OFPPT

ROYAUME DU MAROC

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

DIRECTION RECHERCHE ET INGENIERIE DE FORMATION

RESUME THEORIQUE
&
GUIDE DE TRAVAUX PRATIQUES

MODULE N° 21 LOGIQUE COMBINATOIRE

SECTEUR : ELECTROTECHNIQUE

SPECIALITE : EMI

NIVEAU : TECHNICIEN

ANNEE 2007

Document élaboré par :

Nom et prénom	EFP	DR
Mme ELKORNO NAIMA	CDC - GE	

Révision linguistique

-
-
-

Validation

-
-
-

SOMMAIRE

RESUME THEORIQUE 7
I. Logique booléenne : 8
I.1 Définition : 8
I.2 Les lois de l'algèbre de Boole 8
I.3 Les Théorèmes de l'algèbre de Boole : 9
I.4 Postulats de l'algèbre de Boole : 9
I.5 Simplification algébrique des équations booléennes : 9
II. Les systèmes de numération : 10
II.1 Base d'un système de numération: 10
II.2 Changement de base : 12
II.3 Opérations arithmétiques avec la base binaire: 15
II.4 Les codes : 17
III. Les fonctions logiques de base 19
III.1 Les trois fonctions logiques de base 19
III.2 Les autres fonctions logiques : 19
III.3 Les fonctions logiques matérialisées avec des interrupteurs: 21
III.4 Symbolisation 24
IV. Table de vérité 25
IV.1 Tableau des combinaisons 25
IV.2 Règles de construction d'une table de vérité : 25
IV.3 Ecriture d'une équation à partir d'une table de vérité 26
IV.4 Elaboration d'une table de vérité à partir d'une équation 28
V. Simplification des fonctions logiques par la méthode de Karnaugh : 29
V.1 Transposition d'une équation logique dans un diagramme de Karnaugh 29
V.2 Simplification d'une équation par le diagramme de Karnaugh 32
V.3 Écriture des équations à partir de regroupement 35
VI. Circuits intégrés logiques 37
VI.1 Famille TTL (transistor transistoc logic) 37
VI.2 Famille CMOS (Comptementary Métal Oxyde Semiconductor) 38
VI.3 Configuration des broches pour les différents modèles des C.I 38
VII. Schémas logiques : 42
VII.1 Généralités : 42
VII.2 Différents types de schémas logiques: 43
VIII. Utiliser une sonde logique. 49
IX. Montage des circuits 50
IX.1 Plaque de montage 50
IX.2 Interrupteurs logiques 51
IX.3 Choix de logique positive ou négative, visualisation des sorties 52
IX.4 Technique de travail 53
IX.5 Caractéristiques des circuits intégrés : 55
X. Circuits de base 57
X.1 Additionneur 57
X.2 Soustracteurs logiques 60
X.3 Multiplicateur 62

GUIDE DES EXERCICES ET TRAVAUX PRATIQUES 65
Exercices : 66
Tp 1 - portes logiques fondamentales : et (and), ou (or). Non (not) 68
Tp 2 - portes logiques fondamentales : non-et (nand), non-ou (nor), ou exclusif (xor) 70
Tp 3 - applications de l'algèbre de Boole 72
Tp 4 : applications de l'algèbre de Boole 74
Tp 5 : établir la table de vérité d'un circuit..... 75
Tp 6: monter des circuits de base 76
Evaluation de fin de module..... 79
Liste des références bibliographiques..... 80

MODULE 21:	LOGIQUE COMBINATOIRE

Code : **Durée : 45h**

OBJECTIF OPERATIONNEL

COMPORTEMENT ATTENDU

Pour démontrer sa compétence le stagiaire doit
appliquer des notions de logique combinatoire
selon les conditions, les critères et les précisions qui suivent

CONDITIONS D'EVALUATION

- *A partir :*
  - *De directives ;*
  - *De schémas;*
  - *D'une équation non simplifiée.*
- *A l'aide :*
  - *De manuels techniques ;*
  - *De fiches techniques ;*
  - *De composants logiques ;*
  - *D'outils et d'instruments de mesure ;*

CRITERES GENERAUX DE PERFORMANCE

- *Respect des règles de santé et de sécurité au travail.*
- *Pertinence de l'utilisation des outils et des instruments.*
- *Pertinence de la terminologie utilisée.*
- *Qualité des travaux.*

OBJECTIF OEPRATIONNEL	
PRECISIONS SUR LE COMPORTEMENT ATTENDU	**CRITERES PARTICULIERS DE PERFORMANCE**
A) *Appliquer des notions d'algèbre booléenne.*	- *Respect des règles de l'algèbre de Boole.*
B) *Effectuer des conversions entre des bases numériques et des codes.*	- *Exactitude des conversions.*
C) *Établir les tables de vérité d'un circuit.*	- *Respect des règles prescrites.* - *Exactitude des résultats.*
D) *Simplifier des équations par la méthode de Karnaugh.*	- *Regroupement optimal des variables.* - *Résultats exacts.*
E) *Traduire des équations en schémas.*	- *Conformité du schéma avec l'équation.* - *Tracé adéquat du schéma.*
F) *Monter des circuits de base.*	- *Sélection judicieuse des composants.* - *Conformité du montage avec le schéma.* - *Fonctionnement correct du circuit.*

Présentation du Module :

Ce module de logique combinatoire permet au stagiaire de :

- *Appliquer les notions d'algèbre de Boole;*
- *Effectuer les conversions entre des bases numériques et des codes;*
- *Établir la table de vérité d'une fonction logique;*
- *Transposer avec justesse les variables dans le tableau de Karnaugh et réduire les équations des sorties;*
- *Traduire les équations en schémas clairs, propres et conformes aux équations de départ;*
- *Choisir les composants correspondants aux fonctions logiques attendues ;*
- *Réaliser le montage du circuit choisi avec vérification du fonctionnement qui doit être conforme aux données de départ.*

La durée de ce module est de 45 h dont 25 h de théorie, 17 h de pratique et 3 h d'évaluation.

MODULE N° 21: LOGIQUE COMBINATOIRE

RESUME THEORIQUE

I. Logique booléenne :

I.1 Définition :

Beaucoup de systèmes automatisés fonctionnent en utilisant des organes et des fonctions binaires. Ces organes et fonctions binaires ne peuvent être que dans deux états possibles. Par exemple, un détecteur de niveau peut être immergé ou submergé. Un voyant peut être allumé ou éteint.

Par convention, on représente par la valeur logique « 0 » l'un de ces états et par la valeur logique « 1 » l'autre état. La valeur logique « 0 » correspond à un organe binaire (ou une fonction binaire) dans un état dit ***« non-activé »***, ***« non-actionné »*** *ou* ***« inactif »*** *(exemple : un voyant inactif est éteint). La valeur logique « 1 » correspond à un organe binaire (ou une fonction binaire) dans un état dit* ***« activé »***, ***« actionné »*** *ou* ***« actif »*** *(exemple : un voyant actif est allumé).*

La mathématique des fonctions binaires est appelée ***l'algèbre booléenne****, elle définit trois opérateurs de base ainsi qu'une foule de règles et de postulats. Ainsi, toutes les fonctions binaires (dites aussi logiques) sont des relations entre des entrées et des sorties logiques composées d'opérateurs de base et sur lesquelles on peut appliquer diverses règles d'algèbre de Boole.*

I.2 Les lois de l'algèbre de Boole

	Lois	
Commutativité	L_1	$A \bullet B = B \bullet A$
	L_2	$A + B = B + A$
Associativité	L_3	$(A \bullet B) \bullet C = A \bullet (B \bullet C)$
	L_4	$(A + B) + C = A + (B + C)$
Distributivité	L_5	$A \bullet (B + C) = A \bullet B + A \bullet C$
	L_6	$(A + B) \bullet (A + C) = A + B \bullet C$
Absorption	L_7	$A + (A \bullet B) = A$
	L_8	$A \bullet (A + B) = A$
Expansion	L_9	$(A \bullet B) + (A \bullet \overline{B}) = A$
	L_{10}	$(A + B) \bullet (A + \overline{B}) = A$
De Morgan	L_{11}	$\overline{A \bullet B} = \overline{A} + \overline{B}$
	L_{12}	$\overline{A + B} = \overline{A} \bullet \overline{B}$
	L_{13}	$\overline{\overline{A} + \overline{B}} = A \bullet B$
	L_{14}	$\overline{\overline{A} \bullet \overline{B}} = A + B$
Similitude	L_{15}	$A + \overline{A} \bullet B = A + B$
	L_{16}	$A \bullet (\overline{A} + B) = A \bullet B$

I.3 Les Théorèmes de l'algèbre de Boole :

	Théorèmes	
Invariance	T_1	$A \bullet 0 = 0$
	T_2	$A + 1 = 1$
Élément neutre	T_3	$A \bullet 1 = A$
	T_4	$A + 0 = A$
Idempotence	T_5	$A \bullet A = A$
	T_6	$A + A = A$
Complémentarité	T_7	$A \bullet \overline{A} = 0$
	T_8	$A + \overline{A} = 1$
Involution	T_9	$\overline{\overline{A}} = A$

I.4 Postulats de l'algèbre de Boole :

Postulats	
P_1	$0 \bullet 0 = 0$
P_2	$0 \bullet 1 = 1 \bullet 0 = 0$
P_3	$1 \bullet 1 = 1$
P_4	$0 + 0 = 0$
P_5	$1 + 0 = 0 + 1 = 1$
P_6	$1 + 1 = 1$
P_7	$\overline{0} = 1$
P_8	$\overline{1} = 0$

I.5 Simplification algébrique des équations booléennes :

La simplification d'une équation revient à appliquer les règles :

- *Des opérations booléennes ;*
- *Des relations fondamentales ;*
- *Du théorème de MORGAN ;*

Et à utiliser, éventuellement, comme moyens de simplification le tableau ou diagramme de Karnaugh.

Exemple :

simplifier : $z = (a + b)(\overline{b} + c)(\overline{a} + c)$

$$z = (a\,\overline{b} + a\,c + b\,\overline{b} + b\,c)(\overline{a} + c)$$

Or, $b\,\overline{b} = 0$

$z = (a\,\overline{b} + a\,c + b\,c)(\overline{a} + c)$ *développons, on aura :*

$$z = a\,\bar{b}\,\bar{a} + a\,\bar{b}\,c + a\,c\,\bar{a} + a\,c\,c + b\,c\,\bar{a} + b\,c\,c$$

Or, $a\,\bar{a} = 0$ $\quad a\,\bar{b}\,\bar{a} = 0$ $\quad \bar{a}\,a\,c = 0$ ***et*** $\quad c\,c = c$

$$z = \bar{a}\,b\,c + a\,\bar{b}\,c + a\,c + b\,c$$

$$z = c[\,a\,(\bar{b}+1) + b\,(\bar{a}+1)\,]$$

Or, $\bar{a}+1 = \bar{b}+1 = 1$

$$z = (a+b)\,c$$

II. Les systèmes de numération :

II.1 Base d'un système de numération:

La base d'un système de numération est le nombre de chiffres différents qu'utilise ce système de numération. En électronique numérique, les systèmes les plus couramment utilisés sont : le système binaire, le système octal, le système décimal et le système hexadécimal.

Se rappeler que : $a^0 = 1$.

a) Système décimal :

*C'est le système de numération décimal que nous utilisons tous les jours. C'est le système de **base 10** qui utilise donc 10 symboles différents : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9.*
Un nombre N (entier positif) exprimé dans le système de numération décimal est défini par la relation ci-dessous :

$N = a_n \times 10^n + a_{n-1} \times 10^{n-1} \ldots\ldots\ldots\ldots + a_0 \times 10^0$
(où a_n est un chiffre de rang n)

Exemple : $N = (1975)_{10}$
$N = 1 \times 10^3 + 9 \times 10^2 + 7 \times 10^1 + 5 \times 10^0$

*Les puissances de 10 sont appelées les **poids** ou les **valeurs de position**. Le poids est égal à la base élevée à la puissance de son rang.*

	Unité	Dizaine	Centaine	Milliers	10*Milliers	100*Milliers
Chiffre	a0	a1	a2	a3	a4	a5
Rang	0	1	2	3	4	5
Poids	10^0	10^1	10^2	10^3	10^4	10^5

b) Système binaire

Le système binaire est le système de base 2, c'est à dire qui utilise deux symboles différents : le 0 et le 1. Chacun d'eux est appelé bit ou élément binaire.

Dans ce système, le poids est une puissance de 2.

Exemple : $N = (10110)_2$
$N = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
$N = (22)_{10}$

- *Puissance de 2 :*

n	0	1	2	3	4	5	6	7	8	9	10	11	12
2^n	1	2	4	8	16	32	64	128	256	512	1024	2048	4096

n	13	14	15
2^n	8192	16384	32768

- *Définitions :*

Triplet : *nombre binaire formé de 3 éléments binaires.*
Quartet : *nombre binaire formé de 4 éléments binaires.*
Octet (byte) : *nombre binaire formé de 8 éléments binaires.*
Mot (word) : *nombre binaire formé de 16 éléments binaires.*
L.S.B. : *bit le moins significatif ou bit de poids faible (élément le plus à droite d'un nombre binaire).*
M.S.B. : *bit le plus significatif ou bit de poids fort (élément binaire le plus à gauche d'un nombre binaire).*

c) Système octal

*Le système de numération octal est de **base 8**, ainsi il utilise 8 symboles différents : 0, 1, 2, 3, 4, 5, 6 et 7.*
Dans ce système, le poids est une puissance de 8.

Exemple : $N = (6543)_8$
$N = 6 \times 8^3 + 5 \times 8^2 + 4 \times 8^1 + 3 \times 8^0$
$N = (3427)_{10}$

La succession des nombres par ordre croissant est le suivant :

- 1 chiffre : 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2.....etc.
- 2 chiffres : 10, 11, 12, 13, 14, 15, 16, 17, 20, 21....., 27, 30, 31....etc.

- *Puissance de 8 :*

n	0	1	2	3	4	5
8^n	1	8	64	512	4096	32768

d) Système hexadécimal

*Le système hexadécimal est de **base 16** et utilise 16 symboles différents : les dix premiers chiffres décimaux : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 et les 6 premières lettres de l'alphabet : A, B, C, D, E, F.*
La succession des nombres hexadécimaux par ordre croissant est la suivante :

- 1 chiffre : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 0, 1, 2, 3.....etc.
- 2 chiffres : 00, 01, 02 ....., 09, 0A, 0B,....., 0F, 10, 11, 12,....., 19, 1A, 1B.....etc.

Les lettres de A à F correspondent respectivement aux nombres décimaux 10 à 15.
Dans ce système, le poids est une puissance de 16.

Exemple : $N = (AC53)_{16}$
$N = A \times 16^3 + C \times 16^2 + 5 \times 16^1 + 3 \times 16^0$
$N = 10 \times 16^3 + 12 \times 16^2 + 5 \times 16^1 + 3 \times 16^0$
$N = (44115)_{10}$

- *Puissance de 16 :*

n	0	1	2	3	4	5
16^n	1	16	256	4096	65536	1048576

II.2 Changement de base :

a) Conversion des bases 2, 8 ou 16 en base 10

Pour convertir un nombre de la base 2, 8 ou 16 en nombre de base 10, il suffit de décomposer le nombre en ses quantités et d'en faire la somme.

Exemples :

$(10110, 01)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$
$= 1 \times 16 + 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 + 0 \times 0, 5 + 1 \times 0, 25 = (22, 25)_{10}$

$(372, 06)_8 = 3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 + 0 \times 8^{-1} + 6 \times 8^{-2}$
$= 3 \times 64 + 7 \times 8 + 2 \times 1 + 0 \times 0, 125 + 6 \times 0, 015625 = (250, 09375)_{10}$

$(FD, 2A)_{16} = F \times 16^1 + D \times 16^0 + 2 \times 16^{-1} + A \times 16^{-2}$
$= 15 \times 16 + 13 \times 1 + 2 \times 0, 0625 + 10 \times 0, 00390625 = (253, 1640625)_{10}$

Tableau de correspondance entre nombre de différentes bases

Décimal (base 10)	Binaire (base 2)	Octal (base 8)	Hexadécimal (base 16)
	2^4 2^3 2^2 2^1 2^0		
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8

9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11

b) Conversion d'un nombre décimal en un nombre d'un système d'une autre base

Cette conversion se fait en deux parties :

1- *La partie entière.*
2- *La partie fractionnaire.*

Problème : Un nombre N étant donné en base 10, cherchons à l'écrire dans un système de base b.

- *On traite d'abord la partie entière :*

Pour la partie entière nous la divisons par la base b et nous conservons le reste. Le quotient obtenu est divisé par b et nous conservons le reste. Il faut répéter l'opération sur chaque quotient obtenu. Les restes successifs sont écrits, en commençant par le dernier, de la gauche vers la droite pour former l'expression de N dans le système de base b.

- *On traite après la partie fractionnaire :*

Exemple 1 *: Conversion du nombre $(3786, 4)_{10}$ en base 2 :*

- *On traite d'abord la partie entière :3786*

3786 | 2	1893 | 2
0 | 1893	1 | 946
946 | 2	473 | 2
0 | 473	1 | 236
236 | 2	118 | 2
0 | 118	0 | 59
59 | 2	29 | 2
1 | 29	1 | 14

14 | 2	7 | 2
0 | 7	**1** | 3
3 | 2	1 | 2
1 | 1	**1** | 0

Le nombre binaire ainsi obtenu est :

$3786 = (111011001010)_2$

- *On traite après la partie fractionnaire :*

$0,4 \times 2 = \underline{0}\,,8$

$0,8 \times 2 = \underline{1},6$

$0,6 \times 2 = \underline{1},2$

$0,2 \times 2 = \underline{0},4$

$0,4 \times 2 = \underline{0},8$

D'où $(0,4)_{10} = (0,01100)_2$

Résultat : $(3786,4)_{10} = (111011001010,01100)_2$

Exemple 2 : $(459,3)_{10}$ *le convertir en base 8.*

- *Partie entière :*

Position		Reste	459/8
8^0	↑	3	57/8
8^1		1	7/8
8^2		7	0

$(459)_{10} = (713)_8$

- *Partie fractionnaire :*

$0,3 \times 8 = \underline{2},4$
$0,4 \times 8 = \underline{3},2$
$0,2 \times 8 = \underline{1},6$
$0,6 \times 8 = \underline{4},8$
$(0,3)_{10} = (0,2314)_8$

Résultat : $(459,3)_{10} = (713,2314)_8$

c) Autres conversions

- **Conversion d'un nombre octal en un nombre binaire :**

Chaque symbole du nombre écrit dans le système octal est remplacé par son équivalent écrit dans le système binaire à trois bits (voir tableau de correspondance).

Exemple : $N = (257)_8 = (010\ \ 101\ \ 111)_2$
$2\ \ 5\ \ 7$

- **Conversion d'un nombre binaire en un nombre octal :**

C'est l'opération inverse de la précédente. Il faut regrouper les 1 et 0 du nombre trois par trois en commençant par la droite, puis chaque groupe est remplacé par le chiffre octal correspondant.

Exemple : $N = (11001101111)_2 = 11\ \ 001\ \ 101\ \ 111$
$3\ \ 1\ \ 5\ \ 7$
$N = (3157)_8$

- **Conversion d'un nombre hexadécimal en un nombre binaire :**

Chaque symbole du nombre hexadécimal est remplacé par son équivalent écrit sur quatre bits dans le système binaire.

Exemple : $N = (B\ F\ 8)_{16}$
$N = (1011\ \ 1111\ \ 1000)_2$
B F 8

- **Conversion d'un nombre binaire en un nombre hexadécimal :**

C'est l'inverse de la précédente. Il faut donc regrouper les 1 et 0 du nombre par quartet en commençant par la droite, puis chaque groupe est remplacé par le symbole hexadécimal correspondant.

Exemple : $N = (100001101111)_2$

$N = 1000\ \ 0110\ \ 1111$
$8\ \ 6\ \ F$
$N = (86F)_{16}$

II.3 Opérations arithmétiques avec la base binaire:

a) Addition Binaire : Les règles de base sont :

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 reporte 1

Exemple :

```
   1 1 1     Reports
   1101        0
 +
   1011        1
  11000        1
```

b) Soustraction Binaire : *Les règles de base sont*

0 - 0 = 0
0 - 1 = 1 Emprunte 1
1 - 0 = 1
1 - 1 = 0

Exemple :

```
     0          Emprunt
   11011          1
 -
     110          1
   10101          0
```

c) Multiplication Binaire : *Les règles de base sont :*

*0 * 0 = 0*
*0 * 1 = 0*
*1 * 0 = 0*
*1 * 1 = 1*

Exemple :

```
    101
  *
    110
    000
   101
  101
  11110
```

d) Division Binaire : *Les règles de base sont :*

0 / 0 = Indéterminé
0 / 1 = 0
1 / 0 = Impossible
1 / 1 = 1
Exemple :

```
 1010    /  10
-10       | 101 |
 001
-00
  10
 -10
  00
```

II.4 Les codes :

a) Code binaire :

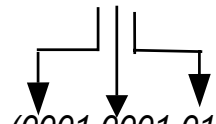
Décimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

b) Code B C D :

(Binary coded décimal) en français (Décimal codé Binaire)

Décimal	B C D
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Exemple : $(115)_{10}$



$= (0001\ 0001\ 0101)_{BCD}$

c) Code ASC II

(American standard code for information interchange)

Ou code américain pour l'échange d'information : c'est un code alphanumérique qui permet de représenter des chiffres, des lettres ainsi que divers caractères spéciaux. Il traduit ces caractères en langage machine.

Colonne C: caractère ASCII ou fonction de contrôle particulière.
Colonne D: décimal.
Colonne O: octal.
Colonne H: hexadécimal.

D	O	H	C	D	O	H	C	D	O	H	C	D	O	H	C
0	000	00	nul	32	040	20	sp	64	100	40	@	96	140	60	‘
1	001	01	soh	33	041	21	!	65	101	41	A	97	141	61	a
2	002	02	stx	34	042	22	‘‘	66	102	42	B	98	142	62	b
3	003	03	etx	35	043	23	#	67	103	43	C	99	413	63	c
4	004	04	eot	36	044	24	$	68	104	44	D	10	144	64	d
5	005	05	enq	37	045	25	%	69	105	45	E	101	145	65	e
6	006	06	acq	38	046	26	&	70	106	46	F	102	146	66	f
7	007	07	bel	39	047	27	`	71	107	47	G	103	147	67	g
8	010	08	BS	40	050	28	(	72	110	48	H	104	150	68	h
9	011	09	HT	41	051	29	)	73	111	49	I	105	151	69	i
10	012	0A	LF	42	052	2A	*	74	112	4A	J	106	152	6A	j
11	013	0B	VT	43	053	2B	+	75	113	4B	K	107	153	6B	k
12	014	0C	FF	44	054	2C	’	76	114	4C	L	108	154	6C	l
13	015	0D	CR	45	055	2D	-	77	115	4D	M	109	155	6D	m
14	016	0E	SO	46	056	2E	.	78	116	4E	N	110	156	6E	n
15	017	0F	SI	47	057	2F	/	79	117	4F	O	111	157	6F	o
16	020	10	dle	48	060	30	0	80	120	50	P	112	160	70	p
17	021	11	dc1	49	061	31	1	81	121	51	Q	113	161	71	q
18	022	12	dc2	50	062	32	2	82	122	52	R	114	162	72	r
19	023	13	dc3	51	063	33	3	83	123	53	S	115	163	73	s
20	024	14	dc4	52	064	34	4	84	124	54	T	116	164	74	t
21	025	15	nak	53	065	35	5	85	125	55	U	117	165	75	u
22	026	16	syn	54	066	36	6	86	126	56	V	118	166	76	v
23	027	17	etb	55	067	37	7	87	127	57	W	119	167	77	w
24	030	18	can	56	070	38	8	88	130	58	X	120	170	78	x
25	031	19	em	57	071	39	9	89	131	59	Y	121	171	79	y
26	032	1A	sub	58	072	3A	:	90	132	5A	Z	122	172	7A	z
27	033	1B	esc	59	073	3B	;	91	133	5B	[	123	173	7B	{
28	034	1C	fs	60	074	3C	<	92	134	5C	\	124	174	7C	|
29	035	1D	gs	61	075	3D	=	93	135	5D	]	125	175	7D	}
30	036	1E	rs	62	076	3E	>	94	136	5E	^	126	176	7E	~
31	037	1F	us	63	077	3F	?	95	137	5F	-	127	177	7F	del

d) Code Gray :

(Code binaire réfléchi, ne peut être utilisé pour les opérations arithmétiques).

Nombre Décimal	*Binaire*	*Gray*
0	*0000*	*0000*
1	*0001*	*0001*
2	*0010*	*0011*
3	*0011*	*0010*
4	*0100*	*0110*
5	*0101*	*0111*
6	*0110*	*0101*
7	*0111*	*0100*
8	*1000*	*1100*
9	*1001*	*1101*
10	*1010*	*1111*
11	*1011*	*1110*
12	*1100*	*1010*
13	*1101*	*1011*
14	*1110*	*1001*
15	*1111*	*1000*

C'est une autre forme de la base binaire.
Un seul bit à la fois change d'état lorsqu'on passe d'un nombre au suivant.

III. Les fonctions logiques de base

III.1 Les trois fonctions logiques de base

Les fonctions logiques reposent sur trois opérateurs de base. Ce sont les fonctions logiques : « NON » (en anglais « NOT »), « ET » (en anglais « AND ») et « OU » (en anglais « OR »). Nous les présentons avec leurs équations et leurs tables de vérité.

a) La fonction logique « NON » :

Soit une variable booléenne nommée A. La fonction logique NON (A), appelé complément de A, sera notée $\overline{A}$ *(lire A barre). Le résultat de NON (A) sera également une variable booléenne. La table ci-contre est* ***la table de vérité*** *de cette fonction.*

Fonction F=NON(A)	
A	F
0	1
1	0

Au niveau algébrique, ***l'équation*** *correspondant à cette table de vérité est : F =* $\overline{A}$

b) La fonction logique « ET » :

Soit deux variables booléennes nommées A et B. Le résultat de la fonction logique A ET B sera également une variable booléenne. Le tableau de droite montre ***la table de vérité*** *de cette fonction. la fonction logique ET n'active la sortie que lorsque toutes les entrées sont actives.*
Au niveau algébrique, ***l'équation*** *correspondant à cette table de vérité est F =* $A \bullet B$ *. Le symbole du ET est semblable à celui du produit.*

Fonction F = A ET B		
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

c) La fonction logique « OU » :

Soit deux variables booléennes nommées A et B. Le résultat de la fonction logique A OU B sera également une variable booléenne. ***La table de vérité*** *de cette fonction est montrée à droite. La fonction logique OU donne une valeur de sortie égale à 1 dès qu'une des entrées est à 1.*

Au niveau algébrique, ***l'équation*** *correspondant à cette table de vérité est F =* $A+B$ *. Le symbole du OU est semblable à celui de la somme.*

Fonction F = A OU B		
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

III.2 Les autres fonctions logiques :

En plus des opérateurs de base, il existe d'autres opérateurs de deux variables et nous présentons ici les plus importants.

a) Opérateur « OUI » ou opérateur égalité

La sortie est à l'état 1 si, et seulement si, l'entrée est à l'état 1.

b) La fonction logique « NON-ET » :

Soit deux variables booléennes nommées A et B. Le résultat de la fonction logique NON(A ET B) sera également une variable booléenne. La figure de droite montre ***la table de vérité*** *de cette fonction. La sortie de la fonction logique NON-ET (en anglais NAND) a un comportement inverse à celle de la fonction logique ET.*

Fonction "NON-ET"		
A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

Au niveau algébrique, ***l'équation*** *correspondant à cette table de vérité est F =* $\overline{A \bullet B}$ *. L'ajout de la barre au-dessus de la fonction traduit ainsi l'inversion du résultat du ET.*

c) La fonction logique « NON-OU »

Soit deux variables booléennes nommées A et B. Le résultat de la fonction logique NON (A OU B) sera également une variable booléenne. La table ci-contre montre ***la table de vérité*** *de cette fonction. La sortie de la fonction logique NON-OU (en anglais NOR) a un comportement inverse à celle de la fonction logique OU. Au niveau algébrique,* ***l'équation*** *correspondant à cette table de vérité est F =* $\overline{A+B}$ *. L'ajout de la barre au-dessus de la fonction traduit ainsi l'inversion du résultat du OU.*

Fonction "NON-OU"		
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Remarque :

Ces deux fonctions (NON-ET et NON-OU) sont très utilisées en électronique, car elles représentent des éléments de connections universels. Toute fonction logique peut en effet être écrite exclusivement à partir de l'une ou l'autre de ces fonctions. Un des avantages de ces éléments de connexion universel est de permettre l'implantation de n'importe quelle fonction logique à l'aide d'un seul type de circuit électronique. Il y a donc ***standardisation*** *sur un seul type de circuit électronique. Ainsi, il est possible d'en acheter une grande quantité pour bénéficier d'un prix de vente avantageux. On ne stocke qu'un seul type de circuit en prévision d'éventuelles pannes.*

d) La fonction logique « OU-EXCLUSIF »

Soit deux variables booléennes nommées A et B. Le résultat de la fonction logique A OUEXCLUSIF B sera également une variable booléenne. ***La table de vérité*** *de droite montre le comportement de cette fonction.*

Fonction "OU-EXCL"		
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

la sortie de la fonction logique OUEXCLUSIF (en anglais EXOR) ne donne un niveau logique 1 que si, et seulement si, une seule entrée est à l'état 1.

Au niveau algébrique, ***l'équation*** *correspondant à cette table de vérité est F =* $A \oplus B$ *. Cette expression peut aussi être écrite sous une autre forme : F =* $A \oplus B = A\overline{B} + \overline{A}B$

e) La fonction logique « NON-OU-EXCLUSIF »

Soit deux variables booléennes nommées A et B. Le résultat de la fonction logique NON (A OUEXCLUSIF B) sera également une variable booléenne. La figure de droite montre ***la table de vérité*** *de cette fonction. La sortie de la fonction logique NON-OUEXCLUSIF est l'inverse de celle obtenue avec la fonction logique OU-EXCLUSIF.*

Fnct "NON-OU-EXC"		
A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Au niveau algébrique, ***l'équation*** *correspondant à cette table de vérité est F =* $\overline{A \oplus B}$ *.*

III.3 Les fonctions logiques matérialisées avec des interrupteurs:

Comme il était mentionné précédemment un interrupteur est un organe binaire pouvant être actionné (lorsque l'opérateur appui dessus), ou non-actionné (lorsque l'opérateur n'y touche pas). Selon le type d'interrupteur, il peut être actionné par un opérateur (bouton de commande), par un mécanisme (interrupteur de fin de course), par la détection de la présence d'une grandeur physique (détecteur de niveau, de température, …).

Dans sa position de repos, un interrupteur peut laisser passer ou non le courant électrique. Si l'interrupteur doit être actionné pour qu'il laisse passer le courant, nous avons un interrupteur dit « normalement ouvert ». Par contre, si en actionnant l'interrupteur le courant est coupé (ou ne passe plus), nous avons un interrupteur dit « normalement fermé »

Une ampoule (ou un voyant) est un organe binaire qui peut être actionné (lorsque le courant passe et que l'ampoule est allumée), ou non-actionné (lorsque le courant ne passe plus et que l'ampoule est éteinte).

Dans la suite nous verrons comment, en utilisant les interrupteurs comme entrées logiques et une ampoule comme sortie obtenir des fonctions logiques.

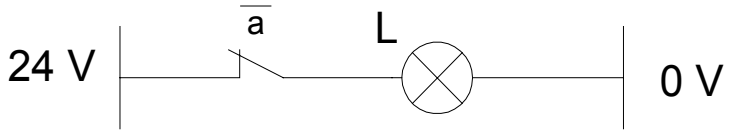
a) Fonction logique « OUI »

La figure ci-dessous montre la fonction logique OUI. Cette fonction utilise simplement un interrupteur normalement ouvert. Le voyant s'allumera si l'interrupteur est actionné.

La fonction logique de ce montage est : L = a

b) Fonction logique « NON »

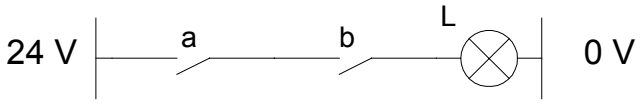
La figure ci-dessous montre la fonction logique NON. Pour obtenir cette fonction, il suffit de brancher un interrupteur normalement fermé. Le voyant s'allume tant que l'interrupteur n'est pas actionné.



La fonction logique de ce montage est : L = $\overline{a}$

c) Fonction logique « ET »

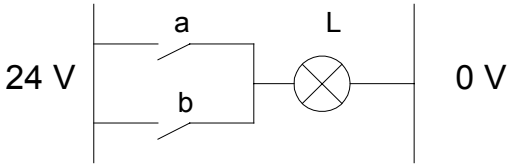
La figure ci-dessous montre la fonction logique ET. Pour obtenir cette fonction, il suffit de brancher deux interrupteurs normalement ouverts en série. Pour que le courant puisse traverser le voyant et l'allume, il faut actionner simultanément les deux interrupteurs.



La fonction logique de ce montage est : L = $a \bullet b$

d) Fonction logique « OU »

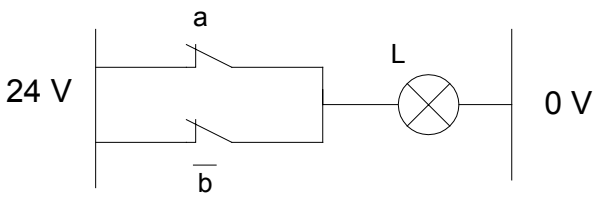
La figure ci-dessous montre la fonction logique OU. Pour obtenir cette fonction, il suffit de brancher deux interrupteurs normalement ouverts en parallèle. Le voyant s'allume dès que l'un des interrupteurs est actionné.



La fonction logique de ce montage est : L = $a + b$

e) Fonction logique « NON-ET »

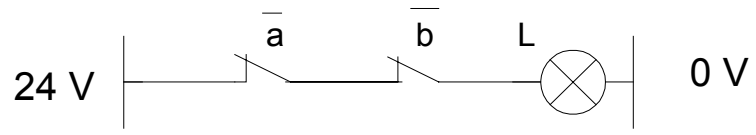
La figure ci-dessous montre la fonction logique NON-ET. Pour obtenir cette fonction, il suffit de brancher deux interrupteurs normalement fermés en parallèle. Le voyant s'éteint seulement si les deux interrupteurs sont actionnés simultanément.



La fonction logique de ce montage est : L = $\overline{a \bullet b} = \overline{a} + \overline{b}$

f) Fonction logique « NON-OU »

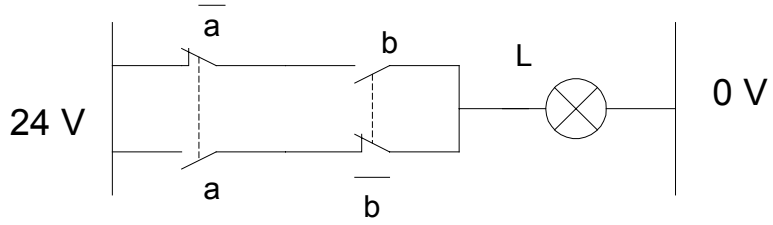
La figure ci-dessous montre la fonction logique NON-OUT. Pour obtenir cette fonction, il suffit de brancher deux interrupteurs normalement fermés en série. Le voyant s'éteint dès qu'un des interrupteurs est actionné.



La fonction logique de ce montage est : L = $\overline{a+b} = \overline{a} \times \overline{b}$

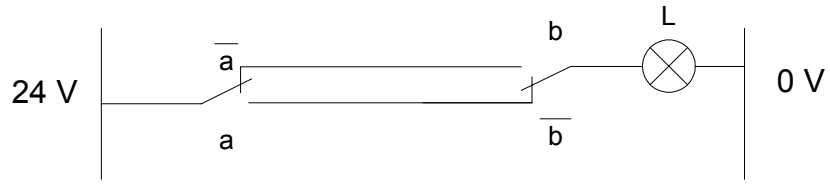
g) Fonction logique « OU-EXCLUSIF »

La figure ci-dessous montre la fonction logique OU-EXCLUSIF. Pour obtenir cette fonction, il suffit de brancher, tel que montré, deux interrupteurs ayant deux contacts chacun, l'un normalement ouvert, l'autre normalement fermé. Le voyant s'allume si et seulement si un seul interrupteur est actionné.



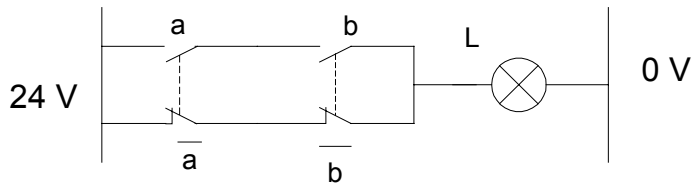
La fonction logique de ce montage est : $\mathrm{L} = a \oplus b = \overline{a}\, b + \overline{b}\, a$

ou encore :



h) Fonction logique « NON-OU-EXCLUSIF »

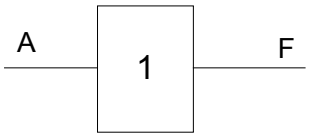
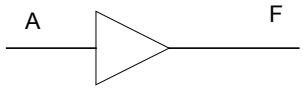
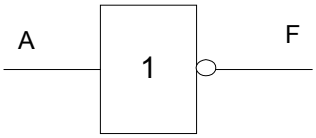
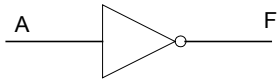
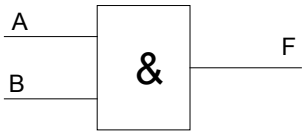
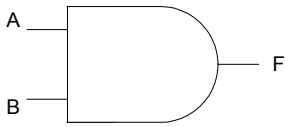
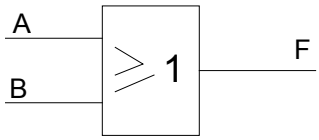
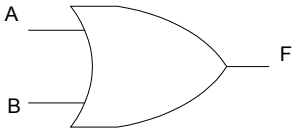
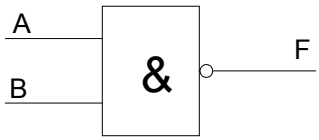
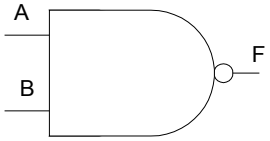
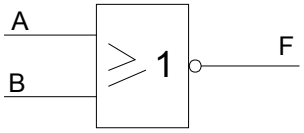
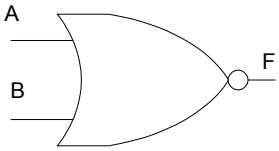
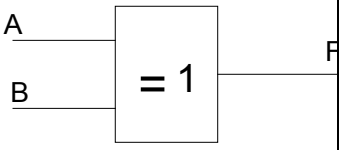
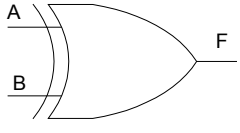
La figure ci-dessous montre la fonction logique NON-OU-EXCLUSIF. Pour obtenir cette fonction, il suffit de brancher, tel que montré, deux interrupteurs ayant deux contacts chacun, l'un normalement ouvert, l'autre normalement fermé. Le voyant s'éteint si et seulement si un seul interrupteur est actionné.



La fonction logique de ce montage est :

L = $\overline{a \oplus b} = \overline{a\,\overline{b} \quad + b\overline{a}} = \overline{a\,\overline{b}} \times \overline{b\,\overline{a}} = (\overline{a}+b)\times(\overline{b}+a) = \overline{a}\times\overline{b} + ab =$

III.4 Symbolisation

Fonction	Symboles	
	NFC03-212	Américain
OUI : F = A		
NON : F = $\overline{A}$		
ET : F = $A \bullet B$ (AND)		
OU : F = $A + B$ (OR)		
NON ET : F = $\overline{A \bullet B}$ (NAND)		
NON OU : F = $\overline{A + B}$ (NOR)		
OU exclusif : F = $A \oplus B$		

IV. Table de vérité

IV.1 Tableau des combinaisons

De nombreux circuits logiques possèdent plusieurs entrées mais seulement une sortie. Une table de vérité nous fait connaître la réaction d'un circuit logique (sa valeur de sortie) aux diverses combinaisons de niveaux logiques appliqués aux entrées.

Pour une table de N entrés il y a 2^N lignes.

On peut construire un tableau de ces combinaisons comportant autant de colonnes que de variables d'entrées et autant de lignes que de combinaisons.
Pour le remplir, il suffit d'écrire pour chaque ligne l'équivalent binaire des nombres décimaux à compter de 0 à 2^{n-1}.

Exemples :

a) 2 variables A et B on a 2^2 = 4 combinaisons à compter de 0 à 3.

A	B	
0	0	→ L'équivalent binaire de 0
0	1	→ L'équivalent binaire de 1
1	0	→ L'équivalent binaire de 2
1	1	→ L'équivalent binaire de 3

b) 3 variables A, B et C on a 2^3 = 8 combinaisons à compter de 0 à 7.

	A	B	C
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

IV.2 Règles de construction d'une table de vérité :

La table de vérité est une compilation, sous forme de tableau, de tous les états logiques de la sortie en fonction des états logiques des entrées.

Les étapes à suivre pour construire une table de vérité :

- *Écrire, sur une première ligne, le nom des variables d'entrées et celui de variable de sortie;*
- *Diviser le tableau en un nombre de colonnes égal au total des entrées et de la sortie;*
- *Déterminer le nombre de combinaisons possibles à l'aide des variables d'entrée : soit $2^{\text{nombre d'entrée}}$*
- *Tracer des lignes horizontales en un nombre égal au nombre de combinaisons possibles;*
- *Remplir chaque ligne par une combinaison possible des variables d'entrée : ça revient à compter en binaire de 0 à $(2^n - 1)$;*
- *Inscrire, dans la colonne « sortie », la valeur de la fonction pour chaque combinaison.*

***Exemple** : Soit $S = A \bullet B + B \bullet C$. La table de vérité à 3 variables d'entrée.*

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	0
1	1	0	1
1	1	1	1

IV.3 Ecriture d'une équation à partir d'une table de vérité

Il existe 2 méthodes :

a) Produit de sommes :

On considère les lignes de la table de vérité dont la sortie est à l'état logique « 0 » sous forme d'une somme logique « OU ».
Les parties d'équation ainsi obtenues peuvent être réunies par le produit logique « ET ».

***Exemples** :*

1) *Soit la table de vérité suivante à 2 variables :*

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

1E ligne : $(A + B)$

4e ligne : $(\overline{A} + \overline{B})$

⇨ *équation :* $S = (\overline{A} + \overline{B}) \bullet (A + B)$

2) Soit la table de vérité suivant à 3 variables :

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

→ 2e ligne : $\left(A+B+\overline{C}\right)$

→ 4e ligne : $\left(A+\overline{B}+\overline{C}\right)$

→ 7e ligne : $\left(\overline{A}+\overline{B}+C\right)$

⇨ équation :

$$S=\left(\overline{A}+\overline{B}+C\right)\bullet\left(A+\overline{B}+\overline{C}\right)\bullet\left(A+B+\overline{C}\right)$$

Remarque :

Variable = 1 ⇨ $\overline{\text{Variable}}$

Variable = 0 ⇨ Variable

b) Somme de produits :

On considère les lignes de la table de vérité dont la sortie est à l'état logique « 1 » sous forme d'un produit logique « ET ».

Les parties d'équation ainsi obtenues peuvent être réunies par la somme logique « OU ».

Exemples :

1)

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

→ 2e ligne : $\left(\overline{A}\bullet B\right)$

→ 3e ligne : $\left(A\bullet\overline{B}\right)$

⇨ L'équation : $S=\left(\overline{A}\bullet B\right)+\left(A\cdot\overline{B}\right)$

2)

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

→ 1eligne : $\left(\overline{A}\bullet\overline{B}\bullet\overline{C}\right)$

→ 3eligne : $\left(\overline{A}\bullet B\bullet\overline{C}\right)$

→ 5eligne : $\left(A\bullet\overline{B}\bullet\overline{C}\right)$

→ 6eligne : $\left(A\bullet\overline{B}\bullet C\right)$

→ 8eligne : $\left(A\bullet B\bullet C\right)$

⇨ l'équation :

$$S=\left(\overline{A}\bullet\overline{B}\bullet\overline{C}\right)+\left(\overline{A}\bullet B\bullet\overline{C}\right)+\left(A\bullet\overline{B}\bullet\overline{C}\right)+\left(A\bullet\overline{B}\bullet C\right)+\left(A\bullet B\bullet C\right)$$

c) Équivalence entre le résultat d'un produit de sommes et celui d'une somme de produits

Le résultat d'un produit de sommes est égal à celui d'une somme de produits.

Exemple :

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

Somme de produits $\Longrightarrow$ $S = \overline{A} \bullet B + A \bullet B$ *(1)*

Produit de sommes $\Longrightarrow$ $S = (A + B) \bullet (\overline{A} + B)$ *(2)*

Preuve de l'égalité de ces deux équations (1) et (2) :

(1) $\Longrightarrow$ $S = \overline{A} \bullet B + A \bullet B$

$= (\overline{A} + A) \bullet B$ *(Distributivité L_5)*

$= 1 \bullet B$ *(T_8 : Complémentarité)*

$= B$ *(T_3 : Élément neutre)*

(2) $\Longrightarrow$ $S = (A + B) \bullet (\overline{A} + B)$

$= A \bullet \overline{A} + A \bullet B + \overline{A} \bullet B + B \bullet B$ *(Distributivité L_6)*

$= 0 + A \bullet B + \overline{A} \bullet B + B$ *(T_7, T_5)*

$= B + B$ *(L_9 : Expansion)*

$= B$ *(T_6 : Idempotence)*

IV.4 Elaboration d'une table de vérité à partir d'une équation

Nous pouvons passer d'une équation logique à une table de vérité. La technique est fort simple, il suffit de tester l'équation logique avec toutes les combinaisons d'entrée et de trouver pour chaque combinaison la valeur de la sortie. Un exemple simple montrera ce qu'il en est.

Soit F = $(A \bullet B) + \overline{C}$ *. Pour trouver la table de vérité de F, il faut premièrement connaître le nombre d'entrées. Ici, nous en avons trois, soit A, B et C. Donc la table de vérité devrait avoir normalement quatre (3+1) colonnes et huit (2^3) lignes. Ainsi la table résultante sera :*

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

V. Simplification des fonctions logiques par la méthode de Karnaugh :

Les tables de vérité permettent de trouver les équations logiques d'un système logique. Malheureusement, ces équations logiques ne sont pas simplifiées donc il faut les simplifier avant de réaliser les fonctions logiques correspondantes. Bien sûr, les règles, postulats et théorèmes de l'algèbre booléenne appliqués à une équation logique mènent à la simplification, mais au prix d'un certain effort. Plus le nombre d'entrées est élevé, plus la simplification est fastidieuse.

Pour simplifier les équations logiques de façon plus efficace. Une technique appelée ***« Diagramme de Karnaugh »*** *est utilisée. Cette méthode présente la table de vérité sous une forme matricielle mettant en évidence les simplifications logiques.*

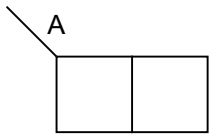
V.1 Transposition d'une équation logique dans un diagramme de Karnaugh

a) Diagramme de Karnaugh

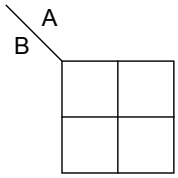
C'est un diagramme qui reprend les indications de la table de vérité pour les mettre sous une autre forme. Le nombre de cases est égal au nombre de lignes de la table de vérité, ou encore au nombre de combinaisons des variables d'entrée.

Exemples :

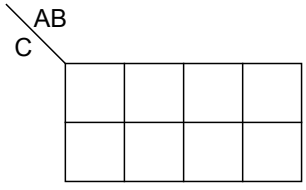
a) 1 variable d'entrée A ⇨ 2^1 *combinaisons = 2 cases.*



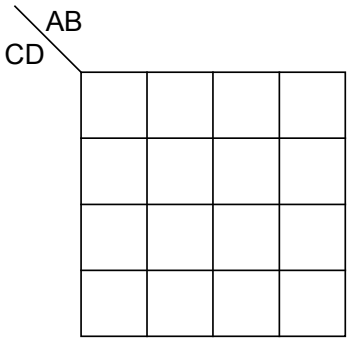
b) 2 variables d'entrée A et B ⇨ 2^2 *combinaisons = 4 cases.*



c) 3 variables d'entrée A, B, et C ⇨ 2^3 *combinaisons = 8 cases.*



d) 4 variables d'entrée A. B. C. et D ⟹ 2^4 combinaisons = 16 cases.



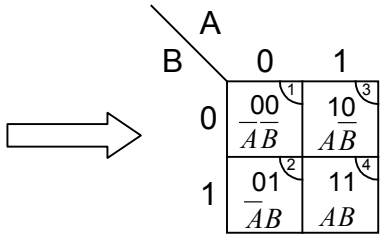
b) Disposition des combinaisons à l'intérieur du diagramme de Karnaugh

Pour pouvoir simplifier par suite l'équation à partir du diagramme de Karnaugh, il faut qu'une seule variable change d'état pour deux cases adjacentes. On utilise donc le code Gray au lieu du code binaire.

Exemples :

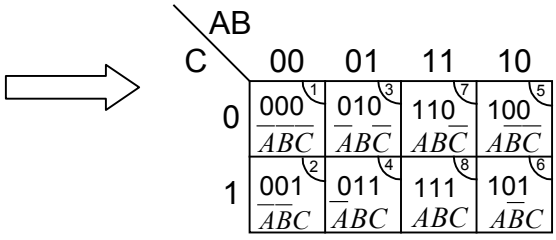
1.

	A	B	S
1-	0	0	
2-	0	1	
3-	1	0	
4-	1	1	



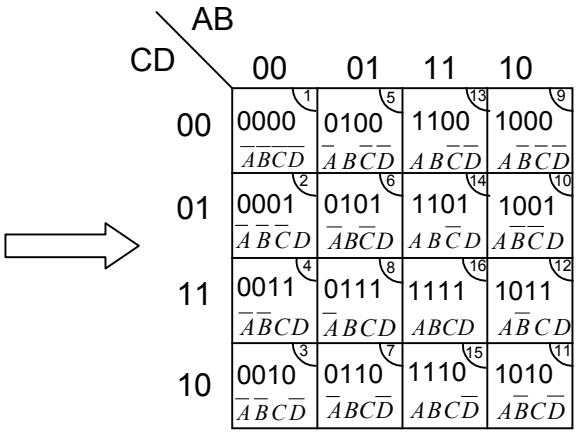
2.

	A	B	C	S
1-	0	0	0	
2-	0	0	1	
3-	0	1	0	
4-	0	1	1	
5-	1	0	0	
6-	1	0	1	
7-	1	1	0	
8-	1	1	1	



3.

	A	B	C	D	S
1-	0	0	0	0	
2-	0	0	0	1	
3-	0	0	1	0	
4-	0	0	1	1	
5-	0	1	0	0	
6-	0	1	0	1	
7-	0	1	1	0	
8-	0	1	1	1	
9-	1	0	0	0	
10-	1	0	0	1	
11-	1	0	1	0	
12-	1	0	1	1	
13-	1	1	0	0	
14-	1	1	0	1	
15-	1	1	1	0	
16-	1	1	1	1	



c) Transposition d'une équation logique dans un diagramme de Karnaugh

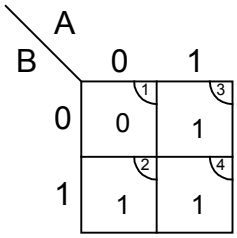
Exemples :

a) Soit l'équation : $S=A+B$

- Table de vérité

	A	B	S
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

- Diagramme de Karnaugh

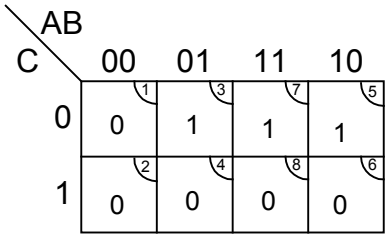


b) Soit l'équation : $S=A\bullet\overline{C}+B\bullet\overline{C}$

- Table de vérité

	A	B	C	S
1	0	0	0	0
2	0	0	1	0
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	0
7	1	1	0	1
8	1	1	1	0

- Diagramme de Karnaugh

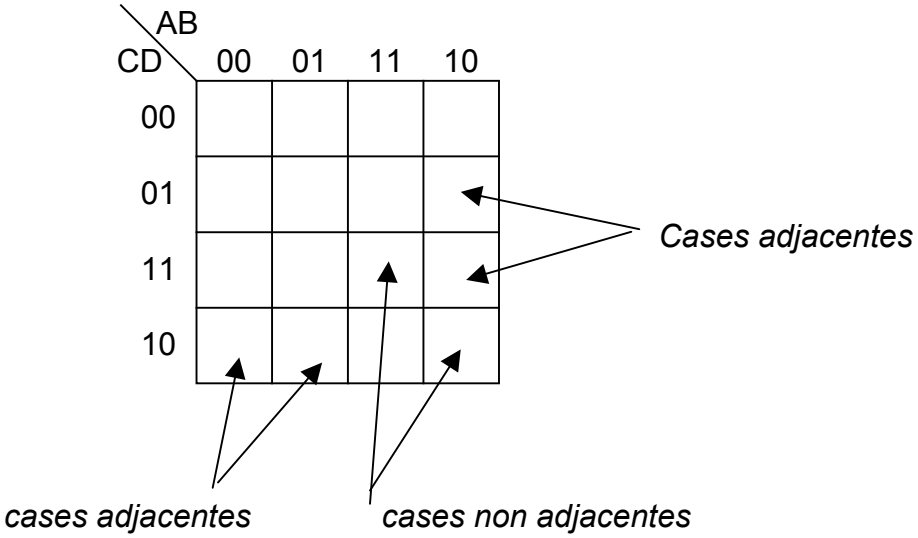


V.2 Simplification d'une équation par le diagramme de Karnaugh

a) Cases adjacentes

Deux cases sont adjacentes lorsqu'elles sont situées côte à côte, que ce soit à l'horizontale ou à la verticale. De plus, une seule variable doit changer d'état pour que deux cases soient considérées comme adjacentes.

Exemples :



b) Règles de regroupement :

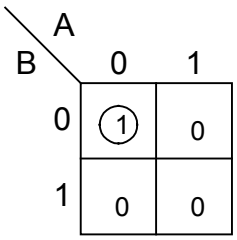
Le regroupement des cases adjacentes permet de réduire une équation logique le plus simplement possible. Pour se faire, certaines règles doivent être respectées :

***Règle 1** :*

Le regroupement des cases adjacentes doit se faire par puissance de deux :
2^0, 2^1, 2^2, 2^3, ….(1, 2, 4, 8 ….)

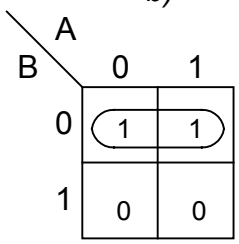
Exemples :

a)



case unique

b)



groupement de deux

c)

C \ AB	00	01	11	10
0	1	1	0	0
1	1	1	0	0

groupement de quatre

d)

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

groupement de huit

Règle 2 *:*

Les cases appartenant au même groupement doivent avoir la même valeur binaire de la variable de sortie. (voir les exemples précédents).

Règle 3 *:*

La longueur et la hauteur des groupements doivent être des puissances de deux.

Exemple :

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

2^2 ou 4

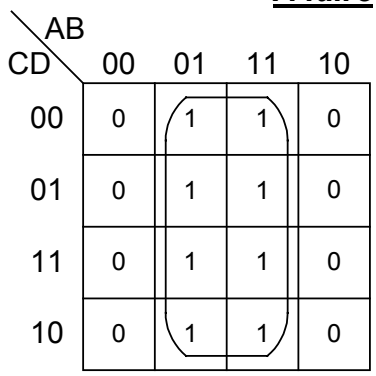
2^0 ou 1 2^1 ou 2

Règles 4 *:*

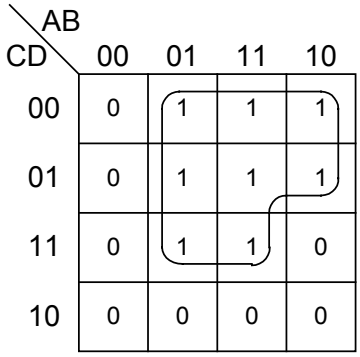
Les regroupements de quatre casses ou plus doivent être disposés symétriquement par rapport à l'un des axes du diagramme.

Exemples :

A faire



A ne pas faire

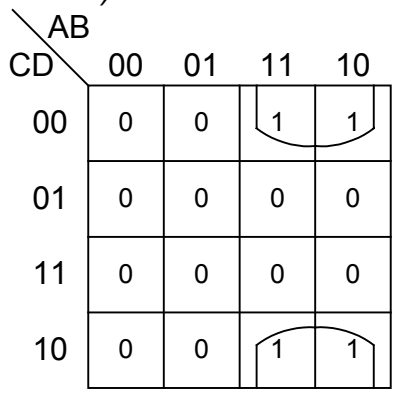


Règles 5 :

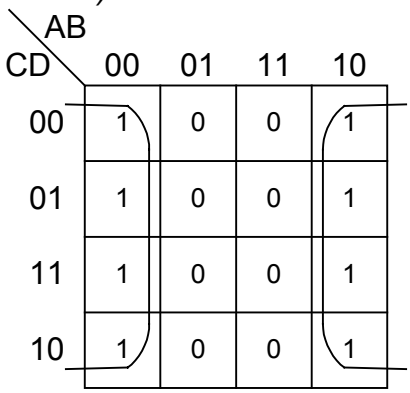
Les cases des extrémités de gauche peuvent être regroupées avec celles de droite, avec celles des bords hauts ou encore avec celles du bas.

Exemples :

a)



b)

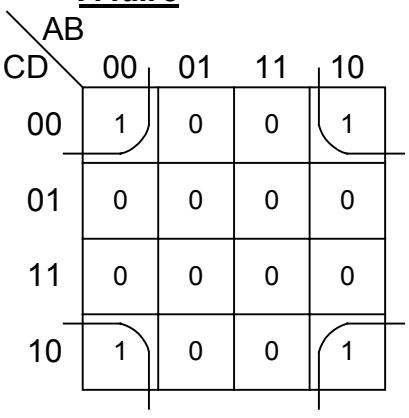


Règle 6 :

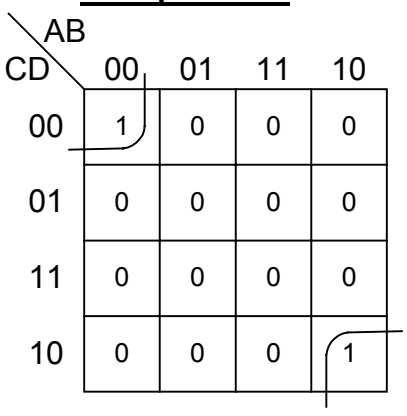
Les quatre cases des 4 coins d'un diagramme de Karnaugh peuvent être regroupées.

Exemples :

A faire



A ne pas faire



V.3 Écriture des équations à partir de regroupement

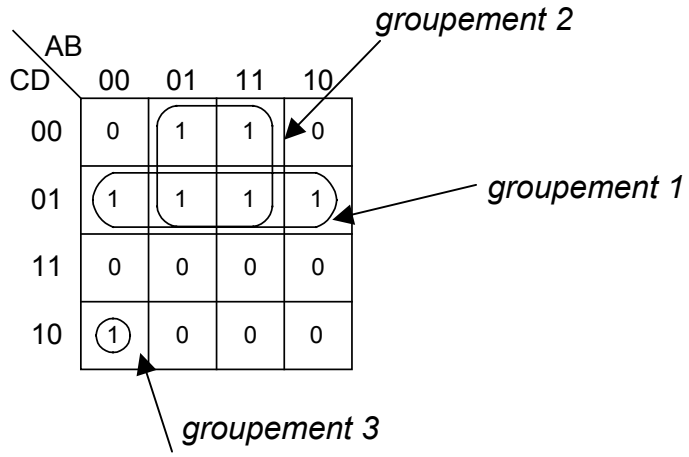
a) Somme de produits :

Chaque regroupement de 1 donne le produit logique des variables d'entrée qui n'ont pas changé d'état. L'ensemble de ces regroupements est une somme logique.

Règle : B = 1 ⟹ On la représente par B ;
B = 0 ⟹ On la représente par $\overline{B}$.

Exemples :

a)



Groupement 1 : A et B changent d'état ⟹ *L'équation du groupement :* $\overline{C}\bullet D$

C = 0 et D = 1 ne changent pas d'état

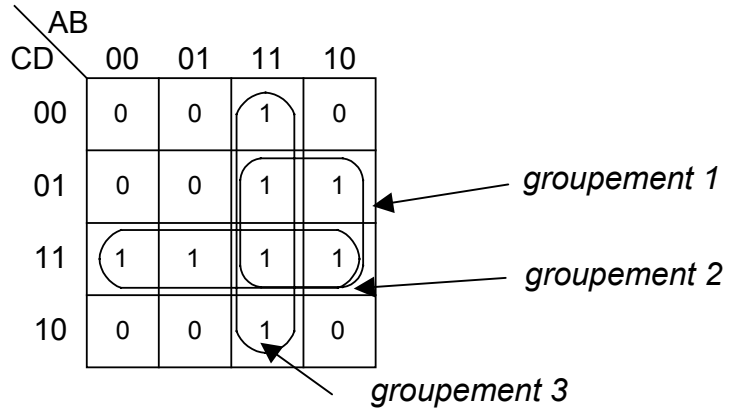
Groupement 2 : A et D changent d'état ⟹ *L'équation du groupement :* $B\bullet\overline{C}$

B = 1 et C = 0 ne changent pas d'état

Groupement 3 : A = 0, B = 0, C = 1 et D = 0 ne change pas d'état ⟹ *L'équation du groupement :* $\overline{A}\bullet\overline{B}\bullet C\bullet\overline{D}$

D'où l'équation finale : $S=\overline{C}\bullet D+B\bullet\overline{C}+\overline{A}\bullet\overline{B}\bullet C\bullet\overline{D}$

b)



Groupement 1 : $A \bullet D$

Groupement 2 : $C \bullet D$ $\Longrightarrow$ $S = A \bullet D + C \bullet D + A \bullet B$

Groupement 3 : $A \bullet B$

b) Produit de sommes

Chaque regroupement de 0 donne la somme logique des variables d'entrée qui n'ont pas changé d'état. L'ensemble de ces regroupements est un produit logique.

Règle : *B = 1* $\Longrightarrow$ *On la représente par* $\overline{B}$;
B = 0 $\Longrightarrow$ *On le représente par B.*

Exemples :

a)

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

groupement 1 *groupement 2*

Groupement 1 : A, B ne changent pas d'état (A = 0, B = 0)
C et D changent d'état $\Longrightarrow$ *L'équation du groupement :* $A+B$

Groupement 2 : A, B ne changent pas l'état (A = 1, B = 1)
C et D changent $\Longrightarrow$ *L'équation du groupement :* $\overline{A}+\overline{B}$

D'où l'équation finale : $S = (A+B) \bullet (\overline{A}+\overline{B})$

b)

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

Un seul groupement : A change d'état

B, C et D ne changent pas d'état
(B = 1, C = 1, D = 0)

⟹ *L'équation du groupement :* $\overline{B}+\overline{C}+D$

D'où l'équation finale : $S=\overline{B}+\overline{C}+D$

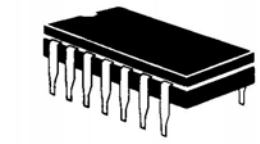
VI. Circuits intégrés logiques

La matérialisation des fonctions logiques a été d'abord réalisée avec des composants discrets puis elle s'est transformée en intégrant plusieurs composants sur un seul circuit.
Un circuit intégré (C.I.) désigne un bloc constitué par un monocristal de silicium de quelques millimètres carrés en forme parallélépipède rectangle aplati, à l'intérieur duquel se trouve inscrit en nombre variable des composants électroniques (transistors, diodes, résistances et, plus rarement des condensateurs).

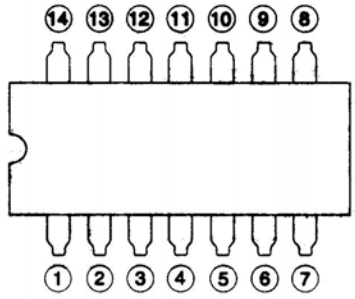
VI.1 Famille TTL (transistor transistoc logic)

Elle fait principalement usage de combinaisons de transistors bipolaires pour la fabrication des circuits intégrés.

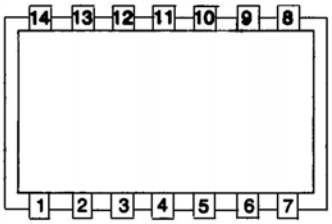
Ces C.I. sont constitués d'un boîtier qui contient la puce, laquelle est reliée à l'extérieur par un certain nombre de pattes (ou broches). Ce nombre varie généralement entre 14 et 28. La tension d'alimentation est +5V.



Vue en trois dimensions



Vue de dessus avec identification des pattes



Représentation généralement employée pour l'identification

Circuit intégré

Suivant la gamme de température d'utilisation, on distingue deux séries des C.I TTL :

- *La série 5400 : gamme de température d'utilisation militaire indiquée par 5 (-55°, +125°C);*
- *La série 7400 : gamme de température d'utilisation générale indiquée par 7 (0°,+70°C).*

La famille TTL se subdivise en cinq sous-groupes, dont chacun possède ses propres caractéristiques de fonctionnement.

TTL standard	*74XX*
TTL low power (faible consommation)	*74LXX*
TTL schottky	*74SXX*
TTL fast	*74FXX*
TTL low power schottky	*74LSXX*
TTL advanced schottky	*74ASXX*
TTL advanced low power schottky	*74ALSXX*

VI.2 Famille CMOS (Comptementary Métal Oxyde Semiconductor)

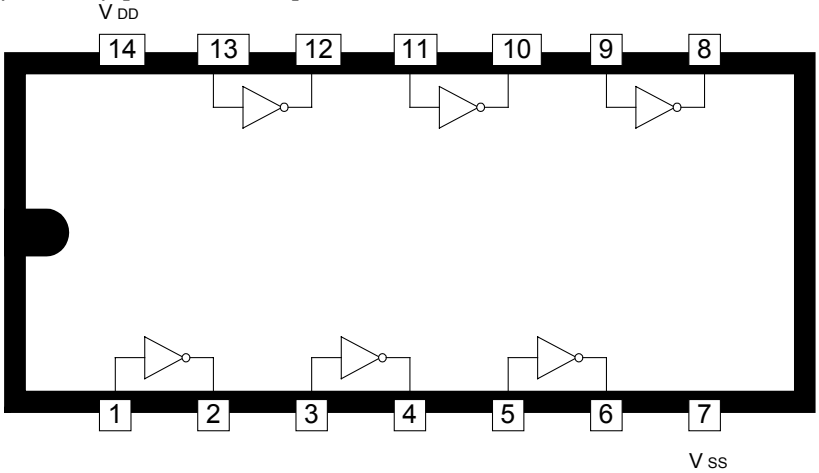
Elle dérive principalement des transistors à effet de champ. Cette famille se divise en deux sous groupes :

- *Le type A, qui peut fonctionner à des tensions variant de + 3V à +12 V (+15V maximum);*
- *Le type B, qui peut fonctionner à des tensions variant de +3V à + 18V (+20V maximum*

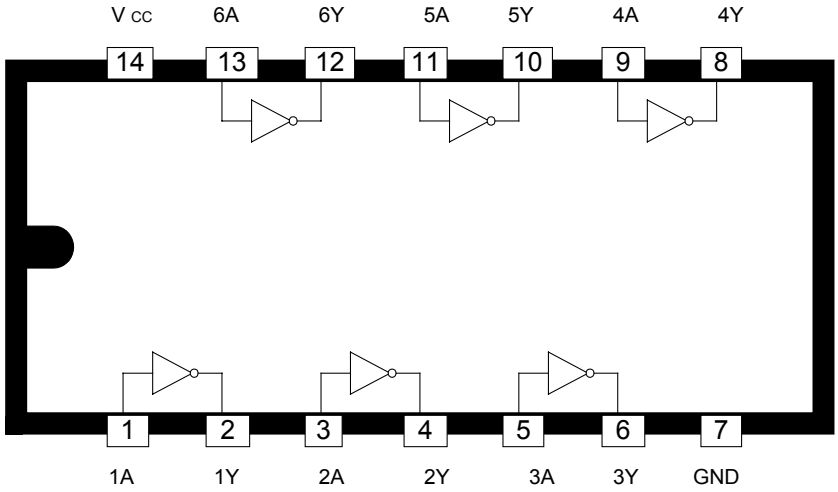
VI.3 Configuration des broches pour les différents modèles des C.I

a) Circuit intégré portes «NON»

- *Six portes NON : 4069 (CMOS) [CD4069UB]*

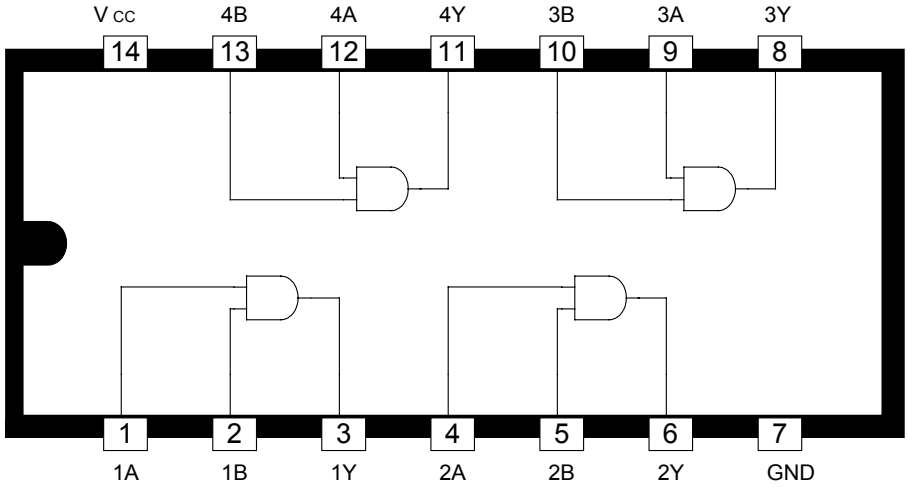


- *Six portes NON : 7404 (TTL) [SN74LSD4]*

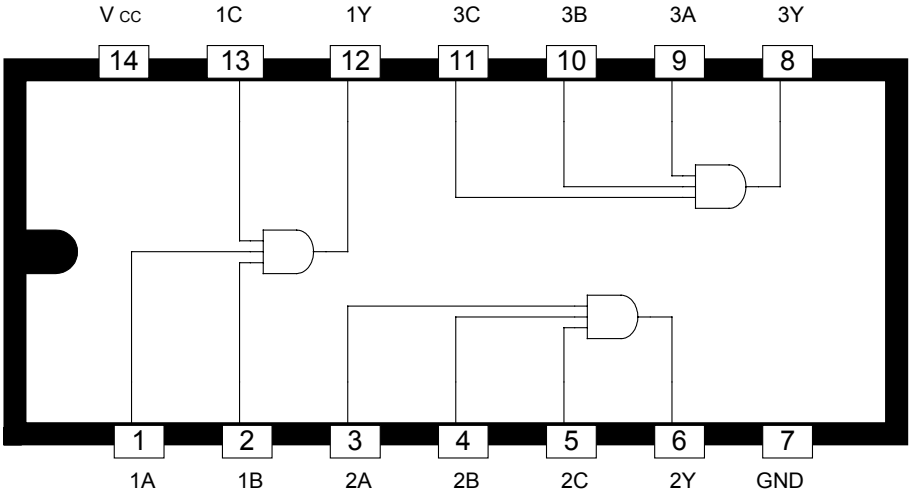


b) ***Circuit intégré Portes «ET»***

- *Quatre portes «ET» à deux entrées : 7408 (TTL) [SN74LSD8]*

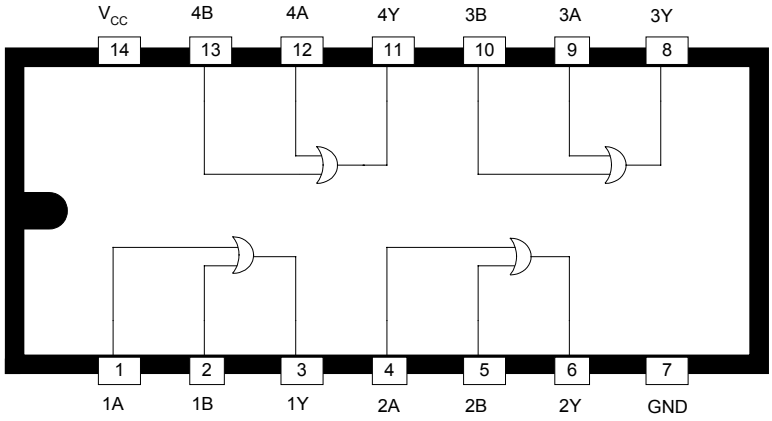


- *Trois portes «ET» à trois entrées : 7411 (TTL) [SN74LS11]*

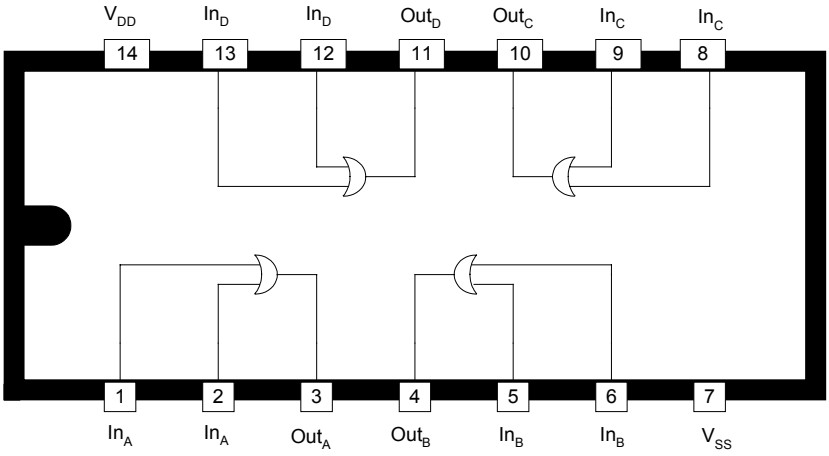


c) Circuits intégrés portes «OU»

- *Quatre portes «OU» à deux entrées : 7432 (TTL) [SN74LS32]*

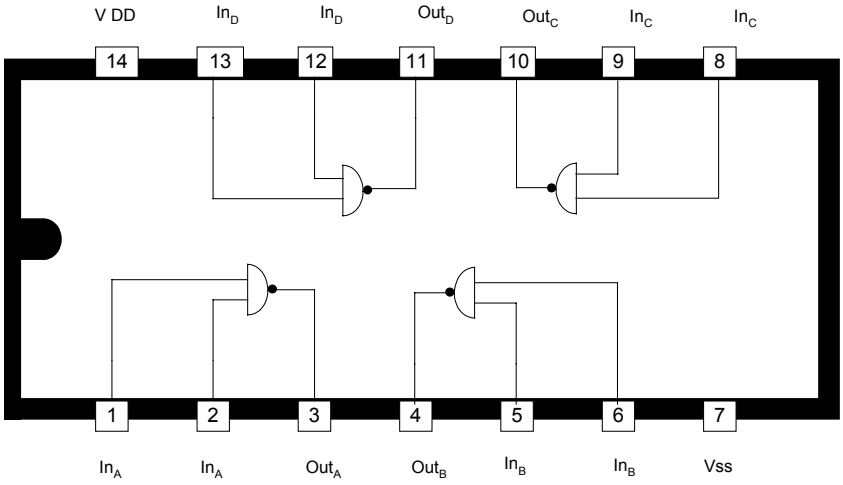


- *Quatre portes «OU» à deux entrées : 4071 (COMS) [CD4071B]*

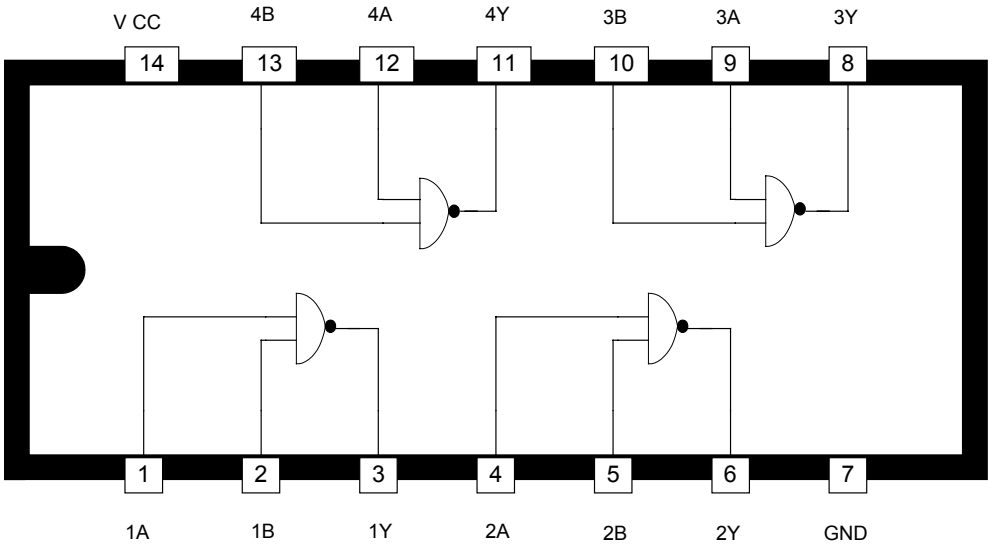


d) Circuits intégrés « NON ET »

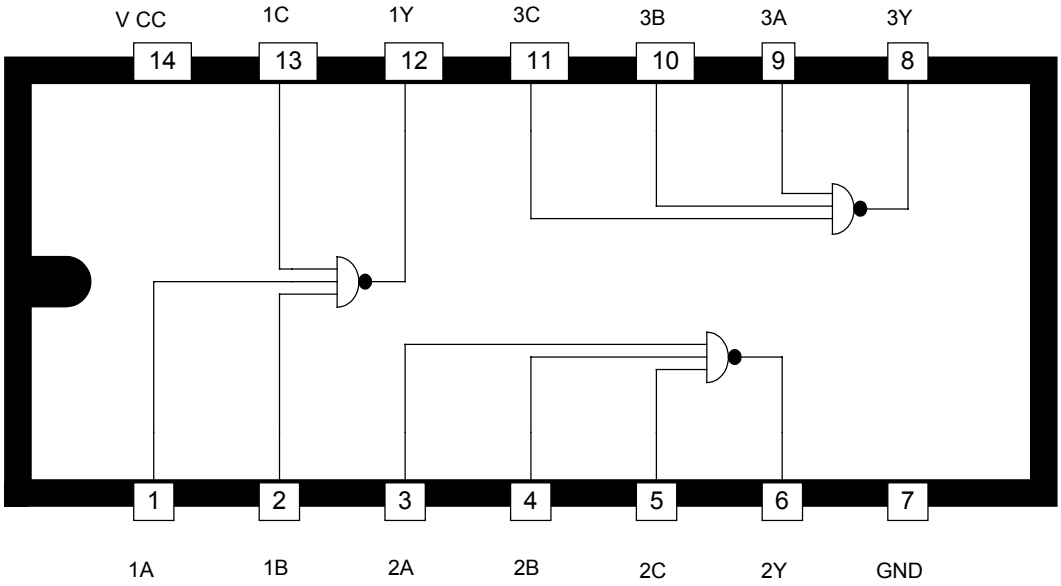
- *Quatre portes «NON ET» à 2 entrées : 4011 (C-MOS) [CD4011*



- *Quatre portes «NON ET» à entrées : 7400 (TTL) [SN74LS00]*

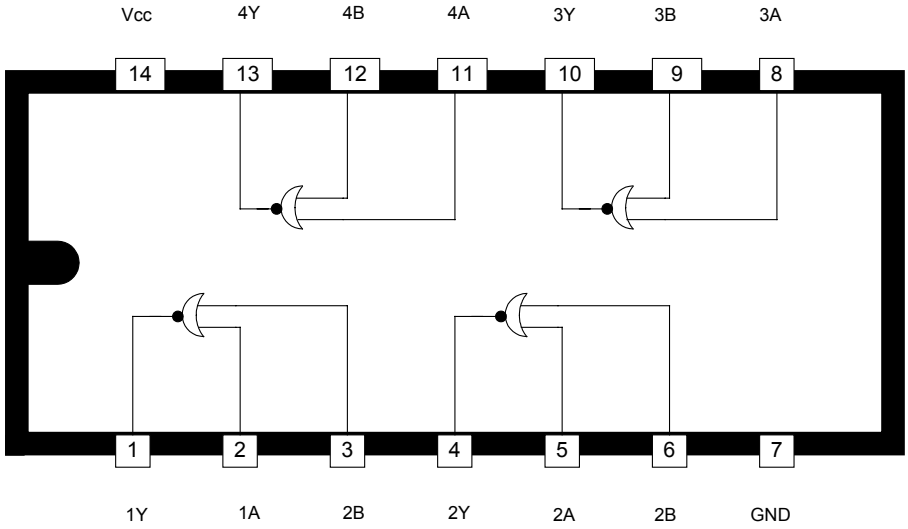


- *Trois portes «NON ET» à 3 entrées : 7410 (TTL) [SN74LS10]*



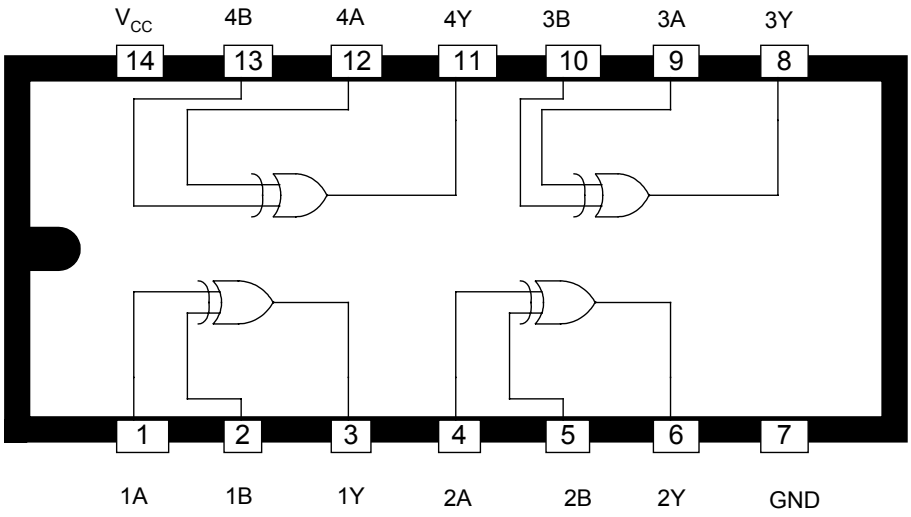
e) ***Circuits intégrés portes «NON OU»***

- *Quatre portes «NON OU» à 2 entrées : 7402 (TTL) [SN74LS02]*



f) Circuits intégrés portes «Ou exclusif» :

- *Quatre portes « OU exclusif » à 2 entrées : 7486 (TTL) [SN 74LS86]*



VII. Schémas logiques :

VII.1 Généralités :

Un schéma logique est la représentation graphique de l'équation d'une ou plusieurs variables de sortie grâce aux opérateurs de base vus précédemment.

On distingue 3 types de schémas logiques :

- *Le 1er type comprend des opérateurs NON, ET, OU;*
- *Le 2^{éme} type ne comprend que des opérateurs NON ET (NAND);*
- *Le 3^{éme} type ne comprend que des opérateurs NON OU (NOR).*

VII.2 Différents types de schémas logiques:

a) Schéma logique comprenant des opérateur NON, ET,OU

Pour traduire une équation en schéma logique avec ces opérateurs, il faut :

- *Déterminer le nombre d'opérateurs NON➔compter le nombre des variables complimentées.*
- *Déterminer le nombre d'opérateurs ET ➔ compter le nombre de groupes de produits logiques et déduire le nombre d'entrées nécessaires sur chaque opérateur.*
- *Déterminer le nombre d'opérateur OU ➔ compter le nombre de groupes de sommes logiques et déduire le nombre d'entrées nécessaires sur chaque opérateur.*
- *Relier les différents opérateurs de base.*

Exemples :

a) $R_1 = B_1 \bullet B_2 \bullet \overline{B_3} + B_4 \bullet \overline{B_1} \bullet B_5 + \overline{B_4} \bullet B_2 \bullet \overline{B_3}$

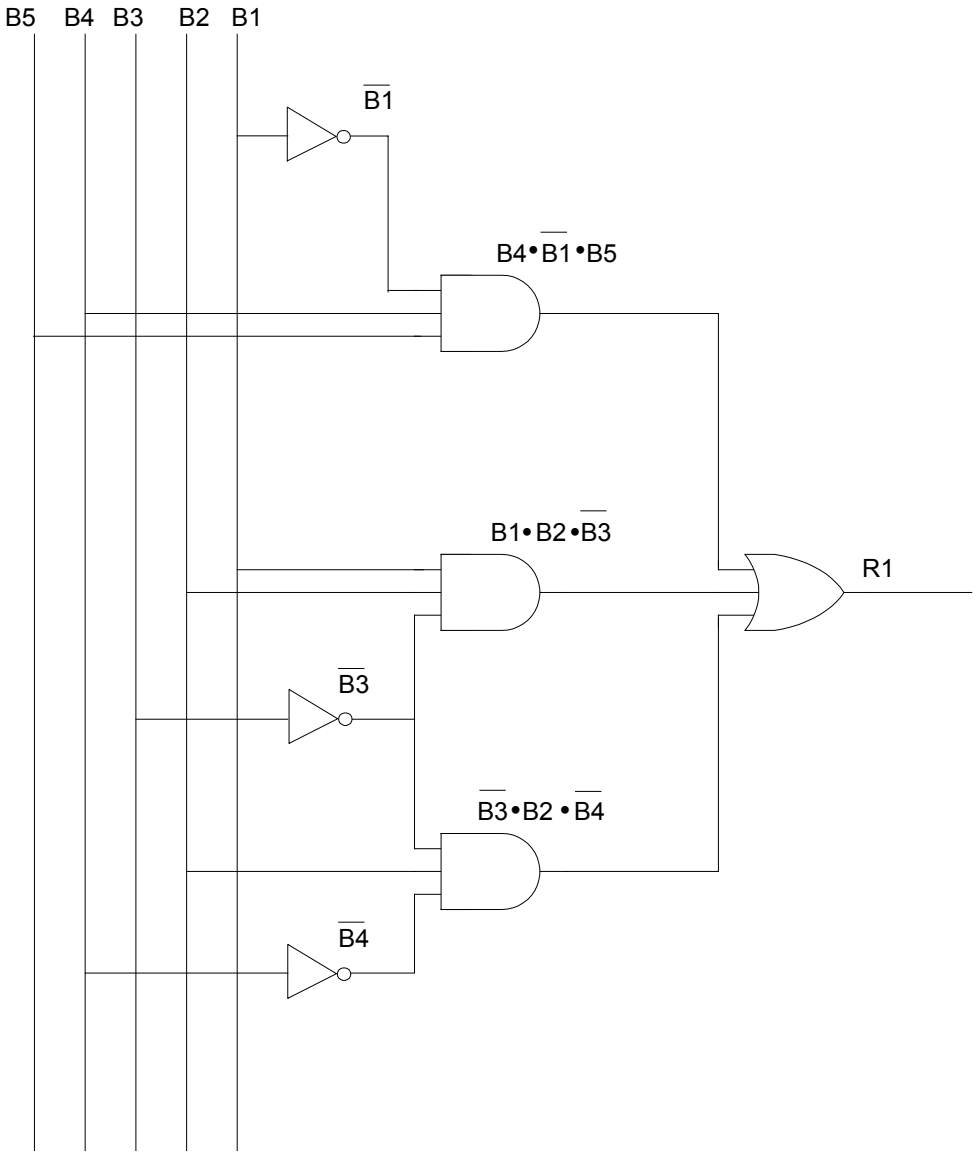
- *Nombre d'opérateurs NON : 4;*
- *Nombre d'opérateurs ET : 3 à 3 entrées;*
- *Nombre d'opérateurs OU : 1 à 3 entrées.*

$$\Longrightarrow \quad R_1 = \underbrace{\left(B_1 \bullet B_2 \bullet \overline{B_3}\right)}_{ET} + \underbrace{\left(B_4 \bullet \overline{B_1} \bullet B_5\right)}_{ET} + \underbrace{\left(\overline{B_4} \bullet B_2 \bullet \overline{B_3}\right)}_{ET} \quad \text{(OU)}$$

$\overline{B_3}$ *(NON)*

L'équation comporte deux fois $\overline{B_3}$ *; pour les obtenir, il suffit d'utiliser un seul opérateur NON, d'où 3 opérateurs NON au lieu de 4.*

Schéma logique :



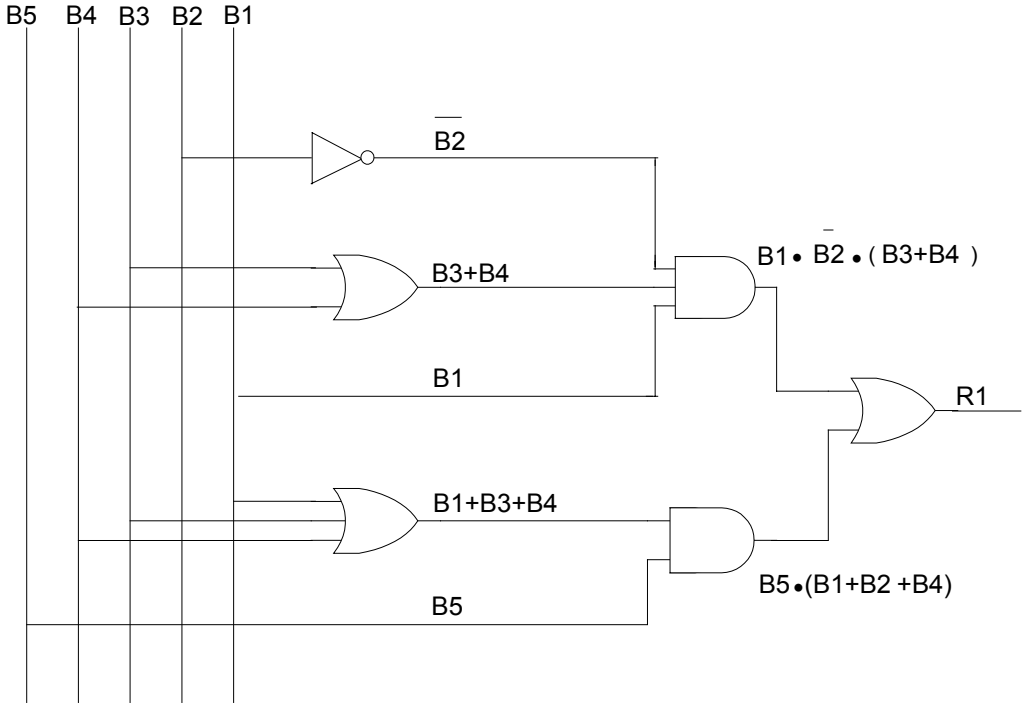
b) $R_1 = B_1 \bullet \overline{B_2} \bullet (B_3 + B_4) + B_5 \bullet (B_3 + B_4 + B_1)$

- *Nombre d'opérateurs NON : 1;*
- *Nombre d'opérateurs OU : 3;*
- *Nombre d'opérateurs ET : 2.*

$\Longrightarrow$ $R_1 = B_1 \bullet \overline{B_2} \bullet (B_3 + B_4) + B_5 \bullet (B_3 + B_4 + B_1)$

NON, OU, OU, ET, ET, OU

Schéma logique



b) Schéma logique ne comprenant que des opérateurs NON ET

Pour réaliser ce schéma, il faut les deux conditions suivantes :

- *L'équation ne doit comporter que des ET logiques ⟹ transformer l'équation en appliquant les théorèmes De Morgan.*
- *L'équation doit être entièrement recouverte par une barre ⟹ utiliser les propriétés de la négation* $S=\overline{\overline{S}}$.

Exemples :

a) $R_1=B_1 \bullet B_2 \bullet \overline{B_3}+B_4 \bullet \overline{B_1} \bullet B_5+\overline{B_4} \bullet B_2 \bullet \overline{B_3}$

Transformation des OU logiques en ET logique en appliquant le théorème De Morgan :

$$R_1=\overline{\overline{R_1}}=\overline{\overline{B_1 \bullet B_2 \bullet \overline{B_3}+B_4 \bullet \overline{B_1} \bullet B_5+\overline{B_4} \bullet B_2 \bullet \overline{B_3}}}=\overline{\overline{B_1 \bullet B_2 \bullet \overline{B_3}} \; \overline{B_4 \bullet \overline{B_1} \bullet B_5} \; \overline{\overline{B_4} \bullet B_2 \bullet \overline{B_3}}}$$

Il y a huit barres ⟹ 8 opérateurs NON ET.

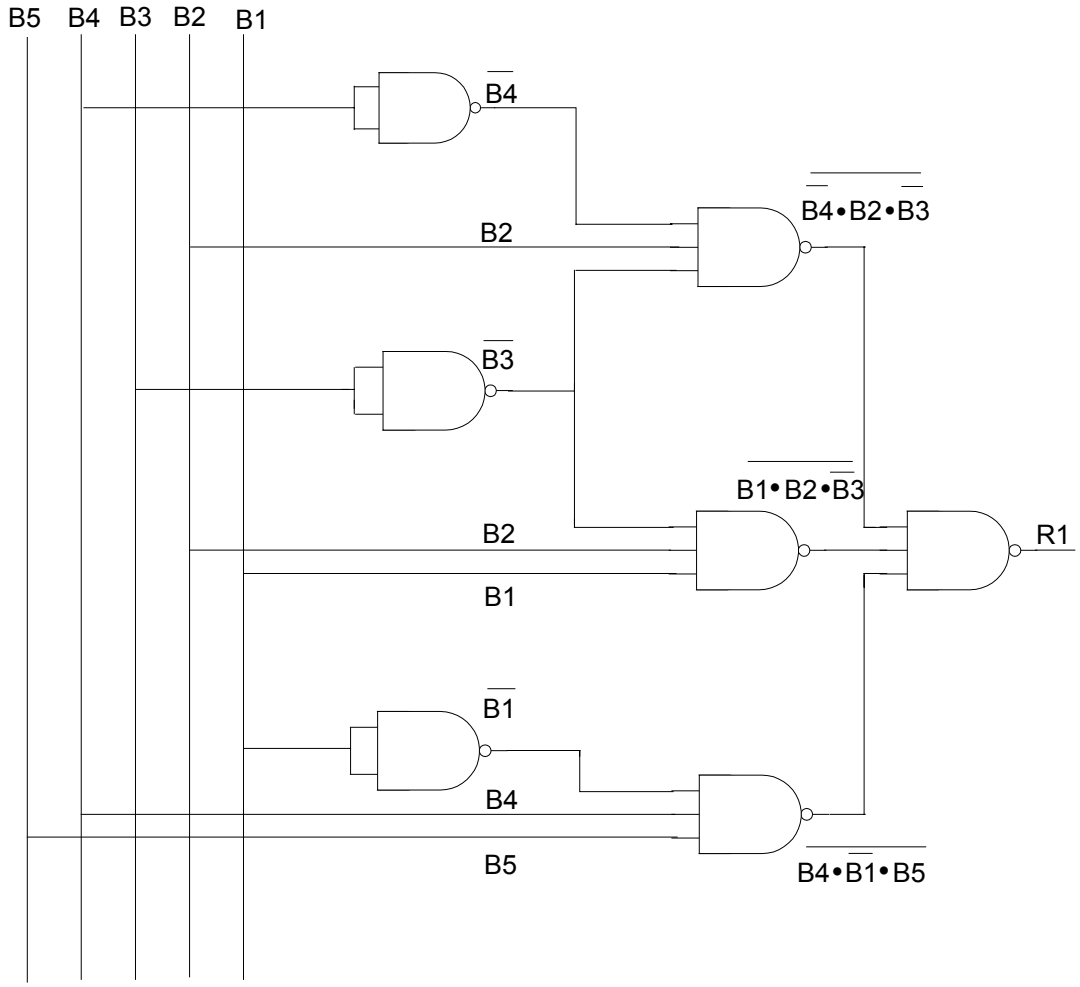
$$R_1=\overline{\overline{B_1 \bullet B_2 \bullet \overline{B_3}} \; \overline{B_4 \bullet \overline{B_1} \bullet B_5} \; \overline{\overline{B_4} \bullet B_2 \bullet \overline{B_3}}}$$

NON ET *NON ET* *NON ET* *NON ET*

NON ET *NON ET* *NON ET*

NON ET

Schéma logique



b) $R_1 = B_1 \bullet \overline{B_2} \bullet (B_3 + B_4) + B_5 \bullet (B_3 + B_4 + B_1)$

Transformation :

$$R_1 = B_1 \bullet \overline{B_2} \bullet \left(\overline{\overline{B_3 + B_4}}\right) + B_5 \bullet \left(\overline{\overline{B_3 + B_4 + B_1}}\right)$$

$$R_1 = B_1 \bullet \overline{B_2} \bullet \left(\overline{\overline{B_3} \bullet \overline{B_4}}\right) + B_5 \bullet \left(\overline{\overline{B_3} \bullet \overline{B_4} \bullet \overline{B_1}}\right)$$

$$\overline{\overline{R_1}} = R_1 = B_1 \bullet \overline{B_2} \bullet \left(\overline{\overline{B_3} \bullet \overline{B_4}}\right) + B_5 \bullet \left(\overline{\overline{B_3} \bullet \overline{B_4} \bullet \overline{B_1}}\right)$$

$$R_1 = \overline{B_1 \bullet \overline{B_2} \left(\overline{\overline{B_3} \bullet \overline{B_4}}\right) \bullet B_5 \bullet \left(\overline{\overline{B_3} \bullet \overline{B_4} \bullet \overline{B_1}}\right)}$$

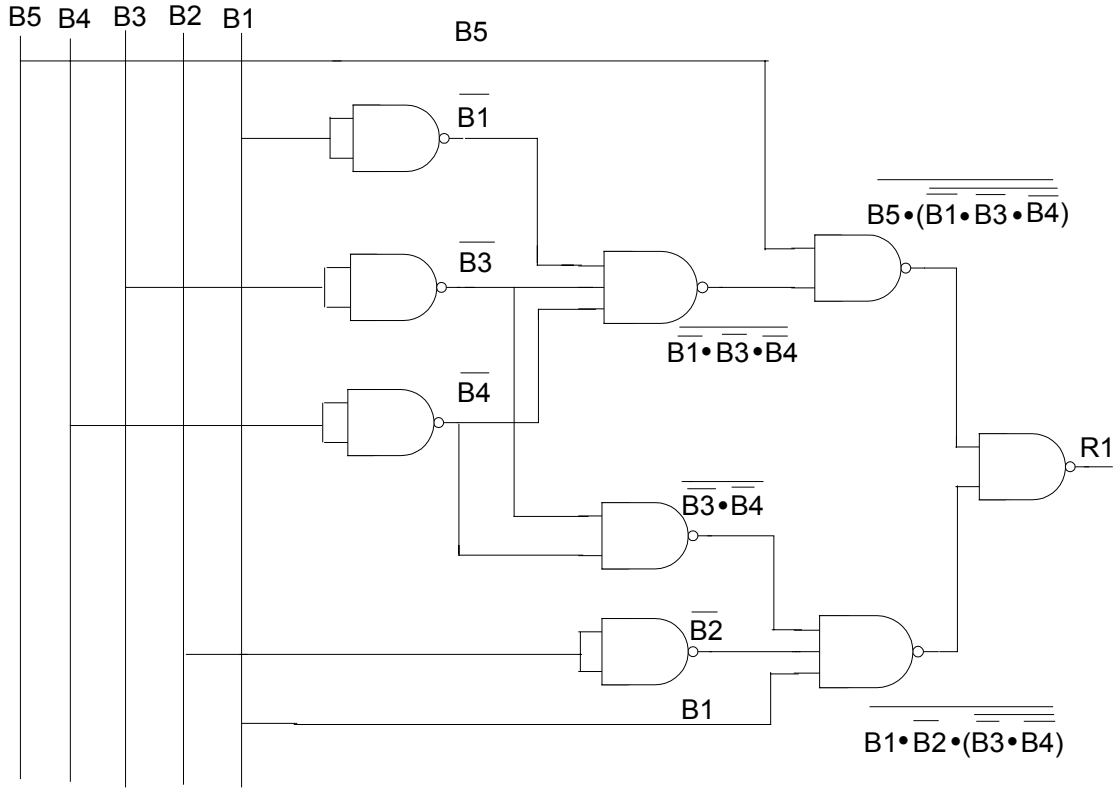
NON ET NON ET NON ET NON ET NON ET NON ET

NON ET *NON ET*

NON ET *NON ET*

NON ET

Schéma logique



c) Schéma logique ne comprenant que des opérateurs NON OU

Pour réaliser ce schéma il faut les deux conditions suivantes :

- *l'équation ne doit comporter que des ou logiques ⇒ transformer l'équation en appliquant les théorèmes de Morgan.*
- *l'équation doit être entièrement recouverte par une barre ⇒ utiliser les propriétés de la négation :* $S=\overline{\overline{S}}$

Exemples :

a) $R_1=B_1 \bullet B_2 \bullet \overline{B_3}+B_4 \bullet \overline{B_1} \bullet B_5+\overline{B_5} \bullet B_2 \bullet \overline{B_3}$

$$\overline{\overline{R_1}}=\overline{\overline{B_1 \bullet B_2 \bullet \overline{B_3}}}+\overline{\overline{B_4 \bullet \overline{B_1} \bullet B_5}}+\overline{\overline{\overline{B_5} \bullet B_2 \bullet \overline{B_3}}}$$

$$R_1=\overline{\overline{B_1}+\overline{B_2}+B_3}+\overline{\overline{B_4}+B_1+\overline{B_5}}+\overline{B_4+\overline{B_2}+B_3}$$

$$R_1=\overline{\overline{R_1}}=\overline{\overline{B_1}+\overline{B_2}+B_3}+\overline{\overline{B_4}+B_1+\overline{B_5}}+\overline{B_4+\overline{B_2}+B_3}$$

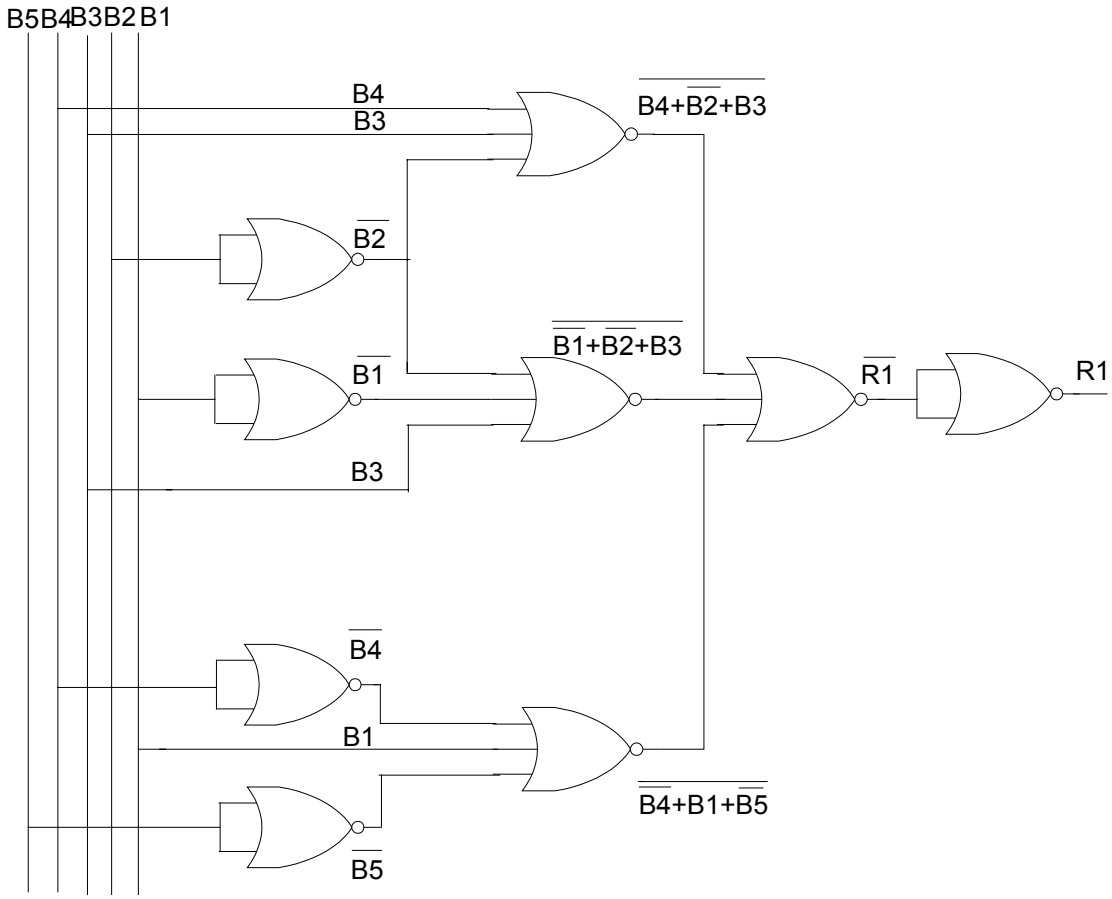
$\underbrace{\quad}_{NON\ OU}$ $\underbrace{\quad}_{NON\ OU}$ $\underbrace{\quad}_{NON\ OU}$ $\underbrace{\quad}_{NON\ OU}$ $\underbrace{\quad}_{NON\ OU}$

$\underbrace{\quad}_{NON\ OU}$ $\underbrace{\quad}_{NON\ OU}$ $\underbrace{\quad}_{NON\ OU}$

$\underbrace{\quad}_{NON\ OU}$

$\underbrace{\quad}_{NON\ OU}$

Schéma logique



b) $R_1 = B_1 \bullet \overline{B_2} \bullet (B_3 + B_4) + B_5 \bullet (B_3 + B_4 + B_1)$

Transformation :

$$R_1 = \overline{\overline{B_1 \bullet \overline{B_2} \bullet (B_3 + B_4)} + \overline{B_5 \bullet (B_3 + B_4 + B_1)}}$$

$$R_1 = \overline{\overline{B_1} + B_2 + \left(\overline{B_3 + B_4}\right) + \overline{B_5} + \left(\overline{B_3 + B_4 + B_1}\right)}$$

$$R_1 = \overline{\overline{R_1}} = \overline{\overline{B_1} + B_2 + \left(\overline{B_3 + B_4}\right) + \overline{B_5} + \left(\overline{B_3 + B_4 + B_1}\right)}$$

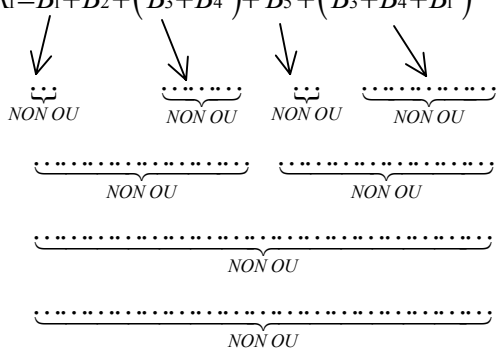
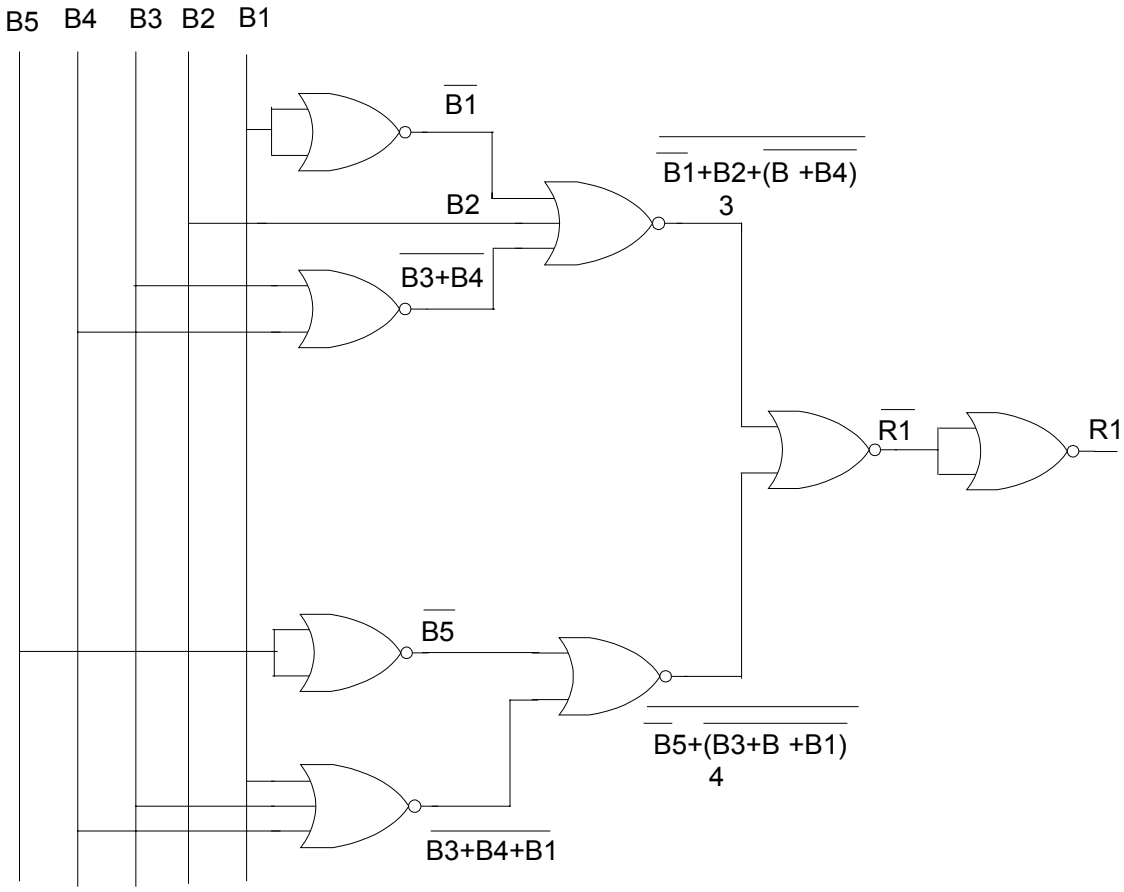


Schéma logique



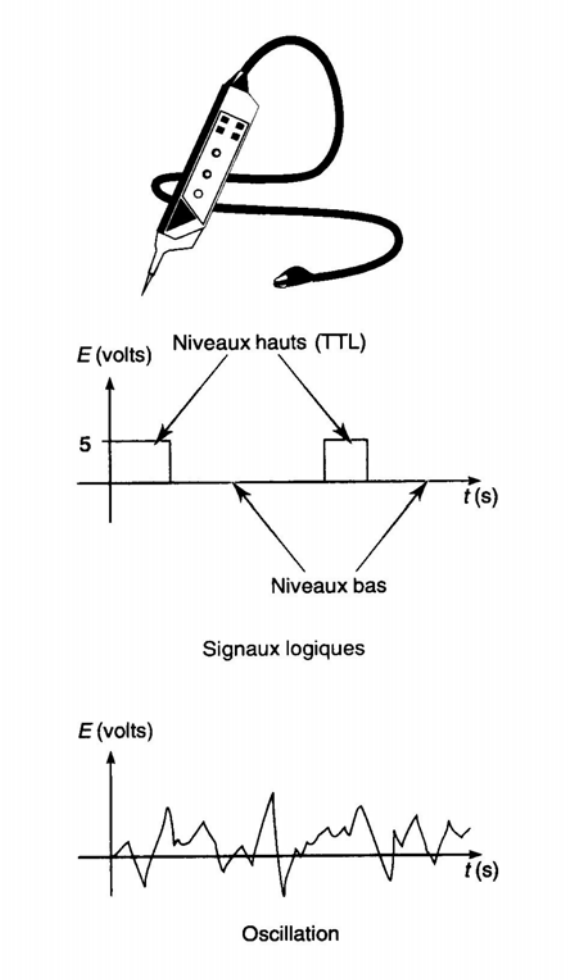
VIII. Utiliser une sonde logique.

a) Généralités

La sonde logique est un appareil de vérification pour les circuits intégrés à portes logiques.
Elle permet de déterminer le niveau (haut ou bas) des entrées et des sorties simplement et rapidement.
La sonde contient généralement deux DEL (c-à-d diodes électroluminescentes) qui s'allument suivant le niveau détecté. Une DEL s'allumera donc au niveau haut (high), tandis que l'autre indiquera le niveau bas (low), certains sondes comportent une troisième DEL pour détecter les trains d'impulsions (Pulse).

b) Utilisations

Pour utiliser cet appareil (voir figure suivante), il suffit de placer la pointe directement sur le point de contact qu'on désire mesurer. Il est à noter qu'on ne doit pas laisser traîner la pointe de la sonde sur le circuit, car cela crée des niveaux de tension qui peuvent endommager les circuits intégrés CMOS non protégés ou produire de mauvais signaux.
Il est à noter que l'appareil ne peut détecter des signaux qui changent rapidement (oscillations



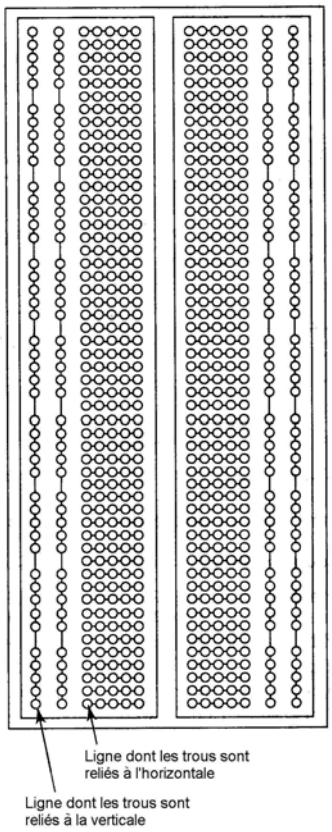
c) Les modes TTL, CMOS :

Les niveaux logiques varient selon la famille de C.I. employé. On devra donc décider du mode d'utilisation de la sonde (TTL ou CMOS) avant de prendre des mesures. L'utilisation d'une sonde TTL sur un circuit CMOS aurait pour effet d'endommager la sonde, alors que dans le cas contraire, la sonde ne pourrait détecter le niveau haut.

IX. Montage des circuits

IX.1 Plaque de montage

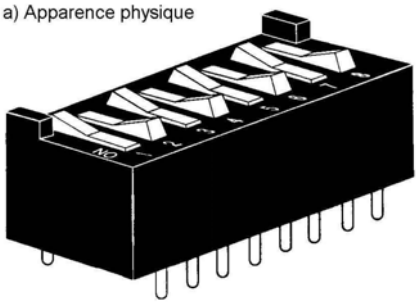
(voir figure suivante)
Les plaques de montage perforées sont spécialement conçues pour les montages temporaires. Elles sont idéales pour expérimenter certains circuits logiques utilisant des circuits intégrés TTL ou C-MOS.
Cette plaque possède des trous perforés interreliés qui forment des lignes indépendantes. Les lignes verticales sont généralement employées pour les sources (Vcc pour les circuits intégrées TTL et V_{DD}-Vss pour les C-I.C-MOS), tandis que les lignes horizontales servent au branchement des divers composants.



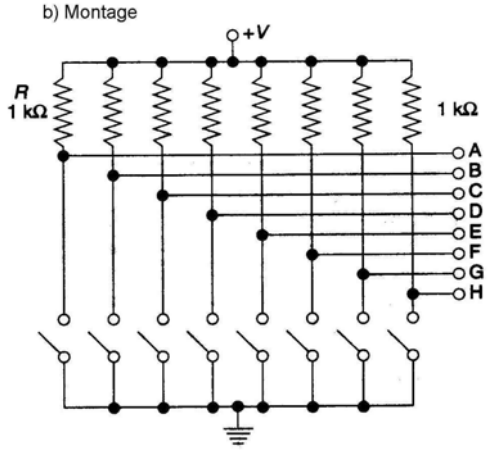
IX.2 Interrupteurs logiques

La simulation des entrées se fait à l'aide d'un groupe d'interrupteurs miniatures (DIP suitch) (voir figure suivante). Ce type de composants permet de faire parvenir des signaux hauts (1) ou bas (0) à l'entrée des portes logiques du circuit.

a) Apparence physique



b) Montage



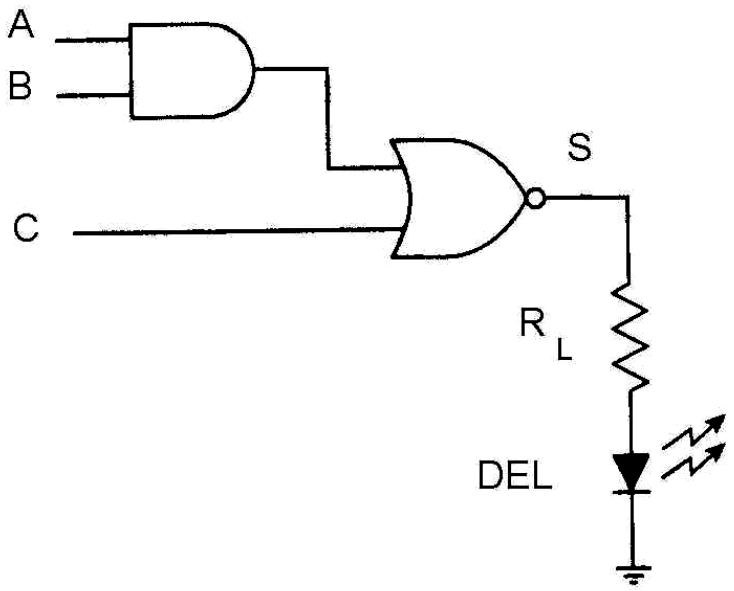
IX.3 Choix de logique positive ou négative, visualisation des sorties

La forme que prend la sortie d'un circuit permet de faire la différence entre les logiques positive et négative.

a) Logique positive

Un circuit est en logique positive lorsque la sortie est branchée à la masse ce qui donne directement la valeur de la sortie.
Cette valeur peut être visualisée par une DEL (diode électroluminescente) qui s'allume en présence d'un niveau haut (1) et s'éteint pour un niveau bas (0).

EXEMPLE :

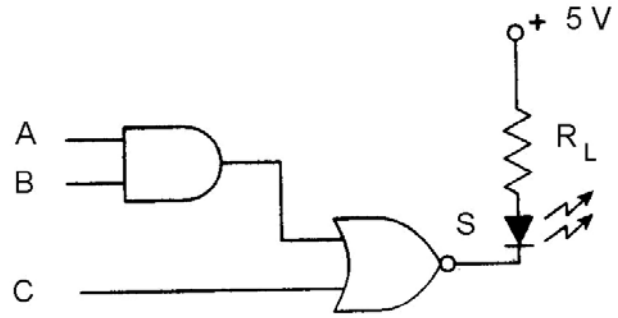


Mais dans ce cas la sortie débite un courant de charge élevé ⟹ logique peu recommandée

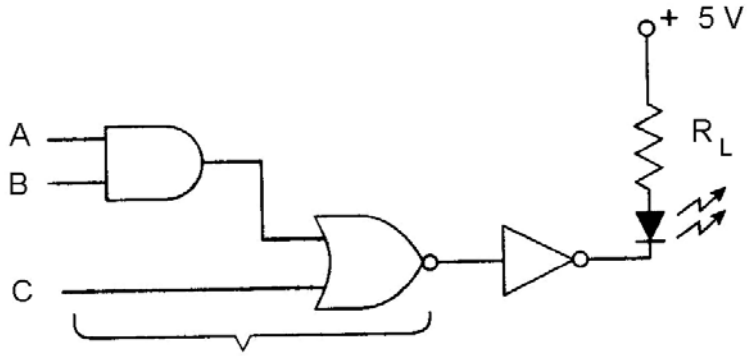
b) Logique négative

Un circuit est en logique négative lorsque la sortie est reliée à la source (Vcc) ce qui donne la valeur de la sortie inversée qu'on peut rétablir en utilisant un inverseur.

EXEMPLE :



Solution avec sortie inversée



Solution avec sortie rétablie

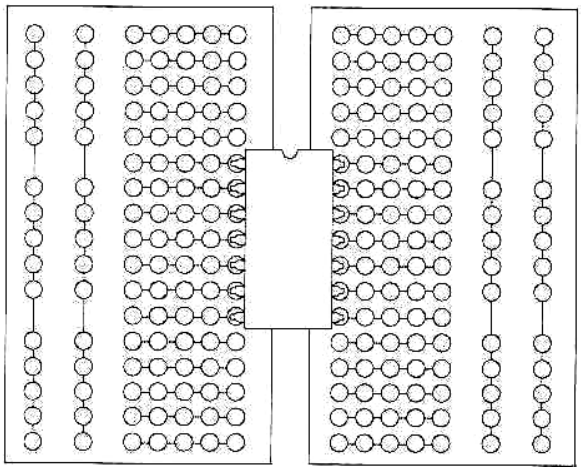
La résistance R_L limite le courant qui passe par la DEL afin d'éviter d'endommager le circuit :

$$R_{L\,\min} = \frac{Tension\ aux\ bornes\ de\ R_L}{Courant\ \max imal\ des\ portes\ \log iques} = \frac{5V - 1{,}5V}{16\,mA} = 220\Omega$$

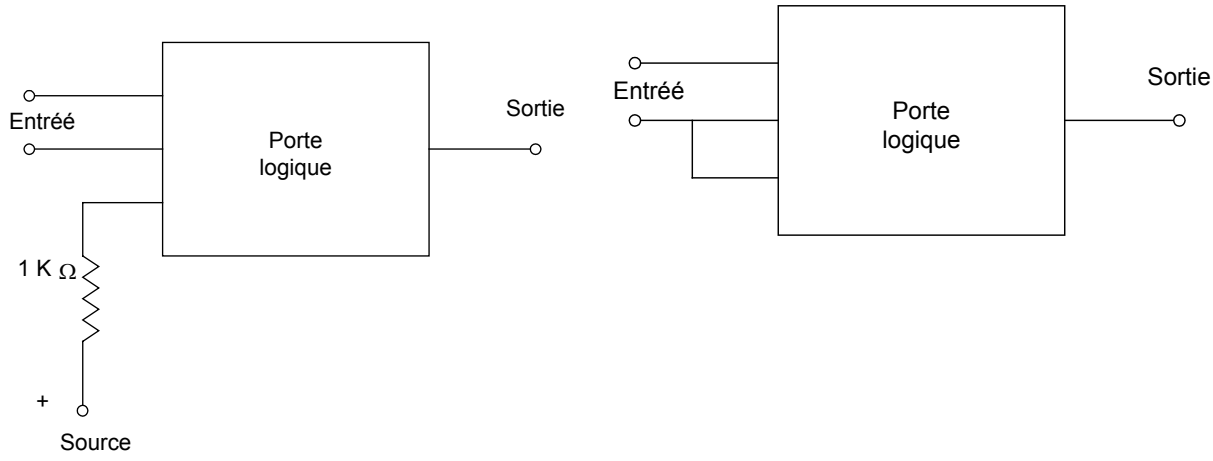
1,5 V = ΔU aux bornes de la DEL.

IX.4 Technique de travail

- *Lorsqu'on utilise des C.I. (circuits intégrées) , on doit les placer de façon qu'il n'y ait aucun contact entre les pattes. La façon de faire consiste à les placer entre les deux groupes de lignes horizontales comme sur la figure suivante.*



- *Les entrées flottantes (libres, non utilisé) ne sont pas recommandées. On peut soit les brancher à la source en passant par une résistance de 1KΩ (le meilleur choix), on les branche ensemble pour n'en faire qu'une (voir figure suivante).*



Branchement à la source (assure un niveau 1)

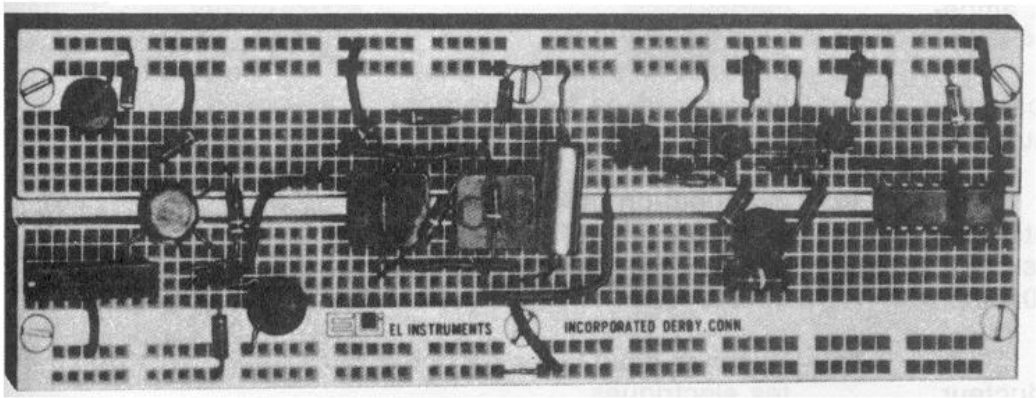
Branchement à une autre entrée (copie le signal d'entrée)

Comment procéder pour monter un circuit ?

- *Sur une feuille simulant la plaque perforée, trouver le meilleur croquis de disposition possible tout en respectant les techniques de montage des composants et en numérotant les différentes bornes des composants.*

- *Monter les composants sur la plaque perforée en respectant le croquis de disposition finale.*

- *Réaliser les branchements entre les composants tout en respectant les techniques de travail avec une facilité de lire le brochage des C.I et une bonne esthétique.*



IX.5 Caractéristiques des circuits intégrés :

Lorsqu'on travaille avec des circuits logiques deux états sont considérés : l'état haut et l'état bas. Ces deux états sont définis par des plages de tensions en fonction de la technologie utilisée. En logique positive l'état haut correspond à une présence de tension et à un « 1 » logique.

Une famille logique est caractérisée par ses paramètres électriques :

- *la plage des tensions d'alimentation et la tolérance admise sur cette valeur,*
- *la plage des tensions associée à un niveau logique, en entrée ou en sortie,*
- *les courants pour chaque niveau logique, en entrée ou en sortie,*
- *le courant maximum que l'on peut extraire d'une porte logique et le courant absorbé en entrée,*
- *La puissance maximale consommée*

Les performances dynamiques principales sont :

- *les temps de montée (transition bas–haut) et de descente (transition haut–bas) des signaux en sortie d'une porte,*
- *Les temps de propagation d'un signal entre l'entrée et la sortie d'une porte logique.*

a) Paramètres caractéristiques :

V_{CC} - tension d'alimentation : *niveau de tension nécessaire pour alimenter le circuit. TTL (+5V); V_{DD}, et Vss = alimentation des circuits CMOS (généralement Vss = 0, V_{DD} = 5,ou 10 ou 15 V);*

V_{IH} (min) - tension d'entrée niveau HAUT : *niveau de tension nécessaire pour avoir un 1 logique en entrée.*

V_{IL} (max) - tension d'entrée niveau BAS : *niveau de tension nécessaire pour avoir un 0 logique en entrée.*

V_{OH} (min) - tension de sortie niveau HAUT : *niveau de tension de la sortie d'un circuit logique correspondant à l'état logique 1.*

V_{OL} (max) - tension de sortie niveau BAS : *niveau de tension de la sortie d'un circuit logique correspondant à l'état logique 0.*

I_{IH} - courant d'entrée niveau HAUT : *le courant qui traverse une borne d'entrée quand une tension niveau haut est appliquée à cette entrée.*

I_{IL} - courant d'entrée niveau BAS : *le courant qui traverse une borne d'entrée quand une tension niveau bas est appliquée à cette entrée.*

I_{OH} - courant de sortie niveau HAUT : *le courant qui traverse une borne de sortie placée au niveau logique 1 dans des conditions de charge spécifiées.*

I_{OL} - courant de sortie niveau BAS : *le courant qui traverse une borne de sortie placée au niveau logique 0 dans des conditions de charge spécifiées.*

- ***Valeurs des tensions des circuits TTL***

	Tension
V_{IL}	0,8 V(max)
V_{IH}	2 V (min)
V_{OL}	0,4 V (max)
V_{OH}	2,4 V(min)

- ***Valeurs des tensions des circuits (CMOS)***

	V_{DD} (5V)	V_{DD} (10V)	V_{DD} (15V)
V_{IL}	1,5 V	3,0 V	4,5 V
V_{IH}	3,5 V	7,0 V	10,5 V
V_{OL}	0,05 V	0,05 V	0,05 V
V_{OH}	4,95 V	9,95 V	14,95 V

On constate que :

$V_{IL} = 0,3\ V_{DD}$
$V_{IH} = 0,7\ V_{DD}$
$V_{OL} = 0,05\ V$
$V_{OH} = V_{DD} - 0,05\ V$

b) Sortance :

Normalement, la sortie d'un circuit logique doit pouvoir piloter plusieurs entrées logiques. La sortance (appelée également facteur de charge) est définie comme le nombre maximal d'entrées logiques standards qui peuvent être pilotées sans problèmes par une sortie. Par exemple, quand il est indiqué qu'une porte logique a une sortance de 10, cela signifie qu'elle peut piloter 10 entrées logiques standards. Si on dépasse ce nombre, il n'est pas assuré que les tensions des niveaux logiques des sorties seront exactes.

c) Retards de propagation :

Un signal logique qui traverse un circuit subit toujours un retard. Deux retards de propagation sont définis :

- t_{PLH} *: retard pour passer du niveau logique 0 au niveau logique 1.*
- t_{PHL} *: retard pour passer du niveau logique 1 au niveau logique 0.*

X. Circuits de base

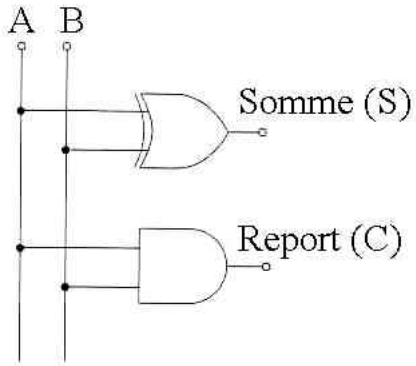
X.1 Additionneur

C'est un circuit logique qui permet d'effectuer l'addition des nombres binaires.

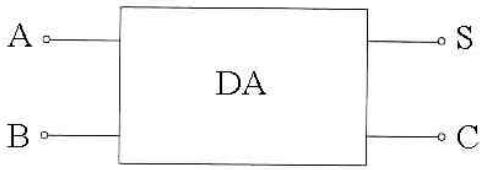
a) Demi-Additionneur

Il ne peut additionner que deux bits ou entrées. On l'identifie par DA (voir figure suivante).

- *Schéma logique*



- *Représentation simplifiée*



- *Table de vérité*

Entrées		Sorties	
A	**B**	**S**	**C**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Le montage possède 2 sorties

- *S* ⟹ *donne la somme* $(\Sigma)=A\oplus B$
- *C* ⟹ *donne le report* $(C)=A\bullet B$

Exemples d'addition :

```
  ° 1
+ 0
  0 1
```

Report (C) *Somme(∑)*

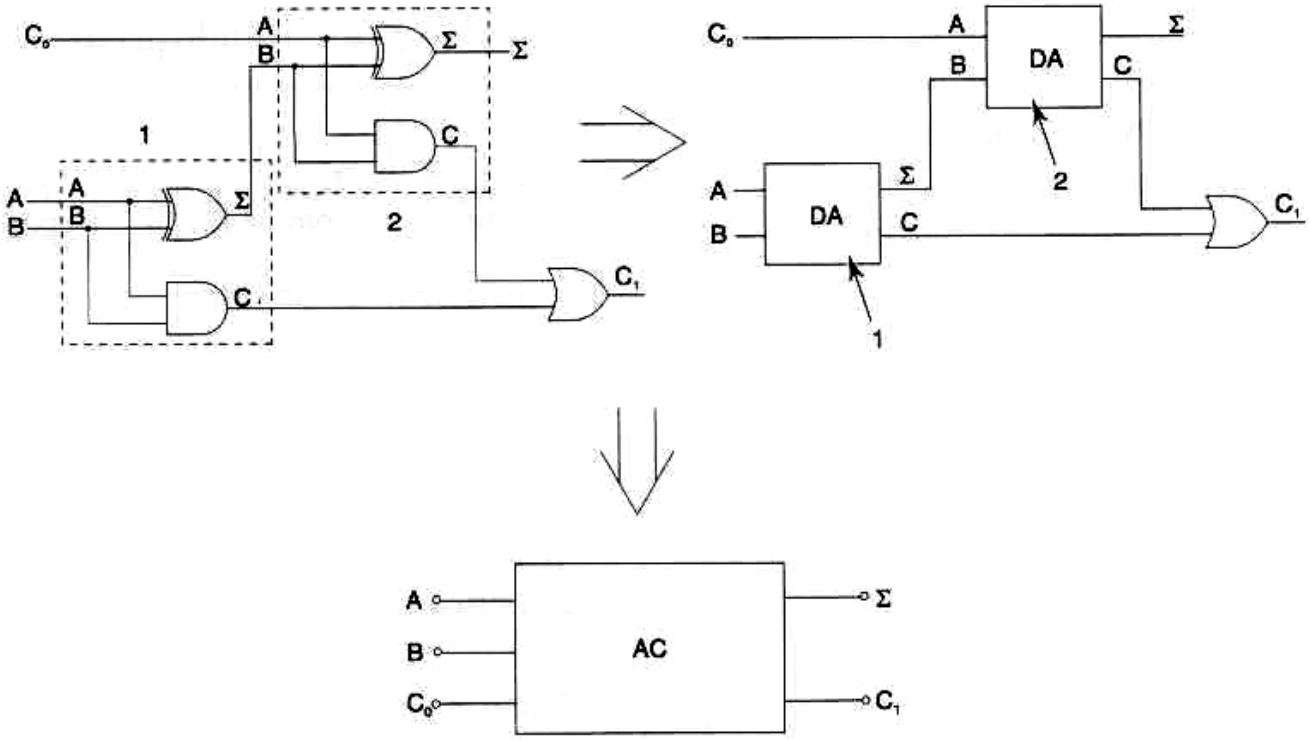
```
  1 1
+ 1
  10
```

Report (C) *Somme(∑)*

Pour pouvoir additionner de plus grands nombres binaires, on doit combiner plusieurs DA pour former des additionneurs complets, qu'on identifie par AC.

b) Additionneurs complets

C'est une évolution du DA, il possède 3 entées (voir figure suivante).

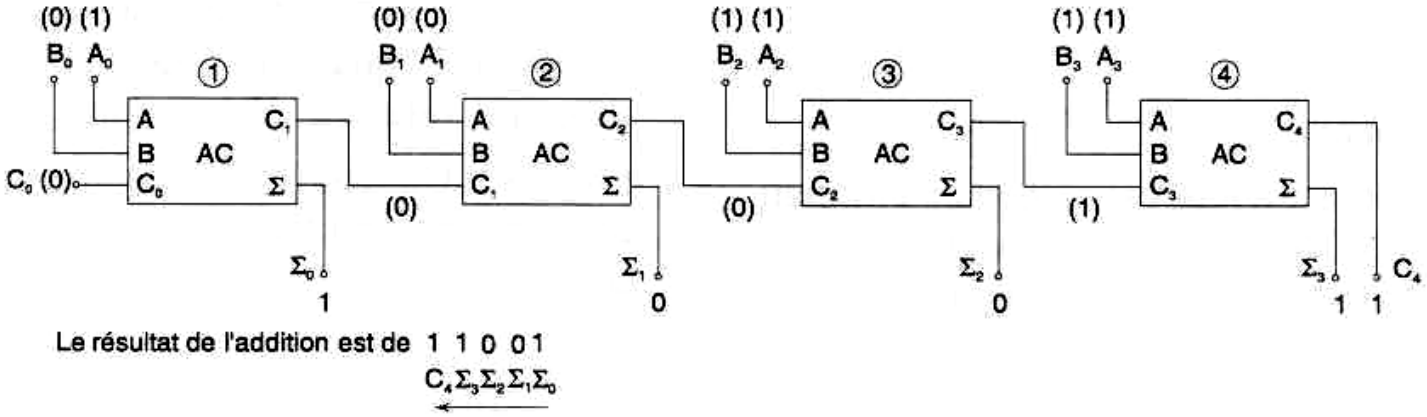


- *Table de vérité :*

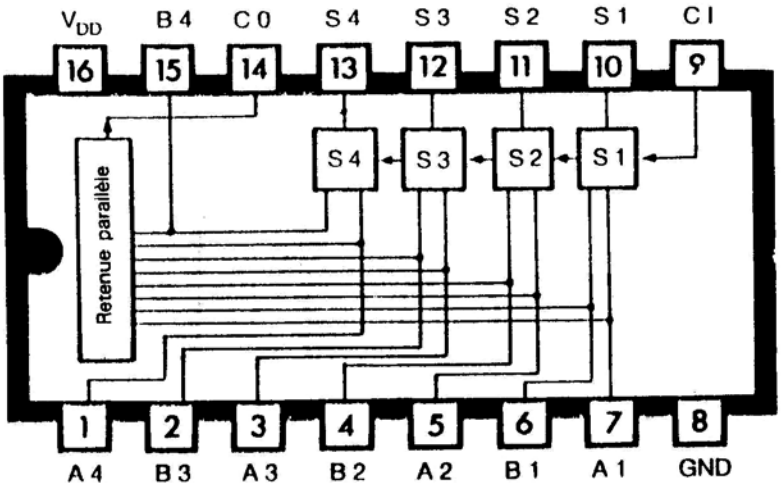
Entrées			Sorties	
A	**B**	**Co**	**Σ**	**C_1**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Il est possible de refaire le même travail en combinant plusieurs AC et en utilisant le report du premier étage en guise d'entrée «C» du second et ainsi de suite. La figure suivante traite l'addition de deux nombres de 4 bits en utilisant des AC.

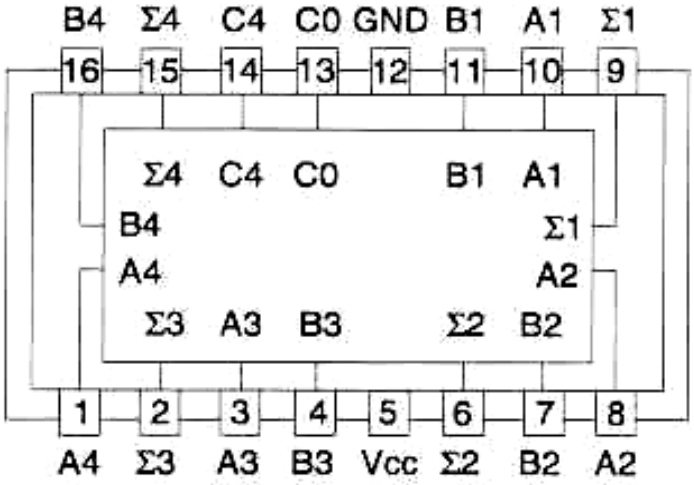
A = 1101
B = 1100



Les familles CMOS et TTL comportent des additionneurs sous la forme des C.I pour éviter de manipuler une quantité énorme de portes logiques (voir figure suivante).



Additionneur complet 4 bits CMOS (4008)



Additionneur complet 4 bits TTL (7483A)

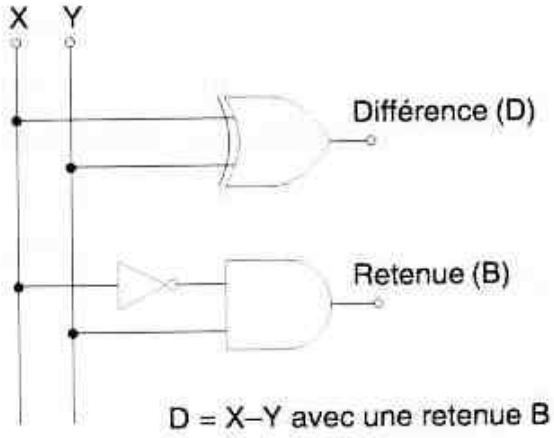
X.2 Soustracteurs logiques

Ils ne sont pas très utilisés dans les domaines industriels. Leur fonctionnement ressemble énormément à celui des additionneurs.

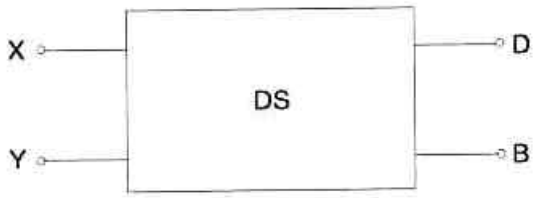
a) Demi-soustracteur

Il est obtenu en ajoutant une porte «NON» à un demi-additionneur.
On l'identifie par les lettres DS (voir figure suivante).

- Schéma logique



- Représentation simplifiée



- *Table de vérité*

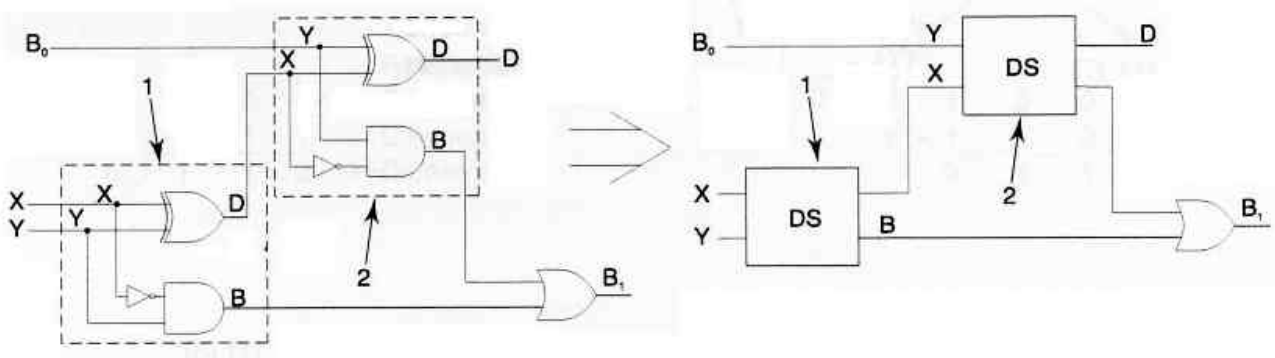
Entrées		Sorties	
X	**Y**	**D**	**B**
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

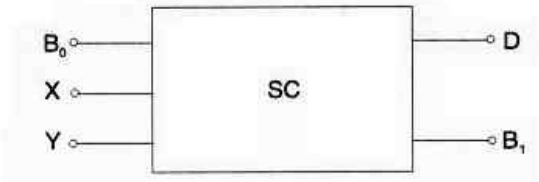
On a deux sorties : → *D qu'on pourrait représenter par* $D=X\oplus Y$

→ *B qu'on pourrait représenter par* $B=\overline{X}\bullet Y$

b) Soustracteurs complets

Le soustracteur complet est une combinaison de deux DS auquel on a ajouté un «OU» logique à la sortie de la retenue (voir figure suivante).

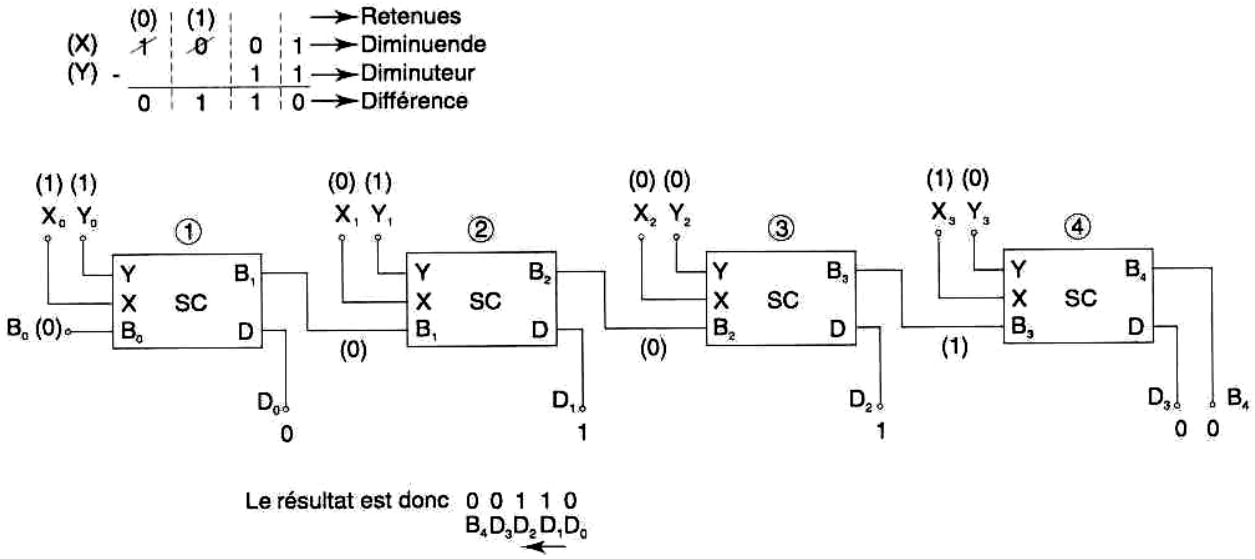




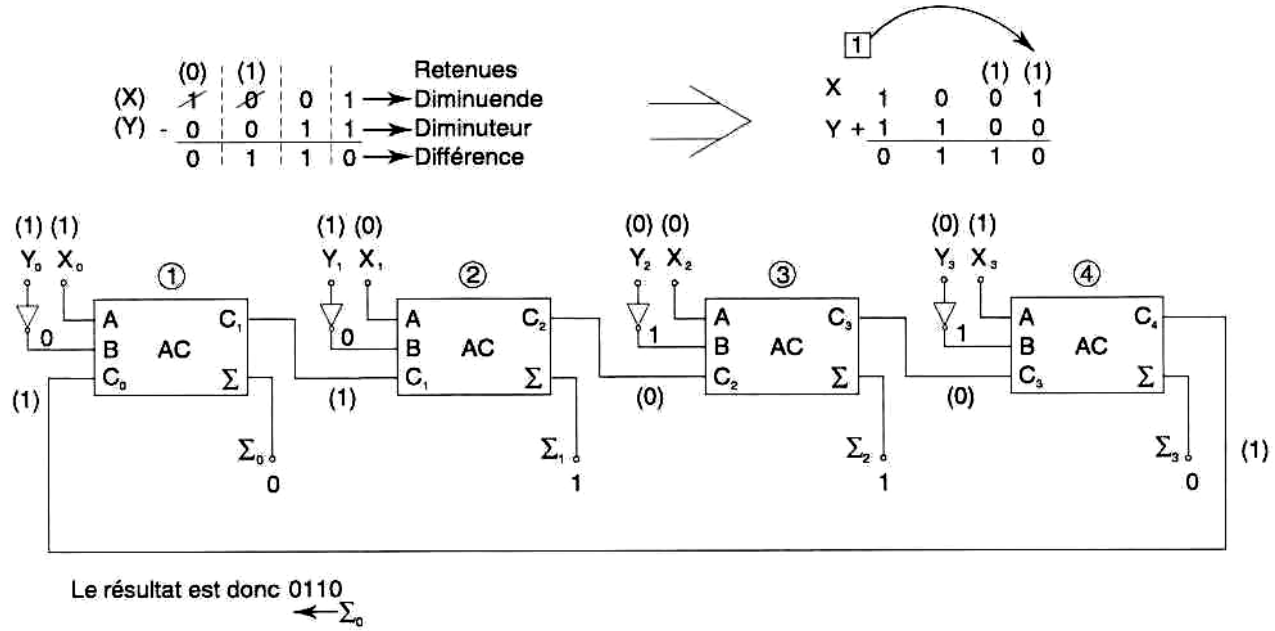
- *Table de vérité*

Entrées			Sorties	
X	**Y**	**Bo**	**D**	**B1**
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

Il est possible de refaire le même travail en combinant plusieurs SC en cascade pour réaliser une soustraction binaire de 4 bits ou plus (voir figure suivante).



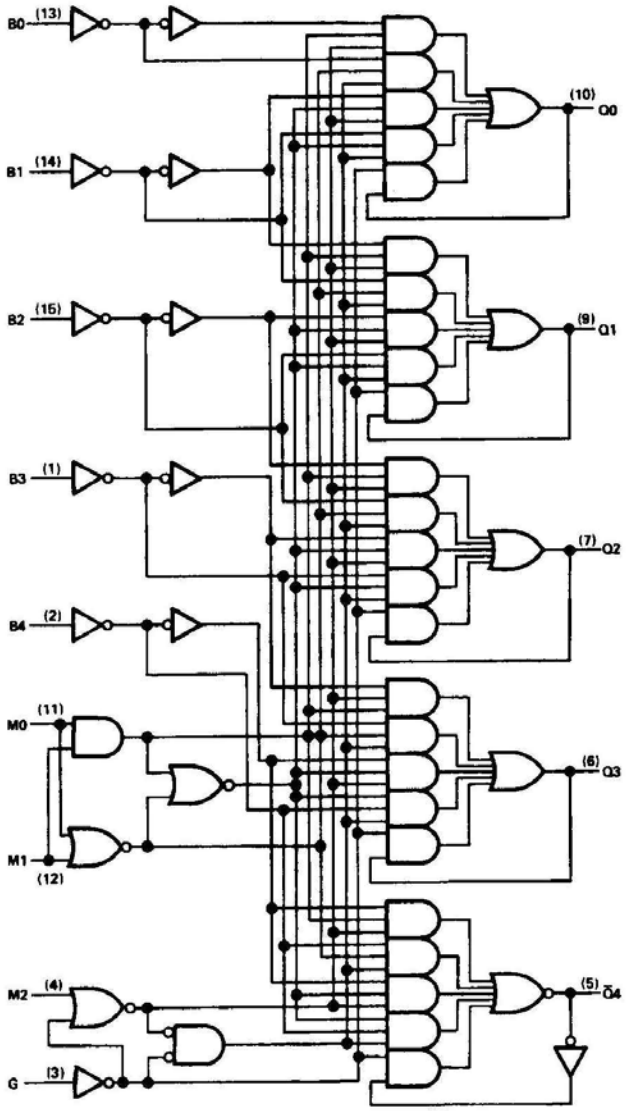
Malheureusement, on ne trouve pas des C.I. pour les soustracteurs. Il est cependant possible d'utiliser un artifice mathématique pou transformer la soustraction en addition et ainsi utiliser des additionneurs logiques (voir figure suivante).

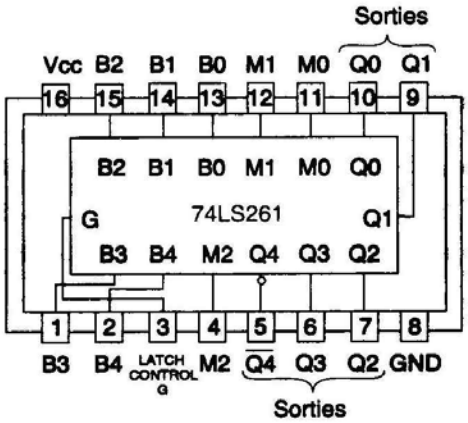


Il est possible de se servir, par exemple d'un C.I TTL 7483 pour monter un circuit soustracteur.

X.3 Multiplicateur

Exemple de multiplicateur de 2 bits par 4 bits (voir figure suivante).





Le montage à partir des portes logiques de base serait très difficile et coûteux. On utilise donc des C.I qui contiennent tous les composants nécessaires à la réalisation de l'opération.

Voici un tableau qui énumérer les divers C.I qui servent à la multiplication.

	Fonction	Numéro
TTL	2 bits par 4 bits	74LS261
	4 bits par 4 bits	74284, 74285, 74S274
CMOS	4 bits	40181, 4057
	Multiplicateur BCD	4527
	Multiplicateur binaire	4089

Remarque :

Il existe plusieurs autres circuits logiques de base tels que :

- *Le décodeur : c'est un circuit permettant de convertir pour chacune des combinaisons possibles d'entrées, une sortie possible.*
- *Le codeur : c'est un circuit logique permettant de convertir un code quelconque en un autre code.*
- *Le comparateur : c'est un circuit permettant de comparer deux nombres binaires à ses entrées pour ainsi désigner lequel des deux est le plus grand.*

MODULE N° 21: LOGIQUE COMBINATOIRE

GUIDE DES EXERCICES ET TRAVAUX PRATIQUES

Exercices :

Exercices 1 :

Réduire les fonctions suivantes en utilisant l'algèbre de Boole

$$F1 = (A\overline{B} + C)(A + \overline{B})C$$

$$F2 = C + AB + AD(B + \overline{C}) + CD$$

$$F3 = A\overline{B} + A\overline{C} + \overline{BC}$$

$$F4 = A + ABC + \overline{A}C$$

$$F5 = (\overline{X} + \overline{Y})(\overline{X} + \overline{Z})$$

Exercice2 :

1) *Donner l'équivalent décimal des nombres suivants :*

$(72)_8$ = (……)$_{10}$
$(1251)_8$ = (……)$_{10}$
$(17,3)_8$ = (……)$_{10}$
$(512,65)_8$ = (……)$_{10}$

2) *Donner l'équivalent BCD des nombres décimaux suivants :*

$(8)_{10}$ = (……)$_{BCD}$
$(17)_{10}$ = (……)$_{BCD}$
$(128)_{10}$ = (……)$_{BCD}$
$(92)_{10}$ = (……)$_{BCD}$

3) *Effectuer les opérations arithmétiques suivantes :*

0011	1101	110	1111/11
+	-	*	
1101	0010	11	

Exercice3:

Construire les tables de vérité des équations suivantes :

a) $S = A \bullet \overline{B}$

b) $S = \overline{A} + B$

c) $S = \overline{A} \bullet (B + C)$

d) $S = A + (\overline{B} \bullet \overline{C})$

Exercice 4:

Dessiner le schéma logique de la sortie F en utilisant seulement des portes NAND.

C	B	A	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Exercice 5 :

1) *Ecrire l'équation simplifiée des tableaux suivants :*

	00	01	11	10
00	0	1	1	0
01	0	0	1	1

	00	01	11	10
00	0	1	1	1
01	0	1	1	1
11	0	1	1	0
10	0	1	1	0

	00	01	11	10
00	1	1	0	0
01	0	1	0	0
11	1	1	0	0
10	1	0	0	1

	00	01	11	10
00	1	0	0	1
01	0	1	1	1
11	0	1	1	1
10	1	0	0	1

2) *Pour chacune des équations logiques suivantes :*

a) *Etablir le diagramme de Karnaugh. Effectuer les regroupements et écrire l'équation logique simplifiée correspondante.*
b) *Traduire ces équations en schémas logiques. comprenant des opérateurs NON, ET,OU :*
c) *Traduire ces équations en schémas logiques. ne comprenant que des opérateurs NON ET (NAND).*

- $S=\overline{A}\bullet\overline{B}\bullet\overline{C}+\overline{A}\bullet B\bullet\overline{C}+\overline{A}\bullet B\bullet C+A\bullet B\bullet C$
- $S=A\bullet\overline{B}+\overline{A}\bullet\overline{B}+A\bullet B$

Tp 1 - portes logiques fondamentales : et (and), ou (or). Non (not)

1. Objectifs visés

Démontrer le rapport existant entre les entrées et la sortie de portes ET (AND) et OU (OR) à deux entrées et NON (NOT) à une entrée, en construisant les tables de vérité.

2. Durée du TP

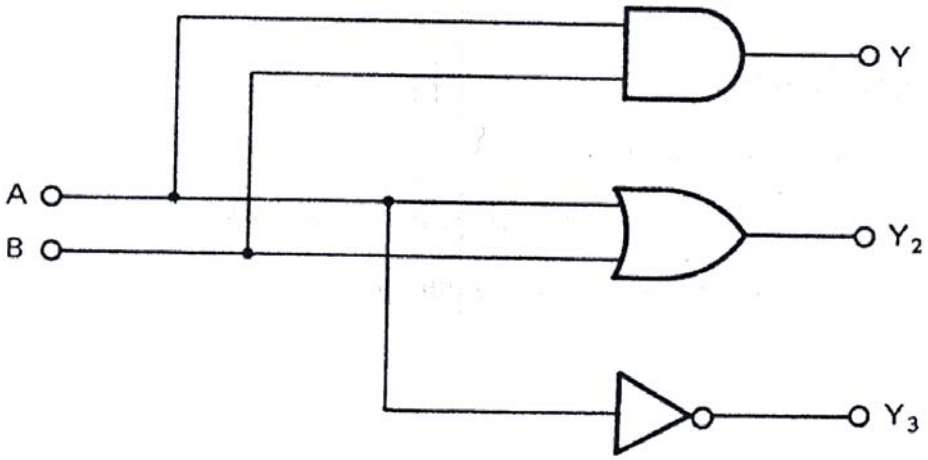
2 heures.

3. Matériels (Equipements et matière d'œuvre) par équipe

- *Alimentation continue (+ 5 V)*
- *Plaquette de montage ou simulateur de fonctions logiques*
- *Interrupteurs logiques*
- *Pinces et fils*
- *Circuits intégrés 7404, 7408, 7432.*
- *Notes de cours et fiches techniques*

4. Description du TP

Soit le circuit montré à la figure suivante :



Réaliser ce circuit

5. Déroulement du TP

1) *Se rappeler que l'on doit relier la broche 14 de chaque circuit intégré à la borne "+5V" de l'alimentation en cc, et la broche 7 à la borne "GND".*

2) *Relier les points A et B du montage aux interrupteurs de données "0" ou "1" suivant la table de vérité. Relier ensuite les sorties Y1, Y2 et Y3 aux bornes de trois LED.*

3) *Remplir la table de vérité suivante : La LED allumée indique que la sortie est au niveau logique 1; la LED.éteinte indique que la sortie est au niveau logique 0.*

A	B	Y_1	Y_2	Y_3
0	0			
0	1			
1	0			
1	1			

Tp 2 - portes logiques fondamentales : non-et (nand), non-ou (nor), ou exclusif (xor)

1. Objectifs visés

Démontrer le rapport existant entre les entrées et la sortie de portes NON - ET (NAND) et NON - OU (NOR) et OU EXCLUSIF (XOR) à 2 entrées, en construisant les tables de vérité.

2. Durée du TP

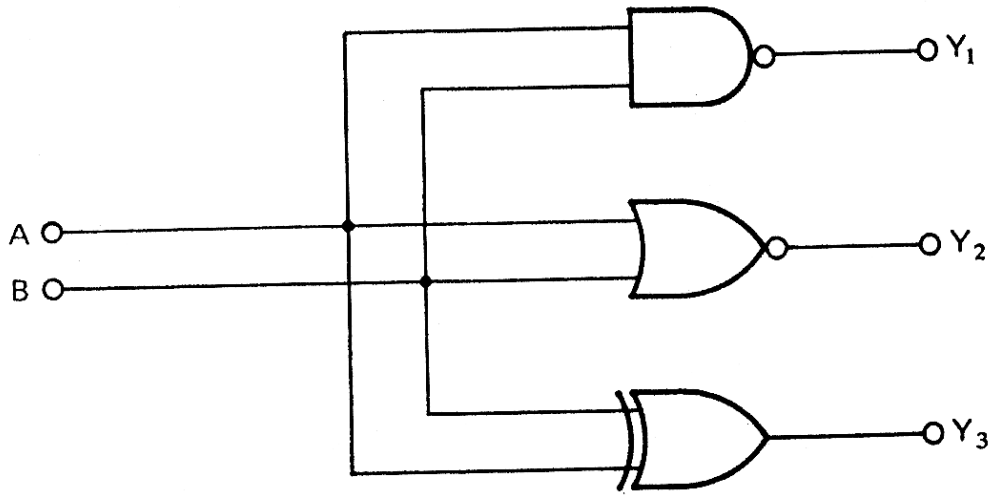
2 heures.

3. Matériels (Equipements et matière d'œuvre) par équipe.

- *Alimentation continue (+ 5 V)*
- *Plaquette de montage ou simulateur de fonctions logiques*
- *Interrupteurs logiques*
- *Pinces et fils*
- *Circuits intégrés 7400, 7402, 7486.*
- *Notes de cours et fiches techniques*

4. Description du TP

Soit le circuit montré à la figure suivante :



5. Déroulement du TP

1) *Réaliser le circuit et effectuer les connexions adéquates. Se rappeler que l'on doit relier la broche 14 du circuit intégré à la borne « + 5V » de l'alimentation et la broche 7 à la borne « GND ».*

2) *Relier les points A et B du montage aux interrupteurs de données "0" ou "1" suivant la table de vérité. Relier ensuite les sorties Y1, Y2 et Y3 aux bornes de trois LED.*

3) *Remplir la table de vérité suivante : La LED allumée indique que la sortie est au niveau logique 1; la LED.éteinte indique que la sortie est au niveau logique 0.*

A	B	Y_1	Y_2	Y_3
0	0			
0	1			
1	0			
1	1			

Tp 3 - applications de l'algèbre de Boole

1. Objectifs visés

1. Analyser les règles de l'algèbre de Boole.
2. Acquérir de l'expérience en travaillant avec des circuits pratiques.
3. Simplifier une fonction complexe en utilisant l'algèbre de Boole.

2. Durée du TP

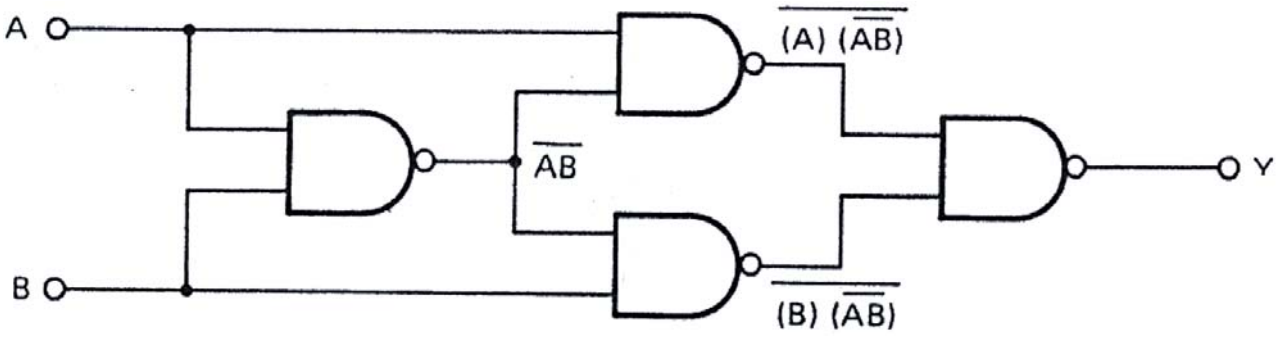
2 heures.

3. Matériels (Equipements et matière d'œuvre) par équipe

- Alimentation continue (+ 5 V)
- Plaquette de montage ou simulateur de fonctions logiques
- Interrupteurs logiques
- Pinces et fils
- Circuits intégrés 7400
- Notes de cours et fiches techniques

4. Description du TP

Soit le circuit montré à la figure suivante



Réaliser le circuit

5. Déroulement du TP

1) Réaliser le circuit et effectuer les connexions adéquates. Se rappeler que l'on doit relier la broche 14 du circuit intégré à la borne « + 5V » de l'alimentation et la broche 7 à la borne « GND ».

2) Relier les points A et B du montage aux interrupteurs de données "0" ou "1" suivant la table de vérité. Relier ensuite la sortie Y à une LED.

3) Remplir la table de vérité suivante : La LED allumée indique que la sortie est au niveau logique 1; la LED.éteinte indique que la sortie est au niveau logique 0.

A	B	Y
0	0	
0	1	
1	0	
1	1	

4) En utilisant l'algèbre de Boole simplifier l'équation pour la sortie

$Y = (A \bullet \overline{AB}) \times (B \bullet \overline{AB})$ =…… ;

5) A quelle fonction correspond l'équation simplifiée ? ;

Tp 4 : applications de l'algèbre de Boole

1. Objectif(s) visé(s) :

- *Simplification des équations :*
- *Traduire les équations en schémas :*
- *Reconnaître différents composants à partir des codes d'identification*

2. Durée du TP:

3 heures

3. Matériel (Équipement et matière d'œuvre) par stagiaire:

- *Alimentation continue (+ 5 V)*
- *Plaquette de montage ou simulateur de fonctions logiques*
- *Interrupteurs logiques*
- *Pinces et fils*
- *Résistance de limitation de 330 Ω*
- *Circuits intégrés 7400 TTL*
- *Notes de cours et fiches techniques*

4. Description du TP :

Soit l'équation :

$$S = \overline{\overline{A \bullet \overline{A \bullet B}} \bullet B \bullet \overline{A \bullet B}}$$

Questions :

1) *Simplifier algébriquement S;*
2) *Faire le logigramme de S en utilisant les portes logiques NON-ET (7400)*
3) *Réaliser pratiquement le circuit à l'aide des circuits intégrés de la famille TTL 7400;*
4) *Dresser la table de vérité du système.*

5. Déroulement du TP :

1) *Le stagiaire, à partir de l'équation donnée, fait les simplifications, en se basant sur les lois, les théorèmes et postulats connus de l'algèbre de Boole et les théorèmes De Morgan.*

2) *Montage du circuit avec les C I TTL 7400.*

3) *Vérification du fonctionnement.*

Tp 5 : établir la table de vérité d'un circuit

1. Objectif(s) visé(s) :

- *Expliquer les fonctions logiques de base ainsi que leurs tables de vérité :*
- *Faire le schéma de brochage et réalisation pratique du schéma*
- *Etablir la table de vérité d'un circuit :*
  - *Règles de construction*
  - *Équations*
  - *Résultats.*

2. Durée du TP:

3 heures.

3. Matériel (Équipement et matière d'œuvre) par stagiaire:

- *Alimentation continue (+ 5 V)*
- *Plaquette de montage ou simulateur de fonctions logiques*
- *Interrupteurs logiques*
- *Pinces et fils*
- *Résistance de limitation de 330 Ω*
- *Circuits intégrés TTL : 7432, 7400, 7408.*
- *Notes de cours et fiches techniques*

4. Description du TP :

Etablir la table de vérité du circuit suivant :

$S = \overline{(A+B)\bullet AC}$ *, en lisant l'état de la sortie S en fonction des entrées A et B.*

5. Déroulement du TP :

1) *Construction du logigramme de l'équation* $S = \overline{(A+B)\bullet AC}$

2) *Choix des composants :*
   - *circuit intégré TTL 7432 – pour la porte logique OU*
   - *circuit intégré TTL 7400 – pour la porte logique NAND*
   - *circuit intégré TTL 7408 – pour la porte logique ET*

3) *Montage des composants : positionnement, technique de travail.*

4) *Réalisation du circuit logique*

5) *Etablir la table de vérité à partir des manœuvres exécutés sur les interrupteurs logiques A, B et C.*

6) *Conclusions*

Tp 6: monter des circuits de base

1. Objectif(s) visé(s) :

- *Montage des circuits de base comme l'additionneur ;*
- *Positionnement des composants ;*
- *Conformité du montage avec le schéma ;*

2. Durée du TP:

3 heures.

3. Matériel (Équipement et matière d'œuvre) par stagiaire:

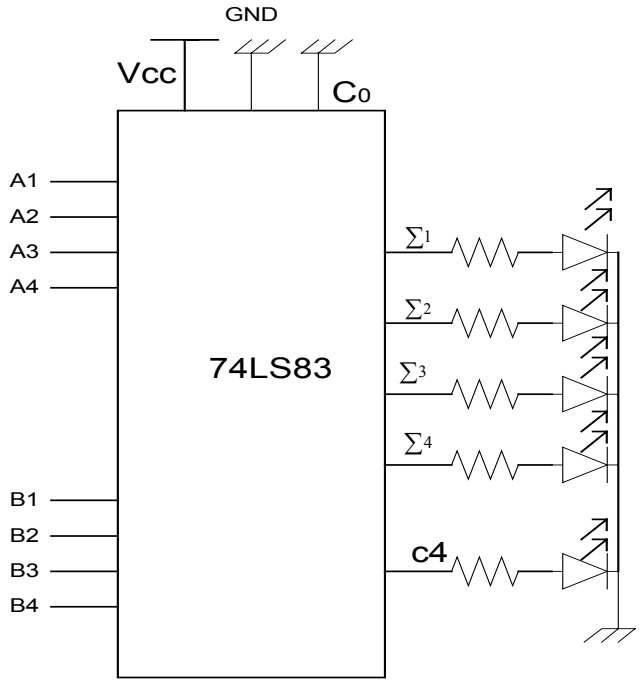
- *Alimentation continue (+ 5 V)*
- *Plaquette de montage ou simulateur de fonctions logiques*
- *Interrupteurs logiques*
- *Pinces et fils*
- *Résistances de limitation de 330 Ω*
- *Circuits intégrés 74LS83*
- *Notes de cours et fiches techniques*

4. Description du TP :

Compétences visées : Monter et vérifier le fonctionnement de l'additionneur binaire 74LS83.

5. Déroulement du TP :

1) *Monter le circuit de la figure suivante d'un additionneur binaire : addition de deux nombres binaires de quatre bits.*



a) En se basant sur le brochage du circuit 7483 mettre à coté de chaque broche le numéro correspondant ?
b) Câbler le circuit.
c) Vérifier le fonctionnement du circuit en prenant quelques exemples :

A1 A2 A3 A4
+ B1 B2 B3 B4

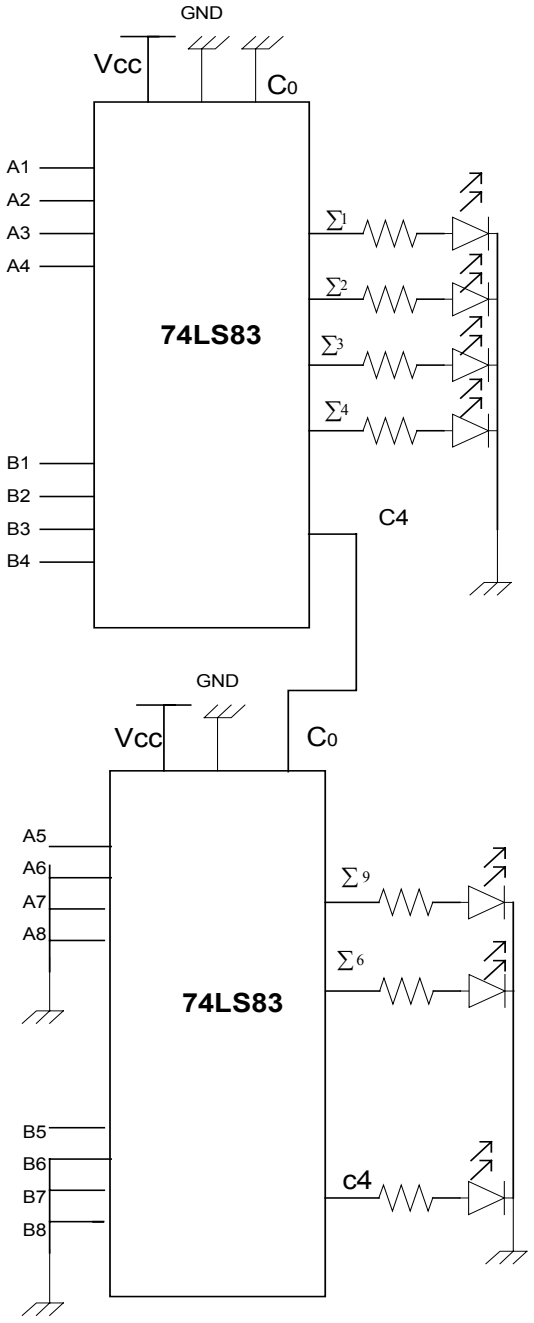
0100
+ 0111

1110
+ 0111

2) Monter le circuit de la figure suivante d'un additionneur binaire : addition de deux nombres binaires de cinq bits.

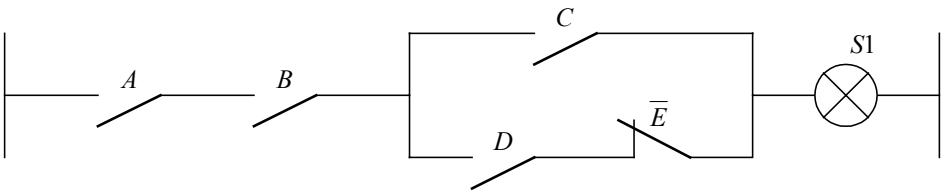
a) En se basant sur le brochage du circuit 7483 compléter les numéros des broches ajoutés?
b) Câbler le circuit.
c) Quelle est la retenue de cette addition ?

11110
+ 11011

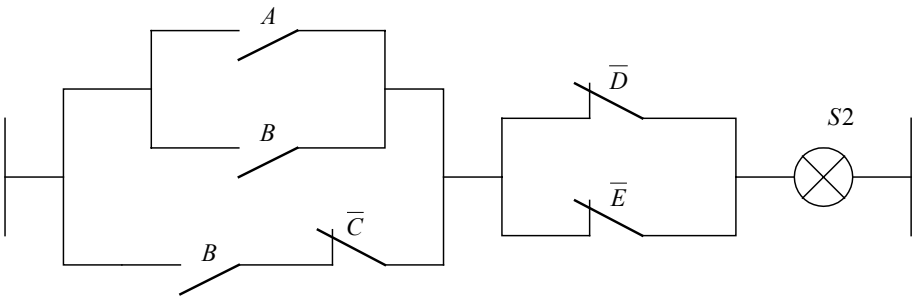

GND
Vcc
C0
A1
A2
A3
A4
74LS83
∑1
∑2
∑3
∑4
B1
B2
B3
B4
C4
GND
Vcc
C0
A5
A6
A7
A8
74LS83
∑9
∑6
B5
B6
B7
B8
c4

Evaluation de fin de module

1. *Transformer les schémas électriques suivants en équation :*
   a)



   b)



2. *Traduire les équations logiques suivantes en logigrammes :*

$$S_1 = \overline{\left(A\overline{B} + A\overline{B}\,\overline{C} + ABC\right)}\left(\overline{A}\,\overline{B} \oplus C\right)$$

$$S_2 = \left(\overline{A} + BD\right)C + \overline{\overline{B}\,C}$$

$$S_3 = A\,B \oplus \overline{A}\,\overline{\left(\overline{B\,C} + D\right)}$$

3. *Réaliser l'opérateur logique OU EXCLUSIF avec seulement des portes NON-OU à 2 entrées.*

4. *Un système électronique de sécurité reçoit trois informations binaires A, B et C. Il allume 3 voyants selon les conditions :*

   - *Si toutes les informations sont présentes, une lampe verte s'allume ;*
   - *Si une des informations est absente, une lampe jaune s'allume ;*
   - *Si au moins 2 informations sont absentes, une lampe rouge s'allume.*

   On vous demande :
   a) *D'établir la table de vérité du système ;*
   b) *De simplifier algébriquement l'équation des 3 voyants ;*
   c) *De donner le schéma logique correspondant.*
   d) *De réaliser à l'aide des portes logiques TTL le montage sur plaquette de montage ou simulateur de fonctions logiques.*

Liste des références bibliographiques

Ouvrage	Auteur	Edition
Introduction aux circuits logiques	Jean Letocha	McGraw-Hill
Circuits numériques	Ronald Tocci	Dunod
Électronique appliquée	P. Léautey, P. Salette, A. Bianciotto, P. Boye	Delagrave
Module logique combinatoire	CEMEQ	