

# BARRAMENTOS E DISPOSITIVOS DE E/S

---

## 1. INTRODUÇÃO

A comunicação em computadores é motivada pela necessidade de intercâmbio de dados entre o microprocessador e outros componentes (teclado, terminal, impressora, etc...).

Qualquer operação de transferência de informação vai envolver basicamente três entidades: uma entidade fonte, uma entidade destino e os dados propriamente ditos. As entidades fonte e destino podem se caracterizar por uma das três classes de elementos: CPU, memória, periférico. A informação a ser transferida pode ser uma instrução ou dados a serem manipulados.

A tecnologia de comunicação é um fator de extrema importância na arquitetura de computadores, uma vez que os meios utilizados para interconectar os diversos componentes de um sistema computacional serão definitivos para a obtenção de um bom ou mal desempenho global.

O objetivo deste capítulo é discutir os principais aspectos relativos à tecnologia existente para a comunicação entre os diferentes componentes de um computador.

## 2. DISPOSITIVOS DE ENTRADA/SAÍDA - E/S

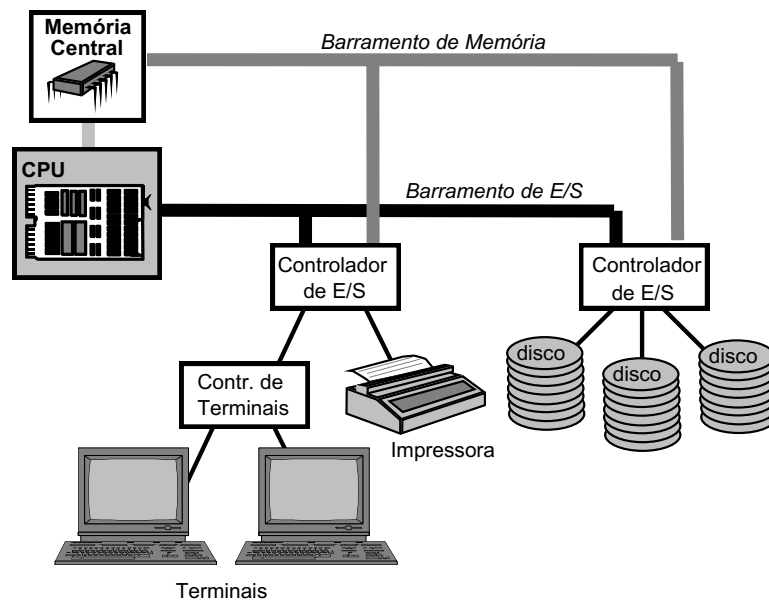
Estes componentes permitem a troca de dados a serem tratados pelo computador com um operador humano (programador, digitador) ou com outros equipamentos (outro computador, por exemplo).

Dois principais modos da organização dos elementos de E/S podem ser considerados:

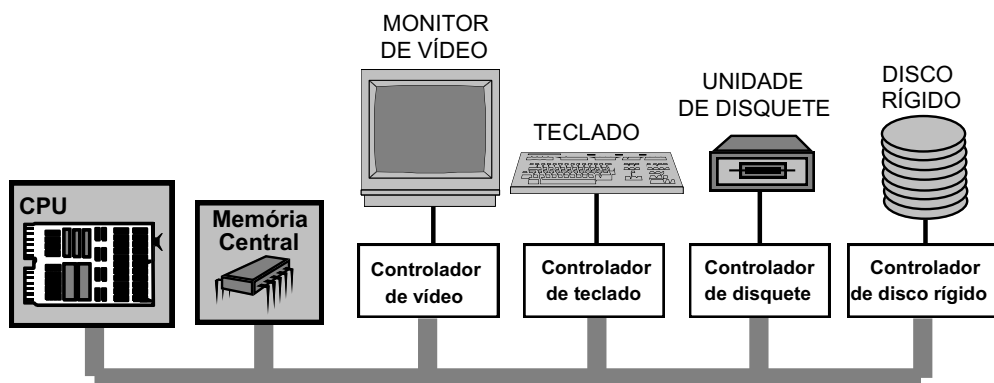
- No primeiro modo, mais adaptado para computadores de grande porte (mainframes), os elementos de E/S consistem de processadores especializados, que tem por função tratar a informação antes de transmiti-la ao computador. Esta transmissão é feita geralmente via interrupção, a qual faz com que o processador interrompa o seu processamento normal para tratar a informação recebida via periférico de E/S (Figura 1). Na figura, pode-se distinguir também a presença de diversos barramentos (barramento de memória, barramento de dados, barramento de E/S, e um terceiro barramento que permite a troca de dados entre a memória principal e a CPU. Uma vantagem deste modo é a simultaneidade de tratamento do processador e do canal;
- No segundo modo, mais adaptado aos microcomputadores, a organização dos elementos de E/S é bem mais simples. Estes sistemas são geralmente dispostos num chassis, na qual é instalada a placa de circuito impresso com os diferentes componentes do computador (placa mãe). A placa mãe comporta também um

barramento em circuito impresso, sobre o qual vão estar conectados um grande número de circuitos e conectores (slots) para a introdução de circuitos ou cartas complementares. A Figura 2 mostra a estrutura lógica de um microcomputador, na qual se pode distinguir o microprocessador, a memória principal e os diversos periféricos. Cada periférico compreende duas partes; o controlador e o periférico propriamente dito. O controlador de periférico se apresenta normalmente na forma de uma carta conectada a um slot do barramento (controlador de vídeo) ou de um circuito integrado instalado na carta mãe (controlador de teclado).

O papel do controlador de periféricos é fazer a gestão do periférico em sí, assim como fazer a gestão do acesso ao barramento do computador para permitir a troca de dados. Um controlador pode igualmente ler ou escrever diretamente as informações na memória principal do computador. Neste caso, a técnica utilizada para este tipo de comunicação é a chamada "acesso direto à memória" ou DMA (Direct Memory Access).



**Figura 1** - Organização das E/S de um computador de grande porte.



**Figura 2** - Organização lógica de um computador.

Alguns exemplos de periféricos de entrada saída são: terminais de vídeo (comunicação com operador); modem (comunicação com outros computadores via linha telefônica); mouse (utilizado como forma alternativa de comunicação com o operador -

apontar opções - "menus" - na tela); impressoras (saída de dados impressos - a impacto, jato de tinta, laser).

### 3. MODOS DE TRANSMISSÃO

Como é de nosso conhecimento, as informações manipuladas por um microprocessador são representadas na forma binária, normalmente organizadas em palavras de dados que podem apresentar diferentes tamanhos: 8 bits (bytes), 16 bits (words), etc...

Dois modos de transmissão de informação codificada em binário podem ser consideradas: a transmissão paralela e a transmissão serial.

#### 3.1. Modo de transmissão paralelo

Neste modo de transmissão, os bits compondo uma palavra de dados são conduzidos ao longo de um conjunto de vias, sendo uma via para cada bit de dado. Um exemplo deste modo de transmissão são os barramentos (internos e externos) de microcomputadores. No modo paralelo, além das vias de dados, outras vias são necessárias para viabilizar a comunicação, particularmente uma via para definir uma referência de tensão (terra) e uma outra para indicar a disponibilidade de um dado (ready).

Como pode ser observado na Figura 3, esta configuração é prevista suportar a comunicação num único sentido. No caso em que se queira implementar uma comunicação bidirecional, outras vias devem ser introduzidas no esquema, particularmente uma que permita indicar o sentido da comunicação (leitura/escrita).

Um exemplo típico de comunicação paralela corresponde ao utilizado nos computadores pessoais da IBM e seus compatíveis para o envio de dados às impressoras — a interface Centronics. Esta interface permite o envio de dados em 8 bits paralelos de cada vez. Este modo de transmissão apresenta uma vantagem particular que é a simplicidade no que diz respeito ao controle dos dados.

Uma vez que o computador pode enviar dados com uma velocidade bastante superior à da impressora (no que diz respeito à impressão), esta vai armazenando os dados recebidos numa região de memória dedicada (buffer) e indica ao computador, através de um sinal quando o buffer está completo, de maneira a evitar que este continue a enviar dados. Isto é feito através de um sinal especial denominado BUSY, que quando setado a um indica a saturação do buffer da impressora.

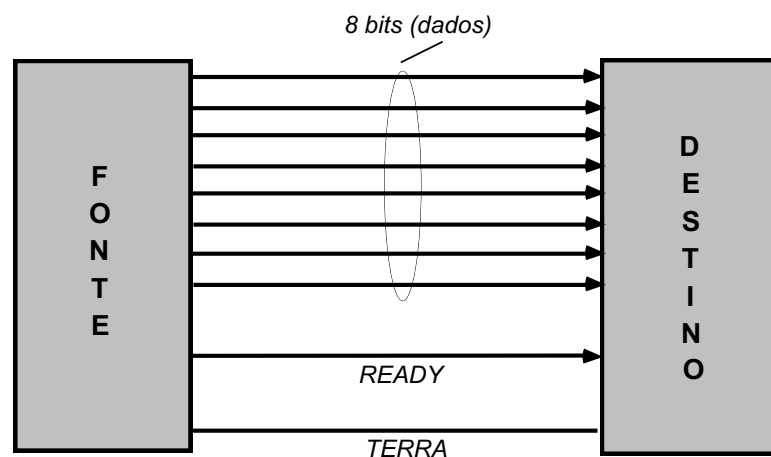


Figura 3 - Esquema simplificado do modo de transmissão paralelo.

Além disso, esta interface permite a implantação de um outro mecanismo de comunicação: quando o computador quer enviar um dado à impressora ele indica através de um sinal denominado STROBE.

Quando a impressora recebeu o dado ela sinaliza o recebimento através de um sinal denominado ACKLNG. A esta técnica denominamos "handshake bidirecional".

No caso do IBM-PC, a linha BUSY é utilizada para implementar um modo de comunicação em "polling".

Isto significa que, quando ele quer enviar um dado ele entra num loop de espera até que o sinal BUSY esteja a zero, permitindo assim o envio do dado. Um exemplo de loop implementando este modo é ilustrado a seguir:

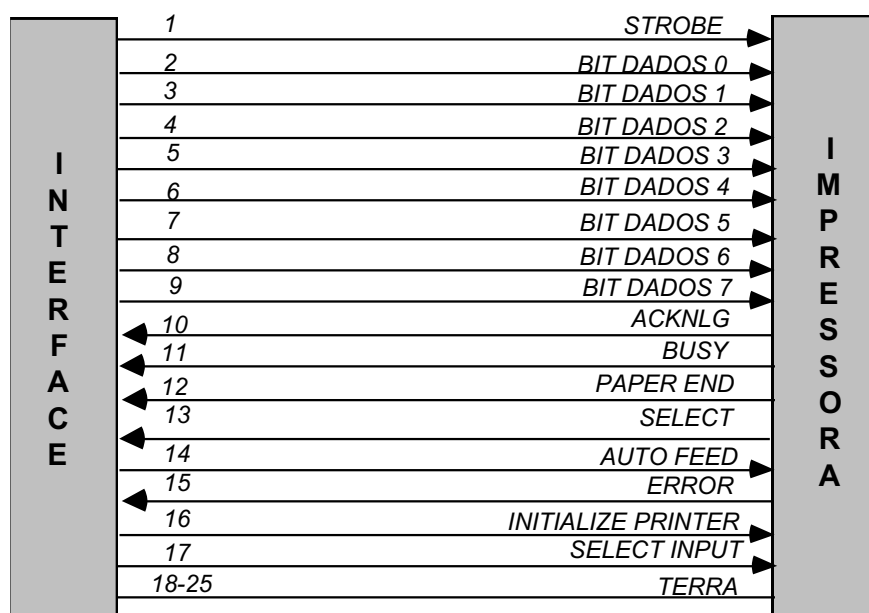
```
MOV DX,ESTADO
INICIO: IN AL,DX      ; fica no loop até que BUSY caia a zero
        TEST AL,80H   ; BUSY a zero implica que o bit 7 da palavra
        JZ INICIO     ; de estado esteja a 1
```

A rotina de envio de dados do IBM PC funciona da seguinte maneira. Quando um caractere deve ser enviado à impressora, ele coloca o dado na porta de dados da interface (o endereço da porta de dados é 03BCH) onde ele será mantido. Em seguida ele inicia a operação de polling sobre o sinal BUSY, testando a palavra de estado da interface (bit 7 da porta 03BDH) até que ele esteja a zero. Neste loop, é ainda inserido um contador que permita sair do loop após um determinado tempo. Isto é feito para evitar que a rotina entre em loop eterno no caso da não existência de uma impressora conectada à interface o tempo de espera (timeout) é assim limitado a cerca de 16 segundos.

Quando a linha BUSY vai a zero, a rotina dá um pulso no sinal STROBE (bit 0 da porta 03BEH) indicando à impressora que um novo byte está pronto para ser enviado.

A linha ACKNLG é utilizada normalmente para gerar um sinal de interrupção informando ao computador que o dado enviado foi lido, permitindo assim o envio de um novo dado. Isto significa que, por esta técnica, os dados podem ser armazenados num buffer FIFO e serem enviados à medida que ocorra uma interrupção gerada pela impressora.

A Figura 4 ilustra as linhas utilizadas pela interface Centronics do IBM-PC que é composta de 25 pinos. Como podemos ver além dos sinais descritos anteriormente, outros sinais são definidos na interface: PAPER END, AUTO FEED, ERROR, INITIALIZE PRINTER e SELECT INPUT.



**Figura 4** - Pinagem da interface Centronics no IBM-PC.

### 3.2. Modo de transmissão serial

Nos casos em que não existe a necessidade de altas taxas de dados a serem transmitidas, o número de linhas necessárias à transmissão de dados pode ser reduzida convertendo-se os dados a serem transmitidos num feixe de bits serial.

Se a taxa de transmissão de dados é suficientemente baixa, são necessárias apenas duas vias para a transmissão do feixe de bits, uma para cada direção, embora, normalmente seja necessária também uma linha de terra conectando os dois dispositivos. A informação contida num feixe de bits serial é representada por um sinal dependente do tempo.

Neste modo de comunicação, os bits são caracterizados por uma duração específica denominada período de baud (baud period). Na sua forma mais simples de representação, um bit 1 é representado por um sinal TTL alto de duração igual a 1 baud period, e um bit 0 por um sinal TTL baixo de duração igual a 1 baud period. Para que o envio de informação funcione corretamente neste modo, transmissor e receptor devem estar sincronizados pela mesma taxa de transmissão (baud rate) que é regida por um relógio (em cada um dos dispositivos envolvidos na comunicação).

Os bits são geralmente transmitidos em grupos separados, tipicamente de 8 bits, denominados caracteres. Num formato assíncrono padrão, o espaço entre a transmissão de dois caracteres é indicado por um sinal TTL alto e é denominado marca. O transmissor indica ao receptor o início da transmissão de um caractere através de um sinal TTL baixo de duração de um baud period — este sinal corresponde a um bit 0 e é denominado start bit.

Seguindo o start bit, temos os bits de dado, iniciando pelo bit mais significativo\_a palavra de dados pode ser definida de 5 a 8 bits. Após a palavra de dados, um bit adicional (P) pode ser inserido para permitir a verificação de eventuais erros de transmissão — o bit de paridade. Este bit permite indicar ao receptor a paridade (o número de bits a 1) da palavra de dados (incluindo o próprio bit de paridade). A paridade pode ser definida como sendo par ou ímpar: na paridade par, o bit P = 1 se o número de bits 1 é ímpar e P = 0 se o número de bits a 1 é par, de modo que a paridade do caractere, incluindo o bit de paridade, seja sempre par.

Se, na transmissão de um dado, a paridade foi modificada, um flag de paridade é modificado e pode ser então testado pela CPU (para a implementação de algum algoritmo de recuperação de erros de transmissão).

Após o bit de paridade o transmissor insere um número de bits indicando o fim da transmissão do caractere — são os stop bits. Os stop bits são caracterizados por um sinal TTL alto de duração igual a 1, 1,5 ou 2 baud period, o que é comumente especificado como 1, 1,5 ou 2 stop bits. A Figura 5 ilustra o formato do sinal de transmissão de uma palavra de 8 bits, mostrando o start bit, o bit de paridade e um stop bit.

Este tipo de comunicação é caracterizado como um modo de comunicação assíncrona, uma vez que não existe a definição do tempo que pode decorrer entre o envio de um caractere e o próximo.

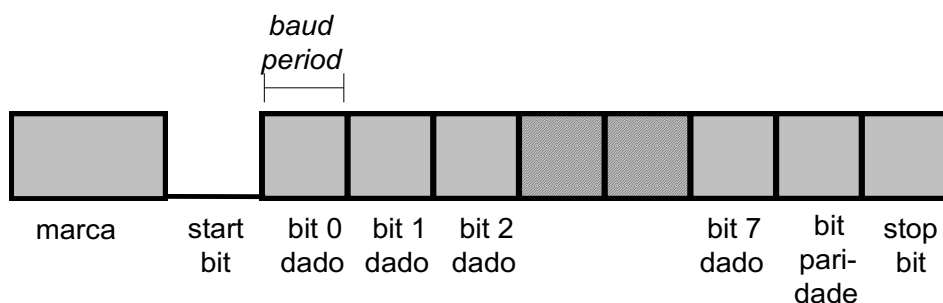
As taxas de transmissão utilizadas na prática podem variar de 50 a 19200. A maioria dos computadores pode trabalhar em qualquer uma destas taxas, mas normalmente alguns periféricos exigem a adoção de uma baixa taxa de transmissão. Só para que possamos compreender a importância da taxa de transmissão: se considerarmos a transmissão de um caractere, sem paridade e com um stop bit, isto significa que teremos, em cada comunicação 10 bits; assim, se a taxa de transmissão é de 300, isto significa que poderemos transmitir 30 caracteres por segundo.

A maneira mais utilizada para implementar a comunicação serial é a utilização de um componente dedicado a esta finalidade denominado UART (Universal Asynchronous Receiver Transmitter). Uma UART é geralmente caracterizada por pelo menos quatro portas de E/S: uma porta de controle, uma porta de estado, uma porta de entrada e uma porta de saída.

Bytes enviados à porta de saída de uma UART são convertidos num feixe de bits serial para a transmissão. Do mesmo modo, um feixe de bits serial chegando na UART serão convertidos num byte e armazenados na porta de entrada. A conversão simultânea de um feixe de dados possibilita a comunicação em duas direções e é denominada full-duplex.

Esta comunicação vai exigir então duas vias para a transmissão dos feixes seriais de bits. Uma variante deste modo de comunicação é o half-duplex que permite a comunicação nas duas direções, sendo que apenas uma das direções está ativa por vez (a vantagem desta técnica é a necessidade da utilização de um único canal de dados).

Um exemplo típico de UART é o componente 8250 fabricado pela National e utilizado nos IBM-PC. Ela permite a troca de dados no modo serial assíncrono numa taxa de transmissão de até 9600 baud, esta taxa sendo obtida pela divisão da frequência do clock de entrada por um número (programável) de 16 bits. Ela é provida também de um controlador interno de interrupções, o que permite a ela tratar até 4 níveis de interrupção.



**Figura 5** - Dados transmitidos de forma serial.

## 4. BARRAMENTOS

Um barramento é uma estrutura de interconexão capaz de associar diversos componentes de um computador. O barramento de um microcomputador é composto de 50 a 100 fios paralelos gravados sobre circuito impresso.

Conectores, regularmente espaçados, vão permitir a ligação mecânica com cartões de memória ou circuitos de E/S (barramento sistema externo). Um barramento pode ainda servir a funções específicas, como por exemplo, a conexão de um microprocessador com uma memória local ou com um coprocessador (barramento local). Finalmente, no interior de microprocessadores, existem barramentos conectando os diversos módulos internos compondo a sua arquitetura (barramento interno).

Segundo a maneira como ocorrem as trocas de informações num barramento, pode-se classificá-los em síncronos e assíncronos.

### 4.1. Barramentos síncronos

Os barramentos síncronos são dotados de uma linha específica de relógio, pilotado por um oscilador a quartzo. Os sinais de relógio apresentam a forma quadrada ou retangular, cuja frequência varia de 1 a 500 Mhz. Toda operação sobre o barramento síncrono deve ser feita num número inteiro de ciclos de relógio, sendo que, neste caso, a um ciclo de relógio, é dada a denominação, ciclo do barramento.

A Figura 6 apresenta o diagrama temporal que permite exemplificar o funcionamento de um barramento síncrono. No exemplo, é considerado um relógio com frequência de 4 Mhz, que define um ciclo de barramento igual a 250 ns (nanosegundos). Os cronogramas representam uma operação de leitura de informação em memória (um byte). A operação necessita de três ciclos de barramento, ou seja 750 ns, do início do ciclo T1 ao fim do ciclo T3.

Pode-se notar sobre os cronogramas que as transições dos sinais não são verticais, mas oblíquos. Isto significa que o tempo de transição é diferente de zero, pois um sinal elétrico não tem o seu valor modificado num tempo nulo, mas ele leva, de fato, alguns ns para ir de um valor a outro. No caso da figura, uma transição deve durar em torno de 10 ns. O início do ciclo T1 é marcado pela subida do sinal do relógio. Na primeira parte de T1, a CPU coloca no barramento o endereço da palavra de memória a ser lida - o endereço, do mesmo modo que os dados, são representados (por razões de simplificação) por dois sinais complementares sobre uma linha única na qual os pontos de intersecção marcam os instantes de mudança de sinal.

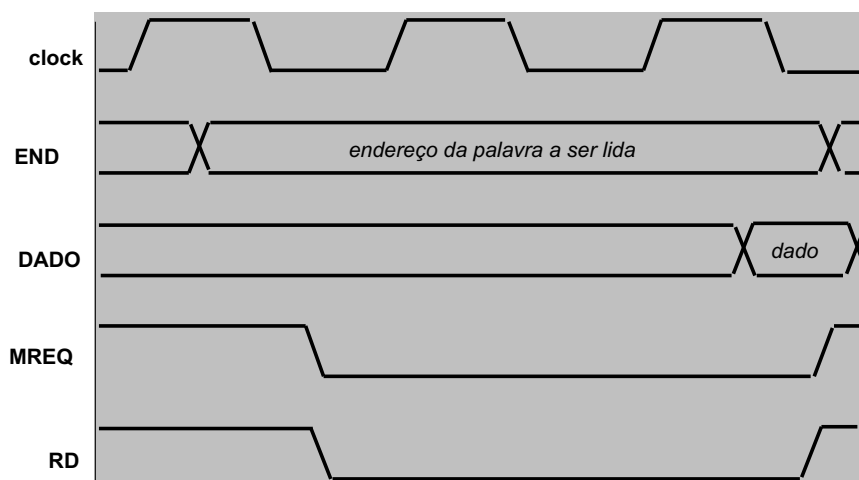


Figura 6 - Diagrama de tempo de um barramento síncrono.

As regiões em cinza que precede (ou sucede) os pontos de intersecção significam que os valores e endereços (ou dados) são irrelevantes naqueles intervalos de tempo.

Quando os sinais de endereço são estabilizados em seus novos valores, os sinais  $\overline{\text{MREQ}}$  e  $\overline{\text{RD}}$  são ativados. O primeiro sinal indica que se trata de um acesso à memória e o segundo define uma operação de leitura. Nada mais acontece no barramento durante o ciclo T2, de modo a permitir à memória a localização da palavra a ser lida e o seu posicionamento sobre o barramento de dados.

Na descida do relógio, em T3, a CPU vai capturar o dado sobre o barramento e armazená-lo num de seus registros internos. Terminada a operação, a CPU vai finalmente desativar os sinais  $\overline{\text{MREQ}}$  e  $\overline{\text{RD}}$ .

Esta operação considera que a memória é rápida o suficiente para decodificar o endereço, localizar o dado e posicioná-lo no barramento de dados num tempo compreendido entre a subida do ciclo T2 e a descida do ciclo T3.

Se isto não for possível, o circuito de memória deverá ativar, antes da descida em T2, um sinal (  $\overline{\text{WAIT}}$  ) que deverá indicar à CPU que a operação não pôde se realizar. Isto vai conduzir a uma introdução de novos ciclos de espera até que a memória desative o sinal  $\overline{\text{WAIT}}$ . Os barramentos síncronos, apesar de apresentarem um comportamento relativamente simples, apresentam alguns inconvenientes, particularmente ligado ao fato que toda operação deve se realizar num número inteiro de ciclos do barramento. Esta não é, evidentemente a melhor política, quando as necessidades em termos de desempenho são levadas em conta. Uma operação que poderia ser realizada em 3,1 ciclos de barramento, não terminará antes de 4 ciclos.

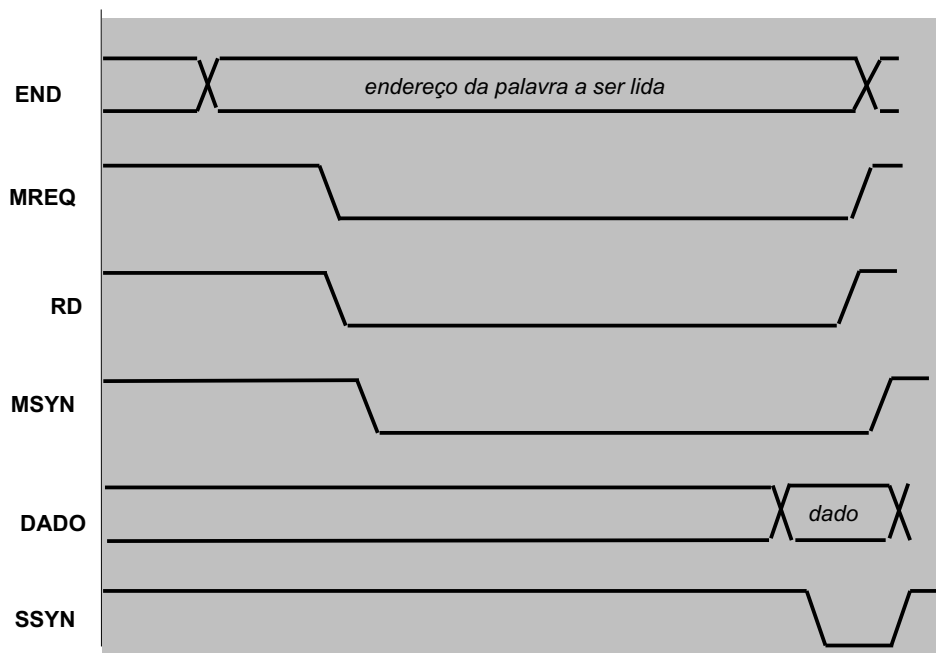
Outro inconveniente ocorre no caso da introdução de novos componentes (memória, por exemplo) mais rápidos. Considerando que os componentes antigos continuam conectados ao barramento, a frequência do clock não poderia ser aumentada e, logo, os novos componentes estariam sendo subutilizados em velocidade.

#### **4.2. Barramentos assíncronos**

Uma maneira de satisfazer de maneira mais adequada a conexão de componentes heterogêneos num barramento é a adoção de um barramento assíncrono, cujo ciclo de barramento é ilustrado pela Figura 7.

Estes barramentos não são comandados por um relógio piloto. Um ciclo do barramento será tão longo quanto necessário para assegurar a troca de dados e não será necessariamente o mesmo para todo o par mestre escravo existente.





**Figura 7** - Diagrama de tempo de um barramento assíncrono.

No caso deste tipo de barramento, a ativação das diferentes ações não se dá como mostrado na figura anterior, mas de fato através de um outro sinal  $\overline{\text{MSYN}}$  (sincronização mestra) que é ativado no momento da solicitação do circuito escravo.

Quando a operação é efetiva, o circuito escravo vai ativar um sinal  $\overline{\text{SSYN}}$  (sincronização escravo) que indica ao mestre a disponibilidade do dado sobre o barramento. Em seguida, o circuito mestre vai armazenar os dados presentes no barramento e desativar os sinais  $\overline{\text{MREQ}}$ ,  $\overline{\text{RD}}$  e  $\overline{\text{MSYN}}$ . Quando o circuito escravo vê a desativação de  $\overline{\text{MSYN}}$ , ele desativa também o sinal  $\overline{\text{SSYN}}$ .