



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Organização de Computadores

GABARITO - AP2 2º semestre de 2009

Nome –

Assinatura –

Observações:

- a) Prova sem consulta e sem uso de máquina de calcular.
 - b) Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 - c) Você pode usar lápis para responder as questões.
 - d) Ao final da prova devolva as folhas de questões e as de respostas.
 - e) Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

1. (2,0) Considere um computador que utiliza 16 bits para armazenar dados.

- a) (0,5) Suponha que o conteúdo dos 16 bits expresso em hexadecimal seja $F400_{16}$. Indique o valor **em decimal (pode deixar as contas indicadas)** para este conjunto de bits quando considerarmos que ele está representando:

$$(F400)_{16} = (1111\ 0100\ 0000\ 0000)_2$$

a.1) (0,1) um inteiro sem sinal

$$2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{10} = 62464$$

a.2) (0,2) um inteiro em sinal e magnitude

$$-(2^{14} + 2^{13} + 2^{12} + 2^{10}) = -29696$$

a.3) (0,2) um inteiro em complemento a 2

$$-2^{15} + (2^{14} + 2^{13} + 2^{12} + 2^{10}) = -3072$$

Na representação para ponto flutuante utilizada por este computador, o número a ser representado deve ser expresso na notação científica normalizada:

$$N = (+/-)(1, b_1 b_2 b_3 \dots b_{11})_2 \times 2^{\text{expoente}}$$

O conjunto de 16 bits deve ser organizado da maneira descrita a seguir. O bit mais à esquerda representa o sinal e deve ser 0 para números positivos e 1 para números negativos. Como no padrão IEEE 754, quando todos os bits do expoente são iguais a 0 ou iguais a 1, os conjuntos de bits representam casos especiais e não representam números na notação científica normalizada. O expoente é representado com 4 bits em excesso de 7. A representação é mostrada na figura abaixo.

S	Expoente em excesso de 7	b ₁ b ₂ b ₃ ... b ₁₁
1	4 bits	11 bits

- b) (0,5) Indique o valor **em decimal (pode deixar as contas indicadas)** para o conjunto de bits acima (F400₁₆) quando considerarmos que ele está representando um número normalizado em ponto flutuante utilizando a representação acima.

1111 0100 0000 0000

Sinal: 1 (negativo)

Expoente = 1110 = 14 - 7 = 7

Mantissa = 100 0000 0000

Normalizado = - 1,100000000000 $\times 2^{+7}$ = 11000000,000 = - (2⁷ + 2⁶) = -192

- c) (0,5) Qual o menor e o maior valor de números expressos na notação científica **normalizada** que podem ser representados utilizando-se a representação em ponto flutuante, descrita no enunciado, para este computador? Os valores devem ser representados **em decimal (pode deixar as contas indicadas)**. Lembre-se que existem casos especiais.

Maior 0 1110 1111111111 = $+(1,1111111111)_2 \times 2^7 =$
 $+(2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4}) = + 255,9375$

Menor 1 1110 1111111111 = $-(1,1111111111)_2 \times 2^7 =$
 $2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} = - 255,9375$

- d) (0,5) Mostre a representação em ponto flutuante do valor em decimal +5,1 na representação do enunciado para ponto flutuante.

Convertendo para binário = 101,00011001100110011... => normalizando 1,01000110011001100 $\times 2^2$

Temos então:

Sinal = 0 (positivo)

Expoente = 2 + 7 = 1001

Mantissa = 01000110011

Resultado: 0 1001 01000110011

2. (2,0) Considere uma máquina cujo controlador de monitor de vídeo possui três registradores para se comunicar com o resto do sistema: o primeiro é utilizado para receber comandos do sistema (comando para indicar que se quer enviar um byte para o monitor de vídeo, por exemplo), o segundo para indicar o estado do monitor de vídeo (indicação de que o monitor de vídeo está pronto para receber um byte do sistema, por exemplo), e o terceiro onde o byte

recebido do sistema é armazenado no controlador. O primeiro registrador possui o endereço 30, o segundo 31 e o terceiro 32. O processo de envio de um byte para o monitor de vídeo consiste no envio do comando 01 para o registrador 30 do controlador do monitor, indicando que o sistema deseja enviar um byte para o monitor. O controlador do monitor recebe este comando e armazena o valor 01 no registrador de estado 31 quando o monitor de vídeo estiver pronto para receber um byte do sistema. Quando o monitor de vídeo estiver pronto para receber um byte, este byte deverá ser armazenado no registrador 32 pelo sistema. Descreva **detalhadamente** os três possíveis métodos de comunicação entre o controlador do monitor de vídeo com a unidade central de processamento e memória principal: por E/S programada, por interrupção e por acesso direto à memória. Você deve descrever estes métodos dentro do contexto do ambiente descrito neste enunciado. **Não serão consideradas descrições gerais dos três métodos.**

Resposta:

E/S programada:

A UCP envia o comando 01 para o registrador com endereço 30. Depois, fica lendo o conteúdo do registrador com endereço 31 e verificando se o seu conteúdo é igual a 01. Quando o conteúdo for igual a 01, a UCP envia, pelo barramento de dados, um byte para ser armazenado no registrador com endereço 32. Este procedimento é repetido até que todos os bytes tenham sido transferidos para o controlador do monitor.

E/S por interrupção:

A UCP envia o comando 01 para o registrador com endereço 30. Depois, vai executar outras instruções. Quando o controlador do monitor está pronto para receber um byte, ele envia um sinal de interrupção para a UCP. A UCP, ao receber o sinal de interrupção, finaliza a instrução que estiver executando, salva o contexto do programa que estava sendo executado e atende a interrupção, enviando o byte a ser enviado para ser armazenado no registrador de endereço 32. Este procedimento é repetido até que todos os bytes tenham sido transferidos para o controlador do monitor.

E/S por acesso direto à memória:

Neste caso, existe um controlador de DMA (Direct Access Memory) que é responsável pela transferência dos bytes. A UCP informa a este controlador que a operação de transferência deverá ser realizada com o monitor, o endereço inicial da memória onde se localizam os bytes a serem impressos, o número de bytes a serem transmitidos e a indicação de que o controlador deve transferir bytes da memória para o monitor. Após receber estas informações, o controlador realiza a transferência dos bytes entre memória e o monitor sem a intervenção da UCP. Após a transferência de todos os bytes, o controlador avisa o fim da operação para a UCP através de um sinal de interrupção.

3. (2,0) Explique, comparando:

a) Sistemas SMP e Sistemas NUMA

Sistemas SMP (ou UMA) têm como característica o acesso a todas as partes da memória principal com tempo de acesso uniforme. Em sistemas NUMA, todos os processadores possuem também acesso a todas as partes da memória principal podendo diferir o tempo de acesso em relação às posições da memória e processador.

Nos sistemas SMP o aumento no número de processadores tem como consequência problemas de tráfego no barramento comum degradando o desempenho. Uma solução para isto é a utilização de clusters, que tem, usualmente, como consequência alterações significativas na aplicação (software). Nos sistemas NUMA podem-se ter vários nós multiprocessadores, cada qual com seu próprio barramento, resultando em pequenas alterações na aplicação (software).

b) Modos de Compilação e Interpretação

A compilação consiste na análise de um programa escrito em linguagem de alto nível (programa fonte) e sua tradução em um programa em linguagem de máquina (programa objeto).

Na interpretação cada comando do código fonte é lido pelo interpretador, convertido em código executável e imediatamente executado antes do próximo comando.

A interpretação tem como vantagem sobre a compilação a capacidade de identificação e indicação de um erro no programa-fonte (incluindo erro da lógica do algoritmo) durante o processo de conversão do fonte para o executável.

A interpretação tem como desvantagem o consumo de memória devido ao fato de o interpretador permanecer na memória durante todo o processo de execução do programa. Na compilação o compilador somente é mantido na memória no processo de compilação e não utilizado durante a execução. Outra desvantagem da interpretação está na necessidade de tradução de partes que sejam executadas diversas vezes, como os loops que são traduzidos em cada passagem. No processo de compilação isto só ocorre uma única vez. Da mesma forma pode ocorrer para o programa inteiro, em caso de diversas execuções, ou seja, a cada execução uma nova interpretação.

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

- a) Projete um mecanismo de endereçamento que permita que um conjunto arbitrário de 128 endereços, não necessariamente contíguos, em um grande espaço de endereçamento, seja especificável em um campo de 7 bits.

Uma solução seria usar endereçamento por registrador base mais deslocamento. Por exemplo, teríamos 2 bits para especificar um registrador e 5 bits para especificar um deslocamento. Poderíamos, assim, usar 4 registradores, cada um com até 32 deslocamentos possíveis, fornecendo 128 endereços diferentes.

- b) Analise os modos de endereçamento base+deslocamento e indexado, estabelecendo diferenças de aplicação, vantagens e desvantagens de cada um.

*O **base mais deslocamento** tem como seu principal objetivo permitir a modificação de endereço de programas ou módulos destes (que é a relocação de programa), bastando para isso uma única alteração no registrador base. Já o **modo indexado** é utilizado para manipulação de estrutura de dados.*

*O **base mais deslocamento** tem como característica o endereço ser obtido da soma do deslocamento com o registrador base, diferindo do **modo indexado** onde o do registrador base é fixo e variar no deslocamento, ao contrário deste onde o deslocamento é fixo e com a alteração do registrador base permite-se a mudança do endereço.*

Em relação aos outros modos de endereçamento, os modos indexado e base mais deslocamento tem um desempenho melhor na execução das instruções de acesso aos dados (a alteração dos endereços é realizada na UCP). O modo indexado tem um desempenho melhor que o base + deslocamento devido a inclusão do registrador base no cálculo.

Vantagens do modo indexado: Foi criado principalmente para trabalhar com vetores, pois os valores destes são armazenados sequencialmente na memória e que a sua localização pode ser referenciado por ponteiro.

Vantagens do modo base + deslocamento: Criado com o propósito de reduzir o tamanho das instruções e facilitar o processo de realocação dinâmica de programas. Outra vantagem é que em vez de ser necessário que o campo operando tenha um tamanho correspondente à capacidade total de endereçamento da MP, basta que o endereço desejado seja obtido pela soma de um valor existente em um dos registradores da CPU com o valor contido na instrução.

5. (2,0) Crie um conjunto de instruções de dois operandos, definidas em Linguagem Assembly, necessárias para a realização de operações aritméticas e elabore um programa para o cálculo da seguinte equação:

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

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

CONJUNTO DE INSTRUÇÕES PARA 2 OPERANDOS:

```
ADD X,Y => X • X + Y
SUB X,Y => X • X - Y
MUL X,Y => X • X * Y
DIV X,Y => X • X / Y
MOV X,Y => X • Y
```

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

```
MOV T1,A => T1 • A
DIV T1,B => T1 • T1 / B
MOV T2,C => T2 • C
SUB T2,T1 => T2 • T2 - T1
MOV T1,B => T1 • B
DIV T1,T2 => T1 • T1 / T2
MOV T2,D => T2 • D
DIV T2,E => T2 • T2 / E
ADD T2,B => T2 • T2 + B
MUL T2,D => T2 • T2 * D
ADD T1,T2 => T1 • T1 + T2
ADD T1,C => T1 • T1 + C
MOV Y,T1 => Y • T1
```