

SISTEMA DE NUMERAÇÃO

Um sistema de numeração é um sistema em que um conjunto de números são representados por numerais de uma forma consistente. Pode ser visto como o contexto que permite ao numeral "11" ser interpretado como o numeral **romano** para dois, o numeral **binário** para três ou o numeral **decimal** para onze. Em condições ideais, um sistema de numeração deve: representar uma grande quantidade de números úteis (ex.: todos os números inteiros, ou todos os números reais); dar a cada número representado uma única descrição (ou pelo menos uma representação padrão); e refletir as estruturas algébricas e aritméticas dos números.

SISTEMAS NUMÉRICOS

- Decimal
 - o sistema Decimal é o sistema mais utilizado pelos seres humanos, normalmente para indicar quantidades, e é constituído por dez algarismos: 0,1,2,3,4,5,6,7,8,9.
- Binário
 - O sistema binário é o sistema mais utilizado por máquinas, uma vez que os sistemas digitais trabalham internamente com dois estados (ligado/desligado, verdadeiro/falso, aberto/fechado). O sistema binário utiliza os símbolos: 0, 1, sendo cada símbolo designado por bit (binary digit)
- Octal
 - O sistema octal é um sistema de numeração de base 8, ou seja, recorre a 8 símbolos (0,1,2,3,4,5,6,7) para a representação de um determinado valor. O sistema octal foi muito utilizado no mundo da computação, como uma alternativa mais compacta do sistema binário, na programação em linguagem de máquina.
- Hexadecimal
 - Sistema de numeração muito utilizado na programação de microprocessadores, especialmente nos equipamentos de estudo e sistemas de desenvolvimento. Utiliza os símbolos: 0,1,2,3,4,5,6,7,8,9 do sistema decimal e as letras A,B,C,D,E,F. Equivalências: A=10, B=11, C=12, D=13, E=14 e F=15

Curiosidade: Incas usavam um sistema de base 10 parecido ao nosso, porém os Maias utilizavam um sistema de base 20 (incluindo o zero na contagem), o que indica que usavam os dedos das mãos e dos pés para criar sua base.

CONVERSÃO DE BINÁRIO PARA DECIMAL

Existem diversas técnicas para a realização desses cálculos e conversões. Apresentarei apenas as que considerar mais práticas e didáticas.

Convertendo no número 13 de base 10 para numeração de base 2.

$13 / 2 = 6$ resto **1**
 $6 / 2 = 3$ resto **0**
 $3 / 2 = 1$ resto **1**

1101

O resultado da última divisão, seguido dos restos das demais divisões formam o nosso número binário.

CONVERSÃO DE DECIMAL PARA BINÁRIO

Convertendo o número 1101 de base 2 para numeração de base 10.

1101

$$1 * 2^0 = 1 * 1 = 1$$

$$0 * 2^1 = 0 * 2 = 0$$

$$1 * 2^2 = 1 * 4 = 4$$

$$1 * 2^3 = 1 * 8 = 8$$

$$1 + 0 + 4 + 8 = 13$$

Multiplica-se cada bit por 2 elevado a sua respectiva casa decimal, somando-se os resultados.

ADIÇÃO BINÁRIA

A adição binária segue o mesmo princípio que utilizamos na adição dos números de base 10, com atenção ao limite da base binária que tem apenas dois dígitos.

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 0$$

$$1 + 1 = 0 \text{ e vai } 1 \text{ (transporte) ou } 10.$$

$$1 + 1 + 1 = 1 \text{ e vai } 1 \text{ (transporte) ou } 11.$$

Ex:

1111

1011

11010

SUBTRAÇÃO BINÁRIA

A subtração é apenas uma operação oposta a adição, ainda seguindo os mesmos princípios.

$$0 - 0 = 0$$

$$0 - 1 = 1 \text{ e vem } 1 \text{ (empresta)}$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Ex:

$$1000 - 111 =$$

Exercícios:

1) Efetue os seguintes cálculos

- a) $1000100 - 11 =$
- b) $10101 - 1010 =$
- c) $1111 + 111 =$
- d) $11010 + 10101 =$

2) Complete com $>$, $<$ ou $=$

- a) $10111 \text{ ___ } 11000$
- b) $47 \text{ ___ } 101111$
- c) $100110 \text{ ___ } 37$
- d) $111^2 \text{ ___ } 11^1$

MULTIPLICAÇÃO BINÁRIA

Segue o mesmo algoritmo da multiplicação em base 10, sendo facilitada por trabalhar apenas com 0 e 1.

$$0 \times 0 = 0$$

$$1 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 1 = 1$$

$$1 \times 1 \times 1 = 1$$

Ex

$$1001 \times 101 = 101101$$

DIVISÃO BINÁRIA

Após seu total entendimento, o algoritmo de divisão pode se assemelhar com uma das formas de se fazer divisão decimal.

$$1 / 1 = 1$$

EX:

$$1100/11 = 100$$

$$1011010 / 110 = 1111$$

$$11001/10 = 1100 \text{ (sobra 1, numeral fracionado)}$$

Exercício:

$$1001011 / 11 =$$

$$111100 / 110 =$$

NÚMEROS POSITIVOS E NEGATIVOS

Números sem sinal

A representação de números sem sinal em um computador aproveita todos os bits do número para representar quantidades: de 0 até $2^n - 1$ (2^n valores diferentes).

Por exemplo, um número de 8 bits pode armazenar números binários de 00000000 a 11111111 (de 0 a 255 em decimal). Isto representa a magnitude do número.

Exemplo:

$N = 8$ bits

Números sem sinal: $0 \leq X \leq 255$

Números positivos e negativos

A representação dos números positivos e negativos em um computador também permite representar quantidades em função do número de bits do número, entretanto, precisam reservar um bit para a representação do sinal (+ ou -). Isto é feito em geral acrescentando ao número um outro bit, chamado bit de sinal.

Quando trabalhamos com binários com sinal, somente podemos representar números com a metade da magnitude de um binário sem sinal, pois o bit mais significativo é reservado para o sinal, por exemplo:

$N = 8$ bits

Binário sem sinal: $0 \leq X \leq 255$

Binário com sinal: $-127 \leq X \leq 127$

OVERFLOW

O *overflow* ocorre quando o resultado de uma operação supera a capacidade do registro usado para guardar este resultado.

Para somas ou subtrações de números com sinal (+ ou -), o *overflow* ocorre caso o sinal do resultado não seja aquele que seria o esperado (por exemplo, um resultado negativo da soma de dois números positivos):

Exemplo

Soma dos números binários +9 com +8, ambos com 4 bits de magnitude e 1 bit de sinal:

+9 -> 0 1001

+8 -> 0 1000

1 0001 <- Magnitude incorreta

|

Sinal incorreto

BIT SIGNAL

O bit mais a esquerda (MSB) representa o **sinal**:

- 0 indica número **positivo**;
- 1 indica número **negativo**.

Binários negativos

Para representar os números **binários negativos** é necessário calcular o **complemento de 2** do número, em dois passos:

1. Calcula-se o **complemento de 1** do número (veja abaixo);
2. Soma-se **1** ao complemento de 1.
3. Despreza-se o transporte no bit mais significativo (chamado de **carry externo**), caso exista.

Complemento de 1

O **complemento de 1** de um binário é o simétrico dele, com todos os bits complementados, incluindo o bit de sinal.

Complemento de 1: Troque 0 por 1 e vice-versa.

Exemplos

Veja a forma de representar números decimais com sinal como números binário com sinal no método do **complemento de 2**, usando um total de 5 bits (incluindo o bit de sinal).

- **Número decimal +13**

É positivo, portanto é representado de forma direta:

$$13_{10} = 1101_2$$

Anexando o **sinal** o temos:

$$+13 = 01101$$

|

Bit de sinal

- **Número decimal -13**

É negativo, portanto sua magnitude deve ser representada na forma de **complemento de 2**:

$$13_{10} = 1101_2 \text{ (magnitude)}$$

Calculando o **complemento de 2**:

$$0010 \text{ (complemento de 1)}$$

$$+ 1$$

$$0011 \text{ (complemento de 2)}$$

Anexando o **sinal** o temos:

$$-13 = 10011$$

|

Bit de sinal