CHAPITRE 4 : ALGÈBRE DE BOOLE BINAIRE

Mme KAZI TANI ADILA

STRUCTURE MACHINE 1

Introduction

- Tout ordinateur est constitués d'un ensemble de **circuits** électroniques qui ont tous une fonction spécialisée (UAL unité arithmétique et logique, mémoire, circuit décodant les instructions etc).
- Ces circuits sont fait à partir de **circuits logiques** dont le but est d'exécuter des <u>opérations</u> sur des <u>variables logiques</u> pour réaliser une <u>fonction logique</u> (addition, comparaison,….).

Introduction

- **Circuit Logique** : est l'interconnexion de plusieurs circuits élémentaires, appelés portes logiques, conçu pour réaliser une fonction désirée.
- Un **circuit logique** peut avoir une ou plusieurs variables d'entrée et une ou plusieurs fonctions de sortie.
- Types de circuits logiques:
  - Combinatoires
  - Séquentiels

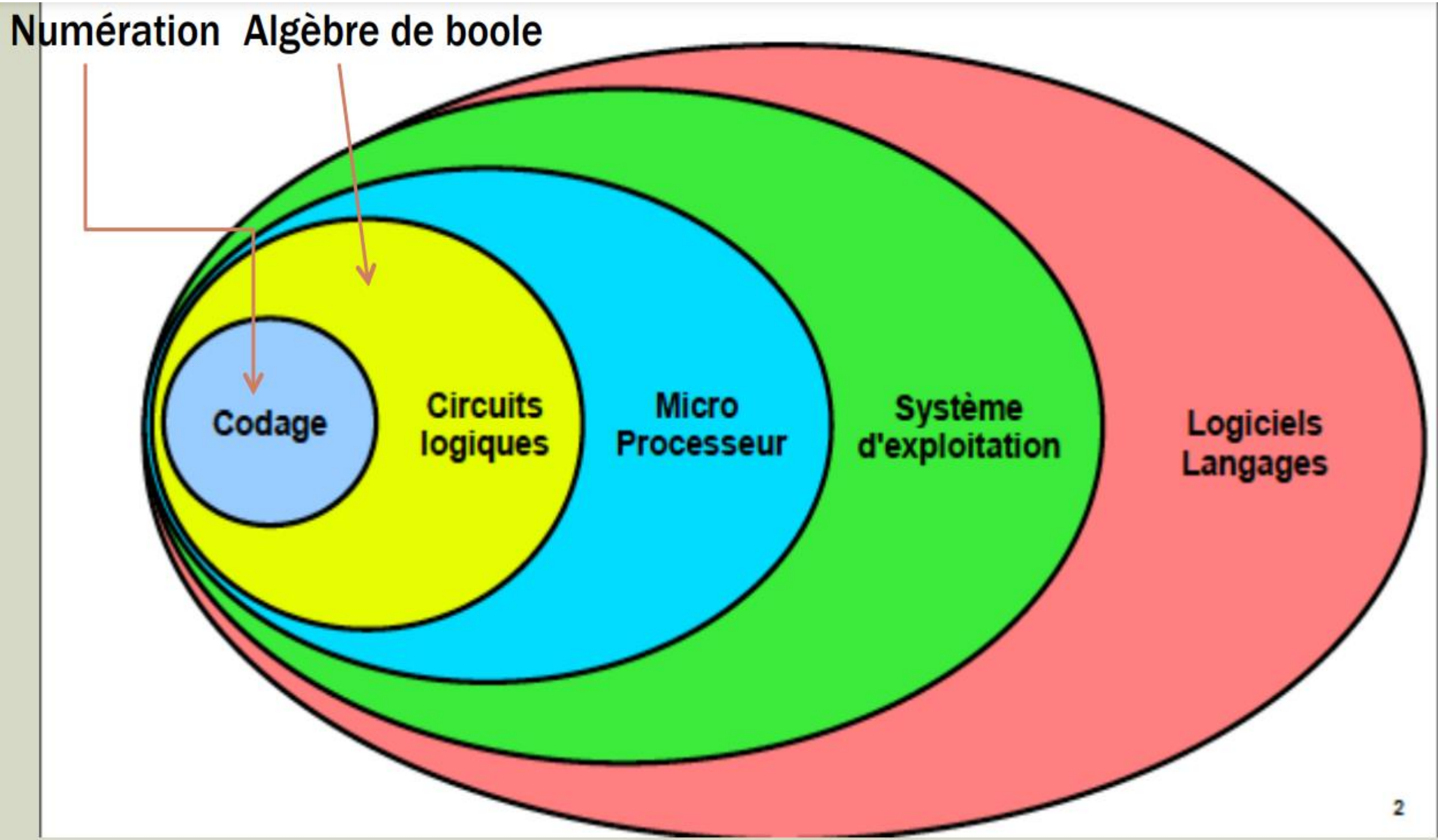
Introduction

- Circuits **combinatoire** et **séquentiel** :
  - Dans un circuit **combinatoire**, la fonction de sortie dépend uniquement des variables d'entrée indépendamment du temps.
  - Dans un circuit **séquentiel**, à l'instant t_i, la fonction de sortie dépend à la fois des variables d'entrée et du temps t_{i-1}.

Introduction

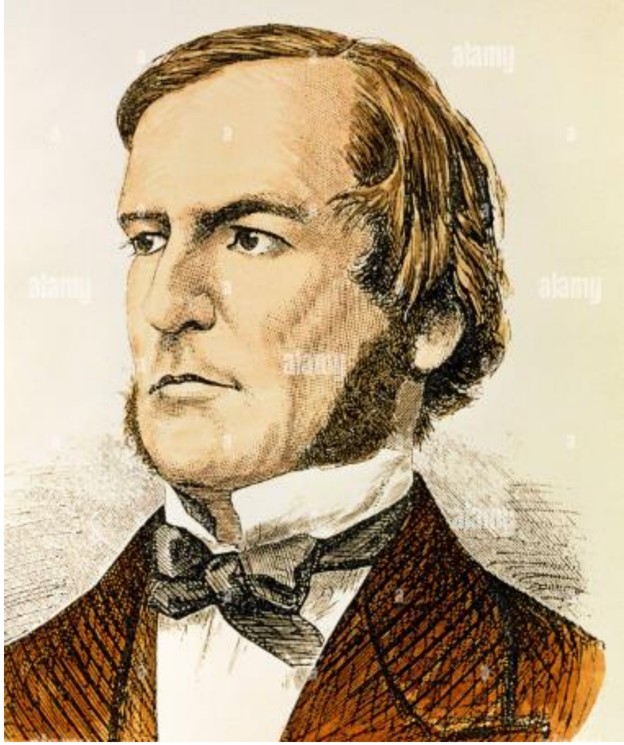
- Dans ce chapitre on s'intéresse aux circuits combinatoires. Pour concevoir et réaliser de tels circuits on doit avoir un modèle mathématique de la fonction réalisée par ce circuit .
- Ce modèle doit prendre en considération le système binaire.
- Le modèle mathématique utilisé est celui de l'Algèbre de Boole.

Introduction



Définitions

- L'algèbre de Boole est une logique découverte en 1847 par le mathématicien britannique **George BOOLE.**



Définitions

- L'algèbre de BOOLE est une structure algébrique qui ne contient que deux éléments, que l'on appelle couramment variables logiques.
- Ces variables ne peuvent avoir que deux états :
  - 0 : Faux (False)
  - 1 : Vrai (True)

ou : ouvert ou fermé, arrêt ou marche, inactif ou actif, relâché ou enfoncé.

Définitions

- Comme toute autre algèbre, celle de Boole manipule des grandeurs représentées par des symboles (les **variables**) selon des **opérations** pour produire des **fonctions.**
- Les variables et les fonctions prennent des valeurs dans l'ensemble [0, 1] (0: faux et 1:vrai)

variable logique ≡ variable binaire ≡ variable Booléenne.

Définitions

- Une variable logique modélise un système qui ne doit avoir que deux états tel un interrupteur par exemple .
- Une **fonction logique** (ou Booléenne) est le résultat d'une combinaison (selon des opérations) de n variables.
- Une **fonction logique** est entièrement définie par la donnée de ses valeurs pour les 2^n combinaisons possibles des n variables . Cette définition se traduit par la **table de vérité** de la fonction.

Définitions

- Une fonction logique modélise la sortie d'un système qui ne peut être que dans deux états.
- Le rapport entre les circuits numériques et l'algèbre de Boole est tel que:
  - Les entrées du circuit sont les variables.
  - La (ou les) sortie(s) du circuit est (sont) la fonction (les fonctions).
  - Le contenu du circuit "calcule" l'expression de la (ou des) fonction(s). L'état binaire (0 et 1) aussi bien des entrées que des sorties est concrétisé par deux et seulement deux niveaux de tensions électriques.

Propriétés

- Comme notre objectif est la réalisation des circuits logiques alors il est évident que plus simples seront ces circuits, plus simple en sera leur réalisation, cette simplicité se traduisant, bien entendu, par un meilleur prix de revient.
- L'obtention d'une expression minimale ou la simplification peut se faire par une méthode purement algébrique en utilisant les différentes propriétés de l'algèbre de BOOLE.

Propriétés

Involution	$\bar{\bar{a}} = a$
Idempotence	$a + a = a$ $a.a = a$
Complémentarité	$a.\bar{a} = 0$ $a + \bar{a} = 1$
Identités remarquables (Elément neutre et élément absorbant)	$a.1 = 1.a = a$ $a + 0 = 0 + a = a$ $a + 1 = 1$ $a.0 = 0$

Propriétés

Distributivité	Du ET sur le OU : $a.(b+c) = a.b + a.c$ Du OU sur le ET : $a + (b.c) = (a+b).(a+c)$
Associativité	$(a.b).c = a.(b.c)$ $(a+b)+c = a+(b+c)$

Propriétés

Absorption	$a + a.b = a$ $a.(a + b) = a$ $a + \bar{a}.b = a + b$ $a.(\bar{a} + b) = a.b$
Règles de Morgan	$\overline{a + b} = \bar{a}.\bar{b}$ $\overline{a.b} = \bar{a} + \bar{b}$
Redondance	$a.b + b.c + c.\bar{a} = a.b + c.\bar{a}$ $(a + b).(b + c).(c + \bar{a}) = (a + b).(c + \bar{a})$

Propriétés

Exemples : Simplifier les expressions suivantes à l'aide des propriétés de l'algèbre de BOOLE :

- $F1 = \overline{A}.B + A.B$
- $F2 = (A + B).(A + \overline{B})$
- $F3 = A + A.B$
- $F4 = \overline{A}.\overline{B} + \overline{A + B + C + D}$

Propriétés

Exemples :

- $F1 = \overline{A}.B + A.B$

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

$$= B.1$$

$$\boxed{\mathbf{F1 = B}}$$

- $F2 = (A + B).(A + \overline{B})$

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

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

$$= A.(1 + \overline{B} + B) = A.1$$

$$\boxed{\mathbf{F2 = A}}$$

Propriétés

Exemples :

- $F3 = A + A.B$

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

$$\boxed{\mathbf{F3 = A}}$$

- $F4 = \overline{A}.\overline{B} + \overline{A + B + C + D}$

$$= \overline{A}.\overline{B} + \overline{A}.\overline{B}.\overline{C}.\overline{D}$$
$$= \overline{A}.\overline{B}.(1 + \overline{C}.\overline{D}) = \overline{A}.\overline{B}.1$$

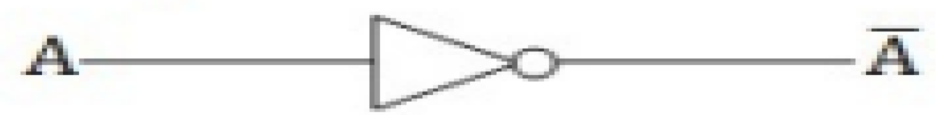
$$\boxed{\mathbf{F4 = \overline{A}.\overline{B}}}$$

Les opérations logiques Principales

- Une opérations de base (également appelées opérateur de base ou fonction de base) est un opérateur mathématique qui permet de lier des variables binaires en vue de décrire avec plus de précision un problème.
- Les opérations de base sont:
  - **AND** logique (produit),
  - **OR** logique (somme)
  - **NOT** logique ou complémentation.

Les opérations logiques Principales

- **NOT (NON)** : Appelé couramment inverseur (négation), a une seule entrée et une seule sortie, c'est un opérateur qui réalise le complément d'une variable logique A, noté : $NOT\,(A) = \bar{A}$



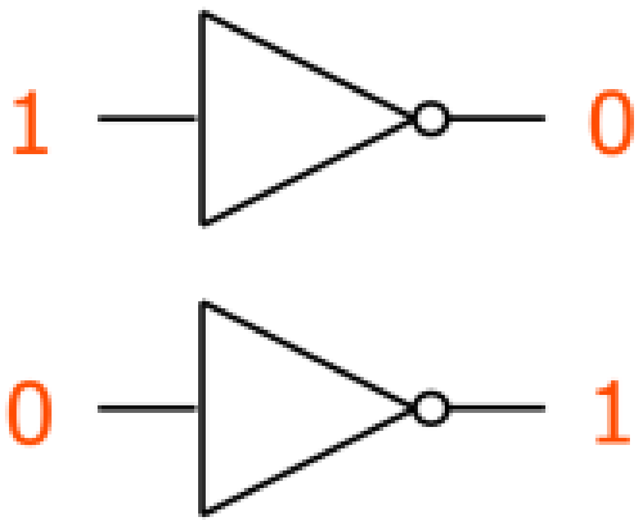
Porte NON (NOT)

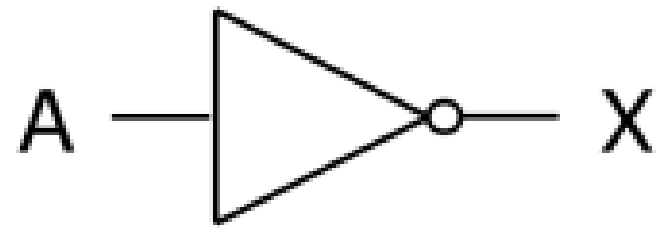
Son fonctionnement est décrit pat la table de vérité suivante :

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

Les opérations logiques Principales

- **NOT (NON) :**





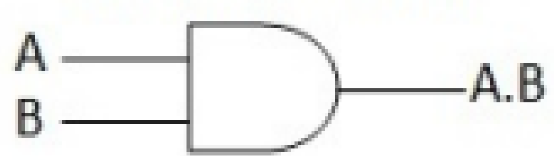
$$X=\overline{A}$$

Les opérations logiques Principales

- **AND (ET)** : C'est le produit logique de deux ou plusieurs variables logiques, le résultat de l'opération est 1, lorsque toutes les variables sont à 1 , sinon 0.
- Si A et B représentent deux variables logiques, le résultat de l'opération ET entre ces deux variables est noté : **A AND B = A . B**

Les opérations logiques Principales

- **AND (ET)** : Une porte logique AND à deux entrées est symbolisée de la manière suivante :



Porte ET (AND)

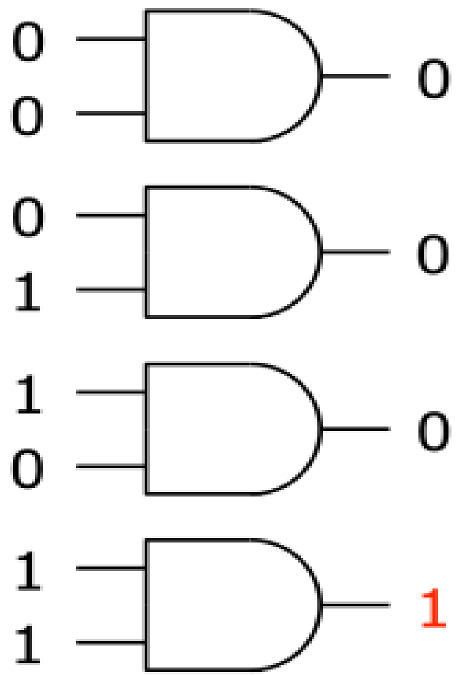
- L'opération logique AND, notée '•' est définie par la table de vérité suivante :

A	B	F = A.B
0	0	0
0	1	0
1	0	0
1	1	1

Les opérations logiques Principales

- **AND (ET) :**

La porte AND représente un "et logique"

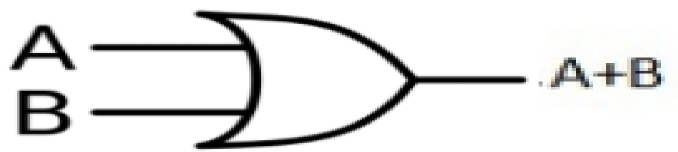


Les opérations logiques Principales

- **OR (OU)** : C'est la somme logique de deux ou plusieurs variables logiques, le résultat de l'opération est 1,lorsque au moins une des variables est égale à 1, et 0 si toutes les variables sont 0.
- Si A et B représentent deux variables logiques, le résultat de l'opération OU entre ces deux variables A et B est noté : **A OR B = A+B**

Les opérations logiques Principales

- **OR (OU)** : Une porte logique OR à deux entrées est symbolisée de la manière suivante :



Porte OU (OR)

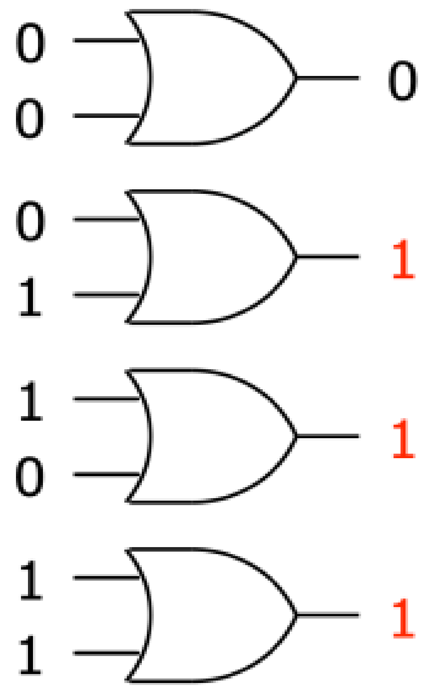
- La fonction OR, notée +, est définie par la table de vérité suivante :

A	B	F= A+B
0	0	0
0	1	1
1	0	1
1	1	1

Les opérations logiques Principales

- **OR (OU) :**

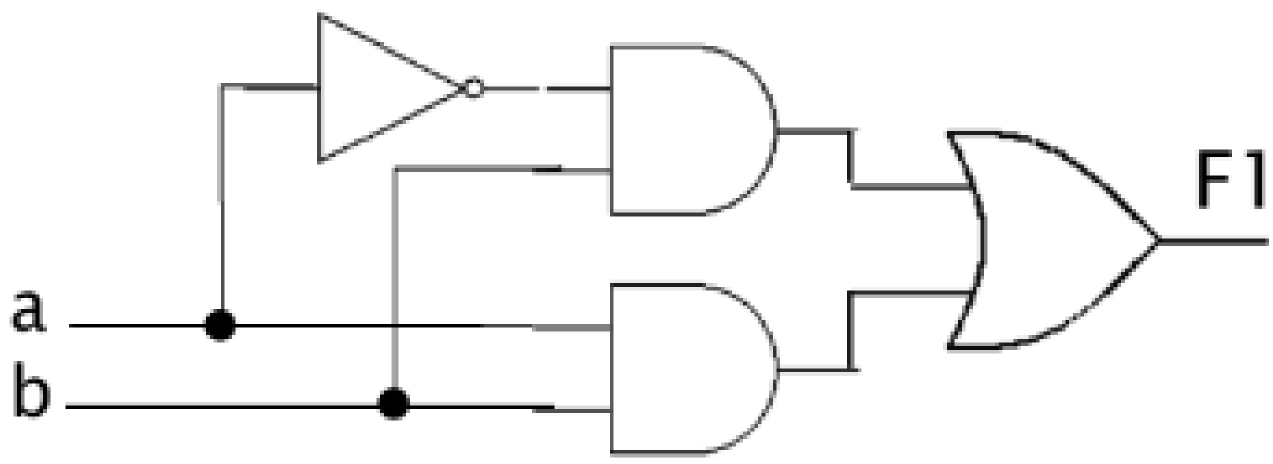
La porte OR représente un "ou (inclusif) logique"



Les opérations logiques Principales

- **Exemple** : Dessiner le schéma logique de la fonction logique F1 :

$$F1 = ab + \bar{a}b$$



Les opérations logiques Secondaires

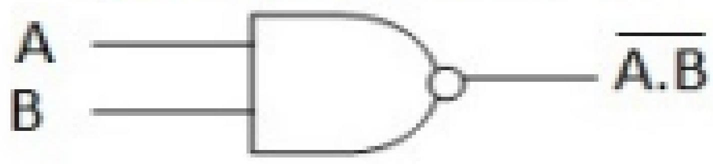
- Les opérations secondaires sont:
  - **NAND** logique (NOT AND : négation du produit),
  - **NOR** logique (NOT OR : négation de la somme)
  - **XOR** logique (OR exclusif).
  - **XNOR** logique (négation du OR exclusif)

Les opérations logiques Secondaires

- **NAND (NON ET)** : C'est le complément (inverse) du produit logique de deux variables logiques A et B noté comme suit :

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

- Le symbole graphique d'une porte logique NAND est celui d'une porte AND à la sortie de laquelle on ajoute la "boule" représentant la négation et il est représenté comme suit:



Porte NON ET (NAND)

Les opérations logiques Secondaires

- **NAND (NON ET)** : Une opération logique NAND fonctionne selon la table de vérité suivante :

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

NON ET (NAND)

Les opérations logiques Secondaires

- **NOR (NON OU)** : C'est l'équivalent d'une opération OU suivie d'une opération NON de la somme logique de deux variables logiques A et B notée : $A \ NOR \ B = \overline{A+B}$

- Le symbole graphique d'une porte logique NOR est celui d'une porte OR à la sortie de laquelle on ajoute la "boule" représentant la négation et il est représenté comme suit:

Les opérations logiques Secondaires

- **NOR(NON OU)** : L'opération logique NOR a la table de vérité suivante :

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

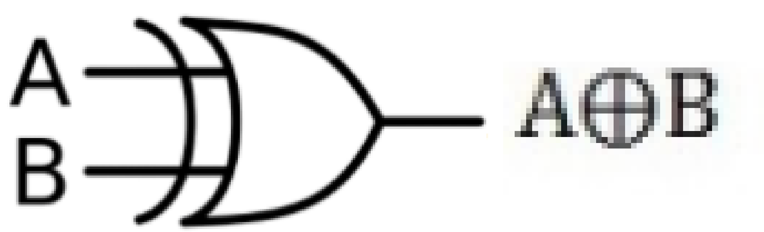
NON OU (NOR)

Les opérations logiques Secondaires

- **XOR (OU exclusif)** : Cette opération donne comme résultat 1, si et seulement si une des deux variables est égale à 1, elle est définit par :

$$A \; XOR \; B = A \oplus B$$

- Elle a pour représentation symbolique :



Les opérations logiques Secondaires

- **XOR (OU exclusif)** : L'opération logique XOR a la table de vérité suivante :

A	B	A *XOR* B
0	0	0
0	1	1
1	0	1
1	1	0

OU Exclusif (XOR)

Les opérations logiques Secondaires

- **XOR (OU exclusif) :**
- XOR est égal à 1 si et seulement si A = 1 ou B = 1 mais pas simultanément
- Une opération XOR fournit un comparateur d'inégalité : XOR ne vaut 1 que si A et B sont différents.
- Le complément du XOR correspond à un détecteur d'égalité.

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

Les opérations logiques Secondaires

- **XOR à plusieurs entrées :**

- Pour calculer le résultat de **S=A XOR B XOR C** , il faut d'abord faire l'opération entre deux termes, puis refaire un ou exclusif entre le résultat obtenu et le troisième terme.

- Ce qui se traduit par **S=(A XOR B) XOR C** ou par **S = A XOR (B XOR C)**

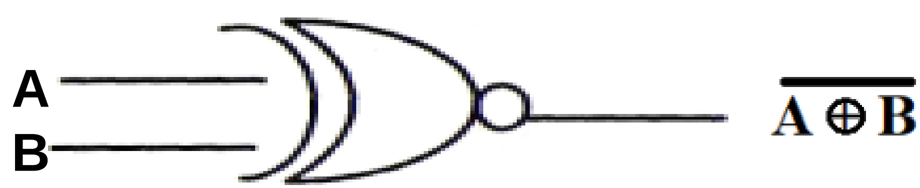
- On constate que l'appellation "Ou exclusif" n'est tout à fait exacte que pour deux variables. Avec trois variables, le résultat vaut 1 si une d'entre elles ou toutes les trois valent 1.

Les opérations logiques Secondaires

- **XNOR (NON OU exclusif)** : Cette opération prend la valeur 1 si et seulement les deux variables binaires A et B prennent la même valeur, pour tous les autres cas prenne la valeur 0, elle est définit par :

$$\overline{A \oplus B}$$

- Elle a pour représentation symbolique :



Les opérations logiques Secondaires

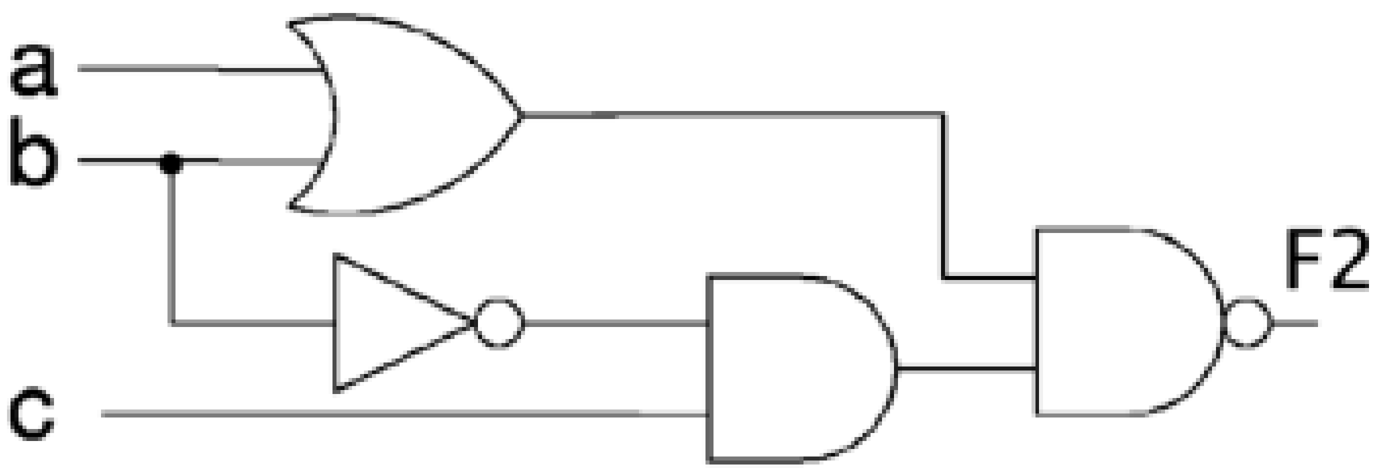
- **XNOR (NON OU exclusif)** : L'opération logique XNOR a la table de vérité suivante :

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

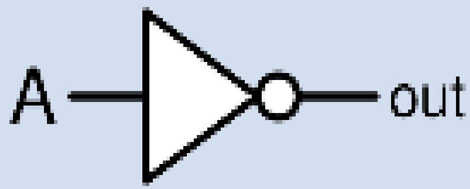
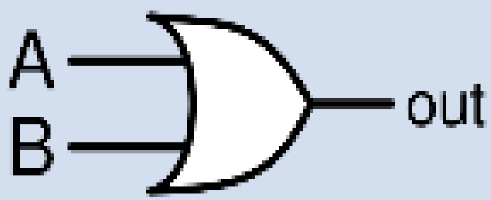
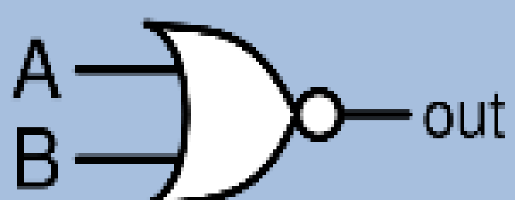
Les opérations logiques Secondaires

- **Exemple** : Dessiner le schéma logique de la fonction logique F2 :

$$F2 = \overline{(a+b)(\overline{b}.c)}$$



Résumé des portes logiques

<table>
<tr><th>Type</th><th>Description</th><th>Schéma universel</th><th>Formule booléenne (cf. lexique)</th><th colspan="3">Table de vérité (cf. lexique)</th></tr>
<tr><td colspan="7">Porte à 1 entrée « A » et 1 sortie « S »</td></tr>
<tr><td rowspan="3">NOT</td><td rowspan="3">Le NLS est l'inverse du niveau logique d'entrée.</td><td rowspan="3">
</td><td rowspan="3">$S = \overline{A}$</td><td>Etat de A</td><td colspan="2">Etat de S</td></tr>
<tr><td>0</td><td colspan="2">1</td></tr>
<tr><td>1</td><td colspan="2">0</td></tr>
<tr><td colspan="7">Portes à 2 entrées (« A » et « B ») et 1 sortie « S »</td></tr>
<tr><td rowspan="5">OR</td><td rowspan="5">Si une des 2 entrées est à 1 (ou les 2), le NLS est à 1 aussi</td><td rowspan="5">
</td><td rowspan="5">$S = A + B$</td><td>Etat de A</td><td>Etat de B</td><td>Etat de S</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td rowspan="5">NOR (NOT OR)</td><td rowspan="5">C'est l'opposé de la porte « OR », si aucune des entrées ne sont à 1, le NLS est à 1 aussi</td><td rowspan="5">
</td><td rowspan="5">$S = \overline{A + B}$</td><td>Etat de A</td><td>Etat de B</td><td>Etat de S</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>

Résumé des portes logiques

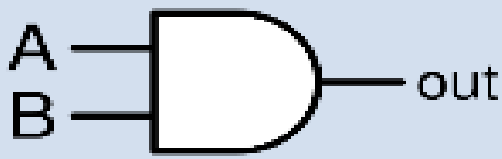
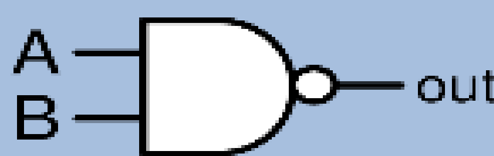
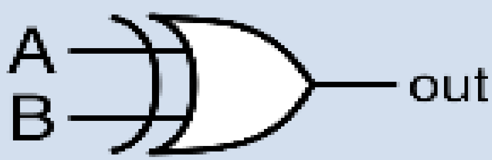
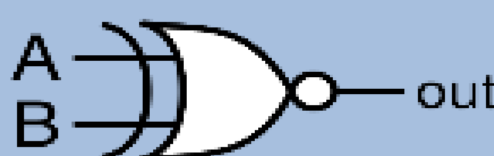
Type	Description	Schéma universel	Formule booléenne (cf. lexique)	Table de vérité (cf. lexique)		
Porte à 1 entrée « A » et 1 sortie « S »						
				Etat de A	Etat de B	Etat de S
AND	Si les 2 entrées sont à 1 alors le NLS est à 1		$S = A.B$	0	0	0
				0	1	0
				1	0	0
				1	1	1
				Etat de A	Etat de B	Etat de S
NAND (NOT AND)	C'est l'opposé de la porte « AND », si les 2 entrées ne sont pas à 1, le NLS est à 1		$S = \overline{A.B}$	0	0	1
				0	1	1
				1	0	1
				1	1	0
				Etat de A	Etat de B	Etat de S
XOR	Si seulement une seule des entrées est à 1, le NLS est à 1 aussi		$S = A \oplus B$	0	0	0
				0	1	1
				1	0	1
				1	1	0
				Etat de A	Etat de B	Etat de S
XNOR	Si les 2 entrés ont le même état, le NLS est à 1		$\overline{S = A \oplus B}$	0	0	1
				0	1	0
				1	0	0
				1	1	1

Table de vérité

- La **table de vérité**, est un moyen de définir la fonction d'un circuit ou d'une porte logique, qui contient simplement l'énumération des différents cas possibles et le résultat délivré par la porte ou le circuit pour chacun de ces cas.
- Une **table de vérité** est un tableau reprenant :
  - autant de **colonnes** qu'il y a d'**entrées** et de **sorties** de la porte (ou du circuit).
  - autant de **lignes** qu'il y a de **combinaisons possibles des entrées** (soit 2^N lignes de la table pour N entrées).

Table de vérité

- **Exemple 1 :**

$F1 = ab + \overline{a}b$, 2 variables

=> 4 valeurs possibles

a	b	ab	$\overline{a}b$	F1
0	0	0	0	0
0	1	0	1	1
1	0	0	0	0
1	1	1	0	1

Table de vérité

Exemple 2 :

$F2 = (a+b)(\overline{a}b+b\overline{c})$ 3 variables => 8 valeurs

a	b	c	$\overline{a}b$	$b\overline{c}$	$\overline{a}b+b\overline{c}$	a+b	F1
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	1	1	1	1	1
0	1	1	1	0	1	1	1
1	0	0	0	0	0	1	0
1	0	1	0	0	0	1	0
1	1	0	0	1	1	1	1
1	1	1	0	0	0	1	0

Expression d'une fonction logique

- L'écriture de l'expression d'une fonction logique combinatoire, vous permet d'apprendre à écrire l'expression d'une fonction logique sous forme **canonique** à partir de sa table de vérité.
- Cependant, pour les réalisations pratiques, il faut penser à simplifier ces expressions pour réduire le coût du matériel et le temps de câblage. **La première technique** de simplification que nous avons vu est la simplification à l'aide des **règles de l'algèbre booléenne.**

Expression d’une fonction logique

- Une expression est **canonique** lorsque tous ses termes renferment toutes les variables, soit sous forme directe, soit sous forme complémentée.
- Pour toute fonction, il est possible d'établir l'expression canonique sous deux formes :
  - La **Première forme canonique** ou **forme disjonctive** ou forme **"somme de produits"**.
  - La **deuxième forme canonique** ou **forme conjonctive** ou forme **"produit de sommes"**.

Forme disjonctive

- **Première forme canonique (forme disjonctive):** c'est la **somme logique** (ou réunion "+") des **mintermes** associés aux combinaisons pour lesquelles la fonction vaut 1 (**somme de produits**).
- Un **minterme** est défini comme étant le produit logique des variables booléennes considérées avec la convention suivante :
  - Si la variable est égale à 1 alors inscrire la variable elle-même.
  - Si la variable est égale à 0 alors inscrire son complément.

Forme disjonctive

- **Exemple 1 :**

Si on considère 4 variables a, b, c et d,

m = a.b.c.d est un minterme,

$m = a.b.\bar{c}.\bar{d}$ est un autre minterme,

m = a.b.d n'est pas un minterme.

Forme disjonctive

- **Exemple 2:** Les mintermes de la table de vérité :

A	B	C	F
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

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

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

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

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

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

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

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

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

Forme disjonctive

- Si chacun des produits contient toutes les variables d'entrée sous une forme directe ou complémentée, alors la forme est appelée « **première forme canonique** » ou « **forme canonique disjonctive** » ou forme **somme de produit standard** (SdP) . Chacun des produits est alors appelé **minterme.**
- Exemple de forme canonique disjonctive :

$$F(x,y,z) = \bar{x}.\bar{y}.z + x.y.\bar{z} + x.\bar{y}.z$$

Forme disjonctive

- Nous pouvons écrire la forme disjonctive d'une fonction logique d'une autre façon dite numérique (voir l'exemple suivant).

Forme disjonctive

- **Exemple** : Soit une fonction F de trois variables définie par sa table de vérité :

Combinaison
0
1
2
3
4
5
6
7

A	B	C	F
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

Forme disjonctive

- **Exemple :**

Nous remarquons que la fonction F(A,B,C) est à l'état 1 pour les combinaisons 3, 5, 6 et 7. On l'écrit sous une forme dite numérique : $F(A,B,C) = \sum m(3,5,6,7)$, (c'est-à-dire réunion des combinaisons 3, 5, 6 et 7). **D'où la première forme canonique de F** :

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

Forme conjonctive

- **Deuxième forme canonique (forme conjonctive):** c'est le produit logique (ou intersection ".") des maxtermes associés aux combinaisons pour lesquelles la fonction vaut 0 (produit de sommes).
- Un « **maxterme** » il est défini comme étant la somme logique des variables booléennes considérées avec la convention suivante :
  - si la variable est égale à 0 alors inscrire la variable elle-même.
  - si la variable est égale à 1 alors inscrire son complément.

Forme conjonctive

- **Exemple 1 :**

Si on considère 4 variables a, b, c et d,

$M = a + b + c + d$ est un maxterme,

$M = a + \overline{b} + c + d$ est un autre maxterme,

$M = a + b + d$ n'est pas un maxterme.

Forme conjonctive

- **Exemple 2:** Les maxtermes de la table de vérité :

A	B	C	F
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 conjonctive

- Si chacune des sommes contient toutes les variables d'entrée sous une forme directe ou complémentée, alors la forme est appelée « **deuxième forme canonique** » ou « **forme canonique conjonctive** » ou forme **produit de somme standard** (PdS). Chacune des sommes est alors appelée **maxterme.**
- Exemple de forme canonique conjonctive :

$$F(x,y,z) = (x+y+z).(x+\bar{y}+\bar{z}).(\bar{x}+y+z)$$

Forme conjonctive

- Nous pouvons écrire la forme conjonctive d'une fonction logique d'une autre façon dite numérique (voir l'exemple suivant).

Forme conjonctive

- **Exemple** : Soit une fonction F de trois variables définie par sa table de vérité :

Combinaison
0
1
2
3
4
5
6
7

A	B	C	F
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

Forme conjonctive

- **Exemple :**

Nous remarquons la fonction F(A,B,C) est à l'état 0 pour les combinaisons 0, 1, 2 et 4. On l'écrit sous la forme numérique : F(A,B,C) $= \prod M(0,1,2,4)$ (c'est-à-dire intersection des combinaisons 0, 1, 2 et 4). D'où la deuxième forme canonique de F :

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

Exercice

- On a trois juges qui contrôlent le départ d'une course. La course a lieu si au moins deux des trois juges sont prêts. Créer le circuit logique qui représente le départ d'une course.
- **Solution :**
  - Les trois juges forment les trois entrées : A, B et C.
  - Le départ de la course représente la sortie F. On peut ensuite créer manuellement la table de vérité de cette fonction

Exercice

A	B	C	F(A,B,C)
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

Exercice

On peut ensuite exprimer cette fonction comme une somme de mintermes : F $= \sum m(3,5,6,7)$. Puis on va simplifier la fonction.

$$F = \sum \mathrm{m}(3,5,6,7)$$

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

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

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

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

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

$$\boxed{\text{F} = C.B + C.A + A.B}$$

Passage de la forme normale à la forme canonique

- La forme d'une fonction ou les termes ne contiennent pas toutes les variables est appelé forme **normale.**

- Pour convertir une fonction normale en une forme **canonique** il faut :

1. Multiplier un **minterme** avec une expression qui vaut un.

2. Additionnez un **maxterme** par une expression qui vaut zéro.

3. Faire la distribution.

Passage de la forme normale à la forme canonique

Exemple :

Soit la fonction :

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

1. Ecrire f sous la première forme canonique
2. Ecrire f sous la deuxième forme canonique

Les formes canoniques

Exemple : f sous la première forme canonique

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

$$f(a,b,c) = a.1.1 + 1.b.\bar{c}$$

$$f(a,b,c) = a.\left(b + \bar{b}\right)(c + \bar{c}) + (a + \bar{a}).b.\bar{c}$$

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

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

$$f(a,b,c) = m_7 + m_6 + m_5 + m_4 + m_2$$

$$= \sum m(2,4,5,6,7)$$

Les formes canoniques

Exemple : f sous la deuxième forme canonique

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

$$f(a,b,c) = (a+b+0).(a+\bar{c}+0)$$

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

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

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

$$f(a,b,c) = M_0.M_1.M_3 = \prod M(0,1,3)$$

Passage d'une forme canonique à une autre

- Pour passer de la première forme canonique à la deuxième forme canonique, il suffit de remplacer tous les mintermes de la fonction par les maxtermes ayant des indices différents. De même ; pour passer de la deuxième forme canonique à la première forme canonique, il suffit d'utiliser une procédure similaire.

Passage d'une forme canonique à une autre

Exemple :

Convertissez la forme canonique disjonctive suivante en forme conjonctive équivalente :

$$f(a,b,c) = \bar{a}.\bar{b}.\bar{c} + \bar{a}.b.\bar{c} + \bar{a}.b.c + a.\bar{b}.c + a.b.c = \sum m(0,2,3,5,7)$$

- D'où

$$f(a,b,c) = (a + b + \bar{c}).(\bar{a} + b + c).(\bar{a} + \bar{b} + c) = \prod M(1,4,6)$$

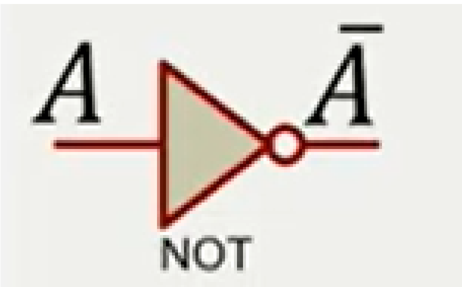
Les fonctions logiques universelles

- Une fonction est universelle lorsqu'elle permet, à elle seule, d'exprimer les fonctions de base : *NOT, AND, OR*.

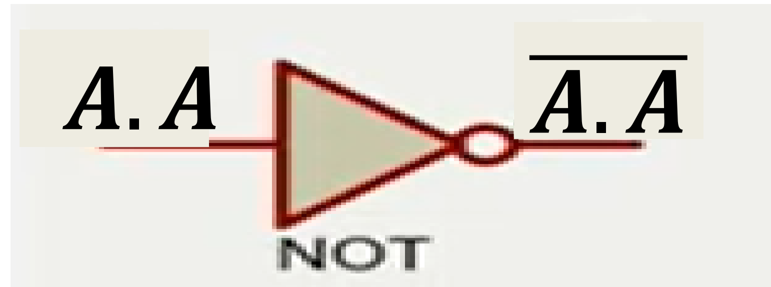
- Les portes logique **NAND** et **NOR** sont des portes logiques universelles car elle permettent de réaliser toutes les opérations logiques élémentaires.

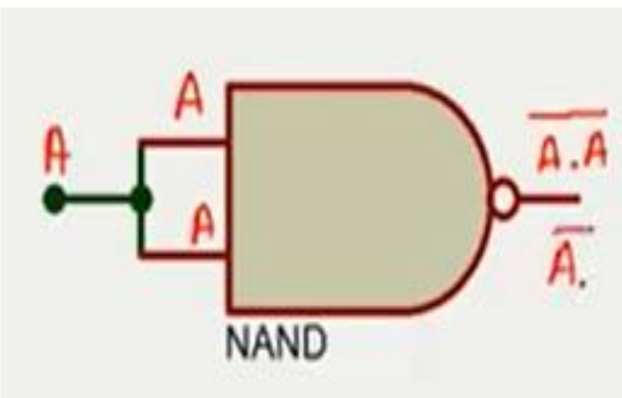
Les fonctions logiques avec NAND

- NOT :

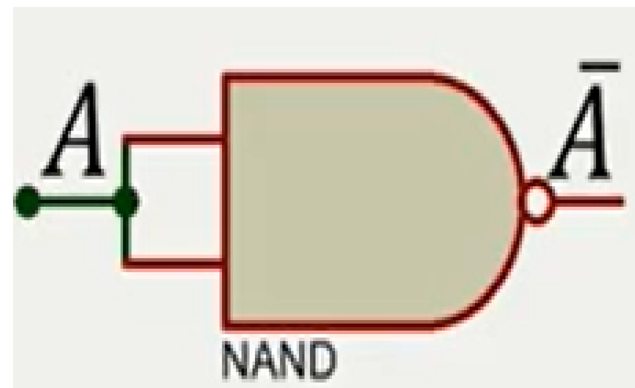


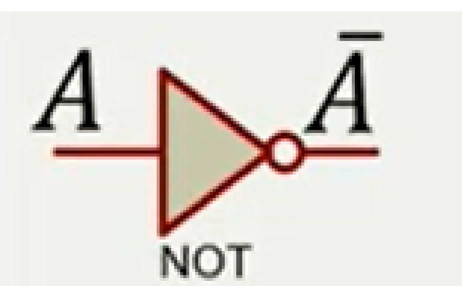
=



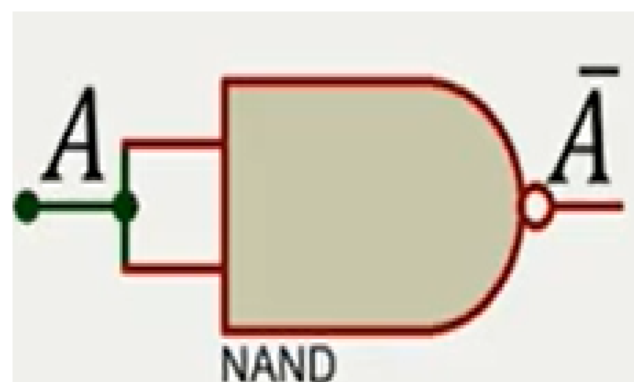


=



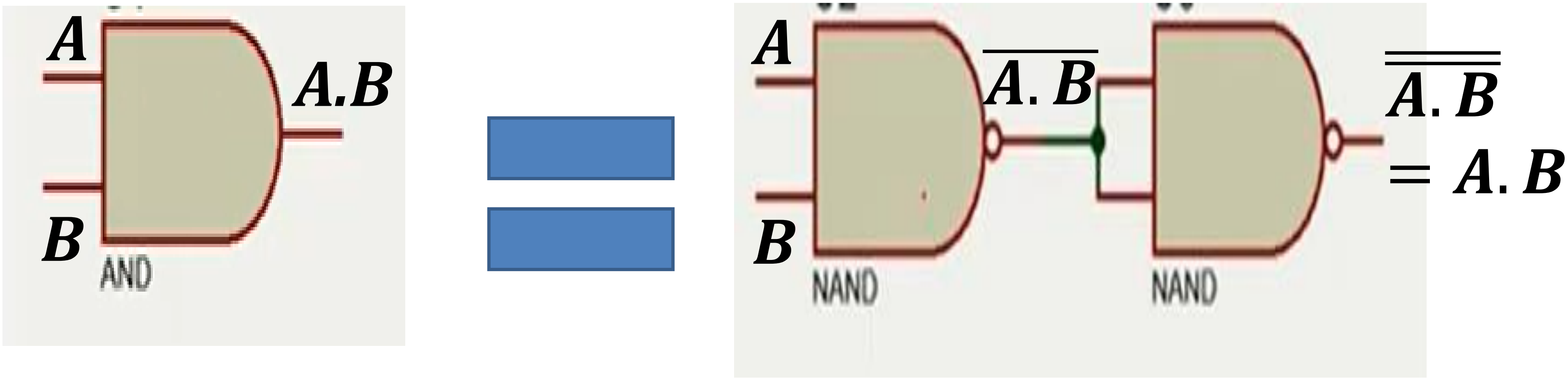


=



Les fonctions logiques avec NAND

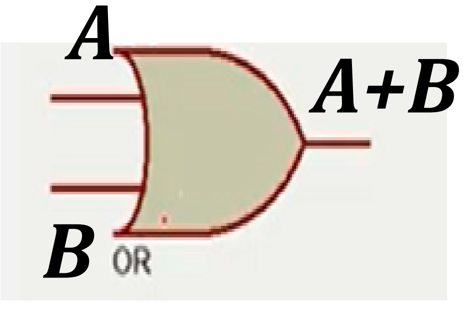
- **AND :**



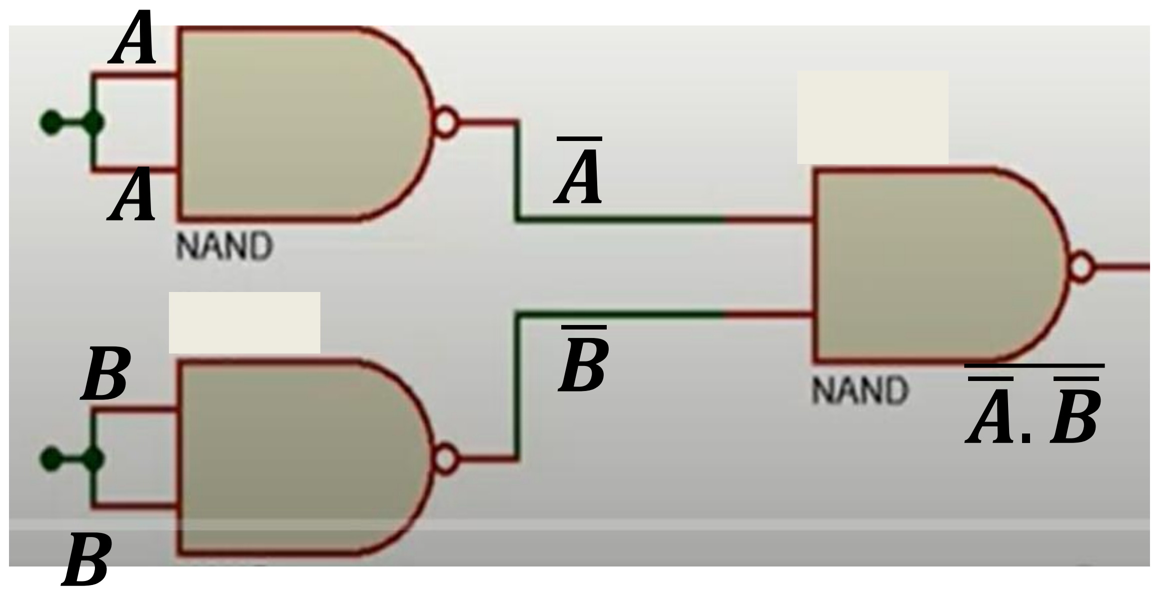
$$A.B = \overline{\overline{A.B}}$$

Les fonctions logiques avec NAND

- **OR :**



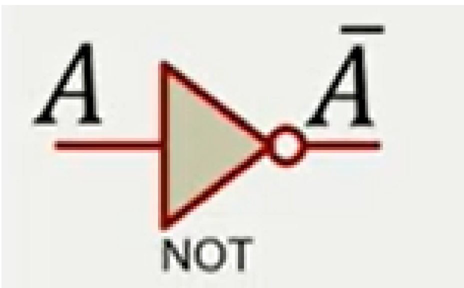
=



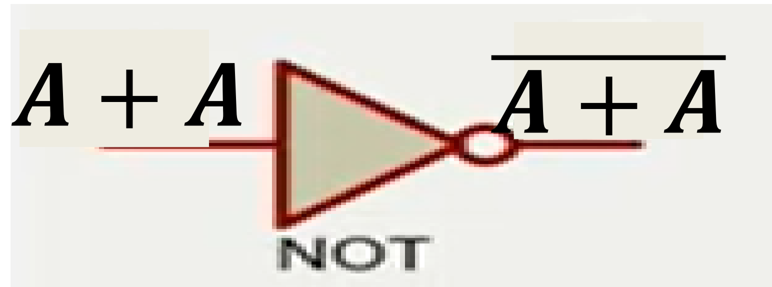
$$A + B = \overline{\overline{A + B}} = \overline{\overline{A}\,.\,\overline{B}}$$

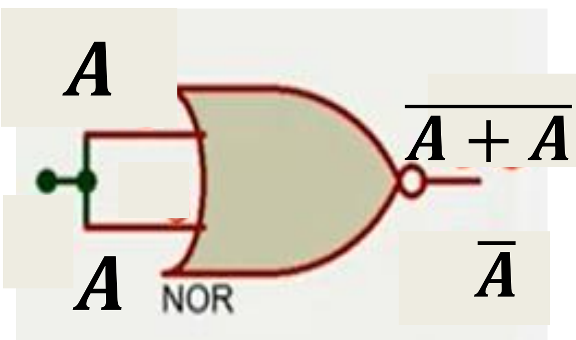
Les fonctions logiques avec NOR

- **NOT :**

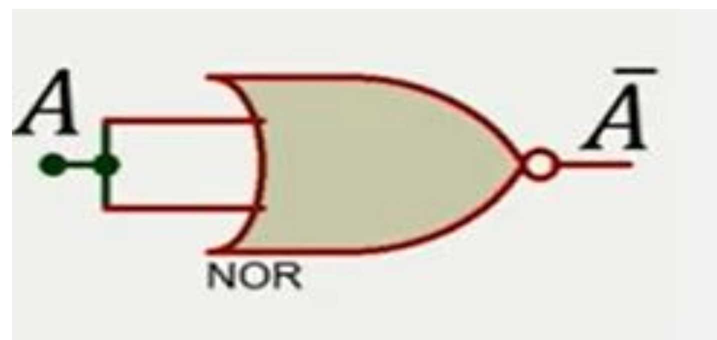


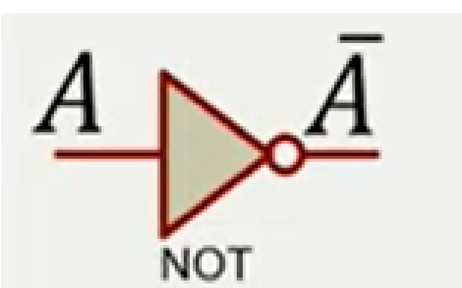
=



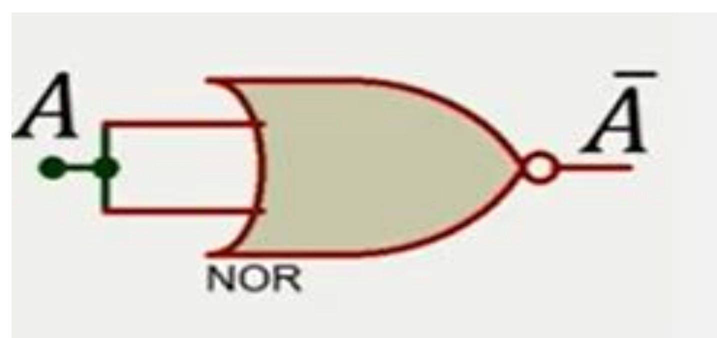


=



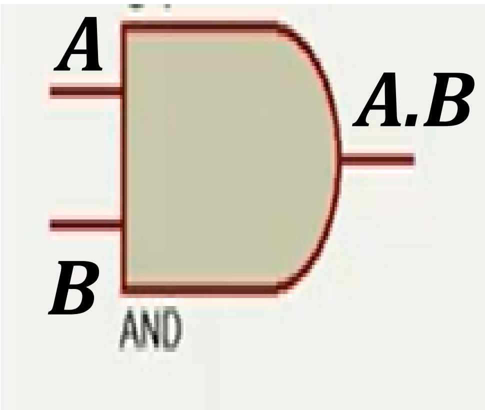


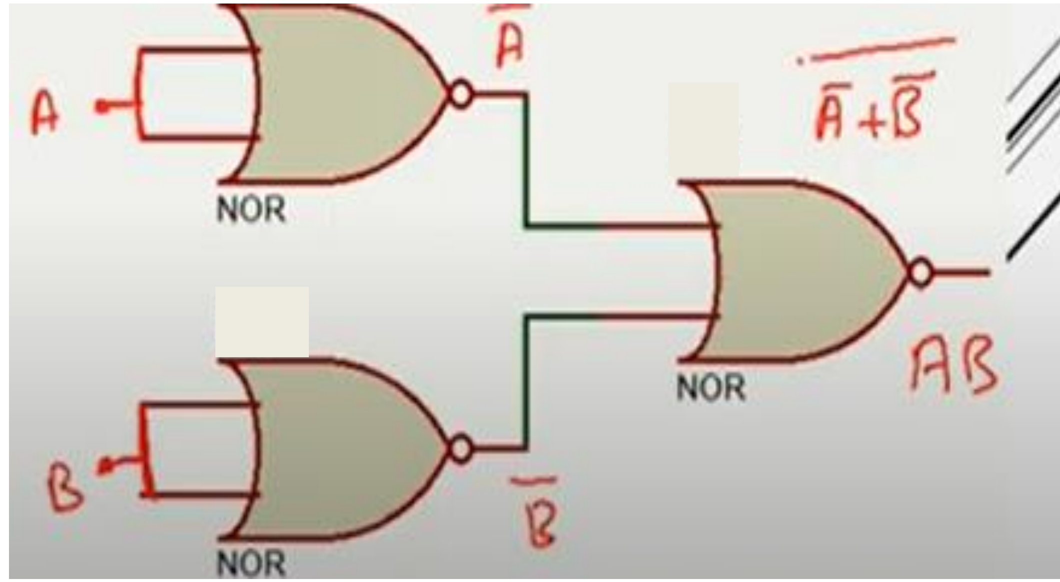
=



Les fonctions logiques avec NOR

- **AND :**

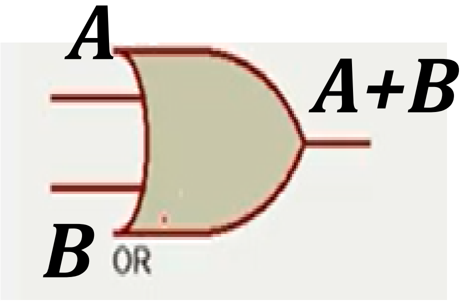


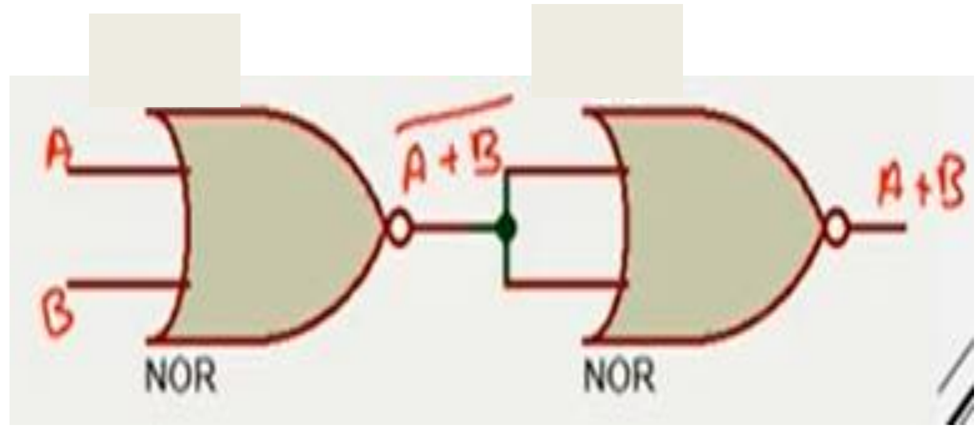


$$A.B = \overline{\overline{A.B}} = \overline{\overline{A} + \overline{B}}$$

Les fonctions logiques avec NOR

- **OR :**





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

Application

Exercice 4 du TD3 :

1. Réalisez les logigrammes des fonctions suivantes :

$\boldsymbol{F} = \bar{A}.\bar{B}.\bar{C} + \bar{C}.D$ Seulement avec des portes NOR,

$\boldsymbol{G} = A.(B + C)$ Seulement avec des portes NAND

$\mathbf{H} = A.B + B.C + A.C$

Seulement avec des portes NAND

2. Simplifier la fonction suivante et dessiner son logigramme à l'aide des portes NAND puis à l'aide des portes NOR à deux entrées :

$$K = B.\bar{C}.\bar{D} + A.B.\bar{D} + \bar{A}.B.C.\bar{D}$$

Application

□ Exercice 4 du TD3 :

1. Réalisez les logigrammes des fonctions suivantes :

$\boldsymbol{F} = \bar{A}.\bar{B}.\bar{C} + \bar{C}.D$ Seulement avec des portes NOR,

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

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

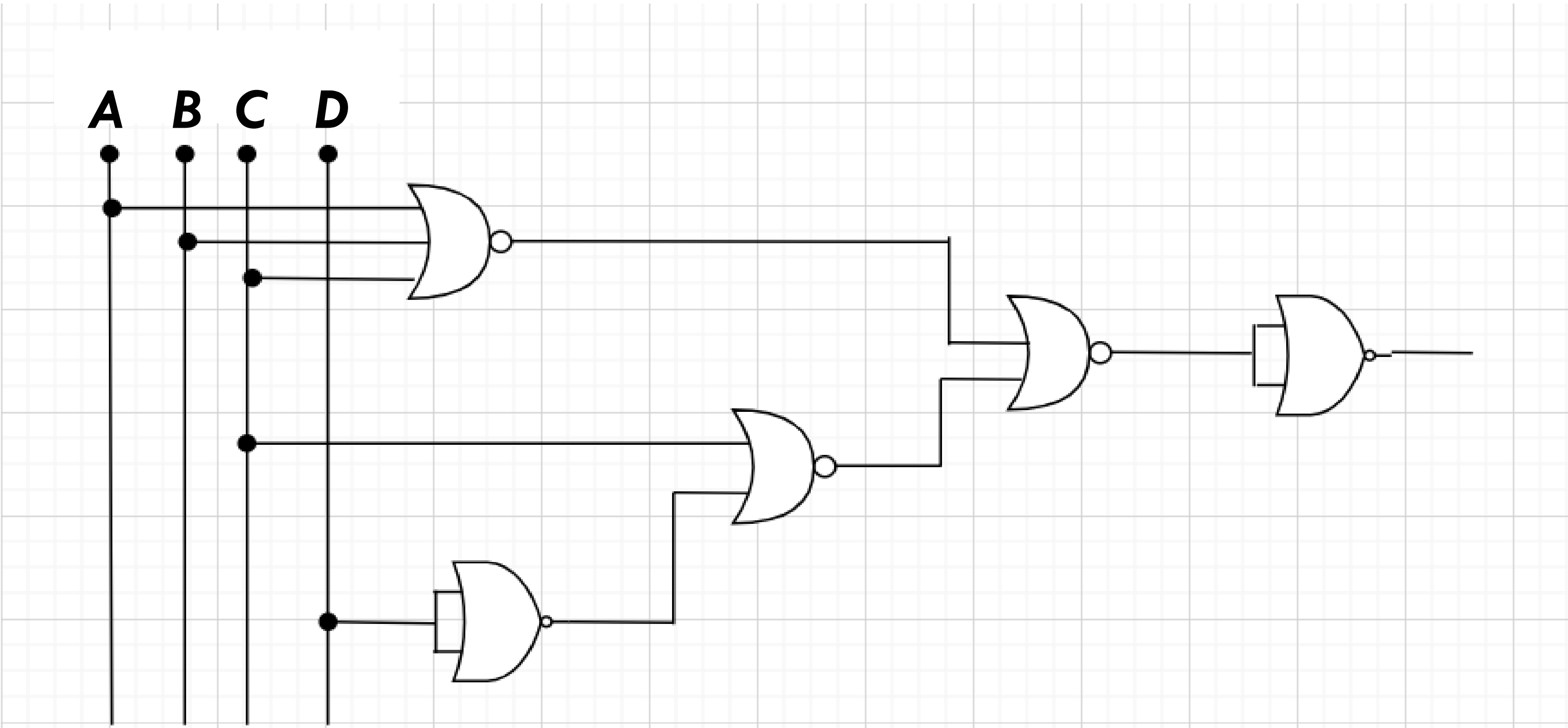
$$= \overline{\overline{\overline{A + B + C} + \overline{C + \bar{D}}}}$$

Application

Exercice 4 du TD3 :

1. Réalisez les logigrammes des fonctions suivantes :

$$\boldsymbol{F} = \bar{A}.\bar{B}.\bar{C} + \bar{C}.D$$



Application

- **Exercice 4 du TD3 :**

1. Réalisez les logigrammes des fonctions suivantes :

$\boldsymbol{G} = A.(B + C)$ Seulement avec des portes NAND

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

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

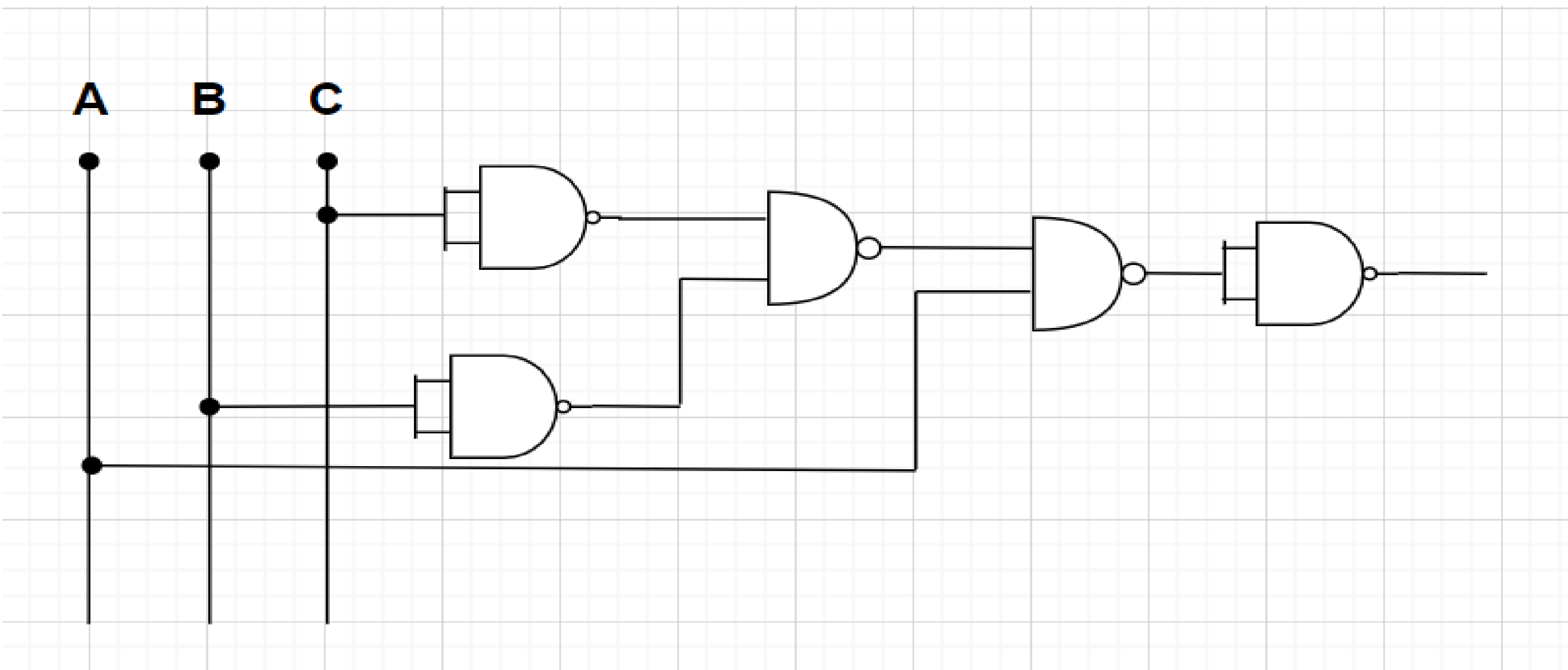
$$= \overline{\overline{A.\overline{(\bar{B}.\bar{C})}}}$$

Application

- **Exercice 4 du TD3 :**

1. Réalisez les logigrammes des fonctions suivantes :

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



Application

□ Exercice 4 du TD3 :

1. Réalisez les logigrammes des fonctions suivantes :

$$\mathbf{H} = A.B + B.C + A.C$$

Seulement avec des portes NAND

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

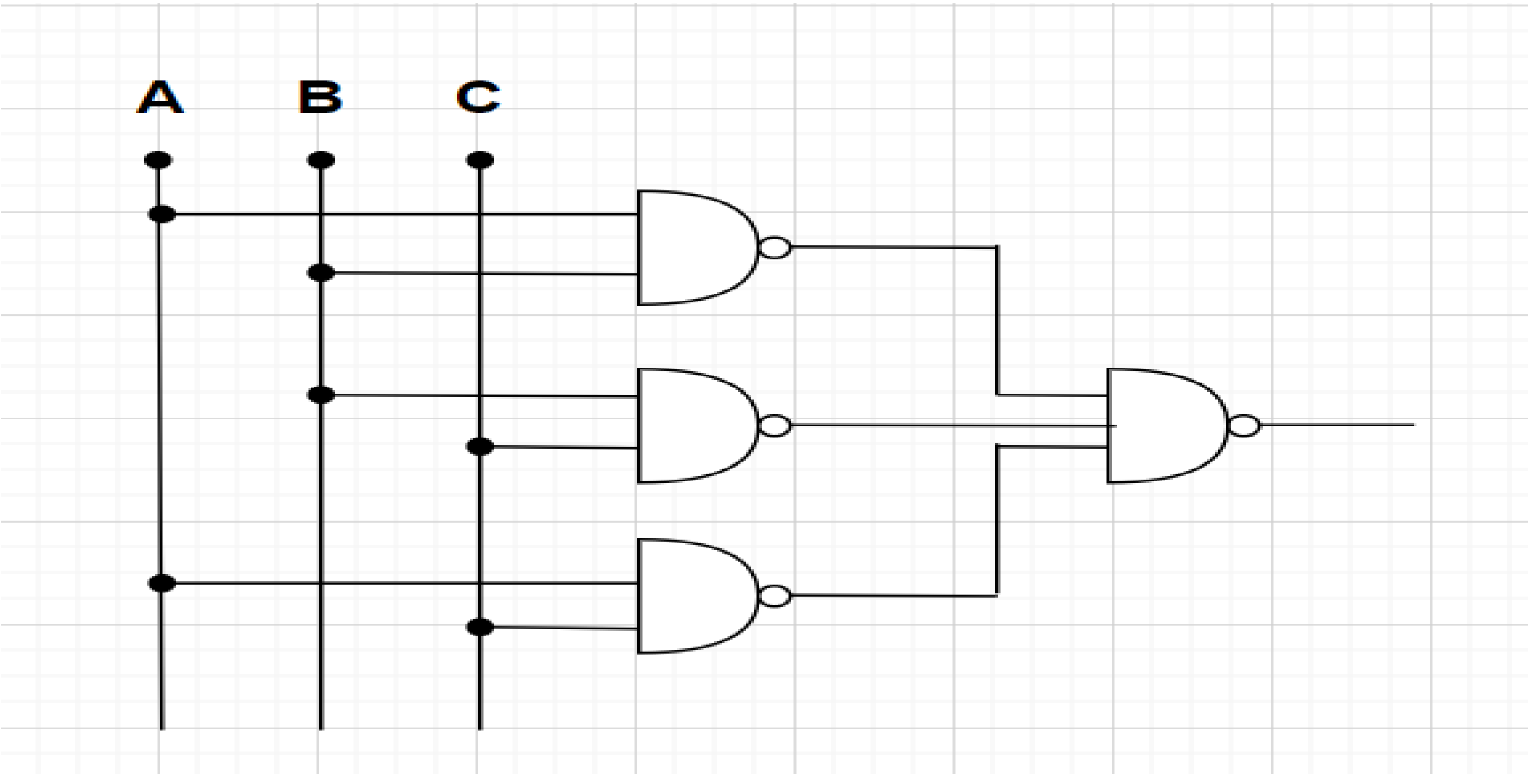
$$= \overline{\overline{A.B}.\overline{B.C}.\overline{A.C}}$$

Application

- **Exercice 4 du TD3 :**

1. Réalisez les logigrammes des fonctions suivantes :

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



Application

□ Exercice 4 du TD3 :

2. Simplifier la fonction suivante et dessiner son logigramme à l'aide des portes NAND puis à l'aide des portes NOR à deux entrées :

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

Application

- **Exercice 4 du TD3 :**

2. $K = B.\bar{C}.\bar{D} + A.B.\bar{D} + \bar{A}.B.C.\bar{D}$

- $K = B.\bar{C}.\bar{D} + A.B.\bar{D} + \bar{A}.B.C.\bar{D}$
- $= B.\bar{D}\,(\bar{C} + \bar{A}.C) + A.B.\bar{D}$
- $= B.\bar{D}\,(\bar{C} + \bar{A}) + A.B.\bar{D}$
- $= B.\bar{D}\,.\bar{C} + B.\bar{D}\,.\bar{A} + A.B.\bar{D}$
- $= B.\bar{D}\,.\bar{C} + B.\bar{D}\,.(\bar{A} + A)$
- $= B.\bar{D}\,.\bar{C} + B.\bar{D}$
- $= B.\bar{D}\,.(\bar{C} + 1)$
- $\boldsymbol{K = B.\overline{D}}$

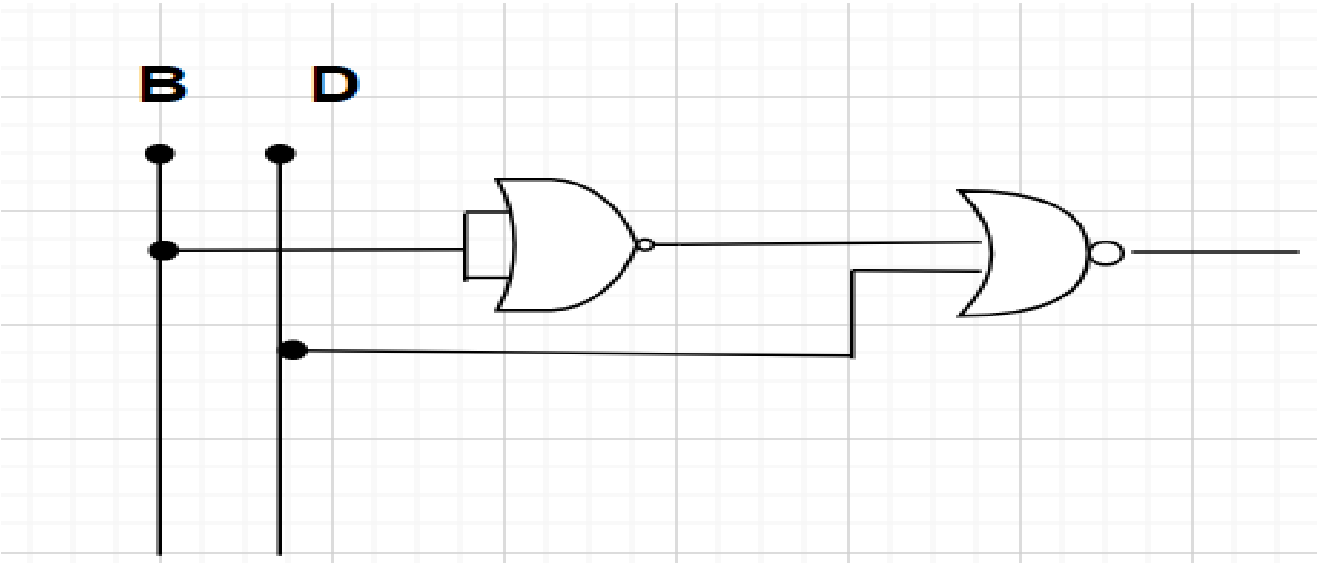
Application

- **Exercice 4 du TD3 :**

Logigramme avec portes NOR à deux entrées :

$$K = B.\overline{D}$$

$$K = \overline{\overline{B.\overline{\overline{D}}}} = \overline{\overline{B} + D}$$



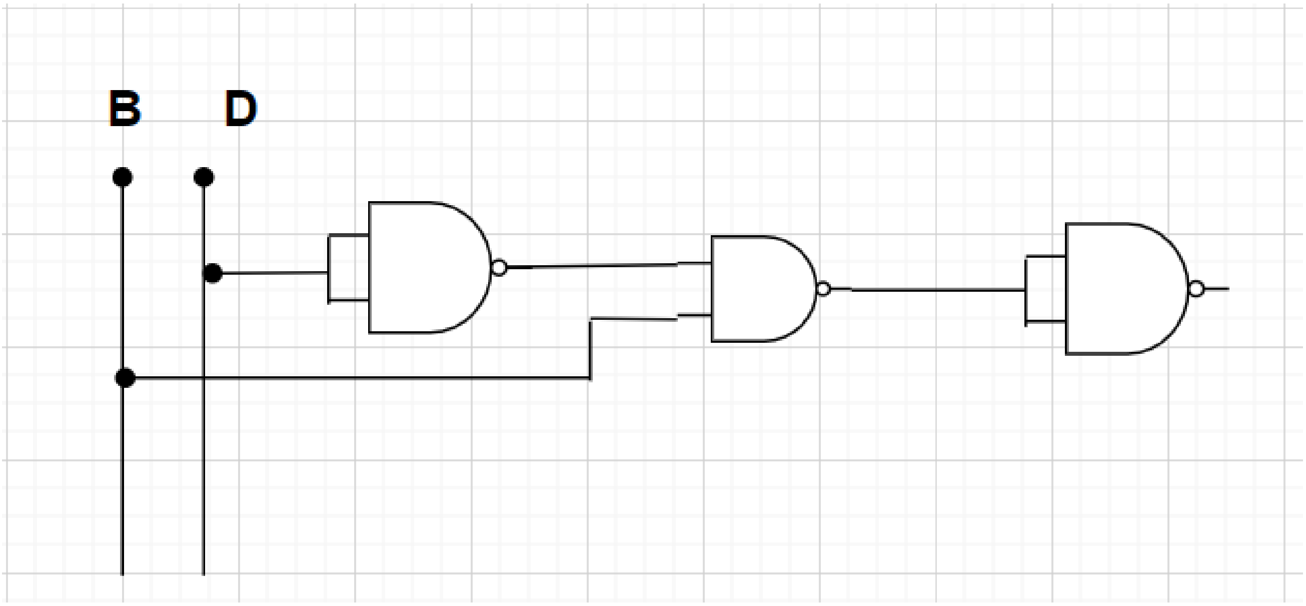
Application

□ Exercice 4 du TD3 :

Logigramme avec portes NAND à deux entrées :

$$K = B.\overline{D}$$

$$K = \overline{\overline{B.\overline{\overline{D}}}}$$



Application

□ Exercice 4 du TD3 :

Question supplémentaire :

- Réalisez le logigramme de H avec seulement des **portes NAND à deux entrées** :

$$\boldsymbol{H} = A.B + B.C + A.C$$

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

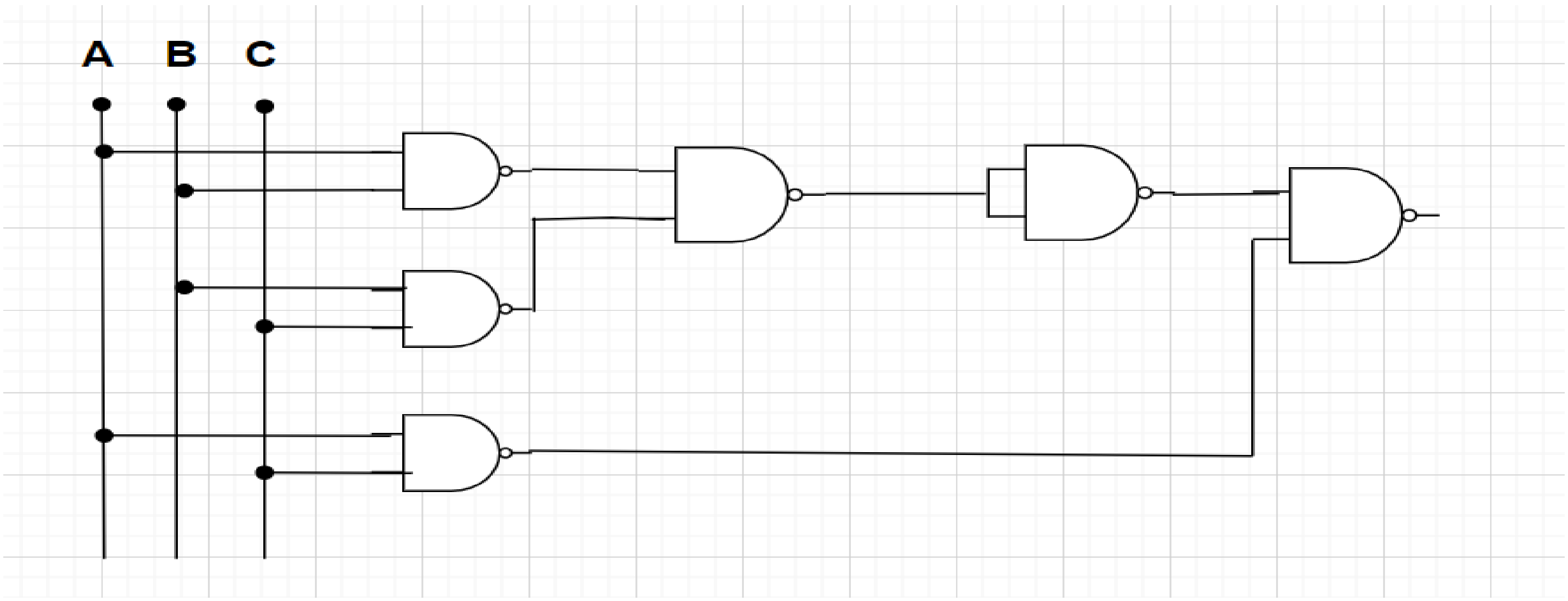
$$= \overline{\overline{A.B}.\overline{B.C}.\overline{A.C}} = \overline{\overline{\overline{\overline{A.B}.\overline{B.C}}}.\overline{A.C}}$$

Application

□ Exercice 4 du TD3 :

<u>Question supplémentaire :</u>

- Réalisez le logigramme de H avec seulement des **<u>portes NAND à deux entrées</u>** :



Application

□ Exercice 4 du TD3 :

<u>Question supplémentaire :</u>

- Réalisez le logigramme de F avec seulement des **<u>portes NOR à deux entrées</u>** :

$$\boldsymbol{F} = \bar{A}.\bar{B}.\bar{C} + \bar{C}.D$$

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

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

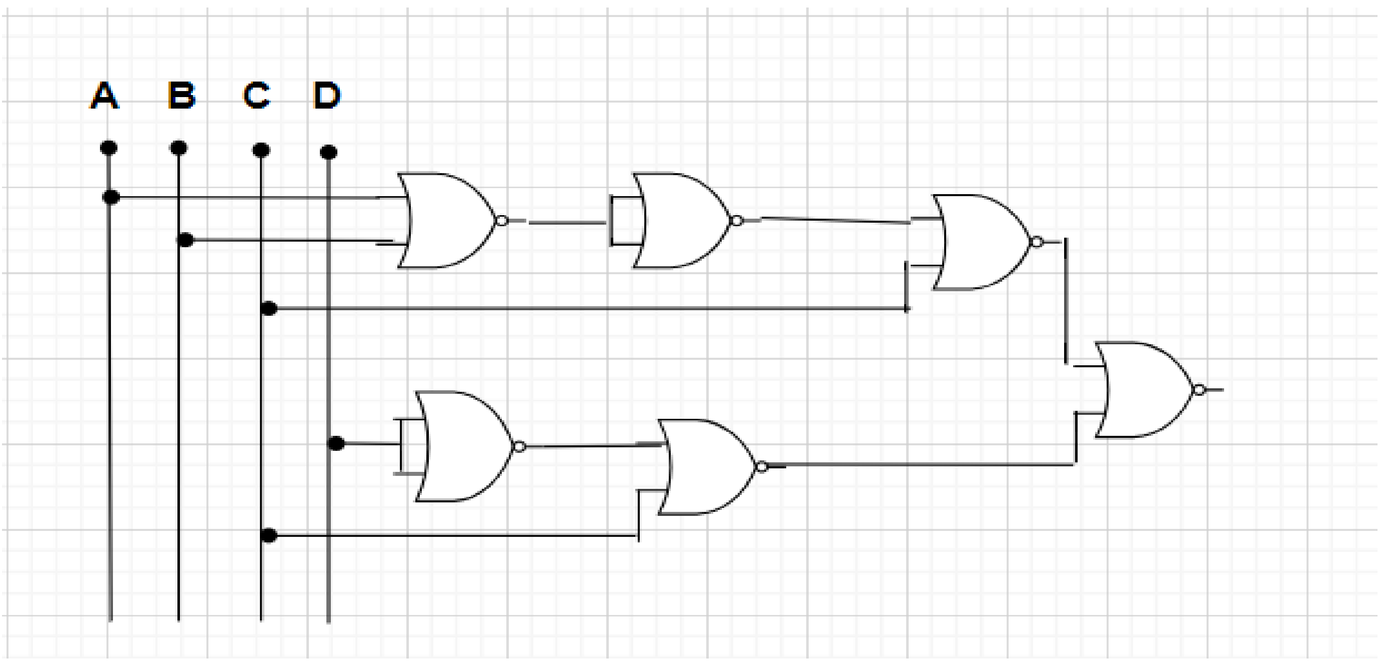
$$= \overline{\overline{\overline{\boldsymbol{A+B+C}} + \overline{\boldsymbol{C+\bar{D}}}}} = \overline{\overline{\overline{\overline{\overline{\boldsymbol{A+B}}} + \boldsymbol{C}} + \overline{\boldsymbol{C+\bar{D}}}}}$$

Application

□ Exercice 4 du TD3 :

<u>Question supplémentaire :</u>

- Réalisez le logigramme de F avec seulement des **<u>portes NOR à deux entrées</u>** :



Révisions

Exercice 1 : Soit la table de vérité suivante :

A	B	C	G
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

- Donnez l'expression de G sous forme de somme de produits.
- Simplifiez G par les règles de l'algèbre de BOOLE
- Réalisez le logigramme de G en utilisant uniquement que les portes NAND à 2 entrées.

Révisions

Exercice :

- **L'expression de G :**

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

Révisions

- **Simplification :**

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

$= \bar{A}.\bar{B}.(\bar{C} + C) + \bar{A}.B.\bar{C} + A.\bar{B}.\bar{C}$	Factorisation
$= \bar{A}.\bar{B}.1 + \bar{A}.B.\bar{C} + A.\bar{B}.\bar{C}$	Complémentarité $a+\bar{a}=1$
$= \bar{A}.\bar{B} + \bar{A}.B.\bar{C} + A.\bar{B}.\bar{C}$	Elément neutre a.1=a .1=1
$= \bar{A}.(\bar{B} + B.\bar{C}) + A.\bar{B}.\bar{C}$	Factorisation
$= \bar{A}.(\bar{B} + \bar{C}) + A.\bar{B}.\bar{C}$	Absorption $a + \bar{a}.b = a + b$
$= \bar{A}.\bar{B} + \bar{A}.\bar{C} + A.\bar{B}.\bar{C}$	Distributivité du + et .
$= \bar{A}.\bar{B} + \bar{C}.(\bar{A} + A.\bar{B})$	Factorisation
$= \bar{A}.\bar{B} + \bar{C}.(\bar{A} + \bar{B})$	Absorption $a + \bar{a}.b = a + b$

$$\boxed{G = \bar{A}.\bar{B} + \bar{C}.\bar{A} + \bar{C}.\bar{B}}$$

Révisions

- Le logigramme de G en utilisant uniquement des portes NAND à 2 entrées :

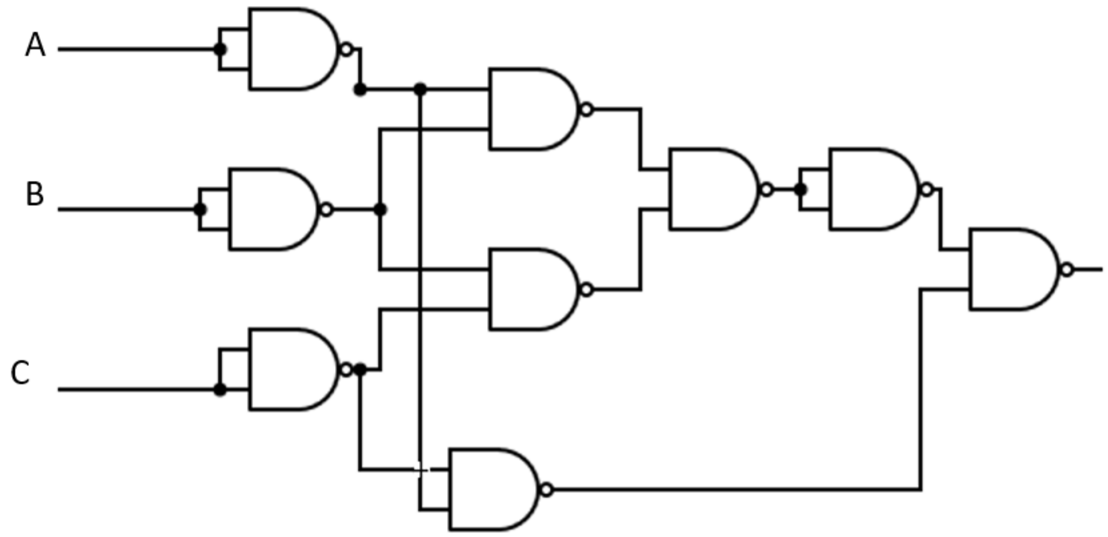
$$G = \overline{\overline{\bar{A}.\bar{B} + \bar{A}.\bar{C} + \bar{B}.\bar{C}}}$$

$$= \overline{\overline{\bar{A}.\bar{B}}\,.\,\overline{\bar{A}.\bar{C}}\,.\,\overline{\bar{B}.\bar{C}}}$$

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

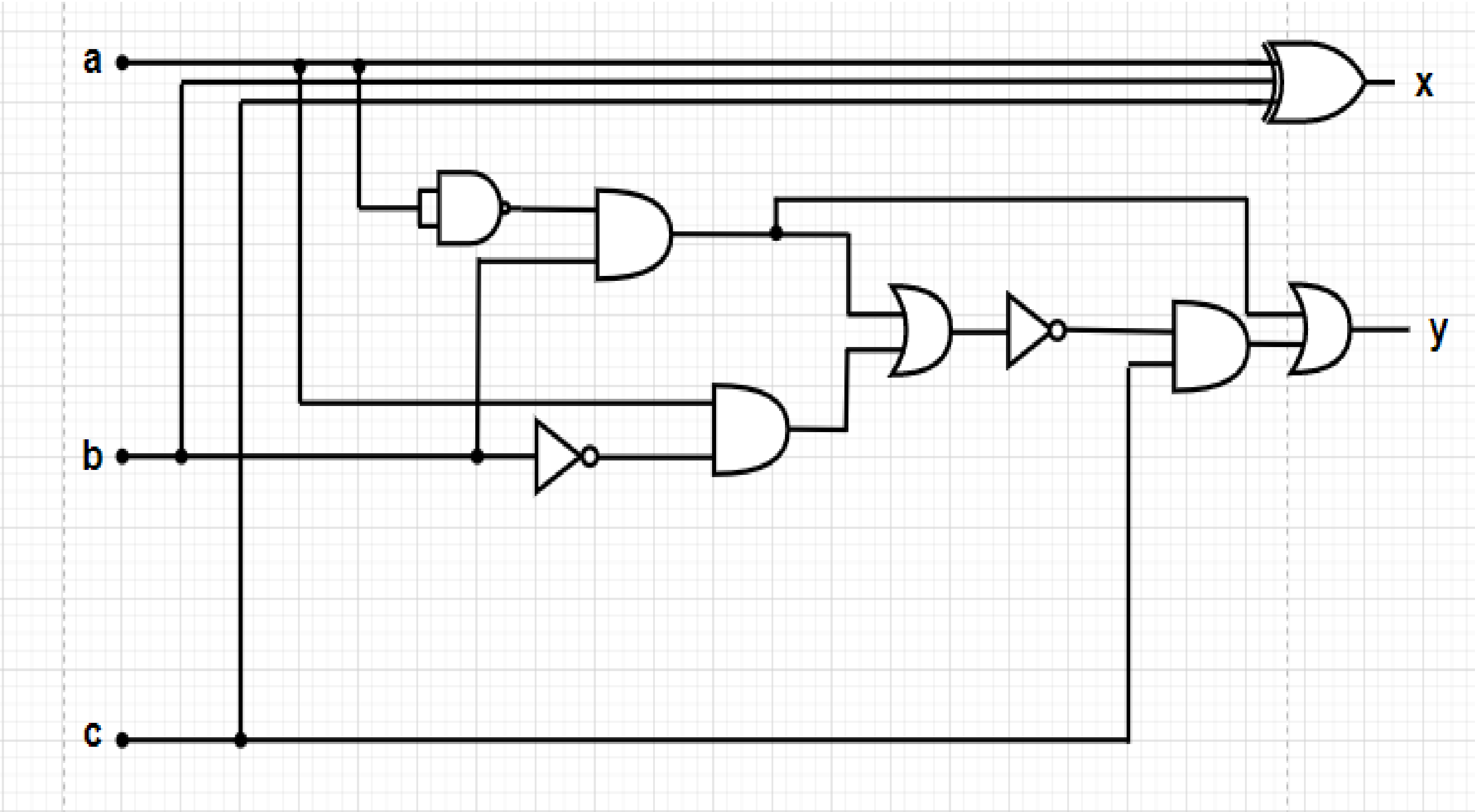
Révisions

- Le logigramme de G en utilisant uniquement des portes NAND à 2 entrées :



Révisions

Exercice 2 : Soit la logigramme suivant :



Révisions

Exercice 2 :

1) Etablir les expressions logiques des sorties x et y :

$$x = a \oplus b \oplus c$$

$$y = \bar{a}.b + c.\overline{(\bar{a}.b + a.\bar{b})}$$

2) Dresser la table de vérité correspondantes aux sorties x et y :

a	b	c	x	y
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Conclusion :

Dans ce chapitre nous avons les principaux points suivants :

- Algèbre de Boole (propriétés)
- Simplification de fonction.
- Fonctions logiques ,opération logique, variable logique.
- Portes logiques
- Circuit logiques (=logigramme=schéma logique)
- Table de vérité
- Formes canoniques disjonctive et conjonctive.
- Réalisation des circuits à l'aide seulement des portes NAND ou NOR.