

Chapitre III : Les Circuits Combinatoires

IV.1 Introduction

Les circuits logiques dans les systèmes digitaux peuvent être combinatoires ou séquentiels. Un circuit combinatoire consiste en des portes logiques dont les valeurs des sorties sont déterminées à partir des valeurs affectées aux entrées indépendamment des valeurs précédentes.

D'une manière plus générale, un circuit combinatoire consiste des variables d'entrée, des portes logiques et des variables de sortie. Les portes logiques reçoivent des signaux provenant des entrées et produisent des signaux pour les sorties. Ce processus transforme une information binaire donnée en entrée à une autre information demandée en sortie. Un circuit combinatoire est schématisé comme suite :



IV.2 Demi-Additionneur

On appelle demi-addition, l'addition simple de 2 nombres quelconques d'un système binaire, avec l'indication du report éventuel.

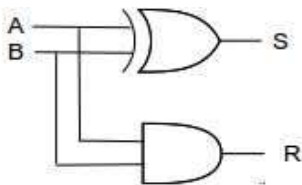
Soit la table de vérité suivant :

<i>A</i>	<i>B</i>	<i>S</i>	<i>R</i>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

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

$$R = A.B$$

S : est appelée somme modulo 2, et *R* est le report (*carry*) son circuit ou symbole graphique :



IV.3 Additionneur complet

L'addition complète est l'opération que consiste à additionner 2 bits A_i et B_i de poids i (on dit aussi de rang i ou ordre i) et un report éventuel R_{i-1} d'ordre $i-1$ résultat de l'addition de rang $i-1$.

Exemple

Soit l'équation suivant : $A_i + B_i + R_{i-1} = S_i$ et R_i

R_{i-1}	A_i	B_i	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

La simplification de S_i et R_i par la table de Karnaugh donne:

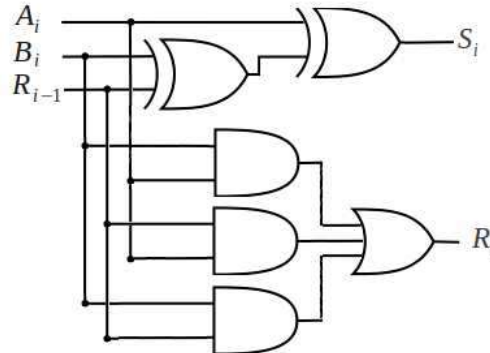
$R_{i-1} \backslash A_i B_i$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$R_{i-1} \backslash A_i B_i$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

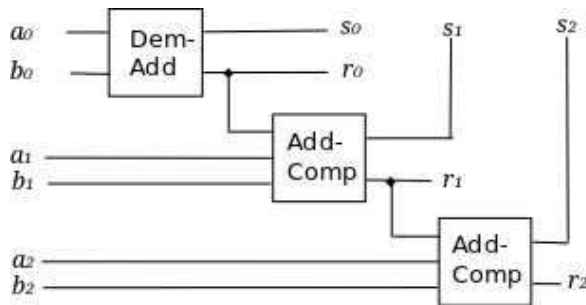
$$\begin{aligned}
 S_i &= R_{i-1} \bar{A}_i \bar{B}_i + \bar{R}_{i-1} \bar{A}_i B_i + R_{i-1} A_i B_i + R_{i-1} A_i \bar{B}_i \\
 &= \\
 &= R_{i-1} (A_i \oplus B_i) + \bar{R}_{i-1} (A_i \oplus B_i) \\
 &= R_{i-1} \oplus (A_i \oplus B_i) \dots (1)
 \end{aligned}$$

$$R_i = A_i B_i + R_{i-1} B_i + R_{i-1} A_i \dots (2)$$

Le schéma du circuit fournissant S_i et R_i est :



Alors le schéma d'une Addition complète de 2 nombres $a_2 a_1 a_0$ et $b_2 b_1 b_0$ est :



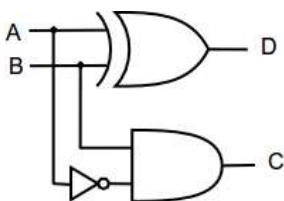
IV. 4 Demi-Soustracteur

La soustraction simple de deux nombres binaires (d'un bit) est représentée dans la TV suivante :

A	B	D	C
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$\begin{aligned}
 D &= A \oplus B \\
 C &= \bar{A} . B
 \end{aligned}$$

Le schéma du Demi-soustracteur :



IV.5 Soustracteur complet

La soustraction est l'opération qui consiste à effectuer la différence entre deux bits A_i et B_i d'ordre i et d'une retenue éventuelle C_{i-1} d'ordre $i-1$

A_i	B_i	C_{i-1}	D_i	C_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

$$D_i = C_{i-1} \oplus (A_i \oplus B_i) \quad (3)$$

$$C_i = \bar{A}_i B_i + C_{i-1} \bar{A}_i + C_{i-1} B_i \quad (4)$$

La simplification de D_i et C_i par la table de Karnaugh donne:

$A_i B_i \backslash C_{i-1}$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

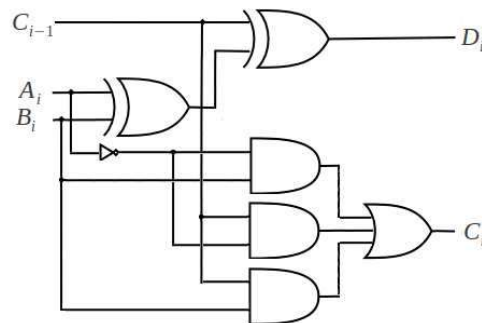
$A_i B_i \backslash C_{i-1}$	00	01	11	10
0	0	1	0	0
1	1	1	1	0

$$D_i = \bar{C}_{i-1} (\bar{A}_i B_i + A_i \bar{B}_i) + C_{i-1} (\bar{A}_i \bar{B}_i + A_i B_i)$$

$$D_i = C_{i-1} \oplus (A_i \oplus B_i)$$

$$C_i = \bar{A}_i B_i + C_{i-1} \bar{A}_i + C_{i-1} B_i$$

Le schéma du circuit fournissant D_i et C_i est :



IV.6 Le Comparateur

Un comparateur est un circuit combinatoire capable de comparer deux nombres A et B pour déterminer leurs positions relatives. Pour comparer deux nombres, nous avons besoin de 3 variables en sortie spécifiant les conditions $A > B$, $A < B$ et $A = B$.

Exemple

Prenons l'exemple d'un circuit qui compare deux nombres de un bit chacun. Soient A_0 et B_0 les deux nombres à comparer. Désignons par A_0 et B_0 les variables en entrée au circuit et par E_0 et C_0 les variables en sortie correspondent respectivement aux conditions $A < B$, $A = B$ et $A > B$. La table suivante est la table de vérité du comparateur :

A_0	B_0	E_0	C_0
0	0	1	0
0	1	0	0
1	0	0	1
1	1	1	0

$$E_0 = \overline{A_0} \overline{B_0} + A_0 B_0 = \overline{A_0 \oplus B_0}$$

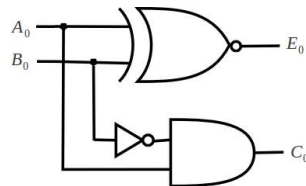
$$C_0 = A_0 \overline{B_0}$$

Si $A_0 = B_0 \Rightarrow E_0 = 1, C_0 = 0$

Si $A > B_0 \Rightarrow E_0 = 0, C_0 = 1$

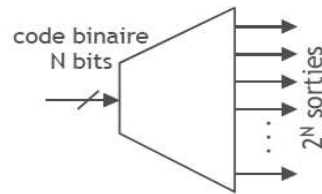
Si $A < B \Rightarrow E_0 = 0, C_0 = 0$

Le logigramme d'un comparateur de deux nombres de 1 bit est montre comme suit :



IV.7 Décodeurs

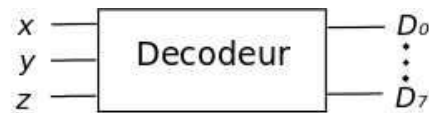
Le décodeur établit une correspondance entre un code d'entrée (exprimé en binaire) et une sortie (des lignes). Si le code d'entrée est $(1101)_2 = (13)_{10}$, c'est la sortie n° 13 qui sera activée.



- En général une seule sortie peut être activée à la fois
- En général le nombre de sorties vaut 2^N
- Certains décodeurs sont conçus pour avoir leur sortie activée au niveau bas (sortie désactivée = niveau haut)

Exemple

Le schéma d'un décodeur à 3 bits (décodeur 1 parmi 8)



x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_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

$$D_0 = \overline{x} \cdot \overline{y} \cdot \overline{z} \quad D_4 = x \cdot \overline{y} \cdot \overline{z}$$

$$D_1 = \overline{x} \cdot \overline{y} \cdot z \quad D_5 = x \cdot \overline{y} \cdot z$$

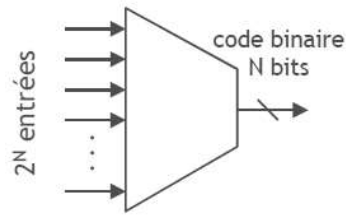
$$D_2 = \overline{x} \cdot y \cdot \overline{z} \quad D_6 = x \cdot y \cdot \overline{z}$$

$$D_3 = \overline{x} \cdot y \cdot z \quad D_7 = x \cdot y \cdot z$$

VI.8 Codeur

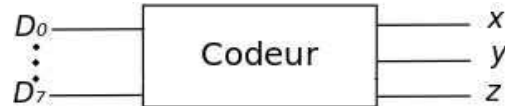
Le codeur (ou encodeur) possède plusieurs entrées, dont une seule est activée à la fois. Il fournit en sortie le code binaire correspondant.

En général, il possède 2^N entrées et N sorties



Exemple

Le schéma d'un codeur binaire-octal est montré sur la figure suivante :



La table de vérité de ce codeur:

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	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

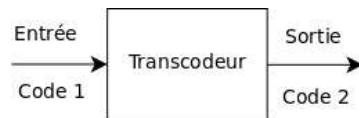
$$x = D_4 + D_5 + D_6 + D_7$$

$$y = D_2 + D_3 + D_6 + D_7$$

$$z = D_1 + D_3 + D_5 + D_7$$

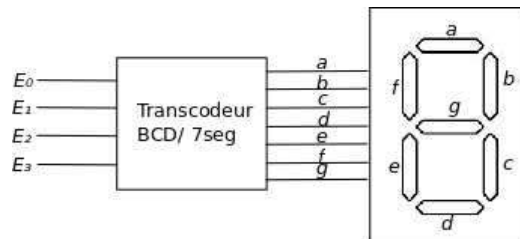
IV.9 Transcodeurs

Un transcodeur est un circuit qui permet de passer d'un code à un autre :



Exemple

Code BCD \rightarrow code affichage chiffre (afficheur 7 segments) :



On peut vérifier que les expressions logiques des sorties sont données par:

$$\begin{aligned}
 a &= E_3 + \bar{E}_0 \cdot \bar{E}_2 + E_1 \cdot \bar{E}_2 + E_0 \cdot E_2 \\
 b &= \bar{E}_2 + \bar{E}_0 \cdot \bar{E}_1 + E_0 \cdot E_1 \\
 c &= \bar{E}_1 + E_0 + E_2 \\
 d &= \bar{E}_0 \cdot \bar{E}_2 + \bar{E}_0 \cdot E_1 + E_1 \cdot \bar{E}_2 + E_0 \cdot \bar{E}_1 \cdot E_2 \\
 e &= \bar{E}_0 \cdot \bar{E}_2 + \bar{E}_0 \cdot E_1 \\
 f &= \bar{E}_0 \cdot \bar{E}_1 + \bar{E}_0 \cdot E_2 + E_3 + \bar{E}_1 \cdot E_2 \\
 g &= E_1 \cdot \bar{E}_2 + \bar{E}_1 \cdot E_2 + E_3 + \bar{E}_0 \cdot E_2
 \end{aligned}$$

IV.10 Multiplexeur

Le multiplexeur est un circuit combinatoire qui permet de sélectionner une ligne d'entrée par une adresse et de faire apparaître à la sortie l'état de cette ligne, c'est à dire un niveau haut ou bas.

Exemple

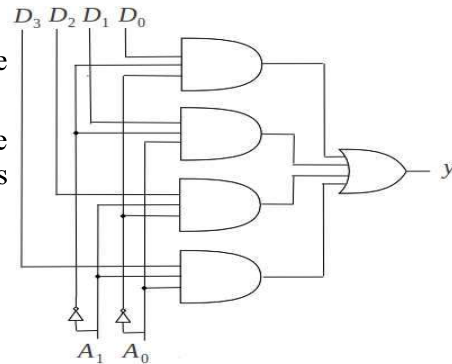
Un multiplexeur de quatre lignes, par exemple, l'adresse aura deux lignes ($2^2=4$) et on veut à la sortie l'état des lignes D_0, D_1, D_2 et D_3 , le chiffre représentant l'adresse affichée par les deux lignes A_1 et A_0 . Soit y la sortie, on veut :

A_1	A_0	y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

D'où l'équation $y = \bar{A}_0 \bar{A}_1 D_0 + \bar{A}_1 A_0 D_1 + A_1 \bar{A}_0 D_2 + A_1 A_0 D_3$ et l'implantation

La sortie y reproduira le niveau logique de la ligne de données sélectionnée par l'adresse $A_1 A_0$.

On appelle ce circuit multiplexeur, car il permet de transmettre sur la ligne y les données apparaissant sur les quatre lignes d'entrée D_0 à D_3 en faisant varier l'adresse.



IV.11 Démultiplexeur

Le démultiplexeur est un circuit combinatoire qui permet de faire apparaître une entrée à une sortie sélectionnée par une adresse, c'est à dire un niveau haut ou bas.

Exemple

Un démultiplexeur de quatre sorties (y_0, y_1, y_2, y_3), par exemple, l'adresse aura deux lignes ($2^2=4$)

A_1	A_0	y_0	y_1	y_2	y_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

$$y_0 = \bar{A}_1 \bar{A}_0 E, \quad y_1 = \bar{A}_1 A_0 E, \quad y_2 = A_1 \bar{A}_0 E, \quad y_3 = A_1 A_0 E$$

IV.12 Pratique

- **Additionneur** : Circuit Intégré : 74 80, 74 82, 74 83.
- **Décodeur** : 74 155 de 2 lignes vers 4 lignes
74 138 de 3 lignes vers 8 lignes
74 154 de 4 lignes vers 16 lignes
- **Transcodeur** : 7442 de BCD à décimal
7446 de BCD vers sept segments
- **Multiplexeur** : 74150 à 16 entrées
74151, 74152 à 8 entrées

Les générateurs et vérificateurs de parité :

La génération et la vérification de parité sont des phases essentielles permettant de s'assurer qu'un transfert de données numériques s'effectue sans erreur au niveau des séquences de bits. Un générateur de parité analyse un flux de bits numériques et ajoute le bit de parité. Ce dernier représente une certaine séquence de données, davantage dans le paragraphe suivant. Dès que le signal atteint la destination, ce bit de parité est vérifié par rapport au flux de données et pour contrôler l'existence d'incohérences. Il s'agit d'une détection d'erreurs.

Un générateur compte une séquence de bits et détermine combien d'entre eux sont égaux à 1. Il existe des bits de parité pairs et impairs. Si une parité paire est utilisée, le générateur ajoute un bit à la séquence qui vérifie que la longueur totale des bits équivaut à un nombre pair de bits égaux à 1. Si une parité impaire est utilisée, une valeur est ajoutée à la fin de la séquence pour s'assurer que l'ensemble du regroupement possède un nombre pair. Par exemple, si un générateur de parité paire recherche des séquences de 7 bits et voit qu'il y a trois 1 et quatre 0, il ajoute un nouveau bit à la fin, c'est-à-dire 1. Ce qui permet au groupe d'avoir quatre bits égaux à 1 et un nombre pair. Le vérificateur analyse mes données au terme d'une réception et détermine si le bit de parité et les bits de données correspondent ou si le nombre pair ou impair est correct. Si tel n'est pas le cas, cela signifie qu'une erreur de transmission est survenue et que les données doivent être ignorées ou renvoyées

Exemple : Un générateur de parité impaire est une fonction qui retourne 1 si le nombre de bits à 1 est impair et 0 sinon.

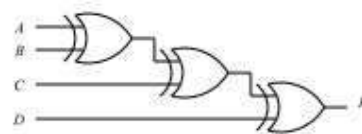
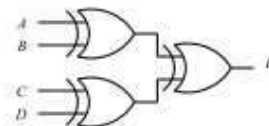
Définir cette fonction pour un mot de 4 bits. Donner un circuit logique implémentant cette fonction.

Correction : La formule pour le générateur de parité impaire sur 4 bits (P) obtenue directement à partir de la table de vérité est :

$$P(A, B, C, D) = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}BCD + A\bar{B}CD + AB\bar{C}D + ABC\bar{D}$$

Ce que donnerait la fonction suivante : $P=A \oplus B \oplus C \oplus D$

A	B	C	D	P
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1



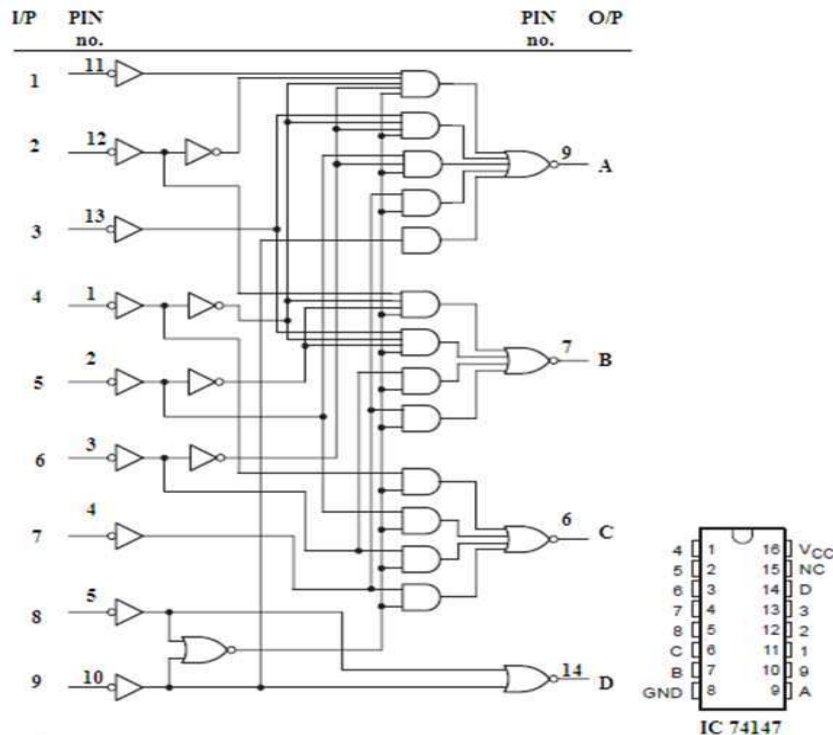
1	1	1	0	1
1	1	1	1	0

Codeur prioritaire:

Le codeur prioritaire est un codeur binaire particulier dont voici les caractéristiques: Si maladroitement plusieurs entrées peuvent être actives en même temps, le codeur fera un choix parmi celles-ci. Il va coder le poids le plus élevé, en effet par exemple si on a appuyé en même temps sur les deux commandes N=1 et N=4, le résultat codé est 101, ce qui ne correspond à aucune de deux combinaisons d'entrée. Un codeur prioritaire donne comme résultat 0100 qui correspond à N=4. Le circuit intégré (74147) est un codeur prioritaire. Ce codeur regroupe à la fois les fonctions de bases qui sont les réunions de commande et les fonctions des conditions de priorités.

Le tableau suivant représente la table de vérité du CI 74147.

INPUT									OUTPUT			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	L	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L



Le logigramme de 74147 (10 lignes vers 4 sorties).

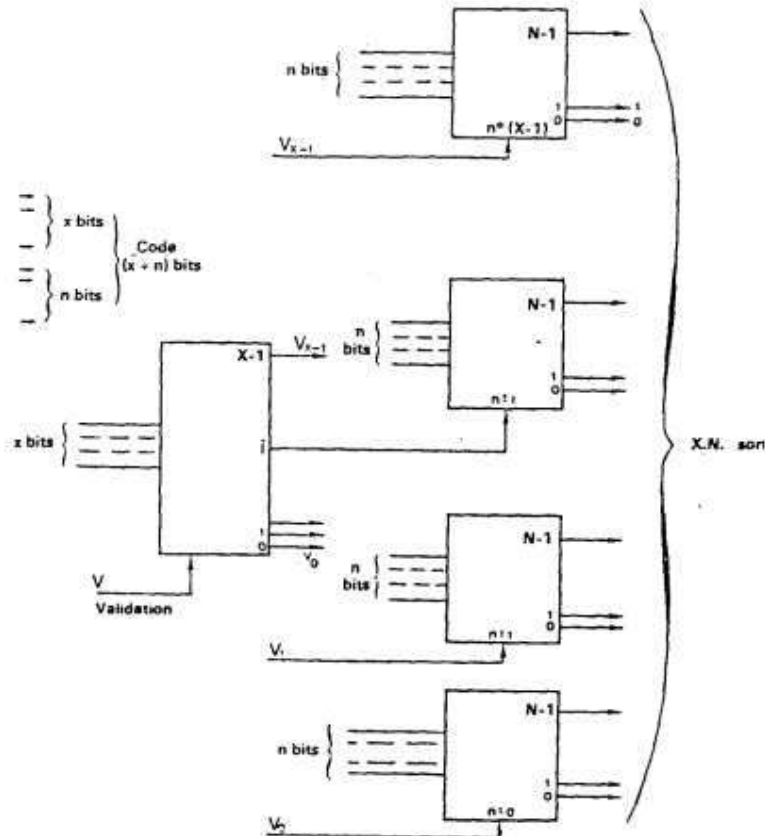
Mise en cascade de décodeurs:

Compte tenu du nombre limité de connections sur un circuit intégré, il est souvent utile de mettre en cascade les décodeurs pour permettre le décodage d'un grand nombre de combinaisons. Une entrée supplémentaire permet cette mise en cascade. Elle est appelée entrée de validation (strobe), et notée V.

N° sortie	Codes							
	x				n			
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
15	0	0	0	0	1	1	1	1
16	0	0	0	1	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
31	0	0	0	1	1	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
255	1	1	1	1	1	1	1	1

Quand le nombre N de sorties est élevée, par exemple 256 (16x16) on a recours à un décodage à 2 niveaux ou plus, suivant les possibilités des modules de base.

Le schéma ci-contre montre l'organisation d'un décodage à 2 niveaux pour le cas général de x.N sorties. En général, on aura $X = 2^x$ et $N = 2^n$. Le code d'entrée est donc constitué de x+n bits. Le décodeur 1 parmi x reçoit les x bits et donne en sortie un signal de sélection qui est utilisé pour valider au deuxième niveau un module à N sorties (il y en a X). Ces modules reçoivent tous les n bits restants et un seul des modules, celui qui est validé, délivre la voie décodée.



Un signal de validation général (V) est appliqué au décodeur du premier niveau :

Si $V = 0$, le décodage est possible.

Si $V = 1$, toutes les sorties du décodeur 1 parmi x sont à 1 et également toutes les sorties des x décodeurs du deuxième niveau.

Le code de chaque sortie est donné par le tableau ci-après ainsi que le schéma fonctionnel.

Exemple de décodage à deux niveaux : réalisation d'un décodeur 5 vers 32 (5 entrées et 32 sorties) en utilisant 4 décodeur 3 vers 8 (modules à 3 entrée et 8 sorties).

