

Disciplina: Organização de Computadores
GABARITO DA AP3 2019-01

1. (3,5) Considere uma máquina com arquitetura semelhante à arquitetura vista no curso, que apresente as seguintes especificações:

- Capaz de endereçar 128 M células de memória principal.
- Deve possuir um registrador Acumulador, além do RDM (Registrador de Dados da Memória), REM (Registrador de Endereços da Memória), CI (Contador de Instrução) e RI (Registrador de Instrução).
- O conjunto de instruções de linguagem de máquina deve ter 31 instruções.
- Cada instrução deve conter um código de operação e um operando como mostrado abaixo, onde o operando indica um endereço de memória e um registrador



- a) (0,3) Calcule o tamanho mínimo em bits do REM e do barramento de endereços.

Capacidade de endereçar 128M células => $N = 128M$ células

tamanho mínimo do REM será o tamanho do barramento de endereços necessário para atender a capacidade de endereçamento de 128M células

Tamanho do Barramento de endereços (BE) = $\log_2 N = \log_2 128M = \log_2 2^{27} = 27$ bits

REM = tamanho do BE = 27 bits

- b) (0,3) Calcule o tamanho mínimo em bits que a instrução deve ter.

Cada instrução = código de operação + 1 operando

código operação = necessário para 31 instruções =>

cod.oper = 5 bits que permite até 32 instruções diferentes

operando = endereço de memória = 27 bits

Então, código de operação + oper. = $5 + 27 = 32$ bits

Tamanho mínimo da instrução é 32 bits

- c) (0,8) Para o valor calculado no item b, indique o tamanho em bits de cada célula da memória principal, o tamanho do RDM e o barramento de dados de modo que a Unidade Central de Processamento obtenha uma instrução da memória principal realizando somente um acesso à memória principal.

Tamanho do RDM = tamanho do barramento de dados

Barramento de dados, em computadores atuais, tem tamanho múltiplo de uma palavra.

A célula tem o tamanho de uma palavra, salvo se possuir outra orientação na questão.

Nesta arquitetura, o barramento de dados tem de ter o tamanho suficiente para transferir uma instrução por vez

Concluindo,

RDM = barramento de dados = tamanho de uma instrução = 32 bits

Barramento de dados transfere no mínimo uma palavra, palavra = 32 bits

A célula tem o tamanho de uma palavra = 32 bits.

- d) (0,6) Calcule o tamanho de RI e CI utilizando-se os valores calculados nos itens anteriores.

CI = tamanho mínimo necessário para endereçar toda a memória = 27 bits

RI = tamanho necessário para uma instrução = 32 bits

- e) (0,3) Calcule a capacidade de armazenamento, em bits, da memória desta máquina.

Total de bits = T

$$T = N \times M \Rightarrow T = 128 \text{ M células} \times 32 \text{ bits/célula} \Rightarrow T = 4096 \text{ Mbits ou } 4 \text{ Gbits ou } 512 \text{ MBytes}$$

- f) (0,6) Descreva detalhadamente a execução da instrução **LDA Op.** nesta máquina. A instrução **LDA Op.** carrega o acumulador com o conteúdo da célula de memória cujo endereço é Op.

Passo 1: A CPU coloca no REM o valor do operando ($REM \leftarrow Op$) e é disponibilizado no barramento de endereço ;

Passo 2: A CPU aciona pelo barramento de controle o sinal de leitura de memória ;

Passo 3: A memória coloca o valor no barramento de dados, correspondente ao endereço contido no barramento de endereços, a seguir chega no RDM da CPU ($RDM \leftarrow MP(Op)$) ;

Passo 4: O valor armazenado no RDM é transferido para o Acumulador $ACC \leftarrow RDM$ (ou $ACC \leftarrow MP(Op)$) ;

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

- g) (0,6) Descreva detalhadamente a execução da instrução **MUL Op.** nesta máquina. **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 ($REM \leftarrow Op$) e é disponibilizado no barramento de endereço ;

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

Passo 3: A memória coloca o valor da célula no barramento de dados chegando ao RDM da CPU ($RDM \leftarrow MP(Op)$) ;

Passo 4: A CPU executa a multiplicação do valor recebido com o contido no acumulador armazenando o resultado no RDM; $RDM \leftarrow ACC \times RDM$ (ou $RDM \leftarrow ACC \times MP(Op)$);

Passo 4.1 : o conteúdo do RDM é liberado para o barramento de dados ;

Passo 5: A CPU coloca no REM o valor do operando ($REM \leftarrow Op$) e é disponibilizado no barramento de endereços ;

Passo 6: A CPU aciona pelo barramento de controle o sinal de escrita em memória;

Passo 7: A memória armazena o conteúdo do barramento de dados na célula cujo endereço consta do barramento de endereços, $MP(Op) \leftarrow RDM$ (ou $MP(Op) \leftarrow ACC \times MP(Op)$) ;

Passo 8: CI é incrementado ($CI \leftarrow CI+1$) para apontar para a próxima instrução a ser lida.

2. (1,5) Considere uma máquina que utiliza 32 bits para representar números em ponto fixo e em ponto flutuante.

- a) (0,8) Mostre a representação de -311,375 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)

$$-311,375_{10} = -100110111,011_2 = -1,00110111011 \times 10^{+8}$$

Sinal = 1 (negativo)

Expoente = +8 = (por excesso de 127) = 8 + 127 = 135₁₀ = 1000 0111₂

Mantissa = ,00110111011

Resultado: 1 1000 0111 0011011101100000000000

- b) (0,7) Para o conjunto de bits obtido no item anterior, indique o que ele representa na base 10, considerando-se as seguintes representações: (**Não precisa fazer as contas, deixe-as indicadas**):

$$(11000011 \ 10011011 \ 10110000 \ 00000000)_2$$

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

$$2^{31} + 2^{30} + 2^{25} + 2^{24} + 2^{23} + 2^{20} + 2^{19} + 2^{17} + 2^{16} + 2^{15} + 2^{13} + 2^{12} = 3.281.760.256$$

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

$$-2^{31} + (2^{30} + 2^{25} + 2^{24} + 2^{23} + 2^{20} + 2^{19} + 2^{17} + 2^{16} + 2^{15} + 2^{13} + 2^{12}) = -1.013.207.040$$

3) (1,5) Considere um sistema de computação que possui uma memória principal (RAM) com capacidade máxima de endereçamento de 64 K 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

=> 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 8 bytes, como 1 célula = 1 byte, 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.

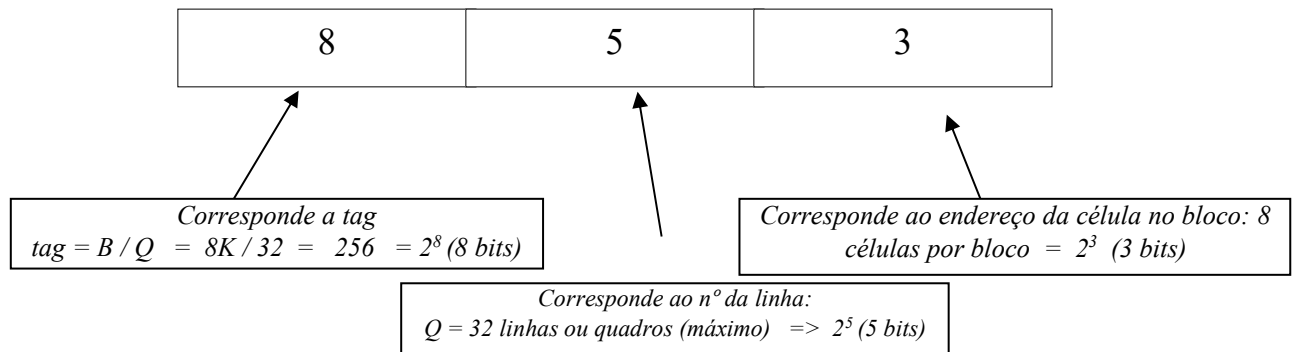
=> Tamanho da memória cache (em blocos ou linhas) => $Q = 32$ linhas

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

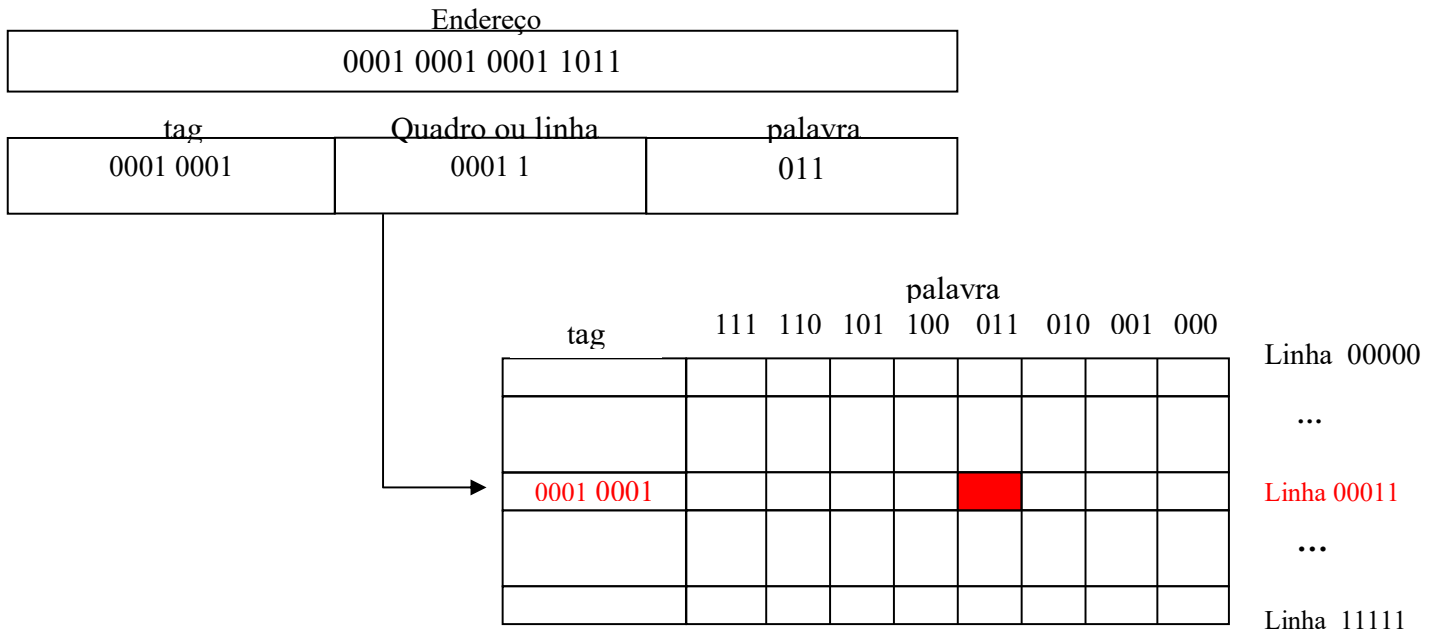
Memória principal

=> 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 capacidade da memória cache em bytes?

Capacidade da cache = $Q \times K = 32 \text{ linhas} \times 8 \text{ palavras/linha}$,
considerando nesta questão, 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

3) Responda as questões abaixo:

- a. (1,5) Analise os modos de endereçamento direto, indireto e imediato, estabelecendo diferenças de desempenho, vantagens e desvantagens de cada um.

Imediato: O campo operando contém o dado, desta forma o dado é transferido da memória juntamente com a instrução. Vantagem: Rapidez na execução da instrução, pois não requer acesso à memória principal, apenas na busca da própria instrução. Desvantagem: Limitação do tamanho do campo operando das instruções reduzindo o valor máximo do dado a ser manipulado. Trabalho excessivo para alteração de valores quando o programa é executado repetidamente e o conteúdo das variáveis serem diferentes em cada execução. Desempenho: Não requer acesso a memória principal. Mais rápido que o modo direto

Direto: O campo operando da instrução contém o endereço onde se localiza o dado. Vantagem: Flexibilidade no acesso a variáveis de valor diferente em cada execução do programa. Desvantagem: Limitação de memória a ser usada conforme o tamanho do operando. Desempenho: Requer apenas um acesso a memória principal. Mais rápido que o modo indireto

Indireto: O campo de operando contém o endereço de uma célula, sendo o valor contido nesta célula o endereço do dado desejado. Vantagem: Usado como "ponteiro". Elimina o problema do modo direto

de limitação do valor do endereço do dado. Manuseio de vetores (quando o modo indexado não está disponível). Desvantagem: Muitos acesso à MP para execução. Desempenho: inferior ao modo direto por exigir pelo menos 2 acessos à memória principal.

b. (1,0) Explique os sistemas NUMA e SMP.

Sistemas SMP (ou UMA) têm como característica o acesso a todas as partes da memória principal com tempo de acesso uniforme. Em sistemas NUMA, todos os processadores possuem também acesso a todas as partes da memória principal podendo diferir o tempo de acesso em relação às posições da memória e processador.

Nos sistemas SMP o aumento no número de processadores tem como consequência problemas de tráfego no barramento comum degradando o desempenho. Uma solução para isto é a utilização de clusters, que tem, usualmente, como consequência alterações significativas na aplicação (software). Nos sistemas NUMA podem-se ter vários nós multiprocessadores, cada qual com seu próprio barramento, resultando em pequenas alterações na aplicação (software).

c. (1,0) Explique Compilação e Interpretação.

A compilação consiste na análise de um programa escrito em linguagem de alto nível (programa fonte) e sua tradução em um programa em linguagem de máquina (programa objeto).

Na interpretação cada comando do código fonte é lido pelo interpretador, convertido em código executável e imediatamente executado antes do próximo comando.

A interpretação tem como vantagem sobre a compilação a capacidade de identificação e indicação de um erro no programa-fonte (incluindo erro da lógica do algoritmo) durante o processo de conversão do fonte para o executável.

A interpretação tem como desvantagem o consumo de memória devido ao fato de o interpretador permanecer na memória durante todo o processo de execução do programa. Na compilação o compilador somente é mantido na memória no processo de compilação e não utilizado durante a execução. Outra desvantagem da interpretação está na necessidade de tradução de partes que sejam executadas diversas vezes, como os loops que são traduzidos em cada passagem. No processo de compilação isto só ocorre uma única vez. Da mesma forma pode ocorrer para o programa inteiro, em caso de diversas execuções, ou seja, a cada execução uma nova interpretação.