



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Organização de Computadores

Gabarito - AP1 1º semestre de 2012.

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 uma capacidade máxima de memória principal com 64K células, cada uma capaz de armazenar uma palavra de 8 bits.

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

Resposta:

$$N = 64K \text{ células} \Rightarrow N = 2^{16}$$

$$\text{Maior endereço} = N - 1 = 2^{16} - 1 = \mathbf{65.535}$$

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

Resposta:

O Tamanho deste barramento será o suficiente para endereçar todas as células da memória (N).

O tamanho do barramento corresponderá ao valor de e em $2^e = N$

*$2^e = N \Rightarrow 2^e = 2^{16} \Rightarrow e = 16$, portanto, **barramento de endereços = 16 bits***

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

Resposta:

*Tamanho do REM = tamanho do barramento de endereço. **REM = 16 bits***

*Tamanho do RDM = tamanho do barramento de dados e terá de ser no mínimo o tamanho de uma célula. **RDM = 8 bits***

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

Resposta:

O total de bits da memória será igual a T

Tamanho da memória (T) = N x M = 64K células x 8bits/célula = $2^{16} \times 2^3 \text{ bits} = 2^{19} \text{ bits} = 524.288 \text{ bits}$

2. (2,0) Considere uma máquina que possa endereçar 256 Mbytes de memória física, utilizando endereço referenciando byte, e que tenha a sua memória organizada em blocos de 16 bytes. Ela possui uma memória cache que pode armazenar 4K blocos, sendo um bloco por linha. Mostre o formato da memória cache, indicando os campos necessários (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.

Resposta:

Memória Principal

=> Tamanho da memória (em bytes) = 256Mbytes, como 1 célula referencia 1 byte, temos então, N = 256M células

=> Será organizada em blocos de 16 bytes, cada bloco = 16 células, K = 16

*=> Sendo N o tamanho endereçável da memória, e K a quantidade de células por bloco, N = 256M células e K = 16 células / blocos, o total de blocos da MP (B) será:
B = N / K => B = 256M células / 16 células/bloco => B = 16 M blocos*

Memória Cache

OBS: K (quantidade de células/bloco) na cache tem de ser igual na MP.

=> Tamanho da memória cache (em blocos ou linhas) => Q = 4K blocos

=> Tamanho da memória cache em células = Q x K =

4K blocos x 16 células/bloco = 64K células

=> Cada célula possui 1 byte, então, cache possui 4K bytes

Endereço da MP: Para endereçarmos toda a MP precisamos da seguinte quantidade de bits (E) sendo N = 2^E => N = 256M células => N = 2^{28} => E = 28 bits

Composição do endereço em função da memória cache

=> tag = B / Q = 16M / 4K = 4K = 2^{12} = 12 bits

=> n° da linha: Q = 4K linhas ou quadros (máximo) => 2^{12} => 12bits

=> células por bloco: 16 células por bloco = 2^4 => 4 bits

28bits		
Tag 12 bits	No. Linha 12 bits	Célula no bloco 4 bits

b) Mapeamento totalmente associativo.

Resposta:

Memória Principal

$\Rightarrow N = 256M \text{ células}$

$\Rightarrow K = 16 \text{ células/bloco}$

$\Rightarrow B = 16Mblocos$

Memória Cache

OBS: K (quantidade de células/bloco) na cache tem de ser igual na MP.

$\Rightarrow Q = 4K \text{ blocos}$

$\Rightarrow \text{Tamanho da memória cache} = 16K \text{ células}$

Endereço da MP = 28 bits

Composição do endereço em função da memória cache

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

$\Rightarrow \text{células por bloco: } 16 \text{ células por bloco} = 2^4 \Rightarrow 4 \text{ bits}$

28bits	
Tag 24bits	Célula no bloco 4 bits

3. (2,0) Considere a arquitetura de máquina estudada em sala de aula. Descreva **detalhadamente** a execução das instruções **STR Op.** e **JMP 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 STR Op., quando executada, armazena o conteúdo da memória cujo endereço é Op. no acumulador e a instrução JMP Op., quando executada, carrega CI com o valor de Op..

Resposta:

STR Op

(Duas respostas serão consideradas: (1) considerando a definição da instrução STR como está na vídeo-aula e livro-texto ou (2) como está no enunciado).

1) Considerando a definição do livro-texto (o conteúdo do acumulador é armazenado na memória na posição Op.)

a) $RI \leftarrow (CI)$

b) $CI \leftarrow CI + 1$

c) Decodificação do código de operação

d) A UC emite sinais para que o valor do campo operando = Op seja transferido para o REM

e) Conteúdo do Acumulador (ACC) é transferido para o RDM ($RDM \leftarrow ACC$)

f) A UC ativa a linha WRITE do barramento de controle

g) O conteúdo do REM é transferido para o barramento de endereços

h) O conteúdo do RDM é transferido para o barramento de dados

i) A memória grava o dado recebido no endereço que consta do barramento de endereços

2) Considerando a definição do enunciado (o conteúdo da memória na posição *Op*. é armazenado no acumulador)

- a) $RI \leftarrow (CI)$
- b) $CI \leftarrow CI + 1$
- c) Decodificação do código de operação
- d) A UC emite sinais para que o valor do campo operando = *Op* seja transferido para o REM
- e) A UC ativa a linha READ do barramento de controle
- f) O conteúdo do REM é transferido para o barramento de endereços
- g) O conteúdo de memória cujo endereço consta no barramento de endereços é transferido para o RDM através do barramento de dados
- h) Conteúdo do RDM é transferido para o acumulador (ACC) ($ACC \leftarrow RDM$)

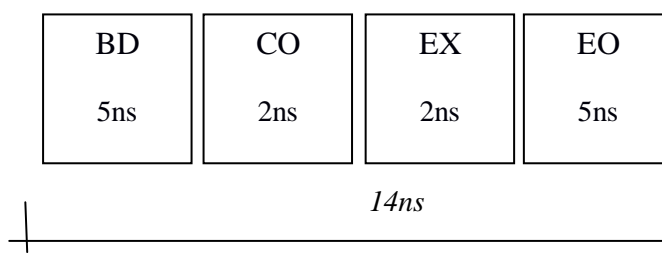
JMP *Op*

- a) $RI \leftarrow (CI)$
- b) Decodificação do código de operação
- c) UC emite sinal para transferir conteúdo do operando para CI
 $CI \leftarrow \text{Operando} \quad (CI \leftarrow Op)$

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 e Decodificação de Instrução (BD), Cálculo de Endereços de Operandos (CO), Execução (EX) e Escrita de Operandos (EO). Cada um dos estágios BD e EO possui a duração de 5 ns e cada estágio CO e EX, 2 ns. Cada instrução desta máquina precisa executar os 4 estágios que serão sempre executados na sequência BD, CO, EX e EO.

- a) (0,2) Uma implementação desta máquina foi realizada de modo que cada instrução é completamente realizada em exatamente um único ciclo de relógio e cada instrução só começa a ser executada quando a execução da instrução anterior é finalizada. Calcule a duração do ciclo de relógio que esta implementação possui.

Resposta:



Ciclo de relógio para execução de uma instrução =
 $5ns + 2ns + 2ns + 5ns = 14ns$

- b) (0,6) Um programa P foi executado na implementação do item a e a duração de execução foi de 90.000 ns. Calcule o número de instruções executadas.

Resposta:

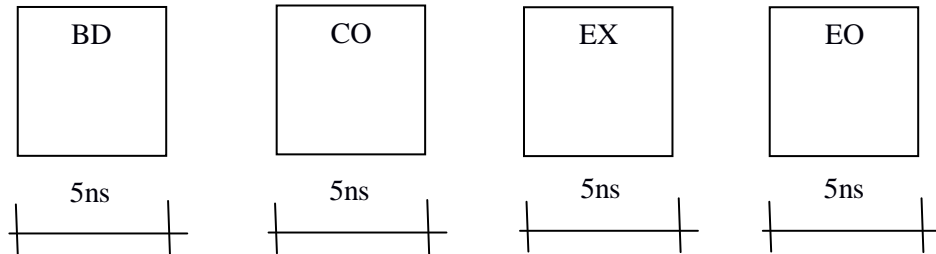
Como cada instrução leva 14ns para ser executada.

Em 90.000 ns teremos $90.000 \text{ ns} / 14 \text{ ns} = 6428,6$ instruções, ou melhor,

6428 instruções completas executadas

- c) (0,2) Como cada estágio é independente um do outro, implementou-se uma nova arquitetura utilizando-se um pipeline de 4 estágios. Nesta nova implementação cada estágio do pipeline é executado em exatamente um ciclo de relógio. Calcule a duração do ciclo de relógio que esta implementação pipeline possui. Lembre-se que todas as instruções necessitam dos 4 estágios.

Resposta:



Ciclo de relógio será igual ao tempo para execução do estágio com maior tempo de execução = 5ns.

- d) (0,6) Com o número de instruções calculado no item b, calcule o tempo de execução do programa P na máquina com pipeline do item c.

Resposta:

Seja T_{ex} = tempo de execução de uma instrução = número de estágios \times ciclo de relógio

Para o item c (pipeline: 4 estágios) :

$$T_{ex} = 4 \text{ estágios} \times 5ns = 20ns$$

$$T_{total} = T_{ex} + 6427 \times \text{tempo de 1 estágio}$$

$$T_{total} = 20ns + 6427 \times 5ns = \underline{\underline{32.155ns}}$$

- e) (0,4) Caso as duas máquinas custem o mesmo preço, indique qual das duas máquinas você compraria para executar o programa P. Explique sua decisão.

Resposta:

Certamente seria a máquina apontada no item c (com uso do pipeline) pois executaria o programa P em menor tempo

5. (2,0) Explique as duas maneiras utilizadas para projetar uma Unidade de Controle: (1) controle programado diretamente no hardware (hardwired), (2) controle por microprogramação.

Resposta:

O projeto da unidade de controle implementada diretamente no hardware consiste no projeto de circuitos combinatórios para transformar os sinais lógicos de entrada da unidade 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.