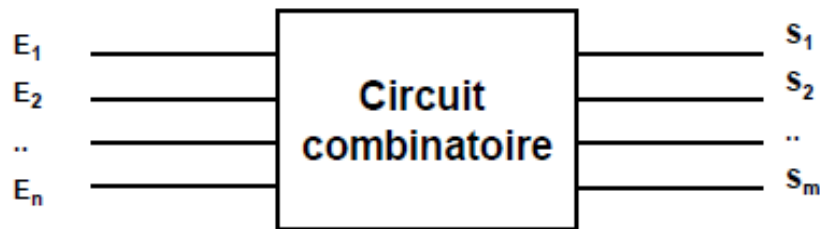


## Chapitre II : conception des Circuits Combinatoires

### 1. Définition :

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, Décodeur

### 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.

Synthèse d'un circuit combinatoire : déterminer un logigramme à partir de la définition d'une fonction logique.

D'une façon générale, la démarche est la suivante:

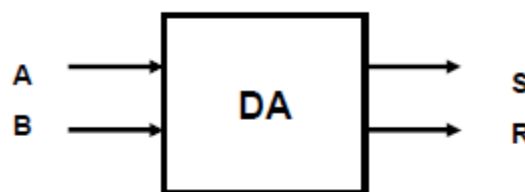
- a. Schéma symbolique :** construire son schéma symbolique en identifiant les entrées et les sorties (IN / OUT) de la fonction.
- b. Table de vérité :** construire sa table de vérité.
- c. équations de sorties :** extraire les équations de sorties à partir de la table de vérité, simplifier les fonctions 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)

### 3. Circuits combinatoires usuelles :

#### 3.1. 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:

$$\begin{cases} 0+0=0 & \text{retenue } 0 \\ 0+1=1 & \text{retenue } 0 \\ 1+0=1 & \text{retenue } 0 \\ 1+1=0 & \text{retenue } 1 \end{cases}$$

**b-Table de vérité :**

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

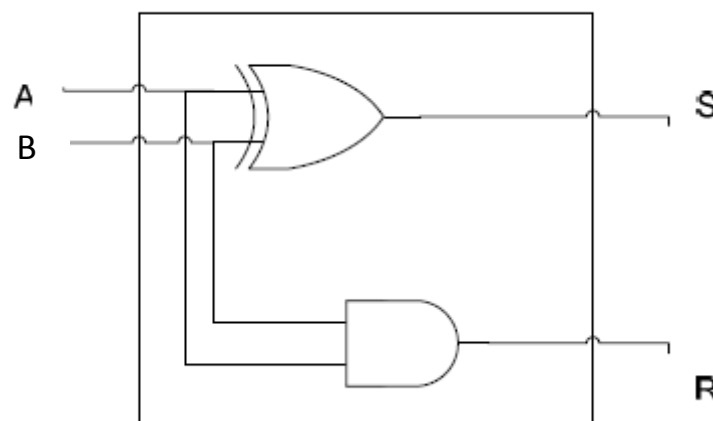
**c-équations de sortie :**

De la table de vérité on trouve :

$$R = AB$$

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

**d-schéma logique :**



**3.2. Additionneur complet**

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

$$\begin{array}{r} \color{red}{r_4} \quad \color{red}{r_3} \quad \color{red}{r_2} \quad \color{red}{r_1} \quad \color{red}{r_0=0} \\ \quad a_4 \quad a_3 \quad a_2 \quad a_1 \\ + \quad b_4 \quad b_3 \quad b_2 \quad b_1 \\ \hline r_4 \quad s_4 \quad s_3 \quad s_2 \quad s_1 \end{array} \qquad \begin{array}{r} \quad \quad \quad r_{i-1} \\ \quad \quad \quad a_i \\ + \quad \quad \quad b_i \\ \hline r_i \quad s_i \end{array}$$

**Exemple d'un additionneur complet 1 bit**

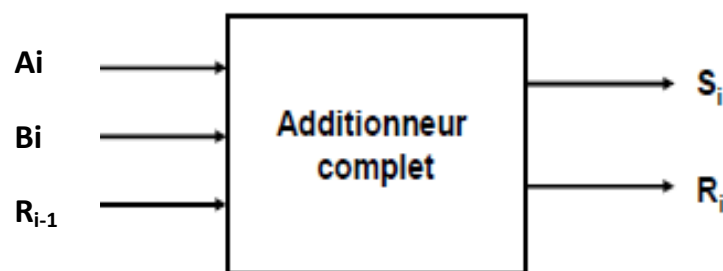
• L'additionneur complet **un bit** possède 3 entrées :

- ai : le premier nombre sur un bit.
- bi : le deuxième nombre sur un bit.
- ri-1 : la retenue entrante sur un bit.

• Il possède deux sorties :

- Si : la somme
- Ri la retenue sortante

**a-schéma symbolique**



**b-Table de vérité :**

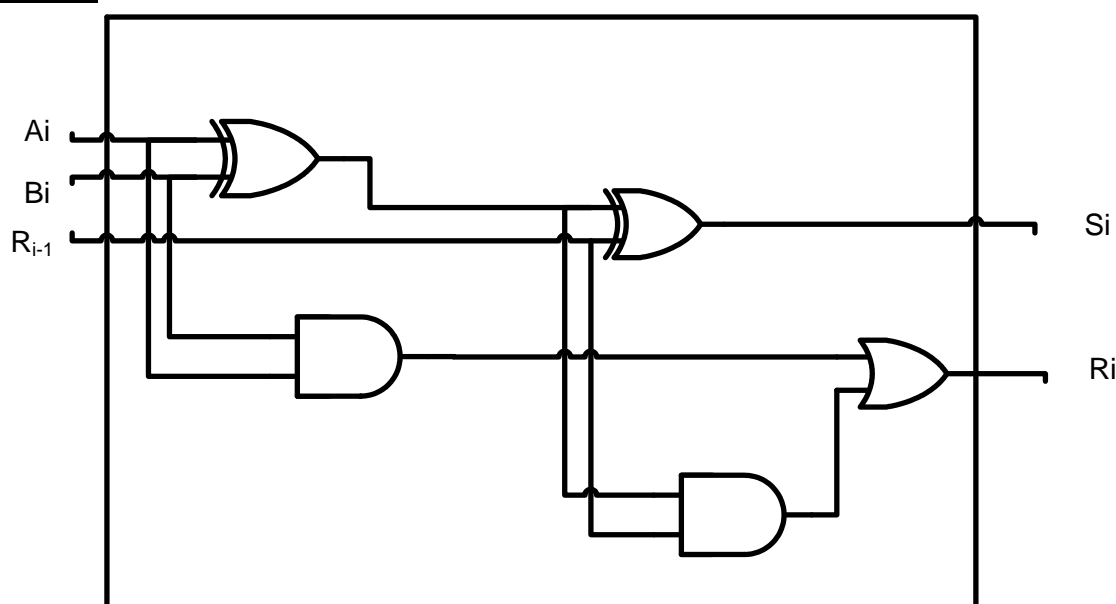
Ai	Bi	Ri-1	Si	Ri
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

**c-équations de sortie :**

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

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

**d-schéma logique :**



**Additionneur sur 4 bits**

- Un additionneur sur 4 bits est un circuit qui permet de faire l'addition de deux nombres A et B de 4 bits chacun :  $A(a_3a_2a_1a_0)$  et  $B(b_3b_2b_1b_0)$   
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.

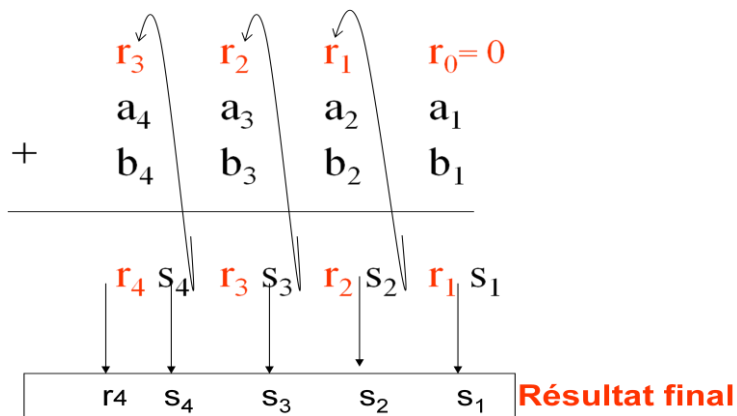
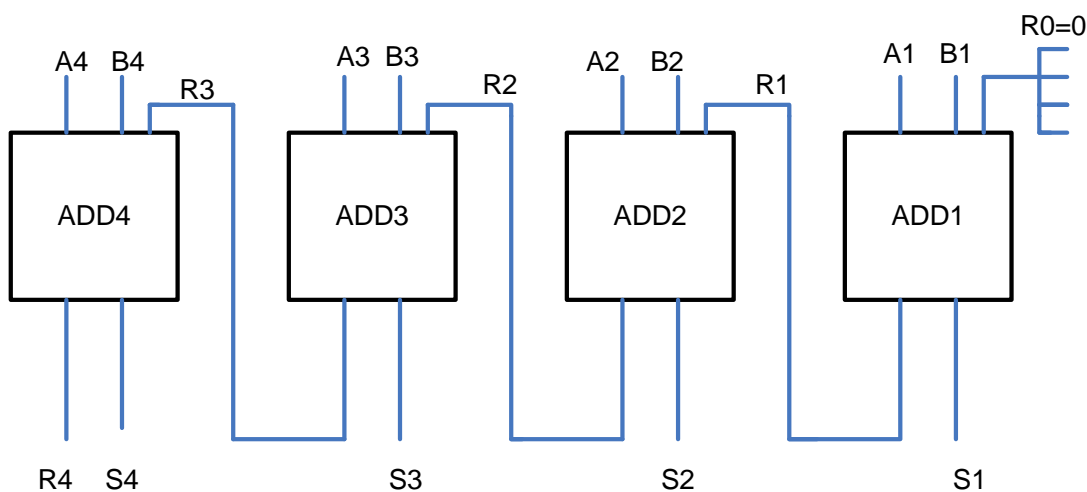


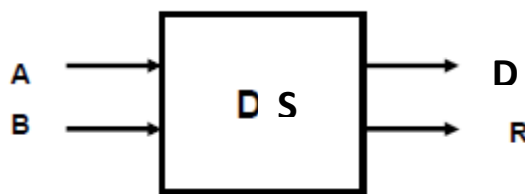
Schéma logique d'un additionneur complet 4 bits



**3.3. 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:

$$\begin{cases} 0-0=0 & \text{retenue } 0 \\ 0-1=1 & \text{retenue } 1 \\ 1-0=1 & \text{retenue } 0 \\ 1-1=0 & \text{retenue } 0 \end{cases}$$

**b-Table de vérité :**

A	B	D	R
---	---	---	---

0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

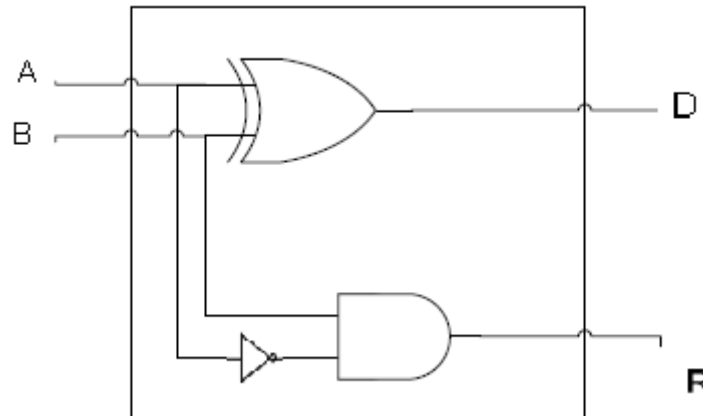
**c-équations de sortie :**

De la table de vérité on trouve :

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

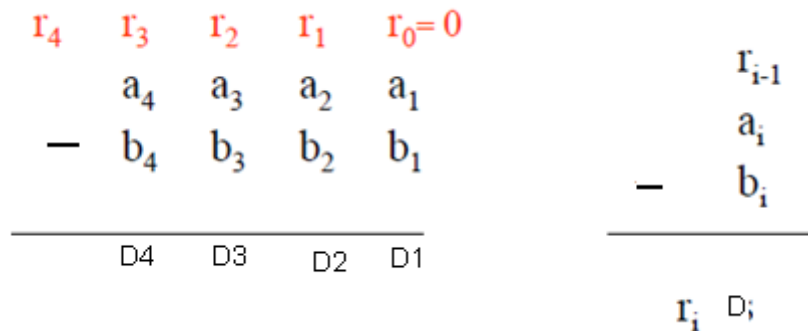
$$D = \bar{A}B + A\bar{B} = A \oplus B$$

**d-schéma logique :**



**3.4. Soustracteur complet**

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



**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.

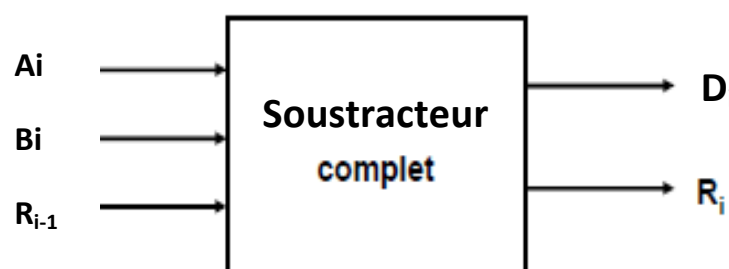
$r_{i-1}$  : la retenue entrante sur un bit.

• Il possède deux sorties :

$D_i$  : la différence

$R_i$  : la retenue sortante

**a-schéma symbolique**



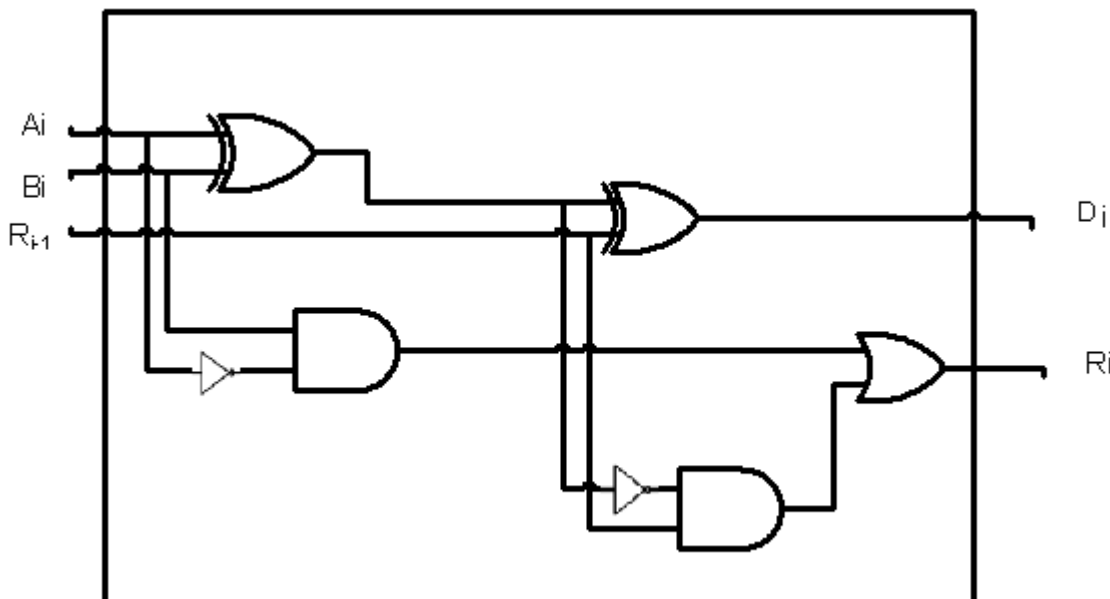
**b-Table de vérité :**

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-équations de sortie :**

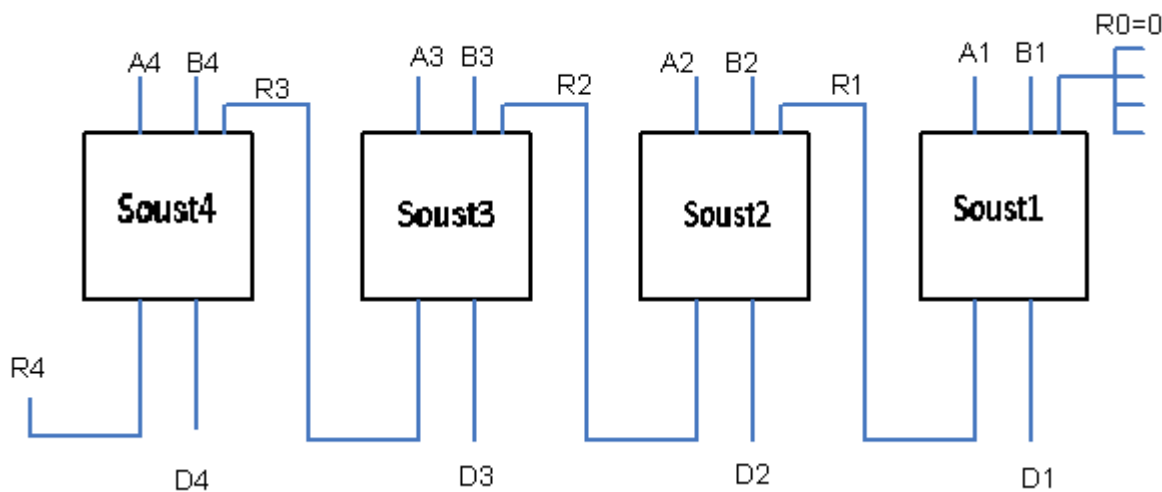
$$\begin{aligned}
 D_i &= \bar{A}_i \bar{B}_i R_{i-1} + \bar{A}_i B_i \bar{R}_{i-1} + A_i \bar{B}_i \bar{R}_{i-1} + A_i B_i R_{i-1} \\
 &= \bar{A}_i (\bar{B}_i R_{i-1} + B_i \bar{R}_{i-1}) + A_i (\bar{B}_i \bar{R}_{i-1} + B_i R_{i-1}) \\
 &= \bar{A}_i (B_i \oplus R_{i-1}) + A_i (\overline{B_i \oplus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

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

**d-schéma logique :**

**Soustracteur sur 4 bits**

- Un soustracteur sur 4 bits est un circuit qui permet de faire la différence de deux nombres A et B de 4 bits chacun :  $A(a_3a_2a_1a_0)$  et  $B(b_3b_2b_1b_0)$
- 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 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.

**Schéma logique d'un soustracteur complet 4 bits**



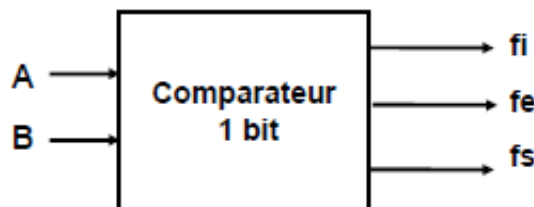
**3.5. Comparateur :**

• 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

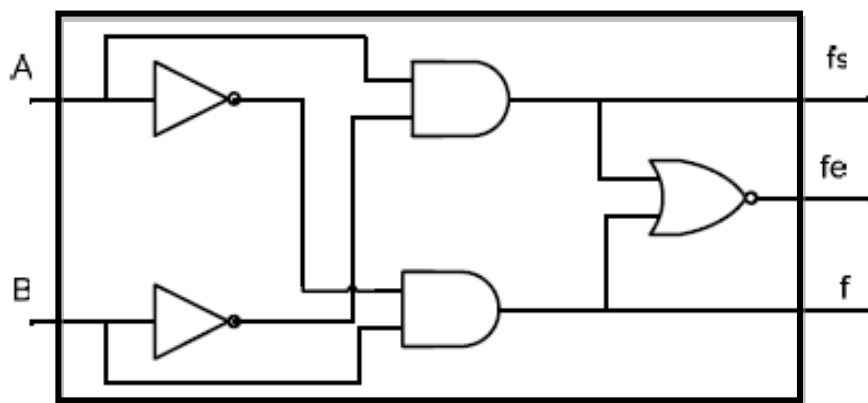
**c-équations de sortie :**

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

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

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

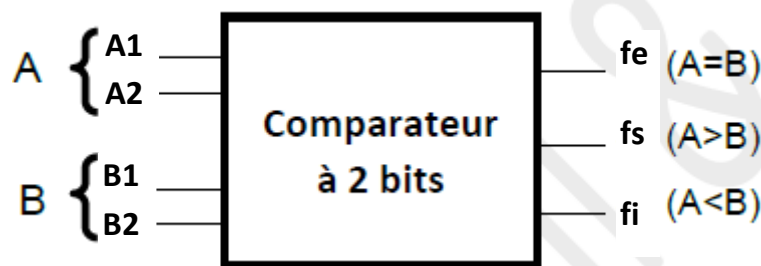
**d-schéma logique :**



**Exemple d'un 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 :**

A2	A1	B2	B1	fs	fe	fi
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-équations de sortie :**

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

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

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

$$= \bar{A}_2 B_2 + (A_2 \oplus B_2)(\bar{A}_1 B_1)$$

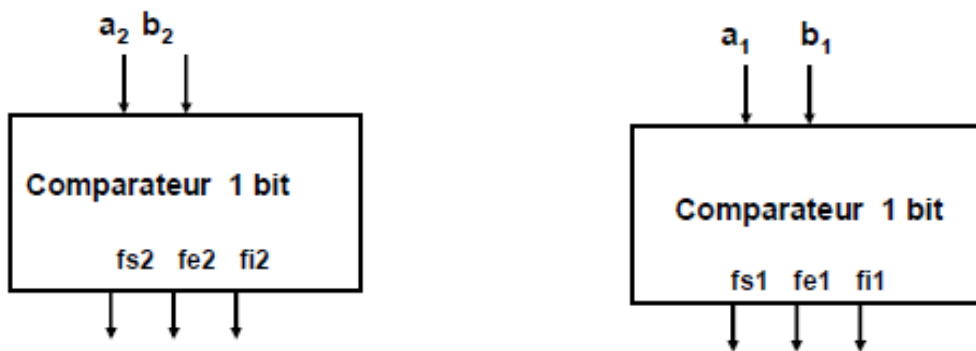
$$f_e = \bar{A}_2 \bar{A}_1 \bar{B}_2 \bar{B}_1 + \bar{A}_2 A_1 \bar{B}_2 B_1 + A_2 \bar{A}_1 B_2 \bar{B}_1 + A_2 A_1 B_2 B_1$$

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

**Comparateur 2 bits avec des comparateurs 1 bit**



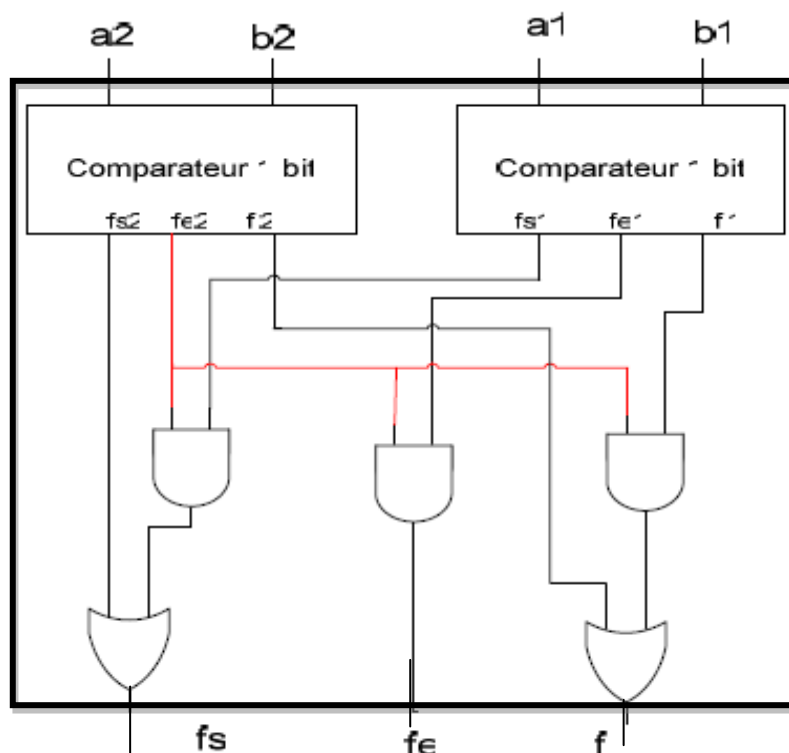
•C'est possible de réaliser un comparateur 2 bits en utilisant des comparateurs 1 bit et des portes logiques. •Il faut utiliser un comparateur pour comparer **les bits du poids faible** et un autre pour comparer **les bits du poids fort**. •Il faut **combiner** entre les sorties des deux comparateurs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



$$A=B \text{ si } A_2=B_2 \text{ et } A_1=B_1 \quad \rightarrow \quad f_e = \overline{(A_2 \oplus B_2)} \cdot \overline{(A_1 \oplus B_1)} = fe_2 \cdot fe_1$$

$$A>B \text{ si } A_2>B_2 \text{ ou } (A_2=B_2 \text{ et } A_1>B_1) \quad \rightarrow \quad f_s = A_2 \overline{B_2} + \overline{(A_2 \oplus B_2)} (A_1 \overline{B_1}) = fs_2 + fe_2 \cdot fs_1$$

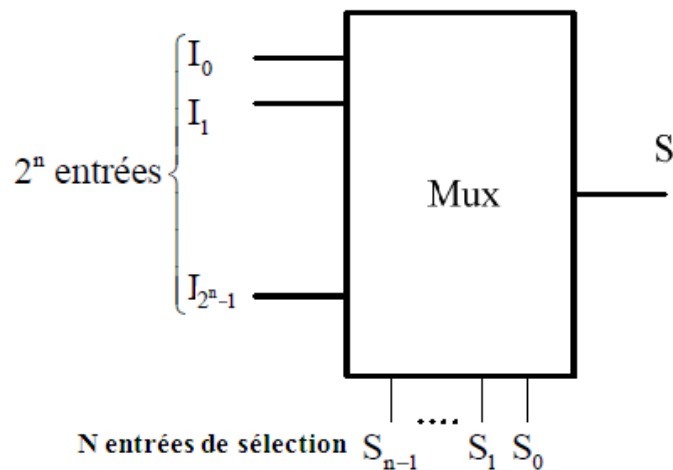
$$A<B \text{ si } A_2<B_2 \text{ ou } (A_2=B_2 \text{ et } A_1<B_1) \quad \rightarrow \quad f_i = \overline{A_2} B_2 + \overline{(A_2 \oplus B_2)} (\overline{A_1} B_1) = fi_2 + fe_2 \cdot fi_1$$



### 3.6. Multiplexeur

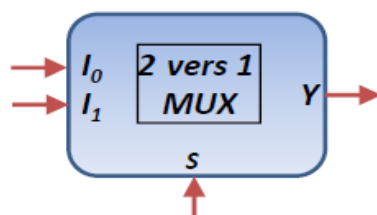
• Un multiplexeur est un circuit combinatoire qui permet de **sélectionner une information** (1 bit) parmi **2n valeurs en entrée**. Il possède :

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



**Exemple d'un multiplexeur 2X1**

**a-schéma symbolique :**



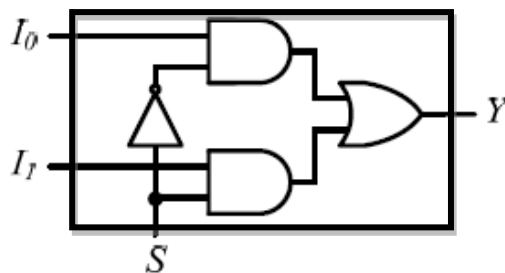
**b-Table de vérité :**

S	Y
0	$I_0$
1	$I_1$

**c-équations de sortie :**

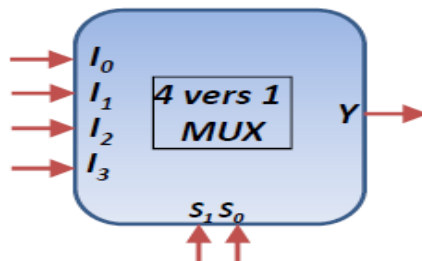
$$Y = \bar{S}I_0 + SI_1$$

**d-schéma logique :**



**Exemple d'un multiplexeur 4X1**

**a-schéma symbolique :**



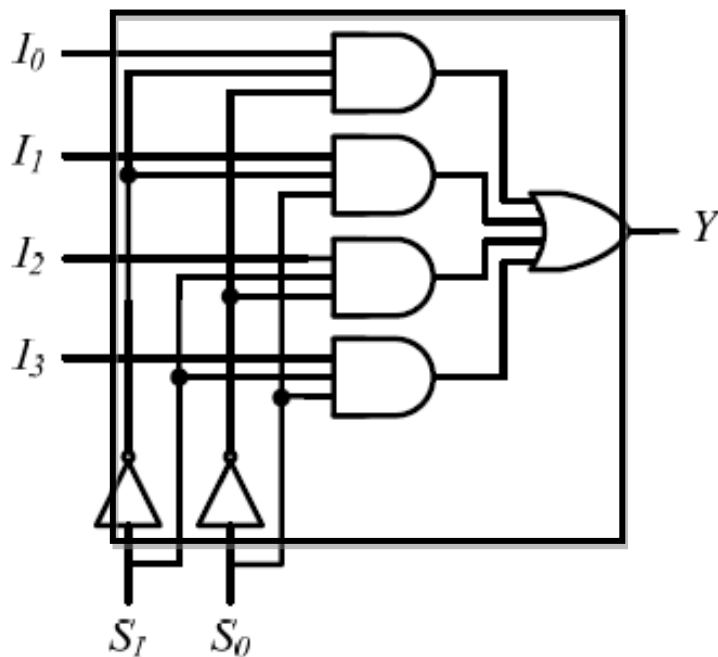
**b-Table de vérité :**

S1	S0	Y
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$

**c-équations de sortie :**

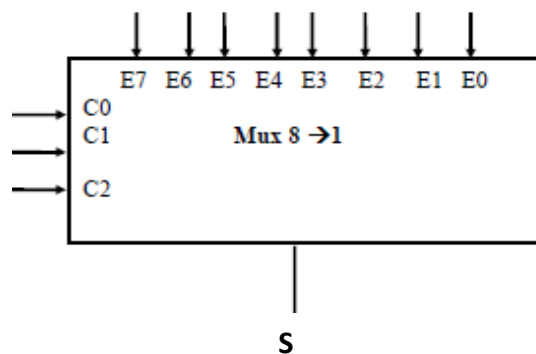
$$Y = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$

**d-schéma logique :**



**Exemple d'un multiplexeur 8X1**

**a-schéma symbolique :**



**b-Table de vérité :**

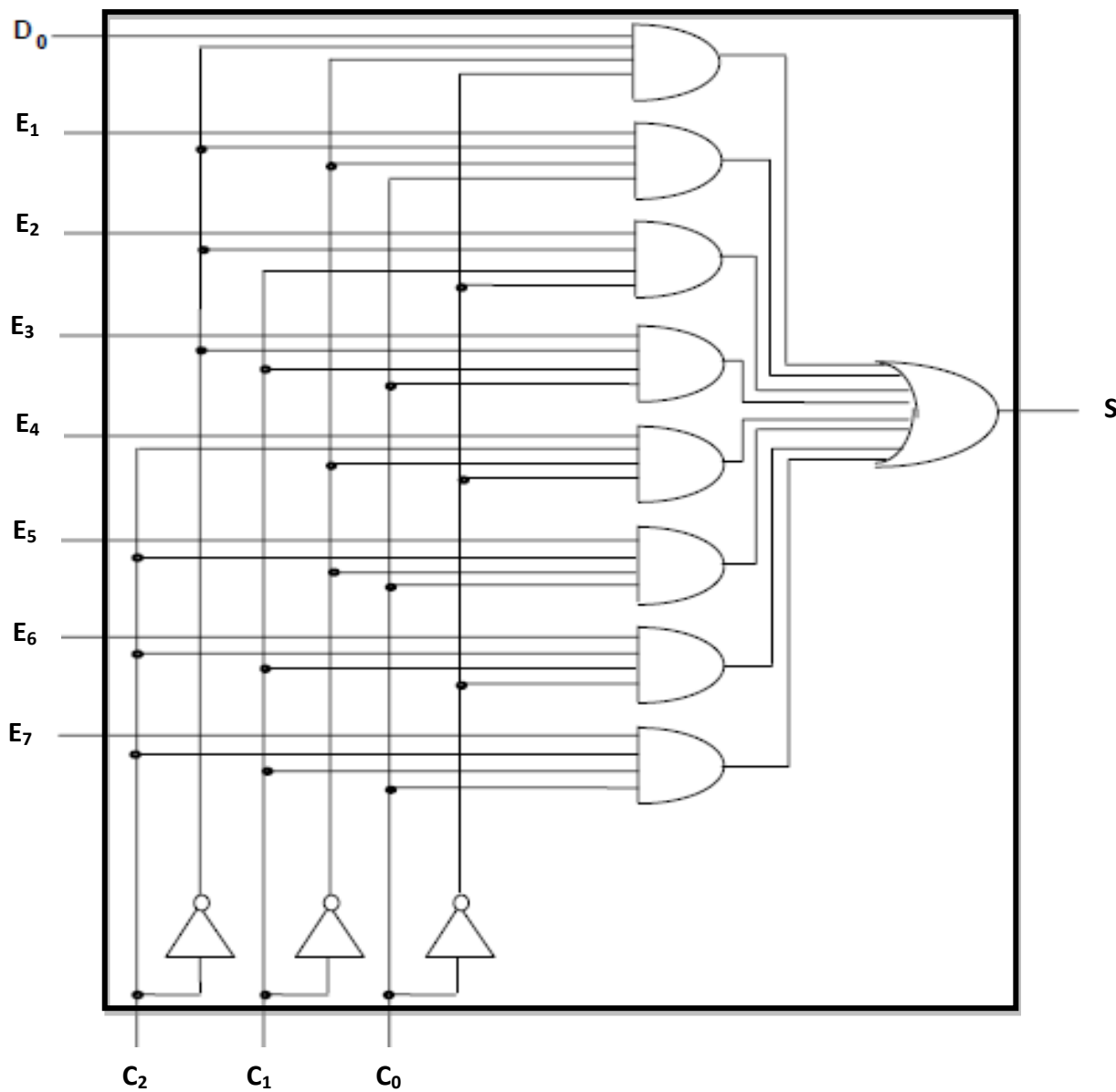
C2	C1	C0	S
0	0	0	E0
0	0	1	E1
0	1	0	E2
0	1	1	E3
1	0	0	E4
1	0	1	E5
1	1	0	E6
1	1	1	E7

**c-équations de sortie :**

$$S = \bar{C}_2 \bar{C}_1 \bar{C}_0 E_0 + \bar{C}_2 \bar{C}_1 C_0 E_1 + \bar{C}_2 C_1 \bar{C}_0 E_2 + \bar{C}_2 C_1 C_0 E_3 + C_2 \bar{C}_1 \bar{C}_0 E_4 + C_2 \bar{C}_1 C_0 E_5 + C_2 C_1 \bar{C}_0 E_6 + C_2 C_1 C_0 E_7$$

**d-schéma logique :**

E<sub>0</sub>



**Exemple d'un 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 :**

Ai	Bi	Ri-1	Si	Ri
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-équations de sortie :**

Réalisation de la fonction de la somme

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

$$S_i = \bar{C}_2 \bar{C}_1 \bar{C}_0 (E_0) + \bar{C}_2 \bar{C}_1 C_0 (E_1) + \bar{C}_2 C_1 \bar{C}_0 (E_2) + \bar{C}_2 C_1 C_0 (E_3) + C_2 \bar{C}_1 \bar{C}_0 (E_4) + C_2 \bar{C}_1 C_0 (E_5) + C_2 C_1 \bar{C}_0 (E_6) + C_2 C_1 C_0 (E_7)$$

$$C_2 = A_i$$

$$C_1 = B_i$$

On pose :

$$C_0 = R_{i-1}$$

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

Réalisation de la fonction de la retenue

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

$$R_i = \bar{C}_2 \bar{C}_1 \bar{C}_0(E_0) + \bar{C}_2 \bar{C}_1 C_0(E_1) + \bar{C}_2 C_1 \bar{C}_0(E_2) + \bar{C}_2 C_1 C_0(E_3) + C_2 \bar{C}_1 \bar{C}_0(E_4) + C_2 \bar{C}_1 C_0(E_5) + C_2 C_1 \bar{C}_0(E_6) + C_2 C_1 C_0(E_7)$$

$$C_2 = A_i$$

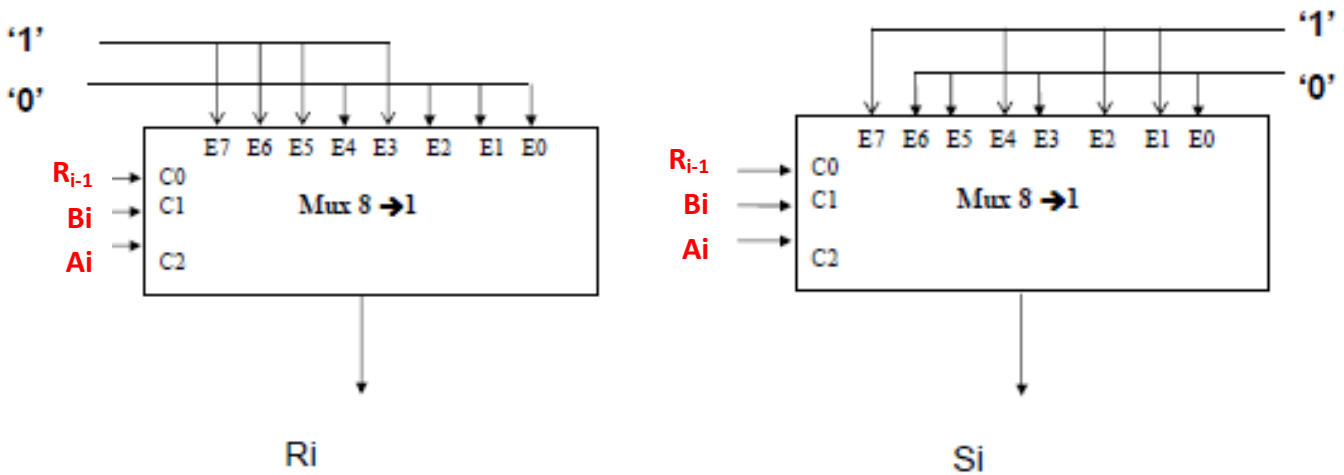
$$C_1 = B_i$$

On pose :

$$C_0 = R_{i-1}$$

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

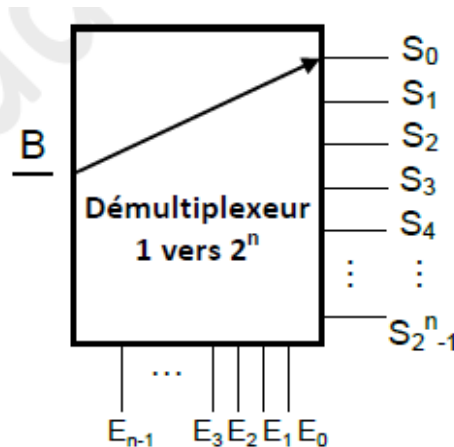
**d-schéma logique :**



### 3.7. Démultiplexeurs

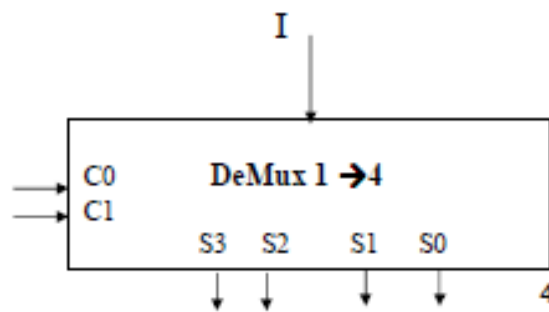
• Il joue le rôle inverse d'un multiplexeur, il permet de faire passer une information dans l'une des sorties selon les valeurs des entrées de commandes. Il possède :

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



#### Exemple d'un démultiplexeur 1X4

**a-schéma symbolique :**



**b-Table de vérité :**

C1	C0	S3	S2	S1	S0
0	0	0	0	0	i
0	1	0	0	i	0
1	0	0	i	0	0
1	1	i	0	0	0

**c-équations de sortie :**

$$S_0 = \overline{C_1}\overline{C_0}I$$

$$S_1 = \overline{C_1}C_0I$$

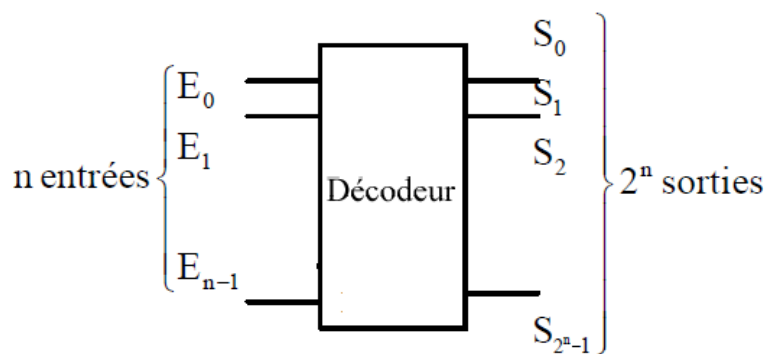
$$S_2 = C_1\overline{C_0}I$$

$$S_3 = C_1C_0I$$

**3.8. Décodeur binaire**

• C'est un circuit combinatoire qui est constitué de :

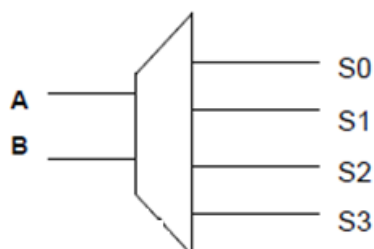
- N entrées de données
- $2^N$  sorties



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

**Exemple d'un décodeur 2X4**

**a-schéma symbolique :**



**b-Table de vérité :**

A	B	S0	S1	S2	S3
---	---	----	----	----	----

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 :**

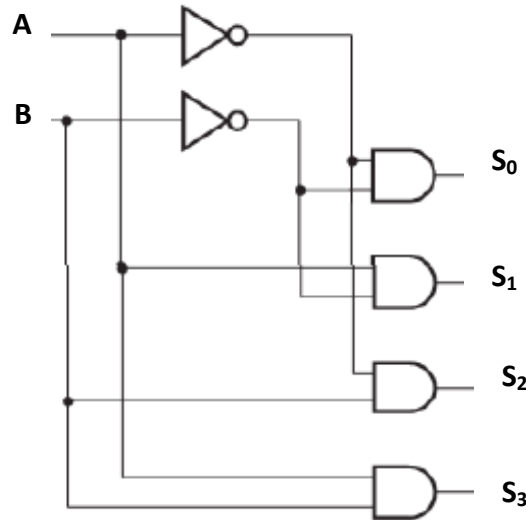
$$S_0 = \overline{A}\overline{B}$$

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

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

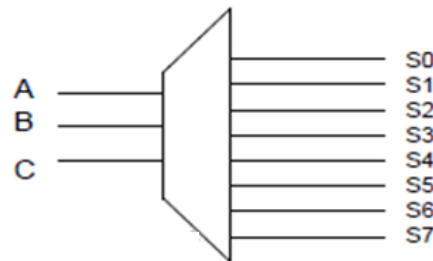
$$S_3 = AB$$

**d-schéma logique :**



**Exemple d'un décodeur 3X8**

**a-schéma symbolique :**



**b-Table de vérité :**

A	B	C	S0	S1	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

**c-équations de sortie :**

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

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

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

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

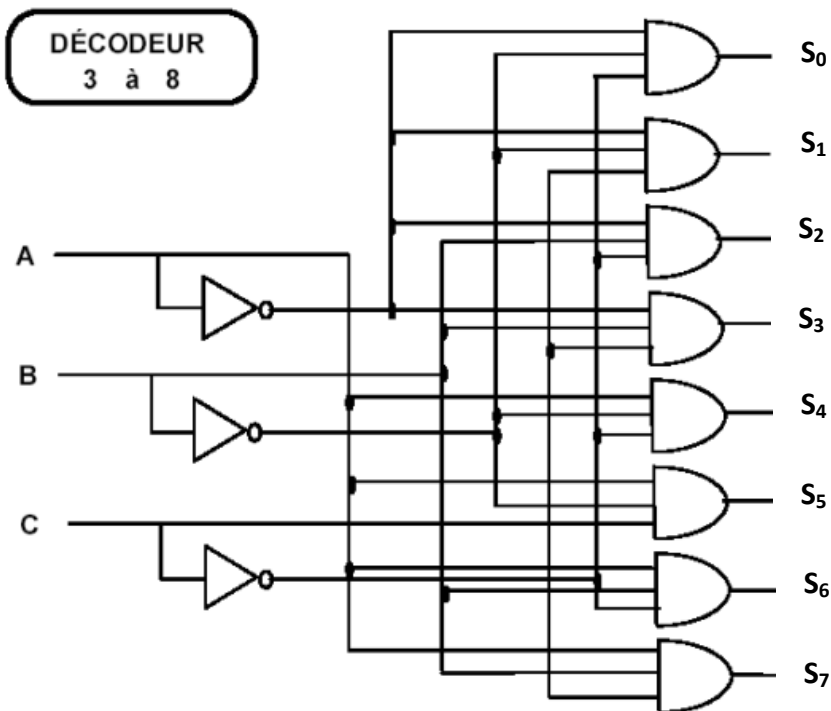
$$S_4 = A\overline{B}\overline{C}$$

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

$$S_6 = ABC\overline{C}$$

$$S_7 = ABC$$

**d-schéma logique :**

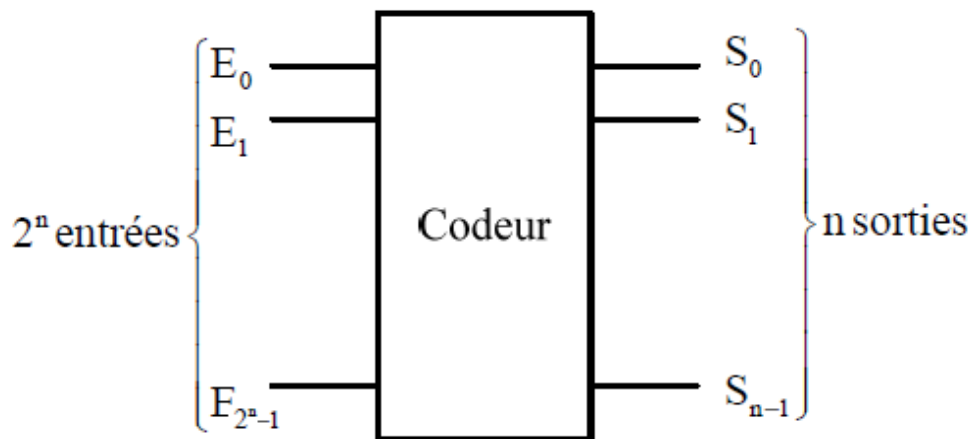


**3.9. Encodeur binaire (codeur)**

• Il joue le rôle inverse d'un décodeur. Il possède :

- $2^N$  entrées
- N sorties

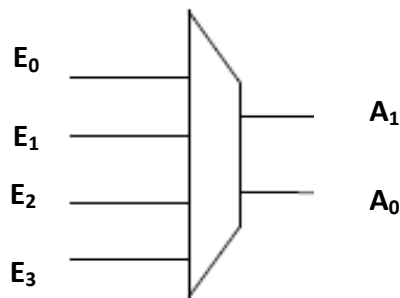
• Pour chaque combinaison en entrée on va avoir sont numéro (en binaire) à la sortie.



**Exemple d'un encodeur 4X2**



**a-schéma symbolique :**



**b-Table de vérité :**

E3	E2	E1	E0	A1	A0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

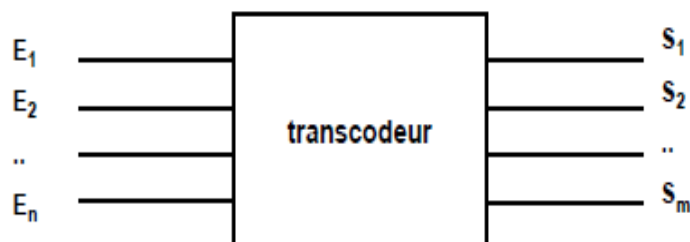
**c-équations de sortie :**

$$A_0 = E_2 + E_0$$

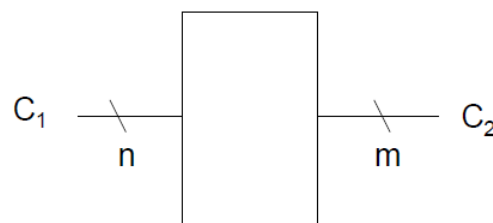
$$A_1 = E_1 + E_0$$

**3.10. 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 C<sub>1</sub> à un code C<sub>2</sub>

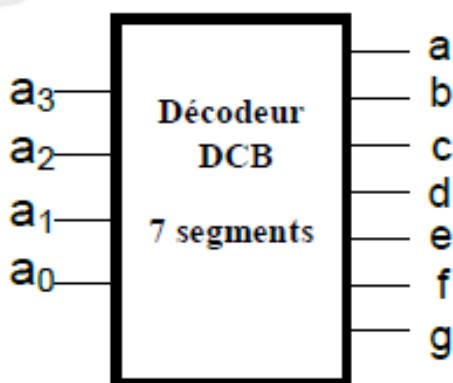


- Exemples de code : Binaire, binaire réfléchi, 7-segments, BCD, ...

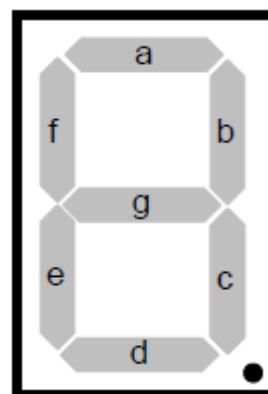
**Exemple d'un transcodeur 7-segments**

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).

**a-schéma symbolique :**



Code binaire 0 à 9



Configuration alimentation des diodes (ou LCD)

**b-Table de vérité :**

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

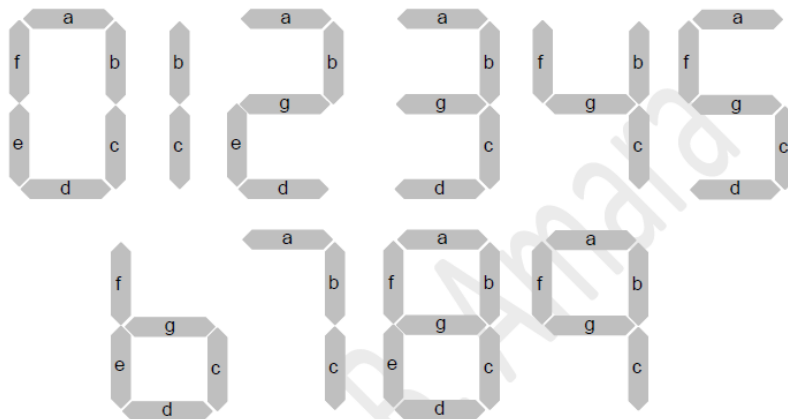


Table de vérité											Affichage
Entrées				Sorties							
$a_3$	$a_2$	$a_1$	$a_0$	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	0	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0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

c-équations de sortie :

**Segment a**

$a_3 a_2$	$\bar{a}_3 \bar{a}_2 00$	$\bar{a}_3 \bar{a}_2 01$	$a_3 a_2 11$	$a_3 \bar{a}_2 10$
$\bar{a}_1 \bar{a}_0 00$	1	0	∅	1
$\bar{a}_1 \bar{a}_0 01$	0	1	∅	1
$a_1 \bar{a}_0 11$	1	1	∅	∅
$a_1 \bar{a}_0 10$	1	0	∅	∅

$$a = \bar{a}_2 a_1 + a_2 a_0 + \bar{a}_2 \bar{a}_0 + a_3$$

**Segment b**

$a_3 a_2$	$\bar{a}_3 \bar{a}_2 00$	$\bar{a}_3 \bar{a}_2 01$	$a_3 a_2 11$	$a_3 \bar{a}_2 10$
$\bar{a}_1 \bar{a}_0 00$	1	1	∅	1
$\bar{a}_1 \bar{a}_0 01$	1	0	∅	1
$a_1 \bar{a}_0 11$	1	1	∅	∅
$a_1 \bar{a}_0 10$	1	0	∅	∅

$$b = \bar{a}_2 + \bar{a}_1 \bar{a}_0 + a_1 a_0 = \bar{a}_2 + a_1 \odot a_0$$

**Segment c**

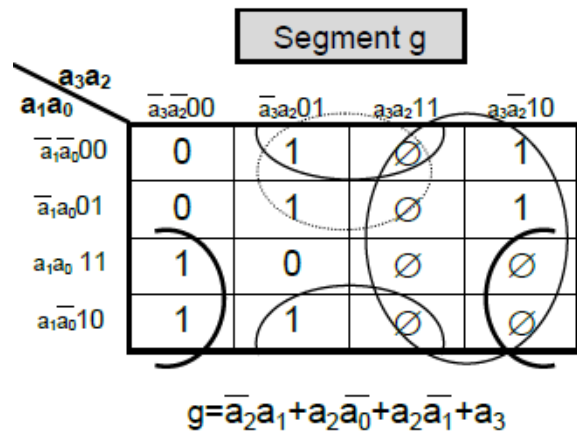
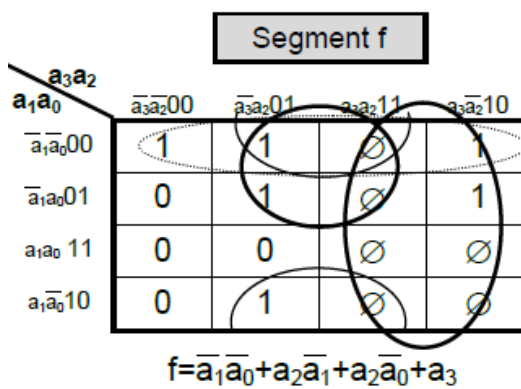
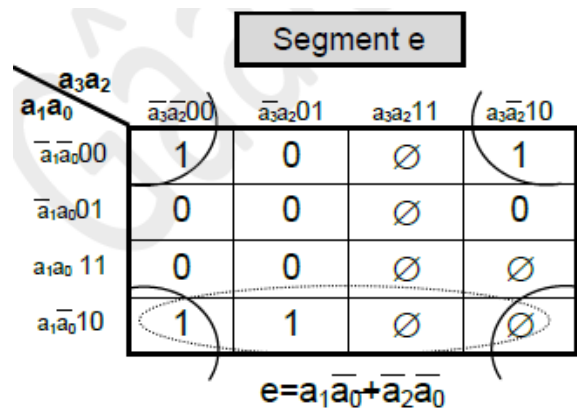
$a_3 a_2$	$\bar{a}_3 \bar{a}_2 00$	$\bar{a}_3 \bar{a}_2 01$	$a_3 a_2 11$	$a_3 \bar{a}_2 10$
$\bar{a}_1 \bar{a}_0 00$	1	1	∅	1
$\bar{a}_1 \bar{a}_0 01$	1	1	∅	1
$a_1 \bar{a}_0 11$	1	1	∅	∅
$a_1 \bar{a}_0 10$	0	1	∅	∅

$$c = a_2 + \bar{a}_1 + a_0$$

**Segment d**

$a_3 a_2$	$\bar{a}_3 \bar{a}_2 00$	$\bar{a}_3 \bar{a}_2 01$	$a_3 a_2 11$	$a_3 \bar{a}_2 10$
$\bar{a}_1 \bar{a}_0 00$	1	0	∅	1
$\bar{a}_1 \bar{a}_0 01$	0	1	∅	0
$a_1 \bar{a}_0 11$	1	0	∅	∅
$a_1 \bar{a}_0 10$	1	1	∅	∅

$$d = \bar{a}_2 \bar{a}_0 + a_3 \bar{a}_0 + \bar{a}_2 a_1 + a_1 \bar{a}_0 + a_2 \bar{a}_1 a_0$$



**Transcodeur BCD/EXESS3**

A	B	C	D	X	Y	Z	T
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

**4. Unité arithmétique et logique à 1 bit.**

Ce circuit permet d'effectuer les opérations logiques, l'addition binaire, la multiplication sur deux bits élémentaires, l'opération étant déterminée par un décodeur.

