

[Aula 17] Entrada e saída

Prof. João F. Mari
joaof.mari@ufv.br

Aula 17 - Entrada e Saída

Introdução

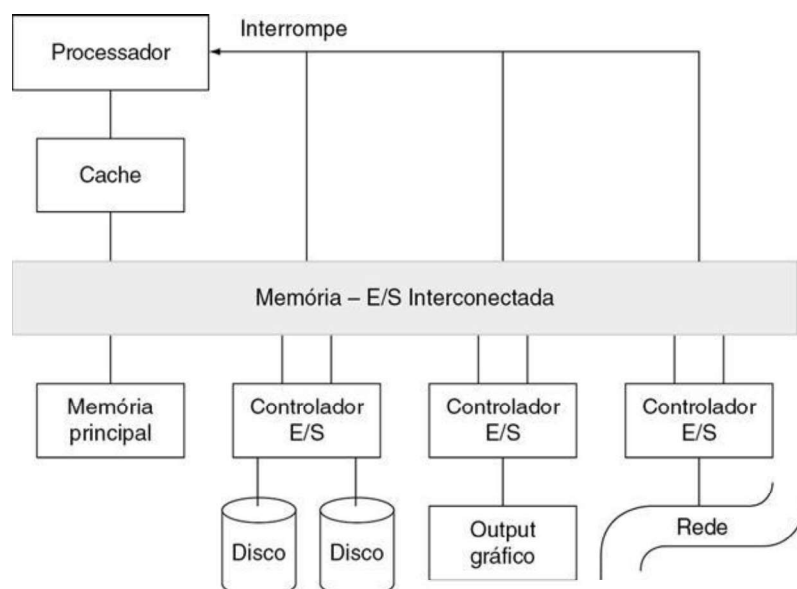
- Ambiente externo ao processador bastante diverso
 - Composto por vários dispositivos adicionais que servem para a entrada e saída (E/S) de dados
- Dispositivos de I/O são diversos em relação a:
 - Comportamento – entrada, saída ou armazenamento
 - Taxa de transferência de dados (*throughput*) – a taxa na qual os dados podem ser transferidos entre o dispositivo de E/S e memória ou processador.
- São medidas de desempenho para um sistema de entrada e saída:
 - Desempenho
 - Capacidade de expansão
 - Necessidade
 - Custo, tamanho, peso

Medidas de desempenho

- Largura de Banda (*throughput*)
 - Quantidade de informação que pode ser trocada por unidade de tempo.
 - Qual é quantidade de dados que pode movimentar-se pelo sistema em um certo intervalo de tempo?
 - Quantas operações de E/S podem ser realizadas por unidade de tempo?
- Tempo de E/S ou latência:
 - O tempo para realizar uma operação de E/S.
- Sistemas desktop e embarcados:
 - Tempo de resposta e custo de E/S
- Outros sistemas:
 - Vazão e facilidade de expansão

Coleção típica de dispositivos de E/S

- **Estrutura de um sistema simples com sua E/S.**
 - As conexões entre os dispositivos de E/S, processador e memória são chamadas de barramentos.
 - A comunicação entre os dispositivos e o processador utiliza interrupções e protocolos no barramento.



A diversidade de dispositivos de E/S

- Dispositivos de entrada e saída:
- Servem como dispositivos de entrada, saída ou armazenamento.
- O parceiro de comunicação (pessoas ou outros computadores)
- Taxa de dados. Variam em oito ordens de grandeza.

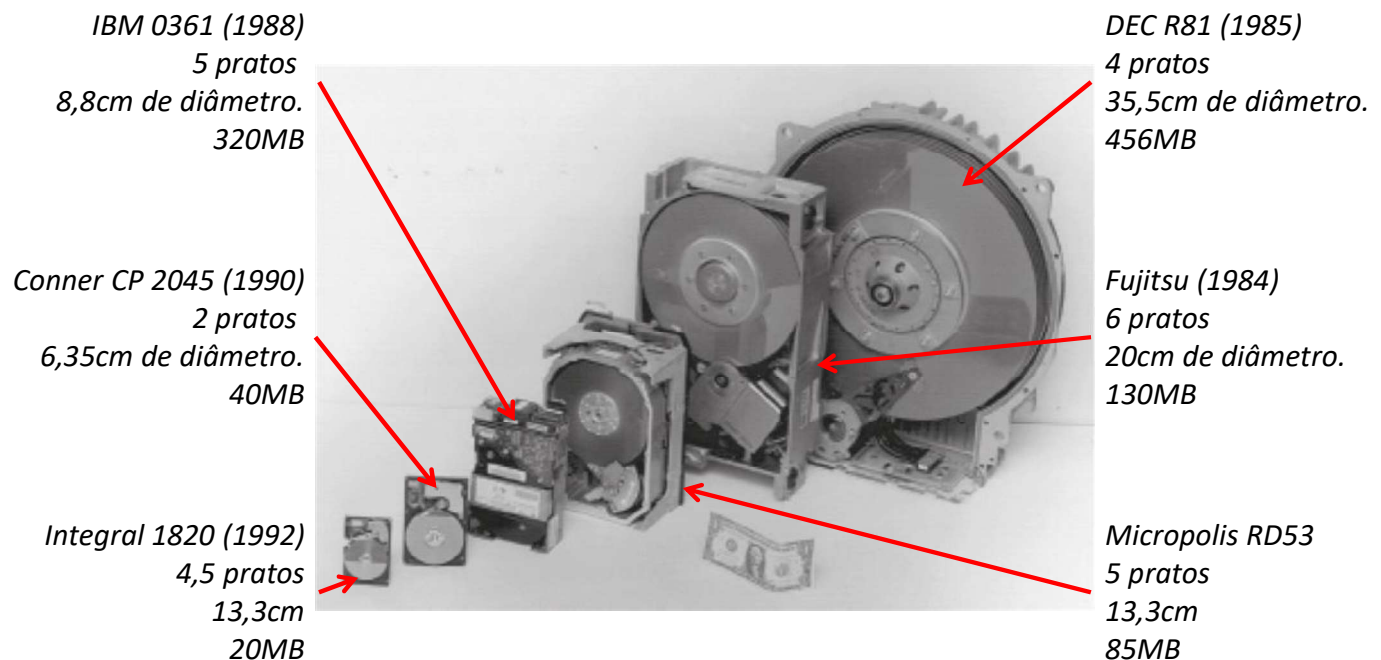
Dispositivo	Comportamento	Parceiro	Taxa de dados (Mbits/seg)
Teclado	Entrada	humano	0,0001
Mouse	Entrada	humano	0,0038
Entrada de voz	entrada	humano	0,2640
Entrada de som	entrada	máquina	3,0000
Scanner	entrada	humano	3,2000
Saída de voz	saída	humano	0,2640
Saída de som	saída	humano	8,0000
Impressora a laser	saída	humano	3,2000
Monitor gráfico	saída	humano	800,0000-8000,0000
Modem a cabo	Entrada ou saída	máquina	0,1280-6,0000
Rede/LAN	Entrada ou saída	máquina	100,0000-10000,0000
Rede/LAN sem fio	Entrada ou saída	máquina	11,0000-54,0000
Disco óptico	Armazenamento	máquina	80,0000-220,0000
Fita magnética	Armazenamento	máquina	5,0000-120,0000
Memória flash	Armazenamento	máquina	32,0000-200,0000
Disco magnético	Armazenamento	máquina	800,0000-3000,0000

Armazenamento em disco e confiabilidade

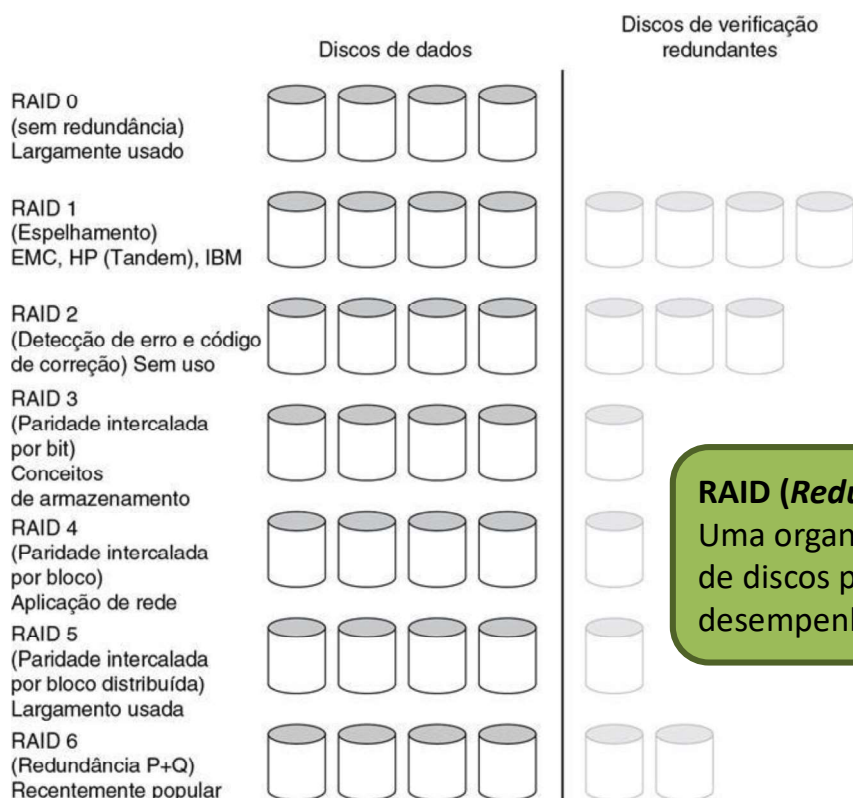
- Discos magnéticos:
 - Compostos por pratos giratórios cobertos com superfície magnética e armazenamento não volátil
 - 5000 a 15000 RPMs
 - Superfícies divididas em trilhas e setores
 - Cilindros: todas as trilhas sob a cabeça de leitura em um instante de tempo
- Acesso em 3 estágios:
 - Posicionamento da cabeça de leitura (seek):
 - Tempo de seek (3 a 15ms), dependente da localidade.
 - Setor desejada encontre a cabeça de leitura: latência rotacional
 - Tempo de transferência: tempo para transferir um bloco de bits.

Aumento da densidade e redução do tamanho físico dos discos.

- Seis discos magnéticos, variando em diâmetro de 35cm até 4,5cm.



RAID - *Redundant arrays of inexpensive disks*

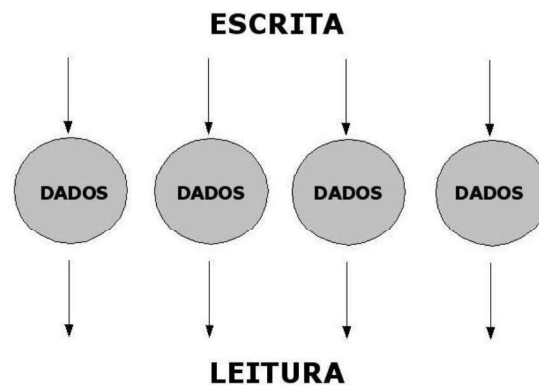


RAID (*Redundant array of inexpensive disks*):
Uma organização de discos que usa um arranjo de discos pequenos e baratos para aumentar o desempenho e a confiabilidade.

RAID - *Redundant arrays of inexpensive disks*

- **Nenhuma redundância (RAID 0):**

- *Striping*: espalhamento de blocos através de múltiplos discos
- Vários blocos podem ser acessados em paralelo aumentando o desempenho
- Mesmo custo que um disco “grande”
- Sem redundância – a falha de um ou mais discos é mais provável na medida que aumenta o número de discos

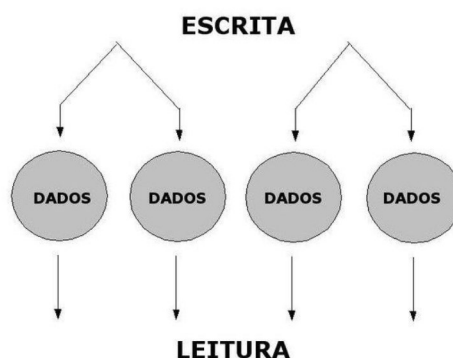


RAID-0: Leitura e escrita ocorrem ao mesmo tempo em cada unidade.

RAID - *Redundant arrays of inexpensive disks*

- **Espalhamento (RAID 1):**

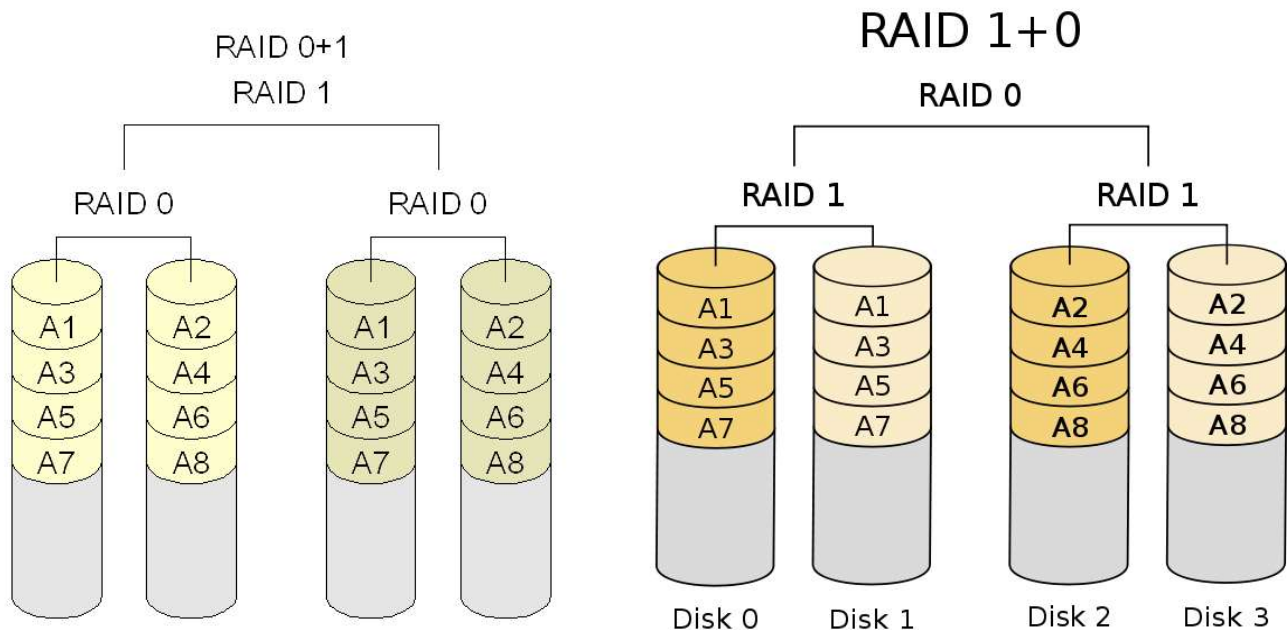
- Espelhamento (*shadowing*): esquema tradicional para tolerar falhas de disco.
- Utiliza o dobro da quantidade de discos do RAID 0.
- Sempre que os dados são gravados em um disco, esses são gravados em um disco redundante.
- Sempre existem duas cópias da informação.
- Se um disco falhar, o sistema obtém o seu conteúdo no disco “espelho”.



RAID-1: A escrita é feita em pares de unidades enquanto a leitura ocorre em todas as unidades ao mesmo tempo.

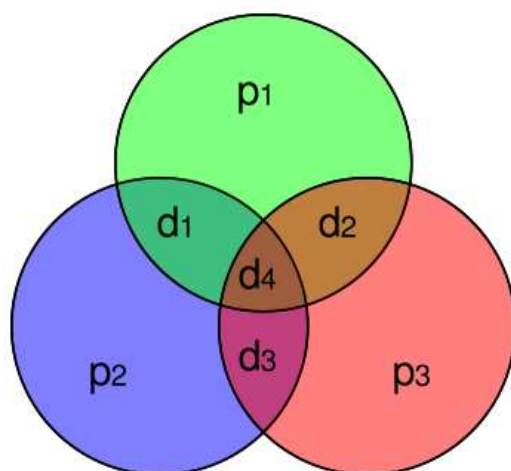
RAID - *Redundant arrays of inexpensive disks*

- **RAID 01 (0 + 1) e RAID 10 (1 + 0):**
 - Redundância + espalhamento;
 - Mínimo de 4 discos.



RAID - *Redundant arrays of inexpensive disks*

- **Código de detecção e correção de erros (RAID 2)**
 - Técnica de acesso paralelo: todos os discos participam de qualquer requisição de E/S
 - Utiliza um esquema de detecção de erros e correção que é mais utilizado em memórias:
 - Código de Hamming.
 - Necessita de vários discos
 - Caiu em desuso e não é muito utilizado na prática.



- d1, d2, d3, d4
- 1, 1, 1, 0
- $p1 = d1 \text{ xor } d2 \text{ xor } d4$
- $p1 = 1 \text{ xor } 1 \text{ xor } 0 = 0$
- $p2 = d1 \text{ xor } d3 \text{ xor } d4$
- $p2 = 1 \text{ xor } 1 \text{ xor } 0 = 0$
- $p3 = d2 \text{ xor } d3 \text{ xor } d4$
- $p3 = 1 \text{ xor } 1 \text{ xor } 0 = 0$

RAID - *Redundant arrays of inexpensive disks*

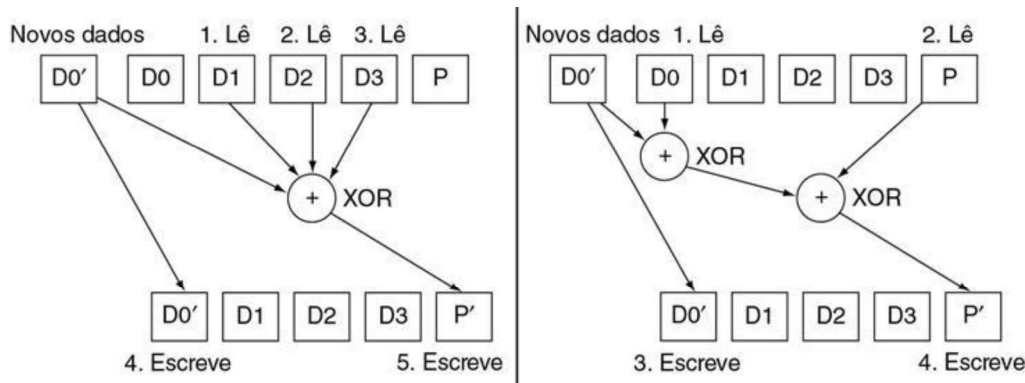
- Paridade intercalada por bit (RAID 3):
 - Paridade:
 - Podemos pensar no disco redundante como aquele com a soma de todos os dados dos outros discos.
 - Quando um disco falha, subtrai-se os dados dos discos bons do disco de paridade.
 - A informação restante é a que esta faltando.
 - Muitos discos precisam ser lidos para determinar os dados que faltam.

RAID - *Redundant arrays of inexpensive disks*

- Paridade intercalada por bloco (RAID 4):
 - Usa a mesma razão de discos de dados e discos de verificação do RAID 3.
 - A paridade é armazenada como blocos e associada a um conjunto de blocos.

RAID - *Redundant arrays of inexpensive disks*

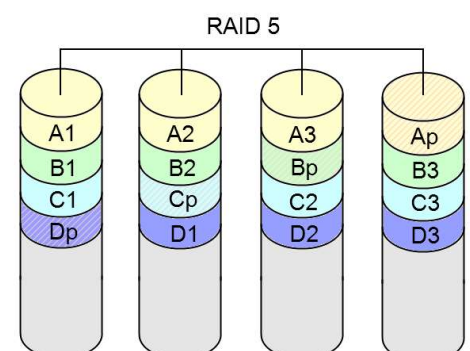
- Pequena atualização de escrita em RAID 3 e RAID 4.



- A otimização para pequenas escritas reduz a quantidade de acesso ao disco, assim como a quantidade de discos.
- No exemplo temos quatro blocos de dados (D0, D1, D2 e D3) e um bloco de paridade (P).
- Para gravar os novos dados D0' (provenientes da CPU) em D0, o RAID 3 lê os blocos D1, D2 e D3 para calcular a nova paridade P0'.
- O RAID 4 lê o valor antigo D0 e compara com o novo valor D0 para ver quais bits mudarão. Depois, lê a paridade antiga P e muda os bits correspondentes para formar P'.

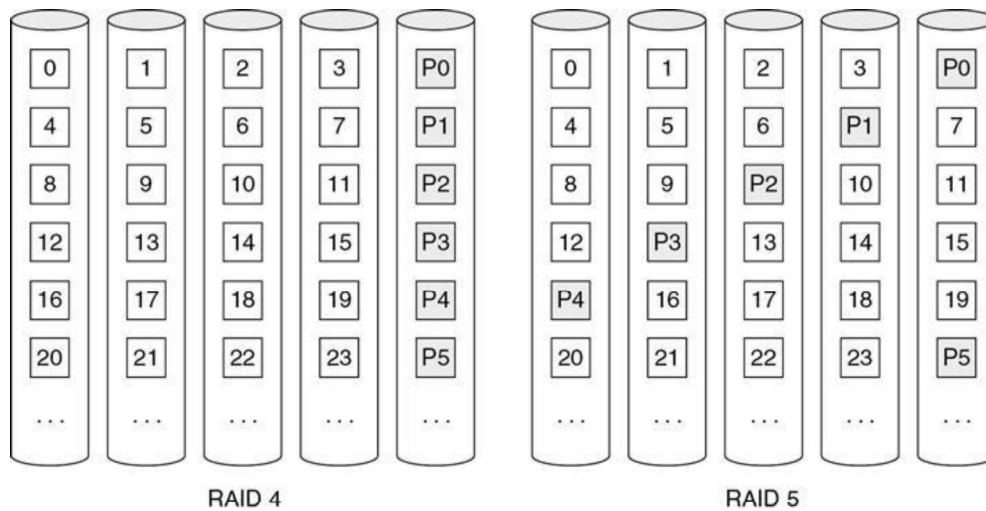
RAID - *Redundant arrays of inexpensive disks*

- Paridade distribuída intercalada por bloco (RAID 5):
 - O RAID 4 executa bem leituras grandes e pequenas e escritas grandes e pequenas.
 - A desvantagem é que o disco de paridade precisa ser atualizado a cada escrita.
 - O RAID 5 distribui os blocos de paridade entre os discos.



RAID - *Redundant arrays of inexpensive disks*

- Paridade intercalar por bloco (RAID 4) versus paridade distribuída intercalada por bloco (RAID 5).

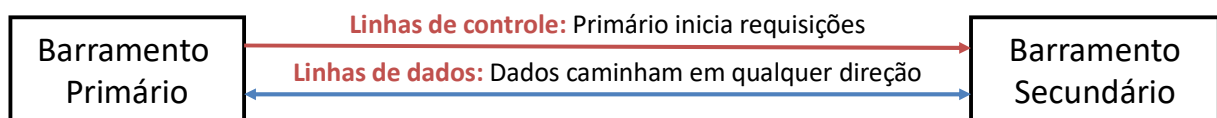


BARRAMENTOS

Barramentos

- Barramento:
 - Linha de comunicação compartilhada;
 - Um conjunto de fios usados para conectar múltiplos dispositivos.
 - Precisa dar suporte a uma grande variedade de dispositivos;
 - Com latências e taxas de transferência de dados variados.
 - VANTAGENS:
 - Versatilidade: novos dispositivos podem ser facilmente inseridos e movidos para outros sistemas computacionais com o mesmo padrão de barramento.
 - DESVANTAGENS:
 - Gargalo na comunicação: o limite da largura do barramento limita a vazão máxima de E/S.
 - A velocidade máxima do barramento é limitado pelo:
 - Comprimento do barramento
 - Numero de dispositivos conectados ao barramento

Fundamentos sobre barramentos



- Conjunto de linhas de controle:
 - Usadas para sinalizar solicitações e confirmações;
 - Indicar que tipo de informação se encontra nas linhas de dados.
- Conjunto de linhas de dados:
 - Transportam informações entre a origem e o destino.
 - Essas informações podem ser dados, endereços ou comandos complexos.
- Transações de barramentos:
 - Requisição: barramento primário (*) solicita o comando (e endereço)
 - Ação: barramento secundário (*) recebe (ou envia) os dados
- Transações podem ser:
 - Entrada: dados caminham do dispositivo de E/S para a memória.
 - Saída: dados caminham da memória para o dispositivo de E/S.
- (*) As denominações mestre/escravo estão sendo substituídas por outras. Nesse caso optou-se por primário e secundário.

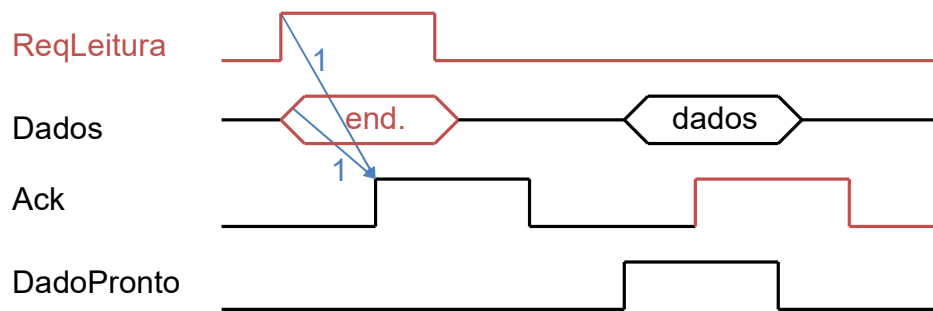
Fundamentos sobre barramentos

- Barramento processador-memória
 - Tecnologia proprietária
 - Curtos e de alta velocidade
 - Velocidade correspondente ao sistema de memória
 - Maximizar a largura de banda memória-processador.
 - Otimizado para transferência de blocos de cache
- **Barramentos de E/S**
 - SCSI, USB, Firewire, ...
 - Extensos e mais lentos
 - Podem ter muitos tipos de dispositivos conectados.
 - Conectado ao barramento processador-memória ou ao barramento *backplane*.
- **Barramento *backplane***
 - ATA, PCI, PCIExpress, ...
 - Usado como barramento intermediário entre os barramentos de E/S e o barramento processador-memória.

Fundamentos sobre barramentos

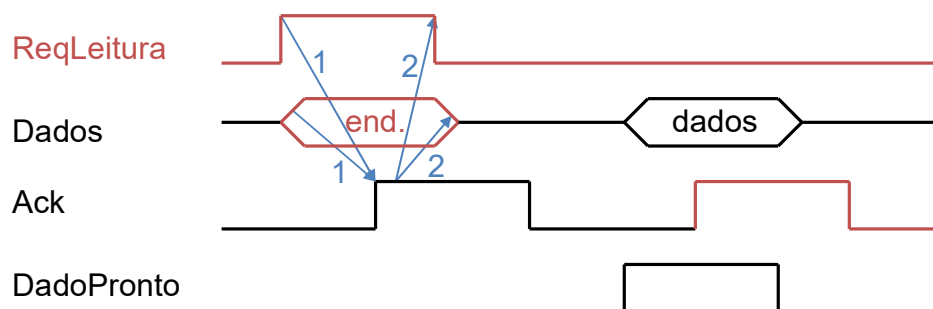
- **Barramentos síncronos:**
 - **[EX]** Barramentos processador-memória
 - Inclui um clock nas linhas de controle
 - Possui um protocolo fixo para comunicação relacionado com o clock.
 - VANTAGENS:
 - É rápido e envolve um circuito lógico simples (pequena máquina de estados finitos).
 - DESVANTAGENS:
 - Todos os dispositivos que utilizam o barramentos precisam utilizar a mesma taxa de clock.
 - Os barramentos não podem ser muito longos
 - Evitar problemas de variação do clock.
- **Barramentos assíncronos:**
 - **[EX]** Barramentos de Entrada/Saída
 - Não possui clock.
 - Implementa um protocolo de aperto de mãos (*handshaking*) e linhas de controle adicionais:
 - ReqLeitura, Ack e DadoPronto.
 - VANTAGENS:
 - Pode acomodar uma grande variedade de dispositivos
 - Pode ser estendido sem preocupação com variações do clock ou sincronismo.
 - DESVANTAGENS:
 - Mais lento do que o barramento síncrono.

Barramentos assíncronos



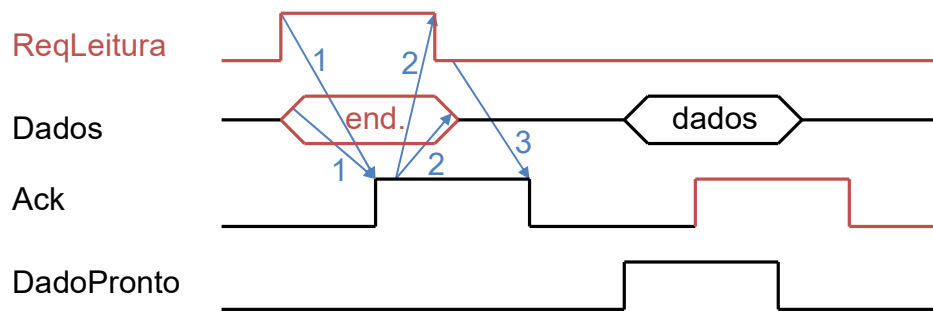
- O dispositivo sinaliza uma requisição levantando **ReqLeitura** e colocando o endereço nas linhas de dados.
 - 1. Memória vê **ReqLeitura**, lê o endereço em Dados e levanta **Ack** para indicar que ele foi visto.
 - 2. Dispositivo de E/S vê **Ack** alto e liberta **ReqLeitura** e Dados.
 - 3. Memória vê **ReqLeitura** baixa e abaixa a linha **Ack** para confirmar **ReqLeitura**
 - 4. Quando a memória possui os dados prontos, ela os coloca em **Dados** e levanta **DadoPronto**
 - 5. O dispositivo de E/S vê **DadoPronto**, lê os dados da linha **Dados** e levanta **Ack**.
 - 6. A memória vê **Ack**, abaixa **DadoPronto** e libera as linhas de dados.
 - 7. O dispositivo de E/S vê **DadoPronto** baixar, abaixa a linha **Ack**, indicando que a transmissão está concluída.

Barramentos assíncronos



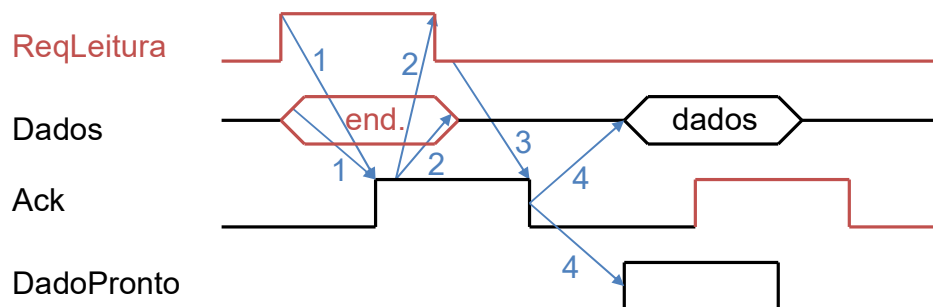
- O dispositivo sinaliza uma requisição levantando **ReqLeitura** e colocando o endereço nas linhas de dados.
 - 1. Memória vê **ReqLeitura**, lê o endereço em Dados e levanta **Ack** para indicar que ele foi visto.
 - 2. Dispositivo de E/S vê **Ack** alto e liberta **ReqLeitura** e Dados.
 - 3. Memória vê **ReqLeitura** baixa e abaixa a linha **Ack** para confirmar **ReqLeitura**
 - 4. Quando a memória possui os dados prontos, ela os coloca em **Dados** e levanta **DadoPronto**
 - 5. O dispositivo de E/S vê **DadoPronto**, lê os dados da linha **Dados** e levanta **Ack**.
 - 6. A memória vê **Ack**, abaixa **DadoPronto** e libera as linhas de dados.
 - 7. O dispositivo de E/S vê **DadoPronto** baixar, abaixa a linha **Ack**, indicando que a transmissão está concluída.

Barramentos assíncronos



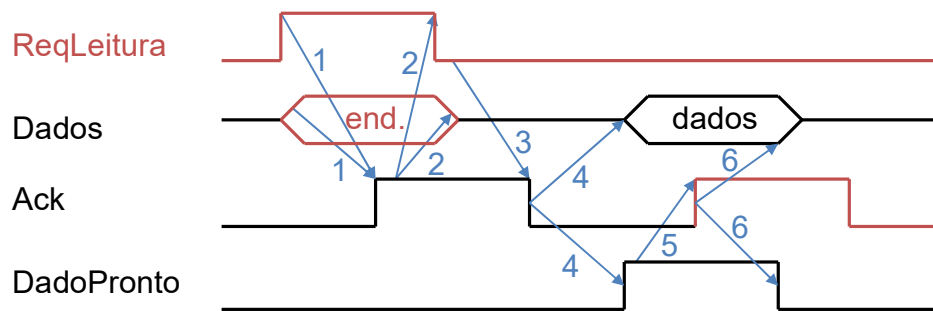
- O dispositivo sinaliza uma requisição levantando **ReqLeitura** e colocando o endereço nas linhas de dados.
 - 1. Memória vê **ReqLeitura**, lê o endereço em Dados e levanta **Ack** para indicar que ele foi visto.
 - 2. Dispositivo de E/S vê **Ack** alto e liberta **ReqLeitura** e Dados.
 - 3. Memória vê **ReqLeitura** baixa e abaixa a linha **Ack** para confirmar **ReqLeitura**
 - 4. Quando a memória possui os dados prontos, ela os coloca em **Dados** e levanta **DadoPronto**
 - 5. O dispositivo de E/S vê **DadoPronto**, lê os dados da linha **Dados** e levanta **Ack**.
 - 6. A memória vê **Ack**, abaixa **DadoPronto** e libera as linhas de dados.
 - 7. O dispositivo de E/S vê **DadoPronto** baixar, abaixa a linha **Ack**, indicando que a transmissão está concluída.

Barramentos assíncronos



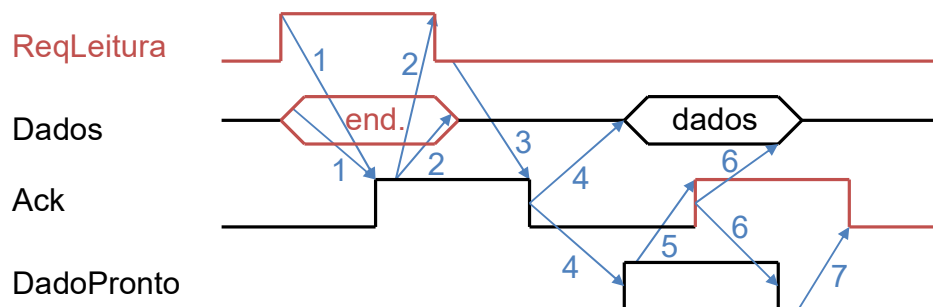
- O dispositivo sinaliza uma requisição levantando **ReqLeitura** e colocando o endereço nas linhas de dados.
 - 1. Memória vê **ReqLeitura**, lê o endereço em Dados e levanta **Ack** para indicar que ele foi visto.
 - 2. Dispositivo de E/S vê **Ack** alto e liberta **ReqLeitura** e Dados.
 - 3. Memória vê **ReqLeitura** baixa e abaixa a linha **Ack** para confirmar **ReqLeitura**
 - 4. Quando a memória possui os dados prontos, ela os coloca em **Dados** e levanta **DadoPronto**
 - 5. O dispositivo de E/S vê **DadoPronto**, lê os dados da linha **Dados** e levanta **Ack**.
 - 6. A memória vê **Ack**, abaixa **DadoPronto** e libera as linhas de dados.
 - 7. O dispositivo de E/S vê **DadoPronto** baixar, abaixa a linha **Ack**, indicando que a transmissão está concluída.

Barramentos assíncronos



- O dispositivo sinaliza uma requisição levantando **ReqLeitura** e colocando o endereço nas linhas de dados.
 - 1. Memória vê **ReqLeitura**, lê o endereço em Dados e levanta **Ack** para indicar que ele foi visto.
 - 2. Dispositivo de E/S vê **Ack** alto e liberta **ReqLeitura** e Dados.
 - 3. Memória vê **ReqLeitura** baixa e abaixa a linha **Ack** para confirmar **ReqLeitura**
 - 4. Quando a memória possui os dados prontos, ela os coloca em **Dados** e levanta **DadoPronto**
 - 5. O dispositivo de E/S vê **DadoPronto**, lê os dados da linha **Dados** e levanta **Ack**.
 - 6. A memória vê **Ack**, abaixa **DadoPronto** e libera as linhas de dados.
 - 7. O dispositivo de E/S vê **DadoPronto** baixar, abaixa a linha **Ack**, indicando que a transmissão está concluída.

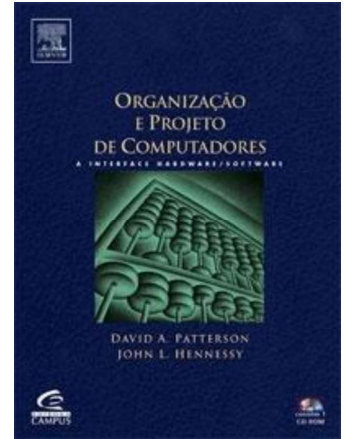
Barramentos assíncronos



- O dispositivo sinaliza uma requisição levantando **ReqLeitura** e colocando o endereço nas linhas de dados.
 - 1. Memória vê **ReqLeitura**, lê o endereço em Dados e levanta **Ack** para indicar que ele foi visto.
 - 2. Dispositivo de E/S vê **Ack** alto e liberta **ReqLeitura** e Dados.
 - 3. Memória vê **ReqLeitura** baixa e abaixa a linha **Ack** para confirmar **ReqLeitura**
 - 4. Quando a memória possui os dados prontos, ela os coloca em **Dados** e levanta **DadoPronto**
 - 5. O dispositivo de E/S vê **DadoPronto**, lê os dados da linha **Dados** e levanta **Ack**.
 - 6. A memória vê **Ack**, abaixa **DadoPronto** e libera as linhas de dados.
 - 7. O dispositivo de E/S vê **DadoPronto** baixar, abaixa a linha **Ack**, indicando que a transmissão está concluída.

Bibliografia

1. PATTERSON, D.A; HENNESSY, J.L. **Organização e Projeto de Computadores: A Interface Hardware/Software**. 3a. Ed. Elsevier, 2005.
 - Capítulo 7.
2. Notas de aula do prof. Luciano J. Senger:
 - <http://www.ljsenger.net/classroom.html>
3. Notas de aula da profa. Mary Jane Irwin
 - CSE 331 Computer Organization and Design
 - <http://www.cse.psu.edu/research/mdl/mji/mjicourses>



FIM

- FIM:
 - **Aula 17 – Entrada e Saída**
- **Fim da disciplina!**