

CHAPITRE 4

Les Circuits combinatoires en Electronique numérique

Plan du Chapitre

- Codeurs
- Décodeurs
- Multiplexeurs
- Démultiplexeurs
- Comparateurs

Codeurs

Un codeur ou encodeur est un circuit logique qui possède 2^N voies d'entrée dont une seule est active et N voies de sortie

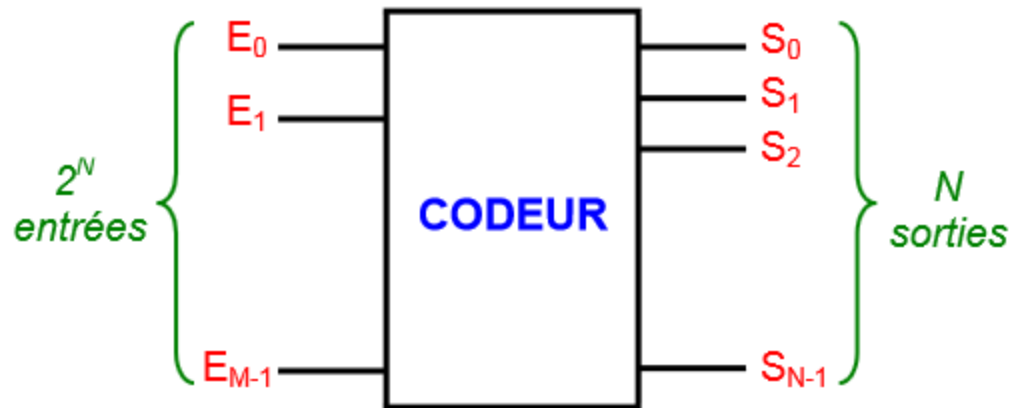


Schéma fonctionnel d'un codeur

Codeur B.C.D

Il s'agit du codeur B.C.D à dix voies d'entrée (les chiffres décimaux), et qui produit en sortie l'équivalent binaire du chiffre décimal appliqué à l'entrée.

Entrées										Sorties			
Chiffre décimal	E ₉	E ₈	E ₇	E ₆	E ₅	E ₄	E ₃	E ₂	E ₁	S ₃	S ₂	S ₁	S ₀
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	0	0	1	0	0	0	1	0
3	0	0	0	0	0	0	1	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0	1	0	1
6	0	0	0	1	0	0	0	0	0	0	1	1	0
7	0	0	1	0	0	0	0	0	0	0	1	1	1
8	0	1	0	0	0	0	0	0	0	1	0	0	0
9	1	0	0	0	0	0	0	0	0	1	0	0	1

Table de vérité du codeur B.C.D

Les équations logiques associées aux sorties Si se déduisent facilement à partir de la table de vérité ci-dessus. On a alors :

$$S_0 = E_1 + E_3 + E_5 + E_7 + E_9$$

$$S_1 = E_2 + E_3 + E_6 + E_7$$

$$S_2 = E_4 + E_5 + E_6 + E_7$$

$$S_3 = E_8 + E_9$$

Codeur de priorité

C'est un dispositif qui réalise le codage du numéro le plus élevé dans le cas où plusieurs entrées seraient actionnées. La table de vérité de ce codeur de priorité est donnée par le tableau suivant :

<i>Entrées</i>									<i>Sorties</i>			
E_9	E_8	E_7	E_6	E_5	E_4	E_3	E_2	E_1	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	x	0	0	1	0
0	0	0	0	0	0	1	x	x	0	0	1	1
0	0	0	0	0	1	x	x	x	0	1	0	0
0	0	0	0	1	x	x	x	x	0	1	0	1
0	0	0	1	x	x	x	x	x	0	1	1	0
0	0	1	x	x	x	x	x	x	0	1	1	1
0	1	x	x	x	x	x	x	x	1	0	0	0
1	x	x	x	x	x	x	x	x	1	0	0	1

x état indifférent

Table de vérité du codeur de priorité

$$S_0 = E_9 + \overline{E}_8.E_7 + \overline{E}_8.\overline{E}_6.E_5 + \overline{E}_8.\overline{E}_6.\overline{E}_4.E_3 + \overline{E}_8.\overline{E}_6.\overline{E}_4.\overline{E}_2.E_1$$

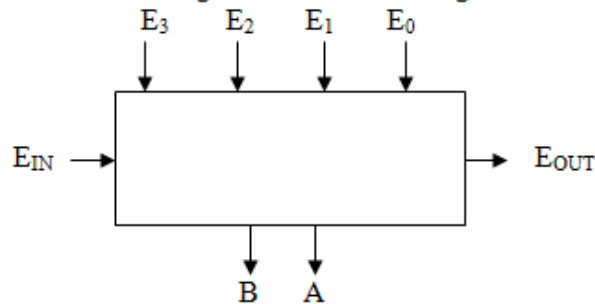
$$S_1 = \overline{E}_9.\overline{E}_8.E_7 + \overline{E}_9.\overline{E}_8.E_6 + \overline{E}_9.\overline{E}_8.\overline{E}_5.\overline{E}_4.E_3 + \overline{E}_9.\overline{E}_8.\overline{E}_5.\overline{E}_4.E_2$$

$$S_2 = \overline{E}_9.\overline{E}_8.E_7 + \overline{E}_9.\overline{E}_8.E_6 + \overline{E}_9.\overline{E}_8.E_5 + \overline{E}_9.\overline{E}_8.E_4$$

$$S_3 = E_9 + E_8$$

Exercice 3 :

Soit un circuit combinatoire à 5 lignes d'entrées et 3 lignes de sorties, comme le montre la figure ci-dessous.



Le fonctionnement est le suivant :

Lorsqu'une seule ligne d'entrée parmi E_0, E_1, E_2, E_3 , se trouve au niveau haut, son numéro est codé en binaire sur les sorties BA.

Si plusieurs lignes sont simultanément au niveau haut, le circuit code le numéro le plus élevé.

Si toutes les lignes d'entrée sont au niveau bas, le circuit code BA=00, mais on signale par EOUT=1 que ce code n'est pas validé. Dans tous les autres cas EOUT=0.

Le fonctionnement décrit jusqu'ici s'observe lorsque $E_{IN}=1$.

Si $E_{IN}=0$, on a : $B=A=E_{OUT}=0$.

- 1) Donner la table de vérité du codeur.
- 2) Donner les expressions logiques des sorties A, B et EOUT en fonction des entrées de E_0, E_1, E_2, E_3 et E_{IN} .
- 3) En déduire le circuit logique du codeur.

E_{IN}	E_3	E_2	E_1	E_0	B	A	EOUT
0	x	x	x	x	0	0	0
1	1	x	x	x	1	1	0
1	0	1	x	x	1	0	0
1	0	0	1	x	0	1	0
1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1

Pour Ein =1

E3E2 \ E1E0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

E_{IN}	E_3	E_2	E_1	E_0	B	A	E_{OUT}
0	x	x	x	x	0	0	0
1	1	x	x	x	1	1	0
1	0	1	x	x	1	0	0
1	0	0	1	x	0	1	0
1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1

$$B = Ein(E2 + E3)$$

Pour Ein =1

E3E2 \ E1E0	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$$A = Ein(E3 + \overline{E2}E1)$$

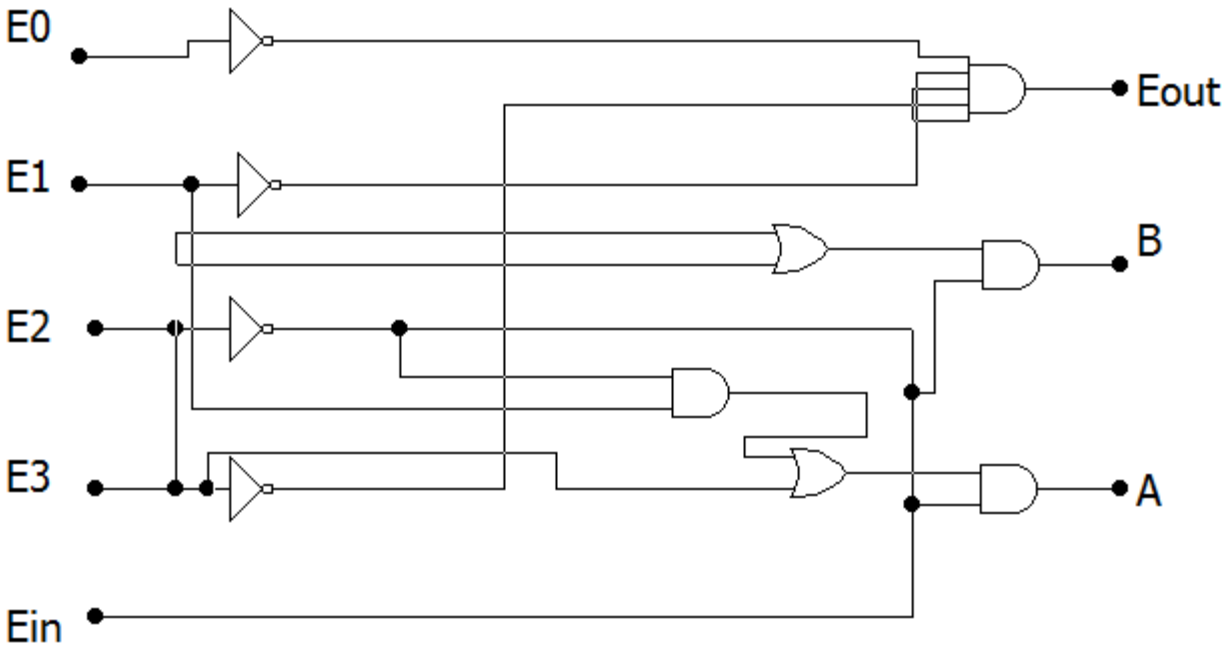
Pour Ein=1

E1E0 E3E2	00	01	11	10
00	1	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

E_3	E_2	E_1	E_0	B	A	E_{OUT}
0	x	x	x	0	0	0
1	1	x	x	1	1	0
1	0	1	x	1	0	0
1	0	0	1	0	1	0
1	0	0	0	0	0	0
1	0	0	0	0	0	1

$$E_{out} = E_{in}(\overline{E_3}\overline{E_2}\overline{E_1}\overline{E_0})$$

Logigramme



Décodeurs

Un décodeur est un circuit numérique qui possède N entrées et 2^N sorties. Pour chacune des combinaisons possibles des entrées, seule une ligne de sortie est validée. Les décodeurs sont souvent dotés d'une ou plusieurs entrées de validation E qui servent à valider son fonctionnement. Le schéma fonctionnel d'un décodeur à N bits d'entrée est donné par la figure suivante :

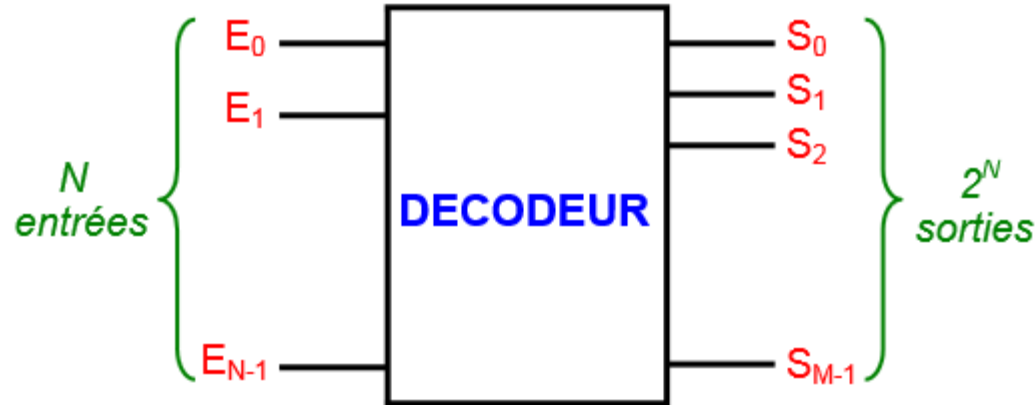


Schéma fonctionnel d'un décodeur

Décodeur 2 vers 4

Avec un décodeur à deux bits d'entrée, on peut avoir quatre combinaisons de sortie. Le décodeur comporte alors deux entrées A, B et quatre sorties S0, S1, S2, S3 validées par exemple à l'état bas. La table de vérité du décodeur est donnée par le tableau suivant :

Entrées		Sorties			
B	A	$\overline{S_3}$	$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

Table de vérité d'un décodeur 2 vers 4 actif à l'état bas

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

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

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

$$\overline{S_3} = B.A$$

Décodeur 3 vers 8

Le décodeur 3 vers 8 comporte trois entrées A, B, C et huit sorties S0, S1, S2.....S7 validées par exemple à l'état haut . La table de vérité du décodeur est donnée par le tableau suivant :

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

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

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

$$S_3 = \overline{C}.B.A$$

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

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

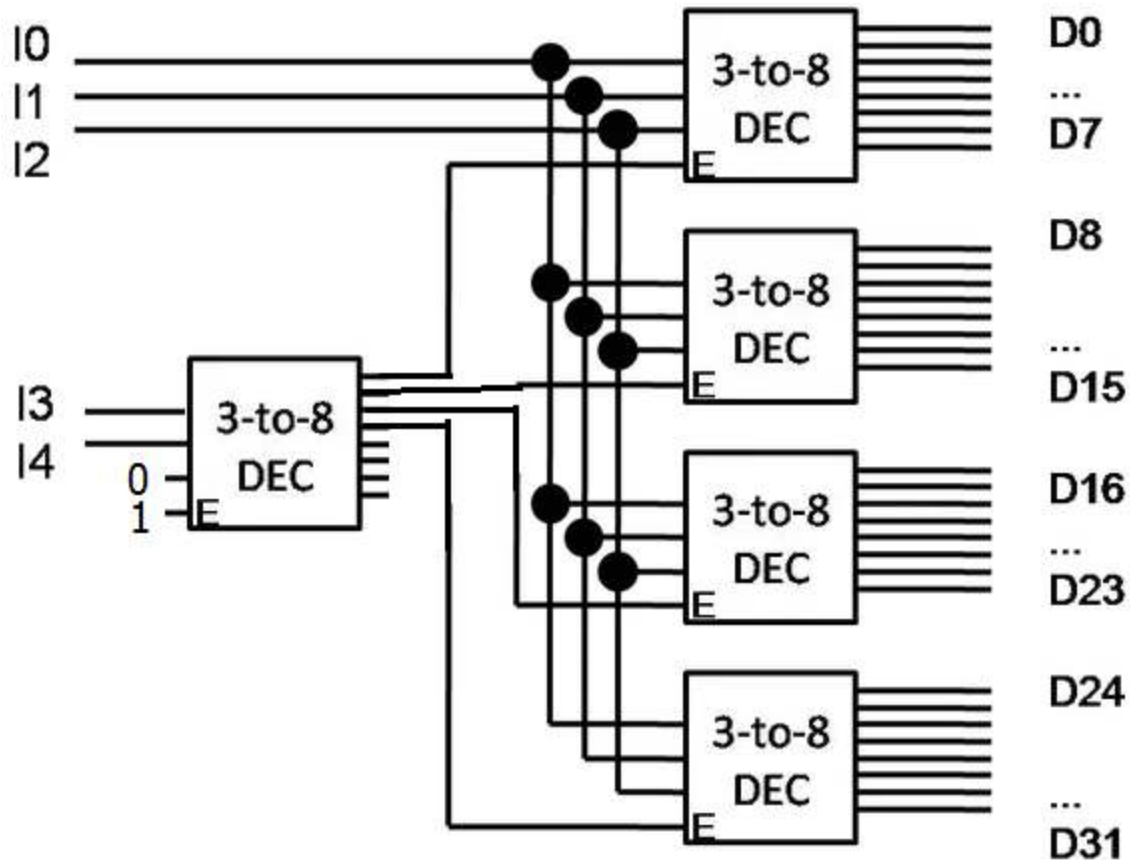
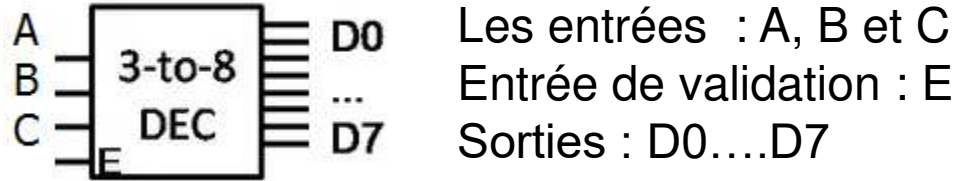
$$S_6 = C.B.\overline{A}$$

$$S_7 = C.B.A$$

Entrées			Sorties							
C	B	A	S ₇	S ₆	S ₅	S ₄	S ₃	S ₂	S ₁	S ₀
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Table de vérité d'un décodeur 3 vers 8 actif à l'état haut

Conception d'un Décodeur 5 vers 32 à partir de 3 vers 8 (5 circuits)



Transcodeurs

Un transcodeur est un dispositif qui permet de faire passer une information écrite dans le code C1 à un autre code C2. Les deux plus importantes applications des transcodeurs sont : la conversion de code et l'affichage par segments.

Conversion de code : Transcodeur Gray- binaire

$$B_2 = G_2$$

$$B_1 = G_2 \bar{G}_1 + \bar{G}_2 G_1 = G_2 \oplus G_1$$

$$B_0 = G_2 \oplus G_1 \oplus G_0$$

Gray			Binaire		
G ₂	G ₁	G ₀	B ₂	B ₁	B ₀
0	0	0	0	0	0
0	0	1	0	0	1
0	1	1	0	1	0
0	1	0	0	1	1
1	1	0	1	0	0
1	1	1	1	0	1
1	0	1	1	1	0
1	0	0	1	1	1

Table de conversion Gray-binaire

Transcodeur BINAIRE-GRAY avec XOR

Décimal	Code Binaire				Code GRAY			
	B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

B ₃ B ₂	B ₁ B ₀			
	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$$G_1 = B_2 \oplus B_1$$

B ₃ B ₂	B ₁ B ₀			
	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$$G_0 = B_1 \oplus B_0$$

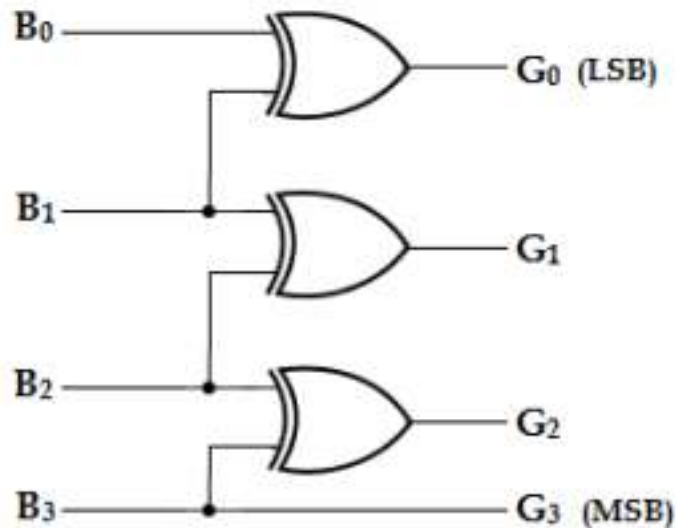
B ₃ B ₂	B ₁ B ₀			
	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$$G_3 = B_3$$

B ₃ B ₂	B ₁ B ₀			
	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

$$G_2 = B_3 \oplus B_2$$

Circuit Logique Binaire-Gray



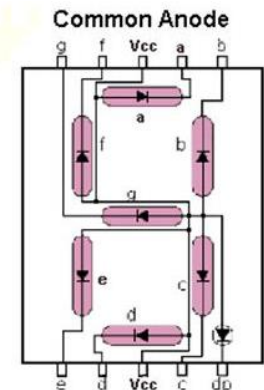
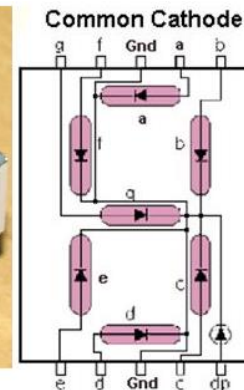
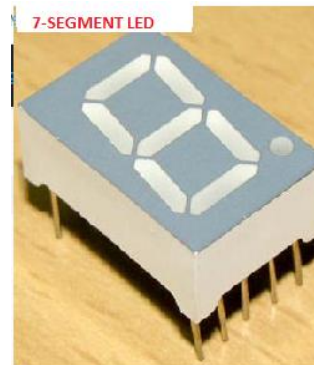
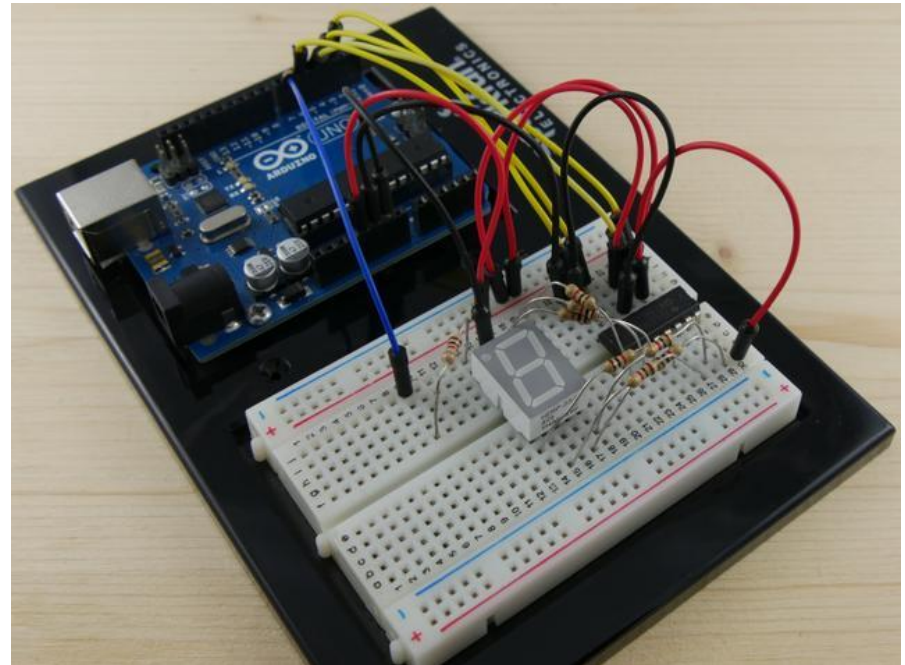
Exercice à faire à la maison : Table de vérité, Equations simplifiées et circuit logique d'un transcodeur GRAY (G₃,G₂,G₁,G₀)-Binaire (B₃,B₂,B₁,B₀)

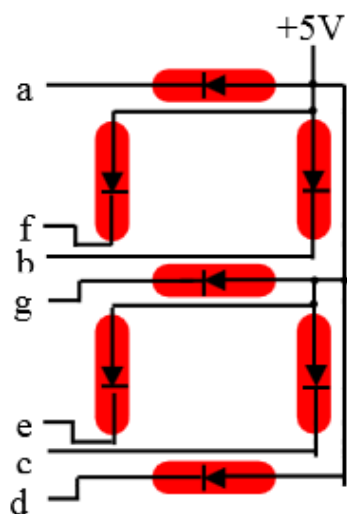
Transcodeur BCD-7 segments

Afficheur 7 segments

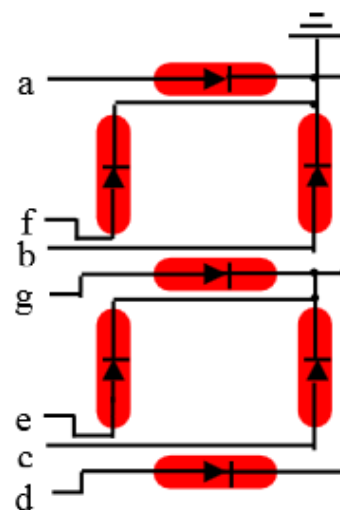
Un domaine d'application considérable des transcodeurs est celui de la conversion de données binaires en une forme se prêtant à un affichage numérique. Les dix chiffres 0 à 9 sont affichés au moyen d'un dispositif appelé afficheur à 7 segments. Cet afficheur est un ensemble de diodes électroluminescentes (D.E.L) disposés comme le montre la figure ci-dessous. On distingue deux types d'afficheurs :

- l'afficheur à anodes communes et celui à cathodes communes. Dans le premier cas, toutes les anodes sont reliées à un même point, mis à 5V, de façon à rendre lumineux le segment qui présente sur sa cathode un niveau bas.
- Dans le cas de l'afficheur à cathodes communes, toutes les cathodes sont reliées un point commun qui doit être à la masse, de façon que pour allumer un segment, il faut lui appliquer un niveau haut sur son anode





Afficheur à anodes communes



Afficheur à cathodes communes

Affici

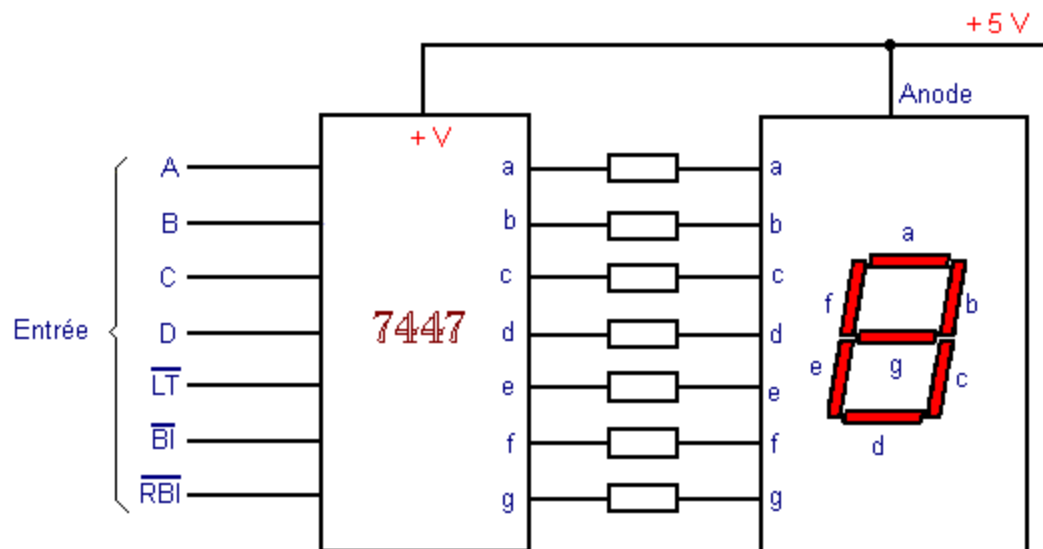
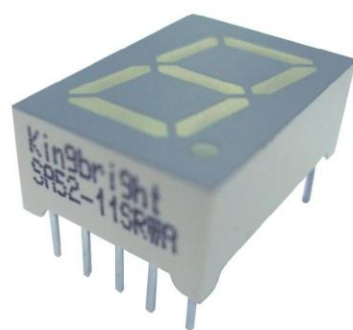
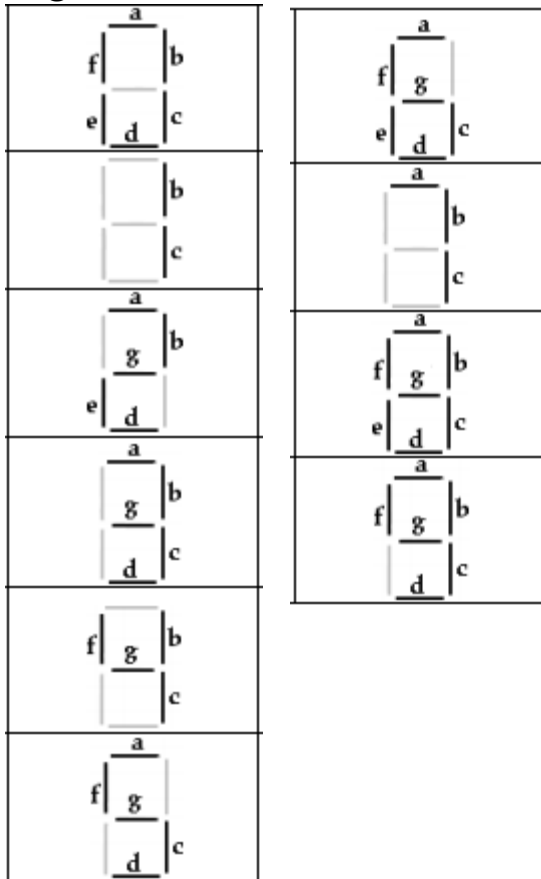


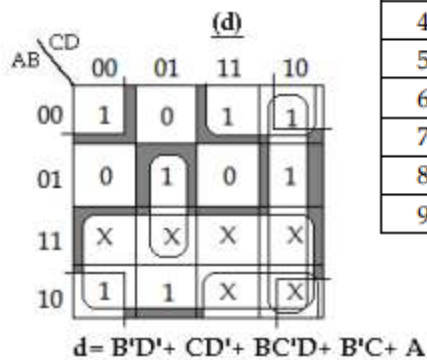
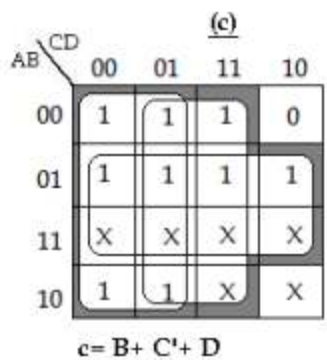
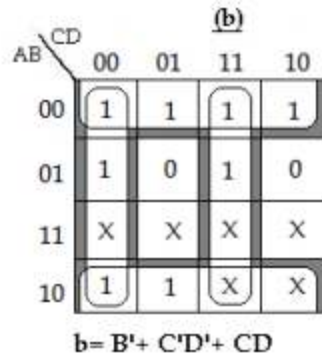
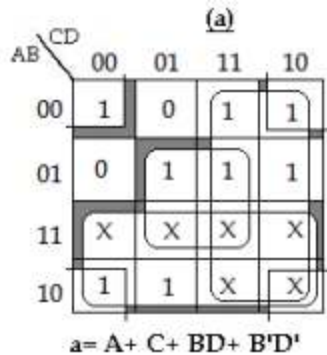
Fig. 37. - Afficheur 7 segments commandé par un décodeur 7447.

Synthèse d'un transcodeur BCD-7 segments

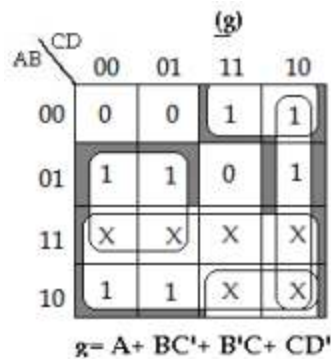
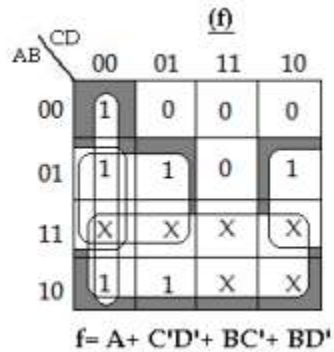
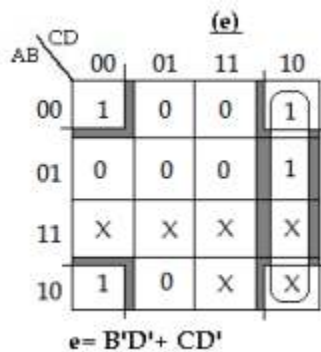
Le tableau ci-dessous donne la table de vérité détaillant le fonctionnement du transcodeur BCD-7 segments permettant l'affichage des différents chiffres décimaux. Les variables d'entrée A, B, C, D sont écrites en B.C.D, les variables de sortie a, b, c, d, e, f, g, correspondent à chacun des segments de l'afficheur.



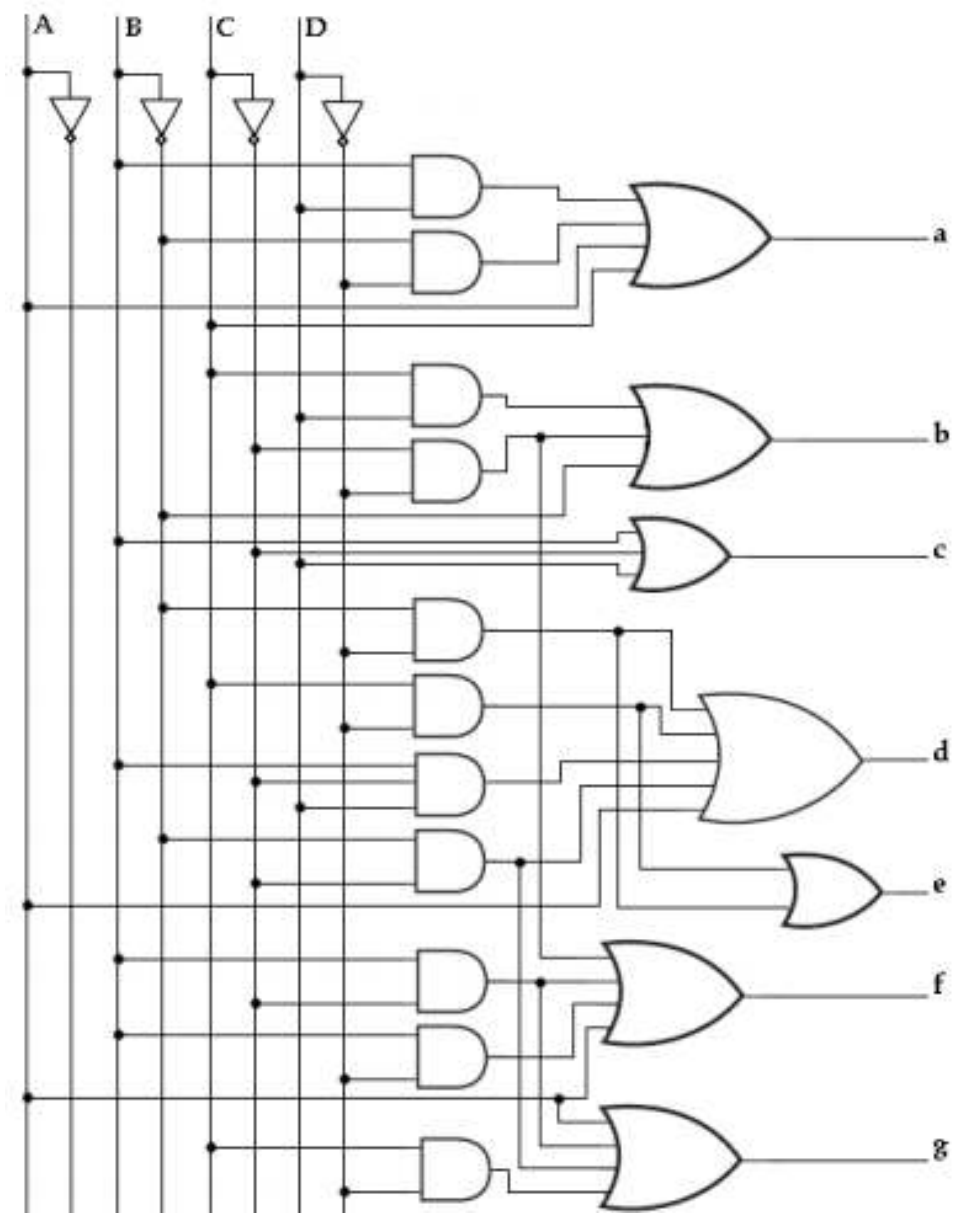
	Code BCD				Code 7 segments						
Digit	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1



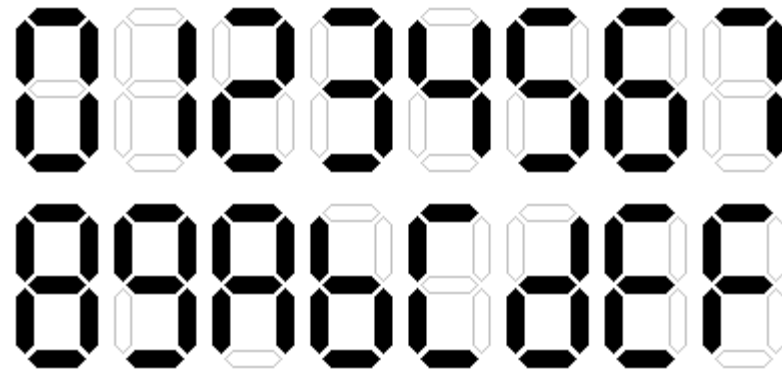
	Code BCD				Code 7 segments						
Digit	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1



Circuit logique de décodeur BCD-7 segments



Exercice à faire à la maison :
Concevoir un décodeur
Hexadécimal-7segments :Table
de vérité, Equations simplifiées et
circuit logique



Multiplexeurs

Un multiplexeur (MUX) est un circuit logique qui possède 2^N entrées d'informations (I_i), N entrées de sélection (S_i) et une sortie unique Z . Sa fonction consiste à effectuer l'aiguillage de l'une des entrées d'information vers la sortie en fonction du code d'adresse appliqué sur les entrées de sélection.

On pourra de plus trouver une entrée de validation E . Si cette broche n'est pas validée, la sortie Z est égale à 1 (ou 0), et ce quelle que soit l'adresse appliquée et le niveau des entrées I_i , par contre quand cette broche est validée, le multiplexeur délivre sur sa sortie Z l'état de l'entrée adressée.

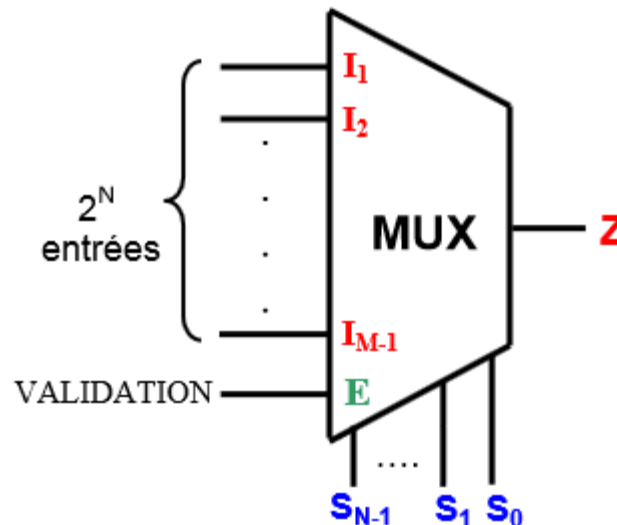
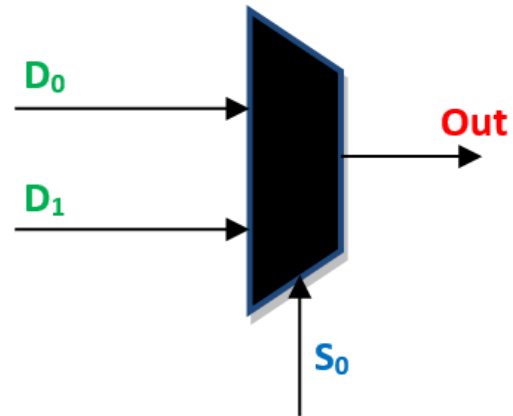


Schéma fonctionnel d'un multiplexeur

Multiplexeur à 2 entrées

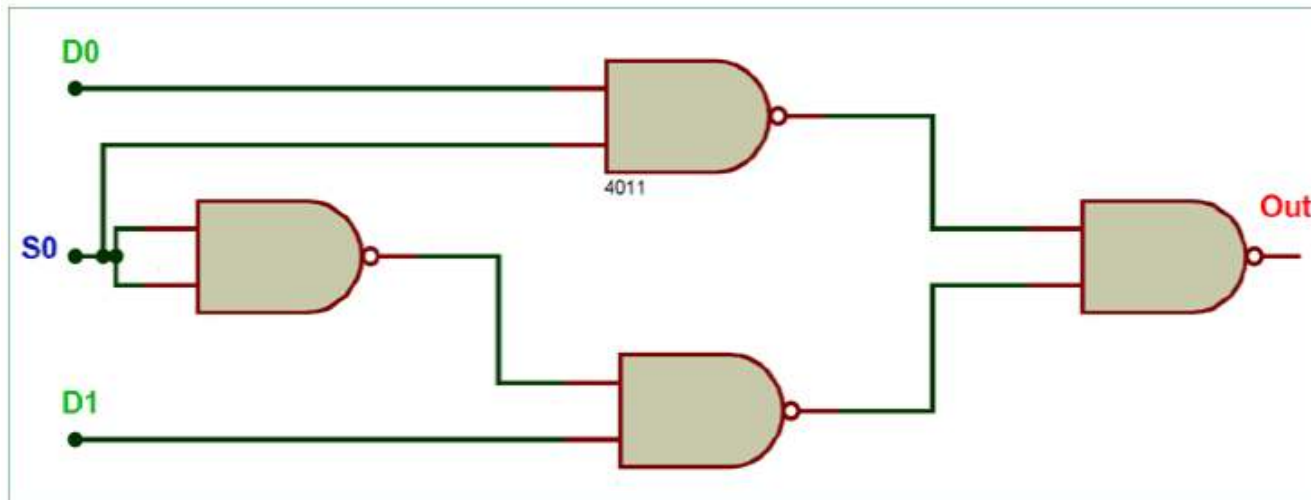


$$Out = D_0 \overline{S_0} + D_1 S_0$$

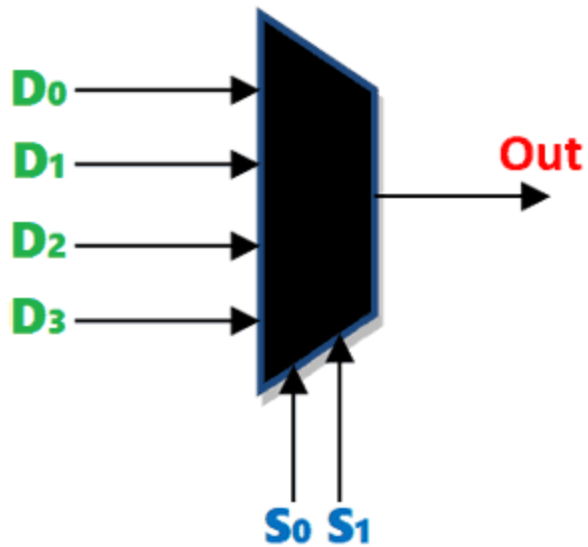
S_0	D_0	D_1	Out
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



S_0	Out
0	D_0
1	D_1



Multiplexeur à 4 entrées

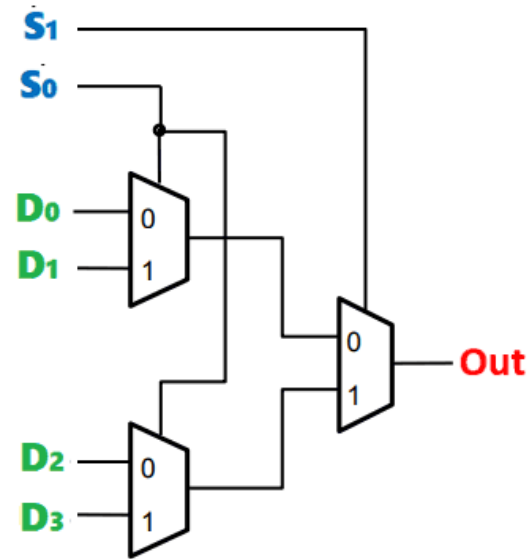


S_1	S_0	Out (f)
0	0	D0
0	1	D1
1	0	D2
1	1	D3

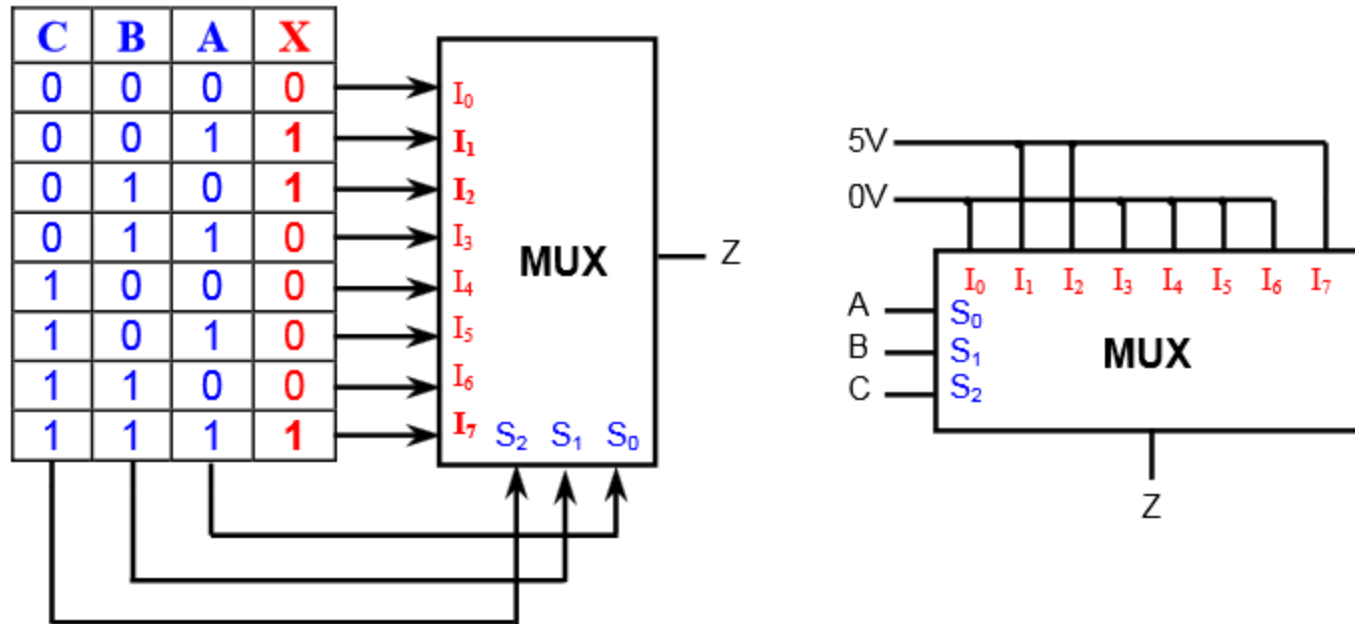
out

$$= D0\overline{S0}\overline{S1} + D1S0\overline{S1} + D2\overline{S0}S1 + D3S0S1$$

$$= \overline{S1}(D0\overline{S0} + D1S0) + S1(D2\overline{S0} + D3S0)$$

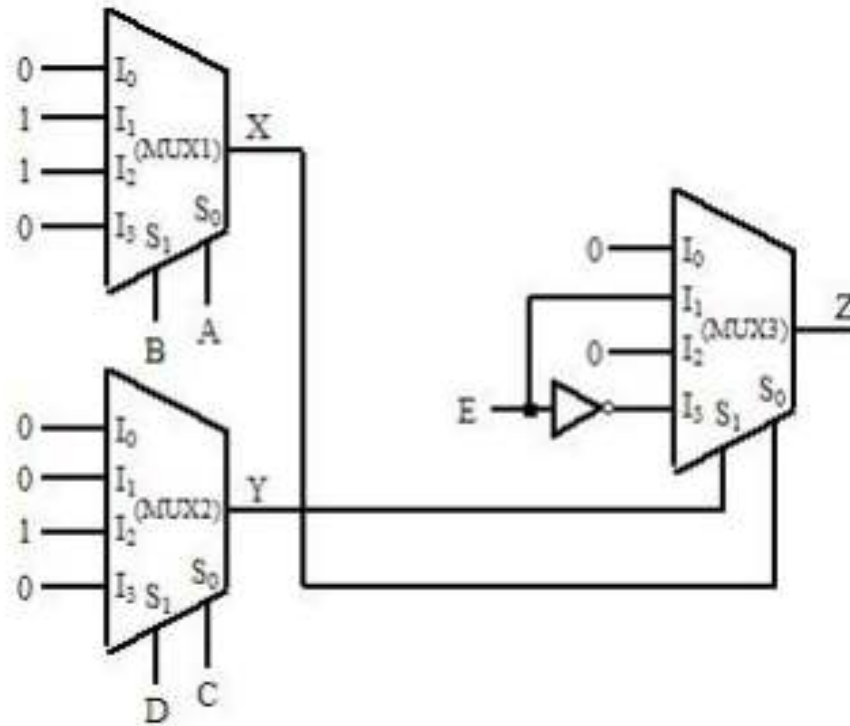


L'exemple de la figure ci-dessous est celui de la génération d'une fonction logique à trois variables décrite par sa table de vérité. On doit utiliser pour cela un multiplexeur à huit entrées (2^3).



Synthèse d'une fonction logique à l'aide d'un multiplexeur

- Soit le montage de la figure ci-dessous, réalisé à partir de trois multiplexeurs MUX-1, MUX-2 et MUX-3 chacun à 4 entrées.



Exercice à
faire à la
maison

- 1-Donner l'expression logique de Z en fonction de A, B, C, D et E.
- 2-Ecrire cette expression qu'avec des OU exclusifs

Démultiplexeurs

Un démultiplexeur (DEMUX) est un circuit logique qui possède une seule entrée I et 2^N voies de sortie. Il transmet les données d'entrée (données séries) vers une des 2^N voies de sortie selon l'adresse du code appliqué sur les N entrées de sélection, il fonctionne comme un commutateur. Certains démultiplexeurs présentent une entrée de validation E qui permet, quand elle est validée, de transférer les données de l'entrée vers les sorties sélectionnées, et d'imposer zéro ou un à l'ensemble des sorties quand elle n'est pas validée. Les données qui parviennent en série à l'entrée du démultiplexeur seront fournies en parallèle en sortie

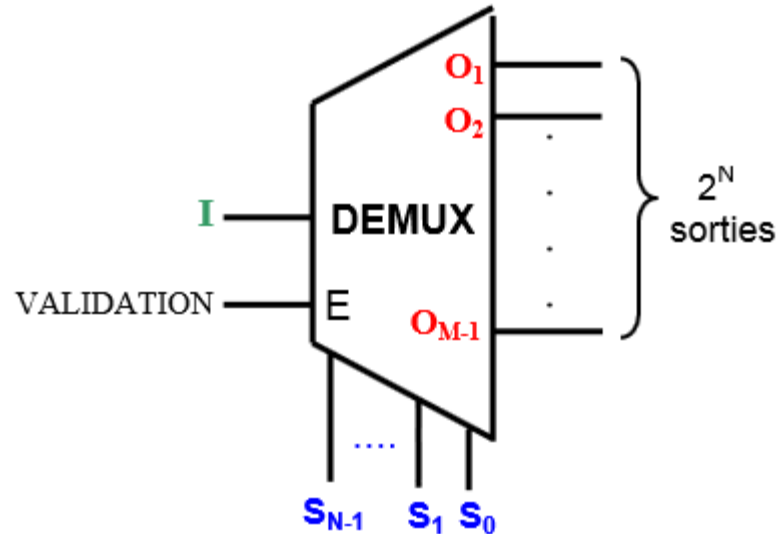


Schéma fonctionnel d'un démultiplexeur

Comparateurs

Un comparateur est un circuit dont la fonction est d'indiquer si un nombre binaire est inférieur, égal ou supérieur à un autre nombre binaire.

Comparateur de deux nombres binaires à 1 bit

a_i	b_i	$S_i(>)$	$I_i(<)$	$E_i(=)$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

Comparaison de deux nombres binaires à 1 bit

$$S_i = a_i \cdot \overline{b_i}$$

$$I_i = \overline{a_i} \cdot b_i$$

$$E_i = a_i \cdot b_i + \overline{a_i} \cdot \overline{b_i} = \overline{a_i \oplus b_i}$$

Comparateur de deux nombres binaires à N bits

Pour comparer deux nombres binaires de N bits A et B, il faut effectuer une comparaison bit par bit, en commençant par les bits de poids le plus fort, s'ils sont égaux on passe aux bits de poids immédiatement inférieur et ainsi de suite.

Soit à comparer les deux nombres binaires suivants :

$$A = a_n a_{n-1} \dots a_1 a_0 \quad \text{et} \quad B = b_n b_{n-1} \dots b_1 b_0$$

- Détermination si $A > B$

$$a_n > b_n \rightarrow S_n = 1$$

$$\text{ou } a_n = b_n \rightarrow E_n = 1 \text{ et } a_{n-1} > b_{n-1} \rightarrow S_{n-1} = 1$$

$$\text{ou } a_n = b_n \rightarrow E_n = 1 \text{ et } a_{n-1} = b_{n-1} \rightarrow E_{n-1} = 1 \text{ et } a_{n-2} > b_{n-2} \rightarrow S_{n-2} = 1$$

.

.

$$\text{ou } a_n = b_n \rightarrow E_n = 1 \text{ et } a_{n-1} = b_{n-1} \rightarrow E_{n-1} = 1 \dots \text{et } a_1 = b_1 \rightarrow E_1 = 1 \\ \text{et } a_0 > b_0 \rightarrow S_0 = 1$$

Donc l'expression booléenne de S est:

- Détermination si $A < B$

I_0

En utilisant le même raisonnement que précédemment on a:

$$I = I_n + E_n \cdot I_{n-1} + E_n \cdot E_{n-1} \cdot I_{n-2} + \dots + E_n \cdot E_{n-1} \dots E_2 \cdot E_1 \cdot I_0$$

- Détermination si $A = B$

Il faut que l'on ait: $a_n = b_n \text{ et } a_{n-1} = b_{n-1} \text{ et } \dots \text{ et } a_0 = b_0$

Correction TD3

Exercice 1

$$1- V = \bar{a}bc + \bar{a}b\bar{c} + a\bar{b}c + a\bar{b}\bar{c}$$

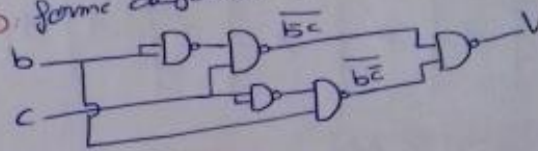
$$2- V = bc(\bar{a}+a) + b\bar{c}(\bar{a}+a) \\ = bc + b\bar{c}$$

$$V = b \oplus c$$

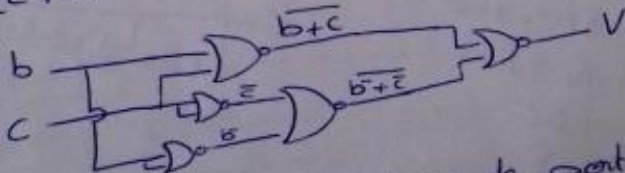
$$3- \begin{array}{c|cccc} ab \backslash c & 00 & 01 & 11 & 10 \\ \hline 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{array} \Rightarrow V = b\bar{c} + bc \\ V = b \oplus c$$

NAND: forme disjonctive (somme des produits)

$$4- V = bc + b\bar{c} \\ V = \overline{\overline{bc} \cdot \overline{b\bar{c}}}$$



$$5- V = bc + b\bar{c} \\ = (b+b)(c+b)(b+\bar{c})(c+\bar{c}) \\ = (c+b)(b+\bar{c}) \\ V = \overline{(c+b) + (b+\bar{c})}$$



Pour construire le logigramme à base des portes logiques NAND

- Il faut décomposer d'une forme : disjonctive (NAND)/conjonctive
- Ajouter le barreau à la 2^e (théorème d'involution).
- Décomposer la barre inférieure (théorème de Morgan)
- Régler le problème des inverseurs (théorème d'impotence).

Exercice N°1 :

On définit l'expression logique V par sa table de vérité :

a	b	c	V
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

1. Écrire l'expression logique de V.
2. Simplifier algébriquement V.
3. Vérifier la simplification de V grâce au tableau de Karnaugh
4. Construire le logigramme de V en utilisant uniquement des opérateurs logiques NAND à deux entrées.
5. Construire le logigramme de V en utilisant uniquement des opérateurs logiques NOR à deux entrées.

Exercice 2.

$$\begin{aligned}
 1- F_{(abcd)} &= (a+b)(c+\bar{d}) \\
 &= (a+b)(\bar{c}+d) \\
 &= a\bar{c}d + b\bar{c}d + abc + abd \\
 &= a\bar{c}d + \bar{a}b\bar{c}d + ab\bar{c}d + abc + abd \\
 &= a\bar{c}d + \bar{a}b\bar{c}d + ab\bar{c}d + abc + abd
 \end{aligned}$$

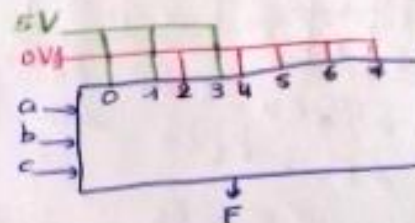
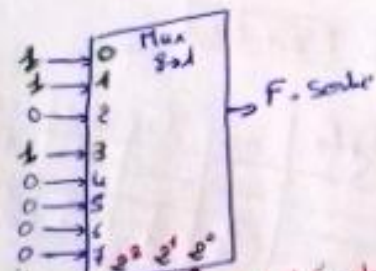
$$F_{(abcd)} = \Sigma(8, 9, 14)$$

2- $F_{(abcd)}$ peut s'écrire :

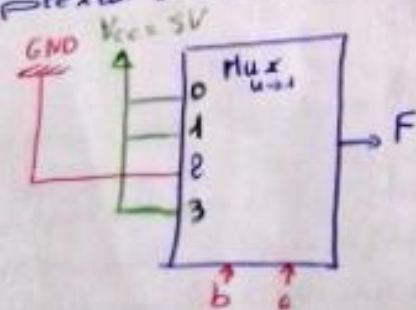
$$d.F_{(abc)} = d \Sigma(0, 1, 3)$$

$F_{(abc)}$ est une f^{on}ction logique à 3 variables d'entrée et par conséquent, elle possède $2^3 = 8$ entrées d'informations. Donc, on doit utiliser un multiplexeur à 8 entrées.

d	c	b	a
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1



Multiplexeur à 2 bits $\Rightarrow 2^2 = 4$ entrées d'informations.
 $F_{(a,b)} = \Sigma(0, 1, 3)$



Exercice N°2 :

Soit la fonction :

$$F(abcd) = (a+b)(c+\bar{d})$$

1. Donner la table de vérité de cette fonction.
2. Réaliser cette fonction à l'aide d'un multiplexeur à 3 bits d'adresses (abc).
3. Réaliser cette fonction à l'aide d'un multiplexeur à 2 bits d'adresses (ab).

Exercice 3

1)

Entrées					Sorties		
E_3	E_2	E_1	E_0		B	A	E_s
0	0	0	0		0	0	1
0	0	0	1		0	0	0
0	0	1	0		0	1	0
0	0	1	1		0	1	0
0	1	0	0		1	0	0
0	1	0	1		1	0	0
0	1	1	0		1	1	0
0	1	1	1		1	1	0
1	0	0	0		1	1	0
1	0	0	1		1	1	0
1	0	1	0		1	1	0
1	0	1	1		1	1	0
1	1	0	0		1	1	0
1	1	0	1		1	1	0
1	1	1	0		1	1	0
1	1	1	1		1	1	0
0	X	X	X		0	0	0

⇒ Codeur prioritaire
4 → 2

$B = E_3 + E_2 \bar{E}_3$

Simplification à l'aide de Karnaugh

$E_3 \backslash E_2$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$B = \bar{E}_0 E_3 + E_0 E_3 + E_2 \bar{E}_3$
 $= E_3 + E_2 \bar{E}_3 \Rightarrow B = E_2 + E_3$

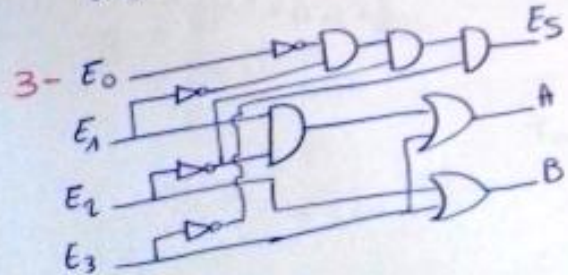
$A = E_1 \bar{E}_3 \bar{E}_1 + E_3$

$A = E_3 + E_1 \bar{E}_1$

$E_s = \bar{E}_0 \bar{E}_1 \bar{E}_2 \bar{E}_3$

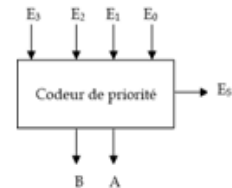
$A = E_3 + E_1 \bar{E}_1$

$E_3 \backslash E_1$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1



Exercice N°3 :

On veut réaliser un codeur de priorité actif à l'état haut à 4 entrées $E_0 \dots E_3$.



On note que la sortie $E_s=1$ si aucune entrées $E_0 \dots E_3$ n'est sélectionnées, dans tous les autres cas $E_s=0$.

- Donner la table de vérité du codeur.
- Donner les expressions logiques des sorties A, B et E_s en fonction des entrées $E_3 \dots E_0$.
- En déduire le circuit logique du codeur.

Exercice 1:

Entrées

Sorties

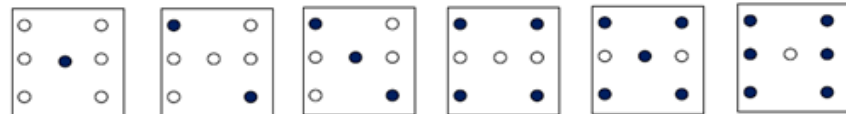
A	B	C	a	b	c	d	e	f	g
0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	1	1	0
0	1	1	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	1	0
1	0	1	0	0	1	1	1	1	1
1	1	0	1	1	1	1	1	1	0



Exercice N°4:

On veut réaliser un Dé électronique.

Les différentes combinaisons d'affichage du Dé électronique sont :



Le circuit logique qui commande les diodes Leds doit comporter 7 sorties (a, b, c, d, e, f et g), (a est le bit de poids le plus fort) et trois entrées A, B et C (A est le bit de poids le plus fort).

1. Dresser la table de vérité.
2. Ecrire les expressions simplifiées des sorties.
3. Réaliser le logigramme des expressions de sorties avec les portes logiques de base.

$$2- a = b = ABC$$

$$c = d = A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} \\ = A\bar{B}(\bar{C} + C) + AB\bar{C} \\ = A\bar{B} + AB\bar{C} \\ = A(B + \bar{C})$$

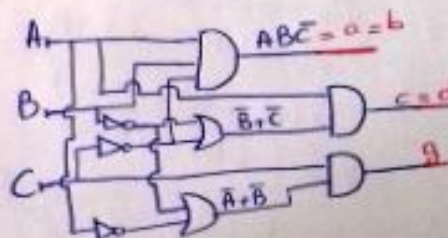
$$c = d = A(B + \bar{C})$$

$$e = f = \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C} + A\bar{B}C + AB\bar{C} \\ = \bar{A}B + A\bar{B} + AB\bar{C} \\ = \bar{A}B + A(\bar{B} + \bar{C})$$

$$e = f = \bar{A}B + A(\bar{B} + \bar{C})$$

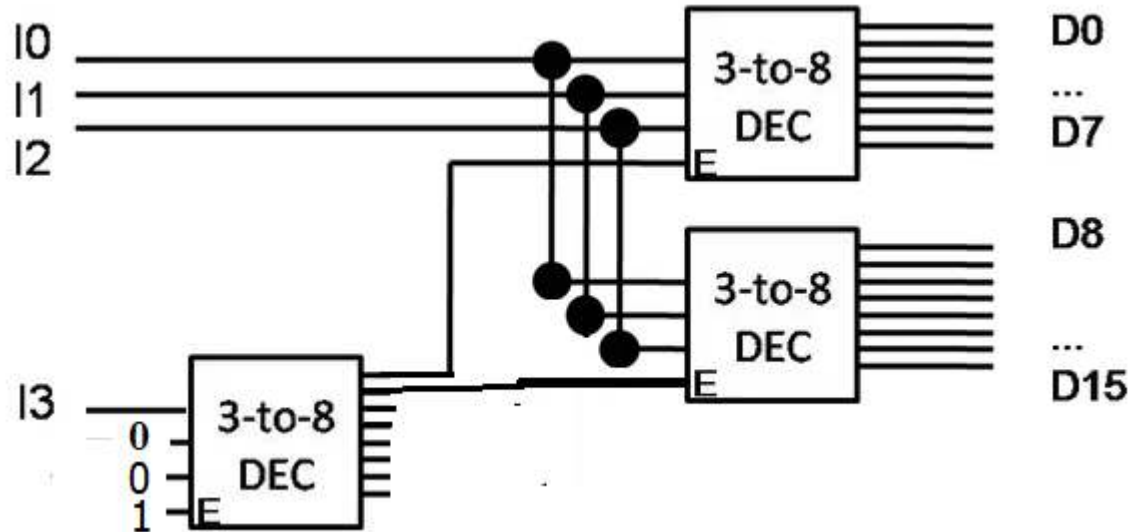
$$g = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C \\ = \bar{B}C(\bar{A} + A) + \bar{A}BC \\ = \bar{B}C + \bar{A}BC \\ = C(\bar{B} + \bar{A})$$

$$g = C(\bar{B} + \bar{A})$$



Correction des exercices de chapitre 4

Conception d'un décodeur 4 vers 16 à partir de 3 vers 8



EX2 : GRAY vers BINAIRE

Code GRAY				Code Binaire			
G ₃	G ₂	G ₁	G ₀	B ₃	B ₂	B ₁	B ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

$$G_3 = \sum_m (8, 9, 10, 11, \tilde{12}, 13, 14, 15)$$

$$G_2 = \sum_m (4, 5, 6, 7, 8, 9, 10, 11)$$

$$G_1 = \sum_m (2, 3, 4, 5, 8, 9, 14, 15)$$

$$G_0 = \sum_m (1, 2, 4, 7, 8, 11, 13, 14)$$

EX2 : GRAY vers BINAIRE

$G_3 G_2 \backslash G_1 G_0$		00	01	11	10
00		0	0	0	0
01		0	0	0	0
11		1	1	1	1
10		1	1	1	1

$$B_3 = G_3$$

$G_3 G_2 \backslash G_1 G_0$		00	01	11	10
00		0	0	0	0
01		1	1	1	1
11		0	0	0	0
10		1	1	1	1

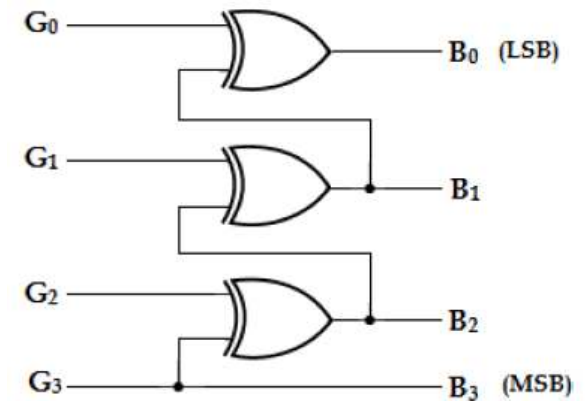
$$B_2 = G_3'G_2 + G_3G_2'$$

$G_3 G_2 \backslash G_1 G_0$		00	01	11	10
00		0	0	1	1
01		1	1	0	0
11		0	0	1	1
10		1	1	0	0

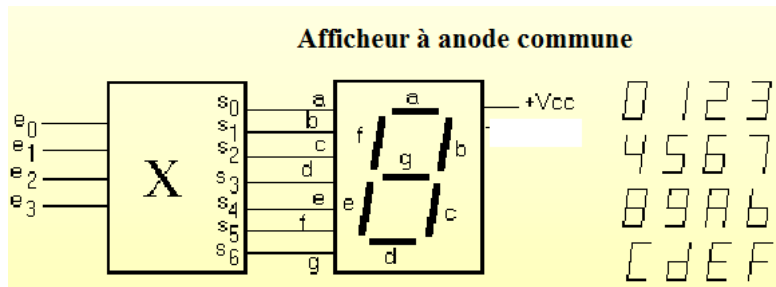
$$B_1 = G_3 \oplus G_2 \oplus G_1$$

$G_3 G_2 \backslash G_1 G_0$		00	01	11	10
00		0	1	0	1
01		1	0	1	0
11		0	1	0	1
10		1	0	1	0

$$B_0 = (G_0 \oplus G_1) \oplus (G_2 \oplus G_3)$$



EX3 : Transcodeur Hexa-7seg



Pour un afficheur à anode commune
il faut un Zéro pour allumer la diode
LED (Segment)

binaire					a	b	c	d	e	f	g
0	0	0	0		0	0	0	0	0	0	1
0	0	0	1		1	0	0	1	1	1	1
0	0	1	0		0	0	1	0	0	1	0
0	0	1	1		0	0	0	0	1	1	0
0	1	0	0		1	0	0	1	1	0	0
0	1	0	1		0	1	0	0	1	0	0
0	1	1	0		0	1	0	0	0	0	0
0	1	1	1		0	0	0	1	1	1	1
1	0	0	0		0	0	0	0	0	0	0
1	0	0	1		0	0	0	0	1	0	0
1	0	1	0		0	0	0	1	0	0	0
1	0	1	1		1	1	0	0	0	0	0
1	1	0	0		0	1	1	0	0	0	1
1	1	0	1		1	0	0	0	0	1	0
1	1	1	0		0	1	1	0	0	0	0
1	1	1	1		0	1	1	1	0	0	0

e1e0

a	00	01	11	10
00	0	1	0	0
01	1	0	0	0
11	0	1	0	0
10	0	0	1	0

$$a = \bar{e}_1 e_0 (\bar{e}_3 \bar{e}_2 + e_3 e_2) + \bar{e}_3 e_2 \bar{e}_1 e_0 + \bar{e}_3 e_2 e_1 e_0$$

e1e0

b	00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	1	0	1	1
10	0	0	1	0

$$b = e_2 \bar{e}_0 (e_1 + e_3) + e_3 e_1 e_0 + \bar{e}_3 e_2 \bar{e}_1 e_0$$

e1e0

c	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	1	0	1	1
10	0	0	0	0

$$c = e_3 e_2 (e_1 + \bar{e}_0) + \bar{e}_3 \bar{e}_2 e_1 \bar{e}_0$$

e1e0

d	00	01	11	10
00	0	1	0	0
01	1	0	1	0
11	0	0	1	0
10	0	0	0	1

$$d = e_2 e_1 e_0 + \bar{e}_3 \bar{e}_2 \bar{e}_1 e_0 + \bar{e}_3 e_2 \bar{e}_1 \bar{e}_0 + e_3 \bar{e}_2 e_1 \bar{e}_0$$

e1e0

e	00	01	11	10
00	0	1	1	0
01	1	1	1	0
11	0	0	0	0
10	0	1	0	0

$$e = \bar{e}_3 (e_0 + e_2 \bar{e}_1) + \bar{e}_2 \bar{e}_1 e_0$$

e1e0

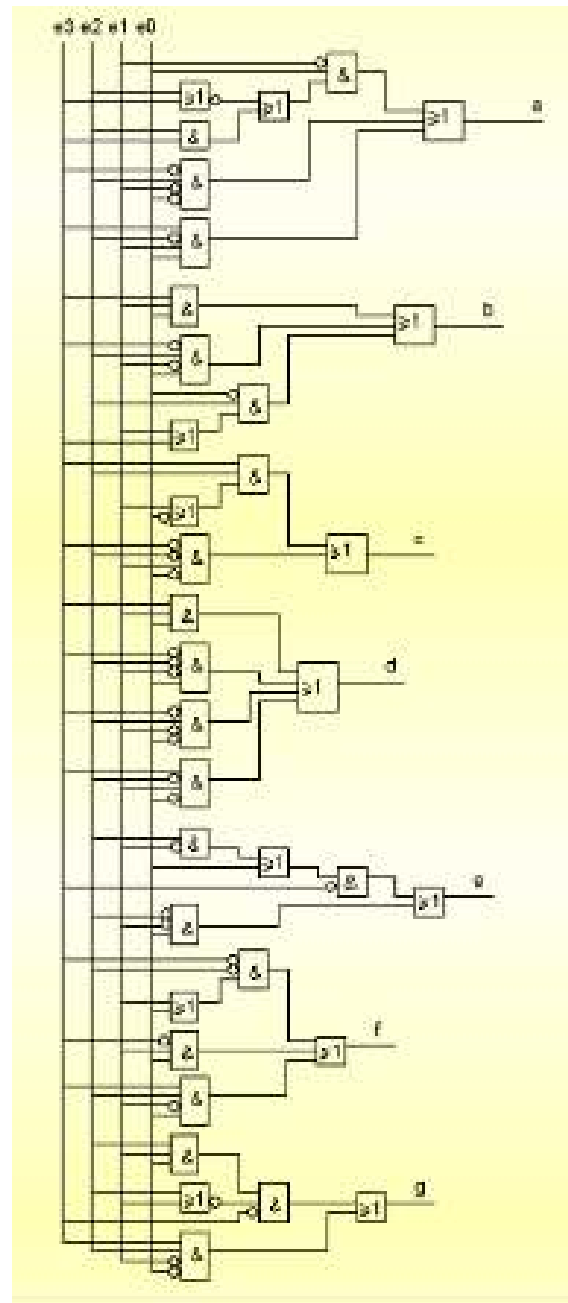
f	00	01	11	10
00	0	1	1	1
01	0	0	1	0
11	0	1	0	0
10	0	0	0	0

$$f = \bar{e}_3 \bar{e}_2 (e_1 + e_0) + \bar{e}_3 e_1 e_0 + e_3 e_2 \bar{e}_1 e_0$$

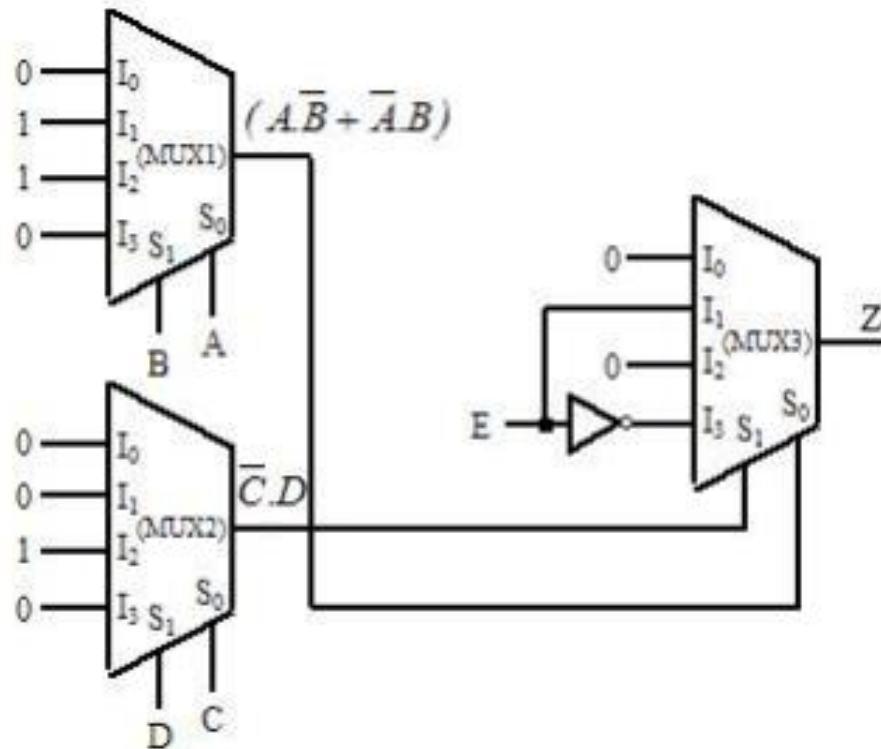
g	00	01	11	10
00	1	1	0	0
01	0	0	1	0
11	1	0	0	0
10	0	0	0	0

$$g = \bar{e}_3 (\bar{e}_2 \bar{e}_1 + e_2 e_1 e_0) + e_3 e_2 \bar{e}_1 \bar{e}_0$$

Logigramme



Ex4 : MUX-4 à partir de 2 MUX-2



- 1) $Z = (A\bar{B} + \bar{A}B) \cdot (E\bar{C}.D + \bar{E}.C.D)$
- 2) $Z = (A \oplus B) \cdot (E \oplus \bar{C}.D)$