

Animation

1. convertisseur binaire / décimal / hexadécimal

Table des matières

Introduction

I) le codage binaire

- 1) définition
- 2) le poids d'un bit
- 3) conversion binaire décimale
- 4) conversion décimale -> binaire
- 5) les différentes unités binaires

II) La numération hexadécimale

- 1) définition
- 2) Conversion hexadécimal <-> décimal

Programme officiel

Introduction: texte officiel du BO.

Dans un contexte informatique, l'information est représentée par une suite de caractères ou, ce qui revient au même, de nombres. La numérisation est l'opération qui transforme un objet réel, issu du monde physique (image, son, etc.) en informations exploitables par un ordinateur ou, plus généralement, toute machine numérique. À cause de l'échantillonnage sous-jacent, la numérisation induit des effets importants sur la qualité de l'information numérique. Elle entraîne des conditions spécifiques de création, de stockage, de traitement et de circulation de l'information. Les capacités de traitement et de stockage des ordinateurs croissent de façon continue depuis leur apparition. Il est donc crucial d'organiser ces flux d'informations, que ce soit en local sur une machine, ou bien de façon distribuée sur un réseau. L'intégration croissante du numérique dans les activités humaines et la numérisation de l'information suscitent des transformations culturelles, socio-économiques, juridiques et politiques profondes qui font apparaître de nouvelles opportunités et de nouveaux risques qu'il convient d'étudier.

I) le codage binaire

1) définition

La numération binaire est due aux travaux des mathématiciens Leibniz, Boole et Couffignal. Son but est de représenter des nombres avec le moins de symboles possibles. Elle a été définie bien avant l'invention de l'ordinateur et s'est révélée admirablement adaptée à cet outil. Comme notre numération décimale usuelle, la numération binaire est de type POSITION : la valeur d'un chiffre dépend de sa place dans le nombre.

On appelle BIT (BInary digiT, C. Shannon 1938) le plus petit élément d'information stockable par un ordinateur. Le bit est la particule élémentaire d'information. Un bit ne peut prendre que deux valeurs (0 ou 1) correspondant à deux états possibles d'un élément de circuit électrique (tension présente ou nulle aux bornes d'un dipôle). L'opération qui consiste à transformer (ou coder) une information en une suite de bits est appelée NUMÉRISATION.

Remarque : extrait du livre de Mr Dowek

la description de l'information avec deux symboles est plus simple à mettre en œuvre dans un ordinateur. Cela se réalise électriquement par un interrupteur ouvert ou fermé, un courant électrique présent ou absent, etc. Ce choix rend aussi les ordinateurs plus robustes : s'il y avait plus de deux valeurs, le système physique pourrait plus facilement les mélanger. Dès que les plages d'incertitude des valeurs se recouvriraient, par exemple avec la décharge de la batterie, des erreurs se créeraient. Cela minimise donc les erreurs d'interprétation en cas de perturbation du signal. C'est sans doute bien la qualité essentielle du code binaire que sa capacité à résister au bruit.

- La numération décimale utilise 10 symboles ou CHIFFRES : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9.
Exemple : $459 = 4 \times 10^2 + 5 \times 10^1 + 9 \times 10^0 = 400 + 50 + 9 = 459$
- La numération binaire utilise 2 symboles ou CHIFFRES : 0 et 1.
Exemple : $101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$

Les chiffres d'un nombre représentent la décomposition du nombre selon les puissances croissantes de la base de numération considérée. Dans la suite du cours, la base sera donnée en indice du nombre, représenté entre parenthèses, comme par exemple $(357)_{10}$, qui signifie que le nombre 357 est exprimé en base 10.

Le nombre binaire 101 sera notée $(101)_2$ ce qui signifie 101 en base binaire (ou base 2). Ce nombre binaire vaut 5 en base décimal :
 $(101)_2 = (5)_{10}$

Un octet (byte en anglais) est constitué de 8 bits. Un octet correspond à un atome d'information. Exemple : valeur d'un octet : 11010110

2) le poids d'un bit

Dans un nombre binaire, la valeur d'un bit, appelée poids, dépend de la position du bit dans le nombre binaire. A la manière des dizaines, des centaines et des milliers pour un nombre décimal, le poids d'un bit croît d'une puissance de deux en allant

de la droite vers la gauche comme le montre le tableau suivant :

| | | | | | | | | | |
|------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Nombre binaire | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Poids | 2^8 | 2^7 | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
| nombre decimal correspondant | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

3) conversion binaire décimale

Si l'on numérote les bits de 0 à 7, le bit numéro 0 est le bit de poids faible, et le bit numéro 7 est le bit de poids le plus fort. Si l'on considère un octet comme un nombre écrit en base 2, sa valeur numérique en base 10 vaut :

$$\sum_{n=0}^{n=7} b_n \cdot 2^n$$

avec b_n valeur du nième bit.

Exemple: $(11010110)_2 = 2^7 + 2^6 + 2^4 + 2^2 + 2^1 = (214)_{10}$

Vérifier votre résultat avec le [convertisseur binaire / décimal / hexadécimal](#). Vous pouvez également utiliser la calculatrice de windows en utilisant l'option **affichage, programmeur**.

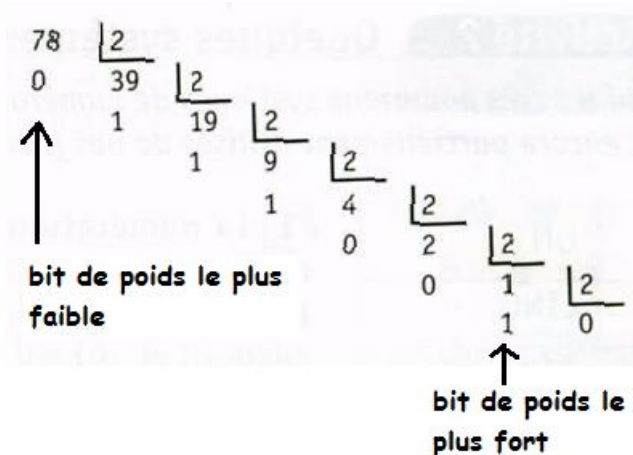
ex 1 : Convertir 11011 en base 10.

ex 2: Quel est le plus grand nombre que l'on peut écrire avec 8 bits ?

4) conversion décimale -> binaire

Pour la **conversion décimale** → **binaire**, on procède par division successives par 2 : les restes des divisions sont les chiffres binaires de la conversion. Le bit « de poids faible » (le plus à droite) est le premier reste obtenu.

Exemple : 78 (décimal) = 1001110 (binaire). Cette conversion est illustrée ci-dessous.



ex 3 : Convertir 13, puis 147 en binaire. Quel est l'avantage de ce système ? Quel en est l'inconvénient ?

Comment passer du codage usuel décimal au codage binaire ? Pour écrire le nombre décimal n en binaire, on utilise l'algorithme suivant :

1. Si $n \% 2 = 0$ (reste de la division entière), écrire 0, sinon écrire 1, à gauche de ce qui est déjà écrit.

2. Remplacer n par $n \div 2$ (division entière).

3. Si $n = 0$, on a fini, sinon on retourne au point 1.

Considérons le déroulement de cet algorithme sur $n = 11$

| | | | | | |
|----------|----|----|-----|------|---|
| n | 11 | 5 | 2 | 1 | 0 |
| $n \% 2$ | 1 | 1 | 0 | 1 | |
| | 1 | 11 | 011 | 1011 | |

5) les différentes unités binaires

Les différentes unités binaires sont :

- **1 bit** : unité élémentaire
 - **4 bits** : un **quartet** (nibble)
 - **8 bits** : un **octet** (byte)
 - **le mot** (machine) : c'est la quantité de bits standard manipulée par un microprocesseur (CPU). La taille du mot s'exprime en bits (ou en octets). Un microprocesseur est d'autant plus performant que ses mots sont longs, car les données qu'il traite à chaque cycle sont plus nombreuses. C'est pourquoi on classe les microprocesseurs par la taille de leur mot : 8, 16, 32, 64 bits (soit 1, 2, 4 ou 8 octets)
- Le mot correspond à une molécule d'information.

L'octet est utilisé pour mesurer, par exemple, la capacité mémoire d'un disque dur. On utilise également les unités suivantes :

[Unités officielles](#)

| Standard SI | | Unités binaires | |
|----------------|-----------|-----------------|----------|
| Unité | Valeur | Unité | Valeur |
| kilobit (kb) | 10^3 | kibibit (Kibit) | 2^{10} |
| mégabit (Mb) | 10^6 | mébibit (Mibit) | 2^{20} |
| gigabit (Gb) | 10^9 | gibibit (Gibit) | 2^{30} |
| térambit (Tb) | 10^{12} | tébibit (Tibit) | 2^{40} |
| pétabit (Pb) | 10^{15} | pébibit (Pibit) | 2^{50} |
| kiloctet (ko) | 10^3 | kibioctet (Kio) | 2^{10} |
| mégaoctet (Mo) | 10^6 | mébioctet (Mio) | 2^{20} |
| gigaoctet (Go) | 10^9 | gibioctet (Gio) | 2^{30} |
| téraoctet (To) | 10^{12} | tébioctet (Tio) | 2^{40} |
| pétaoctet (Po) | 10^{15} | pébioctet (Pio) | 2^{50} |

Usage traditionnel ≠ notations officielles : 1 ko = 1024 octets ...

Ex 4 : Calculer le rapport en % entre unités SI et unités binaires. Quelles unités les marchands ont-ils intérêt à utiliser ?

II) La numération hexadécimale

1) définition

La **numération hexadécimale** est la variante de la numération binaire : le codage binaire de l'information a un défaut d'utiliser des **nombre très longs** à écrire. Il faut 8 symboles pour un octet, 16 pour un mot manipulés par un microprocesseur 16 bits etc.. Les ordinateurs actuels utilisent des mémoires (centrale

ou de masse) de très grande taille : chaque case est définie par son emplacement, appelé aussi adresse. On a donc inventé un nouveau codage des nombres : la **numération en base 16**.

La numération hexadécimale emploie 16 « chiffres » ou symboles différents :

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

Le chiffre A (hexa) correspond donc au nombre décimal 10 : $A_{(hex)} = 10_{(10)}$

ex 5 : Remplir le tableau de correspondance suivant :

| | | | | | | | | | | | | | | | | |
|-------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hexadécimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| Décimal | | | | | | | | | | | | | | | | |

2) Conversion hexadécimal <-> décimal :

La valeur en base 10 du nombre $23A_{(hex)}$ exprimé en système hexadécimal vaut :

$$23A_{(hex)} = 2 \times 16^2 + 3 \times 16^1 + 10 \times 16^0$$

En généralisant à un nombre de n chiffres dans le système hexadécimal, la valeur correspondante dans le système décimale vaut :

$$\sum_{i=0}^{i=n-1} b_i \cdot 16^i$$

b_i valeur du nième chiffre.

ex 6 : Convertir en décimal les nombres hexadécimaux suivants :

$111_{(hex)}$; $C7B_{(hex)}$; $5EOB_{(hex)}$

Représenter ces nombres dans le système binaire.

Conversion d'un nombre décimale en hexadécimal : même principe que la conversion décimale binaire ; on divise le nombre décimal par 16 au lieu de le diviser par 2.

ex 7 : Convertir en hexadécimal les nombres décimaux suivants :

$65_{(10)}$; $1258_{(10)}$; $5381_{(10)}$

ex 8 : Exprimer en hexadécimal puis en décimal la valeur max que peut prendre un mot pour un microprocesseur 16 bits, 32 bits, 64 bits.

Réponse :

$$2^{16} - 1 = (65\ 535)_{10} = (FFFF)_{16}$$

$$2^{32} - 1 = (4\ 294\ 967\ 295)_{10} = (FFFF\ FFFF)_{16}$$

$$2^{64} - 1 = (18\ 446\ 744\ 073\ 709\ 551\ 615)_{10} = (FFFF\ FFFF\ FFFF\ FFFF)_{16}$$

ex 9 : Programmer à l'aide d'un tableur la conversion d'un nombre binaire sur un octet en base 10.

Programme officiel

| savoirs | capacités | observations |
|---|--|---|
| Représentation binaire Un ordinateur est une machine qui manipule des valeurs numériques représentées sous forme binaire. | Manipuler à l'aide d'opérations élémentaires les trois unités de base : - particule élémentaire : valeur binaire stockée dans un bit ; - atome : regroupement de 8 valeurs binaires dans un octet ; - molécule : mot constitué à partir de plusieurs octets. | On met en évidence, sous forme de questionnement, la présence du numérique dans la vie personnelle et professionnelle, au travers d'exemples. |