

Chapitre II : La logique combinatoire

1. Définition

Un circuit logique est un composant électronique qui traite et exécute des opérations logiques (booléennes).

Il existe deux types de circuits logiques :

- Les circuits logiques combinatoires
- Les circuits logiques séquentiels

2. Les circuits combinatoires

Un circuit logique est dit combinatoire si l'état (valeur) de ses sorties ne dépend que de l'état (valeur) de ses entrées. Le circuit combinatoire ne doit donc pas présenter de réactions de la sortie sur l'entrée, de sorte à ce que l'état de la sortie ne dépend pas de l'histoire du circuit.

3. Etapes de conception d'un circuit combinatoire

La conception (synthèse) d'un circuit combinatoire passe par trois principales étapes après une lecture approfondie et attentive pour bien comprendre l'énoncé afin de déterminer les signaux d'entrée et de sortie et leur désignation par des symboles. Les trois principales étapes sont les suivantes :

- **Etablissement de la table de vérité**
Vue dans le cours STM1.
- **Simplification des fonctions logiques**
Vue dans le cours STM1.
- **Réalisation du schéma logique**
Vue dans le cours STM1.

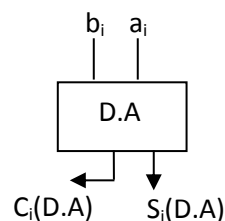
4. Etude de quelques circuits combinatoires usuels :

4.1. Le demi-additionneur

- Définition :

Le demi-additionneur est un circuit logique combinatoire qui permet de faire une opération d'addition arithmétique entre deux bits sans tenir compte la retenue de l'étape précédente.

- Schéma général :



Remarque :

b_i et a_i représentent les bits à additionner.

S_i représente le résultat de l'addition (somme).

C_i représente la retenue de l'opération de l'addition.

- Réalisation du circuit :

Table de vérité:

a_i	b_i	S_i	C_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

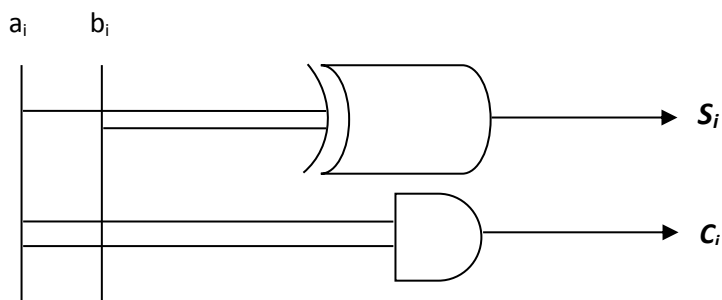
Simplification: directement de la table de vérité

$$S_i = a_i \bar{b}_i + \bar{a}_i b_i \quad (*)$$

$$= a_i \oplus b_i \quad (**)$$

$$C_i = a_i b_i$$

Logigramme: pour S_i , la représentation peut être faite par (*) ou (**)

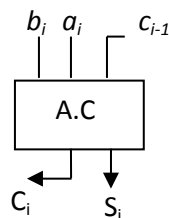


4.2. L'additionneur complet

- Définition :

L'additionneur est un circuit logique combinatoire qui permet de faire une opération d'addition arithmétique entre deux bits en prenant en compte la retenue de l'étape précédente.

- Schéma général :



Remarque :

b_i et a_i représentent les bits à additionner.

c_{i-1} représente la retenue de l'étape précédente.

S_i représente le résultat de l'addition (somme).

C_i représente la retenue de l'opération de l'addition.

- Réalisation du circuit :

Table de vérité:

c_{i-1}	a_i	b_i	S_i	C_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

Simplification par Tableau de Karnaugh:

C_i		$a_i b_i$			
c_{i-1}		00	01	11	10
0		0	0	1	0
1		0	1	1	1

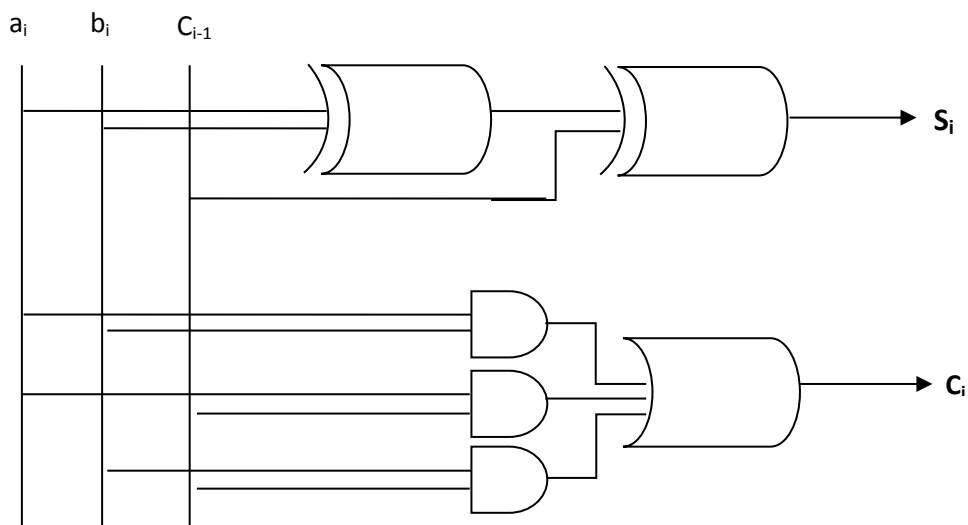
S_i		$a_i b_i$			
c_{i-1}		00	01	11	10
0		0	1	0	1
1		1	0	1	0

$$C_i = a_i b_i + c_{i-1} a_i + c_{i-1} b_i$$

$$S_i = \bar{c}_{i-1} \bar{a}_i b_i + \bar{c}_{i-1} a_i \bar{b}_i + c_{i-1} \bar{a}_i \bar{b}_i + c_{i-1} a_i b_i \quad (*)$$

$$= a_i \oplus b_i \oplus c_{i-1} \quad (**)$$

Logigramme: pour S_i , la représentation peut être faite par (*) ou (**)

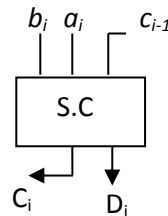


4.3. L'additionneur soustracteur (Soustracteur complet)

- Définition :

L'additionneur soustracteur (Soustracteur complet) est un circuit logique combinatoire qui permet de faire une opération de soustraction arithmétique entre deux bits en prenant en compte la retenue de l'étape précédente.

- Schéma général :



- Réalisation du circuit :

Table de vérité:

c_{i-1}	a_i	b_i	D_i	C_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

Simplification par Tableau de Karnaugh:

C_i		$a_i b_i$			
		c_{i-1}	00	01	11
0	0	0	1	0	0
1	0	1	1	1	0

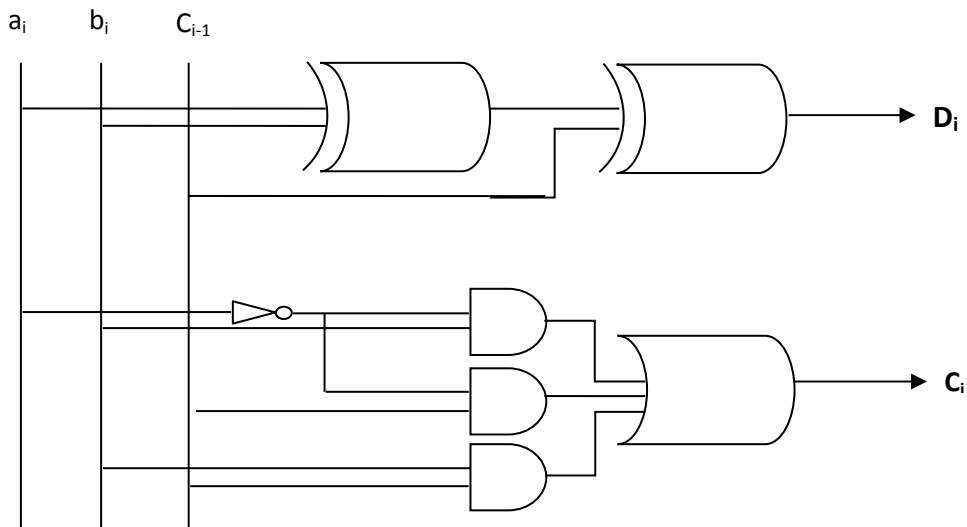
D_i		$a_i b_i$			
		c_{i-1}	00	01	11
0	0	0	1	0	1
1	0	1	0	1	0

$$C_i = \bar{a}_i b_i + c_{i-1} \bar{a}_i + c_{i-1} b_i$$

$$S_i = \bar{c}_{i-1} \bar{a}_i b_i + \bar{c}_{i-1} a_i \bar{b}_i + c_{i-1} \bar{a}_i \bar{b}_i + c_{i-1} a_i b_i \quad (*)$$

$$= a_i \oplus b_i \oplus c_{i-1} \quad (**)$$

Logigramme: pour D_i , la représentation put être faite par (*) ou (**)



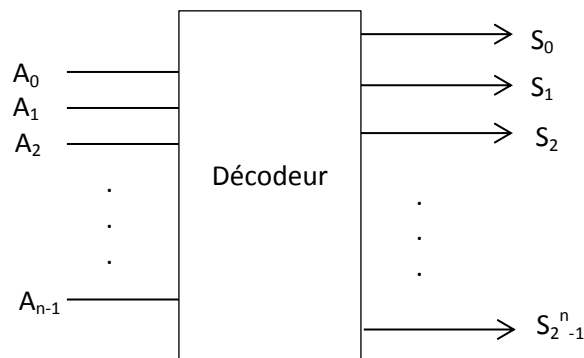
4.4. Les décodeurs

- **Définition :**

Le décodeur est un système combinatoire ayant pour fonction d'activer une des 2^n sorties. La sélection est faite à l'aide de n lignes d'adresse et les sorties sont mutuellement exclusives.

- **Notation :** Décodeur 1 parmi 2^n (Décodeur comme un DEMUX avec $E=1$)

- **Représentation générale:**



- **Exemple :** réalisation d'un Décodeur 1 parmi 4.

Nous avons 4 sorties $\Rightarrow 4 = 2^2 \Rightarrow n=2$

\Rightarrow Nous avons 2 lignes d'adresse.

\Rightarrow Le schéma général de ce décodeur est comme suit :

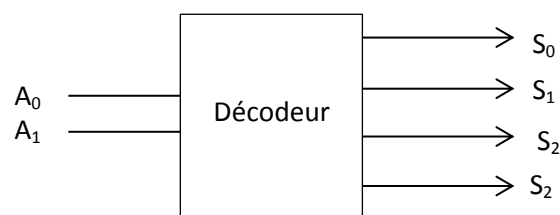


Table de vérité:

A_1	A_0	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

Simplification: directement de la table de vérité

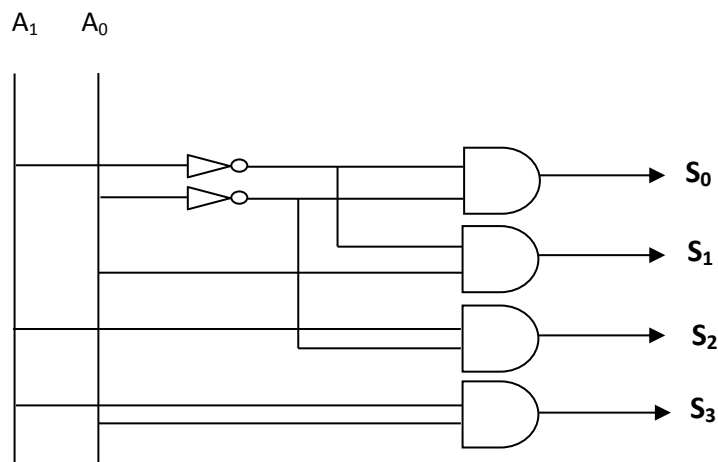
$$S_0 = \overline{A_1} \cdot \overline{A_0}$$

$$S_1 = \overline{A_1} \cdot A_0$$

$$S_2 = A_1 \cdot \overline{A_0}$$

$$S_3 = A_1 \cdot A_0$$

Logigramme:



Remarque:

Le décodeur peut être utilisé pour l'adressage mémoire.

4.5. Les multiplexeurs

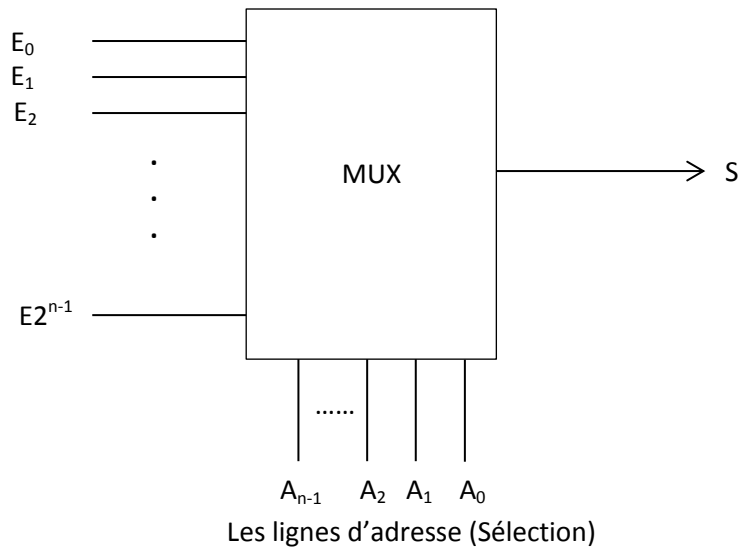
- **Définition :**

Le multiplexeur est un système combinatoire ayant pour fonction de sélectionner une parmi 2^n entrées et la transmettre à la sortie. La sélection est faite à l'aide de n lignes d'adresses.

- **Notation :** MUX 2^n à 1.

- **Représentation générale :**

Les entrées d'informations



- **Exemple** : réalisation d'un multiplexeur MUX 4 à 1.
- Nous avons 4 entrées $\Rightarrow 4 = 2^2 \Rightarrow n=2$
- \Rightarrow Nous avons 2 lignes d'adresse.
- \Rightarrow Le schéma général de ce multiplexeur est comme suit :

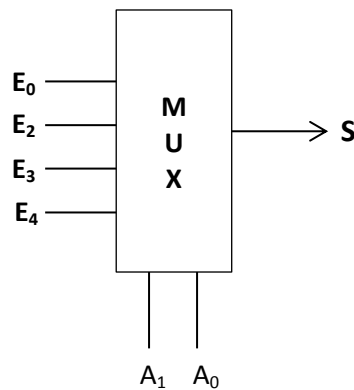


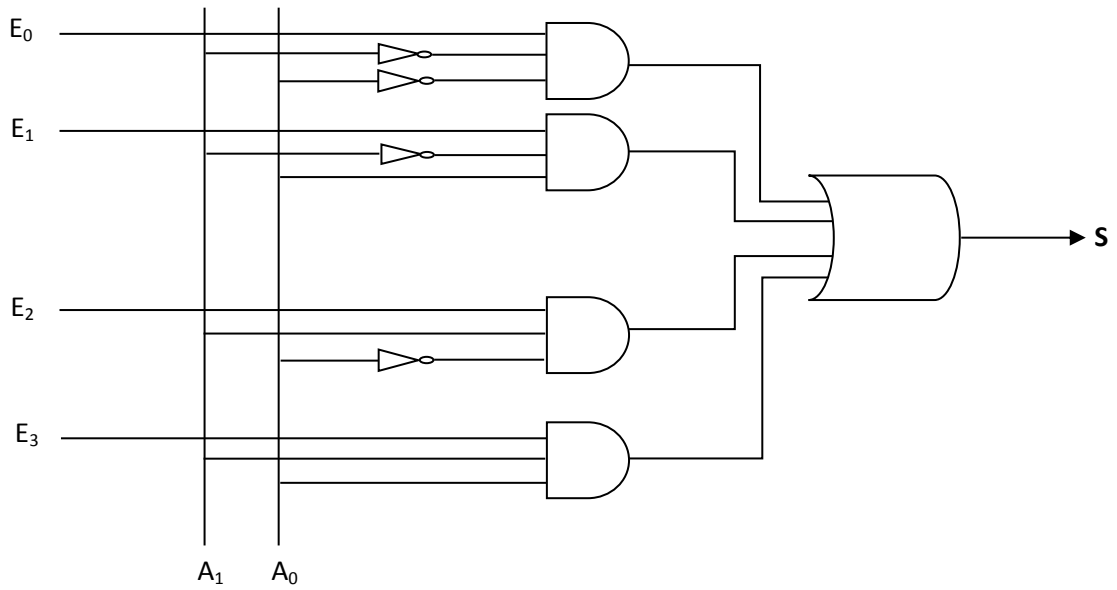
Table de vérité:

A_1	A_0	S
0	0	E_0
0	1	E_1
1	0	E_2
1	1	E_3

Simplification: directement de la table de vérité, on obtient l'expressions suivante :

$$\rightarrow S = E_0 \bar{A}_1 \bar{A}_0 + E_1 \bar{A}_1 A_0 + E_2 A_1 \bar{A}_0 + E_3 A_1 A_0$$

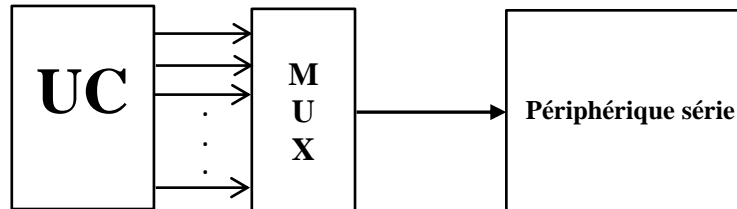
Logigramme:



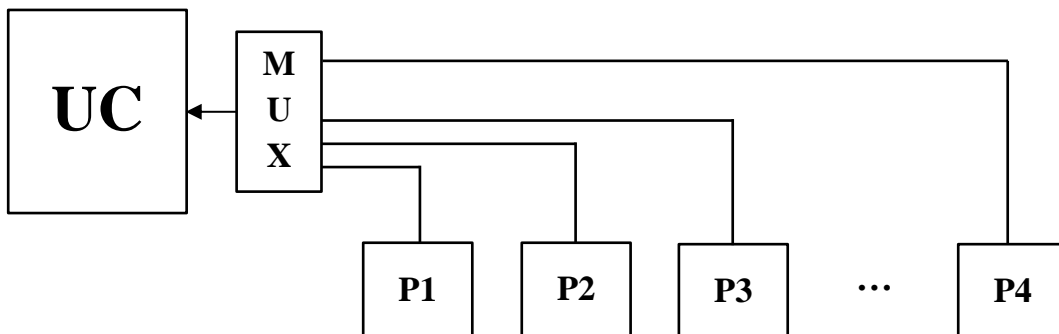
Remarque:

Les Principales applications des MUXs sont :

☞ Conversion parallèle / série.



☞ Concentration des informations sur une seule ligne.



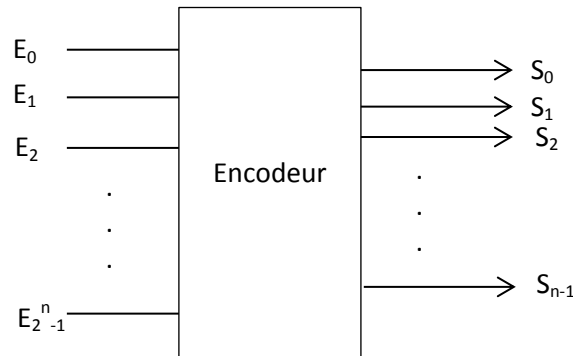
4.6. Les encodeurs de priorité

- **Définition :**

L'encodeur est système combinatoire ayant pour fonction de retourner l'index d'activation d'une parmi 2^n entrées. L'index d'activation est donné sur n lignes d'adresses. Lorsque plusieurs entrées sont activées, l'encodeur accorde la priorité à l'entrée active la plus prioritaire parmi les entrées actives.

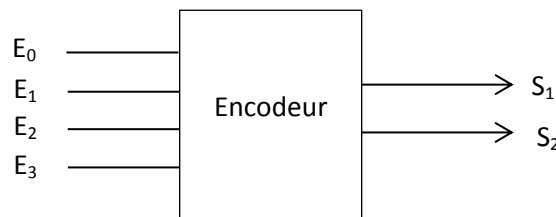
- **Notation :** Encodeur 2^n à n

- **Représentation générale:**



- **Exemple :** réalisation d'un Encodeur 4 à 2.

Le schéma général de ce décodeur est comme suit :



L'encodeur accorde une priorité aux entrées comme suit : $E_0 > E_1 > E_2 > E_3$

Table de vérité:

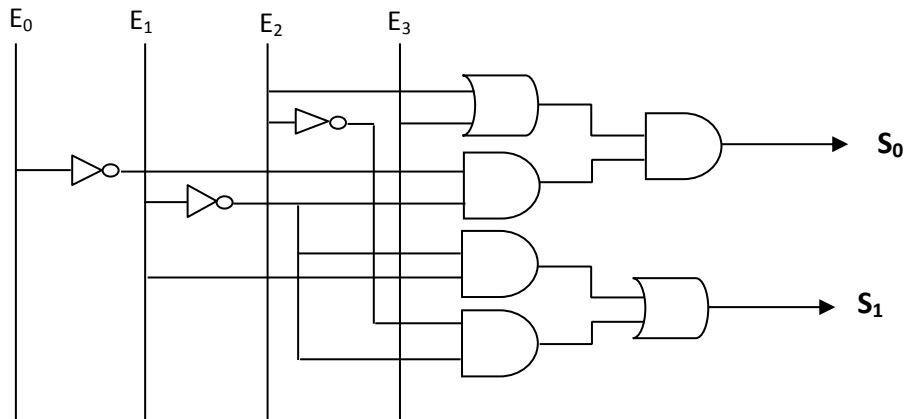
E_0	E_1	E_2	E_3	S_0	S_1
1	X	X	X	0	0
0	1	X	X	0	1
0	0	1	X	1	0
0	0	0	1	1	1

Simplification: directement de la table de vérité, en utilise la propriété : $\alpha + \bar{\alpha}\beta = \alpha + \beta$

$$\begin{aligned}
 S_0 &= \bar{E}_0 \cdot \bar{E}_1 \cdot E_2 + \bar{E}_0 \cdot \bar{E}_1 \cdot \bar{E}_2 \cdot E_3 \\
 &= \bar{E}_0 \cdot \bar{E}_1 \cdot (E_2 + \bar{E}_2 \cdot E_3) \\
 &= \bar{E}_0 \cdot \bar{E}_1 \cdot (E_2 + E_3) \\
 S_1 &= \bar{E}_0 \cdot E_1 + \bar{E}_0 \cdot \bar{E}_1 \cdot \bar{E}_2 \cdot E_3 \\
 &= \bar{E}_0 \cdot (E_1 + \bar{E}_1 \cdot \bar{E}_2 \cdot E_3) \\
 &= \bar{E}_0 \cdot (E_1 + \bar{E}_2 \cdot E_3)
 \end{aligned}$$

$$= \overline{E_0} \cdot E_1 + E_1 \cdot \overline{E_2} \cdot E_3$$

Logigramme:



Remarque:

L'encodeur peut être utilisé pour l'interfaçage d'un clavier et le passage du système décimal vers le système BCD.

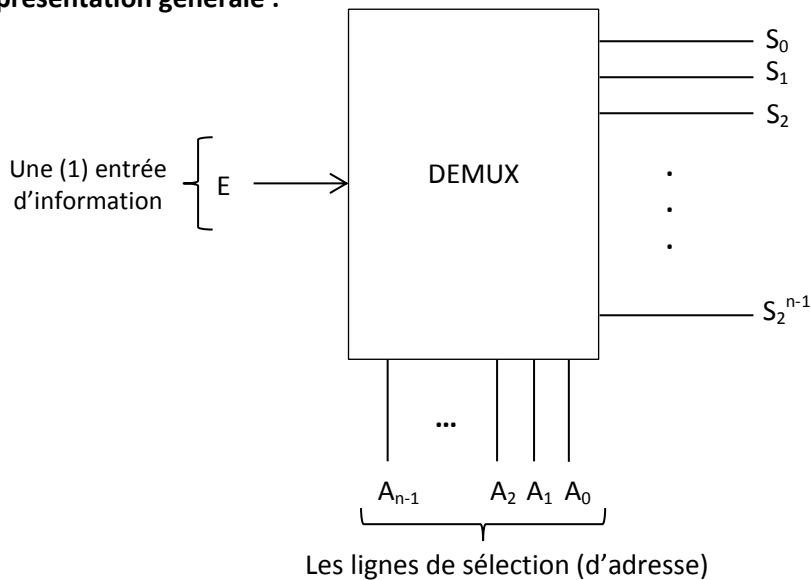
4.7. Les démultiplexeurs

- **Définition :**

Le démultiplexeur est un système combinatoire ayant pour fonction de transmettre une entrée vers une des 2^n sorties. La sélection est faite à l'aide de n lignes d'adresses et les sorties sont mutuellement exclusives.

- **Notation :** DEMUX 1 à 2^n

- **Représentation générale :**



- **Exemple :** réalisation d'un démultiplexeur DEMUX 1 à 4.

Nous avons 4 entrées $\Rightarrow 4 = 2^2 \Rightarrow n=2$

=> Nous avons 2 lignes d'adresse.

=> Le schéma général de ce multiplexeur est comme suit :

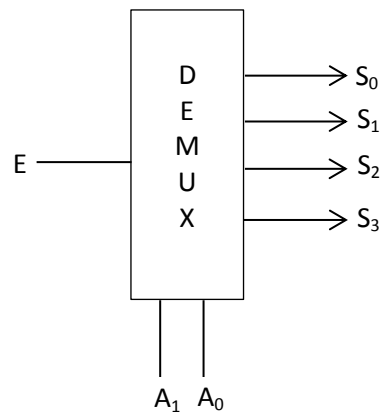


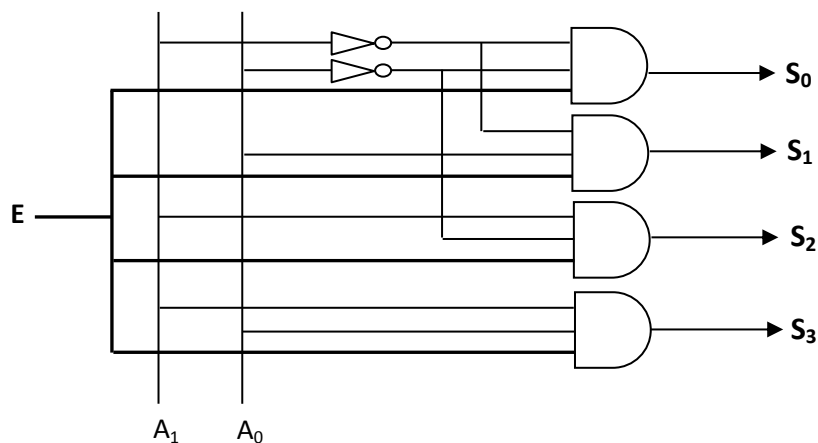
Table de vérité:

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

Simplification: directement de la table de vérité, on obtient les expressions suivantes :

$$\begin{aligned}
 S_0 &= E \overline{A_1} \overline{A_0} \\
 S_1 &= E \overline{A_1} A_0 \\
 S_2 &= E A_1 \overline{A_0} \\
 S_3 &= E A_1 A_0
 \end{aligned}$$

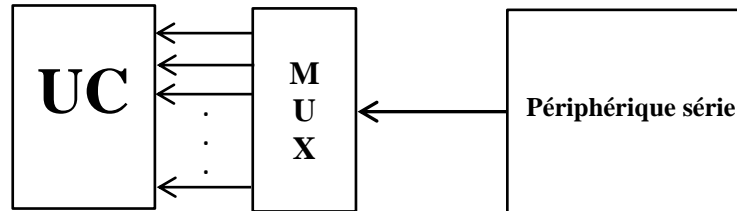
Logigramme:



Remarque:

Les Principales applications des DEMUXs sont :

☞ Conversion série / parallèle.



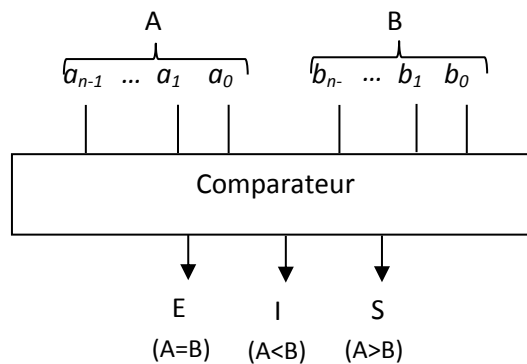
5. Autres exemples de circuits combinatoires

5.1. Le comparateur

- Définition :

Le comparateur est circuit logique combinatoire permettant de comparer deux nombres binaires (à n bits) $A(a_{n-1}, \dots, a_1, a_0)$ et $B(b_{n-1}, \dots, b_1, b_0)$ entre eux en donnant un résultat parmi les trois suivants : égale ($A=B$), inférieur ($A<B$), et supérieur ($A>B$).

- Représentation générale:



- **Exemple** : réalisation d'un comparateur de deux nombres à deux bits chacun: $A(a_1 a_0)$ et $B(b_1 b_0)$.
Le schéma général de ce comparateur est comme suit :

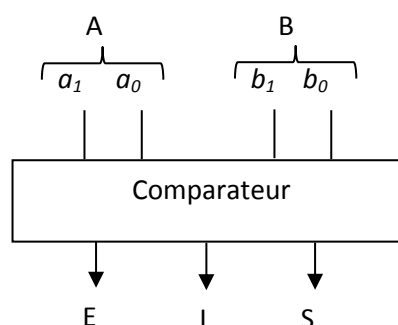


Table de vérité:

a_1	a_0	b_1	b_0	E	I	S
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	0	0	1
0	1	0	1	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

Simplification par table de Karnaugh:

E

$a_1a_0 \backslash b_1b_0$	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

I

$a_1a_0 \backslash b_1b_0$	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

$$E = \bar{a}_1 \bar{a}_0 \bar{b}_1 \bar{b}_0 + \bar{a}_1 a_0 \bar{b}_1 b_0 + a_1 a_0 b_1 b_0 + a_1 a_0 b_1 \bar{b}_0 + a_1 \bar{a}_0 b_1 \bar{b}_0$$

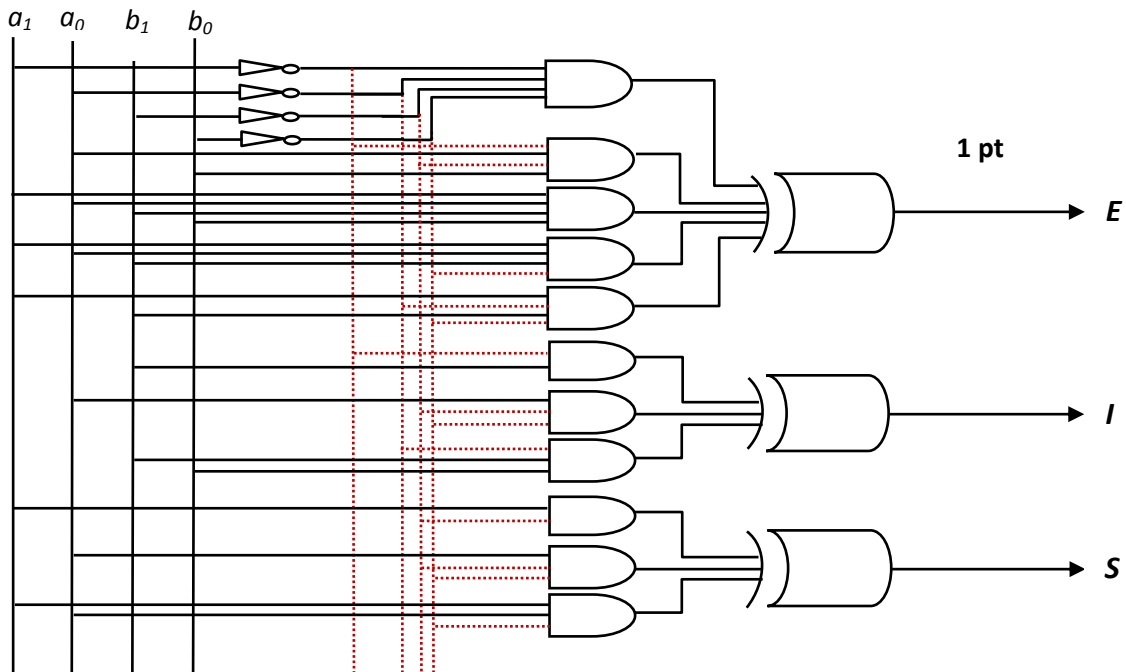
$$I = \bar{a}_1 b_1 + \bar{a}_1 \bar{a}_0 b_0 + \bar{a}_0 b_1 b_0$$

S 0.5 pt

$a_1 a_0 \backslash b_1 b_0$	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

$$S = a_1 \bar{b}_1 + a_0 \bar{b}_1 \bar{b}_0 + a_1 a_0 \bar{b}_0$$

Logigramme:



5.2. Le générateur de parité

- **Définition :**

Un circuit logique générateur de parité est un circuit qui reçoit en entrée une séquence de bits et donne en sortie la séquence de bits entrée ainsi qu'un nouveau bit dit de parité ajouté à cette séquence permettant la vérification de l'exactitude de cette séquence après transmission.

- **Exemple :** réalisation d'un générateur de bit parité pour une séquence de huit bits :

$a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$.

Si l'opérateur logique XOR (ou exclusif) porte sur un nombre de bits, le résultat est 0 si le nombre de 1 est paire, et 1 si le nombre de 1 est impair.

- le nombre de 1 est paire : 10110010

$$F = (1 \oplus 0) \oplus (1 \oplus 1) \oplus (0 \oplus 0) \oplus (1 \oplus 0)$$

$$= (1 \oplus 0) \oplus (0 \oplus 1)$$

$$= 1 \oplus 1$$

$$= 0$$

- le nombre de 1 est impair : 11010110

$$F = (1 \oplus 1) \oplus (0 \oplus 1) \oplus (0 \oplus 1) \oplus (1 \oplus 0)$$

$$= (0 \oplus 1) \oplus (1 \oplus 1)$$

$$= 1 \oplus 0$$

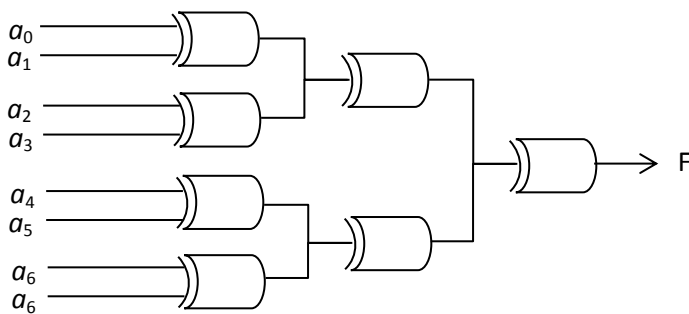
$$= 1$$

Logigramme:

On peut représenter ce circuit de deux manières différentes : en pyramide et en cascade.

En pyramide

$$F = (a_7 \oplus a_6) \oplus (a_5 \oplus a_4) \oplus (a_3 \oplus a_2) \oplus (a_1 \oplus a_0)$$



En cascade

$$F = (a_7 \oplus (a_6 \oplus (a_5 \oplus (a_4 \oplus (a_3 \oplus (a_2 \oplus (a_1 \oplus a_0))))))))$$

