



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Organização de Computadores**

**GABARITO-AP3 2º semestre de 2011.**

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. (1,0) Um computador possui uma capacidade máxima de memória principal com 256K células, cada uma capaz de armazenar uma palavra de 16 bits.

- a) Qual é o maior endereço em decimal desta memória?

$$\text{Maior endereço: } N - 1 = 256K - 1 = 262.143$$

- b) Qual é o tamanho do barramento de endereços deste sistema?

$$\begin{aligned} \text{Tamanho do barramento} &= e \\ \text{Como } e &= \log_2 N = \log_2 2^{18} = 18 \text{ bits} \\ \text{Então, tamanho do barramento de endereços} &= 18 \text{ bits} \end{aligned}$$

- c) Quantos bits podem ser armazenados no RDM e no REM?

$$\begin{aligned} \text{RDM} &= \text{tamanho da palavra} = 16 \text{ bits} \\ \text{REM} &= \text{barramento de endereços} = 18 \text{ bits} \end{aligned}$$

- d) Qual é o número máximo de bits que pode existir na memória?

$$\begin{aligned} \text{Total de bits} &= T \\ T &= N \times M \Rightarrow T = 2^{18} \times 16 = 2^{22} \Rightarrow T = 4.194.304 \text{ bits} \end{aligned}$$

2. (1,5) 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 32 bytes. Ela possui uma memória cache que pode armazenar 4K 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 512 Mbytes, como endereço referenciado a byte, temos  $N = 512 \text{ Mcélulas}$*

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

*$\text{Endereço} = E \Rightarrow N = 2^E \Rightarrow 512 \text{ Mcélulas} = 2^{29} \Rightarrow E = 29 \text{ bits}$*

*Memória Cache*

*$Q = 4 \text{ K blocos (1 bloco por linha)} = 4 \text{ K linhas}$*

*Campos do endereço:*

*$\text{Tag} = B / Q = 16 \text{ Mblocos} / 4 \text{ Klinhas} = 4 \text{ K} = 12 \text{ bits}$*

*$\text{No. da linha} = Q = 4 \text{ K} = 12 \text{ bits}$*

*$\text{End da palavra} = 32 = 5 \text{ bits}$*

<i>Tag = 12 bits</i>	<i>No.linha = 12bits</i>	<i>End da palavra 5 bits</i>
----------------------	--------------------------	----------------------------------

*Endereço = 29 bits*

b) Mapeamento totalmente associativo.

*Memória principal:*

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

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

*$\text{Endereço} = E \Rightarrow N = 2^E \Rightarrow 512 \text{ Mcélulas} = 2^{29} \Rightarrow E = 29 \text{ bits}$*

*Memória Cache*

*$Q = 4 \text{ K blocos (1 bloco por linha)} = 4 \text{ K linhas}$*

*Campos do endereço:*

*$\text{Tag} = B = 16 \text{ Mblocos} = 24 \text{ bits}$*

*$\text{End da palavra} = 32 = 5 \text{ bits}$*

<i>tag = 24 bits</i>	<i>End da palavra 5 bits</i>
----------------------	----------------------------------

3. (2,5) Responda:

a) (0,5) Dados os valores de memória abaixo e uma máquina de 1 endereço com um acumulador:

palavra 20 contém 40

palavra 30 contém 50

palavra 40 contém 60

palavra 50 contém 70

Quais valores as seguintes instruções carregam no acumulador?

-Load imediato 20

-Load direto 20

-Load indireto 20

- **LOAD IMEDIATO 20**

*Nesta instrução o valor a ser colocado no acumulador corresponde ao valor fornecido como operador, portanto:  $\text{ACC} \leftarrow 20$  (o valor a ser colocado no acumulador é 20)*

**- LOAD DIRETO 20**

Nesta instrução o valor a ser colocado no acumulador corresponde ao valor contido no endereço de memória fornecido como operador, portanto:

$ACC \leftarrow (20)$  (o valor a ser colocado no acumulador é 40)

**- LOAD INDIRETO 20**

Nesta instrução o valor a ser colocado no acumulador corresponde ao valor contido no endereço que consta como valor no endereço de memória fornecido como operador, portanto

$ACC \leftarrow ((20))$  (o valor a ser colocado no acumulador é 60)

- b) (1,0) Analise os modos de endereçamento direto, indireto e imediato, estabelecendo diferenças de desempenho, vantagens e desvantagens de cada um.

MODO DE ENDEREÇAMENTO	DEFINIÇÃO	VANTAGENS	DESvantagens	DESEMPENHO
<i>Imediato</i>	<i>O campo operando contém o dado</i>	<i>Rapidez na execução da instrução</i>	<i>Limitação do tamanho do dado. Inadequado para o uso com dados de valor variável</i>	<i>Não requer acesso a memória principal. Mais rápido que o modo direto</i>
<i>Direto</i>	<i>O campo operando contém o endereço do dado</i>	<i>Flexibilidade no acesso a variáveis de valor diferente em cada execução do programa</i>	<i>Perda de tempo, se o dado é uma constante</i>	<i>Requer apenas um acesso a memória principal. Mais rápido que o modo indireto</i>
<i>Indireto</i>	<i>O campo operando corresponde ao endereço que contém a posição onde está o conteúdo desejado,</i>	<i>Manuseio de vetores (quando o modo indexado não está disponível). Usar como "ponteiro"</i>	<i>Muitos acessos à MP para execução</i>	<i>Requer 2 acessos a memória principal</i>

- c) (1,0) Qual é o objetivo do emprego do modo de endereçamento base mais deslocamento? Qual é a diferença de implementação entre esse modo e o modo indexado?

O base mais deslocamento tem como seu principal objetivo permitir a modificação de endereço de programas ou módulos destes (que é a relocação de programa), bastando para isso uma única alteração no registrador base.

O base mais deslocamento tem como característica o endereço ser obtido da soma do deslocamento com o registrador base, diferindo do modo indexado onde o conteúdo do registrador base é fixo e varia-se o deslocamento, ao contrário deste onde o deslocamento é fixo e com a alteração do registrador base permite-se a mudança do endereço.

Exemplos de instruções modo indexado:

$LX\ R_i, Op \Rightarrow ACC \leftarrow ((Op) + (R_i))$

$AD\ R_i, Op \Rightarrow ACC \leftarrow ACC + ((Op) + (R_i))$

Exemplo: instrução base mais deslocamento:

$LDB\ R_b, Op \Rightarrow (ACC) \leftarrow ((Op) + (R_b))$

$ADB\ R_b, Op \Rightarrow ACC \leftarrow ACC + ((Op) + (R_b))$

4. (1,5) Um computador, que apresenta uma arquitetura similar àquela apresentada ao longo do curso, pode endereçar 2048 células de memória, sendo que cada célula de memória é capaz de armazenar uma palavra de 16 bits. Em cada acesso à memória, uma célula é acessada e realiza-se a transferência de uma palavra.

- a) (0,2) Qual é o tamanho do barramento de endereços deste sistema?

$$N = 2048 \text{ células} \Rightarrow N = 2^e \Rightarrow e = 11 \text{ bits} = \text{barramento de endereços}$$

- b) (0,3) Quantos bits podem ser armazenados no RDM (Registrador de Dados da Memória), no REM (Registrador de Endereços da Memória) e no CI (Contador de Instruções)?

*RDM = tamanho da palavra transmitida durante um acesso à memória = 1 palavra de 16 bits = 16 bits*

*REM = barramento de endereços = 11 bits*

*CI = tamanho do barramento de endereços = REM = 11 bits*

- c) (1,0) Descreva detalhadamente a execução da instrução **DIV Op.** nesta máquina, indicando como o REM, RDM, CI, barramentos de dados, de endereços e de controle são utilizados. A instrução **DIV Op.** divide o conteúdo da célula de memória cujo endereço é Op. pelo valor armazenado no acumulador e coloca o resultado no acumulador.

*Passo 1: A UCP coloca no REM o valor do operando ( REM <- Op ), que é disponibilizado no barramento de endereço*

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

*Passo 3: A memória coloca o valor no barramento de dados, que é transferido para o RDM da CPU ( RDM <- MP(Op) )*

*Passo 4: A UCP executa a divisão do valor recebido da memória pelo conteúdo do acumulador  
ACC <- RDM / ACC (ou ACC <- MP(Op) / ACC )*

*Passo 5: CI é incrementado (CI <- CI + 1) para apontar para a próxima instrução a ser lida.*

3. (2,0) Considere o conjunto de 32 bits representado na base hexadecimal (C1180000)<sub>16</sub>. Mostre o que ele representa, **em decimal**, quando for interpretado como:

**OBS: Não precisa fazer as contas, deixe-as indicadas.**

$$(C1180000)_{16} = (11000001000110000000000000000000)_2$$

- a) (0,2) um inteiro sem sinal

$$2^{31} + 2^{30} + 2^{24} + 2^{20} + 2^{19} = 3.239.575.552$$

- b) (0,4) um inteiro utilizando-se a representação sinal e magnitude

$$-(2^{30} + 2^{24} + 2^{20} + 2^{19}) = -1.092.091.904$$

- c) (0,6) um inteiro utilizando-se a representação em complemento a 2

$$-2^{31} + (2^{30} + 2^{24} + 2^{20} + 2^{19}) = -1.055.391.744$$

- 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, 23 bits para mantissa)

*1 10000010 001100000000000000000000*

*Sendo:*

*Sinal = 1 => negativo*

*Expoente = 10000010 = 130 - 127 => expoente = +3*

*Mantissa = 001100000000000000000000*

$$\text{Temos então } \Rightarrow -1,001100000000000000000000 \times 2^{+3} = (-1001,1)_2 = (-9,5)_{10}$$

4. (1,5) Descreva os três possíveis métodos de comunicação entre uma interface de entrada e saída com a unidade central de processamento e memória principal: por programa (*polling*), por interrupção e por acesso direto à memória.

*Por programa: A UCP indica à interface de entrada e saída que deseja realizar uma operação de transferência de dados e fica interrogando a interface para saber se ela está pronta para realizar a transferência de dados. Quando a UCP recebe uma resposta positiva da interface, ela realiza a transferência de dados. Para ler dados da interface e colocar os dados na memória, ela realiza operações de leitura de dados da interface e escrita na memória. Para escrever dados na interface, ela realiza operações de leitura da memória e escrita na interface.*

*Por interrupção: A UCP indica à interface de entrada e saída que deseja realizar uma operação de transferência de dados e realiza outras instruções que não se referenciam a esta operação, ou seja, a UCP não fica interrogando a interface para identificar quando ela está pronta. Quando a interface está pronta para realizar a transferência, ela gera um sinal de interrupção que é recebido pela UCP. A UCP ao receber este sinal, termina de realizar a instrução que estava sendo realizada, salva o contexto onde esta instrução estava sendo realizada, e executa as instruções para realizar a transferência de dados com a interface.*

*Por acesso direto à memória (DMA) : Um controlador de DMA realiza diretamente a transferência de dados entre a interface e a memória sem envolver a UCP nesta transferência. A UCP necessita enviar alguns parâmetros para o controlador de DMA: o endereço da interface, o tipo de transferência (escrita ou leitura de dados), o endereço de memória para ler ou escrever os dados e o número de bytes a serem transferidos. O controlador de DMA realiza toda a transferência de dados entre a interface e a memória e a UCP não necessita executar nenhuma instrução para realizar esta transferência. Quando a transferência acaba, o controlador de DMA gera um sinal de interrupção para a UCP indicando que a transferência foi realizada*