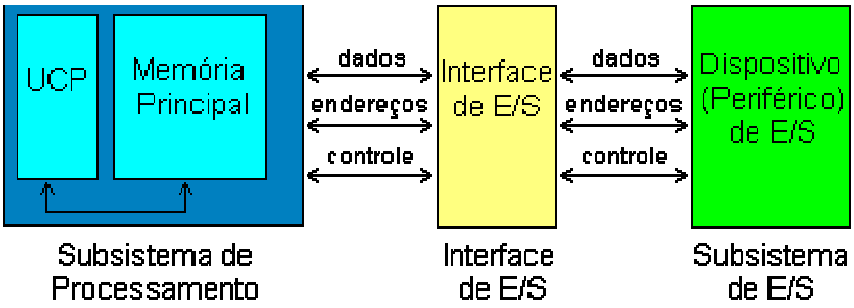
Entradas e Saídas

O usuário se comunica com o núcleo do computador (composto por UCP e memória principal) através de dispositivos de entrada e saída (dispositivos de E/S ou *I/O devices*). Os dispositivos de entrada e saída tem como funções básicas:

- a comunicação do usuário com o computador
- a comunicação do computador com o meio ambiente (dispositivos externos a serem monitorados ou controlados)
- armazenamento (gravação) de dados.

As características que regem a comunicação de cada um dos dispositivos de E/S (entrada e saída) com o núcleo do computador (composto de UCP e memória principal) são muito diferentes entre si. Cada dispositivo de E/S se comunica com o núcleo de forma diferente do outro. Entre outras diferenças, os dispositivos de entrada e saída são muito mais lentos que o computador, característica essa que impõe restrições à comunicação, de vez que o computador precisaria esperar muito tempo pela resposta do dispositivo. Outra diferença fundamental diz respeito às características das ligações dos sinais dos dispositivos.
Os primeiros computadores, especialmente os de pequeno porte, eram muito lentos e os problemas de diferença de velocidade eram resolvidos sem dificuldade e não representavam problema importante. Dessa forma, a ligação dos dispositivos de E/S era feita através de circuitos simples (as *interfaces*) que apenas resolviam os aspectos de compatibilização de sinais elétricos entre os dispositivos de E/S e a UCP. Os aspectos relativos as diferenças de velocidade (especialmente tempo de acesso e *throughput*) eram resolvidas por programa (isto é, por *software*).
Entre esses componentes, trafegam informações relativas a dados, endereços e controle.



Tipos de Dispositivos

Os dispositivos de ENTRADA são:
teclado, *mouses*, *scanners*, leitoras óticas, leitoras de cartões magnéticos, câmeras de vídeo, microfones, sensores, transdutores, etc ...
As funções desses dispositivos são coletar informações e introduzir as informações na máquina, converter informações do homem para a máquina e vice-versa, e recuperar informações dos dispositivos de armazenamento.

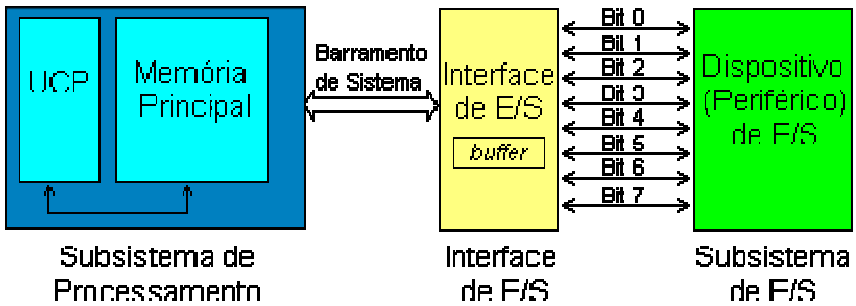
Os dispositivos de SAÍDA são:
impressoras, monitores de vídeo, *plotters*, atuadores, chaves, etc ...
As funções desses dispositivos são exibir ou imprimir os resultados do processamento, ou ainda controlar dispositivos externos.
A UCP não se comunica diretamente com cada dispositivo de E/S e sim com "*interfaces*", de forma a compatibilizar as diferentes características. O processo de comunicação ("protocolo") é feito através de transferência de informações de controle, endereços e dados propriamente ditos. Inicialmente, a UCP interroga o dispositivo, enviando o endereço do dispositivo e um sinal dizendo se quer mandar ou receber dados através da *interface*. O periférico, reconhecendo seu endereço, responde quando está pronto para receber (ou enviar) os dados. A UCP então transfere (ou recebe) os dados através da interface, e o dispositivo responde confirmando que recebeu (ou transferiu) os dados (*acknowledge* ou *ACK*) ou que não recebeu os dados, neste caso solicitando retransmissão (*not-acknowledge* ou *NAK*).

As interfaces de entrada e saída são conhecidas por diversos nomes, dependendo do fabricante: Interface de E/S = Adaptador de Periférico, Controladora de E/S, Processador de Periférico, Canal de E/S
A compatibilização de velocidades é feita geralmente por programa, usando memórias temporárias na *interface* chamadas *"buffers"* que armazenam as informações conforme vão chegando da UCP e as libera para o dispositivo à medida que este as pode receber.

Formas de Comunicação - Comunicação em Paralelo

De uma forma geral, a comunicação entre o núcleo do computador e os dispositivos de E/S poderia ser classificada em dois grupos: comunicação paralela ou serial. Vamos a seguir analisar as características desses grupos.

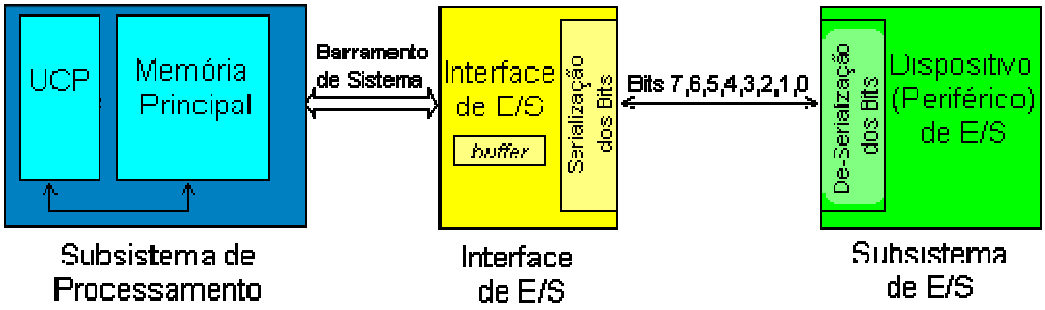
COMUNICAÇÃO EM PARALELO
Na comunicação em paralelo, grupos de bits são transferidos simultaneamente (em geral, byte a byte) através de diversas linhas condutoras dos sinais. Desta forma, como vários bits são transmitidos simultaneamente a cada ciclo, a taxa de transferência de dados ("*throughput*") é alta.



No entanto, o processo de transferência em paralelo envolve um controle sofisticado e é razoavelmente complexo, o que o torna mais caro. Um dos problemas importantes diz respeito à propagação dos sinais no meio físico, isto é, no cabo de conexão entre o dispositivo e a *interface*. Essa propagação deve se fazer de modo que os sinais (os bits) correspondentes a cada *byte* cheguem simultaneamente à extremidade oposta do cabo, onde então serão re-agrupados em *bytes*. Como os condutores que compõem o cabo usualmente terão pequenas diferenças físicas, a velocidade de propagação dos sinais digitais nos condutores poderá ser ligeiramente diferente nos diversos fios. Dependendo do comprimento do cabo, pode ocorrer que um determinado fio conduza sinais mais rápido (ou mais lento) que os demais fios e que desta forma um determinado bit *x* em cada *byte* se propague mais rápido e chegue à extremidade do cabo antes que os outros *n-1* bits do *byte*. Este fenômeno é chamado *skew,* e as conseqüências são catastróficas: os bits *x* chegariam fora de ordem (os *bytes* chegariam embaralhados) e a informação ficaria irrecuperável. Em decorrência desse problema, há limites para o comprimento do cabo que interliga um dispositico ao computador, quando se usa o modo paralelo.
As restrições citadas contribuem para que a utilização da comunicação em paralelo se limite a aplicações que demandem altas taxas de trasferência, normalmente associadas a dispositivos mais velozes tais como unidades de disco, ou que demandem altas taxas de transferência, como CD-ROM, DVD, ou mesmo impressoras, e que se situem muito próximo do núcleo do computador. Em geral, o comprimento dos cabos paralelos é limitado a até um máximo de 1,5 metro.

Formas de Comunicação - Comunicação Serial

Na comunicação serial, os bits são transferidos um a um, através de um único par condutor. Os *bytes* a serem transmitidos são serializados, isto é, são "desmontados" bit a bit, e são individualmente transmitidos, um a um. Na outra extremidade do condutor, os bits são contados e quando formam 8 bits, são remontados, reconstituindo os *bytes* originais. Nesse modo, o controle é comparativamente muito mais simples que no modo paralelo e é de implementação mais barata. Como todos os bits são transferidos pelo mesmo meio físico (mesmo par de fios), as eventuais irregularidades afetam todos os bits igualmente. Portanto, a transmissão serial não é afetada por irregularidades do meio de transmissão e não há *skew*. No entanto, a transmissão serial é intrinsecamente mais lenta (de vez que apenas um bit é transmitido de cada vez).



Como os bits são transmitidos seqüencialmente um a um, sua utilização é normalmente indicada apenas para periféricos mais lentos, como por exemplo teclado, *mouse*, etc. ou quando o problema da distância for mandatório, como nas comunicações a distâncias médias (tal como em redes locais) ou longas (comunicações via linha telefônica usando *modems*).

Obs.: Comparativamente, a transmissão serial tem recebido aperfeiçoamentos importantes (seja de protocolo, de *interface* e de meio de transmissão) que vem permitindo o aumento da velocidade de transmissão por um único par de fios, cabo coaxial ou de fibra ótica. Como o aumento da velocidade de transmissão em interfaces paralelas ocasiona mais *skew*, a tendência tem sido no sentido do aperfeiçoamento das interfaces seriais que hoje permitem taxas de transferência muito altas com relativamente poucas restrições de distância. Em microcomputadores, a *interface USB - Universal Serial Bus* permite hoje ligar até 128 dispositivos a taxas muito altas (centenas de kbps).

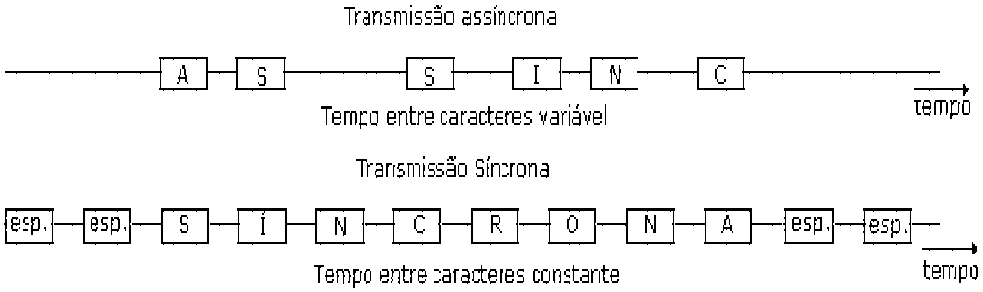
TABELA COMPARATIVA

Característica	**Paralelo**	**Serial**
Custo	maior	menor
Distância	curta	sem limite
Throughput	alto	baixo

TRANSMISSÃO SÍNCRONA E ASSÍNCRONA

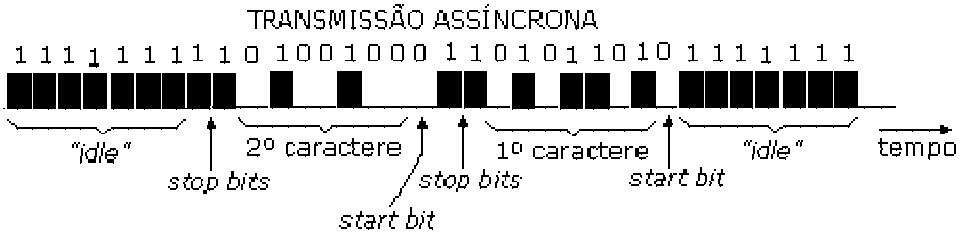
A transmissão de caracteres através de uma linha de comunicação pode ser feita por dois diferentes métodos: transmissão síncrona e assíncrona.

Na **transmissão síncrona**, o intervalo de tempo entre dois caracteres subseqüentes é fixo. Nesse método, os dois dispositivos - transmissor e receptor - são sincronizados, pois existe uma relação direta entre tempo e os caracteres transferidos. Quando não há caracteres a serem transferidos, o transmissor continua enviando caracteres especiais de forma que o intervalo de tempo entre caracteres se mantém constante e o receptor mantém-se sincronizado. No início de uma transmissão síncrona, os relógios dos dispositivos transmissor e receptor são sincronizados através de um *string* de sincronização e então mantém-se sincronizados por longos períodos de tempo (dependendo da estabilidade dos relógios), podendo transmistir dezenas de milhares de bits antes de terem necessidade de re-sincronizar.



Já na **transmissão assíncrona**, o intervalo de tempo entre os caracteres não é fixo. Podemos exemplificar com um digitador operando um terminal, não havendo um fluxo homogêneo de caracteres a serem transmitidos. Como o fluxo de caracteres não é homogêneo, não haveria como distingüir a ausência de bits sendo transmitidos de um eventual fluxo de bits zero e o receptor nunca saberia quando virá o próximo caractere, e portanto não teria como identificar o que seria o primeiro bit do caractere. Para resolver esses problemas de transmissão assíncrona, foi padronizado que na ausência de caracteres a serem transmitidos o transmissor mantém a linha sempre no estado 1 (isto é, transmite ininterruptamente bits 1, o que distingüe também de linha interrompida). Quando for transmitir um caractere, para permitir que o receptor reconheça o início do caractere, o transmissor insere um bit de partida (*start bit*) antes de cada caractere. Convenciona-se que esse *start bit* será um bit zero,

interrompendo assim a seqüência de bits 1 que caracteriza a linha livre (*idle*). Para maior segurança, ao final de cada caractere o transmissor insere um (ou dois, dependendo do padrão adotado) bits de parada (*stop bits*), convencionando-se serem bits 1 para distingüí-los dos bits de partida. Os bits de informação são transmitidos em intervalos de tempo uniformes entre o *start bit* e o(s) *stop bit(s)*. Portanto, transmissor e receptor somente estarão sincronizados durante o intervalo de tempo entre os bits de *start* e *stop*. A transmissão assíncrona também é conhecida como *"start-stop"*.



A taxa de eficiência de uma transmissão de dados é medida como a relação de número de bits úteis dividido pelo total de bits transmitidos. No método assíncrono, a eficiência é menor que a no método síncrono, uma vez que há necessidade de inserir os bits de partida e parada, de forma que a cada caractere são inseridos de 2 a 3 bits que não contém informação.

TRANSMISSÃO SIMPLEX, HALF-DUPLEX E FULL-DUPLEX

Uma comunicação é dita *simplex* quando permite comunicação apenas em um único sentido, tendo em uma extremidade um dispositivo apenas transmissor (*transmitter*) e do outro um dispositivo apenas receptor (*receiver*). Não há possibilidade do dispositivo receptor enviar dados ou mesmo sinalizar se os dados foram recebidos corretamente. Transmissões de rádio e televisão são exemplos de transmissão *simplex*.

Uma comunicação é dita *half-duplex* (também chamada *semi-duplex*) quando existem em ambas as extremidades dispositivos que podem transmitir e receber dados, porém não simultaneamente. Durante uma transmissão *half-duplex*, em determinado instante um dispositivo A será transmissor e o outro B será receptor, em outro instante os papéis podem se inverter. Por exemplo, o dispositivo A poderia transmitir dados que B receberia; em seguida, o sentido da trasmissão seria invertido e B transmitiria para A a informação se os dados foram corretamente recebidos ou se foram detectados erros de transmissão. A operação de troca de sentido de transmissão entre os dispositivos é chamada de *turn-around* e o tempo necessário para os dispositivos chavearem entre as funções de transmissor e receptor é chamado de *turn-around time*.

Uma transmissão é dita *full-duplex* (também chamada apenas *duplex*) quando dados podem ser transmitidos e recebidos simultaneamente em ambos os sentidos. Poderíamos entender uma linha *full-duplex* como funcionalmente equivalente a duas linhas *simplex*, uma em cada direção. Como as transmissões podem ser simultaneas em ambos os sentidos e não existe perda de tempo com *turn-around*, uma linha *full-duplex* pode transmitir mais informações por unidade de tempo (maior *throughput*) que uma linha *half-duplex,* considerando-se a mesma taxa de transmissão de dados.