Représentation des entiers relatifs

Christophe Viroulaud

Première - NSI

DonRep 02

Représentation des entiers relatifs

Christophe Viroulaud

Première - NSI

DonRep 02

2021-08-29

Un système *64 bits* peut représenter 2^{64} entiers.

```
>>> import sys
>>> sys.maxsize
9223372036854775807
```

Code 1 – Cette valeur correspond à $2^{63} - 1$.

Observation

Un des bits ne semble pas utilisé.

Un système *64 bits* peut représenter 2^{64} entiers.

```
>>> import sys
>>> sys.maxsize
9223372036854775807
```

Code 1 – Cette valeur correspond à $2^{63} - 1$.

Observation

Un des bits ne semble pas utilisé.

Comment sont représentés les entiers négatifs en mémoire ?

Comment sont représentés les entiers négatifs en mémoire ?

Sommaire

Addition de deux nombres binaires

Une addition en base 2 applique les mêmes principes qu'en base 10 :
- $0 + 0 = 0$
- $1 + 0 = 1$
- $1 + 1 = 0$ et une retenue de 1
- $1 + 1 + 1 = 1$ et une retenue de 1

Addition de deux nombres binaires

Une addition en base 2 applique les mêmes principes qu'en base 10 :

- $0 + 0 = 0$
- $1 + 0 = 1$
- $1 + 1 = 0$ et une retenue de 1
- $1 + 1 + 1 = 1$ et une retenue de 1

Dans un mot mémoire de 1 octet :

$$25_{10} = 00011001_2$$

D'après l'intro dans 8 bits on peut stocker des nombres allant jusqu'à

$$2^7 - 1 = 127_{10} = 01111111_2$$

.

Dans un mot mémoire de 1 octet :

$$25_{10} = 00011001_2$$

Activité 1 :
1. Convertir 25 et 12 en base 2.
2. Effectuer l'addition binaire de ces nombres.
3. Convertir le résultat en base 10. Le résultat est-il correct ?

Activité 1 :

1. Convertir 25 et 12 en base 2.
2. Effectuer l'addition binaire de ces nombres.
3. Convertir le résultat en base 10. Le résultat est-il correct ?

Correction

$$\begin{array}{ccccccccc} & & & 1 & 1 & & & & \\ & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{array}$$

Correction

$$\begin{array}{ccccccccc} & & & 1 & 1 & & & & \\ & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{array}$$

$0\times2^7+0\times2^6+1\times2^5+0\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0 = 37$

Correction

$$0\times2^7+0\times2^6+1\times2^5+0\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0 = 37$$

Sommaire

1. Addition de deux nombres binaires
2. Une représentation naïve des entiers négatifs
2.1 Bit de poids fort
2.2 Inconvénients de la représentation
3. Le complément à 2 puissance n

Sommaire

Bit de poids fort

Le bit le plus à gauche de la représentation n'est pour l'instant pas utilisé. C'est le **bit de poids fort**.

À retenir

Pour représenter un nombre entier, il faut connaître la taille du mot mémoire.

Bit de poids fort

Le bit le plus à gauche de la représentation n'est pour l'instant pas utilisé. C'est le **bit de poids fort**.

À retenir

Pour représenter un nombre entier, il faut connaître la taille du mot mémoire.

Une première idée serait d'utiliser ce bit comme marqueur de signe :
- 0 pour un entier positif,
- 1 pour un entier négatif.
Ainsi l'entier -25 serait encodé dans un mot mémoire de 1 octet :
$$-25_{10} = 10011001_2$$

Une première idée serait d'utiliser ce bit comme marqueur de signe :

- 0 pour un entier positif,
- 1 pour un entier négatif.

Ainsi l'entier -25 serait encodé dans un mot mémoire de 1 octet :

$$-25_{10} = 10011001_2$$

Sommaire
1. Addition de deux nombres binaires
2. Une représentation naïve des entiers négatifs
2.1 Bit de poids fort
2.2 Inconvénients de la représentation
3. Le complément à 2 puissance n

Sommaire

Le zéro

Dans un système *8 bits* le zéro est représenté par 00000000_2. Cependant 10000000_2 se traduit par -0. Il y a donc deux représentations pour zéro.

Le zéro

Dans un système *8 bits* le zéro est représenté par 00000000_2. Cependant 10000000_2 se traduit par -0. Il y a donc deux représentations pour zéro.

Erreur d'addition
$-25 + 12$
Avec cette représentation :
$2^5 + 2^2 + 2^0 = 37$
$10100101_2 = -37_{10}$
$-25 + 12 \neq -37$
À retenir
Cette représentation est erronée.

Erreur d'addition

$$-25 + 12$$

$$\begin{array}{ccccccccc}
 & & & 1 & & 1 & & & \\
 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
+ & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
\hline
 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1
\end{array}$$

Avec cette représentation :

$$2^5 + 2^2 + 2^0 = 37$$

$$10100101_2 = -37_{10}$$

$$-25 + 12 \neq -37$$

À retenir

Cette représentation est erronée.

Sommaire

Sommaire

Définition

Le complément à 2 puissance n est une représentation qui ne change rien pour les entiers positifs. Ainsi sur 8 bits :

0	1	1	1	1	1	1	1	=	127
0	...							=	...
0	0	0	0	0	0	1	0	=	2
0	0	0	0	0	0	0	1	=	1
0	0	0	0	0	0	0	0	=	0

Définition

Le complément à 2 puissance n est une représentation qui ne change rien pour les entiers positifs. Ainsi sur 8 bits :

0	1	1	1	1	1	1	1	=	127
0	...							=	...
0	0	0	0	0	0	1	0	=	2
0	0	0	0	0	0	0	1	=	1
0	0	0	0	0	0	0	0	=	0

Par contre la valeur $2^n - |x|$ représente l'entier négatif x.
Ainsi sur 8 bits, -1 s'écrit

$$2^8 - 1 = 256 - 1 = 255_{10} = 11111111_2$$

1	1	1	1	1	1	1	1	=	-1	$2^8 - \|-1\| = 255$
1	1	1	1	1	1	1	0	=	-2	$2^8 - \|-2\| = 254$
1	...							=	...	
1	0	0	0	0	0	0	1	=	-127	$2^8 - \|-127\| = 129$
1	0	0	0	0	0	0	0	=	-128	$2^8 - \|-128\| = 128$
0	1	1	1	1	1	1	1	=	127	
0	...							=	...	
0	0	0	0	0	0	1	0	=	2	
0	0	0	0	0	0	0	1	=	1	
0	0	0	0	0	0	0	0	=	0	

Par contre la valeur $2^n - |x|$ représente l'entier négatif x.
Ainsi sur 8 bits, -1 s'écrit

$$2^8 - 1 = 256 - 1 = 255_{10} = 11111111_2$$

1	1	1	1	1	1	1	1	=	-1	$2^8 - \|-1\| = 255$
1	1	1	1	1	1	1	0	=	-2	$2^8 - \|-2\| = 254$
1	...							=	...	
1	0	0	0	0	0	0	1	=	-127	$2^8 - \|-127\| = 129$
1	0	0	0	0	0	0	0	=	-128	$2^8 - \|-128\| = 128$
0	1	1	1	1	1	1	1	=	127	
0	...							=	...	
0	0	0	0	0	0	1	0	=	2	
0	0	0	0	0	0	0	1	=	1	
0	0	0	0	0	0	0	0	=	0	

Sommaire

1. Addition de deux nombres binaires
2. Une représentation naïve des entiers négatifs
3. Le complément à 2 puissance n
3.1 Définition
3.2 Calculer le complément à 2
3.3 Intérêt de la méthode

Sommaire

Calculer le complément à 2

Pour coder (-20) :
- Prendre le nombre positif 20 : 00010100
- Inverser les bits : 11101011
- Ajouter 1 : 11101100
- -20 : 11101100

Calculer le complément à 2

Pour coder (-20) :

- Prendre le nombre positif 20 : 00010100
- Inverser les bits : 11101011
- Ajouter 1 : 11101100
- -20 : 11101100

Seconde méthode

Garder tous les chiffres depuis la droite jusqu'au premier 1 (compris) puis d'inverser tous les suivants.
- Prendre le nombre positif 20 : 00010100
- Garder la partie à droite telle quelle : 00010100
- Inverser la partie de gauche après le premier un : 11101100
- -20 : 11101100

Seconde méthode

Garder tous les chiffres depuis la droite jusqu'au premier 1 (compris) puis d'inverser tous les suivants.

- Prendre le nombre positif 20 : 00010100
- Garder la partie à droite telle quelle : 00010100
- Inverser la partie de gauche après le premier un : 11101100
- -20 : 11101100

Activité 2 : Calculer le complément à 2 (sur 1 octet) de -25.

Activité 2 : Calculer le complément à 2 (sur 1 octet) de -25.

2021-08-29

Correction

- $25_{10} = 00011001_2$
- $0001100\underline{1}$
- $-25_{10} = 11100111$

Correction

- $25_{10} = 00011001_2$
- $0001100\underline{1}$
- $-25_{10} = 11100111$

Sommaire
1. Addition de deux nombres binaires
2. Une représentation naïve des entiers négatifs
3. Le complément à 2 puissance n
3.1 Définition
3.2 Calculer le complément à 2
3.3 Intérêt de la méthode

Sommaire

Il n'y a qu'un seul zéro.

Intérêt de la méthode

Il n'y a qu'un seul zéro.

Addition

$-25 + 12 = 11100111 + 00001100 = 11110011 = 243$ et $243 - 2^8 = 243 - 256 = -13$

$$-25 + 12$$

$$\begin{array}{ccccccccc} & & & & 1 & 1 & & & \\ & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ + & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \end{array}$$

Avec cette représentation :

$$2^7 + 2^6 + 2^5 + 2^4 + 2^1 + 2^0 = 243$$

$$243 - 2^8 = 243 - 256 = -13$$

Addition

$-25 + 12 = 11100111 + 00001100 = 11110011 = 243$ et $243 - 2^8 = 243 - 256 = -13$

$$-25 + 12$$

$$\begin{array}{ccccccccc} & & & & 1 & 1 & & & \\ & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ + & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \end{array}$$

Avec cette représentation :

$$2^7 + 2^6 + 2^5 + 2^4 + 2^1 + 2^0 = 243$$

$$243 - 2^8 = 243 - 256 = -13$$

À retenir
Les nombres entiers négatifs sont représentés par le complément à 2.

À retenir

Les nombres entiers négatifs sont représentés par le complément à 2.