

**Nome –**

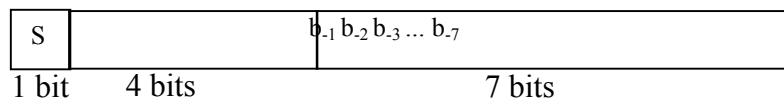
**Assinatura –**

---

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.
  6. Repostas sem justificativa não serão consideradas.
- 

- 1) (2,0) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 12 bits. Na representação para ponto flutuante, o número a ser representado deve ser expresso na notação científica normalizada  $(+/- (1, b_1 b_2 b_3 \dots b_7)_2 \times 2^{\text{expoente}})$ . 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 8. A representação é mostrada na figura abaixo.



- a) (1,0) Suponha que o conteúdo dos 12 bits seja  $A9D_{16}$ . Indique o valor **em decimal (pode deixar as contas indicadas)** para este conjunto de bits quando considerarmos que ele está representando:

$$(A9D)_{16} = (101010011101)_2$$

i) (0,2) um inteiro sem sinal

$$2^{11} + 2^9 + 2^7 + 2^4 + 2^3 + 2^2 + 2^0 = 2.717$$

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

$$-2^{11} + (2^9 + 2^7 + 2^4 + 2^3 + 2^2 + 2^0) = -1.379$$

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

*sinal = 1 = negativo*

*expoente = 0101 = 5 - 8 = -3*

*mantissa = 0011101*

*representando a normalização = -1, 0011101 x 2<sup>-3</sup>*

*-0,0010011101 = -0,153320313*

- 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 1110 1111111 = +1,1111111 x 2<sup>+6</sup> = +127,5*

*menor valor positivo*

*= 0 0001 0000000 = +1,0000000 x 2<sup>-7</sup> = +0,0078125*

- c) (0,5) Mostre a representação em ponto flutuante do valor em decimal -27,175, na representação do enunciado para ponto flutuante.

*-27,175 = -11011,00010110011001100...*  
*1,101100010110011001100... x 2<sup>4</sup>*

*Sinal: 1 (negativo)*

*Expoente: +4 + 8 = 12<sub>10</sub> = 1100<sub>2</sub>*

*Mantissa: 1011000*

*Resposta: 1 1100 1011000*

- 2) (2,0) Considere um sistema onde o número de ciclos de relógio para realizar uma operação de Entrada/Saída (E/S) por interrupção seja igual a 1200 e o processador utiliza um relógio de 800 MHz para executar as instruções. Determine o overhead (relação entre ciclos de CPU consumidos por segundo pelo processo de E/S e número de ciclos disponíveis) que ocorre quando se realiza uma operação de E/S com um disco rígido que transfere dados para o processador em blocos de 32 bytes e possui uma taxa de transferência de 64MB/segundo e que está ativo 8% do tempo total em que a CPU está sendo utilizada.

*Taxa de transferência = 64MB/s = 64.000.000 B/seg.*

*Em cada operação é transferido um bloco de 32 bytes.*

*A cada segundo ocorrem: 64.000.000B / 32B = 2.000.000 operações.*

*Como cada operação leva 1200 ciclos. O total de ciclos será 2.000.000 x 1.200 = 2400 x 10<sup>6</sup> ciclos/seg.*

*O overhead será obtido pela seguinte razão:*

*= 2400 x 10<sup>6</sup> ciclos/seg x 0,08 / 800 x 10<sup>6</sup> ciclos/seg = 0,24 ou 24%*

- 3) (3,0) 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 equação abaixo.

## **I\_ CONJUNTO DE INSTRUÇÕES PARA 1 OPERANDO:**

ADD X           => (ACC) <- (ACC) + (X)  
SUB X           => (ACC) <- (ACC) - (X)  
MUL X           => (ACC) <- (ACC) \* (X)  
DIV X           => (ACC) <- (ACC) / (X)  
LOAD X          => (ACC) <- (X)  
STORE X         => (X) <- (ACC)

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

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

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

- a) Explique Compilação e Interpretação (Dê exemplos de linguagens que se utilizem de compiladores e de linguagens que se utilizem de interpretadores).

*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.*

*Exemplos de linguagem interpretadas: ASP, BASIC, Java, PHP, Python, Lisp entre outras.*

*Exemplos de linguagem compilada: C, Pascal, Delphi, Visual Basic, entre outras.*

- b) Explique Sistemas SMP e Sistemas NUMA (Forneça exemplos atuais de 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).*

*Exemplo de NUMA: Cray T3D*

*Exemplo de SMP: Maioria dos servidores da HP, Compac, IBM.*