

# Organização de computadores

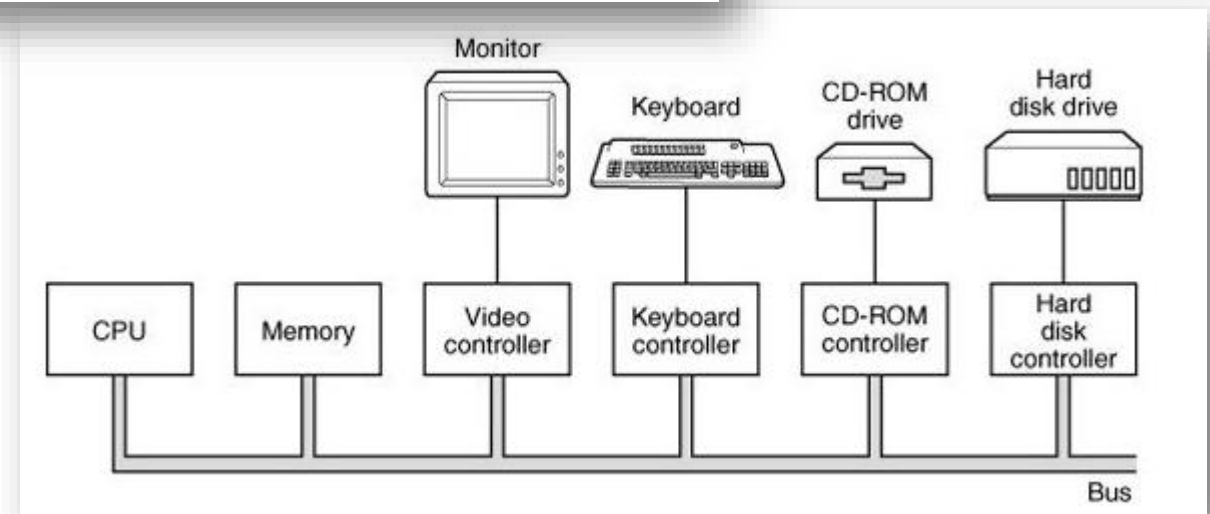
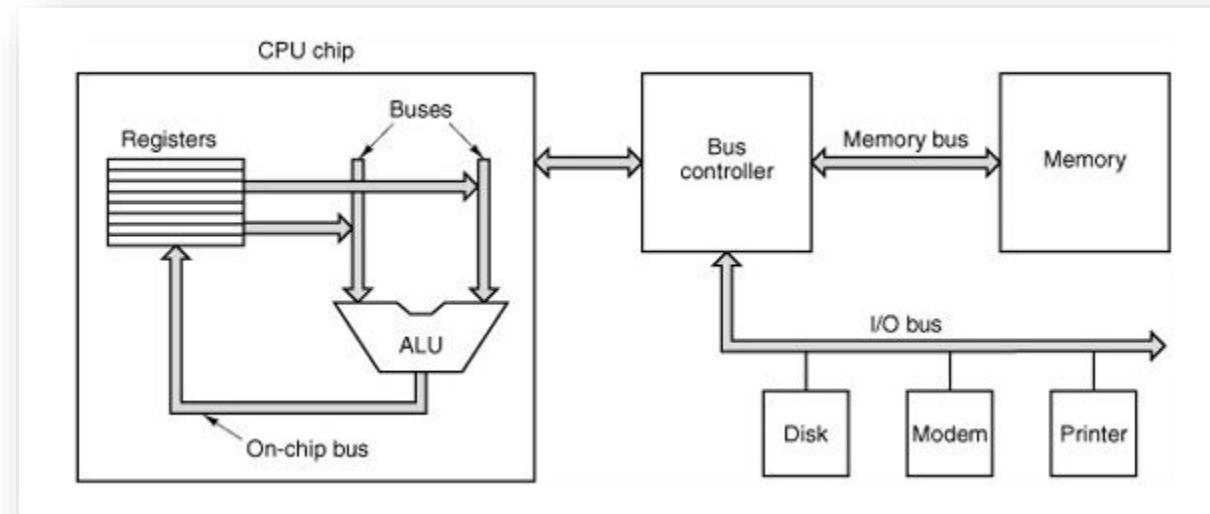
Segundo Bimestre

CNAT – 2015.2

Prof. Jean Galdino

# Sistemas de Interconexão

- BARRAMENTOS



# Alguns pontos

- Um ciclo de instrução consiste:
  - de uma busca de instrução
  - seguida por zero ou mais buscas de operandos
  - seguidas por zero ou mais armazenamentos de operandos
  - seguidos por uma verificação de interrupção ( se estiverem habilitadas)
- Os principais componentes de um computador precisam ser interconectados
  - Trocar dados
  - Trocar sinais de controle

# Alguns pontos

- O meio de interconexão para troca de dados e sinais de controle mais popular é o barramento
  - Múltiplas linhas
  - Nos sistemas contemporâneos existem uma hierarquia para de barramentos
    - Melhorar o desempenho
  - Arbitração
    - Permissão para enviar sinais nas linhas do barramento
  - Temporização - Ciclos de barramento podem ter qualquer largura e não são os mesmos entre todos os pares
    - Síncrona - (Clock central)
    - Assíncrona
  - Largura
    - Linhas de endereços e de dados

# Von Neumann

- Relembrando:
  - Praticamente todos os computadores modernos utilizam este conceito
  - Arquitetura de Von Neumann
    - Dados e instruções são armazenados em uma única memória R/W
    - O conteúdo desta memória é endereçável por local
      - Não considera o tipo de dado nela contida
    - A execução ocorre em um padrão sequencial de uma instrução para a seguinte
  - Os dados são armazenados em binário
    - Sofrem operações lógicas e aritméticas

# Definições básicas

- Um barramento é um caminho que conecta dois ou mais dispositivos
  - Meio de transmissão compartilhado
  - Um sinal transmitido por qualquer dispositivo estará disponível para a recepção de qualquer outro
  - Só um dispositivo transmite por vez
  - Múltiplos caminhos conduzindo bits
  - Várias linhas transmitindo – paralelo
- Barramento de sistema
  - Conecta processador, memória e I/O

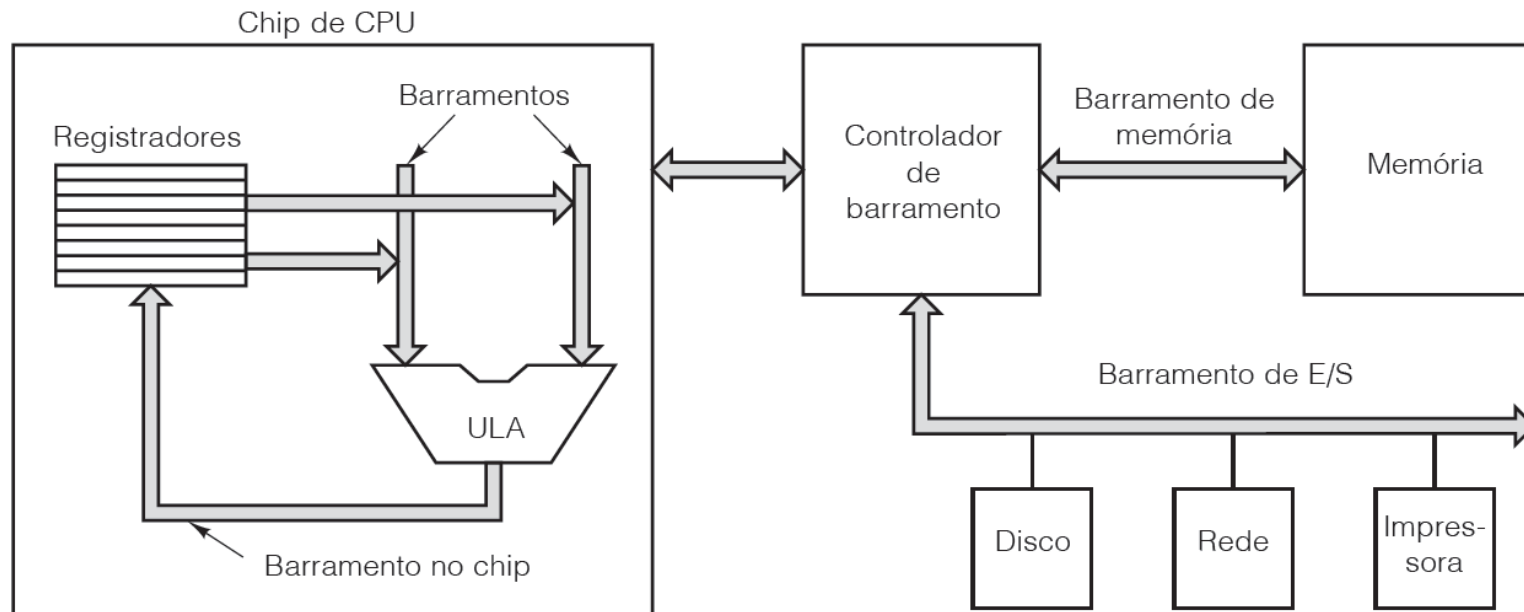
# Definições básicas

---

- Como o barramento conecta diversos dispositivos, deve haver um conjunto de regras que rejam a comunicação (protocolo);
- Um barramento requer um “controlador de barramento” que é um circuito digital que implementa o protocolo de comunicação no barramento;
- Para entendermos como um barramento funciona, primeiro precisamos entender que sinais devem ser considerados.

# Barramentos de computador

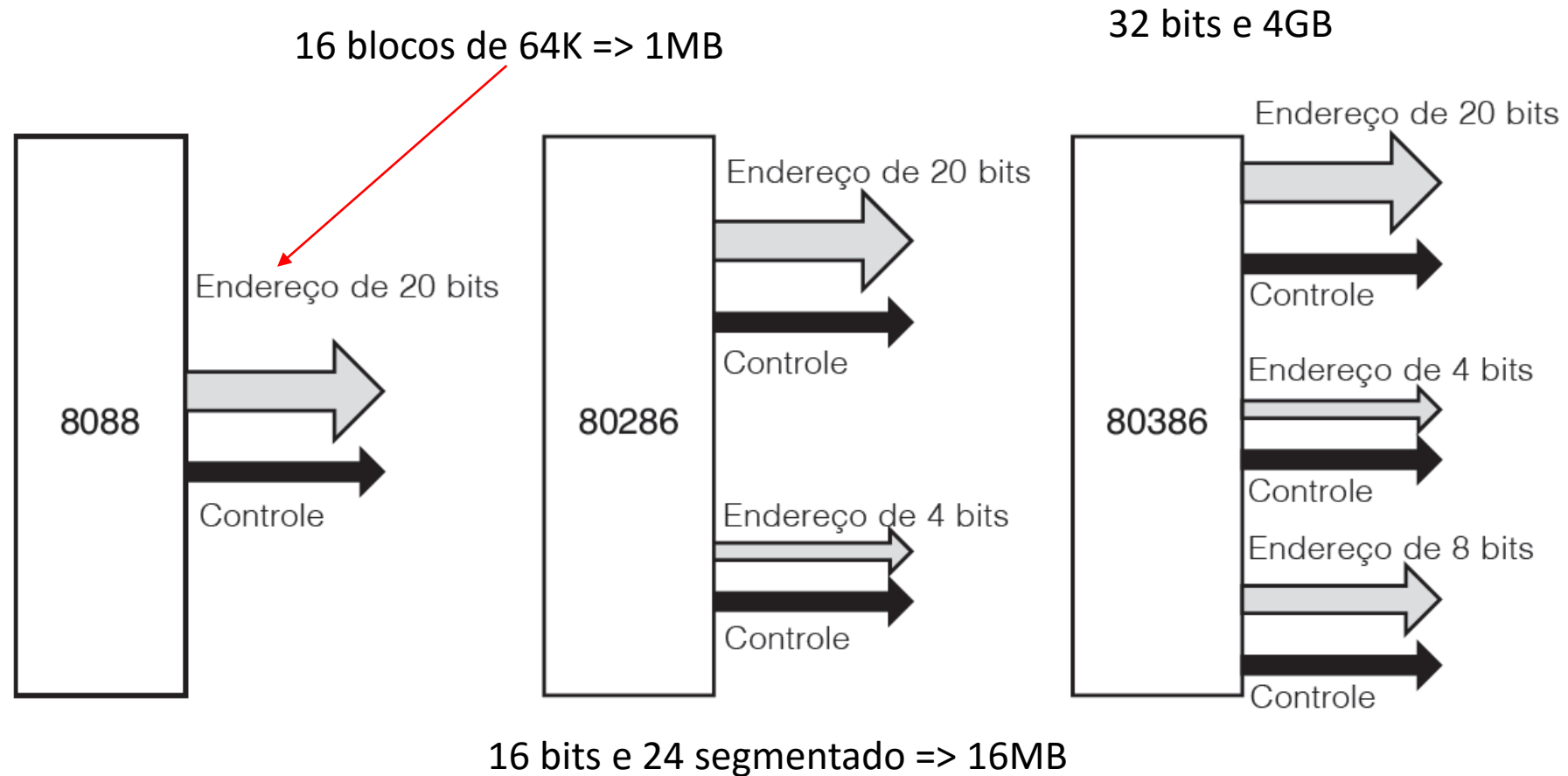
- Um **barramento** é um caminho elétrico comum entre vários dispositivos.
- Sistema mínimo, com um barramento de memória e um barramento de E/S:





# Crescimento dos Barramentos

- Crescimento de um barramento de endereços ao longo do tempo.



# Serial & Paralelo

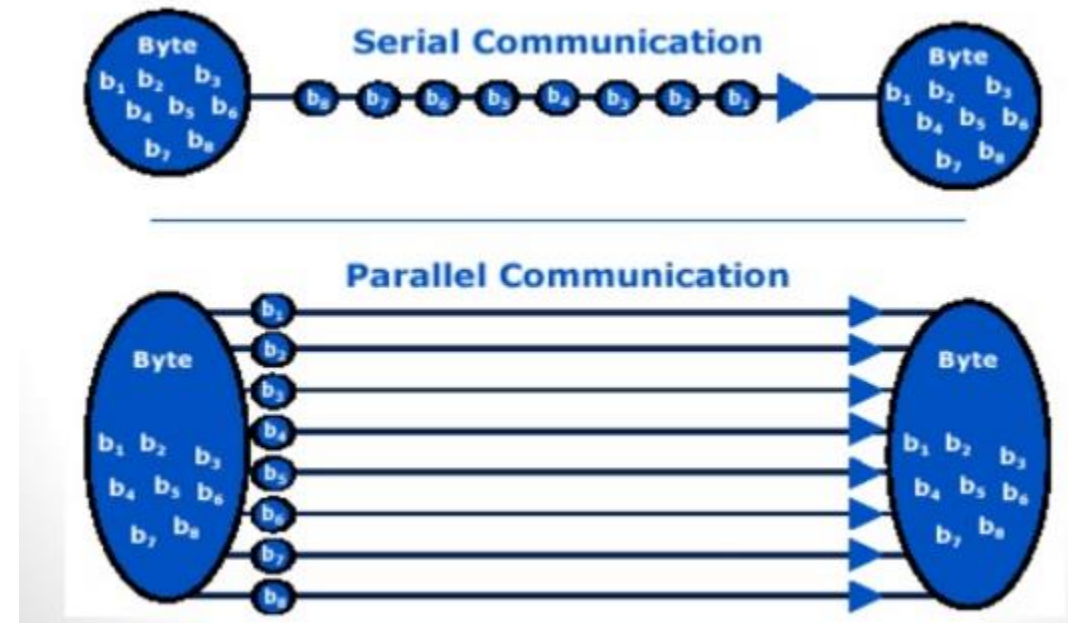
- **Classificação:**

- Barramento Serial

- O barramento pode ter um único fio, por onde passa um sinal de cada vez, um bit de cada vez. Exemplos: USB, PCI Express, Hyper-Transport, Firewire.

- Barramento Paralelo

- O barramento pode ter vários fios, por onde passam vários sinais simultaneamente, um por cada fio. Exemplos: UNIBUS, MCA, ISA, EISA, PCI, AGP.



# Arbitragem de barramento

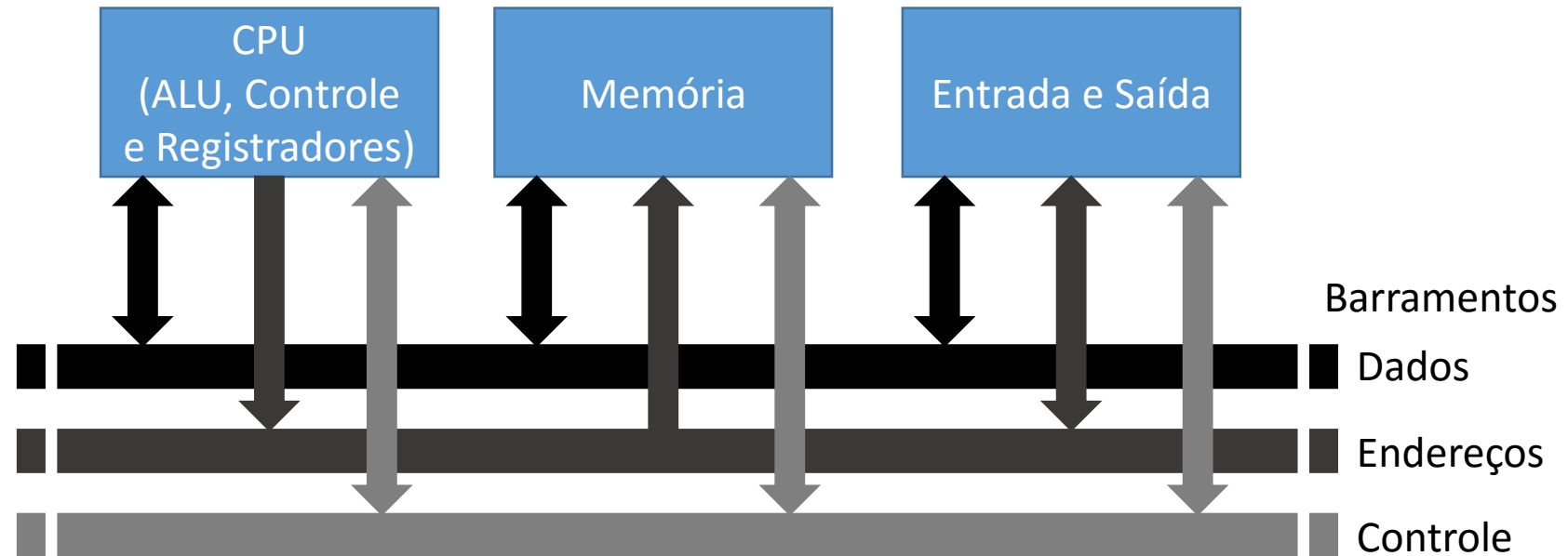
- “O que acontece se dois ou mais dispositivos quiserem se tornar mestres de barramento ao mesmo tempo?”
- É preciso algum mecanismo de **arbitragem de barramento** para evitar o caos.
- Mecanismos de arbitragem podem ser centralizados ou descentralizados.
- Se vários níveis de prioridade são requisitados ao mesmo tempo, o árbitro emite uma concessão somente ao de prioridade mais alta.

# Estrutura de barramento

- De 50 a centenas de linhas
- Cada linha com uma função em particular
  - DADOS – 32,64, 128 ou mais linhas
    - Largura determina em geral o desempenho do sistema
  - ENDEREÇOS - 8, 16 ou 32
    - Endereça memória, E/S, (designa a origem ou destino dos dados)
    - Largura determina a capacidade de memória máxima do sistema
  - CONTROLE
    - Transmitem informações de comando e sincronização
    - Sincronização – validade das informações de dados e endereços
    - Comando – especificam operações a serem realizadas
- Pode haver linha de distribuição de potência

# Estrutura do Barramento

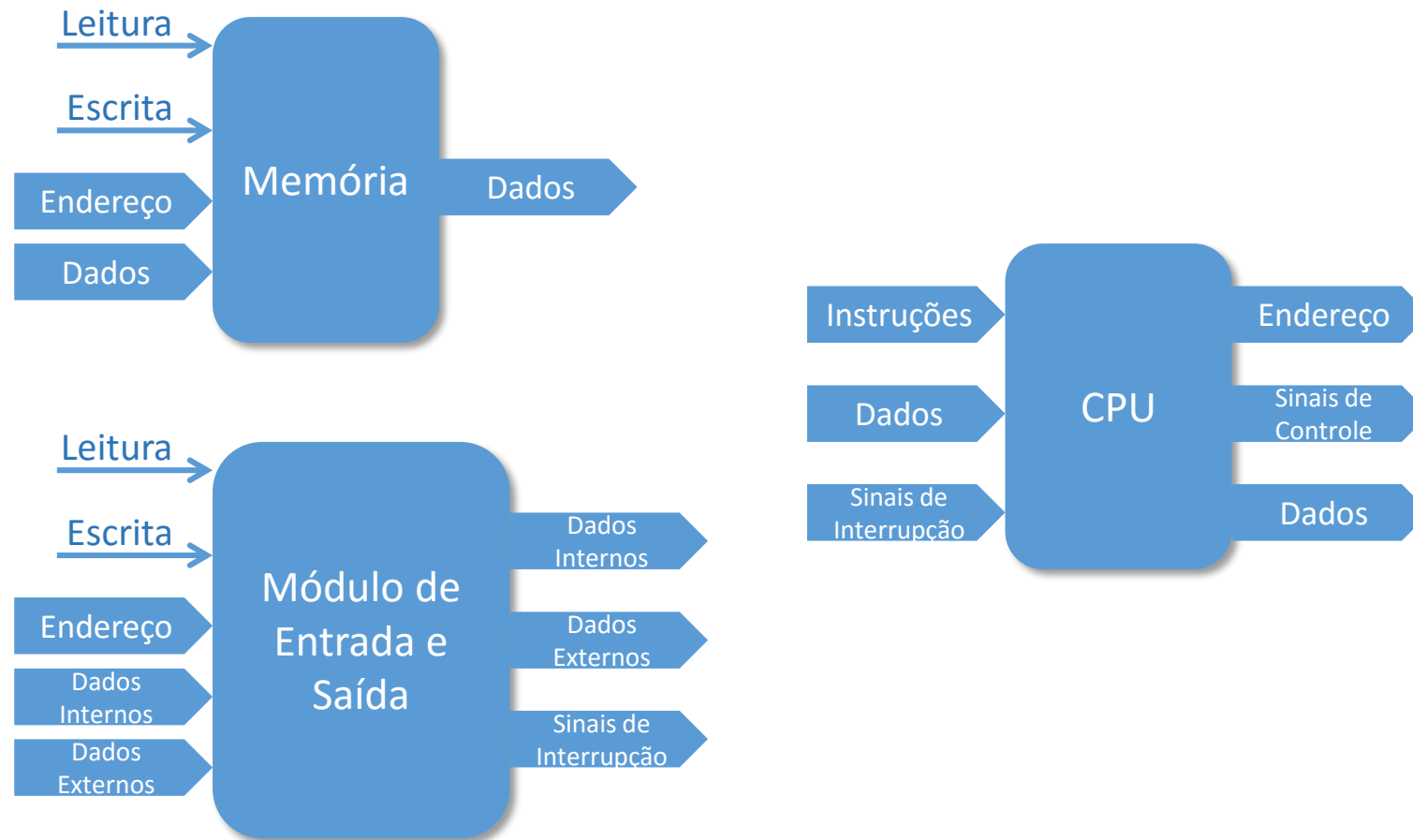
---



# Sinais de controle

- Escrita e Leitura
- Leitura da memória
- Escrita E/S
- Leitura E/S
- ACK de transferência (dados aceitos ou colocados no barramento)
- Bus request
- Bus grant
- ACK de interrupção (interrupção pendente foi reconhecida)
- Clock
- Reset

# Sinais dos Módulos de um Sistema Computacional



# Hierarquia de Barramentos

---

- Muitos dispositivos → barramento se torna o “gargalo” do sistema computacional
  - Barramento longo → atraso de propagação
  - Muitos dispositivos → concorrência → atraso



# Operação no barramento

- Se um módulo deseja enviar dados para o outro
  - Obter o barramento
  - Transferir o dado através deste barramento
- Se ele quiser requisitar
  - Obter o barramento
  - Transferir uma requisição a outro módulo pelas linhas de controle e endereços apropriadas
  - Espera o envio do dados pelo módulo requisitado

# Fisicamente

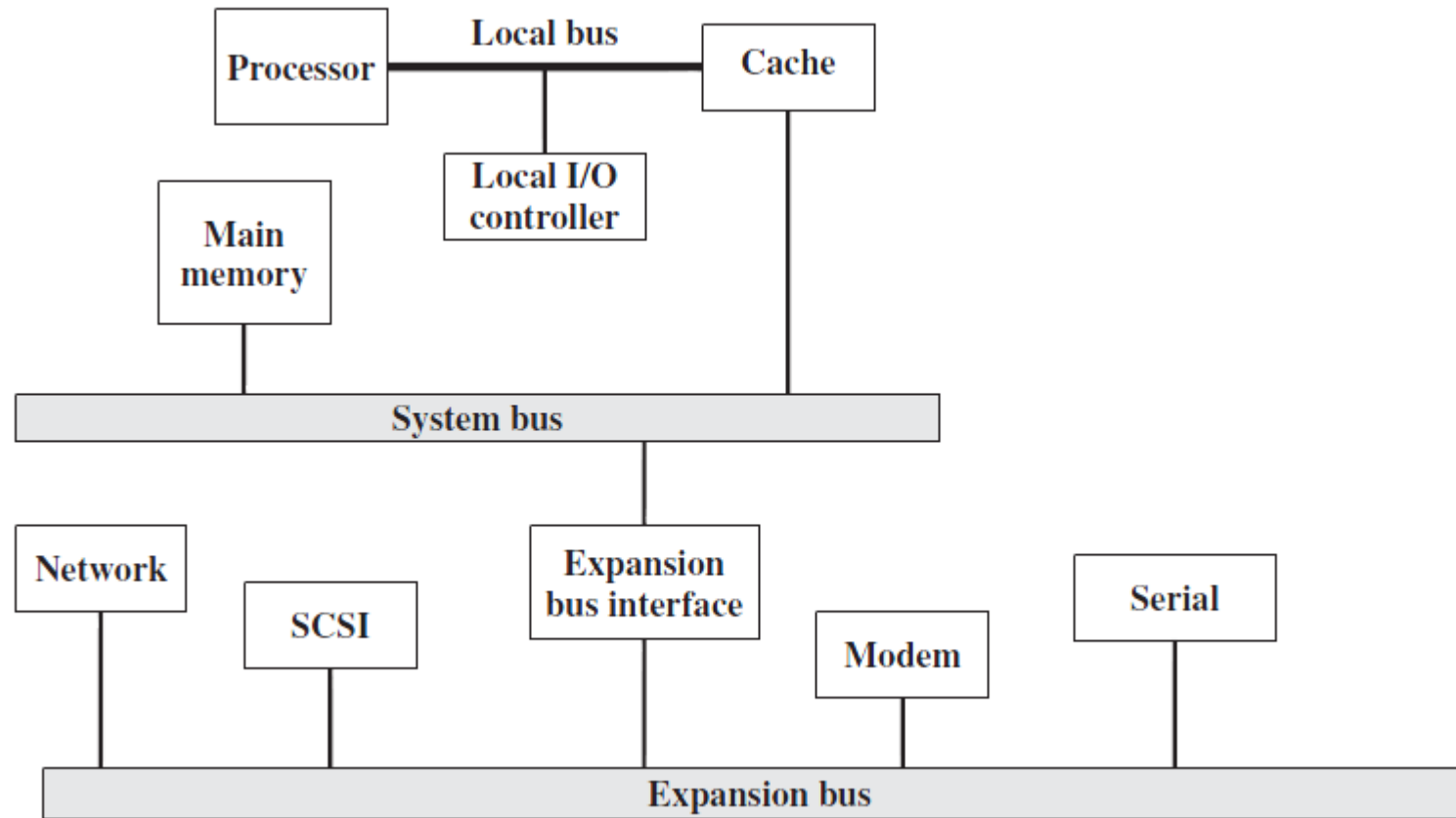
- Fisicamente um barramento é uma série de condutores elétricos paralelos, linhas de metal colocadas em placas de circuito impresso, se estendendo por todos os componentes do sistema.
- Utiliza-se slots como barramento presos a um chassi
- Sistemas modernos tendem a utilizar os principais componentes nas placas, chassi,
- Barramento no chip e barramento na placa

# Hierarquia de barramento múltiplo

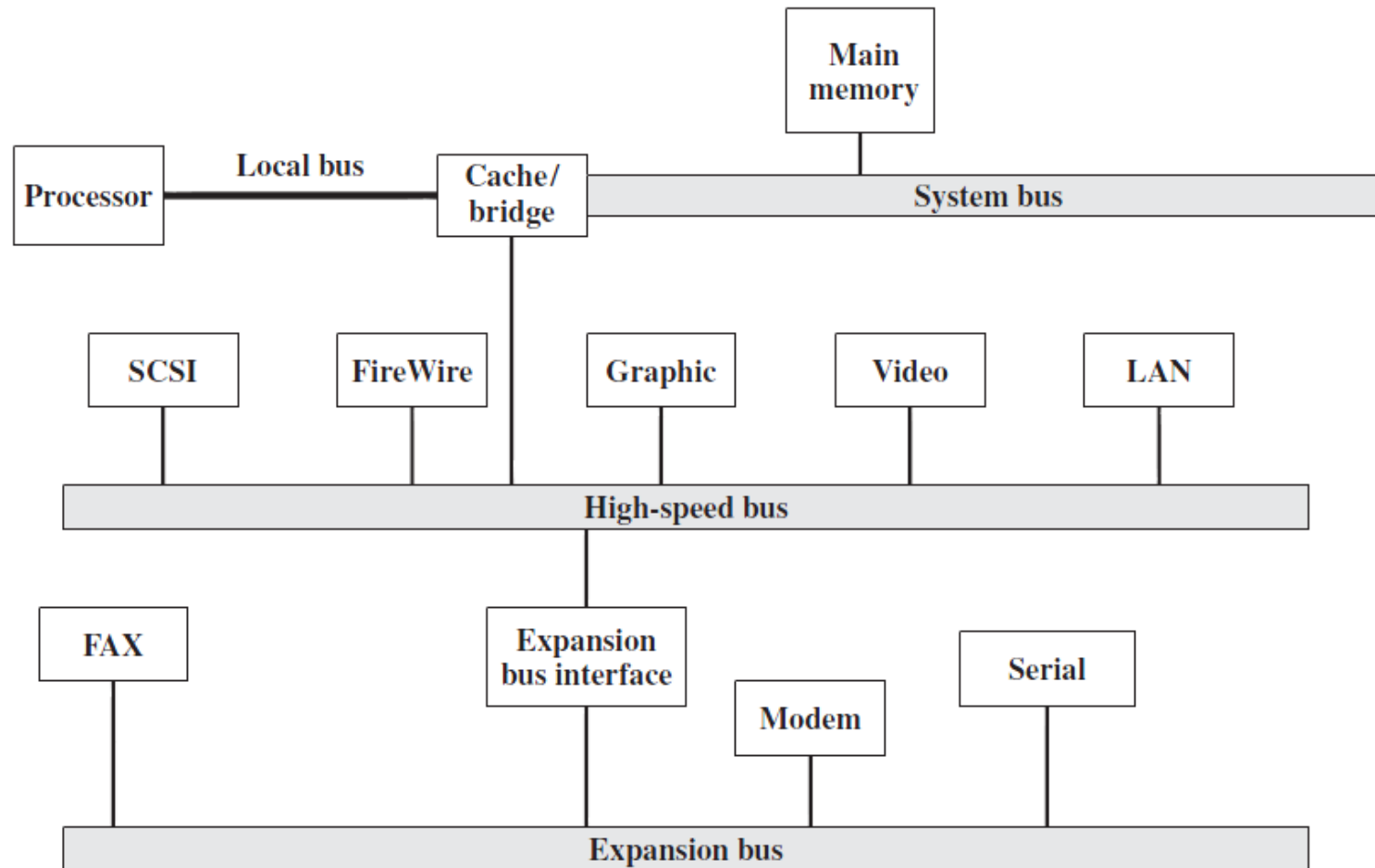
- Quanto maior o número de dispositivos conectados
  - Maior o tamanho do barramento
  - Maior o atraso de propagação
  - Se houver muita mudança de dispositivo pode comprometer o desempenho
- Maioria dos sistemas de computação utilizam barramento múltiplos
  - Barramento local – processador, cache e um outro dispositivo local
  - Barramento de sistema – cache/ponte a memória principal
  - Barramentos de expansão - conectar diversos dispositivos
    - Isola o tráfego memória para processador dos E/S
  - Barramento de alta velocidade
    - Para dispositivos com desempenhos maiores – Rede, SCSI, Vídeo, Gráfico...

# Barramento Comum

---



# Barramento de Alta Velocidade



**FireWare** = é um arranjo de barramento de alta velocidade projetado especificamente para dar suporte a dispositivos de E/S de alta capacidade. (apple)

# Observações

---

- As diferentes características dos componentes levou à criação de diversos tipos de barramentos;
- Motivo: velocidade de taxa de transferência de dados muito diferentes;
- Exemplo: o teclado tem velocidade de transferência de dados menor que a de um disco rígido;
- Atualmente cada barramento interliga dispositivos com velocidades afins, melhorando o desempenho do sistema;
- FSB: frontal side bus ou barramento frontal – interliga o processador e a memória principal;
- BSB: back side bus ou barramento posterior – interliga o processador à memória cache;

- **Exemplos de barramentos:**

- Omnibus: usado no PDP-8 da DEC;
- Unibus: usado no PDP-11 da DEC;
- Multibus: usado no Intel 8086;
- VME: equipamento para laboratório de física;
- Microchannel: usado no PS2/2;
- Nubus: usado no Macintosh
- PCI: usado em muitos computadores;
- SCSI: usado em muitos computadores;
- USB: usado em computadores atuais;
- FireWire: equipamentos eletrônicos de consumo;
- ISA: usado no PC/AT;
- IBM PC: usado no PC/XT;
- EISA: usado no Intel 80386;

# Elementos do projeto de barramento

- Grande variedade de implementações
- Poucos parâmetros ou elemento de projetos básicos
  - Classificam e diferenciam barramentos

TIPO	LARGURA DO BARRAMENTO
Dedicado	Endereço
Multiplexado	Dados
MÉTODO DE ARBITRAGEM	TIPO DE TRANSFERÊNCIA DE DADOS
Centralizado	Leitura
Distribuído	Escrita
SINCRONIZAÇÃO	Ler-modificar-escrever
Síncrona	Leitura-após-escrita
Assíncrona	Bloco

# Elementos do projeto de barramento

- Tipos de Barramento

- Dedicado

- Atribuída permanentemente a uma função ou a um subconjunto físico de componentes de computador
      - Linhas de Endereço e linhas de dados separados

- Multiplexado

- Endereços e dados podem ser transmitidos em uma mesma via(conjunto de linhas) usando uma linha de controle Address Valid.
      - No início de uma transmissão de dados o endereço é colocado no barramento e a linha Address Valid é ativada.
      - Cada módulo tem um período especificado para copia o endereço
      - O endereço é removido e os dados são transferidos
      - Vantagem, menos linhas, mais espaço, menos custo e circuitos mais complexo



# Dedicados & Multiplexados

- **Barramentos Dedicado:**

- Dedicação funcional:

- Refere-se à função das linhas do barramento.
- Exemplo: linhas de endereços e linhas de dados;

- Dedicação física:

- Refere-se ao uso de múltiplos barramentos, cada um conectando apenas um subconjunto de módulos.
- Exemplo: um barramento de E/S para interconectar todos os módulos de E/S.

- **Barramentos Multiplexados:**

- Usa mesmas linhas para múltiplas finalidades;
- As mesmas conexões do barramento podem ser usadas para transferência de dados de escrita ou leitura;
- Exemplo: informações de endereço e dados podem ser transferidas pelo mesmo conjunto de linhas usando uma linha de controle chamada ADDRESS VALID.

# Elementos do projeto de barramento

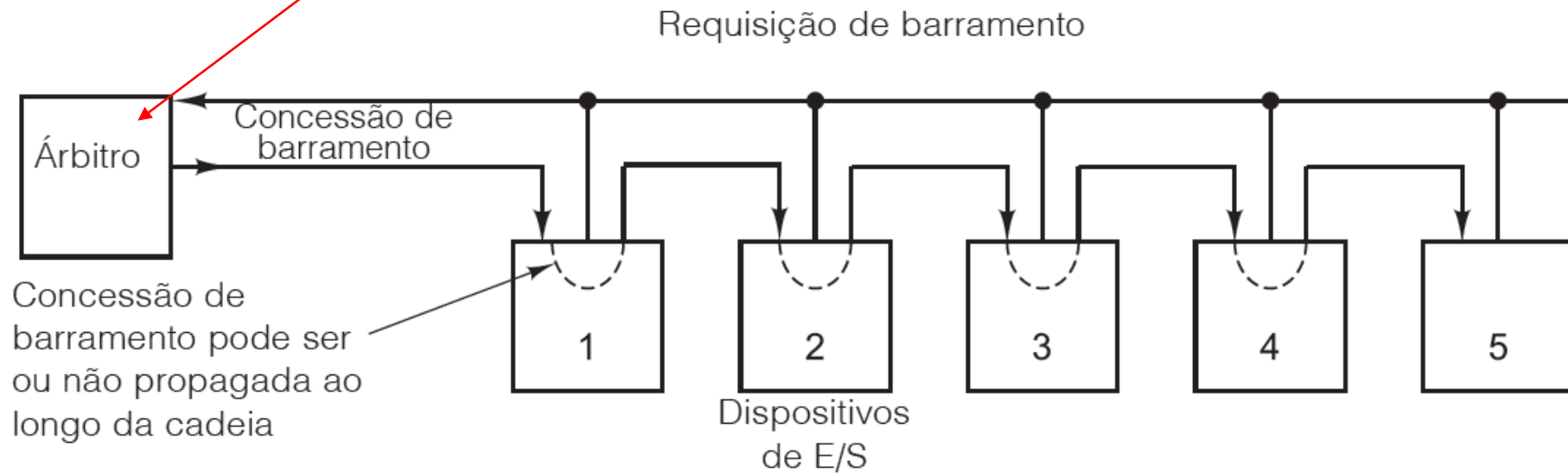
- Método de Arbitração
  - Quando mais de um módulo precisa do controle do barramento
    - Como somente um circuito de cada vez pode fazer a transmissão com sucesso
    - Definição do Mestre (inicia a transmissão – Processador ou I/O) e do Escravo
  - Centralizado
    - Um único Hardware é o responsável (Controlador ou Arbitro)
    - Pode ser um módulo em separado ou parte do processador
  - Distribuído
    - Cada módulo com sua lógica de controle de acesso
    - Os módulos atuam juntos para compartilhar o barramento

Uma única linha de requisição. Assegurada para qualquer um e a qualquer tempo

# Arbitragem de barramento

- Árbitro de barramento centralizado de um nível usando encadeamento em série.

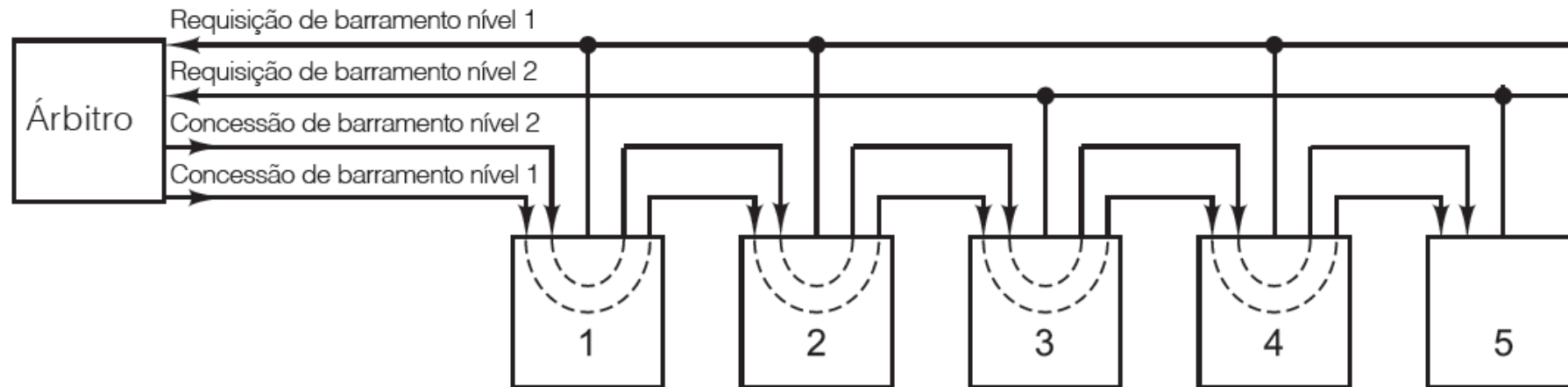
Um único arbitro determina quem entra em seguida



Pode ser um chip na CPU ou um chip separado

# Arbitragem de barramento

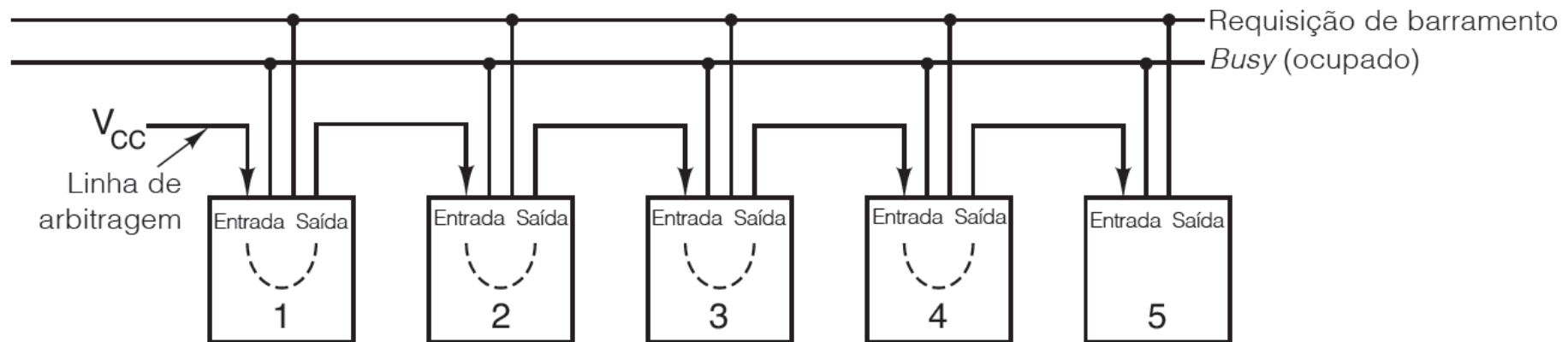
- Mesmo árbitro, mas com dois níveis.



Para cada nível de prioridade uma linha de requisição.

# Arbitragem de barramento

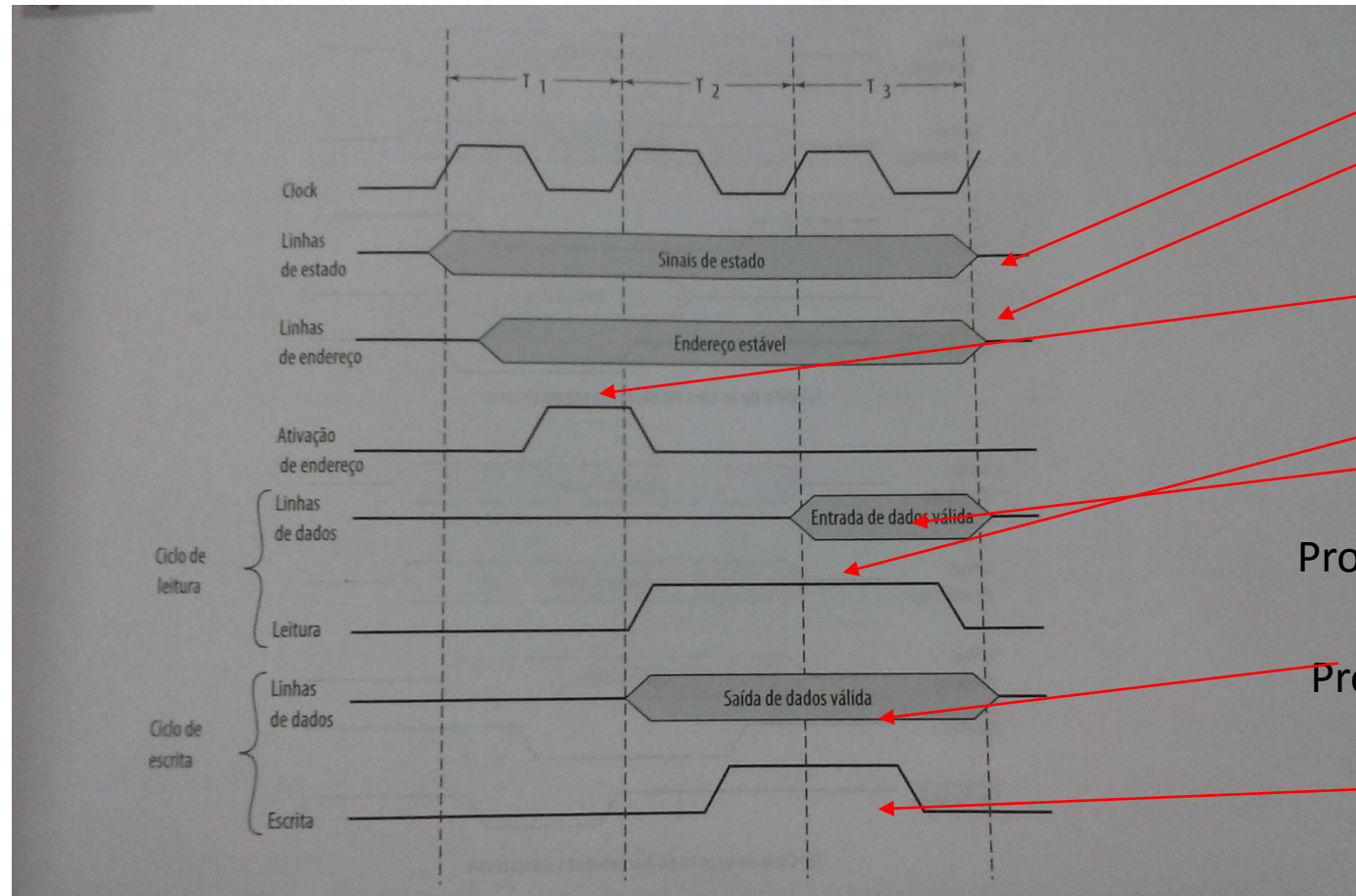
- Outro tipo de arbitragem de barramento descentralizada, usa apenas três linhas, não importando quantos dispositivos estiverem presentes.



# Elementos do projeto de barramento

- Temporização
  - Modo como os eventos são coordenados no barramento
- Síncrono
  - Ocorrência de um evento é determinado por um clock
  - Barramento inclui uma linha de clock
  - Uma transmissão 0-1 é um ciclo de clock – define um slot de tempo
  - Todos os eventos começam no início do ciclo de clock
- Assíncrono
  - Um evento segue a depender da ocorrência do evento anterior
  - Síncrona é mais simples de implementar e testar, menos flexível

# Diagrama de sincronização simplificado



Processador ativa linhas de estado

Processador coloca o endereço

Ativação do endereço

Comando de leitura

Memória coloca os dados nas linhas de dados

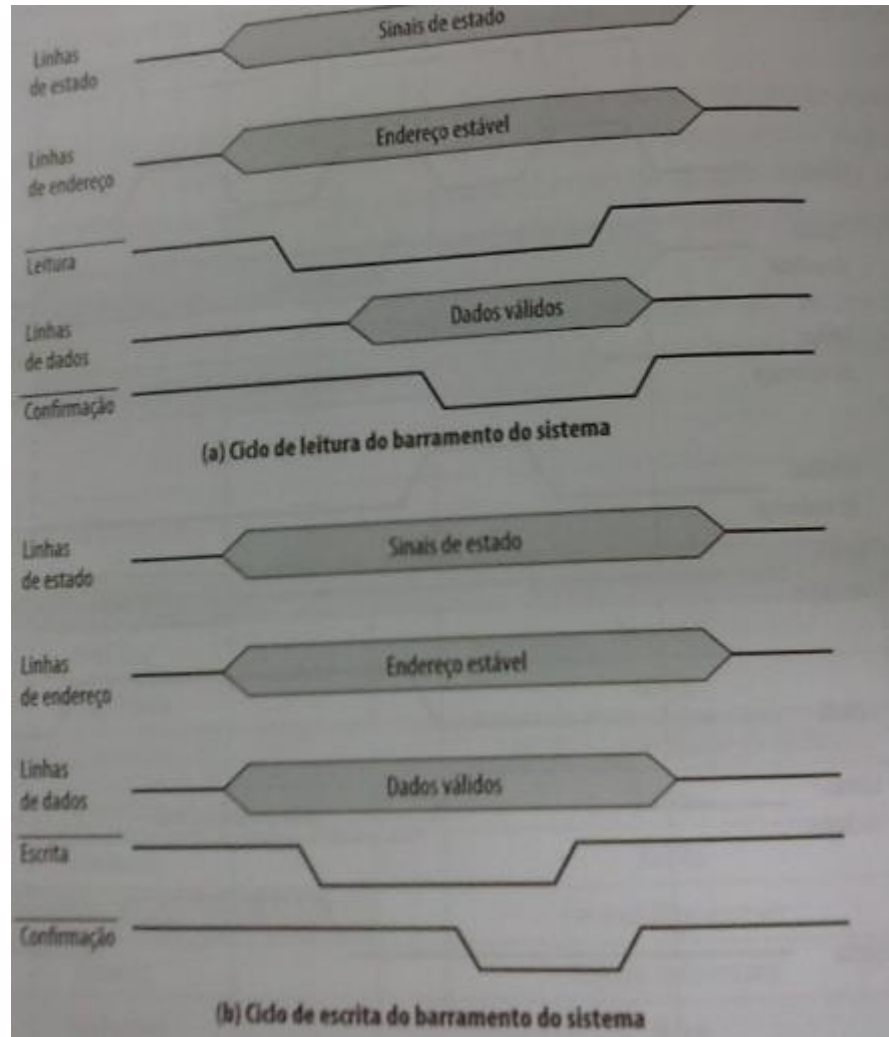
Processador lê os dados e retira o sinal de leitura

Processador coloca os dados nas linhas de dados

Comando de Escrita

Memória copia os dados

# Temporização de operações de barramento assíncronas



Mestre

Processador coloca sinais de estado e de endereço e depois emite um comando de leitura. A memória apropriada decodifica o endereço e responde colocando os dados na linha de dados, por fim módulo de memória ativa a confirmação. **“processador! dados disponíveis!”**

O mestre coloca sinais na linha de dados ao mesmo tempo em que coloca sinais nas linhas de estado e de endereço. A memória responde com o comando de Escrita, copiando os dados na linha de dados, por fim, ativa a confirmação. **Mestre e memória removem os sinais.**



# Síncrono & Assíncrono

- **Barramentos Síncronos:**

- Tem um relógio (clock) mestre;
- Onda quadrada;
- Frequência entre 5 e 100 MHz;
- Regulam o aparecimento/desaparecimento dos sinais nas diversas linhas do barramento;
- Sincroniza o funcionamento do barramento, a ocorrência e a duração de todos os eventos;
- Simples de implementar e testar;
- Qualquer atividade somente pode ser realizada em um intervalo de tempo fixo;
- Dificuldades em trabalhar com dispositivos/componentes que tenham tempos de transferência diferentes

- **Barramentos Assíncronos:**

- Não tem um relógio (clock) mestre;
- Não sincroniza o funcionamento do barramento, a ocorrência e a duração de todos os eventos;
- Cada evento depende da ocorrência de um evento anterior, o qual pode ter duração diferente em tempo.
- Não há unidade fixa de tempo para relacionar as tarefas de uma dada operação;
- Não há qualquer tipo de relação entre os vários sinais que circulam no barramento;
- Facilidade em trabalhar com dispositivos/componentes que tenham tempos de transferência diferentes;
- As atividades são realizadas sem um intervalo de tempo fixo;

# Elementos do projeto de barramento

- Largura do barramento

- Dados -> Impacto sobre o desempenho do sistema
- Endereços -> Impacto sobre a capacidade do sistema

- **Largura do barramento:**

- Largura do barramento de dados:

- O número de linhas no barramento de dados determina quantos bits podem ser transferidos de uma só vez;
- Cada linha só pode transmitir 1 bit de cada vez;
- **Exemplo:** Considere um barramento de dados com largura igual a 32 bits. Considere também que as instruções de máquina tem 64 bits. O que acontece com o processador neste caso?
- **Resposta:** A instrução tem duas vezes o tamanho da largura. Isso fará com que o processador acesse duas vezes o módulo de memória durante cada ciclo de instrução.

- **Largura do barramento:**

- Largura do barramento de dados:

- *Taxa de transferência dos dados:*

- quantidade total de bits que passam pelo barramento na unidade de tempo
- **$T = L * V$**
- T é a taxa de transferência medida em bits por segundo;
- L é a largura do barramento de dados medida em bits;
- V é a velocidade do barramento medida em hertz;
- **Exemplo:** Considere um barramento de dados com largura igual a 10bits e velocidade iguala 100MHz. Qual será a taxa de transferência?
- **Solução:**  $T = L * V \Rightarrow T = 10\text{bits} * 100\text{MHz} \Rightarrow T = 1000\text{Mbps}$  ou 1Gbps

# Elementos do projeto

- Tipos de transferência de dados do barramento

Escrita

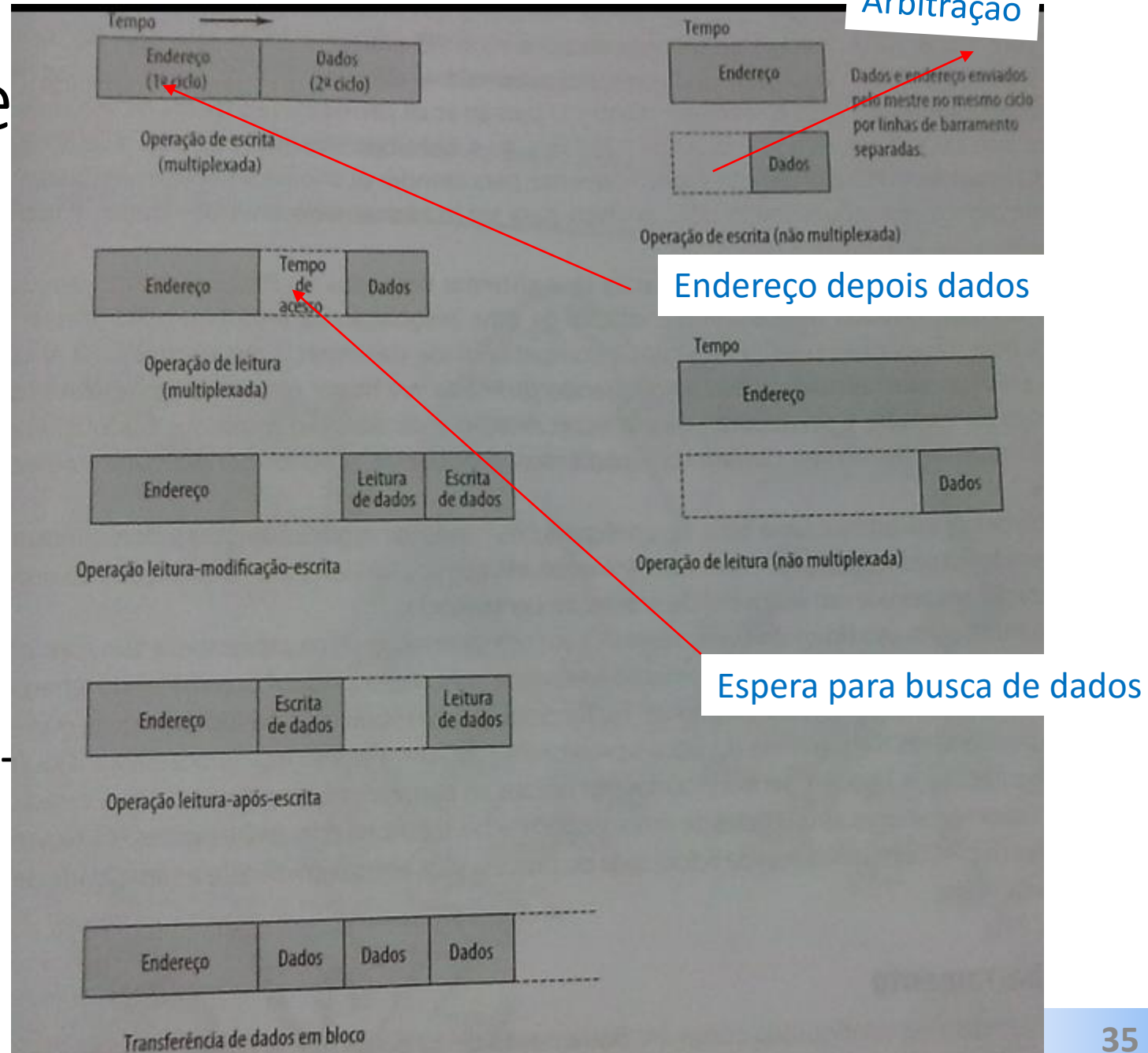
Mestre para escravo

Leitura

Escravo para mestre

Para uma leitura ou uma escrita pode haver espera se

Passar pela arbitração.



# Capacidade de transferência dos barramentos

- Exemplo

CAPACIDADE DE TRANSFERÊNCIA DOS BARRAMENTOS			
Frequência do barramento	Tamanho dos dados	Conversão de unidades	Capacidade de transferência
$\frac{\text{Megaciclos}}{\text{segundo}}$	$\times \frac{\text{bits}}{\text{ciclo}}$	$\times \frac{1 \text{ byte}}{8 \text{ bits}}$	$= \times \frac{\text{Megabytes}}{\text{segundo}}$
Exemplo PCI:			
$33 \frac{\text{Megaciclos}}{\text{segundo}}$	$\times 32 \frac{\text{bits}}{\text{ciclo}}$	$\times \frac{1 \text{ byte}}{8 \text{ bits}}$	$= 132 \frac{\text{Megabytes}}{\text{segundo}}$
Barramento	Frequência	Tamanho dos dados	Capacidade de transferência
ISA	8 MHz	16 bits	16 MB/s
PCI	33 MHz	32 bits	132 MB/s
AGP x2	2 x 66 MHz	32 bits	528 MB/s
AGP x4	4 x 66 MHz	32 bits	1.056 MB/s
FSB Sistema 66 MHz	66 MHz	64 bits	528 MB/s
FSB Sistema 100 MHz	100 MHz	64 bits	800 MB/s
FSB Sistema 133 MHz	133 MHz	64 bits	1.064 MB/s
Cache L2 PII 400 MHz	200 MHz	64 bits	1.600 MB/s
Cache L2 PIII 600 EB	600 MHz	64 bits	4.800 MB/s

# Exemplos de barramentos

- Barramentos são a cola que mantém a integridade dos sistemas de computadores.
- Alguns barramentos populares são:
  - o PCI e
  - o USB.
- O PCI é o principal barramento de E/S usado hoje em dia nos PCs.

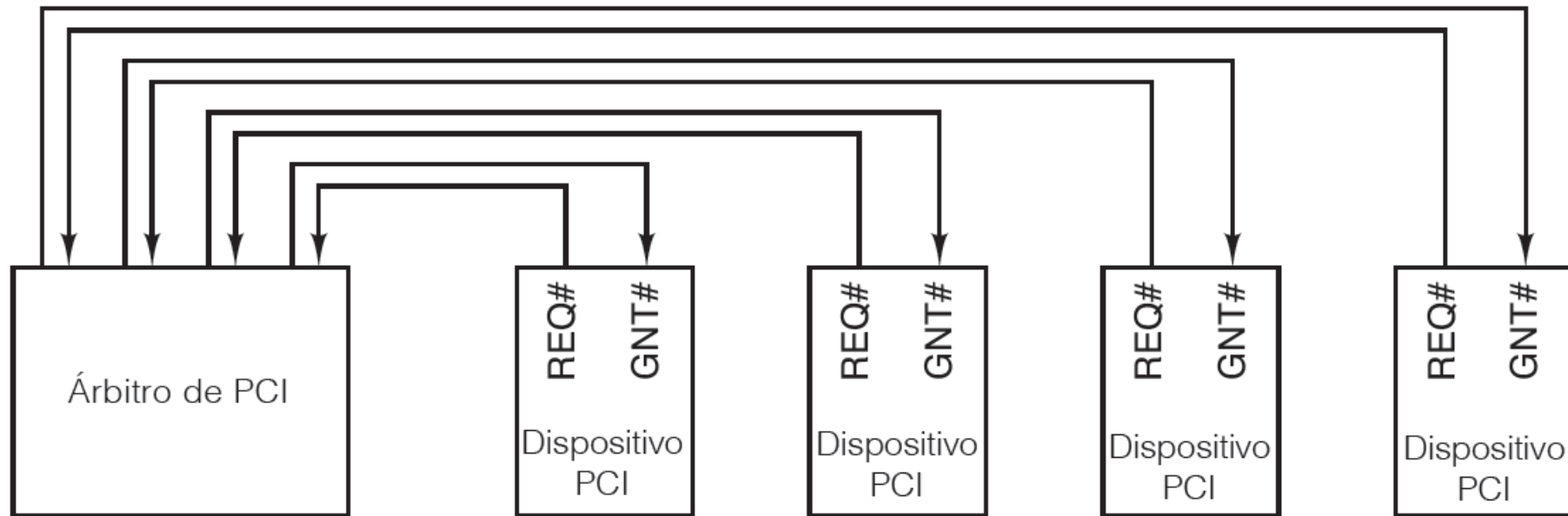
# Exemplos de barramentos

- Ele pode ter duas formas:
  - o barramento PCI mais antigo, e
  - o novo e muito mais rápido barramento PCI Express (PCIe).
- O Universal Serial Bus é um barramento de E/S cada vez mais popular para periféricos de baixa velocidade, como mouses e teclados.
- Uma segunda e terceira versões do barramento USB rodam com velocidades muito mais altas.



# Exemplos de barramentos

- O barramento PCI usa um árbitro de barramento centralizado.



# Exemplos de barramentos

- Sinais obrigatórios de barramento PCI.

Sinal	Linhas	Mestre	Escravo	Descrição
CLK	1			<i>Clock</i> (33 MHz ou 66 MHz)
AD	32	x	x	Linhas de endereço e de dados multiplexadas
PAR	1	x		Bit de paridade de endereço ou dados
C/BE	4	x		Comando de barramento/mapa de bits para bytes habilitados
FRAME#	1	x		Indica que AD e C/BE estão ativadas
IRDY#	1	x		Leitura: mestre aceitará; escrita: dados presentes
IDSEL	1	x		Seleciona espaço de configuração em vez de memória
DEVSEL#	1		x	Escravo decodificou seu endereço e está na escuta
TRDY#	1		x	Leitura: dados presentes; escrita: escravo aceitará
STOP#	1		x	Escravo quer interromper transação imediatamente
PERR#	1			Erro de paridade de dados detectado pelo receptor
SERR#	1			Erro de paridade de endereço ou erro de sistema detectado
REQ#	1			Arbitragem de barramento: requisição de propriedade de barramento
GNT#	1			Arbitragem de barramento: concessão de propriedade de barramento
RST#	1			Restaura o sistema e todos os dispositivos



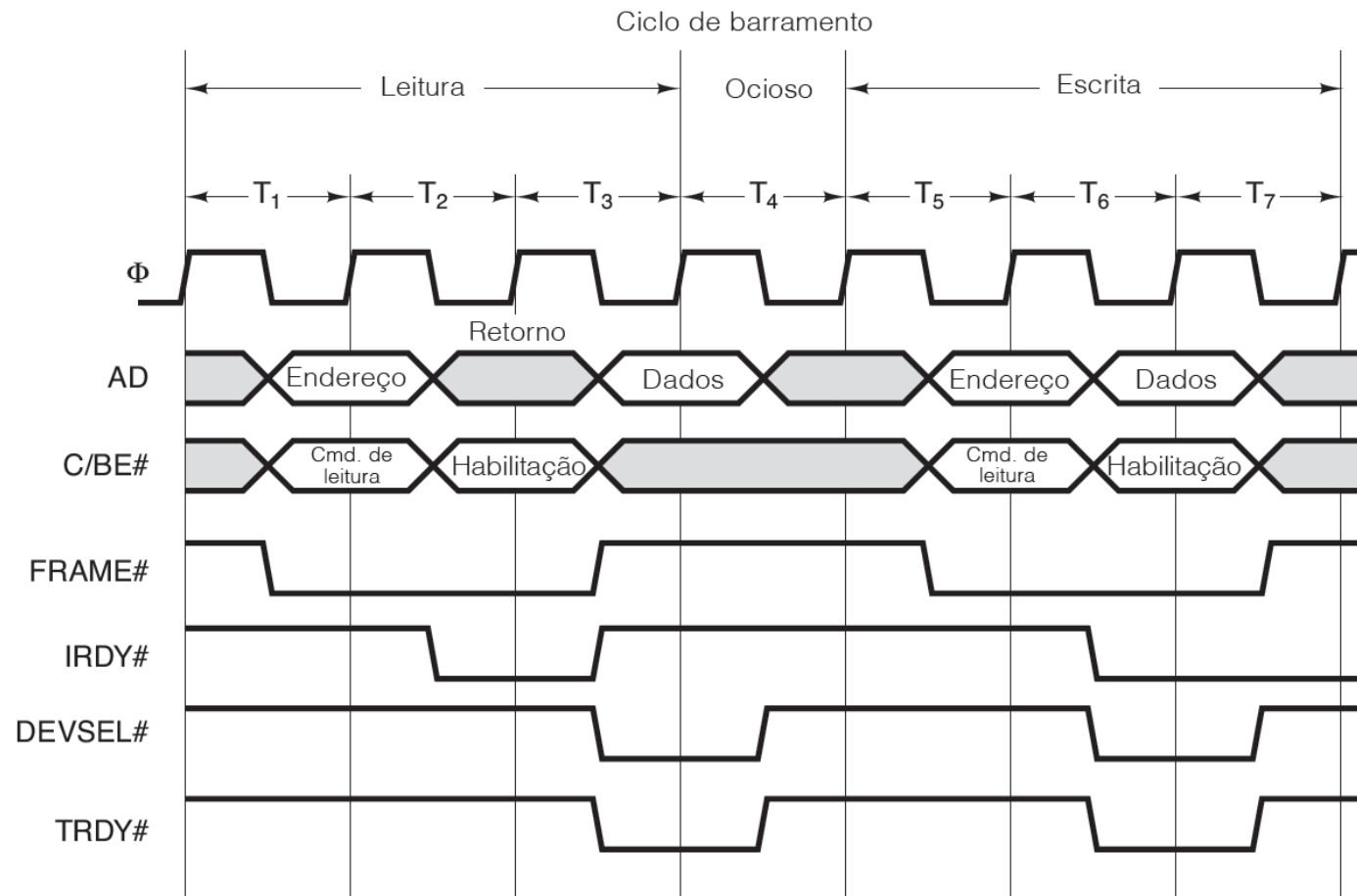
# Exemplos de barramentos

- Sinais opcionais de barramento PCI.

Sinal	Linhas	Mestre	Escravo	Descrição
REQ64#	1	x		Requisição para realizar transação de 64 bits
ACK64#	1		x	Permissão concedida para uma transação de 64 bits
AD	32	x		32 bits adicionais de endereço ou dados
PAR64	1	x		Paridade para os 32 bits extras de endereço/dados
C/BE#	4	x		4 bits adicionais para habilitações de bytes
LOCK	1	x		Trava o barramento para permitir múltiplas transações
SBO#	1			Presença de dados em uma <i>cache</i> remota (para um multiprocessador)
SDONE	1			Escuta realizada (para um multiprocessador)
INTx	4			Requisição de uma interrupção
JTAG	5			Sinais de testes IEEE 1149.1 JTAG
M66EN	1			Ligado à energia ou ao terra (66 MHz ou 33 MHz)

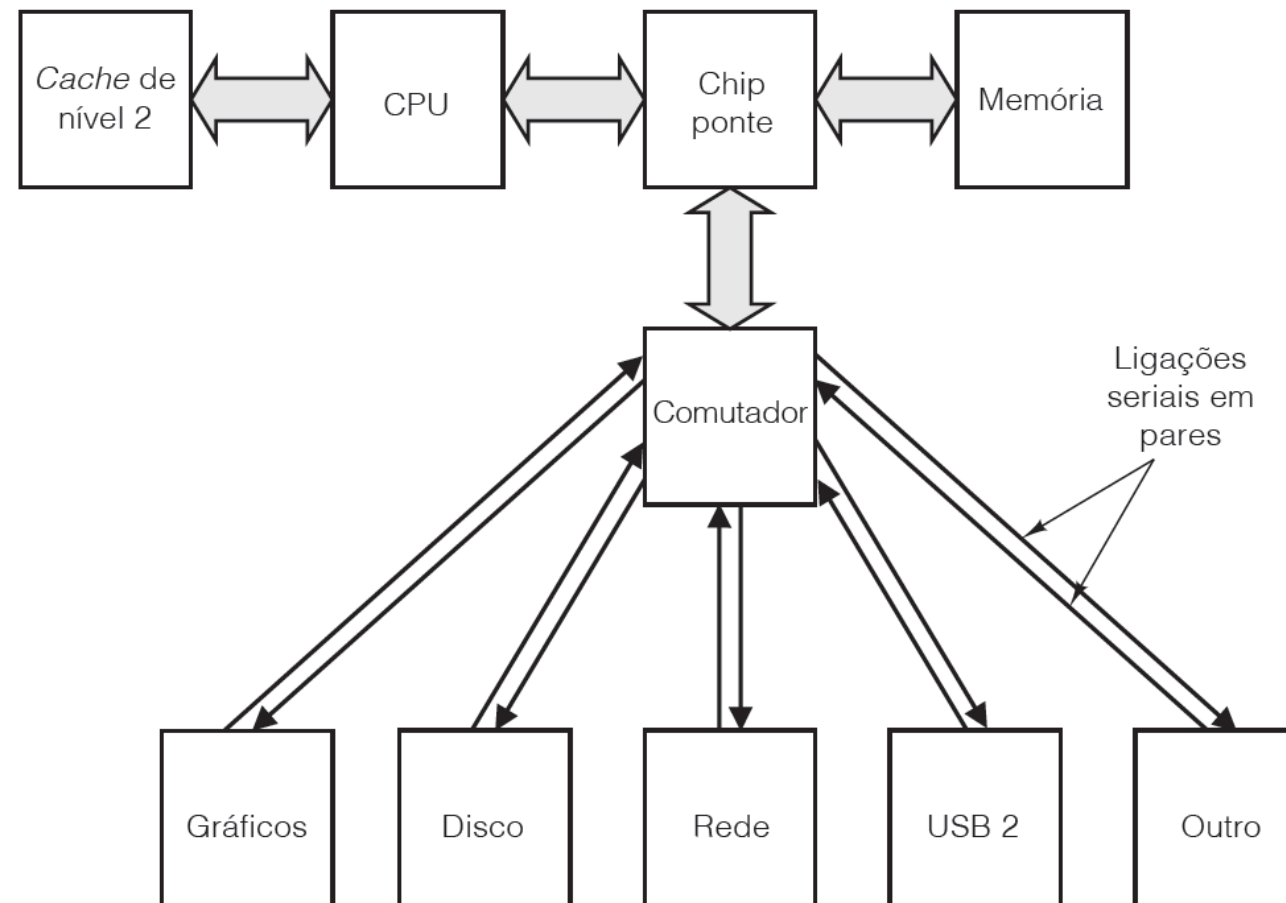
# Exemplos de barramentos

- Exemplos de transações de barramento PCI de 32 bits.



# Exemplos de barramentos

- Sistema PCI express típico.



# Barramento PCI

---

## Trabalho

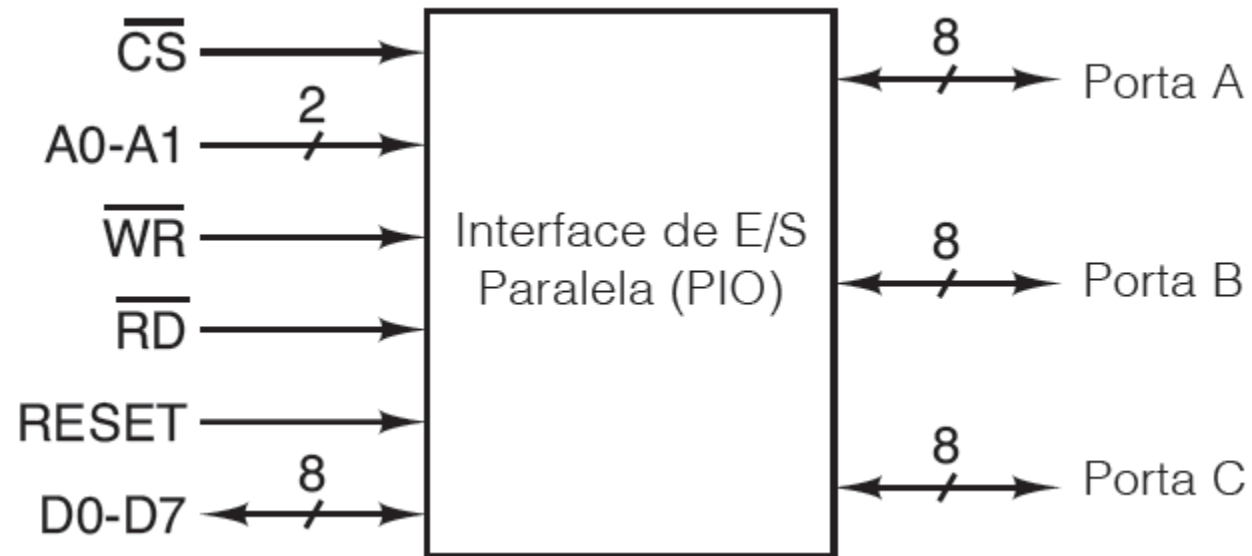
**Identifique a sessão exata no livro do Tanenbaum ou do Stallings que trata de barramentos PCI e leia. A seguir, escreva um artigo sobre o assunto. Utilize fontes adicionais se julgar necessário.**

# Exemplos de barramentos

- Um sistema USB consiste em um **hub-raiz** (*root hub*) que é ligado ao barramento principal.
- Em termos lógicos, o sistema USB pode ser visto como um conjunto de ramificações que saem do *hub-raiz* para os dispositivos de E/S.
- Exatamente a cada  $1,00 \pm 0,05$  ms, o *hub-raiz* transmite um novo quadro para manter todos os dispositivos sincronizados em relação ao tempo.
- O USB suporta quatro tipos de quadros: de controle, isócrono, de volume e de interrupção.

# Interfaces PIO

- Uma interface PIO (**Parallel Input/Output – entrada e saída paralela**) típica é o Intel 8255A:

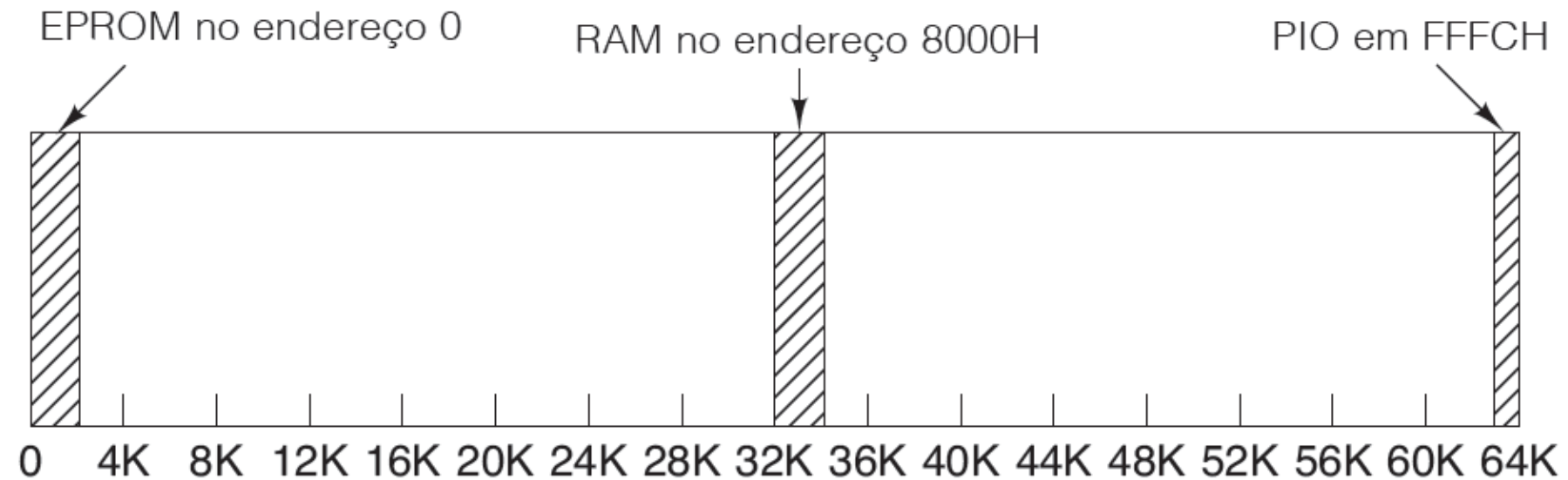


# Interfaces PIO

- A interface PIO pode ser selecionada de um entre dois modos: como um verdadeiro dispositivo de E/S ou como parte da memória.
- Uma opção possível é mostrada na figura a seguir.
- A EPROM ocupa endereços até 2 K, a RAM ocupa endereços de 32 KB a 34 KB e a PIO ocupa os 4 bytes mais altos do espaço de endereço, 65.532 a 65.535.
- Com as designações de endereço da figura, a EPROM deve ser selecionada por quaisquer endereços de memória de 16 bits da forma 00000xxxxxxxxxxx (binário).

# Interfaces PIO

- Localização da EPROM, RAM e PIO em nosso espaço de endereço de 64 KB.



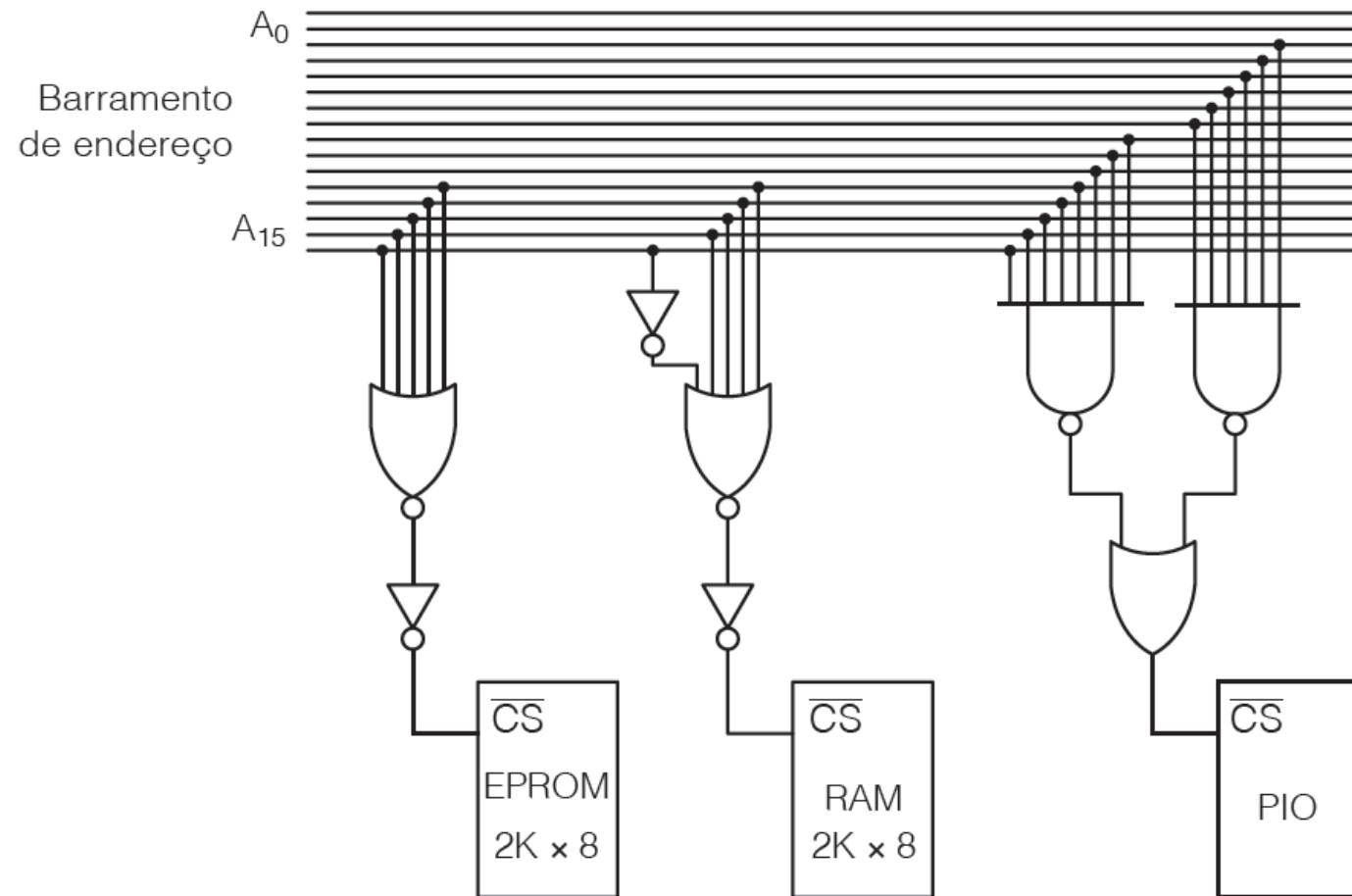


# Interfaces PIO

- O mesmo princípio pode ser usado para a RAM.
- Contudo, a RAM deve responder a endereços binários da forma 10000xxxxxxxxxxx.
- Se o computador tiver apenas uma CPU, dois chips de memória e a PIO, podemos usar um truque para conseguir uma decodificação de endereço muito mais simples.
- Esse truque se baseia no fato de que todos os endereços da EPROM, e somente endereços da EPROM, têm um 0 no bit de ordem alta, A15. Por conseguinte, basta ligar CS a A15 diretamente.

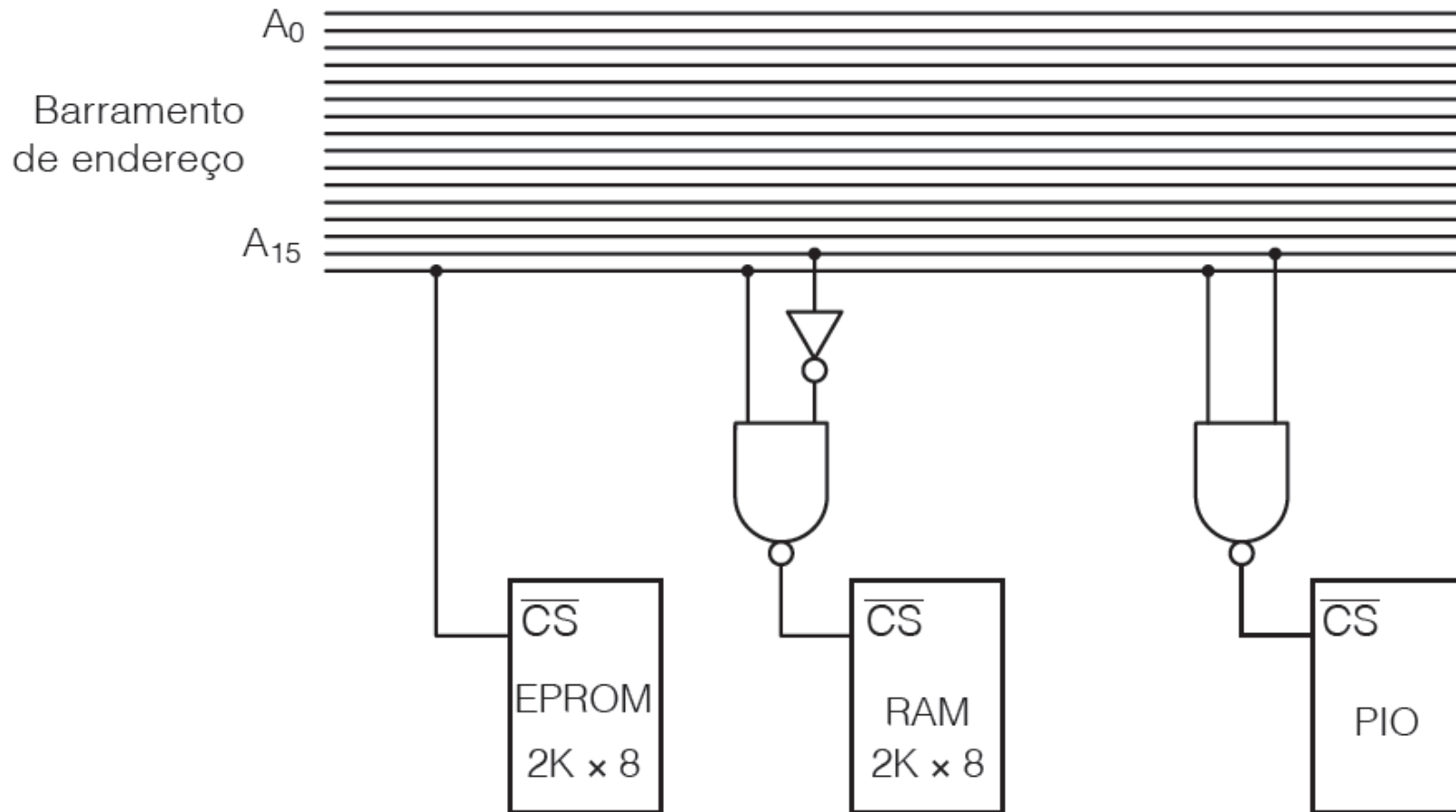
# Interfaces PIO

- Decodificação total de endereço.



# Interfaces PIO

- Decodificação parcial de endereço.



# Assembly & Assembler

Assembly é uma linguagem de programação legível para nós humanos, com maior facilidade de assimilação, desenvolvimento e análise.

Assembler é um compilador de programas Assembly, ou seja, ele é um tradutor, um programa ou conjunto de programas que transformará a linguagem de alto nível entendida pelo homem para um linguagem de baixo nível, a linguagem de máquina. O Assembler é o responsável por gerar o arquivo binário a partir do código Assembly.

# Assembly & Assembler

- Para o computador a instrução-máquina IA-21 (10110000 01100001)
- representação equivalente em instruções mnemónicas MOV AL, 61h.
  - Tal instrução ordena que o valor hexadecimal 61 (97, em decimal) seja movido para o registor 'AL'.
- A conversão da linguagem de montagem para o código de máquina é feita pelo montador ou assembler, que é basicamente um tradutor de comandos, sendo mais simples que um compilador.