



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Organização de Computadores

AP1 1º semestre de 2009 - GABARITO

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. (1,5) Considerando a arquitetura apresentada em sala de aula, explique para que serve o barramento de controle, o barramento de dados e o barramento de endereços.

O barramento de controle é utilizado para transferir sinais de controle entre a Unidade Central de Processamento (UCP) e a Memória Principal (MP). Por exemplo, a UCP pode enviar um sinal de controle para a MP através deste barramento para indicar que deseja realizar uma operação de leitura com a MP. Um outro exemplo seria o envio pela MP de um sinal de controle para a UCP para indicar para a UCP que a UCP deve esperar para realizar alguma operação com a MP.

O barramento de dados é utilizado para transferir informações (instruções ou dados) entre a UCP e a MP.

O barramento de endereços é utilizado para transferir bits que representam um determinado endereço da UCP para a MP.

2. (2,5) Suponha que você tenha que projetar uma máquina com as seguintes especificações:

- Capaz de endereçar 256 M células de memória principal, sendo que cada célula armazena 1 byte.
- Deve possuir os registradores RDM (utilizado para enviar e receber dados para/de uma célula de memória), REM (utilizado para enviar o endereço de uma célula de memória), CI (utilizado para indicar o endereço da célula inicial da instrução a ser lida da memória) e RI (utilizado para armazenar uma instrução).

- Cada instrução tem o tamanho de 32 bits e deve conter um código de operação e um operando como mostrado abaixo:

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

onde Operando é um endereço de uma célula da memória principal.

- Em cada acesso à memória, a Unidade Central de Processamento obtém o conteúdo de 1 célula de memória.

- a) (0,2) Indique qual deve ser a capacidade mínima de armazenamento em bits do REM.

REM = E = tamanho em bits necessários para endereçar uma célula de memória em um total de 256 células
Número total de células $N = 2^E = 256$ M células = 2^{28} células $\Rightarrow E = 28 \Rightarrow$
REM = 28 bits

- b) (0,2) Indique qual deve ser o tamanho mínimo em bits do barramento de endereços.

Barramento de endereços = REM = 28 bits

- c) (0,2) Indique a capacidade mínima de armazenamento em bits do RDM.

Como em cada acesso à memória, a UCP obtém o conteúdo de uma célula de memória que contém 1 byte que são 8 bits, o RDM deve ter uma capacidade mínima de 8 bits.

- d) (0,2) Indique o tamanho em bits do barramento de dados.

Como em cada acesso à memória, a UCP obtém o conteúdo de uma célula de memória que contém 1 byte que são 8 bits, o barramento de dados deve ter uma capacidade mínima de 8 bits.

- e) (0,2) Calcule a quantidade máxima de bits que este sistema pode armazenar na memória.

A memória possui 256 M células e cada célula armazena 8 bits, logo quantidade máxima de bits que pode ser armazenada neste sistema é $2^{28} \times 8$ bits = 2^{31} bits = 2Gbits

- f) (0,5) Calcule a quantidade máxima possível de códigos de operação diferentes.

Tamanho da instrução = 32 bits
Tamanho da instrução = código de operação + operando (endereço de memória)
Código de operação = Tamanho da instrução – operando = 32bits – 28 bits = 4bits
Quantidade máxima de cod.oper. = $2^4 = 16$ códigos de operações diferentes

- g) (0,4) Calcule o número de acessos à memória que devem ser realizados para se obter uma instrução.

Em cada acesso à memória se obtém o conteúdo de uma célula que possui 8 bits. Uma instrução tem o tamanho de 32 bits e portanto ocupa 4 células. Logo serão necessários 4 acessos à memória para se obter uma instrução.

- h) (0,6) Calcule a capacidade de armazenamento em bits dos registradores RI e CI.

RI terá que ter no mínimo o tamanho de uma instrução \Rightarrow RI = 32bits
CI terá que ter tamanho que permita endereçar toda a memória \Rightarrow CI = 28bits

3. (2,0) Considere uma máquina que pode ter seu ciclo de busca e execução de uma instrução dividido em 5 estágios totalmente independentes: Busca de Instrução (BI), Decodificação (DI), Cálculo de Endereços de Operandos (CO), Busca dos Operandos (BO) e Execução (EX). Cada um dos estágios BI, BO e EX possui a duração de 3 ns e cada estágio DI e CO tem duração de 1 ns. Cada instrução desta máquina precisa executar os 5 estágios que serão sempre executados na sequência BI, DI, CO, BO e EX.

- a) (0,5) Uma implementação desta máquina foi realizada de modo que cada instrução deve ser completamente realizada em um único ciclo de relógio e uma instrução só começa a ser realizada após o término da anterior. Calcule a duração do ciclo de relógio que esta implementação deve possuir. Lembre-se que todas as instruções necessitam dos 5 estágios.

<i>BI</i> <i>3ns</i>	<i>DI</i> <i>1ns</i>	<i>CO</i> <i>1ns</i>	<i>BO</i> <i>3ns</i>	<i>EX</i> <i>3ns</i>
-------------------------	-------------------------	-------------------------	-------------------------	-------------------------

$$\begin{array}{c}
 \text{+-----+} \quad 11\text{ns} \quad \text{+-----+} \\
 \text{Ciclo de relógio para execução de uma instrução (sem pipeline)} = \\
 3\text{ns} + 1\text{ns} + 1\text{ns} + 3\text{ns} + 3\text{ns} = \mathbf{11\text{ns}}
 \end{array}$$

- b) (0,5) Como cada estágio é independente um do outro, implementou-se uma **nova** arquitetura utilizando-se um pipeline de 5 estágios. Calcule a duração do ciclo de relógio que a implementação pipeline deve ter. Considere que qualquer estágio do pipeline deve poder ser realizado em um único ciclo de relógio.

<i>1º. estágio</i>	<i>2º. estágio</i>	<i>3º. estágio</i>	<i>4º. estágio</i>	<i>5º. estágio</i>
<i>BI</i> <i>3ns</i>	<i>DI</i> <i>1ns</i>	<i>CO</i> <i>1ns</i>	<i>BO</i> <i>3ns</i>	<i>EX</i> <i>3ns</i>

$$\text{Ciclo de relógio será igual ao tempo para execução do} \\
 \text{estágio de maior tempo de execução} = \mathbf{3\text{ns}}.$$

- c) (1,0) Mostre o tempo em que um programa que contenha 10 instruções será executado pela **implementação do item a e do item b**. Considere que estas 10 instruções possam ser executadas em fluxo constante.

$$\begin{array}{l}
 \text{Seja } Tex = \text{tempo de execução de uma instrução} \\
 = \text{número de estágios} \times \text{ciclo de relógio (determinado nos itens anteriores)}
 \end{array}$$

Item a (sem pipeline) :

$$Tex = 11 \text{ ns}$$

$$T_{total} = 10 \text{ instruções} \times Tex = \mathbf{110 \text{ ns}}$$

Item b (pipeline: 5 estágios)

$$Tex (\text{primeira instrução}) = 5 \text{ estágios} \times 3\text{ns} = 15\text{ns}$$

$$T_{total} = Tex + 9 \times (\text{tempo do ciclo do relógio})$$

$$T_{total} = 15\text{ns} + 9 \times 3\text{ns} = \mathbf{42 \text{ ns}}$$

4. (2.0) Considere um microprocessador hipotético com um tamanho de palavra e barramento de dados de 40 bits e 256 instruções. As instruções são todas do tamanho de uma célula de memória, de mesmo tamanho da palavra e são compostas por dois campos: o primeiro contém o código de operação e o restante contém o endereço do operando. Ele contém somente uma placa de memória com 4 Mbytes.

- a) Qual o tamanho, em bits, do RDM e do REM ?

Barramento de dados = 40 bits

*RDM = barramento de dados = **40 bits***

REM = barramento de endereços = bits necessários para acessar toda memória

Bits necessários para endereçar toda memória = operando

Uma instrução = tamanho da palavra = 40 bits

uma instrução = código de operação + operando

40 bits = 8 bits (necessários para 256 instruções diferentes) + operando

operando = 32bits

*portanto, REM = barramento de endereços = **32 bits***

- b) Seria possível aumentar a memória desta máquina ? Se possível, até quanto (em **bytes**)?

$N = 2^E$, sendo E o tamanho do barramento de endereços =>

$N = 2^{32} \Rightarrow N = 4\text{Gcélulas} \Rightarrow$

Como cada célula = tamanho da palavra = 40 bits = 5 bytes/célula =>

Total desta memória será 4Gcélulas x 5bytes/célula = 20 Gbytes

- c) Se quisesse aumentar o número de instruções para 512, mantendo inalterado o tamanho do barramento de dados e da palavra, quais seriam os tamanhos do RDM e REM ?

*RDM = barramento de dados = **40 bits***

REM = barramento de endereços = bits necessários para acessar toda memória

Bits necessários para endereçar toda memória = operando

Uma instrução = tamanho da palavra = 40 bits

uma instrução = código de operação + operando

40 bits = 9 bits (necessários para 512 códigos diferentes) + operando

operando = 31bits

*portanto, REM = barramento de endereços = **31 bits***

5. (2,0) Considere uma máquina que possa endereçar 2 Gbytes de memória física, sendo que cada endereço referencia uma célula de 1 byte. Ela possui uma memória cache que pode armazenar 2K blocos, sendo um bloco por linha e cada bloco possui 4 K células. Mostre o formato da memória cache, indicando os campos necessários (tag, bloco) e o número de bits para cada campo, o formato de um endereço da memória principal, indicando os bits que referenciam os campos da cache, e a capacidade em bits que a memória cache deve possuir (pode deixar a conta indicada) para os seguintes mapeamentos:
- Mapeamento direto.
 - Mapeamento totalmente associativo.
 - Mapeamento associativo por conjunto, onde cada conjunto possui duas linhas, cada uma de um bloco.

a) Mapeamento direto.

Memória Principal

- ⇒ Tamanho da memória (em bytes) = 2 Gbytes, como 1 célula referencia a 1 byte, temos $N = 2G$ células
 ⇒ Será organizada em blocos de 4Kcélulas, como 1 célula = 1 byte, temos cada bloco = 4Kbytes, $K = 4K$ células
 ⇒ Sendo N o tamanho endereçável da memória e K que é a quantidade de células por blocos temos:
 $N = 2G$ células e $K = 4K$ células / bloco
 Total de blocos: $B = N / K \Rightarrow B = 2G \text{ células} / 4K \text{ células/bloco} \Rightarrow B = 512K \text{ blocos}$

Memória Cache

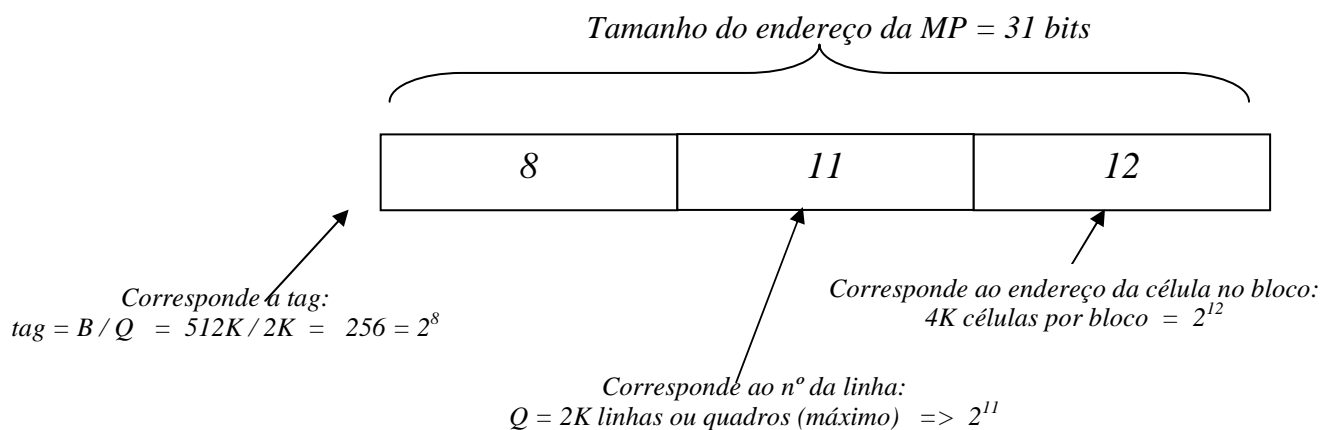
OBS: O K (quantidade de células/bloco) tem de ser igual a MP.

- ⇒ Tamanho da memória cache (em blocos ou linhas) $\Rightarrow Q = 2K$ blocos
 ⇒ Tamanho da memória cachê em células = $Q \times K = 2K \text{ blocos} \times 4K \text{ células/bloco} = 8M \text{ células}$

Endereçamento

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

sendo $N = 2^E \Rightarrow N = 2G \text{ células} \Rightarrow N = 2^{31} \Rightarrow E = 31 \text{ bits}$



b) Mapeamento totalmente associativo.

Memória Principal

=> $N = 2G$ células

=> $K = 4K$

=> $B = 512K$ blocos

Memória Cache

OBS: O K (quantidade de células/bloco) tem de ser igual a MP.

=> $Q = 2K$ blocos

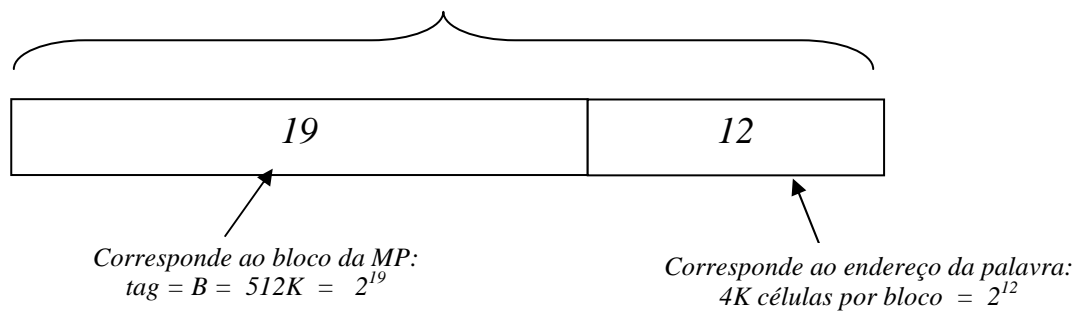
=> Tamanho da memória cache = $8M$ células

Endereçamento

Para endereçarmos toda a MP precisamos da seguinte quantidade de bits: $E = 31$ bits

Como o bloco pode ser alocado em qualquer posição da memória cache a tag indicará qual dos blocos da MP está alocado naquela posição da memória cache:

Tamanho do endereço da MP = 31 bits



c) Mapeamento associativo por conjunto, onde cada conjunto possui duas linhas, cada uma de um bloco.

Memória Principal

=> $N = 2G$ células

=> $K = 4K$ células / bloco

=> $B = 512 K$ blocos

Memória Cache

OBS: O K (quantidade de células/bloco) tem de ser igual a MP.

=> $Q = 2K$ blocos

=> Tamanho da memória cache = $8M$ células

=> 1 conjunto = 2 linhas (ou quadros) => Total de conjuntos = $C = 2K \text{ blocos} / 2 \Rightarrow C = 1K \text{ conjuntos}$

Endereçamento

Para endereçarmos toda a MP precisamos da seguinte quantidade de bits: $E = 31$ bits

Tamanho do endereço da MP = 31 bits

