

# Module:

## Architecture des Ordinateurs

### Chapitre V: Les circuits combinatoires

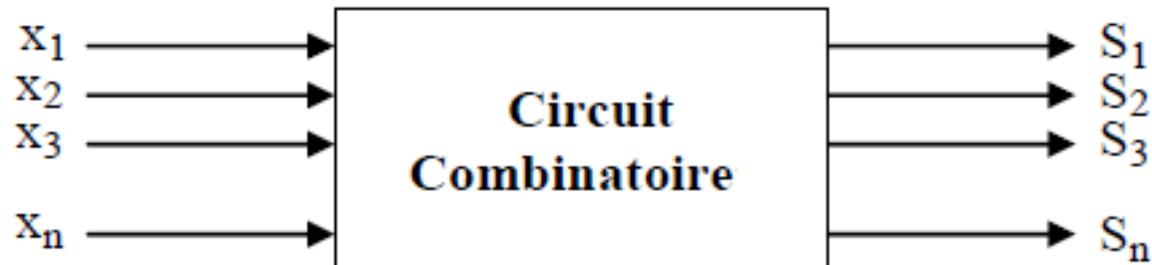
Pr. AMROUNE Med

Chapitre 4

# Les circuits combinatoires

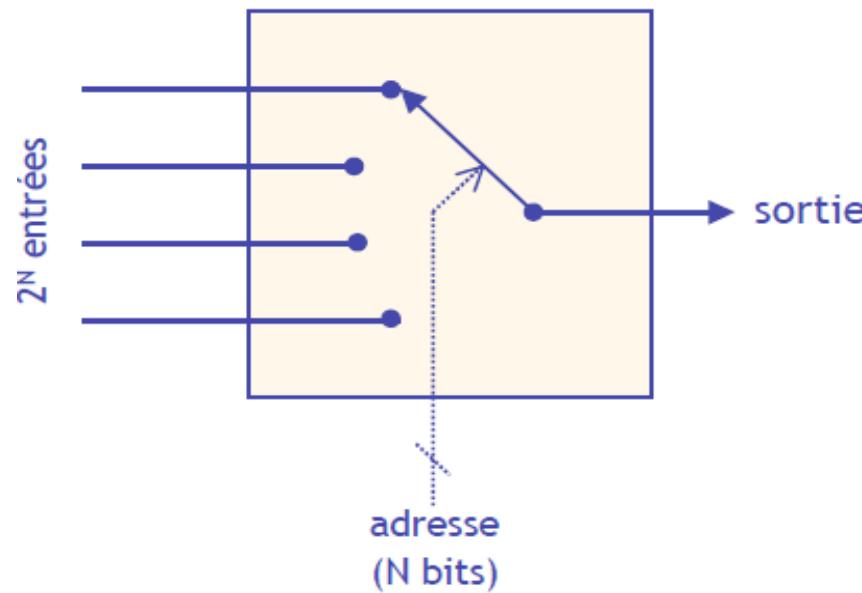
## 1. Définition

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



## Multiplexeur

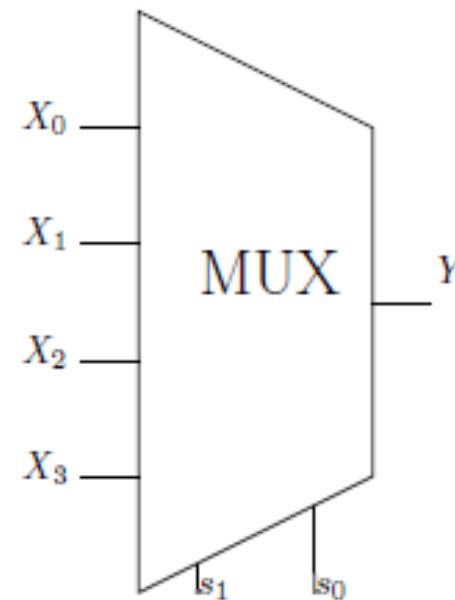
- Un Multiplexeur est un circuit combinatoire qui dispose de  $2^n$  entrées et une sortie. Ce circuit permet d'aiguiller en sortie l'une des entrées à l'aide des bits d'adresse (de sélection)



# Multiplexeur

## Exemple

- Un multiplexeur 4 vers 1 avec 2 bits de sélection. En notant  $X_0, X_1, X_2, X_3$  les bits d'entrées,  $Y$  le bit de sortie et  $s_0, s_1$  les bits de sélection.



# Multiplexeur

## ☐ Table de vérité

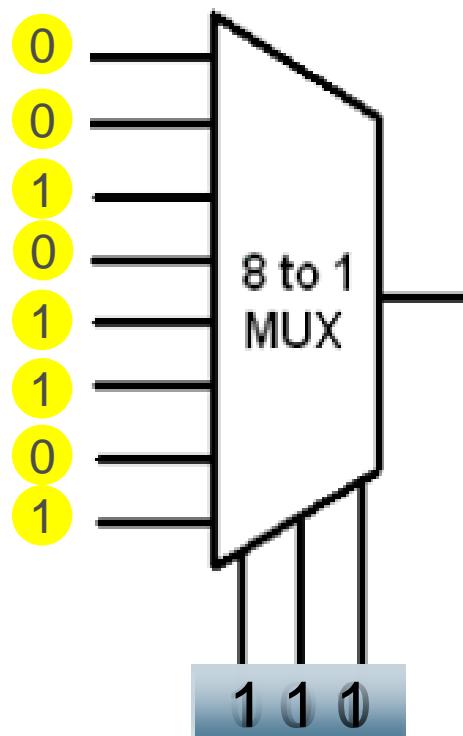
<i>s1</i>	<i>s0</i>	<i>Y</i>
0	0	$X_0$
0	1	$X_1$
1	0	$X_2$
1	1	$X_3$

$$Y = \bar{S}_1 \bar{S}_0 X_0 + \bar{S}_1 S_0 X_1 + S_1 \bar{S}_0 X_2 + S_1 S_0 X_3$$

# Multiplexeur

## ❑ Applications des multiplexeurs

- **Conversion parallèle/série:** aiguiller les informations présentes en parallèle à l'entrée du MUX en des informations de type série en sortie; toutes les combinaisons d'adresses sont énumérées une par une sur les entrées de sélection.



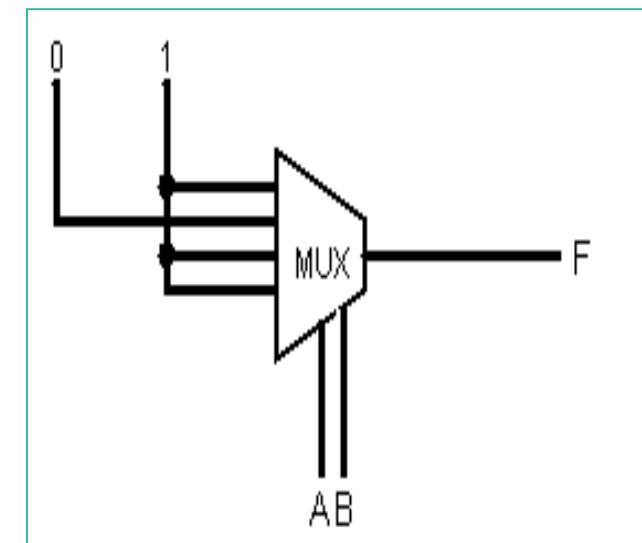
# Multiplexeur

## ❑ Applications des multiplexeurs

- Réalisation de fonctions logiques: toute fonction logique de  $N$  variables est réalisable avec un multiplexeur de  $2^N$  vers 1

Exemple 1:

A	B	F
0	0	1
0	1	0
1	0	1
1	1	1



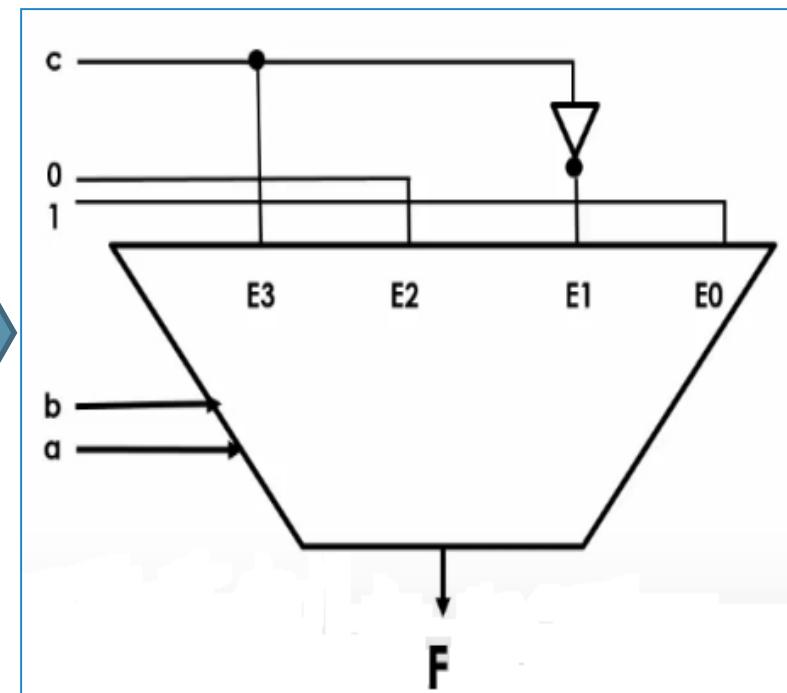
# Multiplexeur

## Exemple 2:

Réaliser la fonction:  $F(a, b, c) = \sum(0, 1, 2, 7)$  à l'aide de:

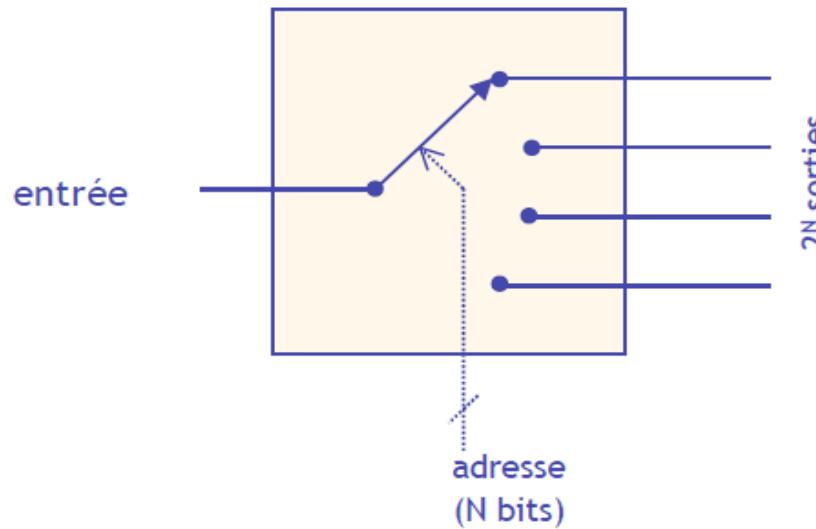
- MUX 4 vers 1

$a$	$b$	$c$	$F$	$E$
0	0	0	1	
0	0	1	1	
0	1	0	1	$E0 = 1$
0	1	1	0	
1	0	0	0	$E1 = \bar{C}$
1	0	1	0	
1	1	0	0	$E2 = 0$
1	1	1	1	$E3 = C$



## Démultiplexeur

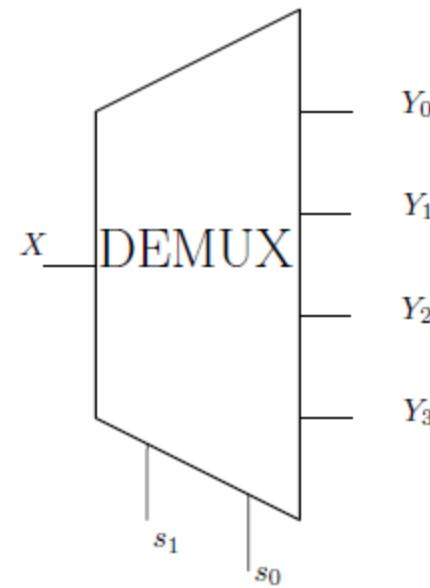
□ Le démultiplexeur réalise l'inverse d'un MUX : il aiguille une seule entrée vers une parmi  $2^n$  sorties. Les démultiplexeur fonctionnent comme un commutateur. Ils comportent une entrée de donné X, n entrées de sélection ( $S_0, S_1, \dots, S_n$ ) et  $2^n$  sorties ( $Y_0, Y_1, \dots, Y_m$ )



# Démultiplexeur

## Exemple

- Démultiplexeur a 2 bits de sélection et donc  $2^2 = 4$  sorties. En fonction de la valeur des bits de sélection, l'entrée est dirigée vers l'une ou l'autre des sorties.



# Démultiplexeur

## ☐ Table de vérité

$S1$	$S0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	$X$
0	1	0	0	$X$	0
1	0	0	$X$	0	0
1	1	$X$	0	0	0

$$Y_0 = \bar{S}_1 \bar{S}_0 X$$

$$Y_1 = \bar{S}_1 S_0 X$$

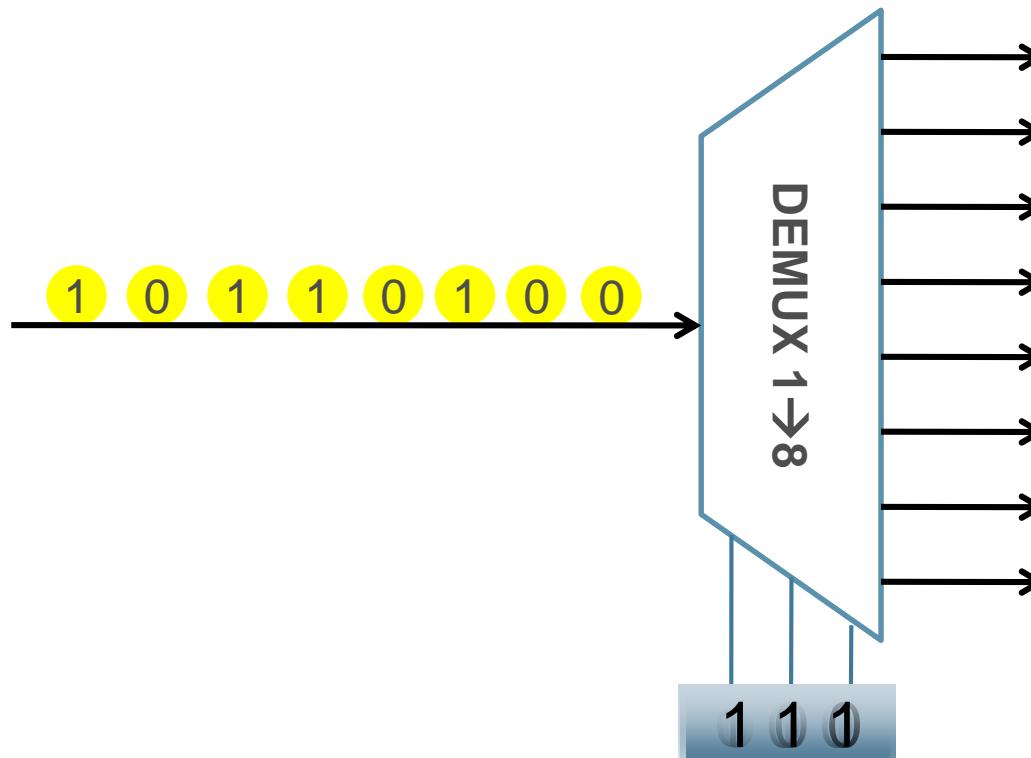
$$Y_2 = S_1 \bar{S}_0 X$$

$$Y_3 = S_1 S_0 X$$

# Démultiplexeur

## ❑ Applications d'un démultiplexeur

- **Conversion série/parallèle:** aiguiller les informations présentes en série à l'entrée du DEMUX en des informations de type parallèle en sortie; toutes les combinaisons d'adresses sont énumérées une par une sur les entrées de sélection.



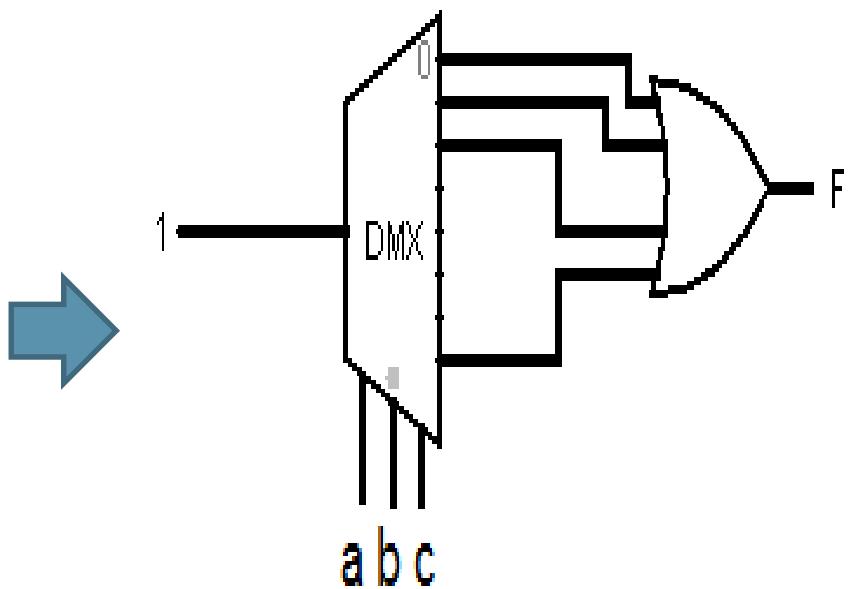
# Démultiplexeur

## ❑ Applications des multiplexeurs

- Réalisation de fonctions logiques: toute fonction logique de N variables est réalisable avec un démultiplexeur de 1 vers  $2^N$

Exemple:

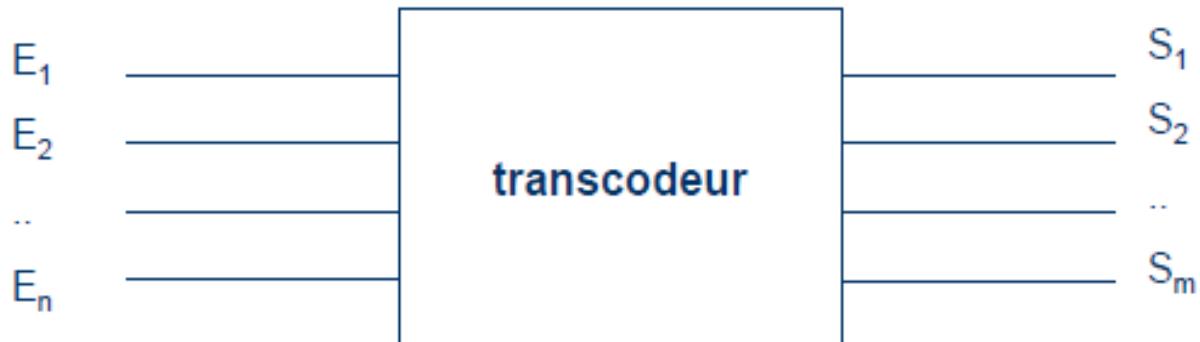
<i>a</i>	<i>b</i>	<i>c</i>	<i>F</i>	<i>Si</i>
0	0	0	1	$S_0=1$
0	0	1	1	$S_1=1$
0	1	0	1	$S_2=1$
0	1	1	0	$S_3=1$
1	0	0	0	$S_4=1$
1	0	1	0	$S_5=1$
1	1	0	0	$S_6=1$
1	1	1	1	$S_7=1$



$$F = S_0 + S_1 + S_2 + S_7$$

## Transcodeur

- C'est un circuit combinatoire qui permet de transformer un code X ( sur n bit) en entrée en un code Y ( sur m bit) en sortie.



# Transcodeur

## Exemple

- BCD → code à excès de 3

<b><i>E3</i></b>	<b><i>E2</i></b>	<b><i>E1</i></b>	<b><i>E0</i></b>		<b><i>S3</i></b>	<b><i>S2</i></b>	<b><i>S1</i></b>	<b><i>S0</i></b>
0	0	0	0	0	0	1	1	1
0	0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	1	1
0	0	1	1	0	1	1	1	0
0	1	0	0	0	1	1	1	1
0	1	0	1	1	0	0	0	0
0	1	1	0	1	0	0	0	1
0	1	1	1	1	0	1	0	0
1	0	0	0	1	0	1	1	1
1	0	0	1	1	1	1	0	0
1	0	1	0	X	X	X	X	X
1	0	1	1	X	X	X	X	X
1	1	0	0	X	X	X	X	X
1	1	0	1	X	X	X	X	X
1	1	1	0	X	X	X	X	X
1	1	1	1	X	X	X	X	X

# Transcodeur

## □ La fonction S<sub>3</sub>:

e1 e0 e3 e2	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

$$S_3 = E_3 + E_2 E_1 + E_2 E_0$$

# Transcodeur

## □ La fonction S2:

e1 e0	00	01	11	10
e3 e2	00	1	1	1
00	0	1	1	1
01	1	0	0	0
11	X	X	X	X
10	0	1	X	X

A Karnaugh map for the function S2. The columns represent inputs e1 and e0, and the rows represent inputs e3 and e2. The minterms are labeled as follows: 00 (top-left), 01 (top-middle), 11 (bottom-middle), and 10 (bottom-right). The map shows the following values: 00 is 0, 01 is 1, 11 is 1, and 10 is 0. The other four cells are marked with X. The map is overlaid with colored lines representing the prime implicants: a red vertical line for m00, a yellow horizontal line for m01, a blue diagonal line for m11, and a blue horizontal line for m10.

$$S_2 = \bar{E}_2 E_1 + \bar{E}_2 E_0 + E_2 \bar{E}_1 \bar{E}_0$$

# Transcodeur

## □ La fonction S1:

e1 e0 e3 e2	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	x	x	x	x
10	1	0	x	x

$$S_1 = \bar{E}_1 \bar{E}_0 + E_1 E_0$$

# Transcodeur

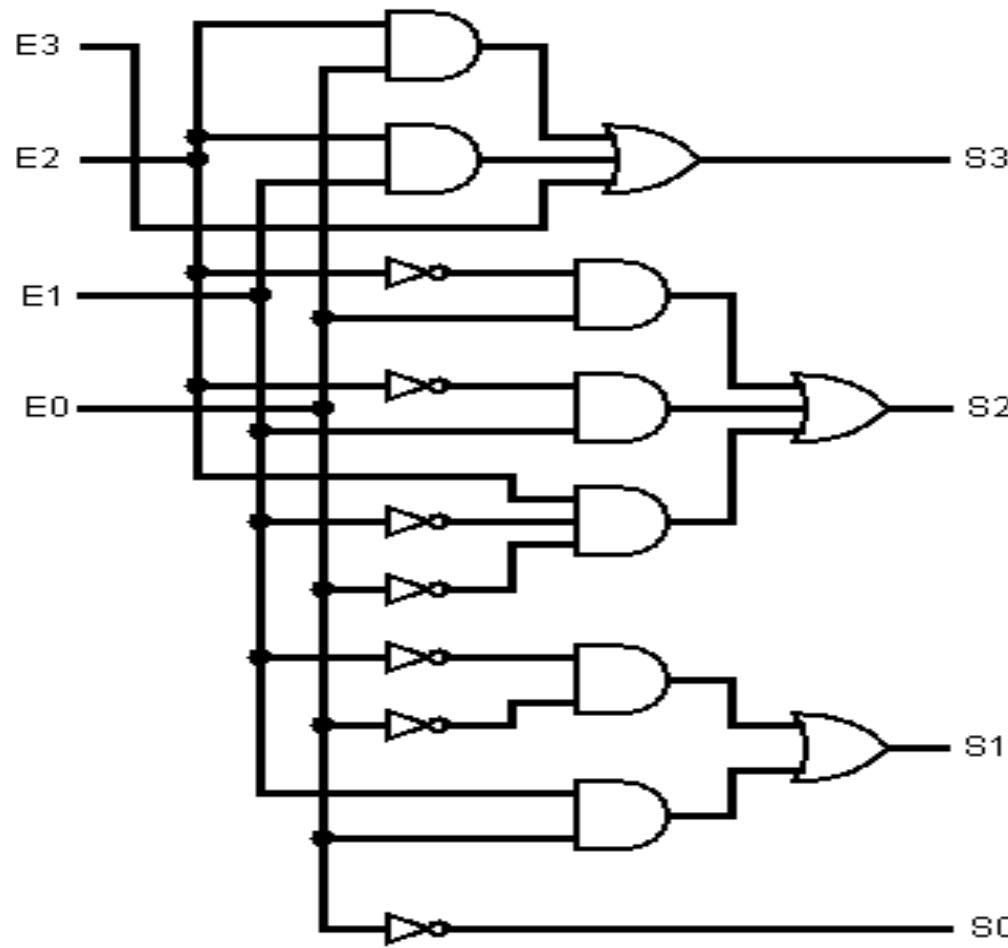
## □ La fonction S0:

e1 e0	00	01	11	10	
e3 e2	00	1	0	0	1
00	1	0	0	1	
01	1	0	0	1	
11	X	X	X	X	
10	1	0	X	X	

$$S_0 = \bar{E}_0$$

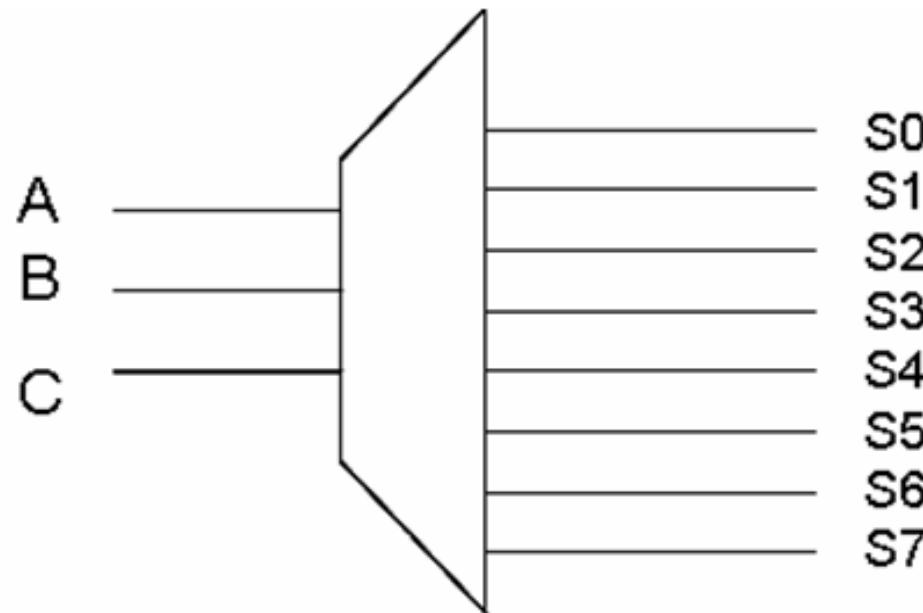
## Transcodeur

### □ Logigramme:



## Décodeur

- Le décodeur est un circuit combinatoire à n entrées et  $2^n$  sorties.
- Pour une configuration binaire en entrée, une seule sortie est active parmi  $2^n$ .



# Décodeur

## □ Table de vérité

A	B	C	S0	S2	S2	S3	S4	S5	S6	S7
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

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

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

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

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

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

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

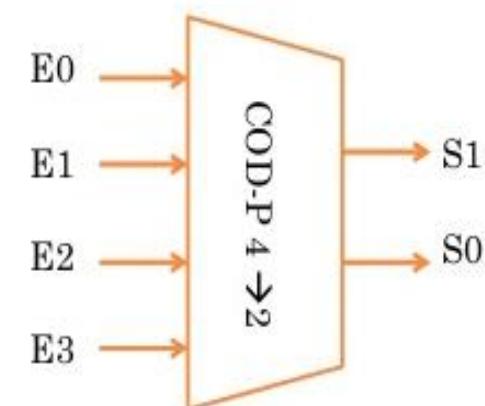
$$S_6 = A \cdot B \cdot \overline{C}$$

$$S_7 = A \cdot B \cdot C$$

## L'encodeur de priorité

- L'encodeur de priorité est un circuit qui détecte la position du premier bit 1 d'un mot en commençant par le bit le plus significatif (le plus à gauche).
- Exemple:  
Encodeur prioritaire (4 entrées vers 2 sorties)

$E_3$	$E_2$	$E_1$	$E_0$		$S_1$	$S_0$
1	X	X	X		1	1
0	1	X	X		1	0
0	0	1	X		0	1
0	0	0	1		0	0

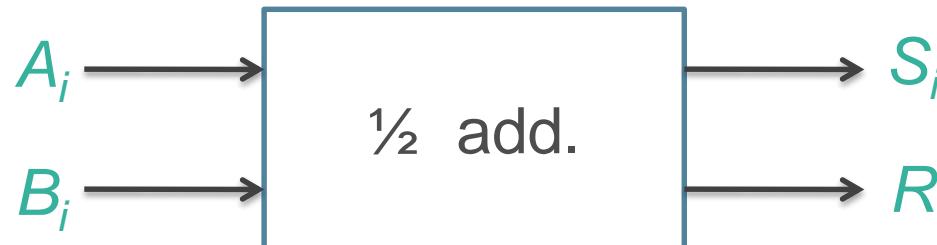


$$S_1 = E_3 + E_2$$

$$S_0 = E_3 + \bar{E}_2 E_1$$

## Demi additionneur

- Le demi additionneur est un circuit combinatoire qui permet de réaliser la **somme arithmétique** de deux bits  $A_i$  et  $B_i$  et délivre en sortie leur somme  $S_i$  et la retenue  $R$ .



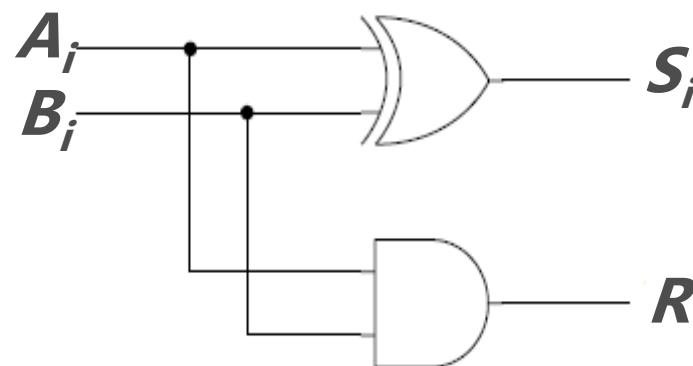
## Demi additionneur

- Table de vérité

<b><i>A<sub>i</sub></i></b>	<b><i>B<sub>i</sub></i></b>	<b><i>S<sub>i</sub></i></b>	<b><i>R</i></b>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

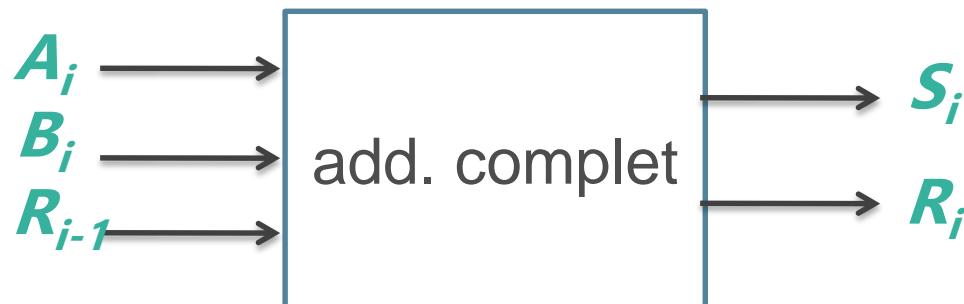
$$S_i = A_i \oplus B_i$$
$$R = A_i \cdot B_i$$

- Circuit logique (logigramme)



## L'additionneur complet

- En binaire, lorsqu'on fait une addition, il faut tenir en compte la retenue entrante .
- L'additionneur complet possède trois entrées:  $A_i$ ,  $B_i$  et  $R_{i-1}$  et deux Sorties  $S_i$  et  $R_i$



# L'additionneur complet

## Table de vérité

<b><i>Ai</i></b>	<b><i>Bi</i></b>	<b><i>Ri-</i></b>	<b><i>Si</i></b>	<b><i>Ri</i></b>
0	0	1	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

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

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

$$S_i = \overline{A}_i \cdot \overline{B}_i \cdot R_{i-1} + \overline{A}_i \cdot B_i \cdot \overline{R}_{i-1} + A_i \cdot \overline{B}_i \cdot \overline{R}_{i-1} + A_i \cdot B_i \cdot R_{i-1}$$

$$S_i = \overline{A}_i \cdot (\overline{B}_i \cdot R_{i-1} + B_i \cdot \overline{R}_{i-1}) + A_i \cdot (\overline{B}_i \cdot \overline{R}_{i-1} + B_i \cdot R_{i-1})$$

$$S_i = \overline{A}_i (B_i \oplus R_{i-1}) + A_i (\overline{B}_i \oplus \overline{R}_{i-1})$$

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

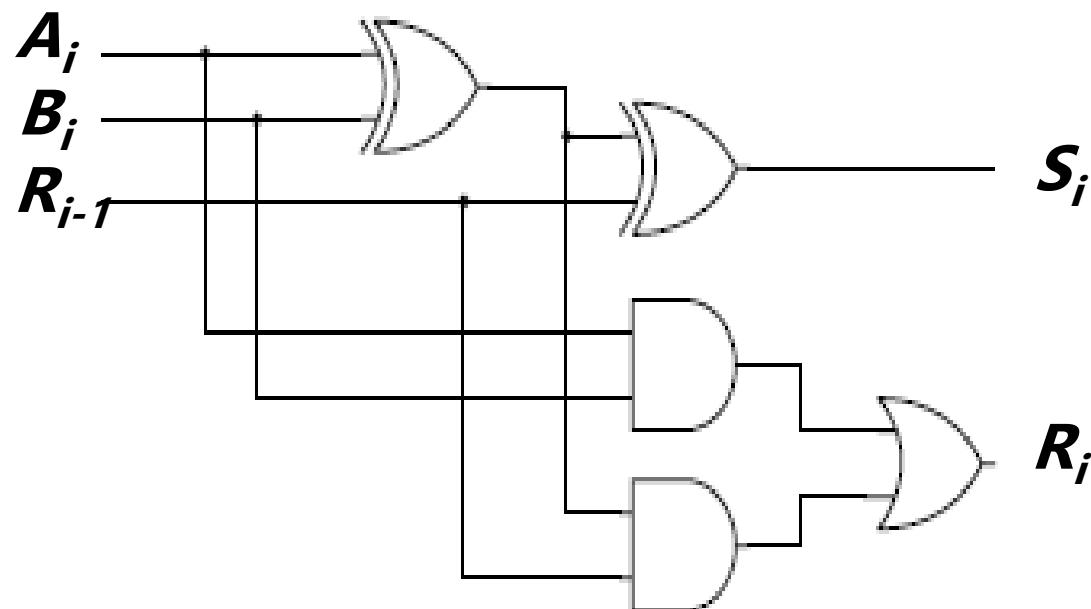
$$R_i = \overline{A}_i B_i R_{i-1} + A_i \overline{B}_i R_{i-1} + A_i B_i \overline{R}_{i-1} + A_i B_i R_{i-1}$$

$$R_i = A_i B_i + R_{i-1} \cdot (\overline{A}_i B_i + A_i \overline{B}_i)$$

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

## L'additionneur complet

### □ Circuit logique (logigramme)



## → Additionneur complet (02 Demi-Additionneurs)

### □ Demi additionneur

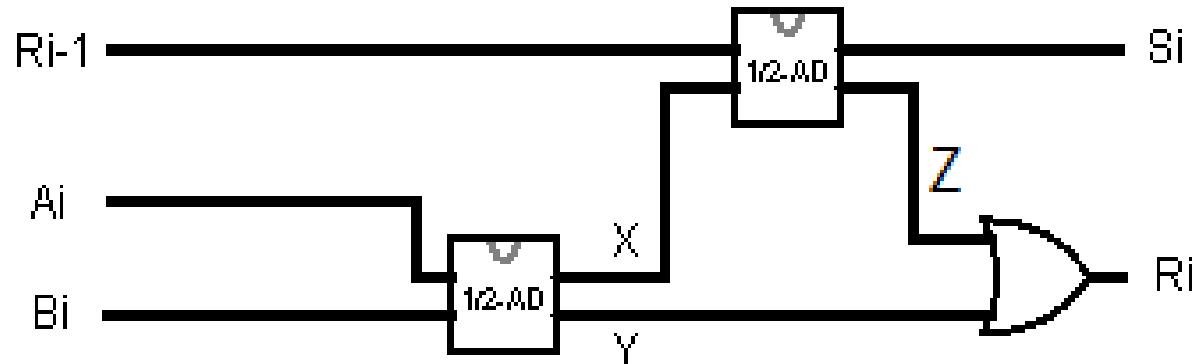
$$S_i = A_i \oplus B_i$$

$$R_i = A_i \cdot B_i$$

### □ Additionneur Complet

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

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

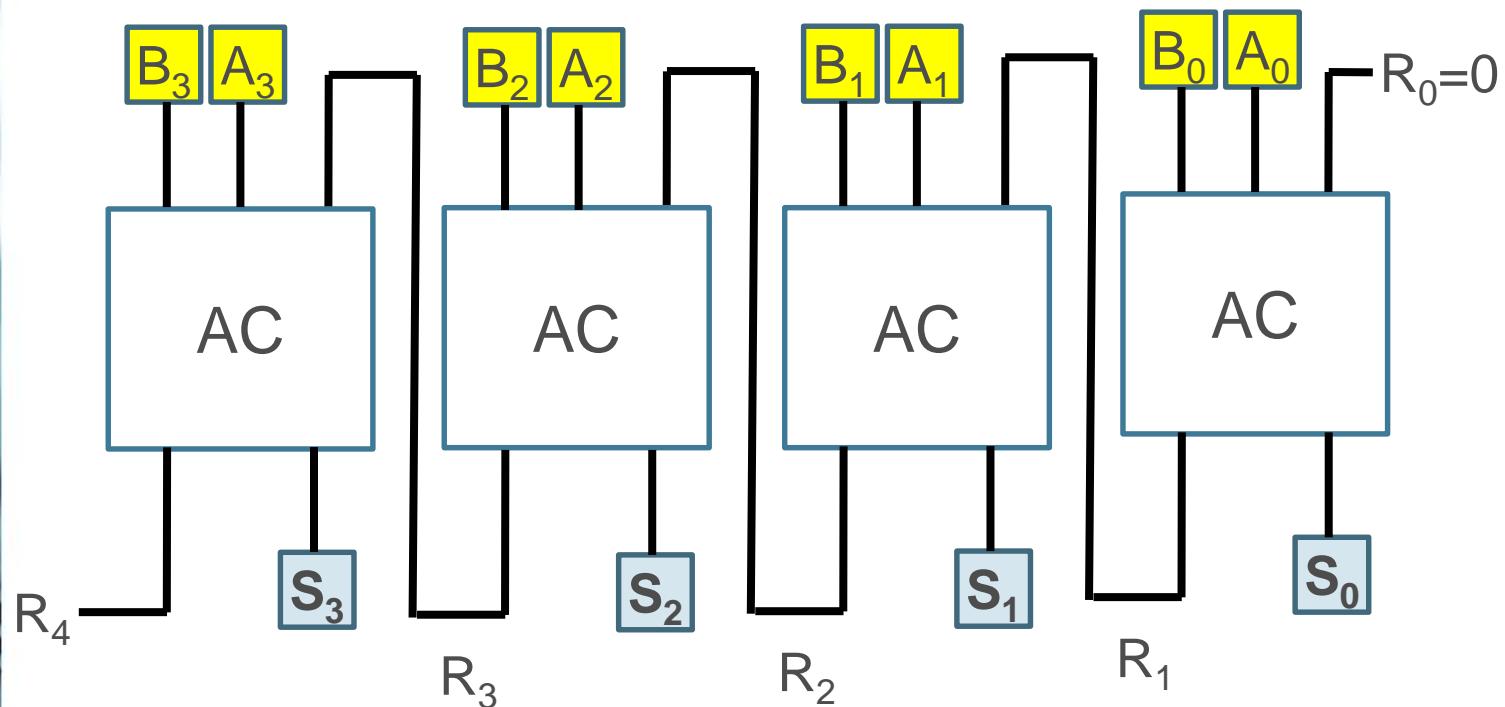


$$X = A_i \oplus B_i \rightarrow S_i = A_i \oplus B_i \oplus R_{i-1}$$

$$\begin{cases} Y = A_i \cdot B_i \\ Z = X \cdot R_{i-1} = (A_i \oplus B_i) \cdot R_{i-1} \end{cases} \rightarrow R_i = A_i B_i + R_{i-1} \cdot (A_i \oplus B_i)$$

## Additionneur 4 bits

### Schéma



# ➤ Additionneur 4 bits

## □ Schéma

