 background image

Fundação CECIERJ - Vice Presidência de Educação Superior a Distância  
**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Organização de Computadores**  
**AP2 1º semestre de 2010**

**Nome**

**Assinatura**

Observações:

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

1. (2,0) Considere um processador que possua as seguintes características:

Um registrador de 8 bits  
 Um registrador de 16 bits  
 Uma barra de dados de 8 bits  
 Uma barra de endereços de 16 bits

Defina instruções que permitam ao processador carregar em um registrador o conteúdo do endereço dado, adicionar a um registrador um valor especificado e carregar no registrador A o conteúdo da posição de memória apontada pelo registrador B. Descreva cada instrução e caracterize o tipo de endereçamento que ela utiliza. Com essas instruções faça um programa que permita carregar no registrador A o elemento de ordem 1F de uma tabela que começa na posição de memória de endereço 013D e que gasta 8 bits de memória para cada elemento.

*Instruções:*

*Seja X registrador qualquer de 8 bits e W registrador qualquer de 16 bits*

*LD-Mb X, Op => X <- Op*

*seja Op um valor de 8 bits  
 registrador X recebe o conteúdo Op  
 (modo imediato)*

*LD-MB W, Op => W <- Op*

*seja Op um valor de 16 bits  
 registrador W recebe o conteúdo Op  
 (modo imediato)*

background image

*LD-D X, (Op)  $\Rightarrow X \leftarrow (Op)$*

*seja Op um valor de 16 bits correspondente a um endereço de memória*

*O registrador X recebe o conteúdo que está no endereço de memória Op (modo direto)*

*ADD-Mb X, Op  $\Rightarrow X \leftarrow X + Op$  seja Op um valor de 8 bits*

*registrador X recebe o conteúdo Op somado ao conteúdo anterior de X (modo imediato)*

*ADD-MB W, Op  $\Rightarrow W \leftarrow W + Op$  seja Op um valor de 16 bits*

*registrador W recebe o conteúdo Op somado ao conteúdo anterior de W (modo imediato)*

*LD-IR X, W  $\Rightarrow X \leftarrow (W)$*

*seja W um registrador com conteúdo correspondente a um endereço de memória  
O registrador X recebe o valor contido no endereço armazenado por W (modo indireto por registrador)*

*LD-B X, Op  $\Rightarrow X \leftarrow (W + Op)$  Seja W um registrador com conteúdo*

*correspondente a um endereço de memória  
Seja Op um valor de 8 bits  
O registrador X recebe o valor contido no endereço indicado por W (base) acrescido do deslocamento Op (modo base mais deslocamento)*

### **Programa solução 1 (por base mais deslocamento)**

*Sejam:*

*Um registrador de 8 bits*

*Um registrador de 16 bits*

*LD-MB B, 013d  $\Rightarrow B \leftarrow$  013d Reg.B recebe valor 013D*

*LD-B A, 1f  $\Rightarrow A \leftarrow (B + 1f)$  Reg.A recebe conteúdo de (B + 1f)*

### **Programa solução 2 (por modo indireto)**

*Sejam:*

*Um registrador de 8 bits*


*Um registrador de 16 bits*

*LD-MB B, 013d  $\Rightarrow B \leftarrow$  013d Reg.B recebe valor 013D*

*ADD-MB B, 1f  $\Rightarrow B \leftarrow B + 1f$  Adiciona 1f ao reg.B*

*LD-IR A, B  $\Rightarrow A \leftarrow (B)$  Reg.A recebe conteúdo do endereço contido em B*

2. (2,0) Responda:

 background image

- a) Um programa em linguagem Assembly é diretamente executável pelo processador? Explique.

*Não. Porque o processador entende apenas a linguagem de máquina que é composta de 0 e 1s, a linguagem assembly representa a linguagem de máquina sob a forma de mnemônicos para que possa ser melhor compreendida ao ser humano.*

*Para ser executável, um programa elaborado em assembly deverá passar pelo processo de montagem, que consiste em traduzir cada instrução (mnemônicos) para seu respectivo em linguagem binária (máquina)*

- b) Um compilador deve ser específico para uma determinada linguagem de programação? E para uma determinada UCP? Explique.

*Sim. Também. Mesmo que as linguagens tenham em comum suas principais estruturas de programação (decisões, loops, procedimentos) diferenciam nas formas léxicas, sintáticas e semânticas, daí a necessidade de cada linguagem ter um compilador próprio. Ainda assim, mesmo que a linguagem a ser compilada seja a mesma, o compilador poderá diferenciar devido ao conjunto de instruções em linguagem de máquina do processador a qual foi destinado.*

*Outro fator relevante como motivador em diferenciar compiladores, mesmo direcionados a uma linguagem de programação comum e também processador, é o sistema operacional. Sistemas operacionais diferentes possuem chamadas também diferentes, chamadas estas que permitem o programa a ter acesso aos recursos de hardware, eventos relacionados aos processos como, criação, encerramento e comunicação entre processos, gerenciamento de memória, entre outros.*

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

*A diferença básica está na forma de tradução: a compilação faz sua tradução a partir de um programa escrito em linguagem de alto nível (programa fonte) para um programa em linguagem de máquina (programa objeto). Já na interpretação, cada comando do código fonte é lido pelo interpretador, convertido em código executável e imediatamente executado antes do próximo comando.*

*Além da diferença básica, seguem vantagens e desvantagens:*

*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) Considere um computador que utiliza 20 bits para armazenar dados.

- a) (0,5) Suponha que o conteúdo dos 20 bits expresso em hexadecimal seja D1C0016. Indique o valor **em decimal (pode deixar as contas indicadas)** para este conjunto de bits quando considerarmos que ele está representando:

$$D1C0016 = ( 1101 00011100 00000000 )_2$$

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

$$= 2^{19} + 2^{18} + 2^{16} + 2^{12} + 2^{11} + 2^{10} = 859.136$$

background image

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

$$= -(2^{18} + 2^{16} + 2^{12} + 2^{11} + 2^{10}) = -334.848$$

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

$$= -2^{19} + (2^{18} + 2^{16} + 2^{12} + 2^{11} + 2^{10}) = 189.440$$

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

$$N = (+/-)(1, b_{-1} b_{-2} b_{-3} \dots b_{-14})_2 \times 2^{\text{expoente}}$$

O conjunto de 20 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 5 bits em excesso de 16. A representação é mostrada na figura abaixo.

S	Expoente em excesso de 16	$b_{-1} b_{-2} b_{-3} \dots b_{-14}$
1 bit	5 bits	14 bits
bit		

- b) (0,5) Indique o valor **em decimal (pode deixar as contas indicadas)** para o conjunto de bits acima ( $D1C00_{16}$ ) quando considerarmos que ele está representando um número normalizado em ponto flutuante utilizando a representação acima.

$$D1C00_{16} = (1101\ 00011100\ 00000000)_2$$

$$\text{sinal} = 1 = \text{negativo}$$

$$\text{expoente} = 10100 = 20_{10} = +4 \text{ (por excesso de 16)}$$

$$\text{mantissa} = 01110000000000$$

$$N = -(1,01110000000000)_2 \times 2^{-4} = -10111_2 = -23_{10}$$

- 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.**

$$\text{Representação do maior valor: } 0\ 11110\ 111111111111 \Rightarrow 1,111111111111 \times 2^{+14} = +32768$$

$$+14 = +2 \quad +15 =$$

$$\text{Representação do menor valor: } 1\ 11110\ 111111111111 \Rightarrow -1,111111111111 \times 2^{+14} = -32768$$

$$+14 = -2 \quad +15 =$$

- d) (0,5) Mostre a representação em ponto flutuante do valor em decimal +0,2 na representação do enunciado para ponto flutuante.

background image

$+0,2 = 0,00110011001100110011... = 1,100110011001100110011... \times 2^{-3}$   
 Sinal  $\Rightarrow 0 = \text{positivo}$   
 Expoente  $\Rightarrow -3 + 16 = 13 \text{ (01101)}$   
 Mantissa  $\Rightarrow 10011001100110$   
 Resposta: 0 01101 10011001100110

5. (2,0) Considere uma máquina cujo controlador de teclado 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 receber um byte do teclado, por exemplo), o segundo para indicar o estado do teclado (indicação de que o controlador do teclado está pronto para enviar um byte para o sistema, por exemplo), e o terceiro onde o byte a ser enviado para o sistema é armazenado no controlador. O primeiro registrador possui o endereço 50, o segundo 51 e o terceiro 52. O processo de recebimento de um byte do teclado consiste no envio do comando 01 para o registrador 50 do controlador do teclado, indicando que o sistema deseja receber um byte do teclado. O controlador do teclado recebe este comando e armazena o valor 01 no registrador de estado 51 quando o controlador de teclado tiver um byte pronto para ser enviado para o sistema. O sistema verifica o registrador 51 para detectar quando o controlador de teclado possui byte a ser acessado. Quando o registrador 51 indica que há byte no controlador de teclado, o sistema obtém o byte no registrador 52. 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. Você deve descrever estes métodos dentro do contexto do ambiente descrito neste enunciado. **Não serão consideradas descrições gerais dos três métodos.**

*E/S programada:*

A UCP envia o comando 01 para o registrador com endereço 50. Depois, fica lendo o conteúdo do registrador com endereço 51 e verificando se o seu conteúdo é igual a 01. Quando o conteúdo for igual a 01, a UCP efetua a operação de leitura do byte armazenado no registrador com endereço 52. Este procedimento é repetido para receber cada byte gerado durante o processo de transferência de dados com o controlador de teclado.

*E/S por interrupção:*

A UCP envia o comando 01 para o registrador com endereço 50. Depois, vai executar outras instruções. Quando o controlador de teclado detecta que existe um byte correspondente a tecla acionada, 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, efetuando a leitura do byte armazenado no registrador de endereço 52. Este procedimento é repetido para receber cada byte gerado durante o processo de transferência de dados do controlador de teclado.

*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 controlador de teclado, o endereço inicial da memória onde devem ser armazenados os bytes obtidos do teclado, o número de bytes a serem recebidos e a indicação de que o controlador de DMA deve transferir bytes do controlador de teclado para a memória. Após receber estas informações, o controlador de DMA realiza a transferência dos bytes entre controlador de teclado e memória sem a intervenção da UCP. Após a transferência de todos os bytes, o controlador de DMA avisa o fim da operação para a UCP através de um sinal de interrupção.



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Organização de Computadores**

**ERRATA – Gabarito AP2 1º semestre de 2010**

4. (2,0) Considere um computador que utiliza 20 bits para armazenar dados.

- a) (0,5) Suponha que o conteúdo dos 20 bits expresso em hexadecimal seja  $D1C00_{16}$ . Indique o valor **em decimal (pode deixar as contas indicadas)** para este conjunto de bits quando considerarmos que ele está representando:

$$D1C00_{16} = (1101\ 00011100\ 00000000)_2$$

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

$$= 2^{19} + 2^{18} + 2^{16} + 2^{12} + 2^{11} + 2^{10} = 859.136$$

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

$$= -(2^{18} + 2^{16} + 2^{12} + 2^{11} + 2^{10}) = -334.848$$

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

$$= -2^{19} + (2^{18} + 2^{16} + 2^{12} + 2^{11} + 2^{10}) = -189.440$$

- 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.**

Representação do *maior valor*:  $0\ 11110\ 111111111111 \Rightarrow 1,111111111111 \times 2^{+14} = +(2^{+15}-1) = +32767$

Representação do *menor valor*:  $1\ 11110\ 111111111111 \Rightarrow -1,111111111111 \times 2^{+14} = -(2^{+15}-1) = -32767$