



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Organização de Computadores

AP1 2º semestre de 2015.

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.
 6. Respostas sem desenvolvimento ou justificativas não serão consideradas
-

1. (2,0) Suponha que você deve projetar uma máquina com as seguintes especificações:

- Capaz de endereçar 64 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 o barramento de dados), REM (utilizado para enviar endereços no barramento de endereços), CI (utilizado para indicar o endereço da instrução a ser lida da memória) e RI (utilizado para armazenar uma instrução).
- Cada instrução deve conter um código de operação, um operando e um registrador como mostrado abaixo:

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

onde Operando é um endereço da memória principal e Reg. é o identificador de um Registrador, sendo que a máquina possui 6 registradores.

- Deve poder ter um máximo de 8 códigos de operação diferentes.

a) (0,3) Indique qual deve ser o tamanho mínimo em bits do REM

Memória com 64Mcélulas $\Rightarrow N = 64M$ células

tamanho mínimo do REM será o tamanho do barramento de endereços necessário para endereçar toda a memória.

Barramento de endereços (BE) = $\log_2 N = \log_2 64M = 26$ bits

REM = tamanho do BE = 26 bits

b) (0,3) Indique qual deve ser o tamanho mínimo em bits do do barramento de endereços.

REM = tamanho do BE = 26 bits

c) (0,6) Calcule o número de células que uma instrução necessita para ser armazenada.

Cada instrução = código de operação + 1 operando + 1 registrador

1o. operando = endereço de uma célula = 26bits

2o. operando = endereço de um registrador = 3 bits (total de $2^3 = 8$ registradores que atende ao que é preciso)

cod.operação = tamanho necessário para 8 códigos diferentes = 3 bits

tamanho da instrução = $3 + 26 + 3 = 32$ bits

cada célula 1 byte $\rightarrow 32$ bits/8 bits = 4 células

d) (0,6) Indique o **tamanho do RDM e do barramento de dados** de modo que a Unidade Central de Processamento obtenha uma instrução da memória principal realizando somente um acesso à memória principal.

RDM = barramento de dados = tamanho necessário para transferir uma instrução

RDM = barramento de dados = 32 bits.

e) (0,6) Calcule a capacidade de armazenamento em bits dos registradores RI e CI, utilizando-se os valores calculados nos itens anteriores.

CI = tamanho necessário para endereçar toda a memória = 26 bits

RI = tamanho necessário para uma instrução = 32 bits

2. (2,0) Explique como funcionam os barramentos síncronos e assíncronos.

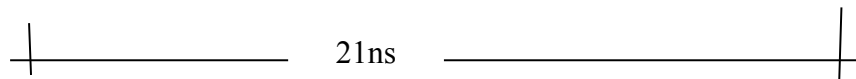
Nos barramentos que possuem operação síncrona, a ocorrência e duração de todos os eventos que acontecem nas diversas linhas do barramento são guiados por pulsos de um relógio. Existe uma linha no barramento por onde circulam os pulsos gerados pelo relógio e todos os acontecimentos nas linhas de barramento, como, por exemplo, envio de endereço e envio de sinal de leitura, tem sua inicialização e duração de ocorrência determinadas por estes pulsos.

Nos barramentos que operam de forma assíncrona, não existe um relógio sincronizador. Os eventos ocorrem no barramento de acordo com um protocolo de aperto de mão (handshaking). Cada evento no barramento não depende dos pulsos do relógio, mas sim de algum evento que deve ocorrer anteriormente a ele e que pode ter qualquer duração de tempo. .

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 (BI), Decodificação (DE), Execução (EX), Leitura de Dados (LD) e Escrita de Resultado (ER). Cada um dos estágios BI e ER possui a duração de 6 ns e cada estágio DE, EX e LD tem duração de 3 ns. Cada instrução desta máquina precisa executar os 5 estágios.

- 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>6ns</i>	<i>DE</i> <i>3ns</i>	<i>EX</i> <i>3ns</i>	<i>LD</i> <i>3ns</i>	<i>ER</i> <i>6ns</i>
-------------------------	-------------------------	-------------------------	-------------------------	-------------------------



Ciclo de relógio para execução de uma instrução (sem pipeline) =
 $6ns + 3ns + 3ns + 3ns + 6ns = 21ns$

- 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.

1º. estágio 2º. estágio 3º. estágio 4º. estágio 5º. estágio

<i>BI</i> <i>6ns</i>	<i>DE</i> <i>3ns</i>	<i>EX</i> <i>3ns</i>	<i>LD</i> <i>3ns</i>	<i>ER</i> <i>6ns</i>
-------------------------	-------------------------	-------------------------	-------------------------	-------------------------

Ciclo de relógio será igual ao tempo para execução do
estágio de maior tempo de execução = 6ns.

- c) (1,0) Mostre o tempo em que um programa que contenha 100 instruções será executado pela **implementação do item a e do item b**. Considere que estas 100 instruções podem ser executadas em fluxo constante.

Seja Tex = tempo de execução de uma instrução
= número de estágios x ciclo de relógio (determinado nos itens anteriores)

Item a (sem pipeline) :

$$Tex = 1 \text{ estágio} \times 21ns \text{ para execução de instrução} = 21ns$$

$$Ttotal = 100 \text{ instruções} \times Tex = \underline{2100ns}$$

Item b (pipeline: 5 estágios)

$$Tex \text{ (primeira instrução)} = 5 \text{ estágios} \times 6ns = 30ns$$

$$Ttotal = Tex + 99 \times (\text{tempo do ciclo do relógio})$$

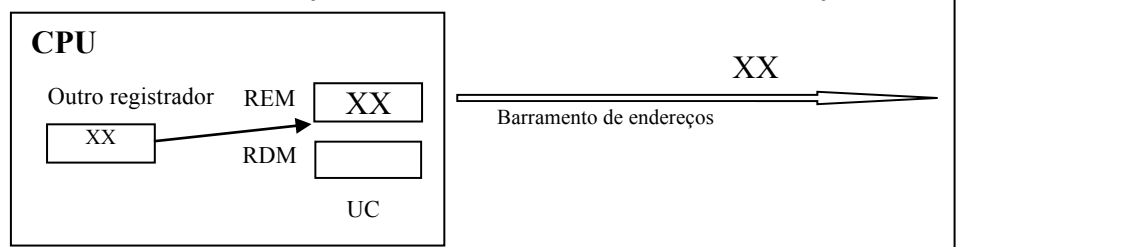
$$Ttotal = 30ns + 99 \times 6ns = \underline{624ns}$$

4. (2,0) Descreva e esquematize graficamente passo a passo as operações de leitura da memória e de escrita na memória, indicando como os registradores RDM e REM são utilizados e como a unidade de controle gera os sinais necessários.

Processo de Escrita em memória

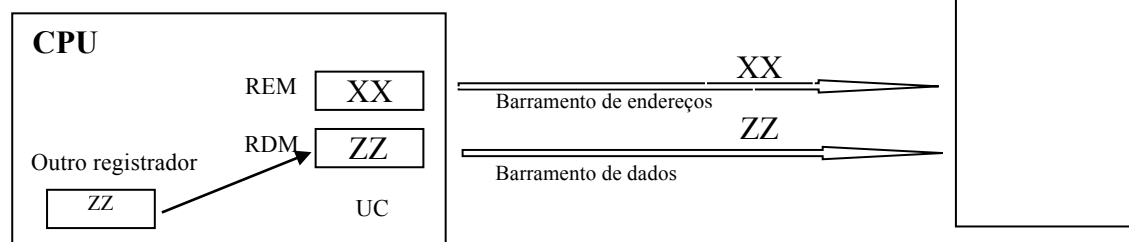
1º Passo) $(REM) \leftarrow (\text{outro registrador})$

O endereço é colocado no barramento de endereços

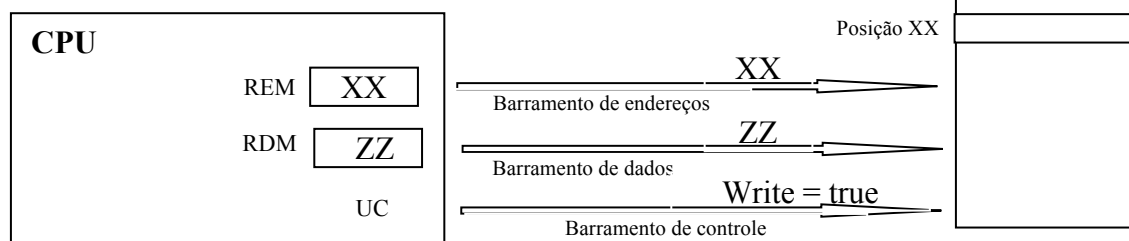


2º.Passo) $(RDM) \leftarrow (\text{outro registrador})$

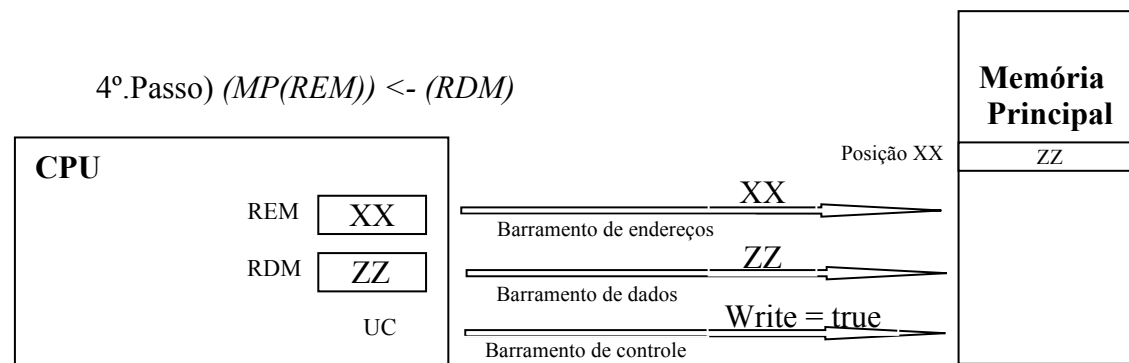
O dado é colocado no barramento de dados



3º. Passo) *Sinal de escrita (Write) é ativado no barramento de controle*



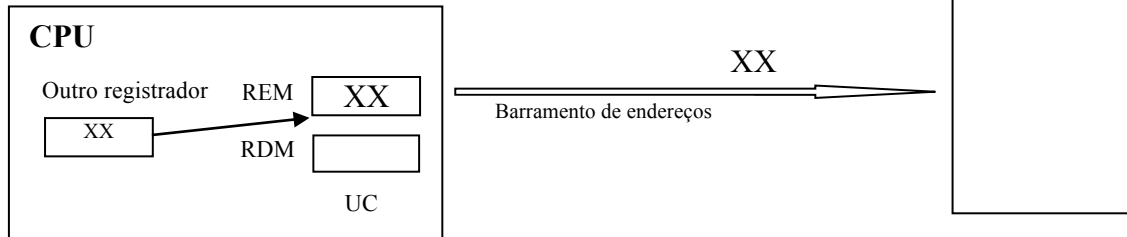
4º.Passo) $(MP(REM)) \leftarrow (RDM)$



Processo de Leitura em memória

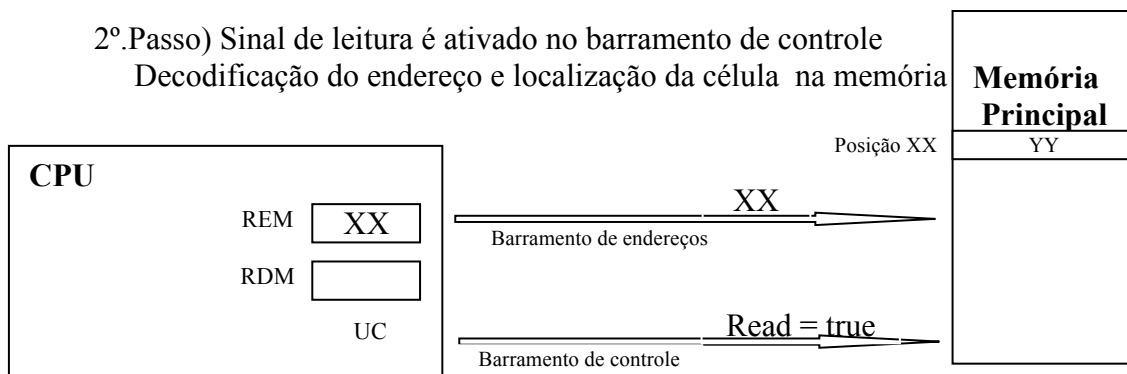
1º Passo) $(REM) \leftarrow (\text{outro registrador})$

O endereço é colocado no barramento de endereços

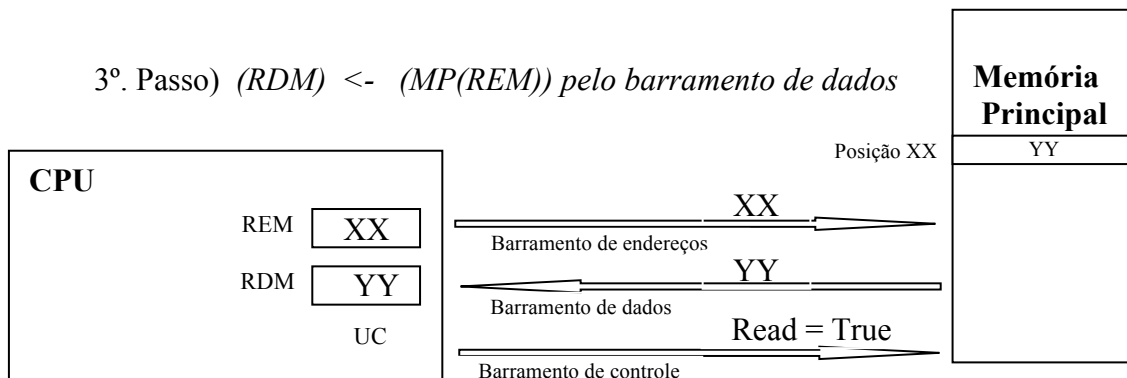


2º Passo) Sinal de leitura é ativado no barramento de controle

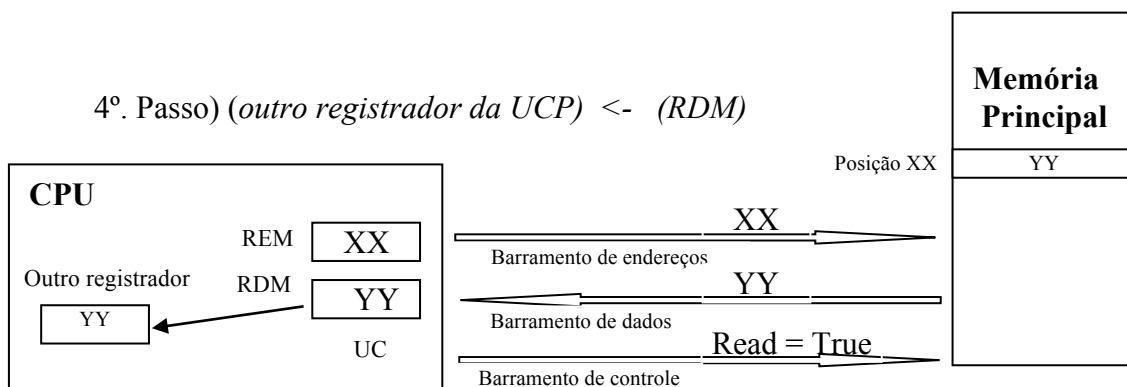
Decodificação do endereço e localização da célula na memória



3º. Passo) $(RDM) \leftarrow (MP(REM))$ pelo barramento de dados



4º. Passo) $(\text{outro registrador da UCP}) \leftarrow (RDM)$



-
- Diagrama de um endereço de 31 bits dividido em três campos:
- 4 bits:** *Corresponde a tag:*
 $tag = B / Q = 4M / 256K = 16 = 2^4$
 $tag = 4 \text{ bits}$
 - 18 bits:** *Corresponde ao n° da linha:*
 $Q = 256K \text{ linhas ou quadros (máximo)} \Rightarrow 2^{18}$
 $linha = 18 \text{ bits}$
 - 9 bits:** *Corresponde ao endereço da célula no bloco:*
 $512 \text{ células por bloco} = 2^9$
 $palavra = 9 \text{ bits}$

b) Mapeamento totalmente associativo.

Memória Principal

=> $N = 2G$ células

=> $K = 512$ células/bloco

=> $B = 4M$ blocos

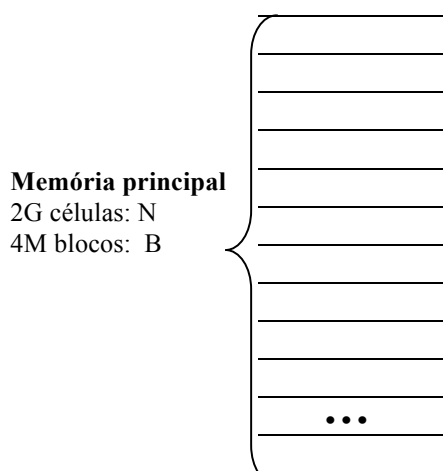
Memória Cache

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

=> $Q = 256K$ blocos

=> Tamanho da memória cache = 256Kbytes

Organização da cache

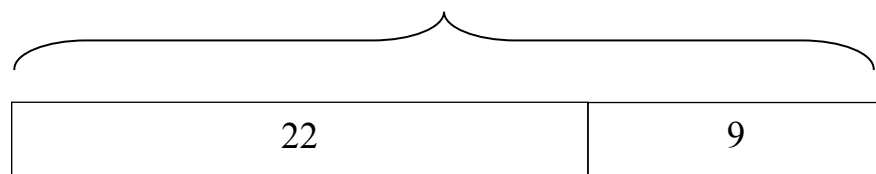


linha	válido	tag	Conteúdo (bloco)
0	1 bit	24 bits	512 células de 8 bits cada = 4Kbits
1			
2			
3			
4			
5			
.....			
Q - 2			
Q - 1			

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



Corresponde ao bloco da MP:
tag = $B = 4M = 2^{22}$
tag = 22 bits

Corresponde ao endereço da palavra:
512 células por bloco = 2^9
palavra = 9 bits