

- 1) (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.*

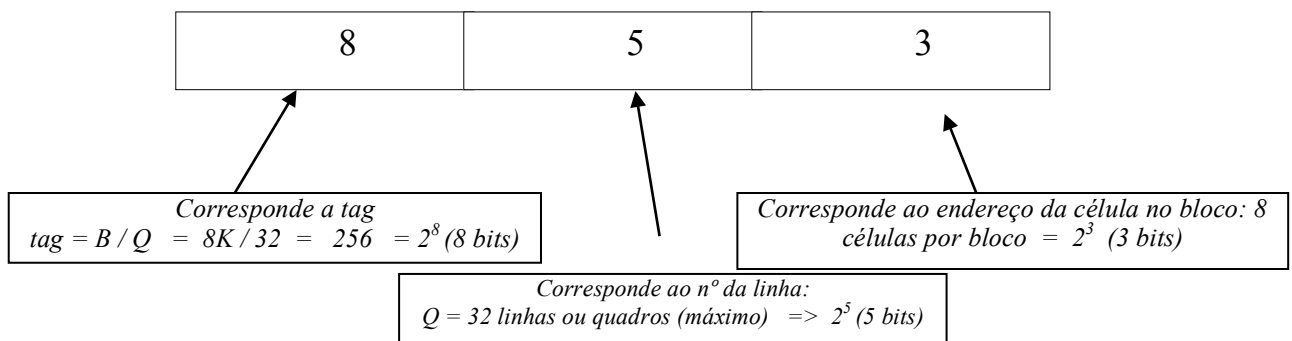
=> *Tamanho da memória cache (em blocos ou linhas) => $Q = 32$ linhas*

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

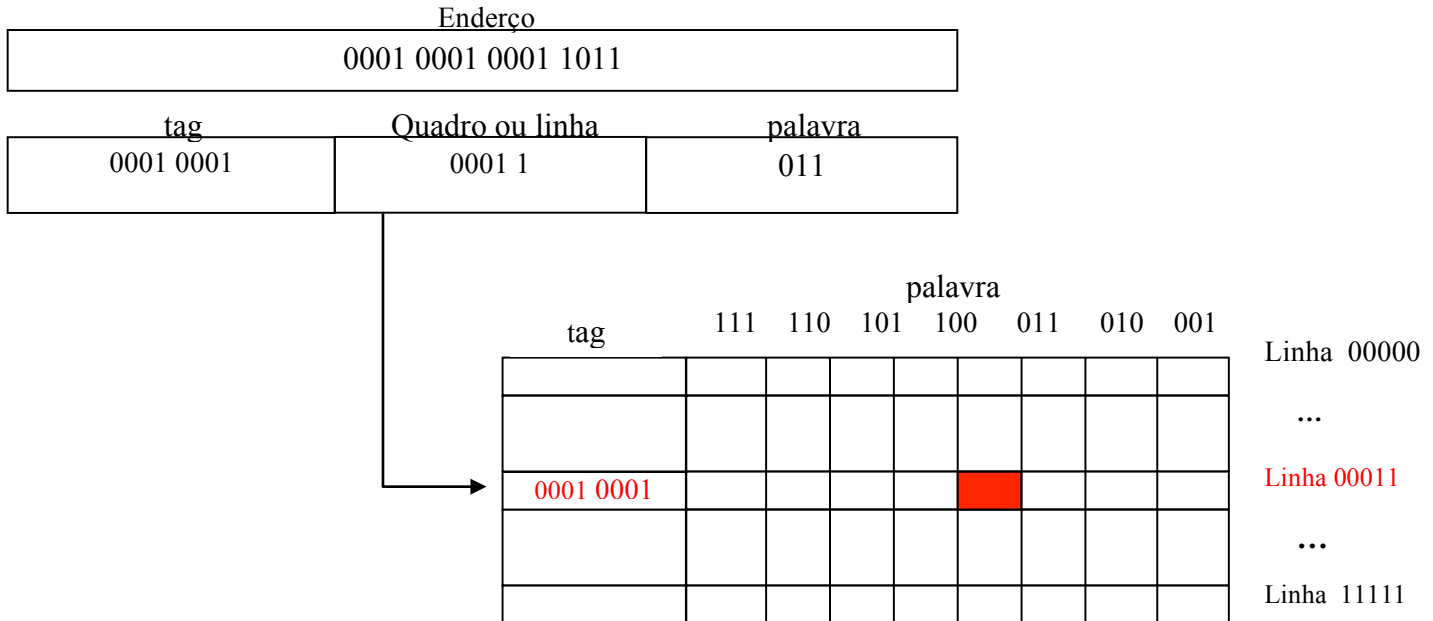
Memória principal

=> *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

2) (3,5) 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. Desempenho: Não requer acesso a memória principal. Mais rápido que o modo direto. 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.

Direto: O campo operando da instrução contém o endereço onde se localiza o dado. Desempenho: Requer apenas um acesso a memória principal. Mais rápido que o modo indireto.

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.

Indireto: O campo de operando contém o endereço de uma célula, sendo o valor contido nesta célula o endereço do dado desejado. Desempenho: Requer 2 acessos a memória principal. Vantagem: Usar 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, requer pelo menos 2 acessos à memória principal.

b. (1,0) Explique Arquiteturas RISC e CISC.

RISC: Reduced Instruction Set Computer – Computador com um conjunto reduzido de instruções

CISC - Complex Instruction Set Computer: Computador com um conjunto complexo de instruções

CISC: Principais características:

Possui microprogramação para aumento da quantidade de instruções incluindo novos modos de endereçamento, de forma a diminuir a complexidade dos compiladores e em consequência permitir linguagens de alto nível com comandos poderosos para facilitar a vida dos programadores. Em contrapartida, muitas instruções significam muitos bits em cada código de operação, instrução com maior comprimento e maior tempo de interpretação.

RISC: Principais características:

Menor quantidade de instruções e tamanho fixo. Não há microprogramação. Permite uma execução otimizada, mesmo considerando que uma menor quantidade de instruções vá conduzir a programas mais longos. Uma maior quantidade de registradores e suas utilizações para passagem de parâmetros e recuperação dos dados, permitindo uma execução mais otimizada de chamada de funções. Menor quantidade de modos de endereçamento com o objetivo de reduzir de ciclos de relógio para execução das instruções. Instruções de formatos simples e únicos tiram maior proveito de execução com pipeline cujos estágios consomem o mesmo tempo.

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.*

3) (2,5) 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 30 instruções.
- Cada instrução deve conter um código de operação e um operando como mostrado abaixo, onde o operando indica um endereço de memória

Cód. Oper.	Operando
------------	----------

- a. (0,2) Calcule o tamanho mínimo em bits do REM e do barramento de endereços.

Capacidade de endereçar 128Mcélulas => $N = 128M$ células
tamanho mínimo do REM será o tamanho do barramento de endereços necessário
para atender a capacidade de endereçamento de 128Mcélulas
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 30 instruções =>
cod.oper = 5 bits que permite até 32 instruções diferentes
operando = endereço de memória = 27 bits

Então, código de operação + oper. = $27 + 5 = 32$ bits
Tamanho mínimo da instrução é 32 bits

- c. (0,5) 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 no título da 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.

$CI = \text{tamanho mínimo necessário para endereçar toda a memória} = 27 \text{ bits}$

$RI = \text{tamanho necessário para uma instrução} = 32 \text{ bits}$

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

$\text{Total de bits} = T$

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

- 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.

- 4) (2,5) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 16 bits. Na representação para ponto flutuante, o número a ser representado deve ser expresso na notação científica normalizada ($\pm (1, b_1 b_2 b_3 \dots b_{10})_2 \times 2^{\text{expoente}}$). O bit mais à esquerda representa o sinal e deve ser 0 para números positivos e 1 para números negativos. O expoente está representado em excesso de 16. A representação é mostrada na figura abaixo.

S	expoente	b-1 b-2 b-3 ... b-10
1	5 bits	10 bits

- a. (1,0) Suponha que o conteúdo dos ^{bit} 16 bits seja (CB68)₁₆. Indique o valor em decimal (pode deixar as contas indicadas) para este conjunto de bits quando considerarmos que ele está representando:

$$(CB68)_{16} = 1100101101101000_2$$

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

$$2^{15} + 2^{14} + 2^{11} + 2^9 + 2^8 + 2^6 + 2^5 + 2^3 = 52.072$$

- ii) (0,3) um inteiro em complemento a 2 **19304**

$$-2^{15} + (2^{14} + 2^{11} + 2^9 + 2^8 + 2^6 + 2^5 + 2^3) = -13.464$$

iii) (0,6) um número em ponto flutuante utilizando a representação do enunciado

Sendo:

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

$$\text{Expoente} = 10010_2 = +2^4 + 2^1 = +18 - 16 = +2$$

$$\text{Mantissa} = 1101101000$$

$$\begin{aligned} \text{Temos então} &\Rightarrow -1,1101101000 \times 2^{+2} = -111,01101_2 \\ &= -(+2^{+2} + 2^{+1} + 2^0 + 2^{-2} + 2^{-3} + 2^{-5}) \\ &= -(4 + 2 + 1 + 0,25 + 0,125 + 0,03125) = -7,40625 \end{aligned}$$

- b. (0,6) Qual o menor e o maior valor **positivo** normalizado na representação em ponto flutuante, descrita no enunciado, para este computador? Os valores devem ser representados **em decimal**.

Maior positivo

$$0111101111111111 = 1,1111111111 \times 2^{+14} = +32.752$$

Menor positivo

$$0000010000000000 = 1,0000000000 \times 2^{-15} = 0,0000305$$

$$11111111110000$$

- c. (0,7) Mostre a representação em ponto flutuante do valor em decimal -87,75, na representação do enunciado para ponto flutuante.

Convertendo para binário = $+1010111,11_2 \Rightarrow$

colocando na notação científica $+1,0101111 \times 2^{+6}$

Temos então:

$$\text{Sinal} = 1 \text{ (negativo)}$$

$$\text{Expoente} = +6_{10} \Rightarrow \text{(excesso de 16)} \quad 10110_2$$

$$\text{Mantissa} = ,0101111$$

$$\text{Resultado: } 110110101111 \text{ ou } 1101100101111$$