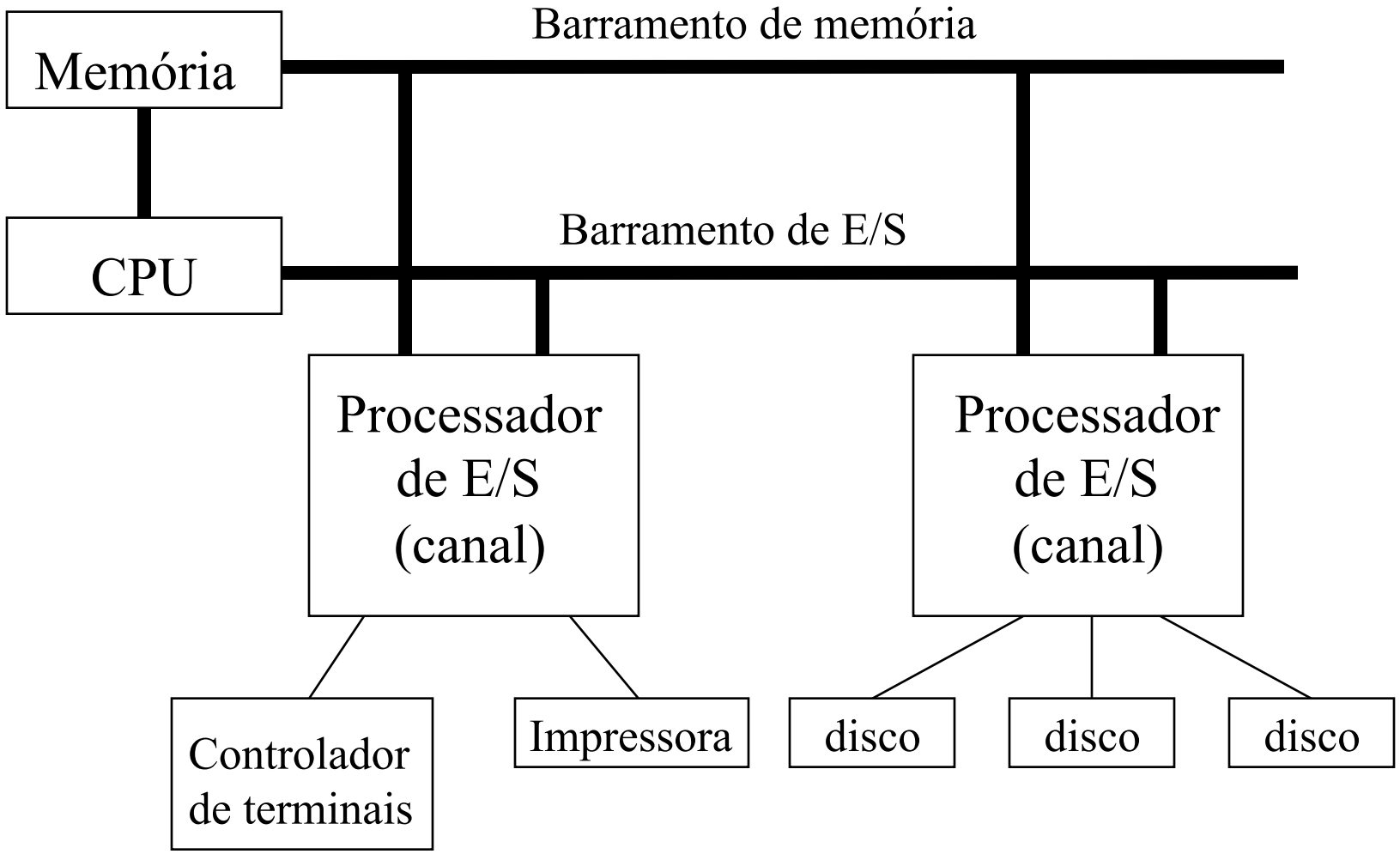
Conceitos de Entrada e Saída

O processo de fornecer informações ou obtê-las dos computadores é chamado **entrada/saída** (**E/S**).

Grandes "mainframes" utilizam uma organização de E/S consistindo de uma ou mais CPUs, memória e um ou mais controladores de E/S chamados **canais de dados**.

Quando a CPU quer realizar E/S, ela carrega um programa especial em um dos canais e diz ao canal para executá-lo.

O canal manipula toda a E/S para e da memória principal, deixando a CPU livre para fazer outras coisas. Ao terminar, ele interrompe a CPU.



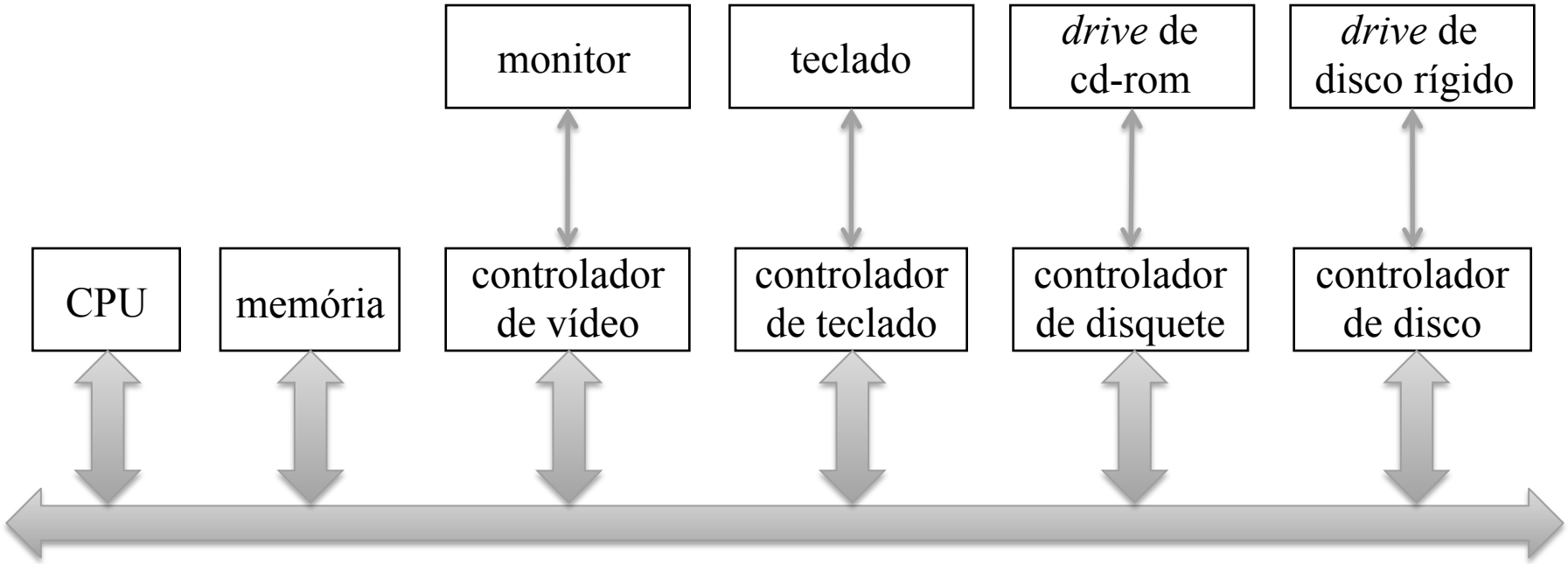
Estrutura de E/S em um "mainfraime"

Computadores pessoais usam uma estrutura mais simples, consistindo de um único barramento, usado para interligar a CPU, memória e dispositivos de E/S.

Cada dispositivo consiste de duas partes: o **controlador** (componentes eletrônicos) e o próprio dispositivo.

A função de um controlador é controlar seu dispositivo de E/S e gerenciar os acessos ao barramento para ele.

Um controlador que lê ou escreve um bloco de dados na memória sem a intervenção da CPU é dito fazer acesso direto à memória – **DMA**.


monitor
teclado
drive de cd-rom
drive de disco rígido
CPU
memória
controlador de vídeo
controlador de teclado
controlador de disquete
controlador de disco

Um árbitro do barramento é utilizado quando a CPU e o controlador querem usar o barramento ao mesmo tempo, sendo, geralmente, dada preferência ao controlador.

Roubo de ciclos ocorre quando um dispositivo requer o barramento e a CPU já o está usando. Isto reduz a velocidade do computador.

Um barramento é um caminho comum elétrico entre múltiplos dispositivos.

O protocolo do barramento define as regras de funcionamento do barramento, permitindo que componentes projetados por terceiros possam ser conectados ao sistema.

Dispositivos que podem iniciar transferências pelo barramento são denominados mestres. Dispositivos que aguardam uma requisição são chamados escravos.

mestre	escravo	
CPU	memória	Busca de instruções e dados
CPU	E/S	Início de transferência de dados
CPU	Co-processador	Instruções de ponto-flutuante
E/S	memória	DMA
Co-processador	memória	Buscando operandos

A maioria dos mestres de barramento estão conectados a ele através de pastilhas chamadas acionadores de barramentos (*bus driver*), que são essencialmente amplificadores digitais.

A maioria dos escravos estão conectados ao barramento através de receptores de barramento (*bus receiver*).

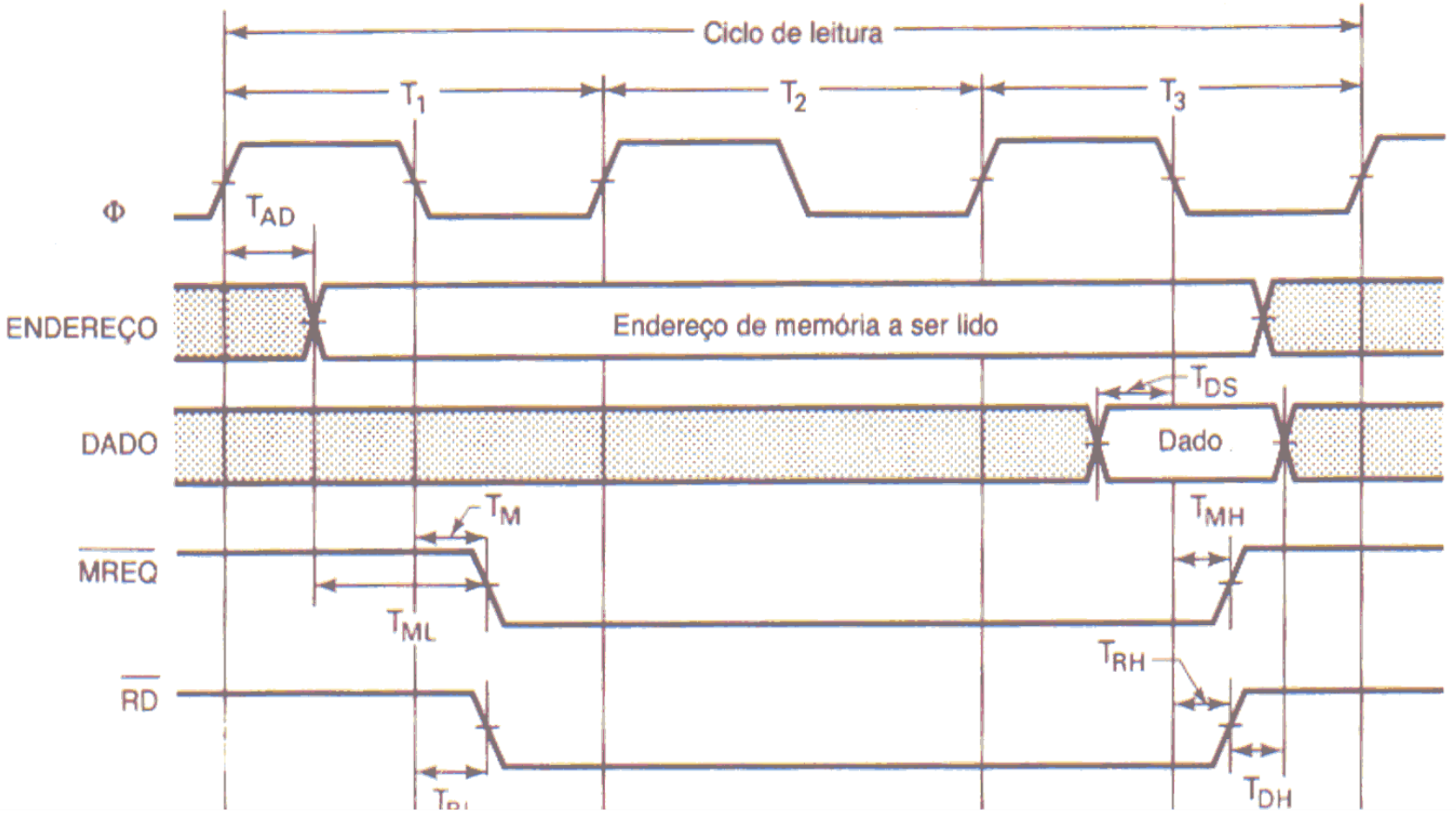
Para os dispositivos que podem agir tanto como mestre quanto como escravo, é utilizada uma pastilha combinada chamada transceptor de barramento (*bus transceiver*).

Estas pastilhas de interface com o barramento são frequentemente dispositivos tri-state, para permitir que eles flutuem quando não são necessários, ou dispositivos de coletor aberto.

Um barramento síncrono possui uma linha acionada por um oscilador a cristal. Todas as atividades do barramento gastam um número inteiro de ciclos do oscilador, chamados **ciclos de barramento**.

Um barramento assíncrono não possui um relógio mestre. Os ciclos podem ter qualquer duração requerida e não precisam ser os mesmos entre todos os pares de dispositivos.

Considerando um relógio de 4MHz, um ciclo de barramento leva 250ns. Assumindo que a leitura de um byte da memória gasta 3 ciclos de barramento, o ciclo de leitura leva 750ns.


Ciclo de leitura
T1
T2
T3
Φ
TAD
ENDEREÇO
Endereço de memória a ser lido
TDS
DADO
Dado
TM
TMH
MREQ
TML
TRH
RD
TDH

Quando uma leitura de bloco é iniciada, o mestre do barramento diz ao escravo quantos bytes serão transferidos. O escravo solta um byte durante cada ciclo até que a quantidade seja exaurida. A leitura de um bloco de n bytes gastaria $(n + 2)$ ciclos, em vez de $3n$.

No barramento assíncrono, em vez de amarrar tudo ao relógio, um conjunto de sinais, denominado *handshake* completo, determina o início e o fim de um ciclo de leitura/escrita.

Sejam os sinais $\overline{\text{MSYN}}$ (*Master SYNchronisation*) e $\overline{\text{SSYN}}$ (*Slave SYNchronisation*). O *handshake* consiste em quatro eventos:

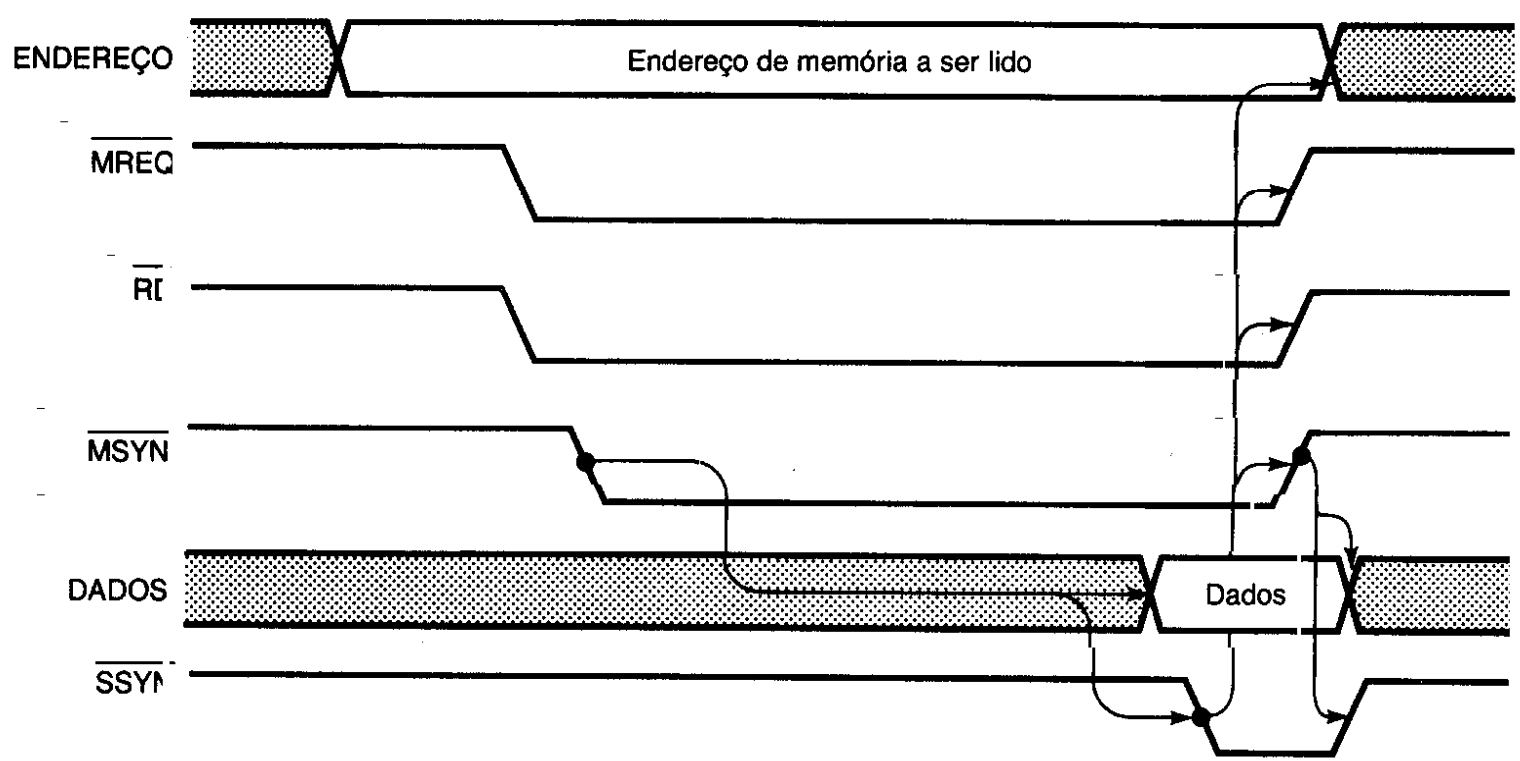
1 - $\overline{\text{MSYN}}$ é ativado;

2 - $\overline{\text{SSYN}}$ é ativado em resposta a $\overline{\text{MSYN}}$;

3 - $\overline{\text{MSYN}}$ é desativado em resposta a $\overline{\text{SSYN}}$;

4 - $\overline{\text{SSYN}}$ é desativado em resposta à desativação de $\overline{\text{MSYN}}$.

Handshakes completos são independentes do tempo. Cada evento é causado por um evento anterior e não por um pulso de relógio.


ENDEREÇO
Endereço de memória a ser lido
MREQ
RD
MSYN
DADOS
Dados
SSYN

No barramento síncrono tudo trabalha em múltiplos inteiros dos ciclos do relógio e para sincronizar as diferentes velocidades dos dispositivos envolvidos, estados de espera (*wait states*) são introduzidos.

No barramento assíncrono a duração de um ciclo de leitura/escrita depende da velocidade dos dispositivos envolvidos.

Se dois ou mais dispositivos querem se tornar mestres ao mesmo tempo, surge a necessidade de algum mecanismo de arbitragem de barramento.

Na arbitragem centralizada, um único árbitro de barramento determina qual o próximo mestre. O barramento possui uma linha única de requisição que pode ser ativada por um ou mais dispositivos ao mesmo tempo.

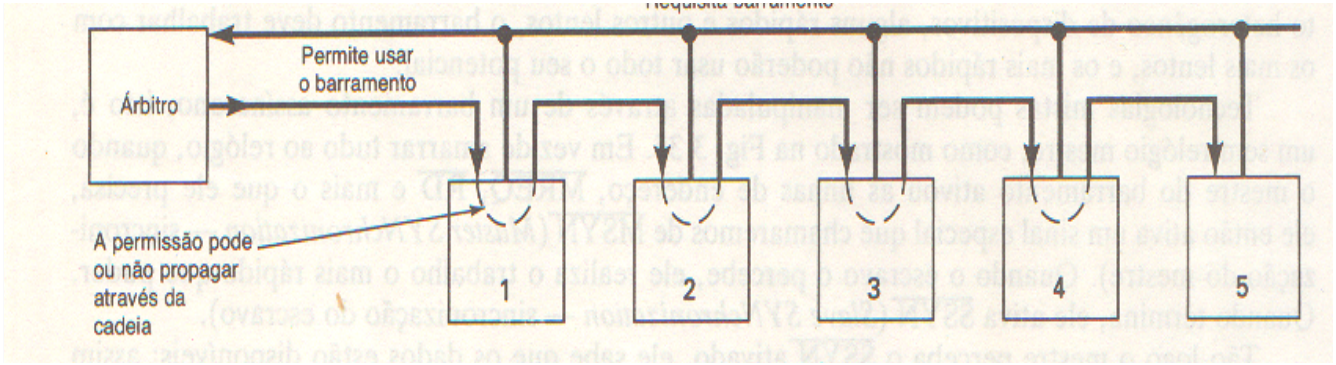
Quando o árbitro percebe uma requisição do barramento (*bus request*), ele gera uma permissão ativando uma linha de permissão de barramento (*bus grant*). Esta linha é ligada em série através de todos os dispositivos.

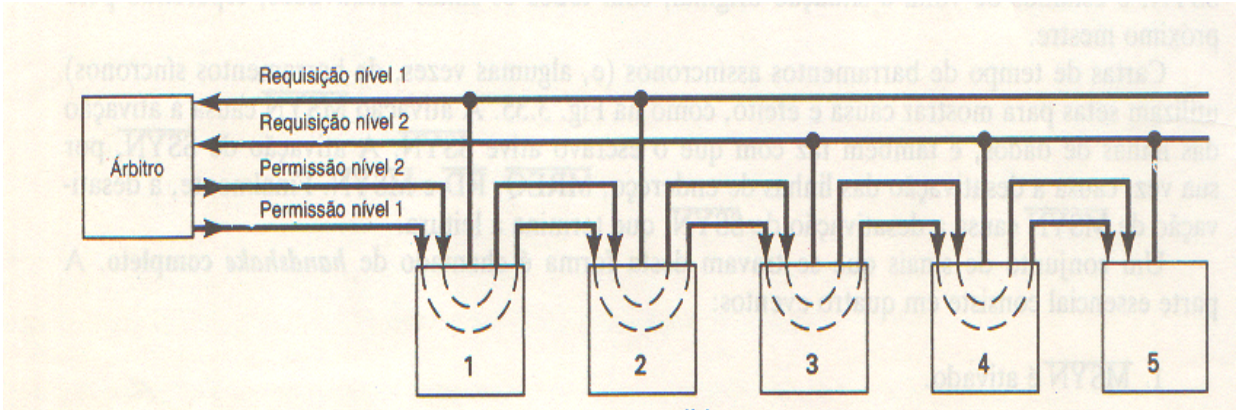
Quando o dispositivo fisicamente mais próximo do árbitro vê a permissão, verifica se foi ele que fez a requisição. Se foi, ele toma conta do barramento e não propaga a permissão pela linha. Se não foi, ele propaga a permissão para o próximo dispositivo na linha.

Este esquema é chamado *daisy chaining*. A prioridade depende da distância do dispositivo ao árbitro.

Para contornar a prioridade implícita baseada na distância do árbitro, podem-se utiliza múltiplos níveis de prioridade. Para cada nível existe uma linha de requisição e uma linha de permissão de barramento.

Alguns árbitros possuem uma terceira linha que um dispositivo ativa quando ele aceitou a permissão e tomou posse do barramento (*bus acknowledge*). Assim que for ativada, as linhas de requisição e permissão podem ser desativadas. Como resultado, outros dispositivos podem requisitar o barramento, enquanto um outro o está utilizando. Quando a transferência atual terminar, o próximo mestre já terá sido selecionado.


Árbitro
Permite usar
o barramento
A permissão pode
ou não propagar
através da
cadeia
1
2
3
4
5

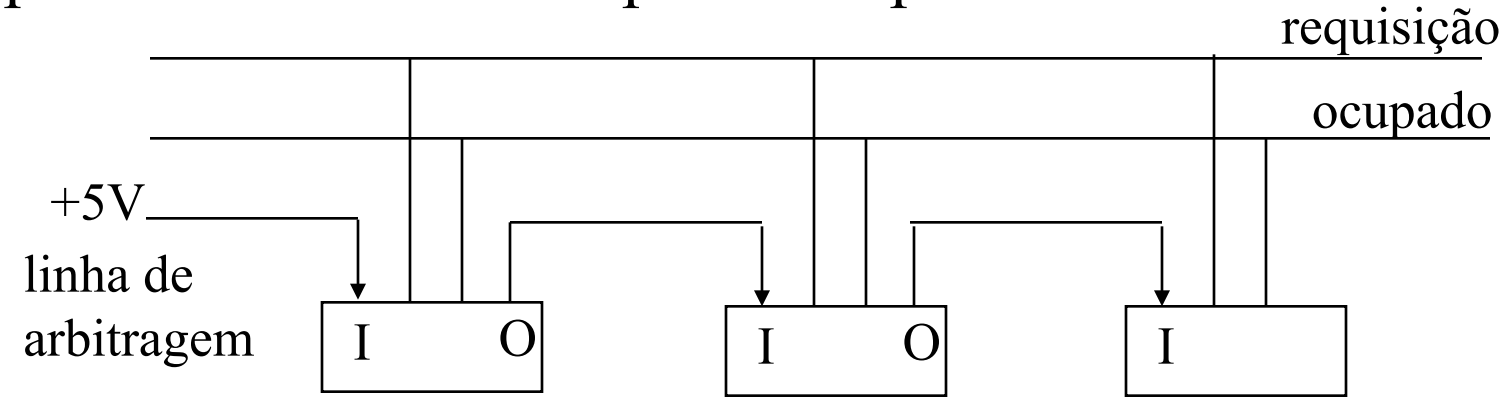

Requisição nível 1
Requisição nível 2
Permissão nível 2
Permissão nível 1
Árbitro
1
2
3
4
5

Quando a arbitragem descentralizada está sendo utilizada, não há árbitro de barramento. Um exemplo consiste no barramento apresentar as linhas de requisição priorizadas, às quais cada dispositivo se conectará (limitação no número de dispositivos).

Quando um dispositivo quer usar o barramento, ele ativa a sua linha de requisição. Todos os dispositivos monitoram todas as linhas de requisição. Assim, ao fim de cada ciclo do barramento, cada dispositivo sabe se ele é o requisitante de maior prioridade e, então, se lhe é permitido usar o barramento no próximo ciclo.

Comparado à arbitragem centralizada, este método requer mais linhas de barramento, evitando, porém, o custo potencial do árbitro.

Outro exemplo consiste no barramento apresentar somente 3 linhas, independentes do nº de dispositivos presentes.



A linha de requisição é ativada pelo dispositivo. A linha de ocupado é ativada pelo mestre corrente do barramento. A linha de arbitragem é usada para arbitrar o barramento. Esta linha é encadeada através de todos os dispositivos. O início da cadeia é mantido ativo, ligando-se à fonte de +5V.

Quando nenhum dispositivo quer o barramento, a linha de arbitragem ativada é propagada através de todos os dispositivos.

Para obter o barramento, um dispositivo primeiro verifica se o barramento está disponível e se a linha de arbitragem que está recebendo (I) está ativada.

Se I estiver desativada, ele não poderá tornar-se mestre do barramento. Se I estiver ativada, entretanto, o dispositivo desativa O, o que faz todos os seguintes na cadeia desativarem I e O.

Ao final, apenas um dispositivo terá I ativado e O desativado. Este se torna mestre do barramento, ativa ocupado e O, e inicia sua transferência.

Este esquema é similar à arbitragem *daisy chain* original, exceto pela ausência do árbitro. Assim sendo, é mais barato, mais rápido e não-vulnerável a falhas no árbitro.