



# Arquitetura e Organização de computadores

---

ENGENHARIA DA COMPUTAÇÃO – UFC/SOBRAL

Prof. Danilo Alves  
danilo.alves@alu.ufc.br

# *Chips* Processadores

---

- Cada processador possui uma pinagem característica. Esses pinos podem ser divididos em: endereços, dados e controle.
- Esses pinos são conectados aos barramentos e então ligados a chips de memória e de E/S.
- O processador se comunica com a memória e com os dispositivos de E/S através de envio e recebimento de sinais em seus pinos.



# *Chips* Processadores

---

- Exemplo: Busca de uma instrução

1. A CPU coloca o endereço da memória da instrução nos respectivos pinos.
2. Em seguida, linhas de controle são ativadas para indicar à memória sobre a operação de leitura.
3. A memória responde colocando em seus pinos (e, conseqüentemente barramento) de dados a palavra selecionada, enviando também sinais indicando o sucesso da operação.
4. O processador aceita a palavra (uma instrução, nesse caso) e executa a instrução recebida.



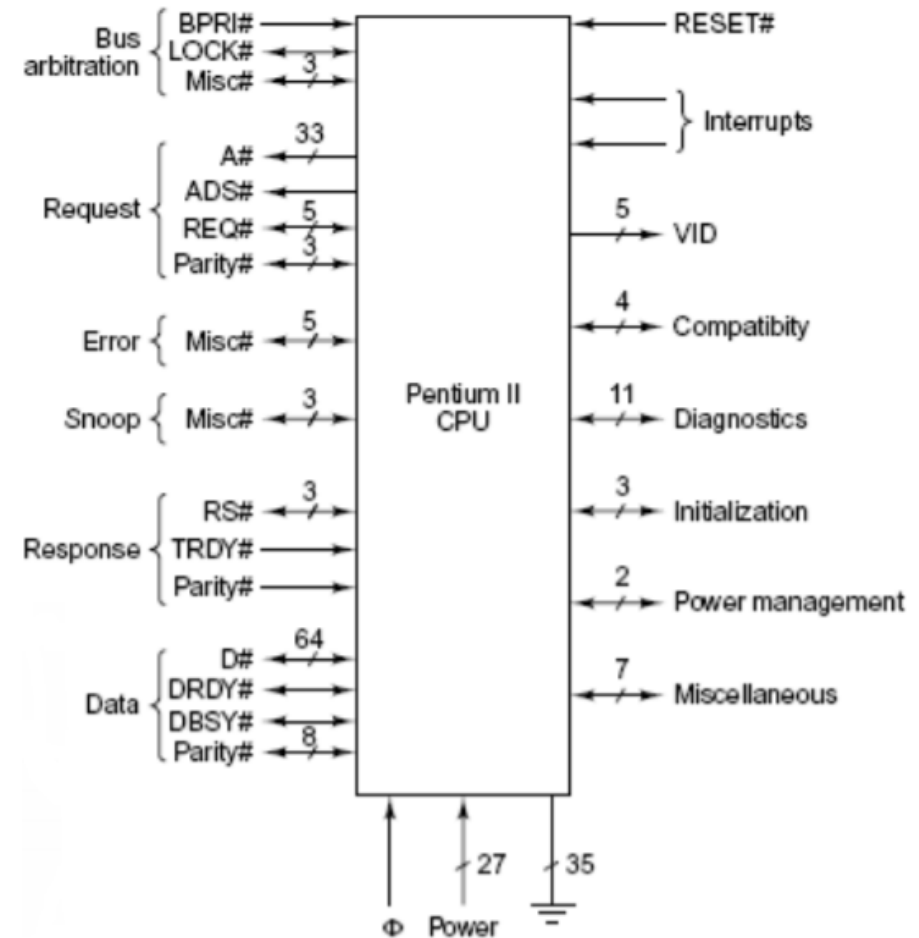
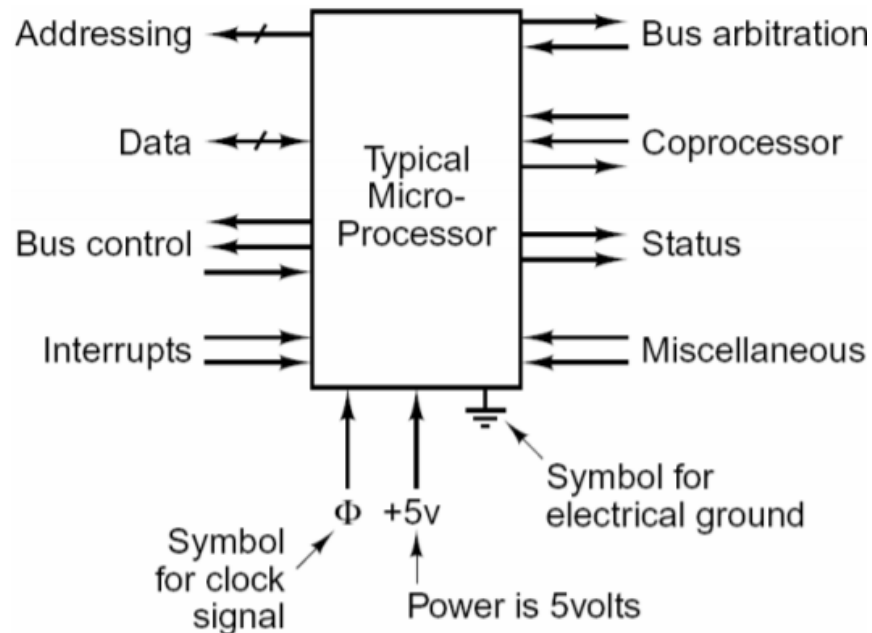
# *Chips* Processadores

---

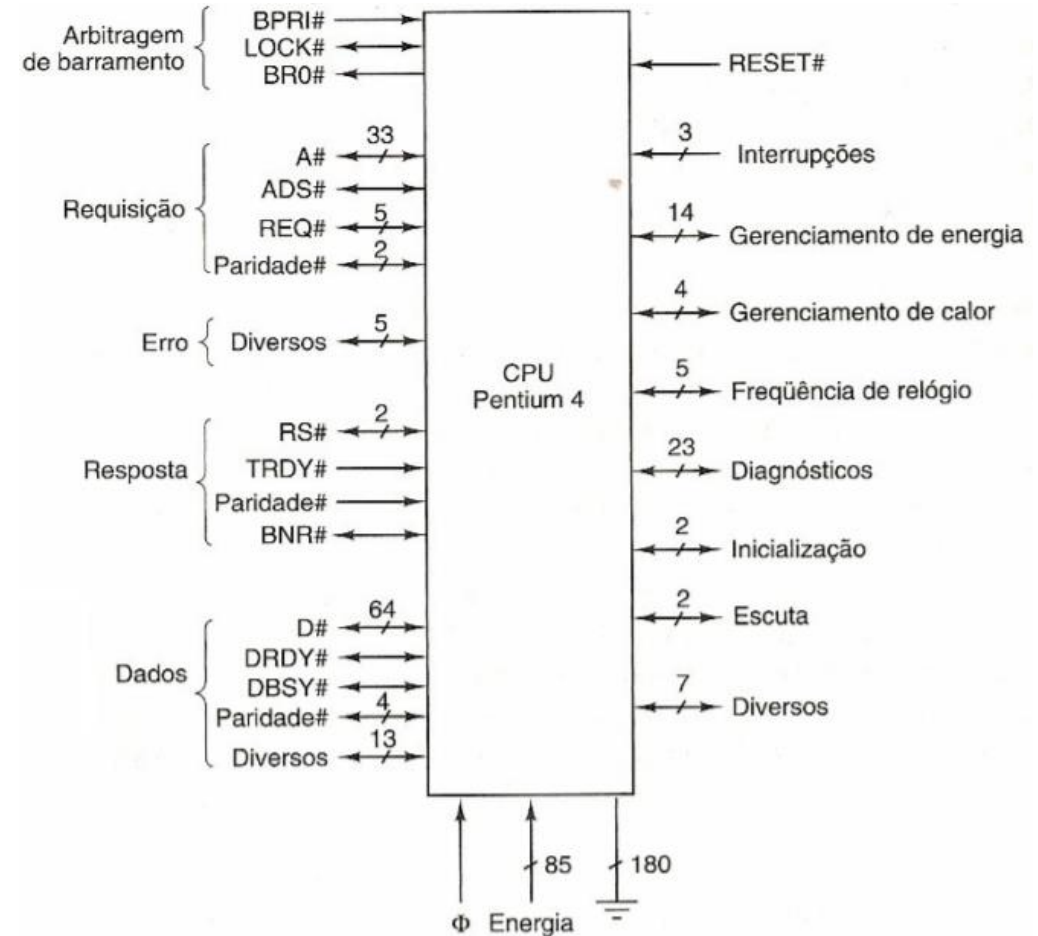
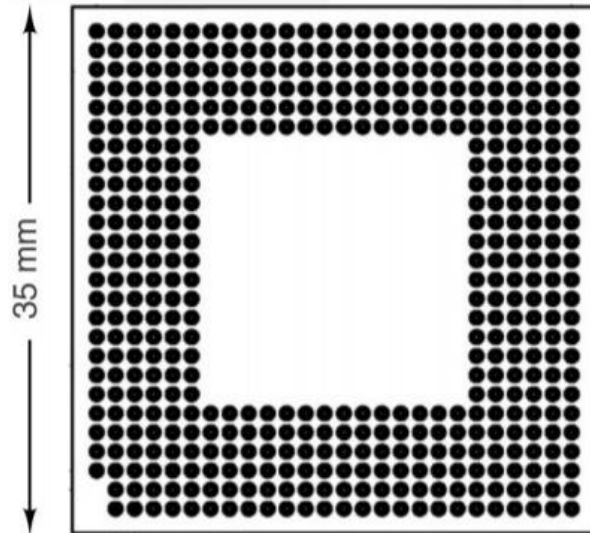
- Além dos pinos de dados e de endereços, existem os pinos de controle.
- Controle: regulam o fluxo e a temporização dos dados que entram e saem do processador, e outros usos.
- Os pinos de controle podem ser divididos (de maneira geral):
  - Controle do Barramento
  - Interrupções
  - Arbitragem do Barramento
  - Sinalização para o co-processador
  - Status
  - Outros (informações de *status* de dispositivos, reboot do processador, compatibilidade com dispositivos, etc)



# *Chips* Processadores



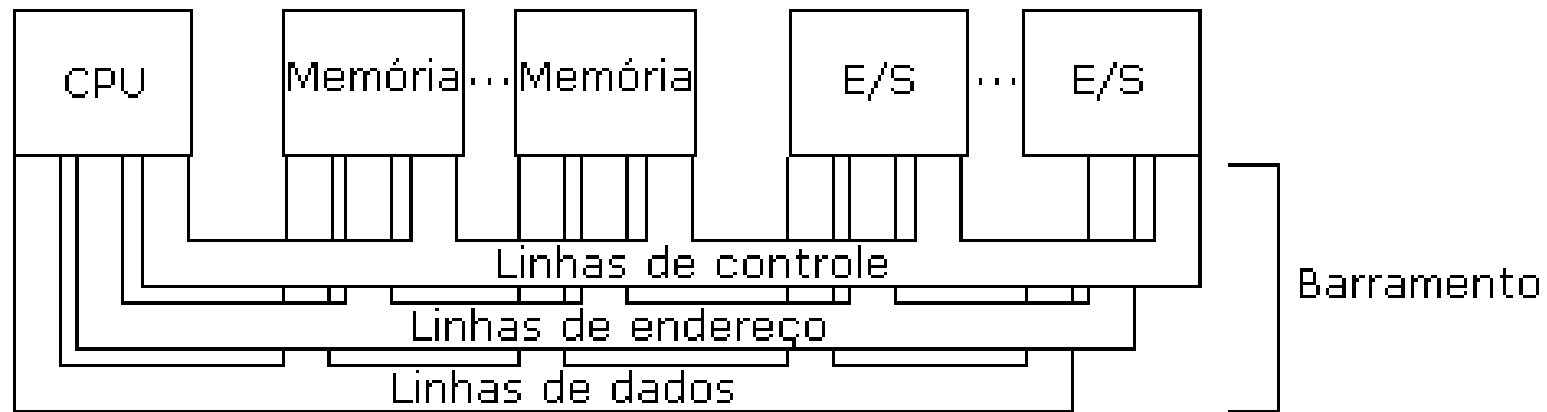
# Chips Processadores



# Barramentos

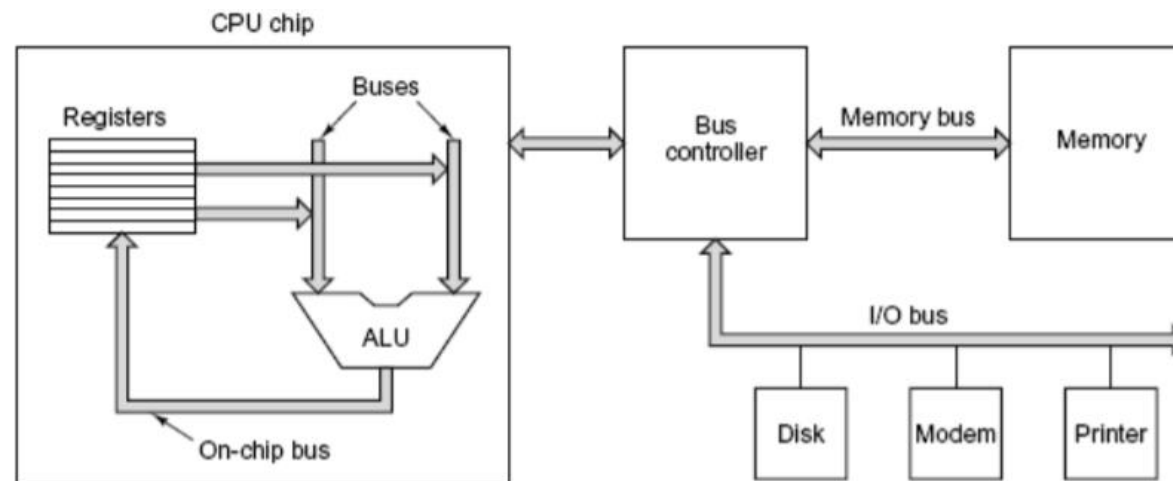
---

- Barramento é o caminho pelo qual os dispositivos se comunicam
- Envio de sinais (dígito 0 ou 1)
- Frequentemente agrupado (8,16,32,64 linhas)



# Barramentos

- Os barramentos são agrupados segundo suas funções, podendo ser internos ao próprio processador ou interligando-o a outros dispositivos.
- Protocolo de Barramento: Conjunto de regras bem definidas que especificam o funcionamento do barramento, de maneira a permitir que outros dispositivos possam se conectar ao processador





# Estrutura de barramento

---

- Estrutura de barramento
- Simples
  - Unibus
- Múltipla
  - Barramento para Controle
  - Barramento para Endereço
  - Barramento para Dados



# Problemas com Barramento Simples

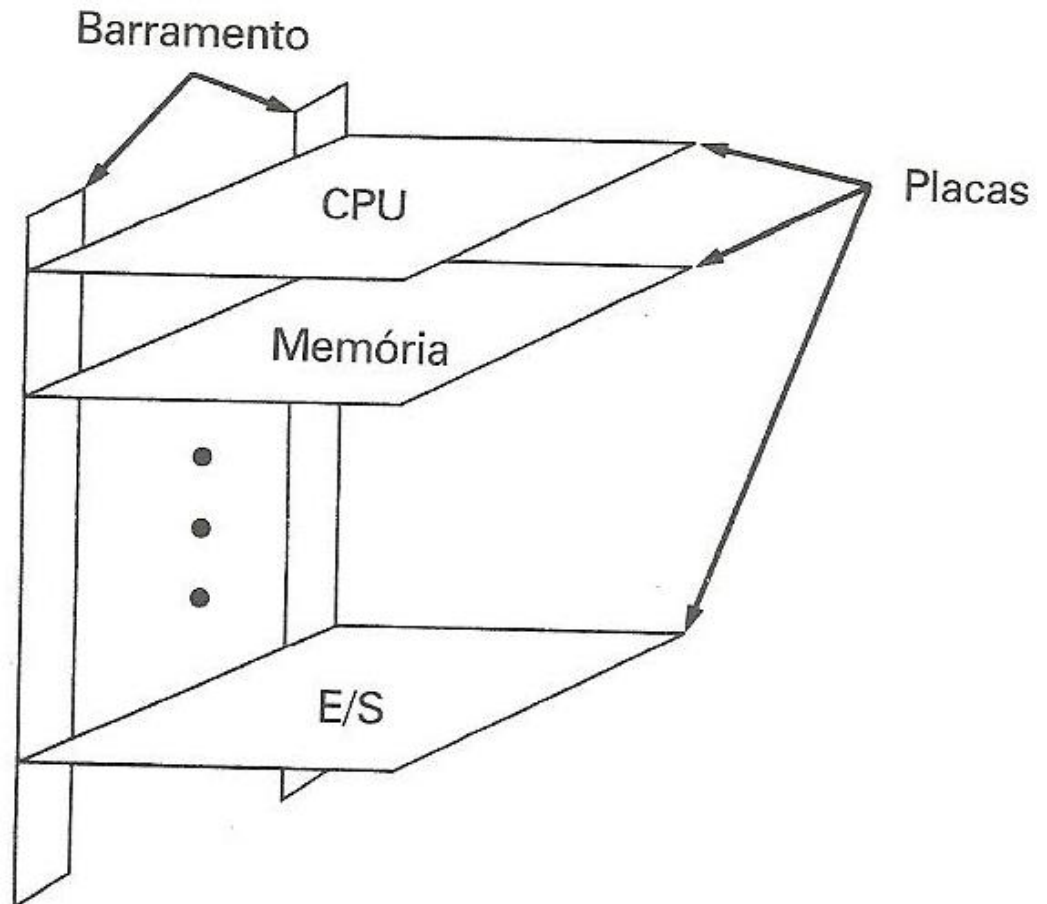
---

- Muitos dispositivos em um barramento ocasiona:
  - Delays de transmissões
    - Transferência de grande volume de dados afeta a performance
- Maioria dos sistemas usa barramento múltiplo para resolver esse problema



# Estrutura de barramento

- Estrutura típica de uma arquitetura de barramentos



# Barramento de Dados

---

- Transporta dados
- Lembre-se que não há diferença entre dados e instruções neste nível
- Largura é determinante para a performance
  - 8, 16, 32, 64 bit



# Barramento de Endereços

---

- Identifica a fonte ou o destino dos dados
- Exemplo: CPU precisa ler uma instrução/dados de uma determinada localização na memória.
- Largura Barramento determina a capacidade máxima de memória do sistema.
- Exemplo: 8080 tem barramento de endereço de 16 bit dando 64k endereços ( $2^{16}/1024=64$ ).



# Barramento de Controle

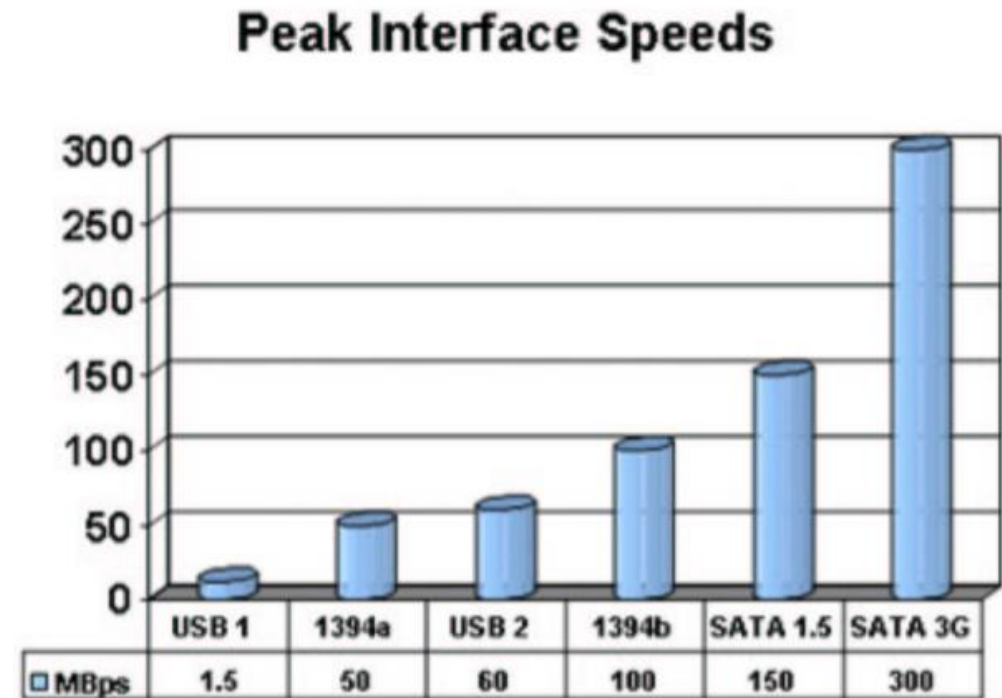
---

- Controla e temporiza a informação
  - Sinais de leitura/escrita na memória
  - Pedidos de interrupções
  - Sinais do Clock



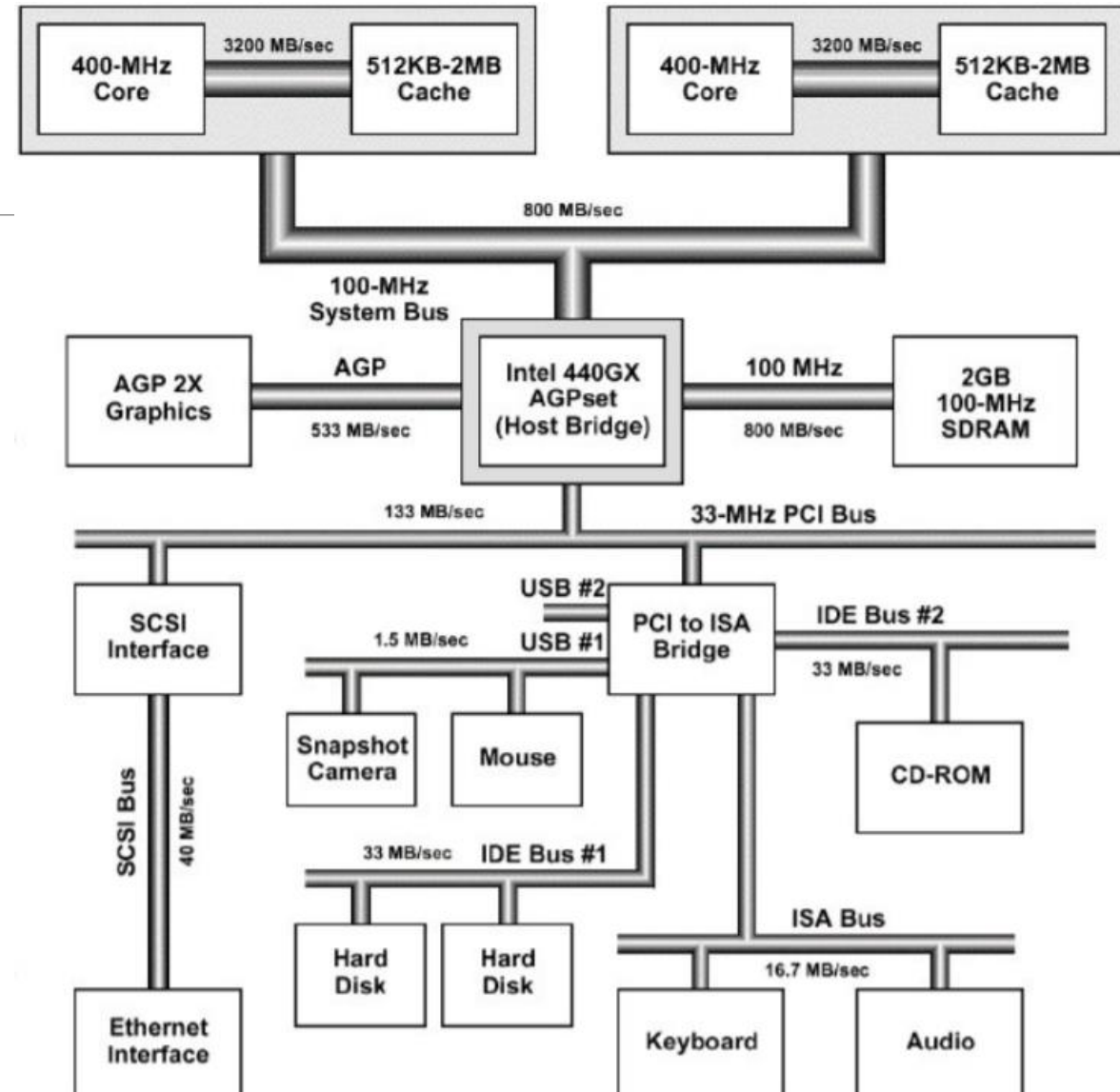
# Barramentos

- Existem diferentes tipos proprietários e padrões de barramentos
  - ISA
  - PCI, PCI-Express
  - SCSI
  - USB 1, 2
  - AGP
  - FireWire
  - SATA
  - ...



# Barramentos

- Arquiteturas de barramentos baseadas em pontes
  - Interfaceamento
  - Transferências de dados simultâneas





# Barramentos

---

- Fios são compartilhados
  - Somente um dispositivo pode enviar dados em um momento
  - Todos escutam, mas em geral só um lê os dados
- Mestres
  - Dispositivos ativos que comandam a transferências sobre o barramento
- Escravos
  - Elementos passivos... devem aguardar solicitações de serviços
- Dependendo da situação um dispositivo/CPU pode ser o mestre ou o escravo



# Barramentos

---

- Ex:
- Quando a CPU solicita a um controlador de disco um bloco de dados. A CPU é o mestre e o controlador de disco, o escravo.
- Em resposta, através de DMA, o controlador de disco transfere o bloco de dados para a memória... Controlador é o mestre, e a memória, escravo.
- Memória sempre será escravo



# Barramentos

---

- De maneira geral, os sinais dos dispositivos precisam ser amplificados para transitar pelo barramento:
  - Os mestres de barramentos são intermediados por dispositivos **controladores de barramento** (amplificador)
  - Os escravos são intermediados por chips **receptores do barramento**.
  - Para os dispositivos que podem atuar das duas maneiras, utilizam-se os **transceptores** (transcievers)



# Barramentos

---

- Cada barramento tem uma banda passante (BW) expressa em capacidade de transações (bytes/seg)
- Principais aspectos de projetos de barramento que afetam a BW e o custo de um barramento
  - **Largura do barramento**
  - **Metodologia de Temporização**
  - **Arbitragem de Barramento**
  - **Operações** que podem ser realizadas sobre o barramento



# Largura do Barramento

---

- Corresponde ao número de linhas de um barramento.
- Exemplo:
- Quanto maior o número de linhas de endereço o barramento tiver, mais memória poderá ser diretamente endereçada pelo processador.
- Ocupa mais espaço físico (na placa-mãe) e conectores maiores.
- Após decidida a largura do barramento em um projeto, a possibilidade de expansão do sistema não é trivial.



# Largura do Barramento

---

- Largura de Banda Passante x Largura do Barramento
  - Aumentar a banda passante?
    - Aumentar a largura do barramento (mais bits por transferência)
    - Diminuir o ciclo do barramento (mais transferências por segundo)
- É possível o aumento da velocidade do barramento. Entretanto a tarefa é dificultada em virtude das diferenças de velocidades nas linhas dos barramentos
  - Fenômeno conhecido como **Atraso Diferencial do Barramento**
  - Tal problema é agravado quando a velocidade do barramento é aumentada



# Largura do Barramento

---

- **Barramentos Multiplexados:** Utilizados ora para transferência de endereços, ora para transferência de dados.
  - Visa controlar o problema de barramentos muito longos
  - O sistema é mais lento!
- Temporização do Barramento
- Barramento Síncrono:
  - Uma de suas linhas é alimentada por um clock do barramento.
  - Todas as atividades do barramento síncrono gastam um número inteiro de ciclos do sinal
  - Ciclos do Barramento



# Temporização do Barramento

---

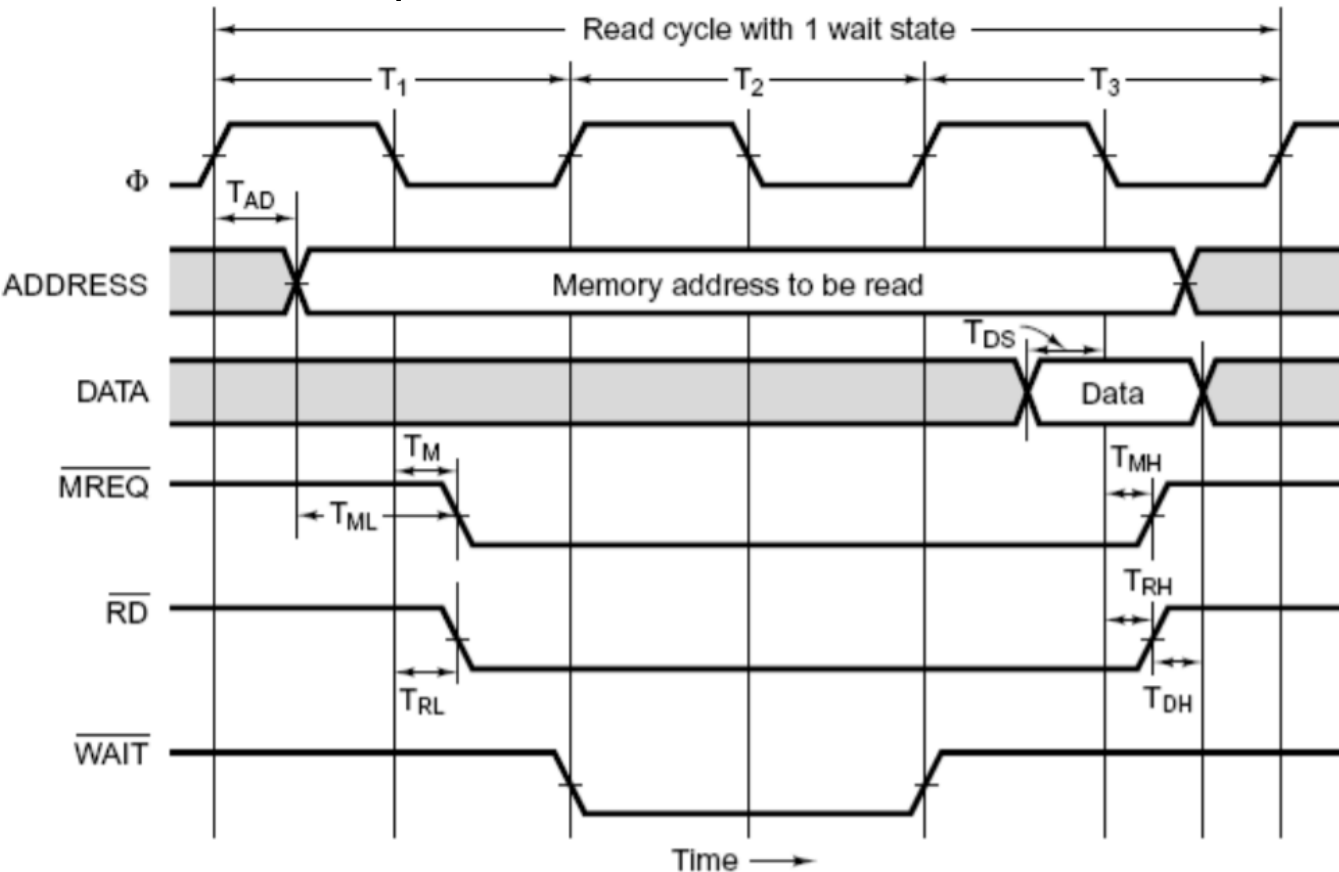
- Exemplo: CPU (mestre) pede a leitura de um dado na memória (escravo)
  - Clock de Barramento de 100Mhz -> Ciclo de Barramento de 10 ns.
  - Tempo de Leitura da Memória: 15ns (a partir da estabilidade do endereço)
    - Serão necessários 3 ciclos de barramento para a leitura da palavra ...
  - Barramentos: **ADDRESS** e **DATA**
  - Sinais (barramento de controle):
    - $\overline{MREQ}$  (*Memory Request*)
    - $\overline{RD}$  (*Read*)
    - $\overline{WAIT}$





# Temporização do Barramento

## Exemplo de Funcionamento de Leitura no Barramento Síncrono



Símbolo	Parâmetro	Mín.	Máx.	Unidade
$T_{AD}$	Atraso de saída do endereço		4	nsec
$T_{ML}$	Endereço estável antes de $\overline{MREQ}$	2		nsec
$T_M$	Atraso de $\overline{MREQ}$ desde a borda descendente de $\Phi$ em T1		3	nsec
$T_{RL}$	Atraso de RD desde a borda descendente de $\Phi$ em T1		3	nsec
$T_{DS}$	Tempo de ajuste dos dados antes da borda descendente de $\Phi$	2		nsec
$T_{MH}$	Atraso de $\overline{MREQ}$ desde a borda descendente de $\Phi$ em T3		3	nsec
$T_{RH}$	Atraso de $\overline{RD}$ desde a borda descendente de $\Phi$ em T3		3	nsec
$T_{DH}$	Tempo de sustentação dos dados desde a negação de $\overline{RD}$	0		nsec



# Temporização do Barramento

---

- Barramento Síncrono:
  - O primeiro sinal (MREQ) indica que a memória (e não um I/O) está sendo acessada.
  - O sinal RD indica que a operação é de leitura ou de escrita.
  - Como a memória leva 15ns para ler, a partir da estabilização do endereço, a memória não poderá fornecer o valor solicitado durante T2. A linha WAIT é ativada com o intuito de informar ao processador que a memória está trabalhando.



# Temporização do Barramento

- Especificação de alguns tempos críticos na leitura
  - $T_{AD}$  indica que a CPU tem, a partir da transição positiva de  $T_1$ , no máximo 4 ns para colocar o endereço a ser lido no barramento de endereços
  - $T_{DS}$  indica que os dados devem ser colocados nas linhas de dados (pela memória) no mínimo 2 ns antes da transição negativa de  $T_3$

Símbolo	Parâmetro	Mín.	Máx.	Unidade
$T_{AD}$	Atraso de saída do endereço		4	nsec
$T_{ML}$	Endereço estável antes de $\overline{MREQ}$	2		nsec
$T_M$	Atraso de $\overline{MREQ}$ desde a borda descendente de $\Phi$ em T1		3	nsec
$T_{RL}$	Atraso de RD desde a borda descendente de $\Phi$ em T1		3	nsec
$T_{DS}$	Tempo de ajuste dos dados antes da borda descendente de $\Phi$	2		nsec
$T_{MH}$	Atraso de $\overline{MREQ}$ desde a borda descendente de $\Phi$ em T3		3	nsec
$T_{RH}$	Atraso de $\overline{RD}$ desde a borda descendente de $\Phi$ em T3		3	nsec
$T_{DH}$	Tempo de sustentação dos dados desde a negação de $\overline{RD}$	0		nsec



# Temporização do Barramento

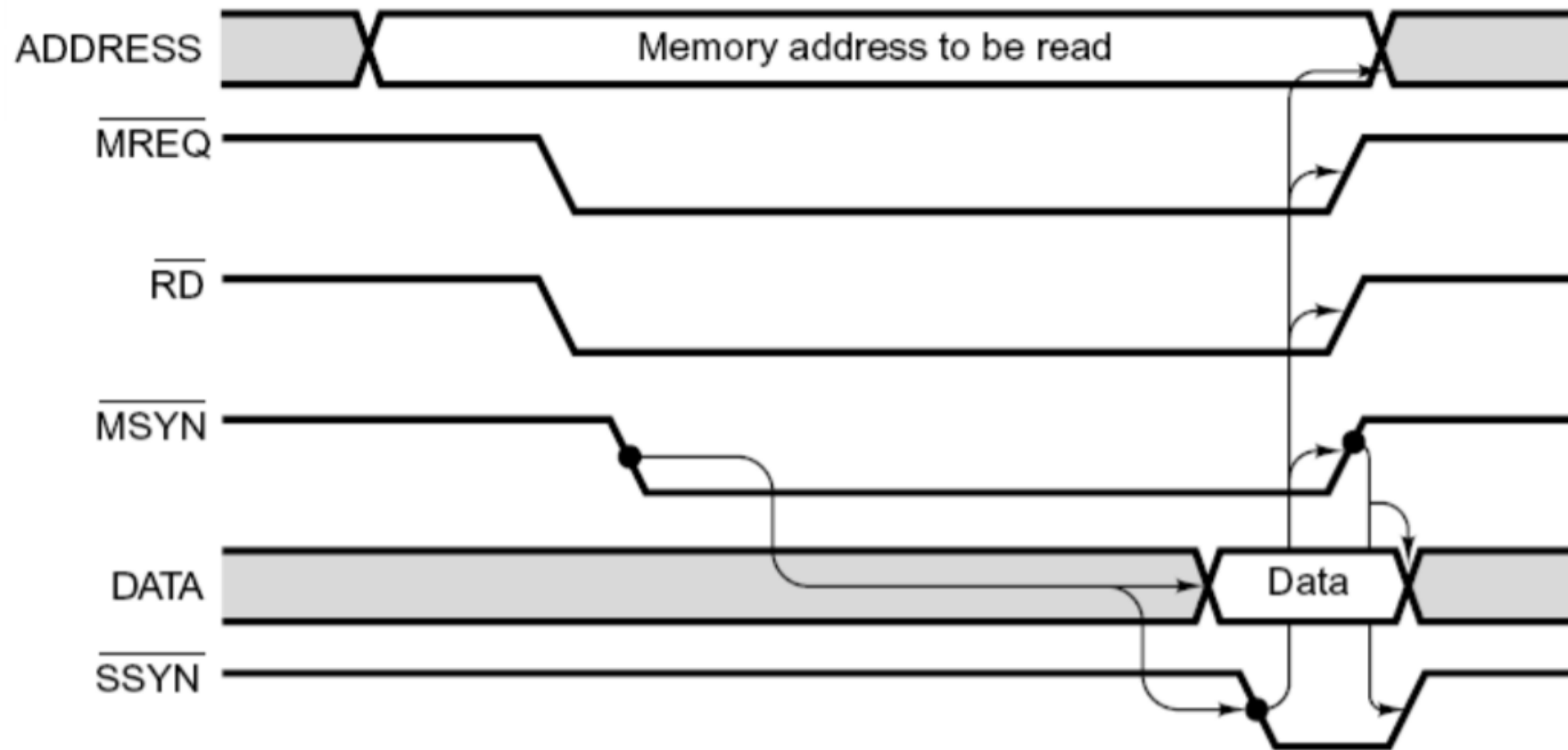
---

- **Barramento Assíncrono:** Não possui nenhuma linha de clock do barramento. Os ciclos de barramento podem ser de qualquer tamanho e não precisam ser os mesmos para todos os pares de dispositivos.
- Exemplo: CPU (mestre) pede a leitura de um dado na memória (escravo)
  - Em vez de amarrar todos os eventos ao clock, quando o mestre ativa o endereço, os sinais  $\overline{\text{MREQ}}$  e  $\overline{\text{RD}}$ , um sinal especial  $\overline{\text{MSYN}}$  (*Master SYNchronization*) deve ser ativado.
  - Quando o escravo enxerga o  $\overline{\text{MSYN}}$  ativo, ele faz seu trabalho o mais rápido possível. Ao terminar, o sinal  $\overline{\text{SSYN}}$  (*Slave SYNchronization*) é ativado.
  - O sinal  $\overline{\text{SSYN}}$  ativo informa ao mestre que os dados estão disponíveis.



# Temporização do Barramento

- Barramento Assíncrono:



# Temporização do Barramento

---

- **Barramento Assíncrono**

- A operação de um conjunto de sinais que se relacionam dessa maneira é regida por um protocolo conhecido como **Aperto de Mão (Handshake) Completo**.
- Cada evento depende de um evento anterior e não de um pulso de clock.
- Seus passos são:
  1.  $\overline{MSYN}$  é ativado
  2.  $\overline{SSYN}$  é ativado em resposta a  $\overline{MSYN}$
  3.  $\overline{MSYN}$  é negado em resposta a  $\overline{SSYN}$
  4.  $\overline{SSYN}$  é negado em resposta à negação de  $\overline{MSYN}$



# Temporização do Barramento

---

- **Barramento Síncrono X Barramento Assíncrono**
- **Barramento Síncrono**
- O barramento síncrono é mais simples de se trabalhar, em virtude dos intervalos discretos de tempo. Tudo acontece em múltiplos de ciclo de clock do barramento.
- Problema: se uma CPU e uma memória forem aptos a completar uma transferência em 3,1 ciclos, será necessário esperar quatro ciclos se completarem.
- Se as velocidades dos dispositivos forem heterogêneas, o desempenho do sistema estará limitado pelo desempenho do dispositivo mais lento, ou seja, o barramento deve operar de acordo com a velocidade desse dispositivo.
- Ganhos de performance em processadores e dispositivos de I/O ficam limitados ao desempenho do barramento síncrono.



# Temporização do Barramento

---

- **Barramento Síncrono X Barramento Assíncrono**

- **Barramento Assíncrono**

- A lentidão de um determinado par mestre-escravo não afeta a velocidade de outros pares que poderão ser mais rápidos quanto mais lentos que o outro par.
    - A vantagens de desempenho com relação ao barramento síncrono.
    - Em geral, um sistema assíncrono é mais complicado e mais custoso de se construir do que um sistema síncrono.





# Arbitragem do Barramento

---

- O que acontece se dois ou mais dispositivos desejarem tornar-se mestres (tomar o controle do barramento) ao mesmo tempo?
- Mecanismos de **Arbitragem do Barramento**
  - Resolver conflitos e evitar “o caos”
  - Em geral, E/S tem preferência
  - Os discos e demais dispositivos cuja operação dependa de movimento mecânico de suas partes não podem ter seu trabalho interrompido no meio.
  - **Roubo de Ciclo**
    - Uma E/S “toma” ciclos de barramento da CPU para executar sua tarefa
    - Podem ser mecanismos centralizados e descentralizados.



# Arbitragem do Barramento

---

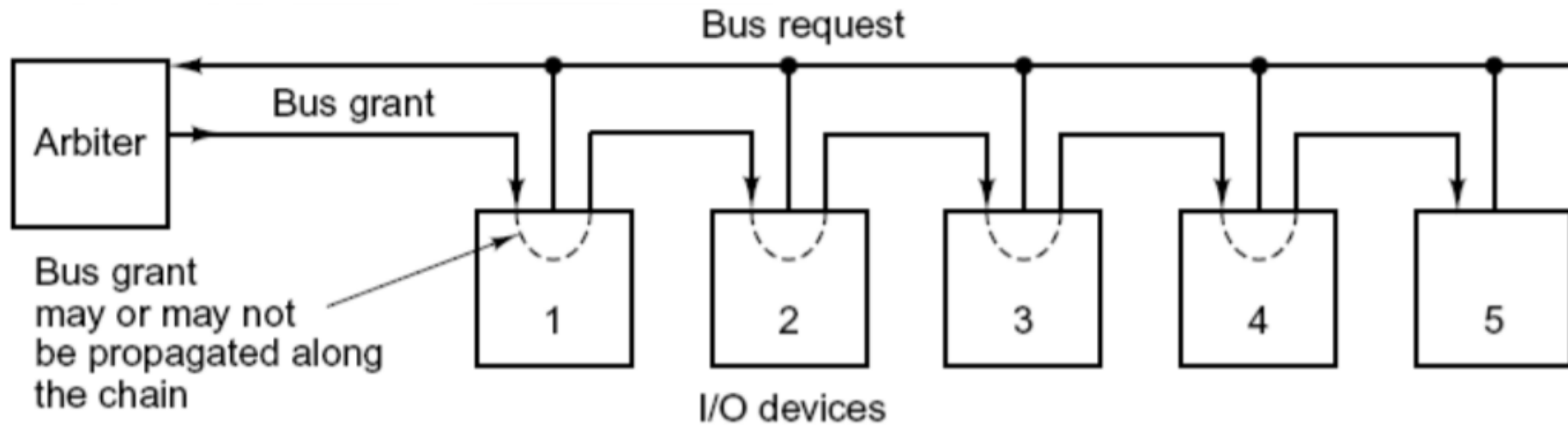
- **Abordagem Centralizada**

- Um único árbitro determina quem será o próximo mestre do barramento. Ele é capaz de distinguir entre ausência e presença de **requisição de uso do barramento (bus request)**.
- Pode ser o processador ou um chip separado para ser o árbitro.
- Quando o árbitro enxerga a requisição do barramento, ele envia um sinal de **garantia de uso do barramento (bus grant)** por meio de uma linha dedicada.



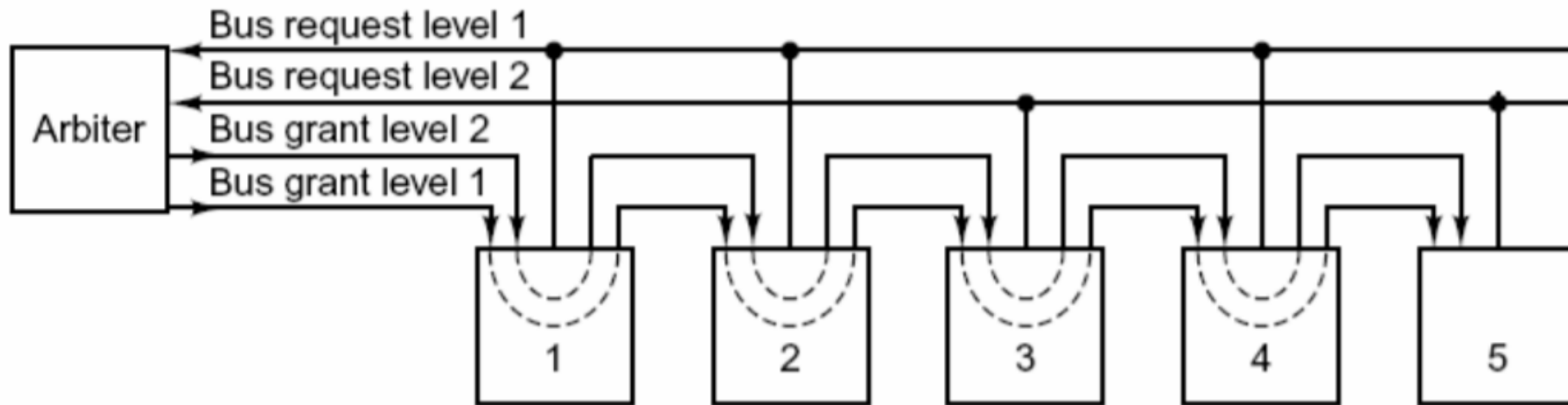
# Arbitragem do Barramento

- A arbitragem é então definida pelo esquema conhecido como Daisy Chaining. (*daisy*: margarida; *chain*: encadear)
- **Daisy Chaining**
  - Tem a propriedade de estabelecer implicitamente uma prioridade para uso do barramento.
  - Quanto mais perto fisicamente do árbitro estiver um dispositivo, mais prioridade ele terá para usar o barramento.



# Arbitragem do Barramento

- Há a possibilidade do estabelecimento **físico** de linhas de requisição e de garantia de uso do barramento com níveis de prioridades diferentes.
- Cada dispositivo é ligado a um dos níveis de prioridade para requisição do barramento.



# Arbitragem do Barramento

---

- **Abordagem Descentralizada**

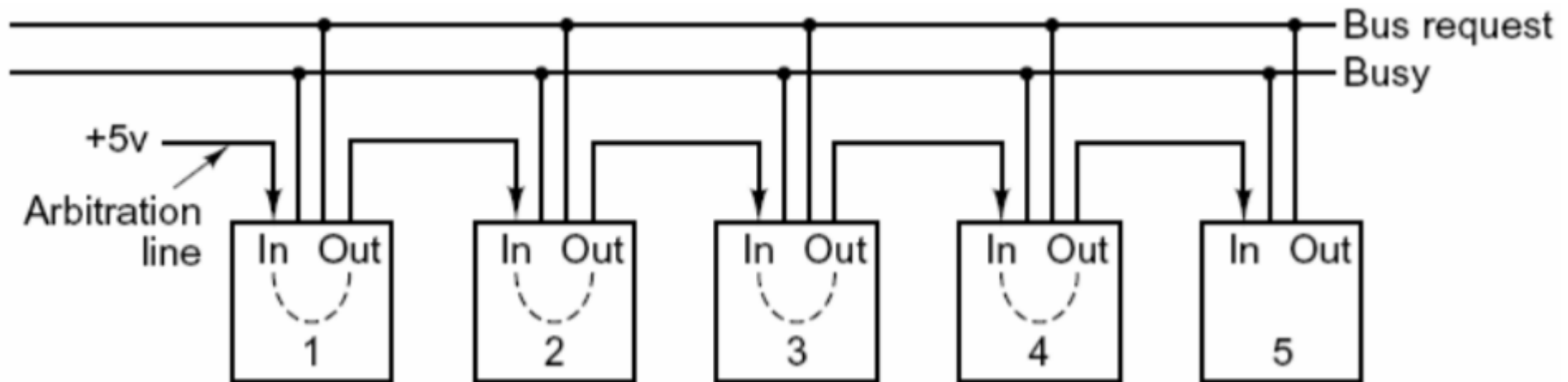
- Mecanismo 1:
- Há, por exemplo, 16 linhas de requisição de barramento, com diferentes prioridades cada. Quando um dispositivo precisar do barramento, ele deve ativar uma das linhas.
- Todos os dispositivos monitoram todas as linhas. Ao final do ciclo de requisição, cada dispositivo sabe se foi ou não ele que requisitou a linha de maior prioridade.
- Apresenta muitas linhas de requisição de barramento, mas não possui o custo de se ter um árbitro.



# Arbitragem do Barramento

- **Abordagem Descentralizada**

- Mecanismo 2:
- Apresenta três linhas: **Bus Request**; **Busy** (ativada pelo mestre atual do barramento); e a linha de arbitragem (**arbitration line**).
- A linha de arbitragem é conectada a todos os dispositivos no modo *daisy chaining* e permanece ativa enquanto nenhum dispositivo precisar do barramento.



# Operações no Barramento

---

- Outros tipos de ciclos de barramento:
  - Ciclo de Leitura de Memória Cache
  - Em Sistemas Multiprocessados (prioridade entre processadores)
  - **Ciclo que trata Interrupções**



# Barramento

---

- **Canais de Comunicação**

- Quando o I/O é serial, apenas um bit pode ser transferido de cada vez, podendo ser de maneira síncrona ou assíncrona
- Esses tipos de **comunicação serial** podem ocorrer em canais de comunicação dos tipos:
- **Simplex:** só permite a transmissão de dados, síncrona ou assíncrona, em uma direção.
- **Half-duplex:** permite a comunicação em ambas direções, mas não simultaneamente.
- **Full-duplex:** permite a comunicação em ambas direções ao mesmo tempo.





# Referências

---

- Andrew S. Tanenbaum, Organização Estruturada de Computadores, 5ª edição, Prentice-Hall do Brasil, 2007.
- Roberta L. Gomes, Nível da Lógica Digital – barramentos, Universidade Federal do Espírito Santo, 2007.

