MATHÉMATIQUES ET INFORMATIQUE
Sciences
Université de Paris

NUMÉRATION LOGIQUE

C2 : Représentations des nombres entiers et conversions

Nicole VINCENT

Numération moderne

- Un entier est représenté par une suite de symboles ou chiffres, dans un ensemble à b éléments
- La **base b** indique le nombre de symboles disponibles

  $\{0, 1, 2, \dots, b - 1\}$
- La **position** des chiffres dans la suite a un sens
- La suite de chiffres en base b s'**interprète** dans le système décimal (base 10)

$$\boxed{(s_3\ s_2\ s_1\ s_0)_b = (s_3\ b^3 + s_2\ b^2 + s_1\ b^1 + s_0\ b^0)_{10}}$$

$(24)_{10} = 2 \,.\, 10 + 4$

$(24)_5 = (2 \,.\, 5 + 4)_{10} = (14)_{10}$

le poids b^i est affecté au symbole à la $i^{ème}$ position : s_i .

Système pondéré

Principes fondamentaux

$(324)_5 = (3 \,.\, 25 + 2 \,.\, 5 + 4)_{10} = (89)_{10}$

- **Principe de position**

  La signification d'un symbole dépend de son poids, c'est-à-dire de sa place dans la suite des symboles
- **Principe du zéro**

  Le zéro indique une position où il n'y a pas d'élément.

Chaque symbole porte le nom de **digit**

10, signifie 0 unité et 1 dizaine en base 10,

En base b, 10 représente le nombre b

Principales bases utilisées

- **Base décimale** b = 10 avec les chiffres : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.

  C'est la base usuelle, utilisée dans le système métrique
- **Base binaire** b = 2 alphabet : {0, 1}.

  Utilisée en informatique, en électricité, base de la logique booléenne ou de l'algèbre de Boole, . .

  Base "minimale"
- **Base octale** b = 8 alphabet : {0, 1, 2, 3, 4, 5, 6, 7}.

  Utilisée par exemple pour les droits d'accès aux fichiers sous Linux, chmod 755 au lieu de `rwxr-xr-x`
- **Base hexadécimale** b = 16 alphabet : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}.

  Très utilisée en micro-informatique (langage assembleur), représentation compacte

Autres bases utilisées

- **Base 5** avec les chiffres : {0, 1, 2, 3, 4}.

  Compter avec ses deux mains : les doigts d'une main expriment les unités ceux de l'autre expriment des paquets de cinq.
- **Base 12** établie sur les signes du zodiaque.

  Intérêt de la divisibilité par 2, 3, 4 et 6.

  Utilisée dans le commerce (oeufs, huîtres, . . .) et pour les heures dans une journée (RV à 2h, cours de 11h à 12h, . . .).
- **Base 20** construite à partir des doigts et des orteils.

  Apparaît dans "quatre-vingts", Hôpital des Quinze Vingts,. . .
- **Base 60**, il y a 360° dans le cercle ; 60 minutes dans une heure,

  60 secondes dans une minute

Base 2

- Alphabet = {0, 1}, i.e. des bits.
- digits/bits de positions particulières

  MSB, most significant bit ou **bit de poids fort**

  LSB, least significant bit ou **bit de poids faible**
- Représentation polynomiale d'un nombre en base 2 :

$$(100110)_2 = \underbrace{1}_{\text{MSB}}\ 2^5 + 0\ \ 2^4 + 0\ \ 2^3 + 1\ \ 2^2 + 1\ \ 2^1 + \underbrace{0}_{\text{LSB}}\ 2^0$$

$$(100110)_2 = (32 + 0 + 0 + 4 + 2 + 0) = (38)_{10}$$

Conversion base 2 → base 10

Calculs en base 2

Additionner deux nombres en base 2

$1 + 1 = (10)_2 = (2)_{10}$.

(Si la somme dépasse $(2)_{10}$ on obtient une retenue!

Exemple $(1100110)_2 + (101110)_2$

	1	1	0	0	1	1	0
+		1	0	1	1	1	0
retenue 1	1	0	1	1	1		
	0	0	1	0	1	0	0

Résultat : $(10010100)_2$

Multiplier deux nombres en base 2

multiplication par $(10)_2 = (2)_{10}$ → un décalage ("shift") vers la gauche (placer 0 à droite)

Exemple : $(1010)_2 .(10)_2 = (10100)_2$

7

Calculs en base b

- Même principe en base b
- Dès que la somme dépasse b, on obtient une retenue

Exemple : $(34)_5 + (11)_5$

$(34)_5$		3	4
$(11)_5$		1	1
retenue	1	1	
		0	0
	1	0	0

$(34)_5 + (11)_5 = (100)_5 = (25)_{10}$

8

Théorème fondamental des bases de la numération

Soit b un entier naturel avec b >= 2.

- Pour tout nombre entier n ∈ N*, il existe un entier k >= 0 tel que $b^{k-1} \leq n < b^k$;
- Tout nombre entier n ∈ N* se décompose de manière unique sous forme d'une fonction polynomiale en b de degré k - 1 et avec des coefficients s_i ∈ {0, 1, … , b - 1}

$$n = \sum_{i=0}^{k-1} s_i b^i$$

k	k-1	…	4	3	2	1	0

9

Exemples

- Base octale, symboles {0, 1, 2, 3, 4, 5, 6, 7}

  Nombre $(n)_8$ = 5 4 6 5

  Exposants 3 2 1 0

  Ecriture en base 10 : 5 8^3 + 4 8^2 + 6 8^1 + 5 8^0 = $(2869)_{10}$

- Base hexadécimale,

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

  Le nombre $(B35)_{16}$ s' écrit en base 10 :

  11 16^2 + 3 16^1 + 5 16^0 = $(2869)_{10}$

10

Conversion entre bases

- Pour passer d'un nombre en base b à un nombre en base 10, on utilise l'écriture polynomiale
- Pour passer d'un nombre en base 10 à un nombre en base b, deux méthodes :
  - Méthode par soustraction
  - Méthode par multiplication

11

Conversion 10 → b : Méthode par soustraction

Soit $(n)_{10}$ ∈ N* à convertir en base b.

∃ k ∈ N tel que $b^{k-1} \leq n < b^k$ on aura k positions $(s_{k-1} \ldots s_1 s_0)_b$

On détermine d'abord les digits de **plus fort poids** puis les digits de **poids faible**.

- s_{k-1} est le nombre de fois que b^{k-1} est dans $n_1 = n$
- s_{k-2} est le nombre de fois que b^{k-2} est dans $n_2 = n_1 - s_{k-1}b^{k-1}$
- s_{k-3} est le nombre de fois que b^{k-3} est dans $n_3 = n_2 - s_{k-2}b^{k-2}$

  …
- s_1 est le nombre de fois que b^1 est dans $n_{k-1} = n_{k-2} - s_2b^2$
- $s_0 = n_k = n_{k-1} - s_1b^1 \in \{0, 1, \ldots, b - 1\}$ est le reste

12

Exemple – méthode par soustraction

1 8 64 512

n = 172 : convertir en base b = 8

$8^2 \leq 172 < 8^3$, on a besoin de k = 3 positions → $s_2s_1s_0$

- Dans $n_1 = 172$, combien de fois $8^2 = 64$? : 2 fois $n_2 = 172 - (2 . 64) = 44$

- Dans 44, combien de fois y a-t-il 8 ? 5 fois $n_3 = 44 - 5 . 8 = 4$

- 4 < 8 → $s_0 = 4$ est le reste

$S_2=2$ $S_1=5$ $S_0=4$

Finalement $(172)_{10} = (254)_8$

Exemple

1 2 4 8 16 32 64 128 256

Soit n = 173 à convertir en base b = 2.

Comme $2^7 \leq 173 < 2^8$, on a besoin de 8 bits

- Dans $n_1 = 173$, combien y a-t-il $2^7 = 128$?	(MSB)1
- $n_2 = 173 - 128 = 45$; dans 45, combien de $2^6 = 64$?	0
- $n_3 = 45$; dans 45, combien de $2^5 = 32$?	1
- $n_4 = 45 - 32 = 13$; dans 13, combien de $2^4 = 16$?	0
- $n_5 = 13$; dans 13, combien de 8 ?	1
- $n_6 = 13 - 8 = 5$; dans 5, combien de 4 ?	1
- $n_7 = 5 - 4 = 1$; dans 1, combien de 2 ?	0
- il reste $n_8 = 1 < 2$, la conversion est finie	(LSB)1

D'où $(173)_{10} = (10101101)_2$

Conversion 10 → b : Méthode par division

Soit $(n)_{10} \in N^*$ à convertir en base b : $(n)_{10} = (s_{k-1} \ldots s_1s_0)_b$

On utilise la division euclidienne, (division entière).

- on effectue la division entière de n par b :
  $n = d_1 . b + r_1$, on garde $s_0 = r_1$
- on effectue la division entière de d_1 par b :
  $d_1 = d_2 . b + r_2$, on garde $s_1 = r_2$

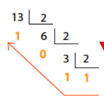
...

- on effectue la division entière de d_{k-2} par b :
  $d_{k-2} = d_{k-1} . b + r_{k-1}$, on garde $s_{k-2} = r_{k-1}$
- quand $d_{k-1} \in \{0 , 1 , \ldots , b - 1\}$, $s_{k-1} = d_{k-1}$ est le reste

On détermine d'abord les digits de faible poids et ensuite les digits de poids fort

Exemple

- Soit n = 13 à convertir en base b = 2



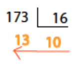
$13 = 6\ 2 + 1$
$6 = 3\ 2 + 0$
$3 = 1\ 2 + 1$

$13 = 6\ 2 + 1 = (.2^2 + 2)\ 2 + 1 = .2^3+2^2+0.2^1+1$
$6 = (.2+1)\ 2 + 0 = .2^2 + 2$
$3 = 1\ 2 + 1$

Le résultat est donc $(13)_{10} = (1101)_2$

Exemple

n = 173 à convertir en base b = 16



173 =10 . 16 + 13

$10=A_{16}$
$13=D_{16}$

Le résultat est donc $(173)_{10} = (AD)_{16}$

Nombres multiples de la base b

- On considère la forme polynomiale d'un entier écrit en base b

$$n_b = (s_ks_{k-1} \ldots s_1s_0)_b$$
$$= s_kb^k + s_{k-1}b^{k-1} + s_{k-2}b^{k-2} \ldots + s_1b^1 + s_0b^0$$

- Remarques

- un multiple de b se termine par 0, $s_0 = 0$;
  il s'écrit $n = b . (s_kb^{k-1} + s_{k-1}b^{k-2} + s_{k-2}b^{k-3} + \quad + s_1)$
- un multiple de b^2 se termine en 00, $s_0 = s_1 = 0$
- un multiple de b^3 se termine en 000
- …

1 ; 2 ; 4 ; 8 ; 16 ; 32 ; 64 ; 128 ; 256 ; 512 ; 1024
0 ; 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 9 ; 10

Représentation binaire des entiers naturels

- Avec n bits on peut représenter 2^n valeurs, soit tous les entiers de 0 à $2^n - 1$
  Le plus grand entier représentable sur n bits s'écrit : $\underbrace{111 \quad 11111}_{\text{n fois}}$ vaut $2^n - 1$
- L'entier $2^n - 1$ est toujours un nombre impair
- Ecriture en binaire des nombres 255, 257, 260, 510, 1024, 1019
  - position par rapport aux puissances de 2 les plus proches
    $255 = 2^8 - 1$; $257 = 2^8 + 1$; $260 = 2^8 + 4$
    $510 = (2^9 - 1) - 1$; $1024 = 2^{10}$; $1019 = (2^{10} - 1) - 4$
  - écriture en base 2
    $(255)_{10} = (11111111)_2$; $(257)_{10} = (100000001)_2$
    $(260)_{10} = (100000100)_2$; $(510)_{10} = (111111110)_2$
    $(1024)_{10} = (10000000000)_2$; $(1019)_{10} = (1111111011)_2$

Conversion Binaire-Octal

- En informatique les bases binaire, octale et hexadécimale sont les plus utilisées
- Toutes ces bases étant des puissances de deux, 2^1, 2^3 et 2^4,
  il y a des conversions particulièrement simples
- Pour écrire les 8 symboles de la base octale on a besoin de trois bits
  $(0)_8 = (000)_2$; $(1)_8 = (001)_2$; … ; $(6)_8 = (110)_2$; $(7)_8 = (111)_2$
  - Pour passer de l'octal en binaire :
  on remplace chaque chiffre octal par les trois bits correspondants
  - Pour passer du binaire en octal :
  on parcourt le nombre binaire de la droite vers la gauche en regroupant les chiffres binaires par paquets de 3 (en complétant éventuellement par des zéros).
  On remplace chaque paquet de 3 par le chiffre octal.

Conversion Binaire-Hexadécimal

Pour écrire les 16 symboles de la base hexadécimale on a besoin de quatre bits
$(0)_{16} = (0000)_2$; $(1)_{16} = (0001)_2$; $(2)_{16} = (0010)_2$; $(3)_{16} = (0011)_2$;
$(4)_{16} = (0100)_2$; $(5)_{16} = (0101)_2$; $(6)_{16} = (0110)_2$; $(7)_{16} = (0111)_2$;
$(8)_{16} = (1000)_2$; $(9)_{16} = (1001)_2$; $(A)_{16} = (1010)_2$; $(B)_{16} = (1011)_2$;
$(C)_{16} = (1100)_2$; $(D)_{16} = (1101)_2$; $(E)_{16} = (1110)_2$; $(F)_{16} = (1111)_2$

- Pour passer de l'hexadécimal en binaire :
on remplace chaque chiffre hexadécimal par les quatre bits correspondants.
- Pour passer du binaire en hexadécimal :
on parcourt le nombre binaire de la droite vers la gauche en regroupant les chiffres binaires par paquets de 4 (en complétant éventuellement par des zéros).
On remplace chaque paquet de 4 par le chiffre hexadécimal.

Exemples

- Convertir $(A01)_{16}$ en binaire :
  on sait que $A_{16} = (1010)_2$; $0_{16} = (0000)_2$ et $1_{16} = (0001)_2$
  donc $(A01)_{16} = (\underbrace{1010}_{A}\underbrace{0000}_{0}\underbrace{0001}_{1})_2$
- Convertir $(10110)_2$ en base 16 :
  le regroupement par paquets de quatre donne 0001 0110 ;
  on associe à chaque paquet le chiffre hexadécimal :
  donc $(10110)_2 = (16)_{16} = (22)_{10}$

Autres bases

- $8 = 2^3$ $16 = 2^4$
  deux bases où l'une est une puissance de l'autre
- Passage de base 9 à 3
  $(722)_9 = (7\ 2\ 2)_9 = (210202)_3$ $7 = 2\ 3 + 1$
- Passage de base 4 à 16
  $(2230002)_4 = (2\ 23\ 00\ 02)_4 = (2B02)_{16}$ $23 = 2 . 4 + 3 = 11$

Exemple de l'adresse MAC

- L'adresse MAC d'une machine, est un identifiant unique au monde qui identifie cette machine (ordinateur, smartphone, imprimante, …) parmi toutes les autres.
- Elle est constituée de 6 octets en hexadécimal,
  exemple 5E:FF:56:A2:AF:15.
- Le nombre de machines que l'on peut coder avec ce système est donc de $(2^8)^6 \approx$ 200 000 milliards, on devrait pouvoir s'en sortir…
- Chaque groupe de 2 chiffres "hexadécimaux" se code en 8 chiffres en base 2 :
  $(5E)_{16} \rightarrow (0101\ 1110)2$
  $(FF)_{16} \rightarrow (1111\ 1111)2$
  ….

On a donc bien 6 octets :(01011110)(11111111) …