INE 5401 – INTRODUÇÃO À COMPUTAÇÃO

ASSUNTO: REPRESENTAÇÃO DE INFORMAÇÕES EM UM COMPUTADOR.

Obs: este Material didático fui elaborado tendo como base nos slides, disponíveis na internet, de autoria da Professora Joseana Macêdo Fechine, da UFCG.

O computador, sendo um equipamento eletrônico, armazena e movimenta as informações internamente sob forma eletrônica; tudo o que faz é ser capaz de reconhecer dois estados físicos distintos, produzidos pela eletricidade, pela polaridade magnética ou pela luz refletida.

O computador, por ser uma máquina eletrônica, só consegue processar duas informações: a **presença** ou **ausência** de energia.

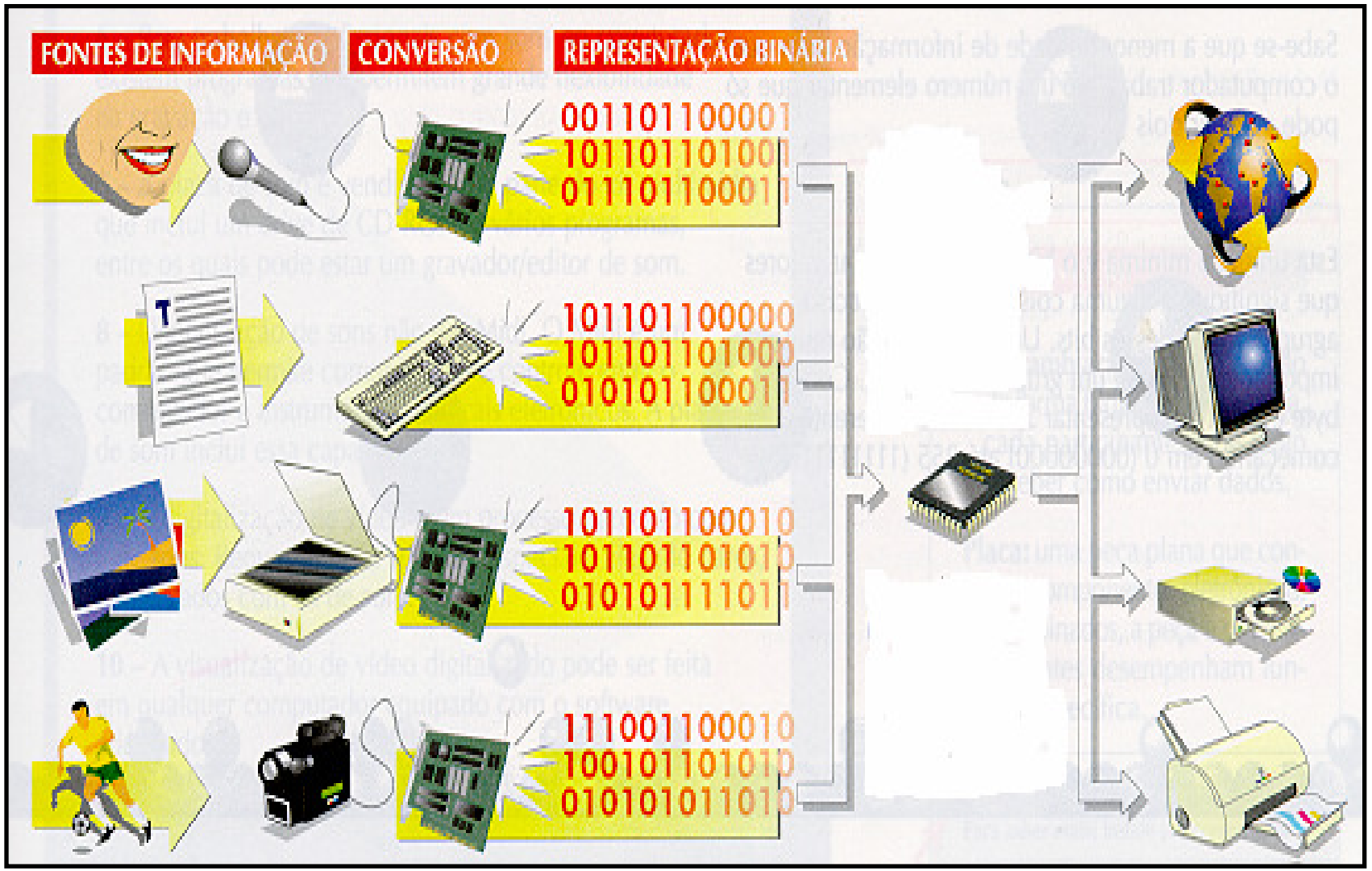
Para que a máquina pudesse representar eletricamente todos os símbolos utilizados na linguagem humana, seriam necessários mais de 100 diferentes valores de tensão (ou de corrente).

Tipos de grandezas

Analógica ≡ contínua
Digital ≡ discreta

Computadores analógicos – Trabalham com sinais elétricos de infinitos valores de tensão e corrente.

Computadores digitais – Trabalham com dois níveis de sinais elétricos: **alto e baixo**. Representam dados por meio de um símbolo facilmente identificado (***dígito***).


FONTES DE INFORMAÇÃO
CONVERSÃO
REPRESENTAÇÃO BINÁRIA
001101100001
101101101001
011101100011
101101100000
101101101000
010101100011
101101100010
101101101010
010101111011
111001100010
100101101010
010101011010

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

A menor unidade de informação no computador é o **bit** (de "**B**inary dig**IT**").Uma quantidade computacional que pode tomar um de dois valores, tais como verdadeiro e falso ou 1 e 0, respectivamente (lógica positiva).

Um bit pode representar apenas **2** símbolos (0 e 1), para representar números e outros símbolos, como os caracteres e os sinais de pontuação que usamos nas linguagens escritas necessitamos de mais bits: uma unidade maior ou um GRUPO de bits.

Unidade maior (**grupo de bits**) - precisa ter bits suficientes para representar todos os símbolos que possam ser usados:

dígitos numéricos,
letras maiúsculas e minúsculas do alfabeto,
sinais de pontuação,
símbolos matemáticos e pixels,etc.

PIXEL:

MENOR UNIDADE DE UMA IMAGEM.

BASEADO NO MODELO **RGB**, CADA PIXEL É COMPOSTO POR TRÊS PONTOS.

CADA PONTO REPRESENTA UMA TONALIDADE DAS CORES: VERMELHA VERDE E AZUL.

PIXEL= (PONTO R, PONTO G, PONTO B)

CADA PONTO É FORMADO POR 8 BITS, OU SEJA, PODE REPRESENTAR 256 TONALIDADES DIFERENTES DA SUA COR.

A COMBINAÇÃO DAS TONALIDADES DESSAS 3 CORES PODE GERAR:

16 MILHÕES DE CORES DIFERENTES.

QUANTO MAIOR É O NUMERO DE PIXELS EM UMA IMAGEM, MELHOR É A SUA RESOLUÇÃO:

640 x 480 : 307 MIL PIXELS

800 x 600 : 480 MIL

1024 x 768 : 786 MIL

Caracteres alfabéticos maiúsculos	**26**
Caracteres alfabéticos minúsculos	**26**
Algarismos	**10**
Sinais de pontuação e outros símbolos	**32**
Caracteres de controle	**24**
Total	**118**

CAPACIDADE DE REPRESENTÇÃO

Bits	Símbolos
2	4
3	8
4	16
5	32
6	64
7	128
8	**256**
9	512
10	1024

O BYTE (BInary TErm)

Grupo ordenado de 8 bits, para efeito de manipulação interna mais eficiente.

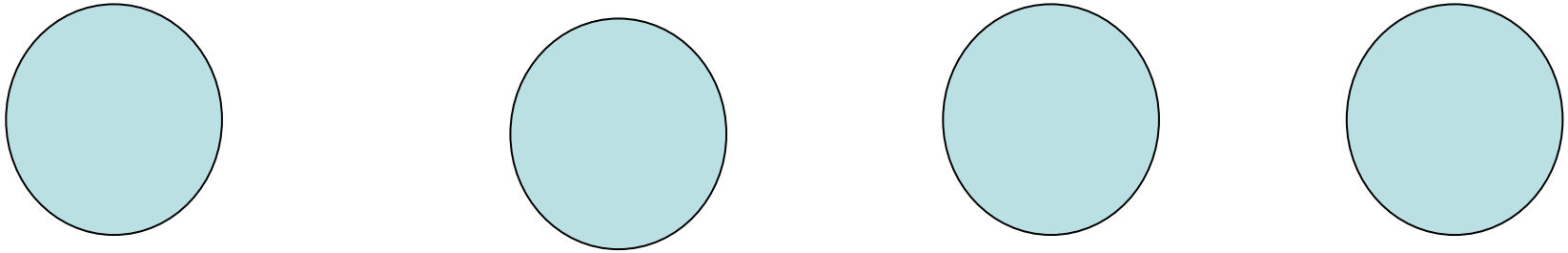
Tratado de forma individual, como unidade de armazenamento e transferência.

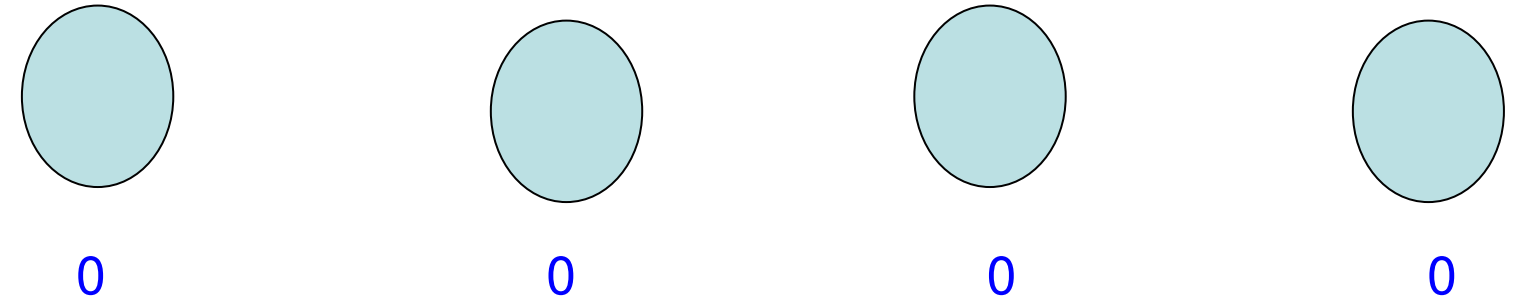
Unidade de memória usada para representar um caractere. Todas as letras, números e outros caracteres são codificados e decodificados pelos equipamentos através dos bytes que os representam, permitindo, dessa forma, a comunicação entre o usuário e a máquina.

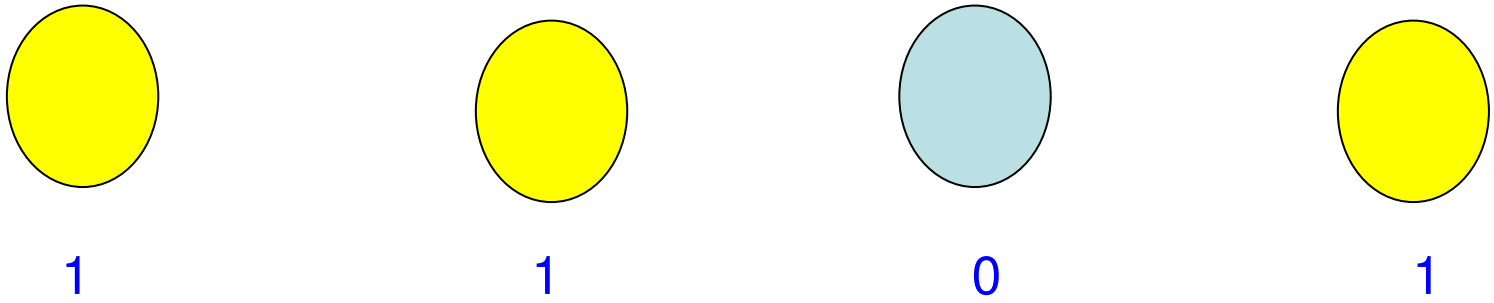
WORD:Podemos definir a **palavra** como um conjunto de bits que representa uma informação útil para os computadores. A palavra nos computadores é um valor fixo e constante para um dado processador (p.ex.: 32 bits, 64 bits).

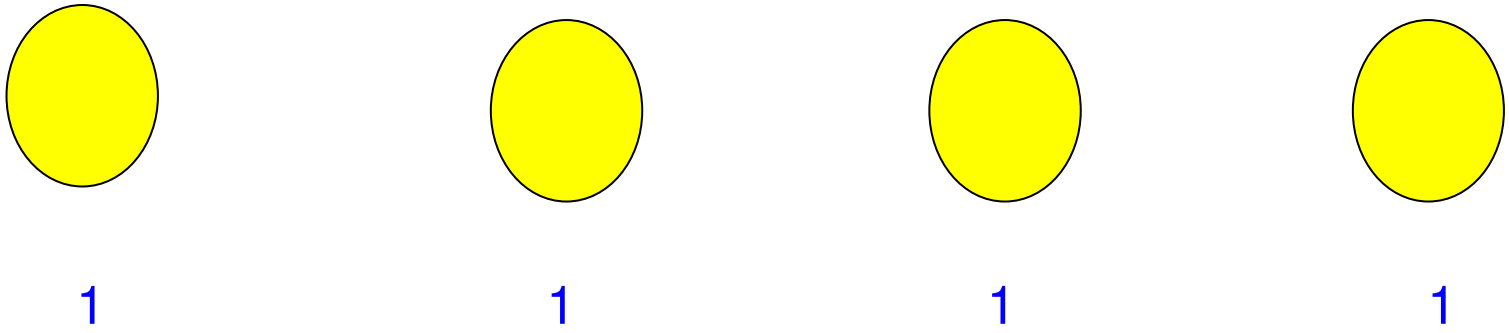
O que é código binário, como surgiram os códigos hoje utilizados?

ALFAIATE MALUCO:




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


1
1
0
1


1
1
1
1

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

Antes de maio de 1961 a maioria dos sistemas de computadores tinha uma maneira particular de representar os seus caracteres alfanuméricos.

Principais Sistemas desenvolvidos para representar símbolos com números binários:

EBCDIC (*Extended Binary Coded Decimal Interchange Code* – Código Ampliado de Caracteres Decimais Codificados em Binário para o Intercâmbio de Dados).

ASCII (*American Standard Code for Information Interchange* – Código Padrão Americano para o Intercâmbio de Informações).

EBCDIC

Código de 8 BITS (256 símbolos).
Usado em *mainframe* IBM e em sistemas de médio porte, raramente encontrado em microcomputadores.

ASCII

Padrão definido pela organização ANSI.
Código de 7 BITS (128 combinações de caracteres).
No PC existe o ASCII Estendido, 8 BITS(utiliza outros 128 códigos para símbolos gráficos, e línguas diferentes do inglês).

UNICODE

Novo padrão para representação de dados, utiliza 2 BITES para a representação de símbolos (mais de 65.000 símbolos). Linguagem JAVA adota esse padrão.

EM ASCII, é importante notar que há apenas 95 caracteres que podem ser impressos. E eles são numerados de 32 a 126 pois os primeiros códigos (de 0 a a 31) foram reservados para caracteres de controle, ou seja, que controlam funções ou equipamentos. Esses caracteres de controle tiveram sua origem nos primórdios da computação, quando eram utilizadas máquinas Teletype (como máquinas de escrever eletro-mecânicas), fitas de papel perfurado e impressoras de cilindro, portanto muitos deles são dirigidos a estes equipamentos. Por exemplo:

- o caráter 10 representa a função "LINE FEED", que faz com que uma impressora avance seu papel,

- o caráter 24 representa a função "cancel",

- o caráter 27 representa a função "escape" determinada pela tecla <ESC>, encontrada no canto superior esquerdo nos teclados.

Duas das mais importantes codificações atualmente utilizadas são:

. ISO: "International Standardization Organization". É o padrão ocidental, utilizado também no Brasil. (SEMELHANTE AO ASCII)
Cada caractere só possui 1 byte (8 bits), gerando um máximo de 256 caracteres.

. UTF-8: "Unicode Transformation Format-8". É o padrão mundial, que pode ser usado em quase todos os idiomas.

Cada caracter possui 2 bytes (16 bits), o que permite um valor máximo bem maior que o anterior: 65.536 caracteres.

Alguns Caracteres ASCII

Binário	Caractere
0100 0001	A
0100 0010	B
0110 0001	a
0110 0010	b
0011 1100	<
0011 1101	=
0001 1011	ESC
0111 1111	DEL

Como os principais códigos de representação de caracteres utilizam grupos de 8 bits por caractere, os conceitos byte e caractere tornam-se semelhantes, e as, palavras, quase sinônimas. O termo caractere é mais usado para fins comerciais e o termo byte é mais empregado na linguagem técnica de profissionais da área. Em JAVA isso já não mais ocorre: ela trabalha com os dois conceitos: BYTE(8bits) e CARACTER(16bits)

Indicações numéricas dos computadores:

Bit - 2 estados: 0 e 1

Byte	**B**	**8 bits**	
Quilobyte (ou Kilobyte)	**KB**	**1.024 bytes**	2^{10}=1.024
Megabyte	**MB**	**1.024 KB**	2^{20}=1.048.576
Gigabyte	**GB**	**1.024 MB**	2^{30}=1.073.741.824
Terabyte	**TB**	**1.024 GB**	2^{40}=1.099.511.627.776

Terabyte	**TB**	**1.024 GB**	**10**12 = 1.000.000.000.000**
Petabyte	**PB**	**1.024 TB**	**10**15 = 1.000.000.000.000.000**
Exabyte	**EB**	**1.024 PB**	**10**18= 1.000.000.000.000.000.000**
Zettabyte	**ZB**	**1.024 EB**	**10**21 = 1.000.000.000.000.000.000.000**
Yottabyte	**YB**	**1.024 ZB**	**10**24 = 1.000.000.000.000.000.000.000.000**

30 TB	7.500 pen drives de 4 GB
41 TB	Dados gerados em 1 dia pelo LHC, a máquina que reproduzirá o Big Bang
59 TB	12.553 filmes em DVD
83 TB	8,3 milhões de downloads do firefox 3.0 feitos no dia do seu lançamento

120 TB	Todas as imagens feitas pelo telescópio Hubble em 17 anos
136 TB	Arquivo digital com 17 milhões de livros (a biblioteca do Congresso Americano)
530 TB	TODOS OS VÍDEOS DO YOUTUBE

- Os computadores manipulam **dados** (sinais brutos e sem significado individual) para produzir **informações**.

- A conversão de dados em informações, e estas novamente em dados, é uma parte tão fundamental em relação ao que os computadores fazem que é preciso saber como a conversão ocorre para compreender como o computador funciona.

- Infelizmente os computadores não utilizam o nosso sistema de numeração: o sistema DECIMAL. Por que utilizamos esse sistema?

QUE NÚMEROS É ESTE?

10

É O NUMERO 1, OU O NÚMERO 2, OU O 3, OU O 4, OU 5, 6, 7, 8,.....?

Sistema de Numeração:

- Conjunto de símbolos utilizados para representação de quantidades e de regras que definem a forma de representação.
- Cada sistema de numeração é apenas um método diferente de representar quantidades. As quantidades em si não mudam; mudam apenas os símbolos usados para representá-las.
- A quantidade de algarismos disponíveis em um dado sistema de numeração é chamada de **base**.
- Representação numérica mais empregada: ***notação posicional***.

Embora os códigos de caracteres sejam úteis para representar dados textuais e números inteiros (0 a 9), eles não são úteis para números que possuem pontos fracionários, como 1,25. Para representar números com frações, bem como números extremamente grandes, por exemplo, os computadores utilizam a **notação de ponto flutuante** (a ser vista posteriormente).

Notação Posicional

- Valor atribuído a um símbolo *depende* 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

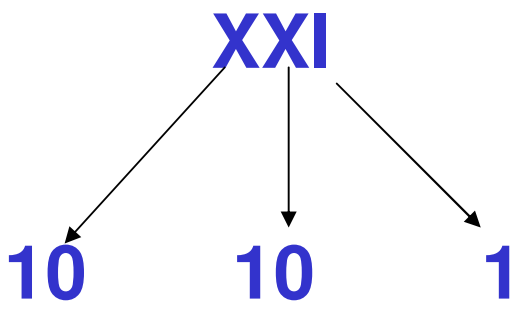
Notação Não Posicional

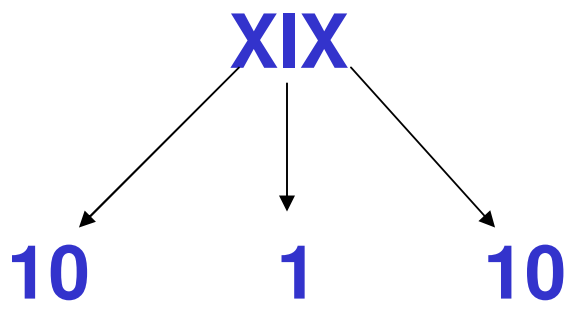
- Valor atribuído a um símbolo é *inalterável*, independe da posição em que se encontre no conjunto de símbolos que representam uma quantidade.

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 - **código**
- Operação básica – **contagem**
- Grupo com um determinado número de objetos – **base (raiz)**

- **Sistemas de numeração básicos:**
  - Decimal
  - Binário
  - Octal
  - Hexadecimal

Exemplos de Sistemas de Numeração

Sistema	Base	Algarismos
Binário	2	0,1
Ternário	3	0,1,2
Octal	8	0,1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Duodecimal	12	0,1,2,3,4,5,6,7,8,9,A,B
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

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 DECIMAL:

- Sistema mais utilizado.

- 10 símbolos para representar quantidades.

0 1 2 3 4 5 6 7 8 9

Peso – para representar quantidades maiores que a base.

- O Peso trouxe: **unidade**, **dezena**, (dez unidades), **centena** (cem unidades), **milhar** (mil unidades), **dezena de milhar**, **centena de milhar**, etc.

- **Exemplo**: 2574 é composto por 4 unidades, 7 dezenas, 5 centenas e 2 milhares, ou 2000 + 500 + 70 + 4 = 2574

SISTEMA DECIMAL:

- Sistema mais utilizado.

- 10 símbolos para representar quantidades.

0 1 2 3 4 5 6 7 8 9

Peso – representar quantidades maiores que a base.

- Peso trouxe: **unidade**, **dezena**, (dez unidades), **centena** (cem unidades), **milhar** (mil unidades), **dezena de milhar**, **centena de milhar**, etc.

- **Exemplo**: 2574 é composto por 4 unidades, 7 dezenas, 5 centenas e 2 milhares, ou 2000 + 500 + 70 + 4 = 2574

2*10**3 + 5*10**2 + 7*10**1 + 4*10**0

SISTEMA BINÁRIO

- 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

SISTEMA BINÁRIO

- 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

$$1*2**2 + 0*2**1 + 1*2**0 = 5$$

SISTEMA OCTAL:

- Utiliza 8 símbolos.
- **0 1 2 3 4 5 6 7**
- Exemplo: 563_8

SISTEMA OCTAL:

- Utiliza 8 símbolos.
- **0 1 2 3 4 5 6 7**
- Exemplo: $563_8 = 371_{10}$

5*8**2 + 6*8**1 + 3*8**0

HEXADECINAL

- Possui 16 símbolos (algarismos) para representar qualquer quantidade.

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

- Uso das letras - **facilidade de manuseio**.

- Exemplo: $5A3_{16}$

HEXADECINAL

- Possui 16 símbolos (algarismos) para representar qualquer quantidade.

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

- Uso das letras - **facilidade de manuseio**.

- Exemplo: $5A3_{16}$

$$5*16**2 + A*16**1 + 3*16**0 = 1443_{10}$$

Ao trabalhar com sistemas de numeração, em qualquer base, deve-se observar o seguinte:

- O número de dígitos usado no sistema é igual à base.

O maior dígito é sempre menor que a base.

O dígito mais significativo está à esquerda, e o menos significativo à direita.

Um “vai-um” de uma posição para outra tem um peso igual a uma potência da base.

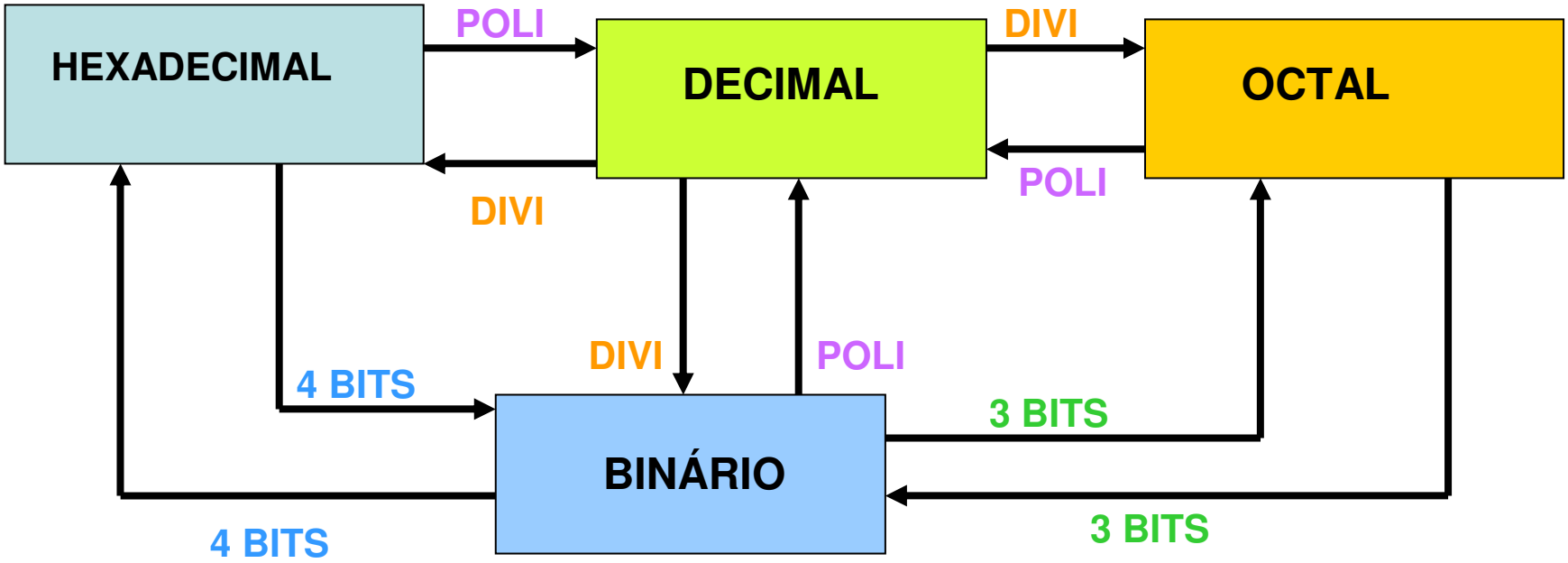
Em geral se toma a base decimal como referência.

DECIMAL	BINÁRIO	OCTAL	HEXA
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9

DECIMAL	BINÁRIO	OCTAL	HEXA
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13

CONVERSÕES ENTRE SISTEMAS

- Procedimentos básicos: - divisão
  (números inteiros) - polinômio
  - agrupamento de bits

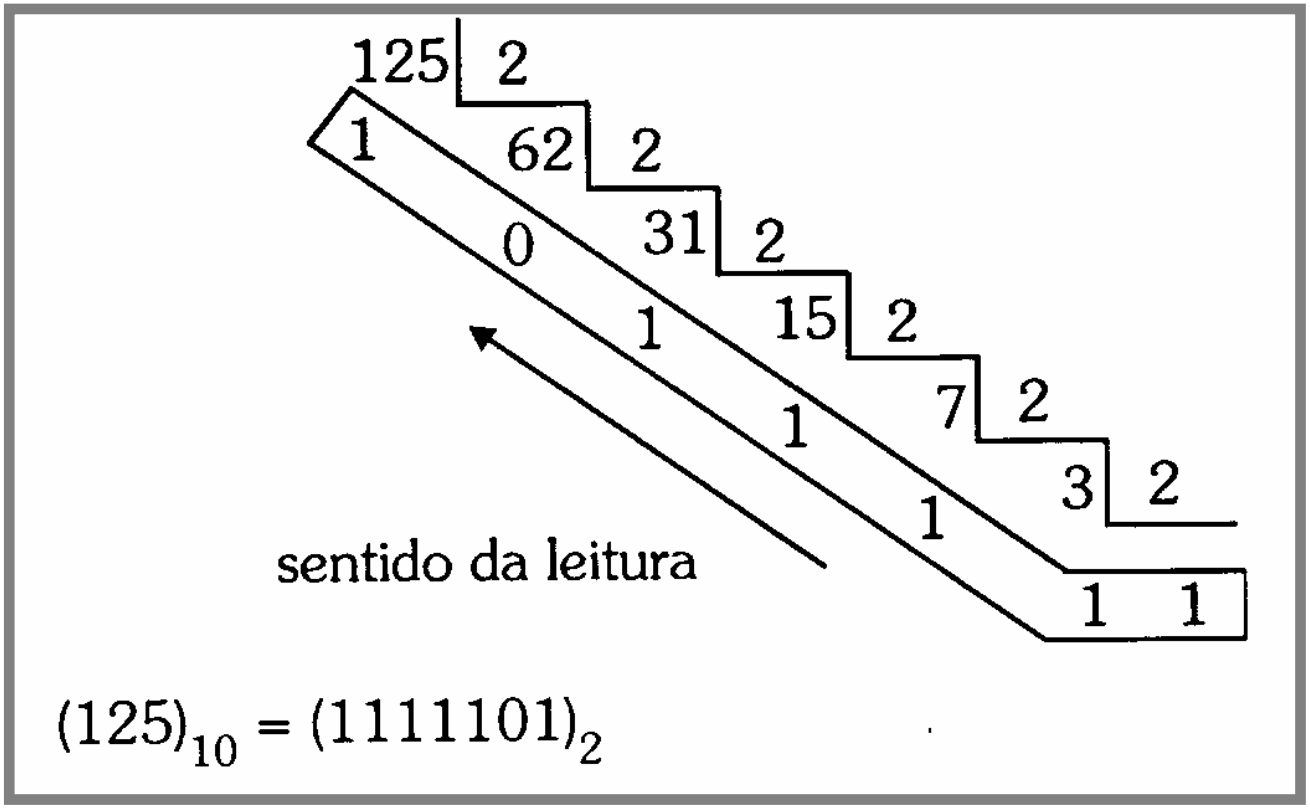


Divisão (Decimal >>> outro sistema)

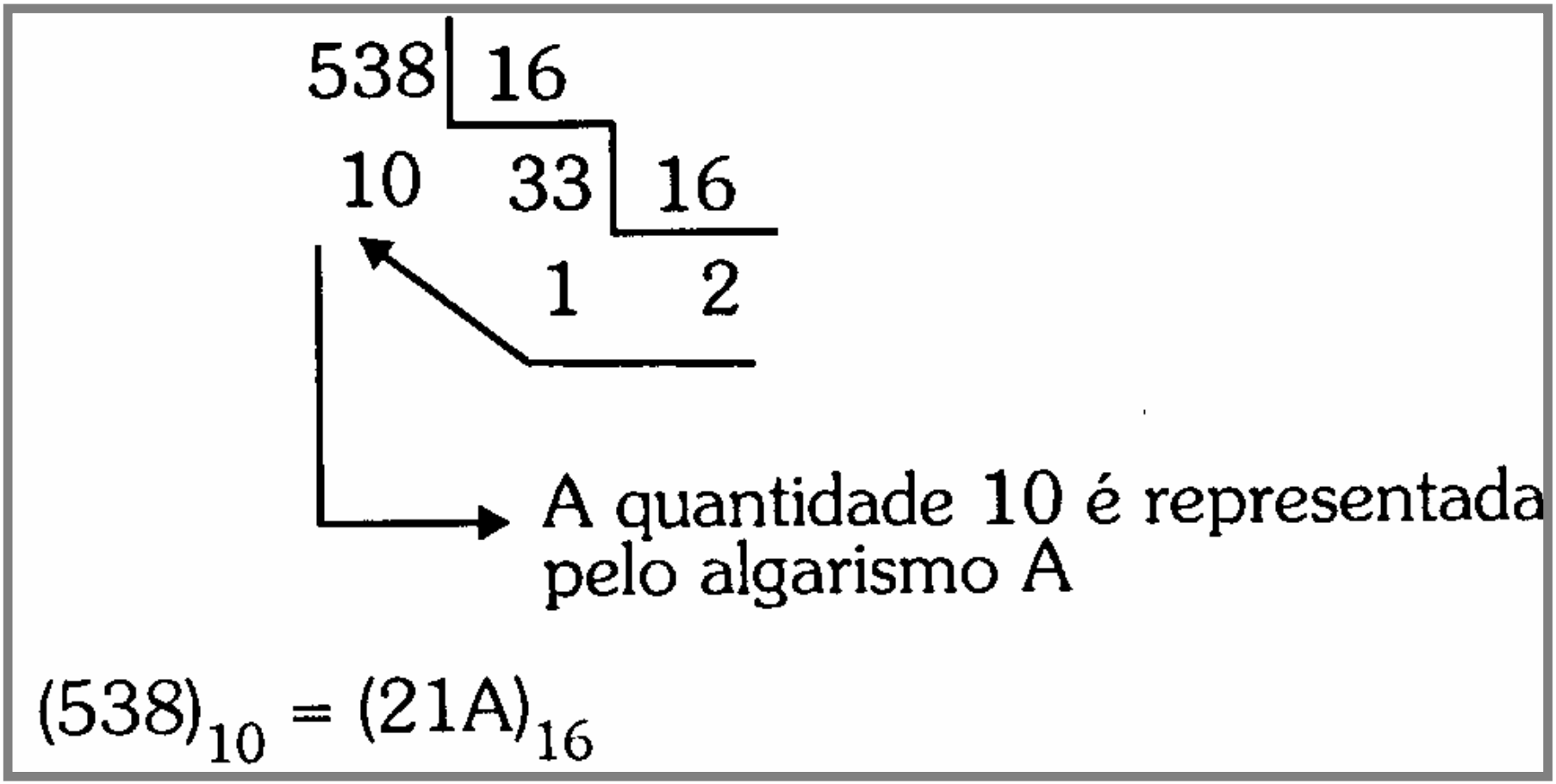
Divisão inteira sucessiva pela base do novo sistema, até que o resto seja menor do que a base.

Dividir o número original pela base b do novo sistema, assim como os quocientes obtidos, até que b > resto.

Ex: $(125)_{10}$ = $(?)_2$



$(538)_{10} = (?)_{16}$



$(538)_{10} = (21A)_{16}$

Notação Polinomial ou Posicional

(Válida para qualquer base numérica)

LEI DE FORMAÇÃO

(**Notação ou Representação Polinomial**):

a_n = algarismo, b = base do número

n = quantidade de algarismo - 1

NUMERO = $$a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \ldots + a_0 b^0$$

Ex.:

a) $(1111101)_2 = (?)_{10}$

(1111101) =

1x2**6 + 1x2**5 + 1x2**4 + 1x2**3 +
1x2**2 + 0x2**1 + 1x2**0 = 125_{10}

b) $(21A)_{16}$ = $(?)_{10}$

$(21A)_{16}$ =

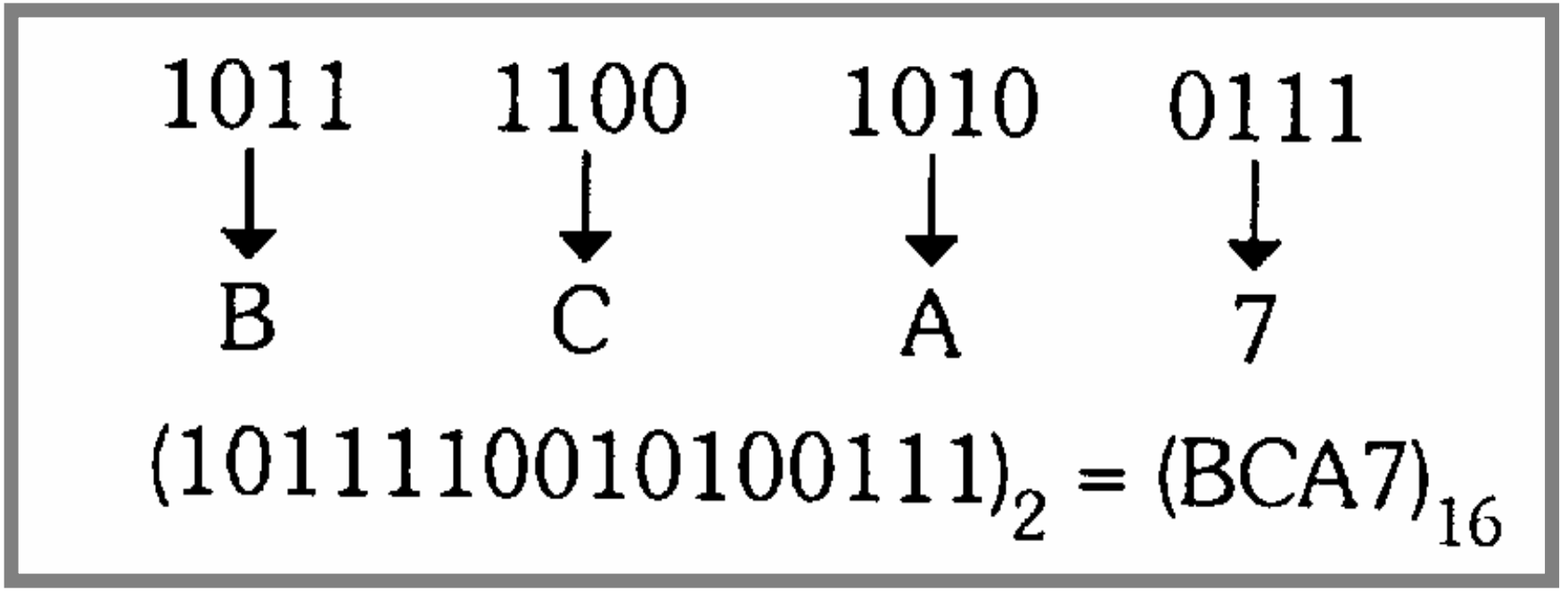
$2x16^2 + 1x16^1 + 10x16^0 = 538_{10}$

Agrupamento de Bits

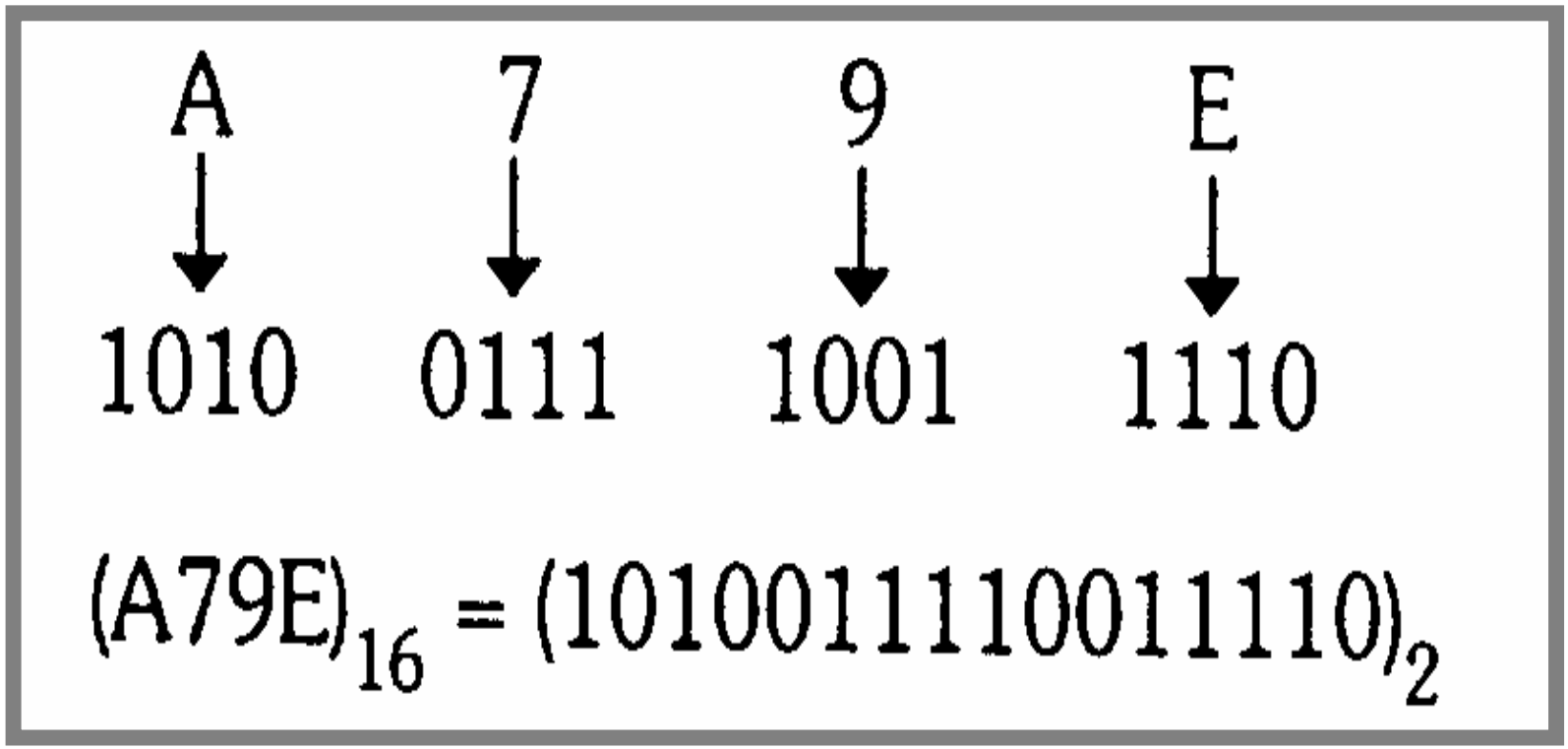
- Sistemas octal e hexa >>>> binário (e vice versa)
- associando 3 bits quando octal ou 4 bits quando hexadecimal, e vice-versa.

MUITO UTILIZADO: MIPS 32 BITS (BINÁRIO >> HEXA)

Ex.: $(1011110010100111)_2 = (\ ?\ )_{16}$



$(A79E)_{16} = (\ ?\ )_2$



$$(FFFF)_{16} = (\ ?\ )_2$$

(1111 1111 1111 1111)

Conversão octal >>>> hexadecimal

- Não é realizada diretamente - não há relação de potências entre as bases oito e dezesseis.

- Semelhante à conversão entre duas bases quaisquer - **base intermediária** (base binária)

- Conversão em duas etapas:

1 - número: base octal (hexadecimal) >>> binária.

2 - resultado intermediário: binária >>> hexadecimal (octal).

EX:

a) $(175)_8 = (\ ?\ )_{16}$

$(175)_8 = (1111101)_2 = (7D)_{16}$

b) $(21A)_{16} = (\ ?\ )_8$

$(21A)_{16} = (001000011010)_2 = (1032)_8$

Conversão de Números Fracionários

Lei de Formação ampliada (polinômio):

$$\text{Número} = \underbrace{a_n.b^n + a_{n-1}.b^{n-1} + a_{n-2}.b^{n-2} + \ldots + a_0.b^0}_{\text{parte inteira}} + \underbrace{a_{-1}.b^{-1} + a_{-2}.b^{-2} + \ldots + a_{-m}.b^{-m}}_{\text{parte fracionária}}$$

Exemplo: $(101,110)_2 = (\ ?\ )_{10}$

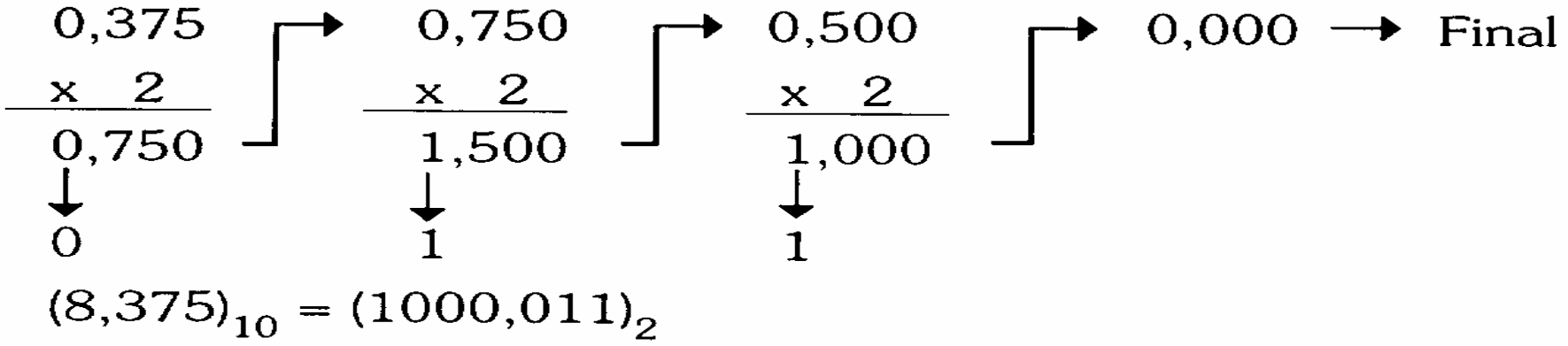
$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} = (5,75)_{10}$$

DECIMAL >>>> OUTROS SISTEMAS

Operação inversa: multiplicar a parte fracionária pela base até que a parte fracionária do resultado seja zero.

EXEMPLO: $(8,375)_{10} = (\ ?\ )_2$

- parte inteira: $(8)_{10} = (1000)_2$
- parte fracionária:



$(8,375)_{10} = (1000,011)_2$

Mostre que:

- $-5{,}8_{10} = 101{,}11001100..._{2}$ **(uma dízima).**

- $-11{,}6_{10} = 1011{,}10011001100..._{2}$

Obs: a vírgula foi deslocada uma casa para a direita, pois 11,6 = 2 x 5,8 .

Uma caixa alienígena com o número 25 gravado na tampa foi entregue a um grupo de cientistas. Ao abrirem a caixa, encontraram 17 objetos. Considerando que o alienígena tem um formato humanóide, quantos dedos ele tem nas duas mãos?

$17_{10} = 25_{\mathbf{b}}$

$17 = 2 \times b^1 + 5 \times b^0$

$17 = 2b + 5$

$b = (17-5)/2$

$b = 6$