

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Organização de Computadores**  
**Gabarito da AP1 1º semestre de 2010.**

1. (1,5) Explique a função de cada um dos seguintes barramentos vistos em aula: barramento de dados, barramento de endereços e barramento de controle.

- *Barramento de dados tem o objetivo de interligar a CPU à Memória principal, por ele trafegam os dados ou instruções da memória para a CPU (processo de leitura em MP) ou da CPU para a memória (processo de gravação em MP).*
- *Barramento de endereços tem o objetivo de interligar a CPU à Memória principal, por ele trafegam os endereços originados da CPU para a memória principal, a transferência é unidirecional.*
- *Barramento de Controle interliga a CPU com diversos componentes do sistema computacional como a MP e o chipset. Trafegam ali sinais de controle como: leitura, escrita, wait, clock, IRQ, entre outros.*

2. (2,5) Suponha que você deve projetar uma máquina com as seguintes especificações:

- Capaz de endereçar 128 M 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 e um operando como mostrado abaixo: onde Operando é um endereço da memória principal.
- Deve poder ter um máximo de 16 códigos de operação diferentes.

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

*Capacidade de endereçamento = 128 M células  $\Rightarrow N = 128 \text{ M células} \Rightarrow N = 2^{27}$*

*O tamanho necessário para essa capacidade de endereçamento =  $\log_2 N = 27 \text{ bits}$*

*Tamanho necessário de REM para endereçar N células = **27 bits***

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

*O tamanho mínimo do BE deverá ser o necessário para acessar N células = **27 bits***

c) (0,5) Calcule o número de células que uma instrução necessita para ser armazenada.

*A instrução é composta de 1 cód.operação e 1 operando.*

*cód.operação deverá ter o tamanho máximo para 16 códigos de operações diferentes.*

*Cód.operação = 4 bits ( $2^4 = 16$  códigos diferentes)*

*O operando é um endereço da MP. Tamanho do operando = 27 bits.*

*O tamanho da instrução = 4 bits + 27 bits = 31 bits.*

*Cada célula armazena 2 bytes (16 bits), portanto, serão necessárias 2 células para armazenar 1 instrução.*

- d) (0,7) 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 deverá ter o tamanho necessário para transferir 2 células em um acesso à memória principal.  $RDM = 32 \text{ bits}$ .*

*Barramento de dados deverá ter o tamanho do RDM.  $BD = 32 \text{ bits}$ .*

- e) (0,7) Calcule a capacidade de armazenamento em bits dos registradores RI e CI, utilizando-se os valores calculados nos itens anteriores.

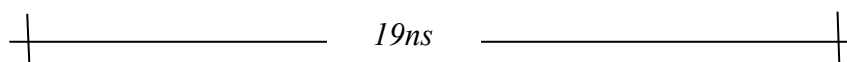
*RI deverá ter o tamanho mínimo de uma instrução, como o tamanho da instrução = 31 bits, RI deverá ter o tamanho mínimo de 31 bits.*

*CI deverá ter o tamanho mínimo p/ endereçar toda a memória. Memória contém  $N$  células CI deverá ter o tamanho mínimo de 27 bits.*

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 5 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>5ns</i>	<i>DE</i> <i>3ns</i>	<i>EX</i> <i>3ns</i>	<i>LD</i> <i>3ns</i>	<i>ER</i> <i>5ns</i>
-------------------------	-------------------------	-------------------------	-------------------------	-------------------------



*Ciclo de relógio para execução de uma instrução (sem pipeline) =*  
 $5ns + 3ns + 3ns + 3ns + 5ns = 19ns$

- 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>5ns</i>	<i>DE</i> <i>5ns</i>	<i>EX</i> <i>5ns</i>	<i>LD</i> <i>5ns</i>	<i>ER</i> <i>5ns</i>

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

- c) (1,0) Mostre o tempo em que um programa que contenha 10 instruções será executado pela **implementação do item a e do item b**. Considere que estas 10 instruções podem ser executadas em fluxo constante.

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) :

$$T_{ex} = 1 \text{ estágio} \times 19ns = 19ns$$

$$T_{total} = 10 \text{ instruções} \times T_{ex} = \underline{190ns}$$

Para o item b (pipeline: 5 estágios) :

$$T_{ex} = 5 \text{ estágios} \times 5ns = 25ns$$

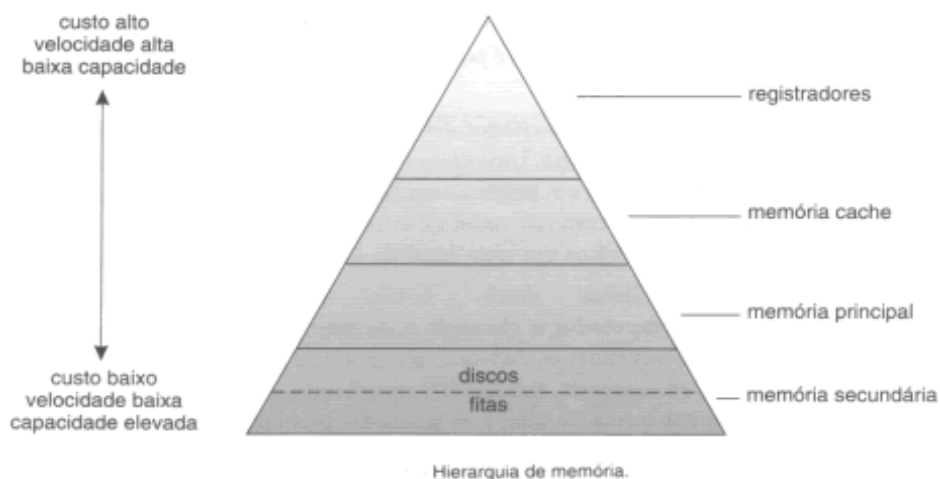
$$T_{total} = T_{ex} + 9 \times \text{tempo de 1 estágio}$$

$$T_{total} = 25ns + 9 \times 5ns = \underline{70ns}$$

4. (2,0) Se a hierarquia de memória fosse representada como uma pirâmide, quais seriam os seus níveis e o que representariam? Explique.

Essa pirâmide seria dividida em 4 níveis hierárquicos. A figura de uma pirâmide nos permite ilustrar uma comparação entre as características de cada memória: Quando mais alto na pirâmide, mais rápida e de maior custo é a memória, por outro lado, quanto mais baixo na pirâmide, menor a velocidade e menor custo/byte da memória. A largura de cada nível representa a capacidade da memória, onde quanto maior for a largura (base da pirâmide) maior a capacidade da memória, e esta decresce conforme vamos subindo de nível na pirâmide.

No 1o. nível (topo da pirâmide) temos os registradores, com baixa capacidade, alto custo e velocidade. Logo abaixo, no 2o. nível, temos a memória cache, com velocidade e custo pouco inferior aos registradores, mas com maior capacidade, ( em PCs comerciais, ordem de alguns MB). No 3o. Nível, temos a Memória principal com custo/byte e velocidade inferior a da cache, mas a sua capacidade é bem maior (na ordem de GB). Ao final da pirâmide temos a memória secundária, esta é a mais lenta, mas com grande capacidade (na ordem de GB ou TB em PCs comerciais), possui ainda como diferencial das anteriores a não volatilidade.



Hierarquia de memória.

5. (2,0) Através de um exemplo de uma arquitetura com 1 Giga bytes de memória física, referenciada por células de 4 bytes e uma memória cache de 1 Mega blocos, cada um com 1024 células, explique as diferenças de mapeamento direto e associativo de memória cache.

*O uso da memória cache tem o objetivo aumentar o desempenho de um computador, pois a cache tem o tempo de resposta menor que a MP, mas com um custo/byte bem superior, e atualmente vem inserida no mesmo chip da CPU. A construção de uma MP com a tecnologia da cache, além do custo alto e talvez proibitivo para computadores comerciais, ocorreriam outros problemas como a necessidade de resfrigeração (similar a CPU).*

*A MP contém um espaço de endereçamento bem superior o da cache. Nos computadores comerciais, a MP está na faixa de alguns GB, e a cache está na ordem de alguns MB. Se a cache tivesse o mesmo tamanho da MP, esta não seria mais necessária. O mapeamento vem para estabelecer uma relação entre os espaços de endereçamento das duas memórias (MP e cache) a fim de verificar se um determinado endereço procurado pelo processador estaria na cache antes de fazer a transferência a partir da MP. A transferência de conteúdos da MP para a cache é feita em blocos. Um bloco corresponde a um conjunto de palavras, ou células. Cada linha da cache tem a capacidade de armazenamento de um bloco.*

*Dois tipos básicos de mapeamento são utilizados: mapeamento direto e o mapeamento totalmente associativo. O mapeamento associativo por conjunto é uma combinação dos 2 básicos anteriores. A principal diferença entre os 2 modos, é que no mapeamento direto, um bloco da MP só poderá ser alocado em uma mesma linha da cache. Já no mapeamento associativo, um bloco da MP poderá ser alocado em qualquer linha da cache. Cada linha da cache possui um indentificador (**tag**) que informa qual bloco da MP está ali alocado.*

*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.*

*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.*

#### **Como resposta para a questão:**

*MP com 1 Giga bytes, e cada célula com 4 bytes, então, a MP possui 256M células. A memória cache tem 1 Mblocos, cada bloco contém 1024 células, teremos uma cache com 1G células.*

*Como o espaço de endereçamento da MP é menor que a da cache, não será necessário estabelecer um mapeamento, já que todo o conteúdo da MP poderá ser alocado na cache. Se for viável a construção comercial de uma MP com as características e tecnologia dessa cache, não haveria a necessidade de termos uma cache.*