

Séquence : 07

Document : TP01

Lycée Dorian

Renaud Costadoat

Françoise Puig



## Avec Correction

### Fonctions combinatoires



Référence	S07 - TP01
Compétences	A3-04: Identifier et décrire les liens entre les chaînes fonctionnelles. A3-07: Analyser un algorithme. E2-01: Choisir un outil de communication adapté à l'interlocuteur.
Description	Conception d'un algorithme de calcul d'addition pour une calculatrice
Système	Calculatrice de bureau

## 1 Présentation du système

Le système étudié est une calculatrice de bureau utilisée pour des opérations financières ou comptables.

Le but de ce TP est de modéliser, de simuler et de valider deux solutions technologiques pour la réalisation d'un additionneur :

- logique câblée (simulée sous Matlab/Simulink),
- logique programmée (simulée sous python)

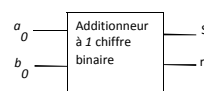


Des fichiers sont donnés afin de démarrer d'exercice :

- Un fichier Simulink : [Calculatrice vierge.slx](#),
- Un fichier Python : [Calculatrice vierge.py](#).

## 2 Additionner deux nombres de 1 bit

Nous allons tout d'abord réaliser un additionneur à 1 chiffre binaire. C'est un composant ayant deux entrées logiques (les deux chiffres binaires à additionner) notés  $a_0$  et  $b_0$ .



La sortie (résultat de l'opération arithmétique d'addition) est un nombre binaire à deux chiffres la somme  $s_0$  et la retenue  $r_1$  qui réalise l'opération suivante :

$$\begin{array}{r} b_0 \\ + a_0 \\ \hline r_1 \quad s_0 \end{array} \quad \text{qui se traduit par} \quad \begin{array}{r} 0 \\ + 1 \\ \hline 0 \quad 1 \end{array}$$

**Question 1** Etablir la table de vérité de la somme  $s_0$  et la retenue  $r_1$  en fonction des entrées  $b_0$  et  $a_0$ .

$b_0$	$a_0$	$s_0$	$r_1$
0	0		
0	1		
1	0		
1	1		

**Question 2** Etablir les équations logiques de la somme  $s_0$  (en utilisant un opérateur logique remarquable) et la retenue  $r_1$  en fonction des entrées  $b_0$  et  $a_0$ . Vérifier le résultat à l'aide des logiciels de simulation.

## 3 Additionner deux nombres de n bit

**Question 3** Etablir la table de vérité de la somme  $s_1$  (en utilisant un opérateur logique remarquable) et la retenue  $r_2$  en fonction des entrées  $b_1$ ,  $a_1$  et  $r_1$ .

$b_1$	$a_1$	$r_1$	$s_1$	$r_2$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

**Question 4** Établir les équations logiques de la somme  $s_1$  et la retenue  $r_2$  en fonction des entrées  $b_1$ ,  $a_1$  et  $r_1$ . Vérifier le résultat à l'aide des logiciels de simulation.

**Question 5** En déduire les équations logiques de la somme  $s_i$  au rang  $i$  (en utilisant un opérateur logique remarquable) et la retenue  $r_{i+1}$  au rang  $i + 1$  en fonction des entrées  $a_i$ ,  $b_i$  et  $r_i$  au rang  $i$ .

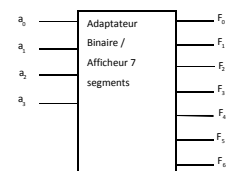
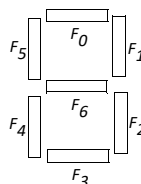
**Question 6** Tracer le schéma de l'ensemble du système de traitement de l'information.

**Question 7** Compléter le fichier python afin d'obtenir le même fonctionnement en logique programmée.

## 4 Afficher l'information sur un afficheur 7 segments

Un afficheur « 7 segments » est constitué de 7 diodes électroluminescentes (LEDs) numérotées de 0 à 6. L'objectif de cette partie est de concevoir un adaptateur qui convertit un nombre binaire pour l'afficher sur un afficheur 7 segments.

L'adaptateur se présente sous la forme d'un opérateur à 4 entrées binaires  $a_3$ ,  $a_2$ ,  $a_1$ ,  $a_0$  et 7 sorties binaires  $F_i$  avec  $i \in [0, 6]$ .



**Question 8** Remplir les colonnes  $F_i$  de la table de vérité.

$a_3$	$a_2$	$a_1$	$a_0$	$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$a_3$	$a_2$	$a_1$	$a_0$	$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$
0	0	0	0								1	0	0	0							
0	0	0	1								1	0	0	1							
0	0	1	0								1	0	1	0							
0	0	1	1								1	0	1	1							
0	1	0	0								1	1	0	0							
0	1	0	1								1	1	0	1							
0	1	1	0								1	1	1	0							
0	1	1	1								1	1	1	1							

**Question 9** Etablir l'équation logique de chaque LED  $F_i = f_i(a_3, a_2, a_1, a_0)$  de l'afficheur 7 segments à l'aide d'un tableau de Karnaugh.

$F_0$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$				
$a_3a_2 = 01$				
$a_3a_2 = 11$				
$a_3a_2 = 10$				

$F_1$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$				
$a_3a_2 = 01$				
$a_3a_2 = 11$				
$a_3a_2 = 10$				

$F_2$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$				
$a_3a_2 = 01$				
$a_3a_2 = 11$				
$a_3a_2 = 10$				

$F_3$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$				
$a_3a_2 = 01$				
$a_3a_2 = 11$				
$a_3a_2 = 10$				

$F_4$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$				
$a_3a_2 = 01$				
$a_3a_2 = 11$				
$a_3a_2 = 10$				

$F_5$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$				
$a_3a_2 = 01$				
$a_3a_2 = 11$				
$a_3a_2 = 10$				

$F_6$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$				
$a_3a_2 = 01$				
$a_3a_2 = 11$				
$a_3a_2 = 10$				

**Question 10** Coder à la suite du code python la fonction  $f$  qui donne la liste  $F = [F_0, F_1, F_2, F_3, F_4, F_5, F_6]$  à partir de la liste  $a = [a_0, a_1, a_2, a_3]$ .

## 5 Correction

**Question 1** Etablir la table de vérité de la somme  $s_0$  et la retenue  $r_1$  en fonction des entrées  $b_0$  et  $a_0$ .

$b_0$	$a_0$	$s_0$	$r_1$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

**Question 2** Etablir les équations logiques de la somme  $s_0$  (en utilisant un opérateur logique remarquable) et la retenue  $r_1$  en fonction des entrées  $b_0$  et  $a_0$ . Vérifier le résultat à l'aide des logiciels de simulation.  
 $s_0 = a_0 \oplus b_0$ ,  $r_1 = a_0 \cdot b_0$

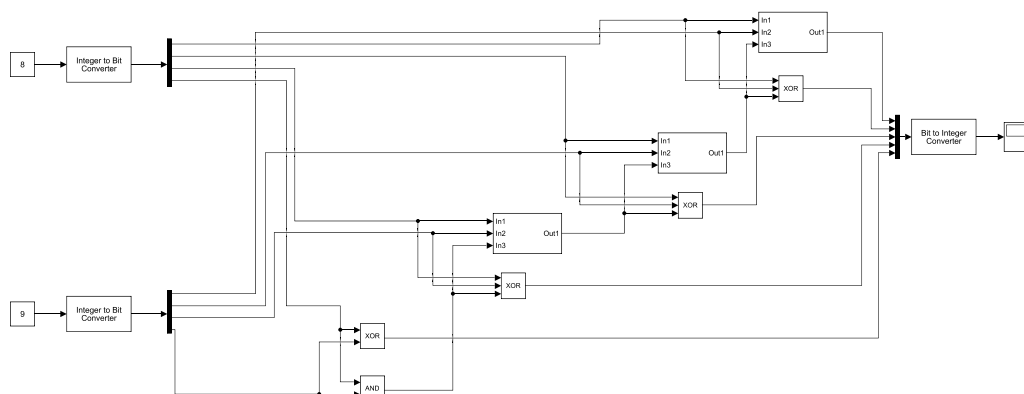
**Question 3** Etablir la table de vérité de la somme  $s_1$  (en utilisant un opérateur logique remarquable) et la retenue  $r_2$  en fonction des entrées  $b_1$ ,  $a_1$  et  $r_1$ .

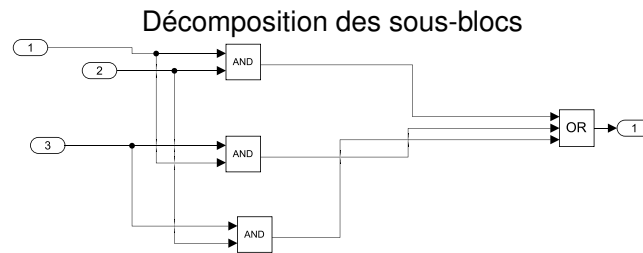
$b_1$	$a_1$	$r_1$	$s_1$	$r_2$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

**Question 4** Établir les équations logiques de la somme  $s_1$  et la retenue  $r_2$  en fonction des entrées  $b_1$ ,  $a_1$  et  $r_1$ . Vérifier le résultat à l'aide des logiciels de simulation.  
 $s_1 = a_1 \oplus b_1 \oplus r_1$ ,  $r_2 = a_1 \cdot b_1 + b_1 \cdot r_1 + r_1 \cdot a_1$

**Question 5** En déduire les équations logiques de la somme  $s_i$  au rang  $i$  (en utilisant un opérateur logique remarquable) et la retenue  $r_{i+1}$  au rang  $i + 1$  en fonction des entrées  $a_i$ ,  $b_i$  et  $r_i$  au rang  $i$ .  
 $s_i = a_i \oplus b_i \oplus r_i$ ,  $r_{i+1} = a_i \cdot b_i + b_i \cdot r_i + r_i \cdot a_i$

**Question 6** Tracer le schéma de l'ensemble du système de traitement de l'information.





**Question 7** Compléter le fichier python afin d'obtenir le même fonctionnement en logique programmée.  
Le code est disponible [ici](#).

**Question 8** Remplir les colonnes  $F_i$  de la table de vérité.

$a_3$	$a_2$	$a_1$	$a_0$	$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$a_3$	$a_2$	$a_1$	$a_0$	$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$
0	0	0	0	1	1	1	1	1	1	0	1	0	0	0	1	1	1	1	1	1	1
0	0	0	1	0	1	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1
0	0	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1	1	1
0	0	1	1	1	1	1	1	0	0	1	1	0	1	1	0	0	1	1	1	1	1
0	1	0	0	0	1	1	0	0	1	1	1	1	0	0	1	0	0	1	1	1	0
0	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	1	1	0	1
0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1

**Question 9** Etablir l'équation logique de chaque LED  $F_i = f_i(a_3, a_2, a_1, a_0)$  de l'afficheur 7 segments à l'aide d'un tableau de Karnaugh.

$F_0$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$	1	0	1	1
$a_3a_2 = 01$	0	1	1	1
$a_3a_2 = 11$	1	0	1	1
$a_3a_2 = 10$	1	1	0	1

$F_1$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$	1	1	1	1
$a_3a_2 = 01$	1	0	1	0
$a_3a_2 = 11$	1	0	1	1
$a_3a_2 = 10$	1	1	0	1

$F_2$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$	1	1	1	0
$a_3a_2 = 01$	1	1	1	1
$a_3a_2 = 11$	0	1	0	0
$a_3a_2 = 10$	1	1	1	1

$F_3$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$	1	0	1	1
$a_3a_2 = 01$	0	1	0	1
$a_3a_2 = 11$	1	1	0	1
$a_3a_2 = 10$	1	1	1	0

$F_4$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$	1	0	0	1
$a_3a_2 = 01$	0	0	0	1
$a_3a_2 = 11$	1	1	1	1
$a_3a_2 = 10$	1	0	1	1

$F_5$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$	1	0	0	0
$a_3a_2 = 01$	1	1	0	1
$a_3a_2 = 11$	1	0	1	1
$a_3a_2 = 10$	1	1	1	1

$F_6$	$a_1a_0 = 00$	$a_1a_0 = 01$	$a_1a_0 = 11$	$a_1a_0 = 10$
$a_3a_2 = 00$	0	0	1	1
$a_3a_2 = 01$	1	1	0	1
$a_3a_2 = 11$	0	1	1	1
$a_3a_2 = 10$	1	1	1	1

$$F_0 = \overline{a_3}.a_1 + a_2.a_1 + \overline{a_2}.\overline{a_0} + a_3.\overline{a_0} + a_3.\overline{a_2}.\overline{a_1} + \overline{a_3}.a_2.a_0$$

$$F_1 = \overline{a_1}.\overline{a_0} + \overline{a_3}.\overline{a_2} + a_2.a_1.a_0 + a_3.\overline{a_0} + \overline{a_2}.\overline{a_1}$$

$$F_2 = \overline{a_3}.\overline{a_1} + \overline{a_3}.a_0 + \overline{a_1}.a_0 + a_3.\overline{a_2} + \overline{a_3}.a_2$$

$$F_3 = a_3.\overline{a_1} + \overline{a_3}.\overline{a_2}.\overline{a_0} + \overline{a_3}.\overline{a_2}.a_1 + a_2.\overline{a_1}.a_0 + a_2.a_1.\overline{a_0} + a_3.\overline{a_2}.a_0$$

$$F_4 = a_3.a_2 + a_1.\overline{a_0} + a_3.a_1 + \overline{a_2}.\overline{a_0}$$

$$F_5 = \overline{a_1}.\overline{a_0} + a_3.a_1 + a_2.\overline{a_0} + \overline{a_3}.a_2.\overline{a_1} + a_3.\overline{a_2}$$

$$F_6 = a_1.\overline{a_0} + a_3.\overline{a_2} + a_3.a_0 + \overline{a_2}.a_1 + \overline{a_3}.a_2.\overline{a_1}$$

**Question 10** Coder à la suite du code python la fonction `f` qui donne la liste  $F = [F_0, F_1, F_2, F_3, F_4, F_5, F_6]$  à partir de la liste  $a = [a_0, a_1, a_2, a_3]$ .

Le code est disponible [ici](#).

```
def f(a):
    fr=[]
    fr.append(not a[3] and a[1] or a[2] and a[1] or not a[2] and not a[0]
              or a[3] and not a[0] or a[3]and not a[2] and not a[1]
              or not a[3]and a[2] and a[0])
    fr.append(not a[1]and not a[0] or not a[3]and not a[2]or a[2]and a[1]and a[0]
              or a[3] and not a[0]or not a[2]and not a[1])
    fr.append(not a[3]and not a[1]or not a[3]and a[0] or not a[1] and a[0]
              or a[3] and not a[2] or not a[3]and a[2])
    fr.append(a[3]and not a[1] or not a[3]and not a[2] and not a[0]
              or not a[3]and not a[2]and a[1]or a[2]and not a[1]and a[0]
              or a[2]and a[1]and not a[0]or a[3]and not a[2]and a[0])
    fr.append(a[3]and a[2]or a[1]and not a[0] or a[3]and a[1]
              or not a[2]and not a[0])
    fr.append(not a[1]and not a[0]or a[3]and a[1]or a[2]and not a[0]
              or not a[3]and a[2]and not a[1]or a[3]and not a[2])
    fr.append(a[1]and not a[0]or a[3]and not a[2]or a[3]and a[0]
              or not a[2]and a[1]or not a[3]and a[2]and not a[1])
    return fr
```