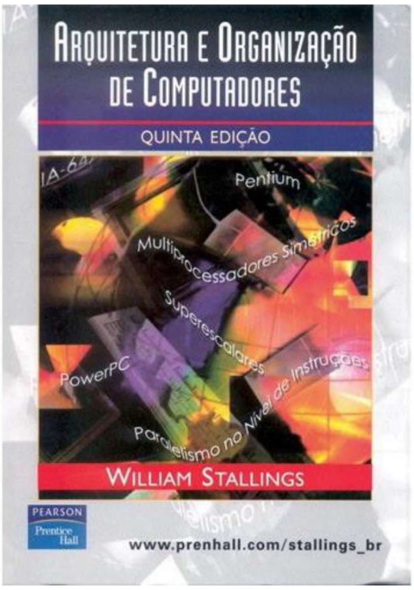
# Aula 01 - Sistemas de Numeração

Prof. João Fernando Mari
*joaof.mari@ufv.br*

## Referências

- STALLINGS, W. **Arquitetura e Organização de Computadores**, 5. Ed., Pearson, 2010.
  - Apêndice A



## Roteiro

- O Sistema Decimal
- O Sistema Binário
- Conversão entre Binário e Decimal
  - Inteiros
  - Frações
- Notação Octal
- Notação Hexadecimal



## O Sistema Decimal

- Sistema Baseado nos Dígitos Decimais
  - **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**
- O que o número 83 significa?
  - Significa 8 vezes 10 mais 3.
  - **83 = (8 x 10) + 3**
- O número 4728 significa...
  - 4 milhares, 7 centenas, 2 dezenas mais 8
  - **4728 = (4 x 1000) + (7 x 100) + (2 x 10) + 8**
    - ou ainda:
  - **4728 = (4 x 1000) + (7 x 100) + (2 x 10) + (8 x 1)**

# O Sistema Decimal

- O sistema decimal é dito possuir base 10
  - Cada número é multiplicado por 10 elevado a uma potencia correspondente a posição do digito
  - **$83 = (8 \times 10^1) + (3 \times 10^0)$**
  - **$4728 = (4 \times 10^3) + (7 \times 10^2) + (2 \times 10^1) + (8 \times 10^0)$**

- O mesmo principio vale para números decimais fracionários
  - São utilizados potências negativas de 10
  - **$0{,}256 = (2 \times 10^{-1}) + (5 \times 10^{-2}) + (6 \times 10^{-3})$**

- Um número composto por parte inteira e parte fracionária
  - **$472{,}256 = (4 \times 10^2) + (7 \times 10^1) + (2 \times 10^0) + (2 \times 10^{-1}) + (5 \times 10^{-2}) + (6 \times 10^{-3})$**



# O Sistema Decimal

- Para a representação decimal do número
  - **$X = \{ d_2 d_1 d_0 , d_{-1} d_{-2} d_{-3} \}$**
    - d = 0, 1, 2, 3, 4, 5, 6, 7, 8 ou 9
- O valor de X é
  - **$X = \sum_i (d_i \times 10^i)$**

## O Sistema Binário

- Sistema Decimal
  - 10 dígitos diferentes usados para representar números com uma base 10

- Sistema Binário
  - Apenas 2 dígitos: 1 e 0
  - Representados com a base 2

- É comum incluir a base do número em subscrito para evitar confusão
  - $\mathbf{83_{10}}$ **e** $\mathbf{4728_{10}}$ → números decimais

- Os dígitos 1 e 0 em notação binária possuem o mesmo significado como quando em notação decimal
  - $\mathbf{0_2 = 0_{10}}$
  - $\mathbf{1_2 = 1_{10}}$



## O Sistema Binário

- Assim como na notação decimal,
  - cada digito de um número binário possui um valor dependendo de sua posição.
  - $\mathbf{10_2 = (1 \times 2^1) + (0 \times 2^0) = 2_{10}}$
  - $\mathbf{11_2 = (1 \times 2^1) + (1 \times 2^0) = 3_{10}}$
  - $\mathbf{100_2 = (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 4_{10}}$

- Valores de frações são representadas com potências negativas da base
  - $\mathbf{1001{,}101_2 = 2^3 + 2^0 + 2^{-1} + 2^{-3} = 8 + 1 + 0{,}5 + 0{,}125 = 9{,}625_{10}}$

# O Sistema Binário

- Para a representação binária do número
  - **$Y = \{ b_2b_1b_0,b_{-1}b_{-2}b_{-3} \}$**
    - b = 0 ou 1
- O valor de Y é
  - **$Y = \sum_i(b_i \times 2^i)$**



# Convertendo entre Binário e Decimal

- Binário para Decimal **(MUITO SIMPLES)**
  - Multiplique cada digito pela potência de 2 apropriada e some os resultados.
  - *Exemplos anteriores!!!*

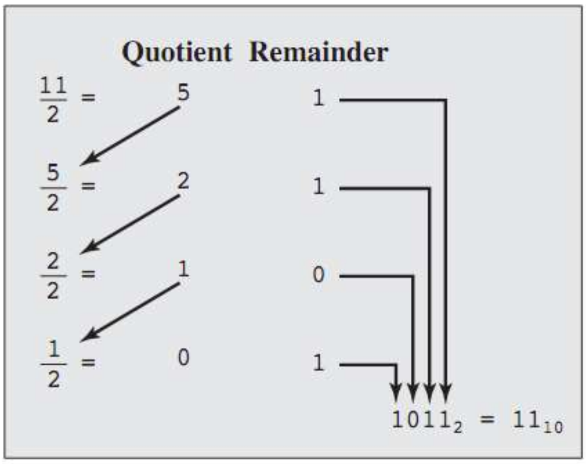
# Convertendo entre Binário e Decimal

- Decimal para Binário **(SIMPLES)**
  - Inteiros e frações são manipulados separadamente.

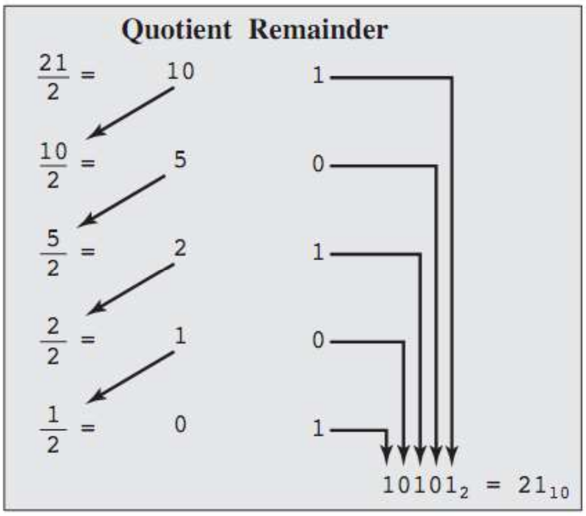
- **Parte INTEIRA**
  - Em notação binária um inteiro é representado por
    - $\mathbf{b_{m-1}b_{m-2}...b_2b_1b_0}$  $\mathbf{b_i = 0 \text{ ou } 1}$
  - Possui o valor
    - $\mathbf{(b_{m-1} \times 2^{m-1}) + (b_{m-2} \times 2^{m-2}) + ... + (b_1 \times 2^1) + b_0}$



# Convertendo entre Binário e Decimal




(a) $11_{10}$



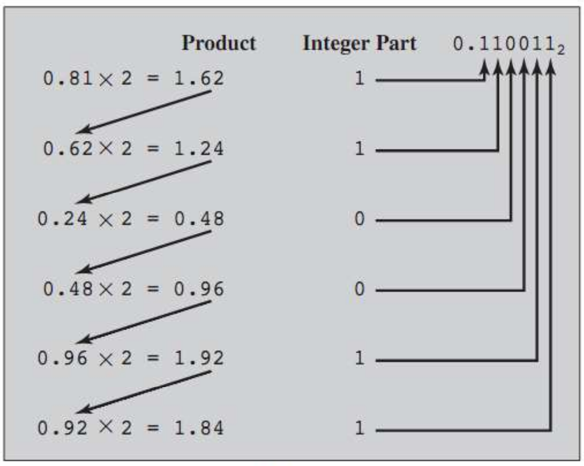

(b) $21_{10}$

# Convertendo entre Binário e Decimal

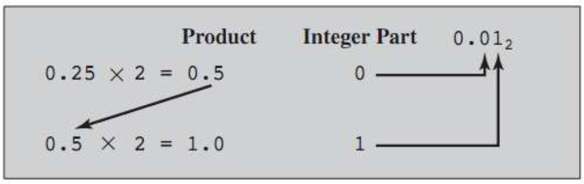
- **Parte FRACIONÁRIA**
  - O algoritmo de conversão envolve **repetidas multiplicações por 2**.
    - A cada passo a parte fracionaria do número é multiplicada por 2.
    - O digito a esquerda da virgula (0 ou 1) contribui para a representação binária.
  - O processo **não é exato**.
    - Uma fração decimal com um número finito de dígitos pode gerar uma representação binária com um número **infinito de bits**.
    - **O processo é cessado após uma sequência predefinida de passos, dependendo da precisão desejada.**



# Conversão entre Binário e Decimal




(a) $0.81_{10} = 0.110011_2$ (approximately)




(b) $0.25_{10} = 0.01_2$ (exactly)

# Notação Octal

- Sistema Octal
  - 8 dígitos diferentes usados para representar números com uma base 8.
  - 0, 1, 2, 3, 4, 5, 6, e 7
- Antigamente utilizado como uma alternativa mais compacta ao **binário**.
  - Programação em linguagem de máquina.
- Atualmente, o sistema hexadecimal é mais utilizado para esse fim.
- A aritmética é semelhante a dos sistemas decimal e binário, o motivo pelo qual não será apresentada.
- EXEMPLO:
  - $4701_8$ em base 10?
  - $\mathbf{4 \times 8^3 + 7 \times 8^2 + 0 \times 8^1 + 1 \times 8^0 = 2048 + 448 + 0 + 1 = 2497_{10}}$



# Notação Octal - Conversões

- Decimal – Octal
  - Parte inteira
    - Sucessivas divisões por 8
  - Parte fracionária
    - Sucessivas multiplicações por 8
- Octal – Decimal
  - $4701_8 = \mathbf{4 \times 8^3 + 7 \times 8^2 + 0 \times 8^1 + 1 \times 8^0 = 2497_{10}}$
- Binário – Octal
  - Dividir os bits em grupos de 3 (partindo do ponto decimal).
  - Substituir cada grupo pelo digito hexadecimal correspondente.
- Octal – Binário
  - Substituir cada digito **octal** pelo grupo de 3 bits correspondente.

# Notação Hexadecimal

- Toda forma de dados nos computadores são representados por códigos binários
  - Natureza binária inerente dos computadores digitais
- Difícil manipulação para humanos.
- Notação mais compacta para profissionais da computação trabalharem com dados brutos
  - Notação decimal
    - Inerente para o ser humano
    - Processo de conversão para binário (e vice-versa) → **TEDIOSO**
  - Notação hexadecimal



# Notação Hexadecimal

- Notação hexadecimal
  - 16 símbolos são usados (dígitos hexadecimais).
  - Cada possível combinação de quatro dígitos binários corresponde a um dígito hexadecimal.

| | | | |
|---|---|---|---|
| 0000 = 0 | 0100 = 4 | 1000 = 8 | 1100 = C |
| 0001 = 1 | 0101 = 5 | 1001 = 9 | 1101 = D |
| 0010 = 2 | 0110 = 6 | 1010 = A | 1110 = E |
| 0011 = 3 | 0111 = 7 | 1011 = B | 1111 = F |

  - Uma sequência de dígitos hexadecimais pode representar um inteiro na base 16:
    - $\mathbf{2C_{16} = (2_{16} \times 16^1) + (C_{16} \times 16^0)}$
      $\mathbf{= (2_{10} \times 16^1) + (12_{10} \times 16^0) = 44_{10}}$

# Notação Hexadecimal

| Decimal (base 10) | Hexadecimal (base 16) | Binário (base 2) |
|---|---|---|
| 0 | 0 | 0000 |
| 1 | 1 | 0001 |
| 2 | 2 | 0010 |
| 3 | 3 | 0011 |
| 4 | 4 | 0100 |
| 5 | 5 | 0101 |
| 6 | 6 | 0110 |
| 7 | 7 | 0111 |
| 8 | 8 | 1000 |
| 9 | 9 | 1001 |
| 10 | A | 1010 |
| 11 | B | 1011 |
| 12 | C | 1100 |
| 13 | D | 1101 |
| 14 | E | 1110 |
| 15 | F | 1111 |
| 16 | 10 | 0001 0000 |
| 17 | 11 | 0001 0001 |
| 18 | 12 | 0001 0010 |
| 31 | 1F | 0001 1111 |
| 100 | 64 | 0110 0100 |
| 255 | FF | 1111 1111 |
| 256 | 100 | 0001 0000 0000 |



# Notação Hexadecimal

- Notação hexadecimal
  - Usada não apenas para representar inteiros
  - Notação concisa para representar qualquer sequência de dígitos binários
    - Textos, números ou qualquer outro tipo de dado.

- Razões para utilizar a notação hexadecimal
  1. Mais compacta do que a notação binária
  2. Na maioria dos computadores, dados binários são organizados em grupos de 4 (1 digito hexadecimal)
  3. Conversão entre binário e hexadecimal extremamente fácil

- Considere a string binária 110111100001

$$\begin{array}{cccccc} 1101 & 1110 & 0001 & = & DE1_{16} \\ D & E & 1 & & \end{array}$$

- A conversão pode ser feita mentalmente.

# Conversões Hexadecimal

- Hexadecimal - Decimal
  - $\mathbf{2C_{16} = (2_{16} \times 16^1) + (C_{16} \times 16^0) = (2_{10} \times 16^1) + (12_{10} \times 16^0) = 44}$
- Decimal – Hexadecimal
  - Divisões sucessivas por 16 (parte inteira)
  - Multiplicações sucessivas por 16 (parte fracionária)
- Hexadecimal – Binário (Extremamente fácil)
  - Substituir cada digito hexadecimal pelo grupo de 4 bits correspondente
- Binário – Hexadecimal (Extremamente fácil)
  - Dividir os bits em grupos de 4 (partindo do ponto decimal)
  - Substituir cada grupo pelo digito hexadecimal correspondente



# FIM – Aula 02