Memórias

Problema

Os programadores sempre ambicionaram ter quantidades ilimitadas de memória rápida.

Contudo, as memórias rápidas são de alto custo e, normalmente, de pequena capacidade também.

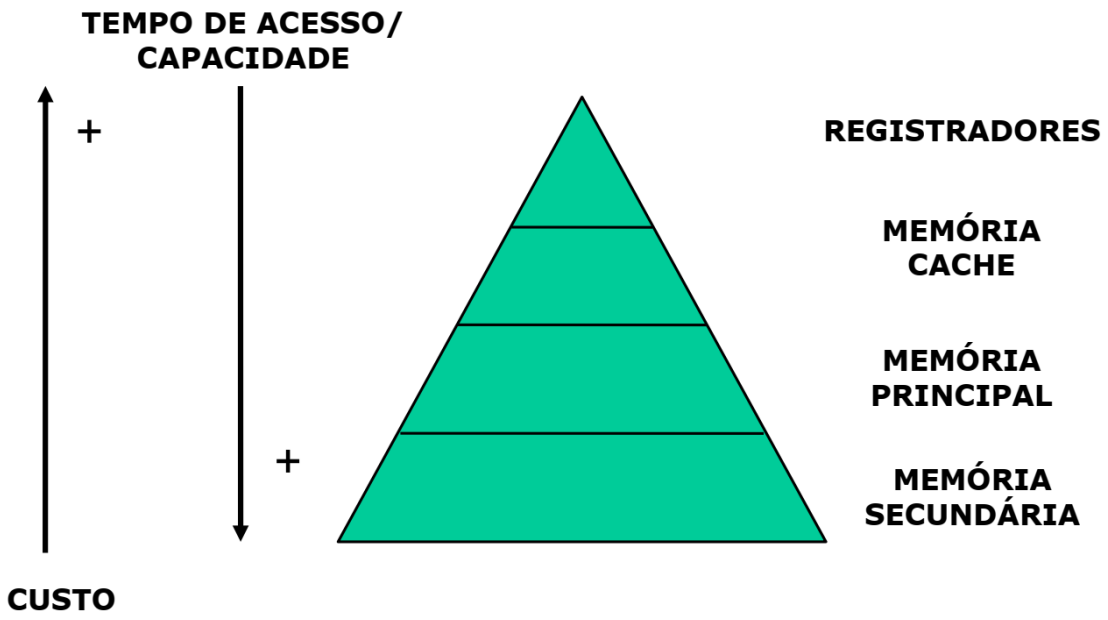
Uma solução é a organização do sistema de memória em uma hierarquia, com diversos níveis, onde memórias cada vez mais rápidas, menores e com um custo por byte maior, são colocadas nos níveis mais altos.

Hierarquia de Memórias

O objetivo é fornecer um sistema de memória com um custo próximo daquele do nível mais baixo da hierarquia, e velocidade próxima daquela do nível mais alto.

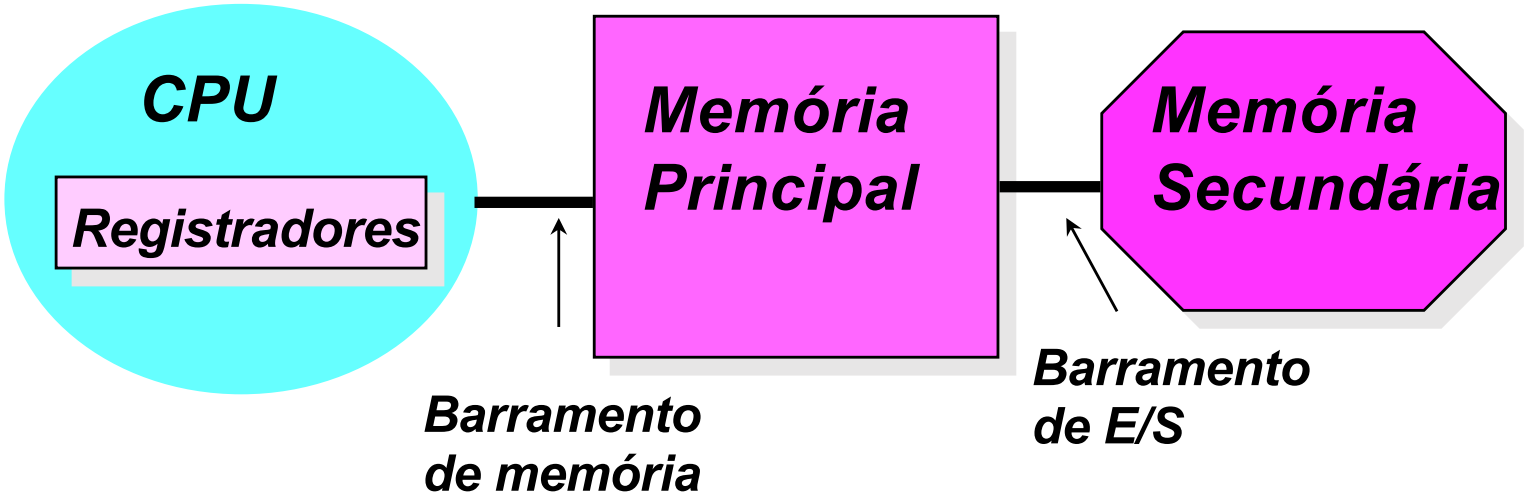
Os níveis de hierarquia mais altos normalmente são um subconjunto dos níveis mais baixos.

À medida que a informação vai sendo utilizada, ela vai sendo copiada para os níveis mais altos da hierarquia de memória.

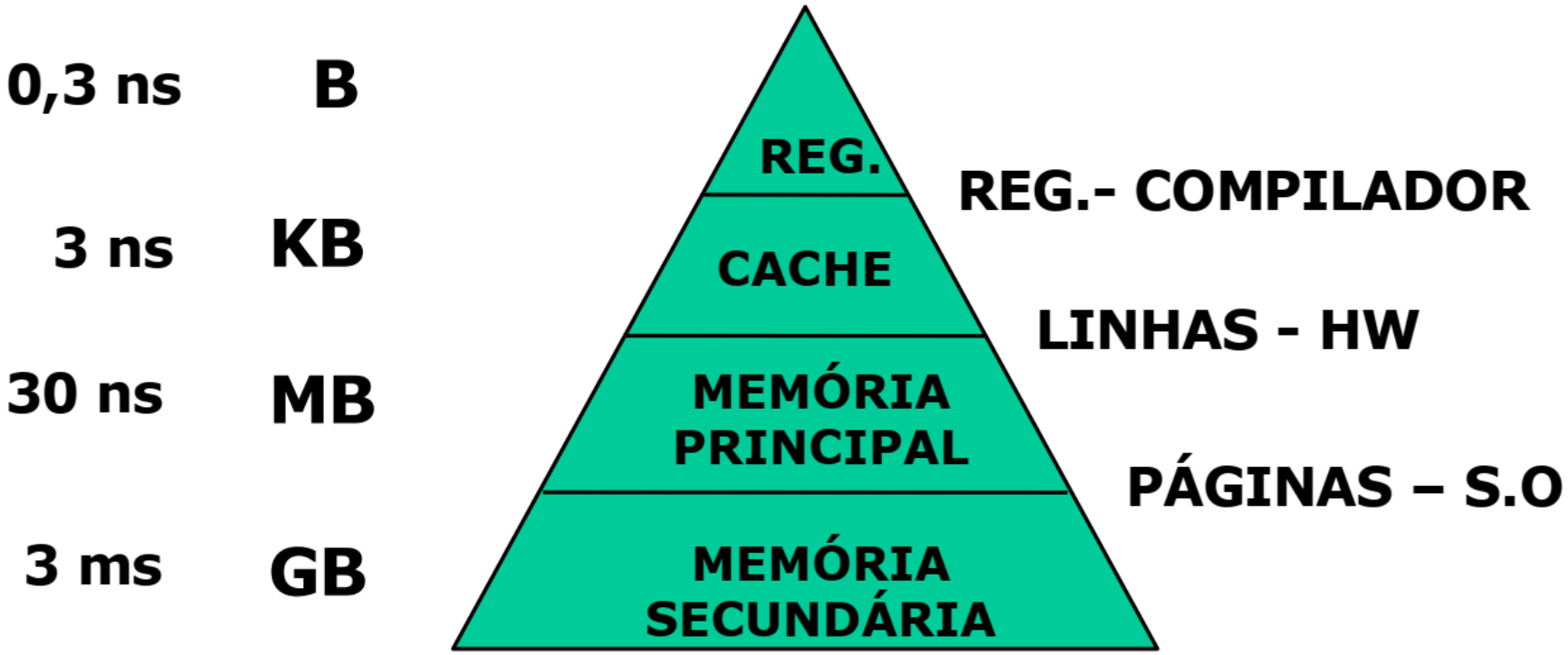


Hierarquia de Memórias

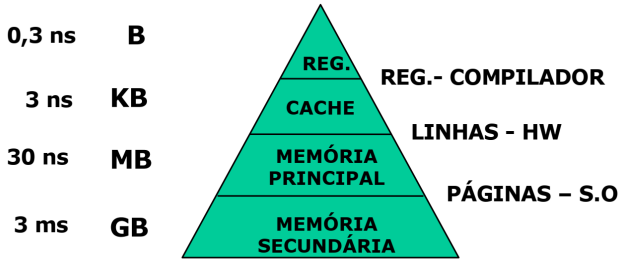
Esquema Básico de Memória



Hierarquia de Memórias



Hierarquia de Memórias

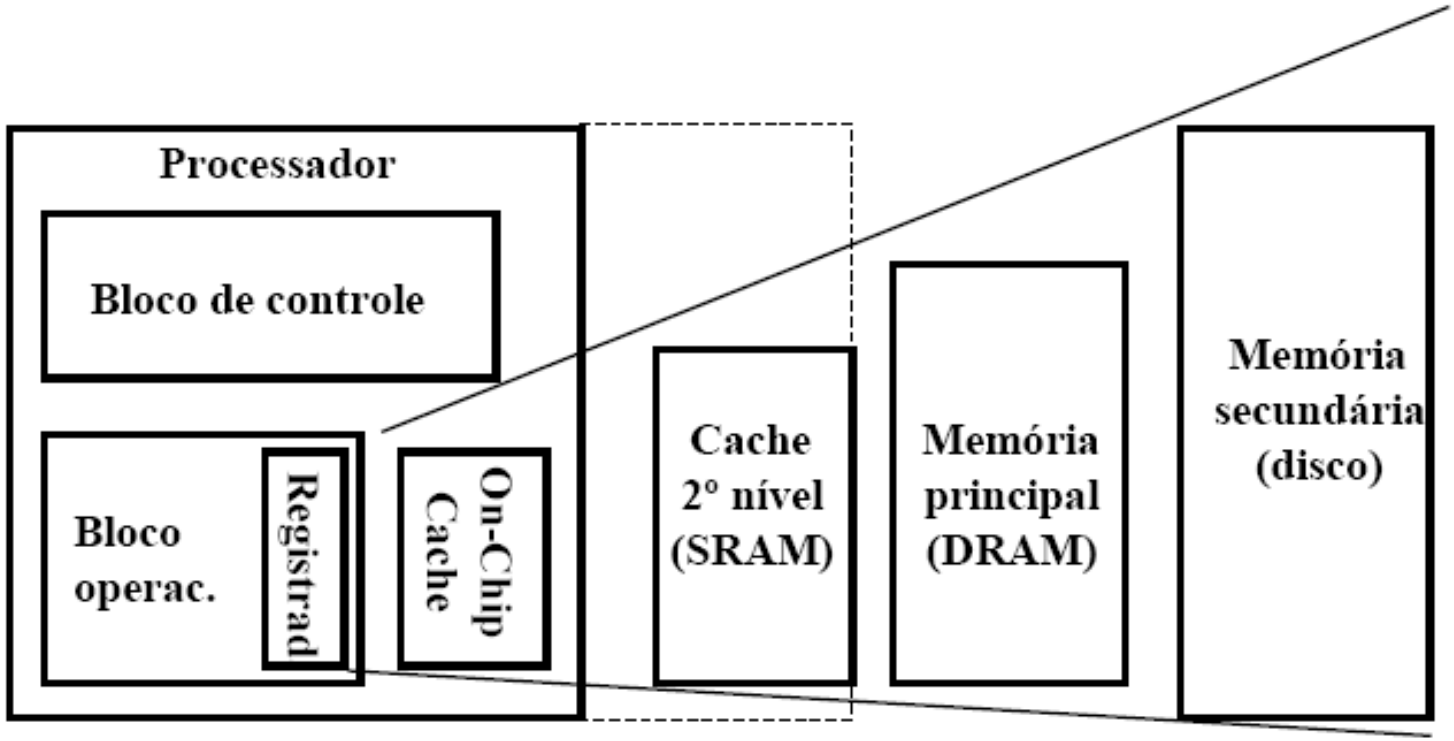


A cada nível que se sobe na hierarquia, endereços de uma memória maior são mapeados para uma memória menor.

Junto com esse mapeamento está associado uma função de proteção, evitando que dados de um usuário sejam modificados por outro.

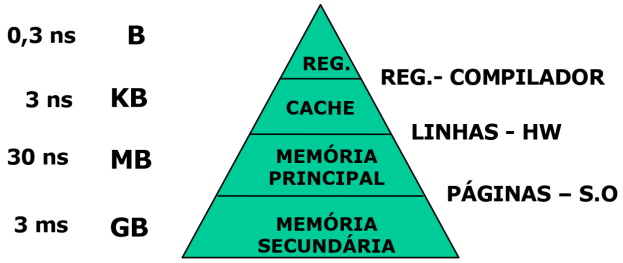
A importância da hierarquia de memória aumentou nos últimos anos devido ao aumento no desempenho dos processadores.

Hierarquia de Memórias



Velocidade (ns):	0,1	1	2-5	10-20	10.000.000 (10 ms)
Tamanho (bytes):	100	16 K	512 K	256 M	Gs

Princípio da Localidade



• O funcionamento da hierarquia de memória está fundamentado em duas características encontradas nos programas.

• Existe uma grande probabilidade de o processador executar os mesmos trechos de código e utilizar repetidamente dados Próximos

• A essa qualidade dos programas denominamos:

– Localidade Temporal

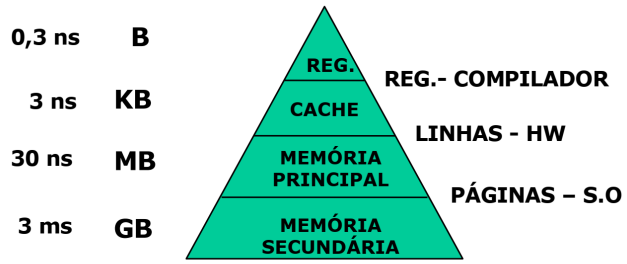
– Localidade Espacial

Princípio da Localidade

Localidade temporal: posições de memória, uma vez referenciadas (lidas ou escritas) , tendem a ser referenciadas novamente dentro de um curto espaço de tempo.
– Usualmente encontrada em laços de instruções e acessos a pilhas de dados e variáveis

Localidade espacial: se uma posição de memória é referenciada, posições de memória cujos endereços sejam próximos da primeira tendem a ser logo referenciados.
– A informação é manipulada em blocos no sistema de hierarquia de memória para fazer uso da localidade espacial.

Princípio da Localidade

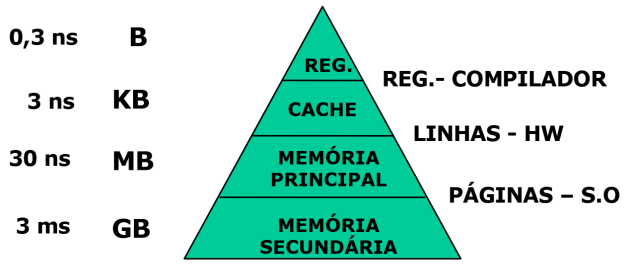


- No início dos tempos o único nível que possui informação válida é o mais inferior de toda a hierarquia, composto de dispositivos de armazenamento não voláteis.

- Na medida em que a informação vai sendo utilizada, ela é copiada para os níveis mais altos da hierarquia.

- Quando fazemos um acesso a um nível da hierarquia e encontramos a informação desejada, dizemos que houve um acerto, em caso contrário dizemos que houve uma falha.

Princípio da Localidade

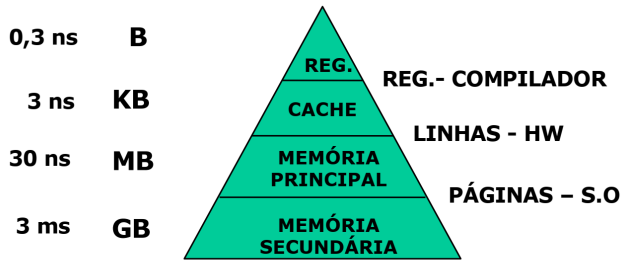


- Acessos que resultam em acertos nos níveis mais altos da hierarquia podem ser processados mais rapidamente.

- Os acessos que geram falhas, obrigando a buscar a informação nos níveis mais baixos da hierarquia, levam mais tempo para serem atendidos.

- Para um dado nível da hierarquia possa ser considerado eficiente é desejável que o número de acertos seja bem maior do que o número de falhas.

Princípio da Localidade

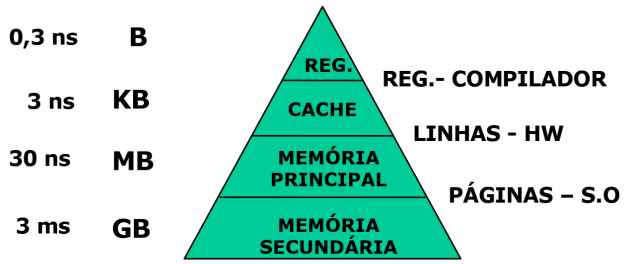


- Define-se como taxa de acertos (h) a relação entre o número de acertos e o número total de acessos para um dado nível da hierarquia de memória.

h = número acertos / total de acessos

O total de acessos inclui tanto os acessos de leitura como os de escrita.

Princípio da Localidade



- O **tempo de acesso com acerto** é o tempo necessário para buscar a informação em um dado nível de hierarquia, que inclui o tempo necessário para determinar se o acesso à informação vai gerar um acerto ou uma falha.

- A **penalidade por falha ou tempo acesso com falha** é o tempo necessário para buscar a informação nos níveis inferiores da hierarquia, armazená-la no nível atual e enviá-la para o nível superior.

Componentes da Hierarquia

Os elementos mais importantes da uma hierarquia de memória são:

- Registradores
- Memória Cache
- Memória Principal
- Memória Secundária

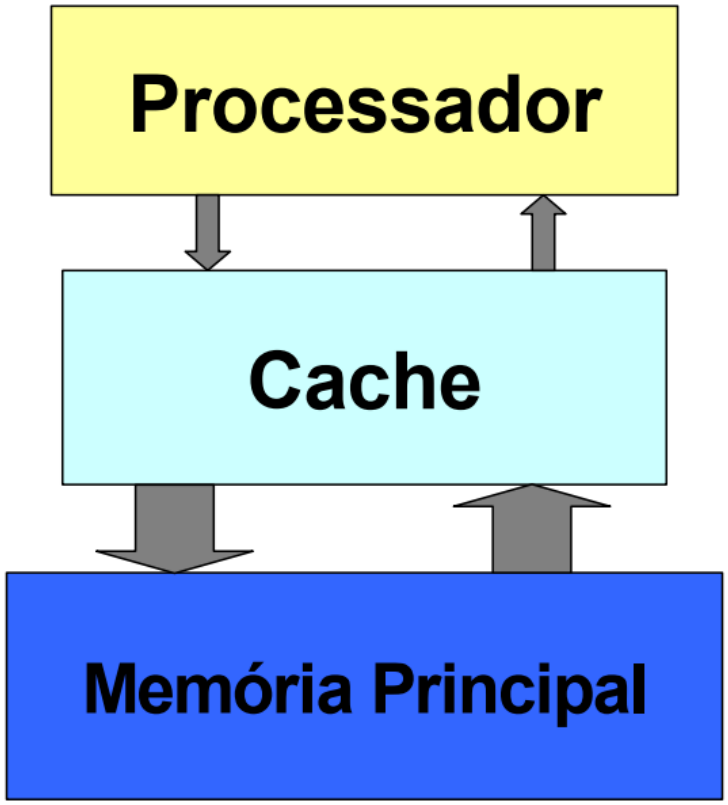
Registrador

• Registradores estão localizados no núcleo do processador. São caracterizados por um tempo de acesso menor que um ciclo de relógio e sua capacidade é da ordem de centenas de bytes.

• O controle de qual informação deve estar nos registradores é feita explicitamente pelo compilador, que determina quais variáveis serão colocadas no registrador.

• É o único nível da hierarquia que permite movimentações iguais apenas ao tamanho da informação desejada

Componentes da Hierarquia



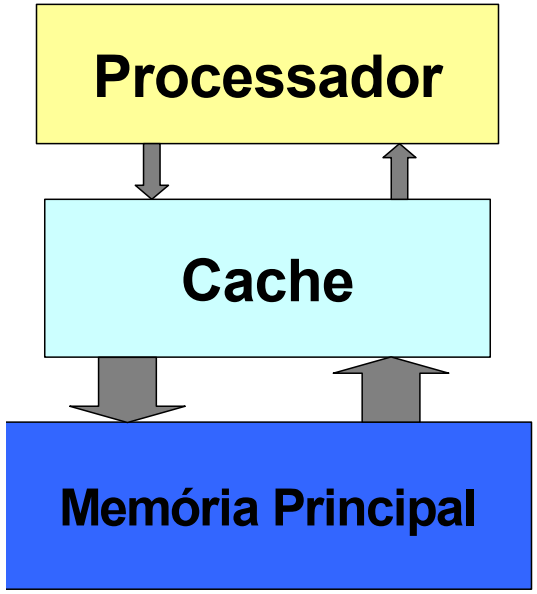
Memória Cache: Elemento de memória intermediário entre o Processador e a Memória Principal

Memória Cache

O processador inicia a busca a instrução ou dados na memória cache.

• Se os dados ou a instrução estiverem na cache (denomina-se acerto), a informação é transferida para o processador.

• Se os dados ou a instrução não estiverem na memória cache (chama-se falha), então o processador aguarda, enquanto a instrução/dados desejados são transferidos da memória principal para a cache e também para o processador

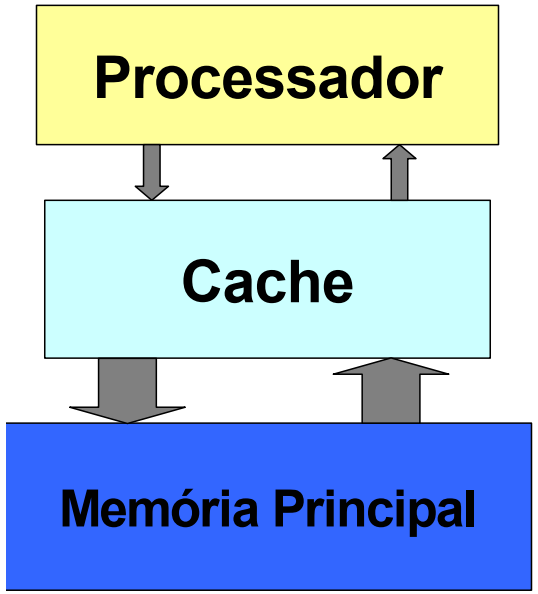


Memória Cache

• Durante a busca da palavra que está faltando na cache, é trazido um bloco (ou linha) inteiro da memória principal, ao
invés de apenas uma palavra.

• O objetivo é minimizar a taxa de falhas nos próximos acessos, seguindo o princípio da localidade espacial.

• O tamanho de um bloco é um parâmetro de projeto na memórias caches, tamanhos usuais são 32, 64 e 128 bytes

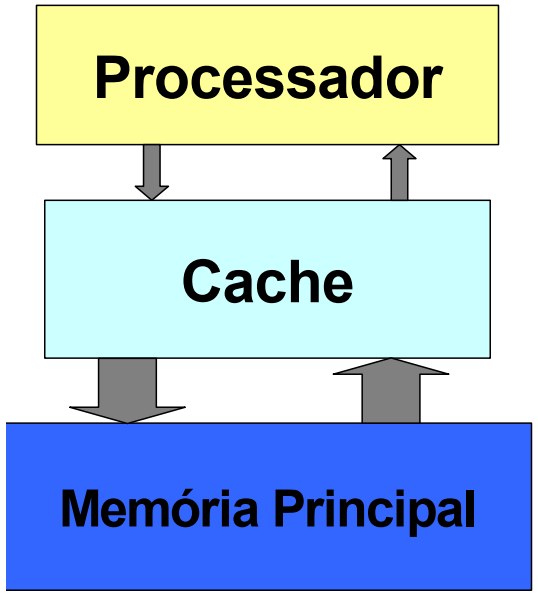


Memória Cache

- As células de memória da cache são elaboradas com tecnologia que permite um tempo de acesso menor que o memória principal.

- Normalmente são células de memória estática (SRAM), com menor capacidade, porém maior custo e maior velocidade.

- Associado a essas memórias está um controlador, normalmente uma máquina de estados, que faz o controle da transferência dos blocos para a memória principal.

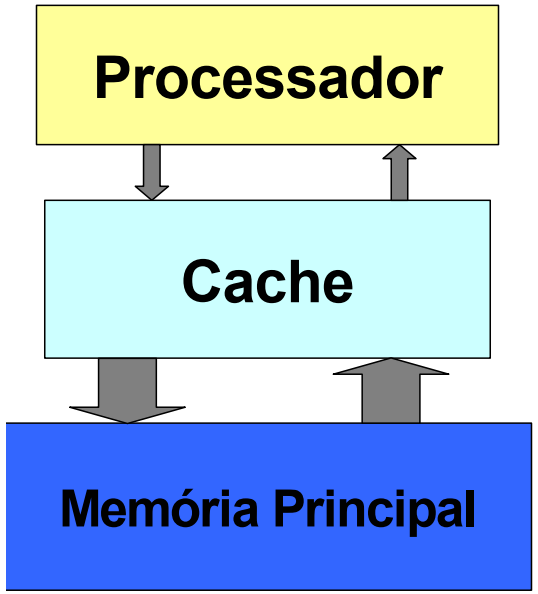


Memória Principal

• A memória principal é constituída por células de memória dinâmica (DRAM).

• As memórias DRAM tem grande capacidade de armazenamento, mas são mais lentas que as memórias estáticas.

• As memórias DRAM possuem também tempos de acesso distintos para leitura e escrita, e necessitam de uma lógica de restauração (“refresh”), quer embutida ou fora da pastilha, que afetam o tempo médio de acesso.



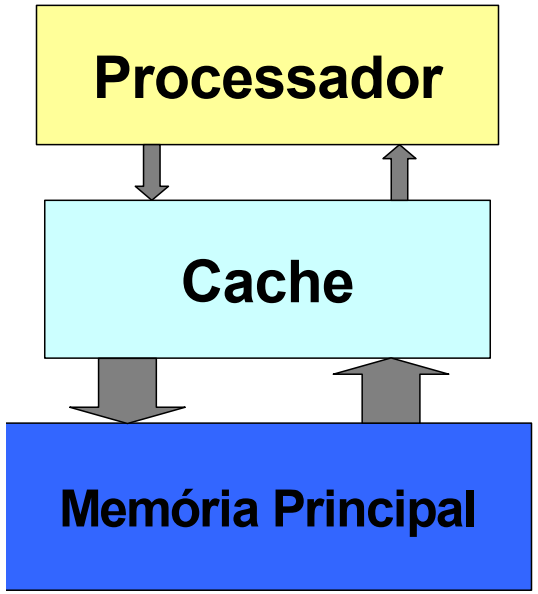
Tecnologias de Memórias Voláteis

DRAM (Dynamic Random Acess Memory)

- Grande capacidade de integração (baixo custo por bit)
- Perda de informação após algum tempo: Necessidade de refreshing

SRAM (Static Random Acess Memory)

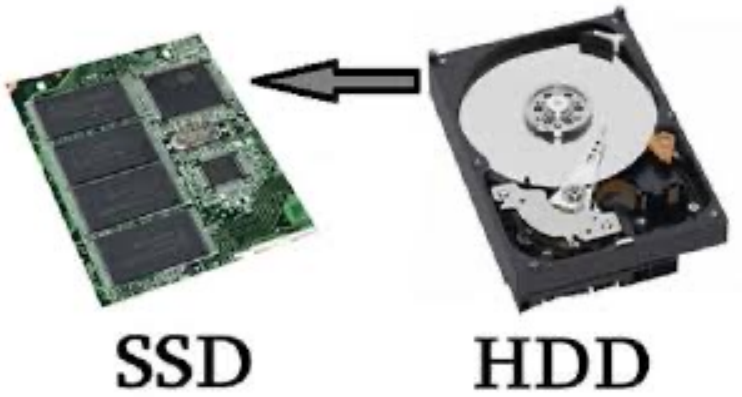
- Pequeno tempo de acesso
- Não existe necessidade de refreshing
- Alto custo por bit (baixa integração)



Memória Secundária

• A memória secundária é o último nível da hierarquia de memória. É composta pelos dispositivos de armazenamento de massa, normalmente discos rígidos, de grande capacidade e menor custo por byte armazenado.

• Os programas e arquivos são armazenados integralmente na memória secundária, que são dispositivos de memória não volátil.

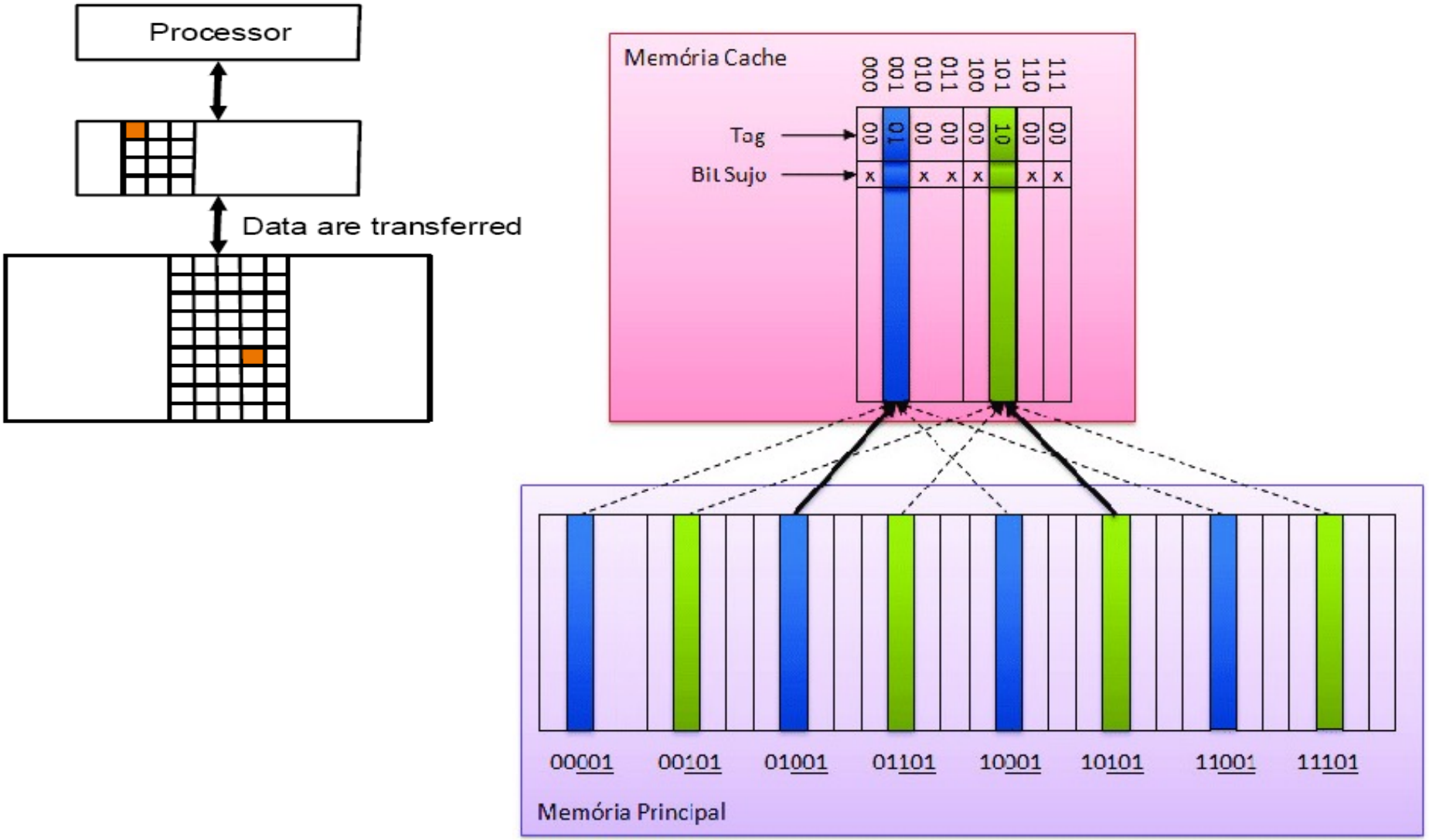


Memória Secundária

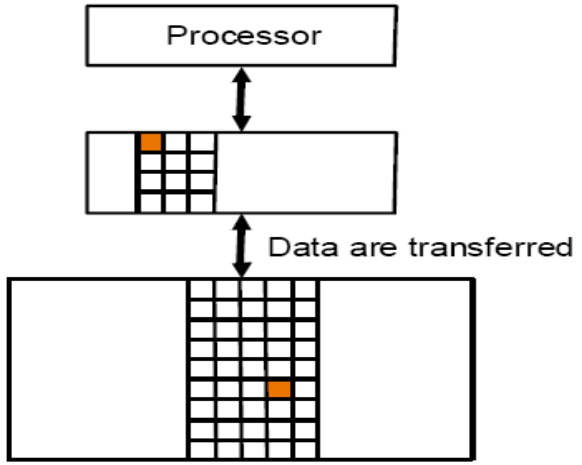
- O controle de qual informação deve permanecer na memória principal ou na memória secundária é feito pela Memória Virtual.

- A memória virtual é um conjunto de “hardware” e de rotinas do sistema operacional. Além do controle da hierarquia entre a memória principal e a memória secundária, ela realiza a proteção, evitando que um programa modifique informações que pertençam a algum outro.

Busca de Dados da Cache



Busca de Dados da Cache



Mapeamento completamente associativo

- Um bloco da memória principal pode ser armazenado em qualquer linha da memória cache.

Mapeamento direto

- Cada bloco da memória principal só pode ser mapeado em uma única linha da memória cache. Normalmente utilizam-se os bits menos significativos do endereço do bloco para definir qual será esta linha.

Mapeamento associativo por conjunto

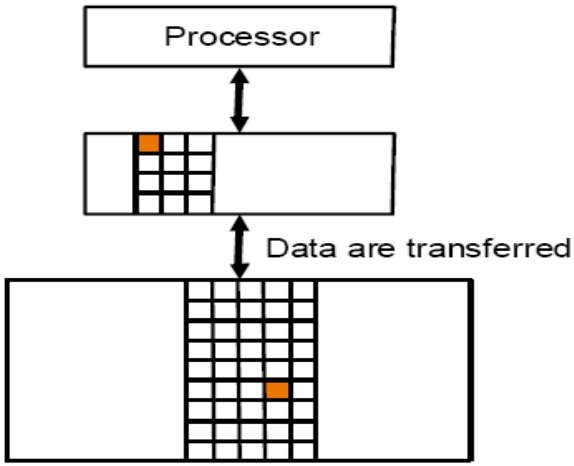
- Cada bloco da memória principal pode ser armazenado apenas em um determinado conjunto de linhas da cache.

Busca de Dados da Cache

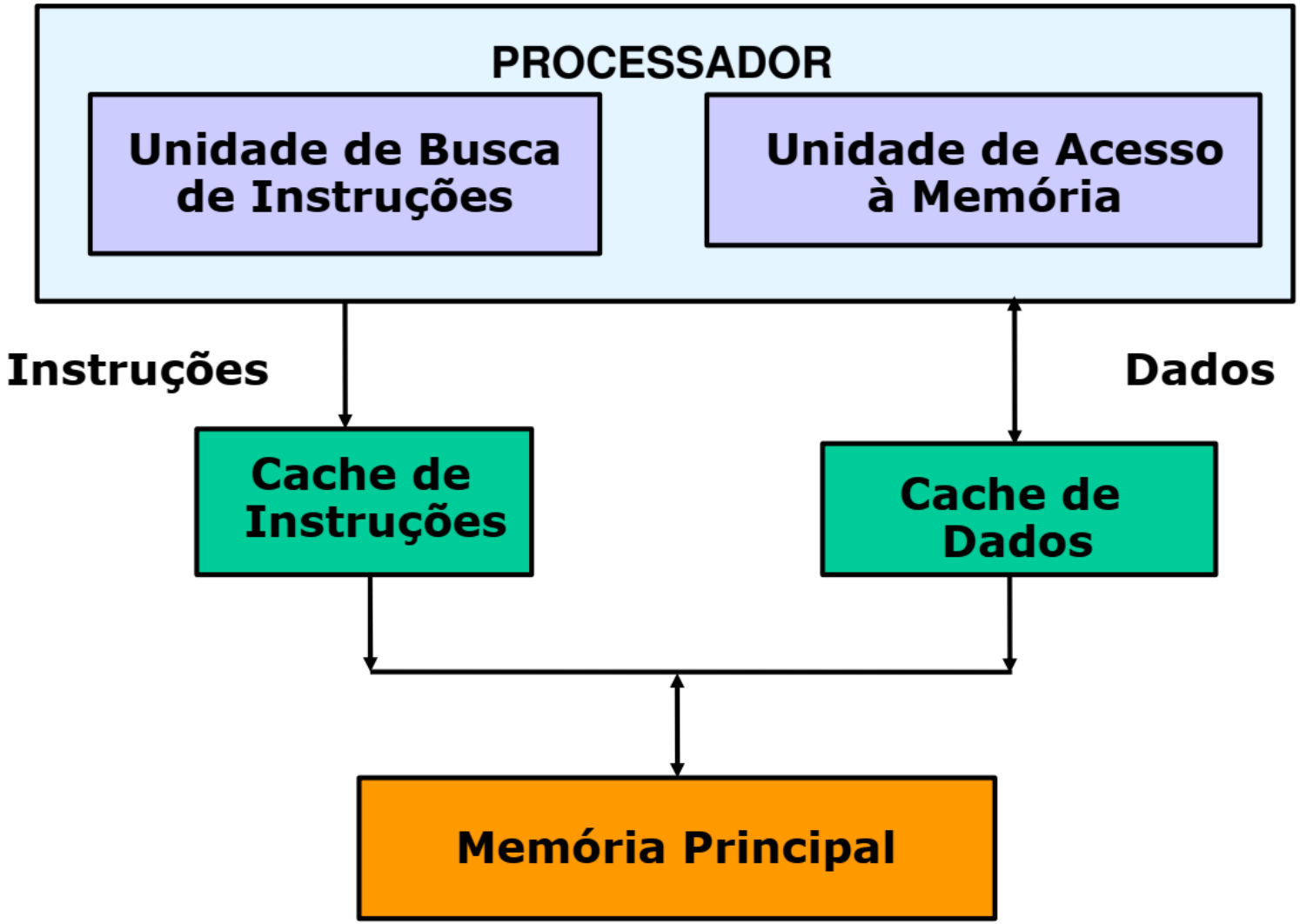
Ao ser escrever em um bloco, temos duas políticas básicas em caso de acerto:

- **Write-through (Cache)**
  - **Escreve-se o dado no nível da hierarquia atual e no inferior**

- **Write-back (Memória Virtual e Cache)**
  - **Escreve-se o dado apenas no nível de hierarquia atual e, quando o bloco ou página for substituído, ele é atualizado no nível inferior.**



Cache de Dados e Instruções



Cache de Dados e Instruções

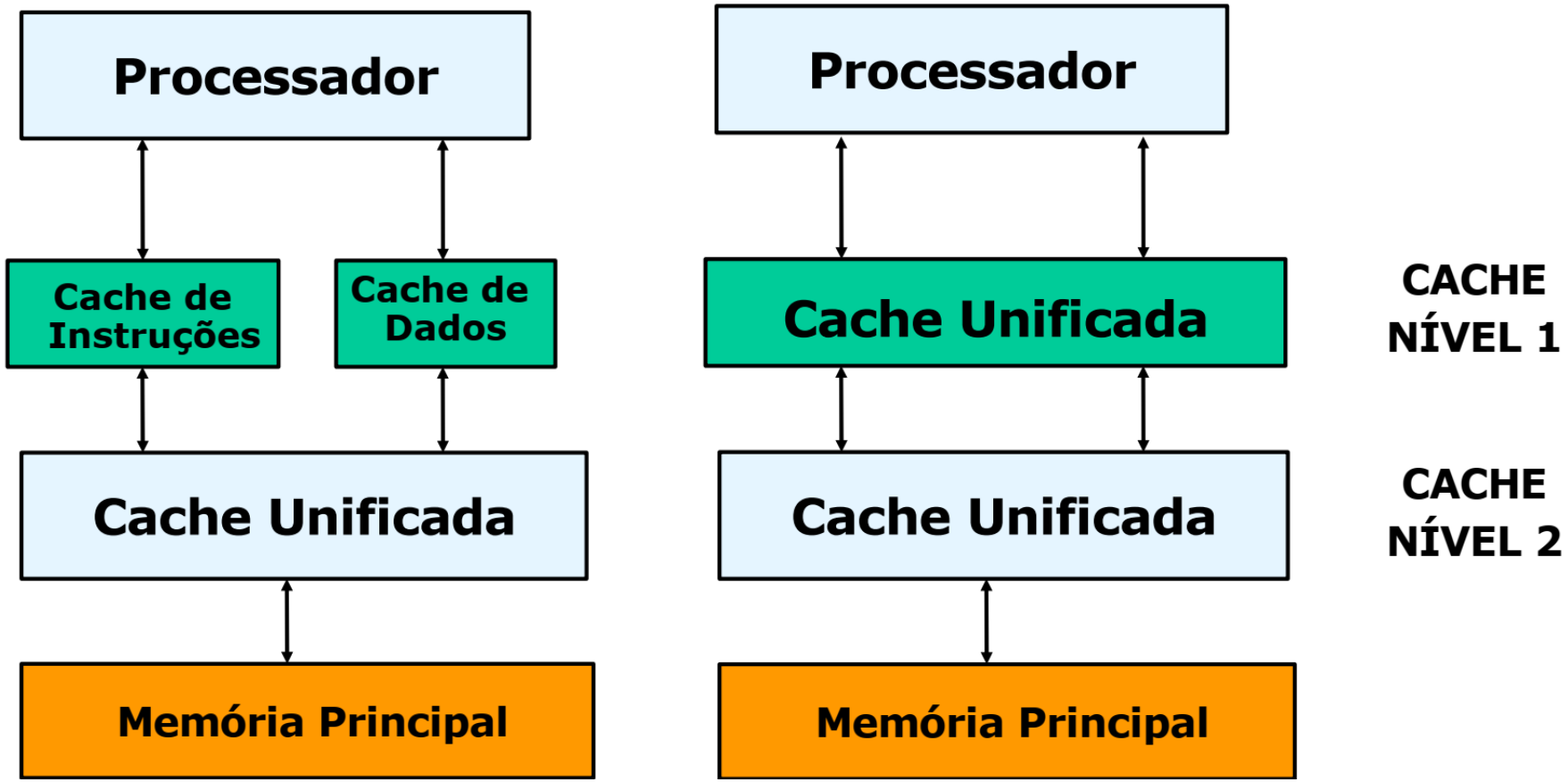
Dados e instruções: cache unificada x caches separadas.

Vantagens das caches separadas:

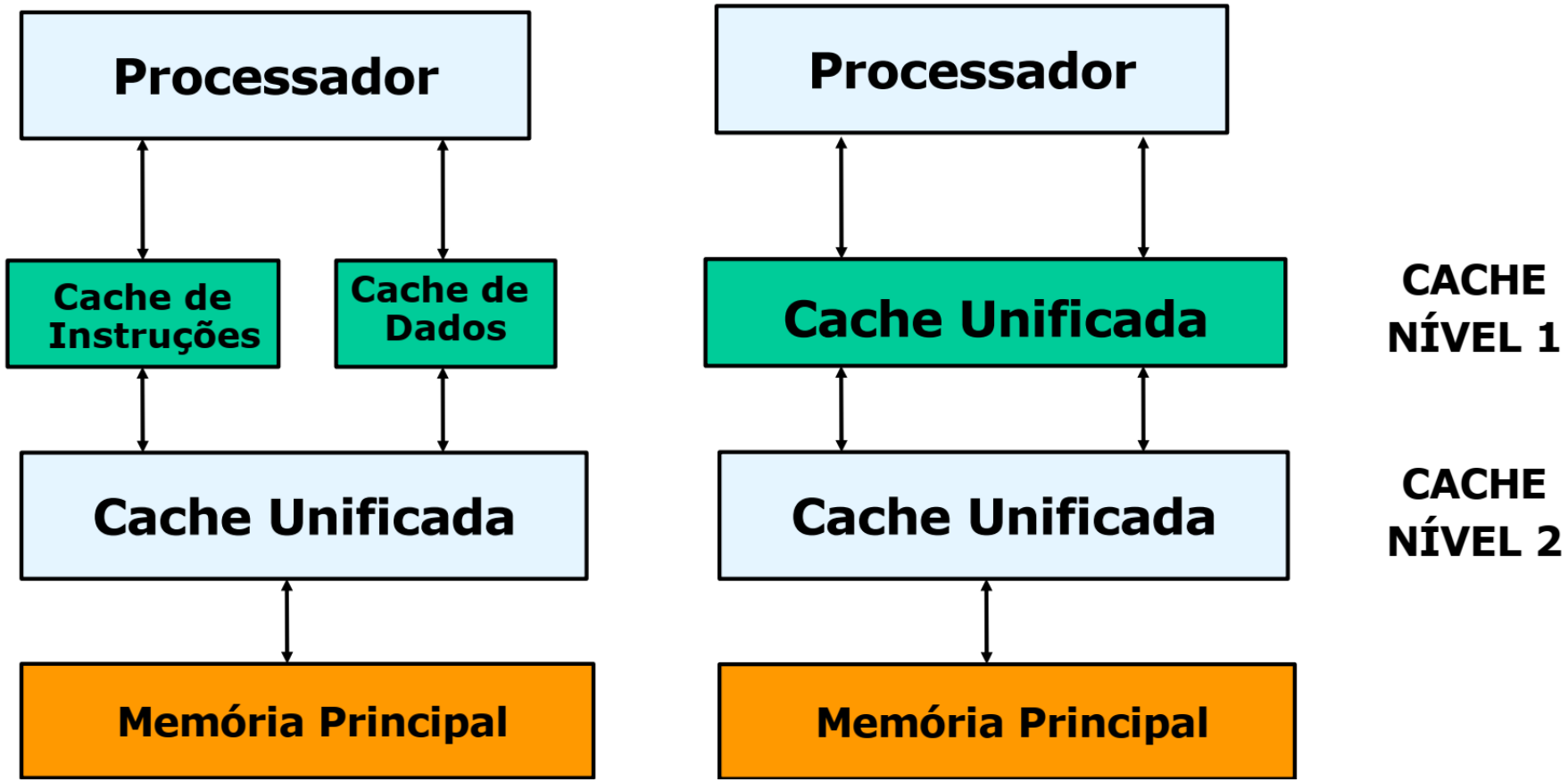
- **política de escrita só precisa ser aplicada à cache de dados;**
- **caminhos separados entre memória principal e cada cache, permitindo transferências simultâneas (p.ex. quando o processador possui um *pipeline*);**
- **Estratégias diferentes para cada cache: tamanho total, tamanho de linha, organização.**

Caches separadas são usadas, p.ex., no Pentium e no 68040.

Cache Multinível



Cache Multinível



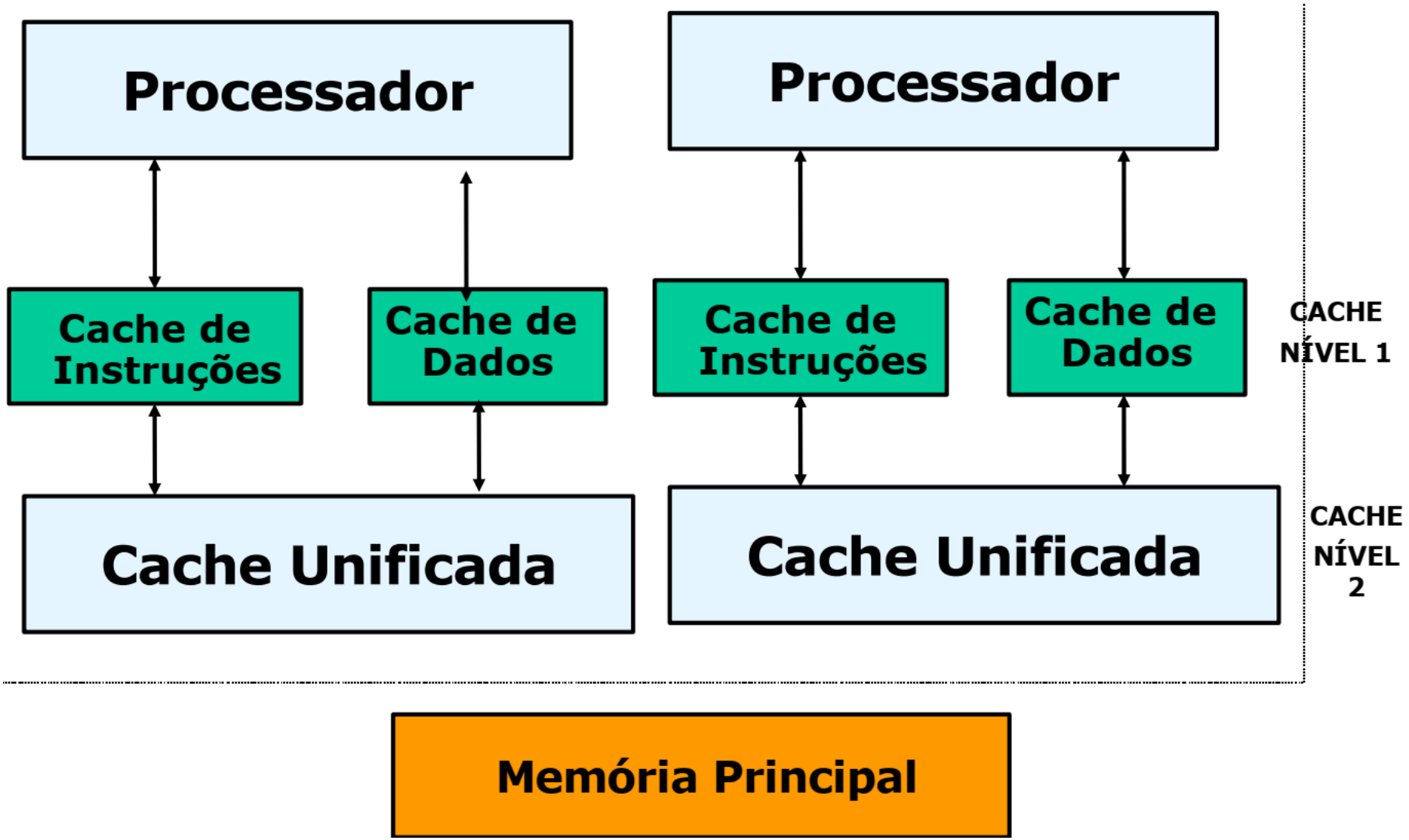
Cache Multinível

Manter a cache de nível 1 pequena e muito rápida, acompanhando o relógio da CPU.

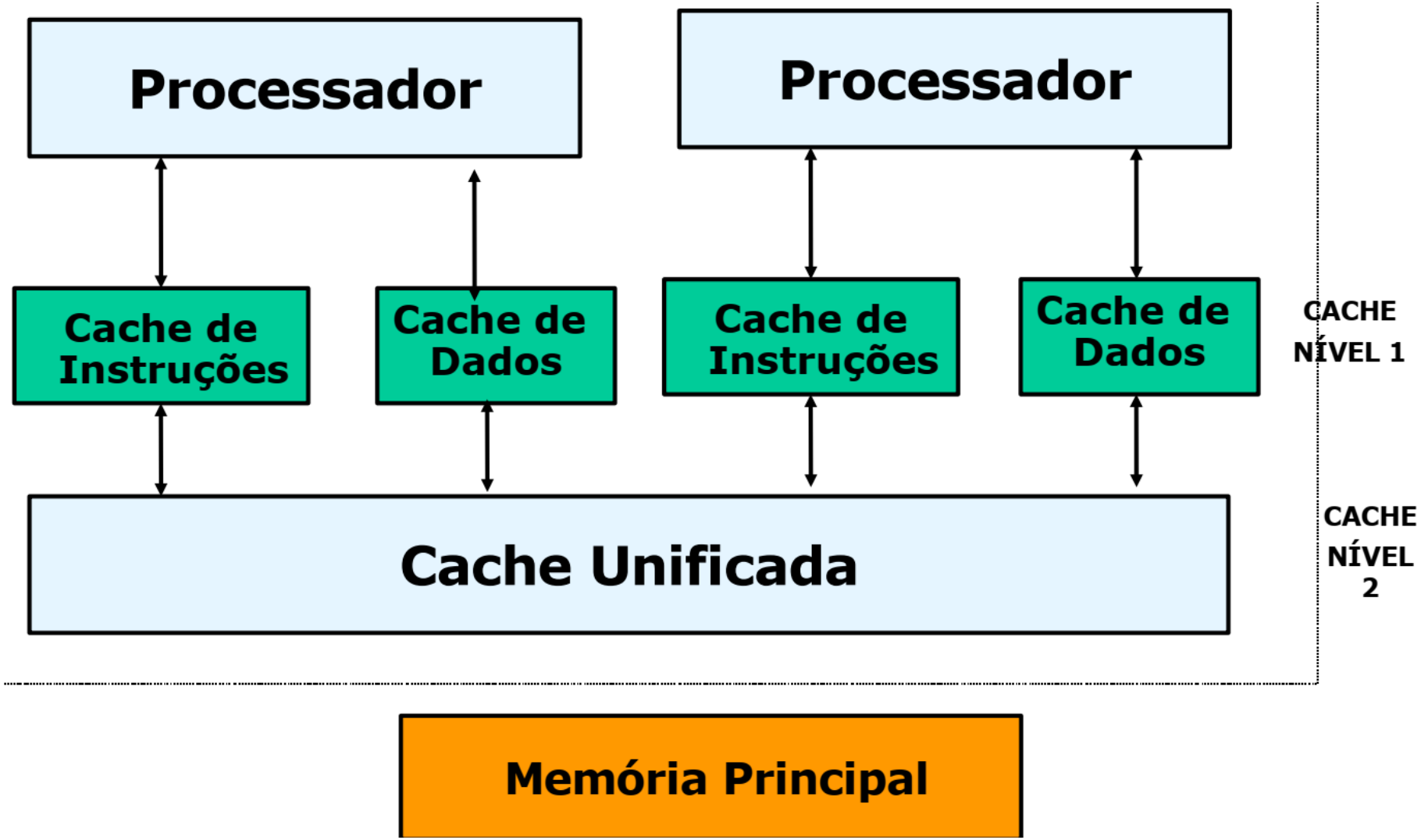
Utilizar um cache de nível 2 grande, mas não tão rápida, mas capaz de reduzir a penalidade das falhas.

Normalmente utiliza-se a cache de nível 1 separada para dados e instruções e a cache de nível 2 unificada.

Cache Multicores



Cache Multicores



Falhas de Acesso ao Dado na Cache

Falhas Compulsórias: São faltas no acesso à cache, causadas pelo primeiro acesso que nunca esteve na cache.

Falhas devido à Capacidade: São faltas que ocorrem porque a cache não pode armazenar todos os blocos necessários à execução de um programa.

Falhas por Conflitos ou Colisão: São faltas que ocorrem no acesso à cache quando diversos blocos competem pelo mesmo conjunto. Não ocorrem em caches totalmente associativas.

Comparação de Cache em Alguns Processadores

PROCESSADOR	TAMANHO DA CACHE
Intel CELERON	L1 – (12 K μop + 16KB) (int.) L2 – 256 KB (int.)
Intel PENTIUM III	L1 – (16 KB + 16KB) L2 – 256 KB (int.) ou 512 KB (ext.)
Intel PENTIUM IV HT	L1 – (12 K μ op + 8KB) (int.) L2 – 512 KB (int.)
AMD DURON	L1 – (64 KB + 64 KB) (int.) L2 – 64 KB (int.) (excl.)
AMD ATHLON	L1 – (64 KB + 64 KB) (int.) L2 – 512 KB (int.)