Table des matières

Liste des abréviations

Bit	*Binary Digit*
LED	*Light Emitting Diode*
LSB	*Least Significant Bit*
MSB	*Most Significant Bit*
SPICE	*Simulation Program with Integrated Circuits Emphasis*
TTL	*Transistor Transistor Logic*

Liste des tableaux

Liste des figures

INTRODUCTION GENERALE

En électronique, que ce soit dans le domaine analogique ou bien dans le domaine du numérique, on parle toujours de signal. Le signal est le support de l'information qui elle –même porte un message. Ce message contenu dans l'information est la composante active du signal qui peut se distinguer en deux grandes catégories : signaux analogiques et signaux numériques.

La discipline s'intéressant aux traitements des signaux analogiques est bien sûr comme son nom l'indique l'électronique analogique qui est caractérisée par le fait que le signal électrique qu'il utilise à une amplitude ou une fréquence étant à tout moment proportionnelle à la grandeur physique qu'il représente. Et il y a toujours analogie dans l'évolution des deux grandeurs dans le temps.

Par opposition, l'électronique numérique s'intéresse au traitement des signaux dans l'espace du temps et discrets. Ainsi le nombre de valeurs que peuvent prendre ces signaux est limité. Et celle-ci utilise comme système de numération le système binaire qui est utilisé en particulier dans les systèmes contenant un microprocesseur ou un microcontrôleur comme un ordinateur.

L'électronique numérique est subdivisée en deux grandes parties : la logique séquentielle et la logique combinatoire qui est le sujet principal de notre travail.

Notre travail consiste en réalité à étudier et puis à réaliser une maquette pour les travaux pratiques des circuits combinatoires.

Nous allons le diviser en deux parties : la première va expliquer d'une manière générale ce qu'est la logique combinatoire et puis dans la seconde et dernière partie nous allons entrer dans les détails de notre travail avec l'explication d'un exemple de simulation.

CHAPITRE I : LA LOGIQUE COMBINATOIRE

I – GENERALITES

La logique combinatoire est une partie de l'électronique numérique et puisqu'elle est numérique donc elle utilise comme technique les nombres plus précisément le système numérique.

Dans un système numérique, il y n'y a plus de rapport direct et permanent entre la grandeur physique de départ et la variation électrique représentative quant à sa forme.

Ce signal est matérialisé par des niveaux de tensions successifs qui ne peuvent avoir que deux valeurs différentes et qui représentent les chiffres « 1 » et « 0 ».

Ce signal électrique constitué de 0 et 1 représente la suite des valeurs que la grandeur physique peut prendre successivement. Ces valeurs sont exprimées par des nombres binaires successifs. Dans les circuits électroniques, ce signal est caractérisé par la présence ou l'absence de tension.

II - SYSTEME DE NUMERATION

En électronique numérique, comme on l'a précisé plus haut le système de numération le plus utilisé est le système binaire mais on peut aussi trouver d'autres à citer le système octal ou encore le système décimal mais aussi le système hexadécimal. La base d'un système de numération est le nombre de chiffres différents qu'utilise ce système de numération. [3]

Dans un système de numération à base « *b* », un nombre « N » peut être exprimé sous la forme suivante :

$$N_{10} = \sum_{i=0}^{n-1} (a_i \cdot b^i)$$

b = base

i = rang

a_i = chiffre de rang i

b^i = poids du chiffre a_i

1 - Système décimal

C'est le système à base 10 que nous utilisons tous les jours. Il comprend dix (10) chiffres différents comme son nom l'indique : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9 qui sont les a_i de ce système. Prenons l'exemple du nombre 2356 ; nous l'écrirons $(2356)_{10}$ et ce nombre N a la forme polynomiale suivante :

$$N = 2x10^3 + 3x10^2 + 5x10^1 + 6x10^0$$

2 - Système binaire

Le système dit à base 2 comprend deux symboles qui sont des chiffres bien sûr. Ce sont le « 0 » et le « 1 ».

Chacun d'eux est aussi appelé « BIT » ou binary digit ou tout simplement élément binaire en français. [2]

Prenons comme exemple N= $(10110)_2$. Ce nombres écrit sous la forme d'un polynôme a pour expression :

$$N = 1x2^4 + 0x2^3 + 1x2^2 + 1x2^1 + 0x2^0 = (22)_{10}$$

Le dernier chiffre de droite est le chiffre de poids le plus faible qui est le « LSB » ou Least Significant Bit, son rang est i=0 ; le 1 e plus à gauche est le chiffre de poids le plus fort c'est-à-dire le « MSB » ou Most Significant Bit, avec i=4. [1]

3 - Système octal

Ce système, dit à base huit comprend donc huit symboles qui sont 0, 1, 2, 3, 4, 5, 6 et 7

Exemple : N = $(6543)_8$

Ici, 3 est le LSB et 6 le MSB. Ce nombre a la forme polynomiale comme suit :

$$N = 6x8^3 + 5x8^2 + 4x8^1 + 3x8^0 = (3427)_{10}$$

4 - Système hexadécimal

Ce système par contre à part d'utiliser les 10 chiffres usuels de la base 10 il utilise aussi des lettres minuscules ou majuscules qui sont a, b, c d, e et f. [1] [2]

Exemple : N = $(AC53)_{16}$

3 est le LSB et le MSB est le A qui est égale à 10 en base 10.

Ce nombre donc peut être représenté sous la forme d'un polynôme :

$$N = Ax16^2 + Cx16^2 + 5x16^1 + 3x16^0$$
$$= 10x16^3 + 12x16^2 + 5x16^1 + 3x16^0 \quad = (44115)_{10}$$

III - OPERATIONS LOGIQUES

1 - Variables booléennes

Une variable logique (dite booléenne ou encore binaire) couramment nommée « bit » de l'algèbre de Boole est une grandeur binaire qui ne peut prendre que deux valeurs : 0 ou 1. Elle peut être utilisée pour représenter une proposition ou l'état d'un objet. [1]

Exemple : elle sera égale à 1 pour un moteur (ou tout autre dispositif comme une lampe) à l'état de marche, et 0 s'il est à l'état arrêt.

2 - Opérations logiques élémentaires

On peut définir des opérations mathématiques portant sur une variable logique.

On définit trois opérations logiques élémentaires :

-l'addition logique généralement symbolisée par le signe « + » ;

-la multiplication logique généralement symbolisée par le signe « **.** » ;

-l'inversion ou complémentation logique généralement symbolisée par le surlignement de la variable.

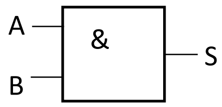
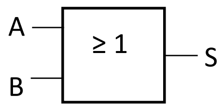
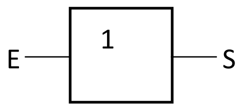
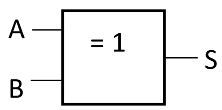
En électronique, des opérations dont les grandeurs d'entrée et de sortie sont des tensions représentant des variables logiques, permettent d'obtenir le résultat de ces opérations.

Rappelons qu'à l'état d'une entrée (ou d'une sortie) sont associées la valeur d'une tension : l'état 1 correspond à une tension voisine de x volts (5 V par exemple), à l'état 0 correspond une tension voisine de y volts (0 V par exemple). [1] [2]

3 - Fonction logique de base

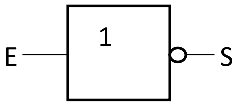
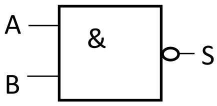
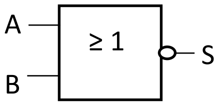
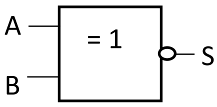
Elles suffisent à l'écriture et la réalisation de toute fonction logique d'un nombre quelconque de variables. On peut les classer en deux grandes catégories : la première est l'ensemble de tous les fonctions logiques de bases sans inverseurs que l'on peut voir sur le Tableau 1 ci-après :

<u>Tableau 1</u> : Liste des fonctions logiques de base [3]

<table>
<tr><th>Fonction</th><th>Symbole</th><th>Equation logique</th><th>Table de vérité</th></tr>
<tr><td>ET
(AND)
Produit logique</td><td></td><td>S = A . B
(S égale A et B)
S = A ∩ B</td><td><table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>OU inclusif
(OR)
Somme logique</td><td></td><td>S = A + B
(S égale A ou B)
S = A U B</td><td><table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>OUI
(buffer)</td><td></td><td>S = E
S égale E</td><td><table><tr><th>E</th><th>S</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>OU exclusif
(XOR)</td><td></td><td>S = A⊕ B
(S égale A ou exclusif B)</td><td><table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
</table>

La deuxième grande catégorie des fonctions logiques de bases est celle avec des inverseurs que l'on peut voir sur le Tableau 2 suivant :

<u>Tableau 2</u> : Liste des fonctions logiques de base avec inverseur [3]

Fonction	Symbole	Equation logique	Table de vérité
NON (Inverter) Complémentation		$S=\overline{E}$ (S égale E barre, E complémenté)	E S 0 1 1 0
NON-ET (NAND)		$S = \overline{A \,.\, B}$ (S = A et B barre, A et B complémenté)	A B S 0 0 1 0 1 1 1 0 1 1 1 0
NON-OU (NOR)		$S = \overline{A + B}$ (S égale A ou B barre, A ou B complémenté)	A B S 0 0 1 0 1 0 1 0 0 1 1 0
IDENTITE (OU exclusif complémenté) (XNOR)		$S = \overline{A \oplus B}$ (S égale A ou exclusif complémenté B)	A B S 0 0 1 0 1 0 1 0 0 1 1 1

<u>4 - Propriétés des fonctions logiques de base</u>

<u>a - Règles de priorité</u>

Dans une équation logique la complémentation a la plus forte priorité, puis vient la fonction ET, et la fonction OU. L'utilisation des parenthèses dans les équations logiques permet de s'affranchir de ces priorités. Autrement dit, ce sont les opérations à l'intérieur de ces derniers qui deviennent prioritaires comme le montre le Tableau 3 ci-après :

<u>Tableau 3</u> : **Les propriétés des fonctions logiques de base** [3]

Propriétés	Fonction OU	Fonction ET
Commutativité	$A + B = B + A$	$A . B = B . A$
Associativité	$A + (B + C) = (A + B) + C$	$A . (B . C) = (A . B) . C$
Distributivité	$A . (B + C) = A . B + A . C$	$A + (B . C) = (A + B) . (A + C)$
Idempotence	$A + A = A$	$A . A = A$
Elément neutre	$A + 0 = A$	$A . 1 = A$
Elément absorbant	$A + 1 = 1$	$A . 0 = 0$
Complémentation	$\overline{\mathbf{A}} + A = 1$	$\bar{A} . A = 0$
Involution	$\overline{\overline{\mathbf{A}}} = A$	

<u>b - Règles de De Morgan</u>

<u>Première règle</u>

Le complément d’un produit de variables est égal à la somme de leurs compléments [1] :

$$\overline{\mathbf{A . B}} = \overline{\mathbf{A}} + \overline{\mathbf{B}}$$

<u>Seconde règle</u>

Le complément d’une somme de variables est égal au produit de leurs compléments [1] :

$$\overline{\mathbf{A + B}} = \overline{\mathbf{A}} . \overline{\mathbf{B}}$$

<u>c - Propriétés induites</u>

- **$A + \overline{A} . B = A + B$ car $\overline{\overline{A + \overline{A} . B}} = \overline{\overline{A} . (A + \overline{B})} = \overline{A . \overline{A} + \overline{A} . \overline{B}} = \overline{\overline{A} . \overline{B}} = A + B$**
- **$A + A . B = A$**
- **$(A + B) . (A + C) = A + B . C$**
- **$A . B + \overline{A} . C = A . B + \overline{A} . C + B . C$**

IV - FONCTION LOGIQUE QUELCONQUE

1 - Fonction logique quelconque complètement définie

Une fonction de *n* variables binaires est complètement définie si sa valeur est connue pour chacune des 2^n combinaisons possibles des variables. La table de vérité d'une telle fonction comportera donc 2^n lignes.[1] [2]

Un exemple de table de vérité d'une fonction *F* de trois variables A, B et C est donné sur la Figure 1.1 ci-dessous.

Ici, on a $n = 3$ variables donc on a $2^3 = 8$ combinaisons possibles mais aussi 8 lignes pour la table de vérité de la fonction *F*

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

Figure 1.1 : Fonction logique quelconque complètement définie

2 - Fonction logique quelconque partiellement définie

Dans ce cas la valeur de la fonction n'est pas déterminée pour toutes les combinaisons des variables. Une valeur indéterminée est notée X ou Ø dans la table de vérité.[1] Pour les cas indéterminés, on peut imposer une valeur 0 ou 1 à cette fonction, dans le but de faciliter sa synthèse.[2] Un exemple d'une telle fonction est donné sur la figure 1.2.

a	B	c	*G*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	X
1	0	0	X
1	0	1	0
1	1	0	X
1	1	1	X

<u>Figure 1.2</u> : Fonction logique quelconque partiellement définie

3 - Fonctions ou termes produits, mintermes

Pour chacune des combinaisons des variables on définit une fonction égale au ET entre les variables vraies ou complémentées. Pour une fonction de trois variables A, B et C, il y a 8 fonctions, ou termes produits, ce sont les mintermes : [3]

$$P_0 = A \cdot B \cdot C$$

$$P_1 = A \cdot B \cdot \overline{C}$$

$$P_2 = A \cdot \overline{B} \cdot C$$

$$P_3 = A \cdot \overline{B} \cdot \overline{C}$$

$$P_4 = \overline{A} \cdot B \cdot C$$

$$P_5 = \overline{A} \cdot B \cdot \overline{C}$$

$$P_6 = \overline{A} \cdot \overline{B} \cdot C$$

$$P_7 = \overline{A} \cdot \overline{B} \cdot \overline{C}$$

4 - Fonctions ou termes sommes, maxtermes

De même, pour chacune des combinaisons des variables on définit une fonction égale au OU entre les variables vraies ou complémentées. Pour une fonction de trois variables A, B et C, il y a huit fontions, ou termes sommes, ce sont les maxtermes : [3]

$$S_0 = A + B + C$$

$$S_1 = A + B + \overline{C}$$

$$S_2 = A + \overline{B} + C$$

$$S_3 = A + B + \overline{C}$$

$$S_4 = \overline{A} + B + C$$

$S_5 = \overline{A} + B + \overline{C}$

$S_6 = \overline{A} + \overline{B} + C$

$S_7 = \overline{A} + \overline{B} + \overline{C}$

5 - Formes canoniques des fonctions logiques

Une fonction logique peut s'écrire sous la forme de somme de variables vraies ou complémentées, ou sous forme de produits de sommes de variables vraies ou complémentées.

On obtient la première forme canonique en réunissant par des fonctions OU tous les termes produits, ou mintermes, pour lesquels la fonction vaut 1. Ainsi, de la table de vérité de la Fig. 1.1, on tire une équation logique donnant ***F*** :

$$\boldsymbol{F} = \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{C} + A \cdot \overline{B} \cdot \overline{C}$$

La deuxième forme canonique s'obtient en réunissant par des fonctions ET tous les termes sommes, ou maxtermes, pour lesquels la fonction vaut 0. Ainsi, de la table de vérité sur la Fig. 1.1, on tire une deuxième équation logique donnant ***F*** :

$$\boldsymbol{F} = (A + B + C) \cdot (A + \overline{B} + \overline{C}) \cdot (A + B + \overline{C}) \cdot (A + \overline{B} + C) \cdot (\overline{A} + \overline{B} + \overline{C})$$

V – PRINCIPE DE MINIMISATION

1 – Introduction

La minimisation consiste à simplifier l'expression d'une fonction logique dans le but d'optimiser le nombre de composants, ou portes nécessaires à sa réalisation. [3]

On distingue deux (2) façons de minimiser une fonction et l'expression obtenue est la forme minimale.

2 - Simplification analytique

Il convient tout d'abord de tirer de la table de vérité la forme canonique comportant le moins de termes. Pour réaliser une simplification analytique, il suffit d'appliquer les différentes propriétés de l'algèbre de Boole. [1]

Présentons cette méthode dans un exemple représenté sur la Figure 1.3 :

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

<u>Figure 1.3</u> : Exemple de table de vérité d'une fonction *F* quelconque

L'équation tirée de la table de vérité da la Fig. 1.1 s'écrit :

$$\boldsymbol{F = \overline{A} . B . C + A . \overline{B} . C + A . B . \overline{C} + A . B . C}$$

Comme $\mathbf{A . B . C = A . B . C + A . B . C + A . B . C}$, nous pouvons transformer l'expression de la fonction :

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

Après regroupement des termes et factorisation, nous obtenons :

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

Or, $\mathbf{\overline{A} + A = 1}$, $\mathbf{\overline{B} + B = 1}$ et $\mathbf{\overline{C} + C = 1}$;

donc nous déduisons une forme minimale :

$$F = B . C + A . C + A . B$$

<u>3 - Simplification au moyen des tableaux de Karnaugh</u>

<u>*a – Principe*</u>

Cette méthode de simplification consiste à mettre en évidence graphiquement les groupements de termes produits qui ne diffèrent que par l'état d'une seule variable d'entrée (termes adjacents). [2]

b - Description des tableaux de Karnaugh

Un tableau de Karnaugh est une autre forme de la table de vérité. Il est organisé en colonnes et lignes dont les intersections donnent une case qui représente un des termes produits de la fonction. Pour une fonction de *n* variables, le tableau comportera 2^n cases. On écrit dans chaque case la valeur correspondante de la fonction : 0 ou 1, et si cette valeur est indéterminée, Ø ou X. [3]

c - Construction des tableaux de Karnaugh

Lignes et colonnes sont repérées par une combinaison des variables, sous forme littérale et par valeurs. La construction du tableau de Karnaugh est telle que les lignes et les colonnes successives sont repérées par des combinaisons adjacentes. Les tableaux de Karnaugh couramment utilisés concernent des fonctions de deux variables (Fig. 1.4), trois variables (Fig. 1.5), ou quatre variables (Fig. 1.6) ;

Remarque : Le variable « a » est toujours le LSB dans les exemples suivants.

F	0	1
0		
1		

Figure 1.4 : Table de Karnaugh d'une fonction F à deux variables a et b

F	00	01	11	10
0				
1				

Figure 1.5 : Table de Karnaugh d'une fonction F à trois variables a, b et c

F	00	01	11	10
00				
01				
11				
10				

Figure 1.6 : Table de Karnaugh d'une fonction à 4 variables a, b, c et d

d - Passage de la table de vérité au tableau de Karnaugh

Il suffit de reporter dans chaque case du tableau de Karnaugh la valeur de la variable de sortie correspondant à chaque combinaison des variables d'entrée.

a	b	c	H
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Figure 1.7 : Table de vérité d'une fonction quelconque H

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

Figure 1.8 : Table de Karnaugh de la fonction quelconque H

e - Utilisation des tableaux de Karnaugh

L'équation est obtenue en procédant à des groupements de cases adjacentes. Les règles à respecter sont :

– le nombre de cases groupées doit être égal à une puissance de 2 (2, 4, 8, voire même 16);

– le nombre de cases entrant dans un groupement doit être le plus grand possible ;

– une case peut être incluse dans plusieurs groupements ;

– une case contenant un état indéterminé peut être incluse dans les groupements.

f - Exemple d'utilisation d'un tableau de Karnaugh

Les groupements mis en évidence dans les Fig. 1.9 et Fig. 1.10 donnent respectivement les deux termes produits $a \cdot \bar{c}$ et $\bar{a} \cdot b$

Une équation de H que l'on peut voir sur les figures : Figure 1.9 et Figure 1.10 est :

$H = a \cdot \bar{c} + \bar{a} \cdot b$

F	00	01	11	10
0		1	1	
1	1	1		

<u>Figure 1.9</u> : Définition du premier groupement des valeurs 1 de la fonction H ci-dessus

F	00	01	11	10
0		1	1	
1				

<u>Figure 1.10</u> : Définition du second groupement des valeurs 1 de la fonction H ci-dessus

4- Conclusion

La première méthode : la simplification analytique repose beaucoup sur l'intuition. Sa mise en œuvre devient difficile et fastidieuse lorsque le nombre de variables est grand. On préfère alors la méthode graphique qui utilise les tableaux de Karnaugh.

Mais les tableaux de Karnaugh sont bien adaptés pour la représentation de fonctions de cinq (5) variables au plus car dès que le nombre de variables est supérieur à six (6), les problèmes posés doivent être traités en informatique par des logiciels comme « l'algorithme de simplification de QUINE et MAC CHESKEY » à cause de leurs complicités. [1]

CHAPITRE II : <u>ETUDE ET REALISATION DE LA MAQUETTE</u>

I – PRESENTATION GENERALE

On peut dire que la maquette en général est assez large. Elle sert à simuler les opérations en électronique numérique : logique combinatoire et logique séquentielle.

Elle est divisée en quatre (4) parties bien distinctes (Fig. 2.1):

-l'alimentation

-les entrées des variables

-la sortie

-le support des circuits intégrés

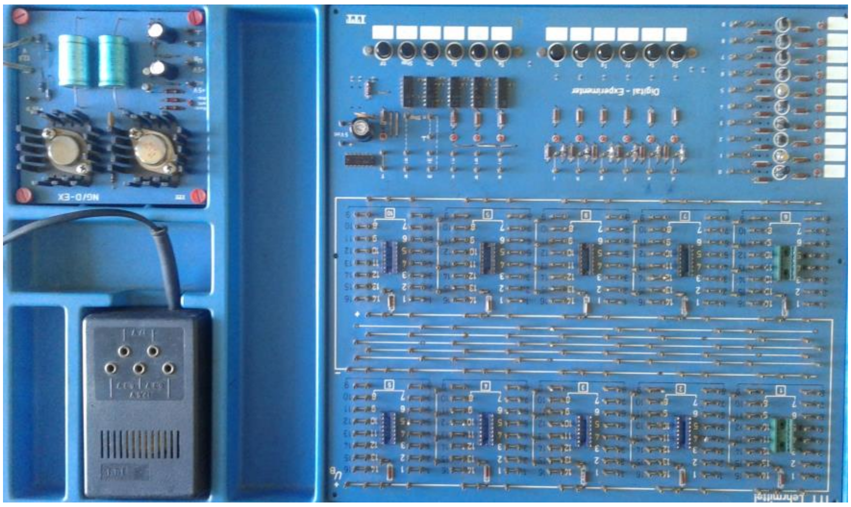


Figure 2.1: Schéma général de la maquette

Voyons une après une ces 4 grandes parties :

La plus grande partie se trouvant au-dessous des trois plus petite est la partie de la maquette où il faut placer les circuits intégrés qu'on va utiliser pour voir son fonctionnement

La petite partie se trouvant dans la partie à droite sur la maquette est la sortie binaire où va s'afficher les valeurs binaires des opérations vues dans le ou les circuit(s) intégré(s) utilisé(s).

Ensuite, à gauche se trouve les entrées des variables en binaires pour la qu'on verra de façon plus claire dans les pages ci-dessous pour celle de la logique combinatoire.

Enfin, se trouvant à gauche sur la Fig. 2.1, on a l'alimentation qui va alimenter la maquette toute entière.

II - ETUDE DE LA MAQUETTE

1-Alimentation

A cause des circuits intégrés utilisés pour les simulations, la maquette ne peut supporter qu'une tension stable ayant une valeur de +5V.

L'alimentation de la maquette est affichée sur la Fig. 2.2 :

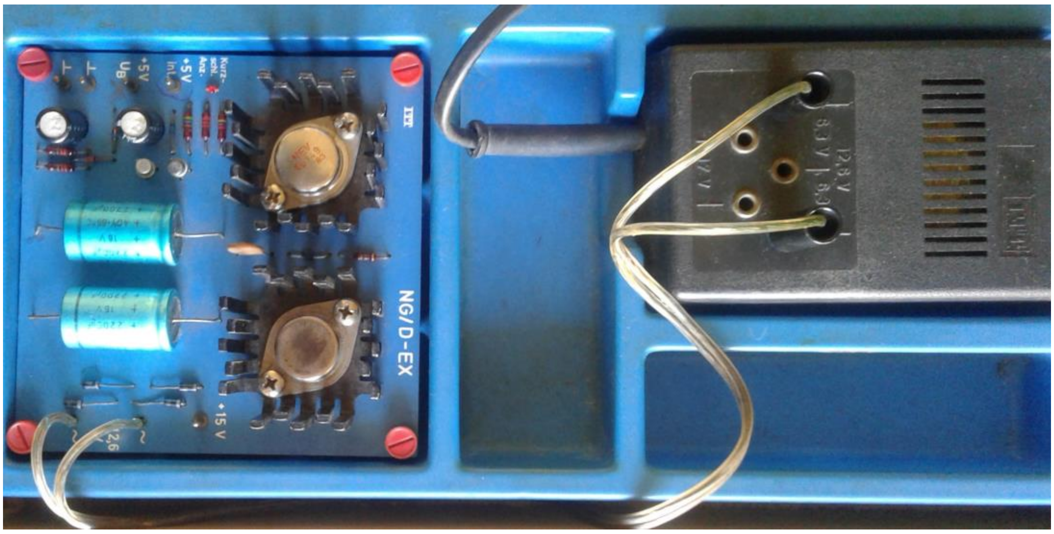


Figure 2.2: Alimentation de la maquette

On a donc besoin d'une alimentation stabilisée ayant une tension de sortie de +5V

La source de tension qu'on utilise à Madagascar a environ une valeur de 220V donc le transformateur utilisé a une tension au primaire égale à 220V et au secondaire 12V.

2-Entrée des variables

a - Généralités

L'entrée est divisée en deux (2) grandes parties :

-l'entrée des variables combinatoires (Fig. 2.3a)

-l'entrée des variables séquentielles avec l'horloge (Fig. 2.3.b)

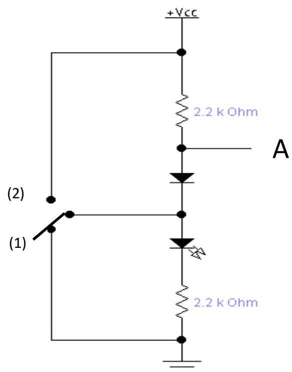
a) b)

<u>Figure 2.3</u> : Entrée des variables : a) variables combinatoires, b) variables séquentielles

<u>b - Fonctionnement</u>

Les variables utilisées sont des variables logiques et comme on l'a dit plus haut une variable logique ne peut prendre que deux valeurs : soit 0 soit 1. Elle sera égale à 1 lorsque la LED est allumée et 0 lorsqu' elle est éteinte.

Le schéma électrique ci-dessous (Fig. 2.4) qui montre le fonctionnement de l'entrée des variables combinatoires a été simulé par le logiciel ELECTRONICS WORKBENCH. **[4]**



<u>Figure 2.4</u> : schéma électrique de l'entrée d'une variable combinatoire

D'après la Fig. 2.4, initialement l'interrupteur se trouve à la position (1) étant reliée à la masse donc la LED est éteinte à cause de la présence du court-circuit.

On aura un niveau de tension bas au point considéré (ici c'est le point A).

Après mesure avec un multimètre on a constaté que ce niveau bas de tension est égal à environ 0,42 Volt et correspond au 0 logique pour la variable considérée soit A = 0.

Si on ferme l'interrupteur (c'est-à-dire en appuyant sur le bouton poussoir de la maquette), on aura la valeur 1 logique pour la variable considérée soit A = 1.

En effet, lorsque l'interrupteur se trouve à la position (2), la LED s'allume, et la diode est en circuit ouvert car on a $U_{AK} < 0$ avec $U_k = V_{cc}$ et $U_A = V_{cc}$ – I. (2,2 k) < V_{cc}

Si on mesure le niveau de tension au point considéré, on aura le niveau haut ayant une valeur de 4,25 Volt.

3-La sortie

Généralités

On a la même sortie pour afficher les résultats des opérations que ce soit en logique combinatoire ou séquentielle (Fig. 2.5).

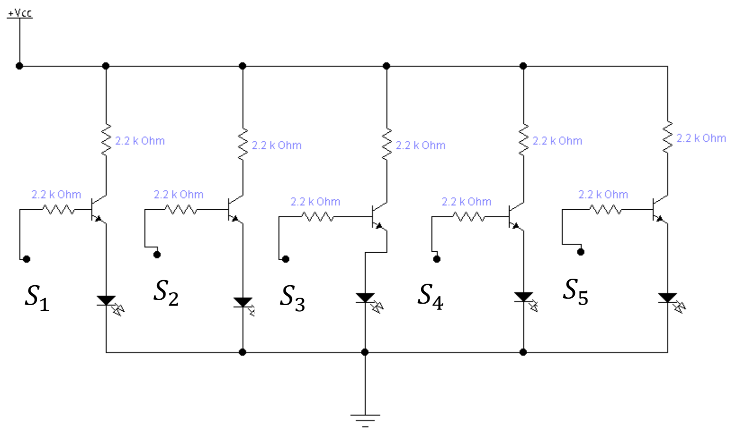
Figure 2.5 : sortie à l'aide de LED

Fonctionnement

Cette sortie affiche des valeurs binaires : soit 0 soit 1.

Elle a le même principe que celui de l'entrée c'est-à-dire on a la valeur 1 logique si la LED est allumée et la valeur vaut 0 logique si elle est éteinte.

Voici, sur la figure ci-après (Fig. 2.6), son schéma électrique d'après le logiciel ELECTRONICS WORKBENCH : **[4]**



<u>Figure 2.6</u> : schéma électrique de la sortie par des diodes LED

Le signal entrant au point S_i est toujours positif mais soit du niveau bas soit du niveau haut. S'il est du niveau haut alors le transistor ne sera pas bloqué. Tandis que s'il est du niveau bas, le courant sera absorbé par la résistance d'où la saturation du transistor.

Le transistor est du type NPN et fonctionne en court-circuit si on a un courant de base $I_B > 0$

Et la LED s'allumera automatiquement car le courant peut circuler, ce qui montre que la sortie vaut 1.

Si le signal reçu par la base du NPN c'est-à-dire $V_{BE} \leq 0.7$ alors celui-ci va le bloquer : le transistor va se comporter comme un circuit ouvert, le courant ne pourra pas donc circuler d'où la LED sera éteinte : ce qui nous donne une sortie valant 0.

<u>Affichage par un afficheur 7 segments :</u>

Au lieu de se contenter d'afficher le résultat des opérations à l'aide des LED, on va utiliser un afficheur 7 segments afin de le visualiser sous forme décimale.

On va utiliser un afficheur constitué de 7 diodes dont l'anode est commune. Il convient donc de les alimenter correctement de manière analogue au LED simples : l'anode commune sur V_{cc} et chaque cathode reliée à une commande.

Afin de faciliter la conversion, nous allons utiliser un circuit de type LS47, décodeur BCD vers 7 segments. Ayant un collecteur ouvert, il est convenable d'ajouter des résistances à ce circuit afin de contrôler les courants qui vont circuler dans les LED.

De la même manière que pour les diodes simples en sortie, on va utiliser 7 résistances ayant toutes la même valeur de 470 Ω.

La connexion se fait d'après la datasheet comme indiquée sur la figure 2.7

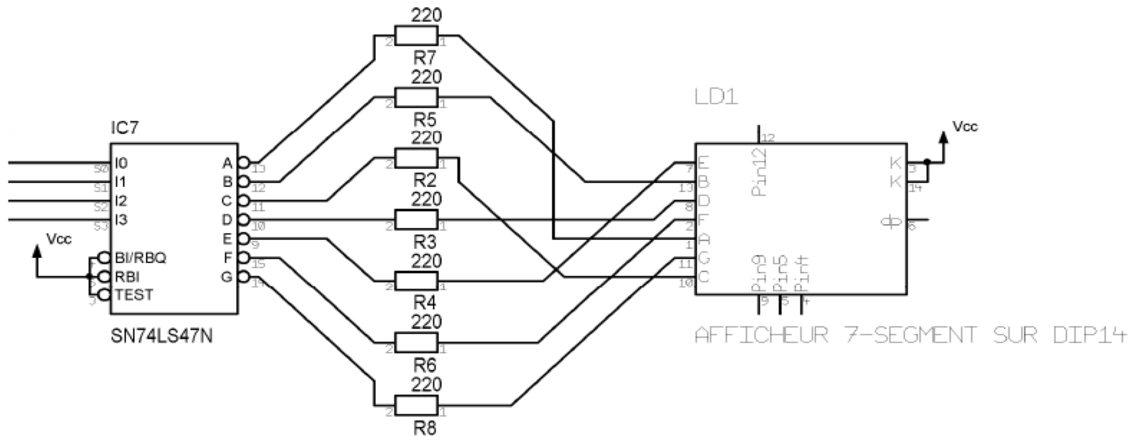


Figure 2.7 : connexıon entre un circuit LS47 et un afficheur 7 segments

4-Les supports des circuits intégrés :

Au total, notre maquette comprend dix (10) supports de circuits intégrés alimentée d'une seule source de tension $+U_B$ (Fig. 2.8) . Ils sont tous des supports pour un circuit intégré à 16 broches et si l'on possède un circuit intégré à 14 broches, deux entrées ne sont pas utilisées.

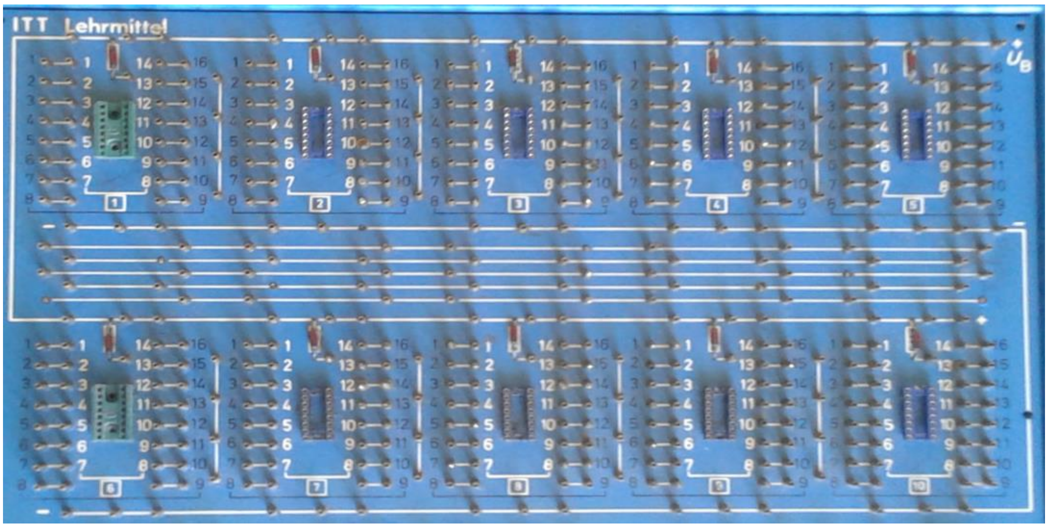


Figure 2.8 : schéma général des 10 supports des circuits intégrés de la maquette

III - EXEMPLE DE SIMULATION SUR LA MAQUETTE

1- Introduction

Dans cet exemple de simulation, nous allons réaliser un circuit permettant d'additionner deux nombres écrits sous forme binaire naturelle. Nous commencerons par concevoir un circuit additionneur 3 bits puis nous pourrons généraliser pour *n* bits. Nous réaliserons un circuit prenant en charge 4 bits.

Pour cela, nous utiliserons uniquement des portes logiques simples ET, OU, NON,

2- Additionneur 4 bits

Nous nous intéresserons uniquement à la logique nécessaire à l'addition.

Voici les deux nombres que nous allons utiliser :

A = $a_n \dots a_2 a_1 a_0$

B = $b_n \dots b_2 \, b_1 b_0$

Nous prendrons donc en premier temps n=2 et pour chaque a_i et b_i nous pourrons calculer

s_i = la somme en binaire (modulo 2) de a_i et b_i

r_i = la retenue associée

a- Le premier bit

Voici, sur la Fig. 2.9 , la table de vérité pour le LSB :

a_0	b_0	s_0	r_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure 2.9 : bit du poids plus faible

On remarque alors les deux lois caractéristiques :

$s_0 = a_0 \oplus b_0$

$r_0 = \ a_0 \times \ b_0$

On peut voir le logigramme correspondant sur la Fig. 2.10 :

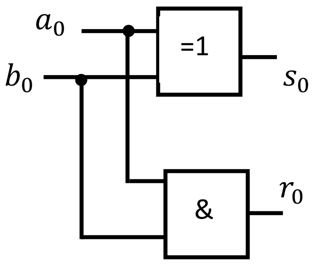


Figure 2.10 : logigramme de l'addition des LSB

b - Deuxième bit

Le deuxième est un peu plus complexe. En effet, pour le calcul de s_1 et r_1 la logique doit non seulement prendre en compte a_1 et b_1 mais aussi la retenue du calcul précédent. Alors la table de vérité correspondante se trouve sur la Fig. 2.11 :

a_1	b_1	r_0	s_1	r_1
0	**0**	**0**	**0**	**0**
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Figure 2.11 : table de vérité de l'addition des seconds bits

Somme s_1

Il est préférable de présenter cette table sous la forme d'un tableau de Karnaugh :

r_0 \ a_1b_1	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Figure 2.12 : table de Karnaugh pour s_1

On reconnaît ici la forme caractéristique d'un OU EXCLUSIF ou XOR : $s_1 = a_1 \oplus b_1 \oplus r_0$

Retenue r_1

On procède de la même manière que pour trouver la valeur de la Somme s_1 (Fig. 2.13)

r_0	a_1b_1	00	01	11	10
0		0	0	1	0
1		1	1	1	1

Figure 2.13 : table de Karnaugh pour r_1

On a : $r_1 = a_1 \times b_1 + r_0 \times (a_1 \oplus b_1)$

Voici sur la Figure 2.14, le logigramme correspondant :

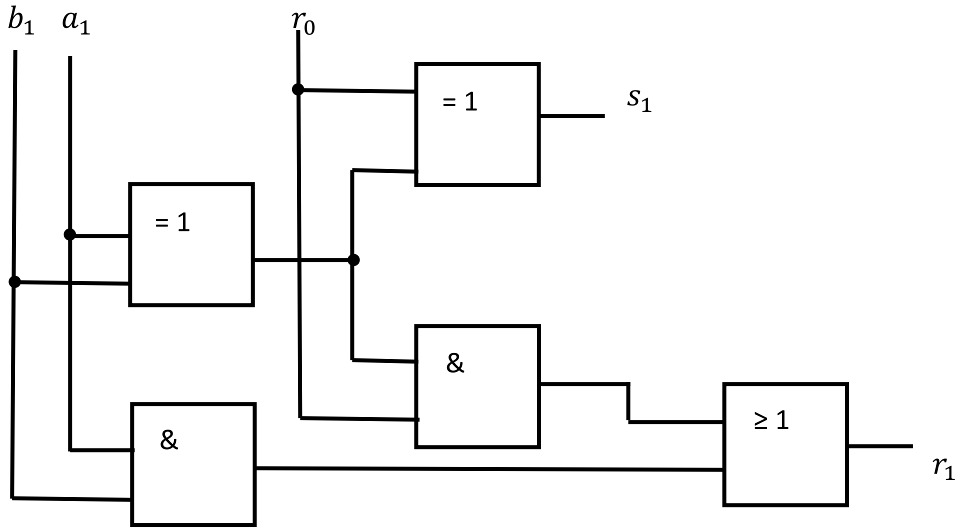


Figure 2.14 : logigramme de l'addition des seconds bits

c - bits suivants

On se rend finalement compte que l'on peut généraliser les équations logiques au **2-)** pour n'importe quel bit de rang i :

$$s_i = a_i \oplus b_i \oplus r_{i-1}$$

$$r_i = a_i \times b_i + r_{i-1} \times (a_i \;\; b_i)$$

On peut alors facilement réaliser un additionneur n-bits en juxtaposant plusieurs fois la même série de portes logiques.

CONCLUSION

Dans la logique combinatoire, toutes les fonctions peuvent toujours s'écrire à partir des trois opérations suivantes : **NON** , **ET** , **OU** avec des variables écrites sous formes vraies ou/et complémentées.

Grâce aux avancés des techniques informatiques, des logiciels de simulation comme **SPICE** ou ***S**imulation **P**rogram with **I**ntegrated **C**ircuits **E**mphasis* et bien d'autres encore permettent aujourd'hui de traiter ces différentes opérations sur la logique combinatoire.

Les résultats donnés par ces logiciels peuvent nous être utiles en travaux pratiques. La raison est de ne pas abimer les circuits intégrés lors de leurs installations : au moins on est sûr du fonctionnement du circuit avant leurs placements sur la maquette.

Dans notre étude, les entrées ont été affichées par des LED mais on peut encore aussi les afficher de la même manière que pour la sortie : avec un afficheur 7 segments. Mais c'est l'objet d'une autre étude à approfondir.

ANNEXE

Brochage du circuit intégré 7400

Le circuit intégré 7400 est un circuit comportant quatre (4) portes NON ET à 2 entrées comme on peut le percevoir sur la Figure A.1 ci-dessous :

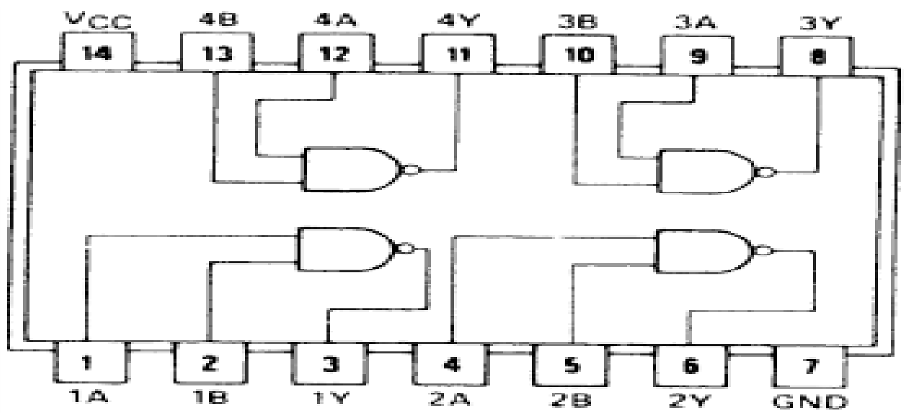


Figure A.1 : Circuit 7400

Brochage du circuit intégré 7404

Quant au circuit 7404, il comprend six (6) inverseurs comme le montre la Figure A.2 :

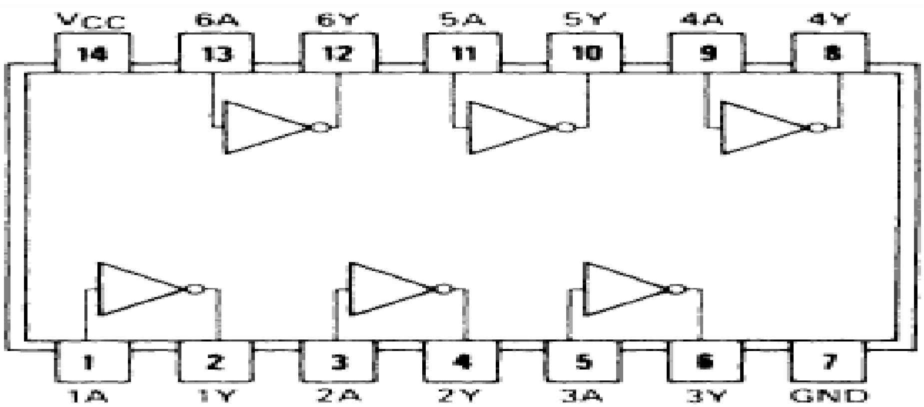


Figure A.2 : Circuit 7404

Brochage du circuit intégré 7408

Le circuit sur la Figure A.3 suivant, montre quatre (4) portes ET à 2 entrées.

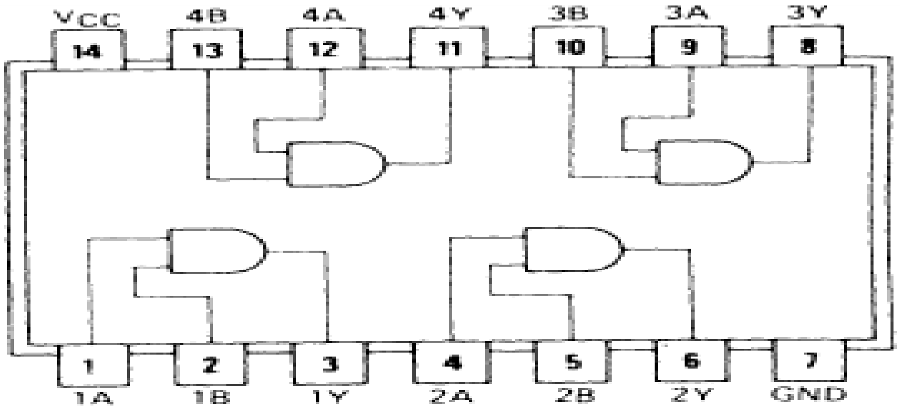


Figure A.3 : Circuit 7408

Brochage du circuit intégré 7410

Sur la Figure A.4 ci-dessous, on a encore un circuit ET mais il comporte seulement trois (3) variables à l’entrée :

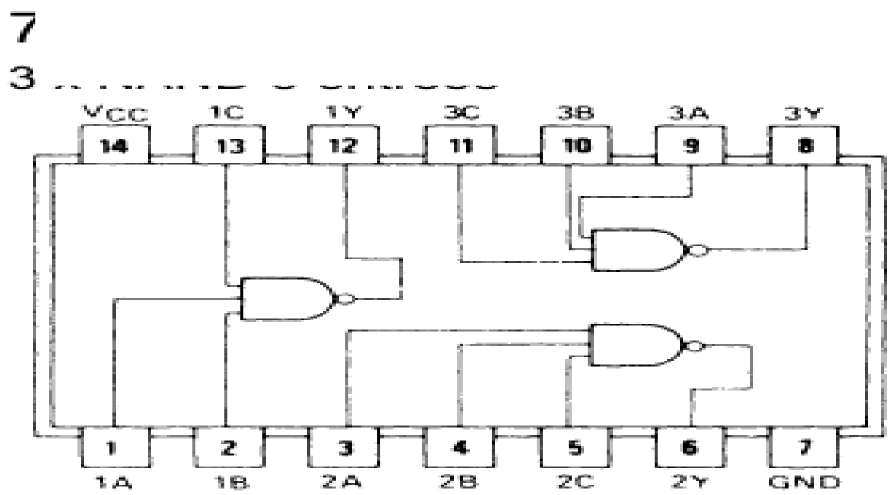
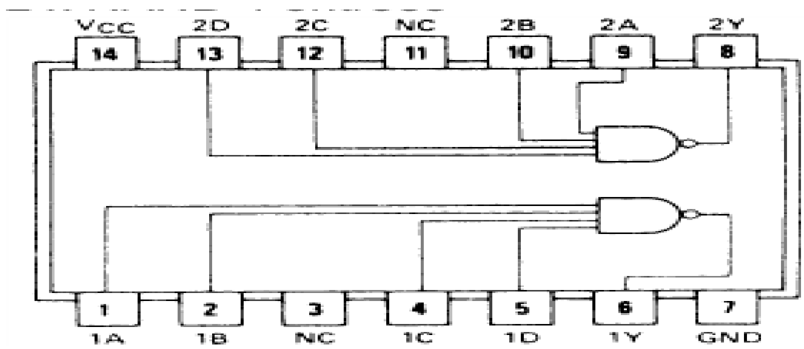


Figure A.4 : Circuit 7410

Brochage du circuit intégré 7420

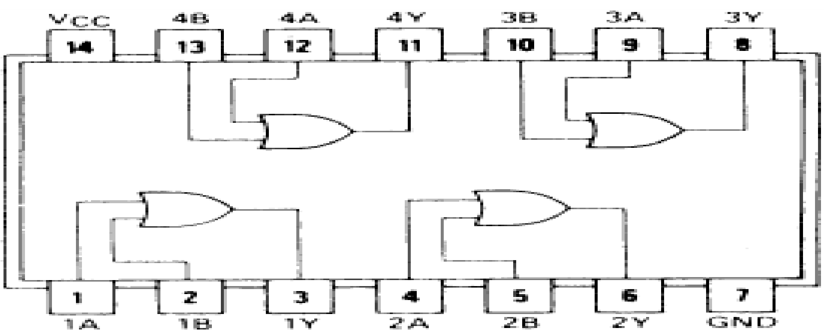
Le circuit 7420 illustré sur la Figure A.5 est le circuit qui peut effectuer un NON ET mais avec quatre (4) variables d'entrées.



<u>Figure A.5</u> : Circuit 7420

Brochage du circuit intégré 7432

Le circuit ci-dessous, qui comprend deux (2) entrées de variables dans quatre (4) portes logiques OU différentes est le 7432 (Fig. A.6) :



<u>Figure A.6</u> : Circuit 7432

Brochage du circuit intégré 7483

Quant au circuit ci-dessous, c'est un additionneur 4 bits (Fig. A.7)

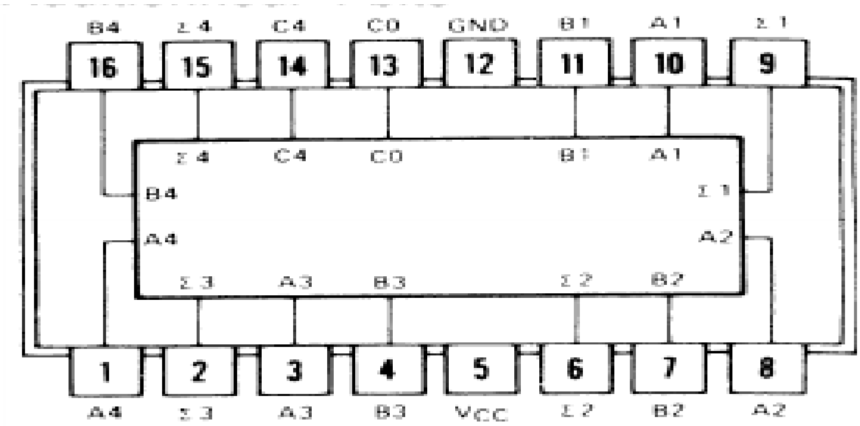


Figure A.7 : **Circuit 7483**

Brochage du circuit intégré 7486

Le 7486 est un circuit avec quatre portes logiques XOR, chacune à deux entrées.

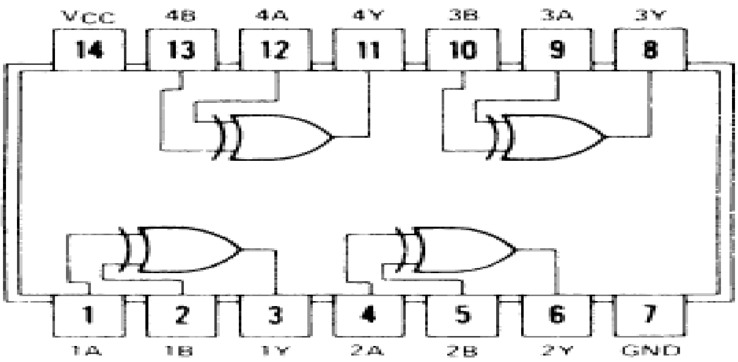


Figure A.8 : **Circuit 7486**

REFERENCES

[1] Logique combinatoire ; Cours en L2 ; Département Electronique ; ESP Antananarivo ; 2014-2015

[2] P. MAYE, Professeur agrégé de physique, Ingénieur en électronique et électromécanique, Enseignant en BTS d'électronique et d'électrotechnique à Arras, et Jacques Bouquet, Professeur d'électronique à Arras; « Electronique numérique en 26 fiches » ; Université d'ARRAS

[3] R. MERAI, R.MOREAU, L.ALLAY, J.-P.DUBOS, J.LAFARGUE, LEGOFF ; « Electronique numérique » ; BEP - BAC PRO

[4] Analyse et conception des circuits électroniques avec le logiciel ELECTRONICS WORKBENCH ; cours de 3ème année ; Département Electronique ; ESP Antananarivo ; 2015-2016