

Sistemas de entrada/saída de dados

João Canas Ferreira

Maio 2016



1 Aspetos gerais

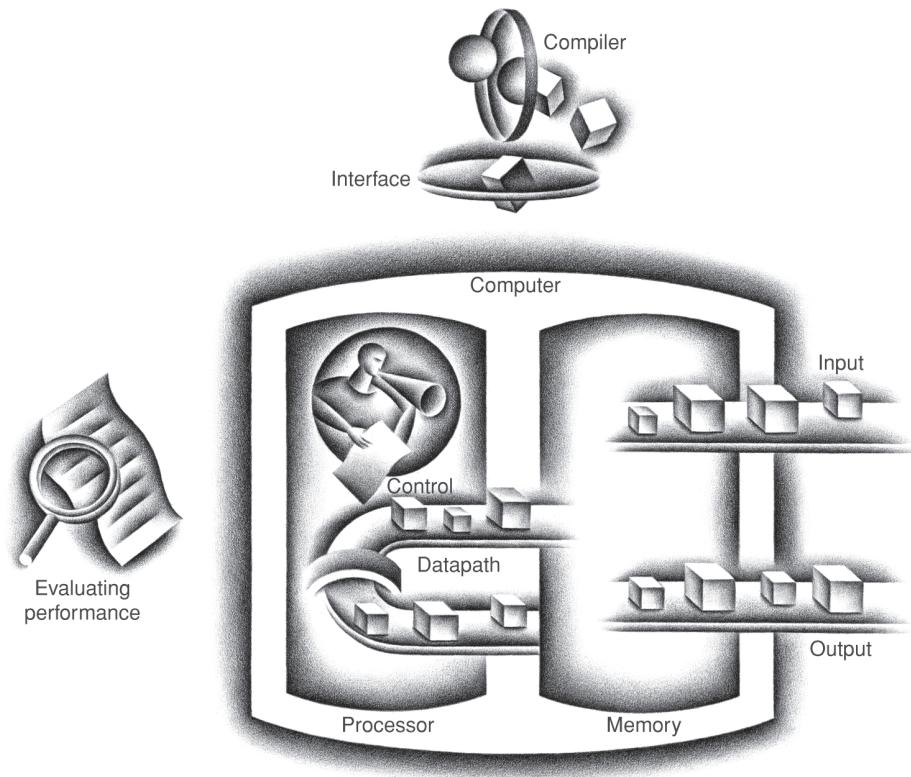
2 Armazenamento de informação

3 Comunicação com periféricos

4 Gestão de periféricos

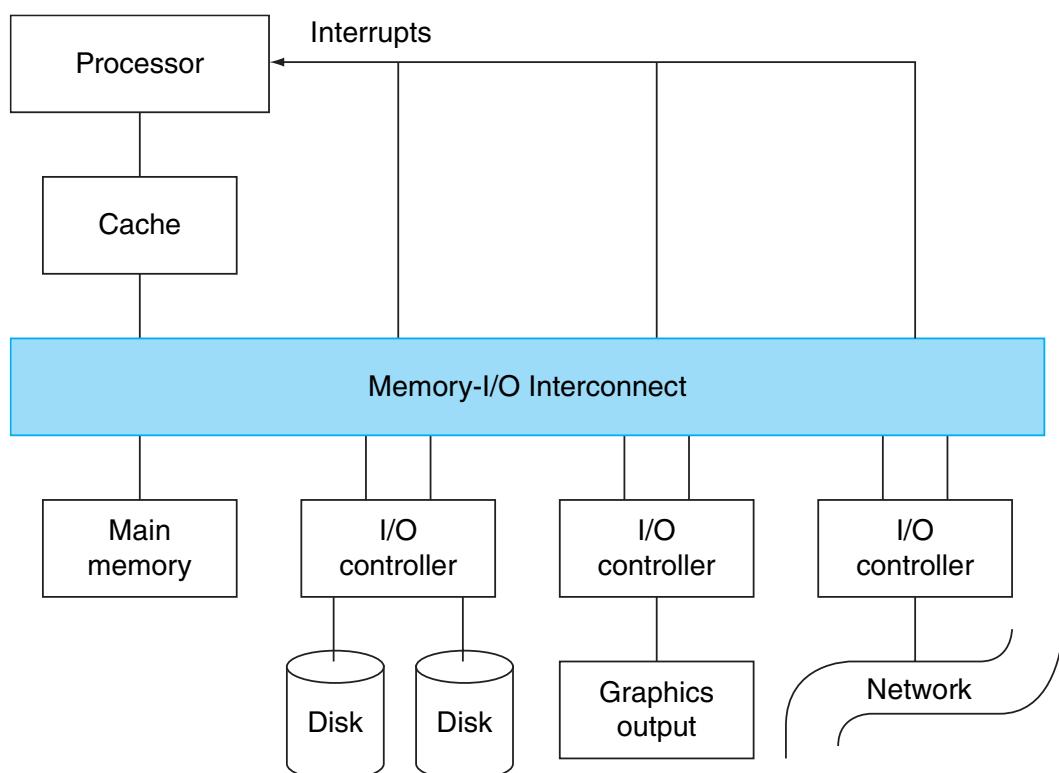
5 Sistemas RAID

Organização de um computador



Fonte: [COD4]

Dispositivos de entrada/saída



Fonte: [COD4]

Diversidade de periféricos

Device	Behavior	Partner	Data rate (Mbit/sec)
Keyboard	Input	Human	0.0001
Mouse	Input	Human	0.0038
Voice input	Input	Human	0.2640
Sound input	Input	Machine	3.0000
Scanner	Input	Human	3.2000
Voice output	Output	Human	0.2640
Sound output	Output	Human	8.0000
Laser printer	Output	Human	3.2000
Graphics display	Output	Human	800.0000–8000.0000
Cable modem	Input or output	Machine	0.1280–6.0000
Network/LAN	Input or output	Machine	100.0000–10000.0000
Network/wireless LAN	Input or output	Machine	11.0000–54.0000
Optical disk	Storage	Machine	80.0000–220.0000
Magnetic tape	Storage	Machine	5.0000–120.0000
Flash memory	Storage	Machine	32.0000–200.0000
Magnetic disk	Storage	Machine	800.0000–3000.0000

Fonte: [COD4]

Características de um sistema E/S

- *Confiabilidade* é importante
 - Em especial, para armazenagem de informação (discos magnéticos)
- Medidas de desempenho
 - Latência (tempo de resposta)
 - Débito (“largura de banda”)
Quantidade de informação processada por unidade de tempo
- PC desktop e sistemas embarcados
 - tempo de resposta e diversidade
- Servidores
 - débito e expansibilidade

Confiabilidade

- *Confiabilidade*: qualidade do sistema que permite confiar justificadamente no serviço oferecido.
- Vários aspectos quantificáveis:
 - Fiabilidade (*reliability*)
Quantificação:
 - ➡ medida do tempo de funcionamento até falhar
 - ➡ probabilidade de não falhar durante o tempo de missão
- Disponibilidade (*availability*)
Quantificação:
 - ➡ tempo (ou %) em que o sistema está operacional
 - ➡ tempo médio de reparação
- Reparabilidade (*Maintainability*)
- Segurança contra acidentes (*safety*)
- Segurança contra acesso não autorizado (*security*)
Fatores:
 - ➡ integridade, confidencialidade, autenticidade

Medidas de confiabilidade

- Fiabilidade: tempo médio até falhar
 - ➡ MTTF: mean time to failure
- Interrupção de serviço: tempo médio de reparação
 - ➡ MTTR: mean time to repair
- Tempo médio entre falhas:
 - ➡ MTBF: mean time between failures
 - ➡ $MTBF = MTTF + MTTR$
- Disponibilidade: $MTTF / (MTTF + MTTR)$
- Para aumentar a disponibilidade:
 - Aumentar MTTF
 - ➡ tolerância a falhas (redundância), previsão de falhas, etc.
 - Reduzir MTTR
 - ➡ melhores ferramentas e processos de diagnóstico e reparação

Exemplo: Falhas de discos na empresa Google

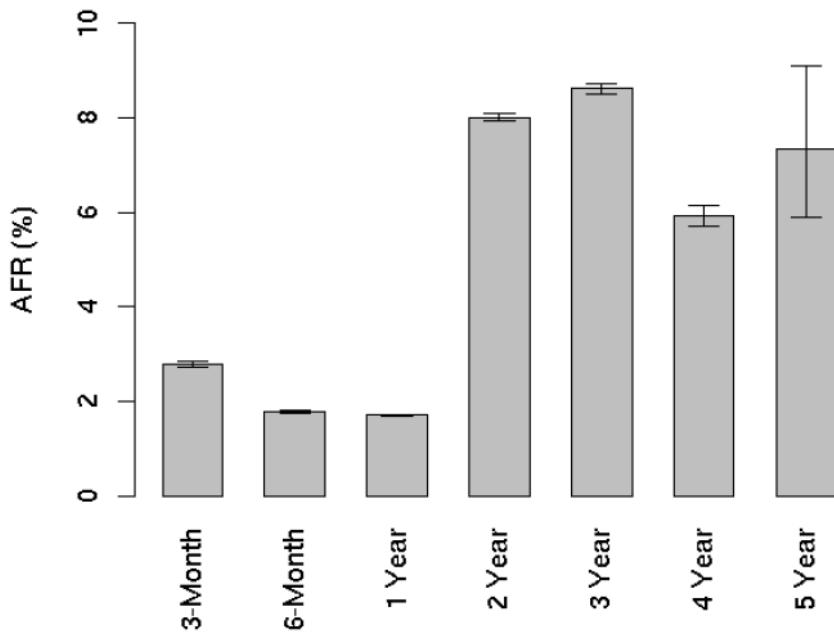


Figure 2: Annualized failure rates broken down by age groups

Fonte: E. Pinheiro et al, "Failure Trends in a Large Disk Drive Population", 5th USENIX Conference on File and Storage Technologies (FAST'07), Fev. 2007

1 Aspetos gerais

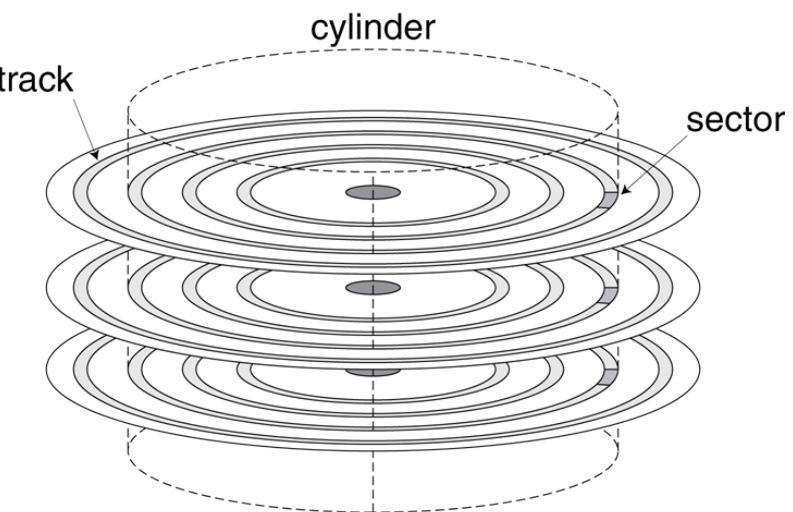
2 Armazenamento de informação

3 Comunicação com periféricos

4 Gestão de periféricos

5 Sistemas RAID

Discos magnéticos

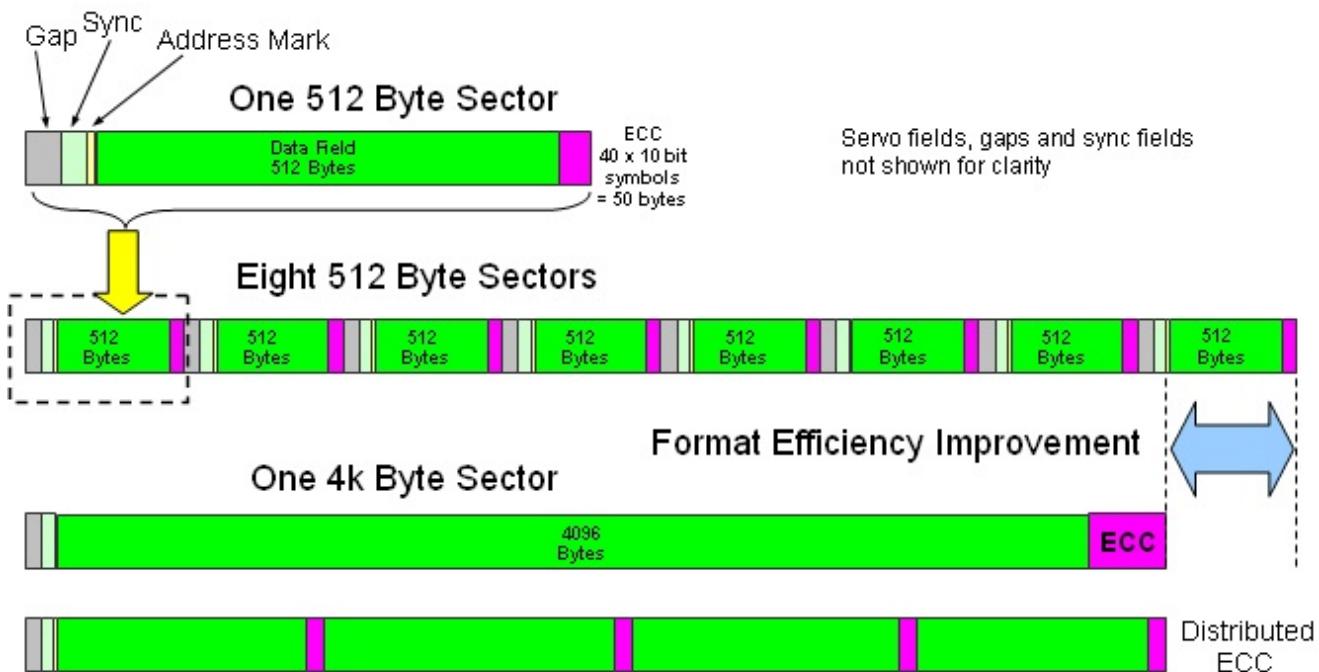


- Pista (*track*): coroa circular numa face de um prato
- Cilindro (*cylinder*): conjunto de pistas que podem ser simultaneamente lidas/escritas
- Setor: segmento de uma pista; unidade básica de armazenamento

Acesso a um setor

- Cada setor físico contém:
 - identificador do setor
 - dados
tradicional: 512 bytes
a partir de 2009: 4096 bytes
 - código corretor de erros (ECC)
permite “esconder” defeitos e problemas de gravação
 - informação de sincronização e lacunas
- Aceder a um setor envolve:
 - esperar se existirem acessos pendentes (fila de espera)
 - movimento da cabeça de leitura: *seek time*
 - latência rotacional: tempo até setor surgir debaixo da cabeça de leitura
 - tempo de transferência dos dados
 - *overhead* do controlador

Formato de setores



[Fonte: Wikipedia]

Exemplo de cálculo de tempo de acesso a disco

Calcular tempo médio de leitura:

- Setor: 512 B
- Velocidade de rotação: 15000 rpm
- Tempo médio de procura: 4 ms
- Taxa de transferência: 100 MB/s
- *Overhead* de controlador: 0,2 ms
- Disco inicialmente inativo

$$t_a = t_{\text{seek}} + t_{\text{rot}} + t_{\text{transf}} + t_{\text{ctrl}}$$

$$t_{\text{rot}} = 0,5 \times \frac{60}{15000} = 2 \text{ ms}$$

$$t_{\text{transf}} = \frac{512}{10^8} = 0.005 \text{ ms}$$

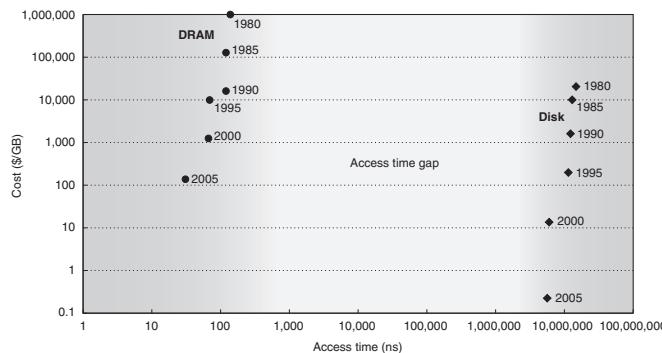
No total: $t_a = 4 + 2 + 0.005 + 0.2 = 6.2 \text{ ms}$

Fatores que afetam o cálculo do desempenho

- Fabricantes indicam o tempo médio de procura t_{seek} baseado na média de todos os acessos
- Na prática, setores sucessivos estão relativamente próximos (proximidade)
 - tempos de procura reais são tipicamente mais baixos que t_{seek}
- Controladores inteligentes determinam a posição física dos setores
 - Apresentam uma interface ao sistema hóspede baseada em setores “lógicos”
- Discos incluem memória *cache*
 - leitura por antecipação (*prefetch*)
 - podem evitar tempo de pesquisa e latência de rotação

Características de discos magnéticos

- ➡ Densidade por unidade de área: $\frac{\text{pistas}}{\text{polegada}} \times \frac{\text{bits}}{\text{polegada}}$
- ➡ Custo e tempo de acesso

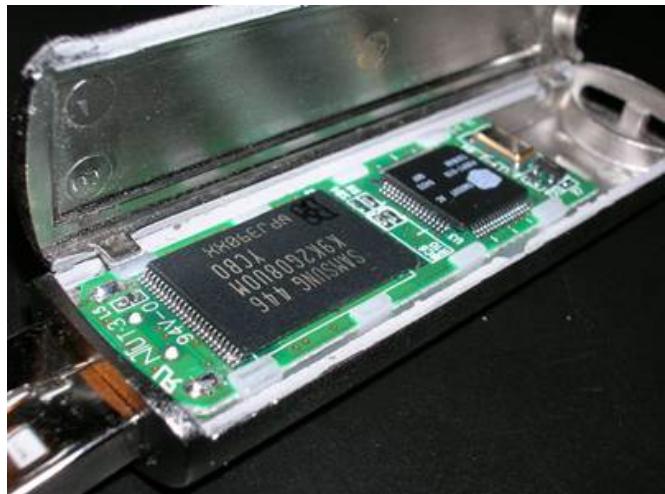


Fonte: [CAQA5]

- ➡ Potência dissipada: $P \propto \text{diâmetro}^{4.6} \times \text{RPM}^{2.8} \times \text{num. de pratos}$

	Capacity (GB)	Price	Platters	RPM	Diameter (inches)	Average seek (ms)	Power (watts)	I/O/sec	Disk BW (MB/sec)	Buffer BW (MB/sec)	Buffer size (MB)	MTTF (hrs)
SATA	2000	\$85	4	5900	3.7	16	12	47	45-95	300	32	0.6M
SAS	600	\$400	4	15,000	2.6	3-4	16	285	122-204	750	16	1.6M

Fonte: [CAQA5]



- Memória *não volátil*

- 100x – 1000x mais rápida que discos magnéticos
- menor consumo, maior robustez
- mais cara em €/GB (entre disco magnético e DRAM)

Tipos de memória Flash

- Flash do tipo NOR

- célula de armazenamento com estrutura semelhante a porta NOR
- acesso direto para leitura/escrita (como SRAM e DRAM)
- memória de instruções em sistemas embarcados

- Flash do tipo NAND

- célula de armazenamento com estrutura semelhante a porta NAND
- mais densa (bits/área), mas acesso é sequencial por blocos
- mais barata

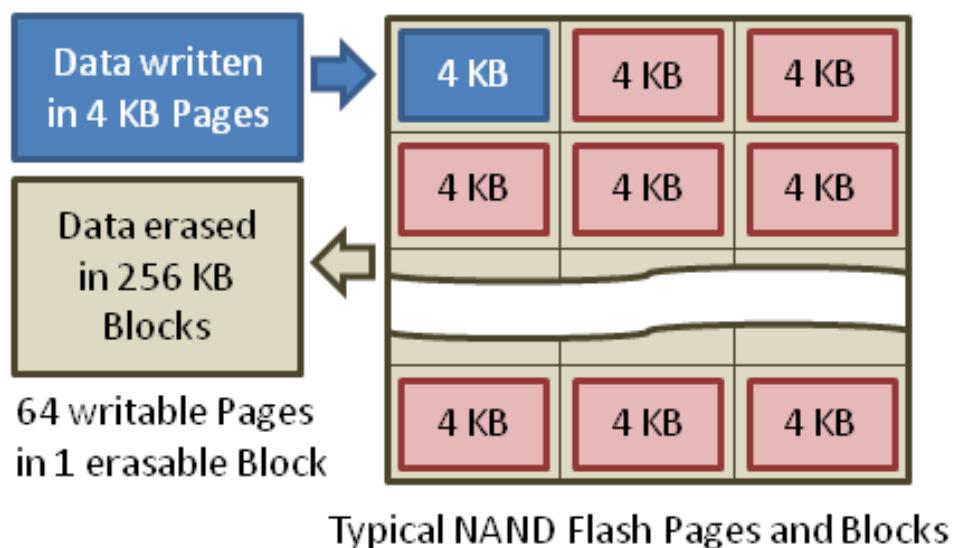
- células perdem capacidade de armazenamento

- gestão de desgaste: mapear dados para zonas menos usadas

Características de memórias Flash

Characteristics	NOR Flash Memory	NAND Flash Memory
Typical use	BIOS memory	USB key
Minimum access size (bytes)	512 bytes	2048 bytes
Read time (microseconds)	0.08	25
Write time (microseconds)	10.00	1500 to erase + 250
Read bandwidth (MBytes/second)	10	40
Write bandwidth (MBytes/second)	0.4	8
Wearout (writes per cell)	100,000	10,000 to 100,000
Best price/GB (2008)	\$65	\$4

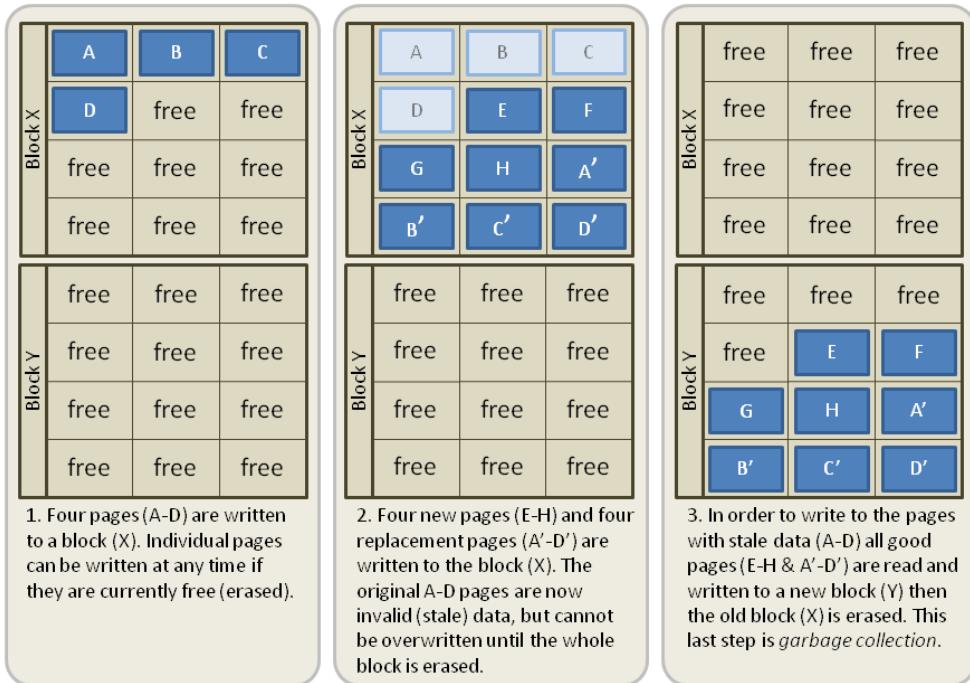
Estrutura interna de NAND Flash: páginas e blocos



[Fonte: Wikipedia]

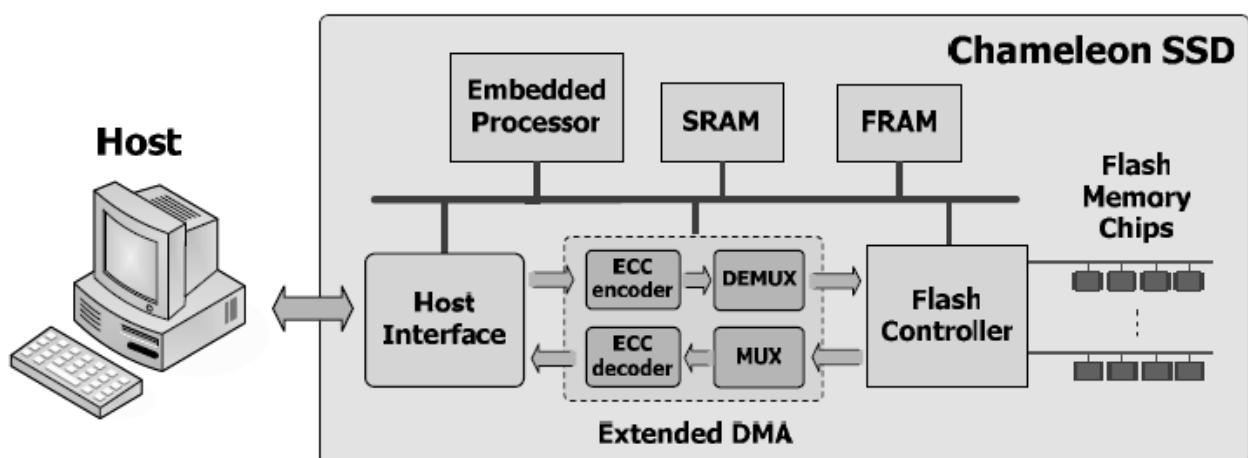
- Escrever por páginas
- Apagar por blocos

Gestão de espaço



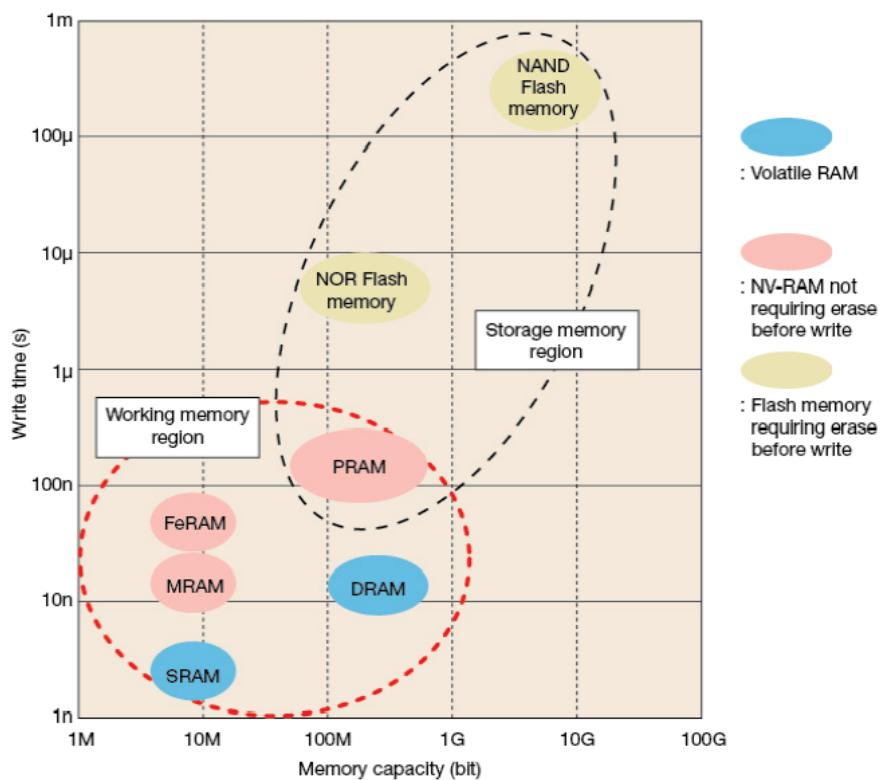
[Fonte: Wikipedia]

Exemplo: Arquitetura de um Solid State Drive



[Fonte: Chameleon: A High Performance Flash/FRAM Hybrid Solid State Disk Architecture, Soon et al., IEEE-CAL Vol.7, N.1, 2008]

Outras tecnologias para memórias não-voláteis



Courtesy: Motoyuki Ooishi

1 Aspetos gerais

2 Armazenamento de informação

3 Comunicação com periféricos

4 Gestão de periféricos

5 Sistemas RAID

Interligar dispositivos

➡ Como interligar CPU, memória e controladores de E/S ?

- Solução mais simples: barramento paralelo
- É um canal de comunicação *partilhado*
- Conjunto de ligações paralelas para dados e sincronização
- Problema: pode limitar o desempenho do sistema (“gargalo”)
- O desempenho de um barramento está limitado por fatores físicos
 - comprimento das ligações
 - número de dispositivos ligados (*slots*)
- Alternativa: ligações série de alta velocidade (ponto-a-ponto)

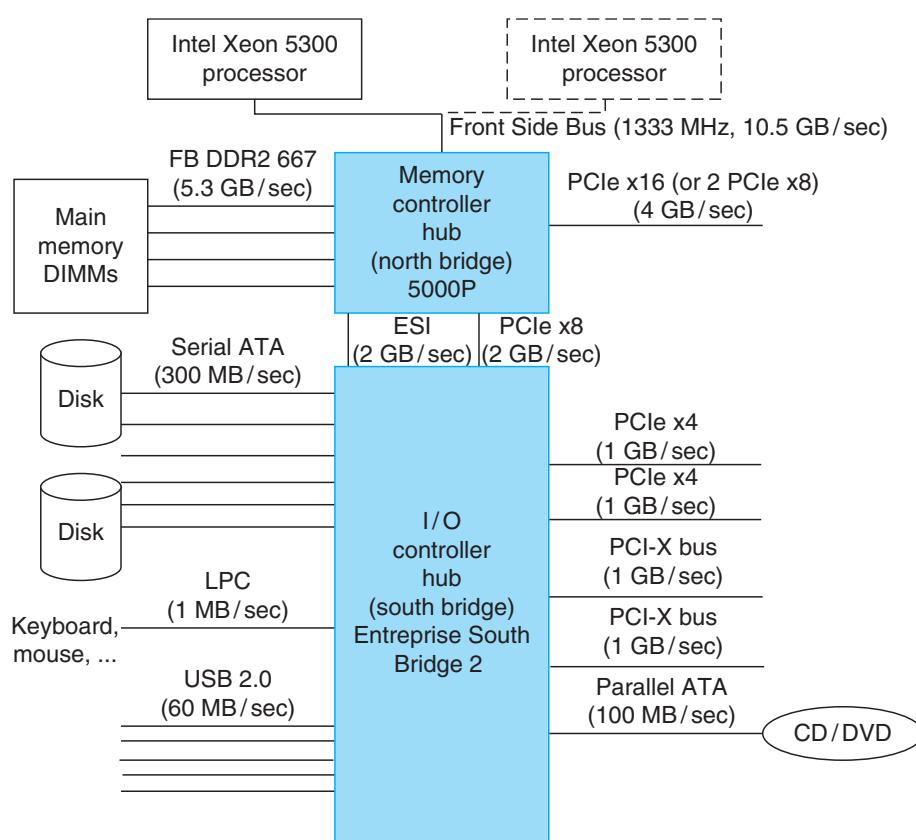
Tipos de barramentos

- **Barramentos processador/memória**
- Curto, alta velocidade
- Características adaptadas à organização da memória
- Exemplo: FSB: *Front Side Bus*
- **Barramento de E/S (periféricos)**
- Mais comprido
- Múltiplas ligações
- Normalizado, para garantir interoperabilidade
- Ligado ao CPU através de uma *ponte*
- Exemplos: PCI, USB, LPC, etc.

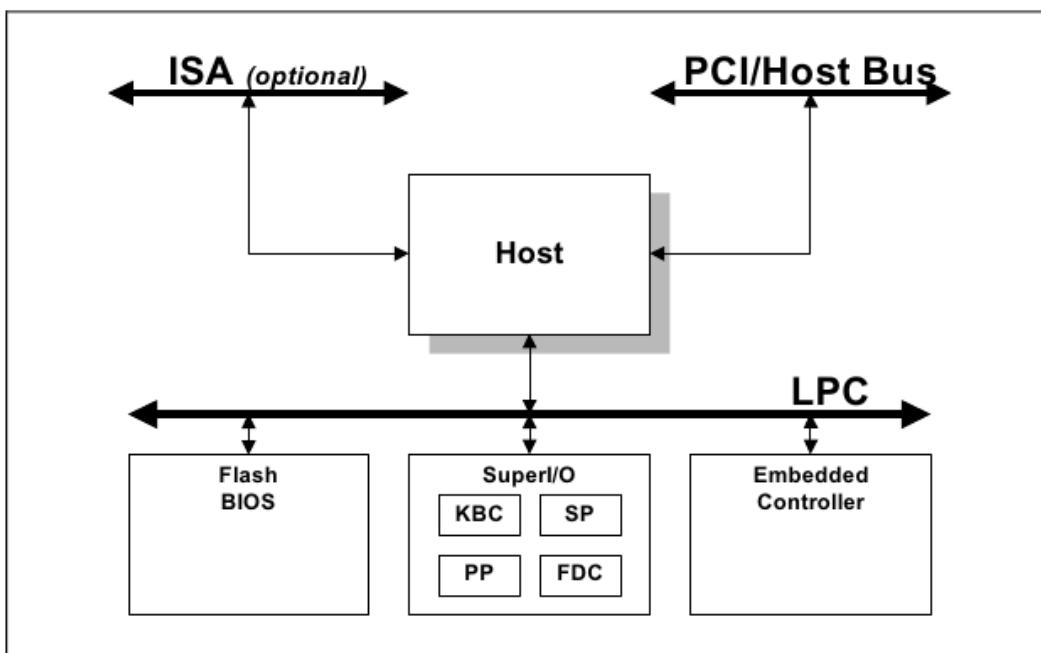
Alguns barramentos atuais

	Firewire	USB 2.0	PCI Express	Serial ATA	Serial Attached SCSI
Intended use	External	External	Internal	Internal	External
Devices per channel	63	127	1	1	4
Data width	4	2	2/lane	4	4
Peak bandwidth	50MB/s or 100MB/s	0.2MB/s, 1.5MB/s, or 60MB/s	250MB/s/lane 1x, 2x, 4x, 8x, 16x, 32x	300MB/s	300MB/s
Hot pluggable	Yes	Yes	Depends	Yes	Yes
Max length	4.5m	5m	0.5m	1m	8m
Standard	IEEE 1394	USB Implementers Forum	PCI-SIG	SATA-IO	INCITS TC T10

Sistema E/S típico (*x86*)



Exemplo: Barramento LPC (Low Pin Count)



KBC: controlador de teclado, SP: porto série(RS232), PP: porto paralelo ,
FDC: controlador de diskette (*floppy disk controller*)

1 Aspetos gerais

2 Armazenamento de informação

3 Comunicação com periféricos

4 Gestão de periféricos

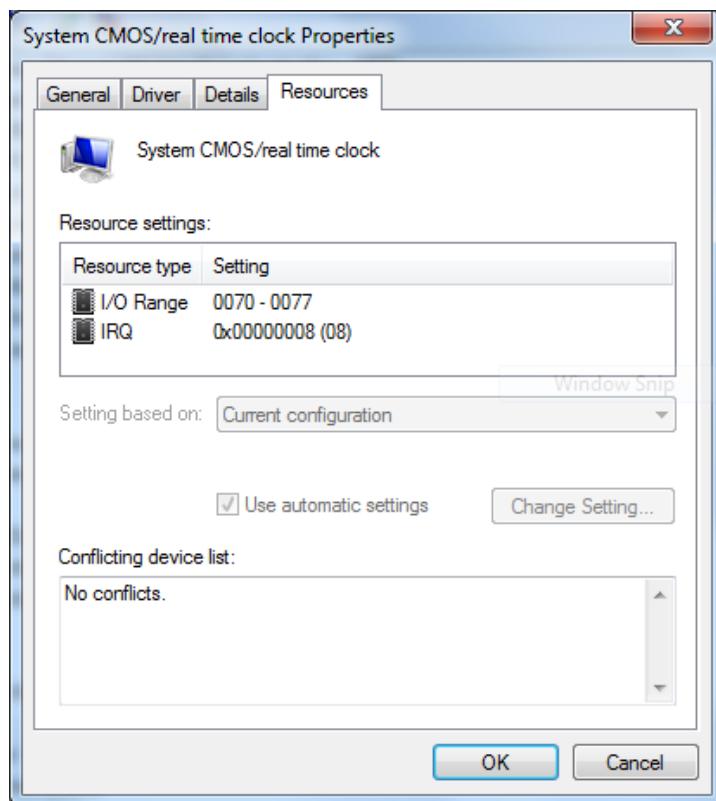
5 Sistemas RAID

- **E/S são geridas pelo sistema operativo**
- Vários programas partilham os recursos E/S
 - Proteção e sequenciamento da utilização
- E/S causam interrupções assíncronas
 - Gerir as rotinas de atendimento de interrupções
- Programação de E/S é “delicada”
 - Sistema operativo disponibiliza abstrações (p.ex. ficheiros)
- **Dispositivos são geridos por controladores de E/S em hardware**
 - Transferências entre memória (ou CPU) e dispositivos
- Controlador tem:
 - Registos de comandos: indicam a tarefa a executar
 - Registos de estado: indicam atividade em execução e situações de erro
 - Registos de dados:
 - escrita (CPU/memória para dispositivo)
 - leitura (do dispositivo para CPU/memória)

Acesso a dispositivos de E/S

- **Mapeamento no espaço de endereçamento de memória**
- Registos de E/S são mapeados em endereços de memória
- Circuito de descodificação de endereços distingue entre memória e dispositivos de E/S
- Sistema operativo usa unidade de gestão de memória virtual para “ocultar” os dispositivos dos programas do utilizador
- Pode ser usado com qualquer CPU
- **Instruções dedicadas de E/S**
- CPU dispõe de instruções separadas para acesso a dispositivos de E/S
- Instruções só podem ser executadas em modo privilegiado
- Exemplo: IA-32 (instruções IN e OUT)

Exemplo: Relógio de tempo real



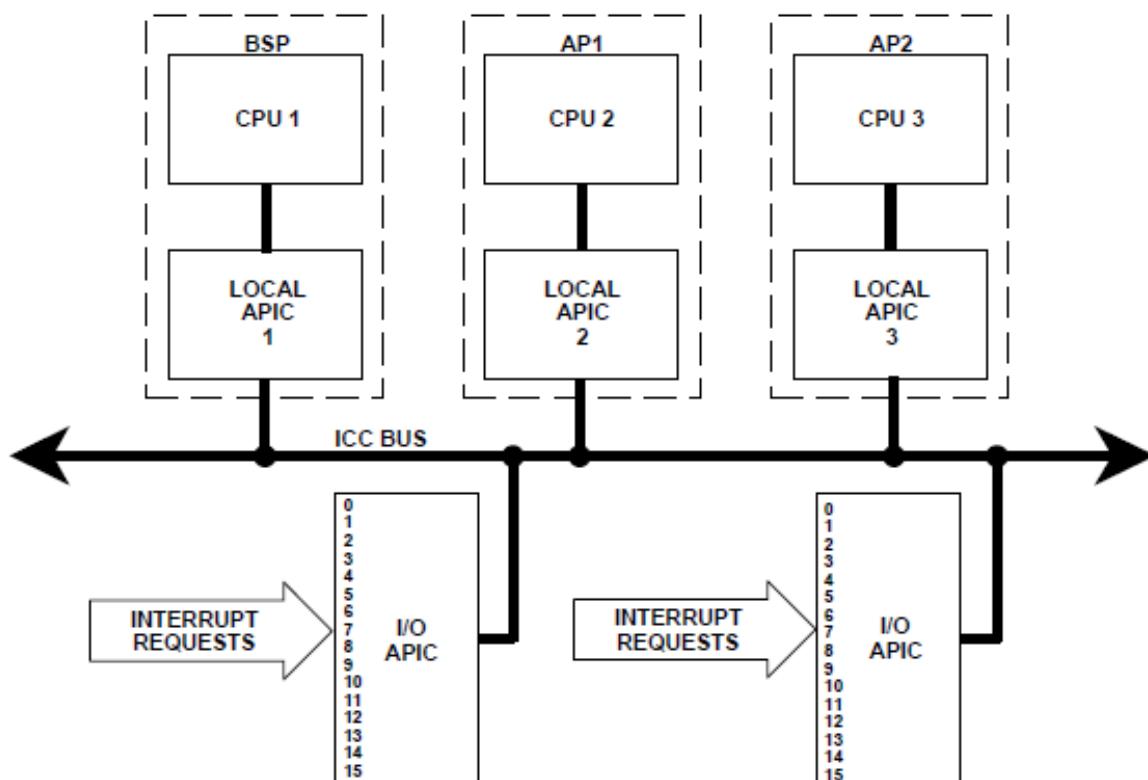
Técnica 1: Varriamento (polling)

- Repetir periodicamente:
 - ① Consultar registos de estado
 - ② Dispositivo pronto? Realizar operação (leitura/escrita)
 - ③ Dispositivo com erro? Recuperar
 - Técnica comum em sistemas pequenos ou de baixo desempenho
 - Comportamento temporal previsível
 - Baixo custo em *hardware*
 - Simples
- ➡ Em sistemas de elevado desempenho: desperdiça recursos de CPU
- ➡ Pode ser apropriado para periféricos lentos:
número de varrimentos necessários é pequeno

Técnica 2: Interrupções

- Dispositivo gera interrupção quando:
 - está pronto
 - ocorre um erro
- Atendimento de interrupção pelo CPU:
 - Interrompe execução do programa
 - Salta para rotina de atendimento
 - Quando a rotina termina, retoma execução “regular”
- CPU suporta diferentes interrupções
- Interrupções são hierarquizadas (prioridades)
 - Dispositivos de atendimento urgente usam interrupções de maior prioridade
 - Atendimento de interrupções de maior prioridade pode interromper atendimento de outras de menor prioridade
- Prioridades são, geralmente, fixas (definidas por hardware)
- Sistema inclui hardware específico para gerir as interrupções
 - Intel 8259 (PCs antigos), circuito integrado ou incluído em “southbridge”
 - APIC (Advanced Programmable Interrupt Controller)

Exemplo: Sistema com APIC



Técnica 3: Acesso directo a memória (DMA)

- Problema: Com as duas técnicas anteriores, a transferência de dados fica a cargo do CPU
 - O CPU é que transfere a informação dos registos de dados do periférico para memória e vice-versa
 - Com dispositivos de alta velocidade (discos, placas de rede), esta abordagem pode afetar gravemente o desempenho das outras tarefas
- Solução: usar um controlador para gerir as transferências
 - sistema operativo define zona de memória (endereço e tamanho) (configuração do controlador de DMA)
 - controlador realiza a transferência, enquanto CPU continua com as suas tarefas
 - quando transferência termina ou encontra um erro, controlador notifica CPU (interrupção)
- Fatores a ter em conta:
 - Interação com memória *cache*: DMA altera memória principal, cópia em *cache* fica desatualizada!
 - Interação com memória virtual: endereços virtuais consecutivos podem não corresponder a endereços físicos consecutivos!

Medidas de desempenho

- Medidas de desempenho de E/S dependem de:
 - Hardware: CPU, memória, controladores, barramentos
 - Software: sistema operativo, sistema de gestão de base de dados, aplicação
 - Carga: taxas e padrões de pedidos
- Projeto pode beneficiar tempo de resposta ou débito
- Frequentemente: medir débito com restrições do tempo de resposta (o tempo de resposta não pode exceder um limiar definido)
- Vários tipos de *benchmarks*
 - Bases de dados: Transaction Processing Council
<http://www.tpc.org>
 - Sistema de ficheiros: SPEC File System (SFS)
<http://www.spec.org/sfs2008>
 - Servidores Web: SPEC Web
<http://www.spec.org/web2009>

➡ Amdahl ataca de novo...

- Desempenho E/S pode comprometer ganhos obtidos por aumento do paralelismo.
- Exemplo:
 - Tarefa demora 90 s (de tempo de CPU) mais 10 s de tempo de E/S
 - Se o número de CPUs duplicar cada dois anos, como evolui o tempo total, admitindo que o sistema E/S não é alterado?

Year	CPU time	I/O time	Elapsed time	% I/O time
now	90s	10s	100s	10%
+2	45s	10s	55s	18%
+4	23s	10s	33s	31%
+6	11s	10s	21s	47%

Fonte: [COD4]

1 Aspetos gerais

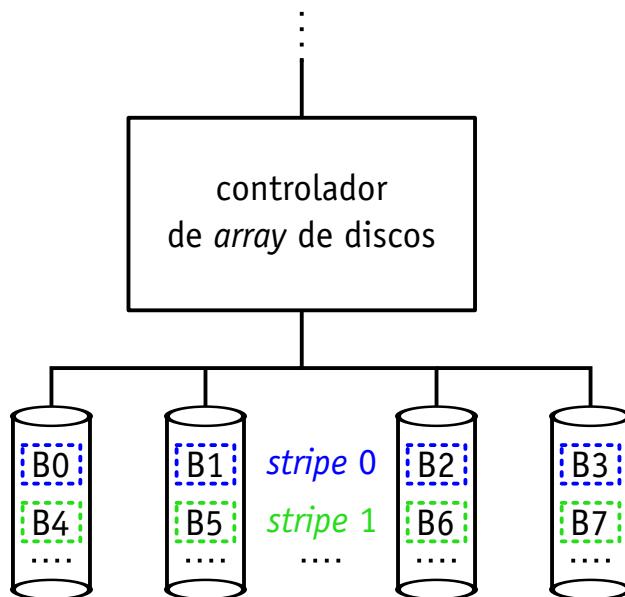
2 Armazenamento de informação

3 Comunicação com periféricos

4 Gestão de periféricos

5 Sistemas RAID

Conjuntos de discos



- Discos físicos em paralelo formam um disco virtual
- Blocos (B_0, \dots) de um ficheiro são "espalhados" pelos discos físicos
- 1 bloco = 1 ou mais setores (número fixo)
- Blocos correspondentes de cada disco formam uma "banda" (*stripe*)

Características de conjuntos de discos

→ Vantagens:

- Melhor desempenho
 - Transferências "grandes": (array com bandas)
Acesso a D blocos de uma banda demora tanto como acesso a um só disco
 - Transferências "pequenas":
Podem efetuar-se D acessos independentes
- Menor custo
 - Discos individuais mais baratos (grande volume de vendas)
 - Controladores sofisticados em circuito integrado
Existem controladores capazes de lidar com 1000 discos individuais

→ Desvantagem: Menor fiabilidade

- Basta falhar um disco físico para levar à falha do conjunto
- Para um conjunto de D discos, cada um com o mesmo MTTF:

$$MTTF_{array} = \frac{MTTF}{D}$$

→ Solução: Introduzir redundância para recuperar de avaria de um disco

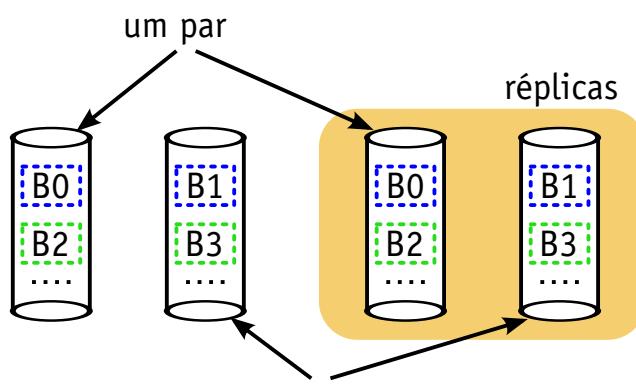
- RAID = *Redundant Array of Inexpensive Disks*
- Paralelismo aumenta a fiabilidade
É mais económico (para o mesmo nível de fiabilidade) ter redundância que aumentar a fiabilidade de um único disco
 - Aproveita economia de escala: existe uma mercado muito grande para discos baratos
- Aumenta MTBF, especialmente se for possível trocar discos sem parar o sistema (*hot swap*)
- Paralelismo aumenta o desempenho
 - Um ficheiro pode ser espalhado por vários discos
 - Leituras “paralelas” permitem obter vários blocos simultaneamente
- Desvantagem: mais discos que o estritamente necessário (aumenta do consumo de energia)
 - Exceção: RAID nível 0 corresponde a uma organização sem redundância

Impacto da introdução de redundância

- D : nº total de discos de dados
- G : nº de discos de dados por grupo
- C : nº de discos com informação de verificação por grupo
- $n_G = D/G$: nº de grupos de proteção
- $N = n_G \times (G + C)$
- falhas independentes; taxa de falhas $t_f = 1/\text{MTTF} = \text{constante}$
- MTTR: o tempo médio de reparação (substituição de um disco)

➡ Array falha se ocorrer pelo menos uma *outra* falha num grupo enquanto a *primeira falha é reparada*.

$$\text{MTTF}_{\text{RAID}} = \frac{\text{MTTF}_{\text{disco}}^2}{(D + C \times n_G) \times (G + C - 1) \times \text{MTTR}}$$



➡ RAID 1: espelho (duplicação concorrente de cada disco)

- Discos: $G = 1$, $C = 1$ (cada disco de dados é duplicado)
- Escrita simultânea no disco de dados e no “disco-espelho”
- Falha de disco: ler do espelho
- Pode ler blocos diferentes de cada disco
- Aproveitamento da capacidade dos discos de 50 %

Cálculo de paridade

- Paridade de D bits (o símbolo \oplus representa a operação “ou-exclusivo”):

$$\text{Paridade}(b_1, b_2, \dots, b_D) = b_1 \oplus b_2 \oplus \dots \oplus b_D = p$$

- Se número de $b_i = 1$ é par \rightarrow paridade = 0; senão paridade = 1
- Usar paridade para determinar um dos b_i em falta:

- 1 Calcular a paridade q dos b_i disponíveis
- 2 Determinar o valor em falta calculando $q \oplus p$

- (Exemplo) Para determinar b_1 calcula-se sucessivamente:

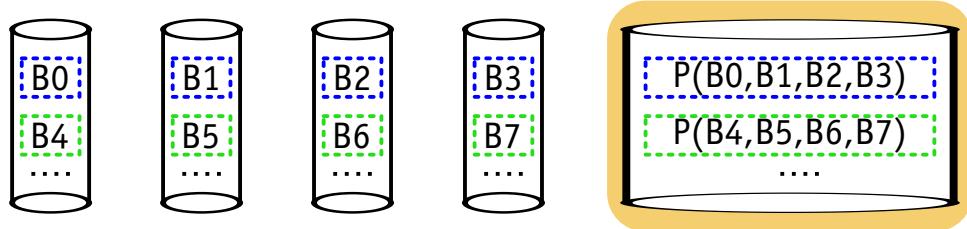
- 1 $q = \text{Paridade}(b_2, \dots, b_D)$
- 2 $b_1 = p \oplus q$

- Nota: $a \oplus (b \oplus c) = (a \oplus b) \oplus c = a \oplus b \oplus c$
 $a \oplus b = b \oplus a$

- Para elementos com vários bits, a paridade é calculada fazendo o “ou-exclusivo” dos bits em posições correspondentes:

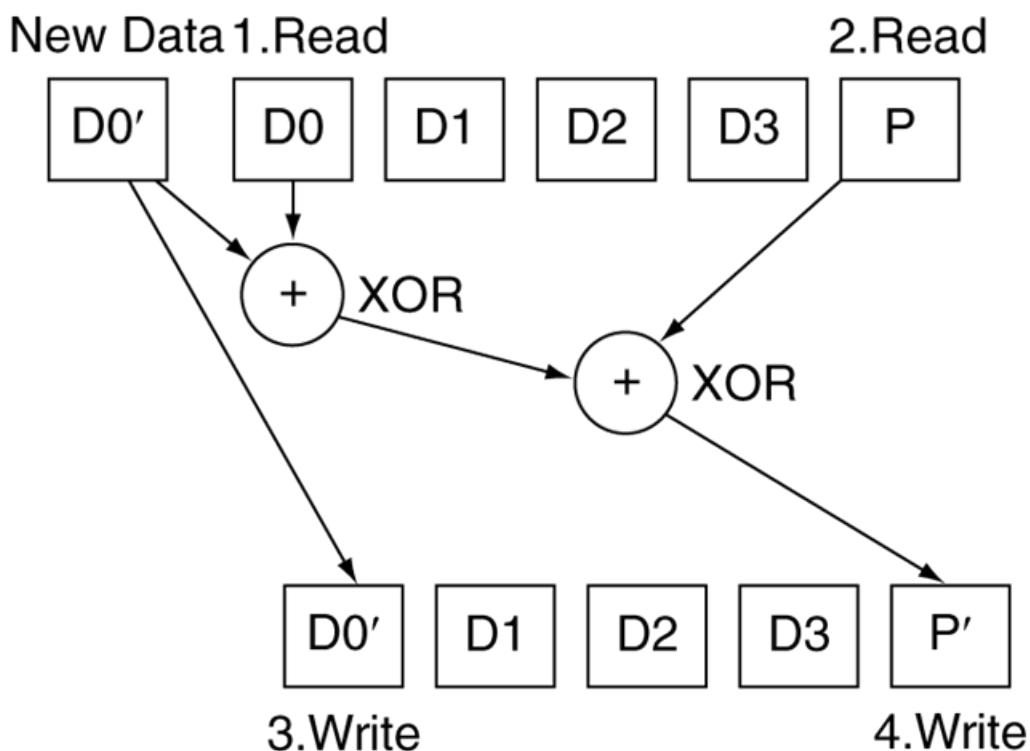
Se $X = [x_1, x_2, \dots, x_n]$ e $Y = [y_1, y_2, \dots, y_n]$, então
 $\text{Paridade}(X, Y) = [x_1 \oplus y_1, x_2 \oplus y_2, \dots, x_n \oplus y_n]$

RAID 4: Block-Interleaved Parity



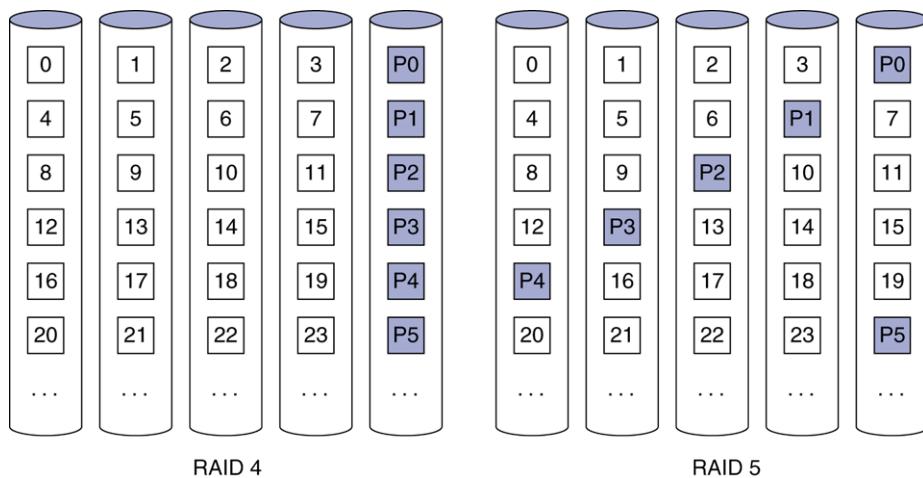
- RAID 4 tem C=1 (1 disco de paridade por grupo)
- Dados espalhados pelos discos ao nível do bloco
- Disco de verificação armazena paridade de blocos da mesma banda
- Leitura: acesso apenas ao disco que contém o bloco
- Escrita: alterar *um* disco de dados e atualizar o disco de paridade (ciclo leitura-modificação-escrita).
- Pouco usado: o acesso ao disco de paridade limita o desempenho na escrita.

RAID 4: leitura-alteração-escrita



RAID 5: Paridade distribuída por vários discos

- Dados espalhados pelos discos ao nível do bloco
- Semelhante a RAID 4, mas com blocos de paridade distribuídos por todos os discos do grupo
- Evita que o disco de paridade restrinja o desempenho do sistema (como acontece com RAID 4)
- Muito usado.



Fonte: [COD4]

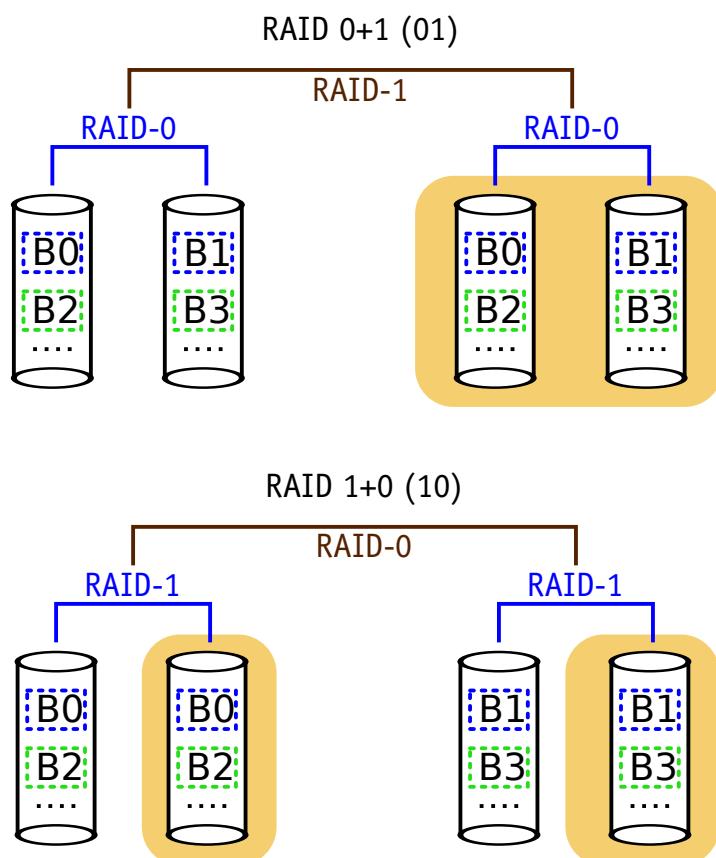
Outros tipos de RAID

➡ RAID 6: Redundância adicional

- Como RAID 5, mas com dois grupos de informação de redundância ($C=2$)
 - Paridade calculada para grupos formados de duas maneiras diferentes.
 - Um dos métodos é igual ao de RAID-5.
- Pode recuperar de situação com dois discos avariados por grupo
- Maior tolerância a falhas à custa de mais redundância

➡ RAID composto

- Composição de duas técnicas de RAID
- RAID 10 (1+0): Aplicar RAID 0 a conjuntos RAID 1 (em vez de discos individuais)
- RAID 01 (0+1): Aplicar RAID 1 a conjuntos RAID 0
- Mais variantes: RAID 50, RAID 60



RAID: conclusão

- ➡ Solução económica para necessidades de armazenamento significativas
É mais barato usar vários discos de fiabilidade normal que um disco de fiabilidade muito elevada.
- ➡ RAID aumenta disponibilidade (exceto RAID 0)
- ➡ Também pode aumentar o desempenho
É preciso ter em atenção o domínio de aplicação, porque o desempenho depende das características dos acessos
- ➡ Elevada disponibilidade requer trocas em funcionamento para reduzir o tempo médio de reparação (MTTR)
- ➡ Atenção: Assume-se que falhas de discos são independentes!
RAID não resolve situações em que vários discos são afetados simultaneamente pelo mesmo evento (p.ex. falha de fonte de alimentação).

Referências

- COD4** D. A. Patterson & J. L. Hennessey, Computer Organization and Design, 4 ed.
- CA5** D. A. Patterson & J. L. Hennessey, Computer Architecture: A Quantitative Approach, 5 ed.

Os tópicos tratados nesta apresentação são abordados nas seguintes secções de [COD4]:

- 6.1–6.7