

Organização e Arquitetura de Computadores

Entrada e saída

Alexandre Amory
Edson Moreno

Nas Aulas Anteriores

- Foco na Arquitetura e Organização internas da Cleo
 - Modelo Von Neuman
 - Circuito combinacional
 - Circuito sequencial e máquinas de estados
 - Assembly

Na Aula de Hoje

- ... Mas o processador é somente uma parte de um sistema computacional
- O sistema de **Entrada e Saída** também é essencial e possui grande impacto no desempenho do sistema computacional
- Veremos:
 - Tipos e características de dispositivos de IO
 - Interface dos dispositivos de IO
 - Modos de transferência

Impacto de IO no desempenho

- Exemplo:
 - Suponha que tenhamos um aplicativo que execute em 100 segundos, dos quais 90 segundos são gastos no processador e o restantes é gasto em atividade de entrada e saída. Se o desempenho do processador melhora 50% a cada ano, mas não há qualquer melhora no sistema de entrada e saída, quanto mais rápido este aplicativo vai ficar mais rápido no final de 5 anos?

Sabendo-se que,

$$T_{\text{melhorado}} = T_{\text{original}} / (1 + \text{melhora})$$

e que o ganho de desempenho é dado pelo

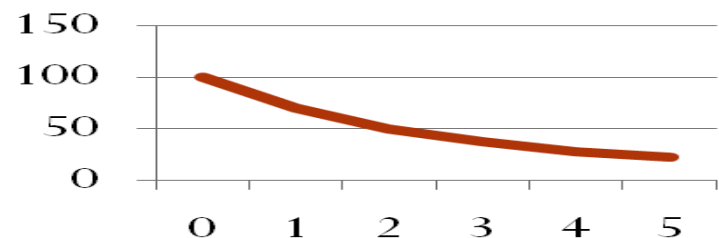
$$\text{Ganho de desempenho} = T_{\text{original}} / T_{\text{melhorado}}$$

Impacto de IO no desempenho

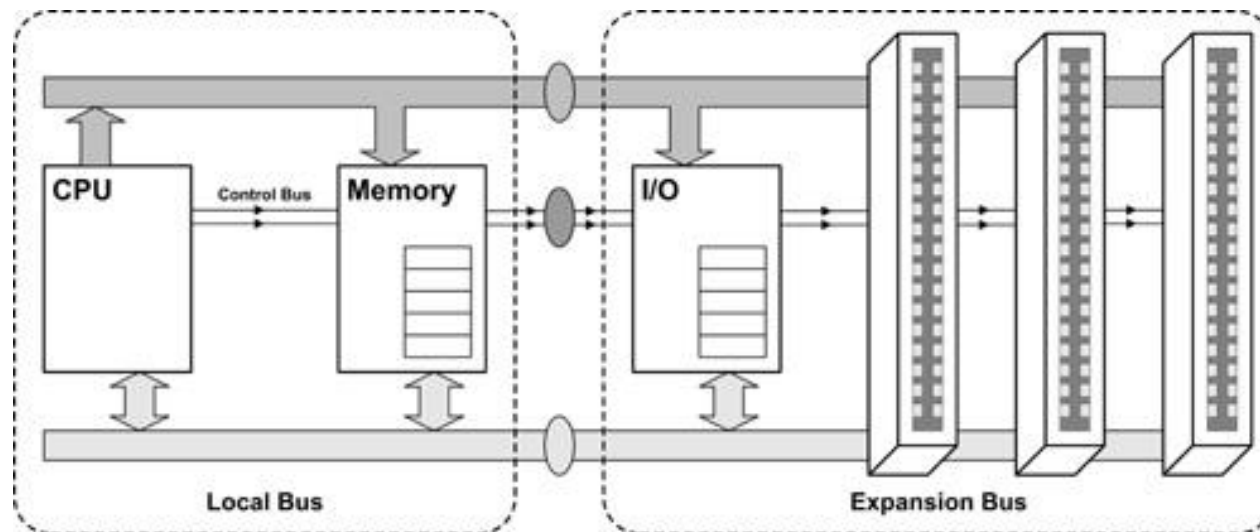
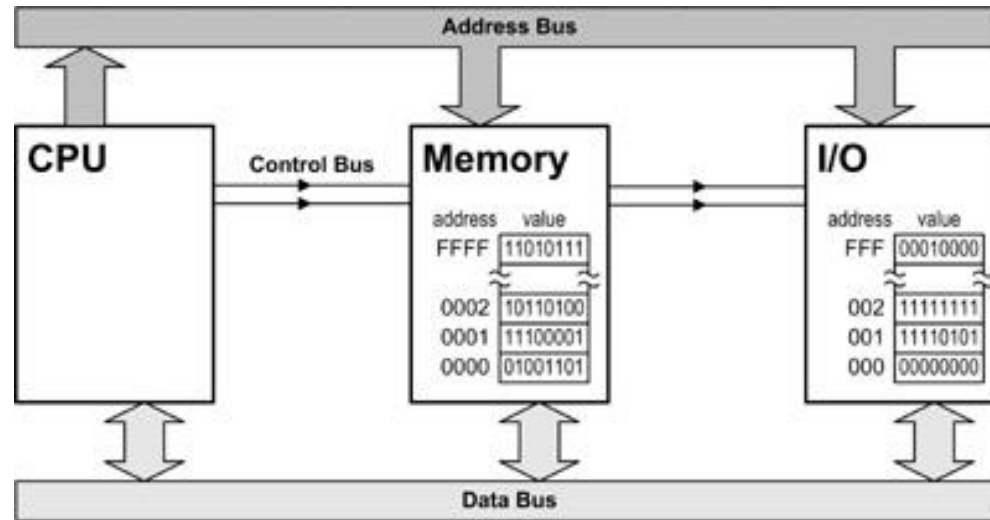
- Exemplo:
 - Suponha que tenhamos um aplicativo que execute em 100 segundos, dos quais 90 segundos são gastos no processador e o restante é gasto em atividade de entrada e saída. Se o desempenho do processador melhora 50% a cada ano, mas não há qualquer melhora no sistema de entrada e saída, quanto mais rápido este aplicativo vai ficar mais rápido no final de 5 anos?

Após n anos	Tempo de processador	Tempo de IO	Tempo decorrido	% tempo decorrido de IO
0	90	10	100	10
1	$90/1,5 = 60$	10	70	14
2	$60/1,5 = 40$	10	50	20
3	$40/1,5 = 27$	10	37	27
4	$27/1,5 = 18$	10	28	36
5	$18/1,5 = 12$	10	22	45

Ganhos
reduzidos !!!!



Esquema Básico



Características de um Dispositivo IO

- Dispositivos de IO podem ser divididos quanto ao:
 - Comportamento do dispositivo
 - **Entrada**: somente pode ser lido - Teclado
 - **Saída**: somente pode ser escrito - Monitor
 - **Armazenamento**: pode ser tanto lido quanto escrito
 - Parceiro do dispositivo
 - Aquilo que está do outro lado do dispositivo que não o processador/a máquina
 - **Pessoa** ou **máquina** que alimenta ou colhe dados da interface
 - Velocidade do dispositivo
 - Definida como a **velocidade** máquina de transferência entre
 - Dispositivo de IO / memória principal / processador

Tipos e Características dos Dispositivos de E/S

Dispositivo	Comportamento	Parceiro	Taxa de dados (KB/segundo)
Teclado	entrada	ser humano	0,0001
Mouse	entrada	ser humano	0,0038
Entrada de voz	entrada	ser humano	0,264
Entrada de som	entrada	máquina	3
Escaner	entrada	ser humano	3,2
Saída de voz	saída	ser humano	0,2640
Saída de som	saída	ser humano	8
Impressora a laser	saída	ser humano	3,2
Monitor gráfico	saída	ser humano	800 a 8000
Modem	entrada ou saída	máquina	0,016 a 0,064
Rede/LAN	entrada ou saída	máquina	100 a 10000
Rede/LAN sem fio	entrada ou saída	máquina	11 a 54
Disco ótico	armazenamento	máquina	80
Disco Magnético	armazenamento	máquina	240 a 2460

* Por prof J. L. Güntzel.

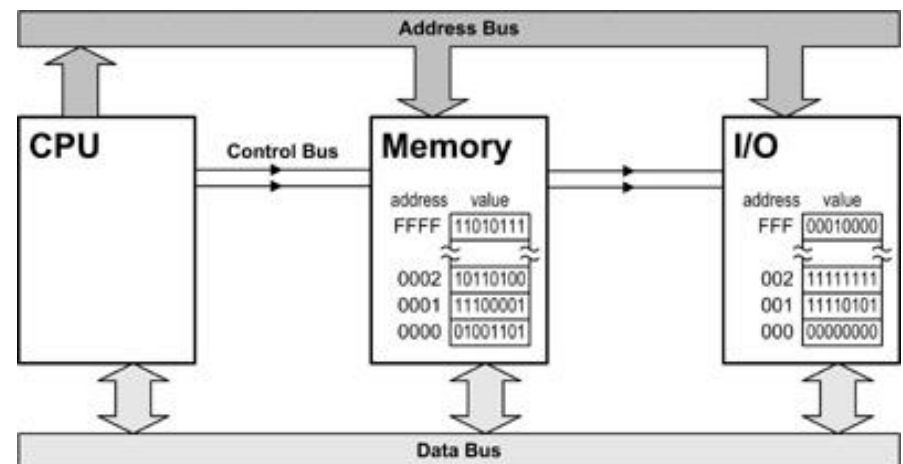
Traduzido de Computer Organization and Design Patterson & Hennessy

Envio de comandos

- O processador precisa ser capaz de ...
 - Endereçar o dispositivo
 - Gerar uma ou mais palavras que contenham comandos
- Métodos de endereçamento
 - E/S mapeada em memória
 - Parte do espaço de endereçamento é atribuída aos dispositivos de E/S
 - Processador deposita e/ou coleta dados a partir deste espaço mapeado
 - Dispositivo interpreta comandos/salva dados no mesmo espaço
 - Comandos especiais de E/S
 - Leituras/escritas são interpretadas como comandos diretos para dispositivos de IO
 - Mecanismos conectado ao barramento fica responsável por encaminhar comando diretamente a porta de saída
 - Não há espaço de memória compartilhado

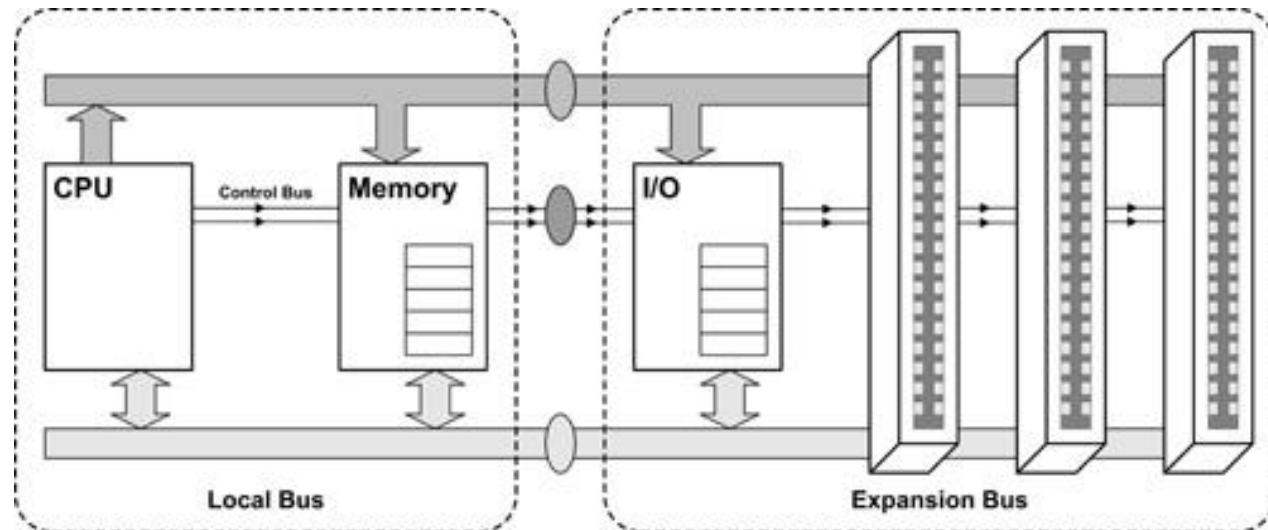
Modos de transferência

- Questões motivadoras para analisar modos de transferência
 - Quando efetuar a transferência, de modo a atender requisitos de projeto?
 - Quais mecanismos para controlar as transferências com eficiência?
- Requisitos
 - Rápida resposta a eventos críticos
 - Eventos de segurança requerem resposta imediata
 - Eventos de tempo real tem um tempo máximo preciso para serem respondidos
 - Não sobrecarregar CPU com atividades de E/S
 - Algumas atividades ,como acesso a disco e *refresh* de memória, exigem altas taxas de E/S
 - Dois modos básicos
 - Programado
 - Não-programado



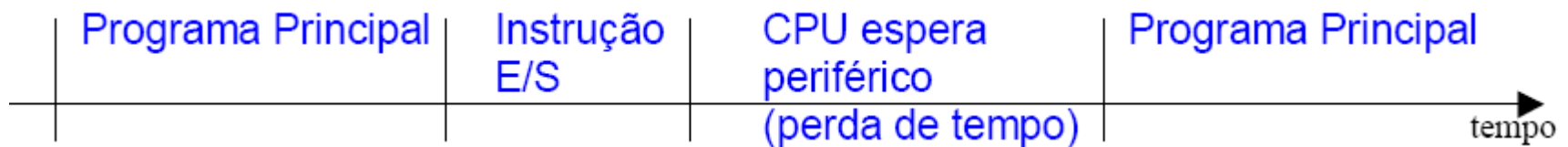
Entrada/saída Programada

- E/S controlada pela CPU
- Procedimento
 1. Em instante determinável, processador pergunta para periféricos se este está apto a receber ou transmitir informação
 2. Em caso afirmativo realiza transferência
- Tipos
 - Modo bloqueado (busy wait)
 - Polling
 - Interjeição



Modo Bloqueado

- Uma vez iniciada a comunicação, a CPU fica bloqueada
 - Ocupada e escrava do periférico até término da operação
- Problema
 - periféricos são muito mais lentos que CPU



- Exemplo de utilização
 - Máquinas reativas → Esperam ação externas captadas por sensores para então responder ao meio
 - Computador dedicado a tarefas dependentes de um único periférico
 - Pesagem → Única atividade da máquina é pesar

Polling

- CPU possui controle total da comunicação
 - Determina os instantes de tempo que ocorrerão transferências
- CPU periodicamente testa se algum dispositivo quer se comunicar
 - A pergunta pode ser diretamente em um porta de entrada da CPU, que sinaliza o pedido
- Otimização pode ser feita com controladores
 - CPU periodicamente testa registradores de estado dos controladores de E/S
 - Estes registradores sinalizam possíveis transferência a serem realizadas

Polling - Controlador e Periférico

1. Periférico transfere dados para controlador
 - Coloca dados no barramento de dados
 - Ativa sinal send
2. Controlador armazena dados
 - Dados lidos são colocados em buffers
 - Responde ack para periférico
 - Sinaliza “dado presente” para CPU ativando *flag*
3. Controlador habilita novo envio de dados
 - Remove ack se
 - Flag desativada (CPU já buscou)
 - Tem espaço no buffer local para mais dados
4. Periférico habilita novo envio de dados
 - Remove send ao detectar remoção de ack

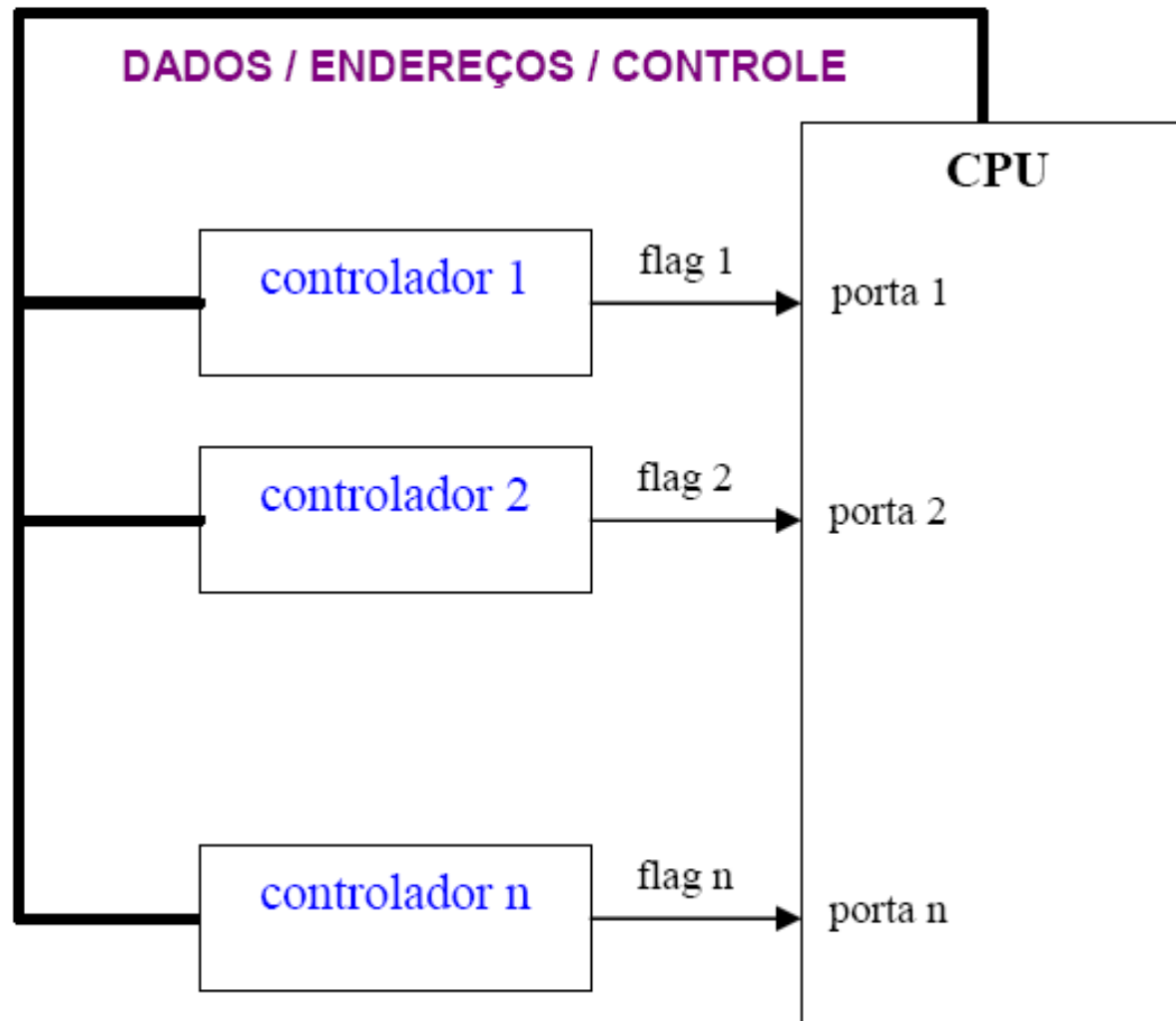


Polling – CPU e Controlador

1. CPU verifica se tem dados para serem consumidos
 - Executa polling periodicamente no registrador de estado
2. CPU lê o dado do controlador
 - Quando flag ativa
 - Lê dado do barramento de dados
 - Armazena dado em memória
3. CPU avisa controlador de leitura com sucesso
 - CPU envia sinal de controle para desativar a flag
4. Periférico pode enviar novo dado

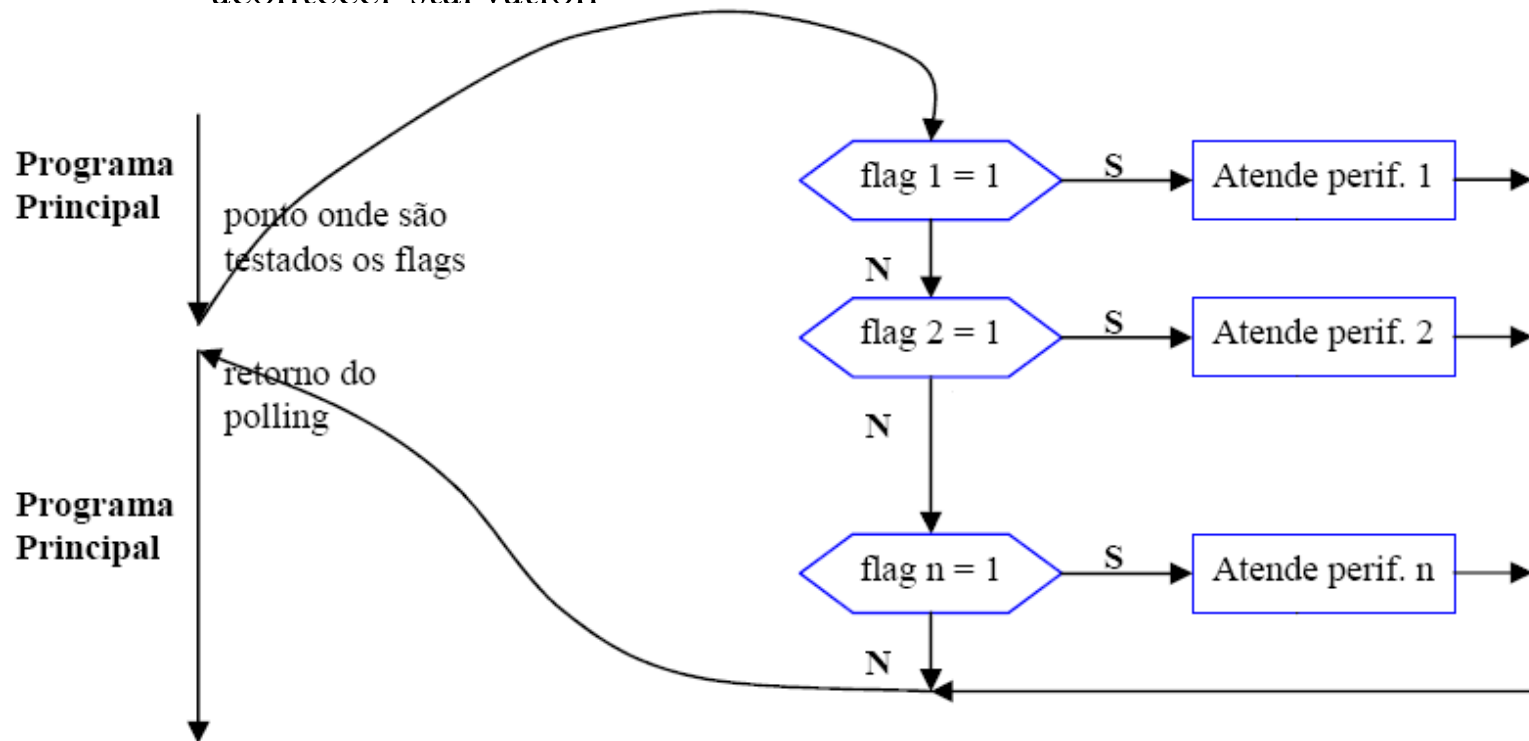


Polling em Vários Dispositivos 1



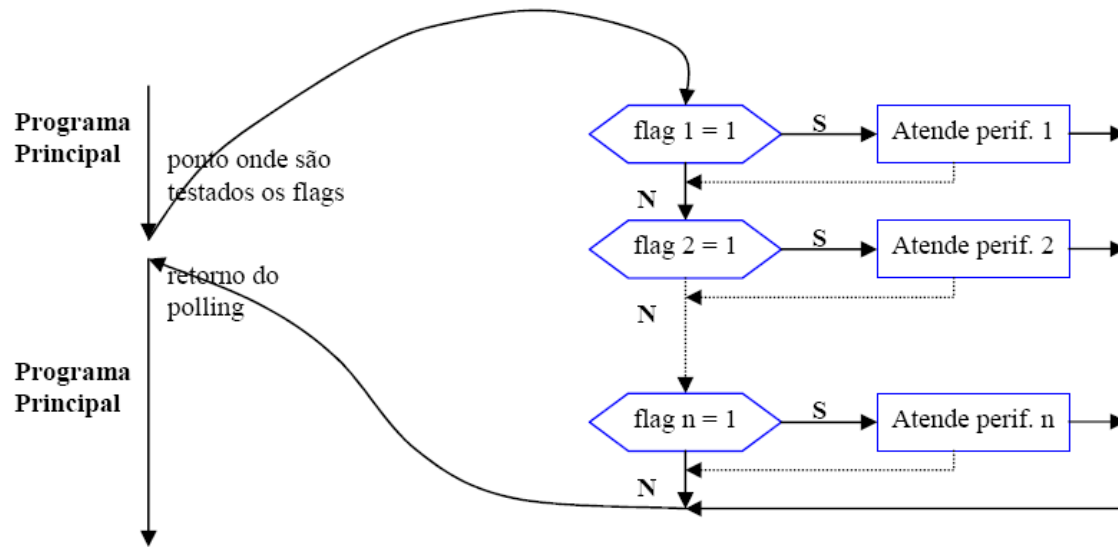
Polling em Vários Dispositivos 2

- Verificação dos *flags*
 - Prioridade implícita
 - Normalmente atende apenas um periférico por polling
 - Prioridade de atendimento ao menor número de periférico pode acontecer starvation



Polling em Vários Dispositivos 3

- Solução para evitar starvation
 - Leitura de mais de um periférico



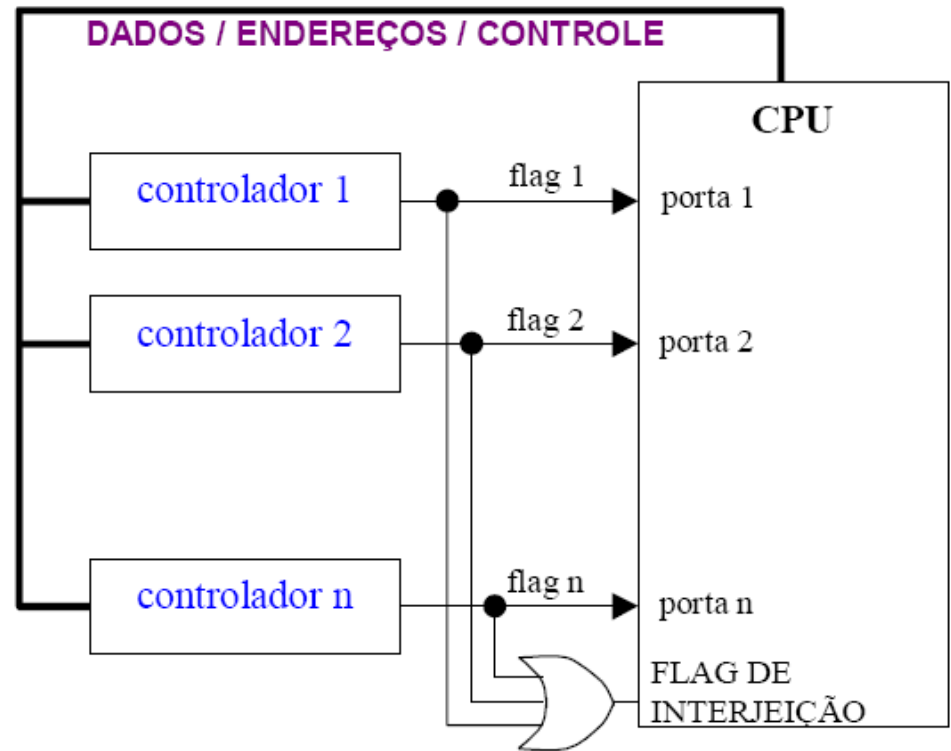
- Esta solução pode causar muito atraso para a CPU
- Projetista decide em função de
 - Número de periféricos
 - Frequência estimada dos pedidos de atendimento
 - Frequência de teste dentro do programa principal
- Existe alguma outra solução melhor?

Polling em Vários Dispositivos 3

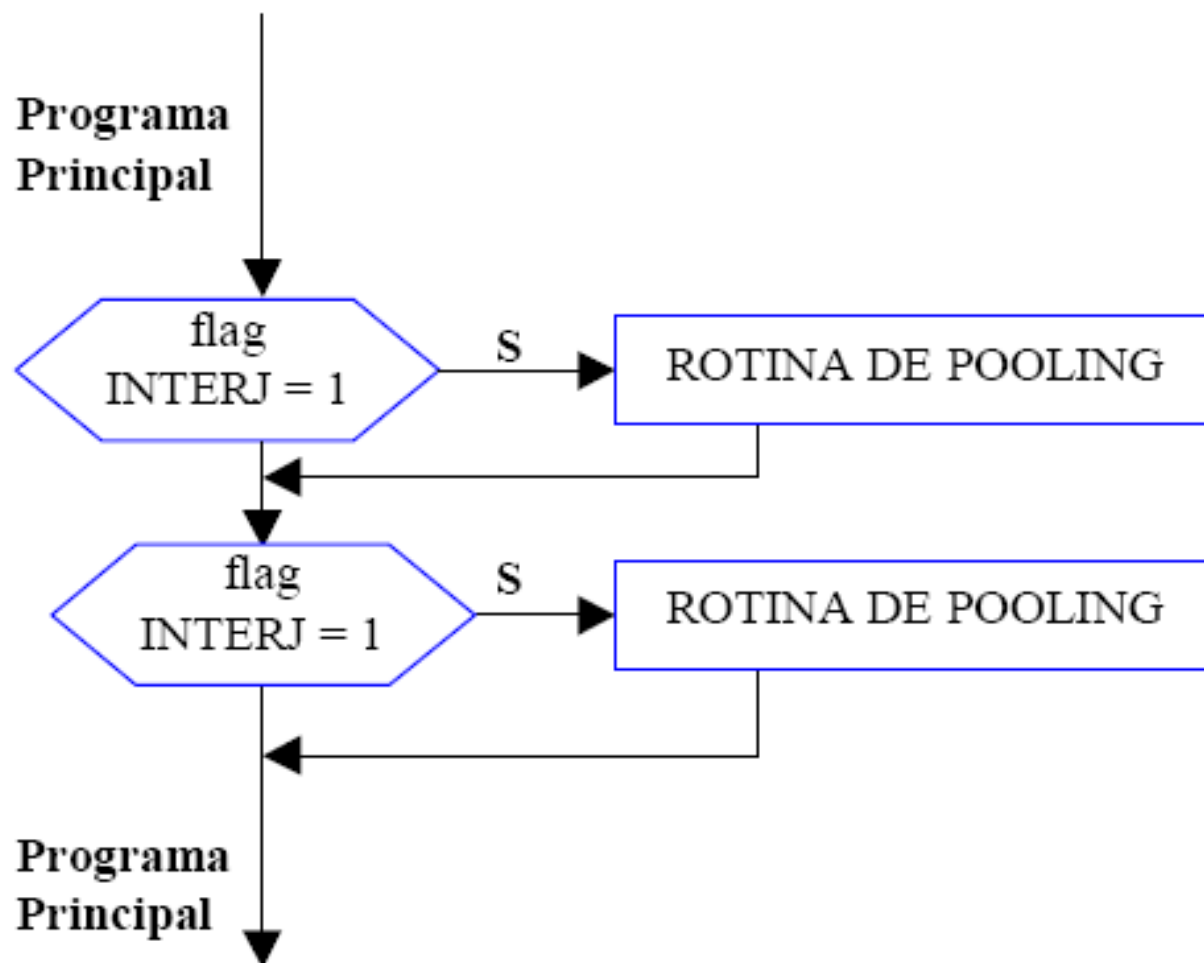
- Existe alguma outra solução melhor?
 1. Atendimento com fila. Quem foi atendido vai para o final da fila e tem a última prioridade no próximo polling.
 - Analise vantagens e desvantagens
 2. ...
 3. Interjeição

Interjeição

- O que é?
 - Otimização do polling
- Funcionalidade
 - Ou lógico de todos os flags
- Consequência
 - Menor tempo com o teste
- Procedimento
 - CPU só testa um bit → independentemente do número de periféricos

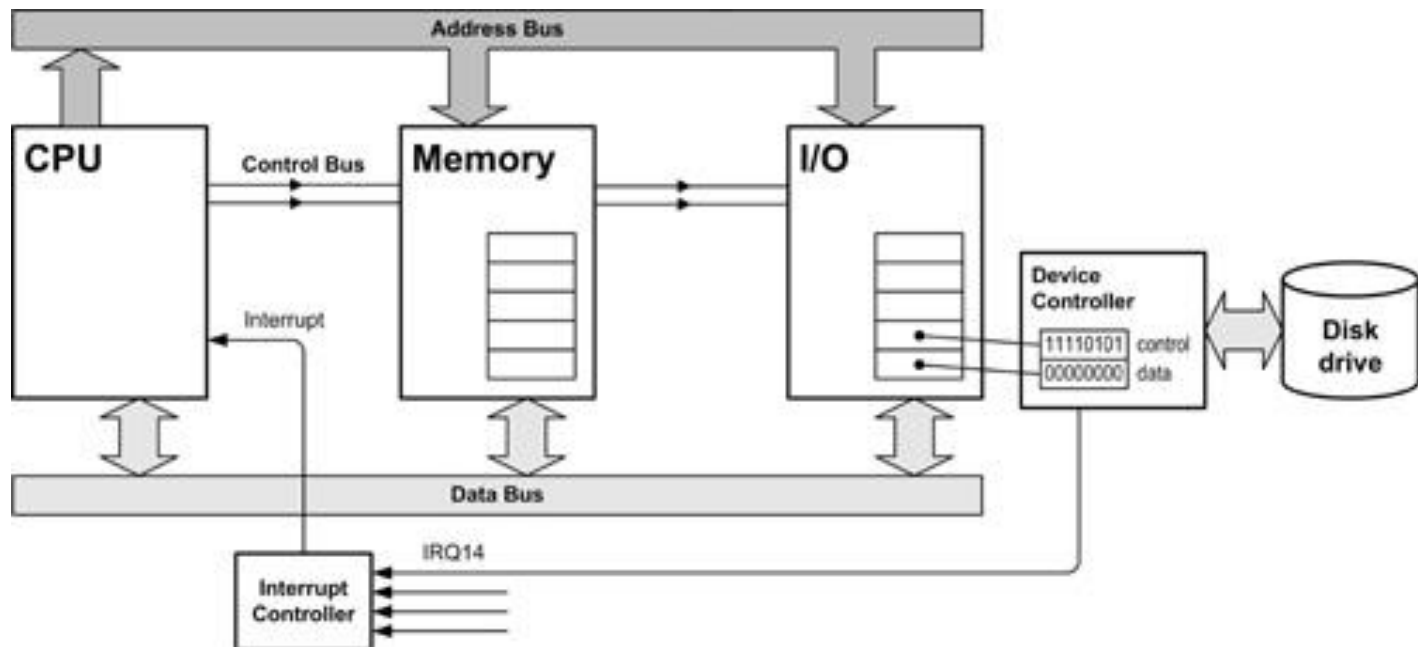


Interjeição – Funcionalidade



Interrupção

- Vantagem
 - Evita tempo desperdiçado com múltiplos testes, que é inerente ao polling
- Funcionamento
 - CPU/controlador é avisado pelo periférico que este deseja transmitir/receber dados



Interrupção (Cont...)

- Principais características de interrupções de E/S
 - Assincronismo em relação a qualquer instrução
 - Ocorre a qualquer instante
 - Seu tratamento NÃO quebra a execução da instrução corrente
 - Teste de interrupção é feito depois da execução da instrução
 - Diferenciar
 - Interrupção de hardware externo a CPU
 - Origem de um periférico
 - Interrupção de hardware interno a CPU
 - Origem de um circuito interno ao processador (exemplo: exceções
→ divisão por zero)
 - Interrupção de software
 - Chamadas ao sistema operacional ou outros programas residentes

Interrupção – Funcionamento

1. Priorizar interrupções
 - Atender de maior prioridade no caso de simultaneidade de interrupções
2. Tratamento de interrupções aninhadas
 - Desabilitar interrupções, caso não seja permitido aninhamento de interrupções
 - Prosseguir normalmente, caso contrário
3. Salvar contexto (PC, registradores)
 - Salvamento rápido (um ciclo → próprio processador)
 - Salvamento lento (diversos ciclos → sequência de instruções para por na pilha)
4. Identificar periférico
5. Desviar PC para rotina de tratamento da interrupção
6. Executar programa de interrupção
7. Recuperar contexto
8. Reabilitar interrupções (caso tenham sido desabilitadas)

DMA – Acesso Direto à Memória

- Problema

- Interrupção não necessita que CPU aguarde por eventos
- Porém **CPU é responsável por realizar transferência de dados**

- Solução

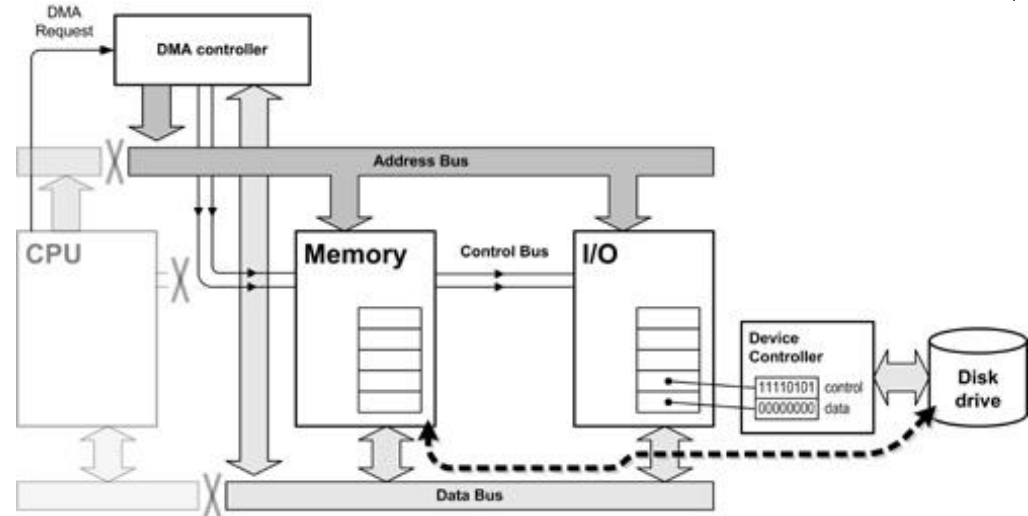
- DMA

- Funcionamento

- Dispositivo controlador (*DMA Controller - DMAC*) é responsável pela transferência de dados

- Características de funcionamento

- Mecanismos de interrupção continuam sendo utilizados
 - Mas apenas no término de um evento de E/S, ou na ocorrência de erros
- Comparando com polling e interrupção, DMA deve ser utilizado para interface de periféricos que necessitem de grande vazão
- Durante operação DMAC se torna mestre do barramento e controla transferência entre E/S e Memória



DMA - Transferência genérica

1. CPU programa DMAC com
 - Identificação do dispositivo que solicitou DMA
 - Operação a ser realizada no dispositivo (escrita/leitura)
 - Endereço origem e destino (da memória principal e do periférico)
 - Número de bytes a serem transmitidos
2. DMAC assume barramento, iniciando a operação
 - Transferência inicia quando dado está disponível. O DMAC fornece o endereço de leitura ou escrita. Se requisição necessitar mais de uma transferência, o DMAC gera próximo endereço de memória
 - Por meio deste mecanismo o DMA pode transferir milhares de bytes do entre periférico e memória, sem interromper a CPU
3. Concluída transferência de DMA, DMAC interrompe CPU que poderá verificar se operação foi realizada com sucesso examinando memória ou interrogando DMAC
 - Uma vez programado DMAC, DMAC e CPU operam em paralelo

Resumo

- E/S podem ter grande impacto no desempenho de sistemas computacionais
- Modos de Comunicação CPU $\leftrightarrow$ Periférico
 - Programado
 - Bloqueado
 - Polling: para transferência CPU $\leftrightarrow$ Periférico com pouca vazão porém com periodicidade
 - Interjeição
 - Não programado
 - Interrupção: eventos pouco frequentes e não periódicos
 - DMA: para transferência CPU $\leftrightarrow$ Periférico com maior vazão. O processador não é interrompido.