

# Arquitetura e Organização de Computadores

## Capítulo 7

### Entrada/saída

slide 1

© 2010 Pearson Prentice Hall. Todos os direitos reservados.

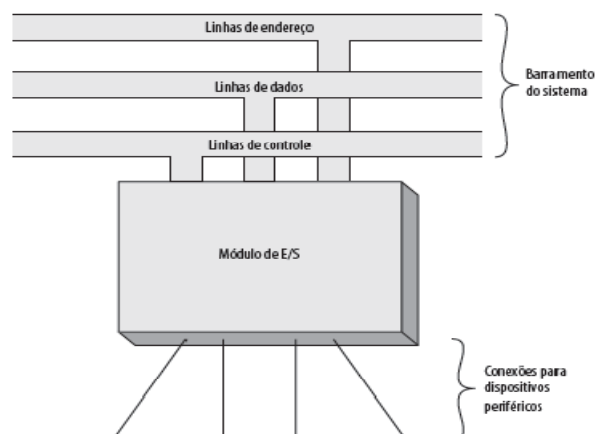
### Problemas de entrada/saída

- Grande variedade de periféricos:
  - Entregando diferentes quantidades de dados.
  - Em velocidades diferentes.
  - Em formatos diferentes.
- Todos mais lentos que CPU e RAM.
- Precisa de módulos de E/S.

## Módulo de entrada/saída

- Interface com CPU e memória.
- Interface com um ou mais periféricos.

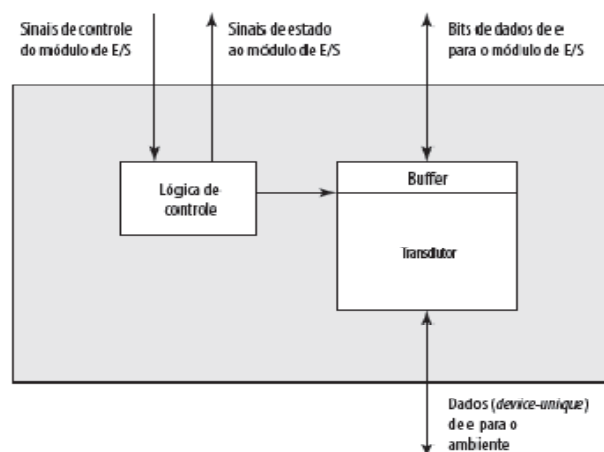
## Modelo genérico de módulo de E/S



## Dispositivos externos

- Legíveis ao ser humano:
  - Monitor, impressora, teclado.
- Legíveis à máquina:
  - Monitoração e controle.
- Comunicação:
  - Modem.
  - Placa de interface de rede (NIC).

## Diagrama em blocos de um dispositivo externo



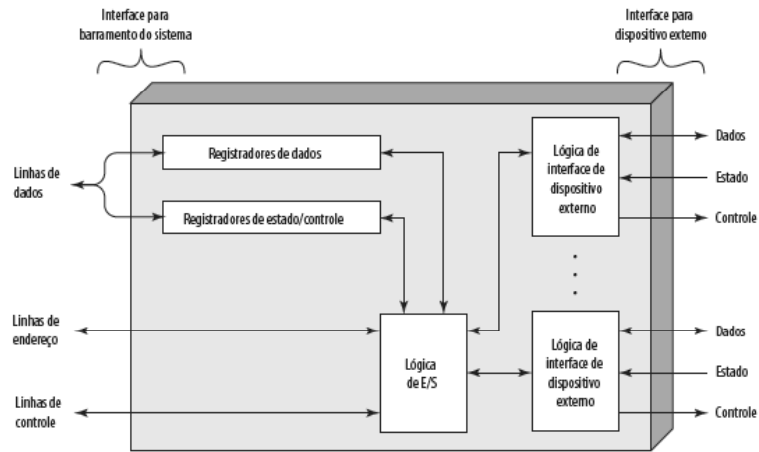
### Função do módulo de E/S

- Controle e temporização.
- Comunicação com CPU.
- Comunicação com dispositivo.
- Buffering de dados.
- Detecção de erro.

### Etapas da E/S

- CPU verifica estado do módulo de E/S.
- Módulo de E/S retorna o estado.
- Se estiver pronto, CPU solicita transferência de dados.
- Módulo de E/S recebe dados do dispositivo.
- Módulo de E/S transfere dados à CPU.
- Variações para saída, DMA etc.

## Diagrama do módulo de E/S



## Decisões do módulo de E/S

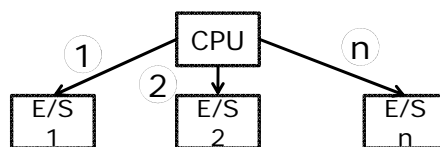
- Ocultar ou revelar propriedades do dispositivo à CPU.
- Admitir dispositivo múltiplo ou único.
- Controlar funções do dispositivo ou sair para CPU.
- Também decisões do SO.
  - P.e., Unix trata de tudo o que pode como arquivo.

## Técnicas de E/S

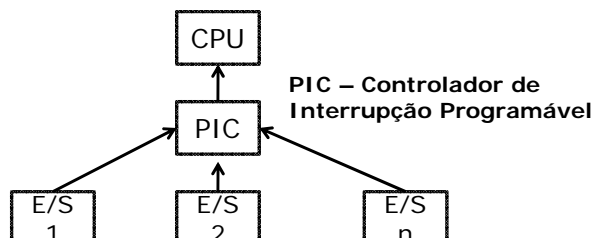
- Programada (**Polling**).
- Controlada por interface (**Interrupção**).
- Acesso direto à memória (**DMA**).

## Técnicas de E/S

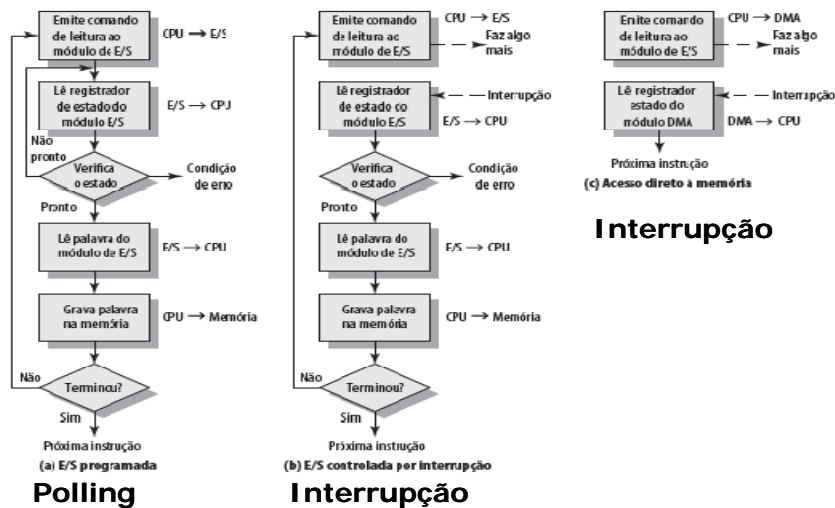
### Polling



### Interrupção



## Três técnicas para entrada de um bloco de dados



## E/S programada

- CPU tem controle direto sobre E/S:
  - Conhecendo o estado.
  - Comandos de leitura/escrita.
  - Transferindo dados.
- CPU espera que módulo de E/S termine a operação.
- **Desperdiça tempo de CPU.**

### E/S programada – detalhe

- CPU solicita operação de E/S.
- Módulo de E/S realiza operação.
- Módulo de E/S altera bits de estado.
- CPU verifica bits de estado periodicamente.
- Módulo de E/S não informa à CPU diretamente.
- Módulo de E/S não interrompe CPU.
- CPU pode esperar ou voltar mais tarde.

### Comandos de E/S

- CPU emite endereço:
  - Identifica módulo (& disp. se >1 por módulo).
- CPU emite comando:
  - Controle – dizendo ao módulo o que fazer.
    - P.e., girar disco
  - Teste – verifica estado:
    - P.e., alimentado? Erro?
  - Leitura/escrita:
    - Módulo transfere dados via buffer de/para dispositivo.



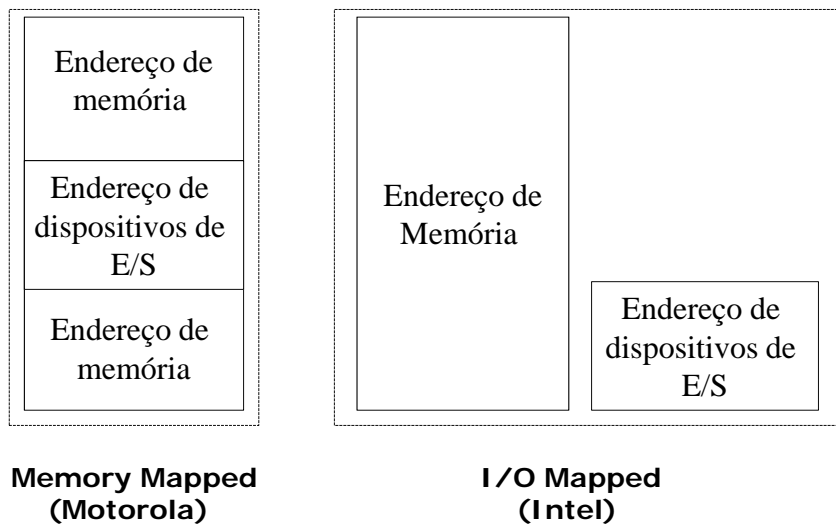
## Endereçando dispositivos de E/S

- Sob E/S programada, transferência de dados é muito semelhante ao acesso à memória (do ponto de vista da CPU).
- Cada dispositivo recebe identificador exclusivo.
- Comandos da CPU contêm identificador (endereço).

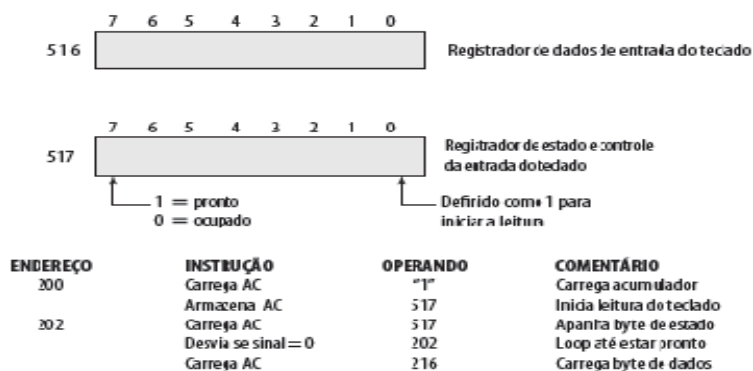
## Mapeamento de E/S

- **E/S mapeada na memória (*Memory Mapped*):**
  - Dispositivos e memória compartilham um espaço de endereços comum.
  - E/S se parece com leitura/escrita na memória.
  - Nenhum comando especial para E/S.
    - Grande seleção disponível de comandos de acesso à memória.
- **E/S independente (*I/O Mapped*):**
  - Espaços de endereços separados.
  - Precisa de linhas de seleção de E/S ou memória.
  - Comandos especiais para E/S.
    - Conjunto limitado.

## Mapeamento de E/S



## E/S mapeada na memória e independente



(a) E/S mapeada na memória

ENDEREÇO	INSTRUÇÃO	OPERANDO	COMENTÁRIO
200	Carrega E/S	5	Inicia leitura do teclado
201	Testa E/S	5	Verifica término
	Desvia se não pronto	201	Loop até estar pronto
	Entrada	5	Carrega byte de dados

(b) E/S isolada

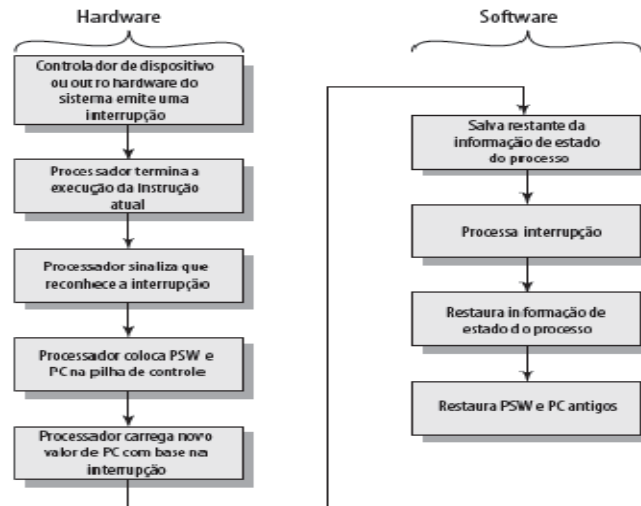
## E/S controlada por interrupção

- Contorna problema de espera da CPU.
- Sem verificação de dispositivo repetida da CPU.
- Módulo de E/S interrompe quando estiver pronto.

## E/S controlada por interrupção – Operação básica

- CPU emite comando de leitura.
- Módulo de E/S recebe dados do periférico enquanto CPU faz outro trabalho.
- Módulo de E/S interrompe CPU.
- CPU solicita dados.
- Módulo de E/S transfere dados.

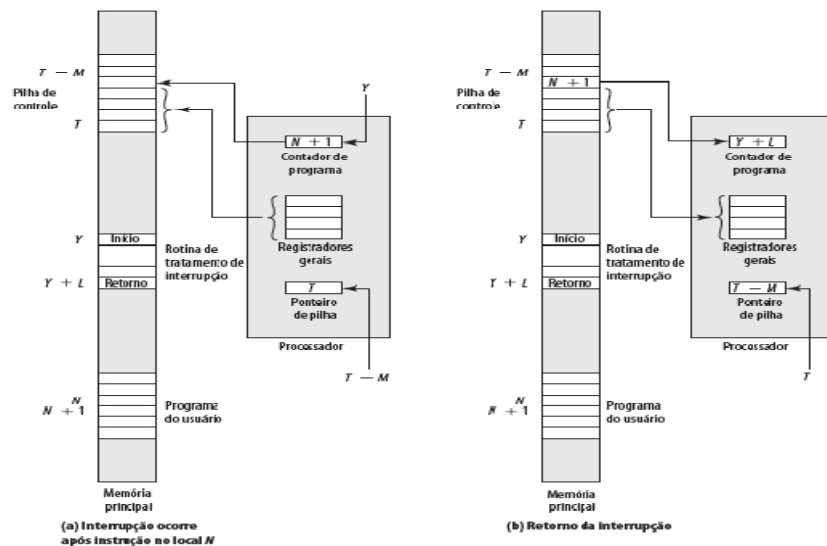
## Processamento de interrupção simples



## Ponto de vista da CPU

- Emite comando de leitura.
- Realiza outro trabalho.
- Verifica interrupção ao final de cada ciclo de instrução.
- Se interrompida:
  - Salva contexto (registradores).
  - Processa interrupção.
  - Busca dados & armazena.
- Ver notas do sistema operacional.

## Mudanças na memória e registradores para uma interrupção



## Aspectos de projeto

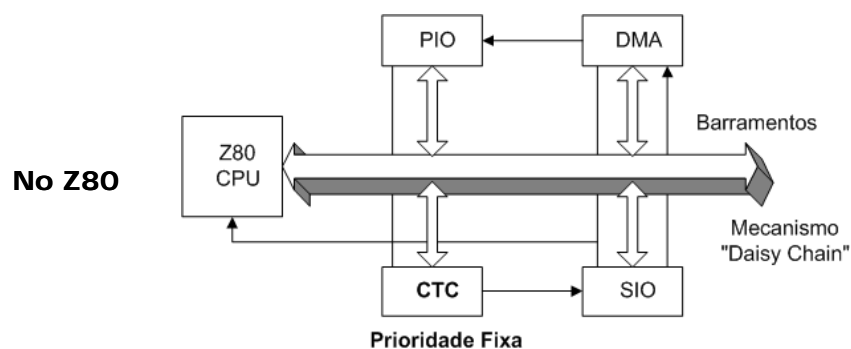
- Como identificar o módulo que emite a interrupção?
- Como lidar com interrupções múltiplas?
  - Ou seja, um tratador de interrupção sendo interrompido.

## Identificando módulo que interrompe

- Linha diferente para cada módulo:
  - PC.
  - Limita número de dispositivos.
- Verificação por software:
  - CPU verifica cada módulo por vez.
  - Lento.
- Daisy chain ou verificação por hardware.
  - Interrupt Acknowledge enviado por uma cadeia.
  - Módulo responsável coloca vetor no barramento.
  - CPU usa vetor para identificar rotina do tratador.
- Arbitração de barramento:
  - Módulo deve reivindicar o barramento antes que possa causar uma interrupção.
  - P.e., PCI & SCSI.

## Identificando módulo que interrompe

- Como lidar com interrupções múltiplas?



Z80A-CTC – Controlador de Prioridade de IRQ e Temporizador Programável  
Z80A-SIO – Controlador Serial Multiprotocolo  
Z80A-PIO – Controlador de Portas Paralelas Programável  
Z80A-DMA - Controlador de DMA

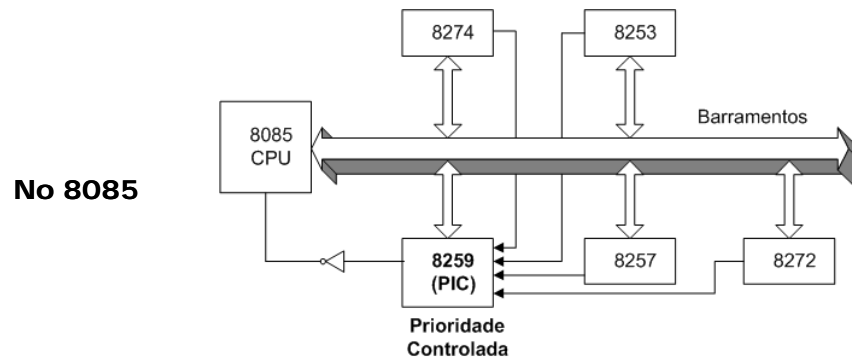
## Múltiplas interrupções

- Cada linha de interrupção tem uma prioridade.
- Linhas com prioridade mais alta podem interromper linhas com prioridade mais baixa.
- Com *bus mastering*, só o mestre atual pode interromper.

## Exemplo – Barramento do PC

- 80x86 tem uma linha de interrupção.
- Sistemas baseados no 8086 usam um controlador de interrupção 8259A.
- 8259A tem 8 linhas de interrupção.

## Exemplo – Barramento do PC



8272 – Controlador de Disco Flexível      8259 – Controlador de Prior. de IRQ  
 8274 – Controlador Serial Multiprotocolo  
 8253 – Temporizador Programável      8257- Controlador de DMA

## Sequência de eventos

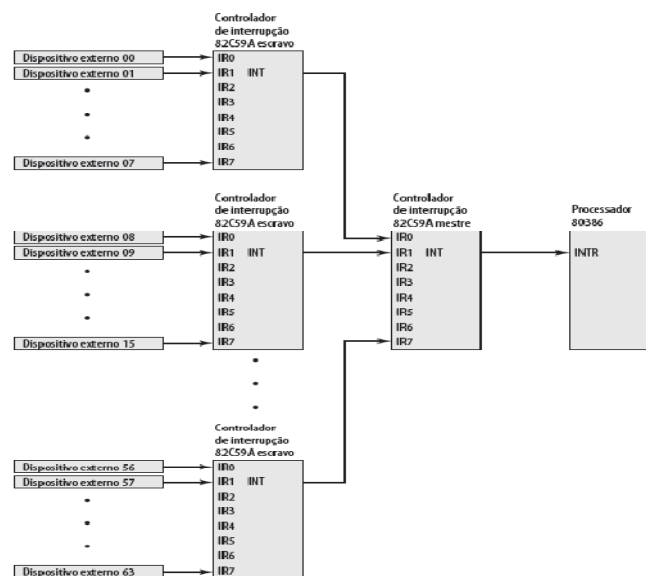
- 8259A aceita interrupções.
- 8259A determina prioridade.
- 8259A signals 8086 (levanta linha INTR).
- CPU confirma.
- 8259A coloca vetor correto no barramento de dados.
- CPU processa interrupção.



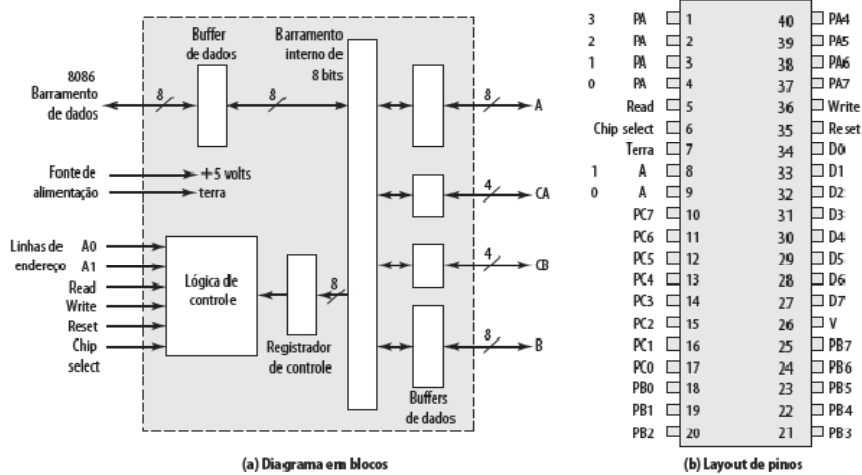
## Sistema de interrupção de barramento ISA

- Barramento ISA encadeia dois 8259As.
- Ligação é via interrupção 2.
- Gera 15 linhas:
  - 16 linhas menos uma para ligação.
- IRQ 9 é usada para redirecionar qualquer coisa tentando usar IRQ 2.
  - Compatibilidade.
- Incorporado no chip set.

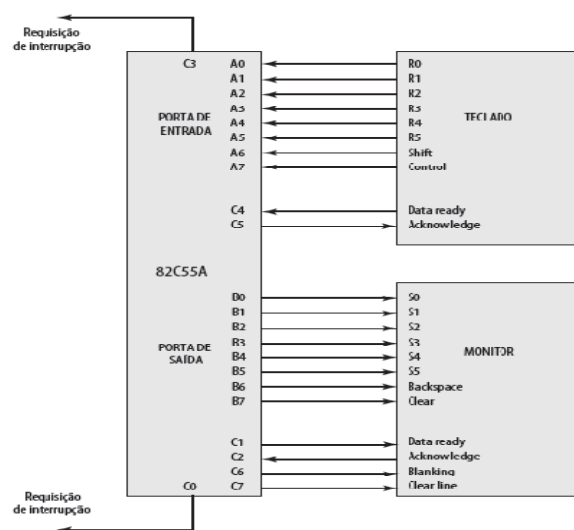
## Controlador de interrupção 82C59A



## Intel 82C55A Programmable Peripheral Interface



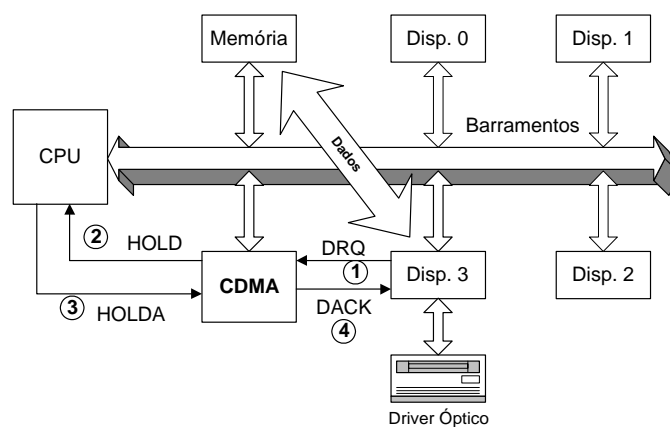
## Interface de teclado/monitor para o 82C55A



## Acesso direto à memória

- E/S controlada por interrupção e programada exige intervenção ativa da CPU.
  - Taxa de transferência é limitada.
  - CPU fica amarrada.
- DMA é a resposta.

## Acesso direto à memória

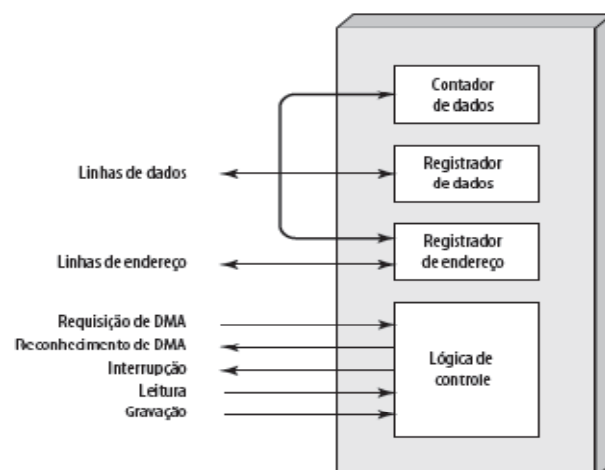


**DMA** - Dinamic Access Memory

## Função do DMA

- Módulo adicional (hardware) no barramento.
- Controlador de DMA toma o comando da CPU para E/S.

## Diagrama típico do módulo de DMA



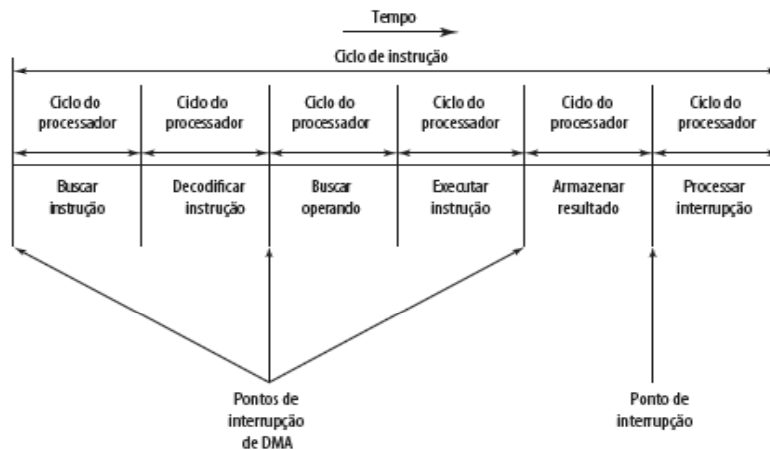
## Operação do DMA

- CPU diz ao controlador de DMA:
  - Leitura/escrita.
  - Endereço do dispositivo.
  - Endereço inicial do bloco de memória para dados.
  - Quantidade de dados a serem transferidos.
- CPU prossegue com outro trabalho.
- Controlador de DMA lida com transferência.
- Controlador de DMA envia interrupção quando terminar.

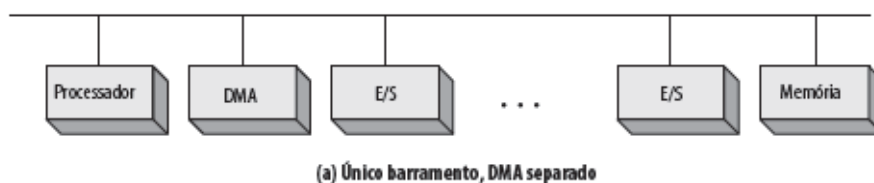
## Transferência de DMA – Roubo de ciclo

- Controlador de DMA assume o barramento por um ciclo.
- Transferência de uma palavra de dados.
- Não uma interrupção.
  - CPU não troca de contexto.
- CPU suspensa logo antes de acessar o barramento.
  - Ou seja, antes de uma busca de operando ou dados ou uma escrita de dados.
- Atrasa a CPU, mas não tanto quanto a CPU fazendo transferência.

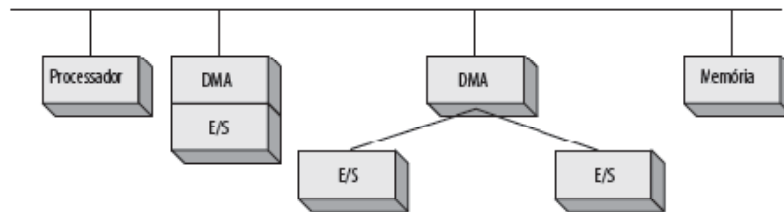
## DMA e pontos de interrupção durante um ciclo de instrução



## Configurações de DMA

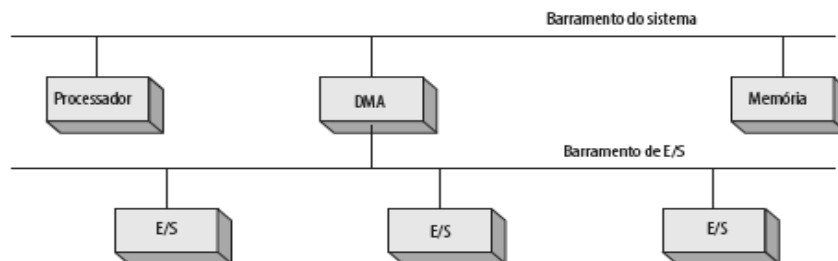


- Único barramento, controle de DMA separado.
- Cada transferência usa barramento duas vezes.  
—E/S para DMA, depois DMA para memória.
- CPU é suspensa duas vezes.



(b) Único barramento, DMA-E/S integrados

- Único barramento, controlador de DMA integrado.
- Controlador pode aceitar mais de um dispositivo.
- Cada transferência usa barramento uma vez.
  - DMA para memória.
- CPU é suspensa uma vez.



(c) Barramento de E/S

- **Barramento de E/S separado.**
- Barramento aceita todos dispositivos habilitados para DMA.
- Cada transferência usa barramento uma vez.
  - DMA para memória.
- CPU é suspensa uma vez.

## Flutuando

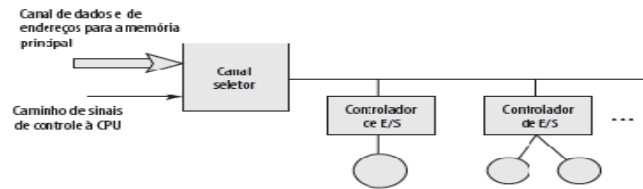
- Enquanto DMA usa barramentos, processador fica ocioso.
- Processador usando barramento, DMA ocioso:
  - Conhecido como controlador de DMA flutuante.
- Dados não passam e são armazenados no chip de DMA.
  - DMA apenas entre porta de E/S e memória.
  - Não entre duas portas de E/S ou dois locais de memória.
- Pode transferir de memória para memória via registrador.
- 8237 contém quatro canais de DMA.
  - Programado independentemente.
  - Qualquer um ativo.
  - Canais numerados com 0, 1, 2 e 3.

## Canais e Processadores de E/S

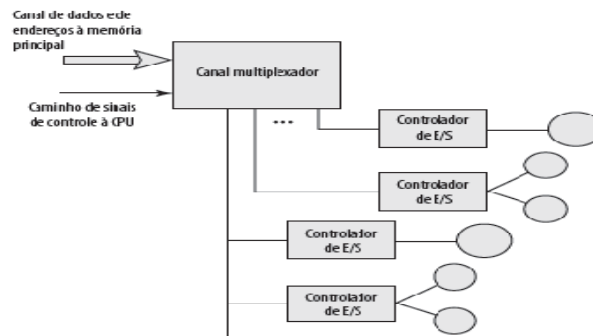
- Extensões do conceito de DMA.
- Dispositivos de E/S se tornam mais sofisticados.
- P.e., placas gráficas 3D.
- CPU instrui controlador de E/S a realizar transferência.
- **Canal de E/S** - Controlador de E/S com conjunto especializado de instruções, realiza transferência completa.
- Melhora velocidade.
  - Retira carga da CPU.
  - Processador dedicado é mais rápido.
- **Processador de E/S** possui memória local própria e de fato é um computador separado



## Arquitetura do canal de E/S



(a) Seleção



(b) Multiplexador

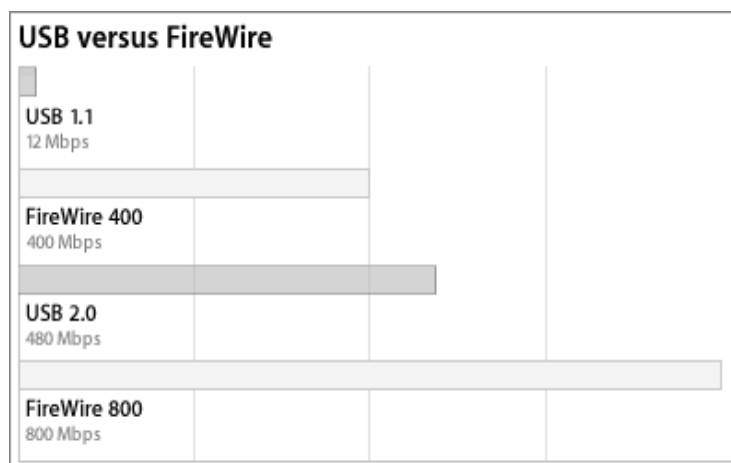
## Interface Externa: Firewire e InfiniBand

- Interface multiponto (e não ponto a ponto)
- Conexão entre módulos de E/S e dispositivos externos.
- São barramentos externos
- P.exemplo, FireWire e InfiniBand.

## IEEE 1394 FireWire

Barramento	eSATA	PATA	Fire Wire 1394b	USB 2.0
<b>Velocidade atual</b>	2.4 >Gb/s	1064 Mb/s	786 Mb/s	~375 Mb/s
<b>Máx comp. do cabo</b>	2 metros	46 cm	4.5 metros 16 cabos podem ser encadeados até 72 metros	
<b>Cabo de Alimentação?</b>	Sim	Sim	Não	Não
<b>Dispositivos por Canal</b>	1 (5 com multiplicador)	3 (3º disp. read -only)	63	127

## IEEE 1394 FireWire



Devido à velocidade e eficiência do FireWire 800, em muitos casos, a largura de banda efetiva supera o USB 2.0 em duas vezes.

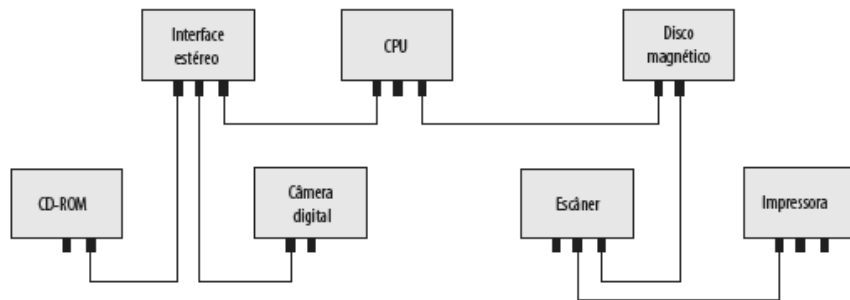
## IEEE 1394 FireWire

- Barramento serial de alto desempenho.
- Rápido.
- Baixo custo.
- Fácil de implementar.
- Também sendo usado em câmeras digitais, VCRs e TV.

## Configuração de FireWire

- **Daisy chain.**
- Até 63 dispositivos em única porta.
  - Na realidade, 64, dos quais um é a própria interface.
- Até 1022 barramentos podem ser conectados com pontes.
- Configuração automática.
- **Sem terminações de barramento.**
- Pode ser estruturada em forma de árvore.
- Hot Plugging (como USB)

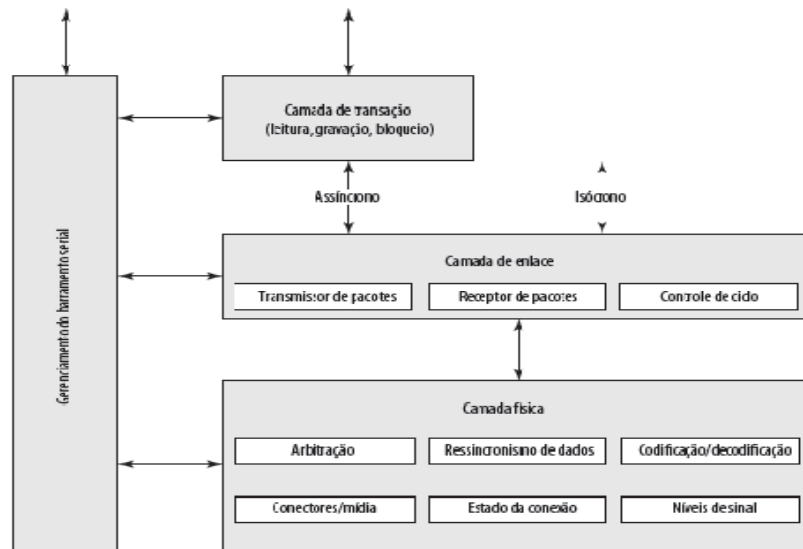
### Configuração FireWire simples



### Pilha de 3 camadas do FireWire

- Física:
  - Meio de transmissão, características elétricas e de sinalização.
- Enlace:
  - Transmissão de dados em pacotes.
- Transação:
  - Protocolo requisição-resposta.

## Pilha de protocolos FireWire



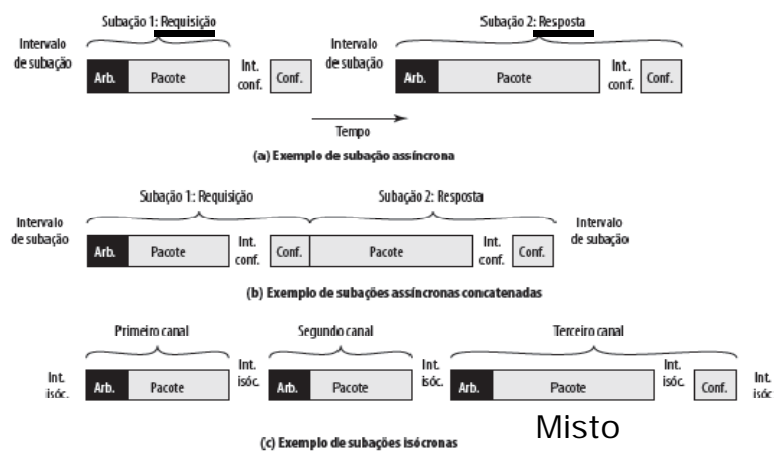
## Fire Wire – Camada física

- **Taxas de dados de 25 a 3200 Mbps.**
- Duas formas de arbitração:
  - Baseado na estrutura em forma de árvore.
  - Raiz atua como árbitro.
  - Primeiro a chegar, primeiro a ser atendido.
  - Prioridade natural controla requisições simultâneas, ou seja, quem está mais próximo da raiz.
- Duas funções adicionais:
  - Arbitração imparcial (mais justa).
  - Arbitração urgente (com prioridade).

## Fire Wire – Camada de enlace

- Dois tipos de transmissão:
  - Assíncrono:
    - **Quantidade variável de dados** e vários bytes de dados de transação transferidos como um pacote.
    - Para endereço explícito.
    - Confirmação retornada.
  - Isócrono:
    - Quantidade variável de dados em **sequência de pacotes de tamanho fixo** em intervalos regulares.
    - Endereçamento simplificado.
    - Sem confirmação.

## Subações do Fire Wire



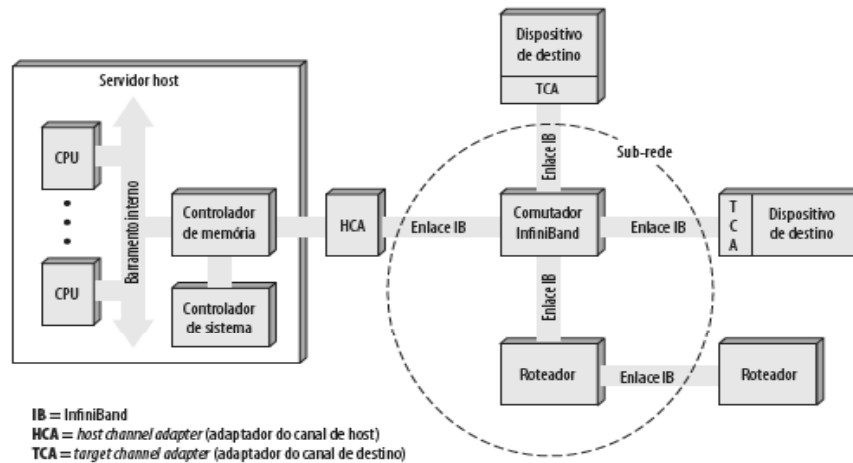
## InfiniBand

- Especificação de E/S **voltada para servidores de ponta**.
  - União da Future I/O (Cisco, HP, Compaq, IBM) e Next Generation I/O (Intel).
- Versão 1 lançada no início de 2001.
- Arquitetura e especificação para fluxo de dados entre processador e dispositivos de E/S inteligentes.
- **Pretende substituir PCI nos servidores.**
- Maior capacidade, facilidade de expansão, flexibilidade.

## Arquitetura InfiniBand

- Armazenamento remoto, rede e conexão entre servidores.
- Conecta servidores, armazenamento remoto, dispositivos de rede à estrutura central de comutadores e conexões.
- Maior densidade de servidores.
- Central de dados expansível.
- Nós independentes acrescentados conforme a necessidade.
- Distribuição de E/S do servidor até:
  - 17m usando cobre.
  - **300m usando fibra óptica multimodo.**
  - 10km usando fibra óptica de monomodo.
- **Até 30Gbps.**

## Fábrica de comutadores InfiniBand

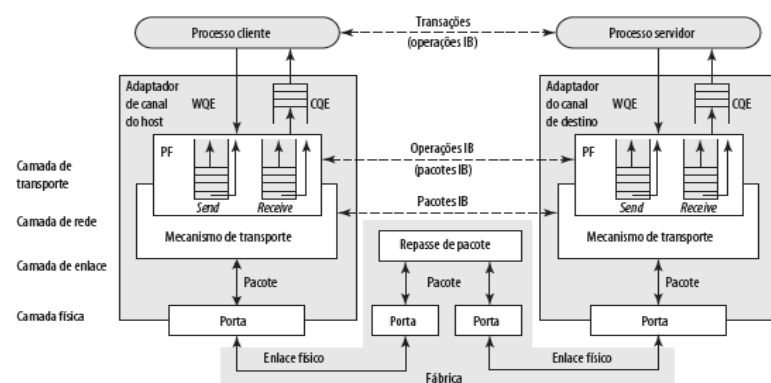


## Operação do InfiniBand

- 16 canais lógicos (pistas virtuais) por enlace físico.
- Uma pista para gerenciamento, restante para dados.
- Dados no fluxo de pacotes.
- Pista virtual dedicada temporariamente à transferência fim a fim.
- Comutador mapeia tráfego da pista de entrada para pista de saída.



## Pilha de protocolos InfiniBand



IB = InfiniBand  
WQE = work queue element (elemento de fila de trabalho)  
CQE = completion queue entry (entrada de fila de término)  
PF = par de filas