



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Organização de Computadores**

**GABARITO - AP1 2º semestre de 2009.**

1) (2,0) Um computador possui uma capacidade máxima de memória principal com 128 K células, cada uma capaz de armazenar uma palavra de 8 bits.

a) Qual é o maior endereço em decimal desta memória?

$$\text{Maior endereço (em decimal)} = N - 1 = 128K - 1 = 2^{17} - 1 = \mathbf{131.071}$$

b) Qual é o tamanho do barramento de endereços deste sistema?

$$\text{Barramento de endereço} = E$$

$$E = \text{tamanho em bits necessários para acessar toda a memória (N)}$$

$$N = 2^E = 128K \text{ células} = 2^{17} \text{ células} \Rightarrow E = 17 \Rightarrow$$

$$\mathbf{\text{Barramento de endereço} = 17 \text{ bits}}$$

c) Quantos bits podem ser armazenados no RDM e no REM?

$$\text{REM} = E = \text{tamanho em bits necessários para acessar toda a memória (N)} \Rightarrow$$

$$\mathbf{\text{REM} = 17 \text{ bits}}$$

$$\text{RDM} = \text{tamanho da palavra} \Rightarrow \mathbf{\text{RDM} = 8 \text{ bits}}$$

d) Qual é o número máximo de bits que pode existir na memória?

$$T = M \times N \Rightarrow T = 8\text{bits/célula} \times 128K\text{células} \Rightarrow \mathbf{T = 1024Kbits}$$

2. (2,0) Considere uma máquina que possa endereçar 64 Mbytes 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 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:

a) Mapeamento direto.

#### **Memória Principal**

$\Rightarrow$  Tamanho da memória (em bytes) = 64Mbytes, como 1 célula referencia a 1 byte, temos  $N = 64M$  células

$\Rightarrow$  Será organizada em blocos de 4 bytes, como 1 célula = 1 byte, temos cada bloco = 4 células,  $K = 4$

$\Rightarrow$  Sendo  $N$  o tamanho endereçável da memória e  $K$  que é a quantidade de células por blocos temos:

$$N = 64M \text{ células e } K = 4 \text{ células / blocos o total de blocos da MP ( B ) será:}$$

$$\text{Total de blocos: } B = N / K \Rightarrow B = 64M \text{ células} / 4 \text{ células/bloco} \Rightarrow$$

$$B = 16 M \text{ blocos}$$

### **Memória Cache**

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

$$\Rightarrow \text{Tamanho da memória cache (em blocos ou linhas)} \Rightarrow Q = 2K \text{ blocos}$$

$$\Rightarrow \text{Tamanho da memória cache em células} = Q \times K = 2K \text{ blocos} \times 4 \text{ células/bloco} = 8K \text{ células}$$

**Endereço da MP:** Para endereçarmos toda a MP precisamos da seguinte quantidade de bits ( $E$ )

$$\text{sendo } N = 2^E \Rightarrow N = 64M \text{ células} \Rightarrow N = 2^{26} \Rightarrow E = 26 \text{ bits}$$

Composição do endereço em função da memória cache

$$\Rightarrow \text{tag} = B / Q = 64 M / 8K = 8K = 2^{13} \Rightarrow \text{tag} = 13 \text{ bits}$$

$$\Rightarrow \text{nº da linha: } Q = 2K \text{ linhas ou quadros (máximo)} \Rightarrow 2^{11} \Rightarrow 11 \text{ bits}$$

$$\Rightarrow \text{células por bloco: } 4 \text{ células por bloco} = 2^2 \Rightarrow 2 \text{ bits}$$

26bits		
Tag = 13bits	No. Linha = 11bits	Célula no bloco=2bits

b) Mapeamento totalmente associativo.

### **Memória Principal**

$$\Rightarrow N = 64M \text{ células}$$

$$\Rightarrow K = 4$$

$$\Rightarrow B = 16 M \text{ blocos}$$

### **Memória Cache**

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

$$\Rightarrow Q = 2K \text{ blocos}$$

$$\Rightarrow \text{Tamanho da memória cache} = 8K \text{ células}$$

$$\text{Endereço da MP} = 26 \text{ bits}$$

Composição do endereço em função da memória cache

$$\Rightarrow \text{tag} = B = 16M = 2^{24} \Rightarrow \text{tag} = 24 \text{ bits}$$

$$\Rightarrow \text{células por bloco: } 4 \text{ células por bloco} = 2^2 \Rightarrow 2 \text{ bits}$$

26bits	
Tag = 24bits	Célula no bloco=2bits

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

### **Memória Principal**

$$\Rightarrow N = 64M \text{ células}$$

$$\Rightarrow K = 4$$

=>  $B = 16 \text{ M blocos}$

### Memória Cache

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

=>  $Q = 2K \text{ blocos}$

=> Tamanho da memória cache =  $8K \text{ células}$

=> 1 conjunto = 4 linhas (ou quadros) =>

Total de conjuntos =>  $C = 2K \text{ blocos} / 2 \Rightarrow C = 1K \text{ conjuntos}$

Endereço da MP = 26 bits

Composição do endereço em função da memória cache

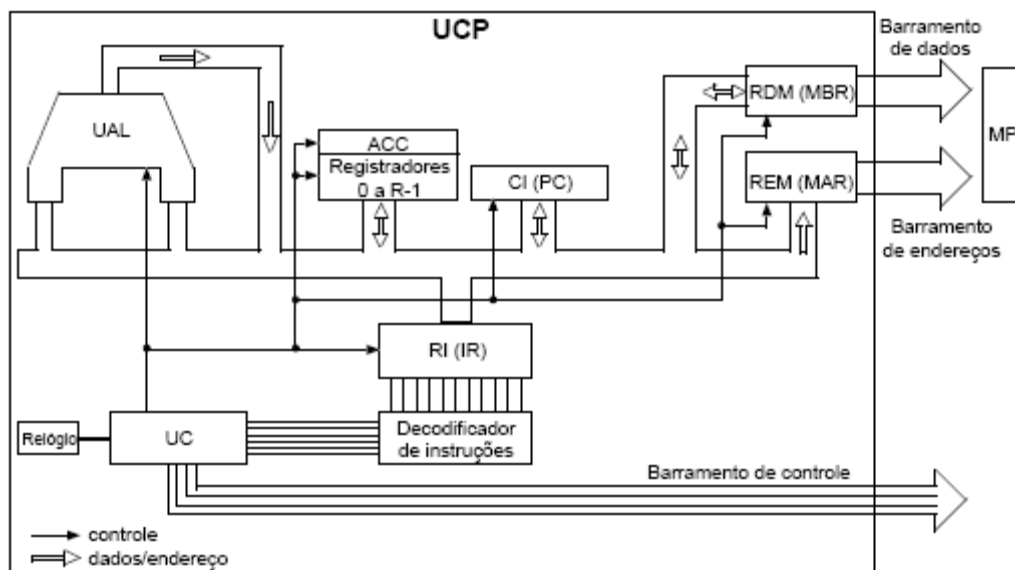
=>  $tag = B / C = 16 \text{ M} / 1K = 16K = 2^{14} \Rightarrow tag = 14\text{bits}$

=> n° do Conjunto:  $Q = 1K \text{ linhas ou quadros (máximo)} \Rightarrow 2^{10} \Rightarrow 10\text{bits}$

=> células por bloco:  $4 \text{ células por bloco} = 2^2 \Rightarrow 2\text{bits}$

26bits		
Tag = 14bits	No. Conjunto = 10bits	Célula no bloco = 2bits

3. (2,5) Considere o sistema apresentado em aula mostrado na figura abaixo.



(Fin. 6.3 do livro texto)

Descreva **detalhadamente** a execução do trecho de código abaixo, indicando como o Acumulador (ACC), RDM, REM, Unidade Aritmética Lógica (UAL) e Barramento de controle, de dados e de endereços são utilizados na execução de cada instrução.

LDA 10  
ADD 20  
STR 30

Lembre-se que (1) na execução da instrução LDA Op. o conteúdo da memória cujo endereço é Op. deve ser armazenado no acumulador, (2) na execução de ADD Op. O conteúdo do acumulador deve ser somado com o conteúdo da

memória cujo endereço é  $Op.$ , e (3) na execução de  $STR\ Op.$  o conteúdo do acumulador deve ser armazenado na memória no endereço  $Op.$

LDA 10

a)  $RI \leftarrow (CI)$

b)  $CI \leftarrow CI + 1$

c) Decodificação do código de operação

d) Busca do operando na memória

- A UC emite sinais para que o valor do campo operando = 10 seja transferido para o REM
- A UC ativa a linha READ do barramento de controle
- Conteúdo de memória no endereço 10 é transferido para RDM
- $ACC \leftarrow RDM$

ADD 20

a)  $RI \leftarrow (CI)$

b)  $CI \leftarrow CI + 1$

c) Decodificação do código de operação

d) Busca do operando na memória

- A UC emite sinais para que o valor do campo operando = 20 seja transferido para o REM
- A UC ativa a linha READ do barramento de controle
- Conteúdo de memória no endereço 20 é transferido para RDM

e) Execução da operação

- Conteúdo do RDM e do ACC são transferidos para a UAL para serem somados
- Resultado da soma é armazenado em ACC

STR 30

a)  $RI \leftarrow (CI)$

b)  $CI \leftarrow CI + 1$

c) Decodificação do código de operação

d) Escrita do operando na memória

- A UC emite sinais para que o valor do campo operando = 30 seja transferido para o REM
- A UC emite sinais para que o valor do registrador acumulador (ACC) seja transferido para o RDM

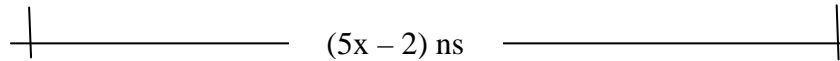
RDM

- A UC ativa a linha WRITE do barramento de controle
- Conteúdo do RDM é transferido, através do barramento de dados, para o endereço 30 da memória, endereço este transferido do REM para a memória através do barramento de endereços.

4. (1,5) 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  $X$  ns e cada estágio DI e CO tem duração de  $(X-1)$  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. Sabe-se que para executar um programa com 200 instruções, utilizando-se esta implementação, foram necessários 4400 ns. Calcule a **duração do ciclo de relógio** que esta implementação possui. Lembre-se que todas as instruções necessitam dos 5 estágios.

<i>BI</i> <i>x ns</i>	<i>DI</i> <i>(x-1) ns</i>	<i>CO</i> <i>(x-1) ns</i>	<i>BO</i> <i>x ns</i>	<i>EX</i> <i>x ns</i>
--------------------------	------------------------------	------------------------------	--------------------------	--------------------------



*Ciclo de relógio para execução de uma instrução (sem pipeline) =*  
 $x\text{ ns} + (x-1)\text{ ns} + (x-1)\text{ ns} + x\text{ ns} + x\text{ ns} = (5x-2)\text{ ns}$   
*(tempo para execução de uma instrução)*

*Para execução de 200 instruções, sendo cada uma executada em um ciclo de relógio, serão necessários 4400ns, então:*

***200 instruções = 4400 ns => Cada instrução executada em 22ns, portanto, o ciclo de relógio = 22ns***

*Se fossemos determinar o tempo de cada estágio:*

$$(5x-2) = 22\text{ ns} \Rightarrow x = 4$$

$$BI, BO, EX = 4\text{ ns}$$

$$DI, CO = (4-1) = 3\text{ ns}$$

b) (1,0) Como cada estágio é independente um do outro, deseja-se implementar uma **nova** arquitetura utilizando-se um pipeline de 5 estágios. Nesta nova implementação **cada estágio do pipeline** deve ser executado em um ciclo de relógio. Calcule a duração máxima do ciclo de relógio que esta implementação pipeline deve possuir de modo que o tempo de execução do mesmo programa executado no item anterior (executa 200 instruções) seja menor do que o tempo calculado no item anterior. Sabe-se que as 200 instruções são executadas em fluxo constante, ou seja, não ocorrem conflitos.

<i>1º. estágio</i>	<i>2º. estágio</i>	<i>3º. estágio</i>	<i>4º. estágio</i>	<i>5º. estágio</i>
<i>BI</i> <i>x ns</i>	<i>DI</i> <i>(x-1) ns</i>	<i>CO</i> <i>(x-1) ns</i>	<i>ER</i> <i>x ns</i>	<i>ER</i> <i>x ns</i>

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

*Para execução de 200 instruções, com execução em pipeline. :*

*Tempo para execução da primeira instrução (T1) = 5 . x ns*

*Tempo para execução das 199 restantes = 199 . tempo de 1 estágio (x) = 199ns*

*Tempo para as 200 instruções = 5x ns + 199x ns = 204x ns*

*Tempo para as 200 instruções < 4400ns*

$$204x\text{ ns} < 4400\text{ ns} \Rightarrow x < 21,57\text{ ns, concluindo,}$$

*O ciclo de um relógio deverá corresponder ao estágio de maior duração (x)*

*E o ciclo de relógio deverá ter no máximo a duração de 21,57ns*

5. (2,0) Explique **detalhadamente** como funciona uma unidade de controle microprogramada.

*A unidade de controle microprogramada gera os sinais de controle para a UCP através da “execução” de um microprograma que consiste de uma seqüência de instruções de uma linguagem de microprogramação. Estas instruções são simples e especificam microoperações. Uma unidade de controle microprogramada é implementada com circuitos lógicos e é capaz de executar uma seqüência de microinstruções que geram sinais de controle para o funcionamento da UCP. Estes sinais de controle são utilizados para causar transferências de dados entre registradores e memória e execução de operações pela ULA, por exemplo.*

*A unidade de controle microprogramada pode utilizar microinstruções horizontais que tem como característica gerar sinais de controle distintos para a UCP diretamente de cada bit que as compõem, ou microinstruções verticais que se caracterizam por possuírem menos bits que os sinais necessários para o controle da UCP, os quais são gerados através da decodificação dos bits das microinstruções verticais.*