Représentation binaire d'un entier relatif

Table des matières

Prérequis : écriture binaire d'un nombre.
Objectifs :

- Connaitre la représentation binaire d'un entier relatif.
- Comprendre la méthode de codage dite *complément à deux*.

Outil nécessaire : aucun (activité débranchée).

I) Rappel sur le codage d'un entier naturel

Un entier naturel est un nombre entier positif ou nul. Le choix à faire (c'est-à-dire le nombre de bits à utiliser) dépend de la fourchette des nombres que l'on désire utiliser. Pour coder des nombres entiers naturels compris entre 0 et 255, il suffit de 8 bits (un octet) car $2^8 = 256$. D'une manière générale un codage sur n bits permet de représenter des nombres entiers naturels compris entre 0 et $2^n - 1$.

Exemples : $9_{10} = 0000\,1001_2$, $128_{10} = 1000\,0000_2$, etc.

II) Représentation binaire d'un entier relatif

1. Le complément à deux

Un entier relatif est un entier pouvant être négatif. Il faut donc coder le nombre de telle façon que l'on puisse savoir s'il s'agit d'un nombre positif ou d'un nombre négatif, et il faut de plus que les règles d'addition soient conservées. L'astuce consiste à utiliser un codage que l'on appelle complément à deux. Cette représentation permet d'effectuer les opérations arithmétiques usuelles naturellement.

- Un entier positif ou nul sera représenté en binaire (base 2) comme un entier naturel, à la seule différence que le bit de poids fort (le bit situé à l'extrême gauche) représente le signe. Il faut donc s'assurer pour un entier positif ou nul que le bit de poids fort est à zéro (0 correspond à un signe positif, 1 à un signe négatif). insi, si on code un entier naturel sur 4 bits, le nombre le plus grand sera 0111 (c'est-à-dire 7 en base décimale).
  - ⋆ Sur 8 bits (1 octet), l'intervalle de codage est $[-128, 127]$.
  - ⋆ Sur 16 bits (2 octets), l'intervalle de codage est $[-32\,768, 32\,767]$.
  - ⋆ Sur 32 bits (4 octets), l'intervalle de codage est $[-2\,147\,483\,648, 2\,147\,483\,647]$.

  D'une manière générale le plus grand entier positif codé sur n bits sera $2^{n-1} - 1$.
- Un entier négatif sera représenté grâce au codage en **complément à deux**.

Cette activité est inspirée du site https://www.apprendre-en-ligne.net et du livre « Informatique et sciences du numérique » de Gilles Dowek (Editions Eyrolles), libre d'accès en ligne.

Complément à 2
Méthode 1

Principe du complément à deux :

① écrire la valeur absolue du nombre en base 2. Le bit de poids fort doit être égal à 0 ;

② inverser les bits (les 0 deviennent des 1 et vice versa : on écrit ce qu'on appelle le complément à un) ;

③ on additionne 1 au résultat (les dépassements sont ignorés).

Cette opération correspond au calcul de $2^n - |x|$, où n est la longueur de la représentation et $|x|$ la valeur absolue (c'est-à-dire le nombre sans son signe) du nombre x à coder.
Ainsi -1 s'écrit comme $256 - 1 = 255 = 1111\,1111_2$ sur 8 bits.

Exemple
On désire coder la valeur -19 sur 8 bits. Il suffit :

① d'écrire 19 en binaire : 0001 0011 ;

② d'écrire son complément à 1 : 1110 1100 ;

③ d'additionner 1 : 1110 1101.

La représentation binaire de -19 sur 8 bits est donc 1110 1101.

On remarquera qu'en additionnant un entier positif et son complément à deux on obtient 0 : par exemple $0001\,0011 + 1110\,1101 = 0000\,0000$ (avec une retenue de 1 qui est éliminée).

Complément à 2
Méthode 2

Astuce
Pour transformer « à la main » un nombre binaire en son complément à deux, on parcourt le nombre de droite à gauche en laissant inchangés les bits jusqu'au premier 1 (compris), puis on inverse tous les bits suivants. Prenons comme exemple le nombre 20 : 0001 0100.

① On garde la partie à droite telle quelle : .....100.

② On inverse la partie de gauche après le premier un : 1110 1....

③ Et voici -20 : 1110 1100.

Autre vision du complément à deux
La représentation binaire sur n bits d'un entier relatif x :

- si l'entier relatif x est positif ou nul, on le représente comme l'entier naturel x en binaire.
- s'il est strictement négatif, on le représente comme l'entier naturel $x + 2^n$.

Par exemple la représentation binaire sur huit bits de -128 est $-128+2^8 = -128+256 = 128 = 1000\,0000_2$.

Exercice 1

Trouver la représentation binaire sur huit bits des entiers relatifs 127 et -127.

Anecdote historique
Le 4 juin 1996, une fusée Ariane 5 a explosé 40 secondes après l'allumage. La fusée et son chargement avaient coûté 500 millions de dollars. La commission d'enquête a rendu son rapport au bout de deux semaines. Il s'agissait d'une erreur de programmation dans le système inertiel de référence. À un moment donné, un nombre codé en virgule flottante sur 64 bits (qui représentait la vitesse horizontale de la fusée par rapport à la plate-forme de tir) était converti en un entier sur 16 bits. Malheureusement, le nombre en question était plus grand que 32 768 (le plus grand entier que l'on peut coder sur 16 bits) et la conversion a été incorrecte.

Exercice 2

① Coder les entiers relatifs suivants sur 8 bits (16 si nécessaire) : 456, -1, -56, -5642.

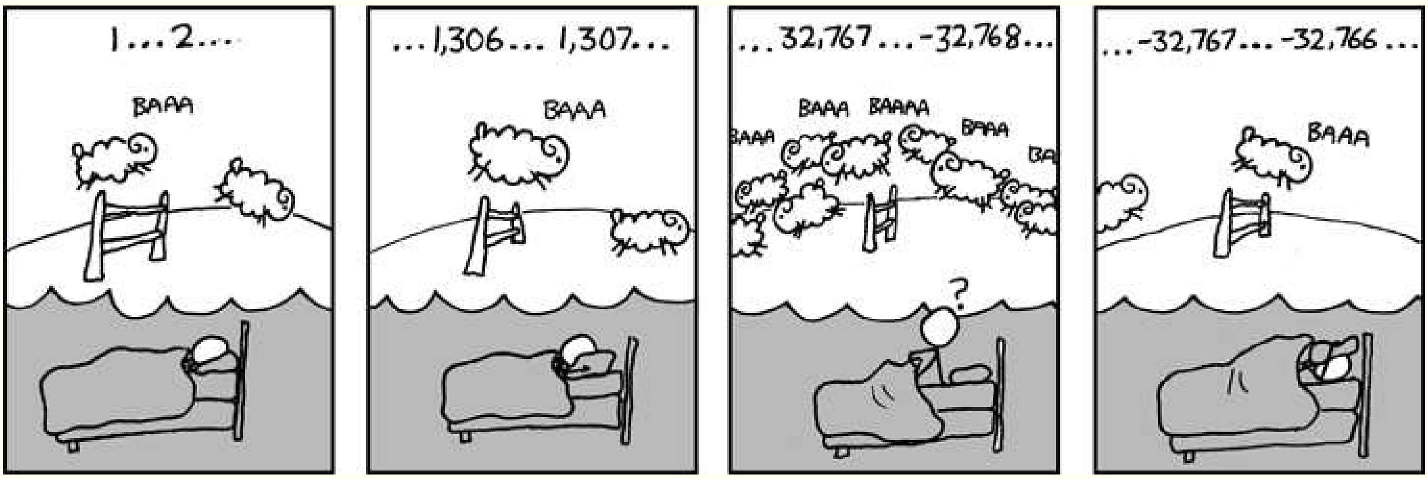
② Donner les signes des trois entiers relatifs suivants ?
- 0110 1100 ;
- 1110 1101 ;
- 1010 1010 1010 1010.

Exercice 3

Quels entiers relatifs peut-on représenter avec des mots de 8 bits (octets) ? Combien sont-ils ? Même question avec des mots de 32 bits et 64 bits.

Exercice 4

Expliquer ce rêve étrange (source de l'image : http ://xkcd.com/571).



Exercice 5

Certains logiciels utilisent la représentation POSIX du temps, dans laquelle le temps est représenté comme un nombre de secondes écoulées avant ou depuis le 1er janvier 1970 à 0 heure. Sur les ordinateurs 32 bits, la plupart des systèmes d'exploitation représentent ce nombre comme un nombre entier <u>signé</u> de 32 bits.

① Quel est le nombre maximal de secondes que l'on peut représenter ?

② À quelle date cela posera-t-il un problème (jour, mois, année, heures, minutes, secondes) ?
Indications :

a) afin de tenir compte des années bissextiles, comptez par cycles de 4 ans composés de $4 \times 365 + 1 = 1461$ jours ;

b) l'an 2 000 est une année bissextile.

③ Que se passera-t-il une seconde plus tard ? Quel sera le nombre de secondes affiché (en base 10) ? À quelle date cela correspond-il ?

④ Proposer une solution pour empêcher ce problème de survenir.

Trouver la représentation décimale d'un entier relatif x en binaire sur n bits
Deux méthodes

① Il faut savoir que pour trouver le nombre dont le complément à deux est x, il suffit de calculer le complément à deux de x, la première méthode s'en déduit donc :

a) si le bit de poids fort de x est 0, alors il suffit de procéder à une conversion de binaire en décimal ;

b) sinon, il suffit de calculer le complément à deux de x, de procéder à une conversion de binaire en décimal puis de penser à prendre l'opposé du résultat.

② La deuxième méthode se déduit de la remarque « autre vision » ci-dessus : on convertit le mot binaire x en le nombre décimal p puis,

a) si p est strictement inférieur à 2^{n-1}, c'est l'entier relatif représenté ;

b) s'il est supérieur ou égal à 2^{n-1}, l'entier relatif représenté est $p - 2^n$.

Exemples

- Le mot 0000 0000 représente l'entier naturel 0 et donc aussi l'entier relatif 0.
- Le mot 1000 0000 représente l'entier naturel $128 = 2^7$ et donc l'entier relatif $128 - 2^8 = 128 - 256 = -128$.

Exercice 6

① Trouver les représentations décimales des entiers relatifs dont les représentations binaires sur huit bits sont respectivement 0111 1111 et 1000 0001.

② Que valent en base dix les trois entiers relatifs codés ci-dessous en binaire ?
- 0110 1100 ;
- 1110 1101 ;
- 1010 1010 1010 1010.

2. Limites de la représentation complément à deux

Taille du mot	Bits	Valeurs décimales
n bits	1 bit de signe n-1 bits de valeur	-2^{n-1} à $+2^{n-1} - 1$
8 bits	1 bit de signe 7 bits de valeur	-128 à $+127$
16 bits	1 bit de signe 15 bits de valeur	$-32\,768$ à $+32\,767$
32 bits	1 bit de signe 31 bits de valeur	$-2\,147\,483\,648$ à $+2\,147\,483\,647$

Les limites du codage des entiers relatifs sont principalement dues au dépassement de capacité (*overflow* en anglais) lors d'un calcul. Une addition de deux nombres positifs ou négatifs peut entrainer un dépassement de capacité, celui-ci peut être détecté en regardant le signe du résultat par rapport au signe des deux opérandes (par exemple deux nombres positifs donnent un résultat négatif...).

Remarque
Dans les versions Python 2.x, lorsque la capacité des entiers machine (**int** : 32 bits ; **long** : 64 bits) a été dépassée, les nombres sont suivis du marqueur **L**, qui explicite qu'on passe dans un autre type appelé **long**.

En Python 3.x, les types **int** et **long** sont fusionnés, le marqueur **L** n'est plus utilisé et les entiers sont toujours de taille illimitée. Toutefois, lorsque les entiers dépassent une certaine taille, le calcul est considérablement ralenti par des calculs supplémentaires.

III) Opérations sur deux entiers relatifs

1. Addition de deux entiers relatifs

Vérifier que la table d'addition en base 2 est la suivante.

+	0	1
0	0	1
1	1	10

Exercice 7

① Lorsque c'est possible, coder les entiers relatifs 75, 128, 201, -75 et -201 sur huit bits en utilisant le codage du complément à deux.

② Lorsque c'est possible, toujours sur huit bits, poser les opérations suivantes sur des entiers signés (avec les codes complément à deux bien sûr !) puis vérifier les résultats avec la base dix : $75+125$, $128+21$, $75-75$, $201-56$, $75+27$, $75-27$, $27-75$, $-27-75$, $56-128$ et $-56-201$.

③ Justifier les impossibilités et trouver un moyen d'y remédier.

④ Quel est le nombre minimal de bits nécessaires pour représenter toutes les sommes possibles de deux nombres chacun représenté sur n bits ?

2. Multiplication de deux entiers relatifs

Vérifier que la table de multiplication en base 2 est la suivante.

×	0	1
0	0	0
1	0	1

Exercice 8

① Coder les entiers relatifs 65, 12 008, 201 et -201 sur seize bits en utilisant le codage du complément à deux.

② Lorsque c'est possible, poser les opérations suivantes (avec les codes complément à deux bien sûr !) puis vérifier les résultats avec la base dix : $65 \times 12\,008$, 65×201 et $65 \times (-201)$.

③ Justifier les impossibilités et trouver un moyen d'y remédier.

④ Quel est le nombre minimal de bits nécessaires pour représenter tous les produits possibles de deux nombres chacun représenté sur n bits ?

À lire pour les curieux :
https://lejournal.cnrs.fr/articles/la-multiplication-reinventee.