



# Organização de Computadores Digitais

## Capítulo 7 - Entrada e Saída



# Tópicos

- o Introdução
- o Dispositivos Externos
- o Módulos de E/S
- o E/S Programadas
- o E/S Ativadas por Interrupções
- o DMA
- o Canais e processadores de E/S



# Entrada e Saída

- Ha vários anos...
- E/S descrevia os componentes e dispositivos para além do processador e da memória
  - O desempenho não era um dos fatores mais importantes
  - Os dispositivos eram caracterizados como sendo não prioritários, usados infrequentemente e lentos
- As atenções estavam viradas para a área de swap do disco

0 0 0

Atualmente

# Entrada e Saída

- o Armazenamento secundário
  - o Discos, fitas, CD ROM, DVD
- o Redes
- o Human Interface
  - o Vídeo, áudio, teclados
  - o Interfaces com o "mundo real"
    - o Temperatura, pressão, posição, velocidade ...
  - o Processamento Digital de Sinais

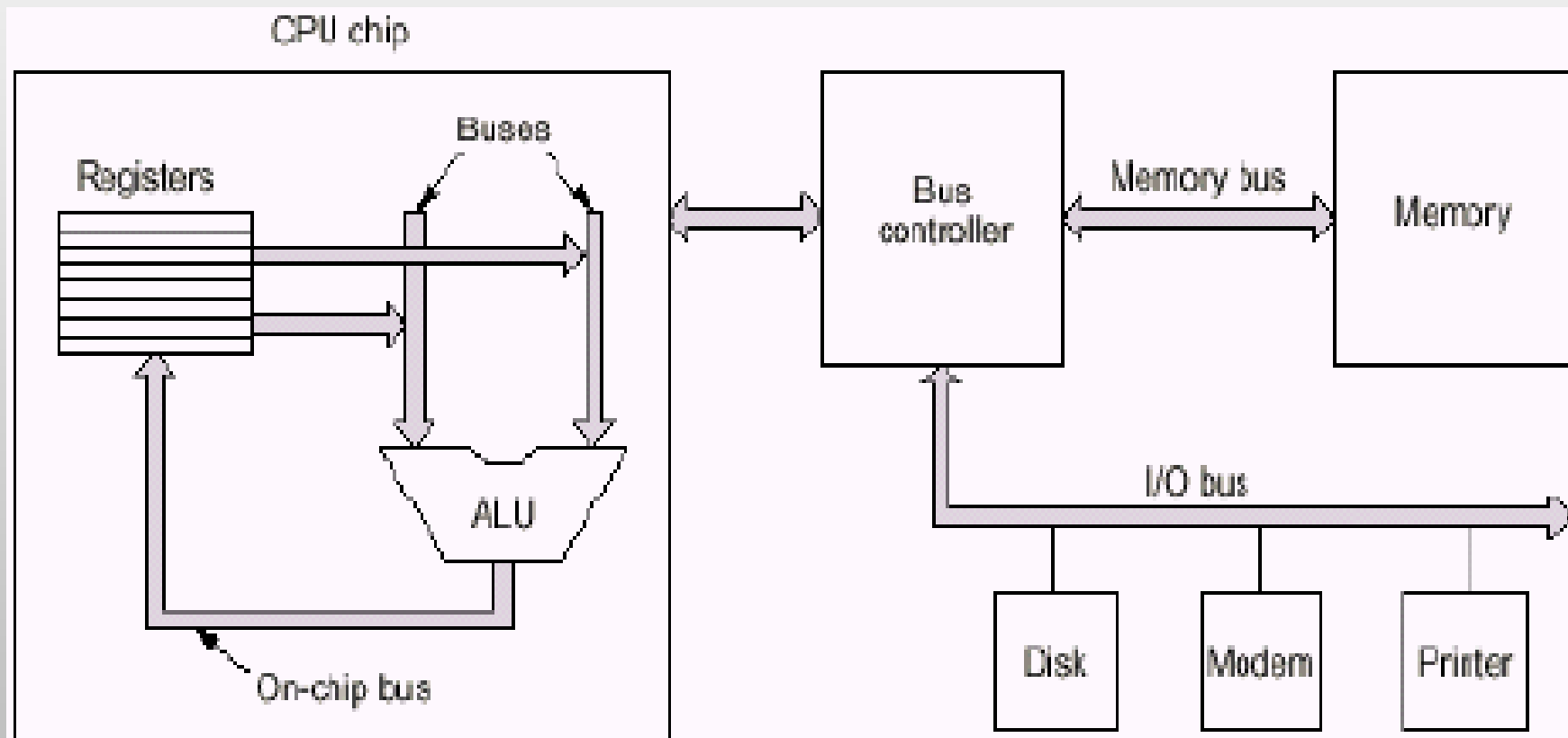


# O Barramento do Sistema

- Para além da **CPU** e da **memória**, o terceiro elemento de um sistema de computador é o conjunto de módulos de **Entradas e Saídas**
- Cada módulo liga-se com ao barramento (bus) do sistema ou com o comutador central e controla um ou mais dispositivos periféricos
- O bus do sistema é usados para ligar dois ou mais elementos do sistema no computador
- Conjunto de linhas paralelas que transportam sinais usados para comunicação entre elementos

0 0 0

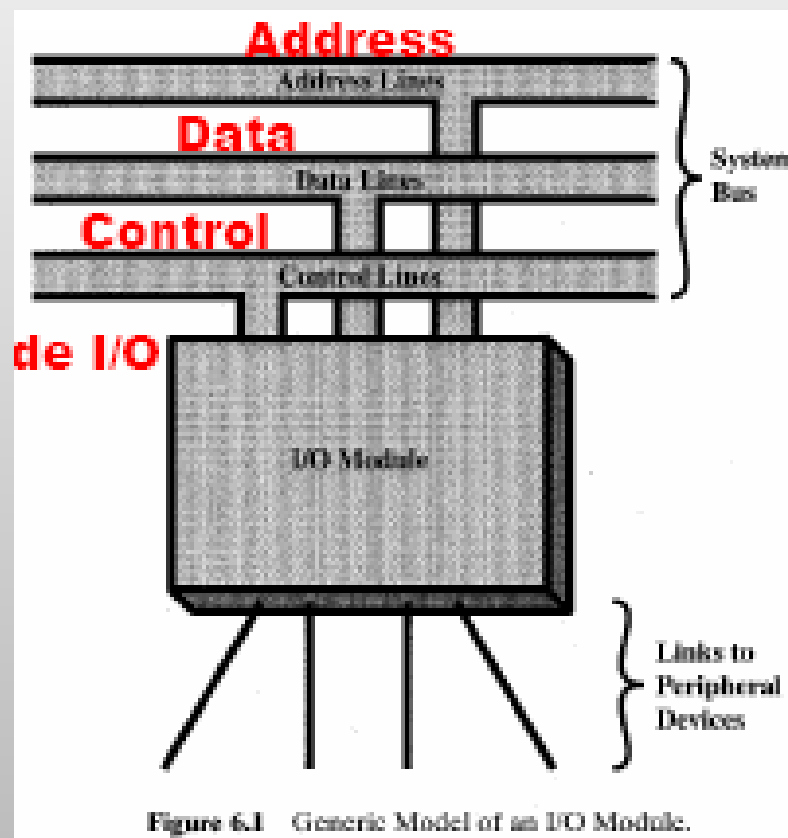
# O Barramento do Sistema



0 0 0

# Módulos de I/O (E/S)

- Os módulos de I/O são dispositivos que possuem alguma "inteligência"
- Possuem funcionalidades lógicas capazes de estabelecer mecanismos de comunicação entre os diferentes periféricos e o bus de sistema





# Problemas relacionados com os

## Dispositivos de I/O (E/S)

- o Cada módulo possui uma interface com o bus do sistema e controla um ou mais dispositivos
- o Os dispositivos de entradas e saídas **não são simples conectores** que se ligam diretamente ao bus do sistema
- o As razões para que isto suceda são as seguintes:
  - o Diferentes níveis de sinal elétrico
  - o Velocidade
  - o Sincronização
  - o Códigos e formatos





# Problemas relacionados com os Dispositivos de I/O (E/S)

- o **Diferentes níveis de sinal elétrico** Muitos equipamentos periférico não funciona com os níveis internos de um computador
- o **Velocidade** A maior parte dos periféricos tem escalas de tempo de ordens de grandeza abaixo dados processadores.
- o **Sincronização** Os periféricos não estão sincronizados com os processadores. Tem que existir protocolos de funcionamento que aliviem o processador da necessidade de se sincronizar com periféricos



# Problemas relacionados com os Dispositivos de I/O (E/S)

- o **Códigos e formatos** Nem todos os periféricos usam os códigos que o processador utiliza para codificar os vários tipos de informação e, do mesmo modo, os formatos em que a informação é utilizada por cada um pode é diferente entre si e do processador
- o **Controle do funcionamento** Por fim é útil dispensar o processador das tarefas de controle detalhado dos periféricos (manipular as partes mecânicas dos periféricos, etc.)

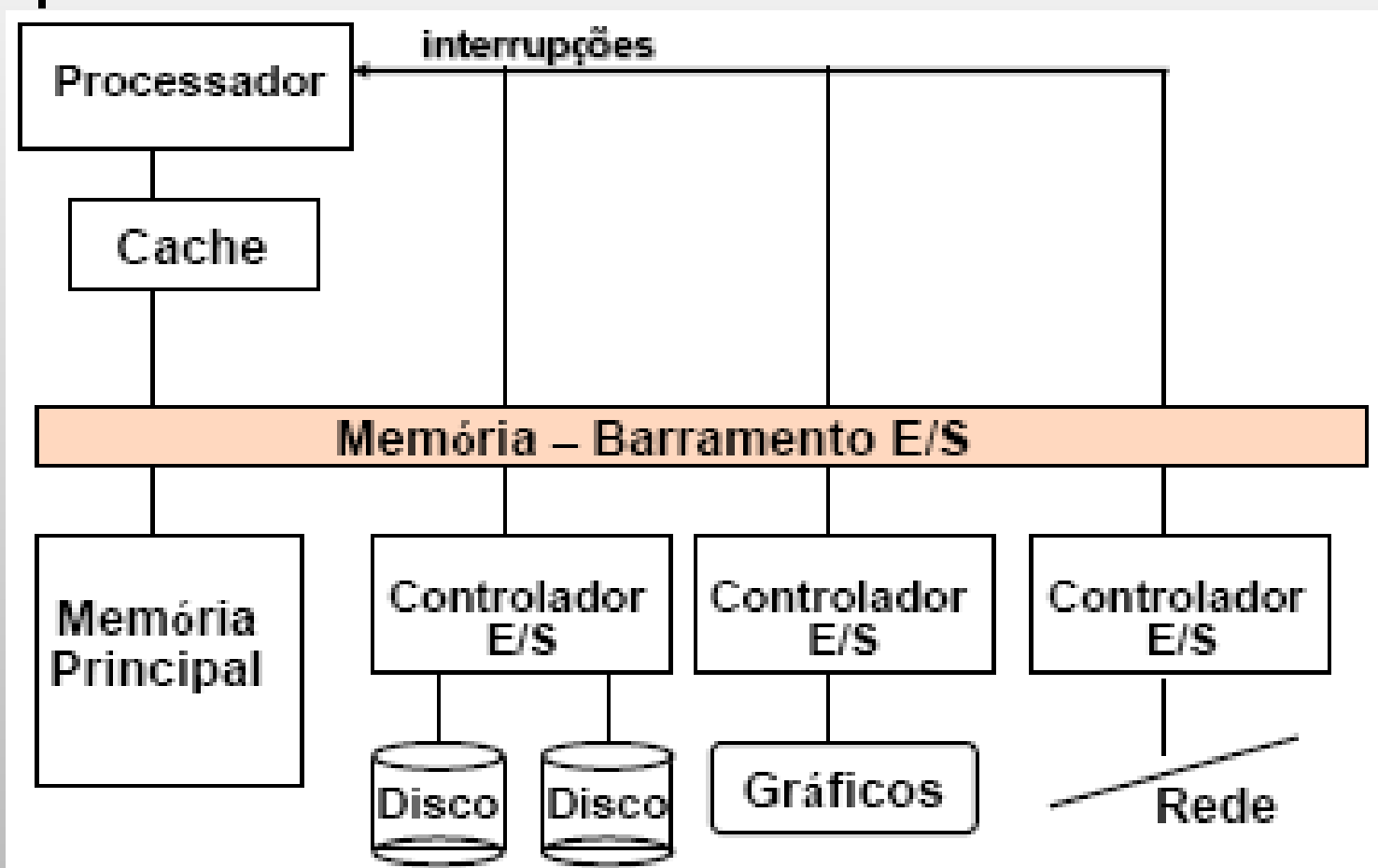


## Módulos de I/O (E/S)

- Por estas razões usa-se um módulo de Entradas e Saídas que tem as seguintes funções:
  - Fornecer uma interface com a CPU e a memória via o bus de sistema ou comutador central (central switch)
  - Permitir a interface com um ou mais dispositivos periféricos

0 0 0

# Sistemas de I/O

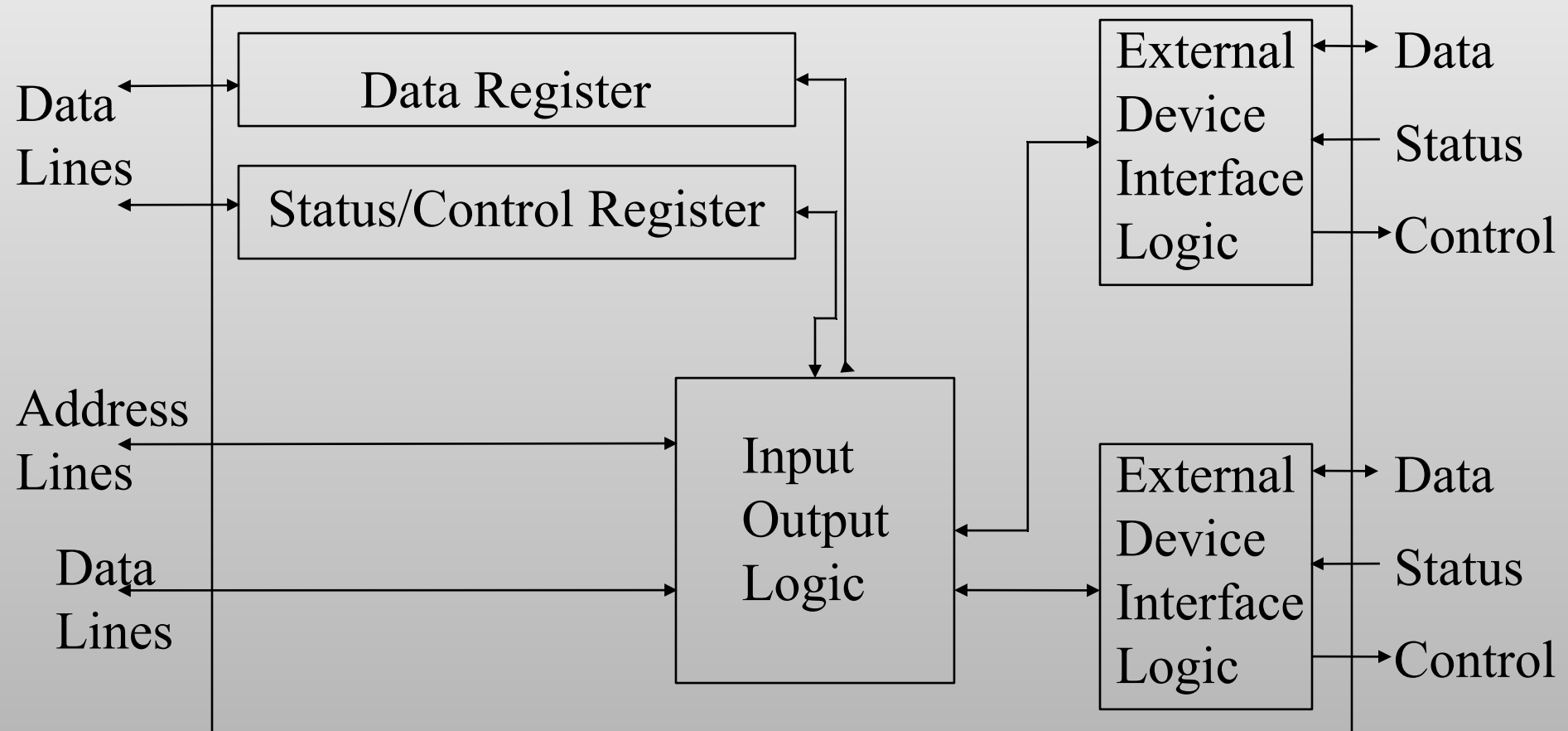


0 0 0

# Diagrama do Módulo de I/O

Systems Bus Interface

External Device Interface





# Dispositivos Externos de I/O

- o Um dispositivo externo é ligado ao computador através de uma ligação a um modulo de I/O
- o Human-readable
  - o Comunicação com o usuário (impressoras, terminais vídeo, etc.)
- o Machine-readable
  - o Comunicação com equipamento (discos, sensores, etc.)
- o Communication
  - o Comunicação com dispositivos remotos (terminais, placas de redes)

# Dispositivos Externos de I/O





# Problemas de I/O

- o The I/O GAP
- o O poder de processamento duplica todos os 18 meses
- o O tamanho da memória duplica todos os 18 meses  
(4X/3anos)
- o A capacidade dos discos duplica todos os 18 meses
- o O velocidade de posicionamento num disco (Seek + Rotate) duplica todos os 10 anos





# Módulos de Entradas/Saídas

Entidade responsável:

- o Pelo controle de um ou mais dispositivos externos e
- o Pela transferência de dados entre esses dispositivos e a memória e/ou registradores do CPU



# Módulos de Entradas/Saídas

## Funções

- Controle e Temporização
  - Controla o tráfego de dados entre os recursos internos e os dispositivos externos
- Comunicação com o CPU
  - Descodificação de comandos, transferência de dados, relato de estado e reconhecimento de endereços
- Comunicação com os dispositivos
  - Comandos, informação de estado e transferência de dados



# Módulos de Entradas/Saídas

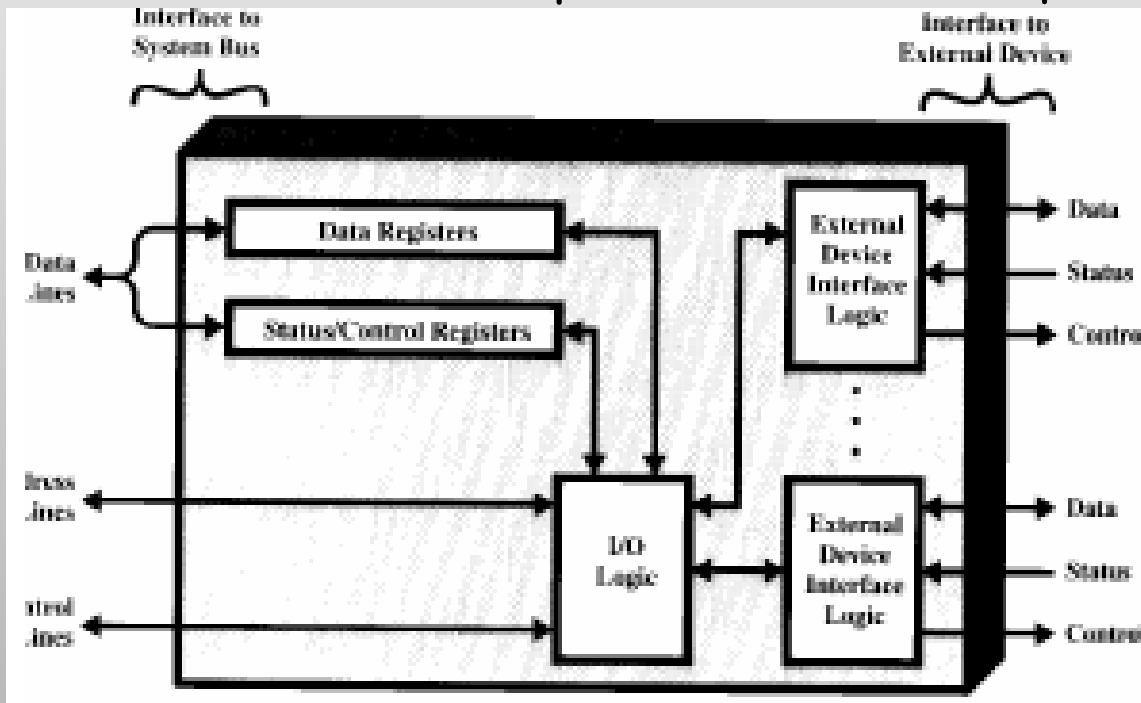
## Funções

- o Buffering de dados
  - o Espaços que permitem manter o ritmo de transferência
  - o A CPU e a memória tem taxas de transferência de dados elevadas quando comparado com os dispositivos de periféricos
- o Detecção de Erros
  - o Relato de erros ao CPU

0 0 0

# Estrutura dos Módulos

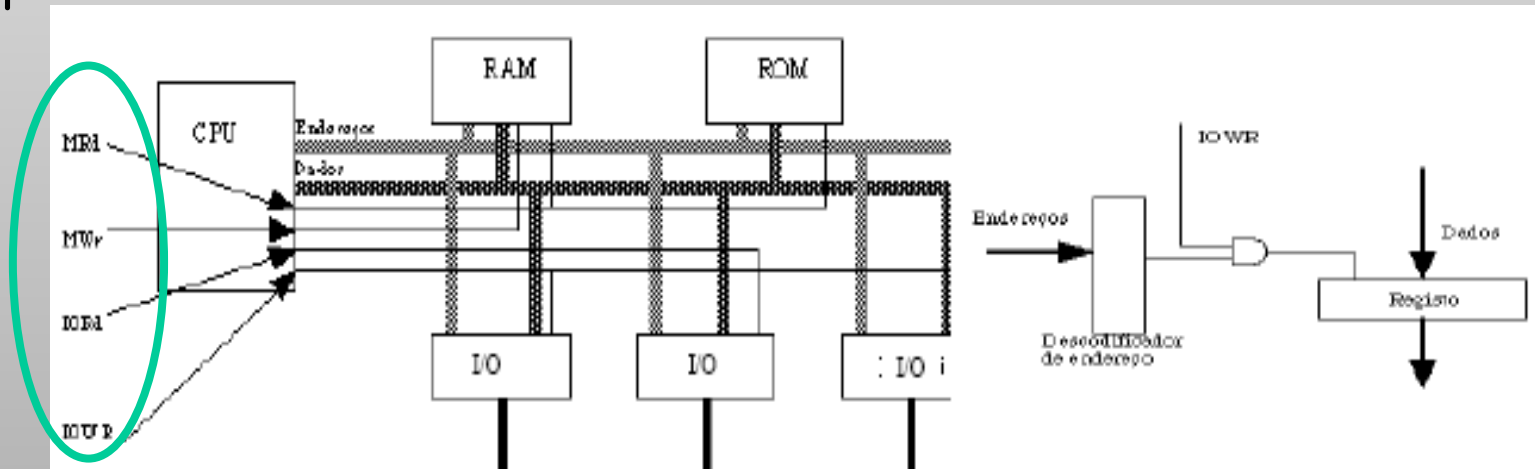
- Os módulos de I/O variam consideravelmente em complexidade tendo em conta o número de dispositivos externos que controlam



O módulo de I/O liga-se ao resto do computador através de um conjunto de linhas de sinais

# Estrutura dos Módulos

- Repare-se na existência das linhas MRd, MWr, IORd e IOWr que especificam o tipo de operação que está ocorrendo em cada momento no barramento
- Na porta, além do registrador onde vai ser escrito o conteúdo do barramento de dados, tem que ser feita a descodificação do endereço da porta





# Estrutura dos Módulos

- Os dados transferidos de e para o módulo são guardados em um ou mais registradores de dados
- A lógica usada nos módulos interage com CPU através de um conjunto de linhas de controle (e.gBUS)
- O módulo tem também de estar apto a reconhecer e gerar endereços associados aos diversos mecanismos de controle
- Cada módulo de entradas e saídas tem um único endereço ou, se cada módulo controlar mais que um dispositivo, possuirá então um conjunto finito de endereços

# Estrutura dos Módulos

- O módulo de entradas e saídas, contém lógica específica para a interface de cada um dos dispositivos que ele controla
- Existem três técnicas de operação com os módulos de I/O:
  - Entradas e saídas programadas
  - Entradas e saídas comandadas por interrupções
  - Direct Memory Access (DMA)

Table 6.3 I/O Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)



# Técnicas de operação de com os módulos de I/O

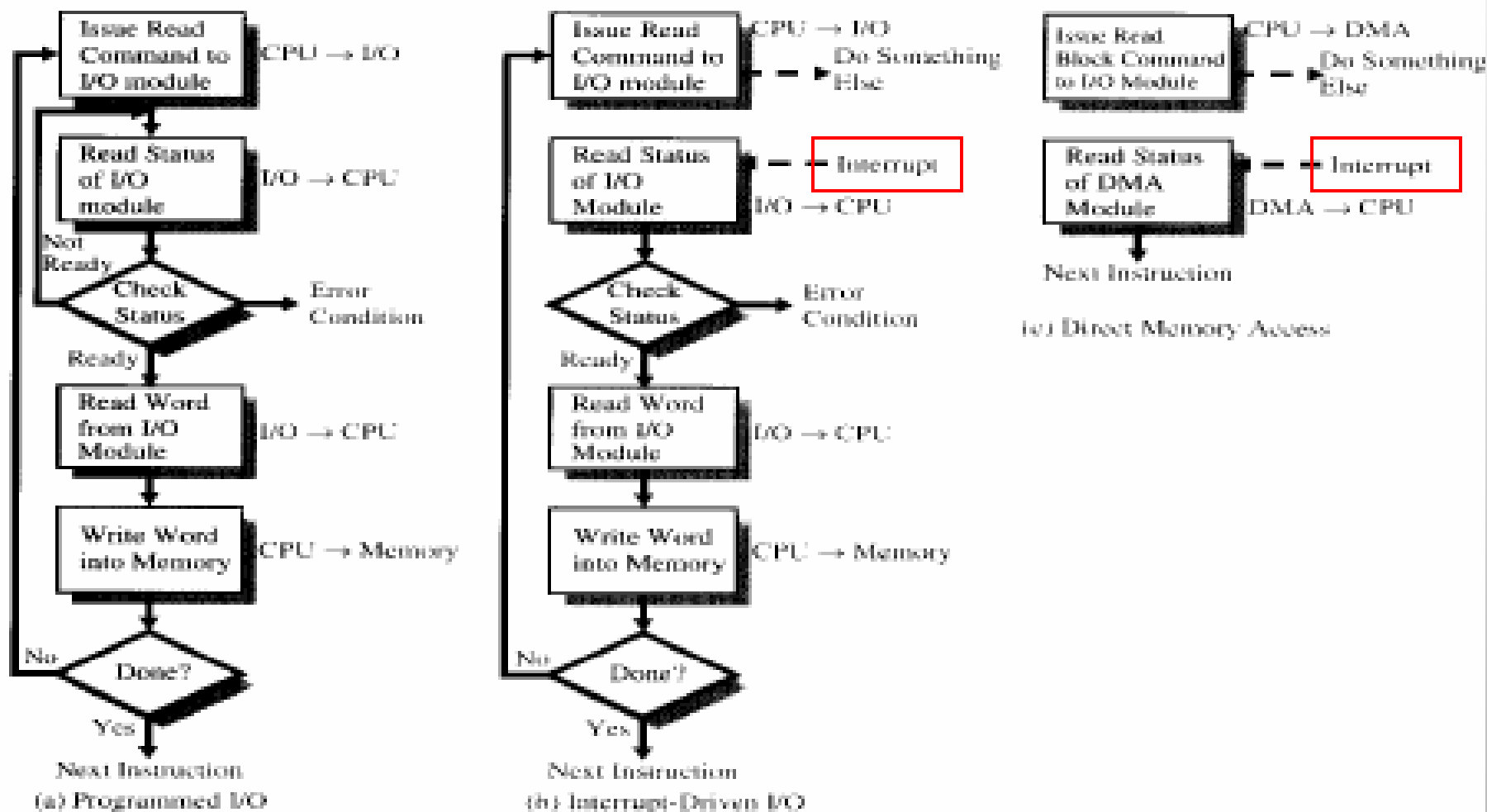
Entrada e Saídas Programada

Entradas e Saídas Comandadas por  
Interrupções

Diret Memory Access (DMA)



# Operação com os Módulos de I/O



**Figure 6.5** Three Techniques for Input of a Block of Data.



# Entrada e Saídas Programadas

- Os dados são transferidos entre a CPU e o módulo de E/S
- A CPU executa um programa que lhe dá controle direto da operação das entradas e saídas incluindo a monitoração do estado, envio de comandos de leitura e escrita e transferência de dados
- A CPU tem de esperar que a operação de entradas e saídas termine
- Se o processador for mais rápido que I/O, existe uma perda de tempo do processador



# Entrada e Saídas Programadas

## Detalhes

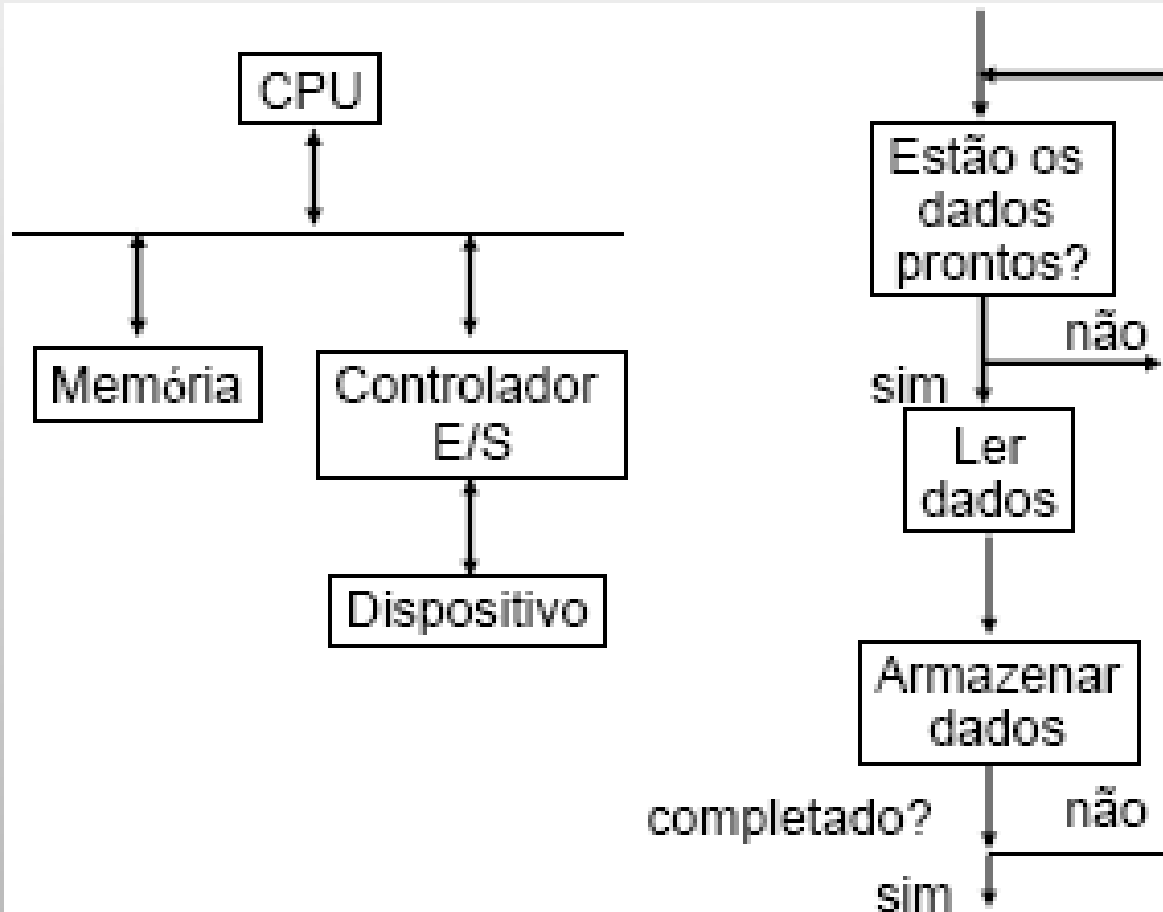
A CPU requer uma operação de E/S

- O módulo de E/S realiza a operação
- O módulo de E/S atualiza o bit de estado
- A CPU verifica o bit de estado periodicamente
- O módulo de E/S não informa a CPU diretamente
- O módulo de E/S não interrompe a CPU
- A CPU pode esperar ou voltar a tentar mais tarde

o o o

# Entrada e Saídas Programadas

## Detalhes



Este ciclo de espera não é uma forma eficiente de usar a CPU, a não ser que o dispositivo seja muito rápido!

0 0 0

# Entrada e Saídas Programadas

## Escrita/Leitura Programada

```
check status register  
while(not ready) {  
    check status register  
}  
set control register  
write to data register
```

```
set control register  
check status register  
while(not ready) {  
    check status register  
}  
read from data register
```



# Entrada e Saídas Programadas

## Comandos das I/O

- Para executar uma instrução de I/O, a CPU envia um endereço (especificando um dispositivo externo) e um comando de I/O
- Existem 4 tipos de comandos de I/O:
  - Controle: ativa o periférico e diz o que fazer
  - Teste: testa as várias condições de estado associadas a um módulo de I/O e aos seus periféricos
  - Leitura: faz com que o módulo obtenha dados do periférico e os coloque no buffer
  - Escrita: faz com que o módulo leia uma palavra do bus e a envie para o periférico



# Entrada e Saídas Programadas

## Instruções de I/O

- o A cada dispositivo é associado um identificador ou endereço distinto
- o Quando o processador, a memória e os módulos de I/O partilham o mesmo bus, existem dois modos de endereçamento possíveis:
  - o Memory-mapped I/O
  - o Isolated I/O



## Instruções de I/O -Memory-mapped

### I/O

- o Existe um único espaço de memória para a localização da memória e do módulos de I/O
- o As I/O funcionam como a leitura/escrita na memória
- o Não existem comandos específicos para I/O





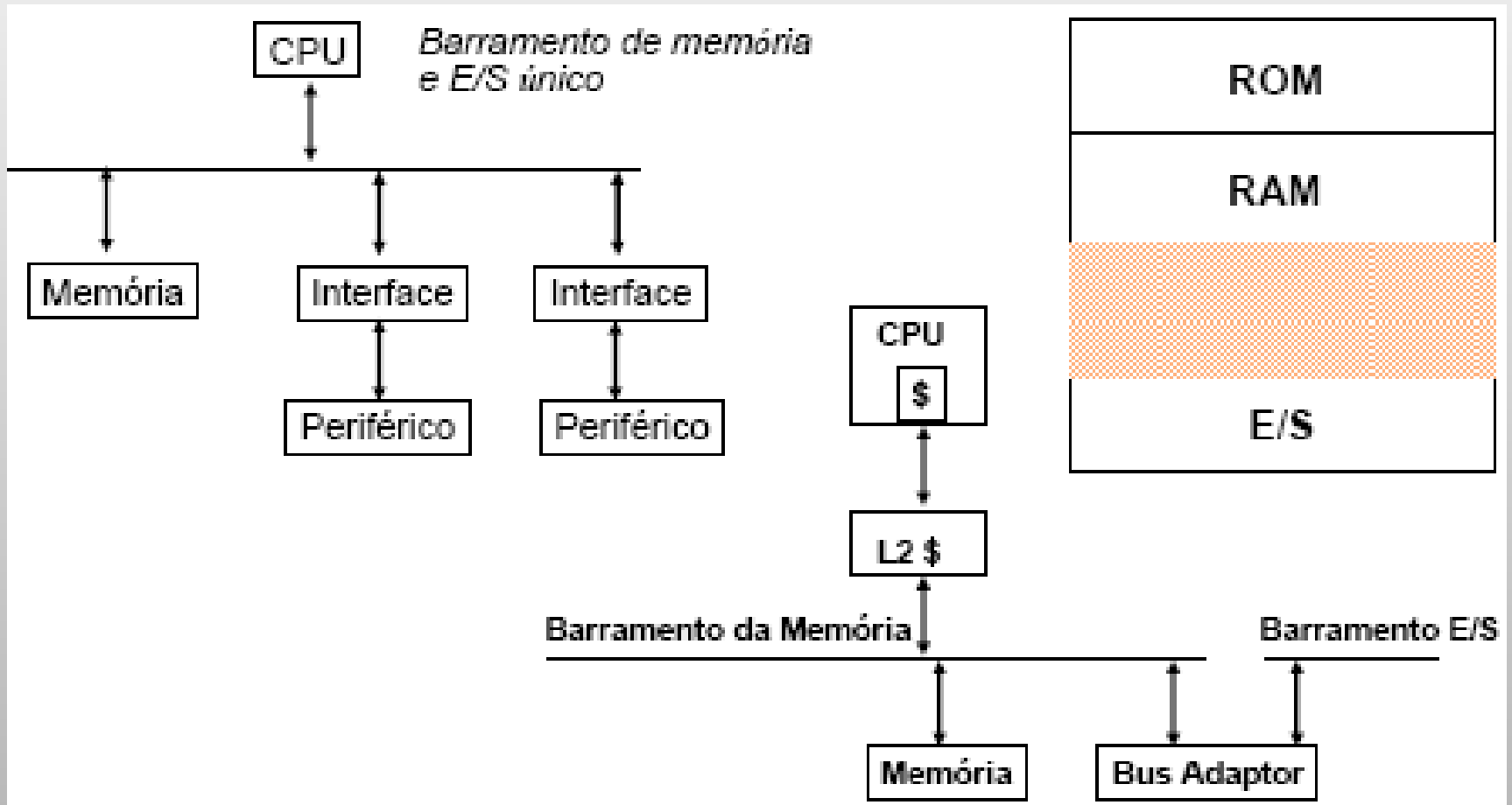
## Instruções de I/O -Memory-mapped

### I/O

- o O processador interpreta os registradores de estado e dados como localizações de memória e utiliza as mesmas instruções máquina para acessar tanto à memória como aos dispositivos de I/O
- o Por exemplo, com 10 linhas de endereço, existe um total de  $2^{10}=1024$  endereços de localizações de memória e de módulos de I/O

0 0 0

# Instruções de I/O -Memory-mapped I/O





# Entrada e Saídas Programadas

## Instruções de I/O -Isolated I/O

- o Uma simples linha de escrita e de uma outra de leitura são necessárias no bus
- o Necessita de linhas de selecção de I/O ou de memória
- o Uma terceira linha de comando especifica se o endereço se refere a uma posição de memória ou se a um dispositivo de I/O
- o Tanto a memória como os módulos de I/O têm o mesmo número de possibilidades de endereçamento



# Entrada e Saídas Programadas

## Instruções de I/O

- o Se se usar o **Isolated I/O**, dispomos apenas de um conjunto limitado de instruções
- o O **Memory-mapped I/O** permite a utilização de um grande repertório de instruções, permitindo uma maior eficiência na programação
- o Uma desvantagem deste sistema tem a ver com a utilização do espaço de endereçamento valioso que é a memória



# I/O Comandada por Interrupções

- o Uma alternativa à I/O programada é a CPU enviar um comando de I/O para um módulo e continuar a execução, este interromperá então o CPU quando estiver pronto para transferir dados
- o A CPU
  - o A CPU envia um comando de leitura e continuar a execução
  - o No fim de cada instrução a CPU testa as interrupções e quando detecta uma interrupção do módulo,
  - o lê os dados do bus e guarda-os em memória



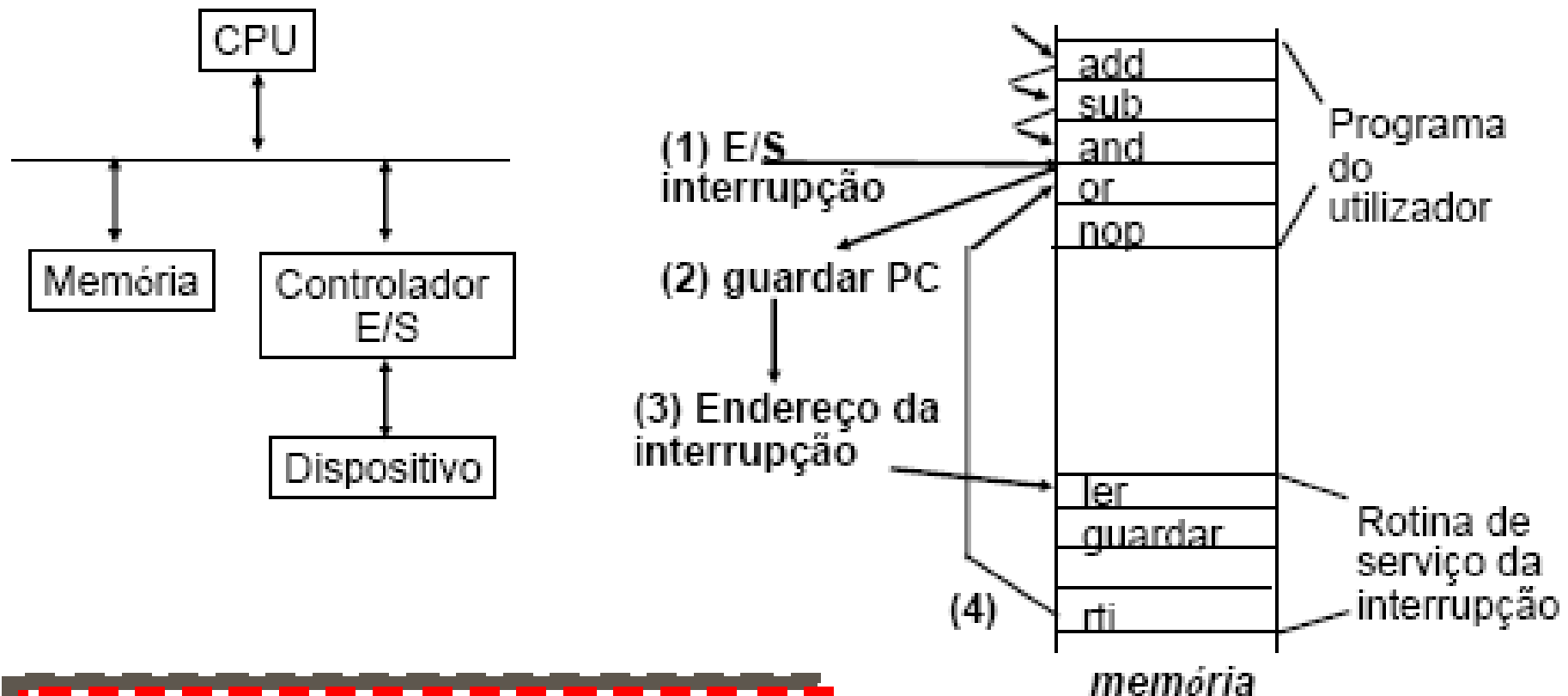
# I/O Comandada por Interrupções

- o O módulo
  - o O módulo recebe um comando de leitura da CPU e lê os dados do periférico associado
  - o Quando os dados estão no registrador de dados do módulo, este interrompe a CPU e aguarda que os dados sejam requisitados, colocando-os no bus

0 0 0

# I/O Comandada por Interrupções

## O Processamento das Interrupções



O programa só pa'ra durante a transferência dos dados



# I/O Comandada por Interrupções

## O Processamento das Interrupções

- 1.O dispositivo envia um sinal de interrupção ao processador
- 2.O processador acaba a instrução atual antes de responder à interrupção
- 3.O processador testa a existência de interrupções e comunica ao dispositivo um sinal de acknowledgement ao dispositivo que enviou a interrupção, para que este retire o sinal de interrupção





# I/O Comandada por Interrupções

## O Processamento das Interrupções

4.O processador prepara a transferência do controle para a rotina de interrupção

- a. Guarda o estado do processador (que esta no registrador PSW - program status word)
- b. Guarda a localização da próxima instrução a ser executada (que está guardada no program counter)
- c. Esta informação pode ser salva na stack do sistema



# I/O Comandada por Interrupções

## O processamento das interrupções

5. O processador carrega no PC a localização do interrupt-handling programa que responde à interrupção
6. Um novo contexto é alcançado com a execução da rotina de tratamento da interrupção
7. À medida que a interrupção é processada, é efetuada uma verificação da ocorrência ou não de outras interrupções. Isto pode envolver o envio adicional de comandos para os dispositivos de I/O



## I/O Comandada por Interrupções

### O processamento das interrupções

8. Quando a interrupção acabou de ser tratada, é necessário recuperar o contexto do sistema que tinha sido salvado na pilha (stack)
9. O programa continua a sua execução a partir da instrução que tinha sido guardada no PC

0 0 0

# I/O Comandada por Interrupções

## O Processamento das Interrupções

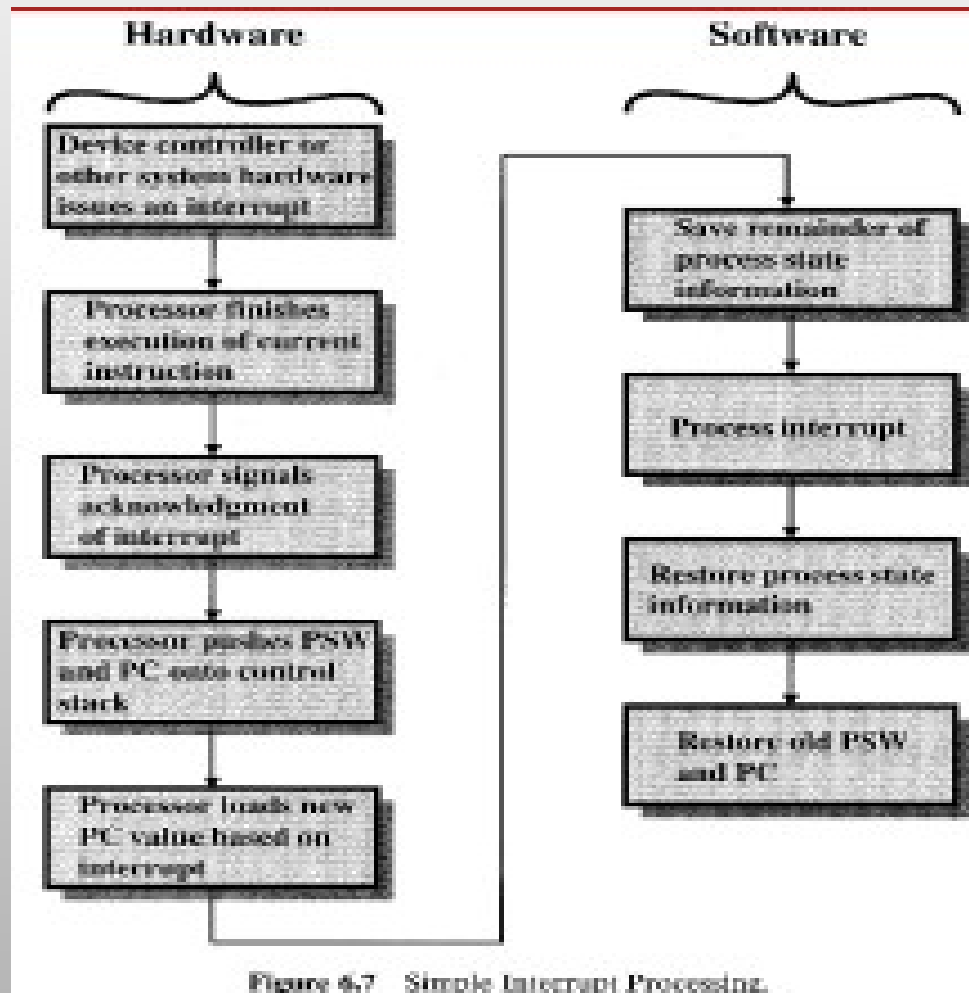


Figure 4.7 Simple Interrupt Processing.



# I/O Comandada por Interrupções

## Questões de Projeto

- Como identificar o módulo que emitiu uma interrupção?
  - Múltiplas Linhas de Interrupção
  - Identificação por Software
  - Daisy Chain
  - Bus Master
- Como gerir várias interrupções?
  - i.e., as ações de resposta a uma interrupção e' interrompida por uma outra interrupção

# I/O Comandada por Interrupções

## Módulo de Identificação de Interrupções (1)

### Múltiplas Linhas de Interrupção

- o Na prática apenas um pequeno número de linhas do Barramento podem ser usadas para as linhas de interrupção
- o Limita o número de dispositivos



# Módulo de Identificação de Interrupções (2)

## Identificação por Software

- A rotina de tratamento de uma interrupção interroga cada módulo de I/O
- Uma linha de comando especial (por exemplo, TEST I/O) pode ser usado para isso
- O módulo de I/O correspondente responde afirmativamente caso tenha enviado a interrupção
- Alternativamente, cada módulo pode conter um registrador de estado endereçável, que é lido pela CPU para identificar o módulo que causou a interrupção
- Desvantagem: consome muito tempo



## Módulo de Identificação de Interrupções (3)

### Daisy Chain

- o Identificação por hardware, usando uma ligação entre os módulos, na forma de uma cadeira circular
- o Quando a CPU recebe um sinal de interrupção, envia um sinal de confirmação de interrupção para a DaisyChain
- o O módulo responsável coloca uma palavra (vetor de interrupção) nas linhas de dados
- o O vetor de interrupção consiste no endereço do módulo de I/O ou algum outro tipo de identificador do módulo
- o A CPU utiliza o vetor para identificar o procedimento para tratar a interrupção





# Módulo de Identificação de Interrupções (4)

## Arbitração do Barramento

- Tal como a técnica anterior, a arbitração do barramento usa interrupções vetoradas
- Antes de lançar uma interrupção, o módulo tem de requisitar o bus
- Desta forma, só um módulo de I/O pode ativar a linha de interrupção de cada vez
- Quando a CPU detecta a interrupção, ele responde por meio da linha de reconhecimento de interrupção
- Então, o módulo responsável coloca um vetor no bus



# Acesso Direto à Memória

## (DMA-Direct Memory Access)

- Problemas da I/O programada e comandada por interrupções:
  - O ritmo de transferência é limitado pela velocidade com que a CPU consegue testar e servir o dispositivo
  - A CPU está ocupada a gerir a transferência, i.e. tem sempre de executar um n.º de instruções por cada transferência



# Acesso Direto à Memória

## (DMA-Direct Memory Access)

- o O DMA é uma técnica eficiente para a transferência de grande volumes de dados
- o O DMA envolve um módulo adicional no bus do sistema, o controlador de DMA
- o O controlador de DMA é capaz de imitar o processador
- o A CPU continua a sua execução e a operação é delegada ao módulo DMA que transfere o bloco de dados palavra-a-palavra
- o Quando o módulo de DMA precisa de tomar o controle do bus para transferir os dados, suspende temporariamente a operação da CPU (cycle-stealing)



# Acesso Direto à Memória

## (DMA-Direct Memory Access)

Quando o processador precisa de ler ou escrever um bloco de dados, envia um comando ao módulo de DMA, enviando a seguinte informação:

- O tipo de pedido (se leitura ou escrita), utilizando as linha de controle entre o processador e o módulo de DMA
- O endereço do dispositivo de I/O envolvido, comunicado através de linhas de dados
- O início da posição de memória a ler ou escrever, comunicados através das linhas de dados e guardados pelos módulos de DMA nos registradores de endereço
- O número de palavras a serem lidas ou escritas, mais uma vez, comunicadas através das linhas de dados e guardadas no registrador contador de dados

0 0 0

# Acesso Direto à Memória

## (DMA-Direct Memory Access)

- Pedido de leitura/escrita;
- Endereço do dispositivo;
- Endereço de memória inicial;
- N.º de palavras.

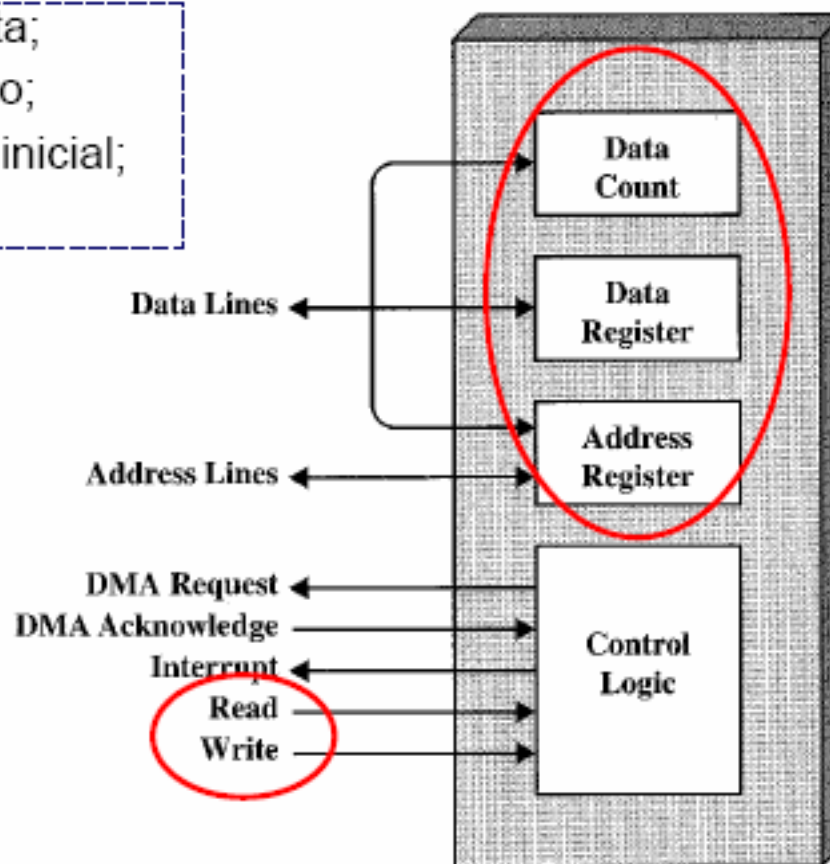


Figure 6.12 Typical DMA Block Diagram.



# Acesso Direto à Memória

## (DMA-Direct Memory Access)

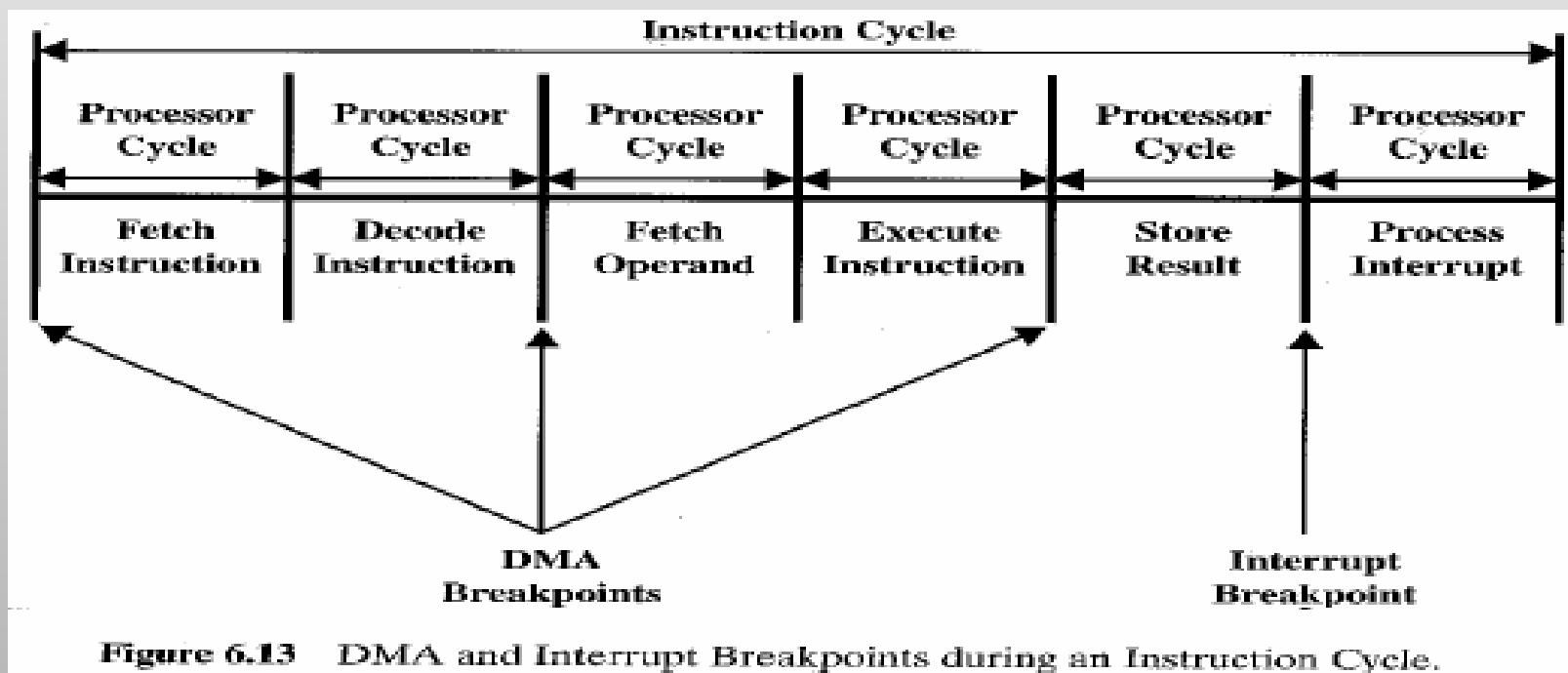
- O módulo de DMA pode usar o bus de sistema quando o processador não precisa dele
- O módulo de DMA transfere os blocos inteiros de dados, uma palavra de cada vez, diretamente e para a memória, sem passar pelo processador
- Quando a transferência estiver completa, o módulo de DMA envia um sinal de interrupção para o processador
- O processador é envolvido apenas no início e no fim da transferência

0 0 0

# Acesso Direto à Memória

## (DMA-Direct Memory Access)

- o A figura seguinte mostra onde e que o ciclo de processador pode ser suspenso
- o Em cada caso, o processador é suspenso antes de precisar de utilizar o bus





# Acesso Direto à Memória

## (DMA-Direct Memory Access)

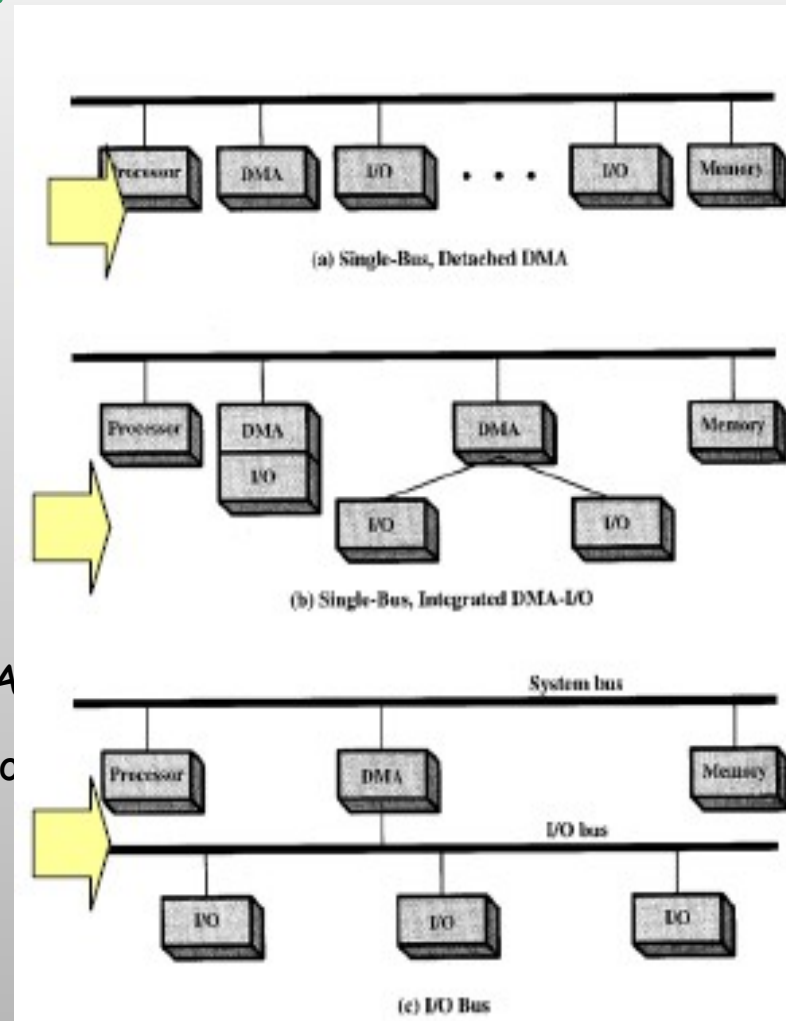
- o Note-se que não se trata de uma interrupção, o processador não salva nenhum contexto apenas executa uma outra função
- o O que pode acontecer é que o processador pare por um ciclo de busNa generalidade o processador pode executar as instruções mais lentamente
- o Em sistemas de transferência de palavras de I/O múltiplos, o DMA torna-se muito mais eficiente do que os mecanismos anteriores



# Acesso Direto à Memória

## (DMA-Direct Memory Access)

- a) Todos os módulos partilham a mesma estrutura de bus. Esta configuração, embora mais barata, torna-se ineficiente pois cada transferência ocupa dois ciclos de bus (para ler e escrever uma palavra)
- b) Os módulos de I/O estão diretamente ligados ao módulo DMA
- c) Utilização de um bus de I/O. Apenas o módulo de DMA partilha o bus de sistema. A partilha de dados entre o DMA e os módulos de I/O têm lugar fora do bus de sistema



# Canais de I/O e Processadores de I/O-Evolução da Função de E/S

- A introdução de capacidade de processamento nos módulos de I/O permitiu que a CPU delegue no módulo de I/O a execução de uma sequência de atividades de I/O, sendo interrompido apenas quando a sequência termina
- Esta evolução levou ao aparecimento de canais de I/O que executam programas de I/O guardados em memória pela CPU e finalmente aos processadores de I/O que incluem memória local

# Canais de I/O e Processadores de I/O-Evolução da Função de E/S

- O módulo de E/S é aprimorado, tornando-se um processador com um conjunto especializado de instruções de E/S
- A CPU envia um comando para o processador de E/S, que executa um programa de E/S carregado na memória
- O processador de E/S busca e executa instruções sem intervenção da CPU
- Isso possibilita à CPU especificar uma sequência de atividades de E/S a serem executadas e apenas ser interrompida quando toda a sequência é completada



# Canais de I/O e Processadores de I/O-Evolução da Função de E/S

- O módulo de E/S inclui uma memória local própria e é, portanto, ele próprio, um computador
- Essa arquitetura possibilita controlar grande número de dispositivos de E/S, com o mínimo envolvimento da CPU
- Ela normalmente é usada para controlar a comunicação com terminais interativos
- O processador de E/ S cuida da maior parte das tarefas envolvidas no controle dos terminais



# Canais de I/O e Processadores de I/O-Evolução da Função de E/S

- Ao longo desse caminho evolutivo, mais e mais funções de E/S são desempenhadas sem o envolvimento da CPU
- A CPU é, cada vez mais, liberada de tarefas relacionadas a E/S, o que contribui para melhorar o desempenho global do sistema
- Nas etapas apresentadas anteriormente ocorre uma mudança importante com a introdução do conceito de um módulo de E/S capaz de executar um programa
- O módulo de E/S com essas características é frequentemente denominado canal de E/S
- Um canal de E/S representa uma extensão do conceito de DMA

o o o

# Canais de I/O e Processadores de I/O-Tipos de Canais de I/O

Existem dois tipos de canais de I/O comuns:

- o **Seletor channel** -Controla múltiplos dispositivos de alta velocidade e em cada altura está dedicado à transferência de dados de 1 dispositivo
- o **Multiplexor channel** -Pode controlar vários dispositivos ao mesmo tempo

