

Le langage binaire et hexadécimal

Date de dernière mise à jour : 18/12/2013 à 16:08

Source : <http://www.vulgarisation-informatique.com/binaire-hexa.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

Le langage binaire vient de la notation de contacteurs, notée 0 pour **faux** (quand le courant ne passe pas) et 1 pour **vrai** (quand le courant passe).

Ce codage est nommé **base binaire** et est utilisé par l'ordinateur pour comprendre et traiter les données. Les humains travaillent avec dix chiffres, c'est la notation décimale (base 10).

Bit

Le bit vient du terme **Binary Digit** (soit 0 soit 1 en langage binaire). On travaille donc avec les puissances de 2. C'est ainsi qu'avec 3 bit, on peut avoir 2^n valeurs codables, soit 2^3 (8) dans notre cas. Il y a donc 8 possibilités.

Valeur décimale

Bit 3 (2^2)

Bit 2 (2^1)

Bit 1 (2^0)

Calcul

$$0 = 0 * (2^2) + 0 * (2^1) + 0 * (2^0)$$

1

1

$$1 = 0 * (2^2) + 0 * (2^1) + 1 * (2^0)$$

2

1

$$2 = 0 * (2^2) + 1 * (2^1) + 0 * (2^0)$$

3

1

1

$$3 = 0 * (2^2) + 1 * (2^1) + 1 * (2^0)$$

4

1

$$4 = 1 * (2^2) + 0 * (2^1) + 0 * (2^0)$$

5

1

1

$$5 = 1 * (2^2) + 0 * (2^1) + 1 * (2^0)$$

6

1

1

$$6 = 1 * (2^2) + 1 * (2^1) + 0 * (2^0)$$

7

1

1

1

$$7 = 1 * (2^2) + 1 * (2^1) + 1 * (2^0)$$

Le code binaire est un code pondéré car il s'écrit sous forme d'additions de puissances de 2. Pour passer du binaire à la valeur décimale correspondante (en base 10), on additionne les nombres en les élevant à leurs puissances de 2 respectives.

Binaire réfléchi (code GRAY)

Il existe une autre forme (non pondérée cette fois) du code binaire, le **code GRAY** (aussi appelé **binaire réfléchi**). On ne peut modifier qu'un seul bit à la fois lorsque l'on passe d'un nombre n à un nombre n+1 et il existe une symétrie verticale :

Valeur

Bits

1

1

2

1

1

3

1

4

1

1

5

1

1

1

6

1

1

7

1

Axe de symétrie

8

1

1

9

1

1

1

10

1

1

1

1

11

1

1

1

12

1

1

13

1

1

1

14

1

1

15

1

Octet

Un octet est un ensemble de 8 bit. Il permet de stocker des nombres pouvant aller jusqu'à 255 car il offre 2^8 soit 256 possibilités. C'est une unité très couramment utilisée en informatique.

2^7

2^6

2^5

2^4

2^3

2^2

2^1

2^0

128

64

32

16

8

4

2

1

Quartet

Peu utilisé en informatique, le quartet est un ensemble de 4 bit.

Octet, Kilo-octet, Méga-octet

-Un octet est un ensemble de 8 bit.

- Un Kilo-octet (Ko) vaut 1000 octets.
- Un Méga-octet (Mo) vaut 1000 Kilo-octets
- Un Giga-octet (Go) vaut 1000 Méga-octets
- Un Téra-octet (To) vaut 1000 Giga-octets
- Le Péta-octet (Po) vaut 1000 Téra-octets

Depuis 1998, la **commission électrotechnique internationale** a normalisé les unités suivantes en puissances de 2 :

- Le kibi-octet (Kio) qui est égal à 1024 octets.
- Le Mébi-octet (Mio) qui est égal à 1024 Kibi-octets.
- Le Gibi-octet (Gio) qui est égal à 1024 Mébi-octets.
- Le Tébi-octet (Tio) qui est égal à 1024 Gibi-octets.
- Le Pébi-octet (Pio) vaut 1024 Tébi-octets.

Opérations en binaire

Addition binaire

L'addition en binaire se fait de la même manière qu'en notation décimale: On commence à additionner les bits de droite (ceux appelés **bits de poids faible**) puis on pose une ou plusieurs retenue(s) lorsque la somme de deux bits de même poids dépasse 1 (chiffre maximum en binaire).

Voyons comment additionner en binaire les nombres 31 (011111) et 25 (011001)

```

1
1
1
1
1
+
1
1

1

1
1
1

```

Soustraction binaire

La soustraction en binaire se fait de la même manière qu'une addition, sauf que lorsque l'on soustrait un bit à un d'un bit à zéro, on soustrait une retenue pour le bit de poids plus élevé.

Voyons comment soustraire 25 du nombre 30 (il restera donc 5) :

1
1
1
1

-

1
1

1

1

1

Multiplication binaire

La multiplication binaire s'effectue de la même façon qu'en notation décimale : un nombre multiplié par 0 est égal à 0. On a donc :

$$-0*0=0$$

$$-0*1=0$$

$$-1*0=0$$

$$-1*1=1$$

Essayons via cet exemple de multiplier 31 (11111) par 6 (110) :

1
1
1
1
1

*

1

1

Résultats partiels

1
1
1
1
1

1
1
1
1
1

Somme des trois termes (résultat final)

1

1
1
1

1

Convertir des nombres binaires et décimaux

Pour convertir un nombre décimal en binaire, on a deux possibilités :

- La méthode des divisions successives (que nous allons voir en premier).
- La méthode des puissances.

On divise par deux le nombre jusqu'à ce que le nombre trouvé soit 1. Voici un exemple avec le nombre 27 :

27
2

1 (5)

13

2

1 (4)

6

2

(3)

3

2

1 (2)

1 (1)

On part ensuite du sens inverse (on prend les bits numérotés successivement 1; 2; 3; 4; 5 en commençant par le chiffre en rouge, bit de poids faible).

Donc $27 = 11011$.

Deuxième méthode : on utilise les puissances :

$$27 = 1 * 2^0 + 1 * 2^1 + 1 * 2^3 + 1 * 2^4$$

Le langage hexadécimal

Le langage hexadécimal est un langage en base 16 venu freiner l'accroissement en longueur des nombres binaires. Pour composer ce langage, les 10 premiers chiffres suivis des six premières lettres de l'alphabet ont été utilisés :

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Notation décimale

1

2

3

4

5

6

7

8

9

10

11

12

13
14
15

Notation binaire

1
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111

Notation hexadécimale

1
2
3
4
5
6
7
8
9
A
B
C
D
E
F

Conversion décimal vers hexadécimal :

Méthode des divisions successives (exemple pour 36 en décimal) :

36
16

4

2

On procède de la même manière qu'en binaire, à savoir que l'on remonte les valeurs de bas en haut. On obtient 36 (décimal) = 24 (hexadécimal). On peut retrouver et contrôler notre résultat en constatant que :

$$2 * 16^1 + 4 * 16^0 = 36$$

De l'hexadécimal au binaire :

Conservons notre nombre de 24 en hexadécimal (36 en décimal) :

On utilise ce qu'on appelle le code **1248** (se lit "un deux quatre huit").

2

4

0010

0100

2 = 0010 en binaire, et 4 est égal à 0100 en binaire. On en déduit ensuite le nombre binaire final qui est la concaténation de nos deux nombres binaires :

0010 0100

Ce nombre est égal à 36 en décimal, nous avons donc converti notre nombre hexadécimal en binaire, puis en décimal.

Source : <http://www.vulgarisation-informatique.com/binaire-hexa.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)