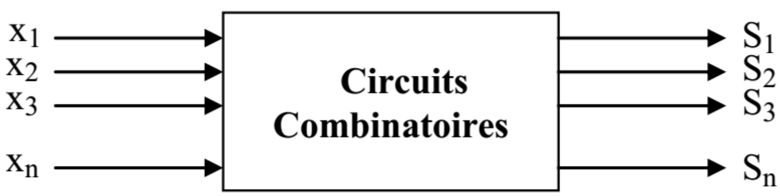
I.INTRODUCTION

La transmission de données nécessite fréquemment des opérations de conversion,. On utilise pour cela des **circuits combinatoires**.

Pour réaliser un circuit logique combinatoire, on doit utiliser plusieurs portes logiques élémentaires. Il existe plusieurs dispositifs logiques combinatoires couramment utilisé dans les systèmes numériques. On peut citer les codeurs (encodeur) , les décodeurs, les transcodeurs, les multiplexeurs, les démultiplexeurs, les comparateurs …

II.DEFINITION

La logique combinatoire concerne l'étude des fonctions dont la valeur de sortie ne dépend que de l'état logique des entrées se traduisant par une modification de la valeur des sorties et non pas non plus de ses états antérieurs : à chaque combinaison des variables d'entrée correspond toujours une seule combinaison des fonctions de sortie



Les portes simples que nous avons vues jusqu'à maintenant permettent de construire des circuits de plus en plus complexes jusqu'aux microprocesseurs les plus puissants.

Nous allons voir quelques-uns des circuits les plus courants que l'on rencontre en Architecture des Systèmes.

III. LES CIRCUITS COMBINATOIRES

III.1 LE DEMI ADDITIONNEUR

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

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

0 + 0 = 00

0 +1 = 01

1 + 0 =01

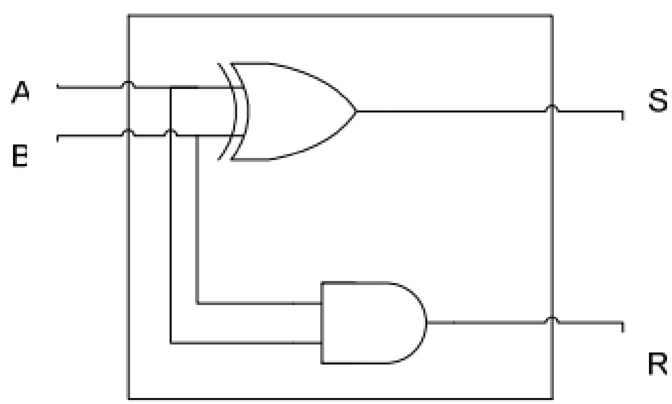
1 + 1 =10

La table de vérité associée

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

$$R = A.B$$
$$S = \overline{A}.B + A.\overline{B} = A \oplus B$$

Logigramme associé



III.2 ADDITIONNEUR COMPLET

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

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

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}

: le retenue entrante sur un bit. Il

possède deux sorties :

S: la somme

R_i : la retenue sortante

Table de vérité d'un additionneur complet sur un bit

a_i	b_i	r_{i-1}		r_i	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

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

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

Si on veut simplifier les deux formules logique de Si et Ri on obtient

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

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

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

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

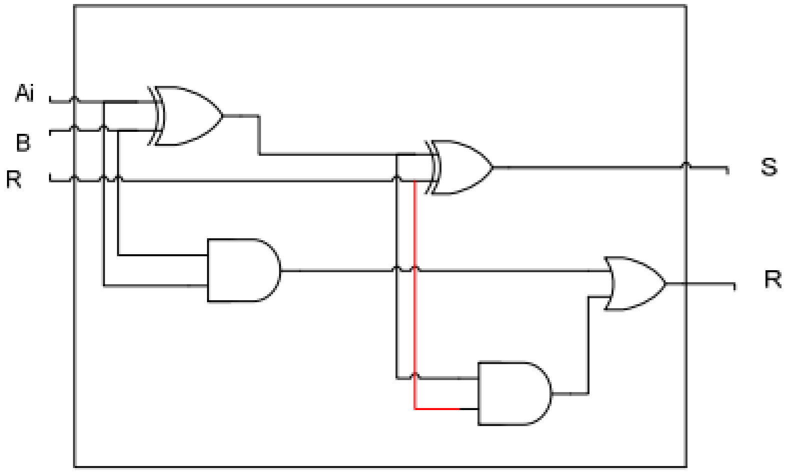
$$R_i = \overline{A}_i B_i R_{i-1} + A_i \overline{B}_i R_{i-1} + A_i B_i \overline{R}_{i-1} + A_i B_i R_{i-1}$$

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

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

Organigramme d'un additionneur complet

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



III.3 LE DEMI SOUSTRACTEUR

Le demi soustracteur est un circuit combinatoire qui permet de réaliser la soustraction arithmétique de deux nombres A et B sur un bit. A la sotie on va avoir la différence D et l'Emprunt Br (borrow).

En binaire la soustraction sur un seul bit se fait de la manière suivante:

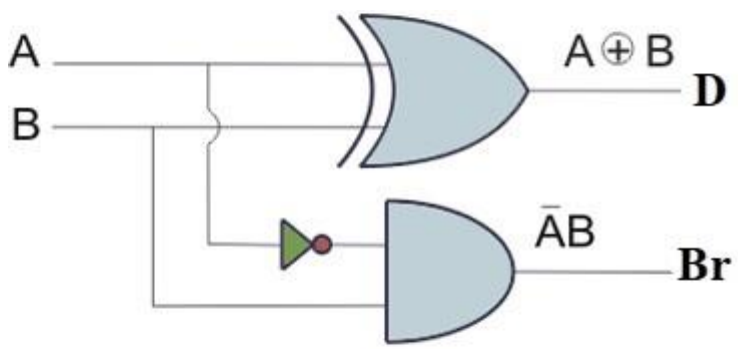
$$0 - 0 = 0$$
$$0 - 1 = 1 \quad \text{Avec emprunt (Borrow)}$$
$$1 - 0 = 0$$
$$1 - 1 = 0.$$

La table de vérité associée

A	B	Différence (D)	borrow (Br)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = A \oplus B$ et $Br = \bar{A}B$

Logigramme associé



III.4 LE SOUSTRACTEUR COMPLET

Le soustracteur complet un bit possède 3 entrées :

A_i : le premier nombre sur un bit.

B_i : le deuxième nombre sur un bit.

InBr: l’emprunt Entrant (Input Borrow)

D: Différence

Br : L’emprunt

Table de vérité d'un Soustracteur complet sur un bit

Ai	Bi	Input borrow (InBr)	Difference (D)	Borrow (Br)
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

$$D = \bar{A}\,\bar{B}\,InBr + \bar{A}\,B\,\overline{InBr} + A\,\bar{B}\,\overline{InBr} + A\,B\,InBr$$

$$D = \bar{A}\,(\bar{B}\,InBr + B\,\overline{InBr}) + A\,(\bar{B}\,\overline{InBr} + B\,InBr)$$

$$D = \bar{A}\,(B \oplus InBr) + A\,\overline{(B \oplus InBr)}$$

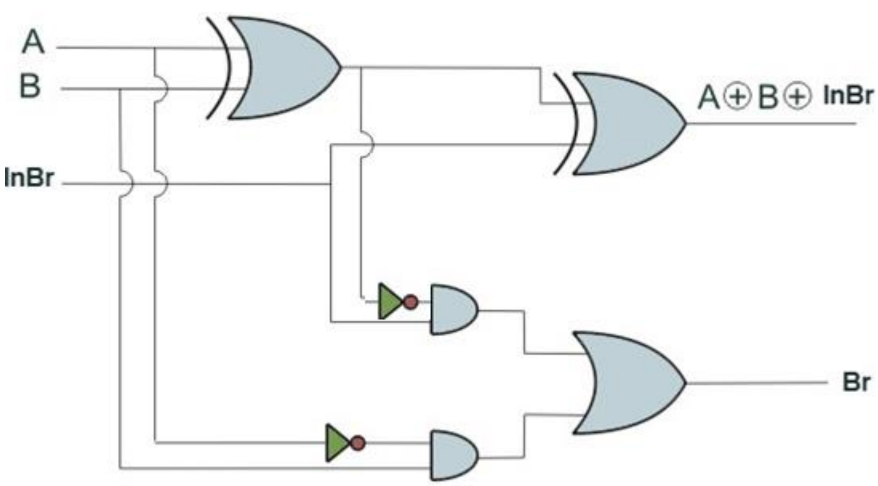
$$D = A \oplus B \oplus InBr$$

$$Br = \bar{A}\,\bar{B}\,InBr + \bar{A}\,B\,\overline{InBr} + \bar{A}\,B\,InBr + A\,B\,InBr$$

$$Br = (\bar{A}\,\bar{B} + AB)\,InBr + \bar{A}\,B\,(\overline{InBr} + InBr)$$

$$Br = \overline{(A \oplus B)}\,InBr + \bar{A}\,B$$

Organigramme d'un soustracteur complet



III.4 LE COMPARATEUR UN BIT

Un comparateur logique est un circuit logique qui effectue la comparaison entre deux nombres binaires généralement notés A et B. IL possède 3 sorties possibles notées :

A = B (A égal à B)

A > B (A est strictement supérieur au nombre B)

et A < B (A est strictement inférieur au nombre B)

Dont :

Si A = B, la sortie A = B passe à l'état 1 tandis que les sorties A > B et A < B passent à l'état 0.

Si le nombre A est strictement supérieur au nombre B, alors la sortie A> B passe à l'état 1 tandis que les sorties A = B et A < B passent à l'état 0.

Si le nombre A est strictement inférieur au nombre B, seule la sortie A < B passe à l'état 1.

Table de vérité d'un comparateur un bit

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

$$fs = A.\overline{B}$$

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

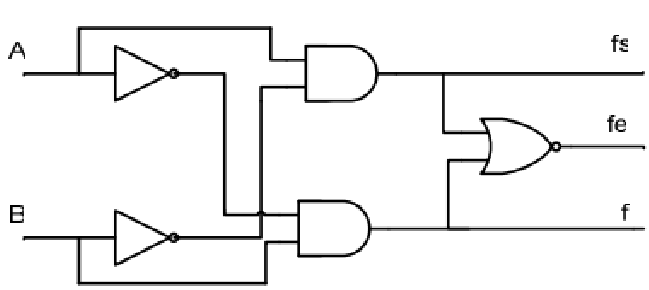
$$fe = \overline{A}\,\overline{B} + AB = \overline{A \oplus B} = \overline{fs + fi}$$

Logigramme d'un comparateur un bit

$$fs = A.\overline{B}$$

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

$$fe = \overline{fs + fi}$$



III.5 LE COMPARATEUR DEUX BITS

Il permet de faire la comparaison entre deux nombres A (a2a1) et B(b2b1) chacun sur deux bits.

1. A=B si

A2=B2 et A1=B1

$$fe = (\overline{A2 \oplus B2}).(\overline{A1 \oplus B1})$$

2. A>B si

A2 > B2 ou (A2=B2 et A1>B1)

$$fs = A2.\overline{B2} + (\overline{A2 \oplus B2}).(A1.\overline{B1})$$

3. A<B si

A2 < B2 ou (A2=B2 et A1<B1)

$$fi = \overline{A2}.B2 + (\overline{A2 \oplus B2}).(\overline{A1}.B1)$$

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

III.6 LE MULTIPLEXEUR

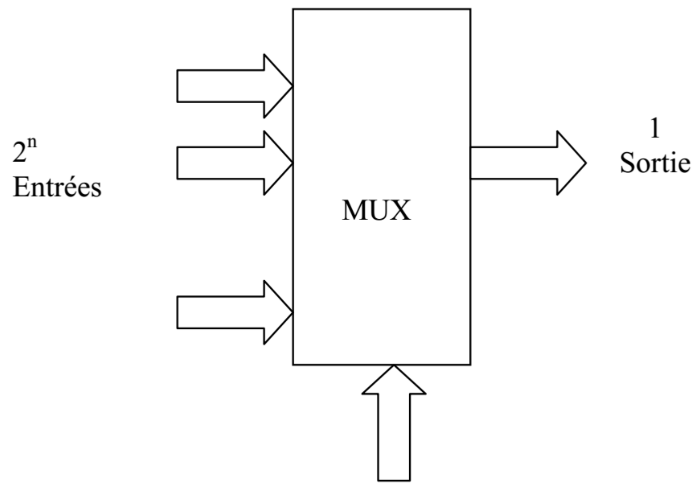
Un multiplexeur est circuit logique ayant plusieurs entrées de données, mais seulement une sortie qui communique les données.

L'aiguillage des données d'entrée vers la sortie est commandé par les entrées Select (ou entrée d'adresse).

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)**



Multiplxeur 4→1

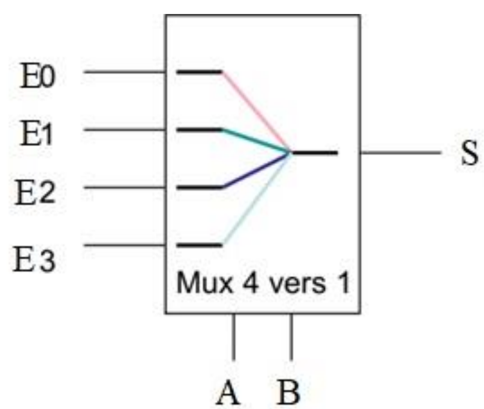
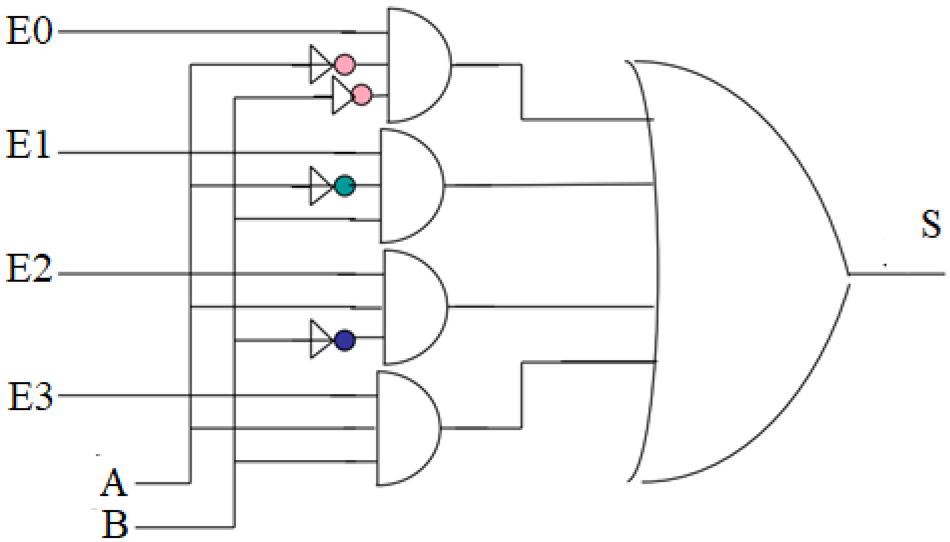


Table de vérité associée

A	B	S
0	0	E_0
0	1	E_1
1	0	E_2
1	1	E_3

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

Logigramme associé



Dans ce cas, il y a 4 entrées E_0, E_1, E_2, E_3 qui sont transmises à la sortie selon le choix indiqué par l'une des quatre combinaisons possibles des sorties de sélection A et B.

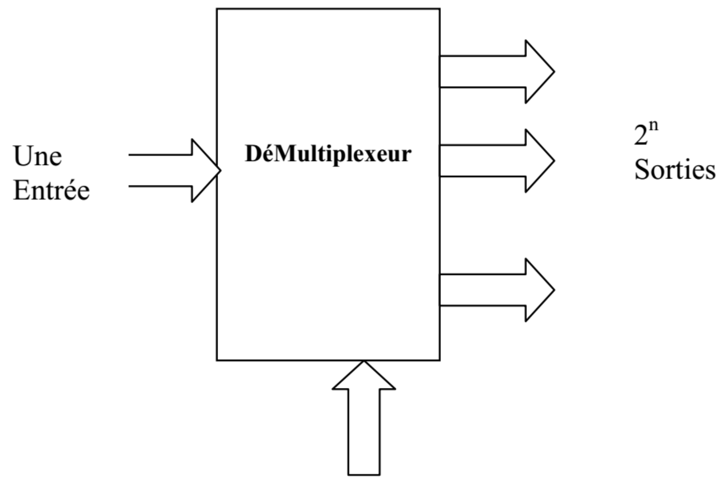
III.7 LE DEMULTIPLEXEUR

Un démultiplexeur est circuit logique qui a une seule entrée et N voies de sortie. Il reçoit les données d'entrée et choisit de les diriger vers une des N voies de sortie possibles,

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 posséde :

- **Une seule entrée**
- **2^n sorties**
- **N entrées de sélection**



Demultiplxeur 4→1

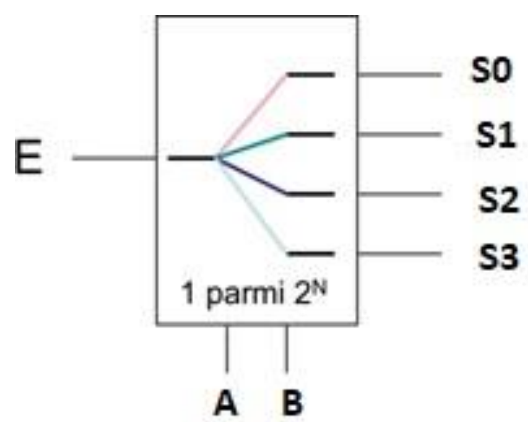
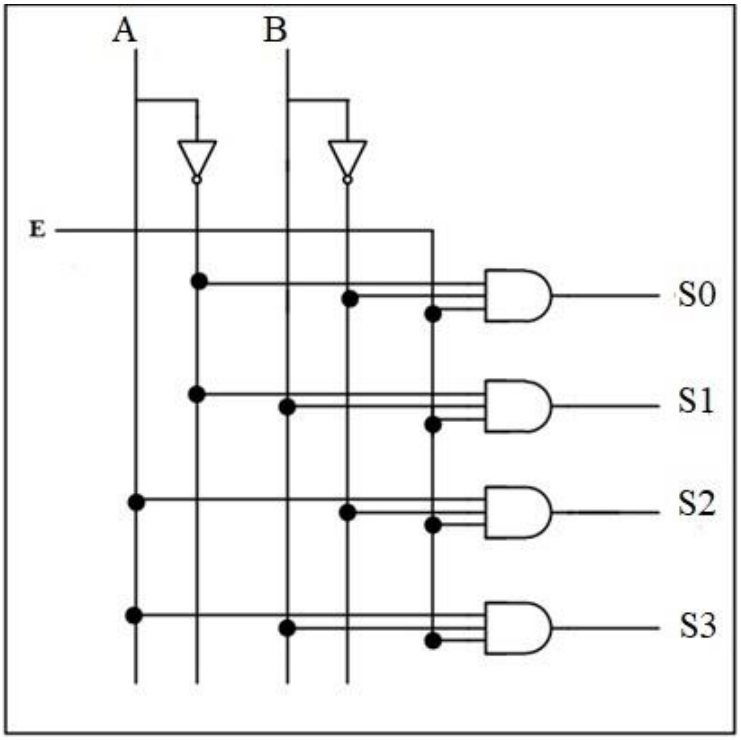


Table de vérité associée

A	B	S0	S1	S2	S3
0	0	**E**	0	0	0
0	1	0	**E**	0	0
1	0	0	0	**E**	0
1	1	0	0	0	**E**

Logigramme associé



$S0 = \overline{A}.\overline{B}\ (E)$

$S1 = \overline{A}.B\ (E)$

$S2 = A.\overline{B}\ (E)$

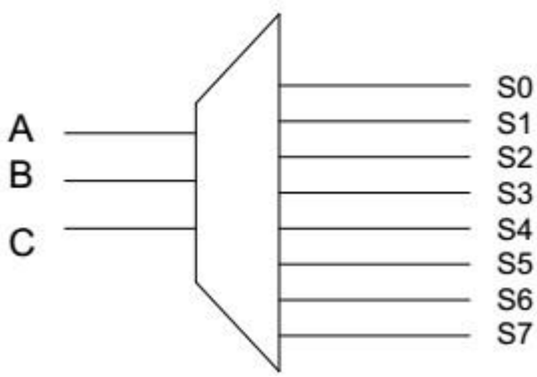
$S3 = A.B\ (E)$

III.8 LE DECODEUR BINAIRE

C'est un circuit combinatoire qui est constitué de :

- **N : entrées de données**
- **2^n sorties**

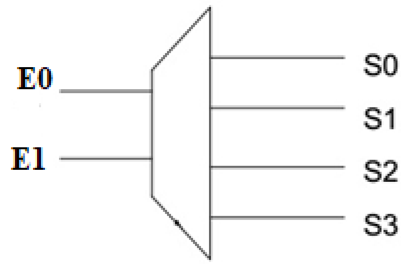
Pour chaque combinaison en entrée une seule sortie est active à la fois



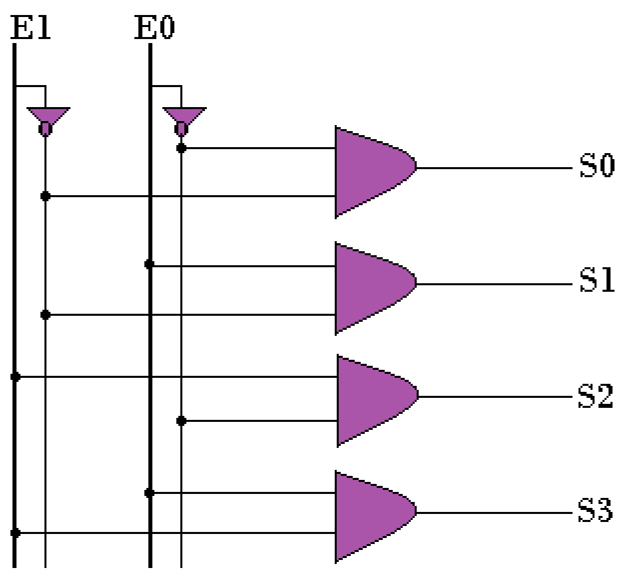
UN DECODEUR 3→8

TABLE DE VERITE D'UN DECODEUR 2→4

E_1	E_2	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



$S_0 = \overline{E1}.\overline{E0}$, $S_1 = \overline{E1}.E0$, $S_2 = E1.\overline{E0}$, $S_3 = E1\,E0$

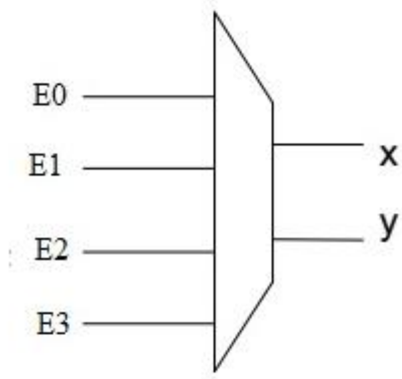


III.9 LE ENCODEUR BINAIRE

Il joue le rôle inverse d'un décodeur, il possède :

- **2^n entrées**
- **N sorties**

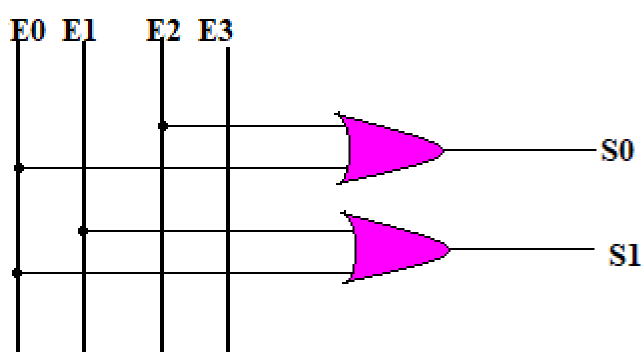
Pour chaque combinaison en entrée on va avoir son numéro (en binaire) à la sortie



Encodeur 4→2

E3	E2	E1	E0	S_1	S_0
0	0	0	0	0	0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

S1 = E1 + E0   **S0 = E2 + E0**



III.10 L'ENCODEUR DE PRIORITE

Souvent dans un codeur, dit alors codeur à priorité, si plusieurs entrées sont actives simultanément, dans ce cas on doit choisir avec quelle priorité de bits d'entrée nous allons choisir dans ce qui **suit c'est un encodeur de priorité de poids forts .**

Table de vérité

E3	E2	E1	E0	S1	S0
0	0	0	0	**0**	**0**
0	0	0	1	**0**	**0**
0	0	1	0	**0**	**1**
0	0	1	1	**0**	**1**
0	1	0	0	**1**	**0**
0	1	0	1	**1**	**0**
0	1	1	0	**1**	**0**
0	1	1	1	**1**	**0**
1	0	0	0	**1**	**1**
1	0	0	1	**1**	**1**
1	0	1	0	**1**	**1**
1	0	1	1	**1**	**1**
1	1	0	0	**1**	**1**
1	1	0	1	**1**	**1**
1	1	1	0	**1**	**1**
1	1	1	1	**1**	**1**

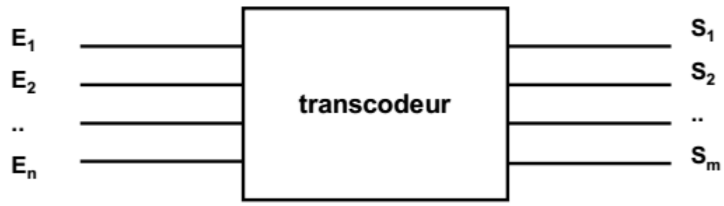
Après simplification (Tableau de Karnaugh)

$S_1 = E2 + E3$

$S_0 = E3 + \overline{E2}\,E1$

III.11 LE 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.



Un **transcodeur** transforme une information disponible en entrée sous forme donnée (généralement un code) en la même information, mais sous une autre forme (généralement un autre code)

Les deux plus importantes applications des transcodeurs sont : la conversion de code et l'affichage par segment.