

Sujet DM : Entiers binaires et hexadécimal

Ce sujet a pour but de revoir les conversions binaire et hexa ainsi que les opérations sur ces nombres.

La fin du sujet correspondra à de l'avance sur le programme avec les notions d'entiers positifs et négatifs.

Conversion d'une base à une autre

Conversion binaire vers décimal:

1. 1010_2
2. 1101_2
3. 10011_2
4. 11100_2

Conversion décimal vers binaire:

1. 52_{10}
2. 67_{10}
3. 89_{10}
4. 123_{10}

Conversion hexadécimal vers décimal:

1. $A3_{16}$
2. $5F_{16}$
3. $B7_{16}$
4. $9E_{16}$

Conversion décimal vers hexadécimal:

1. 345_{10}
2. 456_{10}
3. 567_{10}
4. 678_{10}

Opérations sur les nombres binaires entiers positifs

- Addition:
 - $1010_2 + 1101_2$
 - $10011_2 + 11100_2$
 - $101101_2 + 110010_2$
 - $1001101_2 + 1110010_2$

- Soustraction:
 - $1101_2 - 1010_2$
 - $11100_2 - 10011_2$
 - $110010_2 - 101101_2$
 - $1110010_2 - 1001101_2$
- Multiplication:
 - $1010_2 \times 1101_2$
 - $10011_2 \times 11100_2$
 - $101101_2 \times 110010_2$
 - $1001101_2 \times 1110010_2$
- Division: Donner le reste et le quotient.
 - $1101_2 \div 1010_2$
 - $11100_2 \div 10011_2$
 - $110010_2 \div 101101_2$
 - $1110010_2 \div 1001101_2$

Pour aller plus loin : les nombres négatifs:

Il existe plusieurs méthodes pour pouvoir avoir la représentation des nombres négatifs en binaire.

Bit de poids fort

Il existe une méthode qui permet de définir ce qu'est un entier positif ou négatif **sur un nombre de bits donnés**.

Cette méthode revient à attribuer au bit de poids fort le signe de l'entier. **1 correspond aux négatifs, 0 aux positifs**.

Ainsi, on a $00011011_2 = 27_{10}$ et $10011011 = -27_{10}$.

- Donner la représentation sur 5 bits de -13 et 13 .

Un problème se pose : 0 n'est ni positif, ni négatif. Avec cette méthode, il a donc 2 représentations avec cette notation, ici sur 5 bits avec le bit de poids fort qui représente le signe : $10000_2 = -0$ et $00000_2 = +0$.

Complément à deux

Une autre technique demande plus d'opérations mais évite ce soucis : on parle du complément à 2.

Le complément à deux est obtenu en inversant tous les bits d'un nombre : Les 0 deviennent 1 et inversement. Enfin, on additionne 1 au résultat.

On considère le bit de poids fort comme étant bit de signe : Ainsi, sur 4 bits on n'aura que 2^3 possibilités donc, on peut représenter de -8 à 7.

Cette méthode a l'avantage de n'avoir qu'une seule représentation pour zéro et permet des opérations d'addition et de soustraction simples et uniformes.

Exemples :

Positif vers négatif

On prend $+14_{10} = 01110_2$

On inverse tous ses bits : 10001_2

On rajoute 1 : $10001_2 + 1_2 = 10010_2 = -14_{10}$

Négatif vers positif

On prend $-8_{10} = 11000_2$ sur 5 bits.

On retire 1 : $11000_2 - 1_2 = 10111_2$

On inverse tous les bits: $01000_2 = 8_{10}$.

On peut même réaliser des opérations :

On sait que $8_{10} = 01000_2$ en complément à 2 avec le bit de poids fort à 0 qui indique qu'il est positif.

On va réaliser l'addition de -14 et 8 :

$$10010_2 + 01000_2 = 11010_2$$

On sait que $-14 + 8 = -6$, on va vérifier :

$$00110_2 = 6_{10}.$$

11001_2 en inversant tous les bits.

$$11001_2 + 1_2 = 11010_2 = -6_{10}.$$

À l'aide de cela :

Exercices

Convertir les nombres entiers négatifs suivants de la base 10 en binaire en utilisant le complément à deux. Supposons une largeur de mot de 8 bits pour cet exercice.

1. -18
2. -127
3. -88
4. -45
5. -1
6. -34
7. -99
8. -73
9. -56
10. -128