : 4 variables

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

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

9. Synthèse des circuits logiques

- 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. Bien comprendre le fonctionnement du système à réaliser :
   a. Définir les variables d'entrée.
   b. Définir les variables de sortie.
2. Établir la table de vérité : formuler le fonctionnement du circuit dans une table de vérité qui donne les valeurs de sortie en fonction des valeurs des variables d'entrée.
3. Extraction (écriture) de la fonction logique à partir de la Table de vérité sous la première ou la deuxième forme canonique.
4. Simplifier la fonction trouvée (algébriquement ou par tableau de Karnaugh).
5. Dessiner le schéma logique (logigramme) du circuit correspondant à la fonction simplifiée.

9. Synthèse des circuits logiques (Suite)

- **Exemple : Une serrure de sécurité**

- Description du fonctionnement du 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 le schéma du circuit qui permet de contrôler l'ouverture de la serrure.

9. Synthèse des circuits logiques (Suite)

- **Exemple : Une serrure de sécurité (Suite)**

- Correction :

1. Définir les variables d'entrée et les variables de sortie :

Le système possède trois entrées : chaque entrée représente une clé.

On va correspondre à chaque clé une variable logique :

la 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 fermée).

On va correspondre une variable S pour designer la sortie :

S = 1 si la serrure est ouverte ,

S = 0 si elle est fermée.

9. Synthèse des circuits logiques (Suite)

Exemple : Une serrure de sécurité (Suite)

- Correction (Suite) :

2. Établir la table de vérité :

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

3. Extraction de la fonction logique : il suffi de choisir une seule forme canonique pour représenter le circuit.

La fonction sous la première forme canonique :

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

9. Synthèse des circuits logiques (Suite)

- **Exemple : Une serrure de sécurité (Suite)**

- Correction (Suite) :

3. Simplification de la fonction (algébriquement) :

$$S = \bar{A}.B.C + A.\bar{B}.C + A.B.\bar{C} + A.B.C$$

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

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

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

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

$$S = A.C + B.C + A.B.\bar{C}$$

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

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

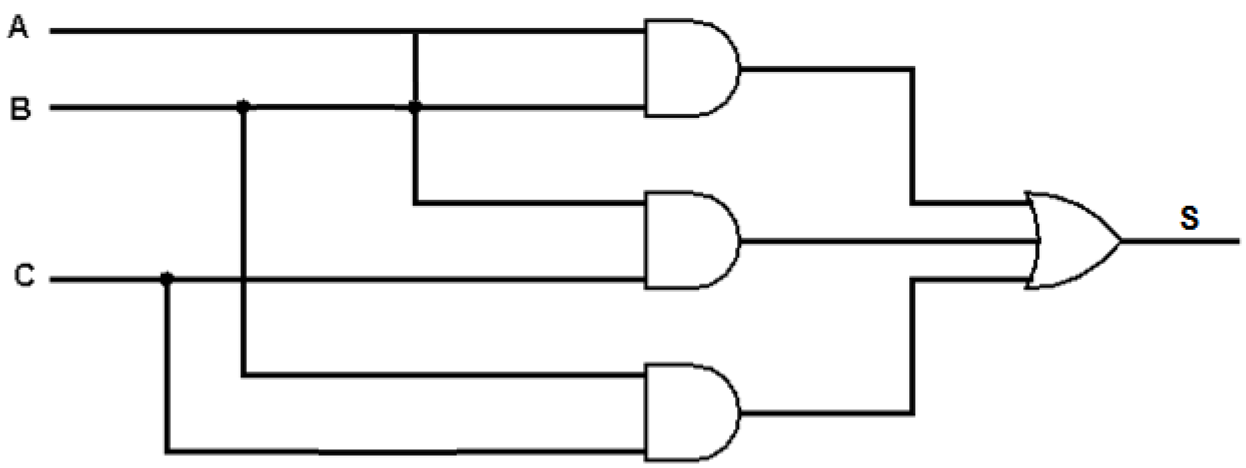
$$S = A.B + A.C + B.C$$

9. Synthèse des circuits logiques (Suite)

- **Exemple : Une serrure de sécurité (Suite)**

- Correction (Suite) :

4. Schéma du cicruit :



9. Synthèse des circuits logiques (Suite)

- **Exercice 1 : Commande de Lampes**

Trois interrupteurs a, b, c commandent l'allumage de 2 lampes R et S suivant les conditions suivantes :

- Dès qu'un ou plusieurs interrupteurs sont activés, la lampe R doit s'allumer.
- La lampe S ne doit s'allumer que si au moins 2 interrupteurs sont activés.

Questions :

a) Calculer la table de vérité associée à ce problème.

b) Exprimer les fonctions de R et S en fonction des trois interrupteurs.

c) Simplifier les fonctions de R et S.

d) Réaliser le logigramme des fonctions de R et S.

9. Synthèse des circuits logiques (Suite)

- **Exercice 1 : Commande de Lampes (Suite)**
- Correction :

a) Table de vérité :

- Définition des entrées- sorties :

On a trois entrées : on va correspondre à chaque interrupteur une variable logique :

le premier interrupteur : A , s'il est activé alors A = 1 , sinon A = 0

le deuxième interrupteur : B , s'il est activé alors B = 1 , sinon B = 0

le troisième interrupteur : C , s'il est activé alors C = 1 , sinon C = 0

Et on a deux sorties R et S :

S = 1 si la lampe S est allumée , S = 0 sinon.

R = 1 si la lampe R est allumée , R = 0 sinon.

9. Synthèse des circuits logiques (Suite)

Exercice 1 : Commande de Lampes (Suite)

- Correction (Suite) :

a) Table de vérité (Suite) :

A	B	C		R	S
0	0	0		0	0
0	0	1		1	0
0	1	0		1	0
0	1	1		1	1
1	0	0		1	0
1	0	1		1	1
1	1	0		1	1
1	1	1		1	1

b) Expressions de R et S :

$$R = \bar{A}.\bar{B}.C + \bar{A}.B.\bar{C} + \bar{A}.B.C + A.\bar{B}.\bar{C} + A.\bar{B}.C + A.B.\bar{C} + A.B.C$$

$$S = \bar{A}.B.C + A.\bar{B}.C + A.B.\bar{C} + A.B.C$$

9. Synthèse des circuits logiques (Suite)

Exercice 1 : Commande de Lampes (Suite)

- Correction (Suite) :

c) Simplification :

$$R = \bar{A}.\bar{B}.C + \bar{A}.B.\bar{C} + \bar{A}.B.C + A.\bar{B}.\bar{C} + A.\bar{B}.C + A.B.\bar{C} + A.B.C$$

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

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

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

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

$$R = A + B + C$$

9. Synthèse des circuits logiques (Suite)

Exercice 1 : Commande de Lampes (Suite)

- Correction (Suite) :

c) Simplification (Suite) :

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

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

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

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

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

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

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

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

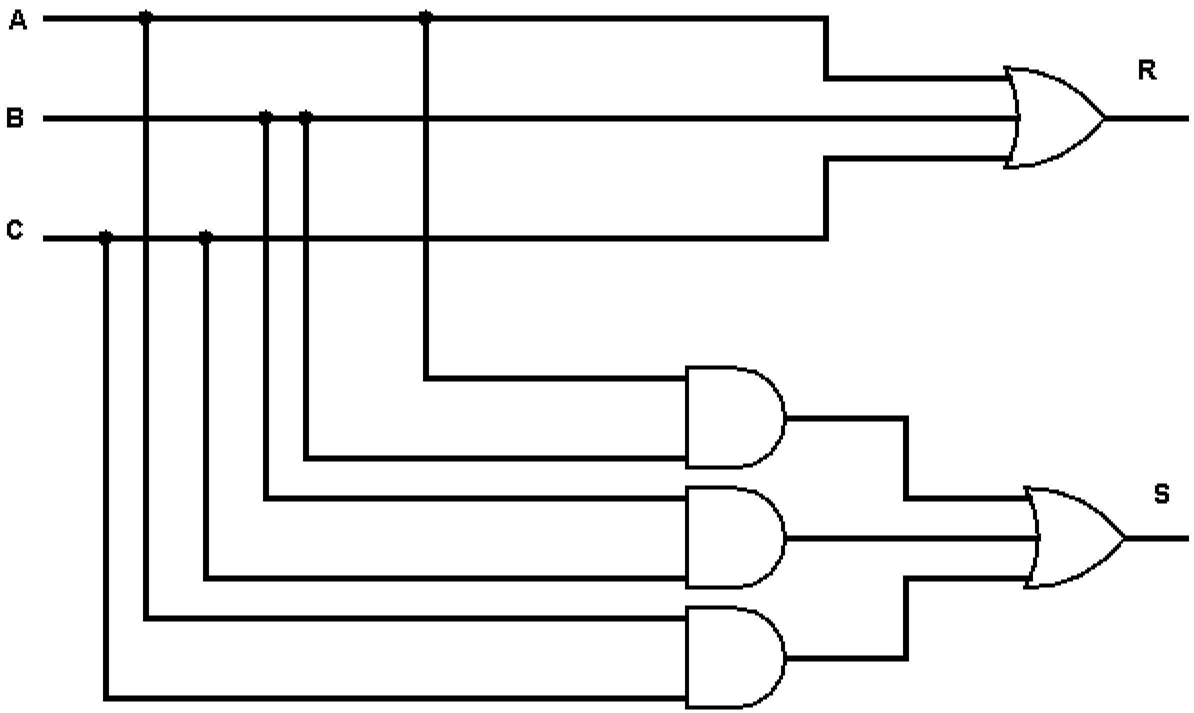
$$S = A.B + A.C + B.C$$

9. Synthèse des circuits logiques (Suite)

Exercice 1 : Commande de Lampes (Suite)

- Correction (Suite) :

d) Logigramme :



9. Synthèse des circuits logiques (Suite)

Exercice 2 : Fonctionnement d'un pont

Un pont peut soutenir 7 tonnes au maximum, on doit alors surveiller le poids des véhicules se présentant aux deux extrémités A et B où deux bascules mesurent le poids respectifs a et b des véhicules.

On suppose que chaque véhicule a un poids inférieur à 7 tonnes.

On suppose, aussi, que à chaque fois se présentent deux véhicules au même temps, chacun dans une extrémité.

Le fonctionnement est alors le suivant :

Si a+b <= 7 tonnes, les barrières A et B s'ouvrent ;

Si a+b > 7 tonnes, la barrière correspondant au véhicule le plus léger s'ouvre,

Si a=b la barrière A s'ouvre en priorité.

a et b n'étant pas des variables binaires, il convient de créer 2 variables binaires X et Y, et de reformuler l'énoncé du problème.

Chercher alors les équations de A et B, en fonction de X et Y, et en donner le schéma du circuit logique correspondant.

9. Synthèse des circuits logiques (Suite)

Exercice 2 : Pont (Suite)

- Correction :

Définition des entrées- sorties :

On a : deux entrées X, Y et deux sorties A et B :

X = 0 si a+b <= 7 tonnes

X = 1 si a+b >= 7 tonnes

Y = 0 si a>b

Y= 1 si a<=b

Table de vérité :

X	Y		A	B
0	0		1	1
0	1		1	1
1	0		0	1
1	1		1	0

9. Synthèse des circuits logiques (Suite)

Exercice 2 : Pont (Suite)

- Correction (Suite) :

Equations et simplifications :

$$A = \overline{X}.\overline{Y} + \overline{X}.Y + X.Y = \overline{X} + Y$$

$$B = \overline{X}.\overline{Y} + \overline{X}.Y + X.\overline{Y} = \overline{X} + \overline{Y}$$

Logigramme :