



# Chapitre 4

## Cours:

## Systemes Logiques et Architecture des Ordinateurs

Dr. Safa Téboulbi

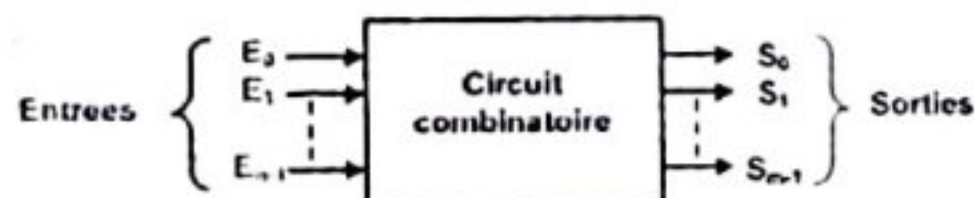
Année universitaire : 2024-2025



## Les circuits combinatoires

### Introduction

- ❖ On appelle circuit ou système combinatoire tout système numérique dont les sorties sont définies uniquement à partir des variables d'entrée.



- ❖ Les circuits combinatoires sont établis à partir d'une opération appelée synthèse combinatoire qui consiste à

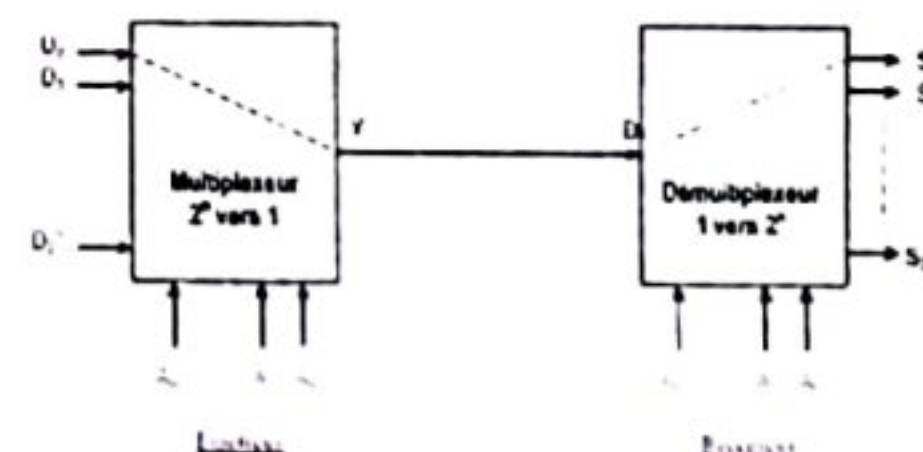
- Traduire le cahier des charges décrivant le fonctionnement du système en une table de vérité
- Déduire les équations des différentes sorties en fonctions des variables d'entrée.
- Simplifier ces équations.
- Etablir le schéma de réalisation (logigramme) correspondant.

### LES CIRCUITS ARITHMETIQUES

### Multiplexeurs et Démultiplexeurs

- ❖ Si on veut transmettre des informations en parallèle, il faut autant de lignes que d'informations.

- ❖ Pour simplifier la transmission (la rendre plus économique) surtout lorsque l'émetteur et le récepteur sont éloignés l'un de l'autre on effectue une conversion parallèle/série (multiplexage) à l'émission et une conversion série/parallèle (démultiplexage) à la réception.





## Les multiplexeurs

- ❖ Un multiplexeur est un composant électronique utilisé en plusieurs domaines.
- ❖ Son principal objectif est de combiner plusieurs signaux d'entrée en un seul signal de sortie, en utilisant un schéma de sélection ou de commutation.
- ❖ Un multiplexeur (MUX) est un circuit combinatoire qui possède  $2^n$  entrées de données ( $D_0, D_1, \dots, D_{2^n-1}$ ),  $n$  entrées ( $E_0, E_1, \dots, E_{n-1}$ ) appelées entrées de sélection ou d'adresse et une seule sortie ( $S$ ).
- ❖ Il permet d'effectuer l'aiguillage de l'une des entrées vers la sortie en fonction de l'adresse appliquée sur les entrées de sélection.

❖ Il est dit : MUX  $2^n$  vers 1 ou MUX  $2^n \times 1$ .

3

### Exemple

Multiplexeur 4 vers 1

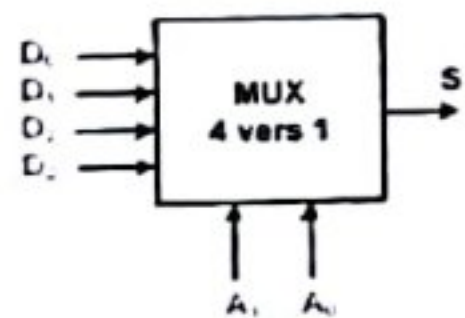


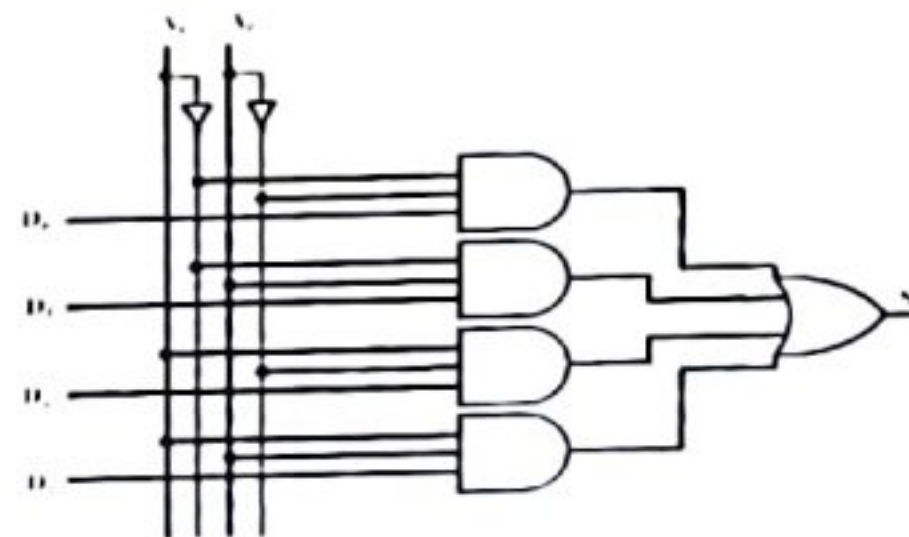
Table de vérité

$A_1$	$A_0$	$S$
0	0	$D_0$
0	1	$D_1$
1	0	$D_2$
1	1	$D_3$

Equation logique de la sortie

$$S = D_0 \bar{A}_1 \bar{A}_0 + D_1 \bar{A}_1 A_0 + D_2 A_1 \bar{A}_0 + D_3 A_1 A_0$$

Logigramme



5

Table de vérité							Logigramme
Décimal	Entrées					Sortie	
	$E_{n-1}$	...	$E_2$	$E_1$	$E_0$	$Y$	
0	0	...	0	0	0	$D_0$	
1	0	...	0	0	1	$D_1$	
2	0	...	0	1	0	$D_2$	
3	0	...	0	1	1	$D_3$	
4	0	...	1	0	0	$D_4$	
5	0	...	1	0	1	$D_5$	
...	...	...	...	...	...	...	
$2^n - 1$	1	...	1	1	1	$D_{2^n-1}$	

4

## Les démultiplexeurs

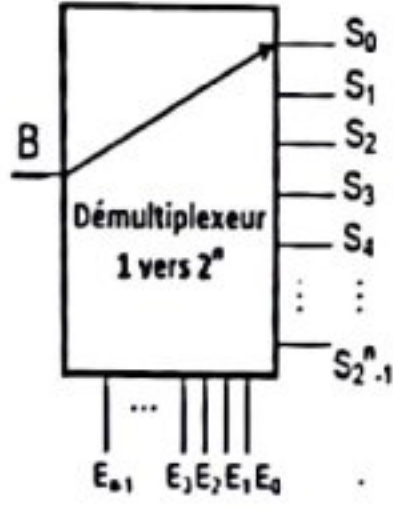
- ❖ Un démultiplexeur (DEMUX) est un circuit combinatoire qui possède une seule entrée de données ( $B$ ),  $n$  entrées de sélection ( $E_0, E_1, \dots, E_{n-1}$ ) et  $2^n$  sorties ( $S_0, S_1, \dots, S_{2^n-1}$ ).
- ❖ Il permet d'effectuer l'aiguillage de l'entrée vers l'une des sorties en fonction de l'adresse appliquée sur les entrées de sélection.

Il est dit : DEMUX 1 vers  $2^n$  ou DEMUX  $1 \times 2^n$

- ❖ Il effectue la fonction inverse d'un multiplexeur, il transmet la donnée d'entrée vers une des sorties selon le mot écrit aux entrées de sélection, il fonctionne comme un commutateur.

6

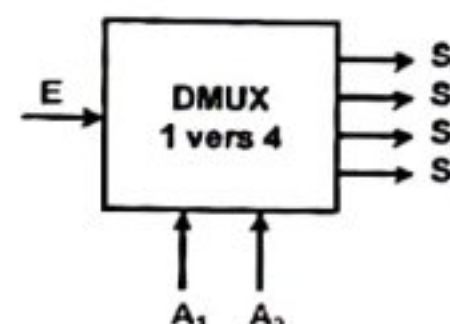


Table de vérité											Logigramme	
Décimal	Entrées					Sorties						
	$E_{n-1}$	...	$E_2$	$E_1$	$E_0$	$S_0$	$S_1$	$S_2$	...	$S_{2^n-1}$		
0	0	...	0	0	0	B	0	0	...	0		
1	0	...	0	0	1	0	B	0	...	0		
2	0	...	0	1	0	0	0	B	...	0		
3	0	...	0	1	1	0	0	0	...	0		
4	0	...	1	0	0	0	0	0	...	0		
5	0	...	1	0	1	0	0	0	...	0		
...	...	...	...	...	...	...	...	...	...	...		
$2^n - 1$	1	...	1	1	1	0	0	0	...	B		

7

## Exemple

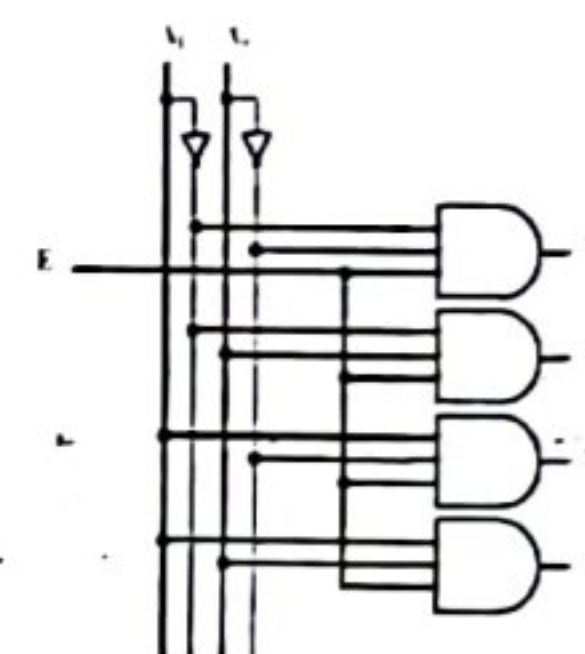
### Démultiplexeur 1 vers 4



### Table de vérité

$A_1$	$A_0$	$S_3$	$S_2$	$S_1$	$S_0$
0	0	0	0	0	E
0	1	0	0	E	0
1	0	0	E	0	0
1	1	E	0	0	0

### Logigramme :



### Equations logiques de la sortie :

$$S_0 = E \bar{A}_1 \bar{A}_0$$

$$S_1 = E \bar{A}_1 A_0$$

$$S_2 = E A_1 \bar{A}_0$$

$$S_3 = E A_1 A_0$$

8

## Les Additionneurs

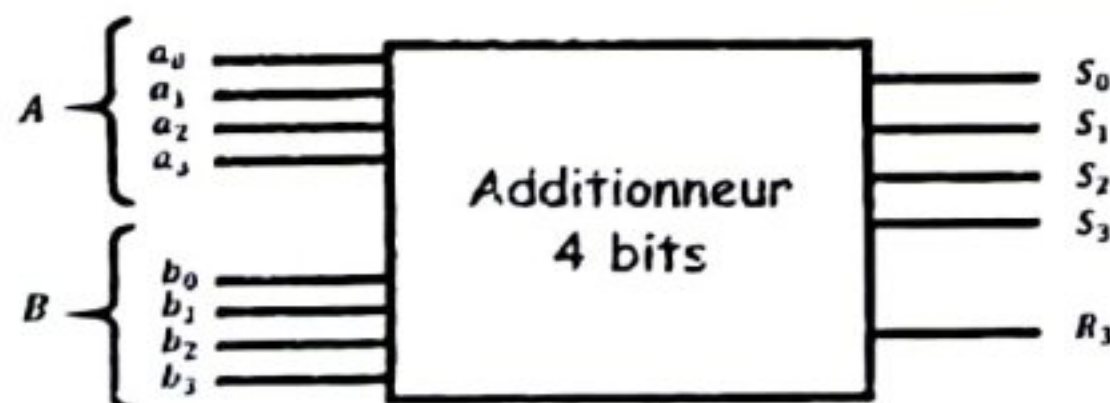
❖ Un additionneur est un circuit capable de faire la somme de deux nombres binaires A et B.

❖ Une addition met en œuvre deux sorties :

- La somme, généralement notée S.
- La retenue, généralement notée R (ou C : carry).

❖ Comme en décimal, nous devons tenir compte de la retenue éventuelle, résultat d'un calcul précédent.

### La décomposition de l'addition de deux nombres binaires de 4 bits.



	$a_3$	$a_2$	$a_1$	$a_0$	Nombre A
+	$b_3$	$b_2$	$b_1$	$b_0$	Nombre B
=	$S_3$	$S_2$	$S_1$	$S_0$	Somme A+B
=	$r_3$	$r_2$	$r_1$	$r_0$	Retenue

9

## Demi-additionneur (2 bits)

❖ C'est un additionneur de deux nombres binaires de 1 bit chacun.

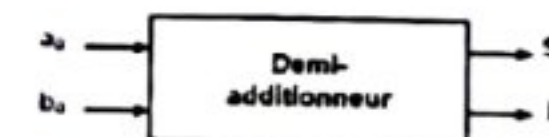
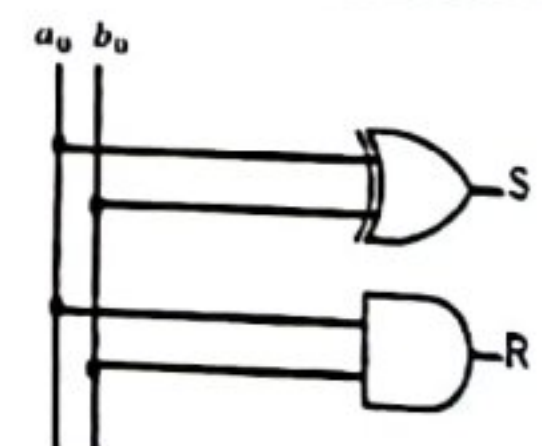


Table de vérité	Equations logiques des sorties	Logigramme																				
<table><tr><th><math>a_0</math></th><th><math>b_0</math></th><th>R</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	$a_0$	$b_0$	R	S	0	0	0	0	0	1	0	1	1	0	0	1	1	1	1	0	$S = \bar{a}_0 b_0 + a_0 \bar{b}_0 = a_0 \oplus b_0$ $R = a_0 b_0$	
$a_0$	$b_0$	R	S																			
0	0	0	0																			
0	1	0	1																			
1	0	0	1																			
1	1	1	0																			

10



## Additionneur complet (2 bits)

- ❖ Un additionneur complet comporte 3 entrées : les deux bits à additionner  $a_i$  et  $b_i$ , et la retenue issue de l'addition des 2 bits de rang inférieur (rang  $n-1$ )  $R_e$  (dite entrante).
- ❖ Il possède 2 sorties : la somme  $S_i$  et la retenue sortante  $R_s$  (rang  $n$ ).

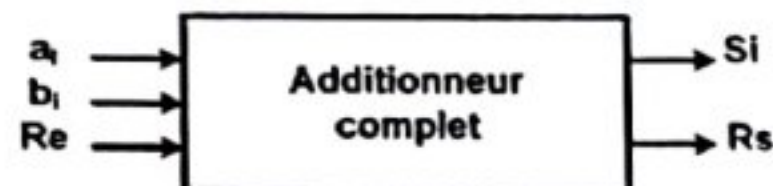


Table de vérité

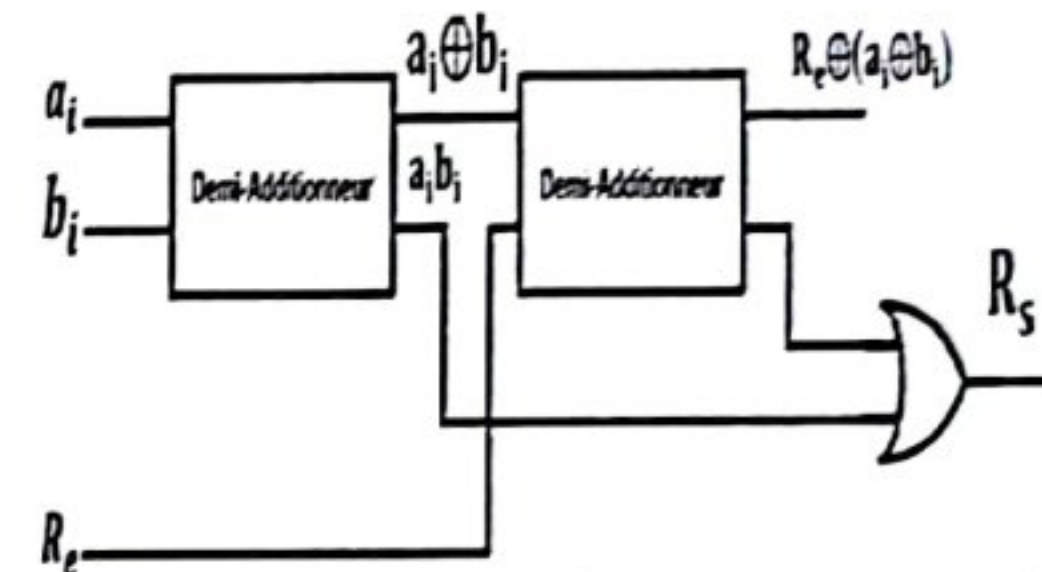
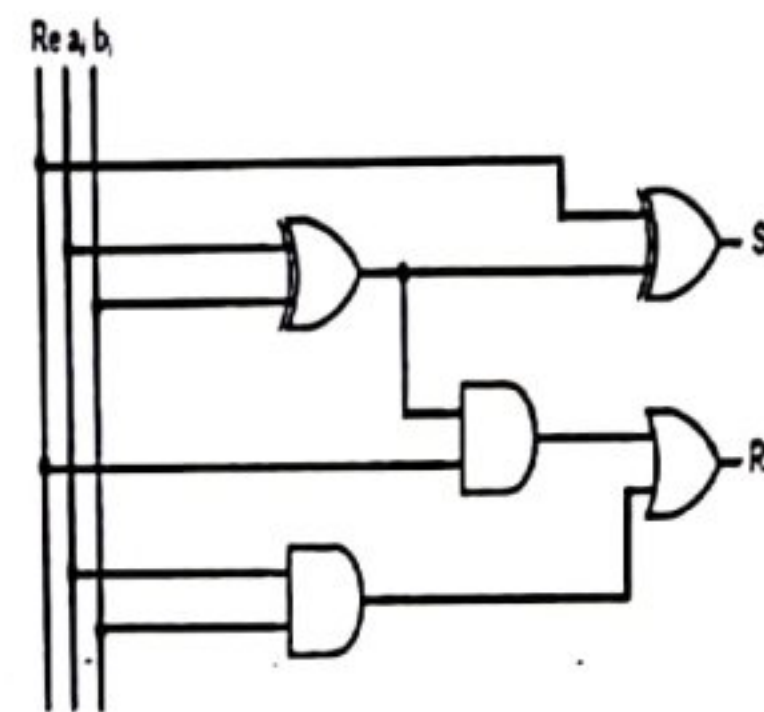
$R_e$	$a_i$	$b_i$	$R_s$	$S_i$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Equations logiques des sorties

$$\begin{aligned}
 S_i &= \bar{R}_e \bar{a}_i b_i + \bar{R}_e a_i \bar{b}_i + R_e \bar{a}_i \bar{b}_i + R_e a_i b_i \\
 S_i &= \bar{R}_e (a_i \oplus b_i) + R_e (a_i \odot b_i) \\
 S_i &= R_e \oplus (a_i \oplus b_i) \\
 R_s &= \bar{R}_e a_i b_i + R_e \bar{a}_i b_i + R_e a_i \bar{b}_i + R_e a_i b_i \\
 R_s &= R_e (a_i \oplus b_i) + a_i b_i (\bar{R}_e + R_e) \\
 R_s &= R_e (a_i \oplus b_i) + a_i b_i
 \end{aligned}$$

11

## Logigramme



12

## Les Soustrakteurs

### Demi-Soustracteur (2 bits)

- ❖ Un demi-soustracteur ne tient pas compte d'une éventuelle retenue provenant des bits de poids inférieurs.
- ❖ D représente le résultat de la différence (A-B) et R la retenue.

Table de vérité	Equations logiques des sorties	Logigramme																				
<table><tr><th>A</th><th>B</th><th>D</th><th>R</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	A	B	D	R	0	0	0	0	0	1	1	1	1	0	1	0	1	1	0	0	$D = \bar{A} B + A \bar{B}$ $= A \oplus B$ $R = \bar{A} B$	
A	B	D	R																			
0	0	0	0																			
0	1	1	1																			
1	0	1	0																			
1	1	0	0																			

13

### Soustracteur Complet (2 bits)

- ❖ Il possède trois entrées A, B et  $R_e$  et deux sorties D et  $R_s$ .
- $R_e$  représente la retenue de rang  $n-1$ .
- $R_s$  celle de rang  $n$ .

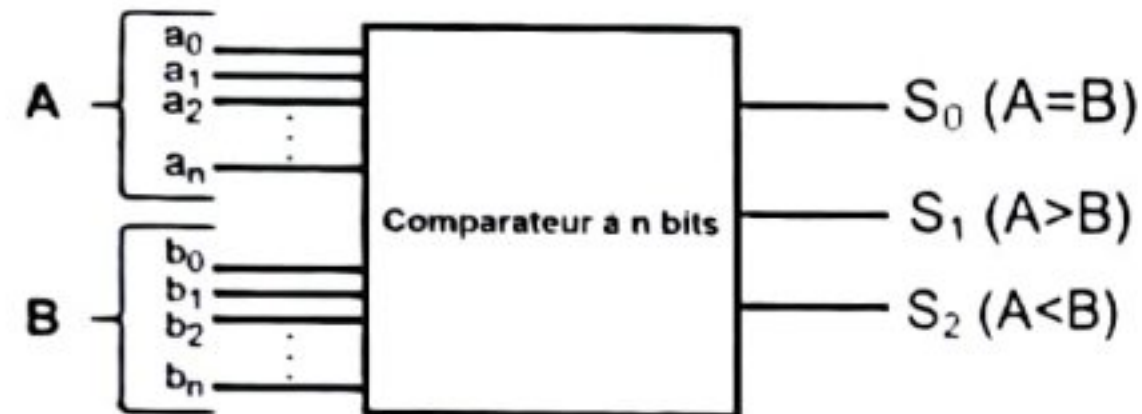
Table de vérité	Equations logiques des sorties	Logigramme																																													
<table><tr><th>A</th><th>B</th><th><math>R_e</math></th><th>D</th><th><math>R_s</math></th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	A	B	$R_e$	D	$R_s$	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	$D = \bar{A} \bar{B} R_e + \bar{A} B \bar{R}_e + A \bar{B} \bar{R}_e + A B R_e$ $= A \oplus B \oplus R_e$ $R_s = R_e A \odot B + \bar{A} B$	<p>The diagram shows a 2-bit full subtractor implementation. It uses two 'Demi-Soustracteur' (Half Subtractor) blocks and an OR gate. The first block takes inputs A, B, and <math>R_e</math> and produces outputs S and <math>R_1</math>. The second block takes inputs A and B and produces outputs <math>A \oplus B</math> and <math>A \odot B</math>. The final output D is the XOR of S and <math>A \oplus B</math>. The final output <math>R_s</math> is the OR of <math>R_1</math> and <math>A \odot B</math>.</p>
A	B	$R_e$	D	$R_s$																																											
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																																											

14



## Comparateur

- ❖ C'est un circuit qui permet de comparer 2 nombres.
- ❖ Il indique si le premier nombre est inférieur ( $S_2$ ), égal ( $S_0$ ) ou supérieur ( $S_1$ ) au second nombre.



### Principe de base

- ❑ Le principe consiste de comparer d'abord les bits les plus significatifs (Most Significant Bit ou **MSB**).
- ❑ S'ils sont différents, il est inutile de continuer la comparaison.
- ❑ Par contre s'ils sont égaux, il faut comparer les bits de poids immédiatement inférieur et ainsi de suite.

15

## Le comparateur de 1 bit

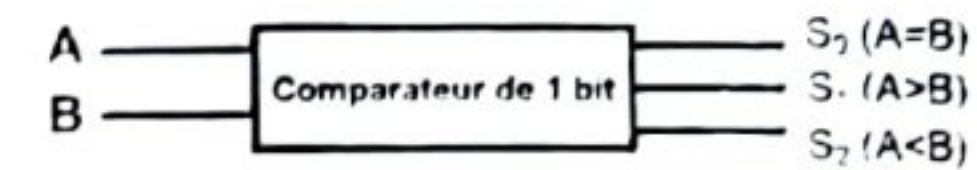
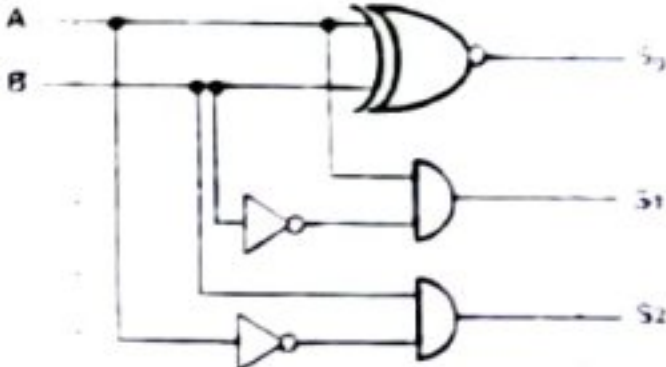
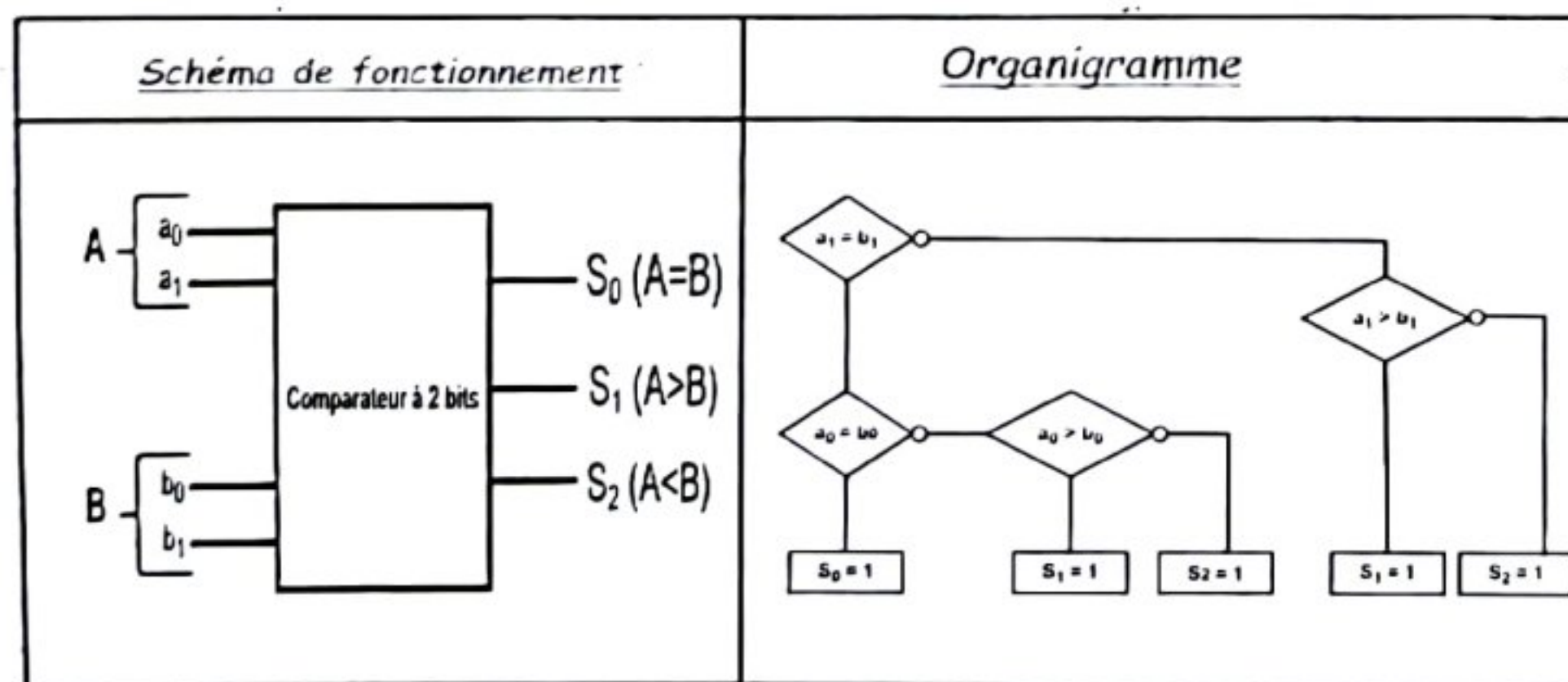


Table de verite	Fonctions logiques des sorties	Logigramme																									
<table><tr><th>B</th><th>A</th><th>S<sub>0</sub></th><th>S<sub>1</sub></th><th>S<sub>2</sub></th></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	B	A	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	0	0	$S_0 = \overline{A}\overline{B} + AB = \overline{A \oplus B}$ $S_1 = A\overline{B}$ $S_2 = \overline{A}B$	
B	A	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>																							
0	0	1	0	0																							
0	1	0	1	0																							
1	0	0	0	1																							
1	1	1	0	0																							

16

## Le comparateur de 2 bits



17

Table de vérité						
b <sub>1</sub>	b <sub>0</sub>	a <sub>1</sub>	a <sub>0</sub>	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>
0	0	0	0	1	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	1	0	0
1	0	1	1	0	1	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	1	0	0

### Equations

On a S<sub>0</sub> vaut 1 si a<sub>1</sub>=b<sub>1</sub> et si a<sub>0</sub>=b<sub>0</sub>

$$S_0 = (a_1 \odot b_1) \cdot (a_0 \odot b_0)$$

Et S<sub>1</sub> vaut 1 si a<sub>1</sub>>b<sub>1</sub> ou si (a<sub>1</sub>=b<sub>1</sub> et a<sub>0</sub>>b<sub>0</sub>)

$$S_1 = a_1 \overline{b_1} + (a_1 \odot b_1) a_0 \overline{b_0}$$

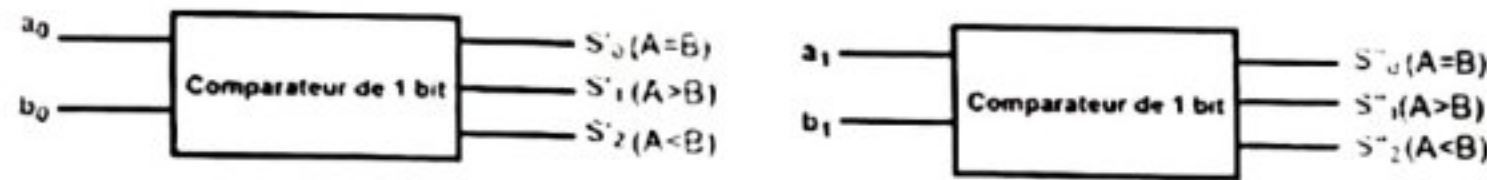
Et S<sub>2</sub> vaut 1 si a<sub>1</sub><b<sub>1</sub> ou si (a<sub>1</sub>=b<sub>1</sub> et a<sub>0</sub><b<sub>0</sub>)

$$S_2 = \overline{a_1} b_1 + (a_1 \odot b_1) \overline{a_0} b_0$$

$$S_2 = \overline{S_0} + S_1$$



## Logigramme à l'aide des 2 comparateurs à 1 bit



### Equations

$$S_0 = (a_1 \odot b_1) \cdot (a_0 \odot b_0) = S''_0 S'_0$$

Et  $S_1$  vaut 1 si  $a_1 > b_1$  ou si  $(a_1 = b_1 \text{ et } a_0 > b_0)$

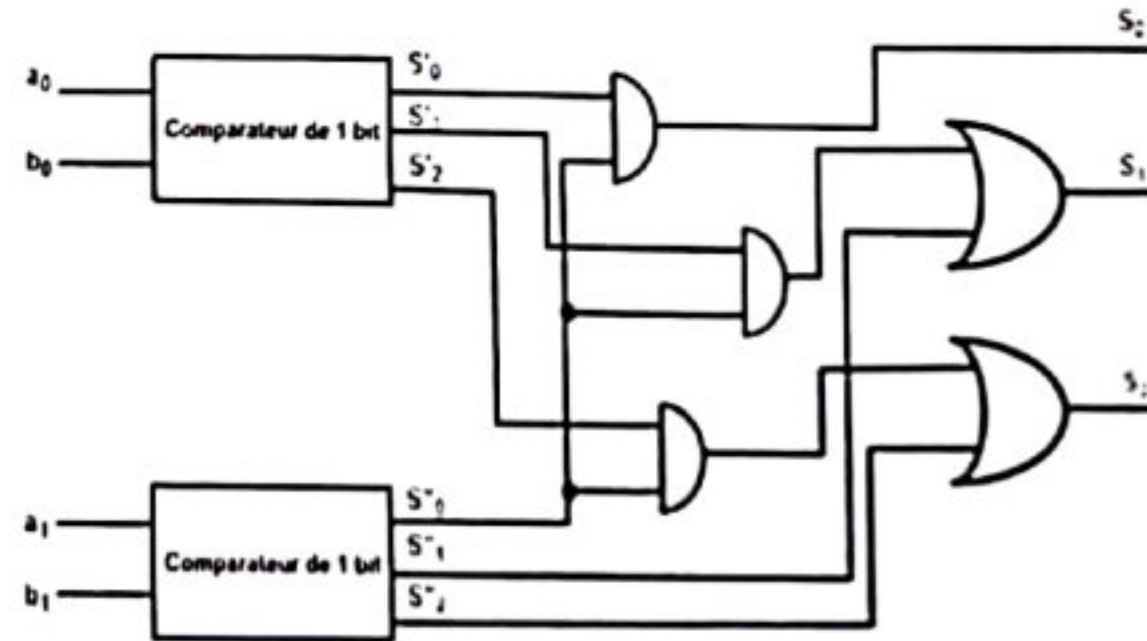
$$S_1 = a_1 \bar{b}_1 + (a_1 \odot b_1) a_0 \bar{b}_0 = S''_1 + S''_0 S'_1$$

Et  $S_2$  vaut 1 si  $a_1 < b_1$  ou si  $(a_1 = b_1 \text{ et } a_0 < b_0)$

$$S_2 = \bar{a}_1 \bar{b}_1 + (a_1 \odot b_1) \bar{a}_0 \bar{b}_0 = S''_2 + S''_0 S'_2$$

$$S_2 = \overline{S_0 + S_1}$$

### Logigramme



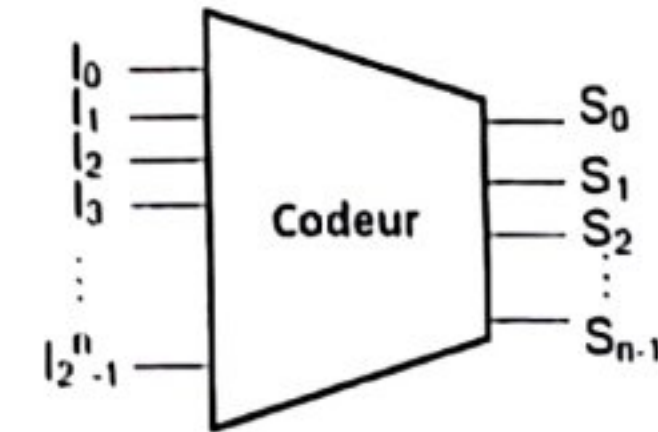
19

## Codeurs et Décodeurs

### Les Codeurs

❖ C'est un circuit qui traduit les valeurs d'une entrée dans un code choisi.

❖ Un codeur (ou encodeur) est un circuit logique qui possède  $2^n$  voies d'entrées dont une seule est activée et N voies de sorties ( $2^n \leq 2^N$ ).



20

### Exemple: Codeur 4 vers 2



- Ce codeur possède 4 entrées et 2 sorties
- Une seule entrée doit être activée à la fois (par un état haut)
  - On retrouve alors en sortie, en binaire, le numéro de l'entrée active entre 0 et 3.

### Table de vérité

Entrées				Sorties	
E3	E2	E1	E0	S1	S0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

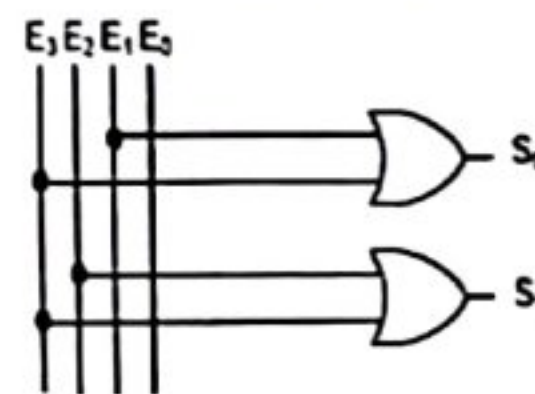
Entrée active	Sorties	
	S1	S0
E0	0	0
E1	0	1
E2	1	0
E3	1	1

### Equations logiques des sorties

$$S_0 = E_1 + E_3$$

$$S_1 = E_2 + E_3$$

### Logigramme



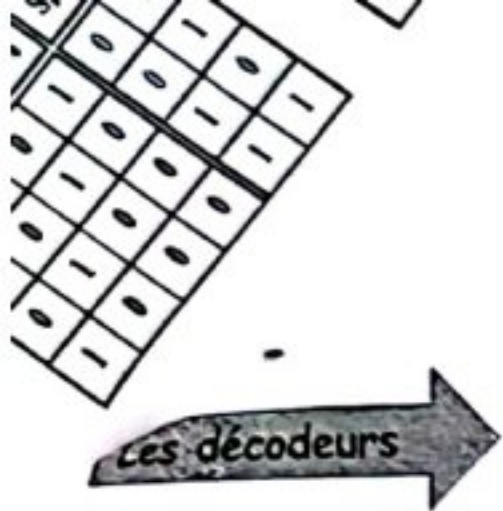
21

### Exemple: Codeur DCB

Table de vérité	Equations logiques des sorties	Logigramme																																																											
<table><tr><th rowspan="2">Entrées</th><th colspan="4">Sorties</th></tr><tr><th>a<sub>3</sub></th><th>a<sub>2</sub></th><th>a<sub>1</sub></th><th>a<sub>0</sub></th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>2</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>3</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>4</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>5</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>6</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>7</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>8</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>9</td><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>	Entrées	Sorties				a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	0	0	0	0	0	1	0	0	0	1	2	0	0	1	0	3	0	0	1	1	4	0	1	0	0	5	0	1	0	1	6	0	1	1	0	7	0	1	1	1	8	1	0	0	0	9	1	0	0	1	$a_0 = 1 \cdot 3 \cdot 5 \cdot 7 \cdot 9$ $a_1 = 2 \cdot 3 \cdot 6 \cdot 7$ $a_2 = 4 \cdot 5 \cdot 6 \cdot 7$ $a_3 = 8 \cdot 9$	<pre>graph LR     subgraph Inputs         direction TB         I0[0]         I1[1]         I2[2]         I3[3]         I4[4]         I5[5]         I6[6]         I7[7]         I8[8]         I9[9]     end     subgraph DCB [Codeur DCB]         direction TB         Oa3[a3]         Oa2[a2]         Oa1[a1]         Oa0[a0]     end     subgraph OR_Gates         direction TB         OR0((OR))         OR1((OR))         OR2((OR))         OR3((OR))     end     I1 --&gt; OR0     I3 --&gt; OR0     I5 --&gt; OR0     I7 --&gt; OR0     I9 --&gt; OR0     OR0 --&gt; Oa0     I2 --&gt; OR1     I3 --&gt; OR1     I6 --&gt; OR1     I7 --&gt; OR1     OR1 --&gt; Oa1     I4 --&gt; OR2     I5 --&gt; OR2     I6 --&gt; OR2     I7 --&gt; OR2     OR2 --&gt; Oa2     I8 --&gt; OR3     I9 --&gt; OR3     OR3 --&gt; Oa3</pre>
Entrées		Sorties																																																											
	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>																																																									
0	0	0	0	0																																																									
1	0	0	0	1																																																									
2	0	0	1	0																																																									
3	0	0	1	1																																																									
4	0	1	0	0																																																									
5	0	1	0	1																																																									
6	0	1	1	0																																																									
7	0	1	1	1																																																									
8	1	0	0	0																																																									
9	1	0	0	1																																																									

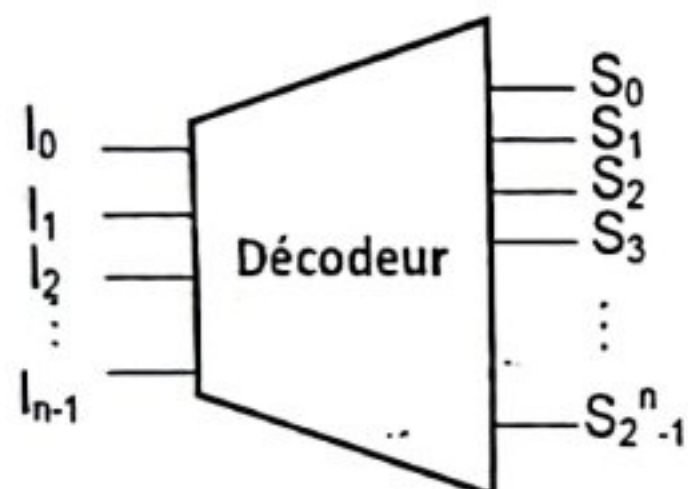
22





## Les décodeurs

- Un décodeur est un circuit à N entrées et  $2^n$  sorties dont une seule est active à la fois.
- Il détecte la présence d'une combinaison spécifique de bits (code) à ces entrées et l'indique par un niveau spécifique de sortie ( $N \leq 2^n$ ).



23

## Exemple: Décodeur 2 vers 4

- Ce décodeur possède 2 entrées et 4 sorties.
- Une seule sortie est activée à la fois (par un état haut) : celle dont l'indice (entre 0 et 3) correspond au nombre (sur 2 bits) appliqué en binaire sur les entrées.

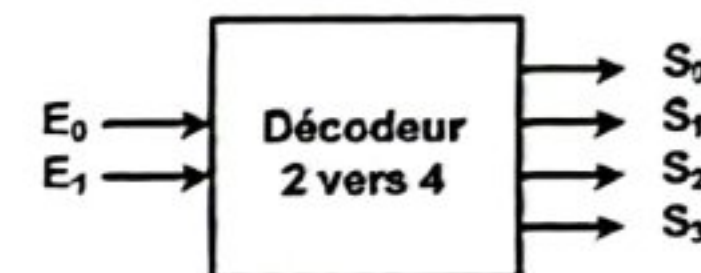


Table de vérité

Entrées		Sorties			
E <sub>1</sub>	E <sub>0</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Entrées		Sortie active
E <sub>1</sub>	E <sub>0</sub>	
0	0	S <sub>0</sub>
0	1	S <sub>1</sub>
1	0	S <sub>2</sub>
1	1	S <sub>3</sub>

Equations logiques des sorties

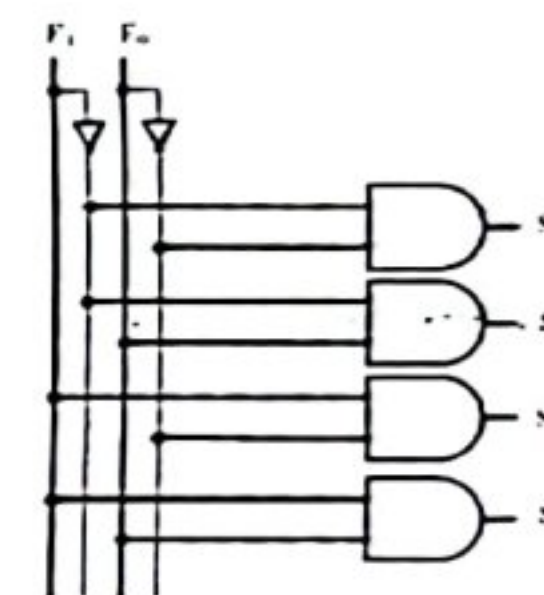
$$S_0 = E_1 \bar{E}_0$$

$$S_1 = \bar{E}_1 E_0$$

$$S_2 = E_1 E_0$$

$$S_3 = \bar{E}_1 \bar{E}_0$$

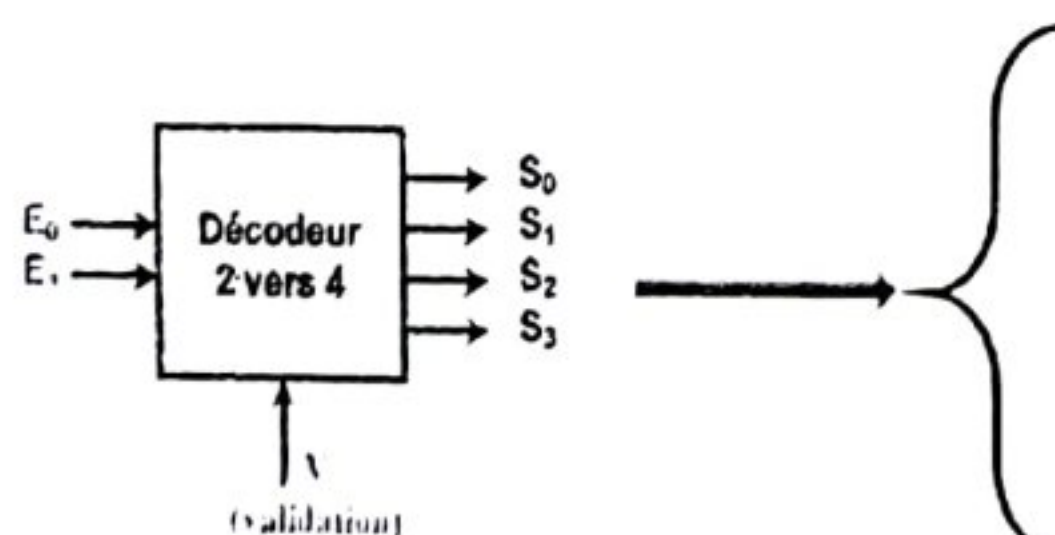
Logigramme



24

## Remarque

- ❑ On peut ajouter une entrée supplémentaire dite entrée de validation (V) pour pouvoir associer plusieurs décodeurs ensembles.



➤ Si  $V=0$  : Le décodeur ne fonctionne pas.

➤ Si  $V=1$  : Le décodeur fonctionne normalement.

25

## Exemple: Décodeur DCB

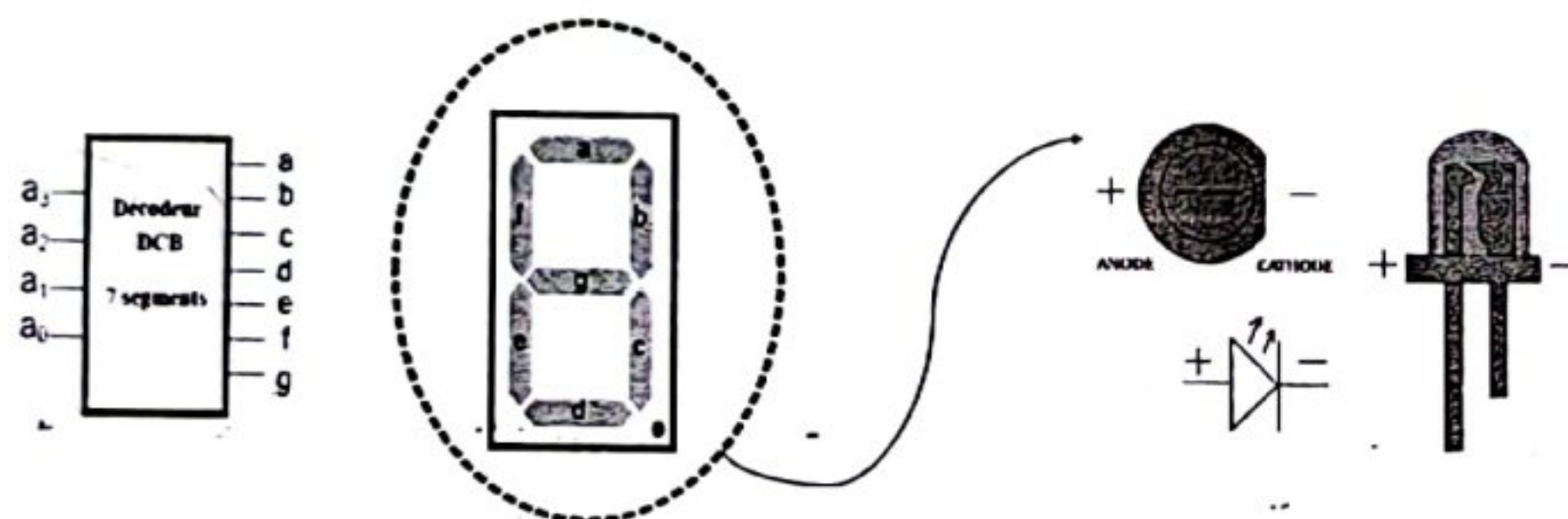
Table de vérité	Equations logiques des sorties	Logigramme																																																											
<table><tr><th colspan="4">Entrées</th><th rowspan="2">Sorties</th></tr><tr><th>a<sub>3</sub></th><th>a<sub>2</sub></th><th>a<sub>1</sub></th><th>a<sub>0</sub></th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>S<sub>0</sub></td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>S<sub>1</sub></td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>S<sub>2</sub></td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>S<sub>3</sub></td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>S<sub>4</sub></td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>S<sub>5</sub></td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>S<sub>6</sub></td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>S<sub>7</sub></td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>S<sub>8</sub></td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>S<sub>9</sub></td></tr></table>	Entrées				Sorties	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	0	0	0	0	S <sub>0</sub>	0	0	0	1	S <sub>1</sub>	0	0	1	0	S <sub>2</sub>	0	0	1	1	S <sub>3</sub>	0	1	0	0	S <sub>4</sub>	0	1	0	1	S <sub>5</sub>	0	1	1	0	S <sub>6</sub>	0	1	1	1	S <sub>7</sub>	1	0	0	0	S <sub>8</sub>	1	0	0	1	S <sub>9</sub>	$S_0 = \overline{a_3} \overline{a_2} \overline{a_1} \overline{a_0}$ $S_1 = \overline{a_3} \overline{a_2} \overline{a_1} a_0$ $S_2 = \overline{a_3} \overline{a_2} a_1 \overline{a_0}$ $S_3 = \overline{a_3} \overline{a_2} a_1 a_0$ $S_4 = \overline{a_3} a_2 \overline{a_1} \overline{a_0}$ $S_5 = \overline{a_3} a_2 \overline{a_1} a_0$ $S_6 = \overline{a_3} a_2 a_1 \overline{a_0}$ $S_7 = \overline{a_3} a_2 a_1 a_0$ $S_8 = a_3 \overline{a_2} \overline{a_1} \overline{a_0}$ $S_9 = a_3 \overline{a_2} \overline{a_1} a_0$	
Entrées				Sorties																																																									
a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>																																																										
0	0	0	0	S <sub>0</sub>																																																									
0	0	0	1	S <sub>1</sub>																																																									
0	0	1	0	S <sub>2</sub>																																																									
0	0	1	1	S <sub>3</sub>																																																									
0	1	0	0	S <sub>4</sub>																																																									
0	1	0	1	S <sub>5</sub>																																																									
0	1	1	0	S <sub>6</sub>																																																									
0	1	1	1	S <sub>7</sub>																																																									
1	0	0	0	S <sub>8</sub>																																																									
1	0	0	1	S <sub>9</sub>																																																									

26



## Le décodeur DCB 7 segments

- Le décodeur 7 segments accepte en entrée les 4 bits DCB ( $a_0, a_1, a_2, a_3$ ) et rend actives les sorties qui vont permettre de faire passer un courant dans les segments d'un afficheur numérique pour former les chiffres décimaux (de 0 à 9).

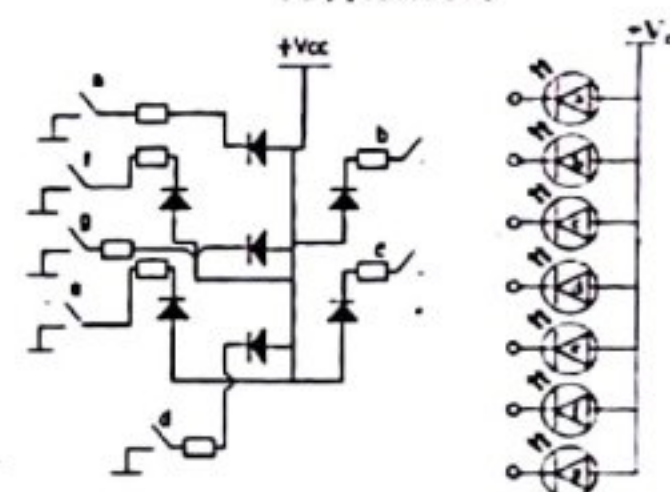


27

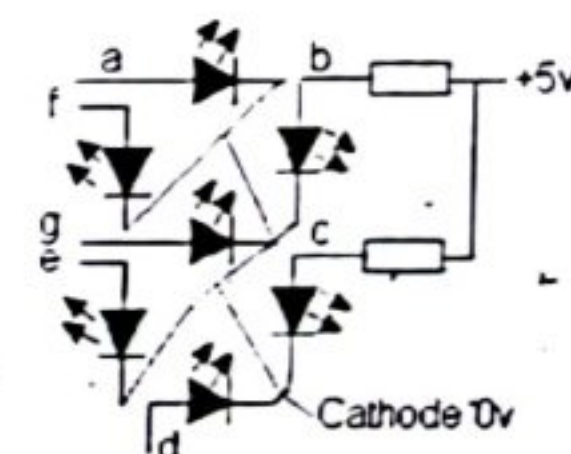
## Remarque :

- L'afficheur est composée de 7 LEDS (segments), a, d, c, d, e, f, g qui nécessitent en fonction du type d'afficheur (anode commune ou cathode commune) une polarisation spécifique :

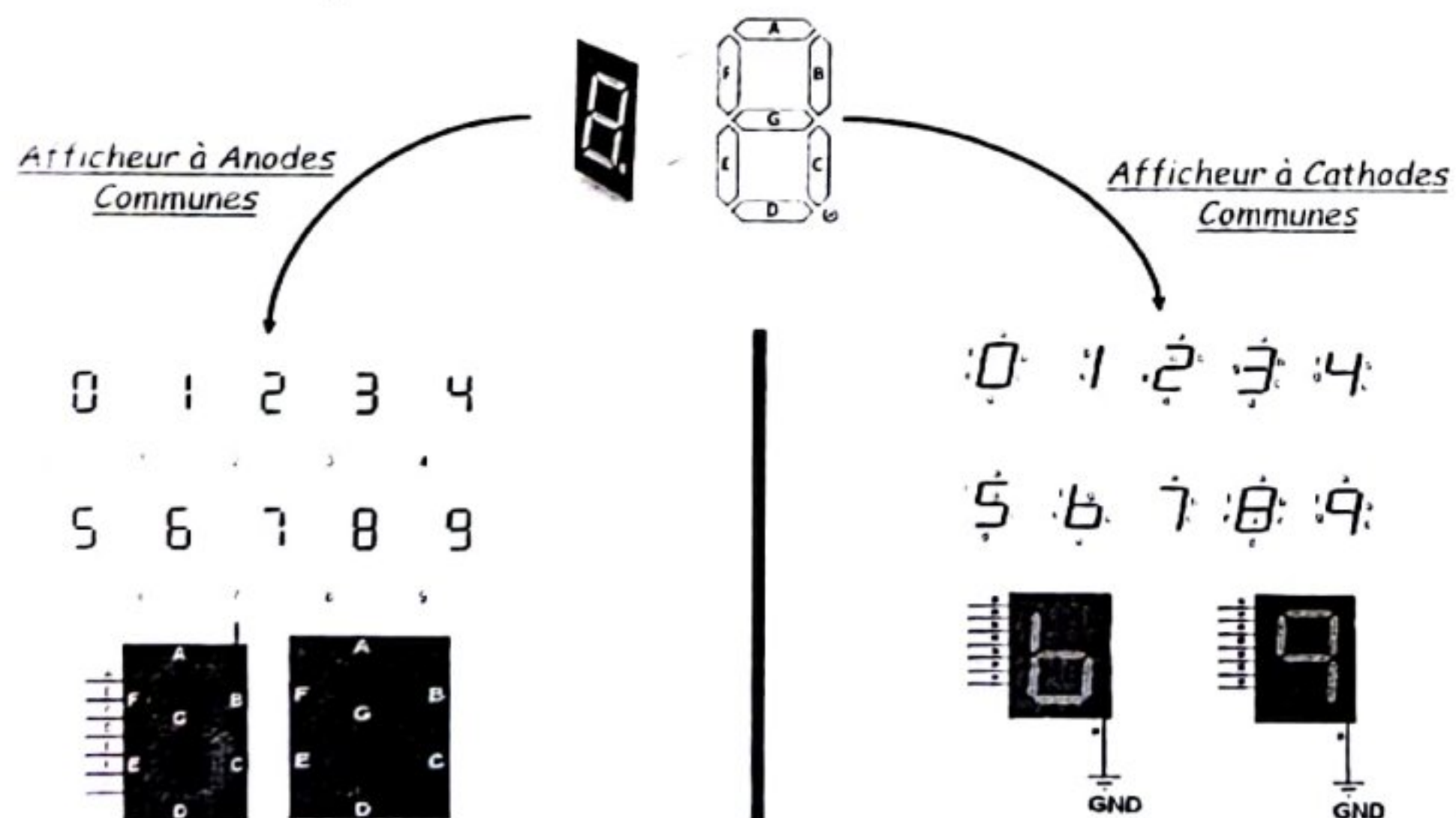
➤ Pour un afficheur à anodes communes : Les anodes sont reliées ensembles au niveau haut et les sorties du décodeur sont actives au niveau bas et sont reliées aux cathodes de l'afficheur.



➤ Pour un afficheur à cathodes communes : Les cathodes sont reliées ensembles à la masse et les sorties du décodeur sont actives au niveau haut et sont reliées aux anodes de l'afficheur.



28



29

## Cas d'un Afficheur à Cathodes Communes

Table de vérité											Affichage
Entrées				Sorties							
$a_3$	$a_2$	$a_1$	$a_0$	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9

30



en fonction du type

**Segment a**

$a_3 a_2$	00	01	11	10
$a_1 a_0$	1	0	0	1
01	0	1	0	1
11	1	1	0	0
10	1	0	0	0

$a = \overline{a_2}a_1 + a_2 a_0 + \overline{a_2} \overline{a_0} + a_3$

**Segment b**

$a_3 a_2$	00	01	11	10
$a_1 a_0$	1	1	0	1
01	1	0	0	1
11	1	1	0	0
10	1	0	0	0

$b = \overline{a_2} + \overline{a_1} \overline{a_0} + a_1 a_0$   
 $= \overline{a_2} + a_0 \odot a_1$

**Segment c**

$a_3 a_2$	00	01	11	10
$a_1 a_0$	1	1	0	1
01	1	1	0	1
11	1	1	0	0
10	0	1	0	0

$c = a_2 + \overline{a_1} + a_0$

**Segment d**

$a_3 a_2$	00	01	11	10
$a_1 a_0$	1	0	0	1
01	0	1	0	0
11	1	0	0	0
10	1	1	0	0

$d = \overline{a_2} \overline{a_0} + \overline{a_2} a_1 + a_1 \overline{a_0} + a_2 \overline{a_1} a_0$

**Segment e**

$a_3 a_2$	00	01	11	10
$a_1 a_0$	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	1	0	0

$e = a_1 \overline{a_0} + \overline{a_2} \overline{a_0}$

**Segment f**

$a_3 a_2$	00	01	11	10
$a_1 a_0$	1	1	0	1
01	0	1	0	1
11	0	0	0	0
10	0	1	0	0

$f = \overline{a_1} \overline{a_0} + a_2 \overline{a_1} + a_2 \overline{a_0} + a_3$

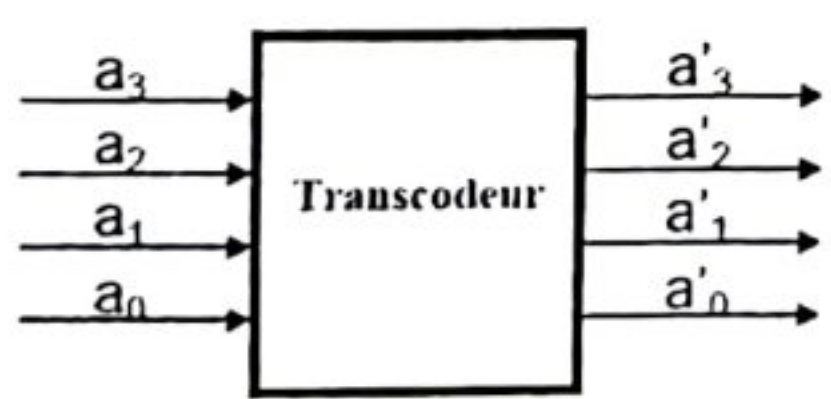
**Segment g**

$a_3 a_2$	00	01	11	10
$a_1 a_0$	0	1	0	1
01	0	1	0	1
11	1	0	0	0
10	1	1	0	0

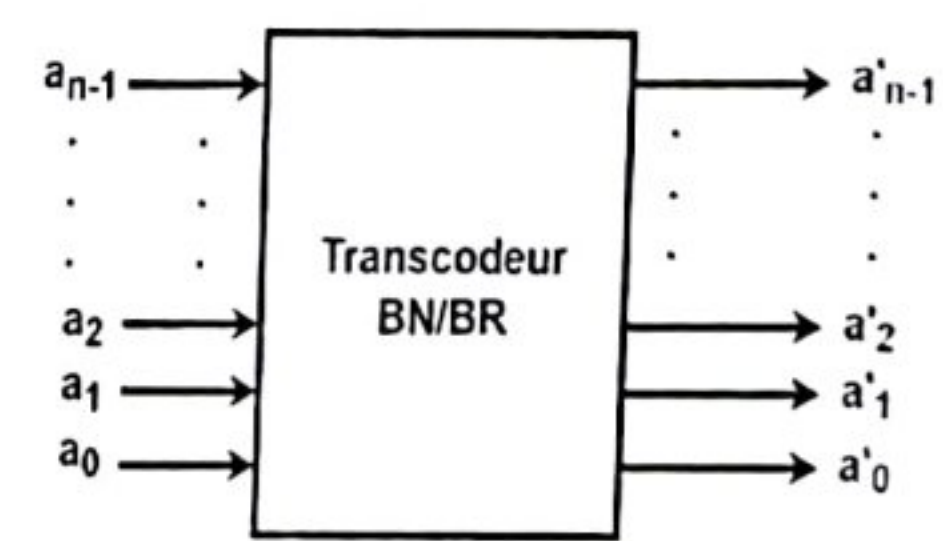
$g = \overline{a_2} a_1 + a_2 \overline{a_0} + a_2 \overline{a_1} + a_3$

## Transcodeurs

❖ Un transcodeur est un circuit qui permet de faire passer une information écrite dans un code  $C_1$  vers un code  $C_2$ .



## Transcodeur Binaire Naturel-Binaire Réfléchi



Exemple: Transcodeur BN/BR (4 bits)

Table de vérité								Décimal
Entrées BN				Sorties BR				
$a_3$	$a_2$	$a_1$	$a_0$	$a'_3$	$a'_2$	$a'_1$	$a'_0$	
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	2
0	0	1	1	0	0	1	0	3
0	1	0	0	0	1	1	0	4
0	1	0	1	0	1	1	1	5
0	1	1	0	0	1	0	1	6
0	1	1	1	0	1	0	0	7
1	0	0	0	1	1	0	0	8
1	0	0	1	1	1	0	1	9
1	0	1	0	1	1	1	1	10
1	0	1	1	1	1	1	0	11
1	1	0	0	1	0	1	0	12
1	1	0	1	1	0	1	1	13
1	1	1	0	1	0	0	1	14
1	1	1	1	1	0	0	0	15



Tables de Fonctionnement

Bit  $a'_3$

$a_3 a_2$ $a_1 a_0$	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	1	1
10	0	0	1	1

Bit  $a'_1$

$a_3 a_2$ $a_1 a_0$	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	1

Bit  $a'_2$

$a_3 a_2$ $a_1 a_0$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

Bit  $a'_0$

$a_3 a_2$ $a_1 a_0$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

Equation des sorties et logigramme

$$a'_3 = a_3$$

$$a'_2 = a_3 \oplus a_2$$

$$a'_1 = a_2 \oplus a_1$$

$$a'_0 = a_1 \oplus a_0$$

