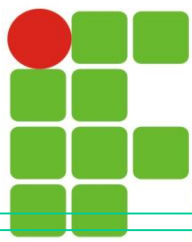


**INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**  
RIO GRANDE DO NORTE  
Campus Santa Cruz

# Aula 10

## Técnicas de Entrada e Saída

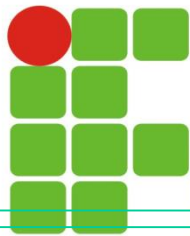


# Técnicas de Entrada e Saída

Programada – Sem interrupções

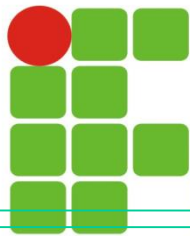
Por interrupção

Acesso Direto à Memória (ADM)



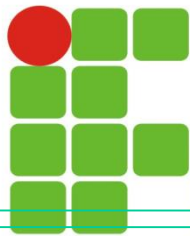
# E/S Programada

- ✓ O processador executa um programa que lhe oferece o controle direto da operação de E/S.
- ✓ Percepção do estado do dispositivo
- ✓ Envio de um comando de leitura ou escrita
- ✓ Transferência de dados
- ✓ Quando o processador emite um comando de E/S, ele precisa esperar até que a operação termine.
- ✓ Perda de tempo



# E/S Programada

- ✓ Em um programa, a execução de uma instrução relacionada a E/S faz com que um comando seja enviado para o módulo de E/S.
- ✓ O módulo de E/S executa a operação requisitada e sinaliza o seu término carregando um valor no registrador de estado.
- ✓ Nenhuma ação é executada pelo módulo para alertar o processador sobre o término da operação.
- ✓ É responsabilidade do processador verificar periodicamente o estado do módulo, para ver se a operação foi completada.



# E/S Programada

- ✓ CPU controla direto o módulo de E/S
  - ✓ Verifica estado frequentemente
  - ✓ Informa comandos de escrita/leitura
  - ✓ Transfere dados
- ✓ CPU espera pela finalização da operação do módulo de E/S
- ✓ Gasta tempo de processamento da CPU

UCP solicita operação de E/S

Módulo de E/S realiza a operação

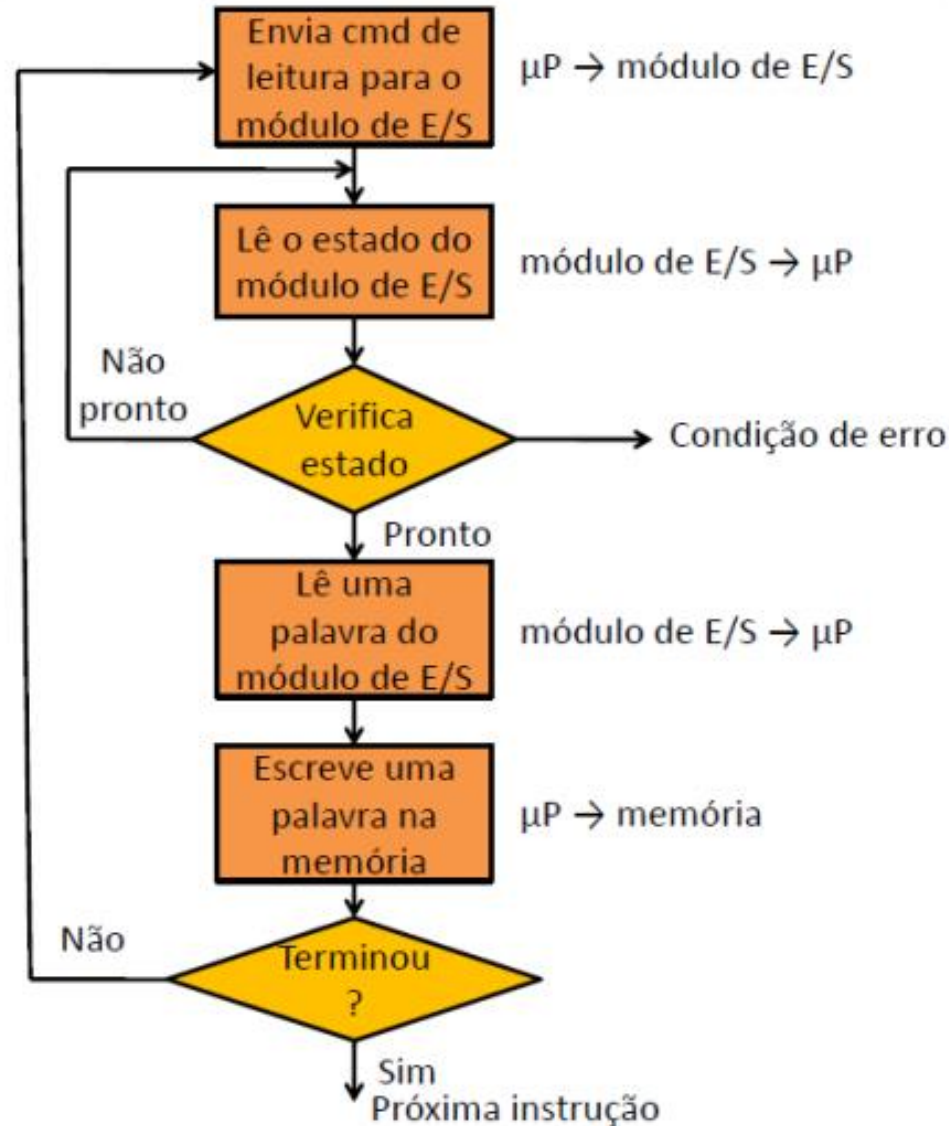
Módulo de E/S seta bits de estado

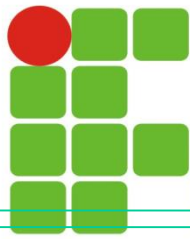
UCP verifica bits de status periodicamente

Módulo de E/S não informa diretamente à UCP

Módulo de E/S não interrompe a UCP

# Operação Básica para Realização de E/S Programada

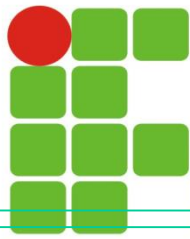




# E/S Dirigida por Interrupção

- O processador emite um comando de E/S, continua a executar outras instruções e é interrompido pelo módulo de E/S quando o último tiver completado seu trabalho.
- O processador é responsável por obter dados da memória principal para saída, e para armazenar dados na memória principal para a entrada.





# E/S Dirigida por Interrupção

- Libera espera de UCP
- UCP não precisa ficar verificando estado do dispositivo repetidamente
- Módulo de E/S interrompe a UCP quando estiver pronto, e a UCP executa a transferência de dados

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 guarda o estado do que estava fazendo)

UCP pede dados para o módulo de E/S

Módulo de E/S transfere dados para UCP



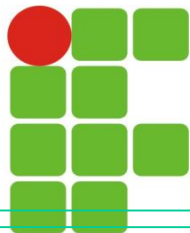
# Operação Básica para Realização de E/S por Interrupção

Processador envia um comando de E/S para o módulo e continua a executar outras instruções.

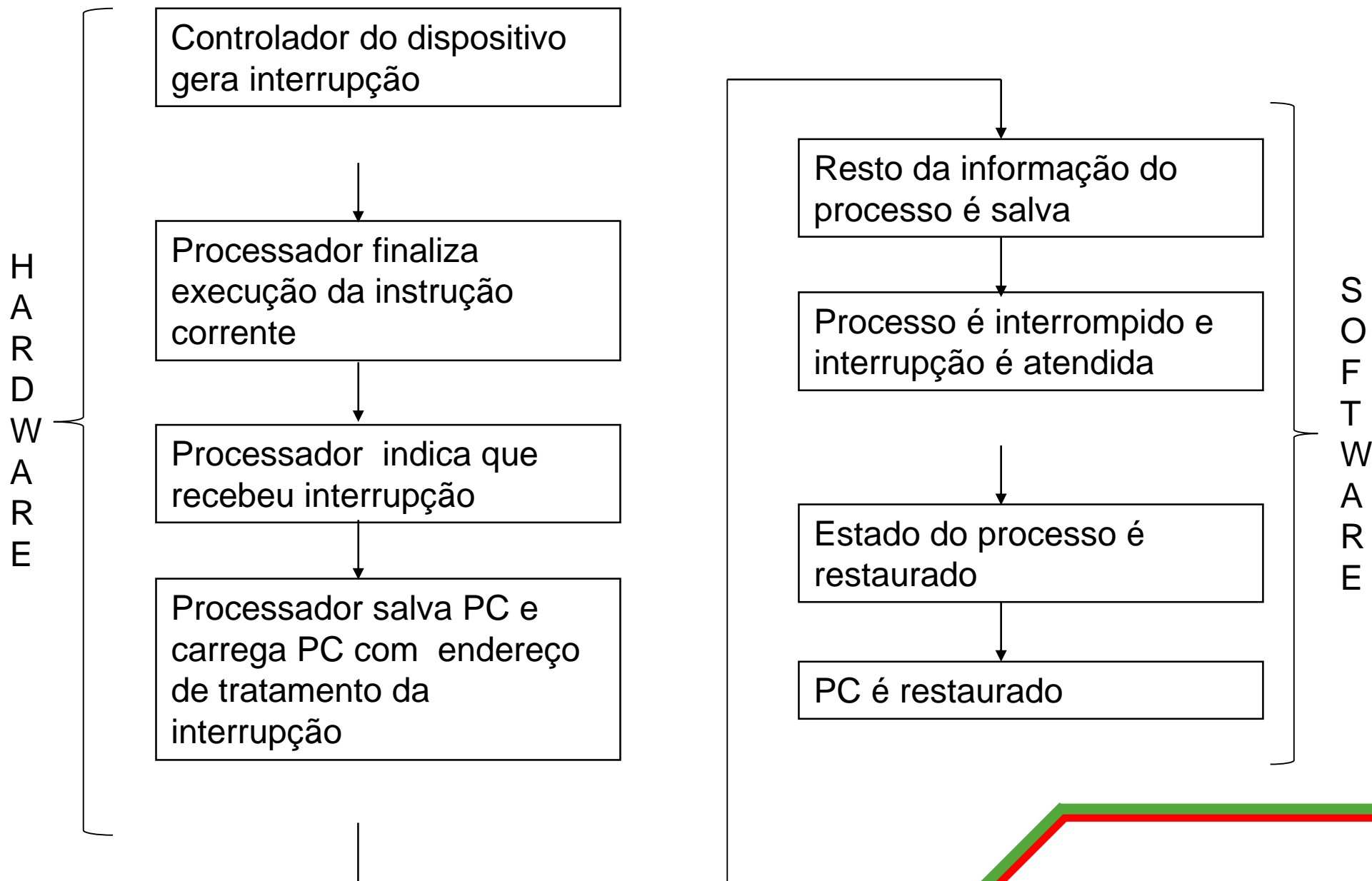
O módulo de E/S interrompe o processador quando o mesmo estiver pronto para trocar dados.

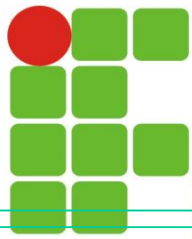
Processador efetua a transferência de dados e depois retorna ao seu processamento original.

É mais eficiente que a E/S programada, pois elimina ciclos de espera desnecessários.

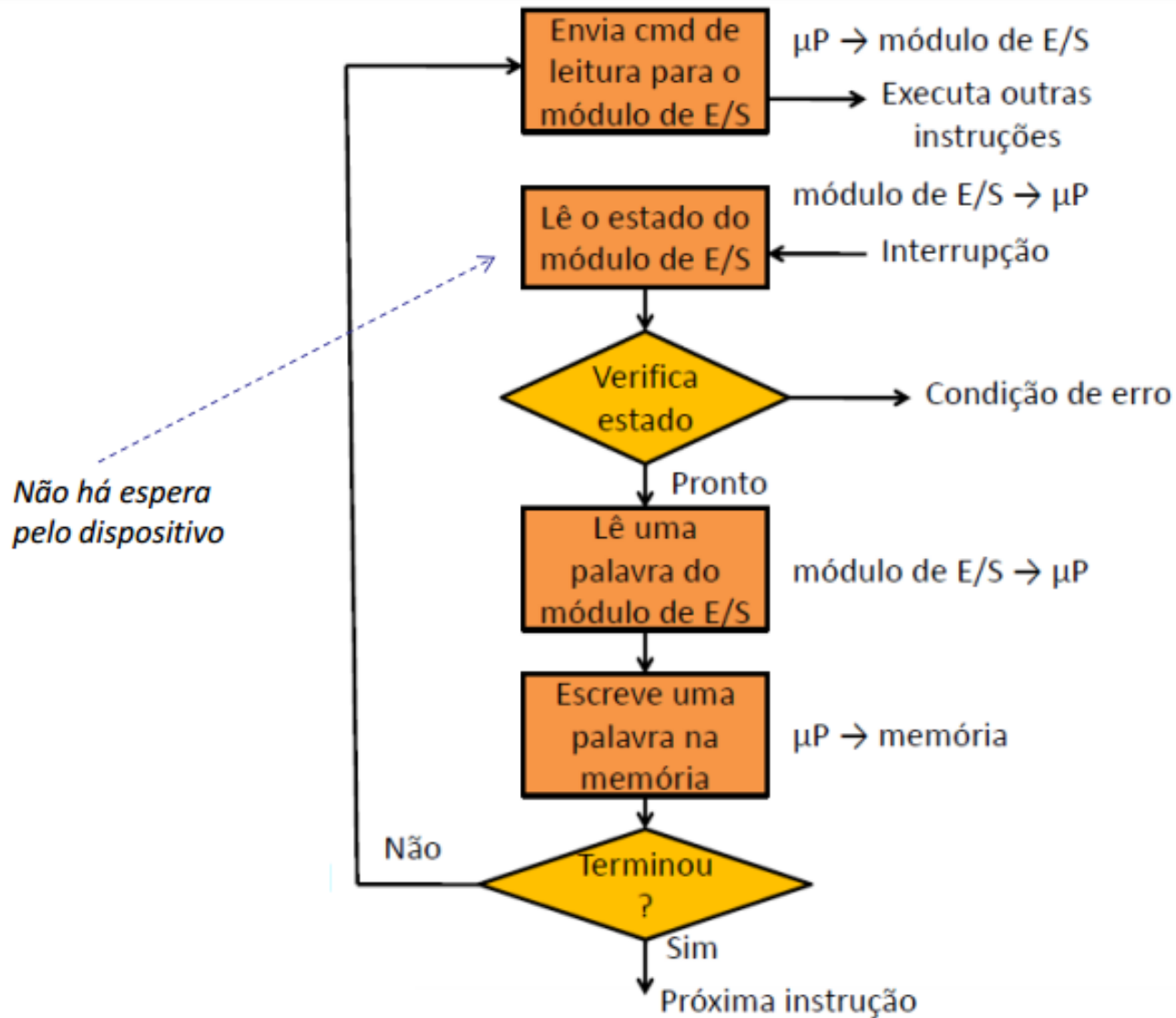


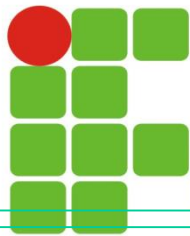
# Processamento da Interrupção





# Processamento da Interrupção





# Atividades da CPU

Envia comando de leitura

Executa outra tarefa

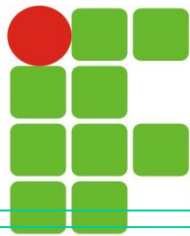
Verifica se existe interrupção ao final de cada instrução

Caso exista interrupção:

- Salva contexto (registradores)

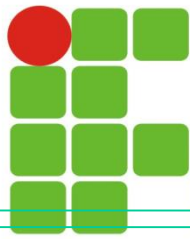
- Interrompe processo

- Obtém dados do módulo de E/S e os armazena



# Comando de E/S

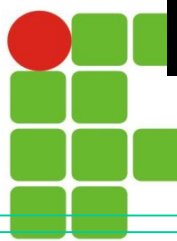
- ✓ Para executar uma instrução relacionada a E/S, o processador emite um endereço, **especifica o módulo de E/S**, e um comando de E/S.
- ✓ Controle – Ativar o periférico e dizer o que fazer
- ✓ Teste – testa as diversas condições de estado associado ao dispositivo
- ✓ Leitura – obter o item e colocar no buffer – pode colocar no barramento
- ✓ Escrita – apanha um item de dado do barramento para o periférico



# Comando de E/S

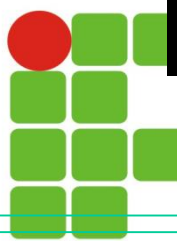
- ✓ Dois modos de endereçamento são possíveis
  - ✓ E/S mapeada na memória
    - ✓ Um único espaço de endereço para locais de memória e dispositivos E/S
    - ✓ As mesmas instruções para acessar a memória e os dispositivos de E/S
  - ✓ E/S independente
    - ✓ Espaços de endereços para E/S independente do espaço de memória.
- ✓





# Identificação do módulo que gerou a Interrupção

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



# Identificação do módulo que gerou a Interrupção

## Arbitragem do barramento

Módulo precisa obter o controle do barramento e depois enviar sinal de interrupção

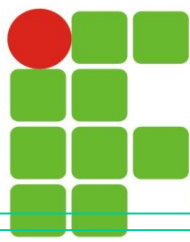
CPU envia sinal de reconhecimento e módulo coloca o vetor de interrupção nas linhas de dados

## Múltiplas linhas de interrupção

Com mais de uma linha de interrupção, cada linha de interrupção possui uma prioridade

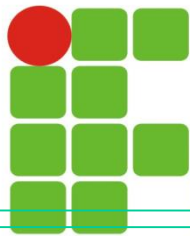
Linhas com prioridade maior podem interromper linhas com prioridade menor

Esquema de prioridades para arbitração de barramento



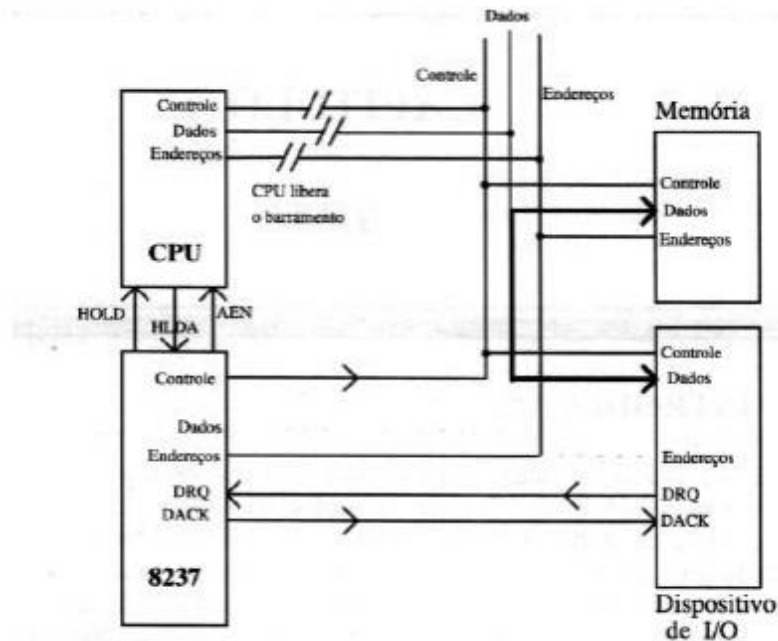
# Acesso Direto à Memória

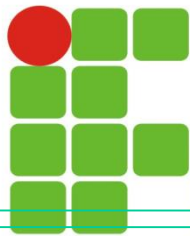
- ✓ E/S programada e por interrupção requerem intervenção ativa da UCP
- ✓ Taxa de transferência é limitada pela capacidade de atendimento da CPU
- ✓ CPU fica ocupada gerenciando a transferência de dados
- ✓ DMA (acesso direto a memória) pode ser uma técnica mais eficiente
  - ✓ Módulo adicional de hardware no barramento
  - ✓ Controlador de DMA “imita” a CPU para realizar operações de E/S



# Controlador DMA

O controlador de DMA pode ser encarado como um terceiro elemento que, quando solicitado, assume o controle do barramento e realiza transferência de dados entre a interface e a memória. Durante as operações de DMA, o dado é transferido diretamente entre a interface e a memória.

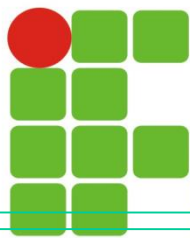




# Operação do DMA

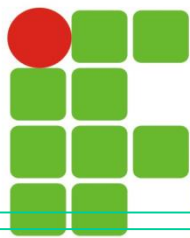
- ✓ Quando a CPU quer ler ou escrever dados envia para o DMA:
  - ✓ Informação se o dado será lido ou escrito
  - ✓ O endereço do dispositivo de E/S
  - ✓ Local na memória que será lido ou escrito
  - ✓ O número de palavras a serem lidas ou escritas

Quando a transferência é feita, o DMA envia um sinal de interrupção para a CPU



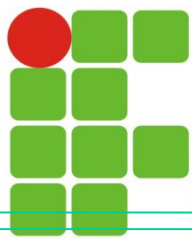
# Operação do DMA

- ✓ Antes que uma operação de DMA possa ocorrer, o controlador deve ser inicializado de acordo com os seguintes itens:
  - ✓ selecionar o tipo de função: leitura ou escrita na memória.
  - ✓ selecionar o tipo de transferência: rajada ou um único byte (ou word – 16 bits).
  - ✓ especificar a quantidade de bytes (ou words) a serem transferidos.
  - ✓ indicar a prioridade dos canais.
  - ✓ especificar o endereço inicial da memória.
  - ✓ habilitar os canais a serem usados.



# Operação do DMA

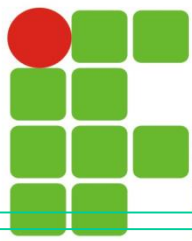
- ✓ A interface, que precisa realizar um ciclo de DMA, envia um sinal DRQ (DMA request) para o controlador.
- ✓ O controlador resolve a prioridade deste pedido e envia um HRQ (hold request) para o circuito gerador de estados de espera do PC.
- ✓ O circuito gerador de estados de espera monitora as linhas de estado da CPU, procurando um estado passivo, ou seja, procura um instante em que o barramento esteja inativo ou próximo do final de um ciclo.



# Operação do DMA

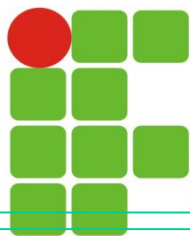
- ✓ Quando um estado passivo é detectado, é enviado um sinal not ready para a CPU, fazendo com que a mesma entre em estado de espera e um sinal de HLDA (hold acknowledge) é enviado para o controlador, indicando que, no próximo pulso de clock, o barramento estará livre e que o ciclo de DMA poderá ocorrer.
- ✓ O controlador de DMA, ao receber o HLDA, envia um sinal DACK para a interface que solicitou o DMA. Esse sinal atua como o selecionador do chip (chip select), conectando a interface ao barramento de dados do PC.



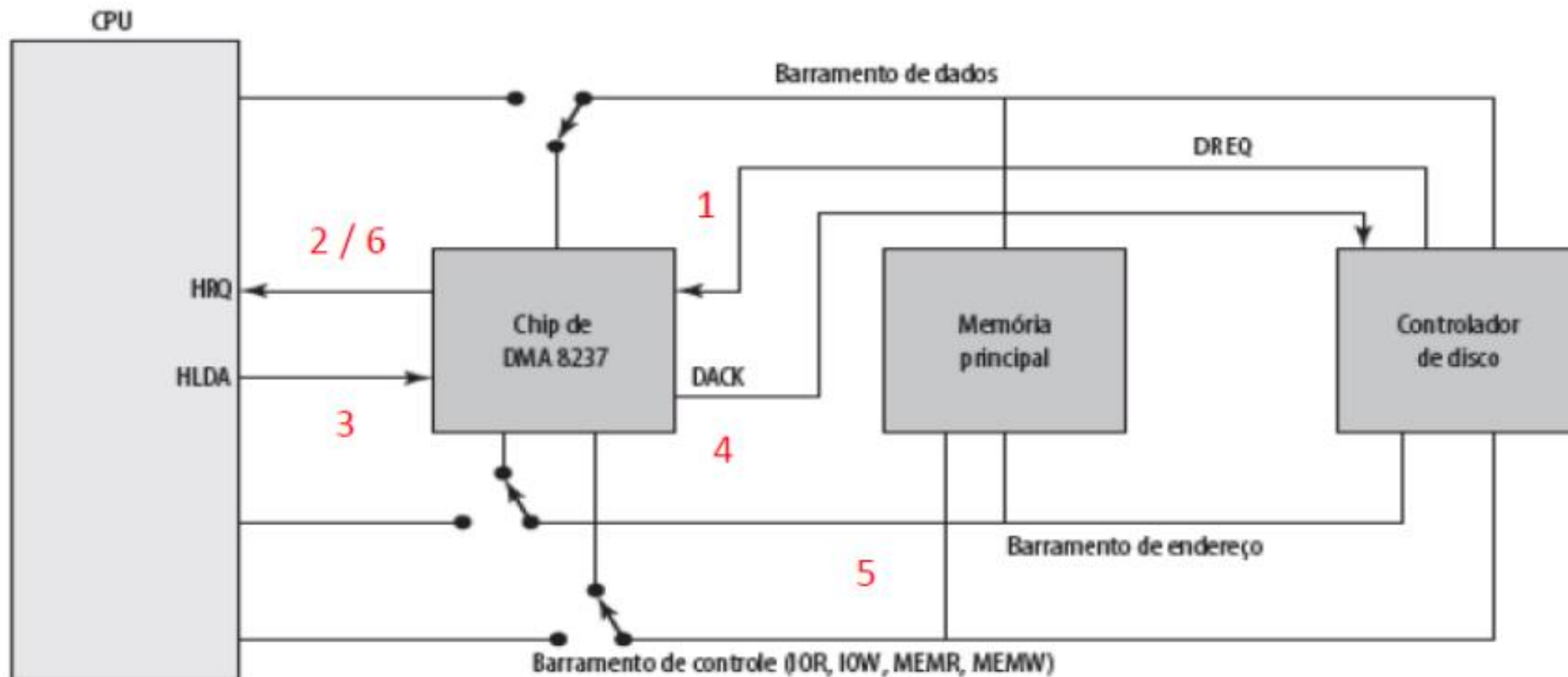


# Operação do DMA

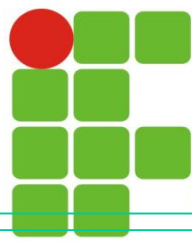
- ✓ Agora, o controlador coloca no barramento o endereço da memória onde acontecerá a operação de DMA. Em seguida, envia os sinais de controle de forma a realizar o ciclo programado.
- ✓ Após receber o DACK, a interface retira o pedido de DMA que havia feito através da linha DRQ. Quando o controlador termina o ciclo, ele retira o pedido HRQ que enviou para o circuito gerador de estados de espera, que, por sua vez, remove o sinal HLDA, indicando que a CPU irá assumir o controle do barramento e o circuito gerador de estado de espera retira o sinal not ready e habilita os buffers da CPU.



# Operação do DMA



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



# Roubo de Ciclo pelo DMA

Controlador de ADM toma conta do barramento por um ciclo

Transfere uma palavra de dados de cada vez

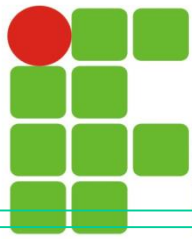
Diferente de interrupção

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



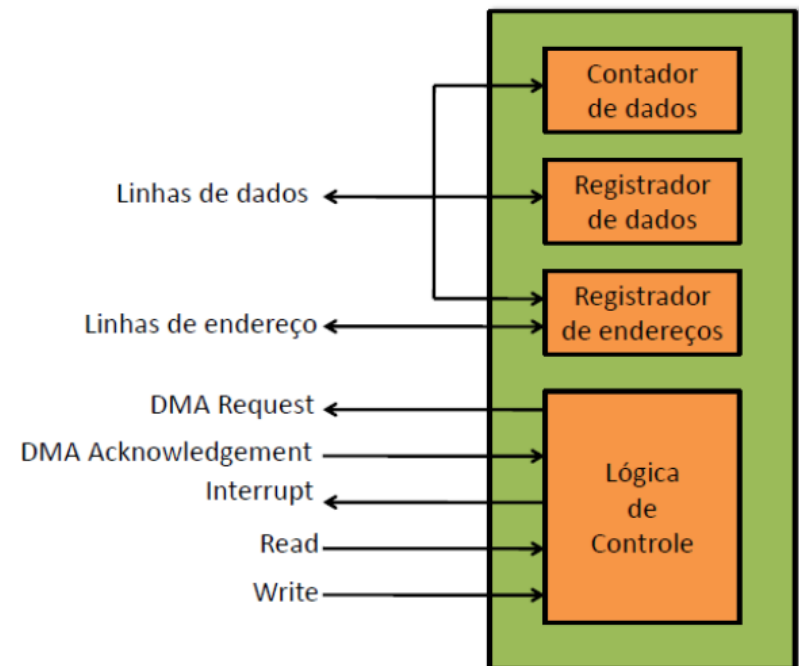
# Estrutura Interna de um DMA

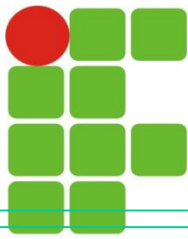
Algumas possibilidades:

Barramento único, DMA separado

Barramento único, DMA-E/S integrados

Barramento específico de E/S

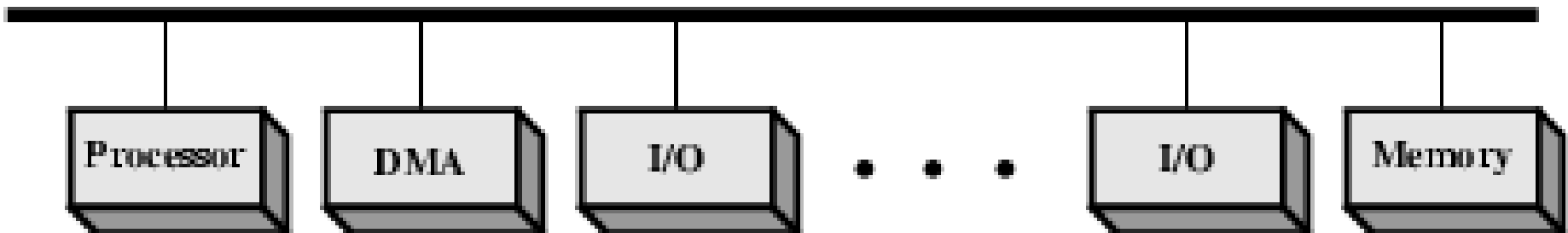




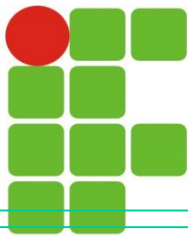
# Configurações do DMA

## 1. Barramento único e Controlador de ADM separado

- Cada transferência utiliza duas vezes o barramento
  - Do dispositivo de E/S para ADM e do ADM para memória



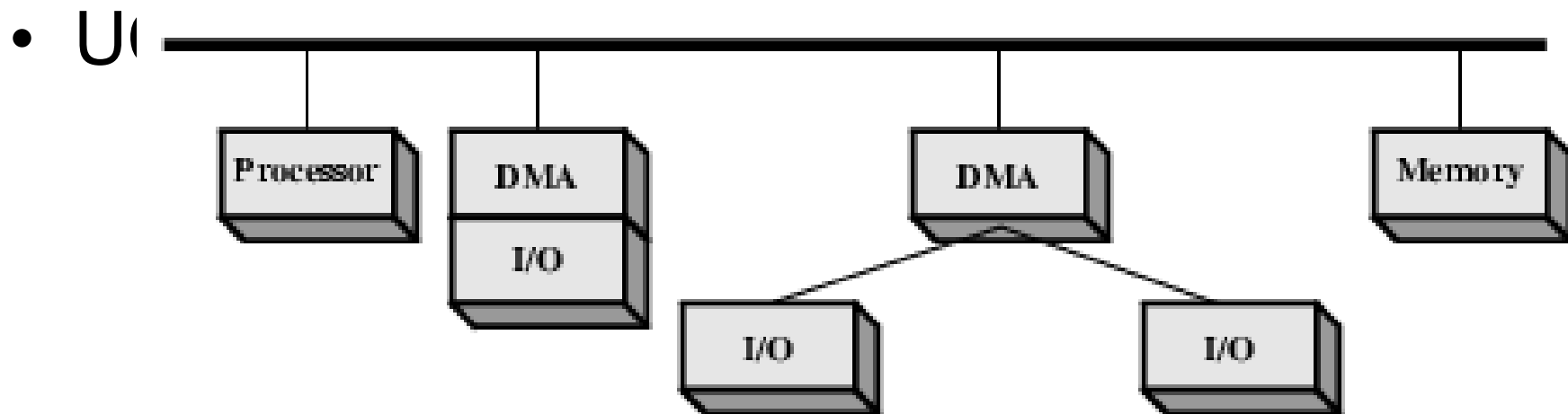
Ineficiente – a transferência de cada palavra consome vários ciclos de barramento



# Configurações do DMA

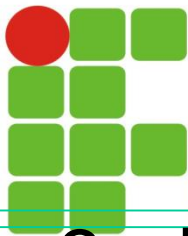
## 2. Barramento único e Controlador de ADM integrado

- Controlador pode suportar mais de um dispositivo
- Cada transferência utiliza o barramento uma única vez
  - ADM para memória



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

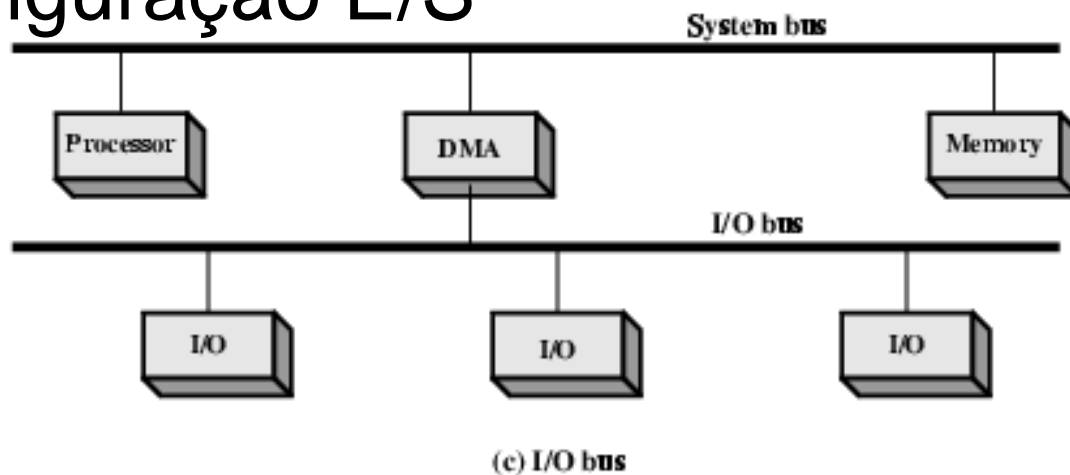
Eficiente e demorada



# Configurações do DMA

## 3. 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
- Vantagem: mais facilidade de expansão de configuração E/S



Eficiente e demorada