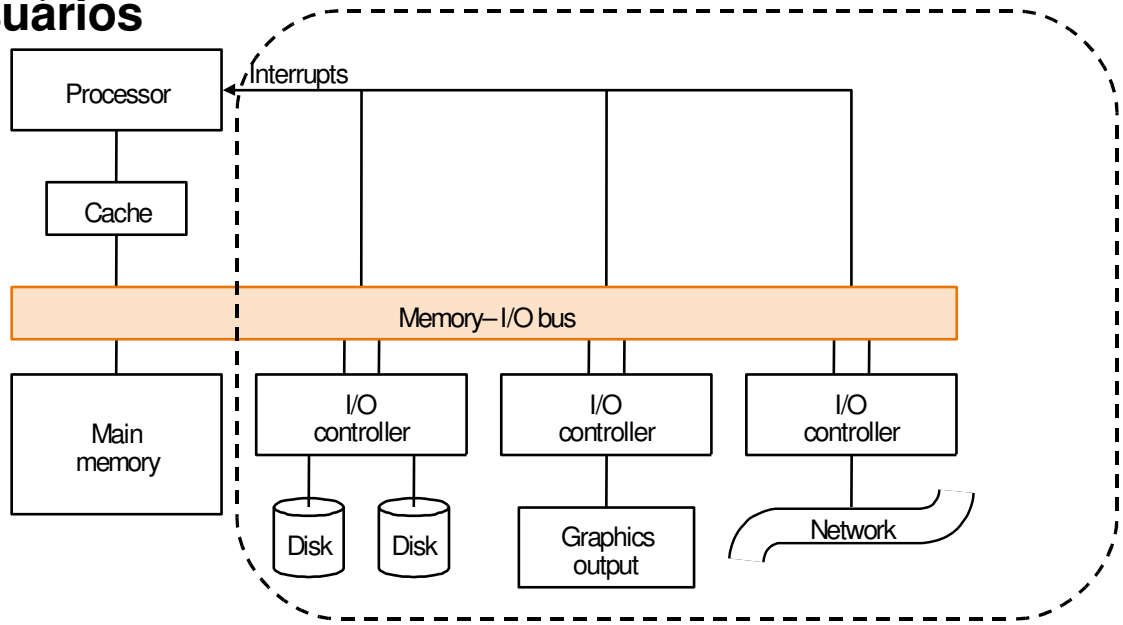
Capítulo 8

Sistemas I/O

©1998 Morgan Kaufmann Publishers

Interface entre Processador e Periféricos

- **Projeto de I/O é afetado por muitos fatores**
- **Desempenho:**
  - **— latência de acesso**
  - **— throughput** (vazão)
  - **— conexão entre dispositivos e o sistema**
  - **— hierarquia de memória**
  - **— sistema operacional**
- **Diferentes usuários**



©1998 Morgan Kaufmann Publishers

I/O

- **É importante mas, geralmente, deixado em segundo plano**

  “The difficulties in assessing and designing I/O systems have often relegated I/O to second class status”

  “courses in every aspect of computing, from programming to computer architecture often ignore I/O or give it scanty coverage”

  “textbooks leave the subject to near the end, making it easier for students and instructors to skip it!”

©1998 Morgan Kaufmann Publishers

Benchmarks para I/O

- **Benchmarks de E/S para Supercomputadores:**
  - **acesso a arquivos grandes em batch**
  - **uma grande operação de leitura seguida de pequenas de escrita (confiabilidade, em caso de crash)**
- **Benchmarks para Processamento de Transações:**
  - **Deve medir tempo de resposta e throughput**
  - **Transações são curtas: I/O rate (acessos / s) mais importante do que data rate (bytes / s)**
  - **I/O rate mais conhecido: TPC-B : simula rede de ATMs**
    - medida (throughput para transações dentro de faixa aceitável de tempo de resposta): TPS: transactions per second
- **Benchmarks para Sistemas de Arquivo:**
  - **em um ambiente de engenharia: 80% acessos a arq < 10K; 67% de operações de leitura e 27% de escrita (6% R/W)**
  - **exemplo: benchmark com 70 arquivos, total de 200 KB (mkdir, cp, make, ScanDir, ReadAll)**

©1998 Morgan Kaufmann Publishers

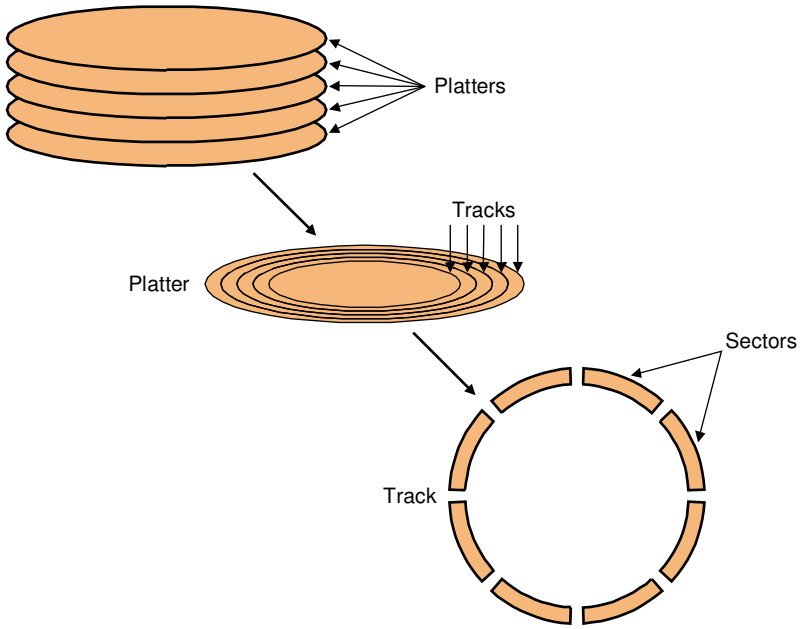
Dispositivos de I/O

- **Diversos dispositivos**
  - **— comportamento (i.e., entrada vs. saída)**
  - **— conexão com (quem está do outro lado?)**
  - **— taxa de dados**

Dispositivo	Comportamento	Conexão com	Taxa de Dados (KB/sec)
Keyboard	input	human	0.01
Mouse	input	human	0.02
Voice input	input	human	0.02
Scanner	input	human	400.00
Voice output	output	human	0.60
Line printer	output	human	1.00
Laser printer	output	human	200.00
Graphics display	output	human	60,000.00
Modem	input or output	machine	2.00-8.00
Network/LAN	input or output	machine	500.00-6000.00
Floppy disk	storage	machine	100.00
Optical disk	storage	machine	1000.00
Magnetic tape	storage	machine	2000.00
Magnetic disk	storage	machine	2000.00-10,000.00

©1998 Morgan Kaufmann Publishers

Exemplo I/O : Drives de disco



- **Para acessar dados:**
  - **— seek: posição da cabeça sobre a trilha correta (8 to 20 ms. avg.)**
  - **— latência rotacional: aguardar pelo setor desejado (.5 / RPM)**
  - **— transferência: obter os dados (um ou mais setores) 2 to 15 MB/sec**

©1998 Morgan Kaufmann Publishers

Exemplo I/O : Drives de disco

- setor = 512 bytes; 5400 rpm; tempo médio de procura = 12 ms
  taxa de transferência = 5 MB/sec; overhead controlador= 2ms
- Qual é o tempo médio para ler ou escrever em um setor?

Tempo médio de acesso = tempo médio de procura
+ latência rotacional
+ tempo de transferência
+ overhead do controlador

$$12ms + 5.6ms + \frac{0.5KB}{5120KB/\text{sec}} + 2ms = 12 + 5.6 + 0.1 + 2 = 19.7ms$$

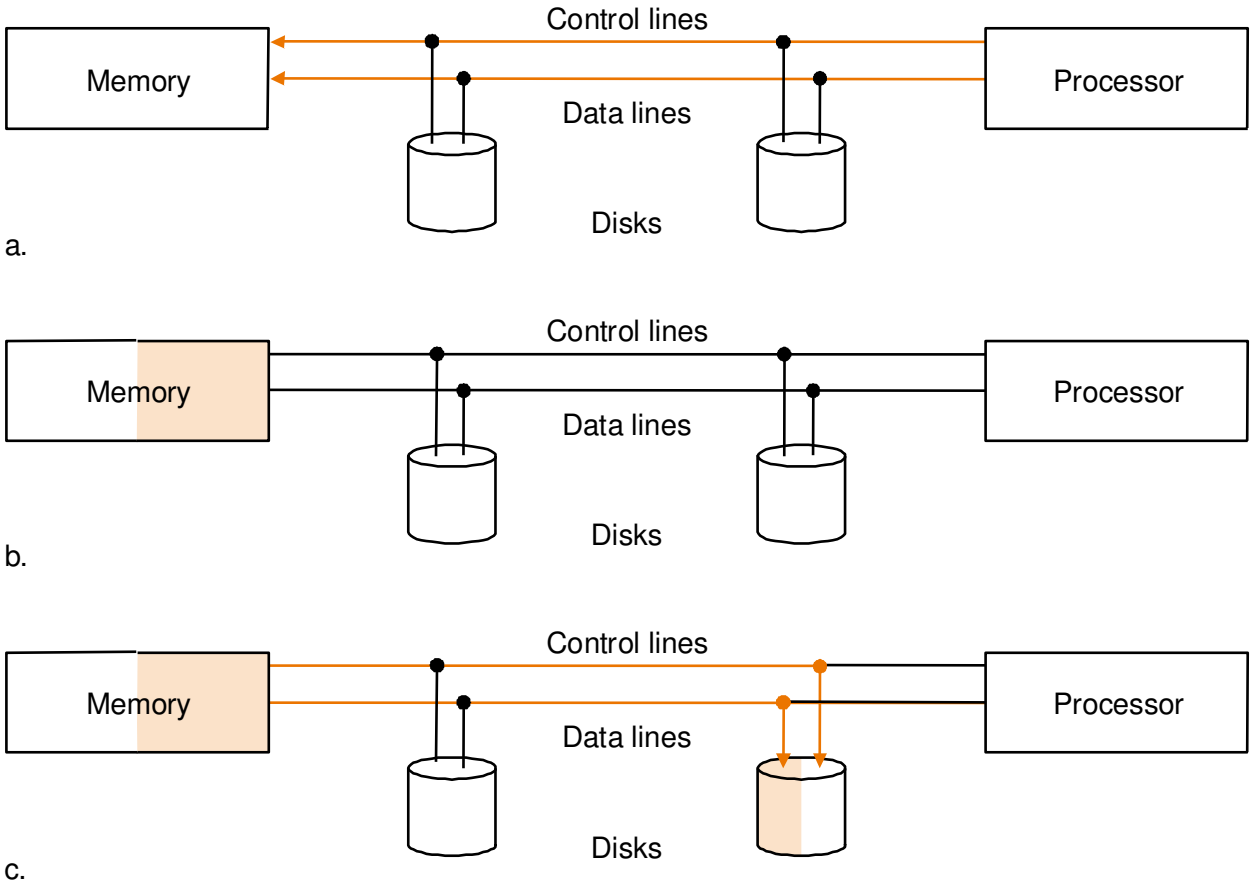
©1998 Morgan Kaufmann Publishers

Exemplo I/O : Barramentos

- **Canal de comunicação compartilhado (um ou mais fios**
- **Dificuldade de projeto:**
  - **— pode ser o gargalo do sistema**
  - **— tamanho do barramento**
  - **— número de dispositivos conectados ao barramento**
  - **— diferenças (buffers buffers com maior largura de banda aumentam a latência)**
  - **— suporte para diferentes dispositivos**
  - **— custo**
- **Tipos de barramentos:**
  - **— processador-memória**
  - **— *backplane* (e.g., PCI)**
  - **— I/O (e.g., SCSI)**
- **Síncrono vs Assíncrono**
  - **— usar clock e protocolo síncrono,**
    rápido mas não escalável
  - **— não usar clock e *handshaking***

©1998 Morgan Kaufmann Publishers

Operação de output



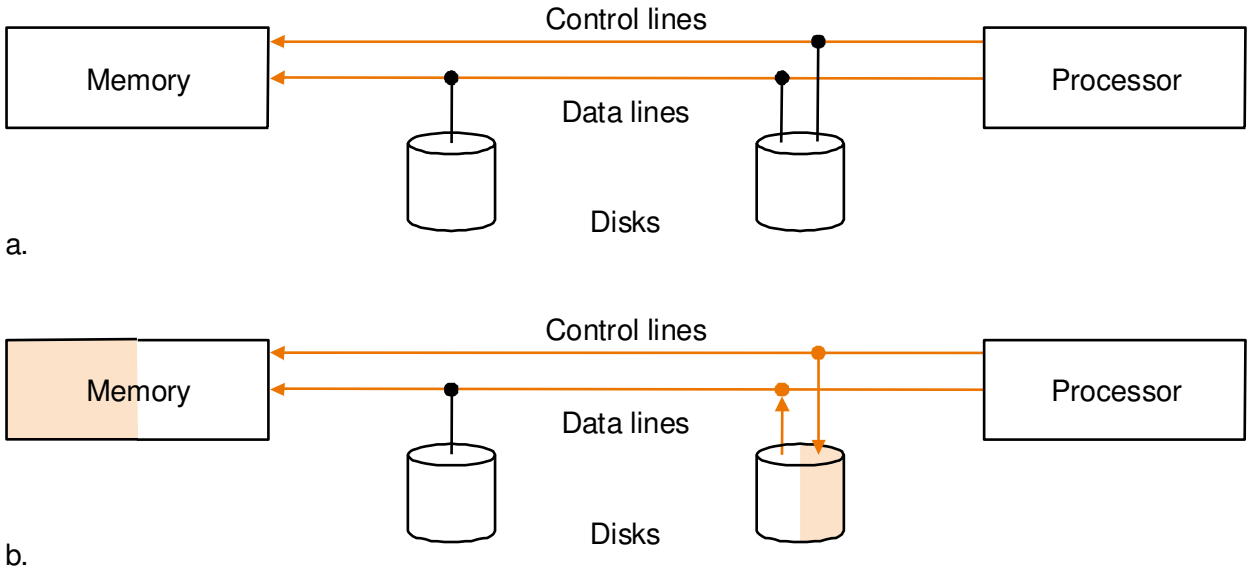
Proc. inicia leitura (RD + endereço)

Memória executa leitura

dados da memória para o disco

©1998 Morgan Kaufmann Publishers

Operação de Input

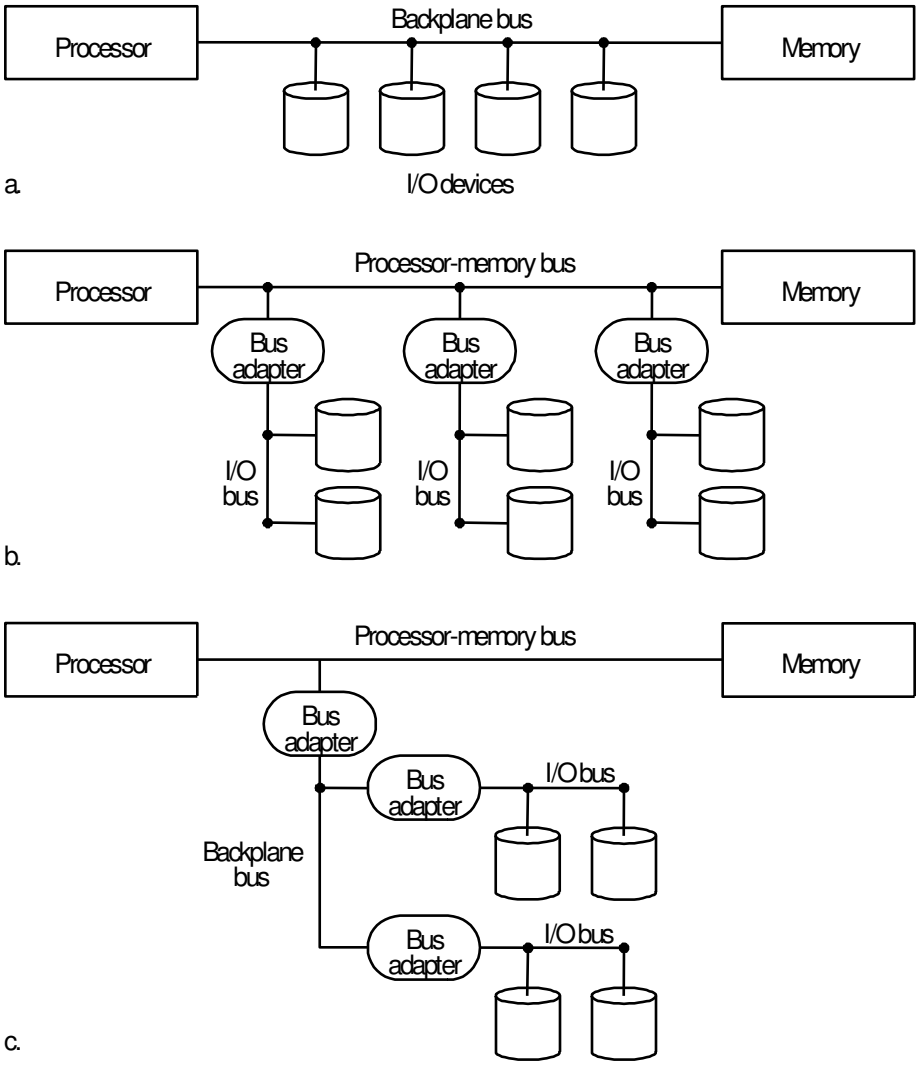


Proc. inicia escrita (WR + endereço)

dados do disco para a memória

©1998 Morgan Kaufmann Publishers

Barramentos



- aspectos de importância
  - baixo custo
  - flexibilidade
  - gargalos
  - compromento
    - clock skew

©1998 Morgan Kaufmann Publishers

Alguns tipos de barramentos

- **CPU-Memória**
  - **geralmente específicos**
  - **curtos**
  - **alta velocidade**
- **Barramentos de I/O**
  - **longo**
  - **muitos dispositivos**
  - **não tem interface direta com a memória**
  - **pode ser padronizado**
- **Backplane**
  - **objetivo: barramento padrão para permitir a interconexão de vários tipos de dispositivos**
  - **pode ser padronizado**

©1998 Morgan Kaufmann Publishers

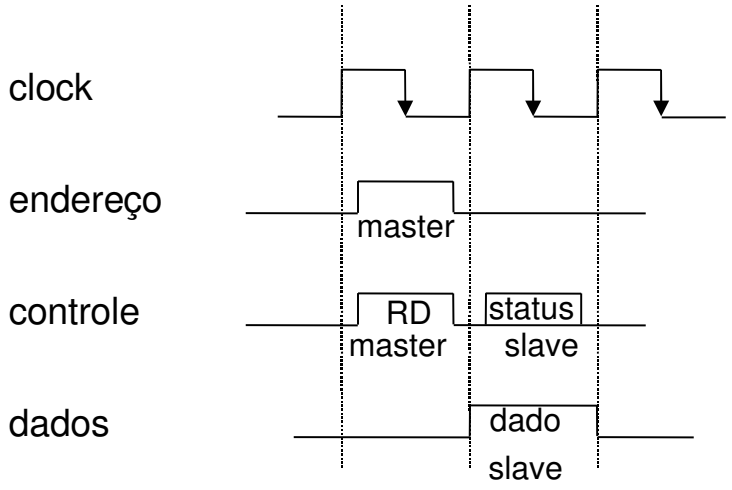
Comunicação síncrona

- Sinais (dados e controle) são sincronizados por um relógio
- Controle pode ser feito por uma das unidades (bus master)
- Cada item é transferido em time slot predefinido e conhecido pelas partes
- Vantagem:
  - controle simplificado
- Desvantagem:
  - pouca flexibilidade
  - velocidade limitada pelo dispositivo mais lento
  - clock skew
  - é normalmente usado na comunicação CPU-MEM

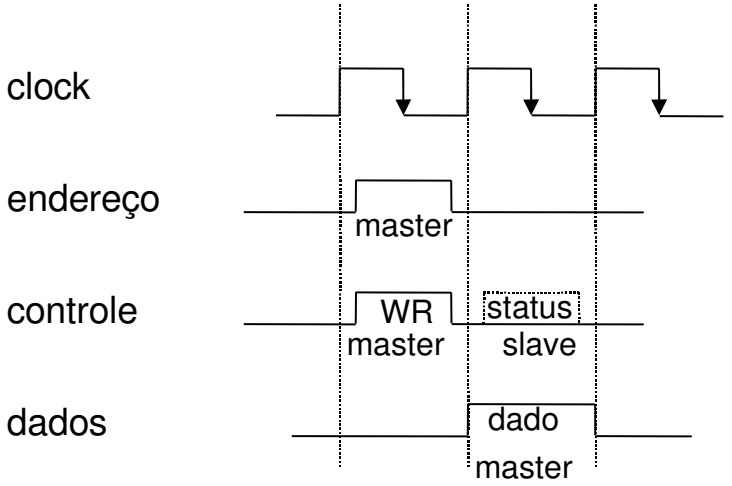
©1998 Morgan Kaufmann Publishers

Exemplo de comunicação síncrona

Master requisita leitura



Master requisita escrita



- bordas do clock:
  - delimitam o time slot
  - definem o instante para a amostragem

©1998 Morgan Kaufmann Publishers

Barramento Assíncrono: I/O - Mem

1- memória percebe ReadReq, lê endereço no barr. de dados e ativa Ack para indicar que já leu

2- I/O percebe Ack ativo e baixa ReadReq e libera o barr. de dados

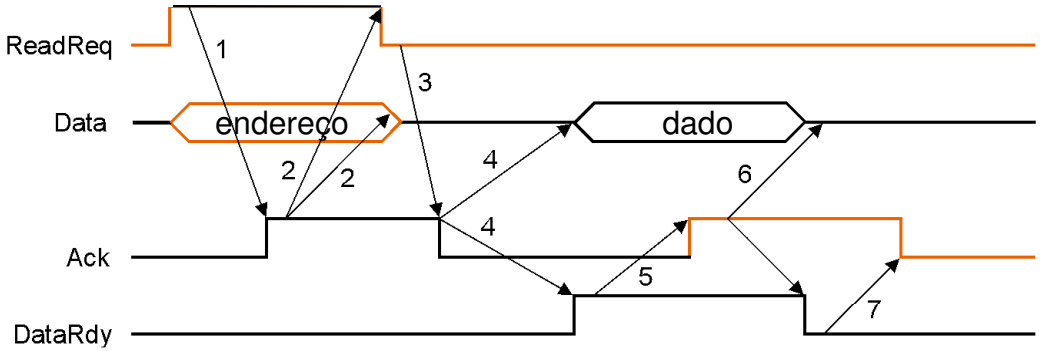
3- memória vê ReadReq baixo e baixa o Ack

4- memória coloca o dado lido no barr. de dados e ativa o DataRdy

5- I/O percebe DataRdy, lê o dado do barr. de dados e ativa Ack para indicar que já leu

6- memória vê Ack ativo, baixa o DataRdy e libera o barr. de dados

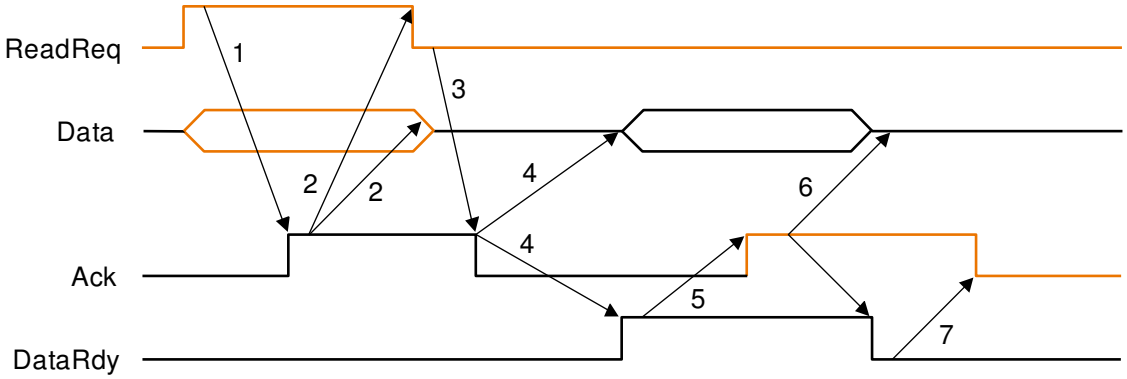
7- I/O vê DataRdy baixo e baixa o sinal Ack, completando o ciclo



- Vantagens:
  - não tem problemas de clock skew
  - flexibilidade
  - velocidade
- Desvantagens:
  - complexidade do controle
- Normalmente usado em barr. de I/O

©1998 Morgan Kaufmann Publishers

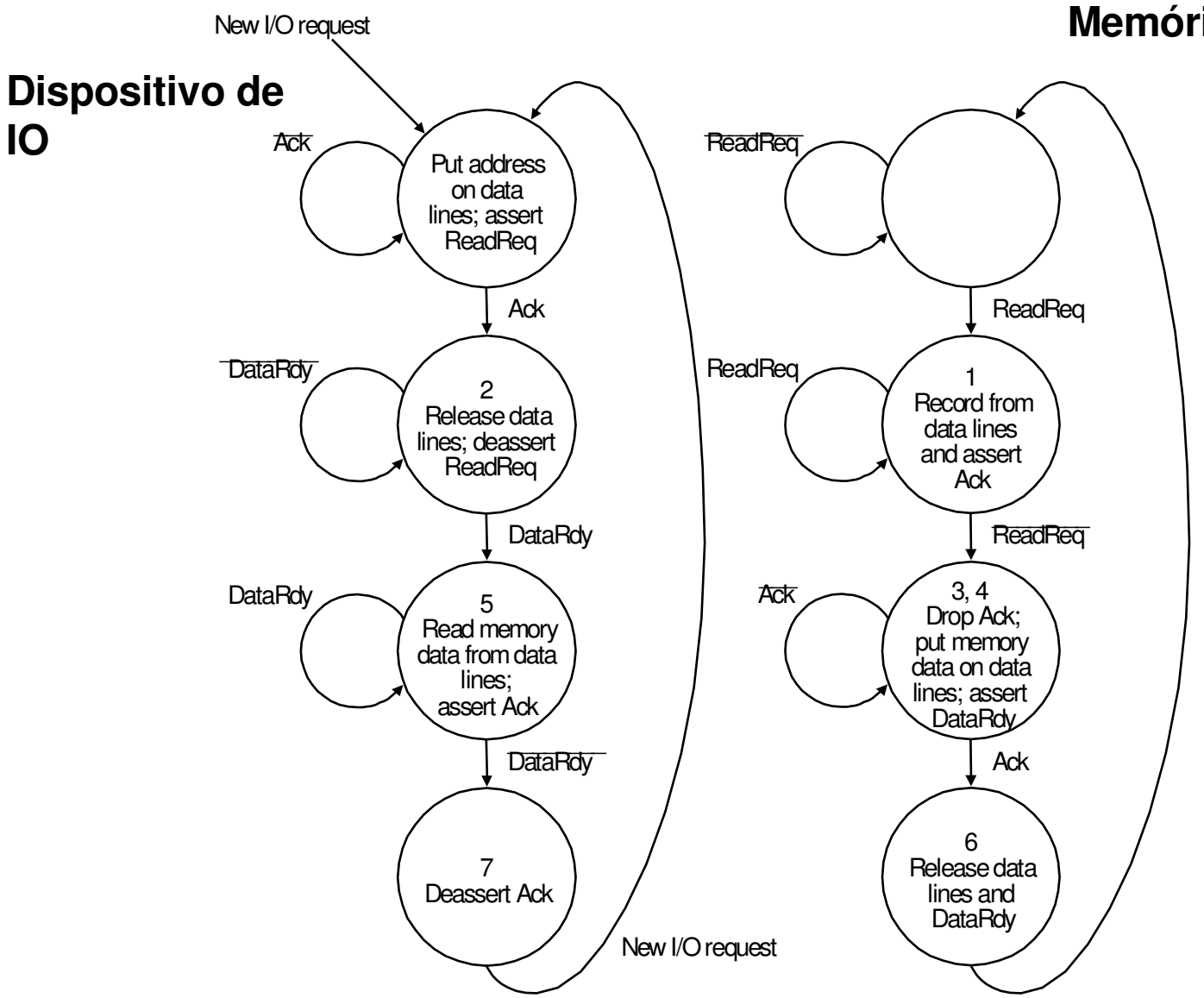
Alguns exemplos de problemas



- **Protocolo de handshaking assíncrono – ler dados na memória e recebe-los em um dispositivo de I/O**

©1998 Morgan Kaufmann Publishers

MEF- Protocolo de Handshaking



©1998 Morgan Kaufmann Publishers

Arbitragem de Barramentos

- **Barramento é compartilhado:**
  - **como ganhar acesso?**
- **Solução mais simples:**
  - **centralizado (bus master)**
  - **CPU-Mem; Master-Slave**
- **Problema: sobrecarga para a CPU**
  - **alternativa: vários bus masters**
- **Arbitragem: Request - Grant**
- **Objetivo: justiça, sem deadlocks ou starvation**

©1998 Morgan Kaufmann Publishers

Tipos de Arbitragem

- **Centralizado: bus master centralizado recebe múltiplos requests e tem múltiplas linhas de Grant; desvantagens: gargalo e confiabilidade**
- **Distribuído: múltiplas linhas de Request; dispositivos examinam o barramento e julgam a sua própria prioridade (NuBus)**
- **Distribuído c/ detecção de colisão (ex: Ethernet; CSMA /CD, carrier sense multiple access / colision detection; meio físico: cabo coaxial, par trançado, fibra óptica, rádio)**

1- envia sinal

2- mede o meio

3- se OK -> FIM

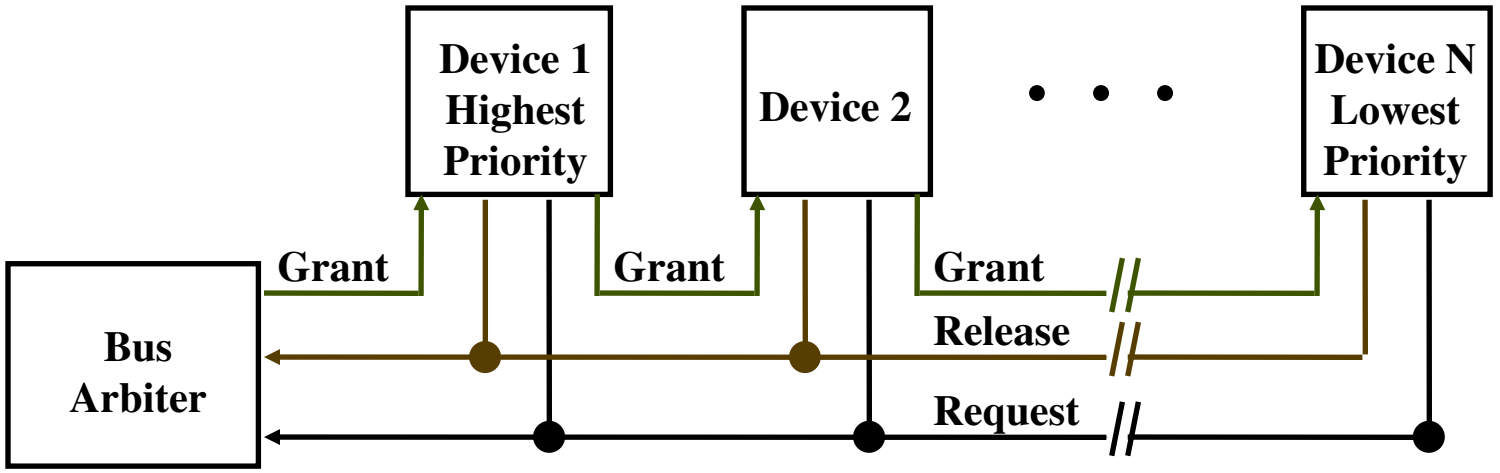
4- espera tempo aleatório

5- volta -> 1

6- FIM

©1998 Morgan Kaufmann Publishers

Tipos (cont.): Daisy Chain



- **Prioridade:**
  - **hardwired pela posição**
- **Expansão: simples (online)**
- **Confiabilidade:**
  - **ruim**

©1998 Morgan Kaufmann Publishers

Comparação entre métodos de arb.

- **Centralizado:**
  - **prioridade: programável (off line)**
  - **expansão**
  - **confiabilidade e deadlock: OK se controle OK**
- **Distribuído e ethernet:**
  - **mais flexível, expansível e confiável**

©1998 Morgan Kaufmann Publishers

Escolha de barramentos

- **Prioridade, flexibilidade, expansão, confiabilidade, velocidade, tipo de tráfego, distância e "starvation"**

Opção	Alto Desempenho	Baixo Custo
Largura de barr.	endereço e dados separados	endereço e dados multiplexados
Largura de dados	>= 32 bits	8 bits
Bloco de transf.	múltiplas palavras	uma palavra
Bus master	múltiplos masters	um master
Clocking	síncrono	assíncrono

©1998 Morgan Kaufmann Publishers

Dois padrões de barramento

Característica	PCI	SCSI
Tipo	backplane	I/O
Largura	32-64	8-32
Dados/end mux´ed	muxed	muxed
Número de masters	multiple	multiple
Arbitração	centralized, parallel arbitration	self-selection
Clocking	Synchr. 33-66 MHz	Synchr. 5-10 MHz or Asynchr.
Largura de banda de pico	133-512 MB/s	5-40 MB/s
Largura de banda estimada	80 MB/s	2.5-40 MB/s (sync) 1.5 MB/s (async)
Número máximo de dispositivos	1024 (32 / bus segment)	7-31 (bus width - 1)
Tamanho máximo do barramento	0,5 meter	2,5 meter
Padrão	PCI	ANSI X3.131

©1998 Morgan Kaufmann Publishers

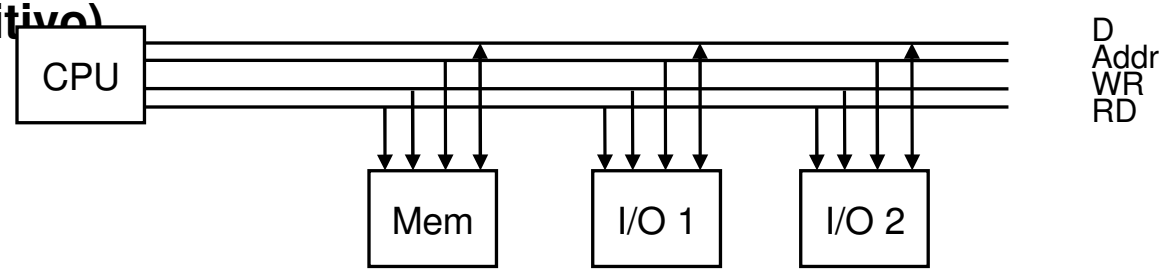
Interface E/S - Mem - CPU - OS

- **Questões:**
  - **como o usuário controla a E/S?**
  - **dados <-> memória ?**
  - **qual é o papel da CPU?**
- **Papel da CPU é grande:**
  - **multiprogramação**
  - **dispositivos de E/S normalmente usam interrupção executada no modo supervisor do OS**
  - **controle de baixo nível de E/S é complexo**

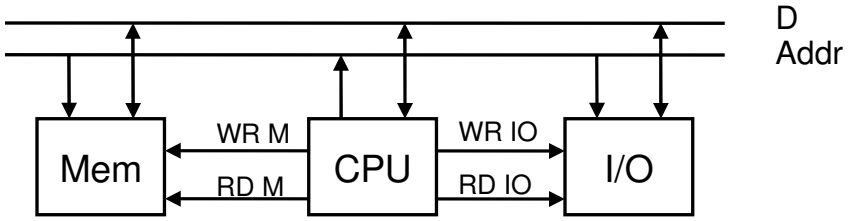
©1998 Morgan Kaufmann Publishers

Comunicação CPU ⇒ E/S

- **I/O programado: CPU inicia, controla, supervisiona e termina a comunicação**
  - **Mapeado na memória: acesso à memória e I/O é feito com as mesmas instruções (espaço de endereçamento define o dispositivo)**



  - **Mapeado em I/O : há instruções e linhas de controle dedicadas à comunicação de I/O**



©1998 Morgan Kaufmann Publishers

Comunicação E/S ⇒ CPU

- **Alternativa 1: *polling***
  - **a CPU periodicamente consulta os dispositivos**
  - **desvantagens:**
    - desperdício do tempo da CPU
    - dispositivo pode ter que ficar muito tempo aguardando o serviço
- **Alternativa 2: interrupção**

©1998 Morgan Kaufmann Publishers

Comunicação E/S ⇒ CPU (cont)

- **Interrupção:**
  - **dispositivo de E/S avisa a CPU que precisa de atenção, por meio de sinal assíncrono**
  - **a CPU deve completar a execução da instrução antes de atender a interrupção:**
    - (pipeline: é necessário stall e tratamento de todas as instruções no pipeline)
  - **fornece indicação à CPU sobre o tipo de interrupção e a identidade do dispositivo**
    - solução comum: vetor de interrupção usado para decidir o endereço da rotina de tratamento da interrupção
  - **em caso de múltiplos dispositivos:**
    - priorização para decidir qual pedido será atendido
      - (é possível o uso de um controlador de interrupção)

©1998 Morgan Kaufmann Publishers

Exemplo

- **Impacto do polling para 3 dispositivos (fck = 500 MHz; ciclos p/ polling = 400)**

Mouse

taxa de polling = 30 / s

30 polling/s = 30 * 400 ciclos/s = 12000

% ciclos da CPU = $12000 / 500*10^6$ =

0,002 %

ACEITÁVEL

Floppy Disk

blocos de 16 bits;
taxa 50 KB/s

pollings / s =
50 KB /s /(2 B / polling) =
25 k pollings / s

400 ciclos /polling ->
25 k *400 =
$100*10^5$ ciclos de polling / s

% de ciclos da CPU =
$10^7 / 500*10^6$ = 2%

ALTO MAS +- OK para sistemas de baixo custo

Disco Rígido

blocos de 4*32 bits; 4 MB / s

pollings / s =
4 MB /s /(16 B / polling) =
$0.25*10^6$ pollings / s

400 ciclos /polling ->
$100*10^6$ ciclos de polling / s

% de ciclos da CPU =
$100*10^6 / 500*10^6$ = 20%

Tempo inaceitável de ocupação da CPU para polling

©1998 Morgan Kaufmann Publishers

Exemplo (cont)

- **Impacto da interrupção sobre o Disco Rígido do exemplo anterior (com overhead da transferência de 500 ciclos, incluindo a interrupção, assumindo que o HD está ativo apenas 5% do tempo):**

Taxa de acessos é o mesmo de polling = 250 K / s

ciclos / s = 250 k * 500 = 125* 10^6 ciclos / s

% da CPU = 125 *10^6 / 500 *10^6 = 25%

perto do Polling, MAS, levando em consideração que o HD não está ativo apenas 5% do tempo:

impacto 5% * 25% = 1,25 % (bastante razoável)

Interrupção só envolve a CPU se houver atividade

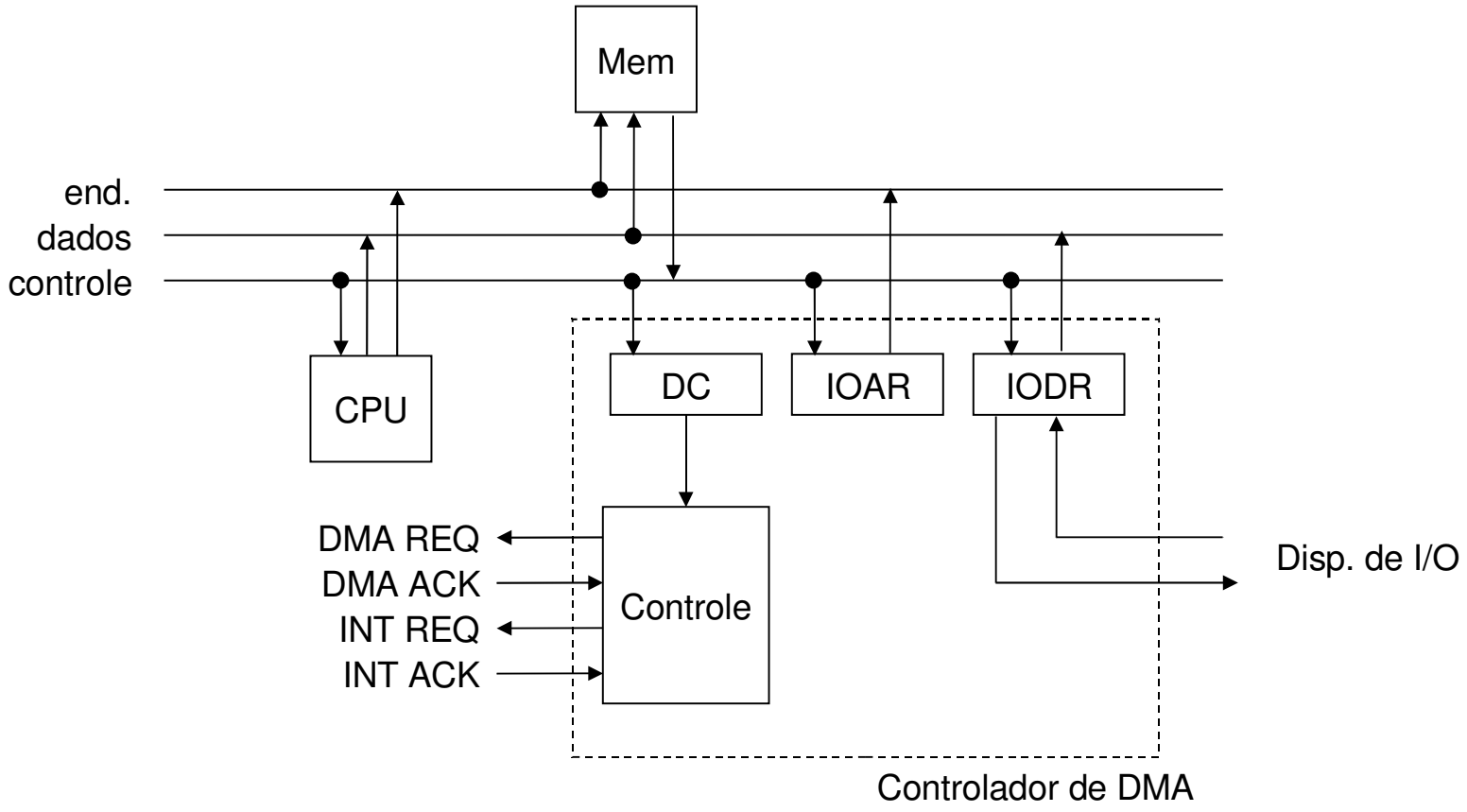
©1998 Morgan Kaufmann Publishers

DMA: dispositivo ⇔ memória

- **Usando polling ou interrupção a CPU é envolvida diretamente na transferência**
  - **interrupção pode ser OK para pequenos blocos; CPU se ocupa de outra coisa enquanto o dispositivo está trabalhando mas a escrita/leitura envolve diretamente a CPU**
- **Solução: DMA (Direct Memory Access)**
  - **não há polling e a CPU não participa da transferência**
  - **Passos:**
    - 1- CPU informa: identidade do dispositivo, qual é a operação, endereço inicial da memória, Nº de bytes da transferência
    - 2- Controlador de DMA inicia a operação. Solicita e ganha controle do barramento. Faz a transferència entre o dispositivo (endereço, RD/WR, incrementa endereço, etc) e a memória.
    - 3- Controlador de DMA interrompe a CPU e avisa que a transferência está completa

©1998 Morgan Kaufmann Publishers

Típico Controlador de DMA



©1998 Morgan Kaufmann Publishers

Tipos de Operação de DMA

- **Block Transfer: seqüência de palavras; usado por discos, fitas (vel. do disp. é importante); a CPU não usa o barramento :**
  - **CPU executa instrução de IO; escreve em IOAR e DC**
  - **Controlador faz DMA REQ; CPU -> DMA ACK**
  - **Controlador faz transferência com Inc IOAR; Decr DC**
  - **Se o dispositivo for lento, retorna o controle para a CPU com DMA REQ = 0; estando pronto DMA REQ = 1**
  - **Fim se DC =0**
- **Cycle Stealing: uma (ou mais palavras transmitidas) e o controle devolvido à CPU**
- **DMA transparente: cycle stealing sincronizado de modo a usar barramento somento nos ciclos em que a CPU não usa**
  - **contenção de memória deixa de ser problema sério com o uso de cache**

©1998 Morgan Kaufmann Publishers

DMA, Cache e Memória Virtual

- **Perguntas:**
  - **DMA deve trabalhar com endereço físico ou virtual?**
  - **como transferir se os dados não estão em uma única página?**
- **Soluções:**
  - **se os dados estão em uma única página, trabalhar com endereço físico**
  - **caso contrário, trabalhar com o endereço virtual; o próprio controlador deve fazer o mapeamento (pequena page table interna)**
- **Outros problemas:**
  - **O sistema operacional não deve mover páginas afetadas enquanto o DMA está em operação**
  - **pode acontecer incoerência de cache-memória devido a atualização da memória pelo DMA**

©1998 Morgan Kaufmann Publishers

Exemplo; impacto em Hard Disk

- **Mesmo exemplo anterior:**
  - **bloco 16B; taxa de transferência de 4 MB / s; clock de 500 MHz**
  - **setup do DMA: overhead de 1000 ciclos**
  - **interrupção do DMA 500 ciclos**
  - **tamanho médio de transferência: 8 KB**
  - **atividade do HD: 100% do tempo (ignorar contenção de barramento)**

tempo por transferência: 8 KB / 4 MB/s = 2 ms

Para atividade constante do HD:

$$\frac{(1000 + 500) \text{ ciclos / transf.}}{2 \text{ ms / trans.}} = 750 * 10^3 \text{ ciclos /s}$$

$(750 * 10^3 \text{ ciclos/s}) / (500 * 10^6 \text{ ciclos/s}) = 1{,}5 * 10^{-3} = 0{,}15\%$

CPU é só envolvida no início e no final

como atividade < 100 % este percentual é ainda menor

©1998 Morgan Kaufmann Publishers

Exemplo: Projeto de Sistemas de I/O (1)

- Dados:
  - CPU com 300 MIPS e 50K instruções (no OS) por operação de I/O
  - Memory backplane: taxa de transmissão de 100 MB/s
  - Controladores SCSI-2: 20 MB/s e capacidade para 7 discos
  - HD com taxa de 5 MB/s e atraso seek+rotational = 10 ms
  - workload de IO: leituras de 64 KB (100 K instruções de usuário / Operação de I/O)
- Encontrar: máxima taxa de I/O e o número de (HD+controladores) necessário

- Dois componentes (CPU e memory bus), qual é o gargalo?
  - cada op de I/O precisa de 50 K + 100K instruções

    IO rate da CPU = (Instr rate) / (Instr. / IO) = $300*10^6$ / $(100+50)*10*^3$ = 2000 IO/s

    IO rate do bus = (bus bandwidth) / (bytes / IO) = $100*10^6/64*10^3$ = 1562 IO/s

    gargalo é o barramento

    agora, configurar o resto do sistema para acomodar 1562 IO/s -> quantos HDs?

©1998 Morgan Kaufmann Publishers

Exemplo: Projeto de Sistemas de I/O (2)

- Meta: 1562 IO/s
- Tempo por IO no disco: 10 ms + 64 KB / (5 MB/s) = 10ms + 12.8ms = 22.8ms

  cada disco: 1/22.8ms = 43.9 IO/s

  para ocupar totalmente o barramento

  1562 / 43.9 = 36 discos
- Número de controladores SCSI

  Taxa de transf. / disco = 64 KB / 22.8ms = 2.74 MB /s

  possível colocar 7 discos por controlador (7*2.74 < 20 MB/s)

  Número de barramentos de IO e controladores SCSI =

  36 discos / (7 discos / barramento) = 6 (barramentos + controladores
- Grande número de hipóteses simplificadoras: melhor precisão somente com simulação

©1998 Morgan Kaufmann Publishers