# Chapitre 4: L'algèbre de Boole

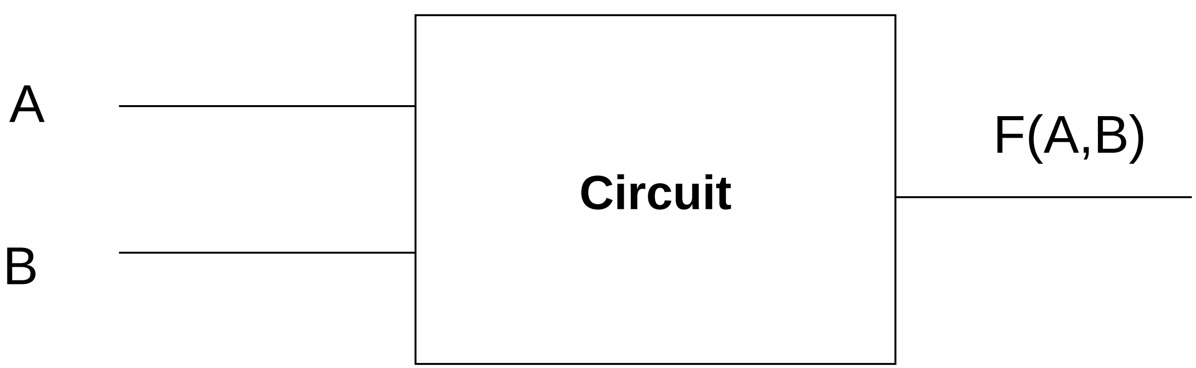
Cours "Représentations des données Machines"

# 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 ,…..).

- 



# Introduction

- 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

1854 : Georges Boole propose une algèbre

Propositions vraies ou fausses
et opérateurs possibles $\longrightarrow$ Algèbre de Boole

$\downarrow$

Étude des systèmes binaires :
Possédant deux états s’excluant mutuellement

C’est le cas des systèmes numériques
(des sous ensembles : les circuits logiques)

# Algèbre binaire

On se limite : Base de l'algèbre de Boole
Propriétés indispensables aux systèmes logiques

<u>Définitions</u> :

- États logiques : 0 et 1, Vrai et Faux,
  (purement symbolique)

- Variable logique : Symbole pouvant prendre
  comme valeur des états logiques (A,b,c, Out ...)

- Fonction logique : Expression de variables et d'opérateurs

# Calcul propositionnel

Algèbre de Boole sur [0,1] = algèbre binaire
Structure d'algèbre de Boole

- 2 lois de composition interne (LCI)
- 1 application unaire

<u>2 LCI : ET, OU</u>

- Somme (OU, Réunion, Disjonction)

$$s = a + b = a \vee b$$

- Produit (ET, intersection, Conjonction)

$$s = a \, . \, b = ab = a \wedge b$$

<u>Application unaire :</u>

- Not (complémentation, inversion, négation, non) $s = \overline{a} = \text{not}(a) = \neg\, a$

# Fonctions logiques

Fonction logique à n variables f(a,b,c,d,...,n)

$$[0,1]^n \longrightarrow [0,1]$$

- Une fonction logique ne peut prendre que deux valeurs
- Les cas possibles forment un ensemble fini ( $2^n$)

- Descriptions, preuves possibles par énumération

  comparer f(a,b,c,..n) et g(a,b,c,..,n)

  = comparer les tables représentant f et g

La table de fonction logique = table de vérité

# 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 |

# 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 |

# 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 | $\overline{A}$ |
|---|---|
| **0** | **1** |
| **1** | **0** |

# Tables de vérité de ET, OU, NON

| a \ b | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |

$s = a + b$

| a \ b | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

$s = a \cdot b$

| a | |
|---|---|
| 0 | 1 |
| 1 | 0 |

$s = \overline{a}$

S est vrai si a OU b est vrai.

S est vrai si a ET b sont vrais.

S est vrai si a est faux

# Tables de vérité de ET, OU, NON

| a | b | s |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

| a | b | s |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| a | s |
|---|---|
| 0 | 1 |
| 1 | 0 |

# Deux autres opérateurs : NAND,NOR

| a \ b | 0 | 1 |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 1 | 0 |

$$s = a \uparrow b = \overline{a.b}$$

S est vrai si a OU b est faux.

NAND (Not-AND)

| a \ b | 0 | 1 |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 0 | 0 |

$$s = a \downarrow b = \overline{a+b}$$

S est vrai si ni a, ni b ne sont vrais.

NOR (Not-OR

NAND et NOR ne sont pas associatifs

# Encore un opérateur : XOR

| a \ b | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

$$s = a \oplus b = a.\overline{b} + \overline{a}.b$$

S est vrai si a OU b est vrai mais pas les deux.

XOR (Ou-Exclusif) vaut 1 si a est différent de b

Opérateur de différence (disjonction)

# Encore un opérateur : XOR

XOR est associatif $s = a \oplus b \oplus c \quad ..... \oplus n$

vaut 1 si le nombre de variables à 1 est impair.

$$s = \overline{a \oplus b} = \overline{a} \oplus b = a \oplus \overline{b} = a \text{ XNOR } b$$

XNOR = $\overline{\text{XOR}}$ vaut 1 si $a = b$

Inverseur programmable : (le programme vaut 0 ou 1)

$a \oplus 1 = \overline{a} \qquad a \oplus 0 = a$

# Simplification des fonctions logiques

# 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 : tableaux de karnaugh )

# Propriétés de ET, OU, NON

- 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

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

# Propriétés de ET,OU,NON

- Elément neutre

  $a+0 = a$

  $a.1 = a$

- Elément absorbant

  $a+1 = 1$

  $a.0 = 0$

- Inverse

  $a+\overline{a} = 1$

  $a.\overline{a} = 0$

- Théorème de "De Morgan"

  $\overline{a+b} = \overline{a} . \overline{b}$

  $\overline{a.b} = \overline{a} + \overline{b}$

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

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

# Exercice 1:

Démontrer la proposition suivante :

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

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

Donner la forme simplifiée de la fonction suivante :

$$F(A, B, C, D) = \overline{A}BCD + A\overline{B}CD + AB\overline{C}D + ABC\overline{D} + ABCD$$

# Correction

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

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

$$BC + AC + AB$$

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

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

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

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

$$= AB + ACD$$

# Simplification par la table de Karnaugh

# Description de la table de karnaugh

•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).

•Un tableau de Karnaugh = table de vérité de $2^n$ cases avec un changement unique entre 2 cases voisines d'où des codes cycliques (Gray ou binaire réfléchi).

•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 ).

# Description de la table de karnaugh

## Règles de regroupement :

- groupe de $2^n$ cases : 1,2,4 ou 8
- en ligne, colonne, rectangle, carré, <u>mais pas diagonale</u>
- 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.

# Description de la table de karnaugh

| B \ A | 0 | 1 |
|---|---|---|
| 0 | | |
| 1 | | |

**Tableau à 2 variables**

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

**Tableaux à 3 variables**

# Tableaux de Karnaugh

f (a,c,d, ..,n) fonction logique à N entrées

sera représentée par une table à $2^N$ lignes

un tableau à $2^N$ cases

| 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)

Code Gray ou binaire réfléchi
=
1 seul changement entre 2 codes successifs

# Tableaux de Karnaugh

## Exemple 1 : 3 variables

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

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

# Exemple 2 : 4 variables

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

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

# Tableaux de Karnaugh

## Exemple 3 : 4 variables

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

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

# Tableaux de Karnaugh

## Exemple 4 : 5 variables

U = 0

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

U= 1

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

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

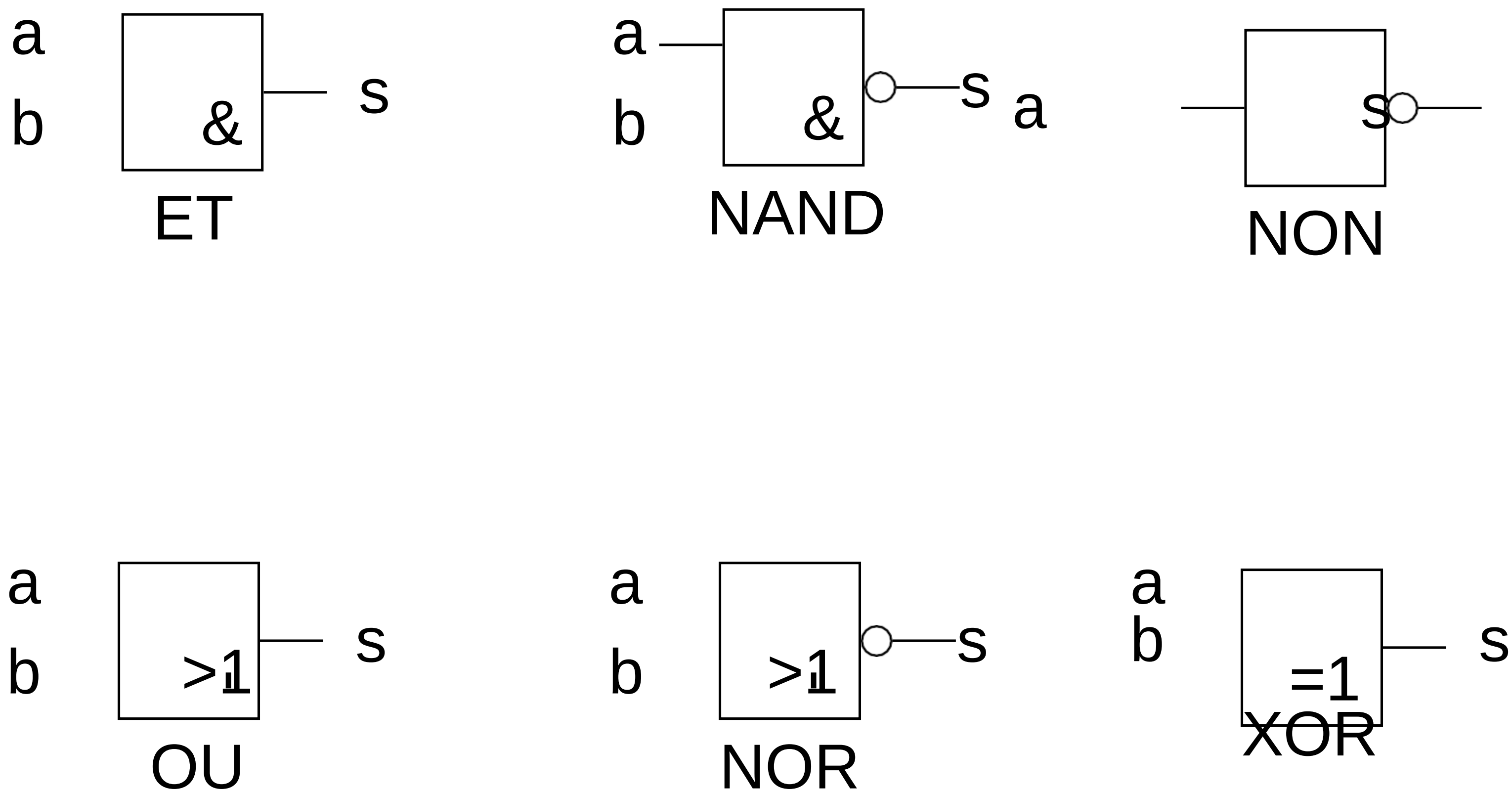
# Exercice

**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 |

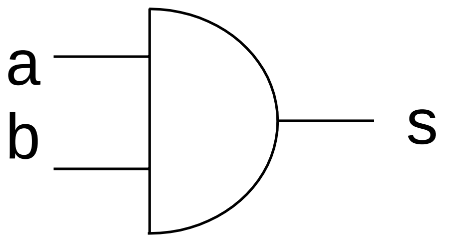
| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | | 1 | 1 |
| 01 | | | | |
| 11 | | | | |
| 10 | 1 | 1 | 1 | 1 |

# Représentation graphique :
# Norme française

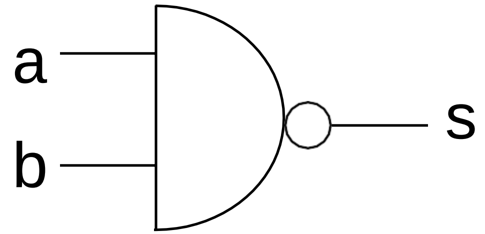


Représentation graphique :

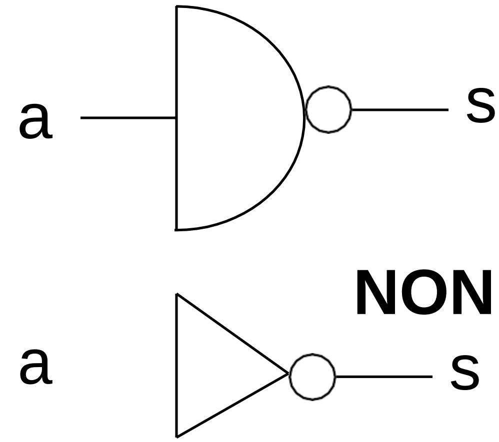
# Norme américaine



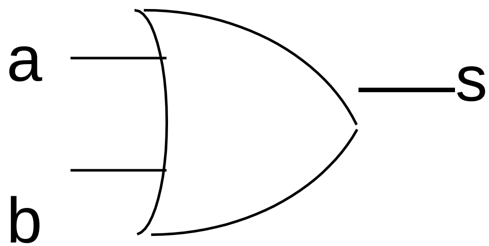

**ET**



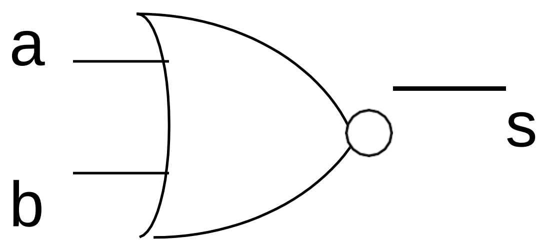

**NAND**



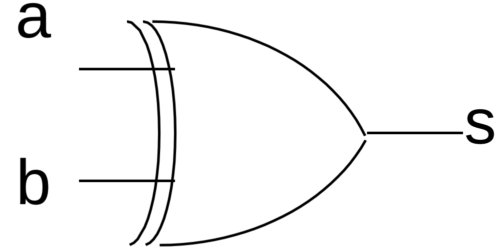

**NON**




**OU**




**NOR**



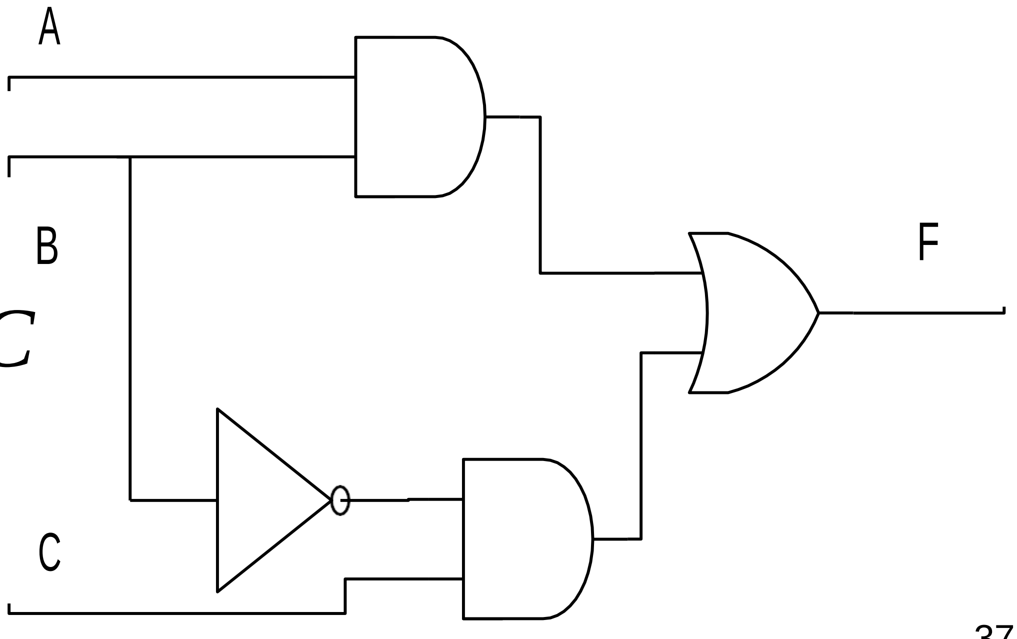

**XOR**

# 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**

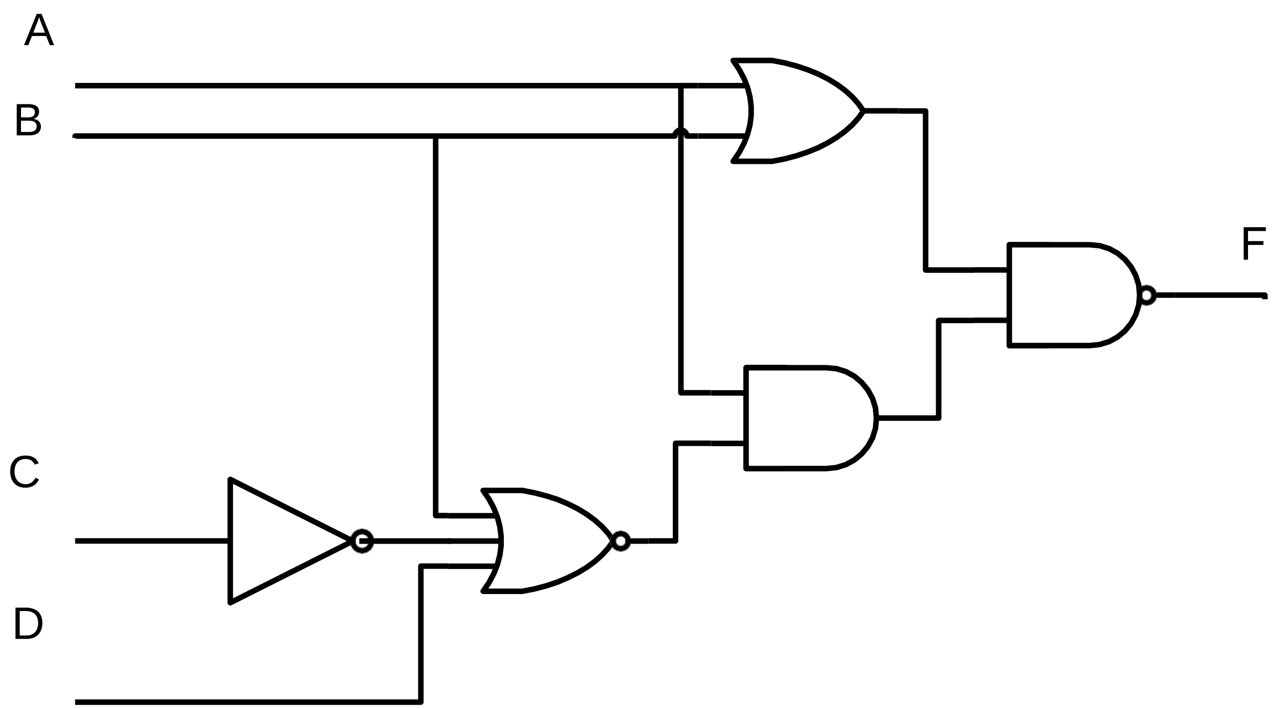
$$F(A,B,C) = A.B + \overline{B}.C$$



# Circuits logiques

**Exemple 2:**

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



# Circuits logiques

## Exercice 1

- Donner le logigramme des fonctions suivantes :

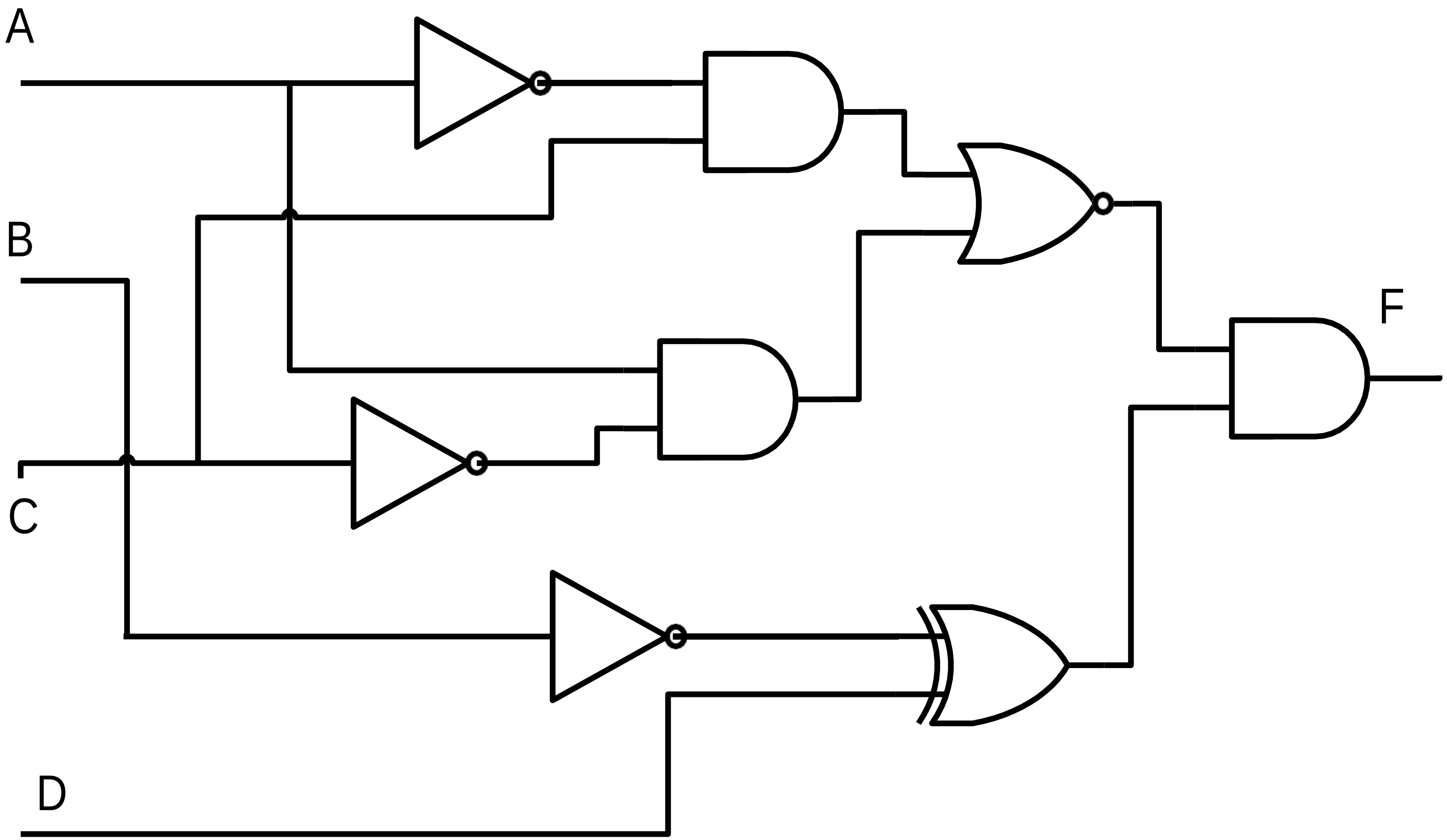
$$F(A,B) = \overline{A}.B + A.\overline{B}$$

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

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

# Circuits logiques

**Exercice 2 :** Donner l'équation de F ?



# Circuits logiques

**Exercice 3 :** Soit la fonction **F**

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

1) Simplifier la fonction F par la méthode des diagrammes de Karnaugh

2) Donner les schémas logiques ou logigrammes de la fonction simplifiée utilisant :
   - Logigramme 1 : avec uniquement des portes NON ET
   - Logigramme 2 : des portes ET, OU, et des inverseurs,

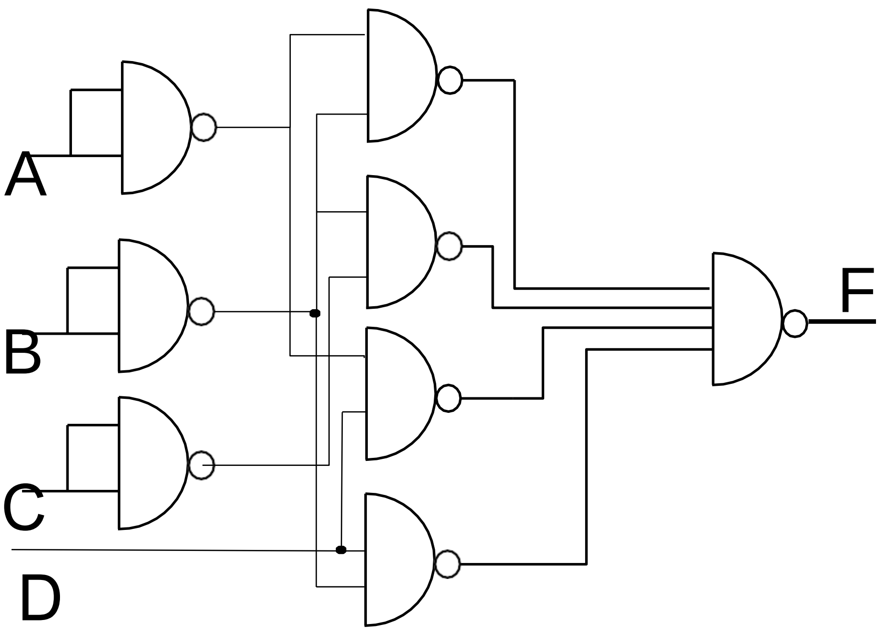
# Circuits logiques

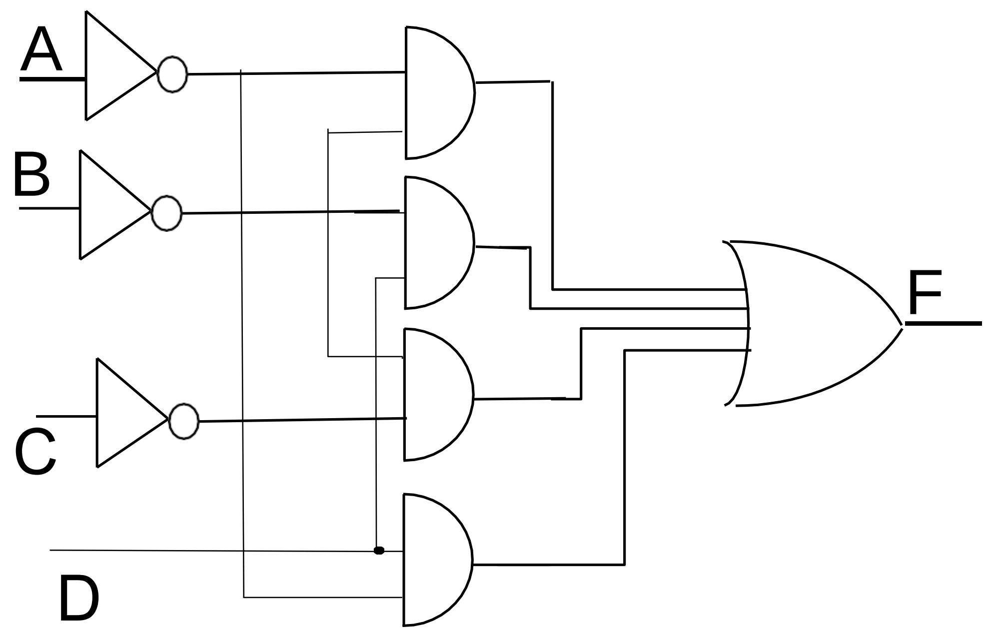
**Exercice 3 :** Soit la fonction **F correction**

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

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

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



Merci pour votre attention