# Arquitetura de Computadores II
# 1COP012

## Entrada/Saída

# Entrada/Saída

## Objetivos

- Explicar o uso dos módulos de E/S
- Entender a diferença entre E/S programada e E/S controlada por interrupção e discutir suas vantagens relativas.
- Apresentar visão geral da operação do acesso direto à memória.
- Apresentar uma visão geral do acesso direto à cache.

# Entrada/Saída

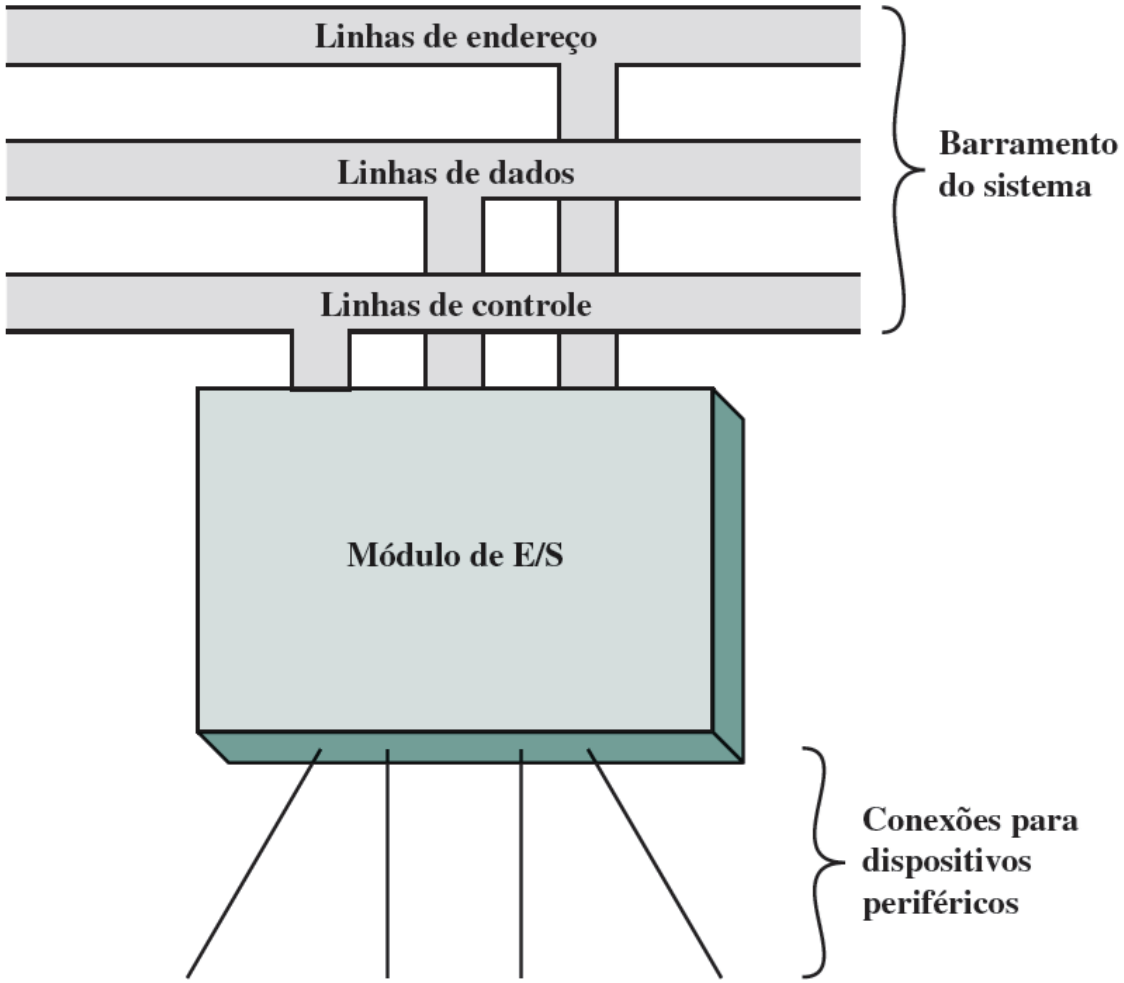
## Dispositivos externos

- As operações de E/S são realizadas por meio de uma grande variedade de dispositivos externos.
- Um dispositivo externo conecta-se ao computador por uma conexão com um módulo de E/S.
- Podemos classificar os dispositivos externos em geral em três categorias:
  1. Inteligíveis ao ser humano
  2. Inteligíveis à maquina
  3. Comunicação

# Entrada/Saída

## Dispositivos externos

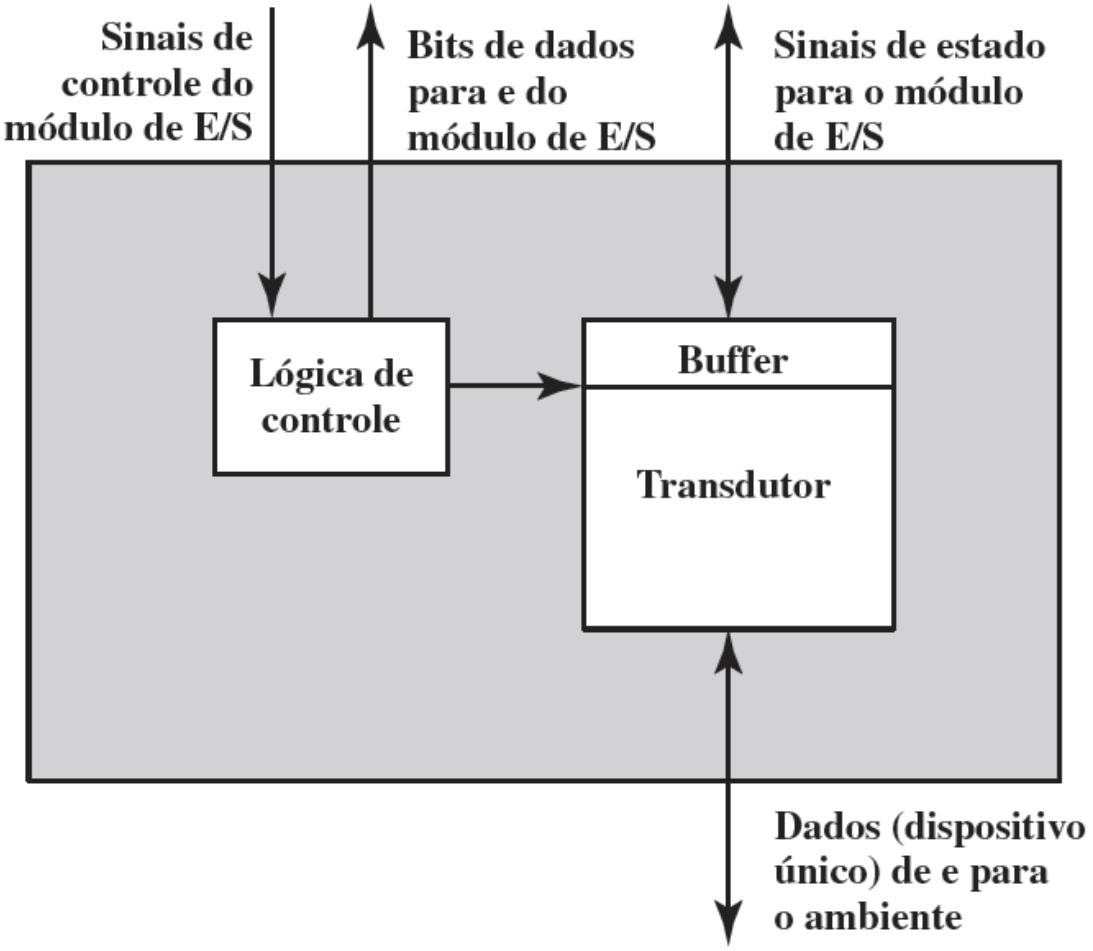
- Modelo genérico de um módulo de E/S:



# Entrada/Saída

## Dispositivos externos

- Diagrama em blocos de um dispositivo externo:



# Entrada/Saída

## Teclado/monitor

- O meio mais comum de interação entre computador/usuário é o conjunto teclado/monitor.
- O usuário fornece entrada pelo teclado.
- A unidade de troca básica é o caractere.
- Associado a cada caractere existe um código, em geral com tamanho de 7 a 8 bits.
- Quando o usuário pressiona uma tecla, isso gera um sinal eletrônico que é interpretado pelo transdutor no teclado e traduzido para o padrão de bits do código IRA correspondente.

# Entrada/Saída

## Teclado/monitor

- Esse padrão de bits é, então, transmitido ao módulo de E/S no computador, onde o texto pode ser armazenado no mesmo código IRA.
- Na saída, os caracteres do código IRA são transmitidos para um dispositivo externo do módulo de E/S.
- O transdutor no dispositivo interpreta esse código e envia os sinais eletrônicos exigidos ao dispositivo de saída, ou para exibir o caractere indicado, ou para realizar a função de controle solicitada.

# Entrada/Saída

## Drive de disco

- Uma unidade de disco contém a eletrônica para trocar sinais de dados, controle e estado com um módulo de E/S mais a eletrônica para controlar os mecanismos de leitura/gravação de disco.
- Em um disco de cabeça fixa, o transdutor é capaz de converter os padrões magnéticos na superfície do disco móvel em bits no buffer do dispositivo.
- Um disco com cabeça móvel também deve ser capaz de fazer o braço do disco se mover radialmente para dentro e fora pela superfície do disco.

# Entrada/Saída

## Módulo de E/S

- As principais funções ou requisitos para um módulo de E/S encontram-se nas seguintes categorias:
  - Controle e temporização
  - Comunicação com o processador
  - Comunicação com o dispositivo
  - Buffering de dados
  - Detecção de erro

# Entrada/Saída

## Módulo de E/S

- A função de E/S inclui um requisito de **controle e temporização** para coordenar o **fluxo de tráfego** entre os recursos internos e dispositivos externos.

- A **comunicação do processador** envolve o seguinte:
  - Decodificação de comando
  - Dados
  - Informação de estado

# Entrada/Saída

## Módulo de E/S

- A função de E/S inclui um requisito de **controle e temporização** para coordenar o **fluxo de tráfego** entre os recursos internos e dispositivos externos.

- A **comunicação do processador** envolve o seguinte:
  - **Decodificação de comando**: o módulo de E/S aceita comandos do processador, normalmente enviado como sinais no barramento de controle.
  - **Dados**: os dados são trocados entre o processador e o módulo de E/S pelo barramento de dados.
  - **Informação de estado**: como os periféricos são muito lentos, é importante conhecer o estado do módulo de E/S. Por exemplo para uma unidade de disco (Read Sector, Write Sector, Seek e Scan Id).
  - **Reconhecimento de endereço**: o módulo de E/S precisa reconhecer um endereço exclusivo para cada periférico que controla.

# Entrada/Saída

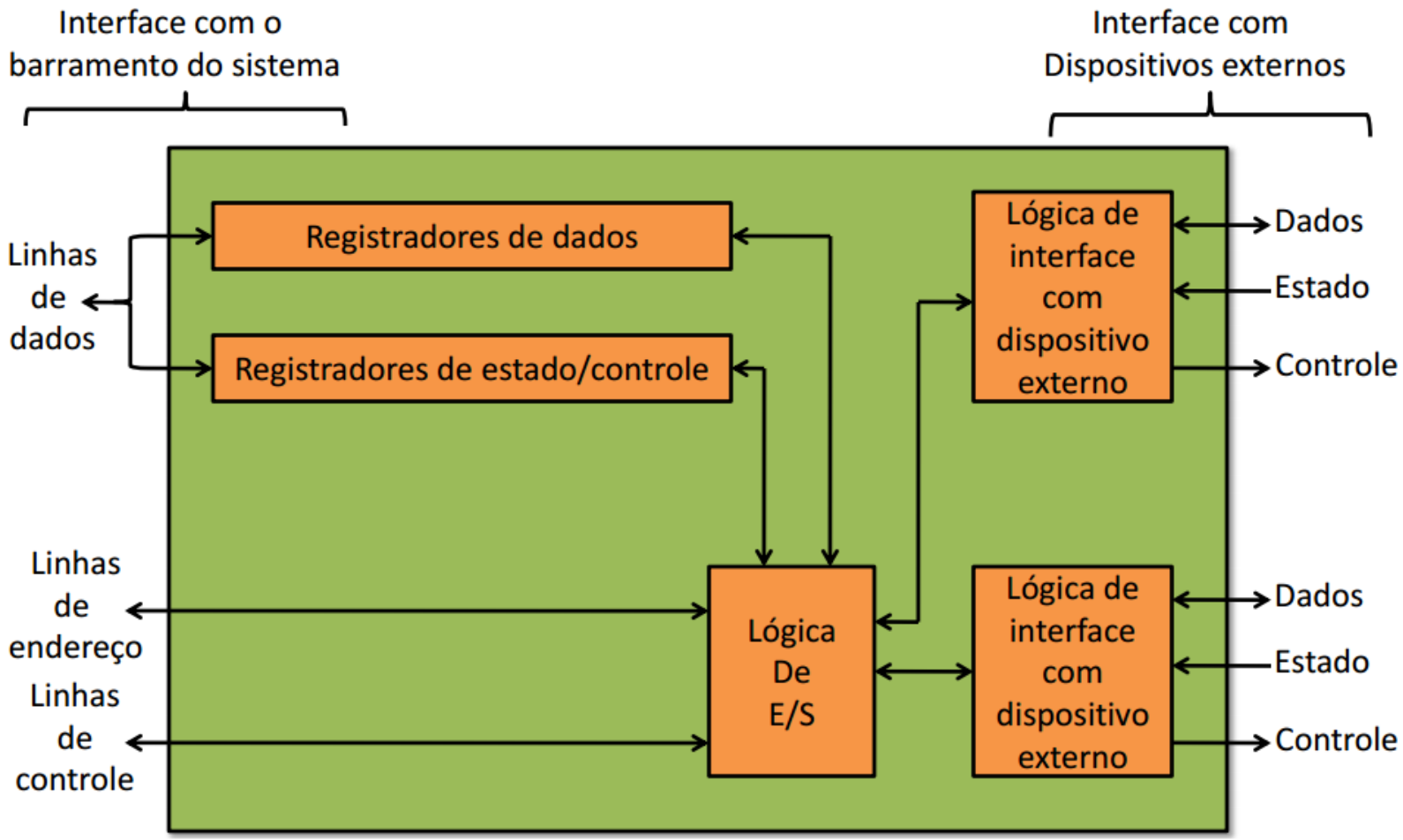
## Módulo de E/S

- O módulo de E/S também deve ser capaz de realizar comunicação com o dispositivo.
- Ela envolve comandos, informação de estado e dados.
- Uma tarefa essencial de um módulo de E/S é o *buffering* de dados.
- Os dados são mantidos em um *buffer* no módulo de E/S e depois enviados ao dispositivo periférico em sua taxa de dados.
- Por fim, um módulo de E/S é responsável pela detecção de erro e, subsequentemente, por relatar erros ao processador.

# Entrada/Saída

## Estrutura do módulo de E/S

- Diagrama do bloco de um módulo de E/S:



# Entrada/Saída

## E/S programada

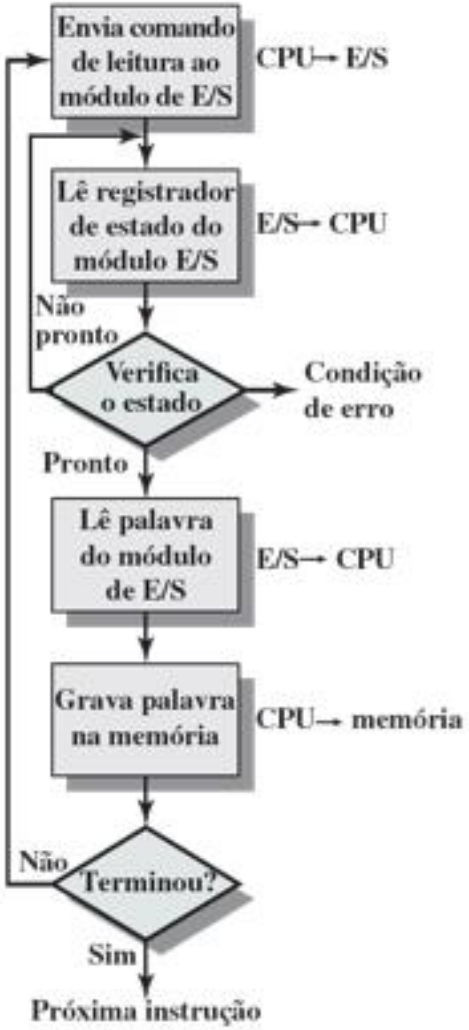
- Três técnicas são possíveis para operações de E/S:

| | Sem interrupções | Uso de interrupções |
|---|---|---|
| Transferência de E/S para memória via processador | E/S programada | E/S controlada por interrupção |
| Transferência direta de E/S para memória | | Acesso direto à memória (DMA) |

- A alternativa a estes modos é conhecida como acesso direto à memória (DMA).
- Nesse modo, o módulo de E/S e a memória principal trocam dados diretamente, sem envolvimento do processador.

# Entrada/Saída

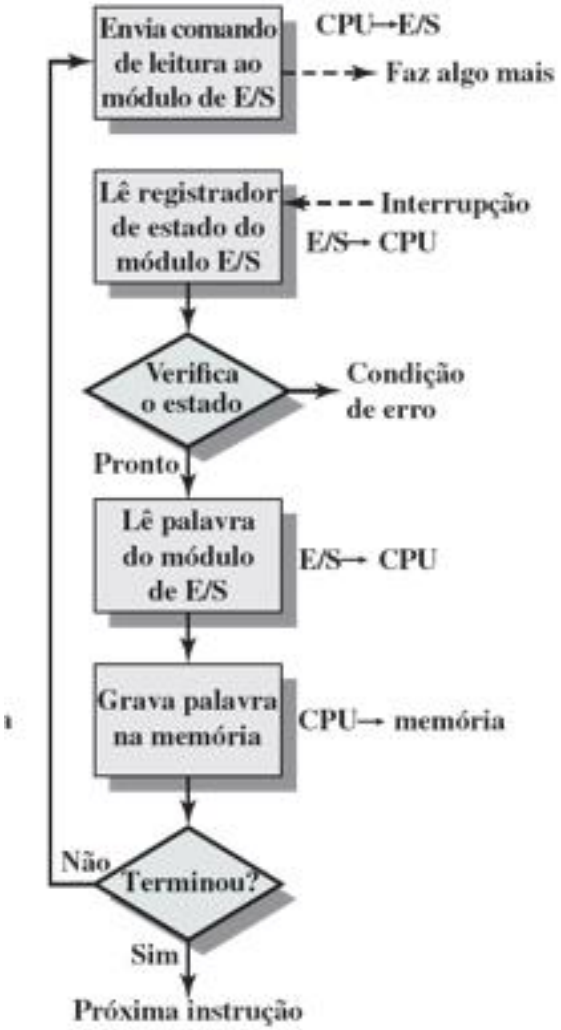
## Visão geral da E/S programada




E/S programada

# Entrada/Saída

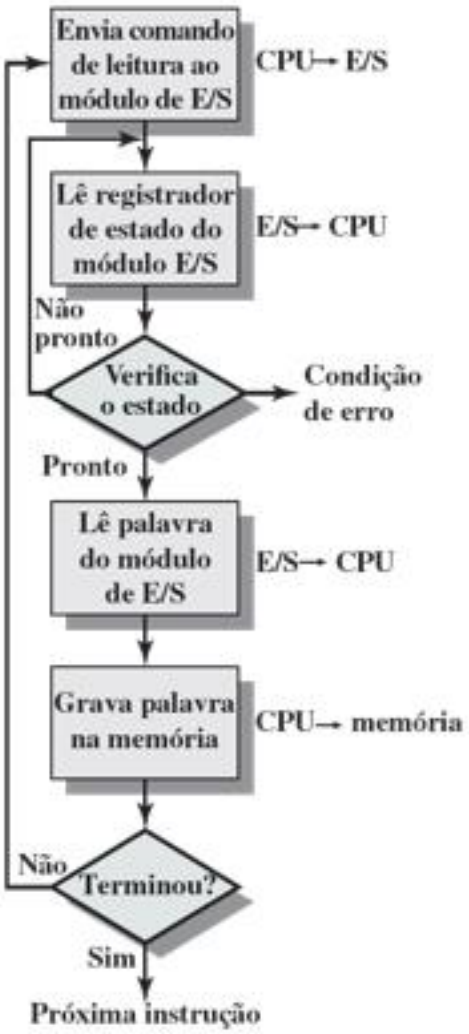
## Visão geral da E/S programada por interrupção



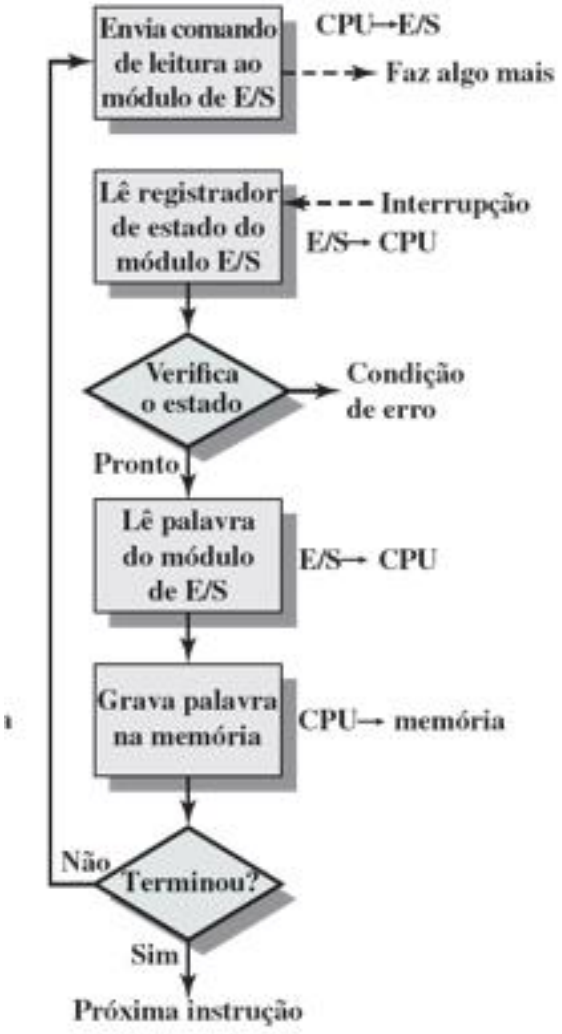

E/S dirigida por interrupção

# Entrada/Saída

## Comparação E/S programada e por interrupção



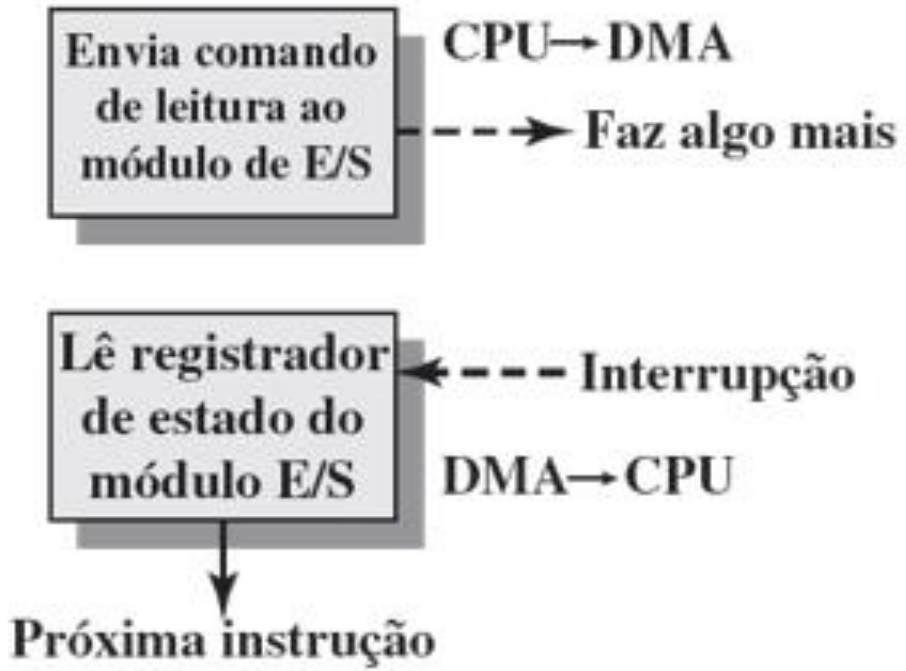

E/S programada




E/S dirigida por interrupção

# Entrada/Saída

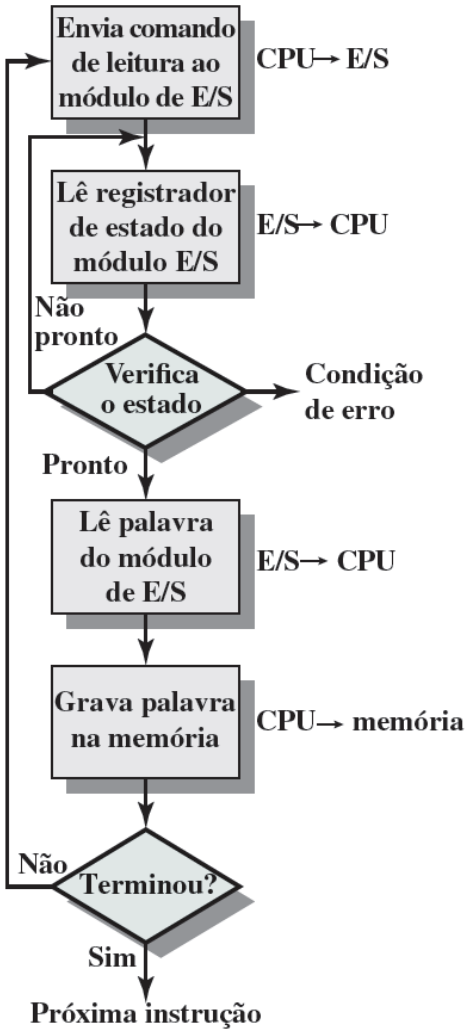
## Acesso Direto à Memória (DMA)



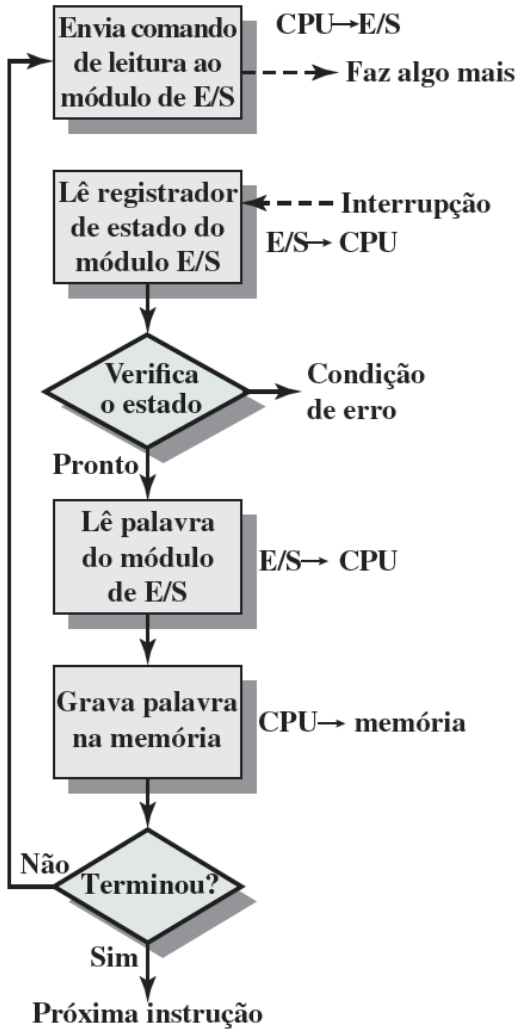

Acesso direto à memória

# Entrada/Saída

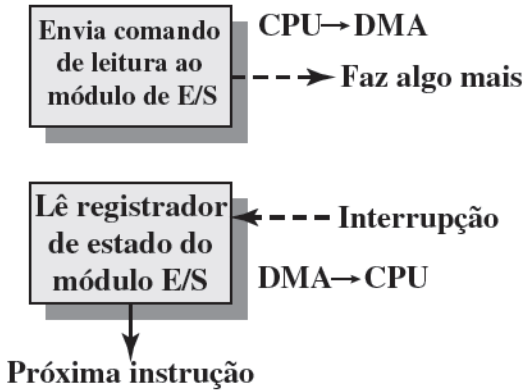
## Visão geral da E/S programada




E/S programada




E/S dirigida por interrupção

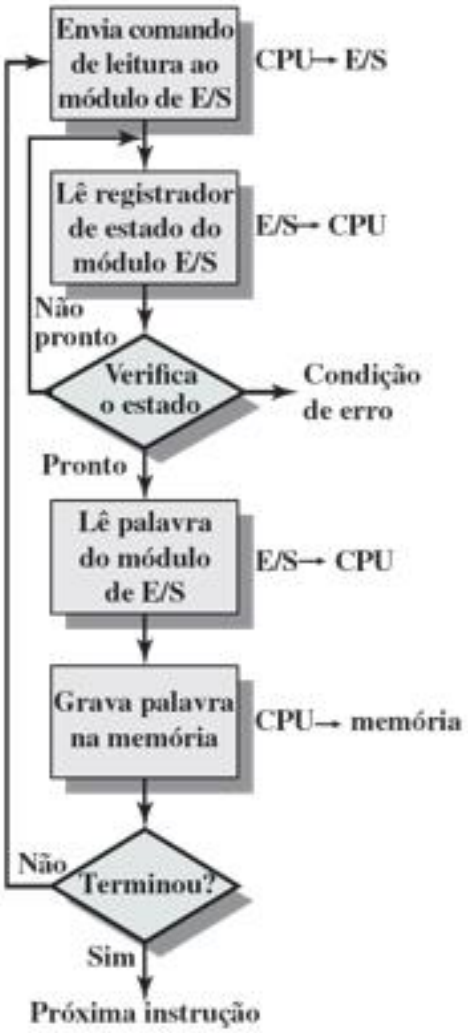



Acesso direto à memória

# Entrada/Saída

## 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**

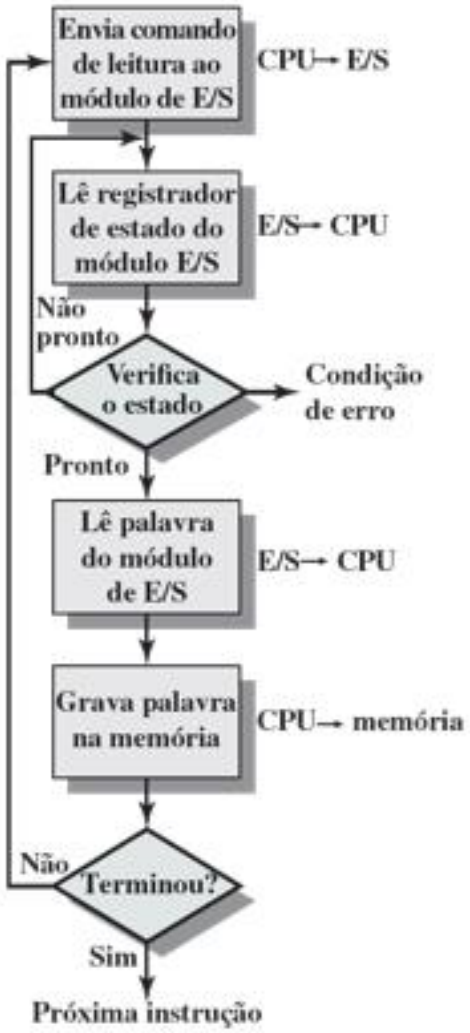



E/S programada

# Entrada/Saída

## E/S programada - detalhe

- CPU solicita operação de E/S
- Módulo de E/S realiza operação
- Módulo de E/S define 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




E/S programada

# Entrada/Saída

## Comando de E/S

- Existem quatro tipos de comandos de E/S:

1. **Controle:** usado para ativar um periférico e dizer-lhe o que fazer.
2. **Teste:** usado para testar diversas condições de estado associadas a um módulo de E/S e seus periféricos.
3. **Leitura:** faz com que o módulo de E/S obtenha um item de dados do periférico e o coloque em um buffer interno.
4. **Escrita:** faz com que o módulo de E/S apanhe um item de dado do barramento de dados e depois transmita-o ao periférico.

# Entrada/Saída

## Endereçamento de dispositivos de E/S

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

# Entrada/Saída

## Mapeamento de E/S

Quando o processador, a memória principal e a E/S compartilham um barramento comum, dois modos de endereçamento são possíveis:

- **E/S mapeada na memória:**
  - 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

# Entrada/Saída

## Mapeamento de E/S

Quando o processador, a memória principal e a E/S compartilham um barramento comum, dois modos de endereçamento são possíveis:

- **E/S independente:**
  - Espaços de endereços separados
  - Precisa de linhas de seleção de E/S ou memória
  - Comandos especiais para E/S
    - Conjunto limitado

# Entrada/Saída

## E/S controlada por interrupção

- O problema com a E/S programada é que o processador tem de esperar muito tempo para que o módulo de E/S de interesse esteja pronto para recepção ou transmissão de dados.
- Uma alternativa é que o processador envie um comando de E/S para um módulo e depois continue realizando algum outro trabalho útil.
- O módulo interromperá o processador para solicitar atendimento quando estiver pronto para trocar dados com o processador.
- O processador, então, executará a transferência de dados, como antes, e depois retomará seu processamento anterior.

# Entrada/Saída

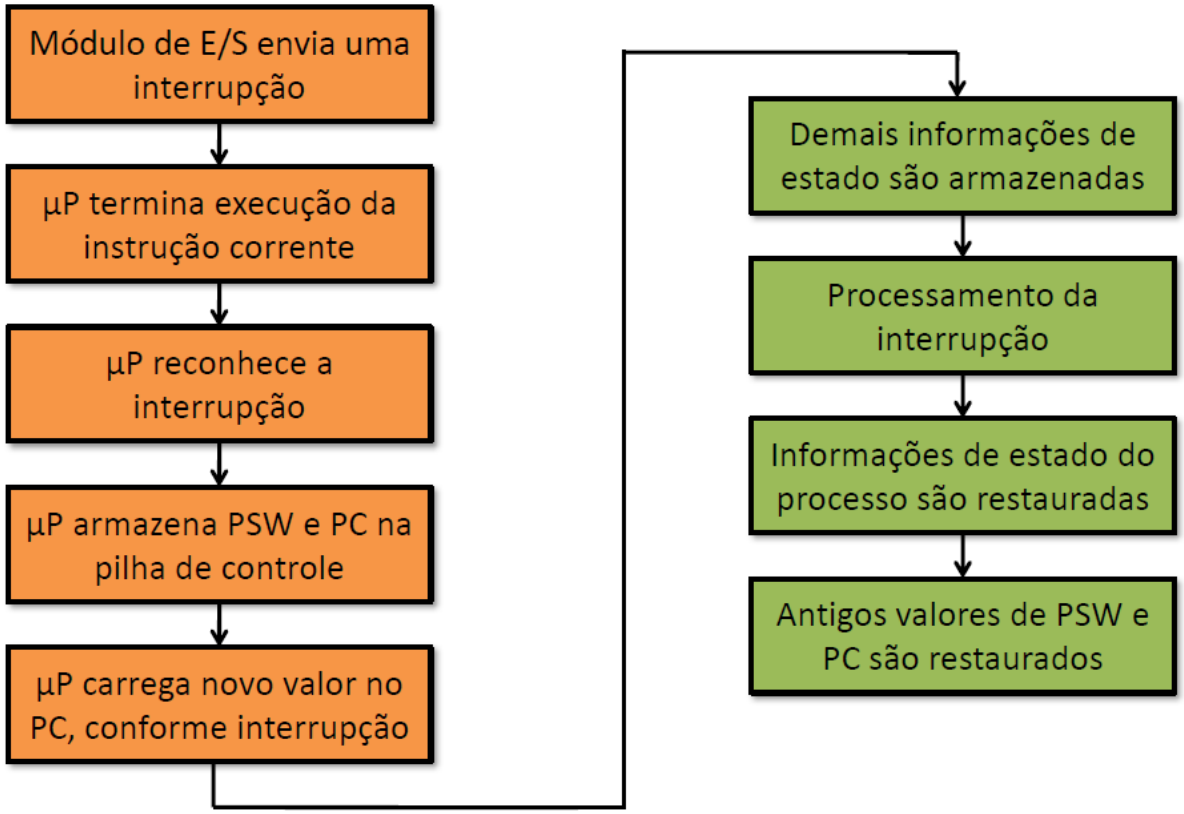
## E/S controlada por interrupção – Operação básica

- 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

# Entrada/Saída

## E/S controlada por interrupção

- Processamento de interrupção simples



# Entrada/Saída

## 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 e armazena

# Entrada/Saída

## E/S controlada por interrupção

- Dois aspectos de projeto surgem na implementação da E/S por interrupção:
  1. Considerando que quase sempre haverá vários módulos de E/S, como o processador determina qual dispositivo emitiu a interrupção?
  2. Se houver várias interrupções, como o processador decide qual deverá processar?

- A técnica mais simples para o problema é oferecer múltiplas linhas de interrupção entre o processador e os módulos de E/S.

# Entrada/Saída

## E/S controlada por interrupção

- Todavia, é impraticável dedicar mais do que algumas poucas linhas de barramento ou pinos de processador às linhas de interrupção.
- Uma alternativa é a verificação por software (software polling).
- A CPU consulta cada módulo em uma sequência circular.
- A desvantagem da verificação por software é que ele é demorado.
- Uma técnica mais eficiente é usar uma configuração *daisy chain*, que oferece uma verificação por hardware.

# Entrada/Saída

## Identificando módulo que interrompe

- Daisy chain (hardware poll) 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 (bus mastering):
  - Módulo deve reivindicar o barramento antes que possa causar uma interrupção
  - Ex. PCI e SCSI

# Entrada/Saída

## Múltiplas interrupções

- Cada linha de interrupção tem uma prioridade.
- Linhas com prioridade mais alta podem interromper linhas com prioridade mais baixa.
- Se o método empregado for *bus mastering* (arbitração de barramento), somente o mestre atual pode interromper.

# Entrada/Saída

## Exemplo – Barramento do PC

- 80x86 tem uma linha de interrupção
- Sistemas baseados no 8086 usam um **controlador de interrupção** 82C59A
- 82C59A tem 8 linhas de interrupção

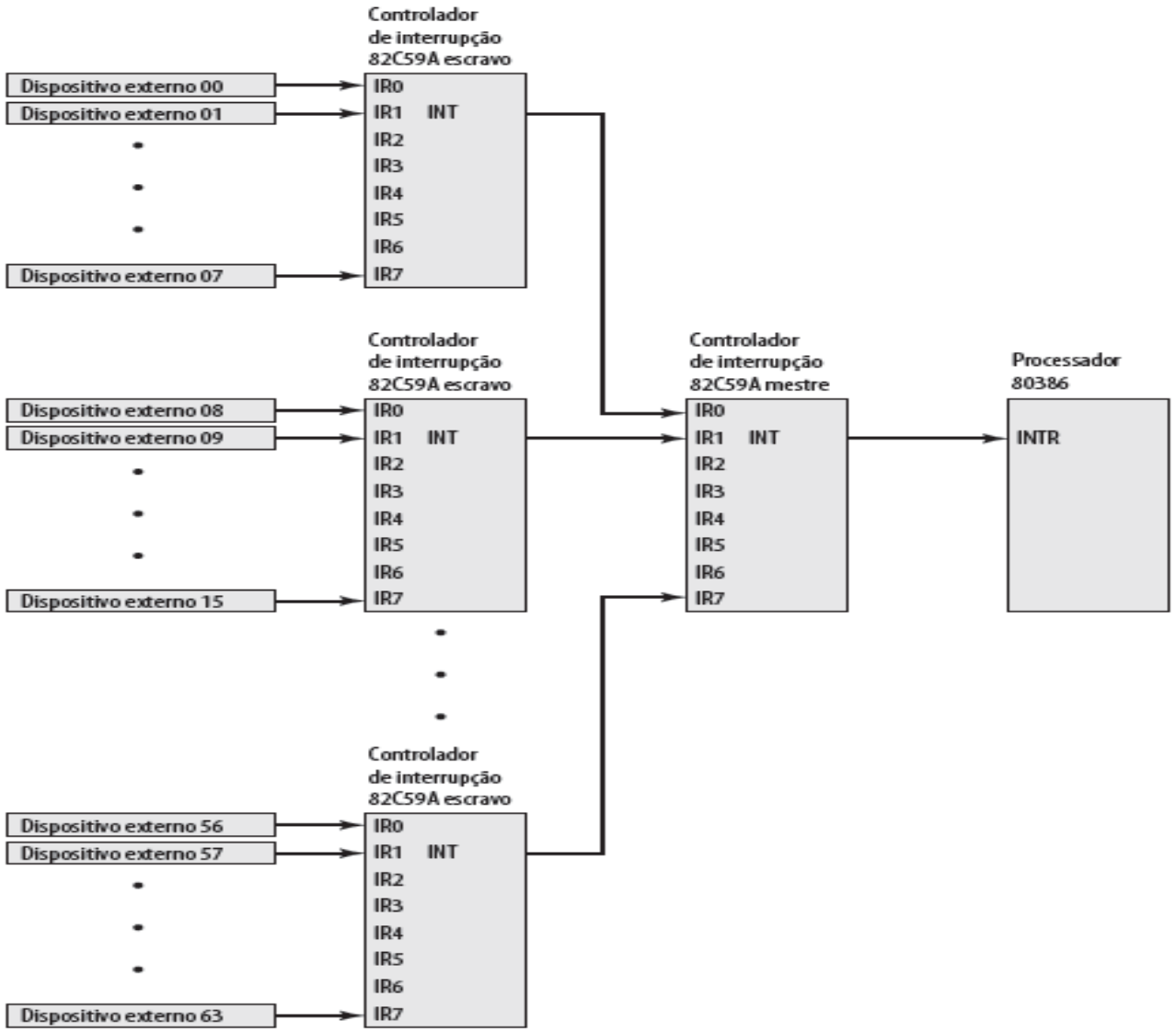
# Entrada/Saída

## Sequência de eventos

- 825C9A aceita interrupções
- 82C59A determina prioridade
- 82C59A signals 8086 (levanta linha INTR)
- CPU confirma (acknowledge)
- 82C59A coloca vetor correto no barramento de dados
- CPU processa interrupção

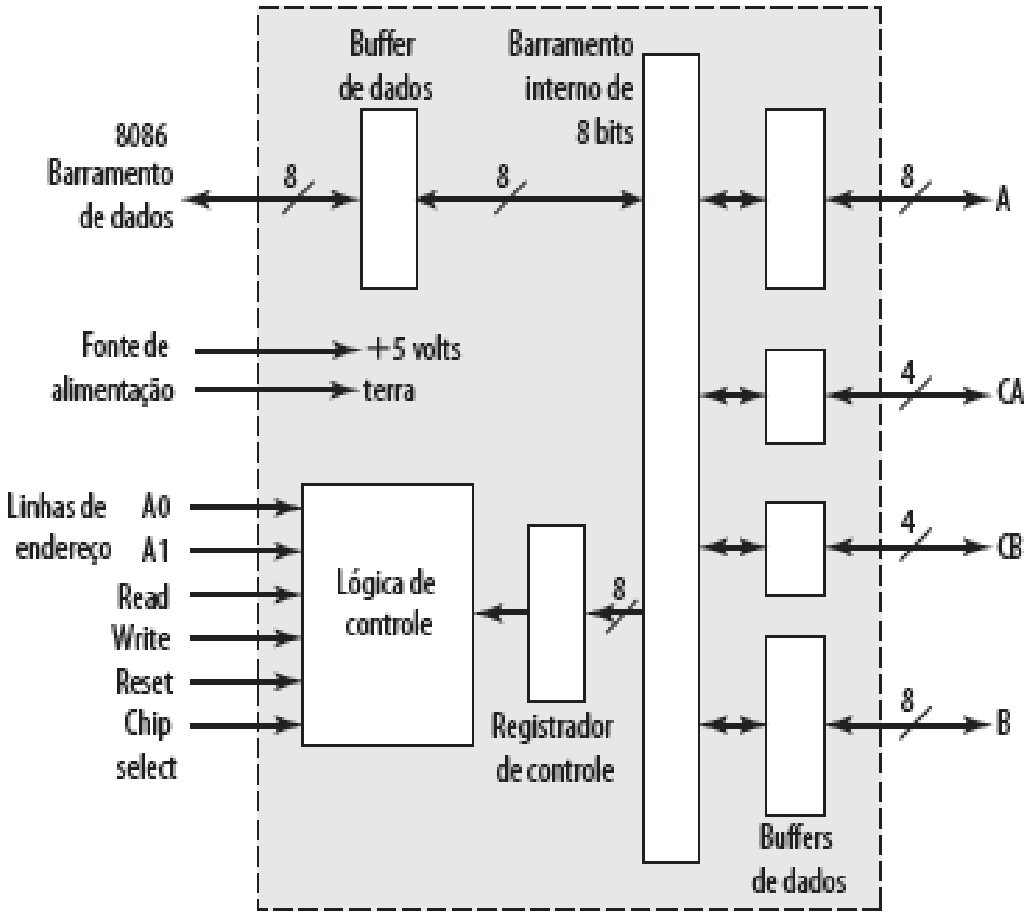
# Entrada/Saída

## Controlador de interrupção do 82C59A

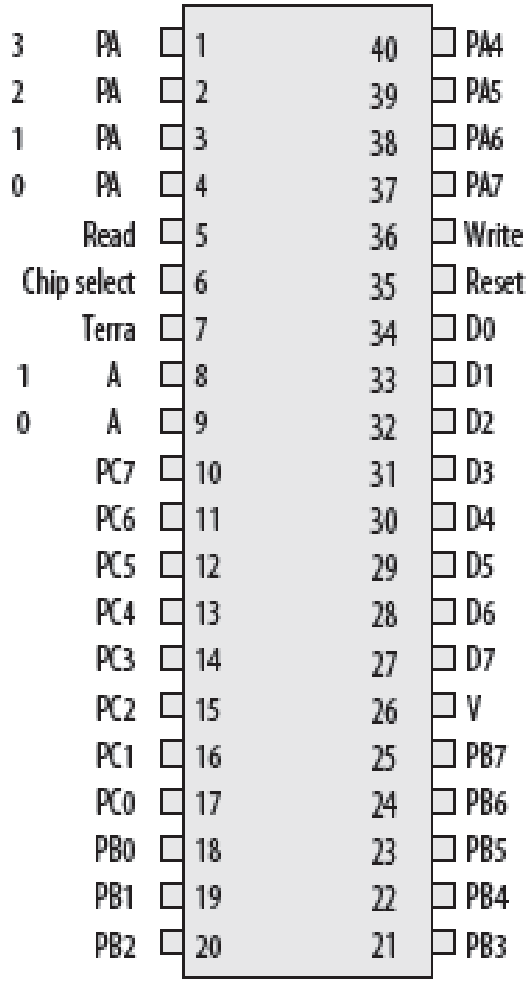


# Entrada/Saída

## Intel 82C55A – Programmable Peripheral Interface



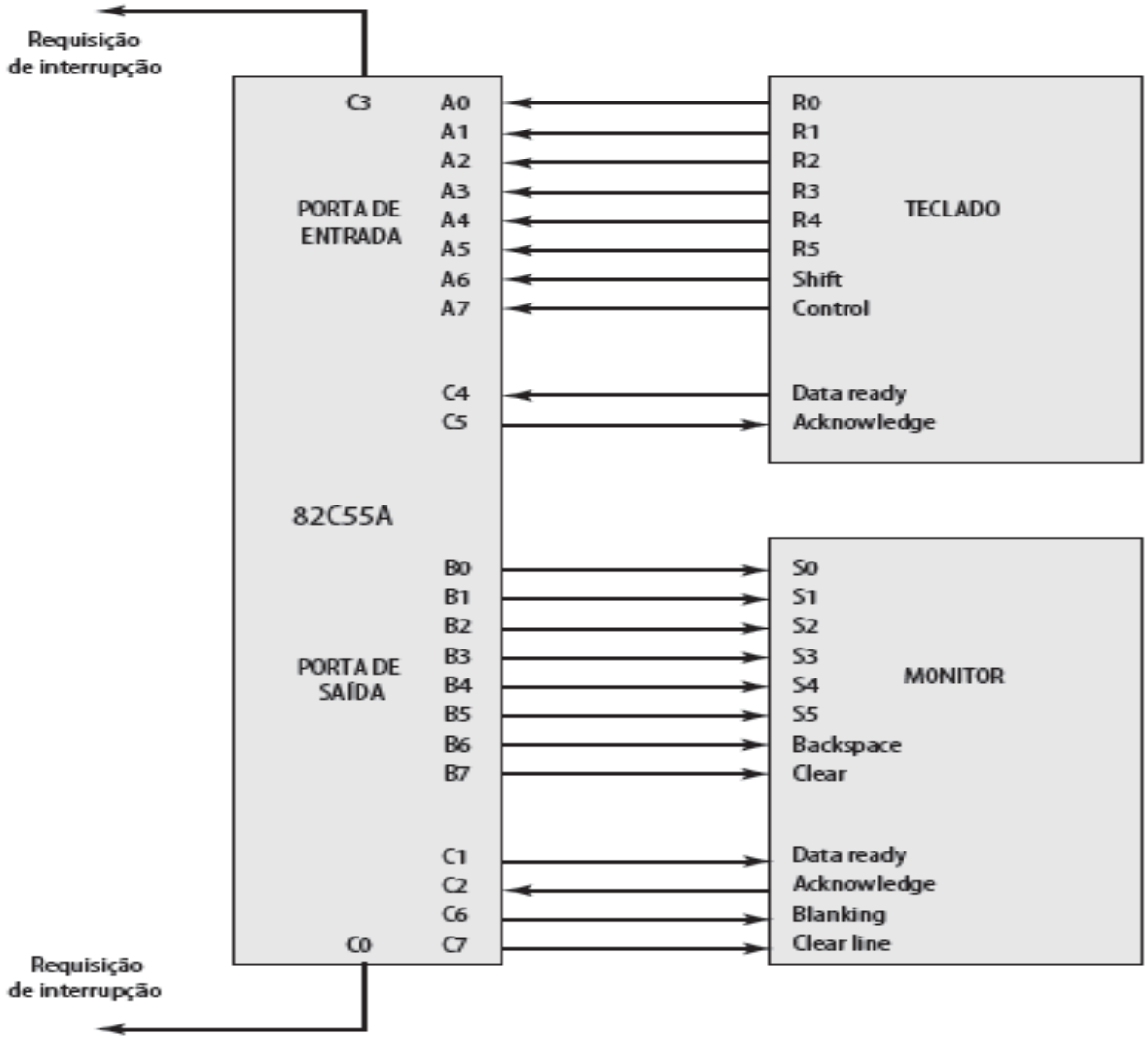

(a) Diagrama em blocos




(b) Layout de pinos

# Entrada/Saída

## Interface de teclado/monitor para o Intel 82C55A



# Entrada/Saída

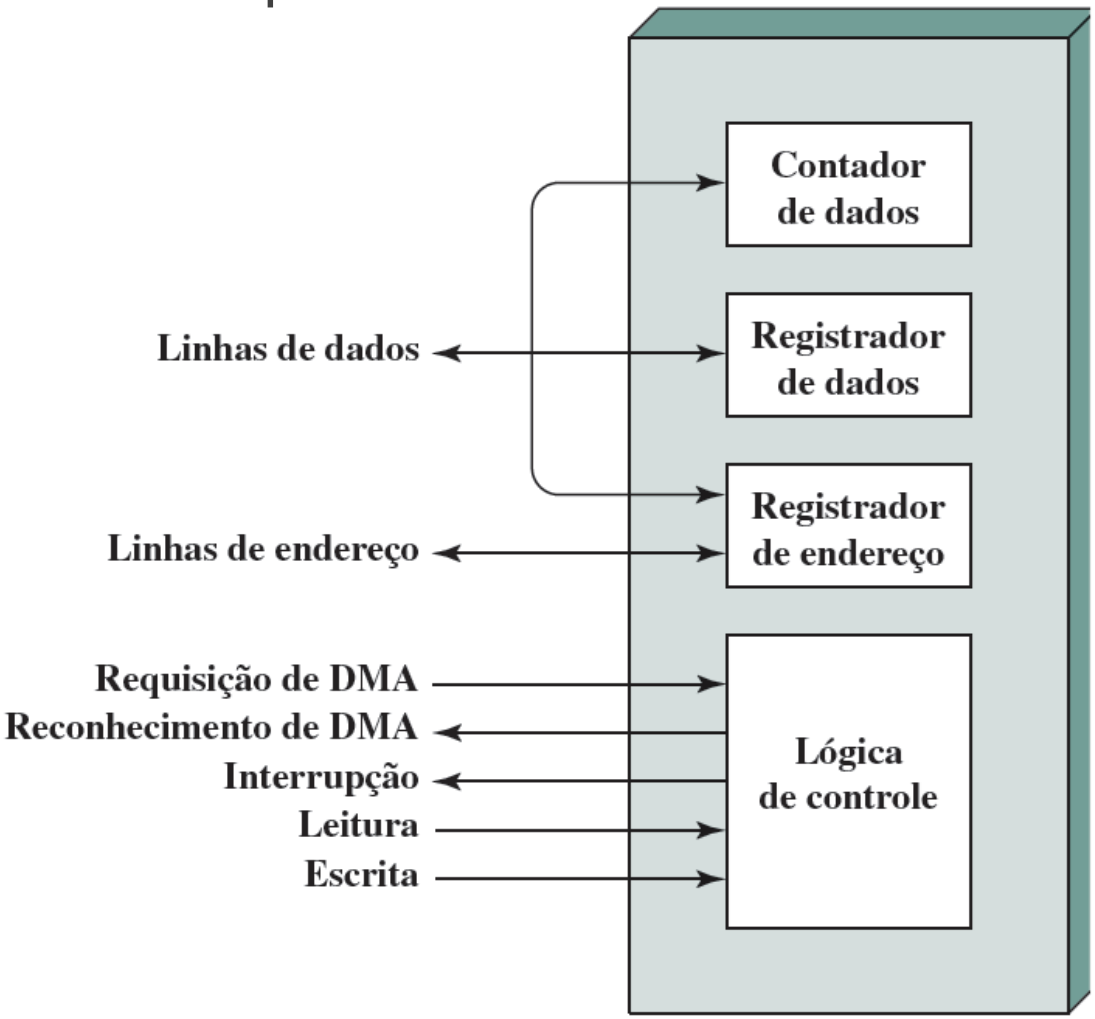
## Acesso direto à memória

- Desvantagens da E/S programada e controlada por interrupção:
  1. A taxa de transferência de E/S é limitada pela velocidade com a qual o processador pode testar e atender a um dispositivo.
  2. O processador fica ocupado no gerenciamento de uma transferência de E/S; diversas instruções precisam ser executadas para cada transferência de E/S.
- O **DMA** envolve um módulo adicional no barramento do sistema.
- O módulo de DMA é capaz de imitar o processador e, na realidade, assumir o controle do sistema do processador.

# Entrada/Saída

## Acesso direto à memória

- Diagrama em blocos típico do DMA:



# Entrada/Saída

## Operação do DMA

- CPU diz (configura) 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/tarefa
- Controlador de DMA lida com as operações transferência
- Controlador de DMA envia interrupção quando terminar

# Entrada/Saída

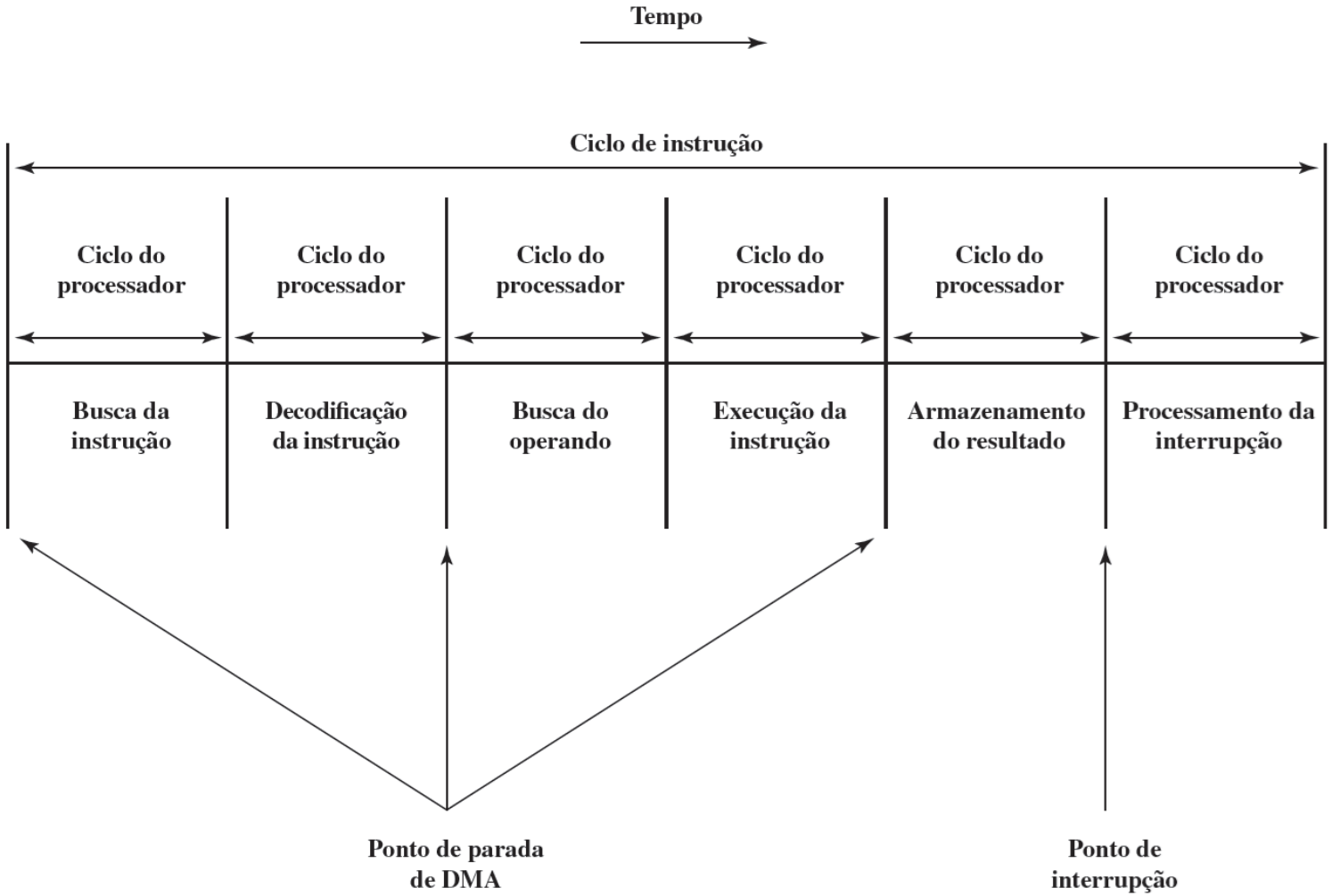
## Transferência de DMA – Roubo de ciclo (*cicle stealing*)

- Controlador de DMA assume o barramento por um ciclo
- Transferência de uma palavra de dados
- Não é 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

# Entrada/Saída

## Acesso direto à memória

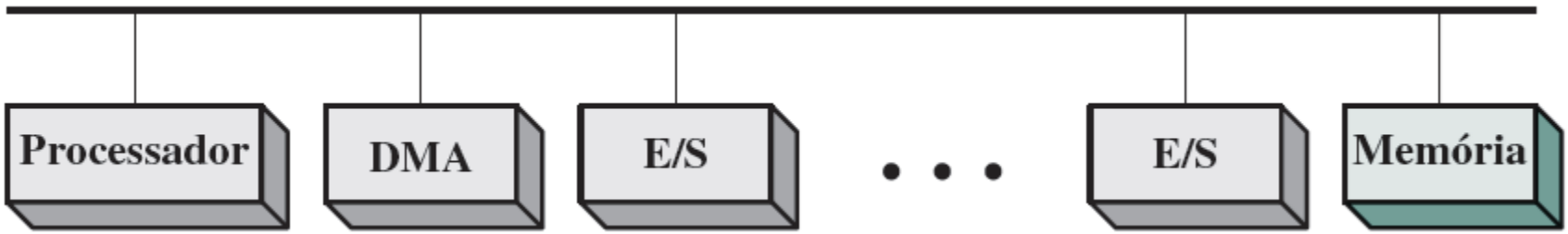
- DMA e pontos de interrupção durante um ciclo de instrução:



# Entrada/Saída

## Acesso direto à memória

- Configurações de DMA alternativas:



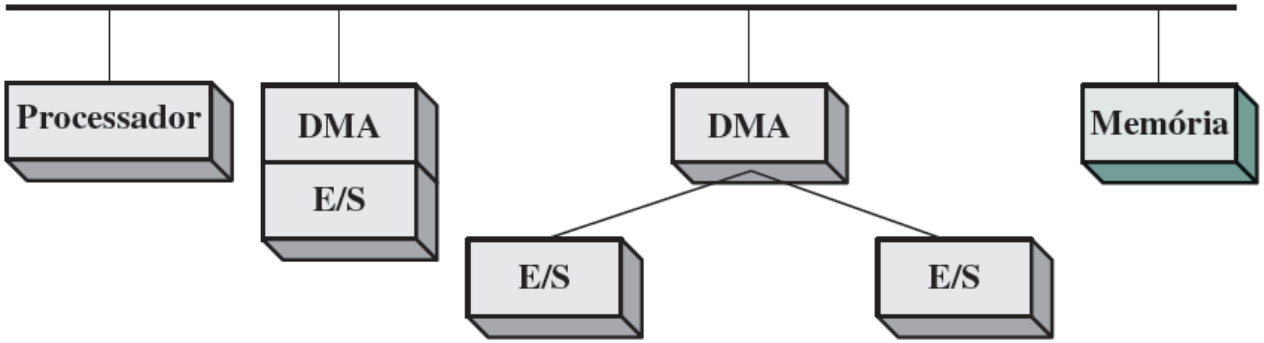

Único barramento, DMA separado

- 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.

# Entrada/Saída

## Acesso direto à memória

- Configurações de DMA alternativas:



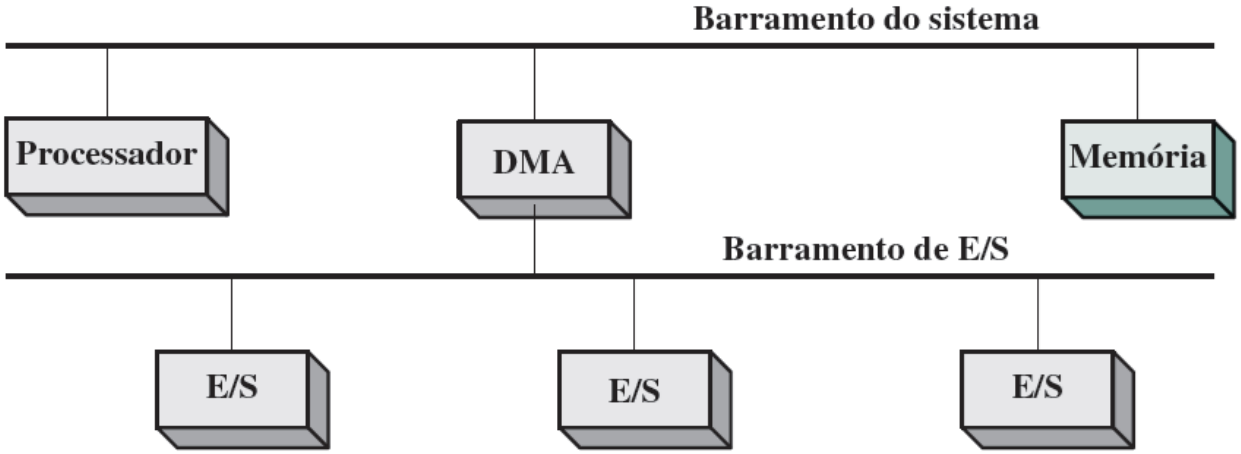

Único barramento, DMA-E/S integrados

- 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

# Entrada/Saída

## Acesso direto à memória

- Configurações de DMA alternativas:




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
- Transferência entre dispositivos não usam o barramento

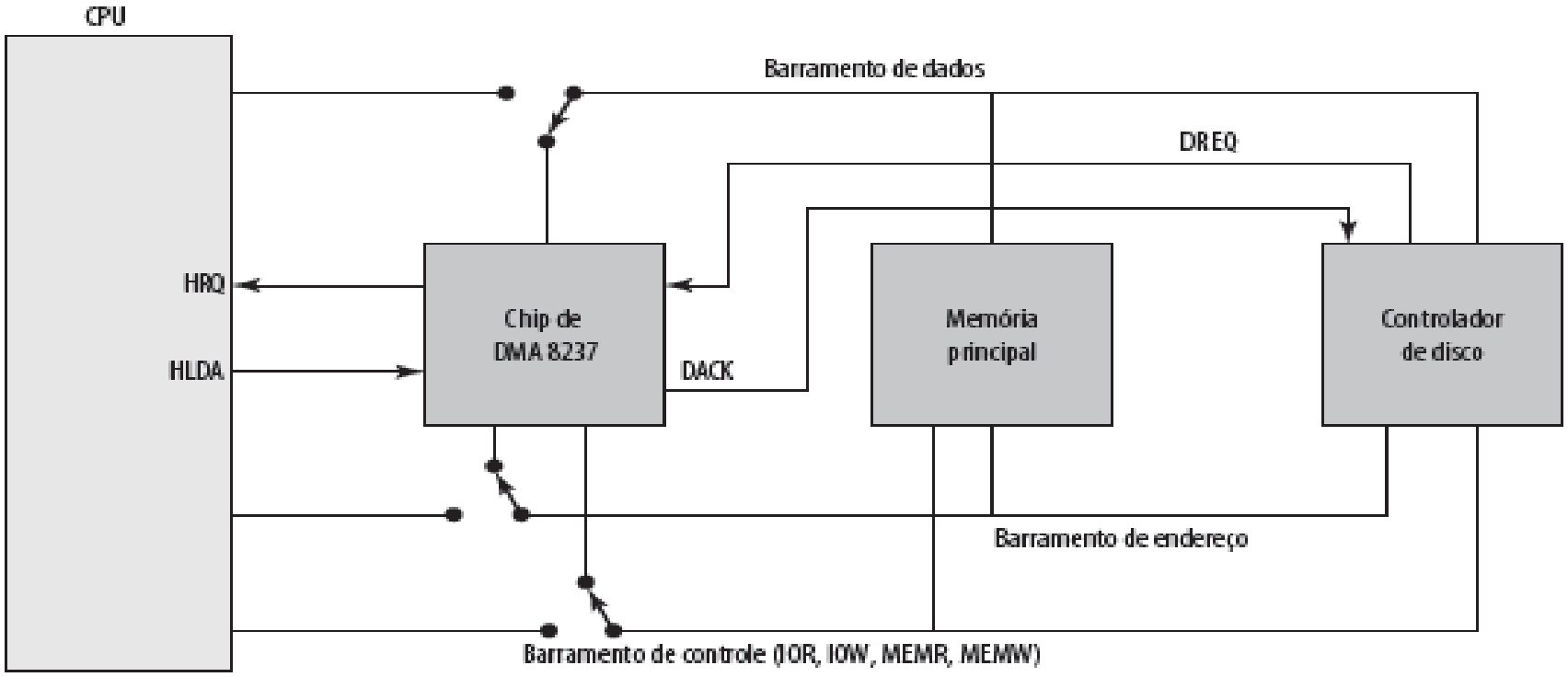
# Entrada/Saída

## Controlador de DMA Intel 8237A

- Interfaces com família 80x86 e DRAM.
- Quando o módulo de DMA precisa de barramentos, ele envia sinal HOLD ao processador.
- CPU responde HLDA (*hold acknowledge*)
  - Módulo de DMA pode usar barramentos
- P.ex., transferir dados da memória para o disco
  1. Dispositivo requisita serviço de DMA levantando DREQ (requisição de DMA)
  2. DMA levanta sua linha HRQ (*hold request*).
  3. CPU termina ciclo de barramento presente (não necessariamente instrução presente) e levanta linha HDLA (*hold acknowledge*). HOLD permanece ativo pela duração do DMA
  4. DMA ativa DACK (DMA *acknowledge*), dizendo ao dispositivo para iniciar a transferência
  5. DMA inicia transferência colocando endereço do primeiro byte no barramento de endereço e ativando MEMR; depois, ativa IOW para escrever no periférico. DMA decrementa contador e incrementa ponteiro de endereço. Repete até contagem chegar a zero
  6. DMA desativa HRQ, retornando o controle do barramento de volta à CPU

# Entrada/Saída

## Uso do barramento do sistema pelo controlador de DMA Intel 8237A




DACK = DMA *acknowledge* (reconhecimento de DMA)
DREQ = *DMA request* (requisição de DMA)
HLDA = HOLD *acknowledge* (reconhecimento de HOLD)
HRQ = HOLD *request* (requisição de HOLD)

# Entrada/Saída

## 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
- 8237A contém quatro canais de DMA
  - Programado independentemente
  - Qualquer um ativo
  - Canais numerados com 0, 1, 2 e 3

# Entrada/Saída

## Acesso direto à cache

- Os sistemas multicore modernos incluem tanto a cache dedicada a cada core como um nível adicional da cache compartilhada, seja L2 ou L3.
- Para esclarecer a interação do DMA e da cache, será útil primeiro descrever uma arquitetura de sistema específica.

**Processador multicore Xeon**

- Trata-se de uma família de processadores de tecnologia de ponta e de alto desempenho usada nos servidores, estações de trabalho de alto desempenho e supercomputadores. Alguns membros da família Xeon usam o sistema de interconexão em anel.
- O E5-2600/4600 pode ser configurado com até oito *cores* em um chip único. Cada core tem caches dedicadas L1 e L2.

# Entrada/Saída

## Acesso direto à cache

- Existe uma cache L3 de até 20 MB.
- A cache L3 é dividida em faixas, cada uma associada com cada core, embora cada um possa se referir a toda a cache.
- Cada faixa tem seu próprio pipeline de cache, de tal maneira que os pedidos podem ser enviados em paralelo a essas faixas.
- A interconexão em anel bidirecional de alta velocidade liga os cores, cache de último nível, PCIe e IMC (controlador integrado de memória).
- Para saída, quando o controlador de E/S emite uma requisição de leitura, a **MCH** (plataforma do controlador de memória) primeiro checa para ver se os dados estão na cache L3.
- A MCH direciona os dados a partir da cache L3 ao controlador de E/S; não são necessários acessos à memória principal.

# Entrada/Saída

## Acesso direto à cache

- Desse modo, a operação de E/S procede de modo eficiente porque ela não requer acesso à memória principal.
- Todavia, se uma aplicação de fato precisar desses dados no futuro, eles devem ser lidos de volta na cache L3 a partir da memória principal.
- Para esclarecer a questão do desempenho e explicar o benefício de DCA como uma maneira de aprimorar o desempenho, vamos analisar o processo de tráfego de protocolo em mais detalhes para o tráfego de entrada. Em termos gerais, ocorrem as seguintes etapas:
  1. Chegada de pacote
  2. DMA
  3. NIC interrompe o host
  4. Cabeçalhos e descritores de recuperação
  5. Ocorrências de falha de cache
  6. O cabeçalho é processado
  7. Transferência do bloco

# Entrada/Saída

## Acesso direto à cache

- Para esclarecer a questão do desempenho e explicar o benefício de DCA como uma maneira de aprimorar o desempenho, vamos analisar o processo de tráfego de protocolo em mais detalhes para o tráfego de entrada. Em termos gerais, ocorrem as seguintes etapas:

**1. Chegada de pacote:** O NIC recebe como entrada um pacote Ethernet. Ele processa e extrai informações de controle da Ethernet (p. ex. cálculo de detecção de erro). O pacote restante de TCP/IP é, então, transferido ao módulo DMA do sistema, que, em geral, é parte do NIC. O NIC também cria um descritor de pacote com informações acerca do pacote, como seu local de buffer na memória.

# Entrada/Saída

## Acesso direto à cache

- Para esclarecer a questão do desempenho e explicar o benefício de DCA como uma maneira de aprimorar o desempenho, vamos analisar o processo de tráfego de protocolo em mais detalhes para o tráfego de entrada. Em termos gerais, ocorrem as seguintes etapas:

**2. DMA: o módulo de DMA transfere dados, inclusive o descritor de pacote, para a memória principal. Ele deve também invalidar as linhas de cache correspondentes, se houve alguma.**

**3. NIC interrompe o host: depois que vários pacotes tiverem sido transferidos, o NIC emite uma interrupção ao processador do host.**

**4. Cabeçalho e descritores de recuperação: o core processa a interrupção, invocando um processo de tratamento de interrupção, o qual lê o descritor e o cabeçalho de pacotes recebidos.**

# Entrada/Saída

## Acesso direto à cache

- Para esclarecer a questão do desempenho e explicar o benefício de DCA como uma maneira de aprimorar o desempenho, vamos analisar o processo de tráfego de protocolo em mais detalhes para o tráfego de entrada. Em termos gerais, ocorrem as seguintes etapas:

**5. Ocorrência de falha de cache: por ser a chegada de novos dados, as linhas de cache correspondentes no sistema de buffer que contém novos dados são invalidadas.**

**6. O cabeçalho é processado: o software de protocolo é executado no core para analisar os conteúdos TCP e de cabeçalho IP. Isso provavelmente vai incluir o acesso ao bloco de controle de transporte (TCB), que contém informação de conteúdo relacionada ao TCP. O acesso ao TCB pode ou não causar uma falha de cache, necessitando de acesso à memória principal.**

# Entrada/Saída

## Acesso direto à cache

- Para esclarecer a questão do desempenho e explicar o benefício de DCA como uma maneira de aprimorar o desempenho, vamos analisar o processo de tráfego de protocolo em mais detalhes para o tráfego de entrada. Em termos gerais, ocorrem as seguintes etapas:

7. **Transferência do bloco: a porção de dados do pacote é transferida a partir do buffer do sistema para o buffer de aplicação apropriado.**

# Entrada/Saída

## Acesso direto à cache

- Uma sequência similar de etapas ocorre para o tráfego de saída de pacote, mas há algumas diferenças que afetam o modo como a cache é gerenciada.
- Para o tráfego de saída, ocorrem as seguintes etapas:
  1. Solicitação de transferência de pacote
  2. Criação do pacote
  3. Chamada de uma operação de saída
  4. Transferência de DMA
  5. Sinais de NIC de finalização
  6. O driver libera buffer

# Entrada/Saída

## Acesso direto à cache

- Uma sequência similar de etapas ocorre para o tráfego de saída de pacote, mas há algumas diferenças que afetam o modo como a cache é gerenciada. Para o tráfego de saída, ocorrem as seguintes etapas:

  1. Solicitação de transferência de pacote: quando uma aplicação tem um bloco de dados para transferir a um sistema remoto, ela coloca os dados em um buffer de aplicação e alerta o SO com algum tipo de chamada de sistema.

# Entrada/Saída

## Acesso direto à cache

- Uma sequência similar de etapas ocorre para o tráfego de saída de pacote, mas há algumas diferenças que afetam o modo como a cache é gerenciada. Para o tráfego de saída, ocorrem as seguintes etapas:

  2. Criação do pacote: o SO invoca um processo de TCP/IP a fim de criar o pacote TCP/IP para transmissão. O processo do TCP/IP acessa o TCB (que pode envolver uma falha de cache) e cria cabeçalhos apropriados. Também lê dados a partir do buffer de aplicação e, então, coloca o pacote completo (cabeçalho mais dados) em um buffer do sistema. Observe que os dados que são escritos no buffer do sistema também existem na cache. O processo de TCP/IP também cria um descritor de pacote que é colocado na memória compartilhada com o módulo DMA.

# Entrada/Saída

## Acesso direto à cache

- Uma sequência similar de etapas ocorre para o tráfego de saída de pacote, mas há algumas diferenças que afetam o modo como a cache é gerenciada. Para o tráfego de saída, ocorrem as seguintes etapas:

  3. Chamada de uma operação de saída: usa um programa de driver de dispositivo para sinalizar ao módulo de DMA que a saída está pronta para o NIC.

  4. Transferência de DMA: o módulo de transferência de DMA lê o descritor de pacote, então a transferência de dados é realizada a partir da memória principal ou da cache de último nível ao NIC. Observe que as transferências de DMA invalidam a linha de cache em uma cache, mesmo no caso de uma leitura (pelo módulo de DMA). Se a linha estiver modificada, ela usa um *write back*. O core não faz as validações. As validações acontecem quando o módulo DMA lê os dados.

# Entrada/Saída

## Acesso direto à cache

- Uma sequência similar de etapas ocorre para o tráfego de saída de pacote, mas há algumas diferenças que afetam o modo como a cache é gerenciada. Para o tráfego de saída, ocorrem as seguintes etapas:

  5. Sinais de NIC de finalização: depois que uma transferência estiver completa, o NIC sinaliza ao driver no core que originou o sinal de envio.

  6. O driver libera buffer: uma vez que o driver recebe o aviso de finalização, ele libera espaço do buffer para reuso. O core deve também invalidar as linhas de cache que contém dados de buffer.

# Entrada/Saída

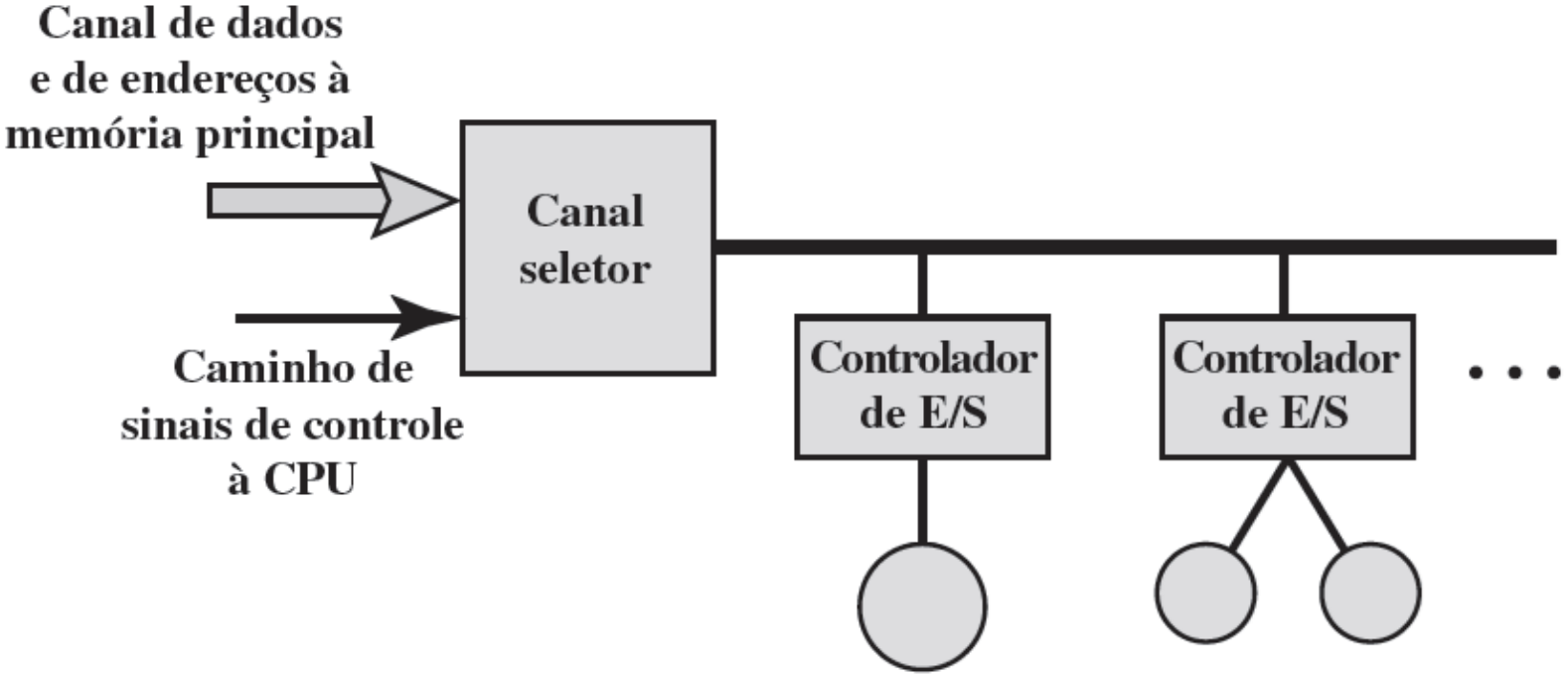
## Processadores e canais de E/S

- Enquanto se prossegue no caminho de evolução, cada vez mais a função de E/S é realizada sem envolvimento da CPU.
- A CPU fica cada vez mais livre das tarefas relacionadas a E/S, melhorando o desempenho.
- Ocorre uma grande mudança com a introdução do conceito de um módulo de E/S capaz de executar um programa.
- O módulo de E/S em geral é conhecido como um canal de E/S.

# Entrada/Saída

## Características dos canais de E/S

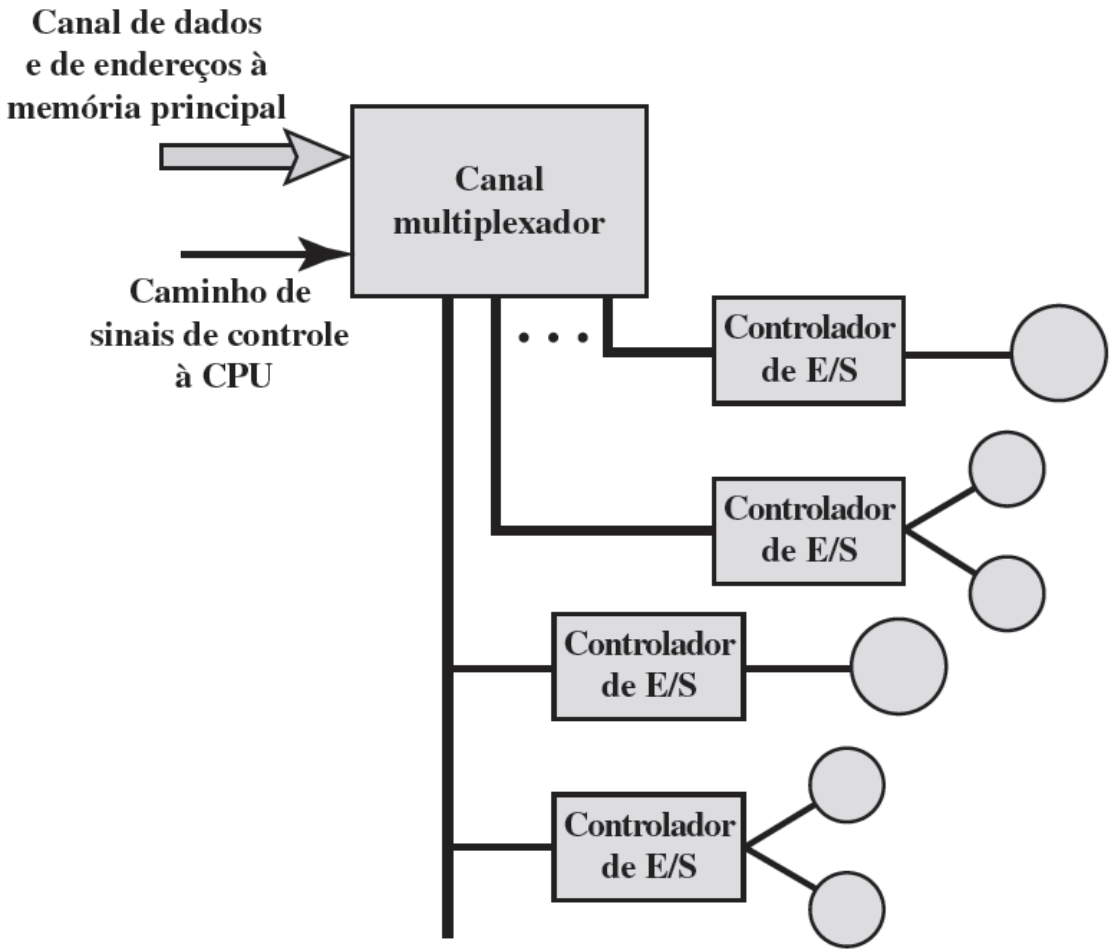
- Um canal seletor controla diversos dispositivos de alta velocidade e, a qualquer momento, é dedicado à transferência de dados com um desses dispositivos:



# Entrada/Saída

## Características dos canais de E/S

- Um Um canal multiplexador pode tratar da E/S com vários dispositivos ao mesmo tempo:



# Entrada/Saída

## Padrões de interconexão externa

- O USB é bastante usado para conexões periféricas.
- É a interface padrão para dispositivos de velocidade mais lenta, como teclado e dispositivos apontadores, mas também é comumente usada para E/S de alta velocidade, incluindo impressoras, drives de disco e adaptadores de rede.
- Ele vem de várias gerações.
- O sistema USB é controlado por um controlador host central, que é conectado aos dispositivos para criar uma rede local com uma topologia hierárquica em árvore.

# Entrada/Saída

## Padrões de interconexão externa

- O FireWire foi desenvolvido como uma alternativa para a interface SCSI.
- O FireWire usa uma configuração *daisy-chain*, com até 63 dispositivos conectados em uma única porta.
- Até 1.022 barramentos FireWire podem ser interconectados usando pontes.
- O FireWire permite o que é conhecido como conexão a quente (*hot plugging*), que significa que é possível conectar e desconectar periféricos sem ter que desligar o sistema de computação ou reconfigurar o sistema.

# Entrada/Saída

## Padrões de interconexão externa

- A interface SCSI é um padrão comum para conectar dispositivos periféricos (discos, modems, impressoras etc.) a computadores pequenos e médios.
- A organização física da SCSI é um barramento compartilhado, que pode suportar até 16 ou 32 dispositivos, dependendo da geração do padrão.
- A velocidade varia de 5 Mbps na especificação original de SCSI-1 a 160 Mbps na SCSI-3 U3.

# Entrada/Saída

## Padrões de interconexão externa

- InfiniBand é uma especificação de E/S, voltada para o mercado de servidores de ponta.
- O PCI Express é um sistema de barramento de alta velocidade que conecta periféricos de uma grande variedade de tipos e velocidades.
- Serial ATA é uma interface para sistemas de armazenamento de disco.
- Ethernet é uma tecnologia de rede predominantemente com fios, usada em casas, escritórios, centros de dados, empresas e redes de área ampla.

# Entrada/Saída

## Padrões de interconexão externa

- O Wi-Fi é uma tecnologia de acesso à internet predominantemente sem fio, usado em casas, escritórios e espaços públicos.
- O Wi-Fi em casa agora conecta computadores, tablets, smartphones e hosts de dispositivos eletrônicos.
- O Wi-Fi nas empresas tem se tornado um meio essencial para aumentar a produtividade dos colaboradores.
- Os hotspots de Wi-Fi público expandiram-se de modo significativo para proporcionar acesso livre à internet em locais públicos.