



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 2010.

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. (2,0) Um computador possui um RDM com 32 bits de tamanho e um REM com capacidade para armazenar números com 20 bits. Sabe-se que a célula deste computador armazena dados com 16 bits de tamanho e que ele possui uma quantidade N de células, igual à sua capacidade máxima de armazenamento. Pergunta-se:

- a) Qual é o tamanho do barramento de endereços?

Barramento de endereços = REM = 20bits

- b) Quantas células de memória são lidas em uma única operação de leitura?

Célula = 16 bits

RDM = barramento de dados = 32 bits

Portanto, são lidas 2 células de memória durante uma única operação de leitura

- c) Quantos bits têm a memória principal?

$N = \text{quantidade de células} = \text{capacidade máxima} = 2^{20} = 1.048.576 \text{ células}$

Total de bits da memória = $T = N \times M$

Como $M = \text{quantidade de bits em uma célula}$, então $M = 16 \text{ bits}$

$T = N \times M = 2^{20} \times 2^4 = 2^{24} = 16.777.216 \text{ bits}$

2. (2,0) Considere uma máquina que possa endereçar 4 Gbytes de memória física, utilizando endereço referenciando byte, e que tenha a sua memória organizada em blocos de 4K bytes. Ela possui uma memória cache que pode armazenar 8K blocos, sendo um bloco por

linha. Mostre o formato da memória cache, indicando os campos necessários (válido, tag, bloco) e o número de bits para cada campo, e o formato de um endereço da memória principal, indicando os bits que referenciam os campos da cache, para os seguintes mapeamentos:

a) Mapeamento direto.

Memória Principal

- ⇒ Tamanho da memória (em bytes) = 4Gbytes, como 1 célula referencia a 1 byte, temos $N = 4G$ células
- ⇒ Será organizada em blocos de 4K bytes, como 1 célula = 1 byte, temos cada bloco = 4K células, $K = 4K$
- ⇒ Sendo N o tamanho endereçável da memória e K que é a quantidade de células por blocos temos:
 $N = 4G$ células e $K = 4K$ células / blocos o total de blocos da MP (B) será:
 Total de blocos: $B = N / K \Rightarrow B = 4G \text{ células} / 4K \text{ células/bloco} \Rightarrow$
 $B = 1M$ 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 $Q = 8K$ blocos
- ⇒ Tamanho da memória cachê em células = $Q \times K =$
 $8K \text{ blocos} \times 4K \text{ células/bloco} = 32M \text{ células}$

Cada célula possui 1 byte = 8bits, então,
como a cache possui 32M células x 8bits, que totaliza 256M bits

Endereço da MP: Para endereçarmos toda a MP precisamos da seguinte quantidade de bits (E)
 sendo $N = 2^E \Rightarrow N = 4G \text{ células} \Rightarrow N = 2^{32} \Rightarrow E = 32 \text{ bits}$

Composição do endereço em função da memória cache

- $\Rightarrow \text{tag} = B / Q = 1M / 8K = 128 = 2^7 = 7 \text{ bits}$
- $\Rightarrow \text{n}^\circ \text{ da linha: } Q = 8K \text{ linhas ou quadros (máximo)} \Rightarrow 2^{13} \Rightarrow 13 \text{ bits}$
- $\Rightarrow \text{células por bloco: } 4K \text{ células por bloco} = 2^{12} \Rightarrow 12 \text{ bits}$

32bits		
Tag = 7 bits	No. Linha =13 bits	Célula no bloco=12 bits

b) Mapeamento totalmente associativo.

Memória Principal

- $\Rightarrow N = 4G \text{ células}$
- $\Rightarrow K = 4K \text{ células/bloco}$
- $\Rightarrow B = 1M \text{ blocos}$

Memória Cache

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

- $\Rightarrow Q = 8K \text{ blocos}$
- $\Rightarrow \text{Tamanho da memória cache} = 32M \text{ células ou } 32M \text{ bytes ou } 256M \text{ bits}$

Endereço da MP = 32 bits

Composição do endereço em função da memória cache

- $\Rightarrow \text{tag} = B = 1M = 2^{20} \Rightarrow \text{tag} = 20 \text{ bits}$
- $\Rightarrow \text{células por bloco: } 4K \text{ células por bloco} = 2^{12} \Rightarrow 12 \text{ bits}$

32bits	
Tag = 20bits	Célula no bloco=12bits

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

Memória Principal

=> $N = 4G$ células

=> $K = 4K$ células/bloco

=> $B = 1M$ blocos

Memória Cache

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

=> $Q = 8K$ blocos

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

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

Total de conjuntos => $C = 8K \text{ blocos} / 4 \Rightarrow C = 2K \text{ conjuntos}$

Endereço da MP = 32 bits

Composição do endereço em função da memória cache

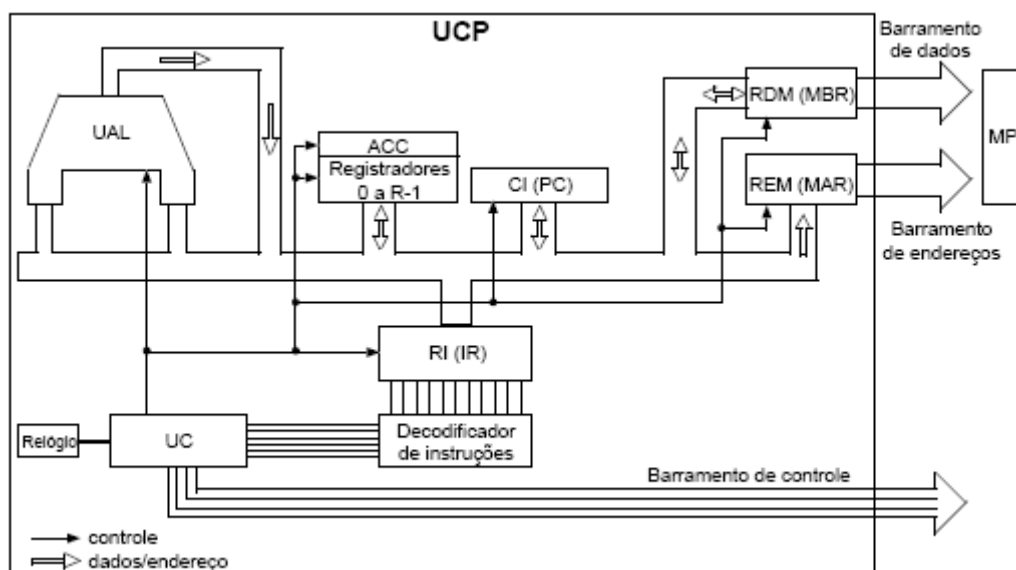
=> $tag = B / C = 1M / 2K = 512 = 2^9 \Rightarrow tag = 9 \text{ bits}$

=> n° do Conjunto: $Q = 2K \text{ linhas ou quadros (máximo)} \Rightarrow 2^{11} \Rightarrow 11 \text{ bits}$

=> células por bloco: $4K \text{ células por bloco} = 2^{12} \Rightarrow 12 \text{ bits}$

32 bits		
Tag = 9 bits	No. Conjunto = 11bits	Célula no bloco=12bits

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



Descreva detalhadamente a execução das instruções LDA Op. e JZ Op., indicando como o Registrador de Instrução (RI), Contador de Instrução (CI), Acumulador (ACC), Registrador de Dados da Memória (RDM), Registrador de Endereços da Memória (REM), Unidade Aritmética Lógica (UAL) e Barramento de controle, de dados e de endereços são utilizados na execução destas instruções. Lembre-se que a instrução LDA Op., quando executada, carrega o conteúdo da memória cujo endereço é Op. no Acumulador e a instrução JZ Op., quando executada, carrega CI com o valor de Op. se o conteúdo do Acumulador é zero, e caso contrário carrega CI com CI+1. .

LDA Op

- $RI \leftarrow (CI)$
- $CI \leftarrow CI + 1$
- Decodificação do código de operação
- Busca do operando na memória
 - A Unidade de Controle (UC) emite sinais para que o valor do campo operando = Op seja transferido para o REM
 - O conteúdo do REM é disponibilizado no barramento de endereços
 - A UC ativa a linha READ do barramento de controle
 - O conteúdo da memória é disponibilizado no barramento de dados
 - Conteúdo de memória (Op) é transferido para o RDM
 - $ACC \leftarrow RDM$

JZ Op

- $RI \leftarrow (CI)$
- $CI \leftarrow CI + 1$
- Decodificação do código de operação
- UC emite sinal para transferir conteúdo do acumulador para UAL
 - $UAL \leftarrow ACC$
- Executa operação de comparação
 - Se Resultado = verdadeiro, isto é, $ACC = 0$
 $CI \leftarrow Op.$

4. (2,0) Considere uma máquina que pode ter seu ciclo de busca e execução de uma instrução dividido em 4 estágios totalmente independentes: Busca de Instrução (BI), Decodificação (DI), Cálculo de Endereços de Operandos (CO) e Execução (EX). Cada um dos estágios BI e EX possui a duração de 4 ns e cada estágio DI e CO tem duração de 2 ns. Cada instrução desta máquina precisa executar os 4 estágios que serão sempre executados na sequência BI, DI, CO e EX.

- a) (0,2) Uma implementação desta máquina foi realizada de modo que cada instrução deve ser completamente realizada em um único ciclo de relógio. Calcule a duração do ciclo de relógio que esta implementação deve possuir. Lembre-se que todas as instruções necessitam dos 4 estágios.

<i>BI</i> 4ns	<i>DI</i> 2ns	<i>CO</i> 2ns	<i>EX</i> 4ns
------------------	------------------	------------------	------------------

—————|————— 12ns —————|—————

Ciclo de relógio para execução de uma instrução (sem pipeline) =
 $4ns + 2ns + 2ns + 4ns = 12ns$

- b) (0,8) Como cada estágio é independente um do outro, deseja-se implementar uma nova arquitetura utilizando-se um pipeline de 4 estágios. Nesta nova implementação cada estágio do pipeline deve ser executado em um ciclo de relógio. Calcule a duração do ciclo de relógio que esta implementação pipeline deve possuir.



Ciclo de relógio será igual ao tempo para execução do estágio com maior tempo de execução = **4ns**.

- c) (1,0) Considere um programa que necessita executar 1000 instruções. Calcule o tempo de execução deste programa na máquina do item a e na máquina do item b.

Seja T_{ex} = tempo de execução de uma instrução = número de estágios \times ciclo de relógio (determinado nos itens anteriores)

Para o item a (sem pipeline) :

$$T_{ex} = 1 \text{ estágio} \times 12ns = 12ns$$

$$T_{total} = 1000 \text{ instruções} \times T_{ex} = \underline{\underline{12.000ns}}$$

Para o item b (pipeline: 4 estágios) :

$$T_{ex} = 4 \text{ estágios} \times 4ns = 16ns$$

$$T_{total} = T_{ex} + 999 \times \text{tempo de 1 estágio}$$

$$T_{total} = 16ns + 999 \times 4ns = \underline{\underline{4012ns}}$$

5. (1,5) Explique como funciona uma Unidade Central de Processamento (UCP) cujo controle é realizado por hardware e uma UCP com controle por microprograma.

A unidade de controle implementada por hardware consiste em circuitos combinatórios. Os sinais lógicos de entrada na unidade são transformados em um conjunto lógico de sinais que controlam a execução da instrução. Para implementar a unidade, necessita-se derivar, para cada sinal de controle a ser gerado para que cada instrução seja executada de forma correta, uma expressão booleana que define esse sinal em função dos sinais de entrada referentes à instrução. As expressões booleanas são implementadas com circuitos combinatórios.

A unidade de controle microprogramada é projetada de modo a executar as microinstruções que compõem uma instrução. Ela é composta por: Memória de controle, Contador de microprograma e Sequenciador.

A Memória de controle armazena as microinstruções que compõem uma instrução e o Contador de microprograma armazena a localização da próxima microinstrução a ser executada. O Sequenciador é o componente que controla a sequência de execução das microinstruções, informando o local da próxima microinstrução que deve ser executada e armazenada no Contador de microprograma.