Capítulo 5

Entrada e Saída

5.1 Introdução

Entende-se como entrada/saída de um sistema computacional a sua interação com o "mundo exterior" através de dispositivos denominados **periféricos** ou com outros sistemas computacionais via **rede**. Temos então por um lado os sianis dos barramentos (de dados, de endereços e de controle) compatíveis com a UCP, e por outro lado, um dos seguintes conjuntos de sinais:

1. sinais (de dados e de controle) dos periféricos que variam com a sua tecnologia e a sua aplicação, ou

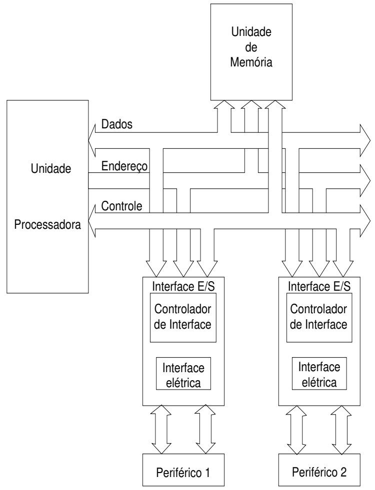
2. sinais de barramento do outro sistema computacional que são dependentes da sua arquitetura.

A compatibilização das caracterísitcas elétricas e temporais entre os dois lados é feita através de circuitos chamados de **circuitos de interface** (de entrada/saída) que consiste essencialmente de dois módulos funcionais:

Interface elétrica: (Fig. 8.5 do livro-texto, p. 632) responsável pela **compatibilização dos níveis elétricos** dos sinais. Compreende os circuitos de *debounce*, conversores de nível, conversores A/D e D/A (Figs. 8.6–8.7 do livro-texto, pp. 633–634) e amplificadores de corrente.

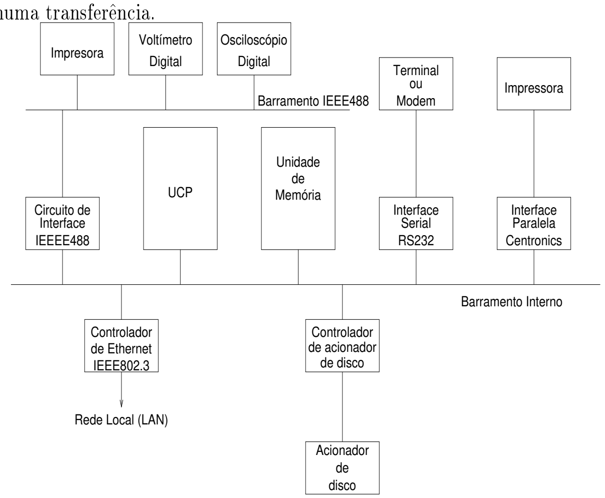
Circuitos controladores de interface: responsável pela compatibilização temporal. Compreende os retentores (*buffers*) para **compatibilizar a taxa de transferência**, os circuitos de **serialização** (Fig. 9.2 do

livro-texto) e os circuitos que coordenam as transferências respeitando o **protocolo** suportado por cada lado. Para aumentar a confiabilidade na transferência é comum ainda incluir circuitos que codifiquem os dados a serem transmitidos em códigos detectores e/ou corretores de erro (seção 7.1 do livro-texto, pp. 517 – 529).



Os fabricantes de microprocessadores procuram oferecer ao mercado circuitos controladores de interface compatíveis com os sinais de controle dos seus processadores, de forma a facilitar a ligação de periféricos. Desta forma são definidas as chamadas **famílias de componentes**. Por exemplo, dentro da Motorola existem os componentes da família 6800 e os da família 68000 e dentro da Intel podemos citar as famílias 8085/88 e 386/486. Em geral, o projeto de um circuito de interface utilizando componentes de uma mesma família do processador não traz dificuldades para o projetista. Entretanto, muitas vezes, uma família de componentes não dispõe de um controlador de interface com a caracterísitca desejada e que está presente num controlador de outra família. Assim sendo o projetista terá que desenvolver seu projeto ligando componentes de famílias diferentes.

O projeto da ligação da interface com o barramento está intimamente ligado com a organização da UCP e o projeto da ligação com um periférico varia de acordo com o tipo do periférico. A variedade nas arquiteturas de computadores e nos tipos de periféricos pode levar a um grande número de projetos de circuitos de interfaces se não houver nenhum esforço no sentido de padronizar certas classes de ligações entre computadores e periféricos. Um **padrão de interface** compreende um conjunto de especificações das características elétricas, temporais, funcionais e mecânicas de cada sinal envolvido numa transferência.



Observações:

- Os códigos mais utilizados nos sistemas computacionais são os códigos de paridade, onde um bit (de paridade) é incluído para indicar a possível ocorrência de erros em cada palavra. Os códigos apresentados na seção 7.1 do livro-texto são mais complexos. Eles não só detectam como também tentam corrigir erros detectados.
- É importante ficar claro que um padrão define os conectores e as linhas utilizadas em uma ligação. A implementação de um padrão em um circuito é de responsabilidade dos projetistas.

Neste capítulo serão focados os seguintes pontos concernentes ao projeto de uma interface de E/S com uso da família de compoenentes 68000 e 6800:

- como a arquitetura de um processador está relacionada com o processamento de E/S?
- como os periféricos podem ser endereçados pela UCP?
- quais são os sinais do barramento de um sistema que controlam as transferências?
- quais são as classes lógicas de periféricos?
- quais são os protocolos de comunicação suportados pelos dispositivos?
- quais são os sinais de controle providos pelos periféricos para transferências?
- como são os circuitos de interface?
- quais são os padrões de interface mais conhecidos?

5.2 Processamento de E/S

Para fins de programação, a lógica dos sinais de controle e de dados dos periféricos são disponíveis nos registradores dos circuitos de interface. Estes registradores devem ser endereçáveis pela unidade processadora. Dependendo de como são definidos os endereços destes registradores para a operação de E/S podemos distinguir dois tipos de arquitetura:

1. E/S isolada: quando os periféricos são conectados num barramento separado do barramento que liga a Unidade de Memória com a UCP, devendo a UCP dispor de instruções específicas de leitura e escrita em periféricos.
2. E/S mapeada em memória: quando os periféricos conectados a UCP utilizam os mesmos sinais de controle e as mesmas instruções necessárias à conexão da Unidade de Memória a UCP.

Alguns processadores, como os da família 6800 e 68000 da Motorola, só suportam E/S mapeada em memória, pois eles não dispõem de sianis de controle nem instruções especiais para a programação de E/S. Outros como os da Intel, possuem recursos para referência direta a dispositivos de E/S e oferecem as duas opções de arquitetura. Neste caso, a decisão por qual das duas arquiteturas vai depender da aplicação desejada para o sistema, uma vez que ambas apresentam vantagens e desvantagens, como já vimos no capítulo 2.

Existem basicamente três formas para iniciar uma transferência de dados entre um sistema computacional e seus periféricos:

E/S controlada por programa , que pode ser por requisição através de um laço de *polling* (E/S programada condicional) e por amostragem (E/S programada incondicional). Como o instante de transferência não é previamente conhecido pelo programador, é necessário introduzir testes do estado do periférico (pronto ou ocupado) antes de executar a instrução de transferência.

E/S por interrupção , ou seja a transferência se dá em resposta a uma interrupção gerada pelos periféricos na ocorrência de eventos como pronto para enviar novos dados, pronto para receber novos dados ou falha no periférico.

E/S por acesso direto à memória (DMA) , quando a transferência se dá diretamente entre o periférico e a Unidade de Memória sem a intervenção da UCP, com o controlador de DMA (DMAC) no papel de mestre de barramento (Fig. 8.14 do livro-texto, p. 642). O acesso pode ser transparente (DMA transparente), "por roubo" (uma fatia do ciclo de barramento é utilizada por DMA) e por "rajada" (alguns ciclos de barramento são utilizados por DMA). O funcionamento básico do acesso DMA pode ser entendido com o auxílio da Fig. 8.14 do livro-texto, onde é suposto que somente existem dois potenciais mestres de barramento – UCP e DMAC:

1. o circuito de interface/periférico solicita o serviço do DMAC com a ativação do sinal REQ*;
2. o DMAC solicita o barramento com a ativação do sinal *DMA request*;

3. ao terminar a instrução corrente, a UCP reconhece a solicitação e coloca os seus barramentos de dados, endereços e de controle em impedância alta (*bus_switch_1*). Em seguida, ela envia o sinal *DMA grant* para o DMAC para avisar que os barramentos foram liberados; e
4. DMAC informa ao circuito de interface/periférico que ele tem o controle do barramento com o sinal ACK*, conecta-se com os barramentos (*bus_switch_2* e *bus_switch_3*) e iniciam-se os ciclos de acesso direto à memória. O DMAC passa a gerar em cada ciclo o endereço da posição da Unidade de Memória a ser acessada e os sinais de controle (R/W*, *strobe* de endereços, *strobe* de dados e/ou sinal de reconhecimento).

Nas pp. 642–648 do livro-texto é descrito o controlador de DMA 68430 da família 68000. Note que um DMAC é provido de um conjunto de registradores (de endereço inicial do bloco de dados, de tamanho do bloco, etc.) cujo conteúdo deve ser previamente configurado (programado) para garantir uma correta operação.

A transferência propriamente dita pode ocorrer através de uma das duas interfaces:

1. interface síncrona: caracteriza-se pela existência de uma linha no barramento de controle que carrega o sinal de relógio (*clock*) do sistema. Este sinal é utilizado como sinal de referência para todos os outros sinais de controle. Fig. 8.8 (p. 635) e 8.10 (p. 637) do livro-texto mostram, respectivamente, o diagrama de tempo e o protocolo de transferência entre um periférico da família 6800 com MC68000.
2. interface assíncrona: o sequenciamento das operações em **uma** transferência é estabelecido por um protocolo pré-definido, que pode ser
   - por sinal de *strobe*: uma linha ou
   - por *handshaking*: duas linhas (Fig. 4.11 do livro-texto, p. 219).

O instante da ativação destes sinais não é previsível, podendo demandar a inserção de estados de espera num ciclo de acesso para garantir a finalização bem sucedida do ciclo.

Observações:

- Os processadores da família suportam interfaces síncronas e assíncronas para facilitar a sua ligação com os periféricos da família 6800 que só operam no modo síncrono.
- Em muitos circuitos de interface, como controladores de disco e controladores/coprocessadores da Ethernet (MC68EN302/Intel82596DX), são encontrados embutidos os DMACs para transferência direta de dados entre o periférico e a Unidade de Memória.
- Quando o DMAC assume o controle dos barramentos, o circuito de interface/periférico comunica com ele através de dois sinais de controle para indicar, respectivamente, que os dados são válidos para transferência e que finalizou um ciclo de acesso.
- Um DMAC pode suportar mais de um canal de acesso direto à memória.
- Na Fig. 8.14 do livro-texto, p. 642, é mostrada uma ligação entre o DMAC e o pino de interrupção IRQ* da UCP. Isso significa que o DMAC, similar a outros periféricos. pode solicitar a atenção da UCP na ocorrência de determinados eventos.

Exercícios de Revisão

1. Resolver os exercícios 1, 2 e 3 do capítulo 8 do livro-texto.
2. Quais são as duas básicas arquiteturas de E/S?
3. Quais são as formas de iniciar o processamento de uma entrada/saída?
4. O que é uma interface síncrona?
5. O que é uma interface assíncrona?

5.3 Periféricos

Funcionalmente os periféricos podem ser divididos em duas categorias:

1. aqueles que interagem de fato com o mundo exterior, realizando operações de entrada e saída dos dados; e
2. aqueles que constituem a memória secundária de um sistema computacional.

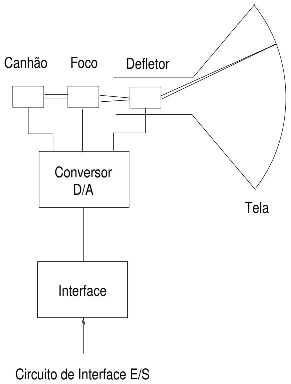
A primeira categoria inclui o teclado, impressoras, *mouse*, vídeo e a segunda, os disquetes, fitas magnéticas e disco rígido. Nesta seção apresentaremos alguns dispositivos comumente utilizados como periféricos de um sistema computacional.

5.3.1 Terminais

Um terminal consiste de um dispositivo de entrada, o teclado, e de um dispositivo de saída, o monitor. Ele é o dispositivo de E/S mais conhecido para um sistema computacional interagir com os usuários.

A tela de um monitor é coberta por material fosforecente (fósforo) que ao ser energizado emite luz. Dentro do vídeo existe ainda um canhão de elétrons que emite feixe de elétrons na direção da tela para excitar seletivamente os pontos da tela. Quanto mais preciso for o mecanismo de focalização do feixe, melhor será a definição do ponto. Dependendo da persistência do material fosforescente utilizado o ponto pode permanecer acesso até alguns segundos. O material fosforescente utilizada na maioria dos monitores tem uma persistência de 10 a 60 microsegundos. Isso significa que se a tela não for **renovada** periodicamente, os pontos começam a apagar, podendo gerar o efeito conhecido como **cintilação** (*flicker*).

Existem basicamente duas tecnologias de monitores: vetoriais e *raster*. Os monitores *raster* são os mais utilizados nos sistemas computacionais, uma vez que o tempo de geração de uma tela independe da complexidade do conteúdo da tela. No caso de monitores monocromáticos (preto e branco), os sinais de controle se reduzem aos sinais de controle da varredura dos feixes e o sinal de liga/desliga do feixe para cada ponto.



O teclado, por sua vez, permite introduzir caracteres alfanuméricos num computador. Sempre que uma tecla for baixada, o código (ASCII, EBCDIC, etc.) do caracter correspondente é gerado automaticamente. Várias tecnologias são utilizadas para detectar quando uma tecla é pressionada. Entre elas podemos citar contatos mecânicos, variações na capacitância e variações no fluxo eletromagnético. Funcionalmente distinguem-se duas classes de teclados: os *coded* e os *unendecoded*. Estes últimos teclados identificam todas as teclas, quando duas ou mais forem pressionadas simultâneamente.

Como a velocidade de operação dos homens é comparativamente muito mais lenta do que a do computador, os bits do código de um caracter são enviados serialmente através de uma só linha. A linha de saída do teclado está normalmente no estado alto, mas, quando uma tecla é pressionada, é gerado um *start bit* num nível lógico baixo. Este bit é seguido pelo código do seu caracter, opcionalmente por um bit de paridade, e pelos *stop bits*. A cada um destes bits é atribuído uma janela de tempo de mesmo tamanho (Fig. 9.3 do livro-texto, p. 703). Esta forma de transmissão de dados é conhecida como **transmissão assíncrona** ou não-contínua, uma vez que o *start bit* pode ocorrer assincronamente, em qualquer instante de tempo. Note-se que nesta forma de transmissão um *stop bit* (nível 1) é, no mínimo, necessário entre o envio de dois caracteres, pois a transição 1 para 0 (*stop–start*) é utilizado pelo receptor para resincronização. Supondo que o período T do receptor difere ligeiramente do período do transmissor de δt, o número N de bits (incluindo o *start* bit) que se pode transmitir numa rajada sem resincronização deve satisfazer a seguinte relação (Fig. 9.4 do livro-texto, p.

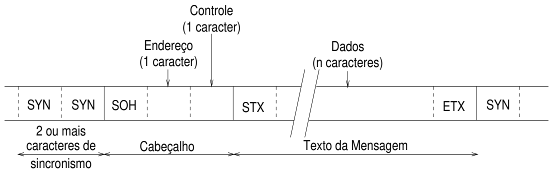
704)

$$\frac{\delta t}{T} < \frac{1}{2N+1}$$

Quando os terminais forem operados a uma certa distância de um computador, o sistema telefônico é utilizado para fornecer o enlace de comunicações necessário. Neste caso, dispositivos especiais são necessários para traduzir os níveis lógicos produzidos pelo terminal em sinais aceitáveis pela linha telefônica. Quando os sinais da rede telefônica forem analógicos, um modulador-demodulador, ou simplesmente MODEM, é utilizado para modular os sinais digitais em analógicos no transmissor e decodificá-los de volta em sinais digitais no receptor (Figs. 9.2 e 9.21 do livro-texto, p. 702 e 744). O formato assíncrono *start-stop* é também utilizado neste tipo de comunicação. Porém, uma grande percentagem de "faixa" de bits transmitidos é disperdiçada para sincronização, uma vez que a relação entre os bits efetivos de dados e os bits de "sincronização" (*start* e *stop* bits) é relativamente alta.

Uma alternativa á a **transmissão síncrona** ou contínua, quando são alocadas aos bits janelas fixas de tempo e aos caracteres posições fixas na sequência de bits. Nesta transmissão aumenta-se a eficiência do uso da faixa para transmissão de dados efetivos (já que os *start* e *stop* não são necessários), a custo da inclusão de um sistema adicional para monitorar constantemente o sincronismo entre o transmissor e o receptor. Quando o receptor não tem acesso aos sinais de relógio do transmissor, técnicas de codificação de sinais de relógio nos dados transmitidos são utilizados, como a codificação de fase (Fig. 9.17 do livro-texto, p. 740).

Na trasmissão síncrona, a organização dos caracteres dentro de uma sequência de bits é estabelecida pelos **protocolos** de comunicação, de forma a assegurar que o receptor consiga extrair corretamente os endereços e os dados da sequência de bits recebida. No caso de comunicação entre os terminais alfanuméricos e o computador, o protocolo **orientado a caracter** é o mais utilizado. Neste protocolo a transmissão de informações (de controle, endereço e dados) é feita na base de códigos ASCII (caracteres de 7 bits ou 8 bits, se o bit de paridade for utilizado). Uma sequência de bits de informação enviada é delimitada pelos caracteres SYN ($16 ou 0010110). A mensagem propriamente dita consiste ainda de duas partes: o cabeçalho precedido pelo caracter SOH ($01 ou 0000001) que contém a informação de endereçamento (do periférico) e de controle e o texto de mensagem delimitado pelos caracteres STX ($03 ou 0000010) e ETX ($04 ou 0000011) (p. 741 do livro-texto).



Quando não se trata de informações codificadas na base de caracteres, a combinação dos bits é totalmente aleatória podendo levar a interpretação errônea de uma sequência de bits de dados como caracteres de controle. Para diferenciar os caracteres de controle dos caracteres de dados, ambos representados pelo mesmo padrão binário, o caracter DLE ($10 ou 0010000) é inserido antes de cada caracter de controle para garantir a **transparência de dados**. Caso exista uma sequência de bits do padrão 0010000 no texto da mensagem, o transmissor insere antes dele outro caracter DLE para avisar o receptor de que o padrão que se segue deve ser interpretado como caracter de dados. É fácil perceber que esta solução não é eficiente para a transmissão de dados puramente binários, além de ser bastante sujeita a erros. Uma segunda alternativa seria utilizar o **protocolo orientado a bit** (pp. 741–744 do livro-texto).

O protocolo orientado a bit suporta a transferência de uma sequência de bits sem a restrição de ser em um número múltiplo da quantidade de bits de um caracter. Existem várias propostas de protocolo orientado a bits, como SDLC (*synchronous data link control*) usada pela IBM, HDLC (*high-level data link control* adotada pela ISO e ADCCP (*advanced data communication control procedure*) recomendada pelo ANSI. Nestas propostas as informações são basicamente organizadas num formato específico denominado **quadro** (*frame*), delimitado pelo padrão de bits conhecido como *flag* (0111110). Um quadro contém, além do campo de mensagem, um campo de endereço (do periférico), um campo de dados de controle e um campo de dados para deteção de erros FCS (Fig. 9.19 do lviro-texto, p. 743).

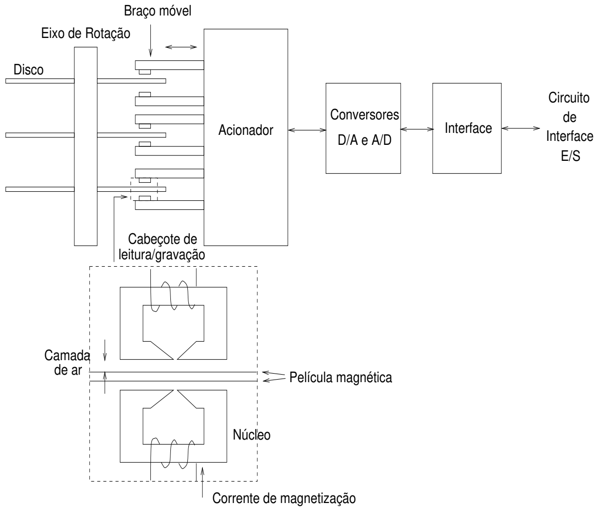
Observe que neste protocolo persiste ainda o problema de diferenciar se o padrão 01111110 é um flag delimitador de um quadro ou são meros bits do texto da mensagem. Isso pode ser evitado com a inserção de bit (*bit-stuffing*) quando o padrão 011111 é detectado pelo transmissor e no receptor é, então, necessário eliminar o bit 0 inserido para recuperar a informação original (Fig. 9.18 do livro-texto, p. 742).

A diversidade na forma de comunicação entre os terminais e os compu-

tadores torna a sua ligação uma tarefa não trivial, se os fabricantes não fornecerem *drivers* de forma que estes detalhes fiquem transparentes para os programadores.

5.3.2 Sistemas de Discos Magnéticos

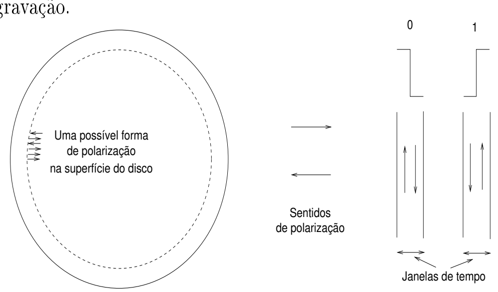
Um sistema de discos magnéticos consiste de um ou mais discos empilhados. Uma fina camada magnética é depositada sobre a superfície de cada disco, usualmente nos dois lados. Os discos são montados sobre um acionador (*drive*) que gira em torno de um eixo, de forma que as superfícies magnetizantes de cada disco movam bem próximas dos cabeçotes de leitura/gravação. A velocidade de rotação do disco é constante. Cada cabeçote é constituído por um núcleo de ferro e uma bobina.



Dados binários podem ser **gravados** na fina película magnética sobre a superfície do disco ao aplicarmos pulsos de corrente de excitação na bobina. Isso cria um fluxo eletromagnético no núcleo capaz de polarizar a área do disco imediatamente debaixo do cabeçote. A direção de polarização nesta área é paralela à direção do fluxo magnético induzido no núcleo.

O mesmo cabeçote pode ser utilizado para **ler** os dados armazenados.

Neste caso, o movimento da fina película magnetizada em relação ao núcleo induz um fluxo magnético no núcleo. Este, por sua vez, induz uma tensão na bobina. Esta tensão é detectada pelo circuito de controle. Dependendo da polaridade da tensão é possível determinar o sentido de polarização na área sob o cabeçote. Note que uma tensão só pode ser induzida na bobina se houver variação no campo magnético. Portanto, se os estados binários 0 e 1 corresponderem, respectivamente, a dois sentidos de polarização, poderemos somente detectar tensões nas transições de estados. Em outras palavras, uma sequência longa de 1 ou 0 só pode ser detctada no início da sequência. Para saber quantos bits de 0 ou de 1 que compõem a sequência, precisaremos de **sincronizar** os bits da sequência, isto é, conhecer a janela de tempo de cada bit na gravação.



Duas alternativas são muito utilizadas para a **sincronização de bit**:

- reservar uma ou mais trilhas para os sinais de relógio; ou
- codificar as informações de tempo nos dados, com uso do código de fase ou de Manchester (Fig. 9.17 do livro-texto, p. 740).

A segunda alternativa tem a vantagem de dispensar uma trilha dedicada para sinais de relógio utilizado na gravação. Porém, cada bit requer um espaço maior, o suficiente para acomodar dois sentidos de polarização.

É interessante ressaltar aqui que a distância entre o cabeçote e a superfície é fundamental tanto na confiabilidade das operações de leitura/gravação como na densidade de armazenamento. Um sistema é incorporado ao acionador para evitar grandes variações.

A superfície de um disco é dividido logicamente em **trilhas** concêntricas e cada trilha em setores. O conjunto de trilhas de mesmo raio de todos os discos define um **cilindro**. Os dados são armazenados serialmente, bit-a-bit,

em uma trilha. Usualmente a unidade endereçável é um setor. O endereço é composto por número da superfície, o número da trilha e o número do setor.

Os cabeçotes de leitura/gravação podem ser fixos ou móveis. No primeiro caso cada trilha é provida de um cabeçote e no segundo caso, só existe um cabeçote para cada superfície que consegue deslocar radialmente sobre a superfície. Assim sendo, ao receber um endereço o acionador necessita, além de um tempo de **latência de giro** (*latency time*) para chegar no início do setor endereçado, um **tempo de posicionamento** para posicionar o cabeçote sobre a trilha que contém o setor (*seek time*) e um **tempo de chaveamento de cabeça** para selecionar o disco a ser acessado. Só então, os dados são transferidos. A soma dos três tempos é conhecido como **tempo de acesso do disco**. O menor tempo de acesso médio (possível) é 50% do tempo de um giro.

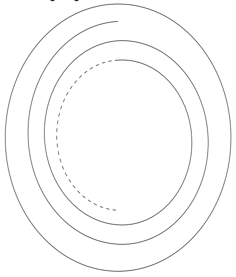
Duas classes de discos magnéticos são amplamente difundidas: discos rígidos (ou discos *Winchester*) e discos flexíveis. Os discos rígidos são caracterizados por terem a sua pilha de discos e os cabeçotes selados em um envoltório, evitando contaminações. Com isso, eles apresentam maior densidade de armazenamento e maior integridade.

Num sistema computacional os acionadores de discos magnéticos são controlados pelos **controladores de disco**.

5.3.3 Discos Óticos

Os discos óticos, ou discos compactos CDs, foram originalmente desenvolvidos para sistemas digitais de áudio. Estes discos eram capazes de armazenar em um lado informações de áudio de duração maior que 60 minutos, o que impulsionou o desenvolvimento da tecnologia de armazenamento por princípio ótico, barateando o seu custo. Esta mesma tecnologia é utilizada para armazenamento de dados nos dispositvos óticos utilizados nos sistemas computacionais, a saber CD-ROMs (*Compact Disk Read-Only Memory*), WORM (*Write-Once Read-Many*) e disco ótico programável (*erasable optical disk*).

Os discos compactos são caracterizados por terem suas superfícies altamente reflexivas. Os dados são **gravados** através da "queima" ou não da película reflexiva pelo feixe de raio *laser* quando o disco rotaciona a uma velocidade angular ou linear constante. Para aumentar ainda mais a densidade de dados nas trilhas mais externas, a rotação em velocidade linear constante, mas angular variada, é a preferida e os dados são organizados em espiral. Com isso, um grande volume de dados (mais de 550 Mbytes) pode ser armazenado em um disco pequeno de 120 mm de diâmetro.



Os dados são **lidos** com base no princípio ótico. Ao incidir o raio *laser* sobre o disco, ele é refletido nos pontos intatos e "absorvido" nos pontos queimados. Um circuito adicional pode interpretar a intensidade do raio refletido em dados binários 1 ou 0.

A tecnologia de um *driver* de CD, altamente sofisticada, esconde os detalhes dos circuitos de acionamento do disco, do feixe de raio *laser* e do sistema ótico, disponibilizando normalmente para os programadores os dados em paralelo.

5.3.4 Impressoras

As impressoras são utilizadas para produzir impressões a partir dos códigos binários de um sistema computacional. Elas são usualmente classificadas em **impressoras por impacto** (eletromecânicas) e sem impacto (eletrostáticas, térmicas e óticas).

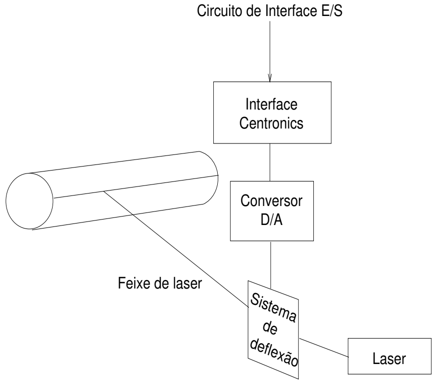
As impressoras originais, em sua maioria, eram máquinas de escrever elétricas convertidas. Se cada caracter for representado por 8 bits e a impressora estiver operando em linha, cada caracter recebido será decodificado para energizar o solenóide que aciona a tecla da máquina correspondente. Caracteres de controle são inclusos para controlar o espaçamento entre palavras, retorno do carro e outras operações necessárias à datilografia. A velocidade de tais impressoras é da ordem de 10 a 30 caracteres por segundo.

Mais tarde surgiram as impressoras matriciais em que um conjunto de 7 a 24 pinos numa matriz acionada por solenóide é usada para imprimir caracteres. Cada pino pode ser acionado individualmente por pequenos martelos, pressionando a fita contra o papel. Impressões coloridas podem ser obtidas com impressão de mesmo ponto com três cores diferentes, tipicamente, ciano,

magenta e amarelo. Nestas impressoras consegue-se atingir velocidades de até 330 caracteres por segundo.

Nas impresoras eletrostáticas, o papel é revestido com um material dielétrico não-condutor que requer cargas quando são aplicadas tensões usando cabeçotes de gravação. Estes cabeçotes "gravam" pontos no papel quando ele passa. Mais tarde, o papel passa através de um virador contendo material com partículas coloridas que transportam uma carga contrária à que foi gravada pelos cabeçotes e, em consequência, as partículas aderem às áreas carregadas formando caracteres impressos. A rasterização dos caracteres em pontos energizados sobre o papel pode ser feito por um circuito embutido na impressora ou pela unidade processadora a qual a impressora é conectada.

As impressoras a *laser* monocromáticas utilizam o feixe de *laser* para neutralizar em alguns pontos a carga positiva do "rolo de impressão" coberto por selênio. Como o pó do *toner* é carregado negativamente, ele é atraído somente às regiões do "rolo" com carga positiva. A impressão se resulta da transferência dos pós sobre o "rolo" em folhas de papel. Nas impressoras coloridas o processo é repetido para as três cores primárias. Como nas impressoras eletrostáticas, a rasterização pode ser feita pela impressora ou pela UCP.

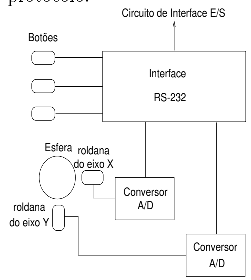


As impressoras de jato de tinta esguicham três correntes de tinta de cores primárias, tipicamente ciano, magenta e amarelo, de alta velocidade em direção ao papel. Esta corrente é defletida, geralmente ao atravessar um campo eletrostático tal como o utilizado nas impressoras a *laser*. Como as duas categorias anteriores, essas impressoras são dispositivos *raster*.

A grande variedade de impressoras disponíveis e o desenvolvimento contínuo de novas idéias tornam difícil a interpretação direta dos protocolos de transferência de dados adotados por cada fabricante. Felizmente, os fabricantes fornecem *drivers* (em *software*) para simplificar o uso (mais precisamente, a programação) das suas impressoras.

5.3.5 Mouses

Um *mouse* é um dispositivo capaz de captar o seu deslocamento relativo sobre uma superfície. Um *mouse* mecânico possui dois ou três botões e uma esfera de borracha, com alto coeficiente de atrito que, apoiado sobre uma superfície plana, faz contato com pelos menos duas roldanas. Estas roldanas funcionam como engrenagens, rolando sem deslizar: a primeira acompanha o movimento da esfera no eixo X e a segunda no eixo Y. Dois transdutores de posição angular convertem esses deslocamentos em dados digitais (deslocamento para direita, para esquerda, para cima e para baixo). O protocolo da transferência de dados referentes aos cliques dos botões e movimentos da esfera são dependentes dos fabricantes. Usualmente as condições de iniciar uma transferência é apertar ou soltar o botão ou rolar a esfera e os dados são codificados em três ou cinco bytes. Além disso, os fabricantes disponibilizam *drivers* (em *software*) que determinam a direção e a magnitude do movimento a partir do protocolo.



Outras tecnologias são utilizadas para sensoreamento dos deslocamentos. Nos *trackballs* são utilizados os potenciômetros e nos *mice* óticos, diodos emissores de luz.

5.3.6 Dispositivos Lógicos

Sob o ponto de vista dos projetistas de interface é conveniente classificar logicamente os periféricos, quanto a sua função, em:

- periféricos de entrada, quando o sentido da transferência de dados é do mundo exteriro ao computador. Por exemplo, *mouse*, teclado, etc.
- periféricos de saída, quando o sentido de transferência de dados é do computador para o mundo externo. Por exemplo, impressoras, monitores, etc.
- periféricos de entrada e saída, quando os dois sentidos de transferência são suportados. Por exemplo, os discos magnéticos.

E quanto às linhas disponíveis para transferências de dados, distinguem-se:

- periféricos com interface paralela; e
- periféricos com interface serial.

Por fim, podemos ainda classificar os periféricos quanto às caracterísitcas dos sinais que eles conseguem interfacear:

- periféricos com interface digital; e
- periféricos com interface analógica.

As interfaces paralelas são utilizadas para a comunicação em paralelo, ou seja, um fio para cada bit de dados. Esta interface é utilizada nas impressoras e nos discos rígidos. A comunicação em paralelo permite uma alta taxa de transferência de dados. A sua principal desvantagem é a necessidade de muitos fios (um para cada bit) o que a inviabiliza para a conexão entre elementos distantes. Pois, além do alto custo dos fios para a ligação entre o transmissor e o receptor, deve ser considerado o custo dos conectores para esta ligação. Outra dificuldade para a comunicação em paralelo para grandes distâncias é que, por ser maior o número de sinais a serem transmitidos, maior é a possibilidade de interferência externa destes sinais (por exemplo, uma descarga elétrica), exigindo assim para cada fio circuitos acionadores de linha (*drivers*) no transmissor e circuitos de filtragem e de proteção no receptor. A comunicação em paralelo só se justifica quando os elementos a serem ligados são elementos rápidos ou estão colocados a uma pequena distância.

As interfaces seriais são utilizadas em comunicação em série, quando os bits da palavra de uma determinada fonte são transmitidos um a um através de um único fio, como no caso do terminal e do *mouse*. Preferencialmente, a comunicação entre o computador e perifércios colocados a uma longa distância é feita de forma serial. As principais razões para isto são de carater econômico. Na comunicação serial o número de fios é bem menor (custo menor para instalação), bem como o número de circuitos necessários para acionamento, filtragem e proteção de sinais. Além disso, a comunicação serial pode utilizar os canais de comunicações já existentes (linahs telefônicas, canais radiofônicos, etc.).

Quando se trata de interfaces seriais distinguem-se ainda, quanto ao número de linhas dedicada aos dados, três tipos de configurações (Figura 9.22 do livro-texto, p. 745):

1. *simplex*: permite a transmissão de dados em uma única direção (só um fio).

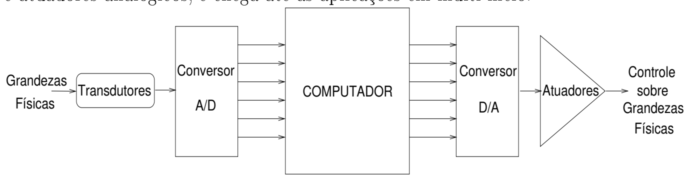
2. *half-duplex*: permite a transmissão de dados em uma única direção em cada instante (só um fio para o circuito de comutação).

3. *full-duplex*: permite a transmissão de dados em duas direções em cada instante (dois fios).

Um dispositivo com interface serial deve suportar algum protocolo de comunicação para codificar os dados a serem transferidos e/ou interpretar a sequência de bits recebidos. Vimos ainda que podemos distinguir o formato do protocolo basicamente em assíncrono e síncrono.

Embora todos os periféricos anteriormente descritos sejam acionados por sinais analógicos – tensões e/ou correntes com caracterísitcas analógicas, seus sinais de entrada e saída são no formato digital. Isso é porque internamente os sinais digitais são convertidos em sinais analógicos e vice-versa, facilitando a sua interligação com computadores de uso genérico. Quando se trata de periféricos de uso específico, como aquecedores elétricos ou máquinas fresadoras controlados por computadores, usualmente não vem incorporado neles os conversores D/A e/ou A/D, requerendo que o projetista desenvolva uma interface para compatibilizar os dois tipos de sinais. Portanto, consideramos como os dispositivos com interface analógica aqueles que só aceitam sinais

elétricos com características analógicas. As aplicações deste tipo de interfaces vão desde o controle de processo, feito através de sensores/transdutores e atuadores analógicos, e chega até as aplicações em multi-meio.



Exercícios de Revisão

1. Resolver os exercícios 1, 9 e 14 do capítulo 9 do livro-texto.

2. O que você entende por renovação de um monitor?

3. O que é o tempo de acesso de disco?

4. Cite dois fatores que influenciam a densidade de armazenamento de um disco magnético?

5. O que é um código de Manchester?

6. Qual é a diferença entre uma impressora sem impacto e de impacto?

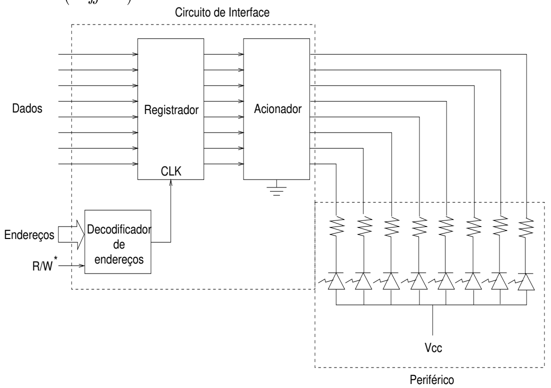
7. Por que as impressoras a *laser* e matriciais tem uma gama de cores bem limitadas (usualmente até 8 cores)?

8. Explique o funcionamento de um *mouse*.

9. Como os periféricos podem ser classificados logicamente?

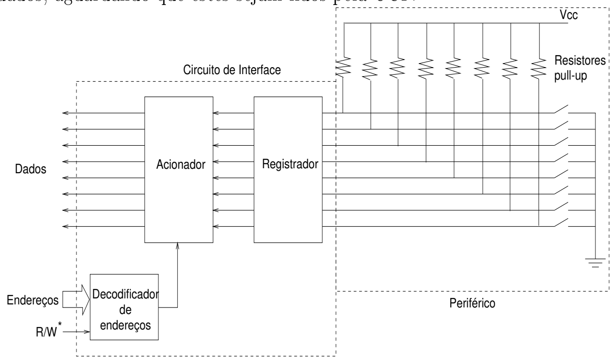
5.4 Circuitos de Interface

Quando se trata de periféricos de saída, a sua ligação com o computador é feita através de uma **interface de saída** ou **porta de saída**. Como um exemplo muito simples da aplicação de uma porta de saída, citamos a ligação entre o barramento de 8 bits de um computador com um conjunto de diodos emissores de luz (LED). Nesta ligação, a entrada do registrador de 8 bits é ligada diretamente ao barramento de dados (um bit de entrada por entrada D do *latch*). Cada saída Q do registrador é conectada a um inversor acionador e cada saída deste acionador é ligada a um LED. O gatilho do registrador é ligado ao decodificador de endereços que é função do endereço onde o registrador está alocado e do sinal de controle que indica um ciclo de escrita (R/W*=0) (Seção 5.4.1). Toda vez que o processador executar uma instrução de escrita neste endereço, o valor binário do barramento de dados é retido no registrador, podendo assim liberar o barramento para o próximo ciclo de acesso. O conteúdo do registrador é então exibido num LED (aceso se o bit for igual a 1, apagado se o bit for igual a zero). A simplicidade deste exemplo ajuda a evidenciar a essência de todo o circuito de interface de saída que é a existência de registradores para reter os dados e um acionador para prover alimentação correta aos periféricos. Os registradores são denominados **retentores** (*buffers*).



Um periférico de entrada é ligada com o computador através de uma **interface de entrada** ou **porta de entrada**. Vejamos a ligação de um conjunto de chaves do tipo liga-desliga ao barramento de 8 bits de um computador. Este circuito pode ser utilizado para a entrada de valor de parâmetros de configuração do sistema de computador, como por exemplo, quantidade de memória instalada e tipo de vídeo instalado. Ele é baseado em um acionador com a configuração de saída de três-estados, ligada diretamente ao barramento de dados. Cada entrada do acionador é ligada a uma saída de

um registrador. Cada entrada do registrador é conectada a um *pull-up* resistor e a uma chave ligada a terra. Se a chave estiver aberta, o resistor força a entrada para o nível lógico "1". Se a chave estiver fechada, a entrada do acionador é forçada para o nível lógico "0". O controle de habilitação do acionador é feito a partir de um sinal gerado por um decodificador de endereços que considera o endereço ao qual o registrador de entrada está alocado e o sinal de controle que indica um ciclo de leitura (R/W*=1). Somente quando o processador estiver executando uma instrução de leitura neste endereço, o valor binário de cada chave é transferido para o registrador onde os dados aguardam a sua vez de serem transferidos para o barramento. Como no circuito de interface de saída, o registrador desempenha o papel de retentor dos dados, aguardando que estes sejam lidos pela UCP.



Os periféricos de entrada e saída utilizam o circuito de **interface bidirecional** ou **porta bidirecional** para se conectarem com o computador. Essencialmente este circuito de interface é uma combinação dos dois circuitos anteriores. Ou seja é uma porta que, nos ciclos de escrita no seu endereço se comporta como uma interface de saúida e nos ciclos de leitura no mesmo enderço se comporta como uma interface de entrada. Os elementos básicos desta interface são também os registradores para os ciclos de escrita e os acionadores com a configuração de saída de três estados para os ciclos de leitura.

Vale ressaltar aqui que vários retentors podem ser ligados em cascata para acomodar melhor o fluxo de dados numa transferência. Dependendo da quantidade de registradroes ligados em cascata, dizemos que o circuito de interface é sem armazenamento (*non-buffering*, sem registrador), com armazenamento simples (*single-buffering*, 1 registrador) ou então armazenamento duplo (*double-buferin*, 2 registradores).

Com estes exemplos podemos listar algumas funções de um circuito de interface:

- suportar os modos de E/S disponíveis no computador;
- reconhecer o endereço dos seus registradores quando este aparecer no barramento de endereços;
- prover retentores para acomodar o fluxo de dados;
- prover sinais apropriados de temporização quando requeridos; e
- compatibilizar eletricamente os sinais entre o barramento e os periféricos.

O projeto de um circuito de interface se divide em 2 partes principais:

1. o projeto de ligação do controlador com o barramento do computador; e
2. o projeto da interface elétrica, ou seja da ligação com o periférico.

5.4.1 Interface com Barramento

O projeto da ligação do controlador com o barraemnto independe do periférico e pode de uma forma geral ser dividido em:

- projeto da ligação do barramento de dados,
- projeto da ligação do barramento de endereços, e
- projeto da ligação dos sinais de controle de acesso.

Interface com Barramento de Dados

Em geral a ligação de um periférico ao barramento de dados é feita de forma direta, cada bit de dados a uma linha do barramento. Entretanto, atenção deve ser prestada aos seguintes itens:

a largura do barramento de dados: Em geral o circuito de interface deve ser projetado com a mesma largura de barramento de dados a que estarão ligados. Por exemplo, para barramento de 8 bits ou de 16 bits.

a ordem dos bits: Quando se liga dispositvos fabricados por fornecedor diferente daquele do barramento, uma atenção especial deve ser tomada quanto o índice do bit menos significativo. A maior parte dos fabricantes adota o índice 0 (p. ex.: D0) para indicar o bit menos significativo, mas existem exceções.

o alinhamento dos bits: Quando se liga dispositivos com número de bits menor que a largura do barramento de dados, deve-se definir o alinhamento dos bits daquele dipositivo com os do barramento. Por exemplo, para ligar um dispositivo de 8 bits a um barramento de 16 bits é necessário definir a partir de que bit do barramento os bits do dispositivo estarão ligados. Em geral, eles podem estar ligados aos 8 bits menos significativos ou aos 8 bits mais significativos.

as caracterísitcas elétricas dos bits de dados: Obviamente os níveis elétricos dos bits dos periféricos devem ser compatíveis eletricamente com o barramento. Se forem sinais digitais, cuidados devem ter com o fenômeno vai-volta (*bounce*) nas chaves mecânicas e níveis de tensão dos periféricos. No primeiro caso um simples circuito de *debouncer* filtraria os sinais espúrios gerados pela inércia do movimento da chave e no segundo caso, tradutores de níveis são utilizados para compatibilizar os níveis de tensões. Quando se tratam dos sinais analógicos, quando o parâmetro de entrada/saída tem uma faixa de variação contínua, conversores A/D e D/A são necessários (Figs. 8.6 e 8.7 do livro-texto, p. 633 e 634). Circuitos de conversosres A/D e D/A será visto mais detalhadamente na seção5.4.2.

Interface com Barramento de Controle

Para a ligação com os sinais de controle é necessário saber os tipos de barramento e os modos de iniciação de uma transferência suportados pela unidade processadora.

Vimos que existem basicamente dois tipos de barramento: síncrono e assíncrono. Nos acessos síncronos o intertravamento das operações é definido pelo sinal de relógio, comum aos dois componenetes que participam da transferência e nos acessos assíncronos, o intertravamento das operações é estabelecido pelos sinais de *strobe* e reconhecimento.

Os processadores que suportam ambos os tipos de barramento, como os 68000, requer que o circuito de interface identifique o tipo de acesso a ser executado através do sinal VPA* (em nível alto, o acesso é assíncrono; caso contrário, o acesso é síncrono). Quando se trata de um acesso assíncrono, o circuito de interface precisa prover o sinal de reconhecimento para concluir um ciclo de acesso. Note que se um periférico tiver uma velocidade de processamento muito menor do que a da UCP, o circuito de interface deve introduzir estados de espera, atrasando o sinal de reconhecimento.

Como já comentamos antes, a inclusão do barramento síncrono nos processadores 68000 tem a finalidade de facilitar a sua conexão com os periféricos da família 6800. Fig. 8.9 do livro-texto (p. 636) apresenta uma lógica para interfacear o barramento síncrono do 68000 com um periférico da família 68000. Note como o sinal VPA* é gerado em função dos sinais de *strobe* e do endereço onde a interface está alocada. Fig. 8.12 do livro-texto (p. 640) mostra uma lógica que interfaceia o barramento assíncrono do 68000 com um periférico da família 6800. Fig. 8.24 do livro-texto (p. 655) mostra uma lógica que interfaceia o barramento assíncrono do 68000 com um periférico provido de uma interface de acesso assíncrono.

Para definir o sentido de uma transferência (entrada ou saída) o circuito de interface deve ser ligado ao sinal que define se o ciclo de barramento é de leitura ou de escrita. No caso do MC68000, o sinal R/W* tem esta finalidade. Como nos acesos à memória, R/W*=1 indica um ciclo de leitura quando o sentido de fluxo de dados é do mundo exterior para o computador e R/W*=0 indica um ciclo de escrita quando o sentido de fluxo de dados é do computador para o mundo exterior. Assim sendo, a lógica de habilitar os registradores e/ou acionadores no circuito de interface deve depender destes sinais, conforme o que foi comentado antes.

Distinguem-se três modos de iniciação de uma transferência: programado, por interrupção e por DMA. No caso de E/S programado, o circuito de interface deve dispor de um ou mais registradores para armazenar o estado do periférico. Esta informação é utilizada pelo programador para iniciar uma E/S. Para que um circuito de interface possa permitir transferências por interrupção, é necessário que os sinais do circuito de interface que identifiquem os eventos estejam ligados aos sinais de interrupção do barramento (Figs. 8.24, p. 655, 9.7, p. 709, 9.16, p. 737 do livro-texto). E, finalmente, para que um circuito de interface possa executar transferências por DMA é necessário que o sinal de seleção da interface esteja relacionado com o sinal de requisição de um canal do DMA. Além disso, ele deve conter uma lógica que desabilite todos os sinais de seleção dos outros circuitos de interface que

não participem do ciclo de DMA (Fig. 8.15 do livro-texto, p. 643).

Observações:

- O barramento Multibus da Intel provê sinais de acesso distintos para acessos à memória (MEMR – leitura/MEMW – escrita) e acessos à entrada e saída (IOR – leitura/IOW – escrita).
- Alguns barramentos com E/S isolada dispõem de três sinais de controle para diferenciar acessos à memória e às interfaces: R/W* (leitura/escrita), M/IO* (acesso à memória ou aos periféricos) e START (*strobe* de dados).

Interface com Barramento de Endereços

A ligação ao barramento de endereços se dá através de uma lógica de decodificador de endereços. Vimos que distinguem-se basicamente duas arquiteturas de entrada e saída: E/S isolada e E/S mapeada em memória. No capítulo 3 vimos detalhadamente o projeto de um decodificador de endereços para arquitetura de E/S mapeada em memória. O decodificador consegue gerar os sinais de seleção das interfaces a partir dos endereços nos quais eles estão alocados. Se for uma arquitetura de E/S isolada, o projeto de decodificador segue o mesmo princípio, diferindo apenas no mapa de endereços.

Resta agora saber como os registradores nos circuitos de interface podem ser mapeados no subespaço de endereços alocados a cada interface. Para isso vale observar aqui que para fins de programação, distinguem-se três classes de registradores num circuito de interface:

Registradores de controle: para definir as caracterísitcas operacionais da interface.

Registradores de estado: para armazenar informação sobre o estado corrente do periférico e a sua atividade.

Registradores de dados: para prover uma área de armazenamento de dados para acomodar melhor o fluxo de dados numa transferência.

Ou seja, nem todos os registradores são de escrita e de leitura. Isso nos permite utilizar várias alternativas para mapeá-los no subespaço de endereços reservado para cada interface:

com endereços únicos: cada registrador é referenciado por um endereço. Exemplo: os registradores do PI/T MC68230 são mapeados um a um no espaço de memória (Figura 8.2 do livro-texto, p. 629).

com endereços ambíguos: dois registradores, um de escrita e outro de leitura, compartilham o mesmo endereço. O sinal R/W* é utilizado para distingui-los. Exemplo: os registradores da DUART MC68680 são mapeados de dois a dois (Figura 8.3 do livro-texto, p. 629).

com uso de bits de referência: os bits de um ou mais registradores são utilizados para especificar univocamente o registrador dentro de um conjunto de registradores com o mesmo endereço na memória (Figura 8.4 do livro-texto, p. 630).

com uso de registradores de referência: o endereço "interno" do registrador de interesse é carregado no registrador de referência antes de acessar o seu conteúdo. Há algumas interfaces que suportam auto-incremento.

Exercícios de Revisão

1. Resolver o exercício 4 do capítulo 8 do livro-texto.
2. Quais são os cuidados que se deve ter para ligar um periférico com o barramento de dados?
3. Qual é a principal função de registradores retentores num circuito de interface?
4. Quais são os dois tipos de barramento? Como els podem influenciar no projeto de um circuito de interface?
5. Como os modos de iniciação de uma transferência pode influenciar no projeto de um circuito de interface?
6. Como se conecta um periférico ao barramento de endereços?

7. Quais são as três classes de registradores num circuito de interface?
8. Quais são as estratégias de mapeamento dos registradores de um circuito de interface no espaço de E/S?

5.4.2 Circuitos de Conversão

A maioria das variávies é de natureza analógica, enquanto os sistemas computacionais processam os dados no formato digital. O circuito de interface entre eles deve conter cinco blocos funcionais:

Transdutor: é o dispositivo que converte sinais físicos e não-elétricos, como pressão, temperatura, posição, intensidade luminosa e velocidade, em sinais elétricos. Exemplos de transdutores são termostatos, fotocélulas, tacômetros.

Conversor analógico-digital: é o dispositivo que converte o sinal analógico em sinal digital.

Sistema computacional: é o dispositivo que processa e monitora os sinais digitais captados.

Conversor digital-analógico: é o dispositivo que converte os sinais digitais em sinais analógicos de diferentes níveis de tensão ou corrente.

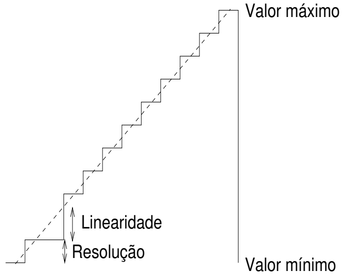
Atuador: é o dispositivo que atua diretamente sobre uma grandeza física, como válvulas.

É fácil ver que os dispositivos que interfaceiam o mundo analógico e o mundo digital propriamente ditos são os conversores D/A e A/D. Estes estão conquistando cada vez mais a importância à medida que aumenta o uso de computadores no controle de processos que antes era considerado inviável. Hoje em dia existe uma uma grande oferta de conversores D/A e A/D em pastilhas de circuitos integrados, alguns dos quais utilizam circuitos especializados para obter maior resolução, maior linearidade, menor erro de escala total, de linearidade e de quantização. Portanto, em muitas situações a função dos projetistas de sistemas computacionais se reduz a entender os dados fornecidos pelos fabricantes e avaliar a adequabilidade de um determinado produto em uma aplicação específica.

A **resolução**, muitas vezes especificada pelos fabricantes em porcentagem, é a menor variação analógica que pode ser distinguida por um conversor A/D ou produzida por um conversor D/A, ou seja

$$K = \frac{A_{fs}}{2^n - 1},$$

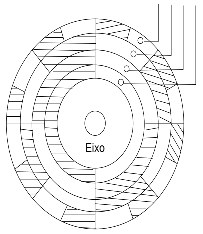
onde A_{fs} é tensão/corrente analógica em escala total e n é o número de bits utilizados para representação. Muitos fabricantes especificam a resolução do seus conversores pelo número de bits. Um conversor de 16 bits tem uma resolução maior do que um de 8 bits. A **linearidade** diz respeito ao desvio máximo de uma linha reta traçada entre os níveis de entrada mínimo e máximo do conversor. Ela pode ser expressa em porcentagem da escala total ou em fração do valor da grandeza analógica (tensão, corrente, etc) do bit menos significativo. O **erro de escala total**, usualmente expresso em porcentagem, tem a ver como o máximo desvio da saída de um conversor D/A em relação ao valor analógico esperado; enquanto o **erro de linearidade** indica o máximo desvio que pode ocorrer no intervalo alocado para cada representação binária. Finalmente, *settling time* é o tempo que o conversor necessita para gerar todas as saídas ao varrer sequencialmente todas as possíveis entradas.



Embora os projetistas de sistemas computacionais possam ver os dispositivos de conversão como caixas pretas, daremos aqui alguma noção básica do esquema de conversão.

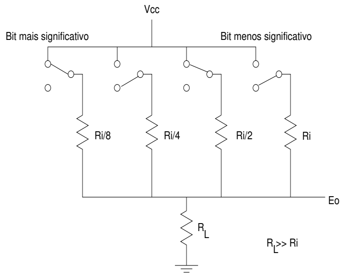
O primeiro conversor analógico/digital é um codificador de posição angular que converte uma posição angular de um eixo num número digital codificado em binário. O seu funcionamento se baseia na leitura de um disco de segmentos codificados por um conjunto de escovas. As escovas são ligadas de modo que uma só escova seja posicionada no centro de cada faixa

concêntrica do disco. Cada faixa é construída com vários segmentos feitos ou de material condutor ou de algum material isolante. Um sinal correspondente ao nível lógico 1 é aplicado nos segmentos de material condutor, de modo que uma escova sobre ele resultará em saída 1; caso contrário, uma saída 0. Outras tecnologias podem ser utilizadas para detectar a existência ou não do sinal, como sinais luminosos (segmentos opacos ou transparentes). Vale observar que neste tipo de conversor há problema de ambiguidade na transição entre dois conjuntos de segmentos, uma vez que a área da escova não é infinitesimal. Para amenizar este problema, utiliza-se o código de Gray (ou código de distância unitátia) para formar o disco codificador.



Disco codificador de 4 bits

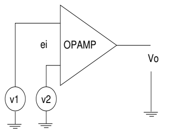
O problema da conversão de um sinal digital numa tensão analógica pode ser resolvido por inúmeras formas. O princípio básico de todas elas consiste em variar a tensão de saída em função das entradas binárias. A mais direta forma de conversão envolve o uso de uma estrutura de resistrores e um conjunto de chaves cuja posição liga e desliga correspondem, respectivamente, ao sinal 1 e 0, como mostra o seguinte circuito. Ao variar as posições das chaves, teremos diferentes valores de tensão na saída E$_o$. O cuidado que se deve ter é que as tensões de entrada que representam os valores binários devem ser estabelecidas exatamente, e, já que a carga nas entradas irá variar com o valor dos números de entrada, a entrada deve ser capaz de fornecer a corrente necessária sem um deslocamente no nível de tensão.

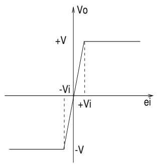


Um dispositivo eletrônico muito utilizado na implementação de conversores é o amplificador operacional. O amplificador operacional é funcionalmente um dispositivo cuja saída é dada por

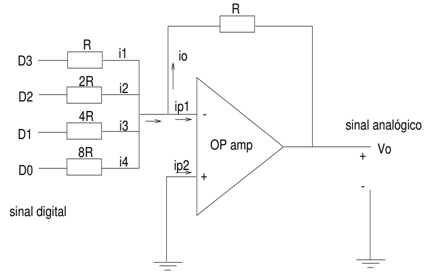
$$V_o = Ae_i,$$

onde A é o ganho do operacional e e_i, o diferencial das duas entradas v_1 (inversora) e v_2 (não inversora). Por um lado, o ganho tende a ser infinito no caso ideal e por outro, as fontes de alimentação são limitadas, resultando numa tensão de saída V_o limitada nos dois níveis de tensão +V e -V. Somente no intervalo (-Vi,Vi), com e_i próximo de 0, o amplificador opera linearmente. Uma forma de forçar que o operacional trabalhe nesta região linear é introduzir uma realimantação negativa, de modo que a saída acompanhe o valor de tensão na entrada inversora. Outra característica interessante do operacional é que a saída V_o não depende da carga, desde que a sua corrente máxima seja respeitada. Em outras palavras, ele consegue acionar uma grande faixa de carga.





Um circuito básico de conversão D/A com uso de operacionais é mostrado no esquema abaixo.



Neste caso, e_i é aproximadamente nulo devido à realimentação negativa. Considerando ainda que as correntes de entrada sejam ip1 e ip2 nulas ($Z_in \rightarrow \infty$), então

$$io = i1 + i2 + i3 + i4 = \frac{V_{in}}{R} + \frac{V_{in}}{2R} + \frac{V_{in}}{4R} + \frac{V_{in}}{8R},$$

ou seja

$$V_o = -Ri_o = -R(\frac{V_{in}}{R} + \frac{V_{in}}{2R} + \frac{V_{in}}{4R} + \frac{V_{in}}{8R}) = -(V_{in} + V_{in}\frac{1}{2} + V_{in}\frac{1}{4} + V_{in}\frac{1}{8})$$

Portanto, os 16 níveis de sinais analógicos na saída é função dos sinais de caracterísitcas digitais (0 V ou V_{in}).

Os principais tipos de conversores A/D eletrônicos utilizam os conversores D/A como parte do seu circuito e um operacional operando na região saturada. O princípio básico da sua operação consiste dos seguintes passos:

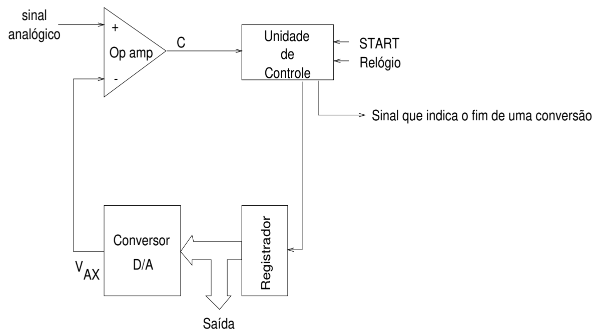
1. o comando START inicia a operação com o conteúdo do registrador zerado;

2. na frequência do relógio, o conteúdo do registrador é incrementado;

3. o conteúdo do registrador é convertido automaticamente para o sinal analógico V_{AX};

4. o sinal analógico V a ser convertido é **comparado** com o sinal V_{AX} correspondente ao código binário no registrador. Se V for diferente de V_{AX}, o sinal C terá nível lógico alto; senão assumirá um valor próximo de zero;

5. O sinal C com valor próximo de zero indica que V e V_{AX} são aproximadamente iguais. Ou seja, o código binário no registrador corresponde à representação digital de V.



Observações:

- A organização dos resistores no circuito de D/A apresentado nesta seção não é factível na prática, principalmente quando a entrada é composta por muitos *bits*, requerendo o uso de resistores de resistências muito altas. Existe uma configuração alternativa onde somente resistores de resistência R e 2R são necessários.
- AD7524 é um conversosr D/A de 8 bits fornecido por vários fabricantes.

Exercícios de Revisão

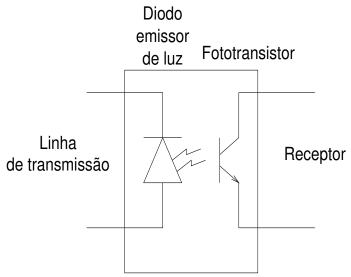
1. Qual é a função de um transdutor?
2. Qual é a função de conversores A/D e D/A?
3. Qual é a função de um atuador?
4. Quantos níveis de tensão um conversor D/A de 8 bits pode produzir?
5. O que pode acontecer com a resolução e a escala total de um conversor D/A quando a tensão de referência for incrementada de 20%?

6. O que é erro de escala total?
7. O que é erro de linearidade de um conversor?
8. Como é feita a conversão de um sinal analógico em um sinal digital com uso de um conversor D/A?

5.4.3 Adicionais Circuitos

Dependendo do tipo de ligações, circuitos adicionais podem ser necessários, como

Circuito de desacoplamento entre a linha e o dispositivo , para isolar os dispositivos (receptores) digitais, usualmente projetados para operarem no nível de tensão até algumas dezenas de volts, dos surtos de tensão que possam aparecer na linha de transmissão. Um exemplo de dispositivo de proteção é o isolador ótico, no qual os sinais da linha são eletricamente desconectados dos sinais dos dispositivos. O contato se dá sob forma ótica através da luz emitida pelo diodo.



Circuito de balanceamento de linhas para garantir a **rejeição de modo comum** (*common mode rejection*). Chamamos de linhas não-balanceadas, quando os sinais elétricos na linha de transmissão são referenciados ao potencial de um terra comum e que o potencial desse terra não é igual em todos os pontos. Neste caso, pode ocorrer o fenômeno conhecido como retorno por terra (*earth loop* ou *ground loop*) e o diferencial de tensão no transmissor pode ser diferente do diferencial no receptor, levando a interpretações errôneas dos sinais. Nas linhas balanceadas, por sua vez, a saída do transmissor é balanceada em relação ao terra local. Ao invés de enviar o sinal (em uma linha), envia-se o diferencial deste sinal em relação ao terra (em duas linhas) (Fig. 9.31 do livro-texto, p. 756).

5.4.4 Exemplos de Circuitos de Interface

Nesta seção são apresentados dois circuitos de interface da família 68000 para ilustrar os blocos funcionais apresentados: um circuito de interface paralela (MC68230) e um circuito de interface serial (MC68681).

Preparo para Próxima Aula

- Leia atentamente as pp. 648–677 do livro texto e descreva sucintamente os blocos funcionais e a interface de programação do MC68230.
- Leia atentamente as pp. 721–739 do livro texto e descreva sucintamente os blocos funcionais e a interface de programação do MC68681.

MC68230

Além de prover dos circuitos de interface com o barramento de dados (D0–D7), o barramento de endereços (RS0–RS5, CS*) e o barramento de controle (RESET*, CLK, DTACK*, R/W*), MC68230 dispõe de um controlador de interrupção (PIRQ* e PIACK*), de um controlador de DMA (DMAREQ*) e de um temporizador (TIACK*, T_{OUT} e T_{IN}) – todos programáveis. MC68230, ou simplesmente PI/T, suporta ainda duas portas paralelas de entrada e saída programáveis de 8 bits, A e B, e uma porta de função dual (porta paralela/temporizador), C, para se interligar com os periféricos. O modo de transferência de dados é controlado por um controlador de *handshaking* (Fig. 8.23 do livro-texto, p. 654).

Este circuito de interface distingue sete modos de operação:

1. Modo 0: transferência unidirecional (programável através do registrador DDR) de 8 bits (Fig. 8.25 do livro-texto, p. 657)

   (a) Submodo 00: entrada como duplo armazenamento e a saída com simples armazenamento.
   (b) Submodo 01: entrada sem armazenamento e saída com duplo armazenamento.
   (c) Submodo 1X: entrada sem armazenmaneto e saída como simples armazenamento.

2. Modo 1: transferência unidirecional (programável através do registrador DDR) de 16 bits (Fig. 8.26 do livro-texto, p. 661)
   (a) Modo X0: entrada com duplo armazenamento e saída com simples armazenamento.
   (b) Modo X1: entrada sem armazenamento e saída como duplo armazenamento.

3. Modo 2: transferência bidirecional de 8 bits para a porta B. Neste caso, H1 e H2 controlam a transferência de saída e H3 e H4, a entrada. A porta A (sem handshaking) pode ser utilizada como entrada sem armazenamento ou como saída com armazenamento simples (Fig. 8.27 do livro-texto, p. 664).

4. Modo 3: transferência bidirecional de 16 bits (Fig. 8.28 do livro-texto, p. 666).

Há dois modos de *handshaking* no controle de transferência de dados com os periféricos:

1. Intertravado: o sinal H2/H4 é ativado quando PI/T está pronto para a transferência e desativado assim que o periférico ativar o sinal H1/H3 indicando que a operação foi reconhecida pelo PI/T. Em seguida, H1/H3 é desativado indicando que reconheceu o término do ciclo (Fig. 8.20, p. 651, e Fig. 8.22, p. 653, do livro-texto).

2. Pulsado: o sinal H2/H4 é ativado quando PI/T está pronto para a transferência e desativado depois de 4 ciclos de relógio.

O temporizador é um contador de contagem regressiva, cujo valor máximo é programável (CPR) (Fig. 8.29 do livro-texto, p. 671).

MC68681

Além de prover dos circuitos de interface com o barramento de dados (D0–D7), o barramento de endereços (RS1–RS4, CS*) e o barramento de controle (RESET*, CLK, DTACK*, R/W*), a interface DUART ou MC68681 dispõe de um controlador de interrupção (IRQ* e IACK*) e de um circuito de serialização nos dois canais assíncronos de entrada e saída, A e B (Fig. 9.12 do livro-texto, p. 722). Cada canal dispõe de quatro registradores de entrada e dois registradores de saída para acomodar melhor o fluxo de dados. Além disso, a interface DUART (*dual universal asynchronous receiver/transmitter*) tem cinco modos de operação: full-duplex, *automatic-echo*, *local-loopback*, *remote-loopback* e *multidrop mode* (Fig. 9.13 do livro-texto, p. 724) e contém um gerador programável de *baud-rate* (Tab. 9.6 do livro-texto, p. 728). Outra caracterísitca interessante da interface DUART é a sua capacidade de mascarar os eventos oriundos dos periféricos (Fig. 9.14 do livro-texto, p. 733).

Exercícios de Revisão

1. Resolver os exercícios 7 e 8 do capítulo 8 do livro-texto.
2. Resolver os exercícios 3, 6, 10, 12 e 13 do capítulo 9 do livro-texto.
3. Qual é a estratégia de mapeamento dos registradores utilizada em MC68230?
4. Qual é a estratégia de mapeamento dos registradores utilizada em MC68230?

5.5 Padrões de Interfaces

Com objetivo de facilitar a interligação dos componentes de um sistema computacional, esforços são feitos no sentido de padronizar as interfaces definindo precisamente as características mecânicas (dimensões e números de linhas/pinos), elétricas (nívies de tensão e corrente, capacidade de carga), funcionais (assinalamento das funções a linhas/pinos) e procedurais (interdependência entre os sinais em linhas/pinos) que os fabricantes dos dispositivos

devem obedecer. Nesta seção apresentaremos alguns destes padrões de interface.

Preparo para Próxima Aula

- Leia atentamente as pp. 677–683 do livro texto e descreva sucintamente as especificações do padrão Centronics.
- Leia atentamente as pp. 683–699 do livro texto e descreva sucintamente as especificações do padrão IEEE488.
- Leia atentamente as pp. 746–759 do livro texto e descreva sucintamente as evolução do padrão RS-232C para RS-422.

5.5.1 Padrão Centronics

O padrão Centronics define 8 linhas de dados, além dos sinais de controle do fluxo de dados e sinais de controle da impressora (Figura 8.33 e 8.34 do livro-texto, p. 678 e 679). Ele é utilizado para conexão paralela dedicada (ponto-a-ponto) entre um periférico e um sistema computacional.

5.5.2 Padrões RS232C, RS-423 e RS-422

É um padrão para ligação serial dedicada (ponto-a-ponto). Os níveis de tensão especificados pelo RS232C permitem comunicações confiáveis numa distância máxima de 15 metros na taxa de transferência de 20.000 bits/s.

O padrão descreve os seguintes sinais de controle:

1. RTS (*request to send*): é o sinal do DTE (*data terminal equipment*) para DCE (*data communication equipment*). Quando ativado indica que DTE deseja transmitir um dado.
2. CTS (*clear to send*): é o sinal do DCE para DTE. Quando ativado indica que DCE está pronto para receber o dado.
3. DSR (*data set ready*): é o sinal do DCE para DTE para indicar que o primeiro está pronto para receber um dado.

4. DTR (*data terminal ready*): é o sinal do DTE para DCE. Quando ativado indica que DTE está pronto para aceitar um dado do DCE.

O protocolo de comunicação é simples, através do procedimento RTS-CTS *handshaking* (Fig. 9.28 do livro-texto, p. 753).

O padrão RS232C e as suas versões revisadas não conseguem acompanhar a demanda da tecnologia disponível, que requer uma transmissão de dados cada vez mais distante e rápida. Dois novos padrões surgiram em resposta a esta demanda: RS422 e RS423. Diferentemente de RS232C, estes novos padrões somente especificam os aspectos elétricos da conexão e não as características funcionais. RS422 foi concebido para caminho de transmissão balanceado e RS423, para caminho de transmissão desbalanceado (Tabela 9.11 do livro-texto, p. 758).

5.5.3 Padrão IEEE488

É um padrão de conexão que especifica a capacidade mínima de um barramento. É também conhecido como GPIB (*general purpose interface bus*), HPIB (*Hawlet Packard instrumentation bus*), IEC 625-1 bus, ANSI MC1-1 bus e ASC II bus. O barramento suporta a interconexão de até 15 dispositivos de instrumentação (osciloscópio, voltímetros, impressora, PCs), com uma taxa de transferência de até 250 000 bytes/s em uma distância de 20 metros. Uma taxa de transferência de 1Mbyte/s é possível com uma distância menor.

Qualquer dispositivo conectado ao barramento pode assumir um ou mais de um dos três seguintes modos de operação:

1. ouvinte (*listener*): recebe os dados;

2. talker (*locutor*): coloca os dados; e

3. controlador (*controller*): gerencia e determina quem é *listener* e quem é *talker*. Só pode ter um controlador no barramento em cada instante.

Ela é composta de 24 linhas, das quais

- 8 linhas são dedicadas à transferência da dados (barramento de dados);

- 3 linhas ao controle de dados no barramento de dados;

- 5 linhas ao gerenciamento de barramento:

1. ATN (*attention*): o barramento de dados contém mensagem de controle do controlador (=1) ou mensagem específico do dispositivo de um locutor para um ou mais ouvintes (=0).
2. IFC (*interface clear*): é utilizado pelo controlador para resetar as interfaces do barramento IEEE.
3. SRQ (*service request*): é utilizado pelo dispositivo para solicitar atenção do controlador, detectável através do mecanismo de *polling*.
4. REN (*remote enable*): permite que o dispositivo seja conectado ao barramento (=1) ou disconectado do barramento, passando a ser controlado localmente (=0).
5. EOI (*end of identify*): é ativado pelo locutor para indicar o fim da mensagem e pelo controlador que causa *poll paralelo* no qual até 8 dispositivos pode inidicar se eles requerem serviço.

- 8 linhas de retorno de terra.

Uma caracterísitca marcante de IEEE488 é o suporte a *broadcasting*, ou seja, um locutor pode enviar um dado a vários ouvintes simultâneamente.

A transferência de dados no barramento de dados é controlada por três linhas através de um protocolo de *handshaking* totalmente intertravado. (Figs. 8.40, p. 690, 8.41, p. 691, 8.42, p. 693, e 8.43, p. 694, do livro-texto).

Exercícios de Revisão

1. Resolver os exercícios 14, 15, 16, 17, 18 e 19 do capítulo 8 do livro-texto.

2. Resolver o exercício 8 do capítulo 9 do livro-texto.

5.6 Auto-avaliação

Após este capítulo, você deve ser capaz de:

- definir entrada/saída de um sistema computacional.

- citar dois tipos de processamento de E/S suportados pelos processadores.
- explicar como é um acesso direto à memória.
- distinguir um periférico de um circuito de interface.
- distinguir os modos de transferência (síncrono e assíncrono) e os modos de comunicação serial (síncrono e assíncrono).
- distinguir os mecanismos de sincronismo utilizados em cada modo de comunicação serial.
- explicar o que é um protocolo de comunicação.
- classificar logicamente os periféricos.
- explicar as funções de um circuito de interface.
- definir os principais módulos de um circuito de interface.
- listar quais as considerações a serem levadas em conta no projeto de ligação com os barramentos de um sistema computacional.
- explicar o princípio básico de funcionamento dos circuitos de conversores D/A e A/D.
- saber quais os parâmetros utilizados pelos fabricantes para especificar um circuito de conversão.
- entender o esquema básico do circuito de conversão D/A e A/D com uso de amplificador operacional.
- programar circuitos de interface através das especificações dos fabricantes.
- explicar o papel dos padrões de interfaces na uniformização dos dispositivos fornecidos pelos diferentes fabricantes.
- distinguir um padrão de uma implementação do padrão.
- descrever sucintamente o padrão Centronics.

- descrever sucintamente os padrões RS-232, RS-423 e RS-422.
- descrever sucintamente o padrão IEEE488.

5.7 Lista de Exercícios

1. Resolver o exercício 5 do capítulo 8 do livro-texto.
2. Resolver o exercício 13 do capítulo 8 do livro-texto.
3. Pesquise sobre as interfaces SCSI e EIDE utilizados nos PCs para interfaceá-los com os dispositivos de alta capacidade de armazenamento. Compare-as.
4. Substituindo os convencionais circuitos lógicos integrados, as unidades microcontroladoras (MCU) permitem controlar uma variedade de periférico com um mesmo *hardare* por meio de programação. Em anexo encontra-se o projeto de um termostato com uso do microcontrolador programável M68HC705C8 que inclui os conversores A/D, controladores de *display* de cristal líquido (LCD) e fluorescente a vácuo (VFD). Analise o projeto, sob o aspecto de *hardware* (as conexões físicas) e de *software* (programação dos sinais, pino a pino) e explique, com suas próprias palavras, o controle do display LCD **ou** o controle do teclado.