



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 2008.**

1. (2,5) Um computador, que apresenta uma arquitetura similar àquela apresentada ao longo do curso, possui uma capacidade máxima de memória principal de 512K células, cada uma capaz de armazenar uma palavra de 16 bits. Em cada acesso à memória, realiza-se a transferência de uma palavra.

- a) (0,3) Qual é o maior endereço em decimal desta memória?

*O maior endereço em decimal é  $2^{19} - 1 = 524\,287$*

- b) (0,3) Qual é o tamanho do barramento de endereços deste sistema?

*Para endereçar 512 K células =  $2^{19}$  células, precisamos de 19 bits para endereçar a memória. Logo o barramento de endereços deverá ter 19 bits.*

- c) (0,9) Quantos bits são armazenados no RDM (Registrador de Dados da Memória), no REM (Registrador de Endereços da Memória) e no CI (Contador de Instruções) ?

*Como em cada acesso à memória se realiza a transferência de uma palavra que possui 16 bits, o RDM deve poder armazenar 16 bits.*

*O REM e CI têm que armazenar o número de bits necessários ao endereço, logo devem armazenar 19 bits*

- d) (0,3) Qual é o número máximo de bits que pode existir na memória?

*O número máximo de bits é igual ao número de células multiplicado pela quantidade de bits de uma célula que é igual a  $512K \times 16 = 2^{19} \times 2^4 \Rightarrow 2^{23} \text{ bits} = 8 \text{ Mbits}$*

- e) (0,7) Descreva detalhadamente a execução da instrução **ADD Op.** nesta máquina. A instrução **ADD Op.** soma o conteúdo do acumulador com o conteúdo da célula de memória cujo endereço é **Op.** e armazena o resultado na memória no endereço **Op.**

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

*Passo 2: A CPU envia o sinal de leitura de memória pelo barramento de controle.*

*Passo 3: A memória coloca os bits no barramento de dados, que são armazenados no RDM da CPU ( RDM <- MP(Op) ).*

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

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

*Passo 6: A CPU coloca no RDM o valor acumulado no acumulador, que é disponibilizado no barramento de dados.*

*Passo 7: A CPU envia o sinal de escrita de memória pelo barramento de controle.*

*Passo 8: A memória armazena os bits recebidos no barramento de dados na célula endereçada pelos bits do barramento de endereços.*

2. (1,0) Considere o conjunto de 32 bits representado na base hexadecimal  $(C0801000)_{16}$ . Mostre o que ele representa, **em decimal**, quando for interpretado como:

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

$$(C0801000)_{16} \Rightarrow (11000000\ 10000000\ 00010000\ 00000000)_2$$

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

$$2^{31} + 2^{30} + 2^{23} + 2^{12} = 3.229.618.176$$

- b) (0,2) um inteiro utilizando-se a representação sinal e magnitude

$$-(2^{30} + 2^{23} + 2^{12}) = -1.082.134.528$$

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

$$-2^{31} + (2^{30} + 2^{23} + 2^{12}) = -1.065.349.120$$

- d) (0,4) 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 10000000 00010000 00000000*

*Bit de sinal = 1, número negativo*

*Representação do expoente: 10000001 em excesso de 127  $\Rightarrow$  expoente + 127 = 129,*

*Expoente = +2*

*Mantissa fracionária = 000000000010000000000000*

$$-(1,00000000001) \times 2^{+2} = -(2^0 + 2^{-11}) \times 2^{+2} = -(2^2 + 2^{-9})$$

*Resultado = -4,001953125*

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. Indique as vantagens e desvantagens de cada um.

*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. As vantagens deste método são: hardware simples e todos os procedimentos estão sobre controle da UCP. As desvantagens são: utilização do processador para interrogar as interfaces, o que acarreta perda de ciclos de processador que poderiam ser utilizados na execução de outras instruções e utilização do processador para realizar a transferência de dados, o que também acarreta perda de ciclos de processador.*

*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. A vantagem deste método é que não ocorre perda de ciclos de processador para interrogar a interface, já que neste caso, não se precisa mais interrogar a interface, ela avisa quando está pronta. As desvantagens são: necessidade de um hardware adicional (controlador de interrupções, por exemplo), gerenciamento de múltiplas interrupções e perda de ciclos de relógio para salvar e recuperar o contexto dos programas que são interrompidos.

*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. As vantagens deste método são: permite transferência rápida entre interface e memória porque existe um controlador dedicado a realizá-la e libera a UCP para executar outras instruções não relacionadas a entrada e saída. A desvantagem é que precisamos de hardware adicional.

4. (1,0) Uma memória associativa por conjunto consiste em 64 quadros divididos em conjuntos de 4 quadros cada. A memória principal contém 4K blocos de 128 palavras cada um. Mostre o formato de um endereço de MP.

*Memória principal:*

$B = \text{Total de blocos} = 4 \text{ Kblocos}$

$\text{Tamanho da palavra} = \text{tamanho do bloco}$

$N = B(\text{quant. de blocos}) \times K (\text{palavras por bloco}) \Rightarrow N = 4 \text{ Kblocos} \times 128 \text{ palavras/bloco}$

$N = 512 \text{ Kpalavras ou } 512 \text{ Kcélulas}$

$\text{Endereço} = E \Rightarrow N = 2^E \Rightarrow 512 \text{ Kcélulas} = 2^{19} \Rightarrow E = 19 \text{ bits}$

*Memória Cache*

$Q = 64 \text{ quadros (ou linhas):}$

$C = 64 \text{ blocos} / 4 \text{ blocos/conjunto} \Rightarrow C = 16 \text{ conjuntos}$

*Campos do endereço:*

$\text{Tag} = B / C = 4 \text{ Kblocos} / 16 \text{ conjuntos} = 256 = 8 \text{ bits}$

$\text{No. do conjunto} = C = 16 \text{ conjuntos} \Rightarrow C = 4 \text{ bits}$

$\text{End da palavra} = 128 = 7 \text{ bits}$

|                     |                              |                                |
|---------------------|------------------------------|--------------------------------|
| <b>Tag = 8 bits</b> | <b>No. conjunto = 4 bits</b> | <b>End da palavra = 7 bits</b> |
|---------------------|------------------------------|--------------------------------|

5. (1,5) Considere um sistema de computação que possui uma memória principal (RAM) com capacidade máxima de endereçamento de 64K células, sendo que cada célula armazena um byte de informação. Para criar um sistema de controle e funcionamento da sua memória cache, a memória RAM é constituída de blocos de 8 bytes cada. A memória cache do sistema é do tipo **mapeamento direto**, contendo 32 linhas. Pergunta-se:

- a. Como seria organizado o endereço da MP (RAM) em termos de etiqueta (tag), número de linha e do byte dentro da linha?

*Memória Principal*

$\Rightarrow \text{Tamanho da memória (em bytes)} = 64 \text{ Kcélulas, cada } 1 \text{ célula armazena } 1 \text{ byte,}$

$$\text{Capacidade da cache} = 32 \text{ linhas} \times 8 \text{ bytes} = 256 \text{ bytes ou } 2^8 \text{ bytes}$$

6. (1,0) Considerando os diversos tipos de endereçamentos de instruções, 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.

*Uma solução seria usar 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.*

7. (1,5) Descreva as categorias da classificação de 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.*

*SIMD - Single instruction stream, multiple data stream. Uma única unidade de controle. 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*