Problemas com Entrada e Saída

- **Periféricos possuem características diferentes**
  - Geram diferentes quantidades de dados
  - Em velocidades diferentes
  - Em formatos diferentes
- **Periféricos são mais lentos que UCP e Memória**
- **Necessita-se de módulos de Entrada/Saída**

Módulo de Entrada/Saída

- **Interface para UCP e memória**
- **Interface para um ou mais periféricos**

Modelo Genérico de um Módulo de E/S

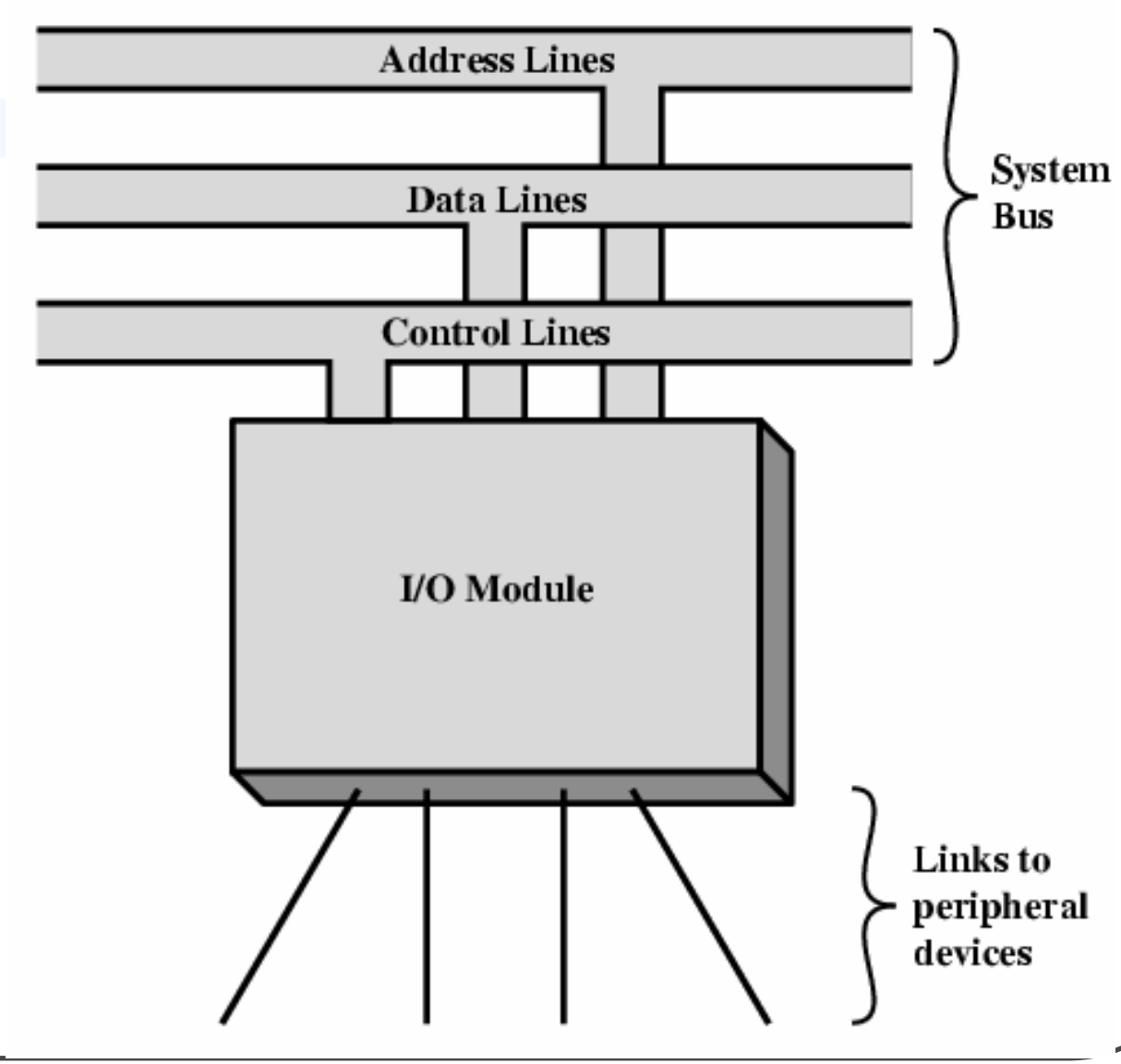
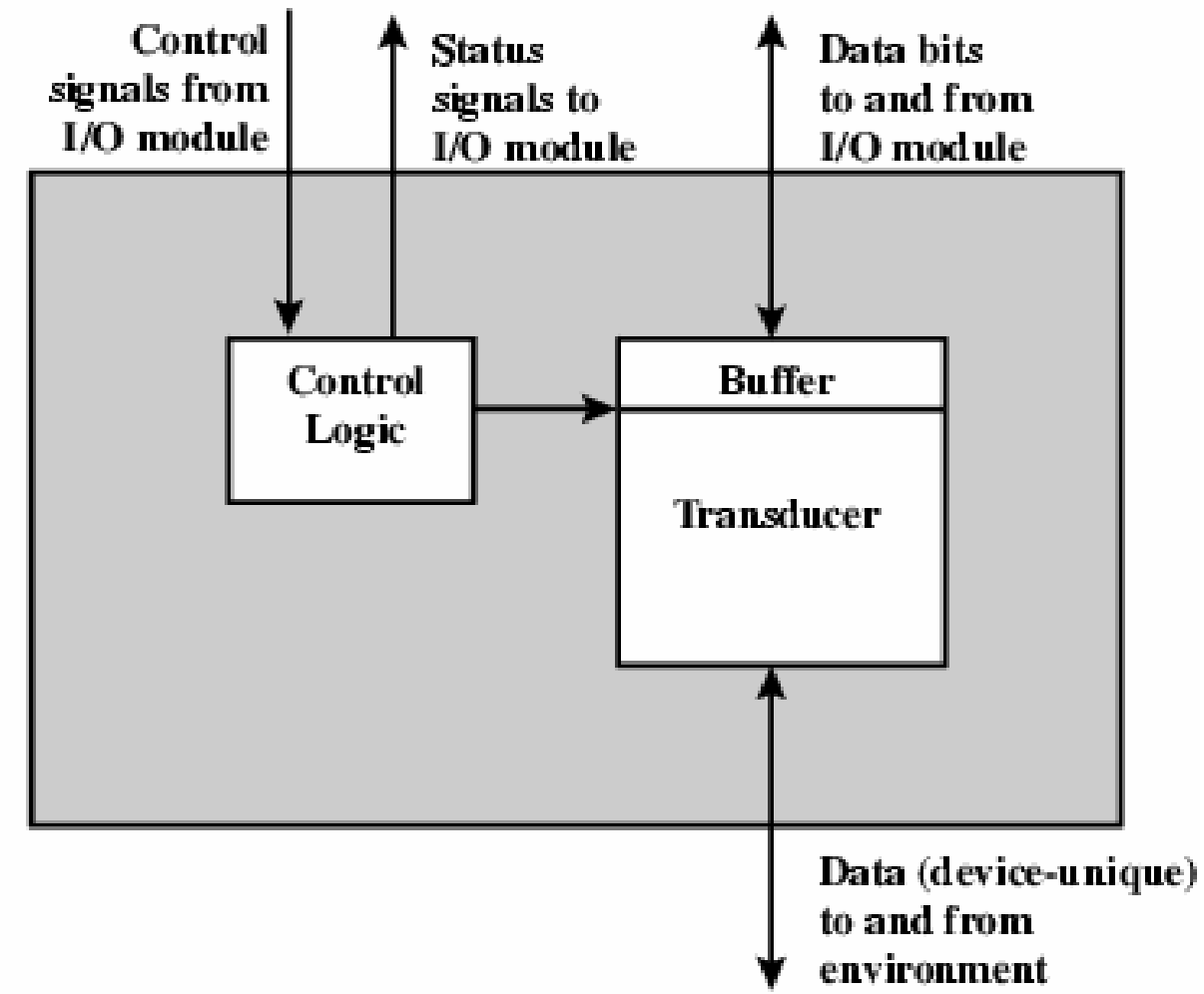


Diagrama em blocos de dispositivo externo



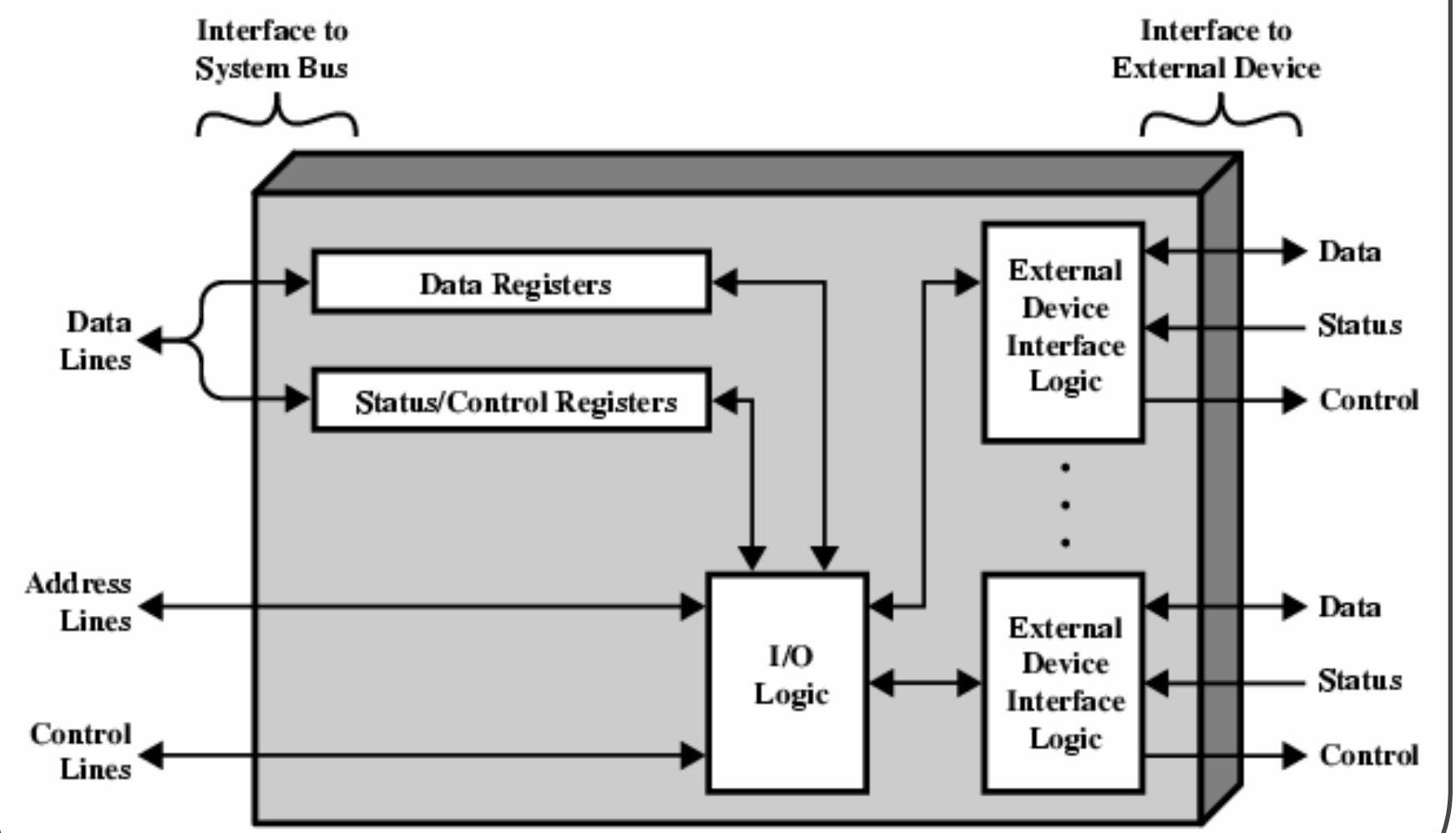
Funções do Módulo de E/S

- Controle & Temporização
- Comunicação com UCP
- Comunicação com dispositivo
- Bufferização de dados
- Detecção de erros

Operação de E/S

- UCP solicita estado do dispositivo para módulo de E/S
- Módulo de E/S retorna estado
- Caso dispositivo pronto, UCP solicita transferência de dados
- Módulo de E/S obtém dados do dispositivo
- Módulo de E/S transfere dados para UCP

Diagrama do Módulo de E/S



Decisões do Módulo de E/S

- **Esconder ou revelar propriedades do dispositivo para UCP**
- **Suportar um ou múltiplos dispositivos**
- **Controlar funções do dispositivo ou deixar para UCP**
- **Decisões do Sistema Operacional de como tratar dispositivos de E/S**
  - Unix trata tudo possível como arquivos

Comandos de E/S

- **Envio de endereço**
  - Identifica módulo (endereço do dispositivo, caso mais de um por módulo)
- **Envio de comando**
  - Controle – indica ao módulo o que fazer
    - Desloca cabeça de leitura e gravação
  - Teste – verifica estado
    - Ligou? Erro?
  - Leitura/escrita
    - O módulo transfere dados via buffer de/para dispositivo

Mapeamento de E/S

- **E/S mapeada na memória**
  - Dispositivos e memória compartilham espaço de endereçamento
  - Operações de escrita/leitura para E/S são executadas da mesma forma que para a memória
  - Não existem comando especiais de E/S
    - Todos os comandos de acesso à memória podem ser utilizados para E/S
- **E/S isolada**
  - Espaços de endereçamento separados
  - Necessita de linhas diferentes para selecionar memória e E/S
  - Comandos especiais de E/S
    - Conjunto limitado

Técnicas de Entrada e Saída

- **Programada**
- **Por interrupção**
- **Acesso Direto à Memória (ADM)**

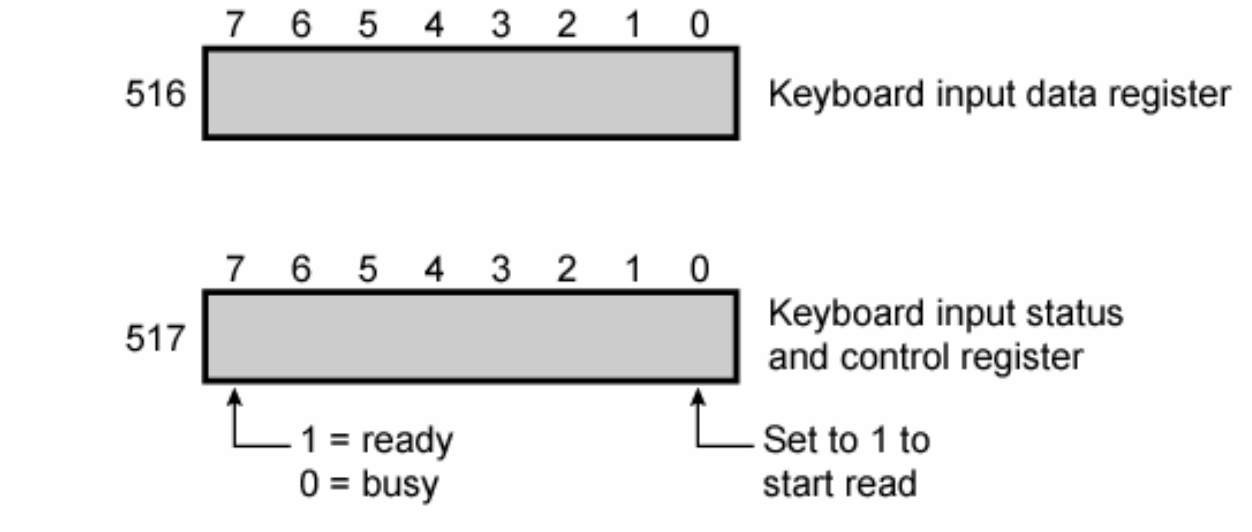
E/S Programada

- **UCP controla direto E/S**
  - Verifica estado
  - Comandos de escrita/leitura
  - Transfere dados
- **UCP espera pela finalização da operação do módulo de E/S**
- **Gasta tempo de processamento da UCP**

E/S Programada

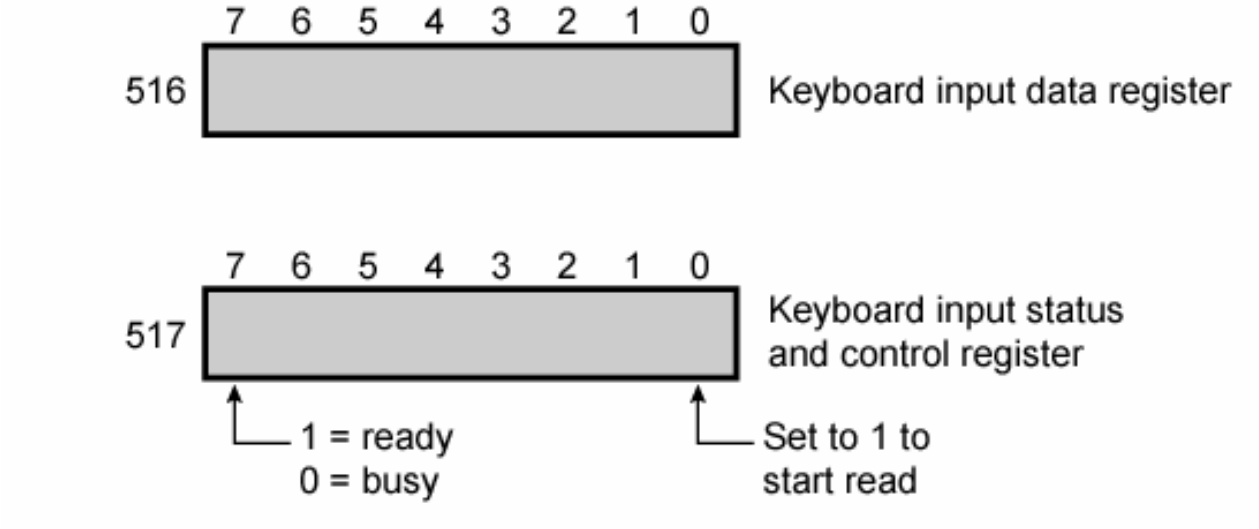
- UCP solicita operação de E/S
- Módulo de E/S realiza a operação
- Módulo de E/S seta bits de estado
- UCP verifica bits de status periodicamente
- Módulo de E/S não informa diretamente à UCP
- Módulo de E/S não interrompe a UCP
- UCP pode esperar ou voltar mais tarde

Memória Mapeada



Endereço	Instrução	Comentário
200	addi 0 1 1	Carrega registrador 1 com comando 1
201	sw 0 1 517	Envia comando para ler teclado
202	lw 0 2 517	Carrega estado do teclado em registrador 2
203	beq 2 0 -2	Fica em loop até teclado estar pronto
204	lw 0 3 516	Carrega dado do teclado em registrador 3

E/S Isolada



Endereço	Instrução	Comentário
200	addi 0 1 1	Carrega registrador 1 com comando 1
201	out 1 517	Envia comando para ler teclado
202	in 2 517	Carrega estado do teclado em registrador 2
203	beq 2 0 -2	Fica em loop até teclado estar pronto
204	in 3 516	Carrega dado do teclado em registrador 3

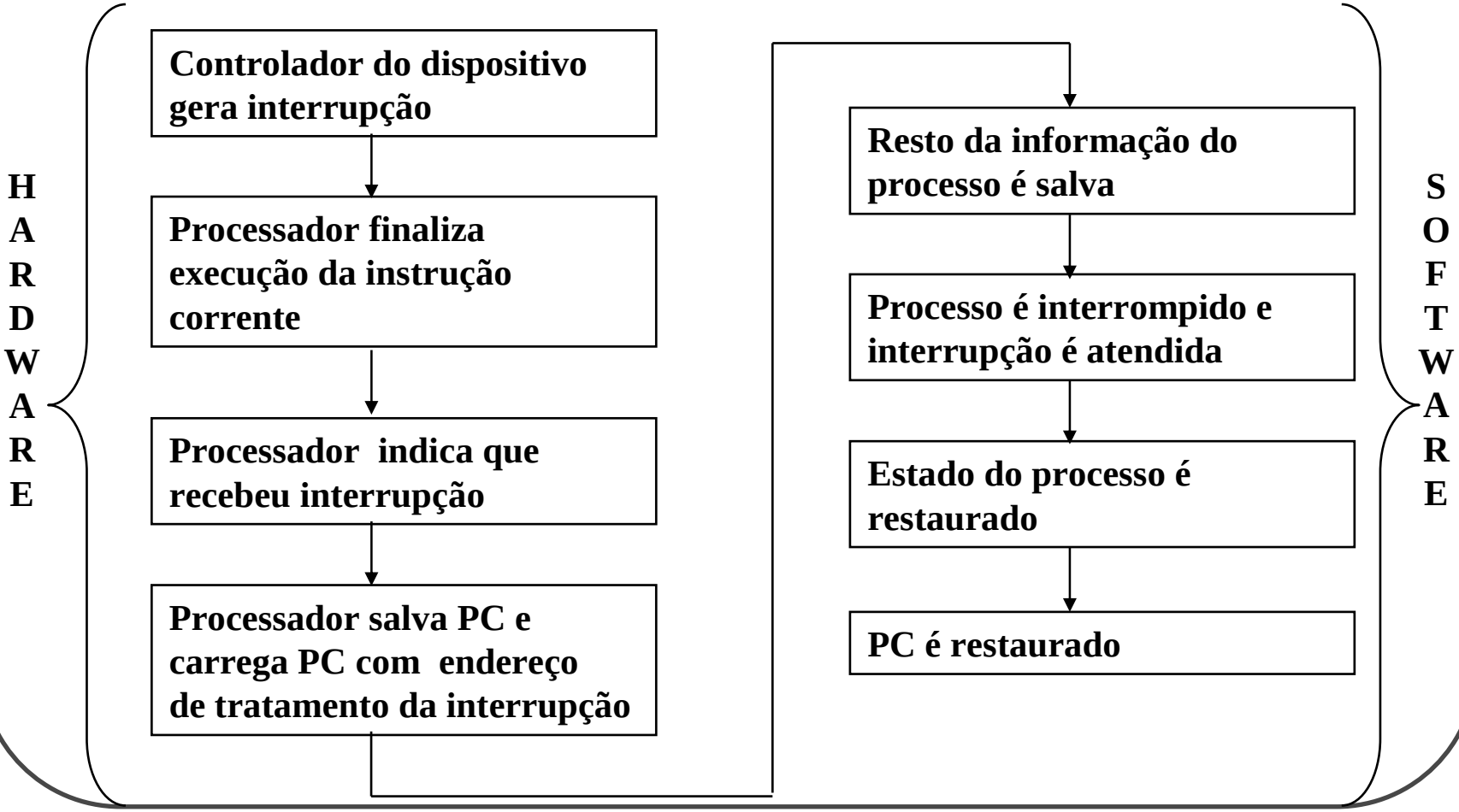
E/S Dirigida por Interrrupção

- **Libera espera de UCP**
- **UCP não precisa ficar verificando estado do dispositivo repetidamente**
- **Módulo de E/S interrompe a UCP quando estiver pronto**

Operação Básica para Realização de E/S por Interrupção

- **UCP envia comando de leitura**
- **Módulo de E/S obtém dado do periférico enquanto a UCP executa outro trabalho**
- **Módulo de E/S interrompe a UCP**
- **UCP pede dados para o módulo de E/S**
- **Módulo de E/S transfere dados para UCP**

Processamento da Interrupção



Atividades da UCP

- **Envia comando de leitura**
- **Executa outra tarefa**
- **Verifica se existe interrupção ao final de cada instrução**
- **Caso exista interrupção:**
  - Salva contexto (registradores)
  - Interrompe processo
    - Obtém dados do módulo de E/S e e os armazena

Questões de Projeto

- **Como identificar o módulo que gerou a interrupção?**
- **Como gerenciar muitas interrupções ?**
  - Qual delas atender ?

Identificação do Módulo que Gera a Interrupção (1)

- **Uma linha diferente para cada módulo**
  - Limita número de dispositivos porque número de linhas no barramento limitado
- **Identificação por software**
  - Uma única linha de interrupção
  - UCP interroga um módulo de cada vez para verificar se ele gerou a interrupção
  - Lento

Identificação do Módulo que Gera a Interrupção (2)

- **Arbitragem do barramento**
  - Módulo precisa obter o controle do barramento e depois envia sinal de interrupção
  - UCP envia sinal de reconhecimento e módulo coloca o vetor de interrupção nas linhas de dados
  - PCI & SCSI

Múltiplas Interrupções

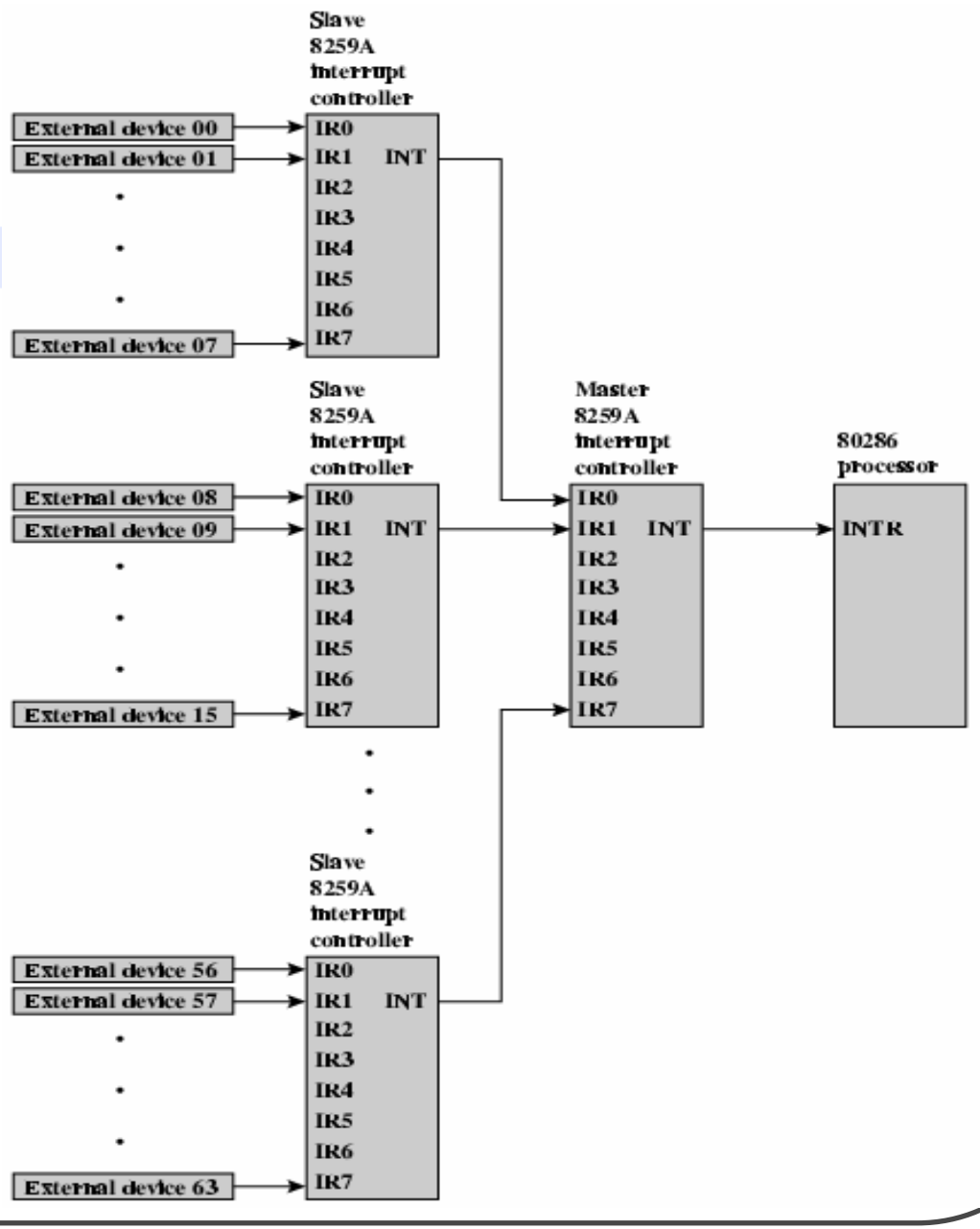
- **Com mais de uma linha de interrupção, cada linha de interrupção possui uma prioridade**
- **Linhas com prioridade maior podem interromper linhas com prioridade menor**
- **Esquema de prioridades para arbitração de barramento**

Exemplo – Barramento PC

- 80x86 possui uma linha de interrupção
- Controladora de interrupções 8259A
- 8259A possui 8 linhas de interrupção

Seqüência de Eventos

- 8259A aceita interrupções
- 8259A determina prioridade
- 8259A sinaliza interrupção para 8086 (levanta linha INTR)
- UCP reconhece interrupção
- 8259A coloca vetor correto no barramento de dados
- UCP processa interrupção


Slave
8259A
Interrupt
controller
External device 00
External device 01
External device 07
IR0
IR1
INT
IR2
IR3
IR4
IR5
IR6
IR7
Slave
8259A
Interrupt
controller
External device 08
External device 09
External device 15
Master
8259A
Interrupt
controller
80286
processor
INTR
Slave
8259A
Interrupt
controller
External device 56
External device 57
External device 63

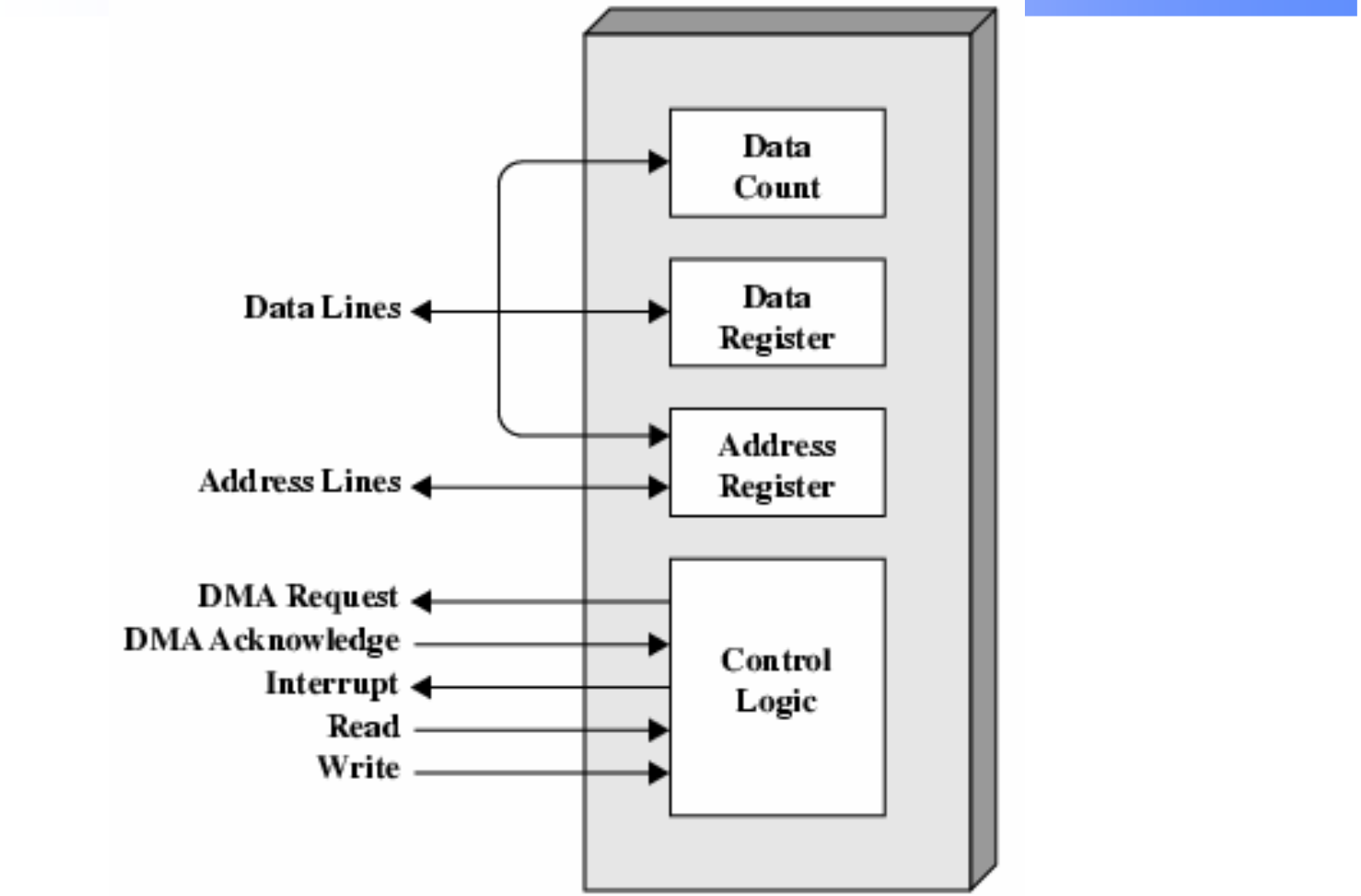
Acesso Direto à Memória

- **E/S programada e por interrupção requerem intervenção ativa da UCP**
  - Taxa de transferência é limitada pela capacidade de atendimento da UCP
  - UCP fica ocupada gerenciando a transferência de dados
- **ADM pode ser uma técnica mais eficiente**

Função do ADM

- **Módulo adicional de hardware no barramento**
- **Controlador de ADM “imita” a UCP para realizar operações de E/S**

Diagrama de um Módulo para ADM



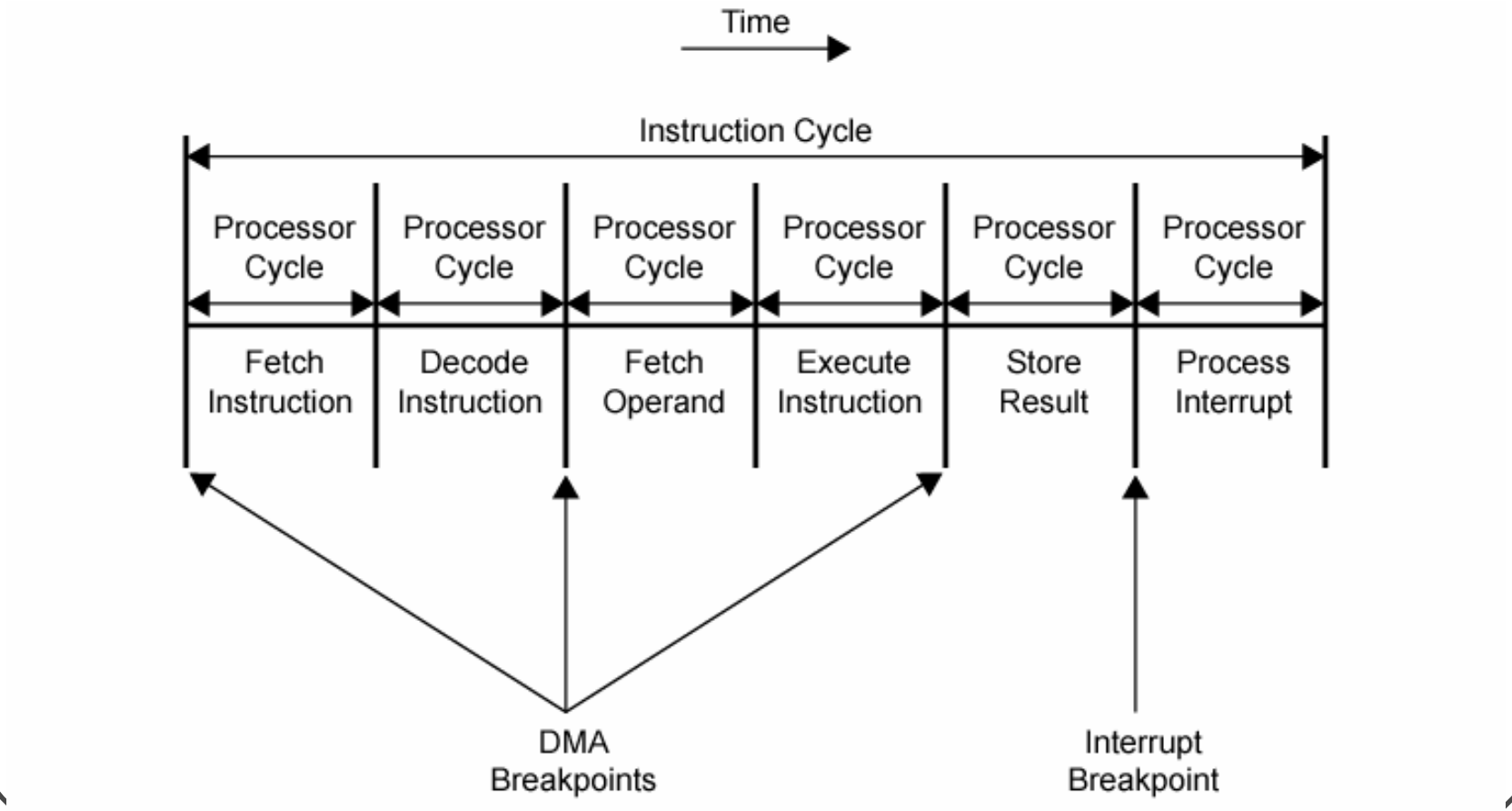
Operação do ADM

- **UCP indica ao controlador de ADM:**
  - Operação: Escrita/Leitura
  - Endereço do dispositivo
  - Endereço inicial do bloco de memória para dados
  - Quantidade de dados a serem transferidos
- **UCP executa outra tarefa**
- **Controlador de ADM processa transferência**
- **Controlador de ADM envia interrupção quando operação é finalizada**

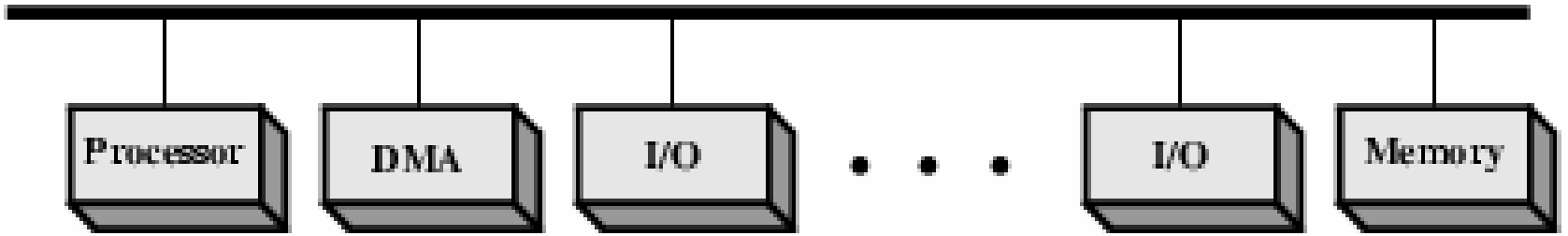
Roubo de Ciclo pelo ADM

- **Controlador de ADM toma conta do barramento por um ciclo**
- **Transfere uma palavra de dados**
- **Diferente de interrupção**
  - UCP não realiza troca de contexto
- **UCP é suspensa imediatamente antes de acessar o barramento**
  - Antes da busca da instrução e do operando, antes de armazenar dados na memória
- **Diminui velocidade de processamento da UCP mas evita que a UCP tenha que realizar a transferência**

Pontos de Suspensão da UCP para ADM e Interrupção

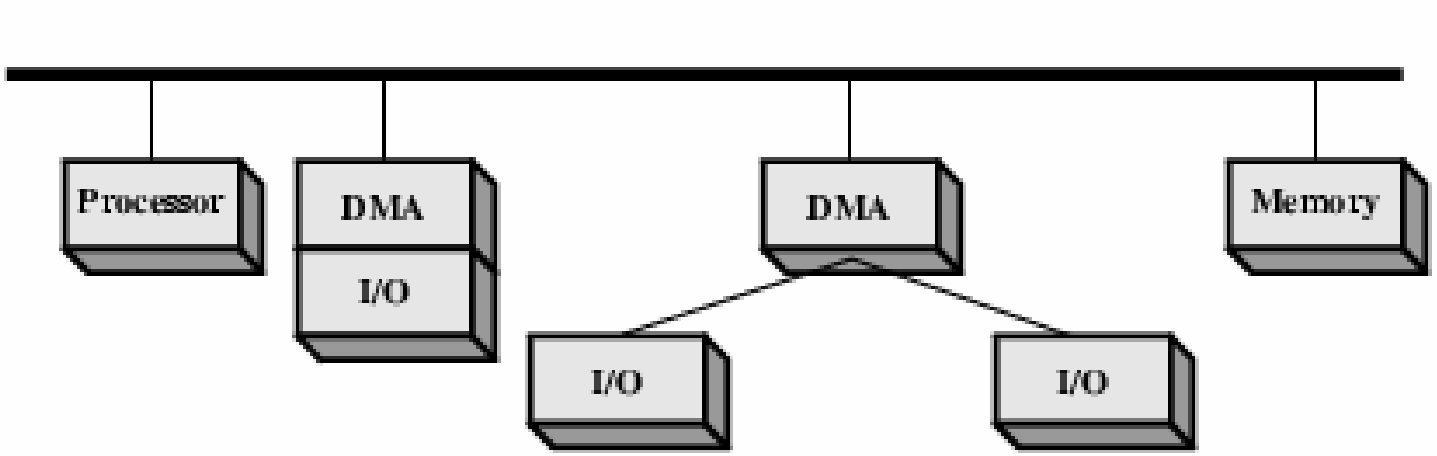


Configurações de ADM (1)



- **Barramento único, Controlador de ADM separado**
- **Cada transferência utiliza duas vezes o barramento**
  - E/S para ADM e do ADM para memória
- **UCP é suspensa 2 vezes**

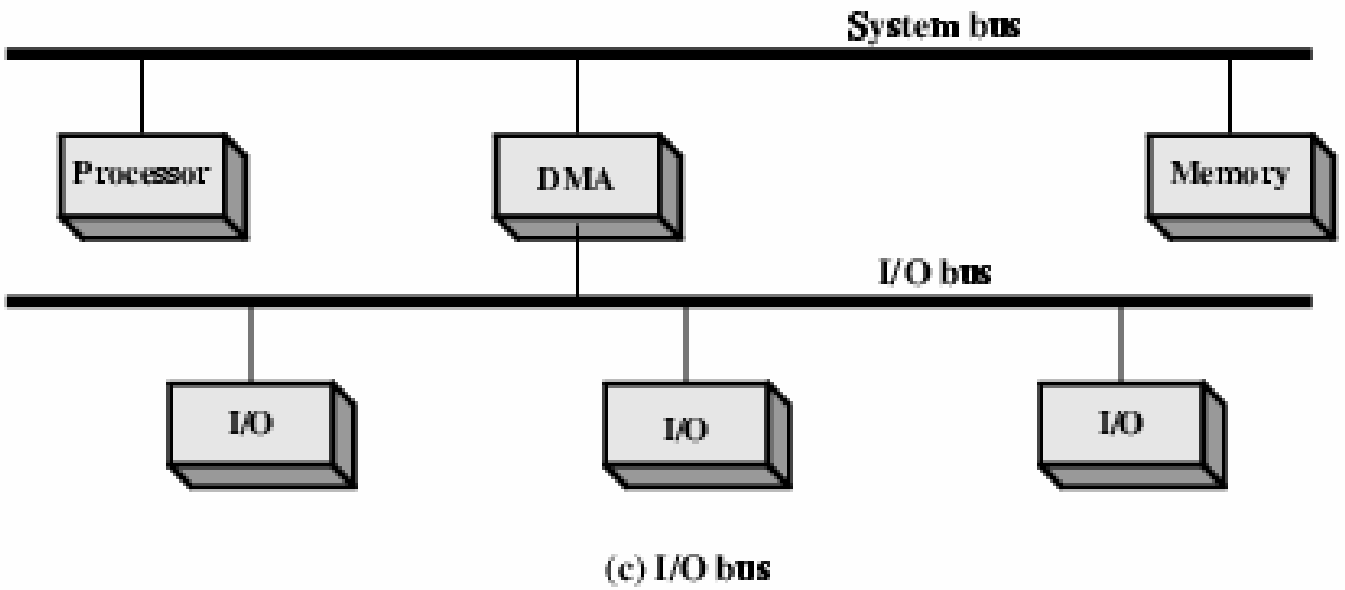
Configurações de ADM (2)



(b) Single-bus, Integrated DMA-I/O

- **Barramento único, Controlador de ADM integrado**
- **Controlador pode suportar mais de um dispositivo**
- **Cada transferência utiliza o barramento uma única vez**
  - ADM para memória
- **UCP é suspensa uma única vez**

Configurações de ADM (3)



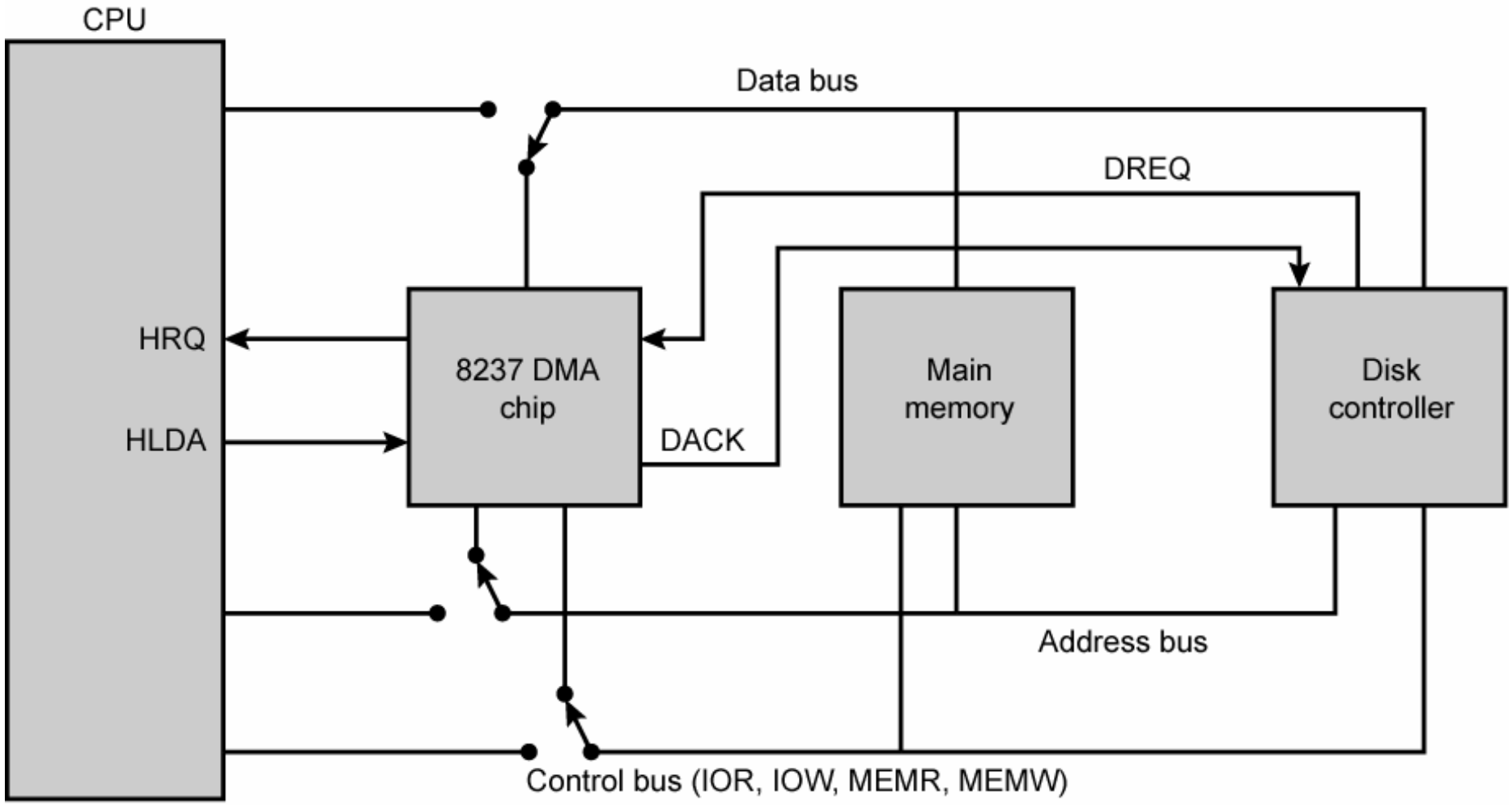
(c) I/O bus

- **Barramento de E/S separado**
- **Barramento suporta todos dispositivos que podem realizar ADM**
- **Cada transferência utiliza o barramento uma vez**
  - ADM para memória
- **UCP é suspensa uma vez**

Controlador ADM Intel 8237A

- **Interfaces para família 80x86 e DRAM**
- **Quando módulo de ADM necessita do barramento envia o sinal HRQ para processador**
- **UCP responde com HLDA (hold acknowledge)**
  - Módulo de ADM pode usar o barramento
- **Exemplo de transferência entre memória e disco**
  1. Dispositivo pede serviço ao ADM colocando a linha DREQ (DMA request) em 1
  2. ADM coloca HRQ (hold request) em 1
  3. UCP finaliza o ciclo de uso do barramento e coloca a linha HDLA (hold acknowledge) em 1
  4. ADM ativa linha DACK (DMA acknowledge), indicando que dispositivo pode começar transferência
  5. ADM inicia transf. colocando o endereço do primeiro byte no barramento de endereços e ativando a linha MEMR; depois ativa a linha IOW para escrever no periférico. ADM decrementa contador e incrementa apontador de endereço
  6. ADM coloca linha HRQ em 0, retornando o controle do barramento para UCP
  7. Volta ao passo 2 até contador chegar a 0

Uso do Barramento pelo 8237



DACK = DMA acknowledge
DREQ = DMA request
HLDA = HOLD acknowledge
HRQ = HOLD request