

Arbitragem de Barramento

Mecanismo de Arbitragem

- Se dois dispositivos quiserem se tornar mestres da transferência de dados ao mesmo tempo, precisamos de um mecanismo para escolher um deles.
- Há dois tipos de mecanismo:
 - Centralizado
 - Descentralizado

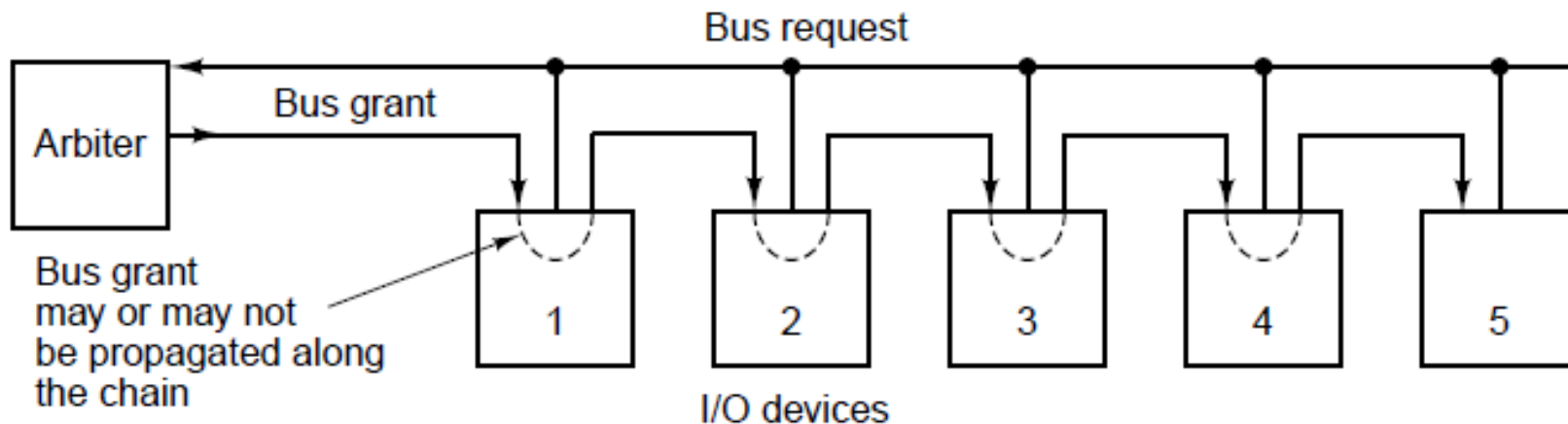
Arbitragem Centralizada

- Um único árbitro determina quem entra em seguida.
- O barramento contém uma única linha de requisição “or cabeada”.
- Esta linha pode ser acionada por vários dispositivos ao mesmo tempo.
- Quando o árbitro vê uma requisição, ativa a linha que faz a concessão do barramento. Essa linha está ligada a todos os dispositivos de saída em série.
- Quando o dispositivo que está fisicamente mais próximo do árbitro vê a concessão, verifica internamente para ver se ele fez a requisição.

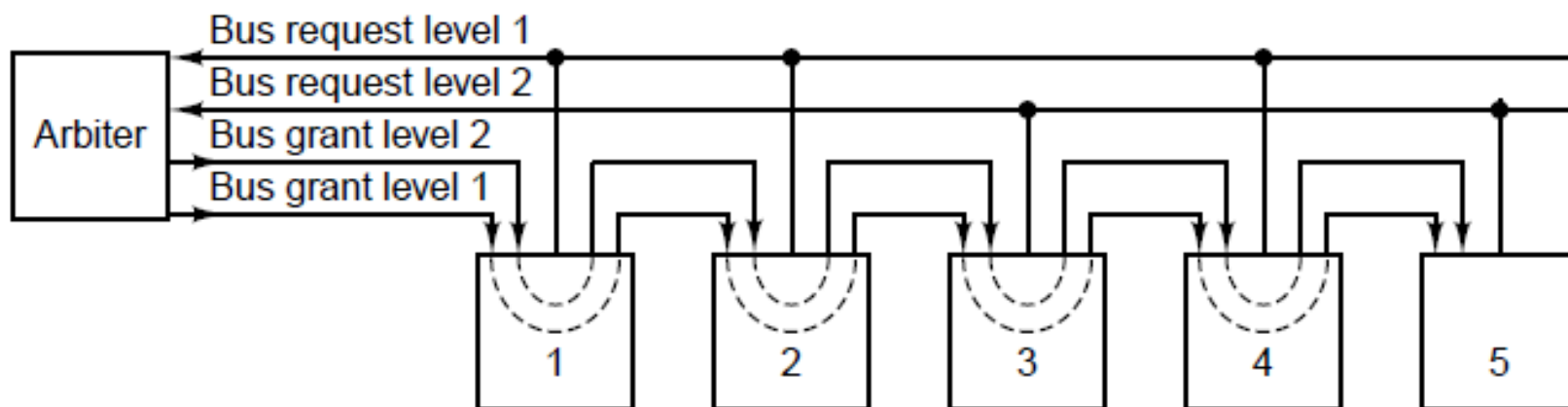
Arbitragem Centralizada

- Caso positivo, ele toma o barramento e não propaga a concessão.
- Caso negativo, ele propaga a concessão para o próximo dispositivo que se comporta da mesma maneira.
- Esse esquema é denominado encadeamento em série (*daisy channing*).

Arbitragem de Barramento Centralizada



(a)



(b)

Arbitragem Centralizada

- No esquema com prioridades, se vários níveis de prioridade são requisitados ao mesmo tempo, o árbitro emite uma concessão somente para o de prioridade mais alta.
- Entre dispositivos com a mesma prioridade é utilizado o encadeamento em série.

Arbitragem Centralizada

- Em sistemas em que a memória está no barramento principal, a CPU deve competir com todos os dispositivos de entrada e saída em praticamente todos os ciclos.
- Uma solução para esta situação é dar a CPU a prioridade mais baixa. Ela só ganha o barramento quando ninguém mais o quer.

Arbitragem Descentralizada

- Usa apenas 3 linhas: Or cabeada, BUSY, linha de arbitragem.
- Quando nenhum dispositivo quiser o barramento, a linha de arbitragem ativada é propagada por todos os dispositivos.
- Para adquirir o barramento, um dispositivo primeiro verifica para ver se o barramento está ocioso ($BUSY=0$) e se o sinal que está recebendo, IN, está ativado ($IN=1$). Se acontecer esta situação, o dispositivo toma o barramento.

Arbitragem Descentralizada

- Se IN estiver negado ($IN=0$), o dispositivo em questão não pode se tornar o mestre do barramento e o sinal OUT é negado ($OUT=0$).
- Quando o dispositivo se torna mestre do barramento, ele nega OUT ($OUT=0$), o que faz com que seu vizinho na cadeia veja IN negado ($IN=0$) e por sua vez negue OUT. Desta forma todos os dispositivos seguintes ficam impedidos de tomar o barramento.
- O dispositivo que ganhou o barramento ativa BUSY e OUT ($BUSY=1$ e $OUT=1$) e inicia a transferência.

Arbitragem de Barramento Descentralizada

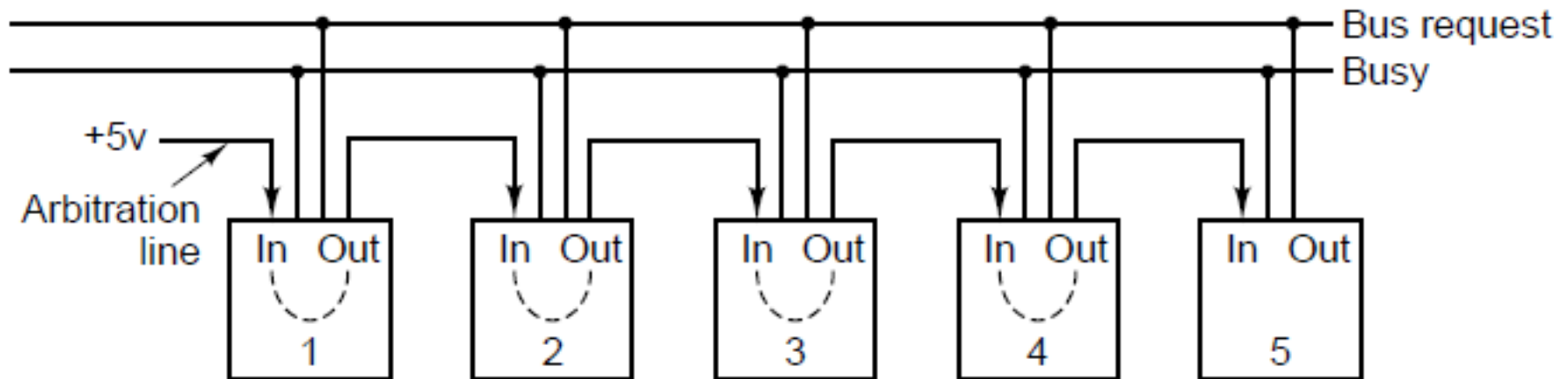


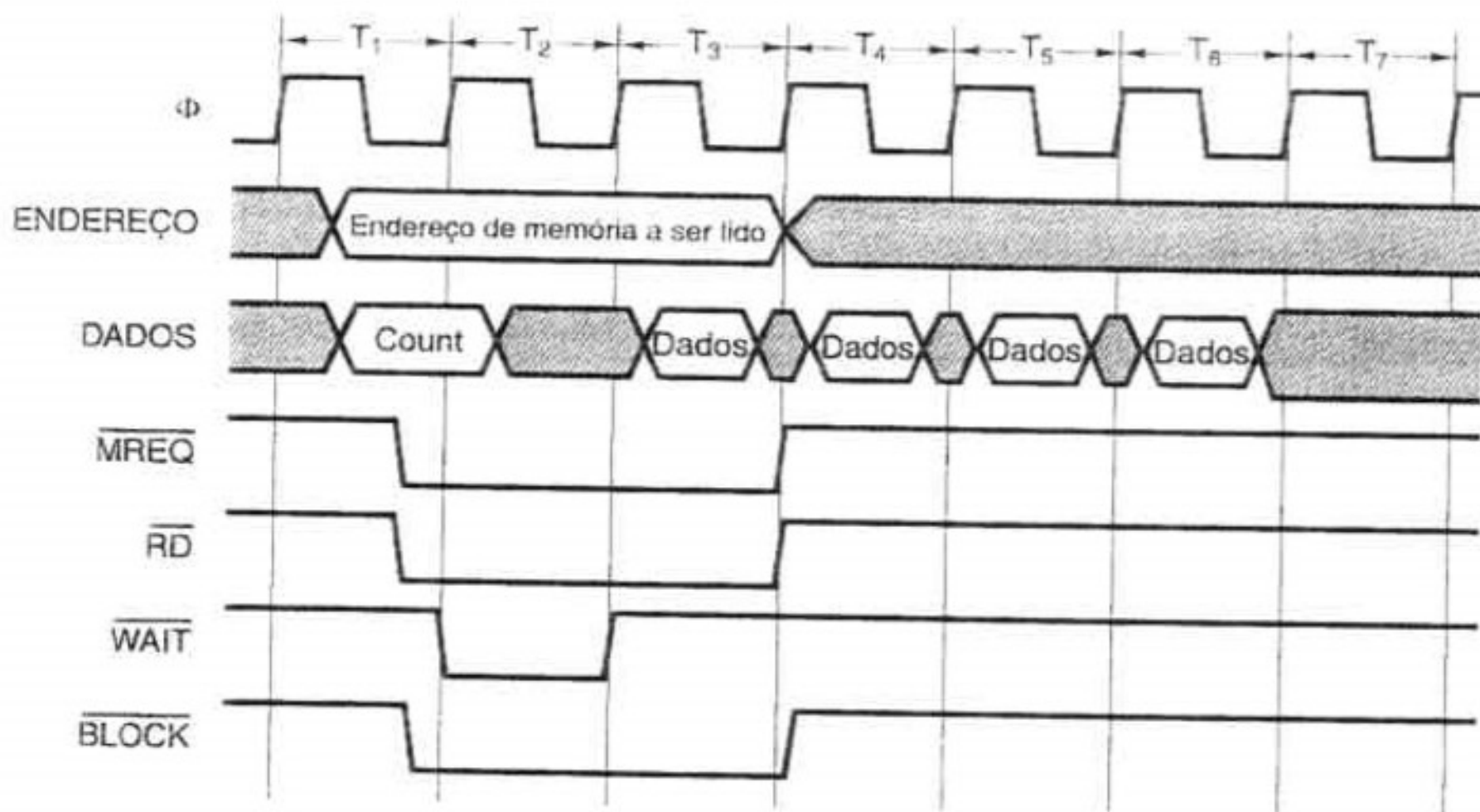
Figure 3-40. Decentralized bus arbitration.

Operações de Barramento

- Os ciclos de barramento, podem fazer transferências em bloco, tanto do processador para CPU, quanto da CPU para o processador.
- Transferir dados em bloco, quando se usa memória cache, é mais eficiente do que transferir dados de forma individual.
- Quando uma leitura de bloco é iniciada, o mestre da transferência de dados informa ao escravo quantas palavras serão lidas, por exemplo, colocando o número de palavras nas linhas de dados em T1.

Operações de Barramento

- Em vez de entregar apenas uma palavra, o escravo entrega uma palavra durante cada ciclo, até se esgotarem o número de palavras.
- A figura seguinte mostra essa situação. Note que agora temos o sinal BLOCK, que é ativado para indicar que se trata de uma transferência de bloco.

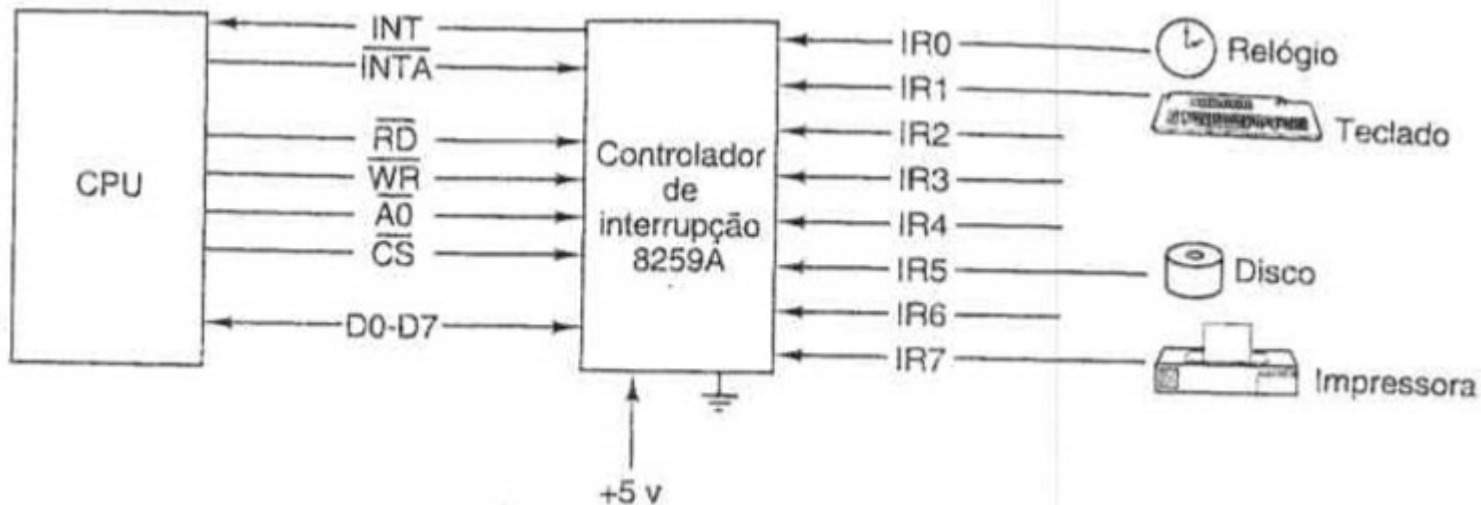


Operações de Barramento

- Um outro ciclo importante de barramento é utilizado para manipular interrupções.
- A sinalização da interrupção requer o barramento.
- Uma vez que vários dispositivos podem causar interrupções simultaneamente, os mesmos problemas de arbitragem estão presentes.
- Solução: usar um árbitro central (controlador de interrupção) e atribuir prioridades para as interrupções.

Controlador de interrupção

- Normalmente presente no chipset da placa mãe. Nos Pcs o chip normalmente é o 8259A.



Controlador de Interrupção

- Quando uma ou mais entradas são ativadas, o controlador ativa INT (sinal de entrada da CPU).
- A CPU responde com INTA (INT Acknowledge).
- O controlador coloca o número da entrada no barramento de dados.
- A CPU usa esse número para indexar um vetor de ponteiros para função (vetor de interrupção).
- Cada entrada deste vetor contém o endereço de uma função que irá tratar a interrupção em questão.

Controlador de Interrupção

- No interior do 8259A há diversos registradores que a CPU pode ler ou escrever usando ciclos de barramento comuns (o ciclo de barramento para tratar a interrupção é um ciclo especial).
- Esses registradores podem ser usados para mascarar (desabilitar) interrupções, além de definirem prioridades.
- Dois chips 8259A podem ser ligados em cascata, caso haja mais de 8 interrupções no sistema.