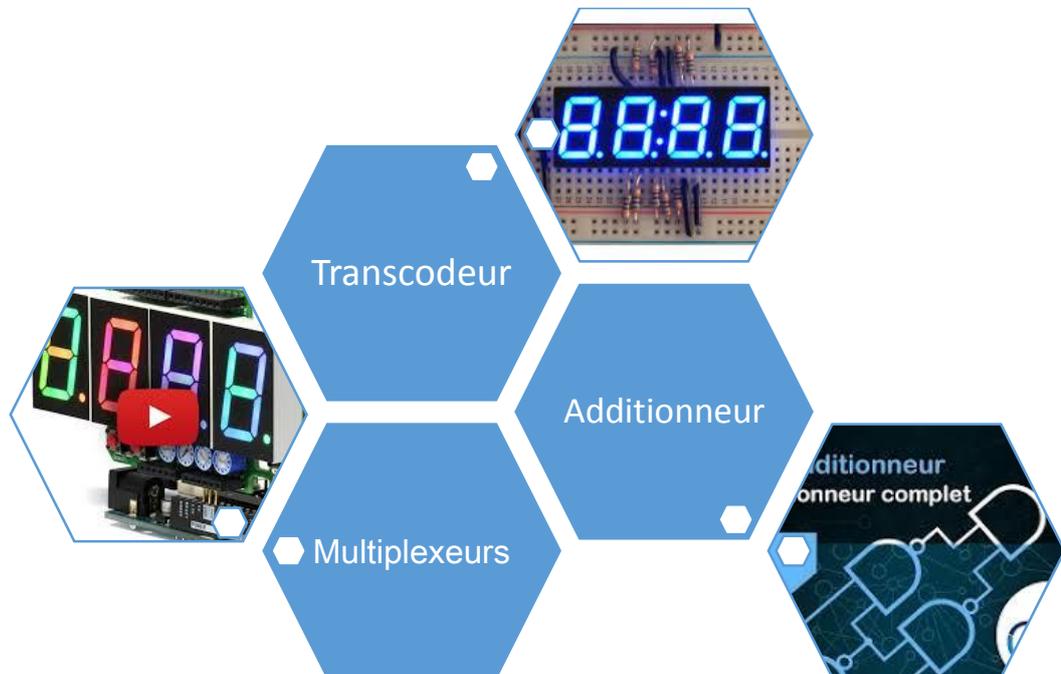


REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique  
Université Djilali Bounaâma de Khemis Miliana  
Faculté des sciences et techniques  
Département de maths et informatique  
Niveau : Licence MI



Rédigé par : Dr. MAHROUG RABIAA  
E-mail : r.mahroug@univ-dbkm.dz

# Chapitre II : Conception des Circuits Combinatoires



Année universitaire 2022-2023

# Table des matières

Table des matières.....	i
Abréviation.....	iii
Chapitre II : Conception des Circuits Combinatoires.....	1
2.1 Définition.....	1
2.2. Synthèse d'une fonction combinatoire.....	1
2.3. Classification.....	2
2.4. Circuits combinatoires usuelles.....	3
2.4.1. Additionneur.....	3
2.4.1.1. Demi-Additionneur.....	3
2.4.1.2. Additionneur complet.....	4
2.4.1.3. Additionneur à 4 bits.....	5
2.4.2. Soustracteur.....	6
2.4.2.1. Demi-soustracteur.....	6
2.4.2.2. Soustracteur complet.....	7
2.4.2.3. Soustracteur à 4 bits.....	8
2.4.3. Additionneur/Soustracteur.....	9
2.4.3. Additionneur/Soustracteur.....	9
2.4.4. Comparateur.....	9
2.4.4.1. Comparateur à 1 bit.....	9
2.4.4.2. Comparateur à 2 bits.....	10
2.4.4.3. Comparateur 2 bits avec des comparateurs 1 bit.....	12
2.4.5. Multiplexeur.....	12
2.4.5.1. Multiplexeur 2x1.....	13
2.4.5.2. Multiplexeur 4X1.....	14
2.4.5.3. Multiplexeur 8X1.....	14
2.4.6. Démultiplexeurs.....	17
2.4.6.1. Démultiplexeurs 1X4.....	17
2.4.6.2. Démultiplexeurs 1X8.....	18
2.4.7. Décodeur binaire.....	19
2.4.7.1. Décodeur binaire 2x4.....	20
2.4.7.2. Décodeur binaire 3X8.....	21
2.4.8. Encodeur binaire (codeur).....	22
2.4.8.1. Encodeur binaire 4x2.....	22
2.4.8.2. Encodeur binaire 8x3.....	23

2.4.9. Transcodeur .....	25
2.4.9.1. Transcodeur 7-segments.....	25
2.4.9.2. Transcodeur BCD/EXESS3.....	28

## Abréviation

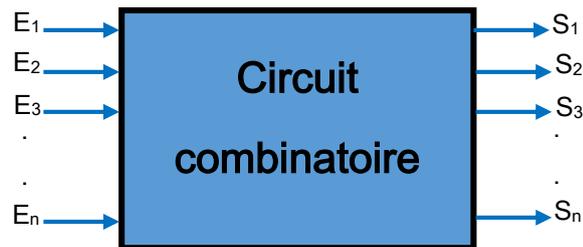
DA	: Demi-Additionneur
ADD	: Additionneur complet
MUX	: Multiplexeur
DEMUX	: Démultiplexeur
CLk / H ou h	: Clock / Horloge
RAM	: Random Access Memory (mémoires vives)
ROM	: Read Only Memory (mémoires mortes)
FSM	: Finit State Machine
MEF	: Machine à Etats Finis
CI	: Circuit Intégré
CMOS	: Complementary Metal Oxide Semiconductor
TTL	: Transistor Transistor Logic

## Chapitre II : Conception des Circuits Combinatoires

### 2.1 Définition

Un circuit combinatoire est défini par une ou plusieurs fonctions logiques. Un circuit combinatoire est un circuit numérique dont les sorties dépendent uniquement des entrées.

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



C'est possible d'utiliser des circuits combinatoires pour réaliser d'autres circuits plus complexes. Exemple de Circuits combinatoires : Demi Additionneur, Additionneur complet, Comparateur, Multiplexeur, Démultiplexeur, Encodeur et Décodeur...etc.

### 2.2. Synthèse d'une fonction combinatoire

Pour faire l'étude et la réalisation d'un circuit combinatoire il faut suivre les étapes suivantes :

- 1- Il faut bien comprendre le fonctionnement du système.
- 2- Il faut définir les variables d'entrée.
- 3- Il faut définir les variables de sortie.
- 4- Etablir la table de vérité.
- 5- Ecrire les équations algébriques des sorties (à partir de la table de vérité).
- 6- Effectuer des simplifications (algébrique ou par Karnaugh).
- 7- Faire le schéma avec un minimum de portes logiques.

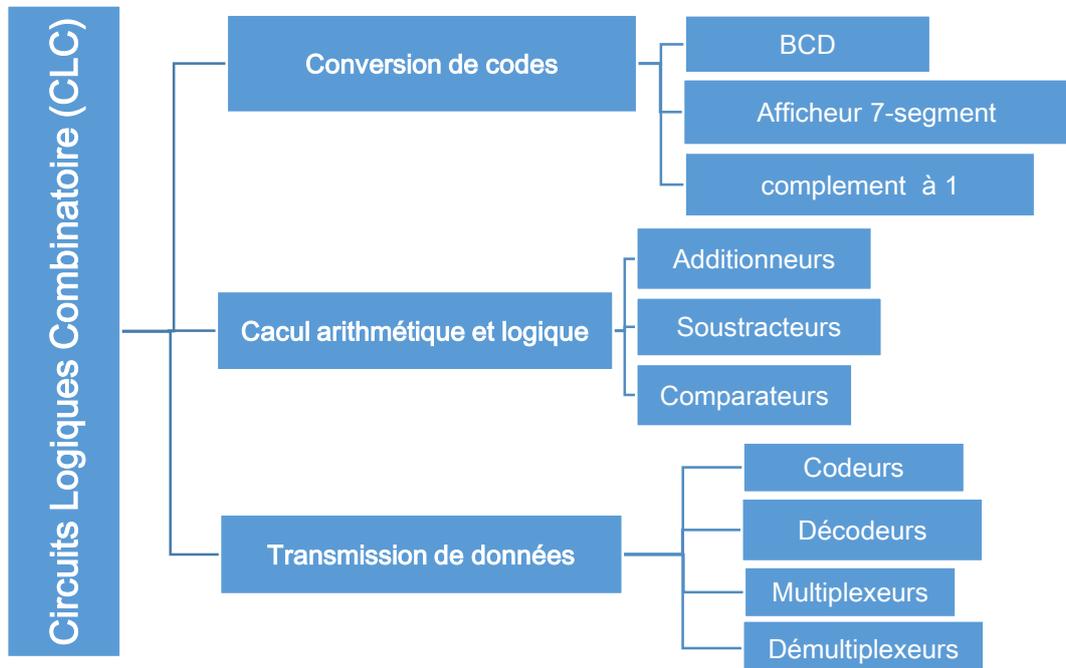
La synthèse d'un circuit combinatoire signifie la détermination d'un logigramme simplifier à partir de la définition d'une fonction logique. D'une façon générale, la démarche est la suivante :

- a- Schéma symbolique : construire son schéma symbolique en identifier les entrées et les sorties (In/Out) de la fonction.
- b- Table de vérité : construire sa table de vérité.
- c- Equations de sorties : extraire les équations de sorties à partir de la table de vérité, simplifier les fonctions de sortie via les théorèmes de l'algèbre de Boole ou les tables de Karnaugh
- d- Schéma logique : Dessiner le schéma du circuit logique à l'aide d'opérateurs (NOT, AND, OR, NAND, NOR)

### 2.3. Classification

On distingue au moins 3 classes de circuits logiques combinatoires :

- Les circuits de calcul arithmétiques et logiques ;
- Les circuits d'aiguillage et de transmission de données.



➤ Les circuits de calcul arithmétiques et logiques : Ce sont généralement des circuits logiques combinatoires permettant d'effectuer des calculs arithmétiques (addition, soustraction, multiplication) sur des entiers ou des nombres en virgule flottantes et des opérations logiques comme des négations, des ET, des OU ou des OU-Exclusifs. On les trouve le plus souvent dans les unités de calculs des ordinateurs communément appelées UAL ou unité arithmétique et logique.

➤ Les circuits de transmission de données : C'est un groupe de circuits permettant d'aiguiller les informations (données) binaires à travers des lignes électriques (souvent appelé BUS) d'une source (une petite mémoire appelée registre ou des capteurs, interrupteurs ou boutons poussoirs) vers une destination (registre ou un afficheur par exemple). Le décodeur, le multiplexeur en sont des exemples.

➤ Les convertisseurs de code Les nombres sont habituellement codés sous une forme ou une autre afin de les représenter ou de les utiliser au besoin. Par exemple, un nombre 'sept' est codé en décimal à l'aide du symbole  $(7)_{10}$ . Ce nombre est affiché sur votre calculatrice en se servant du codage 7 segments, mais au sein de l'unité de calcul de votre calculatrice, ce même nombre est codé en général en complément à 2. Bien que les ordinateurs numériques traitent tous des nombres binaires, il y a des situations où la représentation binaire naturelle des nombres n'est pas pratique ou est inefficace ce qui nécessite des codes plus appropriés. Cette situation fait cohabiter, dans une même machine, divers codes pour représenter une même information. Des circuits de conversion d'un code vers un autre sont donc utilisés.

## 2.4. Circuits combinatoires usuelles

### 2.4.1. Additionneur

L'addition est une opération très courante dans un microprocesseur. Outre dans l'unité arithmétique, elle sert pour incrémenter le compteur de programme et pour les calculs d'adresses. il existe de multiples façons de construire des additionneurs efficaces en temps et en nombre de portes logiques utilisées [1].

#### 2.4.1.1. Demi-Additionneur

- Le demi additionneur est un circuit combinatoire qui permet de réaliser la somme arithmétique de deux nombres A et B chacun sur un bit.
- A la sortie on va avoir la somme S et la retenue R (Carry).

a-Schéma symbolique :



En binaire l'addition sur un seul bit se fait de la manière suivante :

$$\left\{ \begin{array}{l} 0+0=0 \text{ retenue}=0 \\ 0+1=1 \text{ retenue}=0 \\ 1+0=1 \text{ retenue}=0 \\ 1+1=0 \text{ retenue}=1 \end{array} \right.$$

b-Table de vérité :

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

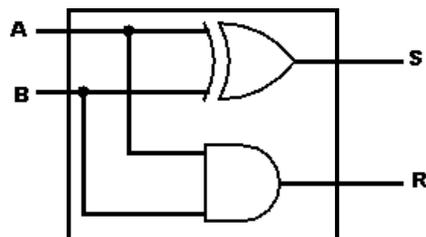
c- Equations de sortie :

De la table de vérité on trouve :

$$S = \bar{A}B + A\bar{B} = A \oplus B$$

$$R = AB$$

d-Schéma logique



Demi-additionneur à 1 bit

### 2.4.1.2. Additionneur complet

- En binaire lorsque on fait une addition il faut tenir en compte de la retenue entrante.

$$\begin{array}{rcccccc}
 R_4 & R_3 & R_2 & R_1 & R_0 = 0 & & R_{i-1} \\
 & A_4 & A_3 & A_2 & A_1 & & A_i \\
 + & B_4 & B_3 & B_2 & B_1 & & + B_i \\
 \hline
 R_4 & S_4 & S_3 & S_2 & S_1 & & \hline
 & & & & & & R_i & S_i
 \end{array}$$

Exemple d'un additionneur complet 1 bit

- L'additionneur complet un bit possède 3 entrées :

$A_i$  : le premier nombre sur un bit.

$B_i$  : le deuxième nombre sur un bit.

$R_{i-1}$  : la retenue entrante sur un bit.

- Il possède deux sorties :

$S_i$  : la somme

$R_i$  : la retenue sortante

#### a-Schéma symbolique



#### b-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

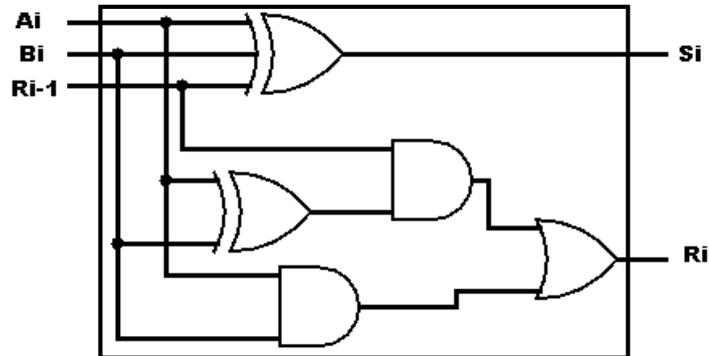
#### c- Equations de sortie :

De la table de vérité on trouve :

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

$$\begin{aligned}
 R_i &= \bar{A}_i B_i R_{i-1} + A_i \bar{B}_i R_{i-1} + A_i B_i \bar{R}_{i-1} + A_i B_i R_{i-1} \\
 &= R_{i-1} (\bar{A}_i B_i + A_i \bar{B}_i) + A_i B_i (\bar{R}_{i-1} + R_{i-1}) \\
 &= R_{i-1} (A_i \oplus B_i) + A_i B_i
 \end{aligned}$$

d-Schéma logique :



Additionneur complet à 1 bit

### 2.4.1.3. Additionneur à 4 bits

- Un additionneur à 4 bits est un circuit qui permet de faire l'addition de deux nombres A et B de 4 bits chacun : A(A4A3A2A1) et B(B4B3B2B1) En plus il tient en compte de la retenue entrante.
- En sortie on va avoir le résultat sur 4 bits ainsi que la retenue (5 bits en sortie)
- Donc au total le circuit possède 9 entrées et 5 sorties.
- Avec 9 entrées on a  $2^9 = 512$  combinaisons
- Il faut trouver une solution plus facile et plus efficace pour concevoir ce circuit ?
- Lorsque on fait l'addition en binaire, on additionne bit par bit en commençant à partir du poids faible et à chaque fois on propage la retenue sortante au bit du rang supérieur. L'addition sur un bit peut se faire par un additionneur complet sur 1 bit.

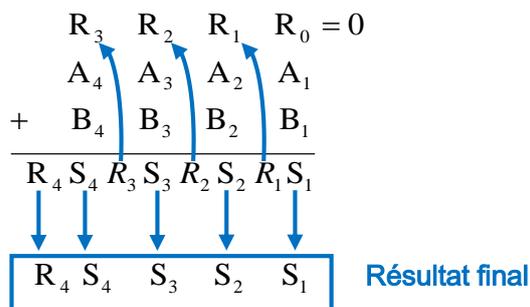
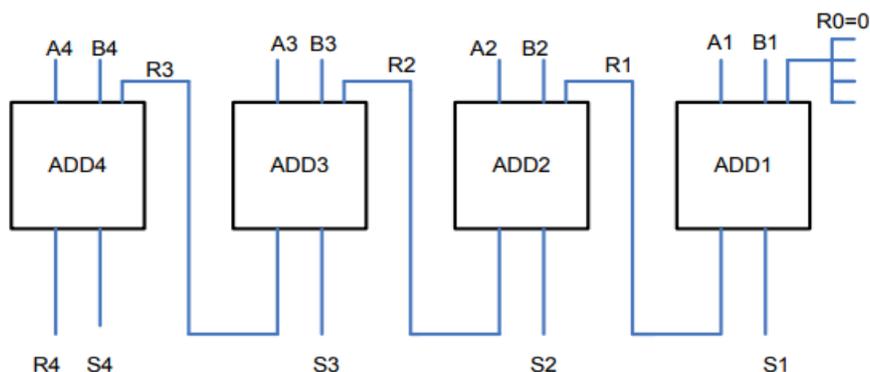


Schéma logique d'un additionneur complet à 4 bits



**Additionneur n-bits.** Pour effectuer l'addition de deux nombres de n bits, il suffit de chaîner entre eux n additionneurs 1-bit complets. La retenue est ainsi propagée d'un additionneur à l'autre. Un tel additionneur est appelé un additionneur série. Bien que tous les chiffres des deux nombres de n-bits X et Y soient disponibles simultanément au début du calcul, à t=0, le temps de calcul est déterminé par la propagation de la retenue à travers les n additionneurs 1-bit [2].

### 2.4.2. Soustracteur

Pour une soustraction de A et B, on peut adopter la même approche que pour l'addition. On commence par la définition de l'opérateur binaire de base et on l'utilise pour réaliser des soustractions de nombres binaires. En pratique, se pose le problème de la représentation des nombres signés dans le cas où  $B > A$ . Pour résoudre ce problème, on convient d'une représentation des nombres négatifs, la soustraction est alors ramenée à une addition. La représentation généralement utilisée est celle du complément vrai ou complément à 2.

#### 2.4.2.1. Demi-soustracteur

- Le demi soustracteur est un circuit combinatoire qui permet de réaliser la différence arithmétique de deux nombres A et B chacun sur un bit.
- A la sortie on va avoir la différence D et la retenue R.

**a-Schéma symbolique :**



En binaire l'addition sur un seul bit se fait de la manière suivante :

$$\left\{ \begin{array}{l} 0 - 0 = 0 \text{ retenue} = 0 \\ 0 - 1 = 1 \text{ retenue} = 1 \\ 1 - 0 = 1 \text{ retenue} = 0 \\ 1 - 1 = 0 \text{ retenue} = 0 \end{array} \right.$$

**b-Table de vérité :**

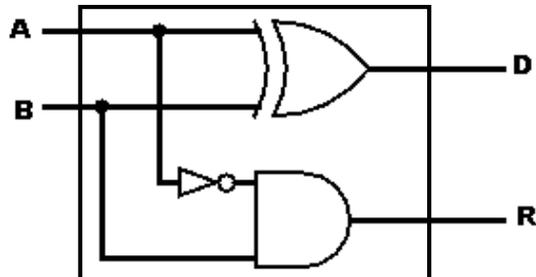
A	B	D	R
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

**c- Equations de sortie :**

De la table de vérité on trouve :

$$\begin{cases} D = \bar{A}B + A\bar{B} = A \oplus B \\ R = \bar{A}B \end{cases}$$

**d-schéma logique :**



**Demi-soustracteur à 1 bit**

**2.4.2.2. Soustracteur complet**

Pour obtenir un soustracteur binaire complet il faut prendre en compte l'éventuelle retenue précédente  $R_{i-1}$ .

- En binaire lorsque on fait une différence il faut tenir en compte de la retenue entrante.

$$\begin{array}{r} R_4 \quad R_3 \quad R_2 \quad R_1 \quad R_0 = 0 \\ A_4 \quad A_3 \quad A_2 \quad A_1 \\ - \quad B_4 \quad B_3 \quad B_2 \quad B_1 \\ \hline D_4 \quad D_3 \quad D_2 \quad D_1 \end{array} \qquad \begin{array}{r} R_{i-1} \\ A_i \\ - \quad B_i \\ \hline R_i \quad D_i \end{array}$$

Exemple d'un soustracteur complet 1 bit

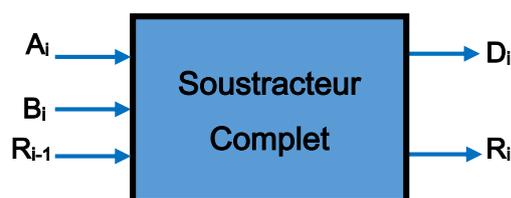
- Le soustracteur complet un bit possède 3 entrées :

- $A_i$  : le premier nombre sur un bit.
- $B_i$  : le deuxième nombre sur un bit.
- $R_{i-1}$  : la retenue entrante sur un bit.

- Il possède deux sorties :

- $D_i$  : la différence
- $R_i$  : la retenue sortante

**a-Schéma symbolique**



**b-Table de vérité :**

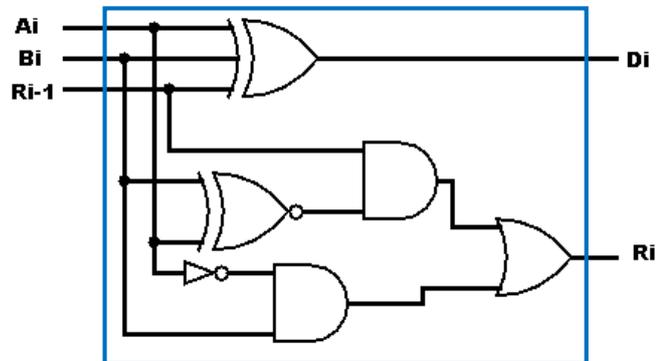
A <sub>i</sub>	B <sub>i</sub>	R <sub>i-1</sub>	D <sub>i</sub>	R <sub>i</sub>
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

**c- Equations de sortie :**

$$\begin{aligned}
 D_i &= \bar{A}_i \bar{B}_i R_{i-1} + \bar{A}_i B_i \bar{R}_{i-1} + A_i \bar{B}_i \bar{R}_{i-1} + A_i B_i R_{i-1} \\
 &= \bar{A}_i (\bar{B}_i R_{i-1} + B_i \bar{R}_{i-1}) + A_i (\bar{B}_i \bar{R}_{i-1} + B_i R_{i-1}) \\
 &= \bar{A}_i (B_i \oplus \bar{R}_{i-1}) + A_i (\bar{B}_i \oplus R_{i-1}) \\
 &= A_i \oplus B_i \oplus R_{i-1}
 \end{aligned}$$

$$\begin{aligned}
 R_i &= \bar{A}_i \bar{B}_i R_{i-1} + \bar{A}_i B_i \bar{R}_{i-1} + \bar{A}_i B_i R_{i-1} + A_i B_i R_{i-1} \\
 &= R_{i-1} (\bar{A}_i \bar{B}_i + A_i B_i) + \bar{A}_i B_i (\bar{R}_{i-1} + R_{i-1}) \\
 &= R_{i-1} (A_i \oplus B_i) + \bar{A}_i B_i
 \end{aligned}$$

**d-Schéma logique :**



**Soustracteur complet à 1 bit**

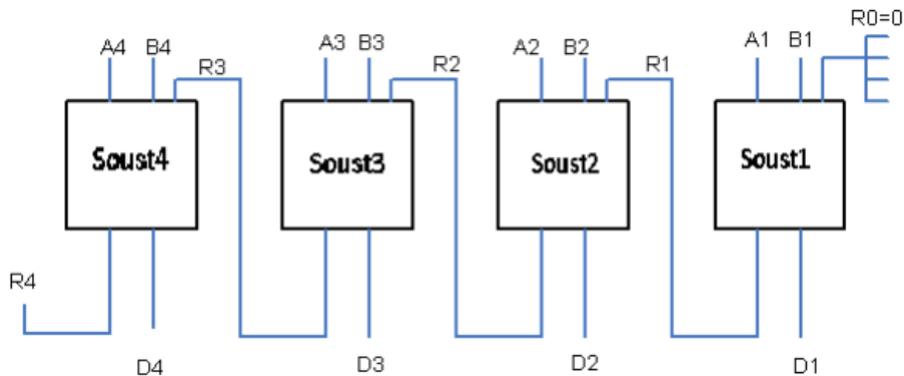
**2.4.2.3. Soustracteur à 4 bits**

- Un soustracteur à 4 bits est un circuit qui permet de faire la différence de deux nombres A et B de 4 bits chacun : A(A<sub>4</sub>A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>) et B(B<sub>4</sub>B<sub>3</sub>B<sub>2</sub>B<sub>1</sub>) En plus il tient en compte de la retenue entrante
- En sortie on va avoir le résultat sur 4 bits ainsi que la retenue (5 bits en sortie)
- Donc au total le circuit possède 9 entrées et 5 sorties.
- Avec 9 entrées on a 2<sup>9</sup> =512 combinaisons
- Il faut trouver une solution plus facile et plus efficace pour concevoir ce circuit ?

• Lorsque on fait la soustraction en binaire, on soustrait bit par bit en commençant à partir du poids faible et à chaque fois on propage la retenue sortante au bit du rang supérieur. La soustraction sur un bit peut se faire par un soustracteur complet sur 1 bit.

$$\begin{array}{r}
 A_4 \quad A_3 \quad A_2 \quad A_1 \\
 - \quad B_4 \quad B_3 \quad B_2 \quad B_1 \\
 - \quad R_3 \quad R_2 \quad R_1 \quad R_0 = 0 \\
 \hline
 D_4 \quad R_3 D_3 \quad R_2 D_2 \quad R_1 D_1 \\
 \hline
 \boxed{D_4 \quad D_3 \quad D_2 \quad D_1} \quad \text{Résultat final}
 \end{array}$$

Schéma logique d'un soustracteur complet à 4 bits



### 2.4.3. Additionneur/Soustracteur

Le circuit de 5 bits ci-dessous effectue une somme ou une différence suivant la valeur de la commande  $C_{md}$ , Figure 3. Si  $C_{md}$  vaut 0, le circuit calcule la somme  $A+B$ . Si, au contraire,  $C_{md}$  vaut 1, le circuit calcule la différence  $A-B$ . En effet, chacune des portes Xor effectue la négation ou non d'une entrée  $B_i$  suivant la valeur de  $C_{md}$ .

### 2.4.3. Additionneur/Soustracteur

Le circuit de 5 bits ci-dessous effectue une somme ou une différence suivant la valeur de la commande  $C_{md}$ , Figure 3. Si  $C_{md}$  vaut 0, le circuit calcule la somme  $A+B$ . Si, au contraire,  $C_{md}$  vaut 1, le circuit calcule la différence  $A-B$ . En effet, chacune des portes Xor effectue la négation ou non d'une entrée  $B_i$  suivant la valeur d'entrée de sélection.

### 2.4.4. Comparateur

#### 2.4.4.1. Comparateur à 1 bit

• C'est un circuit combinatoire qui permet de comparer entre deux nombres binaire A et B. Exemple d'un comparateur à un bit.

• Il possède 2 entrées :

**A** : sur un bit

**B** : sur un bit

• Il possède 3 sorties

**fe** : égalité ( A=B)

**fi** : inférieur ( A < B)

**fs** : supérieur ( A > B)

a-Schéma symbolique :



b-Table de vérité :

A	B	fs	fe	fi
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

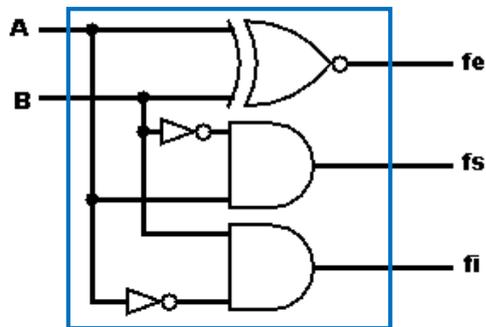
c- Equations de sortie :

$$f_s = A\bar{B}$$

$$f_i = \bar{A}B$$

$$f_e = \bar{A}\bar{B} + AB = \overline{A \oplus B}$$

d-Schéma logique :



Comparateur à 1 bit

#### 2.4.4.2. Comparateur à 2 bits

• Il permet de faire la comparaison entre deux nombres A(A<sub>2</sub>A<sub>1</sub>) et B(B<sub>2</sub>B<sub>1</sub>) chacun sur deux bits.

a-Schéma symbolique :



b-Table de vérité :

A <sub>2</sub>	A <sub>1</sub>	B <sub>2</sub>	B <sub>1</sub>	f <sub>s</sub>	f <sub>e</sub>	f <sub>i</sub>
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

c- Equations de sortie :

$$f_s = \bar{A}_2 A_1 \bar{B}_2 \bar{B}_1 + A_2 \bar{A}_1 \bar{B}_2 \bar{B}_1 + A_2 \bar{A}_1 \bar{B}_2 B_1 + A_2 A_1 \bar{B}_2 \bar{B}_1 + A_2 A_1 \bar{B}_2 B_1 + A_2 A_1 B_2 \bar{B}_1$$

$$= A_2 \bar{B}_2 + (A_2 \oplus B_2) A_1 \bar{B}_1$$

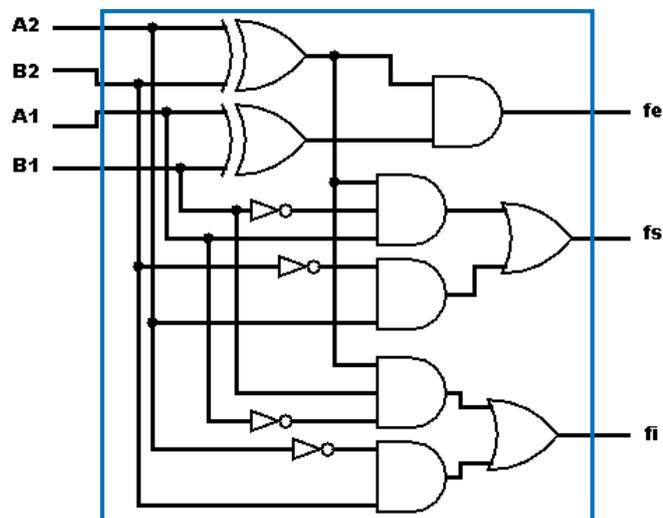
$$f_i = \bar{A}_2 \bar{A}_1 \bar{B}_2 B_1 + \bar{A}_2 \bar{A}_1 B_2 \bar{B}_1 + \bar{A}_2 \bar{A}_1 B_2 B_1 + \bar{A}_2 A_1 B_2 \bar{B}_1 + \bar{A}_2 A_1 B_2 B_1 + A_2 \bar{A}_1 B_2 B_1$$

$$= \bar{A}_2 B_2 + (\bar{A}_2 \oplus B_2) \bar{A}_1 B_1$$

$$f_e = \bar{A}_2 \bar{A}_1 \bar{B}_2 \bar{B}_1 + \bar{A}_2 A_1 \bar{B}_2 B_1 + A_2 \bar{A}_1 B_2 \bar{B}_1 + A_2 A_1 B_2 B_1$$

$$= (A_2 \oplus B_2)(A_1 \oplus B_1)$$

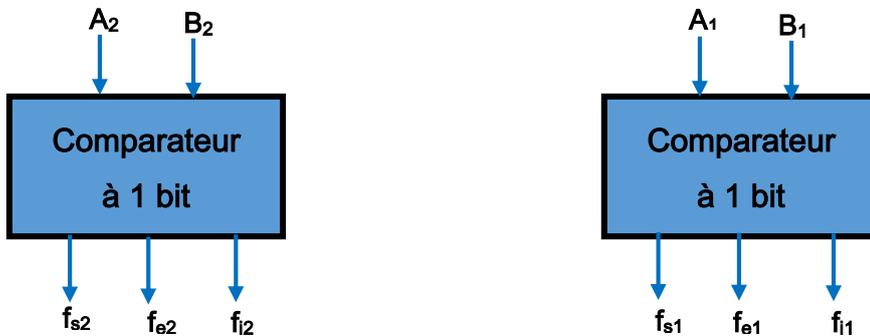
d-Schéma logique :



Comparateur à 2 bit

### 2.4.4.3. Comparateur 2 bits avec des comparateurs 1 bit

C'est possible de réaliser un comparateur 2 bits en utilisant des comparateurs 1 bit et des portes logiques. Il faut utiliser un comparateur pour comparer les bits du poids faible et un autre pour comparer les bits du poids fort. Il faut combiner entre les sorties des deux comparateurs utilisés pour réaliser les sorties du comparateur final.

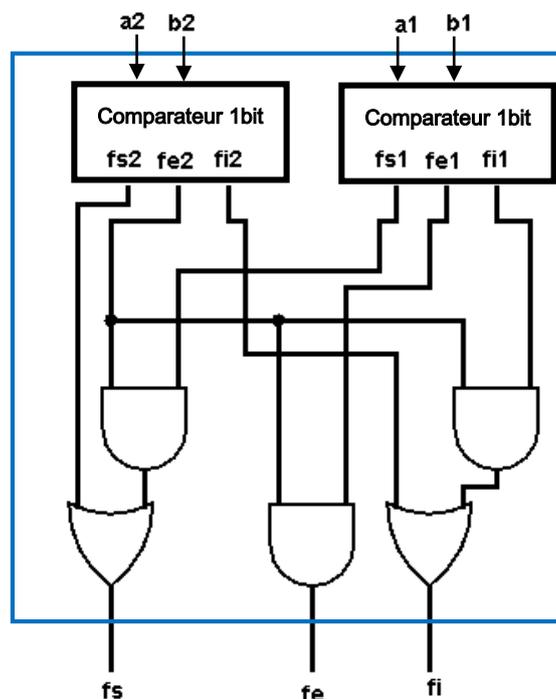


$$A = B \text{ si } A_2 = B_2 \text{ et } A_1 = B_1 \longrightarrow f_e = (\overline{A_2 \oplus B_2})(\overline{A_1 \oplus B_1}) = f_{e2}f_{e1}$$

$$A > B \text{ si } A_2 > B_2 \text{ ou } (A_2 = B_2 \text{ et } A_1 > B_1) \longrightarrow f_s = A_2\overline{B_2} + (\overline{A_2 \oplus B_2})A_1\overline{B_1} = f_{s2} + f_{e2}f_{s1}$$

$$A < B \text{ si } A_2 < B_2 \text{ ou } (A_2 = B_2 \text{ et } A_1 < B_1) \longrightarrow f_i = \overline{A_2}B_2 + (\overline{A_2 \oplus B_2})\overline{A_1}B_1 = f_{i2} + f_{e2}f_{i1}$$

### Schéma logique d'un comparateur à deux entrées à deux bits



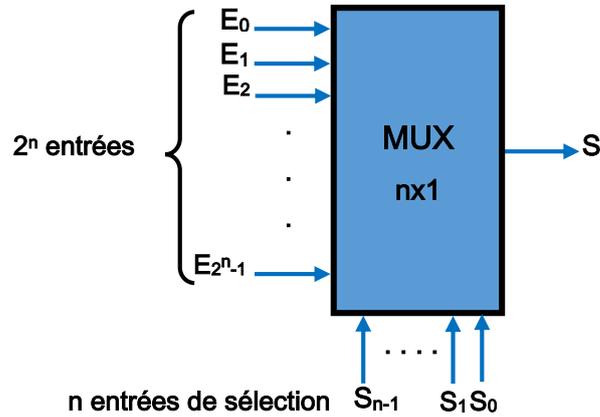
### 2.4.5. Multiplexeur

Un multiplexeur possède plusieurs entrées et une seule sortie. Il agit comme un sélecteur de données en orientant vers sa sortie la donnée présente sur l'une de ses entrées [3]. Un multiplexeur

est un circuit combinatoire qui permet de sélectionner une information (1 bit) parmi  $2^n$  valeurs en entrée.

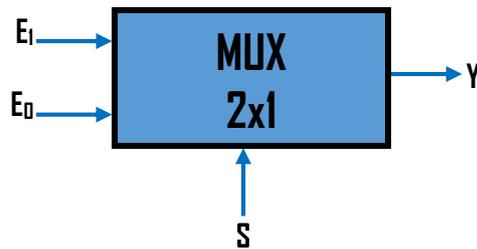
Il possède :

- $2^N$  entrées d'information
- Une seule sortie
- N entrées de sélection (commandes)



#### 2.4.5.1. Multiplexeur 2x1

a-Schéma symbolique:



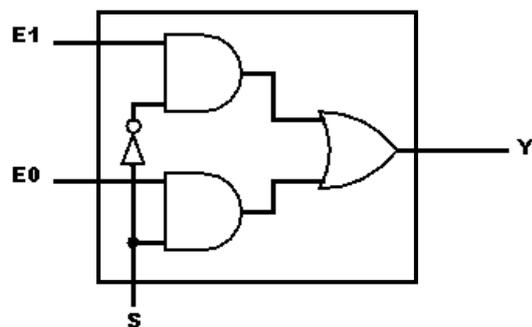
b-Table de vérité :

S	Y
0	E <sub>0</sub>
1	E <sub>1</sub>

c- Equations de sortie :

$$Y = \bar{S}E_0 + SE_1$$

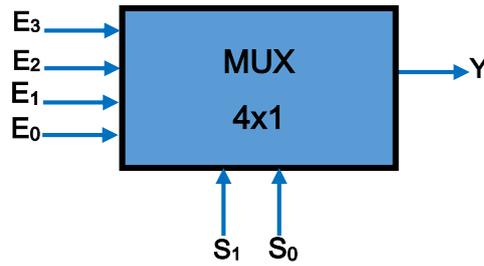
d-Schéma logique :



Multiplexeur 2x1

### 2.4.5.2. Multiplexeur 4X1

a-Schéma symbolique :



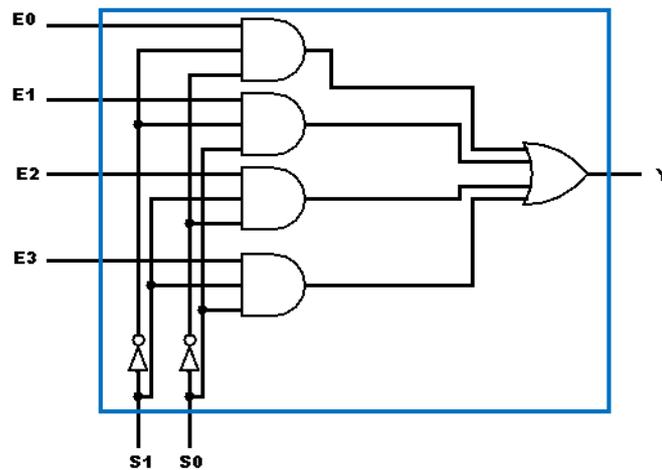
b-Table de vérité :

S <sub>1</sub>	S <sub>0</sub>	Y
0	0	E <sub>0</sub>
0	1	E <sub>1</sub>
1	0	E <sub>2</sub>
1	1	E <sub>3</sub>

c- Equations de sortie :

$$Y = \bar{S}_1 \bar{S}_0 E_0 + \bar{S}_1 S_0 E_1 + S_1 \bar{S}_0 E_2 + S_1 S_0 E_3$$

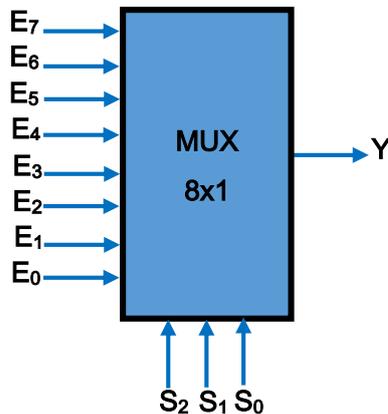
d-Schéma logique :



Multiplexeur 4x1

### 2.4.5.3. Multiplexeur 8X1

a-schéma symbolique :



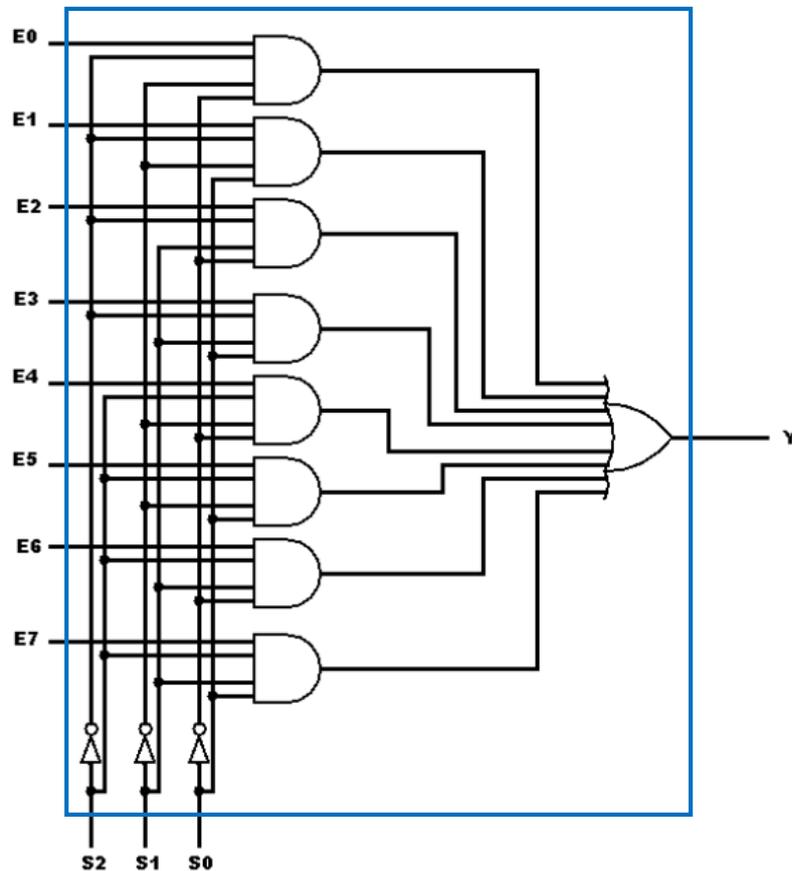
b-Table de vérité :

S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	Y
0	0	0	E <sub>0</sub>
0	0	1	E <sub>1</sub>
0	1	0	E <sub>2</sub>
0	1	1	E <sub>3</sub>
1	0	0	E <sub>4</sub>
1	0	1	E <sub>5</sub>
1	1	0	E <sub>6</sub>
1	1	1	E <sub>7</sub>

c- équations de sortie :

$$Y = \bar{S}_2\bar{S}_1\bar{S}_0E_0 + \bar{S}_2\bar{S}_1S_0E_1 + \bar{S}_2S_1\bar{S}_0E_2 + \bar{S}_2S_1S_0E_3 + S_2\bar{S}_1\bar{S}_0E_4 + S_2\bar{S}_1S_0E_5 + S_2S_1\bar{S}_0E_6 + S_2S_1S_0E_7$$

d-schéma logique :



Multiplexeur 8X1

Pour réaliser une fonction logique par un Mux on effectue les opérations suivantes [4] :

- On écrit l'équation de la fonction logique sous la première forme canonique (on détermine le nombre d'entrée).
- On écrit l'équation du Mux caractérisé par le nombre d'entrée d'adresse (nombre d'entrée d'adresse = nombre d'entrées de la fonction).
- Identification de deux équations.

Exemple : **Additionneur complet avec des multiplexeurs 8X1**

• Nous avons besoin d'utiliser deux multiplexeurs : Le premier pour réaliser la fonction de la somme et l'autre pour donner la retenue.

**b-Table de vérité :**

A <sub>i</sub>	B <sub>i</sub>	R <sub>i-1</sub>	S <sub>i</sub>	R <sub>i</sub>
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

**c- Equations de sortie :**

-La fonction de la somme

$$S_i = \bar{A}_i \bar{B}_i \bar{R}_{i-1}(0) + \bar{A}_i \bar{B}_i R_{i-1}(1) + \bar{A}_i B_i \bar{R}_{i-1}(1) + \bar{A}_i B_i R_{i-1}(0) + A_i \bar{B}_i \bar{R}_{i-1}(1) + A_i \bar{B}_i R_{i-1}(0) + A_i B_i \bar{R}_{i-1}(0) + A_i B_i R_{i-1}(0)$$

$$S_i = \bar{S}_2 \bar{S}_1 \bar{S}_0(E_0) + \bar{S}_2 \bar{S}_1 S_0(E_1) + \bar{S}_2 S_1 \bar{S}_0(E_2) + \bar{S}_2 S_1 S_0(E_3) + S_2 \bar{S}_1 \bar{S}_0(E_4) + S_2 \bar{S}_1 S_0(E_5) + S_2 S_1 \bar{S}_0(E_6) + S_2 S_1 S_0(E_7)$$

$$S_i = \bar{S}_2 \bar{S}_1 \bar{S}_0(0) + \bar{S}_2 \bar{S}_1 S_0(1) + \bar{S}_2 S_1 \bar{S}_0(1) + \bar{S}_2 S_1 S_0(0) + S_2 \bar{S}_1 \bar{S}_0(1) + S_2 \bar{S}_1 S_0(0) + S_2 S_1 \bar{S}_0(0) + S_2 S_1 S_0(1)$$

Tel que, on pose

$$S_2 = A_i$$

$$S_1 = B_i$$

$$S_0 = R_{i-1}$$

$$E_0 = 0, E_1 = 1, E_2 = 1, E_3 = 0, E_4 = 1, E_5 = 0, E_6 = 0, E_7 = 1$$

-La fonction de la retenue

$$R_i = \bar{A}_i \bar{B}_i \bar{R}_{i-1}(0) + \bar{A}_i \bar{B}_i R_{i-1}(0) + \bar{A}_i B_i \bar{R}_{i-1}(0) + \bar{A}_i B_i R_{i-1}(1) + A_i \bar{B}_i \bar{R}_{i-1}(0) + A_i \bar{B}_i R_{i-1}(1) + A_i B_i \bar{R}_{i-1}(1) + A_i B_i R_{i-1}(1)$$

$$R_i = \bar{S}_2 \bar{S}_1 \bar{S}_0(E_0) + \bar{S}_2 \bar{S}_1 S_0(E_1) + \bar{S}_2 S_1 \bar{S}_0(E_2) + \bar{S}_2 S_1 S_0(E_3) + S_2 \bar{S}_1 \bar{S}_0(E_4) + S_2 \bar{S}_1 S_0(E_5) + S_2 S_1 \bar{S}_0(E_6) + S_2 S_1 S_0(E_7)$$

$$R_i = \bar{S}_2 \bar{S}_1 \bar{S}_0(0) + \bar{S}_2 \bar{S}_1 S_0(0) + \bar{S}_2 S_1 \bar{S}_0(0) + \bar{S}_2 S_1 S_0(1) + S_2 \bar{S}_1 \bar{S}_0(0) + S_2 \bar{S}_1 S_0(1) + S_2 S_1 \bar{S}_0(1) + S_2 S_1 S_0(1)$$

Tel que, on pose

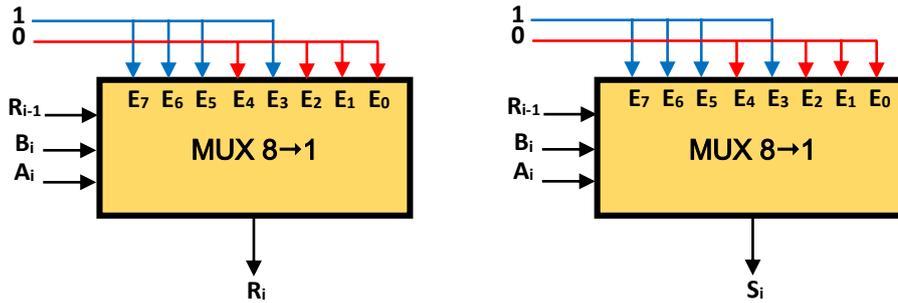
$$S_2 = A_i$$

$$S_1 = B_i$$

$$S_0 = R_{i-1}$$

$$E_0 = 0, E_1 = 1, E_2 = 1, E_3 = 0, E_4 = 1, E_5 = 0, E_6 = 0, E_7 = 1$$

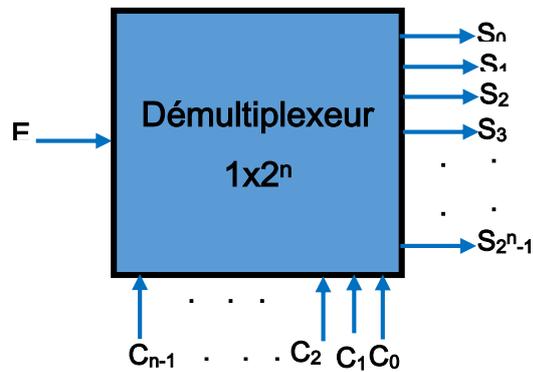
d-Schéma logique :



### 2.4.6. Démultiplexeurs

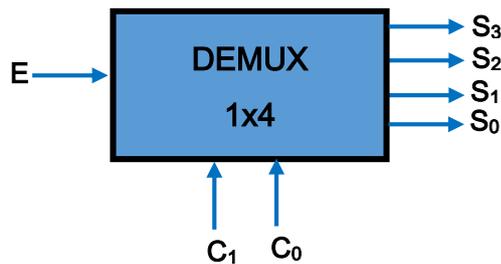
• Il joue le rôle inverse d'un multiplexeur, il permet de faire passer une information dans l'une des sorties selon les valeurs des entrées de commandes. Il possède :

- Une seule entrée
- $2^N$  sorties
- N entrées de sélection (commandes)



#### 2.4.6.1. Démultiplexeurs 1X4

a-schéma symbolique :



b-Table de vérité :

C <sub>1</sub>	C <sub>0</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>
0	0	0	0	0	E
0	1	0	0	E	0
1	0	0	E	0	0
1	1	E	0	0	0

c- équations de sortie :

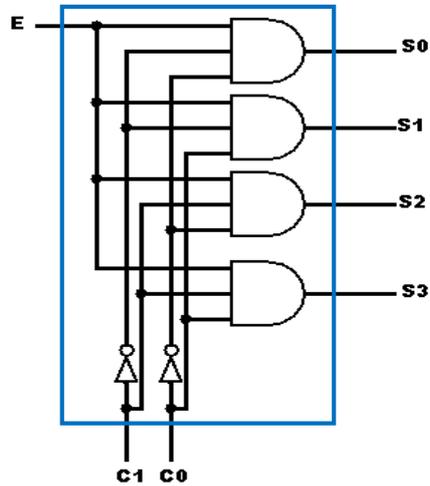
$$S_0 = \bar{C}_1 \bar{C}_0 E$$

$$S_1 = \bar{C}_1 C_0 E$$

$$S_2 = C_1 \bar{C}_0 E$$

$$S_3 = C_1 C_0 E$$

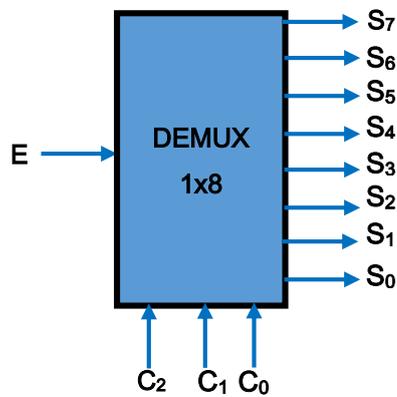
d-Schéma logique :



Démultiplexeurs 1X4

### 2.4.6.2. Démultiplexeurs 1X8

a-Schéma symbolique :



b-Table de vérité :

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

c- Equations de sortie :

$$S_0 = \bar{C}_2 \bar{C}_1 \bar{C}_0 E$$

$$S_1 = \bar{C}_2 \bar{C}_1 C_0 E$$

$$S_2 = \bar{C}_2 C_1 \bar{C}_0 E$$

$$S_3 = \bar{C}_2 C_1 C_0 E$$

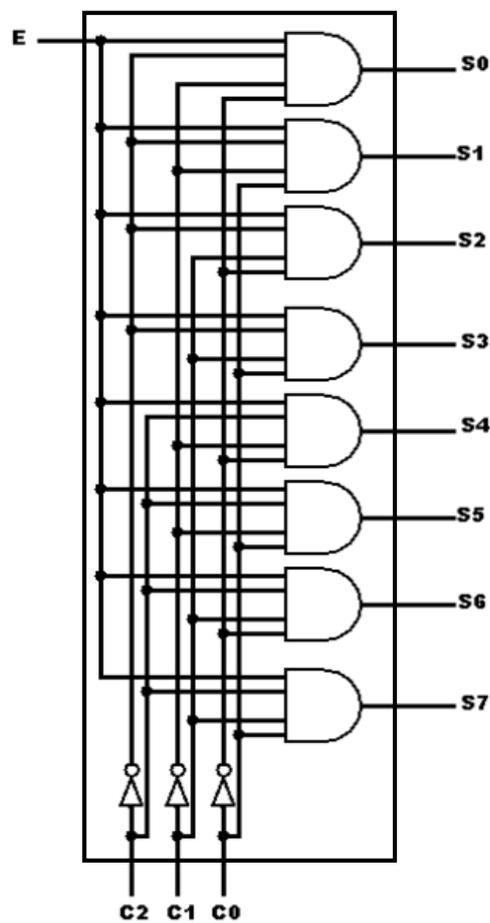
$$S_4 = C_2 \bar{C}_1 \bar{C}_0 E$$

$$S_5 = C_2 \bar{C}_1 C_0 E$$

$$S_6 = C_2 C_1 \bar{C}_0 E$$

$$S_7 = C_2 C_1 C_0 E$$

d-Schéma logique :

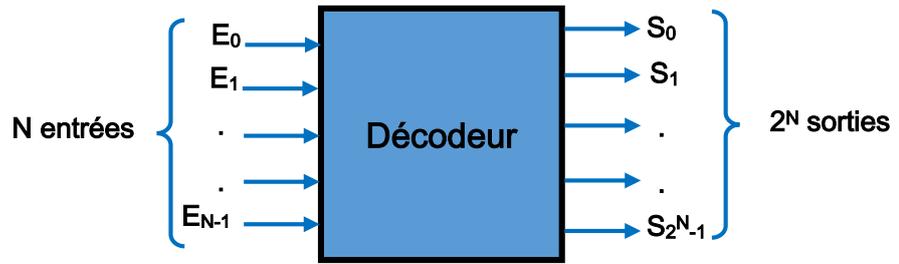


Démultiplexeurs 1X8

#### 2.4.7. Décodeur binaire

C'est un circuit combinatoire qui est constitué de :

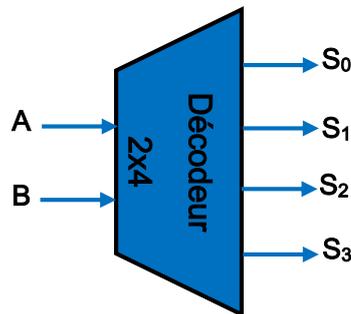
- N entrées de données
- $2^N$  sorties



- Pour chaque combinaison en entrée une seule sortie est active à la fois

### 2.4.7.1. Décodeur binaire 2x4

a-schéma symbolique :



b-Table de vérité :

A	B	$S_0$	$S_1$	$S_2$	$S_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

c- équations de sortie :

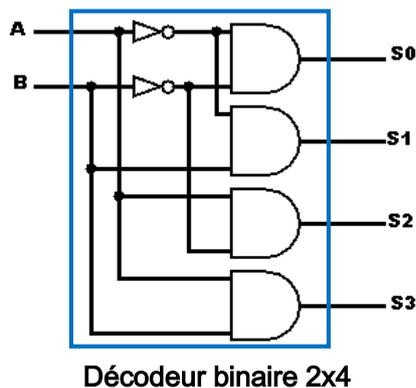
$$S_0 = \overline{A}\overline{B}$$

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

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

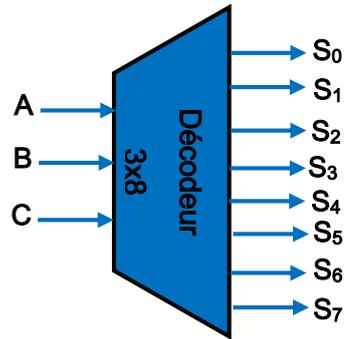
$$S_3 = AB$$

d-schéma logique :



### 2.4.7.2. Décodeur binaire 3X8

a-Schéma symbolique :



b-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

c- Equations de sortie :

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

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

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

$$S_3 = \overline{A}BC$$

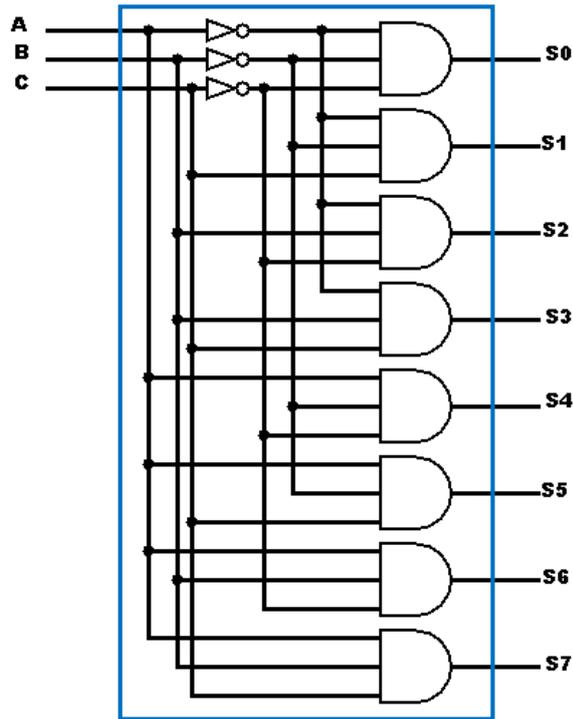
$$S_4 = A\overline{B}\overline{C}$$

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

$$S_6 = ABC$$

$$S_7 = \overline{A}BC$$

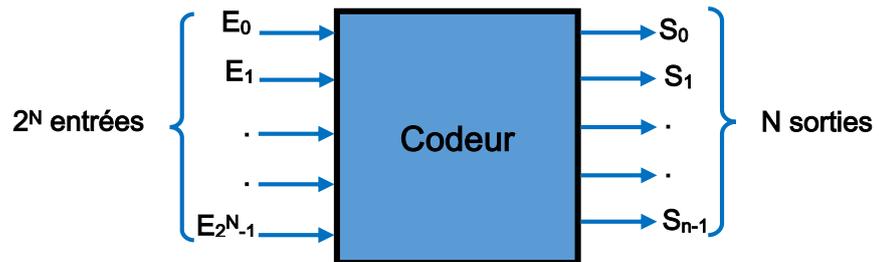
d-Schéma logique :



Décodeur binaire 3X8

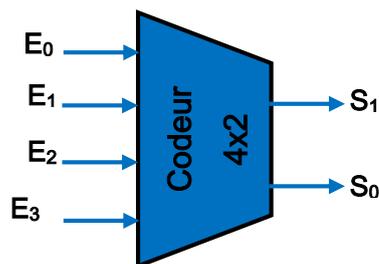
### 2.4.8. Encodeur binaire (codeur)

- Il joue le rôle inverse d'un décodeur. Il possède :
- $2^N$  entrées
- N sorties
- Pour chaque combinaison en entrée on va avoir son numéro (en binaire) à la sortie.



#### 2.4.8.1. Encodeur binaire 4x2

a-Schéma symbolique :



b-Table de vérité :

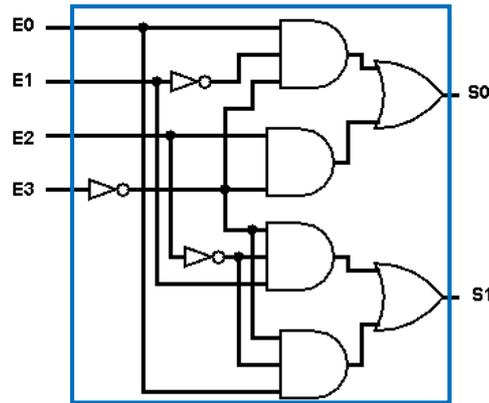
E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>	S <sub>1</sub>	S <sub>0</sub>
1	X	X	X	0	0
0	1	X	X	0	1
0	0	1	X	1	0
0	0	0	1	1	1

c- équations de sortie :

$$S_0 = \bar{E}_3 E_2 + \bar{E}_3 \bar{E}_2 \bar{E}_1 E_0 = \bar{E}_3 (E_2 + \bar{E}_2 \bar{E}_1 E_0) = \bar{E}_3 (E_2 + \bar{E}_1 E_0) = \bar{E}_3 E_2 + \bar{E}_3 \bar{E}_1 E_0$$

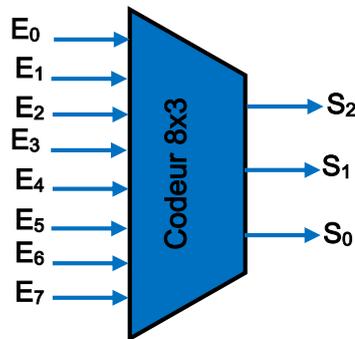
$$S_1 = \bar{E}_3 \bar{E}_2 E_1 + \bar{E}_3 \bar{E}_2 \bar{E}_1 E_0 = \bar{E}_3 \bar{E}_2 (E_1 + \bar{E}_1 E_0) = \bar{E}_3 \bar{E}_2 (E_1 + E_0) = \bar{E}_3 \bar{E}_2 E_1 + \bar{E}_3 \bar{E}_2 E_0$$

d-Schéma logique :



### 2.4.8.2. Encodeur binaire 8x3

a-Schéma symbolique :



b-Table de vérité :

E <sub>7</sub>	E <sub>6</sub>	E <sub>5</sub>	E <sub>4</sub>	E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>
1	X	X	X	X	X	X	X	0	0	0
0	1	X	X	X	X	X	X	0	0	1
0	0	1	X	X	X	X	X	0	1	0
0	0	0	1	X	X	X	X	0	1	1
0	0	0	0	1	X	X	X	1	0	0
0	0	0	0	0	1	X	X	1	0	1
0	0	0	0	0	0	1	X	1	1	0
0	0	0	0	0	0	0	1	1	1	1

**c- Equations de sortie :**

$$S_0 = \bar{E}_7 E_6 + \bar{E}_7 \bar{E}_6 \bar{E}_5 E_4 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 E_2 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 \bar{E}_2 \bar{E}_1 E_0$$

$$= \bar{E}_7 E_6 + \bar{E}_7 \bar{E}_5 E_4 + \bar{E}_7 \bar{E}_5 \bar{E}_3 E_2 + \bar{E}_7 \bar{E}_5 \bar{E}_3 \bar{E}_1 E_0$$

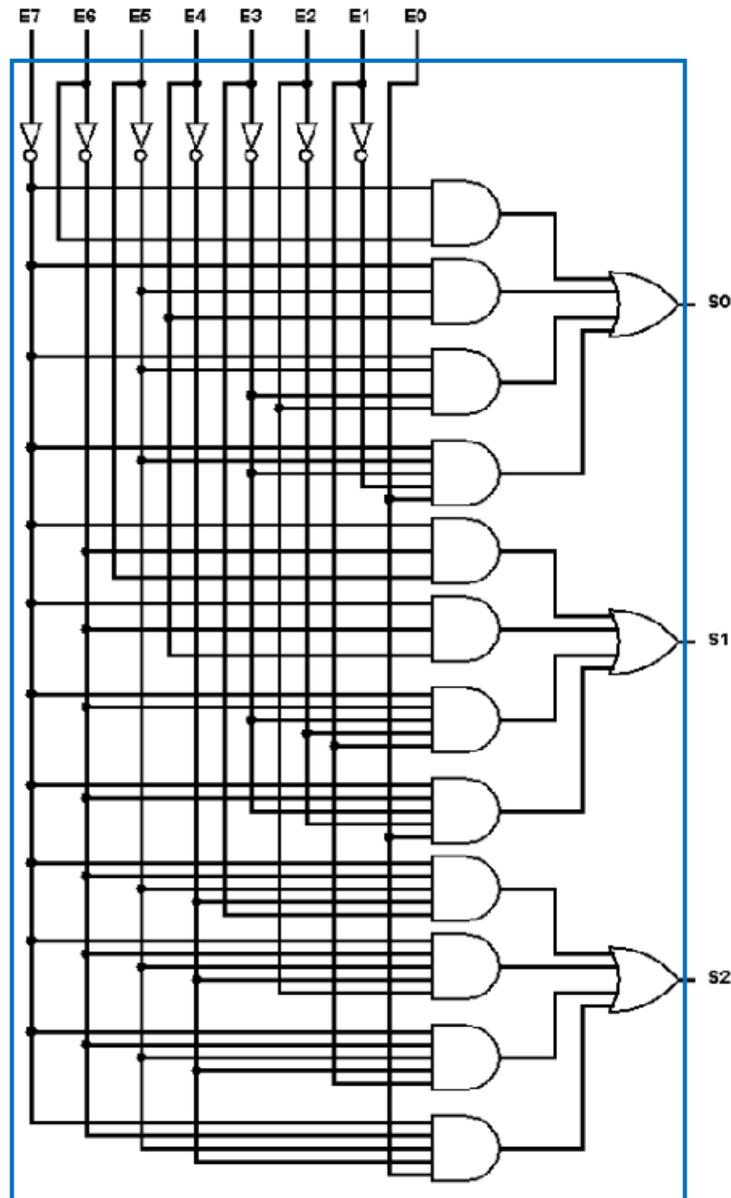
$$S_1 = \bar{E}_7 \bar{E}_6 E_5 + \bar{E}_7 \bar{E}_6 \bar{E}_5 E_4 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 \bar{E}_2 E_1 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 \bar{E}_2 \bar{E}_1 E_0$$

$$= \bar{E}_7 \bar{E}_6 E_5 + \bar{E}_7 \bar{E}_6 E_4 + \bar{E}_7 \bar{E}_6 \bar{E}_3 \bar{E}_2 E_1 + \bar{E}_7 \bar{E}_6 \bar{E}_3 \bar{E}_2 E_0$$

$$S_2 = \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 E_3 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 E_2 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 \bar{E}_2 E_1 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 \bar{E}_2 \bar{E}_1 E_0$$

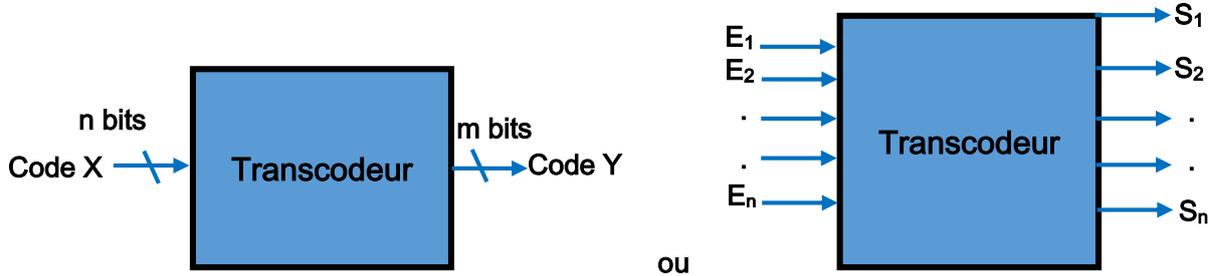
$$= \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 E_3 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 E_2 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 E_1 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 E_0$$

**d-Schéma logique :**



### 2.4.9. Transcodeur

- C'est un circuit combinatoire qui permet de transformer un code X (sur n bits) en entrée en un code Y (sur m bits) en sortie.
- Passage d'un code (Code X) à un code (Code Y)



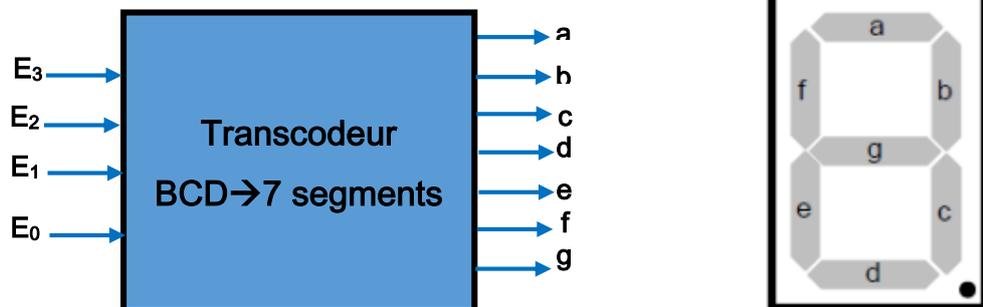
- Exemples de code : Binaire, binaire réfléchi, 7-segments, BCD, ...

#### 2.4.9.1. Transcodeur 7-segments

##### BCD $\Rightarrow$ code affichage chiffre (afficheur 7-segments)

Le transcodeur 7 segments accepte en entrée les 4 bits DCB (a0, a1, a2, a3) 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).

$\rightarrow$  a-schéma symbolique



Code binaire 0 à 9 Configuration alimentation des diodes (ou LCD)

$\rightarrow$  b-Table de vérité

Il y'a 6 combinaisons intitulés 10, 11, 12, 13, 14, 15. Les autres chiffres sont affichés comme suit :



Affichage	Table de vérité										
	Entrées				Sorties						
	E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

↪ c- Equations de sortie

Segment a

E <sub>3</sub> E <sub>2</sub> \ E <sub>1</sub> E <sub>0</sub>	00	01	11	10
00	1	0	X	1
01	0	1	X	1
11	1	1	X	X
10	1	0	X	X

Segment b

E <sub>3</sub> E <sub>2</sub> \ E <sub>1</sub> E <sub>0</sub>	00	01	11	10
00	1	1	X	1
01	1	0	X	1
11	1	1	X	X
10	1	0	X	X

$$a(E_3, E_2, E_1, E_0) = E_3 + E_2E_0 + \bar{E}_2E_1 + \bar{E}_2\bar{E}_0$$

$$b(E_3, E_2, E_1, E_0) = \bar{E}_2 + \bar{E}_1\bar{E}_0 + E_1E_0$$

Segment c

E <sub>3</sub> E <sub>2</sub> \ E <sub>1</sub> E <sub>0</sub>	00	01	11	10
00	1	1	X	1
01	1	1	X	1
11	1	1	X	X
10	0	1	X	X

Segment d

E <sub>3</sub> E <sub>2</sub> \ E <sub>1</sub> E <sub>0</sub>	00	01	11	10
00	1	0	X	1
01	0	1	X	0
11	1	0	X	X
10	1	1	X	X

$$c(E_3, E_2, E_1, E_0) = E_2 + \bar{E}_1 + E_0$$

$$d(E_3, E_2, E_1, E_0) = E_1\bar{E}_0 + \bar{E}_2E_1 + \bar{E}_2\bar{E}_0 + E_2\bar{E}_1E_0$$

Segment e

$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	1	0	X	1
01	0	0	X	1
11	0	0	X	X
10	1	1	X	X

$$e(E_3, E_2, E_1, E_0) = E_1\bar{E}_0 + E_3E_0 + \bar{E}_2\bar{E}_0$$

Segment f

$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	1	1	X	1
01	0	1	X	1
11	0	0	X	X
10	0	1	X	X

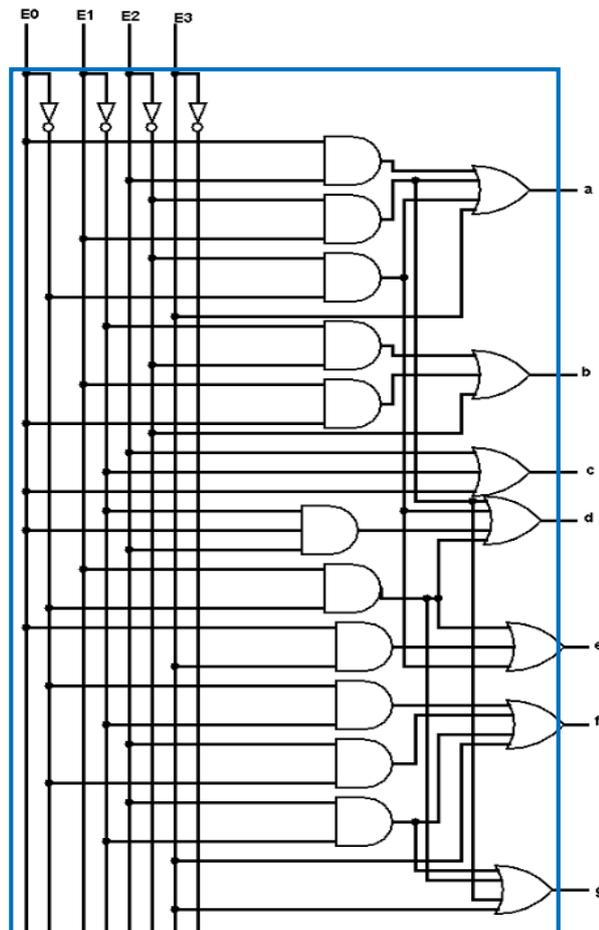
$$f(E_3, E_2, E_1, E_0) = E_3 + \bar{E}_1\bar{E}_0 + E_2\bar{E}_1 + E_2\bar{E}_0$$

Segment g

$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	0	1	X	1
01	0	1	X	1
11	1	0	X	X
10	1	1	X	X

$$g(E_3, E_2, E_1, E_0) = E_3 + E_2\bar{E}_1 + E_1\bar{E}_0 + \bar{E}_2E_1$$

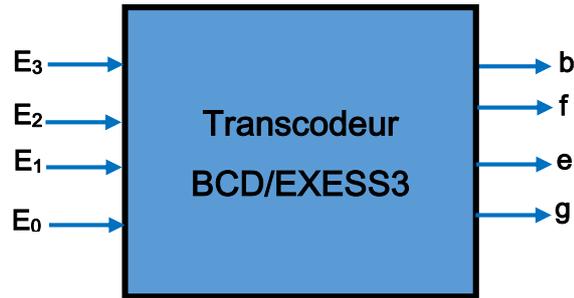
→ d- Schéma logique



Transcodeur BCD → 7 segments

2.4.9.2. Transcodeur BCD/EXESS3

→ a-schéma symbolique



→ b-Table de vérité :

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

→ c- Equations de sortie :

E <sub>3</sub> E <sub>2</sub> \ E <sub>1</sub> E <sub>0</sub>	00	01	11	10
00	0	0	X	1
01	0	1	X	1
11	0	1	X	X
10	0	1	X	X

E <sub>3</sub> E <sub>2</sub> \ E <sub>1</sub> E <sub>0</sub>	00	01	11	10
00	0	1	X	0
01	1	0	X	1
11	1	0	X	X
10	1	0	X	X

$$S_3(E_3, E_2, E_1, E_0) = E_3 + E_2E_1 + E_2E_0$$

$$S_2(E_3, E_2, E_1, E_0) = E_2\bar{E}_1\bar{E}_0 + \bar{E}_2E_0 + \bar{E}_2E_1$$

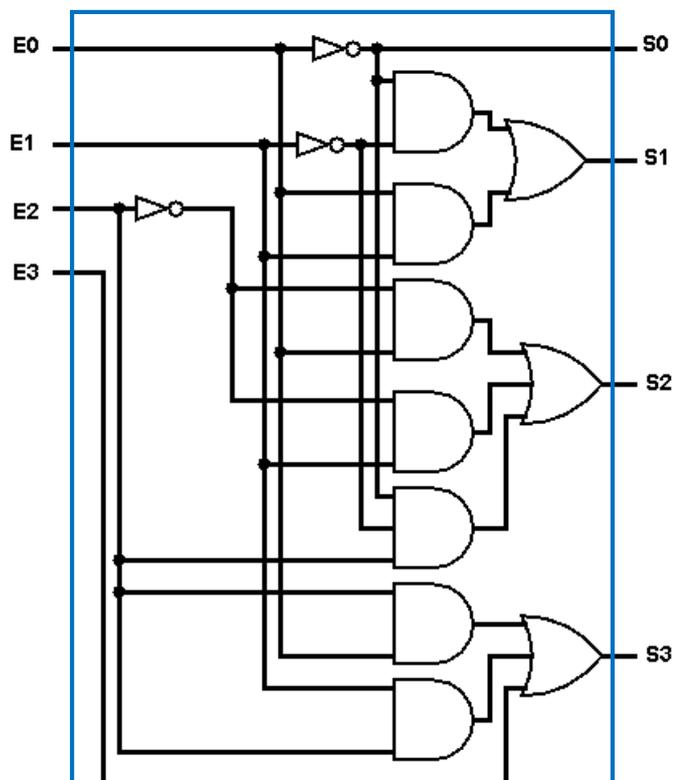
$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	1	1	X	1
01	0	0	X	0
11	1	1	X	X
10	0	0	X	X

$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	1	1	X	1
01	0	0	X	0
11	0	0	X	X
10	1	1	X	X

$$S_1(E_3, E_2, E_1, E_0) = \bar{E}_1\bar{E}_0 + E_1E_0$$

$$S_0(E_3, E_2, E_1, E_0) = \bar{E}_0$$

→ d- Schéma logique



Transcodeur BCD/EXESS3