3.1 Introduction

Ce chapitre concerne l'étude et la réalisation de circuits logiques de fonctions bien précises.

L'étude concerne des fonctions de comparaison, d'addition et de soustraction binaires, des opérations de multiplexage et de démultiplexage, de transcodage et d'affichage numérique et en dernier la transcription d'un cahier des charges en circuit logique.

Les outils permettant d'assimiler ce chapitre sont bien évidemment présentés aux deux précédents chapitres.

3.2 Le comparateur binaire

3.2.1 Le comparateur à un bit

Il s'agit de comparer deux nombres binaires à un bit chacun.

Soient A et B deux nombres binaires, donc A et B ∈ $\{0,1\}$ et soit une trois fonctions logiques $S_{A<B}$, $S_{A=B}$ ou $S_{A>B}$

Traçons la TdV de ces 3 fonctions.

A	B	$S_{A<B}$	$S_{A=B}$	$S_{A>B}$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

On voit bien que d'un coté A et B sont vus comme des variables logiques et d'un autre coté, les sorties prennent l'état 1 ou 0 selon l'état de A et B numériques.

Retrouvons les expressions équivalentes algébriques des 3 fonctions :
$S_{A<B} = \bar{A}.B$, $S_{A=B} = A.B + \bar{A}.\bar{B} = \overline{A \oplus B}$ *et* $S_{A>B} = A.\bar{B}$

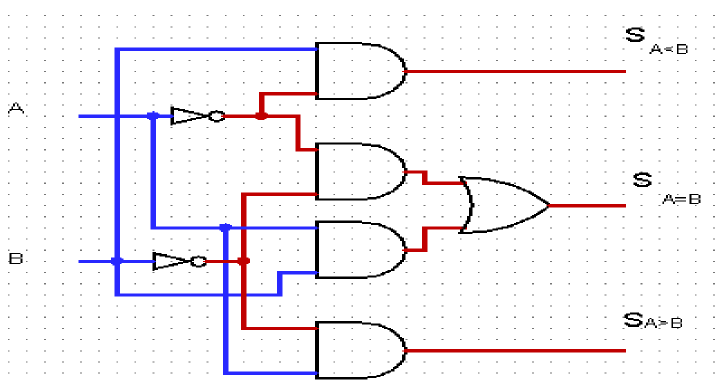

Fig 3.1 Comparateur à un bit

3.2.2 Le comparateur à deux bits (1ere version)

Cette fois-ci, les nombres à comparer A et B sont chacun sur 2bits. $A=a_1a_0$ et $B=b_1b_0$.

A et B $\in$ {00,01,10,11}

Traçons la TdV de ces 3 fonctions

a_1	a_0	b_1	b_0	$S_{A<B}$	$S_{A=B}$	$S_{A>B}$
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

On remarque que les états de chaque sortie varient d'un code de la table à un autre, il est donc nécessaire d'établir un tableau de Karnaugh pour chaque sortie afin d'alléger son expression algébrique et donc le logigramme.

XY\ZT	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

$S_{A<B} = \bar{a}_1\bar{a}_0b_0 + \bar{a}_1b_1 + \bar{a}_0b_1b_0$

XY\ZT	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

$S_{A=B} = \bar{a}_1\bar{a}_0\bar{b}_1\bar{b}_0 + \bar{a}_1a_0\bar{b}_1b_0 + a_1a_0b_1b_0 + a_1\bar{a}_0b_1\bar{b}_0$

XY\ZT	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

$S_{A>B} = a_0\bar{b}_1\bar{b}_0 + a_1\bar{b}_1 + a_1a_0\bar{b}_0$

Note : les variables du TK (tableau de Karnaugh) X, Y, Z et T sont équivalentes à: a_1, a_0, b_1, b_0 respectivement.

Il est clair que la sortie $S_{A=B}$ prend une forme canonique, tous les **1** ne sont pas adjacents l'un à l'autre.

Au moment d'établir son logigramme, on doit faire appel à 4 portes NAND à 4 entrées chacune, mais si on regarde bien les trois TK, on remarque qu'il y a égalité entre A et B lorsque A n'est pas inferieur à B et A n'est pas supérieur à B.

En traduisant cette dernière évidence en expression algébrique :

$S_{A=B} = (\overline{A < B})et(\overline{A > B}) = \bar{S}_{A<B}.\bar{S}_{A>B}$

et en utilisant le théorème de De Morgan :

$$\bar{S}_{A<B}.\bar{S}_{A>B} = \overline{S_{A<B} + S_{A>B}}$$

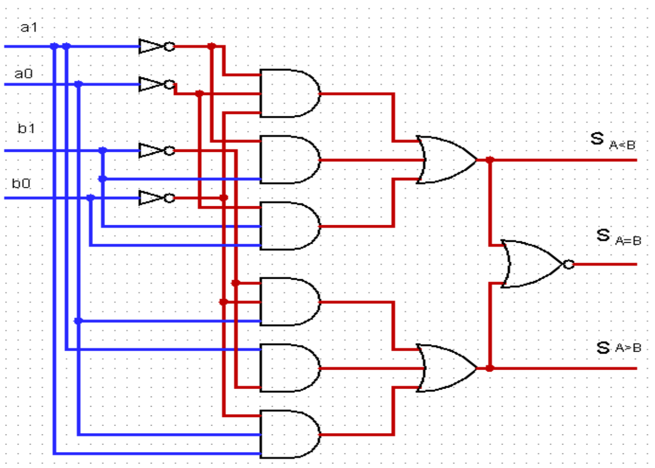

Fig 3.2 Comparateur à deux bits 1ère version

3.2.3 Le comparateur à deux bits (2ème version)

Il est clair que si les nombres à comparer A et B sont chacun sur 3bits. $A=a_2a_1a_0$ et $B=b_2b_1b_0$ ou A et B ∈ {000,001,010,011,100,101,110,111}, la TdV devient trop longue 2^6 codes différents alors que l'ensemble des nombres à comparer est limité.

A et B ∈ {0,1,2,3,4,5,6,7}.

- Sur la base d'un comparateur de deux nombres sur un bit chacun on peut concevoir un comparateur de deux nombres de deux bits chacun.

- Sur la base d'un comparateur de deux nombres sur un bit chacun on peut concevoir un comparateur de deux nombres de trois bits chacun et ainsi de suite.

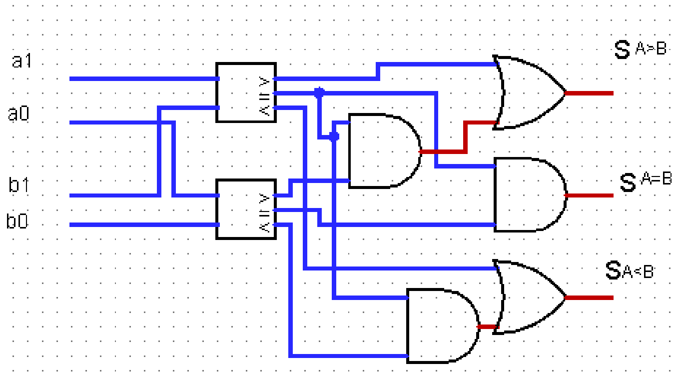

Fig 3.3 Comparateur à deux bits 2eme version

Les rectangles correspondent aux comparateurs de deux nombres sur un bit chacun détaillés en Figure 3.1.

A est superieur à B si a_1 est superieur à b_1 **<u>ou</u>** bien (OR $S_{A>B}$) a_1 est égal à b_1 **<u>et</u>** a_0 est superieur à b_0.

A est egal à B si a_1 est egal à b_1 **<u>et</u>** a_0 est egal à b_0.

A est inferieur à B si a_1 est inferieur à b_1 **<u>ou</u>** bien (OR $S_{A<B}$) a_1 est égal à b_1 **<u>et</u>** a_0 est inferieur à b_0.