|  |  |  |
| --- | --- | --- |
| NSI – 1ere | COURS Séquence 2-AB : Entiers relatifs en binaire | LFV |

Dans le cours précédent nous avons vu comment représenter des nombres entiers *positifs* dans différentes bases.  
Nuos allons ici nous intéresser au cas des entiers *relatifs* mais uniquement en base (binaire).

1. **Représentation des entiers relatifs en complément à (puissance**Dans cette première partie nous ne travaillerons qu'avec des octets.
2. **Un peu de vocabulaire**

Sur un octet (c’est-à-dire un ensemble de bits), on a vu que les bits à droite correspondent aux petites puissances de alors que les bits à gauche correspondent aux grandes puissances de . On dit donc que :  
 *- le bit le plus à droite est le bit de poids faible,  
 - le bit le plus à gauche est le bit de poids fort.*

1. **Méthode naïve pour représenter les entiers relatifs**La méthode naïve est de rtéserver le bit de poids fort pour indiquer le signe. Bit de poids fort à zéro : nombre positif. Bit de poids fort à un : nombre négatif. Ainsi :  
   **0**000 1001 correspond à +9  
   **1**001 1001 correspond à -9

Cette méthode a un inconvénient majeur : si on utilisait cette représentation des entiers négatifs, l'addition de deux entiers poitif et négatif ne pourrait pas être effectuée comme celle de deux entiers positif et positif :

|  |  |
| --- | --- |
| 0000 1000  +0000 0110  -----------  0000 1110 | 0000 1000 +1000 0110 -----------  1000 1110 |
| On a : 8 + 6 = 14 | On a malheureusement : 8 + (-6) = -14 !!! |

Cette méthode naïve n'est donc pas satisfaisante car l'addition binaire est une opération effectuée très très très très souvent au cœur des processeurs. Il est donc primordial – pour des raisons de performances – que l'addition binaire soit effectuée le plus efficacement possible.

1. **Représentation en complément à (puissance ) pour les entiers relatifs***a) Interprétation 1 de cette représentation*  
   La méthode du complément à (qui signifie en réalité complément à puissance ) sur bits consiste à représenter un entier négatif par la représentation de l'entier positif sur bits.  
    *Exemple : soit à représenter sur bits*On a :   
   Donc se représente sous la forme 1000 1110 en complément à deux.

*b) Interprétation 2 de cette représentation*  
La méthode du complément à pour représenter un entier négatif sur bits peut aussi se voir comme la succession suivante d'opérations :  
- Représenter le nombre positif associé en binaire (c’est-à-dire ).  
- Inverser tous les bits de cette représentation.  
- Ajouter à la nouvelle représentation obtenue.

*Exemple : soit à représenter sur bits*Inversion : 1000 1101  
Ajout de : 1000 1110 *c) Interprétation 3 de cette représentation*La méthode du complément à pour représenter un entier négatif sur bits peut aussi se voir comme la succession suivante d'opérations :  
- Représenter le nombre positif associé en binaire (c’est-à-dire ).  
- En partant du bit de poids faible (de la droite), inverser tous les bits situés *strictement après* le premier 1  
  
*Exemple : soit à représenter sur bits*On identifie ce qui est situé *strictement après* le premier 1 en partant de la droite :   
On inverse cela : 1000 1110

1. **Propriétés du complément à (puissance )**La première propriété de la représentation des entiers en complément à est que *le bit de poids fort* (le plus à gauche) indique le signe de l'entier (on parle *de bit de signe*). Lorsque ce bit de signe est égal à l'entier représenté est négatif et, a contrario, lorsque ce bit de signe est égal à l'entier représenté est positif.

La seconde propriété est que la représentation en complément à permet à l'addition binaire de fonctionner de façon similaire pour deux entiers positifs et pour un entier positif et un entier négatif :

|  |  |
| --- | --- |
| 0000 1000  +0000 0110  -----------  0000 1110 | 1111  0000 1000  +1111 1010  -----------  0000 0010 |
| On a : 8 + 6 = 14 | On a bien : 8 + (-6) = -2 |

1. **Passer de la représentation à l'entier en base 10**- Si le bit de signe est zéro, il suffit de le considérer comme un entier positif :

0101 0011 représente l'entier positif   
- Si le bit de signe est un**,** on a deux alternatives :  
 Alternative 1 : on le considère comme un entier positif PUIS on soustrait :  
 1101 0101 représenterait l'entier positif   
 ce qui correspond à l'entier relatif   
 Alternative 2 : on inverse les bits situés strictement à gauche du premier en partant de la droite :  
 1101 0101 conduit à 0010 1011  
 ce qui correspond à soit .

1. **Nombre de bits utilisés**
2. **Le cas des octets**Votre connaissance parfaite des cours précédents vous permet de savoir immédiatement que sur un octet ( bits) on peut coder valeurs (si on code les entiers positifs on va de à ).  
   Si on représente les entiers relatifs en complément à deux , on peut représenter :  
   - les entiers positifs de [0000 0000] à [0111 1111]  
   - les entiers négatifs de [1000 0000] à [1111 1111]

|  |  |  |
| --- | --- | --- |
| Nombre de bits | Plage des entiers positifs | Plage des entiers relatifs en complément à deux |
|  | à | à |
|  | à | à |
|  | à | à |
|  | à | à |
|  | à | à |

Ainsi sur bits, en complément à on peut représenter les entiers de à . Plus généralement :

1. **Impact de l'addition et de la multiplication sur le nombre de bits utilisés**

La règle à retenir est la même que pour le nombre de chiffres utilisés en base .

|  |
| --- |
| 111 11  12 345 789  + 99 854 123  -----------  112 199 912 |

1. Pour l'addition  
   Soit le nombre de chiffres à utiliser pour écrire le résultat d'une somme de deux nombres écrits avec chiffres.  
   En se rappelant des additions vues à l'école primaire (voir ci-contre), il est clair que est égal à au plus .  
     
   C'est la même chose en binaire. Soit deux nombres entiers relatifs et représentés sur bits.  
      
      
   En faisant la somme des termes de gauche, du milieu et de droite :  
      
    (en utilisant le fait que )  
   Ainsi peut se représenter avec au plus bits.  
   En corollaire de cette propriété on pourrait parler avec enthousiasme des dépassement de capacité (que se passe-t-il sur une machine en bits si on obtient une somme sur bits ?) …

Soit la somme de deux entiers représentés sur bits.  
Alors peut se représenter sur au plus ) bits

1. Pour la multiplication  
   Soit le nombre de chiffres à utiliser pour écrire le résultat d'un produit de deux nombres écrits avec et chiffres.  
   En se rappelant que on a aisément l'intuition que .  
     
   C'est la même chose en binaire. Soit deux entiers relatifs et représentés sur et bits.  
      
      
   En faisant le produit on obtient (cela demande un peu de réflexion) :  
   Bref au vu de ce calcul on en déduit (à cause du terme de droite) qu'il faut au plus bits pour représenter le produit.

Soit le produit de deux entiers représentés sur et bits.  
Alors peut se représenter sur au plus bits.

1. **Exercices**

**Exercice 1 :**

Coder les entiers suivants en complément à deux sur un octet :

**Exercice 2 :**

Donner la valeur des entiers représentés en complément à par les octets ci-dessous :

* 1001 0011
* 1010 1010
* 0110 0110
* 0000 0010
* 1111 1111
* 1000 0000
* 0111 1111
* 1010 0101

**Exercice 2 :**

Quelle est la plage d'entiers relatifs que l'on peut coder en complément à sur bits ?

**Exercice 3 :**

*Essayez* d'effectuer les additions binaires suivantes.  
(Les additions en binaire seront vues en cours ultérieurement.)

|  |  |  |  |
| --- | --- | --- | --- |
| 0010 1000  +0011 0110  ----------- | 0000 1010  +1111 1110  ----------- | 1010 1000  +0001 1011  ----------- | 0110 1001  +1000 1010  ----------- |

**Exercice 4 :**

Dans chacun des cas, évaluez le nombre de bits nécessaires pour coder en complément à les entiers et indiqués puis le nombre de bits nécessaires pour coder les entiers et .

* et
* et
* et