

# TD Chapitre 01 : Logique Booléenne

## Objectifs

- Maîtriser les tables de vérité
- Construire des portes à partir de NAND
- Appliquer les lois de De Morgan
- Comprendre le Mux et le DMux

**Durée estimée :** 1h30

## Exercice 1 : Tables de Vérité (Échauffement)

**Objectif :** Remplir des tables de vérité à partir des définitions

### 1.1 Table de la porte NAND

Complétez la table de vérité de  $\text{NAND}(A, B)$  :

A	B	NAND
0	0	?
0	1	?
1	0	?
1	1	?

► Solution

## 1.2 Table de la porte XOR

Complétez la table de vérité de XOR(A, B) :

A	B	XOR
0	0	?
0	1	?
1	0	?
1	1	?

► Solution

## Exercice 2 : Construction à partir de NAND

**Objectif :** Prouver que NAND est universel

### 2.1 Construire NOT

Comment construire un NOT en utilisant uniquement une porte NAND ?

**Indice :** Que se passe-t-il si on connecte le même signal aux deux entrées ?

► **Solution**

## 2.2 Construire AND

Comment construire un AND en utilisant NOT et NAND ?

**Indice :**  $\text{NAND} = \text{NOT}(\text{AND})$ , donc  $\text{AND} = \text{NOT}(\text{NAND})$

► **Solution**

## 2.3 Construire OR (Théorème de De Morgan)

En utilisant le théorème de De Morgan :

$$A \text{ OR } B = \text{NOT}(\text{NOT}(A) \text{ AND } \text{NOT}(B))$$

Écrivez OR uniquement avec des NAND.

► **Solution**

## Exercice 3 : Lois de De Morgan

**Objectif :** Appliquer les lois de De Morgan

### Rappel des lois

1.  $\text{NOT}(A \text{ AND } B) = \text{NOT}(A) \text{ OR } \text{NOT}(B)$
2.  $\text{NOT}(A \text{ OR } B) = \text{NOT}(A) \text{ AND } \text{NOT}(B)$

### 3.1 Simplification

Simplifiez l'expression suivante en utilisant De Morgan :

$\text{NOT}(\text{NOT}(A) \text{ AND } \text{NOT}(B))$

► **Solution**

### 3.2 Vérification par table de vérité

Vérifiez que  $\text{NOT}(A \text{ OR } B) = \text{NOT}(A) \text{ AND } \text{NOT}(B)$  :

A	B	A OR B	NOT(A OR B)	NOT(A)	NOT(B)	NOT(A) AND NOT(B)
0	0	?	?	?	?	?
0	1	?	?	?	?	?
1	0	?	?	?	?	?
1	1	?	?	?	?	?

► Solution

## Exercice 4 : Multiplexeur (Mux)

**Objectif :** Comprendre le fonctionnement du Mux

### 4.1 Table de vérité du Mux

Complétez la table de vérité pour `Mux(a, b, sel)` :

a	b	sel	out
0	0	0	?
0	1	0	?
1	0	0	?
1	1	0	?
0	0	1	?
0	1	1	?
1	0	1	?
1	1	1	?

► Solution

## 4.2 Formule du Mux

Exprimez la sortie du Mux en fonction de a, b et sel :

out = ?

**Indice :** "Soit (a ET non-sel) soit (b ET sel)"

► **Solution**

## Exercice 5 : Démultiplexeur (DMux)

**Objectif :** Comprendre le routage avec DMux

### 5.1 Table de vérité du DMux

Le DMux a une entrée **in**, un sélecteur **sel**, et deux sorties **a** et **b**.

Complétez la table :

in	sel	a	b
0	0	?	?
1	0	?	?
0	1	?	?
1	1	?	?

► Solution

## 5.2 Formules du DMux

Exprimez les sorties a et b :

a = ?  
b = ?

► Solution

## Exercice 6 : Fonction Mystère

**Objectif :** Analyser un circuit à partir de sa description

### Énoncé

On a le circuit suivant :

```
w1 = NOT(A)
w2 = NOT(B)
w3 = AND(w1, B)
w4 = AND(A, w2)
out = OR(w3, w4)
```

### 6.1 Complétez la table de vérité

A	B	w1	w2	w3	w4	out
0	0	?	?	?	?	?
0	1	?	?	?	?	?
1	0	?	?	?	?	?
1	1	?	?	?	?	?

### 6.2 Quelle porte connue ce circuit implémente-t-il ?

► Solution

## Exercice 7 : Défi - Porte IMPLIES

**Objectif :** Construire une nouvelle porte

L'implication logique  $A \rightarrow B$  (si A alors B) vaut FAUX uniquement si  $A=1$  et  $B=0$  :

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Construisez cette porte avec NOT et OR.

► **Solution**

## Récapitulatif

### Compétences validées

Après ce TD, vous devez savoir :

- [ ] Remplir une table de vérité pour toute porte
- [ ] Construire NOT, AND, OR à partir de NAND
- [ ] Appliquer les lois de De Morgan
- [ ] Expliquer le fonctionnement du Mux et DMux
- [ ] Analyser un circuit et identifier sa fonction

### Prochaine étape

➡ **TP Chapitre 01** : Implémenter ces portes en HDL dans le simulateur

📖 **Référence** : Livre Seed, Chapitre 01 - Logique Booléenne