

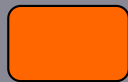
# **William Stallings**

## **Arquitetura e Organização de Computadores**

### **8ª Edição**

## **Capítulo 7**

### **Entrada/saída**



Os textos nestas caixas  
foram adicionados pelo  
Prof. Joubert



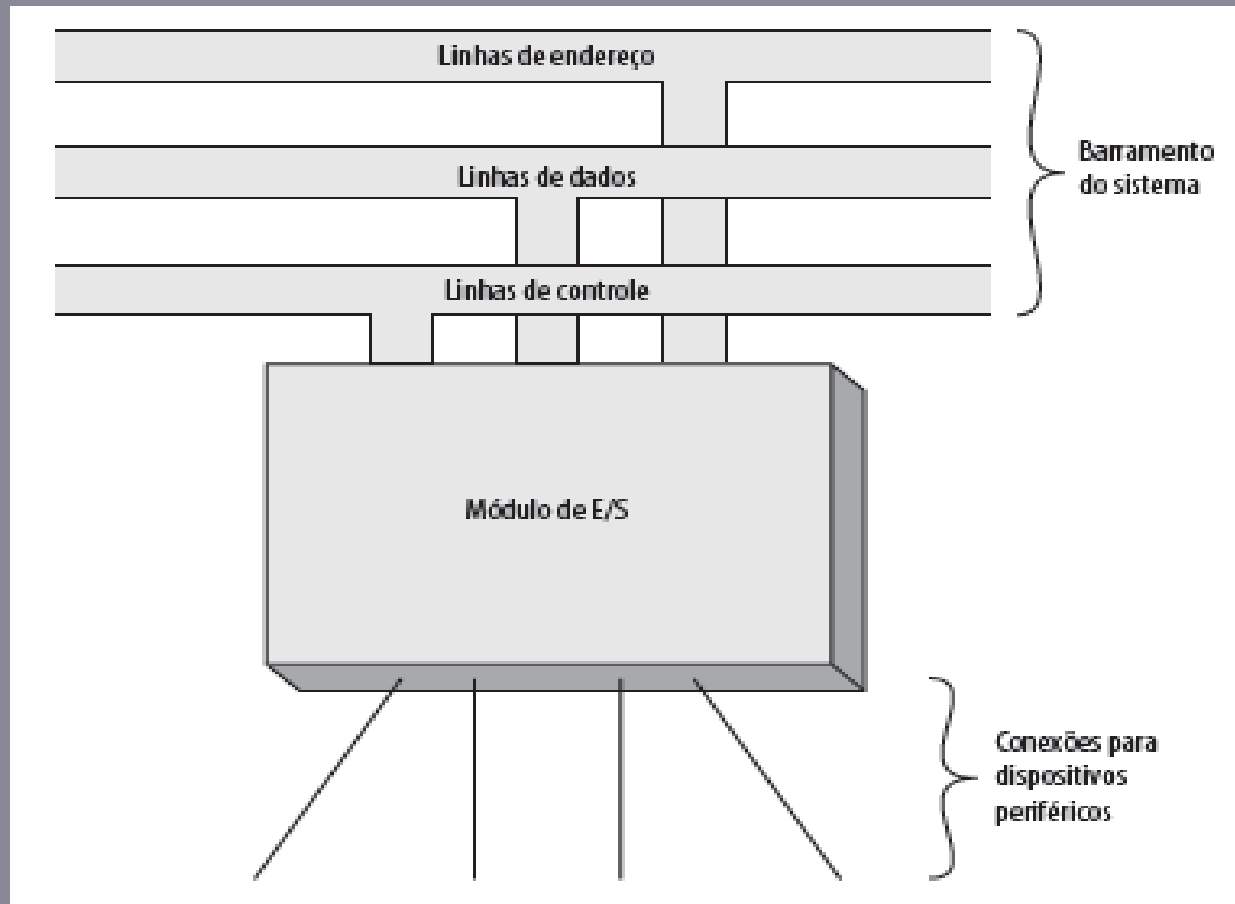
## Problemas de entrada/saída

- 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.

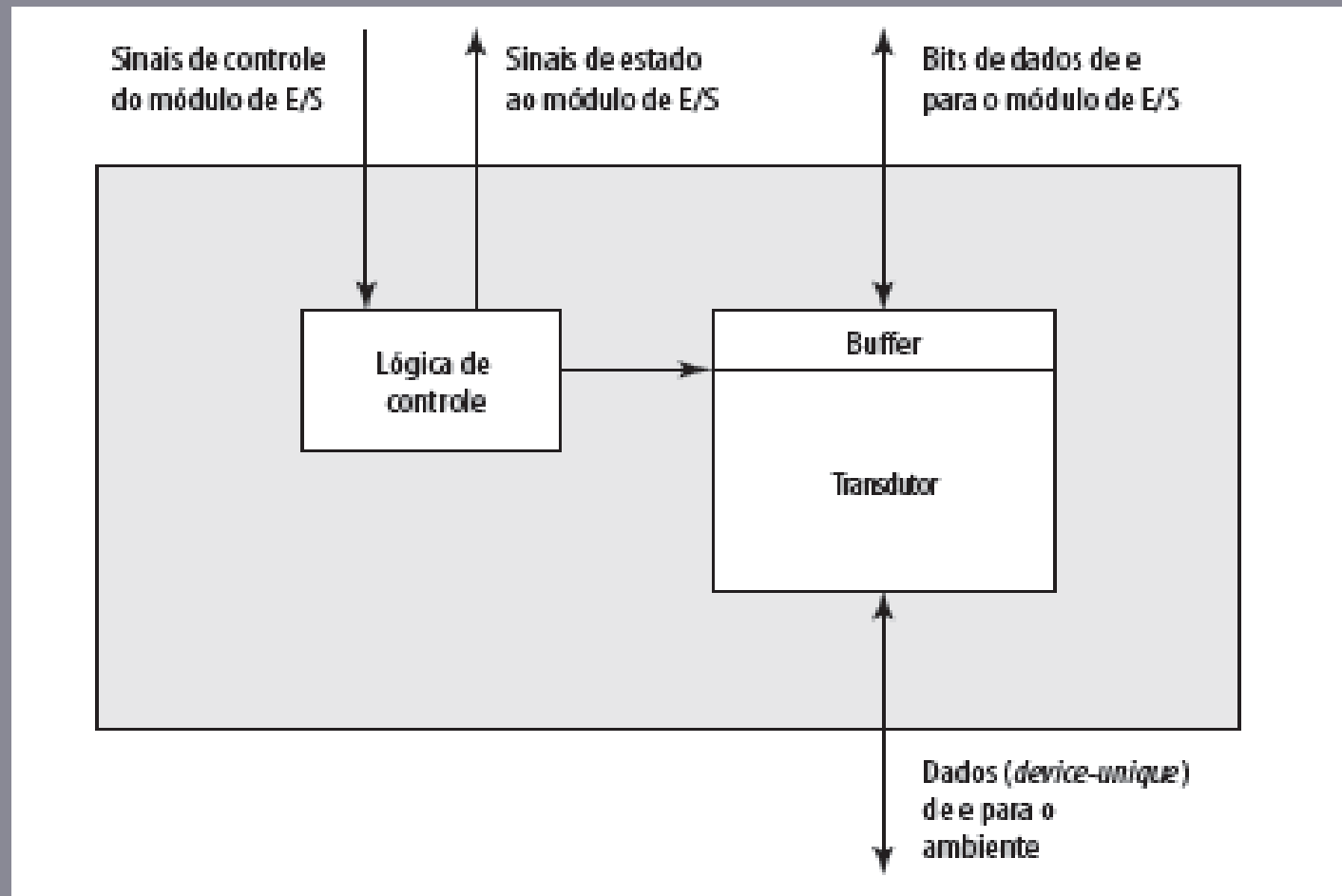
## Modelo genérico de módulo de E/S



## 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

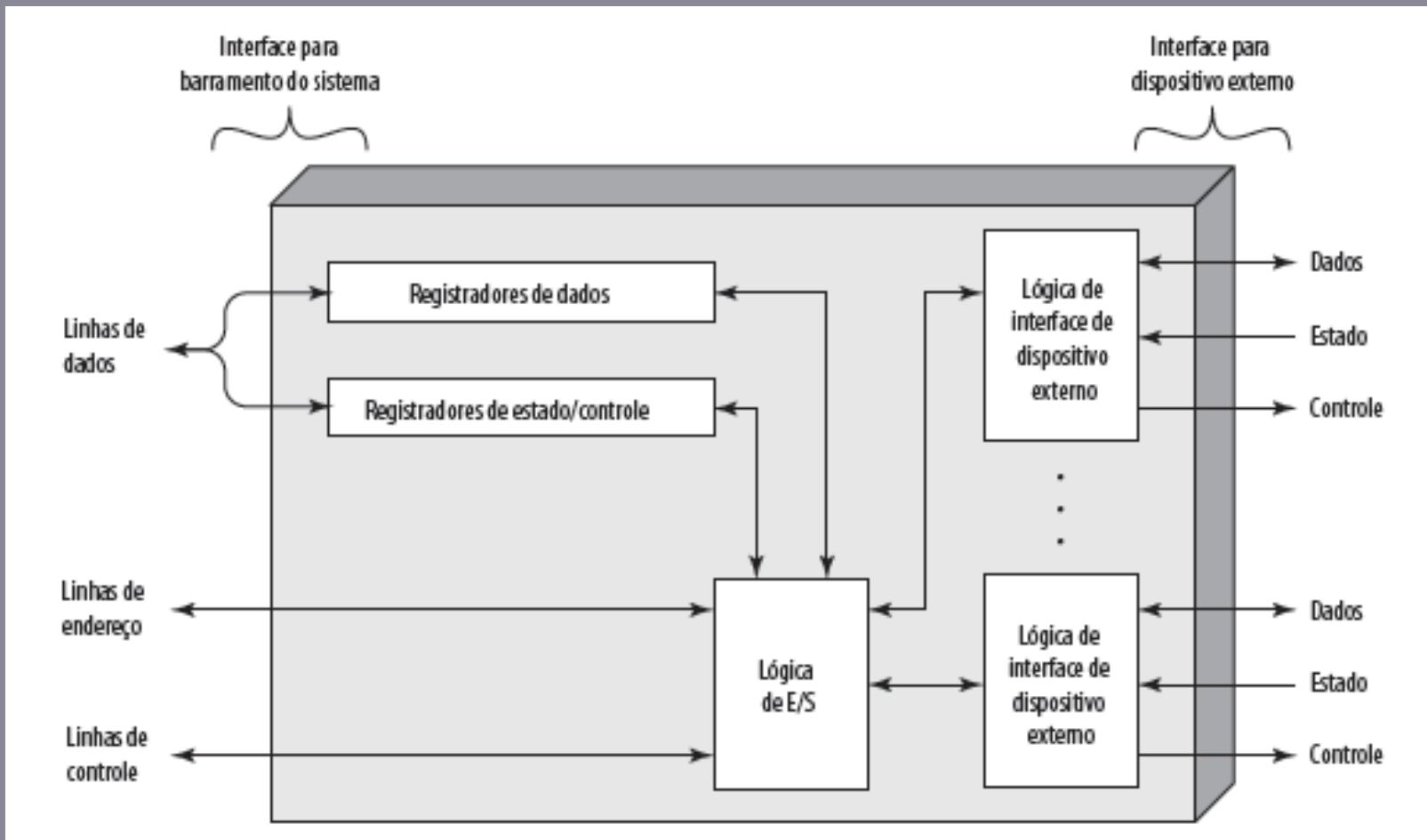
- Controle e temporização.
- Comunicação com CPU.
- Comunicação com dispositivo.
- Buffering de dados.
- Detecção de erro.

## Etapas da E/S

- CPU verifica estado do dispositivo 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

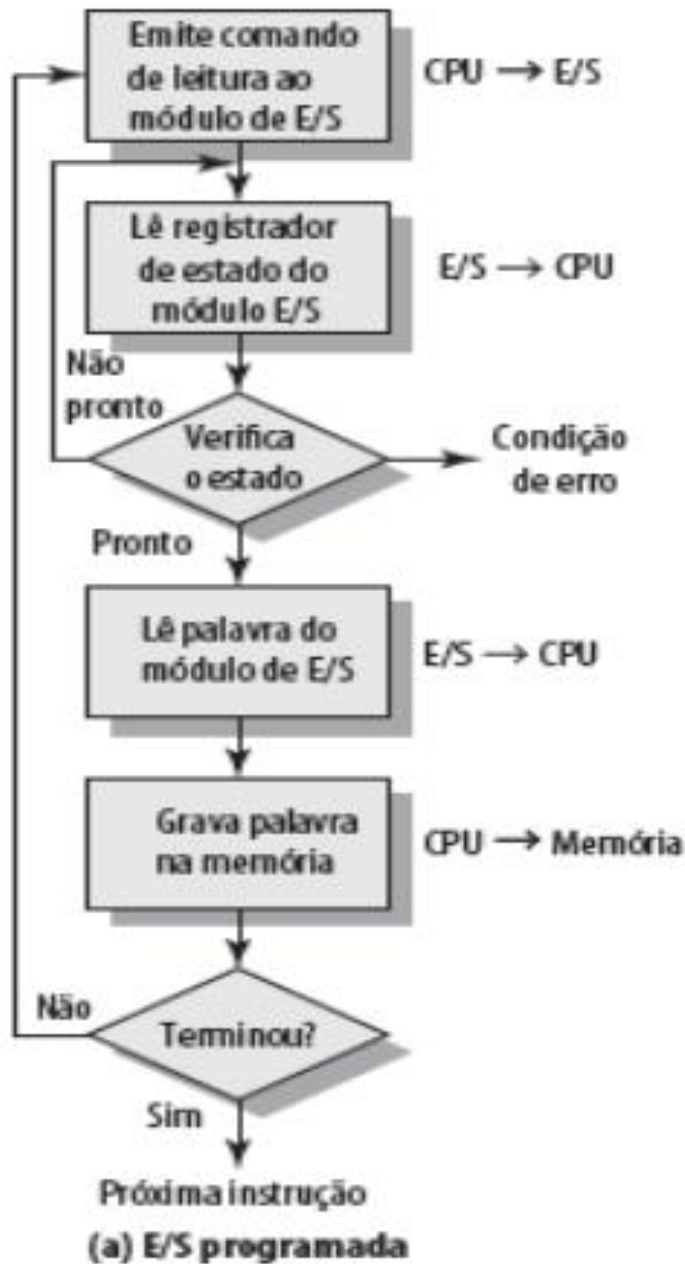


## Decisões do módulo de E/S

- 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.
- Controlada por interface.
- Acesso direto à memória (DMA).

**ARQUITETURA E ORGANIZAÇÃO  
DE COMPUTADORES****E/S programada**

## 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 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.

## 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

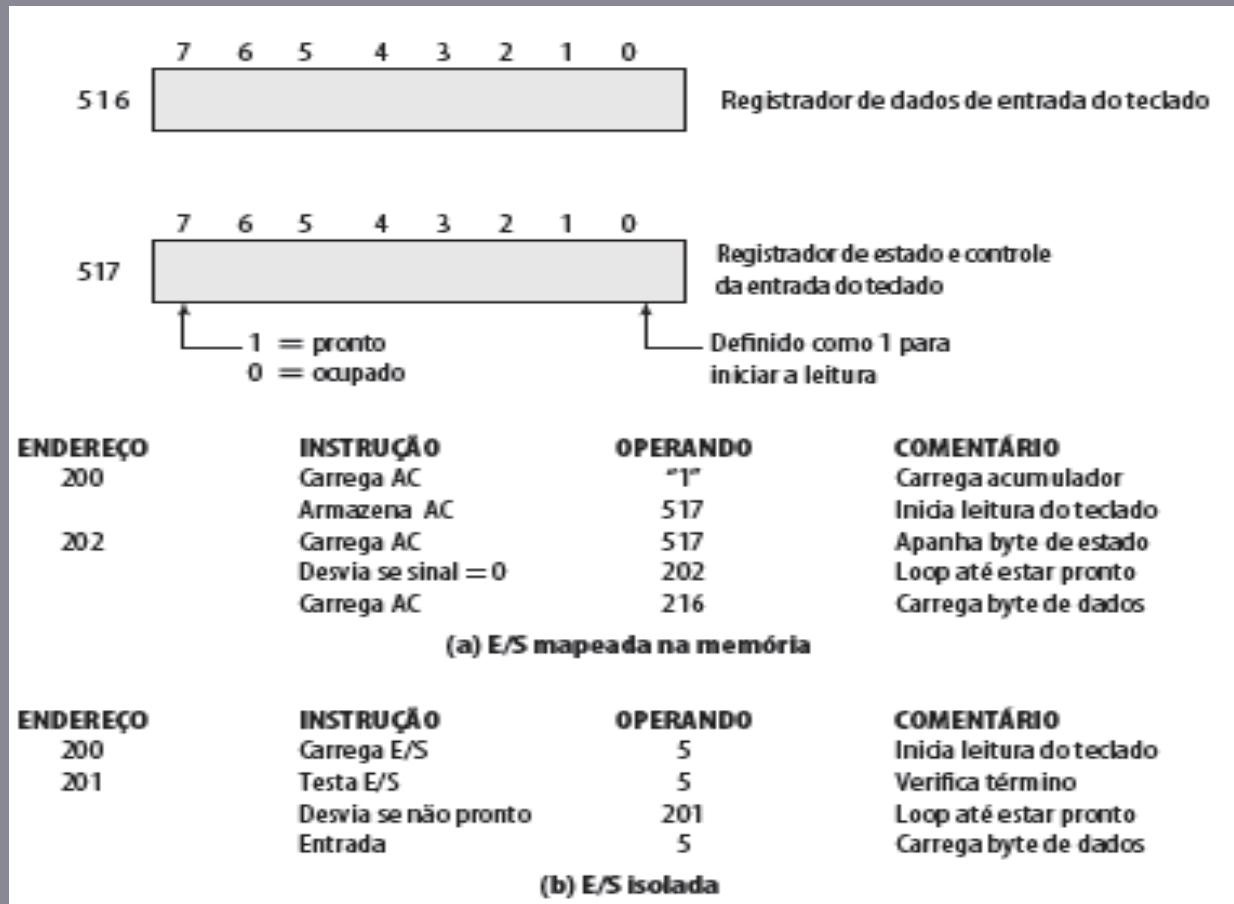
- Sob E/S programada a transferência de dados é muito semelhante ao acesso à memória (ponto de vista da CPU).
- Cada dispositivo recebe identificador exclusivo.
- Comandos da CPU contêm identificador (endereço).

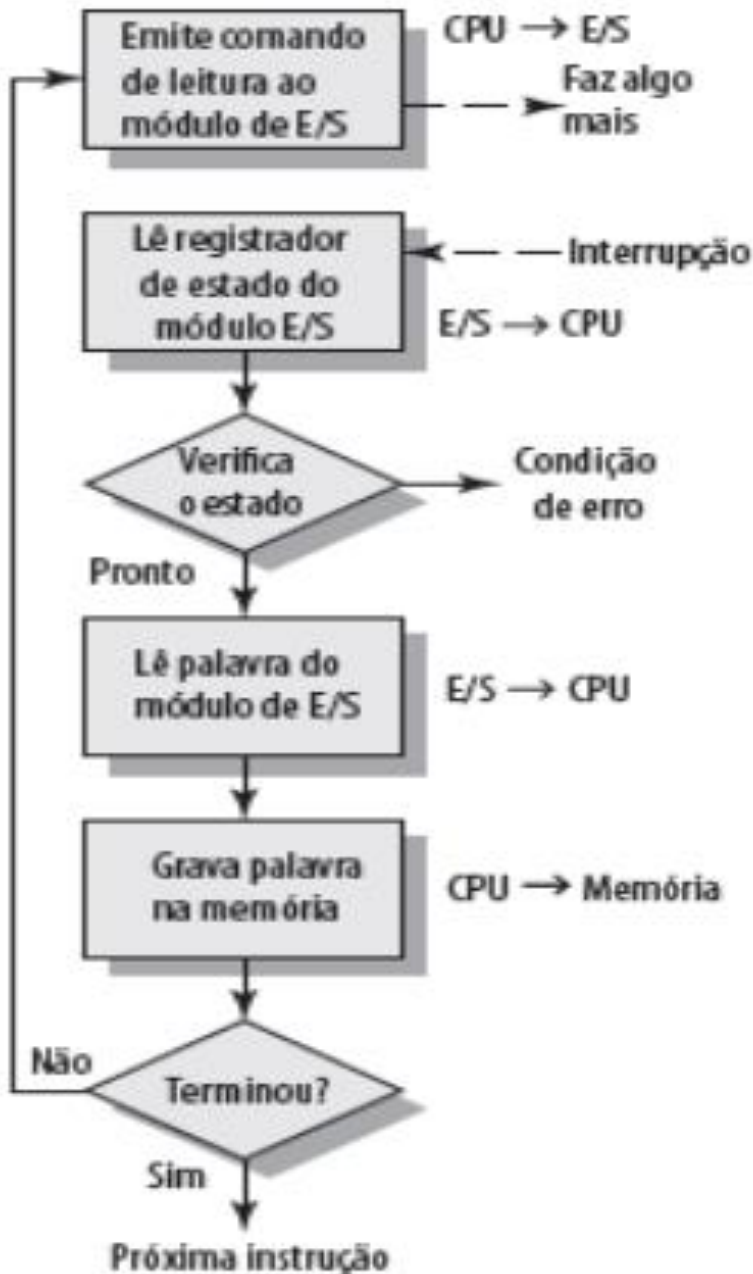


## Mapeamento de E/S

- E/S mapeada na memória:
  - 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:
  - Espaços de endereços separados.
  - Precisa de linhas de seleção de E/S ou memória.
  - Comandos especiais para E/S.
    - Conjunto limitado.

## E/S mapeada na memória e independente



ARQUITETURA E ORGANIZAÇÃO  
DE COMPUTADORES**E/S controlada por  
interrupção**

(b) E/S controlada por interrupção

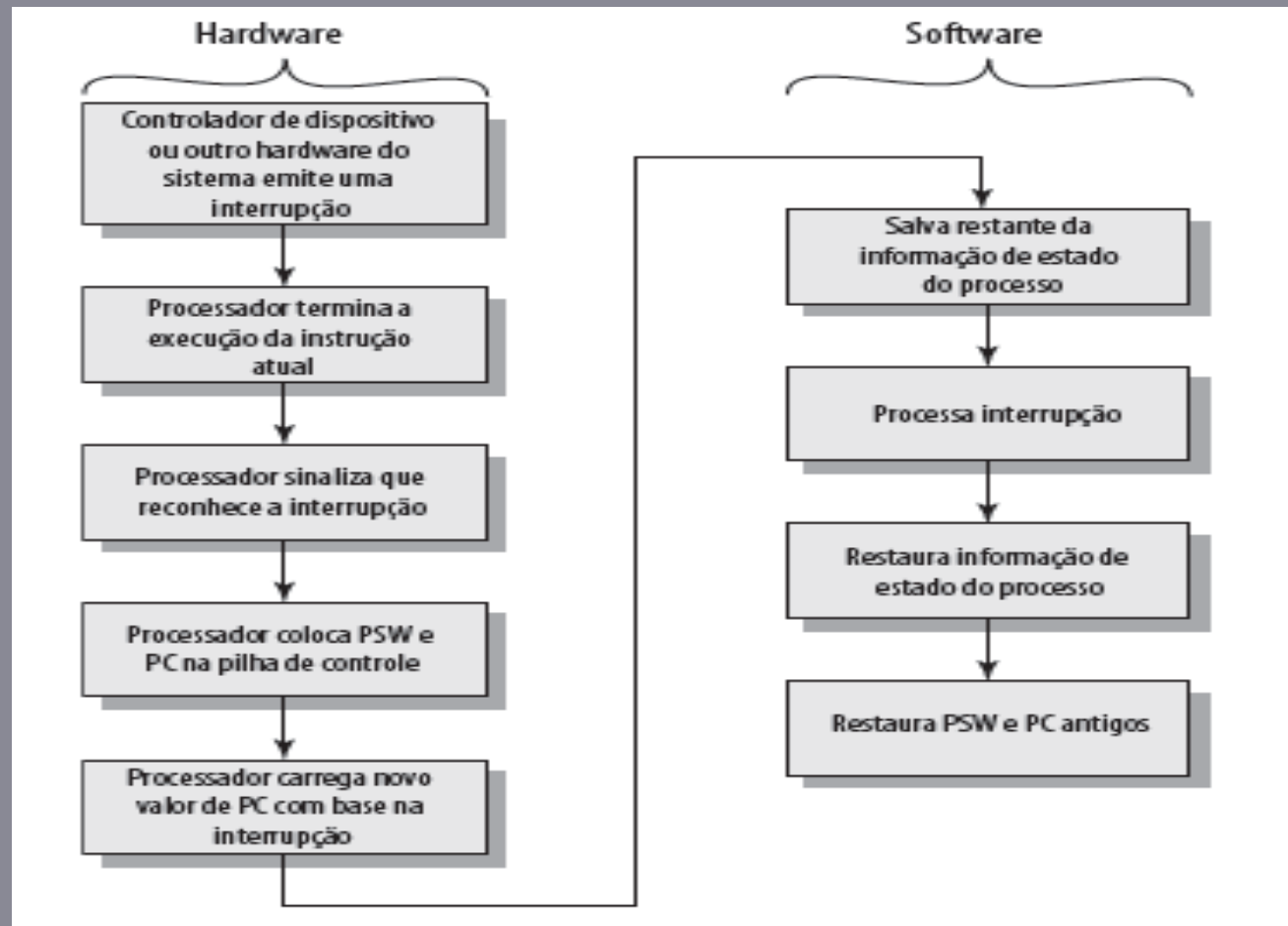
## E/S controlada por interrupção

- 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.

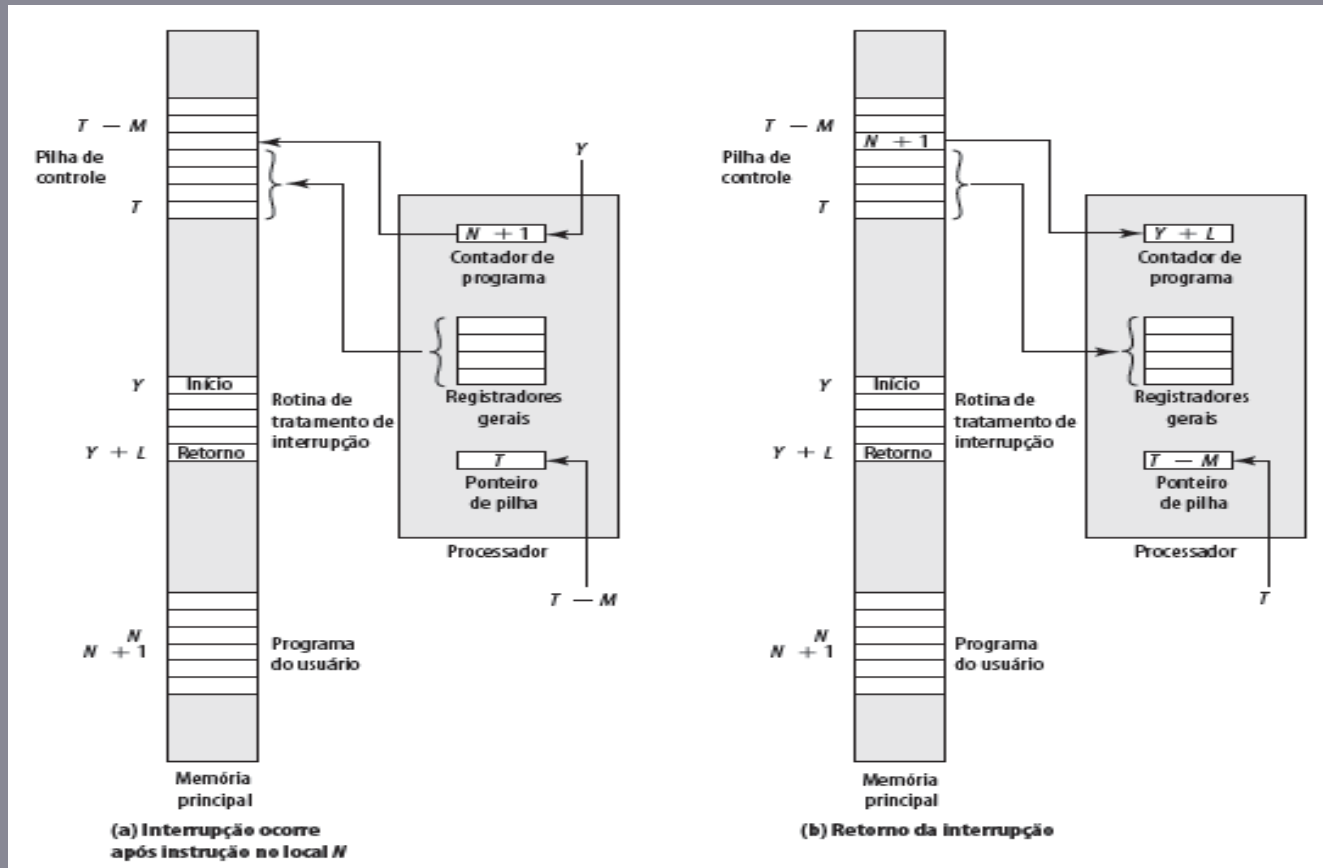
## Processamento de interrupção simples



## 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

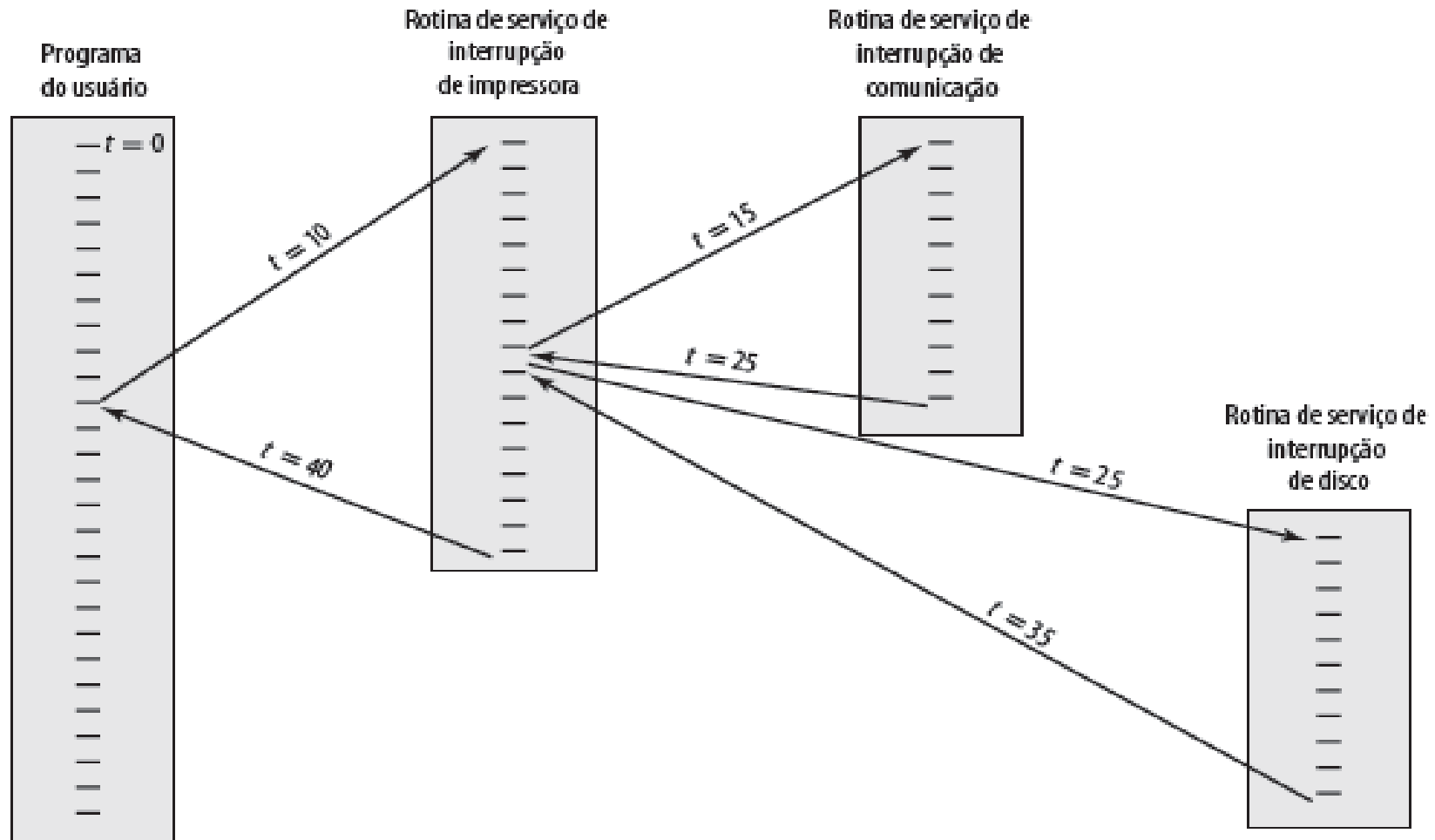
- Linha diferente para cada módulo:
  - PC.
  - Limita número de dispositivos.
- Verificação por software:
  - CPU verifica cada módulo por vez.
  - Lento.

- 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.

## 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.

# Múltiplas interrupções



## Acesso direto à memória



## Acesso direto à memória

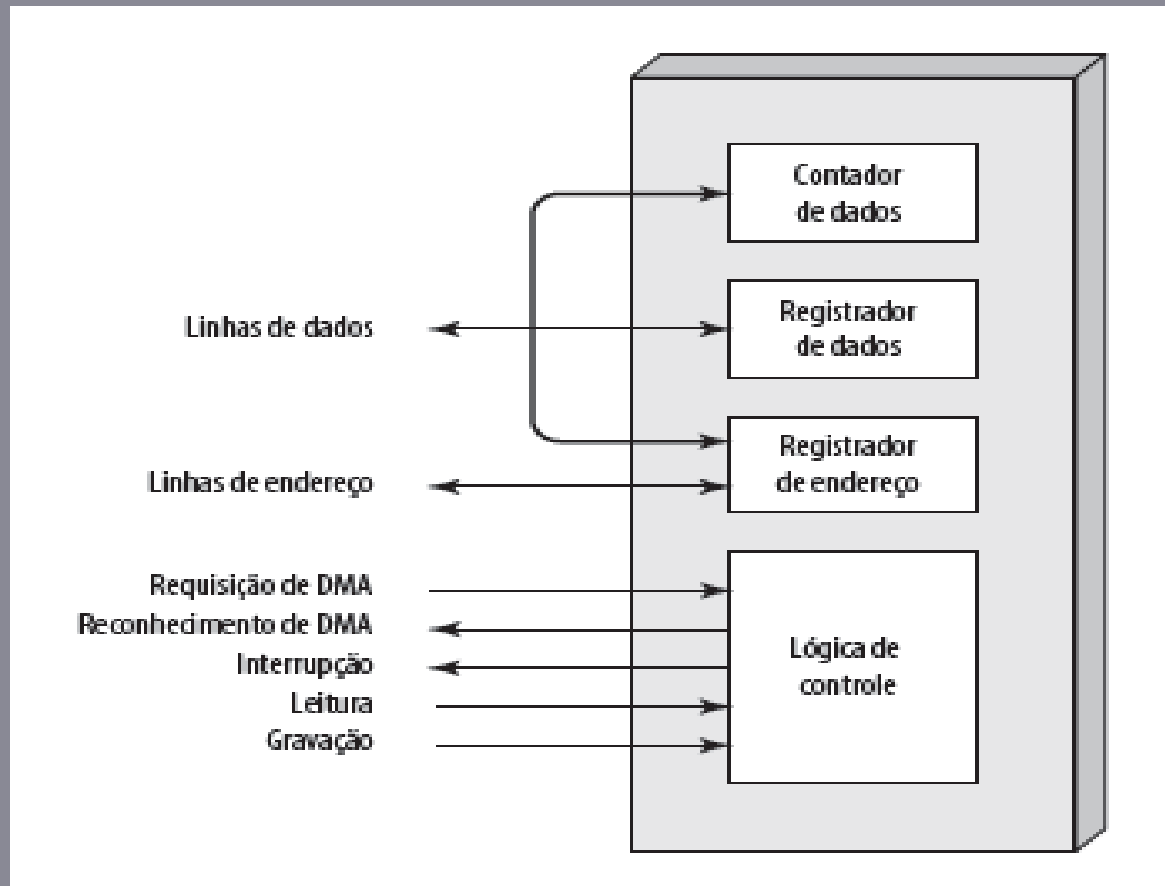
- E/S controlada por interrupção e programada exige intervenção ativa da CPU.
  - Taxa de transferência é limitada.
  - CPU fica amarrada.
- DMA é a resposta.

## 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



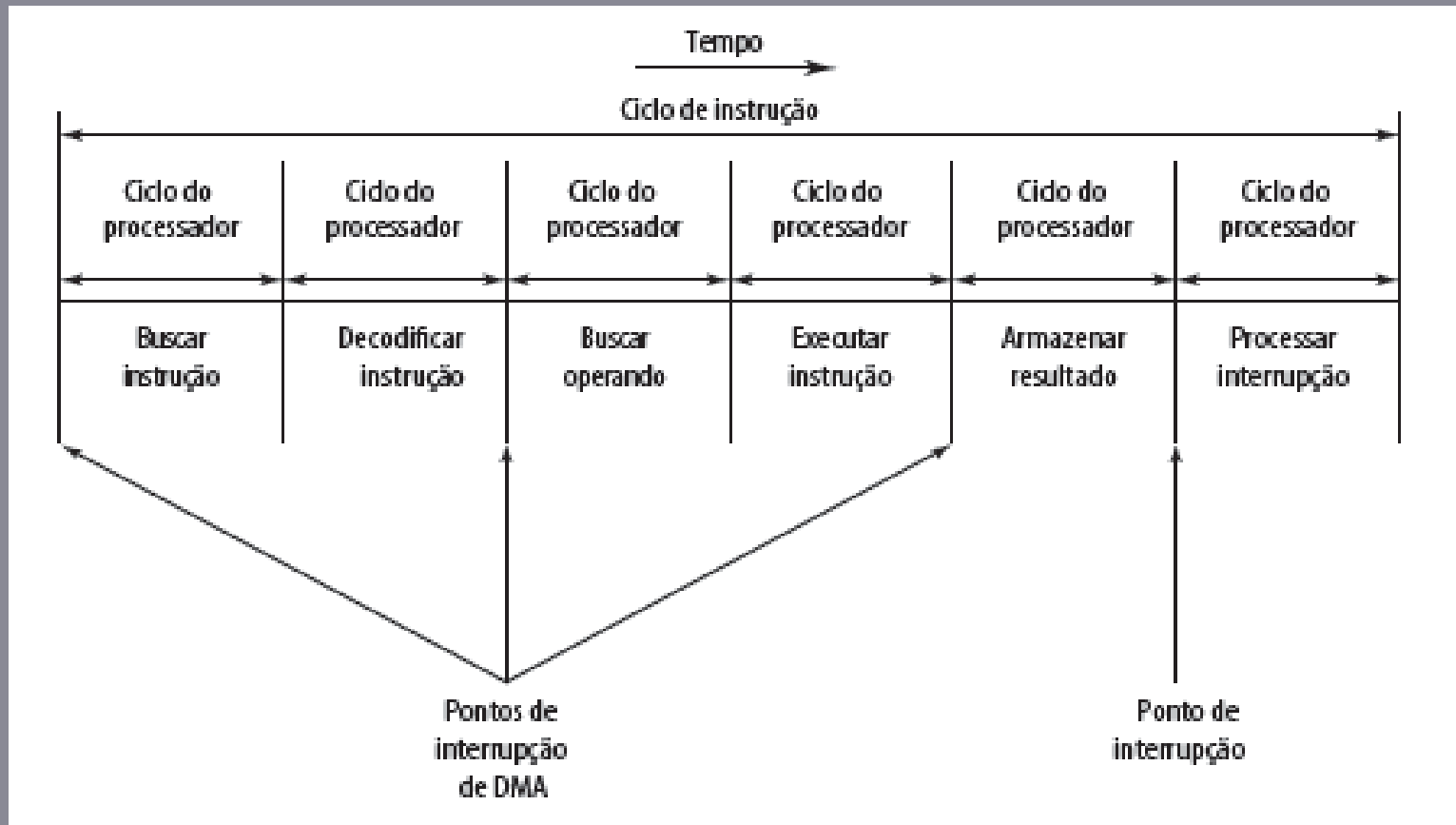
## Operação do DMA

- 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.

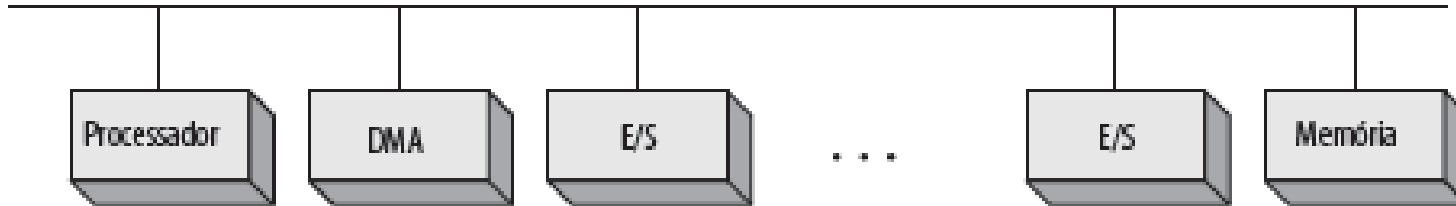
## Transferência de DMA – Roubo de ciclo

- 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

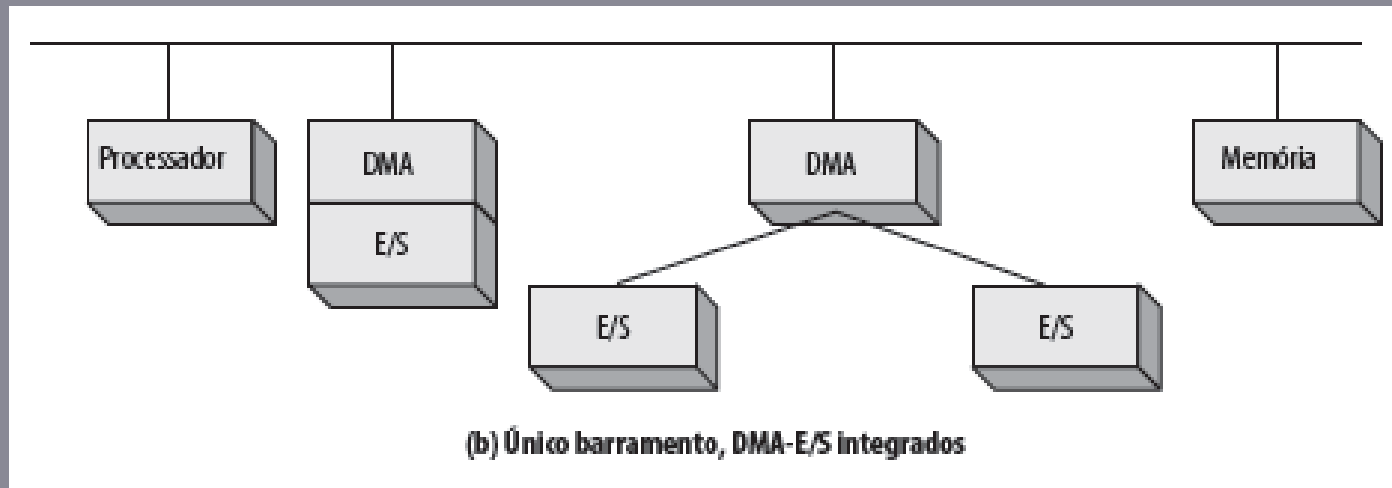


## Configurações de DMA

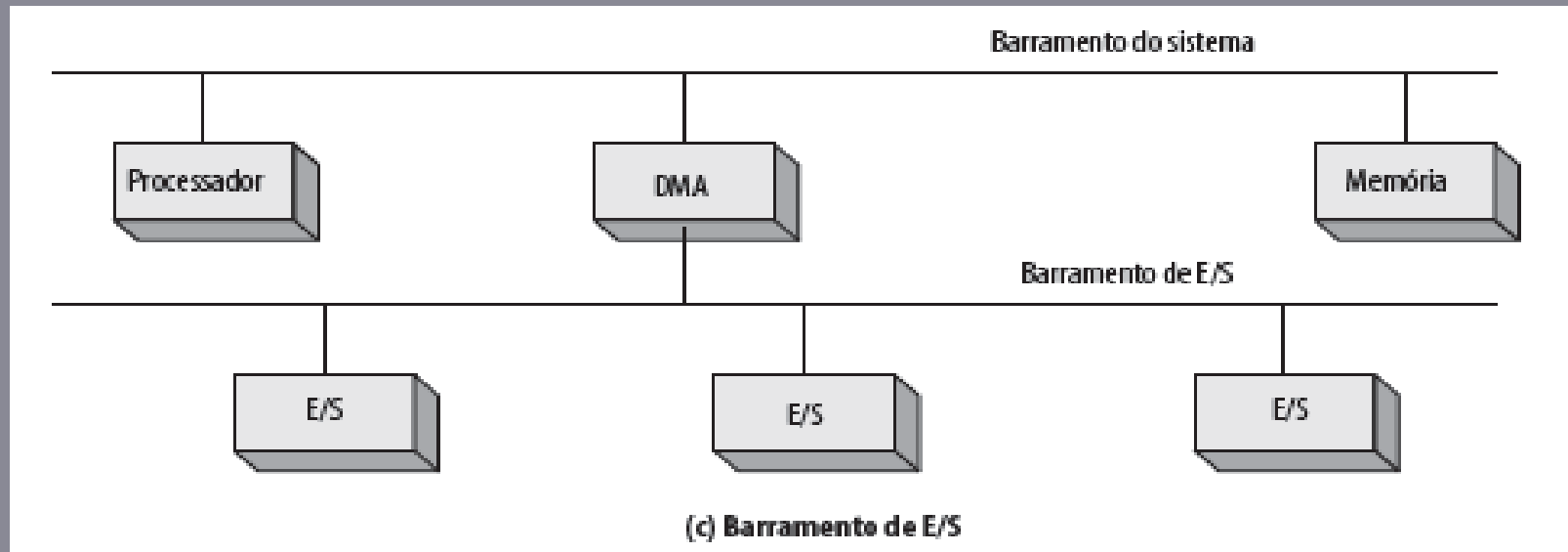


(a) Único barramento, DMA separado

- 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.

**ARQUITETURA E ORGANIZAÇÃO  
DE COMPUTADORES**

- 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.

**ARQUITETURA E ORGANIZAÇÃO  
DE COMPUTADORES**

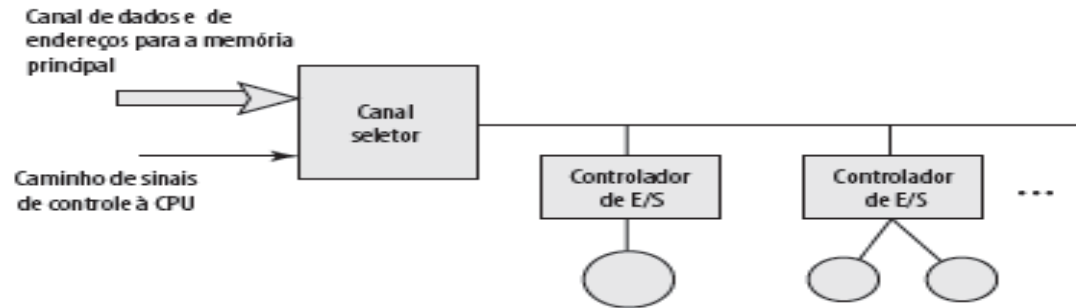
- Barramento de E/S separado.
- Barramento aceita todos dispositivos habilitados para DMA.
- Cada transferência usa barramento uma vez.
  - DMA para memória.
- CPU é suspensa uma vez.

## Canais de E/S

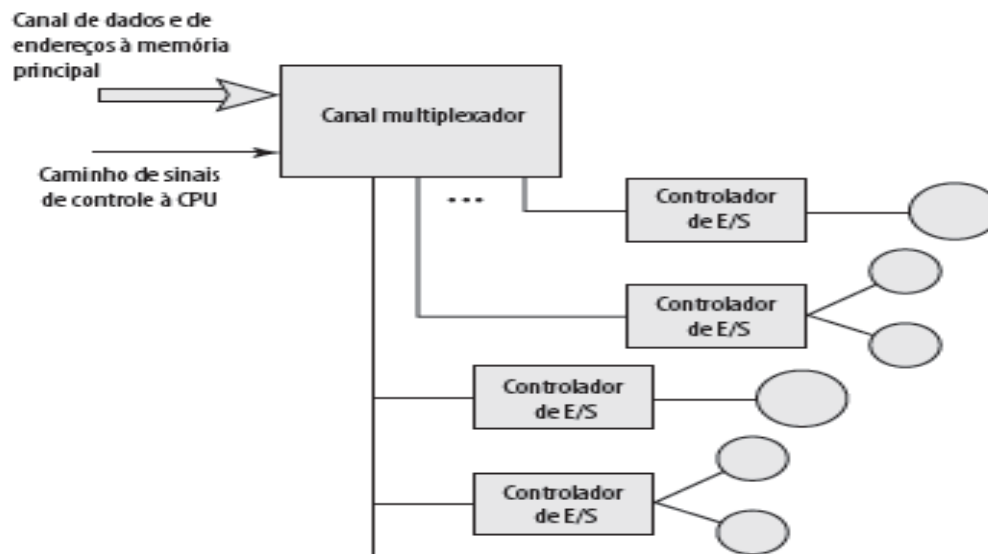
- Dispositivos de E/S se tornam mais sofisticados.
- P.e., placas gráficas 3D.
- CPU instrui controlador de E/S a realizar transferência.
- Controlador de E/S realiza transferência inteira.
- Melhora velocidade.
  - Retira carga da CPU.
  - Processador dedicado é mais rápido.



## Arquitetura do canal de E/S



(a) Seleção



(b) Multiplexador

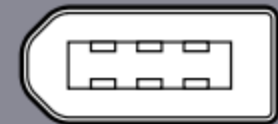
## Interface

- Conexão de dispositivos.
- Bit de fio?
- Processador dedicado/memória/barramentos?
- P.e., FireWire, InfiniBand.

## IEEE 1394 FireWire

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

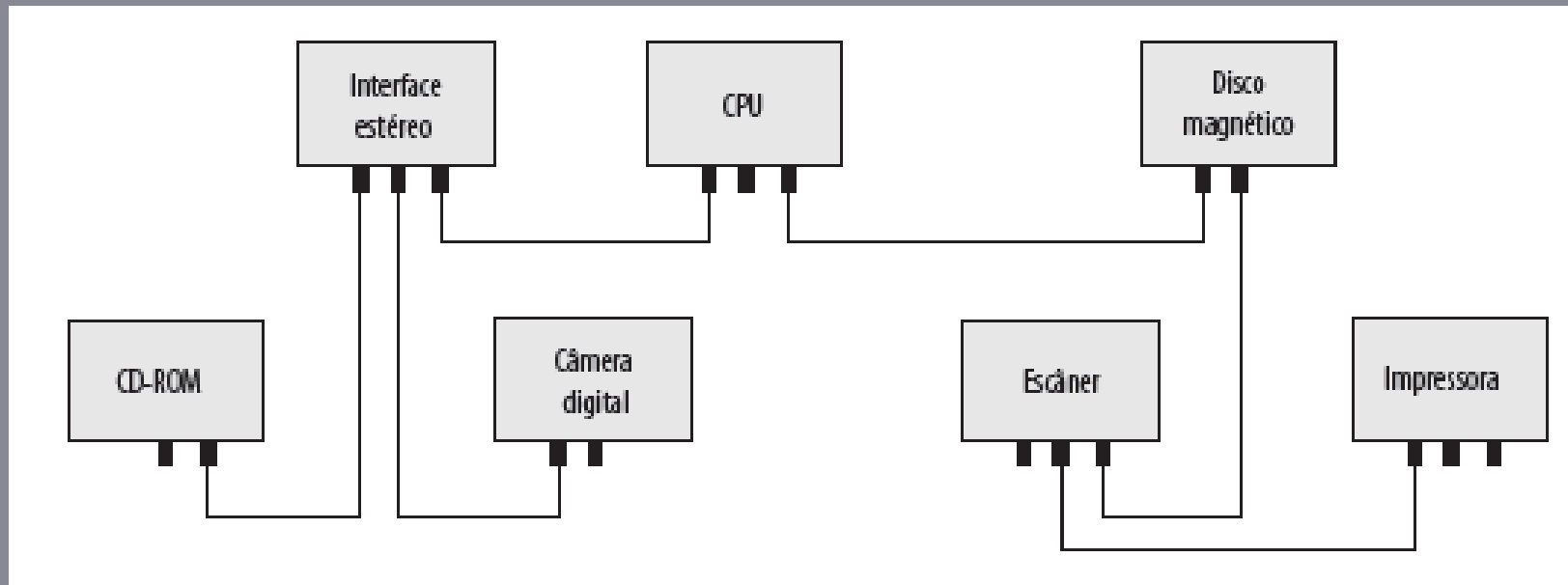
## ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES



## 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.

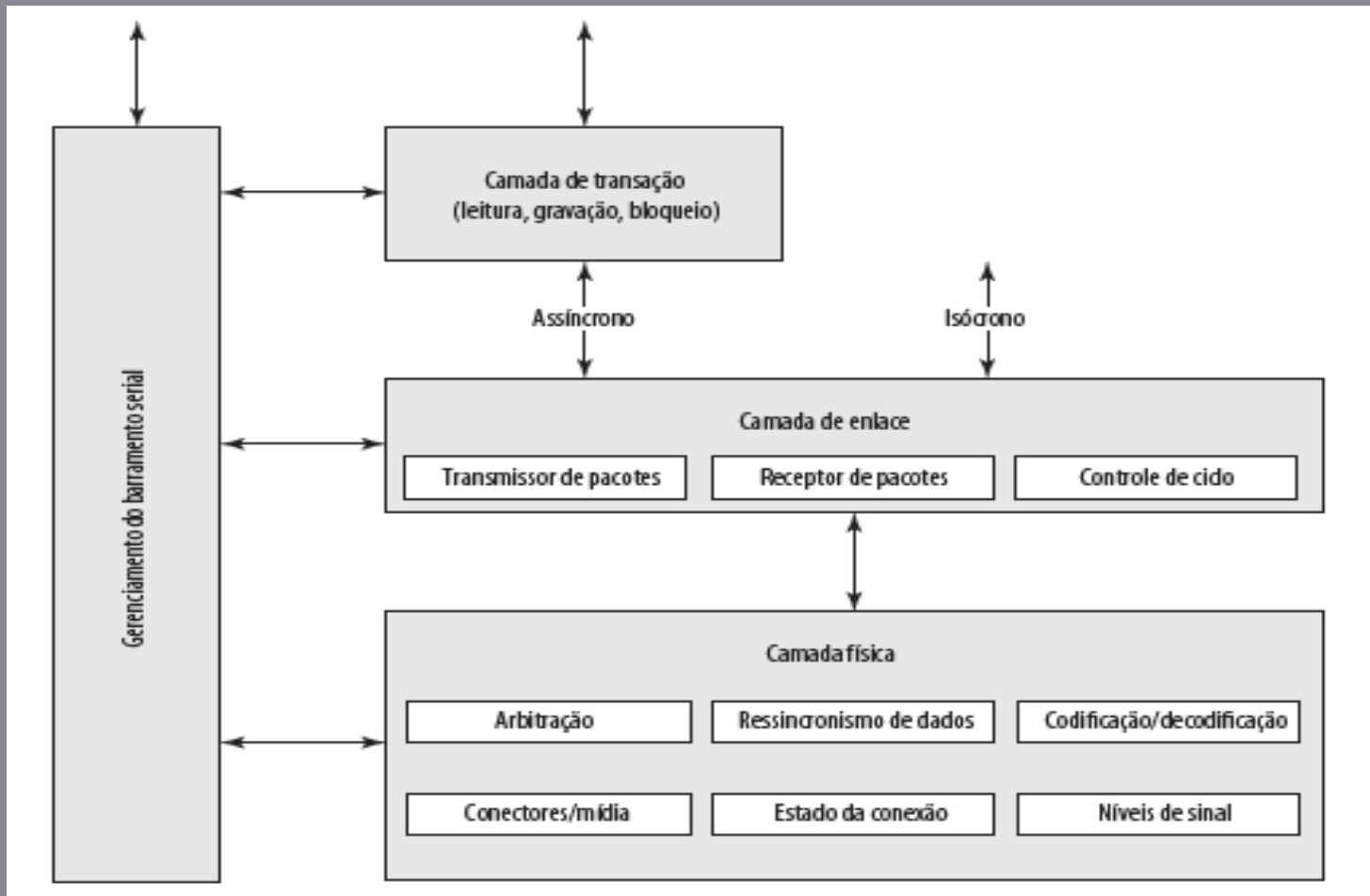
## Configuração FireWire simples



## 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





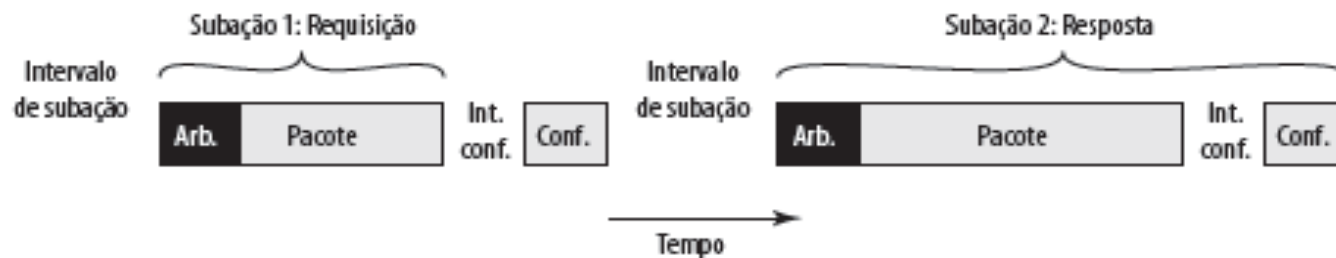
## Fire Wire – Camada física

- Taxas de dados de 25 a 400Mbps.
- Duas formas de arbitração (arranjo e linear):
  - 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.
  - Arbitração imparcial.
  - Arbitração urgente.

## 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.

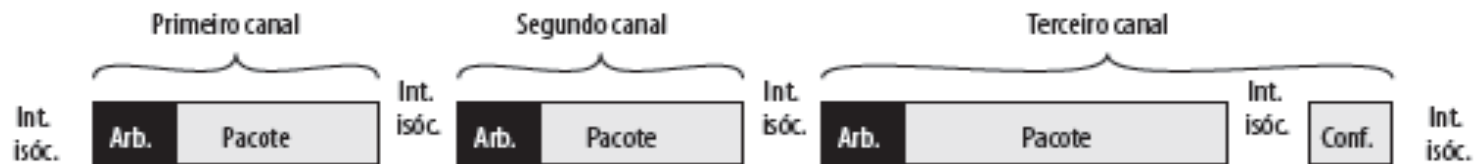
# Subações do Fire Wire



(a) Exemplo de subação assíncrona



(b) Exemplo de subações assíncronas concatenadas



(c) Exemplo de subações isócronas

## FIREWIRE VERSUS USB 3.0

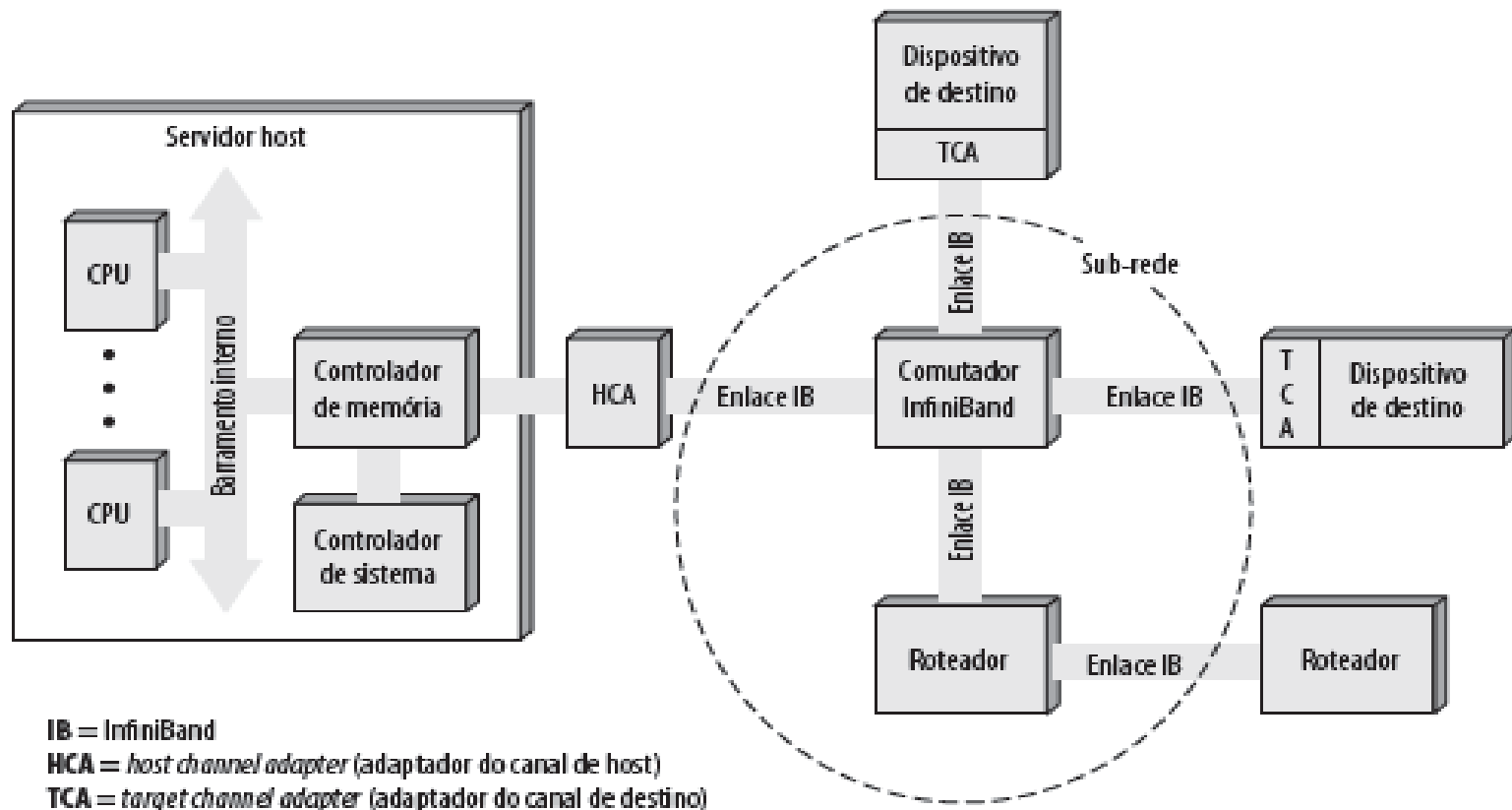
## InfiniBand

- 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 modo único.
- Até 30Gbps.

## Fábrica de comutadores InfiniBand



# ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES



Effective theoretical throughput in different configurations (the actual data rate, not the signaling rate)

	SDR	DDR	QDR	FDR	EDR	HDR	NDR
1X	2 Gbit/s	4 Gbit/s	8 Gbit/s	14 Gbit/s	25 Gbit/s	125 Gbit/s	750 Gbit/s
4X	8 Gbit/s	16 Gbit/s	32 Gbit/s	56 Gbit/s	100 Gbit/s	500 Gbit/s	3000 Gbit/s
12X	24 Gbit/s	48 Gbit/s	96 Gbit/s	168 Gbit/s	300 Gbit/s	1500 Gbit/s	9000 Gbit/s

**InfiniBand** is a switched fabric communications link used in high-performance computing and enterprise data centers. Its features include high throughput, low latency, quality of service and failover, and it is designed to be scalable. The InfiniBand architecture specification defines a connection between processor nodes and high performance I/O nodes such as storage devices. InfiniBand forms a superset of the Virtual Interface Architecture.

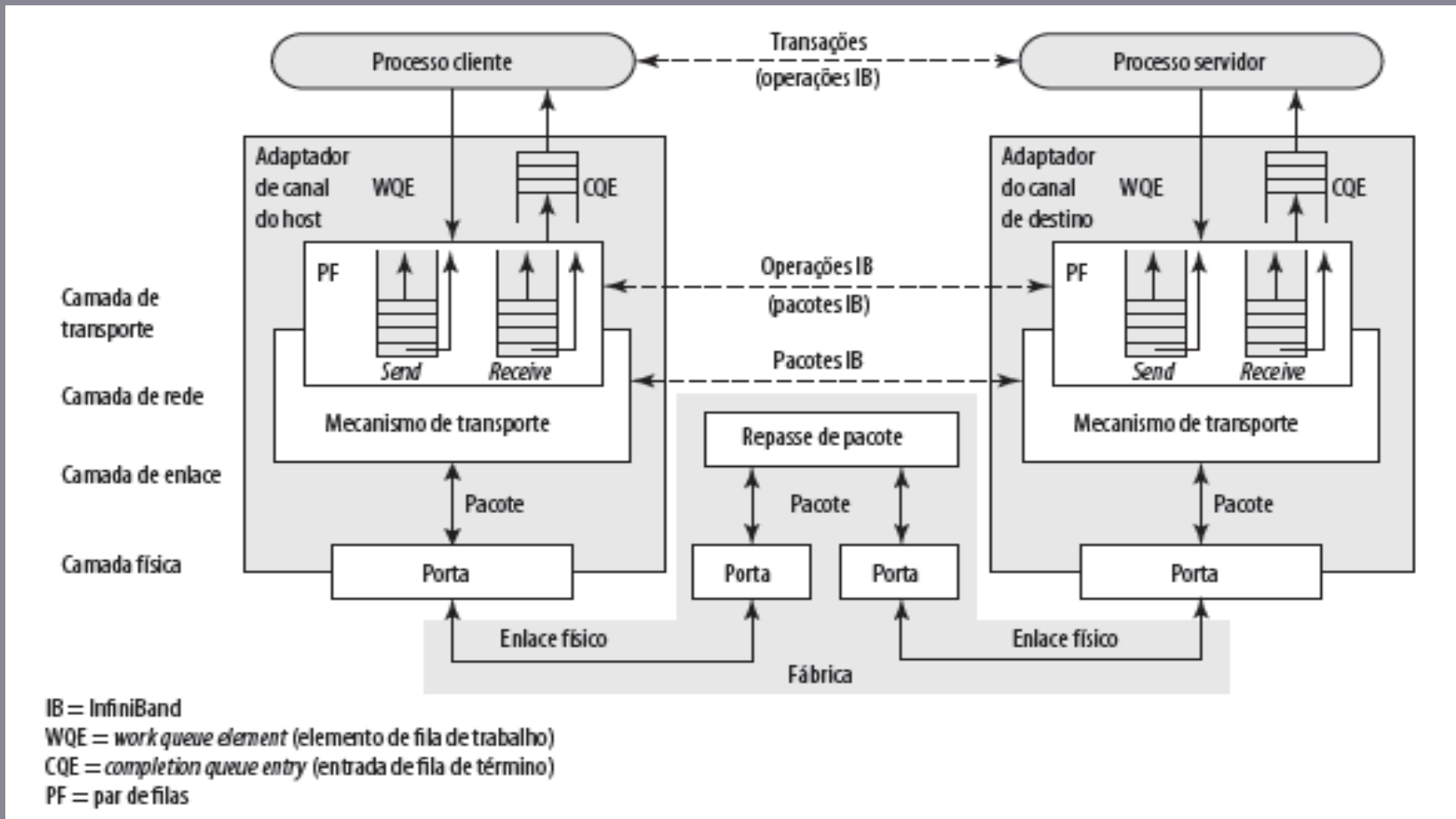
Maiores informações: <http://en.wikipedia.org/wiki/InfiniBand>



## 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 a transferência fim a fim.
- Comutador mapeia tráfego da pista de entrada para pista de saída.

## Pilha de protocolos InfiniBand



## **Leitura de base**

- Procure por Universal Serial Bus (USB).
- Compare com outros padrões de comunicação, p.e., Ethernet.