

AD1 - Organização de Computadores 2007.2
Data de entrega 08/09/2007

1. (1,0) Preencha a tabela abaixo:

	Tamanho da célula (M)	No. bits do endereço (E)	Capacidade de memória (T)	N = no. de células (N)
(a)	8 bits	15 bits	32 Kbytes	32 K células
(b)	2 bytes	21 bits	32 M bits	2 Mcélulas
(c)	32 bits	4 bits	512 bits	16 células
(d)	32 bits	31 bits	8 Gbytes	2 G células

Fórmulas: $T \text{ (em bits)} = M \text{ (em bits / célula)} \times N \text{ (em no. de células)}$
 $2^E = N$, sendo E em bits e N no. de células.

(a) $N = ? \Rightarrow N = T / M \Rightarrow N = (32 \text{ Kbytes} \times 8\text{bits}) / (8\text{bits} / \text{célula})$
 $N = 32 \text{ K células}$
 $E = ? \Rightarrow 2^E = N \Rightarrow 2^E = 32 \text{ K} \Rightarrow 2^E = 2^{15} \Rightarrow E = 15 \text{ bits}$

(b) $E = ? \Rightarrow 2^E = N \Rightarrow 2^E = 2 \text{ M} \Rightarrow 2^E = 2^{21} \Rightarrow E = 21 \text{ bits}$
 $T = M \times N \Rightarrow T = (2 \text{ bytes} \times 8\text{bits} / \text{célula}) \times (2 \text{ M células})$
 $T = 32 \text{ Mbits ou } 4 \text{ Mbytes de capacidade}$

(c) $N = ? \Rightarrow 2^E = N \Rightarrow 2^4 = 16 \text{ células} \Rightarrow N = 16 \text{ células}$

$M = ? \Rightarrow M = T / N \Rightarrow M = (512\text{bits}) / (16 \text{ células})$
 $M = 32 \text{ bits ou } 4 \text{ bytes por célula}$

(d) $N = ? \Rightarrow N = T / M \Rightarrow N = (8 \text{ Gbytes} \times 8\text{bits}) / (32\text{bits} / \text{célula})$
 $N = 2 \text{ G células}$
 $E = ? \Rightarrow 2^E = N \Rightarrow 2^E = 2 \text{ G} \Rightarrow 2^E = 2^{31} \Rightarrow E = 31 \text{ bits}$

2. (1,0) Considere uma máquina que possa endereçar 512 Mbytes de memória física, utilizando endereço referenciando byte, e que tenha a sua memória organizada em blocos de 32 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 (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) = 512Mbytes, como 1 célula referencia a 1 byte, temos $N = 512M$ células

⇒ Será organizada em blocos de 32 bytes, como 1 célula = 1 byte, temos cada bloco = 32 células, $K = 32$

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

$N = 512M$ células e $K = 32$ células / blocos o total de blocos da MP (B) será:

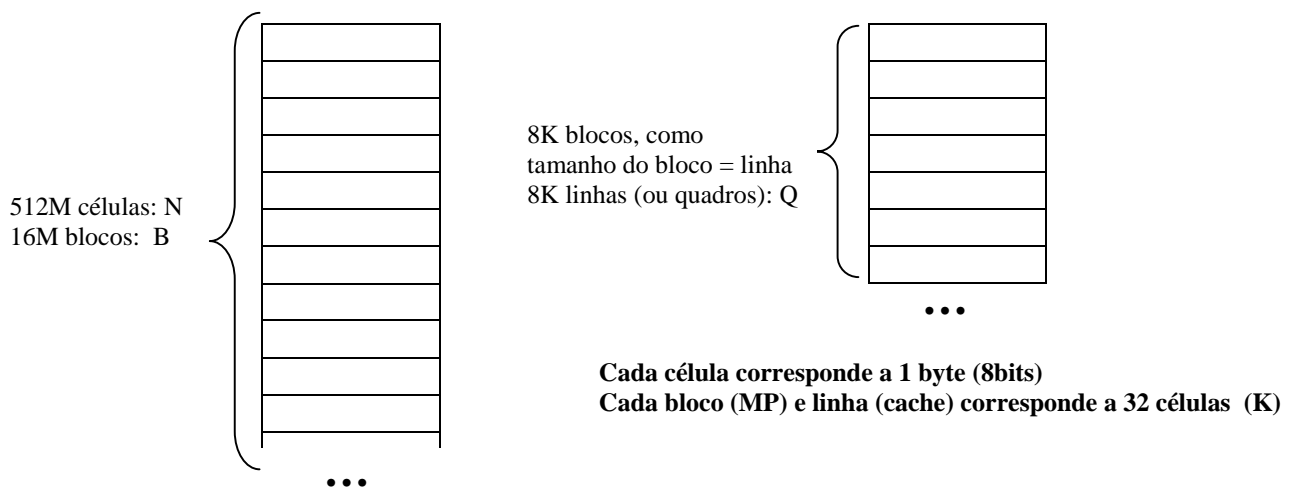
Total de blocos: $B = N / K \Rightarrow B = 512M \text{ células} / 32 \text{ células/bloco} \Rightarrow B = 16 M \text{ blocos}$

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

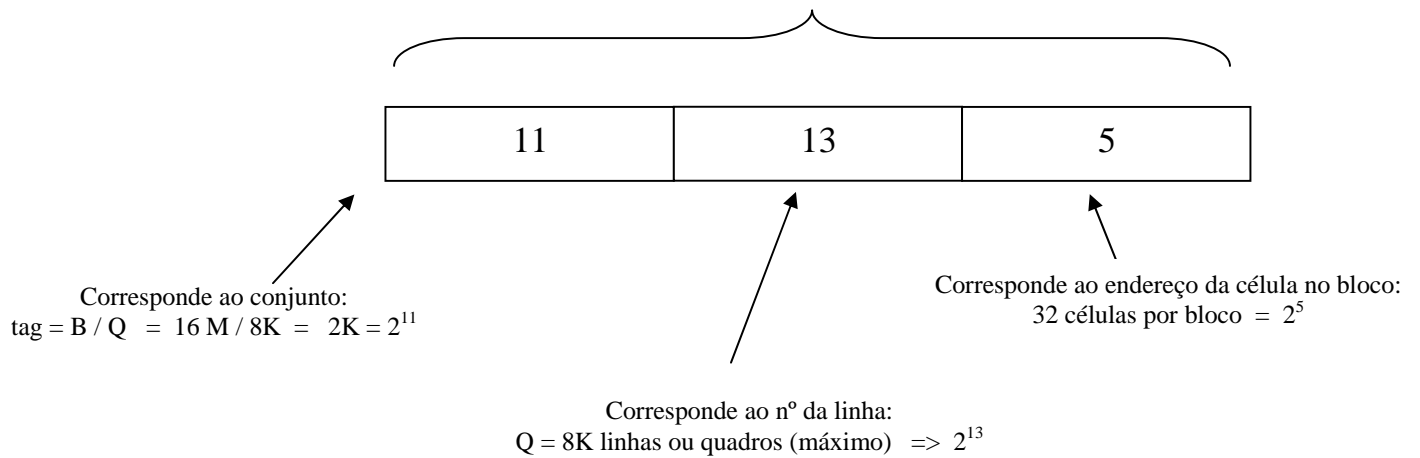
⇒ Tamanho da memória cachê em células = $Q \times K = 8K \text{ blocos} \times 32 \text{ células/blocos} = 256K \text{ células}$



Para endereçarmos toda a MP precisamos da seguinte quantidade de bits (E)

sendo $N = 2^E \Rightarrow N = 512M \text{ células} \Rightarrow N = 2^{29} \Rightarrow E = 29 \text{ bits}$

Tamanho do endereço da MP = 29 bits



b) Mapeamento totalmente associativo.....

Memória Principal

=> N = 512M células

=> K = 32

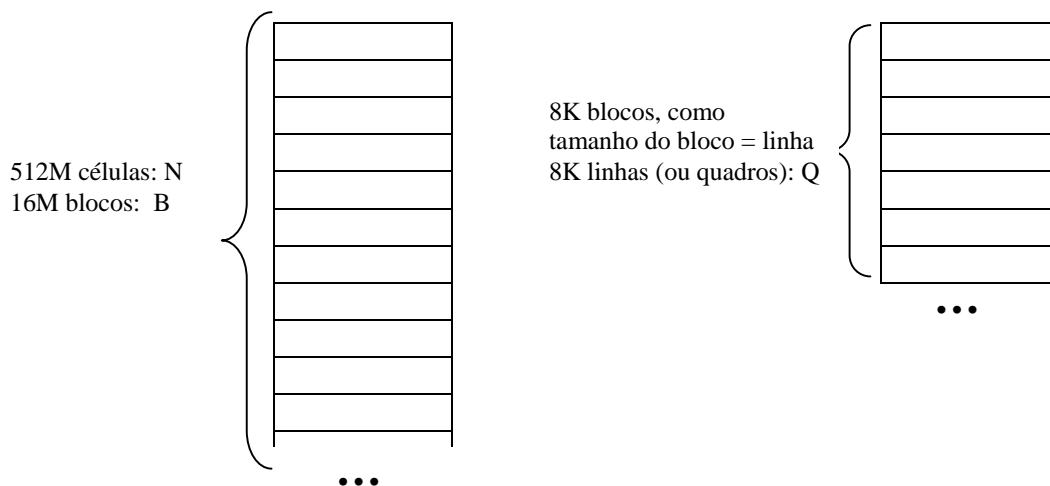
=> B = 16 M blocos

Memória Cache

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

=> Q = 8K blocos

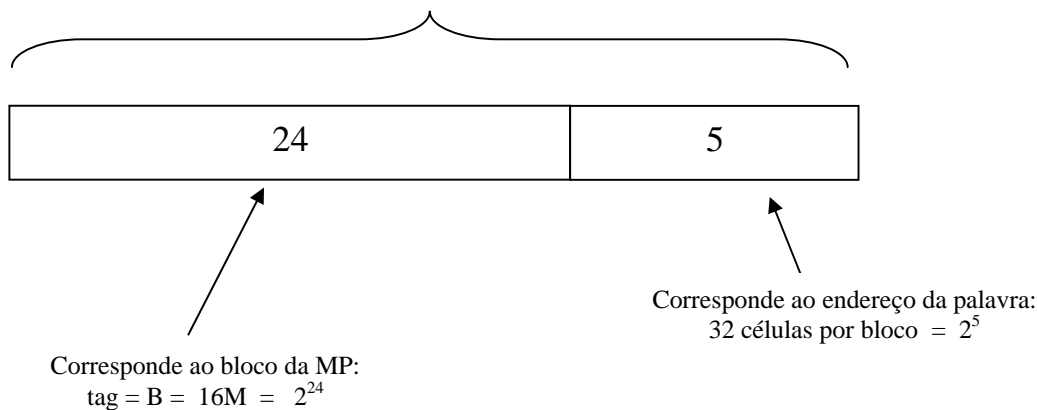
=> Tamanho da memória = 256K células



Para endereçarmos toda a MP precisamos da seguinte quantidade de bits: E = 29 bits

Como o bloco pode ser alocado em qualquer posição da memória cache, a tag indicará qual dos blocos da MP está alocado naquela posição da memória cache

Tamanho do endereço da MP = 29 bits



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

Memória Principal

=> N = 512M células

=> K = 32

=> B = 16 M blocos

Memória Cache

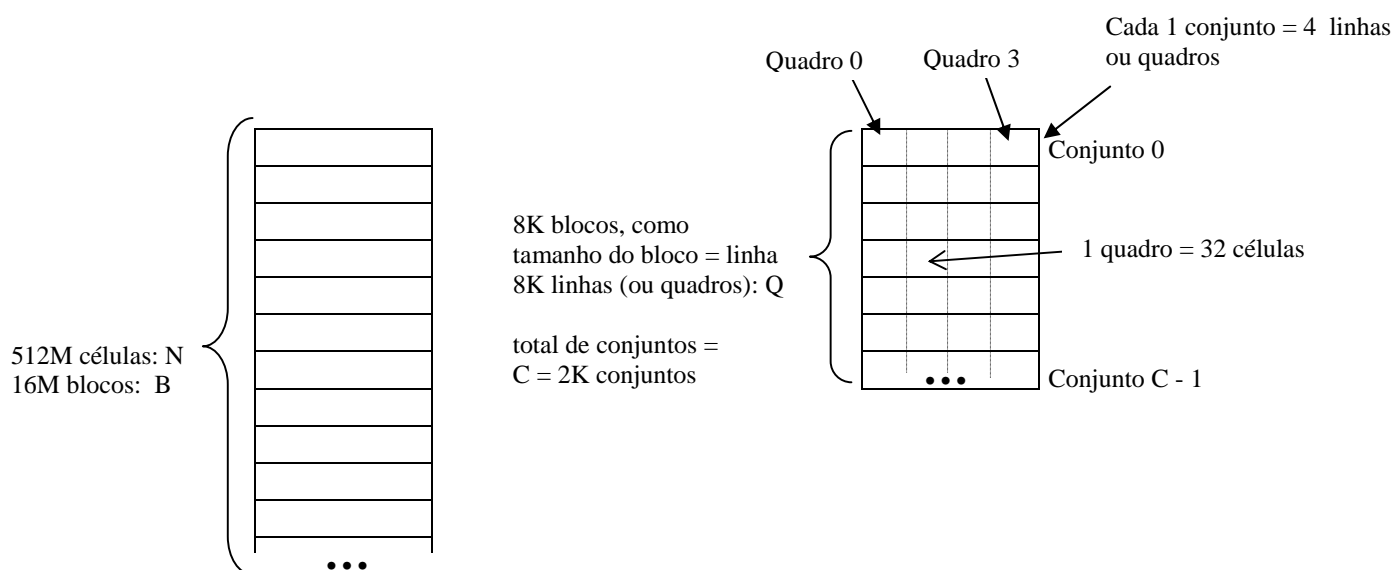
OBS: O K (quantidade de células/bloco) tem de ser igual a MP.

=> Q = 8K blocos

=> Tamanho da memória cache = 256K células

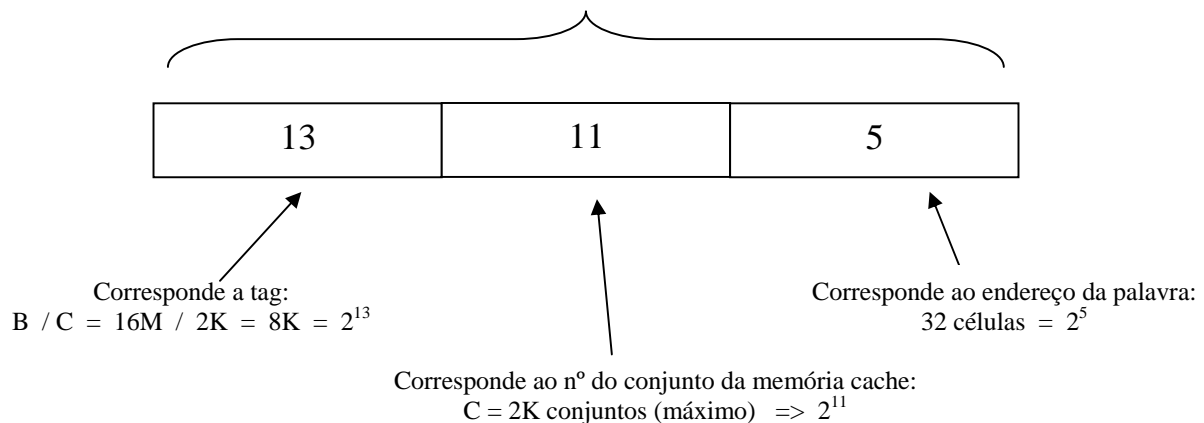
=> 1 conjunto = 4 linhas (ou quadros) =>

Total de conjuntos => C = 8K blocos / 4 => **C = 2K conjuntos**



Para endereçarmos toda a MP precisamos da seguinte quantidade de bits: $E = 29$ bits

Tamanho do endereço da MP = 29 bits



3. (1,0) Explique em detalhes a organização hierárquica do subsistema de memória nos computadores atuais.

Há muitas memórias no computador: O subsistema de memória é interligado de forma bem estruturada e organizado hierarquicamente em uma pirâmide com os níveis descritos a seguir.

No topo da pirâmide teríamos os registradores, que são pequenas unidades de memória que armazenam dados na UCP. São dispositivos de maior velocidade de transferência, menor capacidade de armazenamento e maior custo.

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.

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.

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: discos rígido, outros são conectados quando necessário: disquetes, fitas de armazenamento, CD-ROM. Possuem menor velocidade de transferência, maior capacidade de armazenamento e menor custo.

4. (1,0) Faça uma pesquisa no livro “Arquitetura e Organização de Computadores” de William Stallings e descreva as organizações das Memórias Cache do Pentium II e do Power PC.

As caches de ambos processadores são organizadas em 2 etapas, sendo uma interna ao processador (L1) e outra externa (L2) de forma a alimentar a anterior. A L1 presente na pastilha do processador constitui-se de 2 memórias caches, sendo uma para dados e outra para instruções. Cada uma possui 8 Kbytes de capacidade e utilizam-se do mapeamento associativo por conjuntos de 4 linhas (para a de instruções) e de 2 linhas (para a de dados). A L2 possui capacidade de 256 Kbytes a 1 Mbyte com mapeamento associativo por conjunto de quatro linhas.

5. (1,0) Considere um microprocessador hipotético com um tamanho de palavra e barramento de dados de 40 bits e 256 instruções. As instruções são todas do tamanho de uma célula de memória, de mesmo tamanho da palavra e são compostas por dois campos: o primeiro contém o código de operação e o restante contém o endereço do operando. Ele contém somente uma placa de memória com 5 Kbytes.

a) Qual o tamanho, em bits, do RDM e do REM?

⇒ RDM = tamanho do barramento de dados = 40 bits

⇒ REM = tamanho do campo do operando, corresponderá ao tamanho de endereço de memória.
Como o processador tem 256 instruções, o seu código de operação será de 8 bits ($2^8 = 256$)
Tamanho de instrução = código de operação + operando ⇒ 40bits = 8bits + operando.
Operando = 32 bits, portanto REM = 32 bits

b) Seria possível aumentar a memória desta máquina ? Se é possível, até quanto (em bytes).

⇒ Como o barramento de endereços = REM = 32bits
A capacidade de endereçamento será de $2^{32} = 4 \text{ G células}$.
Será possível aumentar a memória até 4 G células
Como cada célula possui 40 bits (1 palavra) = 5 bytes
Total de memória = 4G células x 5 bytes = 20 Gbytes

c) Se quisesse aumentar o número de instruções para 512, mantendo inalterado o tamanho do barramento de dados e da palavra, quais seriam os tamanhos do RDM e REM.

⇒ RDM = tamanho do barramento de dados = 40 bits

⇒ REM = tamanho do campo do operando, corresponderá ao tamanho de endereço de memória.
Como o processador tem 512 instruções, o seu código de operação será de 9 bits ($2^9 = 512$)
Tamanho de instrução = código de operação + operando ⇒ 40bits = 9bits + operando.
Operando = 31 bits, portanto REM = 31 bits

6. (1,0) Considere uma máquina com 256 K células de memória onde cada célula armazena uma palavra e cada instrução tem o tamanho de uma palavra. Esta máquina possui um conjunto de instruções com 64 códigos de operação distintos, sendo cada uma delas composta do código de operação e um único operando, que indica o endereço de uma célula de memória. A leitura de uma célula de memória requer dois acessos à memória. Esta máquina utiliza os registradores ACUMULADOR, REM, RI, CI e RDM da mesma maneira que a máquina mostrada em aula.

a) (0,1) Qual o tamanho mínimo do REM?

$$\Rightarrow \text{REM} = E \Rightarrow 2^E = 256\text{K células} = 2^{18} \text{ células} \Rightarrow E = 18 \Rightarrow \text{REM} = 18 \text{ bits}$$

b) (0,1) Qual o tamanho mínimo do RI?

$$\begin{aligned} \Rightarrow & \text{Para termos 64 códigos de operação. Serão necessários 6 bits } (2^6 = 64) \\ & \text{Tamanho da instrução} = \text{código de operação} + 1 \text{ operando} = 6 \text{ bits} + 18 \text{ bits (endereço)} \\ & = 24 \text{ bits} \end{aligned}$$

c) (0,1) Qual o tamanho mínimo do CI

$$\begin{aligned} \Rightarrow & \text{CI terá que ter um tamanho que possa fazer referência a qualquer endereço da memória,} \\ & \text{portanto:} \\ & \text{CI} = \text{REM} = E \Rightarrow \text{CI} = 18 \text{ bits} \end{aligned}$$

d) (0,1) Qual o tamanho mínimo do RDM?

$$\Rightarrow \text{Como no exercício está descrito que são necessárias 2 operações de leitura para transferência de uma célula que é o mesmo tamanho da palavra, o RDM deverá ter o tamanho da metade da palavra (24 bits), neste caso, deverá ser de 12 bits, portanto, RDM} = 12 \text{ bits.}$$

e) (0,1) Qual o tamanho da memória em bits?

$$\begin{aligned} \Rightarrow & \text{Quantidade de bits} = 256\text{K células} \\ & \text{Cada célula armazena uma palavra e cada palavra corresponde ao tamanho de uma instrução, conforme o enunciado, temos então:} \\ & \text{Quantidade de bits (T)} = 256 \text{ K células} \times \text{tamanho de cada célula (ou tamanho da instrução neste caso)} \Rightarrow \\ & 256\text{K células} \times 24 \text{ bits/célula} = 2^{18} \times 24 = 6096 \text{ Kbits} \end{aligned}$$

f) (0,1) Qual a largura do barramento de dados?

$$\Rightarrow \text{Como no exercício está descrito que serão necessários 2 operações de leitura para transferência de uma célula que é o mesmo tamanho da palavra, o barramento de dados deverá ter o tamanho da metade da palavra (24 bits), neste caso, deverá ser de 12 bits, portanto, barramento de dados} = 12 \text{ bits.}$$

g) (0,4) Esta máquina possui a instrução **ADDM Op.** que soma o conteúdo da memória no endereço Op com o conteúdo do acumulador e coloca o resultado na memória no endereço Op. Indique como os registradores RDM, REM, CI e RI e os barramentos de dados, de controle e de endereços são utilizados na execução desta instrução.

1. O endereço da instrução é colocado no CI.
2. O endereço é transferido para o REM.
3. O endereço é transferido para a memória através do barramento de endereços.
4. O sinal de leitura é acionado no barramento de controle.
5. Os primeiros 12 bits da instrução são transferidos através do barramento de dados para o RDM , e depois são transferidos para o RI.
6. Os próximos 12 bits da instrução são transferidos através do barramento de dados para o RDM , e depois são transferidos para o RI.
7. A instrução é decodificada.
8. O endereço Op é armazenado no REM.
9. O conteúdo de REM é transferido para a memória através do barramento de endereços.
10. O sinal de leitura é acionado no barramento de controle.
11. Os primeiros 12 bits da memória são transferidos através do barramento de dados para o RDM , e depois são transferidos para a ULA.
12. Os próximos 12 bits da memória são transferidos através do barramento de dados para o RDM , e depois são transferidos para a ULA.
13. O conteúdo do acumulador é enviado para a ULA e um sinal de controle de soma é enviado para a ULA.
14. Os primeiros 12 bits do resultado da ULA são transferidos para o RDM.
15. O endereço Op é armazenado no REM.
16. O conteúdo de REM é transferido para a memória através do barramento de endereços.
17. O conteúdo de RDM é transferido para a memória através do barramento de dados.
18. O sinal de escrita é acionado no barramento de controle.
19. Os próximos 12 bits do resultado da ULA são transferidos para o RDM.
20. O conteúdo de REM é transferido para a memória através do barramento de endereços.
21. O conteúdo de RDM é transferido para a memória através do barramento de dados.
22. O sinal de escrita é acionado no barramento de controle.

7. (1,0) Explique o que é uma placa-mãe e os componentes que se encontram nela. (Fonte de consulta: <http://pt.wikipedia.org/>)

⇒ Definição:

Placa-mãe, também denominada mainboard ou motherboard, é uma placa de circuito impresso eletrônico. É considerado o elemento mais importante de um computador, pois tem como função permitir que o processador se comunique com todos os periféricos instalados.

⇒ Componentes que se encontram nela:

Na placa-mãe encontramos não só o processador, mas também a memória Principal (RAM e a Bios), os circuitos de apoio (chipsets), os slots para conexão das placas expansoras (ISA, PCI, PCI Express, AGP), conectores de dispositivos (USB, serial, mouse, teclado, paralela, etc), e componentes de interfaces que possam já vir integradas à placa mãe como vídeo, rede, som (chamado de “on board”).

- O Processador: Responsável pela interpretação e execução das instruções contidas no software.
- A memória RAM é um tipo de memória que permite a leitura e a escrita, utilizada como memória primária em sistemas eletrônicos digitais. Nesta memória são carregados os programas em execução e os dados a serem processados.
- Os circuitos de apoio correspondem aos circuitos que permitem a utilização dos processadores, como o clock. Todos os elementos lógicos do computador foram projetados para operarem em sincronismo com o clock. Eles executam as operações que lhes cabem passo a passo e cada circuito, executa um passo ao mesmo tempo que todos os circuitos restantes do computador. Sendo então, o clock responsável por estabelecer a marcação deste sincronismo.
- Uma placa controladora é uma parte do hardware do computador que comanda outras partes da máquina. Normalmente é conectada na placa-mãe através de slots apropriados de acordo com o barramento relativo à placa. Estas placas também podem ser consideradas como placas de expansão que são dispositivos que se utilizam para estender as funcionalidades e o desempenho do computador. Existe uma grande diversidade de placas de expansão, como, por exemplo, placas de rede, de vídeo, de som e de modem.
- De uma forma geral, um *chipset* (anglicismo que significa *grupo de chips*) é o cérebro de uma placa de circuitos. se dividindo entre "ponte norte" (*north bridge*, controlador central) e "ponte sul" (*south bridge*, controlador de periféricos). A ponte norte faz a comunicação do processador com as memórias, e em alguns casos com os barramentos de alta velocidade AGP e PCI Express. Já a ponte sul, abriga os controladores de HDs (ATA/IDE e SATA), portas USB, paralela, PS/2, serial, os barramentos PCI e ISA, que já não é usado mais em placas-mãe modernas. O chipset é quem define, entre outras coisas, a quantidade máxima de memória RAM que uma placa-mãe pode ter, o tipo de memória que pode ser usada (SDRAM, DDR-SDRAM, Rambus, etc.), a frequência máxima das memórias e do processador e o padrão de discos rígidos aceitos (UDMA/33, UDMA/66, etc.).

8. (1,0) Para uma máquina que possui pipeline de instruções funcione no máximo do seu desempenho, deve-se assegurar um fluxo constante de instruções, ou seja, quando uma instrução finaliza seu ciclo de Busca, deve-se imediatamente inicializar o ciclo de Busca da próxima instrução a ser executada. Explique como a existência de desvios condicionais impede a constância deste fluxo. Faça uma pesquisa no livro “Arquitetura e Organização de Computadores” de William Stallings e explique 5 maneiras para lidar com desvios condicionais de modo a tentar diminuir seus efeitos negativos no desempenho da máquina.

O Desvio condicional consiste no principal impedimento para um fluxo constante de instruções em um pipeline de instruções. Diversas abordagens são adotadas para lidar com o desvio condicional, entre elas:

Múltiplos fluxos: Consiste em duplicar os estágios iniciais da pipeline para permitir a busca de ambas as instruções, usando assim dois fluxos de instruções. Esta abordagem apresenta problemas como o atraso à contenção de acesso a registradores e à memória, além da entrada de instruções adicionais na pipeline antes que seja tomada a decisão sobre o desvio original.

Busca antecipada da instrução-alvo de desvio. Consiste em buscar antecipadamente tanto a instrução-alvo de desvio quanto a instrução consecutiva ao desvio, no instante em que uma instrução de desvio-condicional é reconhecida. A instrução-alvo é armazenada em um registrador, até que a instrução de desvio seja executada. Seja o desvio tomado ou não, a próxima instrução a ser executada terá sido buscada antecipadamente.

Memória de laço de repetição: Consiste em usar uma pequena memória de alta velocidade, mantida pelo estágio de busca de instrução da pipeline, que é usada para manter as n instruções buscadas mais recentemente, em sequência. Na ocorrência do desvio, será verificado se a instrução-alvo está presente nesta memória.

Previsão de desvio: Consiste em várias técnicas que podem ser usadas para prever se um desvio será tomado ou não, entre as mais comuns estão as seguintes.

- Prever que o desvio nunca será tomado: esta previsão não depende do histórico de execução de instruções até o momento, supõe que o desvio não seja tomado e continua a buscar instruções na sequência em que ocorrem no programa.
- Prever que o desvio sempre será tomado: conforme a anterior diferindo que o desvio será tomado, buscando sempre as próximas instruções a partir do endereço-alvo do desvio.
- Prever se o desvio será tomado ou não conforme o código de operação: em casos de determinados códigos de operação, o processador supõe que o desvio sempre é tomado; em outros, que o desvio nunca é tomado, estudo feito com esta técnica apresenta taxas de sucesso superiores a 75%.
- Prever o desvio com base em chaves de desvio tomado e de desvio não tomado: consiste em uma estratégia dinâmica onde um ou mais bits podem estar associados a cada instrução de desvio condicional, usados para refletir a história recente da execução da instrução. Bits estes chamados de chaves de desvio tomado ou de desvio não tomado.
- Prever o desvio com base em uma tabela de histórico de desvios: Esta técnica mantém uma pequena tabela de histórico de instruções de desvio executadas mais recentemente, incluindo um ou mais bits de entrada. O processador pode endereçar essa tabela de maneira associativa, tal como uma memória cache, ou usar bits de mais baixa ordem do endereço das instruções de desvio. A tabela de histórico de desvios é uma pequena memória cache associada ao estágio de busca de instrução da pipeline.

Atraso de desvio: técnica para reordenar automaticamente as instruções de um programa, de modo que instruções de desvio ocorram mais tarde do que ocorrem de fato na sequência especificada.

9) Para cada um dos casos abaixo, calcule a banda passante máxima (número de bits/segundo) do barramento e indique o sistema que possui maior banda passante.

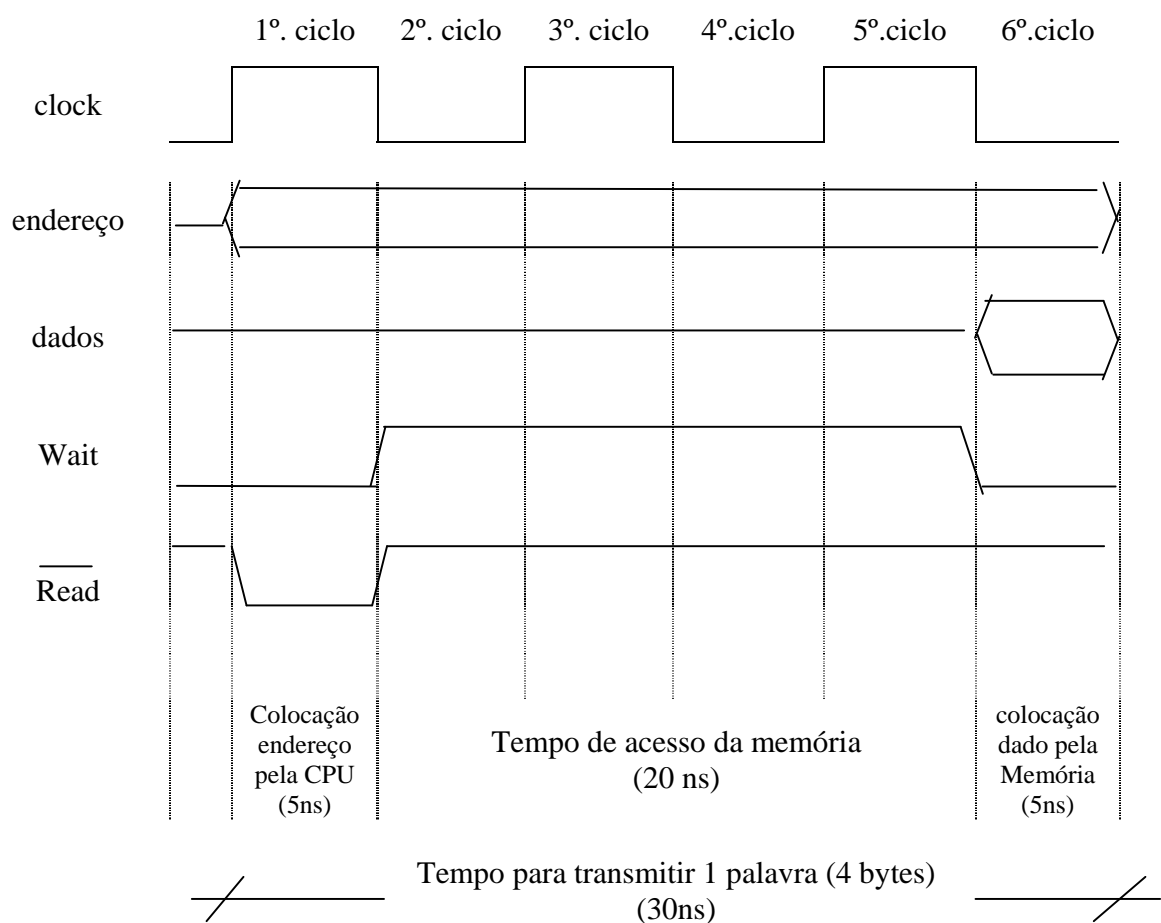
a) Um sistema que possui um barramento paralelo síncrono utilizado para comunicação entre a memória e a UCP. A largura do barramento (número de bytes que podem transferidos em paralelo) é 4 bytes. O relógio utilizado neste barramento possui frequência de 200 MHz. Cada operação de transferência no barramento (envio de um endereço à memória ou envio ou recebimento para/de uma

palavra da memória) utiliza um ciclo de relógio. O acesso à memória tanto para leitura quanto para escrita demanda 20 ns. O endereço e a palavra deste sistema possuem 4 bytes.

Considere que para a transferência de dados (leitura por exemplo) precisaremos dos seguintes passos:

- 1) envio do endereço no barramento à memória
- 2) esperar o acesso da memória
- 3) colocação dos dados no barramento pela memória.

OBS: Considerando o barramento de endereços como sendo separado do barramento de dados, ambos com 4 bytes



Determinação dos tempos:

- 1) envio do endereço no barramento à memória \Rightarrow Tempo = 1 ciclo de clock \Rightarrow
tempo = 1 segundo / 200.000.000 ciclos por segundo \Rightarrow tempo = 5 ns
- 2) esperar o acesso da memória = 20 ns

3) a memória colocar os dados no barramento. = tempo de 1 ciclo de clock = 5 ns

Tempo total para transmissão de 1 palavra = 5ns + 20ns + 5ns = 30ns

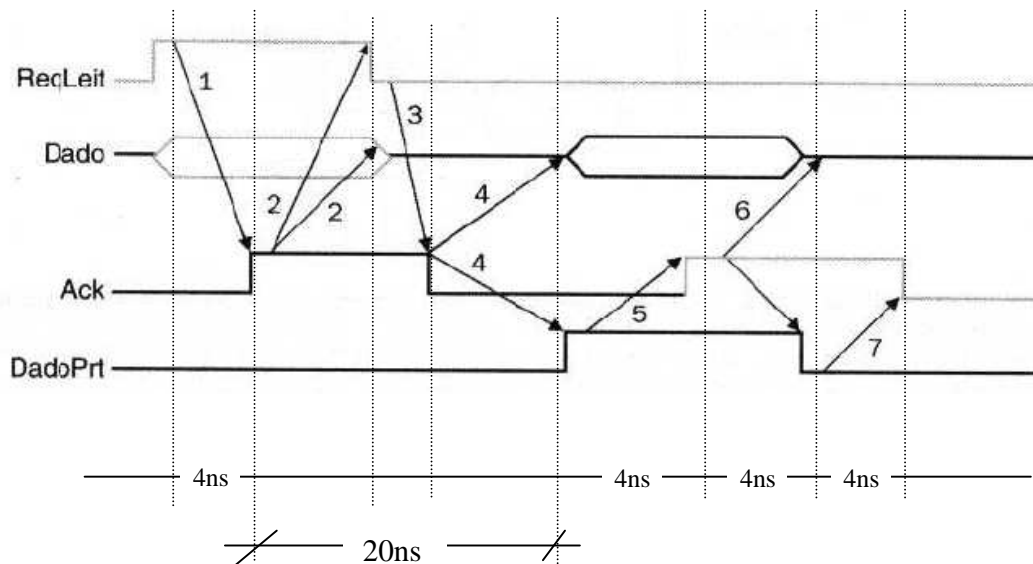
Taxa de transmissão = quantidade de palavras enviadas em 1 segundo x tamanho da palavra.

Taxa de transmissão = $(1s / 30ns) * 4 \text{ bytes} = 133.333.333 \text{ bytes} \approx \mathbf{133 \text{ Mbytes/s}}$ ou seja **1067Mbits/s**

OBS: Para taxas de transmissão, considerar 1Kbyte = 1.000bytes; 1Mbyte = 1.000.000bytes

b) Um sistema que possui um barramento paralelo assíncrono que funciona como mostrado na figura abaixo.

A UCP requer uma leitura ativando a linha ReqLeit e colocando o endereço desejado na linha Dado. A memória reconhece o pedido, e ativa a linha Ack (passo 1). A UCP recebe o sinal Ack e desativa as linhas ReqLeit e retira o endereço da linha Dado (passo 2). A memória desativa o sinal Ack (passo 3), coloca a palavra na linha Dado e ativa o sinal DadoPrt (passo 4). A UCP recebe a palavra e ativa o sinal Ack (passo 5). A memória reconhece que a UCP já recebeu a palavra, a retira da linha Dado e desativa o sinal DadoPrt (passo 6). A UCP percebe que o sinal DadoPrt foi desativado e desativa o sinal Ack (passo 7). Cada um dos 7 passos dura 4 ns e a memória demora 20 ns para liberar a palavra, a partir do momento que ela reconhece um pedido e ativa a linha Ack. Cada palavra possui 4 bytes.



O tempo para cada passo será de 4ns sendo que o tempo para a memória liberar a palavra pela memória é de 20ns, este tempo se inicia a partir do reconhecimento, ou seja, acionamento da linha de Ack (fim do passo 1) até a memória colocar o dado no barramento (fim do passo 4).

Portanto o tempo dos passos 2, 3 e 4 somados: será o maior entre os 2 tempos (a soma do tempo individual de cada passo, cada um com 4ns cujo total é 12ns; ou o tempo de resposta da memória que é de 20ns). Concluímos que o tempo gasto nos passos 2, 3 e 4 é de 20ns.

O tempo total será a soma dos passos 1, 5, 6 e 7 cada um com 4ns + soma dos tempos dos passos 2,3,4 (que é 20ns).

Tempo total = 4ns (passo1) + 20ns (passo2,3,4) + 4ns (passo5) + 4ns (passo6) + 4ns (passo7)
Tempo total para transmissão de 1 palavra = 36ns

Taxa de transmissão = quantidade de palavras enviadas em 1 segundo x tamanho da palavra.
Taxa de transmissão = $(1s / 36ns) * 4 \text{ bytes} = 111.111.111 \text{ bytes} \approx \mathbf{111 \text{ Mbytes/s}}$ ou seja **889Mbits/s**

OBS: Para taxas de transmissão, considerar 1Kbyte = 1.000bytes; 1Mbyte = 1.000.000bytes

Comparação entre estes barramentos

Dos casos apresentados: a) Síncrono e b) Assíncrono. O barramento Síncrono possui a maior taxa de transmissão: **1067Mbits/s**

10. (1,0) Explique as 2 maneiras de implementação de uma unidade de controle: por hardware e microprogramada. Faça uma pesquisa na Internet para descobrir como a unidade de controle dos processadores que implementam o conjunto de instruções X86 é implementada. Indique sua fonte de consulta.

A unidade de controle por hardware é constituída de um circuito combinatório. Seus sinais lógicos de entrada são transformados em um conjunto de sinais lógicos de saída, que constituem os sinais de controle. (ref. William Stallings).

A unidade de controle microprogramada é aquela em que a lógica da unidade de controle é especificada por um microprograma. O projeto desta unidade visa executar uma seqüência de microinstruções (ou conjunto de microoperações) e gerar sinais de controle para os componentes da UCP para a execução de cada microinstrução.

O conjunto de instruções X86 é implementado por microprogramação.

Retirado do livro Organização estruturada de computadores do Andrew S. Tanenbaum. Considerações sobre o Pentium:

No Pentium II a decodificação consiste na conversão de cada instrução do conjunto da Intel X86 (padrão utilizado pelo Pentium é o IA-32) em uma ou mais microoperações. As instruções IA-32 podem precisar de até quatro microoperações. As poucas instruções extremamente complexas precisam de mais microoperações, e usam a ROM seqüenciadora de microoperações para estabelecer a seqüência de execução.