

# Chapitre 1 : Circuits logiques combinatoires

BOULAICHE Ammar

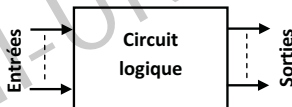
Université de Jijel

Cours structure machine 2 - Première année MI

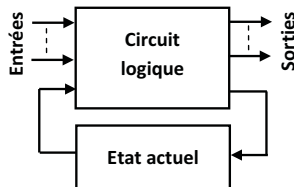
Avril 2021

# Introduction

- Les machines numériques modernes (ordinateurs, tablettes, smartphones, etc.) sont constituées de deux types de circuits : **les circuits combinatoires** et **les circuits séquentiels**.
- Un **circuit logique combinatoire** est un circuit numérique dont les sorties ne dépendent que de l'état logique de ses entrées.
- Un **circuit logique séquentiel** est un circuit numérique dont les sorties dépendent de l'état logique de ses entrées, ainsi que de l'état actuel de ce circuit.



Circuit logique combinatoire



Circuit logique séquentiel

# Circuits logiques combinatoires

- Dans ce chapitre, nous nous intéressons principalement à la synthèse des circuits logiques combinatoires de base (les additionneurs, les décodeurs, les multiplexeurs, etc.), à partir desquels on peut concevoir d'autres circuits plus complexes.
- La synthèse d'un circuit combinatoire consiste tout simplement à réaliser ce circuit à partir de l'énoncé ou d'un cahier des charges décrivant les fonctions ou le rôle que le circuit doit remplir.
- Il s'agit donc de déterminer le logigramme associé aux fonctions logiques constituant le circuit en connaissant la définition de chacune de ces fonctions.

- La synthèse des circuits logiques combinatoires passe généralement par les étapes suivantes :
  - 1 Déterminer les entrées et les sorties du circuit à partir de la description du problème (c'est l'étape la plus importante, il faut bien comprendre l'énoncé du problème afin de déterminer correctement le nombre de variables d'entrée et de variables de sortie du circuit à réaliser).
  - 2 Etablir la table de vérité des différentes sorties en fonction des entrées.
  - 3 Identifier les fonctions logiques des différentes sorties à partir de la table de vérité.
  - 4 Simplifier les fonctions logiques obtenues par la méthode algébrique ou par la méthode de karnaugh.
  - 5 Dessiner le logigramme du circuit.

# Exemple de circuits logiques combinatoires

- Dans un ordinateur, nous pouvons distinguer trois classes différentes de circuits logiques combinatoires.
  - ① Les circuits combinatoires de calcul arithmétiques et logiques, tels que les additionneurs, les soustracteurs, les comparateurs, etc.
  - ② Les circuits combinatoires d'aiguillage et de transmission de données, tels que les codeurs, les décodeurs, les multiplexeurs, les démultiplexeurs, etc.
  - ③ Les circuits combinatoires de codage et de conversion de codes, tels que les transcodeurs, les afficheurs 7 segments, etc.
- Dans la suite de ce chapitre, nous procéderons à la synthèse de quelques circuits de ces trois classes de circuits combinatoires.

# Demi-Additionneur

- **Définition** : Un demi-additionneur est un circuit combinatoire qui permet de réaliser l'addition de deux nombres  $A$  et  $B$  de 1 bit chacun et qui retourne un bit du résultat de l'addition  $S$  et un bit de la retenue  $R$ .

- **Synthèse du circuit** :

- 1 Les entrées et les sorties.



- 2 La table de vérité.

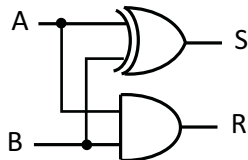
$A$	$B$	$S$	$R$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- 3 Les expressions simplifiées.

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

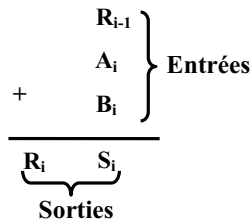
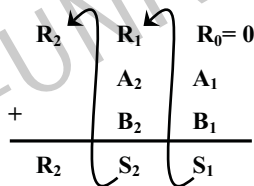
$$R(A, B) = A.B$$

- 4 Le schéma logique.



# Additionneur complet

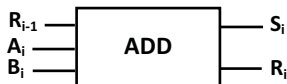
- **Définition** : Un additionneur complet est un circuit combinatoire qui permet de réaliser l'addition de deux bits  $A_i$  et  $B_i$ , plus la retenue entrante  $R_{i-1}$  issue de l'addition de deux bits de rang inférieur, et qui produit un bit du résultat de l'addition  $S_i$  et un bit de la retenue sortante  $R_i$ .
- L'intérêt de la retenue entrante est de permettre le chaînage des additionneurs complets pour pouvoir réaliser des additions de plusieurs bits.



# Additionneur complet

## • Synthèse du circuit :

- ❶ Les entrées et les sorties.



- ❷ La table de vérité.

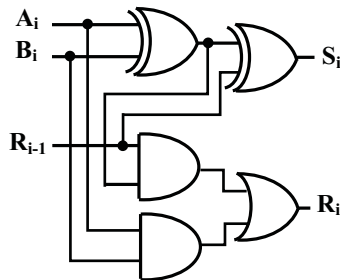
$A_i$	$B_i$	$R_{i-1}$	$S_i$	$R_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- ❸ Les expressions simplifiées.

$$S_i = A_i \oplus B_i \oplus R_{i-1}$$

$$R_i = R_{i-1} \cdot (A_i \oplus B_i) + A_i \cdot B_i$$

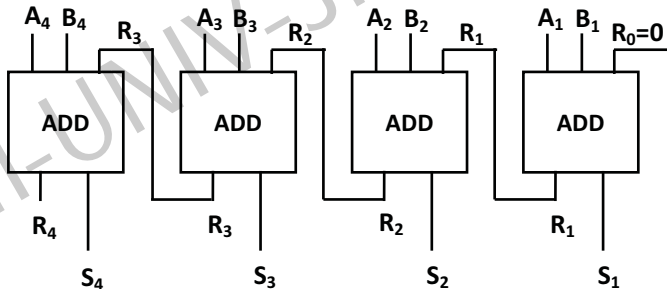
- ❹ Le schéma logique.





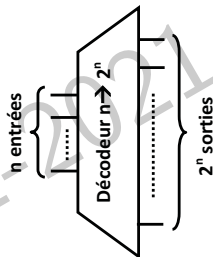
# Additionneur $n$ bits

- Pour réaliser un additionneur  $n$  bits, on met  $n$  additionneurs complets en cascade de telle sorte que la retenue des additionneurs complets de poids le moins significatif se propage vers les additionneurs complets de poids le plus significatif.
- **Exemple :** Pour un additionneur 4 bits, on peut mettre 4 additionneurs complets en cascade comme suite.



# Décodeur binaire

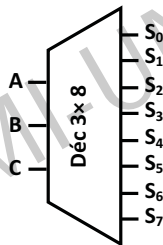
- **Définition** : Un décodeur  $n$  bits est un circuit logique combinatoire de  $n$  entrées et  $2^n$  sorties. Il permet d'activer la ligne de sortie correspondant au code binaire présent en entrée.



- **Synthèse du circuit (décodeur  $3 \times 8$ )**

❶ Les entrées/sorties.

❷ La table de vérité.



A	B	C	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

## ③ Les expressions logiques.

$$S_0 = \overline{A}.\overline{B}.\overline{C}$$

$$S_1 = \overline{A}.\overline{B}.C$$

$$S_2 = \overline{A}.B.\overline{C}$$

$$S_3 = \overline{A}.B.C$$

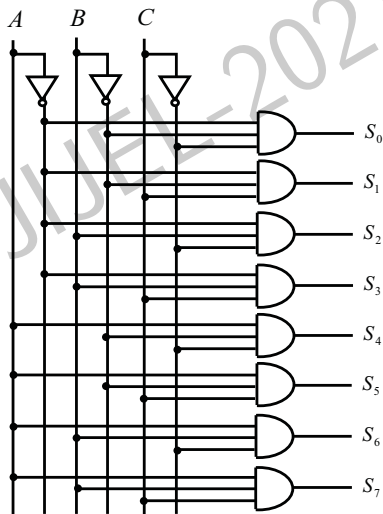
$$S_4 = A.\overline{B}.\overline{C}$$

$$S_5 = A.\overline{B}.C$$

$$S_6 = A.B.\overline{C}$$

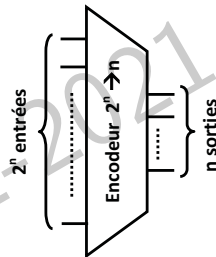
$$S_7 = A.B.C$$

## ④ Le schéma du circuit.



# Encodeur (codeur) binaire

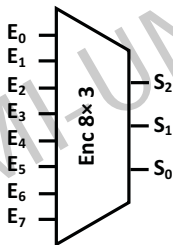
- **Définition** : Un encodeur binaire est un circuit logique combinatoire de  $2^n$  entrées et  $n$  sorties. Il joue le rôle inverse d'un décodeur, c'est à dire si une de ses entrées est active, il retourne son numéro binaire en sortie.



- **Synthèse du circuit (encodeur  $8 \times 3$ )**

① Les entrées/sorties.

② La table de vérité.



$E_0$	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	$E_6$	$E_7$	$S_2$	$S_1$	$S_0$
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

# Encodeur (codeur) binaire

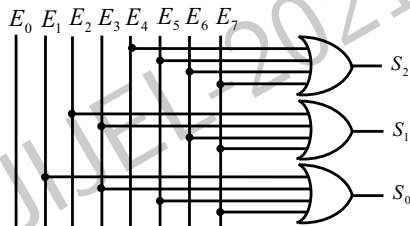
## 3 Les expressions logiques.

$$S_0 = E_1 + E_3 + E_5 + E_7$$

$$S_1 = E_2 + E_3 + E_6 + E_7$$

$$S_2 = E_4 + E_5 + E_6 + E_7$$

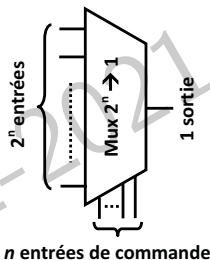
## 4 Le schéma du circuit.



- En plus de leurs entrées et leurs sorties standards, les décodeurs et les encodeurs peuvent être dotés d'une entrée de validation (V) qui commande leur fonctionnement. Dans ce cas là, le décodeur ou l'encodeur ne sera fonctionnel que lorsque son entrée de validation sera active (V=1).

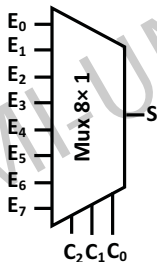
# Multiplexeur

- **Définition** : Un multiplexeur est un circuit logique combinatoire qui comporte  $2^n$  entrées,  $n$  entrées de commande et une seule sortie. Il permet d'aiguiller la valeur de la ligne d'entrée indiquée dans ses entrées de commande vers la ligne de sortie.
- **Synthèse du circuit (multiplexeur  $8 \times 1$ )**



① Les entrées/sorties.

② La table de vérité.



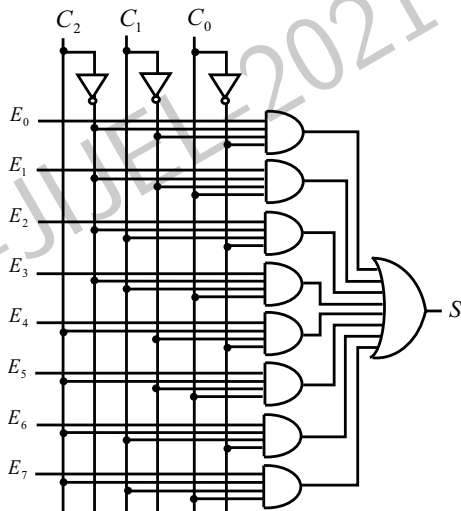
$C_2$	$C_1$	$C_0$	$S$
0	0	0	$E_0$
0	0	1	$E_1$
0	1	0	$E_2$
0	1	1	$E_3$
1	0	0	$E_4$
1	0	1	$E_5$
1	1	0	$E_6$
1	1	1	$E_7$

# Multiplexeur

## 3 Les fonctions logiques.

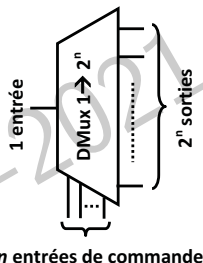
$$\begin{aligned} S = & \overline{C_2}.\overline{C_1}.\overline{C_0}.E_0 + \\ & \overline{C_2}.\overline{C_1}.C_0.E_1 + \\ & \overline{C_2}.C_1.\overline{C_0}.E_2 + \\ & \overline{C_2}.C_1.C_0.E_3 + \\ & C_2.\overline{C_1}.\overline{C_0}.E_4 + \\ & C_2.\overline{C_1}.C_0.E_5 + \\ & C_2.C_1.\overline{C_0}.E_6 + \\ & C_2.C_1.C_0.E_7 \end{aligned}$$

## 4 Le schéma du circuit.



# Démultiplexeur

- **Définition** : Un démultiplexeur est un circuit logique combinatoire qui comporte une seule entrée,  $n$  entrées de commande et  $2^n$  sorties. Il permet d'aiguiller la valeur de la ligne d'entrée vers la ligne de sortie indiquée dans ses entrées de commande.



- **Synthèse du circuit (démultiplexeur  $1 \times 8$ )**

① Les entrées/sorties.



② La table de vérité.

$C_2$	$C_1$	$C_0$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$
0	0	0	E	0	0	0	0	0	0	0
0	0	1	0	E	0	0	0	0	0	0
0	1	0	0	0	E	0	0	0	0	0
0	1	1	0	0	0	E	0	0	0	0
1	0	0	0	0	0	0	E	0	0	0
1	0	1	0	0	0	0	0	E	0	0
1	1	0	0	0	0	0	0	0	E	0
1	1	1	0	0	0	0	0	0	0	E



# Démultiplexeur

## 3 Les fonctions logiques.

$$S_0 = \overline{C_2} \cdot \overline{C_1} \cdot \overline{C_0} \cdot E$$

$$S_1 = \overline{C_2} \cdot \overline{C_1} \cdot C_0 \cdot E$$

$$S_2 = \overline{C_2} \cdot C_1 \cdot \overline{C_0} \cdot E$$

$$S_3 = \overline{C_2} \cdot C_1 \cdot C_0 \cdot E$$

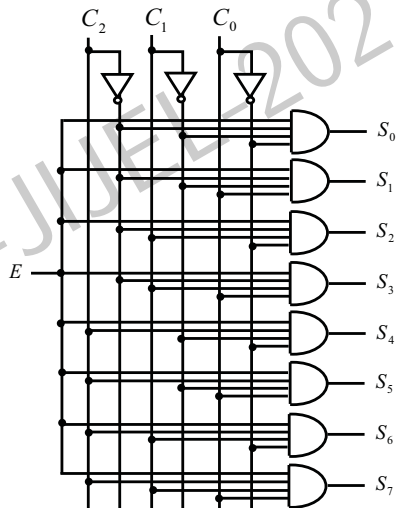
$$S_4 = C_2 \cdot \overline{C_1} \cdot \overline{C_0} \cdot E$$

$$S_5 = C_2 \cdot \overline{C_1} \cdot C_0 \cdot E$$

$$S_6 = C_2 \cdot C_1 \cdot \overline{C_0} \cdot E$$

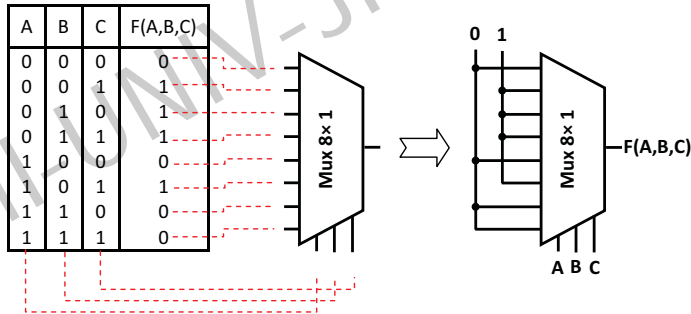
$$S_7 = C_2 \cdot C_1 \cdot C_0 \cdot E$$

## 4 Le schéma du circuit.



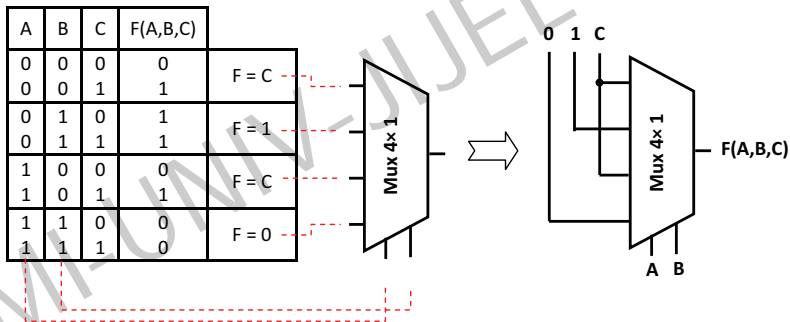
# Fonctions logiques via des multiplexeurs

- On peut toujours générer des fonctions logiques quelconques en utilisant des multiplexeurs et des portes logiques de base. Il suffit de relier les variables de la fonction à générer aux différentes entrées du multiplexeur (entrées standards et entrées de commandes).
- Exemple 1** : réaliser la fonction  $F(A, B, C) = \bar{A}.B + \bar{B}.C$  en utilisant un multiplexeur  $8 \times 1$ .



# Fonctions logiques via des multiplexeurs

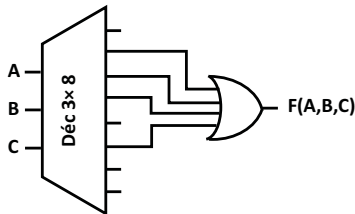
- **Exemple 2** : réaliser la fonction  $F(A, B, C) = \bar{A}.B + \bar{B}.C$  en utilisant un multiplexeur  $4 \times 1$ .



# Fonctions logiques via des décodeurs

- On peut également générer des fonctions logiques quelconques en utilisant des décodeurs et des portes logiques de base. Il suffit de relier les variables de la fonction à générer aux entrées du décodeur et les sorties correspondant aux différents mintermes de la fonction aux entrées d'une ou plusieurs portes logiques de base.
- Exemple 1 :** réaliser la fonction  $F(A, B, C) = \bar{A}.B + \bar{B}.C$  en utilisant un décodeur  $3 \times 8$ .

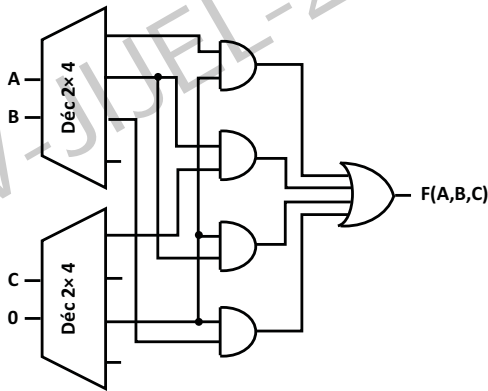
A	B	C	F(A,B,C)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0



# Fonctions logiques via des décodeurs

- **Exemple 2** : réaliser la fonction  $F(A, B, C) = \bar{A}.B + \bar{B}.C$  en utilisant deux décodeurs  $2 \times 4$ .

A	B	C	F(A,B,C)
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





Wladimir Mercouroff.

*Architecture matérielle et logicielle des ordinateurs et des microprocesseurs.*

*Armand Colin, 1990. ISBN : 2-200-42007-2.*

**Bibliothèque centrale, Université de Jijel, Cote :002/07.**



Joel Ristori, Lucien Ungaro.

*Cours d'architecture des ordinateurs.*

*Eyrolles, 1991.*

**Bibliothèque centrale, Université de Jijel, Cote :002/17.**



Pierre-Alain Goupille.

*Technologie des ordinateurs pour les I.U.T et B.T.S informatique avec exercices.*

*Masson, 1993.*

**Bibliothèque centrale, Université de Jijel, Cote :002/01.**



André Poinot.

*Problèmes d'électronique logique.*

Masson , 1994. ISSN : 2225844518.

Bibliothèque centrale, Université de Jijel, Cote :621/270.



Paolo Zanella, Yves Ligier.

*Architecture et technologie des ordinateurs.*

Dunod, 1998. ISBN : 210003801X.

Bibliothèque centrale, Université de Jijel, Cote :004/258.



Habiba Drias-Zerkaoui.

*Introduction à l'architecture des ordinateurs.*

Office des publications universitaires, 2006.

Bibliothèque centrale, Université de Jijel, Cote :002/08.