Matheus Albuquerque Gameiro de Moura
Everaldo Faustino dos Santos Junior
Daniel Carlos Junior

Memória Interna - Focado em Cache

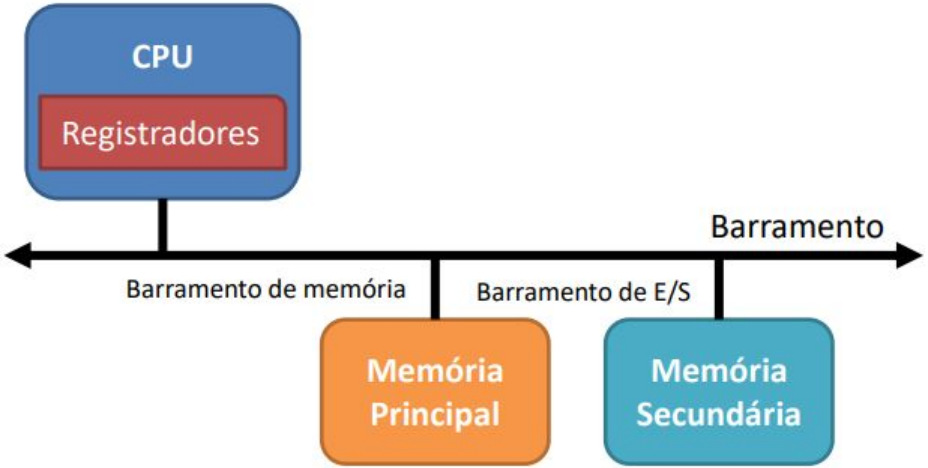
ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

SISTEMA DE MEMÓRIA DO COMPUTADOR

- O sistema de memória de um computador, pode ser mais facilmente compreendido por meio de sua características.
  - Localização
  - Capacidade
  - Unidade de transferência
  - Método de acesso
  - Desempenho

SISTEMA DE MEMÓRIA DO COMPUTADOR
LOCALIZAÇÃO

- Memória interna
- Memória externa



SISTEMA DE MEMÓRIA DO COMPUTADOR - CAPACIDADE

- Palavra
  - Expressa em função de bytes.
  - Tamanho da palavra.
  - Número de palavras.

  - Na memória interna é expressa em byte ou palavras
    - Ordens de grandeza: 10^3= Kb (cache L1); 106= Mb (cache L2); e 10^9= Gb (memória principal).
    -
  - Na memória externa é expressa em byte.
    - Ordens de grandeza: 10^6= Mb; 10^9= Gb e 10^12= Tb

SISTEMA DE MEMÓRIA DO COMPUTADOR - UNIDADE DE TRANSFERÊNCIA

- Unidade de transferência de dados **corresponde ao nº de bits que podem ser lidos ou escritos de cada vez.**

- **Memória interna:** a unidade de transferência é governada pela largura do barramento de dados.
  - Normalmente o nº de linhas de dados = tamanho da palavra
  - Internamente, o endereçamento é feito por palavras.
  -
- **Memória externa:** a unidade de transferência é feita por blocos de dados.
  - Um bloco é muito maior que uma palavra (bloco >> palavra).
  - Em unidades de disco, o bloco é a unidade de endereçamento dos dados(clusters)

SISTEMA DE MEMÓRIA DO COMPUTADOR - MÉTODO DE ACESSO

- Sequencial: o acesso é feito seguindo uma seqüência linear específica.
- Direto: o acesso é feito por um **salto até um bloco** de registros, **seguido por uma pesquisa seqüencial** até o registro (posição) desejado.
- Aleatório: acesso é feito diretamente ao registro através de **seu endereço.**
- Associativo: acesso é feito diretamente ao registro com base em **parte de seu conteúdo.**

SISTEMA DE MEMÓRIA DO COMPUTADOR - DESEMPENHO

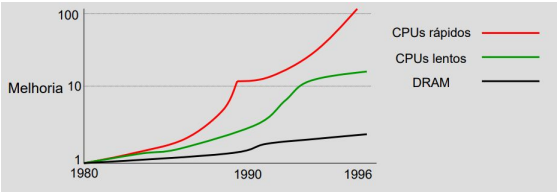
- **Tempo de acesso :** tempo necessário para localizar, ler ou escrever um dado na memória.

- **Tempo de ciclo:** tempo de acesso + tempo adicional requerido pela memória antes de iniciar o próximo acesso.

- **Taxa de transferência:** taxa na qual os dados podem ser movidos.
  - **Acesso aleatório:** 1/Tc onde: Tc é o tempo de ciclo.
  - **Acesso não-aleatório:** N/ (Tn-Ta) onde: Tn é o tempo médio de L/E de Nbits e Ta é o tempo médio de acesso.

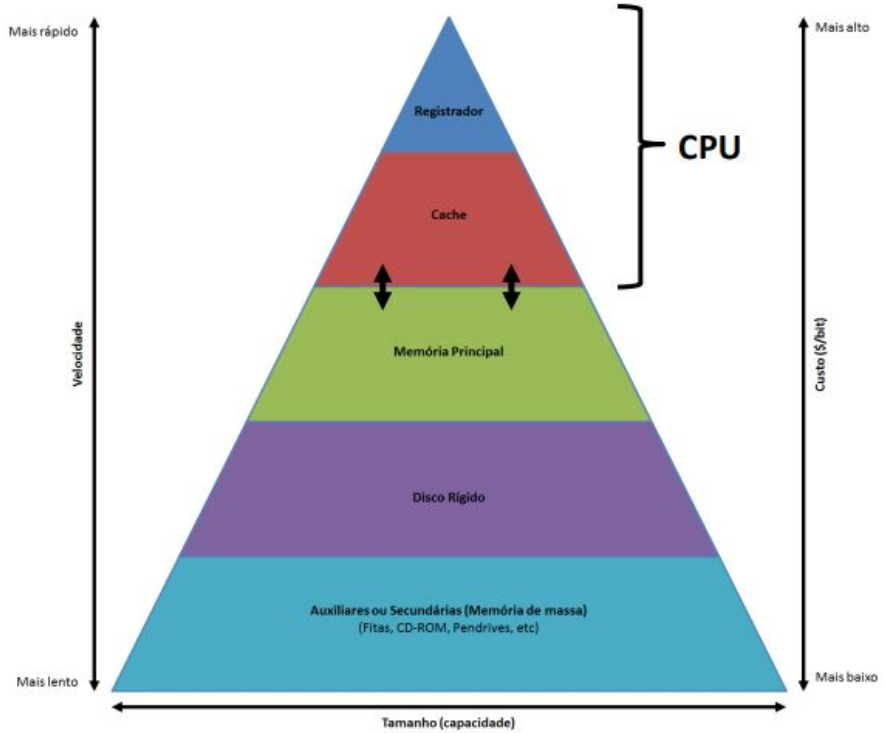
HIERARQUIA DE MEMÓRIAS

- As restrições de projeto de uma memória podem ser resumidas em três questões:
  - Quanto?
  - Com que velocidade?
  - A que custo?



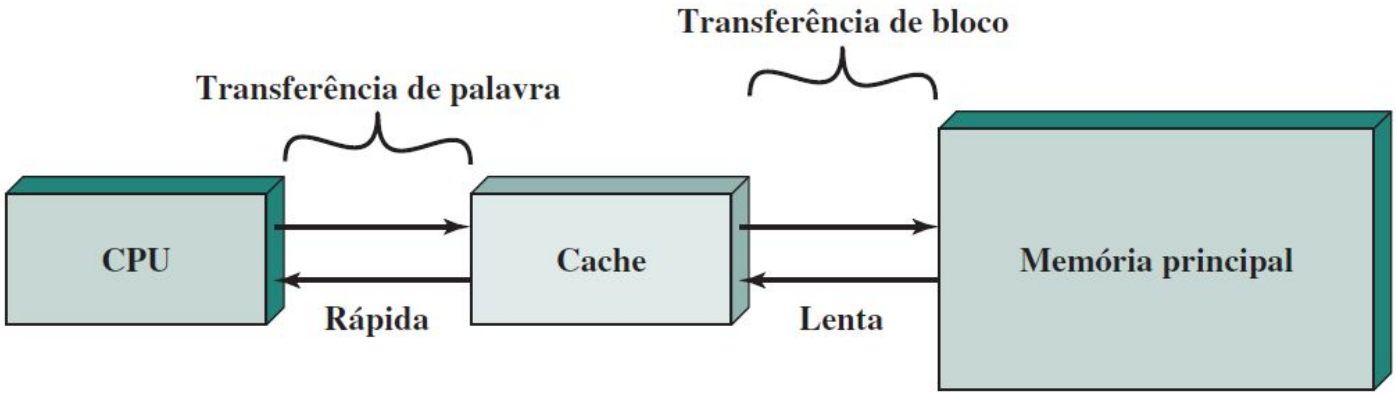
Tipo	Tempo de acesso	Custo
SRAM	0,5 ns à 2,5 ns	$ 2000,00 à $ 5000,00 por GB
DRAM	50 ns à 70 ns	$ 20,00 à $ 75,00 por GB
HD	5 ms à 20 ms	$ 0,20 à $ 2,00 por GB

HIERARQUIA DE MEMÓRIAS



MEMÓRIA CACHE

- Cache é um dispositivo interno a um sistema que serve de intermediário entre uma CPU e o dispositivo principal de armazenamento (MP).
- Memória Cache: memória pequena (capacidade de armazenamento) e rápida.



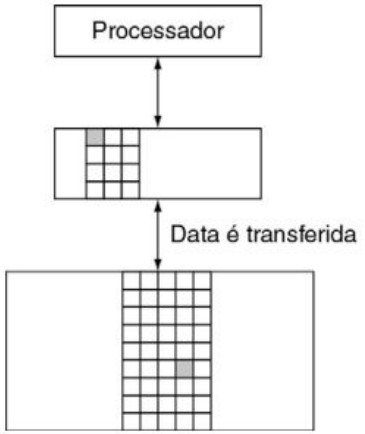
MEMÓRIA CACHE

- Localidade Temporal
  - Uma posição de memória referenciada recentemente tem boas chances de ser referenciada novamente em um futuro próximo Iterações e recursividade.

- Localidade Espacial
  - Uma posição de memória vizinha de uma posição referenciada recentemente tem boas chances de também ser referenciada. Dados tendem a ser armazenados em posições imediatas.

MEMÓRIA CACHE

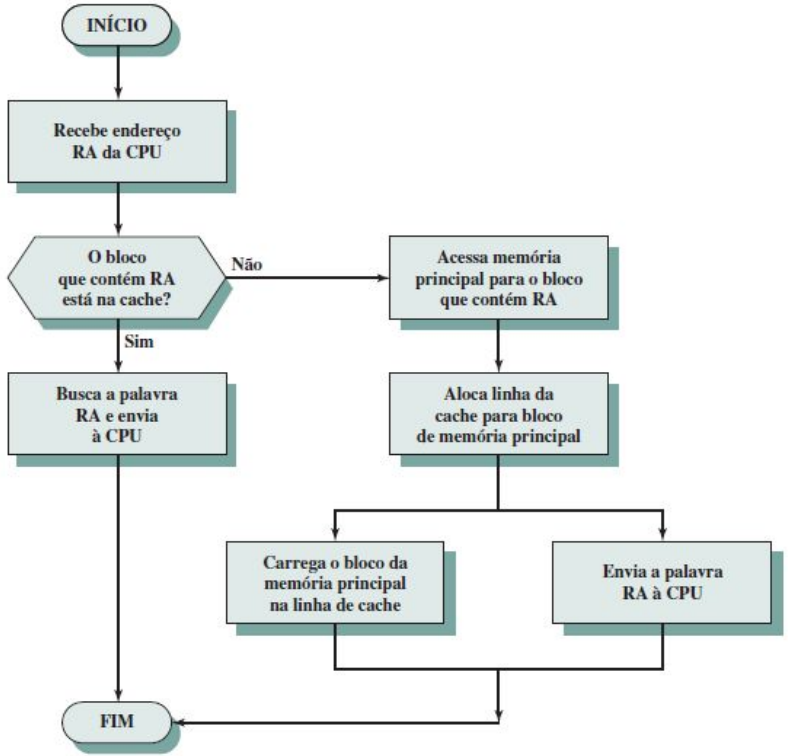
- Hit - dado encontrado no nível procurado.
- Miss - dado não encontrado no nível procurado.
- Hit-rate (ratio) - percentual de hits no nível.
- Miss-rate (ratio) – percentual de misses no nível.
- Hit-time – tempo de acesso ao nível



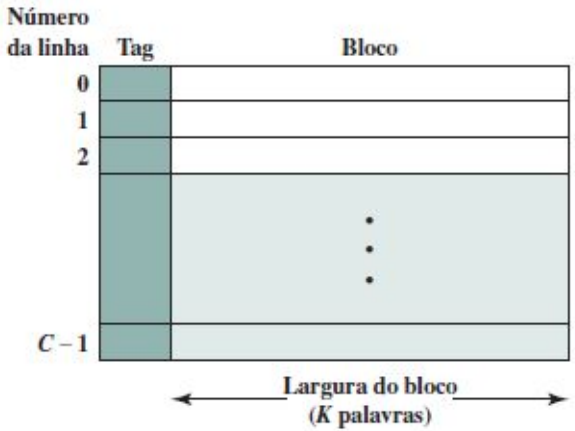
- Miss-penalty – tempo médio gasto para que o dado não encontrado no nível desejado seja transferido dos níveis mais baixos.

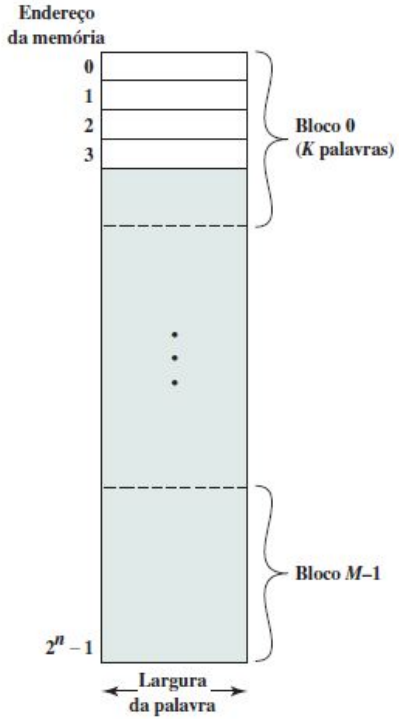
$$Tme = hit\text{-}time + (1 - hit\text{-}rate) * miss\text{-}penalty$$

OPERAÇÃO DE LEITURA



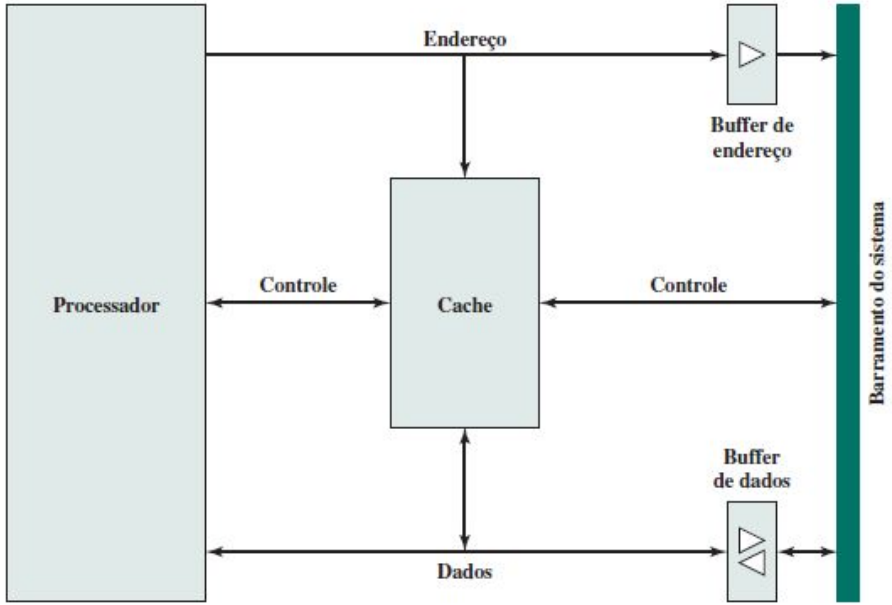
ESTRUTURA DE CACHE/MEMÓRIA PRINCIPAL





ORGANIZAÇÃO DA MEMÓRIA CACHE

- As linhas de dados e de endereços são também conectadas a áreas de armazenamento temporário de dados e de endereços, que se conectam ao barramento do sistema, por meio do qual é feito o acesso à memória principal.



ELEMENTOS DE PROJETO DA CACHE

Embora haja um grande número de implementações de memória cache, existem alguns elementos básicos de projeto que servem para classificar e diferenciar as arquiteturas de memórias cache.

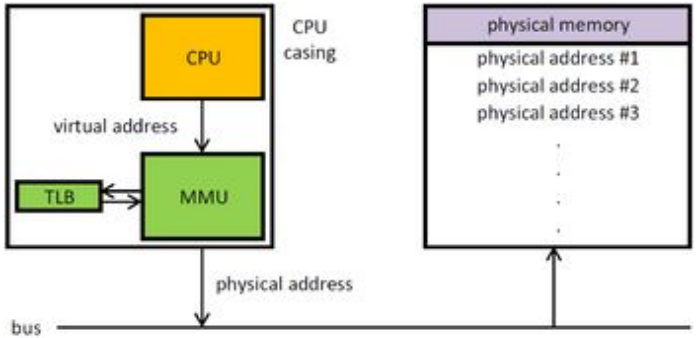
Tabela 4.2

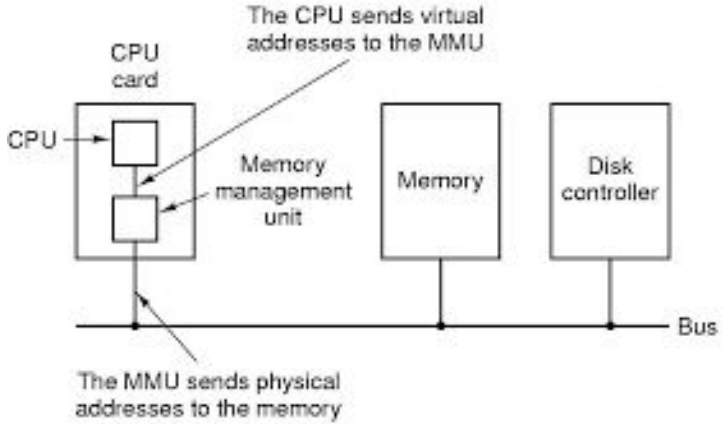
Elementos do projeto de cache.

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	

ELEMENTOS DE PROJETO DA CACHE

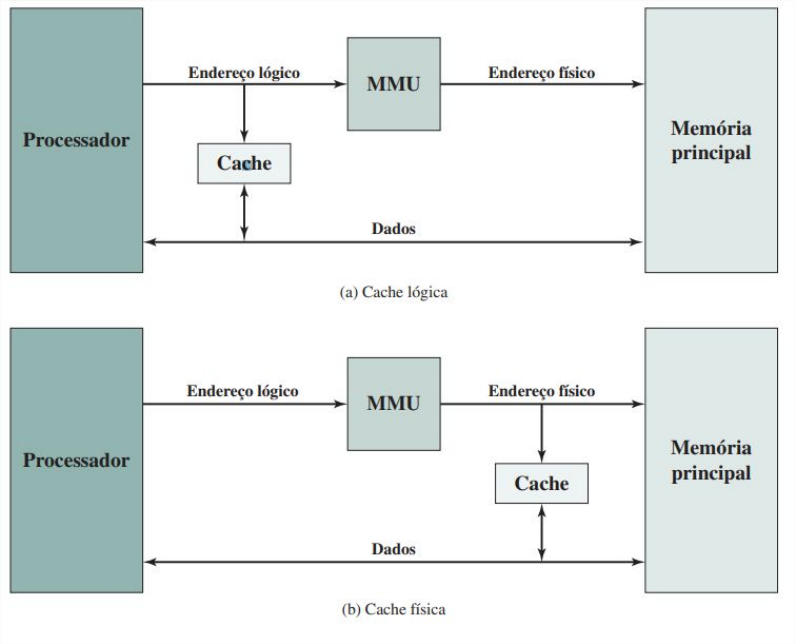
Endereços da Cache





ELEMENTOS DE PROJETO DA CACHE

Endereços da Cache



ELEMENTOS DE PROJETO DA CACHE

Tamanho da Memória Cache

- Pequeno o suficiente para que o custo médio geral por bit fosse próximo do custo médio da memória principal isolada
- Grande o suficiente para que o tempo de acesso médio geral fosse próximo do tempo de acesso médio da cache isolada
- Quanto maior a cache, maior o número de portas envolvidos no endereçamento da cache

Dentre outras motivações para minimizar o tamanho do cache...

Concluímos 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 também limita o tamanho da cache**

ELEMENTOS DE PROJETO DA CACHE

Tamanho da Memória Cache

Processador	Tipo	Ano de introdução	Cache L1[a]	Cache L2	Cache L3
IBM 360/85	Mainframe	1968	16–32 kB	—	—
PDP-11/70	Minicomputador	1975	1 kB	—	—
VAX 11/780	Minicomputador	1978	16 kB	—	—
IBM 3033	Mainframe	1978	64 kB	—	—
IBM 3090	Mainframe	1985	128–256 kB	—	—
Intel 80486	PC	1989	8 kB	—	—
Pentium	PC	1993	8 kB/8 kB	256–512 kB	—
PowerPC 601	PC	1993	32 kB	—	—
PowerPC 620	PC	1996	32 kB/32 kB	—	—
PowerPC G4	PC/servidor	1999	32 kB/32 kB	256 kB a 1 MB	2 MB
IBM S/390 G6	Mainframe	1999	256 kB	8 MB	—
Pentium 4	PC/servidor	2000	8 kB/8 kB	256 kB	—
IBM SP	Servidor avançado/ supercomputador	2000	64 kB/32 kB	8 MB	—
CRAY MTA[b]	Supercomputador	2000	8 kB	2 MB	—
Itanium	PC/servidor	2001	16 kB/16 kB	96 kB	4 MB
Itanium 2	PC/servidor	2002	32 kB	256 kB	6 MB
IBM POWER5	Servidor avançado	2003	64 kB	1.9 MB	36 MB
CRAY XD-1	Supercomputador	2004	64 kB/64 kB	1 MB	—
IBM POWER6	PC/servidor	2007	64 kB/64 kB	4 MB	32 MB
IBM z10	Mainframe	2008	64 kB/128 kB	3 MB	24–48 MB
Intel Core i7 EE 990	Estação de trabalho/ servidor	2011	6 × 32 kB/ 32kB	1,5 MB	12 MB
IBM zEnterprise 196	Mainframe/ servidor	2011	24 × 64 kB/ 128 kB	24 × 1,5 MB	24 MB L3 192 MB L4

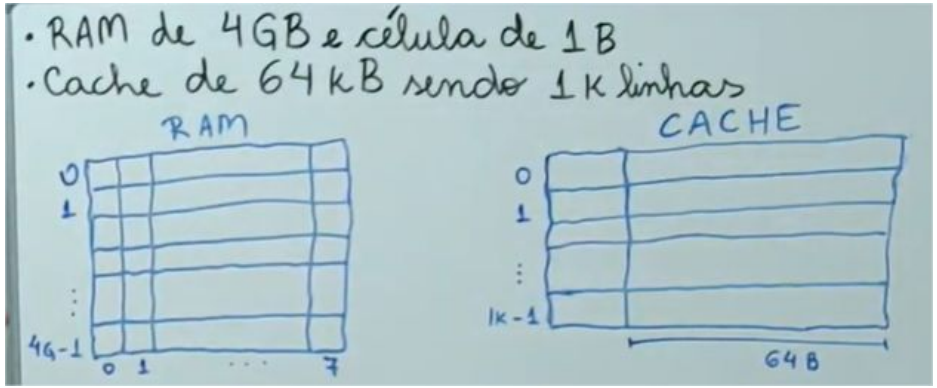
[a] Dois valores separados por uma barra referem-se a caches de instrução e dados. [b] As duas caches são apenas de instrução; não há caches de dados.

ELEMENTOS DE PROJETO DA CACHE

Funções de Mapeamento

Existem menos linhas de cache do que blocos da memória principal, dito isso foi-se necessário pensar em algoritmos, ou seja, para mapear os blocos da memória principal às linhas de cache.

Mapeamento Direto



ELEMENTOS DE PROJETO DA CACHE

Mapeamento Direto

A técnica mais simples, conhecida como mapeamento direto, mapeia cada bloco da memória principal a apenas uma linha de cache possível. O mapeamento é expresso **como: i = j módulo m**

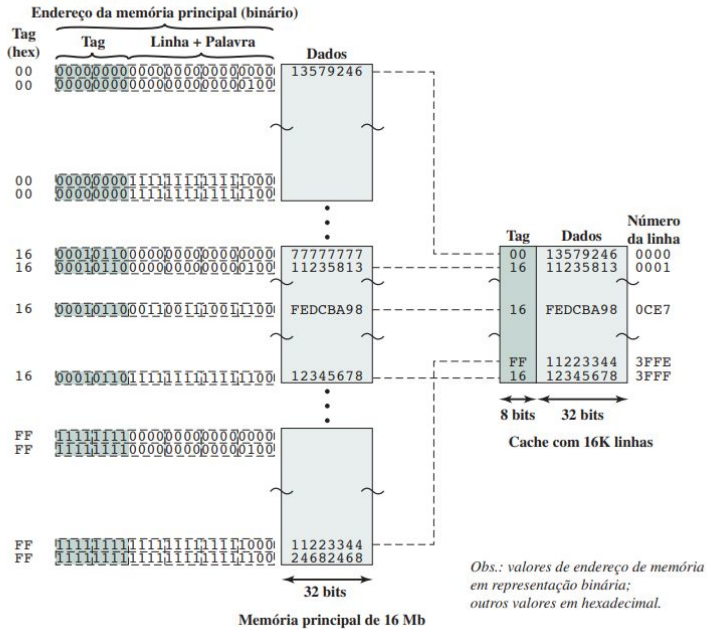
onde:
i = número da linha da cache
j = número do bloco da memória principal
m = número de linhas da cache

ELEMENTOS DE PROJETO DA CACHE

Mapeamento Direto

Figura 4.10

Exemplo de mapeamento direto.



ELEMENTOS DE PROJETO DA CACHE

Mapeamento Associativo

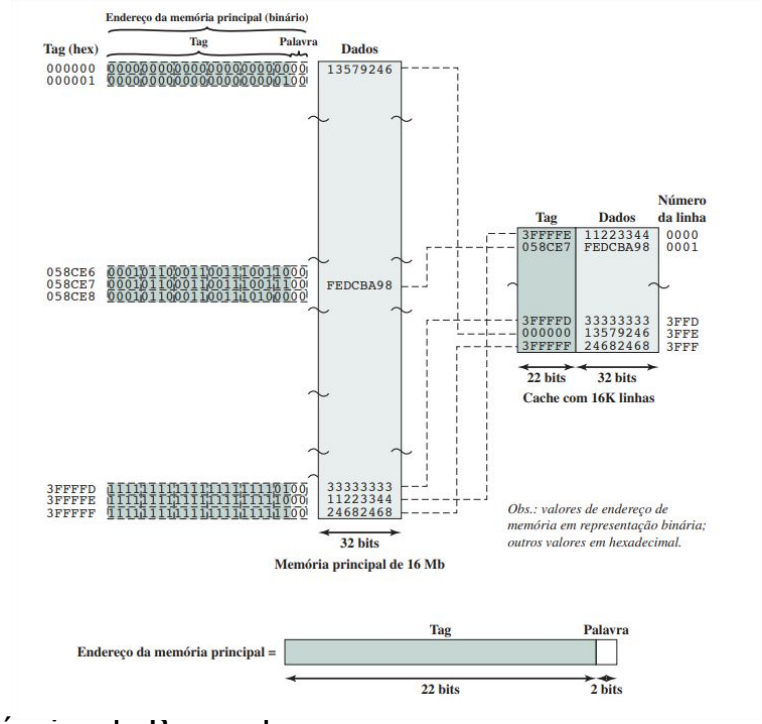
O mapeamento associativo compensa a desvantagem do mapeamento direto, permitindo que cada bloco da memória principal seja carregado em qualquer linha da cache

- **A lógica de controle da cache interpreta um endereço de memória simplesmente como um campo Tag e um campo palavra**.
- **O campo Tag identifica o bloco da memória principal**
- **A lógica de controle da cache precisa comparar simultaneamente o tag de cada linha**

Concluímos que com o mapeamento associativo, existe flexibilidade em relação a qual bloco substituir quando um novo bloco for lido para a cache porém demanda uma complexidade do
circuito necessário para examinar as tags de todas as linhas da cache em paralelo.

ELEMENTOS DE PROJETO DA CACHE

Mapeamento Associativo



ELEMENTOS DE PROJETO DA CACHE

Mapeamento Associativo por Conjunto

É um meio-termo que realça os pontos fortes das técnicas direta e associativa, enquanto reduz suas desvantagens.

- **A cache é uma série de conjuntos, cada um consistindo em uma série de linhas.**

$$m = v \times k$$

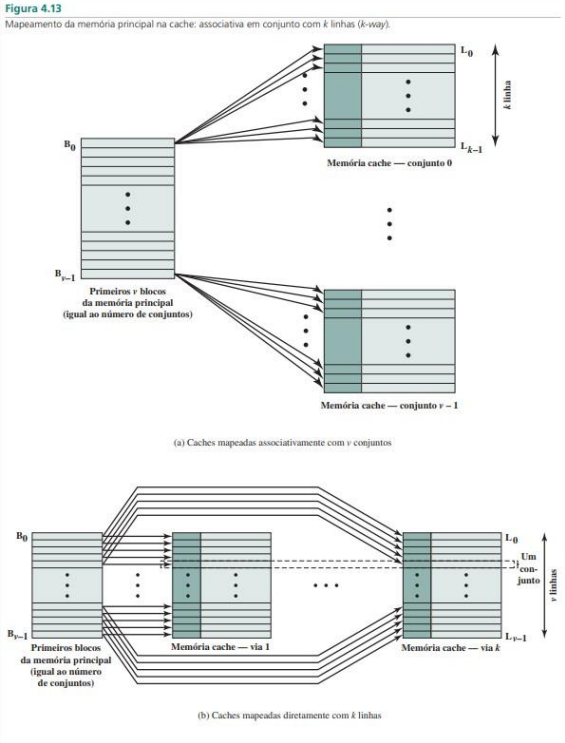
$$i = j \text{ módulo } v$$

em que

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

ELEMENTOS DE PROJETO DA CACHE

Mapeamento Associativo por Conjunto

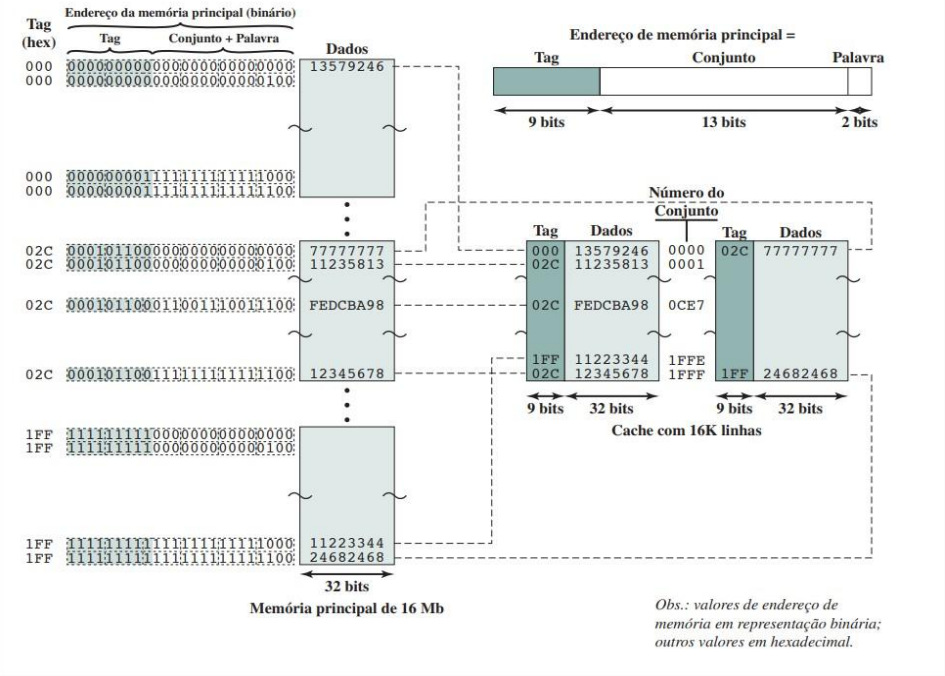


ELEMENTOS DE PROJETO DA CACHE

Mapeamento Associativo por Conjunto

Figura 4.15

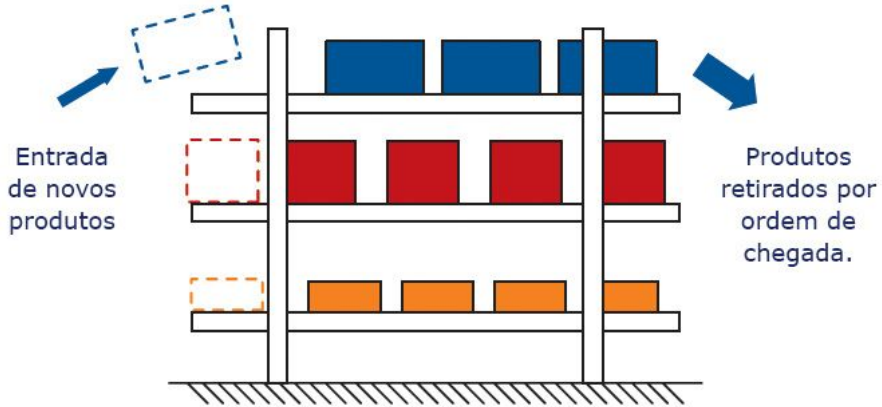
Exemplo de mapeamento associativo em conjunto com duas linhas.



Algoritmos de substituição

FIFO (First in first out)

O método consiste em substituir o bloco que acaba de chegar pelo mais antigo na memória, pode não ser tão interessante, pois o bloco mais antigo pode ser mais importante.

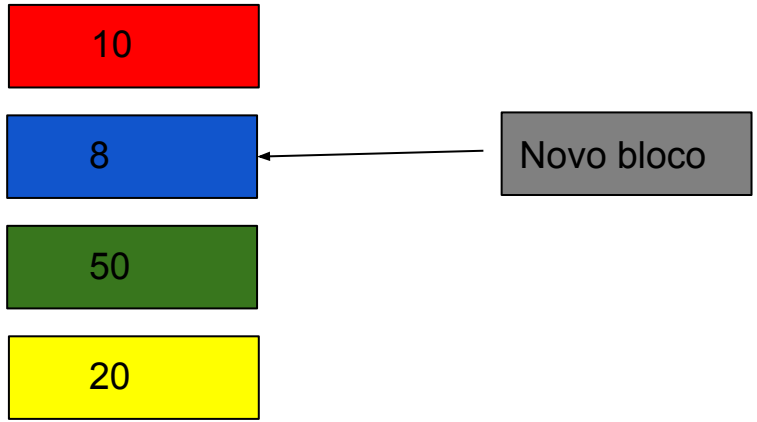


Algoritmos de substituição

LFU (Least Frequently Used)

Nesse método o bloco que chega irá substituído na linha com o bloco usado menos vezes.

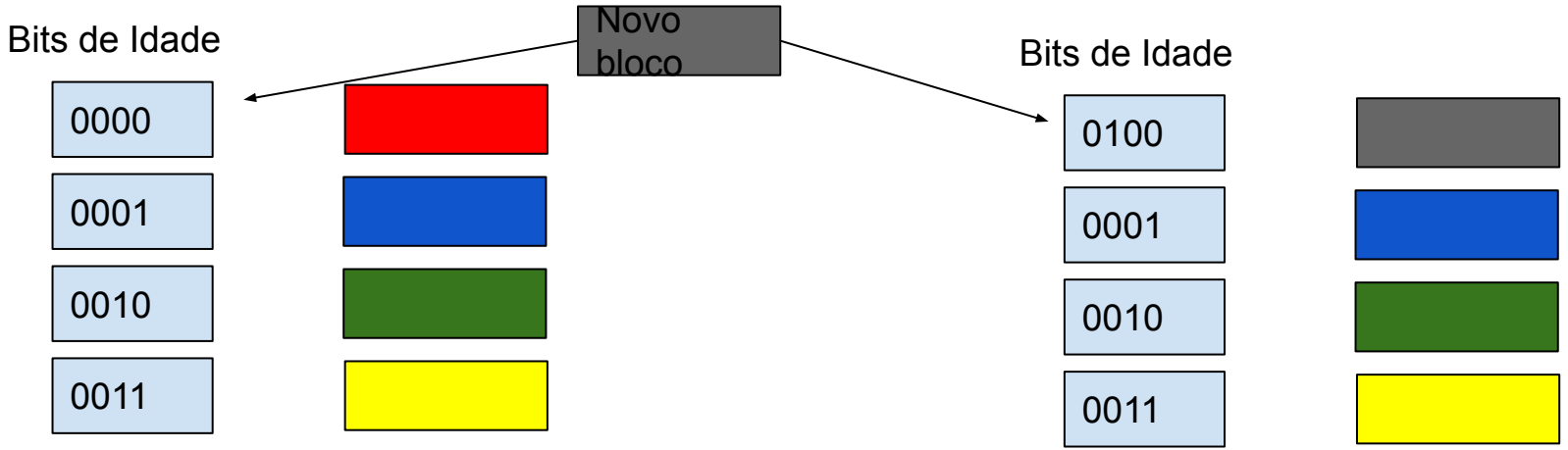
Contra: Pode guardar um bloco que já foi muito utilizado e não será mais necessário.



Algoritmos de substituição

LRU (Least Recently Used)

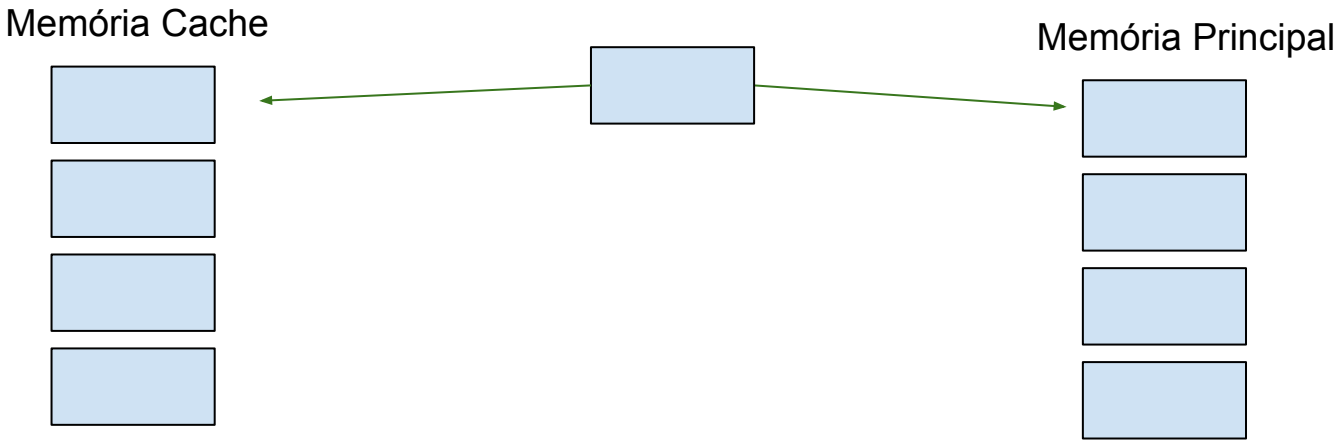
Nesse método o bloco menos recentemente usado será substituido, ou seja, o bloco usado há mais tempo será trocado pelo novo bloco.



Política de Escrita

Write Through

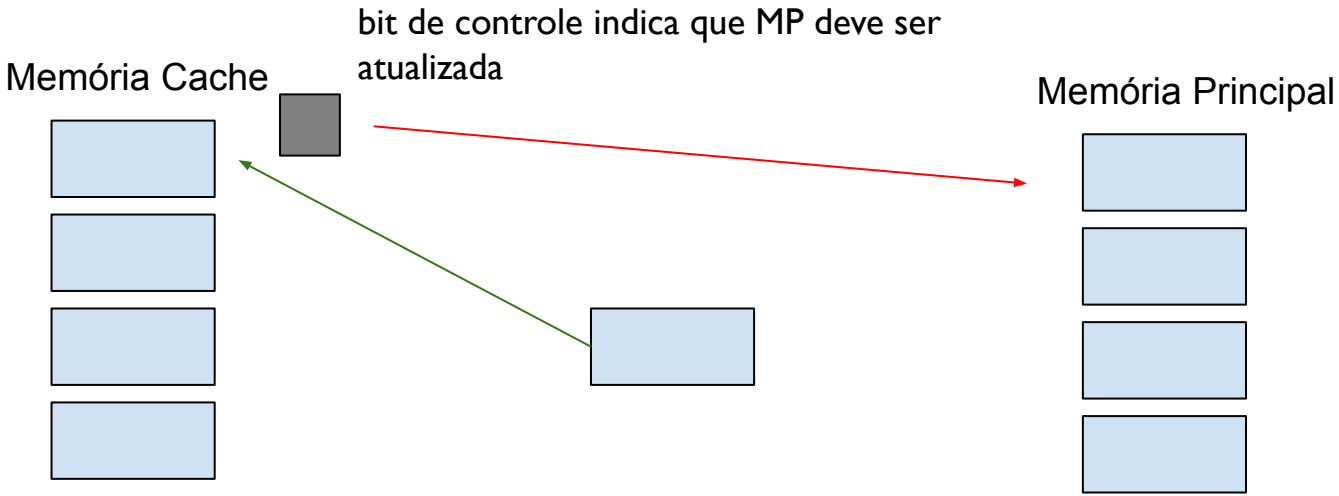
Todas as operações são feitas na cache e na MP. Gera tráfego de memória, possível gargalo.Caso haja outros processadores eles também atualizam suas caches.



Política de Escrita

Write Back

A cache é atualizada e é gerado uma indicação para que a MP seja atualizada.

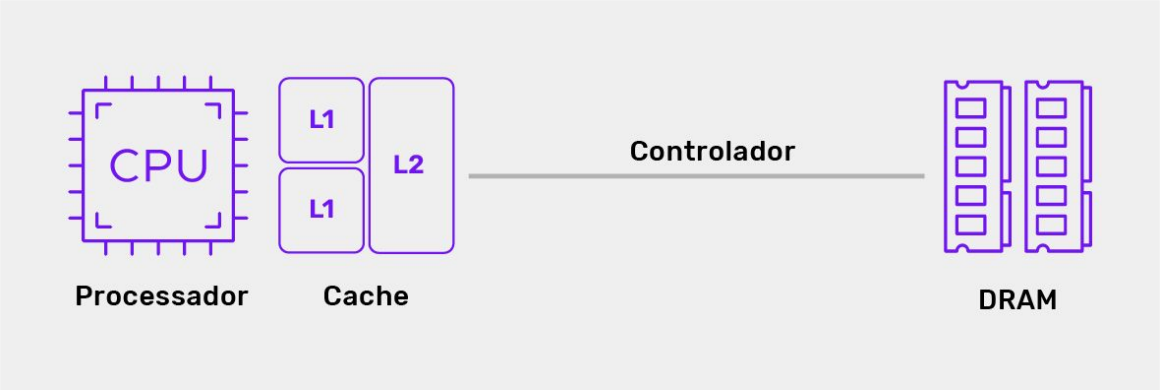


Política de Escrita

- Observação do barramento com write through
- Transparência do hardware
- Memória não cacheável

Número de caches

Inicialmente os sistemas possuíam uma única cache, recentemente o uso de múltiplas caches é mais comum.



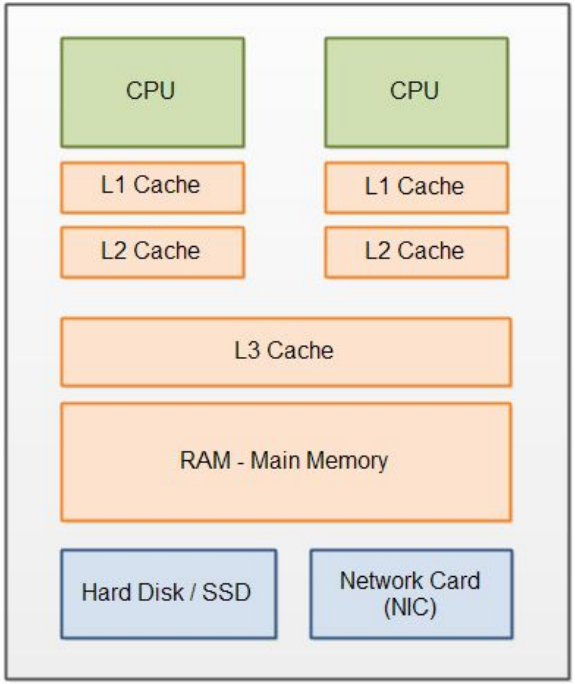
Número de caches

	DRAM (dinâmica)	SRAM (estática)
Vantagens	- alta densidade de integração - baixo consumo de potência - baixa geração de calor - baixo custo	- alta velocidade - não precisam de "*refresh*"
Desvantagens	- baixa velocidade - necessidade de refresco ("*refresh*")	- baixa densidade de integração - alto consumo de potência - alta geração de calor - alto custo
Tempo de Acesso	60 a 70 ηs	10 a 20 ηs

Número de caches

Níveis de Cache:

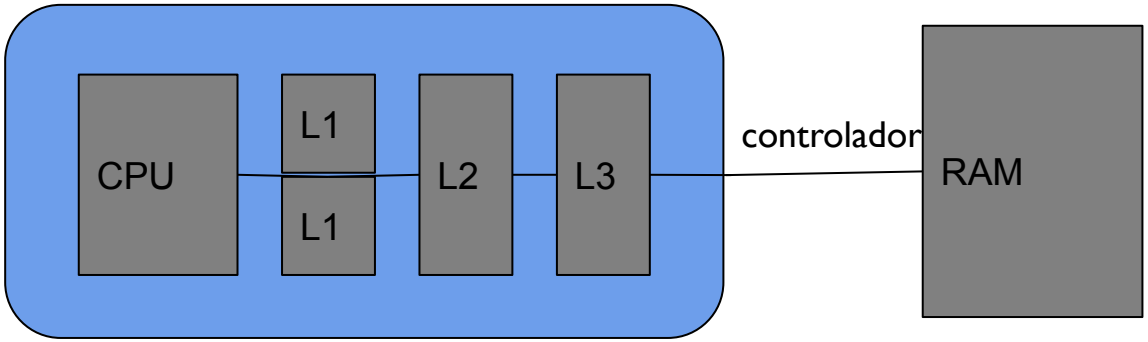
A memória cache é dividida em três níveis, conhecidos como L1, L2 e L3. Eles dizem respeito à proximidade da memória cache das unidades de execução do processador.



Número de caches

Caches Unificadas ou Separadas:

Incluem duas caches L1 no chip, uma para dados e uma para instruções. Para o Pentium 4, por exemplo, a cache de dados L1 tem 16 kB, usando um tamanho de linha de 64 bytes e uma organização associativa em conjunto com quatro linhas.



Organização da cache do Pentium 4

A evolução da organização das memórias cache pode ser vista na evolução dos microprocessadores Intel.

Organização da cache do Pentium 4

Problema	Solução	Processador em que o recurso apareceu inicialmente
Memória externa mais lenta que o barramento do sistema	Acrescentar cache externa usando tecnologia de memória mais rápida	386
Maior velocidade do processador torna o barramento externo um gargalo para o acesso à cache L2	Mover a cache externa para o chip, trabalhando na mesma velocidade do processador	486
Cache interna um tanto pequena, por conta do espaço limitado no chip	Acrescentar cache L2 externa usando tecnologia mais rápida que a memória principal	486
Quando ocorre uma disputa entre o mecanismo de pré-busca de instruções e a unidade de execução no acesso simultâneo à memória cache. Nesse caso, a busca antecipada é adiada até o término do acesso da unidade de execução aos dados	Criar caches separadas para dados e instruções	Pentium

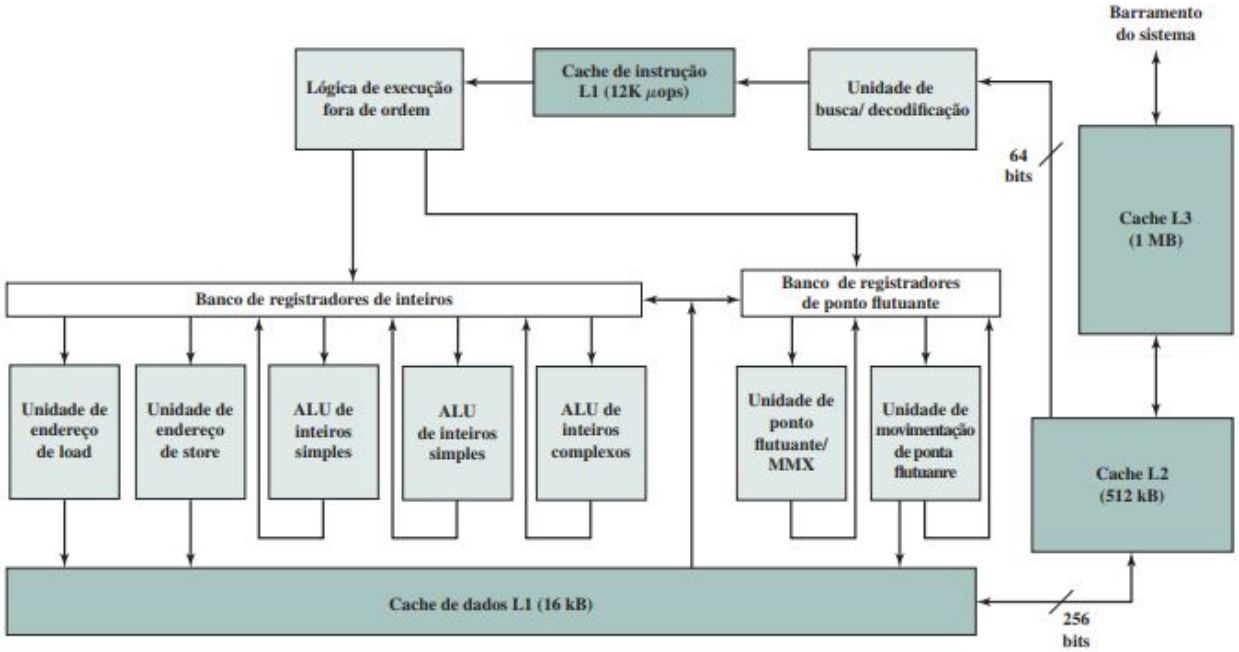
Organização da cache do Pentium 4

Problema	Solução	Processador em que o recurso apareceu inicialmente
Maior velocidade do processador torna o barramento externo um gargalo para o acesso à cache L2	Criar barramento back-side separado, que trabalha com velocidade mais alta que o barramento externo principal (front-side). O barramento back- -side é dedicado à cache L2	Pentium Pro
	Mover cache L2 para o chip do processador	Pentium II
Algumas aplicações lidam com bancos de dados enormes, e precisam ter acesso rápido a grandes quantidades de dados. As caches no chip são muito pequenas	Acrescentar cache L3 externa	Pentium III
	Mover cache L3 para o chip	Pentium 4

Organização da cache do Pentium 4

Figura 4.18

Diagrama em bloco do Pentium 4.



EXERCÍCIOS

1. Em geral, quais são as estratégias para explorar a localidade espacial e a localidade temporal?
2. Quais os principais algoritmos de substituição? Fale sobre as vantagens e desvantagens de cada um.
3. Comente sobre as vantagens e desvantagens de cada tipo de mapeamento: direto, associativo e associativo por conjunto.


obrigado