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.

```
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)

```
\begin{array}{lll} \textit{ADDi z} & => \textit{ACC} <- \textit{ACC} + \textit{z} \\ \textit{SUBi z} & => \textit{ACC} <- \textit{ACC} - \textit{z} \\ \textit{MULi z} & => \textit{ACC} <- \textit{ACC} * \textit{z} \\ \textit{DIVi z} & => \textit{ACC} <- \textit{ACC} / \textit{z} \end{array}
```

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

```
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

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 enderecamento, 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 instrucões 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 da 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
```

(0,3) um inteiro sem sinal.

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

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

(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: $N = +/-(1, b-1b-2b-3...b-m)2 \times 2Expoente$.

Sendo:

Sinal = 1 => negativo
Expoente =
$$00001100_2 = +2^3 + 2^2 = +12$$
 -127 = -115
Mantissa = 11110101110000000000000

e) (0,6) Qual o menor e o maior valor positivo de números expressos na notação científica normalizada que podem ser representados utilizando-se a representação em ponto flutuante, descrita no item "d", para este computador de 32 bits? Os valores devem ser representados em decimal.

4) (1,5) Considere dois sistemas de computação, S1 e S2, que possuem a mesma UCP que pode realizar 120.000.000 instruções por segundo. O sistema S1 possui um barramento de sistema que suporta a realização de 2.400 operações de entrada/saída por segundo e o sistema S2 um barramento de sistema que suporta 1.500 operações de entrada/saída por segundo. Suponha que um gerenciador de banco de dados necessite realizar transações que requerem 12 operações de entrada/saída por segundo, sendo que cada operação requer 10.000 instruções da UCP.

Dados:

UCP realiza 120.000.000 instruções por segundo

S1 => barramento suporta 2400 operações por segundo

S2 => barramento suporta 1500 operações por segundo

Ambos sistemas devem realizar transações que requerem 12 operações por segundo e cada operação requer 10.000 instruções da UCP

a) (0,5) Calcule o número máximo de transações por segundo que a UCP poderá sustentar Cada transação requer 12 operações de E/S e cada operação 10.000 instruções da UCP, assim,

Máximo de operações = total instr. realizadas pela UCP por segundo / quant. instr. para cada operação Máximo de operações = 120.000.000 / 10.000 = 12.000 operações por segundo

Máximo de transações = máximo de operações / 12 operações de E/S = 12.000 / 12 = 1000 transações

- **b)** (0,6) Calcule o número máximo de transações por segundo que o barramento de sistema de cada um dos sistemas, S1 e S2, poderá sustentar
 - S1 => barramento suporta 2400 operações por segundo, como cada transação leva 12 operações O máximo de transações por segundo = 2400 / 12 = **200 transações**
 - S2 => barramento suporta 1500 operações por segundo, como cada transação leva 12 operações O máximo de transações por segundo = 1500 / 12 = **125 transações**
- c) (0,4) Indique o número máximo de transações por segundo que cada um dos sistemas, S1 e S2, poderá sustentar
 - S1 => barramento suporta 2400 operações por segundo, como cada transação leva 12 operações O máximo de transações por segundo = 2400 / 12 = **200 transações**
 - S2 => barramento suporta 1500 operações por segundo, como cada transação leva 12 operações O máximo de transações por segundo = 1500 / 12 = **125 transações**