

Architecture des ordinateurs

L3 Informatique 2013-2014

TD 1 - Représentation des entiers

Points abordés

- Représentation des nombres entiers en différentes bases.
 - Représentations des nombres signés par complément à 2.
 - Calcul en précision fixée.
-

Exercice 1. (Changement de bases)

1. Convertir $(41)_{10}$, $(63)_{10}$, $(128)_{10}$, $(3945)_{10}$ en binaire et en hexadécimal.
2. Convertir $(11000001)_2$, $(1111)_2$, $(11101)_2$ en décimal et en hexadécimal.
3. Convertir $(2BA)_{16}$, $(1CC)_{16}$, $(F0)_{16}$ en décimal et en binaire.

Exercice 2. (Calculs en précision fixée)

Effectuer les calculs suivants sur 8 bits de précision en indiquant pour chacun si un dépassement de capacité (*carry*) est déclenché :

1. $(17)_{10} + (12)_{10}$, $(204)_{10} + (115)_{10}$, $(230)_{10} + (78)_{10}$, $(47)_{10} + (208)_{10}$.
2. $(8)_{10} \times (15)_{10}$, $(18)_{10} \times (15)_{10}$, $(64)_{10} \times (32)_{10}$.

Exercice 3. (Entiers signés et complément à 2)

1. Représenter les entiers suivants en complément à 2 sur 8 bits :

$$(0)_{10}, (1)_{10}, (-1)_{10}, (32)_{10}, (-32)_{10}, (131)_{10} \text{ et } (-100)_{10}$$

2. Effectuer les calculs suivants sur 8 bits de précision. On indiquera pour chaque opération si elle engendre un dépassement de capacité (*overflow*) :

$$(15)_{10} - (19)_{10}, \quad (-32)_{10} + (13)_{10}, \quad (-32)_{10} - (17)_{10}.$$

Exercice 4. (Représentations des suites de bits)

On considère les nombres binaires suivants : 11001100, 01011100, 00111101 et 10100101

1. Quels entiers (en base dix) représentent-ils s'ils codent des entiers non signés ?
2. Quels entiers (en base dix) représentent-ils s'ils codent des entiers signés en complément à 2 ?
3. Additionnez-les deux à deux (en binaire) et interprétez ensuite les résultats en entiers non signés et en entiers signés. Y-a t'il des cas où certaines opérations donnent lieu à un dépassement de capacité dans une interprétation et pas dans l'autre ?

Exercice 5. (Système de numération romain)

On considère le système de numérotation romaine :

- Les symboles utilisés sont (avec leurs valeurs respectives en décimal)

$$I(1), \quad V(5), \quad X(10), \quad L(50), \quad C(100), \quad D(500) \quad \text{et} \quad M(1000).$$

- Tout symbole placé à la droite d'un symbole de valeur supérieure s'ajoute à ce dernier.
- Toute lettre d'unité placée immédiatement à la gauche d'un symbole se retranche à ce dernier. Les lettres d'unité sont I pour V et X, X pour L et C, C pour D et M.
- Les symboles apparaissent en ordre décroissant, sauf pour les valeurs à retrancher selon la règle précédente.

Par exemple, VI vaut $5 + 1 = 6$, IV vaut $5 - 1 = 4$ et IX vaut 9 mais IC n'a pas de sens.

1. Représenter $(17)_{10}$, $(78)_{10}$, $(204)_{10}$ et $(2014)_{10}$ en numérotation romaine.
2. Effectuer les opérations suivantes directement en chiffres Romains : $(17)_{10} + (12)_{10}$, $(204)_{10} + (115)_{10}$, $(8)_{10} \times (15)_{10}$.
3. Quelle différence entre les systèmes de numérotation romain et arabe vous semble essentielle du point de vue des calculs ?

Exercice 6. (Changement de base, bis)

1. Exprimer l'entier $(2911)_{10}$ en base sept et l'entier $(1036)_7$ en décimal.
2. Donner deux méthodes générales pour convertir un nombre d'une base b vers une base b' .

Exercice 7. (Calcul rapide en binaire)

Dans cet exercice, les entiers (non signés) sont représentés sur huit bits.

1. Donner une méthode pour multiplier un nombre binaire par 2. Généraliser pour la multiplication par 2^k , $k \geq 0$.
2. Donner une méthode pour calculer la partie entière de la division d'un nombre binaire par 2. Généraliser pour la division par 2^k , $k \geq 0$.
3. Donner une méthode pour calculer le reste de la division d'un nombre binaire par 2. Généraliser pour le reste pour la division par 2^k , $k \geq 0$.
4. Proposer un algorithme pour multiplier rapidement deux nombres 32 bits $(a_{31}a_{30} \dots a_0)_2$ et $(b_{31}b_{30} \dots b_0)_2$ sachant que seul un petit nombre de b_i vaut 1 (mais on ne sait pas lesquels).