



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Organização de Computadores

AP3 1º semestre de 2017.

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. Respostas não justificadas não serão consideradas.
-

1. (2,5) Um computador, que apresenta uma arquitetura similar àquela apresentada ao longo do curso, possui uma capacidade máxima de memória principal de 64 M células, cada uma capaz de armazenar uma palavra de 8 bits. Em cada acesso à memória, realiza-se o acesso a uma célula. As instruções desta máquina são compostas de 2 campos: código de operação e endereço da célula de memória do operando. Sabe-se que o tamanho de uma instrução é 32 bits.

- a) (0,7) Calcule o número de códigos de operação diferentes que este computador pode ter.

Tamanho da instrução = código de operação + operando

operando = tamanho de endereço de uma célula = tamanho do barramento de endereço (BE)

operando = tamanho do BE = $\log_2 N = \log_2 64\text{Mcélulas} = \log_2 2^{26} = 26 \text{ bits}$

*Tamanho da instrução = código de operação + operando $\Rightarrow 32 = \text{código operação} + 26$
código de operação = 6bits , permite até $2^6 = 64$ códigos de operações diferentes*

- b) (0,6) Indique o número de acessos à memória necessários para se obter uma instrução.

O tamanho da palavra consiste de 8 bits. O barramento de dados terá que ter o tamanho para transferência de uma palavra, 8 bits. Assim, para a transferência de uma instrução de 32 bits, serão necessários 4 acessos.

- c) (1,2) Descreva detalhadamente a execução da instrução **MUL Op.** e **JN Op.** nesta máquina. A instrução **MUL Op.** multiplica o conteúdo da célula de memória cujo endereço é **Op.** pelo conteúdo do acumulador e armazena o resultado na memória no endereço **Op.** A instrução **JN Op.**, quando executada, carrega CI com o valor de **Op.** se o conteúdo do Acumulador é menor que zero, e caso contrário carrega CI com CI+1.

MUL Op

Passo 1: $RI \leftarrow (CI)$, ou seja, $RI \leftarrow$ recebe a Instrução armazenada no endereço contido no CI

Passo 2: $CI \leftarrow CI + 1$

Passo 3: Decodificação do código de operação

- recebe os bits do código de operação

- produz sinais para a execução da operação de multiplicação

Passo 4: Execução da operação

Passo 4.1: A CPU coloca no REM o valor do operando ($REM \leftarrow Op$), e é disponibilizado no barramento de endereço

Passo 4.2: A CPU aciona pelo barramento de controle a leitura de memória

Passo 4.3: A memória coloca o valor no barramento de dados, e consequência no RDM da CPU ($RDM \leftarrow MP(Op)$)

*Passo 4.4: A CPU executa a multiplicação 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: Inicia o procedimento de leitura da instrução armazenada no endereço que consta em CI

JN Op.

Passo 1: $RI \leftarrow (CI)$, ou seja, $RI \leftarrow$ recebe a Instrução armazenada no endereço contido no CI

Passo 2: $CI \leftarrow CI + 1$

Passo 3: Decodificação do código de operação

- recebe os bits do código de operação

- produz sinais para a execução da operação de salto condicional

Passo 4: Execução da operação

- UC emite sinal para transferir conteúdo do acumulador para UAL ($UAL \leftarrow ACC$)

- Executa operação de comparação

Se Resultado é verdadeiro, isto é, $ACC < 0$, $CI \leftarrow Op$

Passo 5: Inicia o procedimento de leitura da instrução armazenada no endereço que consta em CI

2. (2,5) Considere o conjunto de 32 bits representado na base hexadecimal (AC960000)₁₆. Mostre o que ele representa, **em decimal**, quando for interpretado como:

$$(AC960000)_{16} = 10101100\ 10010110\ 00000000\ 00000000_2$$

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

$$= 2^{31} + 2^{29} + 2^{27} + 2^{26} + 2^{23} + 2^{20} + 2^{18} + 2^{17} = 2.895.511.552$$

b) (0,3) um inteiro utilizando-se a representação sinal e magnitude.

$$= -(2^{29} + 2^{27} + 2^{26} + 2^{23} + 2^{20} + 2^{18} + 2^{17}) = -748.027.904$$

c) (0,5) um inteiro utilizando-se a representação em complemento a 2.

$$= -2^{31} + (2^{29} + 2^{27} + 2^{26} + 2^{23} + 2^{20} + 2^{18} + 2^{17}) = -1.399.455.744$$

- d) (0,8) um número utilizando-se a representação ponto flutuante precisão simples IEEE 754 (1 bit de sinal, 8 bits para expoente em excesso de 127 e 23 bits para mantissa).

$$10101100\ 10010110\ 00000000\ 00000000_2$$

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

$$\text{Expoente} = 01011001\ (\text{excesso de } 127) = 89 - 127 = -38$$

$$\text{Mantissa} = ,0010110\ 00000000\ 00000000$$

$$\text{Temos então: } 1, 0010110\ 00000000\ 00000000 \times 2^{-38}$$

$$= -(2^{-38} + 2^{-41} + 2^{-43} + 2^{-44}) = -4,26 \times 10^{-12}$$

- 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.

$$\text{maior valor positivo: } 0\ 11111110\ 111111111111111111111111 =$$

$$= 1, 111111111111111111111111 \times 2^{+127} = +3,40 \times 10^{+38}$$

$$\text{menor valor positivo: } 0\ 00000001\ 000000000000000000000000 =$$

$$= 1, 000000000000000000000000 \times 2^{-126} = +1,18 \times 10^{-38}$$

3. (2,5) Considere uma máquina que possa endereçar 1 Gbytes de memória física, utilizando endereço referenciando byte, e que tenha a sua memória organizada em blocos de 4K bytes. Ela possui uma memória cache que pode armazenar 2 K blocos, sendo um bloco por linha. Mostre o formato da memória cache, indicando os campos necessários (tag, bloco) e o número de bits para cada campo, e o formato de um

endereço da memória principal, indicando os bits que referenciam os campos da cache, para os seguintes mapeamentos:

a) Mapeamento direto.

Memória principal:

A máquina permite endereçar 1 Gbytes, como endereço referenciado a byte, temos $N = 1$ Gcélulas

$B = \text{Total de blocos} = 1 \text{ Gbytes} / 4K \text{ bytes/bloco} = 256 \text{ Kblocos}$

Tamanho do endereço da MP (E) $\Rightarrow N = 2^E \Rightarrow 1 \text{ Gbytes} = 2^{30} \Rightarrow E = 30 \text{ bits}$

Memória Cache

$Q = 2K \text{ blocos (1 bloco por linha)} = 2K \text{ linhas}$

Campos do endereço:

***Tag** = $B / Q = 256 \text{ Kblocos} / 2K \text{ linhas} = 128 \Rightarrow$ necessário 7 bits*

***Linha** = total de linhas = $Q = 2K \Rightarrow$ necessário 11 bits*

***Palavra** = total de 4k \Rightarrow necessário 12 bits*

Tag = 7 bits	No.linha = 11bits	Palavra = 12 bits
--------------	-------------------	-------------------

Endereço da MP = 30 bits

b) Mapeamento totalmente associativo.

Memória principal:

A máquina permite endereçar 1 Gbytes, como endereço referenciado a byte, temos $N = 1$ Gcélulas

$B = \text{Total de blocos} = 1 \text{ Gbytes} / 4K \text{ bytes/bloco} = 256 \text{ Kblocos}$

Tamanho do endereço da MP (E) $\Rightarrow N = 2^E \Rightarrow 1 \text{ Gbytes} = 2^{30} \Rightarrow E = 30 \text{ bits}$

Memória Cache

$Q = 2K \text{ blocos (1 bloco por linha)} = 2K \text{ linhas}$

Campos do endereço:

***Tag** = $B = 256 \text{ Kblocos} \Rightarrow$ necessário 18 bits*

***Palavra** = total de 4k \Rightarrow necessário 12 bits*

tag = 18 bits	Palavra = 12 bits
---------------	-------------------

Endereço da MP = 30 bits

4. (2,5) Para cada uma das situações abaixo, indique o modo de endereçamento mais adequado a ser utilizado:

a) Soma de dois vetores

Modo indexado é o mais adequado para trabalhar com vetores, na ausência deste modo o indireto é utilizado na operação com vetores.

b) Necessidade de realocação do programa

Modo base mais deslocamento é próprio para o uso na modificação de endereço de programas ou módulos destes.

c) Soma de duas variáveis inteiras

Modo direto é o utilizado neste caso.

O modo indireto pode ser usado para este fim, principalmente nos casos onde o tamanho limitado do operando não permita conter o endereço da variável.

d) Soma de uma variável inteira e uma constante

Para a variável inteira o modo direto é o indicado, mencionado no item anterior.

O modo imediato é o indicado para a constante.

e) Copiar um vetor de uma área de memória para outra

Como no item a, o modo indexado é o mais adequado para trabalhar com vetores e na ausência deste o modo indireto é utilizado nesta função.