

Fundação CECIERJ - Vice Presidência de Educação Superior a Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Organização de Computadores
GABARITO - AP3 1º semestre de 2018.

1. (3,5) Considere uma máquina com arquitetura semelhante à arquitetura vista no curso, que apresente as seguintes especificações:

- Capaz de endereçar 64 M células de memória principal.
- Deve possuir um registrador Acumulador, além do RDM (Registrador de Dados da Memória), REM (Registrador de Endereços da Memória), CI (Contador de Instrução) e RI (Registrador de Instrução).
- O conjunto de instruções de linguagem de máquina deve ter 39 instruções.
- Cada instrução deve conter um código de operação e um operando como mostrado abaixo, onde o operando indica um endereço de memória

Cód. Oper.	Operando
------------	----------

a) (0,3) Calcule o tamanho mínimo em bits do REM e do barramento de endereços.

*Memória com 16Mcélulas => $N = 64M$ células
tamanho mínimo do REM será o tamanho do barramento de endereços necessário para endereçar toda a memória.
Barramento de endereços (BE) = $\log_2 N = \log_2 64M = 26$ bits
REM = tamanho do BE = 26 bits*

b) (0,3) Calcule o tamanho mínimo em bits que a instrução deve ter.

*Cada instrução = código de operação + 1 operando
operando = endereço de uma célula = 26 bits
cod.operação = tamanho mínimo para 39 códigos diferentes
cod.operação = 6 bits (permite até 64 códigos diferentes)
tamanho da instrução = 6 + 26 = 32bits*

c) (0,8) Para o valor calculado no item b, indique o tamanho em bits de cada célula da memória principal, o tamanho do RDM e o barramento de dados de modo que a Unidade Central de Processamento obtenha uma instrução da memória principal realizando somente um acesso à memória principal.

*Para transferir uma instrução em apenas um acesso à memória principal, o tamanho do barramento de dados deverá ter o tamanho da instrução = 32 bits
RDM = barramento de dados = 32 bits.
Transferindo uma célula a cada acesso à MP, esta deverá ter o tamanho da instrução = 32bits*

d) (0,6) Calcule o tamanho de RI e CI utilizando-se os valores calculados nos itens anteriores.

*CI = tamanho necessário para endereçar toda a memória = 26 bits
RI = tamanho necessário para uma instrução = 32 bits*

- e) (0,3) Calcule a capacidade de armazenamento, em bits, da memória desta máquina.

$$\text{Total de bits} = T$$

$$T = N \times M \Rightarrow T = 64 \text{Mcélulas} \times 32 \text{ bits/célula} \Rightarrow T = 2048 \text{ Mbits ou } 128 \text{ Mbytes}$$

- f) (0,6) Descreva detalhadamente a execução da instrução **LDA Op.** nesta máquina. A instrução **LDA Op.** carrega o acumulador com o conteúdo da célula de memória cujo endereço é Op.

Passo 1: A CPU coloca no REM o valor do operando ($REM \leftarrow Op$), e é disponibilizado no barramento de endereço

Passo 2: A CPU aciona pelo barramento de controle o sinal de leitura de memória

Passo 3: A memória coloca o valor no barramento de dados, correspondente ao endereço contido no barramento de endereços, a seguir chega no RDM da CPU ($RDM \leftarrow MP(Op)$)

Passo 4: O valor armazenado no RDM é transferido para o Acumulador $ACC \leftarrow RDM$ (ou $ACC \leftarrow MP(Op)$)

Passo 5: CI é incrementado ($CI \leftarrow CI + 1$) para apontar para a próxima instrução a ser lida.

- g) (0,6) Descreva detalhadamente a execução da instrução **ADD Op.** nesta máquina. A instrução **ADD Op.** soma o conteúdo do acumulador com o conteúdo da célula de memória cujo endereço é Op. e armazena o resultado na memória no endereço Op.

Passo 1: A CPU coloca no REM o valor do operando ($REM \leftarrow Op$), e é disponibilizado no barramento de endereço

Passo 2: A CPU aciona pelo barramento de controle a leitura de memória

Passo 3: A memória coloca o valor no barramento de dados, e por consequência no RDM da CPU ($RDM \leftarrow MP(Op)$)

Passo 4: A CPU executa a soma do valor recebido com o contido no acumulador armazenando o resultado no acumulador; $ACC \leftarrow ACC + RDM$ (ou $ACC \leftarrow ACC + MP(Op)$)

Passo 5: CI é incrementado ($CI \leftarrow CI + 1$) para apontar para a próxima instrução a ser lida

2. (1,5) Considere uma máquina que utiliza 32 bits para representar números em ponto fixo e em ponto flutuante.

- a) (0,8) Mostre a representação de -31,625 utilizando-se a representação ponto flutuante precisão simples IEEE 754 (1 bit de sinal, 8 bits para expoente em excesso de 127, 23 bits para mantissa)

$$(-31,625)_{10} = (-11111,101)_2 = (-1,1111101 \times 2^{+4})_2$$

Sinal = 1 = negativo

$$\text{Expoente} = +4 \text{ (excesso de 127)} = +4 + 127 = 131 \text{ (10000011)}_2$$

Mantissa = , 1111101

Na representação: 1 10000011 111110100000000000000000

ou : 11000001 11111101 00000000 00000000

- b) (0,7) Para o conjunto de bits obtido no item anterior, indique o que ele representa na base 10, considerando-se as seguintes representações: **(Não precisa fazer as contas, deixe-as indicadas):**

- i. (0,3) um inteiro sem sinal

$$2^{31} + 2^{30} + 2^{24} + 2^{23} + 2^{22} + 2^{21} + 2^{20} + 2^{19} + 2^{18} + 2^{16} = 3.254.583.296$$

- ii. (0,4) um inteiro utilizando-se a representação em complemento a 2
 $-2^{31} + (2^{30} + 2^{24} + 2^{23} + 2^{22} + 2^{21} + 2^{20} + 2^{19} + 2^{18} + 2^{16}) = -1.040.384.000$

3. (2,5) 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.

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

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*

Solução para a expressão:

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

LD R1, D => *(R1) <- (D)*
SUB R1, A => *(R1) <- (R1) - (A)*
MUL R1, B => *(R1) <- (R1) * (B)*
ADDi R1, 1 => *(R1) <- (R1) + 1*
ST R1, X => *(X) <- (R1)*

4. (2,5) Explique detalhadamente a hierarquia de memória dos sistemas de computação atuais.

Podemos ilustrar essa hierarquia de memória na forma de uma pirâmide dividida em 4 níveis. No topo da pirâmide teríamos os registradores, que são pequenas unidades de memória que armazenam dados na UCP. São dispositivos de maior velocidade com tempo de acesso em torno de 1 ciclo de memória, menor capacidade de armazenamento além de armazenar as informações por muito pouco tempo.

Em um nível abaixo teríamos a memória cache, cuja função é acelerar a velocidade de transferência das informações entre UCP e MP e, com isso, aumentar o desempenho do sistema. A UCP procura informações primeiro na Cache. Caso não as encontre, as mesmas são transferidas da MP para a Cache. A cache possui tempo de acesso menor que a da Memória principal, porém com capacidade inferior a esta, mas superior ao dos registradores e o suficiente para armazenar uma apreciável quantidade de informações, sendo o tempo de permanência do dado menor do que o tempo de duração do programa a que pertence.

Abaixo da memória cache teríamos a memória básica de um sistema de computação, que é a memória principal. Dispositivo onde o programa (e seus dados) que vai ser executado é armazenado para que a UCP busque instrução por instrução para executá-las. A MP são mais lentas que a cache e mais rápidas que a memória secundária, possui capacidade bem superior ao da cache e os dados ou instruções permanecem na MP enquanto durar a execução do programa.

Finalmente, na base da pirâmide teríamos a memória secundária, memória auxiliar ou memória de massa, que fornece garantia de armazenamento mais permanente aos dados e programas do usuário. Alguns dispositivos são diretamente ligados: disco rígido, outros são conectados quando necessário: disquetes, fitas de armazenamento, CD-ROM. São os mais lentos em comparação com os outros níveis de memória, mas possuem a maior capacidade de armazenamento e armazenam os dados de forma permanente.