



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 2007**

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) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 16 bits. Na representação para ponto flutuante, 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 bit mais à esquerda representa o sinal e deve ser 0 para números positivos e 1 para números negativos. O expoente está representado em sinal e magnitude. A representação é mostrada na figura abaixo.

S	Expoente em sinal e magnitude	$b_{-1} b_{-2} b_{-3} \dots b_{-10}$
1	5 bits	10 bits

- a) (1,0) Suponha que o conteúdo dos 16 bits seja  $BA01_{16}$ . Indique o valor **em decimal (pode deixar as contas indicadas)** para este conjunto de bits quando considerarmos que ele está representando:

**Resposta:**

O conjunto de bits é 1011101000000001

a.1) (0,2) um inteiro sem sinal

$$2^{15} + 2^{13} + 2^{12} + 2^{11} + 2^9 + 2^0 = 47617.$$

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

$$-2^{15} + 2^{13} + 2^{12} + 2^{11} + 2^9 + 2^0 = -17919 \text{ ou}$$

$$- \text{inv}(1011101000000001) + 1 = -(2^{14} + 2^{10} + 2^8 + 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0) = -17919$$

a.3) (0,5) um número em ponto flutuante utilizando a representação do enunciado  
 Bit de sinal igual a 1, número negativo  
 Bits de expoente iguais a 01110. Como está representado em sinal e magnitude, o expoente é igual a +14.  
 Bits depois da vírgula: 1000000001  
 Logo o número na base 2 será igual a  $-(1,1000000001) \times 2^{14}$   
 Na base 10,  $-(2^{14} + 2^{13} + 2^4) = -24592$

- b) (0,5) Qual o menor e o maior valor **positivo** normalizado na representação em ponto flutuante, descrita no enunciado, para este computador? Os valores devem ser representados **em decimal**.

**Resposta:**

O menor número positivo, deve possuir sinal igual a 0, o menor expoente possível que representado em sinal magnitude é -15 e a menor mantissa fracionária, ou seja todos os bits depois da vírgula devem ser 0. Logo o menor número positivo na base 2 é  $+(1,0) \times 2^{-15}$ , que na base 10 é  $+2^{-15} = +0,000030517578125$

O maior número positivo, deve possuir sinal igual a 0, o maior expoente possível que representado em sinal magnitude é +15 e a maior mantissa fracionária, ou seja todos os bits depois da vírgula devem ser 1. Logo o maior número positivo na base 2 é  $+(1,111111111) \times 2^{+15}$ , que na base 10 é  $+2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 = +65504$

- c) (0,5) Mostre a representação em ponto flutuante do valor em decimal -16,2, na representação do enunciado para ponto flutuante.

**Resposta:**

-16,2 representado na base 2 é igual a  $-(10000,0011001100110011001100\dots)$ .  
 Na representação descrita no enunciado o número tem que estar na forma  $(+/- (1, b_1 b_2 b_3 \dots b_{10})_2 \times 2^{\text{expoente}})$ . Logo teremos o número representado nesta forma igual a  $-(1,00000011001100110011001100\dots) \times 2^{+4}$ .  
 Logo bit de sinal é igual a 1  
 Representação do expoente é igual a 00100  
 10 bits depois da vírgula iguais a 0000001100  
 Logo a representação será 1001000000001100

2. (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 (pedido de envio de um byte associado a uma tecla, por exemplo), o segundo para indicar se o controlador de teclado possui um byte associado à entrada de uma tecla para enviar para o sistema, e o terceiro onde é armazenado o byte associado a uma tecla pressionada pelo usuário do sistema. O primeiro registrador possui o endereço 30, o segundo 31 e o terceiro 32. O processo de recebimento de um byte associado a uma tecla pressionada pelo usuário consiste no envio do comando 01 para o registrador 30 do controlador de teclado, indicando que o sistema deseja receber um byte associado a uma tecla. O controlador de teclado armazena o valor 01 no registrador de estado 31 quando o controlador de teclado possuir um byte armazenado referente a uma tecla pressionada pelo usuário. Este byte fica armazenado no registrador 32. Descreva **detalhadamente** os três possíveis métodos de comunicação entre o controlador de impressora

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 pede para receber, pelo barramento de dados, um byte armazenado no registrador com endereço 31. Este procedimento é repetido para receber cada byte referente a uma tecla pressionada pelo usuário.

E/S por interrupção:

A UCP envia o comando 01 para o registrador com endereço 30. Depois, vai executar outras instruções. Quando o controlador de teclado detecta que existe um byte referente a uma tecla pressionada por um usuário, ela 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, recebendo o byte armazenado no registrador de endereço 31. . Este procedimento é repetido para receber cada byte referente a uma tecla pressionada pelo usuário.

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 referentes às teclas pressionadas, 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.

3. (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:** Uma solução seria usar endereçamento por registrador base mais deslocamento. Por exemplo, teríamos 2 bits para especificar um registrador e 4 bits para especificar um deslocamento. Poderíamos, assim, usar 4 registradores, cada um com até 16 deslocamentos possíveis, fornecendo 64 endereços diferentes.

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

Resposta:

- **Direto:** Seu campo operando contém o endereço do dado, requer apenas um acesso à memória principal, sendo mais rápido que o modo indireto. Possui como vantagem a flexibilidade no acesso a variáveis de valor diferente em cada execução do programa e como desvantagem a perda de tempo, se o dado for uma constante.
- **Indireto:** O campo operando corresponde ao endereço que contém a posição onde está o conteúdo desejado, necessita de 2 acessos a memória principal, portanto mais lento que o endereçamento anterior. Tem como vantagem o manuseio de vetores e utilização como ponteiro, e como desvantagem muitos acessos à memória principal.

4. (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.

5. (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:

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

Resposta:

LOAD M	=>	ACC	<-	M	(o Acumulador recebe valor contido no Endereço M)
STORE M	=>	M	<-	ACC	(o endereço de memória M recebe o conteúdo do acumulador)

ADD M	=>	ACC	<-	ACC + M	(O conteúdo do acumulador será adicionado pelo conteúdo do endereço M)
SUB M	=>	ACC	<-	ACC - M	(O conteúdo do acumulador será subtraído pelo conteúdo do endereço M)
MUL M	=>	ACC	<-	ACC * M	(O conteúdo do acumulador será multiplicado pelo conteúdo do endereço M)
DIV M	=>	ACC	<-	ACC / M	(O conteúdo do acumulador será dividido pelo conteúdo do endereço M)

#### PROGRAMA:

LOAD C	=>	ACC	<-	C
SUB A	=>	ACC	<-	ACC - A
STORE T1	=>	T1	<-	ACC
LOAD E	=>	ACC	<-	E
DIV B	=>	ACC	<-	ACC / B
STORE T2	=>	T2	<-	ACC
LOAD D	=>	ACC	<-	D
SUB T2	=>	ACC	<-	ACC - T2
STORE T2	=>	T2	<-	ACC
LOAD B	=>	ACC	<-	B
MUL T1	=>	ACC	<-	ACC * T1
STORE T1	=>	T1	<-	ACC
LOAD T2	=>	ACC	<-	T2
MUL D	=>	ACC	<-	ACC * D
STORE T2	=>	T2	<-	ACC
LOAD T1	=>	ACC	<-	T1
ADD T2	=>	ACC	<-	ACC + T2
STORE T1	=>	T1	<-	ACC
LOAD A	=>	ACC	<-	A
ADD T1	=>	ACC	<-	ACC + T1
STORE X	=>	X	<-	ACC

OBS: Sejam A,B,C,D,E,X, T1 e T2 endereços.