



UNIVERSIDADE
FEDERAL DE
SERGIPE



DEPARTAMENTO
DE COMPUTAÇÃO

Hierarquia de memória

Arquitetura de Computadores

Bruno Prado

Departamento de Computação / UFS

Introdução

- ▶ Categorização dos dispositivos de memória
 - ▶ Localização
 - ▶ Interna
 - ▶ Externa

Introdução

- ▶ Categorização dos dispositivos de memória
 - ▶ Localização
 - ▶ Interna
 - ▶ Externa
 - ▶ Estrutura de armazenamento e endereçamento
 - ▶ Tamanho de palavra
 - ▶ Bloco de dados

Introdução

- ▶ Categorização dos dispositivos de memória
 - ▶ Localização
 - ▶ Interna
 - ▶ Externa
 - ▶ Estrutura de armazenamento e endereçamento
 - ▶ Tamanho de palavra
 - ▶ Bloco de dados
 - ▶ Método de acesso
 - ▶ Sequencial
 - ▶ Direto
 - ▶ Aleatório
 - ▶ Associativo

Introdução

- ▶ Categorização dos dispositivos de memória
 - ▶ Localização
 - ▶ Interna
 - ▶ Externa
 - ▶ Estrutura de armazenamento e endereçamento
 - ▶ Tamanho de palavra
 - ▶ Bloco de dados
 - ▶ Método de acesso
 - ▶ Sequencial
 - ▶ Direto
 - ▶ Aleatório
 - ▶ Associativo
 - ▶ Capacidade e desempenho
 - ▶ Latência
 - ▶ Taxa de transferência

Introdução

- ▶ Localização da memória
 - ▶ Interna
 - ▶ São os dispositivos de memória que são diretamente acessados ou controlados pelo processador
 - ▶ Ex: registradores, cache, memória principal

Introdução

- ▶ Localização da memória
 - ▶ Interna
 - ▶ São os dispositivos de memória que são diretamente acessados ou controlados pelo processador
 - ▶ Ex: registradores, cache, memória principal
 - ▶ Externa
 - ▶ O acesso é feito através de controladores de E/S para as unidades de armazenamento
 - ▶ Ex: discos SATA, mídias óticas ou magnéticas

Introdução

- ▶ Estrutura de armazenamento da memória
 - ▶ Tamanho de palavra
 - ▶ Define quantos bytes são processados por vez, geralmente refletindo a capacidade da arquitetura
 - ▶ Ex: registradores múltiplos de 1 byte com 8, 16 ou 32 bits

Introdução

- ▶ Estrutura de armazenamento da memória
 - ▶ Tamanho de palavra
 - ▶ Define quantos bytes são processados por vez, geralmente refletindo a capacidade da arquitetura
 - ▶ Ex: registradores múltiplos de 1 byte com 8, 16 ou 32 bits
 - ▶ Bloco de dados
 - ▶ São conjuntos de dados com tamanhos muito maiores que a palavra do sistema, sendo utilizados em dispositivos de memória externa
 - ▶ Ex: blocos múltiplos de 512 bytes com até 65.536 bytes

Introdução

- ▶ Método de acesso da memória
 - ▶ Sequencial
 - ▶ Os dados estão em sequência na memória
 - ▶ Para obter o último elemento da sequência, todos os dados precisam ser acessados durante a busca
 - ▶ Ex: fita magnética

Introdução

- ▶ Método de acesso da memória
 - ▶ Sequencial
 - ▶ Os dados estão em sequência na memória
 - ▶ Para obter o último elemento da sequência, todos os dados precisam ser acessados durante a busca
 - ▶ Ex: fita magnética
 - ▶ Direto
 - ▶ Cada bloco de dados possui um endereço físico único
 - ▶ Após este endereçamento, é feita a busca pelo bloco
 - ▶ Ex: unidade de disco

Introdução

- ▶ Método de acesso da memória
 - ▶ Aleatório
 - ▶ Os dados são armazenados em endereços ou posições únicas da unidade de memória
 - ▶ O tempo de acesso é constante e é independente da última posição de memória acessada
 - ▶ Ex: memória principal

Introdução

- ▶ Método de acesso da memória
 - ▶ Aleatório
 - ▶ Os dados são armazenados em endereços ou posições únicas da unidade de memória
 - ▶ O tempo de acesso é constante e é independente da última posição de memória acessada
 - ▶ Ex: memória principal
 - ▶ Associativo
 - ▶ É um tipo de acesso aleatório por indexação
 - ▶ O endereçamento é feito por associação do dado ao seu endereçamento na memória, eliminando ou reduzindo a necessidade de busca dos dados
 - ▶ Ex: memória cache

Introdução

- ▶ Capacidade e desempenho da memória
 - ▶ Latência
 - ▶ É o tempo gasto para realizar uma operação de E/S
 - ▶ Este tempo é constante para o acesso aleatório

$$Latência = t_{busca} + t_{operação}$$

Introdução

► Capacidade e desempenho da memória

► Latência

- É o tempo gasto para realizar uma operação de E/S
- Este tempo é constante para o acesso aleatório

$$Latência = t_{busca} + t_{operação}$$

► Taxa de transferência

- É definida como a taxa com que os dados podem ser escritos ou lidos de uma unidade de memória
- Em dispositivos de acesso randômico, a taxa depende somente de sua frequência de relógio
- No acesso não aleatório, a taxa de transferência é influenciado pelo tempo de acesso de operação

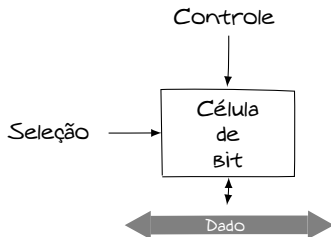
$$Taxa\ de\ transferência = \frac{\#bits}{Latência}$$

Memória interna

- ▶ *Random-Access Memory* (RAM)
 - ▶ Funcionamento eletrônico
 - ▶ Armazenamento volátil
 - ▶ Acesso aleatório

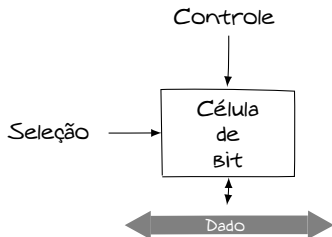
Memória interna

- ▶ A memória interna utiliza um sistema binário para armazenar os dados em unidades básicas (células)
- ▶ São permitidas operações de escrita e leitura
- ▶ É necessário controlar e selecionar as células para realizar as operações de acesso aos bits



Memória interna

- ▶ A memória interna utiliza um sistema binário para armazenar os dados em unidades básicas (células)
 - ▶ São permitidas operações de escrita e leitura
 - ▶ É necessário controlar e selecionar as células para realizar as operações de acesso aos bits



Cada bit é armazenado individualmente
(Em 4 GB existem 34.359.738.368 células)

Memória interna

- ▶ Principais tipos de memória interna
 - ▶ Registrador
 - ▶ É definido pela arquitetura e opera na mesma frequência de relógio do processador
 - ▶ Por possuir o maior custo e velocidade, apresenta a menor capacidade de armazenamento

Memória interna

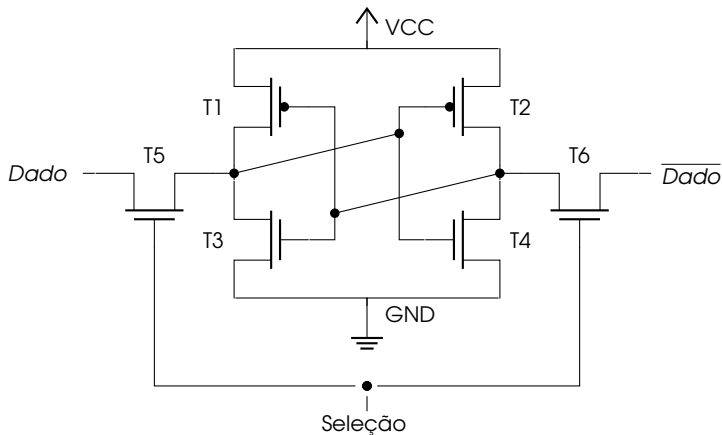
- ▶ Principais tipos de memória interna
 - ▶ Registrador
 - ▶ É definido pela arquitetura e opera na mesma frequência de relógio do processador
 - ▶ Por possuir o maior custo e velocidade, apresenta a menor capacidade de armazenamento
 - ▶ Memória cache
 - ▶ Está embarcada no processador e armazena os últimos valores acessados da memória
 - ▶ *Static RAM* (SRAM)

Memória interna

- ▶ Principais tipos de memória interna
 - ▶ Registrador
 - ▶ É definido pela arquitetura e opera na mesma frequência de relógio do processador
 - ▶ Por possuir o maior custo e velocidade, apresenta a menor capacidade de armazenamento
 - ▶ Memória cache
 - ▶ Está embarcada no processador e armazena os últimos valores acessados da memória
 - ▶ *Static RAM* (SRAM)
 - ▶ Memória principal
 - ▶ Os código e dados das aplicações ficam armazenados nesta memória
 - ▶ *Dynamic RAM* (DRAM)

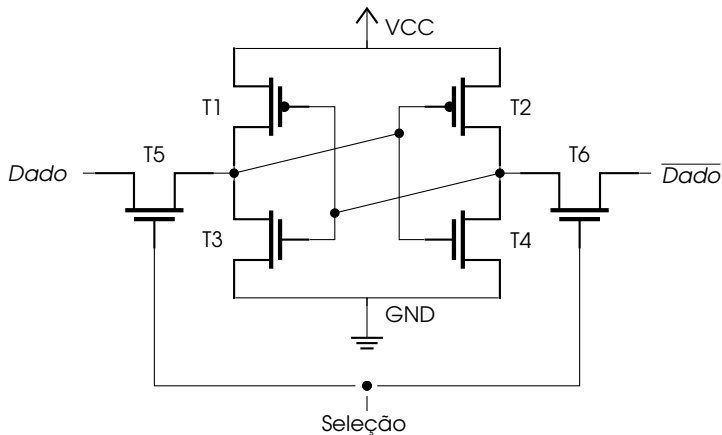
Memória interna

- ▶ Memória cache
 - ▶ Utiliza os mesmos elementos lógicos do processador (transistores) para armazenar os valores binários 0 e 1



Memória interna

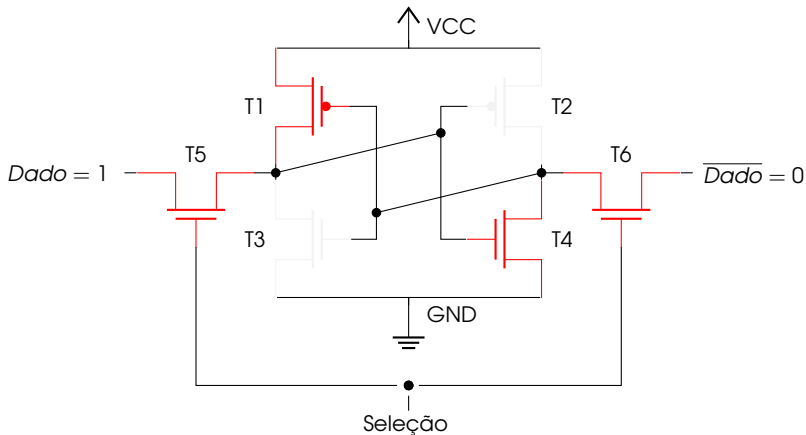
- ▶ Memória cache
 - ▶ Utiliza os mesmos elementos lógicos do processador (transistores) para armazenar os valores binários 0 e 1



Cada célula utiliza 6 transistores

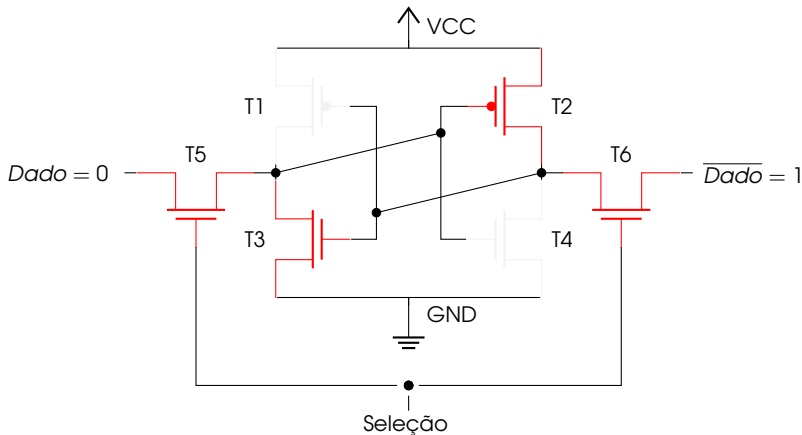
Memória interna

- ▶ Memória cache
 - ▶ Utiliza os mesmos elementos lógicos do processador (transistores) para armazenar os valores binários 0 e 1



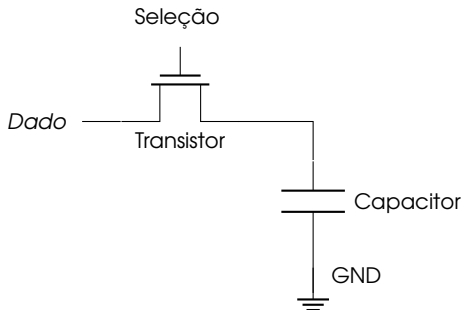
Memória interna

- ▶ Memória cache
 - ▶ Utiliza os mesmos elementos lógicos do processador (transistores) para armazenar os valores binários 0 e 1



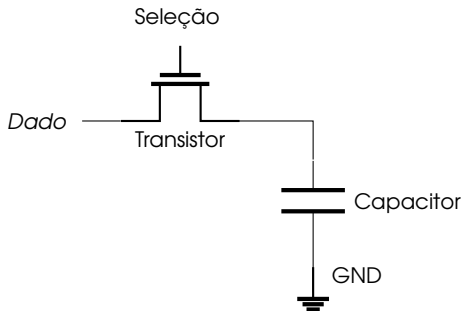
Memória interna

- ▶ Memória principal
 - ▶ A tecnologia dinâmica de armazenamento utiliza capacitores para armazenar o valor binário 0 (descarregado) ou 1 (carregado) da célula
 - ▶ Como existe uma tendência natural de perder a carga, o capacitor precisa ser periodicamente recarregado para manter seu estado



Memória interna

- ▶ Memória principal
 - ▶ A tecnologia dinâmica de armazenamento utiliza capacitores para armazenar o valor binário 0 (descarregado) ou 1 (carregado) da célula
 - ▶ Como existe uma tendência natural de perder a carga, o capacitor precisa ser periodicamente recarregado para manter seu estado



Cada célula utiliza 1 transistor e 1 capacitor

Memória interna

► Comparativo SRAM x DRAM

Característica	SRAM	DRAM
Área e custo de cada célula	↑	↓
Desempenho das operações	↑	↓
Consumo de potência	↓	↑
Densidade de armazenamento	↓	↑

Memória interna

- ▶ O que acontece caso alguma célula de memória tenha seu valor invertido ou esteja defeituosa?
 - ▶ Corrupção dos dados
 - ▶ Falha do sistema

Memória interna

- ▶ O que acontece caso alguma célula de memória tenha seu valor invertido ou esteja defeituosa?
 - ▶ Corrupção dos dados
 - ▶ Falha do sistema



Memória interna

- ▶ O que acontece caso alguma célula de memória tenha seu valor invertido ou esteja defeituosa?
 - ▶ Corrupção dos dados
 - ▶ Falha do sistema

Saldo Bancário



Memória interna

- ▶ O que pode causar erros na memória?

Memória interna

- ▶ O que pode causar erros na memória?
 - ▶ Defeito de fabricação/falha do dispositivo

Memória interna

- ▶ O que pode causar erros na memória?
 - ▶ Defeito de fabricação/falha do dispositivo
 - ▶ Interferência elétrica/magnética/térmica

Memória interna

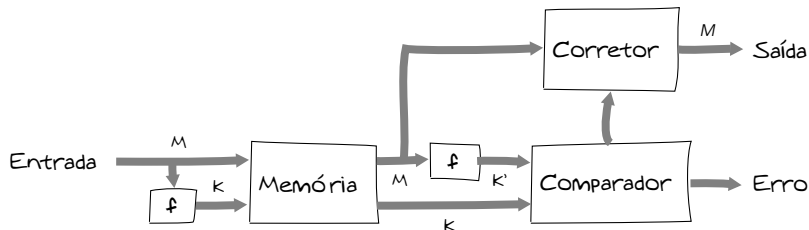
- ▶ O que pode causar erros na memória?
 - ▶ Defeito de fabricação/falha do dispositivo
 - ▶ Interferência elétrica/magnética/térmica
 - ▶ Radiação artificial (raio-x) e natural (cósmica)

Memória interna

- ▶ O que pode causar erros na memória?
 - ▶ Defeito de fabricação/falha do dispositivo
 - ▶ Interferência elétrica/magnética/térmica
 - ▶ Radiação artificial (raio-x) e natural (cósmica)
- ▶ Correção e detecção de erros na memória
 - ▶ Checagem em software (*checksum*)
 - ▶ Memória com correção de erro (ECC)

Memória interna

- ▶ Memória com correção de erro
 - ▶ Armazenamento extra para código de checagem K que é gerado a partir do dado de tamanho M
 - ▶ Componentes adicionais de comparação e correção



Memória interna

- ▶ Código de Hamming
 - ▶ Permite a correção de erros simples em bits em palavras de dados de tamanho M
 - ▶ É necessária uma palavra de síndrome K com valor entre 0 para indicar ausência de erros até $2^K - 1$ para indicar que bit apresenta erro

$$2^K - 1 \geq M + K$$

Memória interna

- ▶ Código de Hamming
 - ▶ Permite a correção de erros simples em bits em palavras de dados de tamanho M
 - ▶ É necessária uma palavra de síndrome K com valor entre 0 para indicar ausência de erros até $2^K - 1$ para indicar que bit apresenta erro

$$2^K - 1 \geq M + K$$

A correção de erro pode ser aplicada tanto nos bits de dados (M) como na palavra de síndrome (K)

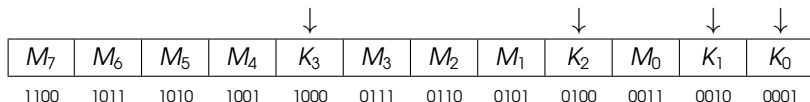
Memória interna

- ▶ Código de Hamming

- ▶ Considerando palavras de dados com M de 8 bits

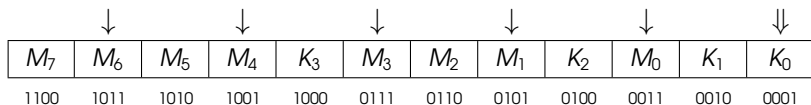
$$2^K - 1 \geq 8 + K$$

- ▶ Para satisfazer a inequação é preciso que K possua pelo menos 4 bits
 - ▶ A palavra com correção de erro possui 12 bits, onde os bits de K_i são posicionados pela fórmula 2^i



Memória interna

- ▶ Código de Hamming
 - ▶ Calculando o código de correção K
 - ▶ Cada bit K_i é calculado a partir do dado M_j que possui a posição i com bit igual a 1



$$K_0 = M_0 \oplus M_1 \oplus M_3 \oplus M_4 \oplus M_6$$

Memória interna

- ▶ Código de Hamming
 - ▶ Calculando o código de correção K
 - ▶ Cada bit K_i é calculado a partir do dado M_j que possui a posição i com bit igual a 1

	↓	↓			↓	↓			↓	↓	
M_7	M_6	M_5	M_4	K_3	M_3	M_2	M_1	K_2	M_0	K_1	K_0
1100	10 <u>1</u> 1	10 <u>1</u> 0	1001	1000	01 <u>1</u> 1	01 <u>1</u> 0	0101	0100	00 <u>1</u> 1	0010	0001

$$K_0 = M_0 \oplus M_1 \oplus M_3 \oplus M_4 \oplus M_6$$

$$K_1 = M_0 \oplus M_2 \oplus M_3 \oplus M_5 \oplus M_6$$

Memória interna

- ▶ Código de Hamming
 - ▶ Calculando o código de correção K
 - ▶ Cada bit K_i é calculado a partir do dado M_j que possui a posição i com bit igual a 1

↓					↓	↓	↓	↓↓			
M_7	M_6	M_5	M_4	K_3	M_3	M_2	M_1	K_2	M_0	K_1	K_0
<u>1</u> 100	1011	1010	1001	1000	<u>0</u> 111	<u>0</u> 110	<u>0</u> 101	0100	0011	0010	0001

$$K_0 = M_0 \oplus M_1 \oplus M_3 \oplus M_4 \oplus M_6$$

$$K_1 = M_0 \oplus M_2 \oplus M_3 \oplus M_5 \oplus M_6$$

$$K_2 = M_1 \oplus M_2 \oplus M_3 \oplus M_7$$

Memória interna

- ▶ Código de Hamming
 - ▶ Calculando o código de correção K
 - ▶ Cada bit K_i é calculado a partir do dado M_j que possui a posição i com bit igual a 1

↓	↓	↓	↓	↓↓							
M_7	M_6	M_5	M_4	K_3	M_3	M_2	M_1	K_2	M_0	K_1	K_0
<u>1</u> 100	<u>1</u> 011	<u>1</u> 010	<u>1</u> 001	1000	0111	0110	0101	0100	0011	0010	0001

$$K_0 = M_0 \oplus M_1 \oplus M_3 \oplus M_4 \oplus M_6$$

$$K_1 = M_0 \oplus M_2 \oplus M_3 \oplus M_5 \oplus M_6$$

$$K_2 = M_1 \oplus M_2 \oplus M_3 \oplus M_7$$

$$K_3 = M_4 \oplus M_5 \oplus M_6 \oplus M_7$$

Memória interna

- ▶ Exemplo de uso do código de Hamming
 - ▶ Palavra de dado M é 133 (10000101)
 - ▶ O código de correção K é 1001

1	0	0	0	1	0	1	0	0	1	0	1
M_7	M_6	M_5	M_4	K_3	M_3	M_2	M_1	K_2	M_0	K_1	K_0
<u>1</u> 00	<u>1</u> 011	<u>1</u> 010	<u>1</u> 001	1000	0111	0110	0101	0100	0011	0010	0001

$$K_0 = M_0 \oplus M_1 \oplus M_3 \oplus M_4 \oplus M_6$$

$$K_1 = M_0 \oplus M_2 \oplus M_3 \oplus M_5 \oplus M_6$$

$$K_2 = M_1 \oplus M_2 \oplus M_3 \oplus M_7$$

$$K_3 = M_4 \oplus M_5 \oplus M_6 \oplus M_7$$

Memória interna

- ▶ Exemplo de uso do código de Hamming
 - ▶ Invertendo o bit 5 da palavra de dado M
 - ▶ O novo código de correção K' é 0011

1	0	1	0	1	0	1	0	0	1	0	1
M_7	M_6	M_5	M_4	K_3	M_3	M_2	M_1	K_2	M_0	K_1	K_0
1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001

$$K_0 = M_0 \oplus M_1 \oplus M_3 \oplus M_4 \oplus M_6$$

$$K_1 = M_0 \oplus M_2 \oplus M_3 \oplus M_5 \oplus M_6$$

$$K_2 = M_1 \oplus M_2 \oplus M_3 \oplus M_7$$

$$K_3 = M_4 \oplus M_5 \oplus M_6 \oplus M_7$$

Memória interna

- ▶ Exemplo de uso do código de Hamming
 - ▶ Invertendo o bit 5 da palavra de dado M
 - ▶ O novo código de correção K' é 0011

1	0	1	0	1	0	1	0	0	1	0	1
M_7	M_6	M_5	M_4	K_3	M_3	M_2	M_1	K_2	M_0	K_1	K_0
1100	1011	<u>1010</u>	1001	1000	0111	0110	0101	0100	0011	0010	0001

$$K_0 = M_0 \oplus M_1 \oplus M_3 \oplus M_4 \oplus M_6$$

$$K_1 = M_0 \oplus M_2 \oplus M_3 \oplus M_5 \oplus M_6$$

$$K_2 = M_1 \oplus M_2 \oplus M_3 \oplus M_7$$

$$K_3 = M_4 \oplus M_5 \oplus M_6 \oplus M_7$$

$K \oplus K' = 1001 \oplus 0011 = 1010$ é a posição do bit com erro

Memória interna

- ▶ Custo da correção de erro na memória
 - ▶ Correção de erro simples de bit

# Bits de dados	# Bits de correção	Aumento de células
8	4	50%
16	5	31,25%
32	6	18,75%
64	7	10,94%
128	8	6,25%

Memória externa

- ▶ Dispositivos de grande capacidade de armazenamento não volátil
 - ▶ Mecânicos
 - ▶ Discos e fitas magnéticas (HDD, DAT, ...)
 - ▶ Mídias óticas (CD, DVD, ...)
 - ▶ Eletrônicos
 - ▶ Discos de estado sólido (SSD)
 - ▶ Mídias removíveis (USB, SD, ...)

Memória externa

- ▶ Dispositivos mecânicos
 - ▶ Tanto as mídias magnéticas como óticas dependem de componentes mecânicos para acessar os dados
 - ▶ O leitor é posicionado no setor para escrever ou ler os dados, utilizando movimentos de rotação da mídia

$$t_{\text{acesso}} = t_{\text{busca}} + t_{\text{rotação}} + t_{E/S}$$

Memória externa

- ▶ Dispositivos mecânicos
 - ▶ Tanto as mídias magnéticas como óticas dependem de componentes mecânicos para acessar os dados
 - ▶ O leitor é posicionado no setor para escrever ou ler os dados, utilizando movimentos de rotação da mídia

$$t_{\text{acesso}} = t_{\text{busca}} + t_{\text{rotação}} + t_{E/S}$$

- ▶ O tempo médio de rotação é de $\frac{1}{2 \times RPS}$ que é o tempo para posicionamento no extremo oposto da mídia

Memória externa

- ▶ Dispositivos mecânicos

- ▶ Tanto as mídias magnéticas como óticas dependem de componentes mecânicos para acessar os dados
- ▶ O leitor é posicionado no setor para escrever ou ler os dados, utilizando movimentos de rotação da mídia

$$t_{\text{acesso}} = t_{\text{busca}} + t_{\text{rotação}} + t_{E/S}$$

- ▶ O tempo médio de rotação é de $\frac{1}{2 \times RPS}$ que é o tempo para posicionamento no extremo oposto da mídia
- ▶ A operação de E/S depende do número de bytes transferidos B e da quantidade de bytes N por trilha, com um tempo definido por $\frac{B}{N \times RPS}$ segundos

Memória externa

- ▶ Transferência sequencial de 16 MB do disco rígido
 - ▶ Tempo de busca médio de 4 ms
 - ▶ Disco com rotação de 15.000 RPM = 250 RPS
 - ▶ Com 512 bytes por setor e 1000 setores por trilha, serão acessadas 32 trilhas (500 KB) adjacentes do disco

Memória externa

- ▶ Transferência sequencial de 16 MB do disco rígido
 - ▶ Tempo de busca médio de 4 ms
 - ▶ Disco com rotação de 15.000 RPM = 250 RPS
 - ▶ Com 512 bytes por setor e 1000 setores por trilha, serão acessadas 32 trilhas (500 KB) adjacentes do disco

$$t_{busca} = 4 \times 10^{-3} \text{ s}$$

$$t_{rotação} = \frac{1}{2 \times 250} \approx 2 \times 10^{-3} \text{ s}$$

$$t_{E/S} = \frac{500000}{512000 \times 250} \approx 4 \times 10^{-3} \text{ s}$$

Memória externa

- ▶ Transferência sequencial de 16 MB do disco rígido
 - ▶ Tempo de busca médio de 4 ms
 - ▶ Disco com rotação de 15.000 RPM = 250 RPS
 - ▶ Com 512 bytes por setor e 1000 setores por trilha, serão acessadas 32 trilhas (500 KB) adjacentes do disco

$$t_{busca} = 4 \times 10^{-3} \text{ s}$$

$$t_{rotação} = \frac{1}{2 \times 250} \approx 2 \times 10^{-3} \text{ s}$$

$$t_{E/S} = \frac{500000}{512000 \times 250} \approx 4 \times 10^{-3} \text{ s}$$

$$\begin{aligned} t_{acesso} &= t_{busca} + 32 \times (t_{rotação} + t_{E/S}) \\ &= 4 \times 10^{-3} + 32 \times (2 \times 10^{-3} + 4 \times 10^{-3}) \text{ s} \\ &= 196 \text{ ms} \end{aligned}$$

Memória externa

- ▶ Transferência sequencial de 16 MB do disco rígido
 - ▶ Tempo de busca médio de 4 ms
 - ▶ Disco com rotação de 15.000 RPM = 250 RPS
 - ▶ Com 512 bytes por setor e 1000 setores por trilha, serão acessadas 32 trilhas (500 KB) adjacentes do disco

$$t_{busca} = 4 \times 10^{-3} \text{ s}$$

$$t_{rotação} = \frac{1}{2 \times 250} \approx 2 \times 10^{-3} \text{ s}$$

$$t_{E/S} = \frac{500000}{512000 \times 250} \approx 4 \times 10^{-3} \text{ s}$$

$$\begin{aligned} t_{acesso} &= t_{busca} + 32 \times (t_{rotação} + t_{E/S}) \\ &= 4 \times 10^{-3} + 32 \times (2 \times 10^{-3} + 4 \times 10^{-3}) \text{ s} \\ &= 196 \text{ ms} \end{aligned}$$

Qual o tempo de acesso para o disco fragmentado?

Memória externa

- ▶ Transferência de 16 MB do disco rígido
 - ▶ Dados aleatoriamente fragmentados no disco
 - ▶ As informações estão armazenadas em 10 setores por trilha, necessitando de 3.125 acessos em trilhas diferentes para leitura completa dos dados

Memória externa

- ▶ Transferência de 16 MB do disco rígido
 - ▶ Dados aleatoriamente fragmentados no disco
 - ▶ As informações estão armazenadas em 10 setores por trilha, necessitando de 3.125 acessos em trilhas diferentes para leitura completa dos dados

$$t_{busca} = 4 \times 10^{-3} s$$

$$t_{rotação} = \frac{1}{2 \times 250} \approx 2 \times 10^{-3} s$$

$$t_{E/s} = \frac{5120}{512000 \times 250} = 0,04 \times 10^{-3} s$$

Memória externa

- ▶ Transferência de 16 MB do disco rígido
 - ▶ Dados aleatoriamente fragmentados no disco
 - ▶ As informações estão armazenadas em 10 setores por trilha, necessitando de 3.125 acessos em trilhas diferentes para leitura completa dos dados

$$t_{busca} = 4 \times 10^{-3} \text{ s}$$

$$t_{rotação} = \frac{1}{2 \times 250} \approx 2 \times 10^{-3} \text{ s}$$

$$t_{E/S} = \frac{5120}{512000 \times 250} = 0,04 \times 10^{-3} \text{ s}$$

$$\begin{aligned} t_{acesso} &= 3125 \times (t_{busca} + t_{rotação} + t_{E/S}) \\ &= 3125 \times (4 \times 10^{-3} + 2 \times 10^{-3} + 0,04 \times 10^{-3}) \text{ s} \\ &= 18,875 \text{ s} \end{aligned}$$

Memória externa

- ▶ Transferência de 16 MB do disco rígido
 - ▶ Dados aleatoriamente fragmentados no disco
 - ▶ As informações estão armazenadas em 10 setores por trilha, necessitando de 3.125 acessos em trilhas diferentes para leitura completa dos dados

$$t_{busca} = 4 \times 10^{-3} \text{ s}$$

$$t_{rotação} = \frac{1}{2 \times 250} \approx 2 \times 10^{-3} \text{ s}$$

$$t_{E/s} = \frac{5120}{512000 \times 250} = 0,04 \times 10^{-3} \text{ s}$$

$$\begin{aligned} t_{acesso} &= 3125 \times (t_{busca} + t_{rotação} + t_{E/s}) \\ &= 3125 \times (4 \times 10^{-3} + 2 \times 10^{-3} + 0,04 \times 10^{-3}) \text{ s} \\ &= 18,875 \text{ s} \end{aligned}$$

Transferência cerca de 100x mais lenta!

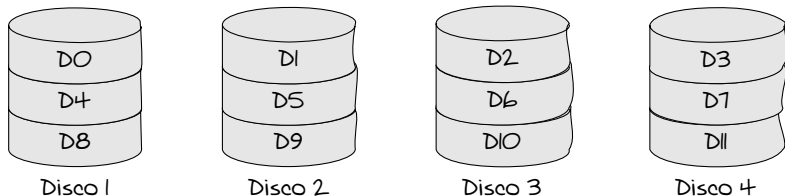
Memória externa

- ▶ *Redundant Array of Independent Disks* (RAID)
 - ▶ Técnicas para aumentar o desempenho e redundância utilizando múltiplos discos

Nível	Categoria	# Discos	Descrição
0	Divisão	N	Sem redundância
1	Espelhamento	$2N$	Redundância dos dados
2	Acesso paralelo	$N + \log N$	Redundância com Hamming
3		$N + 1$	Paridade com bits intercalados
4	Acesso independente dos discos	$N + 1$	Paridade com blocos intercalados
5		$N + 1$	Paridade distribuída com blocos intercalados
6		$N + 2$	Paridade dupla distribuída com blocos intercalados

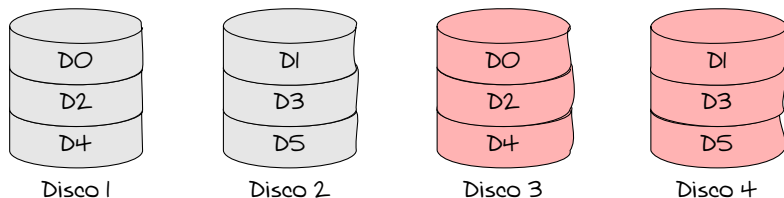
Memória externa

- ▶ *Redundant Array of Independent Disks (RAID)*
 - ▶ Nível 0 (dados divididos nos discos)



Memória externa

- ▶ *Redundant Array of Independent Disks (RAID)*
 - ▶ Nível 1 (redundância com espelhamento nos discos)



Memória externa

- ▶ *Redundant Array of Independent Disks* (RAID)
 - ▶ Nível 2 (redundância por código de Hamming)



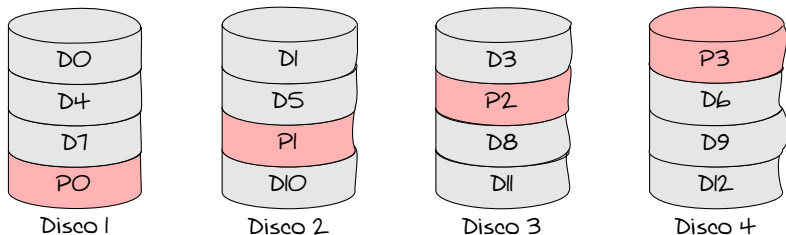
Memória externa

- ▶ *Redundant Array of Independent Disks* (RAID)
 - ▶ Níveis 3 e 4 (paridade de bits e de blocos)



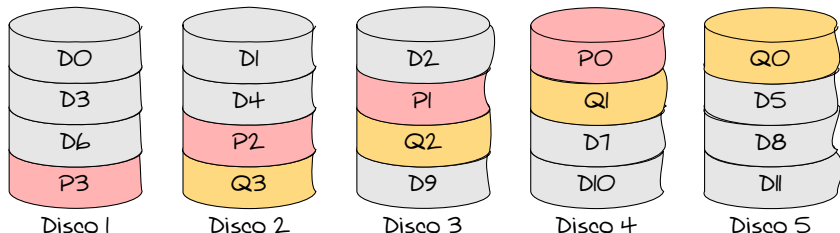
Memória externa

- ▶ *Redundant Array of Independent Disks (RAID)*
 - ▶ Nível 5 (paridade distribuída de blocos)



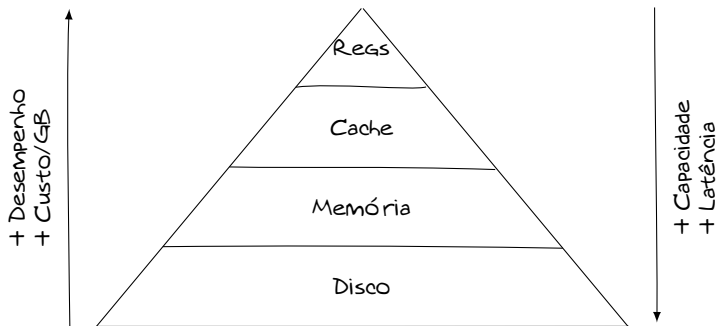
Memória externa

- ▶ *Redundant Array of Independent Disks (RAID)*
 - ▶ Nível 6 (paridade dupla distribuída de blocos)



Hierarquia de memória

- ▶ O que é hierarquia de memória?
 - ▶ Vários tipos de memória com diferentes tamanhos e velocidades de acesso



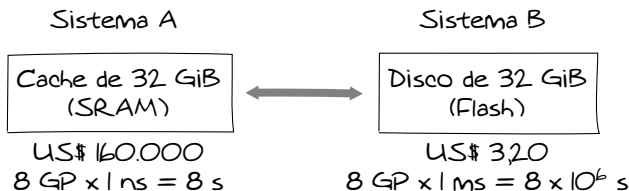
Hierarquia de memória

- ▶ Tempo de acesso e custo por GB típicos de diferentes tecnologias de armazenamento

Tipo	Capacidade	Custo	Latência
SRAM	2 KiB <-> 32 MBit	~US\$ 5k / GiB	~1 ns
DRAM	1 <-> 32 GiB	~US\$ 3 / GiB	~10 ns
Flash	64 <-> 1.000 GiB	~US\$ 0,10 / GiB	~1 ms

Hierarquia de memória

- ▶ Qual o propósito da hierarquia de memória?
 - ▶ Abstrair e combinar as tecnologias que estão sendo utilizadas e reduzir as limitações associadas
 - ▶ Otimizar a relação entre desempenho e custo das tecnologias de armazenamento



Hierarquia de memória

- ▶ Qual o propósito da hierarquia de memória?
 - ▶ Combinar diferentes tecnologias de armazenamento para maximizar o desempenho e reduzir o custo total
 - ▶ Considerando que o dado está disponível na cache em 90% dos acessos a memória

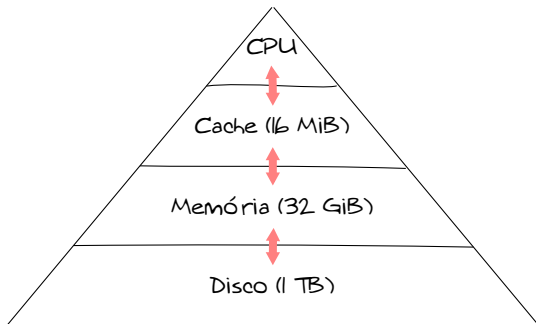
Sistema C

Cache de 16 MiB (SRAM) Memória de 32 GiB (DRAM) Disco de 256 GB (Flash)

$$\text{US\$ } 80 + \text{US\$ } 96 + \text{US\$ } 2560 = \text{US\$ } 20160$$
$$7,2 \text{ GP} \times 1 \text{ ns} + 0,8 \text{ GP} \times 10 \text{ ns} = 15,2 \text{ s}$$

Hierarquia de memória

- ▶ Fluxo de dados na hierarquia de memória
 - ▶ A medida que os dados vão sendo acessados, eles são armazenados nos níveis mais altos



Para reduzir a latência são utilizadas técnicas para maximizar a disponibilidade dos dados nos níveis mais altos da hierarquia