

# TD 4

## Opérateurs Arithmétiques et Logiques

*Préparation*

### Ex. 1 : Codage des nombres en complément à deux

Le codage en complément à deux est un codage permettant de représenter des entiers relatifs en binaire. Il existe d'autres codages moins utilisés (*e.g.* signe et valeur absolue, complément à un) que l'on ne détaillera pas ici.

**Question 1** Remplir un tableau contenant les valeurs en base 2 et en base 16 des entiers relatifs entre -8 et 7 (inclus), codés en complément à  $2^4$ . Comment peut-on calculer  $-X$  à partir de  $X$  en base 2 ?

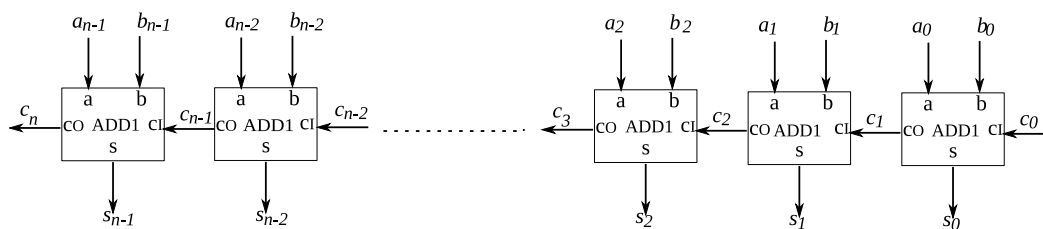
**Question 2** Quel intervalle d'entiers relatifs peut-on coder sur  $n$  bits ?

**Question 3** Effectuer les opérations  $2 + 3$ ,  $-3 + 3$ ,  $-2 + -5$ ,  $7 + 1$ ,  $-7 + -2$  sur 4 bits.

**Question 4** Comment peut-on détecter un dépassement de capacité lors d'une opération sur des nombres codés en complément à deux ?

### Ex. 2 : Additionneur binaire

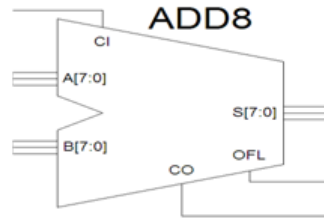
On utilise l'architecture dite de « l'additionneur à propagation de retenue ». Le principe est de construire une cellule élémentaire d'addition 1 bit ADD1 capable d'effectuer une addition entre deux bits  $a$  et  $b$ , en prenant en compte une retenue entrante  $cl$  (carry in), et qui produit en sortie la somme  $s$  et une retenue sortante  $co$  (carry out). Une fois cette cellule disponible, il est facile de construire un additionneur  $n$  bits en reliant la retenue sortante de la cellule de rang  $i$  à la retenue entrante de la cellule de rang  $i + 1$ .



**Question 1** Donner les expressions simplifiées de la somme  $s$  et de la retenue sortante  $co$  d'une cellule élémentaire ADD1 en fonction des opérandes  $a$  et  $b$  et de la retenue entrante  $cl$ . Dessiner le schéma correspondant. Que doit valoir la retenue entrante  $c_0$  de l'additionneur  $n$  bits ? Évaluer le temps de calcul pour une addition  $n$  bits.

**Question 2** Un additionneur dispose également de deux sorties particulières, sur 1 bit chacune (appelées « indicateurs » (*flags*)) et définies par :

- *CO* (*Carry Out*) est la retenue sortante  $c_n$  générée par l'addition, qui vaut 1 ssi une addition portant sur des entiers naturels  $a$  a généré un dépassement de capacité ;
- *OFL* (*Overflow*) vaut 1 ssi une addition portant sur des entiers codés en complément à deux a généré un dépassement de capacité.



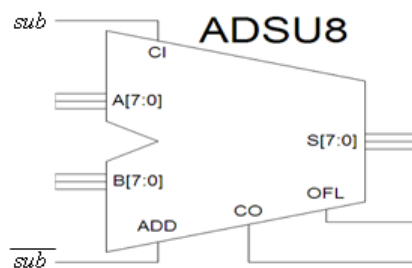
En s'aidant de l'exercice précédent, exprimer le bit *OFL* en fonction des  $c_i$ .

### Ex. 3 : Additionneur/soustracteur

On veut maintenant implanter l'opération de soustraction en utilisant l'additionneur de l'exercice 2.

**Question 1** En se basant sur la relation  $-X = \bar{X} \text{ plus } 1$ , expliquer comment on peut simplement modifier le circuit d'une cellule élémentaire d'additionneur pour gérer aussi la soustraction de deux entiers relatifs codés en complément à deux, suivant la valeur d'une entrée *sub* qui vaut 0 pour une addition, 1 pour une soustraction. Dessiner le schéma correspondant à une cellule de rang *i*. Que faut-il faire pour la cellule de rang 0 ?

**Question 2** La sortie *CO* de l'additionneur/soustracteur est égale à la retenue  $c_n$ . Dans le cas d'une soustraction d'entiers naturels, que veut dire la valeur de *CO* ? Montrer que la variable  $C = CO \oplus \text{sub}$  vaut 1 ssi une addition ou une soustraction portant sur des entiers naturels génère un résultat qui n'est pas un entier naturel codable sur *n* bits. On peut aussi dire que *CO* est actif à 1 pour l'addition, à 0 pour la soustraction des entiers naturels.



Note : l'indicateur *OFL* est toujours valide pour les entiers codés en complément en 2, que l'opération soit une addition ou une soustraction.

*Pour aller plus loin...*

**Question 3** On cherche à effectuer une opération sur  $2n$  bits par composition d'opérations sur  $n$  bits, en utilisant 2 additionneurs/soustracteurs  $n$  bits. Comment connecter les deux circuits ?

*Pour aller plus loin...*

### Ex. 4 : Unité Arithmétique et Logique

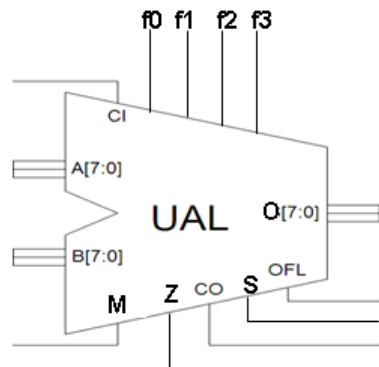
Dans cet exercice, on va étudier l'unité arithmétique et logique, et construire la partie centrale de ce circuit (l'étude complète d'un tel circuit dépasse le cadre du TD).

Une Unité Arithmétique et Logique (UAL ou ALU pour Arithmetic and Logic Unit) est un circuit combinatoire regroupant différentes opérations arithmétiques (addition, soustraction, etc.) et logiques (AND, OR, XOR, etc.) applicables à des valeurs codées sur un nombre  $n$  de bits (on parle alors d'« UAL  $n$  bits »).

Sur le schéma ci-dessous, le bit  $M$  permet de choisir le mode ( $M=0$  : mode logique,  $M = 1$  : mode arithmétique), les bits  $f_i$  représentent les entrées de fonction, qui permet de choisir l'opération que doit effectuer l'UAL. Les entrées  $A$  et  $B$  servent à coder les opérandes et  $O$  le résultat de l'opération (sur 8 bits dans l'exemple du schéma).

Outre  $CO$  et  $OFL$ , l'UAL dispose également de deux indicateurs, définis par :

- $Z$  (*Zero*) vaut 1 ssi le résultat d'une opération est nul ;
- $S$  (*Sign*) vaut 1 ssi le résultat d'une opération, interprété comme un entier relatif codé en complément à deux, est négatif.



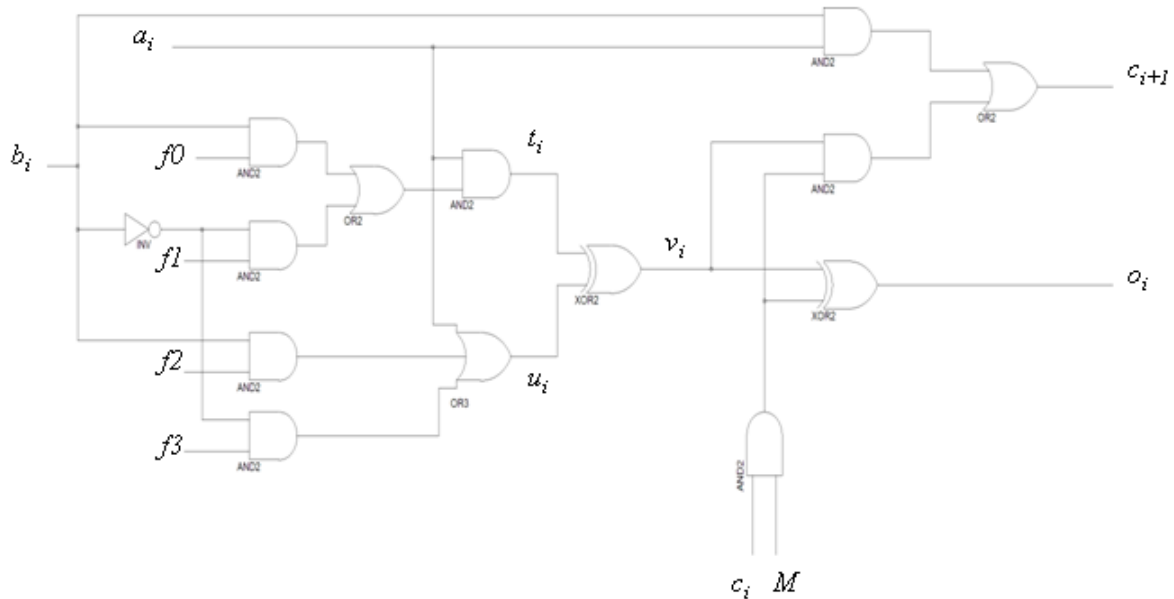
Les opérations que doit réaliser l'UAL que nous concevons sont indiquées dans le tableau ci-dessous.

Opération	Mode $M$	$f_0$ $f_1$ $f_2$ $f_3$	Résultat
const0	0		00000000
const-1	0		11111111
add	1		A plus B plus CI
sub	1		A moins B moins 1 plus CI
notA	0		NOT(A)
notB	0		NOT(B)
xor	0		A XOR B
or	0		A OR B
and	0		A AND B
nopA	0		A
nopB	0		B

**Question 1** Comment peut-on très simplement créer la génération des indicateurs  $Z$  et  $S$  par l'UAL ?

**Question 2** On essaie de créer l'UAL avec un minimum de modifications de la cellule additionneur/soustracteur déjà conçue. Comment doit agir la commande de mode  $M$  sur les retenues  $c_i$  ?

On propose la cellule suivante (schéma pour la cellule de rang i) :



La sortie  $s_i$  de la cellule d'additionneur/soustracteur de l'exercice précédent est égale à  $a_i \oplus (b_i \oplus sub) \oplus c_i$ . Si la propagation de la retenue est inhibée,  $s_i = a_i \oplus (b_i \oplus sub)$ . Pour concevoir l'UAL, on pose  $o_i = v_i \oplus (c_i.M)$ , avec  $v_i = a_i \oplus (b_i \oplus sub)$  pour l'addition/soustraction.

On remarque que :  $x \oplus y = x.y \oplus (x + y) = x.\bar{y} + \bar{x}.y$ . On a donc :

$$v_i = a_i(b_i \oplus sub) \oplus (a_i + b_i \oplus sub) = a_i(b_i.\bar{sub} + \bar{b_i}.sub) \oplus (a_i + b_i.\bar{sub} + \bar{b_i}.sub).$$

Pour avoir plus de choix sur les termes pris en compte, on remplace les occurrences de  $sub$  et  $\bar{sub}$  par  $f0, f1, f2, f3$ . On obtient donc :  $v_i = a_i(b_i.f0 + \bar{b_i}.f1) \oplus (a_i + b_i.f2 + \bar{b_i}.f3)$

Pour simplifier le travail, on notera  $t_i$  le premier terme de l'expression de  $v_i$  et  $u_i$  le deuxième terme. On a donc :  $t_i = a_i(b_i.f0 + \bar{b_i}.f1)$ ,  $u_i = a_i + b_i.f2 + \bar{b_i}.f3$  et  $v_i = t_i \oplus u_i$ .

**Question 3** Quelles valeurs doivent prendre M et les variables de sélection  $f_j$  pour l'addition ? la soustraction ?

**Question 4** Déterminer les valeurs de  $t_i$  et  $u_i$  suivant les valeurs des  $f_j$ , puis ce que vaut  $v_i$ . En déduire les valeurs des  $f_j$  nécessaires pour les opérations logiques indiquées dans le tableau de fonctionnement de l'UAL.

**Question 5** Rechercher l'ensemble des opérations arithmétiques que peut faire cette UAL.