

ARCHITECTURE DES ORDINATEURS

Compte rendu du TP n 1 : Circuits combinatoires



Fait par :

HASSANI Fadoua
HESSANE Mohamed Saâd

Encadrant :

KARIM Abdel Ali

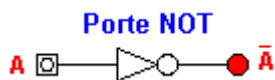
I. UTILISATION DES PORTES LOGIQUES STANDARD

Lors du cours d'architecture des ordinateurs, nous nous sommes proposé d'étudier les fonctions de bases de l'algèbre de Boole.

Pour ce, il a fallu établir la table de vérité pour chacune des fonctions, lui donner un nom et un symbole.

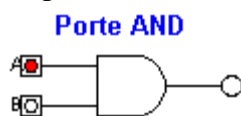
Voici quelque fonction de bases (appelé aussi porte logique) que nous avons pu voir en classe:

1. La porte NOT :



A	\bar{A}
0	1
1	0

2. La porte AND :



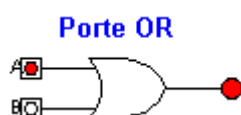
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

3. La porte NAND :



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

4. La porte OR :



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

PS : Vous trouverez les circuits dans le fichier XXX.dwm exécutable avec Digital Works.

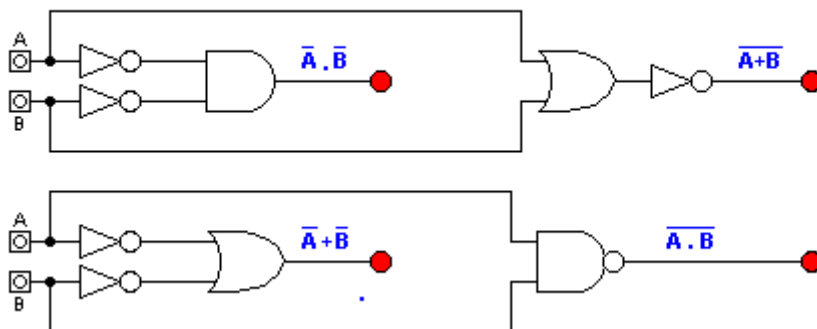
Loi de De Morgan

Les lois de De Morgan se résument en deux relations connues :

- $\overline{A \cdot B \cdot C \dots} = \bar{A} + \bar{B} + \bar{C} \dots$
- $\overline{A + B + C \dots} = \bar{A} \cdot \bar{B} \cdot \bar{C} \dots$

Nous pouvons représenter ces équations par deux circuits :

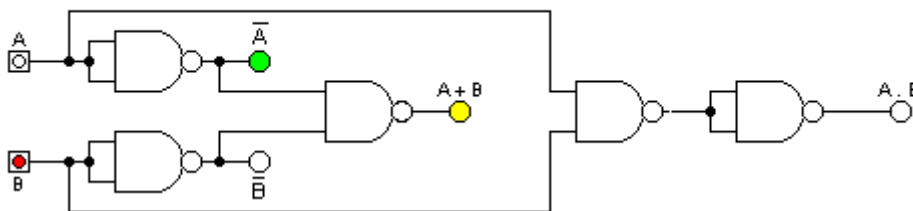
Lois de De Morgan



Système logique complet (ET, OU, Inverseur) en utilisant seulement des portes NAND :

Grâce aux lois de De Morgan, il est possible de réaliser toutes les fonctions connues rien qu'avec des portes NAND.

Système logique complet (ET, AND, NOT)



II. CIRCUIT DE COMPARAISON

➤ Réalisation d'un OU Exclusif

1. A l'aide de porte NAND et NOT

La fonction XOR (OU Exclusif), est une fonction qui est à 1 que si une et seulement une entrée est à 1.

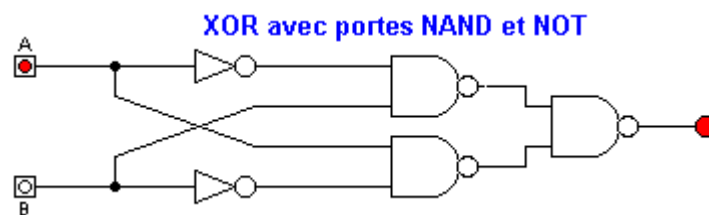


A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

D'après la table de vérité de la XOR nous pouvons déduire équation suivante :

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

Ce qui nous permet de réaliser le circuit suivant :



2. A l'aide de quatre porte NAND

➤ Comparateur ou Fonction Identique

1. Définition

Le comparateur ou la fonction identique est un circuit qui test l'égalité de deux bits.

2. Equation logique

➤ Application

1.

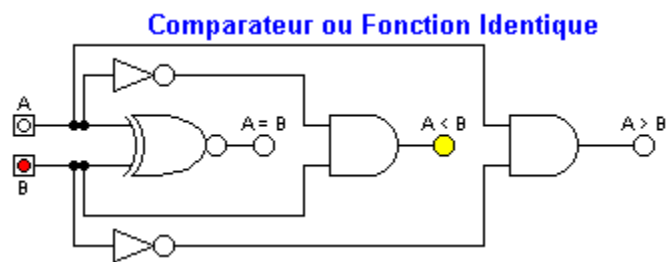
A	B	I	E	S
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Ce qui nous donne les équations suivantes :

$$I = \bar{A} \cdot B$$

$$E = \overline{A \oplus B}$$

2. Réalisation du circuit



III. SYNTHÈSE DES FONCTIONS LOGIQUES

1. Table de vérité pour chaque LED de l'afficheur

	A	B	C	D	N	NE	SE	S	SW	NW	C
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

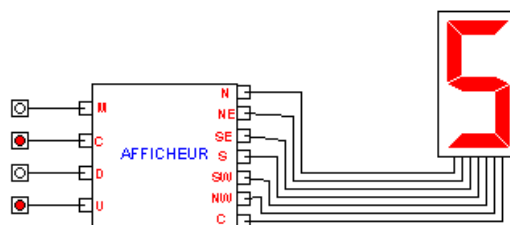
2. Equations simplifiées à l'aide des tableaux de Karnaugh

- $N = \bar{A} \cdot C + \bar{A} \cdot B \cdot D + A \cdot \bar{B} \cdot \bar{C} + \bar{B} \cdot \bar{C} \cdot \bar{D}$
- $NE = A + \bar{B} + C \cdot D + \bar{C} \cdot \bar{D}$
- $SE = A + B + \bar{C} + D$
- $S = A + \bar{B} \cdot \bar{D} + \bar{B} \cdot C + C \cdot \bar{D} + B \cdot \bar{C} \cdot D$
- $SW = \bar{A} \cdot C \cdot \bar{D} + \bar{B} \cdot \bar{C} \cdot \bar{D}$
- $NW = A + B \cdot \bar{C} + B \cdot \bar{D} + \bar{C} \cdot \bar{D}$
- $C = A + \bar{B} \cdot C + C \cdot \bar{D} + B \cdot \bar{C}$

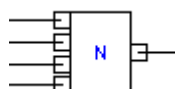
3. Réalisation de l'afficheur

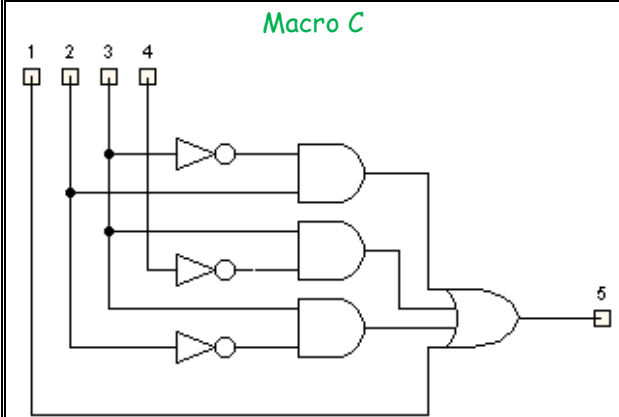
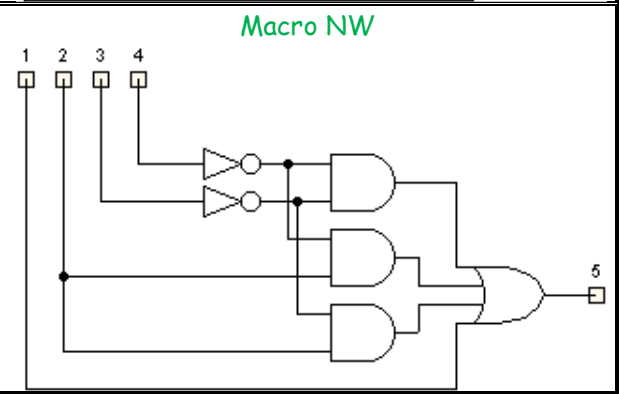
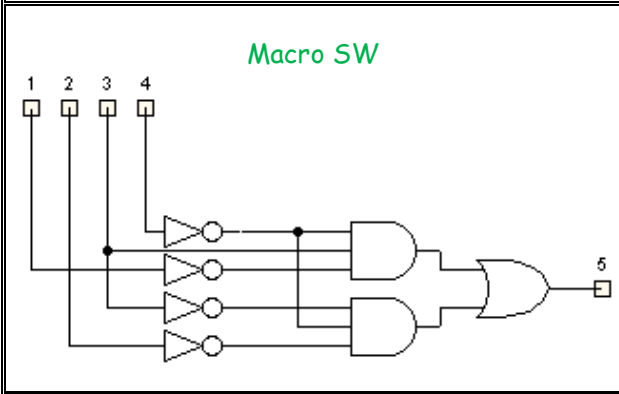
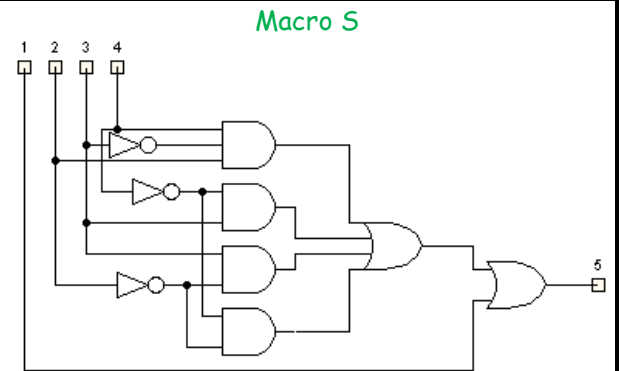
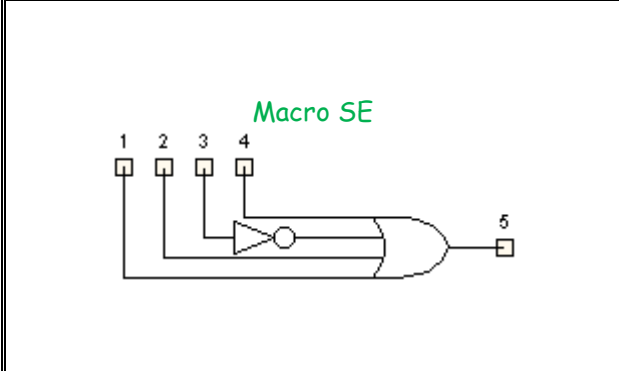
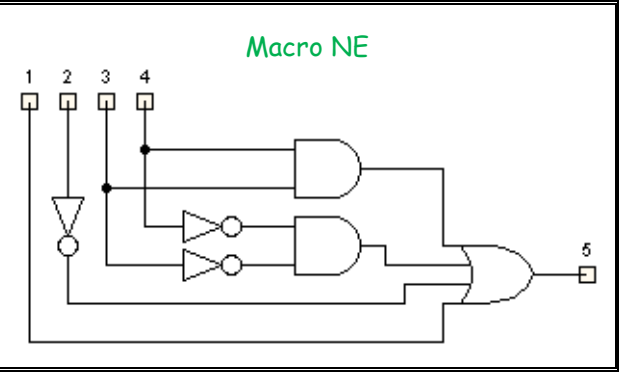
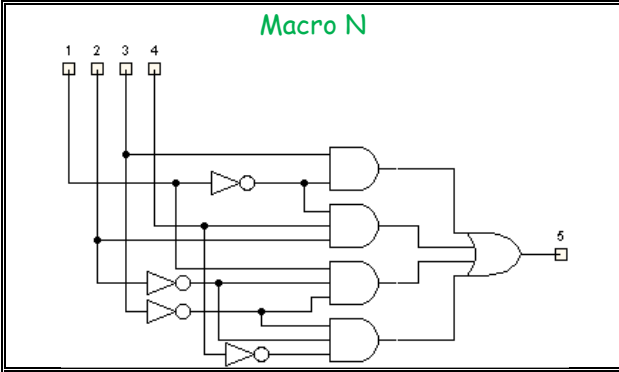
Pour ne pas trop charger le circuit nous avons préféré réaliser pour chaque LED sont propre macro.

L'afficheur étant un composant de quatre entrées et de sept sorties, sera représenté de cette manière :

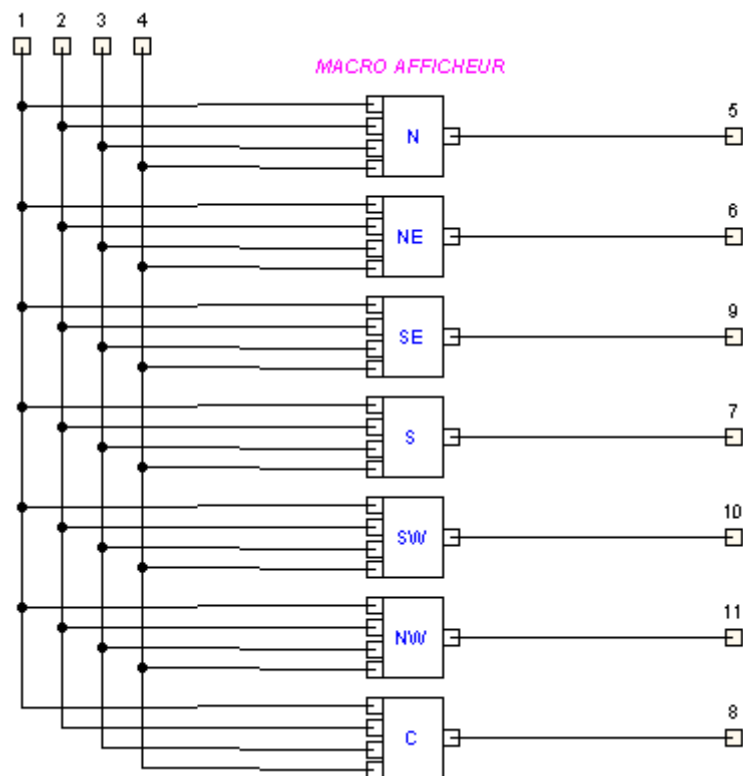


Chaque macro possède quatre entrées et une sortie ; il sera schématisé de cette manière :





Les différents macros sont relié entre eux de cette manière :



IV. DEUX CIRCUITS COMBINATOIRS IMPORTANTS

➤ Multiplexeurs

1. Définition

C'est un circuit à 2^n entrées d'information, n entrées d'adresse et 1 sortie.
On obtient en sortie l'information de la ligne de rang i si on applique l'adresse binaire i sur les entrées d'adresse.

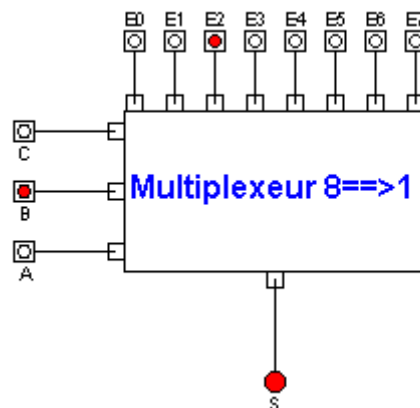
Il permet de transmettre sur une seule ligne des informations en provenance de plusieurs sources ou à destination de plusieurs cibles.

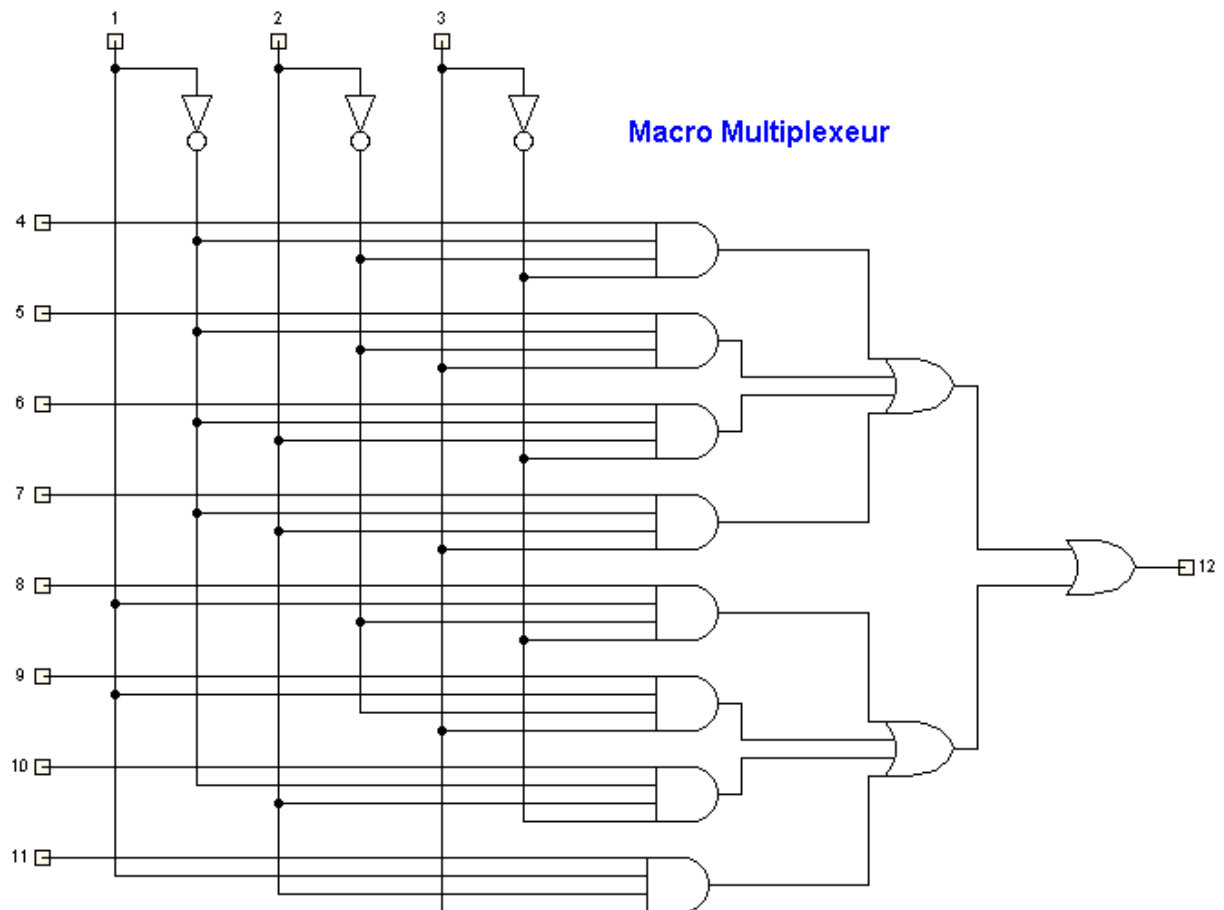
2. Réalisation d'un multiplexeur 8→1

A	B	C	S
X	X	X	0
0	0	0	E_0
0	0	1	E_1
0	1	0	E_2
0	1	1	E_3
1	0	0	E_4
1	0	1	E_5
1	1	0	E_6
1	1	1	E_7

$$S = E_0 \cdot \bar{A} \cdot \bar{B} \cdot \bar{C} + E_1 \cdot \bar{A} \cdot \bar{B} \cdot C + E_2 \cdot \bar{A} \cdot B \cdot \bar{C} + E_3 \cdot \bar{A} \cdot B \cdot C + E_4 \cdot A \cdot \bar{B} \cdot \bar{C} + E_5 \cdot A \cdot \bar{B} \cdot C + E_6 \cdot A \cdot B \cdot \bar{C} + E_7 \cdot A \cdot B \cdot C$$

Même chose pour le macro multiplexeur, nous nous proposons de créer un macro pour simplifier la schématisation du multiplexeur.





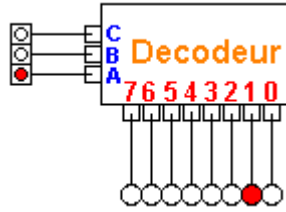
➤ Décodeurs

1. Définition

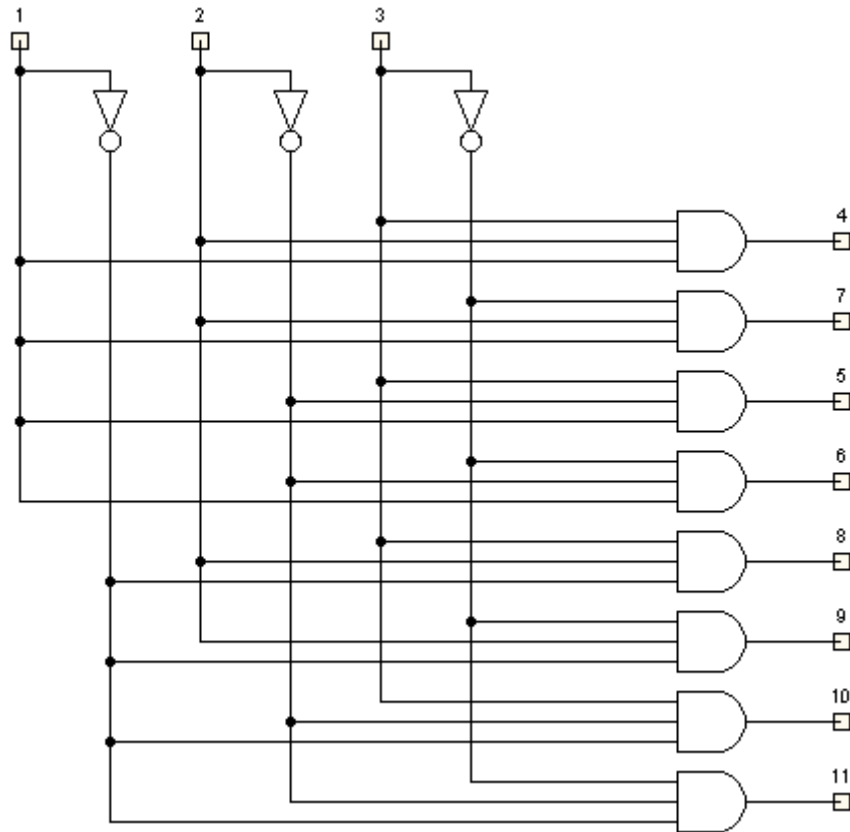
C'est un circuit combinatoire comprenant une entrée E sur n bits et 2^n sorties $O_0 \dots O_{2^n-1}$ sur 1 bit ; seul le signal O_m doit être activé lorsque l'entrée E vaut m (en binaire sur n bits).

2. Réalisation d'un décodeur

C	B	A	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
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



Et voici le macro sous sa forme développé :



V. ADDITIONNEUR

➤ Réalisation d'un additionneur 4 bits (A+B), et des indicateurs CF et OF

1. Demi-additionneur 1 bit (/+)

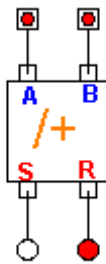
A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

On obtient le résultat suivant :

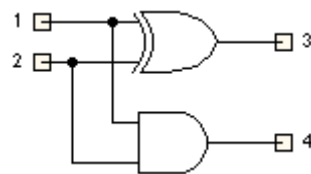
$$S = A \oplus B$$

$$R = A \cdot B$$

Voici le macro que l'on obtient



Et voilà le circuit qui permet de le réaliser



2. Additionneur 1 bit (ADD)

Dans le demi-additionneur nous avons omis la retenu de l'opération précédente.

Nous allons y remédier grâce à l'additionneur 1 bit dont voici la table de vérité :

A	B	R _i	S	R _{i+1}
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

Ce qui nous donne les équations suivantes :

$$S = \bar{A} \cdot \bar{B} \cdot R_i + \bar{A} \cdot B \cdot \bar{R}_i + A \cdot \bar{B} \cdot \bar{R}_i + A \cdot B \cdot R_i$$

$$R_{i+1} = \bar{A} \cdot B \cdot R_i + A \cdot \bar{B} \cdot R_i + A \cdot B \cdot \bar{R}_i + A \cdot B \cdot R_i$$

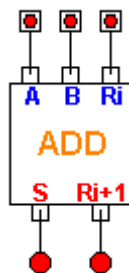
Grâce à la table de Karnaugh on obtient :

$$R_{i+1} = AB + AR + BR$$

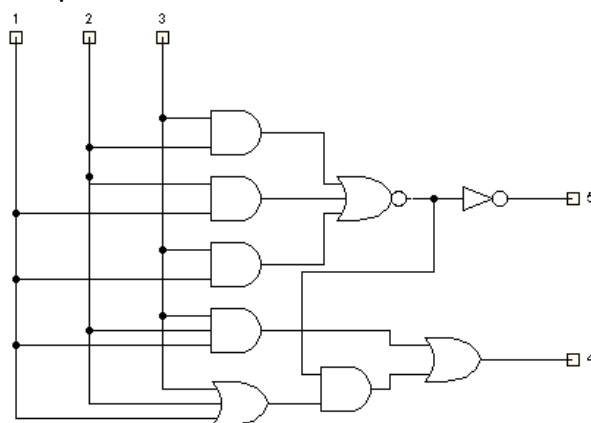
Ce qui nous permet de calculer l'équation de S en appliquant l'algèbre de Boole :

$$S = (A + B + R_i) \cdot R_{i+1} + A \cdot B \cdot R_i$$

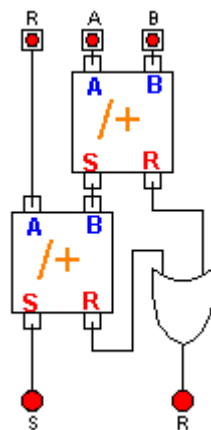
Voici le macro de notre additionneur :



Et voici le circuit correspondant



Nous pouvons aussi concevoir le même circuit grâce aux demi-additionneurs fait précédemment :

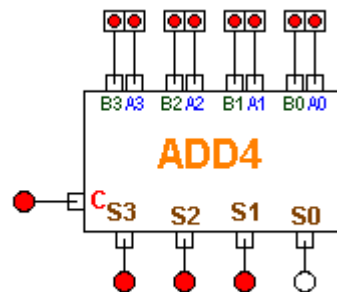


3. Additionneur 4 bits (ADD4) à l'aide de /+ et de ADD

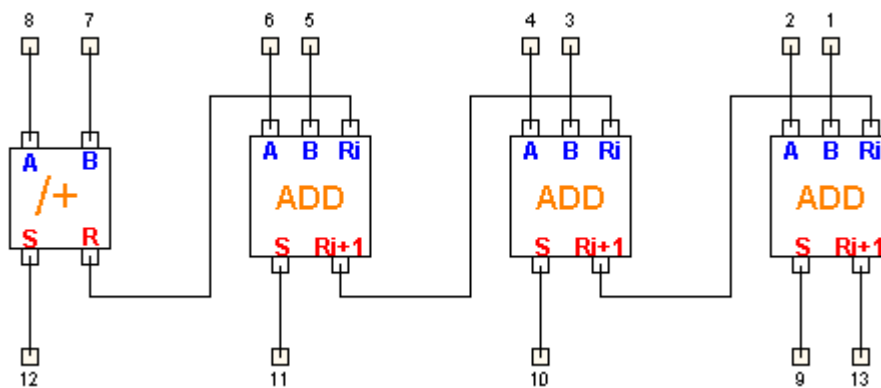
Pour la réalisation d'un additionneur 4 bits il nous faudra 4 additionneurs complets.

Les bits de poids faible n'ayant pas d'opération précédente, il est possible de réaliser le circuit à l'aide d'un demi-additionneur, et de 3 additionneurs complets. La sortie C désigne un dépassement de capacité.

Voici le macro additionneur 4 bits :



Et voici le circuit correspondant



4. Indicateur CF

L'indicateur CF comme indiqué précédemment est relié à la dernière retenue et permet de savoir s'il y a dépassement de capacité. Si c'est le cas, cela veut dire que l'opération ne peut être réalisée sur 4 bits, et qu'il faut passer à 5 bits.

5. Indicateur OF