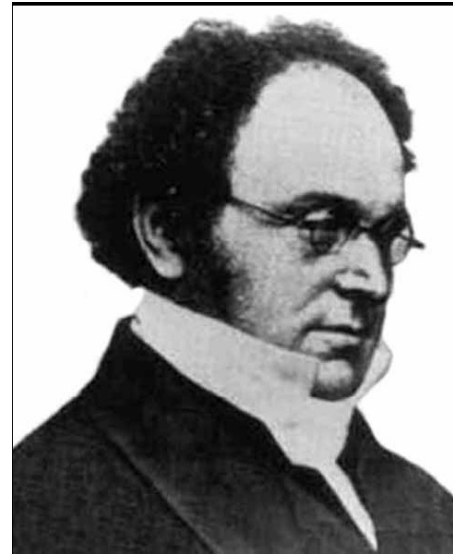


Fonctions combinatoires et circuits associés

Augustus De Morgan 27 juin 1806, Madura, Indes

Le complément de l'intersection d'un nombre quelconque d'ensembles est égal à l'union de leurs compléments.

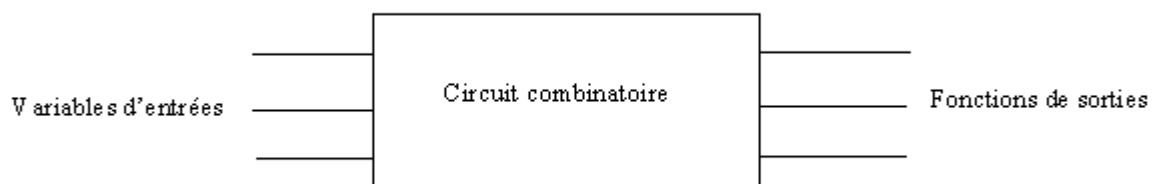
Le complément de l'union d'un nombre quelconque d'ensembles est égal à l'intersection de leurs compléments.



I- Circuit combinatoire

Un circuit est dit combinatoire ses sorties ne dépendent que des combinaisons d'entrées et non pas aussi de ses états antérieurs. A chaque combinaison des variables d'entrées correspond toujours une seule combinaison des fonctions de sortie. Le circuit ne conserve pas en mémoire les états précédents.

Autrement dit, les états des fonctions de sortie sont définis par les combinaisons d'états des variables indépendantes d'entrée.



La réalisation d'un circuit combinatoire consiste à déterminer le système d'équations relatifs aux fonctions de sorties, chaque équation étant une fonction de sortie de toutes ou partie des variables d'entrées.

Les circuits réalisant les fonctions logiques élémentaires (AND, NAND, OR, NOR, XOR, NXOR, NO) sont des fonctions logiques combinatoires.

II- Recherche des équations d'un circuit combinatoire

Pour déterminer les équations d'un circuit combinatoire, il faut réaliser les opérations suivantes :

- Déterminer les différentes variables et les fonctions à calculer
- Déterminer la table de vérité de chaque fonction
- Ecrire les expressions logiques des fonctions de sortie
- Simplification des expressions des fonctions de sortie
- Etablir le logigramme de la fonction simplifié

II-1 Exemple de synthèse d'un circuit utilisant des portes logiques

Concevoir un circuit logique combinatoire ayant 3 entrées et une sortie qui n'est à 1 que lorsqu' au moins deux entrées sont à 1.

La table de vérité de cette fonction est la suivante :

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

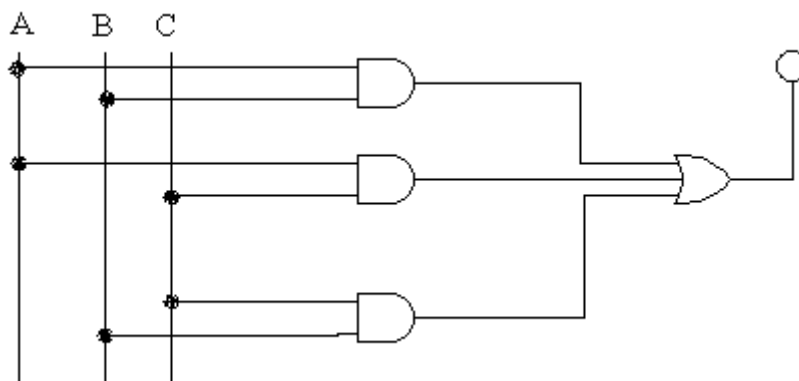
$$S = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

A	BC	00	01	11	10
0				1	
1			1	1	1

Après simplification à l'aide de la table de Karnaugh, on obtient :

$$S = AC + AB + BC$$

Expression qui traduit la fonction cherchée. Le logigramme de cette fonction est le suivant :



On remarque que les portes logiques ne sont pas les seuls éléments possibles pour réaliser le circuit logique. Il existe sous forme de circuit intégré de nombreuses fonctions plus au moins complexes qui conduisent souvent à des solutions simples.

Les circuits logiques que nous allons étudier sont :

- Additionneur
- Soustracteur
- Décodeur
- Codeur
- Transcodeur
- Comparateur
- Multiplexeur
- Démultiplexeur
- Transcodage

III- Additionneur

Les additionneurs sont d'une grande importance non seulement dans les ordinateurs, mais aussi dans un grand nombre de systèmes traitant des données numériques.

III.1- Demi-additionneur

Soient A et B les deux variables d'entrées représentant les bits à additionner, on a par définition de l'addition binaire la table de vérité suivante :

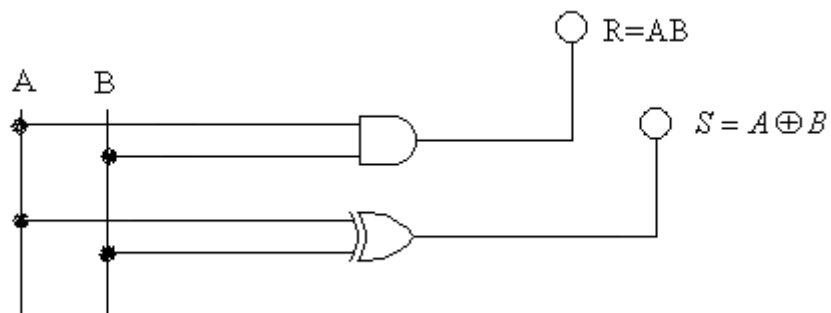
A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Les équations de sorties :

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

$$R = AB$$

Le circuit logique peut être réalisé à partir d'une porte AND et d'une porte OU-EXCLUSIF :



III.2 Additionneur complet (74 LS 83, 74LS 80 1 bit et 74 LS 82 deux bits) :

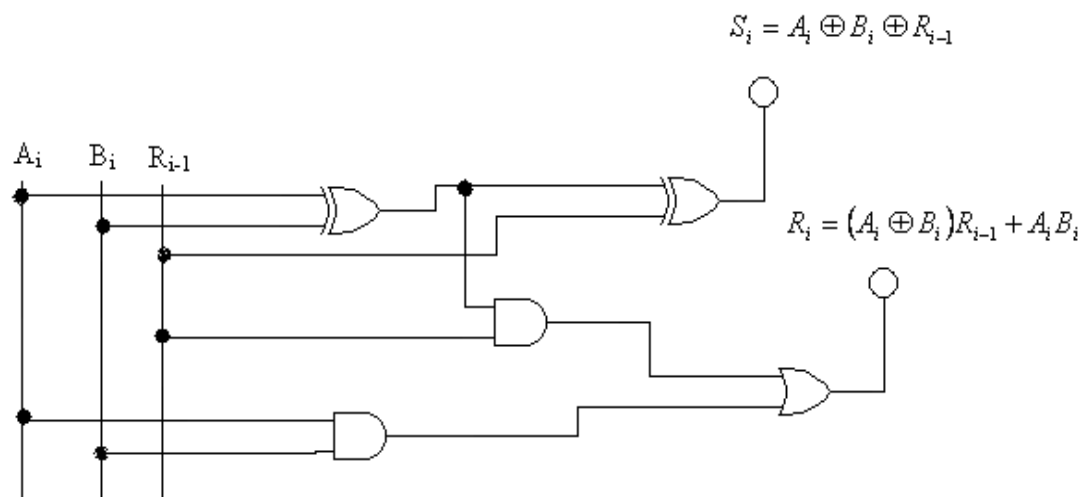
Pour additionner deux nombres binaires, il faut tenir compte de la retenue de l'étage précédent. Il faut donc concevoir un circuit combinatoire à 3 entrées qui sont les entrées A_i et B_i de l'étage i considéré et l'entrée R_{i-1} qui est la retenue de l'étage précédent.

A_i	B_i	R_{i-1}	S_i	R_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

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

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

Le circuit logique peut être réalisé à partir des portes AND et des portes OU-EXCLUSIF :



IV- Soustracteur

Il s'agit de concevoir un circuit combinatoire capable de soustraire un bit d'un autre bit et capable de générer leur différence et leur retenue

IV.1 Demi-soustracteur

Soient A et B les deux variables d'entrées représentant les bits à soustraire, on a par définition de la soustraction binaire la table de vérité suivante :

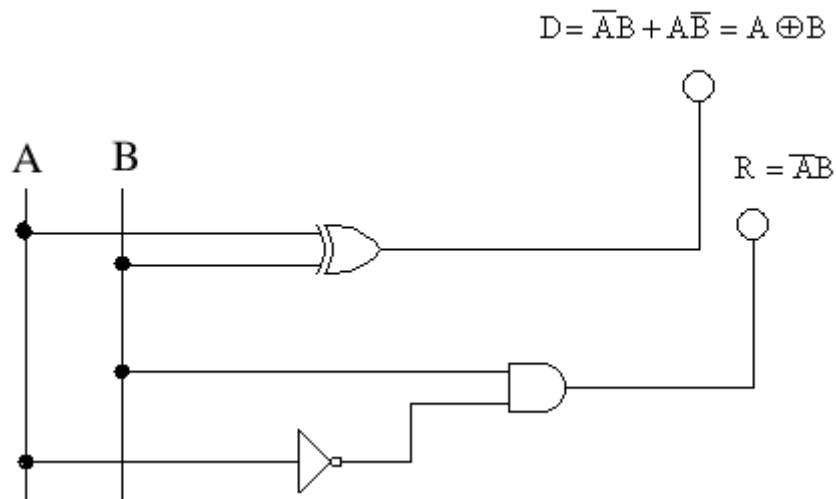
A	B	D	R
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Les équations de sorties :

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

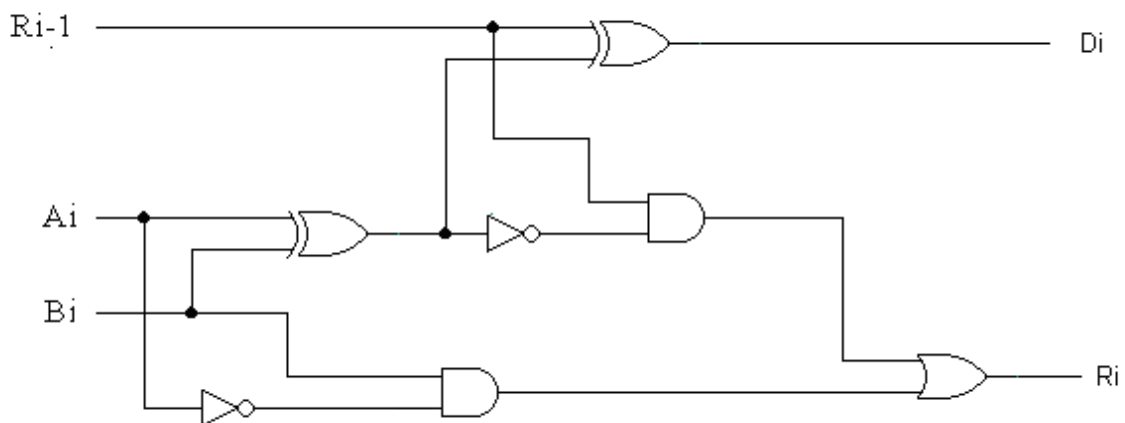
$$R = \bar{A}B$$

Le logigramme est donné ci-dessous :



IV.2 Soustracteur complet

Pour faire la différence de deux nombres binaires, il faut tenir compte de la retenue de l'étage précédent.



V- Les décodeurs

Un décodeur est un circuit à n entrées dites d'adresse et 2^n sorties dont une seule est active à la fois, son rang étant déterminé par la valeur binaire matérialisée par l'état des n entrées.

En plus des n entrées, certains décodeurs possèdent une ou plusieurs entrées de validation. Par exemple, pour une validation active au niveau bas : Si $V=0$ alors le décodage est active et si $V=1$ les sorties sont inhibées. Ces entrées de validation permettent de grouper plusieurs décodeurs afin d'augmenter l'amplitude du décodage.

V.1 Exemple : Décodeur 2 vers 4

Le tableau ci-dessous présente le fonctionnement d'un décodeur binaire 2 vers 4 ou 1 parmi 4 :

Entrées d'adresses		Sorties			
B $2^{(1)}$	A $2^{(0)}$	S3	S2	S1	S0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

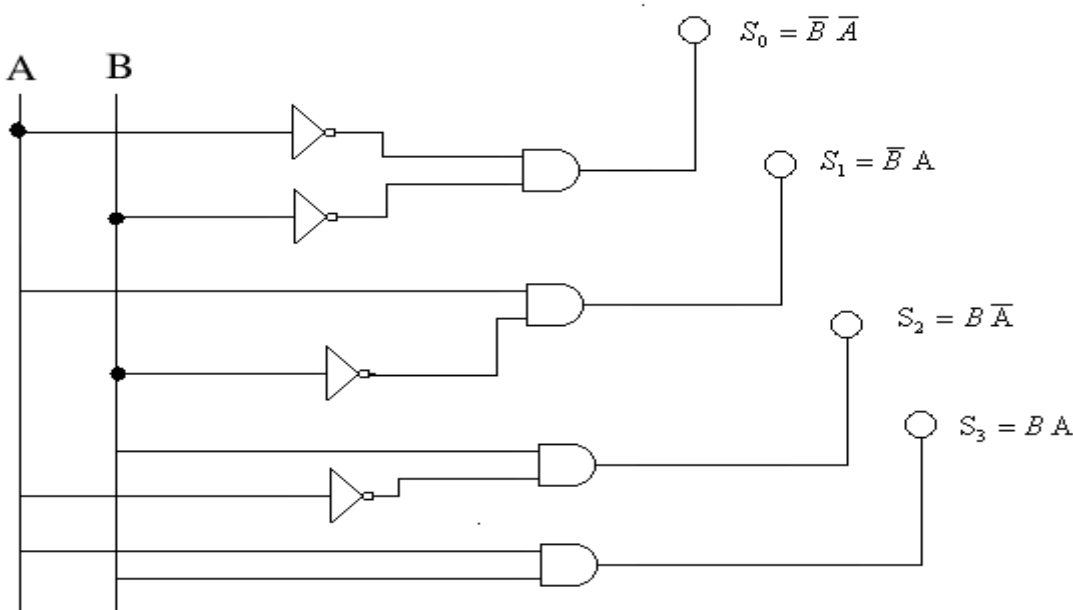
$$S_0 = \bar{B} \bar{A}$$

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

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

$$S_3 = B A$$

On peut réaliser ce décodeur à partir des portes AND et d'inverseurs comme l'indique le schéma ci-dessous :



Les entrées sont appelées adresses car elles expriment en binaire le numéro décimal de la sortie activée.

VI- Quelques réalisations en circuits intégrés

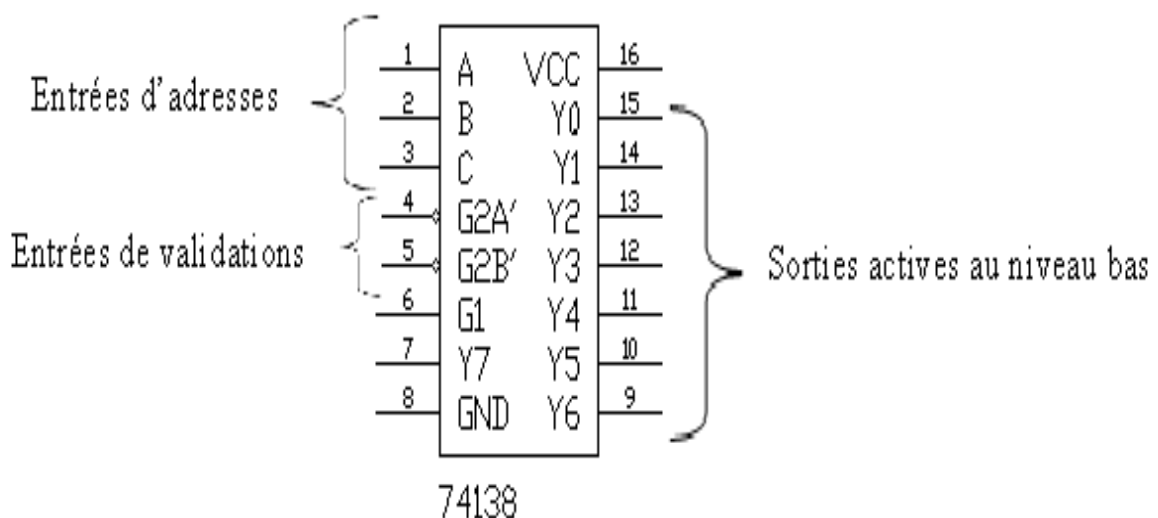
VI.1 Décodeur 74 LS 138

Le décodeur 74 138 est un circuit à 3 entrées d'adresses (décodeur 3 vers 8) ; il possède en outre des entrées validation. Sa table de vérité est indiquée ci-dessous :

$\overline{G1}$	G1	$\overline{G2}$	Select			Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
			C	B	A								
X	X	1	X	X	X	1	1	1	1	1	1	1	1
X	0	X	X	X	X	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	1	1	1	1	1	1	1
0	1	0	0	0	1	1	0	1	1	1	1	1	1
0	1	0	0	1	0	1	1	0	1	1	1	1	1
0	1	0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	1	0	0	1	1	1	1	0	1	1	1
0	1	0	1	0	1	1	1	1	1	1	0	1	1
0	1	0	1	1	0	1	1	1	1	1	1	0	1
0	1	0	1	1	1	1	1	1	1	1	1	1	0
1	1	0	X	X	X	Output corresponding to stored address 0; all others 1							

Les sorties sont actives au niveau bas

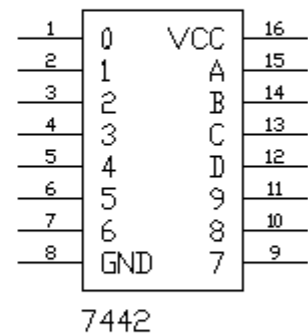
La figure suivante donne le schéma symbolique de ce décodeur :



VI.2 Décodeur DCB- décimal

Le décodeur DCB-Décimal 74LS42, reçoit fréquemment le nom de décodeur entrée 4 voies, sorties 10 voies ou décodeur 1 parmi 10. Le symbole logique et la table de vérité sont indiqués ci-dessous.

No.	BCD Inputs				Decimal Outputs										
	D	C	B	A	0	1	2	3	4	5	6	7	8	9	
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1
2	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1
4	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1
5	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1
6	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1
7	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0
I	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1
N	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
V	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1
A	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
L	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
I	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D					1	1	1	1	1	1	1	1	1	1	1

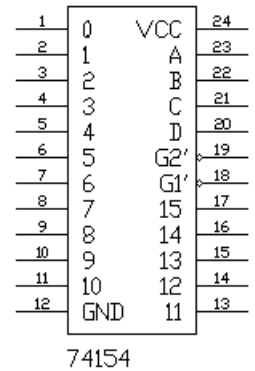


Une sortie ne passe à 0 qu'au moment où son entrée correspondante DCBA est appliquée. Par exemple, la sortie S5 ne devient au niveau bas que lorsque les valeurs sur les entrées sont 0101.

VI.3 Décodeur 74LS 154

Le décodeur 74LS154 est un circuit à 4 entrées d'adresses (décodeur 4 vers 16) ; Ce composant est muni de deux entrées de validations. Il faut appliquer un niveau bas à chaque entrée de validations G1 et G2 du circuit intégré pour que la fonction de validation produise un niveau Haut à sa sortie. Cette sortie est connectée à une entrée de toutes les portes NAND du décodeur. Si les deux entrées de validations ne sont pas mises en marche par des niveaux bas, les 16 sorties du décodeur seront au niveau haut quelque soit les états des quatre variables d'entrées. Le symbole logique et la table de vérité sont indiqués ci-dessous.

Inputs					Outputs																
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	1	x	x	x	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	x	x	x	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	x	x	x	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1



Le circuit 74 155 est également très utilisé, c’est un double décodeur 2 vers 4. Les décodeurs réalisant les mintermes de n variables sont utilisés par exemple dans le pilotage des afficheurs ou dans le multiplexage d’une information.

D’une façon générale les décodeurs réalisent les mintermes de n variables. Ils sont utilisés par exemple dans le pilotage des afficheurs, ou encore dans le démultiplexage des informations.

VII- Les codeurs

Un codeur est un circuit logique combinatoire effectuant la fonction inverse du décodeur. C’est un circuit à 2ⁿ entrées dont une seule est active est qui délivre sur n sorties le numéro de cette entrée.

Soit par exemple un codeur octal-binaire à 8=2³ entrées et produit une représentation de sortie binaire de 3 bits. Le niveau actif des entrées est le 1 logique. Sa table de vérité est indiquée ci-dessous.

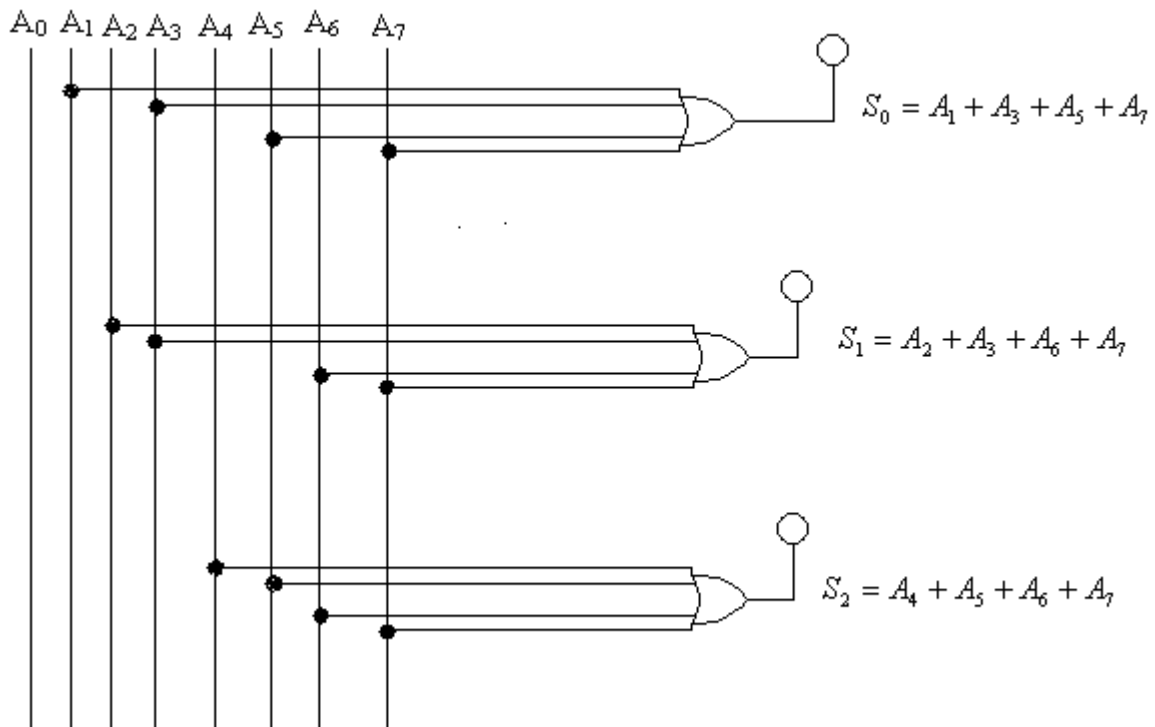
A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	S ₂	S ₁	S ₀
0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$S_0 = A_1 + A_3 + A_5 + A_7$$

$$S_1 = A_2 + A_3 + A_6 + A_7$$

$$S_2 = A_4 + A_5 + A_6 + A_7$$

La figure suivante donne le circuit logique de ce codeur :



On constate qu'un niveau logique 1 sur une seule entrée donne lieu en sortie à un code binaire qui correspond à cette entrée.

On remarque aussi que A_0 n'est pas connectée à aucune porte logique, puisque les sorties de l'encodeur sont normalement à 000 quand aucune des entrées de A_1 à A_7 n'est au niveau 1.

Ce circuit présente l'inconvénient de ne pas faire de différence entre le cas où aucune entrée n'est active et celui où c'est l'entrée 0 qui est sollicitée. Pour résoudre ce problème, on génère un signal supplémentaire de contrôle qui vaut 1 si l'une des 8 entrées passe au niveau haut.

Dans un tel circuit rien n'est prévu en cas où plusieurs entrées seront activées simultanément. Pour éviter toute ambiguïté on fait appel à un encodeur de priorité, les sorties donnent alors, le rang de l'entrée active dont la priorité la plus élevée. Soit par exemple, les deux entrées A_5 et A_6 sont active en même temps, la réponse donnée en sortie est 110 (entrée A_6).

VII.1 Encodeur 74 148

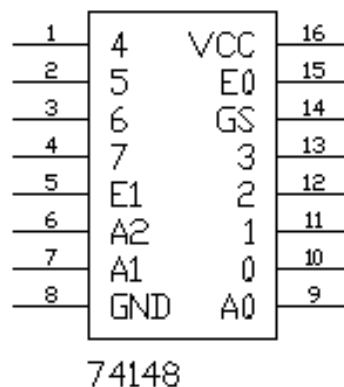
Encodeur de priorité dont les entrées et sorties sont en inverse (entrées actives au niveau bas).

La table de vérité de ce circuit est indiquée ci-dessus :

EI	Inputs							Outputs					
	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
1	X	X	X	X	X	X	X	X	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	X	X	X	X	X	X	X	0	0	0	0	0	1
0	X	X	X	X	X	X	0	1	0	0	1	0	1
0	X	X	X	X	X	0	1	1	0	1	0	0	1
0	X	X	X	X	0	1	1	1	0	1	1	0	1
0	X	X	X	0	1	1	1	1	1	0	0	0	1
0	X	X	0	1	1	1	1	1	1	0	1	0	1
0	X	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

On remarque que l'entrée 7 a la priorité la plus élevée suivie des entrées 6, 5, 4, 3, 2, 1, 0. Les sorties A2, A1, A0 donnent l'adresse sous forme complétée de l'entrée la plus élevée.

La figure suivante donne le schéma symbolique de ce codeur.



- ✚ L'entrée EI (enable Input) à 0 valide le codeur et l'inhibe (toutes les sorties à 1) quand EI=1.
- ✚ La sortie Gs indique la présence d'au moins une information sur une entrée à condition que le codeur ne soit pas inhibé.
- ✚ La sortie E0 (Enable output) permet d'étendre le codeur à plus de 8 entrées par mise en cascade de boîtiers identiques.

Le mot de sortie d'un codeur n'est pas toujours un chiffre binaire. Un codeur de clavier par exemple, délivre la transcription en code ASCII du caractère alphanumérique correspondant à la touche du clavier qui a été enfoncée.

VIII- Les multiplexeurs

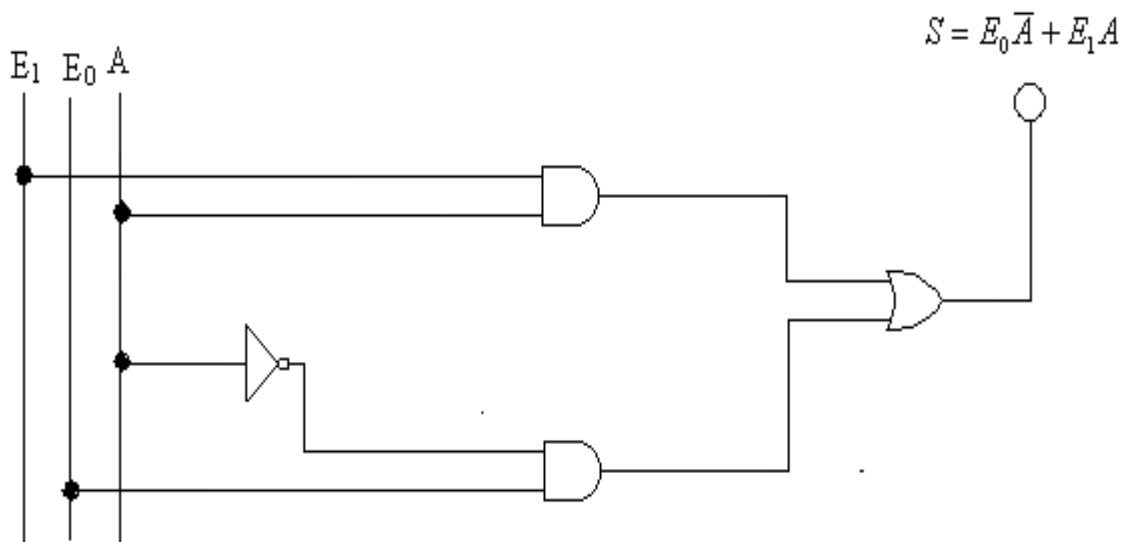
Le multiplexeur (souvent appelé MUX) est sans conteste le circuit usuel le plus utilisé. Le multiplexeur, dans sa plus simple expression possède plusieurs entrées (2^n) d'information, n entrées d'adresse et une sortie. Voici quelques exemples de multiplexeurs :

VIII.1 Multiplexeur à deux entrées

Sa table de vérité est représentée de la façon suivante :

Entrées		Sorties
Informations	Adresses	
E1	E0	E0 E1
	A	
	→ 0	
	→ 1	

Le circuit logique de ce multiplexeur est représenté ci-dessous :



VIII.2 Multiplexeur à deux entrées

Pour aiguiller 4 entrées vers une seule sortie, 2 éléments binaires d'adresse B et A sont nécessaires. Si l'entrée de validation est à 1 on a à la sortie :

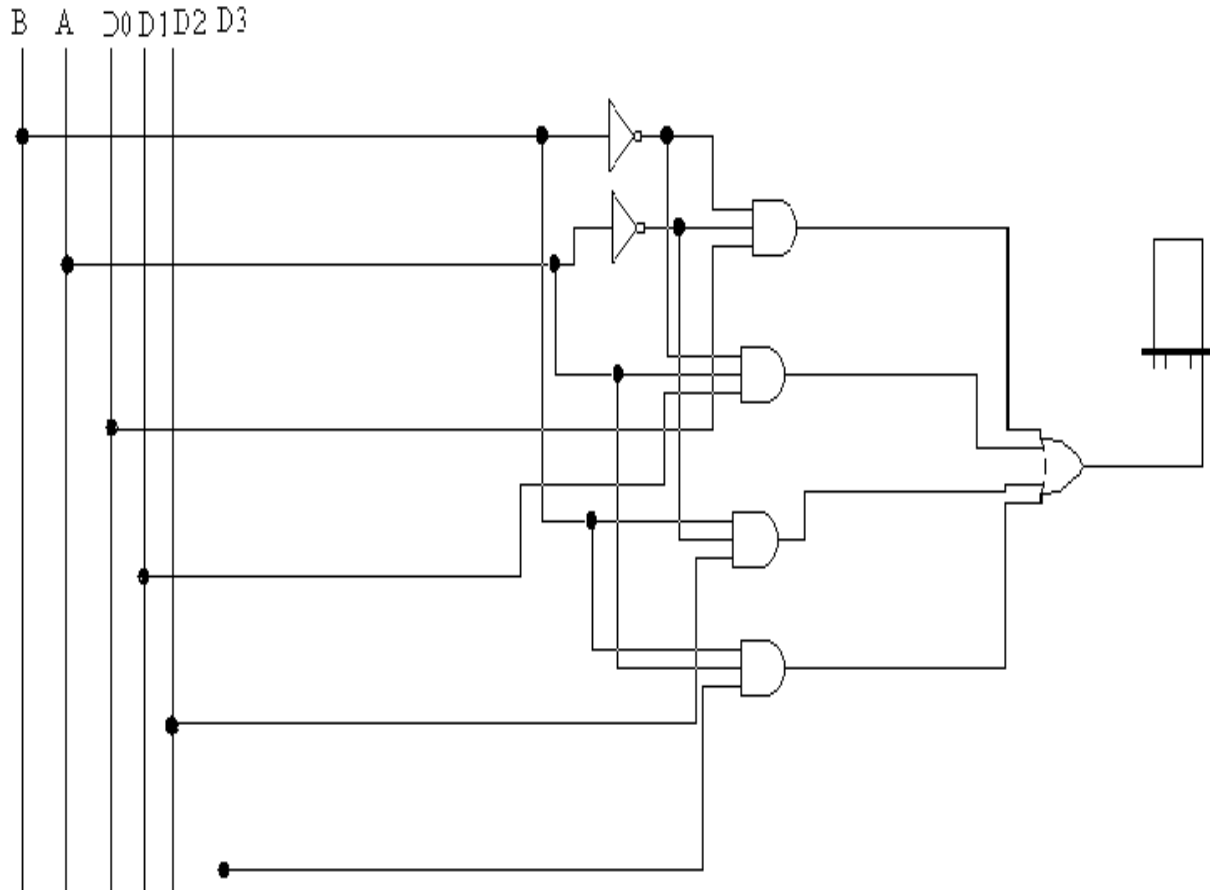
$S_0 = E_0$	entrée N° 0	Si B=A= 0	Adresse 00
$S_1 = E_1$	entrée N° 1	Si B=0 et A=1	Adresse 01
$S_2 = E_2$	entrée N° 2	Si B=1 et A=0	Adresse 10
$S_3 = E_3$	entrée N° 3	Si B=A=1	Adresse 11

Ce qui donne l'expression de S et le schéma du circuit :

$$S = V(\overline{B}\overline{A}D_0 + \overline{B}AD_1 + B\overline{A}D_2 + BAD_3)$$

Si V=0, S=0

Si V=1, multiplexage



✚ Application

Soit une fonction de 4 variables logiques f définie comme :

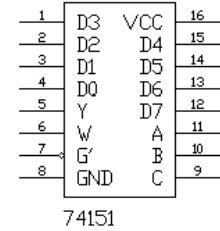
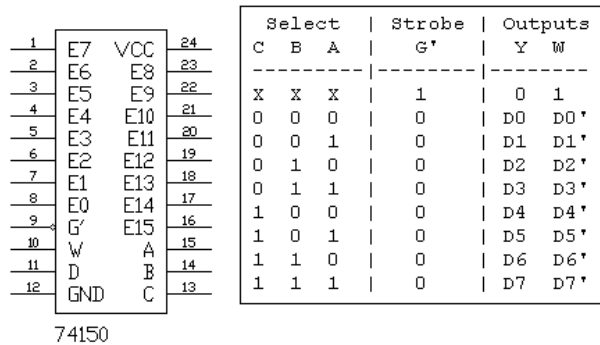
$$f(A, B, C, D) = \overline{A}\overline{C} + ACD + BCD\overline{D}$$

Utiliser un multiplexeur 2 à 1 pour implanter cette fonction.

Parmi les différents types de multiplexeurs, nous distinguons ceux à 16 entrées, 1 sortie (74 150), ceux à 8 entrées, 1 sortie (74151), double multiplexeur à 2 fois 4 entrées de données et 2 sorties (74 153) et quadriple multiplexeur à 4 fois 2 entrées de données et 4 sorties (74157).

Les tables de vérité et les symboles du 74 150 et du 74 151 sont indiquées ci-dessous :

Inputs				Output
D	C	B	A	W
X	X	X	X	H
0	0	0	0	$\overline{E0}$
0	0	0	1	$\overline{E1}$
0	0	1	0	$\overline{E2}$
0	0	1	1	$\overline{E3}$
0	1	0	0	$\overline{E4}$
0	1	0	1	$\overline{E5}$
0	1	1	0	$\overline{E6}$
0	1	1	1	$\overline{E7}$
1	0	0	0	$\overline{E8}$



Le multiplexeur est utilisé pour l'aiguillage des informations, peut encore être utilisé dans la conversion parallèle-série, dans les transmissions de l'information. Il peut être également utilisé comme générateur de fonctions booléennes, et conduire à des schémas beaucoup plus simples que ceux qui utilisent des portes logiques.

IX- Démultiplexeur

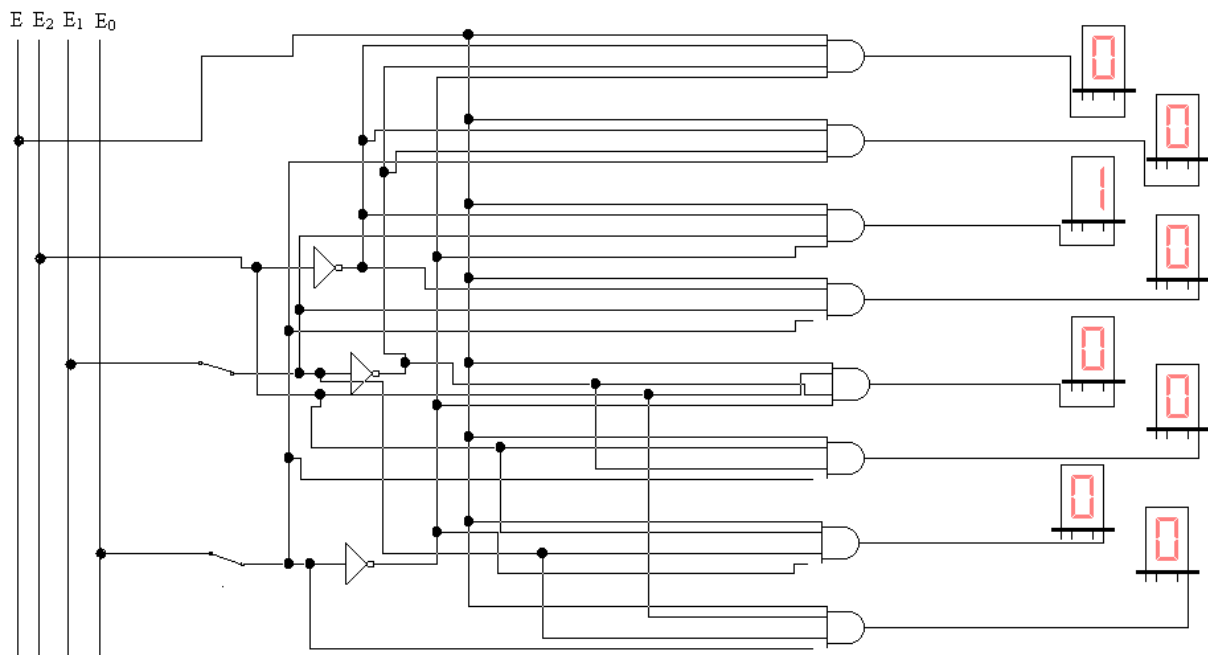
Le démultiplexeur (DEMUX) fonctionne de façon inverse à celle du multiplexeur. Le démultiplexeur reçoit n entrées d'adresses et une entrée à acheminer vers l'une des 2ⁿ sorties possibles. Les autres sorties sont inactives.

IX.1 Démultiplexeur à une entrée et huit sorties

La table de vérité est donnée ci-dessous :

E ₂	E ₁	E ₀	S ₀	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇
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

Le circuit logique :



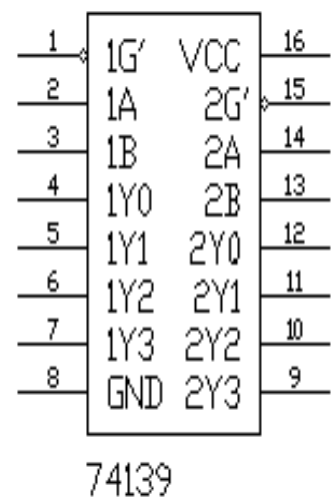
Le fonctionnement du démultiplexeur est relativement simple. Les entrées d'adresses E_i permettent de rédiger le signal d'entrée E vers l'une des sorties S_j de telle sorte que j soit égal à la valeur du mot binaire formé par des entrées d'adresses : $j = (E_2E_1E_0)_2$.

IX.2 Démultiplexeur 74 139

Le démultiplexeur 74 139 est un multiplexeur à deux blocs avec une entrée et quatre sorties par bloc. On parle de DEMUX de 1 vers 4, la sélection des sorties se fait par deux sélecteurs A, B et une broche enable pour valider le bloc :

La table de vérité et le symbole logique sont indiqués ci-dessous :

Enable	\bar{G}	Select		Outputs			
		B	A	Y0	Y1	Y2	Y3
1	X	X		1	1	1	1
0	0	0		0	1	1	1
0	0	1		1	0	1	1
0	1	0		1	1	0	1
0	1	1		1	1	1	0



Remarque : Les décodeurs peuvent être utilisés dans les applications de démultiplexage.

✚ Application du démultiplexeur

Transmission numérique d'un point A vers un point B : le multiplexage et le démultiplexage.

X- Compérateurs de mots

Soit deux nombres binaires A et B de 4 bits ; le comparateur doit fournir un niveau haut sur une sortie de comparaison si A est supérieur à B ou si A égal à B ou si A est inférieur à B.

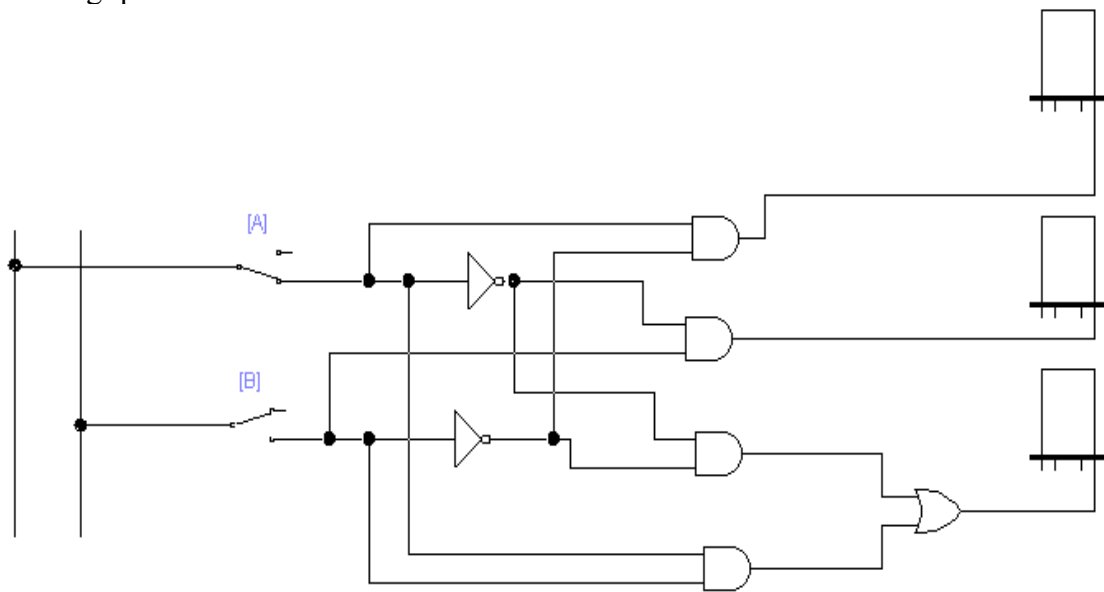
Considérons le cas de deux nombres A et B de 1 bits : A_0 et B_0 .

A_0 et B_0 sont égaux si : $S_3 = \overline{A_0} \overline{B_0} + A_0 B_0 = \overline{A_0 \oplus B_0} = 1$ (c'est la fonction coïncidence).

$A_0 \text{ } \grave{h} \text{ } B_0$: Si $A_0=1$ et $B_0=0$, Soit $S_1 = A_0 \overline{B_0}$

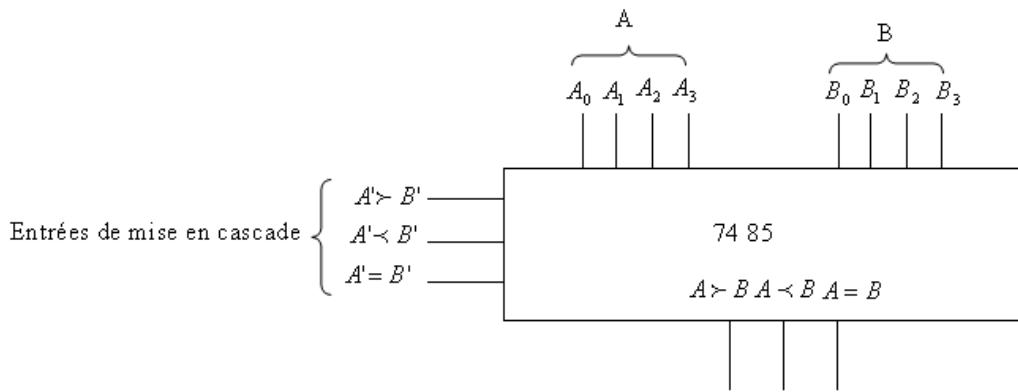
$A_0 \text{ } \pi \text{ } B_0$: Si $A_0=0$ et $B_0=1$, Soit $S_2 = \overline{A_0} B_0$

Le circuit logique :



X.1 Compérateur 74 85

Le circuit 74 85 compare des nombres de 4 bits, en fournissant 3 sorties : $A \text{ } \grave{h} \text{ } B$; $A \text{ } \pi \text{ } B$; $A = B$. Des entrées $A' \text{ } \grave{h} \text{ } B'$; $A' \text{ } \pi \text{ } B'$; $A' = B'$ constituent une façon d'étendre la comparaison à plus de 4 bits.



XI- Convertisseurs de codes

Le passage d'un code à un autre est une opération combinatoire qui peut être effectuée avec les portes logiques AND, OR et NO.

Considérons par exemple le passage du code 2 4 2 1 (Aiken) en code 8 4 2 1 (DCB).

Nombre	M	N	P	Q	D	C	B	A
	2	4	2	1	8	4	2	1
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	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	1	0	1	1	0	1	0	1
6	1	1	0	0	0	1	1	0
7	1	1	0	1	0	1	1	1
8	1	1	1	0	1	0	0	0
9	1	1	1	1	1	0	0	1

Chaque bit de la représentation DCB est une fonction booléenne des 4 bits du code de départ. Par exemple, A est donné en fonction de M N P Q par la table de Karnaugh ci-dessous (les X indiquent les combinaisons qui ne figurent pas dans le code Aiken).

MN \ PQ	00	01	11	10
00	0	1	1	0
01	0	X	X	X
11	0	1	1	0
10	X	X	1	X

Q

Le calcul complet donne :

$$D = NP \quad ; \quad C = \overline{N}P + M\overline{N} \quad ; \quad B = M \oplus P \quad ; \quad A = Q$$

En pratique, les convertisseurs de codes sont souvent réalisés avec des mémoires mortes.

