

# Chapitre 4: Circuits combinatoires

Présenter par: Mme AMGHAR D

# Circuits combinatoires

1. Introduction
2. Addition binaire
3. Soustraction
4. Comparaison
5. Décodage
6. Multiplexage
7. Encodage

# Introduction

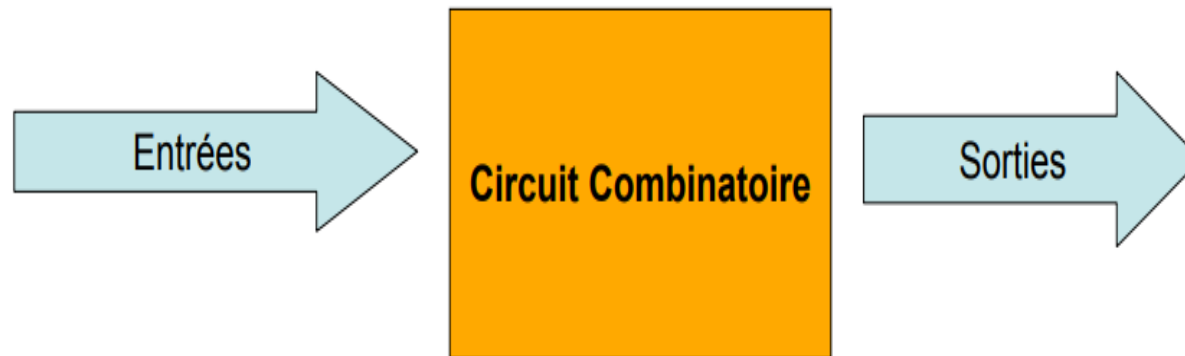
- ▶ Les circuits logiques sont élaborés à partir de composants électroniques - **transistors** .
- ▶ Types de circuits logiques:
  - ▶ **Combinatoires**
  - ▶ **Séquentiels**

# Introduction

- ▶ **Circuits logiques séquentiels** : c'est des circuits dont les valeurs de sortie dépendent d'entrée appliquées ultérieurement.
- ▶ **Circuits logiques combinatoires** : c'est des circuits dont les valeurs de sortie ne dépendent que de ses valeurs d'entrée.

# Circuit combinatoire

- ▶ Un circuit combinatoire est constitué d'éléments logiques élémentaires appelés portes logiques(logic gates), elle reçoivent des signaux appliqués en entrée et produisent des signaux en sortie.



# Circuit combinatoire

L'étude des circuits combinatoires se résume en deux questions :

- **Synthèse** : réaliser le circuit combinatoire à partir de l'énoncé décrivant les fonctions ou le rôle du circuit, en question.
- **Analyse** : déterminer le rôle du circuit combinatoire à partir de son logigramme.

# Circuit combinatoire

les étapes à suivre pour réaliser la synthèse d'un circuit logique combinatoire :

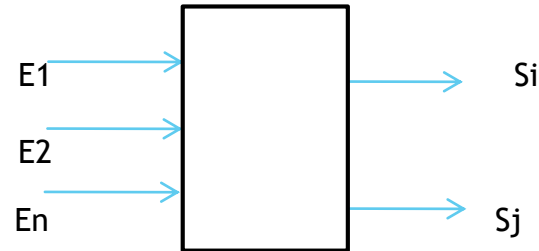
1. Établir la table de vérité de chacune des fonctions impliquées dans le problème à traiter
2. Établir les équations logiques.
3. Simplifier les équations de chacune des fonctions logiques.
4. Établir le logigramme du circuit logique.

# Circuits combinatoires

- ▶ Un circuit combinatoire est un circuit numérique dont les sorties dépendent uniquement des entrées.

$$S_i = F(E_i)$$

$$S_i = F(E_1, E_2, \dots, E_n)$$





# Circuits combinatoires

Parmi les principaux circuits combinatoires, on distingue:

- les circuits d'opérations arithmétiques (addition, soustraction)
- Les circuits logiques (décodage, multiplexage, comparaison).

*Remarque:* Il est possible d'utiliser des circuits combinatoires pour réaliser d'autres circuits plus complexes.

# Additionneurs(Adder):

## ► Additionneurs :

- Demi additionneur : 2 entrées sur 1 bit,  
2 sorties sur 1 bits.
- Additionneur complet : 3 entrées sur 1 bit,  
2 sorties sur 1 bits.
- Additionneur sur n bits.

# Additionneurs(Adder):

## Demi-additionneur (half adder): :

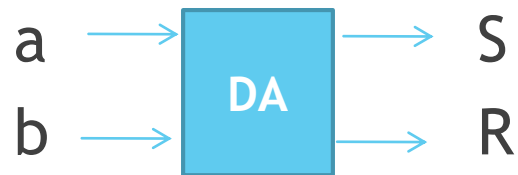
- On commence par l'addition de 2 bits  $a$  et  $b$  en entrée ,avec en sortie la **somme  $S$**  et une retenue  **$R$** .
- On l'appelle demi additionneur , parce qu'il **ne tient pas compte** de la **retenue** qui peut provenir des calculs précédents.

<i>Entrées</i>		<i>Sorties</i>	
<i>A</i>	<i>B</i>	<i>Somme (S)</i>	<i>Retenue (R<sub>sor</sub>)</i>
0 + 0		0	0
0 + 1		1	0
1 + 0		1	0
1 + 1		0	1

# Additionneurs(Adder):

Demi-additionneur (half adder): :

<i>Entrées</i>		<i>Sorties</i>	
<i>A</i>	<i>B</i>	<i>Somme (S)</i>	<i>Retenue (R<sub>sor</sub>)</i>
0 + 0		0	0
0 + 1		1	0
1 + 0		1	0
1 + 1		0	1



$$S = a \oplus b$$

$$R_{sor} = a.b$$

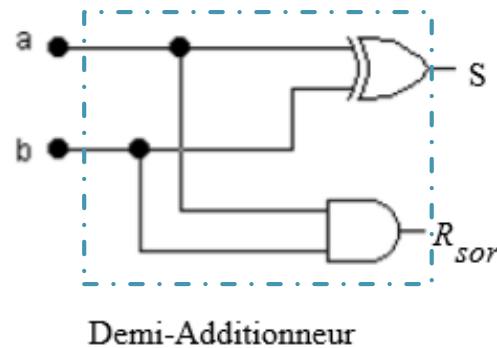
# Additionneurs(Adder):

## Demi additionneur (half adder):

- Expressions logiques:

$$S = A \oplus B$$
$$R_{sor} = A.B$$

- Logigramme :



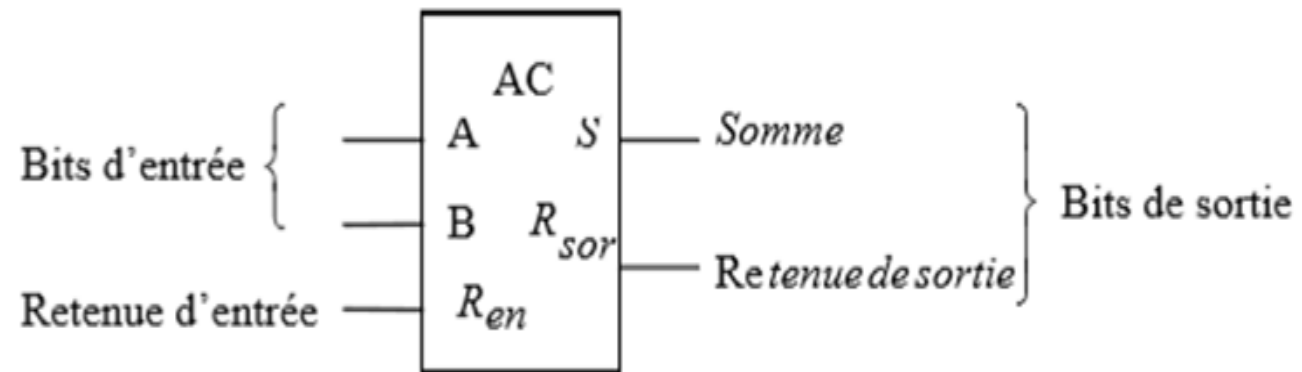
# Additionneurs(Adder):

## L'additionneur complet (full adder):

- ▶ il faut tenir compte de la retenue provenant des bits de poids inférieurs
- ▶ donc l'additionneur complet :
- ▶ 3 entrées :  $a$ ,  $b$  et  $r$
- ▶ 2 sorties  $S$  et  $R$

# Additionneurs(Adder):

**additionneur Complet (full adder) (AC):**



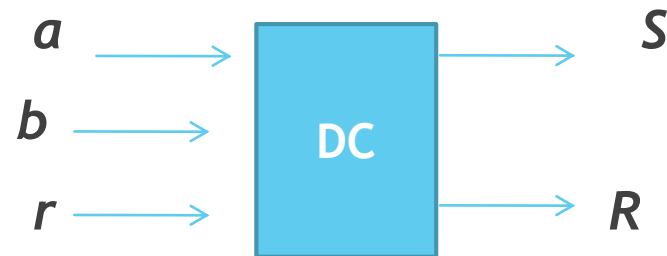
Symbole logique d'un AC

# Additionneurs(Adder):

L'additionneur complet(full adder):

TV de l'additionneur complet :

a	b	r	R	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



$$R = \bar{a}.b.r + a.\bar{b}.r + a.b.\bar{r} + a.b.r$$

$$R = a.b + r(a \oplus b) \quad (\text{après simplification})$$

$$S = \bar{a}.\bar{b}.r + \bar{a}.b.\bar{r} + a.\bar{b}.\bar{r} + a.b.r$$

$$S = (a \oplus b) \oplus r$$



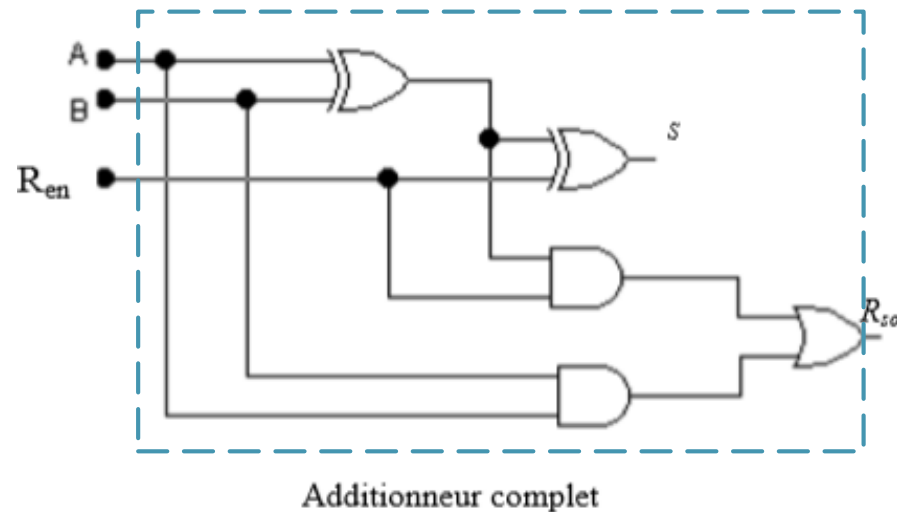
# Additionneurs(Adder):

Additionneur complet (full adder): :

Expressions logiques :

$$S = (A \oplus B) \oplus R_{en}$$
$$R_{sor} = A.B + (A \oplus B).R_{en}$$

► Logigramme :



# Additionneurs(Adder):

Additionneur complet(full adder) :

► Logigramme avec 2 demi-additionneurs :

DA

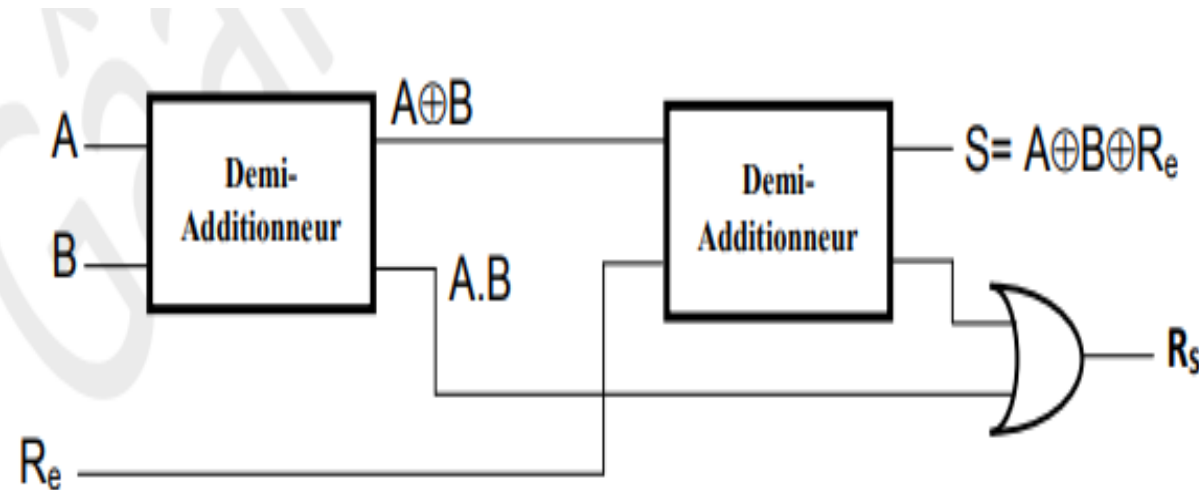
$$S = a \oplus b$$

$$R_{sor} = a.b$$

CA

$$S = (a \oplus b) \oplus r$$

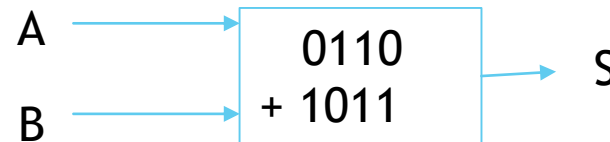
$$R = a.b + r(a \oplus b).$$



# Additionneurs(Adder):

## Additionneur parallèle:

- ▶ Pour additionner 2 nombres
- ▶ Il consiste à mettre des additionneurs **1 bit** en série, avec la retenue sortante de l'un qui devient entrante du suivant, ce qui correspond à la propagation de retenue.



# Additionneurs(Adder):

**Additionneur parallèle:**

**Exemple :** additionneur parallèle à 4 bits.

Soit à additionner les nombres binaires :

$$A = a_0a_1a_2a_3$$

$$B = b_0b_1b_2b_3$$

Additionneur 4 bits:

