Codage des nombres entiers relatifs

M. Combacau
combacau@laas.fr

UNIVERSITÉ TOULOUSE III PAUL SABATIER

NORTHEASTERN UNIVERSITY

12 novembre 2024

Objectif

Savoir coder un nombre entier positif dans une base quelconque
Détails du cas de la base 2

Introduction

- Un nombre $\longleftrightarrow$ valeur
- Représenter la valeur d'un nombre : codage !
- Que signifie **31** ? $3 \times 10 + 1$? pourquoi ?
- Existence d'un ensemble de conventions **admises par tous**
  - 10 chiffres sont utilisés $\{0, 1 \ldots, 9\}$
  - le code d'un nombre est une succession ordonnée de ces chiffres
  - numération de position :
    la $valeur(A)$ représentée par $a_{n-1} \ldots a_1 a_0$ se calcule par

$$valeur(A) = a_{n-1}.10^{n-1} + a_{n-2}.10^{n-2} + \ldots + a_1.10^1 + a_0.10^0$$

$\Rightarrow$ 31 est "codé en base 10"

Codage d'un nombre entier en base b (b chiffres)

Simple généralisation du codage en base 10 :

En base b, $a_{n-1} \ldots a_1 a_0$ code la valeur $a_{n-1} b^{n-1} + \ldots + a_0 . b^0$

Quelques bases classiques

- base 10 (décimal) $\rightarrow$ comptez vos doigts...
- base 2 (binaire) $\rightarrow$ proche des valeurs de variables booléennes
- base 8 (octal) $\rightarrow$? comptez les espaces entre vos doigts...
- base 16 (hexadécimal) $\rightarrow$ binaire compact (chiffres :0..9ABCDEF)

La base devrait être précisée : 31_{10} en effet : $31_{10} \neq 31_8 \neq 31_{16}$

Quelques exemples

Le nombre 143 se représente par :

$$
\begin{aligned}
10001111_2 &= 2^7 + 2^3 + 2^2 + 2^1 + 2^0 \\
217_8 &= 2 \times 8^2 + 1 \times 8^1 + 7 \times 8^0 \\
8F_{16} &= 8 \times 16 + F \quad \text{car} \quad (F_8 = 15_{10})
\end{aligned}
$$

Base 2 : principe et intérêt

- **Seul intérêt** : la valeur d'un **bit** (binary digit) peut être représentée par la tension en un point d'un circuit électronique
- L'intervalle de valeurs codables sur n bits

$$valeur(a_{n-1} \dots a_1 a_0) \in [0, 2^n - 1]$$

- La valeur codée par n bits

$$valeur(a_{n-1} \dots a_1 a_0) = a_{n-1}.2^{n-1} + \dots + a_1.2^1 + a_0.2^0$$

- Décodage : calcul ci-dessus dans la base d'arrivée (généralement 10)
- Codage : changement de base (généralement 10 $\rightarrow$ 2)

Algorithme de codage(1)

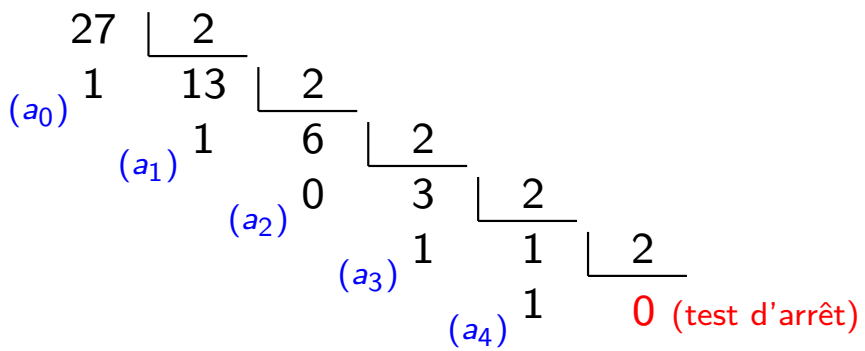
Utilisation du schéma de Horner

$$
\begin{aligned}
& a_{n-1}.2^{n-1} + a_{n-2}.2^{n-2} + \ldots + a_1.2^1 + a_0 \\
= \quad & 2 \times (a_{n-1}.2^{n-2} + a_{n-2}.2^{n-3} + \ldots + a_1.) + a_0 \\
= \quad & 2 \times (2 \times (a_{n-1}.2^{n-3} + a_{n-2}.2^{n-4} + \ldots + a_2) + a_1) + a_0 \\
= \quad & \ldots \\
= \quad & 2 \times (2 \times (\ldots (2 \times (a_{n-1}) + a_{n-2}) + \ldots + a_2) + a_1) + a_0
\end{aligned}
$$

$\rightarrow$ calcul de a_0..... a_{n-1} par divisions successives par 2 (reste)

Algorithme de codage(2)

Exemple : soit à coder 27_{10} en base 2



Les divisions successives donnent le bits de poids faible en commençant par a_0.
Dans cet exemple, tous les bits de rang supérieur à 4 (a_5, a_6, a_7) valent 0.
D'où le code [00011011] pour le nombre 27

Algorithme de codage(3)

Autre exemple : soit à coder 143_{10} en base 2

opération	reste	a_i
$143/2 = 71$	1	(a_0)
$71/2 = 35$	1	(a_1)
$35/2 = 17$	1	(a_2)
$17/2 = 8$	1	(a_3)
$8/2 = 4$	0	(a_4)
$4/2 = 2$	0	(a_5)
$2/2 = 1$	0	(a_6)
$1/2 = 0$	1	(a_7)

Test d'arrêt : quotient $= 0$