

Nome –

Assinatura –

Observações:

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

1. (2,0) Considerando os diversos tipos de endereçamentos de instruções:

- a) Projete um mecanismo de endereçamento que permita que um conjunto arbitrário de 64 endereços, não necessariamente contíguos, em um grande espaço de endereçamento, seja especificável em um campo de 6 bits.

Resposta:

Não podemos usar endereçamento direto, pois com 6 bits representaríamos endereços de 0 a 63, contiguamente. Poderíamos usar endereçamento indireto, considerando que as células de endereços de 0 a 63 possuam endereços de memória não contíguos. Uma abordagem intermediária seria usar o endereçamento de registrador base mais deslocamento. No caso, se fossem usados, por exemplo, dois bits para registrador base e quatro bits para deslocamento, seria possível endereçar 4 blocos de memória, cada um com 16 endereços contíguos.

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

Resposta:

No endereçamento direto, o campo de operando contém o endereço do dado. Há flexibilidade no acesso às variáveis de valores diferentes em cada execução do programa, o que pode ser perda de tempo, se o dado é uma constante. Requer apenas um acesso à memória principal. Mais rápido do que o modo indireto. Já o modo indireto é útil no manuseio de vetores (quando o modo indexado não está disponível). Mas, para execução, requer dois acessos à memória principal.

2. (2,0) Compare os dois modos: compilação e interpretação. Indique em que circunstâncias um modo é mais vantajoso do que o outro.

Resposta:

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.

3. (2,0) Explique, comparando:

a) Computadores vetoriais e Computadores matriciais

Resposta:

Muitos problemas da física e engenharia envolvem operações sobre vetores. Frequentemente os mesmos cálculos são executados em muitos conjuntos diferentes de dados ao mesmo tempo. Neste contexto, existem dois métodos que têm sido usados para executar grandes problemas científicos rapidamente: Os Computadores vetoriais que correspondem a sistemas compostos por processadores vetoriais e frequentemente são associados à organizações de ULAs com pipeline de operações; e os Computadores matriciais que correspondem a sistemas compostos por processadores matriciais cuja organização é formada de ULAs paralelas.

b) Sistemas SMP e Sistemas NUMA

Resposta:

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).

c) Arquiteturas RISC e Arquiteturas CISC

Resposta:

RISC: Reduced Instruction Set Computer – Computador com um conjunto reduzido de instruções

CISC - Complex Instruction Set Computer: Computador com um conjunto complexo de instruções

CISC: Principais características:

Possui microprogramação para aumento da quantidade de instruções incluindo novos modos de endereçamento, de forma a diminuir a complexidade dos compiladores e em consequência permitir linguagens de alto nível com comandos poderosos para facilitar a vida dos programadores. Em contrapartida, muitas instruções significam muitos bits em cada código de operação, instrução com maior comprimento e maior tempo de interpretação

RISC: Principais características:

Menor quantidade de instruções e tamanho fixo. Não há microprogramação. Permite uma execução otimizada, mesmo considerando que uma menor quantidade de instruções vá conduzir a programas mais longos. Uma maior quantidade de registradores e suas utilizações para passagem de parâmetros e recuperação dos dados, permitindo uma execução mais otimizada de chamada de funções. Menor quantidade de modos de endereçamento com o objetivo da redução de ciclos de relógio para execução das instruções. Instruções de formatos simples e únicos tiram maior proveito de execução com pipeline cujos estágios consomem o mesmo tempo.

4. (2,0) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 16 bits. Na representação para ponto flutuante, o expoente está representado em complemento a 2, a mantissa é fracionária, a base implícita de representação é 2 e o bit de sinal é 0 para números positivos e 1 para números negativos. Os números a serem representados devem ter o formato como o utilizado no padrão IEEE 754, ou seja, um dígito 1 implícito à esquerda da vírgula, sendo que todos os padrões de bits representam números normalizados (não existem casos especiais, como no padrão IEEE 754). O formato desta representação está mostrado abaixo:

S	Expoente	Mantissa
1	4 bits	11 bits

- a) (0,2) Mostre a representação do inteiro -321 em complemento a 2 nesta máquina.

Resposta:

Para se obter a representação de -321, converte-se 321 para a base 2 utilizando-se 16 bits, inverte-se cada bit e soma-se 1. A conversão para a base 2 é igual a 0000000101000001, invertendo-se cada bit, obtém-se 1111111010111110 e somando-se 1, obtém-se a representação em complemento a 2 igual a 1111111010111111.

b) (0,4) Indique o valor **em decimal (pode deixar as contas indicadas)** para o conjunto de bits que você obteve acima, quando considerarmos que ele está representando:

i. um inteiro sem sinal

Resposta:

$$2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^7 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$$

ii. um inteiro representado em sinal e magnitude

Resposta:

$$-(2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^7 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0)$$

iii. um número em ponto flutuante

Resposta:

Sinal: Bit = 1, Negativo

Expoente: 1111 que representa um número em complemento a 2, logo expoente = -1

Mantissa: 1101011111

$$\text{Número: } -(1,1101011111)_2 \times 2^{-1} = (0,111010111111)_2 =$$

$$-(2^{-1} + 2^{-2} + 2^{-3} + 2^{-5} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-10} + 2^{-11} + 2^{-12})_{10}$$

c) (0,4) Indique o maior e o menor valor **em decimal (pode deixar as contas indicadas)** que pode ser representado pelos 16 bits para cada uma das representações abaixo:

i. inteiro sem sinal

Resposta:

$$\text{Menor: } 0, \text{ Maior: } 2^{16} - 1$$

ii. inteiro em complemento a 2

Resposta:

$$\text{Menor: } -2^{15}, \text{ Maior: } +(2^{15} - 1)$$

iii. número na representação ponto flutuante definida no enunciado

$$\text{Menor: } -(1,1111111111)_2 \times 2^7 = -(11111111,1111)_2 =$$

$$-(2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4})$$

$$\text{Maior: } +(1,1111111111)_2 \times 2^7 = -(11111111,1111)_2 =$$

$$+(2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4})$$

d) (0,4) Indique a representação em ponto flutuante dos seguintes valores decimais :

i. +21,625

Resposta:

$$+21,625 = +(10101,101)_2 = +(1,0101101)_2 \times 2^{+4}$$

Bit de sinal = 0

Mantissa = 01011010000

Expoente: +4 = 0100 em complemento a 2

Representação: 0010001011010000

ii. -0,1

Resposta:

$$-0,1 = -(0,000110011001100)_2 = -(1,10011001100)_2 \times 2^{-4}$$

Bit de sinal = 1

Mantissa = 10011001100

Expoente: -4 = inv(0100)+1=1100 em complemento a 2

Representação: 1110010011001100

- e) (0,6) Você deverá trocar a representação do expoente na representação em ponto flutuante acima para uma representação por excesso. Indique o excesso que você irá utilizar, explicando porque você escolheu este valor. Mostre a representação dos números do item d utilizando a representação em excesso que você escolheu.

Resposta:

Para se ter uma quantidade semelhante de números negativos e positivos que podem ser representados, deve-se escolher o excesso como $2^{\frac{\text{número de bits}}{2}}$ ou $(2^{\frac{\text{número de bits}}{2}} - 1)$. Como no caso o número de bits é igual a 4 podemos utilizar um excesso de 8 ou excesso de 7.

Utilizando-se excesso de 8 temos as seguintes representações:

+21,625

Expoente: +4. Para obter a representação em excesso de 8, soma-se 8 e obtém-se a representação binária do resultado: $+4+8 = 12$, 1100

Representação: 0110001011010000

-0,1

Expoente: -4. Para obter a representação em excesso de 8, soma-se 8 e obtém-se a representação binária do resultado: $-4+8 = 4$, 0100

Representação: 0010001011010000

Utilizando-se excesso de 7 temos as seguintes representações:

+21,625

Expoente: +4. Para obter a representação em excesso de 7, soma-se 7 e obtém-se a representação binária do resultado: $+4+7 = 11$, 1011

Representação: 0101101011010000

-0,1

Expoente: -4. Para obter a representação em excesso de 7, soma-se 7 e obtém-se a representação binária do resultado: $-4+7 = 3$, 0011

Representação: 0001101011010000

5. (2,0) Considere uma máquina cujo controlador de teclado possui dois registradores para se comunicar com o resto do sistema: um para receber comandos do sistema (pedido de leitura do byte referente a uma tecla pressionada, por exemplo) e para indicar se existe dado para ser enviado, e outro para armazenar o byte referente à tecla pressionada pelo usuário. O primeiro registrador possui o endereço 05 e o segundo 06. Para se obter o byte referente a uma tecla pressionada, deve-se enviar o comando 01 para o controlador de teclado, utilizando-se o endereço 05. O controlador de teclado armazena o valor 01 no registrador de estado (registrador 05) quando uma tecla foi pressionada pelo usuário e o byte referente a ela está armazenado no registrador 06. Descreva **detalhadamente** os três possíveis métodos de comunicação entre o controlador de teclado com a unidade central de processamento e memória principal: por E/S programada, por interrupção e por acesso direto à memória.

Resposta:

E/S programada: A Unidade Central de Processamento (UCP) envia um comando para o controlador de teclado pedindo um byte referente a uma tecla pressionada. O envio de comando pode ser realizado de duas maneiras. No caso de E/S mapeada em memória, a UCP executa uma instrução para escrever no endereço 05 da memória o conteúdo 01. No caso de E/S isolada, a UCP envia um comando de saída para o dispositivo cujo identificador é 5 com o conteúdo 01. Em seguida, a UCP testa se o controlador de teclado já possui o byte referente a uma tecla pressionada. Quando se utiliza memória mapeada, este teste é realizado através da leitura do conteúdo de memória referente ao endereço 5 e posterior comparação deste conteúdo com o valor

01. No caso de E/S isolada, é executado um comando de entrada para o dispositivo com identificador 5, e o conteúdo obtido é comparado com o valor 01.

Caso o controlador não possua o byte (indicado pelo conteúdo do registrador 5 ser diferente de 01), a UCP volta a ler e testar o estado do controlador de teclado. A UCP só para de executar a leitura e teste de estado, quando o controlador indicar que existe um byte pronto para ser lido. Quando isto ocorre, a UCP lê o byte armazenado no registrador 6 do controlador de teclado. Utilizando-se E/S mapeada em memória, esta leitura é realizada através de um comando de leitura do endereço 6 da memória. Para E/S isolada, a UCP executa um comando de entrada para o dispositivo com identificador 6.

Este procedimento deve ser executado com uma frequência suficiente para que nenhuma tecla pressionada seja perdida, ou seja, não seja transferida para a memória.

Interrupção: Da mesma maneira que no procedimento de E/S programada, a UCP envia um comando para o controlador de teclado indicando que deseja obter um byte. A diferença deste procedimento para E/S programada é que após o envio deste comando, a UCP não fica monitorando o controlador de teclado. Ela vai executar outras instruções. Quando o controlador de teclado possui um byte referente a uma tecla pressionada, ele envia um sinal de interrupção para a UCP. A UCP recebe este sinal e atende esta interrupção da seguinte maneira. A UCP finaliza a execução da instrução que estava sendo realizada no momento da interrupção, salva o contexto onde esta instrução estava sendo realizada (conteúdo de registradores, por exemplo) e obtém o byte do controlador de teclado da mesma forma que é realizada pela E/S programada, através da leitura do registrador 6. A UCP então envia novamente um comando para receber um byte e volta a executar outras instruções, só voltando a executar instruções para transferência de dados entre o controlador de teclado e a memória, quando o controlador a interromper novamente.

Acesso direto à memória: Neste caso, existe um controlador de acesso direto à memória (ADM ou DMA do inglês Direct Access Memory) responsável por realizar a transferência de bytes entre o controlador de teclado e a memória principal. A UCP indica para o controlador de DMA a operação que deve ser realizada por ele, através do envio de comandos que fornecem as seguintes informações: identificação do controlador de teclado como o dispositivo com o qual deverá ser realizada a transferência de dados, indicação que será realizada uma operação de leitura de dados do controlador de teclado, número de bytes que devem ser transferidos e endereço inicial da memória onde devem ser colocados os bytes obtidos do controlador de teclado. Após a programação do controlador de DMA, a UCP executa instruções não relacionadas a esta transferência. O controlador de DMA realiza a transferência dos bytes do controlador de teclado para a memória principal utilizando E/S programada ou interrupção, procedimentos já descritos nos itens anteriores. Quando todos os bytes tiverem sido transferidos do controlador de teclado para a memória, o controlador de DMA envia um sinal de interrupção para a UCP indicando que a transferência foi finalizada.