Les opérations binaires

Compétences associées
A2 : Analyser et interpréter une information numérique

Objectifs

Etre capable:

- De coder les nombres entiers en code complément à 2.
- De résoudre les opérations en binaire.

GENERALITES

Pour l'instant les codes nous permettaient de coder un nombre naturel en décimal, en binaire, en hexadécimal et de coder un caractère en code ASCII. Afin de comprendre les opérations arithmétiques calculées par un microprocesseur il est nécessaire de maîtriser le codage des nombres entiers négatifs. Ce code est appelé le code complément à deux. Le complément à deux est la représentation la plus commode utilisée par les microprocesseurs.

Rappels

Pour un nombre binaire de n bits, 2n combinaisons différentes sont possibles et on compte de 0 à 2^{n-1}.

Exemples

- avec 4 bits : il y a 16 combinaisons différentes de 0 à 15
- avec 8 bits : il y a 256 combinaisons de 0 à 255
- avec 10 bits : il y a 1024 combinaisons de 0 à 1023
- avec 16 bits : il y a 65536 combinaisons de 0 à 65535
- avec 20 bits : il y a 1 048 576 combinaisons
- avec 32 bits : il y a 4 294 967 296 combinaisons

ADDITION DE 2 ENTIERS POSITIFS

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 retenue ou carry (1)

Méthode :

L'addition des nombres binaires s'effectue de la même façon qu'en décimal.

Exemples : en décimal

$$\begin{array}{r} \mathbf{1} \\ 19 \\ +\ \underline{\ \ 2} \\ 21 \end{array} \qquad \begin{array}{r} 33 \\ +\ \underline{33} \\ 66 \end{array}$$

en binaire

$$\begin{array}{r} \mathbf{1}\ \ \ \ \\ 0001\ 0011 \\ +\ \underline{0000\ 0010} \\ 0001\ 0101 \end{array} \qquad \begin{array}{r} \mathbf{1}\ \ \ \ \ \ \ \ \ \mathbf{1}\ \ \ \ \\ 0010\ 0001 \\ +\ \underline{0010\ 0001} \\ 0100\ 0010 \end{array}$$

Note :

On ne représente que des nombres positifs.
Ajouter un nombre à son égal revient à le multiplier par 2.

MULTIPLICATION DE DEUX ENTIERS POSITIFS

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1

Méthode :

Le principe de la multiplication en base 2 est identique qu'en base 10. Si le bit du multiplicateur est à "1", on recopie le multiplicande. S'il est à "0", on décale la ligne obtenue d'un cran vers la gauche et on additionne les résultats.

Exemple : en décimal

```
   45
X   5
  225
```

en binaire

```
   10 1101
x      101
   10 1101
+ 1011 01
 1110 0001
```

LA CONVENTION DU SIGNE

En binaire, on ne connaît pas les signes "+" et "-". Par convention, en codage binaire signé, le bit de poids fort indique le signe du nombre.

- ***Si le bit de poids fort est à "0" le nombre est positif.***
- ***Si le bit de poids fort est à "1" le nombre est négatif.***

Nombres de combinaisons, de valeurs négatives et positives

- avec 4 bits : il y a 16 combinaisons différentes de -8 à +7

- avec 8 bits : il y a ____ combinaisons de ____________

- avec 16 bits : il y a ______ combinaisons de ____________

LE COMPLEMENT A UN

Pour obtenir le complément à un d'un nombre binaire, il faut complémenter chaque bit c'est-à-dire transformer les "0" en "1" et les "1" en "0".

Exemple 1 :

+4 décimal 0000 0100 en binaire

/4 décimal 1111 1011 en complément à un

Exemple 2 :

+100 décimal 0110 0100 en binaire

/100 décimal 1001 1011 en complément à un

LE COMPLEMENT A DEUX

Dans la représentation en complément à deux, de la même manière qu'en complément à un, les nombres positifs se représentent en binaire signé habituel. La différence réside dans la représentation des nombres négatifs.

Il existe deux méthodes pour obtenir le complément à deux :

- ***Le complément à 2 d'un nombre revient à calculer le complément à un du nombre puis de lui additionner la valeur "1".***

  Exemple : C2 de 0001 0101 ? ⇨ C1 = 1110 1010 ⇨ C2 = 1110 1010 + 0000 0001 = 1110 1011

- ***Le complément à 2 d'un nombre consiste à conserver tous les bits à partir de la droite jusqu'au premier "1" inclus et de complémenter (inverser) tous les bits suivants.***

  Exemple : C2 de 1011 0100 ? ⇨ C2 = 0100 1100 (les 3 bits de poids faibles sont inchangés)

ADDITION ET SOUSTRACTION DE 2 ENTIERS SIGNES

Pour vérifier en décimal une opération avec des nombres binaires signés, il faut tout d'abord chercher l'équivalent du nombre décimal négatif en le complémentant à deux".

<u>Exemple 1 :</u> (-4) + 6 = ?

+4 décimal ⇨ 0000 0100 en binaire
/4 décimal ⇨ 1111 1011 en complément à un
-4 décimal ⇨ 1111 1100 en complément à deux

-4 décimal ⇨ 1111 1100 en complément à deux
+6 décimal ⇨ + <u>0000 0110</u> en binaire
~~1~~0000 0010

<u>Exemple 2 : 6 + (-4) = ?</u>

-4 décimal ⇨ 1111 1100 en complément à deux

+6 décimal ⇨ 0000 0110 en binaire
-4 décimal ⇨ + <u>1111 1100</u> en complément à deux
~~1~~0000 0010 en binaire = 2 en décimal

La soustraction s'effectue en "ajoutant le complément à deux" du nombre binaire. Si A et B sont deux entiers positifs, l'opération A - B revient à ajouter à A le complément à deux du nombre B.

<u>Exemple 1 : 16 - 6 = ?</u>

+16 décimal ⇨ 0001 0000 en binaire
+6 décimal ⇨ 0000 0110 en binaire
-6 décimal ⇨ 1111 1010 en complément à deux

+16 décimal ⇨ 0001 0000 en binaire
-6 décimal ⇨ + <u>1111 1010</u> en complément à deux
~~1~~0000 1010 en binaire = 10 en décimal

Exemple 2 : _79 - 63 = ?_

	0100 1111 (79)	0100 1111	
-	<u>0011 1111</u> (63)	+ <u>1100 0001</u>	(C2 de 63)
	Résultat :	~~1~~ 0001 0000	(16)

La retenue (9e bit) est à éliminer, le résultat peut directement être converti en décimal.

Exemple 3 : _27 - 61 = ?_

	0001 1011 (27)	0001 1011	
-	<u>0011 1101</u> (61)	+ <u>1100 0011</u>	
	Résultat :	1101 1110	(- 34)

Il n'y a pas de retenue et le MSB = "1" :

Le résultat est négatif et codé en complément à 2. Pour avoir l'équivalent décimal, il faut le complémenter à nouveau : 0010 0010 = (-) 34

Remarque :

Il est possible d'additionner ou de soustraire des nombres signés sans se préoccuper de leur signe.

LES INDICATEURS DE RETENUE ET DE DEBORDEMENT

Le bit de report ou retenue C Carry

C'est un bit supplémentaire. Lorsque les 2 nombres sont codés sur 8 bits, le 9e bit du résultat indique la condition éventuelle de retenue (ou report). Par la suite, nous l'appellerons "C" ("Carry" = retenue en anglais). La retenue est le bit de rang 16 ou le 17e bit du résultat lorsque les nombres sont codés en mots.

Le débordement V Overflow

C'est un bit supplémentaire interne qui indique un dépassement de capacité de calcul

⇨ le résultat obtenu est faux.

Exemple : _64 + 65 = ?_

	1	
+ 64 décimal ⇨	0100 0000	
+ 65 décimal ⇨	<u>0100 0001</u>	
	1000 0001 ⇨	-127 en décimal,
		Le résultat devrait être _____

Une retenue interne "Débordement" a été produite du bit 6 vers le bit 7.
Le résultat est devenu négatif par accident. Cette situation doit être détectée pour être corrigée.

Le débordement se produit dans les cas d'additions ou de soustractions de grands nombres positifs ou négatifs.

Exemples d'addition entre deux entiers

- ***- positif - positif avec débordement***

```
     0111 1111 (+127)
   + 0000 0001 (+1)
     1000 0000 (-128)   V = 1  C = 0
```

⇨ le résultat est FAUX car il y a eu débordement

- ***- négatif - négatif avec débordement***

```
     1000 0001 (-127)
   + 1100 0010 (-62)
    10100 0011 (+67)    V = 1 C = 1
```

⇨ Le résultat est FAUX : (-189) est trop grand pour être codé sur 8 bits.

```
     1111 1111 (-1)
   + 1111 1111 (-1)
    11111 1110 (-2)     V = 0 C = 1
```

⇨ C est ignorée. Le résultat est juste car la retenue du bit 6 vers le bit 7 n'a pas modifié le bit de signe.

10. EXERCICES

1) Donner le complément à 1 de - 1, - 15, - 127, - 1024, - 32767
2) Donner le complément à 2 de 0, 22, + 127, - 1, - 15, - 127, - 128, - 32767, - 32768
3) Ces opérations s'effectuent avec des nombres signés.

   Calculer ces opérations en base binaire et exprimer les deux opérandes et le résultat en décimal.

```
    0 0 0 1  0 0 0 0        1 1 1 0  0 1 1 0        1 1 1 1  1 1 1 0
  + 1 1 1 1  1 1 0 0      + 0 0 0 0  1 1 0 0      + 1 1 1 1  1 1 0 0

    0 0 0 1  0 0 0 0        1 1 1 0  0 1 1 0        0 0 0 0  0 1 1 1
  - 1 1 1 1  1 1 0 0      - 0 0 0 0  1 1 0 0      - 1 1 1 1  1 1 0 0


    0 0 0 0  1 0 0 1
  x 0 0 0 0  1 0 1 1
```