



# Fundamentos da arquitetura de computadores

Sistemas de numeração e representação de dados

Prof. André Renato  
1º Semestre / 2012

# Sistemas de numeração

- Os sistemas de numeração estão na base da matemática.
- Eles são responsáveis por estabelecer a relação entre valores e símbolos.
- Sem eles fica muito difícil realizar operações simples como adição ou divisão.

# Sistemas de numeração

- Um sistema de numeração precisa definir:
  - Uma base numérica: base 10, base 2, base 16, base 134 etc.
  - Um conjunto de símbolos distintos (alfabeto), sendo um para cada valor da base: algarismos arábicos, alfabeto hexadecimal, {0, 1} etc

# Sistemas de numeração

- O sistema de numeração mais utilizado pelos humanos é o decimal (base 10, com os algarismos arábicos).
- Com o advento da computação digital, o sistema de numeração binário (ou simplesmente sistema binário) ganhou uma importância crucial, principalmente para os profissionais da área.

# Sistemas de numeração

- Por que o sistema binário em detrimento do sistema decimal?
- Quando as máquinas (computadores) passaram a ser dispositivos elétricos, começou a ficar complicado representar 10 tipos diferentes de correntes elétricas.
- Era muito mais simples representar apenas: passagem de corrente (circuito fechado) ou não (circuito aberto).

# Sistemas de numeração

- Cada valor é representado por combinação de símbolos 0 e 1, da esquerda para direita.
- Estes símbolos que compõem os números são chamados de bits.
- A cada 8 bits temos 1 byte.
- 1024 bytes equivalem a 1 Kilobyte (KB)
- 1024 KBs equivalem a 1 Megabyte (MB)
- 1024 MBs equivalem a 1 Gigabyte (GB)
- 1024 GBs equivalem a 1 Terabyte (TB)
- 1024 TBs equivalem a 1 Petabyte (PB)

# Sistemas de numeração

- O sistema binário se adequa muito bem às máquinas digitais, permitindo um processamento bastante eficiente.
- Todas as operações matemáticas básicas presentes no sistema decimal podem ser facilmente adaptadas para o sistema binário.
- Entretanto, primeiro é necessário entender como converter valores entre os dois sistemas.

# Sistemas de numeração

- Conversão decimal-binário:
  - Consiste em pegar o número decimal e realizar sucessivas divisões por 2, até que o quociente seja 0 (zero).
  - Depois, para formar o número em binário, são pegos todos os restos das divisões do final para o começo.
  - Exemplos:



- Conversão decimal-binário
- 13 para binário
- 1101
- <https://www.rapidtables.com/convert/number/decimal-to-binary.html>
- <https://www.binaryhexconverter.com/decimal-to-binary-converter>

# Sistemas de numeração

- Conversão binário-decimal:
  - Consiste em realizar o processo inverso, ou seja, multiplicações por potências de 2, seguida da soma de produtos.
  - Um jeito simples é possível através da arrumação dos símbolos da seguinte forma:

256	128	64	32	16	8	4	2	1
1	0	1	1	1	0	1	0	1

# Sistemas de numeração

- Adição de dois números:
  - O resultado da soma de dois algarismos binários é dado pela tabela a seguir:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10 \text{ (2, em decimal)}$$

- Na prática, o último resultado corresponde a deixar o algarismo 0 e aplicar o mundialmente famoso “vai 1”

# Sistemas de numeração

- É importante destacar que o resultado nunca poderá conter algarismos diferentes de 0 e 1.
- Adições de três ou mais valores podem ser feitas através de resultados parciais entre dois valores de cada vez.
- Exemplos de 2 e 3 parcelas:

# Sistemas de numeração

- Como fica a representação de números negativos em binário?
- Obviamente, o computador não pode armazenar  $-101$  (-5).
- O sinal de negativo precisa ser expresso na forma de um símbolo binário.

# Sistemas de numeração

- O primeiro esquema de representação chama-se Sinal e Magnitude (SM).
- Nele, o módulo do número é representado de forma equivalente.
- O sinal de negativo pode ser substituído pelo símbolo (bit) 1 colocado na primeira posição à esquerda.
- Ex: -13 (10001101, em 8 bits)

# Sistemas de numeração

- A representação em SM traz alguns pontos ruins:
  - Se usarmos 8 bits para representar números, precisaremos reservar 1 bit para o sinal, restando apenas 7 bits para o módulo do número.
  - O valor 0 (zero) pode ser representado de duas formas: 00000000 e 10000000 (-0).

# Sistemas de numeração

- Além disso, sempre que realizarmos uma operação aritmética, precisaremos saber se o resultado será positivo ou negativo para colocar ou não o símbolo **1** na frente dos outros bits.
- Embora seja uma representação válida, na prática SM é pouco utilizado.
- O mais comum é utilizar outra representação de números negativos.



# Sistemas de numeração

- Complemento:
  - Existem basicamente dois tipos de complementos para o sistema binário:
    - Complemento à base-1
    - Complemento à base
  - Como estamos falando de base 2, os complementos também são denominados: Complemento à 1 (C1) e Complemento à 2 (C2).

# Sistemas de numeração

- Para calcular o complemento CI, precisamos dos seguintes passos:
  - Converter o módulo do número em decimal para binário (já vimos isso);
  - Completar com 0 (zero) à esquerda até o número ter a quantidade de bits do problema em questão;
  - Calcular quanto falta em cada casa para alcançar o maior valor possível ( $\text{base} - 1 = 1$ );

# Sistemas de numeração

- Exemplo: -13 (com 8 bits)
  - Valor binário (13): 1101
  - Com 8 bits: 00001101
  - Quanto falta em cada casa para chegar a 1?
    - 11110010
  - Na prática, esta última etapa corresponde, **no sistema binário**, a inverter os símbolos 0 e 1.

# Sistemas de numeração

- O complemento à base-1 ainda apresenta um dos problemas do SM:
  - O valor do número 0 pode ser representado de duas formas: 00000000 e 11111111 (-0)
- Para evitar esta situação, o complemento à base (C2) é mais indicado.

# Sistemas de numeração

- Para calcular o complemento C2 de um número, precisamos calcular o complemento C1 e depois adicionar 1 ao resultado.
- É comum acontecer o “vai 1” no último bit (o mais significativo). Neste caso, o “vai 1” deve ser ignorado.
- Exemplo:

# Sistemas de numeração

- No complemento C2, não há duas forma de representar 0 (provar!!!).
- Assim, ele é o mais utilizado para representar números negativos, pois as etapas de inversão e soma do 1 são feitas muito rapidamente pelo computador.

# Sistemas de numeração

- A subtração de dois números em binário pode ser feita de duas maneiras:
  - A primeira é a forma tradicional que utilizamos em decimal.
  - Utilizamos a seguinte tabela:
$$\begin{array}{l} 0 - 0 = 0 \\ 1 - 0 = 1 \\ 0 - 1 = 1 \text{ (vem 1)} \\ 1 - 1 = 0 \end{array}$$

# Sistemas de numeração

- Exemplos:
  - $12 - 7 = \text{????}$
  - $34 - 6 = \text{????}$
  - $5 - 7 = \text{????}$



# Sistemas de numeração

- A segunda forma de realizar a subtração vem de uma propriedade da aritmética:
  - $A - B = A + (-B)$
- Para tanto, precisamos calcular o complemento C2 do subtraendo antes de realizar a operação desejada.

# Sistemas de numeração

- Exemplos:
  - $12 - 7 = \text{????}$
  - $34 - 6 = \text{????}$
  - $5 - 7 = \text{????}$

# Sistemas de numeração

- A multiplicação e a divisão em binário seguem um esquema semelhante ao utilizado em decimal, porém mais simples.
- Exemplos:
  - $17 \times 5 = 85$
  - $21 \times 4 = 84$
  - $85 / 5 = 17$
  - $84 / 21 = 4$

# Sistemas de numeração

- O sistema binário, como já foi dito, é bastante apropriado para máquinas digitais como o computador.
- Por ser utilizada uma base menor do que a decimal, normalmente os números em binários ocupam grandes sequências de 16 ou 32 bits cada.

# Sistemas de numeração

- Para seres humanos, estes números grandes são de difícil tratamento.
- Imagine um HD com apenas 200MB:
  - Se você é um programador de um aplicativo para recuperação de informações, pode ser interessante que você controle o acesso a cada pequena parte (byte) do disco.
  - Um disco de 200MB contém 209715200 bytes. Já imaginou este número em binário?

# Sistemas de numeração

- Para certas aplicações, é necessário ter outro sistema de numeração que permita números “menores” que os representados em binário.
- Um dos sistemas propostos foi o octal (base 8).
- A conversão de decimal para octal é feita de forma idêntica ao sistema binário.

# Sistemas de numeração

- Converter:
  - 12
  - 36
  - 1020

# Sistemas de numeração

- Adição e subtração podem ser feitas de forma análoga, lembrando que o resultado nunca poderá ter algarismo estranhos ao alfabeto do sistema de numeração.
- A partir de agora, fica mais importante a separação entre símbolo e valor.



# Sistemas de numeração

- Como seria o complemento à base – 1 (C7) para os números, com 8 algarismos:
  - 47
  - 125
  - 89
- E o complemento à base (C8)???

# Sistemas de numeração

- A multiplicação e a divisão devem ser feitas de forma semelhante.
- O computador não calcula estes resultados diretamente.
- Mesmo que seja feito um programa que trabalhe com números em octal, o computador os converterá para binário primeiro.

# Sistemas de numeração

- Para fazer a conversão octal – binário, devem analisar a relação entre as bases:
- 8 é equivalente a  $2^3$ ;
- Logo, são necessários três algarismos da base 2 para representar um valor da base 8;

# Sistemas de numeração

- Se pegarmos cada algarismo em octal e o representarmos por três algarismos binários, podemos fazer a conversão diretamente.
- Exemplo:
  - $(2431)_8 = 1305 = (????)_2$
  - $(101)_8 = 65 = (????)_2$
  - $(137)_8 = 95 = (????)_2$

# Sistemas de numeração

- Para fazer a conversão binário – octal precisamos pegar grupos de 3 bits, da direita para a esquerda, e formar um algarismo em octal com cada grupo.
- Exemplo:
  - $(10101011)_2 = 171 = (?????)_8$
  - $(111001)_2 = 57 = (?????)_8$
  - $(11101101011)_2 = 1899 = (?????)_8$

# Sistemas de numeração

- O sistema octal, embora “encurte” bastante o tamanho dos números apresenta uma característica ruim:
  - Grande parte das máquinas trabalha, desde o começo da era digital, com conjunto de 8 bits ou múltiplos;
  - 8, 16, 32, 64 não são divisíveis por 3
  - Logo a representação em octal não fica “redonda”

# Sistemas de numeração

- Passou-se a utilizar um sistema de numeração mais potente, chamado de sistema hexadecimal, com base 16.
- A tabela a seguir mostra a relação de valores e símbolos deste sistema:

Valor	Símbolo	Valor	Símbolo
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

# Sistemas de numeração

- A conversão decimal – hexadecimal se faz de forma semelhante;
- Devemos ter atenção ao valores maiores que 9 e seus respectivos símbolos;
- Exemplo:
  - 1234
  - 156
  - 256
  - 1099



# Sistemas de numeração

- A conversão hexadecimal – decimal se faz com o método das potências:
- Exemplo:
  - 12D
  - A3F
  - 7BC

# Sistemas de numeração

- A conversão hexadecimal – binário se faz transformando cada algarismo da base 16 em 4 algarismos binários (bits), pois  $16 = 2^4$ ;
- Exemplo:
  - A2DE
  - F459
  - 50AB

# Sistemas de numeração

- A conversão binário – hexadecimal se faz através de formação de grupos com 4 bits cada, começando pela direita;
- Exemplo:
  - 101010101111
  - 10111011
  - 1000011001111

# Sistemas de numeração

- A conversão de hexadecimal – octal não pode ser feita diretamente.
- Primeiro passamos para binário e depois para octal.
- O processo inverso também passa pelo sistema binário primeiro.
- Exemplo:
  - $(12DD)_{16} = (????)_8$
  - $(5637)_8 = (????)_{16}$

# Sistemas de numeração

- Como fica o complemento à base - 1 (C15) dos seguintes números em hexadecimal?
- 1372
- ADEF
- 340C
- E o complemento à base (C16)?

# Sistemas de numeração

- Adição e subtração podem ser feitos de forma semelhante às que foram vistas.
- Exemplos:
  - $1234 + ABCD$
  - $982 + 12F$
  - $45A1 - 12FF$
  - $513 - 4ED$

# Sistemas de numeração

- Multiplicações e divisões também podem seguir os métodos já vistos.
- Vale lembrar que o computador sempre fará as operações em binário.
- Ainda falta analisar como o computador trata números reais (fracionários).
- Como será representado 12,125, por exemplo?

# Sistemas de numeração

- Pela forma de conversão tradicional, podemos fazer as sucessivas divisões por 2 na parte inteira.
- A parte fracionária realiza um procedimento diferente.
- Vamos ver o número 12,125!
- Como fica a conversão binário – decimal?



# Sistemas de numeração

- A computação é empregado como ferramenta de diversas áreas da ciência, como física, química, biologia, matemática entre outras.
- Por vezes, é necessário trabalhar com valores muito grandes ou muito pequenos (distâncias entre corpos celestes, quantidade de átomos em um composto, tempo de acesso à memória, distância entre átomos de uma molécula).

# Sistemas de numeração

- Imagine se desejássemos somar  $N1$  e  $N2$ , sendo:
  - $N1 = 0,00000000000076$  e
  - $N2 = 2341890000000,0000000000$
- Quantos bits precisaríamos para representar o resultado?
- A imensa maioria seria de 0 (zeros).

# Sistemas de numeração

- Para tanto, usa-se em computação a chamada ***notação científica***, que recebe o nome de ***ponto flutuante***.
- O número 5 bilhões poderia ser escrito assim:
  - $+5 \times 10^9$
  - Se considerarmos a base 10 como valor constante da representação, precisaríamos de 3 bits para o valor da mantissa 5 (parte fracionária) e mais bits 4 para o expoente 9

# Sistemas de numeração

- Um mesmo número pode ser representado de diversas formas diferentes em notação científica:
  - $4 \times 10^2$
  - $40 \times 10^1$
  - $0,4 \times 10^3$
- Normalmente, utiliza-se a última forma, com a mantissa ente 1 e 0,1 – chamada de ***normalizada***.

# Sistemas de numeração

- Como trabalharemos em base binária, teremos valores do tipo:
- $5 = 101 = 0,101 \times 2^3$
- $12 = 1100 = 0,11 \times 2^4$
- $57 = 111001 = 0,111001 \times 2^6$

# Sistemas de numeração

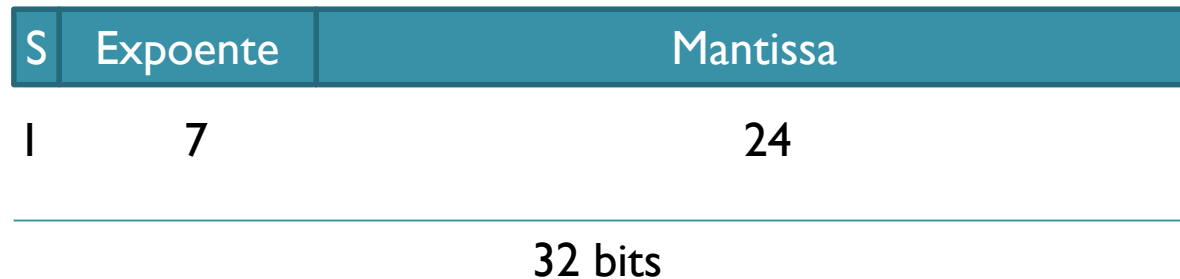
- Internamente, o sistema computacional deve armazenar 4 informações:
  - O sinal da mantissa
  - O valor da mantissa
  - O sinal do expoente
  - O valor do expoente

# Sistemas de numeração

- Além desses valores, outros aspectos precisam ser definidos:
  - A quantidade de bits utilizados;
  - O modo de representação da mantissa
  - O modo de representação do expoente
  - A quantidade de bits da mantissa e do expoente
  - A posição do sinal dos valores (se for o caso)
  - A base de exponenciação

# Sistemas de numeração

- Vejamos um exemplo:



S – sinal do número

Expoente: 1 bit para sinal e 6 para magnitude

Mantissa: normalizada

Base de exponenciação: 2



# Sistemas de numeração

- Exemplos (converter para o formato do slide anterior):
  - 407,375
  - -0,078125

# Sistemas de numeração

- As operações de adição e subtração de dois números em ponto flutuante requer, numa primeira etapa, o “alinhamento” das mantissa de forma que os expoente sejam iguais;
- Depois, realiza-se a soma ou subtração das mantissas, mantendo o expoente.
- Caso o resultado não esteja normalizado, este procedimento é executado.

# Sistemas de numeração

- Para multiplicação ou divisão, não há necessidade de alinhamento das mantissas.
- Realiza-se a operação desejada nas mantissas e soma-se ou subtrai-se os expoentes, se as operações forem de multiplicação ou divisão, respectivamente.

# Sistemas de numeração

- Já aprendemos como os números são armazenados e tratados pelo computador.
- Porém, a computação não envolve somente números. Letras e outros símbolos são muito utilizados para informar o usuário sobre algum evento do sistema, por exemplo.

# Sistemas de numeração

- Como o computador consegue armazenar letras???
- É possível realizar sucessivas divisões por 2 utilizando o caracter “\$”, por exemplo?
- Logicamente deve haver outra forma de representação.
- Uma das formas mais utilizadas é através da Tabela ASCII (*American Standard Code for Information Interchange*).

# Sistemas de numeração

- Nesta tabela, todos os símbolos imprimíveis, como A-Z, 0-9, @, #, \$, %, &, \*, +, -, /, \, {, }, ], [, entre outros estão representados.
- Além desses, outros símbolos não imprimíveis estão presentes.
- Cada símbolo possui um valor único. O símbolo A corresponde ao valor 65 (1000001, em binário), por exemplo.

# Sistemas de numeração

- Como o computador sabe que o valor `|00000|` será tratado como letra A e não número 65?
- Esta diferenciação é feita pelo programa que utiliza o dado.
- Se o programa mandar imprimir o valor como número, será tratado como tal.
- Se o programa manda escrever a mensagem na tela, será tratado como letra.

# ASCII TABLE

- <http://www.asciitable.com/>



# Sistemas de numeração

- Exemplo Pascal:
  - `var codigo:byte;`
  - `codigo := 65;`
  - `writeln(codigo);`
  - `writeln(chr(codigo));`