



## Organização de Computadores

(Aula 5)

### Entrada/Saída

## Entrada e Saída (*Input/Output – I/O*)

- Computador:
  - Processador,
  - Memória (principal e **secundária**)
  - Equipamentos (dispositivos, elementos, unidades, ...) de **Entrada/Saída (E/S ou I/O)**
- Dispositivos de E/S: Periféricos do Computador
- Função geral:
  - Servem para a comunicação do computador com o meio externo.
- **Entrada:** entrada (inserção) de dados por meio de algum código (ex: programa).
  - Ex: teclado, mouses, scanners, leitoras óticas, leitoras de cartões magnéticos, câmeras de vídeo, microfones, sensores, transdutores, etc
- **Saída:** retorno de dados, como resultado de alguma operação de algum programa.
  - Impressoras, monitores de vídeo, plotters, atuadores, chaves, etc ...

## Diferentes Tipos de Dispositivos (1)

- Diferentes funções
  - Comunicação do usuário com o computador
  - Comunicação do computador com o meio ambiente (dispositivos externos a serem monitorados ou controlados)
  - Armazenamento (gravação) de dados.
- Taxa de transferência de dados varia muito para cada dispositivo de I/O
  - Muito lento: ex. teclado, mouse
  - Muito rápido: ex. gráficos escritos no monitor, discos, interfaces de redes

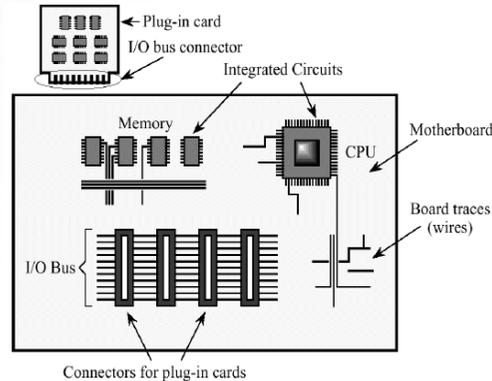
## Diferentes Tipos de Dispositivos (2)

- Taxa de transferência de dados varia muito para cada dispositivo de I/O
  - Muito lento: ex. teclado, mouse
  - Muito rápido: ex. gráficos escritos no monitor, discos, interfaces de redes

Device	Behavior	Partner	Data rate (KB/sec)
Keyboard	Input	Human	0.01
Mouse	Input	Human	0.02
Voice input	Input	Human	0.02
Scanner	Input	Human	200.00
Voice output	Output	Human	0.60
Line printer	Output	Human	1.00
Laser printer	Output	Human	100.00
Graphics display	Output	Human	30,000.00
Network-terminal	Input or output	Machine	0.05
Network-LAN	Input or output	Machine	200.00
Floppy disk	Storage	Machine	50.00
Optical disk	Storage	Machine	500.00
Magnetic tape	Storage	Machine	2000.00
Magnetic disk	Storage	Machine	2000.00

## Entrada e Saída (*Input/Output – I/O*) (2)

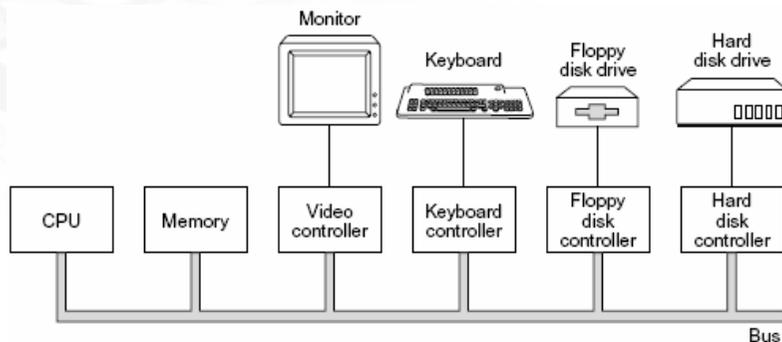
- Geralmente a estrutura física de um PC é composta de uma caixa de metal com uma placa de circuito impresso chamada **placa-mãe**



## Entrada e Saída (*Input/Output – I/O*) (3)

- Na placa-mãe está
  - a CPU
  - alguns slots para inserção de pentes de memória (ex. módulos DIMM)
  - e vários outros chips
- Ela também tem um **barramento** percorrendo todo o seu comprimento, assim como **soquetes** nos quais serão ligados conectores das **placas controladoras** dos dispositivos de entrada e saída.

## Entrada e Saída (*Input/Output – I/O*) (4)

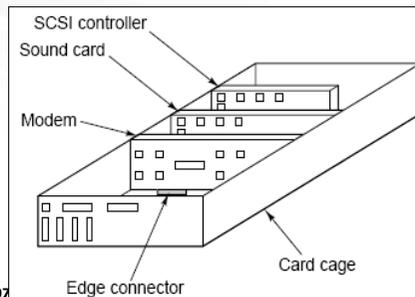


## Controladoras de E/S (1)

- Vários dispositivos de I/O distintos podem fazer parte ou estarem conectados a um processador.
- A CPU não se comunica diretamente com cada dispositivo de E/S e sim com "interfaces"
  - Interface de E/S =
    - Adaptador de Periférico,
    - Controladora de E/S, Processador de Periférico,
    - Canal de E/S, etc.
  - Compatibilizar as diferentes características.
- Circuitos controladores de I/O são usados para permitir a comunicação entre o processador e os dispositivos.

## Controladoras de E/S (2)

- Controladora: seu trabalho consiste em controlar seu dispositivo de E/S e em tratar o acesso do dispositivo ao barramento.
- Placa ligada a um slot livre, ou inserida diretamente na placa-mãe



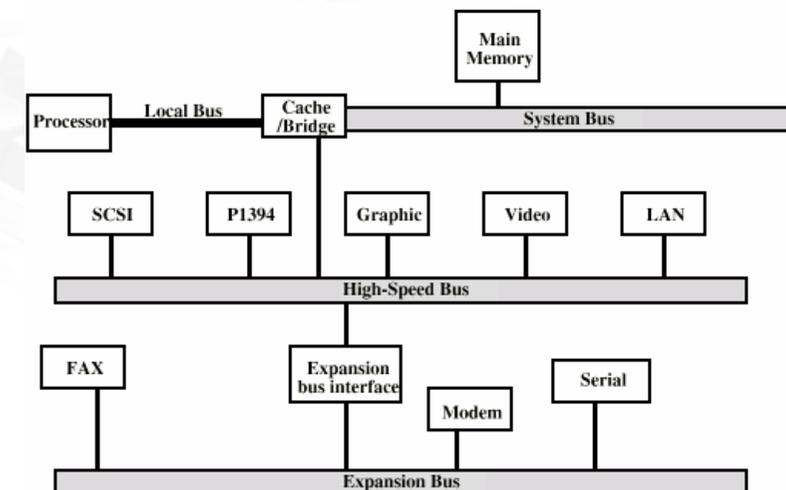
## Controladoras de E/S (3)

- Processo geral de comunicação
  1. Inicialmente, a CPU interroga o dispositivo, enviando o endereço do dispositivo e um sinal dizendo se quer mandar ou receber dados através da controladora
  2. A controladora, reconhecendo seu endereço, responde quando está pronta para receber (ou enviar) os dados.
  3. A CPU então transfere (ou recebe) os dados através da controladora
  4. A controladora responde confirmando que ...
    - ... recebeu (ou transferiu) os dados (*acknowledge* ou *ACK*)
    - ... ou que não recebeu os dados, neste caso solicitando retransmissão (*not-acknowledge* ou *NAK*).

## Barramentos de E/S (1)

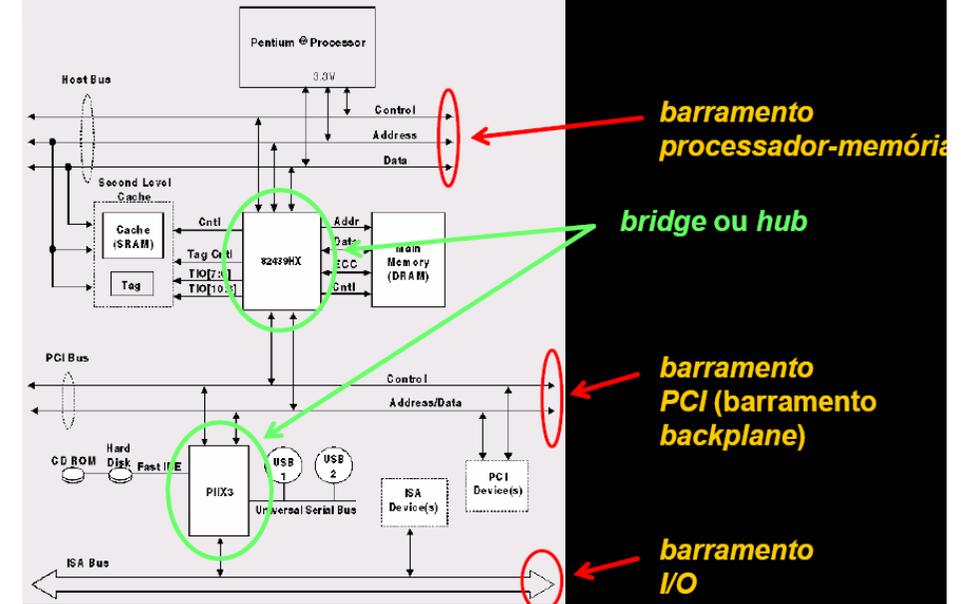
- Além dos barramentos de sistema (CPU-Memória), temos os barramentos de E/S
- Barramentos do tipo entrada/saída
  - São mais longos
  - Podem ter muitos dispositivos conectados a ele
  - Muitas vezes necessitam atender uma ampla faixa de bandas passante
  - Não necessariamente têm uma interface direta com a memória
    - Podem usar um dos outros para se comunicar com a memória
- Pode haver mais de 1 barramento
  - Ex: Um de alta velocidade, um de baixa velocidade
  - Diferentes padrões de E/S

## Barramentos de E/S (2)



## Barramentos de E/S (3)

- Barramentos do tipo "backplane"
  - Nome devido ao local de construção (no "backplane", estrutura onde as placas são fixadas no chassi da máquina)
  - Projetado de modo que processadores, memórias e dispositivos de entrada/saída possam ser interconectados em um único barramento
  - Tanto pode ser usado para transportar toda a informação dentro do computador (caso típico dos primeiros PCs)...
    - ... como pode ainda funcionar de interface entre o barramento processador-memória e os barramentos de I/O.

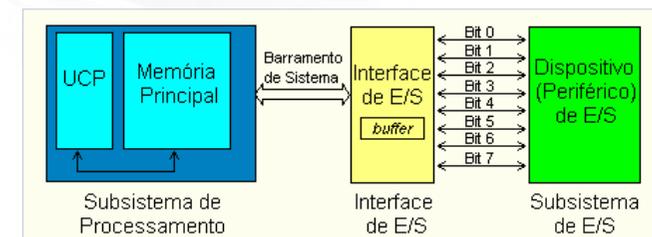


## Formas de Comunicação

- Atividades de E/S são assíncronas
  - *Handshaking* pode ser necessário
    - Quando o dispositivo está pronto para recepção/envio de dados
- Qualidade dos dados pode ser incerta
  - Mecanismos de detecção/correção
- Transferências podem ser interrompidas
  - Ex. impressora sem papel
- *Device drivers* devem tratar estes problemas

## Formas de Comunicação - Comunicação em Paralelo (1)

- Grupos de bits transferidos simultaneamente (em geral, byte a byte) através de diversas linhas condutoras dos sinais.
  - taxa de transferência de dados ("*throughput*") é alta
- Por que não usamos sempre comunicação paralela?

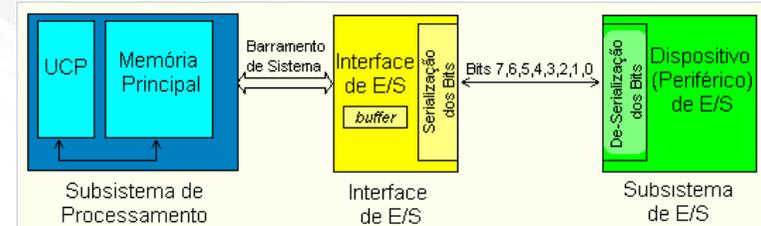


## Formas de Comunicação - Comunicação em Paralelo <sup>(2)</sup>

- Processo de transferência em paralelo envolve um controle sofisticado e complexo
  - Mais caro!
- Exemplo de problema
  - Propagação dos sinais no meio físico: se fazer de modo que os sinais (os bits) correspondentes a cada *byte* cheguem simultaneamente à extremidade oposta do cabo
  - Condutores que compõem o cabo usualmente terão pequenas diferenças físicas => a velocidade de propagação dos sinais poderá ser ligeiramente diferente nos diversos fios
  - Um determinado bit  $x$  em cada *byte* se propague mais rápido e chegue à extremidade do cabo antes que os outros  $n-1$  bits do *byte*.
  - Este fenômeno é chamado *skew*, e as consequências são catastróficas: os bits  $x$  chegariam fora de ordem (os *bytes* chegariam embaralhados) e a informação ficaria irrecuperável.
- A comunicação em paralelo se limita a aplicações que demandem altas taxas de transferência
  - unidades de disco, ou que demandem altas taxas de transferência, como CD-ROM, DVD, ou mesmo impressoras

## Formas de Comunicação - Comunicação Serial

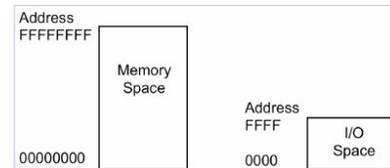
- Os bits são transferidos um a um, através de um único par condutor
- O controle é comparativamente muito mais simples que no modo paralelo
  - Implementação mais barata.
- Muito usada para periféricos mais lentos
  - Teclado, *mouse*, etc.



A transmissão serial tem recebido aperfeiçoamentos importantes (seja de protocolo, de *interface* e de meio de transmissão) que vem permitindo o aumento da velocidade de transmissão -> *USB - Universal Serial Bus*

## Endereçamento de Entrada e Saída

- Controladores possuem registradores para comunicação com a CPU
  - E/S é realizada escrevendo-se comandos nos registradores da controladora de E/S
- Endereçamento de entrada e saída
  - Espaços de endereçamento separados (Arq. Pentium)
    - Linha do barramento M/IO
    - Opcodes separados
      - IN  $reg_d$ ,  $[end_{16}]$ , OUT  $[end_{16}]$ ,  $reg_d$
  - E/S mapeada em memória
    - Portas E/S operadas como posições de memória
    - Determinados endereços de memória correspondem fisicamente a dispositivos de E/S



## Metodologias de Comunicação <sup>(1)</sup>

- Entrada e saída programada (ou polling)
- Entrada e saída por interrupção
- Acesso direto à memória (DMA)

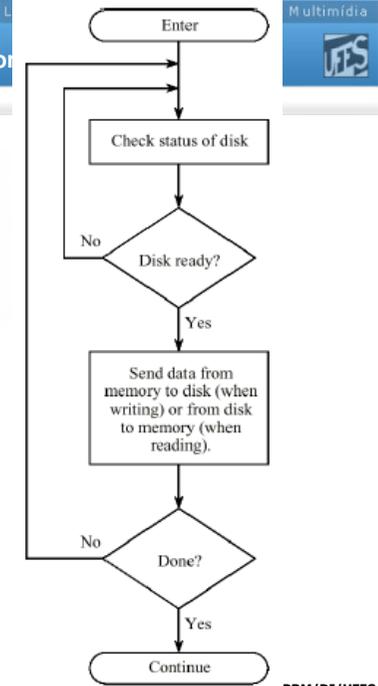
## Metodologias de Comunicação (2)

### ■ Entrada e saída programada (ou polling)

- Forma mais simples de E/S
- Frequentemente utilizada em sistemas de pequeno porte ou dedicados
- A CPU fica dedicada a gerenciar as transferências de dados
- Desvantagem
  - A CPU gasta a maior parte do tempo em um loop de espera (*busy waiting*) até que o dispositivo de entrada esteja pronto para a transferência

## Metodologias de Comunicação (3)

### Fluxograma de E/S Programada (polling) para uma transferência de disco



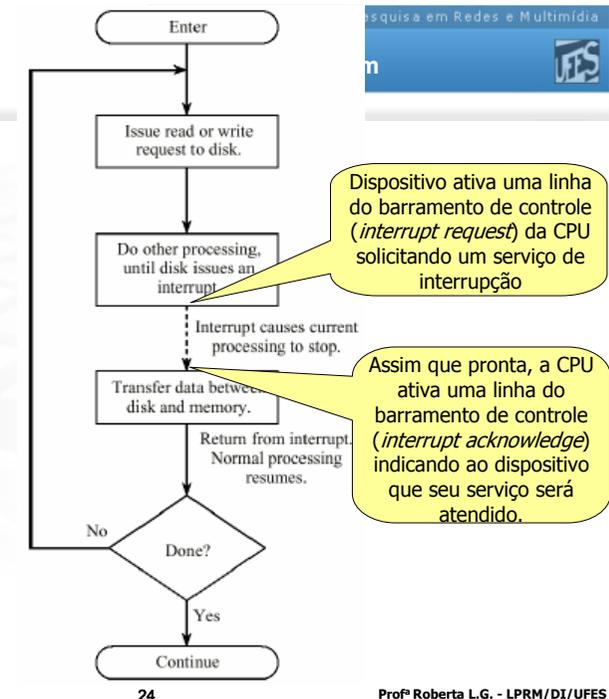
## Metodologias de Comunicação (4)

### ■ Entrada e saída por interrupção

- O dispositivo E/S solicita o serviço de transferência apenas quando este estiver pronto para enviar/receber o(s) dado(s)
- A interrupção causa um desvio no fluxo do programa via hardware
  - A interrupção para o programa que está rodando e desvia o fluxo para a **Rotina de Tratamento da Interrupção**
- Rotina de Tratamento da Interrupção
  - Software que realiza o serviço de transferência daquele dispositivo externo que solicitou a interrupção.
  - Verifica a ocorrência de possíveis erros na transferência de dados
- Quando finaliza a rotina de tratamento da interrupção, o processador retoma a execução do programa interrompido

## Metodologias de Comunicação (5)

### Fluxograma de E/S controlada por interrupção para uma transferência de disco



## Metodologias de Comunicação (64)

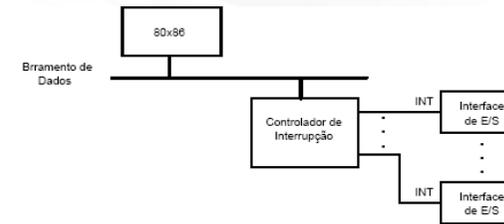
### ■ Entrada e saída por interrupção (cont.)

- É comum existirem várias interfaces diferentes que fazem pedidos de interrupção ao processador
- Cada interface deve ser atendida por uma rotina de serviço de interrupção específica
  - Assim, ao receber um pedido de interrupção, o processador deve determinar qual a rotina de serviço a ser executada
  - Além disso, quando duas ou mais interfaces fazem pedidos de interrupção simultâneos, é necessário decidir qual o pedido de interrupção que será atendido
- Estas duas funções são suportadas por um componente chamado controlador de interrupção (interrupt controller).

## Metodologias de Comunicação (8)

### ■ Entrada e saída por interrupção (cont.)

- O sinal de interrupção de cada interface é ligado ao controlador de interrupção.
  - Um número e uma prioridade são atribuídos a cada um destes sinais
- Quando um pedido de interrupção acontece, o controlador de interrupção envia para o processador, através do barramento de dados, o número do pedido.

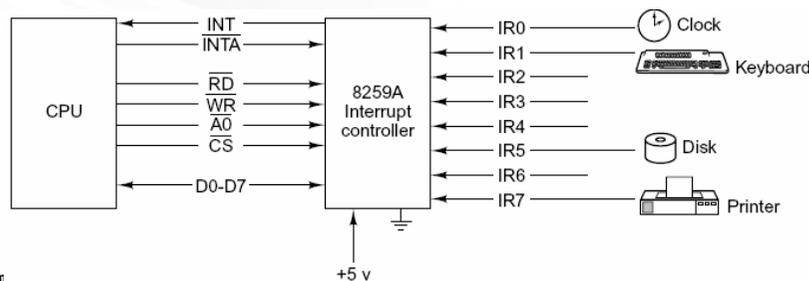


- No caso de dois ou mais pedidos simultâneos, o controlador decide qual é o pedido com maior prioridade e envia para o processador o número correspondente.

## Entrada e Saída (Input/Output – I/O) (16)

### ■ Interrupções na família 80x86

- 8259A permite até 8 interrupções com prioridade (IRQ0 até IRQ7)
  - Se for necessário controlar mais do que 8 dispositivos?  
R: Os chips 8259A podem ser ligados em cascata INT: Pedido de interrupção
- INTA: Reconhecimento de interrupção



## Entrada e Saída (Input/Output – I/O) (17)

### ■ Interrupções na família 80x86 (cont.)

- Quando um dispositivo necessita efetuar alguma operação ele envia um sinal IRx
- 8259A ativa a linha INT indicando uma interrupção ao processador
- Quando pronto, o processador envia um sinal (INTA) de volta ao 8259A
- Ao receber o sinal, o controlador coloca um número inteiro no barramento de dados identificando o tipo de dispositivo
- A CPU então usa este número para indexar uma tabela de 256 entradas (vetor de interrupções) para encontrar o endereço da rotina de tratamento de interrupção a ser executada
- Se uma interrupção subsequente ocorrer:
  - De maior prioridade: o 8259A interrompe a CPU pela segunda vez
  - Prioridade menor, ela é suspensa até a primeira terminar
    - Para isto, uma rotina de interrupção deve enviar explicitamente um comando para o 8259A para informar quando ela terminar

## Entrada e Saída (*Input/Output – I/O*) (18)

- Uma controladora que lê ou escreve dados da/na memória sem que seja necessária a intervenção do processador utiliza o mecanismo de **Acesso Direto à Memória**
- DMA (*Direct Memory Access*)**
  - Libera a CPU do intermédio entre dispositivo e memória
  - Uma transferência por DMA essencialmente copia um bloco de memória de um dispositivo para outro
  - A CPU inicia a transferência, mas não executa a transferência
- Third party DMA
  - A transferência é realizada pelos controladores DMA que são tipicamente parte do chipset da placa mãe

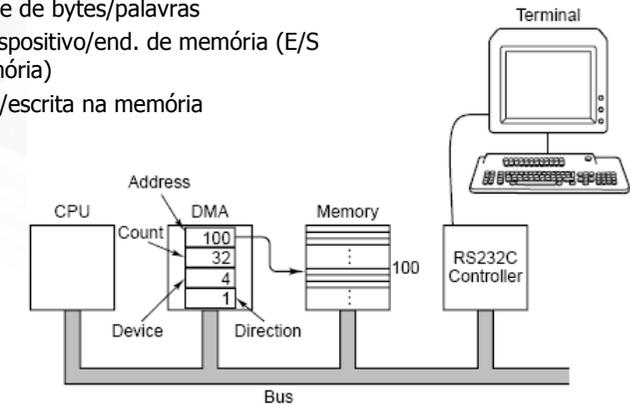
## Entrada e Saída (*Input/Output – I/O*) (19)

**Address:** end. de memória a ser lido/escrito

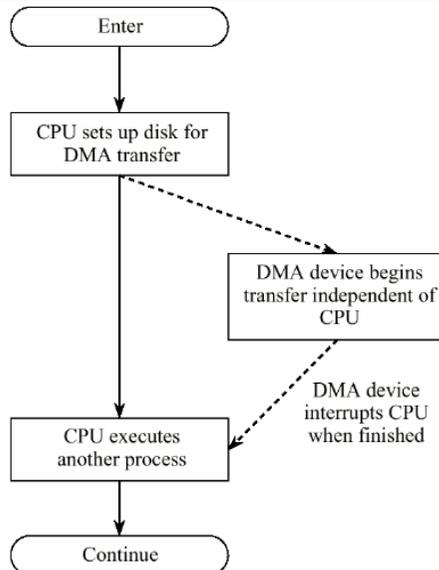
**Count:** quantidade de bytes/palavras

**Device:** no. do dispositivo/end. de memória (E/S mapeada em memória)

**Direction:** leitura/escrita na memória



## Entrada e Saída (*Input/Output – I/O*) (20)



Quando termina a transferência dos dados, o controlador DMA força uma **interrupção**, fazendo com que o processador suspenda a execução do programa corrente (**processo**), e inicie a execução da respectiva **rotina de Tratamento da Interrupção**.

## Entrada e Saída (*Input/Output – I/O*) (21)

- Em geral, é possível ter vários dispositivos de e/s operando com a técnica de acesso direto à memória
- Para tanto, o controlador de DMA possui várias entradas para pedido de DMA
- O controlador de DMA associa a cada uma destas entradas um conjunto independente de registradores
  - Para armazenar o número de dados a serem transferidos, o endereço inicial e o sentido da transferência
- Um grupo de sinais de controle com seus respectivos registradores formam o chamado **canal de DMA**
- O controlador de DMA se encarrega de arbitrar entre dispositivos que fazem pedidos de DMA simultâneos, usando um esquema de prioridades atribuídas aos canais de DMA.

## Referências

- Andrew S. Tanenbaum, **Organização Estruturada de Computadores**, 5ª edição, Prentice-Hall do Brasil, 2007.
- John L. Hennessy and David A. Patterson, **Arquitetura de Computadores: Uma Abordagem Quantitativa**. 3ª edição. Editora Campus, 2003.
- D. A. Patterson & J. L. Hennessy. **Organização e Projeto de Computadores: a interface hardware/software**. Livros Técnicos e Científicos Editora, Inc. Rio de Janeiro, RJ. Tradução da Segunda Edição americana, 2000.