Aula 6 Sistemas Numéricos. Octal e hexadecimal. Aritmética binária

Programação de computadores

Sistemas octal e hexadecimal

- Em projetos e informática (isto é em trabalhos realizados por programadores, analistas e engenheiros) é usual representar quantidades usando sistemas de potência do binário (octal e hexadecimal principalmente).
- Objetivo: Reduzir o número de algarismos na representação para facilitar a compreensão da grandeza evitando erros.

Sistemas Octal e hexadecimal

- Octal
  - Cada três bits se representam apenas com um algarismo
  - De 0 a 7
  - $(7)_8$
- Hexadecimal
  - Hexa = 6 Deci = 10
  - Cada quatro bits um algarismo
  - 16 algarismos 0,..., 9, A, B, C, D, E, F
  - A,B,... Unidades de representação de 10-15
  - $(10)_H$ ou $(10)_{16}$ ou 10H ou 10h

Sistemas octal e hexadecimal

- Exemplo
- $(16)_{10} = 10000_2 = 20_8 = 10_{16}$
- $(9)_{10}$
- $(15)_{10}$
- $(5)_{10}$
- $(17)_{10}$

Conversão

- Para converter um número decimal numa base b usamos divisões sucessivas pela base b. Lendo o número ao contrario.
- N%b => N=q*b + r onde q, b e r são inteiros.
- Ex. $(16)_{10} = (X)_8 = (Y)_{16}$
- $(45)_{10}$
- $(8)_{10}$
- $(123)_{10}$

Binário, Octal e Hexa -> Decimal

- O processo que veremos a continuação serve para conversão de qualquer base a decimal. O processo se deriva da notação posicional comum a todos os sistemas posicionais de numeração de utilizam ordem.

Binário, Octal e Hexa -> Decimal

- Dado (XYZ,WK)B onde XYZWK são algarismos na base b

- Z é o algarismo de 1ra ordem
- Y é o algarismo de 2da ordem
- X é o algarismo da 3ra ordem da parte inteira.
- W e K são os algarismos da parte fracionaria.

Binário, Octal e Hexadecimal -> Decimal

- Podemos dizer que cada um destes algarismos é multiplicado por um peso que depende da posição em que se encontra e da base em que esta expresso o número, assim os pesos dos sistemas, ordenados serão sempre
- $\ldots b^4\ b^3 b^2 b^1 b^0 b^{-1} b^{-2} b^{-3} b^{-4} \ldots$
- E o número genérico XYZ,WK será
- $Xb^2 + Yb^1 + Zb^0 + Wb^{-1} + Kb^{-2}$ com $b^0 = 1$

Binário, Octal e Hexa -> Decimal

- Ex.
- $(1011011,10011)_2$ -> $(X)_{10}$
- $(13A,C)_H$ -> $(Y)_{10}$

Binário, octal e hexadecimal entre si

- Sendo 2, 8, 16 potencias de 2 as conversões entre os sistemas binários, octal e hexadecimal são imediatas.
- Binário => Octal
- $8 = 2^3$ Separa-se o número binário em grupos de 3 e se transformam na base 8
- Ex. $(10110101)_2$ => $(x)_8$
- 10-110-101 ->resolver

Binário, octal e hexadecimal entre si

- Binário => hexadecimal
- $16 = 2^4$
- Separa-se o binário em grupos de 4 algarismos
- Ex.
- $(11010101101)_2 = (x)_H$
- 110-1010-1101

Binário, octal e hexadecimal entre si

- Hexadecimal => binário
- Cada algarismo hexa gera a mesma grandeza em um grupo de 4 algarismos binários
- BF1H => $(x)_2$
- 1 = $(0001)_2$
- F = $(1111)_2$
- B = $(1011)_2$
- X =$(101111110001)_2$

Binário, octal e hexadecimal entre si

- A conversão de um número x na base genérica b1 para outra base b2 é definida através da conversão do primeiro número X_{b1} para a base 10 e da base 10 para a base b2.
- BF1H => $(X)_{10}$ => $(X)_2$ =>resolver
- B- b10-b2
- F-b10-b2
- 1-b10 –b2=>resolver

Aritmética binária

- Soma
- A tabuada da soma aritmética em binário é muito simples
- 0+0 = 0
- 0+1 = 1
- 1+0 = 1
- 1+1 = 0 -> vai 1 para o digito de ordem superior
- 1+1+1 = 1 -> vai 1 para o digito de ordem superior

Aritmética binária

- Ex.
- 011100 + 011010
- 101111 + 100010
- Subtração
- 0-0 = 0
- 0-1 = 1 -> vem 1 da ordem superior
- 1-0 = 1
- 1-1 = 0
- Como é impossível tirar 1 de 0 o artifício e “pedir emprestado” 1 da casa de ordem superior igual a subtração decimal

Aritmética binária

- Ex
- 11100-01010
- 00100-00011

Aritmética binária. Complemento de Base

- A implementação da subtração nos computadores é complexa. Requere numerosos testes assim nos computadores a subtração é feita através de um artifício. O método do “complemento de base” consiste em encontrar o complemento de um número em relação à base e depois somar. No caso dos computadores o complemento é achado na base 2 através do algoritmo:

Aritmética binária. Complemento de Base

- Se o número é positivo, mantenha o número (o complemento de um número positivo é ele mesmo)
- Se o número for negativo:
  - Inverta o número negativo ou o subtraindo da subtração (todo 1 vira 0 todo 0 vira 1)
- Some 1 ao complemento
- Some as parcelas (minuendo e subtraindo)

Aritmética binária. Complemento de Base

- Ex.

- 1101 – 1100 = 0001 -> resolver

Aritmética binária. Multiplicação

- 0x0 = 0
- 0x1 = 0
- 1x0 = 0
- 1x1 = 1
- No computador a multiplicação se faz por artifício. Para multiplicar A*n basta somar A n vezes. Ao dividir usamos subtrações sucessivas.
- QUALQUER OPERAÇÃO PODE SER REDUZIDA A OPERAÇÃO DE SOMA