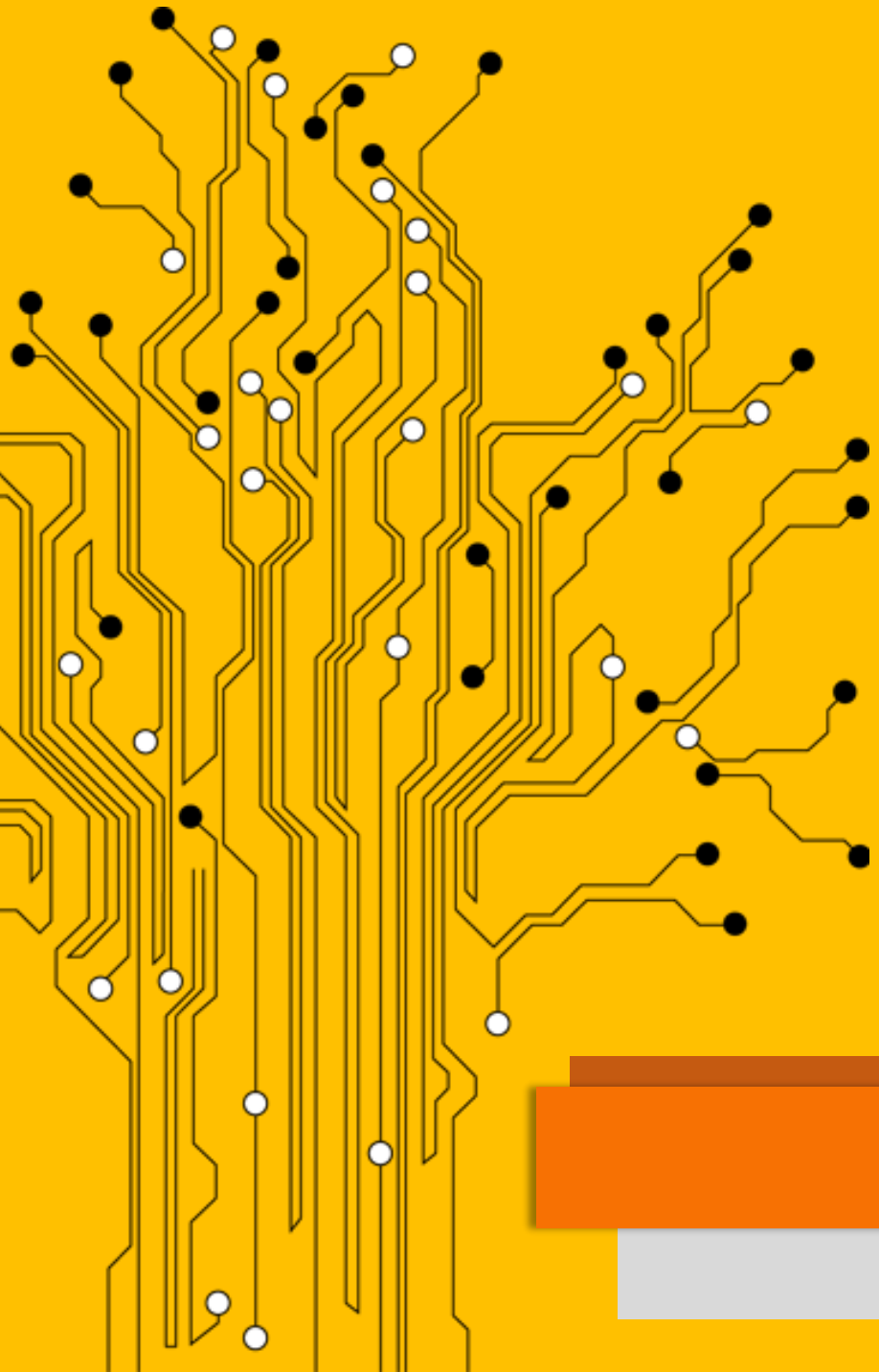


Arquitetura de computadores

MEMÓRIA CACHE

FELIPE G. TORRES

MEMÓRIA CACHE

O uso da memória cache visa obter velocidade de memória próxima das memórias mais rápidas que existem e, ao mesmo tempo, disponibilizar uma memória de grande capacidade ao preço de memórias semicondutoras mais baratas.

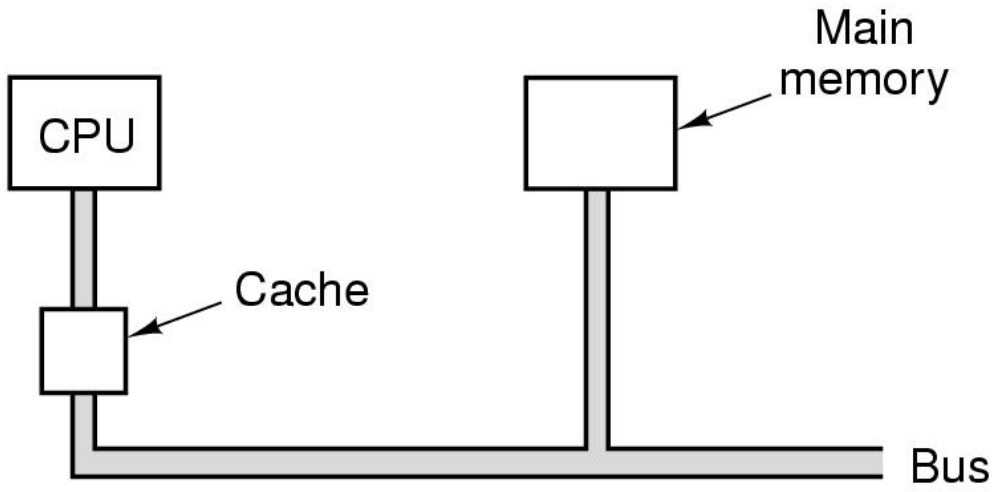
Existe uma memória principal relativamente grande e lenta junto com a memória cache, menor e mais rápida.

A memória cache contem uma cópia de partes da memória principal.

MEMÓRIA CACHE

Quando o processador tenta ler uma palavra da memória, é feita uma verificação para determinar se a palavra está na cache. Se estiver, ela é entregue ao processador.

Se não, um bloco da memória principal, consistindo em algum número fixo de palavras, é lido para a cache e depois a palavra é fornecida ao processador.



REVISÃO E CURIOSIDADES

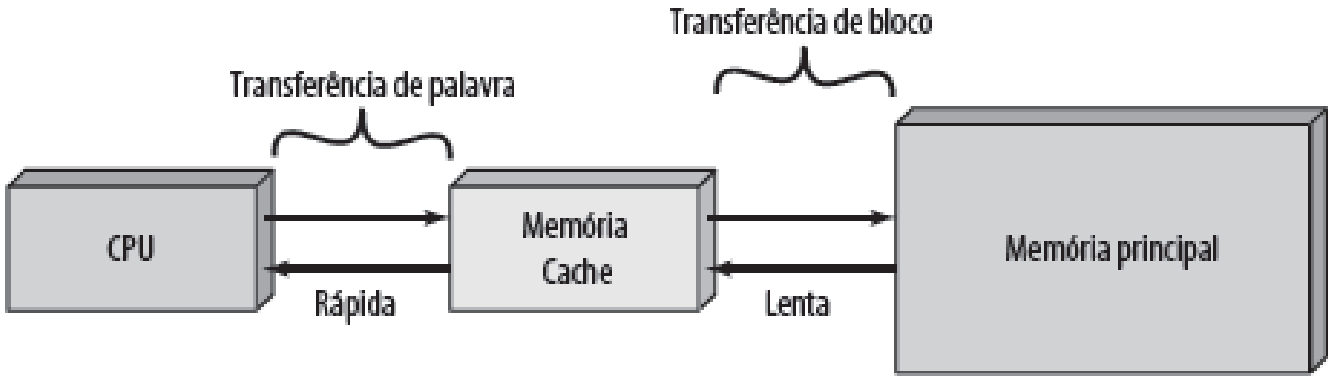
ORGANIZAÇÃO DA MEMÓRIA CACHE

Devido ao fenômeno de localidade de referência, quando um bloco de dados é levado para a cache para satisfazer uma única referência de memória, é provável que haja referências futuras a esse mesmo local da memória ou a outras palavras no mesmo bloco.

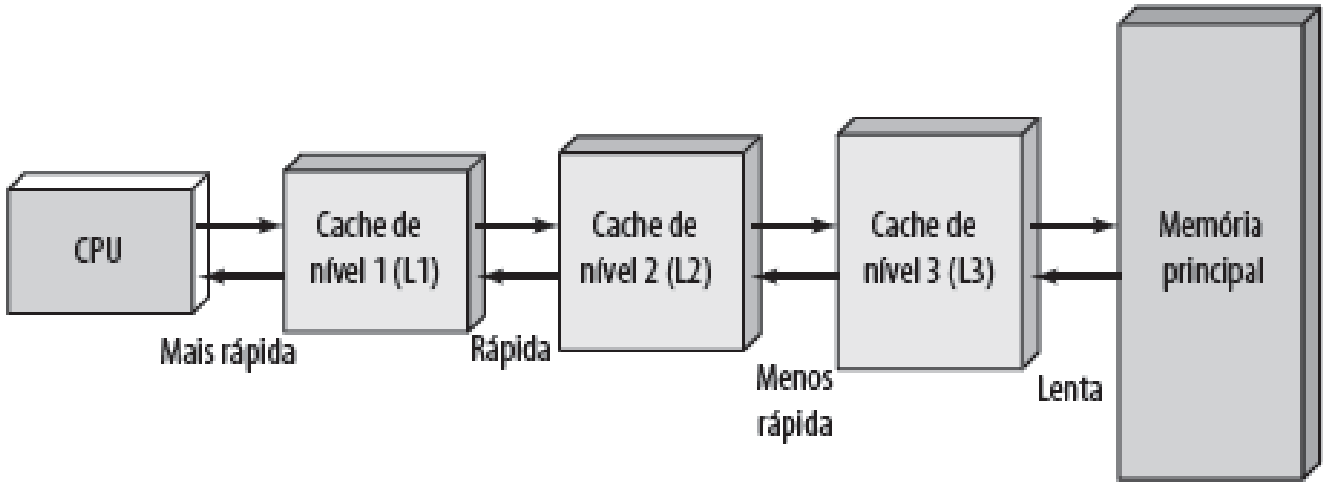
Por isso existe o conceito de múltiplos níveis de cache.

A cache L2 é mais lenta e normalmente maior que a cache L1, e a cache L3 é mais lenta e normalmente maior que a cache L2.

ORGANIZAÇÃO DA MEMÓRIA CACHE



(a) Cache única



(b) Organização de cache em três níveis

ESTRUTURA DE MEMÓRIA CACHE / MEMÓRIA PRINCIPAL

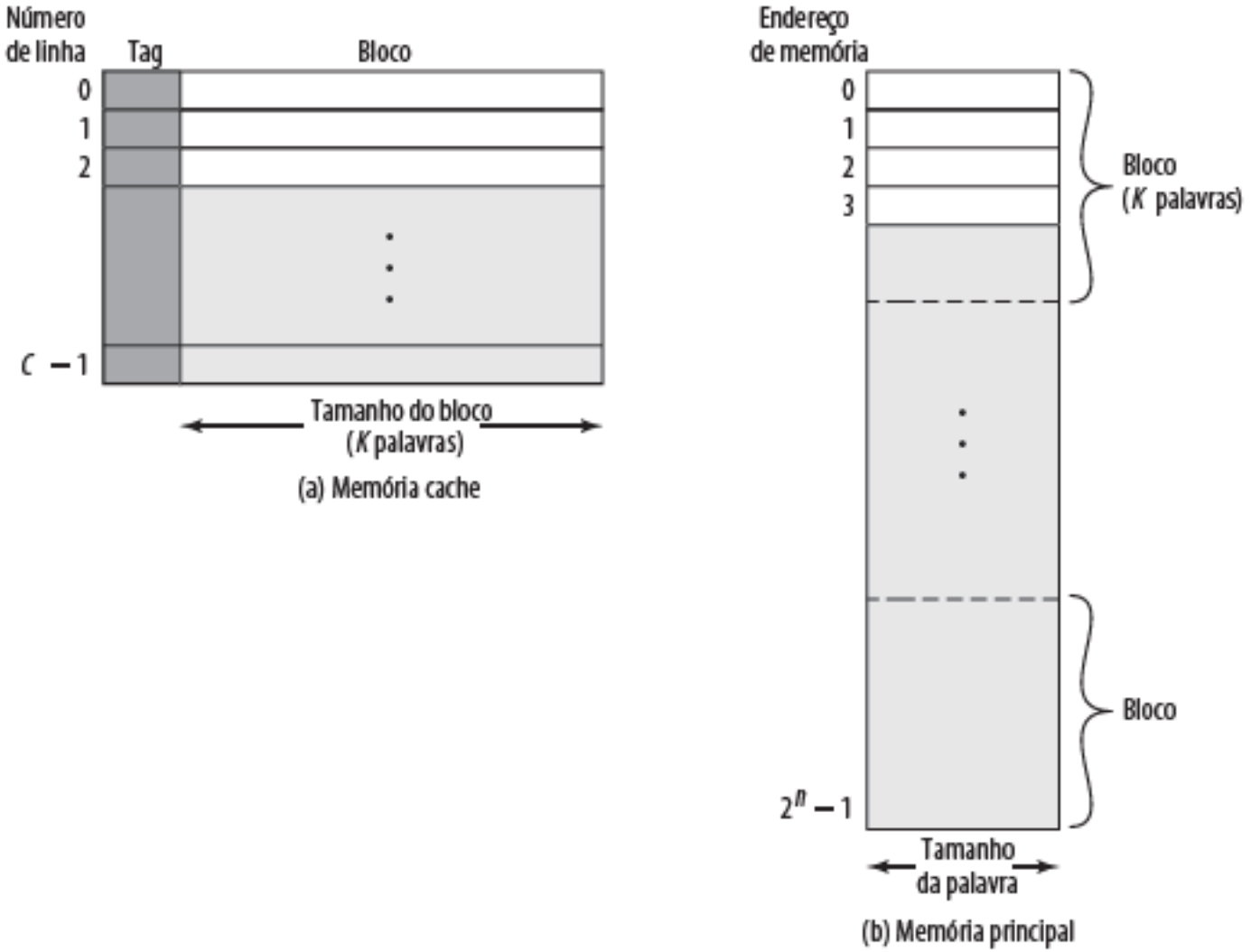
A memória principal consiste em ate 2*n* palavras endereçáveis, com cada palavra tendo um endereço distinto de *n* bits.

Para fins de mapeamento, essa memoria e considerada como sendo uma série de blocos de tamanho fixo com *K* palavras cada.

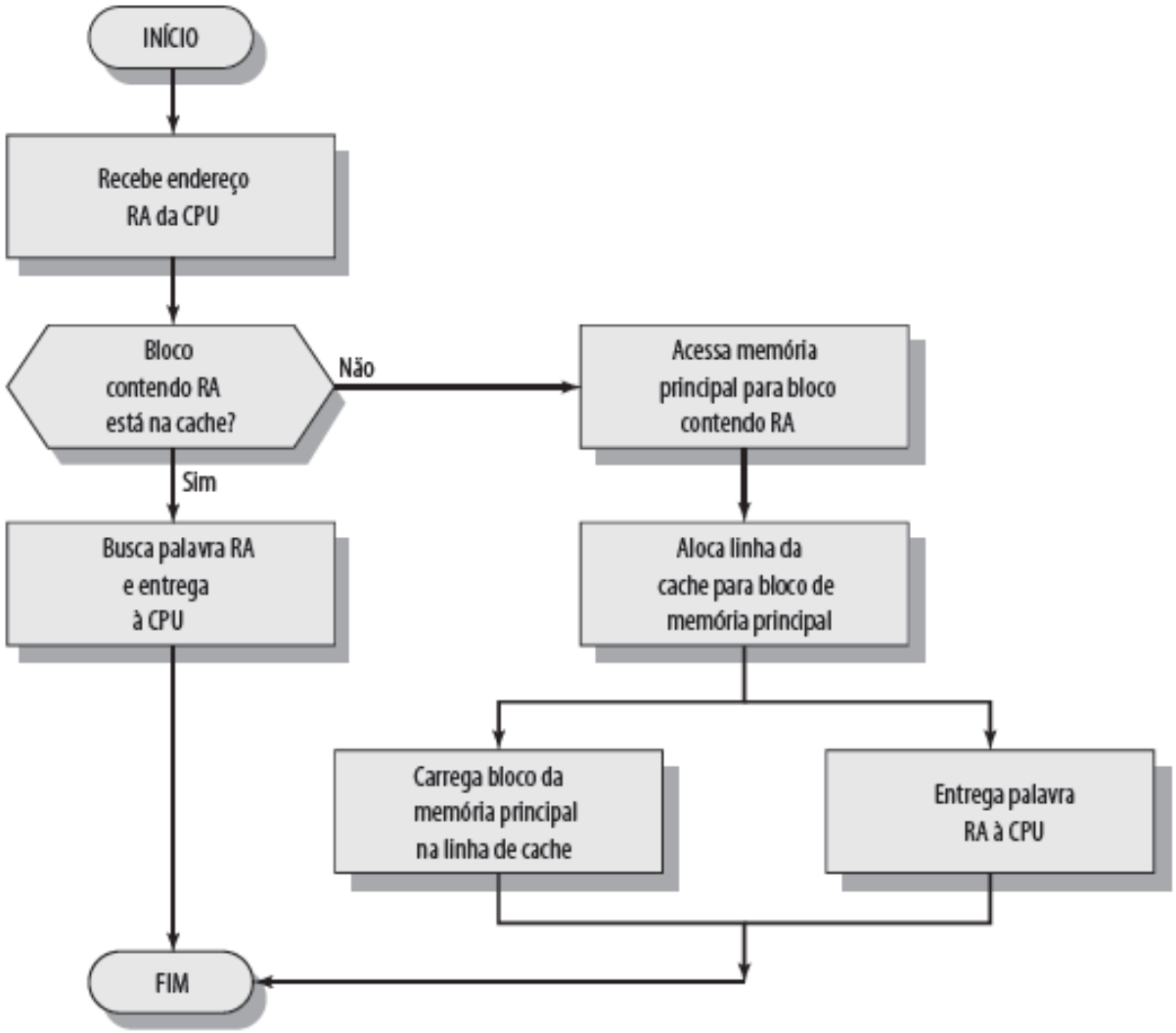
Existem M = 2n/K blocos na memória principal. A cache consiste em m blocos, chamados linhas.

Cada linha contém K palavras, mais um tag de alguns bits.

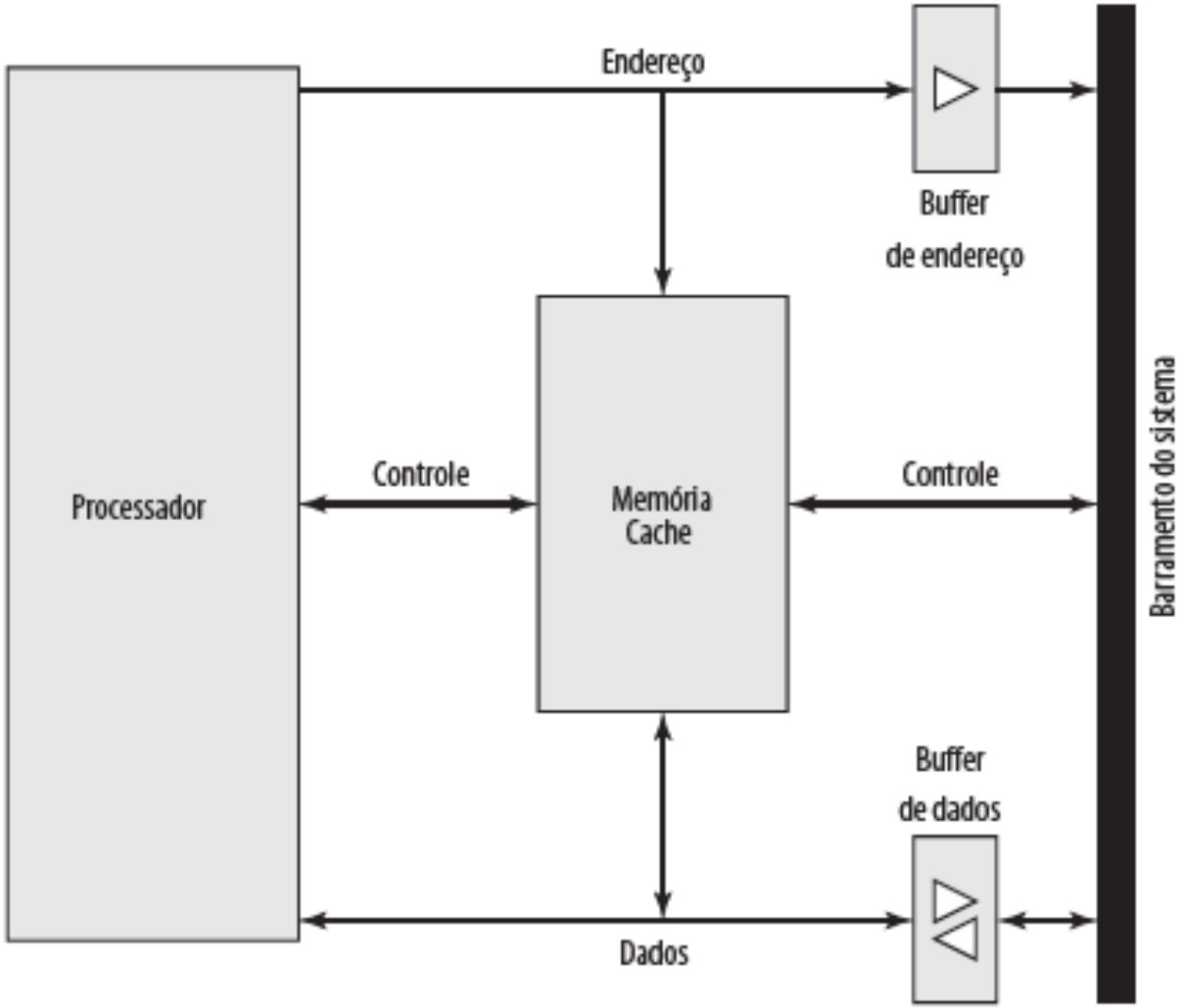
ESTRUTURA DE MEMÓRIA CACHE / MEMÓRIA PRINCIPAL



OPERAÇÃO DE LEITURA DO CACHE



ORGANIZAÇÃO TÍPICA DA MEMÓRIA CACHE



TAMANHO DA MEMÓRIA CACHE EM PROCESSADORES

Processador	Tipo	Ano de introdução	Cache L1[a]	Cache L2	Cache L3
IBM 360/85	Mainframe	1968	16 a 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 a 256 KB	—	—
Intel 80486	PC	1989	8 KB	—	—
Pentium	PC	1993	8 KB/8 KB	256 a 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 G4	Mainframe	1997	32 KB	256 KB	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
SGI Origin 2001	Servidor avançado	2001	32 KB/32 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 a 48 MB

FUNÇÃO DE MAPEAMENTO

Como existem menos linhas de cache do que blocos da memória principal, é necessário haver um algoritmo para mapear os blocos da memória principal às linhas de cache.

A escolha da função de mapeamento dita como a cache é organizada. Três técnicas podem ser utilizadas:

- Direta
- Associativa

FUNÇÃO DE 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 % m, onde

i = número da linha da cache

j = número do bloco da memória principal

m = número de linhas da cache

FUNÇÃO DE MAPEAMENTO DIRETO

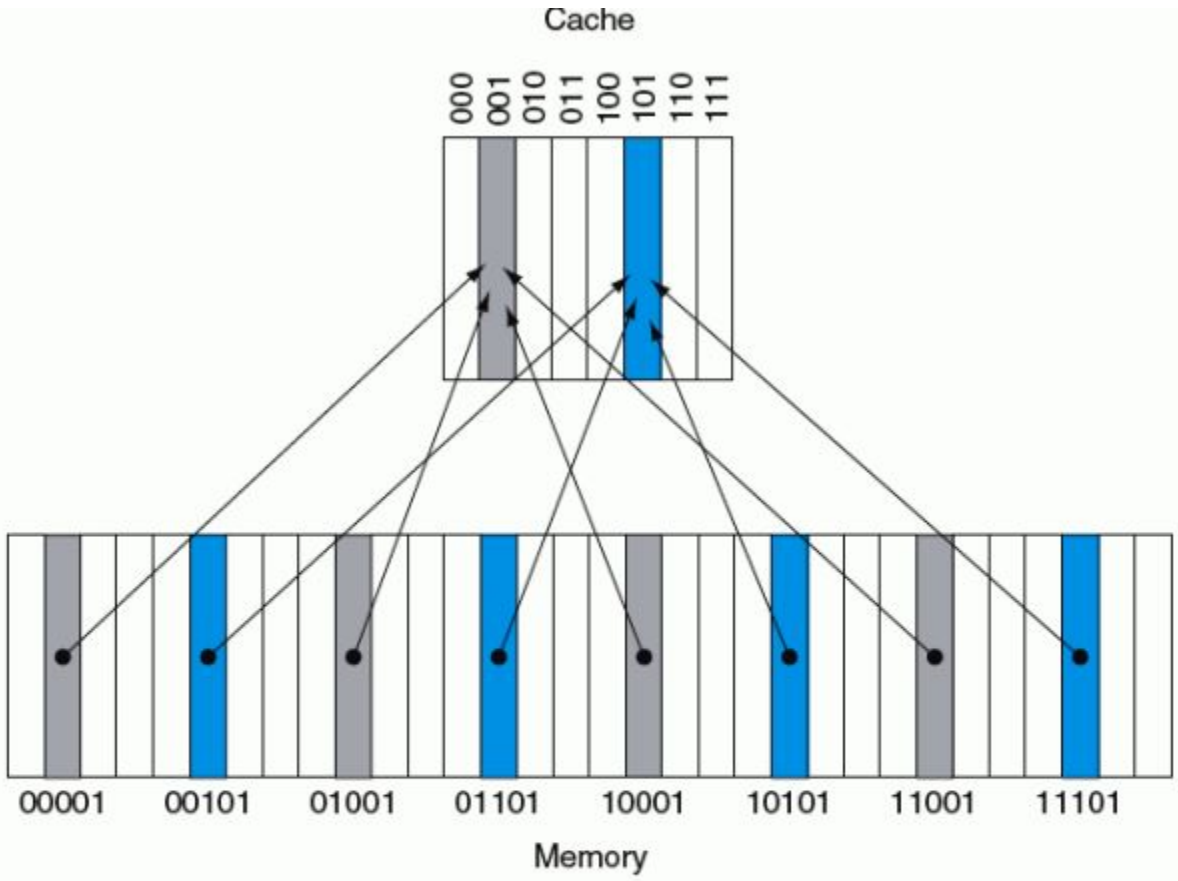
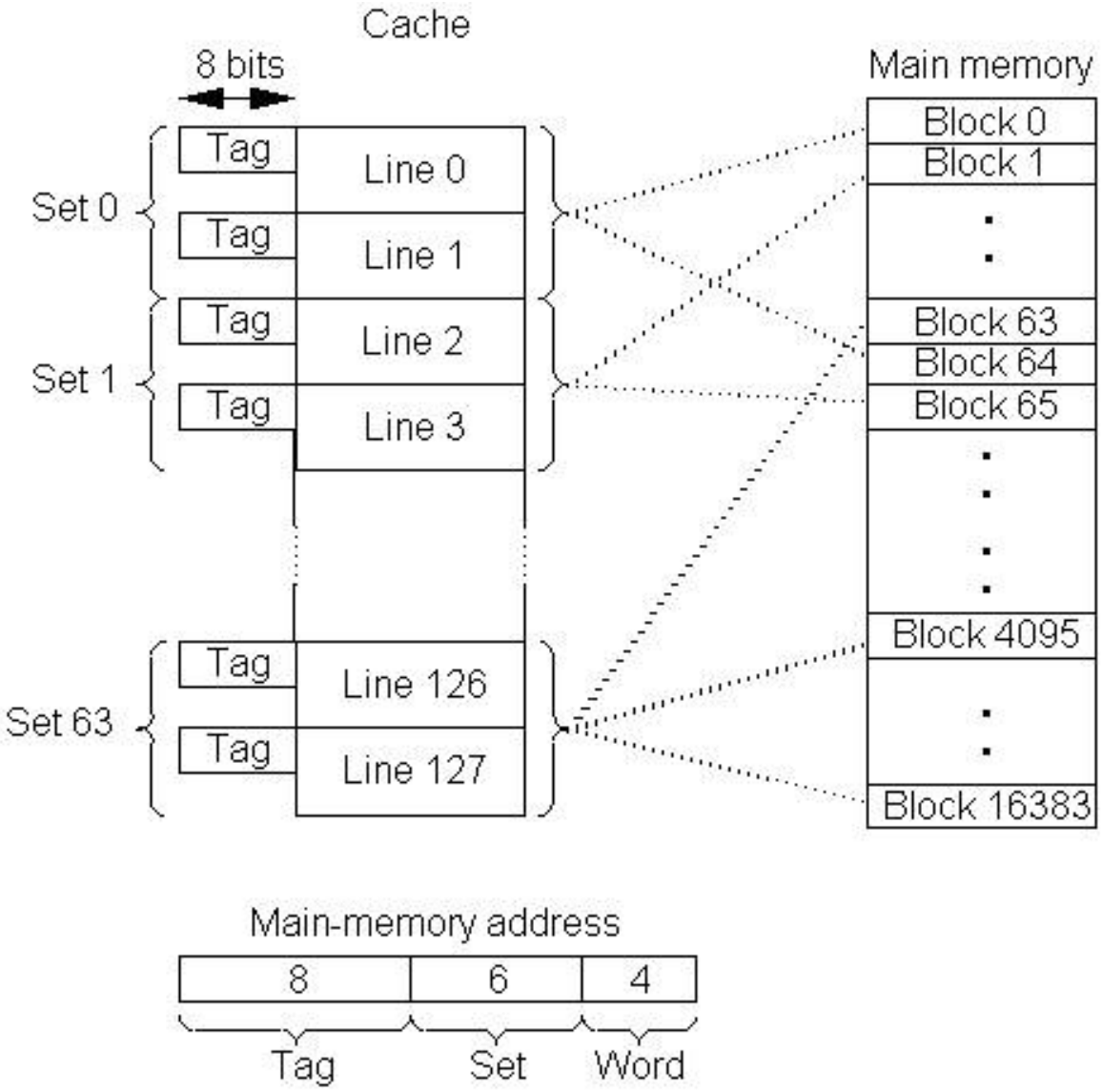


FIGURE 7.5 A direct-mapped cache with eight entries showing the addresses of memory words between 0 and 31 that map to the same cache locations. Because there are eight words in the cache, an address X maps to the cache word X modulo 8. That is, the low-order $\log_2(8) = 3$ bits are used as the cache index. Thus, addresses 00001_{two}, 01001_{two}, 10001_{two}, and 11001_{two} all map to entry 001_{two} of the cache, while addresses 00101_{two}, 01101_{two}, 10101_{two}, and 11101_{two} all map to entry 101_{two} of the cache.

FUNÇÃO DE MAPEAMENTO DIRETO



DESVANTAGENS DO MAPEAMENTO DIRETO

A técnica de mapeamento direto é simples e pouco dispendiosa para se implementar.

Sua principal desvantagem é que existe um local de cache fixo para cada bloco.

Se um programa referenciar palavras repetidamente de dois blocos diferentes, mapeados para a mesma linha, então os blocos serão continuamente trocados na cache, e a razão de acerto será baixa (um fenômeno conhecido como **thrashing**).

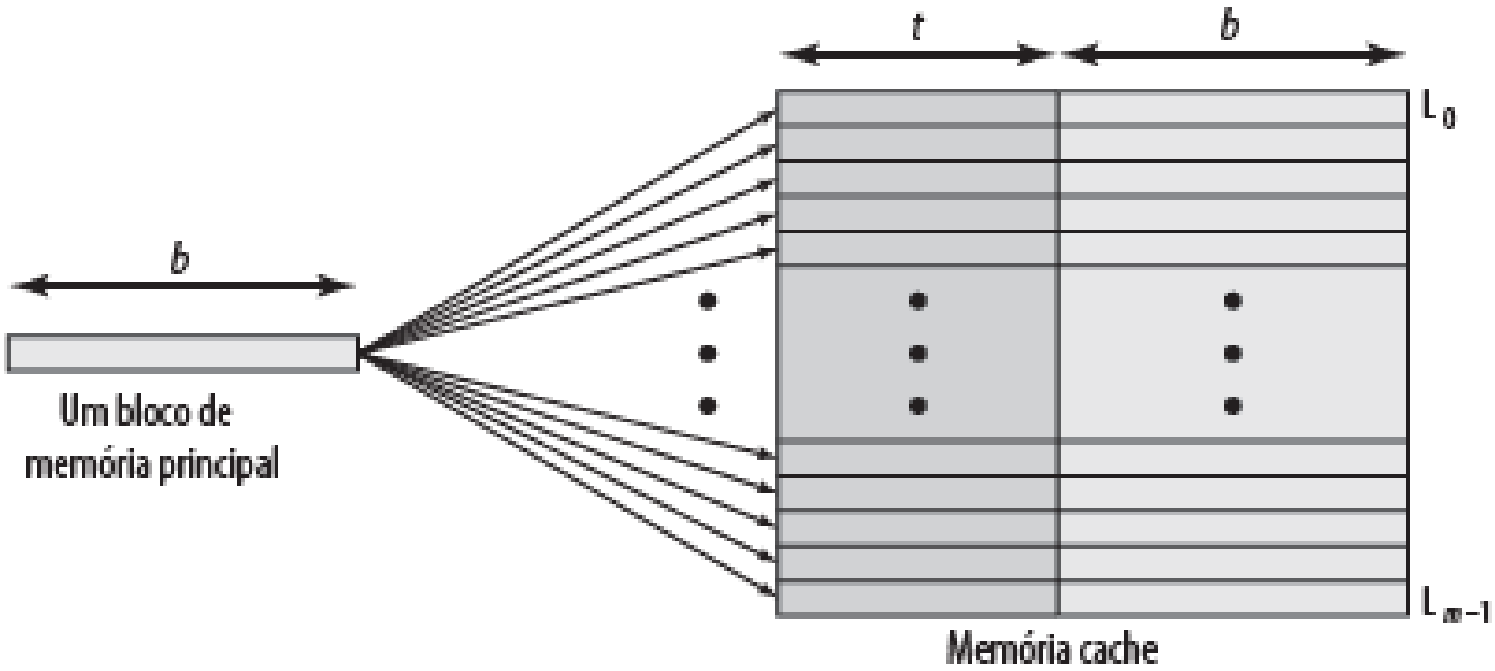
FUNÇÃO DE MAPEAMENTO ASSOCIATIVA

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.

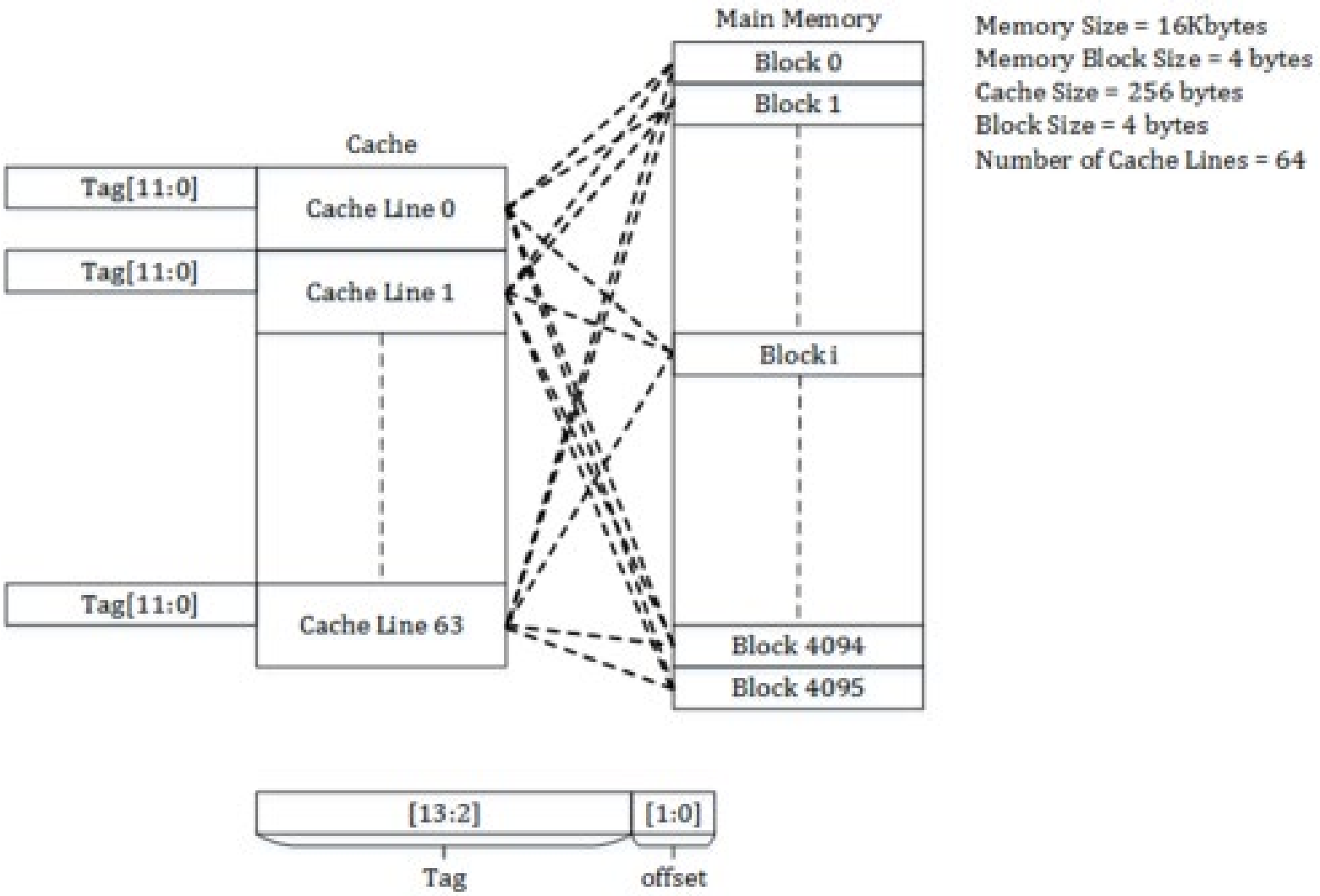
O campo Tag identifica o bloco da memória principal. Para determinar se um bloco está na cache, a lógica de controle da cache precisa comparar simultaneamente a tag de cada linha

FUNÇÃO DE MAPEAMENTO ASSOCIATIVA



(b) Mapeamento associativo

FUNÇÃO DE MAPEAMENTO ASSOCIATIVA



FUNÇÃO DE MAPEAMENTO ASSOCIATIVA

Com o mapeamento associativo, existe flexibilidade em relação a qual bloco substituir quando um novo bloco for lido para a cache.

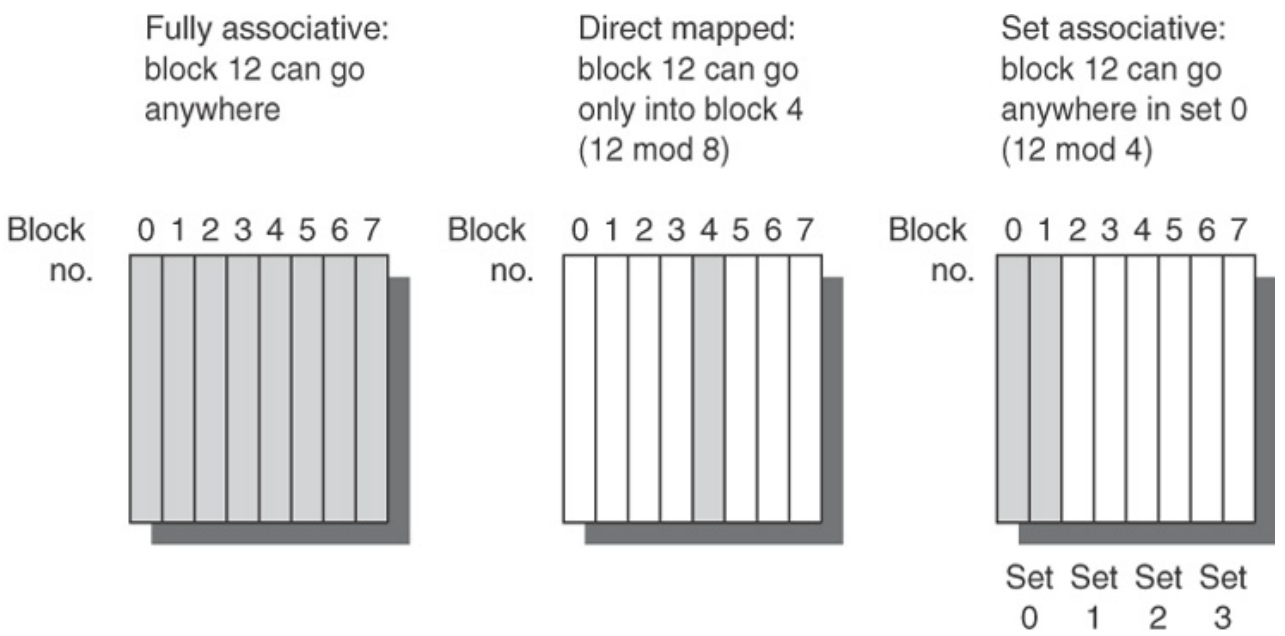
Os algoritmos de substituição, discutidos mais adiante nesta seção, são projetados para maximizar a razão de acerto.

A principal desvantagem do mapeamento associativo é a complexidade do circuito necessário para comparar as tags de todas as linhas da cache em paralelo.

Observe que nenhum campo no endereço corresponde ao número de linha, de modo que o número de linhas na cache não é determinado pelo formato do endereço.

COMPARAÇÃO ENTRE OS TRÊS MAPEAMENTOS DE CACHE

Cache



Main memory

ALGORITMOS DE SUBSTITUIÇÃO

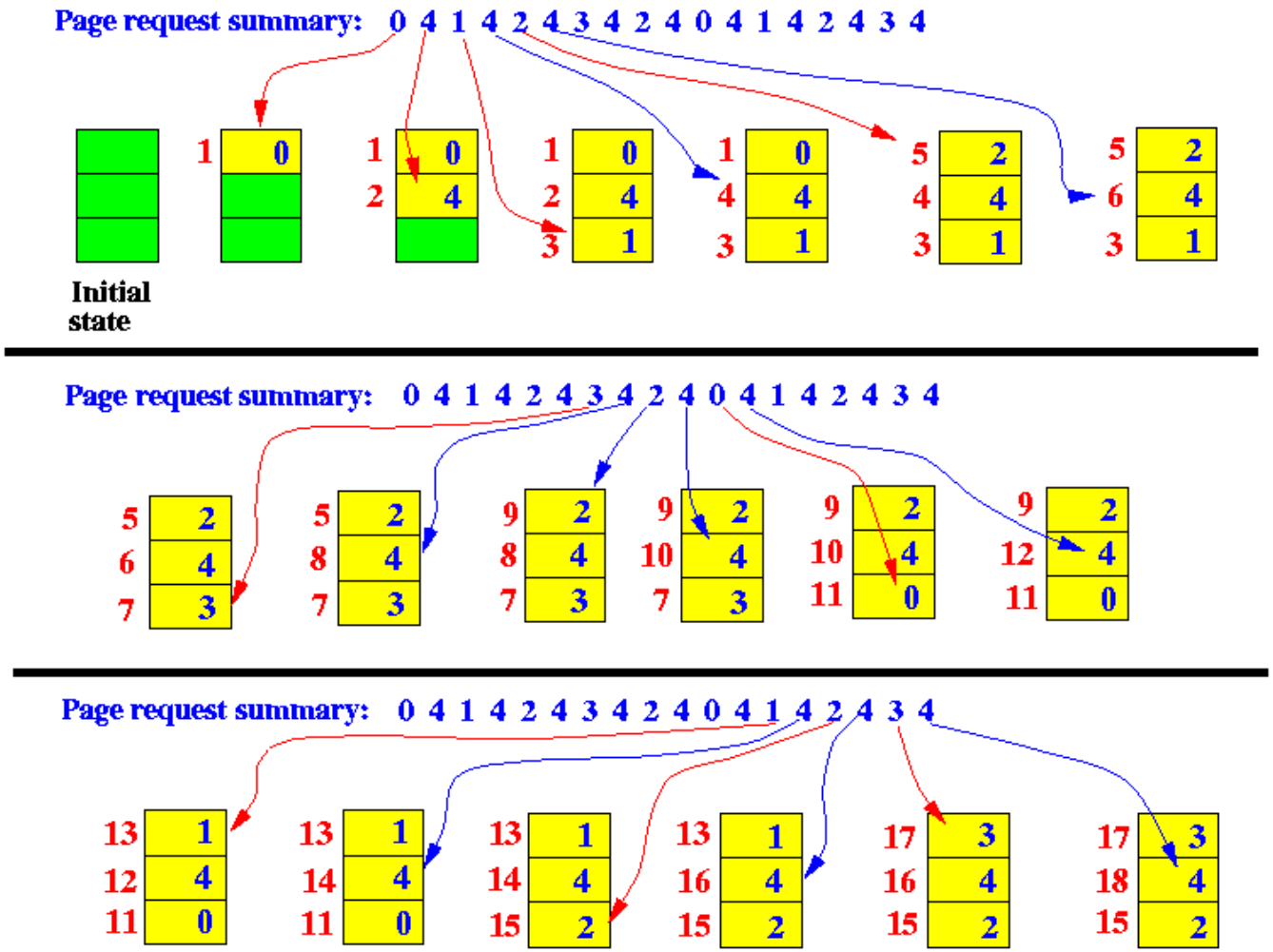
Uma vez que a cache estiver cheia, e um novo bloco for trazido para a cache, um dos blocos existentes precisa ser substituído.

Para o mapeamento direto, existe apenas uma linha possível para qualquer bloco em particular e nenhuma escolha é possível.

Para o mapeamento associativo é necessário que seja utilizado um algoritmo de substituição e precisa ser implementado em hardware.

Provavelmente, o mais efi caz seja o usado menos recentemente **(LRU, do inglês least recently used):** substitua aquele bloco no conjunto que permaneceu na cache por mais tempo sem qualquer referência a ele.

ALGORITMOS LEAST RECENTLY USED (LRU)



POLÍTICA DE ESCRITA

Quando um bloco que está residente na cache estiver para ser substituído, existem dois casos a considerar.

- Se o bloco antigo na cache não tiver sido alterado, então ele pode ser substituído por um 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.

- A técnica mais simples para essa tarefa é denominada **write-through.**

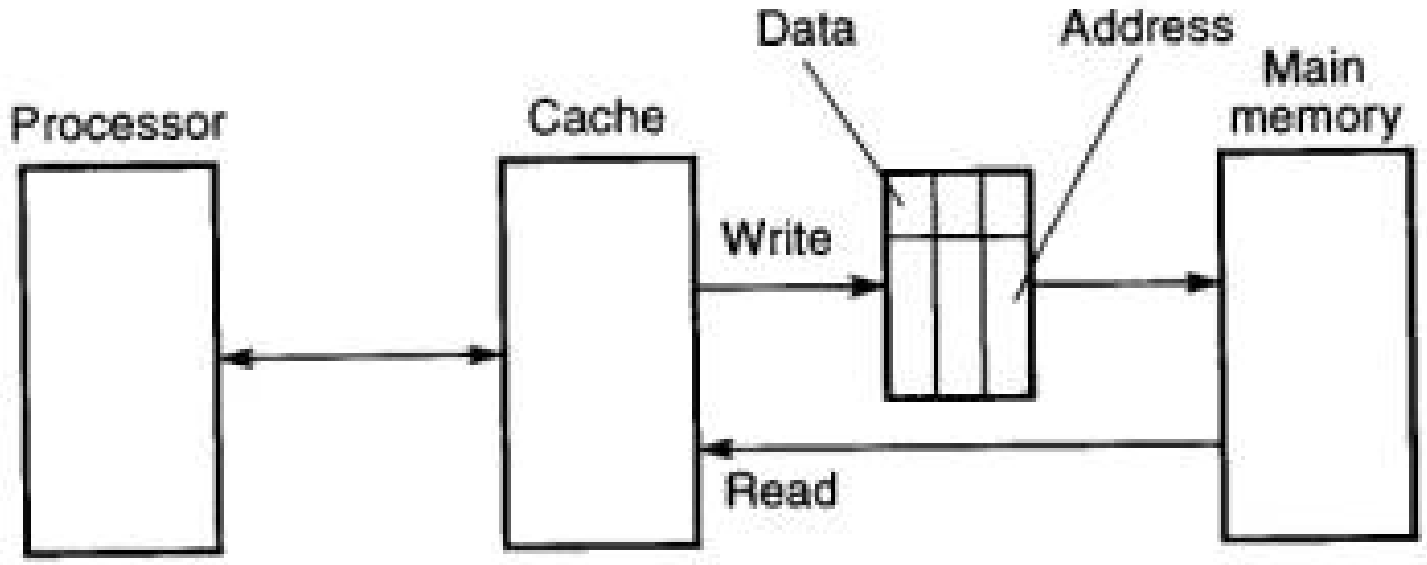
WRITE-THROUGH

Usando essa técnica, todas as operações de escrita são feita na memória principal e também na cache, garantindo que a memória principal sempre seja válida.

Qualquer outro módulo processador-cache pode monitorar o tráfego para a memória principal para manter a consistência dentro de sua própria cache.

A principal desvantagem dessa técnica é que ela gera um tráfego de memória considerável e podendo vir a ser um gargalo.

WRITE-THROUGH



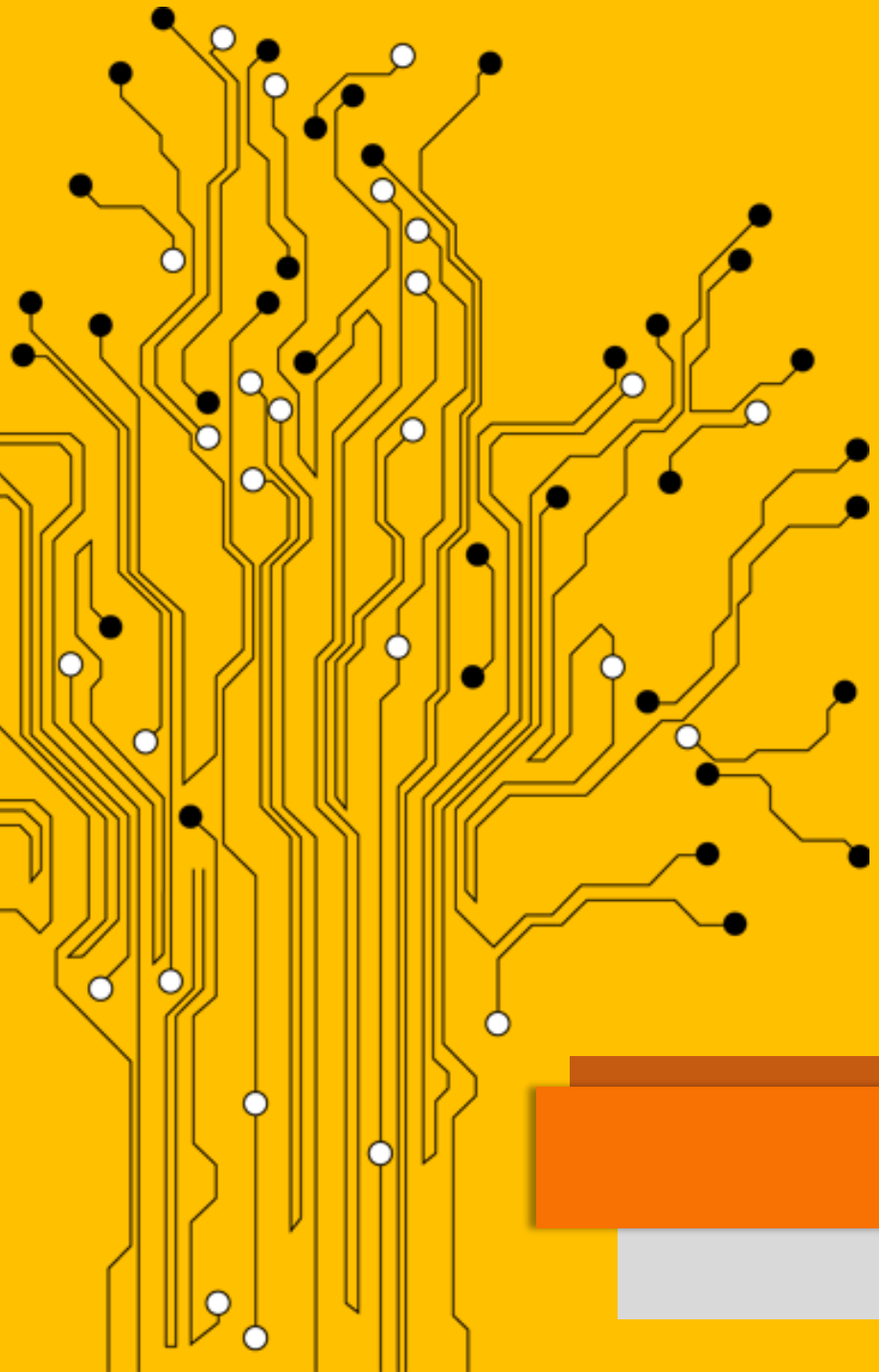
REFERÊNCIAS

STALLINGS, William. **Arquitetura e organização de computadores: projeto para o desempenho**. 8 ed. São Paulo: Prentice Hall : Person Education, 2010. 624 p. ISBN 9788576055648.

TANENBAUM, Andrew S. **Organização estruturada de computadores**. 5. ed São Paulo: Pearson Prentice Hall, 2007. 449 p. ISBN 9788576050674.

VÍDEOS

MEMÓRIA CACHE: o que é? Vale a pena limpar o cache? [Disponível em https://youtu.be/b0jlCzbXZFw]



Arquitetura de computadores

MEMÓRIA CACHE

FELIPE G. TORRES