INTRODUÇÃO A CONCEITOS DE COMPUTAÇÃO

Sistemas de numeração e conversão de bases

SUMÁRIO

- Sistemas de numeração em computação
- Conversão entre bases
- Codificação Binária - Números

Sistemas de numeração em computação

Sistemas de numeração:

- **Representação estruturada e dentro de um contexto de uma coleção de números através de numerais.**
- **Representações para o número 10:**

十 - Chinês
X - Romano
 - Egípcio

Sistemas de numeração em computação

Sistemas numéricos utilizados na Computação:

- → **binário (base 2),**
- → **octal (base 8),**
- → **decimal (base 10),**
- → **hexadecimal (base 16).**

Base							
10	**8**	**16**	**2**	**10**	**8**	**16**	**2**
0	0	0	0	8	10	8	1000
1	1	1	1	9	11	9	1001
2	2	2	10	10	12	A	1010
3	3	3	11	11	13	B	1011
4	4	4	100	12	14	C	1100
5	5	5	101	13	15	D	1101
6	6	6	110	14	16	E	1110
7	7	7	111	15	17	F	1111

Conversão de Base

Convertendo para decimal:

$$(a_n a_{n-1} \ldots a_0, a_{-1} a_{-2} \ldots a_{-m})_x =$$
$$= a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \ldots$$
$$+ a_0 \cdot x^0 + a_{-1} \cdot x^{-1} + a_{-2} \cdot x^{-2} + \ldots + a_{-m} \cdot x^{-m}$$

Exemplos:

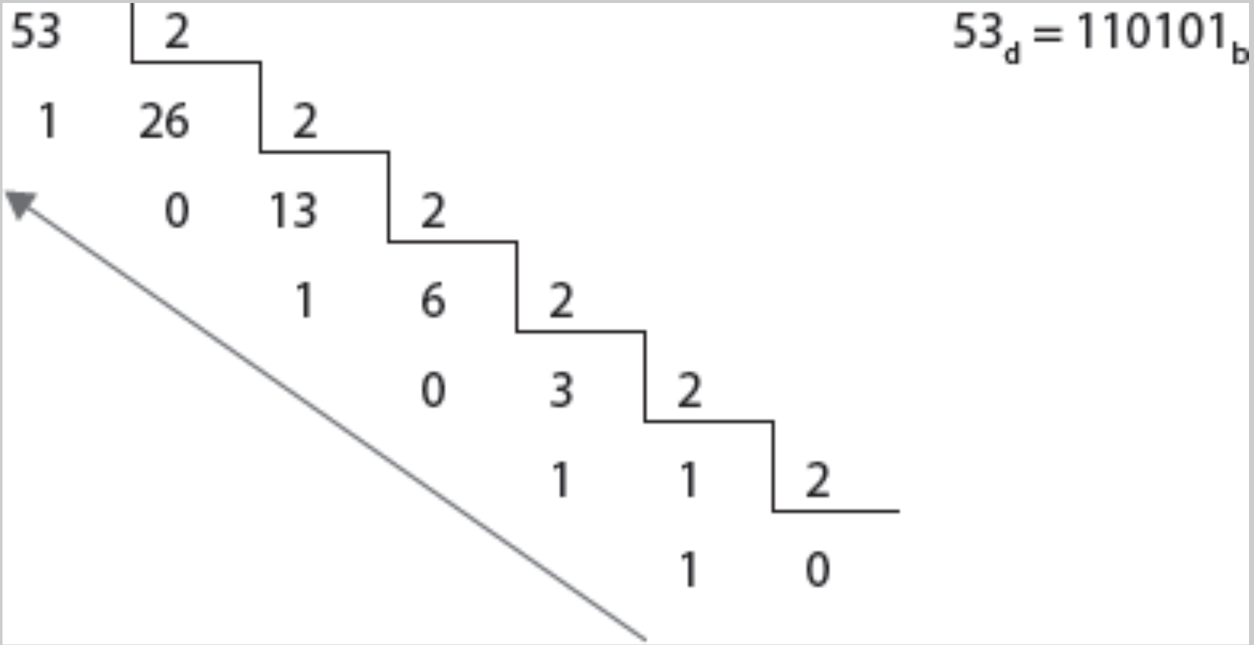
$$(1011)_b = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$
$$= 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 11$$

$$(230,4)_8 = 2 \cdot 8^2 + 3 \cdot 8^1 + 0 \cdot 8^0 + 4 \cdot 8^{-1}$$
$$= 2 \cdot 64 + 3 \cdot 8 + 0 \cdot 1 + 4 \cdot \frac{1}{8} = 152,5$$

Conversão de Base

Convertendo da base decimal para base x:

Parte inteira: divide-se sucessivamente o valor pela base x.



Conversão de Base

Convertendo da base decimal para base x:

Parte fracionária: **multiplicações sucessivas do número fracionário a ser convertido pela base.**

$0{,}3 \times 2 = 0{,}6$

$0{,}6 \times 2 = 1{,}2$

$0{,}2 \times 2 = 0{,}4$

$0{,}4 \times 2 = 0{,}8$

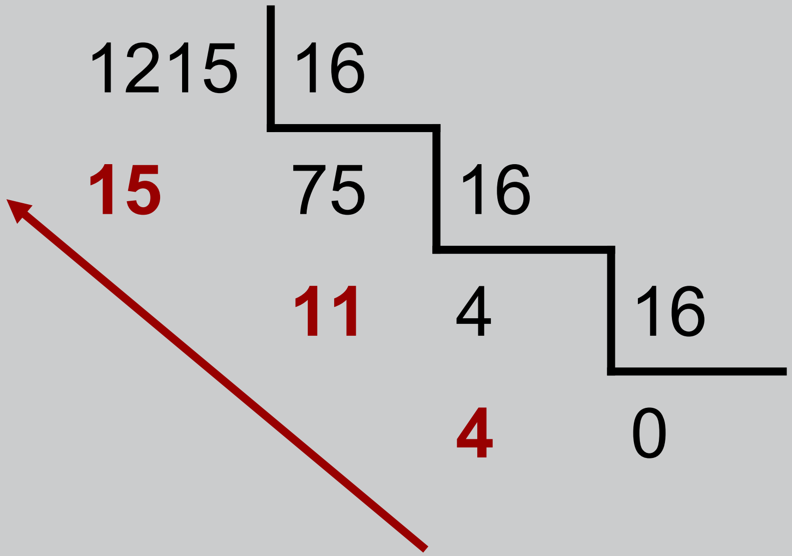
$0{,}8 \times 2 = 1{,}6$

$0{,}6 \times 2 = 1{,}2$

$0{,}3_d = 0{,}010011001\ldots_b$

Conversão de Base

Exemplos: $(1215{,}53)_d$ para base 16



Parte Inteira: $(1215)d = (4BF)_{16}$

Conversão de Base

Exemplos: **$(1215{,}53)_d$ para base 16** **$(1215{,}52)=(4BF{,}87A)_{16}$**

0,53x16= **8**,48

0,48x16= **7**,68

0,68X16= **10**,88

Parte Fracionária: $(0{,}53)d = (0{,}87A)_{16}$

Codificação Binárias - Números

Base decimal: $123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$

Representando um valor inteiro na base decimal como um número binário de tamanho N, precisaremos representar:

- → **valor 0;**
- → **2^{N-1} valores negativos;**
- → **$2^{N-1} - 1$ valores positivos.**

Codificação Binárias - Números

Há duas alternativas para a codificação binária de valores inteiros negativos:

→ notação de excesso;
→ notação de complemento de dois.

O valor do primeiro bit em ambas as notações representar o sinal do número.

Codificação Binárias - Números

Notação de excesso:

- ➔ **Bit de sinal igual a 1 para valores positivos**
- ➔ **Bit de sinal igual a 0 para valores negativos**

0000	-8	1000	0
0001	-7	1001	1
0010	-6	1010	2
0011	-5	1011	3
0100	-4	1100	4
0101	-3	1101	5
0110	-2	1110	6
0111	-1	1111	7

Codificação Binárias - Números

Notação de Complemento de dois:

- ➔ **Bit de sinal igual a 0 para valores positivos**
- ➔ **Bit de sinal igual a 1 para valores negativos**

0000	0	1000	-8
0001	1	1001	-7
0010	2	1010	-6
0011	3	1011	-5
0100	4	1100	-4
0101	5	1101	-3
0110	6	1110	-2
0111	7	1111	-1

Codificação Binárias - Números

$5 = 1x2^2+0x2^1+1x2^1=$

<u>0</u>101, <u>0</u>: sinal positivo

-5

5= 0101

Inverte(0101)= 1010

Soma 1 ao resultado

$$\begin{array}{r} 1010 \\ + \underline{\quad 1} \\ 1011 \end{array}$$

0000	0	1000	-8
0001	1	1001	-7
0010	2	1010	-6
0011	3	1011	-5
0100	4	1100	-4
0101	5	1101	-3
0110	6	1110	-2
0111	7	1111	-1

INTRODUÇÃO A CONCEITOS DE COMPUTAÇÃO

Sistemas de numeração e conversão de bases - Parte I