

1. Introduction

Dans les systèmes numériques, on utilise souvent des fonctions qui ont justifié leurs réalisations en circuits intégrés. On note en particulier les décodeurs, les multiplexeurs, les démultiplexeurs et les circuits arithmétiques. Bien qu'ils soient plus ou moins remplacés actuellement par les systèmes programmables (circuits logiques programmables, microprocesseur, microcontrôleur), ils sont encore utilisés.

2. Décodeurs

La fonction de décodage consiste à faire correspondre à un code présent en entrée sur m lignes, un autre code en sortie sur n lignes avec en général $m \neq n$.



2.1. Décodeur 1 parmi n

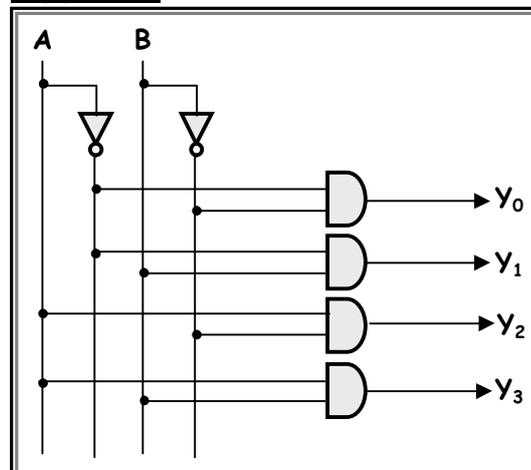
Ce type de décodeur permet de faire correspondre à un code présent en entrée sur m lignes une sortie et une seule active parmi les $n=2^m$ sorties possibles. On le désigne aussi par décodeur m lignes vers n lignes. Pour comprendre le principe d'un tel décodeur, étudions le cas d'un décodeur 1 parmi 4 ou 2 vers 4 donné à la figure ci-dessous. On suppose que le niveau logique actif des sorties est le 1.



Table de vérité					
Entrées		Sorties			
A	B	Y_0	Y_1	Y_2	Y_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

Equations logiques
$Y_0 = \bar{A}\bar{B}$
$Y_1 = \bar{A}B$
$Y_2 = A\bar{B}$
$Y_3 = AB$

Logigramme

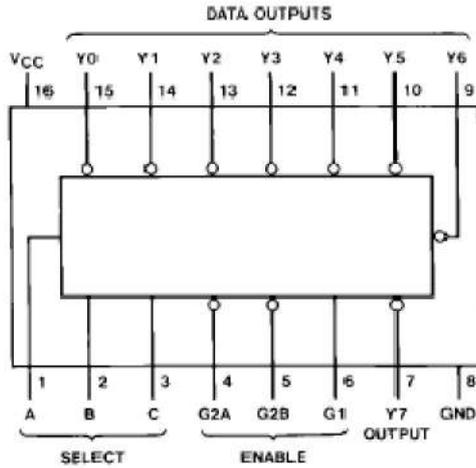


Exemple de circuit intégré : 74LS138

C'est un décodeur 3 vers 8 ou 1 parmi 8. Il permet de sélectionner une sortie parmi 8, de Y_0 à Y_7 , en fonction du mot binaire écrit sur les entrées de sélection A , B et C .

Son diagramme de brochage ainsi que sa table de fonctionnement sont donnés à la page 2. On note que :

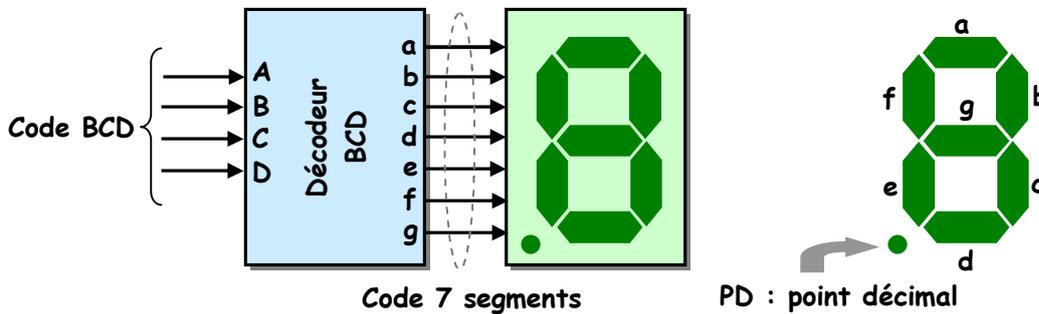
- ☑ Les sorties Y_0 à Y_7 sont actives à l'état logique bas (niveau 0).
- ☑ G_1 , G_{2A} et G_{2B} : Les entrées de validation, c'est-à-dire, que les états de sorties ne sont réellement dépendant du mot de sélection que si :
 - G_1 est au niveau logique haut (niveau 1).
 - G_{2A} et G_{2B} sont toutes les deux au niveau logique bas (niveau 0).
 - $G_2 = G_{2A} + G_{2B}$.



Inputs		Outputs													
Enable		Select													
G1	G2 (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7			
X	H	X	X	X	H	H	H	H	H	H	H	H			
L	X	X	X	X	H	H	H	H	H	H	H	H			
H	L	L	L	L	L	H	H	H	H	H	H	H			
H	L	L	L	H	H	L	H	H	H	H	H	H			
H	L	L	H	L	H	H	L	H	H	H	H	H			
H	L	L	H	H	H	H	H	L	H	H	H	H			
H	L	H	L	L	H	H	H	H	L	H	H	H			
H	L	H	H	L	H	H	H	H	H	L	H	H			
H	L	H	H	H	H	H	H	H	H	H	L	H			
H	L	H	H	H	H	H	H	H	H	H	H	L			

22. Décodeur BCD - 7 segments

Ce décodeur permet de convertir le code BCD, présent à son entrée sous 4 bits, en un code 7 segments disponible à sa sortie. Il est utilisé pour commander un afficheur 7 segments afin d'écrire des chiffres de 0 à 9, certaines lettres et aussi quelques symboles.



221. Afficheur 7 segments

Un afficheur 7 segments est un circuit intégré qui contient 7 diodes électroluminescentes (LED) sous forme de segments et une LED du point décimal. On distingue 2 types d'afficheurs :

↳ Afficheurs à **anode commune**, pour cela :

- ☑ L'anode commune est portée au potentiel +V_{cc}.
- ☑ Pour allumer une LED, on applique un potentiel 0 volt à sa cathode (entrée).

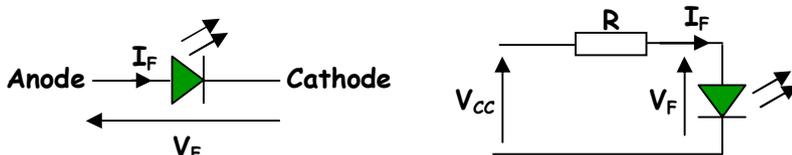
↳ Afficheurs à **cathode commune**, pour cela :

- ☑ La cathode commune est portée au potentiel 0 volt.
- ☑ Pour allumer une LED, on applique un potentiel +V_{cc} à son anode (entrée).

Une LED est caractérisée par :

- ☑ Une tension de seuil V_F.
- ☑ Un courant direct I_F nécessaire à un bon éclairage de la LED.

🔔 **Remarque** : La résistance R permet de limiter le courant dans la LED.



$$R = (V_{cc} - V_F) / I_F$$

Pour : V_{cc} = 5 V, V_F = 2 V et I_F = 20 mA

$$\Rightarrow R = 150 \Omega$$

222. Décodeur BCD

On souhaite concevoir un décodeur BCD pour piloter un afficheur 7 segments à cathode commune.



Table de vérité											
Nombre BCD	Entrées				Sorties						
	D	C	B	A	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

Fonction Traiter

a

CD	00	01	11	10
AB	00	01	11	10
00	1	1	X	0
01	1	X	X	1
11	1	X	X	1
10	0	1	X	1

e

CD	00	01	11	10
AB	00	01	11	10
00	1	1	X	0
01	1	X	X	1
11	0	X	X	0
10	0	0	X	0

b

CD	00	01	11	10
AB	00	01	11	10
00	1	1	X	1
01	1	X	X	0
11	1	X	X	1
10	1	1	X	0

f

CD	00	01	11	10
AB	00	01	11	10
00	1	1	X	1
01	0	X	X	1
11	0	X	X	0
10	0	1	X	1

c

CD	00	01	11	10
AB	00	01	11	10
00	1	1	X	1
01	0	X	X	1
11	1	X	X	1
10	1	1	X	1

g

CD	00	01	11	10
AB	00	01	11	10
00	0	1	X	1
01	1	X	X	1
11	1	X	X	0
10	0	1	X	1

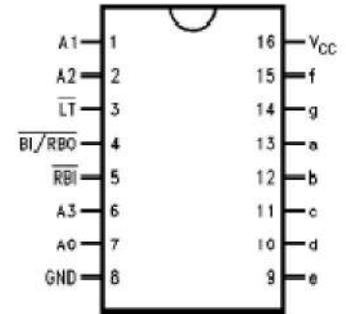
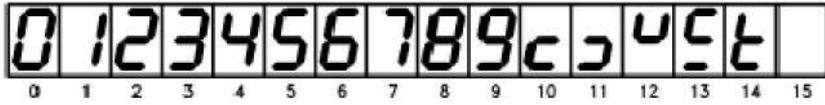
d

CD	00	01	11	10
AB	00	01	11	10
00	1	1	X	0
01	1	X	X	1
11	1	X	X	0
10	0	1	X	1

Equations logiques	
$a = B + D + AC + \bar{A}\bar{C}$	$e = \bar{A}B + \bar{A}\bar{C}$
$b = \bar{C} + AB + \bar{A}\bar{B}$	$f = D + \bar{A}\bar{B} + \bar{B}C + C\bar{A}$
$c = A + C + \bar{B}$	$g = D + \bar{A}\bar{C} + C\bar{B} + B\bar{C}$
$d = D + \bar{A}\bar{B} + B\bar{C} + \bar{C}\bar{A} + A\bar{B}C$	

Exemple de circuit intégré : 74LS48

Un tel décodeur est utilisé pour les afficheurs à cathode commune. Son brochage est donné à la figure ci-contre alors que sa table de vérité est représentée ci-dessous.

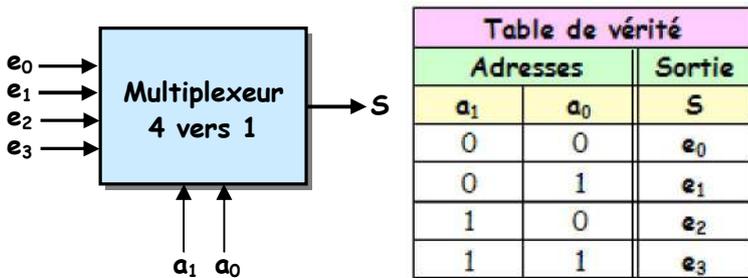


Decimal Or Function	Inputs						Outputs							
	\overline{LT}	\overline{RBI}	A ₃	A ₂	A ₁	A ₀	$\overline{BI/RBO}$	a	b	c	d	e	f	g
0 (Note 1)	H	H	L	L	L	L	H	H	H	H	H	H	H	L
1 (Note 1)	H	X	L	L	L	H	H	L	H	H	L	L	L	L
2	H	X	L	L	H	L	H	H	H	L	H	H	L	H
3	H	X	L	L	H	H	H	H	H	H	H	L	L	H
4	H	X	L	H	L	L	H	L	H	H	L	L	H	H
5	H	X	L	H	L	H	H	H	L	H	H	L	H	H
6	H	X	L	H	H	L	H	L	L	H	H	H	H	H
7	H	X	L	H	H	H	H	H	H	H	L	L	L	L
8	H	X	H	L	L	L	H	H	H	H	H	H	H	H
9	H	X	H	L	L	H	H	H	H	H	L	L	H	H
10	H	X	H	L	H	L	H	L	L	L	H	H	L	H
11	H	X	H	L	H	H	H	L	L	L	H	L	L	H
12	H	X	H	H	L	L	H	L	H	L	L	L	H	H
13	H	X	H	H	L	H	H	H	L	L	H	L	H	H
14	H	X	H	H	H	L	H	L	L	L	H	H	H	H
15	H	X	H	H	H	H	H	L	L	L	L	L	L	L
\overline{BI} (Note 2)	X	X	X	X	X	X	L	L	L	L	L	L	L	L
\overline{RBI} (Note 3)	H	L	L	L	L	L	L	L	L	L	L	L	L	L
\overline{LT} (Note 4)	L	X	X	X	X	X	H	H	H	H	H	H	H	H

Fonction Traiter

3. Multiplexeur

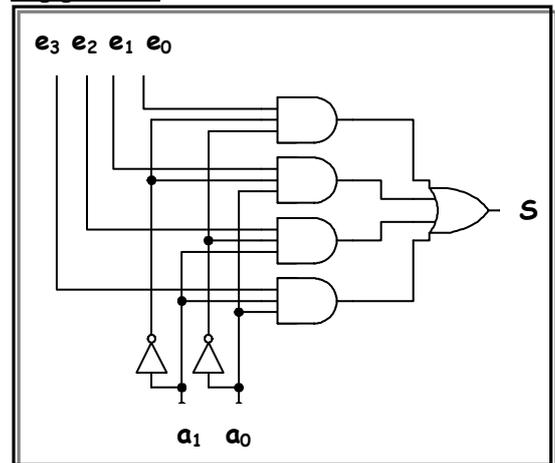
Un multiplexeur permet de sélectionner une entrée parmi 2^n pour transmettre l'information portée par cette ligne à un seul canal de sortie. La sélection de l'entrée se fait alors à l'aide de n lignes d'adressage. Pour comprendre le principe, considérons un multiplexeur à quatre entrées, donc deux lignes d'adressage et une ligne de sortie (Multiplexeur 4 vers 1).



Equation logique

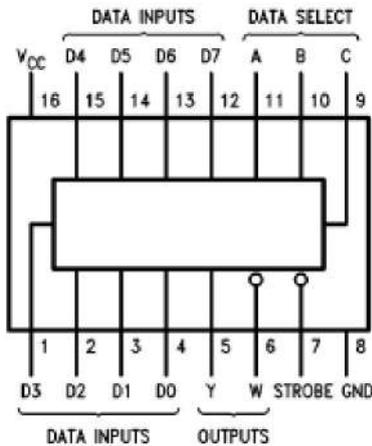
$$S = e_0 \overline{a_1} \overline{a_0} + e_1 \overline{a_1} a_0 + e_2 a_1 \overline{a_0} + e_3 a_1 a_0$$

Logigramme



Exemple de circuit intégré : 74LS151

C'est un multiplexeur 8 vers 1 qui contient une entrée de validation (Strobe ou Enable) permettant de sélectionner le circuit. Son brochage ainsi que sa table de fonctionnement sont donnés ci-dessous.



Inputs			Outputs		
Select			Strobe S	Y	W
C	B	A			
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

4. Démultiplexeur

Le démultiplexeur effectue l'opération inverse d'un multiplexeur, c'est-à-dire qu'il permet d'aiguiller l'information présente à l'entrée unique vers l'une des 2^n sorties. La sélection de la sortie se fait à l'aide de n lignes d'adressage.

Pour comprendre le principe, considérons un démultiplexeur à quatre sorties, donc deux lignes d'adressage et une ligne d'entrée (Démultiplexeur 1 vers 4).

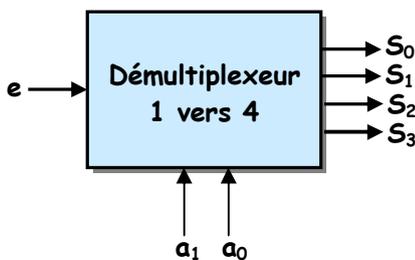
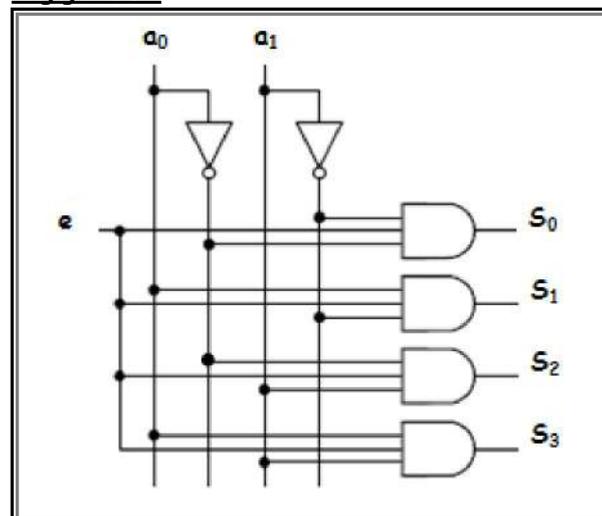


Table de vérité					
Adresses		Sorties			
a_1	a_0	S_0	S_1	S_2	S_3
0	0	e	0	0	0
0	1	0	e	0	0
1	0	0	0	e	0
1	1	0	0	0	e

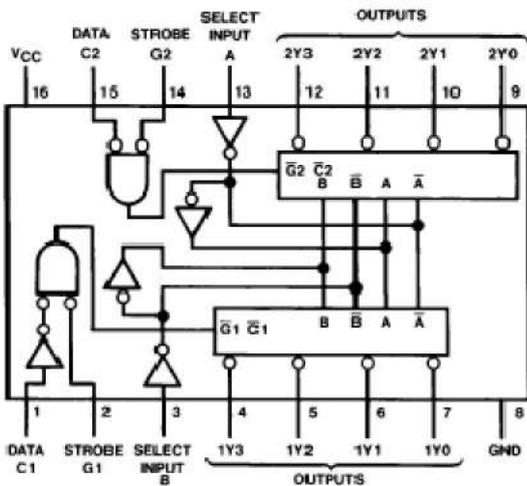
Equations logiques
$S_0 = e \cdot \overline{a_1} \cdot \overline{a_0}$
$S_1 = e \cdot \overline{a_1} \cdot a_0$
$S_2 = e \cdot a_1 \cdot \overline{a_0}$
$S_3 = e \cdot a_1 \cdot a_0$

Logigramme



Exemple de circuit intégré : 74LS155

C'est un double démultiplexeur 1 vers 4 qui contient une entrée de validation (Strobe ou Enable) permettant de sélectionner le circuit. Son brochage ainsi que sa table de fonctionnement sont donnés ci-dessous.



Inputs				Outputs			
Select	Strobe	Data					
B	A	G1	C1	1Y0	1Y1	1Y2	1Y3
X	X	H	X	H	H	H	H
L	L	L	H	L	H	H	H
L	H	L	H	H	L	H	H
H	L	L	H	H	H	L	H
H	H	L	H	H	H	H	L
X	X	X	L	H	H	H	H

Inputs				Outputs			
Select	Strobe	Data					
B	A	G2	C2	2Y0	2Y1	2Y2	2Y3
X	X	H	X	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	X	H	H	H	H	H

5. Comparateur

Le comparateur est un circuit permettant de détecter l'égalité de deux nombres binaires et éventuellement d'indiquer le nombre le plus grand ou le plus petit. Pour comprendre le principe, on va concevoir un comparateur élémentaire de deux mots A et B de 1 bit.



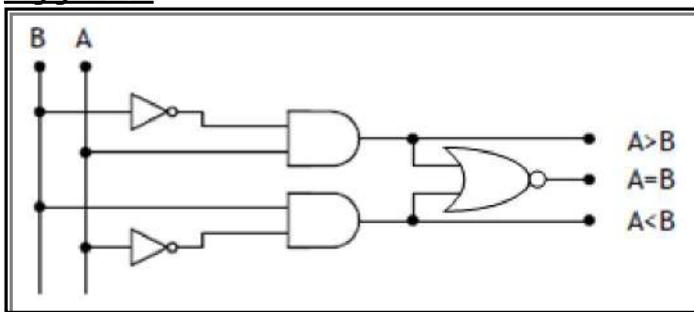
Equations logiques

$(A > B) = A\bar{B}$
$(A = B) = A\oplus\bar{B}$
$(A < B) = \bar{A}B$

Table de vérité

Entrées		Sorties		
B	A	A > B	A = B	A < B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Logigramme



Exemple de circuit intégré : 74LS85

C'est un comparateur qui réalise la comparaison de deux nombres binaires A et B de 4 bits. Son brochage ainsi que sa table de fonctionnement sont donnés à la page 7.

Comparing Inputs				Cascading Inputs			Outputs		
A3, B3	A2, B2	A1, B1	A0, B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

H - High Level, L - Low Level, X - Don't Care

6. Additionneur

61. Demi-additionneur

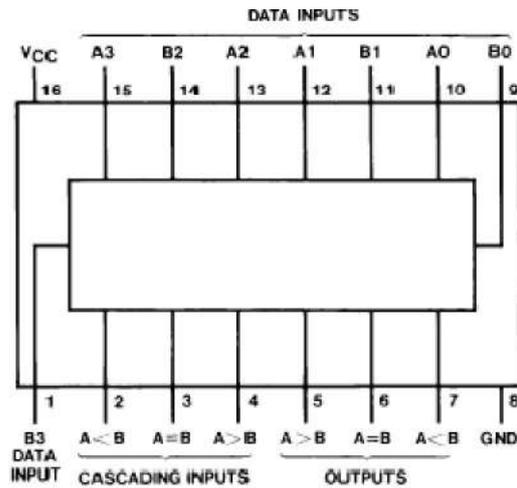
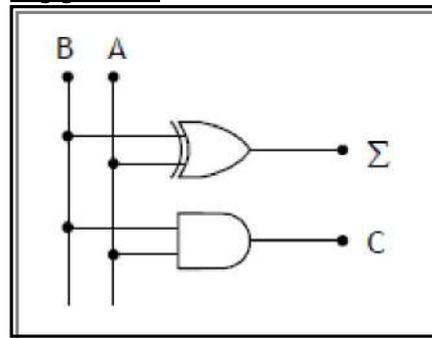
C'est un circuit permettant d'effectuer l'addition de deux nombres binaires A et B de 1 bit, pour générer leur somme Σ et leur retenue C (Carry).



Table de vérité			
Entrées		Sorties	
B	A	Σ	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

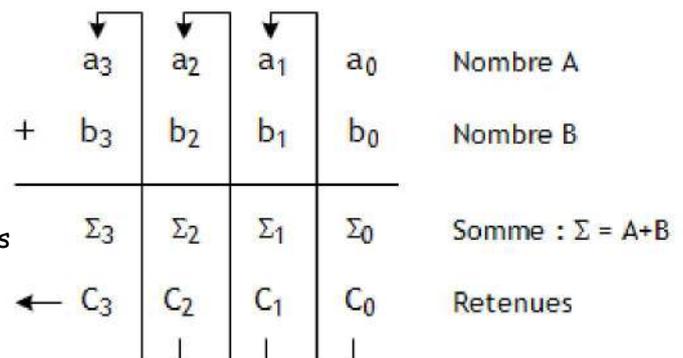
Equations logiques
$\Sigma = A \oplus B$
$C = AB$

Logigramme



62. Additionneur complet

Pour effectuer une addition de deux nombres binaires A et B de n bits, on additionne successivement les bits du même poids en tenant compte de la retenue de l'addition précédente. L'exemple ci-contre illustre le cas de deux nombres binaires A et B de 4 bits.



Il faut donc concevoir une cellule élémentaire appelée additionneur complet et qui permet de réaliser l'addition des bits a_i et b_i en plus de la retenue générée C_{i-1} de l'addition précédente.



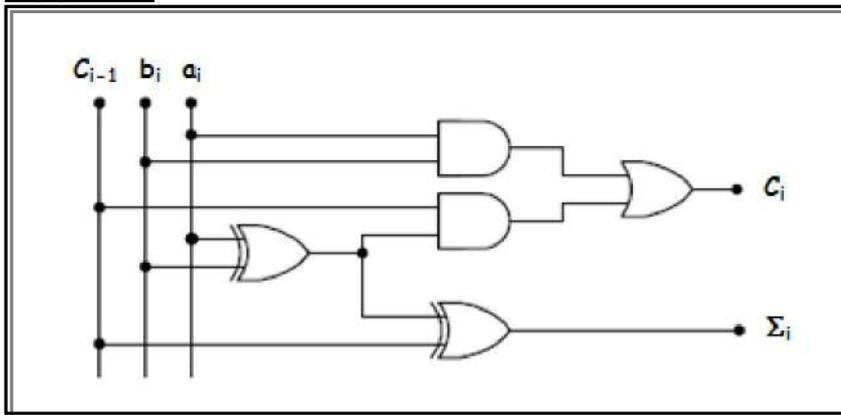
Equations logiques

$$\Sigma_i = a_i \oplus b_i \oplus C_{i-1}$$

$$C_i = a_i b_i + C_{i-1} (a_i \oplus b_i)$$

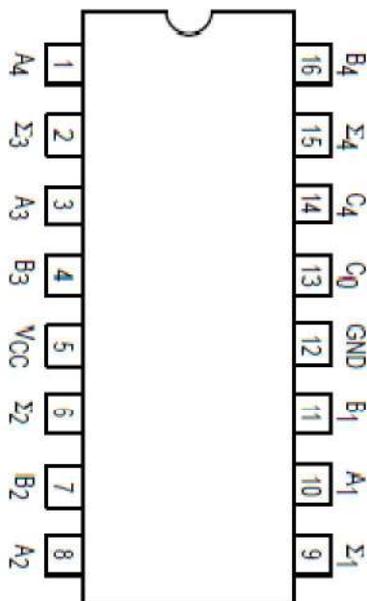
Table de vérité				
Entrées			Sorties	
a_i	b_i	C_{i-1}	Σ_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Logigramme



Exemple de circuit intégré : 74LS83

C'est un additionneur complet qui réalise la somme de deux nombres binaires A et B de 4 bits. Son brochage ainsi que sa table de fonctionnement sont donnés ci-dessous.



$C_{(n-1)}$	A_n	B_n	Σ_n	C_n
L	L	L	L	L
L	L	H	H	L
L	H	L	H	L
L	H	H	L	H
L	L	L	H	L
L	L	H	L	H
L	H	L	L	H
L	H	H	H	H

7. Soustracteur

71. Demi-soustracteur

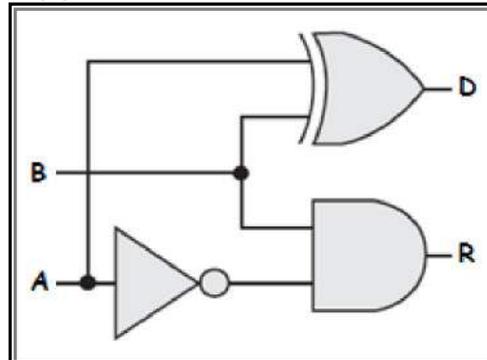
C'est un circuit permettant d'effectuer la soustraction de deux nombres binaires A et B de 1 bit, pour générer leur différence D ($D=A-B$) et leur retenue R.



Table de vérité			
Entrées		Sorties	
A	B	D	R
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Equations logiques
$D = A \oplus B$
$R = \bar{A}B$

Logigramme



72. Soustracteur complet

On peut généraliser la structure du demi-additionneur pour décrire la soustraction des mots de taille supérieure à 1 (n bits). Pour cela il faut introduire une variable supplémentaire qui représente une retenue entrante.

Pour comprendre le principe considérons l'exemple suivant :

Soit à soustraire 43 de 67

$$\Rightarrow (67)_{10} - (43)_{10} = (24)_{10}$$

$$(67)_{10} \Rightarrow (1000011)_2$$

$$(43)_{10} \Rightarrow (101011)_2$$

Ainsi, on obtient :

$$(11000)_2 \Rightarrow (24)_{10}$$

	Pondération							
	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
	1	0	0	0	0	1	1	1 ^{er} terme
-		1	0	1	0	1	1	2 ^{ème} terme
Reports	1	1	1					
	0	0	1	1	0	0	0	Résultat

Cette nouvelle structure s'appelle un soustracteur complet qui réalise la soustraction de deux nombres binaires de 1 bit en tenant compte de la retenue entrante.

L'objectif est donc concevoir une cellule élémentaire (soustracteur complet) qui réalise :

- La différence D_i des bits a_i et b_i en tenant compte de la retenue R_{i-1} de l'étage précédent ($i-1$).
- Soit alors : $D_i = a_i - (b_i + R_{i-1})$.
- Générer la retenue R_i pour l'étage suivant.



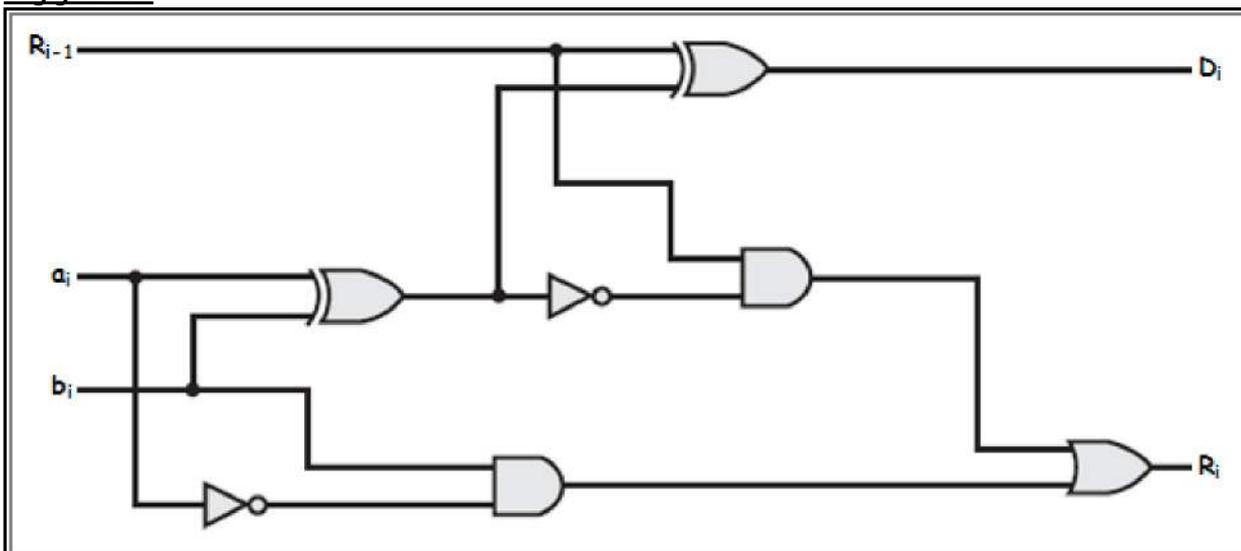
Equations logiques

$$D_i = a_i \oplus b_i \oplus R_{i-1}$$

$$R_i = \bar{a}_i b_i + R_{i-1} (a_i \oplus b_i)$$

Table de vérité				
Entrées			Sorties	
a_i	b_i	R_{i-1}	D_i	R_i
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

Logigramme



Remarques :

- ☑ Cette méthode demeure valable mais difficile à mettre en oeuvre. Pour contourner ces difficultés, on peut exploiter les propriétés du complément à 2 en vue de ramener l'opération soustraction à une simple opération d'addition.
- ☑ La notation en complément à 2 est très utilisée dans les machines numériques du fait qu'avec le même circuit, on peut additionner et soustraire.