

Processador

S. W. Song

MAC 412 - Organização de Computadores

Processador

Veremos:

- Pinos típicos do processador.
- Barramento e arbitragem no seu uso.
- Em aulas futuras:
 - Arquiteturas microprogramadas. CISC - Complex Instruction Set Computer
 - Arquiteturas não microprogramadas. RISC - Reduced Instruction Set Computer

Baseado no livro de Tanenbaum - Structured Computer Organization.

Microprocessador numa pastilha

A interface de um microprocessador com o resto do sistema é através de pinos de entrada e saída da pastilha.

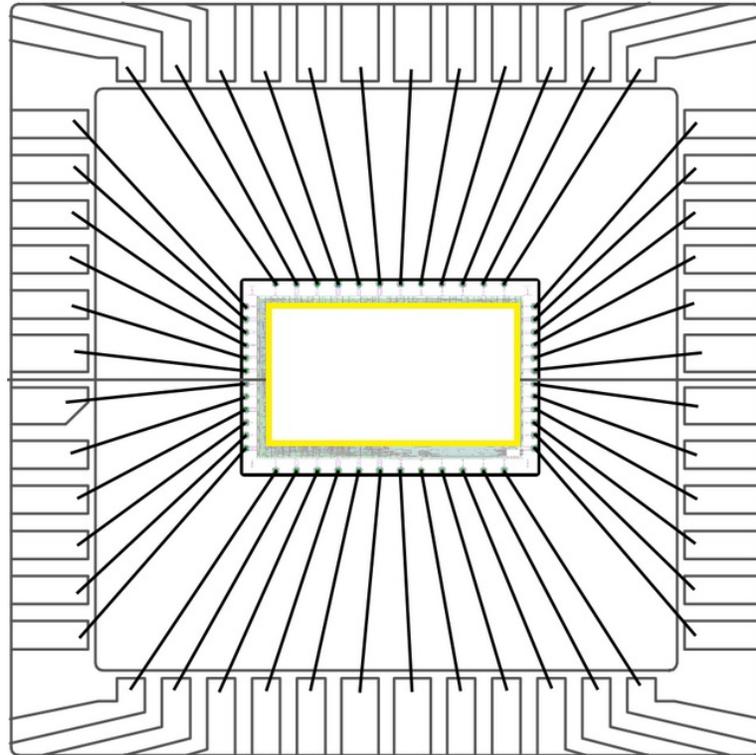
Tipicamente há algumas centenas de pinos numa pastilha.

Existem três tipos de pinos:

- Pinos para endereços
- Pinos para dados
- Pinos para controle

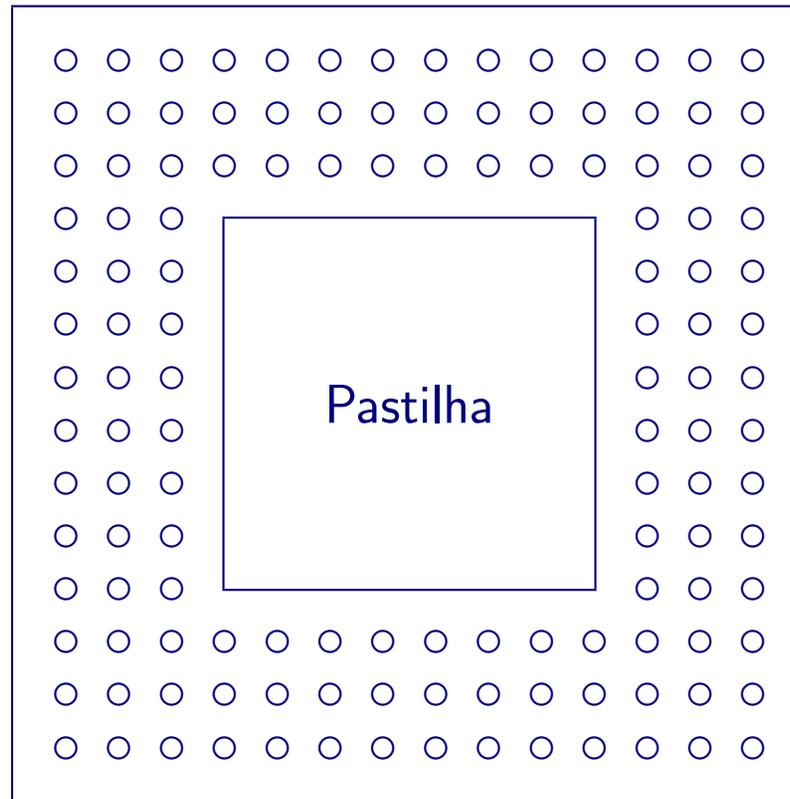
Microprocessador numa pastilha

Os pinos de entrada e saída em geral ficam na periferia da pastilha.



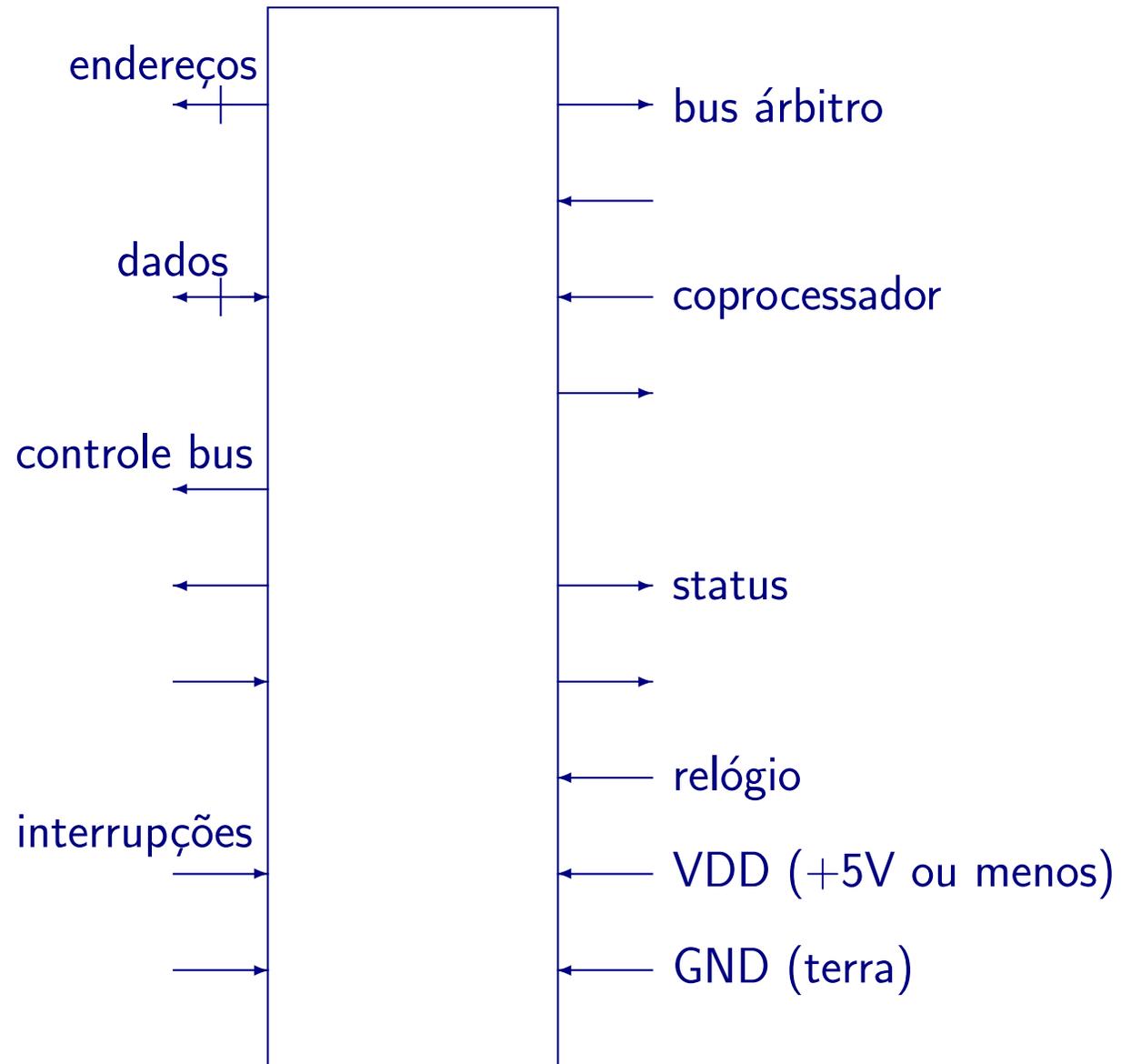
Microprocessador Intel 386

A pastilha do processador Intel 386 possui 132 pinos.

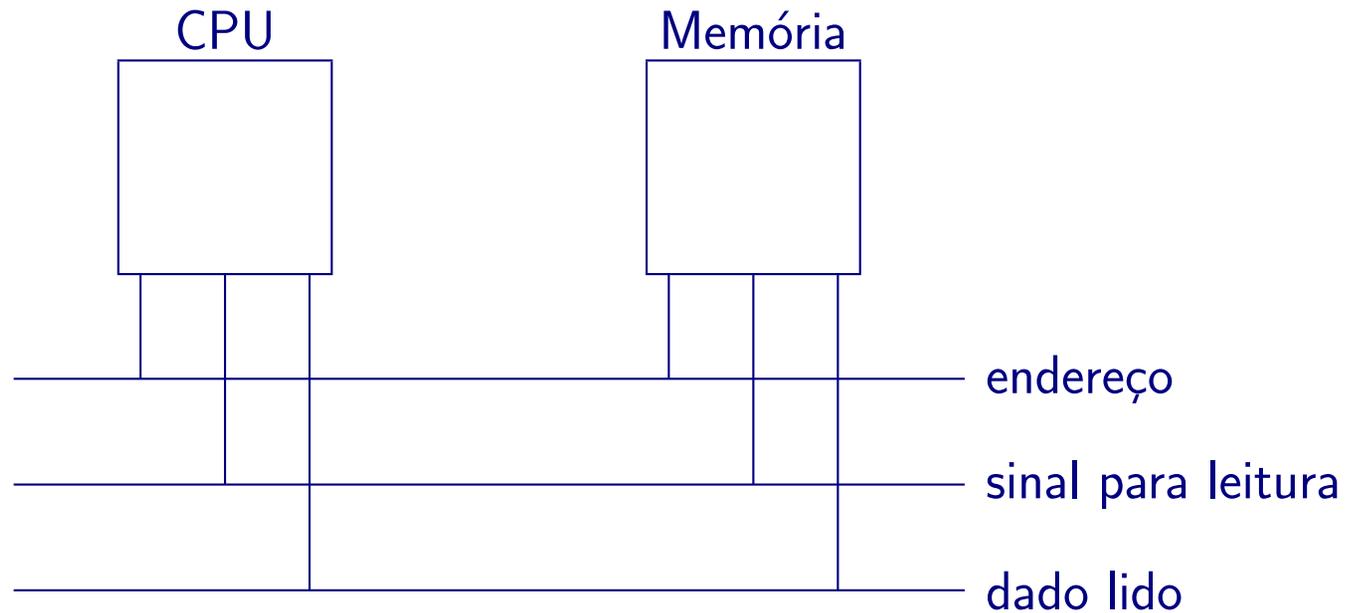


O processador Intel Pentium Pro possui 387 pinos.

Pinos típicos

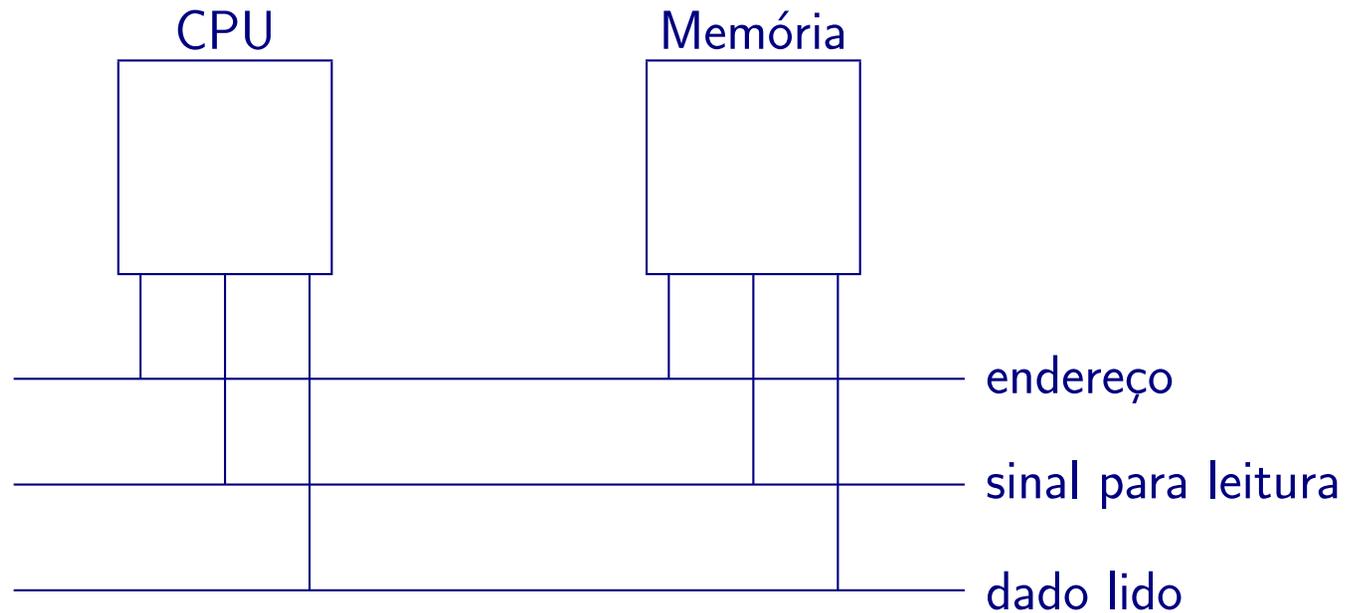


Barramento ou bus



Os pinos do processador são ligados aos pinos de outras pastilhas (como memória) através de linhas paralelas formando um barramento ou “bus”. A largura de um barramento é tipicamente algumas centenas linhas paralelas.

Acesso a memória



Para ler um dado da memória, o processador coloca o endereço nos pinos do endereço, e valida uma linha de controle para informar a memória a ler uma palavra de memória. A memória responde colocando a palavra lida no barramento de dados. (Veremos mais detalhes daqui a pouco.)

Mestre e escravo

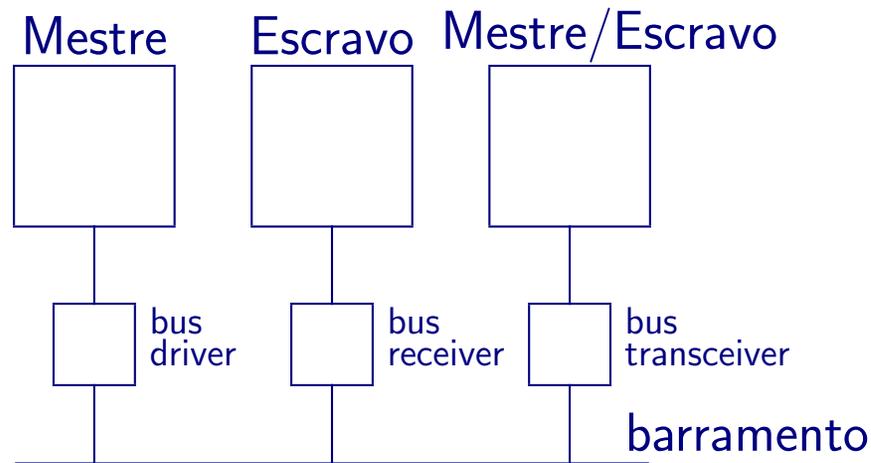
Os dispositivos conectados a um barramento podem ser:

- ativos: tomam iniciativa para começar uma transferência no barramento. Nesse caso o dispositivo é dito Mestre.
- passivos: ficam esperando a solicitação de algum outro dispositivo para fazer a transferência. O dispositivo passivo é conhecido como Escravo.

Exemplo de mestre/escravo

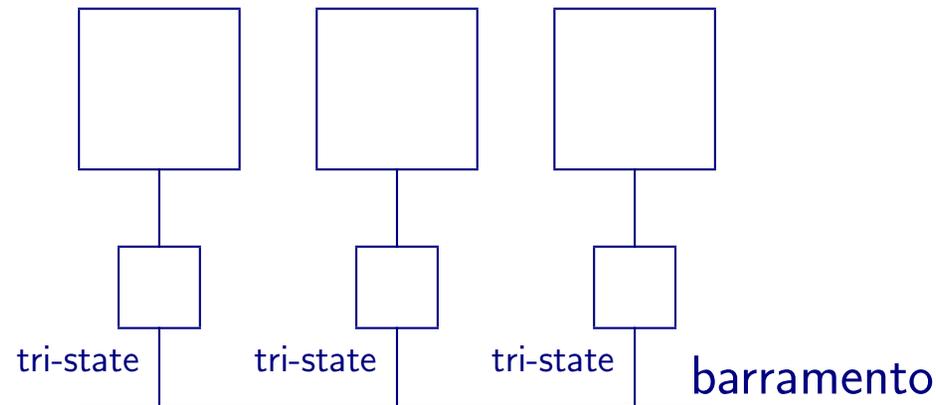
1. A CPU (mestre) manda o controlador de disco (escravo) ler um bloco de disco.
2. O controlador de disco (mestre), tendo lido um bloco, comanda a memória (escravo) para receber os dados lidos, através de **DMA - Direct Memory Access**.
 - Assim, um mesmo dispositivo pode as vezes funcionar como mestre, outras vezes como escravo.
 - A memória, entretanto, nunca pode funcionar como mestre.

“Bus drivers”



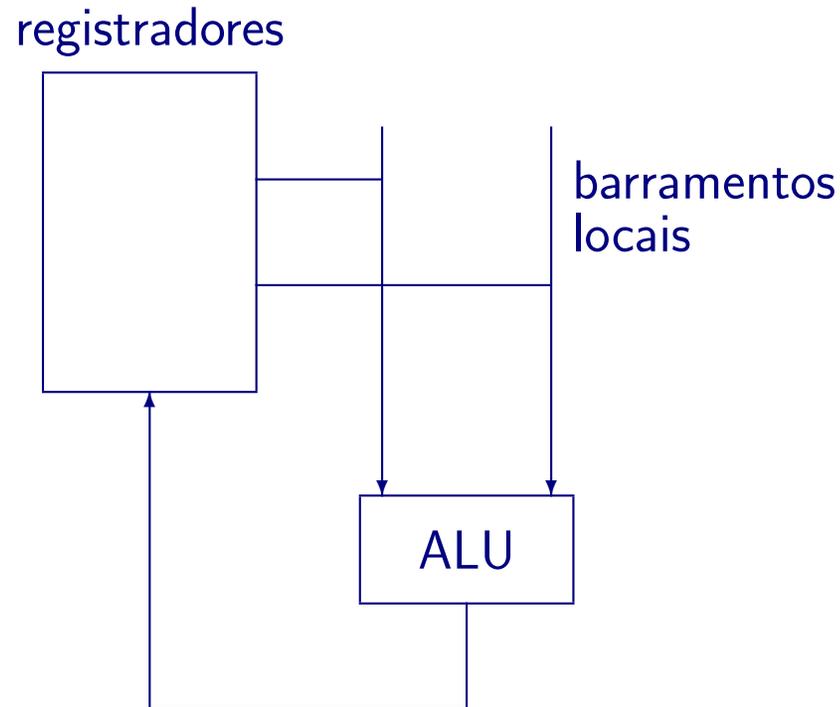
- “Bus driver” não é motorista de ônibus :-)
- “Bus drivers” amplificam os sinais saindo de uma pastilha antes de serem colocados no barramento.
- “Bus receivers” serve de interface para receber um sinal do barramento antes de entrar na pastilha.
- “Bus transceiver” é capaz de enviar e também receber dados do barramento.

Interface de barramento “tri-state”



- Em geral vários dispositivos estão conectados a um barramento, mas apenas alguns deles estão de fato utilizando o mesmo.
- Para não interferir no barramento quando um dispositivo não o está usando, as interfaces são em geral do tipo “tri-state”.
- “Tri-state” significa três estados: 0, 1 ou em aberto (ou alta impedância). Os dispositivos que não estão usando o barramento ficam no estado em aberto.

Barramentos locais (dentro do processador)



Além do barramento central do sistema que liga o processador e memória, há também barramentos locais dentro do processador.

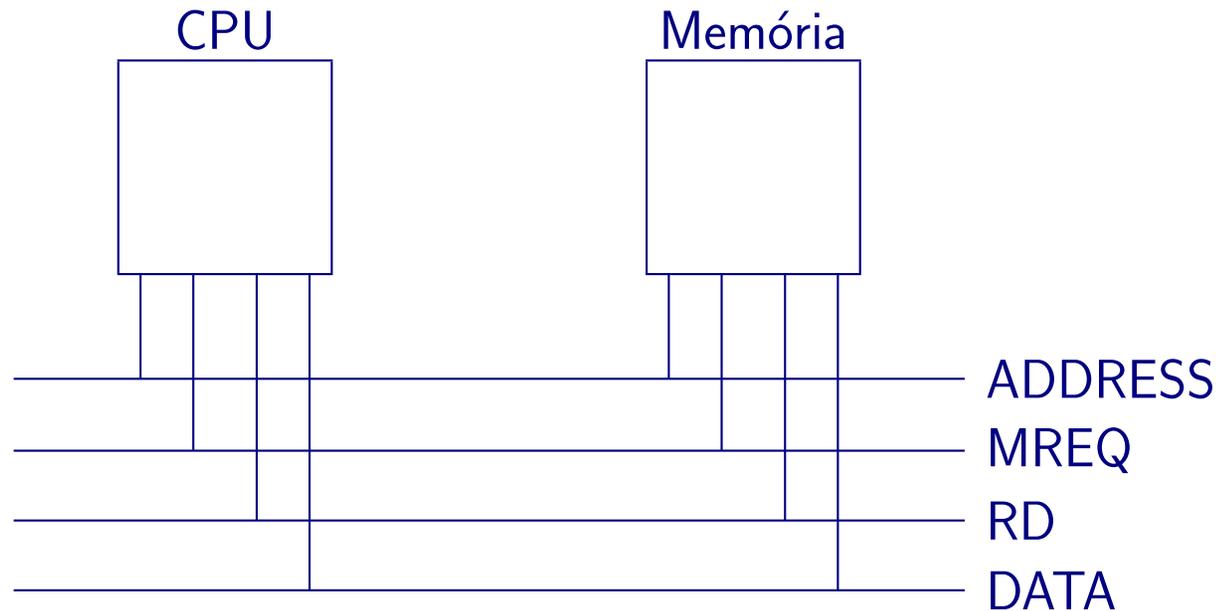
Alguma terminologia

- Um sinal é dito validado (em inglês “asserted”) se ele possui o valor apropriado para causar uma certa ação.
- Para alguns pinos, validar um sinal significa fazer o sinal igual a 1; para outros pinos, validar um sinal significa fazê-lo igual a 0.
- Neste último caso, colocamos uma barra em cima do sinal, para indicar que o sinal validado é o zero.
- Assim, \overline{LEIA} significa a ação de leitura é ligada com o valor 0.

Barramento síncrono

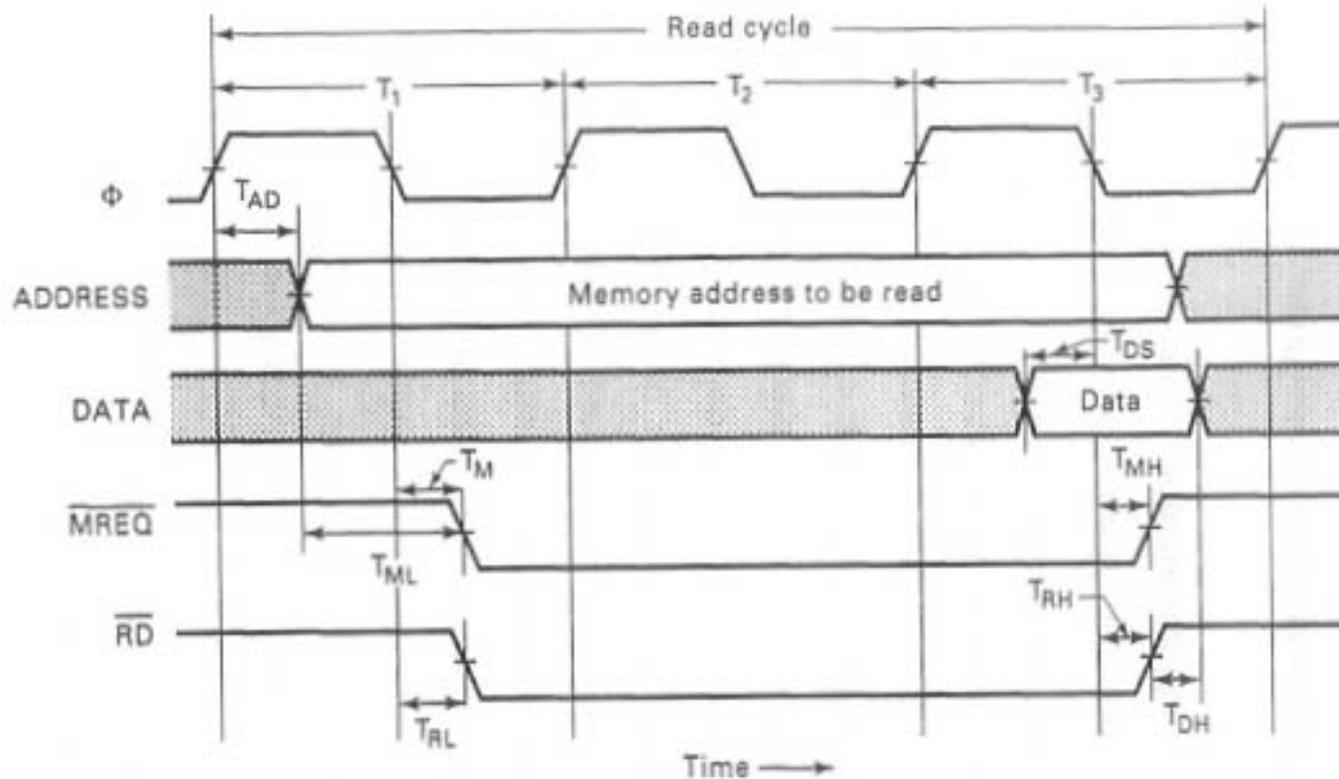
- Um barramento síncrono tem seu funcionamento controlado por um sinal de relógio.
- Por exemplo, uma frequência de 1 MHz corresponde a um ciclo de 1 micro-segundo (μs), 1 GHz corresponde a um ciclo de 1 nano-segundo (ns).
- Todas as atividades do barramento ocorrem em um número inteiro de ciclos do barramento.

Exemplo: ciclo de leitura



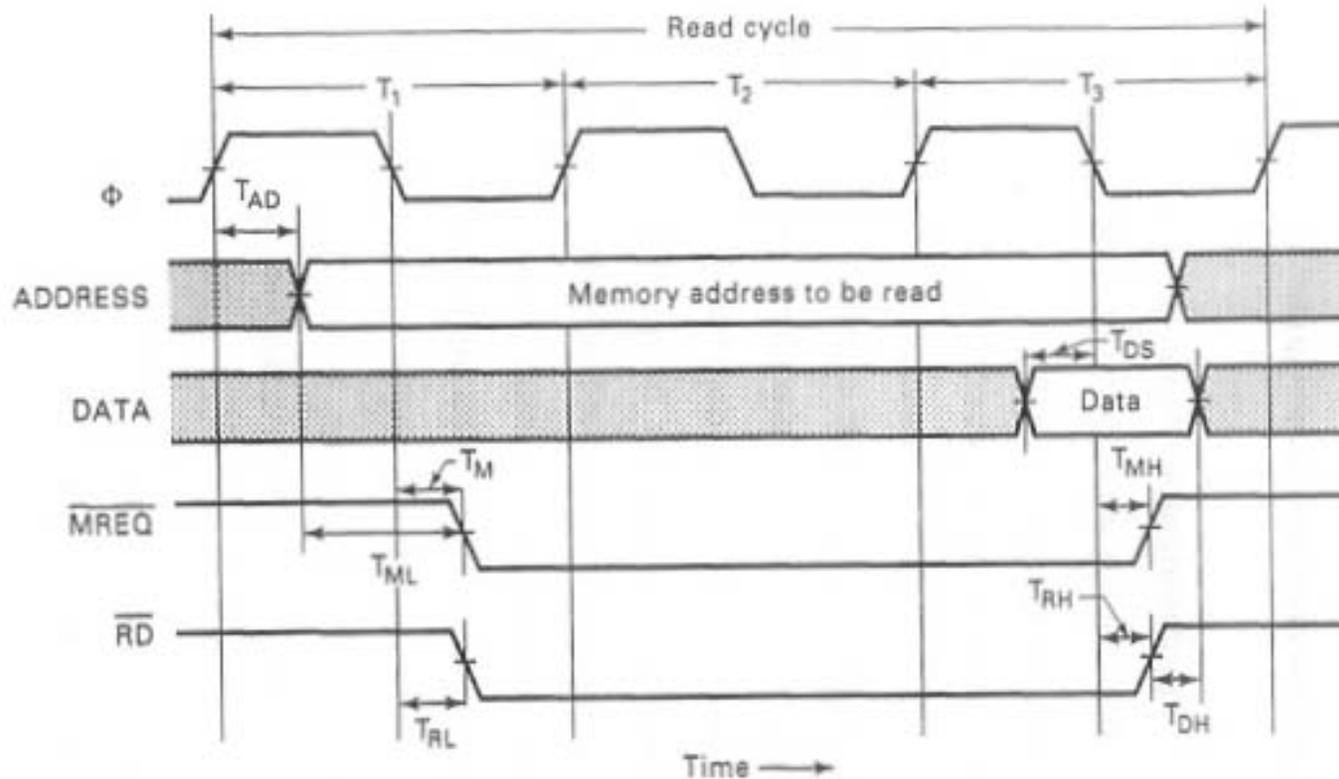
- Considere o ciclo de leitura num barramento síncrono.
- ADDRESS e DATA indicam os barramentos de endereço e dado, respectivamente.
- Os sinais MREQ e RD serão explicados.

Exemplo: ciclo de leitura



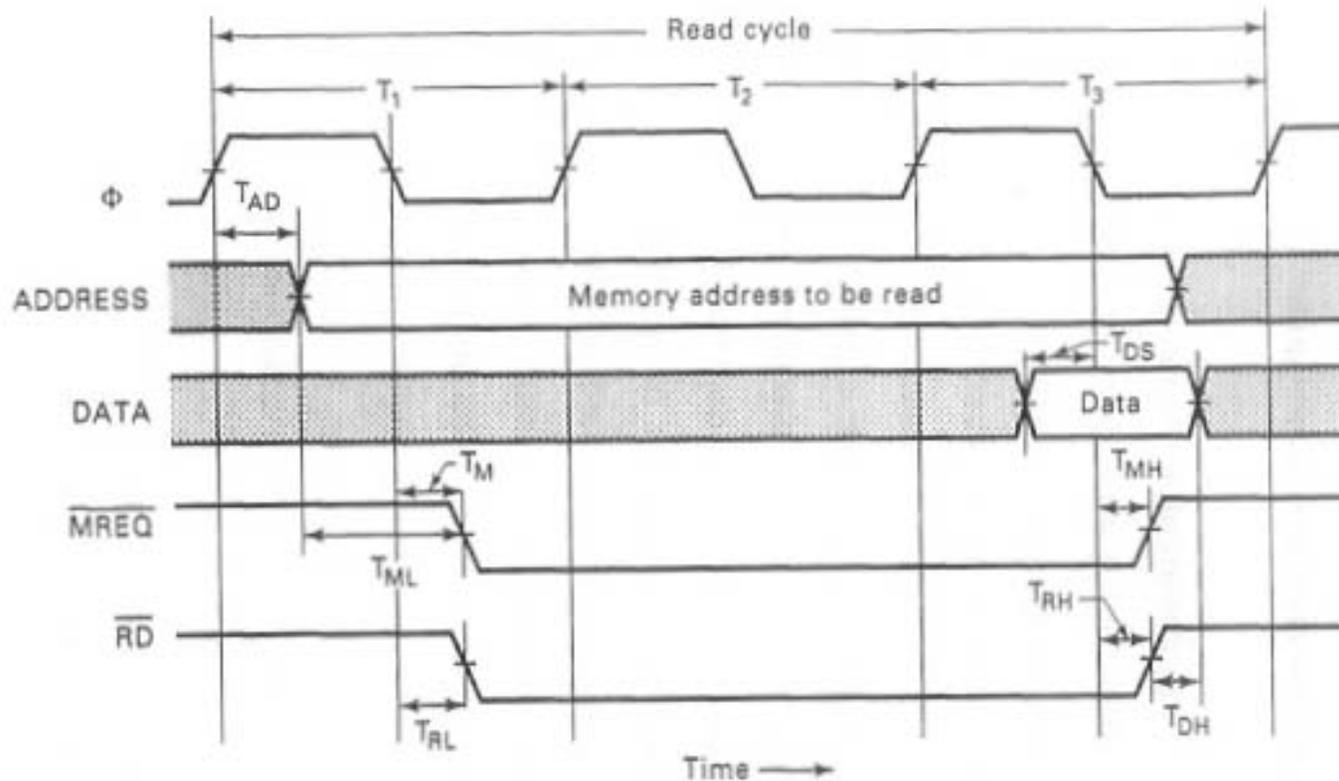
- Um ciclo de leitura é consituído de 3 ciclos do barramento.
- No primeiro ciclo, o endereço é colocado pela CPU nas linhas de endereço.

Exemplo: ciclo de leitura



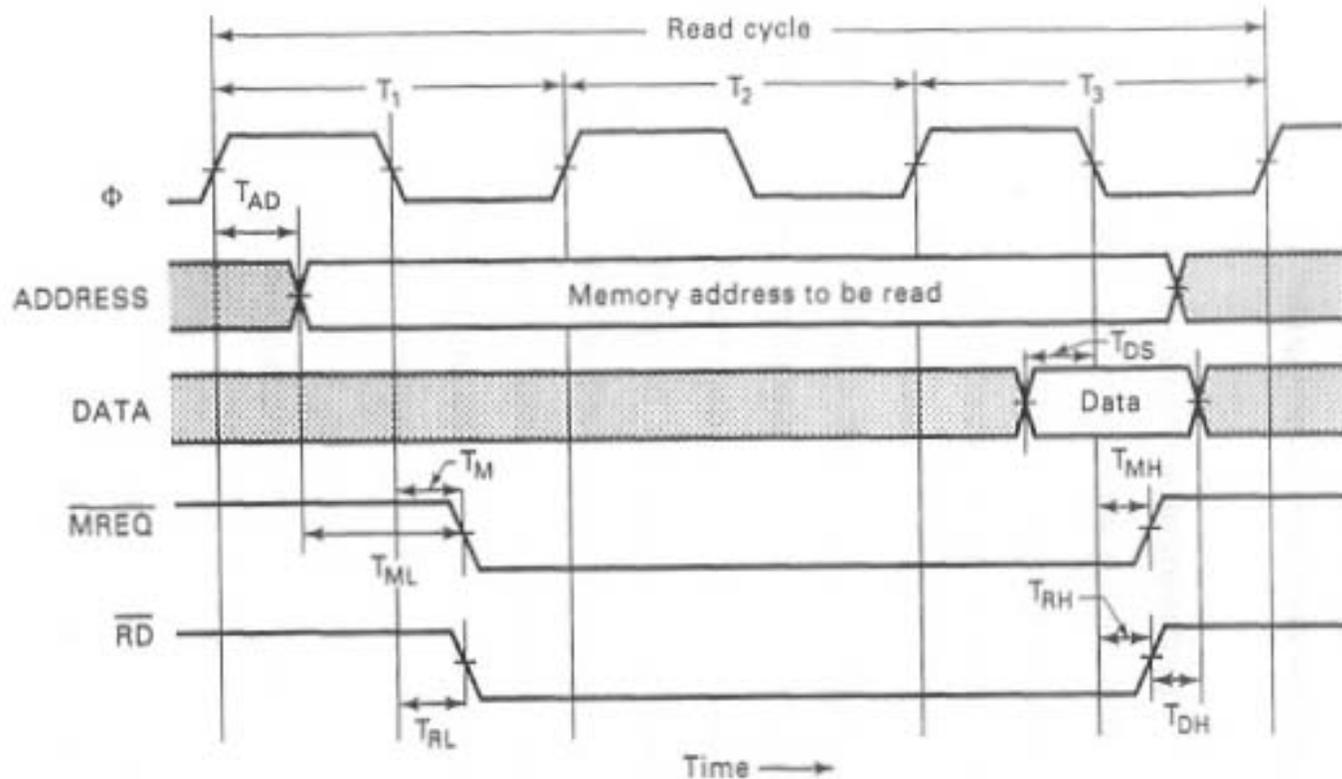
- Em seguida, as linhas \overline{MREQ} e \overline{RD} são validadas (com 0), significando respectivamente pedido de acesso de memória e o tipo de acesso é leitura.

Exemplo: ciclo de leitura



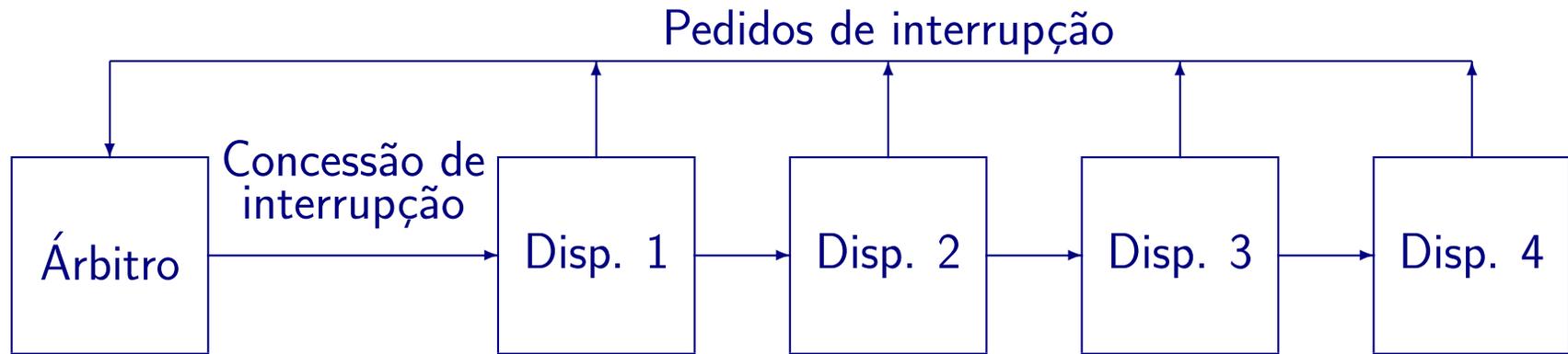
- Nada acontece no segundo ciclo, para dar tempo para a memória decodificar o endereço e colocar a palavra lida, no terceiro ciclo, no barramento de dados.

Exemplo: ciclo de leitura



- Na descida do sinal de relógio do terceiro ciclo, a CPU lê os dados do barramento de dados, guardando-os em um registrador. Em seguida, a CPU coloca 1 nos sinais $MREQ$ e RD , inibindo-os.

“Bus arbiter” por “daisy chaining”



- A concessão para o uso do barramento é enviada para o dispositivo 1. Se este não pediu o uso, então passa o sinal de concessão para o dispositivo 2 e assim sucessivamente até chegar a um dispositivo pediu o uso.
- Este dispositivo, ao receber o sinal de concessão, deixa de encaminhar o sinal adiante e fica dono do barramento.
- Este processo é conhecido pelo nome de “daisy chaining”.

“Bus arbiter” - outras formas

- O árbitro do barramento pode também usar a forma de concessão chamada “polling” em que cada dispositivo é consultado, um a um, para ver se precisa de interrupção.
- Outra forma de interrupção é a chamada **vetorada** em que o endereço de tratamento de endereço é fornecido pelo dispositivo.