

Sistemas de Numeração

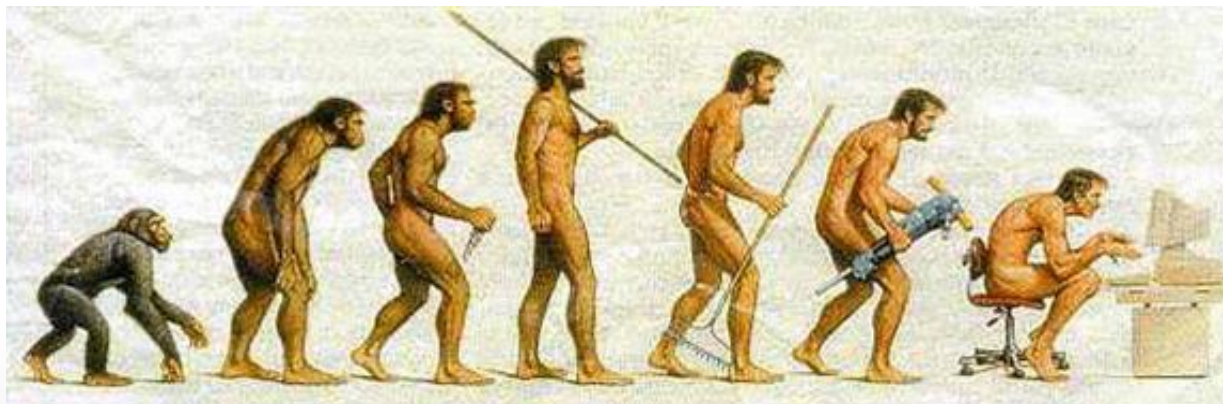
Prof. Gustavo Girão
girao@imd.ufrn.br

Roteiro

- Conhecer os sistemas de numeração binário, octal, decimal e hexadecimal.
- Descrever as principais características de cada um dos sistemas de numeração.
- Aprender a converter um número entre as diversas bases numéricas.
- Conhecer os códigos BCD e Gray.

Origens da Computação

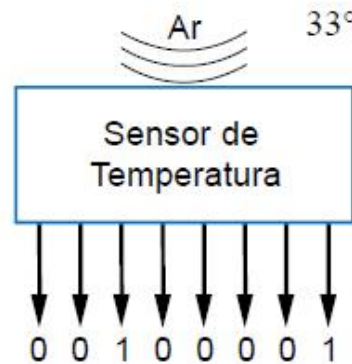
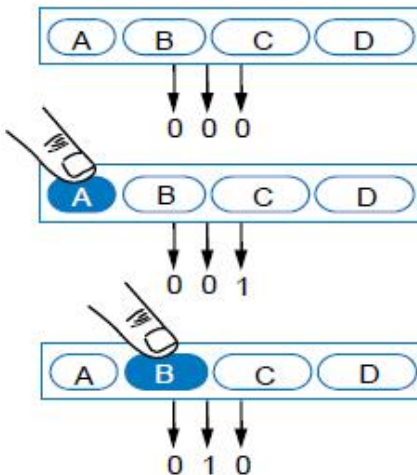
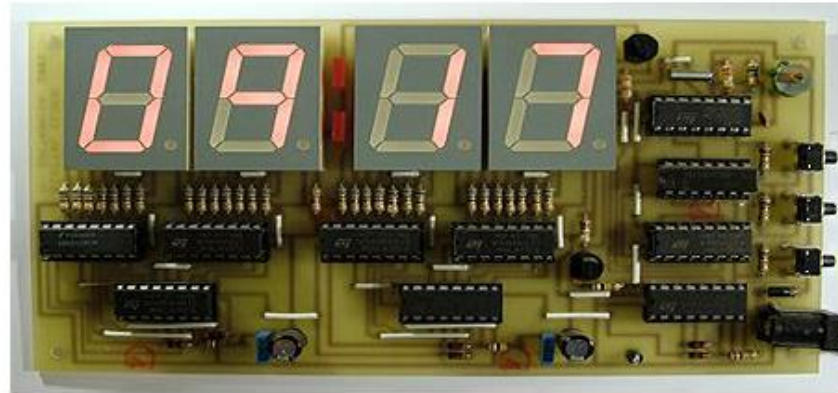
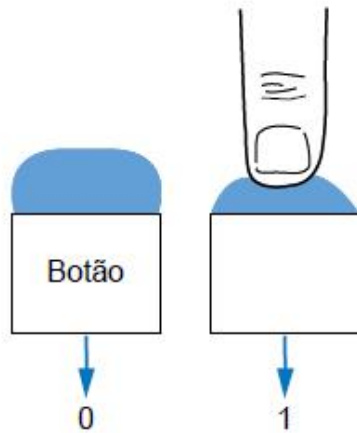
- A capacidade do ser humano em **realizar cálculos** surgiu com sua habilidade de se **comunicar com mais precisão**.
- Evolução da comunicação (desde as pinturas rupestres até o surgimento da escrita).
- Surgimento do alfabeto (não era necessário memorizar uma infinidade de símbolos).



Origens da Computação

- Para representar as quantidades envolvidas em computações, foi necessário o desenvolvimento de sistemas de numeração.
- **Decimal** - o sistema mais evidente devido o uso dos dedos das mãos.
- **Sexagesimal** (base 60) - usado pelos babilônios.
- **Vigesimal** (base 20) - usado pelos maias.
- Romanos, egípcios,

Como representar computações?



Sistemas de numeração

- O que são sistemas de numeração?
 - São sistemas em que um conjunto de números são **representados por numerais de forma consistente**.
 - Os sistemas de numeração são também chamados de **bases**.
- Nomenclatura
 - A base escolhida é indicada por um número sobrescrito após o número em questão.
✧ 1456₁₀
- Exemplos de base
 - Binária, octal, hexadecimal, decimal.

Diferentes Representações

Uma base **b** irá apresentar **b** algarismos, variando entre **0** e **b-1**.

- Sistema Decimal
 - **Base 10**: possui 10 algarismos diferentes.
 - Algarismos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Sistema Binário
 - **Base 2**: possui 2 algarismos diferentes.
 - Algarismos: 0, 1.
- **Quais os motivos que levam o uso do sistema binário nos computadores?**

Aplicação em sistemas computacionais

- O sistema de numeração com o qual estamos mais familiarizados é o **decimal**, cujo alfabeto (coleção de símbolos) é formado por 10 dígitos.
- **Um Computador Decimal:**
 - se trabalhasse com o sistema decimal, um computador precisaria codificar **10 níveis de referência** para caracterizar os 10 dígitos do sistema utilizado.
 - Esses níveis de referência poderiam ser **valores de tensão (0V, 1V, 2V, etc.)** que precisariam ser definidos e interpretados de maneira clara e precisa pela máquina.

Aplicação em sistemas computacionais

- **Desvantagem:** quanto maior o número de interpretações maior a probabilidade de erro. Para decidir que está lendo o número 5 a máquina precisaria ter certeza de que o que leu **não** é: 0, 1, 2, 3, 4, 6, 7, 8, 9.
- **Consequência:** O sistema de numeração mais seguro deveria ser aquele com o menor número de símbolos (dígitos).
- **Conclusão:** o melhor sistema de numeração para uma máquina seria o **binário** com apenas dois dígitos, o zero (0) e o um (1).
 - Obs.: Não há sistema de numeração com alfabeto de um único dígito. Todo sistema de numeração precisa dos conceitos de presença (1) e ausência (0).

Diferentes Representações

- Sistema Octal
 - Base 8: possui 8 algarismos diferentes.
 - Algarismos: 0, 1, 2, 3, 4, 5, 6, 7.
 - Cada dígito é representado por três bits.
- Sistema Hexadecimal
 - Base 16: possui 16 algarismos diferentes.
 - Algarismos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
 - Cada dígito é representado por quatro bits.

Diferentes Representações

Decimal	Binário	Octal	Hexadecimal
0	00000	00	00
1	00001	01	01
2	00010	02	02
3	00011	03	03
4	00100	04	04
5	00101	05	05
6	00110	06	06
7	00111	07	07
8	01000	10	08
9	01001	11	09
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10

Geração de Inteiros

- Algoritmo de avanço de dígitos:

Avançar um dígito de um alfabeto ordenado consiste em substituí-lo pelo próximo dígito na hierarquia. O dígito de maior valor do conjunto é sempre avançado para o aquele de menor valor na hierarquia.

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 0$

- Algoritmo de geração de inteiros:

a) o primeiro inteiro é o zero

b) o próximo inteiro é obtido do precedente na lista avançando-se seu dígito mais à direita. No caso deste dígito avançar para zero, avança-se, então, o dígito adjacente à esquerda.

Geração de Inteiros

Exemplo: Gerar os 26 primeiros inteiros do sistema decimal.

0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 →
12 → 13 → 14 → 15 → 16 → 17 → 18 → 19 → 20 → 21
→ 22 → 23 → 24 → 25

- Observe que o nove avança para o zero, logo o dígito mais à esquerda (o zero, não mostrado explicitamente no número) é avançado para 1 gerando o próximo número na lista, o 10.

Teorema Fundamental da Numeração

Converter qualquer base para a base decimal

$$N_b = a_n.b^n + \dots + a_2.b^2 + a_1.b^1 + a_0.b^0 \\ + a_{-1}.b^{-1} + a_{-2}.b^{-2} + \dots + a_{-n}.b^{-n}$$

a_n : dígito

b : base

n : posição do dígito em relação à vírgula

Faixa de Contagem

- Um número escrito em uma base **B** que possui **N** dígitos, permite a contagem de B^N elementos, com valores variando entre 0 e $B^N - 1$.

Nº de dígitos	Nº mínimo	Nº máximo	Quantidade máxima de números
1	$0 \cdot 10^0 = 0_{10}$	$9 \cdot 10^0 = 9_{10}$	10
2	$0 \cdot 10^1 + 0 \cdot 10^0 = 0_{10}$	$9 \cdot 10^1 + 9 \cdot 10^0 = 99_{10}$	100
3	$0 \cdot 10^2 + 0 \cdot 10^1 + 0 \cdot 10^0 = 0_{10}$	$9 \cdot 10^2 + 9 \cdot 10^1 + 9 \cdot 10^0 = 999_{10}$	1000
N	0	$10^N - 1$	10^N

Faixa de Contagem

- Exercício
 - **Qual o número máximo** que pode ser contado com 4 bits?
 - **Quantos números** podem ser contados com 4 bits?
 - Quantos bits (i.e.: sistema binário) eu preciso para representar 25 valores distintos?
 - Que funções matemáticas representam estes valores?

Sistemas de numeração digital

- O Sistema Decimal:
 - Dez símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
 - ✧ Cada número é um *dígito* (do latim, dedo).
- Dígitos mais significantes (MSD) e dígitos menos significantes (LSD).
- Valor posicional pode ser declarado como um dígito multiplicado por uma potência de 10.

Sistemas de numeração digital

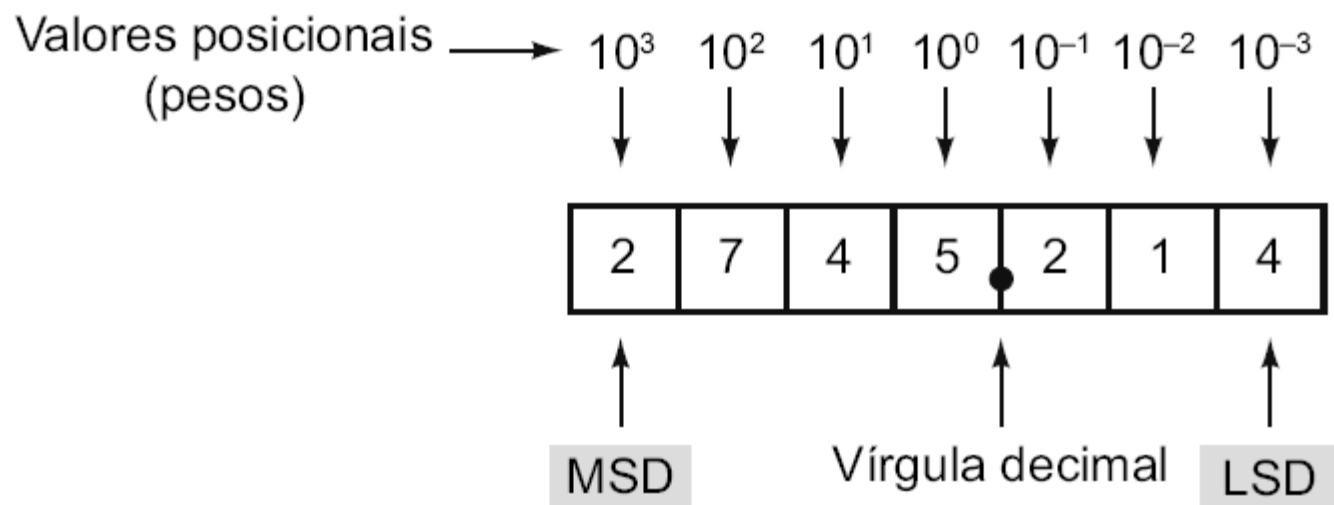


FIGURA 1.8 Valores posicionais de um número decimal expresso como potências de 10.

Sistemas de numeração digital

- O Sistema Binário (base 02) :
 - Dois símbolos: 0 e 1.
 - ✧ Emprega-se no projeto de circuitos eletrônicos com apenas dois diferentes níveis de tensão obrigatórios.
 - Valor posicional pode ser indicado como um dígito multiplicado por uma potência de 2.

Sistemas de numeração digital

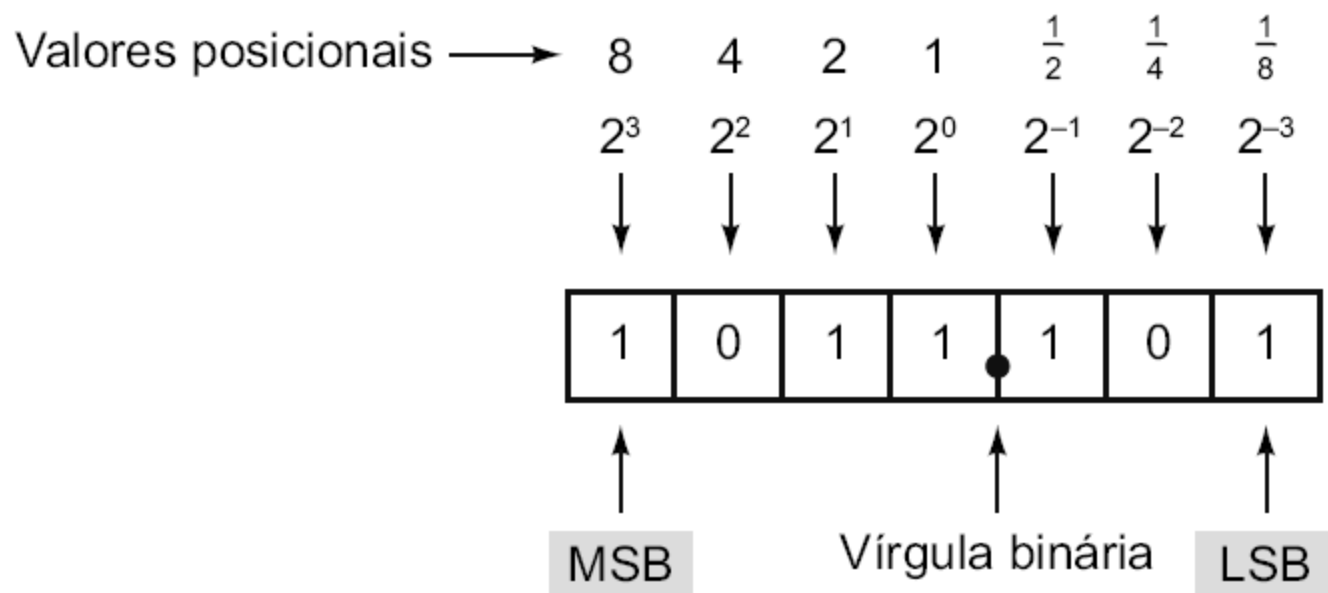
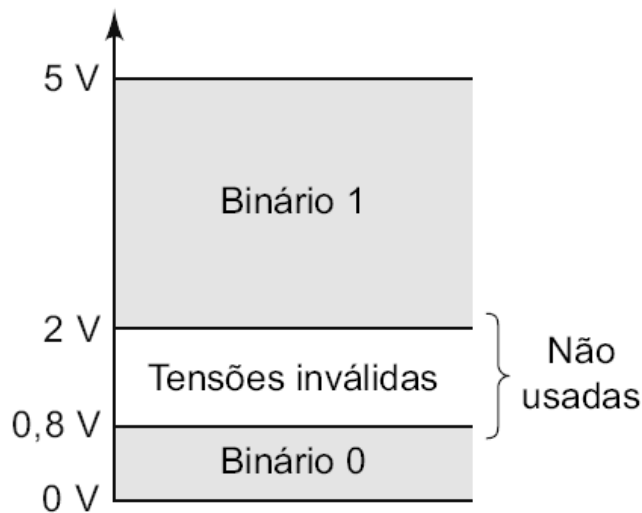


FIGURA 1.10 Valores posicionais de um número binário expresso como potências de 2.

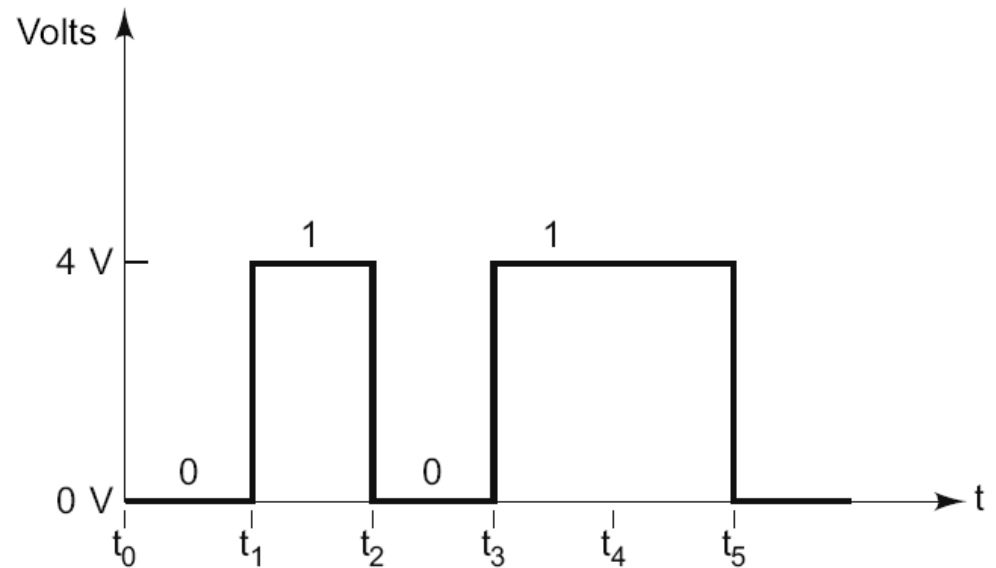
Representação de quantidades binárias

- Representação típica dos dois estados de um sinal digital.
- Muitas vezes, **ALTO** e **BAIXO** são utilizados para descrever os estados de um sistema digital em vez de 1 e 0.
- O osciloscópio é utilizado para produzir diagramas de tempo.
- Os diagramas de tempo mostram a tensão versus o tempo. São usados para demonstrar como os sinais digitais evoluem com o tempo, ou para comparar dois ou mais sinais digitais.

Representação de quantidades binárias



(a)



(b)

FIGURA 1.13 (a) Designações de tensão típicas em um sistema digital; (b) diagrama de tempos de sinal digital típico.

CONVERSÕES ENTRE SISTEMAS

Conversões de Números

- Conversão de base R para decimal
- converte-se a base e cada dígito do número para o equivalente decimal.
- decompõe-se o número de acordo com a estrutura posicional e, usando aritmética decimal, efetuam-se as operações de produtos e somas.
- Notação: (...)R ler como o número do parêntesis expresso na base R.

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13$$

$$(2B0)_{16} = 2 \times 16^2 + (11) \times 16^1 + 0 \times 16^0 = 512 + 176 + 0 = 688$$

Conversões de Números

- Passagem de uma base 10 para a base R
- Parte inteira: Algoritmo da divisão repetida
- Divide-se o inteiro decimal repetidamente pela base R até que se obtenha um quociente inteiro igual a zero. Os restos das divisões sucessivas, lidos do último para o primeiro, constituem o número transformado para a base R.

Conversão Decimal \rightarrow Binário

715 \div 2

① 357 \div 2

① 178 \div 2

① 89 \div 2

① 44 \div 2

① 22 \div 2

① 11 \div 2

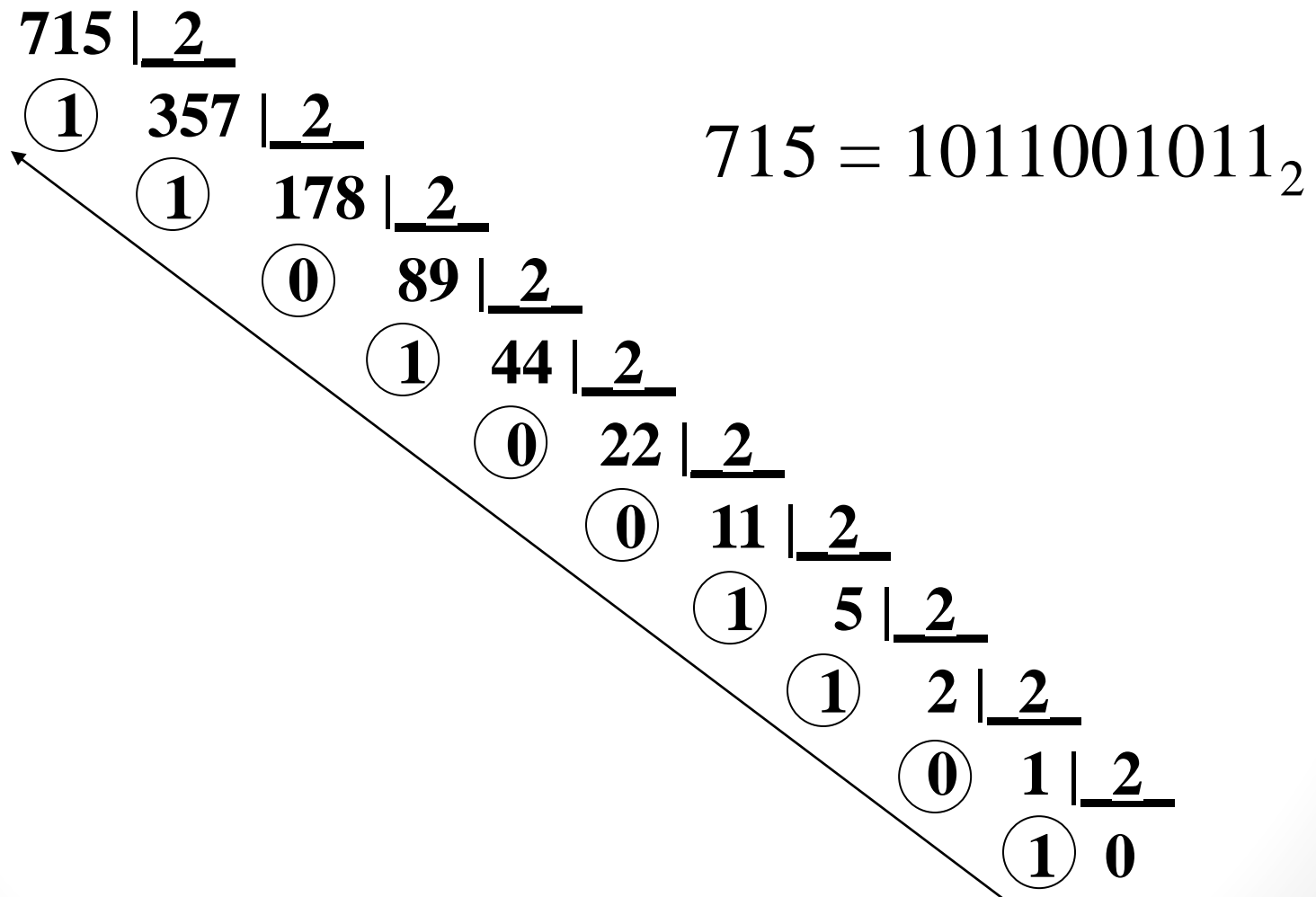
① 5 \div 2

① 2 \div 2

① 1 \div 2

① 0

715 = 1011001011₂



Conversão Decimal → Octal

$$715 \mid \underline{8}$$

$$\textcircled{3} \ 89 \mid \underline{8}$$

$$\textcircled{1} \ 11 \mid \underline{8}$$

$$\textcircled{3} \ 1 \mid \underline{8}$$

$$\textcircled{1} \ 0$$

$$715 = 1313_8$$

Conversão Decimal → Hexadecimal

$$715 \mid \underline{16}$$

$$715 = 2CB_{16}$$

$$\begin{array}{r} \textcircled{11} \quad 44 \mid \underline{16} \\ \textcircled{12} \quad 2 \mid \underline{16} \\ \textcircled{2} \quad 0 \end{array}$$

Hexadecimal

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

A=10 , B=11 , C=12 , D=13 , E=14 , F=15

Conversão Decimal

- Passagem de uma base 10 para a base R
- Parte fracionária: Algoritmo da multiplicação repetida
 - A parte fracionária é multiplicada por R. A parte inteira desse produto é guardada e a parte fracionária é novamente multiplicada por R. O processo é repetido até que se obtenha um número com parte fracionária nula ou até que se considere a aproximação suficiente.
 - As partes inteiras dos produtos sucessivos, lidas da primeira para a última, formam a parte fracionária do número transformado.

Conversão Decimal

- Passagem de uma base 10 para a base R
- Parte fracionária: Algoritmo da multiplicação repetida.
- Exemplo:

Exemplo

Transformar $(0,4375)_{10}$ para a base 2.

$$0,4375 \times 2 = 0,8750$$

$$0,8750 \times 2 = 1,7500$$

$$0,7500 \times 2 = 1,5000$$

$$0,5000 \times 2 = 1,0000$$

- Então $(0,4375)_{10} = (0,0111)_2$

Conversão Binário → Decimal

$1011001011_2 = 1+2+0+8+0+0+64+128+0+512 = 715$

1	0	1	1	0	0	1	0	1	1	2^0	$1 \times 2^0 = 1$
										2^1	$1 \times 2^1 = 2$
										2^2	$0 \times 2^2 = 0$
										2^3	$1 \times 2^3 = 8$
										2^4	$0 \times 2^4 = 0$
										2^5	$0 \times 2^5 = 0$
										2^6	$1 \times 2^6 = 64$
										2^7	$1 \times 2^7 = 128$
										2^8	$0 \times 2^8 = 0$
										2^9	$1 \times 2^9 = 512$

Conversão Octal \rightarrow Decimal

$$1313_8 = 3 + 8 + 192 + 512 = 715$$

Diagram illustrating the positional weights for the octal number 1313_8 :

- $3 \times 8^0 = 3$
- $1 \times 8^1 = 8$
- $3 \times 8^2 = 192$
- $1 \times 8^3 = 512$

Conversão Hexadecimal → Decimal

$$2CB_{16} = 11 + 192 + 512 = 715$$

$B \times 16^0 = 11 \times 16^0 = 11$
 $C \times 16^1 = 12 \times 16^1 = 192$
 $2 \times 16^2 = 512$

Conversões de Números Binários

- **Octal para hexadecimal**

- Opção 1: octal - decimal - hexadecimal
- Opção 2: octal - binário - hexadecimal
- Por motivo de simplicidade, será adotada a opção 2

$$7225_8 = \underbrace{7}_{111} \underbrace{2}_{010} \underbrace{2}_{010} \underbrace{5}_{101}$$

$$7225_8 = 111010010101_2$$

$$111010010101_2 = \underbrace{1110}_E \underbrace{1001}_9 \underbrace{0101}_5$$

$$7225_8 = E95_{16}$$

Outras Conversões

- Binário \rightarrow Octal;
- Binário \rightarrow Hexadecimal;
- Octal \rightarrow Binário;
- Hexadecimal \rightarrow Binário;

Conversões de Números Binários

- Binário para octal
 - Um dígito pode ser representado por 3 bits

$$1101111_2 = \underbrace{001}_1 \underbrace{101}_5 \underbrace{111}_7$$

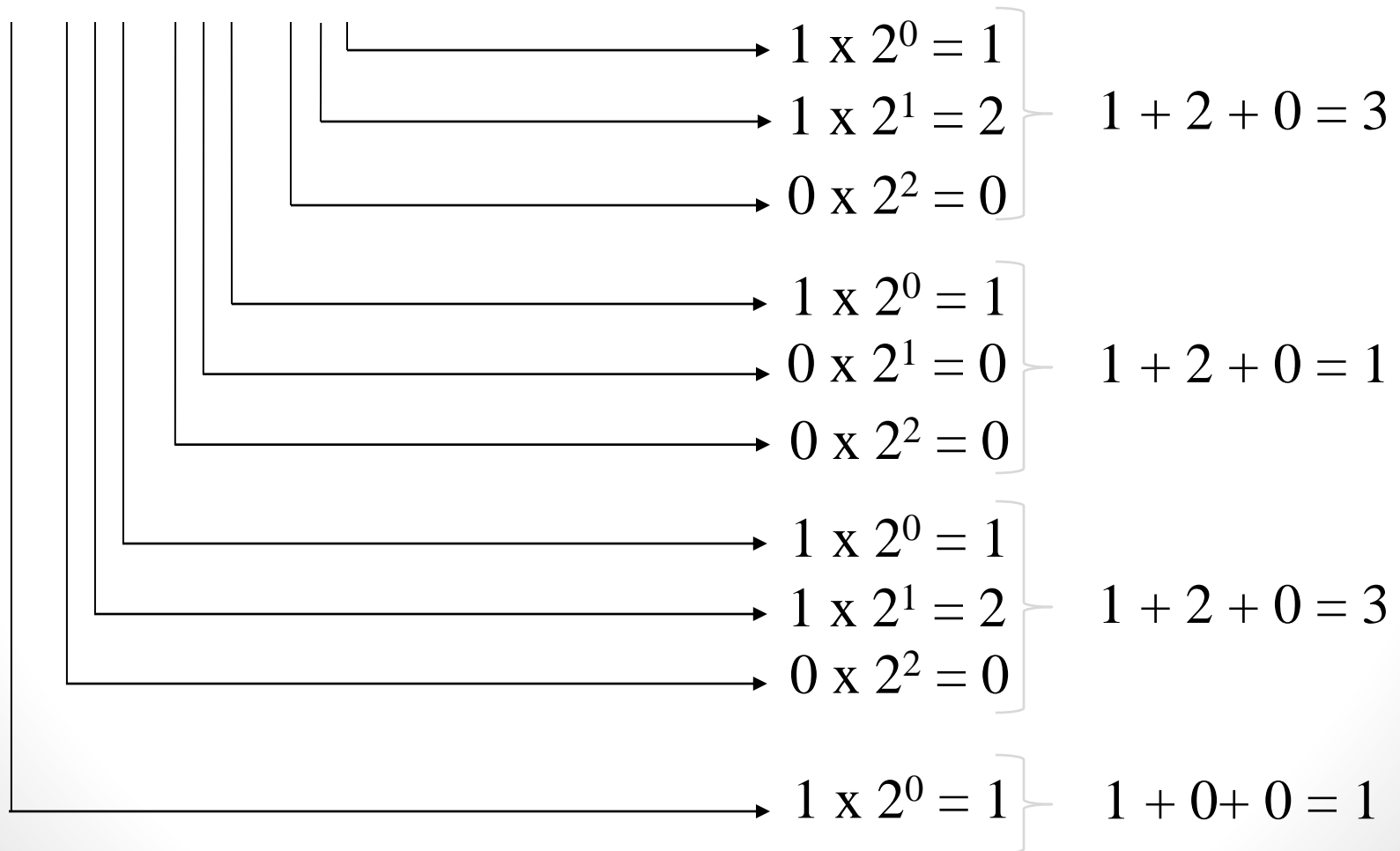
$$1101111_2 = 157_{10}$$

- Binário para decimal
 - Paridade
 - ✧ LSB (least significant bit) = 1: ímpar
 - ✧ LSB (least significant bit) = 0: par

$$11101_2 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 16 + 8 + 4 + 1 = 29_{10}$$

Conversão Binário \rightarrow Octal

1 011 001 011₂ 1011001011₂ $\rightarrow\rightarrow\rightarrow$ 1313₈



Conversões de Números Binários

- Binário para hexadecimal
 - Um dígito pode ser representado por 4 bits

$$1011101111_2 = \underbrace{0010}_2 \underbrace{1110}_E \underbrace{1111}_F$$

$$1011101111_2 = 2EF_{16}$$

Conversão Binário → Hexadecimal

1011001011_2

- Segue o mesmo princípio da conversão de binário para octal, só que agora agrupando de quatro em quatro bits.

$\underbrace{10}_{\text{bits}}$	$\underbrace{1100}_{\text{bits}}$	$\underbrace{1011}_{\text{bits}}$
$0 + 0 + 2 + 0 = 2$	$8 + 4 + 0 + 0 = 12$	$8 + 0 + 2 + 1 = 11$
$\underbrace{\hspace{10em}}$	$\underbrace{\hspace{10em}}$	$\underbrace{\hspace{10em}}$
2	C	B
		16

Conversão Octal \rightarrow Binário

- Simplesmente pega-se cada algarismo na base Octal e converte-se seu valor decimal para a base Binária, representado-se cada um dos algarismos da base Octal com três bits, mantendo-se a ordem original (operação inversa à conversão de Binário para Octal):

$$1313_8 \rightarrow 1 \ 011 \ 001 \ 011_2$$

Conversão Hexadecimal → Binário

- Da mesma forma, simplesmente pega-se cada algarismo na base Hexadecimal e converte-se seu valor decimal para a base Binária, só que agora representado-os com quatro bits (operação inversa à conversão de Binário para Hexadecimal):

$$2CB_{16} \rightarrow 10 \ 1100 \ 1011_2$$

Código BCD

- BCD: Decimal codificado em binário
- Permite a representação de números decimais utilizando números binários
- Remove a longa conversão de binário puro para decimal e vice-versa
- São utilizados **quatro** bits para representar cada dígito decimal

• $876_{10} =$



8	7	6
↓	↓	↓
1000	0111	0110

• $876_{10} = 100001110110(\text{BCD})$

Código BCD

- Conversão de decimal para binário puro e para BCD pode
- resultar em valores diferentes
- Conversão de decimal para BCD:

- $40_{10} =$

4	0
	
0100	0000

- $40_{10} = 01000000(\text{BCD})$
- Conversão de decimal para binário puro:
- $40_{10} = 101000_2$

Código BCD

- Desvantagem do uso do Código BCD
 - Para a representação de cada dígito decimal são utilizados 4 dígitos binários
 - Quantos valores podem ser representados com quatro dígitos binários?
 - ✧ Quantidade de números = BN
 - ✧ $B = 2$; $N = 4$;
 - ✧ Número máximo = $2^4 = 16$
- Podem ser representados 16 números, mas o sistema decimal só possui 10 dígitos
- **Resultado:** o código BCD é ineficiente com relação ao tamanho da palavra resultante
- **Vantagem** do uso do Código BCD
 - Fácil conversão para decimal
 - É necessário memorizar apenas os 10 dígitos decimais e suas representações em BCD
 - Um circuito lógico simples pode realizar esta conversão

Código Gray

- É uma forma de código binário
- Diferencia do binário puro pela forma de incremento para o próximo número
- No Código Gray, apenas 1 dígito pode mudar o estado em cada incremento
- **Vantagem:** em máquinas, apenas um bit deve ser enviado a cada contagem, incrementando o valor resultante.

Decimal	Binário	Código Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Referências

- **TOCCI, Ronald J; Widmer, Neal S. Sistemas Digitais: princípios e Aplicações. 11. ed. São Paulo SP: Pearson, 2011, 817 p. ISBN 9788576050957**
 - Capítulo 1
- **STALLINGS, William. Arquitetura e organização de computadores. 10. ed. São Paulo: Pearson, 2017. 814 p.**
 - Capítulo 9
- **PATTERSON, David A; HENNESSY, John L. Organização e projeto de computadores: A interface HARDWARE/SOFTWARE. Rio de Janeiro: Elsevier, 2005, 3ª edição.**

Sistemas de Numeração

Prof. Gustavo Girão
girao@imd.ufrn.br