

Disciplina: Organização de Computadores

GABARITO DA AP1 2018-02

1. (2,0) Quais são os mapeamentos existentes para armazenamento e recuperação de dados (instruções) na memória cache? Explique em detalhes cada um deles.

*Ao todo são três os tipos mapeamento: dois básicos, mapeamento direto e o mapeamento totalmente associativo, e um terceiro, o mapeamento associativo por conjunto, que consiste em uma combinação dos 2 primeiros. A principal diferença entre os 2 modos básicos é que no mapeamento direto, um bloco da MP só poderá ser alocado em uma mesma linha da cache. Já no mapeamento totalmente associativo, um bloco da MP poderá ser alocado em qualquer linha da cache. Cada linha da cache possui um identificador (**tag**) que informa qual bloco da MP está ali alocado. No mapeamento direto, a verificação do **tag** é feita apenas em uma linha da cache, já no mapeamento totalmente associativo, onde o bloco desejado pode ocupar qualquer linha da cache, há, então, a necessidade de busca pela linha que contenha a tag procurada. No mapeamento associativo por conjunto as linhas da cache são agrupadas em conjuntos e o bloco da MP poderá ser alocado em qualquer uma das linhas do conjunto. O conjunto reduz a espaço de busca que é feito no mapeamento totalmente associativo e fornece uma maior flexibilização de alocação do bloco na cache quando comparado com o mapeamento direto.*

*No mapeamento direto, o endereço fornecido pelo processador é dividido em 3 campos: **tag**, **bloco**, **palavra**. O campo **palavra** corresponde a posição do dado procurado dentro da linha da cache. O campo **bloco** aponta para qual linha da cache poderá estar o bloco procurado. E o campo **tag** a identificação de qual bloco está naquela linha. A verificação se palavra procurada está na linha apontada pelo campo **bloco**, é feita comparando a **tag** da linha da cache com o campo **tag** do endereço fornecido pelo processador.*

*No mapeamento associativo, o endereço fornecido pelo processador é dividido em 2 campos: **tag** e **palavra**. O campo **palavra** tem igual função do mapeamento anterior. E o campo **tag** indica qual o bloco da MP. Para verificar se um determinado bloco está na cache, será necessário comparar o campo **tag** do endereço com a tag de cada linha da cache.*

*No mapeamento associativo por conjunto, o endereço fornecido pelo processador, similar ao mapeamento direto, é dividido em 3 campos: **tag**, **conjunto** e **palavra**. O campo **palavra**, com a mesma função dos anteriores, indica a posição dentro da linha. O campo **conjunto** aponta em qual conjunto será feita a busca pelo bloco, onde a **tag** de cada linha que pertence ao conjunto será comparada ao campo **tag** do endereço fornecido pelo processador. características e tecnologia dessa cache, não haveria a necessidade de termos uma cache.*

2. (2,0) Explique a hierarquia de memória dos computadores atuais, detalhando cada nível e suas respectivas características.

Podemos ilustrar essa hierarquia de memória na forma de uma pirâmide dividida em 4 níveis.

No topo da pirâmide teríamos os registradores, que são pequenas unidades de memória que armazenam dados na UCP. São dispositivos de maior velocidade com tempo de acesso em torno de 1 ciclo de memória, menor capacidade de armazenamento além de armazenar as informações por muito pouco tempo.

Em um nível abaixo teríamos a memória cache, cuja função é acelerar a velocidade de transferência das informações entre UCP e MP e, com isso, aumentar o desempenho do sistema. A UCP procura informações primeiro na Cache. Caso não as encontre, as mesmas são transferidas da MP para a Cache. A cache possui tempo de acesso menor que a da Memória principal, porém com capacidade inferior a esta, mas superior ao

dos registradores e o suficiente para armazenar uma apreciável quantidade de informações, sendo o tempo de permanência do dado menor do que o tempo de duração do programa a que pertence.

Abaixo da memória cache teríamos a memória básica de um sistema de computação, que é a memória principal. Dispositivo onde o programa (e seus dados) que vai ser executado é armazenado para que a UCP busque instrução por instrução para executá-las. A MP são mais lentas que a cache e mais rápidas que a memória secundária, possui capacidade bem superior ao da cache e os dados ou instruções permanecem na MP enquanto durar a execução do programa.

Finalmente, na base da pirâmide teríamos a memória secundária, memória auxiliar ou memória de massa, que fornece garantia de armazenamento mais permanente aos dados e programas do usuário. Alguns dispositivos são diretamente ligados: disco rígido, outros são conectados quando necessário: disquetes, fitas de armazenamento, CD-ROM. São os mais lentos em comparação com os outros níveis de memória, mas possuem a maior capacidade de armazenamento e armazenam os dados de forma permanente.

3. (1,5) 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. .

4. (2,5) Suponha que você deve projetar uma máquina com as seguintes especificações: Capaz de endereçar 512 k células de memória principal, sendo que cada célula armazena 2 bytes. 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 37 registradores. A máquina contém 72 códigos de operação diferentes.

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

Memória com 512K células => $N = 512K$ células

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

Barramento de endereços (BE) = $\log_2 N = \log_2 512K = 19$ bits

REM = tamanho do BE = 19 bits

- b. (0,3) Indique qual deve ser o tamanho mínimo em bits do barramento de endereços.
tamanho do BE = 19 bits

- c. (0,4) 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 = 19 bits

2°. *Opeando* = endereço de um registrador, um dos 37 diferentes = necessário 6 bits
 cod.operação = tamanho necessário para 72 códigos diferentes = necessário 7 bits
 tamanho da instrução = $7 + 19 + 6 = 32$ bits
 Serão necessárias 2 células (2×2 bytes = 32 bits) para armazenar uma instrução

- 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

Uma instrução ocupa 2 células, sendo assim,

RDM = barramento de dados = 32 bits (2 células).

- 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 = 19 bits

RI = tamanho mínimo necessário para uma instrução = 32 bits

- f. (0,3) Indique o tamanho total da memória.

$N = 2^{19}$ células

$M = 2$ bytes

T (tamanho da memória) = $2^{19} \times 2$ bytes = 1 Mbytes ou 8 Mbits

5. (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 8 ns e cada estágio DE, EX e LD tem duração de 5 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>DE</i>	<i>EX</i>	<i>LD</i>	<i>ER</i>
8ns	5ns	5ns	5ns	8ns

O ciclo de relógio deverá ter o tempo de 31ns

- 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.

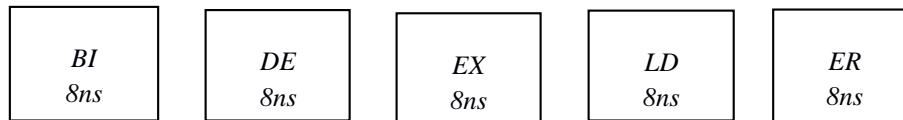
1°. Estágio

2°. Estágio

3°. Estágio

4°. Estágio

5°. Estágio



Ciclo de relógio será igual ao tempo para execução do estágio com maior tempo de execução = 8ns

- c. (1,0) Mostre o tempo em que um programa que contenha 150 instruções será executado pela **implementação do item a e do item b**. Considere que estas 150 instruções podem ser executadas em fluxo constante.

Para o item a (sem pipeline) :

$$T_{ex} = 1 \text{ estágio de } 31ns = 31ns$$

$$T_{total} = 150 \text{ instruções} \times T_{ex} = \mathbf{4.650 \text{ ns}}$$

Para o item b (pipeline: 5 estágios) :

$$T_{ex} = 5 \text{ estágios de } 8ns \text{ cada} = 40ns$$

$$T_{total} = T_{ex} + 149 \times \text{tempo de } 1 \text{ estágio}$$

$$T_{total} = 40ns + 149 \times 8ns = \mathbf{1.232 \text{ ns}}$$

6.