

Module : ~~Maths~~ logique

**TP n° 8. Circuits de demi-additionneur  
et additionneur complet**

PARTIE THEORIQUE:

### Objectif

Objectif  
Comprendre les caractéristiques d'un demi-additionneur et d'un additionneur complet dans un organe de calcul.

## Discussion

Discussion

Les additionneurs peuvent être séparés en < demi-additionneur > et < additionneur complet >. Les demi-additionneurs suivent les règles de l'addition binaire et réalisent seulement l'addition de deux bits. Le résultat d'une addition est composé d'une < retenue > et d'une < somme >. Dans les additions binaires, on a une retenue < Carry > lorsque la somme est supérieure à 1, comme on peut le voir dans l'exemple d'addition de demi-additionneur ci-dessous :

$$\begin{array}{r}
 1 \\
 + \quad 1 \\
 \hline
 1 \quad 0
 \end{array}
 \quad
 \begin{array}{r}
 1 \text{ retenue précédente} \\
 1 \quad 0 \\
 + \quad 1 \quad 0 \\
 \hline
 1 \quad 0 \quad 1
 \end{array}$$

Retenue      Somme

Lorsqu'on additionne  $\langle 1 \rangle$  et  $\langle 1 \rangle$ , la somme vaut 0 et la retenue vaut 1. Le dernier additionneur est limité à l'addition 2 chiffres de 1 bit de long chacun.

Un additionneur complet peut réaliser des additions de nombres de plus de 1 bit long.

long. Pour réaliser des additions de nombres de plusieurs bits de long on utilisera connexion représentée sur la figure (a), ou < entrée parallèle >, qui permet de gérer des sommes de manière simultanée.

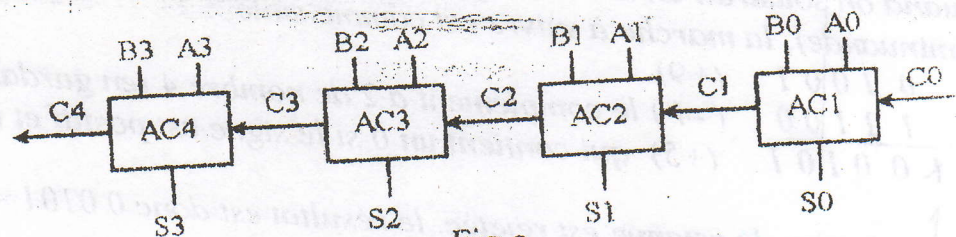


Fig a



Toutefois, la somme de l'additionneur suivant sera stable seulement une fois que la retenue de l'additionneur précédent se sera stabilisée. Par exemple, d'après la figure (1), la somme de AC2 ne sera stable que lorsque la retenue de AC1 sera stable.

Quand AC1 additionne A1 et B1, cela génère une somme S1 et une retenue C1. Cette retenue C1 va être ajoutée à A2 et B2 par AC2, générant une nouvelle somme S2 et une nouvelle retenue C2. dans le cas de la figure (1), les sommes des quatre additionneurs ne vont pas se stabiliser en même temps, l'opération d'addition va donc être retardée. Ce retard peut être supprimé en utilisant un additionneur générateur de retenue anticipée.

Les additionneurs binaires peuvent être convertis en additionneurs BCD (décimal codé en binaire). Etant donné que le code BCD possède 4 bits et admet comme plus grand nombre le 9, alors que le plus grand nombre binaire de quatre bits est le 16, il y a donc une différence de 6 entre l'additionneur binaire et l'additionneur BCD. Si on utilise des additionneurs binaires pour additionner des codes BCD, il faudra donc rajouter un 6 pour les situations suivantes :

- 1- Dès qu'il y a une retenue.
- 2- Quand la somme est supérieure à 9.

Le circuit intégré 7483 remplit cette fonction. Le premier additionneur réalise la somme des deux nombres en binaire et le deuxième réalise la correction si nécessaire pour avoir un résultat compatible BCD. Si  $F_2 = 1$  c.à.d. une au moins des deux conditions de correction est vérifiée alors B2 et B3 du deuxième additionneur connectés à 1 prendraient la valeur 1 et B4 B3 B2 B1 = 0 1 1 0 < 6 > qui est ajouté au résultat du premier additionneur. Si  $F_2 = 0$  pas besoin de correction le nombre B4 B3 B2 B1 = 0000 rajouter est nul.

#### Notation en complément à 2:

Le complément à 2 d'un nombre binaire s'obtient simplement en prenant le complément à 1 de ce nombre et en ajoutant 1 au bit de rang de poids le plus faible exemple :

$$\begin{array}{r}
 101101 \\
 010010 \\
 + \quad 1 \\
 \hline
 = 010011
 \end{array}$$

équivalent binaire de 45.  
complément de chaque bit pour obtenir le complément à 1.  
addition de 1 pour obtenir le complément à 2.  
le complément à 2 du nombre binaire initial.

#### Soustraction en complément à 2:

Quand on soustrait un nombre binaire (le diminueur) d'un autre nombre binaire (le diminuande), la marche à suivre est comme suite:

$$\begin{array}{r}
 01001 \quad (+9) \\
 + 11100 \quad (-4) \text{ le complément à 2 de nombre 4 (en gardant toujours le bit de signe)} \\
 \hline
 = 100101 \quad (+5) \text{ qui contient un 0 si le signe est positif et un 1 si le signe est négatif}
 \end{array}$$

la retenue est rejetée, le résultat est donc 0 0101 = +5.



## Travail a préparer :

### a) demi-additionneur :

1) Réaliser les séquences d'entrée indiquée pour A et B par le tableau 1 et trouver les états de sortie S (somme) et R (retenue).

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

Tableau 1

2) Donner les équations algébriques des sorties S (somme) et R (retenue) en fonction de A et B et tracer leurs logigrammes.

### b) demi-soustracteur :

1) Réaliser les séquences d'entrée indiquée pour A et B par le tableau 2 et trouver les états de sortie D (différence) et R (retenue)

A	B	D	R
0	0		
0	1		
1	0		
1	1		

Tableau 2

2) Donner les équations algébriques des sorties D (différence) et R (retenue) en fonction de A et B et tracer leurs logigrammes.

### c) additionneur complet :

1) Réaliser les séquences d'entrée indiquée pour A, B et C par le tableau 3 et trouver les états de sortie S (somme) et R (retenue).

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

Tableau 3

2) Donner les équations algébriques des sorties S (somme) et R (retenue) en fonction de A, B et C et tracer leurs logigrammes.



d) soustracteur complet :

1) Réaliser les séquences d'entrée indiquée pour A, B et C par le tableau 4 et trouver les états de sortie D (différence) et R (retenue)

A	B	C	D	R
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Tableau 4

2) Donner les équations algébriques des sorties D (différence) et R (retenue) en fonction de A, B et C et tracer leurs logigrammes.

PARTIE EXPERIMENTALE :

**A/ Réalisation d'un demi-add (demi-soust) et d'un add complet (soust complet) à partir de portes logiques:**

1) Alimenter les CI 7400, 7404, 7486. en reliant la broche 14 à +5V de l'alimentation fixe afin de réaliser la tension + Vcc des circuit TTL et la broche 7 à la masse (0V).

2) Relier les entrées A, B et C (dans le cas d'une addition complète) aux commutateurs sur le pupitre

Rq : Les entrées A et B représentent les termes de l'addition, et C la retenue précédente

3) Connecter les sorties S (somme) ou D (différence) et R (retenue) aux indicateurs logique.

4) Réaliser le montage d' :

- a) un demi-additionneur
- b) un demi-soustracteur
- c) un additionneur complet
- d) un soustracteur complet

5) Vérifier à chaque fois les résultats obtenus aux tableaux 1, 2, 3 et 4.

**B/ Circuit d'additionneur complet à partir d'un circuit intégré 7483:**

Cas d'une addition : Connecter Y4 à 0 (c.à.d à la masse).

Connecter les entrées  $x_3, x_2, x_1, x_0$  aux sorties des commutateurs respectivement 1, 2, 3, et 4 les entrées  $y_0, y_1, y_2, y_3$  aux sorties des commutateurs 5, 6, 7 et 8.

Connecter les sorties  $S_3, S_2, S_1, S_0$  aux indicateurs logiques ainsi que la sortie de la retenue R.

Réaliser les séquences d'entrée indiquées dans le tableau 5 et relever les états de sorties en nombres hexadécimaux.

$X (x_3x_2x_1x_0)$	$Y (y_3y_2y_1y_0)$	$S (S_3S_2S_1S_0)$	$R (F1)$
0	0		
6	3		
F	0		
3	1		
B	1		
8	4		
F	4		
7	B		
9	9		
E	C		
F	F		

Tableau 5

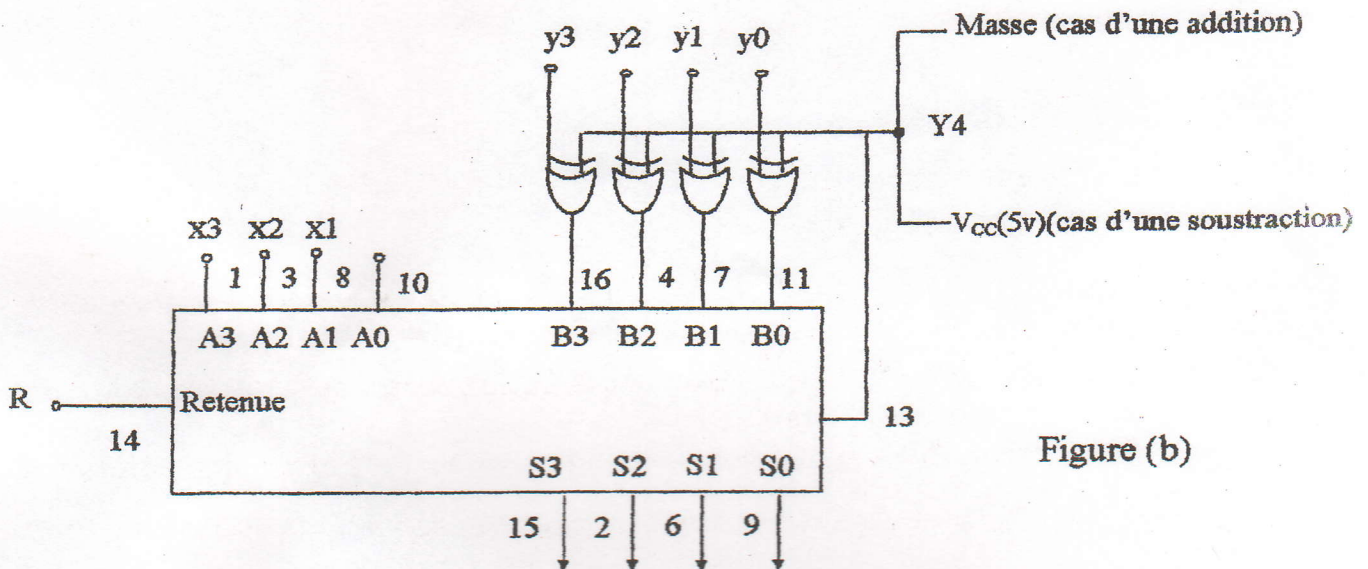


Figure (b)

### C/ Circuit additionneur soustracteur en complément à 2 :

Cas d'une soustraction : Connecter Y4 à 1 (c.à.d à 5V).

On voudrait utiliser l'additionneur 7483 comme additionneur-soustracteur, selon la valeur d'une variable de sélection Y4, on utilisera bien sur la notation du complément à 2 (voir le cours), on complètera le montage de la figure (3) avec celui de la figure (1), on trouvera le résultat sur 5 bits. Pour Y4 = 1 il s'agit d'une soustraction (A-B), Y4 = 0 c'est une addition et on supposera que A > B,

- Vérifiez les résultats pour les soustractions 7 - 5, 4 - 1, 3 - 2, 8 - 3.
- Que proposez vous si B > A exemples 3 - 8, 2 - 7.
- Commentez les résultats.



Départements : Automatique  
Electronique  
Electrotechnique

Module *logique*

Travaux Pratiques N° 3

**Codeurs et multiplexeurs**

**I- Partie théorique :**

**Objectif:**

Comprendre les principes de fonctionnement des circuits codeurs et multiplexeurs

**Débat:**

**A - Circuit codeur :**

Un circuit codeur est un circuit combinatoire qui accepte une ou plusieurs entrées et génère un code de sortie spécifique. Une seule entrée est déclenchée (activée) à la fois. Par exemple le codeur octal-vers-binaire qui possède 8 (A0 à A7) entrées une pour chaque chiffre octal (0 à 7) et 3 sorties A, B et C qui désignent le code binaire de l'entrée activée parmi les huit <si par exemple A5 est activée on aura en sortie ABC = 101>.

Le codeur est désigné aussi par le nombre de ces variables d'entrée et de sortie, et donc le codeur ci-dessus est un codeur 8 lignes vers 3 « figure 1 ».

**A - 1 / Codeur matriciel:**

Si on veut un codeur avec des caractéristiques particulières qu'on ne trouve pas sur le marché, il est possible d'en réaliser un avec des diodes. On peut constater figure 2 que si E1 est déclenchée on aura en sortie Y3 Y2 Y1 Y0 = 1011, pour E2 on aura 0110 ...

Une seule entrée doit être déclenchée à la fois pour un fonctionnement correct, par exemple E1 et E2 sont tous les deux formés on aura en sortie 1111, si c'est E1 et E4 enclenchés on aura 1011 fig.2. Ils existent des circuits codeurs qui acceptent plusieurs entrées en même temps. dans ce cas un ordre de priorité est attribué à chaque entrée c'est la plus prioritaire qui sera prise en considération.

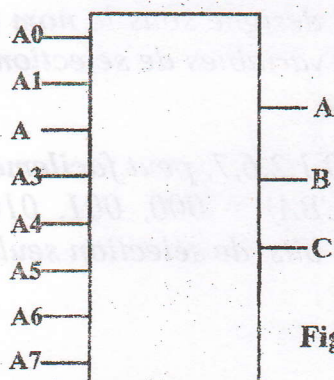


Fig 1

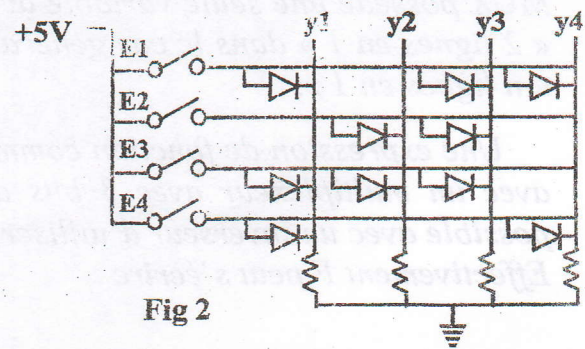
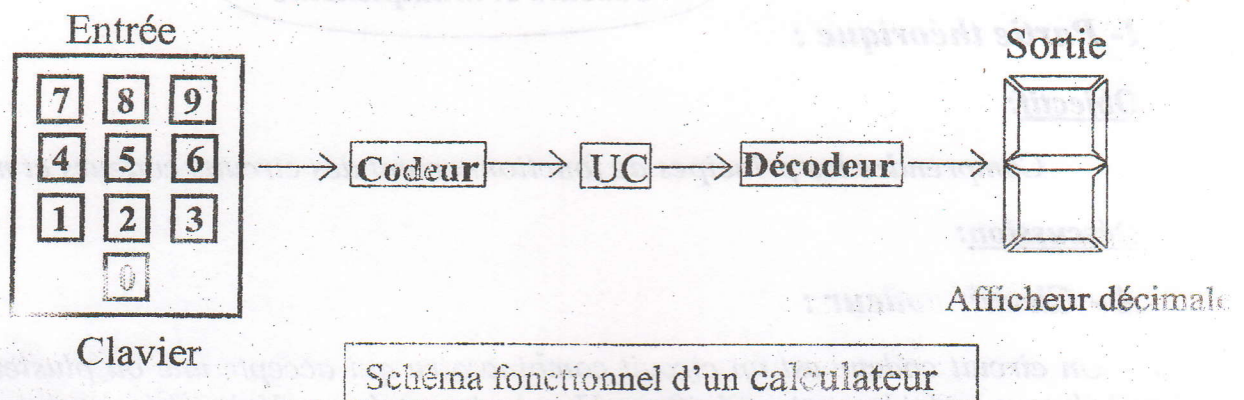


Fig 2



Considérons le schéma fonctionnel simple d'une calculatrice, l'unité d'entrée constituée par le clavier courant. Entre le clavier et l'unité centrale est introduit un codeur dont la fonction est de traduire en binaire, le nombre entré au clavier (décimal).

L'unité centrale effectue le traitement en binaire et émet un résultat en binaire. Le décodeur traduit le résultat binaire en un code particulier qui active les segments appropriés de l'afficheur à 7 segments. par conséquent, le décodeur traduit le binaire en décimal.



#### A - 2 / Codeur avec priorité:

Un codeur avec priorité traite les entrées en fonction de leur priorité, de leurs poids. Lorsqu'on déclenche une entrée de poids élevé, la sortie va correspondre à cette entrée sans tenir compte des états des entrées de moindre poids. Le circuit intégré 74147 est un codeur à sortie BCD à priorité 9 ~1. la priorité des entrées se fait dans l'ordre croissant, la porte 1 à le poids le plus faible et la porte 9 le poids le plus haut. Les sorties sont en code BCD.

Le 74147 est déclenché par le niveau BAS « 0 » des entrées, et les sorties aussi.

#### B - Circuits multiplexeurs :

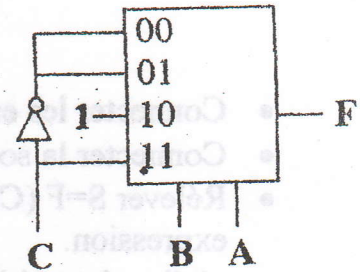
Un multiplexeur, ou MUX, est un circuit logique qui choisit et conduit n'importe quel nombre d'entrée en une seule sortie. Une des nombreuses entrées est sélectionnée par les variables de sélection « ou d'adresse » puis conduite à l'unique sortie, le nombre de variables de sélection détermine la capacité du multiplexeur. Par exemple, si un certain MUX possède une seule variable de sélection, il est désigné sous le nom de multiplexeur « 2 lignes en 1 » dans le cas général s'il possède n variables de sélection c'est un MUX « n lignes en 1 ».

Une expression de fonction comme  $F(CBA) = \Sigma 0,1,2,6,7$ , peut facilement être réalisée avec un multiplexeur avec 3 bits de sélection  $F(CBA) = 000, 001, 010, 111$  et il est possible avec un inverseur d'utiliser un MUX avec 2 bits de sélection seulement. Effectivement F peut s'écrire :



Effectivement F peut s'écrire :

$$F(CBA) = \bar{C}(\bar{B}\bar{A} + \bar{B}A + B\bar{A}) + C(\bar{B}\bar{A} + BA) = \bar{B}\bar{A} + CBA + \bar{C}(\bar{B}\bar{A} + BA)$$



## II- Partie expérimentale :

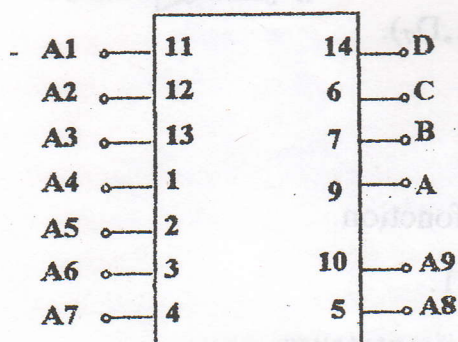
### Matériels et composants nécessaires:

Pupitre de logique numérique, les CI TTL 74147 , 74151 , 7400 et 7404.

### Procédure:

#### 1) Codeur : Manipulation du codeur 10 vers 4 ( le 74147 TTL)

- 1) Connecter la patte 16(Vcc) du CI 74147 à la sortie 5V de l'alimentation fixe.
- 2) Relier la masse patte 8 (GND) du même CI à celle du pupitre.
- 3) Connecter les entrées A<sub>1</sub>, A<sub>2</sub>, ....., A<sub>9</sub> aux commutateurs (Switches).
- 4) Connecter les sorties DCBA aux indicateurs logiques (LED).
- 5) Réaliser la séquence d'entrée indiquée par le tableau 1, et relever les états de sortie.

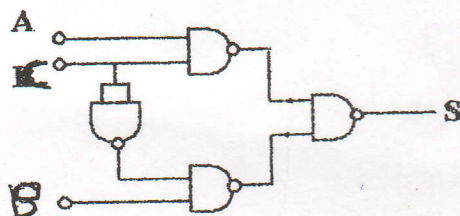


A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	DCBA
0	1	1	1	1	1	1	1	1	
0	0	1	1	0	1	1	1	1	
1	1	1	1	1	1	1	1	0	
1	1	1	1	1	1	1	0	0	
1	1	1	1	1	1	0	1	1	
1	1	0	0	0	1	0	1	1	
1	0	0	0	0	0	1	1	1	

#### 2) Multiplexeur :

##### a) Réalisation d'un multiplexeur 2 lignes en 1 :

- Connecter la patte 14(Vcc) du CI 7400 à la sortie 5V de l'alimentation fixe
- Relier la masse patte 7 (GND) du même CI à celle du pupitre.
- Realiser le montage suivant :





- Connecter les entrée A,B et le sélecteur C aux commutateurs (Swiches).
- Connecter la sortie S à un indicateur logique LED.
- Rélever  $S=F(CBA)$  pour toutes les combinaisons de CBA et donner son expression.
- réaliser le multiplexeur correspondant .

**b) Utilisation de multiplexeur pour réaliser des fonctions :**

On va utiliser le 74151 pour réaliser la fonction suivante :

$$F(DCBA) = \sum 0,2,4,5,7,8,10,11,15$$

- Déterminer à partir de la table de vérité de cette fonction l'expression algébrique correspondente .
- Montrer qu'on peut écrire la fonction F de la forme suivante :

$$F(DCBA) = (D * \alpha) + (\bar{D} * \beta) + (1 * \delta) + (0 * \lambda) .$$

Telsque :

$\alpha$  : équivalente à une seule entrée parmi les huit ( $D_0, D_1, \dots, D_7$ ).

$\beta$  : équivalente à deux entrées // // //

$\delta$  : équivalente à trois entrées // // //

$\lambda$  : équivalente à deux entrées // // //

- Donner le schéma du multiplexeur qui vérifit cette fonction.
- Réaliser ce MUX en utilisant le circuit logique 74151.
- Connecter les entrées DCBA respectivement aux commutateurs.

**RQ:** Puisque le CI 74151 ne supporte que 8 combinaisons d'entrée, on utilisera donc un inverseur pour D.

- Connecter STROBE (entrée de validation) à « 0 » et la sortie Y à l'indicateur logique (LED).
- Relever la sortie Y pour les 16 combinaisons d'entrée ,est ce qu'elle correspond au resultat de la table de verité de la fonction  $F(DCBA)$ .
- Que se passe-t-il si STROBE est connecter à « 1 ».

