



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Organização de Computadores

AP3 2º semestre de 2013.

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.

1. (2,5) Considere o seguinte conjunto de 32 bits (C1700000)₁₆. Indique o valor em decimal para este conjunto de bits quando considerarmos que ele está representando:

$$(C1700000)_{16} = (11000001\ 01110000\ 00000000\ 00000000)_2$$

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

$$+ 2^{31} + 2^{30} + 2^{24} + 2^{22} + 2^{21} + 2^{20} = 3.245.342.720$$

- b) (0,4) um inteiro em complemento a 2

$$- 2^{31} + (2^{30} + 2^{24} + 2^{22} + 2^{21} + 2^{20}) = -1.049.624.576$$

- c) (0,8) um número em ponto flutuante utilizando a representação IEEE 754 (1 bit para o sinal, 8 bits para o expoente representado em excesso de 127 e 23 bits para a mantissa fracionária)

$$(1\ 10000010\ 11100000000000000000000)_2$$

Sinal = 1 = negativo

Expoente = (10000010) = 130 - 127 = +3 (excesso de 127)

Mantissa = , 11100000000000000000000

Notação científica: -1, 111 x 2⁺³ = -1111 = -15

- d) (1,0) qual o menor e o maior valor positivo na representação do item anterior (c)?

Maior valor positivo:

$$0\ 11111110\ 111111111111111111111111$$

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

Menor valor positivo:

$$0\ 00000001\ 000000000000000000000000$$

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

2. (2,5) Suponha que você deve projetar uma máquina com as seguintes especificações:

- Capaz de endereçar 16 M células de memória principal, sendo que cada célula armazena 4 bytes.
- Deve possuir os registradores RDM (utilizado para enviar e receber dados para/de o barramento de dados), REM (utilizado para enviar endereços no barramento de endereços), CI (utilizado para indicar o endereço da instrução a ser lida da memória) e RI (utilizado para armazenar uma instrução), além de 8 registradores de rascunho.
- Cada instrução deve conter um código de operação e dois operandos. O primeiro operando é um endereço da memória principal e o segundo operando é o endereço de um registrador de rascunho.
- Deve poder ter um máximo de 32 códigos de operação diferentes.

a) (0,3) Indique qual deve ser o tamanho mínimo em bits do REM

*Memória com 16Mcélulas $\Rightarrow N = 16M$ células
tamanho mínimo do REM será o tamanho do barramento de endereços necessário
para endereçar toda a memória.
Barramento de endereços (BE) = $\log_2 N = \log_2 16M = 24$ bits
REM = tamanho do BE = 24 bits*

b) (0,3) Indique qual deve ser o tamanho mínimo em bits do do barramento de endereços.

tamanho do BE = 24 bits

c) (0,8) Calcule o número de células que uma instrução necessita para ser armazenada.

*Cada instrução = código de operação + 2 operandos
1o. operando = endereço de uma célula = 24 bits
2o. operando = endereço de um registrador de rascunho = 3 bits (total de 2^3 registradores)
cod.operação = tamanho necessário para 32 códigos diferentes = 5 bits
tamanho da instrução = $5 + 3 + 24 = 32$ bits
Como cada célula armazena 4 bytes (32 bits), uma instrução deverá ocupar 1 célula*

d) (0,5) Indique o **tamanho do RDM e do barramento de dados** de modo que a Unidade Central de Processamento obtenha uma instrução da memória principal realizando somente um acesso à memória principal.

*RDM = barramento de dados = tamanho necessário para transferir uma instrução
RDM = barramento de dados = 32 bits.*

e) (0,6) Calcule a capacidade de armazenamento em bits dos registradores RI e CI, utilizando-se os valores calculados nos itens anteriores.

*CI = tamanho necessário para endereçar toda a memória = 24 bits
RI = tamanho necessário para uma instrução = 32 bits*

3. (2,5) Considere uma máquina que possa endereçar 512 Mbytes de memória física, utilizando endereço referenciando byte, e que tenha a sua memória organizada em blocos de 16 bytes. Ela possui uma memória cache que pode armazenar 8K blocos, sendo um bloco por linha. Mostre o formato da memória cache, indicando os campos necessários (válido, 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:

- Mapeamento direto.

Memória principal:

A máquina permite endereçar 512 Mbytes, como endereço referenciado a byte, temos $N = 512 \text{ Mcélulas}$

$B = \text{Total de blocos} = 512 \text{ Mbytes} / 16 \text{ bytes/bloco} = 32 \text{ Mblocos}$

Tamanho do endereço da MP (E) $\Rightarrow N = 2^E \Rightarrow 512 \text{ Mcélulas} = 2^{29} \Rightarrow E = 29 \text{ bits}$

Memória Cache

$Q = 8 \text{ K blocos (1 bloco por linha)} = 8 \text{ K linhas}$

Campos do endereço:

***Tag** = $B / Q = 32 \text{ Mblocos} / 8 \text{ Klinhas} = 4 \text{ K} \Rightarrow \text{necessário } 12 \text{ bits}$*

***Linha** = total de linhas = $Q = 8 \text{ K} \Rightarrow \text{necessário } 13 \text{ bits}$*

***Palavra** = total de 16 $\Rightarrow \text{necessário } 4 \text{ bits}$*

<i>Tag = 12 bits</i>	<i>No.linha = 13bits</i>	<i>Palavra = 4 bits</i>
----------------------	--------------------------	-------------------------

Endereço da MP = 29 bits

- Mapeamento totalmente associativo.

Memória principal:

A máquina permite endereçar 512 Mbytes, como endereço referenciado a byte, temos $N = 512 \text{ Mcélulas}$

$B = \text{Total de blocos} = 512 \text{ Mbytes} / 16 \text{ bytes/bloco} = 32 \text{ Mblocos}$

Tamanho do endereço da MP (E) $\Rightarrow N = 2^E \Rightarrow 512 \text{ Mcélulas} = 2^{29} \Rightarrow E = 29 \text{ bits}$

Memória Cache

$Q = 8 \text{ K blocos (1 bloco por linha)} = 8 \text{ K linhas}$

Campos do endereço:

***Tag** = $B = 32 \text{ Mblocos} \Rightarrow \text{necessário } 25 \text{ bits}$*

***Palavra** = total de 16 $\Rightarrow \text{necessário } 4 \text{ bits}$*

<i>tag = 25 bits</i>	<i>Palavra = 4 bits</i>
----------------------	-------------------------

Endereço da MP = 29 bits

- Mapeamento associativo por conjunto, onde cada conjunto possui quatro linhas, cada uma de um bloco.

Memória principal:

A máquina permite endereçar 512 Mbytes, como endereço referenciado a byte, temos $N = 512 \text{ Mcélulas}$

$B = \text{Total de blocos} = 512 \text{ Mbytes} / 16 \text{ bytes/bloco} = 32 \text{ Mblocos}$

Tamanho do endereço da MP (E) $\Rightarrow N = 2^E \Rightarrow 512 \text{ Mcélulas} = 2^{29} \Rightarrow E = 29 \text{ bits}$

Memória Cache

$Q = 8 \text{ K blocos (1 bloco por linha)} = 8 \text{ K linhas}$

1 conjunto = 4 linhas (ou quadros) \Rightarrow

Total de conjuntos $\Rightarrow C = 8 \text{ K linhas} / 4 \text{ linhas por conjunto} \Rightarrow C = 2 \text{ K conjuntos}$

Campos do endereço

Tag = $B / C = 32 \text{ Mblocos} / 2 \text{ Kconjuntos} = 16K \Rightarrow$ necessário 14 bits

Conjunto = $C = 2K \Rightarrow$ necessário 11 bits

Palavra = total de 16 \Rightarrow necessário 4 bits

Tag = 14 bits	Conjunto = 11bits	Palavra = 4 bits
Endereço da MP = 29 bits		

4.(2,5) Considerando os diversos tipos de endereçamentos de instruções:

- di) Projete um mecanismo de endereçamento que permita que um conjunto arbitrário de 128 endereços, não necessariamente contíguos, em um grande espaço de endereçamento, seja especificável em um campo de 7 bits.

A solução será a utilização do endereçamento por registrador base mais deslocamento. Por exemplo, teríamos 2 bits para especificar um registrador e 5 bits para especificar um deslocamento. Poderíamos, assim, usar 4 registradores, cada um com até 32 deslocamentos possíveis, fornecendo 128 endereços diferentes.

- dii) Analise os modos de endereçamento base+deslocamento e indexado, estabelecendo diferenças de aplicação, vantagens e desvantagens de cada um.

O modo indexado tem como objetivo principal trabalhar com vetores, cujos valores são armazenados sequencialmente na memória e a sua localização pode ser referenciada por ponteiro. No modo indexado o endereço de cada elemento do vetor é obtido através da soma do valor do campo operando da instrução com o valor armazenado em um dos registradores do processador definido como registrador-índice. A indexação é empregada quando se deseja acessar diferentes dados, através de alteração de endereço, por incremento (ou decremento) do valor do registrador-índice. A vantagem dessa técnica reside na rapidez de execução das instruções de acesso aos dados, visto que a alteração do endereço dos elementos é realizada no próprio processador. Possui como desvantagem o uso de um registrador específico como registrador-índice e a limitação do tamanho do vetor de índice.

Exemplos de instruções modo indexado:

LDX Ri, Op $\Rightarrow ACC \leftarrow (Op + (Ri))$

ADX Ri, Op $\Rightarrow ACC \leftarrow ACC + (Op + (Ri))$

exemplo de programa:

MVI Ra, 5 \Rightarrow Registrador Ra \leftarrow 5 (contador do loop)

MVI Ri, 0 \Rightarrow Registrador de índice Ri \leftarrow 0 (1ª. Posição do vetor)

pos:

ADX Ri, 500 $\Rightarrow ACC \leftarrow ACC + (500 + (Ri))$ (Acrescenta no ACC o valor da posição Ri do vetor. Vetor este que tem início na posição 500 da memória)

INC Ri \Rightarrow incrementa o registrador de índice para posicionar no próximo elemento do vetor

DEC Ra \Rightarrow decrementa o contador de loop

JZ Ra, exit \Rightarrow Se Ra = 0, CI \leftarrow pos (se Ra=0, ir para exit)

JMP pos \Rightarrow Volta para o endereço: pos

exit:

O base mais deslocamento tem como seu principal objetivo permitir a modificação de endereço de programa ou módulos destes (que é a realocação de programa), bastando para isso uma única alteração no registrador base. O base mais deslocamento tem como característica o endereço a ser obtido da soma do deslocamento com o registrador base. Este modo tem a vantagem na redução do tamanho das instruções

economizando memória, e facilitando o processo de relocação dinâmica de programas. Possui a desvantagem do uso de um registrador específico como registrador base.

Exemplo: instrução base mais deslocamento:

LDB Rb, Op ==> (ACC) ← (Op + (Rb))
ADB Rb, Op ==> ACC ← ACC + (Op + (Rb))
JMP Rb, Op ==> CI ← (Op + (Rb))

exemplo de programa:

MVI Rb, 300 ==> Registrador base Rb ← 300 (registrador recebe o endereço inicial do módulo ou programa na memória)
JMP Rb, 100 ==> Salto incondicional para endereço do programa que corresponde a posição 100 a partir da posição inicial do programa, o endereço é obtido da soma do registrador base com o deslocamento será (Rb + 100) = 400 da memória principal. Caso este programa seja realocado na memória, o valor do registrador base deverá ser alterado.