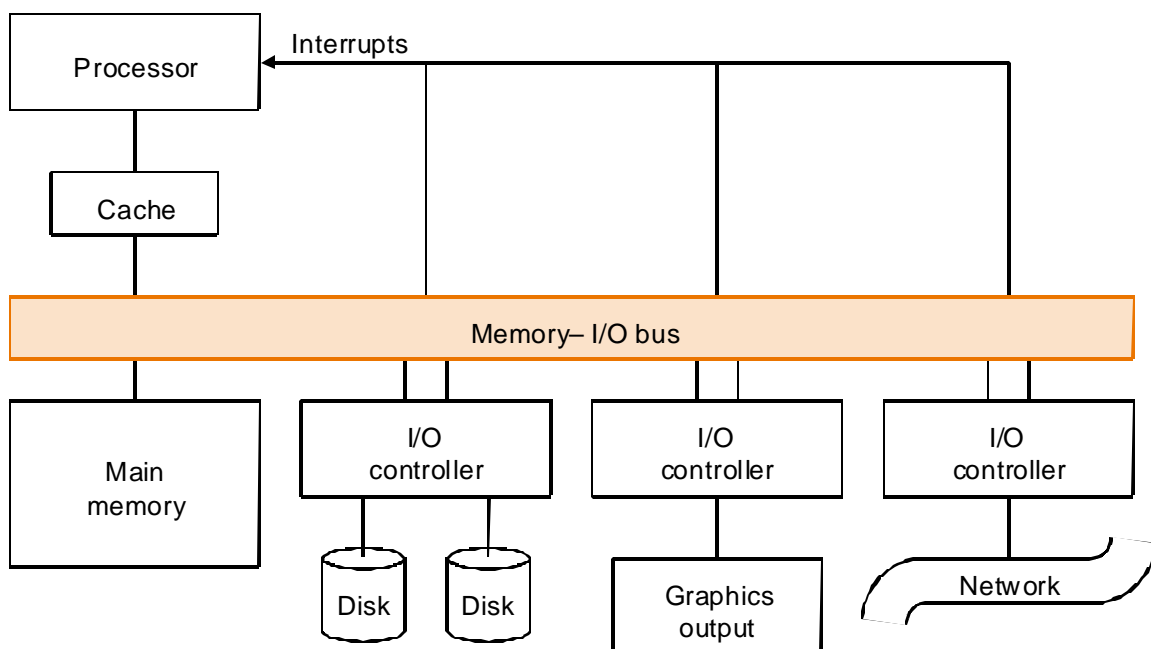


8. Interfaceamento entre Processador e Periféricos

- **Características de projeto de sistemas de I/O** → expansibilidade, robustez frente a falhas
- **Desempenho depende:** da característica do dispositivo, da conexão do dispositivo e o resto do sistema, hierarquia de memória e do sistemas operacional.



- **Impacto do desempenho de um sistema de I/O**

Exemplo

Suponha que temos um programa que gasta 100 segundos de elapsed time em sua execução, onde 90 segundos corresponde ao tempo de CPU o resto de I/O. Se o tempo de CPU melhora em 50% por ano nos próximos 5 anos mas o I/O não melhora, quanto mais rápido rodará o programa no fim de 5 anos ?

Solução:

$$\text{Elapsed time} = \text{CPU time} + \text{I/O time}$$

$$100 = 90 + \text{I/O time}$$

$$\text{I/O time} = 10$$

Após n anos	CPU time (seg)	I/O time (seg)	Elapsed time (seg)	% I/O time
0	90	10	100	10%
1	$90/1.5 = 60$	10	70	14%
2	$60/1.5 = 40$	10	50	20%
3	$40/1.5 = 27$	10	37	27%
4	$27/1.5 = 18$	10	28	36%
5	$18/1.5 = 12$	10	22	45%

A melhoria de desempenho da CPU em 5 anos é : $90/12 = 7.5$

A melhoria do elapsed time em 5 anos é : $100/22 = 4.5$

Portanto o tempo de I/O cresceu de 10% a 45% do elapsed time.

- Desempenho de sistemas de I/O depende da aplicação.
- Devemos nos preocupar inicialmente com throughput → I/O bandwidth.
- I/O bandwidth pode ser medido de duas maneiras:
 - Quantos dados podem ser movimentados através do sistema em um certo tempo (para sistemas que trabalham com blocos grandes de dados);

- **Quantas operações de I/O podem ser feitas por unidade de tempo (para sistemas que trabalham com grande número de pequenos acessos a dados).**
- **A melhor medida dependerá do ambiente.**
- **Alguns sistemas necessitam de ambos, um alto throughput e pequenos tempos de respostas (ATMs, sistemas de reservas aéreas, file servers, etc).**

- **Medidas de desempenho de sistemas de I/O**
 - **Desempenho de sistemas de I/O dependem da taxa na qual o sistema transfere os dados.**
 - **A taxa de transferência ($MBs = 10^6 \text{ bytes/s}$) depende da frequência de clock ($Mhz = 10^6 \text{ ciclos por segundo}$).**

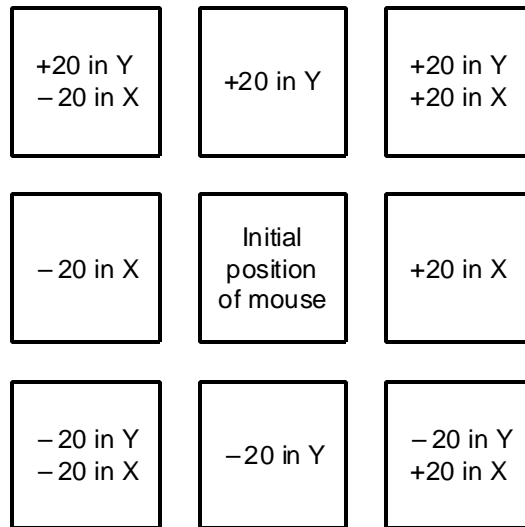
- **Medidas de I/O de supercomputadores**
 - **Supecomputadores → I/O consiste de uma leitura grande de dados seguido de escritas conforme os dados são processados → a medida é throughput de dados → o número de bytes por segundo que podem ser transferidos entre a memória principal do supercomputador e os discos, durante uma transferência grande.**

- **Medidas de Transition Processing (TP) I/O**
 - **Aplicações TP → necessitam de requisitos de tempo de resposta e throughput.**
 - **Aplicações TP → I/O rate = número de acessos a disco por segundo (data rate = bytes de dados por segundo).**
- **Medida de File Systems**
 - **No file systems do Unix, → 80% do acessos são arquivos menores que 10KB e 90% são dos arquivos acessados são dados com endereços seqüenciais no disco. 67% dos acessos são para leitura, 27% para escrita e 6% de RMW (read-modify-write) → benchmark constitui de 70 arquivos, num total de 200KB com 5 fases:**
 - **MakeDir;**
 - **Copy**
 - **ScanDir**
 - **ReadAll**
 - **Make**

- **Tipos e Características de dispositivos de I/O**
- **Três características que identificam a variedade de dispositivos de I/O**
 - **Comportamento:** input (uma só leitura), output (write only) ou storage (releitura e rescrita)
 - **Partner:** é o outro ponto da comunicação: homem ou máquina.
 - **Data rate:** que taxa o dado pode ser transferido de um ponto ao outro.

Dispositivos	Comportamento	Partner	Data rate (KB/seg)
Teclado	input	humano	0,01
Mouse	input	humano	0,02
Entrada de voz	input	humano	0,02
Scanner	input	humano	400,00
Saída de voz	output	humano	0,60
Impressora	output	humano	1,0
Imp. Laser	output	humano	200,0
Display gráfico	output	humano	60.000,00
Modem	input ou output	máquina	2,00 ~ 8,00
Rede/LAN	input ou output	máquina	500,00 ~6.000,00
Floppy disk	storage	máquina	100,00
Optical disk	storage	máquina	1.000,00
Magnetic tape	storage	máquina	2.000,00
Magnetic disk	storage	máquina	2.000,00~10.000,00

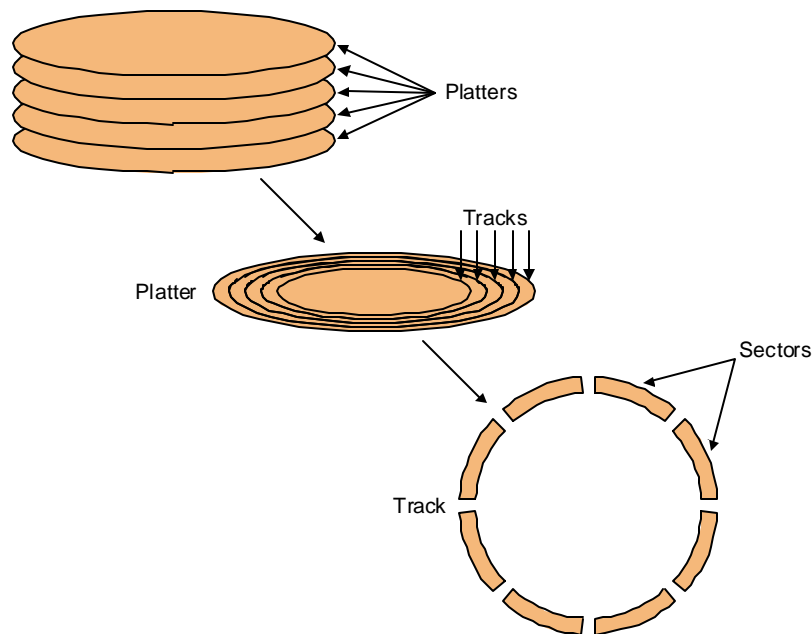
- **Mouse**



- **Disco magnético**

- **Hard disk**

- **1 a 15 pratos com os dois lados cobertos por um material magnético, com diâmetro de 1 a 8 polegadas;**
 - **a pilha de pratos roda a uma velocidade de 3600 a 7200 RPM;**
 - **cada prato é dividido em trilhas e setores → 1000 a 5000 trilhas, todas as trilhas de uma determinada posição em todos os pratos são denominados de cilindros.**
 - **cada trilha é dividido em setores: 64 a 2000 setores;**
 - **cada setor tem 512 bytes (1997);**
 - **seqüência → número do setor, gap, informações gravadas no setor, incluindo check de erro.**



- **Para acessar dados no disco, o sistema operacional deve:**
 - **posicionar a cabeça sobre a trilha → seek → tempo para posicionar a cabeça → seek time (8ms a 20ms) → seek time real é 25% a 35% do tempo dito pelo fabricante → localidade das referências do disco.**
 - **posicionamento do setor → rotacional latency ou rotacional delay → $0.5 \text{ rotação} / 3600 \text{ RPM} = 8.3 \text{ ms}$ (max) ou $0.5 / 7200 = 4.2 \text{ ms}$ (min).**
- **Tempo de leitura em disco**

Exemplo

Qual o tempo médio de leitura ou escrita de um setor (512 bytes) para um disco com rotação de 5400 RPM ? O seek time médio é de 12 ms, a taxa de transferência é de 5 Mb/Seg e o overhead do controlador é 2 ms. Assumir que não tem waiting time.

Solução:

Tempo médio de acesso = seek time médio + rotational delay médio + tempo de transferência + overhead de controle → 12 ms + 5.6 ms + 0.5 KB / 5 MB por seg + 2ms = 19.7 ms

25% do anunciado → 12 ms * 0,25 + 5.6 ms + 0.1 ms + 2ms = 10.7 ms

- **Características de alguns discos**

Características	Seagate ST423451	Seagate ST423451	Seagate ST423451
Diâmetro do disco (pol)	5,25	3,50	2,50
Capacidade formatada (MB)	23.200	9.100	2.250
MTBF (horas)	500.000	1.000.000	300.000
Núm. de superfícies	28	20	10
Veloc. Rotação	5400	7200	4500
Taxa de transf. interba (Mbits/Seg)	86-124	80-124	Maior que 60.8
Interface externa	Fast SCSI-2 (8-16 bits)	Fast SCSI-2 (8-16 bits)	Fast ATA
Taxa de transf. externa	20-40	20-40	Maior que 16,6
seek mínimo (trilha para trilha)	0,9	0,6	4
seek médio + rotacional delay (ms)	11	9	14
Power box (watts)	26	13	2,6
MB/watt	892	700	865
Volume (pol ao cubo)	32	37	8
MB / pol ao cubo	72	246	273

- **Redes**
 - **Características:**
 - **Distância: 0,001 a 100.000 Km;**
 - **Velocidade: 0,001 MB/seg a 100 MB/seg;**
 - **Topologia: Bus, ring, star, tree**
 - **Linhas Compartilhadas: nenhuma (ponto-a-ponto) e compartilhada (multidrop).**

- **Exemplos de Redes**
 - **Padrão RS232 → velocidade de 0,3 a 19,2 Kbit/seg, conexão ponto-a-ponto, distância de 10 a 100 metros, topologia estrela se diversos terminais estão ligados a um computador central.**
 - **LAN – Local Area Network (Ethernet -> mais popular) → 10 Mbit/seg, barramento de um fio, pacotes enviados em blocos de 64 a 1518 bytes. Fast Ethernet → 10 vezes maior (100 Mbit/seg) . Gigabit Ethernet → proposta em 1998.**
 - **ARPANET – Advanced Research Projects Agency Network (Precursora da Internet) → distâncias de 10 a 10.000 Km, protocolo TCP/IP (Transmission Control Protocol/Internet Protocol).**
 - **ATM – Asynchronous Transfer Method → 155 Mbits/seg a 2.5 Gbits/seg.**

- **Exemplo de desempenho de duas redes**

Considere as seguintes medições feitas em duas estações SPARCstation 10 sob Solaris 2.3, conectadas em dois tipos de redes, usando TCP/IP para comunicação:

Características	Ethernet	ATM
Bandwidth do nó para a rede	1,125 MB/seg	10 MB/Seg
Latência de interconexão	15 us	50 us
Latência de HW da/para a rede	6 us	6 us
Overhead de SW para TX para rede	200 us	207 us
Overhead de SW para RX da rede	241 us	360 us

Encontre a latência de comunicação host-to-host para uma mensagem de 250 bytes, para as duas redes.

Solução:

$$\text{tempo TX}_{\text{Ethernet}} = 250 \text{ bytes} / 1,125 \times 10^6 \text{ bytes/seg} = 222 \text{ us}$$

$$\text{tempo TX}_{\text{ATM}} = 250 \text{ bytes} / 10 \times 10^6 \text{ bytes/seg} = 25 \text{ us}$$

Latência para enviar e receber o pacote:

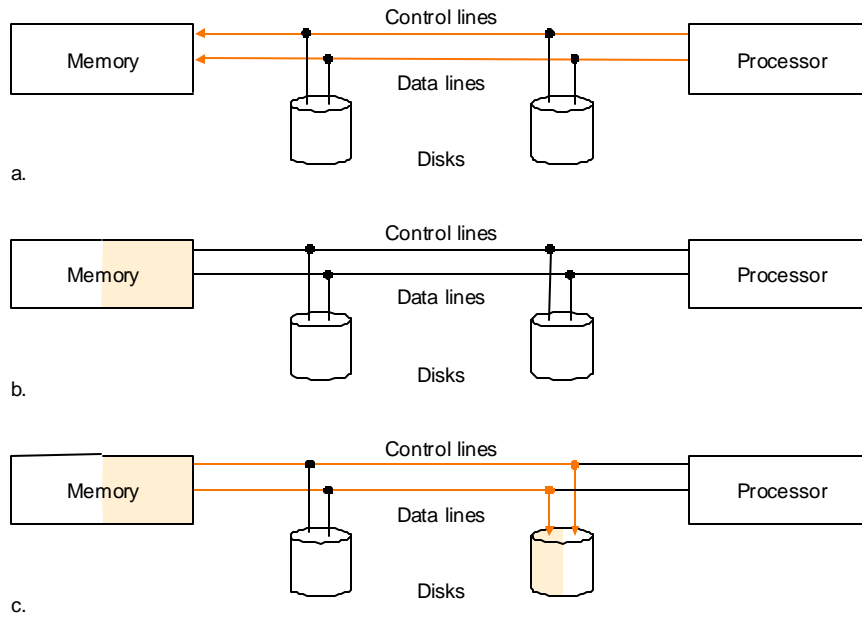
$$\text{Tempo total}_{\text{Ethernet}} = 15 + 6 + 200 + 241 + 222 = 684 \text{ us}$$

$$\text{Tempo total}_{\text{ATM}} = 50 + 6 + 207 + 360 + 25 = 648 \text{ us}$$

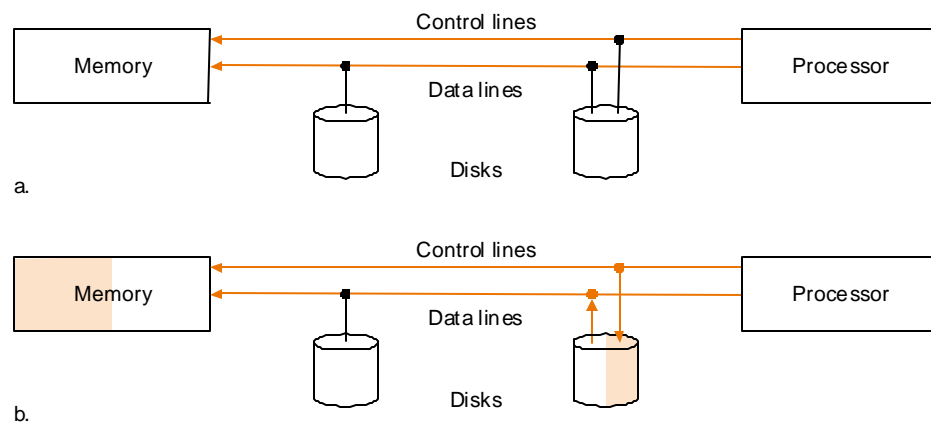
- **Barramentos: Conexão de Dispositivos de I/O ao Processador e Memória**

- **Transação em um barramento:**

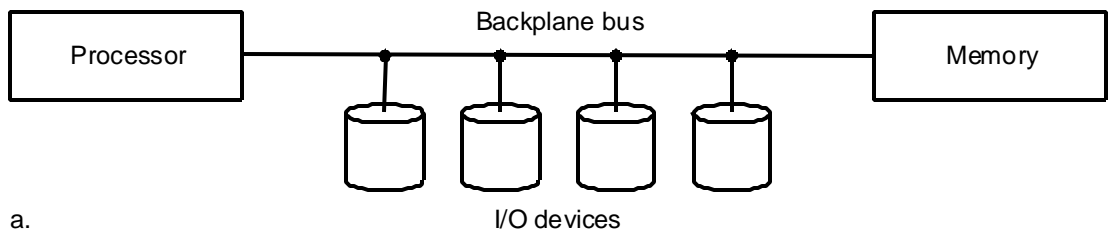
- **Operação de saída:**



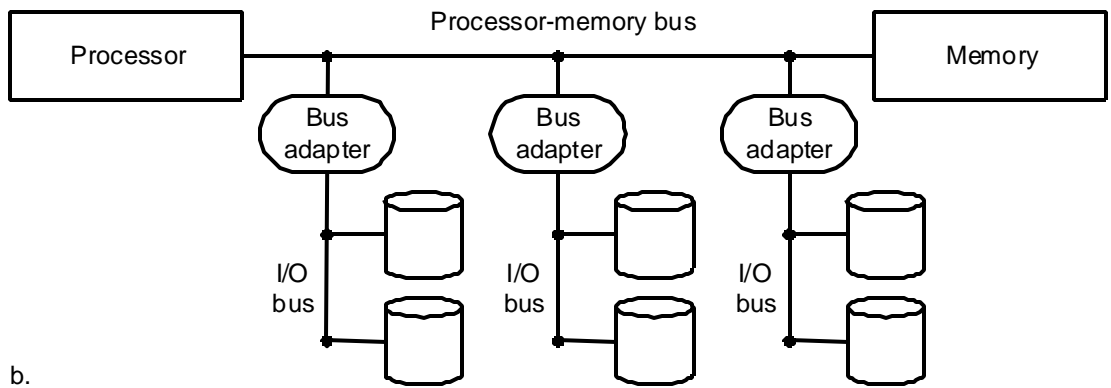
- **Operação de entrada:**



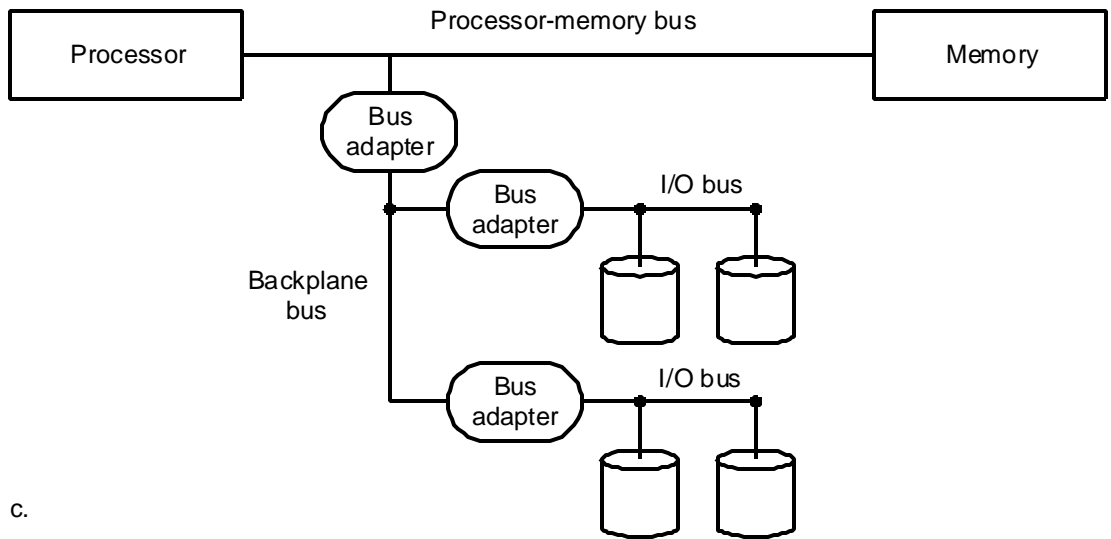
- **Tipos de barramentos**
 - **Barramento processador-memória → curtos, geralmente de grande velocidade com objetivo de maximizar o bandwidth memória-processador;**
 - **Barramento de I/O → longos, podem ter muitos tipos de dispositivos ligados a ele. Não são ligados diretamente à memória e sim um barramento backplane ligado à memória;**
 - **Barramento backplane → recebe este nome porque geralmente é construído em uma estrutura de interconexão no chassi (backpane). Processador, memória e placas de I/O são plugadas no backplane usando este barramento para comunicação.**
 - **Barramentos backplane e I/O → padrões.**
 - **Barramento processador-memória → proprietário.**
- **Na figura a seguir:**
 - **(a) → único barramento usado para comunicação memória-processador e dispositivos de I/O-memória - usado nos PCs.**
 - **(b) Nos PCs → barramento memória-processador é PCI e barramento de I/O é SCSI.**
 - **(c) barramento de I/O → SCSI**



a.

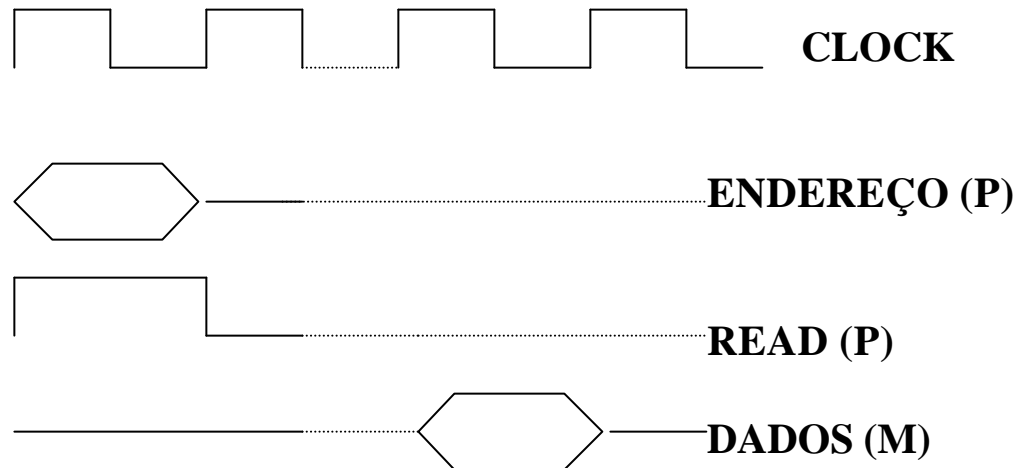


b.



c.

- **Barramento síncrono e assíncrono**
- **Síncrono → Existe um sinal de clock para controle e um protocolo fixo de comunicação relativo ao clock.**
- **Leitura de memória**



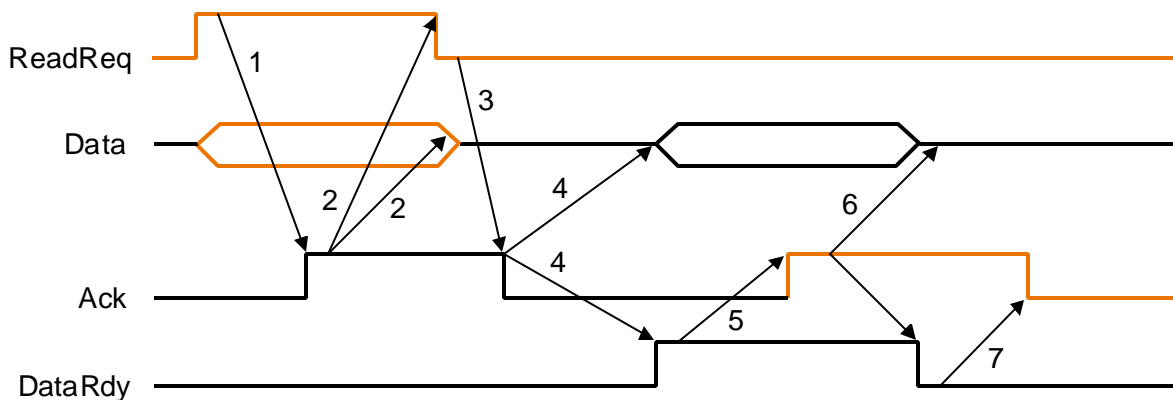
- **Vantagens → fácil implementação, máquina de estados para controle pequena e portanto rápido.**
- **Desvantagens:**
 - **Cada dispositivo deve funcionar na mesma frequência de clock e**
 - **Clock skew → não podem ter distâncias longas se a frequência for grande.**

- Assíncrono → protocolo handshaking

- Sinais de Controle:

- ReadReq
- DataRdy
- Ack

- Leitura de memória



1 → Memória reconhece o pedido de leitura, lê o endereço.

2 → Processador reconhece o Ack gerado pela memória, abaixa o ReadReq e dá um release no barramento.

3 → Memória reconhece que o processador abaixou o ReadReq e abaixa o Ack.

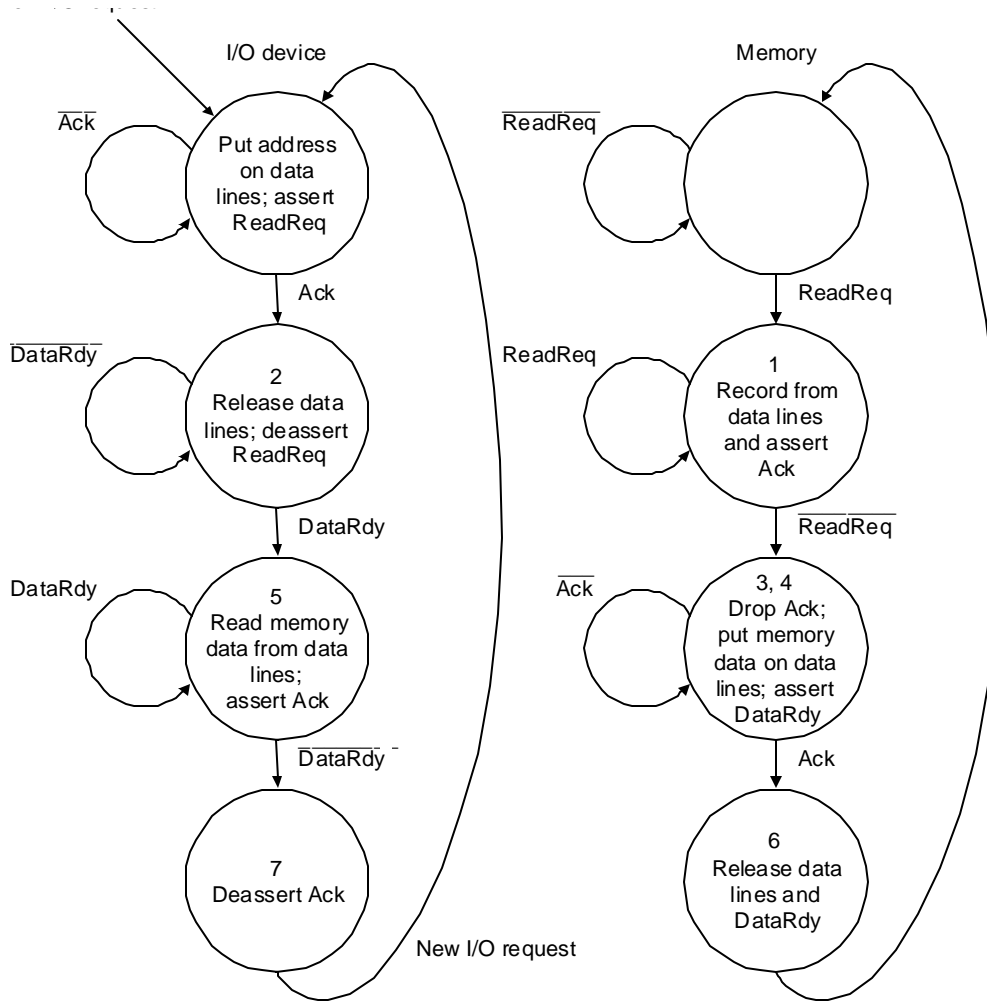
4 → Quando o dado está pronto, a memória o põe no barramento e envia um sinal de DataRdy, para avisar o processador que o dado está no barramento.

5 → Processador reconhece o DataRdy, lê o dado e gera um Ack.

6 → Memória reconhece o Ack, abaixa o DataRdy e dá um release no barramento de dados.

7 → Processador reconhece que o DataRdy está baixo e abaixa o Ack.

- Máquina de estados para controle de I/O assíncrono



- **Análise de performance de barramentos síncronos e assíncronos**

Exemplo

Comparar o bandwidth máximo para um barramento síncrono e um assíncrono. O síncrono tem clock de 50 ns e cada transmissão leva 1 ciclo de clock. O assíncrono requer 40 ns por handshaking. O barramento de dados é de 32 bits. Encontrar o bandwidth para cada barramento em uma leitura de uma palavra de memória, onde a memória tem um tempo de acesso de 200 ns.

Solução:

a) Síncrono

Enviar endereço para a memória → 50 ns

Ler o dado → 200ns

Enviar o dado para o dispositivo → 50 ns

Total → 300 ns

Bandwidth máximo = 4 bytes / 300 ns = 13,3 MB/seg

b) Assíncrono

1 → 40 ns

2,3,4 → max (3 X 40 ns, 200 ns) = 200 ns

5,6,7 → 3 X 40 ns = 120 ns

Total → 360 ns

Bandwidth máximo = 4 bytes / 360 ns = 11,1 MB/seg

- **Aumento do bandwidth do barramento**
 - **Largura de barramento.**
 - **Utilização de barramento de endereços e dados não-multiplexados.**
 - **Transferência em blocos**
- **Análise de desempenho de dois esquemas de barramentos**

Exemplo

Suponha que temos um sistema com as seguintes características:

- 1. Uma memória e um barramento que suportam acesso em blocos de palavras de 4, 16 e 32 bits.**
- 2. Um barramento síncrono com frequência de clock de 200 Mhz, onde cada transferência de 64 bits leva 1 ciclo de clock. O envio do endereço leva também 1 ciclo de clock.**
- 3. Entre cada operação de barramento são necessários 2 ciclos de clock.**
- 4. O tempo de acesso para as quatro primeiras palavras é de 200 ns; cada conjunto adicional de 4 palavras pode ser lido em 20 ns. Assuma que a transferência do último dado e a leitura das próximas 4 palavras possam ser sobrepostas.**

Encontre o bandwidth e a latência para a leitura de 256 palavras para transferências que usem blocos de 4 palavras e para a que usem blocos de 16 palavras. Calcule também o número efetivo de transações por segundo para cada caso. (uma transação de barramento consiste de uma transmissão de um endereço seguido pelo dado).

Solução:

Para bloco de 4 palavras, cada bloco leva:

- 1. 1 ciclo de clock para enviar o endereço á memória.**
- 2. 200ns/5ns por ciclo = 40 ciclos de clock para ler da memória.**
- 3. 2 ciclos de clock para enviar o dado da memória.**
- 4. Dois ciclos entre esta transferência e a próxima.**

Total de 45 ciclos, e 256/4 transações.

Cada transferência leva $45 \times 64 = 2880$ ciclos de clock → latência é de $2880 \times 5 = 14.400$ ns.

Número de transação / seg = 64 transações $\times 1/14.400 = 4,44$ M transações/seg

Bandwidth = $(256 \times 4) \times 1/14,400 = 71,11$ MB /seg

Para bloco de 16 palavras, o primeiro bloco leva:

- 1. 1 ciclo de clock para enviar o endereço á memória.**
- 2. 200ns/5ns por ciclo = 40 ciclos de clock para ler as primeiras 4 palavras da memória.**
- 3. 2 ciclos de clock para enviar o dado da memória. Durante este tempo já se inicia a leitura do próximo bloco.**
- 4. Dois ciclos entre esta transferência e a próxima. Aqui a leitura do próximo bloco será feito.**

Cada um dos três blocos de 4 palavras restante, necessita repetir apenas os dois últimos passos.

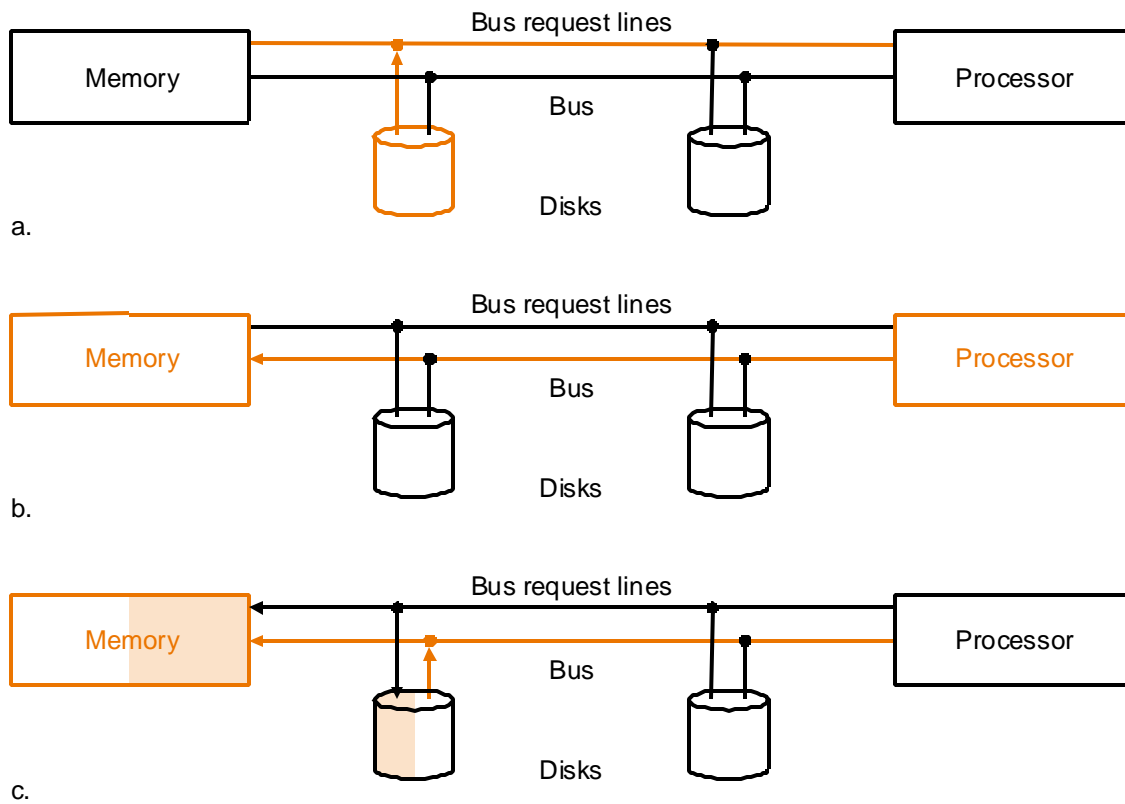
Total de ciclos para cada bloco de 16 palavras = $1 + 40 + 4 \times (2+2) = 57$ ciclos; $256/16 = 16$ transações → total de ciclos = $57 \times 16 = 912$ ciclos.

A latência = $912 \times 5 = 4560$ ns.

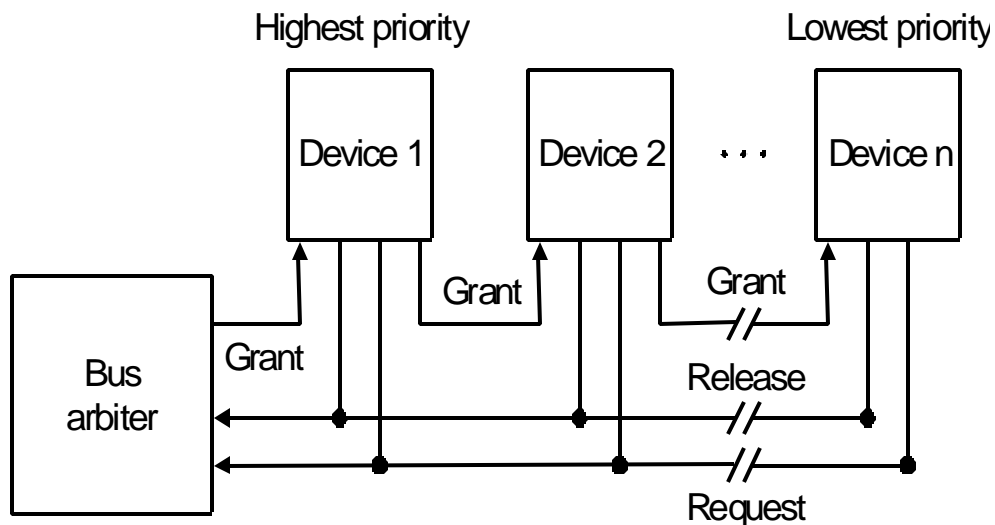
Número de transações por segundo = $16 \times 1/4560 = 3,51$ M transações/seg.

Bandwidth = $(256 \times 4) \times 1/4560 = 224.56$ MB/seg

- **Acesso ao barramento**
 - **Single bus master → processador**
 - **Leitura de memória e escrita no disco**



- **Multiple bus master → arbitração de barramento**
- **Sinais : bus request e bus grant**
- **Daisy Chain**



Vantagem → simplicidade

Desvantagem → não assegura que o dispositivo de menor prioridade seja atendido (fairness)

- **Centralizado, arbitração paralela**
 - **Linhas de request múltiplas e independentes para cada dispositivo. Necessita de um arbitrador central, que pode vir a ser o gargalo do sistema. Ex. PCI**

- **Arbitração distribuída por auto seleção**
 - **Linhas de request múltiplas. Cada dispositivo tem um código. Assim que o master recebe um pedido, ele começa a gerar códigos em ordem de prioridade. O dispositivo que pediu o barramento lê os códigos até que o seu seja gerado. Assim que for gerado, ele assume o barramento. Ex. NuBus**
- **Arbitração distribuída por detecção de colisões**
 - **Múltiplos request de barramentos geram colisões. Esquema para selecionar entre colisões é usado. Ex. Ethernet.**

- **Resumo**

Opção	Alto desempenho	Baixo custo
Largura de barramento	Linhas de endereço e dados separados	Linhas de endereço dados multiplexados
Largura de dados	Grande número de bits é rápido	Pequeno número de bits é barato
Tamanho da transferência	Múltiplas palavras → menos overhead	Palavra única → simples
Bus masters	Multiple masters → necessita arbitração	Single master → não necessita arbitração
clocking	Síncrono	Assíncrono

- Padrões de barramentos

Características	PCI	SCSI
Tipo de barramento	Backplane	I/O
Largura de barramento	32-64	8-32
Bus Endereço/dado multiplexado ?	Multiplexado	Multiplexado
Número de bus masters	Múltiplos	Múltiplos
Arbitração	Centralizado, arbitração paralela	Self-selection
Clocking	Síncrono 33-66 MHz	Assíncrono ou síncrono (5-10 Mhz)
Bandwidth máximo teórico	133-512 MB/Seg	5-40 MB/seg
Bandwidth típico	80 MB/Seg	2.5-40 MB/seg (sinc) 1.5 MB/seg (assinc)
Número máximo de dispositivos	1024 (com múltiplos segmentos – 32 dispositivos por segmento)	7-31 (bus width –1)
Comprimento máximo de barramento	0.5 metros	25 metros
Nome	PCI	ANSI X3.131

- **Dispositivos de I/O, memória, processador e sistema operacional**
 - **Protocolo de barramento → define como uma palavra ou um bloco é enviado através de um conjunto de fios (barramento).**
 - **Sistema Operacional → age como interface entre o programa que requisitou I/O e o hardware.**
- **Operação de I/O**
 - **Para fazer uma operação de I/O, o processador deve endereçar o dispositivo e enviar-lhe uma ou mais palavras de comando.**
 - **Endereçamento → Memory-Mapped I/O e instruções especiais de I/O**
- **Dispositivo têm que comunicar-se com o processador**
 - **Polling → O processador checka o status através da leitura de um registrador onde o dispositivo põe o informações sobre seu estado → desvantagem ?**

Exemplo – Overhead usando polling

Determinar o impacto do overhead do polling para três dispositivos diferentes. Assumir que o número de ciclos de clock para operação de polling (transferência para a rotina de polling, acesso ao dispositivo e restauração do programa) é 400 e que o processador tem uma frequência de 500 MHz.

Determine a fração do tempo de CPU consumido, para os seguintes casos, assumindo que os dispositivos estão sempre ocupados.

- 1. O mouse deve ser checado 30 vezes por segundo para assegurar a não perda de qualquer movimento do usuário.**
- 2. Um floppy disk transfere dados o processador em unidades de 16 bits e tem uma taxa de transferência de 50 KB/seg. Nenhuma transferência pode ser perdida.**
- 3. O hard disk transfere dados em blocos de 4 palavras e pode transferir a 4 MB/seg. Nenhuma transferência pode ser perdida.**

Solução

1. Mouse

**Ciclos de clock para polling = $30 \times 400 = 12.000$
ciclos por segundo**

**Fração dos ciclos de clock do processador
consumido = $12 \times 10^3 / 500 \times 10^6 = 0,002\%$**

**Polling pode ser usado para o mouse sem muito
impacto ao processador.**

2. Floppy disk

**Taxa de polling = $(50 \text{ KB/seg}) / (2\text{B/polling access}) =$
 $25\text{K polling access/seg}$**

**Ciclos por segundo para polling = $25\text{K} \times 400 \sim 10$
 $\times 10^6$ ciclos por segundo**

**Fração dos ciclos de clock do processador
consumido = $10 \times 10^6 / 500 \times 10^6 = 2\%$**

Overhead significativo, mas tolerável.

3. Hard disk

**Taxa de polling = $(4\text{MB/seg}) / (16\text{B/polling access}) =$
 $250\text{K polling access/seg}$**

**Ciclos por segundo para polling = $250\text{K} \times 400$
 $\sim 100 \times 10^6$ ciclos por segundo**

**Fração dos ciclos de clock do processador
consumido = $100 \times 10^6 / 500 \times 10^6 = 20\%$**

Polling é inaceitável para hard disk.

- **I/O por interrupção → dispositivo interrompe a CPU para dizer que está pronto para a operação de I/O → vantagens e desvantagens ????**

Exemplo – Overhead usando interrupção

Supor o mesmo hard disk e o mesmo processador usado no exemplo anterior, só que usando o esquema de interrupção. O overhead para cada transferência, incluindo a interrupção é de 500 ciclos de clock. Encontre a fração do processador consumido se o hard disk está transferindo dados apenas 5% do tempo.

Solução

Ciclos de clock por segundo = $250K \times 500 = 125 \times 10^6$ ciclos por segundo

Fração dos ciclos de clock do processador consumido durante a transferência = $125 \times 10^6 / 500 \times 10^6 = 25\%$

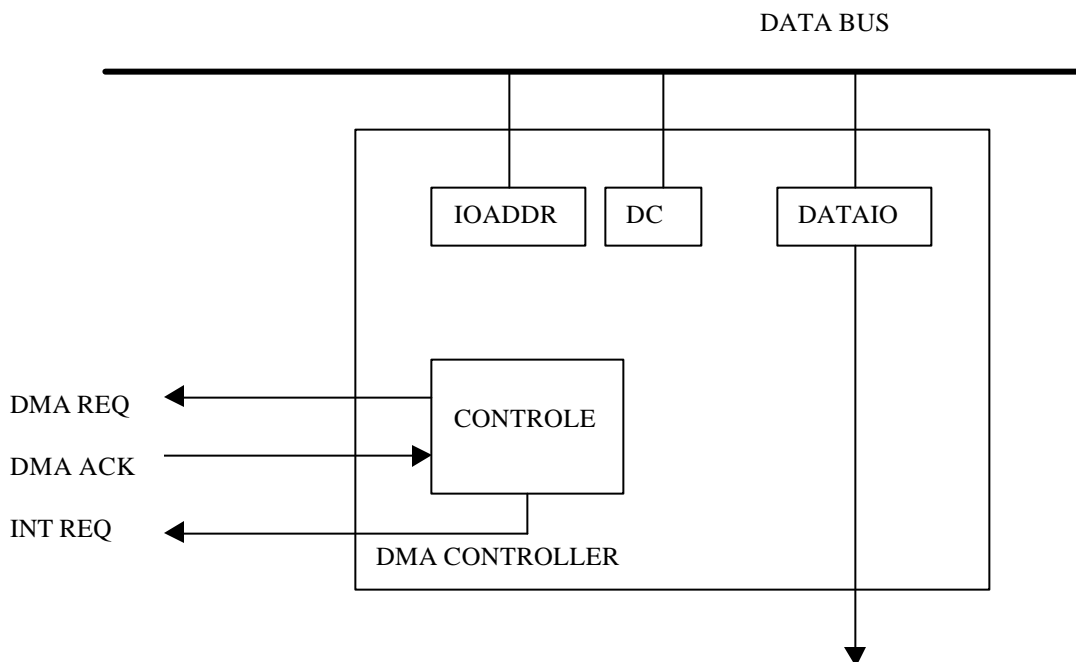
Assumindo que o disco só transfere dados 5% do tempo → Fração dos ciclos de clock do processador consumido na média = $25\% \times 5\% = 1.25\%$

- **Direct Access Memory – DMA → disco transfere milhares de dados → um controlador especializado para controlar a transferência → DMA controller.**

- **Passos da transferência DMA:**

1. **O processador inicializa o DMA enviando o endereço do dispositivo, o endereço de memória e o número de bytes a ser transmitido.**
2. **O controlador DMA pede o barramento e inicia a operação.**
3. **A operação termina quando não há mais bytes a serem transferidos, através de uma interrupção.**

- **Controlador DMA**



- **Tipos de DMA**
 - **Por blocos**
 - **Cycle stealing**
 - **Transparente**

Exemplo – Overhead usando DMA

Para o mesmo processador e mesmo hard disk, assumir que o setup inicial da transferência DMA leva 1000 ciclos de clock para o processador, e assumir que o tratamento da interrupção de fim de transferência DMA necessita de 50 ciclos de clock para o processador. O hard disk tem taxa de transferência de 4 MB/seg e usa DMA. Se a média de transferência do disco é de 8KB, que fração de um processador de 500 MHz é consumido se o disco transfere em 100% do tempo. Ignore o impacto o concorrência de uso do barramento pelo processador e pelo DMA controller.

Solução

Cada transferência DMA leva $8\text{KB} / (4\text{MB/seg}) = 2 \times 10^3 \text{ seg}$

Se o disco está constantemente transferindo $\rightarrow (1000 + 500) \text{ ciclos/transferência} / 2 \times 10^3 \text{ seg/transferência} = 750 \times 10^3 \text{ ciclos de clock/seg}$

Processador de 500 MHz $\rightarrow \text{fração consumida} = 750 \times 10^3 / 500 \times 10^6 = 1.5 \times 10^{-3} = 0,2\%$

- **Projeto de um sistema de I/O**
 - **Requisitos de latência**
 - **Requisitos de bandwidth**
 - **O que fazer ?**
 - **Encontrar o caminho crítico do sistema de I/O → depende do workload, processador, memória, barramento e controladores de I/O ou dispositivos.**
 - **Configurar estes componentes para atender ao bandwidth desejado.**
 - **Determinar os requisitos do resto do sistema para suportar este bandwidth.**

Exemplo

Considere o seguinte sistema de computação:

- ◆ **Uma CPU de 300 milhões de instruções por segundo e em média 50.000 instruções do SO por operações de I/O.**
- ◆ **Um barramento de memória backplane capaz de sustentar uma taxa de transferência de 100 MB/seg.**
- ◆ **Um controlador SCSI-2 com taxa de transferência de 20 MB/seg e que acomoda até 7 discos.**
- ◆ **Disk drives com bandwidth de leitura/escrita de 5 MB/seg uma tempo médio de seek + latência rotacional de 10 ms.**

Se o workload consiste de leitura de 64 KB (onde o bloco está seqüencial na trilha) e o programa do usuário necessita de 100.000 instruções por operação de I/O, encontre a taxa máxima de I/O e o número de discos e

controladores SCSI necessários. Assumir que as leituras são sempre feitas em discos disponíveis (sem conflito de discos).

Solução

Os dois componentes fixos do sistema são o barramento de memória e a CPU. Vamos encontrar a taxa de I/O que este dois componentes podem sustentar e determinar qual deles é o gargalo. Cada I/O leva 100.000 instruções de usuário e 50.000 do SO →

Taxa máxima de I/O = taxa de instruções executadas / instruções de I/O = $300 \times 10^6 / (50 + 100) \times 10^3 = 2000$ I/O'S por segundo.

Cada I/O transfere 64 KB →

Taxa máxima de I/O do bus = bandwidth do bus / bytes por I/O = $100 \times 10^6 / 64 \times 10^3 = 1562$ I/OS por segundo.

Portanto o barramento é o gargalo, então devemos configurar os sistema no nível determinado pelo barramento.

Vamos determinar quantos discos precisamos para acomodar 1562 I/OS por segundo. Para isto vamos primeiro determinar o tempo por operação de I/O no disco.

Tempo por I/O no disco = tempo de seek/ rotacional + tempo de transferência = $10 \text{ ms} + 64 \text{ KB} / (5 \text{ MB/seg}) = 22.8 \text{ ms}$ → cada disco pode então completar → $1/22.8 \text{ ms} = 43.9 \text{ I/OS}$ por segundo. Para saturar o bus → $1562/43.9 \sim 36$ discos.

Para calcular o número de barramentos SCSI, necessitamos saber a taxa média de transferência por disco →

Taxa de transferência = Tamanho da transferência / tempo da transferência = 64 KB / 2.8 ms = 2,74 MB/seg.

Como 2.74 MB/seg < 20 MB/seg, podemos utilizar todo o bandwidth dos buses → precisaremos de 36/7 ~ 6 SCSI buses and controller.

- Um sistema de I/O típico → Apple Machintosh 7200

