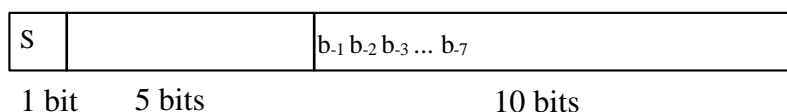


- 1) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 16 bits. Na representação para ponto flutuante, o número a ser representado deve ser expresso na notação científica normalizada  $(+/- (1, b_{-1} b_{-2} b_{-3} \dots b_{-10})_2 \times 2^{\text{expoente}})$ . 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 excesso de 16. A representação é mostrada na figura abaixo.



- a) (1,0) Suponha que o conteúdo dos 16 bits seja  $AC84_{16}$ . Indique o valor **em decimal (pode deixar as contas indicadas)** para este conjunto de bits quando considerarmos que ele está representando:

$AC84_{16} \ (1010\ 1100\ 1000\ 0100)_2$

- i) (0,2) um inteiro sem sinal

$$2^{15} + 2^{13} + 2^{11} + 2^{10} + 2^7 + 2^2 = 44.164$$

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

$$-2^{15} + (2^{13} + 2^{11} + 2^{10} + 2^7 + 2^2) = -21.372$$

- iii) (0,5) um número em ponto flutuante utilizando a representação do enunciado

*Sendo:*

$Sinal = 1 \Rightarrow \text{negativo}$

$Expoente = 01011_2 = +2^3 + 2^1 + 2^0 = +11 - 16 = -5$

$Mantissa = 0010000100$

$$Temos\ então \Rightarrow -1,0010000100 \times 2^{-5} = 0,000010010000100_2 = +2^{-5} + 2^{-8} + 2^{-13} = -0,0353$$

- 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**.

*Maior valor positivo:*

$$0\ 11110\ 1111111111 = 1,1111111111 \times 2^{+14} = +32.752$$

*Menor valor positivo:*

$$0\ 00001\ 0000000000 = 1,0000000000 \times 2^{-15} = +0,00003052$$

- c) (0,5) Mostre a representação em ponto flutuante do valor em decimal -17,125, na representação do enunciado para ponto flutuante.

$$-17,125_{10} = 10001,001_2$$

$$10001,001_2 = 1,0001001_2 \times 2^{+4}$$

Sinal: 1 (negativo)

Expoente  $+4 + 16 = +20_{10} = 10100_2$  (excesso de 16)

Mantissa: 0001001

Resposta: 1 10100 0001001000

- 2) (2,0) Considere uma máquina cujo controlador de scanner 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 scanner, por exemplo), o segundo para indicar o estado do controlador do scanner (indicação de que o controlador de scanner está pronto para enviar um byte ao sistema, por exemplo), e o terceiro onde o byte a ser enviado ao sistema é armazenado pelo scanner. O primeiro registrador possui o endereço 55, o segundo 57 e o terceiro 59. O processo de recebimento de um byte do scanner consiste no envio do comando 01 para o registrador 55 do controlador de scanner, indicando que o sistema deseja receber um byte do scanner. O controlador de scanner recebe este comando e armazena o valor 02 no registrador de estado 57 quando o controlador de scanner tiver um byte pronto para ser enviado pronto. Este byte deverá ser armazenado no registrador 59 do controlador de scanner. Descreva **detalhadamente** os três possíveis métodos de comunicação entre o controlador de scanner 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 de endereço 55 do controlador do scanner. Depois, fica lendo o conteúdo do registrador de endereço 57 e verificando se o seu conteúdo é igual a 02. Quando o conteúdo for igual a 02, a UCP envia, pelo barramento de dados, o byte para o registrador 59 do controlador de scanner.

***E/S por interrupção:***

A UCP envia o comando 01 para o registrador de endereço 55. Depois, vai executar outras instruções. Quando o controlador do scanner está pronto para receber o byte, ele envia um sinal de interrupção para a UCP. Este, 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 59 do controlador do scanner.

***E/S por acesso direto à memória:***

Neste caso, existe um controlador de DMA (Direct Access Memory) que é responsável pela transferência do(s) byte(s). A UCP informa a este controlador que a operação de transferência deverá ser realizada com o controlador do scanner. Informa também o endereço inicial da memória onde está(ão) os byte(s) a ser(em) enviado(s), a quantidade de bytes e a indicação de que o controlador de DMA deve transferir o(s) byte(s) da memória para o controlador do scanner. Após receber estas informações, o controlador de DMA realiza a transferência do(s) byte(s) entre a memória e o controlador do scanner sem a intervenção da UCP. Após a transferência, o controlador de DMA avisa o fim da operação para a UCP através de um sinal de interrupção.

- 3) (3,0) Crie um conjunto de instruções de dois operandos definidas em Linguagem Assembly, utilizando endereçamento direto, imediato e por registrador, necessárias para a realização de operações aritméticas e elabore um programa para o cálculo da equação abaixo.

$$X = B * (D - 2) + 1$$

**Sejam:**        **z** o valor imediato;  
              **R1** um endereço de registrador;  
              **Y** um endereço de memória

*Operações entre registrador (modo registrador) e endereço de memória (modo direto)*

```
ADD R1,Y    => (R1) <- (R1) + (Y)
SUB R1,Y    => (R1) <- (R1) - (Y)
MUL R1,Y    => (R1) <- (R1) * (Y)
DIV R1,Y    => (R1) <- (R1) / (Y)
LD  R1,Y    => (R1) <- (Y)
ST  R1,Y    => (Y) <- (R1)
```

*Operações entre registrador (modo registrador) com valor imediato (modo imediato)*

```
ADDi R1,z   => (R1) <- (R1) + z
SUBi R1,z   => (R1) <- (R1) - z
MULi R1,z   => (R1) <- (R1) * z
DIVi R1,z   => (R1) <- (R1) / z
```

*OBS: Como cada instrução tem um código de operação diferente, seus mnemônicos (nomes) em assembly também deverão ser diferentes. Veja que para diferenciar a instrução de soma entre registrador e endereço de memória (ADD R1,X) da instrução de soma entre registrador e valor imediato (ADDi R1,z) foi adotado o acréscimo do sufixo i. A escolha deste sufixo não é regra, cada projetista define sua forma de nomear e diferenciar as instruções de sua arquitetura.*

*Solução para a expressão:*

$$X = B * (D - 2) + 1$$

```
LD    R1,D  => (R1) <- (D)
SUBi  R1,2  => (R1) <- (R1) - 2
MUL   R1,B  => (R1) <- (R1) * (B)
ADDi  R1,1  => (R1) <- (R1) + 1
ST    R1,X  => (X) <- (R1)
```

- 4) (3,0) Responda as questões abaixo:

**a) Explique Compilação e Interpretação.**

*A compilação consiste na análise de um programa escrito em linguagem de alto nível (programa fonte) e sua tradução para 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*

**b) Explique 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).*