



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Organização de Computadores

AP2 2º semestre de 2011.

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) Crie um conjunto de instruções de um operando, definidas em Linguagem Assembly, necessárias para a realização de operações aritméticas e elabore um programa para o cálculo da seguinte equação:

$$X = A + (B * (C - A))$$

CONJUNTO DE INSTRUÇÕES PARA 1 OPERANDO:

<i>ADD X</i>	$\Rightarrow (ACC) \leftarrow (ACC) + (X)$
<i>SUB X</i>	$\Rightarrow (ACC) \leftarrow (ACC) - (X)$
<i>MUL X</i>	$\Rightarrow (ACC) \leftarrow (ACC) * (X)$
<i>DIV X</i>	$\Rightarrow (ACC) \leftarrow (ACC) / (X)$
<i>LOAD X</i>	$\Rightarrow (ACC) \leftarrow (X)$
<i>STORE X</i>	$\Rightarrow (X) \leftarrow (ACC)$

SOLUÇÃO:

<i>LOAD C</i>	$\Rightarrow (ACC) \leftarrow (C)$
<i>SUB A</i>	$\Rightarrow (ACC) \leftarrow (ACC) - (A)$
<i>MUL B</i>	$\Rightarrow (ACC) \leftarrow (ACC) * (B)$
<i>ADD A</i>	$\Rightarrow (ACC) \leftarrow (ACC) + (A)$
<i>STORE X</i>	$\Rightarrow (X) \leftarrow (ACC)$

OUTRA SOLUÇÃO:

<i>LOAD C</i>	$\Rightarrow (ACC) \leftarrow (C)$
<i>SUB A</i>	$\Rightarrow (ACC) \leftarrow (ACC) - (A)$
<i>STORE X</i>	$\Rightarrow (X) \leftarrow (ACC)$
<i>LOAD B</i>	$\Rightarrow (ACC) \leftarrow (B)$
<i>MUL X</i>	$\Rightarrow (ACC) \leftarrow (ACC) * (X)$
<i>STORE X</i>	$\Rightarrow (X) \leftarrow (ACC)$

$LOAD\ A \Rightarrow (ACC) \leftarrow (A)$
 $ADD\ X \Rightarrow (ACC) \leftarrow (ACC) + (X)$
 $STORE\ X \Rightarrow (X) \leftarrow (ACC)$

2. (2,0) Um processador possui um conjunto de instruções que emprega vários modelos de endereçamento, entre os quais, direto, imediato e indireto, e a Memória Principal (MP) é orientada a bytes, possuindo endereços com 16 bits cada. Em um dado momento, algumas posições da MP possuem os seguintes valores, expressos em hexadecimal:

End	Conteúdo
15B9	7C
15BA	82
15BB	15BC
15BC	BE
15BD	9A
15BE	75

a) Se uma instrução usa o modo direto e o campo operando possui o valor 15BA, qual será o valor do dado?

$Dado = 82$

b) Se uma instrução usa o modo imediato e o campo operando possui o valor 15B9, qual será o valor do dado?

$Dado = 15B9$

c) Se uma instrução usa o modo indireto e o campo operando possui o valor 15BB, qual será o valor do dado?

$Dado = BE$

3) (2,0) Responda:

a) Por que um programa em linguagem Assembly não é diretamente executável pelo processador?

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 mneumônicos 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 (mneumônicos) para seu respectivo em linguagem binária (máquina)

b) Por que um compilador deve ser específico para uma determinada linguagem de programação e para uma determinada UCP?

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 que distingue compiladores, mesmo que sejam específicos a uma mesma linguagem de programação e processador, é o sistema operacional. Sistemas operacionais diferentes possuem chamadas ao sistema também diferentes, estas chamadas permitem ao programa ter acesso aos recursos de hardware e de gerenciamento de processos como: criação, encerramento, comunicação entre processos, gerenciamento de memória, entre outros.

4. Considere um computador que utiliza 20 bits para representar números inteiros e números em ponto flutuante. Na representação para ponto flutuante, como na representação IEEE 754, o bit mais à esquerda representa o sinal e deve ser 0 para números positivos e 1 para números negativos, os próximos 6 bits representam o expoente e os próximos 13 bits representam a parte fracionária da mantissa. Quando todos os bits do expoente são iguais a 0 ou iguais a 1, representam-se os casos especiais referentes a números denormalizados, infinito e Not a Number. Caso contrário, o número a ser representado deve ser expresso na notação científica normalizada $(+/- (1, b_1 b_2 b_3 \dots b_{10})_2 \times 2^{\text{expoente}})$. O expoente deve ser representado em excesso de 31. A representação é mostrada na figura abaixo.

S	Expoente	Mantissa
1 bit	6 bits	13 bits

- a) (1,0) Considere o seguinte conjunto de 20 bits:

11000010100000000000

Indique o valor em decimal (pode deixar as contas indicadas) para este conjunto de bits quando considerarmos que ele está representando:

- um inteiro sem sinal
- um inteiro representado em complemento a 2
- um número em ponto flutuante utilizando a representação do enunciado

- um inteiro sem sinal

$$2^{19} + 2^{18} + 2^{13} + 2^{11} = 796.672$$

- um inteiro representado em sinal e magnitude

$$-2^{19} + (2^{18} + 2^{13} + 2^{11}) = -251.904$$

- um número em ponto flutuante utilizando a representação do enunciado.

1 100001 01000000000000

Sinal: 1 (negativo)

Expoente: 100001 : +33 - 31 = +2

Mantissa: , 01000000000000

$$-1,01 \times 2^{+2} = (-101)_2 = (-5)_{10}$$

b) (0,6) Indique o maior e o menor valor em decimal (pode deixar as contas indicadas) que pode ser representado pelos 20 bits para cada uma das representações abaixo:

I _inteiro sem sinal

Maior valor: 11111111111111111111 =

$$2^{19} + 2^{18} + 2^{17} + 2^{16} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = \mathbf{1.048.575}$$

Menor valor: 00000000000000000000 = 0

II _inteiro em complemento a 2

Maior valor: 01111111111111111111 =

$$2^{18} + 2^{17} + 2^{16} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = \mathbf{+524.287}$$

Menor valor: 10000000000000000000 = 0 = -2^{19} = \mathbf{-524.288}

III_ número expresso na notação científica normalizada na representação ponto flutuante definida no enunciado

Positivo

Maior valor: 0 111110 11111111111111 = +1,11111111111111 $\times 2^{+31}$ = +4.294.967.296

Menor valor: 0 000001 00000000000000 = +1,00000000000000 $\times 2^{-30}$ = +9,3132257 $\times 10^{-10}$

Negativo:

Maior valor: 1 000001 00000000000000 = -1,00000000000000 $\times 2^{-30}$ = -9,3132257 $\times 10^{-10}$

Menor valor: 1 111110 11111111111111 = -1,11111111111111 $\times 2^{+31}$ = -4.294.967.296

c) (0,4) Indique a representação em ponto flutuante utilizando a representação do enunciado para os seguintes valores decimais:

i. +3,5

$$(+3,5)_{10} = (+11,1)_2 = +1,11 \times 2^{+1}$$

Sinal: 0 (positivo)

Expoente: +1 + 31 = +32 = (100000)

Mantissa: , 11000000000000

Resposta: 0 100000 11000000000000

ii. -0,6

$$(-0,6)_{10} = (-0,100110011001100110011...)_{2} = -1,0011001100110011... \times 2^{-1}$$

Sinal: 1 (negativo)

Expoente: $-1 + 31 = +30 = (011110)$

Mantissa: ,0011001100110

Resposta: 1 011110 0011001100110

5. (2,0) Considere uma máquina cujo controlador da placa de vídeo possui três registradores para se comunicar com o resto do sistema: um para receber comandos do sistema (pedido para enviar um byte para a placa de vídeo, por exemplo), outro para indicar se a placa de vídeo está pronta para receber bytes do sistema, e outro para armazenar o byte enviado pelo sistema. O primeiro registrador possui o endereço 60, o segundo 61 e o terceiro 62. Para se enviar um byte para a placa de vídeo, deve-se enviar o comando 30 para o controlador da placa, utilizando-se o endereço 60. O controlador da placa armazena o valor 01 no registrador de estado (registrador 61) quando o byte puder ser enviado para a placa e o byte deve ser disponibilizado no registrador 62. Descreva detalhadamente como será realizado o envio de 3000 bytes do sistema para o controlador de placa de vídeo para cada um dos três seguintes métodos de comunicação: (1) por E/S programada, (2) por interrupção e (3) 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 30 para o registrador do controlador da placa de vídeo com endereço 60. Depois, fica lendo o conteúdo do registrador com endereço 61 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 o registrador da controladora com endereço 62. Este procedimento é repetido para cada um dos 2999 bytes seguintes a serem enviados para a placa de vídeo.

E/S por interrupção:

A UCP envia o comando 30 para o registrador com endereço 60. Depois, vai executar outras instruções. Quando o controlador da placa de vídeo está pronto para receber o 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, encaminhando o byte para o registrador de endereço 62 da controladora de vídeo. Este procedimento é repetido para cada um dos 2999 bytes seguintes a serem enviados para a placa de vídeo.

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 da placa de vídeo, o endereço inicial da memória onde devem ser lidos os bytes

a serem enviados, o número de bytes a serem enviados (3000 bytes) e a indicação de que o controlador de DMA deve transferir os bytes da memória para o controlador da placa de vídeo. Após receber estas informações, o controlador de DMA realiza a transferência dos bytes entre a memória e o controlador da placa de vídeo 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.