Architecture des ordinateur

Cours 1 - Système de numération et arithmétique binaire

Halim Djerroud (hdd@ai.univ-paris8.fr)

LIASD - Université Paris 8

22 septembre 2021

Introduction

Pour manipuler, afficher ou transmettre des nombres en utilisant des circuits électroniques, il est nécessaire de représenter chaque symbole par un état différent du circuit.

- Il faut un circuit à dix états pour représenter les symboles de la base dix, de 0 à 9

Problème :

- Il est difficile et excessivement cher de fabriquer des circuits à dix états

Implication :

- Il faut chercher à utiliser un système de numération avec peu de symboles

Pourquoi on compte jusqu'à 10 ?

- Deux mains, dix doigts

Les symboles de la base 10

- Symboles de la base 10 $= \underbrace{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}_{\text{10 symboles}}$

Exemple :

0	10	.	90	100	1000
1	11	.	91	101	1001
2	12	.	92	102	1002
3	13	.	93	...	...
4	14		94	199	
5	15		95	200	
6	16		96	...	
7	17		97	999	
8	18		98		
9	19		99		

Conversion en base 10

Exemple :

$$(6813)_{10} = 6 \times 10^3 + 8 \times 10^2 + 1 \times 10^1 + 3 \times 10^0$$

Formule :

$$(nombre)_b = \sum_{i=0}^{n-1} a_i \times b^i$$

- La formule permet de convertir un nombre de n'importe quelle base en base dix

Base Binaire

La plus petite base en informatique le nombre binaire s'écrit : b101010 (le nombre commence par b)

Les symboles de la base binaire Base 2

- Symboles de la base binaire 2 = $\underbrace{0,1}_{2 \text{ symboles}}$

Exemple :

0	1001
1	1010
10	1011
11	1100
100	1101
101	1110
110	...
111	
1000	

Octet

- Un octet est formé de 8 bits
- Le bit le plus à droite est appelé bit du poids faible
- Le bit le plus à gauche est appelé bit du poids fort
- L'octet est représenté avec le signe Ø

poids fort $1_7 0_6 1_5 0_4 0_3 1_2 1_1 1_0$ poids faible

Conversion de la base binaire en base 10

Exemple :

$$(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

Formule :

$$(nombre)_b = \sum_{i=0}^{n-1} a_i \times b^i$$

$$(1011)_2 = (11)_{10}$$

Conversion de la base binaire en base 10

Exercice :

Exprimer en base décimale les nombres binaires suivants :

- $(1100)_2 = (?)_{10}$
- $(10101010)_2 = (?)_{10}$
- $(1010101)_2 = (?)_{10}$
- $(1111111)_2 = (?)_{10}$
- $(110011001100)_2 = (?)_{10}$

Base Octale

En informatique le nombre octal s'écrit : 04532 (le nombre commence par O (zero))

Les symboles de la base Octale ou Base 8

- Symboles de la base Octale (8) $= \underbrace{0, 1, 2, 3, 4, 5, 6, 7}_{8 \text{ symboles}}$

Exemple :

0	11
1	12
2	13
3	14
4	15
5	16
6	17
7	20
10	...

Conversion de la base Octale en base 10

Exemple :

$$(04752)_8 = 4 \times 8^3 + 7 \times 8^2 + 5 \times 8^1 + 2 \times 8^0$$

Formule :

$$(nombre)_b = \sum_{i=0}^{n-1} a_i \times b^i$$

$$(04752)_8 = (?)_{10}$$

Conversion de la base Octale en base 10

Exercice :

Exprimer en base décimale les nombres octaux suivants :

- $(02457)_8 = (?)_{10}$
- $(0421561)_8 = (?)_{10}$
- $(0101010)_8 = (?)_{10}$
- $(0441243)_8 = (?)_{10}$
- $(010011001100)_8 = (?)_{10}$

Base Hexadécimale

En informatique le nombre Hexadécimal s'écrit : 0x8A5F3 (le nombre commence par Ox (zero x))

Les symboles de la base Hexadécimale ou Base 16

- Symboles de la base Hexadécimale (16) $= \underbrace{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}_{16 \text{ symboles}}$

Exemple :

0	E
...	F
9	10
A	11
B	...
C	
D	

Conversion de la base Hexadécimale en base 10

Exemple :

$$(0x1D5B)_{16} = 1 \times 16^3 + 13 \times 16^2 + 5 \times 16^1 + 11 \times 16^0$$

Formule :

$$(nombre)_b = \sum_{i=0}^{n-1} a_i \times b^i$$

$$(0x1D5B)_{16} = (?)_{10}$$

Conversion de la base Hexadécimale en base 10

Exercice :

Exprimer en base décimale les nombres Hexadécimaux suivants :

- $(0xABCD)_{16} = (?)_{10}$
- $(0x12345)_{16} = (?)_{10}$
- $(0x1BD4FC3)_{16} = (?)_{10}$
- $(0xCCCCCC)_{16} = (?)_{10}$
- $(0xDFA34CDFFF)_{16} = (?)_{10}$

Conversion entre bases

Nous allons utiliser une méthode simple et rapide pour résoudre les problèmes de conversion. Supposons que nous voulons convertir en base b un nombre N donnée en base a.
La méthode consiste à utiliser une succession de divisions en base a où $N_i(i = 1, 2, \cdots, n)$ représente le quotient de la division, b le diviseur et $R_j(j = 0, 2, \cdots, n)$ le reste de la division.

Conversion entre bases

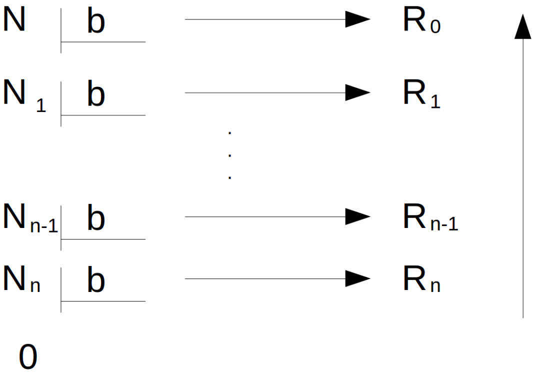


FIGURE – Conversion

Base Décimale à la base Binaire

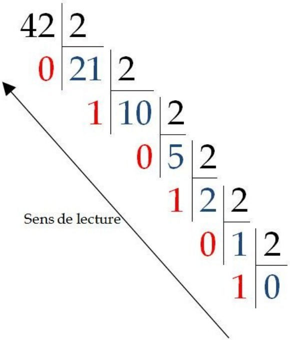


FIGURE – Conversion binaire

Conversion autres bases

- Conversion Octale à la base décimale
- Conversion Hexadécimale à la base décimale

Addition

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

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

Addition binaire

Exemple : Additionnez les deux nombres binaire :

$11101101 + 10111001 = ?$

		1	1	1	1			1		
		1	1	1	0	1	1	0	1	(237)
+		1	0	1	1	1	0	0	1	(185)
	1	1	0	1	0	0	1	1	0	(422)

Exercice

Exercice : Additionnez les deux nombres binaires suivants :

- $10101010 + 10101010$
- $11111111 + 10000000$
- $11111111 + 00111111$
- $01111111 + 01111111$

Soustraction binaire

La soustraction en binaire est faite de la même manière qu'en base décimale. En additionnant un nombre positif à un nombre négatif. Le nombre dont la valeur absolue est plus petite est soustrait du nombre dont la valeur absolue est plus grande et le signe de plus grand nombre est affecté au résultat.

Soustraction binaire

Exemple : Soustraire les deux nombres binaire :

$11001010 + 10101001 = ?$

	1	1	0	0	1	0	1	0	(202)
-	1	0	1	0	1	0	0	1	(169)
	0	0	1	0	0	0	0	1	(33)

Problème des nombres négatifs

- Si l'ordinateur propose de mettre un bit pour distinguer les nombres positifs et négatifs alors le nombre Zéro 0 aura deux représentations -0 et $+0$
- Un ordinateur ne distingue pas entre une addition et une soustraction. Ainsi, le circuit utilisé pour l'opération d'addition peut être utilisé pour l'opération de soustraction grâce au complément à 2.

Complément à 2

Complément à 2

$$Complément2(N) = Complémentà1(N) + 1$$

Complément à 1

$$Complément1(N) = C(N)$$

Complément à 2

Exemple :

Trouvez le complément à 2 de : $N = 10101001$

C1 :

$$\begin{array}{cccccccccc} \sim & & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ \hline & & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}$$

C2 :

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

Exercice complément à 2

Exercice : Effectuez les opérations suivantes en binaire :

- $125 - 135$
- $0x542 - 25$
- $b10100101 + 0001111$
- $054 - 0154$

Nombres à virgule

Un nombre décimal est composé d'une partie entière et d'une partie fractionnaire après la virgule

$$(N)_B = a_{n-1}a_{n-2}...a_0, b_1b_2...b_m$$

- Exemple :

$$128,75 = 1x10^2 + 2x10^1 + 8x10^0 + 7x10^{-1} + 5x10^{-2}$$
$$(101,01)_2 = 1x2^2 + 1x2^0 + 1x2^{-2}$$
$$(AE,1F)_{16} = 10x16^1 + 14x16^0 + 1x16^{-1} + 15x16^{-2}$$

Conversion des nombres à virgule en base B

Exemple : Convertir $28,8625$

- $28 = (11100)_2$, $0.8625 = (?)_2$

0,8625 x 2	1,725	1 + 0,725
0,725 x 2	1,45	1 + 0,45
0,45 x 2	0,9	0 + 0,9
0,9 x 2	1,8	1 + 0,8
0,8 x 2	1,6	1 + 0,6
0,6 x 2	1,2	1 + 0,2
0,2 x 2	0,4	0 + 0,4
0,4 x 2	0,8	0 + 0,8

$28,8625$ peut être représenté par $(11100,1101\underline{1100}\ldots)_2$

Notation exponentielle

Notation exponentielle

$$N = \pm M \times B^{\pm e}$$

Exemple :

- 5×10^8
- 1.9845×10^{-4}
- -85×10^{-14}
- $9,2384 \times 10^4$

La norme IEEE 754

- Un format standardisé
- Format simple précision codé en 32 bits
  - Bit du signe (1 bit)
  - Exposant (8 bits)
  - Mantisse (23 bits)
- Format double précision en 64 bits
  - Bit du signe (1 bit)
  - Exposant (11 bits)
  - Mantisse (52 bits)
- Code ASCII

Notation exponentielle IEEE 754 (Format général)

Un nombre flottant est formé de trois éléments :

- la mantisse notée *m* (les *m* bits de poids faible, représentent la mantisse)
- L'exposant notée *e* (les *e* bits suivants représentent l'exposant décalé)
- Le signe (Le bit de poids fort, si ce bit est à 1, le nombre est négatif, et s'il est à 0, le nombre est positif.)

Notation exponentielle IEEE 754 (Format général)

Signe	Exposant décalé	Mantisse
1 bit	e bits	m bits

Décalage de l'exposant

- L'exposant peut être positif ou négatif.
- La notation en complément à 2 rendrai la comparaison entre les nombres difficile.
- Pour régler ce problème, l'exposant est décalé, afin de le stocker sous forme d'un nombre non signé.
- Le décalage est de $2^{e-1} - 1$ (e représente le nombre de bits de l'exposant)

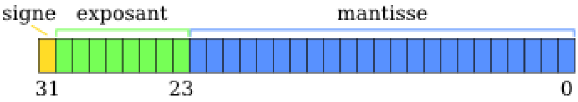
$$valeur = signe \times mantisse \times 2^{(exposant - décalage)}$$

IEEE 754 les exceptions

Type	Exposant décalé	Mantisse
Zéros	0	0
Nombres dénormalisés	0	différente de 0
Nombres normalisés	$1 à 2^e - 2$	quelconque
Infinis	$2^e - 1$	0
NaNs	$2^e - 1$	différente de 0

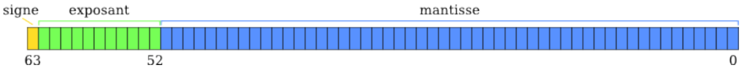
Format simple précision (32 bits)

- 32 bits : 1 bit de signe, 8 bits pour l'exposant et 23 pour la mantisse.
- L'exposant est donc décalé de $2^{8-1} - 1 = 127$
- L'exposant d'un nombre normalisé va donc de -126 à +127. L'exposant -127 (qui est décalé vers la valeur 0) est réservé pour zéro et les nombres dénormalisés, tandis que l'exposant 128 (décalé vers 255) est réservé pour coder les infinis et les NaN.



Format simple précision (64 bits)

- 64 bits : 1 bit de signe, 11 bits pour l'exposant et 52 pour la mantisse.
- L'exposant est donc décalé de $2^{11-1} - 1 = 1023$
- Pour les nombres normalisés, le décalage de l'exposant est $+1023$. Pour les nombres dénormalisés, l'exposant est -1022 (l'exposant minimum pour un nombre normalisé). Ce n'est pas -1023 car les nombres normalisés ont un 1 avant la virgule, et les nombres dénormalisés n'en ont pas. Comme précédemment, zéro et l'infini sont signés.



Les différents codages

- Code Gray
- Code BCD
- Code 7 segments
- Code ASCII

Code ASCII

```
    30  40  50  60  70  80  90  100  110  120
  ----------------------------------------
0:       (   2   <   F   P   Z   d    n    x
1:       )   3   =   G   Q   [   e    o    y
2:       *   4   >   H   R   \   f    p    z
3:   !   +   5   ?   I   S   ]   g    q    {
4:   "   ,   6   @   J   T   ^   h    r    |
5:   #   -   7   A   K   U   _   i    s    }
6:   $   .   8   B   L   V   `   j    t    ~
7:   %   /   9   C   M   W   a   k    u   DEL
8:   &   0   :   D   N   X   b   l    v
9:   '   1   ;   E   O   Y   c   m    w
```