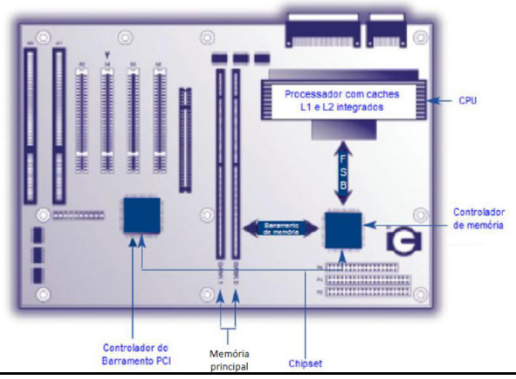
Memória Cache

- CPU mais rápida do que as Memórias
- CPU
  - Mais circuitos nos chips
  - Paralelismo (pipeline)
  - Operações superescalares.
- Memória:
  - Maior capacidade nos chips e não > velocidade

Cache

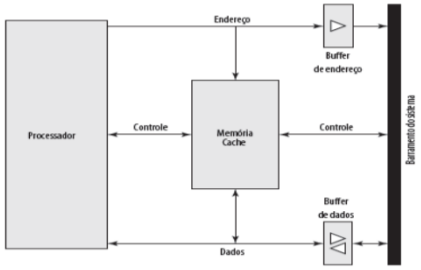
- Os processadores aumentaram suas frequências;
- O acesso a memória tornou-se um gargalo;
- Criaram-se as memórias cache.



Cache

- O processador é bem mais rápido do que a memória DRAM.
- Memória Estática, SRAM de alto desempenho incluída no chip do processador.

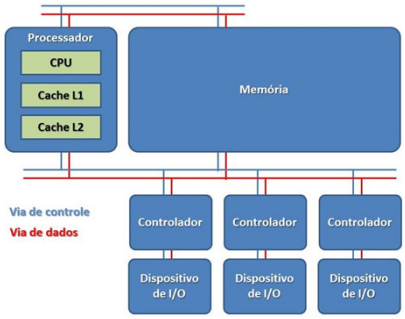
Organização típica da memória cache



Cache - Características

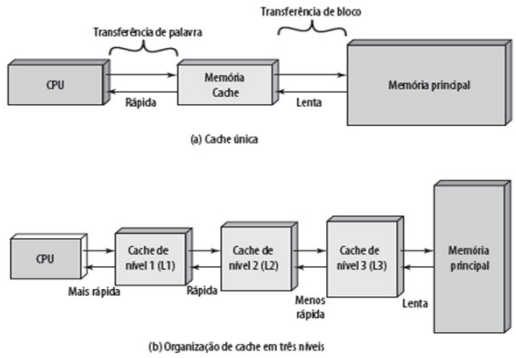
- Rápidas
- Pequena quantidade;
- Custo elevado
- Não contam com o mesmo nível de miniaturização das DRAM.
- Entre a memória principal e a CPU.

Cache



Função da Cache

- Servir como intermediária na leitura e escrita de dados na RAM.
- Acerto da Cache: de 80 a 99%

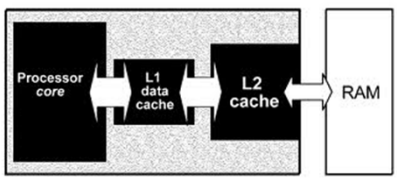


Controlador de Cache

- Quando o processador precisa ler dados na memória RAM, um circuito especial, chamado de controlador de Cache, transfere os dados mais requisitados da RAM para a cache.
- Controlador da Cache: copia os dados que acha que o processador necessitará.

Cache

- Com o uso da memória cache, na maior parte do tempo, o processador encontra nela os dados que precisa.



MEMÓRIA ESTÁTICA - SRAM

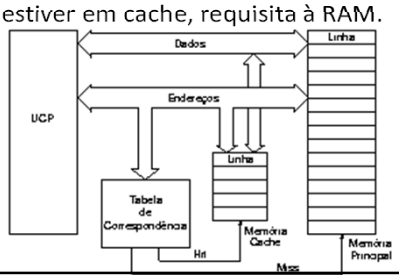
- Não utiliza capacitores
- Utiliza circuítos digitais chamados flip-flop para armazenar os bits.
- Não há necessidade de refresh
- É mais cara
- Exige um espaço maior
- É mais rápida

Memória Cache

- É utilizada para aumentar o desempenho da máquina;
- O controlador de cache tentará entregar a instrução ou dado, ao processador antes que esse procure na RAM que é relativamente lenta;

Operação

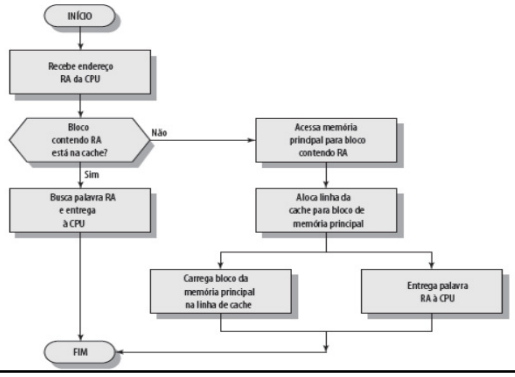
- ULA requisita uma instrução ou dado.
- Cache verifica se possui o dado.
- Se estiver em cache, operação é executada
- Se não estiver em cache, requisita à RAM.

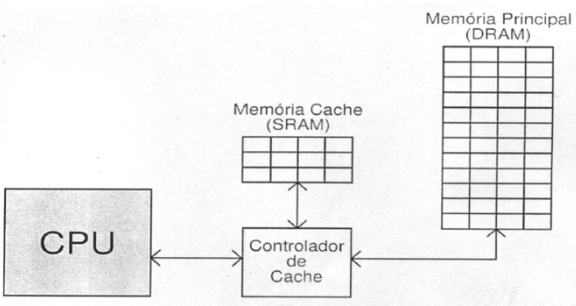


Operação da cache – visão geral

- CPU requisita conteúdo do local de memória.
- Verifica se os dados estão em cache.
- Se estiverem, apanha da cache (rápido).
- Se não, lê bloco solicitado da memória principal para a cache.
- Depois, entrega da cache à CPU.
- Cache inclui tags para identificar qual bloco da memória principal está em cada slot da cache.

Operação de leitura de cache – fluxograma



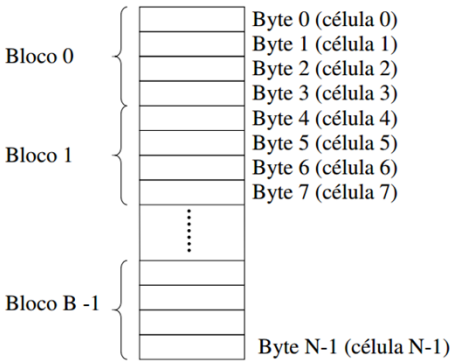


Tamanho

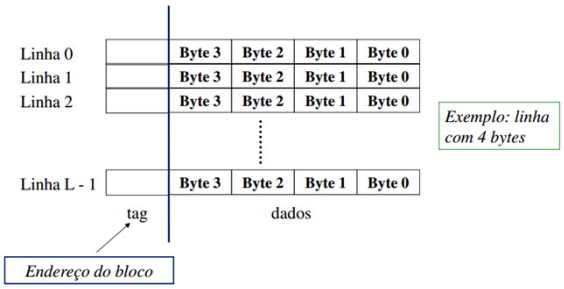
- Mais cache permite melhor velocidade de acesso.
- Mais cache sai mais caro.

- A cache aumenta muito o desempenho do computador, pois o processador é capaz de acessá-lo sem utilizar Wait States e sem precisar usar a frequência do barramento local.

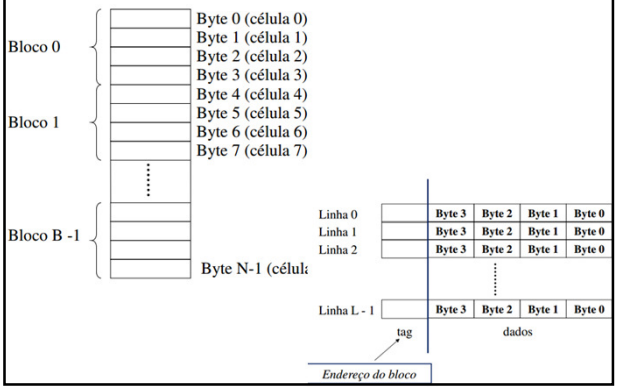
Organização da memória principal



Organização da memória cache

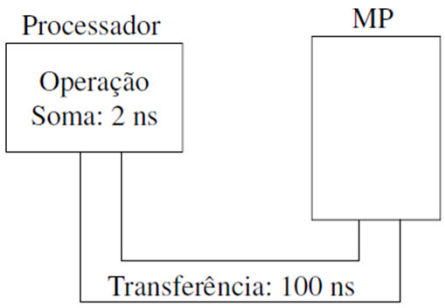


Organização da memória principal



Memórias cache

- MP possui N endereços (0 a N-1) de tamanho E (8 bits, por exemplo), ou seja, $2^E = N$;
- MP possui B Blocos (0 a B-1);
- B possui X Células (X=4, por ex.);
- Número de blocos da MP => B=N/X;
- Cada linha da cache (L) possui X Bytes;
- Tamanho da cache (L*X) é menor que MP (B*X).



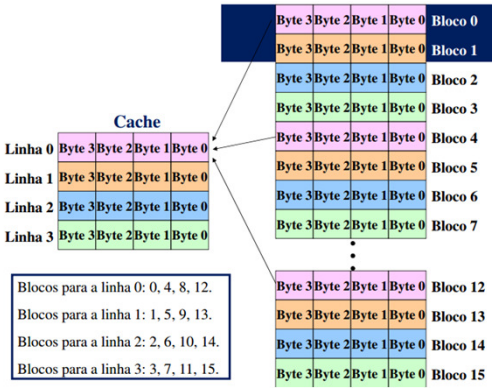
Mapeamento de Cache

- Cache com Mapeamento Direto
- Cache com Mapeamento Completamente Associativo
- Cache com Mapeamento Associativo por Grupo

Mapeamento direto

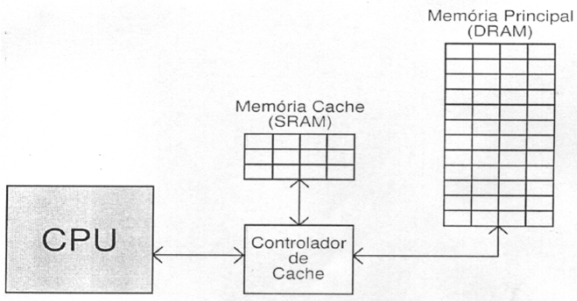
- **C**ada bloco da MP está diretamente mapeado a uma linha específica da cache.

Mapeamento direto

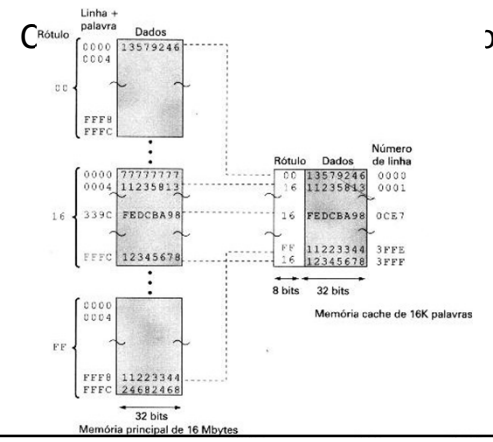


Cache com mapeamento direto

- Quando há substituição de informações na cache, substitui-se a linha a que se refere a informação.
- Cada linha de memória pode ficar em apenas uma linha da cache.



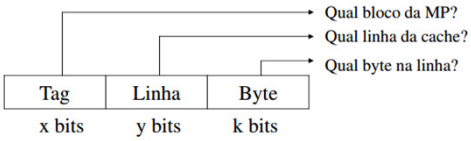
C)



Mapeamento direto - Exemplo

- Capacidade da MP = 64B ou 2^6 → endereço de 6 bits;
- Total de blocos = 64B /4B = 16;
- Capacidade da cache = 16B, 4B por linha;
- Total de linhas = 16/4 = 4 linhas.

Mapeamento direto



Endereço interpretado pela cache:

- Tag (2^x blocos por linha) – bloco armazenado. Ex: x = 2, pois existem 4 blocos associados por linha.
- Linha (2^y linhas) – endereço da linha da cache. Ex: x = 2, pois existem 4 linhas na cache.
- Byte (2^k bytes por bloco) – indica o byte solicitado na linha.

Mapeamento direto - exemplo

- Considere um sistema computacional com memória cache de 32KB de capacidade, constituída de linhas com 8B de largura. A MP possui capacidade de 16MB. Qual o número de bits necessários para a cache?
- Total de bits da cache = total de bits de dados + total de bits das tags

Mapeamento direto - exemplo

- Total de bits da cache = total de bits de dados + total de bits das tags
- Total de bits de dados= cache * 2^x * 8 bits
- Quantidade de linhas = cache * 2^x / tam bloco
- Quantidade de blocos = MP * 2^x / tam bloco
- **Obs.: x depende se é KB(10), MB(20), GB(30), ...**

Mapeamento direto - exemplo

- Total de bits da cache = total de bits de dados + total de bits das tags
- Blocos por linha = qtde blocos / qtde linhas
- Tamanho da tag = qtde linhas * expoente do bloco por linhas
- Total de bits da cache = Total de bits de dados + total de bits da tag

Mapeamento direto - exemplo

- Total de bits da cache = total de bits de dados + total de bits dos tags.
- Total de bits de dados = 32 x 2^{10} x 8 = 262.144 bits.
- Quantidade de linhas = (32 x 1024 bytes) / (8 bytes) = 2^{12} linhas.
- Quantidade de blocos = (16 x 2^{20} bytes) / (8 bytes) = 2^{21} blocos.
- Blocos por linha = 2^{21} / 2^{12} = 2^{9}.
- Tamanho da tag = 2^{12} x 9 = 36.864 bits.
- Total de bits da cache = 262.144 + 36.864 = 299.008 bits.

- Mais exemplos

Cache com mapeamento direto

- A RAM é dividida em blocos do tamanho da cache, e é carregado o bloco todo na cache.
- Se a cache é de 256 KB = 8192 linhas (Total Bytes (256*1024)/32 Bytes por linha)
- Cada linha = 32 Bytes (256 bits)
- Então, 256 KB = 262144 Bytes

Cache com mapeamento direto

- A RAM é dividida em blocos do tamanho da cache.
- Então, a memória principal é dividida em blocos de 8192 linhas, ou seja, de 256 KB
- 512 KB = ??? Bytes e ??? Linhas
- 1 MB = ??? Bytes e ??? Linhas

Prós e contras do mapeamento direto

- Simples.
- Barato.
- Local fixo para determinado bloco.
  - Se um programa acessa 2 blocos que mapeiam para a mesma linha repetidamente, perdas de cache são muito altas.

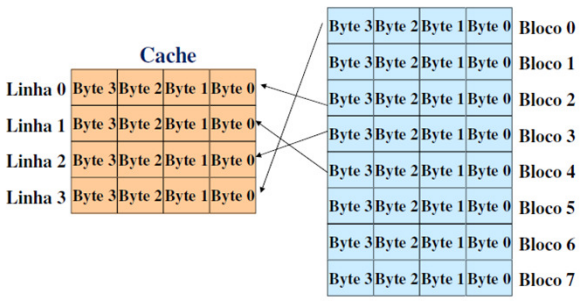
Mapeamento associativo

- Um bloco de memória principal pode ser carregado em qualquer linha de cache.
- Endereço de memória é interpretado como tag e palavra.
- Tag identifica exclusivamente o bloco de memória.

Mapeamento associativo

- Não há local fixo na cache para alocação de um bloco de MP.

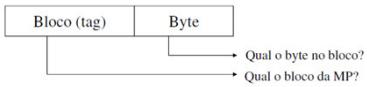
Mapeamento associativo



Mapeamento associativo

- Um bloco pode ser armazenado em qualquer linha;
- É preciso escolher qual bloco será substituído.

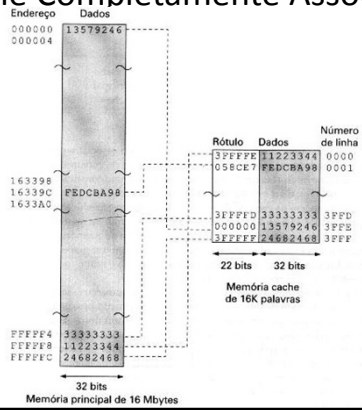
Mapeamento associativo



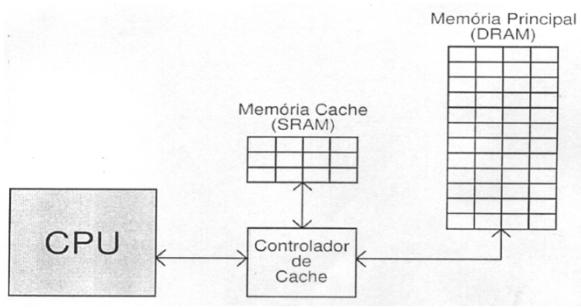
Endereçamento interpretado pela cache:

- Tag → número do bloco. Ex: 3 bits.
- Byte → byte na linha. Ex: 2 bits.

Cache Completamente Associativo



- exemplos



Cache Associativo por Grupo

- É o mais utilizado pois tem o melhor desempenho prático.
- O controlador de cache divide a cache em unidades menores.

Cache Associativo por Grupo

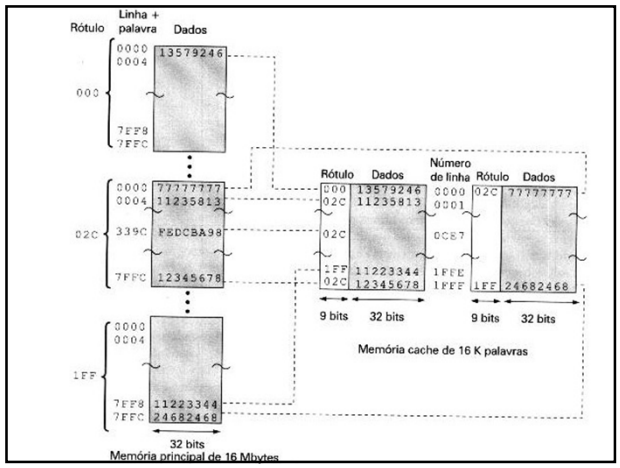
- A cache é dividida num número de conjuntos.
- Cada conjunto contém um número de linhas de cache.
- Cada linha de memória pode ficar em qualquer linha de um conjunto específico.

Cache Associativo por Grupo

- Exemplo:
  - Um cache associativo de 4 vias faz com que um cache de memória de 256 KB seja dividido em 4 caches independentes de 64 KB.
  - 256 KB /4 = 64 KB para cada cache.
- Assim, a RAM será dividida em blocos de 64 KB

Cache Associativo por Grupo

- É como se o computador tivesse 4 caches de 64 KB totalmente independentes um do outro, cada uma funcionando no esquema de mapeamente direto.



- Mapeamento Direto:
  - Cada linha de memória pode ficar em apenas uma linha da cache.
- Completamente Associativo
  - Cada linha de memória pode ficar em qualquer linha da cache.
- Associativo por Grupo
  - Cada linha de memória pode ficar em qualquer linha de um conjunto específico.