

AOC: DISPOSITIVOS DE E/S – parte 1

Referências:

OPC: A interface HW&SW, HENNESSY, J. L., 4ª edição, 2009, Cap 1, 4 e 5.

AOC - William Stallings – cap 3 e 6

OEC - A.S. Tanenbaum – cap 2 e 3

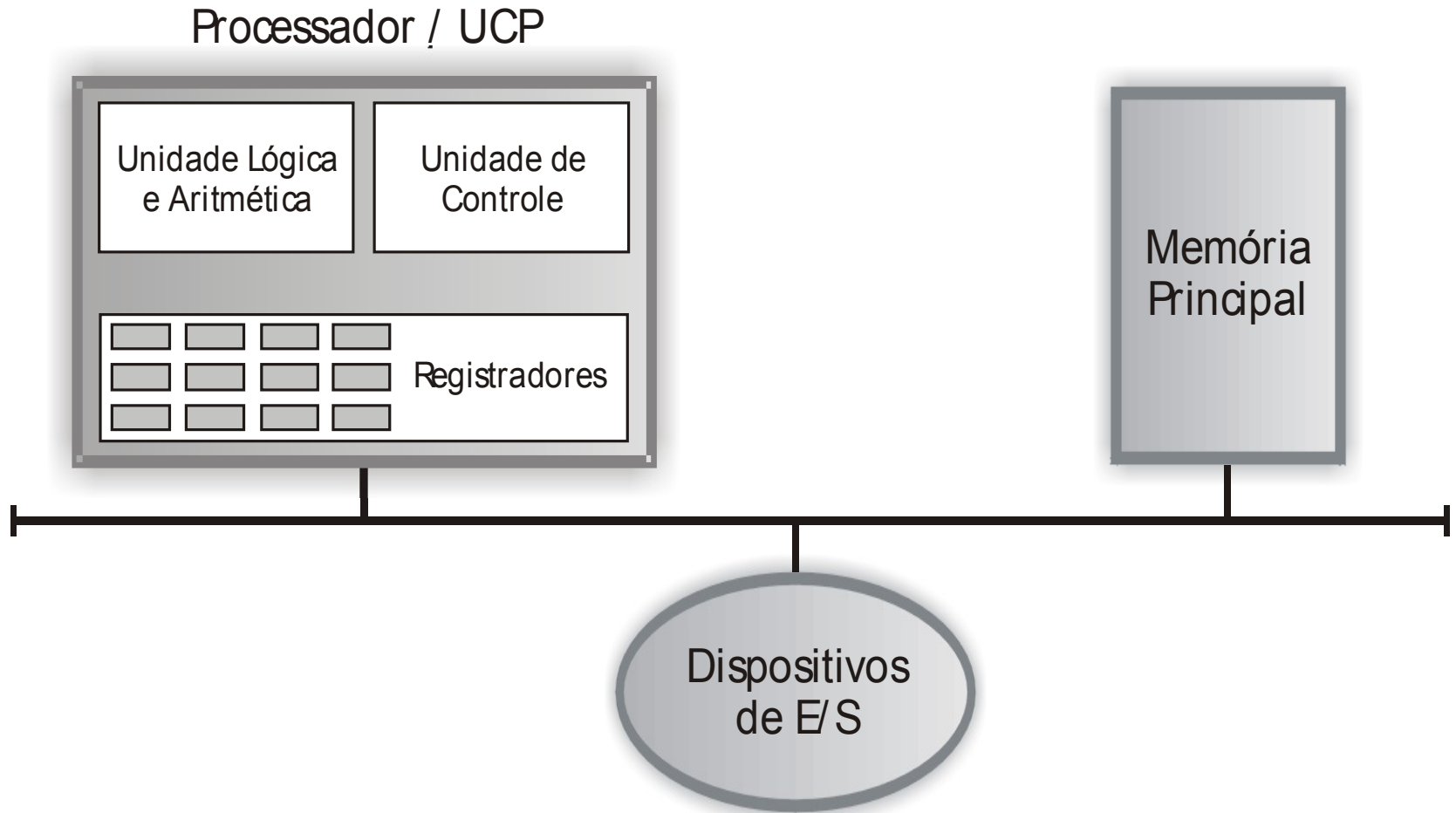
SOM - A.S. Tanenbaum (2ª. ed 2003) – cap 1, 5, 10 e 11

ASO - F. B. Machado (4a. ed 2007) - cap 1 a 4, 12, 14 e 15

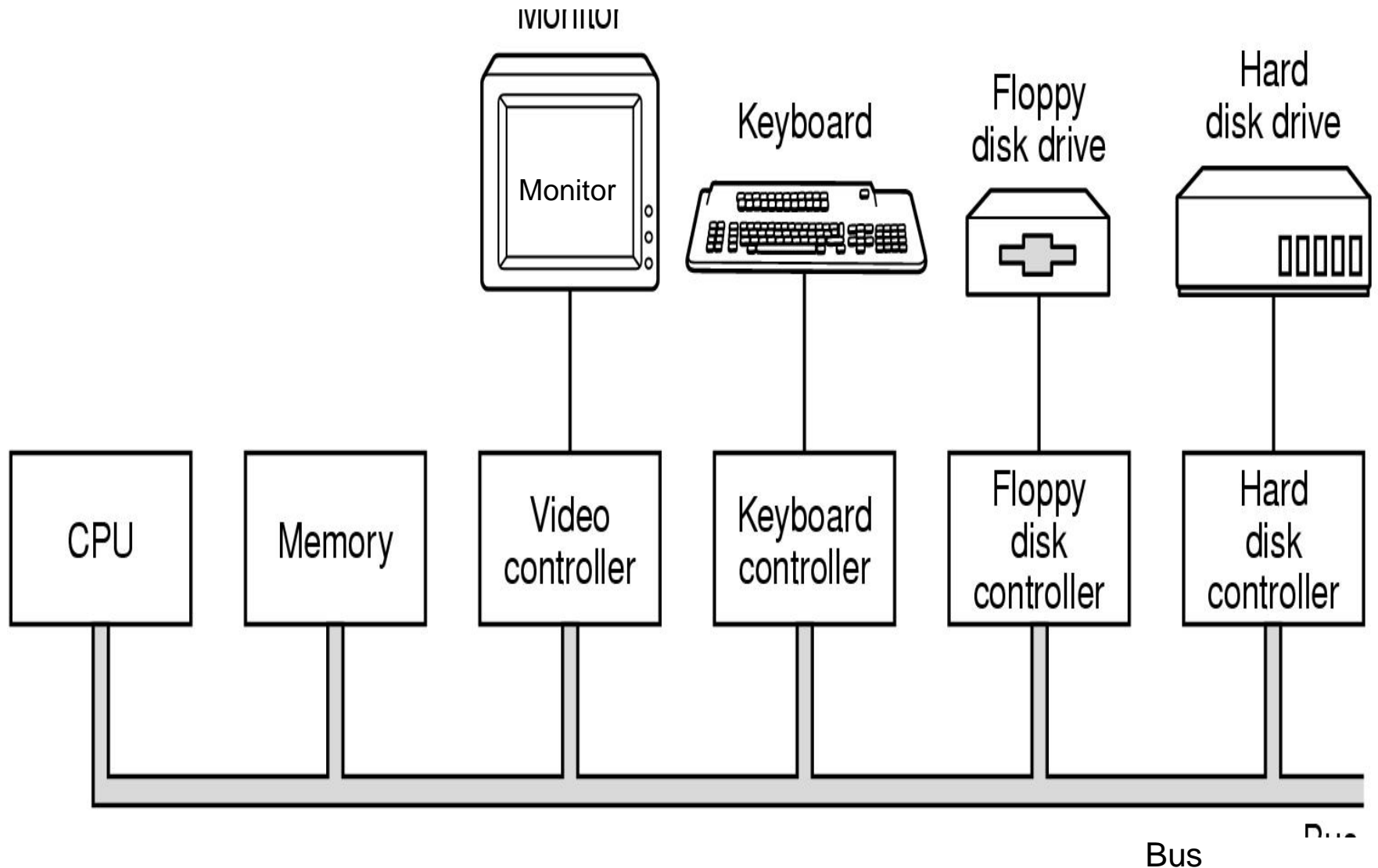
SO com Java - Silberschatz – cap 13, 14, 20 e 21

SO - Deitel (3a. Ed. 2005) - cap 2, 20 e 21

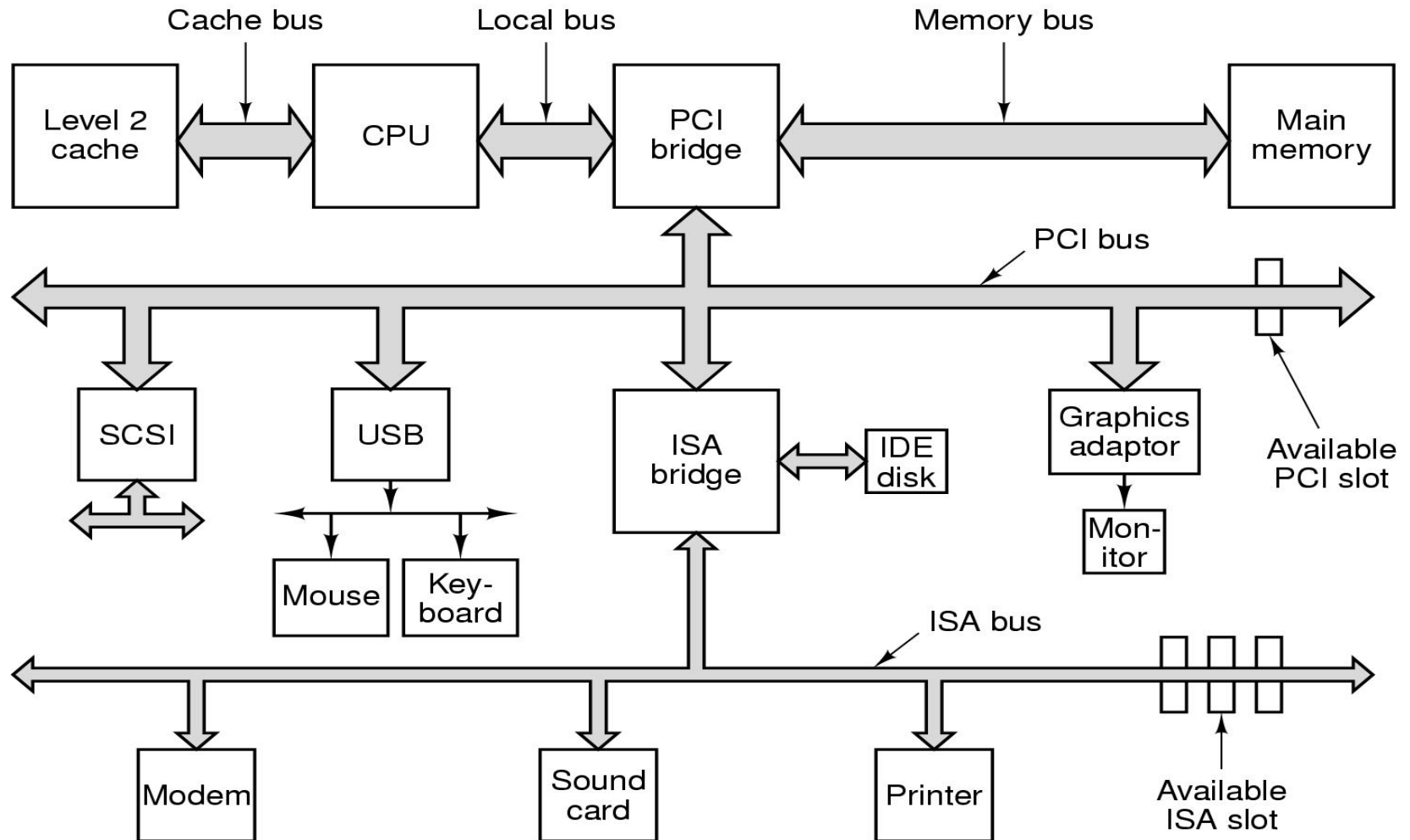
Hardware: principais componentes



HADWARE : COMPONENTES



CONEXÃO E/S e UCP



Dispositivos periféricos (Fig 2.5 SODeitel3ed)

<i>Dispositivo</i>	<i>Descrição</i>
Unidade de CD-RW	Lê e grava dados de e para discos óticos.
Unidade de Zip	Transfere dados de e para um disco magnético durável removível.
Unidade de disco flexível	Lê e grava dados de e para discos magnéticos removíveis.
Mouse	Transmite a mudança de localização de um ponteiro ou cursor em uma interface gráfica com o usuário (GUI).
Teclado	Transmite caracteres ou comandos digitados por um usuário.
Impressora multifuncional	Pode imprimir, copiar, enviar fax e escanear documentos.
Placa de som	Converte sinais digitais em sinais de áudio para alto-falantes. Também pode receber sinais de áudio via microfone e produzir um sinal digital.
Acelerador de vídeo	Exibe gráficos na tela; acelera gráficos bi e tridimensionais.
Placa de rede	Envia e recebe dados de e para outros computadores.
Câmera digital	Grava e muitas vezes exibe imagens digitais.
Dispositivo biométrico	Executa varredura (scan) de características humanas como impressões digitais e retinas, normalmente para finalidades de identificação e autenticação.
Dispositivo de infravermelho	Comunica dados entre dispositivos via conexão sem fio em linha de visada.
Dispositivo sem fio	Comunica dados entre dispositivos via conexão sem fio onidirecional.

Dispositivos periféricos (SODeitel3ed)

- Qualquer dispositivo de hardware não requerido por um computador para executar instruções de software.
- Os dispositivos internos são referidos como dispositivos periféricos integrados.
- Placas de interface de rede, modems, placas de som.
- Unidades de disco rígido, CD e DVD.
- Dispositivos de caracteres: transferem dados: um caractere por vez.
- Teclados e mouses
- Podem ser conectados a um computador por meio de portas e outros barramentos.
- Portas seriais, portas paralelas, USB, portas IEEE 1394 e SCSI

Taxas de dados típicas de dispositivos, redes e barramentos (SOMTB2ed)

Dispositivo	Taxa de dados
Teclado	10 bytes/s
Mouse	100 bytes/s
Modem 56 K	7 KB/s
Canal telefónico	8 KB/s
Linhas ISDN dual	16 KB/s
Impressora a laser	100 KB/s
Scanner	400 KB/s
Ethernet clássica	1,25 MB/s
USB (<i>universal serial bus</i> — barramento serial universal)	1,5 MB/s
Câmara de vídeo digital	4 MB/s
Disco IDE	5 MB/s
CD-ROM 40x	6 MB/s
Ethernet rápida	12,5 MB/s
Barramento ISA	16,7 MB/s
Disco EIDE (ATA-2)	16,7 MB/s
FireWire (IEEE 1394)	50 MB/s
Monitor XGA	60 MB/s
Rede SONET OC-12	78 MB/s
Disco SCSI Ultra 2	80 MB/s
Ethernet Gigabit	125 MB/s
Dispositivo de Fita Ultrium	320 MB/s
Barramento PCI	528 MB/s
Barramento da Sun Gigaplane XB	20 GB/s

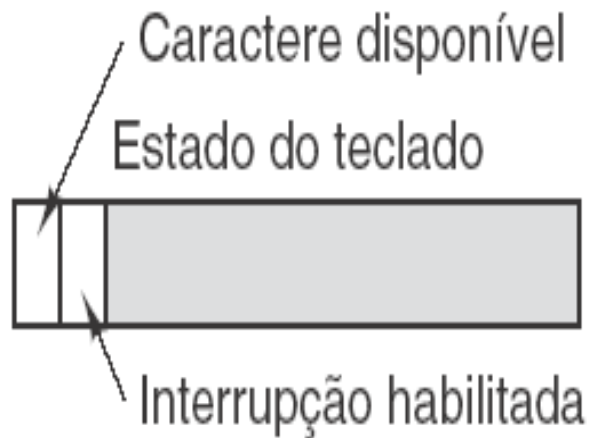
Controladores de Dispositivos (SOMTB2ed)

- Componentes de dispositivos de E/S
 - mecânico
 - eletrônico
- O componente eletrônico é o controlador do dispositivo
 - pode ser capaz de tratar múltiplos dispositivos
- Tarefas do controlador
 - converter fluxo serial de bits em bloco de bytes
 - executar toda correção de erro necessária
 - tornar o bloco disponível para ser copiado para a memória principal

CONTROLADOR/ADAPTADOR E/S

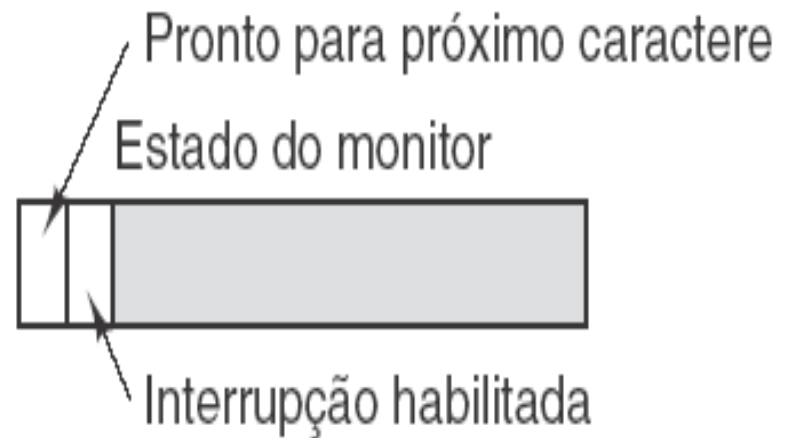
- Os controladores de dispositivos são geralmente programáveis e possuem vários registradores internos, cada um podendo ser acessado por um endereço. Exemplos:
 - COMANDO(S)
 - PARÂMETROS
 - ESTADO
 - DADOS (E/S)

Registradores de dispositivo: terminal simples (OECTB5ed)



Buffer do teclado

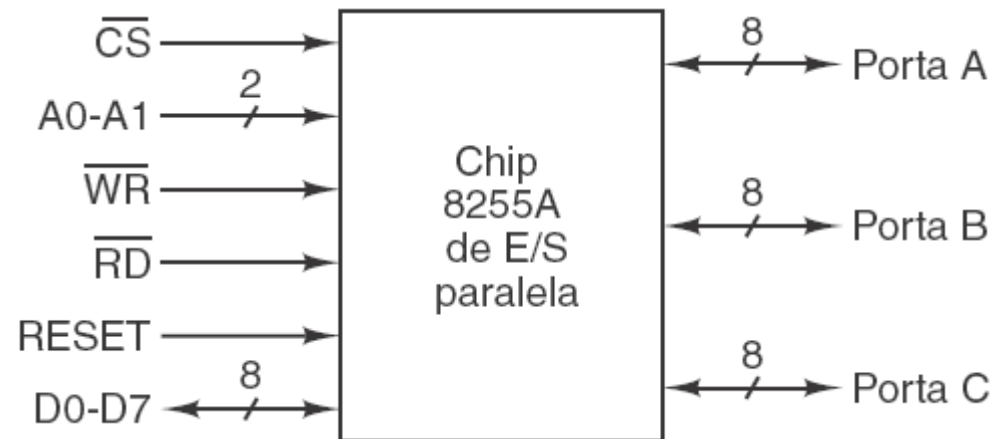
Caractere recebido



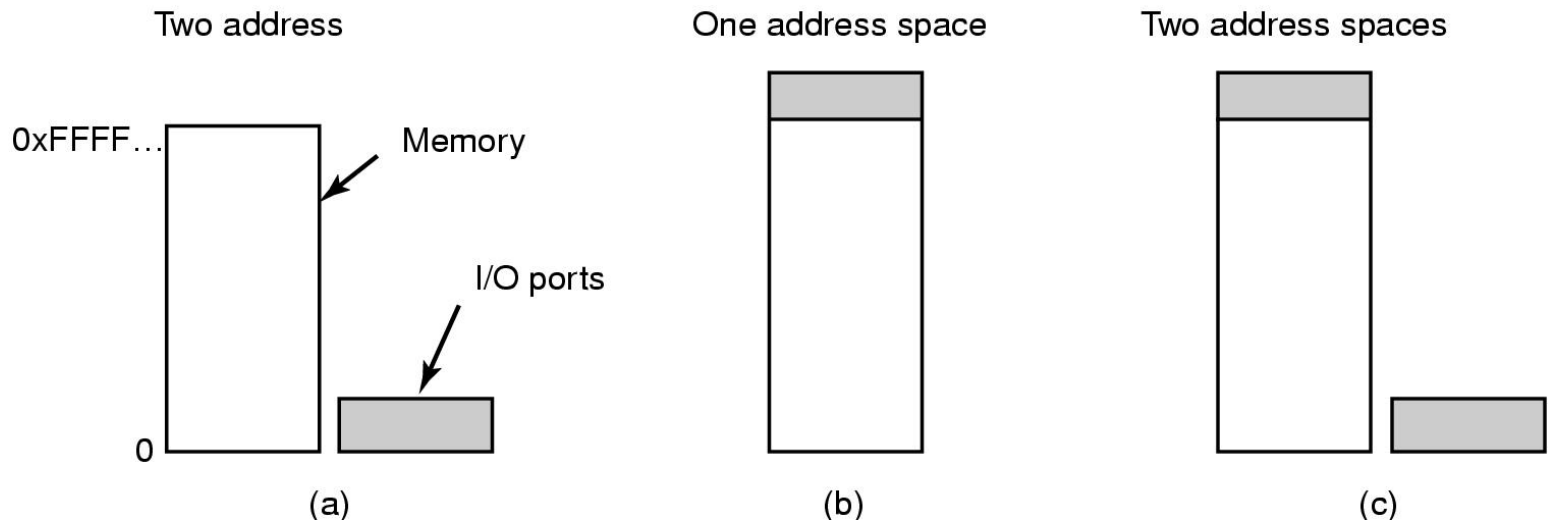
Buffer do monitor

Caractere a apresentar

Chip E/S: 8255A PIO (OECTB5ed)

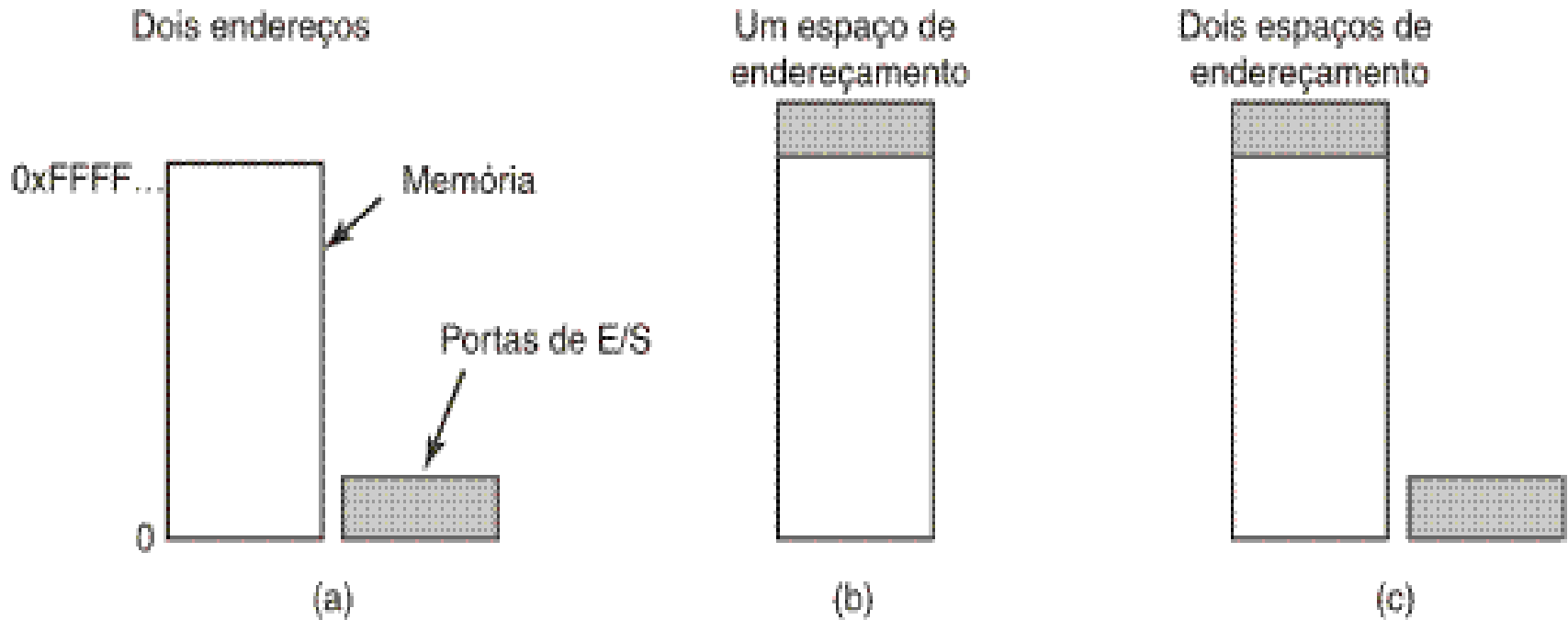


Comunicação UCP – Controlador E/S



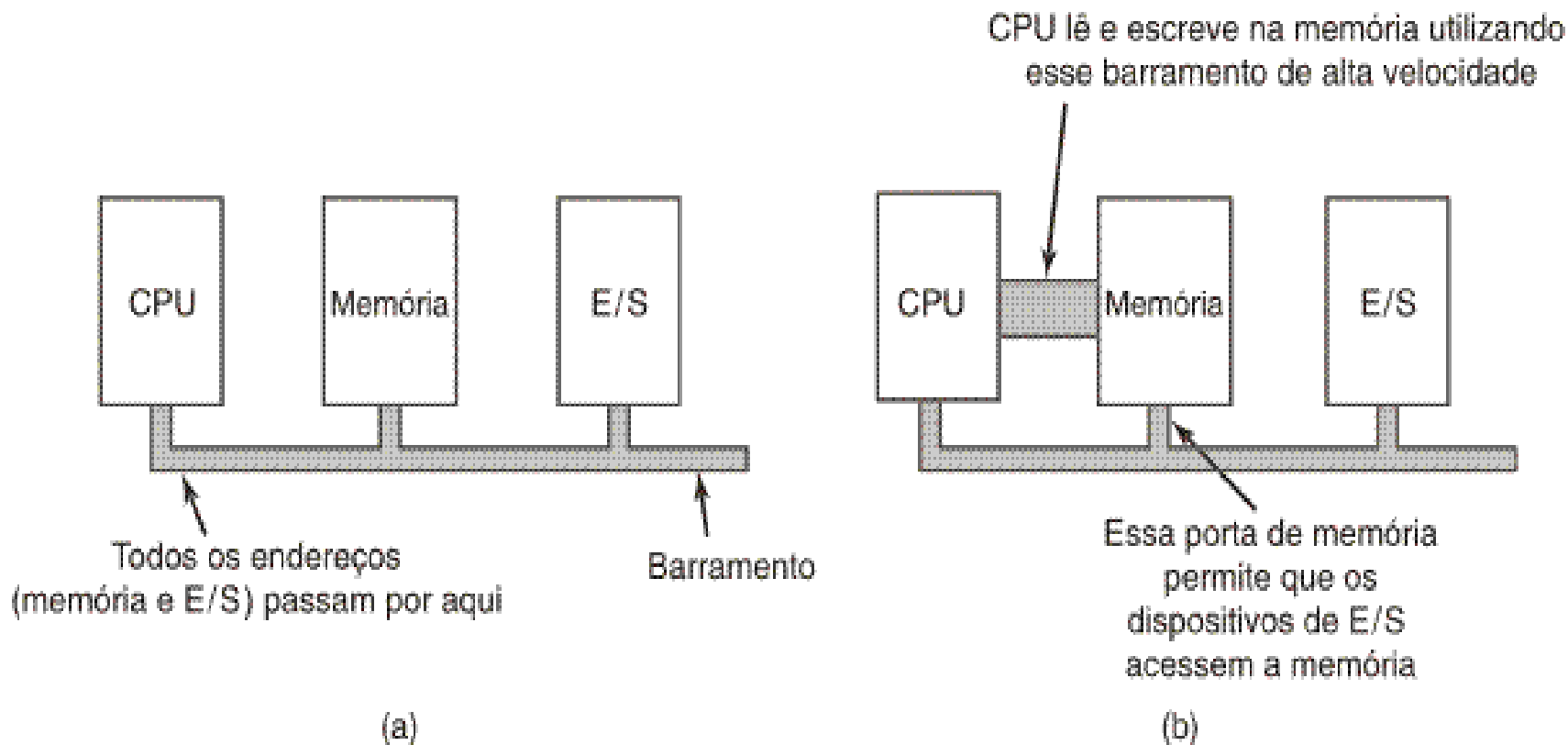
- Espaços separados E/S e Memória
- E/S Mapeada em Memória
- Sistema Híbrido

E/S mapeada na memória (SOMTB2ed)



- a) Espaços de memória e E/S separados
- b) E/S mapeada na memória
- c) Híbrido

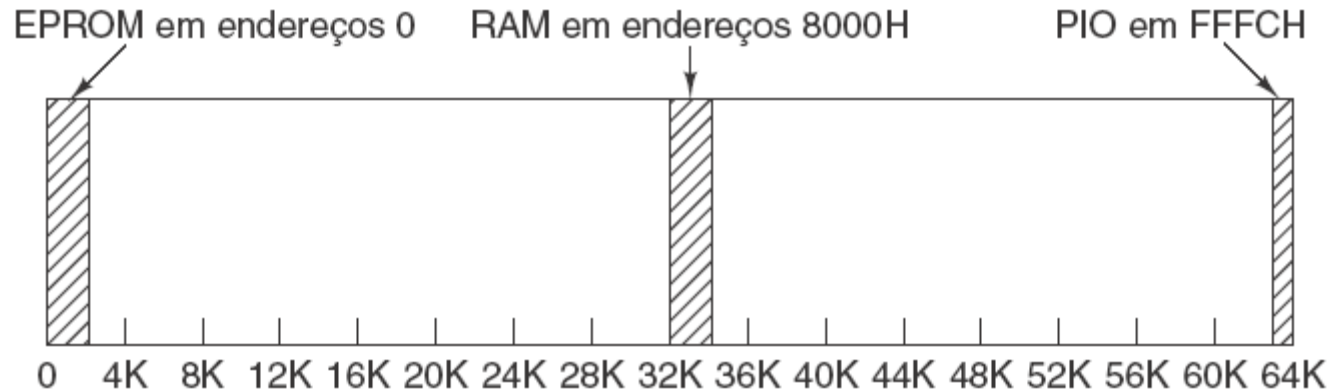
E/S mapeada na memória (SOMTB2ed)



(a) Arquitetura com barramento único

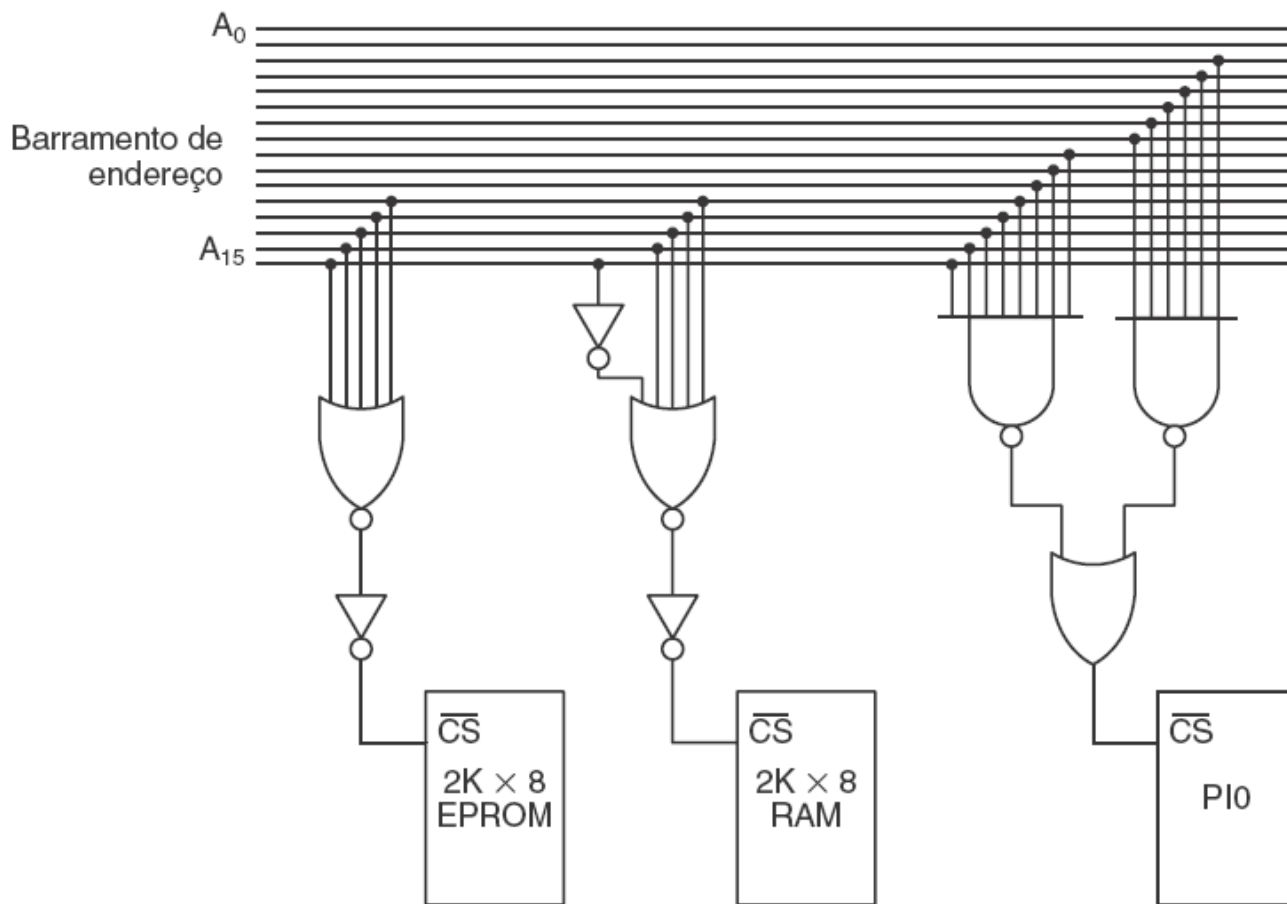
(b) Arquitetura com barramento dual

E/S mapeada na memória (OECTB5ed)



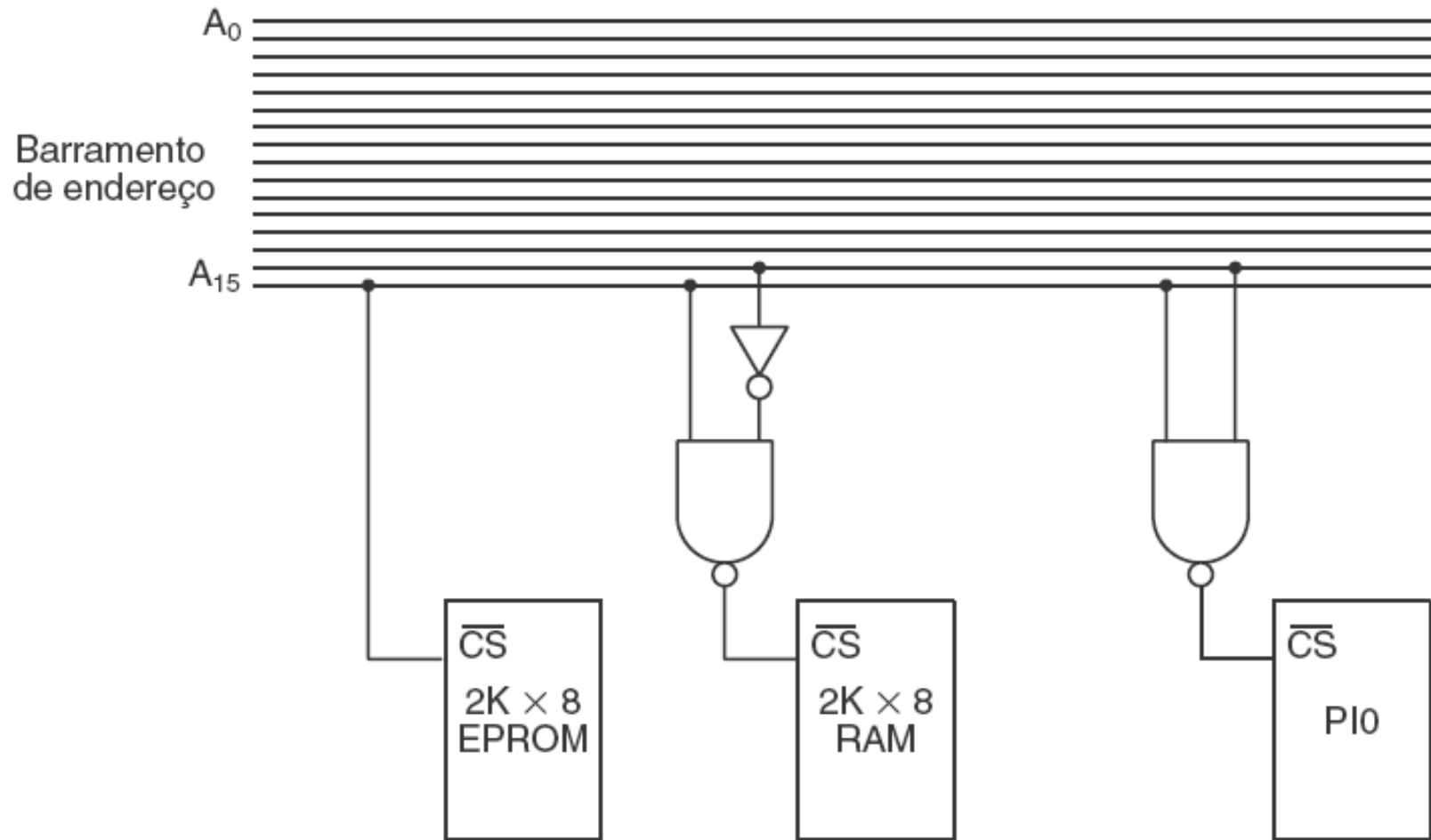
Localização de EPROM, RAM e PIO: espaço de endereço de 64 KB.

Decodificação de endereço completo (OECTB5ed)



(a)

Decodificação parcial de endereço (OECTB5ed)



(b)

MÉTODOS DE E/S

1. CONTROLADA PROGRAMADA
("POLLING")

3. POR INTERRUPÇÃO

5. POR DMA (ACESSO DIRETO À
MEMÓRIA)

E/S PROGRAMADA (*polling*)

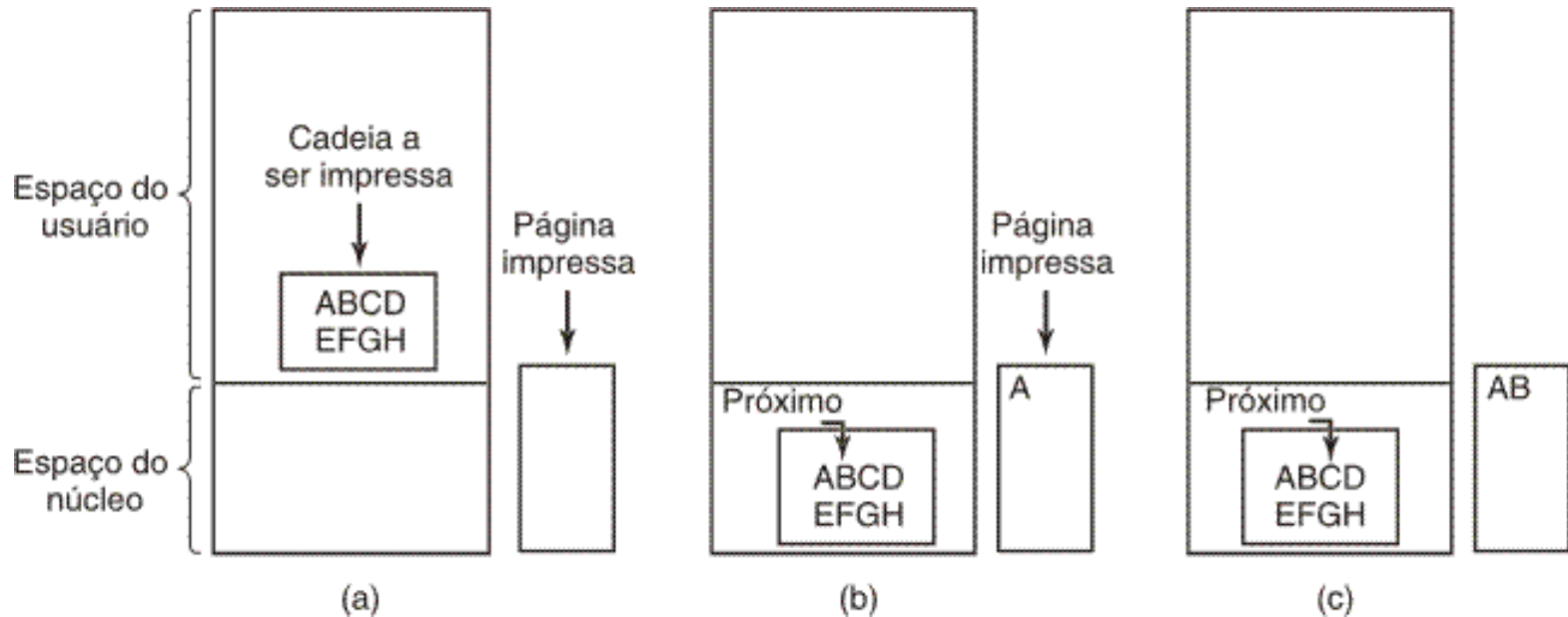
ESPERA:

```
IN AL, REG_ESTADO_CNTRL # LER ESTADO
CMP AL, 0      # 0 (LIVRE) e 1 (OCUPADO)
JNE ESPERA # DESVIA SE ESTIVER OCUPADO
IN  AL, REG_DADO_CNTRL # LER DADO
```

COMENTARIO:

PROBLEMA: ESPERA OCUPADA OU ATIVA
("BUSY WAITING")

E/S Programada (SOMTB2ed) - 1



Passos da impressão de uma cadeia de caracteres

Exemplo: E/S programada (OECTB5ed)

```
public static void output_buffer(char buf[ ], int count) {  
    // Produza um bloco de dados para o  
    int status do dispositivo, i, ready;  
    for (i = 0; i < count; i++) {  
        do {  
            status = in(display_status_reg);           // obtenha estado  
            ready = (status >> 7) & 0x01;              // isole bit ready  
        } while (ready != 1);  
        out(display_buffer_reg, buf[i]);  
    }  
}
```

E/S Programada (SOMTB2ed) - 2

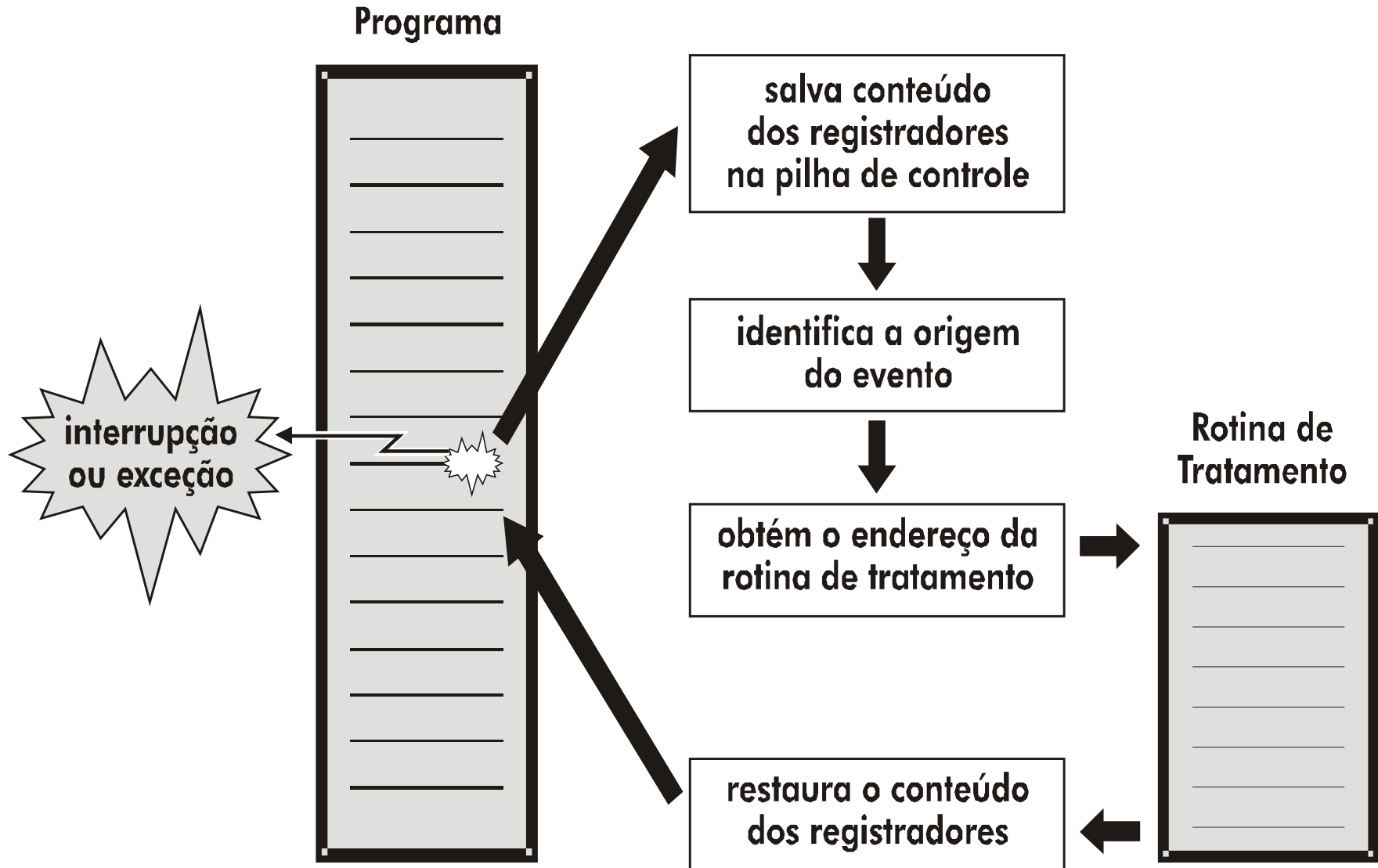
```
copy_from_user(buffer, p, cont);           /* p é o buffer do núcleo */
for (i=0; i < count; i++) {                 /* executa o laço para cada caractere */
    while (*printer_status_reg !=READY);     /* executa o laço até PRONTO */
    *printer_data_register = p[i];           /* envia um caractere para a saída */
}
return_to_user();
```

Escrita de uma cadeia de caracteres para a impressora
usando E/S programada

INTERRUPÇÕES E EXCEÇÕES

- São desvios do fluxo de execução de um programa
- São diferentes de branches e jumps (que também mudam o fluxo normal de execução de instruções).
- As interrupções, segundo a Intel, podem ser de hardware e de software.

INTERRUPÇÕES E EXCEÇÕES



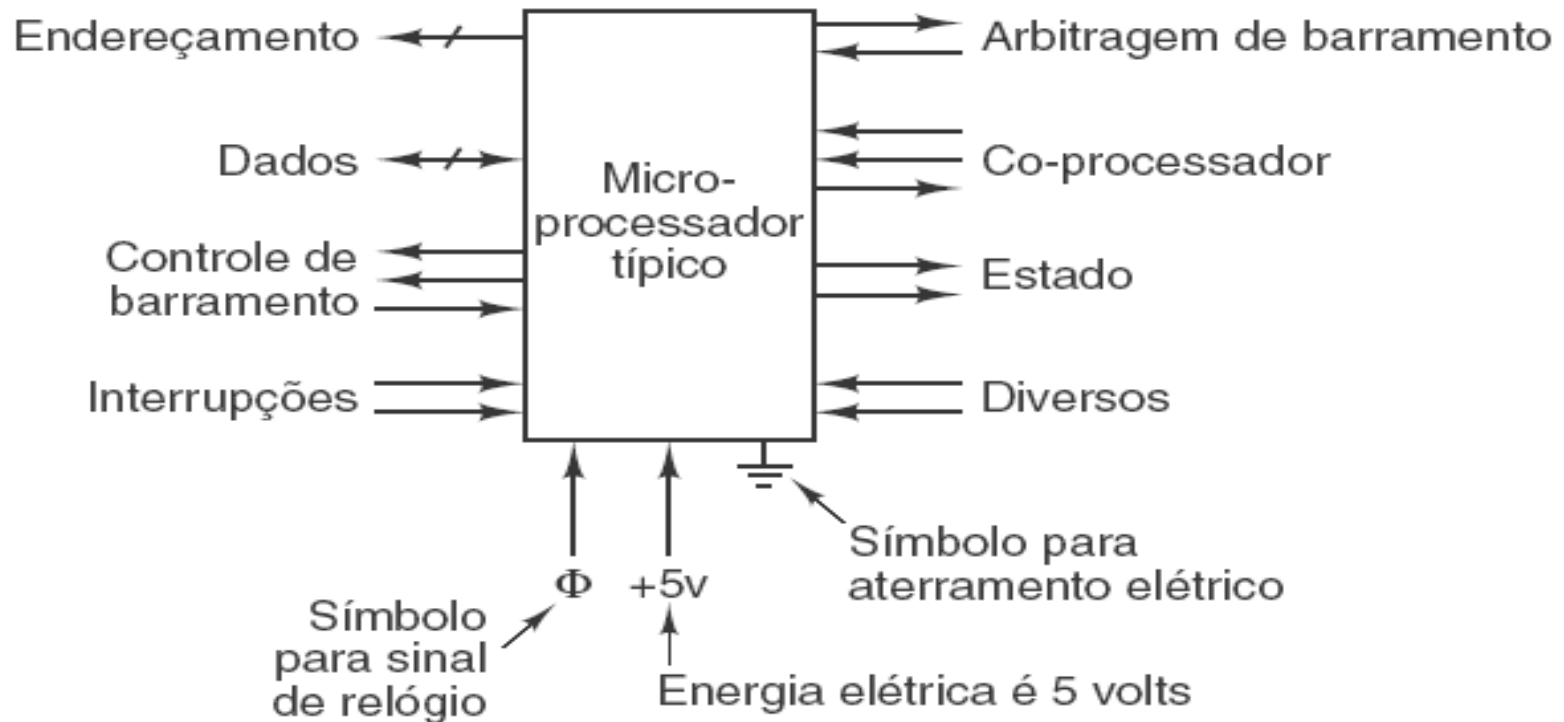
EXCEÇÕES

- Desvios forçados do fluxo de execução de um programa causado por um evento interno síncrono, ou seja, tem origem dentro do processador e ocorrem devido a própria execução das instruções do programa.
- São geralmente geradas em resposta a erros ou condições de exceção. Exemplos: divisão por zero, overflow aritmético, falta de página, erro de segmentação, instrução inválida, etc.

INTERRUPÇÃO DE HARDWARE

- Desvio **forçado** do fluxo de execução de um programa causado por um **evento assíncrono externo** (relógio, teclado, disco, impressora, etc), ou seja, é ocasionada por um evento inesperado que tem origem fora do processador.
- Exemplos: digitar de um caractere no teclado, movimentar o mouse, chegar uma mensagem pela rede, etc.
- Quando ocorre um evento, a maioria dos dispositivos de E/S envia ao processador um sinal denominado interrupção.
- O sistema operacional pode responder a uma interrupção notificando os programas que estão à espera desses eventos.

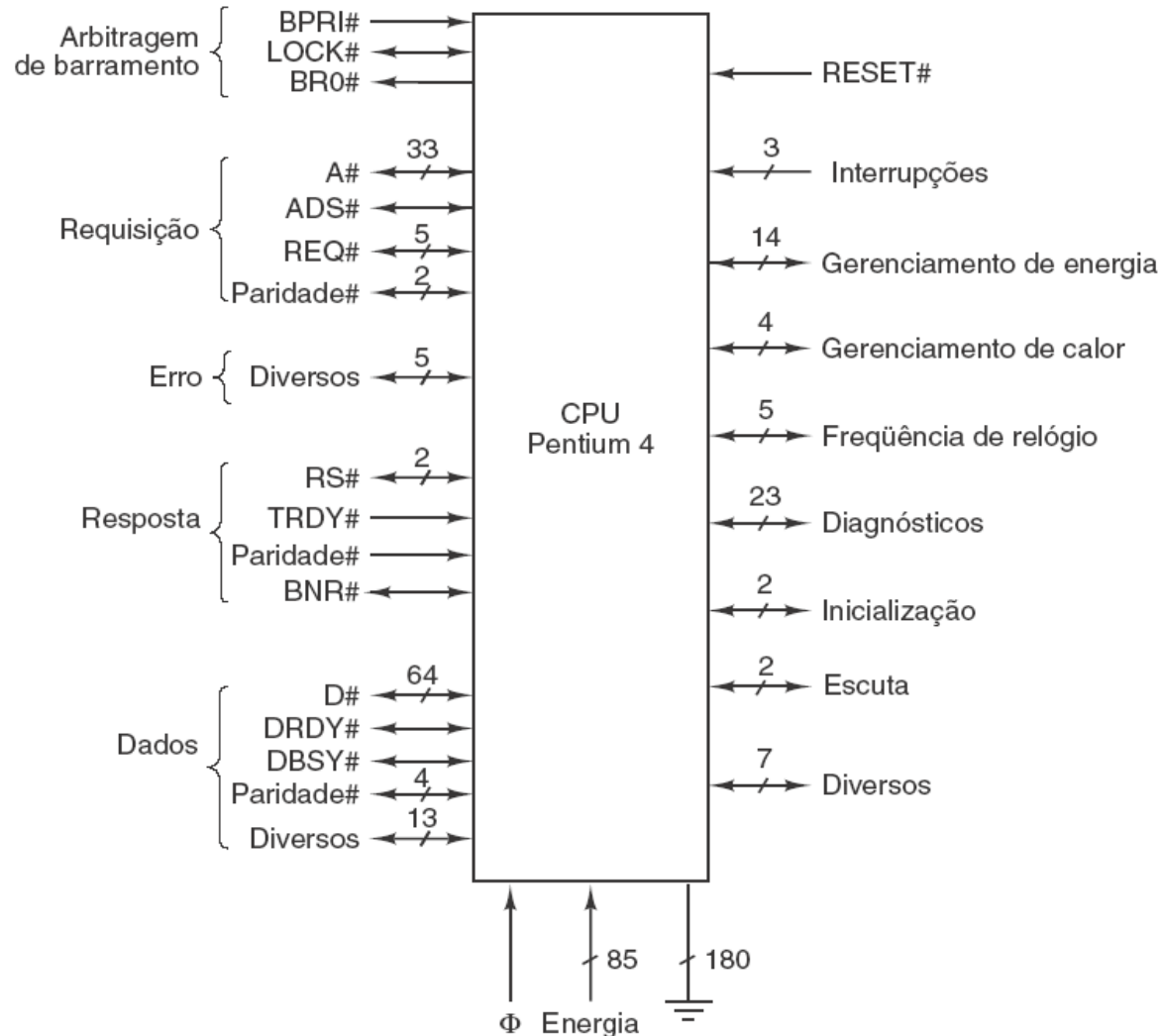
Pinagem lógica de uma CPU genérica (OECTB5ed)



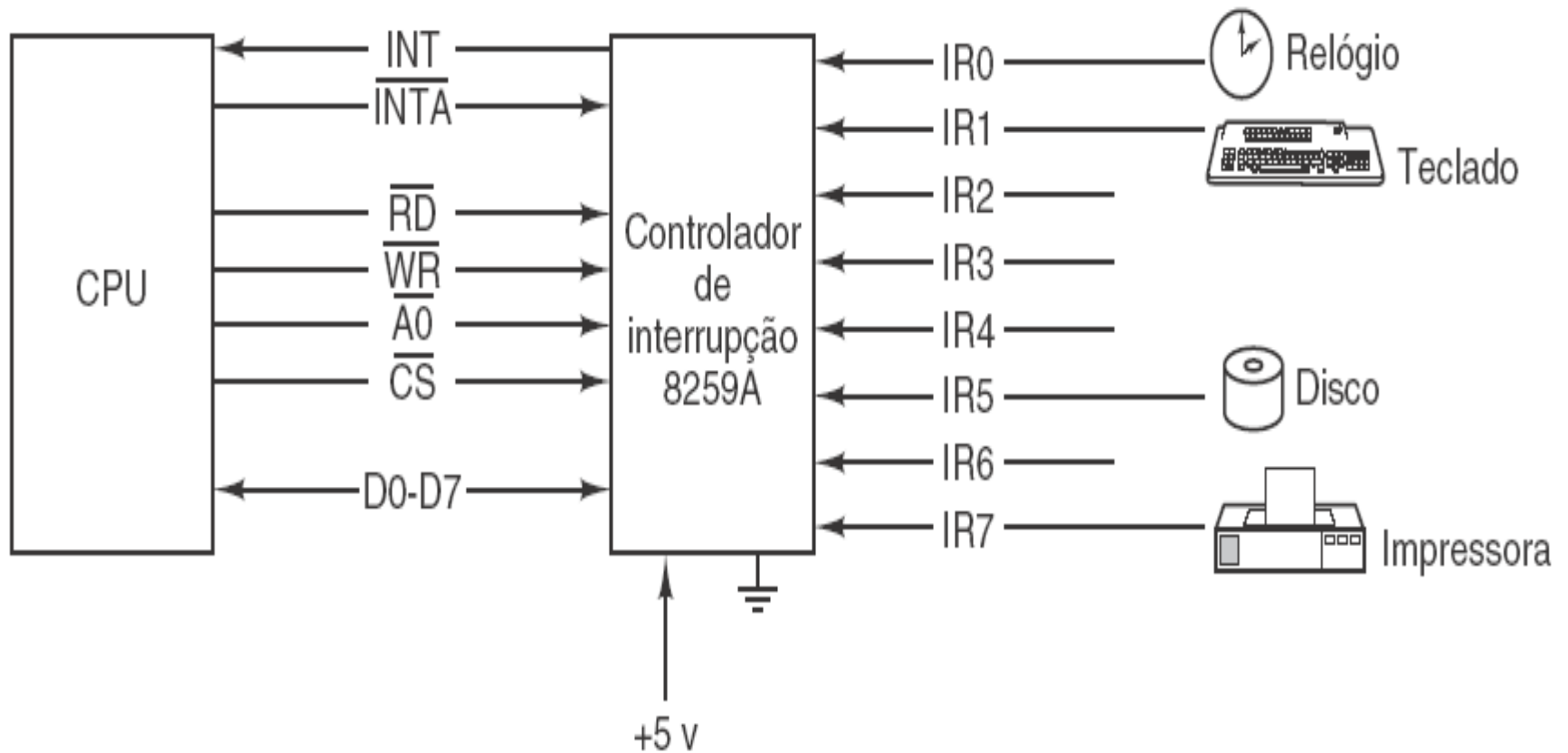
- As setas indicam sinais de entrada e sinais de saída.
- Os segmentos de reta diagonal indicam que são utilizados vários pinos.

Pinagem lógica do Pentium 4 (OECTB5ed)

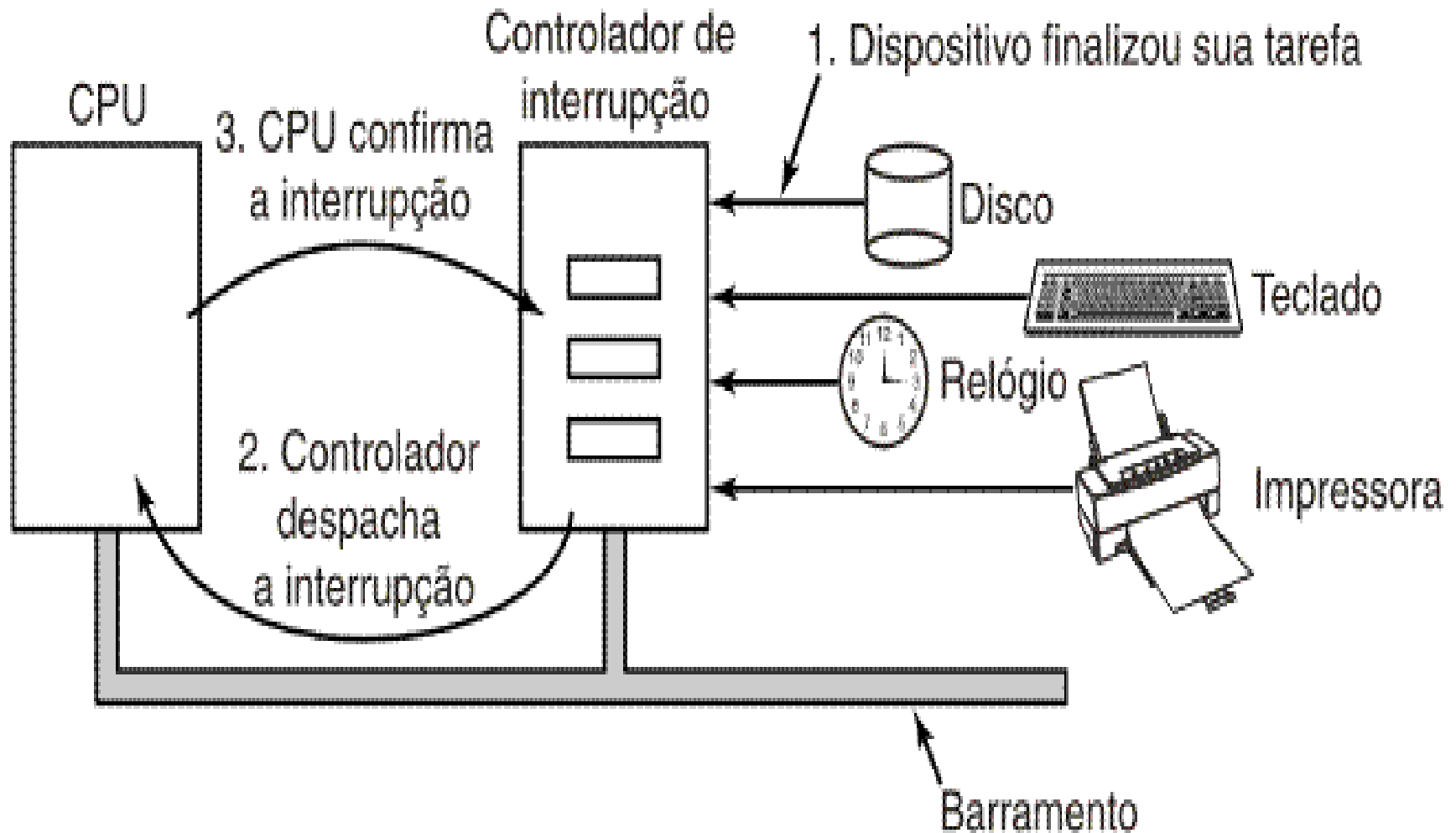
- Sinais individuais: letras maiúsculas
- Grupos de sinais relacionados ou descrições de sinais: letras maiúsculas e minúsculas.



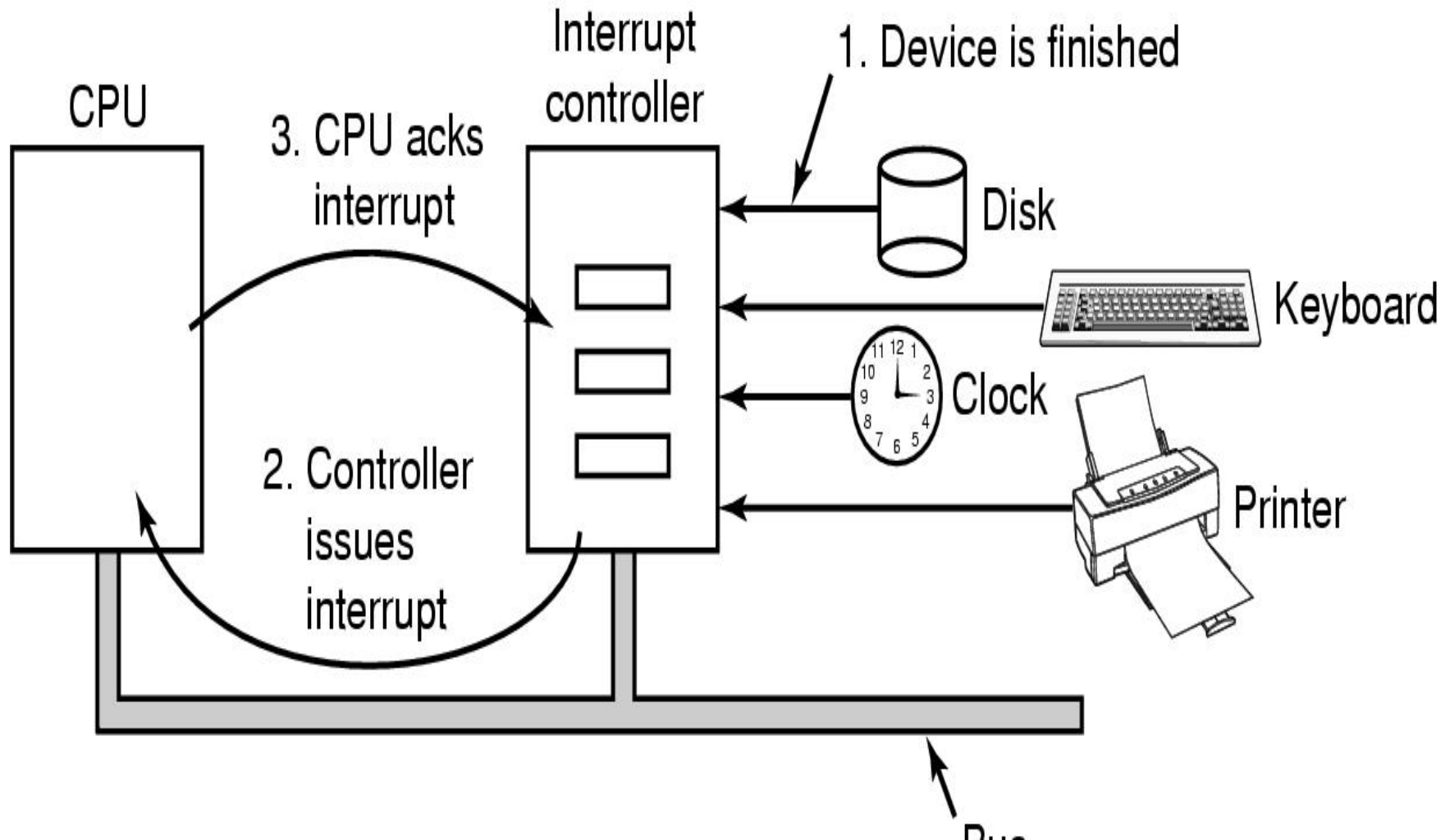
PIC: Controlador de interrupção 8259A (OECTB5ed)



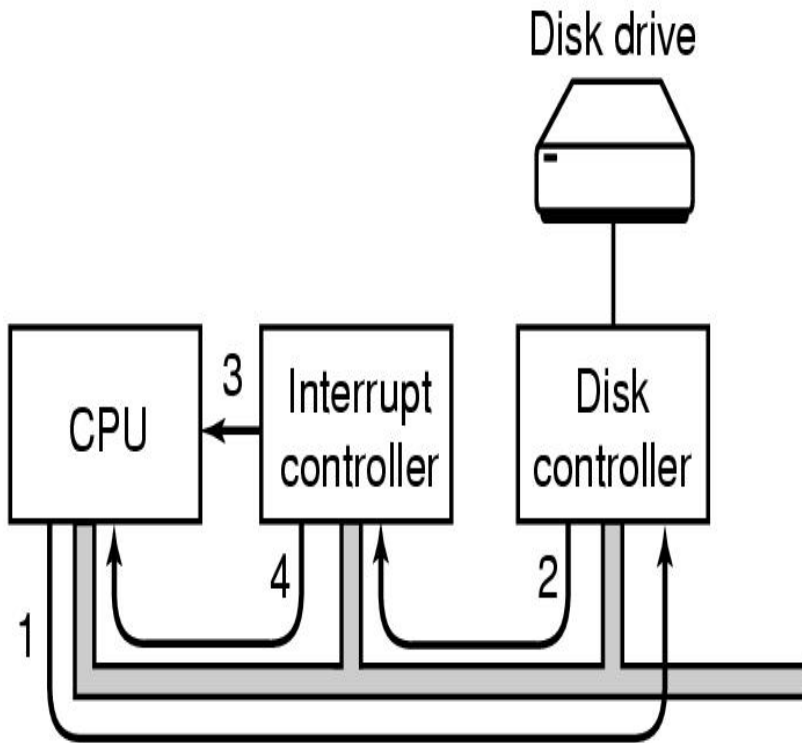
Interrupções de hardware (SOMTB2ed)



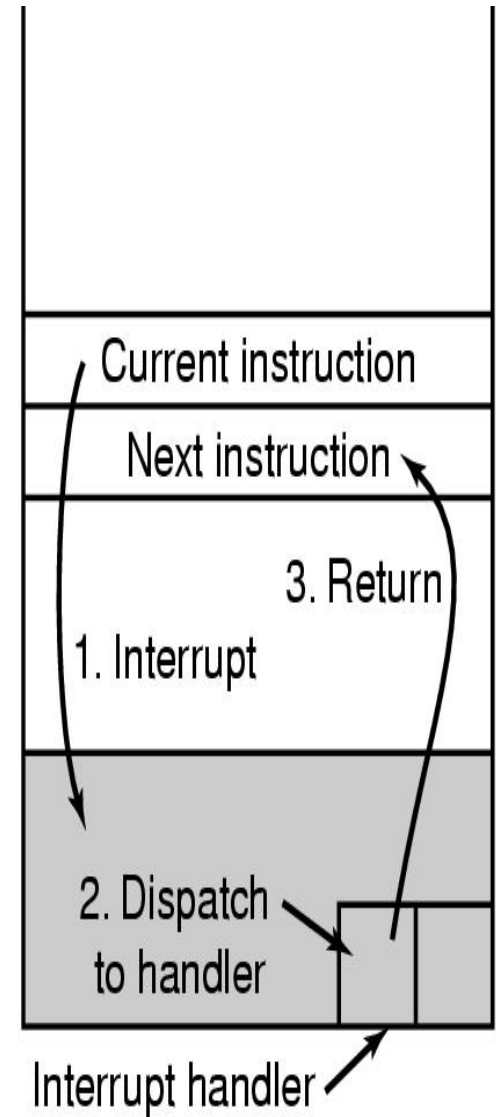
INTERRUPÇÃO DE HARDWARE (1)



INTERRUPÇÃO DE HARDWARE (2)

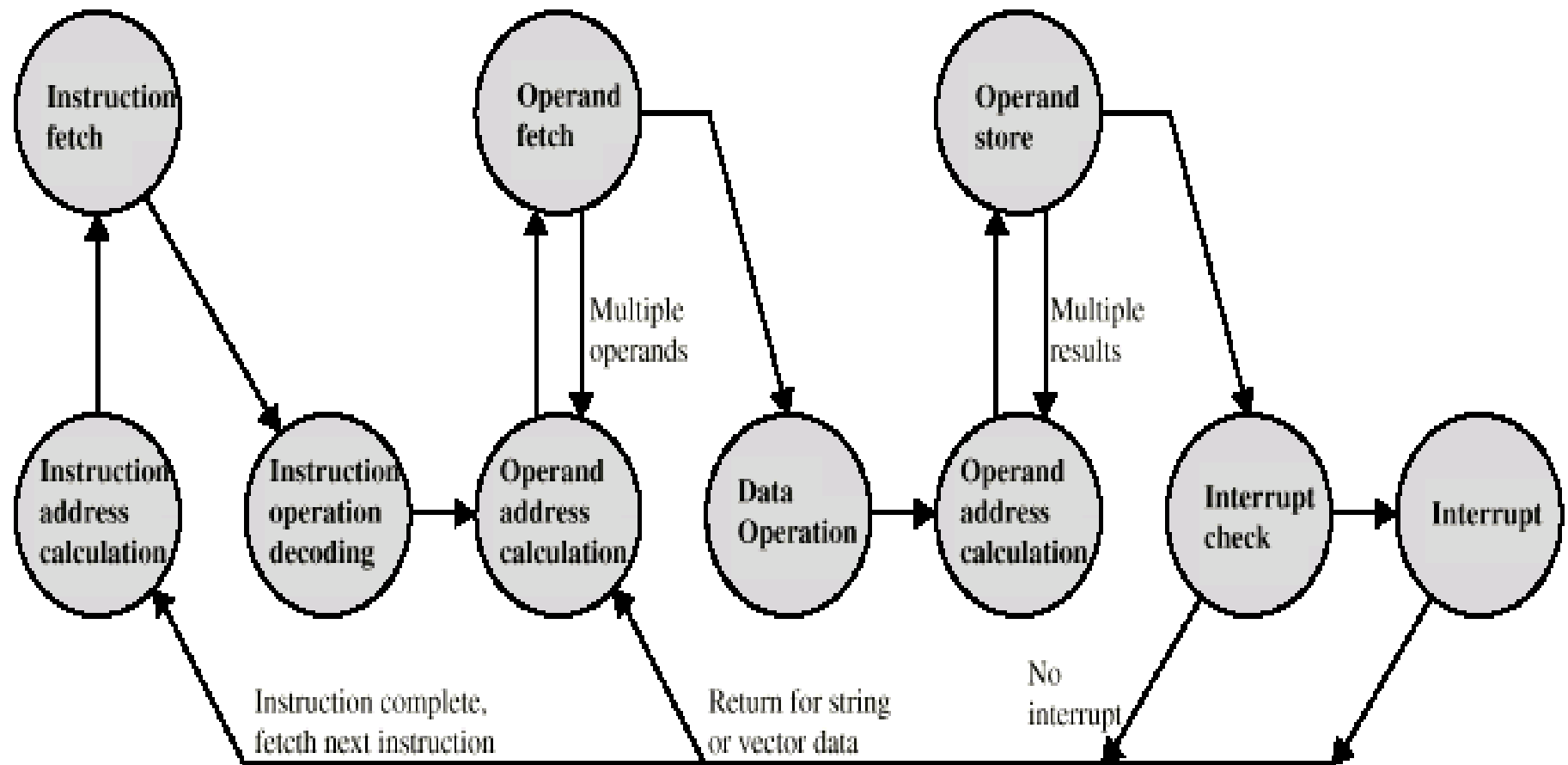


(-)



(-)

Ciclo de instrução com interrupção (AOC-Stallings)



CICLO DE INTERRUPÇÃO (AOC-Stallings)

- Adicionando ao ciclo de instrução
- Processador verifica se existe interrupção
 - Indicado pelo sinal de interrupção
- Se não existe interrupção, buscar a próxima instrução
- Caso existam interrupções pendentes:
 - Suspenda a execução do programa corrente
 - Salvar o contexto
 - Colocar no PC o endereço de início da rotina de tratamento da interrupção (RTI).
 - Processar a interrupção
 - Restaurar o contexto e continuar o programa interrompido

Tratadores de Interrupção (SOMTB2ed) - 1

- As interrupções devem ser escondidas o máximo possível e uma forma de fazer isso é bloqueando o driver que iniciou uma operação de E/S até que uma interrupção notifique que a E/S foi completada.
- Rotina de tratamento de interrupção cumpre sua tarefa e então desbloqueia o driver que a chamou.

Tratadores de Interrupção (SOMTB2ed) - 2

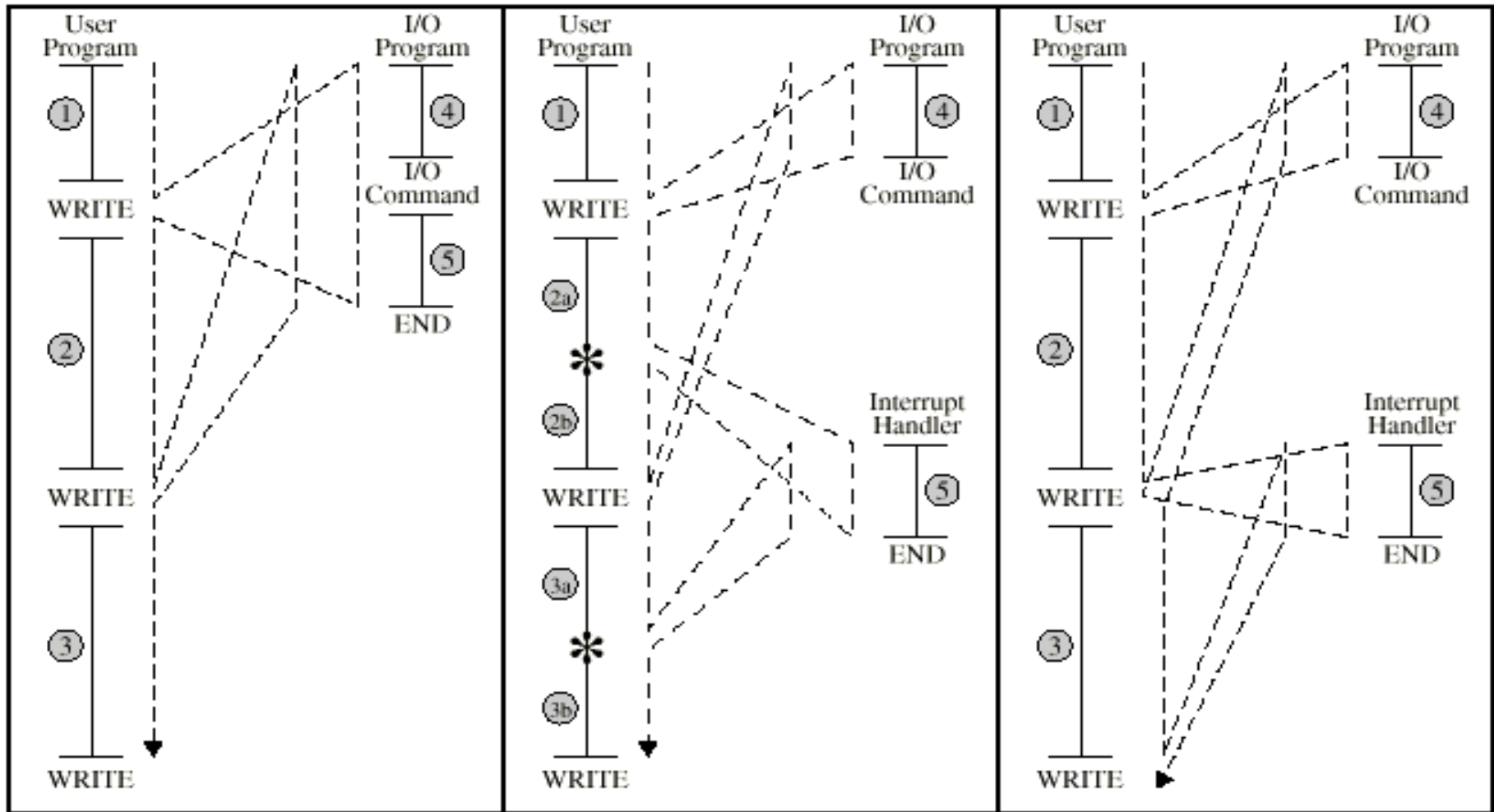
Passos que devem ser executados em software depois da interrupção ter sido concluída

2. salva registradores que ainda não foram salvos pelo hardware de interrupção
3. estabelece contexto para rotina de tratamento de interrupção
4. estabelece uma pilha para a rotina de tratamento de interrupção
5. sinaliza o controlador de interrupção, reabilita as interrupções
6. copia os registradores de onde eles foram salvos
7. executa rotina de tratamento de interrupção
8. escolhe o próximo processo a executar
9. estabelece o contexto da MMU para o próximo processo a executar
10. carrega os registradores do novo processo
11. começa a executar o novo processo

VETORES DE INTERRUPÇÃO

- O **Vetor de interrupção** é um mecanismo usado para **indicar o dispositivo** está solicitando operação de E/S. Eles servem para indicar uma **entrada** na tabela de vetores de interrupção.
- A **Tabela de vetores de interrupção** armazena os endereços das rotinas de tratamento de interrupção (RTI) de cada dispositivo de E/S, de exceções e das interrupções de software.

Controle de fluxo de programa (AOC-Stallings)



(a) No interrupts

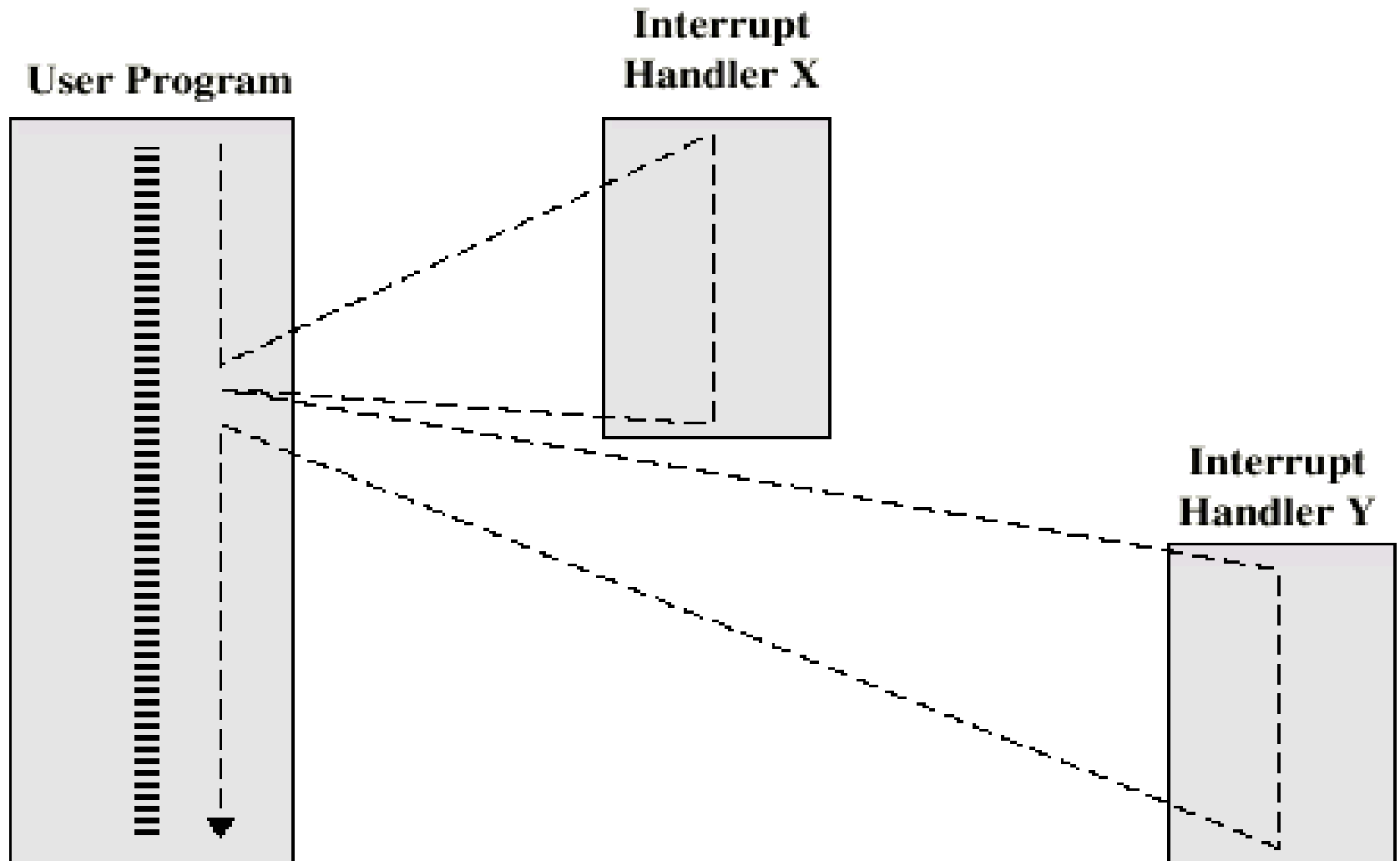
(b) Interrupts; short I/O wait

(c) Interrupts; long I/O wait

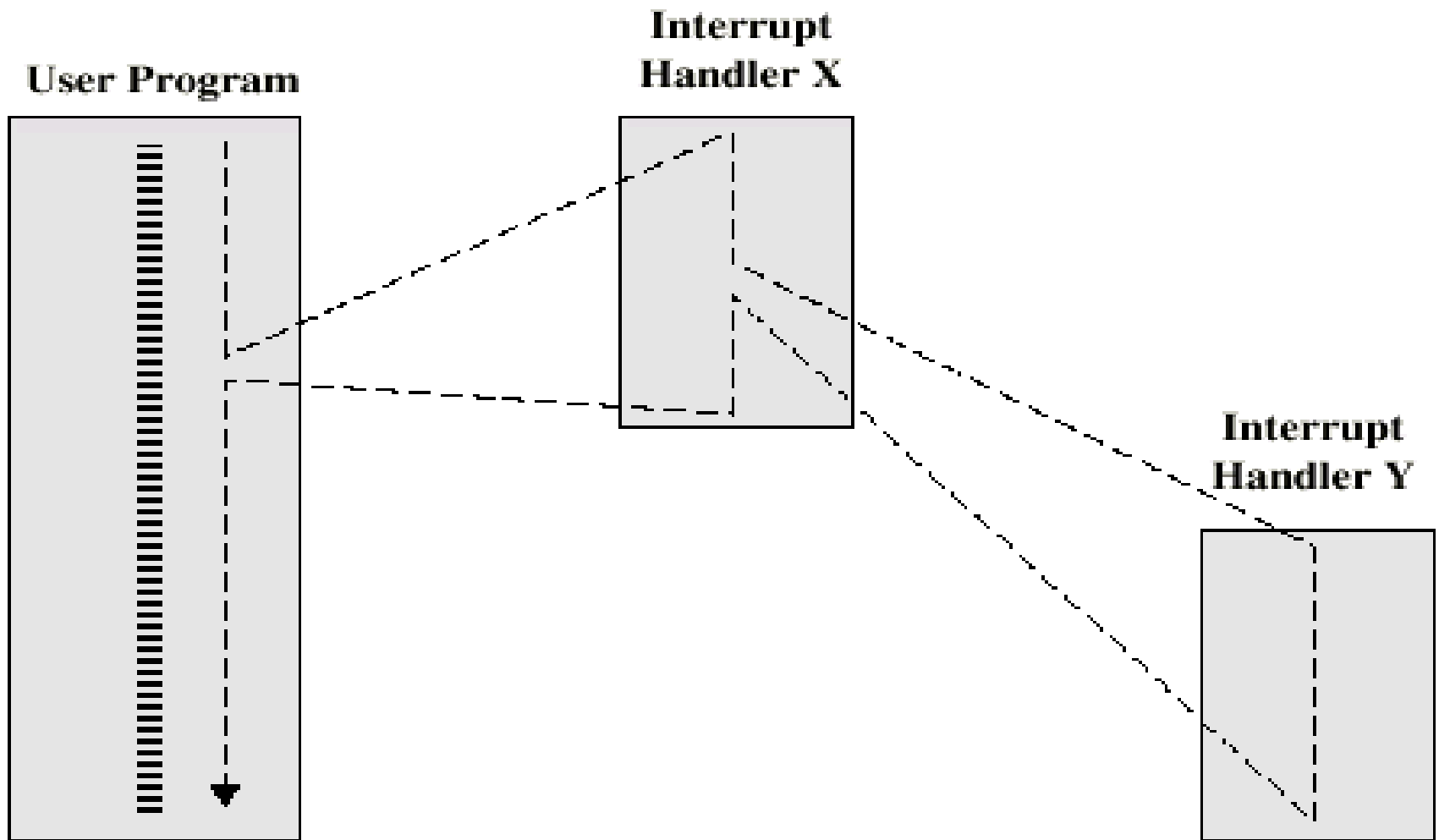
INTERRUPÇÕES MÚLTIPLAS (AOC-Stallings)

- Interrupções desabilitadas
 - O processador ignorará as interrupções enquanto processa uma interrupção
 - As interrupções permanecerão pendentes e serão verificadas após a primeira interrupção ter sido processada.
 - Manipulação das interrupções na seqüências que elas ocorrem
- Definir prioridades
 - Interrupções de baixa prioridade podem ser interrompidas por interrupções de maior prioridade
 - Quando a interrupção de maior prioridade foi processada, o processador retorna para a interrupção previa

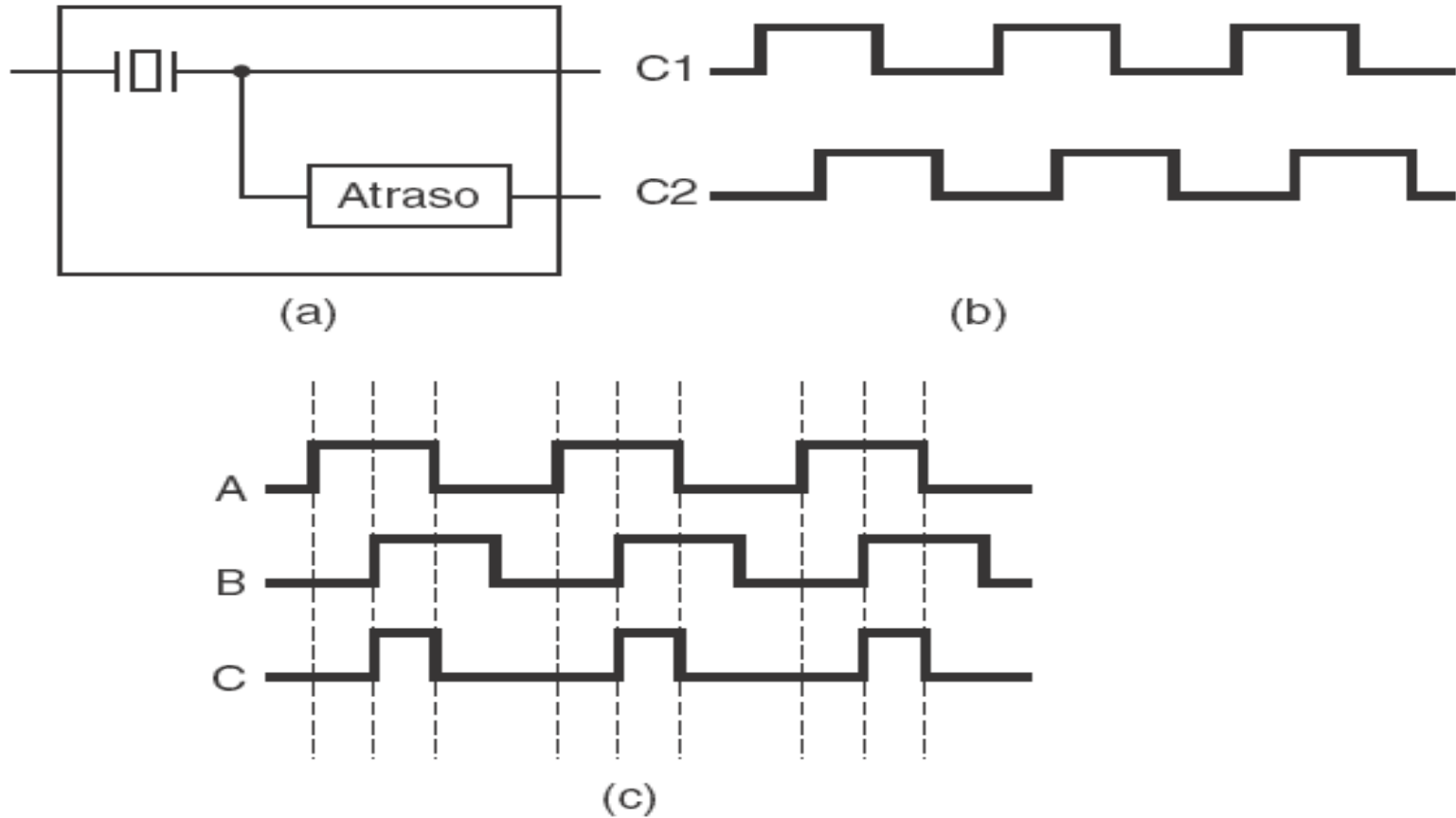
Interrupções Múltiplas: Seqüencial (AOC-Stallings)



Interrupções Múltiplas - Aninhadas (AOC-Stallings)



Relógios (OECTB5ed)

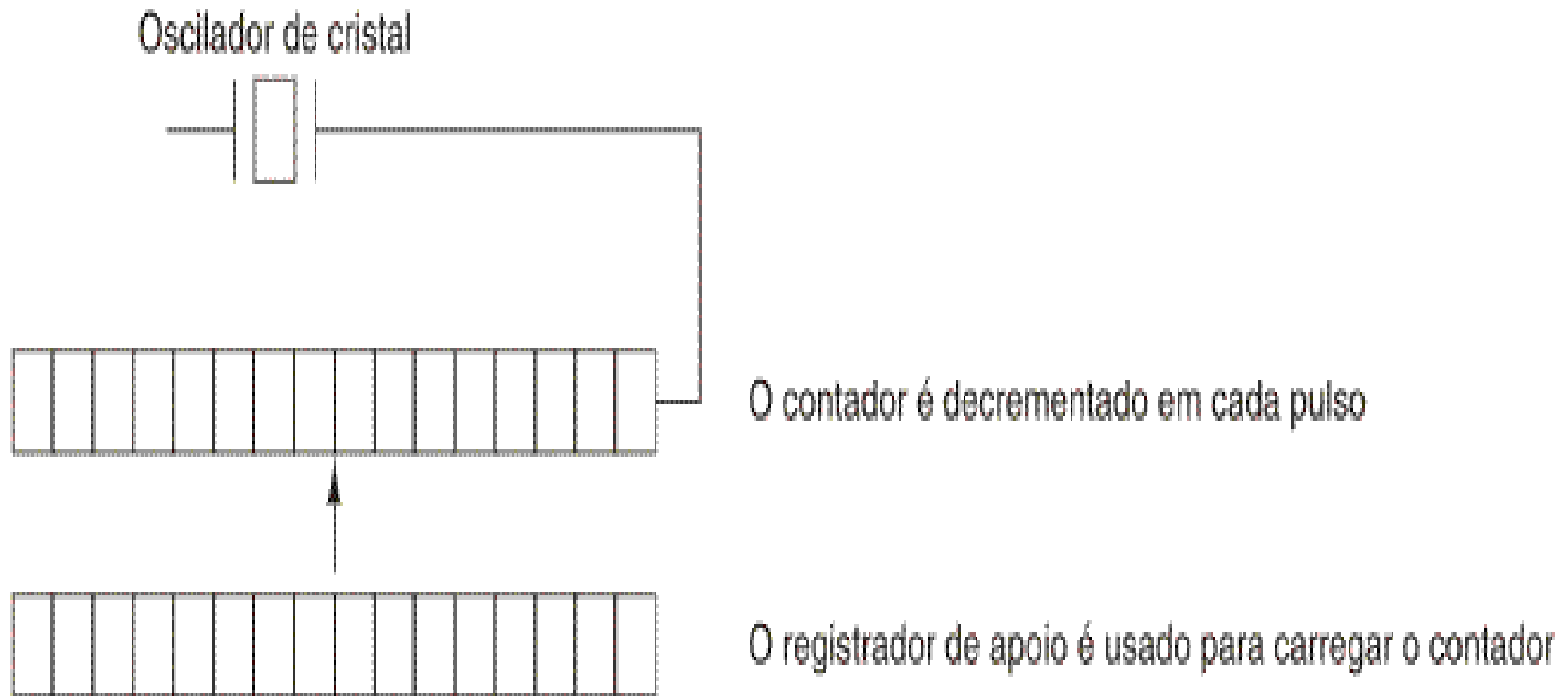


(a) Relógio.

(b) Diagrama de temporização para o relógio.

(c) Geração de um relógio assimétrico.

Hardware do Relógio: programável (SOM-TB2ed)



Temporizadores e relógios (SODeitel3ed)

- **Temporizadores**

- O temporizador de intervalo gera periodicamente uma interrupção.
- Os sistemas operacionais usam temporizadores de intervalo para impedir que processos monopolizem o processador.

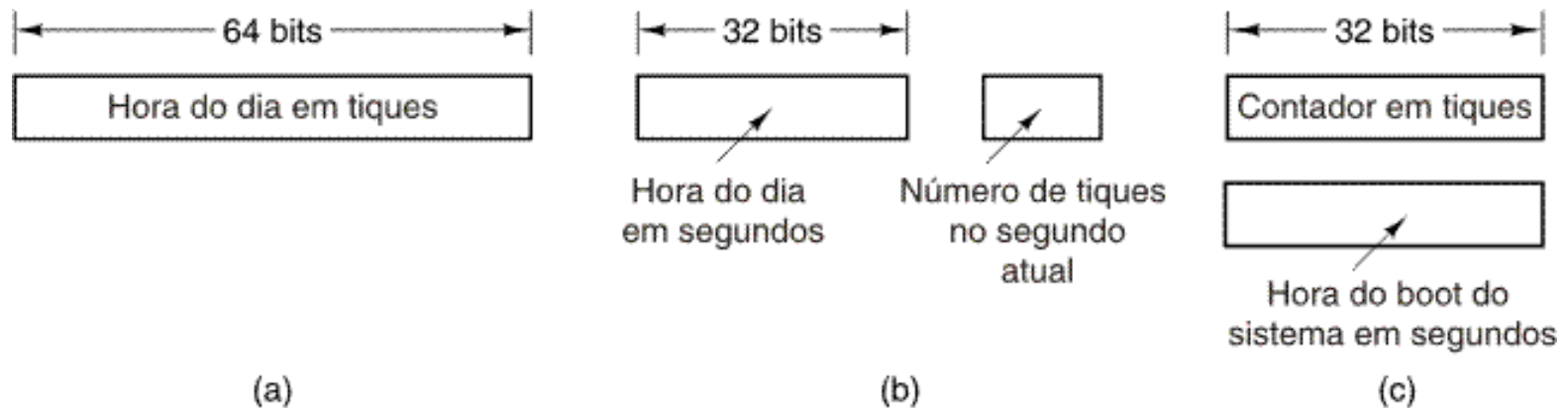
- **Relógios**

- Oferecem uma medida de continuidade.
- Um relógio de 24 horas habilita o sistema operacional a determinar a hora e a data atuais.

Temporizadores de Software

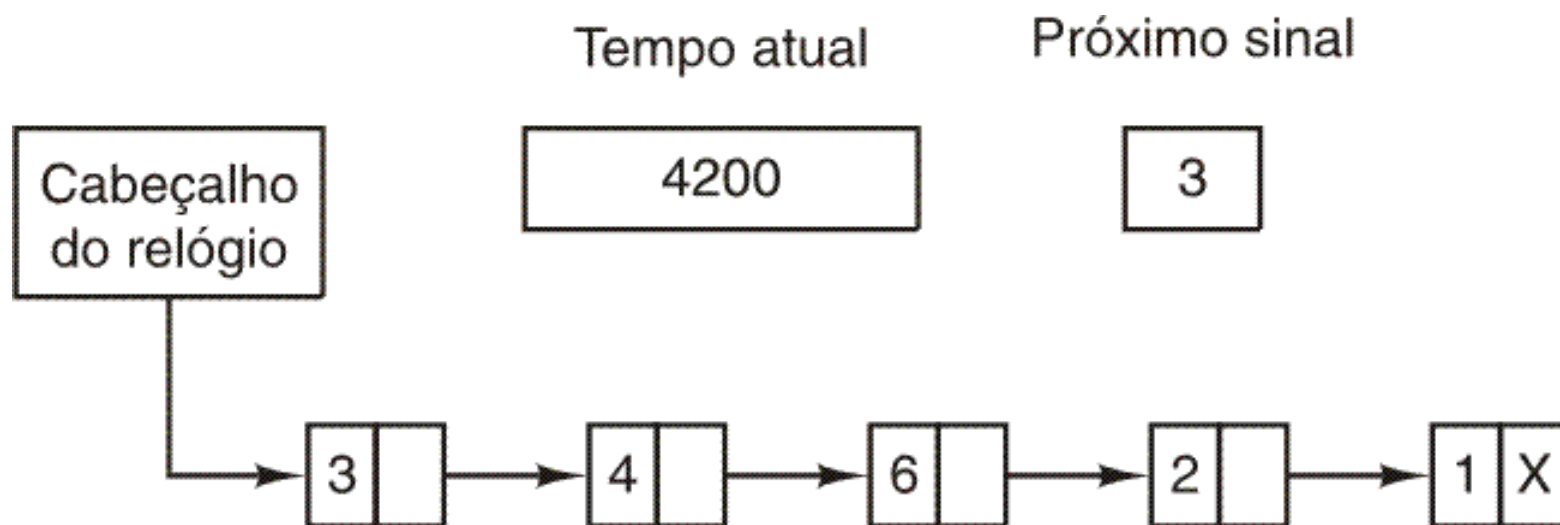
- Um segundo relógio programável para interrupções de temporização
 - ajustado para causar interrupções em qualquer taxa que um programa precisar
 - sem problemas se a frequência de interrupção é baixa
- Temporizadores de software evitam interrupções
 - núcleo checa se o temporizador de software expirou antes de retornar para o modo usuário
 - quão bem isso funciona depende da taxa de entradas no núcleo

Software do relógio (SOM-TB2ed) – 1



Três maneiras para manter a hora do dia

Software do relógio (SOM-TB2ed) – 2



Simulação de vários temporizadores com um único relógio

INTERRUPÇÕES DE SOFTWARE

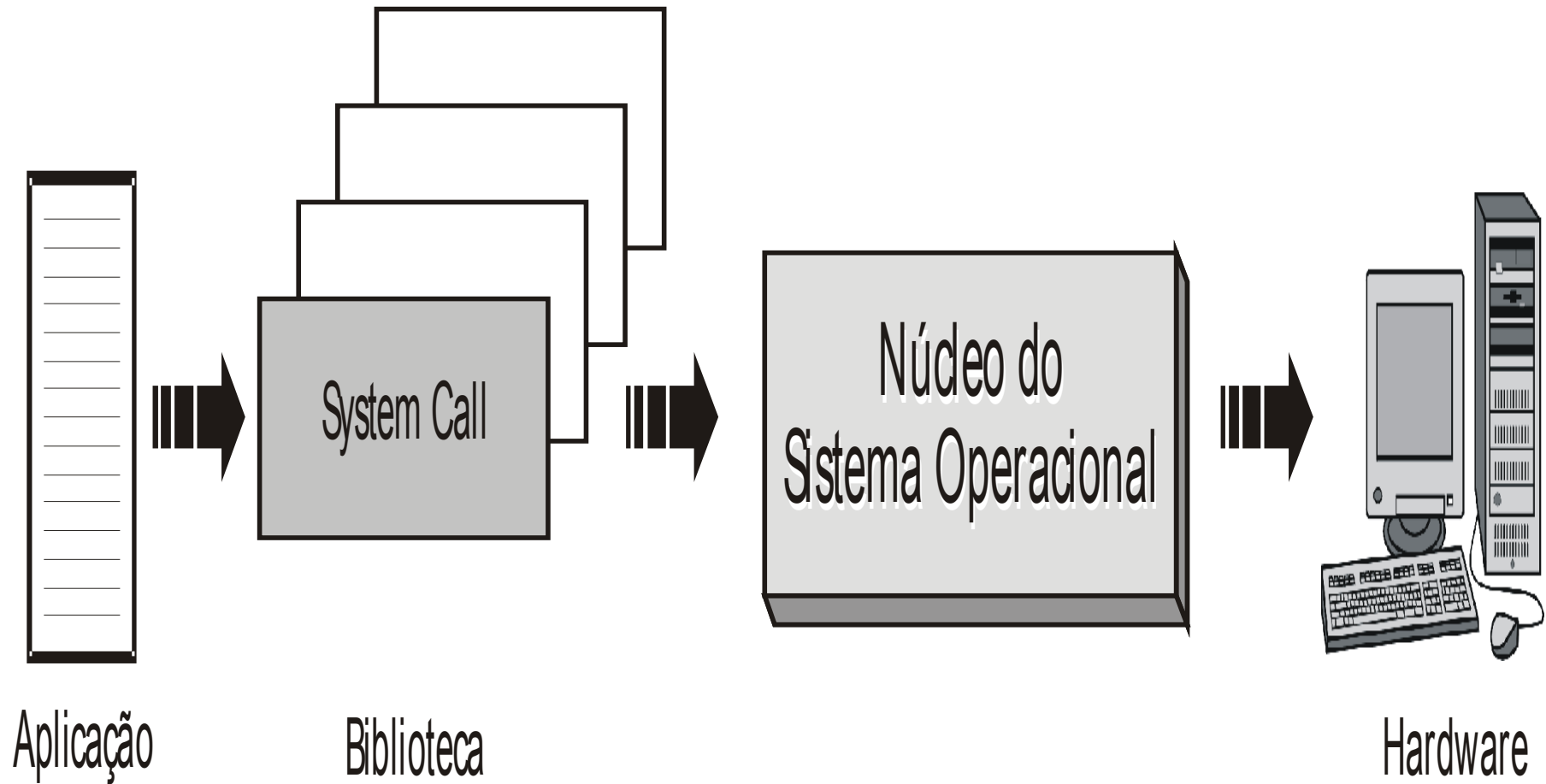
- Na arquitetura x86 (ou IA-32 da Intel) usa o termo interrupção de software para designar **instruções especiais** que fazem parte do conjunto de instruções de um processador, mas que tem um tratamento semelhante às interrupções de hardware.
- Os serviços do Sistema Operacional (SO) são geralmente obtidos através deste mecanismo (chamadas ou “traps” ao SO).

Exemplos:

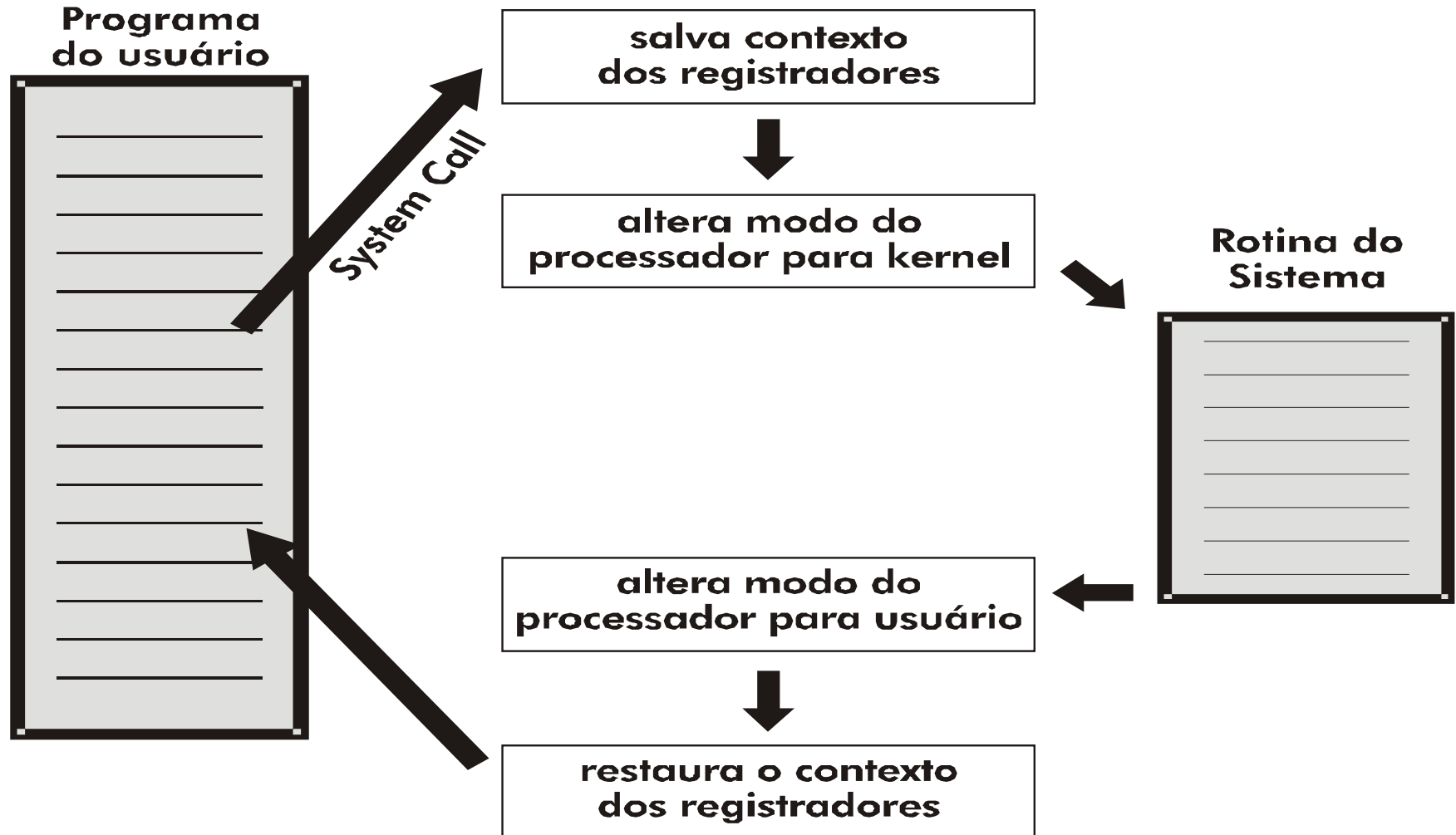
INT 0x80 (Linux)

INT 0x21 (DOS)

CHAMADAS AO SO (INT SW)



INTERRUPÇÃO DE SOFTWARE



Exceções e interrupções no MIPS (exemplos)

TIPO DE EVENTO	ORIGEM	TERMINOLOGIA MIPS
Solicitação de dispositivo de E/S	externa	interrupção
Chamada do SO pelo programa de usuário	interna	exceção
Overflow aritmético	interna	exceção
Uso de uma instrução indefinida	interna	exceção
Mal funcionamento do hardware	ambas	exceção ou interrupção

Como manipular exceções e interrupções (MIPS)

- *Ações básicas a serem tomadas pela máquina quando ocorre uma exceção :*
 - 1) *salvar o endereço da instrução ofendida (instrução atual) num registrador especial EPC (Exception Program Counter).*
 - 2) *transferir o controle (atualizar o conteúdo de PC) para um endereço específico, que possibilite ao computador executar a rotina de tratamento de exceção.*
- *ROTINA DE TRATAMENTO DE EXCEÇÃO: depende do tipo de exceção, por exemplo, reportar ao usuário que ocorreu um erro de overflow.*
- *Ações a serem tomadas após a execução do tratamento de exceção:*
 - *Depende do tipo de exceção: por ex. pode encerrar, ou continuar executando, o programa que estava executando antes da exceção.*
 - *O EPC é usado para determinar o endereço de reinício da execução.*

Como saber a causa da exceção (MIPS)

- Para o computador manipular uma exceção, ele deve conhecer a razão da exceção.
- A razão pode ser obtida de duas formas:
 - 1) usar o registrador de status (cause register) que indica a razão
 - 2) usar interrupção vetorizada, caso em que o endereço de desvio para executar a exceção é determinado pela causa da exceção

Implementação de exceção no MIPS

- Implementação de duas exceções:
 - 1) overflow aritmético
 - 2) instrução indefinida

A causa é determinada pelo valor do registrador CAUSE de 32 bits:

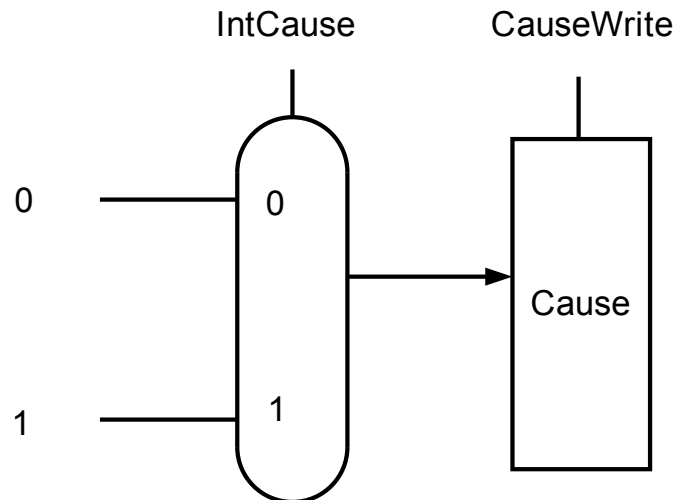
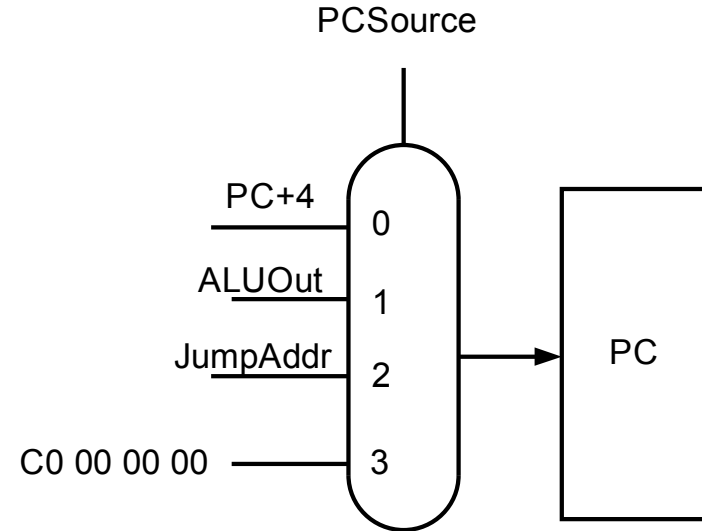
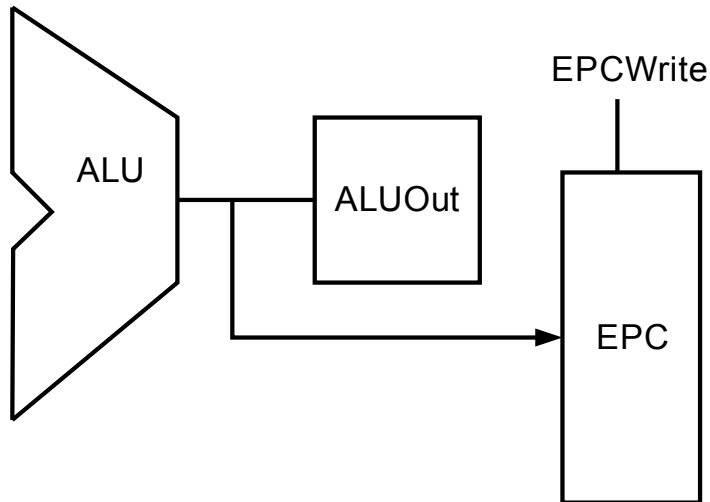
CAUSE = 00 00 00 00 (em hexa), overflow aritmético

CAUSE = 00 00 00 01 (em hexa), instrução indefinida

O endereço da instrução ofendida é guardada em EPC

O endereço de desvio para o início da execução da rotina de tratamento de exceção é dado por C0 00 00 00 (em hexa).

Alterações no fluxo de dados (MIPS)



Registadores adicionais:
EPC, Cause

Sinais de controle adicionais:
EPCWrite, CauseWrite,
IntCause

E/S Orientada à Interrupção (SOMTB2ed)

```
copy_from_user(buffer, p, count);  
enable_interrupts( );  
while (*printer_status_reg != READY) ;  
*printer_data_register = p[0];  
scheduler( );
```

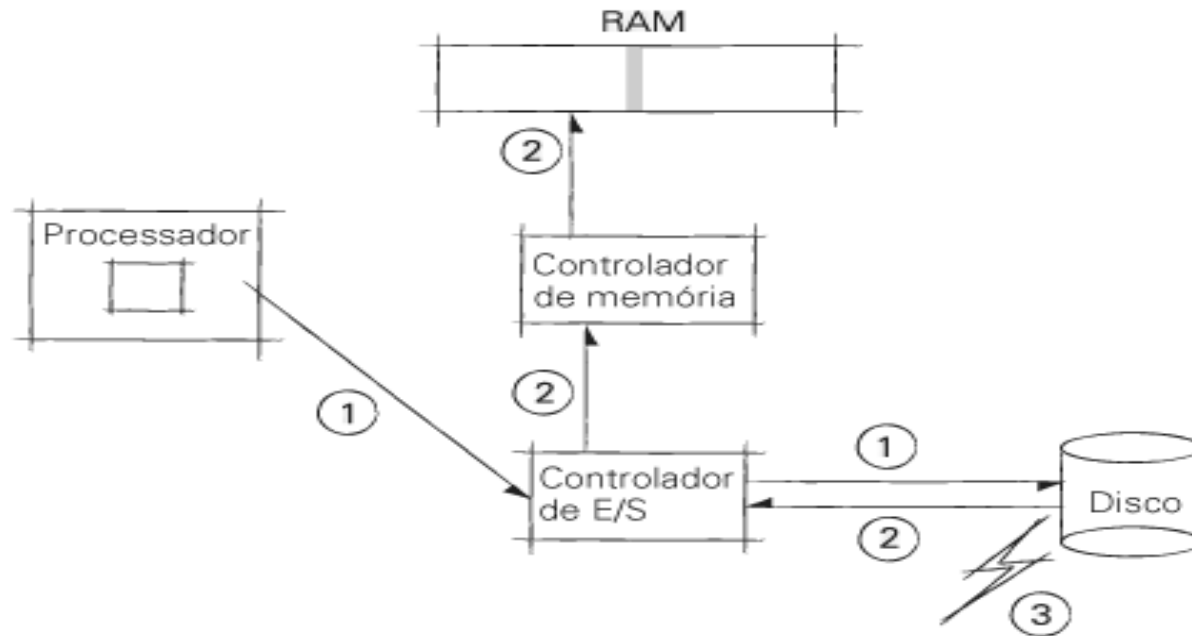
(a)

```
if (count == 0) {  
    unblock_user( );  
} else {  
    *printer_data_register = p[i];  
    count = count - 1;  
    i = i + 1;  
}  
acknowledge_interrupt( );  
return_from_interrupt( );
```

(b)

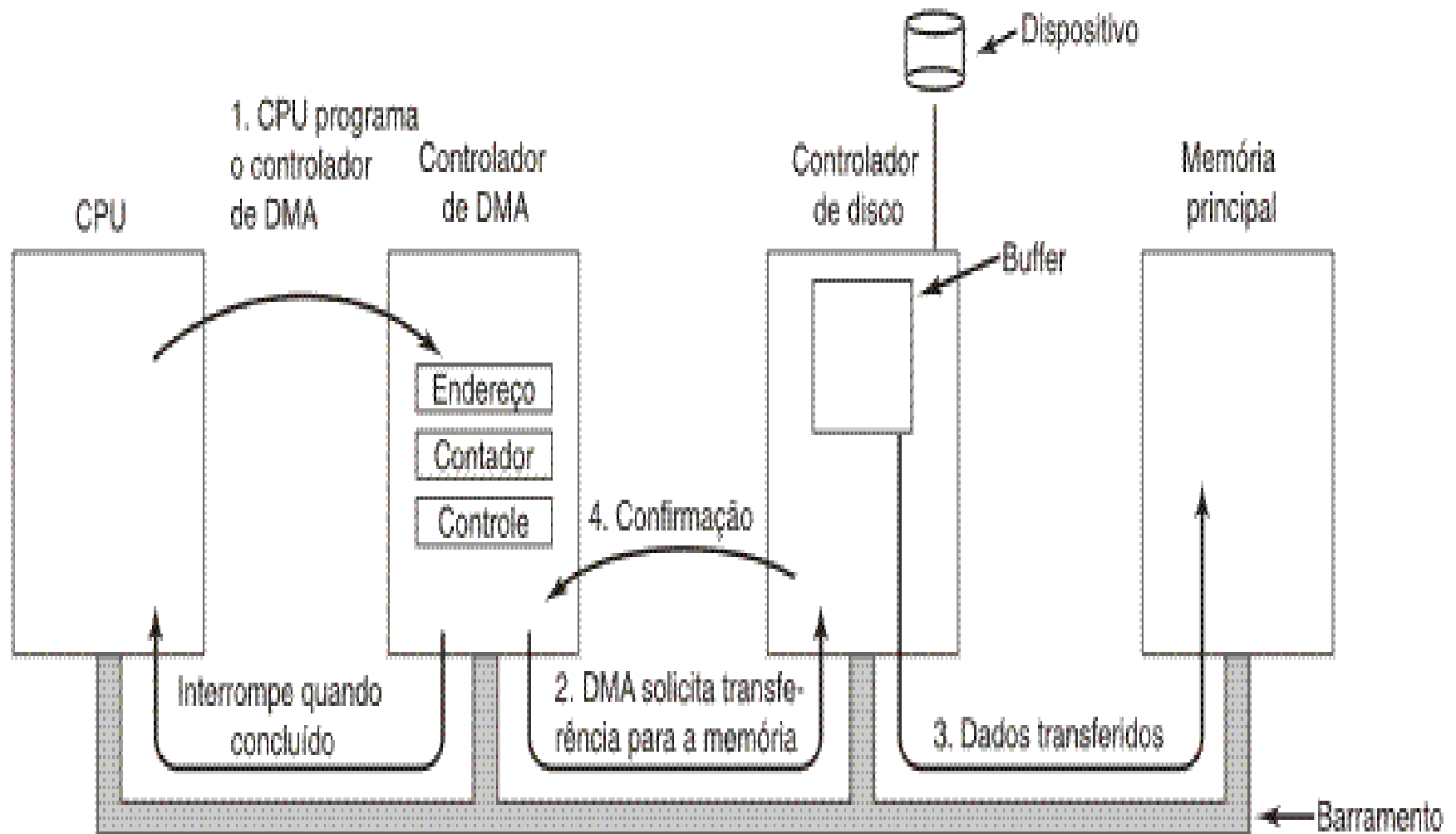
- Escrita de uma cadeia de caracteres para a impressora usando E/S orientada à interrupção
- Código executado quando é feita a chamada ao sistema para impressão
- Rotina de tratamento de interrupção

DMA: Acesso Direto à Memória (Fig 2.4 SODeitel3ed)

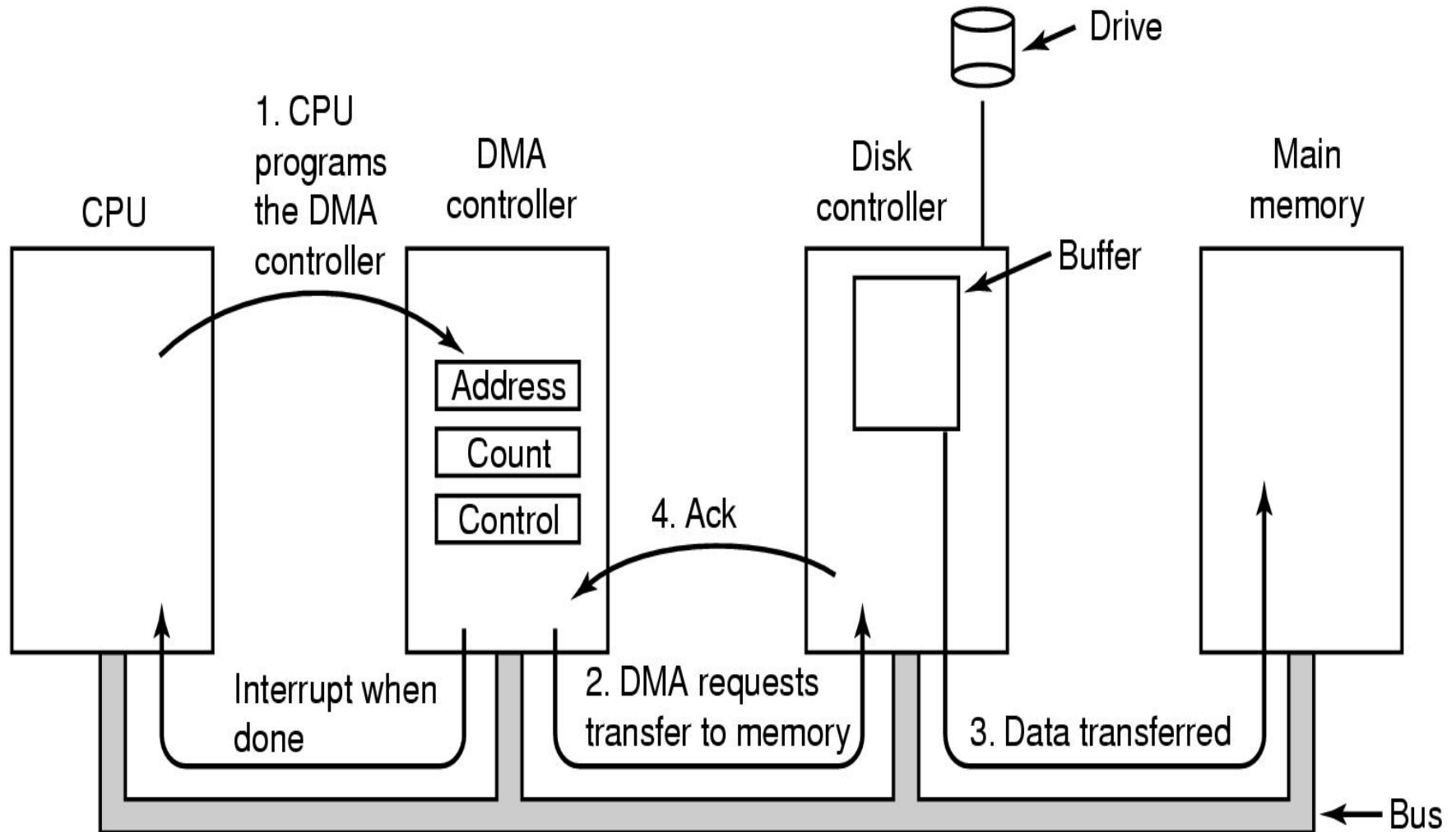


- ① Um processador envia uma solicitação de E/S ao controlador de E/S, que envia a solicitação ao disco. O processador continua executando instruções.
- ② O disco envia dados ao controlador de E/S; os dados são colocados no endereço de memória especificado pelo comando do DMA.
- ③ O disco envia uma interrupção ao processador indicando que a E/S foi concluída.

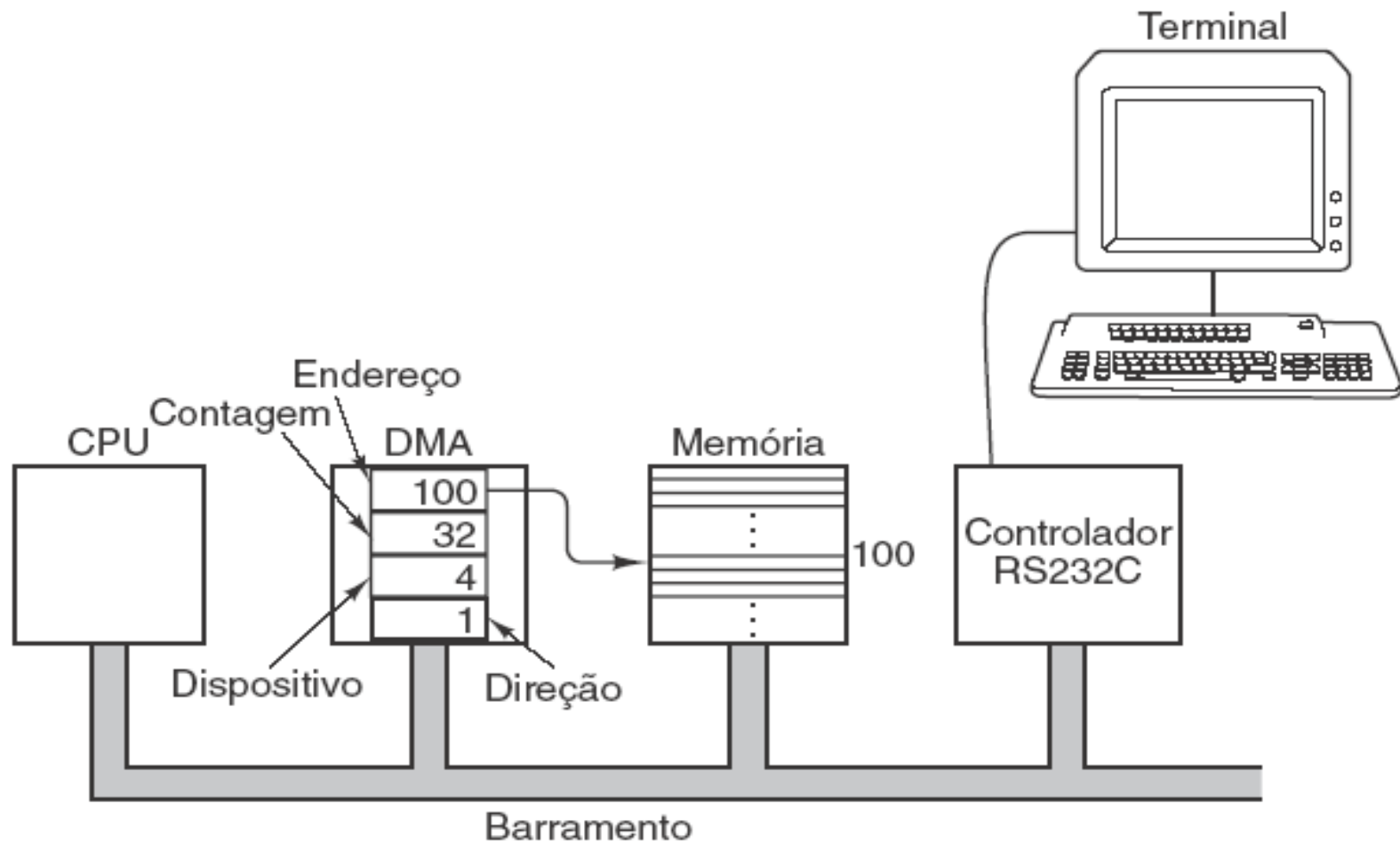
DMA: operação de uma transferência (SOMTB2ed)



Acesso Direto à Memória (DMA)



Sistema com controlador de DMA (OECTB5ed)



DMA : Acesso direto à memória (SODeitel3ed)

- Melhora a transferência de dados entre a memória e os dispositivos de E/S.
- Dispositivos e controladores transferem dados para e da memória principal diretamente.
- O processador fica livre para executar instruções.
- O canal DMA usa um controlador de E/S para gerenciar a transferência de dados.
- Notifica o processador quando uma operação de E/S é concluída.
- Melhora o desempenho em sistemas que realizam grande quantidade de operações de E/S (por exemplo, computadores de grande porte e servidores).

E/S Usando DMA (SOMTB2ed)

```
copy_from_user(buffer, p, count);  
set_up_DMA_controller();  
scheduler();
```

(a)

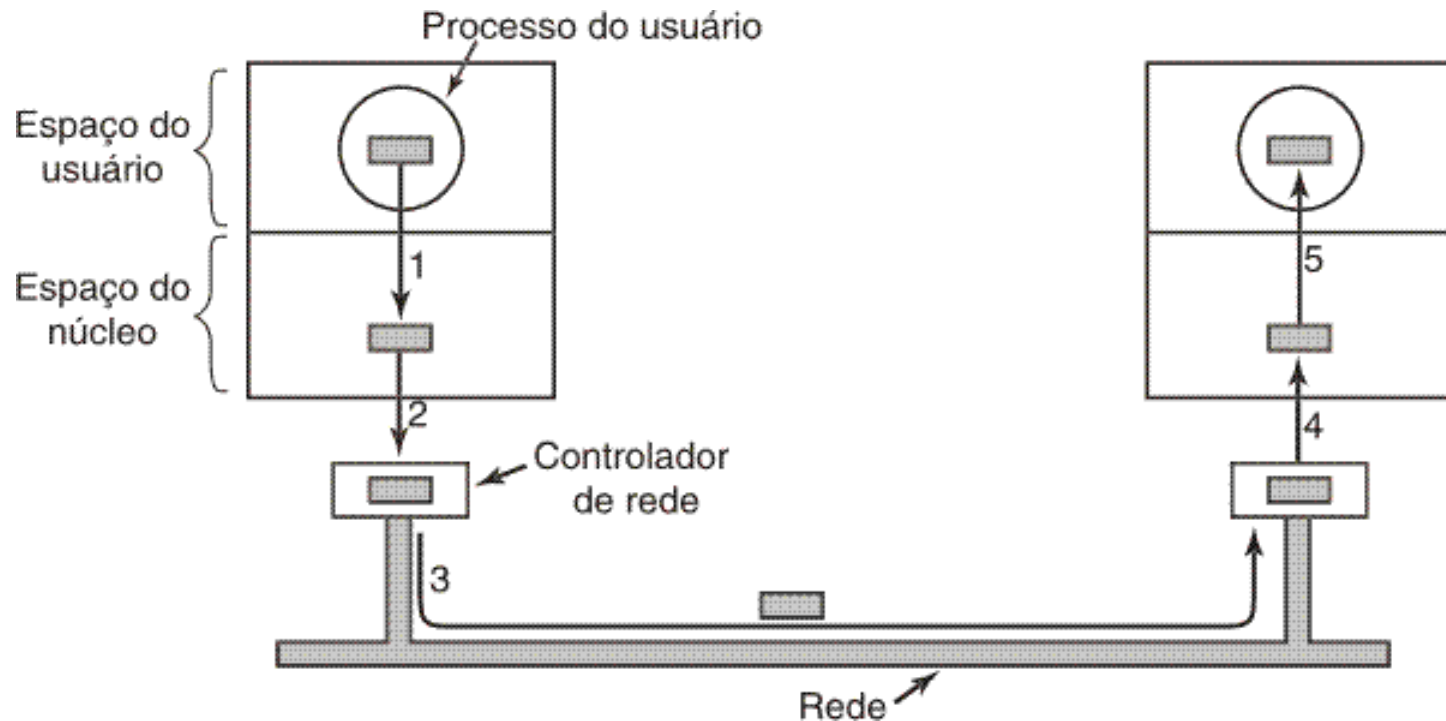
```
acknowledge_interrupt();  
unblock_user();  
return_from_interrupt();
```

(b)

Impressão de uma cadeia de caracteres usando DMA

- a) Código executado quando é feita a chamada ao sistema para impressão
- b) Rotina de tratamento de interrupção

Software de E/S Independente de Dispositivo (SOMTB2ed)

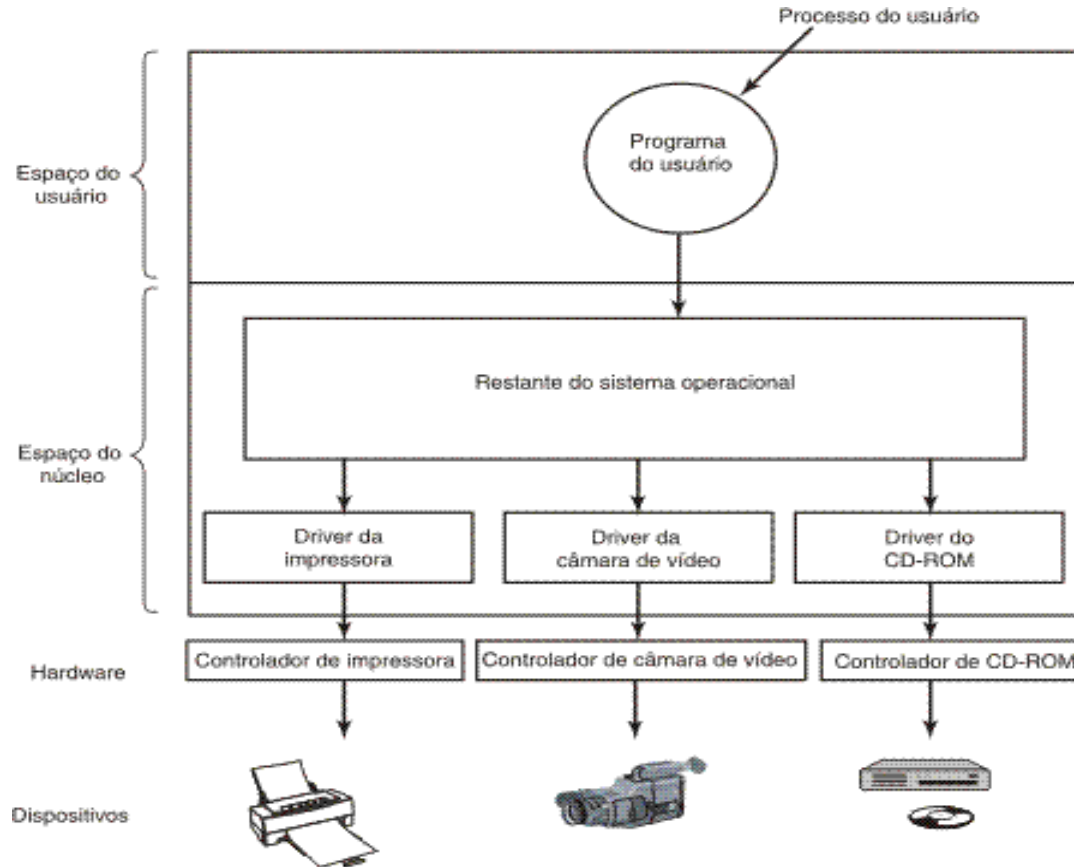


A operação em rede pode envolver muitas cópias de um pacote

DD (“Device Driver”)

- Programa que se comunica com o controlador de dispositivo de E/S.
- Realiza a comunicação do subsistema de E/S com os dispositivos através dos controladores de E/S.

Drivers dos Dispositivos (SOMTB2ed)



- Posição lógica dos drivers dos dispositivos
- A comunicação entre os drivers e os controladores de dispositivos é feita por meio do barramento

Plug-and-play (PnP) (SODeitel3ed)

- Os dispositivos de hardware são adicionados dinamicamente.
- Os recursos são redistribuídos (portas de E/S, canais de DMA).
- **Suporte ao hardware**
 - Reconhece novos componentes;
 - Notifica o sistema.
- **Suporte ao software**
 - Gerenciador de PnP do Windows XP
 - Modo núcleo: configura dispositivos e aloca recursos.
 - Modo usuário: interage com programas de instalação, notifica processos de eventos de modo usuário.
 - Requisições de E/S PnP: enviadas para os drivers de dispositivo.

Gerenciador de energia (SODEitel3ed)

- Administra a política de energia.
- Conservação *versus* desempenho.
- Estados de energia do dispositivo: D0, D1, D2 e D3.
- D0 está totalmente energizado.
- D3 está desligado.
- Estados de energia do sistema: S0, S1, S2, S3, S4, S5.
- S0 está totalmente energizado.
- S5 está desligado.