



# ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

ENGENHARIA DA COMPUTAÇÃO – UFC/SOBRAL

Prof. Wendley S. Silva



# CHIPS DE CPU E BARRAMENTOS

PINAGEM (PINOUT)



# CHIPS PROCESSADORES



UNIVERSIDADE  
FEDERAL DO CEARÁ

- 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.



## ■ 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.



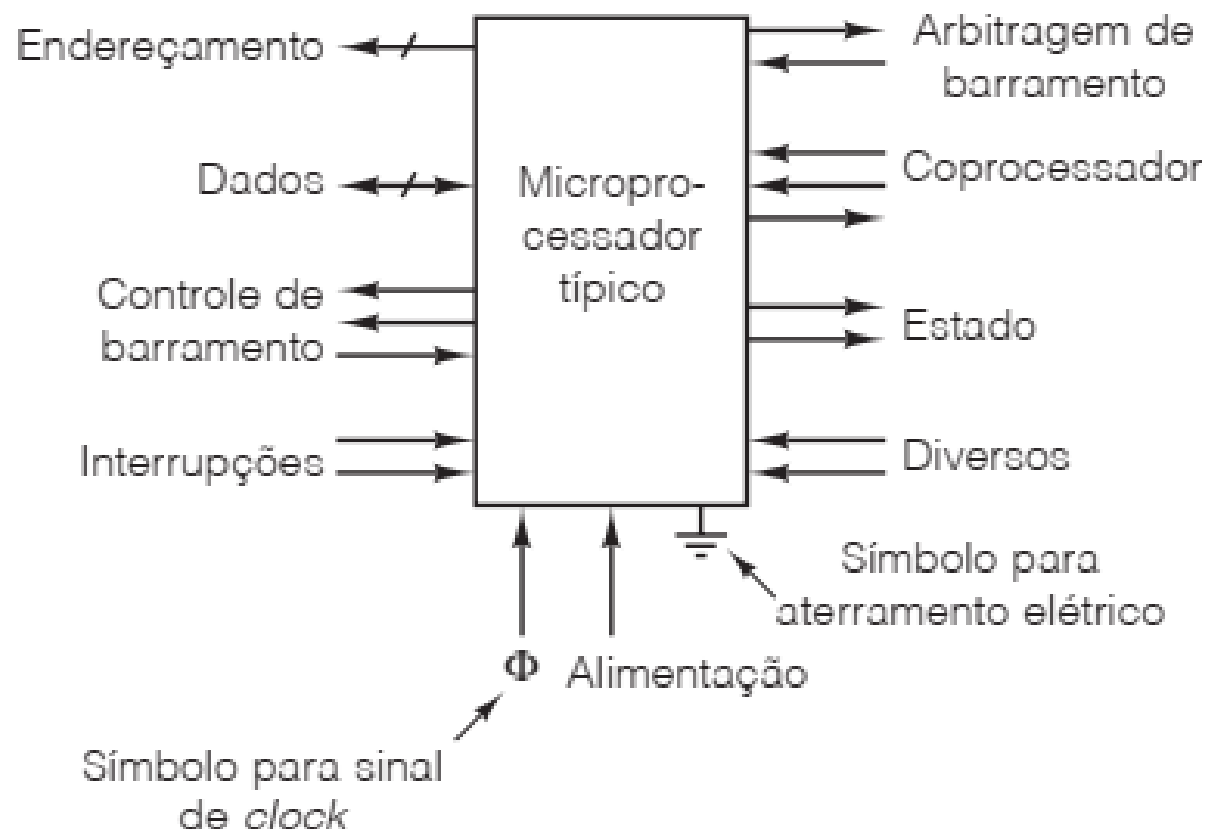
- Além dos pinos de dados e de endereços, existem os pinos de controle.
- Controle: regulam o fluxo e a temporização dos dados que entram e saem do processador, e outros usos.
- Os pinos de controle podem ser divididos (de maneira geral):
  - Controle do Barramento
  - Interrupções
  - Arbitragem do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

# CHIPS PROCESSADORES



UNIVERSIDADE  
FEDERAL DO CEARÁ

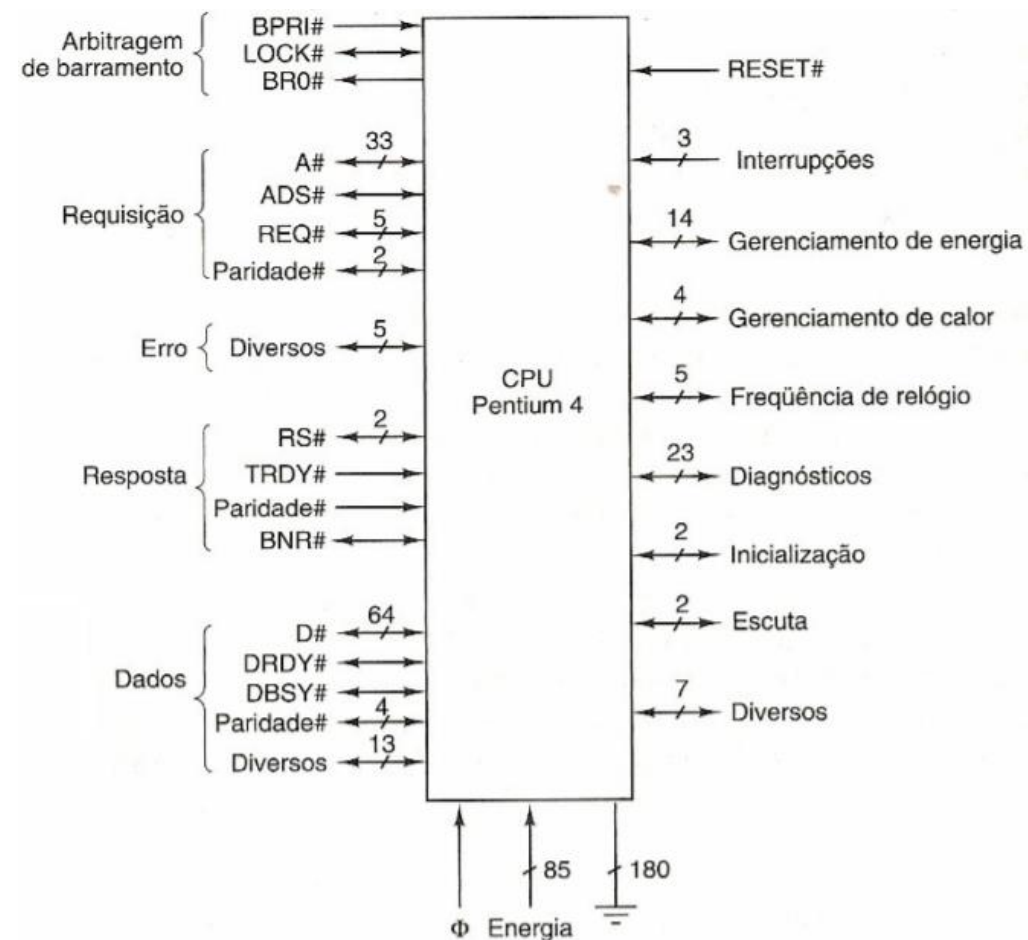
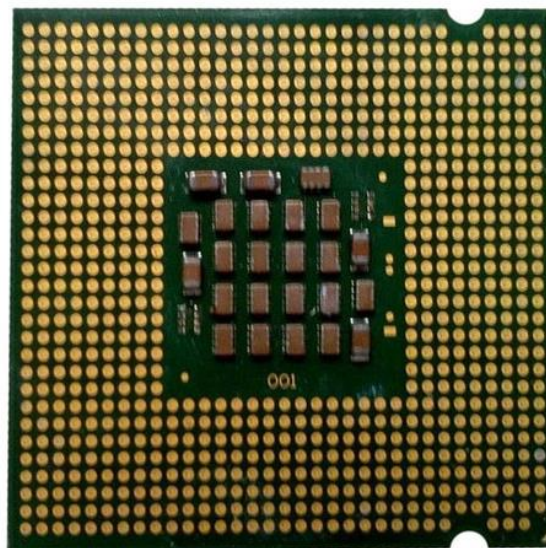
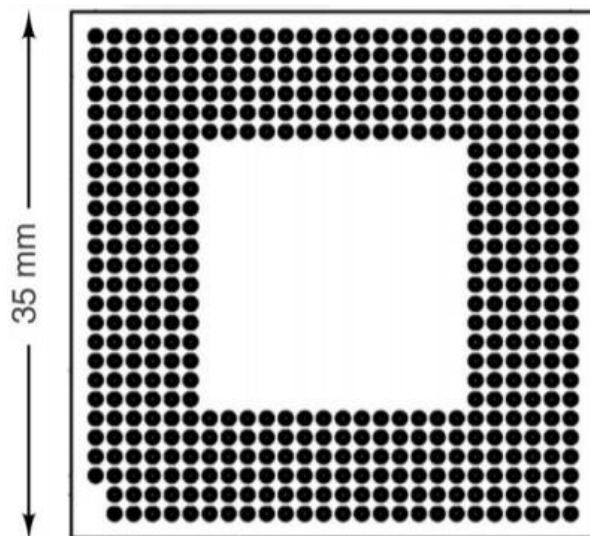
Pinagem lógica de uma CPU genérica. As setas indicam sinais de entrada e sinais de saída. Os segmentos de reta diagonais indicam que são utilizados vários pinos. Há um número que indica quantos são os pinos para uma CPU específica.



# CHIPS PROCESSADORES



UNIVERSIDADE  
FEDERAL DO CEARÁ

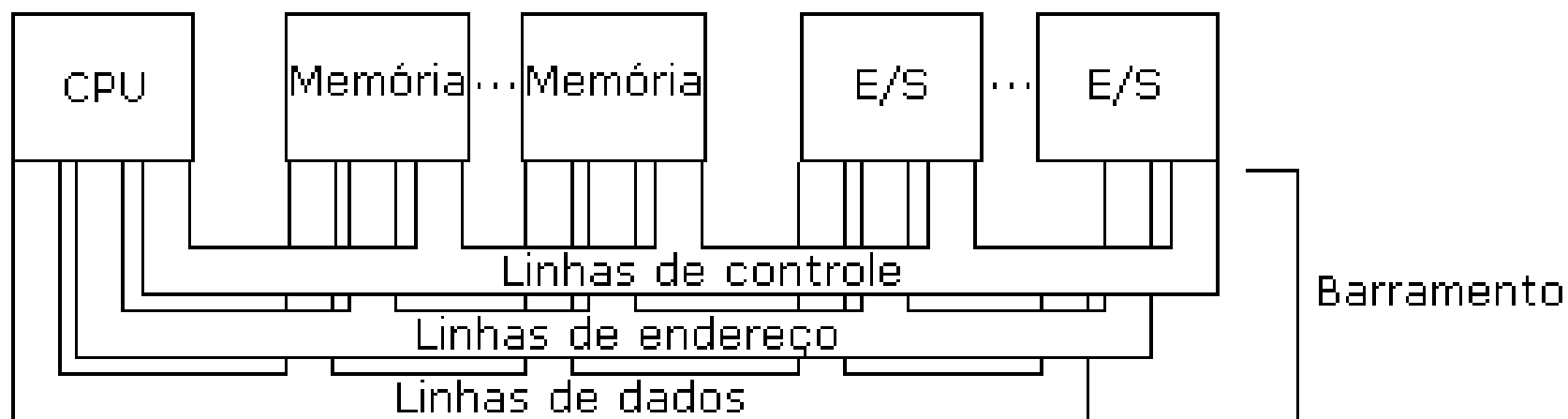


# BARRAMENTOS



UNIVERSIDADE  
FEDERAL DO CEARÁ

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



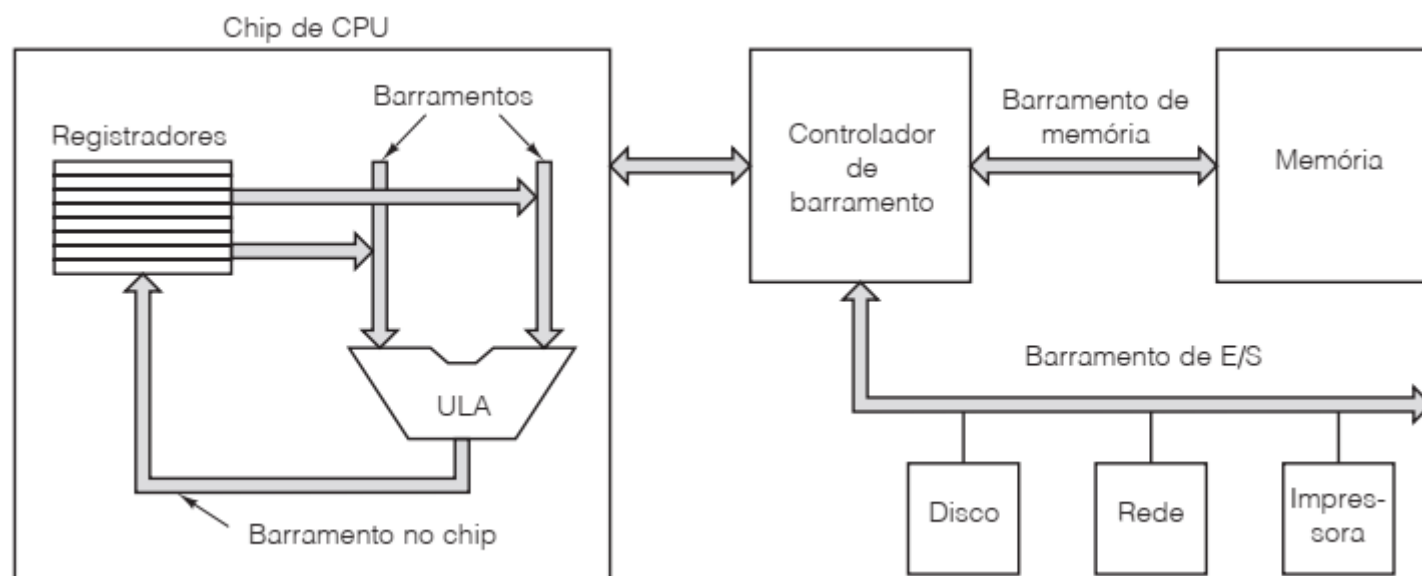


# BARRAMENTOS



UNIVERSIDADE  
FEDERAL DO CEARÁ

- 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



UNIVERSIDADE  
FEDERAL DO CEARÁ

- Estrutura de barramento
- Simples
  - Omnibus (PDP-8)
  - Unibus (PDP-11)
- Múltipla
  - Barramento para Controle
  - Barramento para Endereço
  - Barramento para Dados

# PROBLEMAS COM BARRAMENTO SIMPLES



UNIVERSIDADE  
FEDERAL DO CEARÁ

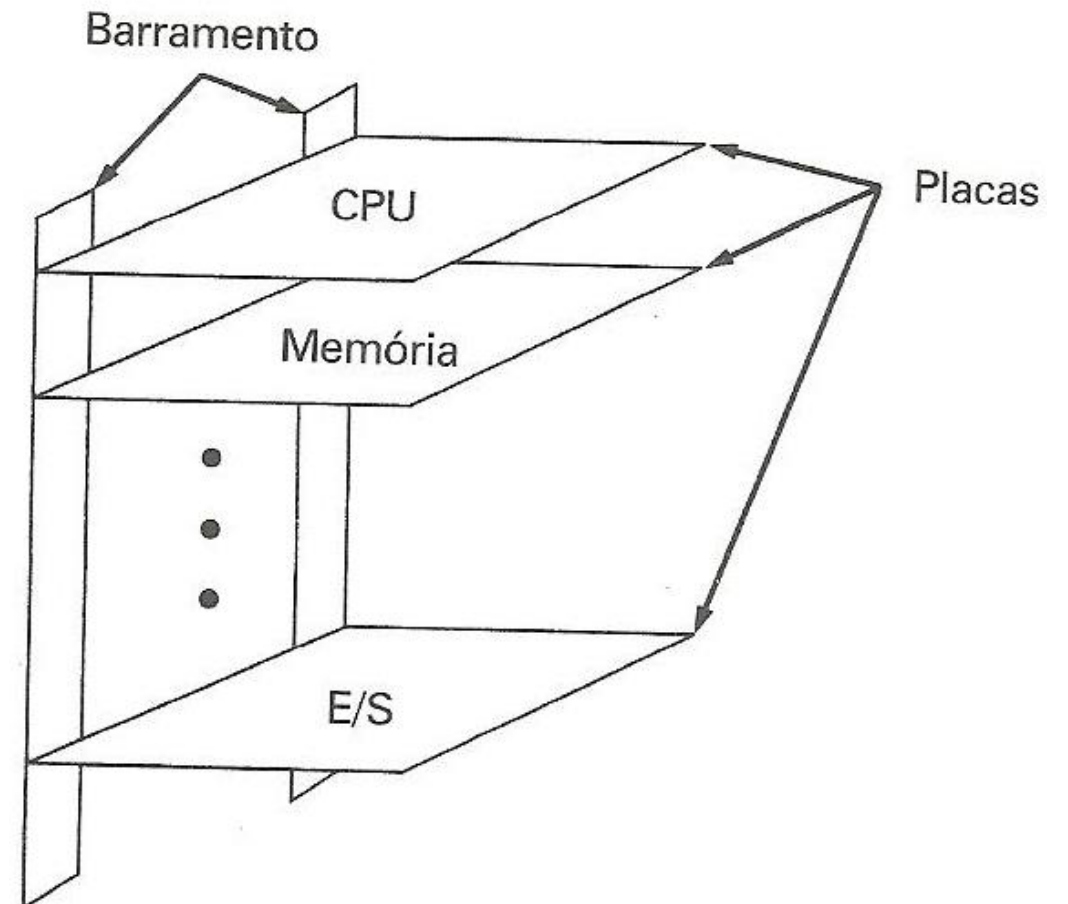
- Muitos dispositivos em um barramento ocasiona:
  - Delays de transmissões
  - Transferência de grande volume de dados afeta a performance
- Maioria dos sistemas usa barramento múltiplo para resolver esse problema

# ESTRUTURA DE BARRAMENTO



UNIVERSIDADE  
FEDERAL DO CEARÁ

- Estrutura típica de uma arquitetura de barramentos



# BARRAMENTO DE DADOS



UNIVERSIDADE  
FEDERAL DO CEARÁ

- 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

- 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** (*transceivers*)

# BARRAMENTOS

- Exemplos de mestres e escravos de barramentos

| Mestre             | Escravo            | Exemplo                                       |
|--------------------|--------------------|---|
| CPU                | Memória            | Buscar instruções e dados                     |
| CPU                | Dispositivo de E/S | Iniciar transferência de dados                |
| CPU                | Coprocessador      | CPU que passa instruções para o coprocessador |
| Dispositivo de E/S | Memória            | DMA (acesso direto à memória)                 |
| Coprocessador      | CPU                | Coprocessador que busca operandos na CPU      |

# BARRAMENTOS

- Cada barramento tem uma banda passante (BVV) expressa em capacidade de transações (bytes/seg)
- Principais aspectos de projetos de barramento que afetam a BVV 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

- 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

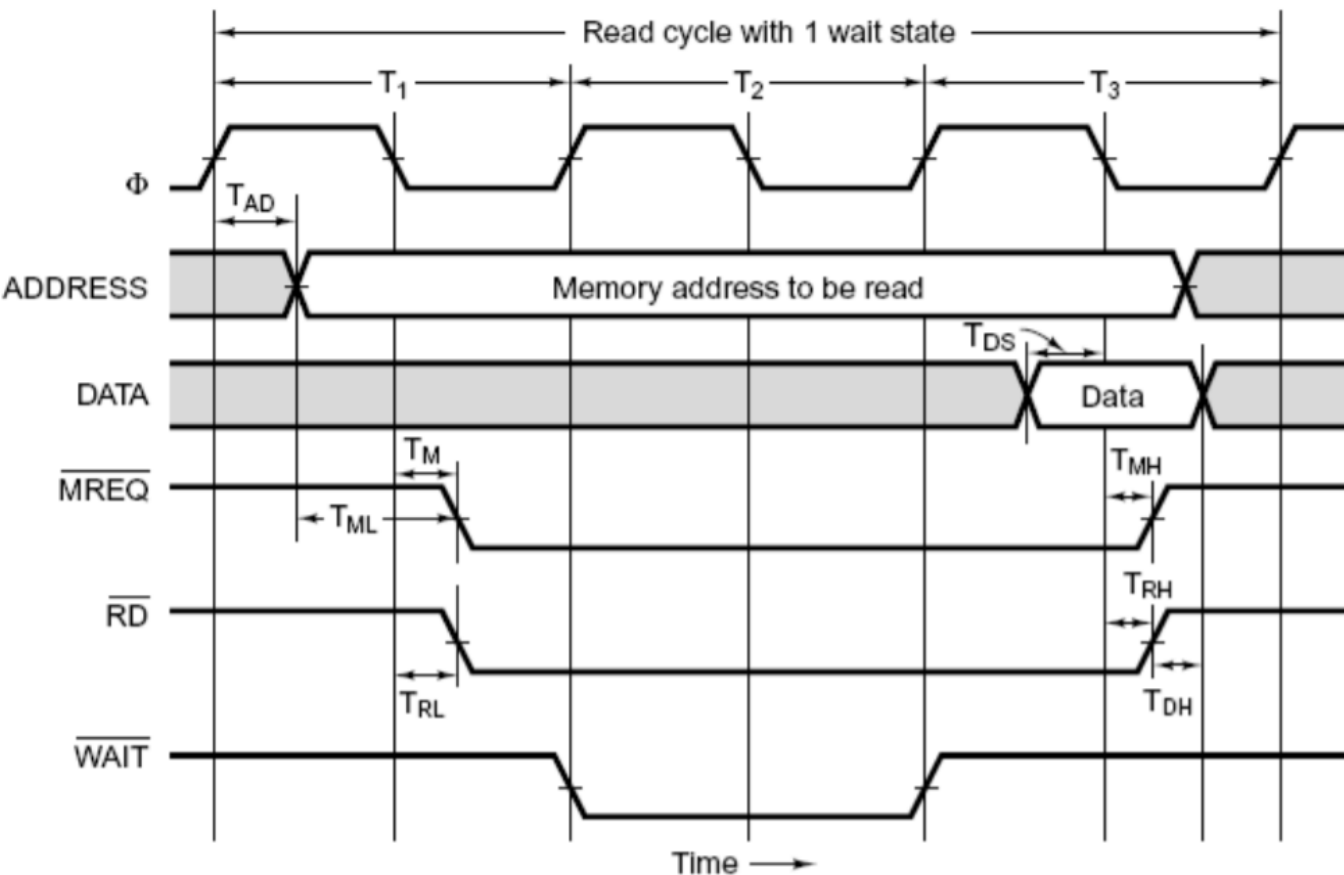
- **Barramentos Multiplexados:** Utilizados ora para transferência de endereços, ora para transferência de dados.
  - Visa controlar o problema de barramentos muito longos
  - O sistema é mais lento!
- Temporização do Barramento
- Barramento Síncrono:
  - Uma de suas linhas é alimentada por um clock 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



| 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  $T_3$

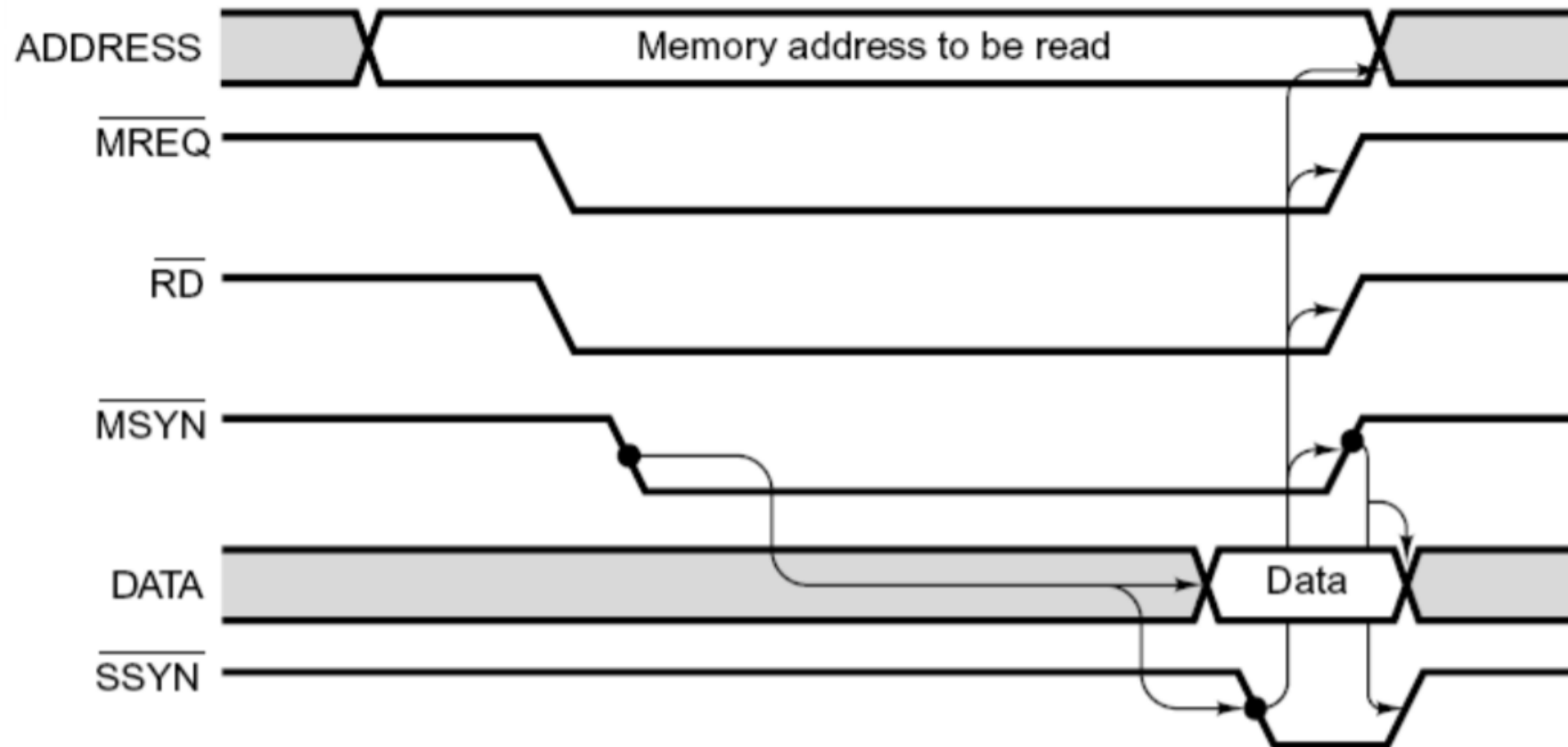
| Símbolo  | Parâmetro   | Mín. | Máx. | Unidade |
|----------|---|------|------|---------|
| $T_{AD}$ | Atraso de saída do endereço   |      | 4    | nsec    |
| $T_{ML}$ | Endereço estável antes de $\overline{MREQ}$                           | 2    |      | nsec    |
| $T_M$    | Atraso de $\overline{MREQ}$ 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  $\overline{MREQ}$  e  $\overline{RD}$ , um sinal especial  $\overline{MSYN}$  (*Master SYNchronization*) deve ser ativado.
  - Quando o escravo enxerga o  $\overline{MSYN}$  ativo, ele faz seu trabalho o mais rápido possível. Ao terminar, o sinal  $\overline{SSYN}$  (*Slave SYNchronization*) é ativado.
  - O sinal  $\overline{SSYN}$  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.  $\overline{\text{MSYN}}$  é ativado
  2.  $\overline{\text{SSYN}}$  é ativado em resposta a  $\overline{\text{MSYN}}$
  3.  $\overline{\text{MSYN}}$  é negado em resposta a  $\overline{\text{SSYN}}$
  4.  $\overline{\text{SSYN}}$  é negado em resposta à negação de  $\overline{\text{MSYN}}$

# 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 ou mais lentos que o outro par.
    - Há 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

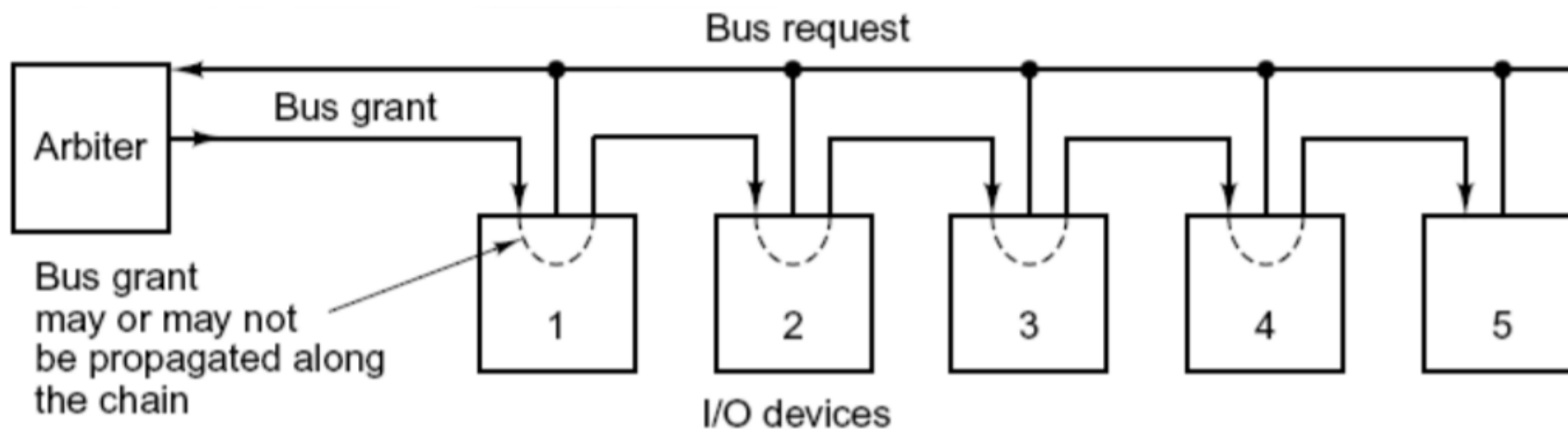
- O que acontece se dois ou mais dispositivos desejarem tornar-se mestres (tomar o controle do barramento) ao mesmo tempo?
- Mecanismos de **Arbitragem do Barramento**
  - Resolver conflitos e evitar “o caos”
  - Em geral, E/S tem preferência
  - Os discos e demais dispositivos cuja operação dependa 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.

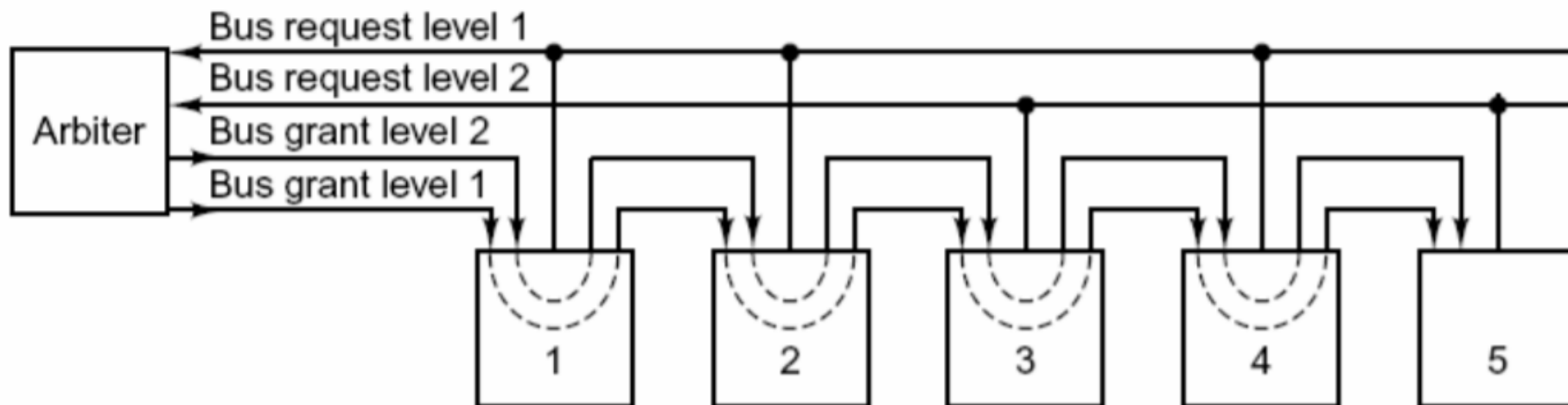
# ARBITRAGEM DO BARRAMENTO

- A arbitragem é então definida pelo esquema conhecido como Daisy Chaining (*conexão em cadeia*)
- **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

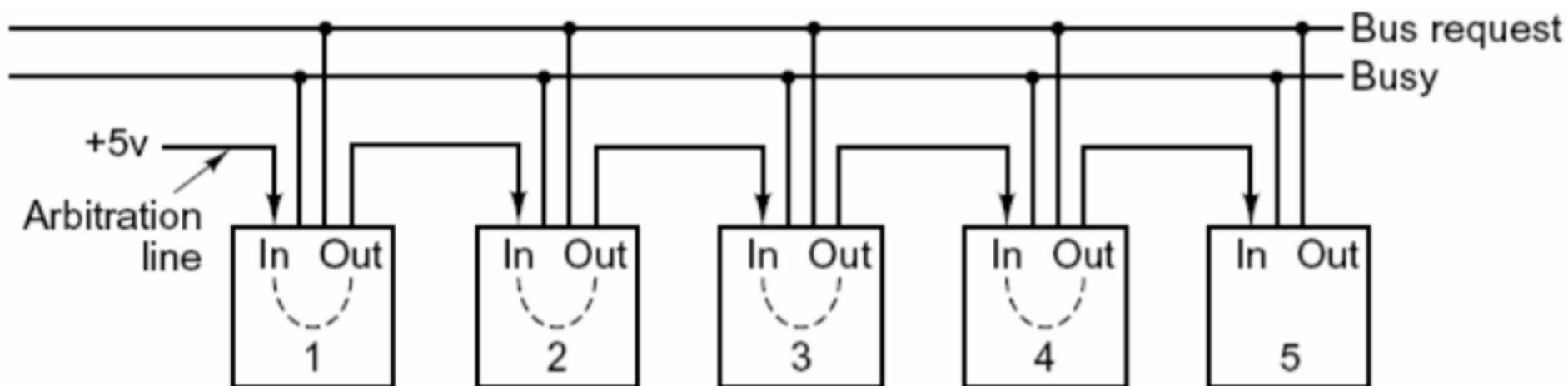
- **Abordagem Descentralizada**

- Mecanismo I:
- 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



UNIVERSIDADE  
FEDERAL DO CEARÁ

- TANENBAUM, A. S. Organização Estruturada de Computadores. Editora LTC, 5 ed, Rio de Janeiro, 2007.
- STALLINGS, W. Arquitetura e Organização de Computadores. Editora Prentice Hall, 5 edição, 2002.