William Stallings
Arquitetura e Organização de Computadores
8ª Edição

Capítulo 4
Memória cache

Os textos nestas caixas foram adicionados pelo Prof. Joubert

Características

- Localização.
- Capacidade.
- Unidade de transferência.
- Método de acesso.
- Desempenho.
- Tipo físico.
- Características físicas.
- Organização.

Localização

- CPU.
- Interna.
- Externa.

Capacidade

- Tamanho de palavra:
  - —A unidade de organização natural.
- Número de palavras:
  - —ou Bytes.

Métodos de acesso

- Sequencial:
  - Começa no início e lê em ordem.
  - Tempo de acesso depende da localização dos dados e local anterior.
  - Por exemplo, fita.
- Direto:
  - Blocos individuais possuem endereço exclusivo.
  - Acesso saltando para vizinhança, mais busca sequencial.
  - Tempo de acesso depende da localização e local anterior.
  - Por exemplo, disco.

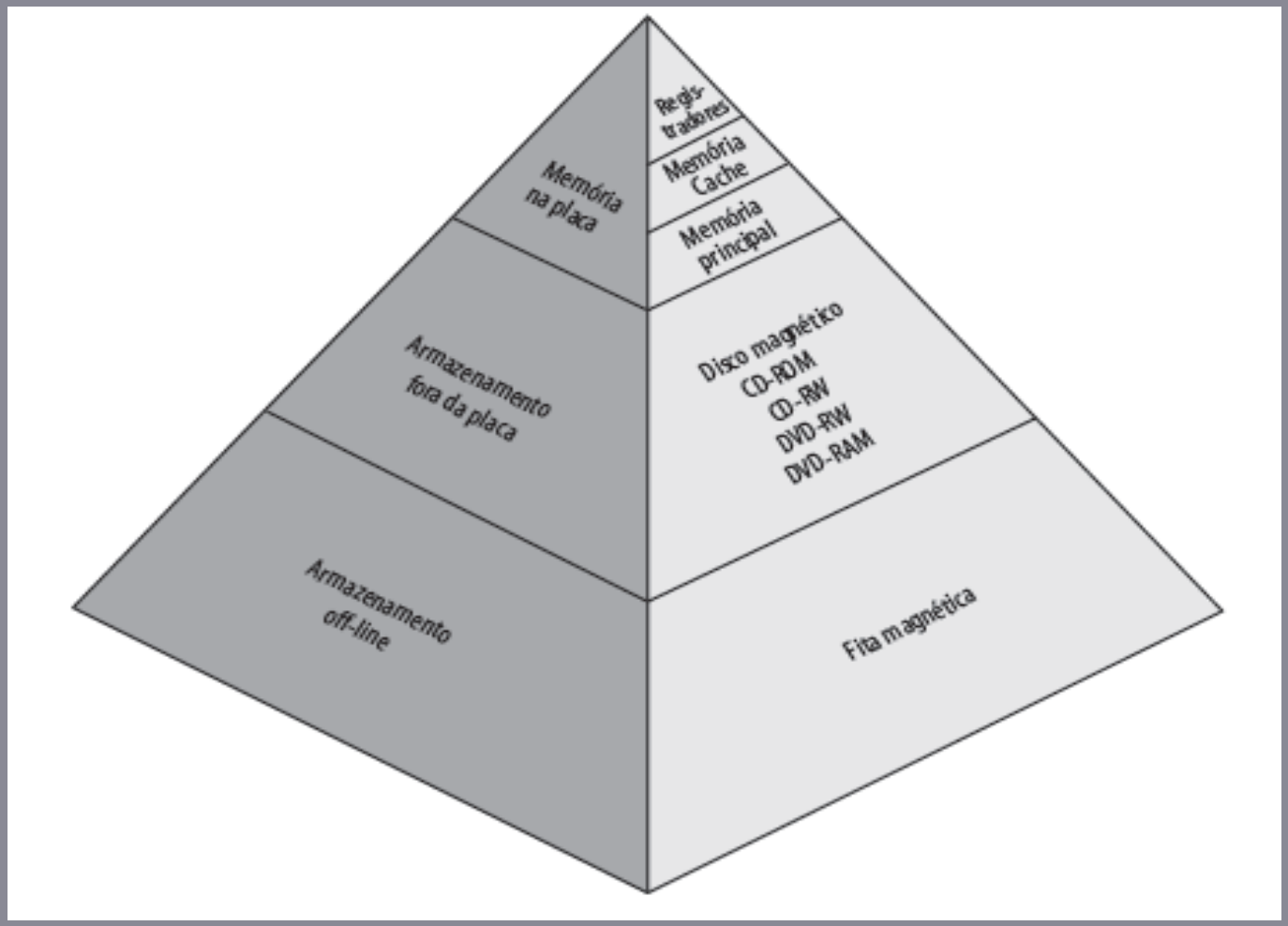
- Aleatório:
  - Endereços individuais identificam localizações com exatidão.
  - Tempo de acesso é independente da localização ou acesso anterior.
  - P.e., RAM.

- Associativo:
  - Dados são localizados por uma comparação com conteúdo de uma parte do armazenamento.
  - Tempo de acesso é independente do local ou acesso anterior.
  - P.e., cache.

Hierarquia de memória

- Registradores:
  - Na CPU.
- Memória interna ou principal:
  - Pode incluir um ou mais níveis de cache.
  - "RAM".
- Memória externa:
  - Armazenamento de apoio.

Hierarquia de memória – Diagrama



Desempenho

- Tempo de acesso (latência):
  - Tempo entre apresentar o endereço e obter os dados válidos.
- Tempo de ciclo de memória:
  - Tempo que pode ser exigido para a memória se "recuperar" antes do próximo acesso.
  - Tempo de ciclo é acesso + recuperação.
- Taxa de transferência:
  - Taxa em que os dados podem ser movidos.

Tipos físicos

- Semicondutor:
  - RAM.
- Magnético:
  - Disco e fita.
- Óptico:
  - CD e DVD.
- Outros:
  - Bolha.
  - Holograma.

Características físicas

- Deterioração.
- Volatilidade.
- Apagável.
- Consumo de energia.

Organização

- Arranjo físico dos bits em palavras.
- Nem sempre óbvia.
- P.e., intercalada.

Lista de hierarquia

- Registradores.
- Cache L1.
- Cache L2.
- Memória principal.
- Cache de disco.
- Disco.
- Óptica.
- Fita.

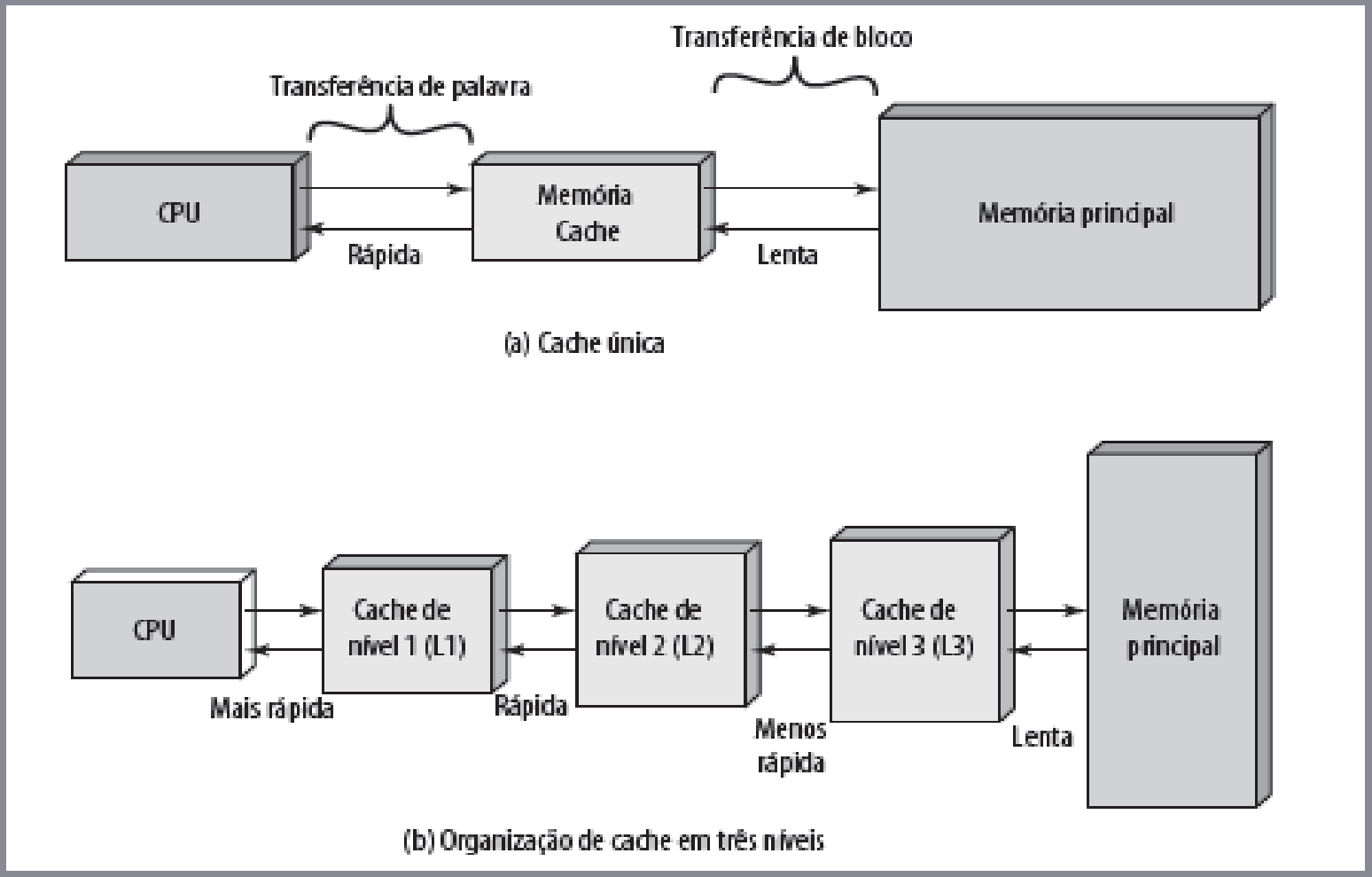
Localidade de referência

- Durante o curso da execução de um programa, as referências à memória tendem a se agrupar.
- P.e., loops.

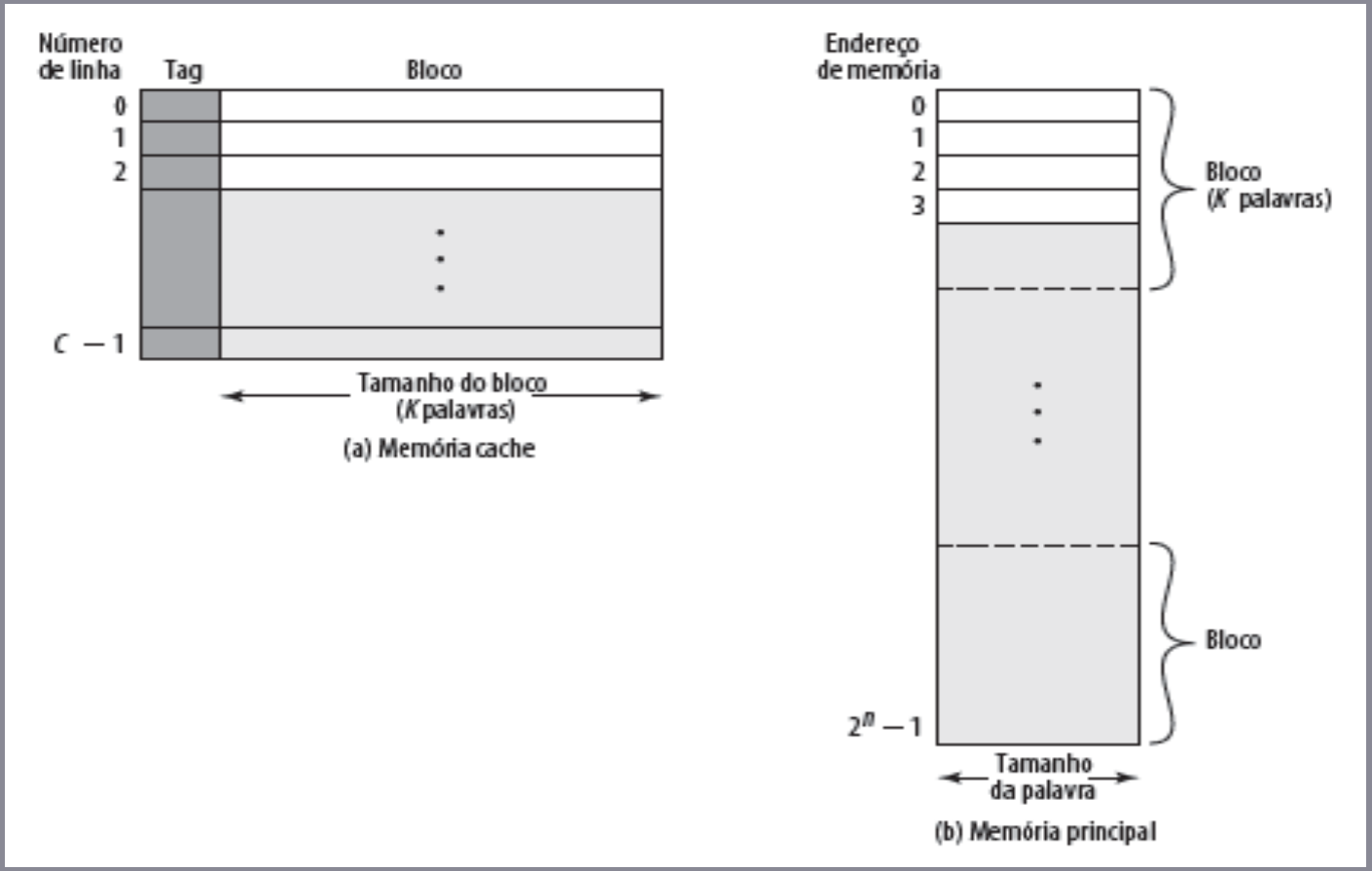
Cache

- Pequena quantidade de memória rápida.
- Fica entre a memória principal normal e a CPU.
- Pode estar localizada no chip da CPU ou módulo.

Cache e memória principal



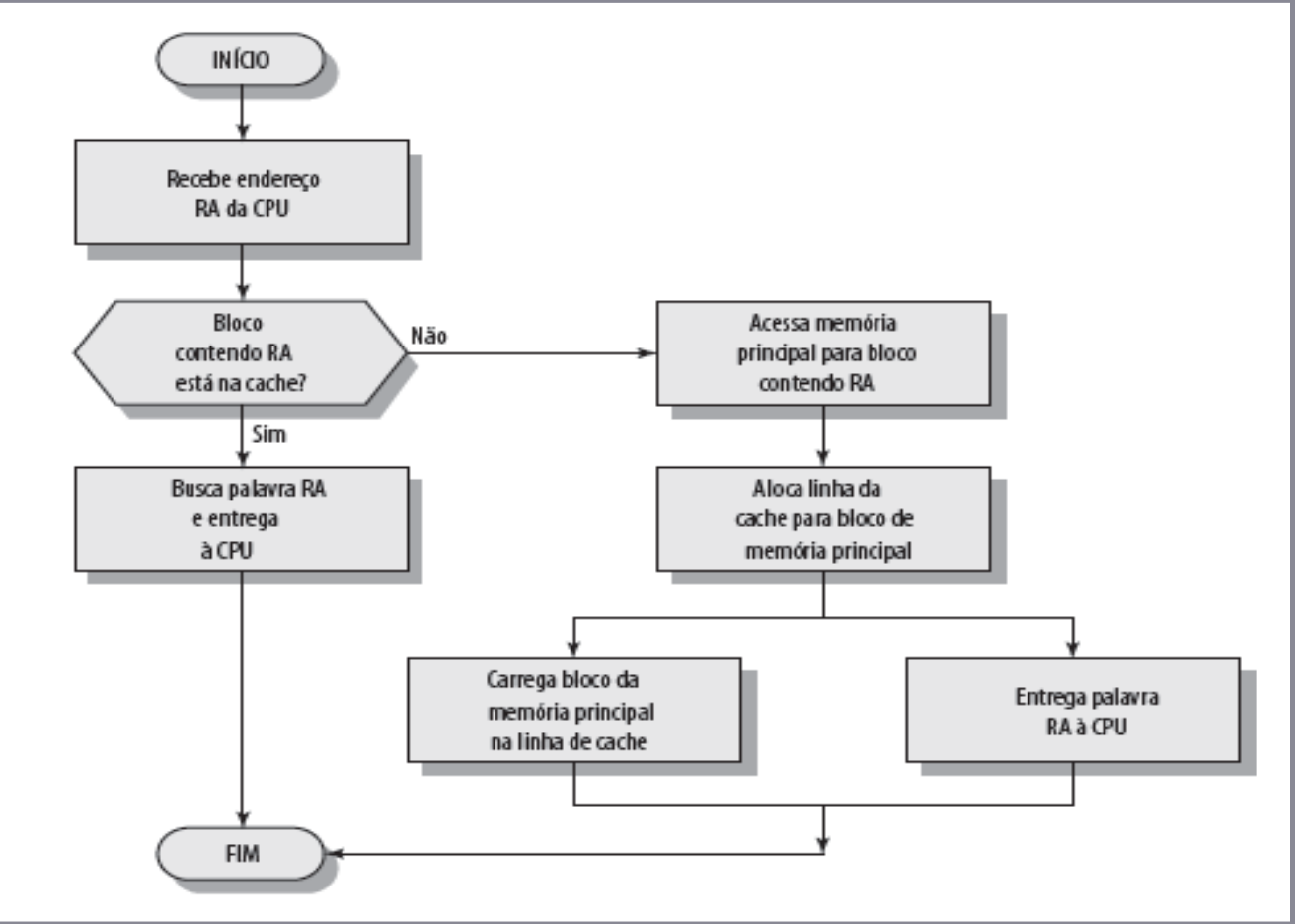
Estrutura de cache/memória principal



Operação da cache – visão geral

- CPU requisita conteúdo do local de memória.
- Verifica se os dados estão em cache.
- Se estiverem, apanha da cache (cache hit).
- Se não (cache miss), 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



Projeto de cache

- Endereçando.
- Tamanho.
- Função de mapeamento.
- Algoritmo de substituição.
- Política de escrita.
- Tamanho de bloco.
- Número de caches.

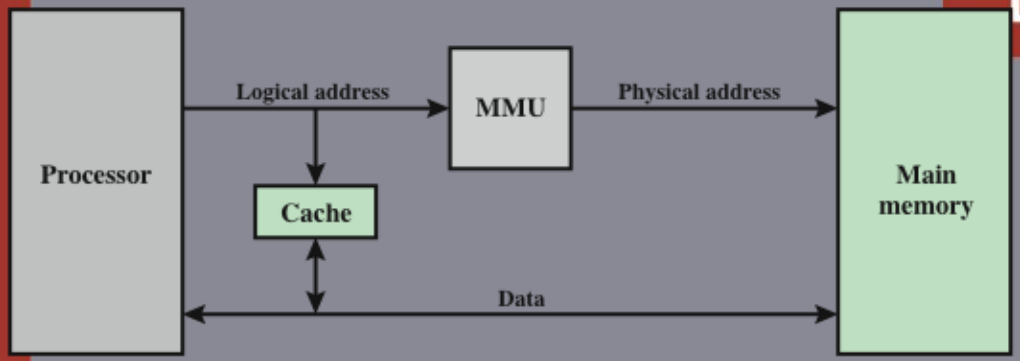
Endereçamento de cache

- Onde fica a cache?
  - Entre processador e unidade de gerenciamento de memória virtual.
  - Entre MMU e memória principal.
- Cache lógica (cache virtual) armazena dados usando endereço virtual.
  - Processador acessa cache diretamente, não através da cache física.
  - Acesso à cache mais rápido, antes da tradução de endereço da MMU.
  - Endereços virtuais usam o mesmo espaço de endereços para diferentes aplicações.
    - Deve esvaziar cache a cada troca de contexto.
- Cache física armazena dados usando endereços físicos da memória principal.

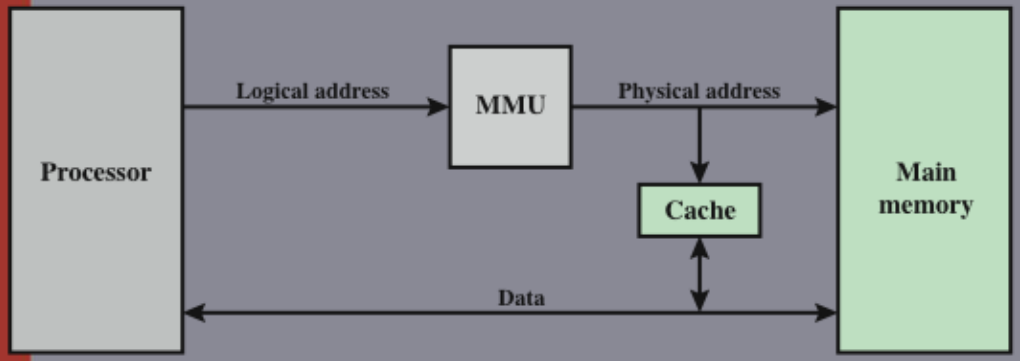
Tamanho não importa

- Custo:
  - Mais cache é caro.
- Velocidade:
  - Mais cache é mais rápido (até certo ponto).
  - Verificar dados na cache leva tempo.

Logical and Physical Caches



(a) Logical Cache



(b) Physical Cache

Figure 4.7 Logical and Physical Caches

Comparação de tamanhos de memória cache

Processor	Type	Year of Introduction	L1 Cache[a]	L2 cache	L3 Cache
IBM 360/85	Mainframe	1968	16 to 32 kB	—	—
PDP-11/70	Minicomputer	1975	1 kB	—	—
VAX 11/780	Minicomputer	1978	16 kB	—	—
IBM 3033	Mainframe	1978	64 kB	—	—
IBM 3090	Mainframe	1985	128 to 256 kB	—	—
Intel 80486	PC	1989	8 kB	—	—
Pentium	PC	1993	8 kB/8 kB	256 to 512 KB	—
PowerPC 601	PC	1993	32 kB	—	—
PowerPC 620	PC	1996	32 kB/32 kB	—	—
PowerPC G4	PC/server	1999	32 kB/32 kB	256 KB to 1 MB	2 MB
IBM S/390 G6	Mainframe	1999	256 kB	8 MB	—
Pentium 4	PC/server	2000	8 kB/8 kB	256 KB	—
IBM SP	High-end server/ supercomputer	2000	64 kB/32 kB	8 MB	—
CRAY MTA[b]	Supercomputer	2000	8 kB	2 MB	—
Itanium	PC/server	2001	16 kB/16 kB	96 KB	4 MB
Itanium 2	PC/server	2002	32 kB	256 KB	6 MB
IBM POWER5	High-end server	2003	64 kB	1.9 MB	36 MB
CRAY XD-1	Supercomputer	2004	64 kB/64 kB	1MB	—
IBM POWER6	PC/server	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	Workstaton/ server	2011	6 × 32 kB/32 kB	1.5 MB	12 MB
IBM zEnterprise 196	Mainframe/ Server	2011	24 × 64 kB/ 128 kB	24 × 1.5 MB	24 MB L3 192 MB L4

Função de mapeamento

Usando o exemplo....

- Cache de 64 Kbytes.
- Bloco de cache de 4 bytes.
  - Ou seja, cache é de 16k (2^{14}) linhas de 4 bytes.
- 16 MB de memória principal.
- Endereço de 24 bits.
  - (2^{24}=16M)

Mapeamento direto

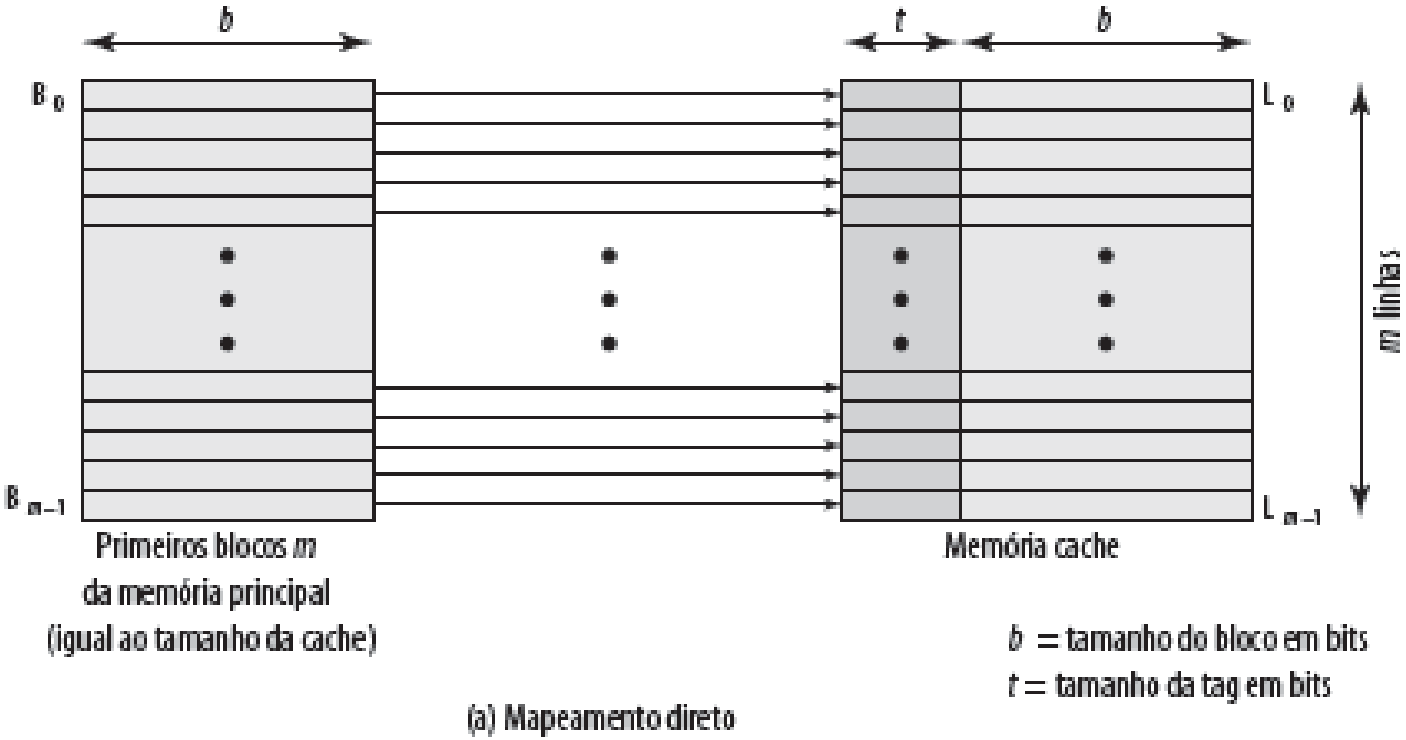
- Cada bloco de memória principal mapeado apenas para uma linha de cache.
  - Ou seja, se um bloco está na cache, ele deve estar em um local específico.
- Endereço está em duas partes.
- W bits menos significativos identificam uma palavra exclusiva.
- S bits mais significativos especificam um bloco de memória.
- Os MSBs são divididos em um campo de linha de cache r e uma tag de s-r (parte mais significativa).

Mapeamento direto
Estrutura de endereços

Tag s-r	Linha ou slot r	Palavra w
8	14	2

- Endereço de 24 bits.
- Identificador de palavra de 2 bits (bloco de 4 bytes).
- Identificador de bloco de 22 bits.
  - Tag de 8 bits (=22-14).
  - Slot ou linha de 14 bits.
- Dois blocos na mesma linha não têm o mesmo campo de tag.
- Verifica conteúdo da cache localizando linha e verificando tag.

Mapeamento direto da cache para memória principal

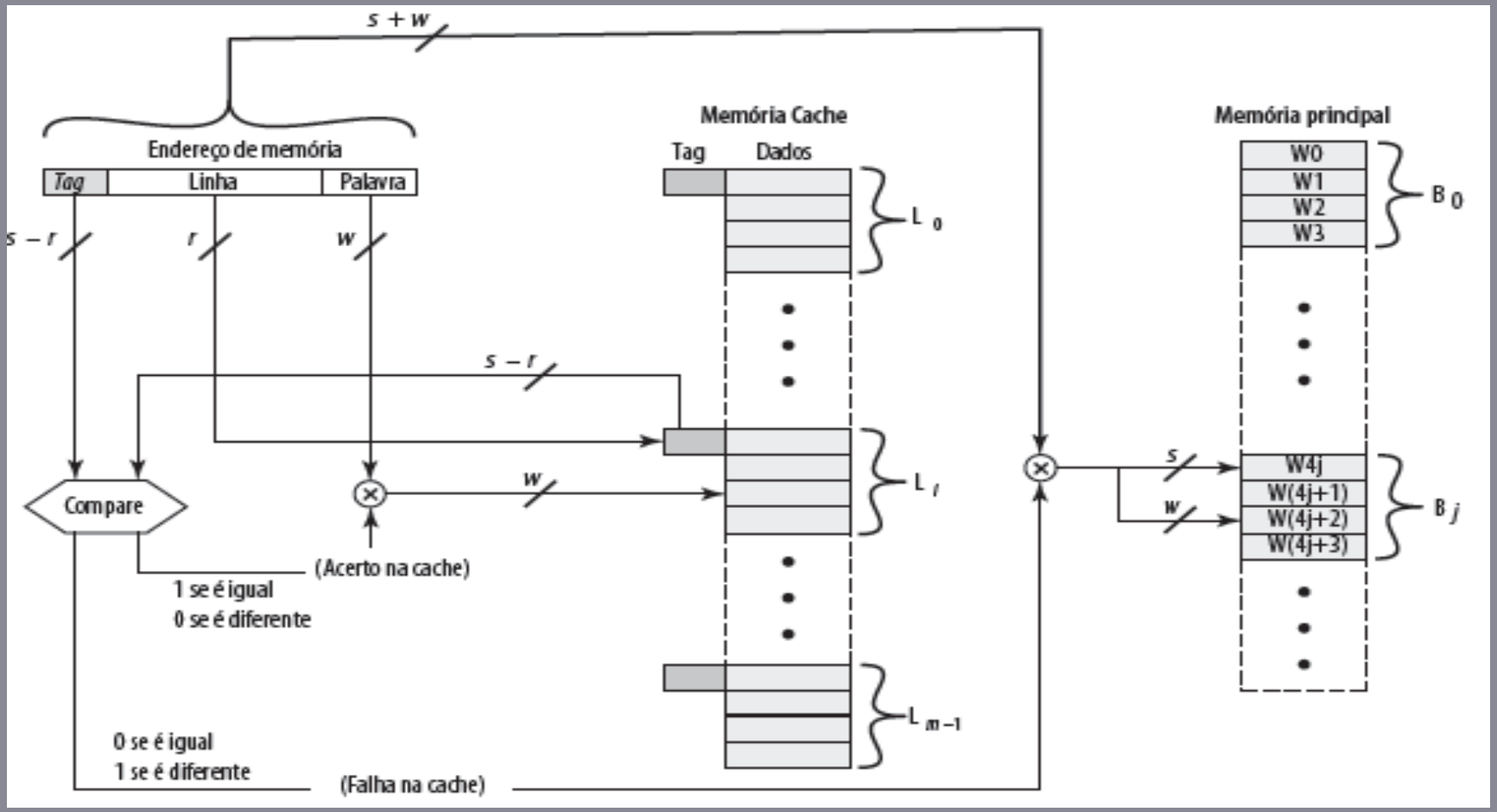


(a) Mapeamento direto

Mapeamento direto
Tabela de linhas de cache

Linha de cache	Blocos de memória principal mapeados
0	0, m, 2m, 3m…2s-m
1	1,m+1, 2m+1…2s-m+1
…	
m-1	m-1, 2m-1,3m-1…2s-1

Organização da cache com mapeamento direto



Resumo de mapeamento direto

- Tamanho de endereço= (s + w) bits.
- Número de unidades endereçáveis= 2s+w palavras ou bytes.
- Tamanho de bloco= tamanho de linha= 2w words ou bytes.
- Número de blocos na memória principal= 2s+ w/2w = 2s.
- Número de linhas na cache= m= 2r.
- Tamanho da tag= (s – r) bits.

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.

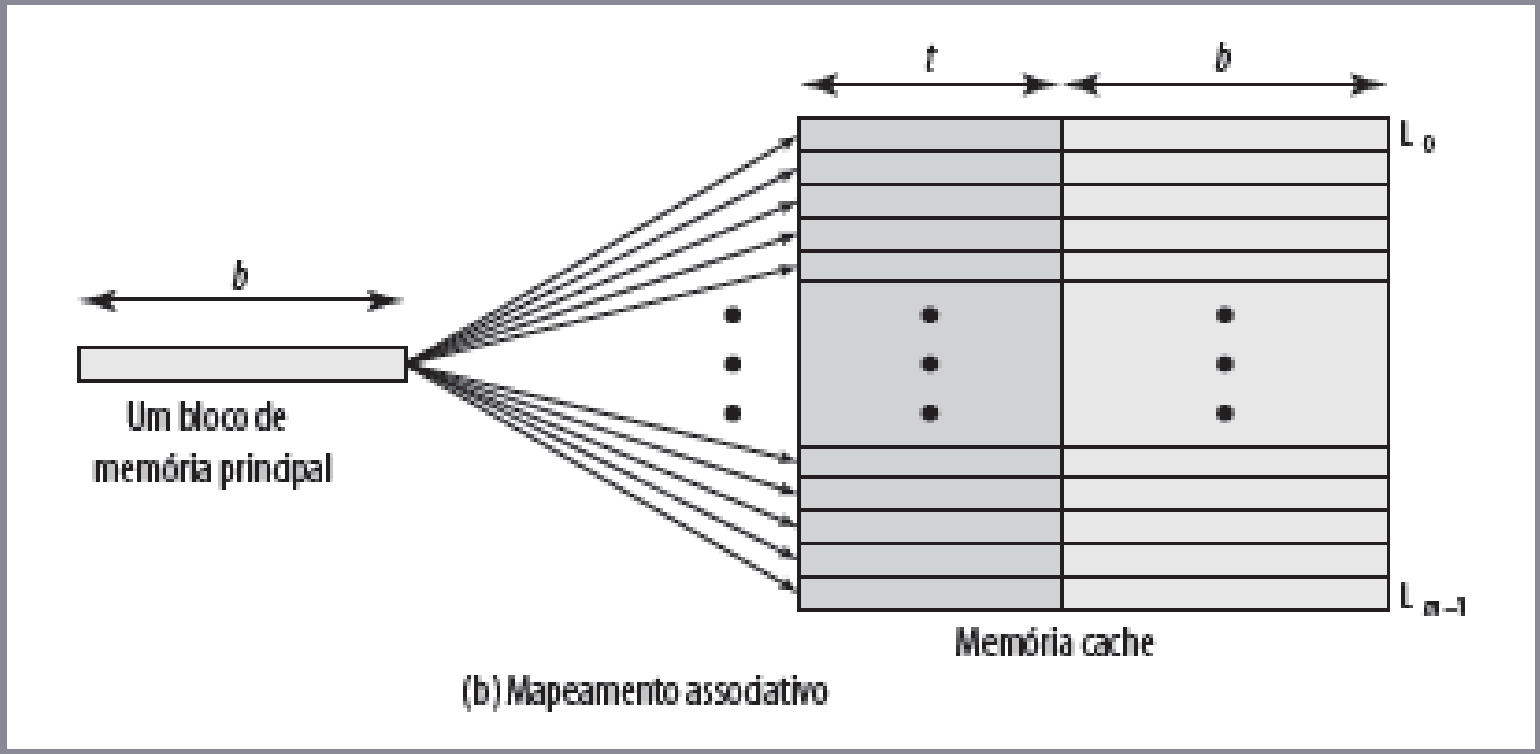
Cache vítima

- Menor penalidade de falha.
- Lembra-se do que foi descartado.
  - Já buscado.
  - Usa novamente com pouca penalidade.
- Totalmente associativa.
- 4 a 16 linhas de cache.
- Entre cache L1 mapeada diretamente e nível de memória seguinte.

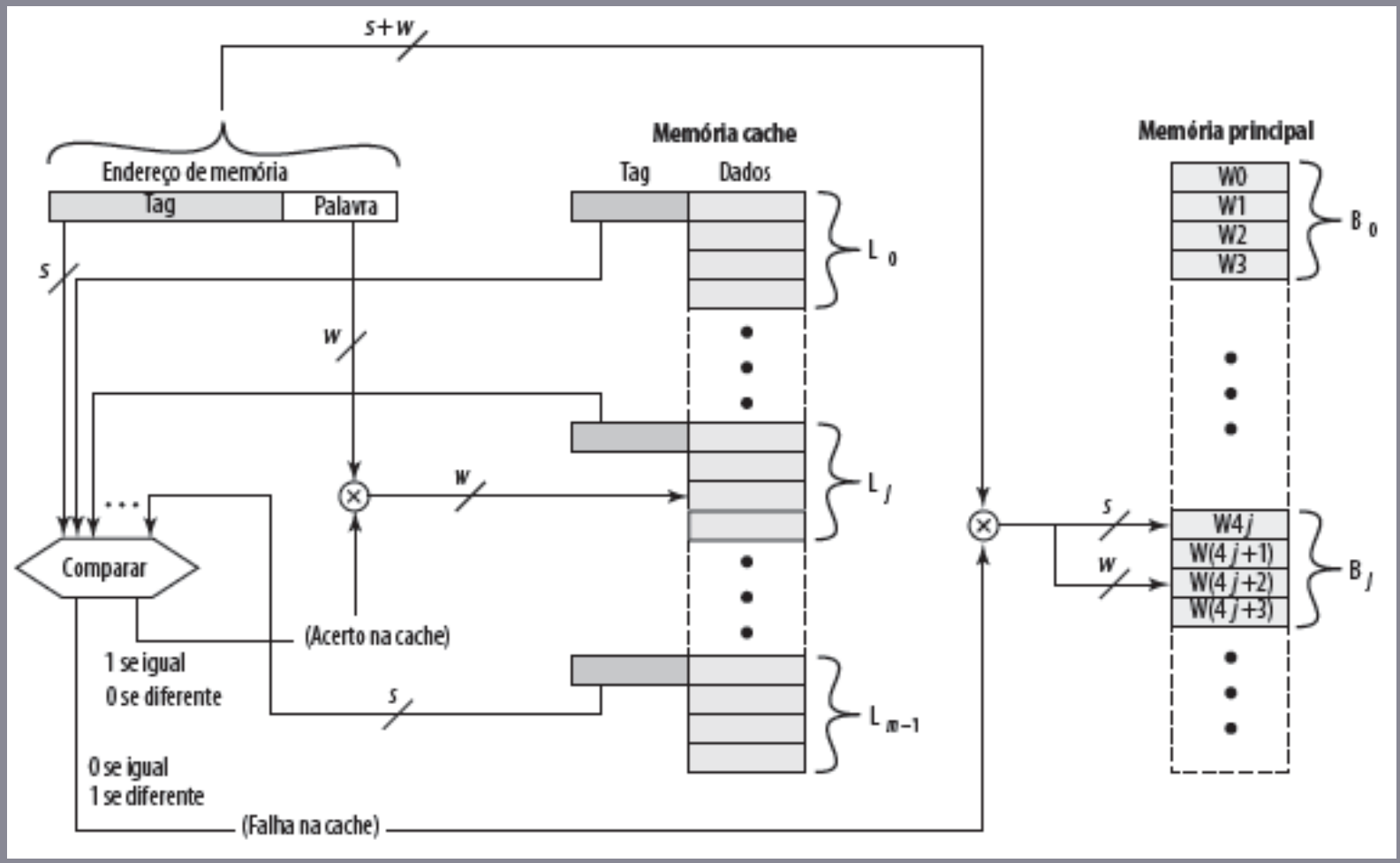
Mapeamento associativo

- 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.
- Tag de cada linha é examinada em busca de combinação.
- Pesquisa da cache é dispendiosa.

Mapeamento associativo da cache para a memória principal



Organização de cache totalmente associativa



Mapeamento associativo

Estrutura de endereço

Tag 22 bit	Palavra 2 bit

- Tag de 22 bits armazenado a cada bloco de 32 bits de dados.
- Compara campo de tag com entrada de tag na cache para procurar acerto.
- 2 bits menos significativos do endereço identificam qual word de 16 bits é exigida do bloco de dados de 32 bits.
- P.e.:

Endereço	Tag	Dados	Linha de cache
FFFFFC	FFFFFC	24682468	3FFF

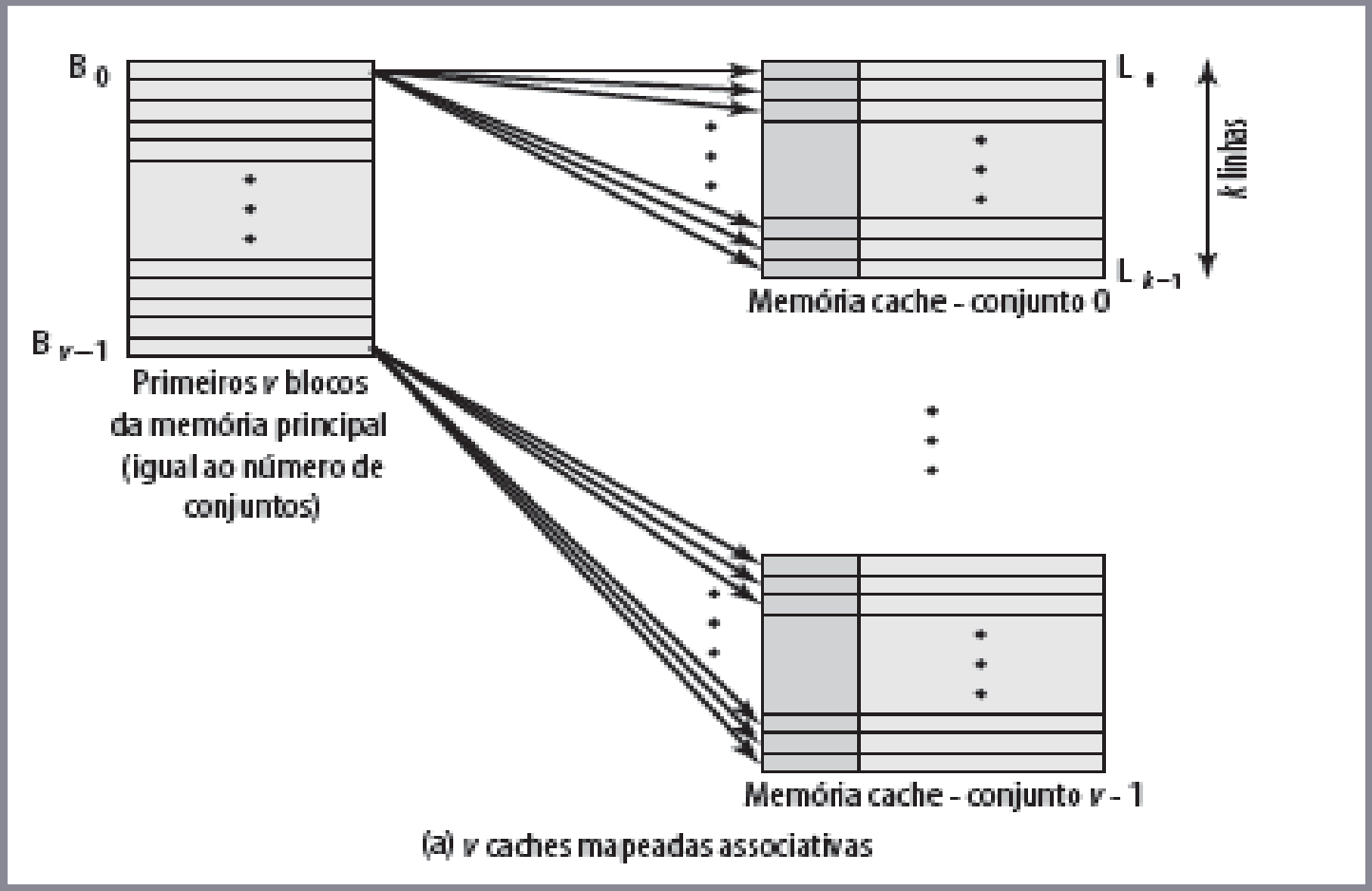
Resumo do mapeamento associativo

- Tamanho do endereço= (s + w) bits.
- Número de unidades endereçáveis= 2s+w words ou bytes.
- Tamanho do bloco= tamanho de linha= 2w palavras ou bytes.
- Número de blocos na memória principal= 2s+ w/2w = 2s.
- Número de linhas na cache= indeterminado.
- Tamanho da tag= s bits.

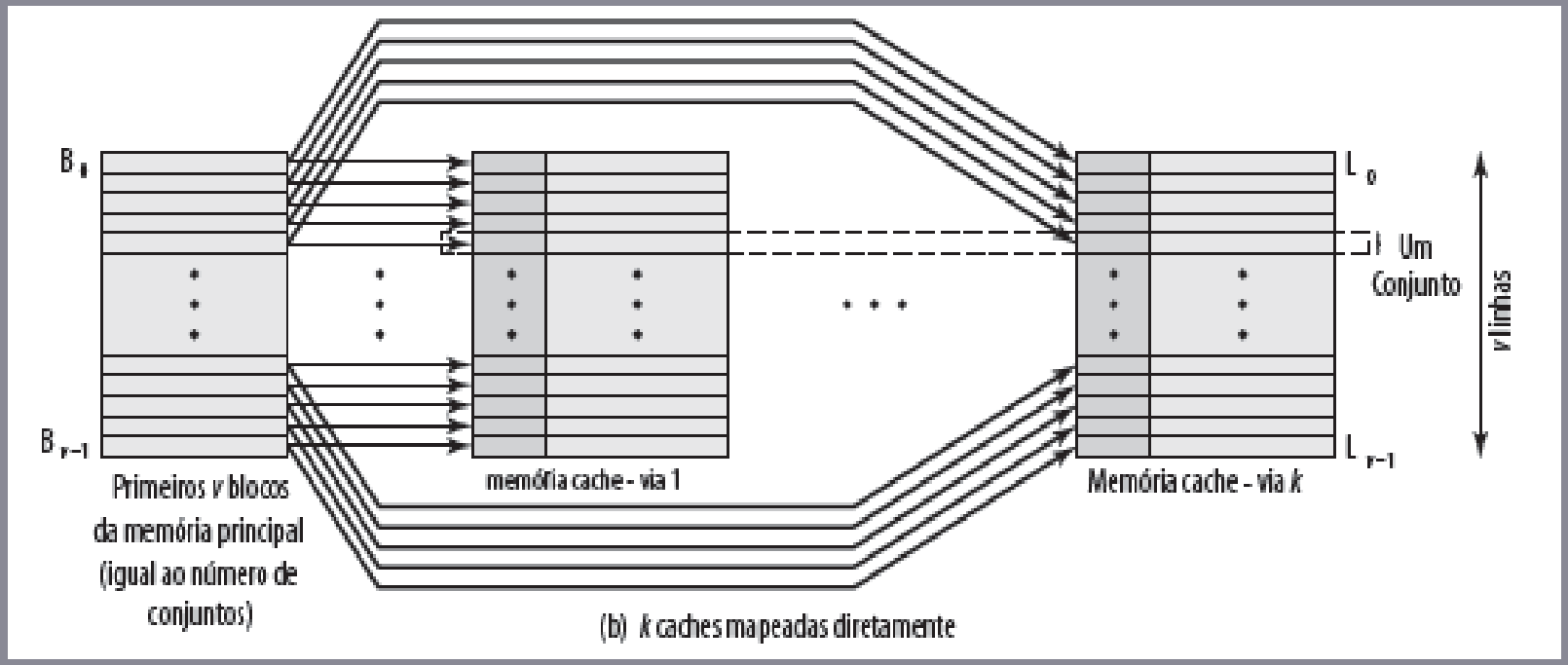
Mapeamento associativo em conjunto

- Cache é dividida em uma série de conjuntos.
- Cada conjunto contém uma série de linhas.
- Determinado bloco é mapeado a qualquer linha em determinado conjunto.
  - P.e.,Bloco B pode estar em qualquer linha do conjunto i.
- P.e., 2 linhas por conjunto:
  - Mapeamento associativo com 2 linhas.
  - Determinado bloco pode estar em uma de 2 linhas em apenas um conjunto.

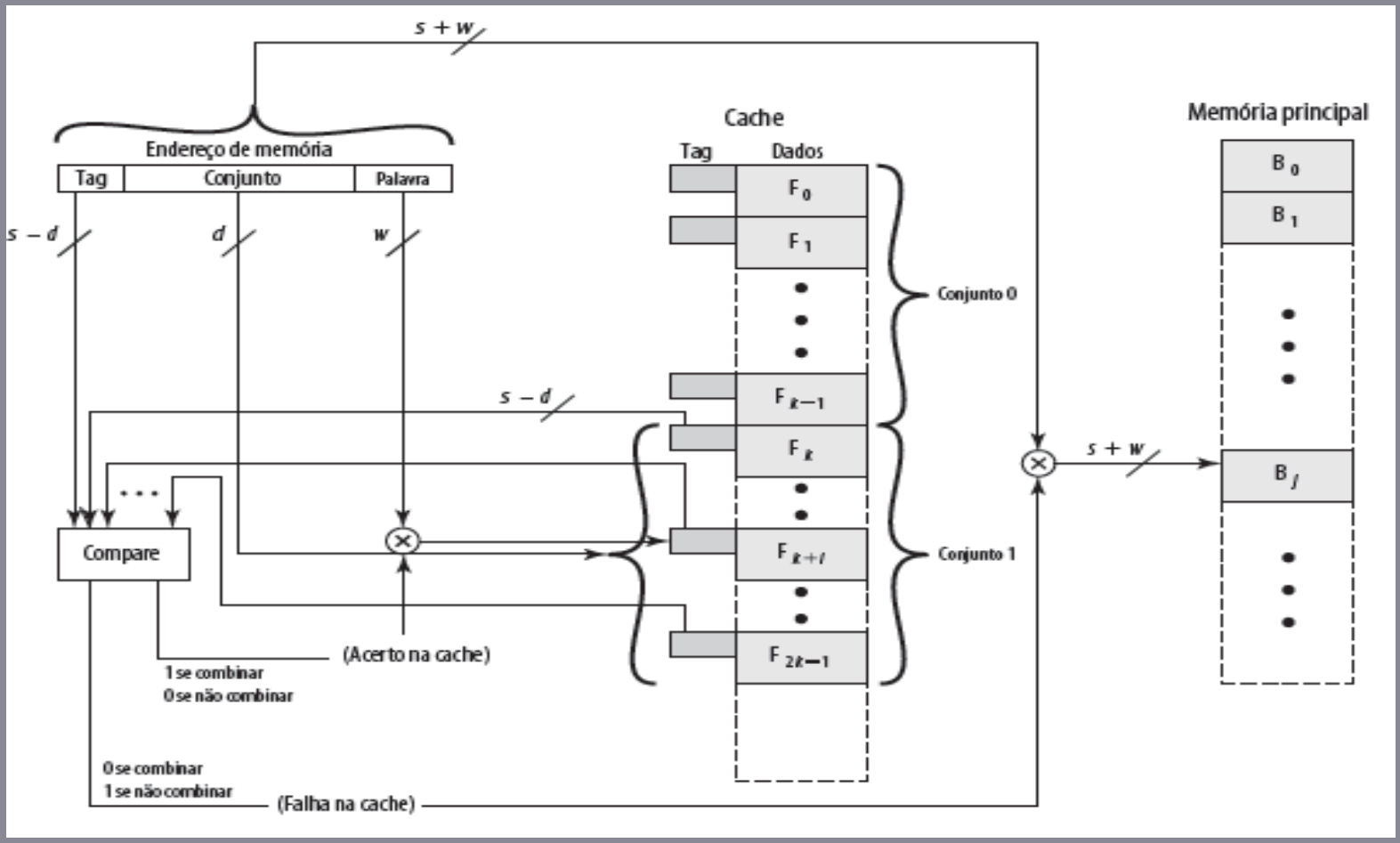
Mapeamento da memória principal para cache: associativo com *v* linhas



Mapeamento da memória principal para cache: associativo com *k* linhas



Organização da cache associativa em conjunto com *k* linhas



Mapeamento associativo em conjunto

Estrutura de endereços

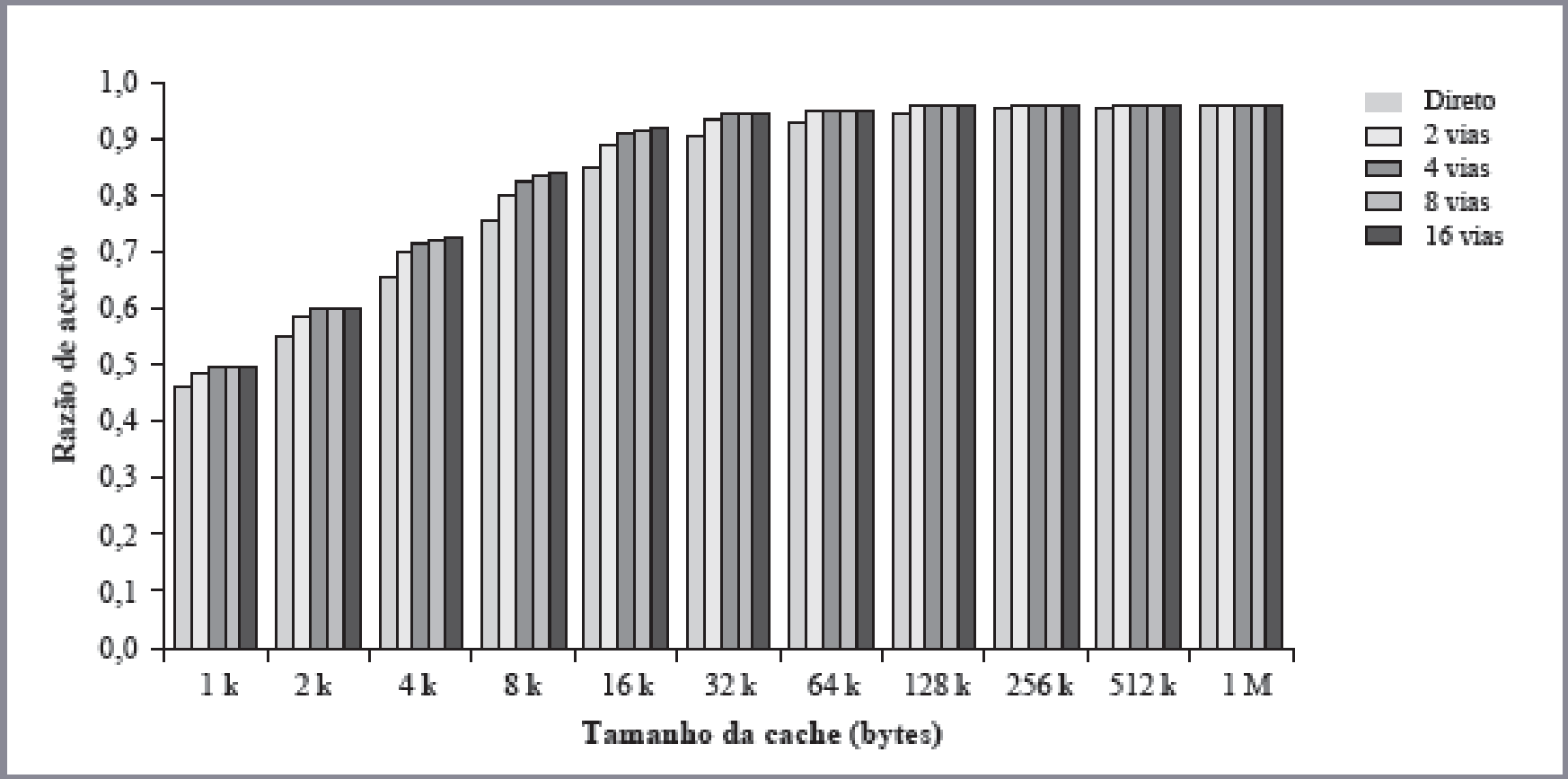
Tag 9 bits	Conjunto 13 bit	Palavra 2 bit

- Use campo de conjunto para determinar conjunto de cache a examinar.
- Compare campo de tag para ver se temos um acerto.
- P.e.,
  - Endereço Tag Dados Conjunto
  - 1FF 7FFC 1FF 12345678 1FFF
  - 001 7FFC 001 11223344 1FFF

Resumo de mapeamento associativo em conjunto

- Tamanho do endereço= (s + w) bits.
- Número de unidades endereçáveis= 2s+w palavras ou bytes.
- Tamanho do bloco= tamanho da linha= 2w palavras ou bytes.
- Número de blocos na memória principal= 2d.
- Número de linhas no conjunto= k.
- Número de conjuntos= v= 2d.
- Número de linhas na cache= kv= k * 2d.
- Tamanho da tag= (s – d) bits.

Associatividade variável pelo tamanho da cache



Algoritmos de substituição

Mapeamento direto

- Sem escolha.
- Cada bloco mapeado apenas a uma linha.
- Substitui essa linha.

Algoritmos de substituição

Associativa e associativa em conjunto

- Algoritmo implementado no hardware (velocidade).
- Least Recently Used (LRU).
  - Substitui o bloco com maior tempo sem ser referenciado/utilizado. MAIS POPULAR ENTRE OS DEMAIS
- First In First Out (FIFO).
  - Substitui bloco que está na cache há mais tempo.
- Least Frequently Used (LFU).
  - Substitui bloco que teve menos acertos.
- Aleatório.

Política de escrita

- Não deve sobrescrever bloco de cache a menos que a memória principal esteja atualizada.
- Múltiplas CPUs podem ter caches individuais.
- E/S pode endereçar memória principal diretamente.

Write-through

- Todas as escritas vão para a memória principal e também para a cache.
- Múltiplas CPUs podem monitorar o tráfego da memória principal para manter a cache local (à CPU) atualizada.
- Muito tráfego.
- Atrasa as escritas.
- Lembre-se de caches *write-through* falsos!

Write-back

- Atualizações feitas inicialmente apenas na cache.
- Bit de atualização para slot de cache é definido quando ocorre a atualização.
- Se o bloco deve ser substituído, escreve na memória principal apenas se o bit atualizado estiver marcado.
- Outras caches saem de sincronismo.
- E/S deve acessar a memória principal através da cache.
- 15% das referências de memória são escritas.

Tamanho de linha

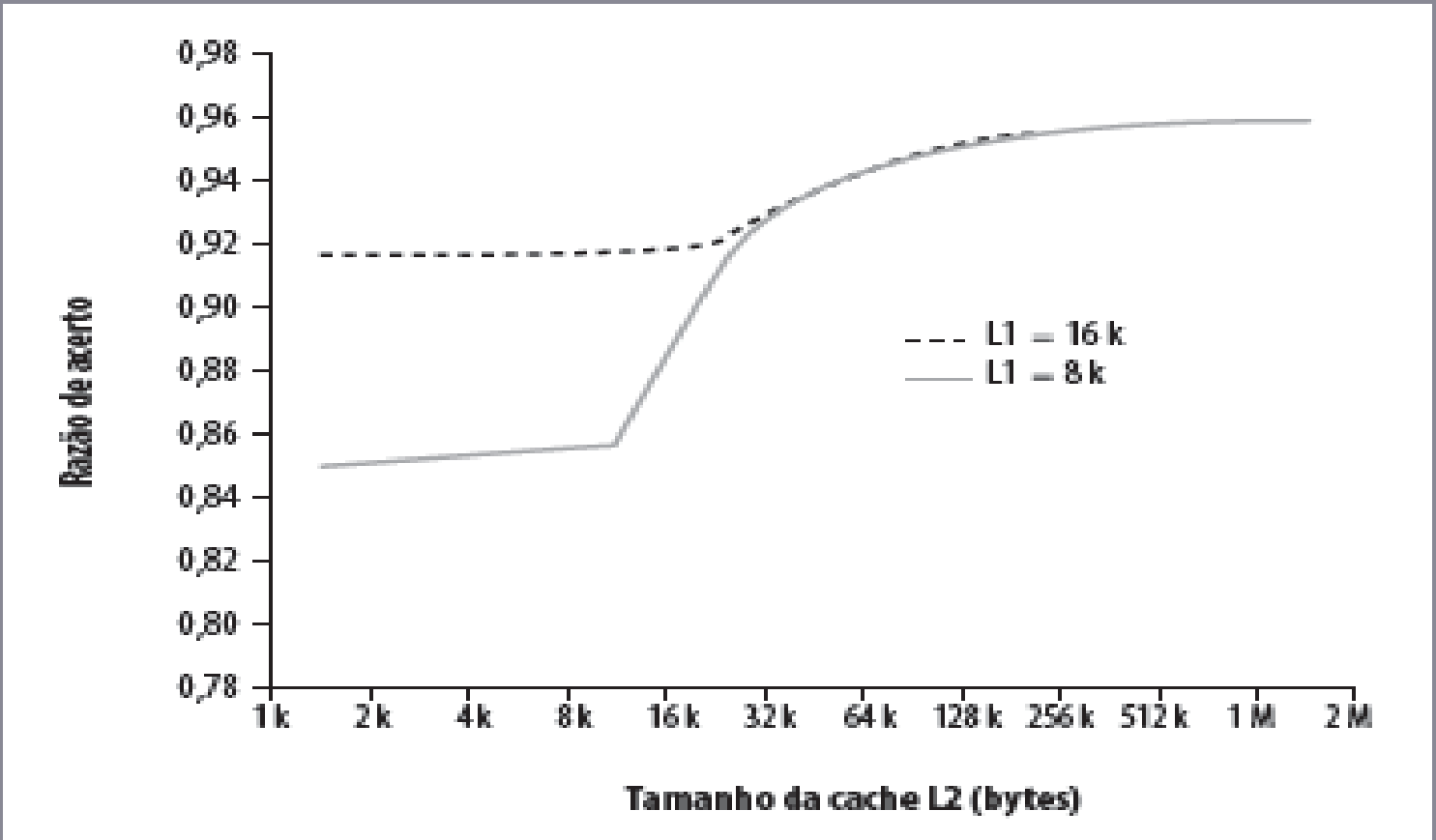
- Recupere não apenas a palavra desejada, mas também uma série de palavras adjacentes.
- Tamanho de bloco aumentado aumentará razão de acerto a princípio.
  - O princípio da localidade.
- Razão de acerto diminuirá à medida que o bloco se tornar ainda maior.
  - Probabilidade de uso de informações recém-buscadas torna-se menor que probabilidade de reutilizar informações substituídas.

- Blocos maiores:
  - Reduzem número de blocos que cabem na cache.
  - Dados sobrescritos pouco depois de serem buscados.
  - Cada palavra adicional é menos local, de modo que é menos provável de ser necessária.
- Nenhum valor ideal definitivo foi descoberto.
- 8 a 64 bytes parece ser razoável.
- Para sistemas HPC, 64 e 128 bytes mais comum.

Caches multinível

- Alta densidade lógica permite caches no chip.
  - Mais rápido que acesso ao barramento.
  - Libera barramento para outras transferências.
- Comum usar cache dentro e fora do chip.
  - L1 no chip, L2 fora do chip na RAM estática.
  - Acesso L2 muito mais rápido que DRAM ou ROM.
  - L2 normalmente usa caminho de dados separado.
  - L2 pode agora estar no chip.
  - Resultando em cache L3.
    - Acesso ao barramento agora no chip.

Razão de acerto total (L1 & L2) Para L1 de 8 KB e 16 KB



Caches unificadas *versus* separadas

- Uma cache para dados e instruções ou duas, uma para dados e uma para instruções.
- Vantagens da cache unificada:
  - Maior taxa de acerto.
    - Equilibra carga entre buscas de instrução e dados.
    - Apenas uma cache para projetar e implementar.
- Vantagens da cache separada:
  - Elimina disputa pela cache entre a unidade de busca/decodificação de instrução e a unidade de execução.
    - Importante no pipeline de instruções.

Pentium 4 – cache

- 80386 – nenhuma cache no chip.
- 80486 – 8 KB usando linhas de 16 bytes organização associativa em conjunto com 4 linhas.
- Pentium (todas as versões) – duas caches L1 no chip.
  - Dados e instruções:
- Pentium III – cache L3 adicionada fora do chip.
- Pentium 4:
  - Caches L1.
    - 8 KB.
    - Linhas 64 bytes.
    - Associativa em conjunto com 4 linhas.

- Cache L2:
  - Alimentando ambas as caches L1.
  - 256k.
  - Linhas de 128 bytes.
  - Associativa em conjunto com 8 linhas.
- Cache L3 vai para o chip.

Processador Pentium 4 Core

- Unidade de busca/decodificação:
  - Busca instruções da cache L2.
  - Decodifica para micro-operações.
  - Armazena micro-operações na cache L1.
- Lógica de execução fora de ordem:
  - Escalona micro-operações.
  - Baseada em dependência de dados e recursos.
  - Pode executar especulativamente.
- Unidades de execução:
  - Executa micro-operações.
  - Dados da cache L1.
  - Resultados em registradores.
- Subsistema de memória.
  - Cache L2 e barramento do sistema.

Raciocínio de projeto do Pentium 4

- Decodifica instruções para RISC como micro-operações antes da L1.
- Micro-operações de tamanho fixo.
  - Pipelining e escalonamento superescalar.
- Instruções Pentium longas e complexas.
- Desempenho melhorado separando decodificação do escalonamento e pipelining.
  - (Mais adiante – Capítulo 14)

- Cache de dados é *write-back.*
  - Pode ser configurada para *write-through*.
- Cache L1 controlada por 2 bits no registrador.
  - CD= Cache Disable.
  - NW= Not *write-through*.
  - 2 instruções para invalidar (esvaziar) cache e *write-back* depois invalidação.
- L2 e L3 associativas em conjunto com 8 linhas.
  - Tamanho de linha 128 bytes.

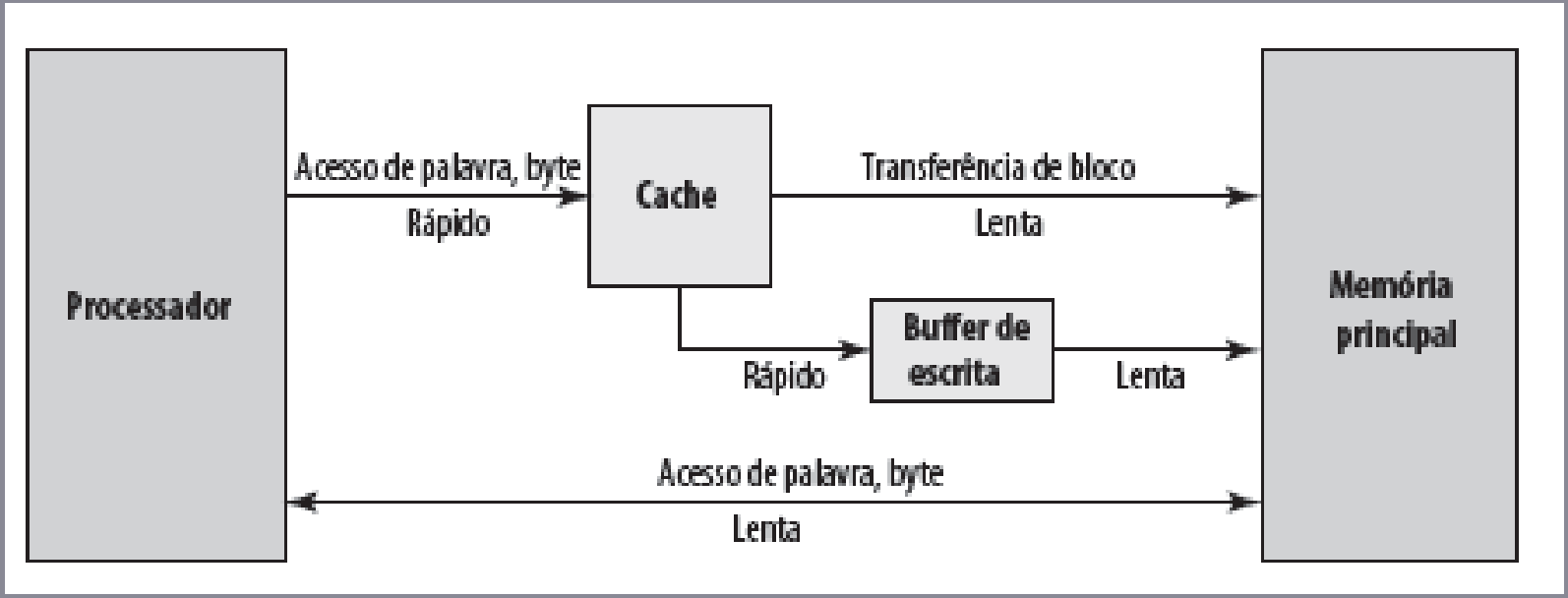
Características da memória cache do ARM

Core	Cache Type	Cache Size (kB)	Cache Line Size (words)	Associativity	Location	Write Buffer Size (words)
ARM720T	Unified	8	4	4-way	Logical	8
ARM920T	Split	16/16 D/I	8	64-way	Logical	16
ARM926EJ-S	Split	4-128/4-128 D/I	8	4-way	Logical	16
ARM1022E	Split	16/16 D/I	8	64-way	Logical	16
ARM1026EJ-S	Split	4-128/4-128 D/I	8	4-way	Logical	8
Intel StrongARM	Split	16/16 D/I	4	32-way	Logical	32
Intel Xscale	Split	32/32 D/I	8	32-way	Logical	32
ARM1136-JF-S	Split	4-64/4-64 D/I	8	4-way	Physical	32

Organização de cache da ARM

- Pequeno buffer de escrita FIFO.
  - Melhora o desempenho de escrita da memória.
  - Entre cache e memória principal.
  - Pequena cache c.f.
  - Dados colocados no buffer de escrita na velocidade de clock do processador.
  - Processador continua a execução.
  - Escrita externa em paralelo até vazio.
  - Se buffer encher, processador adiado (*stall*).
  - Dados no buffer de escrita não disponíveis até serem escritos.
    - Assim, mantém buffer pequeno.

Organização da cache e do buffer de escrita do ARM



Fontes na Internet

- Sites de fabricantes:
  - Intel.
  - ARM.
- Procure sobre cache.