Memórias

Tipos de Memórias

- Memória Cache;
- Memória interna;
- Memória externa;

Características das memórias

- Localização;
- Capacidade;
- Unidade de transferência;
- Método de acesso;

Características das memórias

- Desempenho;
- Tipo físico;
- Detalhes físicos.

Caracterísitca: Localização

- UCP;
- Interna;
- Externa;

Característica: Capacidade

- Tamanho de palavra;

- Número de palavras;

Característica: Unidade de transferência

- Interna: Normalmente controlada pela largura do barramento;

- Externa: Normalmente um bloco que é muito maior que uma palavra;

Característica: 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;

Característica: Métodos de acesso

- Aleatório:
  - ✓ Endereços individuais identificam localizações com exatidão;
  - ✓ Tempo de acesso é independente da localização ou acesso anterior;
  - ✓ Por exemplo, memória 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;
  - ✓ Por exemplo, cache.

Característica: Desempenho

- **Tempo de acesso**: Tempo entre apresentar o endereço e obter os dados válidos;
- **Tempo de ciclo de memória**: Tempo que pode ser exigido antes do próximo acesso;
- **Taxa de transferência:** Taxa em que os dados podem ser movidos;

Característica: Tipos físicos

- Semicondutor: RAM;
- Magnético: Disco e fita;
- Óptico: CD e DVD.

Característica: Detalhes físicos

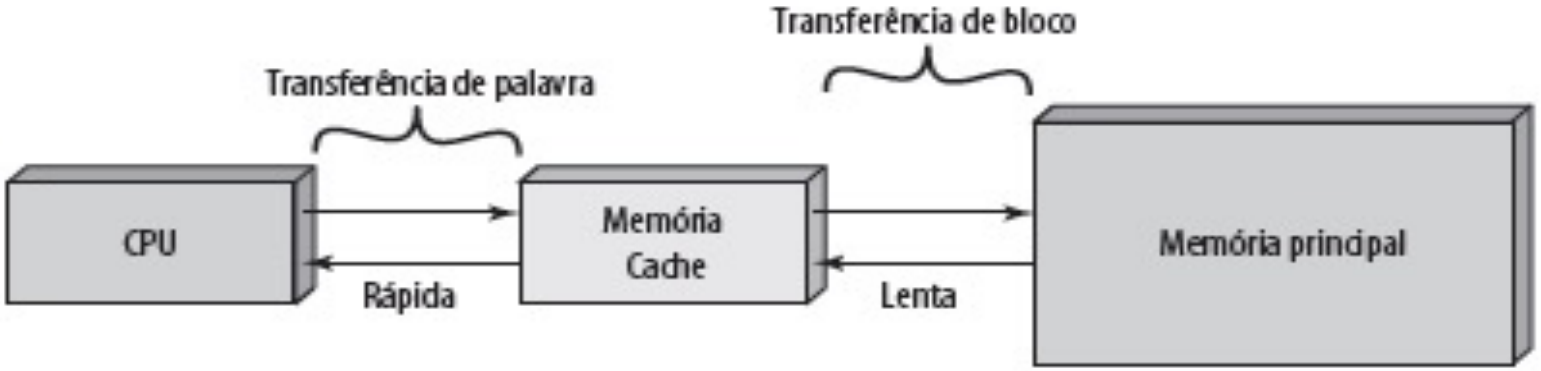
- Deterioração;
- Volatilidade;
- Apagável;
- Consumo de energia.

Memória Cache

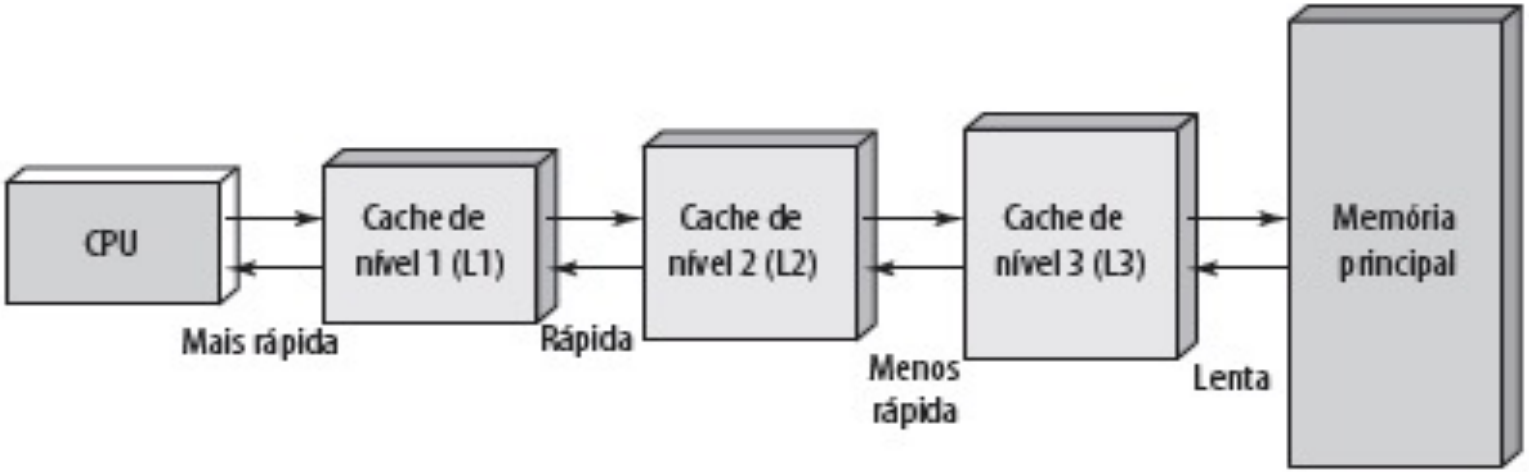
Cache

- Pequena quantidade de memória: **rápida**;
- Fica entre a memória principal e a UCP;
- Pode estar localizada no chip da UCP ou em módulo a parte.

Modelo da cache



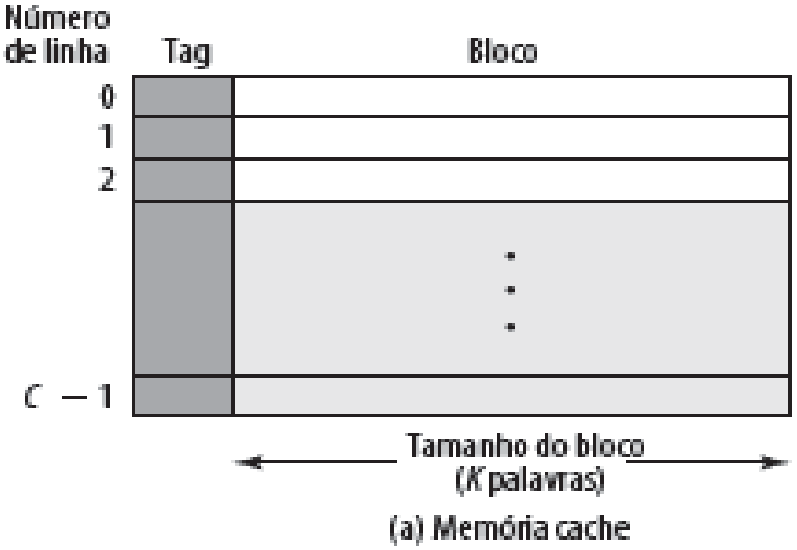
(a) Cache única

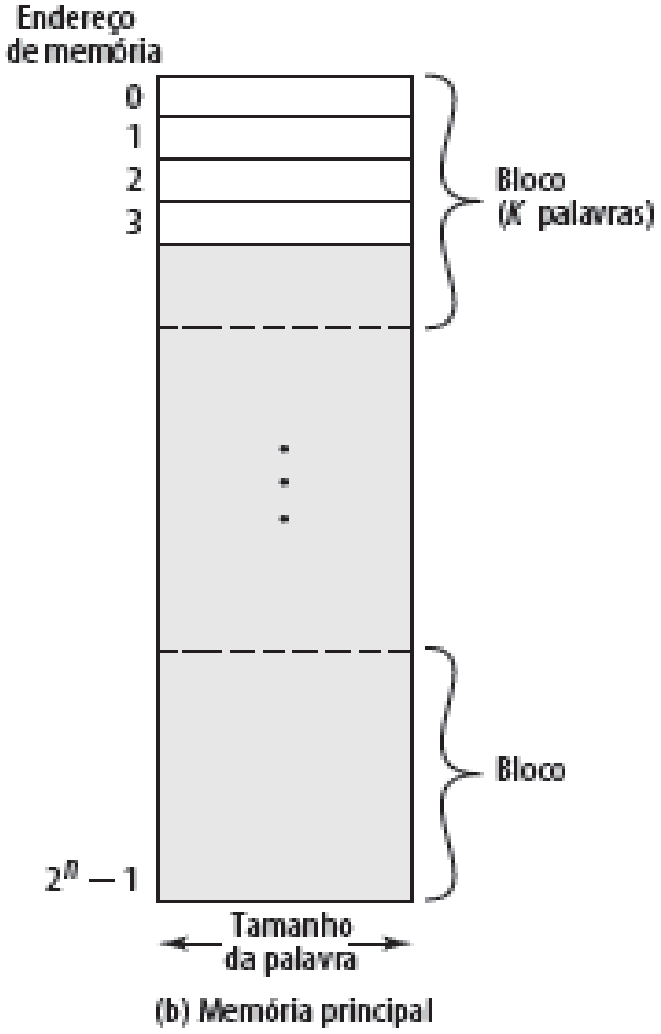


(b) Organização de cache em três níveis

Fonte: Arquitetura e Organização de Computadores – William Stallings

Estrutura de cache e memória principal





Fonte: Arquitetura e Organização de Computadores – William Stallings

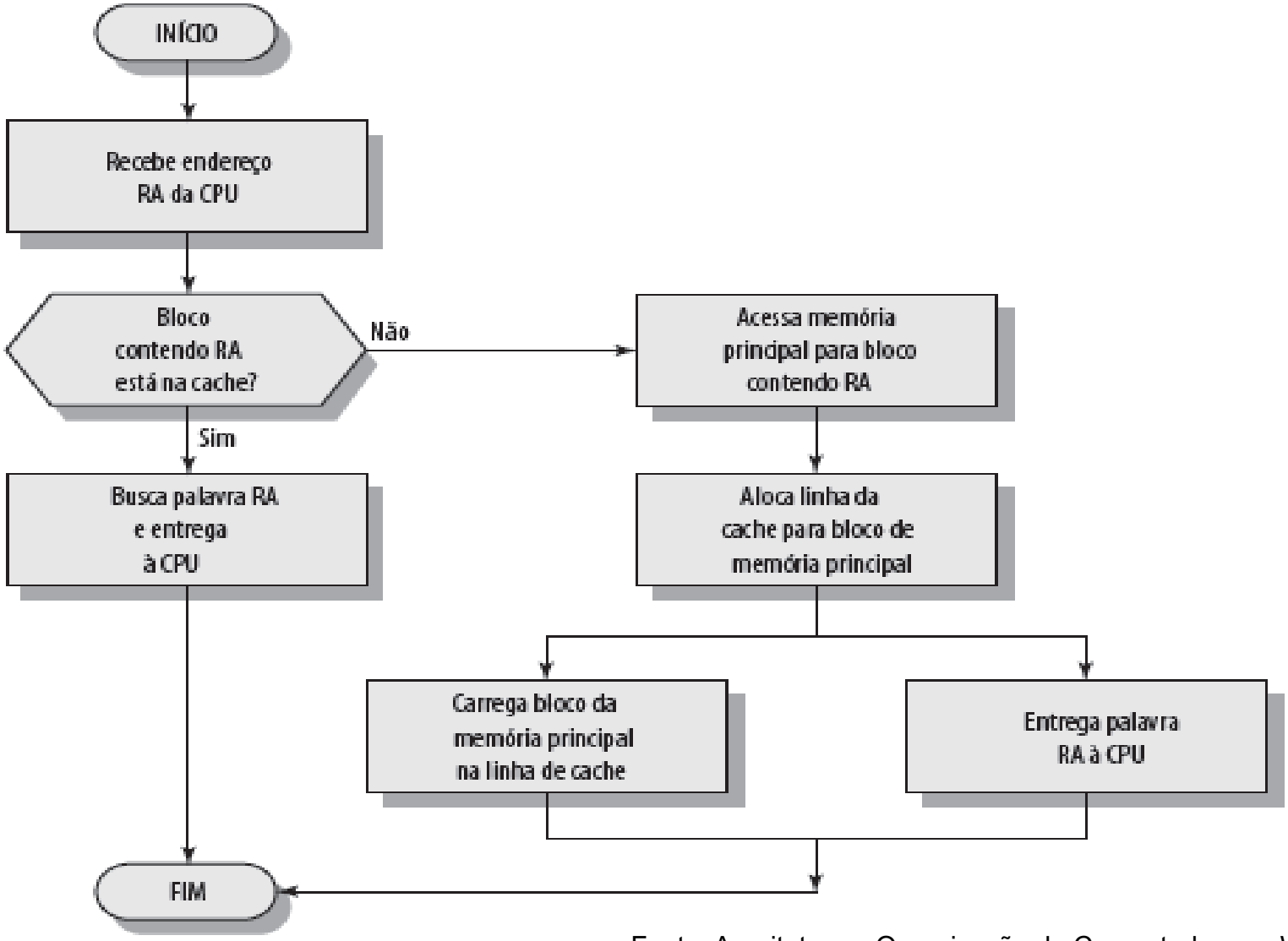
Cache – operação

- UCP requisita conteúdo do local de memória;
- Verifica se os dados estão em cache;
- Se estiverem, recupera-se da cache;

Cache – operação

- Se não, lê bloco solicitado da memória principal para a cache;
- Depois, entrega da cache à UCP;
- Cache inclui ***tags*** para identificar qual bloco da memória principal está em cada um de seus slots.

Cache – operação - fluxograma



Fonte: Arquitetura e Organização de Computadores – William Stallings

Projeto de cache

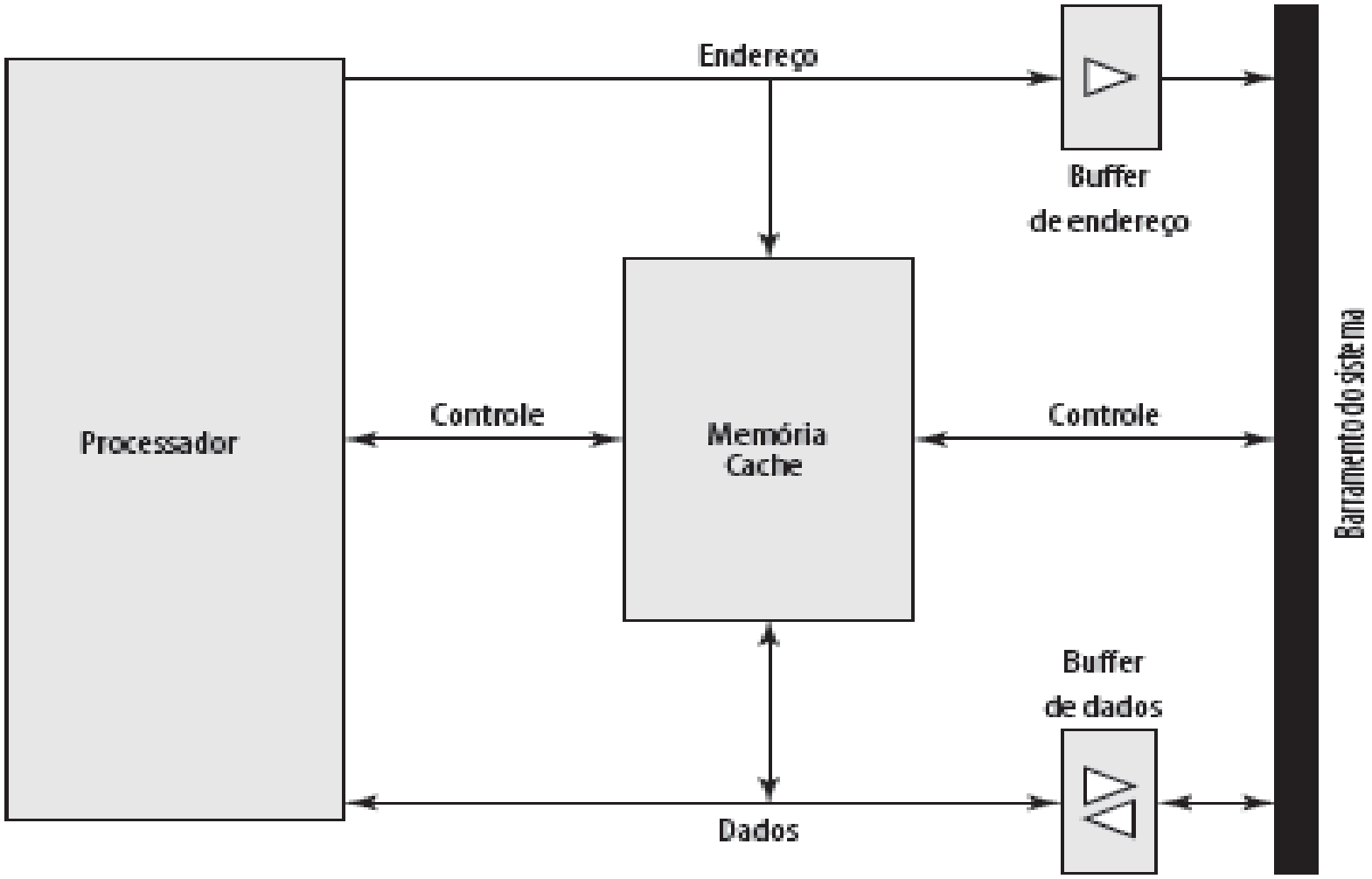
- O projeto de uma memória cache exige alguns cuidados especiais;
- Esses cuidados garantem o bom funcionamento da cache;
- Cache eficiente → Sistema Computacional eficiente.

Projeto de cache

O que se deve considerar:

- Organização;
- Tamanho;
- Função de mapeamento;
- Algoritmos de substituição;
- Política de escrita;
- Tamanho de bloco;
- Número de caches.

Projeto: Organização típica



Fonte: Arquitetura e Organização de Computadores – William Stallings

Projeto: Tamanho

O que considerar:

- **Custo**: Mais cache é caro;

- **Velocidade:**
  - ✓ Mais cache é mais rápido (geralmente);
  - ✓ Verificar dados na cache leva tempo.

Projeto: Tamanho

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

Fonte: Arquitetura e Organização de Computadores – William Stallings

Projeto: Função de mapeamento

- É forma como as informações são mapeadas da memória principal para a cache;

- Um bom mapeamento é fundamental para garantia do melhor desempenho;

- Cada função tem as suas vantagens e desvantagens.

Mapeamento direto

- Cada bloco de memória principal mapeado apenas para uma linha de cache;

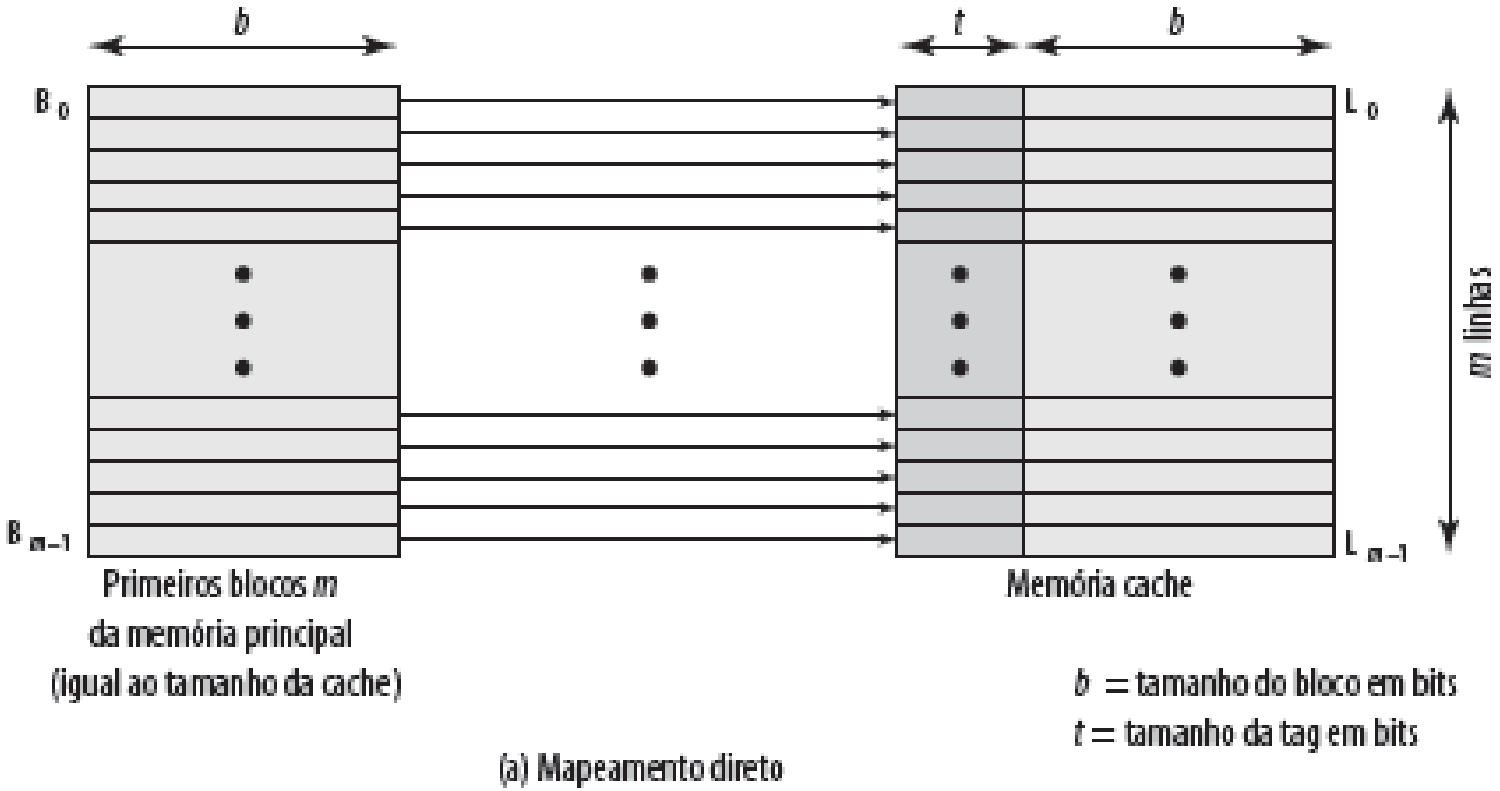
- Endereço está em duas partes:
  - ✓ Linha;
  - ✓ Tag

Mapeamento direto - Estrutura de endereços

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

Fonte: Arquitetura e Organização de Computadores – William Stallings

Mapeamento direto da cache para memória principal



(a) Mapeamento direto

Fonte: Arquitetura e Organização de Computadores – William Stallings

Mapeamento direto: Vantagens e Desvantagens

- 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**.

Mapeamento associativo

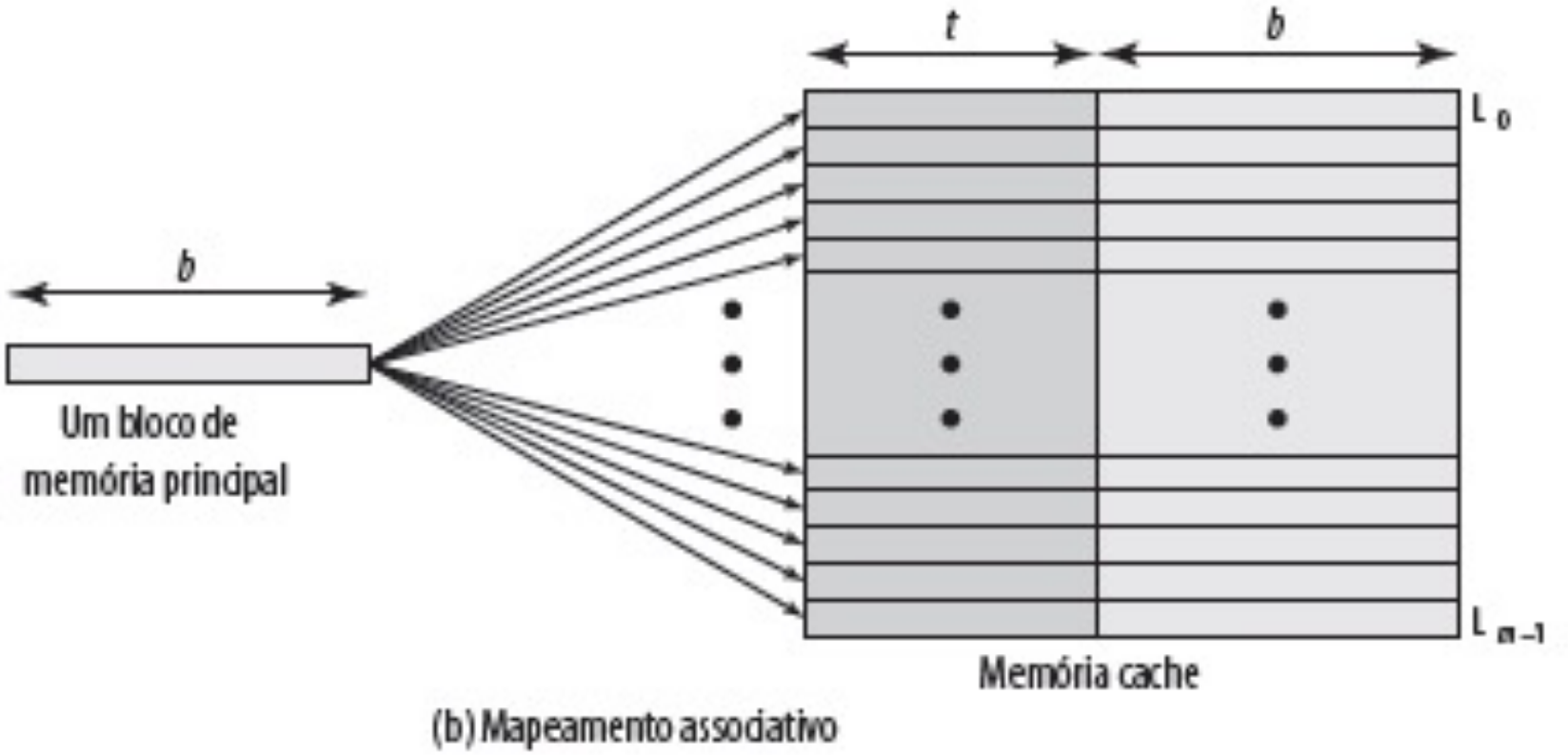
- Um bloco de memória principal pode ser carregado em qualquer linha de cache;

- Endereço de memória é interpretado como *tag* e palavra;

Mapeamento associativo

- 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



(b) Mapeamento associativo

Fonte: Arquitetura e Organização de Computadores – William Stallings

Mapeamento associativo - Estrutura de endereço

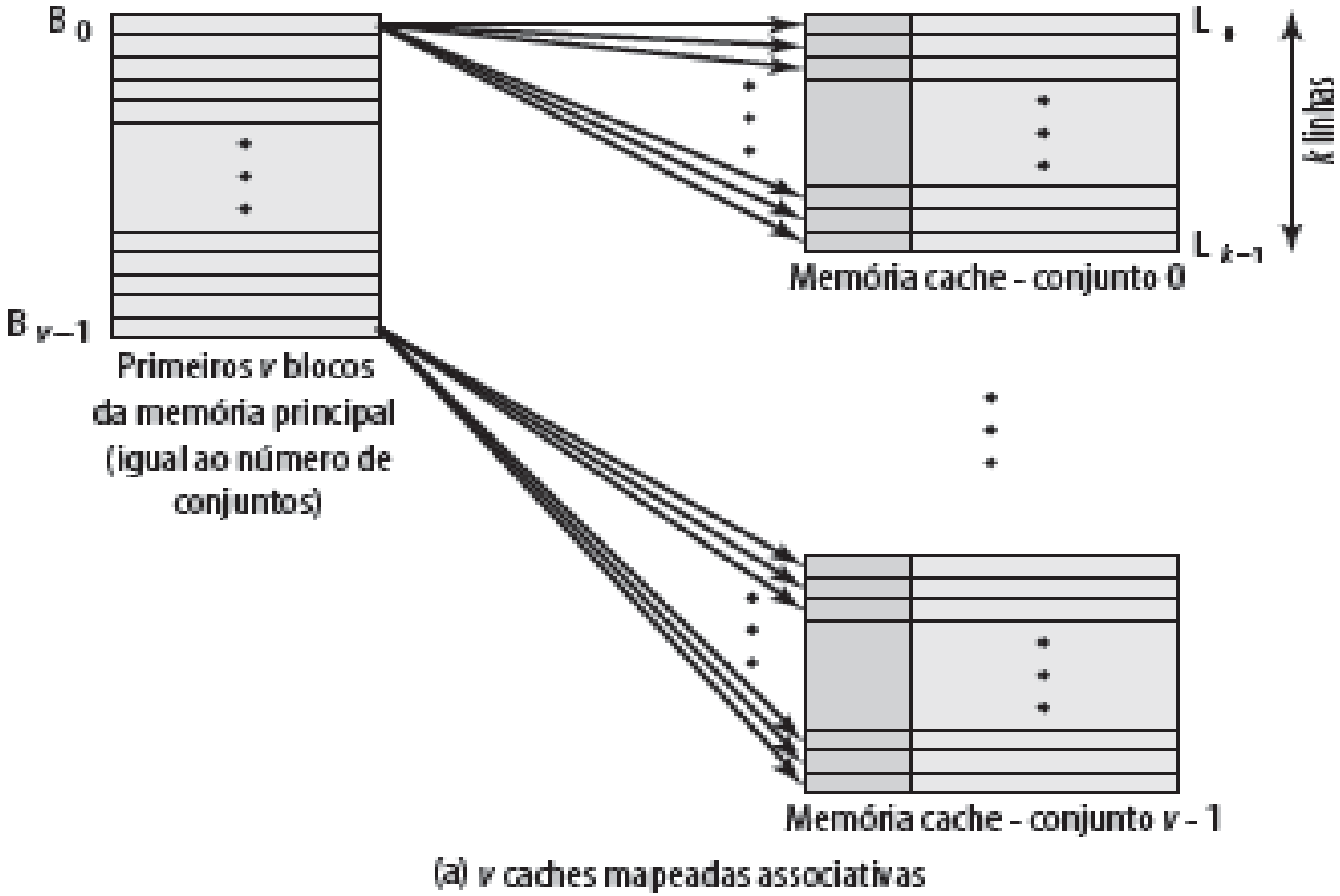
Tag 22 bit	Palavra 2 bit

Fonte: Arquitetura e Organização de Computadores – William Stallings

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;

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



(a) *v* caches mapeadas associativas

Fonte: Arquitetura e Organização de Computadores – William Stallings

Mapeamento associativo em conjunto - estrutura de endereços

Tag 9 bits	Conjunto 13 bit	Palavra 2 bit

Fonte: Arquitetura e Organização de Computadores – William Stallings

Projeto: Algoritmos de substituição

- A cache não possui um tamanho **infinito**;
- Cache cheia: alguma informação sai → novas informações entram;
- Esta substituição não pode ser aleatória, deve utilizar algumas regras;

Algoritmos de substituição

• Existem algoritmos implementados em *Hardware* para conseguir maior velocidade;

• Os mais utilizados são:

- ✓ ***LRU*** (*Least Recently Used): Menos Recentemente Usado;*
- ✓ ***MRU*** *(Most Recently Used): Mais Recentemente Usado;*
- ✓ ***FIFO*** *(First-In-First-Out): Fila;*
- ✓ ***LFU*** *(Least Frequently Used): Menos Frenquentemente Usado.*

Projeto: Política de escrita

- Preocupam-se com a coerência dos dados;
- Não deve sobrescrever bloco de cache a menos que a memória principal esteja atualizada;
- **Problema:** Múltiplas UCPs podem ter caches individuais.

Políticas de Escrita: *Write-through*

- Todas as escritas vão para a memória principal e também para a cache;
- Múltiplas UCPs podem monitorar o tráfego da memória principal para manter a cache local à UCP atualizada;
- Muito tráfego;
- Atrasa as escritas;

Política de escrita: *Write-back*

- Atualizações feitas inicialmente apenas na cache;

- Bit de atualização para slot de cache é definido quando ocorre a atualização;

Política de escrita: *Write-back*

- Se o bloco deve ser substituído, escreve na memória principal apenas se o bit atualizado estiver marcado;
- Outras caches saem de sincronismo;
- Reduz o tráfego.

Projeto: 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 (**localidade**).

Projeto: Tamanho de linha

- **Inversão posterior**: 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.

Projeto: Caches multinível

- Varia entre os tipos de arquiteturas;

- Antes era usada uma única *cache;*

- Hoje se faz uso de múltiplas *caches;*

- Muito se discute sobre *caches* com múltiplos níveis e se essas múltiplas *caches* serão agrupadas ou separadas.

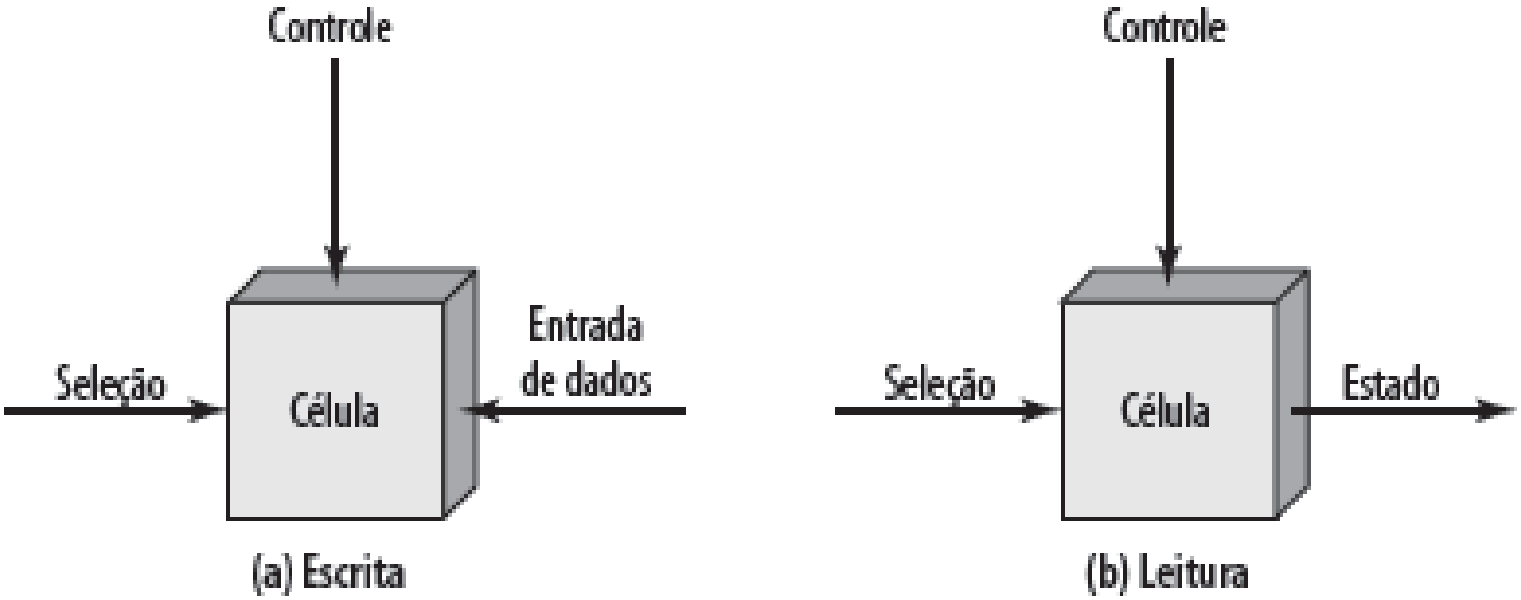
Memória Interna

Tipos de memória de semicondutor

<table>
<tr><th>Tipo de memória</th><th>Categoria</th><th>Apagamento</th><th>Mecanismo de escrita</th><th>Volatilidade</th></tr>
<tr><td>Memória de acesso aleatório (RAM)</td><td>Memória de leitura-escrita</td><td>Eletricamente, em nível de byte</td><td>Eletricamente</td><td>Volátil</td></tr>
<tr><td>Memória somente de leitura (ROM)</td><td rowspan="2">Memória somente de leitura</td><td rowspan="2">Não é possível</td><td>Máscaras</td><td rowspan="6">Não volátil</td></tr>
<tr><td>ROM programável (PROM, do inglês programmable ROM)</td><td rowspan="4">Eletricamente</td></tr>
<tr><td>PROM apagável (EPROM, do inglês erasable PROM)</td><td rowspan="3">Memória principalmente de leitura</td><td>Luz UV, nível de chip</td></tr>
<tr><td>PROM eletricamente apagável (EEPROM, do inglês electrically erasable PROM)</td><td>Eletricamente, nível de byte</td></tr>
<tr><td>Memória flash</td><td>Eletricamente, nível de bloco</td></tr>
</table>

Fonte: Arquitetura e Organização de Computadores – William Stallings

Célula de memória - Funcionamento

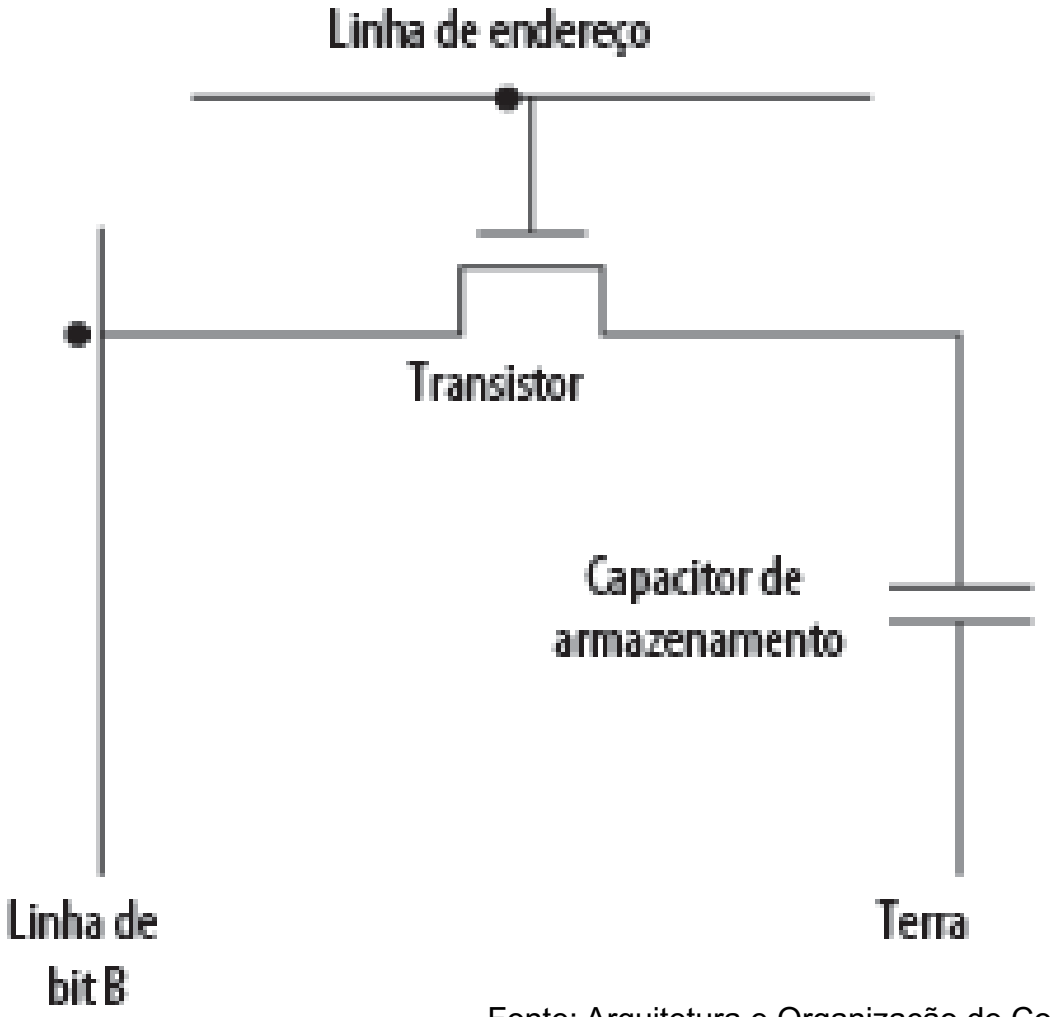


Fonte: Arquitetura e Organização de Computadores – William Stallings

RAM dinâmica

- Bits armazenados com carga em capacitores;
- Precisa de renovação mesmo se alimentada;
- Construção mais simples;
- Mais barata e mais lenta;
- Precisa de circuitos de *refresh;*
- Memória principal.

Estrutura da RAM dinâmica

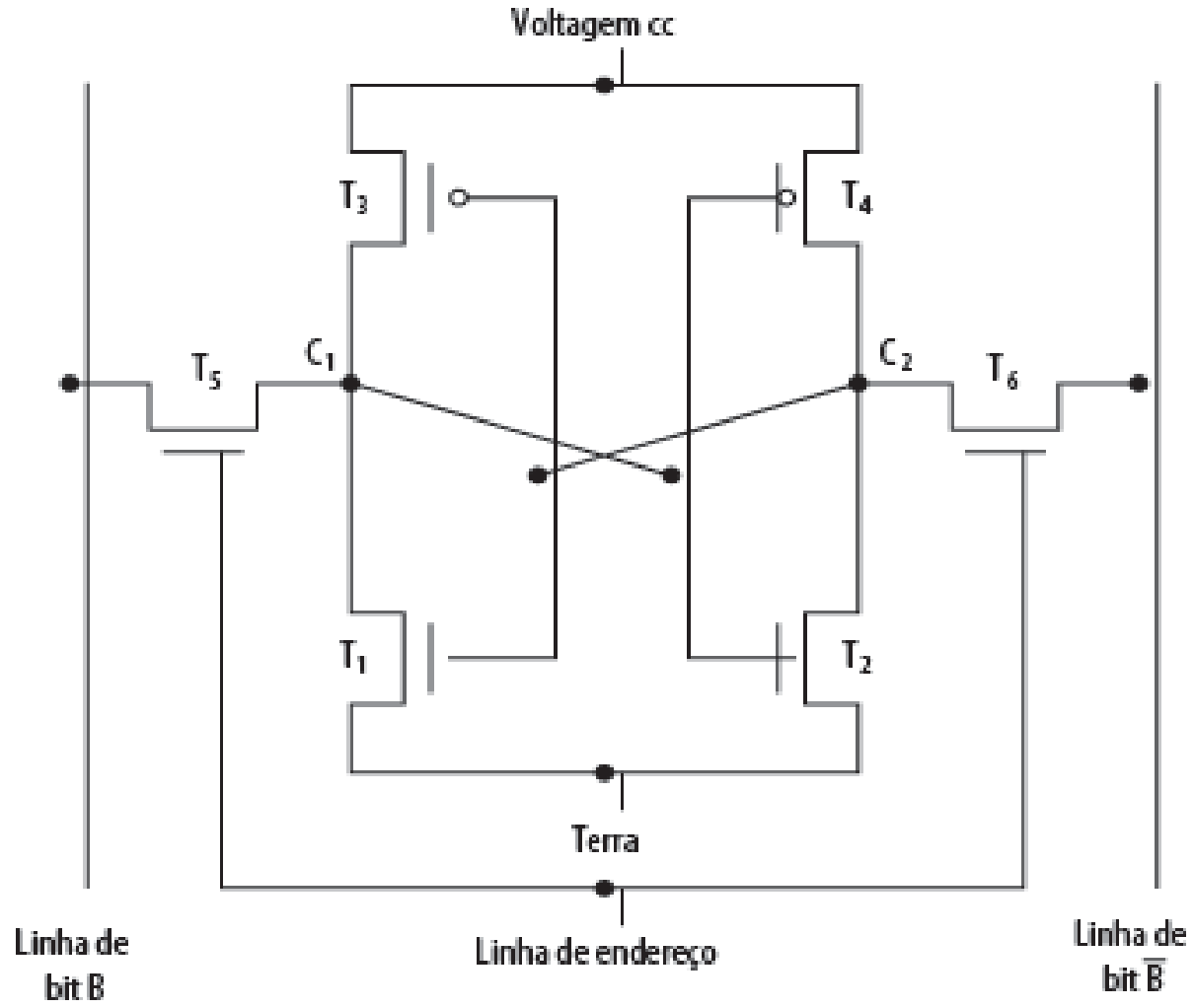


Fonte: Arquitetura e Organização de Computadores – William Stallings

RAM estática

- Bits armazenados como chaves ligado/desligado;
- Sem perder carga;
- Construção mais complexa;
- Mais cara e mais rápida;
- Não precisa de circuitos de *refresh;*
- Mais rápida;
- Cache.

Estrutura da RAM estática



Fonte: Arquitetura e Organização de Computadores – William Stallings

Read Only Memory (ROM)

- Armazenamento permanente;

- Aplicações:
  - ✓ Sub-rotinas de biblioteca;
  - ✓ Programas do sistema (BIOS);
  - ✓ Tabelas de função.

Tipos de ROM

- Gravada durante a fabricação: Muito cara para pequenas quantidades;

- Programável (uma vez):
  - ✓ PROM;
  - ✓ Precisa de equipamento especial para programar;

Tipos de ROM

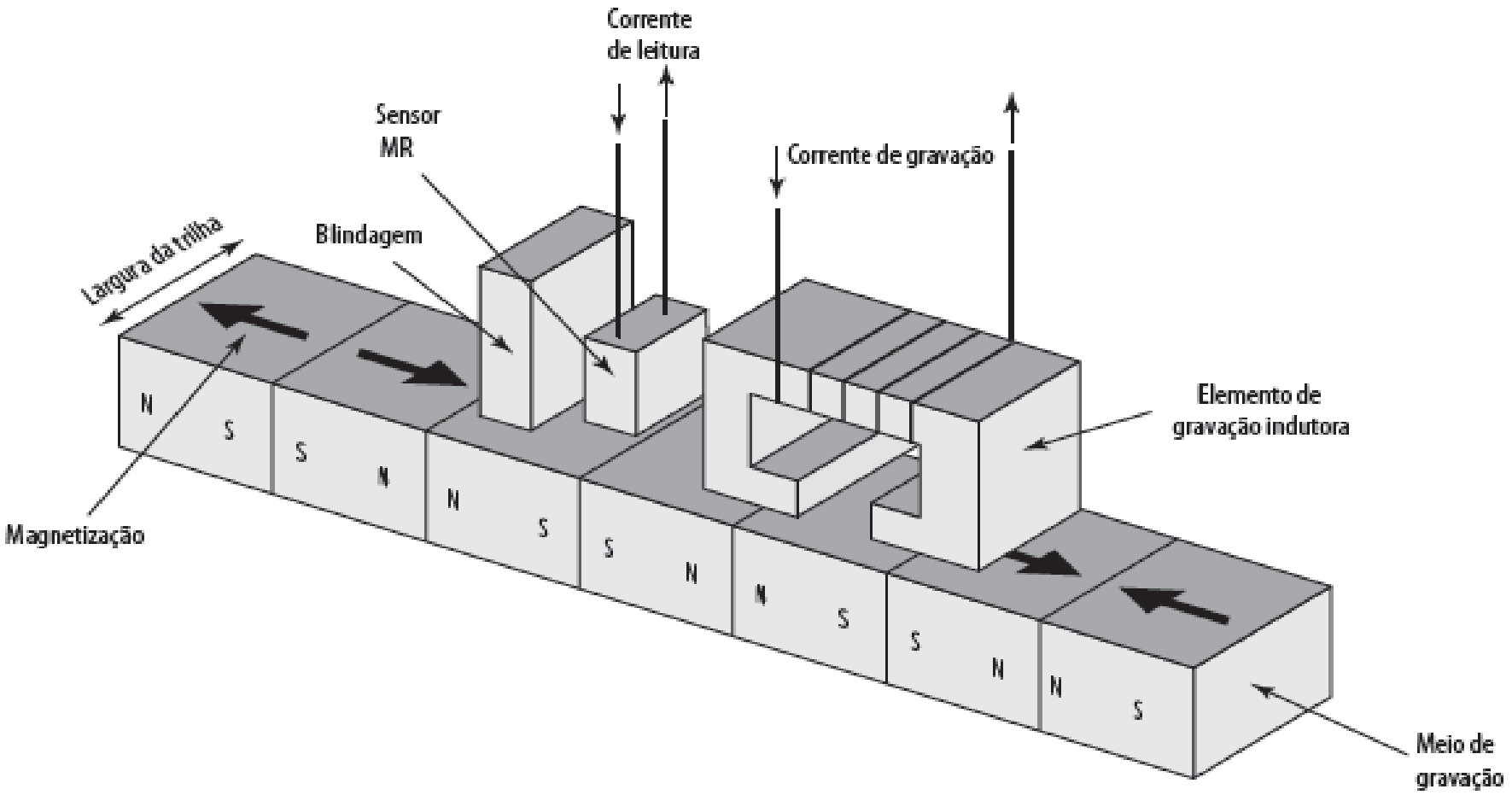
- Memória de Leitura (geralmente):
  - ✓ Erasable Programmable (EPROM): Apagada por UV;
  - ✓ Electrically Erasable (EEPROM): Leva muito mais tempo para escrever que para ler;
  - ✓ Memória flash: Apaga memória inteira eletricamente.

Disco magnético

- Substrato de disco coberto com material magnetizável;

- Substrato de vidro:
  - ✓ Maior uniformidade da superfície: Aumenta confiabilidade.
  - ✓ Redução nos defeitos da superfície: Erros reduzidos de leitura/gravação.
  - ✓ Melhor rigidez.
  - ✓ Maior resistência a choques e dados.

Cabeça de gravação



Fonte: Arquitetura e Organização de Computadores – William Stallings

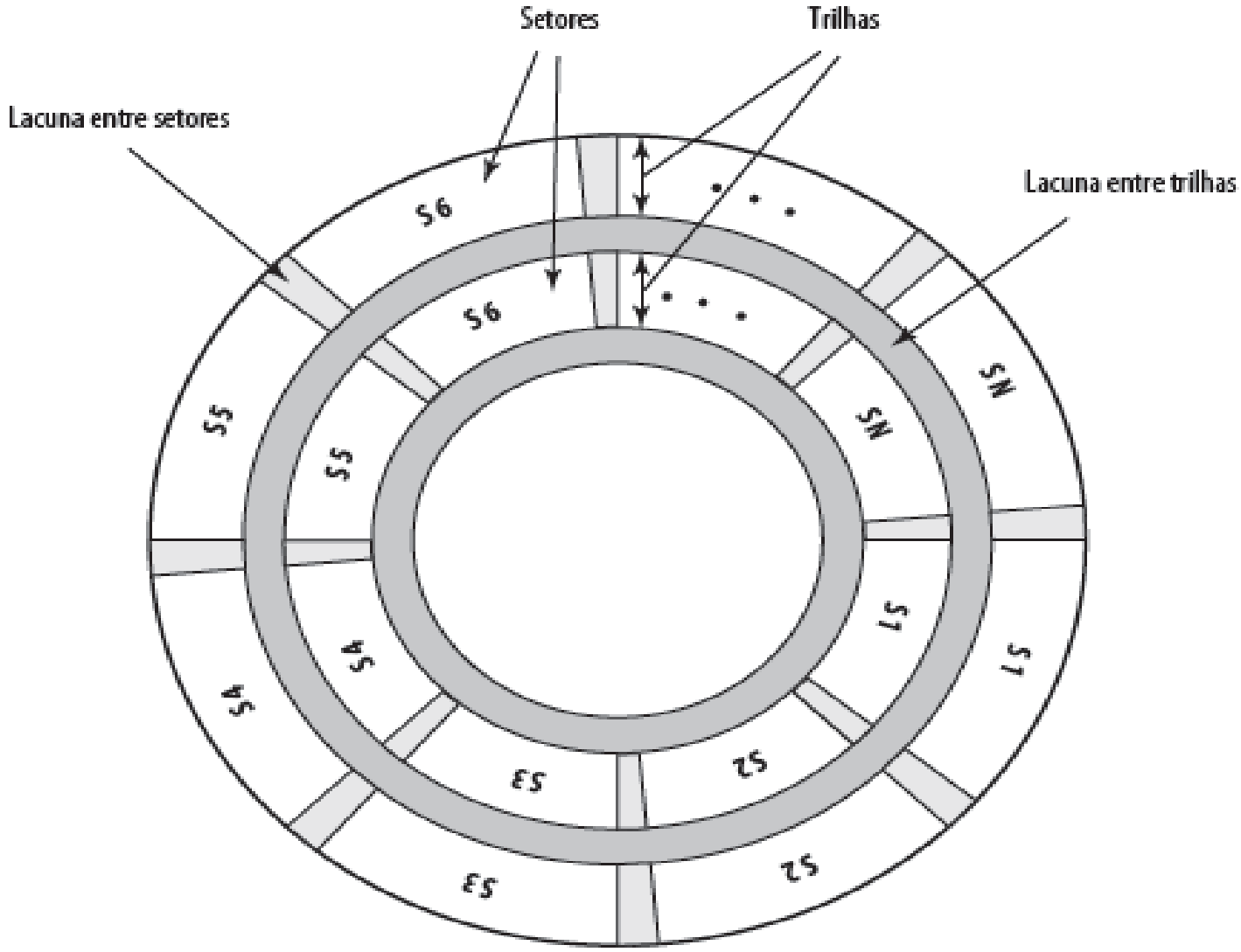
Organização e formatação de dados

- Trilhas concêntricas:
  - ✓ Lacunas entre as trilhas;
  - ✓ Reduza a lacuna para aumentar a capacidade;
  - ✓ Mesmo número de bits por trilha (densidade de compactação variável);
  - ✓ Velocidade angular constante.

Organização e formatação de dados

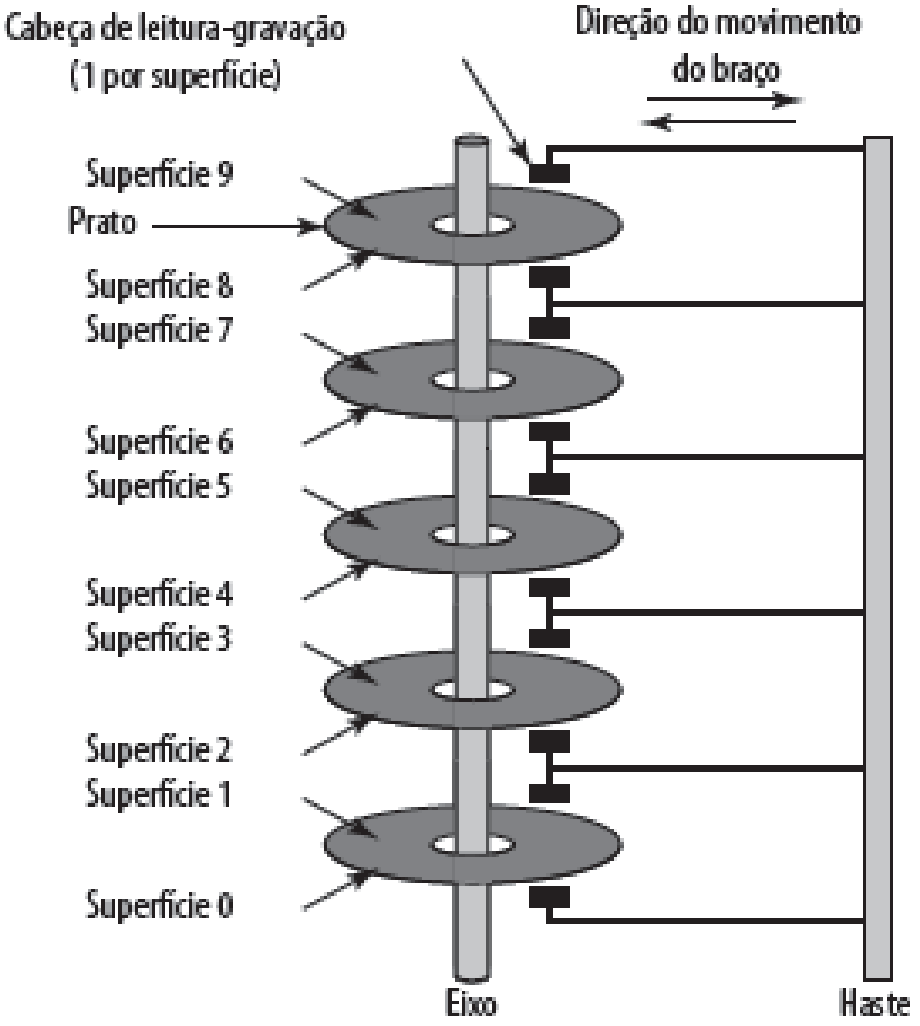
- Trilhas divididas em setores;
- Tamanho de **bloco mínimo** é de um setor;
- Pode haver mais de um setor por bloco;

Organização e formatação de dados



Fonte: Arquitetura e Organização de Computadores – William Stallings

Discos com múltiplos pratos e cabeças



Fonte: Arquitetura e Organização de Computadores – William Stallings

Velocidade do disco

- **Tempo de busca:** Movendo cabeça para trilha correta;
- **Latência (rotacional):** Esperando dados passarem sob a cabeça;
- ***Tempo de acesso= Busca + Latência;***
- Influencia na Taxa de Transferência.

RAID (Redundant Array of Independent Disks)

- Conjunto de discos para melhorar o sistema de armazenamento;
- 7 níveis (0 à 6);
- Conjunto dos principais discos vistos como uma única unidade lógica pelo S/O;

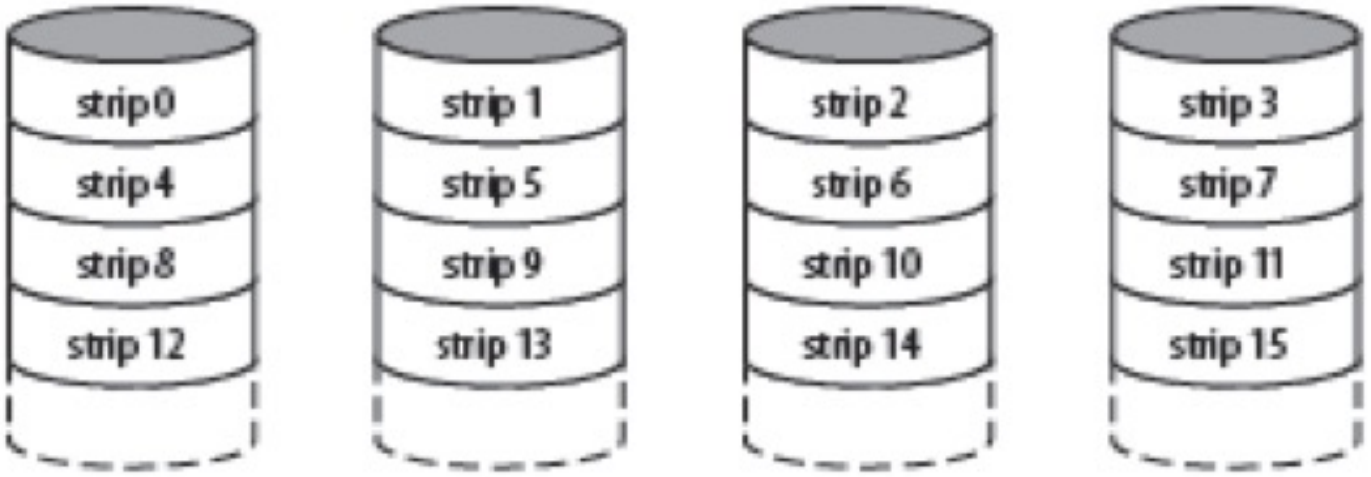
RAID (Redundant Array of Independent Disks)

- Dados distribuídos pelas unidades físicas;
- Pode usar capacidade redundante;
- Pode usar capacidade redundante para armazenar informação de paridade.

RAID 0

- Não redundante;
- Dados espalhados por todos os discos;
- Mapeamento Round Robin;
- Maior velocidade:
  - ✓ Múltiplas solicitações de dados provavelmente não no mesmo disco.
  - ✓ Discos buscam em paralelo.
  - ✓ Um conjunto de dados provavelmente será espalhado por múltiplos discos.

RAID 0



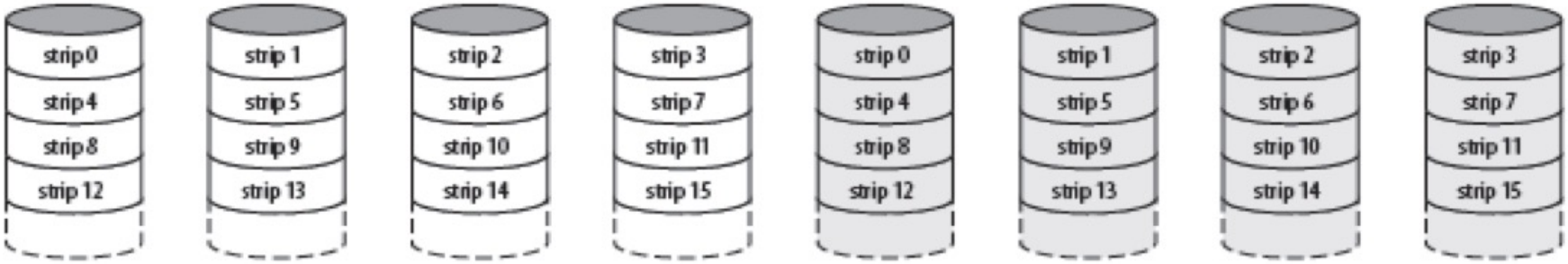
(a) RAID 0 (não redundante)

Fonte: Arquitetura e Organização de Computadores – William Stallings

RAID 1

- Discos espelhados;
- Dados espalhados pelos discos;
- 2 cópias de cada ***stripe*** em discos separados;
- **Leitura** e **Gravação** em ambos.
- Mecanismo caro de ser implementado.

RAID 1



(b) RAID 1 (espelhado)

Fonte: Arquitetura e Organização de Computadores – William Stallings

RAID 2

- Discos são sincronizados;
- Correção de erro calculada pelos bits correspondentes nos discos;
- Múltiplos discos de paridade armazenam correção de erro via **código de Hamming** em posições correspondentes.
- Muita redundância: caro e pouco utilizado.

RAID 2

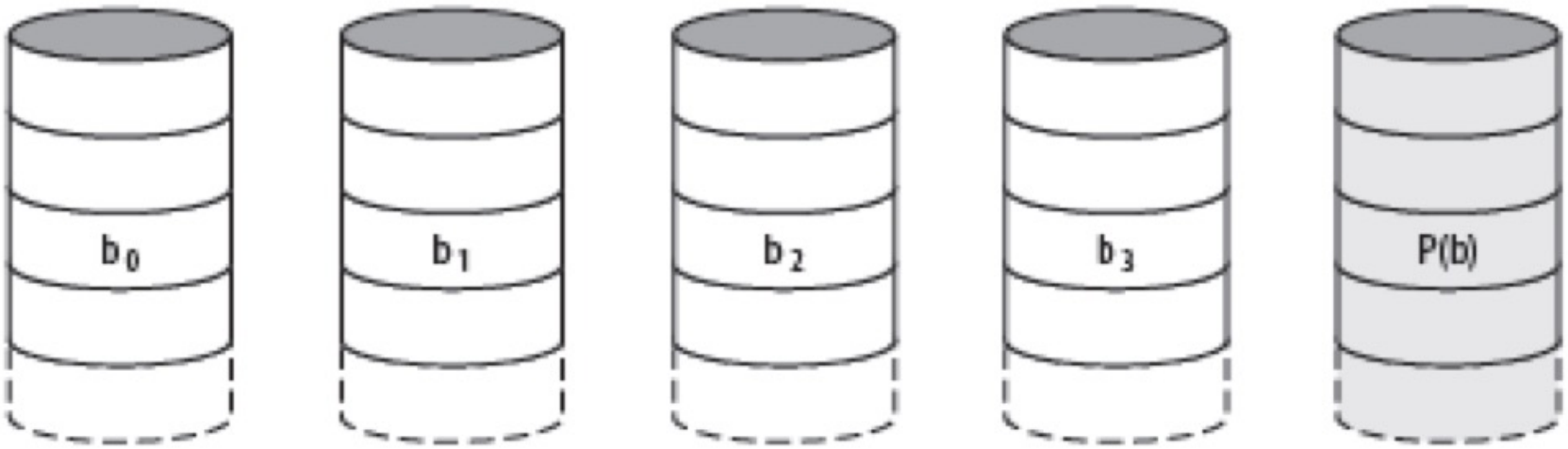
(c) RAID 2 (redundância por código de Hamming)

Fonte: Arquitetura e Organização de Computadores – William Stallings

RAID 3

- **Somente um** disco redundante, não importa o tamanho do *array*;
- Bit de **paridade simples** para cada conjunto de bits correspondentes;
- Dados sobre unidade com defeito podem ser reconstruídos a partir de dados sobreviventes e informação de paridade;
- Taxas de transferência muito altas.

RAID 3



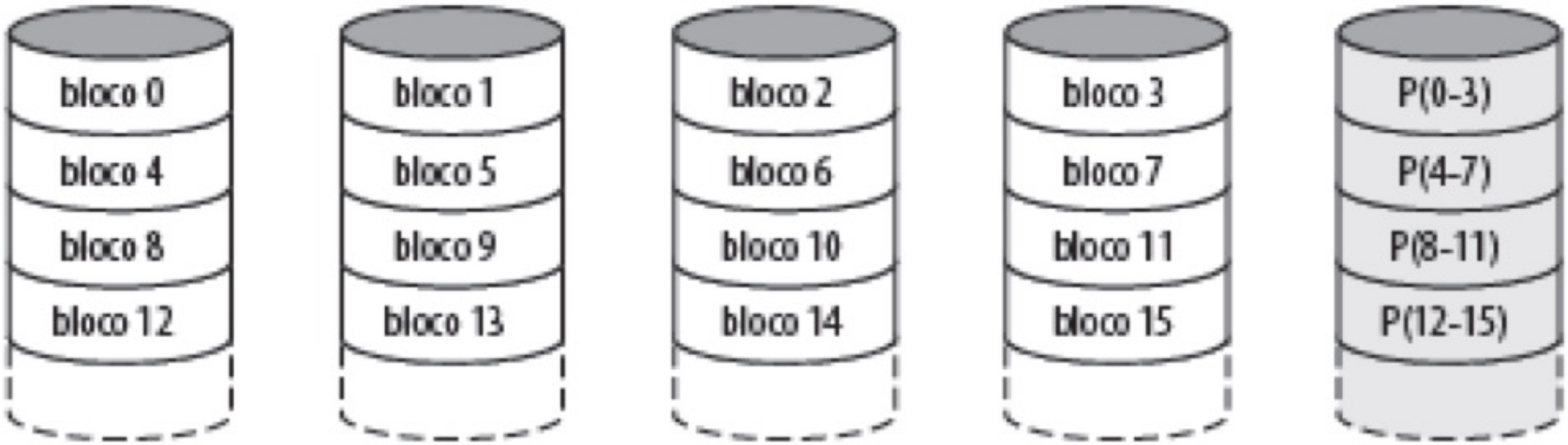
(d) RAID 3 (paridade de bit intercalada)

Fonte: Arquitetura e Organização de Computadores – William Stallings

RAID 4

- Cada disco opera independentemente;
- Bom para taxa de solicitação de E/S alta;
- Paridade bit a bit calculada por stripes em cada disco;
- Paridade armazenada no **disco de paridade**.

RAID 4



(e) RAID 4 (paridade em nível de bloco)

Fonte: Arquitetura e Organização de Computadores – William Stallings

RAID 5

- Paridade espalhada por todos os discos;
- Alocação round-robin para stripe de paridade;
- Evita gargalo do RAID 4 no disco de paridade.

RAID 5

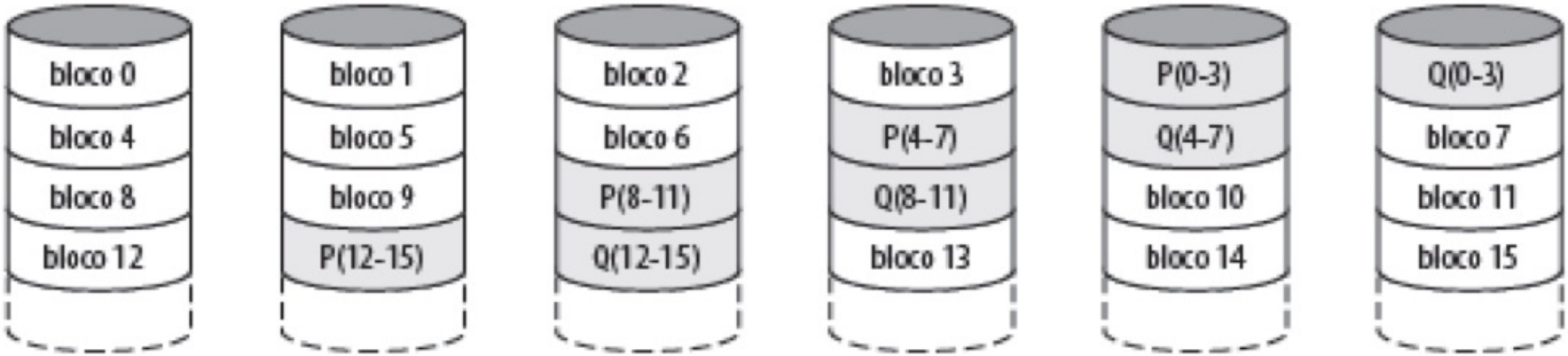
(f) RAID 5 (paridade em nível de bloco distribuída)

Fonte: Arquitetura e Organização de Computadores – William Stallings

RAID 6

- Dois cálculos de paridade;
- Armazenado em blocos separados em discos diferentes;
- Alta disponibilidade de dados: **Três discos** precisam falhar para haver perda de dados;

RAID 6



(g) RAID 6 (redundância dual)

Fonte: Arquitetura e Organização de Computadores – William Stallings