



---

# Organização e Arquitetura de Computadores

Profa. Lucilena de Lima

[luma.delima@gmail.com](mailto:luma.delima@gmail.com)

# Bases Numéricas

---

## Conceitos Básicos

Podemos considerar, a fim de simplificação, que base numérica é um conjunto de símbolos (ou algarismos) com o qual podemos representar uma certa quantidade ou número.

- Representação dos dados
  - Bases Numéricas
  - Conversão de bases
-

# Arquitetura de Computadores

---

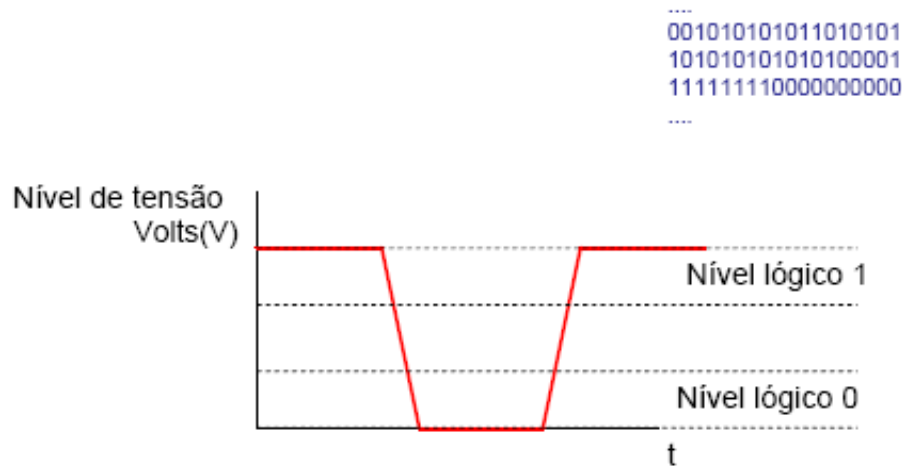
## Organização de um Computador

- O computador é uma máquina capaz de sistemicamente coletar, manipular e fornecer resultados.
  - Processamento de dados consiste em uma série de atividades ordenadamente realizadas com o objetivo de produzir um arranjo determinado de informações e resultados.
  - Um computador digital é um conjunto de componentes integrados segundo uma estrutura de sistema que manipula dados na forma digital (0s e 1s).
    - Unidade Central de Processamento (CPU)
    - Memória
    - Unidades de E/S
-

# Arquitetura de Computadores

## Representação dos Valores Lógicos

- O computador trabalha com a base de numeração binária onde os dados/informação são representados por um conjunto de 0s e 1s.



Os circuitos integrados digitais mais utilizados atualmente podem ser de dois tipos: **TTL** e **CMOS**.

Os circuitos TTL (Transistor-Transistor Logic) são comercializadas em duas séries de modelos: aqueles que começam com 54 são destinados ao uso militar e os que começam com 74 à utilização comercial.

# Arquitetura de Computadores

---

## Representação dos Valores Lógicos

Família TTL (Transistor-Transistor Logic) :-

o sinal de entrada pode variar em uma amplitude que vai de 0V a 5V.

Bit 0 – quando o sinal está entre 0V e 0,8V.

Bit 1 - quando, o sinal está entre 2V e 5V.

As tensões entre 0,8V e 2V não são reconhecidas pelo circuito e deve-se, portanto, evitar seu uso em circuitos digitais.

# Arquitetura de Computadores

---

## Representação dos Valores Lógicos

Família CMOS (Complementary Metal Oxid Semiconductor):- tensões que estejam entre 0V e 1,5V são interpretadas como o baixo nível de energia, ou o Falso lógico. Já as tensões de 3,5V a 5V são reconhecidas como o nível alto ou 1. As tensões intermediárias não são reconhecidas

# Arquitetura de Computadores

---

## Representação dos dados

- A informação é
    - armazenada, transferida e manipulada em grupos de bits
  - Num mesmo computador podem ser empregadas grupos de bits de tamanhos diferentes.
  - Os dados em um computador são representados na forma binária :
    - Bit (1 dígito binário) ou Binary Digit: valor 0 ou 1
    - Byte = 8 bits
    - Palavra (word), conforme a arquitetura ocupará  $n$  bytes.
    - Caracter: conjunto de  $n$  bits que define  $2^n$  caracteres
-

# Arquitetura de Computadores – Sistemas de Numeração

---

- Na base decimal são usados 10 dígitos (ou algarismos) diferentes  $\{0,1,2,3,4,5,6,7,8,9\}$ .
  - Na base hexadecimal dispomos de 16 algarismos  $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$ , onde os dígitos de 0 a 9 são idênticos as decimais, e os dígitos de A a F.
  - Na base octal temos 8 algarismos  $\{0,1,2,3,4,5,6,7\}$ .
  - A base binária ou base 2 é a maneira usual de representação de números em computadores eletrônicos. Nesta forma de representação temos apenas dois algarismos disponíveis  $\{0,1\}$ .
-



# Arquitetura de Computadores

## Tabela Resumo

Binario	Octal	Hexadecimal	Decimal
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	A	10
1011	13	B	11
1100	14	C	12
1101	15	D	13
1110	16	E	14
1111	17	F	15

# Arquitetura de Computadores-

## Notação Posicional

---

- Para a conversão de um número em uma base qualquer para a base decimal usa-se a seguinte fórmula:

$$N = d_{n-1} \times b_{n-1} + d_{n-2} \times b_{n-2} + \dots + d_1 \times b_1 + d_0 \times b_0$$

Onde:

- $d$  : Indica cada algarismo do número
  - $n-1, n-2, 1, 0$  : Indicam a posição de cada algarismo
  - $b$  : Indica a base de numeração
  - $n$  : Indica o número de dígitos inteiros
-

# Arquitetura de Computadores

## Notação Posicional

A notação posicional determina o valor de um número em função da **posição** e do **valor** de cada algarismo dentro do número

Base Decimal

✓ Exemplo: 1303(10)

✓ 1 3 0 3

$$3 \times 10^0 = 3 \times 1 = 3$$

$$0 \times 10^1 = 0 \times 10 = 0$$

$$3 \times 10^2 = 3 \times 100 = 300$$

$$1 \times 10^3 = 1 \times 1000 = 1000$$

# Arquitetura de Computadores-

## Base Decimal - 10

---

- Na base decimal são usados 10 dígitos (ou algarismos) diferentes {0,1,2,3,4,5,6,7,8,9}, formando números em base 10.
- Cada algarismo N de um número possui um valor que depende de sua posição. ( $N * B_{pos}$ )

Exemplo: 1999

- O dígito mais a direita (9) representa a quantidade de unidades, pois está na posição 0. O dígito 9 mais a esquerda, pode ser interpretado como sendo  $9 * 100$

O valor completo do número pode ser calculado como sendo

$$1 * 10^3 + 9 * 10^2 + 9 * 10^1 + 9 * 10^0$$

$$1 * 1000 + 9 * 100 + 9 * 10 + 9 * 1$$

$$1000 + 900 + 90 + 9$$

$$1999$$

---

# Arquitetura de Computadores- Hexadecimal(16) para Decimal

- Na base hexadecimal dispomos de 16 algarismos {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}, onde os dígitos de 0 a 9 são idênticos as decimais, e os dígitos de A a F, correspondem aos valores decimais de 10 a 15, sucessivamente.

Exemplo:

$10AC_{16}$

$$1 \cdot 16^3 + 0 \cdot 16^2 + 10 \cdot 16^1 + 12 \cdot 16^0$$

$$1 \cdot 4096 + 0 \cdot 256 + 10 \cdot 16 + 12 \cdot 1$$

$$4096 + 0 + 160 + 12$$

$4268_{10}$

# Arquitetura de Computadores- Octal(8) para Decimal

---

- Na base octal temos 8 algarismos {0,1,2,3,4,5,6,7}

Exemplo:

$521_8$

$$5 \cdot 8^2 + 2 \cdot 8^1 + 1 \cdot 8^0$$

$$5 \cdot 64 + 2 \cdot 8 + 1 \cdot 1$$

$$320 + 16 + 1$$

$337_{10}$

---

# Arquitetura de Computadores- Binária(2) para Decimal

- A base binária ou base 2 é a maneira usual de representação de números em computadores eletrônicos.
- Nesta forma de representação temos apenas dois algarismos disponíveis {0,1} que correspondem aos sinais elétricos ligado e desligado.

10011<sub>2</sub>

$$1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1$$

$$16 + 0 + 0 + 2 + 1$$

19<sub>10</sub>

- Cada algarismo de uma representação numérica binária é denominados de *bit*, que corresponde à abreviatura de *binary digit*.
- Outras denominações usadas com frequência:
  - byte = conjunto de 8 bits
  - nibble = conjunto de quatro bits ou meio byte

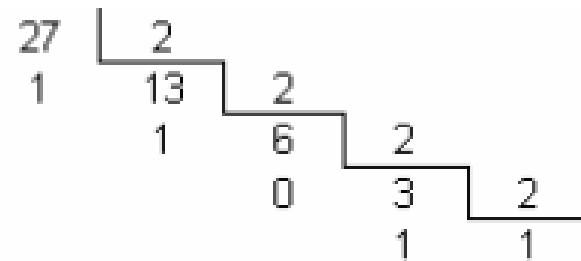
# Arquitetura de Computadores-Conversão da base 10 para uma base qualquer

- Este procedimento consiste em dividir o número representado na base 10 sucessivamente pela nova base em que se deseja representá-lo, até que o quociente da divisão seja menor que a base em questão.
- Em seguida toma-se o último quociente e os restos das sucessivas divisões em ordem inversa e obtém-se, assim, a representação do número na nova base.

Exemplo:

Converter  $27_{10}$  para base 2:

Portanto  $27_{10} = 11011_2$





# Arquitetura de Computadores-Conversão da base 10 para uma base qualquer

•  $3964_{10} \rightarrow \text{base } 8$

3964 8

(4) 495 8

(7) 61 8

(5) 7 8

(7) 0

•  $3964_{10} \rightarrow 7574_8$

•  $3964_{10} \rightarrow \text{base } 16$

$3964 / 16 = 247, \text{ resto} = 12 = C$

$247 / 16 = 15, \text{ resto} = 7$

$15 / 16 = 0, \text{ resto} = 15 = F$

•  $3964_{10} \rightarrow F7C_{16}$

$964 / 2 = 482, \text{ resto} = 0$

$482 / 2 = 241, \text{ resto} = 0$

$241 / 2 = 120, \text{ resto} = 1$

$120 / 2 = 60, \text{ resto} = 0$

$60 / 2 = 30, \text{ resto} = 0$

$30 / 2 = 15, \text{ resto} = 0$

$15 / 2 = 7, \text{ resto} = 1$

$7 / 2 = 3, \text{ resto} = 1$

$3 / 2 = 1, \text{ resto} = 1$

$1 / 2 = 0, \text{ resto} = 1$

$964_{10} = 1111000100_2$

# Arquitetura de Computadores-Conversão da base 2 para uma base $8(2^3)$ e $16(2^4)$

## Base 2 $\rightarrow$ Base 8

- ✓  $2^3 = 8$
- ✓ Três dígitos binários equivalem a um dígito octal
- ✓  $110111010_2$
- ✓  $672_8$

## Base 2 $\rightarrow$ Base 16

- ✓  $2^4 = 16$
- ✓ Quatro dígitos binários equivalem a um dígito hexadecimal
- ✓  $110111010_2$
- ✓  $1BA_{16}$

# Arq.Computadores-Conversão da base:

## 8 para a 16.

## 16 para a 8.

- Usa-se a base 2 como intermediária
- base 8  $\rightarrow$  base 2  $\rightarrow$  base 16
  - ✓  $752_8 \rightarrow 111\ 101\ 010_2 \rightarrow 1EA_{16}$
- base 16  $\rightarrow$  base 2  $\rightarrow$  base 8
  - ✓  $A0C5_{16} \rightarrow 1010\ 0000\ 1100\ 0101_2 \rightarrow 120305_8$

# Representação de Dados

---

- Como os computadores são binários, todas as indicações numéricas referenciam a potência de 2.
  - K representa 1.024 unidades ( $2^{10}$ )
  - M representa 1.048.576 unidades ( $2^{20}$ )
- Logo:
- 64K bytes =  $64 \times 1.024$  bytes = 65.536 bytes,
- 2 M bytes =  $2 \times 1.048.576$  bytes = 2.097.152 bytes

G = ?

T = ?

...

---

# Representação de Caracteres

---

- Os caracteres são representados nos computadores através de códigos numéricos, onde cada combinação de bits representa uma letra ou símbolo diferente.
  - Principais Conjuntos de Caracteres:
    - **BCD - *Binary Code Decimal***: 6 bits por caracter permitindo 64 caracteres diferentes. Não é mais usado atualmente.
    - **EBCDIC - *Extended Binary Coded Decimal Interchange Code***: Codificação exclusiva da IBM, ainda em uso em mainframes. Caracteres de 8 bits, permitindo 256 símbolos.
    - **ASCII - *American Standard Code for Information Interchange*** - 7 bits para representar os caracteres mais um de paridade. permitindo 128 caracteres. Há uma versão extendida (ANSI) com 8 bits, sem paridade, onde estão disponíveis 256 caracteres.
    - **UNICODE - Novo padrão proposto**. Caracteres com 16 bits.
-

Fonte:

---

Introdução a Organização de Computadores – Quarta edição –  
Mário A Monteiro – LTC – 2002 - Capítulo 3 e Apêndice A

Notas de aula – sites específicos

Organização estruturada de computadores, quarta  
edição, Andrew S. Tanenbaum – LTC – 1999

# Bases Numéricas – Exercícios

---

Exercícios

---