

GABARITO AP2 - Organização de Computadores 2016.1

Prova aplicada em : 29/05/2016

1. (2,0) Considere um sistema onde o número de ciclos de relógio para realizar uma operação de Entrada/Saída (E/S) por interrupção seja igual a 1000 e o processador utiliza um relógio de 1 GHz para executar as instruções. Determine o overhead (relação entre ciclos de CPU consumidos por segundo pelo processo de E/S e número de ciclos disponíveis) que ocorre quando se realiza uma operação de E/S com um disco rígido que transfere dados para o processador em blocos de 32 bytes e possui uma taxa de transferência de 32MB/segundo e que está ativo 15% do tempo total em que a CPU está sendo utilizada.

Taxa de transferência = 32MB/s = 32.000.000 B/seg.

Em cada operação é transferido um bloco de 32 bytes.

A cada segundo ocorrem: 32.000.000B / 32B = 1.000.000 operações/seg.

Como cada operação leva 1000 ciclos.

*O total de ciclos consumidos é igual a 1.000.000 operações/seg x 1.000ciclos/operação
= 1.000 x 10⁶ ciclos/seg.*

Clock = 1GHz = 1000 x 10⁶ ciclos/seg

O overhead será obtido pela seguinte razão:

= (1000 x 10⁶ ciclos/seg x 15% de atividade) / (1000 x 10⁶ ciclos/seg)

= 0,15 ou 15%

2. (2,0) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 12 bits. Na representação para ponto flutuante, o número a ser representado deve ser expresso na notação científica normalizada (+/- 1,b₀b₁b₂b₃b₄b₅b₆ × 2^{expoente})

O bit mais à esquerda representa o sinal e deve ser 0 para números positivos e 1 para números negativos. O expoente utiliza a representação em complemento a 2 e todos os expoentes possíveis de serem representados são utilizados para representar números expressos na notação científica normalizada.

a. (1,4) Suponha que o conteúdo dos 12 bits seja CE8₁₆ para este conjunto de bits quando considerarmos que ele está representando:

$$(CE8)_{16} = (110011101000)_2$$

i. um inteiro em complemento a 2. (0,4 pontos)

$$- 2^{11} + (2^{10} + 2^7 + 2^6 + 2^5 + 2^3) = -792$$

ii. um número inteiro representado em sinal e magnitude. (0,4 pontos)

$$- (2^{10} + 2^7 + 2^6 + 2^5 + 2^3) = -1256$$

iii. um número em ponto flutuante utilizando a representação do enunciado. (0,6 pontos)

$$1 \ 10011 \ 101000$$

Sendo:

$$\text{Sinal} = 1 \Rightarrow \text{negativo}$$

$$\text{Expoente} = 10011_2 = -2^4 + (2^1 + 2^0) = -13$$

$$\text{Mantissa} = 101000$$

$$\text{Temos então} \Rightarrow -1,101000 \times 2^{-13} = - (2^{-13} + 2^{-14} + 2^{-16}) = -0,000198$$

b. (0,6) Qual o menor e o maior valor positivos (na base 2) na representação em ponto flutuante para números normalizados, descrita no enunciado, para este computador?

$$\text{Menor valor positivo : } 0 \ 10000 \ 000000 = 1,000000 \times 2^{-16} = +0,00001526$$

$$\text{Maior valor positivo : } 0 \ 01111 \ 111111 = 1,111111 \times 2^{+15} = +65024$$

3. (2,0) Explique a diferença entre interpretação e compilação.

Tanto a interpretação como a compilação têm a função de tradução de programas escritos em linguagem de alto nível para linguagem de máquina. Na primeira, cada comando do código fonte é lido, convertido em código executável e imediatamente executado antes do próximo comando, já na segunda, uma tradução completa do programa de alto nível é armazenada em um arquivo chamado de programa objeto que poderá ser executado sem a necessidade do compilador.

4. (2,0) Considerando os diversos tipos de endereçamentos de instruções:

a) Projete um mecanismo de endereçamento que permita que um conjunto arbitrário de 256 endereços, não necessariamente contíguos, em um grande espaço de endereçamento, seja especificável em um campo de 8 bits.

Uma solução seria usar endereçamento por registrador base mais deslocamento. Por exemplo, teríamos 2 bits para especificar um registrador e 6 bits para especificar um deslocamento. Poderíamos, assim, usar 4 registradores, cada um com até 64 deslocamentos possíveis, fornecendo 256 endereços diferentes

b) Analise os modos de endereçamento direto e indireto, estabelecendo diferenças de aplicação, vantagens e desvantagens de cada um.

Direto: O campo operando contém o endereço do dado / Vantagem: Flexibilidade no acesso a variáveis de valor diferente em cada execução do programa / Desvantagem: Perda de tempo, se o dado é uma constante / Requer apenas um acesso à memória principal. Mais rápido que o modo indireto

Indireto: O campo de operando contém o endereço do dado / Vantagem: Manuseio de vetores (quando o modo indexado não está disponível). Usar como "ponteiro" / Desvantagem: Muitos acessos à MP para execução / Requer 2 acessos à memória principal.

5. (2,0) Crie um conjunto de instruções de um operando (o outro implícito na instrução é o registrador Acumulador, ACC), definidas em Linguagem Assembly, necessárias para a realização de operações aritméticas e elabore um programa para o cálculo da seguinte equação:

$$Y = ((B/A + B)*C) - A$$

OBS: Elabore o programa em Assembly de forma que os conteúdos dos endereços A, B e C sejam mantidos. Utilize endereços temporários, como T1, T2, etc.

I_ CONJUNTO DE INSTRUÇÕES PARA 1 OPERANDO:

ADD X	=> (ACC) <- (ACC) + (X)
SUB X	=> (ACC) <- (ACC) - (X)
MUL X	=> (ACC) <- (ACC) * (X)
DIV X	=> (ACC) <- (ACC) / (X)
LOAD X	=> (ACC) <- (X)
STORE X	=> (X) <- (ACC)

$$Y = ((B/A + B)*C) - A$$

LOAD B	=> (ACC) <- (B)
DIV A	=> (ACC) <- (ACC) / (A)
ADD B	=> (ACC) <- (ACC) + (B)
MUL C	=> (ACC) <- (ACC) * (C)
SUB A	=> (ACC) <- (ACC) - (A)
STORE X	=> (X) <- (ACC)