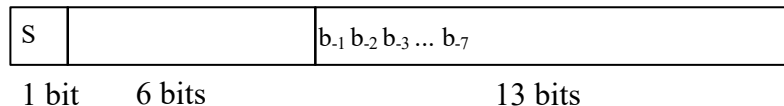


Disciplina: Organização de Computadores
GABARITO DA AP2 2019-01

- 1) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 20 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 8F668₁₆. Indique o valor **em decimal (pode deixar as contas indicadas)** para este conjunto de bits quando considerarmos que ele está representando:

$$8F668_{16} = 1000\ 1111\ 0110\ 0110\ 1000_2$$

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

$$2^{19} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{10} + 2^9 + 2^6 + 2^5 + 2^3 = 587.368$$

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

$$-2^{19} + (2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{10} + 2^9 + 2^6 + 2^5 + 2^3) = -461.208$$

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

$$1\ 000111\ 1011001101000_2$$

Temos então:

$$\text{Sinal} = 1 \text{ (negativo)}$$

$$\text{Expoente} = 000111 \text{ (por excesso)} = +7 - 16 = -9$$

$$\text{Mantissa} = ,1011001101000$$

$$\text{Normalizado: } -1,1011001101000 \times 2^{-9}$$

$$= - (2^{-9} + 2^{-10} + 2^{-12} + 2^{-13} + 2^{-16} + 2^{-17} + 2^{-19}) = -0,00332$$

- 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**.

$$\text{Maior positivo: } 0\ 111110\ 111111111111 = (\text{expoente} = 62 - 16 = 46)$$

$$1,111111111111 \times 2^{+46} = +1,407 \times 10^{+14}$$

$$\text{Menor positivo: } 0\ 000001\ 00000000000000 = (\text{expoente} = 1 - 16 = -15)$$

$$1,0000000000000 \times 2^{-15} = +3,052 \times 10^{-5}$$

- c) (0,5) Mostre a representação em ponto flutuante do valor em decimal -49,875, na representação do enunciado para ponto flutuante.

Convertendo para binário = -110001,111

$$-110001,111 = -1,10001111 \times 2^{+5}$$

Temos então:

Sinal = 1 (negativo)

Expoente = +5 = (por excesso de 16) = $5 + 16 = 21_{10} = 010101_2$

Mantissa = , 10001111

Resultado: 1 010101 10001111000000

- 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 2,4 GHz 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 64 bytes e possui uma taxa de transferência de 48 MB/segundo e que está ativo 4% do tempo total em que a CPU está sendo utilizada.

Taxa de transferência = 48MB/s = 48.000.000 B/seg.

Em cada operação é transferido um bloco de 64 bytes.

A cada segundo ocorrem: $48.000.000B / 64B = 750.000$ operações/seg.

Como cada operação leva 1200 ciclos.

O total de ciclos consumidos é igual a $750.000 \text{ operações/seg} \times 1.200 \text{ ciclos/operação}$
 $= 900 \times 10^6 \text{ ciclos/seg.}$

Clock = 2,4GHz = $2400 \times 10^6 \text{ ciclos/seg}$

O overhead será obtido pela seguinte razão:

$$= (900 \times 10^6 \text{ ciclos/seg} \times 4\% \text{ de atividade}) / (2400 \times 10^6 \text{ ciclos/seg})$$
$$= 0,015 \text{ ou } 1,5\%$$

- 3) Responda as questões abaixo:

- a) (2,0) Analise os modos de endereçamento direto, indireto e imediato, estabelecendo diferenças de desempenho, vantagens e desvantagens de cada um.

Imediato: O campo operando contém o dado, desta forma o dado é transferido da memória juntamente com a instrução. Vantagem: Rapidez na execução da instrução, pois não requer acesso à memória principal, apenas na busca da própria instrução. Desvantagem: Limitação do tamanho do campo operando das instruções reduzindo o valor máximo do dado a ser manipulado. Trabalho excessivo para alteração de valores quando o programa é executado repetidamente e o conteúdo das variáveis serem diferentes em cada execução. Desempenho: Não requer acesso a memória principal. Mais rápido que o modo direto

Direto: O campo operando da instrução contém o endereço onde se localiza o dado. Vantagem: Flexibilidade no acesso a variáveis de valor diferente em cada execução do programa. Desvantagem: Limitação de memória a ser usada conforme o tamanho do operando. Desempenho: Requer apenas um acesso a memória principal. Mais rápido que o modo indireto

Indireto: O campo de operando contém o endereço de uma célula, sendo o valor contido nesta célula o endereço do dado desejado. Vantagem: Usado como "ponteiro". Elimina o problema do modo direto de limitação do valor do endereço do dado. Manuseio de vetores (quando o modo indexado não está disponível). Desvantagem: Muitos acesso à MP para execução. Desempenho: inferior ao modo direto por exigir pelo menos 2 acessos à memória principal.

- b) (1,5) Qual é o objetivo do emprego do modo de endereçamento base mais deslocamento? Qual é a diferença de implementação e utilização entre esse modo e o modo indexado?

O base mais deslocamento tem como seu principal objetivo permitir a modificação de endereço de programas ou módulos destes (que é a relocação de programa), bastando para isso uma única alteração no registrador base.

O base mais deslocamento tem como característica o endereço ser obtido da soma do deslocamento com o registrador base, diferindo do modo indexado onde o do registrador base é fixo e variar no deslocamento, ao contrário deste onde o deslocamento é fixo e com a alteração do registrador base permite-se a mudança do endereço.

Exemplos de instruções modo indexado:

LDX Ri, Op ==> ACC <--- ((Op) + (Ri))

ADX Ri, Op ==> ACC <--- ACC + ((Op) + (Ri))

Exemplo: instrução base mais deslocamento:

LDB Rb, Op ==> (ACC) <--- ((Op) + (Rb))

ADB Rb, Op ==> ACC <--- ACC + ((Op) + (Rb))

Sendo,

Op = Operando

Ri = Registrador de índice

Rb = Registrador base

- c) (1,5) 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.

- d) (1,0) Sistemas MIMD e Sistemas SIMD(Forneça exemplos atuais desses sistemas).

MIMD – Multiple instruction stream, multiple data stream, classificação esta onde um conjunto de processadores executam simultaneamente sequências diferentes de instruções sobre conjunto de dados diferentes. Exemplo: SMPs, clusters, sistemas NUMA. Os multicores, processadores com 2 ou mais núcleos que atuam como processadores distintos e executam instruções independentes, também são classificados como MIMD.

SIMD – Single instruction stream, multiple data stream, onde vários elementos de processamento executam a mesma sequência de instruções ao mesmo tempo sobre um conjunto de dados diferente. Exemplo: Processadores matriciais. Os aceleradores de hardware baseados em

placas gráficas GPU também pertencem a essa classificação. Uma GPU é composta de vários multiprocessadores e cada multiprocessador consiste de um grupo de processadores escalares.