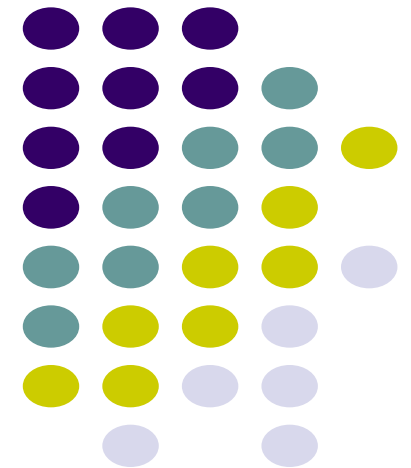
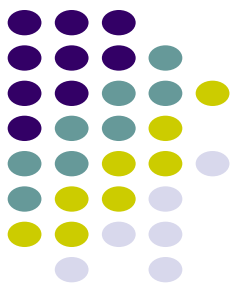


Arquitetura e Organização de Computadores

Armazenamento e E/S Parte II

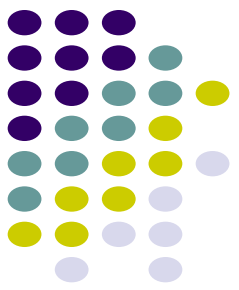
Prof. Sílvio Fernandes





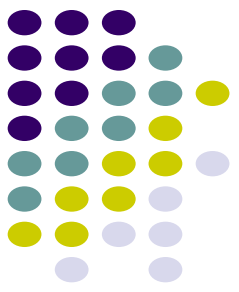
Conectando processadores, memória e E/S

- Barramentos
 - Os diversos subsistemas precisam ter interfaces e precisam se comunicar
 - Durante anos isso tem sido feito com um *barramento*
 - É um link de comunicação compartilhado, que utiliza um conjunto de fios paralelos
 - Vantagens
 - Versatilidade
 - Baixo custo
 - Desvantagem
 - Gargalo de comunicação



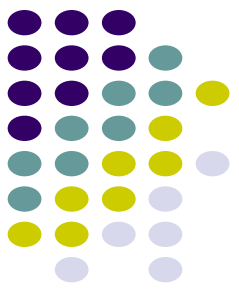
Conectando processadores, memória e E/S

- Barramentos
 - A largura de banda limita a vazão máxima de E/S
 - Barramentos tradicionais
 - Barramentos processador-memória
 - Curtos e de alta velocidade
 - Barramentos de E/S
 - Podem ser extensos
 - Podem ter muitos dispositivos conectados
 - Grande faixa de largura de banda
 - Normalmente não realizam interface direta com a memória
 - Utilizam barramento processador-memória
 - ou backplane: permite conexão processador, memória e E/S



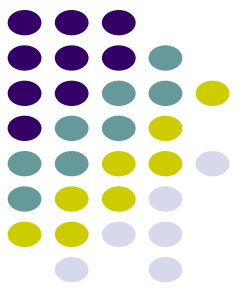
Conectando processadores, memória e E/S

- Barramentos
 - Fatores físicos limitam a velocidade
 - Extensão do barramento
 - Número de dispositivos nele conectado
 - Os barramentos estão sendo substituídos por interconexões ponto a ponto de alta velocidade com switches: redes de interconexão ou redes em chip



Conectando processadores, memória e E/S

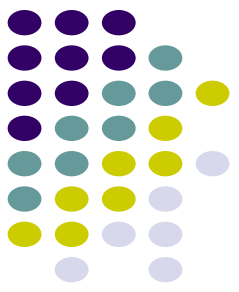
- Fundamentos sobre conexão
 - Uma **transação de E/S** corresponde a enviar o endereço e receber ou enviar os dados
 - As transações podem ser de *entrada* e *saída* do ponto de vista do processador
 - Entrada: entrar dados do dispositivo para memória para leitura do processador
 - Saída: sai dados para um dispositivo a partir da memória, na qual o processador escreveu
 - A interconexão de E/S serve como um modo de expandir a máquina e conectar novos periféricos



Conectando processadores, memória e E/S

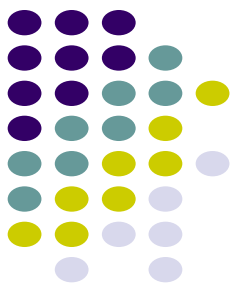
- Fundamentos sobre conexão
 - Padrões de E/S dominantes

| Characteristic | Firewire (1394) | 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 |
| Basic data width (signals) | 4 | 2 | 2 per lane | 4 | 4 |
| Theoretical peak bandwidth | 50 MB/sec (Firewire 400) or 100 MB/sec (Firewire 800) | 0.2 MB/sec (low speed), 1.5 MB/sec (full speed), or 60 MB/sec (high speed) | 250 MB/sec per lane (1x); PCIe cards come as 1x, 2x, 4x, 8x, 16x, or 32x | 300 MB/sec | 300 MB/sec |
| Hot pluggable | Yes | Yes | Depends on form factor | Yes | Yes |
| Maximum bus length (copper wire) | 4.5 meters | 5 meters | 0.5 meters | 1 meter | 8 meters |
| Standard name | IEEE 1394, 1394b | USB Implementors Forum | PCI-SIG | SATA-IO | T10 committee |



Conectando processadores, memória e E/S

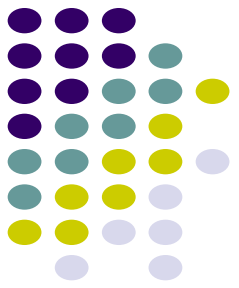
- Fundamentos sobre conexão
 - Barramentos tradicionais são **síncronos**
 - Inclui um *clock* que serve de referência nas transmissões
 - O dispositivo precisa executar na velocidade do clock
 - Os barramentos não podem ser longos se forem rápidos devido a problemas de variação de clock
 - Interconexões **assíncronas**
 - Não utilizam clock
 - Os barramentos podem ser estendidos sem problemas com variação do clock ou sincronismo



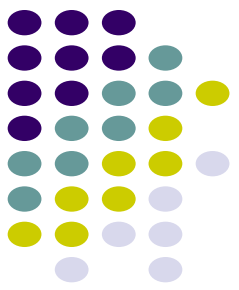
Conectando processadores, memória e E/S

- Fundamentos sobre conexão
 - Interconexões **assíncronas**
 - Utilizam um protocolo de transmissão
 - **Handshaking**: uma série de etapas usadas para coordenar as transferências que só acontecem se emissor e receptor concordarem que a etapa atual foi concluída

Interface de E/S com processador, memória e SO



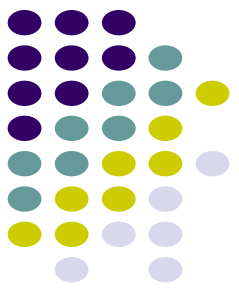
- Como uma solicitação de E/S de um usuário é transformada em um comando de dispositivo e comunicada ao dispositivo?
- Como os dados são realmente transferidos de ou para um local de memória?
- Qual é o papel do sistema operacional?



Interface de E/S com processador, memória e SO

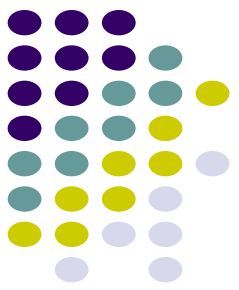
- Sistema operacional (SO)
 - Atua como interface entre o HW e o programa que solicita a E/S
 - Responsabilidades do SO surgem de
 1. Diversos programas usando o processador compartilham o sistema de E/S
 2. Os sist. E/S normalmente usam interrupções para comunicar informações sobre E/S – causam transferência ao modo kernel
 3. Controle de baixo nível de um dispositivo de E/S é complexo, exige o gerenciamento de um conjunto de eventos simultâneos

Interface de E/S com processador, memória e SO



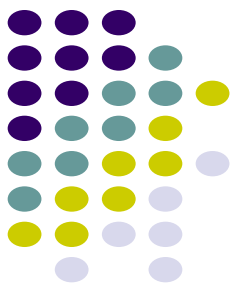
- Dando comando a dispositivos de E/S
 - O processador deve ser capaz de endereçar o dispositivo e fornecer uma ou mais palavras de comando
 - Dois métodos de endereçar os dispositivos
 - **E/S mapeada em memória:** parte do espaço de endereçamento é atribuída a dispositivos de E/S e a memória ignora-os. Endereços são interpretados como comandos aos dispositivos
 - **Instruções de E/S:** podem especificar o número do dispositivo e a palavra comando (ou local de comando na memória)

Interface de E/S com processador, memória e SO



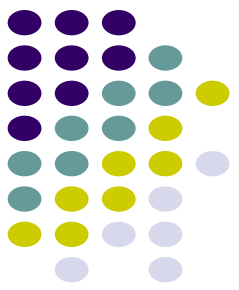
- Dando comando a dispositivos de E/S
 - Uma leitura ou escrita de dados para cumprir uma solicitação do programa normalmente exige várias operações de E/S separadas
 - O processador pode interrogar o status do dispositivo entre comandos individuais para determinar se o comando foi concluído com sucesso

Interface de E/S com processador, memória e SO

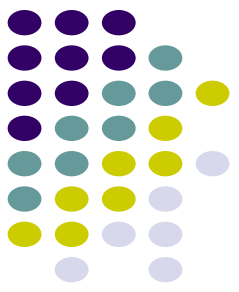


- Comunicação com o processador
 - O processo de verificar periodicamente os bits de status de um dispositivo de E/S para determinar a necessidade de atendê-lo é chamado de **polling**
 - É a forma mais simples do dispositivo se comunicar com o processador
 - O processador está no controle e realiza todo trabalho
 - É uma técnica boa para aplicações embutidas de tempo real onde o tempo de resposta é predeterminado
 - O overhead do polling levou a invenção da **E/S controlada por interrupção**

Interface de E/S com processador, memória e SO



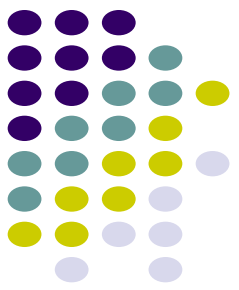
- Comunicação com o processador
 - E/S controlada por interrupção
 - Usada em quase todos os sistemas
 - O dispositivo notifica o processador quando completou alguma operação ou que precisa de atenção
 - O processador é interrompido, assim como nas exceções, entretanto
 - Uma interrupção de E/S é assíncrona com relação à execução da instrução (nada impede o término da instrução)
 - É importante transmitir informações adicionais como a identidade do dispositivo gerando a interrupção que podem ter diferentes prioridades



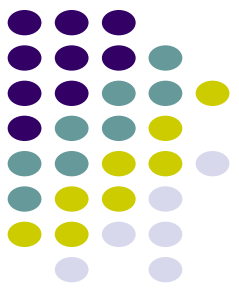
Interface de E/S com processador, memória e SO

- Comunicação com o processador
 - Para comunicar informações ao processador
 - Interrupções vetorizadas
 - O dispositivo envia o endereço do vetor
 - Registrador de causa
 - O dispositivo coloca sua identidade no campo do registrador de causa
 - Como resultado o SO adquire o controle e sabe a identidade do dispositivo e pode interrogar imediatamente o dispositivo

Interface de E/S com processador, memória e SO

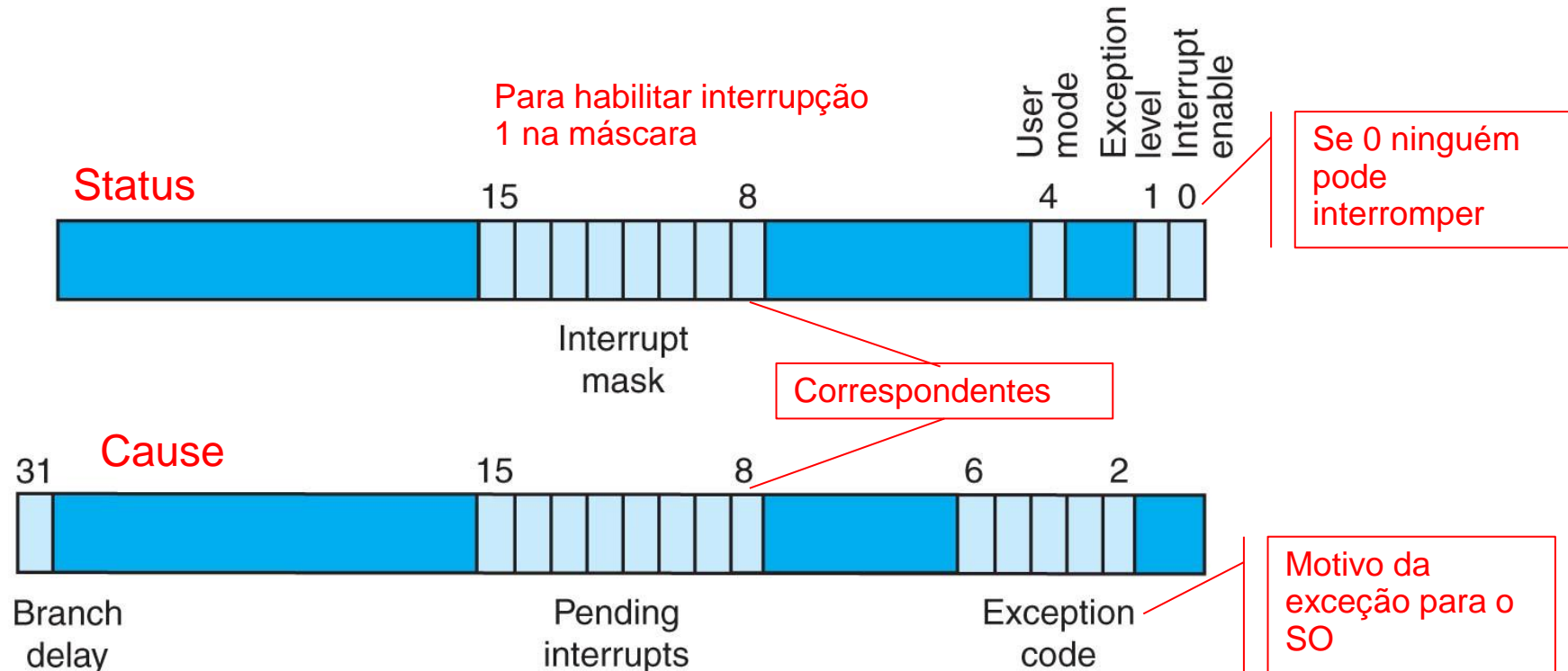


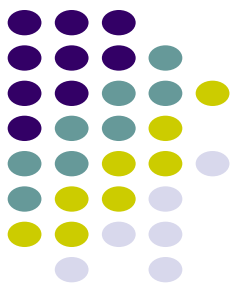
- Níveis de prioridade de interrupção
 - Indicam a ordem em que o processador deverá processar interrupções
 - Exceções gerada internamente e interrupções de E/S externas possuem prioridades
 - Em geral as de E/S possuem prioridades menor do que as exceções



Interface de E/S com processador, memória e SO

- Níveis de prioridade de interrupção
 - O MIPS oferece as primitivas que deixam o SO implementar a política usando os registradores Cause e Status

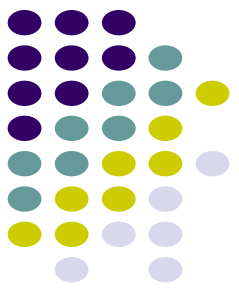




Interface de E/S com processador, memória e SO

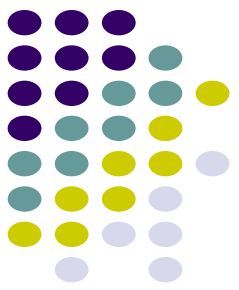
- Níveis de prioridade de interrupção
 - Tratamento de uma exceção
 1. Realize um AND lógico entre o campo interrupções pendentes e o campo máscara de interrupções para ver quais poderiam ser as culpadas
 2. Selecione a prioridade mais alta (mais à esquerda)
 3. Salve o campo de máscara de interrupções
 4. Mude o campo de máscara para desativar todas as interrupções de prioridade igual ou inferior
 5. Salve o estado do processador necessário para lidar com as interrupções
 6. A fim de permitir interrupções de prioridade mais alta, coloque o bit interrupções habilitadas do reg. Cause em 1

Interface de E/S com processador, memória e SO



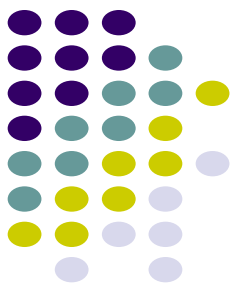
- Níveis de prioridade de interrupção
 - Tratamento de uma exceção
 7. Chame a rotina de interrupção
 8. Antes de restaurar o estado, coloque o bit interrupções habilitadas do reg. Cause em 0. Isso permite restaurar o campo de máscara de interrupções
 - O Nível de Prioridade de Interrupção ou IPL (*Interrupt Priority Levels*) é uma invenção do SO, armazenado na memória do processo, para bloquear ou desbloquear interrupções através do campo de máscara

Interface de E/S com processador, memória e SO



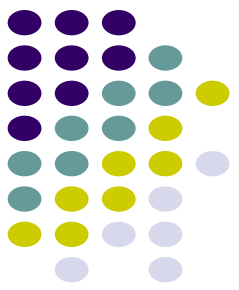
- Transferindo os dados entre dispositivo e mem.
 - *Polling* e interrupções de E/S são a base para 2 métodos de implementação da transferência de dados
 - No *polling* usamos o processador para transferir dados entre um dispositivo e a memória, o qual também verifica constantemente o status da transferência
 - Na interrupção de E/S o SO assume um papel importante de colocar o processador para realizar a transferência mas enquanto a operação de E/S está em curso outro programa pode usar o processador

Interface de E/S com processador, memória e SO



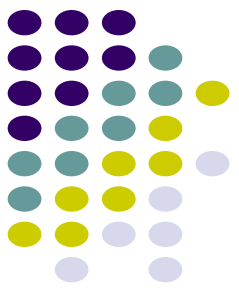
- Transferindo os dados entre dispositivo e mem.
 - Mesmo usando interrupções de E/S durante a transferência acontecem várias operações de E/S e, conseqüentemente, várias interrupções ao processador
 - Para desafogar o processador inventou-se o mecanismo de **acesso direto à memória** ou DMA (*Direct Memory Access*)
 - A interrupção ainda é usada mas somente quando a transferência é concluída ou quando ocorrer um erro
 - O DMA é implementado por um controlador especializado que transfere dados entre o dispositivo e a memória, independente do processador

Interface de E/S com processador, memória e SO



- Transferindo os dados entre dispositivo e mem.
 - Existem 3 etapas em transferência DMA
 1. O processador configura o DMA fornecendo a identidade do dispositivo, a operação a realizar, o endereço de memória (para ler ou escrever) e o número de bytes a transferir
 2. O DMA inicia a operação, fornecendo o endereço de memória (para leitura ou escrita) e na conclusão gerando o próximo endereço
 3. Quando a transferência termina, o controlador interrompe o processador que pode interrogar o DMA ou verificar a memória se a operação inteira foi concluída

Interface de E/S com processador, memória e SO



- Transferindo os dados entre dispositivo e mem.
 - Com DMA os acessos a memória não são mais restritos apenas ao processador
 - Isso gera problemas em sistema de memória virtual e sistema de caches
 - Esse problemas normalmente são solucionados com uma combinação de técnicas de HW e suporte de SW

Referências

- PATTERSON, D. A. ; HENNESSY, J.L. Organização e projeto de computadores – a interface hardware software. 4. ed. Editora Campus, 2014.

