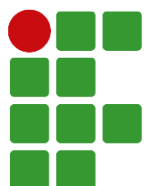


# Conversão entre bases

Organização Estruturada de Computadores

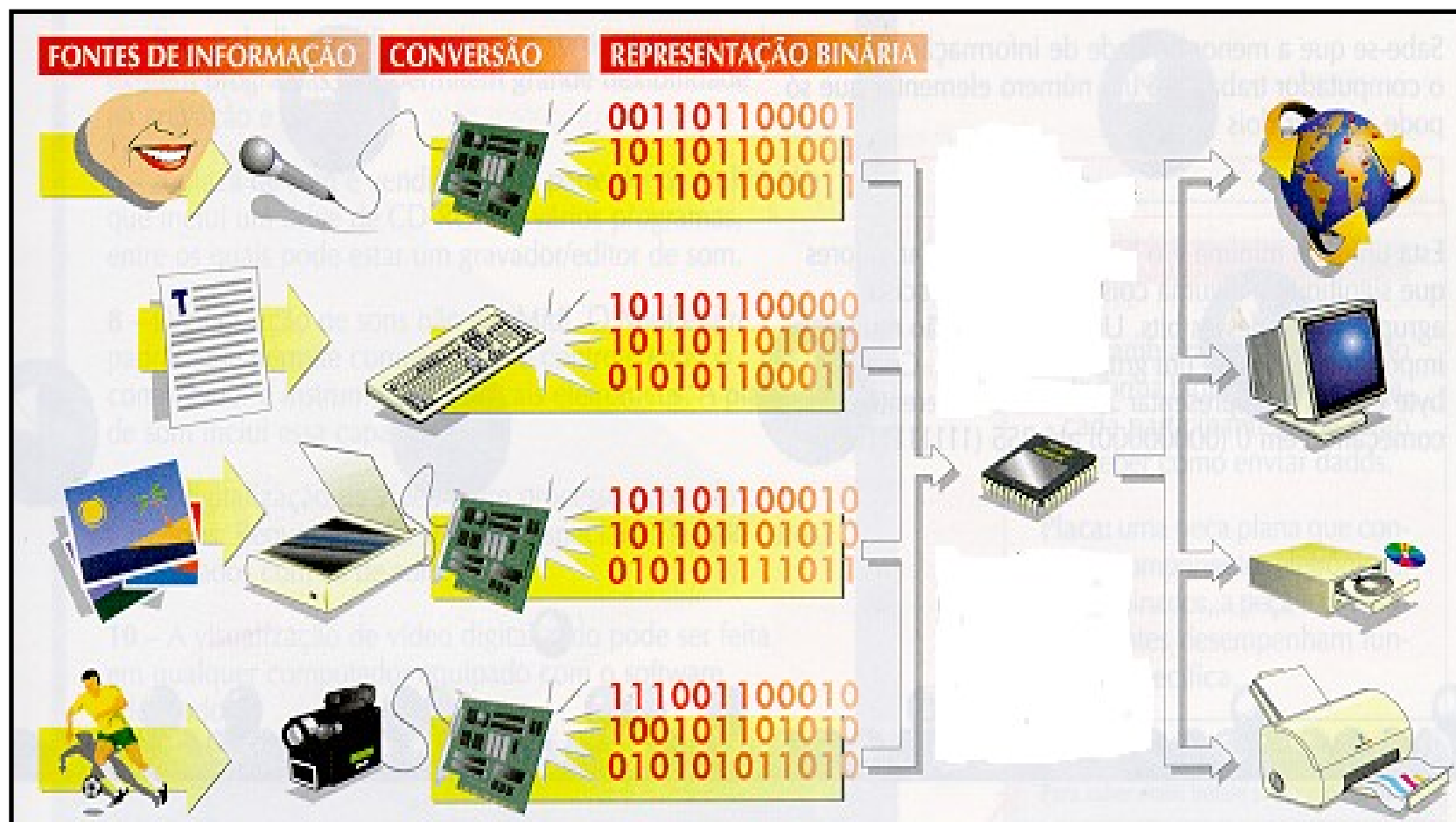
Prof. Munir Mussa



**INSTITUTO FEDERAL**

Espírito Santo  
Campus de Alegre

# Como os computadores representam as informações?



# A informação e sua Representação

- Como os computadores representam as informações utilizando apenas dois estados possíveis - eles são totalmente adequados para números binários.

**0 – desligado**

**1 – ligado**

- Número binário no computador: **bit** [de “**B**inary dig**IT**”]
  - A unidade de informação.
  - Uma quantidade computacional que pode tomar um de dois valores, tais como verdadeiro e falso ou 1 e 0, respectivamente (lógica positiva).

# A informação e sua Representação

- Um bit pode representar apenas **2** símbolos (0 e 1);
- **Necessidade** - unidade maior (**grupo de bits**):
  - dígitos numéricos,
  - letras maiúsculas e minúsculas do alfabeto,
  - sinais de pontuação,
  - símbolos matemáticos e assim por diante.

# Informação e sua representação

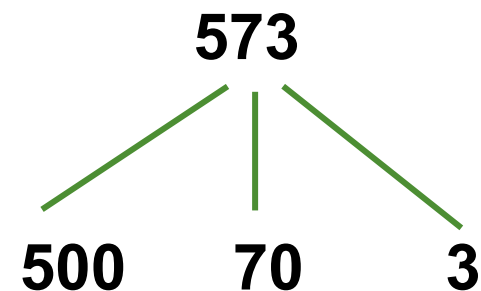
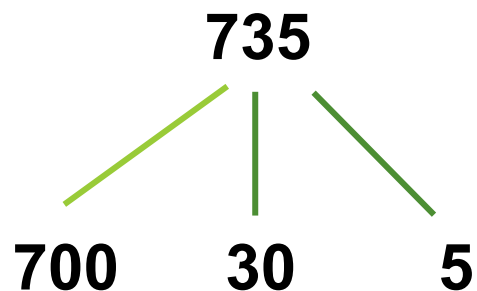
**1 byte = 8 bits = 1 caractere (letra, número ou símbolo)**

- Podemos definir a **palavra** como um conjunto de bits que representa uma informação útil para os computadores. A palavra nos computadores é um valor fixo e constante para um dado processador (p.ex.: 32 bits, 64 bits).

# Sistema de numeração

- **Notação Posicional**
- Valor atribuído a um símbolo *dependente* da posição em que ele se encontra no conjunto de símbolos que representa uma quantidade.
- O valor total do número é a soma dos valores relativos de cada algarismo (decimal).

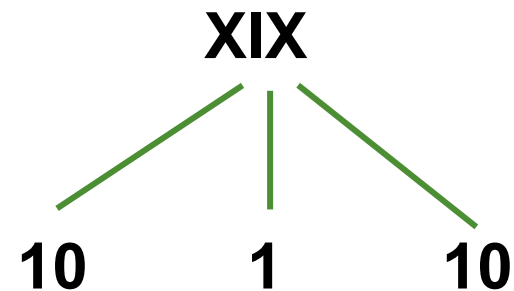
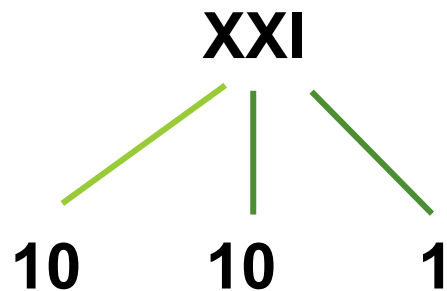
## Sistema de numeração decimal



# Sistema de numeração

- **Notação Não Posicional**
- Valor atribuído a um símbolo é *inalterável*, independente da posição em que se encontre no conjunto de símbolos que representam uma quantidade.

## Sistema de numeração *Romano*



# Sistema de numeração

- **Sistemas de numeração básicos:**

- Decimal
- Binário
- Octal
- Hexadecimal



# Sistema de numeração

Sistema	Base	Algarismos
Binário	2	0,1
Octal	8	0,1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

# Sistema de numeração

- Padrões de Representação
- Letra após o número para indicar a base;
- Número entre parênteses e a base como um índice do número.
- **Exemplo:**
  - Sistema Decimal – 2763**D** ou  $(2763)_{10}$  ou  $2763_{10}$

# Sistema Binário (BASE 2)

- Utiliza dois símbolos para representar quantidades;

0	e	1
---	---	---

- Segue as regras do sistema decimal - válidos os conceitos de **peso** e **posição**. Posições não têm nome específico;
- Cada algarismo é chamado de **bit**. Exemplo:  $101_2$
- **Expressão oral** - diferente dos números decimais;
  - Caractere mais à esquerda - *Most-Significative-Bit* - “**MSB**”.
  - Caractere mais à direita - *Least-Significative-Bit* - “**LSB**”.

# Sistema Binário (BASE 2)

- Cada byte pode representar um intervalo de 0 a 255, ou seja, pode representar até 256 combinações diferentes de dados, pois  **$2^8 = 256$**
- Para melhor compreender a aritmética de números binários, basta fazer todas as combinações possíveis destes 8 bits:

**00000000 = 0**

**00000001 = 1**

**00000010 = 2**

**00000011 = 3**

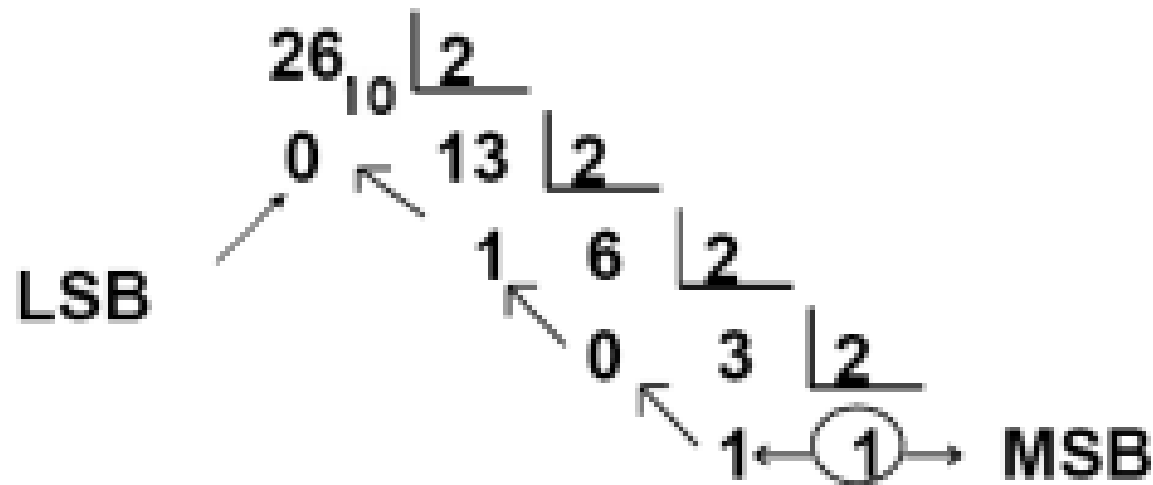
**00000100 = 4**

**....**

**11111111 = 255**

# Mudança de base

- Método das sucessivas divisões
- **Decimal -> Binário**



$$26_{10} = 11010_2$$

# Mudança de base

- Binário -> Decimal

$2^3$	$2^2$	$2^1$	$2^0$
↑	↑	↑	↑
0	1	0	0

→	$0 \times 2^0 = 0$
→	$0 \times 2^1 = 0$
→	$1 \times 2^2 = 4$
→	$0 \times 2^3 = \underline{0} +$
	4

$$0100_2 = 4_{10}$$

# Mudança de base

- Método das sucessivas divisões
- **Decimal -> Octal**
- **Decimal -> Hexadecimal**

# MUDANÇA DE BASE

- Octal -> Decimal

$8^2$	$8^1$	$8^0$	
↑	↑	↑	
<b>2</b>	<b>7</b>	<b>3</b>	
			$3 \times 8^0 = 3$
			$7 \times 8^1 = 56$
			$2 \times 8^2 = \underline{128} +$
			187

$$273_8 = 187_{10}$$



# MUDANÇA DE BASE

- Hexadecimal -> Decimal

$16^2$	$16^1$	$16^0$
↑	↑	↑
<b>3</b>	<b>A</b>	<b>C</b>
		→ $C \times 16^0 = 12$
	→ $A \times 16^1 = 160$	
→ $3 \times 16^2 = \underline{768}$		
+ 940		

$$3AC_{16} = 940_{10}$$

# Conversão entre bases

■ Base 10  $\longleftrightarrow$  Base 2

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
Posição 7	Posição 6	Posição 5	Posição 4	Posição 3	Posição 2	Posição 1	Posição 0

# Sistema Binário (BASE 2)

- Como possuímos oito (8) posições o BIT 1 assume o valor dessa posição na BASE 2;
- Na tabela, o último BYTE é 10000000, observe que o Bit 1 está na posição sete (7), como nossa base é 2 dizemos que essa posição equivale ao valor  $2^7 = 128$
- O maior valor que um único Byte pode representar é 256 o mesmo que a soma de todos os bits (11111111).

# Conversão entre bases

## ■ Binário -> Hexadecimal

0 0 1 0 1 1 0 1 0 1 0 0

2 D 4

8	4	2	1	
0	0	1	0	= 2
1	1	0	1	= 13
0	1	0	0	= 4

$$001011010100_2 = 2D4_{16}$$

# Conversão entre bases

- Hexadecimal -> Binário

**3 F A C D**

8	4	2	1	
0	0	1	1	= 3
1	1	1	1	= 15
1	0	1	0	= 10
1	1	0	0	= 12
1	1	0	1	= 13

$$3FACD_{16} = 00111111101011001101_2$$

# Conversão entre bases

## ■ Binário -> Octal

0 1 0 1 0 1 0 1 0  
└─┬─┘ └─┬─┘ └─┬─┘  
2 5 2

4	2	1	
0	1	0	= 2
1	0	1	= 5
0	1	0	= 2

$$010101010_2 = 252_8$$

# Referências

- Desta aula:
- <http://pt.slideshare.net/fabioeaandrade>
- <http://www.simonsen.br/its/pdf/apostilas/base-tecnica/1/introducao-a-informatica-1-ano-de-informatica-2-capitulo.pdf>