



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 2010.**

1) (2,0) Um computador, que apresenta uma arquitetura similar àquela apresentada ao longo do curso, possui uma capacidade máxima de memória principal de 256 M células, cada uma capaz de armazenar uma palavra de 8 bits. Em cada acesso à memória, realiza-se o acesso **a duas células**. 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.

$N = 256M \text{ células} = 2^{28} \text{ células} \Rightarrow E = 28 = \text{barramento de endereços} = \text{REM}$   
 $\text{Tamanho da instrução} = \text{cód.oper.} + \text{operando (endereço de memória)} \Rightarrow 32\text{bits} = \text{cód.oper} + 28$   
 $\text{Cód.oper} = 4 \text{ bits}$   
 $\text{Número de códigos de operação} = 2^4 = \mathbf{16}$

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

*Como cada célula pode armazenar uma palavra de 8 bits, e cada acesso são transferidas 2 células. Então, para transferir uma instrução de 32 bits, serão necessários 2 acessos*

c) (1,0) Descreva detalhadamente a execução da instrução **ADD Op.** nesta máquina. A instrução **ADD Op.** soma o conteúdo da célula de memória cujo endereço é **Op.** com o conteúdo do acumulador e armazena o resultado na memória no endereço **Op.**

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

*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 armazenado no RDM da UCP ( RDM <- MP(Op) )*

*Passo 4: A UCP executa a soma do valor recebido com o valor armazenado no acumulador; ACC <- ACC + RDM (ou ACC <- ACC + MP(Op) )*

*Passo 5: A UCP coloca no REM o valor do operando ( REM <- Op ), que é disponibilizado no barramento de endereços*

*Passo 6: A UCP coloca o valor do ACC no RDM, cujo conteúdo é disponibilizado no barramento de dados*

*Passo 7: A UCP aciona pelo barramento de controle a escrita de memória*

2. (1,5) Considere o conjunto de 32 bits representado na base hexadecimal (C0400000)<sub>16</sub>. Mostre o que ele representa, **em decimal**, quando for interpretado como:

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

$$(C0400000)_{16} = (11000000\ 01000000\ 00000000\ 00000000)_2$$

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

$$2^{31} + 2^{30} + 2^{22} = 3.225.419.776$$

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

$$-(2^{30} + 2^{22}) = -1.077.936.128$$

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

$$-2^{31} + (2^{30} + 2^{22}) = -1.069.547.520 \text{ ou } -(2^{29} + 2^{28} + 2^{27} + 2^{26} + 2^{25} + 2^{24} + 2^{23} + 2^{22}) = -1.069.547.520$$

e) (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).

*Resposta: O conjunto de bits na base 2 corresponde a:*  
11000000 01000000 00000000 00000000

*Bit de sinal = 1, negativo*

*Representação do expoente: 10000000 em excesso de 127 => expoente + 127 = 128,  
Expoente = 128 - 127 = +1*

*Mantissa fracionária = 100000000000000000000000  
-(1,100000000000000000000000) × 2<sup>+1</sup> = -11,0000000000 × 2<sup>0</sup> = -(11)<sub>2</sub>  
Resultado = -3*

3. (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 Unidade Central de Processamento (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 (ADM) : Um controlador de ADM 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 ADM: 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 ADM 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 ADM gera um sinal de interrupção para a UCP indicando que a transferência foi realizada*

4. (2,0) Um computador, possui uma capacidade máxima de memória principal de 2G células, cada uma capaz de armazenar uma palavra de 8 bits. Em cada acesso à memória, realiza-se a transferência de duas palavras (**você pode deixar as contas indicadas**).

a) (0,5) Qual é o maior endereço em decimal desta memória?

$$N = 2G \text{ células} = 2^{31}$$

$$\text{Último endereço} = 2^{31} - 1 = 2.147.483.648 - 1 = \mathbf{2.147.483.647}$$

b) (0,5) Qual é o tamanho do barramento de endereços deste sistema?

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

$$N = 2^E = 2^{31}, \text{ portanto } E = 31, \text{ Barramento de endereços} = \mathbf{31 \text{ bits}}$$

c) (0,5) 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)?

*O REM terá que ter o tamanho do barramento de endereços = **31 bits***  
*CI terá o tamanho mínimo necessário para endereçar toda a memória = **31 bits***  
*RDM = tamanho necessário para armazenar o número de bits transferidos em um acesso à memória (duas palavras) = **16 bits***

d)(0,5) Qual é o número máximo de bits que pode existir na memória?

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

$$T = N \times M = 2G \text{ células} \times 8\text{bits/célula} = \mathbf{16Gbits}$$

5. (2,0) Considere uma máquina que possa endereçar 256 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 4K blocos, sendo um bloco por linha (ou quadro). 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:*

$N = 256 \text{ Mbytes}$ , como endereço referenciado byte, temos 256 Mcélulas

$B = \text{Total de blocos} = 256 \text{ Mbytes} / 16 \text{ bytes/bloco} = 16 \text{ Mblocos}$

$\text{Endereço} = E \Rightarrow N = 2^E \Rightarrow 256 \text{ Mcélulas} = 2^{28} \Rightarrow E = 28 \text{ bits}$

*Memória Cache*

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

*Campos do endereço:*

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

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

$\text{End da palavra} = 16 = 4 \text{ bits}$

$\text{Tag} = 12 \text{ bits}$	$\text{No. linha} = 12 \text{ bits}$	$\text{End da palavra} = 4 \text{ bits}$
--------------------------------	--------------------------------------	--

Endereço = 28 bits

b. Mapeamento totalmente associativo.

*Memória principal:*

$N = 256 \text{ Mbytes}$ , como endereço referenciado byte, temos 256 Mcélulas

$B = \text{Total de blocos} = 256 \text{ Mbytes} / 16 \text{ bytes/bloco} = 16 \text{ Mblocos}$

$\text{Endereço} = E \Rightarrow N = 2^E \Rightarrow 256 \text{ Mcélulas} = 2^{28} \Rightarrow E = 28 \text{ bits}$

*Memória Cache*

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

*Campos do endereço:*

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

$\text{End da palavra} = 16 = 4 \text{ bits}$

$\text{Tag} = 24 \text{ bits}$	$\text{End da palavra} = 4 \text{ bits}$
--------------------------------	--

6. (1,0) Explique a classificação das arquiteturas segundo Flynn.

**SISD** - *Single instruction stream, single data stream*. Um único processador executa uma única sequência de instruções sobre dados armazenados em uma única memória. Exemplo: Processadores de computadores pessoais (núcleo simples)

**SIMD** – *Single instruction stream, multiple data stream*. Vários elementos de processamento. Cada um tem uma memória de dados. Cada instrução é executada sobre um conjunto de dados diferente. Exemplo: Processadores matriciais.

**MISD** – *Multiple instruction stream, single data stream*. A sequência de dados é transmitida para um conjunto de processadores, cada um dos quais executa uma sequência de instruções diferente. Não existem processadores comerciais que utilizam este modelo.

**MIMD** – *Multiple instruction stream, multiple data stream*. Conjunto de processadores executa simultaneamente sequências diferentes de instruções sobre conjuntos de dados diferentes. Exemplo: SMPs, clusters, sistemas NUMA.