



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 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. (1,8) Um computador, que apresenta uma arquitetura similar àquela apresentada ao longo do curso, pode ter o endereço 2047 ($2047=2^{11}-1$) como o maior endereço decimal de memória, sendo que cada célula de memória é capaz de armazenar uma palavra de 8 bits. Em cada acesso à memória, realiza-se a transferência de uma palavra.

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

$$N = 2^e \text{ ultimo endereço} + 1 = 2^{11} - 1 + 1 = 2^{11} \text{ células} \Rightarrow \\ e = 11\text{bits} = \text{barramento de endereços}$$

- b) (0,6) 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)?

$$REM = \text{barramento de endereços} = 11 \text{ bits}$$

$$RDM = \text{tamanho da palavra transmitida durante um acesso à memória} = 8 \text{ bits}$$

$$CI = \text{tamanho do barramento de endereços} = REM = 11 \text{ bits}$$

- c) (0,4) Qual é o número máximo de bits que pode existir na memória?

$$\text{Tamanho máximo em bits} = T = N \times M \Rightarrow T = 2048 \text{ células} \times 8 \text{ bits/célula} \Rightarrow \\ T = 16.384 \text{ bits ou } 16\text{Kbits}$$

- d) (0,6) Descreva detalhadamente a execução da instrução **MUL Op.** nesta máquina, indicando como o REM, RDM, CI, barramentos de dados, de endereços e de controle são utilizados. A instrução **MUL Op.** multiplica o conteúdo da célula de memória cujo endereço é Op. com o valor armazenado no acumulador e coloca o resultado no acumulador.

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 conteúdo da célula de memória cujo endereço é Op. 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 ($ACC \leftarrow ACC * RDM$ ou $ACC \leftarrow -ACC * MP(Op)$)

Passo 5: CI é incrementado ($CI \leftarrow CI + 1$) para apontar para a próxima instrução a ser lida.

2. (1,4) Considere o conjunto de 32 bits representado na base hexadecimal $(BFA00000)_{16}$. Mostre o que ele representa, **em decimal**, quando for interpretado como:

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

$$(BFA00000)_{16} = (10111111101000000000000000000000)_2$$

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

$$2^{31} + 2^{29} + 2^{28} + 2^{27} + 2^{26} + 2^{25} + 2^{24} + 2^{23} + 2^{21} = 3.214.934.016$$

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

$$-(2^{29} + 2^{28} + 2^{27} + 2^{26} + 2^{25} + 2^{24} + 2^{23} + 2^{21}) = -1.067.450.368$$

- c) (0,4) um inteiro utilizando-se a representação em complemento a 2

$$-2^{31} + (2^{29} + 2^{28} + 2^{27} + 2^{26} + 2^{25} + 2^{24} + 2^{23} + 2^{21}) = -1.080.033.280$$

- 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)

$$1 \ 01111111 \ 010000000000000000000000$$

Sendo:

$$\text{Sinal} = 1 \Rightarrow \text{negativo}$$

$$\text{Bits do Expoente} = 01111111 \Rightarrow 127 - 127 = 0 \Rightarrow \text{expoente} = 0$$

$$\text{Mantissa} = 010000000000000000000000$$

$$\text{Temos então} \Rightarrow -1,010000000000000000000000 \times 2^0 = (-1,01)_2 = (-1,25)_{10}$$

3. (1,8) 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 uma vantagem e uma desvantagem 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.

Vantagem: Método simples de implementar.

Desvantagem: A UCP fica ocupada interrogando o dispositivo, enquanto poderia estar executando outras instruções.

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.

Vantagem: A UCP não fica ocupada interrogando o dispositivo, e pode executar outras instruções.

Desvantagem: A UCP ainda fica responsável por realizar a transferência de dados entre memória e dispositivo.

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

Vantagem: A UCP não fica ocupada realizando transferência de dados entre memória e dispositivo e pode executar outras instruções.

Desvantagem: Necessita de um hardware adicional, o controlador de DMA.

4. (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 da linha e do byte dentro de uma linha?

Memória Principal

=> Tamanho da memória (em bytes) = 64K células, cada 1 célula armazena 1 byte, temos $N = 64K$ células (ou 64 Kbytes)

=> Será organizada em blocos de 8bytes, como 1 célula = 1byte, temos cada bloco = 8 células, $K = 8$

=> $N = 64K$ células e $K = 8$ células por bloco, o total de blocos da MP (B) será:

Total de blocos: $B = N / K \Rightarrow B = 64K \text{ células} / 8 \text{ células por bloco} \Rightarrow B = 8K \text{ células}$

Memória Cache

=> O K (quantidade de células/bloco) tem de ser igual a MP.

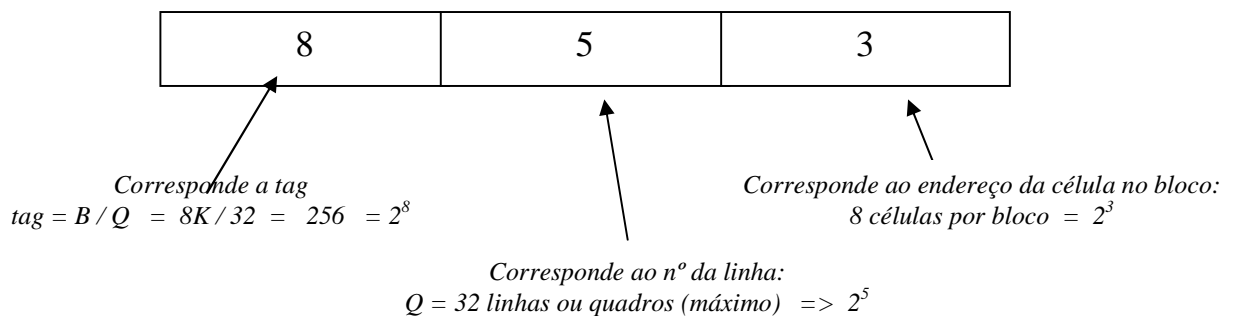
=> Tamanho da memória cache (em blocos ou linhas) => $Q = 32$ linhas

=> Tamanho da memória cache em células = $Q \times K = 32 \text{ linhas} \times 8 \text{ células/linha} = 256$ células (ou 256 bytes)

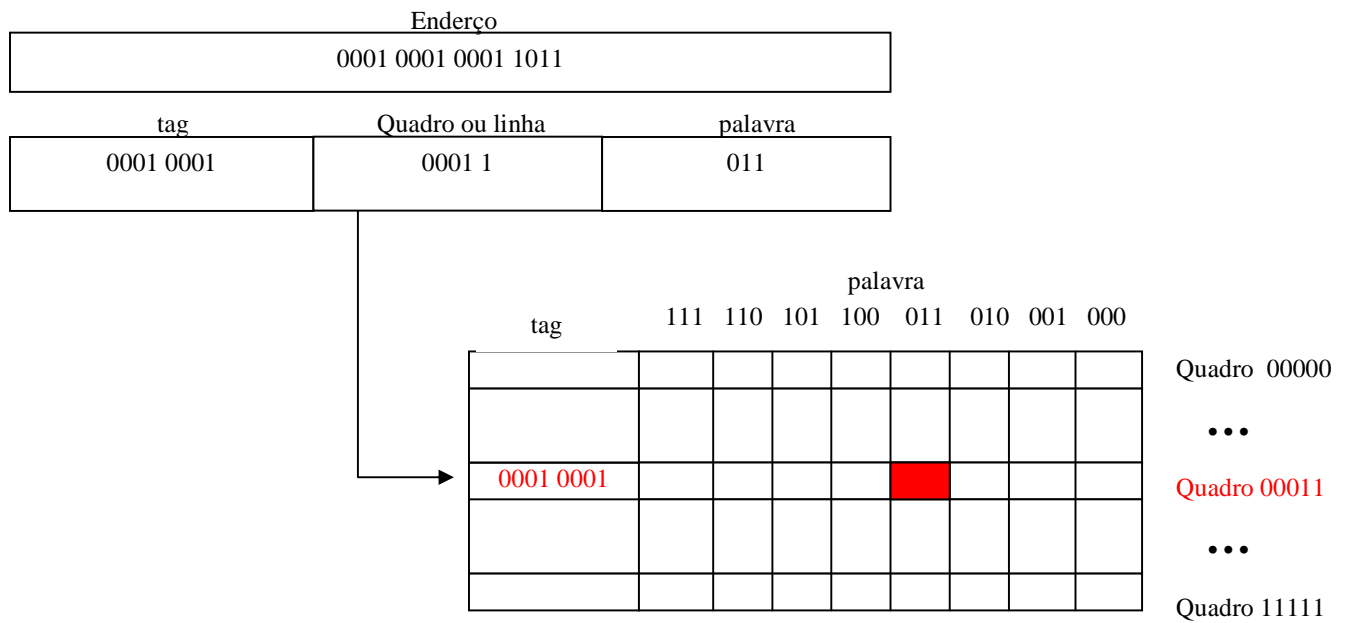
Memória principal

=> Para endereçarmos toda a MP precisamos da seguinte quantidade de bits (E)

sendo $N = 2^E \Rightarrow N = 64K \text{ células} \Rightarrow N = 2^{16} \Rightarrow E = 16 \text{ bits}$



- b) Em que linha estaria contido o byte armazenado no seguinte endereço da MP: 0001 0001 0001 1011?



- c) Qual é a capacidade da memória cache em bytes?

Capacidade da cache = $Q \times K = 32 \text{ linhas} \times 8 \text{ palavras/linha}$, considerando neste problema, tamanho da palavra = tamanho da célula e tamanho da célula = 1 byte, então:

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

5. (2,0) Responda:

a) Por que um programa em linguagem Assembly não é diretamente executável pelo processador? Como esse problema é resolvido?

Porque o processador entende apenas a linguagem de máquina que é composta de 0 e 1s, a linguagem assembly representa a linguagem de máquina sob a forma de mnemônicos para que possa ser melhor compreendida ao ser humano.

Para ser executável, um programa elaborado em assembly deverá passar pelo processo de montagem, que consiste em traduzir cada instrução (mnemônicos) para seu respectivo em linguagem binária (máquina)

b) Por que um compilador deve ser específico para uma determinada linguagem de programação e para uma determinada UCP?

Apesar das linguagens terem em comum suas principais estruturas de programação (decisões, loops, procedimentos), diferenciam nas formas léxicas, sintáticas e semânticas, daí a necessidade de cada linguagem ter um compilador próprio. Todavia, para uma mesma linguagem, o compilador poderá diferenciar devido ao conjunto de instruções da CPU a que foi destinado. Outro importante fator é o Sistema operacional. Sistemas diferentes, em geral possuem “chamadas ao sistema” distintas, necessitando de compiladores próprios para aquele sistema operacional.

6. (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? Explique.

-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)