

# Compte rendu TP2 Hard ALM

Dorian Mounier

Eloi Charra

30/11/2021

## 1 Etude de l'UAL

### 1.1 Etudes des fonctions de l'UAL

Une unité arithmétique et logique est un outil permettant de réaliser plusieurs opérations booléennes. Elle réalise des additions, des soustractions, des incrémentations et des opposés.

Nous avons vu au TP précédent comment réaliser un additionneur et un soustracteur, nous allons passer rapidement dessus.

#### 1.1.1 L'additionneur

Voici la table de vérité de l'additionneur :

$a_i$	$b_i$	$c_i$	$z_i$	$c_{i+1}$
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

Nous pouvons donc en conclure les équations de  $z_i$  et  $c_{i+1}$  :

$$z_i = a_i \oplus b_i \oplus c_i$$

$$c_{i+1} = a_i.b_i + b_i.c_i + a_i.c_i$$

#### 1.1.2 Le soustracteur

Voici la table de vérité du soustracteur :

$x_i$	$y_i$	$t_i$	$t_{i+1}$	$d_i$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Nous avons vu au TP précédent que :

$$t_{i+1} = \text{NAND}(\text{NAND}(\overline{x_i}, t_i), \text{NAND}(\overline{x_i}, y_i), \text{NAND}(y_i, t_i))$$

$$d_i = x_i \oplus y_i \oplus t_i$$

Nous pouvons utiliser un additionneur pour réaliser une soustraction. En effet, il suffit d'inverser la seconde opérande ainsi que la retenue sortante tout en initialisant la retenue entrante à 1.

$x_i$	$y_i$	$t_i$	$d_i$	$z_i$	$\overline{t_i}$	$d_i$	$z_i$
0	0	0	0	0	1	1	1
0	1	0	1	1	1	0	0
1	0	0	1	1	1	0	0
1	1	0	0	0	1	1	1

On remarque que lorsque la retenue entrante est à 1, nous obtenons des valeurs  $d_i$  et  $z_i$  inverses par rapport aux résultats  $d_i$  et  $z_i$  lorsque la retenue entrante est à 0. Si maintenant nous inversons  $y_i$  :

$x_i$	$y_i$	$t_i$	$d_i$	$z_i$	$\overline{t_i}$	$d_i$	$z_i$
0	1	0	1	1	1	0	0
0	0	0	0	0	1	1	1
1	1	0	0	0	1	1	1
1	0	0	1	1	1	0	0

Nous pouvons voir qu'après avoir inversé  $y_i$  et en ayant la retenue entrante à 1, nous obtenons les mêmes valeurs que lorsque  $y_i$  n'est pas inversé et que la retenue entrante est à 0.

### 1.1.3 Implémentation des autres fonctions

L'opération à effectuer est décidée à l'aide de 2 bits `uc1` et `uc0`. La combinaison de ces 2 bits nous donnera quelle opération effectuer. Nous pouvons résumer les combinaisons dans un tableau :

<code>uc0</code>	<code>uc1</code>	opération
0	0	$T = A - B$
0	1	$T = -A$
1	0	$T = A + B$
1	1	$T = B + 1$

Premièrement il nous faut calculer la retenue entrante, elle est différente pour chaque opération. Elle est à 0 lorsque nous effectuons l'addition mais est à 1 pour toutes les autres opérations.

- En effet, pour la soustraction, comme nous l'avons évoqué précédemment, la retenue entrante doit être à 1, ainsi que la seconde opérande inversée.
- Pour calculer l'opposé, nous avons décidé d'inverser A et de mettre la retenue entrante à 1 pour calculer l'opposé du complément à 2, nous n'avons pas besoin de B ici.
- Pour l'incrément, comme son nom l'indique, il faut ajouter 1 à B, la retenue entrante représente le 1 en plus, nous n'avons pas besoin de A ici.
- Pour l'addition, nous n'avons pas besoin d'avoir la retenue entrante à 1.

Nous pouvons résumer cela dans un tableau :

	X	Y	v0
Soustraction	A	$\bar{B}$	1
Opposé	$\bar{A}$	0	1
Addition	A	B	0
Incrémentation	0	B	1

### 1.1.3 Détermination de X et Y

Pour déterminer quelle valeur doivent prendre X et Y parmi, nous avons utiliser un multiplexeur 4 vers 1 pour chaque opérande X et Y:

uc0	uc1	A	$\bar{A}$	A	0	X
0	0	0	*	*	*	0
0	0	1	*	*	*	1
0	1	*	0	*	*	0
0	1	*	1	*	*	1
1	0	*	*	0	*	0
1	0	*	*	1	*	1
1	1	*	*	*	0	0
1	1	*	*	*	1	1

uc0	uc1	$\bar{B}$	0	B	B	Y
0	0	0	*	*	*	0
0	0	1	*	*	*	1
0	1	*	0	*	*	0
0	1	*	1	*	*	1
1	0	*	*	0	*	0
1	0	*	*	1	*	1
1	1	*	*	*	0	0
1	1	*	*	*	1	1

Nous pouvons remarquer que  $X = Y$ , cela veut dire qu'il suffit de placer les entrée dans le bon ordre pour obtenir le résultat que nous voulons et nous pouvons écrire les équations de X et Y:

$$X = \overline{uc0}.\overline{uc1}.A + \overline{uc0}.uc1.\bar{A} + uc0.\overline{uc1}.A + uc0.uc1.0$$

$$Y = \overline{uc0}.\overline{uc1}.\bar{B} + \overline{uc0}.uc1.0 + uc0.\overline{uc1}.B + uc0.uc1.B$$