

aula 5: **MEMÓRIA**

disciplina: **Organização e Arquitetura
de Computadores**

professora: **Sara Guimarães Negreiros**

“ Agradecimento especial ao professor **Pedro Souza** por ter sido referencial para o conteúdo e artes (a maioria) utilizadas nessas aulas.

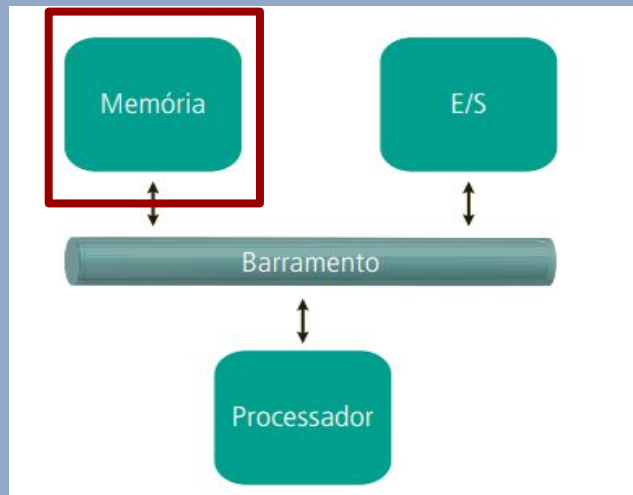
1

VISÃO GERAL



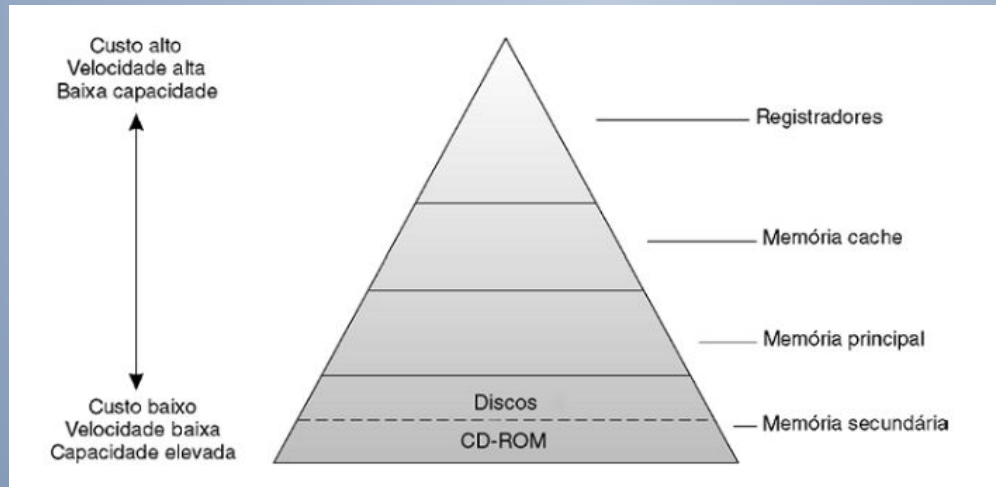
Memória no computador

- Armazenamento de dados e programas.



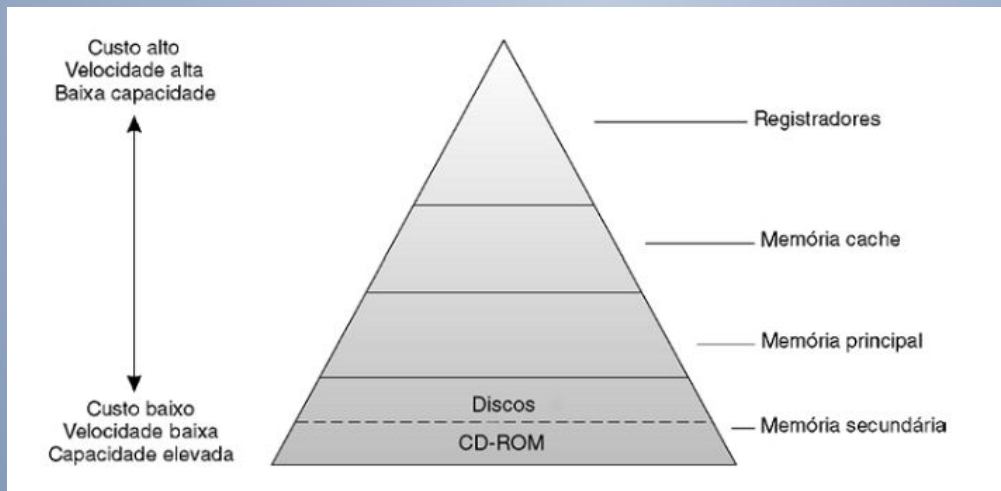
- Armazenamento varia desde dados como arquivos, números, endereços de memória até instruções inteligíveis para o computador.
- Todos os dados são armazenados em bits.
- Dispositivos com diferentes taxas de comunicação.
- Sistema hierárquico de memória para otimizar custo e agilidade.

- Capacidade, velocidade e custo.
 - ▷ Velocidade próxima da velocidade do processador.
 - ▷ Custo compatível com o custo dos demais componentes.



ALGUMAS RELAÇÕES VÁLIDAS

- Tempo de acesso mais rápido, custo por bit maior.
- Capacidade maior, custo por bit menor.
- Capacidade maior, tempo de acesso menor.





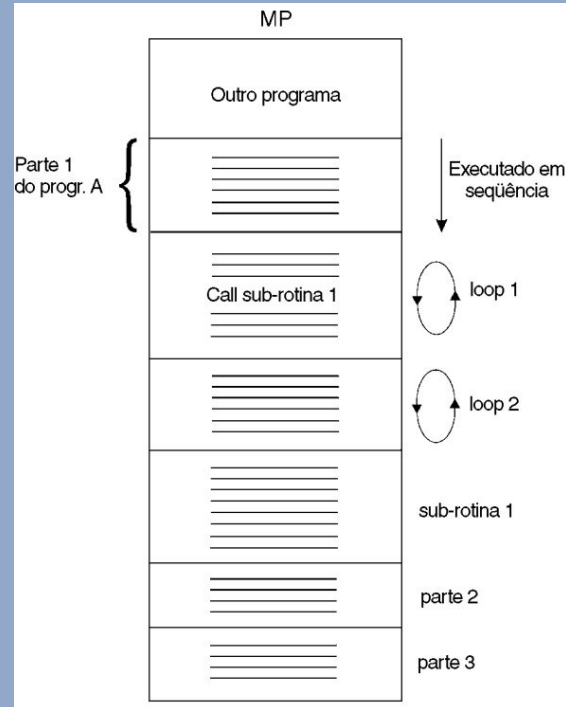
Princípio da localidade

- Sistema hierárquico é possível devido a troca de dados entre as memórias.
- Princípio da localidade otimiza a troca de dados.
- Algoritmos para manter na cache os dados mais utilizados pelo processador.
- Localidade espacial e temporal.



Localidade Espacial

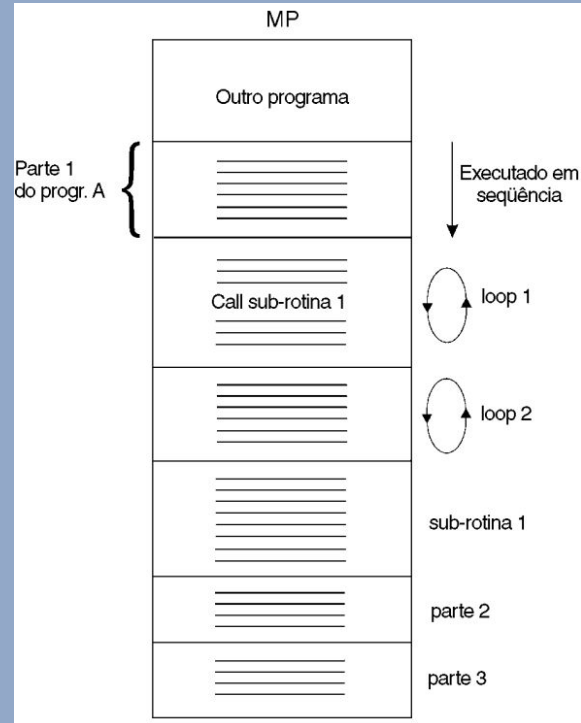
- Programas sequenciais.
- Armazenamento do código sequencial.





Localidade Temporal

- Estruturas de controle de repetição.
- Acesso constante a uma mesma palavra.





Características

LOCALIZAÇÃO

- Memórias internas e externas (acessíveis por módulos E/S).

CAPACIDADE

- Bytes = 8bits. Bytes ou palavras são usualmente de 8, 16 ou 32 bits.

UNIDADE DE TRANSFERÊNCIA DE DADOS

- Palavra ou blocos

MÉTODO DE ACESSO AOS DADOS

- **Sequencial:** sequencia linear específica (unidades de fita)
- **Direto:** acesso direto a uma vizinhança e depois acesso sequencial (unidades de disco)
- **Aleatório:** posições acessadas de modo aleatório pelo endereço (memória principal e cache)
 - ▷ **Associativo:** palavra é buscada com base no conteúdo e não no endereço (memória cache)

DESEMPENHO

- **Tempo de acesso** decorrido entre a apresentação do endereço à memória e o armazenamento/disponibilidade dos dados.
 - ▷ Não-aleatório: tempo decorrido para posicionar o mecanismo leitura/escrita.
 - ▷ Aleatório: tempo para um segundo acesso devido desaparecimento transiente linhas de sinais ou regeneração dos dados.
- **Taxa de transferência.**

TECNOLOGIA

- Semicondutor, magnética, ópticas e magneto-ópticas.

CARACTERÍSTICAS FÍSICAS

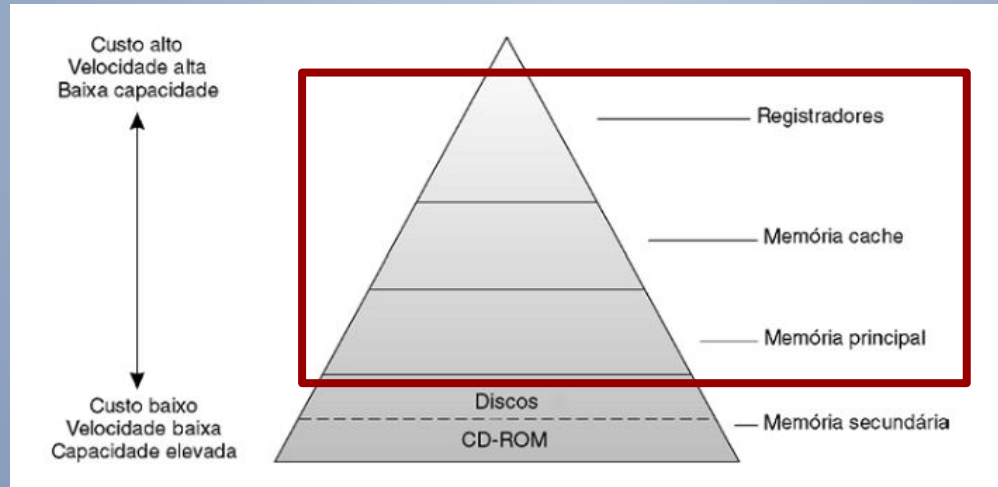
- não volátil e volátil, não apagável e apagável.

2

MEMÓRIA INTERNA

MEMÓRIAS INTERNAS

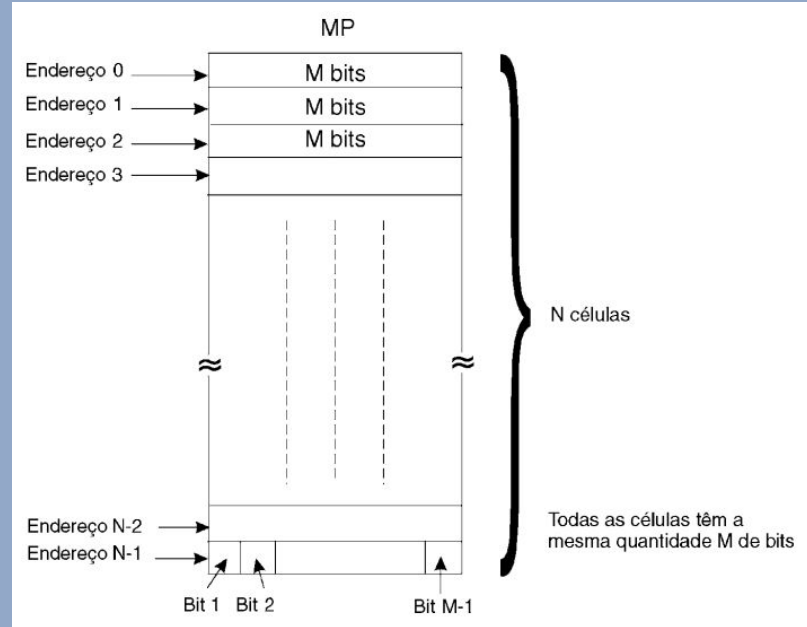
- Velocidade: rápidas.
- Preço: mais caras de todas.
- Tamanho: menores.





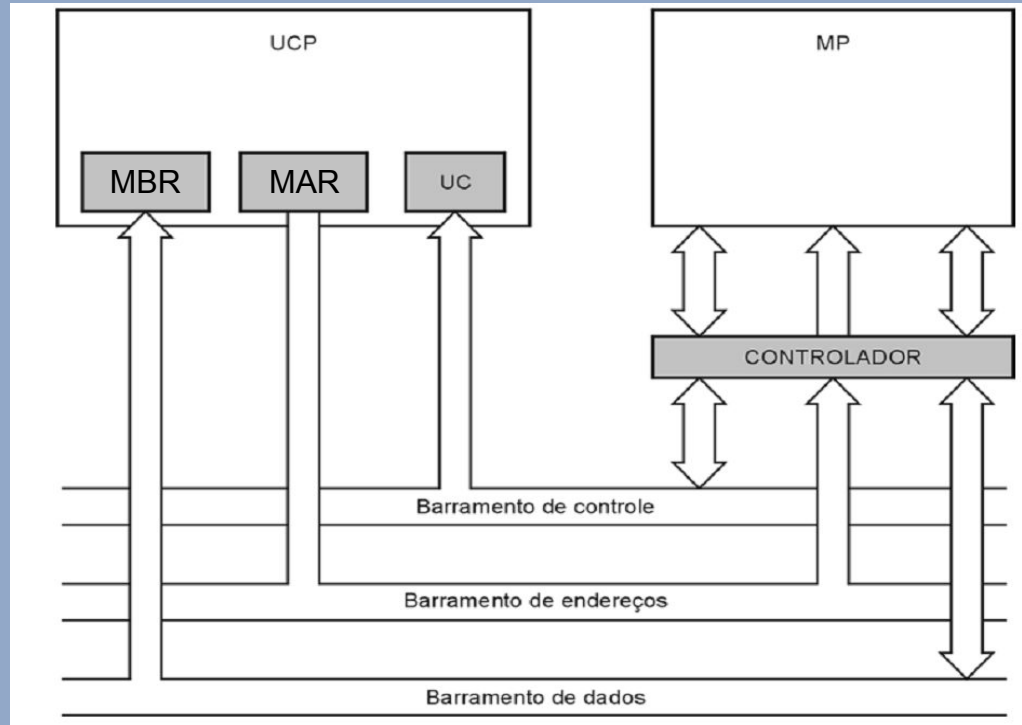
Memória principal

- Conceito do programa armazenado de Von Neumann;
- Memória básica de um sistema de computação;
- $M = 8$ bits.
- Leitura e Escrita.



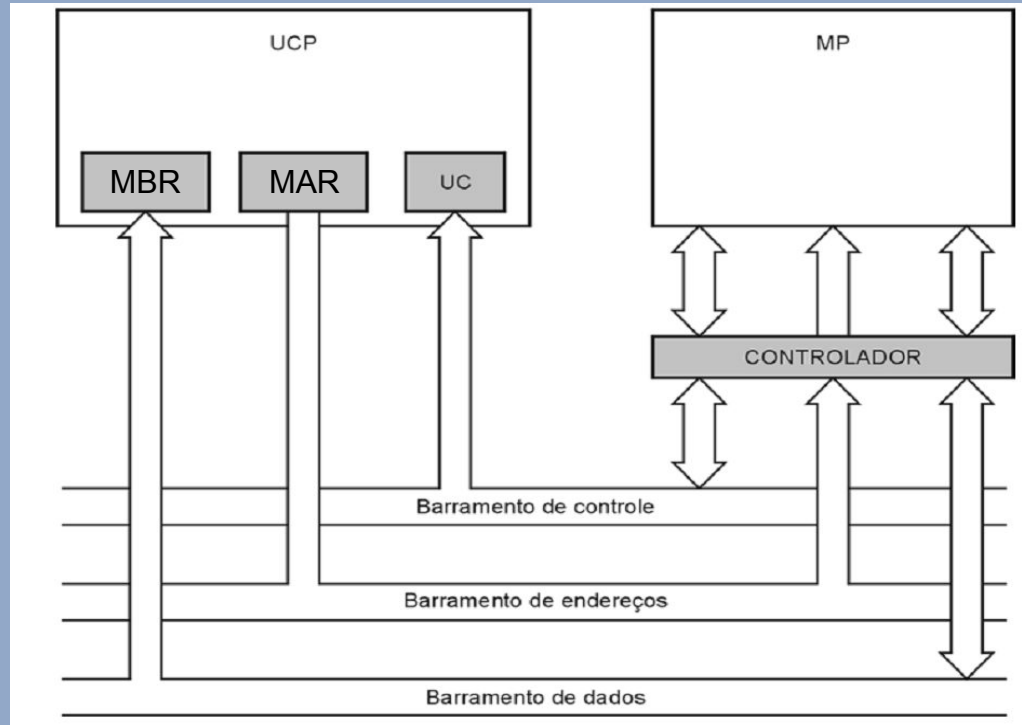


Memória principal e o processador

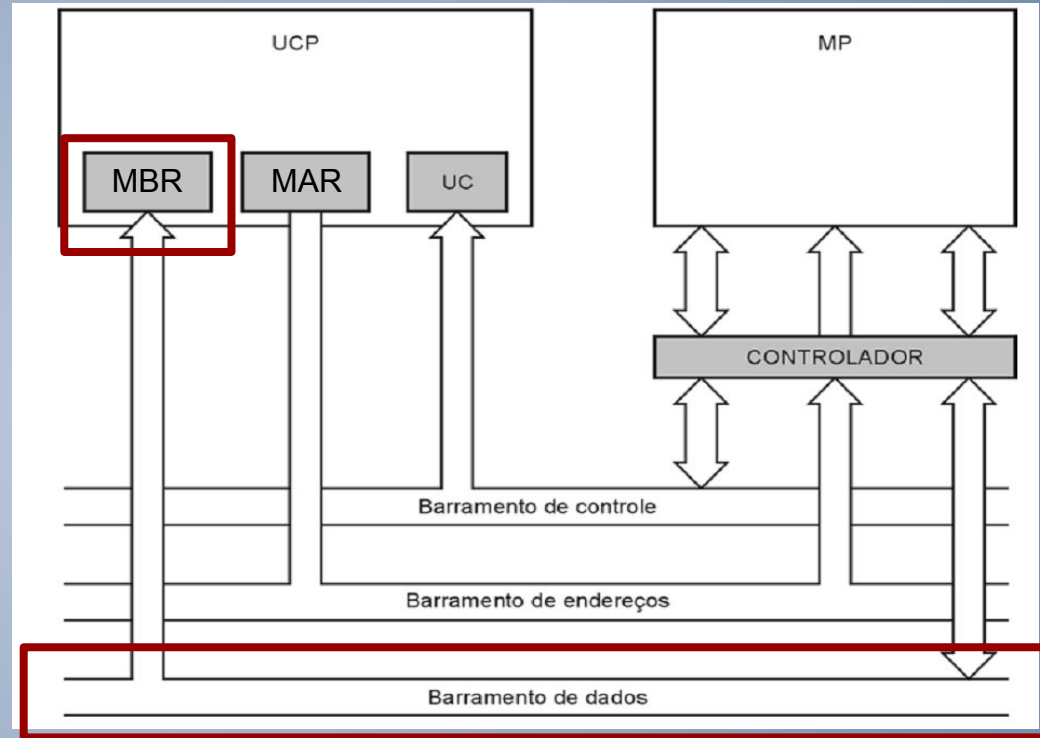




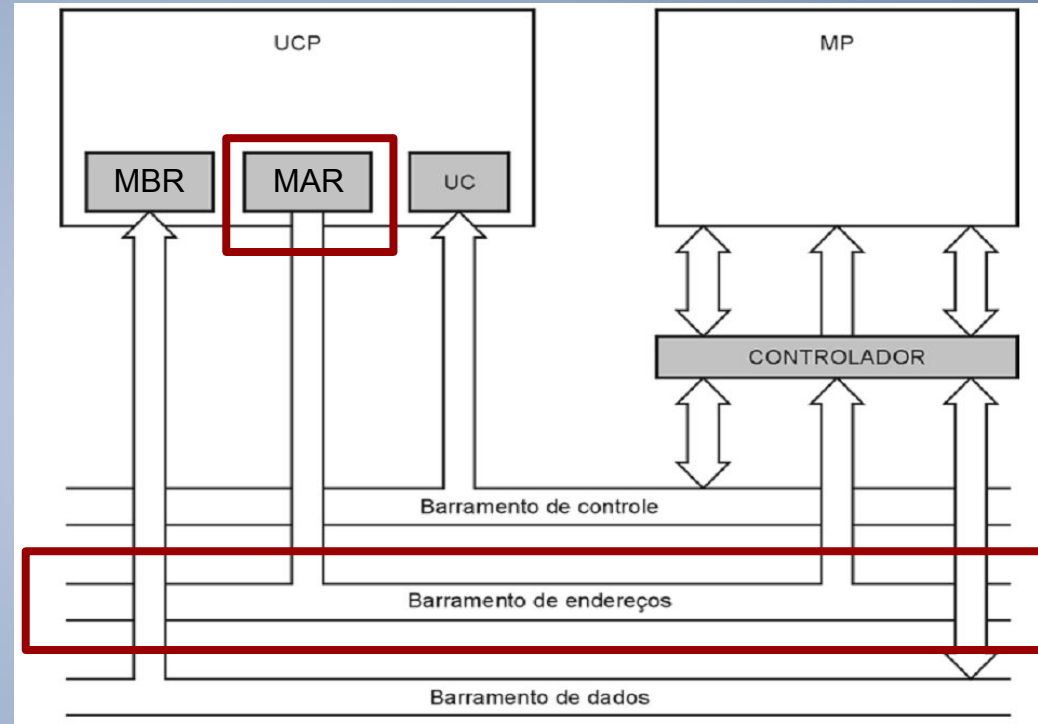
Memória principal e o processador



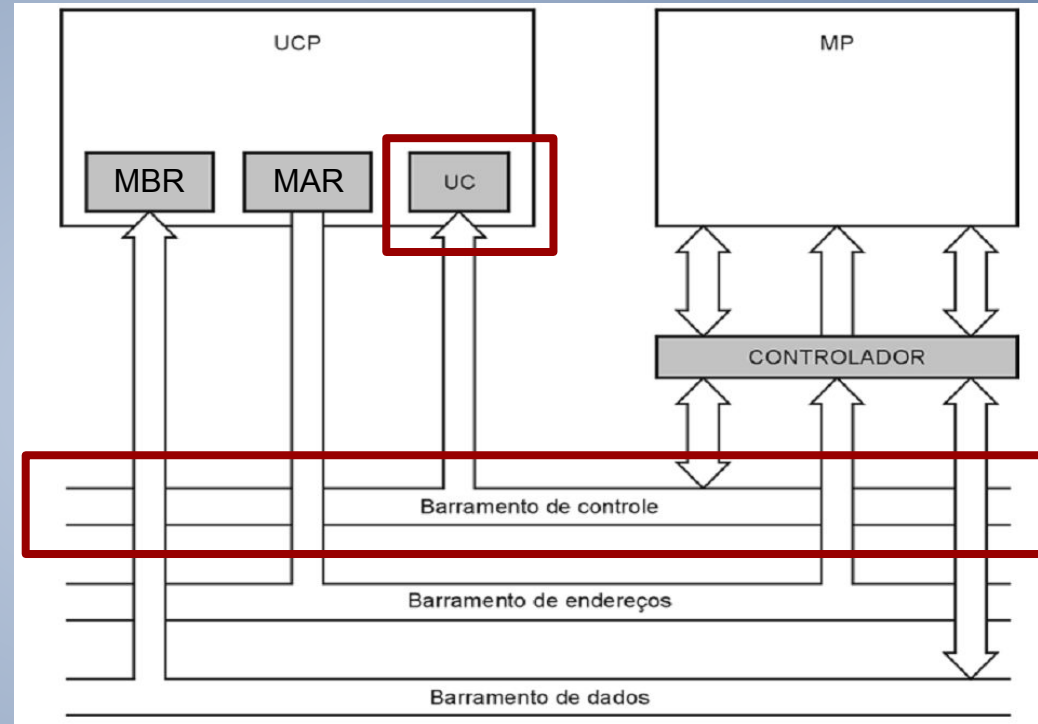
- Barramento bidirecional.
- MBR(Memory Buffer Register)
 - ▷ Mesma quantidade de bits que o barramento



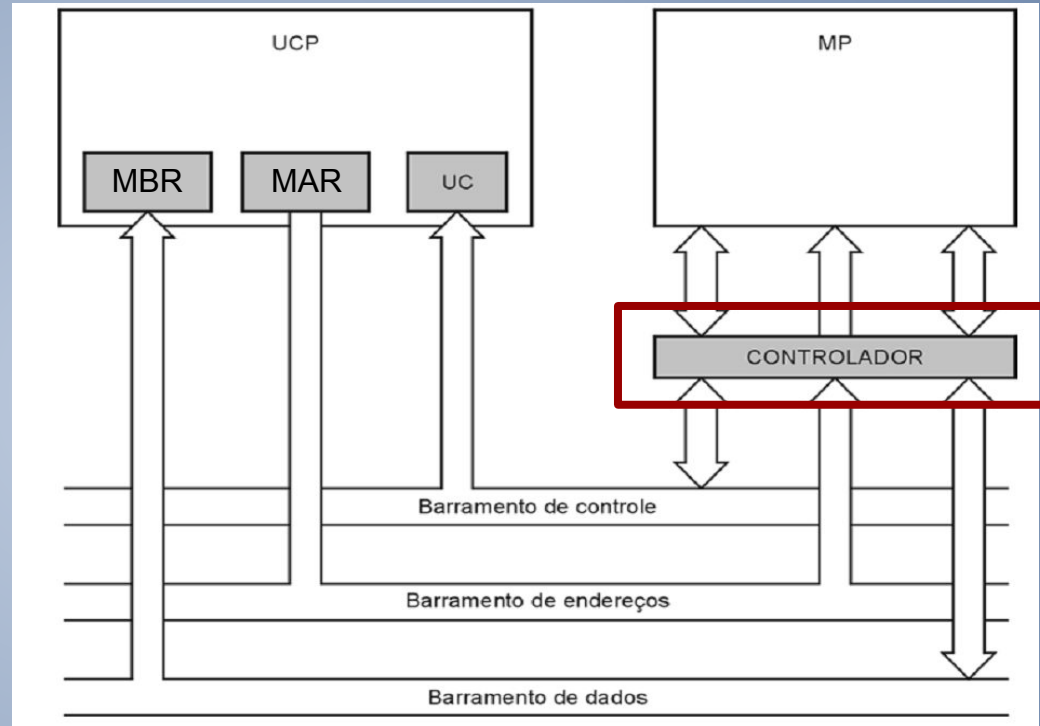
- Barramento unidirecional.
- MAR(Memory Address Register)
 - ▷ Mesma quantidade de bits que o barramento

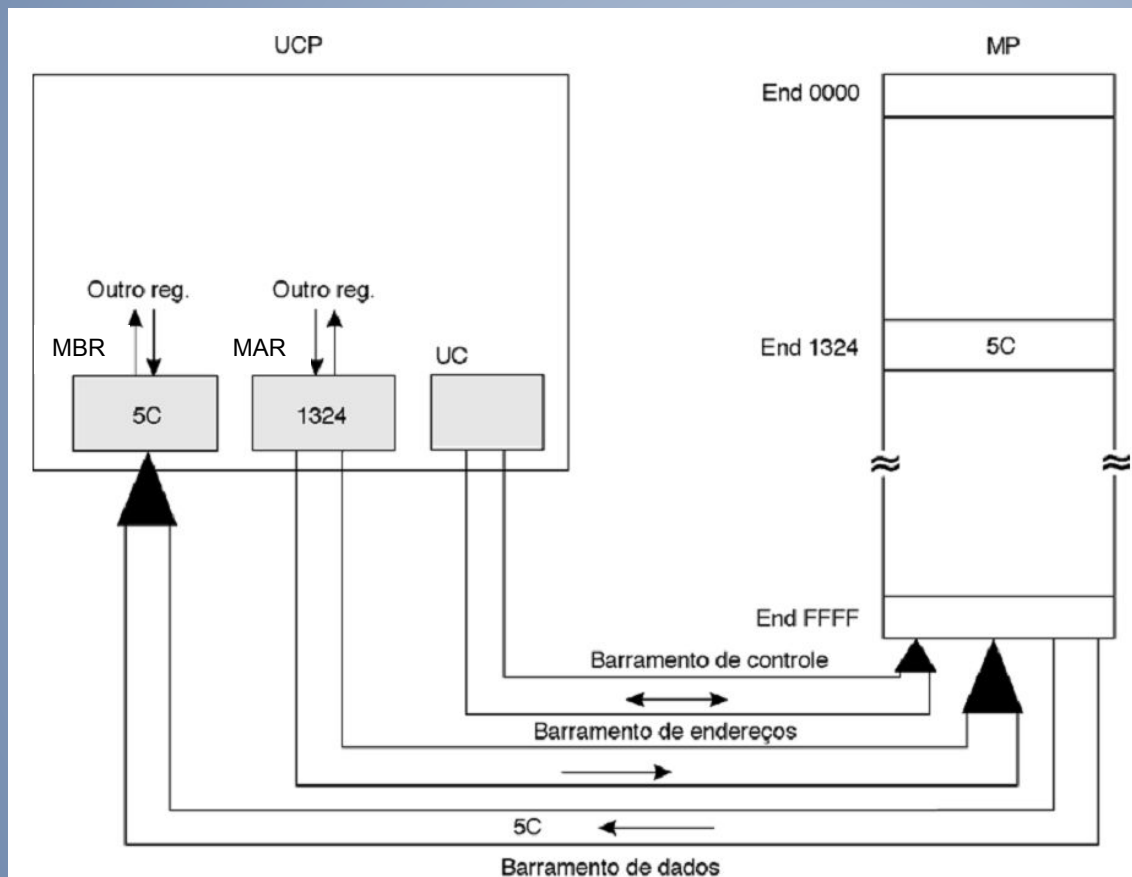


- Barramento bidirecional.
- UC (Unit Control)



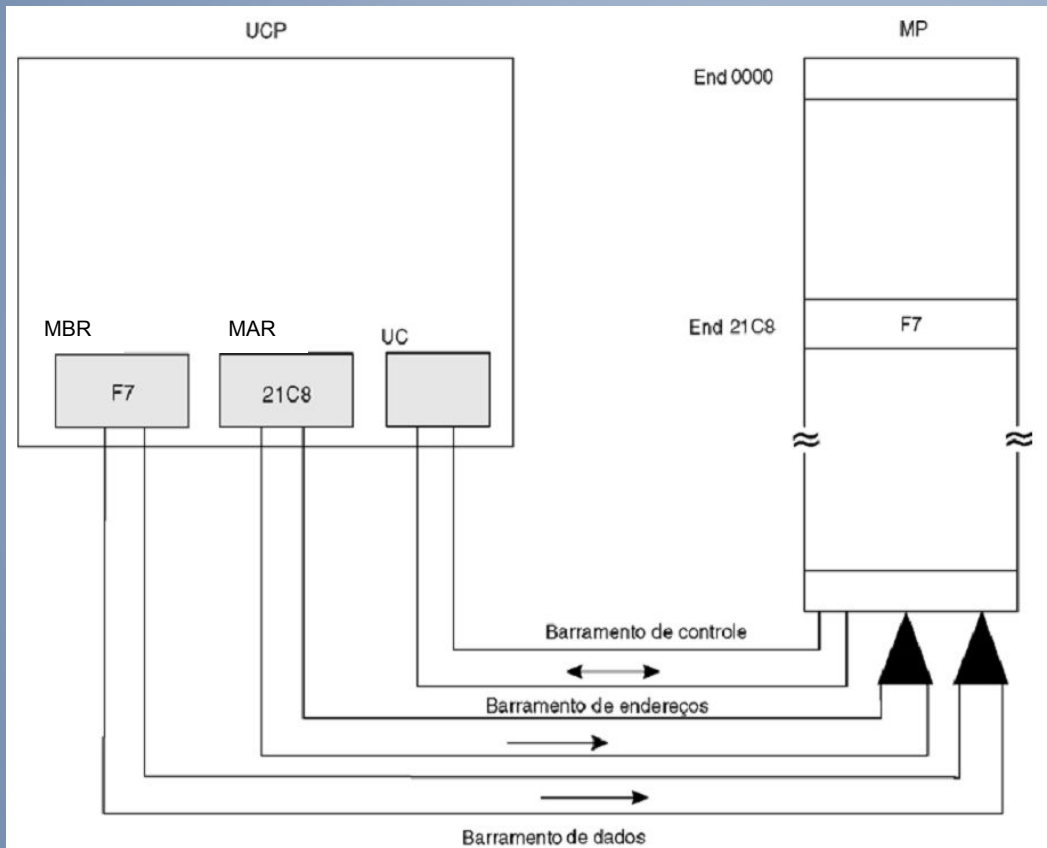
- Decodificar os sinais e realizar operações.





■ OPERAÇÃO LEITURA

1. $(MAR) \leftarrow (\text{outro registrador});$
2. O endereço é colocado no barramento de endereços;
3. Sinal de leitura pelo barramento de controle;
4. $(MBR) \leftarrow (MP(MAR)).$

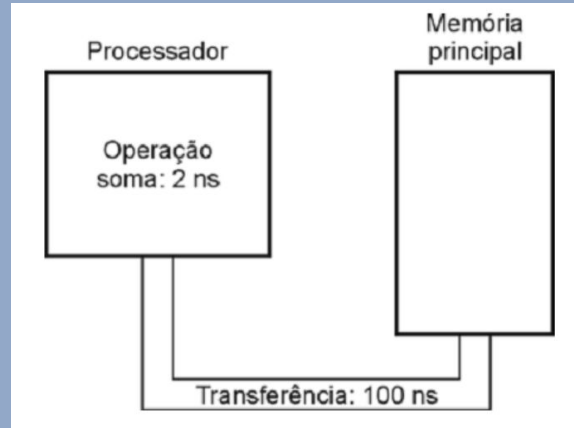


■ OPERAÇÃO ESCRITA



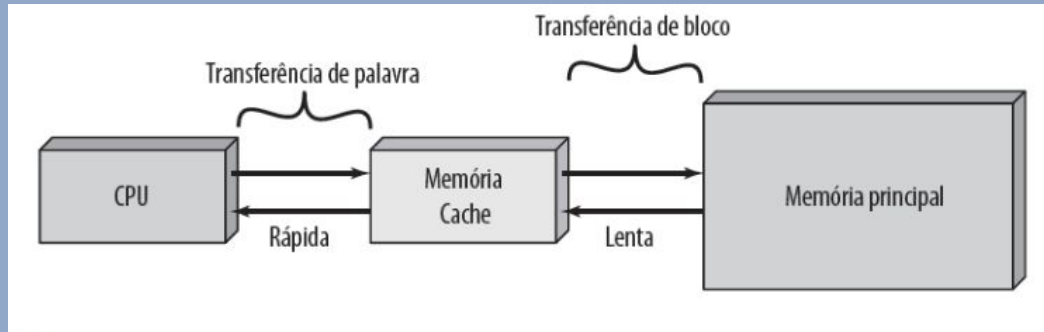
Gargalo

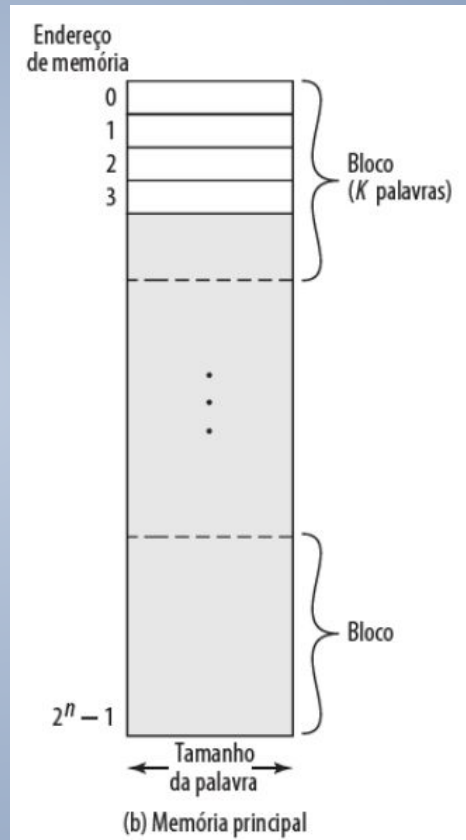
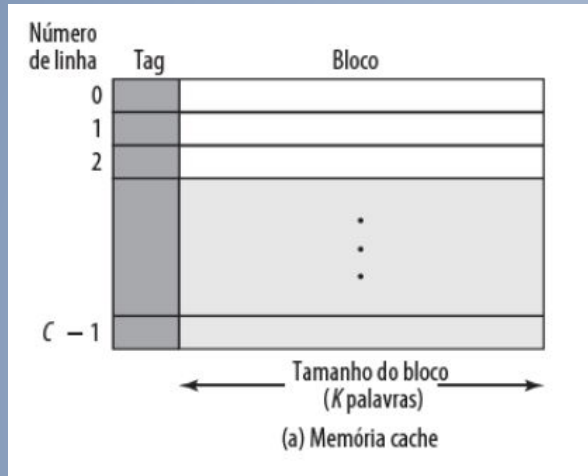
- A CPU busca por dados/instruções na memória principal.
- A memória responde a solicitação.
- Gargalo de Von Neumman: memória principal é lenta para a CPU.





Memória principal e memória cache

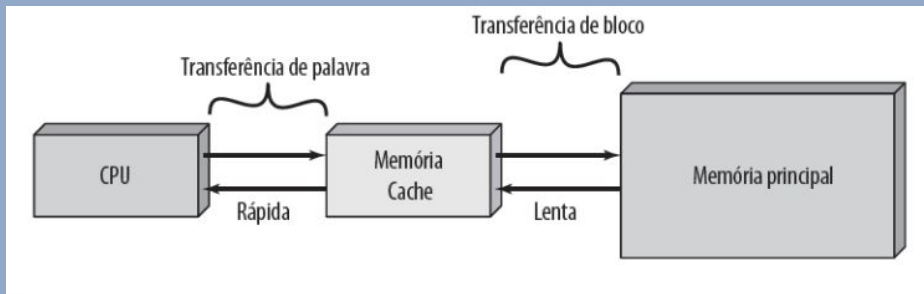




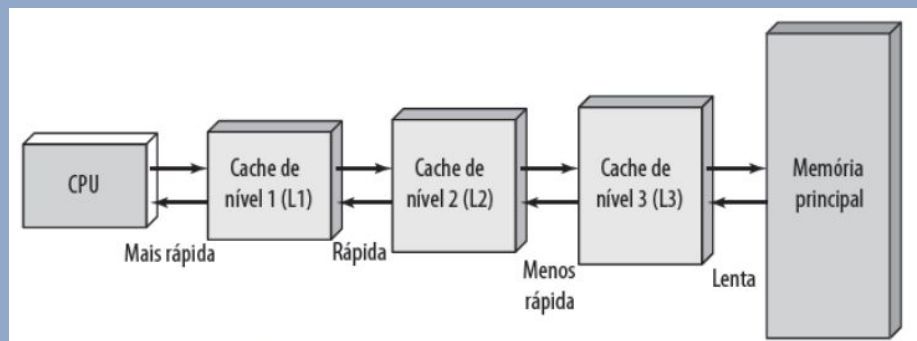


Organização

ÚNICA



VÁRIOS NÍVEIS





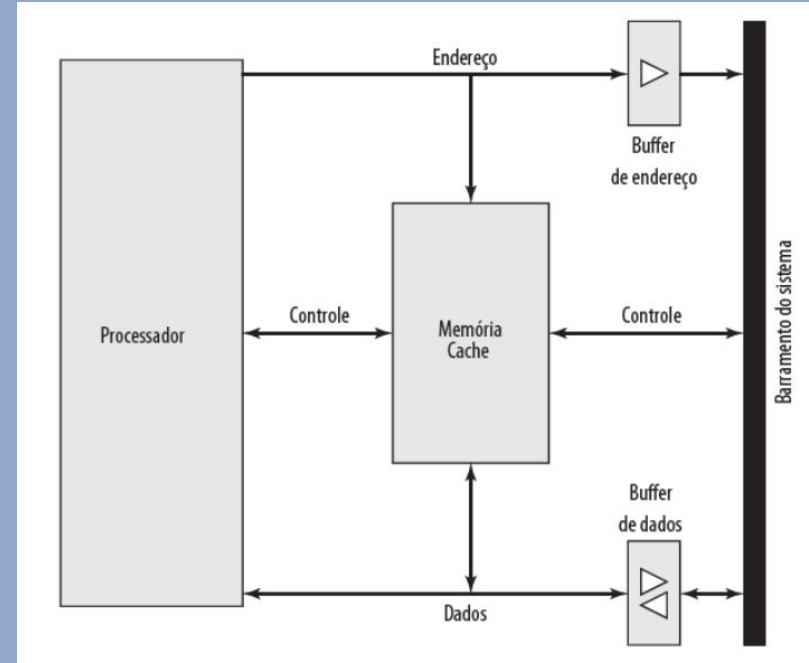
Cache hit e cache miss

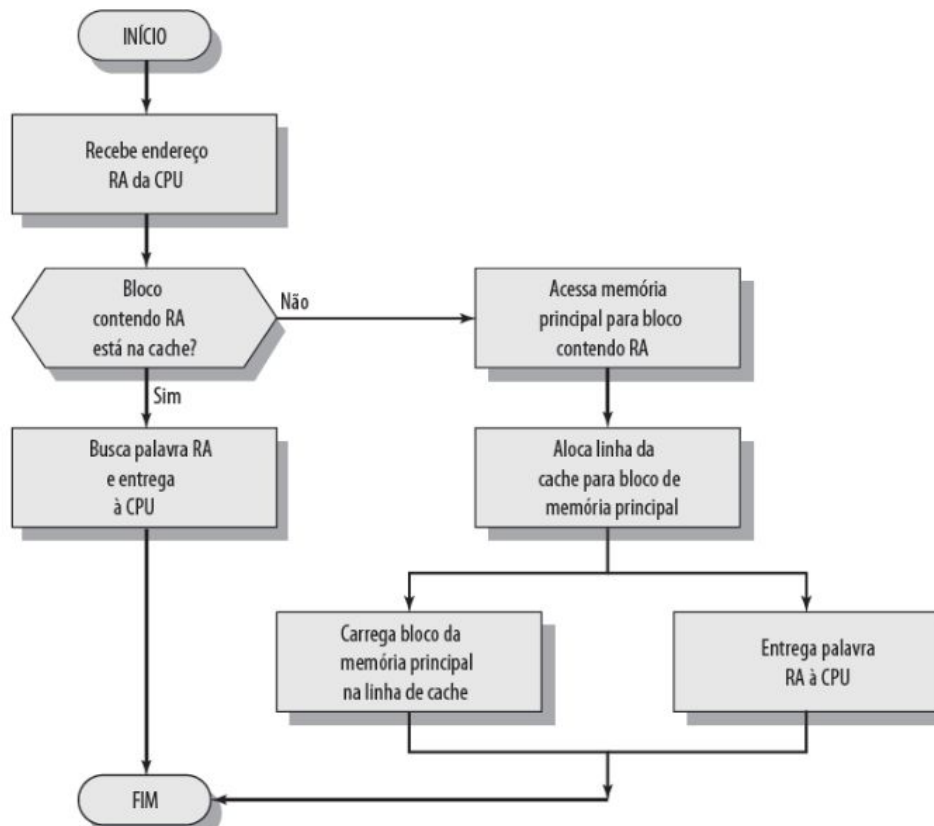
Cache hit

- ▷ As linhas de comunicação do processador com a MP são desativadas;

Cache miss

- ▷ As linhas de comunicação do processador com a MP são ativadas
- ▷ Copia o bloco da MP para a cache;







Exemplo cache hit

Endereço

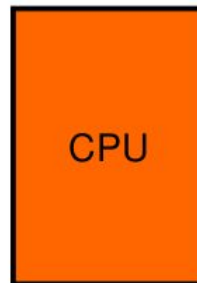
0	Palavra 0
4	Palavra 1
8	Palavra 2
12	Palavra 3
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7
32	Palavra 8
36	Palavra 9
40	Palavra 10
44	Palavra 11

Memória Cache

Endereço

0	Palavra 8
4	Palavra 9
8	Palavra 10
12	Palavra 11
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7

Exemplo 1



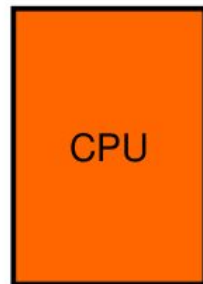
Memória Principal

Endereço	
0	Palavra 0
4	Palavra 1
8	Palavra 2
12	Palavra 3
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7
32	Palavra 8
36	Palavra 9
40	Palavra 10
44	Palavra 11

Memória Cache

Endereço	
0	Palavra 8
4	Palavra 9
8	Palavra 10
12	Palavra 11
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7

CPU busca uma
informação armazenada na
palavra 10 da memória
principal.



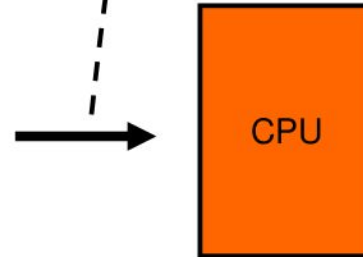
CPU

Como o tempo de resposta da cache é curto, a CPU pode concluir seu processamento rapidamente!

A Memória cache responde a solicitação feita pela CPU.

Endereço	
0	Palavra 0
4	Palavra 1
8	Palavra 2
12	Palavra 3
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7
32	Palavra 8
36	Palavra 9
40	Palavra 10
44	Palavra 11

Endereço	
0	Palavra 8
4	Palavra 9
8	Palavra 10
12	Palavra 11
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7



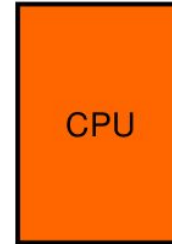


Exemplo cache miss

Endereço	
0	Palavra 0
4	Palavra 1
8	Palavra 2
12	Palavra 3
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7
32	Palavra 8
36	Palavra 9
40	Palavra 10
44	Palavra 11

Memória Cache	
Endereço	
0	Palavra 8
4	Palavra 9
8	Palavra 10
12	Palavra 11
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7

Exemplo 2

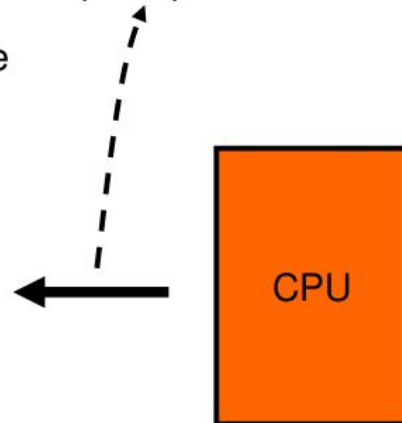


CPU

Memória Principal	
Endereço	
0	Palavra 0
4	Palavra 1
8	Palavra 2
12	Palavra 3
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7
32	Palavra 8
36	Palavra 9
40	Palavra 10
44	Palavra 11

Memória Cache	
Endereço	
0	Palavra 8
4	Palavra 9
8	Palavra 10
12	Palavra 11
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7

CPU busca uma
informação armazenada na
palavra 1 da memória
principal.



Memória Principal

Endereço

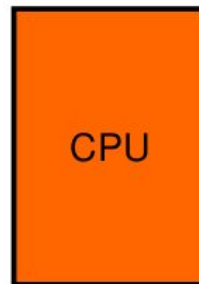
0	Palavra 0
4	Palavra 1
8	Palavra 2
12	Palavra 3
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7
32	Palavra 8
36	Palavra 9
40	Palavra 10
44	Palavra 11

Como a informação não está na cache, é gerado uma Falha de Cache.

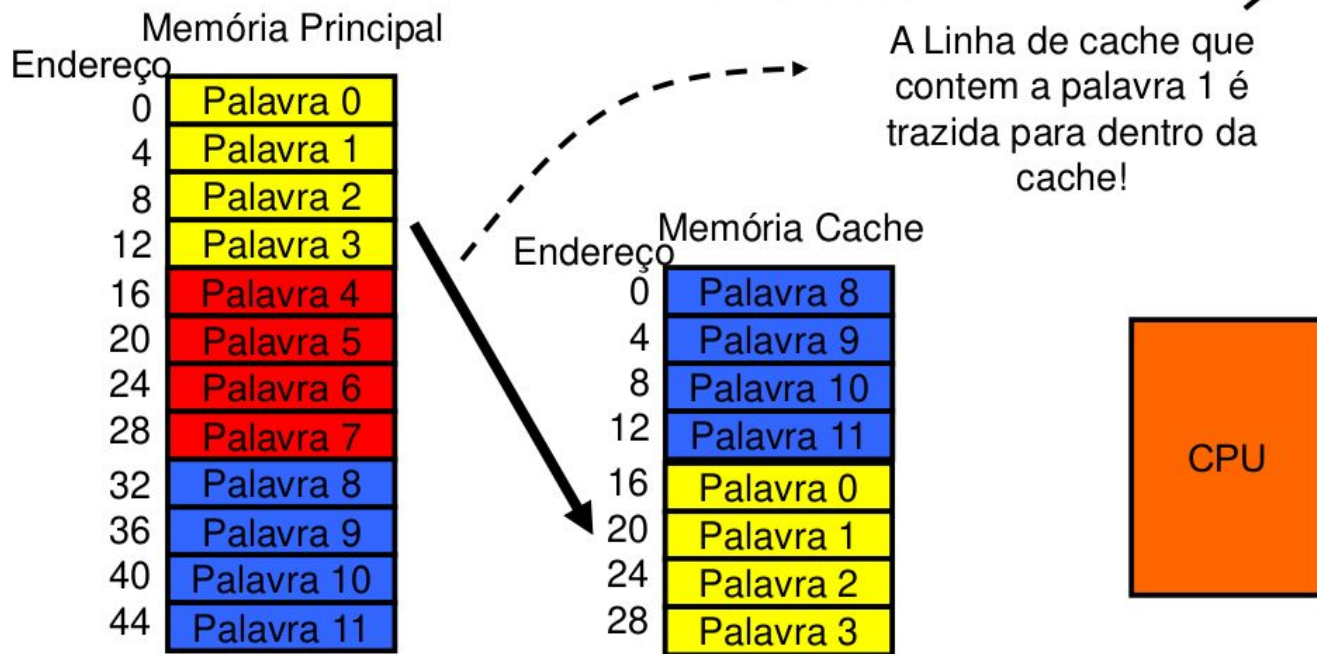
Memória Cache

Endereço

0	Palavra 8
4	Palavra 9
8	Palavra 10
12	Palavra 11
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7



Como a memória cache é pequena, as vezes, trazer uma linha de cache para dentro dela requer sobrescrever outra linha de cache previamente armazenada.



Memória Principal

Endereço

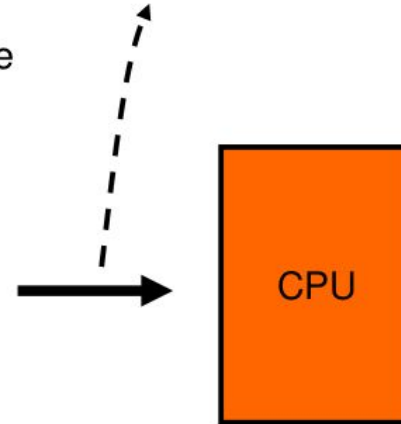
0	Palavra 0
4	Palavra 1
8	Palavra 2
12	Palavra 3
16	Palavra 4
20	Palavra 5
24	Palavra 6
28	Palavra 7
32	Palavra 8
36	Palavra 9
40	Palavra 10
44	Palavra 11

Memória cache responde a solicitação feita pela CPU.

Memória Cache

Endereço

0	Palavra 8
4	Palavra 9
8	Palavra 10
12	Palavra 11
16	Palavra 0
20	Palavra 1
24	Palavra 2
28	Palavra 3



3

ELEMENTOS DE PROJETO DE CACHE



Algoritmos de substituição

Definir qual dos blocos atualmente armazenados na cache deve ser retirado para dar lugar a um novo bloco que está sendo transferido;

- LRU - least recently used;
- FIFO - first-in-first-out;
- LFU - least frequently used;
- Escolha aleatória;

LRU - Least Recently Used

O sistema escolhe para ser substituído o bloco que está mais tempo sem ser utilizado;

FIFO - First-In-First-Out

O sistema escolhe o bloco que está há mais tempo na cache, independentemente de estar sendo usado ou não com frequência pelo processador;

LFU - Least Frequently Used

O sistema de controle escolhe o bloco que tem tido menos acessos por parte do processador (menos referências);

Escolha aleatória

Trata-se de escolher aleatoriamente um bloco para ser substituído, independentemente de sua situação no conjunto.



Políticas de escrita

Atualização de um dado na memória principal quando um dado na cache for substituído;

- Write-through;
- Write-back;

Write-through

Todo dado escrito na cache também é escrito na Principal;

- ▷ Vantagem: A memória cache sempre está válida;
- ▷ Desvantagem: Grande fluxo de informações;

Write-back

Todo dado é escrito apenas na cache. Se um dado for modificado, um bit associado, denominado de bit de modificação, é setado;

- ▷ Vantagem: Diminui o gargalo;
- ▷ Desvantagem: Complexidade. Entradas da MP podem ser inválidas e todos os acessos de E/S tem que ser feitos na cache;



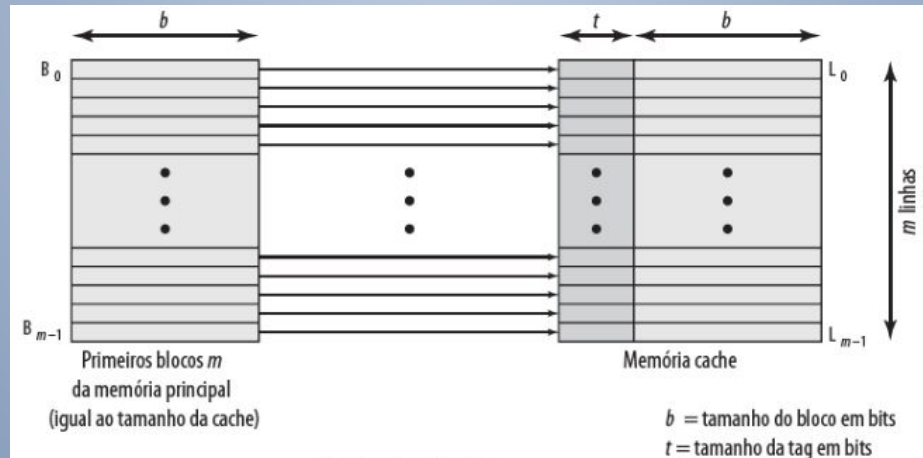
Função de mapeamento

Como o número de linhas de cache é menor que o número de blocos de memória, é necessário um algoritmo para mapear os blocos da memória principal em linhas da memória cache;

- Mapeamento direto;
- Mapeamento associativo.

Mapeamento direto

- Vantagem: Método simples de ser implementado;
- Desvantagem: Existe um local de cache fixo para cada bloco.
 - ▷ Referência de palavras repetidamente de dois blocos diferentes, mapeados para a mesma linha (thrashing).



Memória Principal:

64 posições de memória;

Cada posição 1 byte;

Bloco de 4 bytes;

Memória Cache: 4 linhas;

Formato do Endereço:

5	4	3	2	1	0
---	---	---	---	---	---

Memória Principal

00	00	00	
00	00	01	
00	00	10	
00	00	11	
00	01	00	
00	01	01	
00	01	10	
00	01	11	
00	10	00	
00	10	01	
00	10	10	
00	10	11	
00	11	00	
00	11	01	
00	11	10	
00	11	11	
01	00	00	
01	00	01	
01	00	10	
01	00	11	
01	01	00	
01	01	01	
01	01	10	
01	01	11	

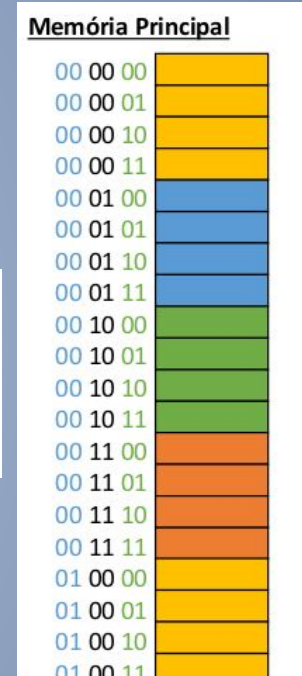
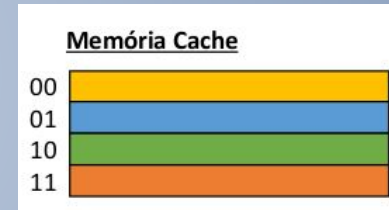
Memória Cache

00	
01	
10	
11	

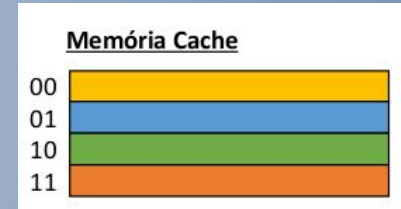
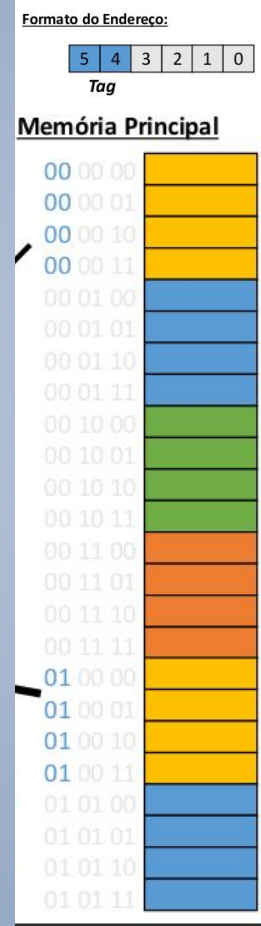
- Cada bloco da memória principal é mapeado em uma única linha de cache;

$$i = j \bmod m$$

- ▷ i = número da *linha* da memória cache;
- ▷ j = número do *bloco* da memória principal;
- ▷ m = número de *linhas* na memória cache;



Tag: Indica qual bloco dos possíveis está armazenado na linha de cache;



Linha: Indica em qual linha será armazenado aquele bloco de memória principal;

- Endereços com os bits 3 e 2 com 00 são mapeados na linhas 00
- Endereços com os bits 3 e 2 com 01 são mapeados na linha 01

Formato do Endereço:

5	4	3	2	1	0
---	---	---	---	---	---

Linha

Memória Principal

00	00	00	
00	00	01	
00	00	10	
00	00	11	
00	01	00	
00	01	01	
00	01	10	
00	01	11	
00	10	00	
00	10	01	
00	10	10	
00	10	11	
00	11	00	
00	11	01	
00	11	10	
00	11	11	
01	00	00	
01	00	01	
01	00	10	
01	00	11	
01	01	00	
01	01	01	
01	01	10	
01	01	11	

Memória Cache

00	
01	
10	
11	

Palavra: Indica qual das possíveis palavras deve ser acessada no respectivo bloco;

Formato do Endereço:

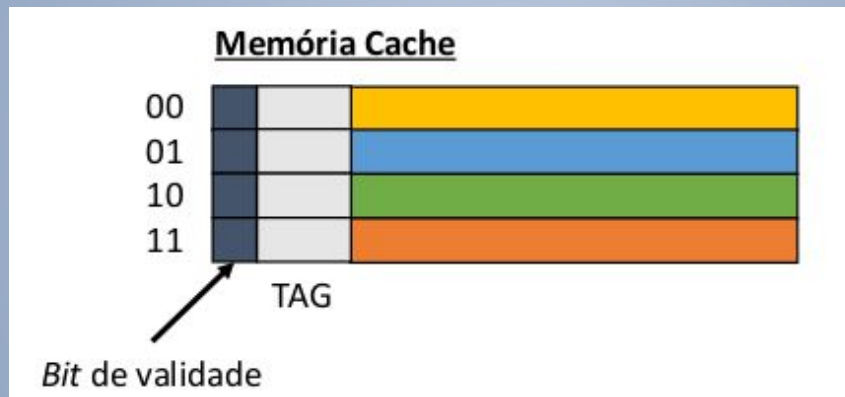
5	4	3	2	1	0
---	---	---	---	---	---

Palavra

Memória Principal

00 00 00	
00 00 01	
00 00 10	
00 00 11	
00 01 00	
00 01 01	
00 01 10	
00 01 11	
00 10 00	
00 10 01	
00 10 10	
00 10 11	
00 11 00	
00 11 01	
00 11 10	
00 11 11	
01 00 00	
01 00 01	
01 00 10	
01 00 11	
01 01 00	
01 01 01	
01 01 10	
01 01 11	

- Bit validade: informação da MP ou lixo.



1. Apresentar um endereço à memória cache;
2. Utilizar os bits do campo LINHA para determinar qual linha deve ser buscada na cache;
3. Verificar o bit de validade;
4. Verificar se a TAG do endereço corresponde à TAG da linha da cache;

Cache hit

5. Se sim: Utilizar o campo PALAVRA para determinar qual palavra do bloco deve ser passada ao processador;

1. Apresentar um endereço à memória cache;
2. Utilizar os bits do campo LINHA para determinar qual linha deve ser buscada na cache;
3. Verificar o bit de validade;
4. Verificar se a TAG do endereço corresponde à TAG da linha da cache;

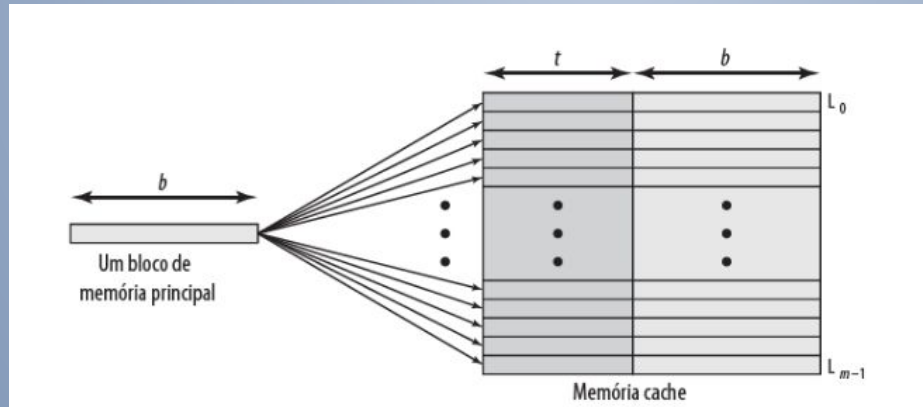
Cache miss

5. Se não: Apresentar o Endereço à MP;
6. Copiar o bloco requisitado (campo TAG+LINHA) para memória cache (na linha indicado pelo campo LINHA);
7. Escrever o campo TAG na linha de cache;

Mapeamento associativo

- Permite que cada bloco da memória principal seja carregado em qualquer linha da cache;
- O campo tag identifica o bloco da memória principal;
- O campo Palavra indica qual palavra do bloco vai ser buscada;
- Para determinar se um bloco está na cache, a lógica de controle da cache precisa comparar simultaneamente a tag de cada linha;
- Flexibilidade em relação a qual bloco substituir quando um novo bloco for lido para a cache.

- Vantagem: Resolve o problema thrashing presente no mapeamento direto;
- Desvantagem: Complexidade do circuito necessário para comparar as tags de todas as linhas da cache em paralelo;
 - ▷ Busca na memória cache mais lenta;



1. O endereço é apresentado à memória cache;
2. O valor da TAG é comparado com todas as TAG das linhas de cache;
3. Verifica-se se existe alguma TAG do endereço igual a TAG da linha de cache;

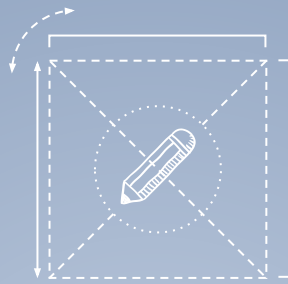
Cache hit

4. Sim: Cache Hit, usa-se o endereço de Byte para passar o dado ao processador;

Cache miss

5. Não: Cache Miss, o dado é buscado na MP e substituído na cache.

EXERCÍCIOS



1. Nos sistemas de computação existem memórias como RAM, ROM, DRAM, PROM, EPROM, EEPROM. Descreva diante de suas características, medidas de desempenho e aplicações.
2. A hierarquia de memória tem relação com três variáveis (custo, velocidade, capacidade). Suponha que a relação entre estas é linear e esboce gráficos para mostrar o comportamento.
3. Discutir sobre a hierarquia de memória. É coerente dizer que existe uma memória com melhor aplicação que as demais?
4. Defina o conceito de trashing na computação e suas implicações.



DÚVIDAS?

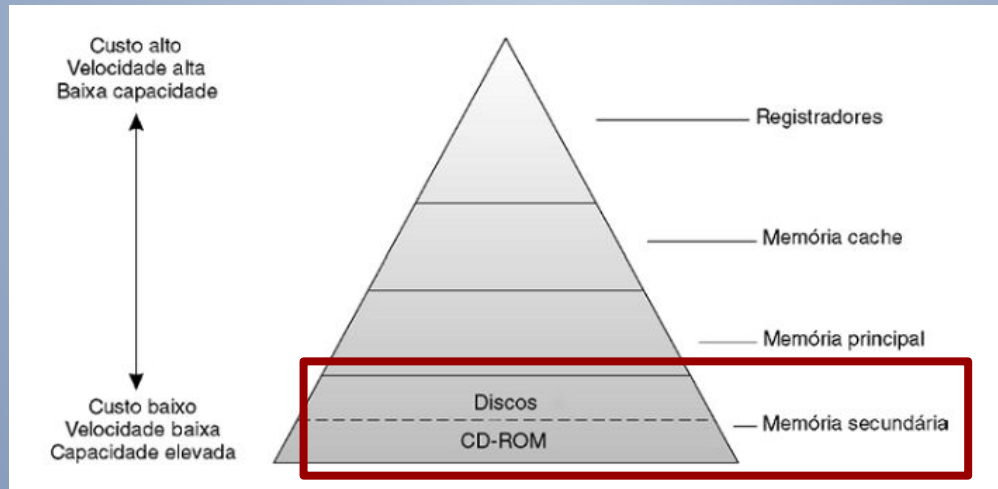
sara.negreiros@ufersa.edu.br

4

MEMÓRIA EXTERNA

MEMÓRIAS EXTERNAS

- Velocidade: lentas.
- Preço: mais baratas de todas.
- Tamanho: maiores.

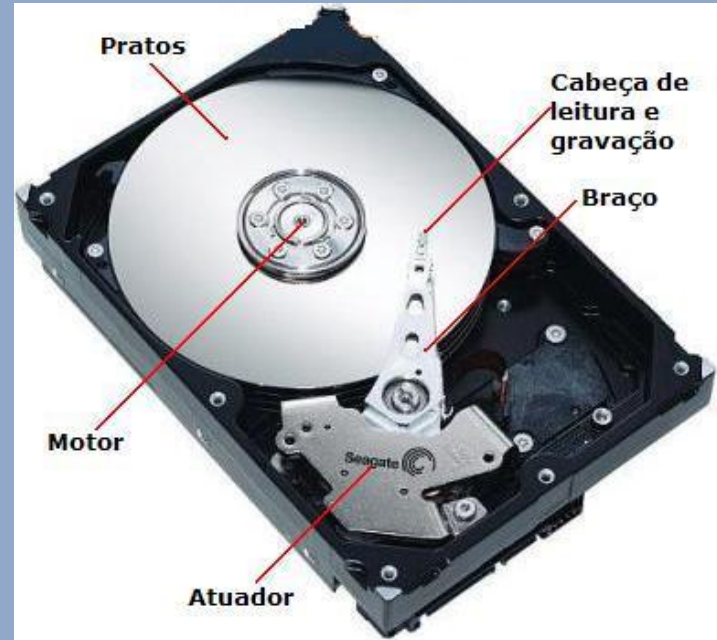


- Armazenamento permanente
 - ▷ **Disco magnético;**
 - ▷ Memórias óticas;
 - ▷ Fitas;
 - ▷ Unidades de estado sólido;

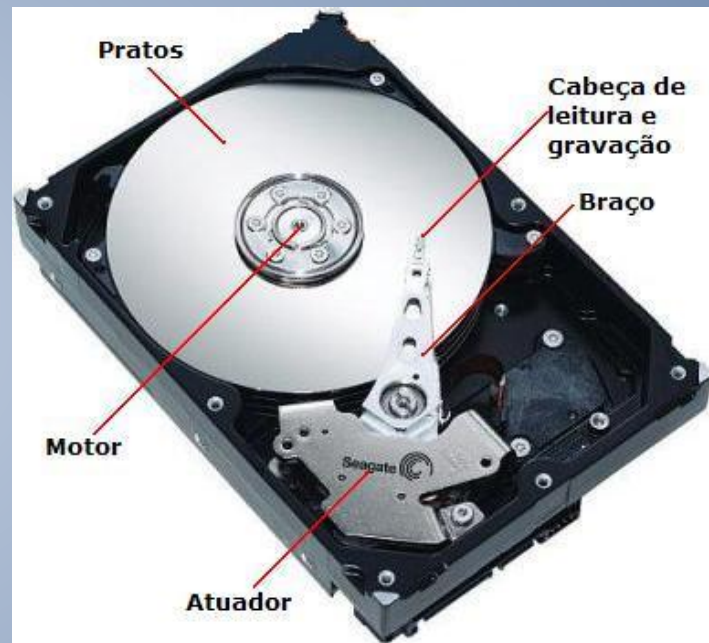


Disco magnético

- Forma de armazenamento secundário mais comum nos sistemas de computação;
- Prato circular construída de material não magnético, chamado de substrato, coberto com um material magnetizável;

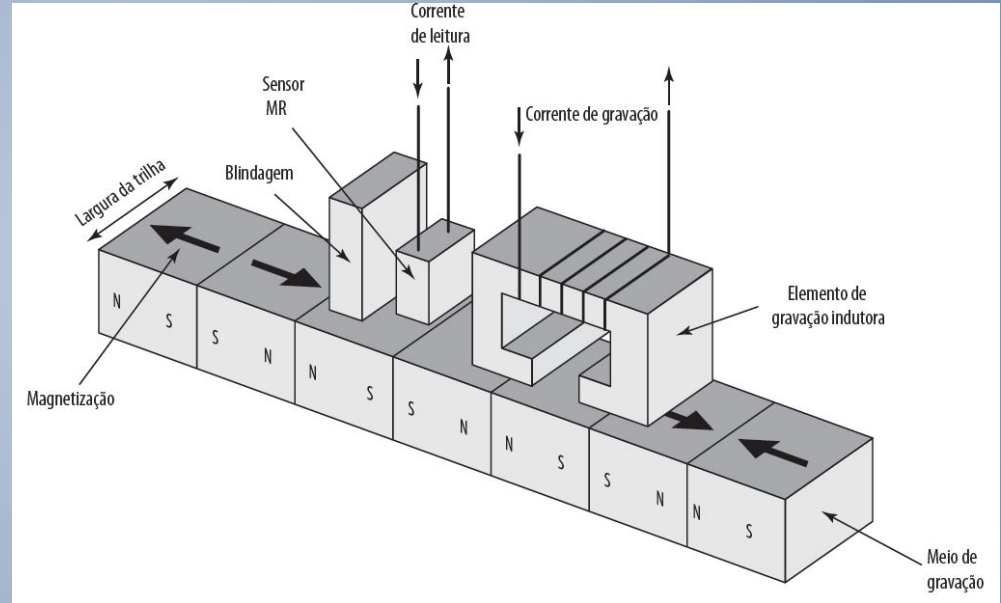


- Os dados são gravados/recuperados do disco por meio de uma bobina condutora, chamada cabeçote;
- Em muitos sistemas, existem dois cabeçotes, uma de leitura e uma de gravação;



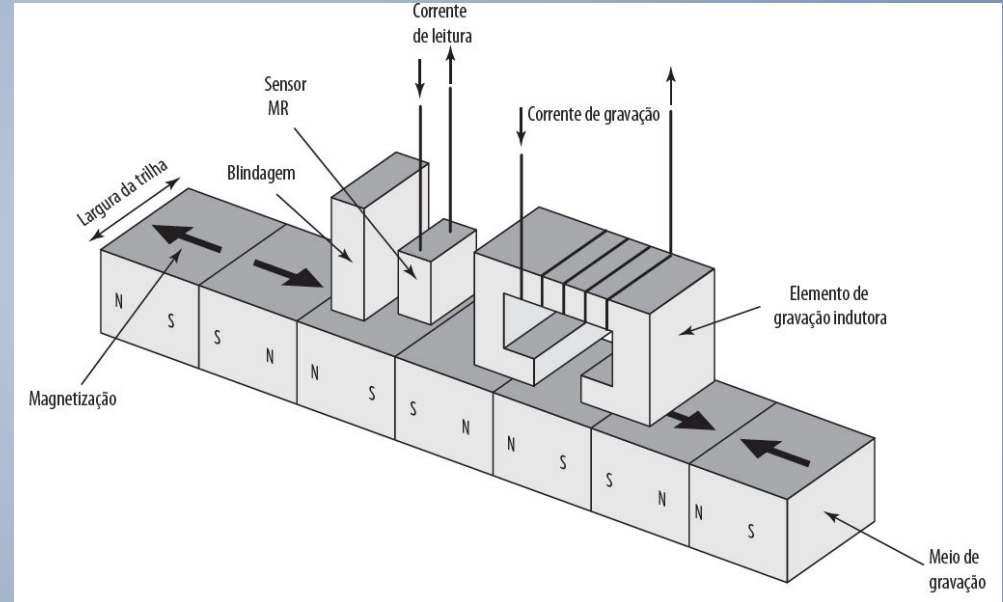
GRAVAÇÃO

- A eletricidade que flui pela bobina produz um campo magnético;
- Os pulsos elétricos são enviados à cabeça de gravação, e os padrões magnéticos são gravados na superfície do disco;



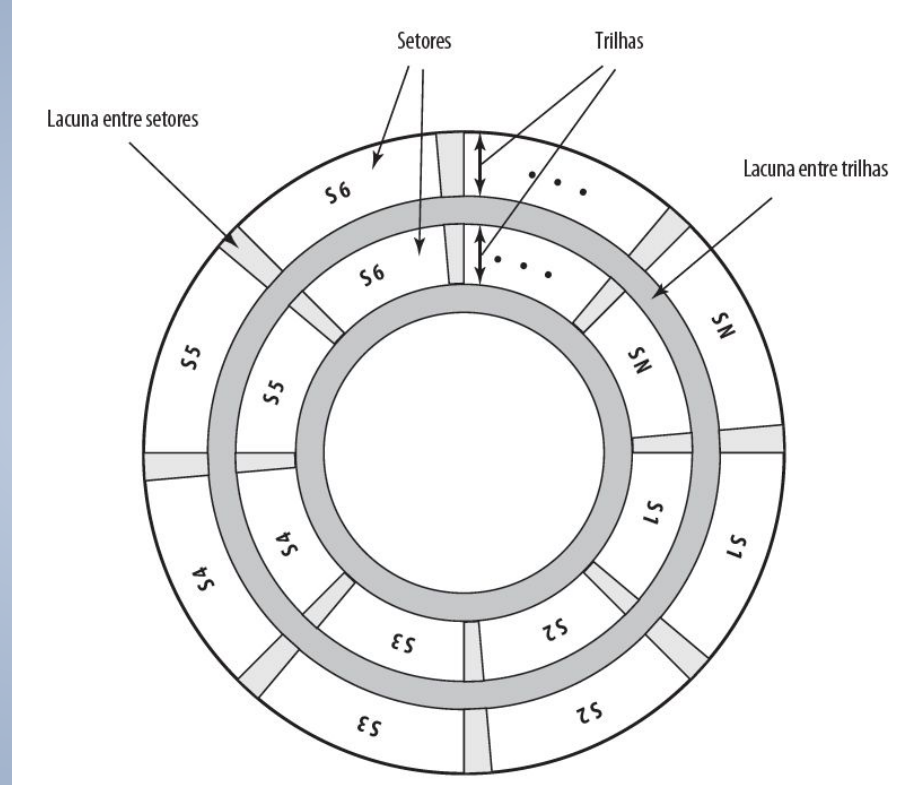
LEITURA

- Um campo magnético movendo-se em relação a uma bobina produz uma corrente elétrica na bobina;
- Quando a superfície do disco passa sob a cabeça, ela gera uma corrente com a mesma polaridade daquela já gravada.



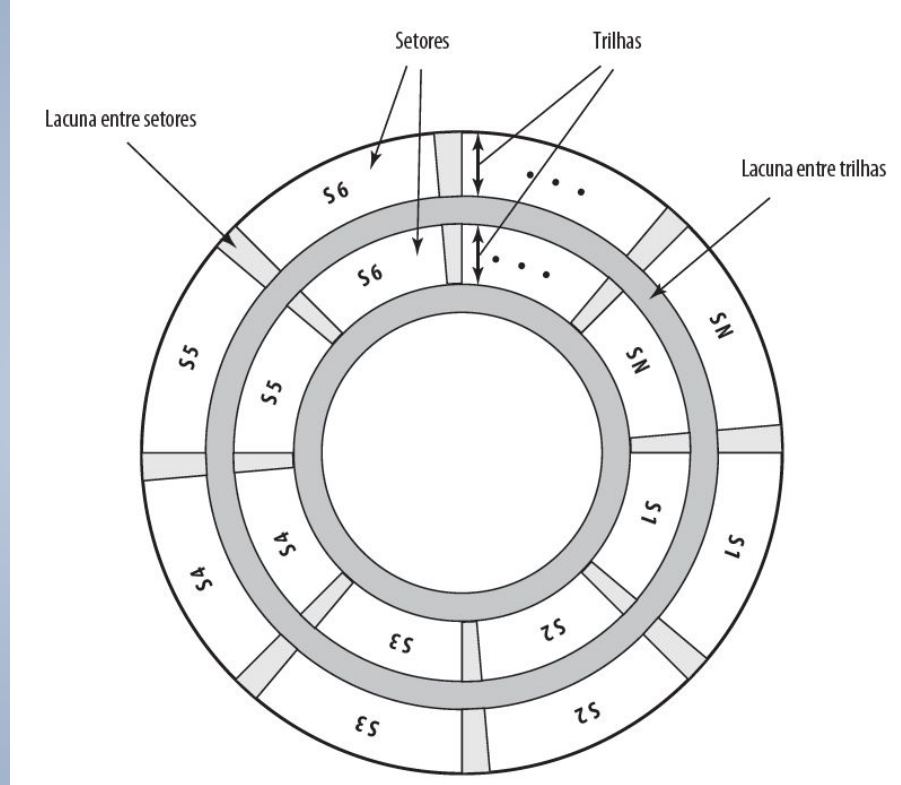
ORGANIZAÇÃO

- Organização dos dados no prato em um conjunto concêntrico de anéis, chamados de trilhas;
- Trilhas adjacentes são separadas por lacunas;



TRANSFERÊNCIA DE DADOS

- Os dados são transferidos do/para o disco em blocos;
- Normalmente um bloco tem tamanho menor do que a trilha;



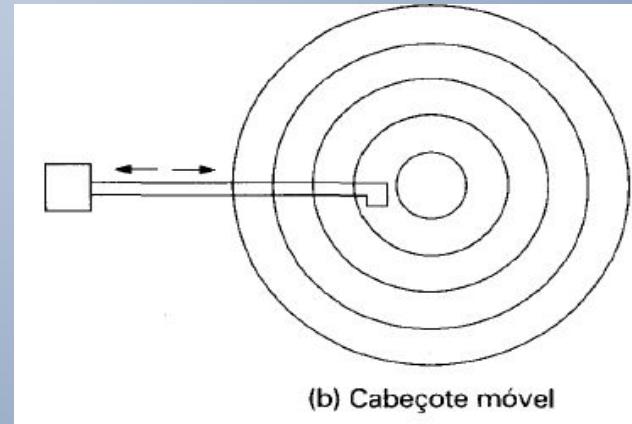
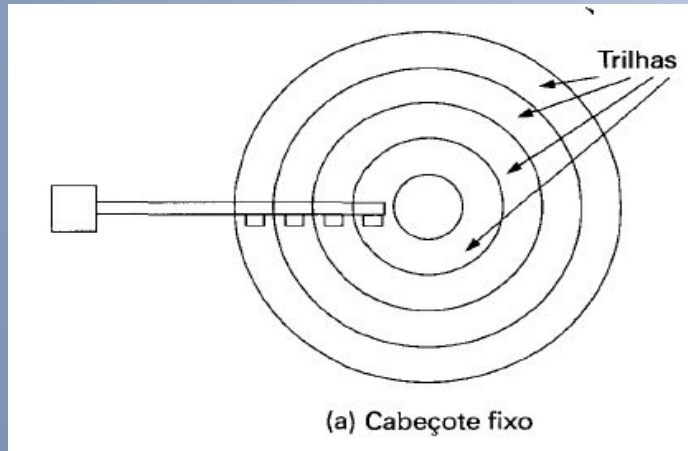


Características físicas

- Movimento do cabeçote;
- Transportabilidade do disco;
- Quantidade de Lados;
- Quantidade de Pratos;

MOVIMENTO DO CABEÇOTE

- Cabeçote Fixo: Um cabeçote de leitura e escrita para cada trilha;
- Cabeçote Móvel: Somente um cabeçote de leitura-escrita. Os cabeçotes são montados em um braço móvel;



TRANSPORTABILIDADE DO DISCO

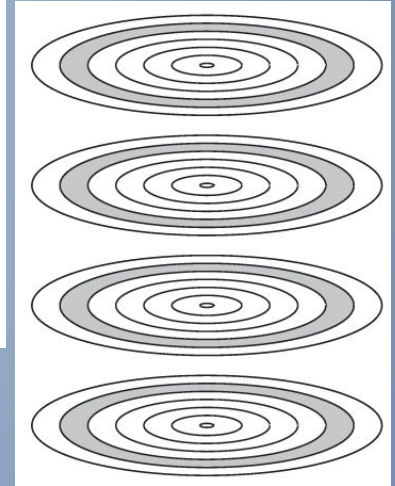
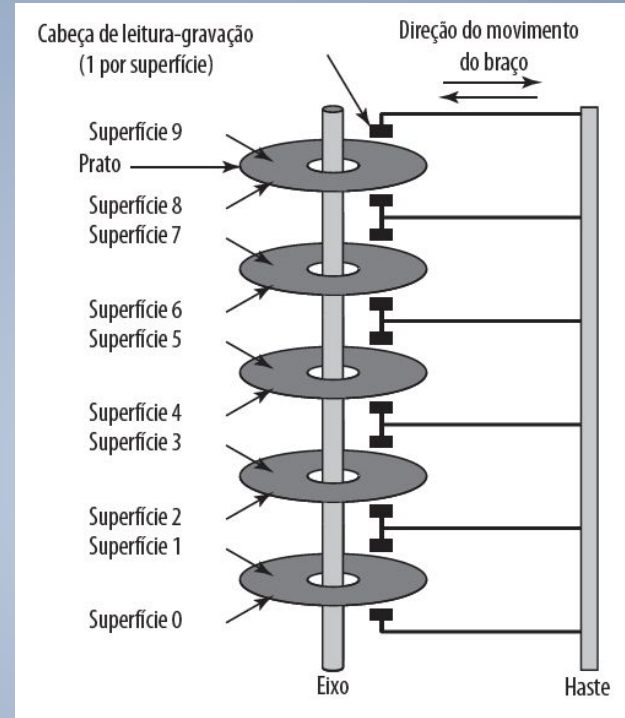
- Disco não removível: O disco é montado permanentemente na unidade de disco (disco rígido).
- Disco removível: O disco pode ser removido e substituído por outro disco (disquetes).

QUANTIDADE DE LADOS

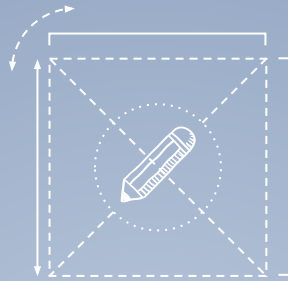
- Lado único.
- Lado duplo.

QUANTIDADE DE PRATOS

- Placa única.
- Múltiplas placas.
 - ▷ Múltiplos braços.
 - ▷ Cilindro: conjunto de trilhas na mesma posição.



EXERCÍCIOS



1. Pesquise e descreva o funcionamento das seguintes memórias externas:
 - a. Memória óptica.
 - b. Fita magnética.
 - c. Unidades de estado sólido.
2. Descreva os principais componentes mecânicos de discos magnéticos e qual a função de cada um.



DÚVIDAS?

sara.negreiros@ufersa.edu.br