



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 2009.**

**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. (2,0) 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?

$$N = 256K \text{ células} = 2^{18}$$

$$\text{Último endereço} = N - 1 = 262144 - 1 = \mathbf{262.143}$$

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

$$\text{Barramento de endereços} = E$$

$$N = 2^E = 2^{18}, \text{ portanto } E = 18, \text{ Barramento de endereços} = \mathbf{18 \text{ bits}}$$

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

$$\text{O REM terá que ter o tamanho do barramento de endereços} = \mathbf{18 \text{ bits}}$$

$$\text{RDM} = \text{tamanho da palavra} = \mathbf{16 \text{ bits}}$$

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

$$T = \text{capacidade em bits}$$

$$T = N \times M = 256K \text{ células} \times 16 \text{ bits/célula} = \mathbf{4096K \text{ bits}}$$

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:*

$N = 512 \text{ Mbytes}$ , como endereço referenciado byte, temos 512 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*

$Q = 4 \text{ K linhas (ou quadros):}$

*Campos do endereço:*

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

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

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

Tag = 12 bits	No.linha = 12bits	End da palavra 5 bits
---------------	-------------------	--------------------------

Endereço = 29 bits

b) Mapeamento totalmente associativo.

*Memória principal:*

$N = 512 \text{ Mbytes}$ , como endereço referenciado byte, temos 512 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*

$Q = 4 \text{ K linhas (ou quadros):}$

*Campos do endereço:*

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

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

tag = 24 bits	End da palavra 5 bits
---------------	--------------------------

3. (1,5) 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:  $ACC \leftarrow 20$  ( 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)

4. (2,0) Um computador, que apresenta uma arquitetura similar àquela apresentada ao longo do curso, possui uma capacidade máxima de memória principal de 128 M células, cada uma capaz de armazenar uma palavra de 16 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) (0,5) Calcule o número de códigos de operação diferentes que este computador pode ter.

$$N = 128M \text{ células} = 2^{27}$$

Tamanho da instrução = código de operação + operando (memória)

$$32 = \text{cód.oper} + 27 \Rightarrow \text{cód.oper.} = 5 \text{ bits}$$

Quantidade de códigos de operação diferentes =  $2^5 = 32$  códigos

- b) (0,5) Indique o número de acessos à memória necessários para se obter uma instrução.

Cada acesso a memória obtém-se 16 bits

Para obter uma instrução (32bits) são necessários 2 acessos à memória

- c) (1,0) Descreva detalhadamente a execução da instrução **MUL 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.**

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

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

Passo 3: A memória coloca o valor no barramento de dados, que é armazenado no RDM da CPU (  $RDM \leftarrow MP(Op)$  )  
 Passo 4: A CPU executa a multiplicação do valor recebido com o acumulador armazenando o resultado no acumulador;  $ACC \leftarrow ACC * RDM$   
 Passo 5: A CPU coloca o conteúdo do acumulador no RDM (  $RDM \leftarrow ACC$  )  
 Passo 6: A CPU coloca no REM o valor do operando (  $REM \leftarrow Op$  ), que é disponibilizado no barramento de endereços  
 Passo 7: A CPU coloca o conteúdo do RDM no barramento de dados  
 Passo 8: A CPU aciona pelo barramento de controle a escrita de memória (  $MP(Op) \leftarrow RDM$  )

5. (1,5) Considere o conjunto de 32 bits representado na base hexadecimal  $(FE100000)_{16}$ . Mostre o que ele representa, **em decimal**, quando for interpretado como:  
**OBS: Não precisa fazer as contas, deixe-as indicadas.**

$$(FE100000)_{16} = (1111\ 1110\ 0001\ 0000\ 0000\ 0000\ 0000\ 0000)_2$$

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

$$2^{31} + 2^{30} + 2^{29} + 2^{28} + 2^{27} + 2^{26} + 2^{25} + 2^{20} = 4.262.461.440$$

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

$$-(2^{30} + 2^{29} + 2^{28} + 2^{27} + 2^{26} + 2^{25} + 2^{20}) = -2.114.977.792$$

- c) (0,3) um inteiro utilizando-se a representação em complemento a 2.

$$-2^{31} + (2^{30} + 2^{29} + 2^{28} + 2^{27} + 2^{26} + 2^{25} + 2^{20}) = -32.505.856$$

- d) (0,6) 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).

*sinal* = 1 (negativo)

*expoente* = 11111100 (por excesso de 127) =  $252 - 127 = +125$

*mantissa* = 00100000000000000000000

$$\text{normalizado} = -(1, 001000000000000000000000)_2 \times 2^{125} =$$

$$\text{resposta: } -(2^{125} + 2^{122})$$

6. (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.*