

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° 5 MAINTENANCE D'UN SYSTEME
COMBINATOIRE**

SECTEUR : ELECTROTECHNIQUE

SPECIALITE : MMOAMPA

NIVEAU : TECHNICIEN SPECIALISE

ANNEE 2007

Document élaboré par :

Nom et prénom

EFP

DR

Mme ELKORNO NAIMA

CDC - GE

Révision linguistique

-
-
-

Validation

-
-
-

SOMMAIRE

RESUME THEORIQUE	7
I. Les fonctions logiques de base.....	8
I.1 Les trois fonctions logiques de base.....	8
I.2 Les autres fonctions logiques :	9
I.3 Les fonctions logiques matérialisées avec des interrupteurs:.....	10
I.4 Symbolisation	14
II. Logique booléenne :	15
II.1 Définition :	15
II.2 Les lois de l'algèbre de Boole.....	15
II.3 Les Théorèmes de l'algèbre de Boole :	16
II.4 Postulats de l'algèbre de Boole :	16
II.5 Simplification algébrique des équations booléennes :	16
III. Circuits intégrés logiques	17
III.1 Famille TTL (transistor transistor logic).....	17
III.2 Famille CMOS (Complementary Metal Oxide Semiconductor).....	20
III.3 Caractéristiques des circuits intégrés :	20
III.4 Configuration des broches pour les différents modèles des C.I.....	24
IV. Les systèmes de numération :	28
IV.1 Base d'un système de numération:	28
IV.2 Changement de base :	30
IV.3 Opérations arithmétiques avec la base binaire:.....	34
IV.4 Les codes :	35
V. Table de vérité.....	37
V.1 Tableau des combinaisons.....	37
V.2 Règles de construction d'une table de vérité :	38
V.3 Ecriture d'une équation à partir d'une table de vérité	39
V.4 Elaboration d'une table de vérité à partir d'une équation.....	41
VI. Simplification des fonctions logiques par la méthode de Karnaugh :	41
VI.1 Transposition d'une équation logique dans un diagramme de Karnaugh	42
VI.2 Simplification d'une équation par le diagramme de Karnaugh.....	45
VI.3 Écriture des équations à partir de regroupement.....	48
VII. Schémas logiques :	50
VII.1 Généralités :	50
VII.2 Différents types de schémas logiques:.....	50
VIII. Montage des circuits	57
VIII.1 Plaque de montage	57
VIII.2 Interrupteurs logiques	57
VIII.3 Choix de logique positive ou négative, visualisation des sorties	58
VIII.4 Technique de travail.....	60
IX. Circuits arithmétiques.....	61
IX.1 Additionneur	61
IX.2 Soustracteurs logiques	65
IX.3 Multiplicateur	67
IX.4 Circuit comparateur	69
X. Circuits de décodage des informations.....	70
X.1 Le codeur :	70

X.2	Le décodeur.....	71
X.3	Circuits de transcodage.....	73
XI.	Circuits de transfert d'information.....	74
XI.1	Multiplexeur.....	74
XI.2	Demultiplexeur.....	75
GUIDE DES EXERCICES ET TRAVAUX PRATIQUES		77
Exercices :		78
Tp 1 - portes logiques fondamentales : et (and), ou (or). Non (not)		80
Tp 2 - portes logiques fondamentales : non-et (nand), non-ou (nor), ou exclusif (xor)		82
Tp 3 - applications de l'algèbre de Boole		84
Tp 4 : applications de l'algèbre de Boole		86
Tp 5 : établir la table de vérité d'un circuit.....		87
Tp 6: monter des circuits de base.....		88
Tp 7: monter des circuits de base.....		90
Tp 8: monter des circuits de base.....		93
Tp 9: monter des circuits de base.....		95
Evaluation de fin de module.....		97
Liste des références bibliographiques.....		99

MODULE 5 : MAINTENANCE D'UN SYSTEME COMBINATOIRE

Code :

Durée : 60 h

**OBJECTIF OPÉRATIONNEL DE PREMIER NIVEAU
DE COMPORTEMENT**

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'ÉVALUATION

- 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 OPERATIONNEL

**PRECISIONS SUR LE
COMPORTEMENT ATTENDU**

**CRITERES PARTICULIERS DE
PERFORMANCE**

- | | |
|--|---|
| <p>A) Réduire l'encombrement d'un circuit logique en utilisant les lois et les relations de base de l'algèbre de BOOLE.</p> <p>B) Décrire les caractéristiques d'un circuit intégré numérique en utilisant le DATA BOOK et l'associer correctement à un autre circuit de technologie différente.</p> <p>C) Exprimer l'information binaire en différents bases et codes.</p> <p>D) Réaliser la partie commande d'un système combinatoire et vérifier son fonctionnement</p> <p>E) Utiliser et vérifier les principaux circuits intégrés utilisés dans un système de contrôle numérique</p> | <ul style="list-style-type: none">- Matérialisation optimale des équations d'état en utilisant correctement les lois et les relations de l'algèbre de BOOLE.
- Exploitation juste des caractéristiques techniques des principales familles des circuits intégrés.
- Exactitude des conversions.- Interprétation correcte des codes utilisés dans les calculateurs.
- Respect des règles prescrites.- Regroupement optimal des variables.- Exactitude des résultats.
- Utilisation correcte des principales fonctions rencontrées dans les unités centrales.- Lecture et interprétation correcte d'un schéma de commande.- Localisation et remplacement opérationnel de la partie défectueuse.- Application des règles de sécurité.- Protection de l'environnement. |
|--|---|

Présentation du Module :

Ce module 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.

**MODULE N° 5: MAINTENANCE D'UN SYSTEME
COMBINATOIRE**

RESUME THEORIQUE

I. Les fonctions logiques de base

➤ Variables booléennes :

Une variable booléenne est une grandeur physique qui ne peut prendre que deux états Stables.

Exemples :

- Appareils de commande :
 - Un interrupteur peut être fermé ou ouvert;
 - Un bouton poussoir peut être actionné ou non actionné.
- Récepteurs :
 - Une lampe d'éclairage peut être allumée ou éteinte;
 - Un électro-aimant peut être excité ou non excité.

➤ Etat logique d'une variable booléenne :

Ce sont les deux états stables d'une variable. Par convention, chaque état stable est désigné par un chiffre qui est zéro (0) ou un (1).

I.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 \bar{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 : $F = \bar{A}$

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

1.2 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.

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

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

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 connexions 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 OU-EXCLUSIF B sera également une variable booléenne. **La table de vérité** de droite montre le comportement de cette fonction. la sortie de la fonction logique OU-EXCLUSIF (en anglais EXOR) ne donne un niveau logique 1 que si, et seulement si, une seule entrée est à l'état 1.

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

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 : $F = A \oplus B = \overline{A}B + A\overline{B}$

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

Soit deux variables booléennes nommées A et B. Le résultat de la fonction logique NON (A OU-EXCLUSIF 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-OU-EXCLUSIF est l'inverse de celle obtenue avec la fonction logique OU-EXCLUSIF.

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 $F = \overline{A \oplus B}$.

1.3 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 »**

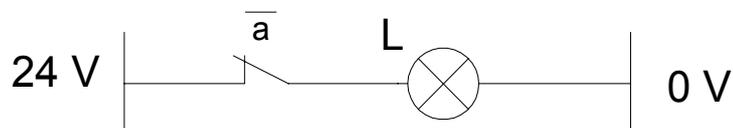
La figure ci-dessous montre la fonction logique OUI. Cette fonction utilise simplement un interrupteur normalement ouvert. Le voyant s'allumera si l'interrupteur est actionné.



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

b) **Fonction logique « NON »**

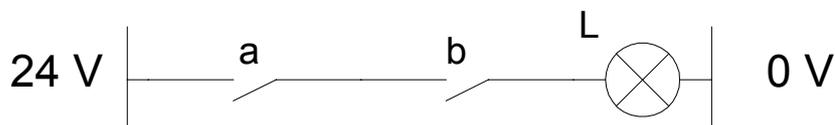
La figure ci-dessous montre la fonction logique NON. Pour obtenir cette fonction, il suffit de brancher un interrupteur normalement fermé. Le voyant s'allume tant que l'interrupteur n'est pas actionné.



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

c) **Fonction logique « ET »**

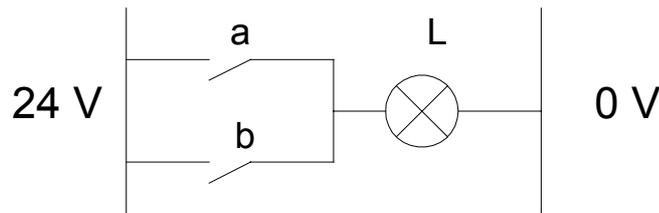
La figure ci-dessous montre la fonction logique ET. Pour obtenir cette fonction, il suffit de brancher deux interrupteurs normalement ouverts en série. Pour que le courant puisse traverser le voyant et l'allume, il faut actionner simultanément les deux interrupteurs.



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

d) **Fonction logique « OU »**

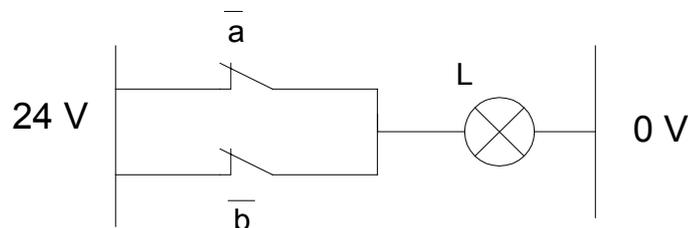
La figure ci-dessous montre la fonction logique OU. Pour obtenir cette fonction, il suffit de brancher deux interrupteurs normalement ouverts en parallèle. Le voyant s'allume dès que l'un des interrupteurs est actionné.



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

e) Fonction logique « NON-ET »

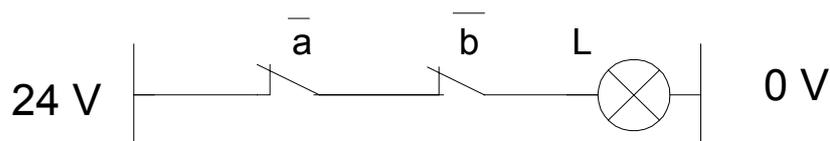
La figure ci-dessous montre la fonction logique NON-ET. Pour obtenir cette fonction, il suffit de brancher deux interrupteurs normalement fermés en parallèle. Le voyant s'éteint seulement si les deux interrupteurs sont actionnés simultanément.



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

f) Fonction logique « NON-OU »

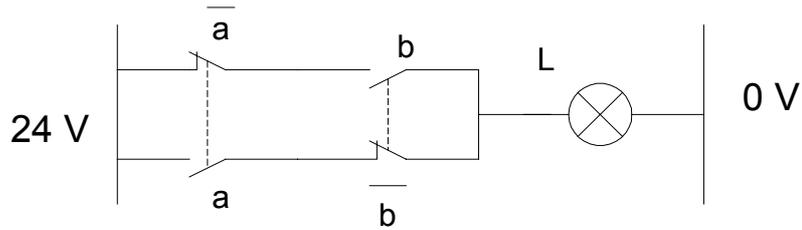
La figure ci-dessous montre la fonction logique NON-OU. Pour obtenir cette fonction, il suffit de brancher deux interrupteurs normalement fermés en série. Le voyant s'éteint dès qu'un des interrupteurs est actionné.



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

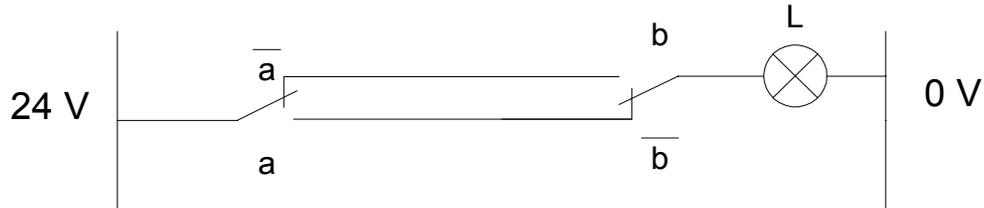
g) Fonction logique « OU-EXCLUSIF »

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.



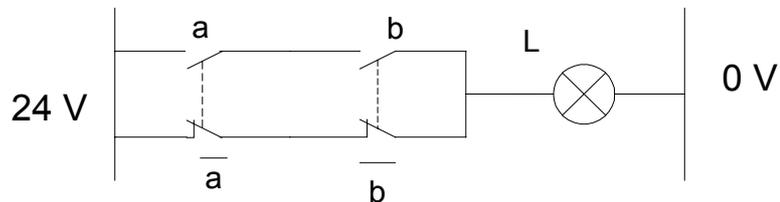
La fonction logique de ce montage est : $L = a \oplus b = \bar{a} b + \bar{b} a$

ou encore :



h) Fonction logique « NON-OU-EXCLUSIF »

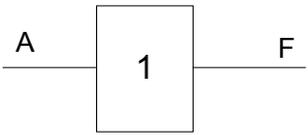
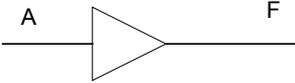
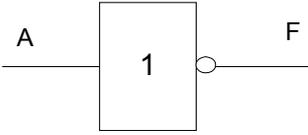
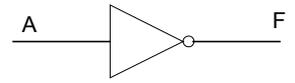
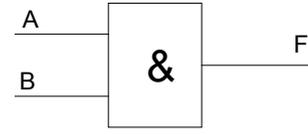
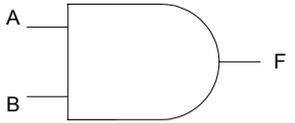
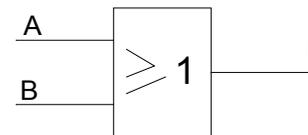
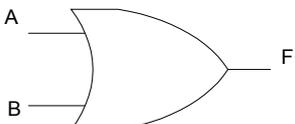
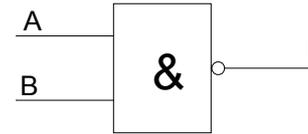
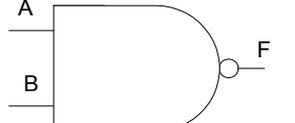
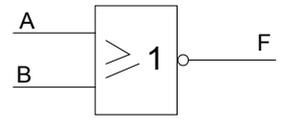
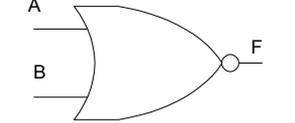
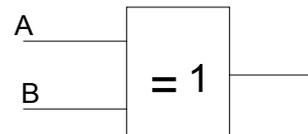
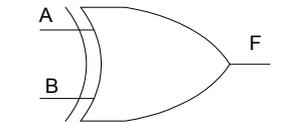
La figure ci-dessous montre la fonction logique NON-OU-EXCLUSIF. Pour obtenir cette fonction, il suffit de brancher, tel que montré, deux interrupteurs ayant deux contacts chacun, l'un normalement ouvert, l'autre normalement fermé. Le voyant s'éteint si et seulement si un seul interrupteur est actionné.



La fonction logique de ce montage est :

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

I.4 Symbolisation

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

II. Logique booléenne :

II.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.

II.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}	$\overline{\overline{A + B}} = A \bullet B$
	L_{14}	$\overline{\overline{A \bullet B}} = A + B$
Similitude	L_{15}	$A + \overline{A} \bullet B = A + B$
	L_{16}	$A \bullet (\overline{A} + B) = A \bullet B$

II.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	$A \bullet \bar{A} = 0$
	T_8	$A + \bar{A} = 1$
Involution	T_9	$\bar{\bar{A}} = A$

II.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	$\bar{0} = 1$
P_8	$\bar{1} = 0$

II.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 : $z = (a + b)(\bar{b} + c)(\bar{a} + c)$

$$z = (a\bar{b} + ac + b\bar{b} + bc)(\bar{a} + c)$$

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

$$z = (a\bar{b} + ac + bc)(\bar{a} + c) \text{ développons, on aura :}$$

$$z = a\bar{b}\bar{a} + a\bar{b}c + a\bar{c}\bar{a} + acc + bc\bar{a} + bcc$$

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

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

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

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

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

III. Circuits intégrés logiques

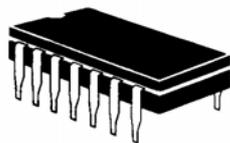
La matérialisation des fonctions logiques a été d'abord réalisée avec des composants discrets puis elle s'est transformée en intégrant plusieurs composants sur un seul circuit. Un circuit intégré (C.I.) désigne un bloc constitué par un monocristal de silicium de quelques millimètres carrés en forme parallélépipède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

Les composants électroniques peuvent être de deux familles : TTL ou CMOS.

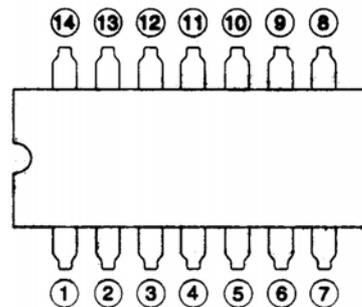
III.1 Famille TTL (transistor transistor logic)

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

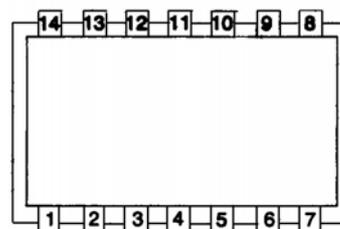
Ces C.I. sont constitués d'un boîtier qui contient la puce, laquelle est reliée à l'extérieur par un certain nombre de pattes (ou broches). Ce nombre varie généralement entre 14 et 28. La tension d'alimentation nominale (V_{cc}) est +5 V.



Vue en trois dimensions



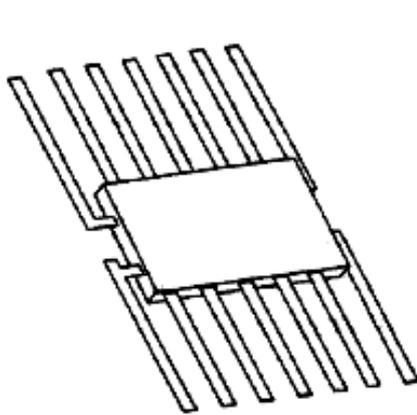
Vue de dessus avec identification des pattes



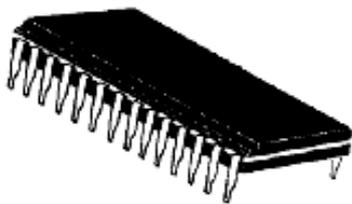
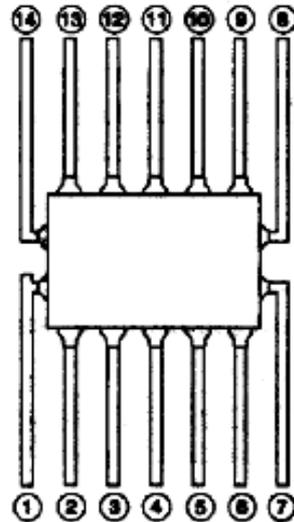
Représentation généralement employée pour l'identification

Circuit intégré

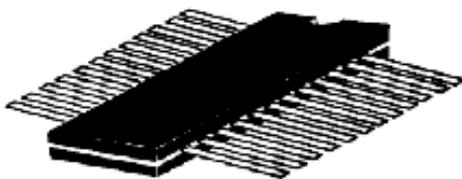
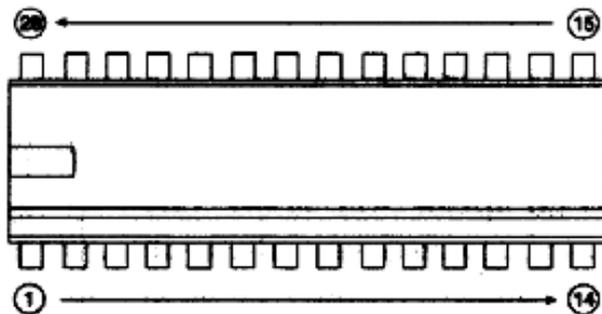
On peut remarquer qu'il y a encoche sur des côtés du boîtier. Elle permet de localiser rapidement le numéro de chaque patte et d'obtenir un branchement simple et rapide. Il existe plusieurs formes de boîtiers pour les circuits intégrés (voir figure suivante).



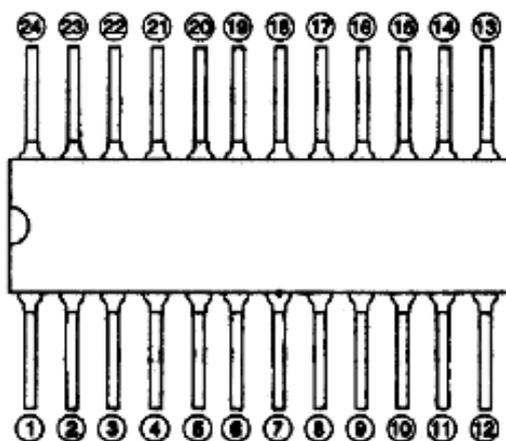
Type « T »



Type « N »



Type « W »



Types de boîtiers

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

Exemples :

a) SN 54L121 N

SN : préfix standard;
5 : gamme militaire;
4 : circuit logique;
L : sous groupe faible consommation;
121 : fonction du circuit intégré;
N : boîtier de plastique enfichable 14 ou 16 broches.

b) SN 74LS10J

SN : préfix standard;
7 : gamme générale;
4 : circuit logique;
LS : sous groupe, Schottky faible consommation;
10 : fonction du circuit intégré (NON ET);
J : type de boîtier : boîtier céramique enfichable DIL 14 ou 16 broches.

c) SN 74LS00

SN : préfix standard;
7 : gamme générale;
4 : circuit logique;
LS : (sous groupe), Schottky faible consommation;
00 : fonction du circuit logique (NON ET).

III.2 Famille CMOS (Complementary Métal Oxyde Semiconductor)

Elle dérive principalement des transistors à effet de champ.

a) Circuits intégrés CMOS série 74 :

- CMOS RAPIDES (High-speed CMOS) 74 HC xx et 74 HCT xx ;
- CMOS classiques 74 C xx (technologie identique à la série 4000) ;
- CMOS advanced (ACL) 74 AC 11 xx et 74 ACT 11 xx ;
- CMOS Low-voltage (faible tension) 74 LV xx et 74 LVC xx ;

b) Circuits intégrés CMOS série 4000 :

Deux types de circuits sont couramment commercialisés: la série 4000 UB et la série 4000 B. Cette dernière a toutes ses sorties bufférisées (sorties amplifiées). Elles sont entièrement compatibles.

III.3 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.

1) tension d'alimentation :

La tension nécessaire pour alimenter le circuit. TTL est +5 V.

Pour La série 4000 et la série 74 C des circuits intégrés CMOS, ils sont fonctionnels pour des valeurs de V_{DD} comprises entre 3 et 15 V. Les séries 74 HC et 74 HCT fonctionnent sous des tensions d'alimentation allant de 2 à 6 V.

2) Courants et tensions

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.

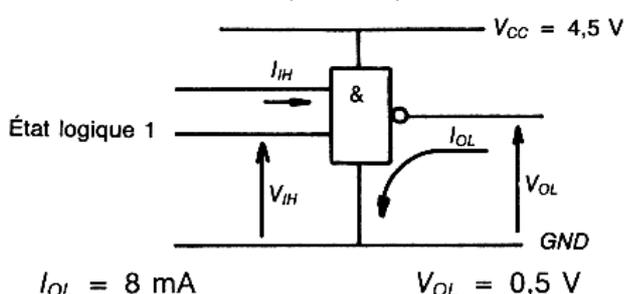
I_{OL} - courant de sortie niveau BAS : le courant qui traverse une borne de sortie placée au niveau logique 0 dans des conditions de charge spécifiées.

Exemples :

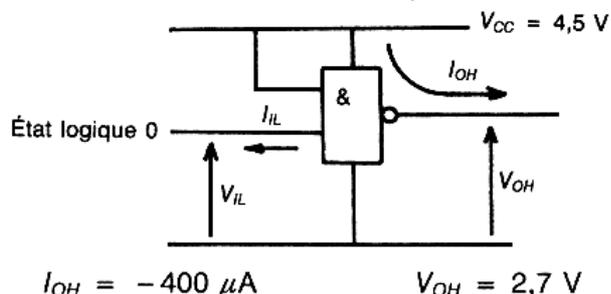
➤ pour le circuit intégré 74 LS00 (voir tableau suivant)

COURANTS ET TENSIONS

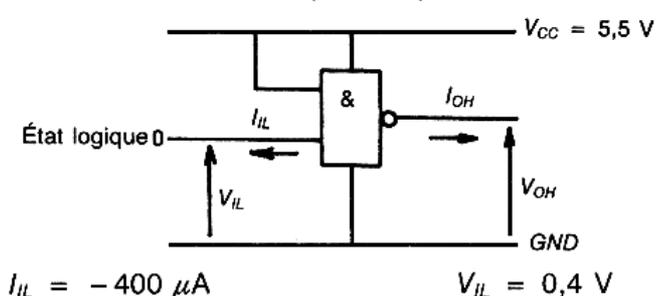
Niveau de sortie bas (74LS00)



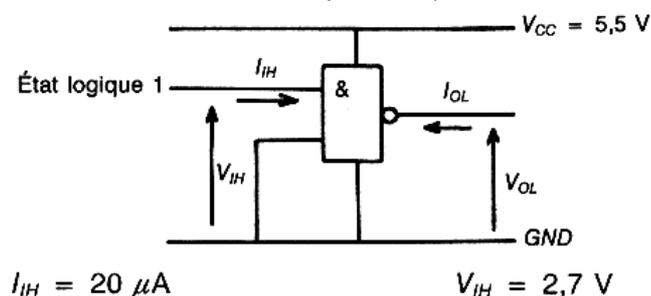
Niveau de sortie haut (74LS00)



Niveau d'entrée bas (74LS00)



Niveau d'entrée haut (74LS00)



Les courants sont comptés positifs lorsqu'ils rentrent dans le circuit.

➤ Valeurs des courants et tensions d'une porte des différentes familles de la série 74

Grandeurs $V_{CC} = 5V$	Familles de la série 74									Unités
	7400	74L00	74S00	74LS00	74AS00	74ALS00	74F00	74HC00	74HCT00	
I_{OL} min.	16	3,6	20	8	20	8	20	4	4	mA
I_{OH} max.	-400	-200	-1000	-400	-2000	-400	-1000			μA
I_{IL} max.	-1,6	-0,18	-2	-0,36	-0,5	-0,2	-0,6	$\pm 0,001$	$\pm 0,001$	mA
I_{IH} max.	40	10	50	20	20	20	20			μA
V_{OL} max.	0,4	0,4	0,5	0,5	0,5	0,5	0,5	0,1	0,1	V
V_{OH} min.	2,4	2,4	2,7	2,7	$V_{CC} - 2$	$V_{CC} - 2$	2,5	4,9	4,9	V
V_{IL} max.	0,8	0,7	0,8	0,7	0,8	0,8	0,8	1	0,8	V
V_{IH} min.	2	2	2	2	2	2	2	3,5	2	V

Ces valeurs peuvent légèrement varier en fonction des conditions de test.

3) 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.

4) 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.

5) Consommation :

Tous les CI, quand ils sont en fonctionnement, consomment une certaine énergie électrique. Cette énergie est délivrée par une ou plusieurs tensions d'alimentation raccordées aux broches d'alimentation du boîtier. Généralement, il n'y a qu'une borne d'alimentation par boîtier, appelée V_{CC} (pour les TTL) ou V_{DD} (pour les MOS).

La puissance dissipée par un CI est généralement spécifiée par le courant I_{CC} qu'il tire de l'alimentation V_{CC} , de sorte que la puissance réelle est le produit $I_{CC} \times V_{CC}$. Dans le cas de nombreux CI, le courant qu'ils drainent de l'alimentation varie en fonction des états logiques des circuits de la puce. Par exemple une puce NON-ET dont toutes les sorties des portes sont au niveau HAUT, le courant tiré de l'alimentation V_{CC} dans ce cas est appelé I_{CCH} . Par contre quand toutes les sorties des portes sont au niveau BAS. Ce dernier courant est appelé I_{CCL} .

Habituellement, I_{CCH} et I_{CCL} ne sont pas égaux. Le courant moyen est :

$$I_{CC(moy)} = (I_{CCH} + I_{CCL})/2$$

Et c'est cette valeur qu'on utilise pour calculer la puissance moyenne consommée :

$$P_D(\text{moy}) = I_{CC}(\text{moy}) \times V_{CC}$$

6) Produit vitesse-consommation

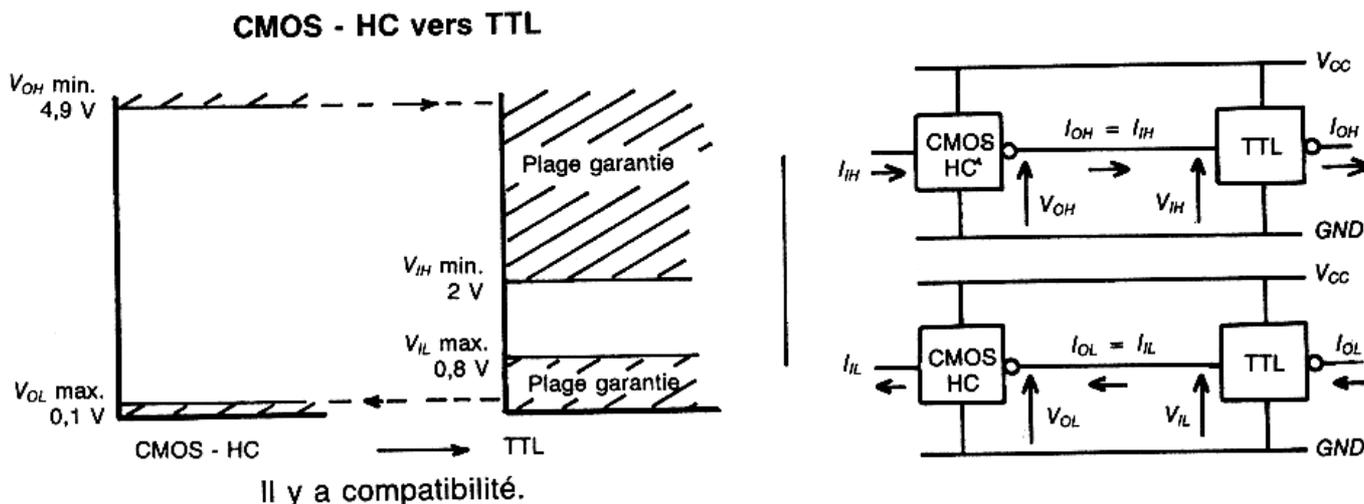
Les familles de circuits intégrés ont toujours été caractérisées à la fois par la vitesse et la consommation. Généralement, il est souhaitable d'avoir un délai de propagation pour la porte de même qu'une consommation d'énergie les plus faibles possibles. Les diverses familles et sous-familles de circuits logiques ont des valeurs pour la vitesse et la consommation très variables qui se répartissent dans un spectre très large. Une façon commode de mesurer la performance globale d'une famille est le calcul du produit vitesse-consommation que l'on obtient en multipliant le retard de propagation d'une porte par la consommation d'énergie de celle-ci. Par exemple, supposons une famille dont le retard moyen est 10 ns et la consommation moyenne, 5 mW. Le produit vitesse-consommation est donc:

$$10 \text{ ns} \times 5 \text{ mW} = 50 \times 10^{-12} \text{ watt-seconde} = 50 \text{ picojoules (pj)}$$

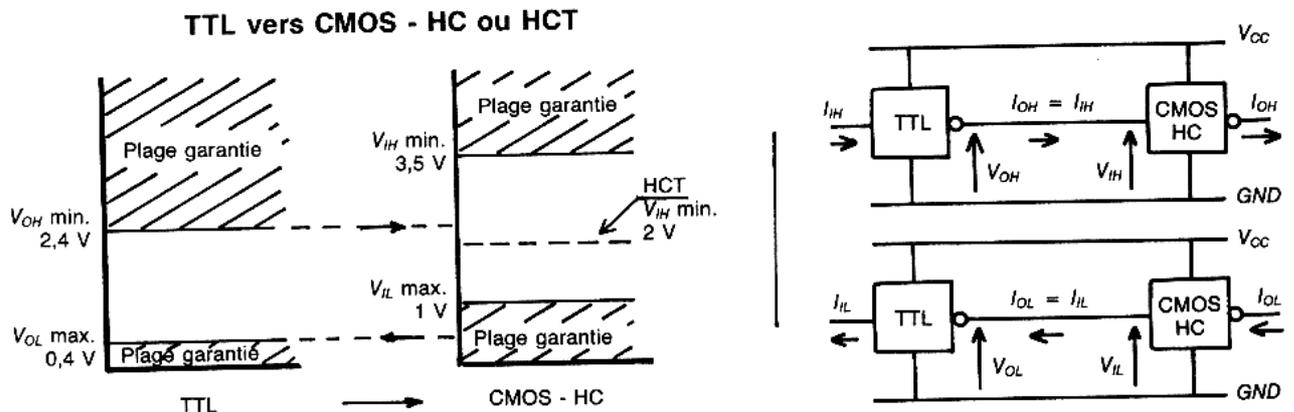
7) Association des circuits intégrés

Il est nécessaire de prendre certaines précautions lors de l'association de circuits de familles différentes. Les niveaux des tensions d'entrée et de sortie pour les états haut et bas doivent être compatibles pour assurer correctement le changement d'état du circuit commandé. Il faut s'assurer que V_{OH} du circuit amont correspond à V_{IH} du circuit commandé ainsi que V_{OL} à V_{IL} .

- CMOS-HC vers TTL (voir figure suivante) :

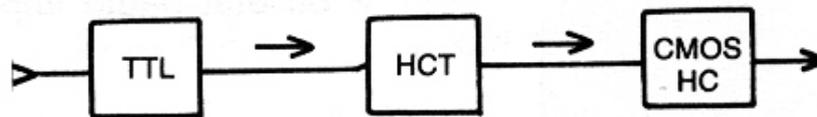


- TTL vers CMOS-HC ou HCT (voir figure suivante) :



Il n'y a pas de compatibilité avec CMOS-HC, mais il y a compatibilité avec HCT. Il est nécessaire d'adapter $V_{OH\ min}$ -TTL au niveau de $V_{IH\ min}$ - CMOS-HC. Dans ce cas on peut utiliser par exemple un circuit HCT.

En effet le circuit HCT est compatible avec les circuits TTL, HC et MOS. Il peut servir d'interface pour adapter les niveaux. Il suffit de remplacer le circuit TTL ou CMOS par un circuit HCT ou encore en utiliser un comme interface.

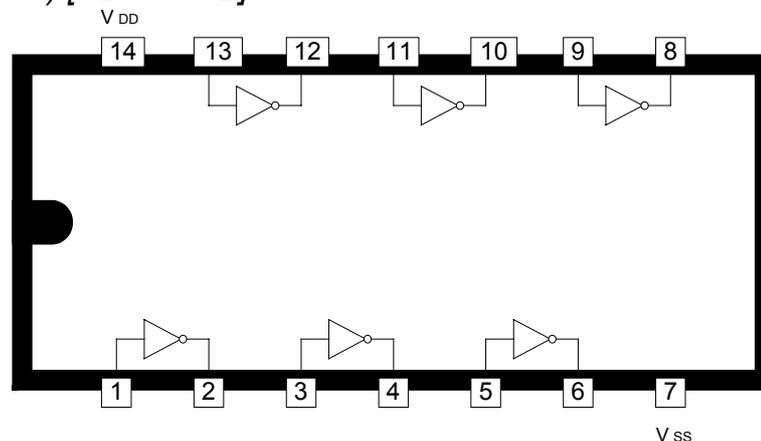


Pour le choix d'un circuit intégré c'est la vitesse (temps de propagation interne du signal) ou la puissance dissipée qui détermine le choix d'une famille.

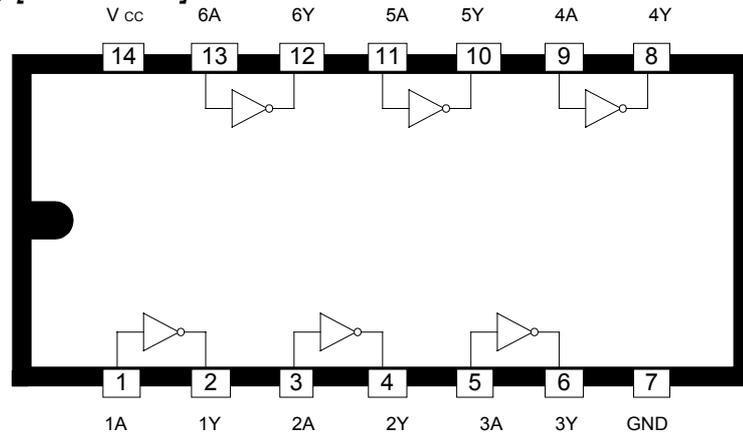
III.4 Configuration des broches pour les différents modèles des C.I

a) Circuit intégré portes «NON»

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

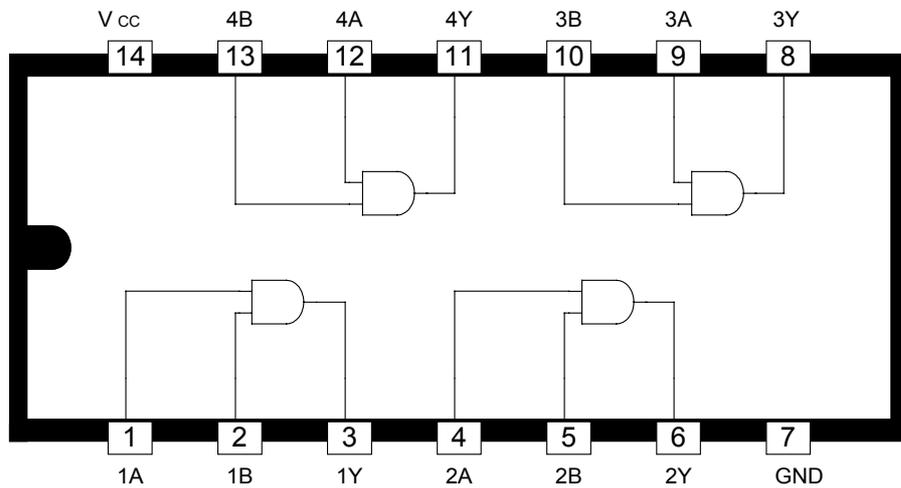


➤ Six portes NON : 7404 (TTL) [SN74LS04]

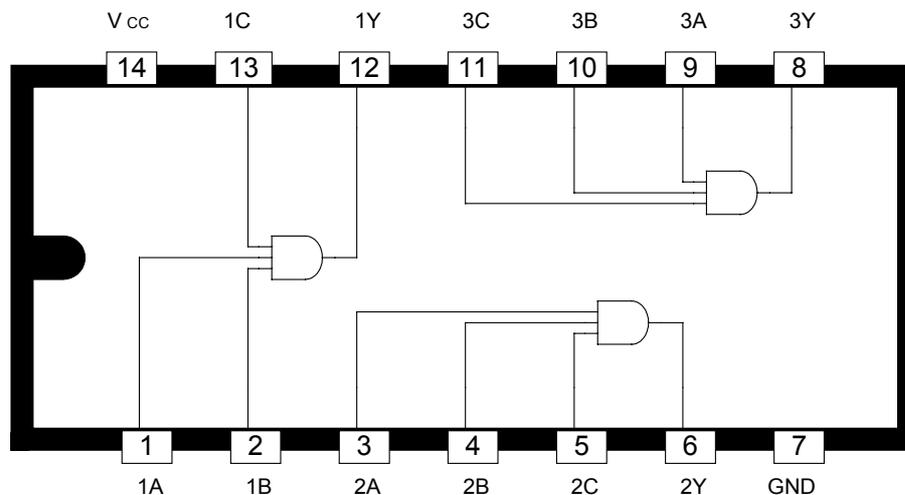


b) Circuit intégré Portes «ET»

➤ Quatre portes «ET» à deux entrées : 7408 (TTL) [SN74LS08]

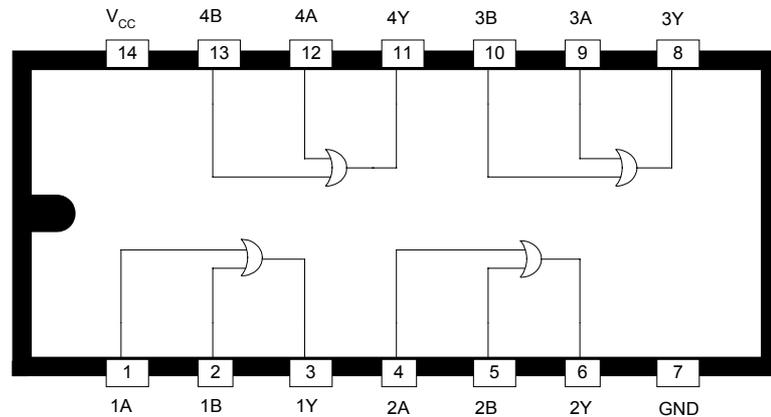


➤ Trois portes «ET» à trois entrées : 7411 (TTL) [SN74LS11]

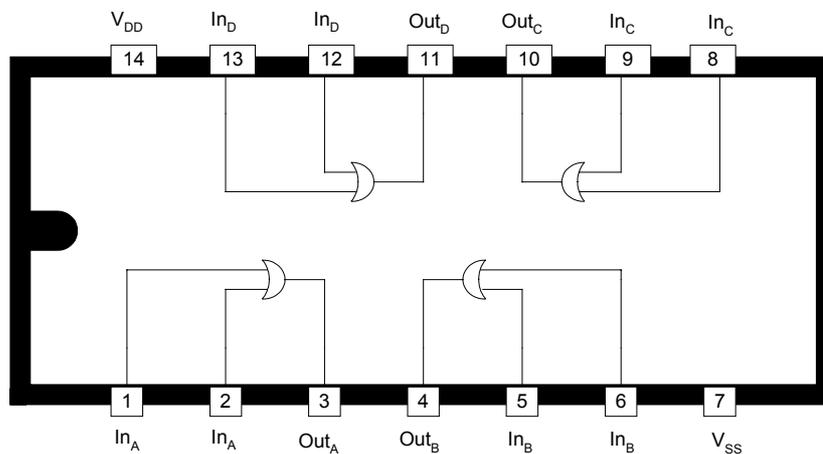


c) Circuits intégrés portes «OU»

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

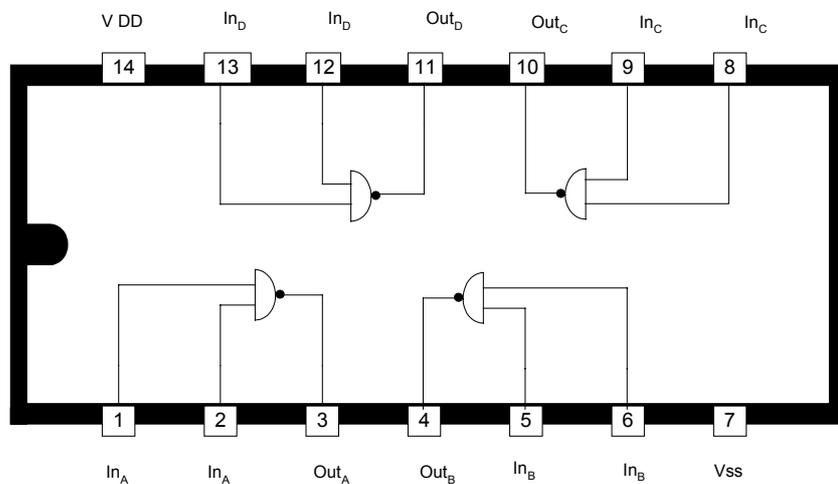


➤ Quatre portes «OU» à deux entrées : 4071 (COMS) [CD4071B]

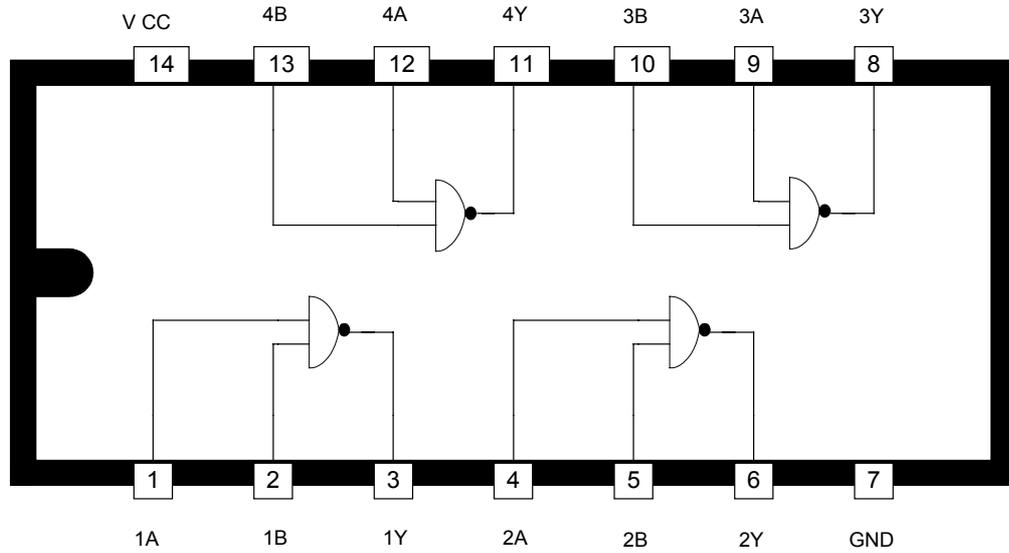


d) Circuits intégrés «NON ET»

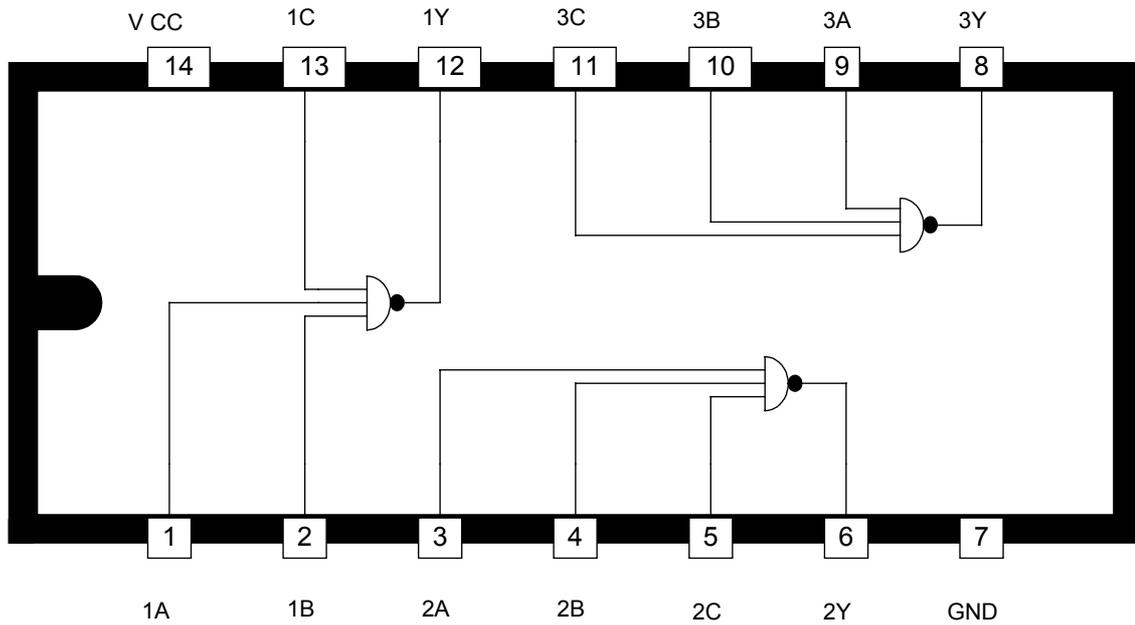
➤ Quatre portes «NON ET» à 2 entrées : 4011 (C-MOS) [CD4011]



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

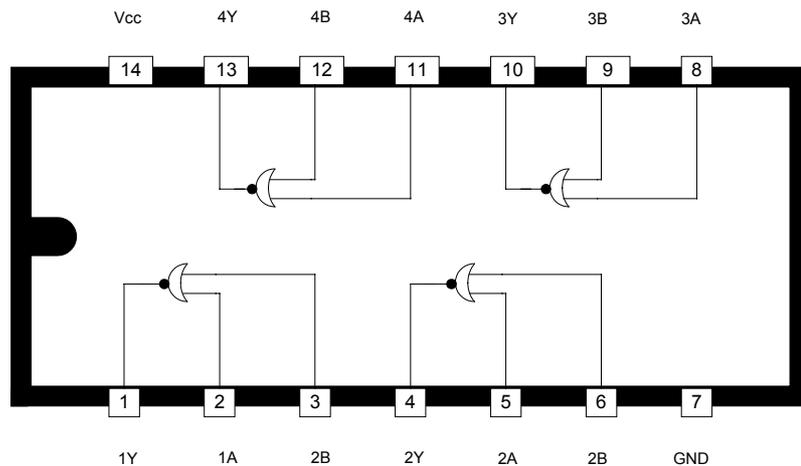


➤ Trois portes «NON ET» à 3 entrées : 7410 (TTL) [SN74LS10]



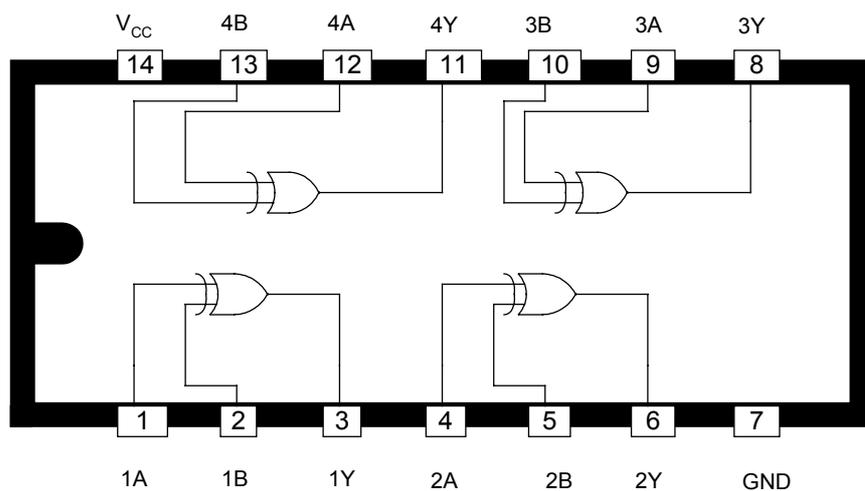
e) Circuits intégrés portes «NON OU»

➤ 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]*



IV. Les systèmes de numération :

IV.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 :

$$N = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \dots + a_0 \times 10^0$$

(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	a_0	a_1	a_2	a_3	a_4	a_5
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

IV.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)
	$2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$		
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

$$\begin{array}{r|l} 3786 & 2 \\ \hline 0 & 1893 \end{array}$$

$$\begin{array}{r|l} 1893 & 2 \\ \hline 1 & 946 \end{array}$$

$$\begin{array}{r|l} 946 & 2 \\ \hline 0 & 473 \end{array}$$

$$\begin{array}{r|l} 473 & 2 \\ \hline 1 & 236 \end{array}$$

$$\begin{array}{r|l} 236 & 2 \\ \hline 0 & 118 \end{array}$$

$$\begin{array}{r|l} 118 & 2 \\ \hline 0 & 59 \end{array}$$

$$\begin{array}{r|l} 59 & 2 \\ \hline 1 & 29 \end{array}$$

$$\begin{array}{r|l} 29 & 2 \\ \hline 1 & 14 \end{array}$$

$$\begin{array}{r|l} 14 & 2 \\ \hline 0 & 7 \end{array}$$

$$\begin{array}{r|l} 7 & 2 \\ \hline 1 & 3 \end{array}$$

$$\begin{array}{r|l} 3 & 2 \\ \hline 1 & 1 \end{array}$$

$$\begin{array}{r|l} 1 & 2 \\ \hline 1 & 0 \end{array}$$

Le nombre binaire ainsi obtenu est : $3786 = (111011001010)_2$

- On traite après la partie fractionnaire :

$$\begin{array}{r|l} 0,4 \times 2 = & \underline{0},8 \\ 0,8 \times 2 = & \underline{1},6 \\ 0,6 \times 2 = & \underline{1},2 \\ 0,2 \times 2 = & \underline{0},4 \\ 0,4 \times 2 = & \underline{0},8 \end{array}$$

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 :

$0,3 \times 8 =$		<u>2</u> , 4
$0,4 \times 8 =$		<u>3</u> , 2
$0,2 \times 8 =$		<u>1</u> , 6
$0,6 \times 8 =$	▼	<u>4</u> , 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 : $N = (257)_8 = \begin{matrix} (010 & 101 & 111)_2 \\ 2 & 5 & 7 \end{matrix}$

- **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 : $N = (11001101111)_2 = \begin{matrix} 11 & 001 & 101 & 111 \\ 3 & 1 & 5 & 7 \end{matrix}$
 $N = (3157)_8$

- **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
 $N = (86F)_{16}$

IV.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 :

111	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 :

$$\begin{array}{r}
 101 \\
 * \\
 \hline
 110 \\
 000 \\
 101 \\
 \hline
 101 \\
 \hline
 11110
 \end{array}$$

d) Division Binaire : Les règles de base sont :

0 / 0 = Indéterminé

0 / 1 = 0

1 / 0 = Impossible

1 / 1 = 1

Exemple :

$$\begin{array}{r}
 1010 \quad / \quad 10 \\
 \hline
 -10 \quad \boxed{101} \\
 001 \\
 \hline
 -00 \\
 10 \\
 \hline
 -10 \\
 00
 \end{array}$$

IV.4 Les codes :

a) Code binaire :

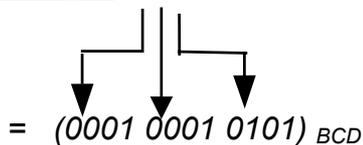
Décimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

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 : $(115)_{10}$



c) Code ASCII

(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.

V. Table de vérité**V.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 entrées 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

V.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 $(2^n - 1)$;
- Inscire, 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

V.3 Écriture 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

\rightarrow 1^{ère} ligne : $(A + B)$
 \rightarrow 4^{ème} ligne : $(\bar{A} + \bar{B})$

\Rightarrow équation : $S = (\bar{A} + \bar{B}) \bullet (A + B)$

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

\rightarrow 2^{ème} ligne : $(A + B + \bar{C})$
 \rightarrow 4^{ème} ligne : $(A + \bar{B} + \bar{C})$
 \rightarrow 7^{ème} ligne : $(\bar{A} + \bar{B} + C)$

\Rightarrow équation : $S = (\bar{A} + \bar{B} + C) \bullet (A + \bar{B} + \bar{C}) \bullet (A + B + \bar{C})$

Remarque :

Variable = 1 \Rightarrow $\overline{\text{Variable}}$
Variable = 0 \Rightarrow 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

\rightarrow 2^e ligne : $(\overline{A} \bullet B)$ } \Rightarrow L'équation : $S = (\overline{A} \bullet B) + (A \bullet \overline{B})$
 \rightarrow 3^e ligne : $(A \bullet \overline{B})$ }

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

\rightarrow 1^e ligne : $(\overline{A} \bullet \overline{B} \bullet \overline{C})$ } \Rightarrow l'équation :
 \rightarrow 3^e ligne : $(\overline{A} \bullet B \bullet \overline{C})$ } $S = (\overline{A} \bullet \overline{B} \bullet \overline{C}) + (\overline{A} \bullet B \bullet \overline{C}) + (A \bullet \overline{B} \bullet \overline{C}) +$
 \rightarrow 5^e ligne : $(A \bullet \overline{B} \bullet \overline{C})$ } $(A \bullet \overline{B} \bullet C) + (A \bullet B \bullet C)$
 \rightarrow 6^e ligne : $(A \bullet \overline{B} \bullet C)$ }
 \rightarrow 8^e ligne : $(A \bullet B \bullet C)$ }

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 $\Rightarrow S = \overline{A} \bullet B + A \bullet B$ (1)
 Produit de sommes $\Rightarrow S = (A + B) \bullet (\overline{A} + B)$ (2)

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

$$\begin{aligned}
 (1) \implies S &= \bar{A} \cdot B + A \cdot B \\
 &= (\bar{A} + A) \cdot B && \text{(Distributivité } L_5) \\
 &= 1 \cdot B && \text{(} T_8 : \text{ Complémentarité)} \\
 &= B && \text{(} T_3 : \text{ Élément neutre)}
 \end{aligned}$$

$$\begin{aligned}
 (2) \implies S &= (A + B) \cdot (\bar{A} + B) \\
 &= A \cdot \bar{A} + A \cdot B + \bar{A} \cdot B + B \cdot B && \text{(Distributivité } L_6) \\
 &= 0 + A \cdot B + \bar{A} \cdot B + B && \text{(} T_7, T_5) \\
 &= B + B && \text{(} L_9 : \text{ Expansion)} \\
 &= B && \text{(} T_6 : \text{ Idempotence)}
 \end{aligned}$$

V.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 $F = (A \cdot B) + \bar{C}$. 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

VI. 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.

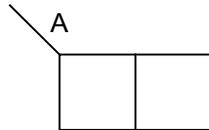
VI.1 Transposition d'une équation logique dans un diagramme de Karnaugh

a) Diagramme de Karnaugh

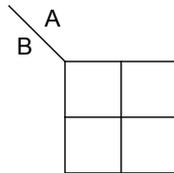
C'est un diagramme qui reprend les indications de la table de vérité pour les mettre sous une autre forme. Le nombre de cases est égal au nombre de lignes de la table de vérité, ou encore au nombre de combinaisons des variables d'entrée.

Exemples :

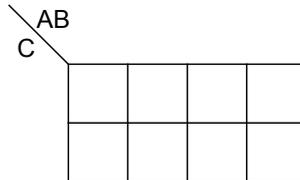
a) 1 variable d'entrée A \Rightarrow 2^1 combinaisons = 2 cases.



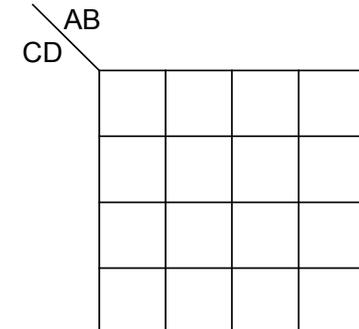
b) 2 variables d'entrée A et B \Rightarrow 2^2 combinaisons = 4 cases.



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



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



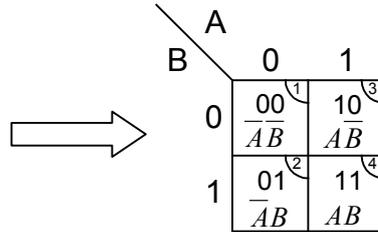
b) Disposition des combinaisons à l'intérieur du diagramme de Karnaugh

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 :

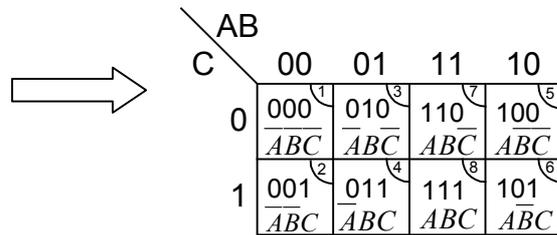
1.

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



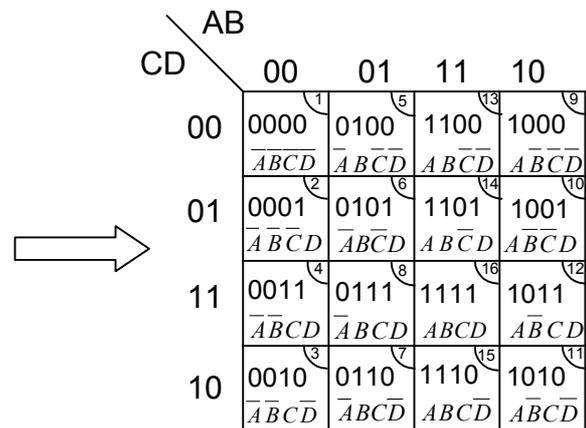
2.

	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

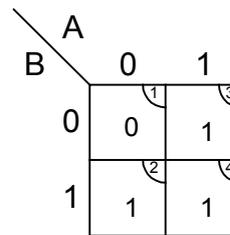
Exemples :

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

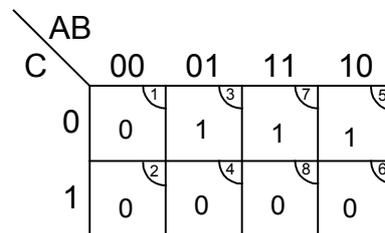


b) Soit l'équation : $S=A \cdot \bar{C} + B \cdot \bar{C}$

- 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

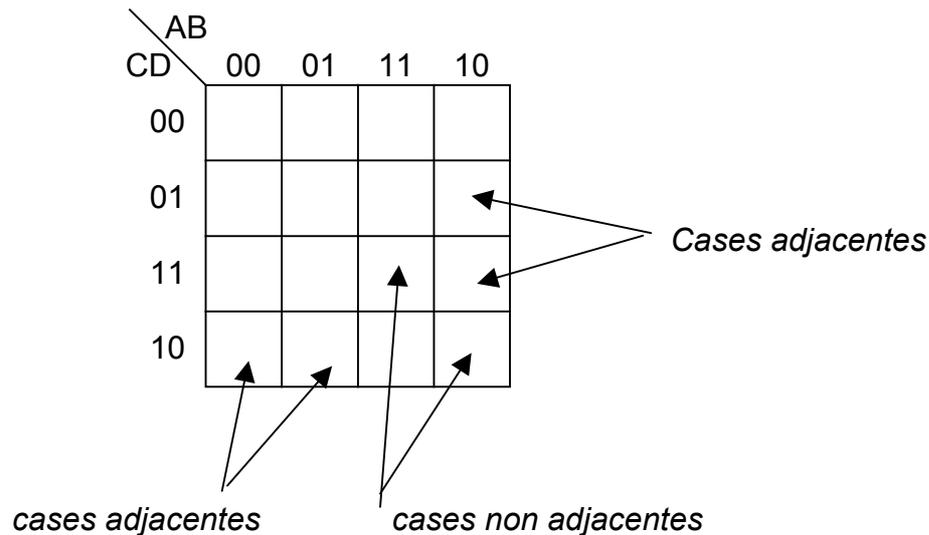


VI.2 Simplification d'une équation par le diagramme de Karnaugh

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 :



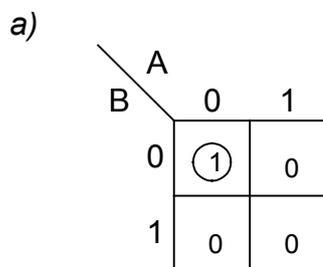
b) Règles de regroupement :

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

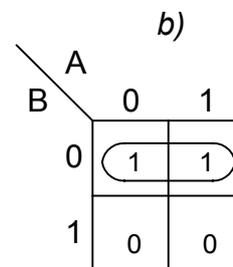
Règle 1 :

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

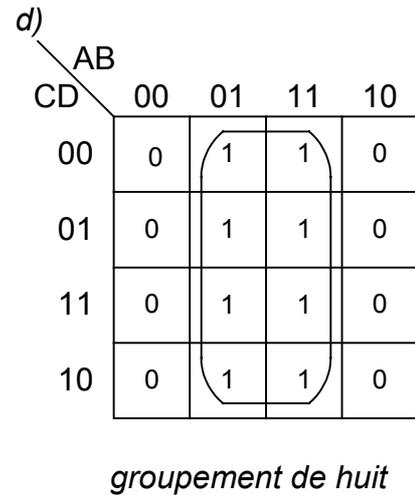
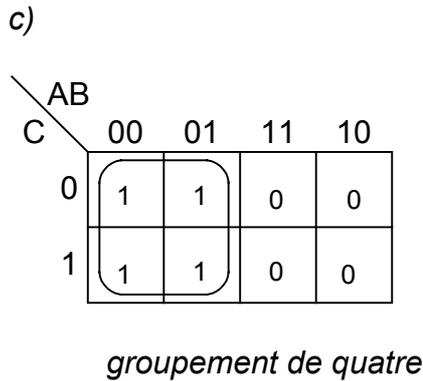
Exemples :



case unique



groupement de deux



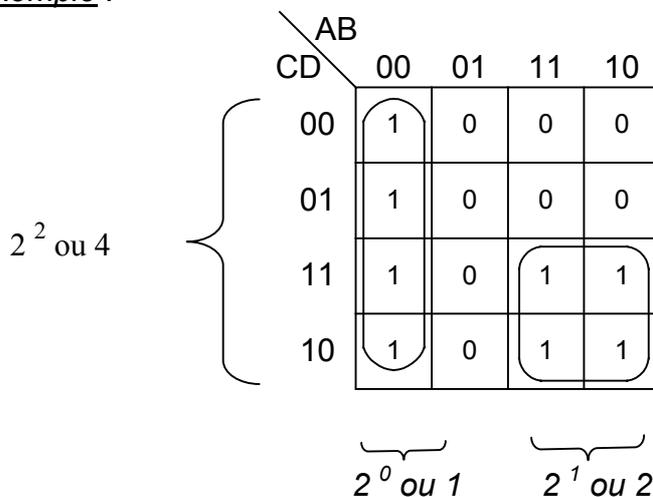
Règle 2 :

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 :



Règles 4 :

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

Exemples :

A faire

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

A ne pas faire

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

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)

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

b)

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

Règle 6 :

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

Exemples :

A faire

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

A ne pas faire

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

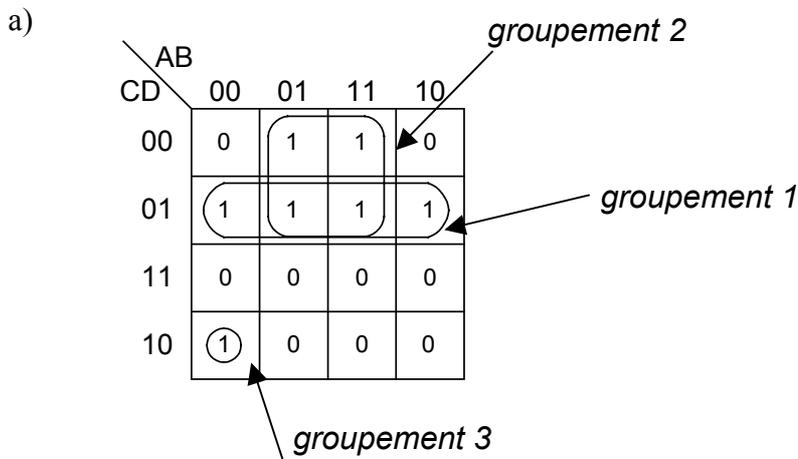
VI.3 Écriture des équations à partir de regroupement

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 : $B = 1 \implies$ On la représente par B ;
 $B = 0 \implies$ On la représente par \bar{B} .

Exemples :



Groupement 1 : A et B changent d'état

} \implies L'équation du groupement :
 $\bar{C} \cdot D$

$C = 0$ et $D = 1$ ne changent pas d'état

Groupement 2 : A et D changent d'état

} \implies L'équation du groupement :
 $B \cdot \bar{C}$

$B = 1$ et $C = 0$ ne changent pas d'état

Groupement 3 : $A = 0, B = 0, C = 1$ et $D = 0$
 ne change pas d'état

} \implies L'équation du groupement :
 $\bar{A} \cdot \bar{B} \cdot C \cdot \bar{D}$

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

b)

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

groupement 1 (vertical group in column 3)
 groupement 2 (horizontal group in row 3)
 groupement 3 (vertical group in column 3)

Groupement 1 : $A \cdot D$

Groupement 2 : $C \cdot D$

Groupement 3 : $A \cdot B$

$$\left. \begin{array}{l} \text{Groupement 1 : } A \cdot D \\ \text{Groupement 2 : } C \cdot D \\ \text{Groupement 3 : } A \cdot B \end{array} \right\} \Rightarrow S = A \cdot D + C \cdot D + A \cdot B$$

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 : $B = 1 \Rightarrow$ On la représente par \bar{B} ;

$B = 0 \Rightarrow$ On le représente par B .

Exemples :

a)

	AB			
CD	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 (vertical group in column 1)
 groupement 2 (vertical group in column 3)

Groupement 1 : A, B ne changent pas d'état ($A = 0, B = 0$)

C et D changent d'état

\Rightarrow L'équation du groupement : $A+B$

Groupement 2 : A, B ne changent pas l'état ($A = 1, B = 1$)

C et D changent

\Rightarrow L'équation du groupement : $\bar{A}+\bar{B}$

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

b)

		AB			
		00	01	11	10
CD	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 :
 $\overline{B+C}+D$

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

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 1^{er} type comprend des opérateurs NON, ET, OU;
- Le 2^{ème} type ne comprend que des opérateurs NON ET (NAND);
- Le 3^{ème} 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

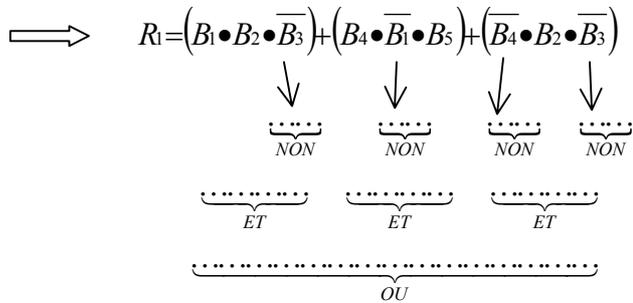
Pour traduire une équation en schéma logique avec ces opérateurs, il faut :

- 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 :

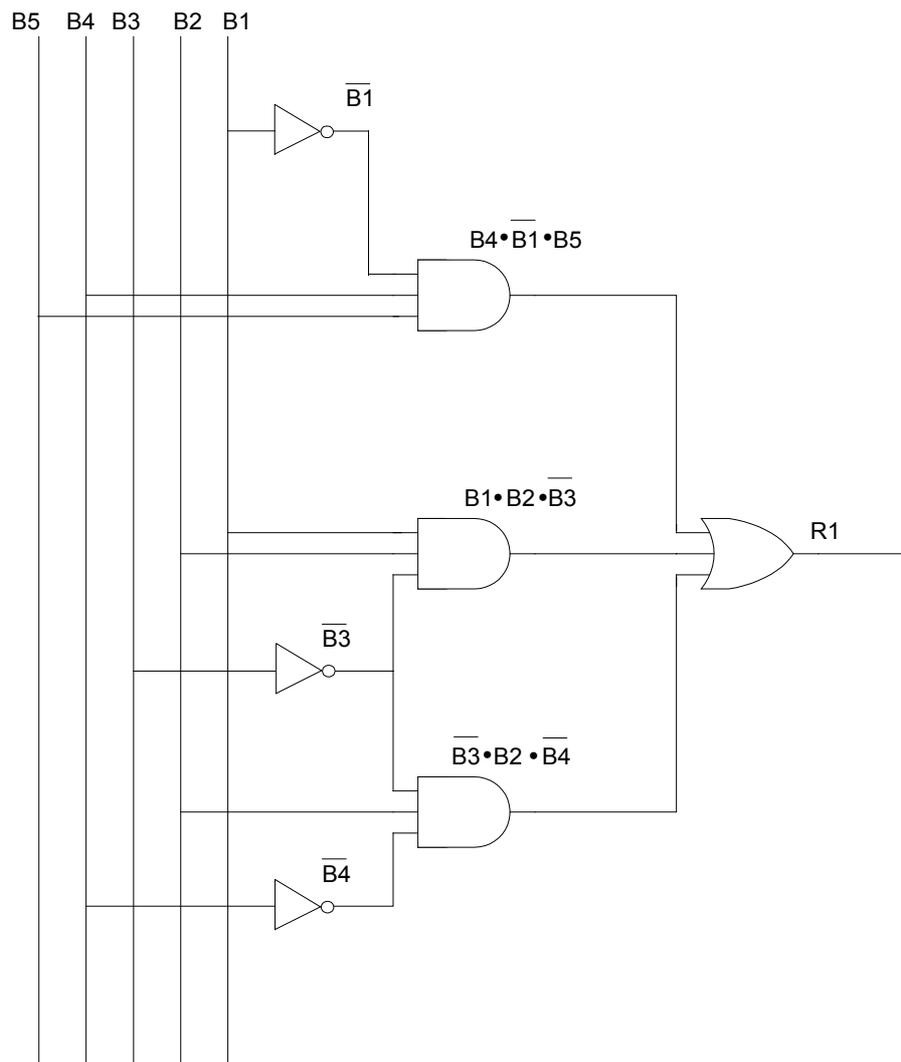
a) $R_1 = B_1 \bullet B_2 \bullet \overline{B_3} + B_4 \bullet \overline{B_1} \bullet B_5 + \overline{B_4} \bullet B_2 \bullet \overline{B_3}$

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



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 :

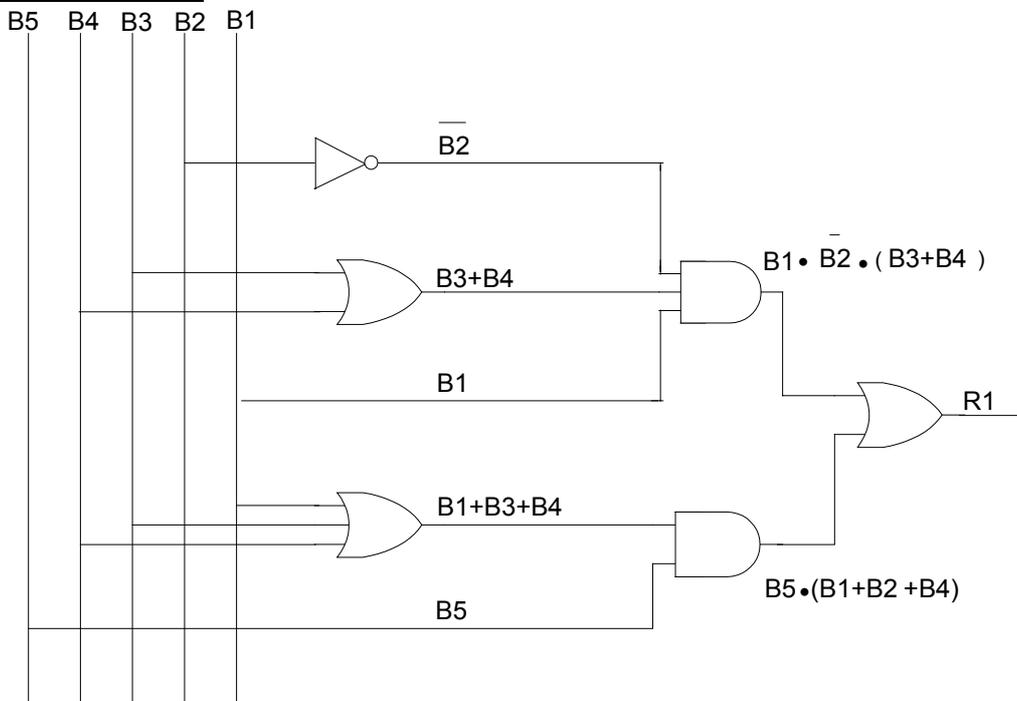


$$b) R_1 = B_1 \cdot \overline{B_2} \cdot (B_3 + B_4) + B_5 \cdot (B_3 + B_4 + B_1)$$

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

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

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 \Rightarrow transformer l'équation en appliquant les théorèmes De Morgan.
- L'équation doit être entièrement recouverte par une barre \Rightarrow utiliser les propriétés de la négation $\overline{\overline{S}} = S$.

Exemples :

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

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

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

Il y a huit barres \implies 8 opérateurs NON ET.

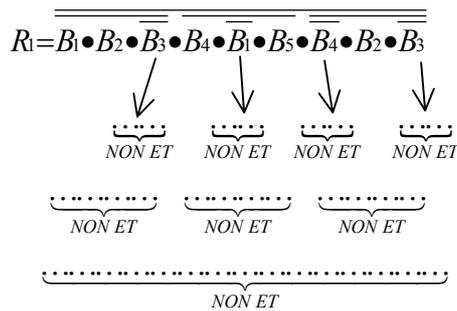
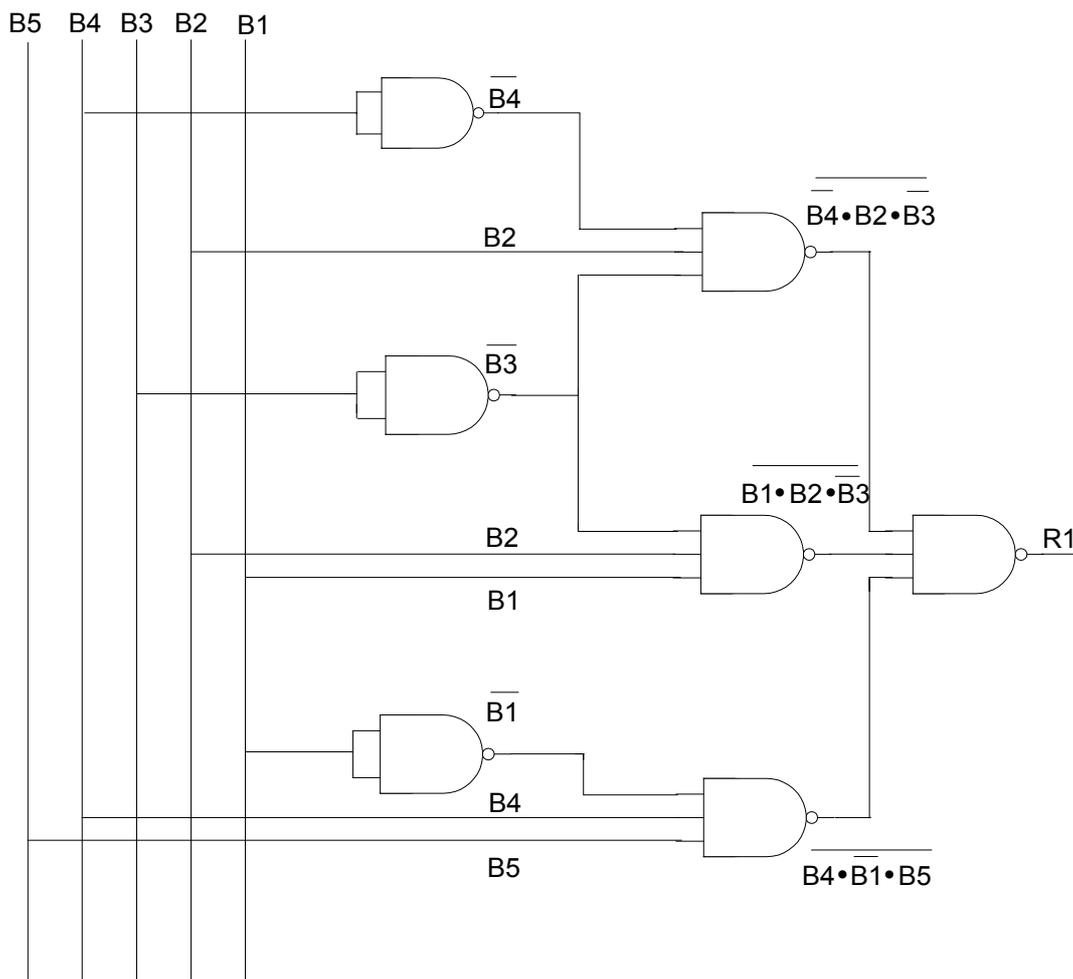


Schéma logique



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

Transformation :

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

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

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

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

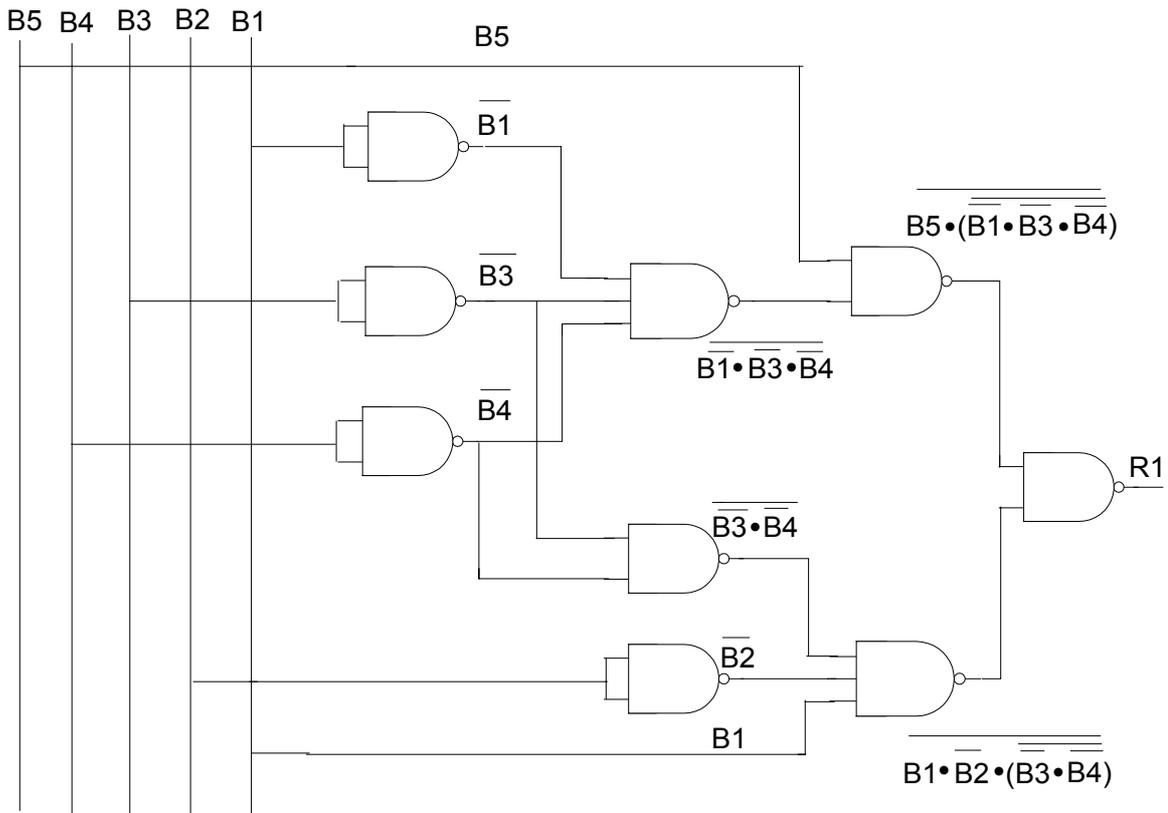
\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow
 NON ET NON ET NON ET NON ET NON ET NON ET NON ET

$\underbrace{\hspace{10em}}_{NON ET}$ $\underbrace{\hspace{10em}}_{NON ET}$

$\underbrace{\hspace{20em}}_{NON ET}$ $\underbrace{\hspace{20em}}_{NON ET}$

$\underbrace{\hspace{40em}}_{NON ET}$

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 \implies transformer l'équation en appliquant les théorèmes de Morgan.
- l'équation doit être entièrement recouverte par une barre \implies utiliser les propriétés de la négation : $S = \overline{\overline{S}}$

Exemples :

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

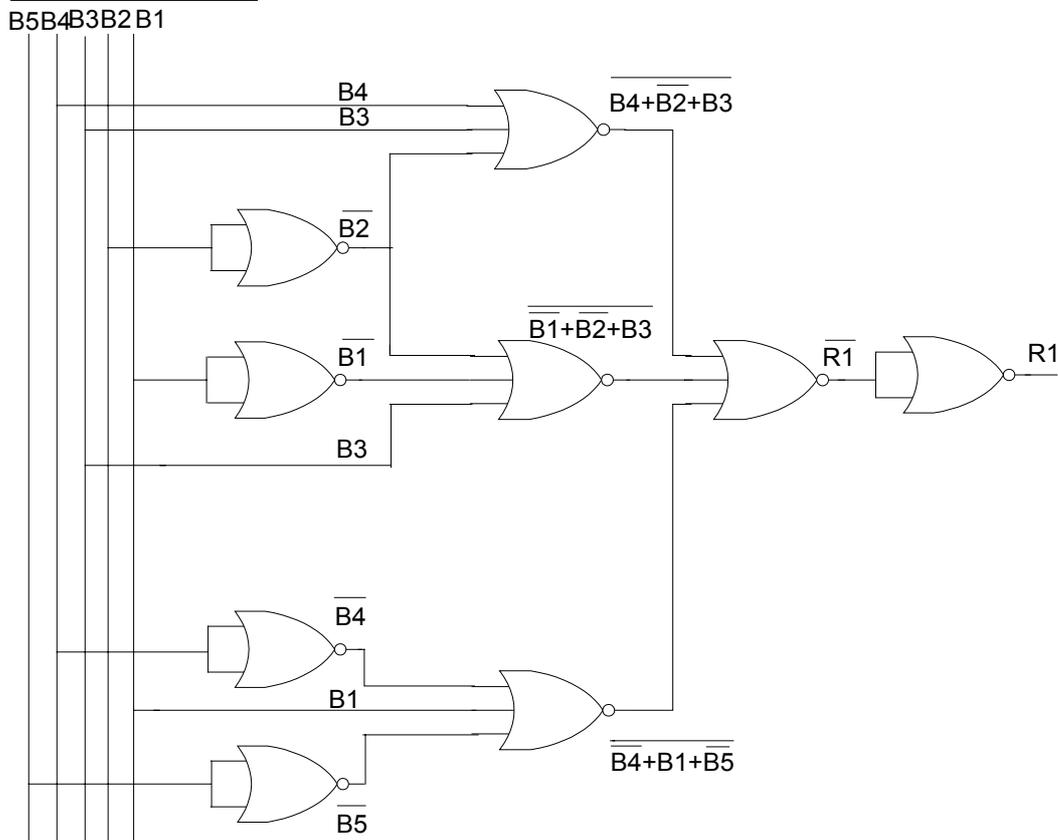
$$\overline{\overline{R_1}} = \overline{\overline{B_1 \bullet B_2 \bullet \overline{B_3} + B_4 \bullet \overline{B_1} \bullet B_5 + \overline{B_5} \bullet B_2 \bullet \overline{B_3}}}$$

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

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

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

Schéma logique



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

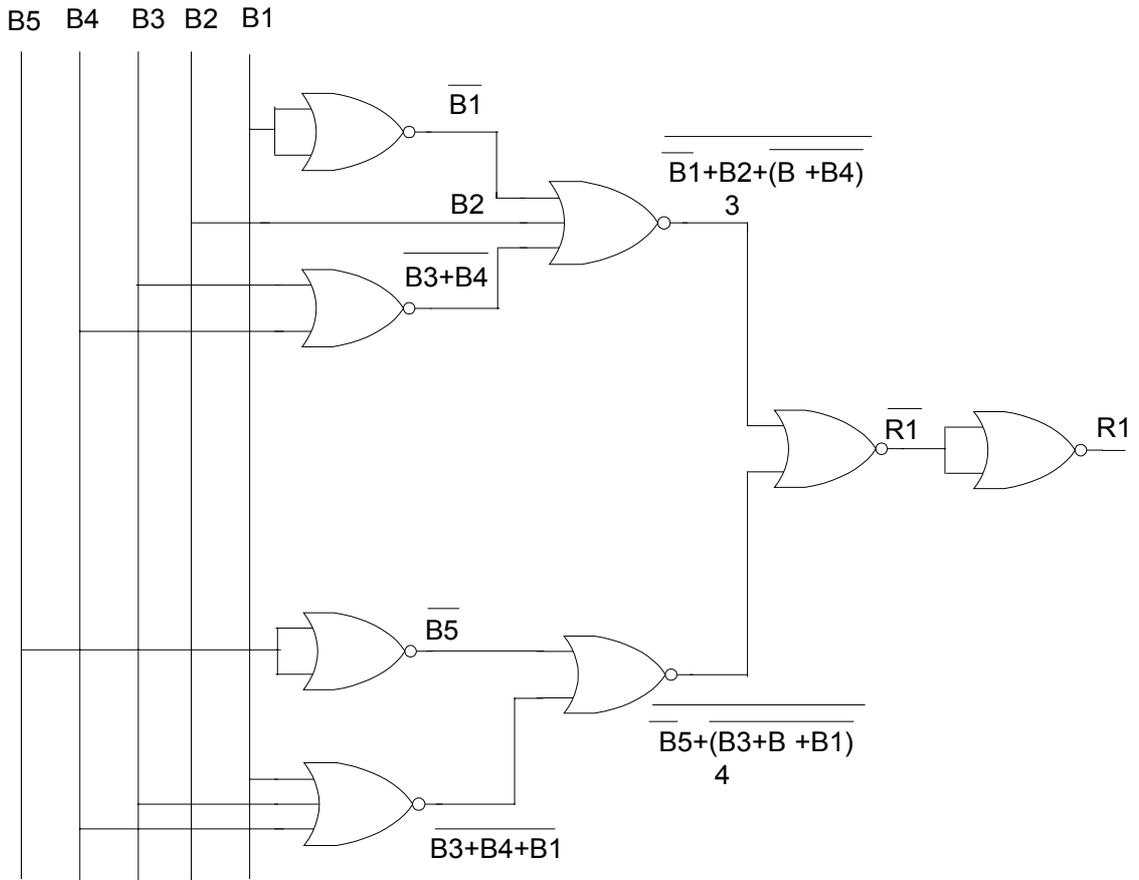
Transformation :

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

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

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

Schéma logique



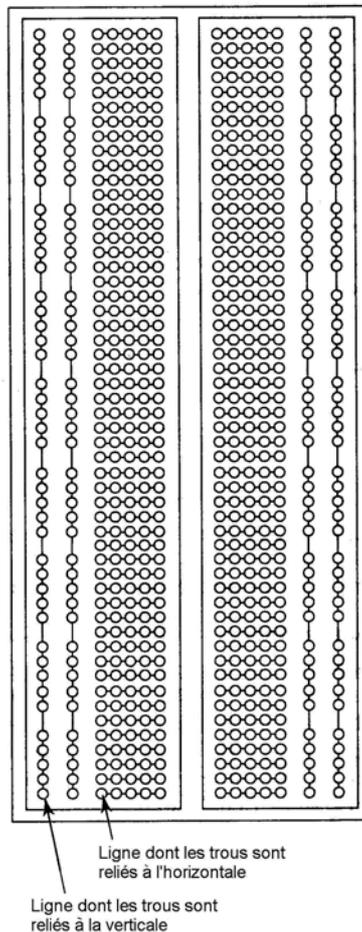
VIII. Montage des circuits

VIII.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.

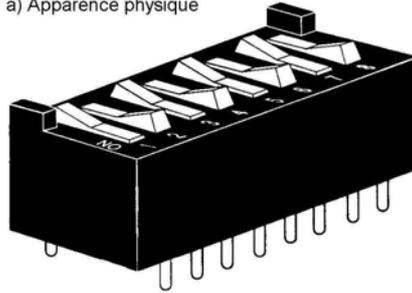
Cette plaque possède des trous perforés interreliés qui forment des lignes indépendantes. Les lignes verticales sont généralement employées pour les sources (V_{CC} pour les circuits intégrés TTL et V_{DD} - V_{SS} pour les C-I.C-MOS), tandis que les lignes horizontales servent au branchement des divers composants.



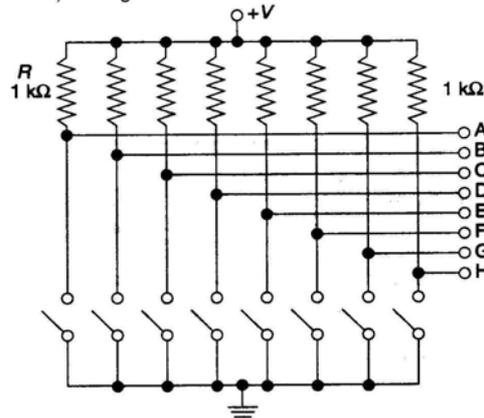
VIII.2 Interrupteurs logiques

La simulation des entrées se fait à l'aide d'un groupe d'interrupteurs miniatures (DIP switch) (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



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

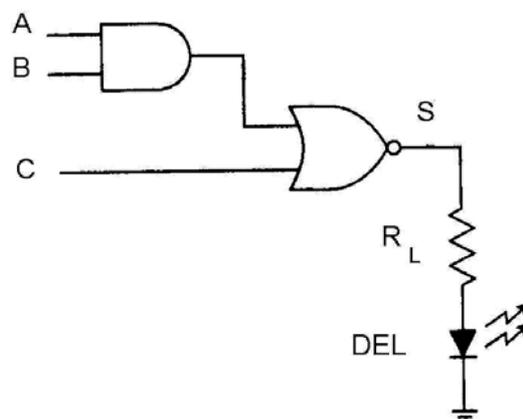
La forme que prend la sortie d'un circuit permet de faire la différence entre les logiques positive et négative.

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 :

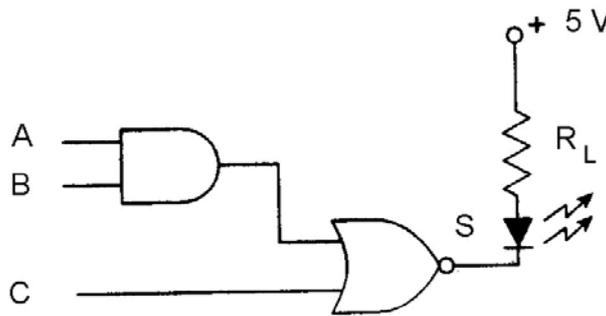


Mais dans ce cas la sortie débite un courant de charge élevé \Rightarrow logique peu recommandée

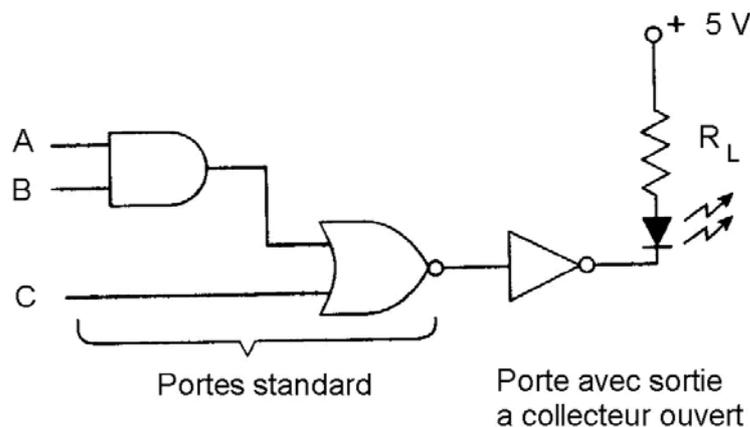
b) Logique négative

Un circuit est en logique négative lorsque la sortie est reliée à la source (V_{cc}) 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

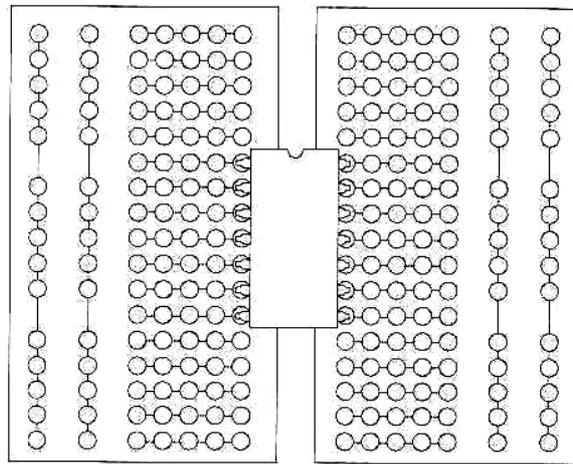
La résistance R_L limite le courant qui passe par la DEL afin d'éviter d'endommager le circuit :

$$R_{L \min} = \frac{\text{Tension aux bornes de } R_L}{\text{Courant maximal des portes logiques}} = \frac{5V - 1,5V}{16 \text{ mA}} = 220\Omega$$

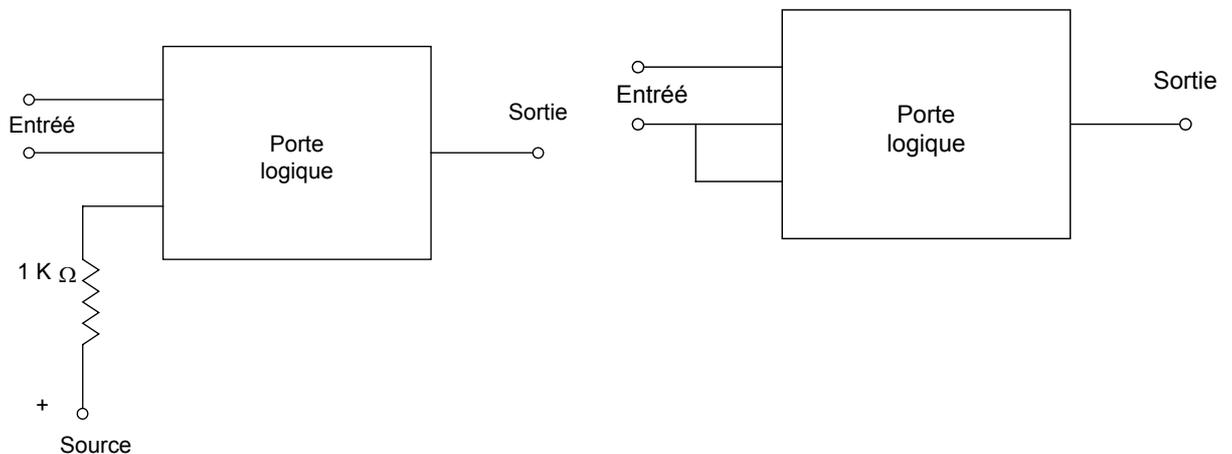
$1,5V = \Delta U$ aux bornes de la DEL.

VIII.4 Technique de travail

- Lorsqu'on utilise des C.I. (circuits intégrés), 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 $1K\Omega$ (le meilleur choix), on les branche ensemble pour n'en faire qu'une (voir figure suivante).



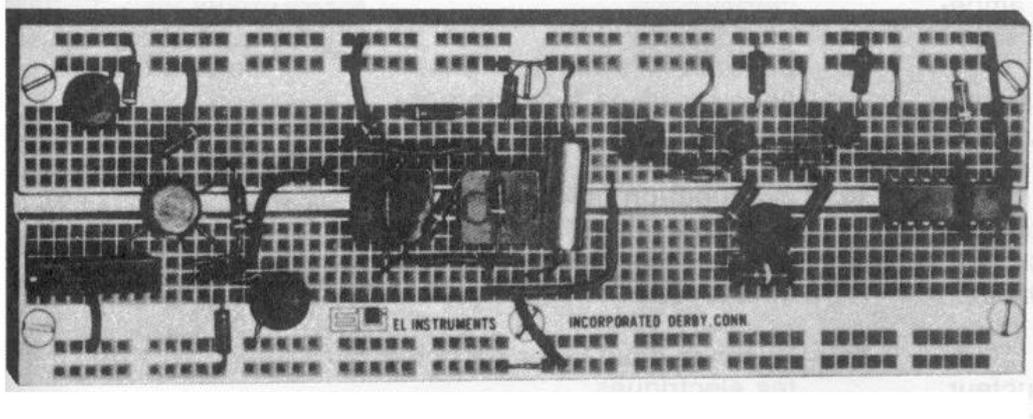
*Branchement à la source
(assure un niveau 1)*

*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. Circuits arithmétiques

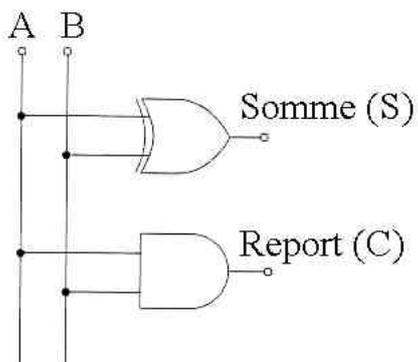
IX.1 Additionneur

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

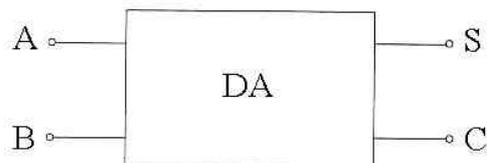
a) Demi-Additionneur

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

- Schéma logique

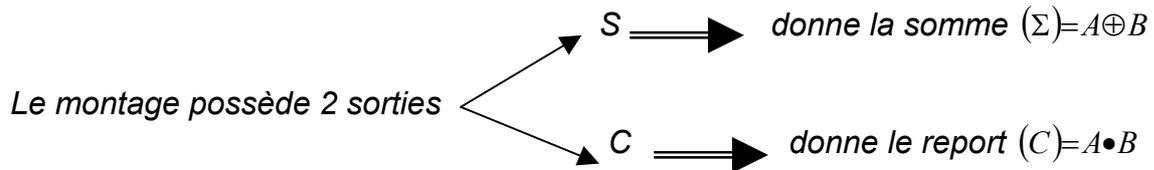


- Représentation simplifiée

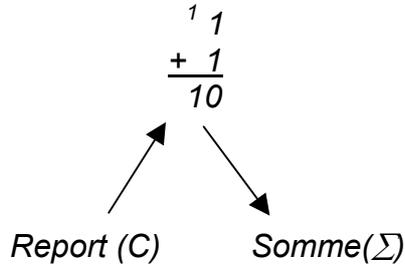
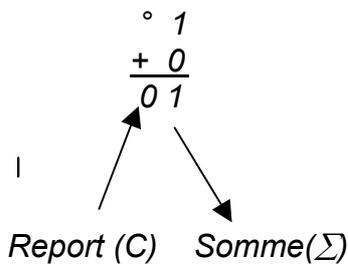


- Table de vérité

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



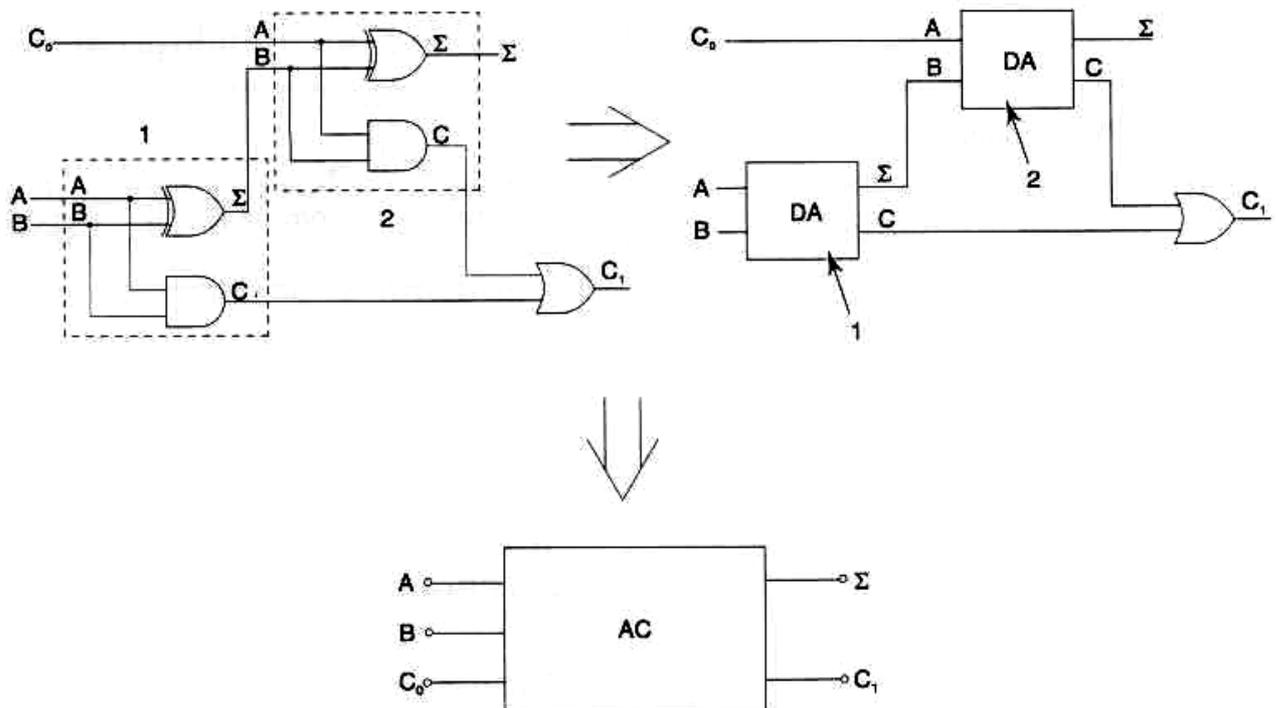
Exemples d'addition :



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).

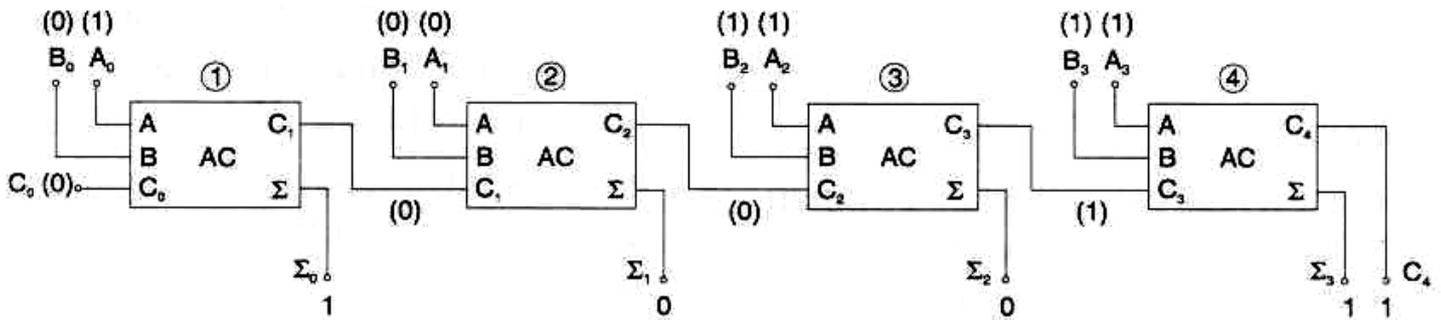


- Table de vérité :

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

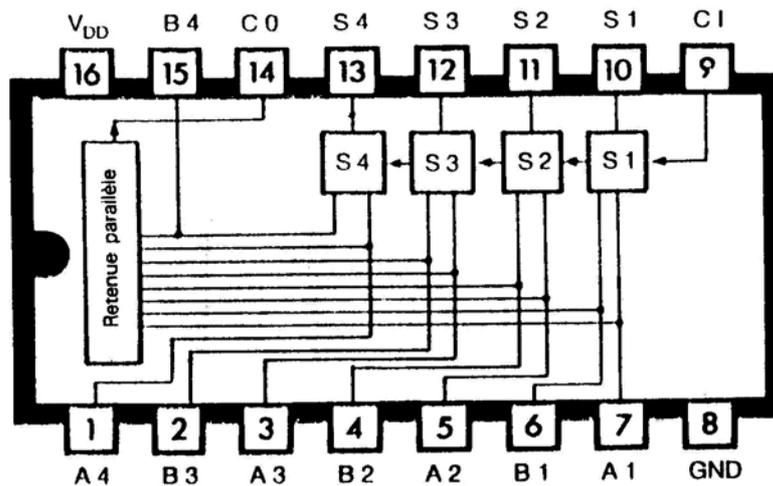
Il est possible de refaire le même travail en combinant plusieurs AC et en utilisant le report du premier étage en guise d'entrée «C» du second et ainsi de suite. La figure suivante traite l'addition de deux nombres de 4 bits en utilisant des AC.

A = 1101
B = 1100

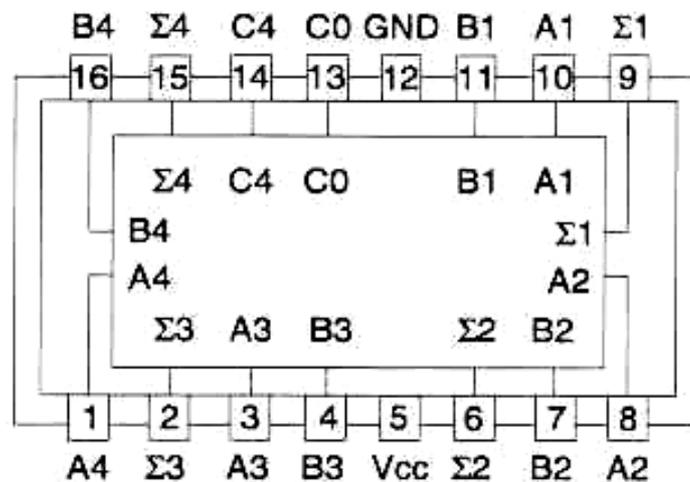


Le résultat de l'addition est de 1 1 0 0 1
 $C_4 \Sigma_3 \Sigma_2 \Sigma_1 \Sigma_0$

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)

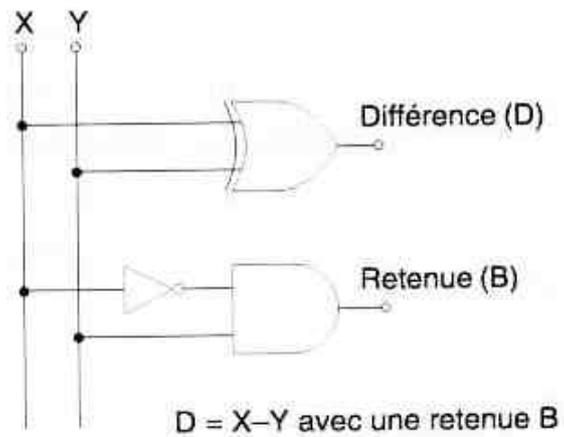
IX.2 Soustracteurs logiques

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

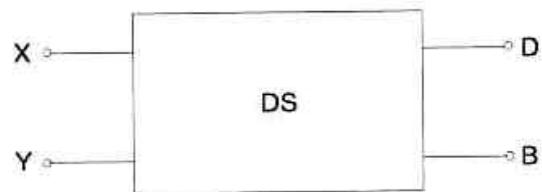
a) Demi-soustracteur

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

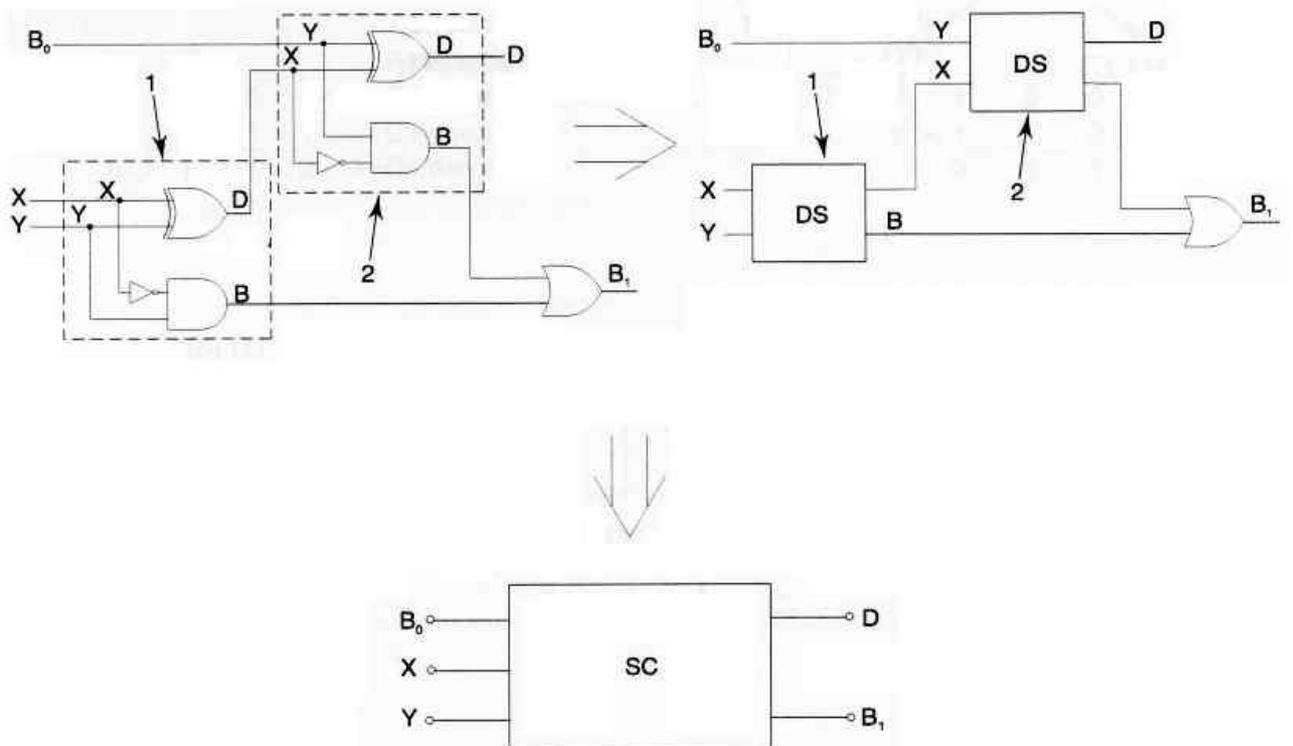
Entrées		Sorties	
X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

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

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

b) Soustracteurs complets

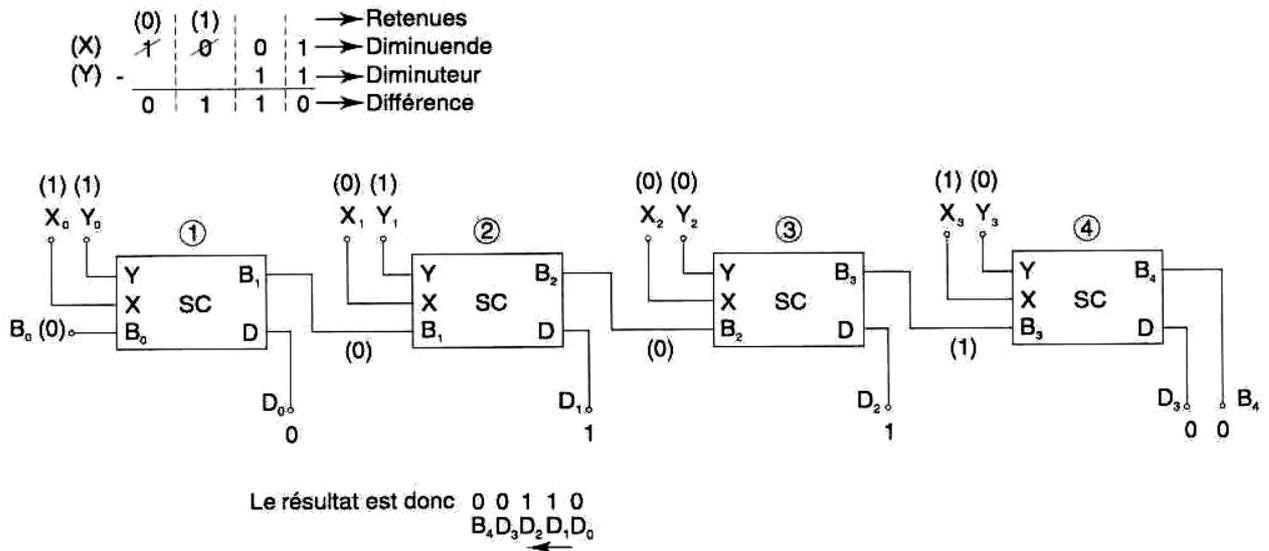
Le soustracteur complet est une combinaison de deux DS auquel on a ajouté un «OU» logique à la sortie de la retenue (voir figure suivante).



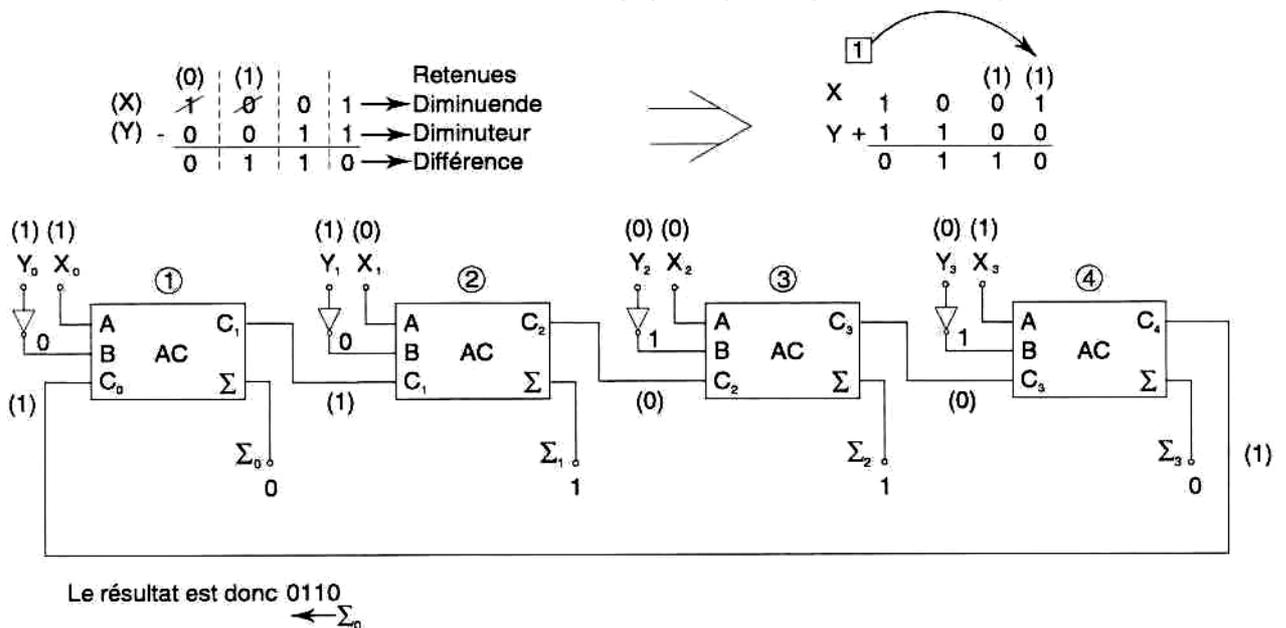
- Table de vérité

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).



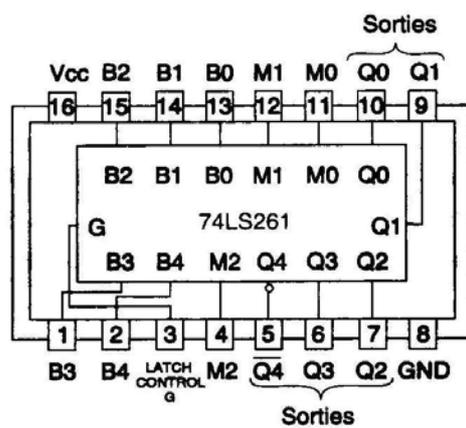
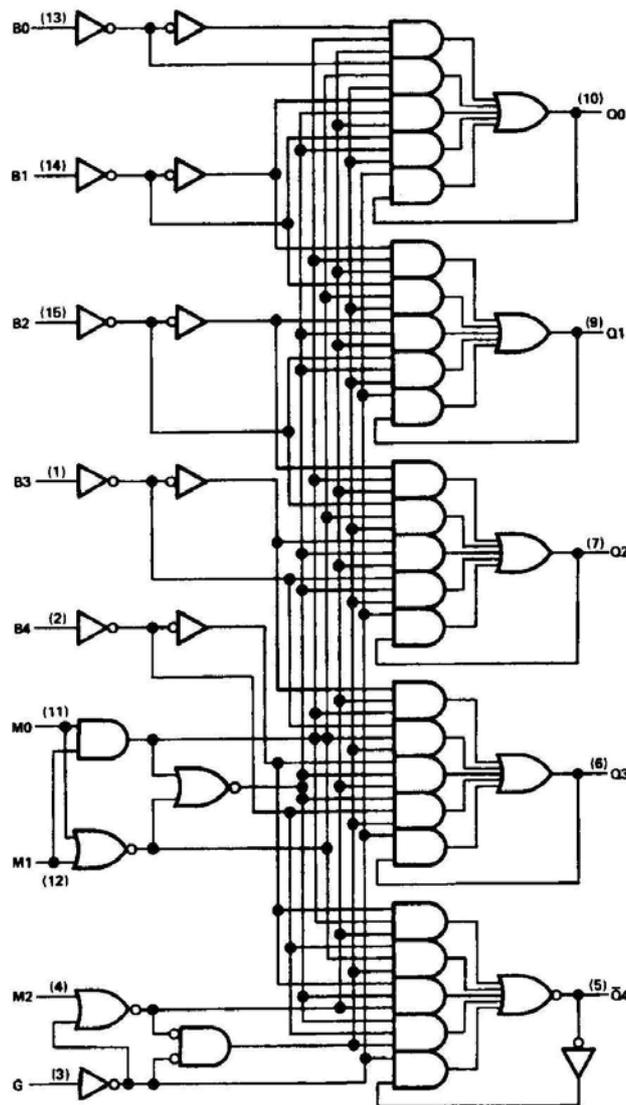
Malheureusement, on ne trouve pas des C.I. pour les soustracteurs. Il est cependant possible d'utiliser un artifice mathématique pour transformer la soustraction en addition et ainsi utiliser des additionneurs logiques (voir figure suivante).



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

IX.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 énumère 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

IX.4 Circuit comparateur

Ces circuits permettent de comparer deux mots de n bits chacun en indiquant sur leurs sorties si le premier mot est plus grand, plus petit ou égal au second.

Le principe consiste à comparer d'abord les bits les plus significatifs. S'ils sont différents, il est inutile de continuer la comparaison. Par contre s'ils sont égaux, il faut comparer les bits de poids immédiatement inférieur et ainsi de suite.

Exemple : comparaison de nombres à un bit

Tableau d'analyse.

a	b	E	S	I
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

$$\text{Pour } A = B : E = a b + \bar{a} \bar{b} = \overline{a \oplus b}$$

$$\text{Pour } A > B : S = a \bar{b}$$

$$\text{Pour } A < B : I = \bar{a} b$$

$$E = \overline{S+I} = \overline{a \cdot \bar{b} + \bar{a} \cdot b} = \overline{a \oplus b}$$

Remarque : On peut donc réaliser un comparateur à l'aide de circuits logiques.

Circuits intégrés assurant cette fonction :

- TTL : 7485, 74680, 74682, 74683.....
- CMOS : 4063, 4585.....

Le circuit intégré TTL 7485 permet de comparer deux mots de 4 bits (A3 à A0 et B3 à B0).

Ce circuit possède trois sorties :

- A supérieur à B : $Q_{A>B}$
- A inférieur à B : $Q_{A<B}$
- A égal à B : $O_{A=B}$

Trois entrée d'extension ($I_{A>B}$, $I_{A<B}$, $I_{A=B}$) permettent la mise en cascade de plusieurs circuits afin d'effectuer une comparaison sur des mots plus grands.

X. Circuits de décodage des informations

X.1 Le codeur :

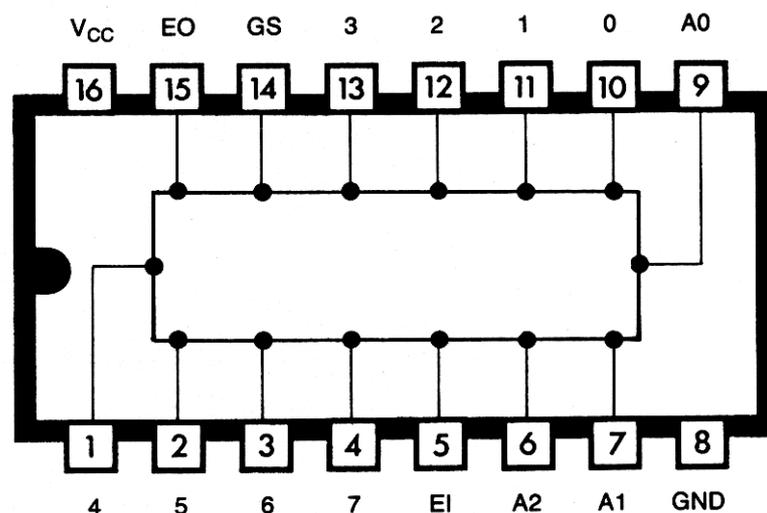
C'est un circuit à N entrées dont une seulement est active et qui délivre sur n sorties (en code binaire ou autre) le numéro de l'entrée.

Exemple :

On prend comme exemple le codeur 74148 (voir figure suivante). Cette élément décode 8 lignes d'entrée en 3 lignes de sortie (code binaire). Pour le montage en cascade de plusieurs éléments, la sortie EO du premier élément est reliée à l'entrée EI du deuxième élément et ainsi de suite. Il n'est pas nécessaire d'utiliser des éléments supplémentaires.

Lorsque toutes les lignes d'entrée se trouvent au niveau H, les sorties ont également le niveau H. En état actif, tous les niveaux d'entrée et de sortie sont à l'état bas.

Ce codeur est un codeur de priorité car il permet le choix du code correspondant au chiffre décimal le plus élevé des entrées activées.



Entrées									Sorties				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

X = sans importance

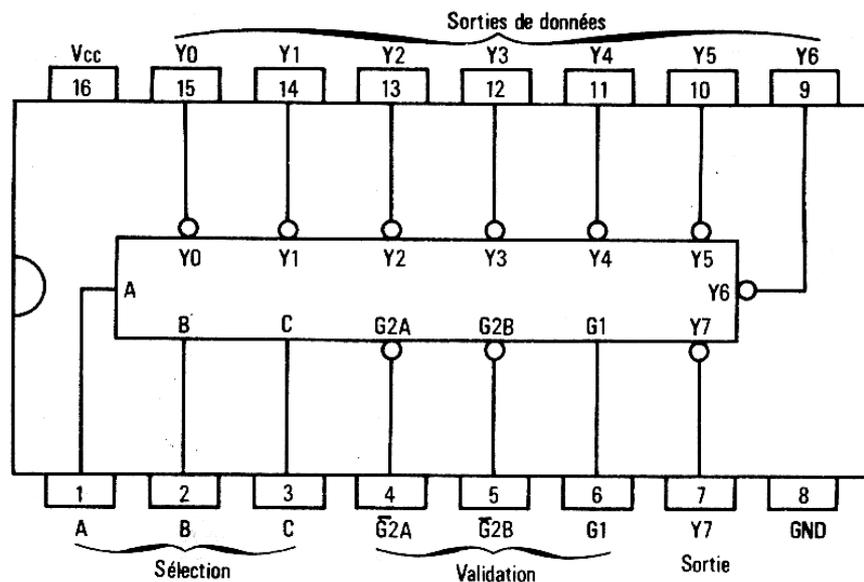
Caractéristiques du C I 74148.

X.2 Le décodeur.

Le décodage consiste à sélectionner une sortie ou un groupe de sortie en fonction d'un code d'entrée. On dit que le code d'entrée identifie l'adresse de la sortie.

Exemple : Le C I 74138 (voir figure suivante)

Ces circuits sont des décodeurs de trois lignes en huit lignes. Ce décodeur est en fonction seulement lorsque les entrées $G2A = G2B = 0$ et que $G1 = 1$. Autrement les sorties du décodeur sont paralysées, peu importe le code d'entrée. Ces entrées de validation réduisent l'emploi de portes externes et d'inverseurs.



Entrées					Sorties							
Validation		Sélection			Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
G1	G2*	C	B	A								
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	L	H	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

* $G2 = G2A + G2B$ H = niveau haut L = niveau bas
X = sans importance

Caractéristiques du C I 74138.

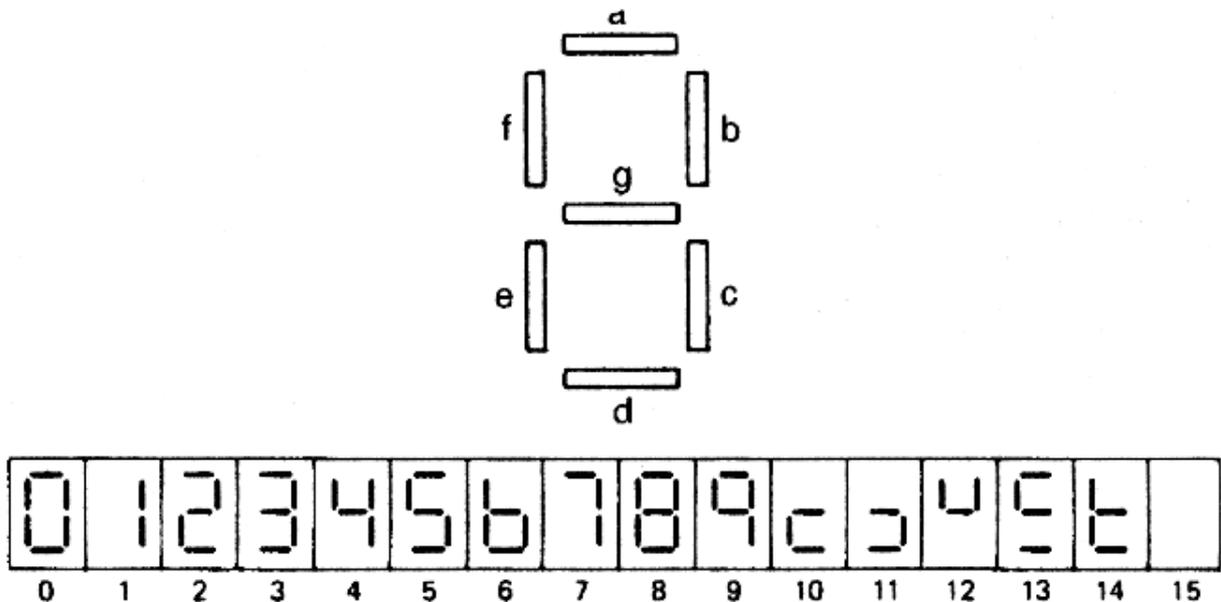
➤ **Le décodeur pilote d'affichage :**

Plusieurs dispositifs de décodage servent à la représentation de données à l'aide d'un afficheur décimal. C'est le cas des afficheurs à sept segments.

➤ **Afficheur DCB à sept segments :**

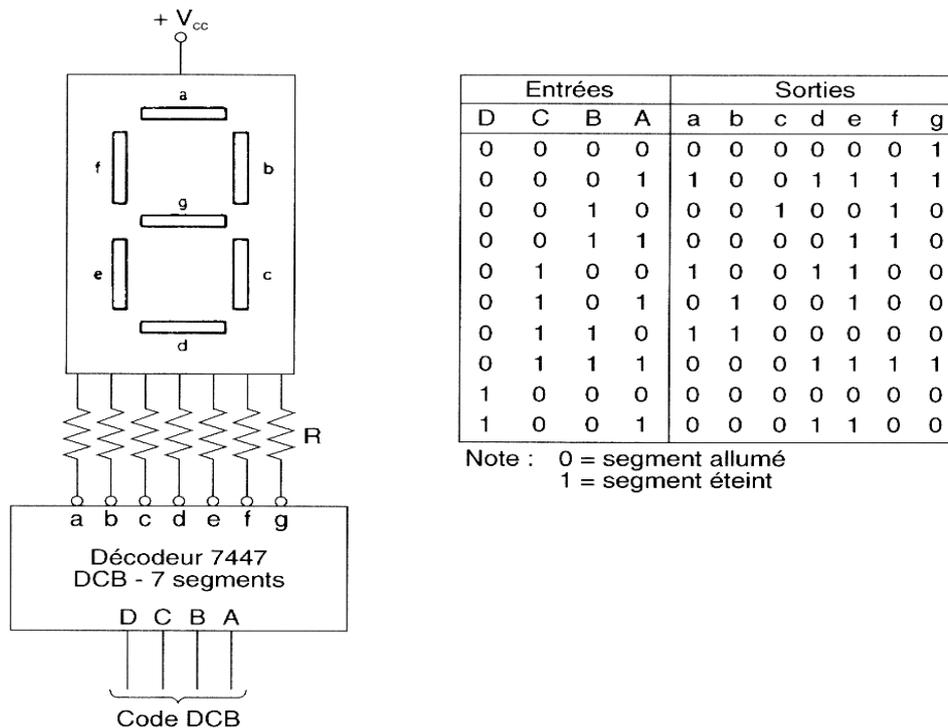
L'appellation « sept segments » provient de l'agencement de sept DEL de manière à pouvoir afficher les dix chiffres de 0 jusqu'à 9.

La particularité des décodeurs pilotes d'affichage vient du fait que, pour un code d'entrée distinct, plusieurs sorties deviennent actives simultanément afin d'afficher les différents caractères. Un décodeur pilote d'affichage est donc conçu pour être relié directement à un afficheur. La figure suivante montre la disposition des segments d'un afficheur ainsi que les seize combinaisons possibles obtenues avec la plupart des décodeurs.



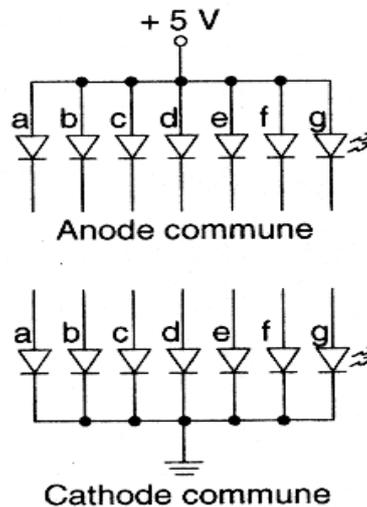
Identification et disposition des segments

La figure ci-dessous montre le raccordement d'un afficheur ainsi que la table de vérité du décodeur. Comme le décodeur 7447 est doté de sorties vraies au niveau BAS, les «0» présents aux sorties de la table de vérité se traduisent par des segments allumés. De plus, l'anode de chacun des segments est reliée à une borne commune connectée à l'alimentation Vcc. On décrit cet afficheur à DEL comme étant à anode commune.



Raccordement d'un afficheur à anode commune

Il existe aussi des afficheurs à DEL à cathode commune. La figure suivante montre le raccordement interne des DEL de chacun des types d'afficheurs. Pour piloter un afficheur à cathode commune, le décodeur doit être doté de sorties vraies au niveau HAUT, car la borne commune de l'afficheur se relie à la masse du circuit.



Les types d'afficheurs

X.3 Circuits de transcodage

Ces circuits permettent de convertir un code binaire en un autre code binaire (exemple : binaire naturel en code GRAY ou inversement). Ils sont d'usage peu courant et en général, réalisés avec des mémoires mortes (ROM).

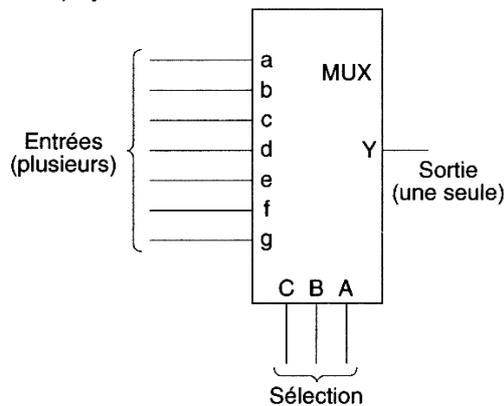
XI. Circuits de transfert d'information

XI.1 Multiplexeur

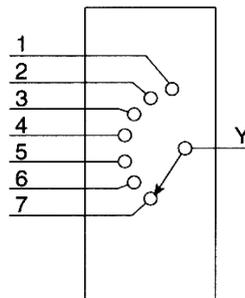
Un multiplexeur agit comme un interrupteur à pôles multiples. Il sélectionne une des entrées pour la faire apparaître à la sortie. C'est la raison pour laquelle on l'appelle parfois sélecteur de données. L'aiguillage de l'entrée des données s'effectue par des lignes de sélection qui répondent à un code binaire. La figure suivante montre un multiplexeur à huit entrées. Le code d'entrée appliqué sur les lignes de sélection C B A détermine l'entrée qui sera transmise à la sortie.

La table de vérité de cette figure prouve la transparence d'un multiplexeur lorsqu'une entrée est activée. Peu importe le contenu de l'entrée sélectionnée, l'information apparaît intégralement à la sortie.

a) Symbole



b) Équivalence



c) Table de vérité

Lignes de sélection			Entrées								Sortie
C	B	A	0	1	2	3	4	5	6	7	Y
0	0	0	0	x	x	x	x	x	x	x	0
0	0	1	1	x	x	x	x	x	x	x	1
0	1	0	x	0	x	x	x	x	x	x	0
0	1	1	x	1	x	x	x	x	x	x	1
	0						0				0
	0						0				0
	0						0				0
1	1	1	x	x	x	x	x	x	x	0	0
1	1	1	x	x	x	x	x	x	x	1	1

Multiplexeur à huit entrées

Exemple d'un multiplexeur :

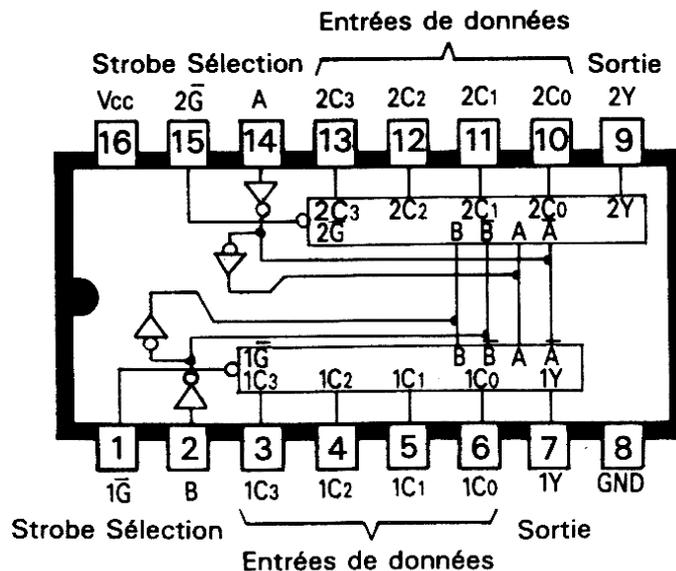
La figure suivante montre le composant 74153. Ce circuit comporte deux sélecteurs/multiplexeurs de données avec chacun un décodage binaire permettant de sélectionner une ligne de données entre quatre. Il a une entrée de commande de strobe pour chaque multiplexeur, qui doit être au niveau bas pour valider le circuit; un niveau haut du strobe force la sortie y correspondante au niveau bas. La table de vérité précise le fonctionnement.

**Table de vérité
(pour une section)**

Entrées de sélection		Entrées de données				Strobe	Sortie
B	A	C0	C1	C2	C3	G	Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	B

Les entrées de sélection sont communes aux deux sections
H = niveau haut L = niveau bas X = sans importance

Brochage



Caractéristiques du 74153.

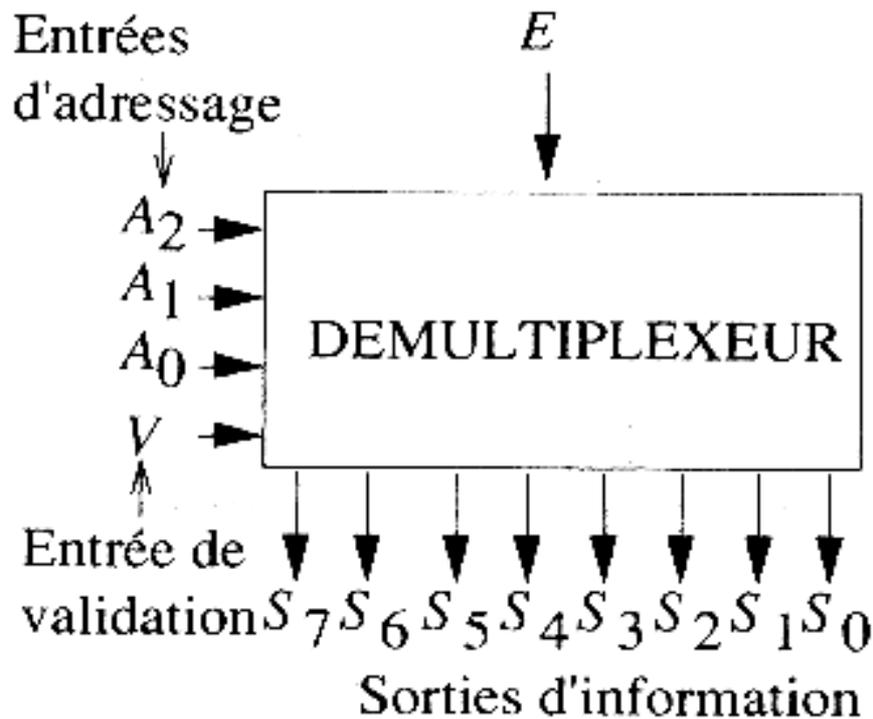
XI.2 Demultiplexeur

Un démultiplexeur est un circuit logique à une entrée de données ou d'informations et N sorties qui reçoivent les informations d'entrée (voir figure ci-dessous). Pour sélectionner la sortie qui doit être active, le démultiplexeur reçoit un ordre de n entrées d'adresse, comme dans le cas d'un multiplexeur. Ces entrées orientent les

données d'entrée vers la sortie sélectionnée. Les nombres N et n sont liés par la relation : $N = 2^n$

Certains démultiplexeurs présentent une entrée de validation qui permet :

- Lorsqu'elle est mise à l'état 0, d'imposer un zéro à l'ensemble des sorties quelle que soit la commande.
- Lorsqu'elle est mise à l'état 1, d'autoriser la transmission des données de l'entrée vers les sorties successivement sélectionnées.



Représentation d'un démultiplexeur à 8 entrées d'informations et 3 entrées d'adressage

Intérêt :

- Un démultiplexeur joue le rôle d'un commutateur à plusieurs positions qui oriente les informations d'entrée vers chaque sortie choisie.
- Les entrées d'adressage permettent de connaître à chaque instant l'état d'une sortie.
- Par une utilisation cyclique des entrées d'adressage, les données des entrées, qui parviennent en série à l'entrée du démultiplexeur, sont fournies en parallèle à la sortie.

**MODULE N° 5: MAINTENANCE D'UN SYSTEME
COMBINATOIRE**
GUIDE DES EXERCICES ET TRAVAUX PRATIQUES

Exercices :

Exercices 1 :

Réduire les fonctions suivantes en utilisant l'algèbre de Boole

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

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

$$F3 = A\bar{B} + A\bar{C} + \bar{B}\bar{C}$$

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

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

Exercice2 :

1) Donner l'équivalent décimal des nombres suivants :

$$(72)_8 = (\dots\dots\dots)_{10}$$

$$(1251)_8 = (\dots\dots\dots)_{10}$$

$$(17,3)_8 = (\dots\dots\dots)_{10}$$

$$(512,65)_8 = (\dots\dots\dots)_{10}$$

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

$$(8)_{10} = (\dots\dots\dots)_{BCD}$$

$$(17)_{10} = (\dots\dots\dots)_{BCD}$$

$$(128)_{10} = (\dots\dots\dots)_{BCD}$$

$$(92)_{10} = (\dots\dots\dots)_{BCD}$$

3) Effectuer les opérations arithmétiques suivantes :

$$\begin{array}{r} 0011 \\ + \\ \hline 1101 \end{array}$$

$$\begin{array}{r} 1101 \\ - \\ \hline 0010 \end{array}$$

$$\begin{array}{r} 110 \\ * \\ \hline 11 \end{array}$$

$$1111/11$$

Exercice3:

Construire les tables de vérité des équations suivantes :

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

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

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

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

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	01	11	10		00	01	11	10		00	01	11	10			
00	0	1	1	0		00	0	1	1	1		00	1	1	0	0		00	1	0	0	1
01	0	0	1	1		01	0	1	1	1		01	0	1	0	0		01	0	1	1	1
						11	0	1	1	0		11	1	1	0	0		11	0	1	1	1
						10	0	1	1	0		10	1	0	0	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).

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

- $S = A \cdot \overline{B} + \overline{A} \cdot \overline{B} + A \cdot B$

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

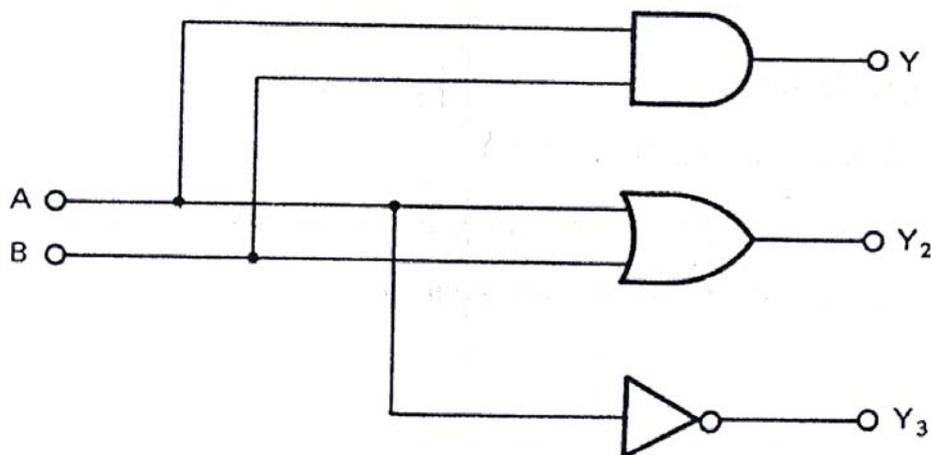
1 heure

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

- Alimentation continue (+ 5 V)
- Plaquette de montage ou simulateur de fonctions logiques
- Interrupteurs logiques
- Pincettes 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

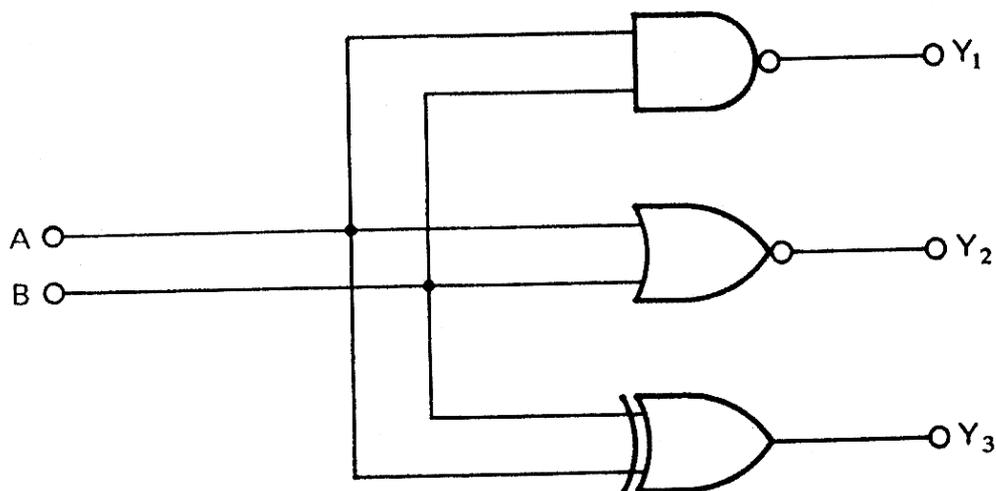
1 heure

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

- Alimentation continue (+ 5 V)
- Plaquette de montage ou simulateur de fonctions logiques
- Interrupteurs logiques
- Pincettes 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 en utilisant l'algèbre de Boole.

2. Durée du TP

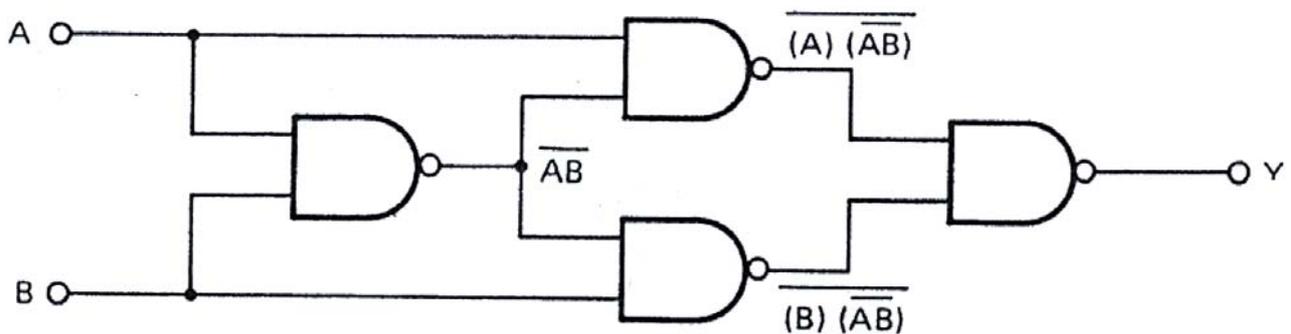
2 heures.

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

- Alimentation continue (+ 5 V)
- Plaquette de montage ou simulateur de fonctions logiques
- Interrupteurs logiques
- Pincettes et fils
- Circuits intégrés 7400
- 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 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

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

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:

2 heures

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

- Alimentation continue (+ 5 V)
- Plaquette de montage ou simulateur de fonctions logiques
- Interrupteurs logiques
- Pincettes 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 :

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

Questions :

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

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 CI 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:

2 heures.

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

- Alimentation continue (+ 5 V)
- Plaquette de montage ou simulateur de fonctions logiques
- Interrupteurs logiques
- Pincettes 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, B et C.

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) Vérifier 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) :

Comparer un additionneur complet sous forme de C.I avec un additionneur construit avec des portes logiques.

2. Durée du TP:

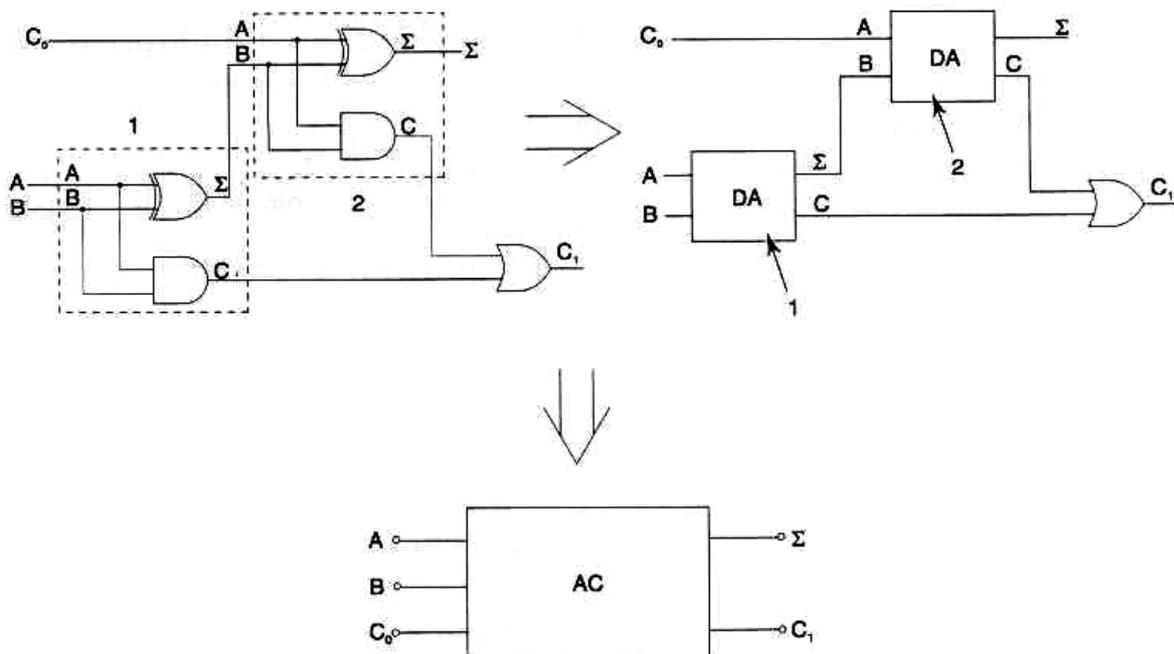
3 heures.

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

- Alimentation continue (+ 5 V)
- Plaque de montage ou simulateur de fonctions logiques
- Interrupteurs logiques
- Pinces et fils
- Résistance de limitation de 330 Ω
- Circuits intégrés TTL : 7432, 7486, 7408, 7483
- Notes de cours et fiches techniques

4. Déroulement du TP :

- a) Tracer le circuit d'un additionneur complet en utilisant uniquement des portes logiques de base. Ajouter le circuit d'une DEL pour afficher le résultat des sorties



- b) Monter votre circuit sur une plaque

- c) Simuler les diverses conditions d'entrées et inscrire les résultats dans le tableau suivant.

Entrées			Sorties			
			Portes Logiques		7483	
A	B	C ₀				
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

- d) Remplacer cet additionneur complet par un C.I. additionneur (7483A)
- e) Simuler les diverses possibilités d'addition.
- f) Ecrire le résultat des sorties dans les colonnes appropriées du tableau.
- g) Comparer les résultats des deux montages.

Tp 7: 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:

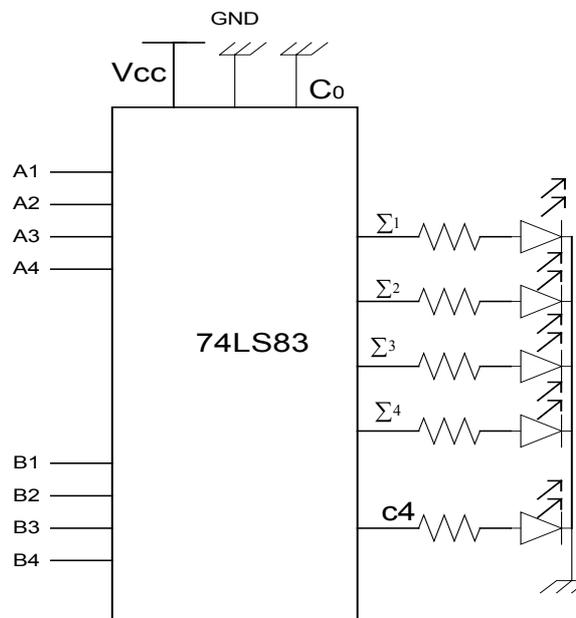
- Alimentation continue (+ 5 V)
- Plaquette de montage ou simulateur de fonctions logiques
- Interrupteurs logiques
- Pincettes et fils
- Résistances de limitation de 330 Ω
- Circuits intégrés 74LS83
- Notes de cours et fiches techniques

4. Description du TP :

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 côté de chaque broche le numéro correspondant ?
- b) Câbler le circuit.

c) Vérifier le fonctionnement du circuit en prenant quelques exemples :

$$\begin{array}{r} A1 A2 A3 A4 \\ + B1 B2 B3 B4 \\ \hline \end{array}$$

$$\begin{array}{r} 0100 \\ + 0111 \\ \hline \end{array}$$

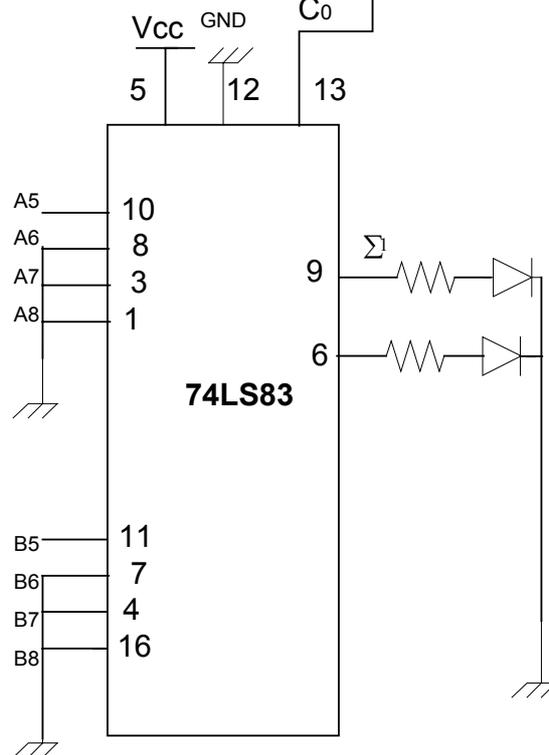
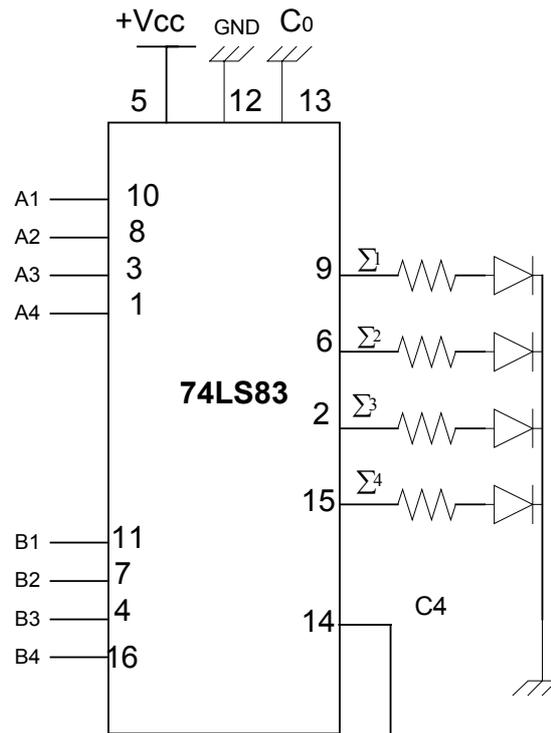
$$\begin{array}{r} 1110 \\ + 0111 \\ \hline \end{array}$$

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

a) Câbler le circuit.

b) Quelle est la retenue de cette addition ?

$$\begin{array}{r} 11110 \\ + 11011 \\ \hline \end{array}$$



Tp 8: monter des circuits de base

1. Objectif(s) visé(s) :

- Montage des circuits de base comme le comparateur
- 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
- Pincettes et fils
- Résistances de limitation de 330Ω
- Circuits intégrés 74LS85
- Notes de cours et fiches techniques

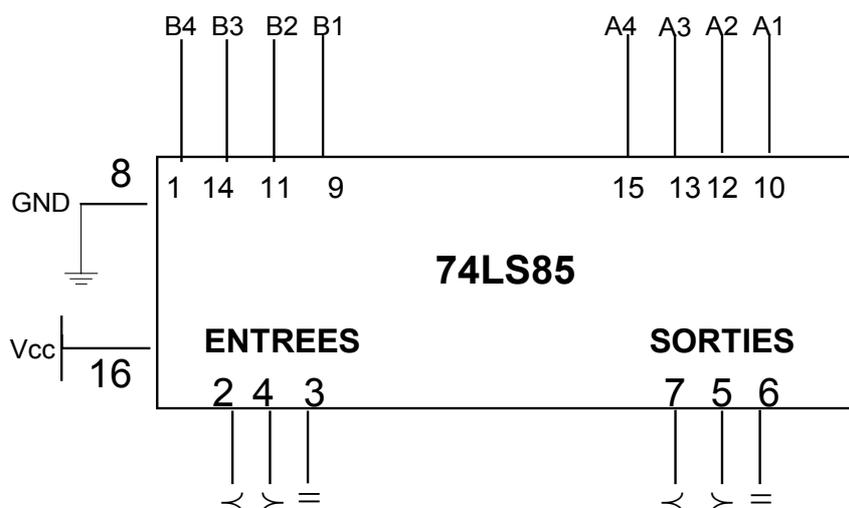
4. Description du TP :

On veut comparer deux nombres binaires en utilisant le circuit intégré 7485

5. Déroulement du TP :

I- Comparaison de nombres de 4 bits chacun :

- 1) Câbler le circuit suivant
- 2) Comparer deux nombres que vous choisissez (17 et 20; 10 et 10; 6 et 4)

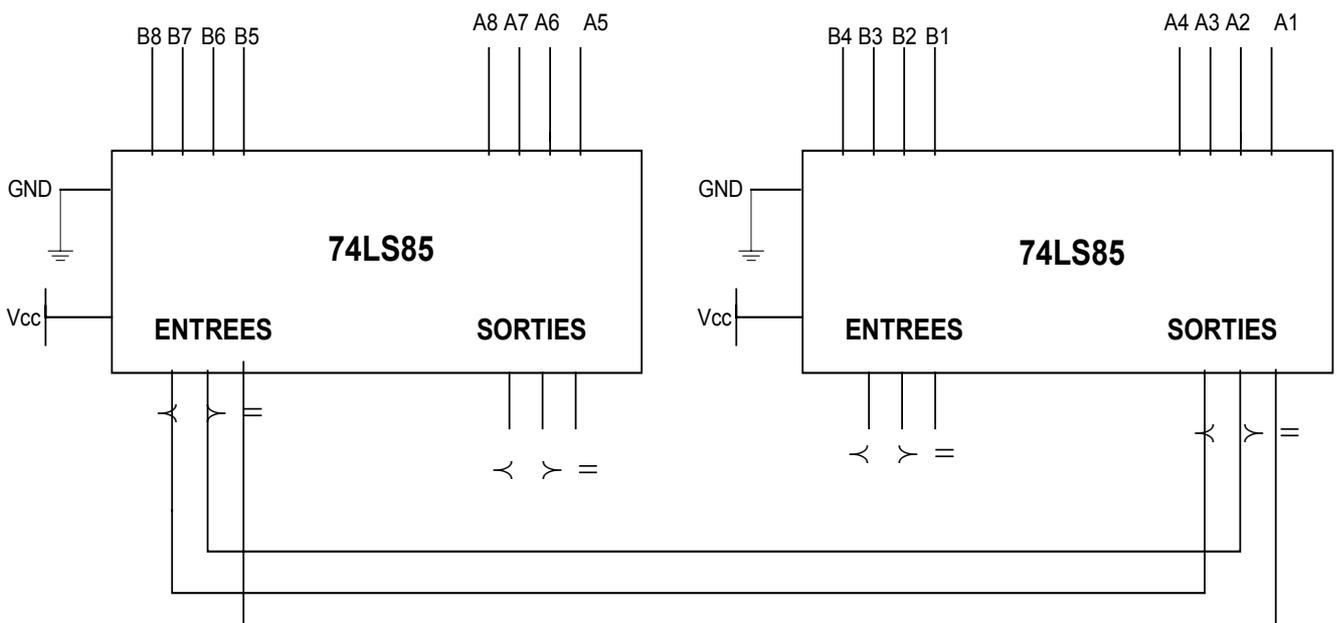


Remarque :

L'entrée de mise en cascade (A=B) doit être reliée à une tension de niveau haut

II- Comparaison de deux nombres de 8 bits chacun en utilisant le circuit intégré 7485

- 1) En se basant sur le brochage du circuit 7485, compléter les numéros des broches ajoutés
- 2) Câbler le circuit
- 3) Comparer deux nombres que vous choisissez



Remarque :

- Les sorties (A<B), (A>B), et A=B de 1^{er} C.I sont reliées aux entrées de mise en cascade de 2^{ème} C.I
- L'entrée de mise en cascade (A=B) de premier C.I doit être reliée à une tension de niveau haut

Prendre par exemple pour les entées de mise en cascade du 1^{er} CI:

- (A=B) = 1
- (A>B) = 0
- (A<B) = 0

Tp 9: monter des circuits de base**1. Objectif(s) visé(s) :**

- Montage des circuits de base comme le démultiplexeur
- Positionnement des composants ;
- Conformité du montage avec le schéma ;

2. Durée du TP:

3 heures.

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

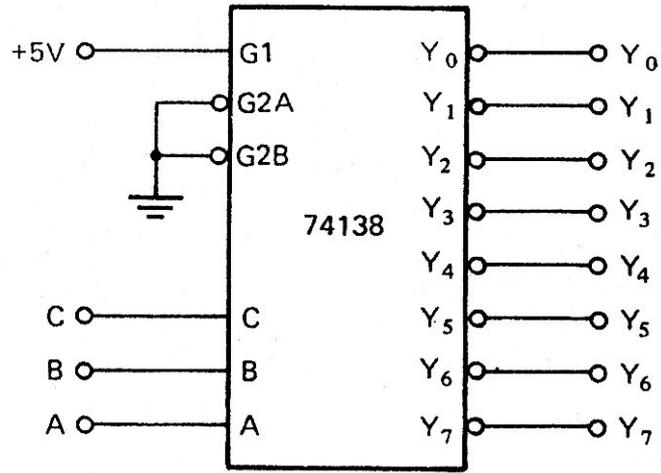
- Alimentation continue (+ 5 V)
- Plaquette de montage ou simulateur de fonctions logiques
- Interrupteurs logiques
- Pincettes et fils
- Résistances de limitation de 330 Ω
- Circuits intégrés 74138
- Notes de cours et fiches techniques

4. Déroulement du TP :

1. Réaliser le circuit de la figure suivante (A, B, et C représentent les entrées de sélection, G1, G2A, G2B représentent les entrées de validation).
2. Relier les points A, B et C respectivement aux interrupteurs de données « 2 », « 1 » et « 0 ». Relier ensuite les points Y0 à Y7 aux afficheurs à LEDS
3. Changer continuellement l'état des trois interrupteurs de données comme indiqué dans le tableau suivant et écrire le résultat dans la colonne des sorties.

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

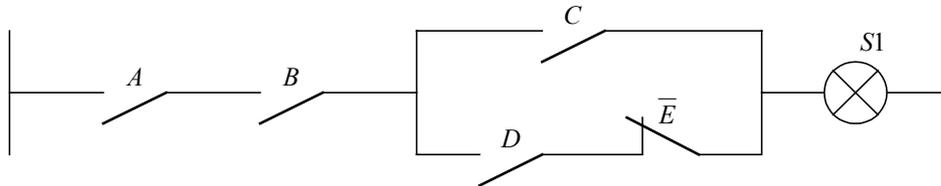
4. Si la broche G1 est reliée à la masse ou si G2A, ou G2B sont reliés à « +5V » que se passe-t-il ?



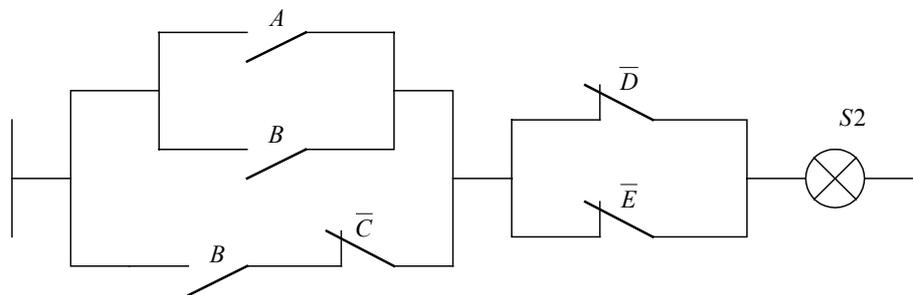
Evaluation de fin de module

1. Transformer les schémas électriques suivants en équation :

a)



b)



2. Traduire les équations logiques suivantes en logigrammes :

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

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

$$S_3 = \overline{AB \oplus \overline{A}(\overline{BC} + D)}$$

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.

5. On se propose d'étudier un distributeur de boisson. Celui-ci offre le choix entre de la menthe (m) et de l'orange (o) à condition d'avoir inséré une pièce (p). L'eau (e) est offerte gracieusement. L'utilisateur dispose de 3 boutons poussoirs ainsi qu'un monnayeur pour commander ce qu'il désire.

D'un point de vue système, l'appareil dispose de trois électrovannes permettant de délivrer l'eau, le sirop de menthe ou d'orange. Les 3 fonctions étudiées sont E, M et O qui représentent l'état des trois électrovannes.

La table de vérité fait apparaître un tableau constitué par 7 colonnes et 16 lignes comme le montre le tableau ci-dessous:

- Compléter la table de vérité ci-dessous.
- Donner le logigramme complet validant le fonctionnement du distributeur de boisson.
- Réaliser le montage du circuit correspondant au fonctionnement de ce distributeur

p	e	m	o	E	M	O
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Liste des références bibliographiques

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