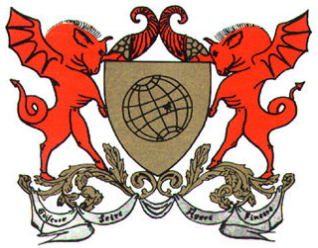
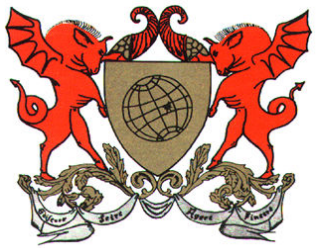


Universidade Federal de Viçosa
Campus Rio Paranaíba
Instituto de Ciências Exatas e Tecnológicas

SIN 110
Programação

Sistemas de Informação
Prof. Rodrigo Smarzaro
smarzaro@ufv.br

Slides criados pelo Prof. Guilherme C. Pena



Universidade Federal de Viçosa
Campus Rio Paranaíba
Instituto de Ciências Exatas e Tecnológicas

Aula de Hoje

Representação e Aritmética Binária

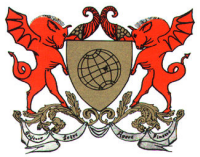
Introdução

ANÚNCIO

"Vende-se computador com processador Intel Core i5, 2.5 Ghz, **8 MB** Cache, Memória RAM de **6 GB**, HD de **1 TB**, placa de vídeo integrada de **750 MB**."

O que pode ser medido em um computador?

- Capacidade da Memória RAM
- Capacidade do HD
- Tamanho de arquivos
- Etc.



A informação e sua representação

O computador, sendo um equipamento eletrônico, armazena e movimenta as informações internamente sob forma eletrônica.

O computador reconhece dois estados físicos distintos, produzidos pela eletricidade:

- Presença de energia
- Ausência de energia

A informação e sua representação

Como os computadores representam as informações usando dois estados, eles são adequados para números binários:

- Desligado → 0
- Ligado → 1

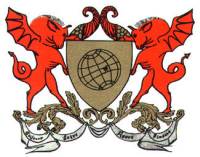
O computador é um sistema baseado em representação binária (base 2):

- 0 (zero) ou
- 1 (um)

A informação e sua representação

A razão pela qual os computadores usam o sistema binário (base 2) é porque isso torna mais fácil a implementação da tecnologia eletrônica atual.

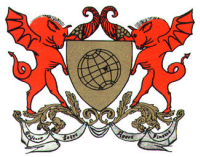
Obs: Até seria possível construir computadores que operassem na base decimal (base 10) que estamos acostumados, dígitos de 0 a 9. O problema é que esses computadores seriam **extremamente caros**.



A informação e sua representação

BIT

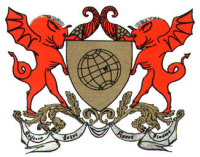
- Número binário no computador: “**B**inary dig**IT**”
- É a **menor unidade de informação**
- Um bit pode representar apenas **2 símbolos (0 e 1)**



A informação e sua representação

Um número de ***N bits*** pode representar ***2^N valores distintos***.

Bits	Símbolos (2^N)
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024



A informação e sua representação

BYTE (**B**inar**Y** **TE**rm)

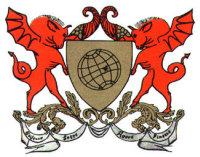
- **Grupo ordenado de 8 bits;**

- 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 através dos bytes que os representam:

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

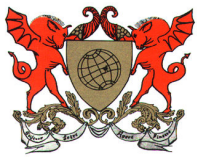


A informação e sua representação

Tabela ASCII:

Bin	Oct	Dec	Hex	Sinal
0100 0000	100	64	40	@
0100 0001	101	65	41	A
0100 0010	102	66	42	B
0100 0011	103	67	43	C
0100 0100	104	68	44	D
0100 0101	105	69	45	E
0100 0110	106	70	46	F

Bin	Oct	Dec	Hex	Sinal
0110 0000	140	96	60	`
0110 0001	141	97	61	a
0110 0010	142	98	62	b
0110 0011	143	99	63	c
0110 0100	144	100	64	d
0110 0101	145	101	65	e
0110 0110	146	102	66	f



A informação e sua representação

Para referenciar grandes volumes de dados, unidades foram criadas. Estas unidades representam **grandes agrupamentos de bits**:

Unidade	Símbolo	Tamanho	Bytes
Byte	B	8 bits	1
Quilobyte (ou Kilobyte)	KB	1.024 B	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

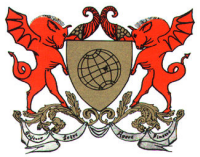
A informação e sua representação

Pode-se dizer que:

- 1 Kilobyte é aproximadamente MIL bytes
- 1 Megabyte é aproximadamente um MILHÃO de bytes
- 1 Gigabyte é aproximadamente um BILHÃO de bytes

E assim por diante..

Entretanto, há controvérsias...



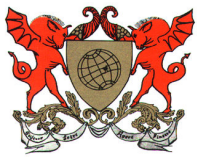
A informação e sua representação

Quando alguém diz:

“Este computador tem um HD de 500 gigas”

Ou seja, com 500 gigabytes, ele deveria poder armazenar aproximadamente 536.870.912.000 bytes”

Entretanto este HD tem exatamente 500.000.000.000 bytes, o que na prática resulta em 465,6613 GB



A informação e sua representação

A indústria/fabricantes trabalham com as unidades em **potência de 10**, enquanto a computação trabalha com **potência de 2**.

Nome	Símbolo	Múltiplo
Byte	B	2^0
kibibyte	KiB	2^{10}
mibibyte	MiB	2^{20}
gibibyte	GiB	2^{30}
tebibyte	TiB	2^{40}
pebibyte	PiB	2^{50}
exbibyte	EiB	2^{60}
zebibyte	ZiB	2^{70}
yobibyte	YiB	2^{80}

Nome	Símbolo	Múltiplo
byte	B	10^0
kilobyte	KB	10^3
megabyte	MB	10^6
gigabyte	GB	10^9
terabyte	TB	10^{12}
petabyte	PB	10^{15}
exabyte	EX	10^{18}
zettabyte	ZB	10^{21}
yottabyte	YB	10^{24}

A informação e sua representação

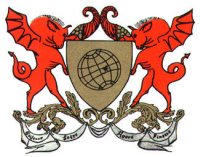
Faça o seguinte exercício:

- Abra o bloco de notas (Notepad) do Windows e adicione um texto de até 1024 caracteres, lembrando que o espaço em branco também é contado como um caracter.

- Grave o arquivo no HD com o nome teste.txt e verifique o tamanho do arquivo.

- Observe que o arquivo não pode passar de 1KB que equivale a 1024 bytes = 1024 caracteres.

- Em seguida, adicione mais um caractere ao arquivo e observe novamente o tamanho do arquivo.



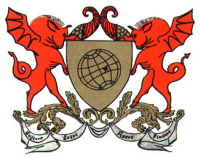
Sistema de Numeração

Sistema de Numeração

- Conjunto de símbolos utilizados para representação de quantidades

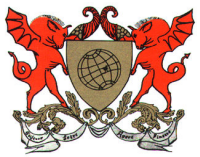
Cada sistema de numeração é um método diferente de representar quantidades

- As quantidades em si não mudam, mudam apenas os símbolos usados para representá-las



Sistema 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



Sistema Binário – Base 2

Utiliza dois símbolos para representar quantidades:

- 0 e 1

Cada algarismo é chamado de **bit**

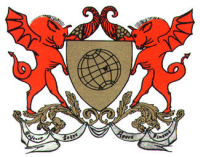
- Exemplo: 101_2

Caractere mais à esquerda - *Most-Significative-Bit* - "MSB".

- Em português (MSB) significa **"bit mais significativo"**

Caractere mais à direita - *Least-Significative-Bit* - "LSB".

- Em português (LSB) significa **"bit menos significativo"**

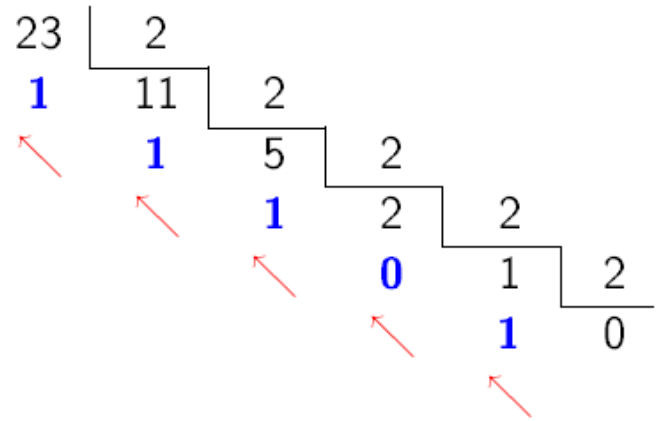


Conversão de Decimal para Binário

Método de Conversão:

Divida o número por 2 até que o quociente seja 0 (zero).

O número binário correspondente será formado pelos restos das divisões, sendo o resto da última divisão o dígito binário mais à esquerda (bit mais significativo):



Resultado: $(23)_{10} = (10111)_2$

Conversão de Binário para Decimal

Método de Conversão:

Multiplique cada bit *i* por 2^i começando do mais significativo.

Converta o número binário 10111 para decimal:

$(10111)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

$16 + 0 + 4 + 2 + 1 = (23)_{10}$

Converta o número binário 1111101 para decimal:

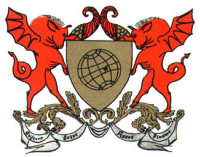
$(1111101)_2 = 1x2^6 + 1x2^5 + 1x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0$

$= (125)_{10}$

Conversão Decimal ↔ Binário

Exemplo: $(137)_{10} = (?)_2$

Exemplo: $(10110)_2 = (?)_{10}$



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

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

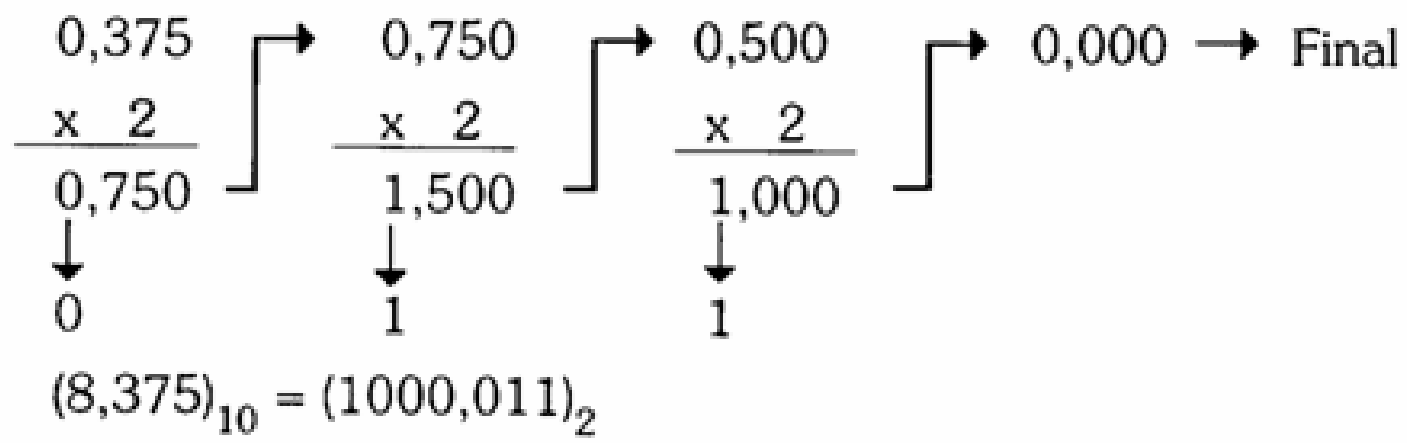
Conversão de Números Fracionários

Decimal → Outro sistema

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

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

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

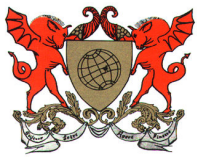


$(8,375)_{10} = (1000,011)_2$

Operações com Binários

Assim como no sistema decimal (base 10), o sistema binário possui as quatro operações elementares da aritmética:

- Adição
- Subtração
- Multiplicação
- Divisão



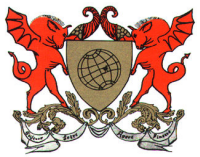
Operações - Adição

Binário

- Existe uma tabuada para números binários

- O resultado da adição entre dois números de um algarismo pode resultar em um número com um ou dois dígitos

  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 0 = 1
  - 1 + 1 = 0 (e "vai 1" para o dígito de ordem superior)
  - 1 + 1 + 1 = 1 (e "vai 1" para o dígito de ordem superior)



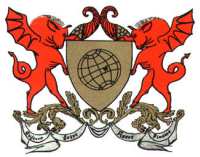
Operações - Adição

Binário

- Na soma, segue-se sempre a ordem das colunas da direita para a esquerda, tal como uma soma em decimal

- Exemplo: 011100 + 011010

```
  011100
+ 011010
 --------
  110110
```



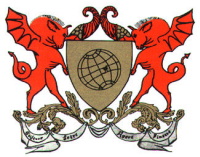
Operações - Subtração

Binário

- O resultado da subtração entre dois números de um algarismo resulta em nº com um dígito

- Tabuada da subtração
  - 0 - 0 = 0
  - **0 - 1 = 1** (e "vem um" do dígito de ordem superior)
  - 1 - 0 = 1
  - 1 - 1 = 0

- Estouro -> ***borrow*** (empréstimo) ou vem-um.



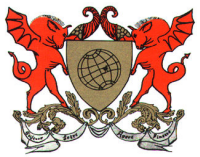
Operações - Subtração

Binário

- Como é impossível tirar 1 de zero, o artifício é "pedir emprestado" 1 da casa de ordem superior

- Quando o digito de ordem superior for 0, então procuramos pelo próximo digito de ordem superior, até que ele seja 1

- Após isso, este bit torna-se então 0 e a todos os bit pulados (bits de valor 0) damos o valor 1



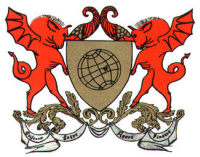
Operações - Subtração

Binário

- Na subtração, segue-se sempre a ordem das colunas da direita para a esquerda, tal como uma subtração em decimal atentando-se para a regra do empréstimo.

- Exemplos:

```
  11100     11011     11000100
- 01010   - 1101    -00100101
 --------  --------  ----------
  10010     01110     10011111
```

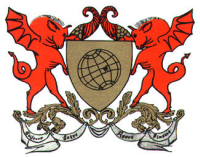


Operações - Multiplicação

Binário

- Tabuada da multiplicação
  - 0 x 0 = 0
  - 0 x 1 = 0
  - 1 x 0 = 0
  - 1 x 1 = 1

- A única diferença ao se realizar multiplicação em binários, em relação à multiplicação em decimal, é que a soma final deve ser feita em binário



Operações - Multiplicação

Binário

Na multiplicação, multiplica-se cada algarismo do número inferior por todo o número superior, tal como uma multiplicação em decimal, atentando-se para escrever o resultado a partir da mesma coluna do número que está multiplicando.

- Exemplo: 1011 x 1101

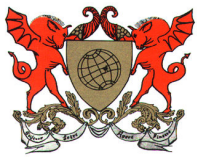
```
          1 0 1 1
        * 1 1 0 1
  -------------
          1 0 1 1
        0 0 0 0
      1 0 1 1
+   1 0 1 1
  -------------
  1 0 0 0 1 1 1 1
```

Operações - Divisão

Binário

- Pode ser feita de maneira idêntica à divisão decimal

- A diferença reside no fato das multiplicações e subtrações internas ao processo serem feitas em binário.



Operações - Divisão

Binário

- Algoritmo (Passos)

1) Na primeira vez, tome o bit mais significativo

2) Examine o novo dividendo e verifique se ele é maior ou igual que o divisor (Ir passo 3 ou 4)

3) Caso seja maior ou igual, coloca-se 1 no quociente e subtrai-se o divisor do dividendo em análise e se houver um próximo bit mais significativo, adicione no resto (Ir passo 5)

4) Caso contrário, seja menor, adiciona-se ao dividendo o próximo bit mais significativo, coloca-se 0 no quociente (Ir passo 5)

5) Este processo é realizado até que todos os bits do dividendo sejam processados (Ir passo 2 ou fim)

Operações - Divisão

Binário

- Exemplo 1: 100011 / 101

Operações - Divisão

Binário

- Exemplo 2: 11011 / 101

```
1 1 0 1 1 = 2 7
    1 0 1 = 5

    1 1 0 1 1   |1 0 1
  - 1 0 1       | 1 0 1
  -------
    0 0 1 1 1
      - 1 0 1
      -------
        0 1 0

o quociente é   1 0 1
e o resto é     1 0
```

Exercícios

1) Qual o decimal equivalente a $(11011011)_2$?

2) Qual o binário equivalente à sua idade?

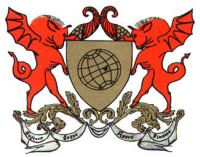
3) Converter os seguintes números decimais para números binários:

a) 39

b) 0,5625

c) 256,75

d) 129,625



Exercícios

4) Execute as seguintes operações

a) 0011 + 1110

b) 1110 – 0100

c) 10011 * 1101

d) 1101101/1011