#2 : Representação de Números Inteiros

Computer Architecture 2019/2020

João Soares & Ricardo Rocha

Computer Science Department, Faculty of Sciences, University of Porto

Números Inteiros sem Sinal

A **base** de um sistema de numeração define o conjunto de símbolos válidos para representar números inteiros sem sinal nessa base:

- Base 2 utiliza 2 símbolos (0, 1)
- Base 10 utiliza 10 símbolos (0, ..., 9)
- Base 16 utiliza 16 símbolos (0, ..., 9, A, ..., F)

Um número pode ter representações diferentes conforme a base utilizada, mas o seu valor é sempre calculado da mesma forma:

- $v = a_{n-1} \times b^{n-1} + a_{n-2} \times b^{n-2} + \ldots + a_1 \times b^1 + a_0 \times b^0$, em que $a_{n-1}\ a_{n-2} \ldots a_1\ a_0$ são a sequência de símbolos da representação de v na base b
- Exemplo: 137_{16} tem o valor de $1 \times 16^2 + 3 \times 16^1 + 7 \times 16^0 = 311$

Conversão de Base

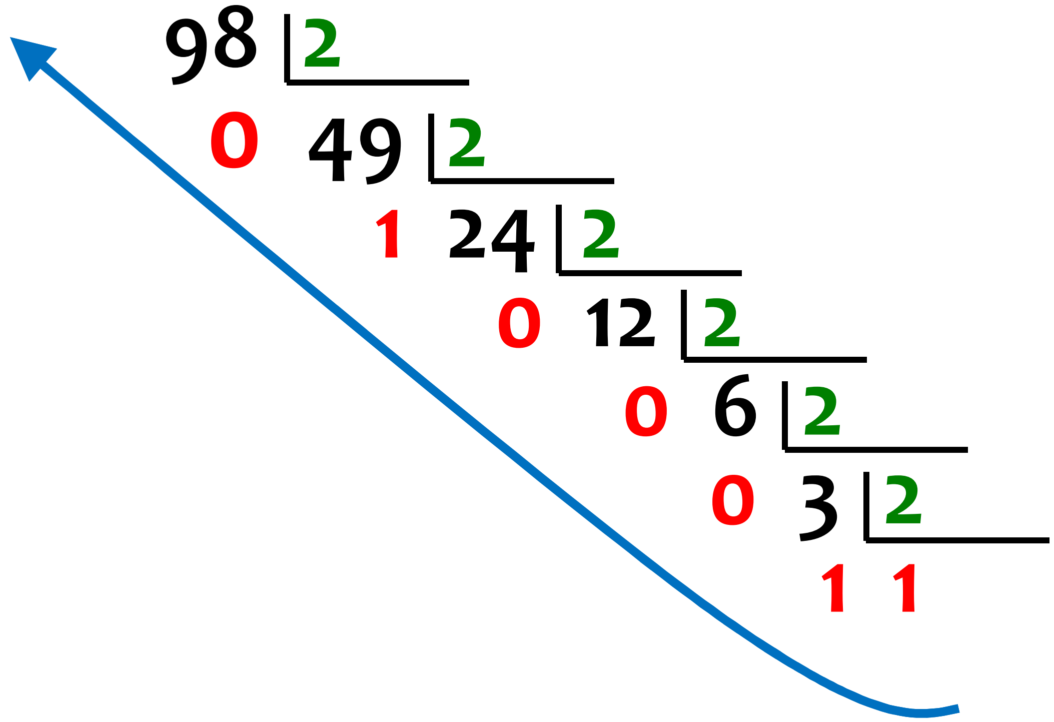
Partindo duma representação decimal (base 10), a conversão de base é conseguida pela **realização de divisões inteiras sucessivas pela nova base até se atingir um quociente inferior à nova base**. A representação do valor na nova base é a concatenação dos restos obtidos no processo.

$98_{10} = 1x8^2 + 4x8^1 + 2x8^0 = 142_8$

98 | 8
2   12 | 8
    4   1

Conversão de Base

$$98_{10} = 1\text{x}2^6 + 1\text{x}2^5 + 0\text{x}2^4 + 0\text{x}2^3 + 0\text{x}2^2 + 1\text{x}2^1 + 0\text{x}2^0 = 1100010_2$$



Conversão de Base

Alternativamente podemos **utilizar diretamente a representação em potências da base**:

$620_{10} = ? \times 16^3 + ? \times 16^2 + ? \times 16^1 + ? \times 16^0 = ????_{16}$

$16^3 = 4096 > 620$

$16^2 = 256 \rightarrow 2 \times 256 <= 620\ (3 \times 256 > 620) \rightarrow 620 - (2 \times 256) = 108$

$16^1 = 16 \rightarrow 6 \times 16 <= 108\ (7 \times 16 > 108) \rightarrow 108 - (6 \times 16) = 12$

$16^0 = 1 \rightarrow 12 \times 1 <= 12$

$620_{10} = 2 \times 16^2 + 6 \times 16^1 + 12 \times 16^0 = 26C_{16}$

Conversão de Base

A conversão entre bases que são **potência uma da outra** é ainda mais direta pois o **expoente da potência** determina o número de **dígitos necessários para representar o valor na nova base**.

Por exemplo, conversões entre a base 2 e a base 16. Como **$16 = 2^4$**, cada dígito em base 16 pode ser representado com 4 dígitos em base 2. Da mesma forma, 4 dígitos em base 2 podem ser representados com 1 dígito em base 16.

$7_{16} = 0111_2$

$F_{16} = 1111_2$

$F7_{16} = 1111\ 0111_2$

Aritmética (Adição)

Independentemente da base usada, o **algoritmo usado para operações aritméticas é o mesmo**.

$$
\begin{array}{r}
\color{red}{c3\ c2\ c1\ \ \ } \\
y2\ y1\ y0_b \\
+\ z2\ z1\ z0_b \\
\hline
r3\ r2\ r1\ r0_b
\end{array}
$$

$c1 = (y0 + z0) / b$
$c2 = (c1 + y1 + z1) / b$
$c3 = (c2 + y2 + z2) / b$

$r0 = (y0 + z0) \bmod b$
$r1 = (c1 + y1 + z1) \bmod b$
$r2 = (c2 + y2 + z2) \bmod b$
$r3 = c3$

Números Inteiros com Sinal

Na base decimal, os números inteiros negativos são representados pelo sinal '-' antes do número. Em binário (base 2), usamos o **bit mais significativo (o da esquerda) para representar o sinal**.

- Bit 0 representa um número positivo e bit 1 representa um número negativo
- Para usar bit de sinal é necessário fixar o número de bits usado na representação

Exemplo com 4 bits:

$$5_{10} = \mathbf{0}101_2$$

$$-5_{10} = \mathbf{1}101_2$$

Complemento para 1

Números positivos são representados de igual forma. Números negativos são representados como o resultado da subtração da codificação positiva ao valor '111 … 111'. Ou seja, os **números negativos podem ser vistos como a negação do positivo correspondente por inversão dos bits (os 0's passam a 1's e vice-versa)**.

Exemplo com 4 bits:

$$\mathbf{5_{10} = 0101_2}$$

$$\mathbf{-5_{10} = 1111_2 - 0101_2 = 1010_{2\ (compl\ p/1)}}$$

Complemento para 2

Números positivos são representados de igual forma. **Números negativos são representados como o complemento para 1 + 1_2.**

Em complemento para 2, a sequência de bits $a_{n-1}\ a_{n-2} \ldots a_1\ a_0$ na base **b** representa o valor $v = -a_{n-1} \times b^{n-1} + a_{n-2} \times b^{n-2} + \ldots + a_1 \times b^1 + a_0 \times b^0$.

Exemplo com 4 bits:

$5_{10} = 0101_2$

$-5_{10} = 1010_2 + 1_2 = 1011_{2\ (\text{compl p/ 2})} = -1\text{X}2^3 + 1\text{X}2^1 + 1\text{X}2^0 = -8+2+1$

$-(-5_{10}) = 0100_2 + 1_2 = 0101_{2\ (\text{compl p/ 2})}$

Resumo

Decimal	Binário sem sinal	Binário com sinal	Complem. para 1	Complem. para 2
3	011	011	011	011
2	010	010	010	010
1	001	001	001	001
0	000	000 / 100	000 / 111	000
-1	–	101	110	111
-2	–	110	101	110
-3	–	111	100	101
-4	–	–	–	100