Aula 04: Arquitetura de Computadores – Hierarquia de Memória; Memória cache

Prof. Hugo Puertas de Araújo
hugo.puertas@ufabc.edu.br
Sala: 509-2 (5º andar / Torre 2)

Arquitetura de Computadores

Objetivos de aprendizagem

- Apresentar uma visão geral das características dos sistemas de memória do computador e do uso da hierarquia da memória.
- Descrever os conceitos básicos e o objetivo da memória cache.
- Discutir os elementos-chave do projeto da cache.
- Fazer distinção entre mapeamento direto, mapeamento associativo e mapeamento associativo por conjunto.
- Compreender as implicações do desempenho dos diversos níveis de memória.

Visão geral do sistema de memória do computador

Localização	Desempenho
Interna (por exemplo, registradores do processador, memória principal, cache)	Tempo de acesso
Externa (por exemplo, discos ópticos, discos magnéticos, fitas)	Tempo de ciclo
	Taxa de transferência
Método de acesso	**Tipo físico**
Sequencial	Semicondutor
Direto	Magnético
Aleatório	Óptico
Associativo	Magneto-óptico
Unidade de transferência	**Características físicas**
Palavra	Volátil/não volátil
Bloco	Apagável/não apagável
Capacidade	**Organização**
Número de palavras	Módulos de memória
Número de bytes	

Visão geral do sistema de memória do computador

- O termo **localização** indica se a memória é interna ou externa ao computador.
- Uma característica óbvia da memória é a sua capacidade.
- Um conceito relacionado é a unidade de transferência.
- Para a memória interna, a unidade de transferência é igual ao número de linhas elétricas que chegam e que saem do módulo de memória.

Visão geral do sistema de memória do computador

- O termo **localização** indica se a memória é interna ou externa ao computador.
  E.: O que é interno e externo ao computador?
- Uma característica óbvia da memória é a sua capacidade.
- Um conceito relacionado é a unidade de transferência.
- Para a memória interna, a unidade de transferência é igual ao número de linhas elétricas que chegam e que saem do módulo de memória.
  E.: Por que isso não vale p/ mem. ext.?

Visão geral do sistema de memória do computador

- O método de acesso das unidades de dados inclui:
  - **Acesso sequencial**: a memória é organizada em unidades de dados chamadas registros, acessados sequencialmente. **Ex.: fita magnética.**
  - **Acesso direto**: acesso é realizado pelo acesso direto, para alcançar uma vizinhança geral, mais uma busca sequencial, contagem ou espera, até chegar ao local final. Ex.: disco magnético ou óptico.
  - **Acesso aleatório**: cada local endereçável na memória tem um mecanismo de endereçamento exclusivo, fisicamente interligado. **Ex.: RAM, Flash.**
  - **Associativo**: permite fazer uma comparação de um certo número de bits com uma combinação específica, fazendo isso com todas as palavras simultaneamente. Assim, uma palavra é recuperada com base em uma parte de seu conteúdo, em vez de seu endereço. **Ex.: cache associativo.**

Visão geral do sistema de memória do computador

- Do ponto de vista do usuário, as duas características mais importantes da memória são capacidade e desempenho.
- Três parâmetros de desempenho são usados:
  i. Tempo de acesso (latência)
  ii. Tempo de ciclo de memória
  iii. Taxa de transferência
- Uma variedade de tipos físicos da memória têm sido empregados. Os mais comuns hoje em dia são memória semicondutora; memória de superfície magnética, usada para disco e fita; óptica e magneto-óptica.

A hierarquia de memória

- As restrições de projeto podem ser resumidas por três questões:
  i. Quanto?
  ii. Com que velocidade?
  iii. A que custo?
- A questão da quantidade é, de certa forma, livre.
- Para conseguir maior desempenho, a memória deve ser capaz de acompanhar a velocidade do processador.
- O custo da memória deve ser razoável em relação a outros componentes.

A hierarquia de memória

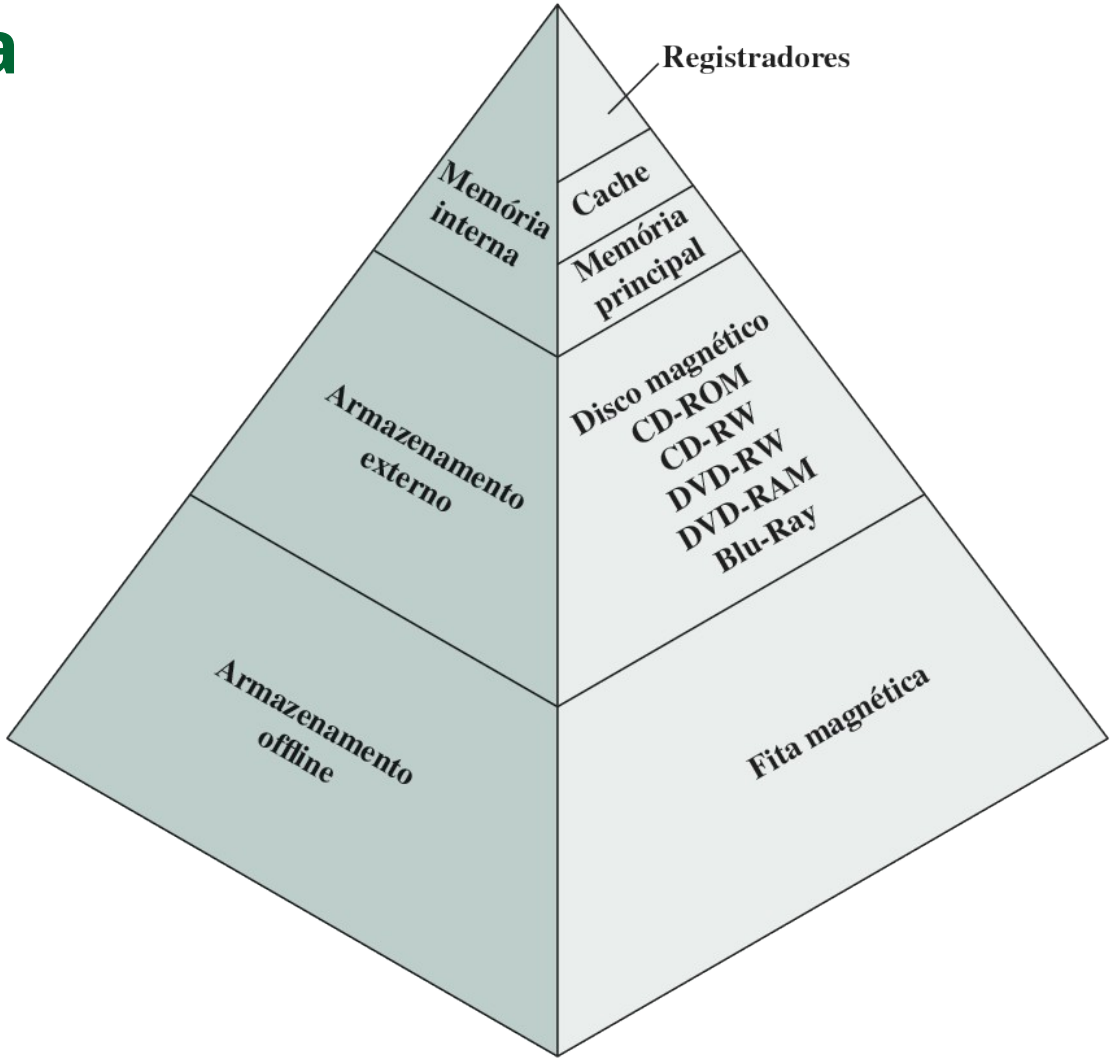
- Diversas tecnologias são usadas para implementar sistemas de memória e, por meio desse espectro de tecnologias, existem as seguintes relações:
  i. Tempo de acesso mais rápido → maior custo por bit.
  ii. Maior capacidade → menor custo por bit.
  iii. Maior capacidade → tempo de acesso mais lento.
- O dilema que o projetista enfrenta é **claro**.

A hierarquia de memória

- Diversas tecnologias são usadas para implementar sistemas de memória e, por meio desse espectro de tecnologias, existem as seguintes relações:
  i. Tempo de acesso mais rápido → maior custo por bit.
  ii. Maior capacidade → menor custo por bit.
  iii. Maior capacidade → tempo de acesso mais lento.
- O dilema que o projetista enfrenta é **claro**.
  E.: Qual é?

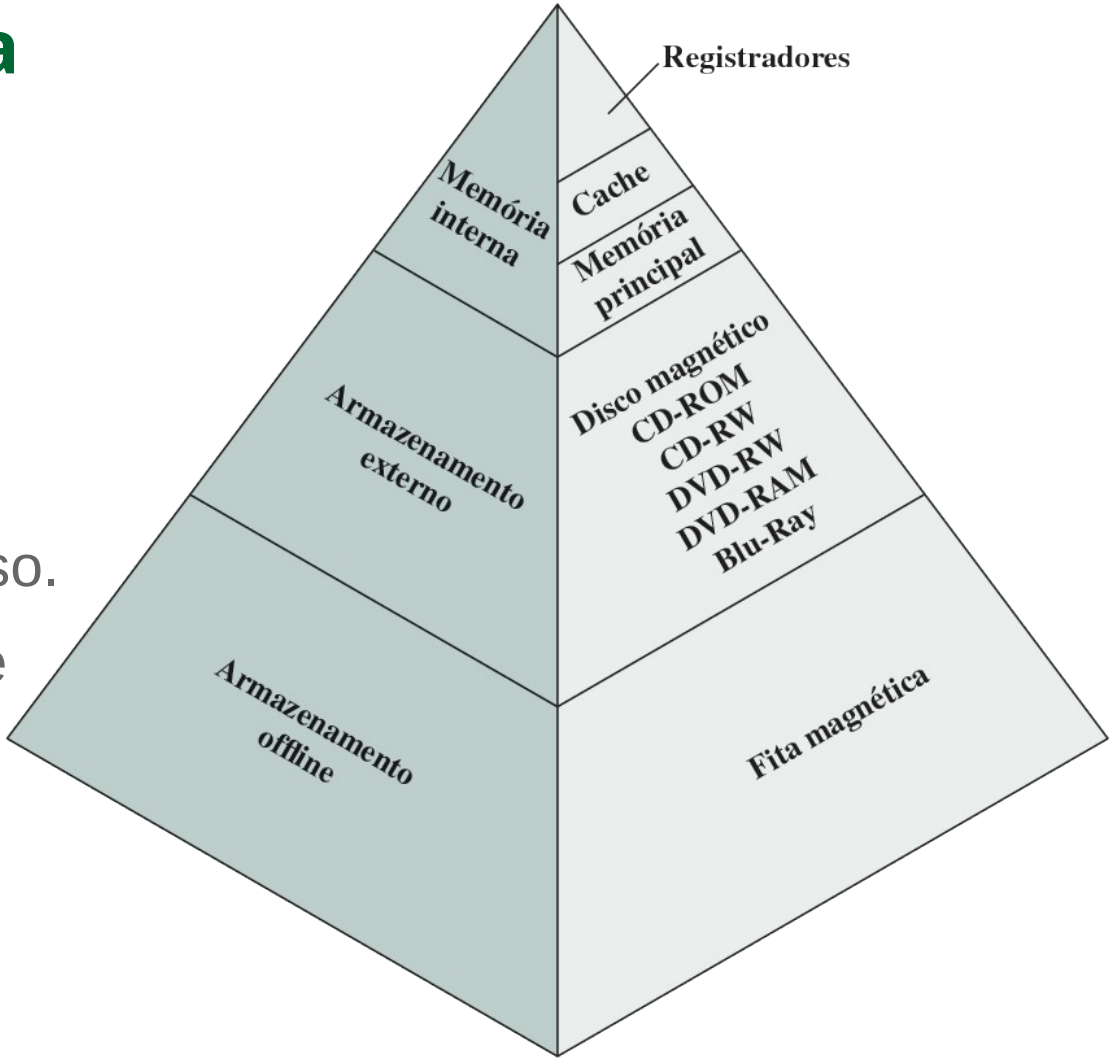
A hierarquia de memória

- Para sair desse **dilema**, é preciso não contar com um único componente ou tecnologia de memória, mas empregar uma hierarquia de memória.
- Uma hierarquia típica é ilustrada na figura:



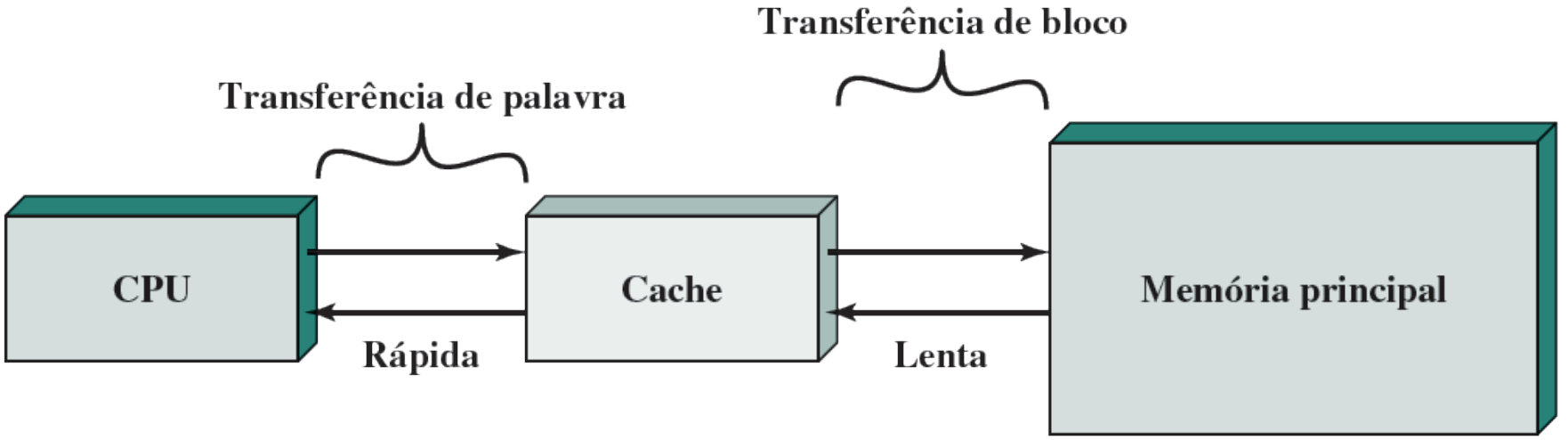
A hierarquia de memória

- Conforme se desce na hierarquia, ocorre o seguinte:
  i. Diminuição do custo por bit.
  ii. Aumento da capacidade.
  iii. Aumento do tempo de acesso.
  iv. Diminuição da frequência de acesso à memória pelo processador.



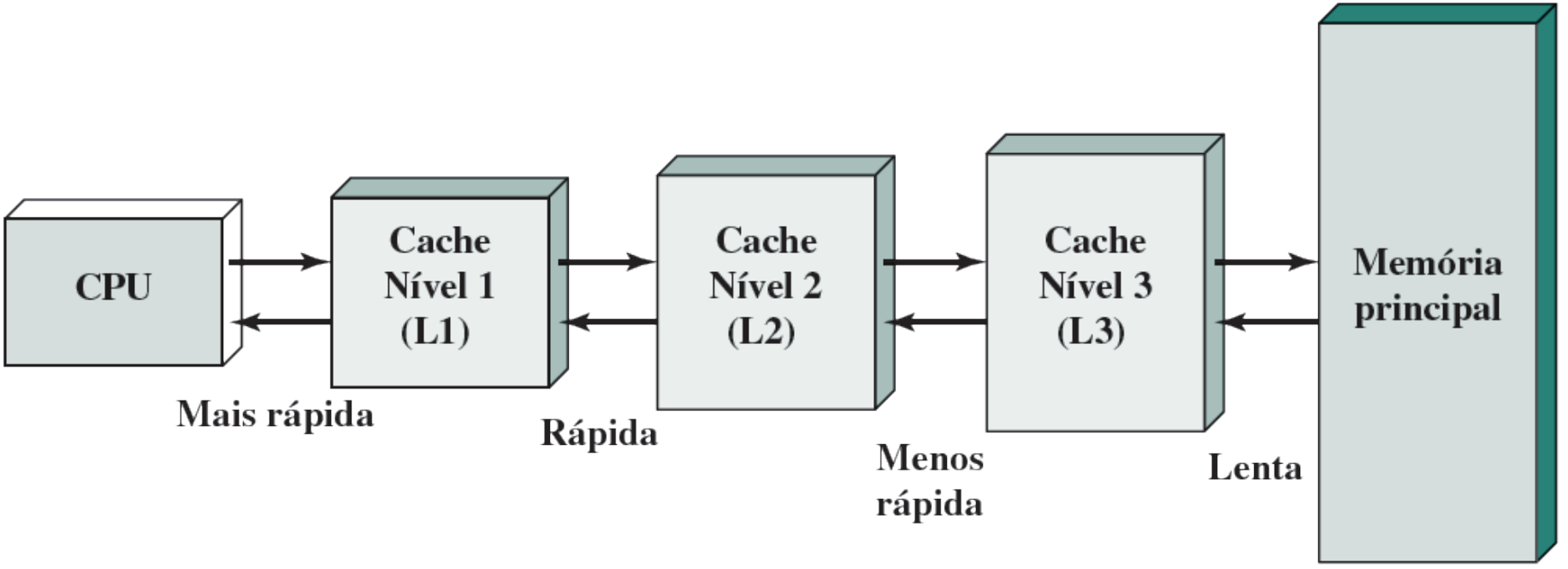
Princípios da memória cache

- A memória cache é desenvolvida para combinar o tempo de acesso de memórias de alto custo e alta velocidade com as memórias de menor velocidade, maior tamanho e mais baixo custo.



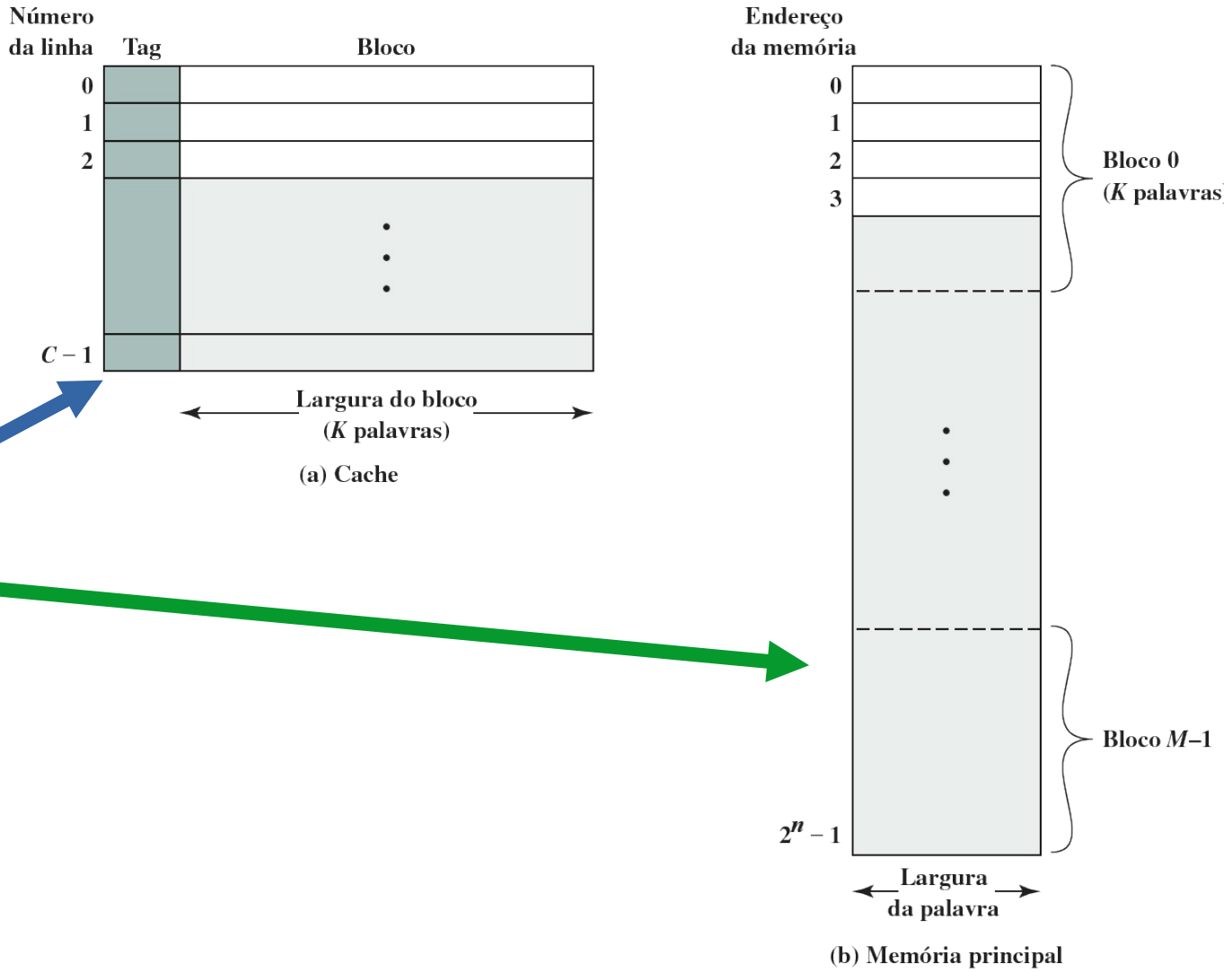
Princípios da memória cache

- A figura abaixo representa o uso de múltiplos níveis de cache:



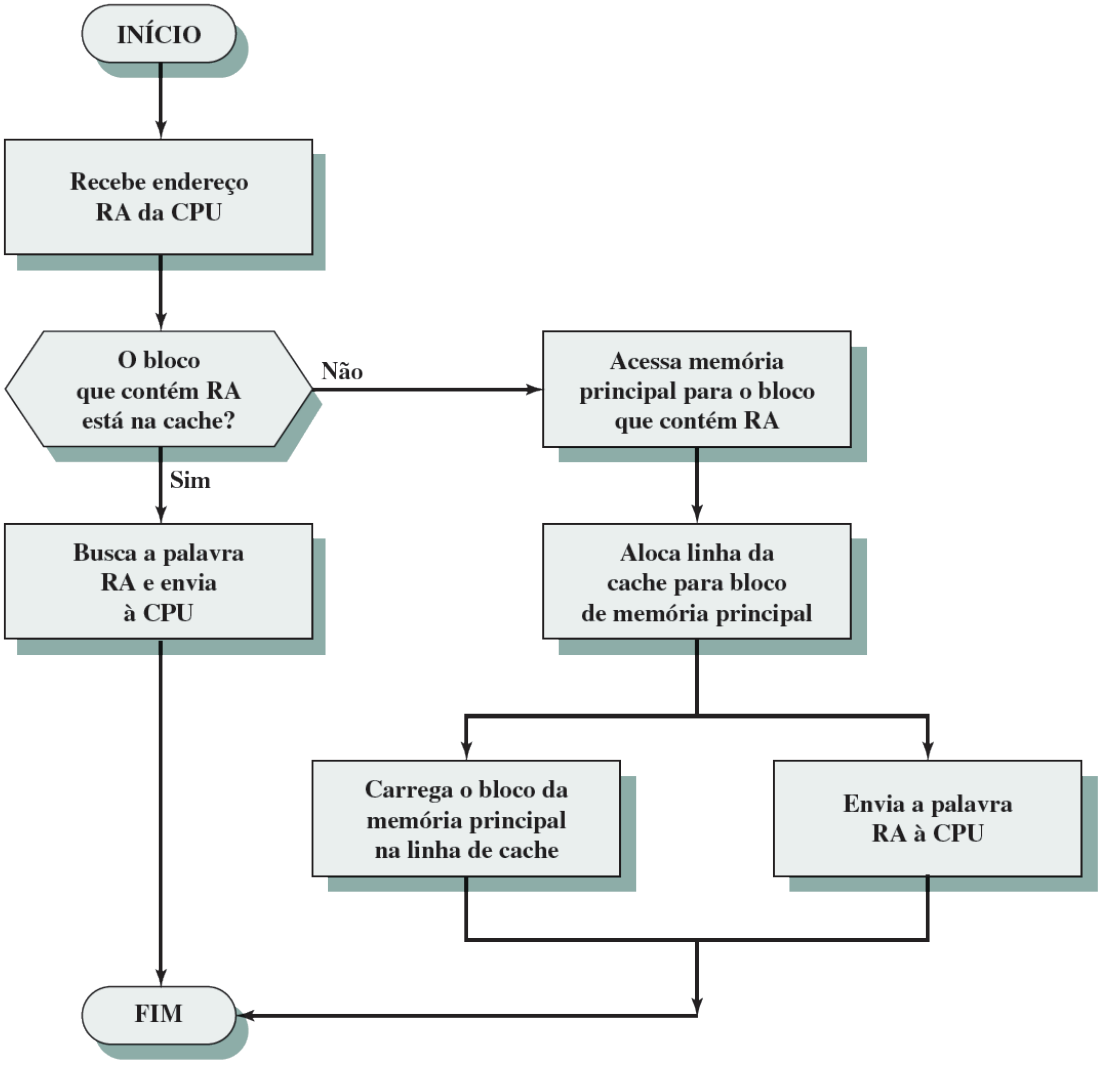
Princípios da memória cache

- Estrutura de cache / memória principal:



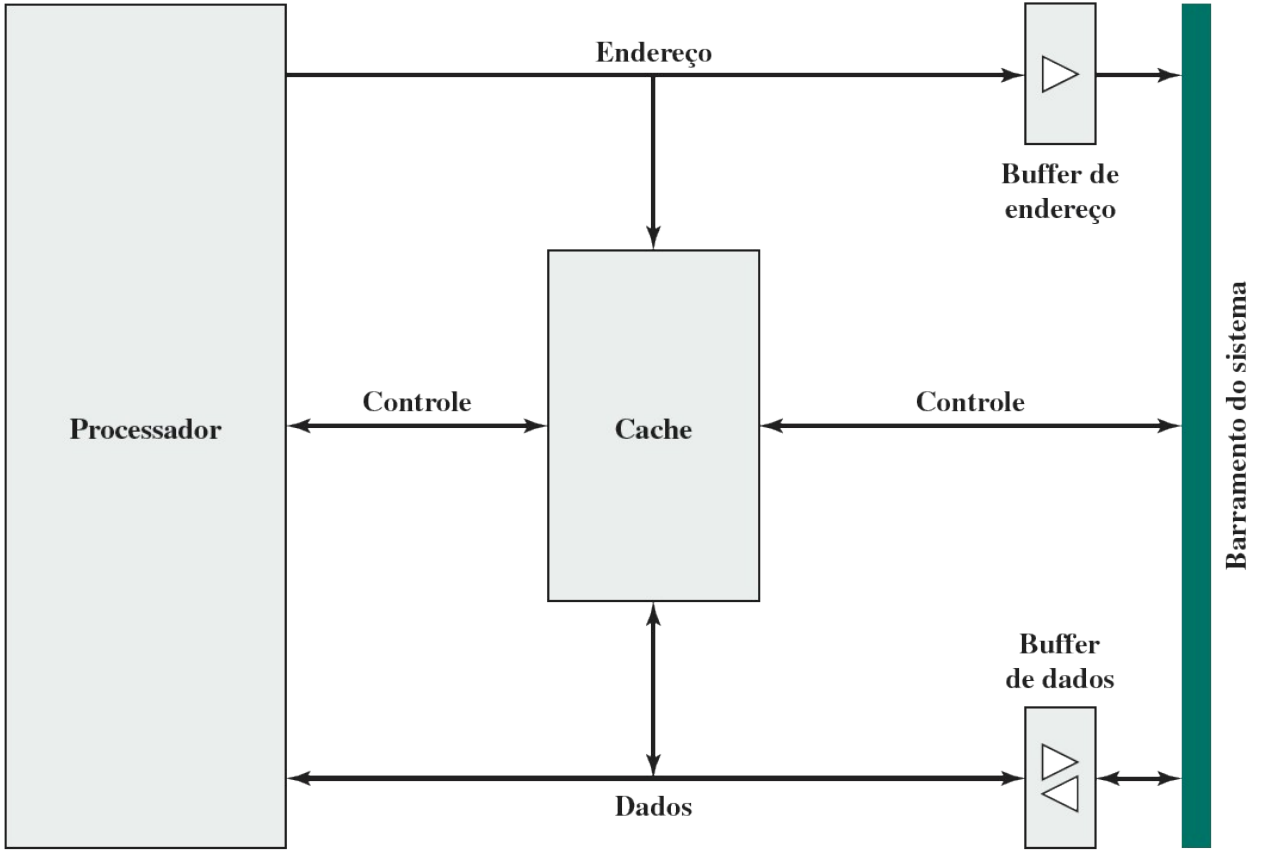
Princípios da memória cache

- Operação de leitura de cache:



Princípios da memória cache

- Organização típica da memória cache:



Elementos de projeto da cache

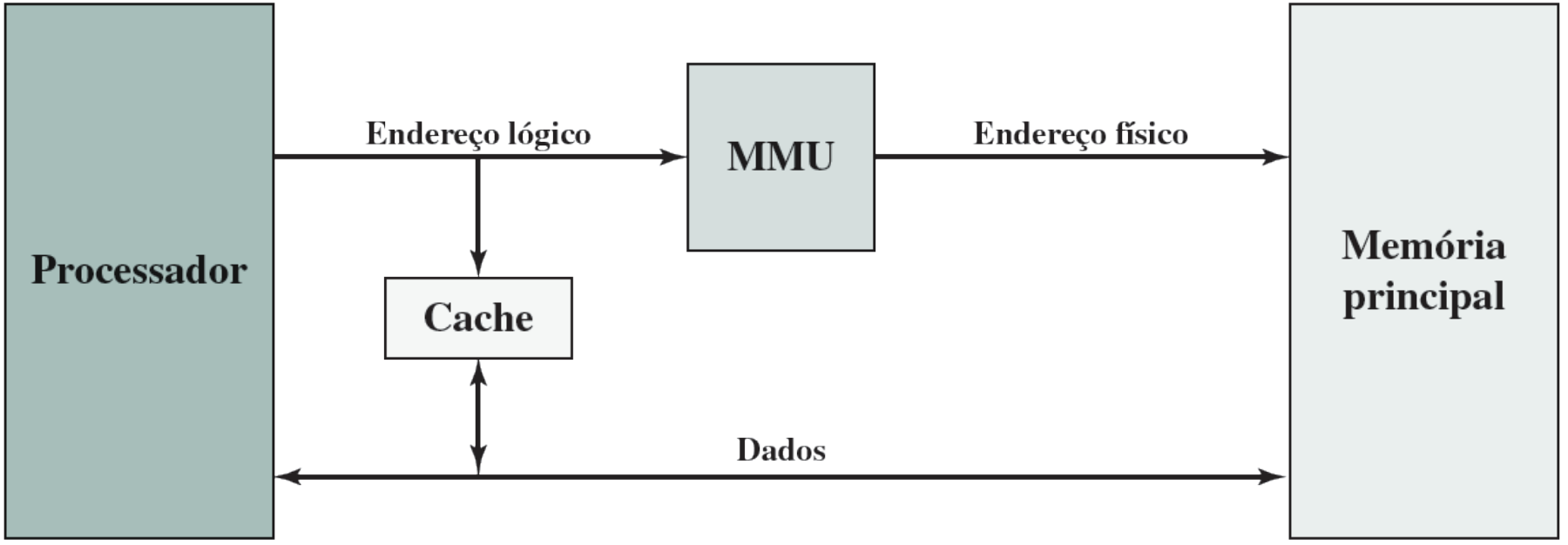
Endereços da cache	Política de escrita
Lógico	*Write through*
Físico	*Write back*
Tamanho da memória cache Função de mapeamento	**Tamanho da linha Número de caches**
Direto	Um ou dois níveis
Associativo	Unificada ou separada
Associativo em conjunto	
Algoritmo de substituição	
Usado menos recentemente (LRU — do inglês, *Least Recently Used*)	
Primeiro a entrar, primeiro a sair (FIFO — do inglês, *First In, First Out*)	
Usado menos frequentemente (LFU — do inglês, *Least Frequently Used*)	
Aleatória	

Endereços da cache

- Uma cache lógica, também conhecida como cache virtual, armazena dados usando endereços virtuais.
- Uma cache física armazena dados usando endereços físicos da memória principal.
- Uma vantagem da cache lógica é que a velocidade de acesso a ela é maior do que para uma cache física, pois a cache pode responder antes que a MMU realize uma tradução de endereço.
- A desvantagem é que a maioria dos sistemas de memória virtual fornece, a cada aplicação, o mesmo espaço de endereços de memória virtual.

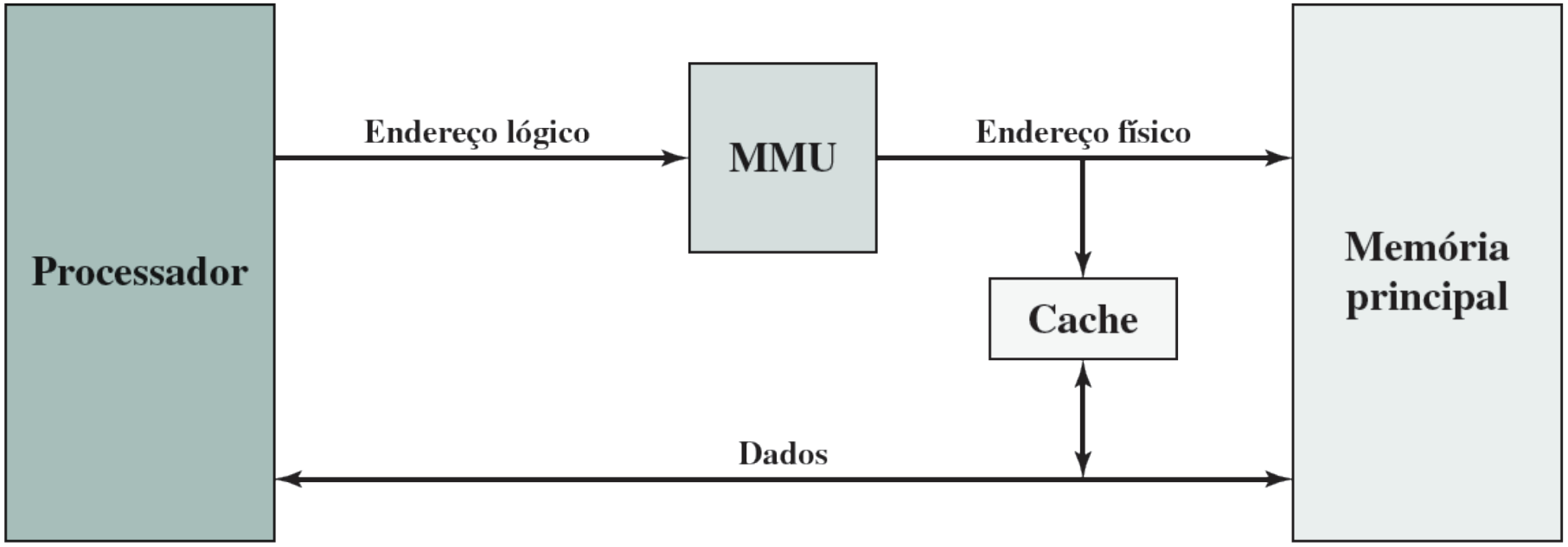
Endereços da cache

- Cache lógica:



Endereços da cache

- Cache física:



Tamanho da memória cache

- Quanto maior a cache, maior o número de portas envolvidos no endereçamento da cache.
- O resultado é que caches grandes tendem a ser ligeiramente mais lentas que as pequenas — mesmo quando construídas com a mesma tecnologia de circuito integrado e colocadas no mesmo lugar no chip e na placa de circuito.
- A área disponível do chip e da placa limita o tamanho da cache.
- Como o desempenho da cache é muito sensível à natureza da carga de trabalho, é impossível chegar a um único tamanho ideal de cache.

Função de mapeamento

- É necessário haver um algoritmo para mapear os blocos da memória principal às linhas de cache.
- É preciso haver um meio para determinar qual bloco da memória principal atualmente ocupa uma linha da cache.
- A escolha da função de mapeamento dita como a cache é organizada. Três técnicas podem ser utilizadas:
  i. direta,
  ii. associativa e
  iii. associativa por conjunto.

Função de mapeamento

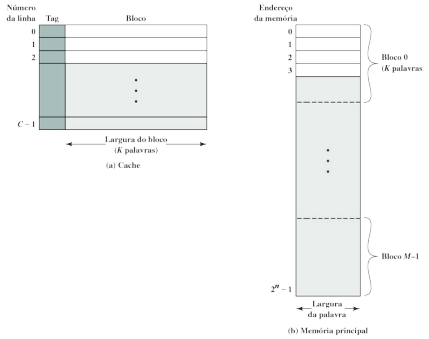
- Mapeamento direto é a técnica que mapeia cada bloco da memória principal a apenas uma linha de cache possível.
- O mapeamento é expresso como:

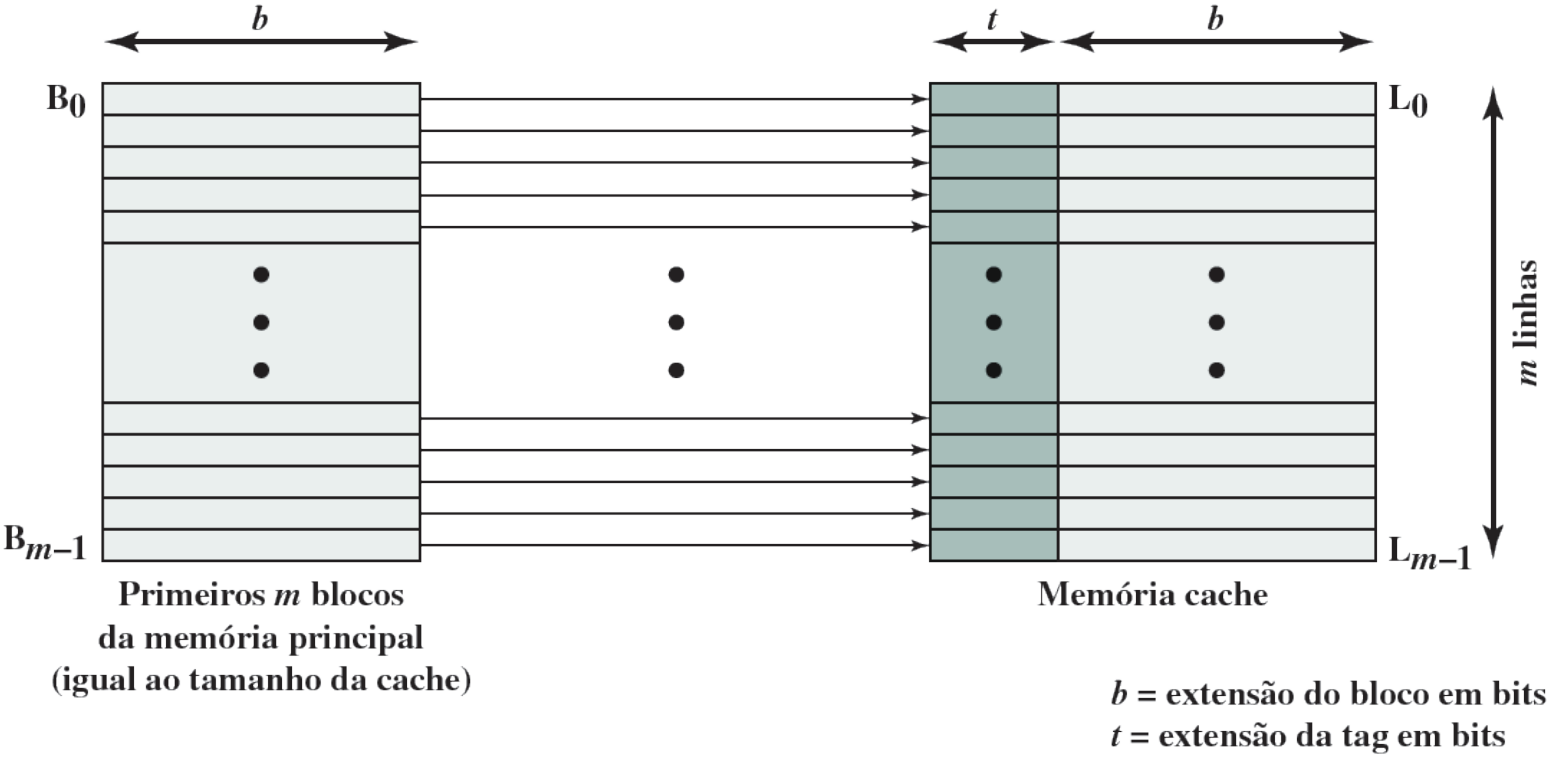
$$i = j \text{ módulo } m \qquad (i = j \,\%\, m)$$

- em que:
  - i = número da linha da cache
  - j = número do bloco da memória principal
  - m = número de linhas da cache

Função de mapeamento

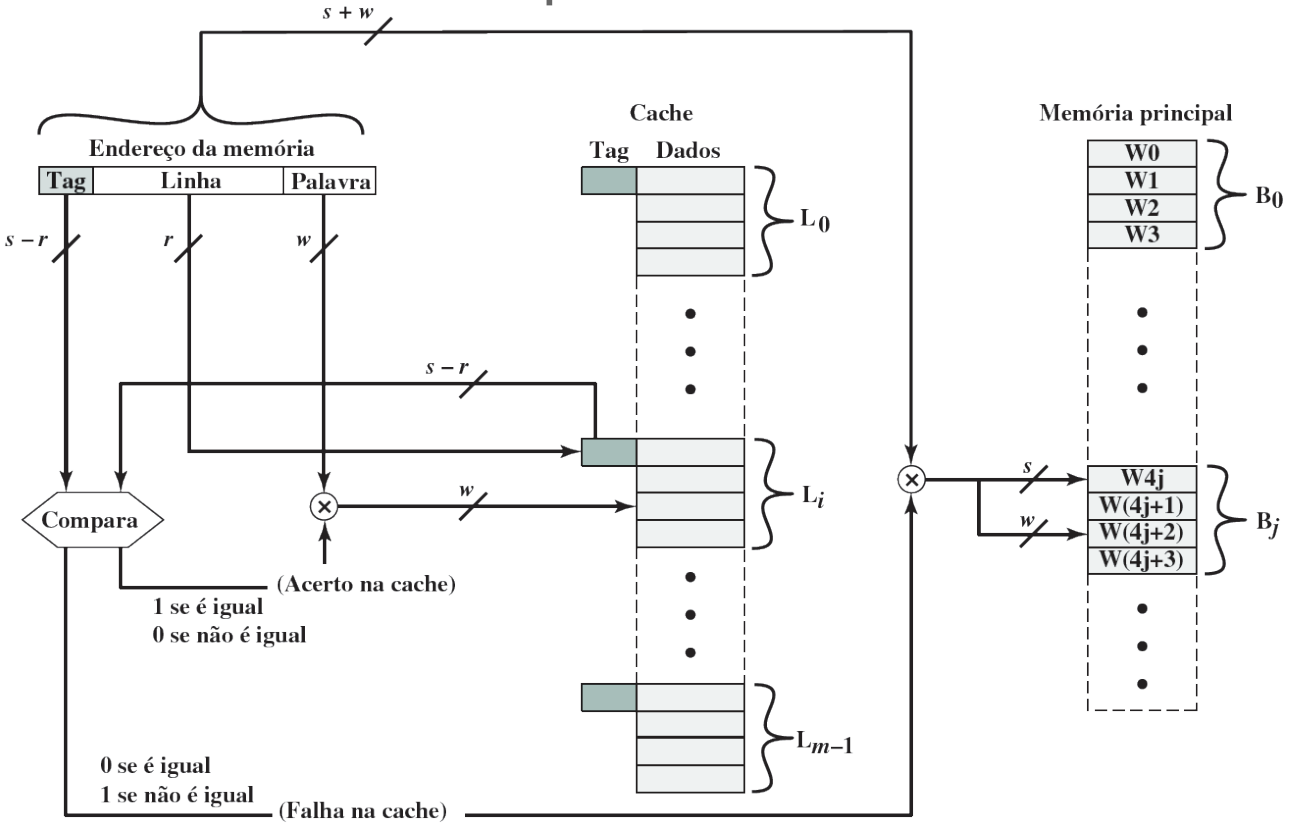
- Mapeamento da memória principal para a cache – direto:





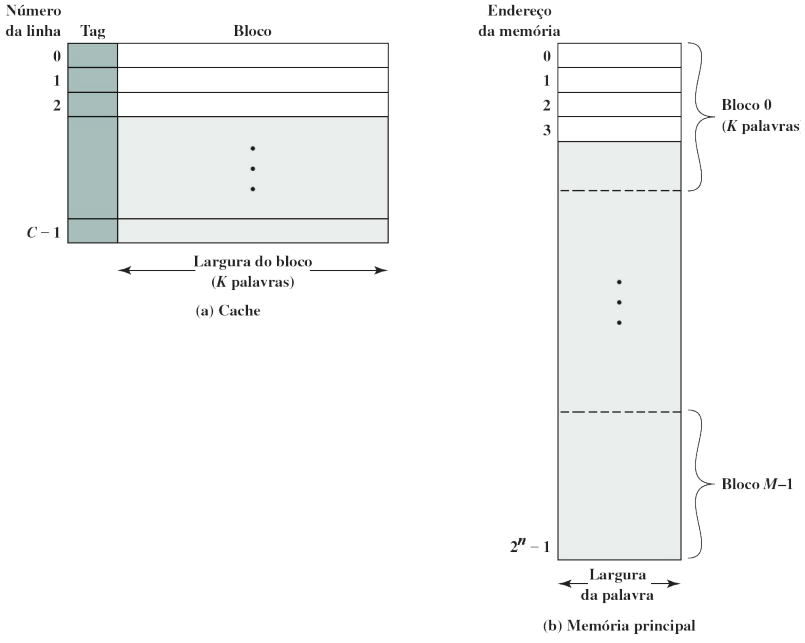
Função de mapeamento

- Organização de cache com mapeamento direto:

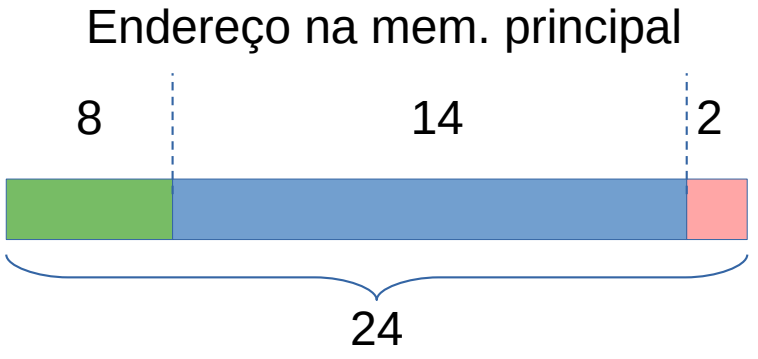


Função de mapeamento

- Mapeamento direto. Ex.: Mem = 16 MB; Cache = 64 kB; Blocos = 4 B

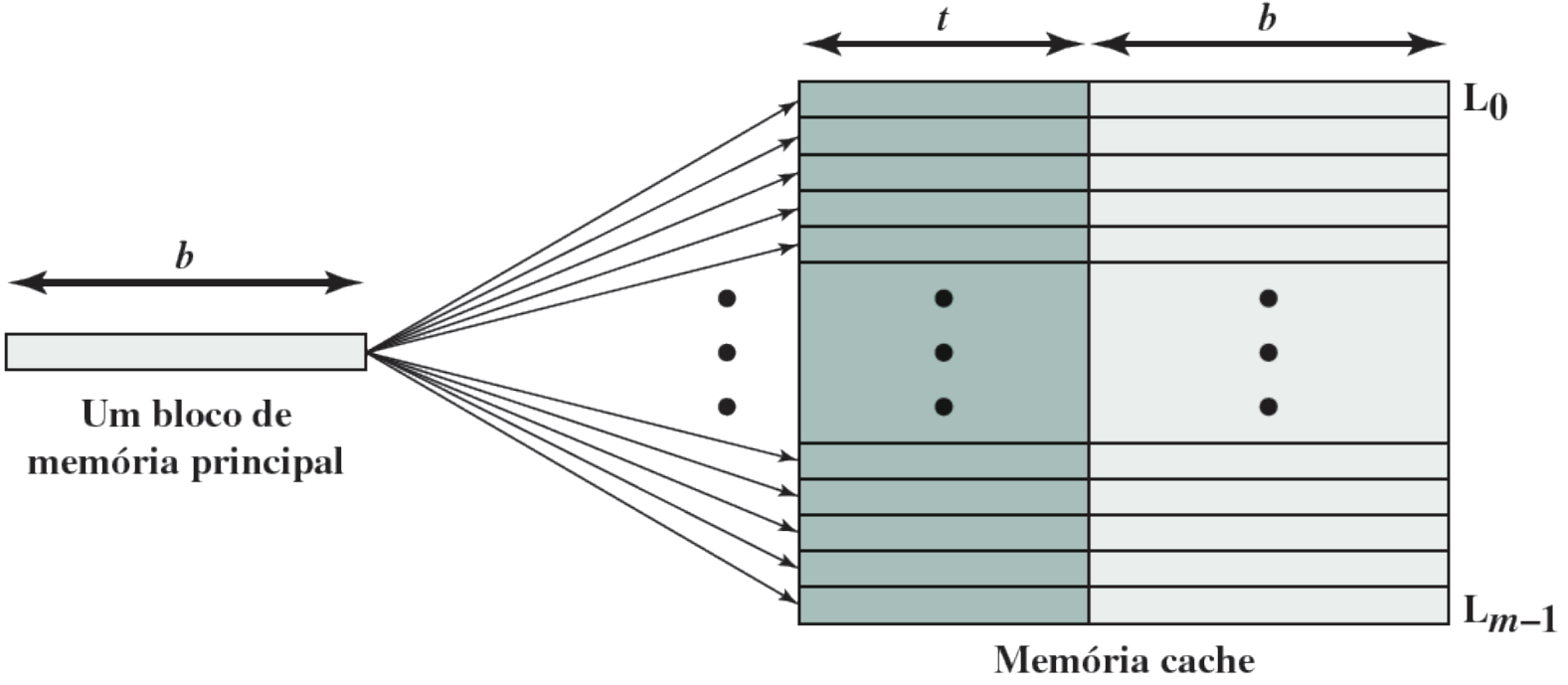


$$\text{Mem} = 2^{24}\ \text{B} \Rightarrow 2^{22}\ \text{Blocos}$$
$$\text{Cache} = 2^{16}\ \text{B} \Rightarrow 2^{14}\ \text{Linhas}$$



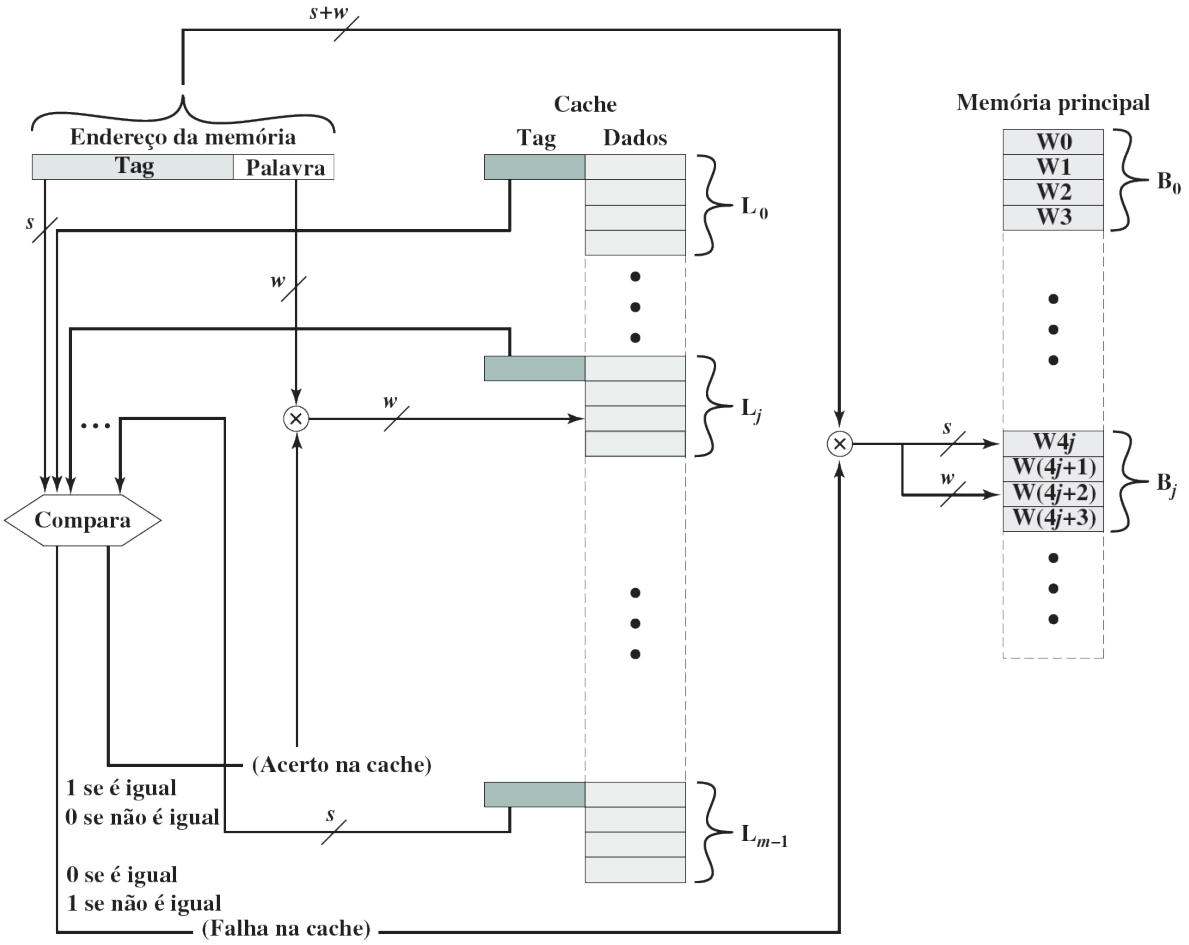
Função de mapeamento

- O mapeamento associativo compensa a desvantagem do mapeamento direto, permitindo que cada bloco da memória principal seja carregado em qualquer linha da cache:



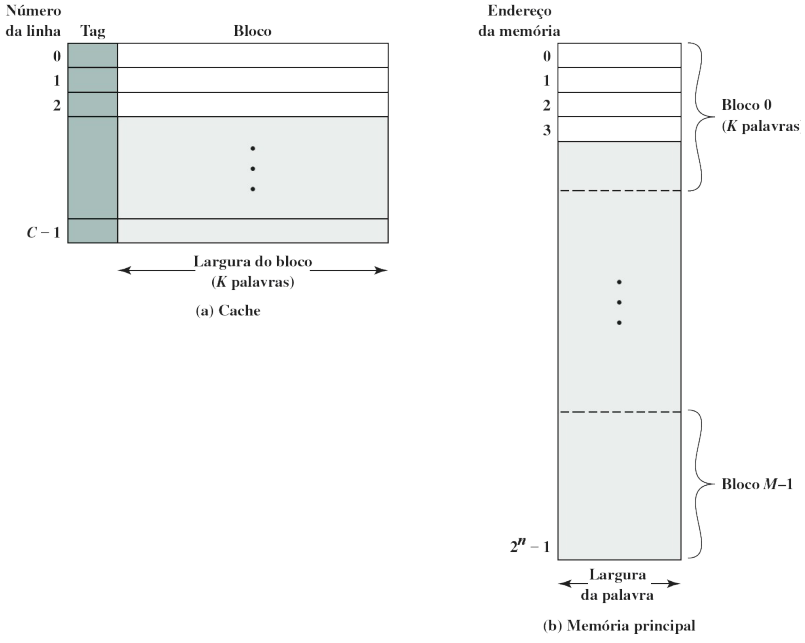
Função de mapeamento

- Organização da memória cache totalmente associativa:

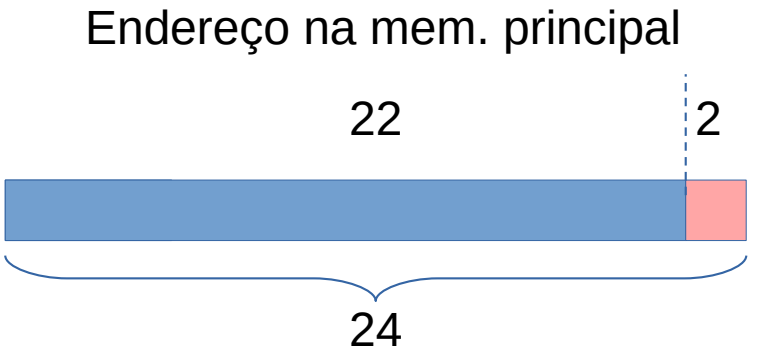


Função de mapeamento

- Mapeamento associativo. Ex.: Mem = 16 MB; Cache = 64 kB; Blocos = 4 B



$$\text{Mem} = 2^{24}\ \text{B} \Rightarrow 2^{22}\ \text{Blocos}$$
$$\text{Cache} = 2^{16}\ \text{B} \Rightarrow 2^{14}\ \text{Linhas}$$



Função de mapeamento

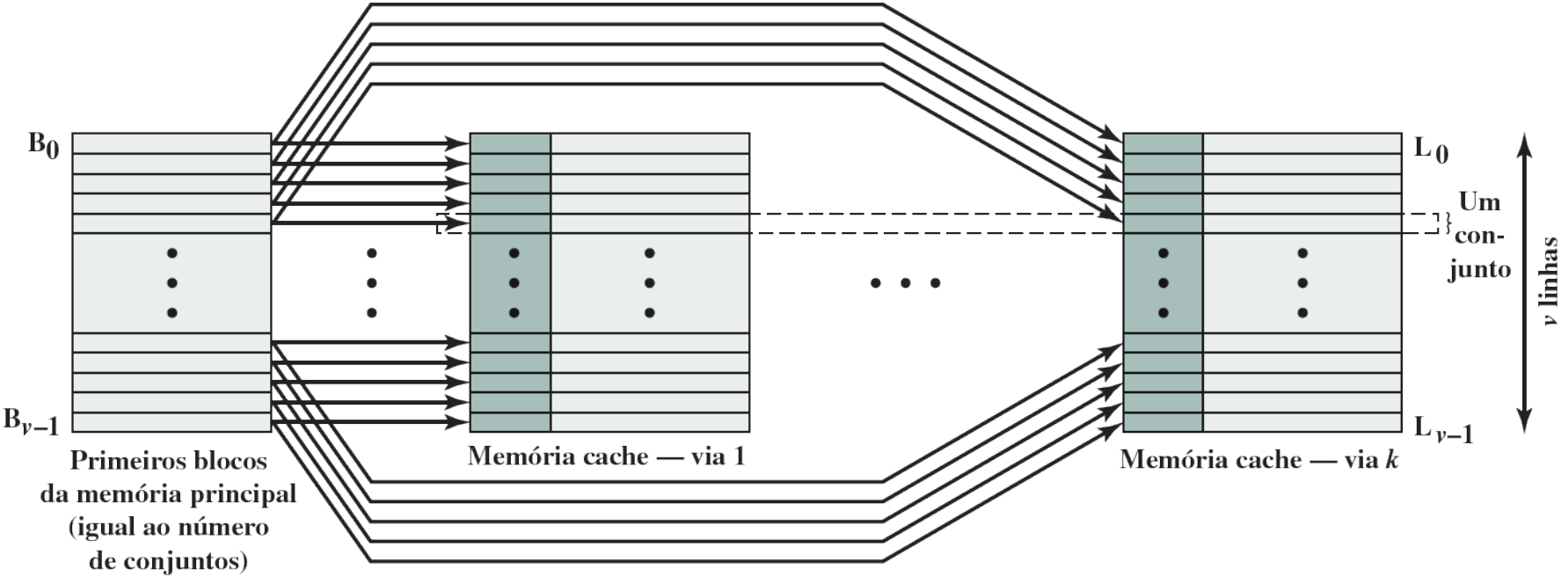
- O mapeamento associativo por conjunto é um meio-termo que realça os pontos fortes das técnicas direta e associativa, enquanto reduz suas desvantagens.
- Neste caso, a cache é uma série de conjuntos, cada um consistindo em uma série de linhas. As relações são:

$$m = v * k$$
$$i = j \text{ módulo } v$$

  - i = número do conjunto de cache
  - j = número de bloco da memória principal
  - m = número de linhas na cache
  - v = número de conjuntos
  - k = número de linhas em cada conjunto

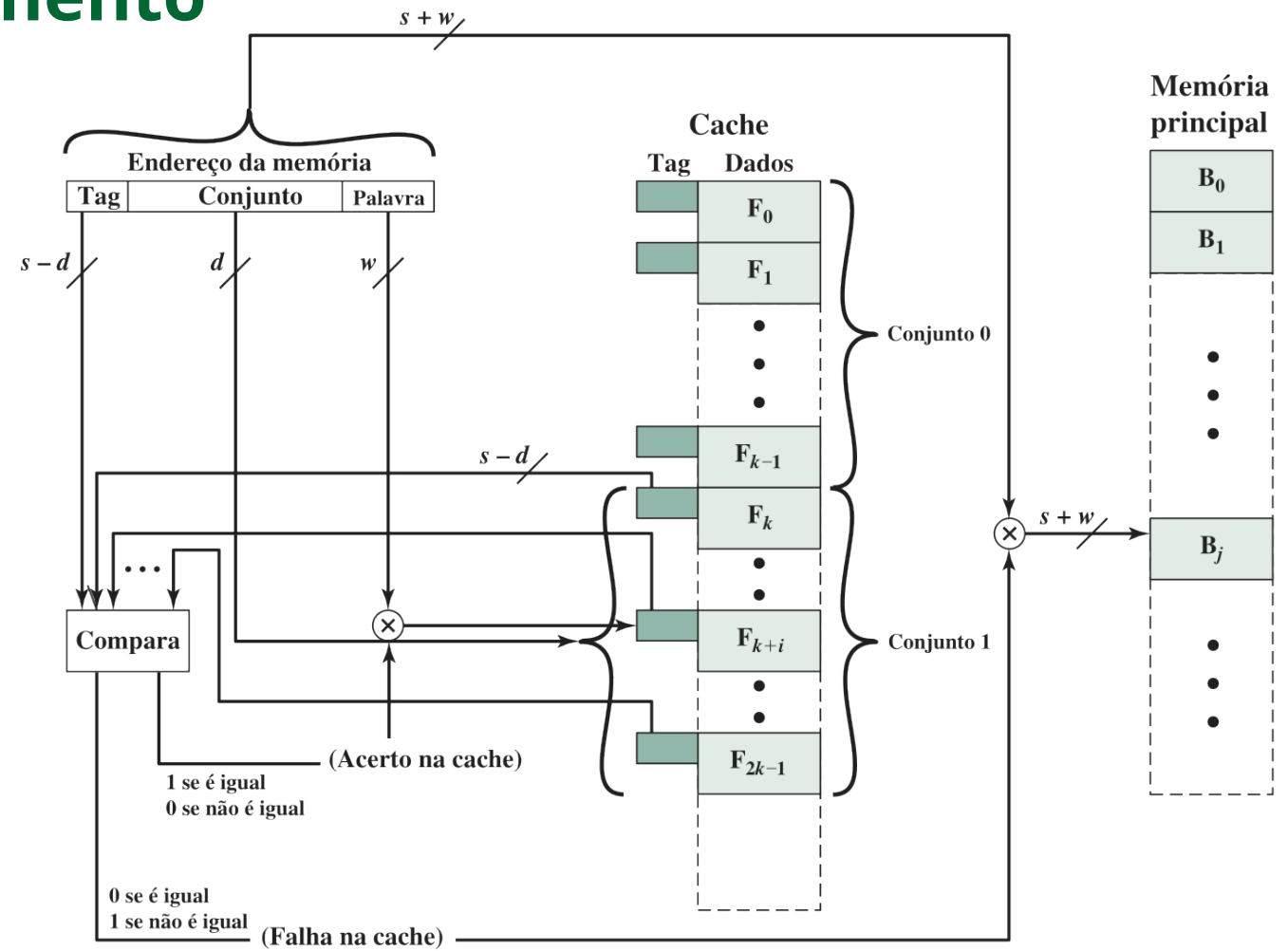
Função de mapeamento

- Também é possível implementar a cache associativa em conjunto como k caches de mapeamento direto:



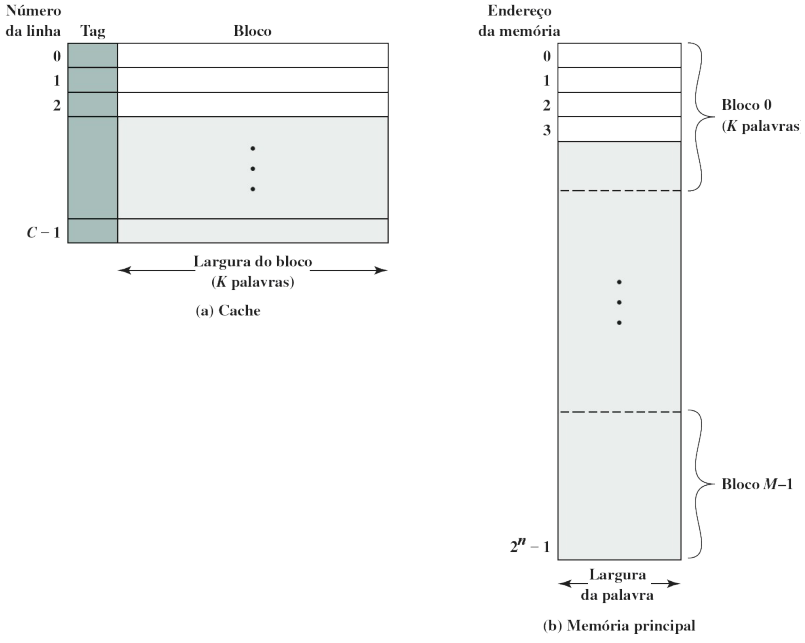
Função de mapeamento

- Organização da memória cache associativa por conjuntos:



Função de mapeamento

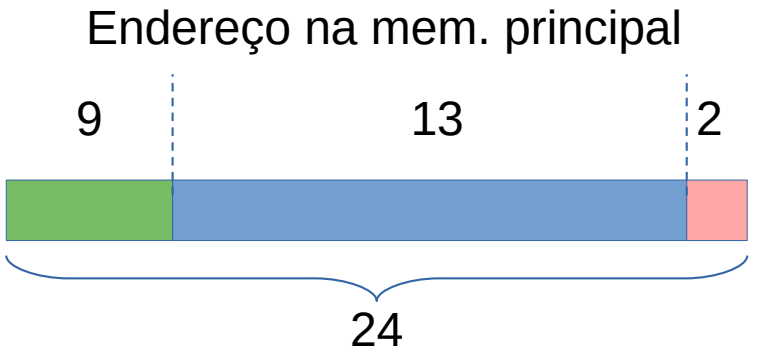
- Mapeamento assoc. conj. Ex.: Mem = 16 MB; Cache = 64 kB; Blocos = 4 B



$\text{Mem} = 2^{24}\ \text{B} \Rightarrow 2^{22}\ \text{Blocos}$

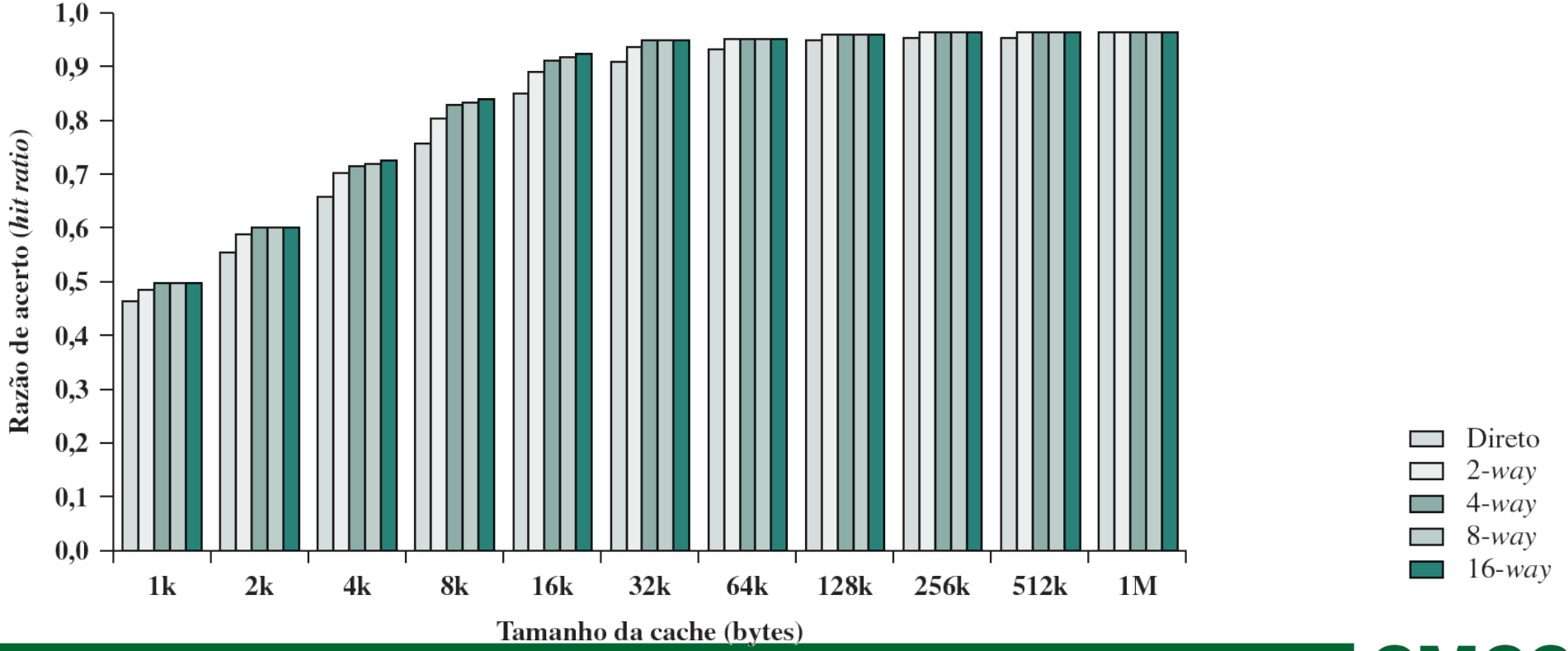
$\text{Cache} = 2^{16}\ \text{B} \Rightarrow 2^{14}\ \text{Linhas}$

$2\ \text{linhas por conjunto}\ (2^{13}\ \text{conj.})$



Função de mapeamento

- Associatividade variável pelo tamanho da cache:



Algoritmos de substituição

- Uma vez que a cache esteja cheia, e um novo bloco seja trazido para a cache, um dos blocos existentes precisa ser substituído.
- Para as técnicas associativa e associativa em conjunto, um algoritmo de substituição é necessário.
- O mais eficaz é o usado menos recentemente (**LRU**).
- Outra possibilidade é “primeiro a entrar, primeiro a sair” (**FIFO**).
- Outra possibilidade de algoritmo, ainda, é o usado menos frequentemente (**LFU**).
- Outra possibilidade é a substituição aleatória (desempenho apenas ligeiramente inferior aos anteriores).

Política de escrita

- Quando um bloco que está residente na cache estiver para ser substituído, existem dois casos a serem considerados.
- Se o bloco antigo na cache não tiver sido alterado, ele pode ser substituído por novo bloco sem primeiro atualizar o bloco antigo.
- Se pelo menos uma operação de escrita tiver sido realizada em uma palavra nessa linha da cache, então a memória principal precisa ser atualizada escrevendo a linha de cache no bloco de memória antes de trazer o novo bloco.
- Diversas políticas de escrita são possíveis, com escolhas econômicas e de desempenho.

Política de escrita

- A técnica mais simples é denominada write through.
- Usando esta técnica, todas as operações de escrita são feitas na memória principal e também na cache, garantindo que a memória principal sempre seja válida.
- Numa técnica alternativa, conhecida como write back, as atualizações são feitas apenas na cache.
- Em uma organização de barramento em que mais de um dispositivo (em geral, um processador) tem uma cache e a memória principal é compartilhada, um novo problema é introduzido.

Política de escrita

- A técnica mais simples é denominada write through.
- Usando esta técnica, todas as operações de escrita são feitas na memória principal e também na cache, garantindo que a memória principal sempre seja válida.
- Numa técnica alternativa, conhecida como write back, as atualizações são feitas apenas na cache.
- Em uma organização de barramento em que mais de um dispositivo (em geral, um processador) tem uma cache e a memória principal é compartilhada, um novo problema é introduzido.

E.: Qual a desvantagem da técnica write through?

Política de escrita

- Se os dados em uma cache forem alterados, isso invalida não apenas a palavra correspondente na memória principal, mas também essa mesma palavra em outras caches (se qualquer outra cache tiver essa mesma palavra).
- Mesmo que uma política write through seja usada, as outras caches podem conter dados inválidos.
- Diz-se que um sistema que impede esse problema mantém coerência de cache.

Política de escrita

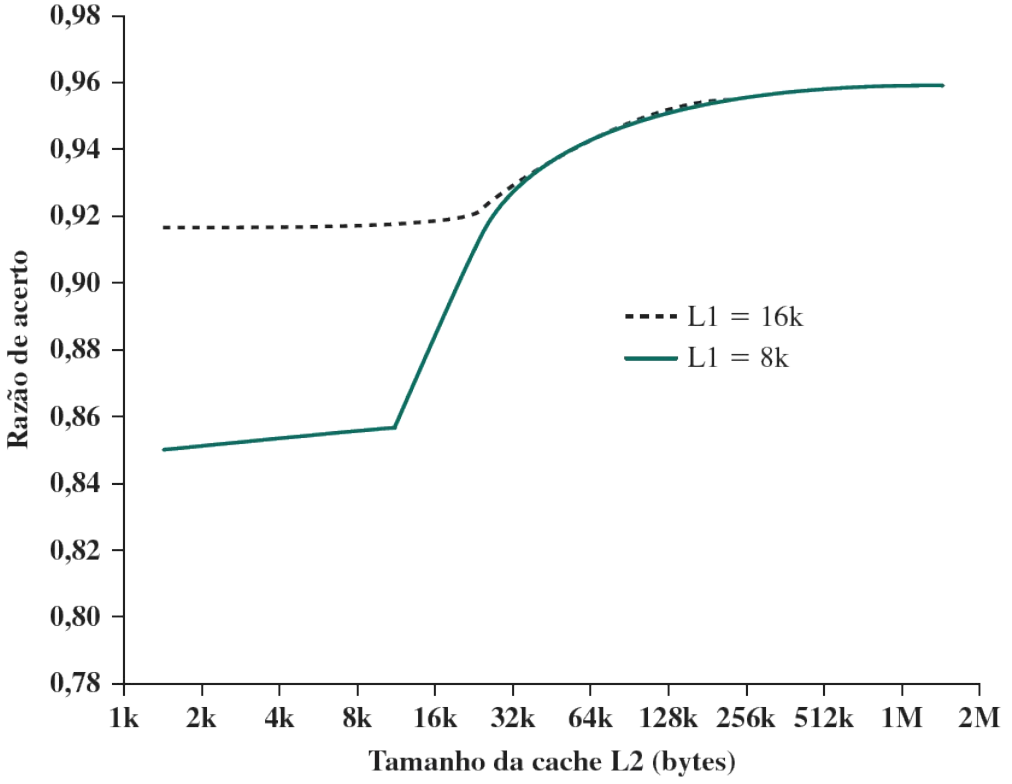
- Algumas das técnicas possíveis para a coerência de cache são:
  - **Observação do barramento com write through**: cada controlador de cache monitora as linhas de endereço para detectar as operações de escrita para a memória por outros mestres de barramento.
  - **Transparência do hardware**: um hardware adicional é usado para garantir que todas as atualizações na memória principal por meio da cache sejam refletidas em todas as caches.
  - **Memória não cacheável**: somente uma parte da memória principal é compartilhada por mais de um processador, e esta é designada como não cacheável.

Tamanho da linha

- À medida que o tamanho do bloco aumenta, a razão de acerto a princípio aumentará por causa do princípio da localidade, que diz que os dados nas vizinhanças de uma palavra referenciada provavelmente serão referenciados no futuro próximo.
- À medida que o tamanho do bloco aumenta, dados mais úteis são trazidos para a cache.
- Contudo, a razão de acerto começará a diminuir enquanto o bloco se torna ainda maior e a probabilidade de uso da informação recém-trazida se torna menor que a probabilidade de reutilizar as informações que foram substituídas.

Número de caches

- Razão de acerto total (L1 e L2) para L1 de 8 kB e 16 kB:



Número de caches

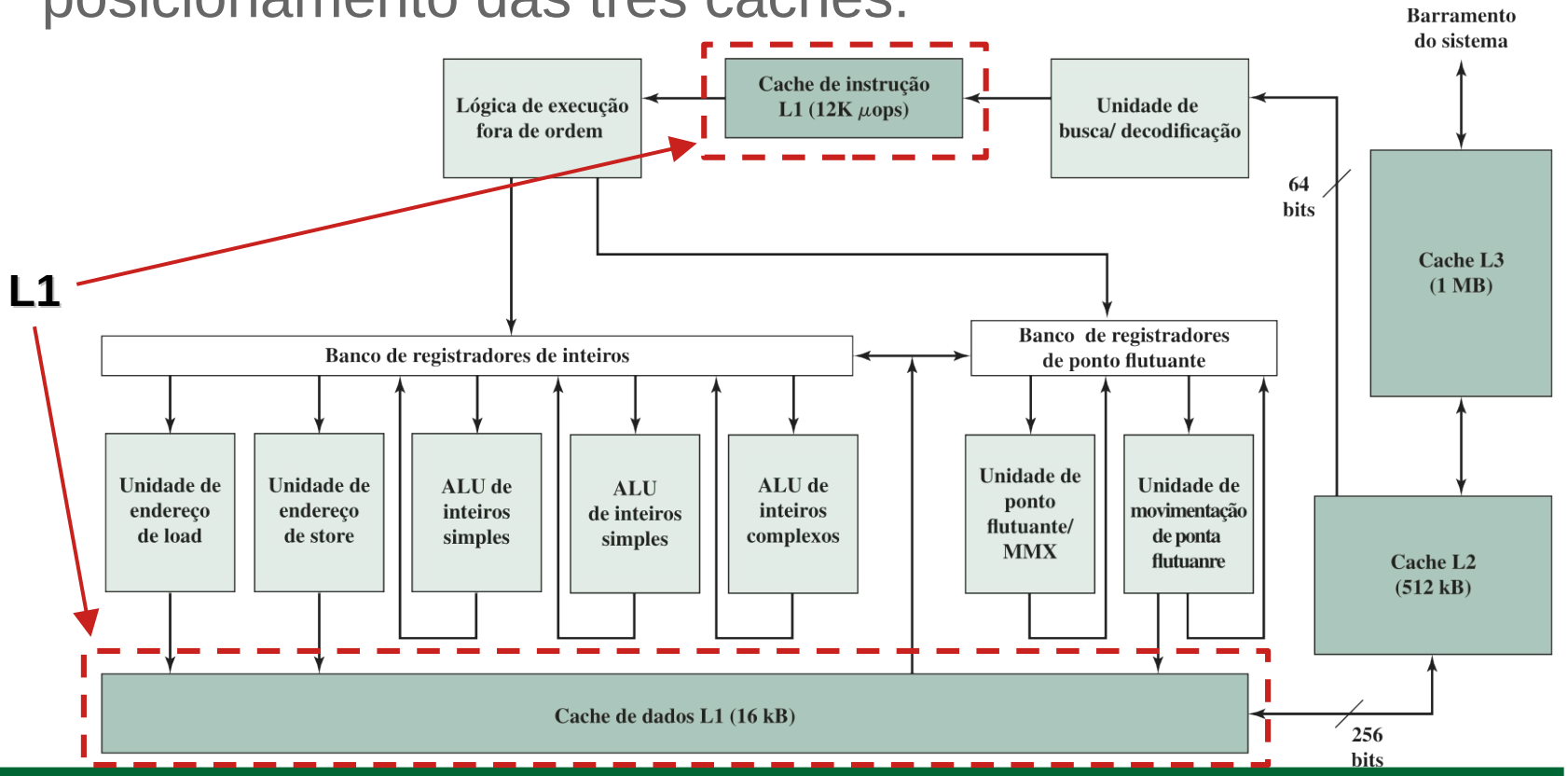
- Caches multinível
  - À medida que a densidade lógica aumenta, torna-se possível ter uma cache no mesmo chip que o processador: a cache no chip.
  - A cache no chip reduz a atividade do barramento externo do processador e, portanto, agiliza o tempo de execução e aumenta o desempenho geral do sistema.
  - A organização mais simples desse tipo é conhecida como uma cache de dois níveis, com a cache interna designada como nível 1 (L1) e a cache externa designada como nível 2 (L2).

Número de caches

- Caches unificadas versus separadas
  - Mais recentemente, tornou-se comum dividir a cache em duas: uma dedicada a instruções e uma dedicada a dados.
  - Essas duas caches existem no mesmo nível, normalmente como duas caches L1.
  - Quando o processador tenta buscar uma instrução da memória principal, ele primeiro consulta a cache L1 de instrução.
  - Quando o processador tenta buscar dados da memória principal, ele primeiro consulta a cache L1 de dados.

Organização da cache do pentium 4

- Visão simplificada da organização do Pentium 4, destacando o posicionamento das três caches:



Organização da cache do pentium 4

- Modos de operação da cache do Pentium 4:

Bits de controle		Modo de operação		
CD	NW	Preenchimento da cache	*Write throughs*	Invalidado
0	0	Habilitado	Habilitado	Habilitado
1	0	Desabilitado	Habilitado	Habilitado
1	1	Desabilitado	Desabilitado	Desabilitado

Obs.: CD = 0; NW = 1 é uma combinação inválida.