

CHAPITRE 3 : LES CIRCUITS COMBINATOIRES

3.1 Introduction

Le but de ce chapitre est d'étudier les principaux circuits logiques combinatoires, connu aussi sous le nom de circuits indépendant de l'horloge (non utilisée dans le fonctionnement). Ils sont utilisés dans les systèmes numériques (tels que : les circuits arithmétiques, les codeurs, les additionneurs, ...), ainsi que la réalisation de quelques fonctions logiques utilisant des portes logiques.

3.2 Définition d'un circuit combinatoire

Un circuit combinatoire est un circuit logique où ses *sorties* dépendent seulement de ses *entrées*, c.-à-d. Il ne doit pas y avoir de feedback entre la sortie et l'entrée, donc, il n'utilise aucune mémoire, l'état précédent de l'entrée n'a aucun effet sur l'état actuel du circuit.

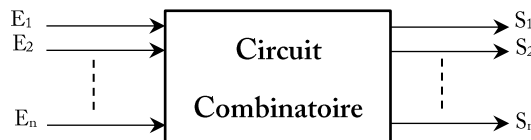


Figure 3.1 : Circuit combinatoire.

- Circuits SSI (Small Scale Integration) : portes de petite taille AND, OR, XOR.
- Circuits MSI (Medium Scale Integration): décodeurs, encodeurs, multiplexeurs, démultiplexeurs, additionneurs.

3.3 Quelques circuits combinatoires

3.3.1 Additionneur

3.3.1.1 Demi -Additionneur

Le demi additionneur permet d'effectuer la somme arithmétique de deux nombres *A* et *B sur un bit*. Il a deux sorties : la somme *S* et la retenue *R* (ou *C* : Carry).

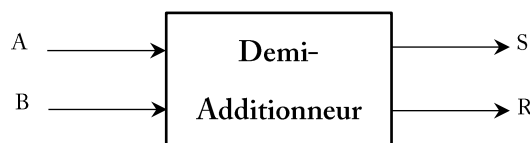


Figure 3.2 : Demi -Additionneur.

L'addition en binaire sur un seul bit s'effectue comme suit :

	$\begin{array}{r} \hline 0 + 0 = 00 \\ \hline 0 + 1 = 01 \\ \hline 1 + 0 = 01 \\ \hline 1 + 1 = 10 \\ \hline \end{array}$	<table border="1" style="border-collapse: collapse;"> <thead> <tr> <th>A</th> <th>B</th> <th>R</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	R	S	0	0	0	0	0	1	0	1	1	0	0	1	1	1	1	0
A	B	R	S																			
0	0	0	0																			
0	1	0	1																			
1	0	0	1																			
1	1	1	0																			

Les fonctions de sortie *R* et *S* sont égales à: Table 3.1 : Table de vérité du demi-additionneur.

- $R = A B$
- $S = \bar{A}B + A\bar{B} = A \oplus B$

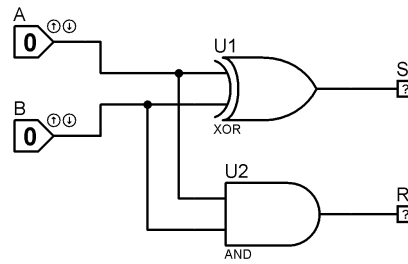


Figure 3.3: Logigramme du demi-additionneur.

3.3.1.2 Additionneur complet 1 bit

L'additionneur complet prend en considération la *retenue entrante*. Il possède 3 entrées :

- A_i : le 1^{er} nombre sur un bit.
- B_i : le 2^{ème} nombre sur un bit.
- R_{i-1} : la retenue entrante sur un bit.

et deux sorties :

- S_i : la somme.
- R_i : la retenue sortante.

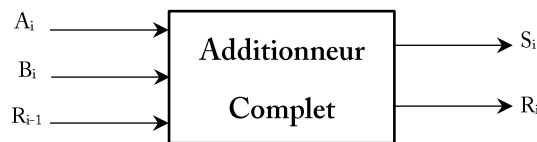


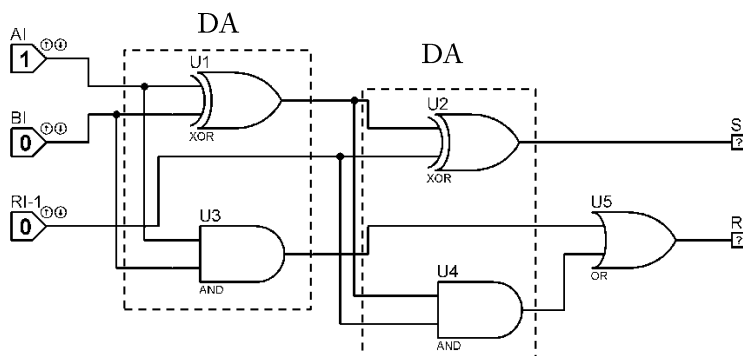
Figure 3.4 : Additionneur complet.

Son principe est comme suit :

R_4	R_3	R_2	R_1	$R_0=0$
A_4	A_3	A_2	A_1	A_0
B_4	B_3	B_2	B_1	B_0
R_4	S_4	S_3	S_2	S_1

Donc, les fonctions de sorties, le logigramme et la table de vérité sont comme suit :

- $S_i = \bar{A}_i \bar{B}_i R_{i-1} + \bar{A}_i B_i \bar{R}_{i-1} + A_i \bar{B}_i \bar{R}_{i-1} + A_i B_i R_{i-1} = A_i \oplus B_i \oplus R_{i-1}$
- $R_i = \bar{A}_i B_i R_{i-1} + A_i \bar{B}_i R_{i-1} + A_i B_i \bar{R}_{i-1} + A_i B_i R_{i-1} = A_i B_i + R_{i-1} (A_i \oplus B_i)$



A_i	B_i	R_{i-1}	R_i	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figure 3.5: Logigramme de l'additionneur complet.

Table 3.2 : Table de vérité de l'additionneur complet.

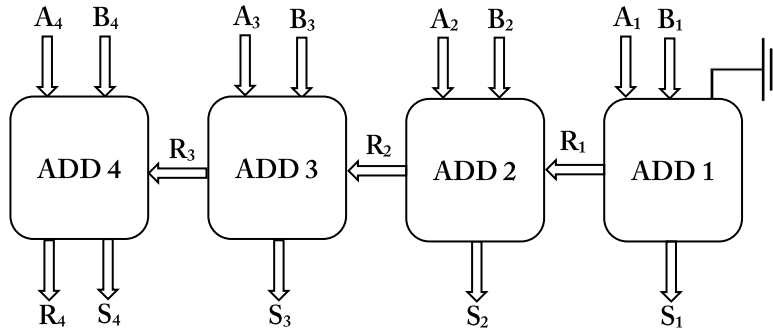


Figure 3.6: Schéma d'un additionneur 4 bits.

3.3.2 Soustracteur

3.3.2.1 Demi-soustracteur

La table de vérité pour un demi-soustracteur (qui ne prend pas en compte la retenue résultant des bits de poids inférieurs) est la suivante :

Où, S représente le résultat de la soustraction $A - B$ et R la retenue.

Ce qui donne les expressions logiques de sorties suivantes :

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

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

Table 3.3 : Table de vérité du semi soustracteur.

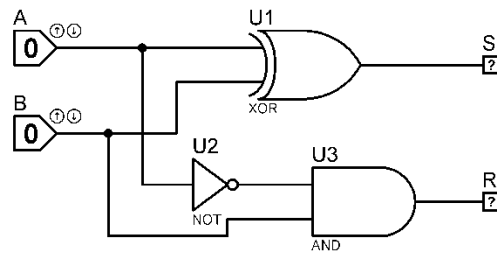


Figure 3.7: Logigramme d'un demi-soustracteur.

3.3.2.2 Soustracteur complet

Il est complet car par rapport au demi-soustracteur, car il tient compte de l'emprunt fait à l'étage précédent.

- $S_i = (A_i \oplus B_i) \oplus R_{i-1}$
- $R_i = B_i R_{i-1} + \bar{A}_i R_{i-1} + \bar{A}_i B_i$

A_i	B_i	R_{i-1}	S_i	R_i
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

Table 3.4 : T. V. du soustracteur complet.

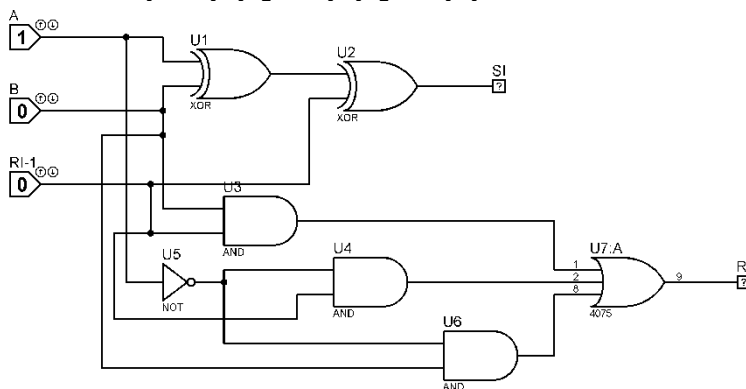


Figure 3.8: Logigramme d'un soustracteur complet.

3.3.3 Comparateur

3.3.3.1 Comparateur 1 bit

C'est un circuit logique combinatoire qui fait la comparaison entre deux nombres binaire A et B.

Il comprend 2 entrées :

- A : le 1^{er} nombre sur un bit.
- B : le 2^{ème} nombre sur un bit.

et 3 sorties :

- S_e : Sortie égalité ($A=B$).
- S_i : Sortie inférieur ($A < B$).
- S_s : Sortie supérieur ($A > B$).

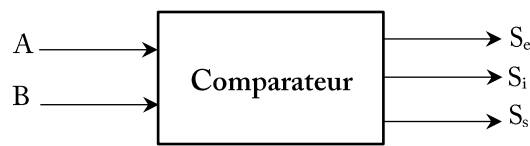


Figure 3.9: Comparateur.

- $S_s = A\bar{B}$
- $S_i = \bar{A}B$
- $S_e = \bar{A}\bar{B} + AB = \overline{A \oplus B} = \bar{S}_s + \bar{S}_i$

A	B	S_e	S_i	S_s
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

Table 3.5 : Table de vérité du comparateur.

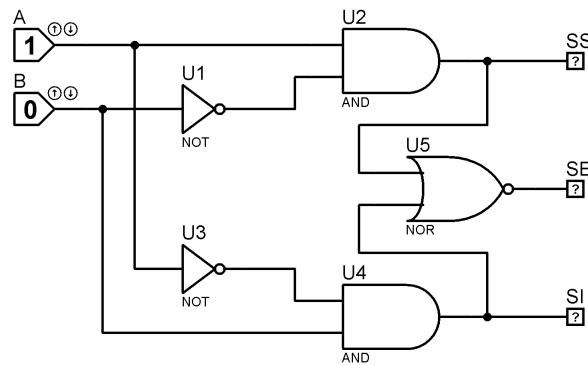


Figure 3.10: Logigramme du comparateur.

3.3.3.2 Comparateur 2 bits

Il fait la comparaison entre deux nombres sur deux bits: A ($a_2 a_1$) et B ($b_2 b_1$).

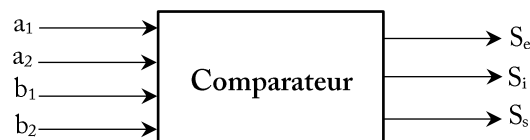


Figure 3.11: Comparateur 2 bits.

Règles :

- $A=B$ si $a_2=b_2$ et $a_1 = b_1$
 $S_e = \overline{a_2 \oplus b_2} + \overline{a_1 \oplus b_1}$
- $A < B$ si $a_2 < b_2$ ou $(a_2 = b_2$ et $a_1 < b_1)$
 $S_i = \overline{a_2}b_2 + \overline{a_2 \oplus b_2} \cdot (\overline{a_1}b_1)$
- $A > B$ si $a_2 > b_2$ ou $(a_2 = b_2$ et $a_1 > b_1)$
 $S_s = a_2\overline{b_2} + \overline{a_2 \oplus b_2} \cdot (a_1\overline{b_1})$

a ₂	a ₁	b ₂	b ₁	S _e	S _i	S _s
0	0	0	0	1	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	1	0	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	1	0	0
1	0	1	1	0	1	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	1	0	0

Table 3.6 : Table de vérité du comparateur 2 bits.

3.3.4 Multiplexeur

Un multiplexeur est un circuit combinatoire qui permet de sélectionner une information (1 bit) parmi 2ⁿ valeurs en entrée (l'entrée sélectionnée est définie par son adresse) et conduire cette entrée à une sortie unique.

Il possède :

- 2ⁿ entrées d'information.
- Une seule sortie.
- N entrées de sélection (commandes).

C ₁	C ₀	S
0	0	E ₀
0	1	E ₁
1	0	E ₂
1	1	E ₃

Table 3.7 : Table de vérité du MUX 4 :1.

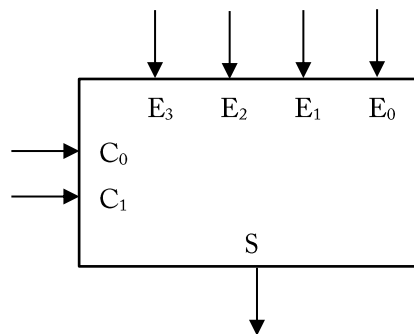


Figure 3.12: Multiplexeur 4 :1.

$$S = \overline{C_1} \overline{C_0}(E_0) + \overline{C_1}C_0(E_1) + C_1\overline{C_0}(E_2) + C_1C_0(E_3)$$

3.3.5 Demultiplexeur

Il fait l'inverse du multiplexeur, en conduisant une information d'entrée E dans l'une des sorties (S₀, S₁,...) selon les valeurs des entrées de commandes C₁ et C₀.

La table de vérité est comme suit :

C ₁	C ₀	S ₃	S ₂	S ₁	S ₀
0	0	0	0	0	E
0	1	0	0	E	0
1	0	0	E	0	0
1	1	E	0	0	0

Table 3.8 : Table de vérité du DEMUX 1 :4.

- $S_0 = \bar{C}_1 \bar{C}_0(E)$
- $S_1 = \bar{C}_1 C_0(E)$
- $S_2 = C_1 \bar{C}_0(E)$
- $S_3 = C_1 C_0(E)$

3.3.6 Décodeur

C'est un circuit qui fait la conversion d'un code binaire de n bits à un code de m bits ($n \leq m \leq 2^n$), où chaque combinaison d'entrées n'active qu'une seule sortie à la fois.

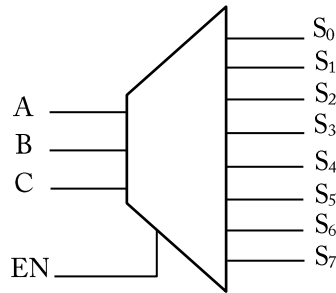


Figure 3.14: Décodeur 3 :8.

La majorité des décodeurs ont une entrée de validation (enable).

- Si $EN = 0$: sorties égales à 0.
- Si $EN = 1$: fonctionnement normale.

EN	A	B	S ₀	S ₁	S ₂	S ₃
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Table 3.9 : Table de vérité du Décodeur 2 :4.

Les fonctions de sorties sont comme suit :

- $S_0 = (\bar{A} \cdot \bar{B}) \cdot EN$
- $S_1 = (\bar{A} \cdot B) \cdot EN$
- $S_2 = (A \cdot \bar{B}) \cdot EN$
- $S_3 = (A \cdot B) \cdot EN$

3.3.7 Encodeur

Il fait le rôle inverse du décodeur, il a 2^n entrées et n sorties. Pour chaque combinaison en entrée, la sortie donne l'équivalent en binaire.

E ₀	E ₁	E ₂	E ₃	A	B
0	0	0	0	0	0
1	X	X	X	0	0
0	1	X	X	0	1
0	0	1	X	1	0
0	0	0	1	1	1

Table 3.10 : Table de vérité de l'encodeur 4 :2.

- $A = \bar{E}_0 \bar{E}_1 (E_2 + E_3)$
- $B = \bar{E}_0 (E_1 + \bar{E}_2 E_3)$

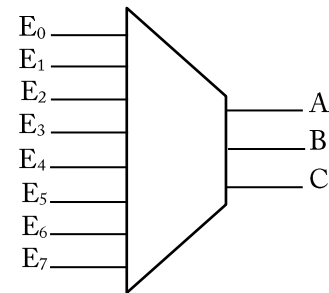


Figure 3.15: Encodeur 8:3.

3.3.8 Transcodeur

C'est un circuit combinatoire qui permet de transformer un code X (sur n bits) en entrée en un code Y (sur m bits) en sortie.

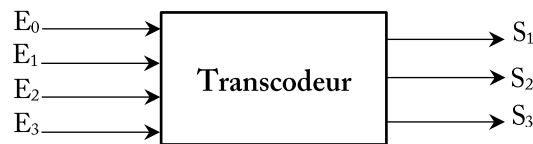


Figure 3.16: Transcodeur 4 entrées - 3 sorties.

Exemple :

Soit un transcodeur (4 :4) qui fait la transformation binaire naturel en code binaire réfléchi (Gray).

A partir de la table de vérité ci-dessous, on peut avoir les fonctions de sorties suivantes :

- $G_1 = B_1 \bar{B}_2 + \bar{B}_1 B_2 = B_1 \oplus B_2$
- $G_2 = B_2 \bar{B}_3 + \bar{B}_2 B_3 = B_2 \oplus B_3$
- $G_3 = B_3 \bar{B}_4 + \bar{B}_3 B_4 = B_3 \oplus B_4$
- $G_4 = B_4$

Nombre décimal	B ₄	B ₃	B ₂	B ₁	G ₄	G ₃	G ₂	G ₁
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Table 3.11 : Table de vérité du transcodeur 4 bits binaire-Gray.