



Arquitetura e Organização de Computadores

Aula 11

Centro Universitário 7
Setembro - Uni7
Sistemas de Informação

Prof. MSc Manoel Ribeiro

manoel@opencare.com.br

Representação de Dados



Representação dos Dados

- Um computador funciona por meio da execução sistemática de instruções que o orientam a realizar algum tipo de operação sobre valores (numéricos, alfanuméricos ou lógicos).
- Esses valores são genericamente conhecidos como dados

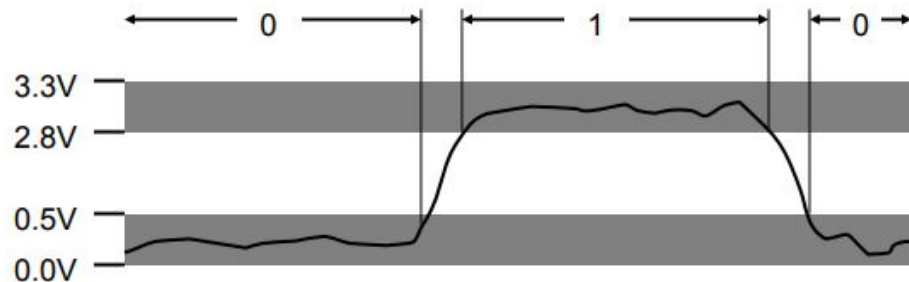
Representação binária

Representação binária

Exemplo: $15213_{10} = 11101101101101_2$

Vantagens:

- Implementação eletrônica
 - Possibilidade de armazenar elementos com dois estados
 - Transmissão eletrônica confiável (robustez a ruídos)
 - Implementação eficiente de operações aritméticas



Representação dos Dados

- $2017_{10} \rightarrow 11011001_2$
- $9,5_{10} \rightarrow 1001,0101_2$

Representação Binária

- IEEE 754: Variáveis de Ponto Flutuante
- Números inteiros sem sinal
- Números com sinal em complemento

IEEE 754 - características

- Padrão IEEE 754 para Aritmética Binária de Ponto Flutuante
- Recomendado pelos institutos ANSI (American National Standard Institute) e IEEE (Institute of Electrical and Electronic Engineers)
- Simplifica as operações de comparação e matemáticas

- Forma numérica

$$(-1)^s M 2^E$$

- Bit de sinal s determina se número é negativo ou positivo
- Mantissa M é um valor fracionário no intervalo $[1.0, 2.0)$, na representação normalizada.
- Expoente E

- Codificação



- bit mais significativo é s
- Campo exp codifica E
- Campo $frac$ codifica M

IEEE 754 - tamanhos

- Tamanhos
 - float: exp = 8 bits, frac = 23 bits, s = 1 bit
 - Total: 32 bits
 - Faixa de valores: 2^{-126} até 2^{127}
 - double: exp = 11 bits, frac = 52 bits, s = 1 bit
 - Total: 64 bits
 - Faixa de valores: 2^{-1022} até 2^{1023}
 - Precisão estendida: exp = 15 bits, frac = 63 bits, s = 1 bit
 - Total: 80 bits
 - Faixa de valores: 2^{-16382} até 2^{16383}

IEEE 754 - conversão float

- Convert número decimal para representação binária com parte fracionária
 - $9,5 \rightarrow 1001,0101_2$
- Ajustar para notação na forma de Mantissa e Expoente na base 2
 - $1001,0101_2 \rightarrow 1,0010101 \times 2^3$
- Fazer a padronização do expoente somando 127 ou 01111111_2
 - $3 \rightarrow 0011_2 + 01111111_2 \rightarrow \text{exp} = 10000010_2$
- Representar a fração sem o primeiro 1 com 23 bits
 - $\text{frac} = 00101010000000000000000$
- Representar o bit de sinal com 1 se negativo ou 0 positivo
 - $s = 0$
- IEEE754 - float
 - $9,5 \rightarrow 0.10000010.00101010000000000000000$

IEEE 754 - conversão float

- Convert número decimal para representação binária com parte fracionária
 - $9,4 \rightarrow 1001,0100_2$
- Ajustar para notação na forma de Mantissa e Expoente na base 2
 - $1001,0100_2 \rightarrow 1,0010100 \times 2^3$
- Fazer a padronização do expoente somando 127 ou 01111111_2
 - $3 \rightarrow 0011_2 + 01111111_2 \rightarrow \text{exp} = 10000010_2$
- Representar a fração sem o primeiro 1 com 23 bits
 - $\text{frac} = 00101000000000000000000$
- Representar o bit de sinal com 1 se negativo ou 0 positivo
 - $s = 0$
- IEEE754 - float
 - $9,4 \rightarrow 0.10000010.00101000000000000000000$

IEEE 754 - conversão float

- Convert número decimal para representação binária com parte fracionária
 - $95 \rightarrow 01011111_2$
- Ajustar para notação na forma de Mantissa e Expoente na base 2
 - $01011111_2 \rightarrow 1,011111 \times 2^6$
- Fazer a padronização do expoente somando 127 ou 01111111_2
 - $6 \rightarrow 0110_2 + 01111111_2 \rightarrow \text{exp} = 10000101_2$
- Representar a fração sem o primeiro 1 com 23 bits
 - $\text{frac} = 01111100000000000000000$
- Representar o bit de sinal com 1 se negativo ou 0 positivo
 - $s = 0$
- IEEE754 - float
 - $95 \rightarrow 0.10000101.01111100000000000000000$

IEEE 754 - conversão double

- [illegible]

IEEE 754 - Operações

- Comparação
 - Dado dois números IEEE a comparação é feita bit a bit da esquerda para direita
 - Após comparar o bit de sinal, o primeiro que apresentar um bit maior que o outro será maior
- Exemplos
 - 95 -> 0.10000101. 011111000000000000000000
 - 9,5 -> 0.10000010. 001010100000000000000000
 - 9,4 -> 0.10000010. 001010000000000000000000
 - -9,5 -> 1.10000010. 001010100000000000000000

IEEE 754 - Operações

- Soma
 - Dado dois números IEEE a é feita com seguintes passos
 - O expoente do resultado é o maior dos dois expoentes
 - Ajustar o número de menor expoente para o expoente do resultado
 - O sinal do resultado será o sinal do maior
 - Se sinais iguais: fração = soma das frações
 - Se sinais diferentes: fração = diferença das frações
- Exemplo
 - $9,4 \rightarrow 0.10000010.001010000000000000000000 + -9,5 \rightarrow 1.10000010.001010100000000000000000$
 - $R = 0.10000010.000000000000000000000000$

Implementação do IEEE 754

- Programe as seguintes funções para operações IEEE754

1. `public String ConvertToIEEE754(Float f)`
 - a. deve retornar um string com uma sequencia de 1's e 0's com a representação IEEE do número
2. `public boolean MaiorIEEE754(String n1, String n2)`
 - a. dado dois números na representação IEEE754 retorne true e o primeiro é maior que o segundo
3. `public String SomaIEEE754(String n1, String n2)`
 - a. dados dos número na representação IEEE754 retorne um novo número com a soma de n1 e n2

Fim