- Circuits logiques combinatoires:

Un circuit logique est un dispositif qui traite et exécute des opérations logiques sur des variables logiques. On rencontre deux types de circuit logiques:

→ Les circuits logiques combinatoires.

→ Les circuits logiques séquentiels.

Un circuit logique est dit combinatoire si ses valeurs de sortie ne dépendent que de ses valeurs d'entrée. Un circuit est séquentiel quand les valeurs de sortie dépendent des valeurs d'entrée appliquées ultérieurement. Donc, il y a le facteur temps qui joue un rôle important, contrairement aux circuit combinatoires où le temps de propagation des signaux n'est pas pris en considération.

1- les portes logiques:

Les portes logiques sont des éléments logiques élémentaires qui peuvent avoir une ou plusieurs entrées et une seule sortie. Les portes logiques de base sont les suivantes:

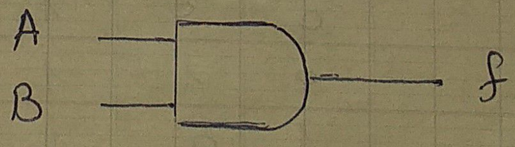
1-1 La porte ET (AND):

$f(A,B) = A.B$; f est vraie si A est vraie et B est vraie.

La Table de vérité de cette fonction est la suivante.

A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

* Symbole:



1.2 La porte OU (OR):

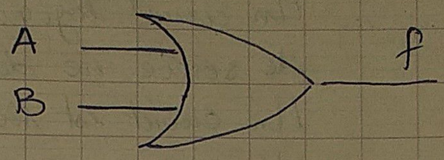
$f(A,B) = A+B$

f est vraie si A est vraie ou B est vraie, ou les deux.

* Table de Vérité:

A	B	f
0	0	0
0	1	1
1	0	1
1	1	1

* Symbole:



1.3 La porte NON ou l'inverseur:

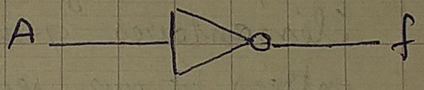
$f(A) = \overline{A}$

f est vraie si A est fausse.

* Table de Vérité:

A	f
0	1
1	0

* Symbole:



1.4 La porte NON-ET (NAND):

$f(A,B) = \overline{A.B}$

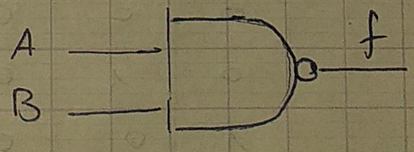
f est vraie si (A et B) sont fausse

f est vraie A est fausse ou B est fausse, ou les deux.

* Table de Vérité:

A	B	f
0	0	1
0	1	1
1	0	1
1	1	0

* symbole:



- 1-5- La porte NON OU (NOR):

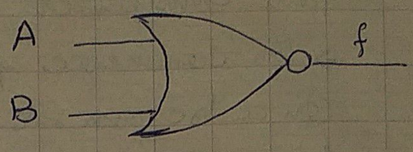
$f(A,B) = \overline{A+B}$

f est vraie si (A et B) sont fausse.

* Table de vérité :

A	B	f
0	0	1
0	1	0
1	0	0
1	1	0

* Symbole :



- 1-6. La porte OU exclusif (XOR):

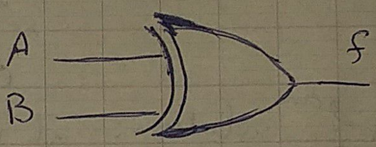
$f(A,B) = A \oplus B = \overline{A}B + A\overline{B}$

f est vraie si A est vraie ou B est vraie.

* Table de vérité

* symbole :

A	B	f
0	0	0
0	1	1
1	0	1
1	1	0



- 1-7. La porte NON OU exclusif (NXOR):

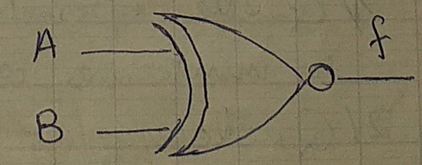
$f(A,B) = A \otimes B = \overline{A \oplus B} = AB + \overline{A}\,\overline{B}$

f est vraie si (A et B) vraie ou fausse.

* Table de vérité :

* symbole :

A	B	f
0	0	1
0	1	0
1	0	0
1	1	1



2- Synthèse d'un circuit combinatoire:

L'objectif de cette étude est de générer le logigramme du circuit à partir de la fonction logique correspondante.

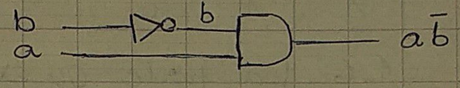
- Exemple:

Soit la fonction : $f(a,b) = a + a\bar{b} + c$

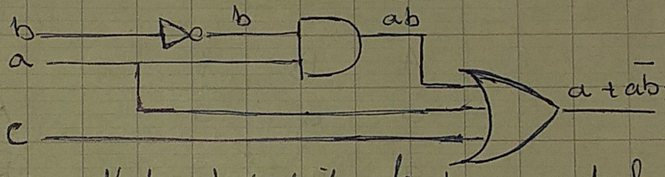
Pour construire le circuit logique correspondant à cette fonction

→ On commence par représenter le terme $a\bar{b}$. Pour cela, on a besoin de deux portes:

La porte NON et la porte ET.

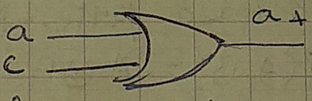


→ Maintenant, nous allons représenter la totalité de notre fonction On a besoin de la porte OU pour la somme de trois termes a, c et $a\bar{b}$:



* 2ème solution: il va d'abord réduit les termes de la fonction c-à-d la simplifier: $f(a,b) = a + a\bar{b} + c = a(1+\bar{b}) + c = a + c$

- Pour représenter la fonction f, on a besoin que d'une seule porte OU



Cette dernière solution est la plus économique.

- Conclusion: Avant d'effectue la synthèse d'un circuit combinatoire, on doit d'abord, répondre aux questions suivantes:

°1/ Quelle représentation de la fonction choisir ? (somme de minterme ou produit de maxterme).

°2/ Quelles portes logiques utiliser pour la représentation schématique (le logigramme).

°3/ Est-ce que la fonction est simplifiée?

3- Les circuits combinatoires particuliers:

Nous allons étudier les circuits combinatoires qui réalisent des fonctions particulières telles que:

→ Le multiplexage.
→ le codage
→ L'addition
→ La comparaison ... etc.

3-1 Le multiplexeur et le demultiplexeur:

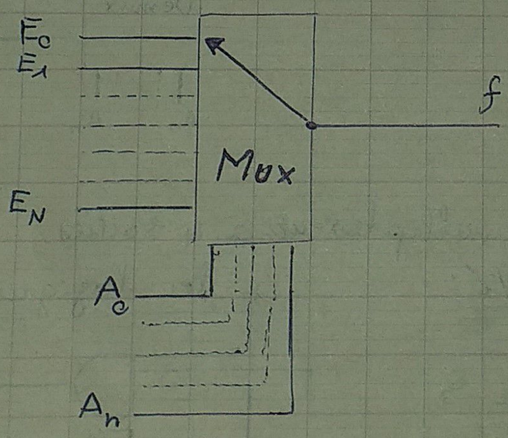
3-1-1 Le multiplexeur (Mux):

C'est un système combinatoire qui réalise la fonction à n variables qui correspondent aux n lignes de sélection:

Un multiplexeur dispose de:

→ 2^n entrées
→ 1 sortie
→ n ligne de sélection

* Symbole:



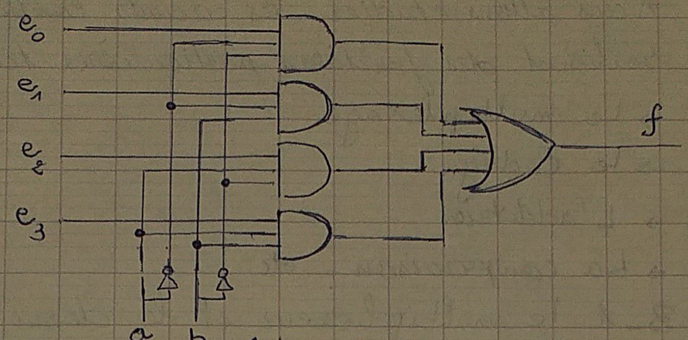
Exemple:

$f(a,b) = e_0 \bar{a}\bar{b} + e_1 \bar{a} b + e_2 a\bar{b} + e_3 ab$

* Table de vérité:

a	b	f
0	0	e_0
0	1	e_1
1	0	e_2
1	1	e_3

* le logigramme:



a et b sont appelées lignes de commande.

3-1-2- le demultiplexeur (Demux):

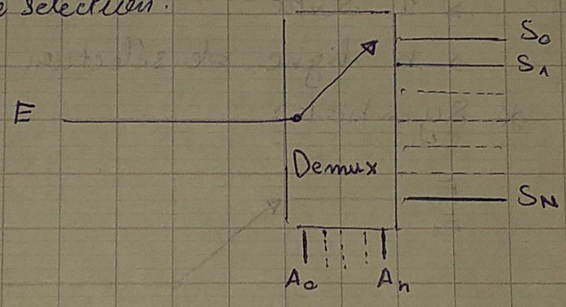
C'est un ~~circuit~~ système combinatoire qui représente une fonction à n variable qui réalise un circuit à:

→ 1 entrée

→ 2^n sorties

→ n lignes de sélection.

* Symbole:



- Exemple:

Réaliser un demultiplexeur à 4 sorties.

* Table de vérité:

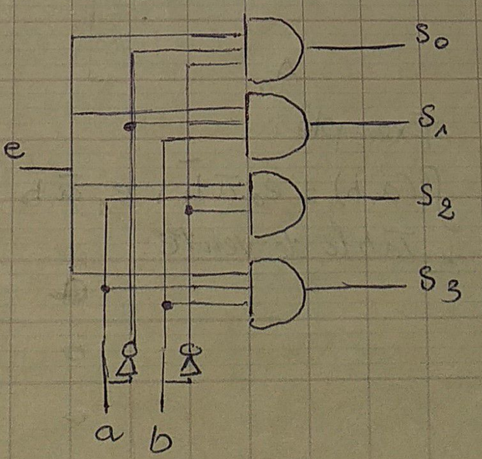
a	b	S_0	S_1	S_2	S_3
0	0	e	0	0	0
0	1	0	e	0	0
1	0	0	0	e	0
1	1	0	0	0	e

où:

$S_0 = \bar{a}\bar{b}e$; $S_2 = a\bar{b}e$

$S_1 = \bar{a}be$; $S_3 = abe$

* le logigramme:

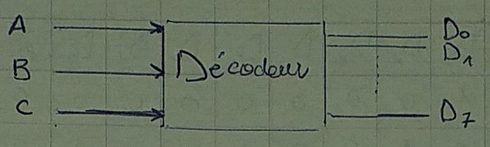


3-2. Décodeurs et codeurs:

3-2-1 Le décodeur:

Un décodeur est un circuit combinatoire qui traduit l'information binaire présente sur n ligne d'entrée et l'utilise pour mettre à l'état 1 l'une seulement de ses 2^n ligne de sortie. Un décodeur est dit de type n vers 2^n.

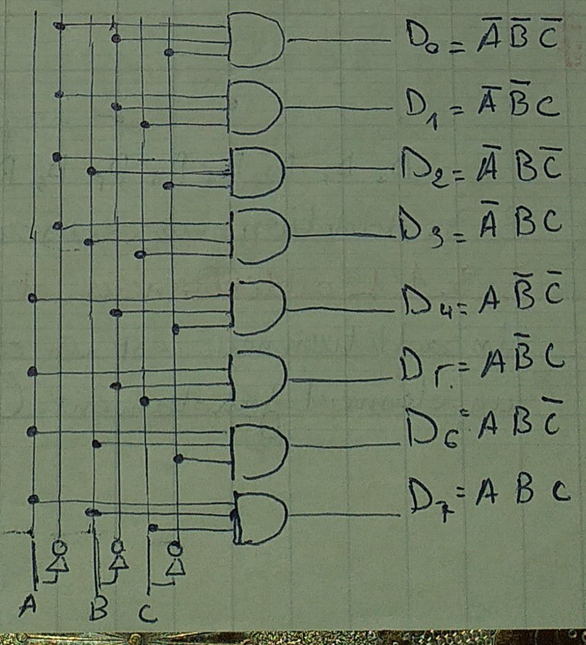
Soit le décodeur 3 entrées et 8 sorties:



* Table de vérité:

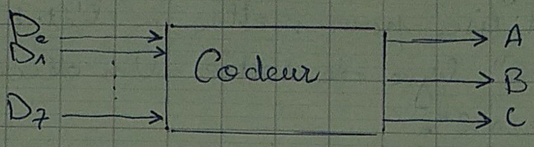
A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

* Logigramme:



3.2.2. Le codeur (ou l'encodeur):

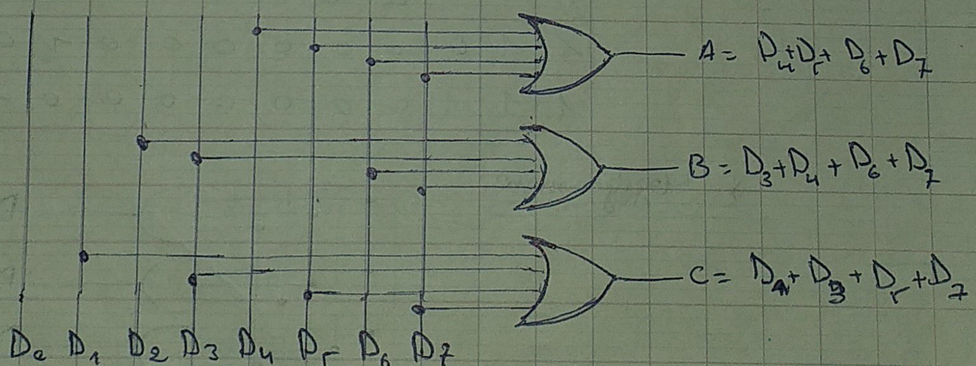
Le codeur réalise la fonction inverse du décodeur, c.à.d à une entrée active (etat 1). parmi les 2^n entrées, il fait correspondre en sortie un code sur n lignes.



* Table de vérité:

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A	B	C
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

* Logigramme:



3.3. Additionneur, soustracteur et comparateur:

3.3.1 L'additionneur et demi additionneur:

Un additionneur est un circuit combinatoire qui est un élément fondamental de toute unité de traitement

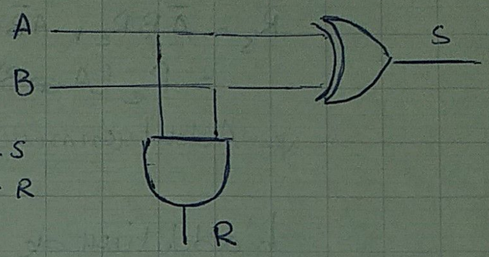
son rôle est d'additionner des bits.
L'addition de deux mots d'un bit nous donne le circuit appelé "demi - additionneur".

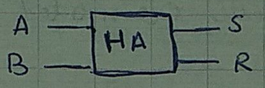
* Table de vérité:

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

S: Somme
R: Retenue

* Logigramme:





$S = A \oplus B = \bar{A}B + A\bar{B}$; $R = AB$

le circuit précédent convient pour effectuer la somme de deux bits de plus faibles poids de deux nombres de n bits, mais il ne convient pas pour réaliser la somme des autres bits, car il ne prend pas en compte l'éventuelle retenue provenant de la somme des bits de poids inférieur.

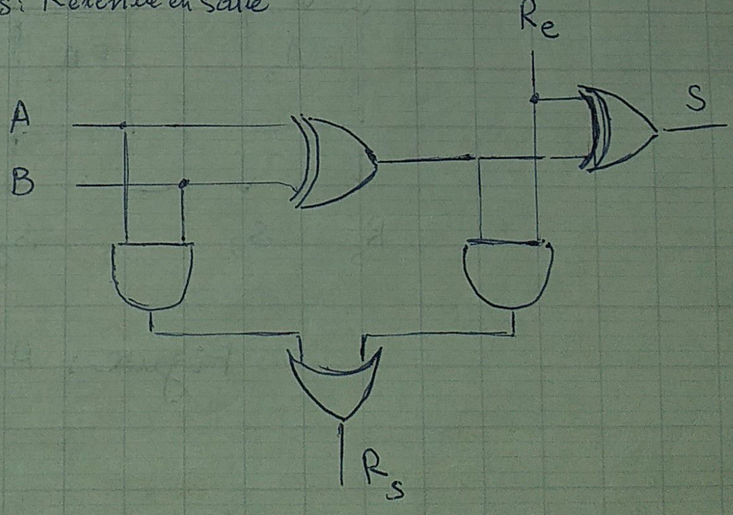
la solution : utiliser un "additionneur complet"

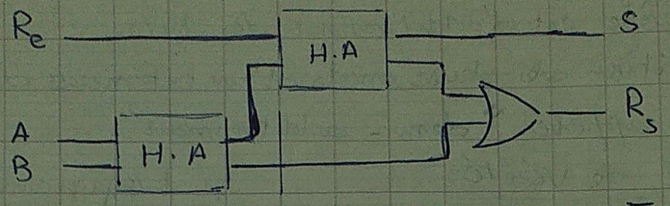
* Table de vérité:

A	B	R_e	S	R_s
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

R_e: Retenue en entrée
R_s: Retenue en satie

* logigramme:





$S = \bar{A}\bar{B}R_e + \bar{A}B\bar{R}_e + A\bar{B}\bar{R}_e + ABR_e = \bar{R}_e(\bar{A}B + A\bar{B}) + R_e(\bar{A}\bar{B} + AB)$

$= \bar{R}_e(A \oplus B) + R_e(\overline{A \oplus B}) = R_e \oplus A \oplus B$

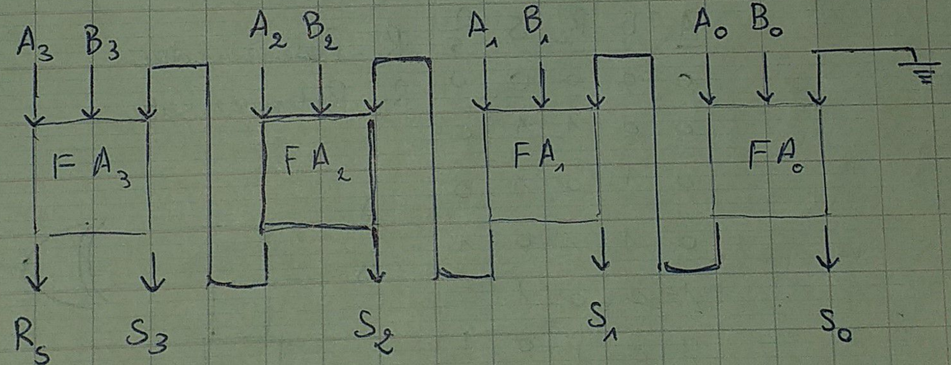
$R_s = \bar{A}BR_e + A\bar{B}R_e + AB\bar{R}_e + ABR_e = R_e(\bar{A}B + A\bar{B}) + AB(R_e + \bar{R}_e)$

$= R_e(A \oplus B) + AB$

* Additionneur complet de n bits (Additionneur en parallèle):

L'addition de nombres comptant plusieurs bits peut se faire en série (bit après bit) ou en parallèle (tous les bits simultanément).
Par exemple:

$$\begin{array}{r} 1\ 1\ 1\ 1 \\ +\ 0\ 1\ 1\ 1 \\ \hline 1\ 0\ 1\ 1\ 0 \end{array}$$



- Figure : Additionneur en parallèle -

3.3.2 Le Soustracteur:

Un soustracteur réalise la soustraction de deux nombres binaires, il suit le même principe que l'additionneur.

*Table de vérité:

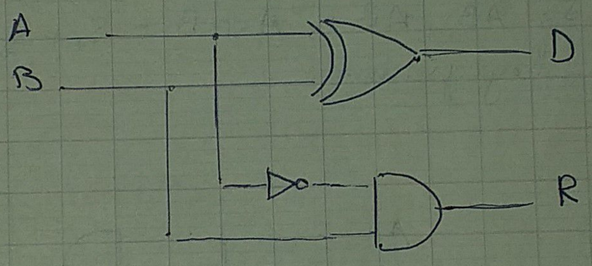
R: Retenue
D: Différence

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

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

$R = \overline{A}B$

* Logigramme:



-Remarque:

Un soustracteur peut se faire en additionnant le premier nombre avec le complement du deuxième nombre, dans ce cas on se sert d'additionneurs pour réaliser la soustracteur.

3.3.3 Le comparateur:

Un comparateur effectue la comparaison de deux mots A et B de 1 bits présents sur 2 ligne d'entrée.

Un circuit comparateur dispose de trois sorties (en général) qui indiquent le résultat de la comparaison

→ Une sortie pour $A = B$

→ Une Sortie pour $A > B$

→ Une sortie pour $A < B$

* Table de vérité:

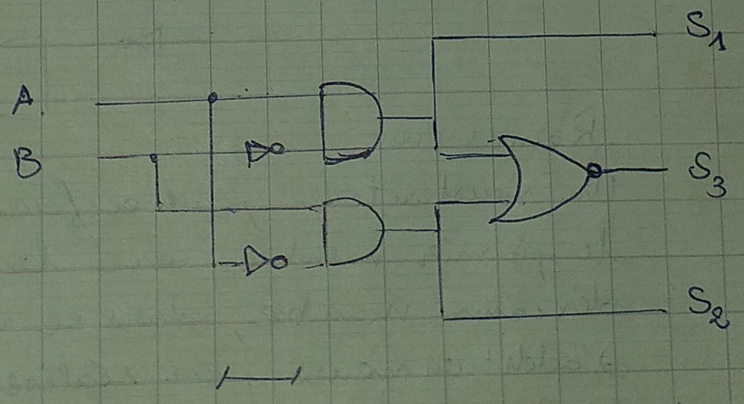
A	B	$S_1 (A>B)$	$S_2 (A<B)$	$S_3 (A=B)$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

$S_1 = A\overline{B}$

$S_2 = \overline{A}B$

$S_3 = \overline{A}\,\overline{B} + AB = \overline{A \oplus B} = \overline{S_1 + S_2}$

* Logigramme:



Mini projet:

1/ Afficheur +