Barramento

Bus é um conjunto de condutores elétricos em um computador que permite a comunicação entre vários componentes do computador, tais como; CPU, memória, dispositivos de I/O. Sinais que trafegam:

- Dados;
- Relógio;
- Endereços;
- Sinais de controle.

Bus Standard (protocolo) é um conjunto de regras que governam **como** as comunicações no barramento serão efetuadas.

Vantagens

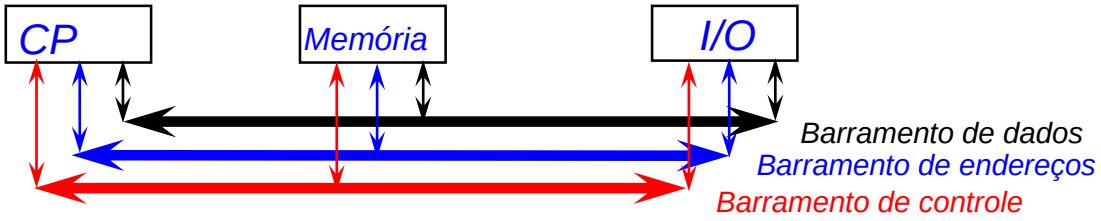
- Baixo custo na comunicação entre componentes, desde que um simples conjunto de fios é compartilhado em múltiplo sentidos;
- Versatilidade, que permite a fácil adição de novos dispositivos no computador.

Desvantagens

- Criação de engarrafamento (**bottleneck**) na comunicação, limitando a máxima vazão de dados (**throughput**) para dispositivos de I/O.

Funcionalidade

- Linhas de Dados - Barramento de Dados: fornecem o meio de transmissão de dados entre os módulos do sistema.
- Linhas de Endereço - Barramento de Endereços: usadas para designar fonte e destino dos dados do barramento de dados.
- Linhas de Controle - Barramento de Controle: usadas para controlar o acesso e o uso de linhas de dados e endereços.

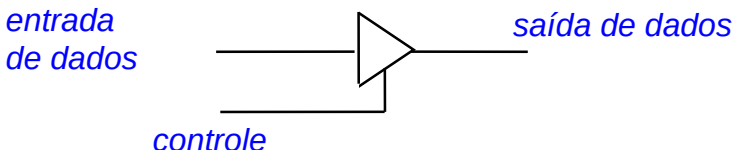


Sinais de controle típicos

- **Memory Write** - Causa a escrita de dados do barramento de dados no endereço especificado no barramento de endereços.
- **Memory Read** - Causa dados de um dado endereço especificado pelo barramento de endereço ser posto no barramento de dados.
- **I/O Write** - Causa dados no barramento de dados serem enviados para uma porta de saída (dispositivo de I/O).
- **I/O Read** - Causa a leitura de dados de um dispositivo de I/O, os quais serão colocados no barramento de dados.
- **Bus request** - Indica que um módulo pede controle do barramento do sistema.
  Reset - Inicializa todos os módulos

Características de acesso

- Todo dispositivo de memória ou I/O deve ser exclusivo no acesso ao barramento.
- A seleção é feita através de sinais especiais de controle como:
  - Memory Read
  - Memory Write
  - I/O Read
  - I/O Write
- Todo dispositivo deve escrever no barramento através de “buffers Tristate”.



Dispositivos

- **Ativos ou Mestres** - dispositivos que controlam o protocolo de acesso ao barramento para leitura ou escrita de dados;
- **Passivos ou Escravos** - dispositivos que simplesmente obedecem à requisição do mestre.

Exemplo:

- CPU ordena que o controlador de disco leia ou escreva um bloco de dados.
- A CPU é o mestre e o controlador de disco é o escravo.

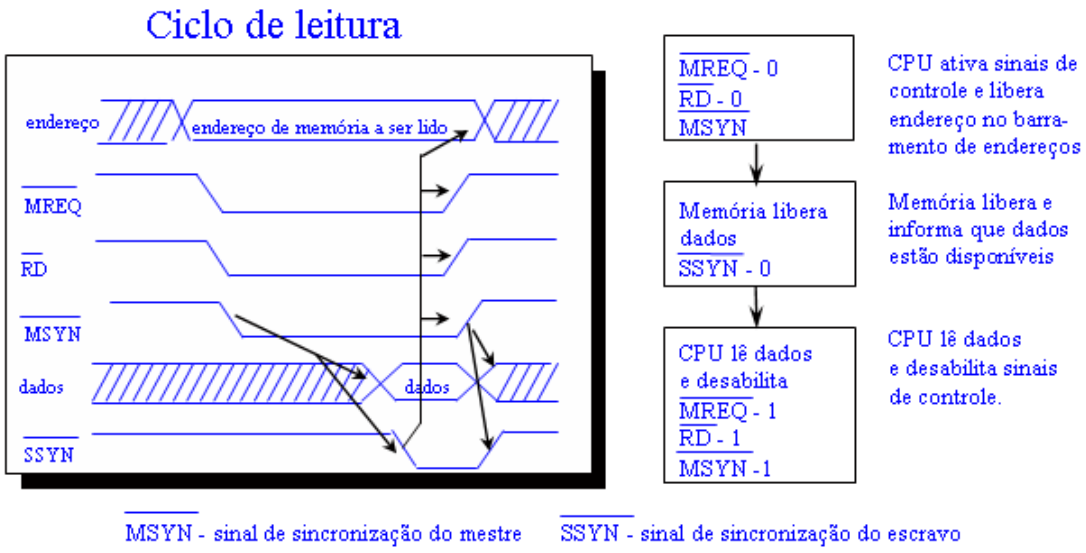
Quanto à temporização

Barramento Assíncrono

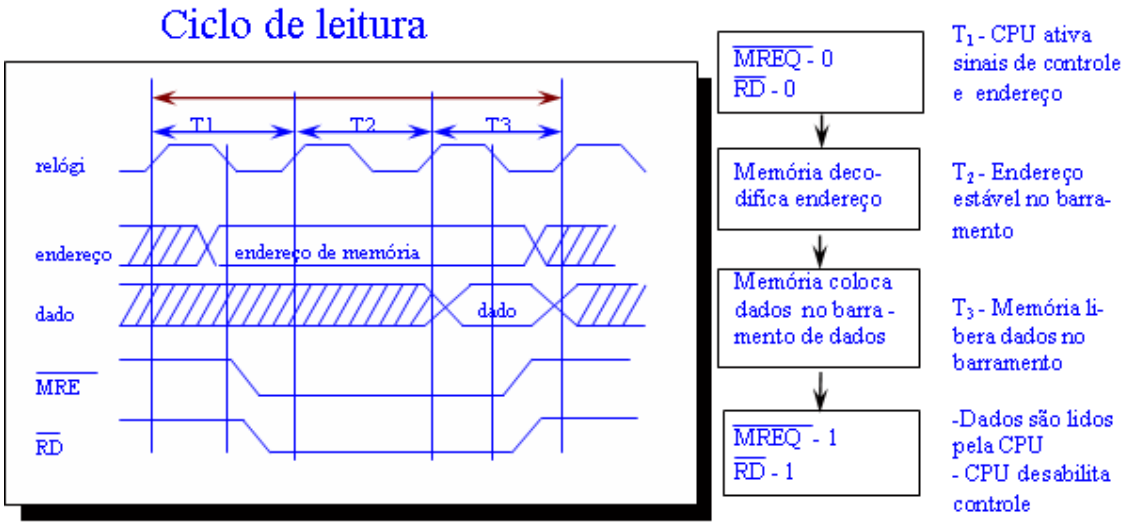
- O controle ocorre exclusivamente por meio de sinais trocados entre os dispositivos. Os ciclos de barramentos podem ter qualquer duração e não precisam ser iguais para todas as situações.
- São barramentos mais rápidos que os síncronos.

Barramento Síncrono

- Este tipo de barramento exige que todo o tráfego de dados e controle seja sincronizado sob uma mesma base de tempo, relógio ou *clock*



Barramento Assíncrono



Barramento Síncrono

Barramento síncrono:

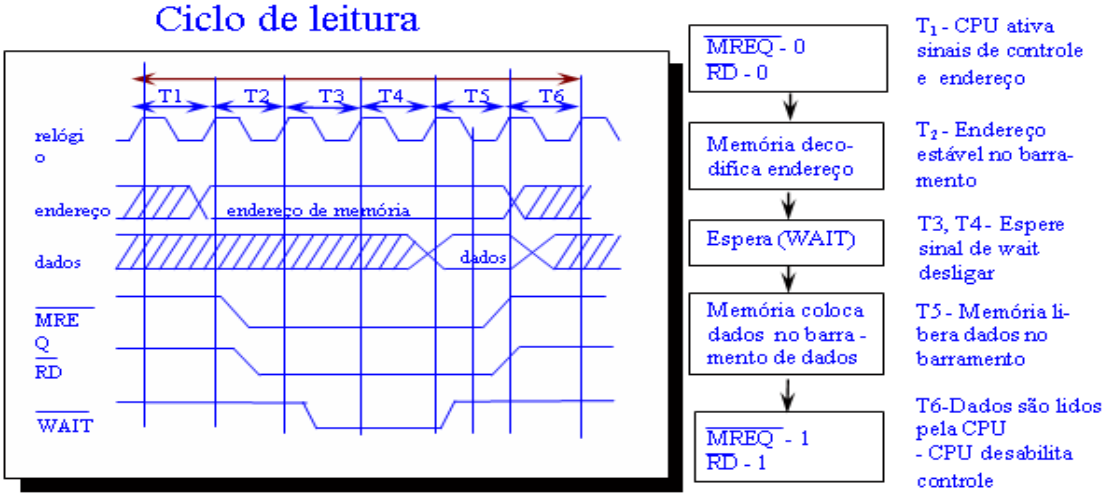
Incluindo wait-states

- Usando um sinal extra (*wait*) este barramento pode se comportar como um misto de síncrono e assíncrono.
- Sempre que o dispositivo escravo não puder responder no tempo padrão do barramento, este liga o sinal de wait para fazer com que o mestre pare o protocolo. Quando o escravo puder prosseguir, desliga o wait.

Arbitragem

Quando dois ou mais dispositivos querem se tornar mestres do barramento ao mesmo tempo?

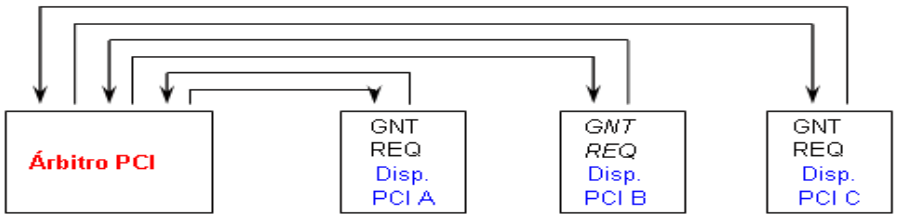
- Pode existir uma inviabilidade de operações (caos) do sistema se não houver um mecanismo adequado de arbitragem do barramento.
- A arbitragem decide qual mestre terá o controle do barramento num dado instante: **Arbitragem centralizada** ou **Arbitragem descentralizada**.



Arbitragem Centralizada

Arbitragem no barramento PCI (centralizado)

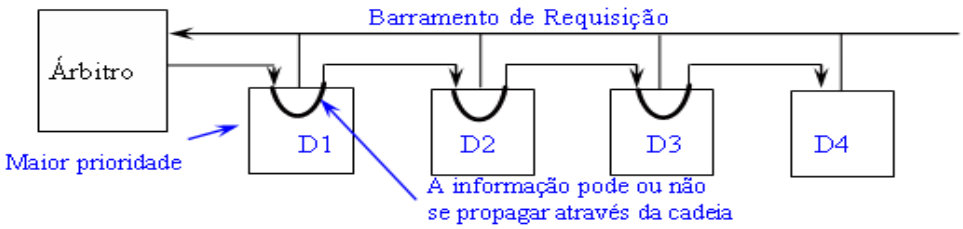
- Exemplo para três dispositivos
- O árbitro decide qual mestre controla o barramento



Arbitragem Híbrida

Este tipo de arbitragem é gerenciado por um árbitro que, juntamente com um daisy chain, estabelece a ordem de acesso ao barramento.

- **Barramento de um nível usando daisy chaining**

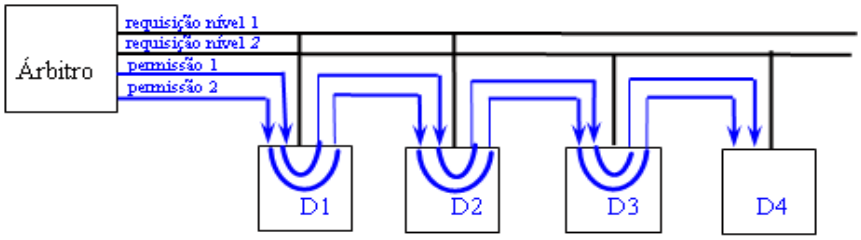


Características

1. Todos os dispositivos são ligados em série, assim a permissão, dada pelo árbitro, pode ou não se propagar através da cadeia.

2. Cada dispositivo deve solicitar acesso ao barramento.
3. O dispositivo mais próximo do árbitro tem maior prioridade.

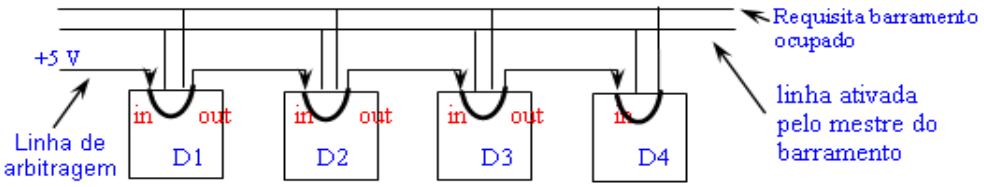
- **Arbitragem com dois níveis de prioridade**



Características

1. Cada dispositivo se liga a um dos níveis de requisição.
2. Os dispositivos com tempos mais críticos se ligam aos níveis de maior prioridade.
3. Se múltiplos níveis de prioridade são requeridos ao mesmo tempo, o árbitro solta a permissão apenas para os de prioridade mais alta.

Barramento descentralizado é o barramento Multibus - Daisy Chain sem árbitro



Características:

1. Quando nenhum dispositivo quer barramento, a linha de arbitragem ativada é propagada através de todos os dispositivos.
2. Para obter o barramento um dispositivo primeiro verifica se o barramento está disponível, e se a linha de arbitragem que está recebendo, **in**, está ativada.
3. Se **in** estiver desativada, ela não poderá tornar-se mestre do barramento.
4. Se **in** estiver ativada, o dispositivo requisita o barramento, desativa **out**, o que faz com que todos os dispositivos seguintes na cadeia desativem **in** e **out**.
5. O tempo de propagação do sinal **in** do primeiro dispositivo (D1) ao **out** do último dispositivo (D4) tem que ser menor que 1 período de clock.

Barramento Multibus

- O dispositivo mais próximo do início da cadeia que requer o barramento tem maior prioridade. Este esquema é similar ao sistema híbrido com *daisy chain*, exceto por:
  - Não existe mais a figura do árbitro
  - É mais rápido
  - Não vulnerável a falhas do árbitro
  - O barramento Multibus também oferece arbitragem centralizada, permitindo que os projetistas façam a escolha.

Quanto aos dispositivos a ele acoplados

Barramentos de Memória CPU-Memory Buses

- São pequenos
- Operam em alta velocidade
- São em geral conectados diretamente a CPU para maximizar a largura de banda entre memória e CPU (bandwidth)
- Tipos de dispositivos são conhecidos

Barramentos de Entrada e Saída CPU-I/O Buses

- Podem ser longos.
- Podem ter diferentes tipos de dispositivos conectados a ele.
- São, em geral, mais lentos que os barramentos de memória.