

Disciplina: Organização de Computadores

GABARITO DA AP2 2018-02

- 1) (3,0) Crie um conjunto de instruções de um operando definidas em Linguagem Assembly, utilizando endereçamento direto e imediato, 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;
 Y um endereço de memória

Operações com endereço de memória (modo direto)

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

Operações com valor imediato (modo imediato)

ADDi Z	=> ACC <- ACC + Z
SUBi Z	=> ACC <- ACC - Z
MULi Z	=> ACC <- ACC * Z
DIVi Z	=> ACC <- ACC / 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 modo direto e modo indireto foi adotado, nesta solução, o acréscimo do sufixo "i" para as instruções do modo imediato. 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	D	=> ACC <- (D)
SUBi	2	=> ACC <- ACC - 2
MUL	B	=> ACC <- ACC * (B)
ADDi	1	=> ACC <- ACC + 1
ST	X	=> (X) <- ACC

- 2) (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 arquiteturas RISC e CISC.

RISC: Reduced Instruction Set Computer – Computador com um conjunto reduzido de instruções

CISC - Complex Instruction Set Computer: Computador com um conjunto complexo de instruções

CISC: Principais características:

Possui microprogramação para aumento da quantidade de instruções incluindo novos modos de endereçamento, de forma a diminuir a complexidade dos compiladores e em consequência permitir linguagens de alto nível com comandos poderosos para facilitar a vida dos programadores. Em contrapartida, muitas instruções significam muitos bits em cada código de operação, instrução com maior comprimento e maior tempo de interpretação.

RISC: Principais características:

Menor quantidade de instruções e tamanho fixo. Não há microprogramação. Permite uma execução otimizada, mesmo considerando que uma menor quantidade de instruções vá conduzir a programas mais longos. Uma maior quantidade de registradores e suas utilizações para passagem de parâmetros e recuperação dos dados, permitindo uma execução mais otimizada de chamada de funções. Menor quantidade de modos de endereçamento com o objetivo de reduzir de ciclos de relógio para execução das instruções. Instruções de formatos simples e únicos tiram maior proveito de execução com pipeline cujos estágios consomem o mesmo tempo.

3) (2,5) Considere o seguinte conjunto de 32 bits representado em hexadecimal $(867AE000)_{16}$. Indique o valor **em decimal (deixe as contas indicadas) para este conjunto de bits, quando considerarmos que ele está representando:**

$$(867AE000)_{16} = (1000\ 0110\ 0111\ 1010\ 1110\ 0000\ 0000\ 0000)_2$$

a) (0,3) um inteiro sem sinal.

$$2^{31} + 2^{26} + 2^{25} + 2^{22} + 2^{21} + 2^{20} + 2^{19} + 2^{17} + 2^{15} + 2^{14} + 2^{13} = 2.256.199.680$$

b) (0,3) um inteiro representado em sinal e magnitude.

$$-(2^{26} + 2^{25} + 2^{22} + 2^{21} + 2^{20} + 2^{19} + 2^{17} + 2^{15} + 2^{14} + 2^{13}) = -108.716.032$$

c) (0,5) um inteiro em complemento a 2.

$$-2^{31} + (2^{26} + 2^{25} + 2^{22} + 2^{21} + 2^{20} + 2^{19} + 2^{17} + 2^{15} + 2^{14} + 2^{13}) = -2.038.767.616$$

d) (0,8) um número em ponto flutuante que utiliza a representação padrão IEEE 754 para precisão simples, ou seja, um bit para o sinal (0 para positivos e 1 para negativos), 8 bits para o expoente representado em excesso de 127 e 23 bits para a parte fracionária. Os números a serem representados devem estar no seguinte formato: $N = +/- (1, b-1b-2b-3 \dots b-m)2^{\times 2\text{Expoente}}$.

$$(1000\ 0110\ 0111\ 1010\ 1110\ 0000\ 0000\ 0000)_2$$

Sendo:

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

$$\text{Expoente} = 00001100_2 = +2^3 + 2^2 = +12 -127 = -115$$

$$\text{Mantissa} = 11110101110000000000000$$

$$\text{Temos então} \Rightarrow -1, 11110101110000000000000 \times 2^{-115}$$

$$= +2^{-115} + 2^{-116} + 2^{-117} + 2^{-118} + 2^{-119} + 2^{-121} + 2^{-123} + 2^{-124} + 2^{-125} = -4,72 \times 10^{-35}$$

O máximo de transações por segundo = $1500 / 12 = 125$ transações

