

Arquitetura e Organização de Computadores

Capítulo 3

Visão de alto nível da função e interconexão do computador

slide 1

© 2010 Pearson Prentice Hall. Todos os direitos reservados.

O que é um programa?

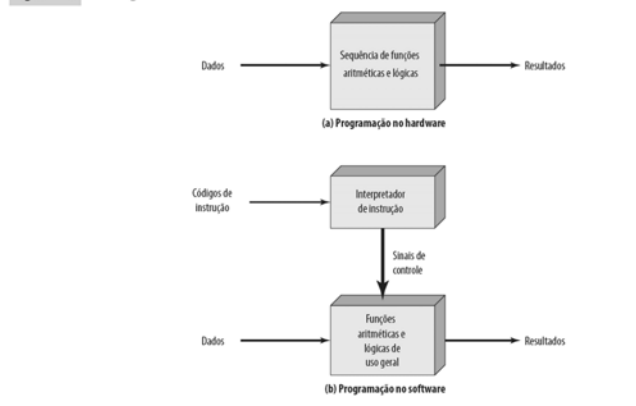
- Uma **sequência de etapas**.
- Para cada etapa, é feita uma **operação** aritmética ou lógica.
- Para cada operação, é necessário um conjunto diferente de **sinais de controle**.

Função da unidade de controle

- Para cada operação, um **código** exclusivo é fornecido.
—P.e. ADD, MOVE.
- Um segmento de **hardware aceita o código e emite os sinais de controle**.
- Temos um computador!

Hardware x software

Figura 3.1 Abordagens de hardware e software



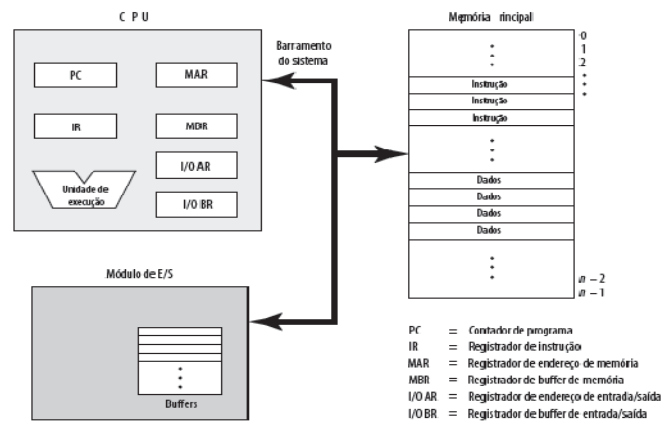
Conceito de programa

- Sistemas “hardwired” são inflexíveis.
- Hardware de uso geral pode fazer diferentes tarefas, dependendo dos dados e dos sinais de controle corretos.
- Ao invés de “religar” o hardware, forneça um conjunto de sinais de controle.

Componentes

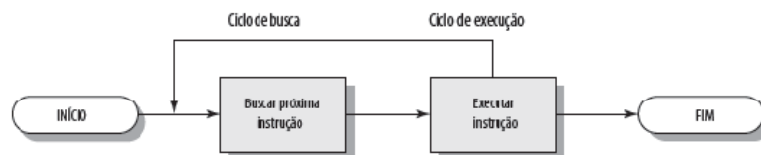
- A Unidade de Controle e a Unidade Lógica e Aritmética constituem a **Unidade Central de Processamento**.
- Dados e instruções precisam entrar no sistema, e resultados saem dele.
 - Entrada/saída**.
- É necessário um armazenamento temporário de código e resultados.
 - Memória principal**.

Componentes do computador: visão de alto nível



Ciclo de instrução básico

- Duas etapas:
 - Busca
 - Execução



Ciclo de busca

- Contador de Programa (PC) mantém endereço da próxima instrução a ser buscada.
- Processador busca instrução do local de memória apontado pelo PC.
- Incrementar PC:
 - A menos que seja informado de outra forma.
- Instrução carregada no Registrador de Instrução (IR).
- Processador interpreta instrução e realiza **ações** exigidas.

Ciclo de execução (ações)

- Processador-memória:
 - Transferência de dados entre CPU e memória principal.
- E/S do processador:
 - Transferência de dados entre CPU e módulo de E/S.
- Processamento de dados:
 - Alguma operação aritmética ou lógica sobre dados.
- Controle:
 - Alteração da sequência de operações.
 - P.e. salto.
- Combinação dos itens anteriores.

Ciclo de instrução (outra visão)

1. BUSCA

- a) Envio de um endereço para a memória e execução de uma leitura;
- b) Incremento do PC;

2. EXECUÇÃO

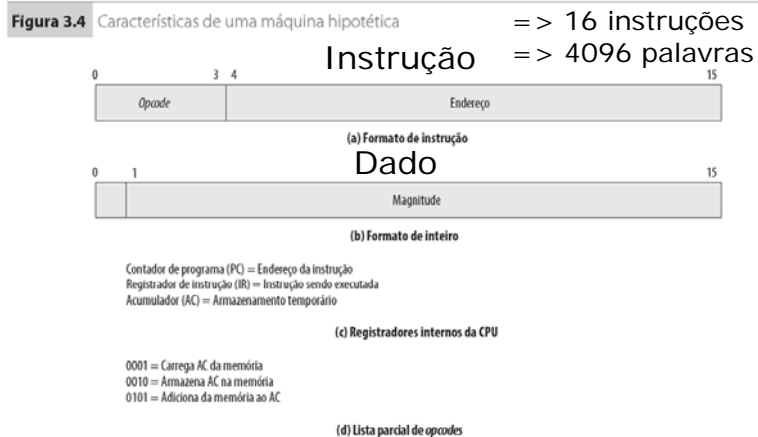
- a) Decodificação do código de operação;
- b) Execução da instrução;

3. VOLTA PARA FASE 1

Pode haver desvio:

- Incondicional: o valor do PC é alterado;
- Condicional: se condição satisfeita, a sequência linear é interrompida.

Exemplo de execução de programa



Exemplo de execução de programa

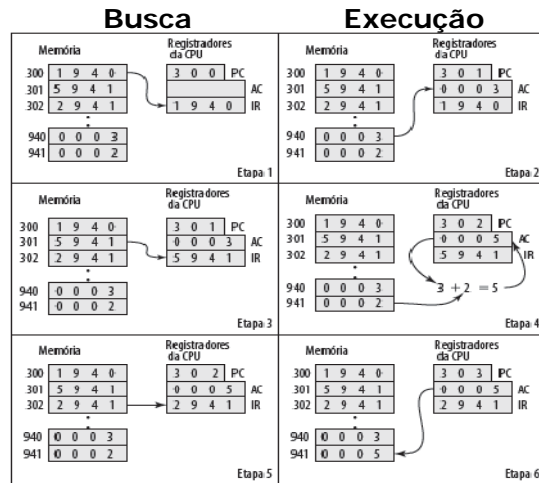
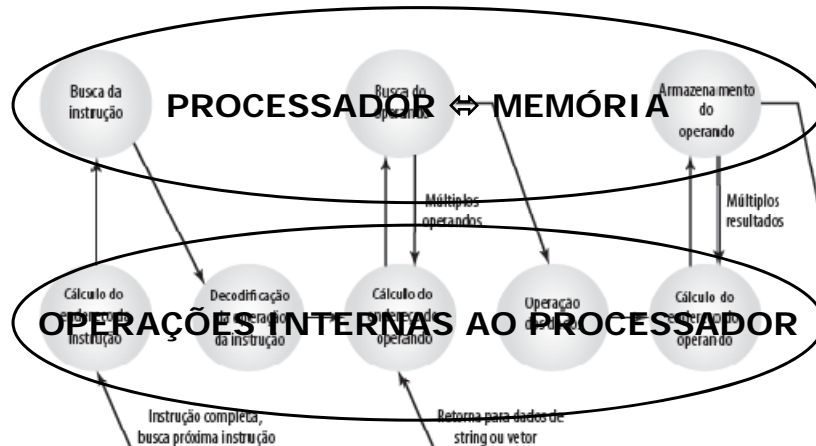


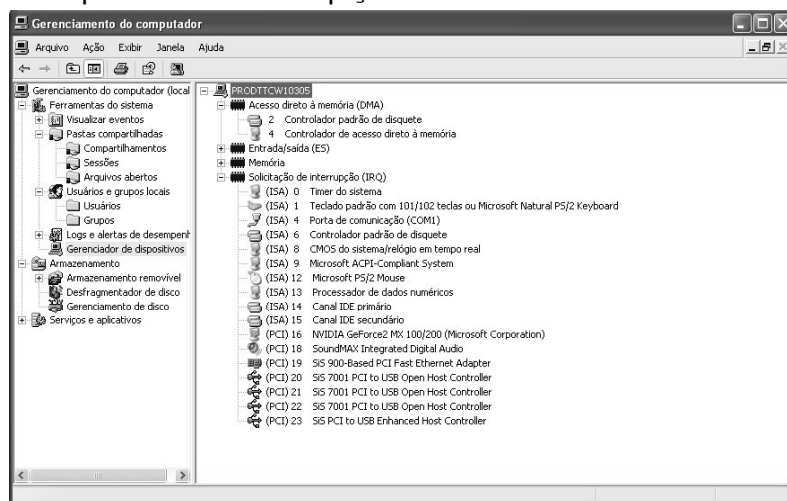
Diagrama de estado do ciclo de instrução



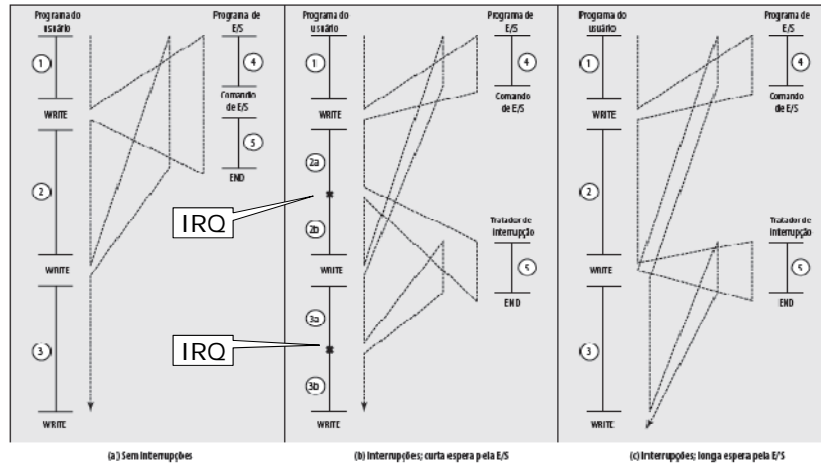
Interrupções

- Mecanismo pelo qual outros módulos (p.e. E/S) podem interromper a sequência de processamento normal.
- Classes de Interrupção
 - Programa:
 - P.e. estouro, divisão por zero.
 - Timer:
 - Gerado por timer dentro do processo.
 - Usado na multitarefa preemptiva.
 - E/S:
 - Do controlador de E/S.
 - Falha de hardware:
 - P.e. erro de paridade de memória.

Exemplo de Interrupções em micros



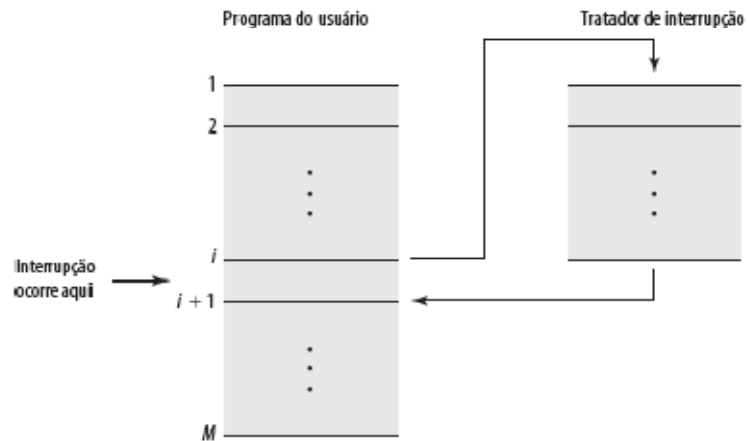
Controle de fluxo de programa



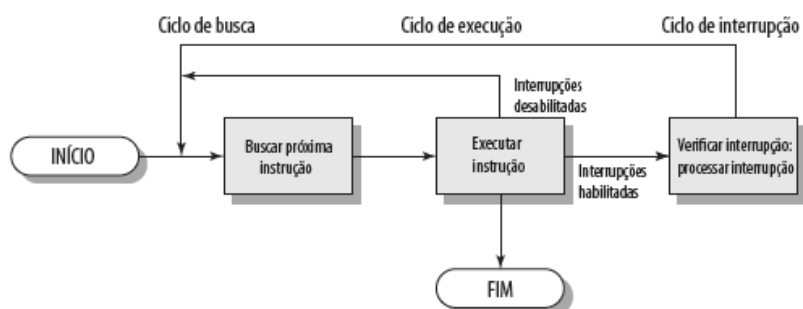
Ciclo de interrupção

- Adicionado ao ciclo de instrução.
- Processador verifica interrupção.
 - Indicado por um sinal de interrupção.
- Se não houver interrupção, busca próxima instrução.
- Se houver interrupção pendente:
 - Suspende execução do programa atual.
 - Salva contexto.
 - Define PC para endereço inicial da rotina de tratamento de interrupção.
 - Trata a Interrupção
 - Restaura contexto e continua programa interrompido.

Transferência de controle via interrupções

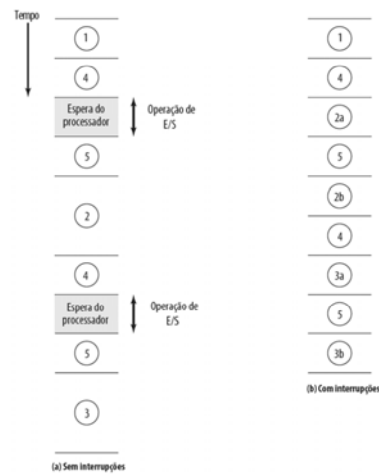


Ciclo de instrução com interrupções



Sincronização do programa – espera curta pela E/S

Figura 3.10 Sincronização do programa: espera curta pela E/S



Sincronização do programa – espera longa pela E/S

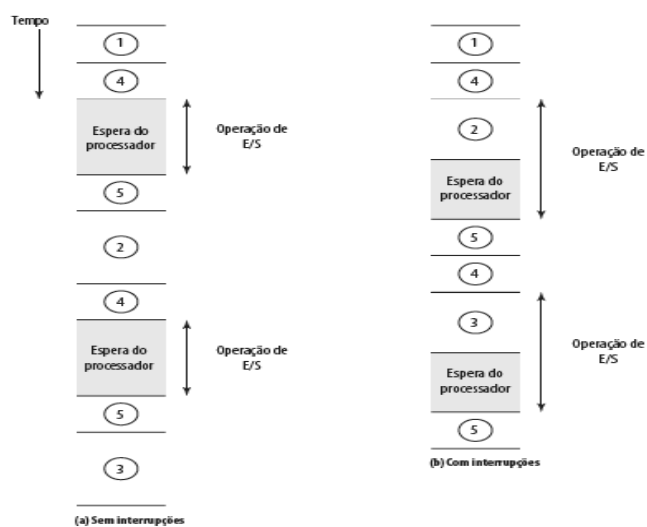
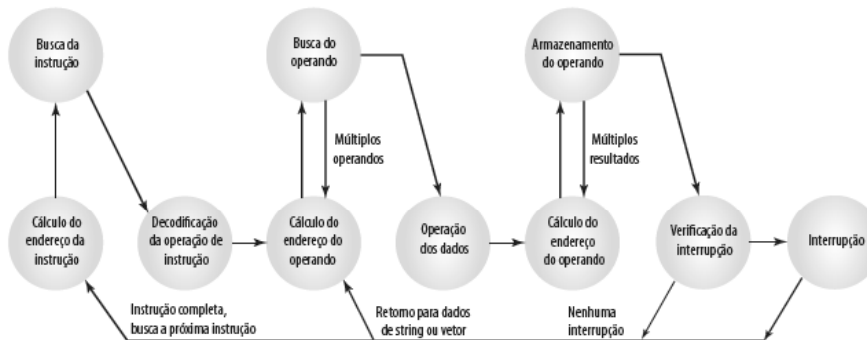


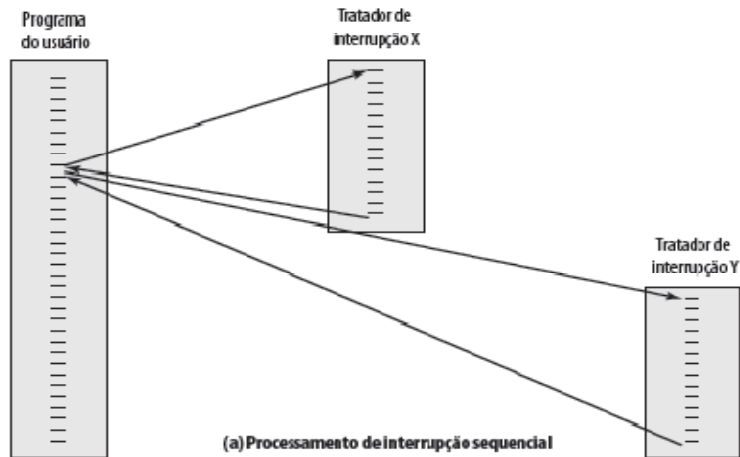
Diagrama de estado do ciclo de instrução (com interrupções)



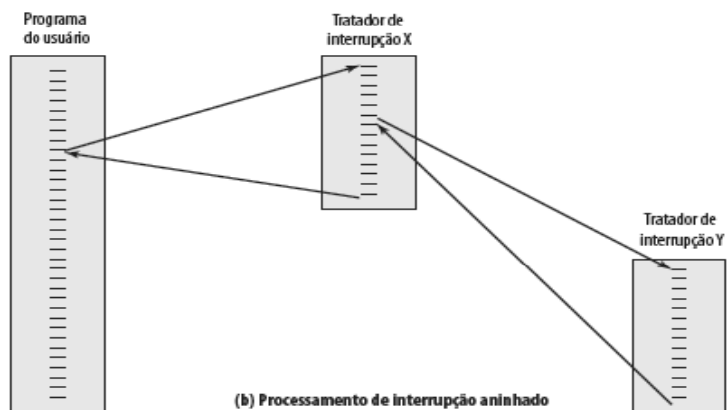
Técnicas para Interrupções múltiplas

- Desativar interrupções:
 - Processador ignorará outras interrupções enquanto processa uma interrupção.
 - Interrupções permanecem pendentes e são verificadas após primeira interrupção ter sido processada.
 - Interrupções tratadas em sequência enquanto ocorrem.
- Ativar interrupções, mas definir prioridades:
 - Interrupções de baixa prioridade podem ser interrompidas por interrupções de prioridade mais alta.
 - Quando interrupção de maior prioridade tiver sido processada, processador retorna à interrupção anterior.

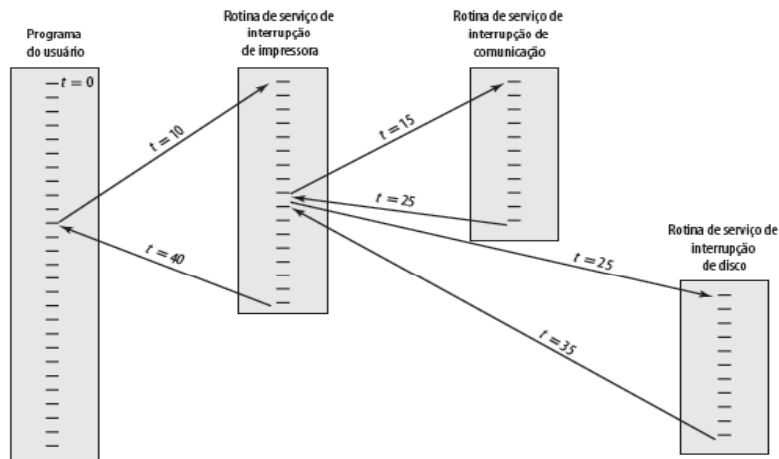
Interrupções múltiplas - sequenciais



interrupções múltiplas - aninhadas



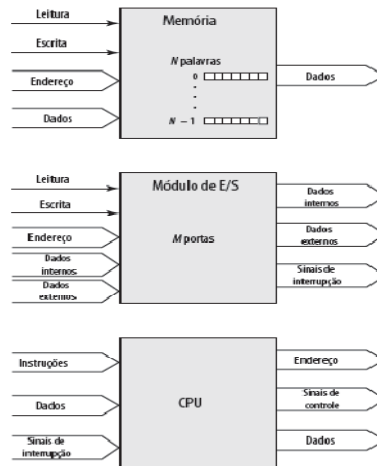
Sequência de tempo de interrupções múltiplas com prioridade



Estrutura de Interconexão

- Todas as unidades devem ser conectadas.
- Um tipo de conexão diferente para cada tipo de unidade diferente.
 - Memória
 - Entrada/Saída
 - CPU

Módulos do computador



Conexão de memória

- Recebe e envia dados.
- Recebe endereços (de locais).
- Recebe sinais de controle:
 - Leitura.
 - Escrita.
 - Temporização.

Conexão de entrada/saída

- Semelhante à memória do ponto de vista do computador.
- Saída:
 - Recebe dados do computador.
 - Envia dados a periféricos.
- Entrada:
 - Recebe dados de periféricos.
 - Envia dados ao computador.

Conexão de entrada/saída (cont.)

- Recebe sinais de controle do computador.
- Envia sinais de controle aos periféricos.
 - P.e., girar disco.
- Recebe endereços do computador.
 - P.e., número de porta para identificar periférico.
- Envia sinais de interrupção (controle).

Conexão da CPU

- Lê instruções e dados.
- Escreve dados (após processamento).
- Envia sinais de controle a outras unidades.
- Recebe (e atua sobre) interrupções.

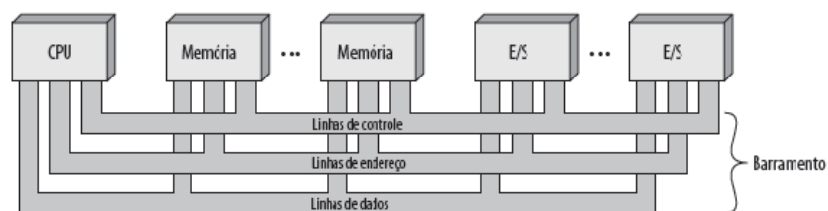
Interconexão de barramento

- Existem diversos sistemas de interconexão possíveis.
- Estrutura de barramento único e múltiplo são mais comuns.
- P.e., barramento de Controle/Endereço/Dados (PC).
- P.e., Unibus (DEC-PDP).

O que é um barramento?

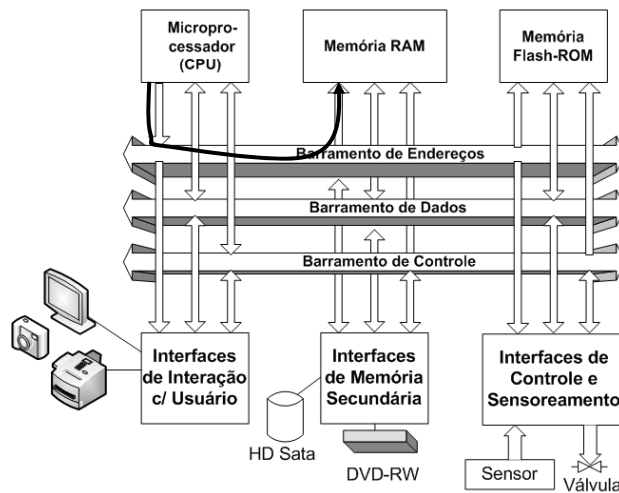
- É um caminho de comunicação conectando dois ou mais dispositivos.
- Normalmente, é um meio de transmissão compartilhado.
- Frequentemente agrupado.
 - Uma série de canais em um barramento.
 - P.e., barramento de dados de 32 bits são 32 canais de bits separados.
- Um barramento que conecta CPU, memória e E/S é chamado de **Barramento de Sistema**.

Esquema de interconexão de barramento



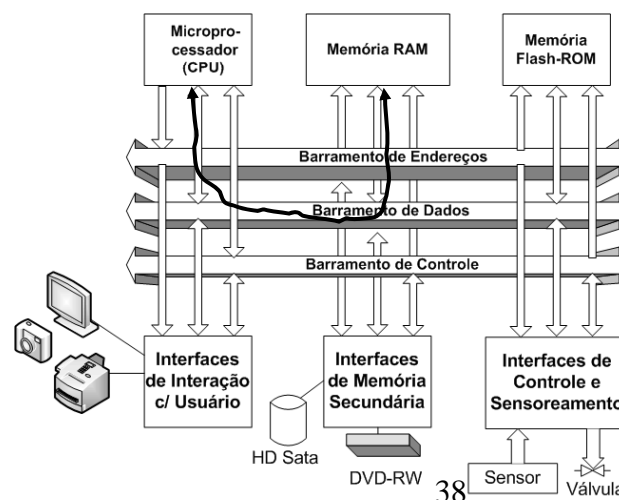
Esquema de interconexão de barramento

Barramento de
Endereços
=
Caminho de
Comunicação !

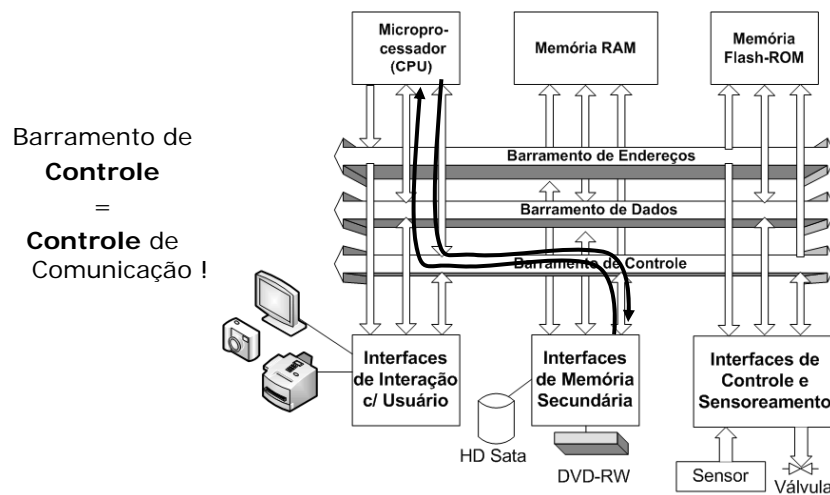


Esquema de interconexão de barramento

Barramento de
Dados
=
Meio de
Comunicação !



Esquema de interconexão de barramento



Barramento de dados

- Transporta dados.
 - Lembre-se de que não existe diferença entre “dados” e “instruções” neste nível.
- Largura é um fator fundamental do desempenho.
 - 8, 16, 32, 64 bits.

Barramento de endereço

- Identifica origem ou destino dos dados.
- P.e., CPU precisa ler uma instrução (dados) de determinado local na memória.
- Largura do barramento determina capacidade máxima da memória do sistema.
 - P.e., 8080 tem barramento de endereço de 16 bits, gerando espaço de endereços de 64K.
 - P.e., Pentium Pro tem barramento de endereço de 32 bits, gerando espaço de endereços de 4GB.
 - Hoje são 40 bits de endereço = 1 TB

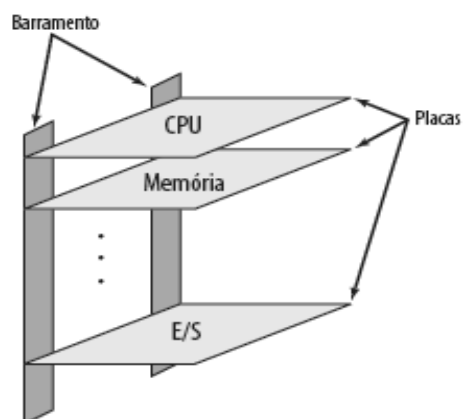
Barramento de controle

- Informação de controle e temporização:
 - Sinal de leitura/escrita de memória
 - Sinal de leitura/escrita de E/S
 - Solicitação de barramento (Bus Request)
 - Concessão de barramento (Bus Grant)
 - Solicitação de interrupção (Interrupt Request)
 - ACK de Interrupção (Interrupt Acknowledgement)
 - Sinais de clock
 - Sinal de Reset

Como os barramentos se parecem?

- Linhas paralelas em placas de circuito.
- Cabos de fita.
- Conectores em tira nas placas mãe.
—P.e., PCI.
- Conjuntos de fios.

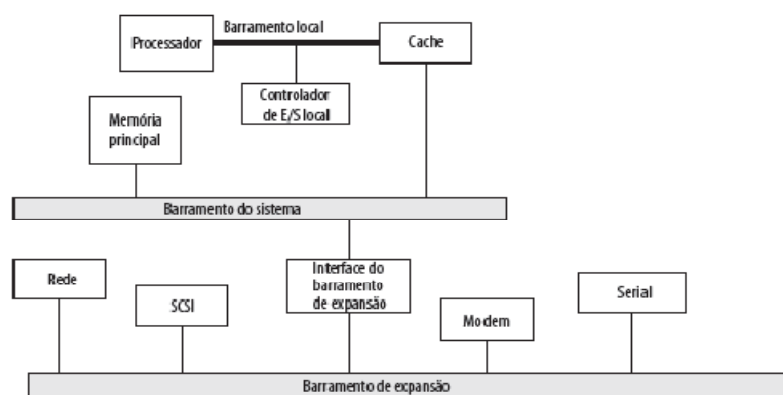
Realização física da arquitetura de barramento



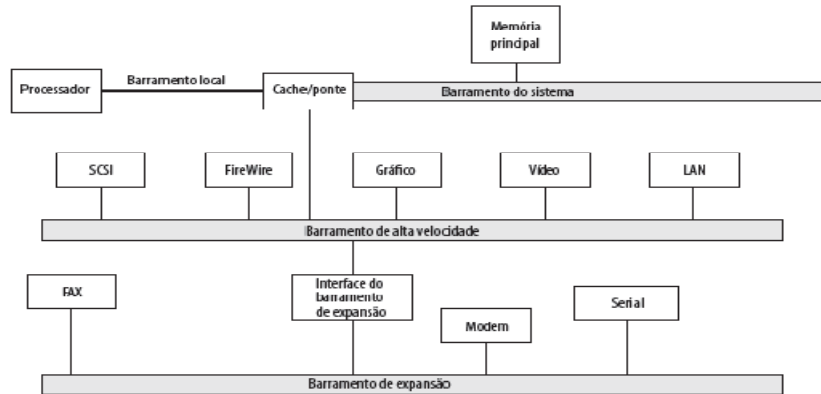
Problemas do barramento único

- Muitos dispositivos em um barramento levam a:
 - Atrasos de propagação
 - Longos caminhos de dados significa que a coordenação do uso do barramento pode afetar contrariamente o desempenho.
 - Se a demanda de transferência de dados agregada se aproxima da capacidade do barramento.
- A maioria dos sistemas utiliza **múltiplos barramentos** para contornar esses problemas.

Estrutura de barramento tradicional (ISA) (com cache)



Arquitetura de alto desempenho



Dispositivos de alta demanda integrados mais perto do processador

Tipos de barramento

- Dedicado:
 - Linhas separadas para dados e endereço.
- Multiplexado.
 - Linhas compartilhadas.
 - Linha de controle válida de endereço ou dados.
 - Vantagem – menos linhas, menos custo
 - Desvantagens:
 - Controle mais complexo.
 - Desempenho reduzido.

Arbitração de barramento

- Mais de um módulo controlando o barramento.
- P.e., CPU e controlador de DMA.
- Apenas um módulo pode controlar barramento de uma só vez.
- Arbitração pode ser centralizada ou distribuída.
- Finalidade é designar um dispositivo como Mestre (processador ou E/S), o outro será Escravo. O Mestre inicia uma transferência de dados

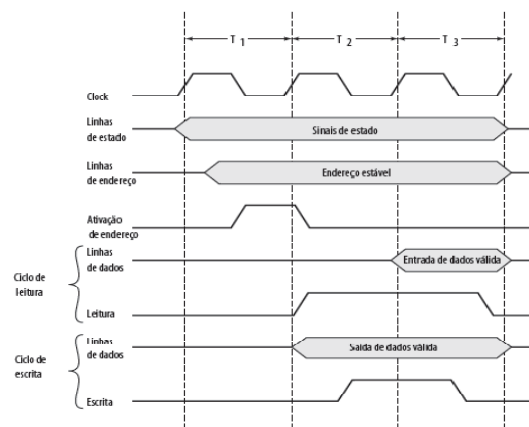
Arbitração centralizada e distribuída

- Centralizada:
 - Único dispositivo de hardware controlando o acesso ao barramento.
 - Controlador de barramento.
 - Também chamado de Árbitro.
 - Pode ser parte da CPU ou separada.
- Distribuída:
 - Cada módulo pode reivindicar o barramento.
 - Lógica de controle em todos os módulos.

Temporização

- Coordenação de eventos no barramento.
- Síncrona:
 - Eventos determinados por sinais de clock.
 - Barramento de controle inclui linha de clock.
 - Uma única transmissão 1-0 é um ciclo do barramento ou **ciclo de clock**.
 - Todos os dispositivos podem ler linha de clock.
 - Normalmente, sincronismo na borda inicial.
 - Geralmente, um único ciclo para um evento.

Diagrama de temporização síncrona



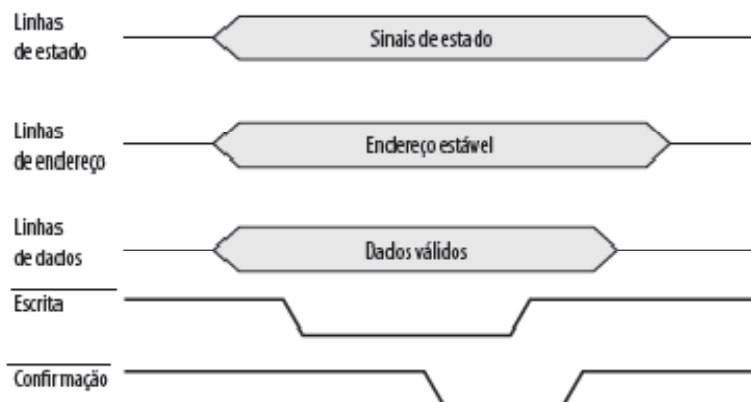
Ocorrência de um evento é determinada por um clock

Temporização assíncrona – diagrama de leitura



Ocorrência de um evento segue e depende de evento anterior

Temporização assíncrona – diagrama de escrita

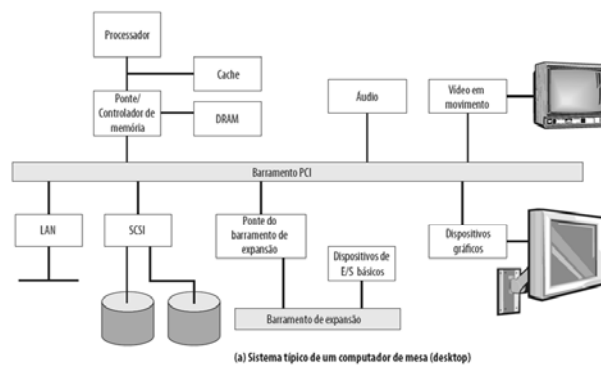


Barramento PCI

- PCI (*Peripheral Component Interconnect*)
- Interconexão de componente periférico.
- Intel lançou em 1993 para domínio público.
- 32 ou 64 bits.
- 50 linhas.

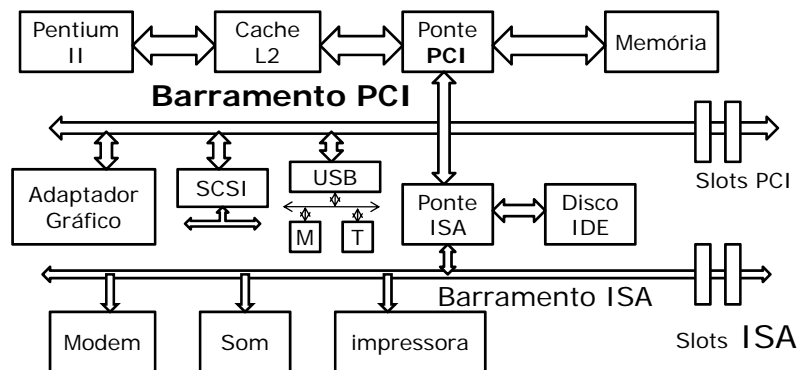
PCI para sistema de processador único

Figura 3.22 Exemplo de configurações PCI

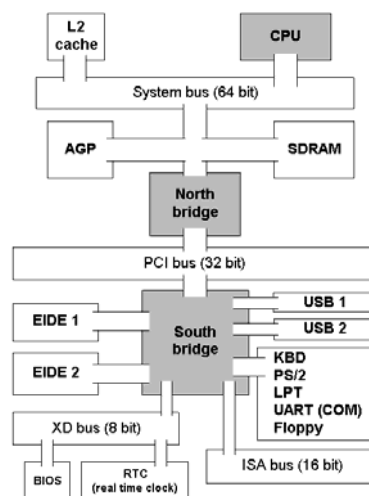


Ponte atua como um buffer de dados

PCI para sistema de processador único (segundo outra fonte)

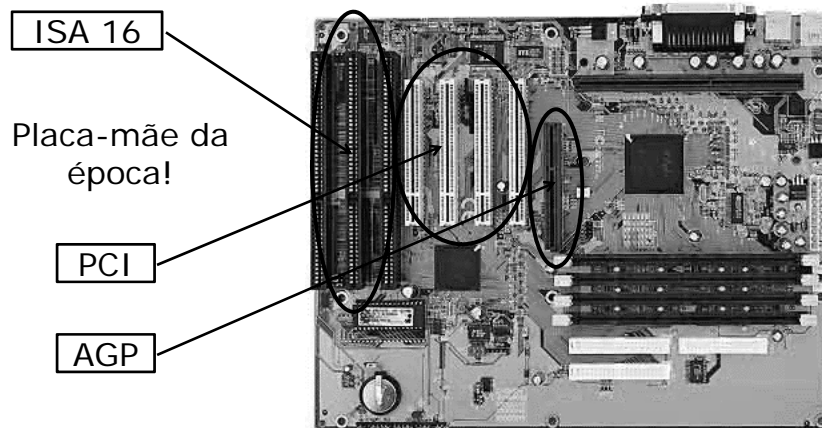


PCI para sistema de processador único (segundo outra fonte)



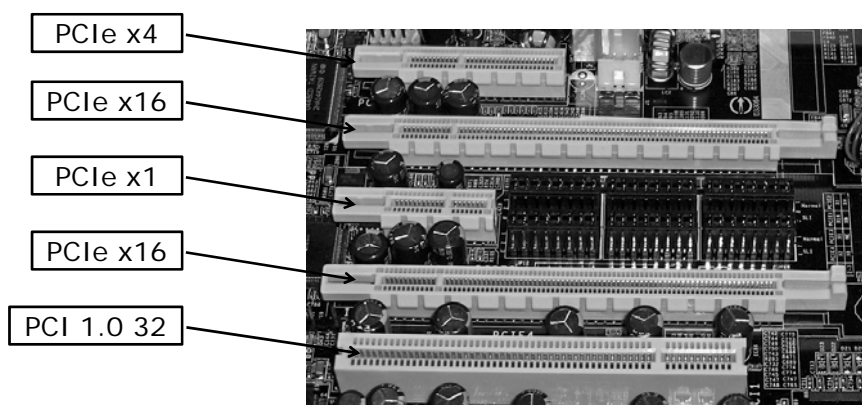
PCI na prática

Onde estão as pontes Norte e Sul?



PCI hoje

Barramento **PCI Express** (criado em 2004)

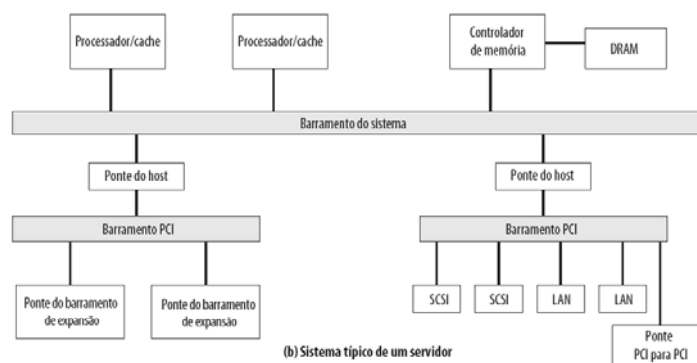


PCI hoje

Comparando ...

Versão do Barramento	Relógio Placa-Mãe (MHz)	Velocidade no Barramento de Dados (MHz)	Tamanho do Barramento de Dados (bits)	Velocidade no Barramento (MBytes/s)
ISA 8	8	8	8	8
ISA 16	8	8	16	16
MCA 16	10	10	16	20
MCA 32	10	10	32	40
EISA		8,25	32	33
PCI 32 (PCI 1.0)		33	32	132
PCI 32		66	32	264
PCI 32		66	32	264
PCI 64		33	64	264
PCI 64 (PCI 2.1)		66	64	528
PCI 64		100	64	800
PCI-X 133		133	64	1.064
PCI-X 266 (2x)		133	64	2.132
PCI-X 533 (4x)		133	64	4.266
PCI-Express 1X		2.5 GHz	serial/1 via	250
PCI-Express 2X		2.5 GHz	serial/2 vias	500
PCI-Express 4X		2.5 GHz	serial/4 vias	1.000
PCI-Express 16X		2.5 GHz	serial/16 vias	4.000
PCI-Express 32X		2.5 GHz	serial/32 vias	8.000

PCI para sistema de multiprocessador



Barramento do sistema admite apenas proc/cache, memória e pontes PCI

Linhas de barramento PCI (obrigatórias)

- Pinos do sistema:
 - Incluindo clock e reset.
- Pinos de Endereços e dados:
 - 32 linhas multiplexadas para endereços e dados.
 - Linhas de interpretação e validação.
- Pinos de Controle da interface: controlam a temporização síncrona
- Pinos de Arbitração:
 - Centralizada.
 - Conexão direta ao arbitrador PCI.
- Pinos de erro: erros de paridade e outros

Linhas de barramento PCI (obrigatórias)

Tabela 3.3 Linhas de sinal do PCI obrigatórias

Designação	Tipo	Descrição
Pinos do sistema		
CLK	in	Fornecer a temporização para todas as transações e todas as entradas são amostradas na transição de subida. São admitidas taxas de clock de até 33 MHz.
RST#	in	Inicializa todos os registradores específicos do PCI, sequenciadores e sinais.
Pinos de endereço e de dados		
AD[31:0]	t/s	Linhas multiplexadas usadas para endereços e dados.
C/BE[3:0]#	t/s	Sinais multiplexados para comandos de barramento e para habilitação de bytes de dados. Durante a fase de dados, as linhas indicam qual das quatro pistas transporta dados significativos.
PAR	t/s	Fornecer paridade par pelas linhas AD e C/BE no ciclo de clock seguinte. O mestre envia o sinal PAR para as fases de escrita de dados e de endereço; o destino envia o sinal PAR para as fases de leitura de dados.

Linhas de barramento PCI (obrigatórias)

Pinos de controle de interface		
FRAME#	s/t/s	Controlado pelo mestre corrente para indicar o início e a duração de uma transação. Ele é ativado no início e desativado quando o iniciador estiver pronto para começar a última fase de dados.
IRDY#	s/t/s	Iniciador pronto, controlado pelo mestre de barramento corrente (iniciador da transação). Durante uma leitura, indica que o mestre está preparado para receber dados; durante uma escrita, indica que os dados válidos estão presentes em AD.
TRDY#	s/t/s	Destino pronto, controlado pelo destino (dispositivo selecionado). Durante uma leitura, indica que os dados válidos estão presentes em AD; durante uma escrita, indica que o destino está pronto para receber dados.
STOP#	s/t/s	Indica que o destino corrente deseja que o iniciador interrompa a transação atual.
IDSEL	in	Seleção de dispositivo de inicialização. Usado como uma seleção de chip durante as transações de leitura e escrita de configuração.
DEVSEL#	in	Seleção de dispositivo. Ativado pelo destino quando ele tiver reconhecido seu endereço. Indica ao iniciador corrente se algum dispositivo foi selecionado.
Pinos de arbitragem		
REQ#	t/s	Indica ao arbitrador que esse dispositivo requer o uso do barramento. Essa é uma linha ponto a ponto específica do dispositivo.
GNT#	t/s	Indica ao dispositivo que o arbitrador concedeu acesso ao barramento. Essa é uma linha ponto a ponto específica do dispositivo.
Pinos de erro		
PERP#	s/t/s	Erro de paridade. Indica que um erro de paridade de dados é detectado por um destino durante uma fase de escrita de dados ou por um iniciador durante uma fase de leitura de dados.
SERR#	o/d	Erro do sistema. Pode ser enviado por qualquer dispositivo para relatar erros de paridade de endereço e erros críticos diferentes de paridade.

Linhas de barramento PCI (opcionais)

- Linhas de interrupção:
 - Não compartilhadas.
- Suporte de cache.
- Extensão de barramento de 64 bits:
 - 32 linhas adicionais.
 - Multiplexada no tempo.
 - 2 linhas para ativar dispositivos a combinar com o uso da capacidade de 64 bits.
- JTAG/Boundary Scan:
 - Para procedimentos de teste.

Linhas de barramento PCI (opcionais)

Tabela 3.4 Linhas de sinal PCI opcionais

Designação	Tipo	Descrição
Pinos de interrupção		
INTA#	o/d	Usado para requisitar uma interrupção.
INTB#	o/d	Usado para requisitar uma interrupção; só tem significado para dispositivos multifuncionais.
INTC#	o/d	Usado para requisitar uma interrupção; só tem significado para dispositivos multifuncionais.
INTD#	o/d	Usado para requisitar uma interrupção; só tem significado para dispositivos multifuncionais.
Pinos de suporte de cache		
SB0#	in/out	Recuo de snoop. Indica um acerto, em uma linha modificada na cache.
SDONE	in/out	Snoop completo. Indica o estado do snoop para o endereço corrente. Ativado quando o snoop tiver sido concluído.
Pinos de extensão de barramento de 64 bits		
AD[63:32]	t/s	Linhas multiplexadas para endereço e dados para estender o barramento até 64 bits.
C/BE[7:4]#	t/s	Sinais multiplexados para comandos de barramento e para habilitação de bytes de dados. Durante a fase de endereço, as linhas fornecem comandos de barramento adicionais. Durante a fase de dados, as linhas indicam qual das quatro vias de bytes da extensão carregam dados significativos.
REQ64#	s/t/s	Usado para requisitar transferência de 64 bits.
ACK64#	s/t/s	Indica que o destino está querendo realizar uma transferência de 64 bits.
PAR64	t/s	Oferece paridade par pelas linhas AD e C/BE estendidas um ciclo de clock adiante.

Linhas de barramento PCI (opcionais)

JTAG/pinos de testes		
TCK	in	Clock de teste. Usado para informação de estado do clock e teste de dados entrando e saindo do dispositivo durante varredura de fronteira.
TDI	in	Entrada de teste. Usado para deslocar dados e instruções de teste em forma serial para dentro do dispositivo.
TDO	out	Saída de teste. Usado para deslocar dados e instruções de teste em forma serial para fora do dispositivo.
TMS	in	Seleção de modo de teste. Usado para controlar o estado do controlador da porta de acesso de teste.
TRST#	in	Reset de teste. Usado para inicializar o controlador da porta de acesso de teste.

in Sinal apenas de entrada

out Sinal apenas de saída

t/s Bidirecional, tri-state. Sinal de E/S

s/t/s Sinal tri-state sustentado, controlado apenas por um proprietário de cada vez

o/d Dreno aberto: permite que múltiplos dispositivos compartilhem como um wire-OR

O estado ativo do sinal ocorre na voltagem baixa

Comandos PCI

- Transação entre iniciador (mestre) e destino.
- Mestre reivindica barramento.
- Determina tipo de transação.
 - P.e., leitura/escrita de E/S.
- Transação significa:
 - Uma fase de endereço
 - Uma ou mais fases de dados

Comandos PCI (p.e. transferência de dados)

Tabela 3.5 Interpretação dos comandos de leitura PCI

Tipo de comando de leitura	Para memória cacheável	Para memória não cacheável
<i>Memory Read</i>	Transferência de metade ou menos de uma linha de cache	Transferência de dados durante 2 ciclos de clock ou menos
<i>Memory Read Line</i>	Transferência de mais de metade de uma linha de cache para três linhas de cache	Transferência de dados durante 3 a 12 ciclos de clock
<i>Memory Read Multiple</i>	Transferência de mais de três linhas de cache	Transferência de dados por mais de 12 ciclos de clock

Diagrama de temporização de leitura PCI

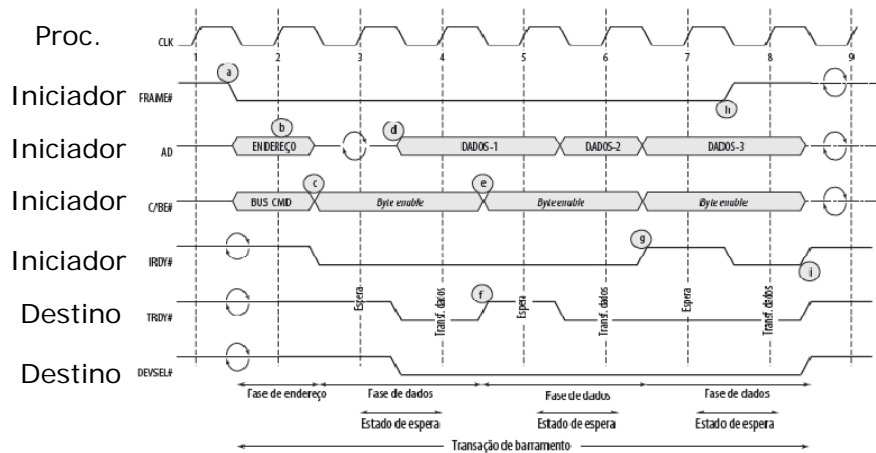


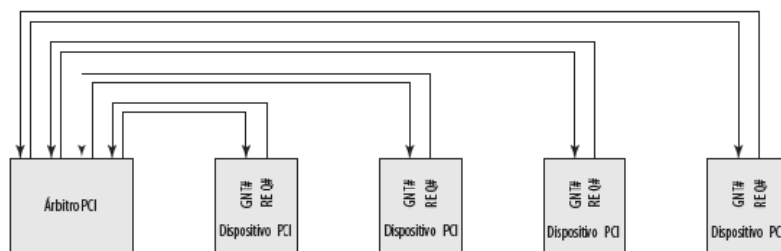
Diagrama de temporização de leitura PCI

- a – um Iniciador inicia transação ativando FRAME e coloca endereço e comando de leitura.
- b – Destino reconhece endereço nos pinos AD.
- c – Iniciador retira endereço e muda informação nos pinos C/BE#, indicando quais pinos de endereço devem ser usados para dados. Também ativa IRDY#, indicando que está pronto para o 1º item de dado.
- d – Destino ativa DEVSEL indicando que reconheceu seu endereço, coloca os dados em AD e ativa TRDY indicando que dados válidos estão no barramento.
- e – Iniciador lê o primeiro bloco de dados e muda pinos C/BE#, em preparação para a próxima leitura.

Diagrama de temporização de leitura PCI

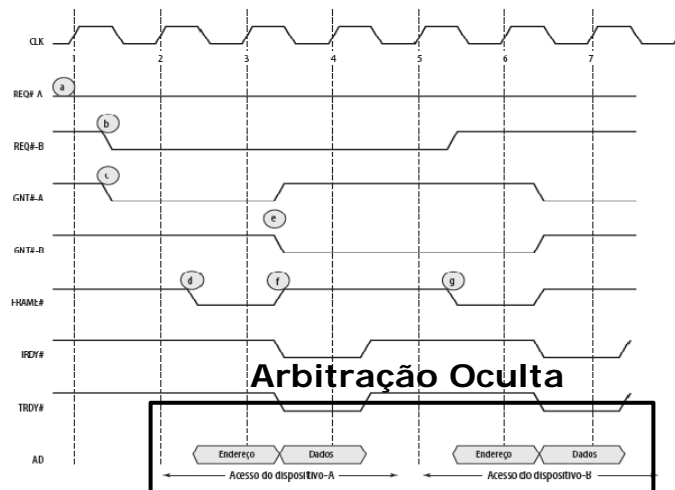
- f – destino desativa TRDY por 1 clock. Iniciador lerá o segundo bloco de dados com 1 clock de atraso.
- g – Destino coloca o terceiro bloco de dados no barramento, mas Iniciador não está pronto e desativa IRDY, fazendo com que o Destino mantenha o terceiro bloco de dados no barramento por mais 1 clock.
- h - O Iniciador sabe que esta é a última transferência de dados e desativa FRAME para sinalizar ao Destino e ativa IRDY para sinalizar que está pronto para completar a transferência.
- i – Iniciador desativa IRDY, retornando o barramento ao estado ocioso e o Destino desativa TRDY e DEVSEL

Árbitro de barramento PCI



Árbitro centralizado e síncrono. Cada Mestre tem um único sinal de requisição (REQ) e concessão ou *grant* (GNT)

Arbitração de barramento PCI (entre 2 Mestres)



Arbitração de barramento PCI (entre 2 Mestres)

- a – A ativa REQ. Árbitro verifica este sinal.
- b – B também ativa seu REQ#B.
- c – Ao mesmo tempo, árbitro responde A com GNT#A.
- d – A verifica GNT#A no início do 2º clock e também verifica que IRDY e TRDY estão desativados, indicando barramento ocioso, então ativa FRAME, coloca endereço e comando C/BE e mantém REQ#A, pois tem uma 2ª transação a realizar.
- e – Árbitro verifica todas os pinos REQ no 3º clock e toma a decisão de conceder o barramento a B para a próxima transação e ativa GRT#B e desativa GRT#A. B não pode usar o barramento até que ele volte a ocioso.

Arbitração de barramento PCI (entre 2 Mestres)

- f – A desativa FRAME indicando que a última transferência de dados está em andamento, sinaliza o Destino com IRDY que lê os dados no início do próximo clock.
- g – No início do 5º ciclo de clock, B verifica FRAME e IRDY desativados, portanto consegue tomar o controle do barramento, ativando FRAME e colocando o endereço. Também desativa sua linha REQ#B, pois só deseja mais uma transação.
- Em seguida, o Mestre A recebe acesso ao barramento para a sua próxima transação.
- Nenhum ciclo do barramento é perdido com a arbitração, conhecida como *arbitração oculta*.

Outras fontes de Informação

- PCI Special Interest Group
<http://www.pcisig.com/home>