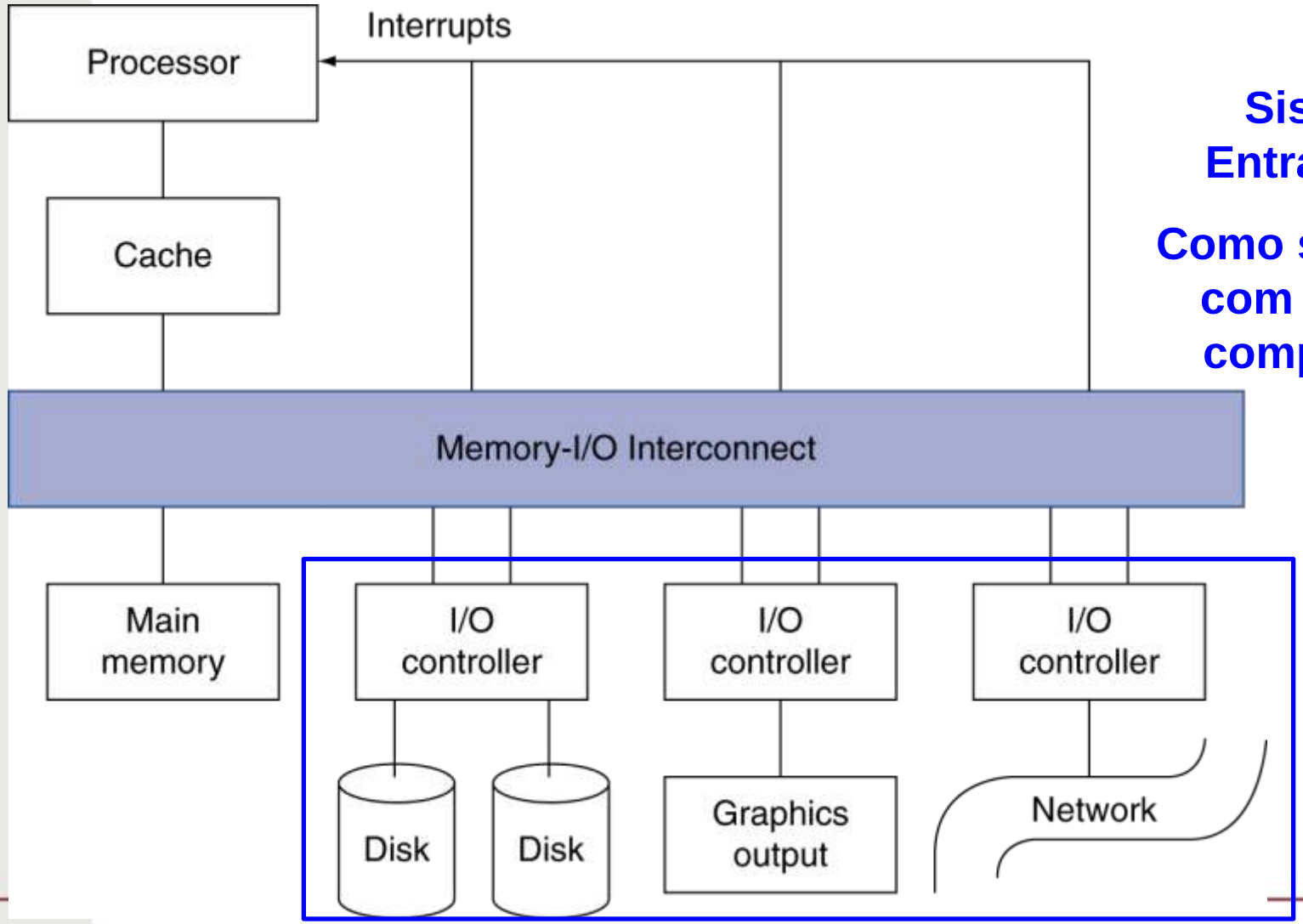
Infraestrutura de Hardware

Entrada/Saída: Comunicação Processador, Memória e E/S

Perguntas que Devem ser Respondidas ao Final do Curso

- Como um programa escrito em uma linguagem de alto nível é entendido e executado pelo HW?
- Qual é a interface entre SW e HW e como o SW instrui o HW a executar o que foi planejado?
- O que determina o desempenho de um programa e como ele pode ser melhorado?
- **Que técnicas um projetista de HW pode utilizar para melhorar o desempenho?**

Organização Típica de Sistemas Computacionais



Sistema de Entrada/Saída:

Como se comunica com os demais componentes?

Comunicação entre Componentes

- Barramento: canal de comunicação compartilhado, geralmente com uma interface padrão
  - Conjunto de fios paralelos para transferência de dados e sincronização desta transferência

- Redes de E/S: conexões seriais ponto-a-ponto de alta velocidade com switches
  - Alternativa mais recente
  - Sistemas Embarcados: NoC (Network on Chip)

Barramento

- Vantagens
  - Flexibilidade
    - novos dispositivos podem ser adicionados facilmente e podem até ser movidos para outros computadores que utilizarem o mesmo padrão de barramento
  - Baixo custo
    - um único conjunto de fios é compartilhado
- Desvantagem
  - Gargalo de comunicação
    - Largura de banda limita o throughput de E/S
- Velocidade máxima do barramento limitado por:
  - Comprimento do barramento
  - Número de dispositivos conectados

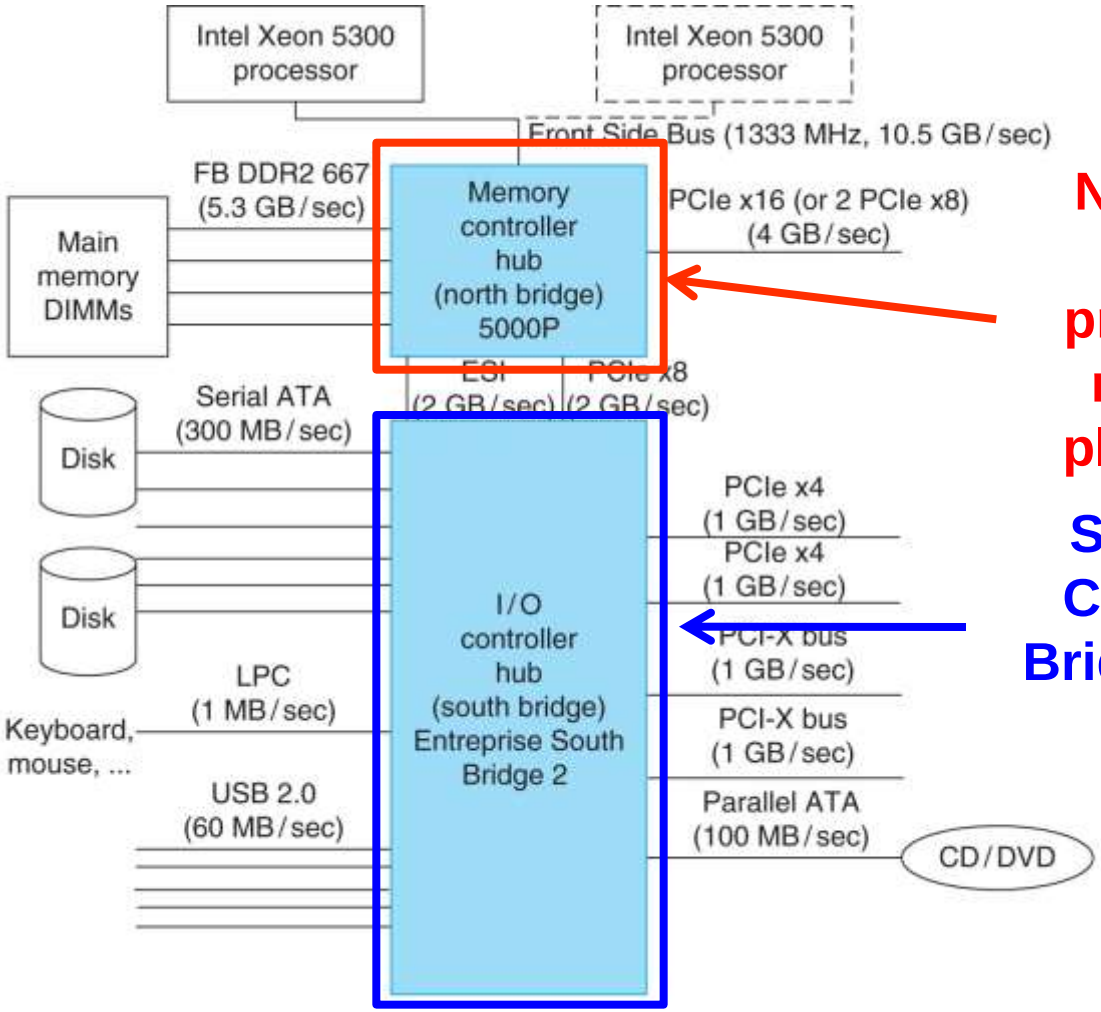
Tipos de Barramentos

- **Processor-memory bus**
  - Curto e de alta velocidade
  - Adaptado à memória para maximizar largura de banda entre processador e memória
  - Otimizado para transferência de blocos de cache
- **I/O bus**
  - Mais comprido e lento
  - Acomoda uma extensa variedade de dispositivos de E/S
  - Conecta-se ao processor-memory bus através de um bridge (ponte) e/ou um backplane bus
  - Exs: SCSI, USB, Firewire
- **Backplane bus**
  - Usado como um barramento intermediário para conectar I/O buses a processor-memory bus
  - Ex: PCIexpress

Exemplos de I/O Bus

	Firewire	USB 2.0	PCI Express	Serial ATA	Serial Attached SCSI
Intended use	External	External	Internal	Internal	External
Devices per channel	63	127	1	1	4
Data width	4	2	2/lane	4	4
Peak bandwidth	50MB/s or 100MB/s	0.2MB/s, 1.5MB/s, or 60MB/s	250MB/s/lane 1×, 2×, 4×, 8×, 16×, 32×	300MB/s	300MB/s
Hot pluggable	Yes	Yes	Depends	Yes	Yes
Max length	4.5m	5m	0.5m	1m	8m
Standard	IEEE 1394	USB Implementers Forum	PCI-SIG	SATA-IO	INCITS TC T10

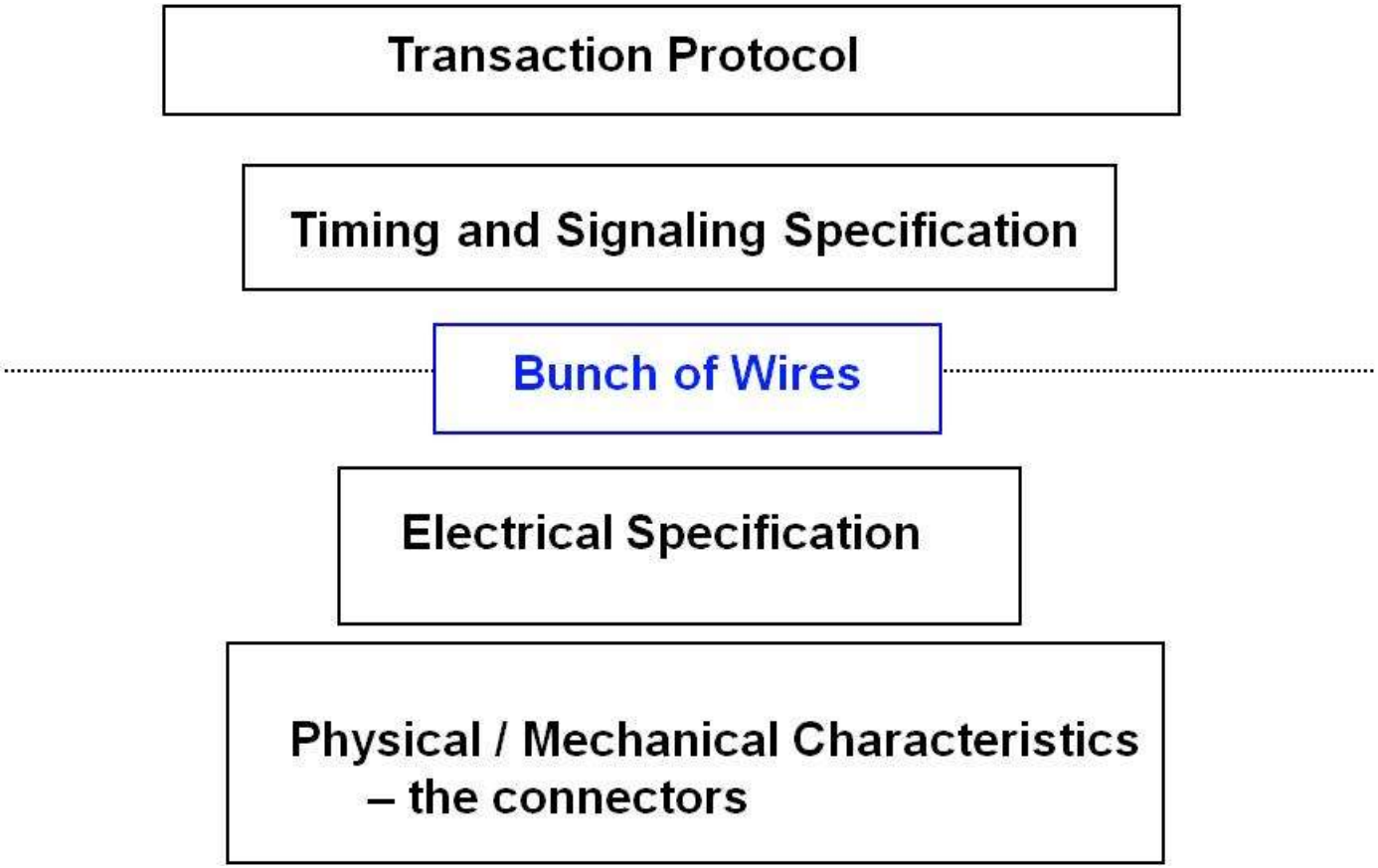
Exemplo de Organização de E/S: Intel 5000P



North Bridge: Conecta processador à memória e à placa de vídeo

South Bridge: Conecta North Bridge aos vários I/O buses

O Que Define um Barramento?



Transação de E/S

- Sequencia de operações sobre um canal de comunicação que inclui um pedido e pode incluir uma resposta
  - Pedido ou resposta pode incluir a transmissão de dados
  - Iniciada por um pedido que pode ser realizado através de várias operações sobre um barramento
- Inclui tipicamente duas partes
  - Enviar um endereço
  - Receber ou enviar dados
- É definida pelo tipo de acesso à memória
  - Transação de leitura lê dados da memória e envia dados ao processador ou dispositivo E/S
  - Transação de escrita escreve dados na memória vindo do processador ou dispositivo de E/S

Barramentos Síncronos e Assíncronos

- **Barramento Síncrono**
  - Inclui um clock nas linhas de controle
  - Protocolo de comunicação fixo baseado no clock
  - Exemplo: Processor-Memory Bus
- **Barramento Assíncrono**
  - Não utiliza o sinal de clock
  - Requer um protocolo de handshake e mais linhas de controle
  - Exemplo: I/O Bus

Síncronos x Assíncronos

- **Barramento Síncrono**

  Vantagens:

  - envolve muito menos lógica
  - pode operar em altas velocidades

  Desvantagens:

  - Todo dispositivo no barramento deve operar no mesmo clock rate
  - Não podem ser longos se são rápidos (clock skew)

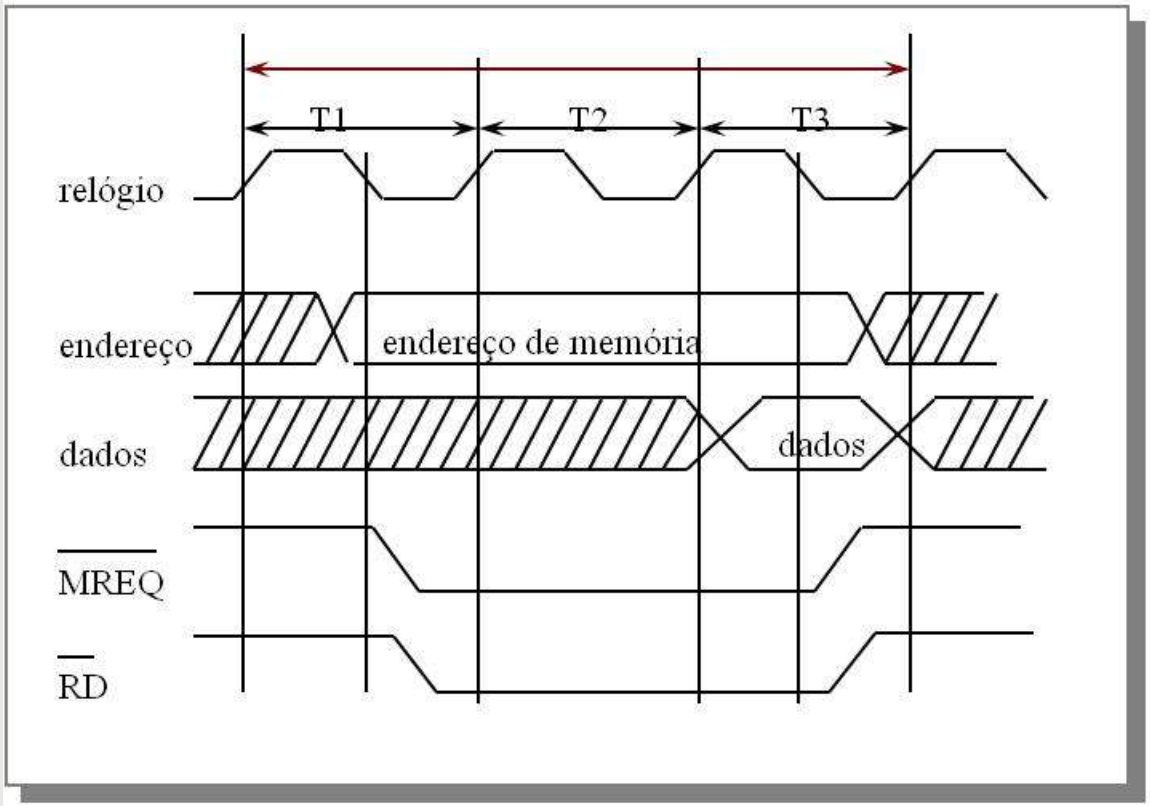
- **Barramento Assíncrono**

  Vantagens:

  - Acomoda grande variedade de dispositivos com diferentes velocidades
  - Podem ser longos sem se preocupar com clock skew ou com problemas de sincronização

  Desvantagem: Mais lento

Barramento Síncrono



$\overline{MREQ}$ - 0
RD - 0

T_1 - CPU ativa sinais de controle e endereço

Memória decodifica endereço

T_2 - Endereço estável no barramento

Memória coloca dados no barramento de dados

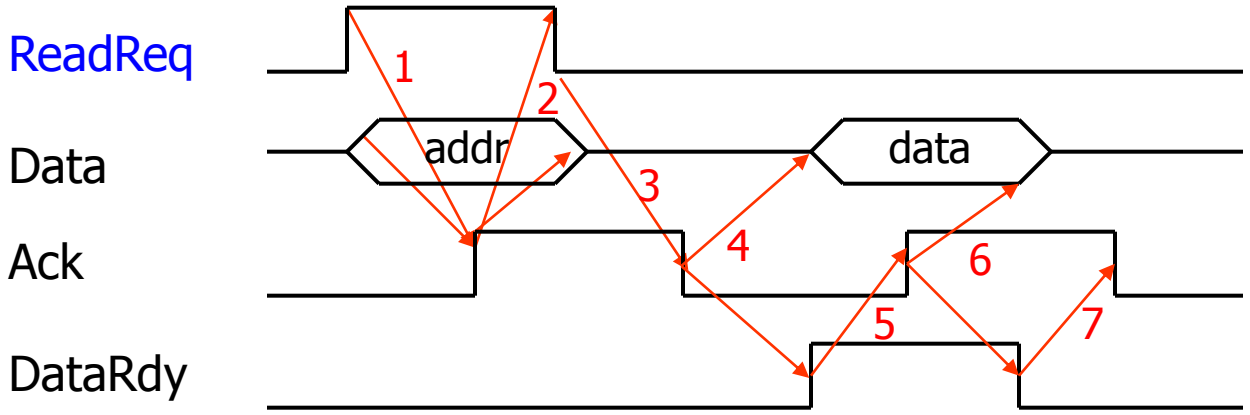
T_3 - Memória libera dados no barramento

$\overline{MREQ}$ - 1
$\overline{RD}$ - 1

-Dados são lidos pela CPU
-CPU desabilita controle

Protocolo de Barramento Assíncrono

- Leitura da memória por um I/O device



Dispositivo E/S sinaliza requisição ativando ReadReq e colocando addr nas linhas de dados.

1. Memória vê ReadReq, lê addr da linha de dados e ativa Ack
2. Dispositivo E/S vê Ack e desativa ReadReq e linhas de dados
3. Memória vê ReadReq e desativa Ack
4. Quando dado da memória está pronto disponibiliza e ativa DataRdy
5. Dispositivo E/S vê DataRdy, lê dado e ativa Ack
6. Memória vê Ack, libera linhas de dados e desativa DataRdy
7. Dispositivo vê DataRdy desativado e desativa Ack

Acessando o Barramento

- **Mestre do barramento**

  Inicia e controla todas as requisições ao barramento

- **Quem pode ser mestre?**

  Mestre Único: Processador

  - Simplicidade
  - Processador coordena todas as transações do barramento → degradação de desempenho

  Múltiplos Mestres: Processador e Dispositivos de E/S

  - Necessidade de protocolo: request, grant e bus-release
  - Necessidade de esquema de arbitragem

Executando Operações de E/S

- **Comunicação com os dispositivos de E/S:**
  - envio de comandos aos dispositivos
  - transferência de dados de/para dispositivos
  - análise do status dos dispositivos e da transmissão
- **Sistema Operacional : Interface entre usuário (programa) e dispositivo**
  - Desenvolvimento de programas é facilitado pois detalhes de baixo nível dos dispositivos são abstraídos pelo S.O

Suporte do S.O. a E/S

- **Proteção**

  garante o acesso a dispositivos/dados para os quais se tenha permissão

- **Abstração (dispositivo)**

  possui rotinas específicas com detalhes de cada dispositivo

- **Gerenciamento**

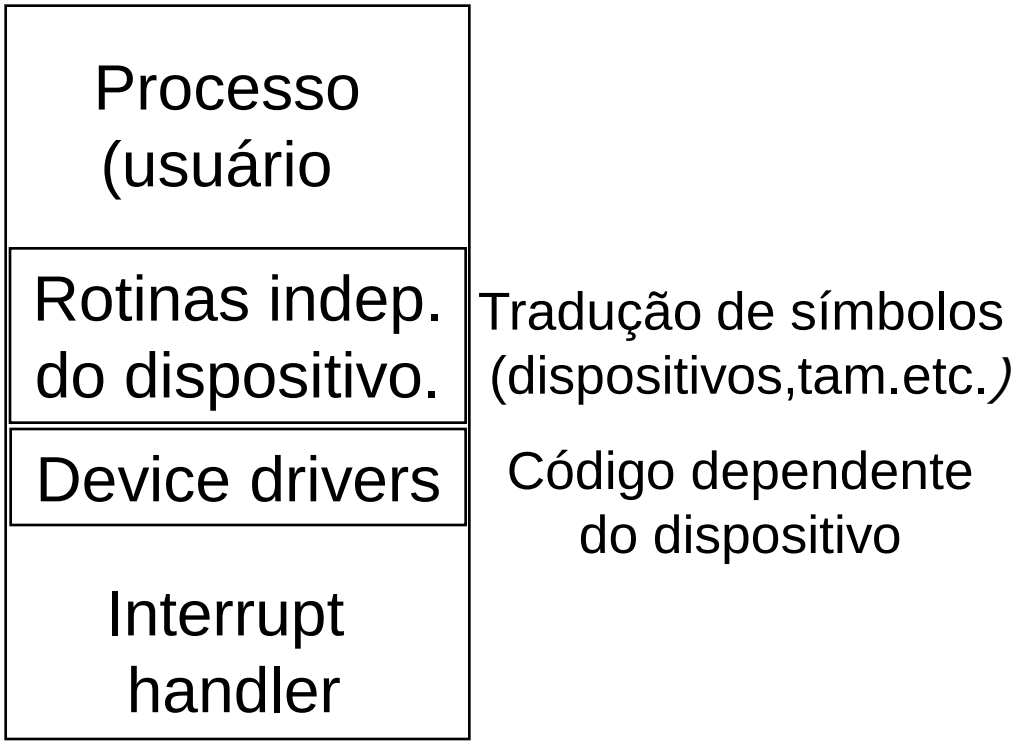
  trata as interrupções causadas pelos dispositivos

- **Escalonamento**

  controla a utilização de dispositivos compartilhados entre processos

Software de E/S

- **Organizado em camadas**



Controladores de E/S

- Dispositivos de E/S são gerenciados por controladores de HW
  - Transfere dados do/para o dispositivo
  - Sincroniza operações com software
- Controladores possuem diferentes registradores:
  - Registradores de comando
    - Faz o dispositivo começar alguma operação
  - Registradores de Status
    - Indica o que o disposivo está fazendo e a ocorrência de erros
  - Registradores de Dados
    - Escrita: transfere dados para o dispositivo
    - Leitura: transfere dados do dispositivo

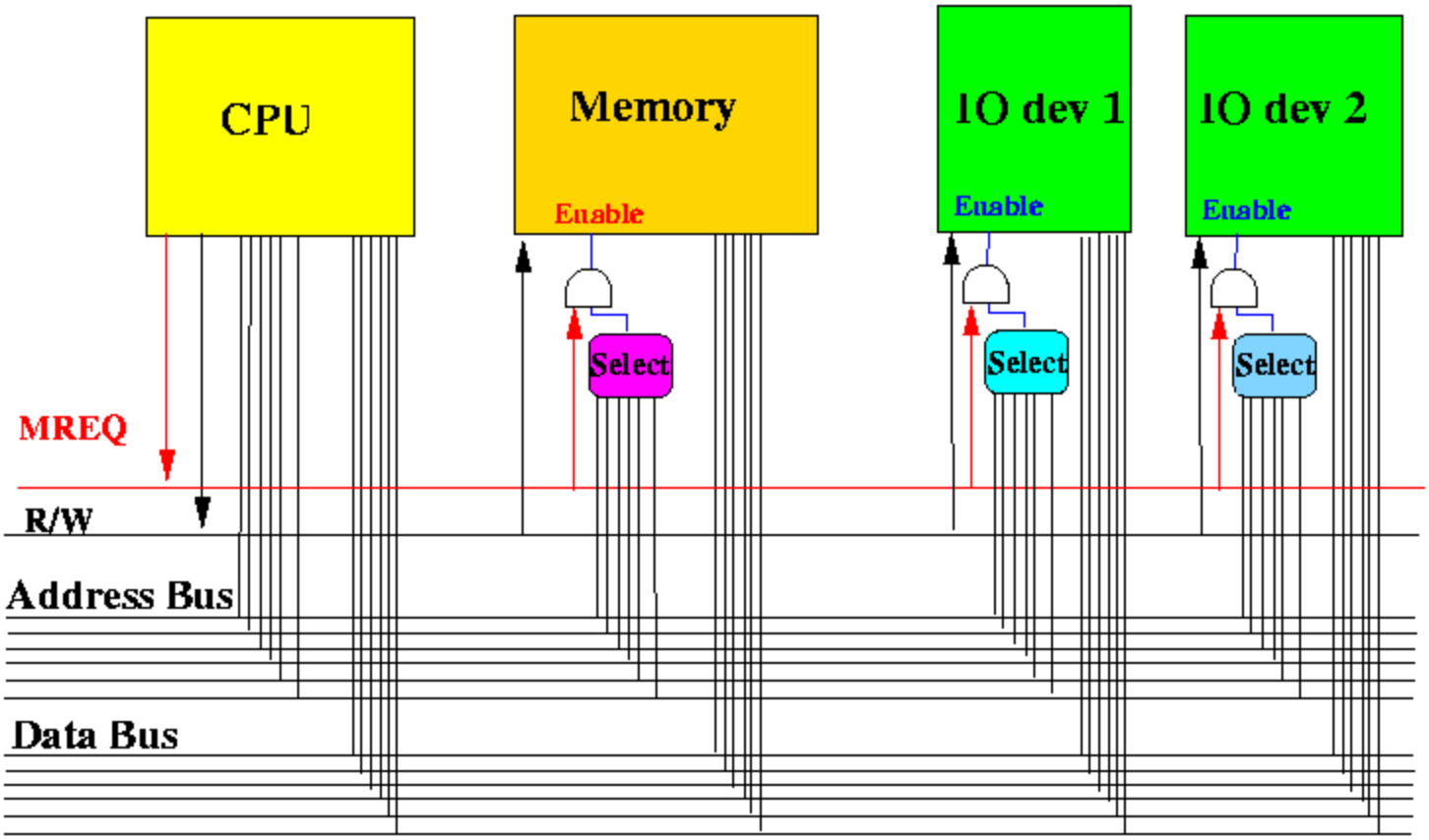
Implementando Comando de E/S

- Para enviar um comando a um dispositivo de E/S, processador deve acessar registradores do controlador
- Processador deve endereçar dispositivo e enviar comandos
- Dois métodos de endereçamento do dispositivo
  - E/S mapeada em memória
  - Instruções específicas de E/S

E/S Mapeada em Memória

- Parte do espaço de endereçamento é destinado a E/S
  - Instruções de Leitura/Escrita a endereços destinados aos dispositivo são interpretados como comandos
  - Memória ignora endereços, mas controlador de dispositivo examina endereço, grava dados nos seus registradores e manda sinal para dispositivo
- Instruções de escrita e leitura aos endereços de E/S só podem ser feitas através de S.O
- Vantagem:
  - Simplicidade de implementação
- Desvantagem:
  - Processadores que possuem espaço de endereçamento limitados são penalizados

Implementação de E/S Mapeada em Memória



A memória só é ativada para um determinado intervalo de endereços

Exemplo: E/S Mapeada em Memória no MIPS

Problema:

Suponha que um dispositivo E/S seja mapeado para endereço 0xFFFFFFF4

Escreva código para o MIPS que escreva o valor 7 para este dispositivo e depois leia a saída do dispositivo.

Solução:

```
addi $t0,$zero,7
sw $t0, 0xFFF4($zero)
lw $s0, 0xFFF4($zero)
```

Instruções Específicas de E/S

- ISA do processador contém instruções específicas para acessar dispositivos de E/S
- Instrução deve especificar o comando e o dispositivo
  - Só podem ser executadas através de S.O
- Utilizado em processadores Intel
  - Instruções **`in`** e **`out`**
  - Intel também utiliza E/S mapeada em memória
- Vantagem:
  - Permite que todo espaço de endereçamento seja utilizado para a memória
- Desvantagem:
  - ISA mais complexa de implementar

Comunicação do Dispositivo E/S para o Processador

- Dois métodos comumente utilizados
  - Polling
    - Processador periodicamente checa o status do dispositivo para determinar se dispositivo precisa de algum serviço (ou que acabou uma operação)
  - Interrupção
    - Dispositivo sinaliza processador que precisa de algum serviço (ou que acabou uma operação)
- Ambos os métodos podem ser utilizados em um sistema para diferentes tipos de dispositivos

Polling

- Dispositivo coloca informação no registrador do controlador e processador lê registrador
- Processador detém controle e faz todo o trabalho
- Utilizado em dispositivos como mouse e impressoras
- Vantagem:

  Simplicidade de implementação
- Desvantagem:

  CPU fica esperando pelo dispositivo, gastando ciclos de processamento

Comunicação com Polling

CPU

1- lê reg do status do disp.

3- Inspeciona status,
se não está pronto va para 1

4- escreve no reg. dado

6- se existe mais dados vá
para 1

DISPOSITIVO

2- envia status para reg.

5- aceita dado,
status=ocupado até escrita
terminada

Interrupções

- Uma interrupção é similar a uma exceção, porém:
  - É assíncrona em relação a execução de instruções
    - Não pára a execução da instrução
  - Pode ser tratada quando for conveniente
- Informação sobre a causa da interrupção frequentemente identifica dispositivo
- Interrupções podem ter diferentes urgências
  - Necessidade de mecanismo para atribuir prioridades

E/S Com Interrupções

- CPU executa outras instruções enquanto espera E/S
- Sincronização: interrupção

  disp. E/S está pronto para nova transferência

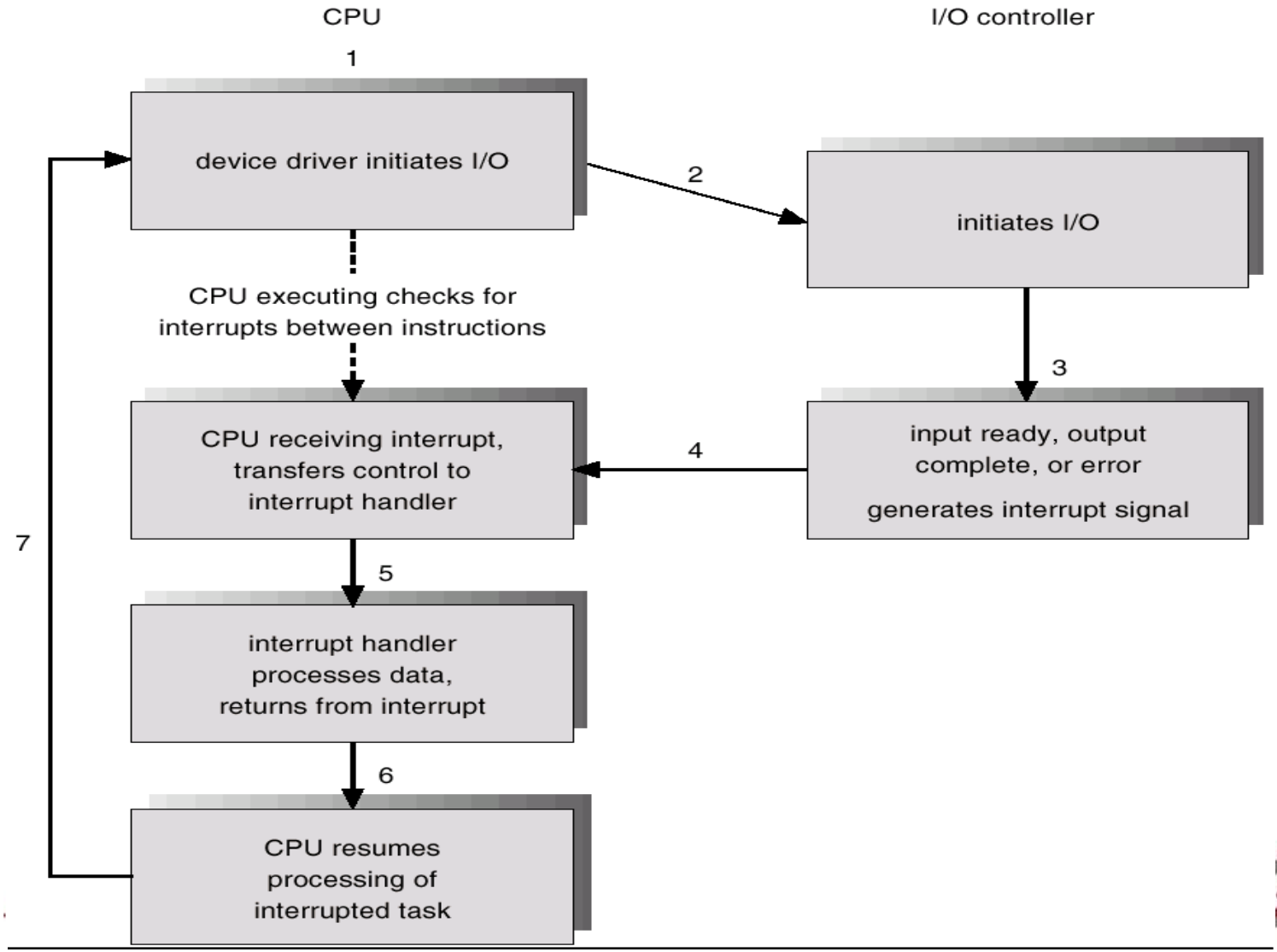
  operação de E/S terminou

  ocorreu erro
- Vantagem:

  Melhor utilização da CPU
  - Não precisa ficar verificando o status do dispositivo
  - Programa só é suspenso quando E/S interromper
- Desvantagem:

  HW especial é necessário
  - Para indicar dispositivo
  - Para salvar contexto antes de atender dispositivo

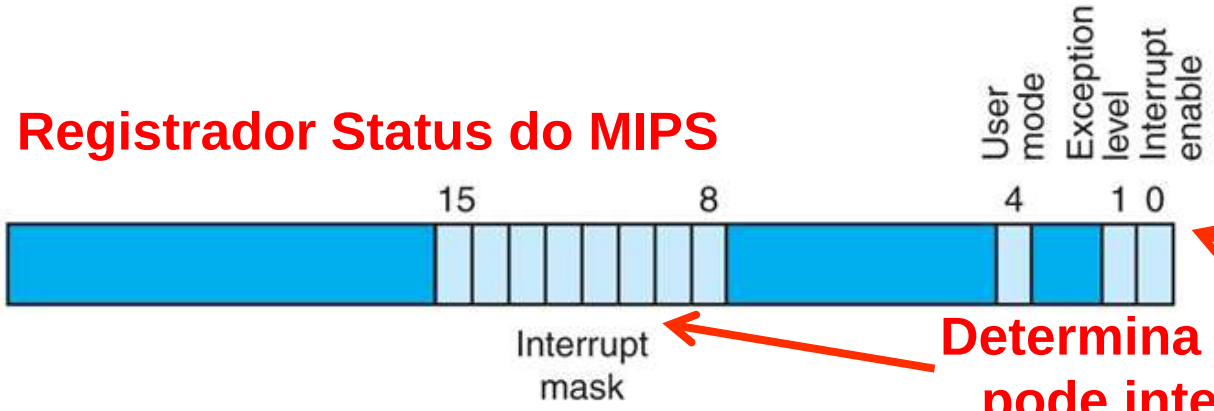
Comunicação com Interrupção



Níveis de Prioridades de Interrupções

- Níveis de prioridade podem ser usados para direcionar a ordem em que o S.O atende as interrupções

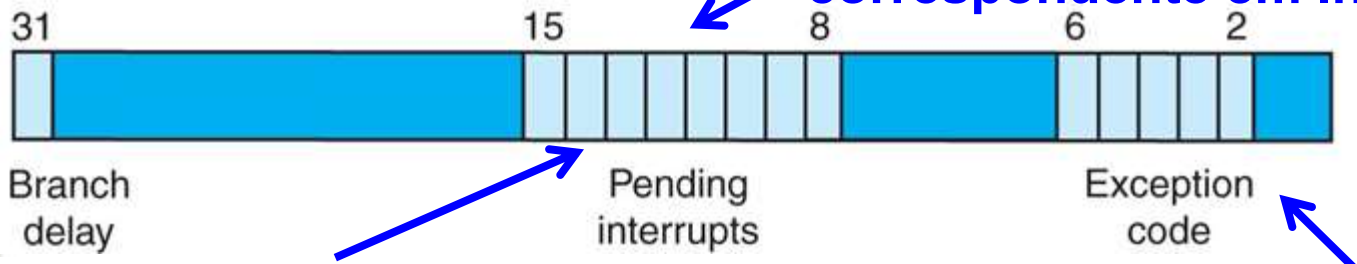
Registrador Status do MIPS



Determina que nível de prioridade pode interromper processador

Registrador Causa do MIPS

Para habilitar interrupção, bit correspondente em Interrupt mask = 1



Convenção: mais a esquerda maior prioridade

Causa da interrupção

Tranferência entre Dispositivos E/S e Memória - Direct Memory Access (DMA)

- Polling e Interrupções requerem que processador seja encarregado de transferir dados entre dispositivos e a memória
  - Gasta muitos ciclos do processador
  - Bom para quantidade de dados pequena

- **DMA** é um mecanismo que permite a transferência de blocos de dados diretamente entre dispositvos e memória

Funcionamento do DMA

- Processador fornece ao controlador de DMA:
  - Endereço do dispositivo, operação, endereço de memória e número de bytes
- Controlador de DMA gerencia a transferência
  - Quando controlador de DMA termina, interrompe o processador
- Processador pode ter que esperar enquanto memória está realizando uma transferência de DMA
  - Uso de caches evita acesso do processador à memória

Interação DMA-Cache

- Se DMA escreve para um bloco de memória que está na cache
  - Cópia da cache se torna desatualizada (stale)
- Se cache com write-back tiver bloco “sujo” e DMA lê o bloco da memória
  - DMA lê dados desatualizados (stale)
- Necessário assegurar coerência da cache
  - Flushing
    - Invalidar blocos da cache em uma leitura de E/S
    - Forçar escrita do bloco da cache na memória em uma escrita de E/S