



Mecatrônica

4ª edição

Uma abordagem multidisciplinar

W. Bolton

Sobre o autor

William Bolton foi consultor do *Further Education Unit and Head of Research and Development and Monitoring* no BTEC. É consultor da UNESCO e autor de vários livros de sucesso em engenharia.



B694m Bolton, William.

Mecatrônica [recurso eletrônico] : uma abordagem multidisciplinar / William Bolton ; tradução: José Lucimar do Nascimento. – 4. ed. – Dados eletrônicos. – Porto Alegre : Bookman, 2010.

Editado também como livro impresso em 2010.
ISBN 978-85-7780-712-3

1. Mecatrônica. I. Título.

CDU 621

William Bolton

Mecatrônica ^{4ª edição}

Uma abordagem multidisciplinar

Tradução:

José Lucimar do Nascimento
Engenheiro Eletrônico e de Telecomunicação (PUC-MG)
Especialista em sistemas de controle (UFMG)
Professor e Coordenador de Ensino do CETEL

Consultoria, supervisão e revisão técnica desta edição:

Antonio Pertence Júnior
Engenheiro Eletrônico e de Telecomunicações
Especialista em Processamento de Sinais (Ryerson University – Canadá)
Professor de Telecomunicações da FUMEC (MG)
Professor Titular da Faculdade de Sabará/MG

Versão impressa
desta obra: 2010



2010

Obra originalmente publicada sob o título *Mechatronics: A Multidisciplinary Approach, 4th Edition*
ISBN 9780132407632

Copyright © Pearson Education Limited 2008.

The translation of MECHATRONICS – ELECTRONIC CONTROL SYSTEMS IN
MECHANICAL AND ELECTRONIC ENGINEERING 04 Edition is published by arrangement
with Pearson Education Limited.

All Rights Reserved.

Capa: *Rogério Grilho*

Leitura final: *Théo Amon*

Editora Sênior: *Denise Weber Nowaczyk*

Projeto e editoração: *Techbooks*

Reservados todos os direitos de publicação, em língua portuguesa, à
ARTMED® EDITORA S.A.
(BOOKMAN® COMPANHIA EDITORA é uma divisão da ARTMED® EDITORA S.A.)
Av. Jerônimo de Ornelas, 670 - Santana
90040-340 Porto Alegre RS
Fone (51) 3027-7000 Fax (51) 3027-7070

É proibida a duplicação ou reprodução deste volume, no todo ou em parte,
sob quaisquer formas ou por quaisquer meios (eletrônico, mecânico, gravação,
fotocópia, distribuição na Web e outros), sem permissão expressa da Editora.

SÃO PAULO

Av. Embaixador Macedo Soares, 10.735 - Pavilhão 5 - Cond. Espace Center
Vila Anastácio 05095-035 São Paulo SP
Fone (11) 3665-1100 Fax (11) 3667-1333

SAC 0800 703-3444

IMPRESSO NO BRASIL
PRINTED IN BRAZIL

Prefácio

O termo **mecatrônica** foi “inventado” por um engenheiro japonês em 1969, como uma combinação de “meca”, de *mecanismos*, e “trônica”, de *eletrônica*. Atualmente o mundo associa esta palavra a um significado maior: mecatrônica é uma filosofia na tecnologia de engenharia em que há uma integração coordenada e desenvolvida de forma concorrente da engenharia mecânica com a eletrônica e o controle computacional inteligente no projeto e manufatura de produtos e processos. Como resultado, muitos produtos que usavam apenas recursos mecânicos incorporaram sistemas envolvendo microprocessadores. Isto resultou em maior flexibilidade, em reprojeto e reprogramação mais fáceis e em capacidade de processar dados e informações automatizados.

Uma consequência desta abordagem é a necessidade de os engenheiros e técnicos adotarem uma abordagem integrada e interdisciplinar da engenharia. Portanto, engenheiros e técnicos necessitam de habilidades e conhecimentos não restritos a uma única área. Eles precisam ser capazes de desenvolver e relacionar uma gama de disciplinas de engenharia e se ligar àqueles com habilidades mais especializadas. Este livro aborda os fundamentos da mecatrônica e fornece indicações para outras fontes com habilidades mais especializadas.

A primeira edição foi projetada para abordar conteúdos de mecatrônica do Business and Technology Education Council (BTEC) para cursos da Higher National Certificate/Diploma para técnicos e também para se adequar a unidades de conhecimentos mais específicos, como projeto, manufatura e manutenção determinadas pela área de aplicação do curso. Este livro foi amplamente utilizado nestes cursos bem como em cursos universitários na Grã-Bretanha e nos Estados Unidos. Em conformidade com as avaliações feitas pelos leitores destes dois países, a segunda edição foi consideravelmente estendida e, com sua abordagem mais profunda, foi não apenas relevante para os leitores originais, mas também adequada para cursos de graduação. A terceira edição incorporou refinamentos em algumas explicações, mais discussões de microcontroladores e programação, aumento do uso de modelos de sistemas mecatrônicos e a reunião de informações importantes nos apêndices. A quarta edição é uma reconsideração completa de todos os aspectos do texto, leiaute e conteúdo, com alguns reagrupamentos de tópicos, transferência de mais tópicos

para os apêndices para evitar descontinuidade no estudo do texto, novos assuntos, em particular a introdução à inteligência artificial, mais estudos de caso e o refinamento de alguns tópicos para melhorar a compreensão. Além disso, os objetivos e o resumo dos principais pontos foram incluídos em cada capítulo.

O objetivo geral do livro é proporcionar uma abordagem abrangente da mecatrônica que pode ser usada em cursos técnicos e de engenharia e, portanto, ajudar o leitor em:

- Adquirir um conjunto de habilidades em engenharia mecânica, eletrônica e computação necessárias a fim de o leitor ser capaz de compreender e projetar sistemas mecatrônicos.
- Ser capaz de desenvolver e inter-relacionar uma gama de disciplinas de engenharia necessárias em mecatrônica.
- Ser capaz de projetar sistemas mecatrônicos.

Cada capítulo inclui objetivos, um resumo, ilustrações e contém problemas, sendo que as respostas deles se encontram no final do livro. No Capítulo 24 foram incluídas também atividades de pesquisa e de projeto, bem como as possíveis respostas.

O Capítulo 1 é uma introdução à mecatrônica. Os Capítulos de 2 a 6 formam um bloco coerente sobre sensores e condicionamento de sinais, os Capítulos de 7 a 9 tratam de atuadores, os Capítulos de 10 a 16, de modelos de sistemas, os Capítulos de 17 a 23, de sistemas microprocessados, e o Capítulo 24 apresenta uma conclusão geral sobre projetos de sistemas microprocessados.

Um guia para o professor (*Instructor's Guide*), em inglês, e apresentações em Power Point (em português) estão disponíveis para download na exclusiva Área do Professor em:

www.bookman.com.br

Reconhecemos a grande contribuição das publicações dos equipamentos dos fabricantes citadas neste livro. Gostaria de agradecer aos revisores na Grã-Bretanha e nos Estados Unidos que cuidadosamente leram a terceira edição desta obra e fizeram sugestões de melhorias.

W. Bolton

Sumário

1 Introdução à Mecatrônica	11	2.11 Seleção de sensores	69
1.1 O que é mecatrônica?	11	2.12 Entrada de dados por chaves	70
1.2 Desenvolvimento de um projeto	13	Resumo	73
1.3 Sistemas	15	Problemas	74
1.4 Sistemas de medição	17	3 Condicionamento de Sinais	77
1.5 Sistemas de controle	18	3.1 Condicionamento de sinais	77
1.6 Controlador lógico programável	28	3.2 O amplificador operacional	78
1.7 Exemplos de sistemas mecatrônicos	29	3.3 Proteção	89
Resumo	32	3.4 Filtragem	91
Problemas	33	3.5 Ponte de Wheatstone	92
2 Sensores e Transdutores	34	3.6 Modulação por pulso	97
2.1 Sensores e transdutores	34	Resumo	99
2.2 Terminologia para definição de desempenho	35	Problemas	99
2.3 Deslocamento, posição e proximidade	40	4 Sinais Digitais	101
2.4 Velocidade e movimento	52	4.1 Sinais digitais	101
2.5 Força	56	4.2 Sinais analógicos e digitais	102
2.6 Pressão de fluido	56	4.3 Conversores digital-analógico e analógico-digital	105
2.7 Vazão de líquido	60	4.4 Multiplexadores	112
2.8 Nível de um líquido	62	4.5 Aquisição de dados	114
2.9 Temperatura	63	4.6 Processamento de sinais digitais	116
2.10 Sensores de luz	68		

Resumo	118	8.4 Cames.....	206
Problemas	118	8.5 Engrenagens	207
5 Lógica Digital	120	8.6 Catraca e lingueta	211
5.1 Lógica digital	120	8.7 Transmissão por correia e por corrente	211
5.2 Portas lógicas	121	8.8 Rolamentos	213
5.3 Aplicações de portas lógicas.....	128	8.9 Aspectos mecânicos da seleção de motores	216
5.4 Lógica sequencial	133	Resumo	217
Resumo	141	Problemas	218
Problemas	142	9 Sistemas de Acionamento Elétrico	220
6 Sistemas de Apresentação de Dados	144	9.1 Sistemas elétricos	220
6.1 Displays	144	9.2 Chaves mecânicas	221
6.2 Elementos de apresentação de dados	146	9.3 Chaves de estado sólido	222
6.3 Gravação magnética	151	9.4 Solenoides	229
6.4 Gravação óptica	156	9.5 Motores CC	229
6.5 Displays	156	9.6 Motores CA	237
6.6 Sistemas de aquisição de dados.....	161	9.7 Motores de passo	239
6.7 Sistemas de medição.....	165	Resumo	245
6.8 Teste e calibração.....	168	Problemas	246
Resumo	170	10 Modelos de Sistemas Básicos	248
Problemas	171	10.1 Modelos matemáticos.....	248
7 Sistemas de Acionamento Pneumático e Hidráulico	174	10.2 Blocos construtivos de sistemas mecânicos.....	249
7.1 Sistemas de acionamento	174	10.3 Blocos construtivos de sistemas elétricos.....	256
7.2 Sistemas hidráulicos e pneumáticos	174	10.4 Blocos construtivos de um sistema fluídico	261
7.3 Válvulas de controle direcional.....	179	10.5 Blocos construtivos de um sistema térmico	268
7.4 Válvulas de controle de pressão.....	183	Resumo	271
7.5 Cilindros	184	Problemas	271
7.6 Válvulas de controle proporcional e servo-assistida	188	11 Modelos de Sistemas	275
7.7 Válvulas de controle de processos	190	11.1 Sistemas de engenharia	275
7.8 Atuadores rotativos	196	11.2 Sistemas rotacional-translacional	275
Resumo	196	11.3 Sistemas eletromecânicos.....	276
Problemas	197	11.4 Linearidade	280
8 Sistemas de Acionamento Mecânico	199	11.5 Sistemas hidráulico-mecânico	282
8.1 Sistemas mecânicos.....	199	Resumo	285
8.2 Tipos de movimentos	200	Problemas	285
8.3 Cadeia cinemática	202		

12 Resposta Dinâmica de Sistemas 286

12.1 Modelagem de sistemas dinâmicos...	286
12.2 Terminologia	287
12.3 Sistemas de primeira ordem	289
12.4 Sistemas de segunda ordem	296
12.5 Medidas de desempenho para sistemas de segunda ordem	302
12.6 Identificação de sistemas	306
Resumo	306
Problemas	308

13 Funções de Transferência de Sistemas 310

13.1 Função de transferência	310
13.2 Sistemas de primeira ordem	313
13.3 Sistemas de segunda ordem	316
13.4 Sistemas em série	318
13.5 Sistemas com malhas de realimentação	319
13.6 Efeito da localização dos polos na resposta transiente	321
Resumo	324
Problemas	324

14 Resposta de Frequência 326

14.1 Entrada senoidal	326
14.2 Fasores	327
14.3 Resposta de frequência	330
14.4 Gráficos de Bode	333
14.5 Especificação de desempenho	342
14.6 Estabilidade	343
Resumo	345
Problemas	345

15 Controladores em Malha Fechada 347

15.1 Processos de controle contínuo e discreto	347
15.2 Terminologia	348
15.3 Modo <i>on/off</i>	350
15.4 Modo proporcional	352
15.5 Controle derivativo	354

15.6 Controle integral	356
15.7 Controlador PID	357
15.8 Controladores digitais	359
15.9 Desempenho dos sistemas de controle	361
15.10 Sintonia do controlador	363
15.11 Controle de velocidade	365
15.12 Controle adaptativo	365
Resumo	368
Problemas	369

16 Inteligência Artificial 371

16.1 Qual é o significado de inteligência artificial?	371
16.2 Percepção e conhecimento	372
16.3 Raciocínio	373
16.4 Aprendizado	377
Resumo	378
Problemas	378

17 Microprocessadores 379

17.1 Controle	379
17.2 Sistemas microprocessados	380
17.3 Microcontroladores	391
17.4 Aplicações	411
17.5 Programação	412
Resumo	416
Problemas	416

18 Linguagem Assembly 417

18.1 Linguagens	417
18.2 Conjuntos de instruções	418
18.3 Programas em linguagem assembly	425
18.4 Sub-rotinas	430
18.5 Listagens (LUTs)	434
18.6 Sistemas embarcados	437
Resumo	441
Problemas	441

19 Linguagem C 443

19.1 Por que C?	443
19.2 Estrutura de um programa	443
19.3 Desvios e <i>loops</i>	450
19.4 Arranjos	454

19.5	Ponteiros	456
19.6	Desenvolvimento de programas	457
19.7	Exemplos de programas	459
	Resumo	461
	Problemas	462
20	Sistemas de Entrada/Saída	464
20.1	Interfaceamento	464
20.2	Endereçamento de entrada/saída	465
20.3	Requisitos de interfaces	467
20.4	Adaptadores de interface para periférico	475
20.5	Interface de comunicação serial	480
20.6	Exemplos de interfaceamento	484
	Resumo	487
	Problemas	488
21	Controladores Lógicos Programáveis	489
21.1	Controlador lógico programável	489
21.2	Estrutura básica de um CLP	490
21.3	Processamento de entrada/saída	493
21.4	Programação em ladder	495
21.5	Lista de instruções	499
21.6	Operação <i>latch</i> e relés internos	502
21.7	Sequenciamento	504
21.8	Temporizadores e contadores	506
21.9	Registradores de deslocamento	509
21.10	Controles mestre e de salto	510
21.11	Manipulação de dados	511
21.12	Entrada/saída analógica	514
	Resumo	516
	Problemas	516
22	Sistemas de Comunicação	519
22.1	Comunicações digitais	519
22.2	Controle centralizado, hierárquico e distribuído	520
22.3	Redes	522
22.4	Protocolos	524
22.5	Modelo de comunicação por interconexão de sistemas abertos	526
22.6	Interfaces de comunicação serial	529
22.7	Interfaces de comunicação paralela	534
22.8	Protocolos <i>wireless</i>	538
	Resumo	538
	Problemas	539
23	Identificação de Falhas	540
23.1	Técnicas de detecção de falhas	540
23.2	Temporizador <i>watchdog</i>	541
23.3	Verificação por paridade e por codificação de erro	542
23.4	Falhas comuns de <i>hardware</i>	544
23.5	Sistemas microprocessados	545
23.6	Emulação e simulação	549
23.7	Sistemas de CLP	551
	Resumo	553
	Problemas	554
24	Sistemas Mecatrônicos	555
24.1	Projetos de mecatrônica	555
24.2	Estudos de caso	566
	Resumo	582
	Problemas	582
	Atividades de pesquisa	583
	Atividades de projeto	583
Apêndices		
A	Transformada de Laplace	585
B	Sistemas de numeração	595
C	Álgebra booleana	602
D	Conjuntos de instruções	613
E	Funções de biblioteca da linguagem C	620
F	MATLAB e SIMULINK	624
	Informações complementares	630
	Respostas	632
	Índice	651

1

Introdução à Mecatrônica

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Explicar o significado que tem a mecatrônica e compreender a sua importância em projetos de engenharia.
- Explicar o significado de sistema e definir seus elementos de medida.
- Descrever as várias formas e elementos de sistemas de controle em malha fechada e malha aberta.
- Reconhecer a necessidade do modelamento de sistemas para poder prever o seu comportamento.

1.1 O que é mecatrônica?

O termo **mecatrônica** foi inventado por um engenheiro japonês em 1969 como uma combinação de *meca*, de mecanismos, e *trônica*, de eletrônica. Essa palavra agora tem um significado amplo, sendo usada para descrever uma filosofia da tecnologia de engenharia na qual existe uma integração coordenada e simultânea entre a engenharia mecânica, a eletrônica e o controle computacional inteligente no projeto e fabricação de produtos e processos. Como resultado, os produtos de mecatrônica têm muitas funções mecânicas desempenhadas por sistemas eletrônicos. Isto proporciona uma flexibilidade muito maior, facilidade de reprogramação e reprojeção e uma capacidade do sistema de realizar automaticamente o envio e a recepção de dados.

Um sistema mecatrônico não se resume a apenas um casamento de sistemas elétricos e mecânicos, e é mais do que um sistema de controle: ele é uma integração completa de todos estes sistemas na qual há uma abordagem simultânea destes no projeto. Tal abordagem integrada e interdisciplinar está sendo cada vez mais adotada no projeto de engenharia de veículos, robôs, ferramentas mecânicas, máquinas de lavar, câmeras e diversas outras máquinas. Essa integração que vai além dos limites tradicionais das engenharias mecânica, elétrica, eletrônica e de controle tem

ocorrido em fases mais iniciais de projetos quando é necessário desenvolver sistemas mais baratos, confiáveis e flexíveis. A mecatrônica deve combinar simultaneamente essas disciplinas nos projetos, e não fazer apenas uma associação sequencial delas desenvolvendo, digamos, um sistema mecânico e depois desenvolvendo uma parte elétrica e outra microprocessada. Portanto, a mecatrônica é uma filosofia de projeto, uma abordagem integrada aplicada na engenharia.

À mecatrônica estão associadas áreas tecnológicas que envolvem sensores e sistemas de medidas, acionamentos e sistemas atuadores, e sistemas microprocessados (Figura 1.1), juntamente com a análise do comportamento dos sistemas e dos sistemas de controle. Esta é essencialmente a abordagem deste livro. Este capítulo, que é uma introdução ao assunto, desenvolve alguns dos conceitos básicos dando ao leitor uma visão geral do livro, cujos capítulos tratam de modo detalhado os assuntos aqui apresentados.

1.1.1 Exemplos de sistemas mecatrônicos

Considere uma câmera de foco automático e controle automático de exposição. O que você precisa para usar a câmera é enquadrar o objeto e pressionar um botão para obter a imagem. A câmera é capaz de ajustar o foco automaticamente assim que o objeto é enquadrado, e ajustar, também automaticamente, a abertura e a velocidade do obturador de forma a obter a exposição correta. Não é necessário fazer estes ajustes de forma manual. Considere agora a suspensão inteligente de um caminhão. Ela se ajusta quando há uma carga irregular para manter a plataforma nivelada, quando o caminhão faz uma curva, quando se move em um terreno irregular, etc., proporcionado um rodar macio. Considere uma linha de produção com vários processos automatizados. Ou seja, esses processos são realizados automaticamente em uma sequência correta, tendo um monitoramento em cada estágio do processo. Esses três sistemas são exemplos de uma integração entre a eletrônica, os sistemas de controle e a engenharia mecânica.

1.1.2 Sistemas embarcados

O termo **sistema embarcado** é usado onde microprocessadores são embutidos em sistemas, e é nesse tipo de sistema que geralmente estamos interessados em mecatrônica. Um microprocessador pode ser considerado como sendo essencialmente um conjunto de portas lógicas e elementos de memória que não são conectados como

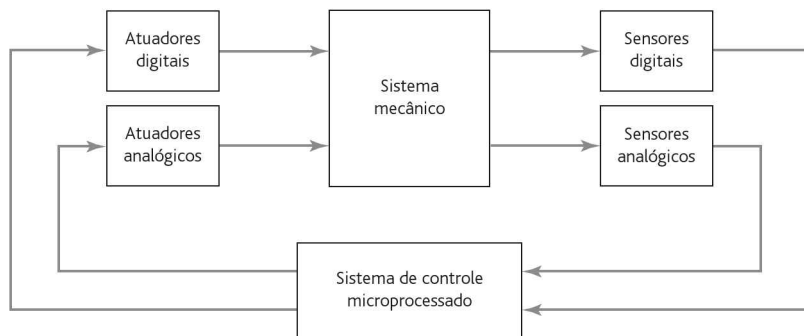


Figura 1.1 Elementos básicos de um sistema mecatrônico.

componentes individuais, mas cujas funções lógicas são implementadas conforme determinado pelo software. Como uma ilustração do significado de uma porta lógica, podemos necessitar de uma saída ativa quando as entradas A E (AND) B estiverem ambas ativas. Isso poderia ser implementado por um circuito conhecido como porta lógica AND. Uma porta lógica OR ativa sua saída quando a entrada A OU (OR) a entrada B estiver ativa. Um microprocessador procura identificar se suas entradas estão ativadas (*on*) ou desativadas (*off*), processa os resultados a partir dessa identificação, conforme definido pelo programa, e em seguida coloca esses resultados (*on* ou *off*) nas saídas. Consulte o Capítulo 15 para saber mais detalhes sobre microprocessadores.

Para utilizar um microprocessador em um sistema de controle, são necessários chips (circuitos integrados) adicionais para armazenamento de dados (memória) e portas de entrada/saída (*in/out*) para troca de sinais entre o microprocessador e o mundo externo. **Microcontroladores** são microprocessadores com essas funções integradas em um único chip.

Um **sistema embarcado** é um sistema microprocessado projetado para controlar uma gama de funções e que não foi projetado para ser programado pelo usuário da mesma forma que ocorre com um computador. Portanto, nos sistemas embarcados, o usuário não pode alterar o que o sistema faz acrescentando ou substituindo software.

Podemos ilustrar o uso de microcontroladores em um sistema de controle usando como exemplo uma máquina moderna de lavar roupas que tem um sistema de controle microprocessado para controlar o ciclo de lavagem, as bombas, o motor e a temperatura da água. Um carro moderno possui controles microprocessados, como o que evita o bloqueio das rodas em frenagens (ABS) e o que gerencia o funcionamento do motor. Outros exemplos de sistemas embarcados são as câmeras de foco automático e exposição automática, filmadoras, telefones celulares, aparelhos de DVD, leitores de cartão eletrônico, fotocopiadoras, impressoras, *scanners*, televisores e controladores de temperatura.

1.2 Desenvolvimento de um projeto

O desenvolvimento de projeto de qualquer sistema pode ser considerado como envolvendo vários estágios:

1 *A necessidade*

O desenvolvimento de um projeto pode iniciar a partir da necessidade de um consumidor ou um cliente. Essa necessidade pode ser identificada por uma pesquisa de mercado com a finalidade de definir as necessidades de consumidores potenciais.

2 *Análise do problema*

O primeiro estágio no desenvolvimento de um projeto é descobrir a verdadeira natureza do problema, isto é, analisá-lo. Esse estágio é importante porque a imprecisão na definição do problema pode conduzir a um desperdício de tempo num projeto que pode não satisfazer a necessidade que o originou.

3 *Definição da especificação*

Após a análise, fazemos a especificação dos requisitos. Isso descreve o problema, com qualquer restrição imposta à solução, e define o critério com o qual se

pode avaliar a qualidade do projeto. Na definição do problema devem ser especificadas todas as funções necessárias ao projeto, juntamente com quaisquer características desejáveis. Assim, podem ser definidas características como massa, dimensões, tipos de movimentos e suas extensões, precisão, requisitos de entrada e saída dos componentes, interfaces, alimentação, ambiente de operação, padrões relevantes e práticos, etc.

4 *Geração de soluções possíveis*

Essa etapa é normalmente denominada de **estágio conceitual**. As propostas de soluções são estruturadas com o nível de detalhamento suficiente para mostrar a implementação da solução dada a cada função especificada, ou seja, dimensões aproximadas, formatos, materiais e custos. Nesta etapa também são analisadas as soluções dadas a problemas similares; não faz sentido reinventar a roda.

5 *Seleção de uma solução adequada*

As diversas soluções são avaliadas e a mais adequada é escolhida. A avaliação geralmente envolve a representação do sistema através de um modelo, seguida de uma simulação para saber como ele reage às entradas.

6 *Construção do projeto detalhado*

Agora o detalhamento do projeto escolhido deve ser realizado. Pode ser necessária a produção de protótipos ou simulações para otimização do projeto.

7 *Documentação do projeto*

O projeto selecionado é então descrito em desenhos, diagramas de circuitos, etc., de forma que ele possa ser executado.

Não se deve considerar que os estágios de um projeto ocorrem na sequência, estágio por estágio. Normalmente há a necessidade de se retornar a um estágio anterior e tecer novas considerações. Por exemplo, quando estamos no estágio de geração de soluções possíveis, pode ser necessário retornar à análise do problema.

1.2.1 **Projetos tradicionais e de mecatrônica**

Um projeto de engenharia é um processo complexo que envolve a interação entre muitos conhecimentos e experiências. Em um projeto tradicional, o engenheiro mecânico projeta os elementos mecânicos, em seguida o engenheiro de controle projeta o sistema de controle. Por isso, essa abordagem pode ser denominada de projeto sequencial. Entretanto, o princípio da abordagem de um projeto de mecatrônica é a inclusão simultânea de conhecimentos de engenharia mecânica, eletrônica, tecnologia de computadores e engenharia de controle. A simultaneidade inerente dessa abordagem depende muito do modelamento do sistema e da simulação de como o modelo reage às entradas e, conseqüentemente, como o sistema real pode, reagir às entradas.

Como exemplo de como uma abordagem multidisciplinar pode ajudar na solução de um problema, considere o projeto de uma balança como as usadas em banheiro. Tais balanças podem ser consideradas apenas em termos da compressão de molas e do mecanismo usado para converter o movimento em rotação de um eixo e assim em movimento de um ponteiro sobre uma escala; um problema que pode ser considerado no projeto é que o peso indicado não deve depender da posição da pessoa sobre a balança. Entretanto, outras possibilidades podem ser consideradas se os conhecimentos aplicados forem além de um projeto puramente mecânico. Por

exemplo, as molas podem ser substituídas por células de carga com *strain gauges* (transdutor usado para medição de força), com indicação digital da medida do peso indicada por display de LED através de um microprocessador. Essa balança pode ser mecanicamente mais simples, envolvendo apenas alguns componentes e partes móveis. Entretanto, a complexidade é transferida para o software.

Como outro exemplo, podemos citar o projeto tradicional de um controlador de temperatura para uma central de aquecimento doméstica que tem um termostato bimetalico e um sistema de controle em malha fechada. A curvatura do par bimetalico varia de acordo com a variação da temperatura e é usado para acionar uma chave liga/desliga (*on/off*) no sistema de aquecimento. Entretanto, uma solução multidisciplinar para o problema pode fazer uso de um sistema microprocessado que pode empregar um termodiodo como sensor. Este sistema tem muitas vantagens em relação ao anterior. O sistema com termostato bimetalico é comparativamente mais simples, e a temperatura não é controlada com precisão; além disso, o método para controle de diferentes temperaturas em qualquer momento do dia é complexo e não é fácil de implementar. Entretanto, um sistema microprocessado pode controlar facilmente e com precisão por meio de um controle programável. Este sistema é muito mais flexível. A melhoria na flexibilidade é uma característica comum dos sistemas mecatrônicos em comparação com os tradicionais.

1.3 Sistemas

No projeto de sistemas mecatrônicos, uma das etapas é a criação de um modelo do sistema de forma que previsões possam ser feitas quanto ao seu comportamento quando as entradas ocorrerem. Tais modelos envolvem diagramas em bloco para representar os sistemas. Um **sistema** pode ser representado por um diagrama em bloco que tem uma entrada e uma saída em que não nos interessa saber o que há dentro do bloco, mas a relação entre entrada e saída. O termo **modelamento** é usado quando representamos o comportamento de um sistema real usando equações matemáticas que representam a relação entre as entradas e saídas do sistema. Por exemplo, uma mola pode ser considerada um sistema com uma força F como entrada e uma extensão x como saída (Figura 1.2(a)). A equação usada para modelar a relação entre a entrada e a saída pode ser $F = kx$, onde k é uma constante. Como um outro exemplo de sistema, podemos citar um motor que tem como entrada a energia elétrica e como saída a rotação de um eixo (Figura 1.2(b)).

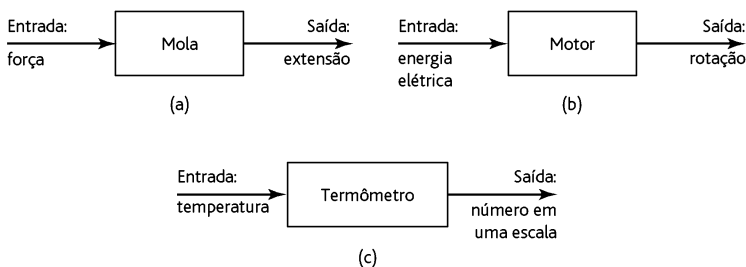


Figura 1.2 Exemplos de sistemas: (a) mola, (b) motor, (c) termômetro.

Um **sistema de medidas** pode ser considerado como uma caixa que é usada para realizar medições. Ele tem como entrada a grandeza medida e como saída o valor dela. Por exemplo, um sistema de medição de temperatura, ou seja, um termômetro, tem uma entrada de temperatura e uma saída que é um número em uma escala (Figura 1.2(c)).

1.3.1 Modelamento de sistemas

A resposta de qualquer sistema a uma entrada não é instantânea. Por exemplo, para o sistema mola representado na Figura 1.2(a), imagine que a relação entre a entrada, força F , e a saída, extensão x , seja dada por $F = kx$, que descreve apenas a relação quando o sistema está em regime estacionário. Quando a força é aplicada, é provável que oscilações ocorram antes da mola estabilizar no valor da extensão de estado estacionário (Figura 1.3). As respostas dos sistemas são funções do tempo. Portanto, para saber como o sistema se comporta quando entradas são aplicadas, precisamos elaborar modelos para os sistemas que relacionem a saída à entrada de forma que possamos determinar, para uma dada entrada, como a saída irá variar no tempo e em que valor ela irá estabilizar.

Em outro exemplo, se você ligar uma chaleira elétrica, levará algum tempo para que a água alcance o ponto de ebulição (Figura 1.4). Da mesma forma, quando um controlador microprocessado gera um sinal, digamos, para mover as lentes de focalização em uma câmera automática, leva algum tempo até a lente alcançar a posição correta do foco.

Em geral, a relação entre a entrada e a saída de um sistema é descrita por uma equação diferencial. Estas equações e sistemas são estudadas no Capítulo 10.

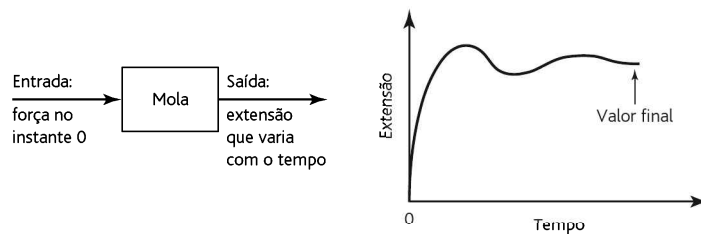


Figura 1.3 Resposta de uma mola a uma entrada.

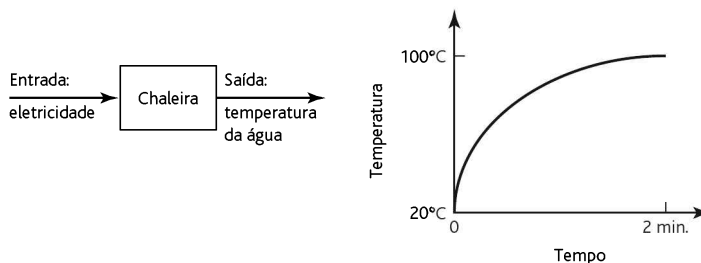


Figura 1.4 Resposta de um sistema chaleira elétrica a uma entrada.

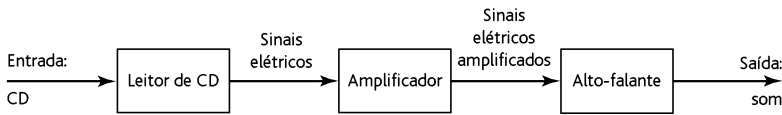


Figura 1.5 Um aparelho de CD.

1.3.2 Sistemas conectados

Exceto em sistemas muito simples, é normalmente apropriado considerá-los como uma série de blocos interconectados, com cada bloco tendo uma função específica. Assim, a saída de um bloco torna-se a entrada do bloco seguinte no sistema. Ao desenhar um sistema como este, é necessário saber que as linhas que conectam um bloco a outro indicam o fluxo da informação na direção indicada pela seta, não havendo necessariamente conexões físicas. Um exemplo de um sistema deste tipo é um aparelho de CD. Podemos considerá-lo como sendo três blocos interconectados: um leitor de CD que tem uma entrada CD e uma saída de sinais elétricos, um amplificador que tem como entrada a saída do leitor de CD e como saída os sinais elétricos amplificados, e um alto-falante que tem como entrada os sinais elétricos amplificados e como saída o som (Figura 1.5). A próxima seção, que trata de sistemas de medição, apresenta um outro exemplo de um conjunto de blocos interconectados.

1.4 Sistemas de medição

Os **sistemas de medição** têm uma importância específica em qualquer discussão na mecatrônica. Em geral, estes sistemas podem ser considerados como sendo constituídos de três partes básicas, conforme ilustra a Figura 1.6:

- 1 Um **sensor** que responde à grandeza sob medição produzindo em sua saída um sinal relacionado à grandeza. Por exemplo, um termopar é um sensor de temperatura. A entrada deste sensor é a temperatura e a saída é uma FEM (força eletromotriz) que está relacionada ao valor da temperatura.
- 2 Um **condicionador de sinal** que recebe o sinal do sensor e o transforma para um formato que seja apropriado para apresentação ou, no caso de um sistema de controle, para uso na operação de controle. Assim, por exemplo, a saída de um termopar é uma FEM muito pequena e pode ser amplificada através de um amplificador. Este circuito é um condicionador de sinal.
- 3 Um **sistema de display** é onde a saída do condicionador de sinal é mostrada. Esta parte pode, por exemplo, ser um ponteiro que se move sobre uma escala, ou então um mostrador digital.

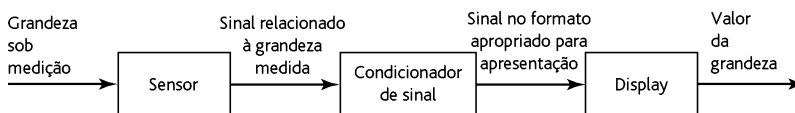


Figura 1.6 Um sistema de medição e suas partes constituintes.

Como exemplo, considere um termômetro digital (Figura 1.7). Ele tem uma entrada de temperatura por meio de um sensor que, provavelmente, é um diodo semiconductor. A diferença de potencial no sensor é, para uma corrente constante, uma medida da temperatura. Esta diferença de potencial é então amplificada por um amplificador operacional para produzir uma tensão que pode acionar diretamente um display. O sensor e o amplificador operacional podem ser incorporados no mesmo chip de silício.

Os sensores serão estudados no Capítulo 2 e os condicionadores de sinais, no Capítulo 3. Os sistemas de medição envolvendo todos os elementos serão estudados no Capítulo 6.

1.5 Sistemas de controle

Um **sistema de controle** pode ser considerado como aquele que pode ser usado para:

- 1 Controle de algumas variáveis em determinados valores. Por exemplo, um sistema central de aquecimento em que a temperatura é controlada em um determinado valor.
- 2 Controle de uma sequência de eventos. Por exemplo, uma máquina de lavar ajustada para, digamos, “roupas brancas” que passa a ser controlada dentro de um determinado ciclo de lavagem, ou seja, uma sequência de eventos apropriados ao tipo de roupa.
- 3 Controle de ocorrência de eventos. Por exemplo, uma trava de segurança em uma máquina em que ela não pode ser operada antes que uma trava esteja posicionada.

1.5.1 Realimentação

Considere o exemplo de um sistema de controle que todos nós temos no nosso corpo. A temperatura do nosso corpo, a menos que estejamos doentes, permanece quase constante independente de estarmos em um ambiente frio ou quente. Para manter a temperatura constante, o nosso corpo tem um sistema de controle de temperatura. Se a temperatura do nosso corpo começar a aumentar acima do normal, suamos; e se ela diminui, sentimos calafrios. Esses dois mecanismos são usados para restaurar a temperatura do nosso corpo de volta para o valor normal. O sistema de controle é que mantém a temperatura constante. Este sistema tem uma entrada a partir de sensores que “dizem” como a temperatura está, compara essa informação com o valor que a temperatura deveria ter, e fornece uma resposta apropriada para conseguir a temperatura desejada.

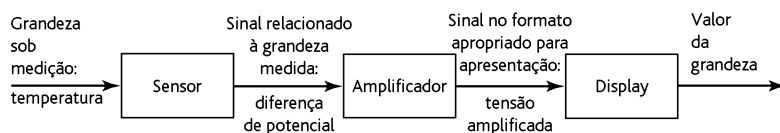


Figura 1.7 Um sistema de termômetro digital.

Este é um exemplo de um **controle por realimentação**: os sinais são realimentados a partir da saída, por exemplo, a temperatura real, para modificar a reação do corpo para que este restabeleça a temperatura ao seu valor normal. O controle por realimentação é feito pelo sistema de controle, que compara o que é realimentado da saída do sistema com o valor desejado e ajusta a saída em resposta a esta comparação. A Figura 1.8(a) ilustra esse sistema de realimentação.

Uma forma de controlar a temperatura de uma central de aquecimento doméstica é fazer uma pessoa ficar próxima à chave liga/desliga do aquecedor com um termômetro e ligar ou desligar o aquecedor de acordo com a leitura do termômetro. Essa é uma forma grosseira de um controle por realimentação usando uma pessoa como elemento de controle. O termo realimentação (*feedback*) é usado porque os sinais são realimentados da saída para modificar a entrada. Um dos sistemas de controle por realimentação mais usados tem um termostato ou um controlador que liga ou desliga automaticamente o aquecedor de acordo com a diferença entre a temperatura ajustada e a real (Figura 1.8(b)). Esse sistema de controle mantém a temperatura constante.

Quando pegamos um lápis em um banco, temos a necessidade de usar um sistema de controle para garantir que a nossa mão chegue até o lápis. Isso é feito por nós ao observarmos a posição da nossa mão em relação ao lápis e ajustarmos a sua posição à medida que a movemos em direção ao lápis. Existe uma realimentação da informação da posição da nossa mão em relação ao lápis de forma que possamos modificar as nossas reações para proporcionarmos o posicionamento e o movimento desejado da nossa mão (Figura 1.8(c)). Esse sistema de controle controla o posicionamento e o movimento da nossa mão.

Sistemas de controle por realimentação são bastante utilizados, não apenas na natureza e em casa, mas também na indústria. Existem muitos processos e máquinas industriais onde o controle, seja manual ou automático, é necessário. Por exemplo,

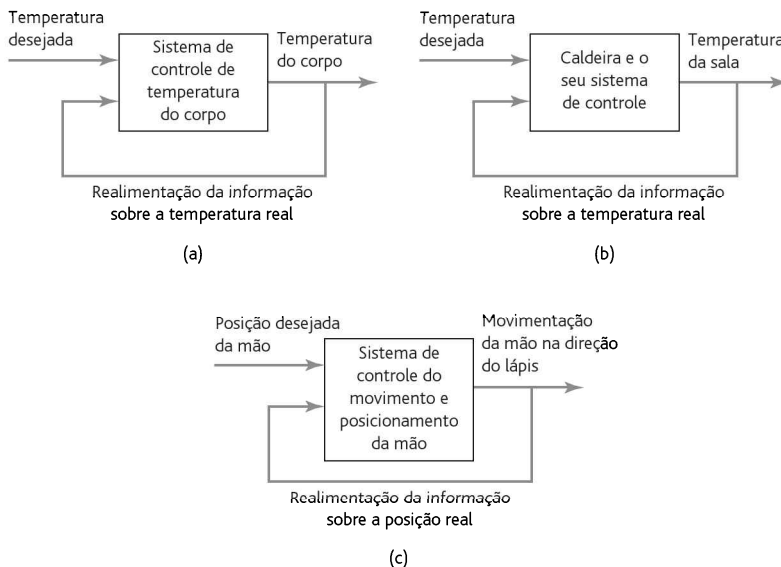


Figura 1.8 Controle por realimentação: (a) temperatura do corpo humano, (b) temperatura de uma sala com aquecimento central, (c) pegando um lápis.

existem controles de processos em que variáveis como temperatura, nível de líquido, vazão de fluido, pressão, etc. são mantidas constantes. Desta forma, em um processo químico pode haver a necessidade de manter o nível de um líquido em um tanque em um determinado valor ou temperatura. Existem também sistemas de controle que envolvem o posicionamento preciso e consistente de partes móveis ou a manutenção de uma velocidade constante. Como exemplo, podemos citar um motor projetado para girar em uma velocidade constante, ou ainda uma máquina na qual a posição, velocidade e operação de uma ferramenta são controladas automaticamente.

1.5.2 Sistemas em malha fechada e aberta

Existem dois tipos básicos de sistemas de controle: um denominado **malha aberta** e o outro, **malha fechada**. A diferença entre eles pode ser ilustrada por um exemplo simples. Considere um aquecedor que tem uma chave seletora que permite selecionar um elemento de aquecimento de 1 kW ou um outro de 2 kW. Se uma pessoa usar o elemento para aquecer uma sala, ela pode ligar apenas o elemento de 1 kW caso não se deseje uma temperatura alta. A sala será aquecida e alcançará a temperatura determinada pelo fato do elemento de aquecimento de 1 kW estar ligado, e não o de 2 kW. Se houver mudanças nas condições, digamos, alguém abre uma janela, não há uma forma da saída de aquecimento ser ajustada para compensar. Este é um exemplo de um controle em malha aberta no qual não há realimentação para o elemento de aquecimento para ajustá-lo e manter a temperatura constante. O sistema de aquecimento com o elemento de aquecimento pode ser implementado como um sistema em malha fechada se a pessoa tiver um termômetro e ligar ou desligar os elementos de 1kW e 2 kW de acordo com a diferença entre a temperatura real e a desejada, para manter a temperatura da sala constante. Neste caso há realimentação, pois a entrada do sistema é ajustada conforme sua saída esteja na temperatura desejada. Isso significa que o acionamento da chave depende do desvio da temperatura atual a partir da temperatura desejada, sendo essa diferença determinada por um elemento de comparação (neste caso, a pessoa). A Figura 1.9 ilustra estes dois tipos de sistemas.

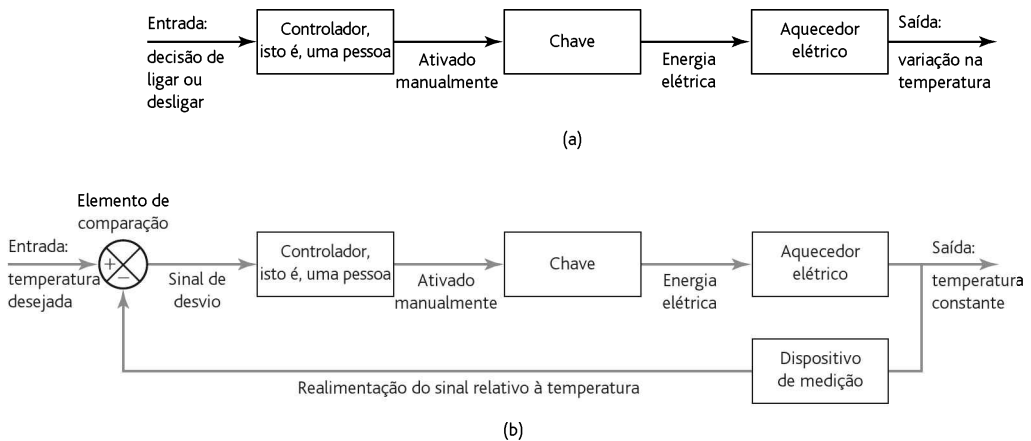


Figura 1.9 Aquecimento de uma sala: (a) um sistema em malha aberta, (b) um sistema em malha fechada.

Um exemplo de um sistema de controle em malha aberta comum é uma sandueira doméstica. O controle é feito ajustando-se um temporizador que determina o tempo para tostar o pão. Não há realimentação para controlar o nível desejado em que o pão seja tostado.

Para ilustrar mais a diferença entre sistemas em malha aberta e em malha fechada, considere um motor. Num sistema em malha aberta, a velocidade de rotação do eixo pode ser determinada unicamente pelo ajuste inicial no botão que afeta a tensão aplicada ao motor. Qualquer variação na tensão de alimentação, nas características do motor em função de variações na temperatura, ou ainda na carga no eixo faz variar a velocidade do eixo, mas sem compensação. Não há malha de realimentação. Entretanto, num sistema em malha fechada, o valor inicialmente ajustado no botão de controle para uma determinada velocidade do eixo é mantido pela realimentação, independente de qualquer variação na tensão de alimentação, nas características do motor ou na carga. Num sistema de controle de malha aberta, a saída do sistema não tem efeito sobre o sinal de entrada. Já num sistema de controle em malha fechada, a saída tem efeito sobre o sinal de entrada modificando-o para manter o sinal de saída no valor desejado.

Os sistemas em malha aberta têm a vantagem de ser relativamente simples e consequentemente de baixo custo e, geralmente, com boa confiabilidade. Entretanto, eles normalmente não são precisos, visto que não há correção de erro. Os sistemas em malha fechada têm a vantagem de ser relativamente precisos em relação à igualdade entre os valores real e desejado. Entretanto, eles são mais complexos, um pouco mais caros e possuem uma probabilidade maior de ficar inoperante em função do maior número de componentes.

1.5.3 Elementos básicos de um sistema em malha fechada

A Figura 1.10 mostra a forma geral de um sistema em malha fechada básico. Ele é constituído dos seguintes elementos:

1 Elemento de comparação

Compara o valor desejado ou de referência com o valor medido da variável controlada e produz um sinal de erro. Esta operação pode ser considerada como uma soma do valor do sinal de referência, que é positivo, com o valor do sinal medido, que é negativo neste caso:

$$\text{sinal de erro} = \text{valor do sinal de referência} - \text{valor do sinal medido}$$

Geralmente, o símbolo usado para um elemento no qual os sinais são somados é um círculo segmentado, com as entradas conectadas nos segmentos. Todas as entradas são somadas, portanto, a entrada de realimentação é assinalada como

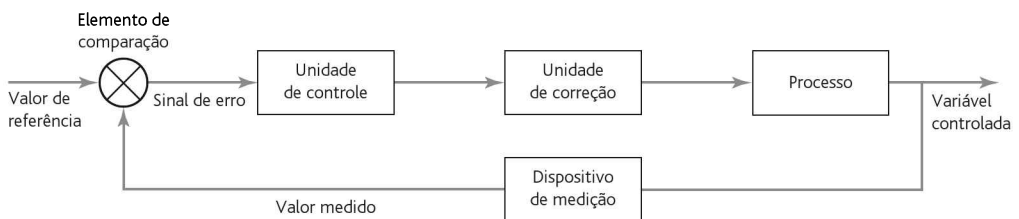


Figura 1.10 Elementos de um sistema de controle em malha fechada.

negativa e a de referência, como positiva, de forma que a soma resulte na diferença entre os sinais. Uma **malha de realimentação** é uma forma por meio da qual um sinal relativo à condição a ser obtida é realimentado para modificar o sinal de entrada para o processo. Quando o sinal que é realimentado é subtraído a partir do valor de entrada, temos uma **realimentação negativa**. Essa realimentação é a que é usada para controlar um sistema. A **realimentação positiva** ocorre quando o sinal realimentado se soma ao sinal de entrada.

2 *Elemento de controle*

Este elemento decide qual ação tomar quando recebe um sinal de erro. O elemento de controle pode gerar, por exemplo, um sinal que aciona uma chave ou abre uma válvula. O plano de controle usado por este elemento pode ser apenas fornecer um sinal de liga/desliga quando houver um erro, assim como no caso do termostato da sala, ou então um sinal que abra ou feche proporcionalmente uma válvula de acordo com a amplitude do erro. Os planos de controle podem ser **sistemas não programáveis**, nos quais os planos de controle são fixos, sendo determinados pela forma com que os elementos são interconectados, ou **sistemas programáveis**, nos quais o plano de controle é armazenado em uma unidade de memória e pode ser alterado (reprogramado). Os controladores são estudados no Capítulo 11.

3 *Elemento de correção*

O elemento de correção produz uma alteração no processo para corrigir ou mudar a condição controlada. Portanto, ele pode ser uma chave que liga um aquecedor, aumentando assim a temperatura do processo, ou uma válvula que abre e permite que mais líquido entre no processo. O termo **atuador** é usado para o elemento de uma unidade de correção que fornece a energia para executar a ação de controle. As unidades de correção são estudadas nos Capítulos 5 e 6.

4 *Elemento de processo*

O processo é o que está sendo controlado. Pode ser uma sala de uma casa em que a temperatura é controlada ou um tanque de água em que o nível é controlado.

5 *Elemento de medição*

O elemento de medição produz um sinal relacionado à variável do processo controlado. Ele pode ser, por exemplo, uma chave que liga quando uma determinada posição é alcançada, ou um termopar que fornece uma FEM relacionada à temperatura.

No sistema de malha fechada ilustrado na Figura 1.10 em que uma pessoa controla a temperatura de uma sala, estes elementos são:

Variável controlada	– a temperatura da sala
Valor de referência	– a temperatura desejada para a sala
Elemento de comparação	– a pessoa que compara os valores medido e desejado de temperatura
Sinal de erro	– a diferença entre as temperaturas medida e desejada
Unidade de controle	– a pessoa
Unidade de correção	– a chave no aquecedor
Processo	– o aquecimento feito pelo aquecedor
Dispositivo de medição	– um termômetro

Um sistema de controle automático da temperatura de uma sala pode envolver um termostato, um elemento sensível à temperatura, que liga quando a temperatura cai abaixo do valor ajustado e desliga quando a temperatura desejada é alcançada (Figura 1.11). Essa chave sensível à temperatura é usada para ligar o aquecedor. O termostato tem as funções combinadas de comparação entre as temperaturas desejada e atual e, a partir disto, controla a operação de uma chave. Este é um caso comum em que elementos do sistema de controle são capazes de combinar diversas funções.

A Figura 1.12 mostra um exemplo de um sistema de controle simples usado para manter o nível de água constante em um tanque. O valor de referência é ajustado previamente no mecanismo de alavanca da boia, de forma que a entrada de água seja cortada com o tanque no nível desejado. Quando o nível da água no tanque baixar, a esfera flutuante se move junto com o nível. Isso faz com que o mecanismo da alavanca gire e permita que mais água entre no tanque. Esse fluxo de água continua até que a esfera flutuante suba de forma que o movimento do mecanismo da alavanca corte a entrada de água no tanque. Este é um sistema de controle em malha fechada com os seguintes elementos:

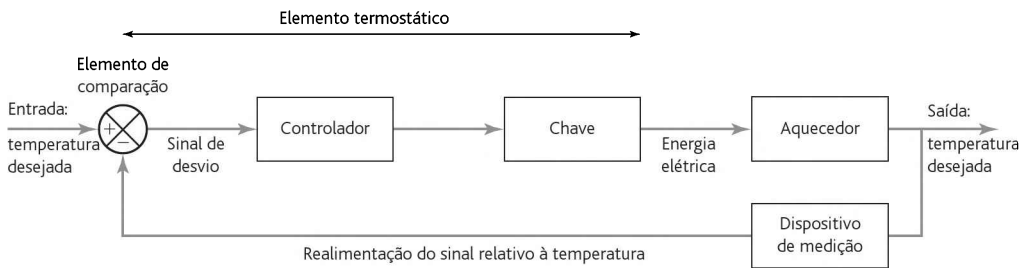


Figura 1.11 Aquecimento de uma sala: um sistema em malha fechada.

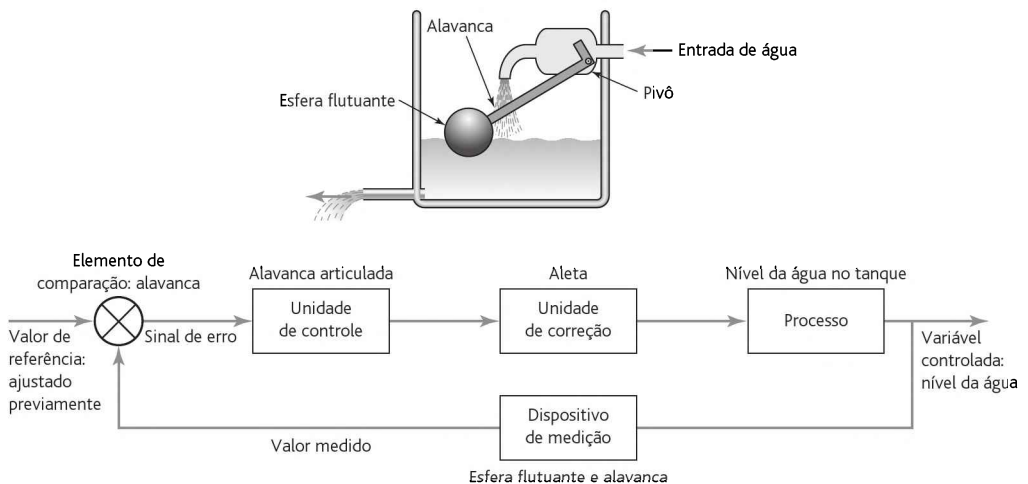


Figura 1.12 Controle automático do nível de água em um tanque.

Variável controlada	– nível da água no tanque
Valor de referência	– ajustado previamente posicionando-se o mecanismo constituído pela esfera flutuante e pela alavanca
Elemento de comparação	– a alavanca
Sinal de erro	– a diferença entre o valor real e o ajustado no posicionamento da alavanca
Unidade de controle	– a alavanca com pivô
Unidade de correção	– a aleta que abre ou fecha a entrada de água
Processo	– o nível da água no tanque
Dispositivo de medição	– a esfera flutuante e a alavanca

Este é um exemplo de um sistema de controle em malha fechada que envolve apenas dispositivos mecânicos. Entretanto, poderíamos certamente controlar o nível do líquido com um sistema de controle eletrônico. Assim, poderíamos ter um sensor que forneceria um sinal elétrico que seria usado, após um condicionamento adequado, como a entrada de um computador, onde este sinal seria comparado com um valor ajustado e a diferença entre eles, que seria o sinal de erro, seria usada para fornecer uma resposta apropriada na saída do computador. Este sinal, após um condicionamento, seria usado para controlar o movimento de um atuador na válvula de controle de vazão, determinando assim a quantidade de água que entra no tanque.

A Figura 1.13 mostra um sistema de controle automático simples da velocidade de rotação de um eixo. Um potenciômetro é usado para ajustar o valor de referência, ou seja, qual tensão é fornecida ao amplificador diferencial como valor de referência para a velocidade de rotação desejada. O amplificador diferencial é usado para comparar e amplificar a diferença entre os valores de referência e de realimentação, ou seja, amplifica o sinal de erro. Este sinal amplificado aciona um motor, que por sua

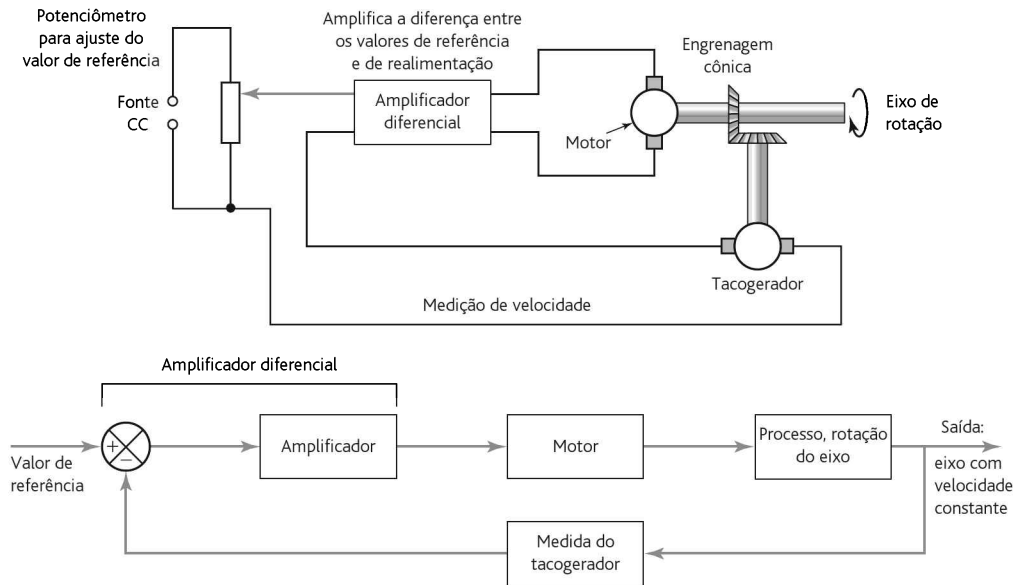


Figura 1.13 Controle de velocidade de um eixo.

vez determina a velocidade de rotação do eixo. Essa velocidade é medida usando-se um tacogerador conectado ao eixo do motor através de um par de engrenagens cônicas. O sinal do tacogerador é então realimentado no amplificador diferencial:

- Variável controlada – velocidade de rotação do eixo
- Valor de referência – ajustado no cursor do potenciômetro
- Elemento de comparação – amplificador diferencial
- Sinal de erro – a diferença entre as saídas do potenciômetro e do tacogerador
- Unidade de controle – o amplificador diferencial
- Unidade de correção – o motor
- Processo – a rotação do eixo
- Dispositivo de medição – o tacogerador

1.5.4 Sistemas de controle analógico e digital

Os **sistemas analógicos** são aqueles em que todos os sinais são funções contínuas do tempo e a amplitude do sinal é a medida da variável (Figura 1.14(a)). Os exemplos discutidos até agora neste capítulo são sistemas deste tipo. Os **sinais digitais** podem ser considerados uma sequência de sinais *on/off*, o valor de uma variável representado por uma sequência de pulsos *on/off* (Figura 1.14(b)).

A Figura 1.15 mostra os elementos básicos de um sistema de controle em malha fechada; compare-o com o sistema analógico em malha fechada na Figura 1.10. Os elementos conversores analógico-digital (ADC – *analog-to-digital converter*) e digital-analógico (DAC – *digital-to-analog converter*) são incluídos na malha para que o sistema microprocessado possa receber sinais digitais a partir do sistema de medição analógico e suas saídas digitais possam ser convertidas para o formato ana-

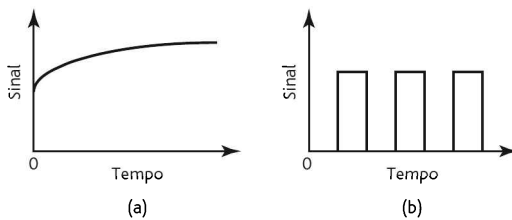


Figura 1.14 Sinais: (a) analógico, (b) digital.

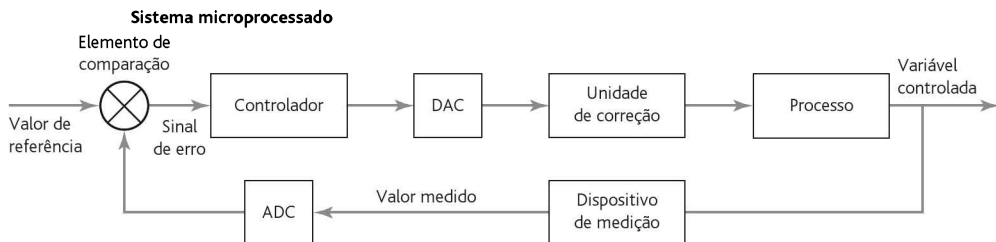


Figura 1.15 Elementos de um sistema de controle em malha fechada.

lógico para acionar a unidade de correção. Este sistema combina as funções de comparação e controle.

Como exemplo de um sistema como este podemos citar um sistema de controle automático da temperatura de uma sala composto de um sensor de temperatura que fornece um sinal analógico, o qual, após um condicionamento adequado para o formato digital, é inserido em um sistema microprocessado, onde é comparado com um valor ajustado, gerando um sinal de erro. Este sinal faz com que o sistema microprocessado gere um sinal digital na saída, o qual, após um condicionamento adequado para o formato analógico equivalente, pode ser usado para controlar um aquecedor e, conseqüentemente, a temperatura da sala. Este sistema pode ser rapidamente programado para diferentes temperaturas em momentos diferentes do dia.

A Figura 1.16 ilustra outro sistema de controle digital da velocidade de um motor. Compare este sistema com o analógico visto na Figura 1.13.

1.5.5 Controladores sequenciais

Existem muitas situações em que o controle é exercido por elementos que ligam ou desligam em determinados momentos ou valores para controlar processos e fornecer uma seqüência de etapas de operações. Por exemplo, após o término da etapa 1, inicia-se a etapa 2. Quando esta é finalizada, inicia-se a etapa 3, e assim por diante.

O termo **controle sequencial** é usado quando as ações de controle são estritamente ordenadas no tempo ou por eventos em seqüência. Este controle pode ser implementado por um circuito elétrico composto de um conjunto de relés ou chaves com cames que são conectadas de forma a se obter a seqüência desejada. Atualmente, estes circuitos de conexão são muito provavelmente substituídos por sistemas microprocessados em que a seqüência é controlada por um software.

Como exemplo de um controle sequencial, considere uma máquina de lavar de uso doméstico. Diversas operações têm que ocorrer em uma seqüência correta. Entre estas pode existir um ciclo de pré-lavagem em que as roupas no cesto são lavadas com água fria, seguida por um ciclo principal de lavagem em que as roupas são lavadas com água quente, e então ocorre um ciclo enxágue em que as roupas são enxaguadas com água fria algumas vezes, seguido por uma rotação do cesto para remover a água da roupa. Cada uma destas operações é constituída por diversas etapas. Por exemplo, faz parte do ciclo de pré-lavagem a abertura de uma válvula para encher de água o cesto até um nível desejado, seguida do fechamento da válvula, do acionamento do motor do cesto para girá-lo por um determinado tempo e do acionamento da bomba para retirar a água do cesto. A seqüência de operações é denominada **programa**, que é uma seqüência de instruções em cada programa definida e “construída” no controlador usado.

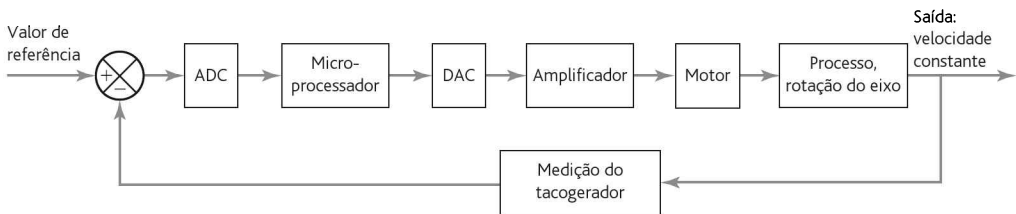


Figura 1.16 Controle de velocidade de um eixo.

A Figura 1.17 mostra o sistema de uma máquina de lavar básica e uma ideia geral de suas partes constituintes. O sistema usado no controlador da máquina é mecânico e é constituído de um conjunto de chaves com cames, ou seja, chaves mecânicas, um sistema que é facilmente ajustável a uma grande variedade de programas.

A Figura 1.18 mostra o princípio básico de uma chave como esta. Quando a máquina é ligada, um pequeno motor elétrico gira vagarosamente sobre seu eixo, produzindo uma intensidade de rotação proporcional ao tempo. Essa rotação gira os cames do controlador de forma que, em cada volta, chaves elétricas são acionadas, ligando circuitos na sequência correta. O contorno de um came determina o momento em que a chave é acionada. Portanto, estes contornos representam a forma com que o programa é especificado e armazenado na máquina. A sequência e os

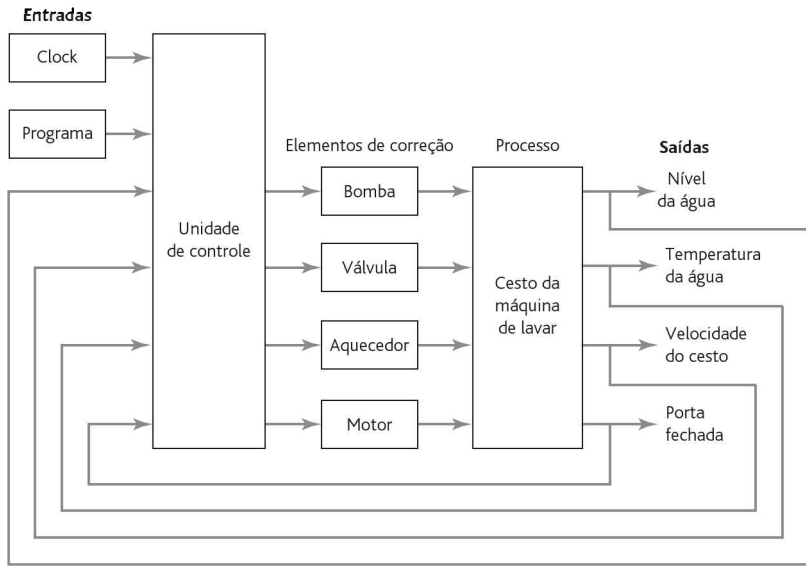


Figura 1.17 Sistema de uma máquina de lavar.

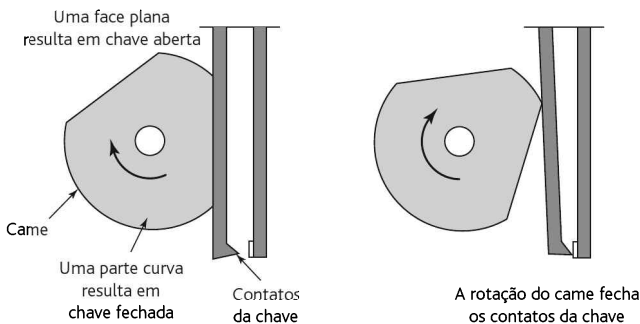


Figura 1.18 Chave acionada por came.

tipos de instruções de um determinado programa de lavagem são determinados pelo conjunto de cames escolhidos. Nas máquinas de lavar modernas, o controlador é um microprocessador, e o programa não é determinado por um conjunto mecânico de cames, mas por um software. Uma máquina de lavar microprocessada pode ser considerada um exemplo de uma abordagem mecatrônica na qual o sistema mecânico está integrado com controles eletrônicos. Como consequência, um enorme sistema mecânico é substituído por um sistema microprocessado muito mais compacto.

Em um ciclo de pré-lavagem, a válvula elétrica é aberta quando uma corrente é fornecida, e desligada quando esta corrente cessa. Esta válvula permite que a água fria entre no cesto durante um tempo determinado pelo perfil do came ou pela saída do microprocessador usada para acionar a chave. Entretanto, visto que o requisito é um nível específico de água no cesto da máquina de lavar, há a necessidade de um outro mecanismo que pare a entrada de água no tanque, durante o tempo permitido, quando ela atingir o nível desejado. Um sensor é usado para gerar um sinal quando o nível da água atingir o nível ajustado, fazendo a saída do microprocessador desligar a corrente na válvula. No caso de uma válvula controlada por came, o sensor atua uma chave que fecha a válvula de admissão de água no cesto da máquina de lavar. Ao final do evento, o microprocessador, ou a rotação dos cames, aciona uma bomba para esvaziar o cesto.

No ciclo de lavagem principal, o microprocessador aciona uma saída quando o ciclo de pré-lavagem do programa é finalizado; no caso do sistema operado por came, este tem um perfil que entra em operação quando o ciclo de pré-lavagem termina. Ele faz circular uma corrente no circuito para abrir a válvula que permite a entrada de água fria no cesto. Este nível é detectado e a água cessa quando o nível desejado é atingido. O microprocessador, ou os cames, fornecem então uma corrente para ativar uma chave pela qual passa uma grande corrente no aquecedor elétrico da água. Um sensor de temperatura é usado para desligar esta corrente quando a temperatura da água atingir o valor ajustado. O microprocessador, ou os cames, ligam então o motor do cesto, o que o faz girar. Isso se mantém por um tempo determinado pelo microprocessador ou perfil do came antes de ser desligado. Em seguida o microprocessador, ou o came, liga a corrente na bomba de descarga para esvaziar a água do cesto.

O ciclo de enxágue da operação passa então a ser executado como uma sequência de sinais para abrir a válvula que permite a entrada de água fria na máquina, para desligá-la, para acionar o motor que gira o cesto, para acionar a bomba que esvazia a água do cesto e repetir essa sequência algumas vezes.

A parte final da operação ocorre quando o microprocessador ou o came liga apenas o motor, numa velocidade maior que a de enxágue, para secar as roupas por centrifugação.

1.6 Controlador lógico programável

Em sistemas muito simples pode haver apenas um microcontrolador, que é um microprocessador com memória integrados em um único chip, programado especificamente para a tarefa de interesse. Um sistema mais adaptável é um **controlador lógico programável (CLP)**. Este é um controlador microprocessado que usa uma memória programável para armazenar instruções e implementar funções de lógica,

seqüência, temporização e aritmética para controlar eventos e pode ser facilmente reprogramado para diferentes tarefas. A Figura 1.19 mostra a ação de controle de um controlador lógico programável, as entradas recebendo sinais, digamos, de chaves que fecham, e o programa usado para determinar como o controlador deve responder às entradas e a saída que ele deve acionar.

Os controladores lógicos programáveis são muito usados em indústrias onde o controle *on/off* (liga/desliga) é desejado. Por exemplo, eles podem ser usados no controle de processos em que um tanque é cheio com líquido e, em seguida, aquecido até uma temperatura específica antes de ser esvaziado. A seqüência do controle desta operação pode ser a seguinte:

- 1 Ligar a bomba para pôr o líquido para dentro do tanque.
- 2 Desligar a bomba quando o detector de nível gerar um sinal que indique que o líquido atingiu o nível desejado.
- 3 Ligar o aquecedor.
- 4 Desligar o aquecedor quando o sensor de temperatura gerar um sinal que indique que a temperatura desejada foi atingida.
- 5 Ligar a bomba para esvaziar o líquido do tanque.
- 6 Desligar a bomba quando um detector de nível gerar um sinal que indique que o tanque está vazio.

Veja o Capítulo 21 para um estudo mais detalhado sobre os controladores lógicos programáveis e exemplos de aplicações.

1.7 Exemplos de sistemas mecatrônicos

A mecatrônica traz consigo a tecnologia de sensores e sistemas de medição, sistemas microprocessados embutidos, atuadores e projetos de engenharia. A seguir são apresentados alguns exemplos de sistemas mecatrônicos que ilustram como sistemas microprocessados são capazes de executar não apenas as tarefas que eram realizadas por sistemas mecânicos, mas também aquelas que não eram facilmente automatizadas pelos mesmos sistemas.

1.7.1 A câmera digital e o foco automático

A maioria das câmeras digitais possui controle automático de foco. Um sistema básico usado em câmeras mais baratas usa malha aberta (Figura 1.20(a)). Quando o fotógrafo pressiona o botão do obturador, um transdutor, localizado na parte frontal

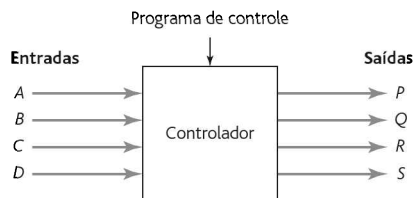


Figura 1.19 Controlador lógico programável.

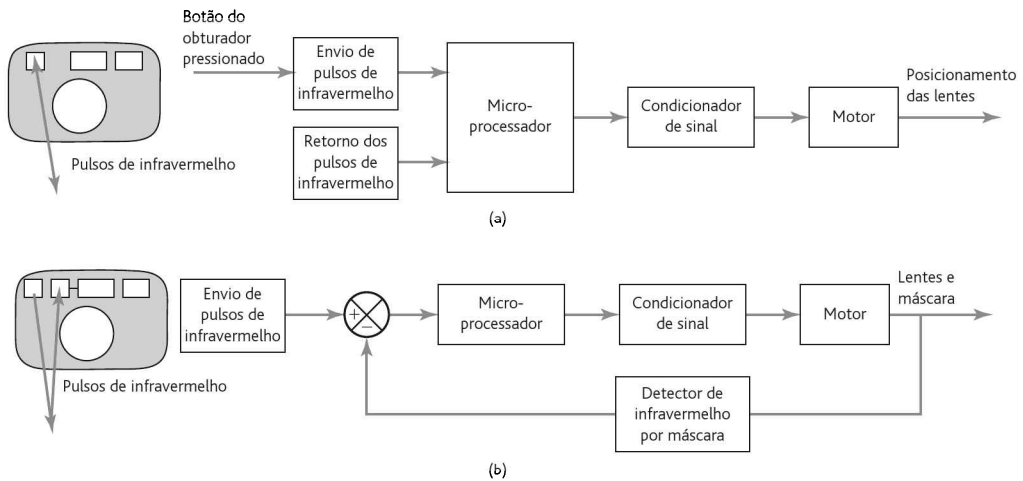


Figura 1.20 Foco automático.

da câmera, envia pulsos de luz infravermelha em direção ao objeto a ser fotografado. Esses pulsos de luz que atingem o objeto são refletidos de volta para a câmera, onde o mesmo transdutor os detecta. Para cada metro de distância do objeto da câmera, o tempo de ida e vinda do pulso é de cerca de 6 ms. A diferença de tempo entre o envio e o retorno dos pulsos é identificada e enviada para o microprocessador. Este tem um conjunto de valores armazenados em sua memória que produz o acionamento de uma saída que faz girar o mecanismo para movimentar as lentes para a posição em que o objeto esteja no foco. Esse tipo de foco automático pode ser usado apenas para distâncias de até aproximadamente 10 m, já que os pulsos de infravermelho que retornam são muito fracos em distâncias maiores. Para distâncias maiores, o microprocessador aciona uma saída que produz o movimento das lentes para o ajuste de infinito.

Um sistema usado em câmeras mais caras usa uma técnica de triangulação (Figura 1.20(b)). Os pulsos de radiação infravermelha são enviados, mas não são detectados pelo mesmo transdutor responsável pela transmissão. Um outro transdutor é usado para isso. Entretanto, esse transdutor tem inicialmente uma máscara sobre ele. O microprocessador ativa a saída que faz com que as lentes se movimentem e, simultaneamente, a máscara se move sobre o transdutor. Esta máscara contém uma fenda que se move na frente do transdutor. O movimento das lentes e da fenda se mantém até os pulsos que retornam consigam passar pela fenda e chegar ao transdutor. Existe uma saída do transdutor que informa ao microprocessador o momento de parar o movimento das lentes, obtendo assim uma imagem do objeto dentro do foco.

1.7.2 Sistema de controle de um motor de combustão

O sistema de controle de um motor de combustão de um veículo é responsável pelo gerenciamento da ignição e a dosagem de combustível no motor. Em um motor de combustão interna de quatro tempos, existem alguns cilindros, cada um com um pistão conectado a um eixo de manivela. Cada conjunto destes executa uma sequência de operações de quatro tempos (Figura 1.21).

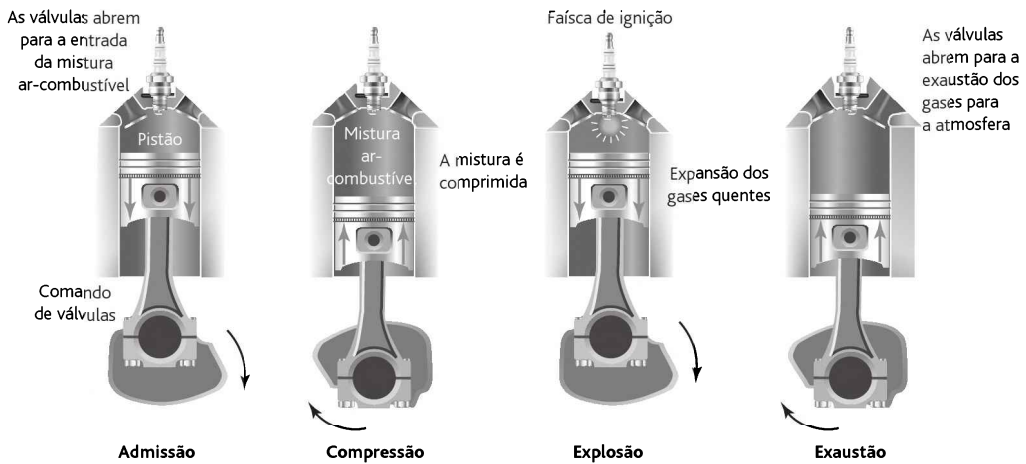


Figura 1.21 Sequência de quatro tempos de um motor de combustão.

Quando o pistão se move para baixo, a válvula abre e uma mistura ar-combustível entra no cilindro. Quando o pistão se move para cima, a válvula é fechada e a mistura ar-combustível é comprimida. Quando o pistão está próximo do topo do cilindro, a vela de ignição gera uma faísca que incendeia a mistura, resultando em uma explosão que expande os gases quentes. Essa expansão provoca o movimento de retorno do pistão e o ciclo se repete. Os pistões dos cilindros são conectados a um mesmo eixo de manivela, e suas explosões ocorrem em diferentes momentos de forma que haja uma força contínua que movimenta o eixo de manivela.

A potência e a velocidade do motor são controladas variando-se o tempo de ignição e a concentração da mistura ar-combustível. Nos motores dos carros modernos isso é feito por um microprocessador. A Figura 1.22 mostra os elementos bá-

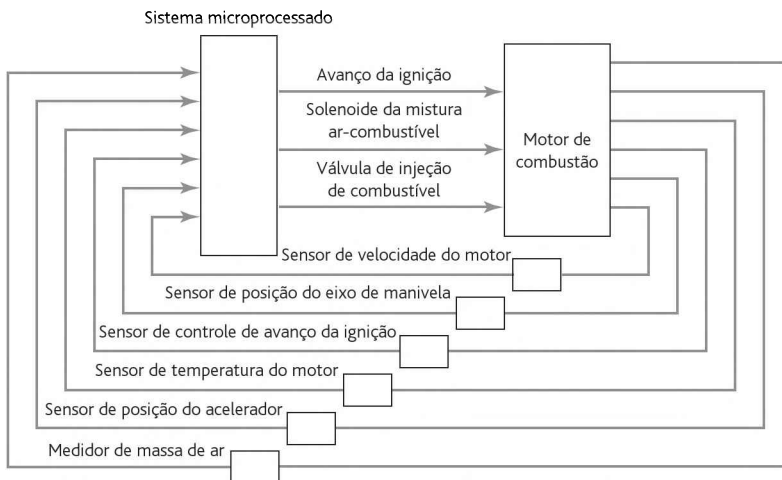


Figura 1.22 Elementos de um sistema de gerenciamento de um motor de combustão.

sicos de um sistema microprocessado. No momento da ignição, o eixo de manivelas aciona um distribuidor que faz contatos elétricos para cada vela de ignição de acordo com a roda dentada. Essa roda dentada gera pulsos para indicar a posição do eixo de manivelas. Então, o microprocessador ajusta a temporização na qual os pulsos de alta tensão são enviados ao distribuidor de forma que eles ocorram nos momentos certos. Para controlar a quantidade da mistura ar-combustível que entra no cilindro durante a admissão, o microprocessador varia o tempo no qual o solenoide é ativado para abrir a válvula de admissão baseado no sinal de entrada da temperatura do motor e da posição da borboleta. A quantidade de combustível a ser injetada no air stream pode ser determinada por uma entrada proveniente de um sensor de massa de ar ou calculada a partir de outras medidas, e então o microprocessador gera uma saída para controlar a válvula de injeção de combustível. Observe que essa descrição do controle de um motor de combustão foi bastante simplificada.

Resumo

A **mecatrônica** é uma integração coordenada e simultânea da engenharia mecânica com a eletrônica e o controle computacional inteligente no projeto e fabricação de produtos e processos. Ela traz consigo as tecnologias da engenharia mecânica, engenharia eletrônica, tecnologia de computadores e engenharia de controle. A mecatrônica proporciona uma oportunidade de uma nova abordagem de problemas de engenharia em que os engenheiros não veem um problema apenas em termos de aplicação de princípios mecânicos, mas consideram-no em termos de um conjunto de tecnologias. A eletrônica, etc. não deve ser vista como um item construído sobre um hardware mecânico. A abordagem da mecatrônica precisa ser adotada desde a fase do projeto.

Os **microprocessadores** são geralmente usados em sistemas mecatrônicos de forma **embarcada**. Um sistema embarcado é aquele que é projetado para controlar uma gama de funções e não é projetado para ser programado pelo usuário final, diferentemente de um computador. Portanto, em um sistema embarcado, o usuário não pode alterar o que o sistema faz adicionando ou substituindo software.

Um **sistema** pode ser idealizado como uma caixa ou um diagrama em bloco que tem uma entrada e uma saída e onde não estamos interessados no que acontece dentro da caixa, mas apenas na relação entre a saída e a entrada.

Para prever como os sistemas se comportam quando as entradas recebem sinais, precisamos criar modelos que relacionem a saída com a entrada para que possamos saber, para uma determinada entrada, como a saída irá variar no tempo.

Em geral, os **sistemas de medida** podem ser considerados como sendo constituídos de três elementos básicos: sensor, condicionador de sinal e display.

Existem duas formas básicas de **sistemas de controle**: em **malha aberta** e em **malha fechada**. No sistema em malha fechada há realimentação, um elemento de comparação, um elemento de controle, um elemento de correção, um elemento de processo e a realimentação que envolve um elemento de medição.

Problemas

- 1.1 Identifique os elementos sensor, condicionador de sinal e display nos sistemas: (a) um termômetro de mercúrio em vidro, (b) medidor de pressão tipo Bourdon.
- 1.2 Explique a diferença entre o controle em malha aberta e o controle em malha fechada.
- 1.3 Identifique os elementos que podem estar presentes em um sistema de controle que envolve um aquecedor elétrico controlado termostaticamente.
- 1.4 O sistema de controle automático da temperatura de um líquido consiste de uma tensão de referência conectada a um amplificador diferencial. Este é conectado a um relé que liga ou desliga a energia elétrica em um aquecedor dentro do líquido. A realimentação negativa é fornecida por um sistema de medição que fornece uma tensão para o amplificador diferencial. Faça o desenho de um diagrama em bloco deste sistema e explique como o sinal de erro é produzido.
- 1.5 Explique a função de um controlador lógico programável.
- 1.6 Explique o que significa controle sequencial e ilustre sua resposta com um exemplo.
- 1.7 Cite as etapas que podem estar presentes no controle sequencial de uma máquina de lavar pratos.
- 1.8 Compare e contraste o projeto tradicional de um relógio com um feito a partir de um projeto mecatrônico que envolve o uso de um microprocessador.
- 1.9 Compare e contraste um sistema de controle de uma central de aquecimento doméstica que usa um termostato bimetalico com a que usa um microprocessador.

2

Sensores e Transdutores

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Descrever o desempenho dos sensores normalmente usados em termos de faixa de operação, *span* (alcance), erro, precisão, sensibilidade, erros de histerese e não linearidade, repetibilidade, estabilidade, zona morta, resolução, impedância de saída, tempo de resposta, constante de tempo, tempo de subida e tempo de acomodação.
- Avaliar sensores usados na medição de deslocamento, posição e proximidade, velocidade e movimento, força, pressão de fluido, vazão de líquido, nível de líquido, temperatura e intensidade luminosa.
- Explicar o problema do repique quando chaves mecânicas são usadas na entrada de dados e como superá-lo.

2.1 Sensores e transdutores

O termo **sensor** é usado para um elemento que produz um sinal relacionado à grandeza medida. Portanto, no caso de, digamos, um elemento de temperatura por resistência elétrica, a grandeza medida é a temperatura, e o sensor transforma a temperatura medida em variação de resistência. O termo **transdutor** é normalmente usado no lugar de sensor. Os transdutores são definidos como elementos que, quando submetidos a uma variação física, respondem com uma outra variação associada. Portanto, sensores são transdutores. Entretanto, um sistema de medição pode usar transdutores, além de sensores, em outras partes do sistema para converter sinais de uma forma em outra. Diz-se que um sensor/transdutor é **analógico** se a sua saída for analógica, variando assim continuamente e tendo tipicamente uma saída cuja amplitude é proporcional à amplitude da variável medida. O termo **digital** é usado se o sistema fornece uma saída de natureza digital, ou seja, uma seqüência de sinais *on/off* que informa um número cujo valor está relacionado à amplitude da variável medida.

Este capítulo aborda os transdutores, e em particular os que são usados como sensores. A terminologia que é usada para especificar as curvas características de desempenho de transdutores é definida e são discutidos exemplos de transdutores normalmente usados em engenharia.

2.1.1 Sensores inteligentes

Alguns sensores também possuem no mesmo encapsulamento o circuito de condicionamento de sinal. Entretanto, este sensor integrado demanda ainda um processamento de dados adicional. É possível ainda encontrar um sensor que combina o circuito de condicionamento do sinal com um microprocessador no mesmo encapsulamento. Este arranjo é denominado **sensor inteligente**. Este sensor pode ter funções como a capacidade de compensar erros aleatórios para se adaptar a mudanças no ambiente, fornecer um cálculo automático da precisão da medida, compensar não linearidade para que a saída seja linear, auto calibração e auto diagnóstico de defeitos.

Estes sensores têm um padrão próprio, IEEE 1451, de forma que, ao representar os sinais neste padrão, eles podem ser usados na forma *plug-and-play*, mantendo a comunicação de dados em uma forma padrão. A informação é armazenada na forma de uma TEDS (*Transducer Electronic Datasheet* – Folha de Dados Eletrônica do Transdutor), geralmente em uma EEPROM, identifica cada dispositivo e fornece os dados de calibração.

2.2 Terminologia para definição de desempenho

Os termos a seguir são usados para definir o desempenho de transdutores e, geralmente, os sistemas de medida como um todo.

1 Faixa de operação e span

A faixa de operação (*range*) de um transdutor define os limites entre os quais a entrada pode variar. O *span* (alcance) é o valor máximo de entrada menos o valor mínimo. Portanto, por exemplo, uma célula de carga usada para medição de forças pode ter uma faixa de operação de 0 a 50 kN e um *span* de 50 kN.

2 Erro

Erro é a diferença entre o resultado da medição e o valor real da grandeza medida:

$$\text{erro} = \text{valor medido} - \text{valor real}$$

Portanto, se um sistema de medição fornece uma indicação de temperatura de 25°C quando a temperatura real é 24°C, então o erro é de +1°C. Se a temperatura real for de 26°C, então o erro será de -1°C. Um sensor pode produzir uma variação de 10,2 Ω quando a variação real deveria ser 10,5 Ω. O erro é -0,3 Ω neste caso.

3 Precisão

Precisão é o quanto o valor indicado pelo sistema de medida pode estar errado. É, portanto, a soma de todos os erros possíveis de ocorrer, assim como a precisão para a qual o transdutor foi calibrado. Um instrumento de medição de temperatura pode, por exemplo, ser especificado como tendo uma precisão de ±2°C. Isto significa que a leitura dada pelo instrumento pode ser um valor dentro de mais ou menos 2°C do valor real. A precisão é geralmente expressa

como uma porcentagem da faixa de operação de saída ou da deflexão de fundo de escala. A porcentagem da deflexão de fundo de escala resulta de quando a saída do sistema de medição era apresentada quase que exclusivamente em uma escala circular ou linear. Um sensor pode, por exemplo, ser especificado como tendo uma precisão de $\pm 5\%$ da faixa de operação de saída. Portanto, se a faixa de operação do sensor era, digamos, de 0 a 200°C , então a leitura dada poderia estar dentro de mais ou menos 10°C do valor real.

4 Sensibilidade

A sensibilidade é a relação que indica o valor da saída por unidade de entrada, ou seja, saída/entrada. Por exemplo, um termômetro baseado em resistência pode ter uma sensibilidade de $0,5 \Omega/^{\circ}\text{C}$. Esse termo também é usado frequentemente para indicar a sensibilidade para entradas diferentes da que está sendo medida, ou seja, variações no ambiente. Portanto, pode ser que a sensibilidade do transdutor à temperatura varie no ambiente ou talvez por flutuações na tensão de alimentação principal. Um transdutor usado para medição de pressão pode ser especificado como tendo uma sensibilidade de temperatura de $\pm 0,1\%$ da leitura por $^{\circ}\text{C}$ de variação na temperatura.

5 Erro de histerese

Os transdutores podem fornecer saídas diferentes para o mesmo valor da grandeza medida conforme o valor tenha sido atingido por meio de uma variação continuamente crescente ou decrescente. Este efeito é denominado de histerese. A Figura 2.1 mostra uma saída com erro de histerese no ponto de máxima diferença na saída para aumento e diminuição de valores.

6 Erro de não linearidade

Para muitos transdutores, é considerada uma relação linear entre a entrada e a saída ao longo da faixa de operação, ou seja, um gráfico da entrada versus a saída é admitido como sendo uma linha reta. Entretanto, poucos transdutores têm uma relação verdadeiramente linear e, portanto, erros correm como resultado de se admitir a linearidade. Este erro é definido como a diferença máxima a partir da linha reta. Alguns métodos são usados para a expressão numérica do erro de não linearidade. As diferenças ocorrem na determinação da linha reta que representa a relação contra o erro que é especificado. Um método é desenhar a linha reta unindo os valores de saída nos pontos extremos da faixa de operação; outro

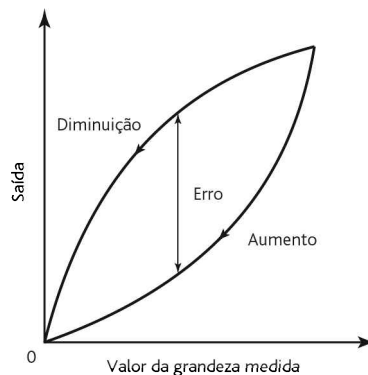


Figura 2.1 Histerese.

método é encontrar a linha reta usando o método de mínimos quadrados para determinar a melhor linha de ajuste quando o erro de todos os valores de dados são considerados igualmente prováveis; um terceiro método de determinar a linha reta é usando o método de mínimos quadrados para determinar a melhor linha de ajuste que passa pelo ponto zero. A Figura 2.2 ilustra estes três métodos e como eles podem afetar o erro de linearidade especificado. Este erro é geralmente especificado como uma porcentagem da faixa de operação de saída. Por exemplo, um transdutor usado para medição de pressão pode ser especificado como tendo um erro de não linearidade de $\pm 0,5\%$ da faixa de operação.

7 Repetibilidade/reprodutibilidade

Os termos repetibilidade e reprodutibilidade de um transdutor são usados para descrever a capacidade dele de fornecer a mesma saída para aplicações repetidas com o mesmo valor de entrada. O erro resultante do mesmo valor de saída não ser obtido para aplicações repetidas é geralmente expresso como uma porcentagem da faixa de operação de saída:

$$\text{repetibilidade} = \frac{\text{valor dado máximo} - \text{valor dado mínimo}}{\text{faixa de operação}} \times 100$$

Um transdutor usado para medição de velocidade angular pode tipicamente ser especificado como tendo uma repetibilidade de $\pm 0,01\%$ da faixa de operação para uma determinada velocidade angular.

8 Estabilidade

A estabilidade de um transdutor é a sua capacidade de fornecer a mesma saída quando usado para medir uma entrada constante ao longo de um período de tempo. O termo **deriva** (*drift*) é frequentemente usado para descrever a variação na saída que ocorre ao longo do tempo. O termo **deriva zero** é usado para as variações que ocorrem na saída quando a entrada é zero.

9 Banda morta/tempo morto

A zona morta ou o espaço morto de um transdutor é a faixa de valores de entrada para a qual não há saída. Por exemplo, o atrito do rolamento em um medidor de vazão que usa um rotor pode não apresentar saída até que a entrada atinja um

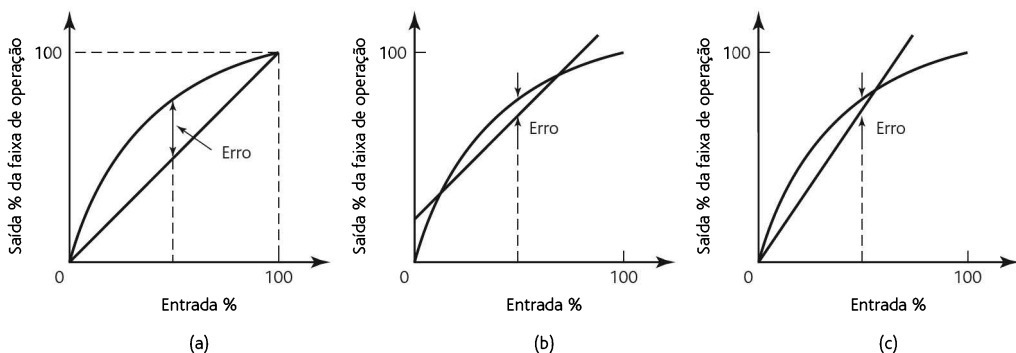


Figura 2.2 Erro de não linearidade usando: (a) valores extremos da faixa de operação, (b) melhor linha reta para todos os valores, (c) melhor linha reta a partir do ponto zero.

limiar de velocidade determinado. O tempo morto é o tempo a partir da aplicação de uma entrada até o instante em que a saída começa a responder.

10 Resolução

Quando a entrada varia continuamente ao longo da faixa de operação, os sinais de saída para alguns sensores podem variar em pequenos degraus. Um potenciômetro de fio é um exemplo deste tipo de sensor em que a saída varia em degraus conforme o cursor do potenciômetro se move de uma espira do fio para a próxima. A resolução é a menor variação no valor de entrada que produz uma variação na saída. No caso de um potenciômetro de fio, a resolução pode ser especificada como, digamos, $0,5^\circ$, ou ainda uma porcentagem da deflexão de fundo de escala. No caso de um sensor com saída digital, a menor variação no sinal de saída é 1 bit. Portanto, para um sensor que fornece uma palavra de dados de N bits, ou seja, um total de 2^N bits, a resolução é normalmente expressa como sendo $1/2^N$.

11 Impedância de saída

Quando um sensor que fornece uma saída elétrica é interfaceado com um circuito eletrônico, é necessário saber a impedância de saída, visto que essa impedância está sendo conectada em série ou em paralelo com a do circuito. A inclusão do sensor pode modificar significativamente o comportamento do sistema no qual ele é conectado. Veja a Seção 6.1.1 para um estudo sobre efeito de carga.

Para ilustrar isto, considere o significado deste termo na seguinte especificação de um transdutor de pressão com *strain gauge*:

Faixas de operação: 70 a 1.000 kPa, 2.000 a 70.000 kPa

Tensão de alimentação: 10 V CC ou ac rms

Faixa de operação total de saída: 40 mV

Não linearidade e histerese: $\pm 0,5\%$ da faixa de operação total de saída

Temperatura de operação: -54°C a $+120^\circ\text{C}$

Deslocamento térmico do zero: 0,030% da faixa de operação total/ $^\circ\text{C}$

A faixa de operação indica que o transdutor pode ser usado para medir pressão entre 70 e 1.000 kPa ou entre 2.000 e 70.000 kPa. Ele necessita de uma fonte de 10 V CC ou ac rms para operar e fornece uma saída de 40 mV quando a pressão na faixa inferior for de 1.000 kPa e na faixa superior for de 70.000 kPa. A não linearidade e a histerese podem produzir erros de até $\pm 0,5\%$ de 1.000, ou seja, ± 5 kPa, na faixa inferior, e $\pm 0,5\%$ de 70.000, isto é, ± 350 kPa, na faixa superior. Este transdutor pode operar entre as temperaturas de -54 e $+120^\circ\text{C}$. Quando a temperatura variar 1°C , para uma entrada zero, a saída do transdutor irá variar 0,030% de 1.000 = 0,3 kPa na faixa inferior e 0,030% de 70.000 = 21 kPa na faixa superior.

2.2.1 Características estáticas e dinâmicas

As **características estáticas** são os valores fornecidos na condição de estado estacionário, ou seja, os valores fornecidos quando o transdutor estiver estabilizado após uma entrada. A terminologia definida antes se refere a este estado. As **características dinâmicas** se referem ao comportamento entre o instante em que a entrada varia e o instante em que a saída do transdutor estabiliza no valor de estado estacionário. As características dinâmicas são expressas em termos da resposta do transdutor a formatos de entrada determinados. Por exemplo, a entrada pode ser em degrau, quando varia rapidamente de zero a um valor constante, ou na forma de

rampa, quando a entrada varia a uma taxa constante, ou ainda na forma de uma senoíde com uma determinada frequência. Portanto, podemos encontrar os seguintes termos (ver o Capítulo 12 para um estudo mais detalhado de sistemas dinâmicos):

1 *Tempo de resposta*

Este é o tempo decorrido após uma entrada constante, uma entrada em degrau, ser aplicada ao transdutor até o instante em que o transdutor forneça uma saída correspondente a uma percentagem especificada, por exemplo, 95%, do valor de estado estacionário da saída (Figura 2.3). Por exemplo, se um termômetro de vidro à base de mercúrio for colocado em um líquido quente, pode decorrer um tempo apreciável, talvez 100 s ou mais, antes do termômetro indicar 95% da temperatura real do líquido.

2 *Constante de tempo*

Este é o tempo de resposta que corresponde a 63,2% da saída em estado estacionário. Um termopar no ar pode ter uma constante de tempo por volta de 40 a 100 s. A constante de tempo é uma medida da inércia do sensor e indica com que rapidez ele reage a variações na entrada: quanto maior a constante de tempo, mais lenta a reação a variação no sinal de entrada. Veja a Seção 12.3.4 para uma análise matemática da constante de tempo em termos do comportamento de um sistema quando submetido a uma entrada em degrau.

3 *Tempo de subida*

Este é o tempo que a saída gasta para variar de forma ascendente correspondente a um percentual da saída em estado estacionário. Geralmente o tempo de subida se refere ao tempo gasto para a saída ir de 10% do valor de estado estacionário a 90 ou 95% deste mesmo valor.

4 *Tempo de acomodação*

Este é o tempo que a saída gasta para se ajustar dentro de alguma percentagem, por exemplo, 2%, do valor de estado estacionário.

Para ilustrar isto, considere o gráfico na Figura 2.4, que indica como a leitura de um instrumento varia com o tempo, sendo aquele um termômetro inserido no líquido no instante $t = 0$. O valor de estado estacionário é 55°C e, portanto, como 95% de 55 é $52,25^{\circ}\text{C}$, o tempo de resposta de 95% é cerca de 228 s.

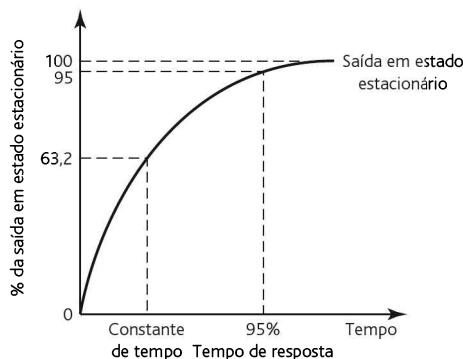


Figura 2.3 Resposta a uma entrada em degrau.

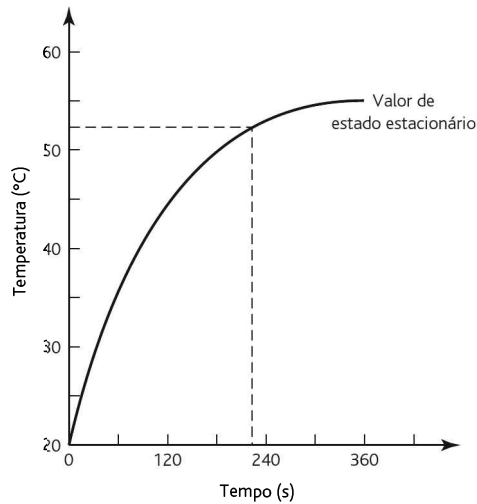


Figura 2.4 Termômetro dentro de um líquido.

As seções a seguir fornecem exemplos de transdutores agrupados conforme o tipo de medida que eles realizam. As medidas consideradas são aquelas frequentemente encontradas na engenharia mecânica, isto é: deslocamento, proximidade, velocidade, força, pressão, vazão de fluido, nível de líquido, temperatura e intensidade luminosa.

2.3 Deslocamento, posição e proximidade

Sensores de **deslocamento** são destinados a medidas da quantidade de movimento de algum objeto; sensores de **posição** são destinados a determinar a posição de algum objeto em relação a um ponto de referência. Sensores de **proximidade** são um tipo de sensor de posição usados para determinar quando um objeto se moveu dentro de uma determinada distância crítica do sensor. Estes são essencialmente dispositivos com saída *on/off*.

Os sensores de deslocamento e posição podem ser agrupados em dois tipos básicos: sensores de contato, nos quais o objeto medido tem algum contato mecânico com o sensor, ou sensores sem contato, nos quais não há contato físico entre o objeto medido e o sensor. No caso dos métodos de deslocamento linear que envolvem contato, há geralmente uma haste em contato direto com o objeto monitorado. O deslocamento desta haste é então monitorado pelo sensor. O movimento de uma haste pode ser usado para alterar uma tensão elétrica, resistência, capacitância ou indutância mútua. No caso de métodos de deslocamento angular que envolvem uma conexão mecânica, a rotação de um eixo pode acionar diretamente, através de engrenagens, a rotação do elemento transdutor. Os sensores sem contato físico podem, com a presença do objeto medido nas proximidades do sensor, sentir uma variação, por exemplo, na pressão do ar, ou então uma variação na indutância ou capacitância. Os exemplos a seguir mostram sensores de deslocamento de uso comum.

2.3.1 Sensor baseado em potenciômetro

Um **potenciômetro** consiste em um elemento de resistência com um contato deslizante que pode ser movimentado ao longo do comprimento deste elemento. Estes dispositivos podem ser usados em deslocamento linear ou rotativo, sendo o deslocamento convertido em diferença de potencial. O potenciômetro rotativo consiste de uma trilha com um fio enrolado ou com um filme plástico condutor sobre a qual um contato rotativo deslizante pode se deslocar (Figura 2.5). A trilha pode ter uma única volta ou ter uma forma helicoidal. Com uma tensão de entrada V_s constante, entre os terminais 1 e 3, a tensão de saída V_o entre os terminais 2 e 3 é uma fração da tensão de entrada que depende da razão entre a resistência R_{23} , entre os terminais 2 e 3, comparada com a resistência total R_{13} , entre os terminais 1 e 3, ou seja, $V_o/V_s = R_{23}/R_{13}$. Se a trilha tiver uma resistência constante por unidade de comprimento, ou seja, por unidade de ângulo, então a saída é proporcional ao ângulo em que o cursor é girado. Portanto, um deslocamento angular pode ser convertido em uma diferença de potencial.

No caso de um trilha feita com fio enrolado, o movimento do cursor de uma espira para a próxima faz variar a tensão em degraus, sendo que cada degrau corresponde ao movimento de uma espira. Se o potenciômetro tiver N espiras, então a resolução, em porcentagem, é $100/N$. Assim, a resolução de uma trilha de fio é limitada pelo diâmetro do fio usado, o que varia tipicamente de cerca de 1,5 mm, para uma trilha de menor resolução, a 0,5 mm, para uma de maior resolução. Os erros devidos à não linearidade da trilha variam de menos de 0,1% a cerca de 1%. A resistência da trilha tende a variar de cerca de 20 Ω a 200 k Ω . O plástico condutivo tem, idealmente, uma resolução infinita, erros devido a não linearidade da trilha na ordem de 0,05%, e valores de resistência de cerca de 500 a 80 k Ω . Este material tem um coeficiente de temperatura da resistência maior que o do fio, portanto, as variações de temperatura têm um maior efeito na precisão.

Um efeito importante a ser considerado no potenciômetro é o da carga R_L conectada na saída. A diferença de potencial na carga, V_L , somente será diretamente proporcional a V_o se a resistência da carga for infinita. Entretanto, para cargas finitas o efeito dela é transformar o que era uma relação linear entre a tensão de saída e o ângulo em uma relação não linear. A resistência R_L está em paralelo com a fração x da resistência do potenciômetro, R_p . Esta resistência combinada é $R_L x R_p / (R_L + x R_p)$. A resistência total “vista” pela fonte de alimentação é então

$$\text{resistência total} = R_p(1 - x) + R_L x R_p / (R_L + x R_p)$$

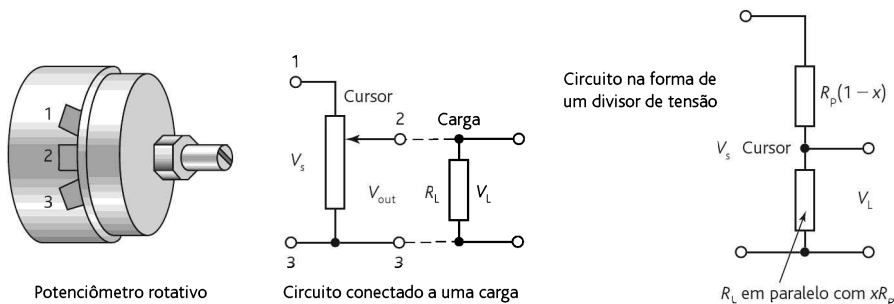


Figura 2.5 Potenciômetro rotativo.

O circuito é um divisor de tensão em que a tensão na carga corresponde à fração da resistência nos terminais da carga pela resistência total na qual é aplicada a tensão de alimentação:

$$\begin{aligned} \frac{V_L}{V_s} &= \frac{xR_L R_p / (R_L + xR_p)}{R_p(1 - x) + xR_L R_p / (R_L + xR_p)} \\ &= \frac{x}{(R_p/R_L)x(1 - x) + 1} \end{aligned}$$

Se a carga tiver resistência infinita, então temos $V_L = xV_s$. Assim, o erro introduzido pela carga que tem uma resistência finita é

$$\begin{aligned} \text{erro} &= xV_s - V_L = xV_s - \frac{xV_s}{(R_p/R_L)x(1 - x) + 1} \\ &= V_s \frac{R_p}{R_L} (x^2 - x^3) \end{aligned}$$

Para ilustrar isto, considere o erro de não linearidade em um potenciômetro de resistência de 500Ω , para um deslocamento de metade do percurso máximo do cursor, considerando uma carga de resistência de $10 \text{ k}\Omega$. A tensão da fonte de alimentação é de 4 V . Usando a equação deduzida antes, temos

$$\text{erro} = 4 \times \frac{500}{10000} (0,5^2 - 0,5^3) = 0,025 \text{ V}$$

Em termos de percentagem da faixa de operação total, este valor representa $0,625\%$.

Os potenciômetros são usados como sensores nos sistemas eletrônicos de veículos, como por exemplo no pedal do acelerador e no afogador.

2.3.2 Elementos com *strain gauge*

A resistência elétrica do tipo *strain gauge* (extensômetro) (Figura 2.6) pode ser um fio metálico, tira metálica, ou uma tira de material semicondutor na forma de uma pequena placa e pode ser fixada em uma superfície como se fosse um selo de cartas.

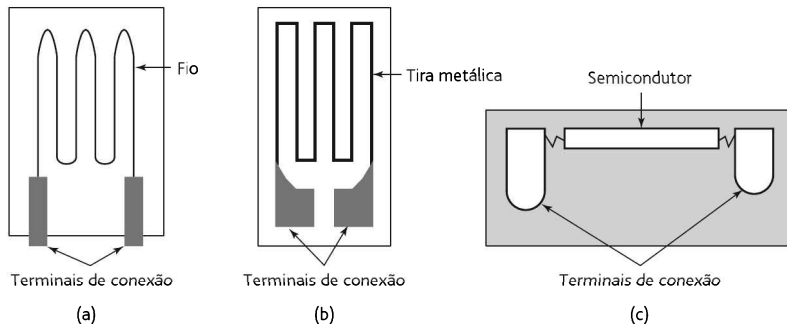


Figura 2.6 *Strain gauges*: (a) fio metálico, (b) tira metálica, (c) semicondutor.

Quando submetida a um esforço, sua resistência R varia, sendo a fração de variação na resistência, $\Delta R/R$, proporcional ao esforço ε , ou seja,

$$\frac{\Delta R}{R} = G\varepsilon$$

em que G , a constante de proporcionalidade, é denominada de fator *gauge* (*gauge factor*) ou fator do extensômetro.

Como o esforço é a razão (variação no comprimento/comprimento original), então a variação de resistência de um *strain gauge* é uma medida da variação no comprimento do elemento no qual o *strain gauge* está fixado. O fator *gauge* do *strain gauge* de fio metálico ou tira metálica para os metais geralmente usados é de cerca de 2,0, e as resistências são geralmente da ordem de 100 Ω . Os *strain gauges* de semicondutor silício do tipo n ou p possuem fatores *gauge* por volta de +100 ou mais para o silício tipo p e -100 ou mais para silício tipo n, e resistências da ordem de 1.000 a 5.000 Ω . O fator *gauge* é normalmente informado pelo fabricante dos *strain gauges* a partir de uma calibração feita de uma amostra de *strain gauges* extraída de um lote. A calibração envolve submeter a amostra de *strain gauges* a esforços conhecidos e medir as variações de resistência. Um problema que existe com todos os *strain gauges* é que sua resistência varia não apenas sob esforço, mas com a variação da temperatura. São usados mecanismos para eliminar o efeito da temperatura, os quais são abordados no Capítulo 3. Os *strain gauges* de semicondutor são mais sensíveis à temperatura do que os de metal.

Para ilustrar esses dados, considere um *strain gauge* com resistência de 100 Ω e um fator *gauge* de 2,0. Qual é a variação na resistência deste quando submetido a um esforço de 0,001? A variação fracional na resistência é igual ao fator *gauge* multiplicado pelo esforço, assim

$$\text{variação na resistência} = 2,0 \times 0,001 \times 100 = 0,2 \Omega$$

Um tipo de sensor de deslocamento usa *strain gauge* fixo em elementos flexíveis na forma de viga em balanço (Figura 2.7(a)), anéis (Figura 2.7(b)) ou em formas de U (Figura 2.7(c)). Quando o elemento flexível é dobrado ou deformado em função de uma força aplicada através de um ponto de contato que se move, então os *strain gauges* de resistência elétrica montados sobre o elemento são estendidos ou comprimidos, provocando assim uma variação de resistência que pode ser monitorada. A variação na resistência é, portanto, uma medida do deslocamento ou deformação do elemento flexível. Essas configurações são normalmente usadas para deslocamentos lineares da ordem de 1 a 30 mm, com erro de não linearidade de cerca de $\pm 1\%$ da faixa de operação total.

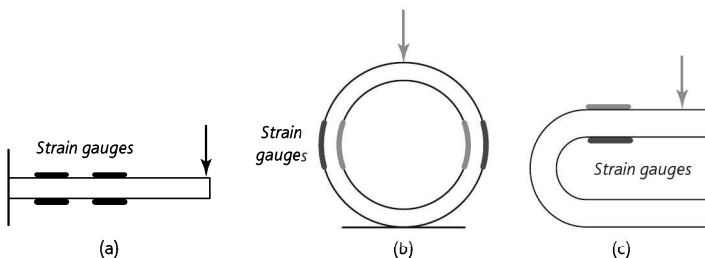


Figura 2.7 Elementos com *strain gauges* montados.

2.3.3 Elemento capacitivo

A capacitância C de um capacitor de placas paralelas é dada por

$$C = \frac{\epsilon_r \epsilon_0 A}{d}$$

em que ϵ_r é a permissividade relativa do dielétrico entre as placas, ϵ_0 é uma constante denominada de permissividade do espaço livre, A é a área de sobreposição das duas placas e d é a distância de separação delas. Os sensores capacitivos usados para monitorar deslocamentos lineares podem ter as formas mostradas na Figura 2.8. Em (a) uma das placas se move em função do deslocamento de forma que a separação entre as placas é alterada; em (b) o deslocamento provoca uma alteração na área de sobreposição das placas; em (c) o deslocamento faz com que o dielétrico entre as placas seja movido.

No caso de um deslocamento que faz variar a distância de separação entre as placas (Figura 2.8(a)), se a distância d aumentar em função de um deslocamento x , então a capacitância passa a ser

$$C - \Delta C = \frac{\epsilon_0 \epsilon_r A}{d + x}$$

Portanto, a variação na capacitância, ΔC , como uma fração da capacitância inicial é dada por

$$\frac{\Delta C}{C} = -\frac{d}{d + x} - 1 = -\frac{x/d}{1 + (x/d)}$$

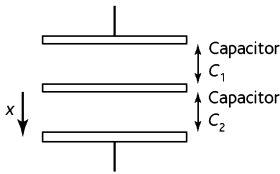


Figura 2.9 Sensor push-pull.

Portanto, há uma relação não linear entre a variação na capacitância, ΔC , e o deslocamento x . Essa não linearidade pode ser superada usando-se o que denominamos de **sensor de deslocamento push-pull** (Figura 2.9). Este tem três placas, sendo um capacitor formado pelo par superior e o outro, pelo par inferior. O deslocamento move a placa central entre as duas outras. O resultado, por exemplo, da placa central se mover para baixo é aumentar a distância entre as placas do capacitor superior e diminuir a do inferior. Assim, temos

$$C_1 = \frac{\epsilon_0 \epsilon_r A}{d + x}$$

$$C_2 = \frac{\epsilon_0 \epsilon_r A}{d - x}$$

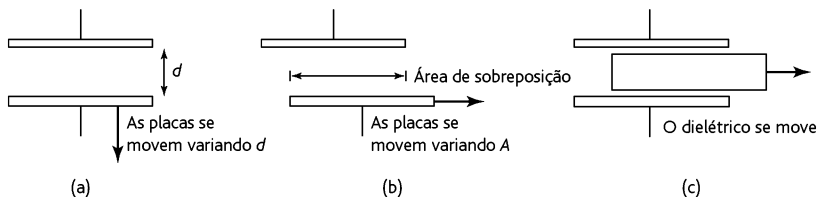


Figura 2.8 Formas do elemento sensor capacitivo.

Quando C_1 está em um ramo de uma ponte CA e C_2 , em outro, resulta em um desequilíbrio da tensão que é proporcional a x . Este sensor geralmente é usado para monitorar deslocamentos de alguns milímetros a centenas de milímetros. A não linearidade e a histerese estão em torno de $\pm 0,01\%$ da faixa de operação total.

Outro tipo de sensor de proximidade capacitivo consiste de um capacitor formado por uma placa constituída por uma sonda e a outra placa formada pelo próprio objeto, que tem que ser metálico e aterrado (Figura 2.10). À medida que o objeto se aproxima, a distância entre as placas do capacitor é alterada, tornando-se significativa e detectável quando o objeto está próximo da sonda.

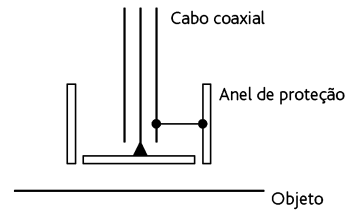


Figura 2.10 Sensor de proximidade capacitivo.

2.3.4 Transformadores diferenciais

Um transformador diferencial linear variável, geralmente conhecido pela abreviação LVDT (*Linear Variable Differential Transformer*), consiste de três bobinas simetricamente espaçadas ao longo de um tubo isolado (Figura 2.11). A bobina central é o primário e as outras, que são idênticas, representam o secundário, sendo conectadas em série de forma que as saídas são opostas entre si. Um núcleo magnético se move através do tubo central em função do deslocamento monitorado.

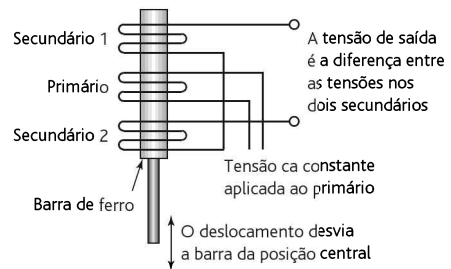


Figura 2.11 LVDT.

Quando há uma tensão alternada na entrada da bobina do primário, fems (forças eletromotrizes) são induzidas nas bobinas do secundário. Com o núcleo magnético na posição central, a quantidade de material magnético em cada uma das bobinas do secundário é igual. Portanto, as fems induzidas em cada bobina são iguais. Visto que elas são conectadas com as saídas opostas entre si, a tensão de saída resultante é zero.

Entretanto, quando o núcleo é deslocado da posição central, passa a existir uma quantidade maior do material magnético do núcleo dentro de uma bobina em comparação com a outra, por exemplo, mais na bobina do secundário 2 do que na do 1. O resultado é que a fem induzida em uma bobina é maior do que na outra. Passa a existir então uma tensão de saída a partir das duas bobinas. Visto que uma deslocamento maior significa mais núcleo em uma bobina do que na outra, a saída, a diferença entre as duas fems, aumenta quanto maior for o deslocamento monitorado (Figura 2.12).

A FEM induzida na bobina do secundário por uma corrente variável i no primário é dada por

$$e = M \frac{di}{dt}$$

em que M é a indutância mútua, sendo seu valor dependente do número de espiras nas bobinas e do núcleo ferromagnético. Portanto, para uma corrente senoidal de entrada $i = I \text{ sen } \omega t$ na

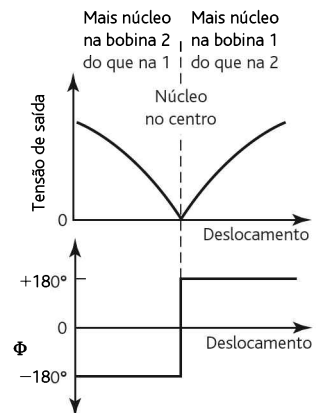


Figura 2.12 Saída do LVDT.

bobina do primário, as fems induzidas nas bobinas 1 e 2 do secundário podem ser representadas por

$$v_1 = k_1 \text{sen}(\omega t - \phi) \text{ e } v_2 = k_2 \text{sen}(\omega t - \phi)$$

em que os valores de k_1 , k_2 e ϕ dependem do grau de acoplamento entre as bobinas do primário e do secundário para uma determinada posição do núcleo. ϕ é a diferença de fase entre as tensões alternadas no primário e no secundário. Devido às duas saídas estarem em série, a saída é a diferença entre elas:

$$\text{tensão de saída} = v_1 - v_2 = (k_1 - k_2) \text{sen}(\omega t - \phi)$$

Quando o núcleo está igualmente nas duas bobinas, k_1 é igual a k_2 , e assim a tensão de saída é zero. Quando o núcleo está mais na bobina 1 do que na 2, temos $k_1 > k_2$ e

$$\text{tensão de saída} = (k_1 - k_2) \text{sen}(\omega t - \phi)$$

Quando o núcleo está mais na bobina 2 do que na 1, temos $k_1 < k_2$. Como consequência disto, existe uma variação de fase de 180° na saída quando o núcleo se move mais para a bobina 1 do que para a 2. Portanto,

$$\begin{aligned} \text{tensão de saída} &= -(k_1 - k_2) \text{sen}(\omega t - \phi) \\ &= (k_2 - k_1) \text{sen}[\omega t + (\pi - \phi)] \end{aligned}$$

A Figura 2.12 mostra como a amplitude e a fase da saída variam com o deslocamento do núcleo.

Com a saída dessa forma, a mesma amplitude da tensão de saída é produzida em dois deslocamentos diferentes. Para proporcionar uma tensão de saída única para cada valor de deslocamento, precisamos distinguir os casos em que as amplitudes são iguais, porém com uma diferença de fase de 180° . Para que cada deslocamento tenha um valor único, é usado um demodulador sensível à fase juntamente com um filtro passa-baixas (Figura 2.13). Circuitos como este são fornecidos na forma de circuitos integrados (CIs).

Os LVDTs têm tipicamente faixas de operação desde cerca de ± 2 a ± 400 mm com erros de não linearidade de cerca de $\pm 0,25\%$. Esses dispositivos são muito usados como transdutores primários no monitoramento de deslocamentos. A extremidade livre do núcleo pode ser acionada por mola para contato dele com a superfície monitorada, ou aparafusada por conexão mecânica. Eles também são usados como transdutores secundários em medidas de força, peso e pressão; essas variáveis são transformadas em deslocamentos que podem ser monitorados por LVDTs.

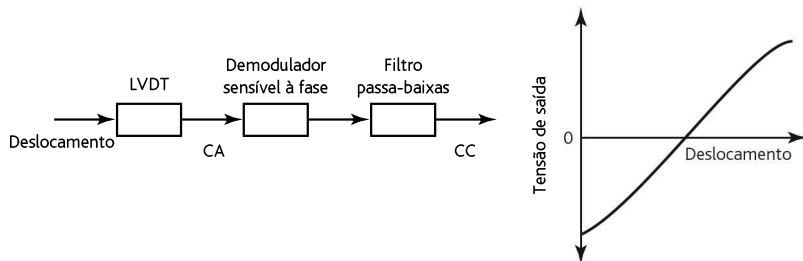


Figura 2.13 Saída CC de um LVDT.

Um transformador diferencial rotacional variável (RVDT – *Rotary Variable Differential Transformer*) pode ser usado para medidas de rotação (Figura 2.14); ele opera com base no mesmo princípio do LVDT. O núcleo é uma peça de material magnético na forma de uma cardioide, e sua rotação faz com que uma parte maior da peça passe dentro de uma bobina do secundário do que em outra. A faixa de operação é tipicamente de $\pm 40^\circ$ com um erro de linearidade de cerca de $\pm 0,5\%$ da faixa de operação.

2.3.5 Sensores de proximidade por corrente parasita

Se uma bobina for alimentada com uma corrente alternada, um campo magnético variável é produzido. Se houver um objeto metálico nas proximidades deste campo magnético, então serão induzidas correntes parasitas nele. Estas correntes, por sua vez, produzem um campo magnético. Este campo distorce o campo magnético responsável pela sua origem. Como resultado, varia a impedância da bobina e, da mesma forma, a amplitude da corrente alternada. Esta variação pode, em um determinado nível, ser usada para acionar uma chave. A Figura 2.15 mostra a forma básica deste sensor; ele é usado para detecção de materiais condutivos, porém não magnéticos. Estes sensores possuem as vantagens de serem relativamente baratos, de pequeno tamanho, de alta confiabilidade e podem ter uma alta sensibilidade a pequenos deslocamentos.

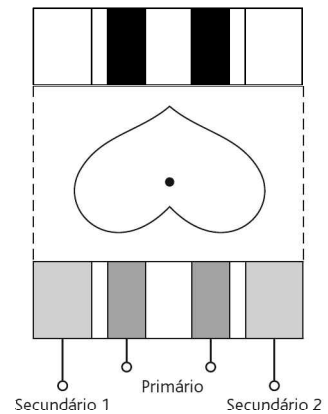


Figura 2.14 RVDT.

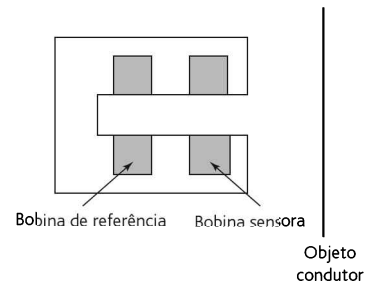


Figura 2.15 Sensor baseado em corrente parasita.

2.3.6 Chave de proximidade indutiva

Este sensor consiste de uma bobina em torno de um núcleo. Quando a extremidade do núcleo se aproxima de um objeto metálico, sua indutância varia. Essa variação pode ser monitorada pelo seu efeito sobre um circuito ressonante, e é usada para acionar uma chave. Ele pode ser usado apenas para a detecção de objetos metálicos, sendo melhor com metais ferrosos.

2.3.7 Encoders ópticos

Um **encoder** é um dispositivo que fornece uma saída digital como resultado de um deslocamento linear ou angular. Os encoders de posição podem ser agrupados em duas categorias: **encoders incrementais** que detectam variações na rotação a partir de uma posição dada, e **encoders absolutos** que fornece a posição angular real.

A Figura 2.16(a) mostra uma forma básica de um encoder incremental usado para medida de deslocamento angular. Um feixe de luz passa através de fendas em um disco e é detectado por um sensor de luz adequado. Quando o disco é girado, uma saída pulsada é produzida pelo sensor, sendo que o número de pulsos é proporcional ao ângulo de giro do disco. Portanto, a posição angular do disco, e portanto a rotação do eixo, pode ser determinada pelo número de pulsos produzidos a partir da

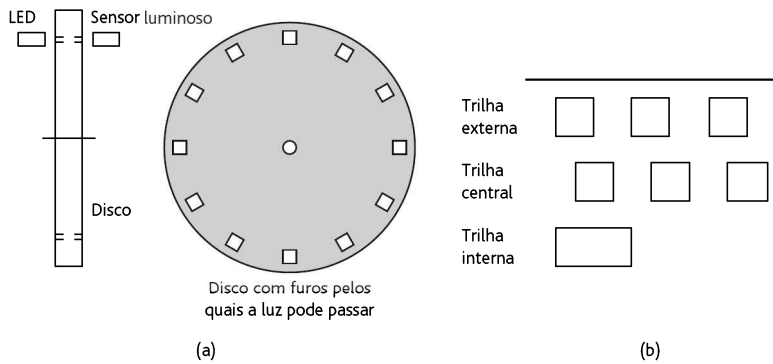


Figura 2.16 Encoder incremental: (a) princípio básico, (b) trilhas concêntricas.

posição de referência. Na prática, são usadas três trilhas concêntricas com três sensores (Figura 2.16(b)). A trilha interna tem apenas um furo e é usada para identificar a posição *home* (de referência) do disco. As outras duas trilhas têm uma série de furos igualmente espaçados em torno de todo o disco, sendo os furos da trilha central deslocados em relação aos furos da trilha externa por uma distância que é metade da largura de um furo. Esse deslocamento possibilita a determinação do sentido de rotação. No sentido horário, os pulsos na trilha externa estão adiantados em relação ao da trilha central; no sentido anti-horário, eles estão atrasados. A resolução é determinada pelo número de furos no disco. Com 60 furos em uma revolução, temos uma resolução de $360/60 = 6^\circ$, já que uma revolução corresponde a 360° .

A Figura 2.17 mostra a forma básica de um encoder absoluto usado para medição de deslocamento angular. Este fornece uma saída no formato de um número binário com uma quantidade específica de dígitos, e cada número representa uma determinada posição angular. O disco girante tem três círculos concêntricos e três sensores para detectar os pulsos de luz. Os furos são posicionados de forma que a saída sequencial dos sensores é um número em código binário. Os encoders comuns têm de 10 a 12 trilhas. O número de bits no número binário é igual ao número de trilhas. Portanto, 10 trilhas corresponde a 10 bits, e o número de posições que podem ser detectadas é 2^{10} , ou seja, 1024, o que resulta em uma resolução de $360/1024 = 0,35^\circ$.

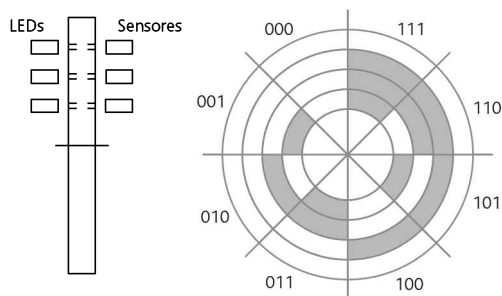


Figura 2.17 Encoder de 3 bits.

Geralmente, o formato normal de um código binário não é usado porque uma variação de um número binário para o próximo pode resultar em mudanças em mais de um bit, por exemplo, na mudança de 101 para 110 temos mudança em dois bits, e se houver um desalinhamento, um dos bits muda uma fração de tempo antes do outro, gerando momentaneamente um número binário intermediário, podendo assim levar o sistema a uma contagem falsa. Para evitar isto, geralmente o **código Gray** é usado (veja o Apêndice B). Neste código, somente um bit muda cada vez que a contagem passa de um número para o próximo. A Figura 2.18 mostra a configuração das trilhas para representações nos códigos binário e Gray.

	Binário normal	Código Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

Figura 2.18 Códigos binário e Gray.

Os encoders ópticos, por exemplo, o HEDS-5000 da Hewlett Packard (HP), são fornecidos para serem montados nos eixos e contêm uma fonte de luz com LED e um disco codificado. São disponibilizados também circuitos integrados para decodificar o encoder e converter do código Gray para uma saída binária apropriada para um microprocessador. No caso de um encoder absoluto com sete trilhas no disco codificado, cada trilha fornece um dos bits do número binário, e assim temos 2^7 posições especificadas, ou seja, 128. Com oito trilhas temos 2^8 posições, ou seja, 256.

2.3.8 Sensores pneumáticos

Os sensores pneumáticos fazem uso de ar comprimido, sendo que o deslocamento ou a proximidade de um objeto é transformado em variação na pressão do ar. A Figura 2.19 mostra a forma básica deste sensor. O ar de baixa pressão flui através de um orifício na parte frontal do sensor. O fluxo deste ar, na ausência de qualquer objeto próximo, faz reduzir a pressão nas proximidades do orifício de saída do sensor. Entretanto, se existir um objeto próximo, o ar não poderá fluir com facilidade, e o resultado será um aumento de pressão no orifício de saída do sensor. Deste modo, a pressão de saída do sensor depende da proximidade do objeto.

Estes sensores são usados para medidas de deslocamentos de frações de milímetros em faixas típicas de cerca de 3 a 12 mm.

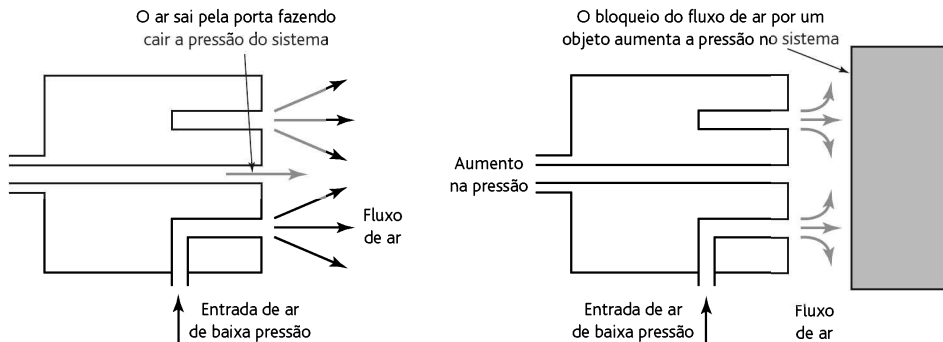


Figura 2.19 Sensor de proximidade pneumático.

2.3.9 Chaves fim de curso

Existem alguns tipos de chaves que podem ser ativadas pela presença de um objeto originando um sensor de proximidade com uma saída que está *on* ou *off*.

Uma **chave fim de curso** (*microswitch*) é uma pequena chave elétrica que necessita de contato físico e uma pequena força para fechar os contatos. Por exemplo, para determinar a presença de um item em uma correia transportadora, essa chave pode ser acionada pelo peso do item que faz baixar a correia e, por consequência, uma plataforma sob mola abaixo do item. O movimento desta plataforma aciona a chave. A Figura 2.20 mostra exemplos de como esse tipo de chave pode ser acionado.

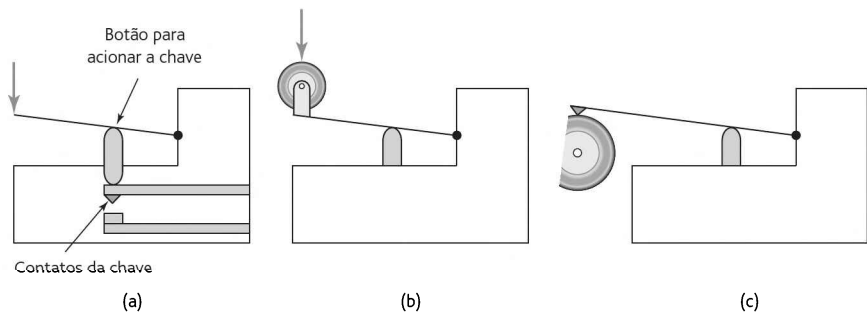


Figura 2.20 Chave acionada por (a) nível, (b) rolete, (c) came.

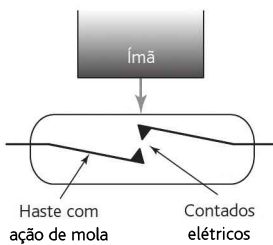


Figura 2.21 Reed switch (chave magnética).

A Figura 2.21 mostra a forma básica de uma *reed switch* (chave magnética). Ela consiste de dois contatos de chave magnéticos fechados hermeticamente em um tubo de vidro. Quando um ímã se aproxima da chave, as palhetas magnéticas são atraídas uma contra a outra, fechando os contatos dela. Esta é uma chave de proximidade sem contato físico. Esse tipo de chave é muito usado na verificação de porta fechada. Ele também é usado em dispositivos como tacômetros, que envolvem a rotação de uma roda dentada em que os dentes passam na frente de um *reed switch*. Se cada um dos dentes da roda tiver um ímã fixado, cada vez que um deles passar próximo da chave ela fecha os contatos, produzindo um pulso de tensão/corrente no circuito elétrico associado.

Os **dispositivos fotossensíveis** podem ser usados para detectar a presença de um objeto opaco pela interrupção de um feixe de luz ou de radiação infravermelha, que ocorre quando o objeto corta o feixe de luz ou por detecção da luz refletida pelo objeto (Figura 2.22).

2.3.10 Sensores de efeito Hall

Quando um feixe de partículas carregadas passa através de um campo magnético, forças atuam sobre essas partículas e o feixe é defletido do percurso em linha reta seguido antes. Uma corrente fluindo em um condutor é semelhante a um feixe de cargas que se move e, desta forma, pode ser defletido por um campo magnético. Esse

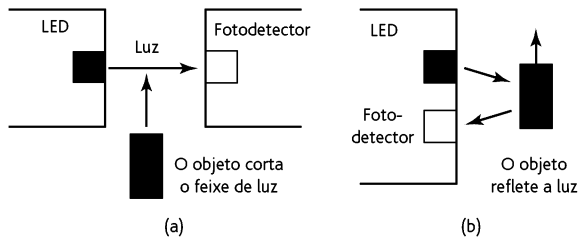


Figura 2.22 Uso de sensores fotoelétricos para detectar objetos por (a) corte do feixe de luz pelo objeto, (b) reflexão da luz pelo objeto.

efeito foi descoberto por E. R. Hall em 1879 e é denominado **efeito Hall**. Considere elétrons se movendo em uma placa condutora com um campo magnético aplicado de forma perpendicular ao plano da placa (Figura 2.23). Como consequência deste campo magnético, os elétrons em movimento são defletidos para um lado da placa e, conseqüentemente, este lado se torna carregado negativamente, enquanto que o lado oposto se torna carregado positivamente, visto que os elétrons são direcionados para fora deste lado. Esta separação de cargas produz um campo elétrico no material. A separação de cargas continua até que as forças sobre as partículas carregadas, provenientes do campo elétrico, sejam equilibradas pelas forças produzidas pelo campo magnético. O resultado é uma diferença de potencial transversal, V , dada por

$$V = K_H \frac{BI}{t}$$

onde B é a densidade de fluxo magnético perpendicular à placa, I é a corrente através dela, t é a espessura da placa e K_H é uma constante denominada **coeficiente Hall**. Portanto, se uma fonte de corrente constante for usada com um determinado sensor, a tensão Hall é uma medida da densidade de fluxo magnético.

Os sensores por efeito Hall são geralmente fornecidos na forma de um circuito integrado contendo o circuito de processamento de sinal necessário. Existem dois tipos básicos deste sensor: o linear, em que a saída varia de forma razoavelmente linear em relação à densidade de fluxo magnético (Figura 2.24(a)); e o de limiar, em que a saída apresenta uma queda acentuada em uma determinada densidade de fluxo magnético (Figura 2.24(b)). O sensor de efeito Hall com saída linear 634SS2

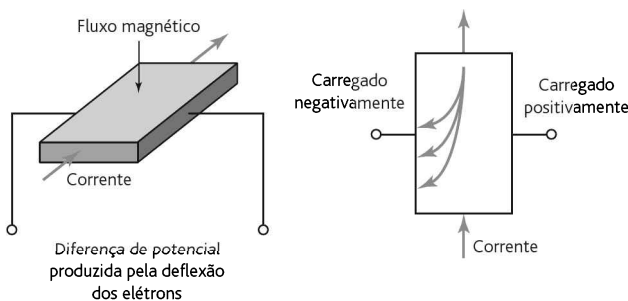


Figura 2.23 Efeito Hall.

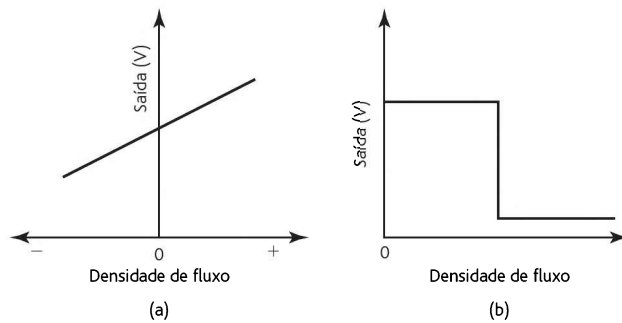


Figura 2.24 Sensores de efeito Hall: (a) linear, (b) de limiar.

fornece uma saída que é bastante linear ao longo da faixa de -40 a $+40$ mT (-400 a $+400$ gauss) para cerca de 10 mV por mT (1 mV por gauss) quando alimentado com uma fonte de 5 V. O sensor de efeito Hall do tipo limiar Allegro UGN3132U fornece uma saída que muda de praticamente zero para cerca de 145 mV, momento no qual a densidade do fluxo magnético é de cerca de 3 mT (30 gauss). Os sensores de efeito Hall têm as vantagens de serem capazes de operar como chaves em uma taxa de repetição de até 100 kHz, custando menos do que chaves eletromecânicas e não apresentando problemas como repique no contato da chave gerando um sinal com uma sequência de comutações em vez de apenas uma. O sensor de efeito Hall é imune a contaminações ambientais e pode ser usado em condições severas de operação.

Estes sensores podem ser usados como sensores de proximidade, posição e deslocamento se o objeto sob análise estiver submetido a um pequeno ímã permanente. Como exemplo, ele pode ser usado para determinar o nível de combustível no tanque de um automóvel. Um ímã é fixado a uma boia e, à medida que o nível de combustível varia, a distância da boia para o sensor Hall varia (Figura 2.25). O resultado é uma tensão de saída Hall que é a medida da distância da boia ao sensor e, portanto, o nível do combustível no tanque.

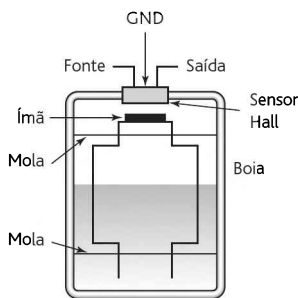


Figura 2.25 Detector de nível de fluido.

Outra aplicação dos sensores de efeito Hall é em motores CC sem escovas. Nestes motores é necessário determinar quando o rotor de ímã permanente está corretamente alinhado com o enrolamento do estator de forma que a corrente através dos enrolamentos possa ser ligada no instante exato para manter a rotação do rotor. Estes sensores são usados para detectar quando o alinhamento está correto.

2.4 Velocidade e movimento

Os exemplos a seguir são de aplicações de sensores para monitorar velocidades linear e angular e detectar movimento. A detecção de movimento inclui sistemas de segurança para detectar intrusos e brinquedos e eletrodomésticos interativos, por exemplo, a tela do caixa eletrônico que ativa quando nos aproximamos dela.

2.4.1 Encoder incremental

O encoder incremental descrito na Seção 2.3.7 pode ser usado para medição de velocidade angular determinando o número de pulsos gerados por segundo.

2.4.2 Tacogerador

O tacogerador é usado para medir velocidade angular. Um dos tipos, o **tacogerador de relutância variável**, consiste de uma roda dentada de material ferromagnético que é fixada ao eixo de rotação (Figura 2.26). Uma bobina sensora é enrolada em um ímã permanente. A variação cíclica resultante no fluxo concatenado produz uma FEM (força eletromotriz) alternada na bobina.

Se a roda contiver n dentes e girar com uma velocidade angular ω , então o fluxo que varia com o tempo na bobina pode ser considerado da forma

$$\Phi = \Phi_0 + \Phi_a \cos n\omega t$$

em que Φ_0 é o valor médio do fluxo e Φ_a é a amplitude da variação do fluxo. A FEM induzida e nas N espiras da bobina sensora é $-N d\Phi/dt$ e, portanto,

$$e = N\Phi_a n\omega \sin \omega t$$

e assim podemos escrever

$$e = E_{\text{máx}} \sin \omega t$$

em que o valor máximo da FEM induzida, $E_{\text{máx}}$, é $N\Phi_a n\omega$, que é a medida da velocidade angular.

Em vez de usar o valor máximo da FEM como uma medida de velocidade angular, um condicionador de sinal em formato de pulso pode ser usado para transformar a saída em uma sequência de pulsos que podem ser contados por um contador no qual o número contado em um determinado intervalo de tempo é a medida da velocidade angular.

Outra forma de um tacogerador é essencialmente um **gerador CA**. Ele consiste de uma bobina, denominada de rotor, que gira conforme a rotação do eixo. Esta bobina gira no campo magnético produzido por um ímã permanente ou eletromagnético estacionário (Figura 2.27), e assim uma FEM alternada é induzida nela. A amplitude ou a frequência desta FEM alternada pode ser usada como uma medida da velocidade angular do rotor. A saída pode ser retificada para fornecer

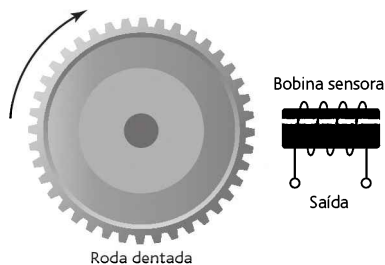


Figura 2.26 Tacogerador de relutância variável.

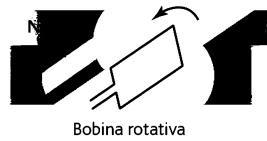


Figura 2.27 Um gerador CA na forma de um tacogerador.

uma tensão CC com uma amplitude que é proporcional à velocidade angular. A não linearidade destes sensores é tipicamente da ordem de $\pm 0,15\%$ da faixa de operação total, e os sensores são usados normalmente para rotacionarem até cerca de 10.000 revoluções/min.

2.4.3 Sensores piroelétricos

Os *materiais piroelétricos*, por exemplo, o tantalato de lítio, são materiais cristalinos que geram carga em resposta a um fluxo de calor. Quando um material destes é aquecido a um ponto um pouco abaixo da temperatura Curie, que para o caso do tantalato de lítio é cerca de 610°C , em um campo elétrico e o material é esfriado enquanto permanece no campo, os dipolos elétricos dentro do material se alinham, tornando-o polarizado (Figura 2.28, (a) antecedendo (b)). Quando o campo é removido, o material retém esta polarização; o efeito é muito semelhante ao da magnetização de um pedaço de ferro quando este é exposto a um campo magnético. Quando um material piroelétrico é exposto a uma radiação infravermelha, sua temperatura aumenta, provocando a redução de polarização no material, os dipolos se tornam mais desordenados e seus alinhamentos são perdidos (Figura 2.28(c)).

Um sensor piroelétrico consiste de um cristal piroelétrico polarizado com eletrodos constituídos de um fino filme metálico nas faces opostas. Devido ao cristal estar polarizado com superfícies carregadas, íons são drenados do ar em torno dele e elétrons são drenados de qualquer circuito de medição conectado ao sensor para equilibrar a carga superficial (Figura 2.29(a)). Se a radiação infravermelha incidir no cristal e mudar sua temperatura, a polarização nele é reduzida, e consequentemente isso é uma redução de carga na superfície do cristal. Há então um excesso de carga nos eletrodos de metal além do necessário para equilibrar a carga nas superfícies do

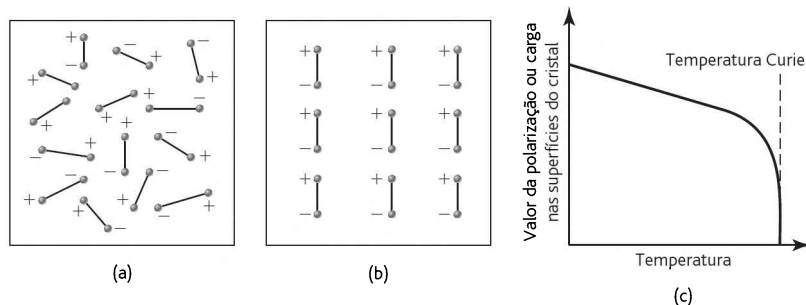


Figura 2.28 (a), (b) Polarização de um material piroelétrico, (c) o efeito da temperatura no valor da polarização.

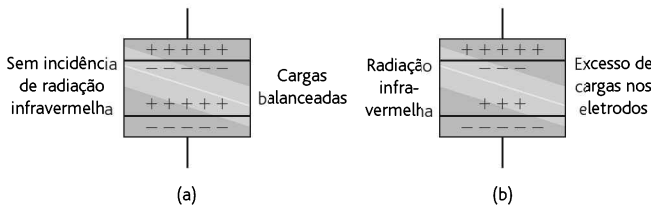


Figura 2.29 Sensor piroelétrico.

crystal (Figura 2.29(b)). Essa carga escoo pelo circuito de medição até que a carga no cristal esteja novamente equilibrada com a carga nos eletrodos. O sensor piroelétrico se comporta então como um gerador de carga que gera carga quando há uma variação em sua temperatura em função de uma radiação infravermelha incidente. Considerando a parte linear do gráfico da Figura 2.28(c), quando há uma variação na temperatura, Δt , a variação na carga, Δq , é proporcional à variação na temperatura:

$$\Delta q = k_p \Delta t$$

em que k_p é uma constante de sensibilidade para o cristal. A Figura 2.30 mostra o circuito equivalente de um sensor piroelétrico, em que ele é efetivamente um capacitor carregado pela carga excessiva com uma resistência R para representar a resistência de fuga interna ou aquela combinada com a resistência de entrada de um circuito externo.

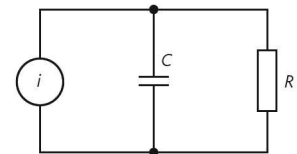


Figura 2.30 Circuito externo.

Para detectar o movimento de uma pessoa ou outra fonte de calor, o elemento sensor tem que distinguir entre a radiação de calor do meio em geral e aquela fornecida pela movimento da fonte de calor. Um único sensor piroelétrico não pode ser capaz disto, por isto é usado um elemento duplo (Figura 2.31). Um elemento sensor deste pode ter um único eletrodo frontal, mas dois eletrodos na face oposta. O resultado é dois sensores que podem ser conectados de forma que, quando ambos receberem o mesmo sinal de calor, suas saídas se cancelem. Quando uma fonte de calor se move de forma que a radiação passa de um elemento sensor para o outro, então a corrente resultante no resistor se alterna, estando primeiro em uma direção e depois invertida. Tipicamente, o movimento de uma pessoa resulta em uma corrente alternada da ordem de 10^{-12} A. Portanto, a resistência R tem que ser muito alta para proporcionar uma tensão significativa. Por exemplo, $50 \text{ G}\Omega$ com a corrente citada resulta em 50 mV . Por esta razão, um transistor é incluído no circuito na configuração de seguidor de tensão para que a impedância de saída caia para alguns quilo-ohms.

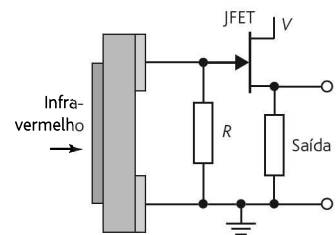


Figura 2.31 Sensor piroelétrico duplo.

Um dispositivo de focalização é necessário para direcionar a radiação infravermelha para o sensor. Espelhos parabólicos podem ser usados; porém, o método mais usado é o de lentes plásticas de Fresnel. Estas lentes também protegem a superfície frontal do sensor e são normalmente usadas em sensores que acionam alarmes de intrusos ou ligam uma luz quando alguém se aproxima dele.

2.5 Força

Uma balança de mola é um exemplo de um sensor de força no qual a força, normalmente um peso, é aplicada ao prato da balança e provoca um deslocamento, ou seja, as molas se estendem. Este deslocamento é uma medida da força. As forças são normalmente medidas através da medição de deslocamentos. Os seguintes métodos ilustram isto.

2.5.1 Célula de carga com *strain gauge*

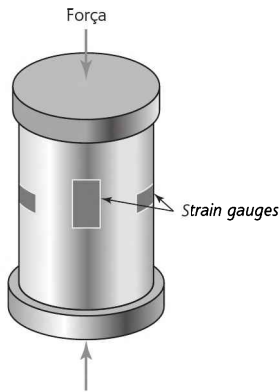


Figura 2.32 Célula de carga com *strain gauge*.

Uma forma muito comum de transdutor de medição de força é baseada na medição de resistência elétrica de *strain gauges* para monitorar esforços produzidos em algum dos elementos quando estendidos, comprimidos ou dobrados com a aplicação de uma força. A configuração destes elementos é geralmente conhecida como **célula de carga**. A Figura 2.32 mostra um exemplo deste tipo de célula. Esta é um tubo cilíndrico no qual os *strain gauges* são fixados. Quando forças são aplicadas ao cilindro, comprimindo-o, então os *strain gauges* apresentam uma variação de resistência que é uma medida do esforço e, por conseguinte, das forças aplicadas. Visto que a temperatura também produz uma variação na resistência, o circuito de condicionamento de sinal usado tem a capacidade de eliminar os efeitos provocados pela temperatura (veja a Seção 3.5.1). Normalmente estas células de carga são usadas para forças de até 10 MN, com erro de não linearidade de cerca de $\pm 0,03\%$ da faixa de operação total, erro de histerese de $\pm 0,02\%$ da faixa de operação total e erro de repetibilidade de $\pm 0,02\%$ da faixa de operação total. As células de carga com *strain gauge* baseada em

esforço de dobra de um elemento de metal com *strain gauges* fixados tendem a ser usadas para forças menores, ou seja, em faixas de valores de 0 a 5 N até 0 a 50 kN. Os erros são tipicamente o de não linearidade de cerca de $\pm 0,03\%$ da faixa de operação total, o de histerese de $\pm 0,02\%$ da faixa de operação total e o de repetibilidade de $\pm 0,02\%$ da faixa de operação total.

2.6 Pressão de fluido

Muitos dos dispositivos usados para monitorar a pressão de um fluido em processos industriais envolvem o monitoramento da deformação elástica de diafragmas, cápsulas, foles e tubos. Os tipos de medidas de pressão que podem ser demandadas são: pressão absoluta, a qual é medida em relação à pressão zero, ou seja, o vácuo; a pressão diferencial, em que uma diferença de pressão é medida; e a pressão manométrica, que é a pressão medida em relação à pressão barométrica.

Para um diafragma (Figura 2.33(a) e (b)), onde há uma diferença de pressão entre os dois lados, então o centro do diafragma se desloca. As estrias em um diafragma resultam em maior sensibilidade. Este movimento pode ser monitorado por algum tipo de sensor de deslocamento, por exemplo, um *strain gauge*, conforme ilustrado na Figu-

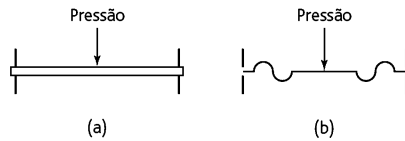


Figura 2.33 Diafragmas: (a) liso, (b) corrugado.

ra 2.34. Um *strain gauge* projetado especialmente para este caso é frequentemente usado. Ele consiste de quatro *strain gauges*, com dois deles medindo o esforço em uma direção circunferencial enquanto os outros dois medem o esforço em uma direção radial. Os quatro *strain gauges* são conectados na forma de uma ponte de Wheatstone (veja o Capítulo 3). Ainda que *strain gauges* possam ser fixados em um diafragma, a alternativa é criar um diafragma de silício com os *strain gauges* como áreas especialmente dopadas no diafragma. Este arranjo é usado com sistemas eletrônicos em carros para monitorar a pressão de admissão.

Com o sensor de pressão MPX da Motorola, o elemento *strain gauge* é integrado, juntamente com um circuito resistivo, em um único chip no diafragma de silício. Quando uma corrente passa no elemento *strain gauge* e uma pressão é aplicada em direção perpendicular, uma tensão é produzida. Este elemento, juntamente com circuitos de condicionamento de sinal e compensação de temperatura, é encapsulado, formando o sensor MPX. A tensão de saída é diretamente proporcional à pressão. Estes sensores estão disponíveis para uso em medições de pressão absoluta (o sistema de numeração MX termina com A, AP, AS ou ASX), pressão diferencial (o sistema de numeração MX termina com D ou DP) e pressão manométrica (o sistema de numeração MX termina com GP, GVP, GS, GVS, GSV ou GVSX). Por exemplo, a série MPX2100 tem uma faixa de pressão de 100 kPa e com uma fonte de alimentação de 16 V CC fornece, para pressões absoluta e diferencial, uma tensão de saída de 40 mV para a faixa de operação total. O tempo de resposta, de 10 a 90%, para uma variação em degrau de 0 a 100 kPa é de cerca de 1,0 ms, e a impedância de saída é da ordem de 1,4 a 3,0 k Ω . Os sensores de pressão absoluta são usados em aplicações como altímetros e barômetros, os de pressão diferencial são usados em medidas de vazão de ar, e os de pressão manométrica são usados em medidas de pressão de motor e pneus.

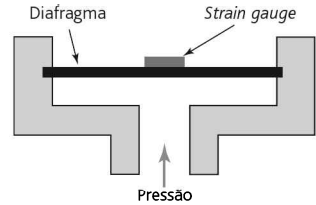


Figura 2.34 Diafragma de pressão manométrica.

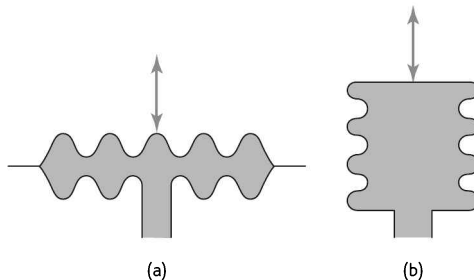


Figura 2.35 (a) Cápsula, (b) fole.

As cápsulas (Figura 2.35(a)) podem ser consideradas como dois diafragmas corrugados combinados e apresentando uma sensibilidade muito maior. A Figura 2.35(b) mostra uma pilha de cápsulas com uma sensibilidade ainda maior. Na Figura 2.36 vemos como um fole pode ser combinado com um LVDT para produzir um sensor de pressão com uma saída elétrica. Os diafragmas, as cápsulas e os foles são feitos de materiais como aço inox, bronze fosforoso e níquel. A borracha e o nylon também estão sendo usados em alguns diafragmas. Pressões na faixa de 10^3 a 10^8 Pa podem ser monitoradas com estes sensores.

Uma forma diferente de deformação é obtida usando-se um tubo com uma seção transversal elíptica (Figura 2.37(a)). Um aumento de pressão em um tubo como este faz ele tender a ter uma seção reta de circunferência maior. Quando um tubo deste tem a forma da letra C (Figura 2.37(b)), geralmente conhecido como **tubo de Bourdon**, ele se abre quando a pressão no tubo aumenta. Uma forma helicoidal deste tubo (Figura 2.37(c)) resulta em uma sensibilidade maior. Os tubos são constituídos de materiais como aço inox e bronze fosforoso e são usados para medidas de pressão na faixa de 10^3 a 10^8 Pa.

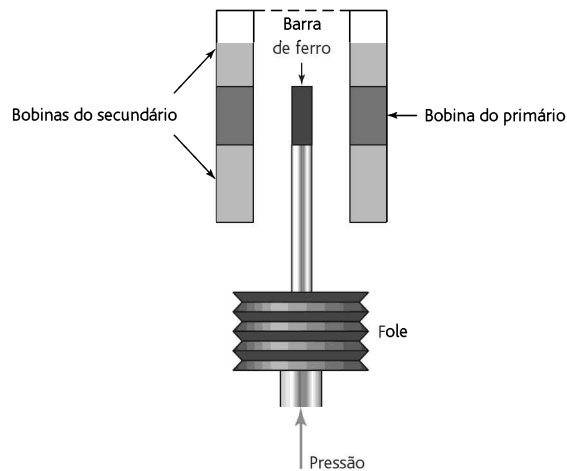


Figura 2.36 LVDT associado a um fole.

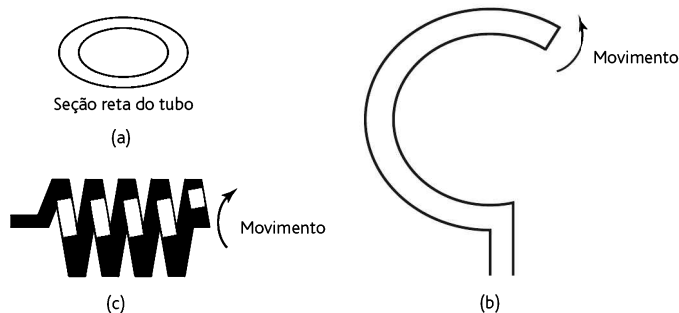


Figura 2.37 Sensores de pressão em forma de tubo.

2.6.1 Sensores piezoelétricos

Os **materiais piezoelétricos**, quando estendidos ou comprimidos, geram cargas elétricas, sendo que uma das faces se torna carregada positivamente e a outra, negativamente (Figura 2.38(a)). Como resultado, uma tensão é produzida. Estes materiais são cristais iônicos que, quando estendidos ou comprimidos, provocam uma variação na distribuição de carga no cristal, de forma que o deslocamento resultante de cargas torna uma face do material carregada positivamente e a outra, negativamente. A carga resultante q na superfície é proporcional ao valor x pelo qual as cargas foram deslocadas, e como o deslocamento é proporcional à força aplicada, F :

$$q = kx = SF$$

em que k é uma constante e S é uma constante denominada **sensibilidade de carga**. Esta constante depende do material e da orientação dos seus cristais. O quartzo tem uma sensibilidade de carga de 2,2 pC/N quando o cristal é cortado em uma determinada direção e as forças são aplicadas em uma direção específica; o titanato de bário tem uma sensibilidade de carga muito maior, da ordem de 130 pC/N, e, no caso do titanato zirconato de chumbo, é por volta de 265 pC/N.

Eletrodos de metal são depositados nas faces opostas do cristal piezoelétrico (Figura 2.38(b)). A capacitância C do material piezoelétrico entre as placas é

$$C = \frac{\epsilon_0 \epsilon_r A}{t}$$

em que ϵ_r é a permissividade relativa do material, A é a área e t é sua espessura. Como a carga $q = Cv$, em que v é a diferença de potencial produzida no capacitor, então

$$v = \frac{St}{\epsilon_0 \epsilon_r A} F$$

A força F é aplicada ao longo da área A e, desta forma, a pressão aplicada p é F/A , e se escrevermos $S_v = (S/\epsilon_0 \epsilon_r)$, o que é denominado de **fator de sensibilidade da tensão**, então

$$v = S_v t p$$

A tensão é proporcional à pressão aplicada. A sensibilidade de tensão para o quartzo é por volta de 0,055 V/m Pa. Para o titanato de bário, é cerca de 0,011 V/m Pa.

Os sensores piezoelétricos são usados para medição de pressão, força e aceleração. Entretanto, as aplicações devem ser tais que a carga produzida em função da pressão não tenha muito tempo para escoar e, portanto, eles tendem a ser usados principalmente em pressões transitórias em vez de situações de estado estacionário.

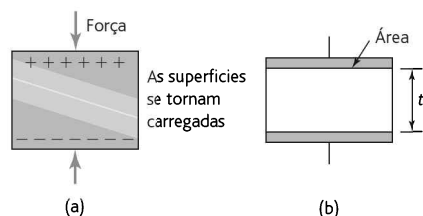


Figura 2.38 (a) Piezoelectricidade, (b) capacitor piezoelétrico.

O circuito elétrico equivalente para um sensor piezoelétrico consiste de um gerador de carga em paralelo com uma capacitância C_s e em paralelo com uma resistência R_s devido a fuga pelo dielétrico (Figura 2.39(a)). Quando o sensor é conectado via cabo, de capacitância C_c , a um amplificador, de capacitância de entrada C_A e resistência R_A , temos efetivamente o circuito mostrado na Figura 2.39(b) e a capacitância total do circuito é $C_s + C_c + C_A$ em paralelo com uma resistência que vale $R_A R_s / (R_A + R_s)$. Quando o sensor é submetido a uma pressão ele se torna carregado, mas, devido à resistência, o capacitor se descarrega com o tempo. Este tempo gasto na descarga depende da constante de tempo do circuito.

2.6.2 Sensor de toque

Um sensor de toque é uma forma particular de sensor de pressão. Este sensor é usado nas “pontas dos dedos” de “mãos” de robôs para determinar quando uma “mão” está em contato com um objeto. Ele também é usado em *touch screens* (telas com sensores de toque) onde o contato físico é detectado. Um tipo de sensor de toque usa um filme de fluoreto de polivinilideno piezoelétrico (PVDF – *piezoelectric polyvinylidene fluoride*). São usadas duas camadas deste filme separadas por um filme macio que transmite vibrações (Figura 2.40). O filme PVDF inferior tem uma tensão alternada aplicada a ele, e isso resulta em oscilações mecânicas do filme (a ação contrária ao efeito piezoelétrico descrito acima). O filme intermediário transmite estas vibrações para o filme PVDF superior. Como consequência do efeito piezoelétrico, estas vibrações provocam uma tensão alternada no filme superior. Quando a pressão é aplicada ao filme PVDF superior, suas vibrações são afetadas e a tensão alternada de saída varia.

2.7 Vazão de líquido

O método tradicional de medir a vazão de líquidos envolve dispositivos baseados em medidas da queda de pressão que ocorre quando o fluido atravessa uma constricção (Figura 2.41). Para um tubo horizontal, em que v_1 é a velocidade do fluido, P_1 é a pressão e A_1 é a área da seção reta do tubo antes da constricção, v_2 é a velocidade, P_2 é a pressão e A_2 é a área da seção reta na constricção, com ρ sendo a densidade do fluido, a equação de Bernoulli fica da seguinte forma:

$$\frac{v_1^2}{2g} + \frac{P_1}{\rho g} = \frac{v_2^2}{2g} + \frac{P_2}{\rho g}$$

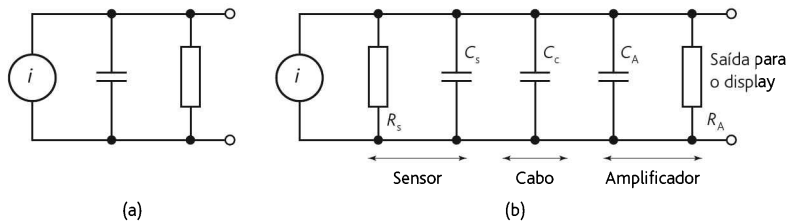


Figura 2.39 (a) Circuito equivalente do sensor, (b) sensor conectado a um amplificador.

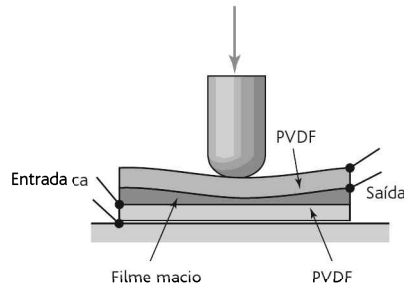


Figura 2.40 Sensor de toque PVDF.

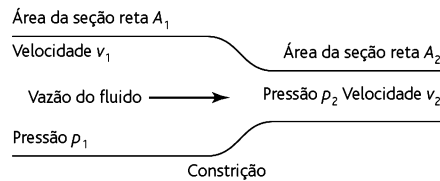


Figura 2.41 Vazão de um fluido através de uma constrição.

Como a massa de líquido que passa por segundo no tubo antes da constrição tem que ser igual àquela após a constrição, temos $A_1 v_1 \rho = A_2 v_2 \rho$. Mas a quantidade Q de líquido que passa através do tubo por segundo é $A_1 v_1 = A_2 v_2$. Portanto,

$$Q = \frac{A}{\sqrt{1 - (A_2/A_1)^2}} \sqrt{\frac{2(P_1 - P_2)}{\rho}}$$

Desta forma, a quantidade de fluido que passa pelo tubo por segundo é proporcional a $\sqrt{\text{diferença da pressão}}$. Assim, medidas de diferença de pressão podem ser usadas para se obter uma medida da vazão. Existem muitos dispositivos baseados neste princípio, e o exemplo que vem a seguir mostra um dos mais usados.

2.7.1 Placa de orifício

A placa de orifício (Figura 2.42) é simplesmente um disco, com um furo central, que é colocado em um tubo por onde o fluido passa. A diferença de pressão é medida entre um ponto igual ao diâmetro do tubo à jusante e um ponto igual à metade do diâmetro à vazante. A placa de orifício é simples, barata, sem partes móveis e é bastante usada. Entretanto, ela não funciona bem com pastas. A precisão dela é tipicamente de $\pm 1,5\%$ da faixa de operação total, ela não é linear e produz uma perda de pressão considerável no sistema no qual está conectada.

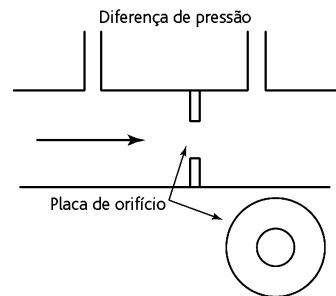


Figura 2.42 Placa de orifício.

2.7.2 Medidor de turbina

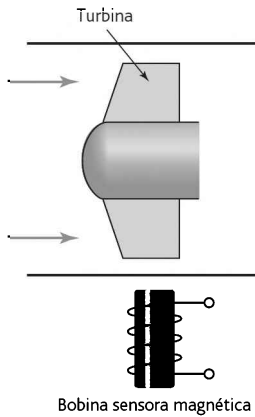


Figura 2.43 Medidor de vazão de turbina.

O medidor de vazão de turbina (Figura 2.43) consiste de um rotor laminado que é sustentado no centro do tubo ao longo do qual passa o fluido. A vazão do fluido coloca o rotor em movimento, sendo a velocidade angular aproximadamente proporcional à vazão. A taxa de revolução do rotor pode ser determinada usando-se uma bobina sensora. Os pulsos são contados e, assim, o número de revoluções do rotor pode ser determinado. Um medidor, com uma precisão de aproximadamente $\pm 0,3\%$, é caro.

2.8 Nível de um líquido

O nível de um líquido em um reservatório pode ser medido diretamente monitorando-se a posição da superfície do líquido ou indiretamente medindo-se alguma variável relacionada à altura. Os métodos diretos podem envolver flutuadores; os métodos indiretos incluem o monitoramento do peso do reservatório por, digamos, células de carga. O peso de um líquido é determinado por $Ah\rho g$, em que A é a área da seção reta do reservatório, h é a altura do líquido, ρ é sua densidade e g é a aceleração da gravidade. Portanto, uma variação na altura do líquido resulta em uma variação no peso. Geralmente, os métodos indiretos envolvem a medição de pressão em algum ponto no interior do líquido, a pressão devido a uma coluna de líquido de altura h sendo dada por $h\rho g$, em que ρ é a densidade do líquido.

2.8.1 Boias

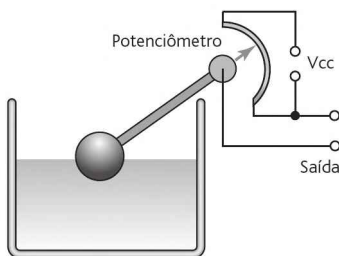


Figura 2.44 Sistema com boia.

Um método direto de monitoramento do nível de um líquido em um reservatório é a monitoração do movimento de uma boia. A Figura 2.44 ilustra este método com um sistema de boia simples. O deslocamento da boia faz com que o braço da alavanca gire, provocando o movimento do cursor de um potenciômetro. O resultado é uma saída na forma de tensão que está relacionada à altura do líquido. Outras formas deste método envolvem uma alavanca que provoca o deslocamento do núcleo de um LVDT, ou ainda a distensão ou compressão de um elemento *strain gauge*.

2.8.2 Pressão diferencial

A Figura 2.45 mostra duas formas de medição de nível baseadas na medição da pressão diferencial. Na Figura 2.45(a), a célula de pressão diferencial determina a diferença de pressão entre o líquido na base do reservatório e a pressão atmosférica, estando o reservatório aberto e submetido à pressão atmosférica. Para um sistema de reservatório fechado ou aberto, o sistema ilustrado em (b) pode ser usado. A célula de pressão diferencial monitora a diferença entre a pressão da base do reservatório e a pressão do ar ou gás acima da superfície do líquido.

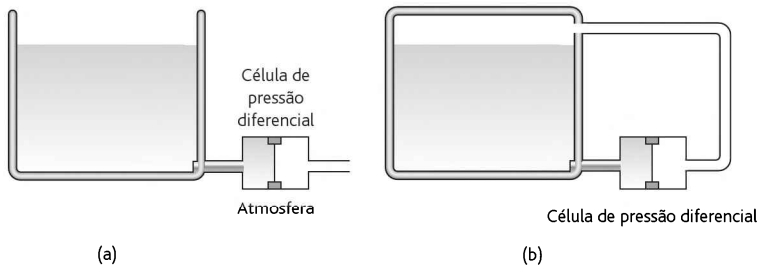


Figura 2.45 Usando um sensor de pressão diferencial.

2.9 Temperatura

As variações na expansão ou contração de sólidos, líquidos ou gases e as variações na resistência elétrica de condutores e semicondutores e fems termoeletricas são normalmente usadas para monitorar a temperatura de um meio. A seguir são apresentados alguns métodos normalmente usados em sistemas de controle de temperatura.

2.9.1 Lâminas bimetálicas

Este dispositivo consiste de duas tiras de metais diferentes unidas. Os metais têm diferentes coeficientes de expansão, e quando a temperatura varia a lâmina composta se dobra encurvando a lâmina. Esta deformação pode ser usada como uma chave de temperatura controlada, assim como em um termostato simples que é normalmente usado em sistemas de aquecimento doméstico (Figura 2.46). O pequeno ímã permite que o sensor apresente a característica de histerese, em que o contato da chave fecha em uma temperatura diferente da qual ele abre.

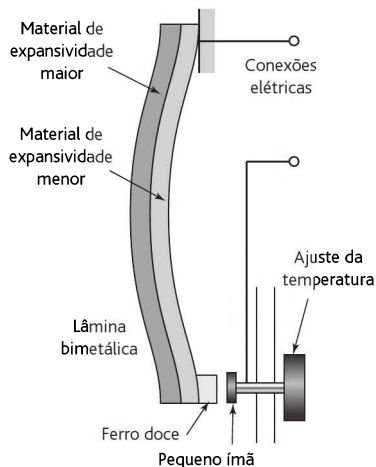


Figura 2.46 Termostato bimetálico.

2.9.2 Detectores de temperatura por variação de resistência

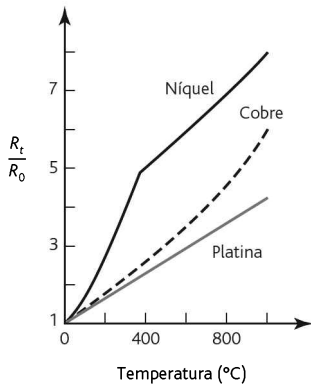


Figura 2.47 Variação da resistência de metais com a temperatura.

A resistência da maioria dos metais aumenta com a temperatura, ao longo de uma faixa limitada de temperatura, de uma forma razoavelmente linear (Figura 2.47). A equação a seguir representa essa relação linear:

$$R_t = R_0(1 + \alpha t)$$

em que R_t é a resistência à temperatura $t(^{\circ}\text{C})$, R_0 é a resistência a 0°C e α é uma constante para o metal denominada de coeficiente de temperatura da resistência. Os detectores de temperatura por variação de resistência (RTDs – *Resistance Temperature Detectors*) são elementos resistivos simples na forma de bobinas de fio de metais como platina, níquel ou ligas de níquel-cobre; a platina é o metal mais usado. Os elementos de filme fino de platina são normalmente construídos por deposição do metal em um substrato adequado, elementos de fio enrolado envolvendo um fio de platina preso por um adesivo de vidro de alta temperatura fixado dentro de um tubo cerâmico. Estes detectores são altamente estáveis e proporcionam

respostas reproduzíveis por longos períodos de tempo. Eles tendem a ter tempos de resposta da ordem de 0,5 a 5 s ou mais.

2.9.3 Termistores

Os termistores são pequenas peças de material feitas de misturas de óxidos metálicos, como os de cromo, cobalto, ferro, manganês e níquel. Estes óxidos são semicondutores. O componente é construído em diversos formatos, como contas (*beads*), discos e barras (Figura 2.48(a)).

A resistência de termistores de óxido metálico convencionais diminuem de forma bastante não linear com o aumento de temperatura, conforme ilustrado na Figura 2.48(b). Estes termistores têm coeficientes de temperatura negativos (NTCs – *Negative Temperature Coefficients*). Entretanto, termistores com coeficiente de temperatura positivos (PTCs – *Positive Temperature Coefficients*) estão disponíveis. A variação na resistência por grau de variação na temperatura é consideravelmente maior em comparação com os metais. A relação resistência-temperatura para um termistor pode ser descrita por uma equação da forma

$$R_t = K e^{\beta/t}$$

em que R_t é a resistência à temperatura t , sendo K e β constantes. Os termistores têm muitas vantagens quando comparados com outros sensores de temperatura. Eles são robustos e podem ser muito pequenos, possibilitando desta forma que temperaturas sejam monitoradas em praticamente qualquer ponto. Devido ao pequeno tamanho eles respondem muito rapidamente a variações na temperatura. Eles produzem grandes variações de resistência por grau de variação na temperatura. A principal desvantagem deles é a não linearidade. Os termistores são usados em sistemas eletrônicos de carros para monitorar variáveis como a temperatura externa e a interna.

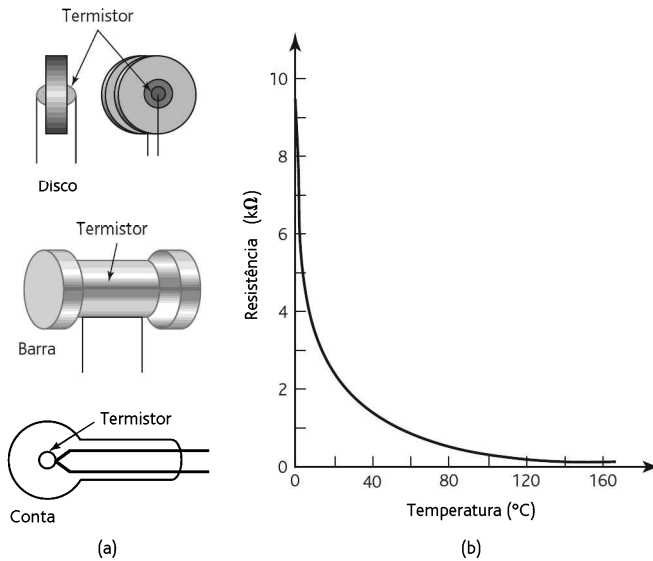


Figura 2.48 Termistores: (a) formas comuns, (b) variação típica da resistência com a temperatura.

2.9.4 Termiododos e transistores

Uma junção de um diodo semiconductor é bastante usada como um sensor de temperatura. Quando a temperatura de semicondutores dopados varia, a mobilidade dos seus portadores de carga varia, e isso afeta a taxa na qual elétrons e lacunas podem se difundir na junção $p-n$. Portanto, quando uma junção $p-n$ tem uma diferença de potencial V nela, a corrente I na junção é uma função da temperatura, sendo dada por

$$I = I_0(e^{eV/kT} - 1)$$

em que T é a temperatura na escala Kelvin, e é a carga de um elétron, k e I_0 são constantes. Aplicando a teoria de logaritmos, podemos escrever a equação anterior em termos de tensão como

$$V = \left(\frac{kT}{e}\right) \ln\left(\frac{I}{I_0} + 1\right)$$

Portanto, para uma corrente constante, V é proporcional à temperatura na escala Kelvin, e desta forma a medida da diferença de potencial em um diodo com corrente constante pode ser usada como uma medida da temperatura. Este sensor é compacto como um termistor, mas tem a grande vantagem de gerar uma resposta que é uma função linear da temperatura. Os diodos para uso como sensores de temperatura, juntamente com o condicionador de sinal necessário, são fornecidos na forma de circuitos integrados, como por exemplo o LM3911, que é um sensor compacto bastante pequeno. A tensão de saída deste componente é proporcional à temperatura na taxa de 10 mV/°C.

Tabela 2.1 Termopares

Ref.	Materiais	Faixa de temperatura (°C)	($\mu\text{V}/^\circ\text{C}$)
B	Platina 30% Ródio/platina 6% ródio	0 a 1800	3
E	Cromel/constantan	-200 a 1000	63
J	Ferro/constantan	-200 a 900	53
K	Cromel/alumel	-200 a 1300	41
N	Nirosil/nisil	-200 a 1300	28
R	Platina/platina 13% ródio	0 a 1400	6
S	Platina/platina 10% ródio	0 a 1400	6
T	Cobre/constantan	-200 a 400	43

denominado tipo J. A Figura 2.50(b) mostra como a FEM varia com a temperatura para diversos pares de metais normalmente usados.

Um circuito com termopar pode ter outros metais no circuito, e eles não terão efeito na FEM termoelétrica desde que todas as junções estejam na mesma temperatura. Isto é conhecido como **lei dos metais intermediários**.

Um termopar pode ser usado com a junção de referência em uma temperatura diferente de 0°C . Entretanto, as tabelas padrões consideram uma junção em 0°C e, desta forma, uma correção nos dados da tabela tem que ser aplicada antes de usá-los. A correção é aplicada usando-se o que é conhecido como a **lei das temperaturas intermediárias**, isto é,

$$E_{t,0} = E_{t,I} + E_{I,0}$$

A FEM $E_{t,0}$ na temperatura t quando a junção fria estiver em 0°C é igual à FEM $E_{t,I}$ na temperatura intermediária I mais a FEM $E_{I,0}$ na temperatura I quando a junção fria estiver em 0°C . Para ilustrar isto, considere um termopar do tipo E que é usado para medição de temperatura com uma junção fria a 20°C . Qual será a FEM termoelétrica em 200°C ? Os dados mostrados a seguir são obtidos de uma tabela padrão:

Temp. ($^\circ\text{C}$)	0	20	200
FEM (mV)	0	1,192	13,419

Usando a lei das temperaturas intermediárias,

$$E_{200,0} = E_{200,20} + E_{20,0} = 13,419 - 1,192 = 12,227 \text{ mV}$$

Note que esta não é uma FEM dada pelas tabelas para uma temperatura de 180°C com uma junção fria a 0°C , isto é, 11,949 mV.

Manter uma junção de um termopar em 0°C , ou seja, mantê-lo imerso em uma mistura de gelo e água, geralmente não é conveniente. Entretanto, um circuito de compensação pode ser usado para fornecer uma FEM que varie com a temperatura da junção fria de tal forma que, quando esta tensão for somada com a FEM do termopar, ele gere uma FEM combinada que é a mesma que seria gerada se a junção fria estivesse a 0°C (Figura 2.51). A FEM compensada pode ser fornecida por um elemento termômetro de resistência.

Os termopares de metais tipo E, J, K e T são relativamente baratos, mas se deterioram com o tempo. Eles têm precisões que são tipicamente de aproximadamente $\pm 1\%$ ou melhores.

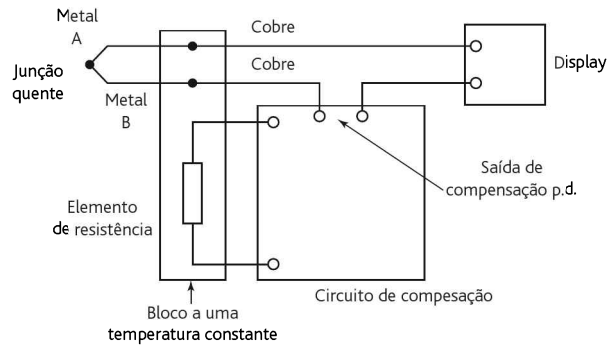


Figura 2.51 Compensação de junção fria.

Os termopares são geralmente montados com um revestimento para conferir a eles proteção mecânica e química. O tipo de revestimento usado depende das temperaturas nas quais o termopar será usado. Em alguns casos, o revestimento é encapsulado com um mineral que é um bom condutor de calor e um bom isolante elétrico. O tempo de resposta de um termopar sem revestimento é muito menor. Esse tempo pode ser aumentado para alguns segundos se um revestimento espesso for usado. Em alguns casos, um grupo de termopares é conectado em série de forma que haja, digamos, 10 ou mais junções quentes medindo a temperatura. As fems produzidas são somadas. Este arranjo é conhecido como **pilha termoeletrica**.

2.10 Sensores de luz

Os **fotodiodos** são diodos semicondutores de junção (veja a Seção 9.3.1 para um estudo sobre diodos) que são conectados em um circuito com polarização reversa, resultando assim em uma resistência muito alta (Figura 2.52(a)). Sem luz incidente, a corrente reversa é praticamente desprezível e é denominada corrente de escuro. Quando a luz incide na junção, são gerados pares extras de elétron-lacuna, provocando um aumento na corrente reversa, e a resistência do diodo cai (Figura 2.52(b)). Esta corrente reversa é quase proporcional à intensidade luminosa. Por exemplo, a corrente na ausência de luz com uma polarização reversa de 3 V pode ser 25 μA , e quando iluminada por 25.000 lúmens/m^2 a corrente aumenta para 375 μA . A resistência do dispositivo na ausência de luz é de $3/(25 \times 10^{-6}) = 120 \text{ k}\Omega$ e com a presença de luz é de $3/(375 \times 10^{-6}) = 8 \text{ k}\Omega$. Um fotodiodo pode portanto ser usado como um dispositivo de resistência variável controlado pela incidência de luz nele. Os fotodiodos têm uma resposta muito rápida à luz.

Os **fototransistores** (veja a Seção 9.3.3 para um estudo sobre transistores) têm uma junção $p-n$, base-coletor, sensível à luz. Quando não há luz incidente, a corrente coletor-emissor é muito pequena. Quando há luz incidente, uma corrente de base é produzida e é diretamente proporcional à luz incidente. Com isto, uma corrente de coletor é produzida, a qual é proporcional à intensidade luminosa. Os fototransistores são geralmente encapsulamentos integrados contendo um fototransistor e um transistor convencional em uma configuração Darlington (Figura 2.53). Como esta

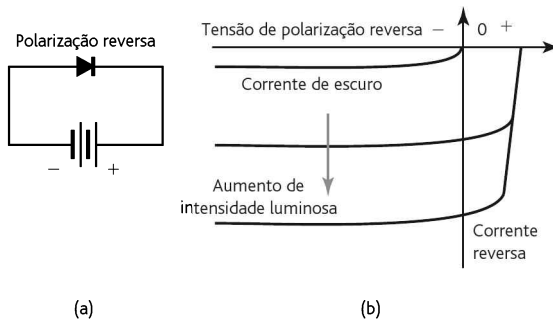


Figura 2.52 Fotodiodo.

configuração proporciona um alto ganho de corrente, o dispositivo produz uma corrente de coletor muito maior para uma dada intensidade luminosa.

Um fotorresistor tem uma resistência que depende da intensidade luminosa que incide sobre ele, diminuindo linearmente conforme a intensidade luminosa aumenta. O fotorresistor de sulfeto de cádmio responde melhor à luz com comprimentos de onda menores que cerca de 515 nm, e o fotorresistor de seleneto de cádmio responde melhor para comprimentos de onda menores do que cerca de 700 nm.

Um arranjo de sensores de luz é frequentemente necessário em pequenos espaços para determinar as variações na intensidade luminosa nestes espaços. Um exemplo disso ocorre em uma câmera digital para capturar a imagem fotografada e convertê-la para o formato digital. Com esta finalidade, um **dispositivo de acoplamento de carga (CCD – Charge-coupled Device)** é normalmente usado. Um CCD é um arranjo de células sensíveis à luz muito pequenas denominadas pixels. Estas células são basicamente uma camada p de silício separada de uma camada de silício tipo n por uma camada de depleção. Quando exposta à luz, uma célula se torna eletricamente carregada e esta carga é então convertida por um circuito eletrônico em um número digital de 8 bits. Ao tirar uma foto, o circuito eletrônico da câmera digital descarrega as células sensíveis à luz, ativa um obturador eletromecânico para expor as células à imagem, lê em seguida o valor de carga de 8 bits de cada célula, capturando assim a imagem. Visto que as células pn são “daltônicas” e precisamos de fotos coloridas, a luz passa por uma matriz de filtro colorida antes de atingir as células. Isto permite que apenas a luz verde incida sobre algumas células, a azul sobre outras, e as vermelhas sobre as restantes. Então, levando por fim em consideração a saída a partir das células vizinhas, uma imagem colorida pode ser criada.

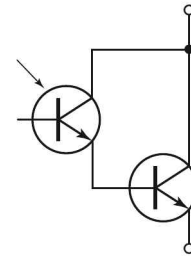


Figura 2.53 FotoDarlington.

2.11 Seleção de sensores

Na seleção de um sensor para uma determinada aplicação, existem alguns fatores que precisam ser considerados:

- 1 A natureza da medida desejada, ou seja, a variável a ser medida, seu valor nominal, a faixa de valores, a precisão desejada, a velocidade desejada da medida, a confiabilidade desejada, as condições ambientes sob as quais a medida será feita.
- 2 A natureza da saída desejada a partir do sensor, que determina a necessidade de condicionamento do sinal para produzir um sinal de saída adequado a partir da medida.
- 3 Então, os possíveis sensores podem ser identificados, levando-se em consideração estes fatores bem como a faixa de operação, a precisão, a linearidade, a velocidade de resposta, a confiabilidade, manutenibilidade, o tempo de vida, os requisitos de alimentação, a robustez, a disponibilidade e o custo.

A seleção de sensores não pode ser feita isoladamente sem se considerar a forma da saída que o sistema necessita após o condicionador de sinal, e, portanto, deve existir uma combinação satisfatória entre sensor e condicionador de sinal.

Para ilustrar isto, considere a seleção de um sensor para medir o nível de um ácido corrosivo em um reservatório. O nível pode variar de 0 a 2 m em um reservatório circular com diâmetro de 1 m. O reservatório vazio tem um peso de 100 kg. A variação mínima no nível a ser detectada é 10 cm. O ácido tem uma densidade de 1050 kg/m^3 . A saída do sensor é elétrica.

Devido à natureza corrosiva do ácido, parece apropriado um método indireto na determinação do nível. Portanto, é possível usar uma célula de carga, ou células de carga, para monitorar o peso do reservatório. Estas células produzem uma saída elétrica. O peso do líquido varia de 0, quando o reservatório está vazio, até $1050 \times 2 \times \pi(1^2/4) \times 9,8 = 16,2 \text{ kN}$, quando cheio. Somando este valor ao peso do reservatório vazio, obtemos um peso que varia de aproximadamente 1 a 17 kN. A resolução necessário corresponde a uma variação de nível de 10 cm, ou seja, uma variação no peso de $0,10 \times 1050 \times \pi(1^2/4) \times 9,8 = 0,8 \text{ kN}$. Se forem usadas três células de carga para suportar o tanque, então cada uma necessitará de uma faixa de operação de aproximadamente 0 a 6 kN com uma resolução de 0,27 kN. Os catálogos de fabricantes podem ser consultados para ver se uma célula de carga como esta é disponibilizada.

2.12 Entrada de dados por chaves

As **chaves mecânicas** consistem de um ou mais pares de contatos que podem ser fechados ou abertos mecanicamente, ligando ou desligando circuitos elétricos. Assim, os sinais 0 ou 1 podem ser transmitidos pela ação de abrir ou fechar uma chave. O termo **chave fim de curso** é usado quando as chaves são abertas ou fechadas através do deslocamento de um objeto e usadas para indicar o limite deste deslocamento antes que uma ação seja iniciada.

As chaves mecânicas são especificadas em termos do número de polos e posições. Os **polos** correspondem ao número de circuitos separados que podem ser completados pela mesma ação de comutação, e as **posições** correspondem ao número de contatos individuais de cada polo. A Figura 2.54(a) mostra uma chave de um polo e uma posição (SPST – *single pole-single throw*), a Figura 2.54(b) mostra uma chave de um polo e duas posições (SPDT – *single pole-double throw*) e a Figura 2.54(c) mostra uma chave de dois polos e duas posições (DPDT – *double pole-double throw*).

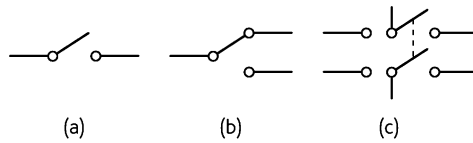


Figura 2.54 Chaves: (a) SPST, (b) SPDT, (c) DPDT.

2.12.1 Eliminação do efeito de repique

Um problema que ocorre com chaves mecânicas é o **repique da chave**. Quando uma chave mecânica é acionada para fechar os contatos, temos o movimento de um contato em direção ao outro. Um bate contra o outro e, devido os elementos de contato serem elásticos, ocorre o repique. Eles podem repicar certo número de vezes (Figura 2.55(a)) antes de finalmente estabilizar em seu estado fechado após, tipicamente, cerca de 20 ms. Cada um dos contatos durante o repique pode ser identificado como um em separado. Portanto, para um microprocessador, pode parecer que ocorreu dois ou mais acionamentos da chave. De forma similar, quando uma chave mecânica é aberta, o repique pode ocorrer. Para evitar este problema, podemos usar recursos de hardware ou de software.

Com o uso de software, o microprocessador é programada para detectar se a chave é fechada e então esperar, digamos, 20 ms. Após verificar o término do repique e que a chave está na mesma posição fechada, a próxima parte do programa pode ser executada. A solução com o uso de hardware para o problema do repique é baseada no uso de um flip-flop. A Figura 2.55(b) mostra um circuito para eliminar o efeito do repique em uma chave SPDT que é baseado no uso de um flip-flop SR (veja a Seção 5.4.1). Conforme mostrado, temos S em nível 0 e R em nível 1 com uma saída em nível 0. Quando a chave se move para a posição inferior, inicialmente S se torna nível 1 e R, nível 0. Isto faz com que a saída seja nível 1. O repique faz com que S varie de 1 para 0 para 1 para 0, etc., não resultando em mudança na saída. Um flip-flop como este pode ser implementado a partir de duas portas NOR ou NAND. Uma chave de um polo e duas posições pode ser o efeito do repique eliminado com o uso de um flip-flop D (veja a Seção 5.4.4). A Figura 2.55(c) mostra o circuito. A

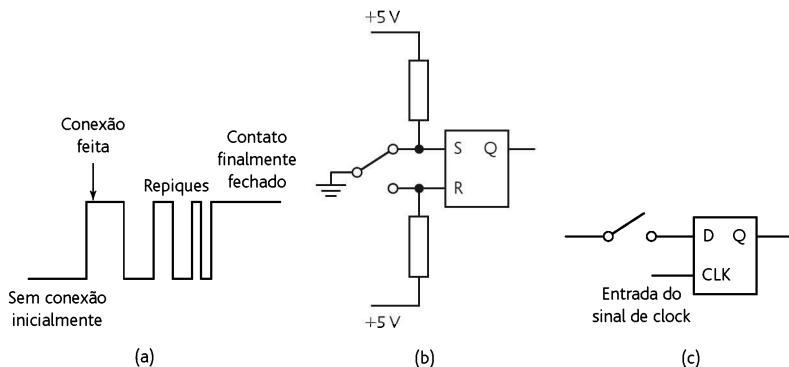


Figura 2.55 (a) Repique da chave ao fechá-la, (b) eliminado o efeito do repique com o uso de um flip-flop SR, (c) eliminado o efeito do repique com o uso de um flip-flop D.

saída deste flip-flop varia apenas quando o sinal de clock for ativado. Portanto, escolhendo um período de clock que seja maior do que o tempo no qual o repique dura, digamos, 20 ms, o sinal de repique será ignorado.

Um método alternativo de eliminação do efeito do repique por hardware faz uso de um *Schmitt trigger*. Este dispositivo tem a característica de histerese mostrada na Figura 2.56(a). Quando a tensão de entrada está além do limiar de comutação superior e resultando em uma saída de nível baixo, então a tensão de entrada precisa cair abaixo do limiar de comutação inferior antes que a saída comute para nível alto. Reciprocamente, quando a tensão de entrada estiver abaixo do limiar de comutação inferior produzindo uma saída em nível alto, então a entrada precisa aumentar acima do limiar de comutação superior antes que a saída possa comutar para nível baixo. Este dispositivo pode ser usado para acentuar sinais de variações lentas: quando o sinal passa o nível de comutação, ele se torna uma borda acentuadamente definida entre dois níveis lógicos bem definidos. O circuito mostrado na Figura 2.56(b) pode ser usado para eliminação do efeito do repique; observe o símbolo do circuito para um *Schmitt trigger*. Com a chave aberta, o capacitor carrega e a tensão aplicada ao *Schmitt trigger* se torna alta, resultando em uma saída baixa. Quando a chave é fechada, o capacitor se descarrega muito rapidamente e, assim, o primeiro repique descarrega o capacitor; o *Schmitt trigger* comuta então a saída para nível alto. Repiques sucessivos da chave não permitem que o capacitor se recarregue até o valor de limiar necessário e, portanto, os repiques posteriores não comutam o *Schmitt trigger*.

2.12.2 Teclado

Um teclado é um arranjo de chaves, podendo ser um teclado comum de computador ou na forma de membrana em um dispositivo como um forno de microondas. Um tipo de tecla de contato geralmente usada em um teclado é mostrado na Figura 2.57(a). Ao se pressionar a tecla, é forçado um contato contra o outro, com a mola retornando a tecla para a posição desligada quando esta é liberada. Uma chave de membrana típica (Figura 2.57(b)) é construída a partir de dois filmes de plástico finos nos quais camadas condutivas são impressas. Estas camadas estão separadas uma da outra. Quando a superfície da membrana é pressionada, o contato superior fecha com o inferior, fazendo a conexão, e se abre quando a membrana é liberada.

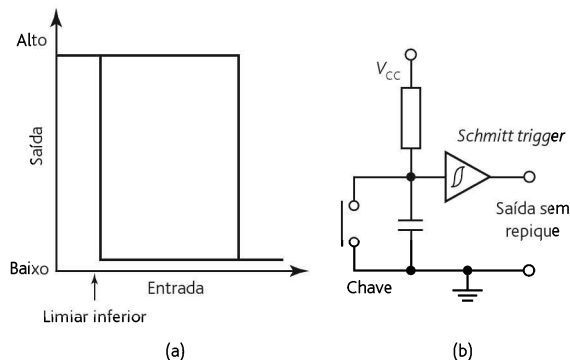


Figura 2.56 *Schmitt trigger*: (a) características, (b) usado para eliminar o efeito do repique da chave.

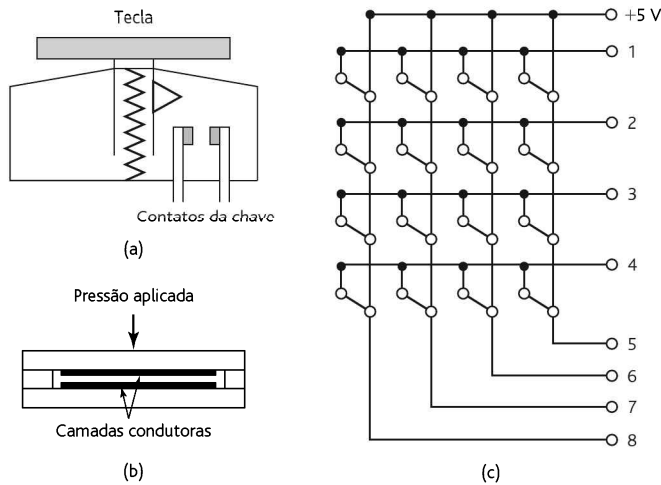


Figura 2.57 (a) Chave de contato, (b) tecla de membrana, (c) teclado com 16 teclas.

Cada chave poderia ser conectada individualmente para gerar sinal quando fechada, porém, um método mais econômico é conectá-las em um arranjo em que uma saída individual para cada tecla não é necessária, mas cada tecla fornece uma única combinação linha–coluna. A Figura 2.57(c) mostra a conexão de um teclado de 16 teclas.

Resumo

Um **sensor** é um elemento que produz um sinal relacionado à grandeza medida. Um **transdutor** é um elemento que, quando submetido a alguma alteração física, apresenta uma variação relacionada. Portanto, os sensores são transdutores. Entretanto, um sistema de medida pode usar transdutores, além do sensor, em outras partes do sistema para converter sinais de um formato em outro.

A **faixa de operação** de um transdutor define os limites entre os quais a entrada pode variar. O **span** é o valor máximo da entrada menos o valor mínimo. O **erro** é a diferença entre o resultado de uma medição e o seu valor real. A **precisão** é o quanto o valor de uma medida pode estar errado. A **sensibilidade** indica o quanto há de saída por unidade de entrada. O **erro de histerese** é a diferença entre os valores obtidos quando alcançados por meio de um aumento contínuo e uma variação continuamente decrescente. O **erro de não linearidade** é o erro dado considerando-se uma relação linear. A **repetibilidade/reprodutibilidade** é uma medida da capacidade do transdutor de fornecer saídas iguais para aplicações repetidas de entradas iguais. A **estabilidade** é a capacidade de fornecer a mesma saída para uma entrada constante. A **zona morta** é a faixa de valores de entrada para os quais não há saída. A **resolução** é a menor variação na entrada que produz uma variação observável na saída. O **tempo de resposta** é o tempo decorrido após uma entrada em degrau até a saída alcançar uma percentagem especificada, por exemplo, 95% da entrada. A **constante de tempo** é 63,2% do tempo de resposta.

O **tempo de subida** é o tempo que a saída gasta para subir de alguma percentagem especificada até o estado estacionário. O **tempo de acomodação** é o tempo gasto para a saída se ajustar dentro de uma percentagem, digamos, 2% do valor em estado estacionário.

Problemas

- 2.1 Explique o significado da seguinte informação dada na especificação de transdutores:
- Um acelerômetro piezoelétrico.
Não linearidade: $\pm 0,5\%$ da faixa de operação total.
 - Um transdutor de deslocamento linear capacitivo.
Não linearidade e histerese: $\pm 0,01\%$ da faixa de operação total.
 - Um transdutor de medição de força do tipo *strain gauge* de resistência.
Sensibilidade de temperatura: $\pm 1\%$ da faixa de operação total para temperaturas ambiente normais.
 - Um transdutor de pressão de fluido por resistência.
Precisão: $\pm 1\%$ da leitura mostrada.
 - Termopar.
Sensibilidade: termopar de níquel cromo/níquel alumínio: $0,039 \text{ mV}/^\circ\text{C}$ quando a junção fria estiver a 0°C .
 - Giroscópio para medição de velocidade angular.
Repetibilidade: $\pm 0,01\%$ da faixa de operação total.
 - Transdutor de deslocamento indutivo.
Linearidade: $\pm 1\%$ da carga especificada.
 - Célula de carga.
Erro total devido a não linearidade, histerese e não repetibilidade: $\pm 0,1\%$.
- 2.2 Um termopar de cobre-constantan é usado para medir temperaturas entre 0 e 200°C . A FEM a 0°C é 0 mV , a 100°C é $4,277 \text{ mV}$ e a 200°C é $9,286 \text{ mV}$. Qual será o erro de não linearidade a 100°C como uma percentagem da faixa de operação total de saída se for considerada uma relação linear entre a FEM e a temperatura ao longo de toda a faixa?
- 2.3 Um elemento termopar quando retirado de um líquido a 50°C e inserido em um líquido a 100°C no instante $t = 0$ forneceu os valores de FEM mostrados a seguir. Determine o tempo de resposta de 95%.
- | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|
| Tempo (s) | 0 | 20 | 40 | 60 | 80 | 100 | 120 |
| FEM (mV) | 2,5 | 3,8 | 4,5 | 4,8 | 4,9 | 5,0 | 5,0 |
- 2.4 Qual é o erro de não linearidade, como uma percentagem da faixa de operação total, produzido quando um potenciômetro de $1 \text{ k}\Omega$ tem uma carga de $10 \text{ k}\Omega$ e está a um terço do seu deslocamento máximo?
- 2.5 Qual é a alteração na resistência de um *strain gauge* de resistência elétrica com um fator *gauge* de 2,1 e resistência de 50Ω se ele for submetido a um esforço de 0,001?

- 2.6 Você pode escolher entre um encoder incremental e um absoluto para a medição de um deslocamento angular. Qual é a diferença principal entre os resultados que podem ser obtidos com estes métodos?
- 2.7 Um codificador é usado com uma roda dentada de raio de 50 mm para monitorar um deslocamento linear. Se o encoder produz 256 pulsos por revolução, qual será o número de pulsos produzidos por um deslocamento linear de 200 mm?
- 2.8 Um transformador diferencial rotacional variável tem especificações que incluem a seguinte informação:
- Faixa de operação: $\pm 30^\circ$, erro de linearidade de $\pm 0,5\%$ da faixa de operação total
 $\pm 60^\circ$, erro de linearidade de $\pm 2,0\%$ da faixa de operação total
- Sensibilidade: $1,1 \text{ (mV/V entrada) } = /\text{grau}$
 Impedância: 750Ω no primário, 2.000Ω no secundário
- Qual será (a) o erro na leitura de 40° devido a não linearidade quando o RDVT é usado na faixa de $\pm 60^\circ$, e (b) a variação da tensão de saída que ocorre por grau se houver uma tensão de entrada de 3 V?
- 2.9 Quais são as vantagens e desvantagens do potenciômetro de filme plástico quando comparado com o de fio enrolado?
- 2.10 Um sensor de pressão que consiste de um diafragma com *strain gauges* fixado na sua superfície tem a seguinte especificação:
- Faixa de operação: 0 a 1400 kPa, 0 a 35.000 kPa
 Erro de não linearidade: $\pm 0,15\%$ da faixa de operação total
 Erro de histerese: $\pm 0,05\%$ da faixa de operação total
- Qual é o erro total devido a não linearidade e histerese para uma leitura de 1.000 kPa na faixa de 0 a 1.400 kPa?
- 2.11 O nível de água em um reservatório aberto é monitorado por uma célula de pressão diferencial que responde a uma diferença de pressão entre a da base do reservatório e a da atmosfera. Determine a faixa de operação das pressões diferenciais a que a célula terá que responder se o nível da água pode variar entre a altura zero acima do ponto de medição da célula e 2 m acima deste ponto.
- 2.12 Um termopar de ferro-constantan é usado para medir temperaturas entre 0 e 400°C . Qual será o erro de não linearidade como uma porcentagem da leitura de fundo de escala em 100°C se for considerada uma relação linear entre a FEM e a temperatura?
- FEM a $100^\circ\text{C} = 5,268 \text{ mV}$; FEM a $400^\circ\text{C} = 21,846 \text{ mV}$
- 2.13 Um detector de temperatura com resistência de platina tem uma resistência de $100,00 \Omega$ a 0°C , $138,50 \Omega$ a 100°C e $175,83 \Omega$ a 200°C . Qual será o erro de não linearidade em $^\circ\text{C}$ a 100°C se for considerado que o detector tem uma relação linear entre 0 e 200°C ?

- 2.14 Um sensor de pressão com *strain gauge* tem a especificação mostrada a seguir. Ele é capaz de medir uma pressão da ordem de 100 kPa para uma precisão de ± 5 kPa em um ambiente em que a temperatura é praticamente constante em torno de 20°C?

Faixa de operação: 2 a 70 MPa, 70 kPa a 1 MPa

Alimentação: 10 V CC ou CA (rms)

Faixa de operação total de saída: 40 mV

Erros de não linearidade e de histerese: $\pm 0,5\%$

Faixa de temperatura de operação: -54 a $+120^\circ\text{C}$

Deslocamento térmico do zero: 0,030% da faixa de operação total da saída/ $^\circ\text{C}$

Sensibilidade do deslocamento térmico: 0,030% da faixa de operação total da saída/ $^\circ\text{C}$

- 2.15 Um sensor com boia para a determinação do nível de água em um reservatório tem uma boia cilíndrica de massa 2,0 kg, uma área de seção reta de 20 cm² e um comprimento de 1,5 m. Ele flutua verticalmente na água e é pressionado para cima contra uma haste fixada em sua extremidade superior. Qual é a força de empuxo máxima e mínima exercida pela boia na haste? Sugira um mecanismo através do qual a deformação da haste sob a ação da força de empuxo possa ser monitorada.
- 2.16 Sugira um sensor que poderia ser usado como parte de um sistema de controle de um forno para monitorar a vazão do óleo de aquecimento no tubo. A saída do sistema de medição é um sinal elétrico que pode ser usado para ajustar a velocidade da bomba de óleo. O sistema tem que ser capaz de operar continuamente e automaticamente, sem ajustes, por longos períodos de tempo.
- 2.17 Sugira um sensor que possa ser usado como parte de um sistema de controle, para determinar a diferença nos níveis dos líquidos em dois reservatórios. A saída deve fornecer um sinal elétrico para o sistema de controle.
- 2.18 Sugira um sensor que possa ser usado como parte de um sistema para controle da espessura de uma folha enrolada monitorando sua espessura à medida que ela é desenrolada. A folha de metal está em movimento contínuo e a medida precisa ser feita rapidamente para possibilitar que uma ação de correção seja tomada rapidamente. O sistema de medição tem que fornecer um sinal elétrico.

3

Condicionamento de Sinais

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Explicar os requisitos para o condicionamento de sinais.
- Explicar como os amplificadores operacionais podem ser usados.
- Explicar os requisitos de proteção e filtragem.
- Explicar os princípios da ponte de Wheatstone e, em particular, como ela é usada com *strain gauges*.
- Explicar o princípio da modulação por pulso.

3.1 Condicionamento de sinais

O sinal de saída do sensor de um sistema de medição geralmente tem que ser processado de alguma forma para torná-lo adequado ao próximo estágio da operação. O sinal pode ser, por exemplo, muito pequeno, necessitando ser amplificado, pode conter interferência que tem que ser removida, pode ser não linear e precisar ser linearizado, pode ser analógico e precisar ser digitalizado, pode ser digital e ter que ser transformado em analógico, pode ser uma variação de resistência e ter que ser transformado em uma variação de corrente, pode ser uma variação de tensão e ter que ser transformado em uma variação de corrente de amplitude adequada, etc. Todas estas transformações podem ser denominadas **condicionamento de sinais**. Por exemplo, a saída de um termopar é uma tensão pequena, de poucos milivolts. Um módulo de condicionamento de sinal pode então ser usado para converter este sinal em um sinal de corrente de amplitude adequada, com rejeição a ruído, linearização e compensação de junção fria (isto é, no caso da junção fria não estar em 0°C).

O Capítulo 4 continua com a discussão do condicionamento de sinais envolvendo sinais digitais.

3.1.1 Processos de condicionamento de sinais

Os itens listados a seguir são alguns dos processos que podem ocorrer no condicionamento de um sinal:

- 1 *Proteção* para evitar danos ao próximo elemento, por exemplo, um microprocessador, o que ocorreria com uma alta tensão ou corrente. Portanto, podem existir resistores de limitação de corrente em série, fusível para interromper o circuito caso a corrente seja muito alta, proteção de polaridade e circuitos de limitação de tensão (veja a seção 3.3).
- 2 Obtenção do sinal na *grandeza correta*. Isto quer dizer converter o sinal em uma tensão ou corrente contínua. Portanto, por exemplo, a resistência que varia em um *strain gauge* tem que ser convertida em uma variação de tensão. Isto pode ser feito através do uso de uma ponte de Wheatstone e usando-se a tensão de saída desbalanceada (veja a Seção 3.5). Isto pode significar uma transformação do sinal digital ou analógico (veja a Seção 4.3 para conversores analógico-digital e digital-analógico).
- 3 Obtenção do *nível* correto do sinal. O sinal de um termopar pode ser de apenas alguns milivolts. Se este sinal for inserido em um conversor analógico-digital para ser tratado por um microprocessador, então será necessário torná-lo muito maior. Os amplificadores operacionais são bastante usados em ampliações (veja a Seção 3.2).
- 4 Eliminação ou redução de *ruído*. Por exemplo, os filtros podem ser usados para eliminar os principais ruídos de um sinal (veja a Seção 3.4).
- 5 *Manipulação* do sinal, como, por exemplo, torná-lo uma função linear de alguma variável. Os sinais de alguns sensores, por exemplo, um medidor de vazão, é não linear, e portanto pode ser usado um condicionador de sinal de forma que o sinal enviado para o próximo elemento seja linear (veja a Seção 3.2.6).

As próximas seções apresentam alguns dos elementos que podem ser usados no condicionamento de sinais.

3.2 O amplificador operacional

Um amplificador pode ser considerado essencialmente um sistema que tem uma entrada e uma saída (Figura 3.1), e seu **ganho de tensão** é a razão entre as tensões de saída e entrada quando elas são medidas em relação ao terra (GND). A **impedância de entrada** de um amplificador é definida como a tensão de entrada dividida pela corrente de entrada, e a **impedância de saída** é a tensão de saída dividida pela corrente de saída.

A base de muitos módulos condicionadores de sinais é o **amplificador operacional**. Este dispositivo é um amplificador CC de alto ganho, tipicamente da ordem de 100.000 ou mais, e é fornecido na forma de um chip de silício em um circuito integrado (CI). Ele tem duas entradas, conhecidas como entrada inversora (−) e entrada não inversora (+). A saída depende das conexões feitas nestas entradas. O amplificador operacional tem outras entradas, isto é, uma tensão de alimentação negativa, uma tensão de alimentação positiva e duas entradas denominadas compensação de *offset*, sendo estas capazes de corrigir um comportamento não ideal do amplificador (veja a Seção 3.2.8). A Figura 3.2 mostra as conexões dos pinos do amplificador operacional 741.

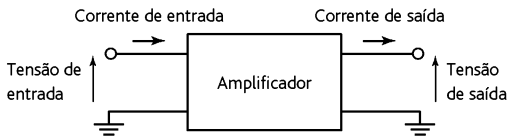


Figura 3.1 Amplificador.

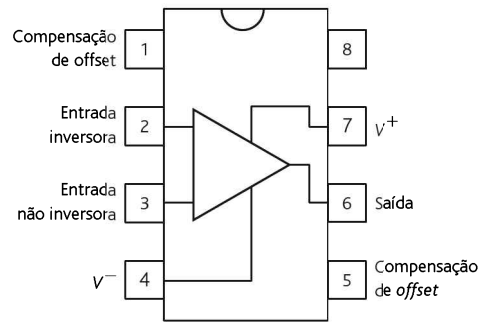


Figura 3.2 Conexão dos pinos para um amplificador operacional 741.

Um modelo ideal para um amplificador operacional apresenta um ganho infinito, impedância de entrada infinita e impedância de saída zero, ou seja, a tensão de saída é independente da carga.

Os tópicos a seguir indicam os tipos de circuitos que podem ser usados com os amplificadores operacionais quando usados como condicionadores de sinal.

3.2.1 Amplificador inversor

A Figura 3.3 mostra as conexões feitas no amplificador quando usado como um **amplificador inversor**. A entrada do circuito é conectada à entrada inversora através do resistor R_1 , e a entrada não inversora é conectada no terra (GND). Um caminho de realimentação é estabelecido, via resistor R_2 , da saída para a entrada inversora. O amplificador operacional tem um ganho de tensão em torno de 100.000, e a variação da tensão de saída é tipicamente limitada em torno de ± 10 V. A tensão de entrada tem que estar entre $+0,0001$ e $-0,0001$. Isto é virtualmente zero, portanto o ponto X está virtualmente no potencial GND. Por esta razão ele é denominado de **terra virtual**. A diferença de potencial em R_1 é $(V_{in} - V_X)$. Portanto, para um amplificador operacional ideal com um ganho infinito e $V_X = 0$, o potencial de entrada, V_{in} , pode ser considerado como sendo sobre R_1 . Portanto,

$$V_{in} = I_1 R_1$$

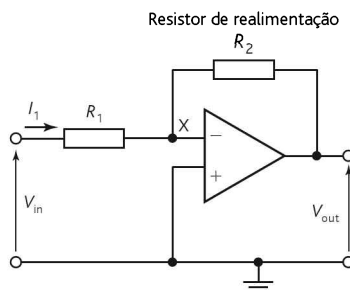


Figura 3.3 Amplificador inversor.

O amplificador operacional tem uma impedância muito alta entre seus terminais de entrada; no caso do 741, esta impedância está em torno de $2\text{ M}\Omega$. Portanto, virtualmente nenhuma corrente flui do ponto X para a entrada. Em um amplificador ideal, a impedância de entrada é considerada infinita e, desta forma, não há corrente de X para a entrada. Portanto, a corrente I_1 em R_1 tem que ser igual à corrente em R_2 . A diferença de potencial em R_2 é $(V_X - V_{out})$, e assim, como V_X é zero em um amplificador ideal, a diferença de potencial em R_2 é $-V_{out}$. Portanto,

$$-V_{out} = I_1 R_2$$

Dividindo-se estas duas equações,

$$\text{ganho de tensão do circuito} = \frac{V_{out}}{V_{in}} = -\frac{R_2}{R_1}$$

Assim, o ganho de tensão do circuito é determinado unicamente pela relação entre os valores de R_2 e R_1 . O sinal negativo indica que a saída é invertida, ou seja, defasada 180° em relação à entrada.

Para ilustrar isto, considere o circuito de um amplificador operacional inversor que tem uma resistência de $1\text{ M}\Omega$ na entrada inversora e uma resistência de realimentação de $10\text{ M}\Omega$. Qual é o ganho de tensão deste circuito?

$$\text{ganho de tensão do circuito} = \frac{V_{out}}{V_{in}} = -\frac{R_2}{R_1} = -\frac{10}{1} = -10$$

3.2.2 Amplificador não inversor

A Figura 3.4(a) mostra o amplificador operacional conectado como um amplificador não inversor. Pode-se considerar a saída a partir do circuito divisor de tensão constituído por R_1 em série com R_2 . A tensão V_X corresponde à fração $R_1/(R_1 + R_2)$ da tensão de saída, ou seja,

$$V_X = \frac{R_1}{R_1 + R_2} V_{out}$$

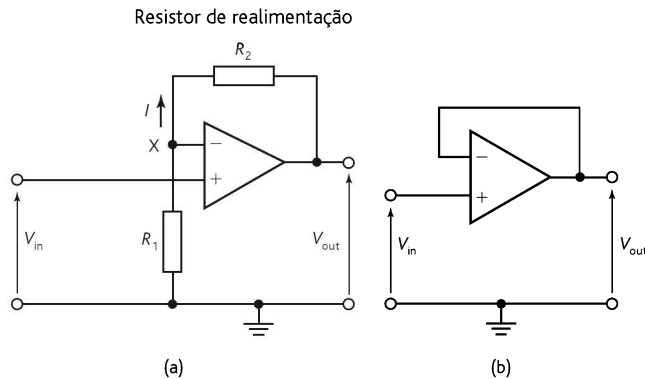


Figura 3.4 (a) Amplificador não inversor, (b) seguidor de tensão.

Como virtualmente não há corrente entre as entradas do amplificador operacional, não há diferença de potencial entre elas. Portanto, em um amplificador operacional ideal, temos $V_X = V_{in}$. Desta forma,

$$\text{ganho de tensão do circuito} = \frac{V_{out}}{V_{in}} = \frac{R_1 + R_2}{R_1} = 1 + \frac{R_2}{R_1}$$

Uma forma particular deste amplificador é quando a malha de realimentação é um curto-circuito, ou seja, $R_2 = 0$. Neste caso o ganho de tensão é 1. A entrada do circuito tem uma resistência alta, tipicamente $2 \text{ M}\Omega$. A resistência de saída, ou seja, a resistência entre o terminal de saída e GND é muito menor, como, por exemplo, 75Ω . Portanto, a resistência no circuito seguinte é relativamente pequena e é menos provável que sobrecarregue o circuito anterior. Este amplificador é conhecido como **seguidor de tensão**; a Figura 3.4(b) mostra o circuito básico.

3.2.3 Amplificador somador

A Figura 3.5 mostra o circuito de um amplificador somador. Assim como no caso do amplificador inversor (Seção 3.2.1), o ponto X é um terra virtual. Portanto, a soma das correntes que entram no ponto X tem que ser igual à soma das que saem dele. Desta forma,

$$I = I_A + I_B + I_C$$

Porém, $I_A = V_A/R_A$, $I_B = V_B/R_B$ e $I_C = V_C/R_C$. A mesma corrente I passa através do resistor de realimentação. A diferença de potencial em R_2 é $(V_X - V_{out})$. Portanto, como V_X pode ser considerada zero, essa diferença de potencial é igual a $-V_{out}$, e assim $I = -V_{out}/R_2$. Portanto,

$$-\frac{V_{out}}{R_2} = \frac{V_A}{R_A} + \frac{V_B}{R_B} + \frac{V_C}{R_C}$$

A saída é portanto a soma ponderada das entradas, ou seja,

$$V_{out} = -\left(\frac{R_2}{R_A}V_A + \frac{R_2}{R_B}V_B + \frac{R_2}{R_C}V_C\right)$$

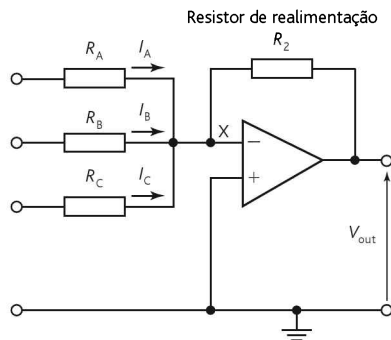


Figura 3.5 Amplificador somador.

Se $R_A = R_B = R_C = R_1$, então

$$V_{out} = -\frac{R_2}{R_1}(V_A + V_B + V_C)$$

Para ilustrar isto, considere o projeto de um circuito que pode ser usado para gerar uma tensão de saída que é a média das tensões de entrada de três sensores. Considerando que uma saída invertida é aceitável, um circuito como o da Figura 3.5 pode ser usado. Cada uma das três entradas tem que ser ponderada para 1/3 para resultar em uma saída média. Portanto, é necessário ter um ganho de tensão de 1/3 para cada uma das entradas. Desta forma, se a resistência de realimentação for de 4 kΩ, cada ramo de entrada será 12 kΩ.

3.2.4 Amplificador integrador

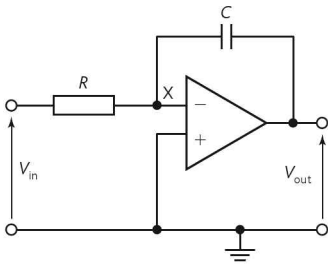


Figura 3.6 Amplificador integrador.

Considere um circuito amplificador operacional inversor com realimentação através de um capacitor, conforme ilustra a Figura 3.6.

A corrente é a taxa de movimentação da carga q . Como para um capacitor a carga $q = Cv$, em que v é a tensão no capacitor, então a corrente nele é $i = dq/dt = C dv/dt$. A diferença de potencial em C é $(v_X - v_{out})$, e como v_X é efetivamente zero, pois é um terra virtual, ela é $-v_{out}$. Portanto, a corrente no capacitor é $-C dv_{out}/dt$. Mas esta também é a corrente na resistência de entrada R . Portanto,

$$\frac{v_{in}}{R} = -C \frac{dv_{out}}{dt}$$

Rearranjando os termos, temos

$$dv_{out} = -\left(\frac{1}{RC}\right)v_{in} dt$$

Integrando os dois membros, temos

$$v_{out}(t_2) - v_{out}(t_1) = -\frac{1}{RC} \int_{t_1}^{t_2} v_{in} dt$$

$v_{out}(t_2)$ é a tensão de saída no instante t_2 e $v_{out}(t_1)$ é a tensão de saída no instante t_1 . A saída é proporcional à integral da tensão de entrada, ou seja, a área sob o gráfico da tensão de entrada com o tempo.

Um circuito de diferenciação pode ser obtido a partir do circuito do amplificador integrador trocando entre si o capacitor e o resistor.

3.2.5 Amplificador de diferença

Um amplificador de diferença amplifica a diferença entre duas tensões de entrada. A Figura 3.7 mostra este circuito. Como virtualmente não há corrente entre a alta resistência que existe entre as entradas do amplificador operacional, não há queda de potencial entre elas, ou seja, elas estão com potenciais iguais. A tensão V_2 está sobre os resistores R_1 e R_2 em série. Portanto, o potencial V_X é

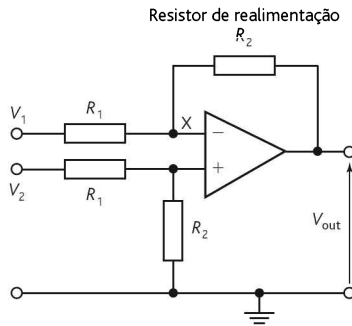


Figura 3.7 Amplificador de diferença.

$$\frac{V_X}{V_2} = \frac{R_2}{R_1 + R_2}$$

A corrente na resistência de realimentação tem que ser igual àquela proporcionada por V_1 em R_1 . Portanto,

$$\frac{V_1 - V_X}{R_1} = \frac{V_X - V_{out}}{R_2}$$

Isto pode ser rearranjado para obtermos

$$\frac{V_{out}}{R_2} = V_X \left(\frac{1}{R_2} + \frac{1}{R_1} \right) - \frac{V_1}{R_1}$$

Portanto, substituindo V_X na equação anterior, temos

$$V_{out} = \frac{R_2}{R_1} (V_2 - V_1)$$

Portanto, a saída é uma medida da diferença entre as duas tensões de entrada.

A Figura 3.8 mostra um exemplo da aplicação deste circuito com um sensor (um termopar). A diferença de tensão entre as fems das duas junções do termopar é amplificada. Os valores de R_1 e R_2 podem, por exemplo, ser escolhidos de forma que a saída do circuito seja 10 mV para uma diferença de temperatura entre as junções

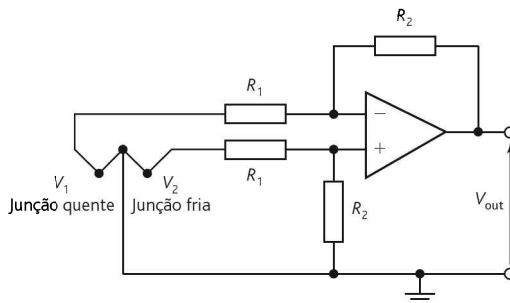


Figura 3.8 Amplificador de diferença com um termopar.

do termopar de 10°C se esta diferença produzir uma diferença de FEM entre as junções de $530\ \mu\text{V}$. Para este circuito temos

$$V_{\text{out}} = \frac{R_2}{R_1}(V_2 - V_1)$$

$$10 \times 10^{-3} = \frac{R_2}{R_1} \times 530 \times 10^{-6}$$

Portanto, $R_2/R_1 = 18,9$. Portanto, se a resistência de R_1 for $10\ \text{k}\Omega$, R_2 tem que ser $189\ \text{k}\Omega$.

Um amplificador de diferença pode ser usado com uma ponte de Wheatstone (veja a Seção 3.5), que pode ter sensores *strain gauges* nos ramos, para amplificar a diferença de potencial desbalanceada que ocorre quando a resistência em um ou mais ramos varia. Quando a ponte está em equilíbrio, os dois terminais de saída da ponte estão com potenciais iguais; desta forma, não há diferença de potencial na saída. Estes terminais podem estar em, digamos, $5\ \text{V}$. Portanto, o amplificador de diferenças tem $5\ \text{V}$ nas entradas. Quando a ponte entra em desequilíbrio, um terminal de saída pode estar com $5,01\ \text{V}$ e o outro, com $4,99\ \text{V}$, e assim as entradas do amplificador de diferenças são $5,01$ e $4,99\ \text{V}$. O amplificador amplifica esta diferença de tensão de $0,02\ \text{V}$. O sinal inicial de $5\ \text{V}$, que era comum às duas entradas, é denominado **tensão em modo comum**, V_{CM} (*common mode voltage*). O amplificador amplifica apenas a diferença entre os dois sinais pressupõe que os dois canais de entrada sejam perfeitamente casados e o amplificador operacional tenha o mesmo valor alto de ganho de tensão relativo a ambos. Um dispositivo real não apresenta esta perfeição, e portanto, a saída não é exatamente proporcional à diferença entre as duas tensões de entrada. Portanto, podemos escrever a equação de saída da seguinte forma

$$V_{\text{out}} = G_d \Delta V + G_{\text{CM}} V_{\text{CM}}$$

em que G_d é o ganho para a diferença de tensão, ΔV , e G_{CM} é o ganho para a tensão de modo comum, V_{CM} . Quanto menor o valor de G_{CM} , menor o efeito da tensão de modo comum na saída. O quanto o amplificador operacional se desvia da situação ideal para entradas em modo comum é denominado de razão de rejeição em modo comum (CMRR – *Common Mode Rejection Ratio*):

$$\text{CMRR} = \frac{G_d}{G_{\text{CM}}}$$

Para minimizar o efeito da tensão em modo comum na saída, é necessário uma alta CMRR. Este parâmetro é geralmente especificado em decibéis (dB). Portanto, em uma escala de decibel, um CMRR de, digamos, 10.000 seria $20 \log 10.000 = 80\ \text{dB}$. Um amplificador operacional típico pode ter um CMRR entre 80 e $100\ \text{dB}$.

Um tipo comum de **amplificador de instrumentação** envolve três amplificadores operacionais (Figura 3.9), em vez de um único amplificador de diferenças, e está disponível na forma de um único circuito integrado. Este circuito é projetado para ter uma impedância de entrada alta, tipicamente por volta de $300\ \text{M}\Omega$, um alto ganho de tensão e um excelente CMRR, tipicamente maior que $100\ \text{dB}$. O primeiro estágio envolve os amplificadores A_1 e A_2 , sendo um conectado como um amplificador inversor e o outro, como não inversor. O amplificador A_3 é um amplificador de diferença com entradas A_1 e A_2 .

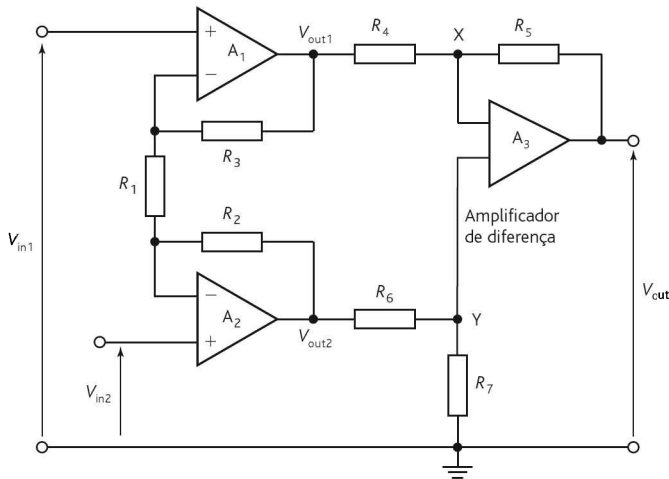


Figura 3.9 Amplificador de instrumentação.

Como praticamente nenhuma corrente entra em A_3 , a corrente de R_4 é igual à de R_5 . Portanto,

$$\frac{V_{out1} - V_X}{R_4} = \frac{V_X - V_{out}}{R_5}$$

A entrada diferencial de A_3 é praticamente zero, de modo que $V_Y = V_X$. Portanto, a equação anterior pode ser escrita como

$$V_{out} = \left(1 + \frac{R_5}{R_4}\right)V_Y - \frac{R_5}{R_4}V_{out1}$$

R_6 e R_7 formam um divisor de potencial para a tensão V_{out2} , de forma que

$$V_Y = \frac{R_6}{R_6 + R_7}V_{out2}$$

Portanto, podemos escrever

$$V_{out} = \frac{1 + \frac{R_5}{R_4}}{1 + \frac{R_7}{R_6}}V_{out2} - \frac{R_5}{R_4}V_{out1}$$

Então, através de uma escolha adequada de valores de resistências, obtemos fatores de multiplicação iguais para as duas entradas para o amplificador de diferença. Isto requer que

$$1 + \frac{R_5}{R_4} = \left(1 + \frac{R_7}{R_6}\right) \frac{R_5}{R_4}$$

e, portanto, $R_4/R_5 = R_6/R_7$.

Podemos aplicar o **princípio da superposição**, ou seja, podemos considerar a saída produzida por cada fonte em separado e, em seguida, somá-las para obter a resposta total. O amplificador A_1 tem uma entrada do sinal diferença V_{in1} em sua entrada não inversora, e a amplifica com um ganho de $1 + R_3/R_1$. Ele também tem uma entrada de V_{in2} em sua entrada inversora, e esta é amplificada para dar um ganho de $-R_3/R_1$. A tensão em modo comum, V_{cm} , na entrada inversora também é amplificada por A_1 . Portanto, a saída de A_1 é

$$V_{out1} = \left(1 + \frac{R_3}{R_1}\right)V_{in1} - \left(\frac{R_3}{R_1}\right)V_{in2} + \left(1 + \frac{R_3}{R_1}\right)V_{cm}$$

De forma similar, para o amplificador A_2 temos

$$V_{out2} = \left(1 + \frac{R_2}{R_1}\right)V_{in2} - \left(\frac{R_2}{R_1}\right)V_{in1} + \left(1 + \frac{R_2}{R_1}\right)V_{cm}$$

A entrada diferença para A_3 é $V_{out1} - V_{out2}$, e assim

$$\begin{aligned} V_{out2} - V_{out1} &= \left(1 + \frac{R_3}{R_1} + \frac{R_2}{R_1}\right)V_{in1} - \left(1 + \frac{R_2}{R_3} + \frac{R_3}{R_1}\right)V_{in2} \\ &\quad + \left(\frac{R_3}{R_1} - \frac{R_2}{R_1}\right)V_{cm} \end{aligned}$$

Com $R_2 = R_3$, o termo relativo à tensão de modo comum desaparece, assim

$$V_{out2} - V_{out1} = \left(1 + \frac{2R_2}{R_1}\right)(V_{in1} - V_{in2})$$

Então, o ganho total é $(1 + 2R_2/R_1)$ e é geralmente ajustado através de R_1 .

A Figura 3.10 mostra as conexões dos pinos e alguns detalhes das especificações para um amplificador de instrumentação de propósito geral e baixo custo (Burr-Brrown INA114) que usa a configuração de três amp-ops do projeto. O ganho é ajustado através de uma conexão de um resistor externo R_G entre os pinos 1 e 8, sendo dado por $1 + 50/R_G$ com R_G em $k\Omega$. O termo $50 k\Omega$ surge da soma dos dois resistores de realimentação internos.

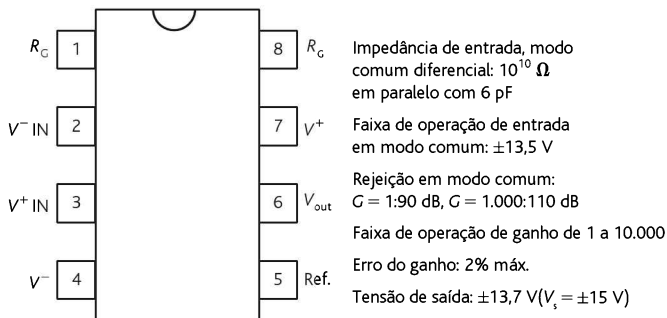


Figura 3.10 INA114.

3.2.6 Amplificador logarítmico

Alguns sensores têm saídas que são não lineares. Por exemplo, a saída de um termopar não é uma função perfeitamente linear da diferença de temperatura entre as junções. Um condicionador de sinal pode ser usado para linearizar a saída de um sensor como este. Isto pode ser feito usando-se um circuito com amplificador operacional que é projetado para se ter uma relação não linear entre entrada e a saída, de modo que quando sua entrada for não linear, a saída será linear. Isto é obtido através de uma escolha adequada do componente conectado na malha de realimentação.

O amplificador logarítmico mostrado na Figura 3.11 é um exemplo de um condicionador de sinal como o que estamos discutindo. A malha de realimentação contém um diodo (ou um transistor com a base aterrada). O diodo tem uma característica não linear. Ela pode ser representada por $V = C \ln I$, em que C é uma constante. Então, como a corrente na malha de realimentação é igual à corrente na resistência de entrada, e a diferença de potencial no diodo é $-V_{out}$, podemos escrever

$$V_{out} = -C \ln(V_{in}/R) = K \ln V_{in}$$

em que K é uma constante. Entretanto, se a entrada V_{in} for gerada por um sensor com uma entrada t , em que $V_{in} = A e^{at}$, com A e a constantes, então

$$V_{out} = K \ln V_{in} = K \ln(A e^{at}) = K \ln A + Kat$$

O resultado é uma relação linear entre V_{out} e t .

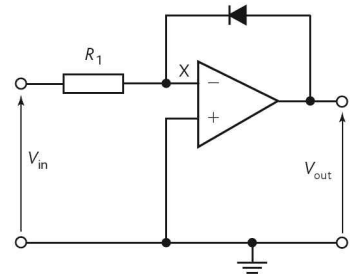


Figura 3.11 Amplificador logarítmico.

3.2.7 Comparador

Um comparador indica qual entre duas tensões é a maior. Um amplificador operacional usado sem realimentação ou outros componentes pode ser usado como um comparador. Uma das tensões é aplicada na entrada inversora e a outra, na não inversora (Figura 3.12(a)). A Figura 3.12(b) mostra a relação entre a tensão de saída e a diferença entre as duas tensões de entrada. Quando as duas entradas são iguais, não há saída. Entretanto, quando a entrada não inversora for maior que a entrada inversora

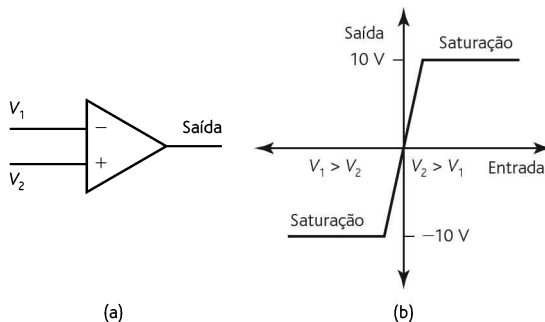


Figura 3.12 Comparador.

em mais que uma pequena fração de um volt, então a saída salta para o estado de tensão de saturação positiva de tipicamente $+10\text{ V}$. Quando a entrada inversora for maior que a não inversora, a saída salta para um estado de tensão de saturação negativa de tipicamente -10 V . Este circuito pode ser usado para determinar quando uma tensão excede certo nível, sendo a saída então usada para talvez iniciar alguma ação.

Como ilustração de uma utilização como esta, considere o circuito mostrado na Figura 3.13. Este circuito foi projetado de forma que quando uma temperatura crítica for atingida, um relé é ativado e inicia alguma resposta. O circuito tem uma ponte de Wheatstone com um termistor em um dos ramos. Os resistores na ponte têm suas resistências selecionadas de modo que, na temperatura crítica, a ponte esteja em equilíbrio. Quando a temperatura estiver abaixo deste valor, a resistência do termistor R_1 será maior do que R_2 e a ponte estará desequilibrada. Como consequência, haverá uma diferença de tensão entre as entradas do amplificador operacional, e a saída dele estará no nível de saturação baixo. Isto mantém o transistor desligado, ou seja, as duas junções, base-emissor e base-coletor, estarão polarizadas reversamente, e assim nenhuma corrente passará através da bobina do relé. Quando a temperatura subir e a resistência do termistor cair, a ponte entrará em equilíbrio e o amplificador operacional comutará então para o seu nível de saturação superior. Consequentemente, o transistor estará ligado, ou seja, suas junções estarão polarizadas diretamente, e o relé será energizado.

Em uma outra ilustração do uso de um comparador, considere o sistema usado para garantir que em um aparelho de CD o feixe laser esteja focado na superfície do disco. No caso de um aparelho de CD, as lentes são usadas para focalizar o feixe laser no CD, que tem a informação de áudio armazenada como uma sequência microscópica de depressões e partes planas. A luz é refletida de volta a partir do disco para um arranjo de quatro fotodiodos (Figura 3.14). A saída destes fotodiodos é então usada para reproduzir o som. A razão para se ter quatro fotodiodos é que o arranjo pode ser usado também para determinar se o feixe de luz do laser está focalizado. Quando o feixe está focalizado no disco, então a luz que retorna do disco incide no arranjo de fotodiodos, formando uma área circular com quantidades iguais de luz em cada fotodiodo. Como resultado, a saída do amplificador operacional, conectado como comparador, é zero. Quando o feixe está fora de foco, é gerada uma área elíptica de luz. Isto resulta em diferentes quantidade de luz incidindo em cada fotocélula. As saídas de dois conjuntos de células diagonais são comparadas e, por

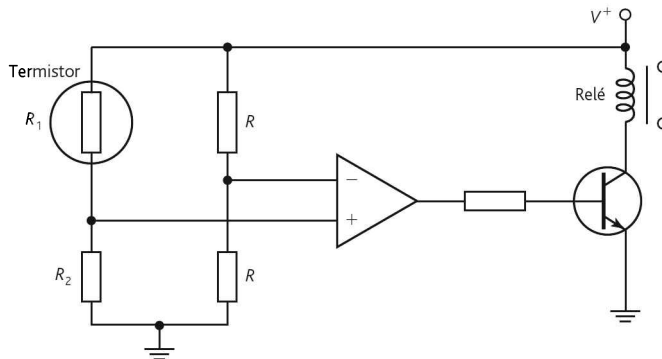


Figura 3.13 Circuito de controle de temperatura.

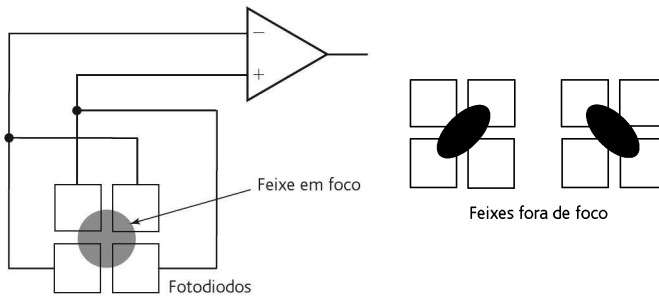


Figura 3.14 Sistema de focalização para um aparelho de CD.

serem diferentes, o comparador gera uma saída que indica que o feixe está fora de foco e em que direção isto está ocorrendo. A saída pode então ser usada para iniciar uma ação de correção ajustando as lentes para focalizar o feixe no disco.

3.2.8 Erros de amplificadores

Os amplificadores operacionais reais não são dispositivos perfeitos (ideais) conforme discutido nas seções anteriores deste capítulo. Um problema particularmente significativo é a **tensão de offset***.

Um amplificador operacional é um amplificador de alto ganho que amplifica a diferença entre as duas entradas. Portanto, se as duas entradas são colocadas em curto-circuito entre si, poderíamos esperar que a saída fosse zero. Entretanto, na prática isso não ocorre, e uma tensão de saída relativamente grande pode ser detectada. Este efeito é produzido por desequilíbrios no circuito interno do amplificador operacional. A tensão de saída pode ser colocada em zero aplicando-se uma tensão adequada entre os terminais de entrada. Isto é conhecido como tensão de *offset*. Muitos amplificadores operacionais possuem uma configuração interna que permite a aplicação desta tensão de *offset* via potenciômetro. No caso do 741 isto é feito conectando-se um potenciômetro de 10 k Ω entre os pinos 1 e 5 (veja a Figura 3.2) e conectando-se o contato do cursor na tensão negativa da fonte de alimentação (Figura 3.15). Os desequilíbrios internos do amplificador operacional são corrigidos ajustando-se a posição do cursor até que, sem tensão de entrada no amplificador, não tenha tensão de saída.

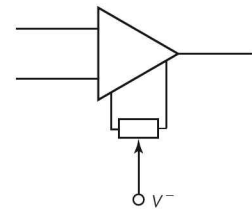


Figura 3.15 Correção da tensão de *offset*.

3.3 Proteção

Existem várias situações em que a conexão de um sensor à próxima unidade, por exemplo, um microprocessador, pode danificar esta unidade em função de uma tensão ou corrente alta. Uma proteção contra corrente alta pode ser feita incorporando-se

* N. de T.: Optamos por manter o termo *offset* em inglês devido ao seu uso consagrado (deslocamento).

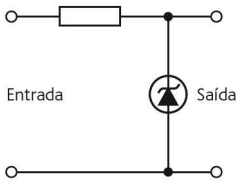


Figura 3.16 Circuito de proteção com diodo Zener.

na linha de entrada um resistor em série para limitar a corrente a um nível aceitável e um fusível para interromper a corrente caso ela exceda o nível de segurança. Tensões altas e polaridades invertidas podem ser protegidas através do uso de um circuito com diodo Zener (Figura 3.16). Os diodos Zener funcionam como diodos comuns até a tensão de ruptura, quando eles se tornam condutores. Portanto, para permitir uma tensão máxima de 5 V, mas impedir tensões acima de 5,1 V, podemos escolher um diodo Zener de 5,1 V. Quando a tensão subir até 5,1 V, o diodo Zener entra em ruptura e sua resistência cai para um valor muito baixo. O resultado é que a tensão no diodo e, portanto, a do circuito à frente, cai. Devido ao diodo Zener ter uma baixa resistência

para a corrente em um sentido e uma alta resistência no sentido oposto, ele também fornece proteção contra polaridades invertidas. Ele é conectado com a polaridade correta para proporcionar uma resistência alta na saída e, assim, uma queda de tensão alta. Quando a polaridade da fonte é invertida, o diodo tem uma baixa resistência, e uma pequena queda de tensão aparece na saída.

Em algumas situações é desejável isolar circuitos completamente e remover todas as conexões elétricas entre eles. Isto pode ser feito usando-se um **optoisolador**. Assim, podemos ter a saída de um microprocessador aplicada a um diodo emissor de luz (LED – *Light-Emitting Diode*) que emite radiação infravermelha. Esta radiação é detectada por um fototransistor ou triac no qual a corrente replica as variações que ocorrem com a tensão aplicada no LED. A Figura 3.17 mostra alguns tipos de optoisoladores. O termo **razão de transferência** é usado para especificar a relação entre a corrente de saída e a de entrada. Tipicamente, um optoisolador com um transistor simples (Figura 3.17(a)) produz um corrente de saída que é menor do que a de entrada, podendo a razão de transferência ser de uns 30%, com um valor máximo de 7 mA. Entretanto, a configuração Darlington (Figura 3.17(b)) pode produzir uma corrente de saída maior do que a de entrada, por exemplo, o 6N139 da Siemens tem uma razão de transferência de 800% com um valor de saída máximo de 60 mA. Um outro tipo de optoisolador (Figura 3.17(c)) usa um triac e, assim, pode ser aplicado em circuitos de corrente alternada. Um optoisolador com triac típico é capaz de operar com a tensão média. Um outro tipo ainda (Figura 3.17(d)) usa um triac

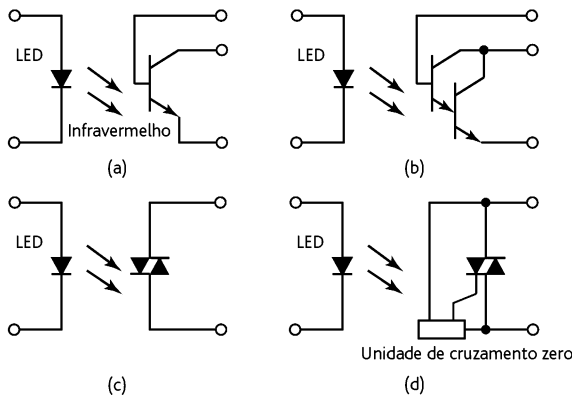


Figura 3.17 Optoisoladores: (a) transistor, (b) Darlington, (c) triac, (d) triac com unidade de cruzamento zero.

juntamente com uma unidade de cruzamento zero, como, por exemplo, o MOC3011 da Motorola, para reduzir transientes e interferências eletromagnéticas.

As saídas de um optoisolador podem ser usadas diretamente para acionar circuitos de carga de baixa potência. Portanto, um optoisolador Darlington pode ser usado como interface entre um microprocessador e lâmpadas ou relés. Para comutar um circuito de potência alta, um optoacoplador pode ser usado para acionar um relé e este aciona um dispositivo de potência maior.

A Figura 3.18 mostra um circuito de proteção que pode ser usado na entrada de um microprocessador; para evitar que o LED tenha a polaridade invertida ou uma tensão aplicada alta, ele pode ser protegido por um circuito com o diodo Zener mostrado na Figura 3.16, e se houver sinal alternado na entrada, um diodo pode ser inserido na linha de entrada para retificá-lo.

3.4 Filtragem

O termo **filtragem** é usado para descrever o processo de remoção de certa banda de frequências de um sinal e permitir que outras sejam transmitidas. A faixa de frequências que passam pelo filtro é conhecida como **banda de passagem**, a que não passa é a **banda de corte**, e na fronteira entre estas duas está a **frequência de corte**. Os filtros são classificados de acordo com as faixas de frequências que eles transmitem ou rejeitam. Um **filtro passa-baixas** (Figura 3.19(a)) tem uma banda de passagem que permite que todas as frequências de zero até uma determinada frequência sejam transmitidas. Um **filtro passa-altas** (Figura 3.19(b)) tem uma banda de passagem que permite que todas as frequências desde um determinado valor até o infinito sejam transmitidas. Um **filtro passa-faixa** (Figura 3.19(c)) permite que todas as frequências dentro de uma banda especificada sejam transmitidas. Um **filtro rejeita-faixa** (Figura 3.19(d)) impede que todas as frequências dentro de uma determinada banda sejam transmitidas. Em todos os casos, a frequência de corte é definida como sendo aquela em que a tensão de saída é 70,7% do valor na banda de passagem. O termo **atenuação** é usado para a razão entre as potências de saída e entrada, sendo ela expressa como a relação do logaritmo desta razão, sendo dada em unidades de bels. Visto que essa é uma unidade muito grande, os decibéis (dB) são usados e, desta forma, a atenuação em $\text{dB} = 10 \log(\text{potência de entrada}/\text{potência de saída})$. Como a potência na impedância é proporcional ao quadrado da tensão, a atenuação em $\text{dB} = 20 \log(\text{tensão de entrada}/\text{tensão de saída})$. A tensão de saída de 70,7% da tensão na banda de passagem corresponde a uma atenuação de 3 dB.

O termo **passivo** é usado para descrever um filtro constituído apenas de resistores, capacitores e indutores. O termo **ativo** é usado quando o filtro é consti-

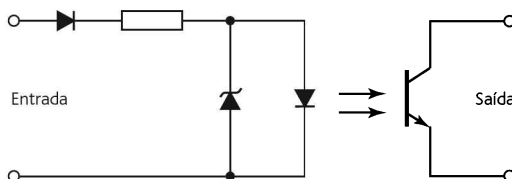


Figura 3.18 Circuito de proteção.

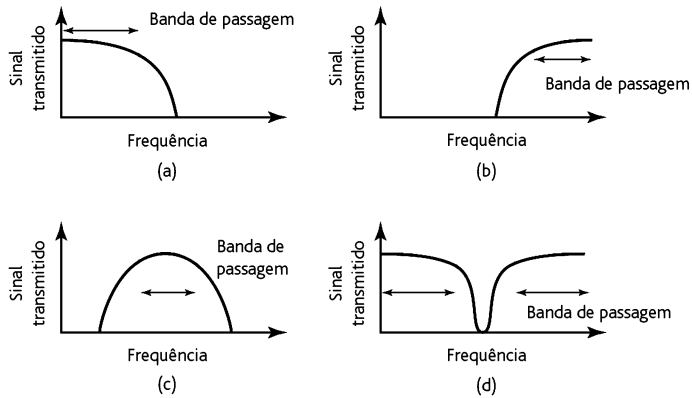


Figura 3.19 Características de filtros ideais: (a) passa-baixas, (b) passa-altas, (c) passa-faixa, (d) rejeita faixa.

tuído também de amplificador operacional. Os filtros passivos têm a desvantagem da corrente que é drenada pelo circuito a seguir poder variar as características de frequência do filtro. Este problema não ocorre com os filtros ativos.

Os filtros passa-baixas normalmente são usados como parte do condicionador de sinal. Isto porque a maioria das informações transmitidas é de baixa frequência. Como o ruído tende a ocorrer em frequências maiores, um filtro passa-baixas pode ser usado para bloqueá-lo. Portanto, um filtro passa-baixas pode ser selecionado com uma frequência de corte de 40 Hz, bloqueando assim qualquer sinal de interferência proveniente de fontes de alimentação CA e ruído em geral. A Figura 3.20 mostra uma configuração básica de um filtro passa-baixas.

3.5 Ponte de Wheatstone

A **ponte de Wheatstone** pode ser usada para converter uma variação de resistência em uma variação de tensão. A Figura 3.21 mostra a configuração básica desta ponte.

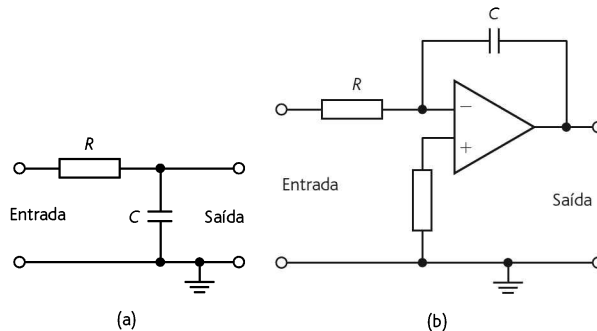


Figura 3.20 Filtro passa-baixas: (a) passivo, (b) ativo usando um amplificador operacional.

Quando a tensão de saída, V_o , é zero, então o potencial em B tem que ser igual ao de D. A diferença de potencial em R_1 , ou seja, V_{AB} , tem que ser igual à de R_3 , ou seja, V_{AD} . Portanto, $I_1 R_1 = I_2 R_2$. Isto significa também que a diferença de potencial em R_2 , ou seja, V_{BC} , tem que ser igual à de R_4 , ou seja, V_{DC} . Como não há corrente em BD, então a corrente de R_2 tem que ser igual à de R_1 , e a corrente de R_4 igual à de R_3 . Portanto, $I_1 R_2 = I_2 R_4$. Dividindo estas duas equações, obtemos

$$\frac{R_1}{R_2} = \frac{R_3}{R_4}$$

Nesta situação, dizemos que a ponte está em **equilíbrio**.

Agora considere o que acontece quando uma das resistências varia a partir desta condição de equilíbrio. A tensão de alimentação, V_s , é conectada entre os pontos A e C e, desta forma, a queda de tensão em R_1 é a fração $R_1/(R_1 + R_2)$ da tensão de alimentação. Portanto,

$$V_{AB} = \frac{V_s R_1}{R_1 + R_2}$$

De forma similar, a diferença de potencial em R_3 é

$$V_{AD} = \frac{V_s R_3}{R_3 + R_4}$$

Portanto, a diferença de potencial entre B e D, ou seja, a diferença de potencial na saída, V_o , é

$$V_o = V_{AB} - V_{AD} = V_s \left(\frac{R_1}{R_1 + R_2} - \frac{R_3}{R_3 + R_4} \right)$$

Esta equação fornece a condição de equilíbrio quando $V_o = 0$.

Considere que a resistência R_1 seja um sensor cuja resistência varia. Uma variação nesta resistência de R_1 para $R_1 + \delta R_1$ resulta em uma variação na saída de V_o para $V_o + \delta V_o$, em que

$$V_o + \delta V_o = V_s \left(\frac{R_1 + \delta R_1}{R_1 + \delta R_1 + R_2} - \frac{R_3}{R_3 + R_4} \right)$$

Portanto,

$$(V_o + \delta V_o) - V_o = V_s \left(\frac{R_1 + \delta R_1}{R_1 + \delta R_1 + R_2} - \frac{R_1}{R_1 + R_2} \right)$$

Se δR_1 for muito menor que R_1 , então a equação anterior pode ser aproximada para

$$\delta V_o \approx V_s \left(\frac{\delta R_1}{R_1 + R_2} \right)$$

Com esta aproximação, a variação na tensão de saída é proporcional à variação na resistência do sensor. Essa equação fornece a tensão de saída quando não há resistência de carga nela. Se houver uma, o efeito de carga tem que ser considerado.

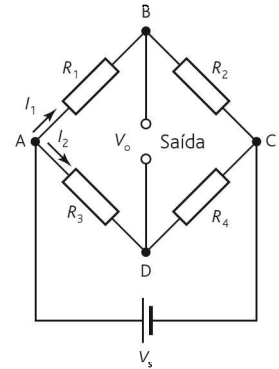


Figura 3.21 Ponte de Wheatstone.

Para ilustrar isto, considere um sensor de temperatura de resistência de platina que tem uma resistência, em 0°C, de 100 Ω e está situado em um ramo de uma ponte de Wheatstone. Nesta temperatura, a ponte está em equilíbrio, tendo cada um dos outros ramos uma resistência de 100 Ω. Se o coeficiente de temperatura da resistência de platina é de 0,0039/K, qual será a tensão de saída da ponte por grau de variação na temperatura se a carga na saída pode ser considerada infinita? A tensão de alimentação, com resistência interna desprezível, é de 6,0 V. A variação da resistência da platina com a temperatura pode ser representada por

$$R_t = R_0(1 + \alpha t)$$

em que R_t é a resistência em $t(^{\circ}\text{C})$, R_0 é a resistência em 0°C e α é o coeficiente de temperatura da resistência. Portanto,

$$\begin{aligned} \text{variação na resistência} &= R_t - R_0 = R_0\alpha t \\ &= 100 \times 0,0039 \times 1 = 0,39 \text{ } \Omega/\text{K} \end{aligned}$$

Como a variação de resistência é pequena em comparação com 100 Ω, a equação aproximada pode ser usada. Assim,

$$\delta V_o \approx V_s \left(\frac{\delta R_1}{R_1 + R_2} \right) = \frac{6,0 \times 0,39}{100 + 100} = 0,012 \text{ V}$$

3.5.1 Compensação de temperatura

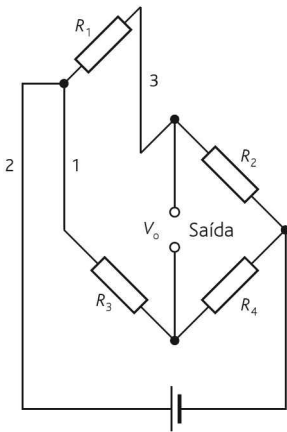


Figura 3.22 Compensação da resistência dos terminais.

Em muitas medições que envolvem um sensor resistivo, o elemento sensor real pode ter que estar na extremidade de longos terminais. Não apenas a resistência do sensor, mas a dos terminais será afetada pela variação na temperatura. Por exemplo, o sensor de temperatura de resistência de platina consiste de uma bobina de platina com terminais nas extremidades. Quando a temperatura varia, não apenas a resistência da bobina varia, mas também a dos terminais. Neste caso, o que interessa é apenas a resistência da bobina e, portanto, algum recurso deve ser empregado para compensar a resistência dos terminais da bobina. Um método para fazer isto é usar três terminais na bobina, como mostra a Figura 3.22. A bobina é conectada na ponte de Wheatstone de forma que o terminal 1 esteja em série com R_3 enquanto o terminal 3 esteja em série com a bobina de resistência de platina R_1 . O terminal 2 é conectado na fonte de alimentação. Qualquer variação na resistência de terminal certamente afetará os três terminais igualmente, visto que eles têm o mesmo material, diâmetro e comprimento e estão próximos entre si. O resultado é que variações na resistência de terminal ocorrem igualmente nos dois ramos da ponte e se cancelam se R_1 e R_3 forem resistências iguais.

O *strain gauge* de resistência elétrica é um outro sensor no qual a compensação tem que ser feita para efeitos de temperatura. O *strain gauge* varia a resistência quando o esforço aplicado nele varia. Infelizmente, ela também varia com a temperatura. Uma forma de eliminar o efeito da temperatura é usar um ***strain gauge de compensação***. Este é um *strain gauge* que é idêntico ao que é submetido a esforços,

o ativo, e é montado no mesmo material, mas não é submetido a esforços. Ele é posicionado próximo ao *strain gauge* ativo de forma que ele sofra a mesma variação de temperatura. Assim, a variação de temperatura causa variações nas resistências de cada um com a mesma intensidade. O *strain gauge* ativo é montado em um dos ramos da ponte de Wheatstone (Figura 3.23(a)), e o *strain gauge* de compensação é montado em outro ramo, de modo que os efeitos da temperatura que induzem variações de resistência se cancelam.

Os *strain gauges* frequentemente são usados com outros sensores, como células de carga, ou em medições de pressão com diafragmas medindo a quantidade de deslocamento ocorrido. Nestas situações, a compensação de temperatura ainda é necessária. Embora *strain gauges* de compensação possam ser usados, a melhor solução é usar quatro *strain gauges*. Dois deles são fixados de forma que os esforços aplicados a eles sejam de tração, os outros dois de forma que sejam de compressão. A célula de carga na Figura 3.23(b) mostra uma montagem como a citada. Os *strain gauges* sob tração têm a resistência aumentada, enquanto os que são submetidos a compressão têm as resistências diminuídas. Como os *strain gauges* são conectados nos quatro ramos da ponte de Wheatstone (Figura 3.23(b)), e visto que todos são igualmente afetados pelas variações de temperatura, este arranjo apresenta compensação de temperatura. Ele também fornece uma tensão de saída maior do que se tivesse um único *strain gauge*.

Para ilustrar isto, considere uma célula de carga com quatro *strain gauges* dispostos conforme a Figura 3.23 que usa uma ponte com quatro ramos ativos contendo *strain gauges*. As medições têm um fator *gauge* de 2:1 e uma resistência de 100 Ω . Quando a célula de carga é submetida a uma força de compressão, as medições verticais mostram compressão e, já que quando ocorre de um corpo ser pressionado há uma consequente extensão lateral, as medidas horizontais são submetidas a esforços de tração (a razão entre os esforços transversais e longitudinais é denominada de razão de Poisson e geralmente está em torno de 0,3). Portanto, se as medidas de compressão sofrerem um esforço de $-1,0 \times 10^{-5}$ e a de tração $+0,3 \times 10^{-5}$, a tensão de alimentação para a ponte for 6 V e a tensão de saída da ponte for amplificada por um circuito diferencial com amplificador operacional, qual será a razão entre a

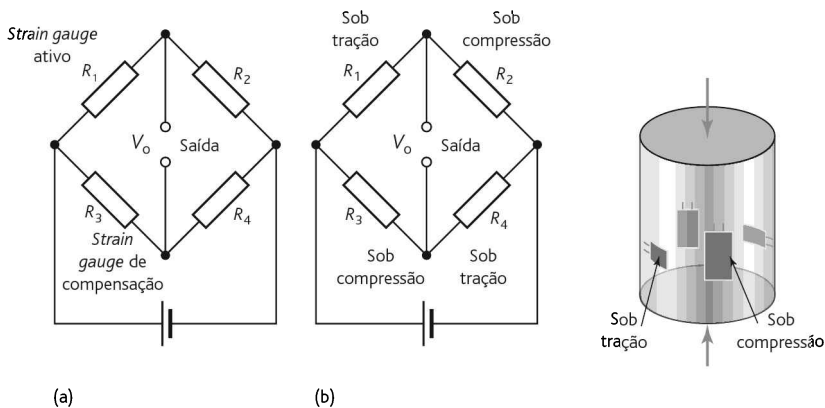


Figura 3.23 Compensação com *strain gauges*: (a) uso de um *strain gauge* de compensação, (b) uma ponte com quatro ramos ativos.

resistência de realimentação e as resistências nas duas entradas do amplificador se a carga produz uma saída de 1 mV?

A variação na resistência de um *strain gauge* sujeito a esforços de compressão é dada por $\Delta R/R = G\varepsilon$:

$$\begin{aligned}\text{variação na resistência} &= G\varepsilon R = -2,1 \times 1,0 \times 10^{-5} \times 100 \\ &= -2,1 \times 10^{-3} \Omega\end{aligned}$$

Para um *strain gauge* submetido a uma tração, temos

$$\begin{aligned}\text{variação na resistência} &= G\varepsilon R = 2,1 \times 0,3 \times 10^{-5} \times 100 \\ &= 6,3 \times 10^{-4} \Omega\end{aligned}$$

A diferença de potencial da ponte em desequilíbrio é dada por (veja a Seção 3.5)

$$\begin{aligned}V_o &= V_s \left(\frac{R_1}{R_1 + R_2} - \frac{R_3}{R_3 + R_4} \right) \\ &= V_s \left(\frac{R_1(R_3 + R_4) - R_3(R_1 + R_2)}{(R_1 + R_2)(R_3 + R_4)} \right) \\ &= V_s \left(\frac{R_1 R_4 - R_2 R_3}{(R_1 + R_2)(R_3 + R_4)} \right)\end{aligned}$$

Agora temos cada um dos resistores variando. Entretanto, podemos desprezar as variações em relação aos denominadores onde os efeitos destas variações sobre a soma das duas resistências é insignificante. Portanto,

$$V_o = V_s \left(\frac{(R_1 + \delta R_1)(R_4 + \delta R_4) - (R_2 + \delta R_2)(R_3 + \delta R_3)}{(R_1 + R_2)(R_3 + R_4)} \right)$$

Desprezando os produtos dos termos com δ , e como temos uma ponte inicialmente equilibrada com $R_1 R_4 = R_2 R_3$, então

$$V_o = \frac{V_s R_1 R_4}{(R_1 + R_2)(R_3 + R_4)} = \left(\frac{\delta R_1}{R_1} - \frac{\delta R_2}{R_2} - \frac{\delta R_3}{R_3} + \frac{\delta R_4}{R_4} \right)$$

Portanto,

$$V_o = \frac{6 \times 100 \times 100}{200 \times 200} \left(\frac{2 \times 6,3 \times 10^{-4} + 2 \times 2,1 \times 10^{-3}}{100} \right)$$

A saída é então $3,6 \times 10^{-5}$ V. Esta é a entrada do amplificador diferencial; portanto, usando a equação desenvolvida na Seção 3.2.5,

$$\begin{aligned}V_o &= \frac{R_2}{R_1}(V_2 - V_1) \\ 1,0 \times 10^{-3} &= \frac{R_2}{R_1} \times 3,6 \times 10^{-5}\end{aligned}$$

Portanto, $R_2/R_1 = 27,8$.

3.5.2 Compensação de termopar

Um termopar fornece uma FEM que depende da temperatura das duas junções (veja a Seção 2.9.5). Idealmente, se uma junção for mantida em 0°C, então a temperatura relativa à FEM pode ser lida diretamente das tabelas. Entretanto, isto nem sempre é possível, e a junção fria é frequentemente mantida na temperatura ambiente. Para compensar isto, uma diferença de potencial tem que ser somada à do termopar. Ela tem que ser igual à FEM que seria gerada pelo termopar com uma junção em 0°C e a outra na temperatura ambiente. Esta diferença de potencial pode ser gerada usando-se um sensor de temperatura de resistência em uma ponte de Wheatstone. A ponte está em equilíbrio em 0°C e a tensão de saída da ponte fornece a diferença de potencial de correção para as outras temperaturas.

A resistência de um sensor de temperatura com resistência de metal pode ser descrita pela relação

$$R_t = R_0(1 + \alpha t)$$

em que R_t é a resistência em t (°C), R_0 é a resistência em 0°C e α é o coeficiente de temperatura da resistência. Portanto,

$$\text{variação na resistência} = R_t - R_0 = R_0\alpha t$$

A tensão de saída da ponte, considerando R_1 como o sensor de temperatura de resistência, é dada por

$$\delta V_o \approx V_s \left(\frac{\delta R_1}{R_1 + R_2} \right) = \frac{V_s R_0 \alpha t}{R_0 + R_2}$$

A FEM do termopar, e , provavelmente varia com a temperatura t de forma razoavelmente linear ao longo da pequena faixa de temperatura considerada (de 0°C à temperatura ambiente). Portanto, $e = kt$, em que k é uma constante, ou seja, a FEM produzida por grau de variação na temperatura. Portanto, para a compensação temos

$$kt = \frac{V_s R_0 \alpha t}{R_0 + R_2}$$

e assim

$$kR_2 = R_0(V_s\alpha - k)$$

Para um termopar de ferro-constantan fornecendo 51 $\mu\text{V}/^\circ\text{C}$, a compensação pode ser gerada por um elemento de resistência de níquel com uma resistência de 10 Ω em 0°C e um coeficiente de temperatura da resistência de 0,0067/K, uma fonte de alimentação para a ponte de 1,0 V e R_2 igual a 1304 Ω .

3.6 Modulação por pulso

Um problema frequentemente encontrado quando se trabalha com a transmissão de sinais CC de baixa amplitude proveniente de sensores é que o ganho de um amplificador operacional usado para amplificá-los pode flutuar e, da mesma forma, a saída.

Este problema pode ser superado se o sinal for uma sequência de pulsos em vez de ser contínuo no tempo.

Uma forma de conversão é feita segmentando-se, ou recortando-se, o sinal CC da forma como sugere a Figura 3.24. A saída do circuito *chopper*, que segmenta o sinal, é uma cadeia de pulsos em que a amplitude de cada um está relacionada com o nível CC do sinal de entrada. Esse processo é denominado de **modulação por amplitude de pulso**. Após a amplificação e qualquer outro condicionamento do sinal, o sinal modulado pode ser demodulado retornando uma saída CC. Com a modulação por amplitude de pulso, a amplitude dos pulsos está relacionada ao nível da tensão CC.

A **modulação por largura de pulso (PWM – Pulse Width Modulation)** é bastante usada em sistemas de controle como uma forma de controle do valor médio de uma tensão CC. Portanto, se uma tensão analógica constante for segmentada em pulsos, através da variação da largura dos pulsos o valor médio da tensão pode ser alterado. A Figura 3.25 ilustra isto. O termo **ciclo de trabalho** é usado para a fração de cada ciclo para o qual a tensão é de nível alto. Portanto, para um sinal PWM onde o sinal é alto em metade de cada ciclo, o ciclo de trabalho é $\frac{1}{2}$ ou 50%. Se ele for apenas um quarto de cada ciclo, então o ciclo de trabalho é $\frac{1}{4}$ ou 25%.

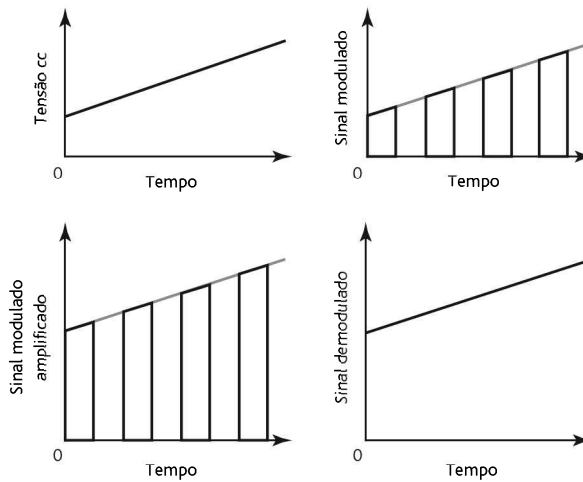


Figura 3.24 Modulação por amplitude de pulso.

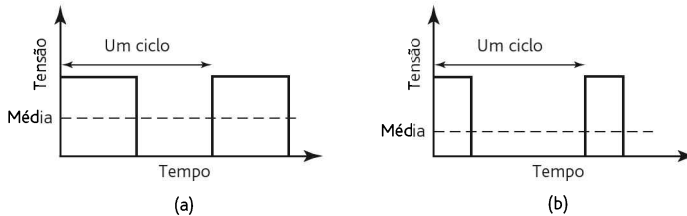


Figura 3.25 PWM para controle de tensão: (a) ciclo de trabalho de 50%, (b) ciclo de trabalho de 25%.

Resumo

O **condicionamento de sinal** pode envolver proteção para evitar danos ao próximo estágio do sistema, obtenção de um sinal na forma desejada, obtenção do nível correto do sinal, redução de ruído, manipulação para, por exemplo, torná-lo linear.

Os elementos usados normalmente no condicionamento de sinais são os **amplificadores operacionais**, que são amplificadores de alto ganho CC (da ordem de 100.000 ou mais)

A **proteção** contra tensões ou correntes altas pode envolver o uso de resistores e fusíveis; os diodos Zener pode ser usados para proteção contra polaridades invertidas e tensões altas. Os optoisoladores são usados para isolar os circuitos completamente, removendo todas as conexões elétricas entre eles.

Os **filtros** podem ser usados para remover uma determinada faixa de frequências a partir do sinal e permitir que outras faixas sejam transmitidas.

A **ponte de Wheatstone** pode ser usada para converter uma variação de resistência elétrica em variação de tensão.

Problemas

- 3.1 Projete um circuito com amplificador operacional que possa ser usado para gerar uma saída que varia de 0 a -5 V quando a entrada varia de 0 a 100 mV.
- 3.2 Um amplificador inversor tem uma resistência de entrada de $2\text{ k}\Omega$. Determine a resistência de realimentação necessária para que o circuito tenha um ganho de tensão de 100.
- 3.3 Projete um circuito amplificador somador que possa ser usado para gerar uma saída que varia de -1 a -5 V quando a entrada varia de 0 a 100 mV.
- 3.4 Um amplificador diferencial é usado com um sensor termopar na forma mostrada pela Figura 3.8. Que valores de R_1 e R_2 fazem com que o circuito tenha uma saída de 10 mV para uma diferença de temperatura de 100°C entre as junções de um termopar de cobre-constantan se considerarmos que ele tem uma sensibilidade constante de $43\text{ }\mu\text{V}/^\circ\text{C}$?
- 3.5 A saída de um sensor de pressão diferencial que é usado com uma placa de orifício para a medida de vazão é não linear, sendo a tensão de saída proporcional ao quadrado da vazão. Determine a característica necessária do elemento a ser colocado na malha de realimentação de um circuito condicionador de sinal com amplificador operacional para linearizar esta saída.
- 3.6 Um amplificador diferencial tem que apresentar um ganho de tensão de 100. Qual deve ser a resistência de realimentação necessária se as resistências de entrada são de $1\text{ k}\Omega$?
- 3.7 Um amplificador diferencial tem um ganho de tensão diferencial de 2.000 e um ganho em modo comum de 0,2. Qual é a razão de rejeição em modo comum (CMRR) dele em dB?

- 3.8 Os sinais digitais de um sensor estão “poluídos” por ruído e interferências em geral que são tipicamente da ordem de 100 V ou mais. Explique como se pode propiciar uma proteção para um microprocessador que recebe estes sinais?
- 3.9 Um sensor de temperatura com resistência de platina tem uma resistência de $120\ \Omega$ em 0°C e está localizado em um dos ramos de uma ponte de Wheatstone. Nesta temperatura, a ponte está em equilíbrio, tendo cada um dos outros ramos uma resistência de $120\ \Omega$. O coeficiente de temperatura da resistência de platina é $0,0039/\text{K}$. Qual será a tensão de saída da ponte para uma variação de 20°C na temperatura? A carga na saída pode ser considerada, em termos práticos, um circuito aberto, e a tensão que alimenta a ponte é proveniente de uma fonte de 6,0 V com resistência interna desprezível.
- 3.10 Um medidor de pressão com diafragma emprega quatro *strain gauges* para monitorar o movimento do diafragma. Estes quatro elementos constituem os ramos de uma ponte de Wheatstone, conforme a Figura 3.23. As medidas têm um fator *gauge* de 2,1 e resistência de $120\ \Omega$. Uma pressão diferencial aplicada no diafragma resulta em dois *strain gauges* de um lado do diafragma submetidos a um esforço de tração de $1,0 \times 10^{-5}$ e os dois do outro lado submetidos a um esforço de compressão de $1,0 \times 10^{-5}$. A tensão de alimentação da ponte é de 10 V. Qual é a tensão de saída da ponte?
- 3.11 Uma ponte de Wheatstone tem um único *strain gauge* em um ramo e resistores nos outros ramos, os quais têm a mesma resistência do *strain gauge* não submetido a esforços. Mostre que a tensão de saída da ponte é dada por $\frac{1}{4}V_s G \varepsilon$, em que V_s é a tensão de alimentação da ponte, G é o fator *gauge* do *strain gauge* e ε é a ação de esforço sobre ele.

4

Sinais Digitais

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Explicar os princípios e principais métodos de conversões analógico-digital e digital-analógico.
- Explicar os princípios e aplicações dos multiplexadores.
- Explicar os princípios do processamento de sinais digitais.

4.1 Sinais digitais

A saída da maioria dos sensores tende a ser no formato analógico, com a amplitude de saída relacionada à de entrada. Nos casos em que se aplica um microprocessador como parte do sistema de controle ou medição, a saída analógica de um sensor tem que ser convertida para o formato *digital* antes que ela possa ser usada como uma entrada do microprocessador. Do mesmo modo, a maioria dos atuadores opera com entradas analógicas e, portanto, a saída digital do microprocessador tem que ser convertida para a forma analógica antes de ser usada como entrada do atuador.

4.1.1 Números binários

O **sistema binário** é baseado em apenas dois símbolos ou estados, 0 e 1, sendo estes normalmente sinais de 0 V e 5 V. Estes são denominados dígitos binários (*binary digits*) ou **bits**. Quando um número é representado neste sistema, a posição do dígito no número indica o peso associado a cada dígito, com o peso aumentando por um fator de 2 conforme percorremos o número da direita para a esquerda:

...	2^3	2^2	2^1	2^0
	bit 3	bit 2	bit 1	bit 0

Por exemplo, o número decimal 15 é $2^0 + 2^1 + 2^2 + 2^3 = 1111$ no sistema binário. Em um número binário, o bit da posição 0 é denominado de **bit menos significativo** (LSB – *Least Significant Bit*) e o bit da posição de maior peso é o **bit mais significativo** (MSB – *Most Significant Bit*). Uma combinação de bits para representar um número é denominada **palavra**. Portanto, 1111 é uma palavra de 4 bits. Esta palavra pode ser usada para representar a amplitude de um sinal. O termo **byte** é usado para um grupo de 8 bits. Veja o Apêndice B para mais detalhes sobre números binários.

4.2 Sinais analógicos e digitais

Uma conversão analógico-digital envolve a conversão de sinais analógicos em palavras binárias. A Figura 4.1(a) mostra os elementos básicos de uma conversão analógico-digital.

A técnica usada envolve o uso de um sinal de clock que gera pulsos regularmente no tempo para o conversor analógico-digital (ADC – *Analogue-to-Digital Converter*), e a cada vez que ele recebe um pulso, faz a amostragem do sinal analógico. A Figura 4.1 ilustra esta conversão analógico-digital mostrando os tipos de sinais envolvidos nos diversos estágios. A Figura 4.1(b) mostra o sinal analógico, e a Figura 4.1(c) mostra o sinal de clock que fornece os sinais de tempo nos quais as amostras ocorrem. O resultado da amostragem é uma série de pulsos estreitos (Figura 4.1(d)).

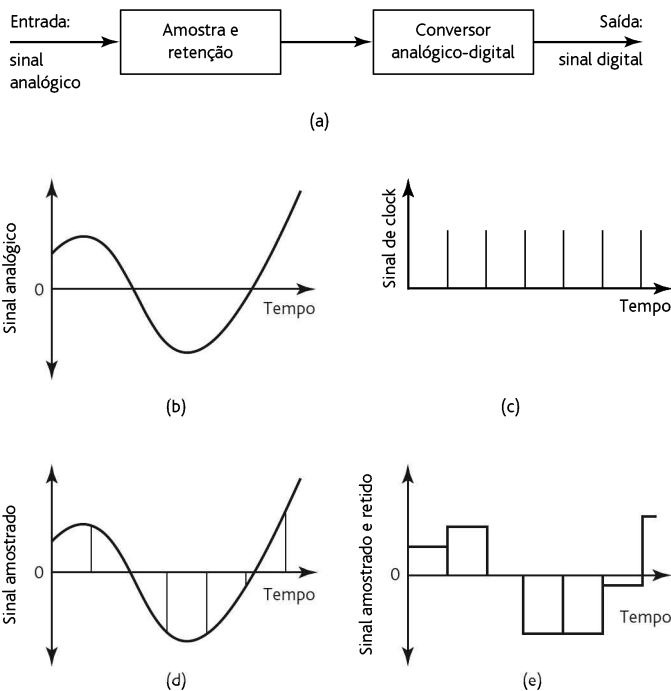


Figura 4.1 (a) Conversão analógico-digital, (b) entrada analógica, (c) sinal de clock, (d) sinal amostrado, (e) sinal amostrado e retido.

Uma unidade de **amostragem e retenção** é então usada para reter cada valor amostrado até que o próximo pulso ocorra, sendo o resultado mostrado na Figura 4.1(e). A unidade de amostragem e retenção é necessária porque o ADC requer uma quantidade finita de tempo, denominada **tempo de conversão**, para converter o sinal analógico em digital.

A relação entre a entrada amostrada e retida e a saída para o ADC é ilustrada pelo gráfico mostrado na Figura 4.2 para uma saída digital limitada a 3 bits. Com 3 bits, existem $2^3 = 8$ possíveis níveis de saída. Portanto, como a saída do ADC que representa a entrada analógica pode ter apenas um destes oito níveis possíveis, há uma faixa de valores de entrada para os quais a saída permanece inalterada. Os oito níveis de saída possíveis são denominados **níveis de quantização**, e a diferença entre dois níveis adjacentes de tensão analógica é denominado **intervalo de quantização**. Portanto, para o ADC dado na Figura 4.2, o intervalo de quantização é 1 V. Devido ao degrau natural da relação, a saída digital não é sempre proporcional à entrada analógica e, assim, há um erro que é denominado de **erro de quantização**. Quando a entrada se situa no centro do intervalo, o erro de quantização é zero, e o erro máximo é igual a metade de um intervalo, ou seja, $\pm 1/2$ bit.

O comprimento da palavra possibilita determinar a resolução do componente, ou seja, a menor variação na entrada que resulta em uma variação na saída digital. A menor variação na saída digital é 1 bit na posição do bit menos significativo da palavra, ou seja, o bit mais à direita. Portanto, com uma palavra de comprimento de n bits, a entrada analógica de fundo de escala, V_{FS} , é dividida em 2^n partes e, desta forma, a variação mínima na entrada que pode ser detectada, ou seja, a resolução, é $V_{FS}/2^n$.

Portanto, se temos um ADC com uma palavra de comprimento de 10 bits e a faixa do sinal analógico de entrada for 10 V, então o número de níveis para uma palavra de 10 bits é $2^{10} = 1024$ e a resolução é $10/1024 = 9,8$ mV.

Considere um termopar fornecendo uma saída de $0,5$ mV/°C. Qual será o tamanho necessário da palavra quando sua saída passa por um ADC se temperaturas de 0 a 200°C serão medidas com uma resolução de $0,5$ °C? A saída de fundo de escala do sensor é $200 \times 0,5 = 100$ mV. Com uma palavra de comprimento n , esta tensão



Figura 4.2 Entrada/saída de um ADC.

será dividida em degraus de $100/2^n$ mV. Para uma resolução de $0,5^\circ\text{C}$, o sistema tem que ser capaz de detectar um sinal do sensor de $0,5 \times 0,5 = 0,25$ mV. Portanto, precisamos de

$$0,25 = \frac{100}{2^n}$$

Então, $n = 8,6$. Assim, é necessário uma palavra de 9 bits.

4.2.1 Teorema da amostragem

Os ADCs amostram sinais analógicos em intervalos regulares e convertem estes valores em palavras binárias. Com que frequência um sinal analógico deve ser amostrado para obtermos uma saída que seja representativa deste sinal?

A Figura 4.3 ilustra o problema com diferentes taxas de amostragem para o mesmo sinal analógico. Quando o sinal é reconstruído a partir das amostras, o formato original do sinal é obtido apenas quando a taxa de amostragem for pelo menos duas vezes a maior frequência no sinal analógico. Este critério é conhecido com **critério de Nyquist** ou **teorema da amostragem de Shannon**. Quando a taxa de amostragem é menor do que duas vezes a maior frequência, a reconstrução pode representar um outro sinal analógico, gerando assim uma imagem falsa do sinal real. Isto é conhecido como *aliasing* (falseamento). O sinal mostrado na Figura 4.3(c) pode ser um com uma frequência muito menor do que a do sinal analógico que foi amostrado.

Sempre que um sinal for amostrado com uma frequência muito baixa, pode ocorrer uma falsa interpretação de componentes de alta frequência como originários de uma suposta frequência baixa. Um ruído de alta frequência também pode gerar erros no processo de conversão. Para minimizar erros devido ao *aliasing* e a ruídos de alta frequência, é usado um filtro passa-baixas antes do ADC que tem uma largura de banda que permite a passagem apenas das frequências que não produzirão erros de *aliasing*. Este filtro é denominado de **filtro anti-aliasing**.

4.2.2 Conversão digital-analógica

A entrada de um conversor digital-analógico (DAC) é uma palavra binária; a saída é um sinal analógico que representa a soma ponderada dos bits não nulos represen-

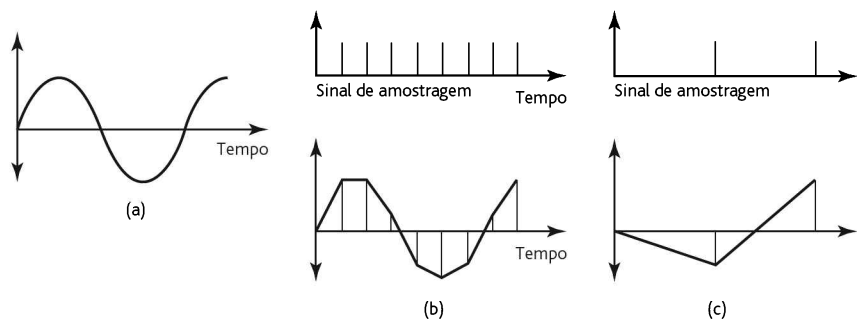


Figura 4.3 Efeitos da frequência de amostragem: (a) sinal analógico, (b) sinal amostrado, (c) sinal amostrado.

tados pela palavra. Portanto, por exemplo, um entrada de 0010 tem que ser uma saída analógica que é duas vezes aquela dada pela entrada 0001. A Figura 4.4 ilustra isto para uma entrada de um DAC com resolução de 1 V para palavras binárias não sinalizadas. Cada bit adicional aumenta a tensão de saída em 1 V.

Considere a situação em que um microprocessador fornece uma saída de 8 bits. Esta é transferida, através de um conversor digital-analógico, para uma válvula de controle. Esta válvula necessita de 6 V para abrir totalmente. Se o estado de abertura total for indicado por 1111111, qual será a saída correspondente a uma variação de 1 bit?

A tensão de saída de fundo de escala de 6,0 V é dividida em 2^8 intervalos. Uma variação de 1 bit corresponde a uma variação na tensão de saída de $6,0/2^8 = 0,023$ V.

4.3 Conversores digital-analógico e analógico-digital

As técnicas a seguir são encontradas em DACs e ADCs.

4.3.1 DACs

Uma forma simples de DAC usa um amplificador somador (veja a Seção 3.2.3) para produzir uma soma ponderada de todos os bits diferentes de zero na palavra de entrada (Figura 4.5). A tensão de referência é conectada aos resistores por chaves eletrônicas que respondem ao binário 1. Os valores das resistências de entrada dependem do bit da palavra ao qual a chave está respondendo; o valor do resistor para bits sucessivos, a partir do LSB, é dividido por 2. Portanto, a soma das tensões é uma soma ponderada dos dígitos na palavra. Este sistema é conhecido como **rede de resistores ponderados**. A função do circuito amp-op (amplificador operacional) é atuar como um buffer para garantir que a corrente de saída da rede de resistores não seja afetada pela carga de saída, e também para que o ganho possa ser ajustado em uma faixa apropriada de tensões de saída para uma determinada aplicação.

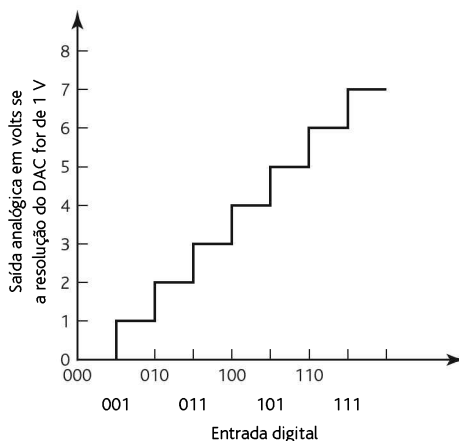


Figura 4.4 Entrada/saída para um DAC.

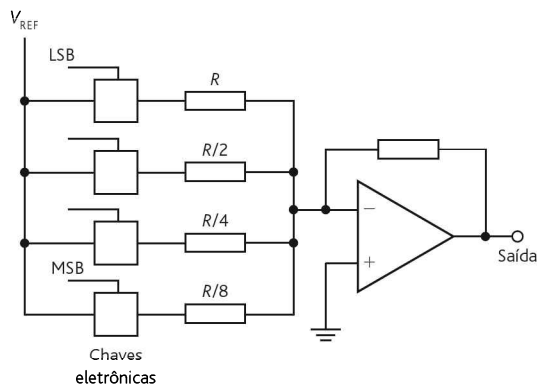


Figura 4.5 DAC com resistores ponderados.

Um problema com a rede de resistores ponderados é que se deve usar resistências de precisão e que os resistores são difíceis de serem obtidos ao longo de uma faixa ampla. Como consequência, este tipo de conversor se limita a conversões de 4 bits.

Outra versão, normalmente mais usada, emprega uma **rede escada R-2R** (Figura 4.6). Com esta versão, supera-se o problema da obtenção de resistores de precisão em uma ampla faixa de valores, sendo necessário apenas dois valores. A tensão de saída é gerada comutando-se seções do circuito escada para a tensão de referência ou para 0 V, conforme a entrada digital seja 1 ou 0.

A Figura 4.7 mostra detalhes do DAC GEC Plessey ZN558D com entrada com latch usando uma rede escada R-2R. Ao final da conversão, o resultado de 8 bits é colocado em um latch interno até o término da próxima conversão. Os dados são mantidos pelo latch quando a entrada HABILITAÇÃO for nível alto. Diz-se que o latch é transparente quando a entrada HABILITAÇÃO for nível baixo. Um latch é apenas um dispositivo para retenção da saída até que uma outra substitua a anterior. Quando um DAC tem um latch, ele pode ser interfaceado diretamente com o barramento de dados de um microprocessador e tratado por ele como apenas um endereço para enviar dados. Um DAC sem latch deve ser conectado ao barramento via

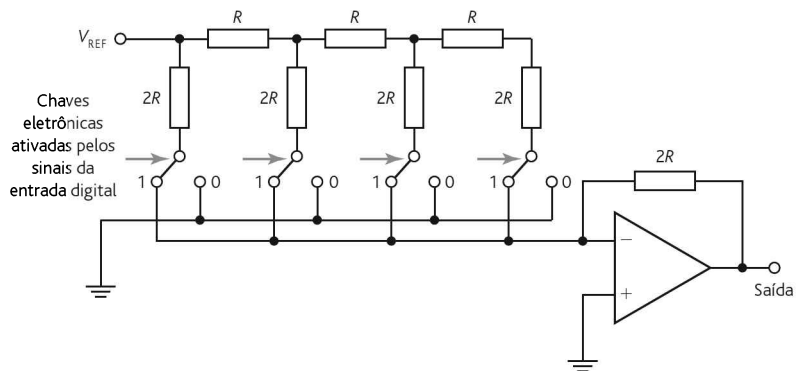


Figura 4.6 DAC do tipo escada R-2R.

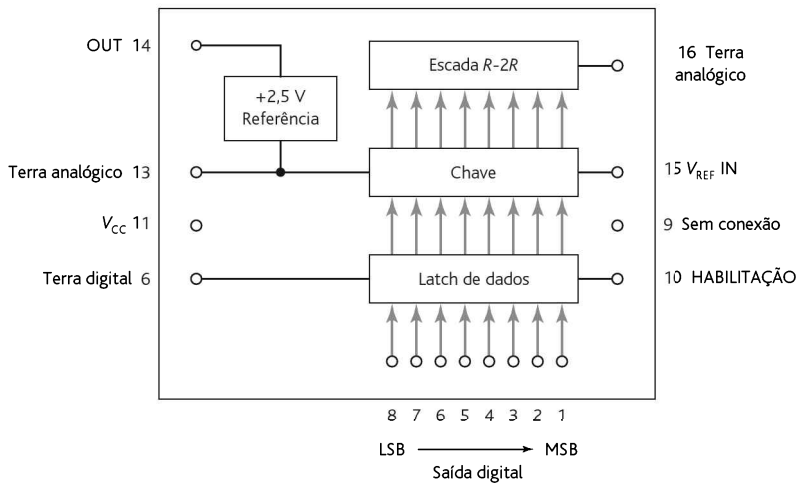


Figura 4.7 DAC ZN558D.

um adaptador de interface paralela (PIA – *Peripheral Interface Adapter*), que é um dispositivo que possui a função latch (veja a Seção 21.4). A Figura 4.8 mostra como o ZN558D pode ser usado com um microprocessador quando é necessário que a tensão de saída varie entre zero e a tensão de referência, o que é denominado **operação unipolar**. Com $V_{ref\ in} = 2,5\text{ V}$, a faixa de operação da saída é +5 V quando $R_1 = 8\text{ k}\Omega$ e $R_2 = 8\text{ k}\Omega$, e a faixa de operação é +10 V quando $R_1 = 16\text{ k}\Omega$ e $R_2 = 5,33\text{ k}\Omega$.

As especificações de DACs incluem termos como:

- 1 A *saída de fundo de escala*, ou seja, a saída quando a palavra de entrada possui todos os bits uns. No caso do ZN558D, este parâmetro é tipicamente 2,550 V.

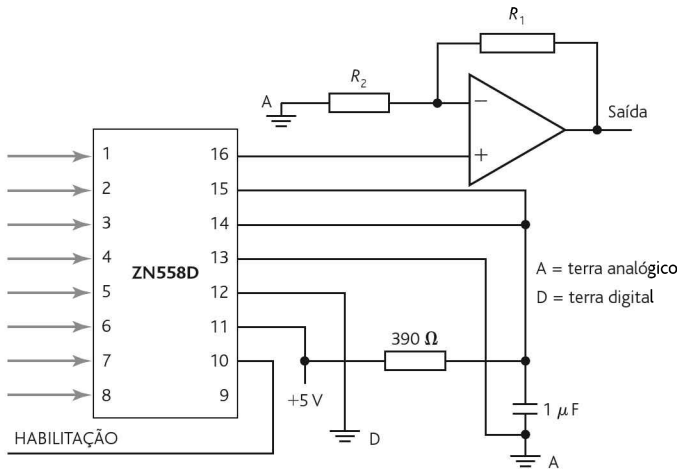


Figura 4.8 Operação unipolar.

- 2 A *resolução*, sendo os DACs de 8 bits geralmente satisfatórios para a maioria dos sistemas de controle microprocessados. O ZN558D é de 8 bits.
- 3 O *tempo de acomodação*, que é o tempo que o DAC gasta para alcançar dentro de $\frac{1}{2}$ LSB de sua nova tensão após uma mudança binária. Para o ZN558D, este parâmetro vale 800 ns.
- 4 A *linearidade*, que é o máximo desvio da linha reta entre zero e o ponto de operação máximo na saída. Este é no máximo $\pm 0,5$ LSB para o ZN558D.

4.3.2 ADCs

A entrada de um ADC é um sinal analógico, e a saída é uma palavra binária que representa o nível do sinal de entrada. Existem ADCs que utilizam alguns métodos, sendo os mais comuns o de aproximações sucessivas, o da rampa, o dupla rampa e o flash.

O método de **aproximações sucessivas** é provavelmente o mais usado. A Figura 4.9 ilustra os subsistemas envolvidos. Uma tensão é gerada por um clock que emite uma sequência regular de pulsos que são contados, de forma binária, e a palavra binária resultante convertida em uma tensão analógica por um DAC. Esta tensão aumenta em degraus e é comparada com a tensão de entrada analógica proveniente do sensor. Quando a tensão gerada pelo DAC ultrapassa a tensão analógica de entrada, a porta é fechada. A saída do contador neste instante é a representação digital da tensão analógica de entrada. Enquanto a comparação poderia ser realizada começando a contagem em 1, o LSB, e então crescendo bit a bit, um método mais rápido é o de aproximações sucessivas. Este método envolve a seleção do MSB que é menor do que o valor analógico, acrescentando em seguida os bits menos significativos para os quais o total não exceda ao valor analógico. Por exemplo, podemos iniciar a comparação com 1000. Se este for maior, tentamos 0100. Se este for menor, tentamos 0110. Se este for maior, tentamos 0101. Como cada um dos bits da palavra é testado em sequência, com uma palavra de n bits se gastam n passos para fazer a comparação. Portanto, se o clock tem uma frequência f , o tempo entre pulsos é $1/f$. Assim, o tempo gasto para gerar a palavra, ou seja, o tempo de conversão, é n/f .

A Figura 4.10 mostra a forma típica de um ADC de 8 bits (GEC Plessey ZN439) projetado para ser usado com microprocessadores e empregando-se o método de aproximações sucessivas. A Figura 4.11 mostra como ele pode ser conectado de forma que seja controlado por um microprocessador e envie sua saída digital ao mi-

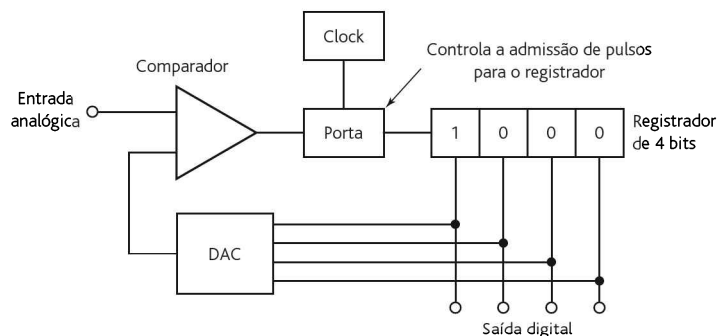


Figura 4.9 ADC de aproximações sucessivas.

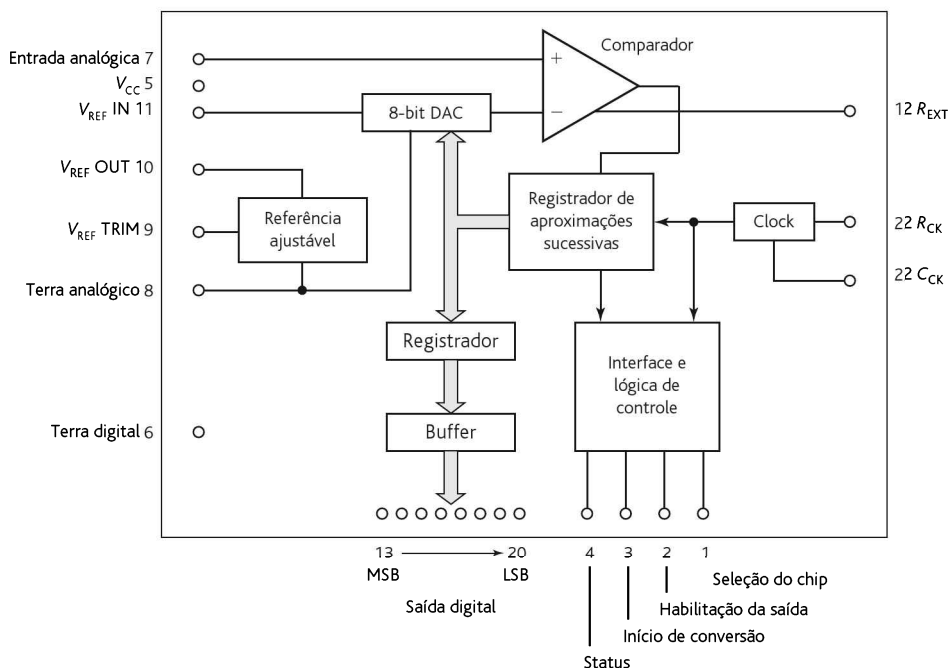


Figura 4.10 ADC ZN439.

croprocessador. Todo o circuito ativo, incluindo o clock, está contido em um único chip. O ADC é selecionado primeiro colocando-se o pino de seleção de chip em nível baixo. Quando o pino de início de conversão (*start conversion*) recebe um pulso

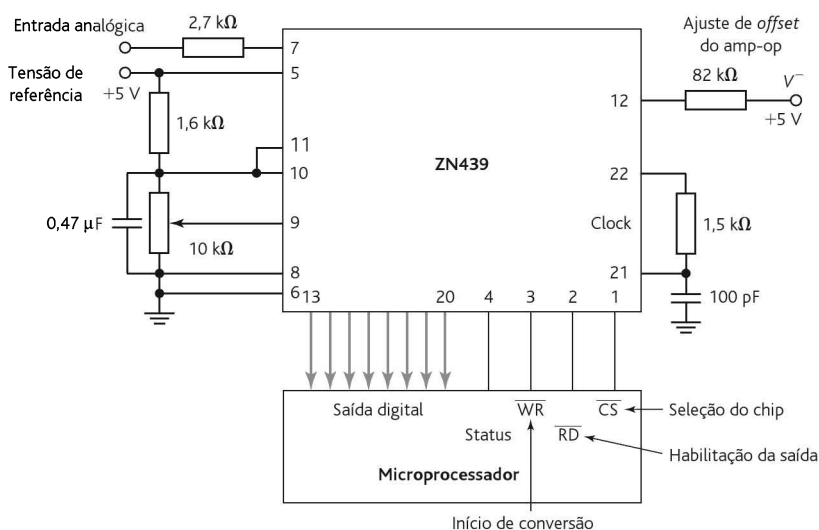


Figura 4.11 ZN439 conectado a um microprocessador.

negativo (borda de descida), a conversão começa. No final da conversão, o pino de status vai para nível baixo. A saída digital é enviada para o buffer interno, onde ela é armazenada até que seja lida como resultado do pino de habilitação da saída ser levado ao nível baixo.

O ADC do tipo **rampa** envolve uma tensão analógica que é aumentada a uma taxa constante, mais conhecida como tensão em rampa, e aplicada a um comparador, sendo lá comparada com a tensão analógica proveniente do sensor. O tempo gasto para a tensão em rampa aumentar até o valor da tensão do sensor depende da amplitude da tensão analógica amostrada. Quando a tensão na forma de rampa inicia, uma porta permite o início da contagem de pulsos regulares de um clock por um contador binário. Quando as duas tensões forem iguais, a porta inibe a passagem dos pulsos e a palavra binária indicada pelo contador é a representação digital da tensão analógica amostrada. A Figura 4.12 indica os subsistemas envolvidos no ADC do tipo rampa.

O **conversor de dupla rampa** é mais comum do que o de rampa simples. A Figura 4.13 mostra o circuito básico. A tensão analógica é aplicada a um integrador cuja saída é a entrada de um comparador. A saída deste comparador vai para nível alto logo que a saída do integrador for poucos milivolts maior do que GND. Quando a saída do comparador for nível alto, uma porta AND permite a passagem de pulsos para um contador binário. Este contador conta até o *overflow* (transbordamento). Então, o contador retorna para zero, envia um sinal para uma chave que desconecta a tensão desconhecida e conecta uma tensão de referência, e inicia a contagem novamente. A polaridade da tensão de referência é oposta à da tensão de entrada. Assim, a tensão do integrador diminui a uma taxa proporcional à tensão de referência. Quando a saída do integrador chega a zero, a saída do comparador vai para nível baixo, levando a entrada da AND para nível baixo e impedindo a passagem do clock. Então, a contagem do contador representa a medida da tensão analógica de entrada. ADCs de dupla rampa têm uma excelente rejeição a ruído porque a ação de integração tira a média das contribuições negativas e positivas aleatórias ao longo de um período de amostragem. Entretanto, eles são muito lentos.

O **ADC flash** é muito rápido. Para um conversor de n bits, são necessários $2^n - 1$ comparadores de tensão em paralelo, tendo cada um a tensão de entrada analógica como uma das entradas (Figura 4.14). Uma tensão de referência é aplicada a uma escada de resistores de forma que a tensão aplicada na outra entrada de cada

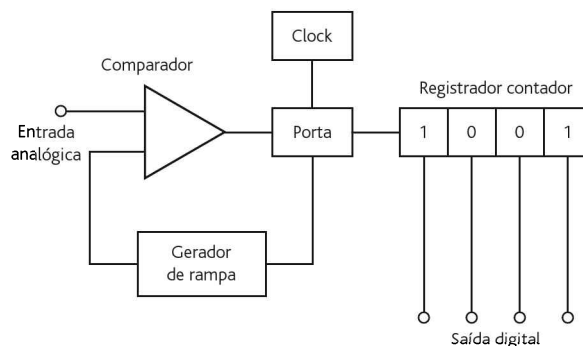


Figura 4.12 ADC do tipo rampa.

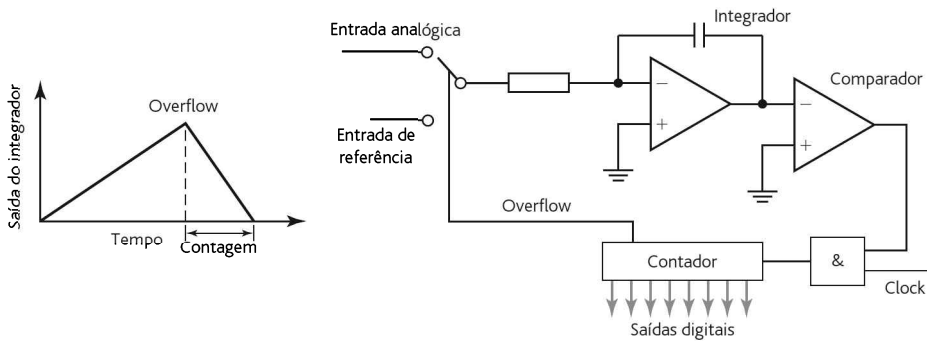


Figura 4.13 ADC de dupla rampa.

comparador tenha uma amplitude que equivale a um bit em relação à tensão aplicada ao comparador anterior. Portanto, quando a tensão analógica é aplicada ao ADC, em todos os comparadores nos quais a tensão analógica for maior que a tensão de referência deles a saída correspondente é nível alto, e naqueles em que ela for menor, a saída correspondente é nível baixo. Estas saídas em paralelo passam por um sistema de portas lógicas que as convertem em uma palavra digital.

Ao considerarmos as especificações de ADCs, os termos a seguir são encontrados:

- 1 *Tempo de conversão*, ou seja, o tempo necessário para completar uma conversão do sinal de entrada. Este parâmetro estabelece a frequência superior do sinal que pode ser amostrado sem aliasing; a frequência máxima é $1/(2 \times \text{tempo de conversão})$.
- 2 *Resolução*, que é o valor do sinal de fundo de escala dividido por 2^n , em que n é o número de bits. Normalmente é especificada apenas pelo número de bits.
- 3 *Erro de linearidade*, que é o desvio a partir de uma linha reta de zero ao valor de fundo de escala. Ele é no máximo de $\pm 1/2$ LSB.

A Tabela 4.1 apresenta algumas especificações detalhadas de ADCs de uso comum.

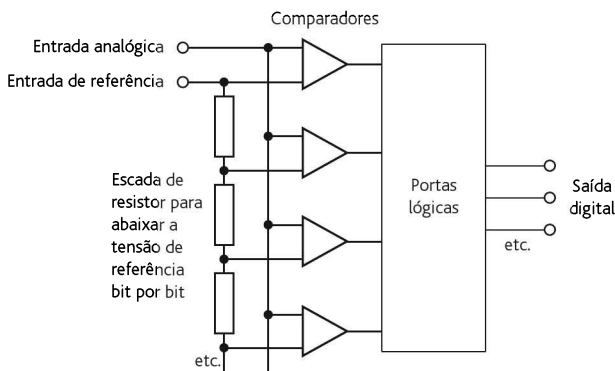


Figura 4.14 ADC flash.

Tabela 4.1 ADCs

ADC	Tipo	Resolução (bits)	Conversão (ns)	Erro de linearidade (LSB)
ZN439	SA	8	5.000	$\pm 1/2$
ZN448E	SA	8	9.000	$\pm 1/2$
ADS7806	SA	12	20.000	$\pm 1/2$
ADS7078C	SA	16	20.000	$\pm 1/2$
ADC302	F	8	20	$\pm 1/2$

AS = aproximação sucessiva, F = flash

4.3.3 Amplificadores de amostragem e retenção

Um ADC gasta um tempo finito para converter um sinal analógico em digital e, caso este sinal varie durante o tempo de conversão, podem surgir problemas. Para evitar isto, é usado um sistema de amostra e retenção para amostrar o sinal analógico e mantê-lo enquanto a conversão ocorre.

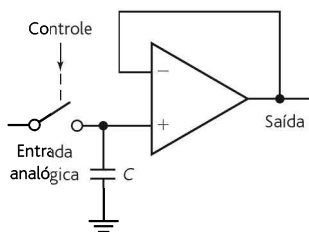


Figura 4.15 Circuito de amostragem e retenção.

O circuito básico (Figura 4.15) consiste de uma chave eletrônica para se obter a amostra, com um capacitor para reter o valor e um amplificador operacional na configuração seguidor de tensão. A chave eletrônica é controlada de modo que a amostra é obtida no instante determinado pela entrada de controle. Quando a chave é fechada, a tensão de entrada é aplicada no capacitor e a tensão de saída passa a ter o mesmo valor. Se a tensão de entrada variar enquanto a chave estiver fechada, a tensão no capacitor e na saída varia da mesma forma. Quando a chave é aberta, o capacitor retém sua carga e a tensão de saída permanece igual à tensão de entrada no instante em que a chave foi aberta. Portanto, a tensão é mantida até que a chave seja fechada novamente. O tempo necessário para que o capacitor carregue com o valor de uma nova amostra da tensão analógica de entrada é denominado **tempo de aquisição** e depende do valor da capacitância e da resistência do circuito quando a chave estiver fechada. Este tempo tem valores típicos da ordem de 4 μ s.

4.4 Multiplexadores

Um **multiplexador** é um circuito que recebe dados através dos seus vários canais de entrada e então, através da seleção de um destes canais, produz uma saída a partir de apenas um deles. Em aplicações onde há a necessidade de se fazer medições de fontes diferentes, em vez de se usar um ADC e um microprocessador para cada medição pode-se usar um multiplexador para selecionar uma entrada de cada vez e enviá-la a um ADC e um microprocessador (Figura 4.16). O multiplexador é essencialmente um dispositivo de chaveamento eletrônico que por sua vez habilita cada uma das entradas a ser amostrada.

Como uma ilustração dos tipos de multiplexadores disponíveis, o DG508ACJ tem oito canais de entrada com um endereço de 3 bits para fins de seleção do canal. O tempo de transição entre as amostras é 0,6 μ s.

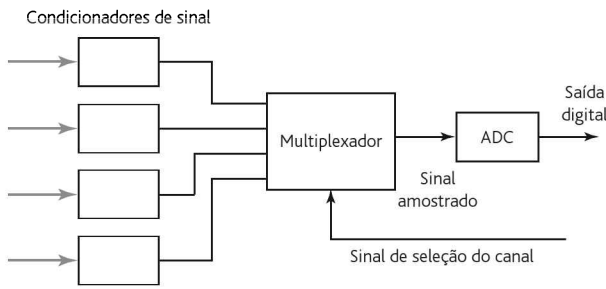


Figura 4.16 Multiplexador.

4.4.1 Multiplexador digital

A Figura 4.17 mostra o princípio básico de um multiplexador que pode ser usado para selecionar entradas de dados digitais; por questão de simplicidade, é mostrado apenas um sistema com dois canais de entrada. O nível lógico aplicado na entrada de seleção determina qual porta AND é habilitada de modo que o dado em sua entrada passe pela porta OR para a saída (veja o próximo capítulo para um estudo destas portas). Diversos tipos de multiplexadores são disponibilizados em encapsulamentos integrados. O tipo 151 habilita uma das oito linhas selecionáveis, o tipo 153 habilita uma das quatro linhas que é selecionada pelos dados em duas outras linhas, o tipo 157 habilita uma dentre duas entradas que é selecionada conforme os dados em quatro linhas.

4.4.2 Multiplexação por divisão de tempo

Muitas vezes é necessário que diversos dispositivos periféricos compartilhem as mesmas linhas de entrada/saída de um microprocessador. Para que cada periférico possa receber dados diferentes, é necessário alocar cada um em um *slot* (ou segmento) de tempo durante o qual os dados são transmitidos. Isto é denominado **multiplexação por divisão de tempo** (TDM – *Time Division Multiplexing*). A Figura 4.18 ilustra como esta técnica pode ser usada para acionar dois displays. Na Figura 4.18(a) o sistema não é multiplexado no tempo, diferentemente de (b).

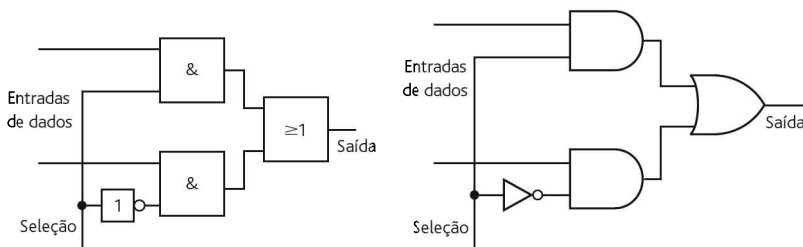


Figura 4.17 Multiplexador de dois canais.

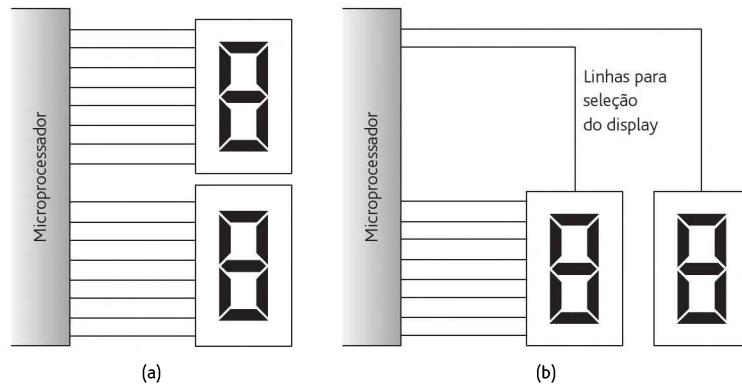


Figura 4.18 Multiplexação por divisão de tempo.

4.5 Aquisição de dados

O termo **aquisição de dados**, ou **DAQ** (*data acquisition*), é usado no processo de obtenção de dados a partir de sensores e o envio deles a um computador para processamento. Os sensores são conectados, geralmente via algum condicionador de sinal, em uma placa de aquisição de dados que é conectada a um computador (Figura 4.19(a)). A placa de aquisição de dados é uma placa de circuito impresso que, para entradas analógicas, geralmente é composta de multiplexador, amplificador, conversor analógico-digital, registradores e circuito de controle, de maneira que os sinais digitais amostrados são enviados ao sistema de computador. A Figura 4.19(b) mostra os elementos básicos desta placa.

Um software do computador é usado para controlar a aquisição de dados via placa DAQ. Quando o programa necessita de uma entrada de um determinado sensor, ele ativa a placa enviando uma palavra de controle para controlar o registrador de status. Esta palavra indica o tipo de operação que a placa tem que realizar. Como consequência, a placa comuta o multiplexador para o canal de entrada desejado. A entrada do sensor conectado a este canal é então enviada ao ADC via amplificador. Após a conversão, o sinal digital resultante é enviado ao registrador de dados, e a palavra no registrador de controle e status muda indicando que o sinal chegou. Em seguida a este sinal de aviso, o computador emite um sinal para que os dados

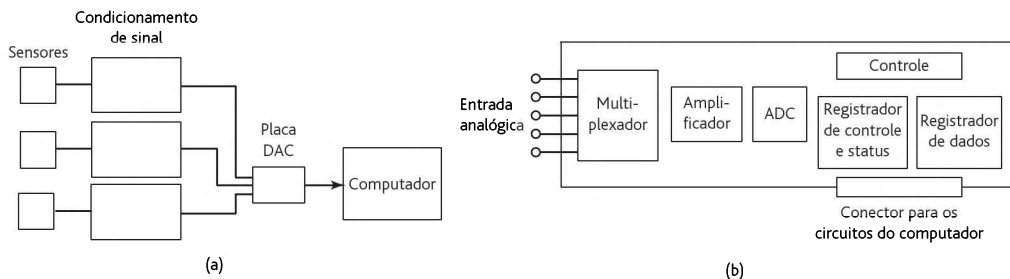


Figura 4.19 Sistema de aquisição de dados.

sejam lidos e recebidos por ele para serem processados. Este sinal de aviso é necessário para garantir que o computador não fique esperando, sem fazer nada, a placa realizar a aquisição dos dados, mas ele serve para avisar ao computador quando a aquisição está completa e, desta forma, o computador pode interromper o programa que está executando, ler os dados do DAQ e então continuar com o programa que foi interrompido. Um sistema mais rápido não envolve o computador na transferência dos dados para a memória, mas transfere os dados da placa de aquisição diretamente para a memória sem envolver o computador. Esta operação é denominada **acesso direto da memória** (DMA – *Direct Memory Access*).

As especificações para uma placa DAQ incluem a taxa de amostragem para entradas analógicas, que pode ser de 100 kS/s (100.000 amostras [*samples*] por segundo). O critério de Nyquist para a amostragem indica que se a máxima frequência do sinal analógico que pode ser amostrado pela placa é 50 kHz, a taxa de amostragem tem que ser duas vezes a componente de maior frequência do sinal. Além destas funções básicas de uma placa DAQ, ela também pode fornecer saídas analógicas, temporizadores e contadores que podem ser usados para fornecer sinais de disparo para o sistema do sensor.

A Figura 4.20 mostra um exemplo de uma placa multifunção de baixo custo para uso com computadores IBM. Esta figura mostra a estrutura organizacional básica da placa de aquisição de dados PC-LPM-16 da National Instruments. Esta placa tem 16 canais de entrada analógicas, uma taxa de amostragem de 50 kS/s, uma entrada digital de 8 bits e uma saída digital de 8 bits, e um contador/temporizador que pode gerar uma saída. Os canais podem ser acessados em uma sequência, obtendo uma leitura de cada canal por vez, ou pode-se acessar continuamente um único canal.

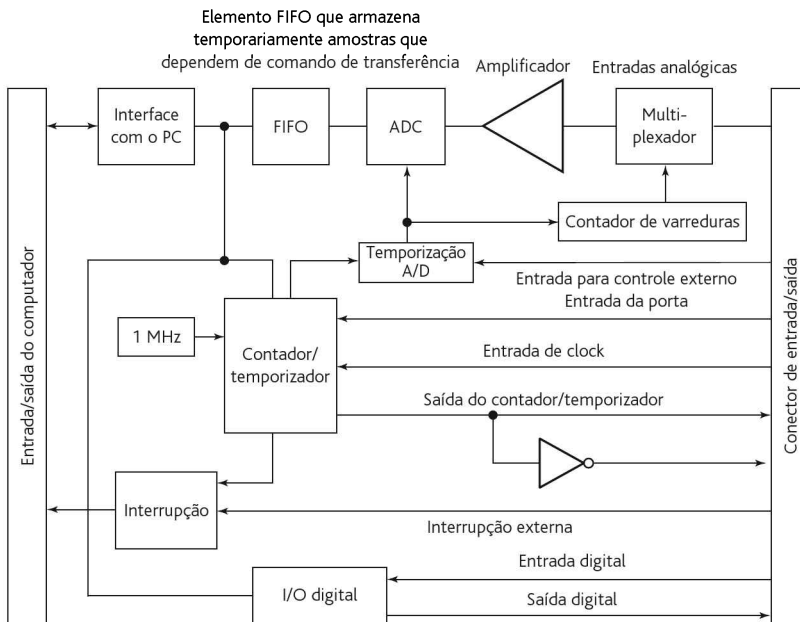


Figura 4.20 Placa DAQ PC-LPM-16.

4.5.1 Precisão dos dados

Uma vantagem do processamento de sinais digitais é que duas faixas de tensões são usadas em vez de dois níveis de tensão para distinguir entre dois estados binários para cada bit. Portanto, a precisão dos dados é menos afetada pelo ruído, pela deriva, pela tolerância dos componentes e por outros fatores que provocam flutuações nas tensões que são críticas para transmissão como tensões analógicas. Por exemplo, em um sistema de 5 V, a diferença entre os dois estados binários é tipicamente de 3 V no mínimo. Assim, dois sinais podem ser 0 e 5 V ou 1 V e 4 V e ainda ser distinguidos como 0 e 1.

4.5.2 Método da paridade para a detecção de erro

O movimento de dados digitais de um local para outro pode resultar em erros de transmissão, em que o receptor não recebe o mesmo sinal transmitido em virtude de um ruído elétrico no processo de transmissão. Muitas vezes, um pulso de ruído pode ser grande o suficiente em algum ponto de forma a alterar o nível lógico do sinal. Por exemplo, a sequência 1001 pode ser transmitida e recebida como sendo 1101. Para detectar erros como este, frequentemente é usado um **bit de paridade**. Este é um bit extra, 0 ou 1, acrescentado ao grupo de bits do código na transmissão. No caso da **paridade par**, o valor deste bit é escolhido de forma que o número total de uns no grupo, incluindo o bit de paridade, seja um número par. Por exemplo, na transmissão de 1001, o bit de paridade usado deve ser 0, resultando em 01001, que tem um número par de uns. Na transmissão de 1101, o bit de paridade usado deve ser 1, resultando em 11101, que tem um número par de uns. No caso da **paridade ímpar**, o bit de paridade é escolhido de forma que o número total de uns, incluindo o bit de paridade, seja ímpar. Portanto, se no receptor o número de uns em um grupo de bits de um código não tiver a paridade esperada, saberemos que há um erro, podendo ser solicitada uma retransmissão.

Uma extensão da verificação de paridade é o *sum check* (verificação de somas), no qual blocos de código podem ser verificados enviando-se uma série de bits que representam suas somas binárias. A paridade e o sum check podem detectar apenas erros simples em blocos de código; erros duplos não são detectados. Além disso, não é identificado o local do erro de forma que a correção possa ser feita pelo receptor. Os métodos e as técnicas de detecção de múltiplos erros para definir os erros têm que ser projetados.

4.6 Processamento de sinais digitais

O termo **processamento de sinais digitais** ou **processamento de sinais discretos no tempo** é usado para o processamento aplicado a um sinal por um microprocessador. Os sinais digitais são sinais discretos no tempo porque não são funções contínuas no tempo, mas existem apenas em tempos discretos. Considerando que o condicionamento de sinais analógicos requer componentes como amplificadores e circuitos de filtro, o condicionamento de sinais digitais pode ser realizado por um programa aplicado a um microprocessador, ou seja, que processe o sinal. Para alterar as características de um filtro usado com sinais analógicos, é necessário alterar componentes de hardware, ao passo que para alterar as características de um filtro digital basta alterar o software, ou seja, o programa com as instruções que o microprocessador executa.

Em um sistema de processamento de sinais digitais há uma entrada de uma palavra que representa a amplitude do pulso e a saída de uma outra palavra. O pulso de saída em um determinado instante é calculado pelo sistema como resultado do processamento do pulso de entrada, juntamente com os pulsos de entrada anteriores e possivelmente saídas anteriores do sistema.

Por exemplo, o programa usado pelo microprocessador pode ler o valor atual da entrada e somá-lo ao valor anterior da saída para obter uma nova saída. Se considerarmos a entrada atual como sendo o pulso de ordem k na sequência de entrada de pulsos, podemos representar esses pulsos como $x[k]$. A saída de ordem k na sequência de pulsos pode ser representada por $y[k]$. A saída anterior, ou seja, o pulso $(k - 1)$, pode ser representado por $y[k - 1]$. Portanto, podemos descrever o programa que fornece uma saída obtida pela soma do valor atual de entrada com a saída anterior como a seguir

$$y[k] = x[k] + y[k - 1]$$

Esta equação é denominada **equação de diferença**. Ela fornece a relação entre a saída e a entrada para um sistema discreto no tempo e é comparável com a equação diferencial que é usada para descrever a relação entre a saída e a entrada em um sistema contendo entradas e saídas que variam continuamente no tempo.

No caso da equação de diferença anterior, suponha que temos uma entrada de uma onda senoidal amostrada com a seguinte sequência de pulsos:

$$0,5, 1,0, 0,5, -0,5, -1,0, -0,5, 0,5, 1,0, \dots$$

O pulso de entrada $k = 1$ tem amplitude de 0,5. Se considerarmos que a saída anterior era zero, então $y[k - 1] = 0$ e assim $y[1] = 0,5 + 0 = 0,5$. O pulso de entrada $k = 2$ tem uma amplitude de 1,0, e assim $y[2] = x[2] + y[2 - 1] = 1,0 + 0,5 = 1,5$. O pulso de entrada $k = 3$ tem uma amplitude de 0,5, e assim $y[3] = x[3] + y[3 - 1] = 0,5 + 1,5 = 2,0$. O pulso de entrada $k = 4$ tem uma amplitude de -0,5, e assim $y[4] = x[4] + y[4 - 1] = -0,5 + 2,0 = 1,5$. O pulso de entrada $k = 5$ tem uma amplitude de -1,0, e assim $y[5] = x[5] + y[5 - 1] = -1,0 + 1,5 = 0,5$. Portanto, a saída de pulsos é

$$0,5; 1,5; 2,0; 1,5; 0,5; \dots$$

Podemos continuar desta forma até obter a saída para todos os pulsos.

Como um outro exemplo de uma equação de diferença, temos

$$y[k] = x[k] + ay[k - 1] - by[k - 2]$$

A saída é o valor do pulso atual de entrada a vezes a saída anterior menos b vezes a penúltima saída. Se tivermos $a = 1$ e $b = 0,5$, e considerando a entrada como o sinal senoidal amostrado apresentado anteriormente, então a saída será

$$0,5; 1,5; 1,75; 0,5; -1,37; \dots$$

Podemos ter uma equação de diferenças que produz uma saída que é similar à que obteríamos ao integrar um sinal contínuo no tempo. A integração de um sinal contínuo no tempo entre dois instantes pode ser considerada a área sob a função contínua no tempo entre os dois instantes. Portanto, se considerarmos dois sinais discretos no tempo $x[k]$ e $x[k - 1]$ ocorrendo em intervalos de tempo T entre eles (Figura 4.21), a variação na área é $\frac{1}{2}T(x[k] + x[k - 1])$. Portanto, se

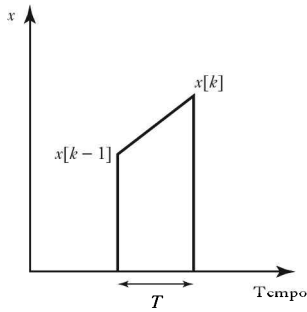


Figura 4.21 Integração.

a saída for a soma da área anterior com esta variação na área, a equação de diferença é

$$y[k] = y[k - 1] + \frac{1}{2}T(x[k] + x[k - 1])$$

Esta é conhecida como *aproximação de Tustin* para a integração.

A diferenciação pode ser aproximada pela determinação da taxa na qual a entrada varia. Portanto, quando a entrada varia de $x[k - 1]$ a $x[k]$ no tempo T , a saída é

$$y[k] = (x[k] - x[k - 1])/T$$

Resumo

A **conversão analógico-digital** envolve a conversão de sinais analógicos em palavras binárias. Um clock fornece um sinal de tempo regular para um conversor analógico-digital (ADC), e este amostra o sinal analógico a cada pulso de clock. Uma unidade de amostragem e retenção mantém o valor amostrado até que o próximo pulso ocorra. Os ADCs usam métodos de conversão como aproximações sucessivas, rampa, dupla rampa e flash.

A **conversão digital-analógico** envolve a conversão de uma palavra binária em um sinal analógico. Os conversores digital-analógico (DACs) usam métodos como resistor ponderado e escada $R-2R$.

Um **multiplexador** é um circuito que é capaz de ter entrada de dados a partir de algumas fontes e, através da seleção do canal de entrada, fornece na saída o sinal de uma destas fontes.

O termo **aquisição de dados**, ou DAQ, é usado para o processo de obtenção de dados a partir de sensores e a transferência destes para um computador processá-los.

O termo **processamento de sinais digitais** ou **processamento de sinais discretos no tempo** é usado para o processamento realizado sobre um sinal por um microprocessador.

Problemas

- 4.1 Qual é a resolução de um ADC com uma palavra de 12 bits e um sinal analógico de entrada de amplitude 100 V?
- 4.2 Um sensor fornece uma saída analógica máxima de 5 V. Qual é o tamanho da palavra que um ADC necessita para ter uma resolução de 10 mV?
- 4.3 Uma escada $R-2R$ de resistores tem sua saída conectada a um amplificador operacional inversor com uma resistência de realimentação de $2R$. Se a tensão de referência for 5 V, determine a resolução do conversor.
- 4.4 Para um DAC com resistor ponderado, quais devem ser os valores das resistências de entrada a serem ponderadas para um DAC de 4 bits?

- 4.5 Qual é o tempo de conversão para um ADC de 12 bits com uma frequência de clock de 1 MHz?
- 4.6 Ao monitorar a entrada de alguns sensores de termopar, a sequência dos módulos apresentada a seguir é usada para cada termopar em sua interface com o microprocessador:
- Proteção, compensação da junção fria, amplificação, linearização, amostragem e retenção, conversão analógico-digital, buffer, multiplexação.
- Explique a função de cada um dos módulos.
- 4.7 Sugira os módulos que podem ser necessários na interface da saída de um microprocessador com um atuador.
- 4.8 Para o DAC com resistor ponderado de 4 bits mostrado na Figura 4.5, determine a saída dos resistores para o amplificador para as entradas 0001, 0010, 0100 e 1000 se as entradas são 0 V para o nível lógico 0 e 5 V para o nível lógico 1.
- 4.9 Se o menor resistor em um DAC de resistor ponderado de 16 bits for R , qual é o valor do maior resistor?
- 4.10 Um ADC do tipo rampa de 10 bits tem uma entrada de fundo de escala de 10 V. Quanto tempo ele gasta para converter uma entrada de fundo de escala se o período do clock for de 15 μ s.
- 4.11 Para um ADC de 12 bits com uma entrada de fundo de escala, entre as técnicas de aproximações sucessivas e rampa, qual é o ADC mais rápido?

5

Lógica Digital

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Reconhecer os símbolos das portas lógicas AND, OR, NOT, NAND, NOR e EX-OR e usar estas portas em aplicações.
- Explicar como os flip-flops SR, JK e D podem ser usados em sistemas de controle.
- Explicar a operação de decodificadores e do temporizador 555.

5.1 Lógica digital

Muitos sistemas de controle referem-se a eventos em que algo é acionado ou desligado quando determinadas condições ocorrerem. Por exemplo, no caso de uma máquina de lavar doméstica, o aquecedor é ligado apenas quando houver água no tambor e ela estiver no nível determinado. Este controle envolve sinais *digitais* em que há apenas dois níveis possíveis para o sinal. Os circuitos digitais são a base dos computadores digitais e sistemas de controle microprocessados.

Com um **controle digital** podemos, por exemplo, ligar a entrada de água em uma máquina de lavar se a tampa estiver fechada e se um determinado momento do ciclo de operação for atingido. Neste caso, existirão dois sinais de entrada que podem ser sinais “sim” ou “não”. O controlador é programado para fornecer uma saída “sim” apenas se as duas entradas forem “sim”, ou seja, se as entradas A e B forem 1, então a saída será 1. Diz-se que esta operação é controlada por uma **porta lógica**, que neste exemplo é uma porta AND. Existem diversas máquinas e processos controlados desta forma. O termo **lógica combinacional** é usado no caso da combinação de duas ou mais portas lógicas para formar a função desejada. Por exemplo, uma necessidade poderia ser que o som da buzina de um carro fosse ativado se a chave estivesse na ignição e uma porta estivesse aberta, ou se os faróis estivessem ligados e a porta do motorista estivesse aberta. A lógica combinacional depende apenas dos valores das entradas em um determinado instante.

Além do estudo da lógica combinacional, este capítulo também inclui um discussão de **lógica sequencial**. Este tipo de circuito é usado para executar um controle em uma sequência específica determinada por um clock de controle ou sinais de controle de habilitação/deshabilitação. Este é um circuito combinacional com memória. Portanto, a temporização ou histórico sequencial dos sinais de entrada representa uma parte na determinação da saída.

5.2 Portas lógicas

As portas lógicas são os blocos construtivos básicos para os circuitos eletrônicos digitais.

5.2.1 Porta AND

Suponha que temos uma porta em que a saída é nível alto apenas quando as entradas A e B forem nível alto; para todas as outras condições, a saída será nível baixo. Esta é uma porta AND (E). Podemos visualizar a porta AND como um circuito elétrico que envolve duas chaves em série (Figura 5.1(a)). Apenas quando as chaves A e B estiverem fechadas haverá a passagem de corrente. Têm sido usados conjuntos diferentes de símbolos de circuitos padrão para portas lógicas, sendo que o principal é originário dos Estados Unidos. Entretanto, um padrão internacional (IEEE/ANSI) foi desenvolvido; ele removeu o formato que distinguia as funções lógicas e usou um retângulo com a função lógica escrita nele. A Figura 5.1(b) mostra o símbolo padrão americano usado para uma porta AND e em (c) vemos o novo padrão internacional com o símbolo & indicando AND. Os dois símbolos são usados neste livro. Conforme mostra esta figura, podemos expressar a relação entre as entradas e a saída de uma porta AND na forma de uma equação, denominada **equação booleana** (veja Apêndice C). A equação booleana para uma porta AND é escrita como a seguir

$$A \cdot B = Q$$

Um exemplo de uma porta AND é um sistema de controle de intertravamento para uma máquina-ferramenta em que, se o dispositivo de segurança estiver no lugar e fornecer um sinal 1 e a alimentação for ligada, gerando um sinal 1, então a saída será ativada, um sinal 1, e a máquina entra em operação. Um outro exemplo é um alarme antifurto no qual ele gera uma saída, o som do alarme, quando o alarme for ligado e a porta for aberta ativando um sensor.

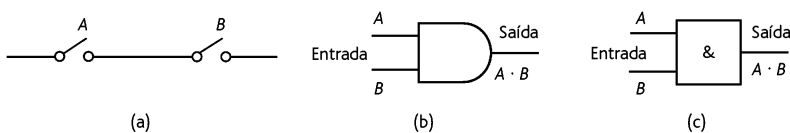


Figura 5.1 Porta AND: (a) representada por chaves, (b) símbolo padrão americano, (c) símbolo padrão internacional.

A relação entre as entradas e saídas de uma porta lógica podem ser tabuladas numa forma conhecida como **tabela-verdade**. Esta especifica as relações entre as entradas e saídas. Portanto, para uma porta AND com entradas A e B e uma única saída Q , teremos uma saída 1 se, e somente se, $A = 1$ e $B = 1$. Todas as outras combinações de A e B geram uma saída 0. Podemos escrever a tabela-verdade como

Entradas		Saída
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

Considere o que acontece quando temos duas entradas digitais que são funções do tempo, como mostra a Figura 5.2. O que está nesta figura é denominado de diagrama de temporização da porta AND. A saída da porta AND será nível alto apenas quando cada uma das entradas for nível alto. A saída é como mostra a mesma figura.

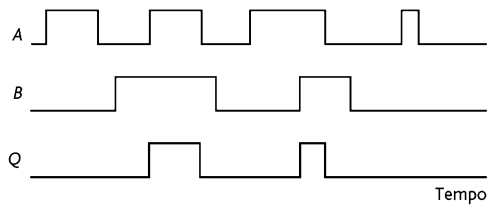


Figura 5.2 Porta AND.

5.2.2 Porta OR

Uma porta OR (OU) com entradas A e B fornece uma saída 1 quando A ou B for 1. Podemos visualizar esta porta como sendo um circuito elétrico envolvendo duas chaves em paralelo (Figura 5.3(a)). Quando a chave A ou a B for fechada, então haverá corrente. As portas OR podem também ter mais que duas entradas. A tabela-verdade para esta porta é a seguinte:

Entradas		Saída
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Podemos escrever a equação booleana para uma porta OR como

$$A + B = Q$$

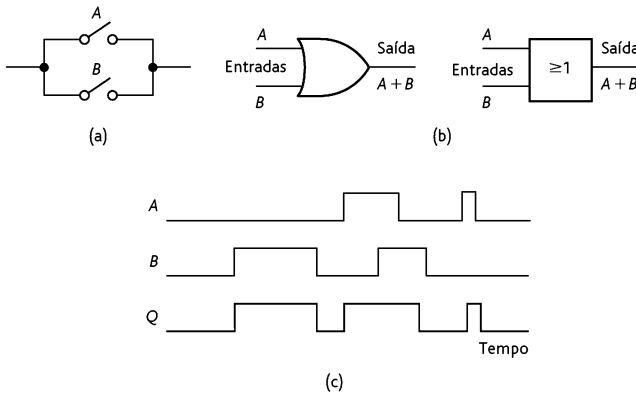


Figura 5.3 Porta OR: (a) representação por chaves, (b) símbolos, (c) diagrama de temporização.

Os símbolos usados para uma porta OR são mostrados na Figura 5.3(b); o uso do sinal “maior ou igual a 1” para representar a OR vem da função OR ser verdadeira se pelo menos uma entrada for verdadeira. A Figura 5.3(c) mostra um diagrama de temporização.

5.2.3 Porta NOT

A porta NOT (NÃO) tem apenas uma entrada e uma saída, em que a saída será 1 quando a entrada for 0, e 0 quando a entrada for 1. A porta NOT fornece uma saída que é a inversão da entrada, sendo denominado **inversor**. A Figura 5.4(a) mostra os símbolos usados para uma porta NOT. O 1 dentro do retângulo da porta NOT representa na verdade uma identidade lógica, ou seja, sem operação, e a inversão é representada pelo pequeno círculo na saída. Portanto, se há uma entrada digital que varia com o tempo, como na Figura 5.4(b), a saída varia com o tempo de forma inversa.

A seguir é mostrada a tabela-verdade da porta NOT:

Entrada A	Saída Q
0	1
1	0

A equação booleana que descreve a porta NOT é

$$\bar{A} = Q$$

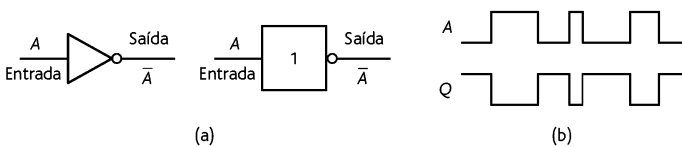


Figura 5.4 Porta NOT.

A barra sobre o símbolo é usada para indicar que o inverso, ou o complemento, está sendo considerado; portanto, a barra sobre o A indica que a saída Q é o inverso do valor de A .

5.2.4 Porta NAND

A porta NAND pode ser considerada como uma combinação de uma porta AND seguida por uma NOT (Figura 5.5(a)). Portanto, quando a entrada A for 1 e a B for 1, a saída será 0, e todas as outras combinações resultam em uma saída 1.

A porta NAND é simplesmente a tabela-verdade da AND com a saída invertida. Uma forma alternativa de considerar esta porta é como uma porta OR com uma porta NOT, uma em cada entrada, antes da porta OR. A Figura 5.5(b) mostra os símbolos usados para a porta NAND, que é o símbolo da AND seguido por um pequeno círculo para indicar a inversão. A seguir é apresentada a tabela-verdade:

Entradas		Saída
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

A equação booleana que descreve a porta NAND é

$$\overline{A \cdot B} = Q$$

A Figura 5.5(c) mostra a saída de uma porta NAND quando suas duas entradas são sinais digitais que variam com o tempo. A saída é nível baixo apenas quando as entradas são nível alto.

5.2.5 Porta NOR

A porta NOR pode ser considerada como uma combinação de uma porta OR seguida por uma NOT (Figura 5.6(a)). Portanto, quando a entrada A ou B for 1, a saída será 0. Isto é simplesmente uma porta OR com uma saída invertida. Uma forma alternativa de considerar esta porta é como uma porta AND com uma porta NOT, em cada uma das entradas, antes da porta AND. A Figura 5.6(b) mostra os símbolos

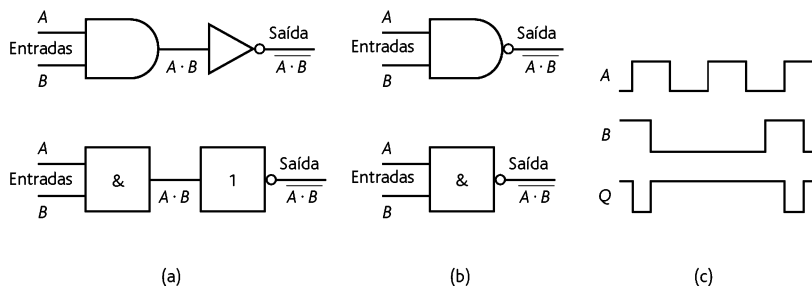


Figura 5.5 Porta NAND.

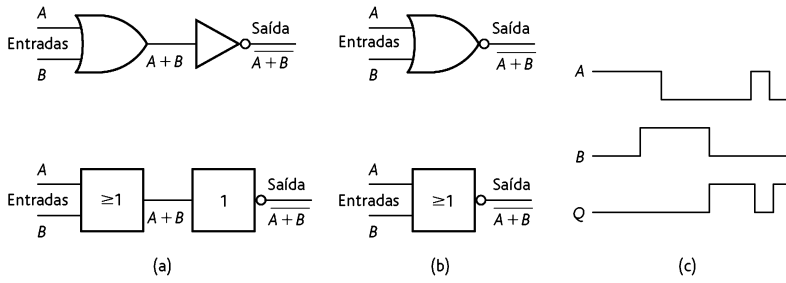


Figura 5.6 Porta NOR.

usados para a porta NOR; é o símbolo da porta OR seguido por um pequeno círculo para indicar inversão. A equação booleana para a porta NOR é

$$\overline{A + B} = Q$$

A seguir é apresentada a tabela-verdade para a porta NOR, e a Figura 6.5(c) mostra o seu diagrama de temporização:

Entradas		Saída
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

5.2.6 Porta EX-OR

A porta OR EXCLUSIVO (EX-OR) pode ser implementada com uma porta OR, duas portas AND e dois inversores, como mostra a Figura 5.7(a). Alternativamente, ela pode ser implementada substituindo-se, na Figura 5.7(a), a porta OR por NAND

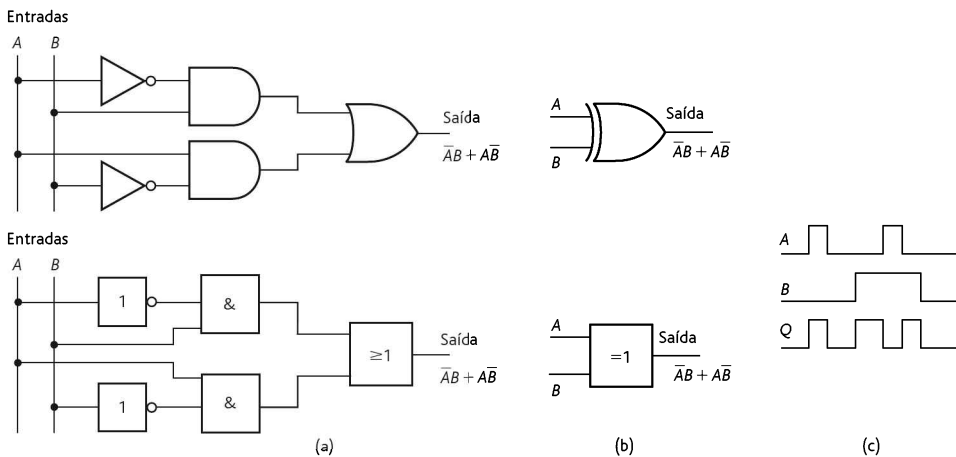


Figura 5.7 Porta EX-OR.

e as portas AND por OR. O símbolo da EX-OR é mostrado na Figura 5.7(b); o símbolo $\neq 1$ representa que a saída é verdadeira se apenas uma entrada também for verdadeira. A seguir é mostrada a tabela-verdade, e a Figura 5.7(c) mostra um diagrama de temporização:

Entradas		Saída
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

5.2.7 Combinação de portas

Pode parecer que para construir sistemas lógicos necessitamos de um grupo de portas. Entretanto, conforme mostrado a seguir, podemos implementar todas as funções lógicas das portas a partir de apenas uma delas. Considere a combinação de três portas NOR como mostra a Figura 5.8.

A tabela-verdade, com as saídas intermediárias e final para esta combinação, é mostrada a seguir:

A	B	C	D	Q
0	0	1	1	0
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

O resultado final é o mesmo que o de uma porta AND. Se acrescentarmos a essa associação de portas uma porta NOT na saída, obteremos a tabela-verdade da porta NAND.

Uma combinação de três portas NAND é mostrada na Figura 5.9. A tabela-verdade com as saídas intermediária e final é mostrada a seguir:

A	B	C	D	Q
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

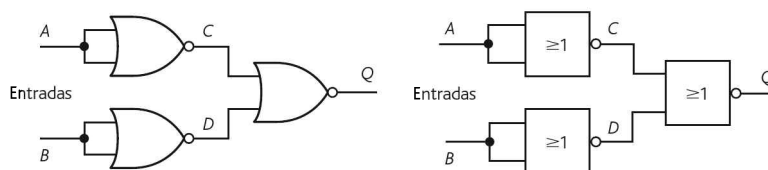


Figura 5.8 Três portas NOR.

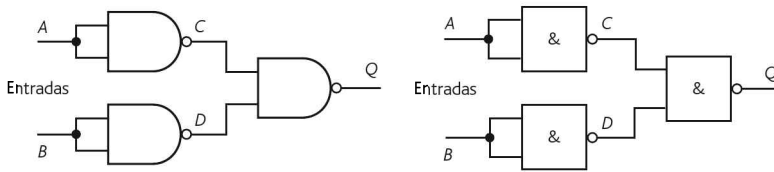


Figura 5.9 Três portas NAND.

O resultado é o mesmo de uma porta OR. Se na saída desta associação de portas acrescentarmos uma porta NOT, obteremos a mesma tabela-verdade de uma porta NOR.

Estas duas ilustrações de combinações de portas mostra como um tipo de porta, NOR ou NAND, pode ser usado para substituir outras portas, desde que usemos mais de uma. As portas também podem ser combinadas para formar circuitos combinacionais mais complexos e circuitos sequenciais.

As portas lógicas estão disponíveis em circuitos integrados. Os fabricantes têm padronizado a numeração dos esquemas de forma que os números de identificação (*part numbers*) básicos sejam os mesmos independentemente do fabricante. Por exemplo, a Figura 5.10(a) mostra um conjunto de portas disponíveis no circuito integrado 7408; ele tem internamente quatro portas AND e é alimentado no pino 14. As conexões de alimentação que são feitas nos pinos 7 e 14 fornecem a tensão de operação para todas as quatro portas AND. Para indicar em que extremidade está localizado o pino 1, existe um pequeno corte entre os pinos 1 e 14. O circuito integrado 7411 tem três portas AND, tendo cada uma três entradas; o circuito integrado 7421 tem duas portas AND, tendo cada uma quatro entradas. A Figura 5.10(b) mostra o conjunto de portas disponíveis no circuito integrado 7402. Este tem quatro portas NOR de duas entradas em um encapsulamento de 14 pinos e conexões de alimentação nos pinos 7 e 14. O circuito integrado 7427 tem três portas, cada uma com três entradas; o circuito integrado 7425 tem duas portas, cada uma com quatro entradas.

Para um estudo de como a álgebra booleana e técnicas como o teorema de De Morgan e mapas de Karnaugh podem ser usadas para gerar as funções lógicas desejadas a partir de portas lógicas, veja o Apêndice C.

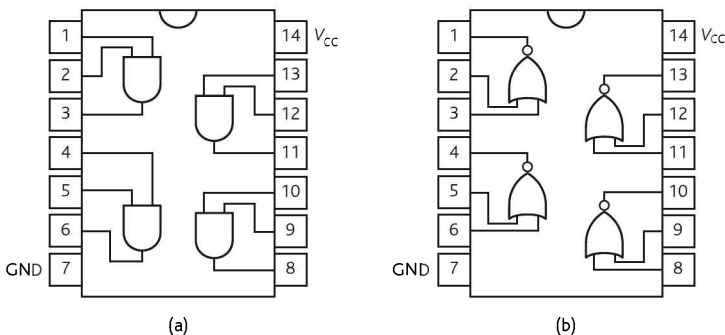


Figura 5.10 Circuitos integrados: (a) 7408, (b) 7402.

5.3 Aplicações de portas lógicas

Os exemplos a seguir fazem uso de portas lógicas em algumas aplicações simples.

5.3.1 Geradores de paridade

No capítulo anterior estudamos o uso de bits de paridade e um método de detecção de erro. Um único bit é acrescentado a cada bloco de código para forçar o número de uns no bloco, incluindo o bit de paridade, a ser uma quantidade ímpar, caso este paridade esteja sendo usada, ou uma quantidade par, no caso da paridade usada ser par.

A Figura 5.11 mostra um circuito com portas lógicas que poderia ser usado para determinar e acrescentar o bit de paridade apropriado. Este sistema emprega portas EX-OR; se todas as entradas de uma porta EX-OR forem 0 ou todas forem 1, a saída será 0, e se as entradas estiverem em níveis diferentes, a saída será 1. Os pares de bits são verificados, e resulta uma saída 1 se eles estiverem em níveis diferentes. Se for desejada uma paridade ímpar neste caso, o bit de definição da paridade é 0; se for a paridade par, ele será 1. O bit de definição da paridade apropriada pode então ser acrescentado ao sinal para transmissão. O mesmo circuito pode ser usado para verificar a paridade no receptor, que indica nível 1 quando há um erro. Estes circuitos estão disponíveis na forma de circuito integrado.

5.3.2 Comparador digital

Um comparador digital é usado para comparar duas palavras digitais para determinar se elas são exatamente iguais. As duas palavras são comparadas bit a bit e uma saída 1 é gerada se elas forem iguais. Para comparar a igualdade entre dois bits, pode ser usada uma porta EX-OR; se os bits forem ambos 0 ou ambos 1, a saída é 0, e se

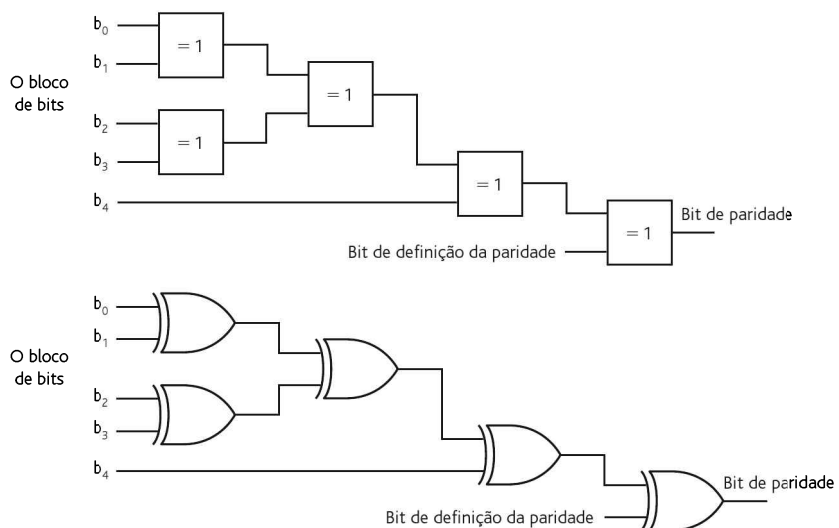


Figura 5.11 Gerador de bit de paridade.

os bits forem diferentes, a saída é 1. Para obter uma saída 1 quando os bits forem iguais, precisamos acrescentar uma porta NOT. A combinação de uma EX-OR e uma NOT é denominada de porta EX-NOR. Para comparar cada um dos pares de bits em duas palavras, precisamos de uma porta EX-NOR para cada par. Se os pares de bits forem iguais, então a saída da porta EX-NOR será 1. Podemos então usar uma porta AND para gerar uma saída 1 quando todas as saídas das portas EX-NOR forem 1. A Figura 5.12 mostra este sistema.

Os comparadores digitais estão disponíveis como circuitos integrados e podem determinar geralmente não apenas se duas palavras são iguais, mas qual é a maior. Por exemplo, o 7485 é um comparador de magnitude de 4 bits que compara duas palavras de 4 bits, A e B , gerando uma saída 1 a partir do pino 5 se A for maior do que B , uma saída 1 a partir do pino 6 se A for igual a B , e uma saída 1 a partir do pino 7 se A for menor do que B .

5.3.3 Codificador

A Figura 5.13 mostra um sistema simples através do qual um controlador pode enviar um sinal digital codificado para um semáforo de forma que o código determina qual cor, entre vermelho, amarelo ou verde, será ligada. Para ativar a luz vermelha podemos usar o sinal transmitido $A = 0$, $B = 0$; para a luz amarela, $A = 0$ e $B = 1$; e para a luz verde, $A = 1$ e $B = 0$. Podemos ligar as luzes usando estes códigos usando três portas AND e duas NOT.

5.3.4 Conversor de código

Em muitas aplicações há a necessidade de modificar os dados de um tipo de código para outro. Por exemplo, a saída de um sistema microprocessado pode ser BCD e podemos precisarmos transformá-lo em um código adequado para acionar um display de sete segmentos. O termo **decodificação de dados** é usado para o processo de conversão de algum grupo de bits codificado, por exemplo BCD, binário, ou hexa, em uma saída ativa individual que representa um grupo. Um decodificador tem n linhas de entrada binária para a entrada codificada de uma palavra de n bits e fornece m linhas de saída, das quais apenas uma é ativada para uma combinação possível de entrada, ou seja, apenas uma linha de saída fornece a saída para uma

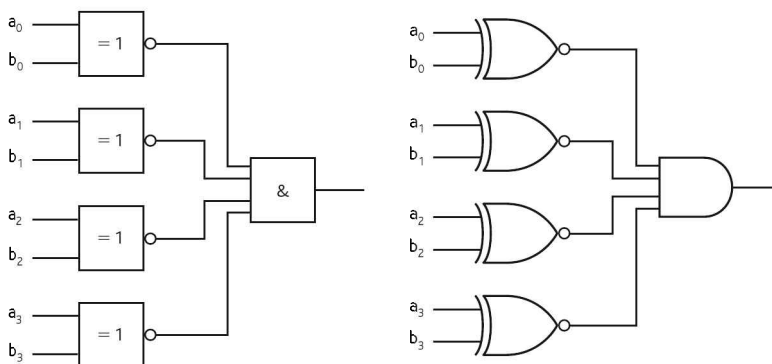


Figura 5.12 Comparador.

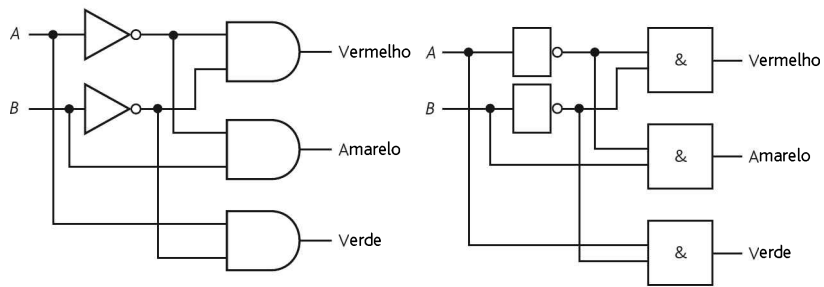


Figura 5.13 Semáforo.

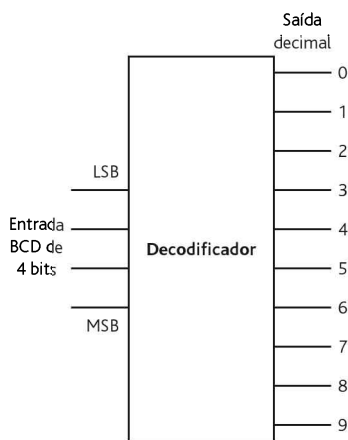


Figura 5.14 Decodificador.

determinada palavra de código de entrada. Por exemplo, um decodificador de BCD para decimal tem um código de entrada de 4 bits e 10 linhas de saída, de forma que uma determinada entrada BCD faz com que apenas uma das linhas de saída seja ativada, e desta forma indica um determinado número decimal, com cada linha de saída tendo em correspondência um número decimal (Figura 5.14).

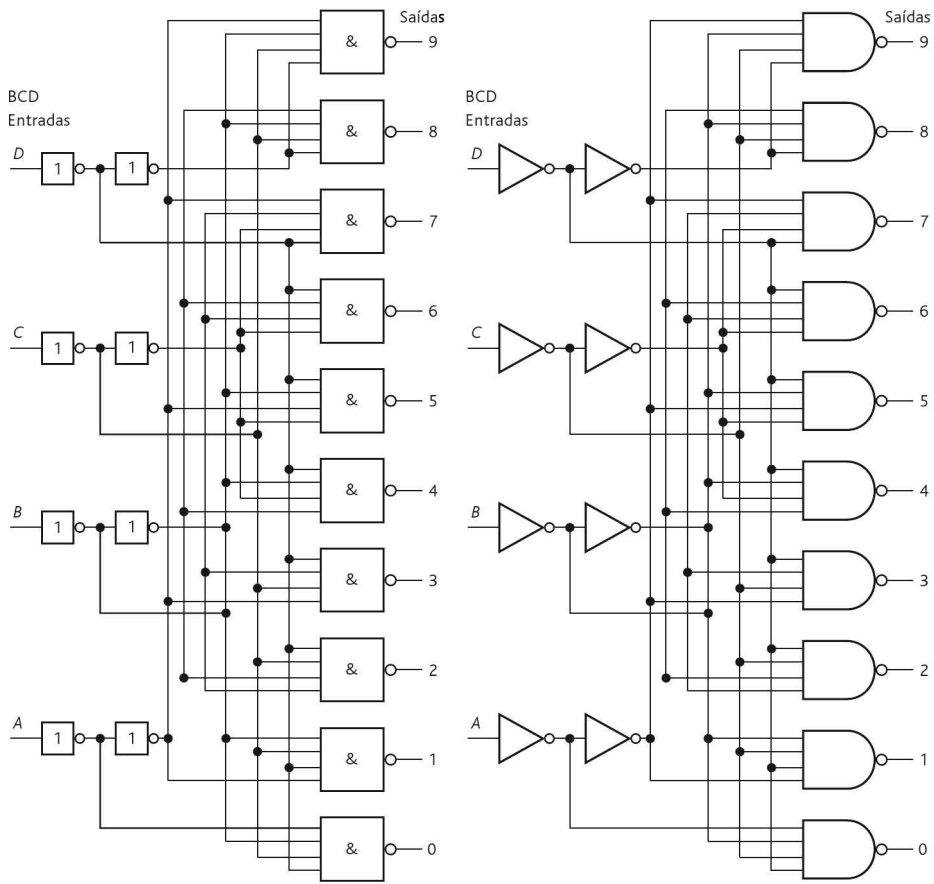
Portanto, em geral, um **decodificador** é um circuito lógico que, para uma entrada qualquer, determina qual é o número de entrada e ativa a saída que corresponde a este número. Os decodificadores são bastante usados em circuitos microprocessados.

Os decodificadores podem ter a saída ativa em nível alto e inativa em nível baixo, ou o contrário. Para o caso de uma saída ativa em nível alto, o decodificador pode ser implementado com portas AND, enquanto que para uma saída ativa em nível baixo pode-se usar portas NAND. A Figura 5.15 mostra como um decodificador de BCD para decimal, para uma saída ativa em nível baixo, pode ser implementado, e a tabela-verdade resultante. Este decodificador é facilmente encontrado na forma de circuito integrado, como por exemplo o 74LS145.

Um tipo de decodificador que é muito usado é o de BCD para sete segmentos, como por exemplo o 74LS244, que tem uma entrada BCD de 4 bits e gera uma saída que comanda os sete segmentos de um display.

O termo **decodificador de 3 para 8 linhas** é usado no caso de um decodificador que tem três linhas de entrada e oito de saída. Ele recebe um número binário de 3 bits e ativa uma das oito saídas que corresponde àquele número. A Figura 5.16 mostra como este decodificador pode ser implementado a partir de portas lógicas e a sua tabela-verdade.

Alguns decodificadores têm uma ou mais entradas de HABILITAÇÃO (ENABLE) que são usadas para controlar sua operação. Portanto, com uma linha de HABILITAÇÃO em nível ALTO, o decodificador funciona normalmente e as entradas determinam qual saída será nível ALTO; com a linha de HABILITAÇÃO em nível BAIXO, todas as saídas são mantidas em nível BAIXO independentemente das entradas. A Figura 5.17 mostra o CI 74LS138, que é um decodificador de 3 para 8 linhas de uso comum. Observe que as saídas são ativas em nível BAIXO, em vez



Entradas				Saídas									
A	B	C	D	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figura 5.15 Decodificador de BCD para decimal: 1 = nível ALTO, 0 = nível BAIXO.

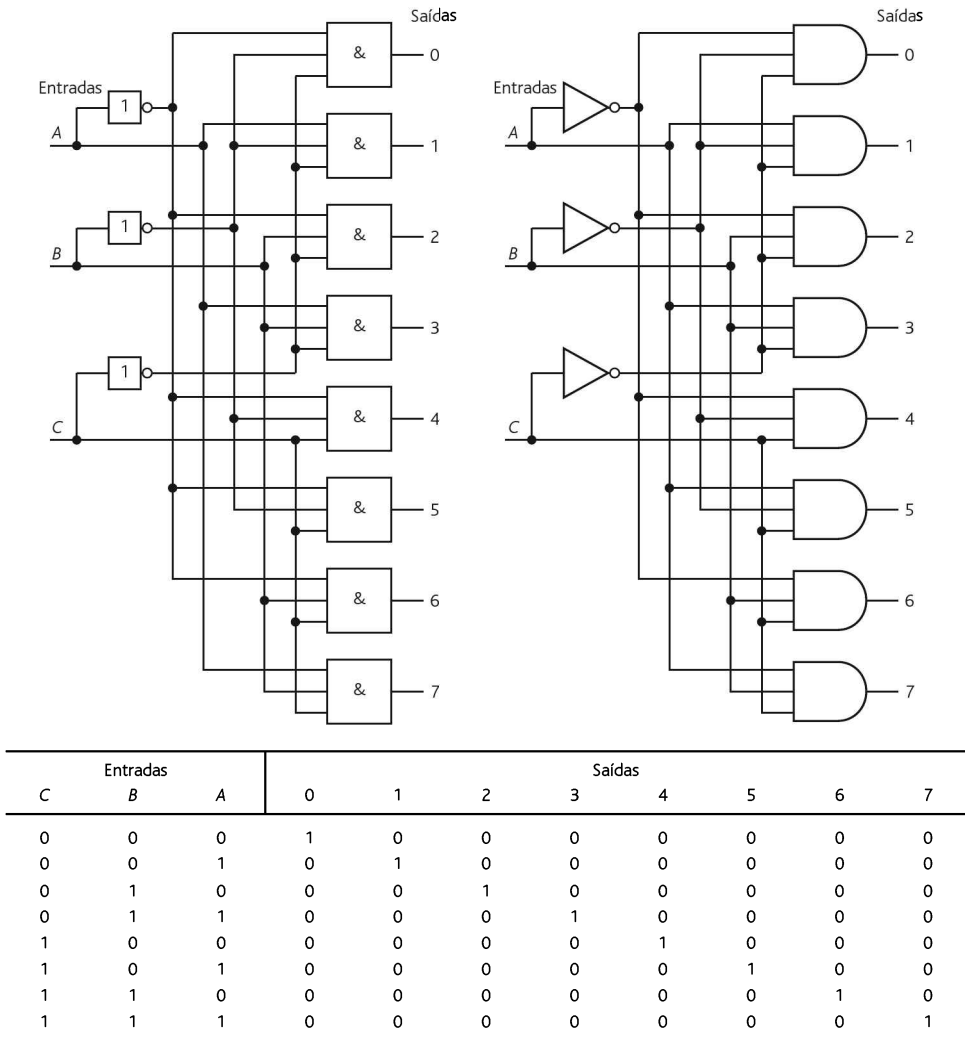
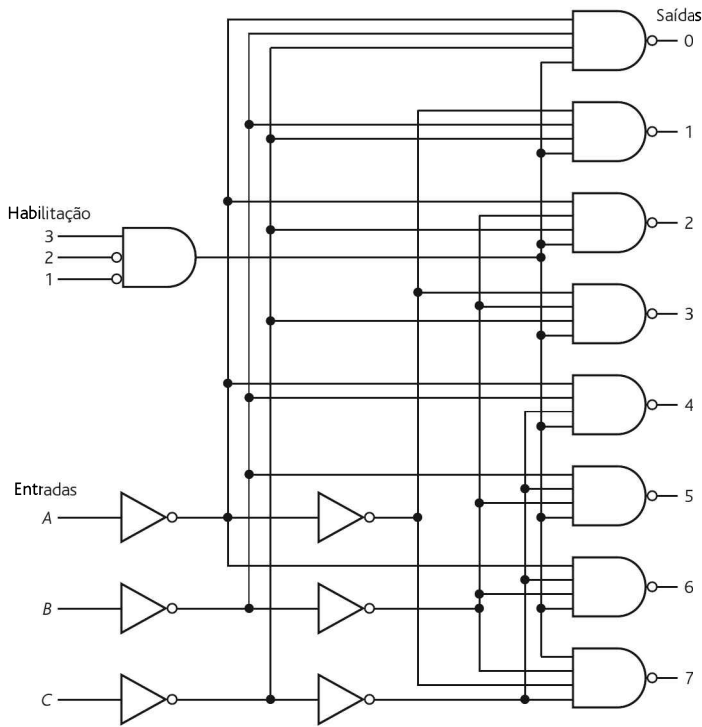


Figura 5.16 Decodificador de 3 para 8 linhas.

de nível ALTO como na Figura 5.16, e que o decodificador tem três linhas de HABILITAÇÃO em que, para um funcionamento normal, E1 e E2 devem ser nível BAIXO e E3 ALTO. Todas as outras variações resultam em um decodificador desabilitado com as saídas em nível ALTO.

A Figura 5.18 ilustra o tipo de resposta que podemos ter a partir de um decodificador 74LS138 para diferentes entradas.

Um decodificador 74LS138 pode ser usado com um microprocessador com a entrada HABILITAÇÃO usada para ligar o decodificador e então, dependendo da saída das três linhas do microprocessador, uma das oito saídas do decodificador recebe uma saída de nível BAIXO enquanto todas as outras permanecem em nível ALTO. Portanto, podemos considerar cada dispositivo de saída como tendo um endereço, ou seja, um único número de saída binário, de forma que quando o micro-



Habilitação			Entradas			Saídas							
E1	E2	E3	C	B	A	0	1	2	3	4	5	6	7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Figura 5.17 O 74LS138: 1 = nível ALTO, 0 = nível BAIXO, X = não importa.

processador envia um endereço para decodificá-lo, ativa o dispositivo alocado neste endereço. O 74LS138 pode então ser chamado de decodificador de endereço.

5.4 Lógica sequencial

Os circuitos lógicos considerados nas seções anteriores deste capítulo são exemplos de sistemas lógicos combinacionais. Nestes sistemas, a saída é determinada pela combi-

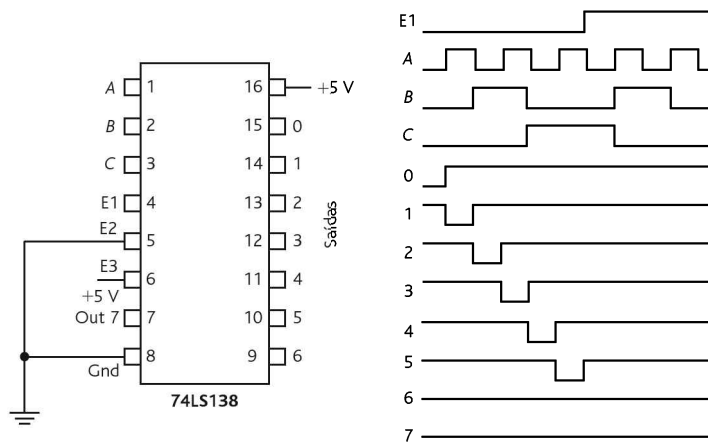


Figura 5.18 O CI 74LS138.

nação das variáveis de entrada em um determinado instante do tempo. Por exemplo, se a entrada *A* e a *B* estiverem ativas no mesmo instante, então uma porta AND retorna uma saída ativa. A saída não depende dos estados anteriores das entradas. No caso de um sistema necessitar de uma saída que dependa do valor anterior das entradas, é necessário um sistema de **lógica sequencial**. A principal diferença entre um sistema de lógica combinacional e um sequencial é que este tem alguma forma de memória.

A Figura 5.19 mostra a forma básica de um sistema de lógica sequencial. A parte combinacional do sistema recebe sinais lógicos de entradas externas e das saídas da memória. O sistema combinacional opera então sobre estas entradas, produzindo suas saídas. Estas saídas são então uma função das entradas externas e da informação armazenada em sua memória.

5.4.1 Flip-flop

O **flip-flop (FF)** é um elemento básico de memória que é constituído de portas lógicas e é um dispositivo de lógica sequencial. Existem alguns tipos de flip-flops. A Figura 5.20(a) mostra um tipo, o SR (*set-reset**), implementado com portas NOR.

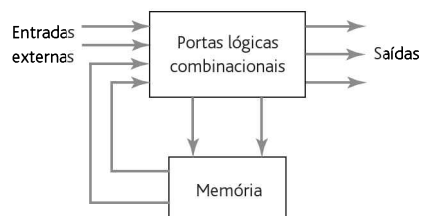


Figura 5.19 Sistema lógico sequencial.

* N. de T.: Estes termos são usados normalmente em inglês. SET significa colocar a saída do FF em 1, e RESET significa colocá-la em 0. Os termos “setar” e “resetar” já têm uso consolidado como jargão técnico.

Se inicialmente tivermos as duas saídas em 0 e $S = 0$ e $R = 0$, então quando ativamos a entrada SET, com S variando de 0 para 1, a saída da porta NOR 2 se torna 0. Com isto, as duas entradas da porta NOR 1 se tornam 0, e então sua saída se torna 1. Esta realimentação funciona como uma entrada para a porta NOR 2, que passa a ter suas duas entradas em 1, não resultando em alteração posterior.

Agora se S mudar de 1 para 0, a saída da porta NOR 1 permanecerá em 1 e a saída da porta NOR 2 permanecerá em 0. Não há alterações nas saídas quando a entrada S muda de 1 para 0. O FF permanece neste estado indefinidamente se a única mudança for em S . Ele “lembra” deste estado em que foi setado. A Figura 5.20(b) ilustra isto com um diagrama de temporização no qual um pulso retangular é usado como entrada S .

Se mudarmos R de 0 para 1 enquanto S for 0, a saída da porta NOR 1 muda para 0 e, portanto, a saída da porta NOR 2 muda para 1. Diz-se que o FF foi resetado. Uma mudança de R para 0 não terá efeito sobre estas saídas.

Portanto, quando S for 1 e R for 0, a saída Q mudará para 1 se ela era anteriormente 0, ou permanecerá em 1 se ela era anteriormente 1. Este é o estado SET, e a saída permanecerá neste estado mesmo quando S mudar para 0. Quando S for 0 e R for 1, a saída Q é resetada para 0 se ela era anteriormente 1, ou permanecerá em 0 se ela era anteriormente 0. Este é o estado de RESET. A saída Q em um determinado instante depende das entradas S e R e também do valor anterior da saída. Os estados mostrados na tabela a seguir ilustram isto:

S	R	$Q_t \rightarrow Q_{t+1}$	$\bar{Q}_t \rightarrow \bar{Q}_{t+1}$
0	0	$0 \rightarrow 0$	$1 \rightarrow 1$
0	0	$1 \rightarrow 1$	$0 \rightarrow 0$
0	1	$0 \rightarrow 0$	$1 \rightarrow 1$
0	1	$1 \rightarrow 0$	$0 \rightarrow 0$
1	0	$0 \rightarrow 1$	$1 \rightarrow 0$
1	0	$1 \rightarrow 1$	$0 \rightarrow 0$
1	1	Não permitido	
1	1	Não permitido	

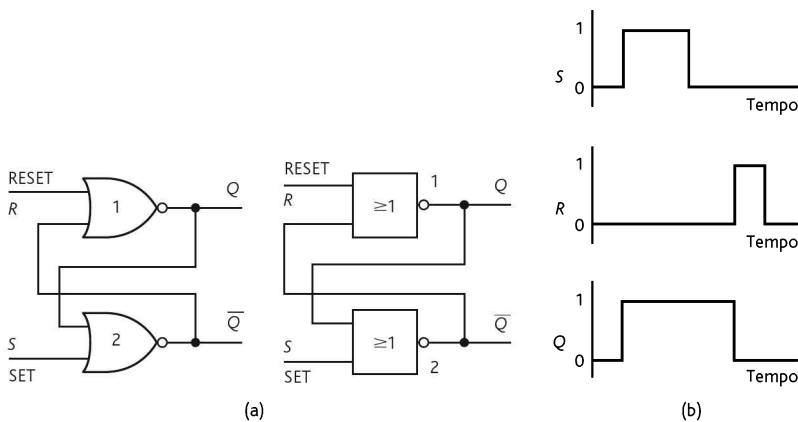


Figura 5.20 Flip-flop SR.

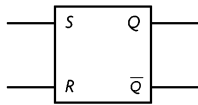


Figura 5.21 Flip-flop SR.

Note que se S e R são simultaneamente iguais a 1, pode ocorrer um estado não estável e, desta forma, esta condição de entrada não é permitida. A Figura 5.21 mostra o símbolo em bloco simplificado usado para representar um flip-flop SR.

Como uma simples ilustração do uso de um flip-flop, considere um sistema de alarme simples em que o alarme é acionado quando um feixe de luz é interrompido e permanece ativo mesmo quando o feixe de luz não está mais interrompido. A Figura 5.22 mostra um possível circuito para este sistema. Um fototransistor pode ser usado como sensor e conectado de modo que, quando iluminado, produza um 0 V virtual na entrada S , mas quando a iluminação é interrompida, ele produza uma tensão de aproximadamente 5 V na entrada S . Quando o feixe de luz é interrompido, S se torna 1 e a saída do flip-flop se torna 1, soando o alarme. A saída permanecerá em 1 mesmo quando S mudar para 0. O alarme poderá ser desativado apenas se a chave de RESET for momentaneamente aberta, gerando 5 V na entrada R .

5.4.2 Sistemas síncronos

Frequentemente é necessário realizar operações de SET e RESET em determinados momentos. Em um **sistema assíncrono**, ou sem clock, as saídas das portas lógicas podem mudar de estado em qualquer momento quando se muda uma ou mais entradas. Em um **sistema síncrono**, ou com clock, os instantes exatos nos quais qualquer saída pode mudar de estado são determinados por um sinal denominado de clock. Este sinal é geralmente um trem de pulsos retangulares, e quando este mesmo sinal é usado para todas as partes do sistema, as saídas são sincronizadas. A Figura 5.23(a) mostra o princípio de funcionamento de um **flip-flop SR implementado com portas**. Os sinais de SET e clock passam por uma porta AND antes de chegarem na entrada S do flip-flop. Portanto, o sinal SET chega no flip-flop apenas quando este sinal e o de clock são 1. Da mesma forma, os sinais de RESET e clock passam por uma porta AND antes de chegarem na entrada R do flip-flop. Como consequência, SET e RESET apenas têm efeito no flip-flop no momento determinado pelo clock. A Figura 5.23(b) mostra o diagrama de temporização.

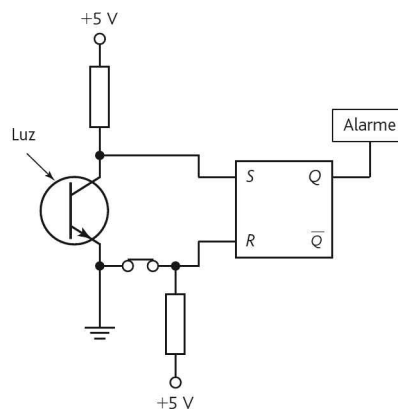


Figura 5.22 Circuito de um alarme.

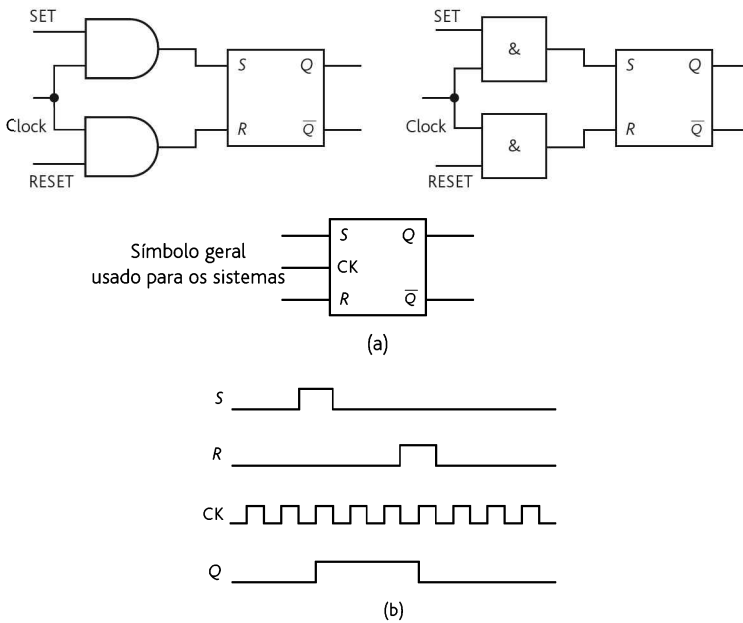


Figura 5.23 Flip-flop SR com clock.

5.4.3 Flip-flop JK

Em muitas aplicações, o estado indeterminado que ocorre com o flip-flop SR quando $S = 1$ e $R = 1$ não é aceitável, e um outro tipo de flip-flop é usado, o **flip-flop JK** (Figura 5.24). Este se tornou um dispositivo muito usado.

A seguir é mostrada a tabela-verdade deste flip-flop; observe que a única diferença para os estados do flip-flop SR ocorre quando as entradas são 1:

J	K	$Q_t \rightarrow Q_{t+1}$	$\bar{Q}_t \rightarrow \bar{Q}_{t+1}$
0	0	$0 \rightarrow 0$	$1 \rightarrow 1$
0	0	$1 \rightarrow 1$	$0 \rightarrow 0$
0	1	$0 \rightarrow 0$	$1 \rightarrow 1$
0	1	$1 \rightarrow 0$	$0 \rightarrow 1$
1	0	$0 \rightarrow 1$	$1 \rightarrow 0$
1	0	$1 \rightarrow 1$	$0 \rightarrow 0$
1	1	$0 \rightarrow 1$	$1 \rightarrow 0$
1	1	$1 \rightarrow 0$	$0 \rightarrow 1$

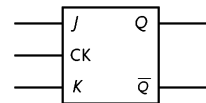


Figura 5.24 Flip-flop JK.

Como uma ilustração do uso deste flip-flop, considere a necessidade de uma saída em nível alto quando a entrada A for para nível alto ao mesmo tempo em que a entrada B . Uma porta AND pode ser usada para determinar se as duas entradas são nível alto, mas a saída desta porta será nível alto independentemente de qual entrada foi para nível alto primeiro. Entretanto, se as entradas A e B são de um flip-flop JK, então A tem que ser nível alto primeiro para que a saída seja nível alto quando B for nível alto posteriormente.

5.4.4 Flip-flop D

O **flip-flop D**, ou de dado, é basicamente um flip-flop SR ou um JK com a entrada D sendo conectada diretamente nas entradas S ou J e via uma porta NOT nas entradas R ou K (Figura 5.25(a)); no símbolo do flip-flop D, as junções nas entradas R e K são denominadas D . Nesta configuração, uma entrada 0 ou 1 faz com que as saídas sigam a entrada quando o pulso de clock for 1 (Figura 5.25(b)). Um uso particular do flip-flop D é garantir que a saída receberá o valor da entrada D em momentos precisamente definidos. A Figura 5.25(c) mostra o símbolo usado para um flip-flop D.

Com o flip-flop D anterior, quando o clock ou entrada de habilitação for para nível alto, a saída segue o dado presente na entrada D . Diz-se assim que esse flip-flop é transparente. Quando há uma transição de alto para baixo na entrada de habilitação, a saída Q é mantida no nível do dado que tinha antes da transição. Diz-se que o dado na transição é **memorizado** (operação *latch*). Os flip-flops D são disponíveis como circuitos integrados. O 7475 é um exemplo deste; ele contém quatro *latches* D.

O flip-flop D 7474 difere do 7475 por ser um dispositivo disparado por borda; em um encapsulamento do 7474 existem dois flip-flops deste tipo. Em um flip-flop D disparado por borda, a transição em Q ocorre sempre na borda do pulso de clock de entrada, e no caso do 7474 essa borda é positiva, ou seja, na transição de baixo para alto. A Figura 5.26(a) ilustra isto. O símbolo básico para um flip-flop D disparado por borda difere de um flip-flop D por um pequeno triângulo colocado na entrada CK (Figura 5.26(b)). Existem também duas outras entradas denominadas de preset e clear. Um nível baixo na entrada preset leva a saída Q para 1, enquanto que um nível baixo na entrada clear leva a saída Q para 0.

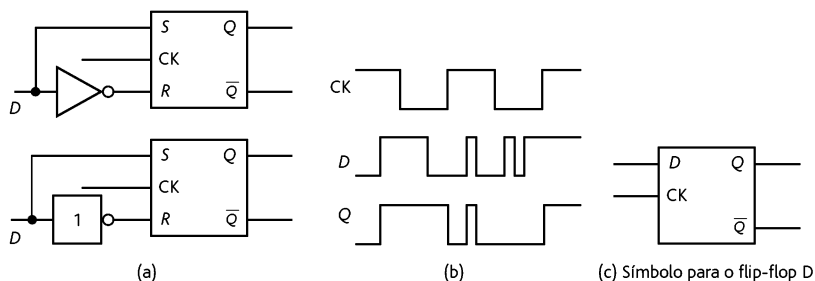


Figura 5.25 Flip-flop D.

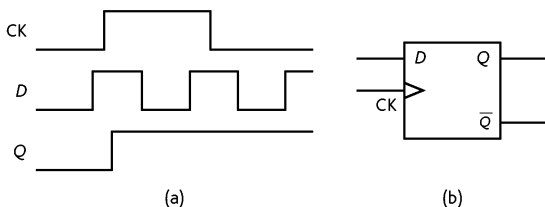


Figura 5.26 (a) Borda de disparo positiva, (b) símbolo para o flip-flop D disparado por borda.

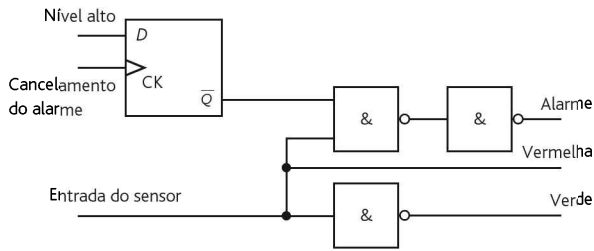


Figura 5.27 Sistema de alarme.

Como uma ilustração simples de uma aplicação para este flip-flop, a Figura 5.27 mostra um sistema que poderia ser usado para mostrar uma luz verde quando a entrada do sensor fosse nível baixo e uma luz vermelha quando ela fosse nível alto e acionaria um alarme. É para a luz vermelha permanecer ligada enquanto a entrada do sensor for nível alto, porém o alarme deve ser desligado. Este pode ser um sistema de monitoração da temperatura de algum processo, o sensor e o condicionador de sinal gerando um sinal de nível baixo quando a temperatura estiver abaixo do nível de segurança e um nível alto quando a temperatura estiver acima. O flip-flop tem uma entrada em nível alto. Quando uma entrada em nível baixo for aplicada na entrada CK e a entrada do sensor for nível baixo, a luz verde é ligada. Quando a entrada do sensor muda para nível alto, a luz verde apaga, a luz vermelha liga e o alarme é acionado. O alarme pode ser cancelado aplicando-se um nível alto na entrada CK, mas a luz vermelha permanece ligada enquanto a entrada do sensor for nível alto. Um sistema como este pode ser construído usando-se um 7474 e um circuito integrado contendo três portas NAND.

5.4.5 Registradores

Um **registrador** é um conjunto de elementos de memória e é usado para manter informação até que seja necessário. Ele pode ser implementado por um conjunto de flip-flops. Cada um destes armazena um sinal binário, ou seja, um 0 ou um 1. A Figura 5.28 mostra a forma que um registrador de 4 bits pode ter quando usa flip-flops D.

Quando o sinal de carga for 0, nenhum pulso de clock chega aos flip-flops, e desta forma nenhuma mudança ocorre nos estados dos flip-flops. Quando o sinal de carga for 1, as entradas podem mudar os estados dos flip-flops. Enquanto o sinal de carga for 0, os flip-flops manterão os estados anteriores.

5.4.6 Temporizador 555

O chip temporizador 555 é muito usado em circuitos digitais, visto que ele pode oferecer uma ampla variedade de soluções de temporização. Ele consiste de um flip-flop SR que recebe sinais de entrada a partir de dois comparadores (Figura 5.29). Cada um dos comparadores tem uma tensão de entrada derivada de uma rede de resistores de mesmo valor. Assim, o comparador A tem uma tensão na entrada não inversora de $V_{CC}/3$, e o comparador B tem uma tensão na entrada inversora de $2V_{CC}/3$.

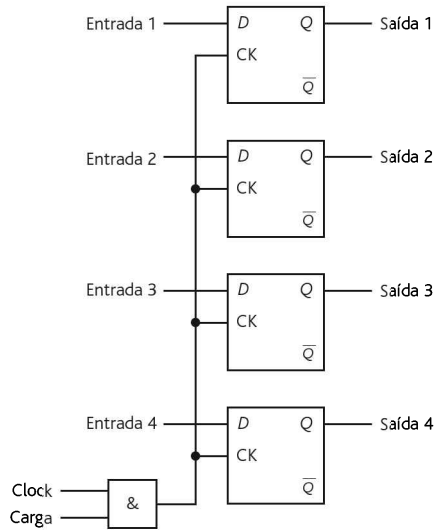


Figura 5.28 Registrador.

Uma das aplicações do temporizador 555 é como **multivibrador monoestável**, que é um circuito que gera um único pulso com o tempo de duração desejado após receber um pulso de disparo. A Figura 5.30(a) mostra como o temporizador é conectado para ser usado desta forma. Inicialmente, a saída é nível baixo, com o transistor colocando o capacitor em curto-circuito e a saída dos dois comparadores em nível baixo (Figura 5.30(b)). Quando o pulso de disparo cai abaixo de $V_{CC}/3$, o comparador de disparo vai para nível alto e seta o flip-flop. A saída vai então para

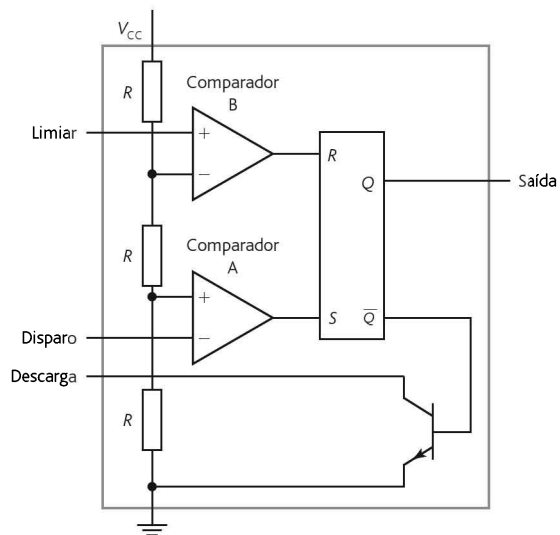


Figura 5.29 Temporizador 555.

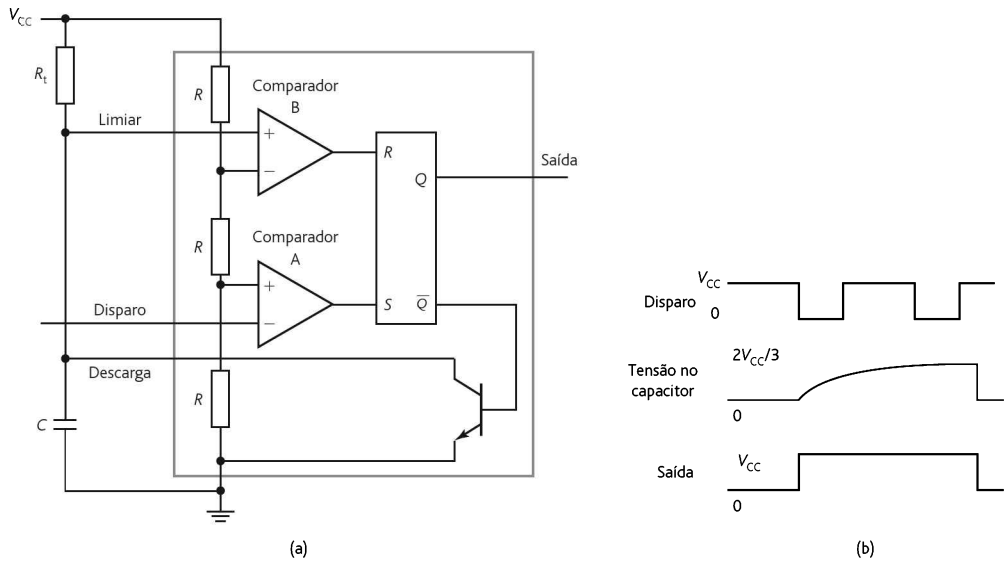


Figura 5.30 Multivibrador monoestável.

nível alto e o transistor entra em corte, permitindo a carga do capacitor. Quando o capacitor atinge $2V_{CC}/3$, o comparador de limiar reseta o flip-flop, que reseta a saída para nível baixo e descarrega o capacitor. Se a entrada de disparo for ativada enquanto a saída estiver em nível alto, esta operação não terá efeito. A largura do pulso corresponde ao tempo que o capacitor leva para se carregar até $2V_{CC}/3$, e isto depende da constante de tempo, ou seja, do valor de R_1C , e é dado pela relação normal para a carga de um capacitor através de uma resistência como $1,1 R_1C$. Como ilustração, considere a situação em que um alarme contra roubo é acionado se uma porta for aberta e o dono da casa não inserir o número solicitado em um teclado dentro de 30 s. Se o circuito da Figura 5.30 for usado com um capacitor de $1 \mu\text{F}$, então R_1 tem que ter um valor de $30/(1,1 \times 1 \times 10^{-6}) = 27,3 \text{ M}\Omega$.

Resumo

Nos **sistemas de lógica combinacional**, a saída é determinada pela combinação das variáveis de entrada em um determinado instante. A saída não depende do valor anterior das entradas. Um **circuito sequencial** é usado onde o sistema necessita de uma saída que depende dos valores anteriores das entradas. A principal diferença entre um sistema de lógica combinacional e um de lógica sequencial é que este último tem algum tipo de memória.

Um **decodificador** é um circuito lógico que, a partir das entradas, verifica qual o número que é equivalente e ativa uma saída que corresponde ao número identificado.

O **flip-flop** é um elemento de memória básico que é constituído de portas lógicas e é um dispositivo de lógica sequencial.

Um **registrador** é um conjunto de elementos de memória e é usado para manter uma informação até que ela seja necessária.

O chip **temporizador 555** consiste de um flip-flop SR em que as entradas são conectadas nas saídas de dois comparadores.

Problemas

- 5.1 Explique como portas lógicas podem ser usadas para controlar as seguintes situações:
- A emissão de bilhetes de passagem em uma máquina automática em uma estação de metrô.
 - O sistema de trava de segurança para operação de uma máquina-ferramenta.
 - A chave de desligamento de uma caldeira quando a temperatura atinge, digamos, 60°C e a bomba de circulação é desligada.
 - Um sinal para acionar um elevador quando a porta do elevador estiver fechada e um botão de seleção do andar for pressionado.
- 5.2 Para os sinais de tempo mostrados como A e B na Figura 5.31, qual será o sinal de saída se A e B são as entradas de (a) uma porta AND, (b) uma porta OR?

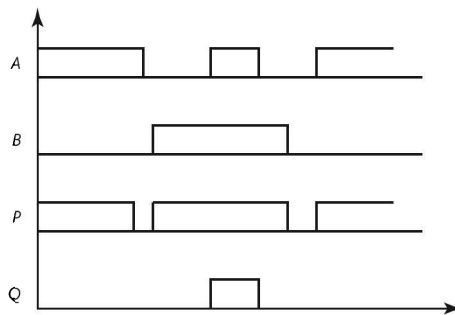


Figura 5.31 Problema 5.2.

- 5.3 Um sinal de clock, na forma de uma sequência contínua de pulsos, é aplicado em uma porta lógica e aparece na saída apenas quando um sinal de habilitação também é aplicado na porta. Qual porta lógica pode ser usada?
- 5.4 A entrada A é aplicada diretamente a uma porta de duas entradas. A entrada B é aplicada em uma porta NOT e, em seguida, na porta AND. Qual condição nas entradas A e B resultaria em uma saída 1 a partir da porta AND?
- 5.5 A Figura 5.32(a) mostra os sinais de entrada A e B aplicados ao sistema de portas mostrado na Figura 5.32(b). Desenhe as formas de onda de saída resultantes em P e Q .

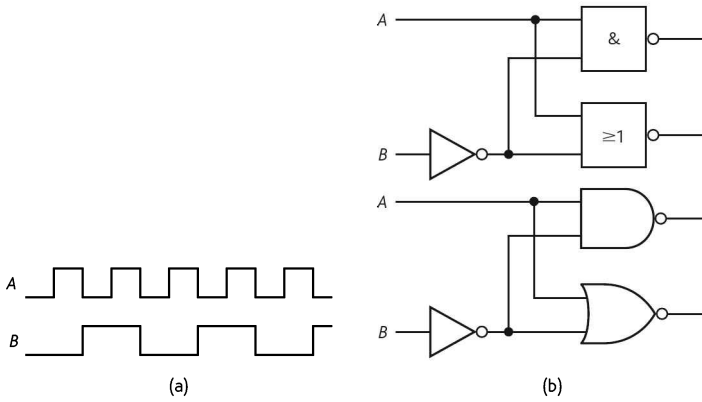


Figura 5.32 Problema 5.5.

5.6 A Figura 5.33 mostra o diagrama de temporização para as entradas S e R de um flip-flop SR. Complete o diagrama acrescentando a saída Q .

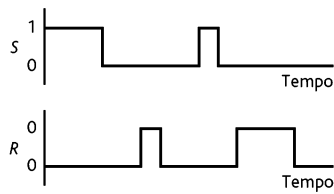


Figura 5.33 Problema 5.6.

5.7 Explique como a configuração de portas mostrada na Figura 5.34 desempenha a função de um flip-flop SR.

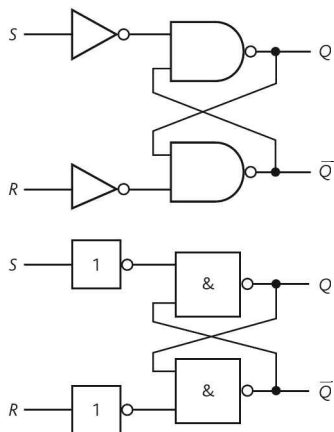


Figura 5.34 Problema 5.7.

6

Sistemas de Apresentação de Dados

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Explicar o problema do efeito de carga nos sistemas de medição.
- Identificar os elementos de apresentação de dados normalmente usados e descrever suas características.
- Explicar os princípios da gravação magnética e óptica.
- Explicar os princípios dos displays, em particular os de LED de sete segmentos e os de matriz de pontos.
- Descrever os elementos básicos dos sistemas de aquisição de dados e instrumentos virtuais.

6.1 Displays

Este capítulo aborda como os dados podem ser apresentados, por exemplo, como dígitos em um display de LEDs ou exibidos em uma tela de computador, e armazenados, por exemplo, em um disco rígido ou CD.

Os sistemas de medida consistem de três elementos: sensor, condicionador de sinal e display ou elemento de apresentação de dados (veja a Seção 1.4). Existe uma grande variedade de elementos que podem ser usados para a apresentação de dados. Tradicionalmente eles são classificados em dois grupos: indicadores e gravadores. Os **indicadores** fornecem uma saída visual imediata da variável que o sensor mede, enquanto que os **registradores** gravam o sinal de saída ao longo de um período de tempo e fornecem automaticamente uma gravação permanente.

Este capítulo também pode ser considerado como o complemento do grupo de capítulos relacionados aos sistemas de medição, ou seja, sensores, condicionamento de sinais e agora display. Desta forma, este capítulo é usado para reunir os itens ao considerar exemplos de sistemas de medição completos.

6.1.1 Efeito de carga

Um ponto geral a ser levado em consideração quando se interliga qualquer sistema de medição é o **efeito de carga**, ou seja, o efeito da conexão de uma carga nos terminais de saída de qualquer elemento de um sistema de medição.

A conexão de um amperímetro em um circuito para medir a corrente altera a resistência do circuito e, conseqüentemente, a corrente. A tentativa de fazer esta medição provoca uma alteração no circuito que altera a corrente que está sendo medida. Quando um voltímetro é conectado em um resistor, efetivamente temos duas resistências em paralelo, e se a resistência do voltímetro não for consideravelmente maior que a do resistor, a corrente através do resistor é perceptivelmente alterada, e assim a tensão medida é diferente da real. A tentativa desta medição modifica a tensão que está sendo medida. Esta ação é denominada efeito de carga.

O efeito de carga também pode ocorrer dentro de um sistema de medição quando a conexão de um elemento a outro modifica as características do elemento anterior. Considere, por exemplo, um sistema de medição que consiste de um sensor, um amplificador e um display que é elemento de apresentação da medida (Figura 6.1). O sensor tem uma tensão de saída em circuito aberto de V_s e uma resistência R_s . O amplificador tem uma resistência de entrada de R_{in} . Portanto, esta é a resistência conectada ao sensor. Assim, a tensão de entrada proveniente do sensor é dividida de modo que a diferença de potencial na carga, e portanto a tensão de entrada, V_{in} , do amplificador, é

$$V_{in} = \frac{V_s R_{in}}{R_s + R_{in}}$$

Se o amplificador tem um ganho de tensão de G , então a tensão de saída em circuito aberto dele é GV_{in} . Se o amplificador tem uma resistência de saída de R_{out} , então a tensão de saída do amplificador é dividida de forma que a diferença de potencial V_d no elemento de apresentação, resistência R_d , é

$$\begin{aligned} V_d &= \frac{GV_{in} R_d}{R_{out} + R_d} = \frac{GV_s R_{in} R_d}{(R_{out} + R_d)(R_s + R_{in})} \\ &= \frac{GV_s}{\left(\frac{R_{out}}{R_d} + 1\right) \left(\frac{R_s}{R_{in}} + 1\right)} \end{aligned}$$

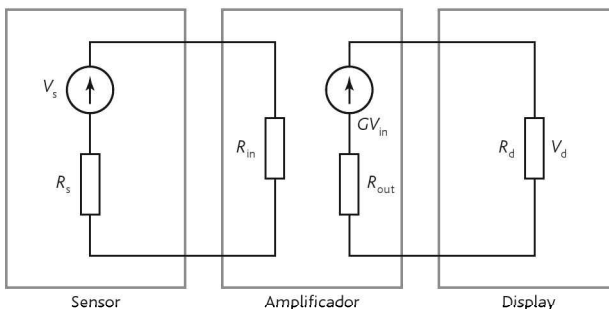


Figura 6.1 Efeito de carga em um sistema de medição.

Portanto, para que os efeitos de carga sejam desprezíveis é necessário que $R_{out} \gg R_d$ e $R_s \gg R_{in}$.

6.2 Elementos de apresentação de dados

Esta seção apresenta uma abordagem geral de exemplos de elementos de apresentação de dados.

6.2.1 Medidores analógicos e digitais

Um **medidor de bobina móvel** é um indicador analógico com um ponteiro que se move sobre uma escala. O galvanômetro básico é um microamperímetro CC com *shunts* (resistências de desvio em paralelo), multiplicadores e retificadores usados para se obter outras faixas de medição de corrente e tensão contínua e alternada. Nos casos de correntes e tensões alternadas, o instrumento é restrito às frequências entre 50 Hz e 10 kHz. A precisão deste medidor depende de alguns fatores, dentre eles a temperatura, a presença de campos magnéticos próximos ou materiais ferrosos, a maneira com que o medidor é montado, fricção do mancal, imprecisões na escala no processo de fabricação, etc. Além disso, existem erros que envolvem a leitura do medidor, por exemplo, erros de paralaxe em que a leitura do medidor com o ponteiro sobre a escala é feita em um ângulo que não é reto em relação à escala, e erros que surgem da estimativa do posicionamento do ponteiro entre marcas na escala. Em geral, a precisão é da ordem de $\pm 0,1$ a $\pm 0,5\%$. O tempo gasto para que um medidor de bobina móvel alcance a deflexão estacionária é tipicamente de alguns segundos. A baixa resistência do medidor pode apresentar problemas de efeito de carga.

O **voltímetro digital** apresenta sua leitura na forma de uma sequência de dígitos. Um display desta forma elimina os erros de paralaxe e interpolação e pode apresentar precisões tão altas quanto $\pm 0,005\%$. O voltímetro digital é essencialmente uma unidade de amostragem e retenção conectada a um conversor analógico-digital cuja saída é a contagem de um contador (Figura 6.2). Ele tem uma alta resistência, da ordem de $10\text{ M}\Omega$, e, desta forma, os seus efeitos de carga são bem menores em comparação a um medidor de bobina móvel com uma resistência menor. Portanto, se a especificação de um voltímetro digital inclui a afirmação “taxa de amostragem de aproximadamente 5 leituras por segundo”, então isto significa que a cada $0,2\text{ s}$ a tensão de entrada é amostrada. Este é o tempo para o instrumento processar o sinal e fornecer a leitura. Portanto, se a tensão de entrada variar a uma taxa que resulte em variações significativas durante $0,2\text{ s}$, então a leitura do voltímetro pode apresentar erro. Um voltímetro digital de baixo custo tem tipicamente uma taxa de amostragem de 3 por segundo e uma impedância de entrada de $100\text{ M}\Omega$.

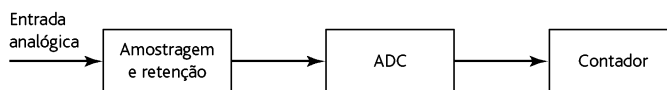


Figura 6.2 Princípio do voltímetro digital.

6.2.2 Registradores gráficos analógicos

Os registradores gráficos analógicos registram os dados em papel através de penas com pontas de fibra à base de tinta, através do impacto de uma ponta que pressiona uma fita de carbono contra o papel, através do uso de um papel termicamente sensível que muda de cor quando uma ponta aquecida se move sobre ele, através de um feixe de luz ultravioleta que incide em um papel sensível a essa luz e através de um fio de tungstênio que se move sobre a superfície de um papel especialmente coberto de uma fina camada de alumínio sobre uma tinta colorida, e uma descarga elétrica que remove o alumínio expondo a tinta. Em muitas aplicações, eles foram substituídos por instrumentos virtuais (abordados mais adiante neste capítulo).

6.2.3 Osciloscópio de raios catódicos

O osciloscópio de raios catódicos é um instrumento de medição de tensão que é capaz de mostrar sinais de frequências extremamente altas. Um instrumento deste de propósito geral pode responder a sinais de até 10 MHz, enquanto que instrumentos mais específicos podem responder a até 1 GHz. Os osciloscópios de duplo feixe permitem que dois traços independentes sejam mostrados simultaneamente na tela. Os osciloscópios de memória são capazes de manter o traço na tela após o sinal ter cessado, sendo removido apenas por uma ação que apaga o traço na tela. Os osciloscópios de memória digital digitalizam o sinal de entrada e armazenam o sinal digital em uma memória. Então, o sinal pode ser analisado e manipulado e mostrado na tela do osciloscópio no formato analógico, sendo este obtido a partir da reconstrução do sinal analógico. A memorização permanente do traço pode ser feita com o uso de câmeras de aplicação especial fixadas diretamente no osciloscópio.

Um osciloscópio de propósito geral tem geralmente deflexão vertical, ou seja, deflexão em Y, sensibilidade que varia entre 5 mV e 20 V por divisão da escala. Para que as componentes CA possam ser visualizadas na presença de tensões CC altas, pode-se comutar um capacitor de bloqueio da componente CC na linha de entrada. Quando o amplificador está em modo ca, sua largura de banda se estende de 2 Hz a 10 MHz, e quando em modo CC, tem extensão desde CC a 10 MHz. A impedância da entrada Y é tipicamente em torno de 1 M Ω com um capacitor *shunt* de 20 pF. Quando é conectado um circuito externo na entrada Y, os problemas devido ao efeito de carga e interferência podem distorcer o sinal de entrada. Apesar da interferência poder ser reduzida com o uso de cabo coaxial, a capacitância deste cabo e de qualquer ponta de prova conectada a ele pode ser suficiente, particularmente em baixas frequências, para introduzir uma impedância relativamente baixa na impedância de entrada do osciloscópio e, desta forma, introduzir um efeito de carga significativo. Existem várias pontas de prova para conexão com o cabo de entrada que são projetadas para aumentar a impedância de entrada, evitando este problema de efeito de carga. Uma ponta de prova de tensão passiva normalmente usada tem um atenuador de 10 para 1 (Figura 6.3). Esta tem um resistor de 9 M Ω e um capacitor variável na extremidade da ponta. Entretanto, isto reduz não apenas a carga capacitiva, mas também a sensibilidade de tensão, e por isso é frequente o uso de uma ponta de prova ativa que emprega um FET.

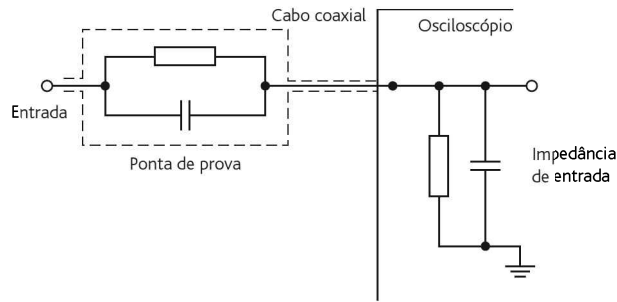


Figura 6.3 Ponta de prova de tensão passiva.

6.2.4 Unidade de apresentação de dados

Cada vez mais os dados de saída são apresentados usando-se uma unidade de apresentação de dados (VDU – *Visual Display Unit*). No caso da tela de um tubo de raios catódicos, a imagem é formada pelo movimento de um feixe de elétrons em uma série de linhas de varredura horizontal, percorrendo uma após a outra até a parte de baixo da tela. A imagem é formada através da variação da intensidade do ponto luminoso na tela à medida que cada linha é percorrida. Esta imagem formada por linhas é denominada **não entrelaçada** (Figura 6.4(a)). Para reduzir os efeitos de *flicker* (cintilação), são usadas duas varreduras percorrendo a tela de cima para baixo para gerar uma imagem completa. Na primeira varredura da tela, todas as linhas ímpares são percorridas, e na segunda, todas as linhas pares. Esta técnica é denominada de **varredura entrelaçada** (Figura 6.4(b)).

A tela de uma unidade de apresentação de dados é coberta com uma grande quantidade de pontos de fósforo que formam os **pixels**. Este termo é usado para o menor ponto endereçável em um dispositivo display. Um caractere de texto ou um diagrama é produzido na tela iluminando-se seletivamente estes pontos luminosos. A Figura 6.5 mostra, em uma matriz 7×5 , como os caracteres são gerados através de um feixe de elétrons que se move em ziguezague percorrendo a tela de cima para baixo. Os dados de entrada da unidade de apresentação de dados são geralmente digitais e no padrão **ASCII** (*American Standard Code for Infor-*

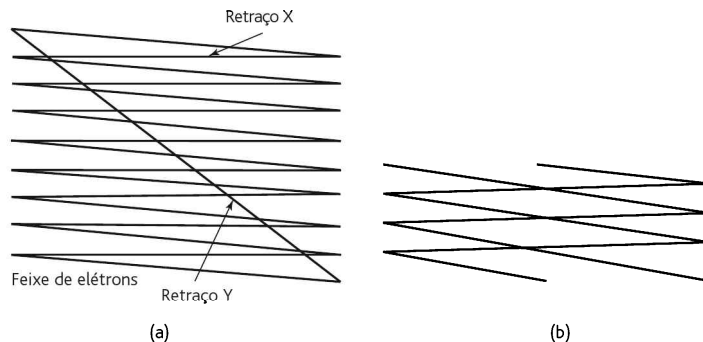


Figura 6.4 (a) Display não entrelaçado, (b) entrelaçado.

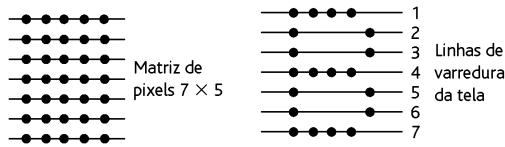


Figura 6.5 Caractere construído através de pontos luminosos.

mation Interchange – Código Padrão Americano para Troca de Informações). Este é um código de 7 bits e pode ser usado para representar $2^7 = 128$ caracteres. Ele permite que todos os caracteres de um teclado padrão sejam convertidos, bem com algumas funções de controle como RETURN, que é usada para indicar o retorno do final de uma linha para o início da próxima. A Tabela 6.1 mostra uma lista abreviada deste código.

6.2.5 Impressoras

As impressoras têm a função de gravar uma informação em papel. Existem vários tipos de impressoras: a de matriz de ponto, a de jato de tinta e a laser.

A **impressora de matriz de ponto** tem um cabeçote de impressão (Figura 6.6) que consiste de 9 ou 24 pinos em uma linha vertical. Cada pino é controlado por um eletroímã que, quando acionado, impulsiona o pino contra a fita tintada. Isto transfere uma pequena gota da tinta para o papel atrás da fita. Um caractere é formado movendo-se o cabeçote de impressão em linhas horizontais de um lado para outro sobre o papel e acionando-se os pinos apropriados.

A **impressora jato de tinta** usa uma tinta condutiva que é forçada a passar por um pequeno bico para produzir um jato de gotas de tinta muito pequenas de diâmetro constante em uma frequência também constante. Com isto forma-se um fluxo constante de tinta que passa ao longo de um tubo e é pulsado para formar finas gotas por um cristal piezoelétrico que vibra em uma frequência de aproximada-

Tabela 6.1 Código ASCII

Caractere	ASCII	Caractere	ASCII	Caractere	ASCII
A	100 0001	N	100 1110	0	011 0000
B	100 0010	O	100 1111	1	011 0001
C	100 0011	P	101 0000	2	011 0010
D	100 0100	Q	101 0001	3	011 0011
E	100 0101	R	101 0010	4	011 0100
F	100 0110	S	101 0011	5	011 0101
G	100 0111	T	101 0100	6	011 0110
H	100 1000	U	101 0101	7	011 0111
I	100 1001	V	101 0110	8	011 1000
J	100 1010	W	101 0111	9	011 1001
K	100 1011	X	101 1000		
L	100 1100	Y	101 1001		
M	100 1101	Z	101 1010		

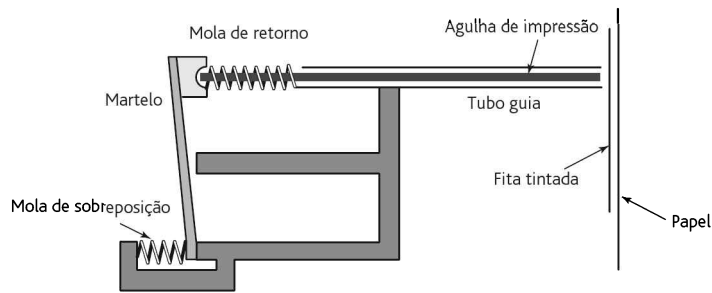


Figura 6.6 Mecanismo do cabeçote de impressora matricial.

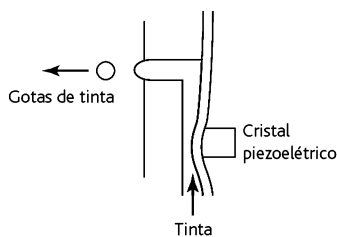


Figura 6.7 Produção de uma série de gotas.

mente 100 kHz (Figura 6.7). Um outro tipo usa um pequeno aquecedor no cabeçote de impressão com tinta vaporizada em um tubo capilar, produzindo assim bolhas de gás que expulsam gotas de tinta (Figura 6.8). Em uma outra versão de impressora, cada gota de tinta recebe uma carga elétrica como resultado da passagem através de um eletrodo carregado, e estas gotas carregadas são defletidas ao passarem entre placas entre as quais é mantido um campo elétrico; em outra versão, uma pilha de agulhas é usada e cada jato é ligado ou desligado conforme a necessidade. As impressoras jato de tinta podem imprimir em cores através do uso de um sistema de três jatos de tinta colorida. O tamanho das gotas é tão pequeno que as impressoras podem produzir mais de 600

pontos por polegada.

A impressora laser tem um tambor fotossensível coberto com um material sensível à luz à base de selênio (Figura 6.9). No escuro, o selênio tem uma alta resistência e, conseqüentemente, se torna eletricamente carregado à medida que passa próximo a um fio com carga e tensão. Um feixe de luz percorre o tambor ao longo do seu comprimento através de um pequeno espelho de rotação de oito faces. Quando a luz atinge o selênio, a resistência deste cai, não permanecendo mais carregado. Através do controle da intensidade do feixe de luz, pontos no tambor podem ser descarregados ou deixados carregados. À medida que o tambor passa pelo reservatório de toner, as áreas com carga atraem partículas do toner, que aderem às áreas que não foram expostas à luz, diferentemente do que acontece com as áreas expostas

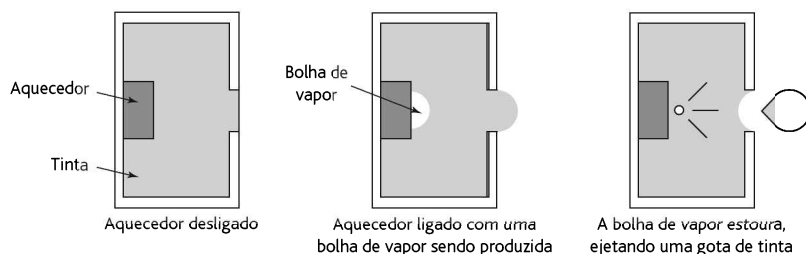


Figura 6.8 Princípio do jato de bolhas.

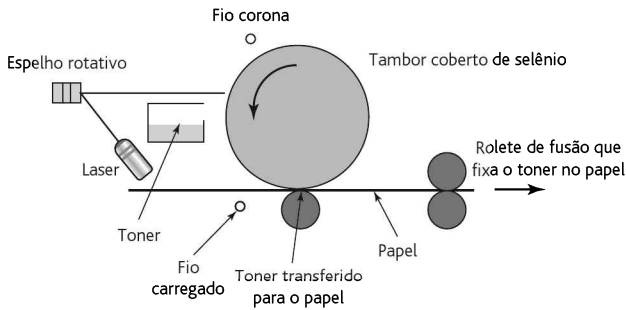


Figura 6.9 Elementos básicos de uma impressora laser.

à luz. O papel recebe uma carga elétrica à medida que passa próximo a um outro fio carregado, denominado fio corona, de forma que, ao passar próximo ao tambor, ele atrai o toner do tambor. Um rolete de fusão aquecido é então usado para derreter as partículas do toner de forma que, após passar entre os roletes, elas aderem firmemente ao papel. Em geral as impressoras laser são capazes de produzir impressões com 1200 pontos por polegada.

6.3 Gravação magnética

A gravação magnética é usada para armazenamento de dados em discos flexíveis e discos rígidos de computadores. Os princípios básicos compreendem um cabeçote de gravação, que responde a um sinal de entrada e produz padrões magnéticos correspondentes em uma fina camada de material magnético, e um cabeçote de leitura que força uma saída convertendo os padrões magnéticos sobre o material magnético em sinais elétricos. Além destes cabeçotes, é necessário um sistema de transporte que move o material magnético de forma controlada sob os cabeçotes.

A Figura 6.10(a) mostra os elementos básicos do cabeçote de gravação; ele consiste de um núcleo de material ferromagnético que tem um entreferro. Quando os sinais elétricos acionam a bobina que está em torno do núcleo, um fluxo magnético é produzido no núcleo. A proximidade do disco plástico, coberto magneticamente, com o entreferro faz com que o fluxo magnético percorra um caminho através do núcleo e da parte da cobertura magnética na região próxima ao entreferro. Quando há um fluxo magnético passando na região do disco com a cobertura magnética, esta se torna permanentemente magnetizada. Portanto, uma gravação magnética é produzida por um sinal elétrico de entrada. A inversão da direção da corrente inverte a direção do fluxo.

O cabeçote de leitura (Figura 6.10(b)) tem uma construção similar ao de gravação. Quando uma parte da cobertura magnética do disco reduz o entreferro, então um fluxo magnético é induzido no núcleo. As variações de fluxo no núcleo induzem forças eletromotrizes (fems) na bobina em torno do núcleo. Portanto, a saída da bobina é um sinal elétrico que está relacionado à gravação magnética na superfície do disco.

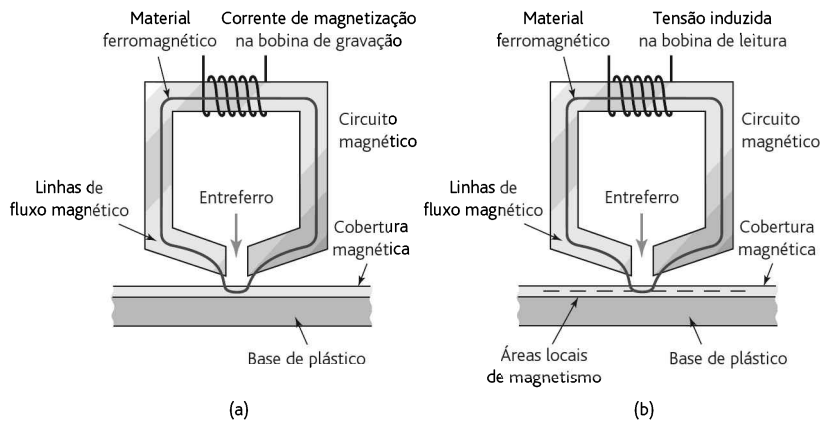


Figura 6.10 (a) Princípio da gravação magnética, (b) cabeçote de leitura magnética.

6.3.1 Códigos de gravação magnética

A gravação digital envolve a gravação de sinais como uma combinação codificada de bits. Uma célula de bit é o elemento da superfície magnética em que o magnetismo é saturado completamente em uma direção ou na direção contrária. A saturação ocorre quando o campo de magnetização foi aumentado de tal forma que o material magnético atinge o seu fluxo magnético máximo e aumentos adicionais na corrente de magnetização não produzem variações.

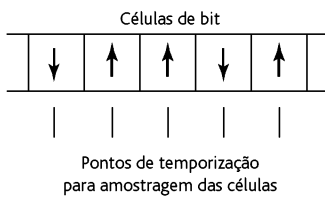


Figura 6.11 Células de bit.

As células de bit na superfície magnética podem ser como mostra a Figura 6.11. Um método óbvio de colocar os dados no material magnético pode ser fazendo-se uso do fluxo magnético em uma direção para representar um 0 e na direção contrária para representar um 1. Entretanto, é necessário ler cada célula e, assim, faz-se necessário pontos de temporização precisos para indicar exatamente quando a amostragem deve ocorrer. Problemas podem surgir se for usado algum clock externo para gerar os sinais de temporização, como um pequeno erro entre os sinais de temporização e a taxa na qual a superfície magnética se

move sob o cabeçote de leitura pode resultar talvez em uma célula não lida ou lida duas vezes. O sincronismo é essencial. Este é conseguido usando-se as próprias células de bit para gerar os sinais da amostragem. Um método é usar as transições na superfície magnética a partir das mudanças nos sentidos de saturação, ou seja, onde a demarcação entre dois bits for claramente evidente, para sinalizar para a geração do sinal de temporização a fim de ajustá-lo de forma a ter sincronismo com as células de bit.

Se a reversão de fluxo não ocorrer em uma frequência suficiente, este método de sincronismo ainda pode resultar na ocorrência de erros. Uma forma de superar este problema é usar uma forma de codificação. Os métodos a seguir são alguns dos mais usados:

1. Sem retorno para zero (NRZ – Non-Return-to-Zero)

Com este sistema, o fluxo é gravado no disco de modo que nenhuma variação no fluxo representa um 0 e uma variação representa um 1 (Figura 6.12(a)). Entretanto, ele não tem clock próprio.

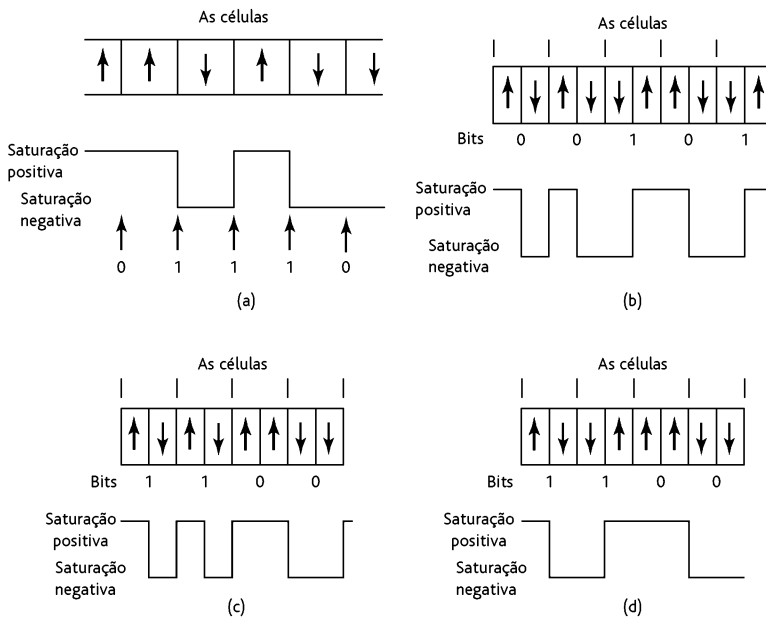


Figura 6.12 Modulação (a) sem retorno para zero (NRZ), (b) de fase (PE), (c) de frequência (FM), (d) de frequência modificada (MFM).

2. Codificação de fase (PE – Phase Encoding)

A codificação de fase tem a vantagem de ter clock próprio, sem a necessidade de um sinal de clock externo. Cada célula é dividida em duas, sendo metade com fluxo de saturação positivo e a outra metade, negativo. Então um dígito 0 é gravado como uma saturação positiva de meio bit seguida de uma saturação negativa na outra metade; um dígito 1 é representado por uma saturação negativa em meio bit seguida por uma saturação positiva na outra metade. A transição no meio da célula do positivo para o negativo indica um 0, e do negativo para o positivo indica um 1 (Figura 6.12(b)).

3. Modulação de frequência (FM – Frequency modulation)

Este tem um clock próprio e é similar à codificação de fase, porém sempre há uma inversão direcional no fluxo no início de cada célula (Figura 6.12(c)). No caso do bit 0 não há uma inversão de fluxo adicional durante a célula. Esta característica é contrária à do bit 1.

4. Modulação de frequência modificada (MFM – Modified Frequency Modulation)

Esta é uma modificação do código de modulação de frequência. A diferença é que a reversão de fluxo no início de cada código de bit existe apenas se os bits atual e anterior forem 0 (Figura 6.12(d)). Isto significa que apenas uma inversão de fluxo é necessária para cada bit. Este e o código de comprimento limitado são os mais usados em discos magnéticos.

5. Comprimento limitado (RLL – Run Length Limited)

Este é um grupo de códigos com clock próprio que especifica uma distância mínima e máxima, ou seja, a duração entre fluxos reversos. A duração máxi-

ma é curta o suficiente para garantir que os fluxos reversos sejam suficientemente frequentes para o código ter clock próprio. Uma forma normalmente usada deste código é a RLL_{2,7}, em que 2,7 significa que a distância mínima entre reversões de fluxo é de 2 bits e a máxima é de 7. A sequência de códigos é descrita como uma sequência de códigos S e códigos R. Um código S, código de espaço, não tem reversões de fluxo, enquanto que o código R, código de reversão, tem uma reversão durante o bit. Dois códigos S/R são usados para representar cada bit. Os bits são agrupados em sequências de 2, 3 e 4 bits, e um código é associado a cada grupo, sendo os códigos mostrados a seguir:

Sequência de bits	Sequência do código
10	SRSS
11	RSSS
000	SSSRSS
010	RSSRSS
011	SSRSSS
0010	SSRSSRSS
0011	SSSSRSSS

A Figura 6.13 mostra a codificação para a sequência 0110010, sendo dividida nos grupos 011 e 0010 e assim representada por SSRSSSRSSRSS. Há pelos menos dois códigos S entre códigos R, e não pode haver mais de sete códigos S entre códigos R.

O código ótimo é aquele que permite que os bits sejam “empacotados” tão próximos quanto possível e que possam ser lidos sem erro. Os cabeçotes de leitura podem localizar inversões muito facilmente, mas eles não podem estar muito próximos um do outro. O código RLL tem a vantagem de ser mais compacto que os outros, sendo os códigos PE e FM os que ocupam o maior espaço. Os códigos MFM e NRZ ocupam o mesmo espaço. O NRZ tem a desvantagem de, diferentemente dos outros códigos, não ter clock próprio.

6.3.2 Discos magnéticos

A gravação digital frequentemente é feita em discos rígidos. Os dados digitais são armazenados na superfície do disco ao longo de círculos concêntricos denominados trilhas. Em um disco simples cabem muitas trilhas. É usado um cabeçote de leitura/escrita único para cada superfície de disco, e eles são movidos, através de um atua-

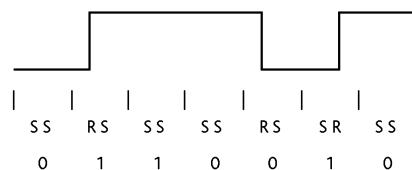


Figura 6.13 Código RLL.

dor mecânico, para frente e para trás para ter acesso às diferentes trilhas. O disco é girado por um acionador e os cabeçotes de leitura/escrita leem ou escrevem dados em uma trilha. Os discos rígidos (Figura 6.14(a)) são unidades hermeticamente fechadas com os dados armazenados na superfície ao longo de círculos concêntricos. Um disco rígido é constituído por mais de um disco e os dados são armazenados nos dois lados de cada disco. Estes são rotacionados em alta velocidade e as trilhas são acessadas pelo movimento dos cabeçotes de leitura/escrita. Uma grande quantidade de dados pode ser armazenada em discos como estes; capacidades de armazenamentos de centenas de gigabytes são comuns atualmente.

A superfície do disco é dividida em setores (Figura 6.14(b)) e, desta forma, uma unidade de informação no disco tem um endereço que consiste de um número de trilha e um número de setor. Um disco flexível tem normalmente de 8 a 18 setores e cerca de 100 trilhas. Um disco rígido pode ter cerca de 2.000 trilhas por superfície e 32 setores. Para procurar um dado, o cabeçote tem que ser movido ao longo da trilha desejada, e o tempo gasto para isto é denominado de **tempo de busca**; então ele espera que o setor desejado passe sob ele, sendo este tempo denominado de **latência**. Para que um endereço possa ser identificado, é necessário que uma informação seja gravada no disco para identificar os segmentos e as trilhas. A gravação destas informações é denominada **formatação** e tem que ser feita antes que os dados sejam armazenados no disco. Geralmente, a técnica usada é armazenar a informação desta localização nas trilhas de modo que, quando os dados são armazenados, a sequência da informação na trilha seja:

marca de índice,
cabeçalho do setor 0, dados do setor 0, reboque do setor 0,
cabeçalho do setor 1, dados do setor 1, reboque do setor 1,
cabeçalho do setor 2, dados do setor 2, reboque do setor 2,
etc.

A marca de índice contém o número da trilha com o cabeçalho que identifica o setor. O reboque do setor contém informação, ou seja, verificação de redundância cíclica, que pode ser usada para verificar que um setor foi lido corretamente.

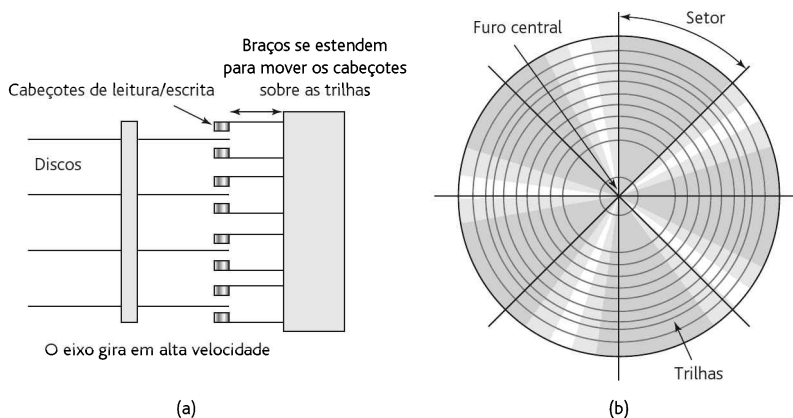


Figura 6.14 Disco rígido: (a) arranjo de discos, (b) trilhas e setores.

6.4 Gravação óptica

Assim como os discos magnéticos, os CD-ROMs armazenam os dados ao longo de suas trilhas. Diferentemente de um disco magnético que tem uma série de trilhas concêntricas, um CD-ROM tem uma trilha espiral. A superfície de gravação é coberta com alumínio e é altamente reflexiva. A informação é então armazenada em uma trilha de cerca de $0,6\ \mu\text{m}$ de largura como uma série de pequenas covas (*pits*), gravadas em relevo na superfície através de um feixe de luz laser com diâmetro de cerca de $1\ \mu\text{m}$. Isto resulta em uma reflexão intensa, se a luz incide em uma área da superfície sem depressão, ou ausência de reflexão, se a luz incide em uma depressão na superfície do disco. Os dados são lidos como uma sequência de pulsos de luz refletida e não refletida.

A gravação óptica usa métodos similares de codificação aos usados na gravação magnética, sendo a codificação RLL a mais usada. Devido às gravações ópticas serem facilmente corrompidas por arranhões ou obstrução por sujeira do feixe laser usado na leitura do disco, métodos têm que ser usados para detectar e corrigir erros. Um método é a **verificação de paridade**. Com este método, os grupos de bits são ampliados com um bit extra, o de paridade, que pode ser 0 ou 1 conforme o número total de bits 1 no grupo seja ímpar ou par. Quando a informação é lida, se um dos bits foi corrompido, então o número de bits 1 é alterado e detectado como um erro.

6.5 Displays

Muitos sistemas de display usam indicadores luminosos para mostrar estados *on/off* ou mostrar caracteres alfanuméricos. O termo **alfanumérico** é uma contração dos termos “alfabeto” e “numérico” e descreve displays que mostram as letras do alfabeto e os números de 0 a 9 com vírgula decimal. Um display deste tipo envolve sete segmentos de luz para gerar caracteres alfabéticos e numéricos. A Figura 6.15 mostra os segmentos e a Tabela 6.2 mostra como um código binário de 4 bits pode ser usado para gerar entradas para acionar os segmentos.

Um outro formato de display usa uma matriz de pontos 7×5 ou 9×7 (Figura 6.16). Os caracteres são gerados pela ativação dos pontos apropriados.

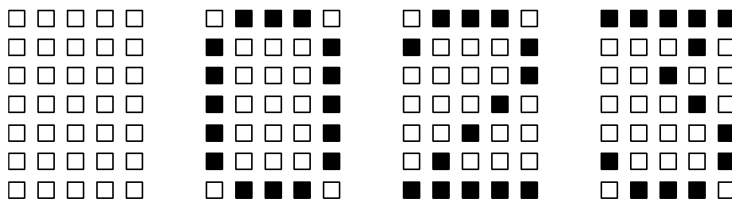
Os indicadores luminosos para estes displays podem ser de lâmpadas néon, lâmpadas incandescentes, **diodos emissores de luz** (LEDs – *Light-Emitting Diodes*) ou **displays de cristal líquido** (LCD – *Liquid Crystal Displays*). As **lâmpadas néon** precisam de tensões altas e baixas correntes e podem ser alimentadas diretamente a partir da tensão principal, mas produzem apenas luz ver-



Figura 6.15 Display de sete segmentos.

Tabela 6.2 Display de sete segmentos

Entrada binária				Segmentos ativados							Número mostrado
				a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	0	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9

Figura 6.16 Um display de matriz de pontos de 7×5 .

melha. As **lâmpadas incandescentes** podem ser usadas em uma ampla faixa de tensões, mas necessitam comparativamente de uma corrente alta. Elas emitem luz branca e, a partir do uso de lentes, pode-se obter a cor desejada. A principal vantagem delas é o brilho.

6.5.1 Diodos emissores de luz

Os LEDs necessitam de baixas tensões e baixas correntes e são baratos. Esses diodos, quando polarizados diretamente, emitem luz em uma faixa de comprimentos de onda. A Figura 6.17 mostra a forma básica de um LED, a luz emitida a partir do diodo sendo concentrada em uma direção com o uso de refletores. Os materiais normalmente usados nos LEDs são o arsenieto de gálio, fosfeto de gálio e ligas de arsenieto de gálio com fosfeto de gálio. Os LEDs normalmente usados fornecem luz vermelha, amarela ou verde. Nos sistemas microprocessados, os LEDs são indicadores mais comuns.

Geralmente é necessário um resistor de limitação de corrente para que a corrente no LED seja abaixo da corrente máxima especificada, que é cerca de 10 a 30 mA. Tipicamente um LED tem uma queda de tensão de 2,1 V quando a corrente é limitada em 20 mA. Assim, quando, digamos, uma

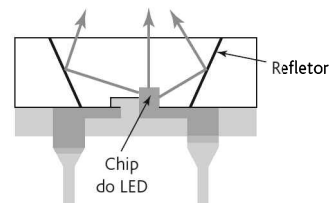


Figura 6.17 LED.

tensão de 5 V for aplicada, a queda de tensão no resistor em série será de 2,9 V. Isto significa que uma resistência de $2,9/0,020 = 145 \Omega$ é necessária e que um resistor padrão de 150Ω deve ser usado. Alguns LEDs são fornecidos com resistores internos, de modo que eles podem ser conectados diretamente a sistemas microprocessados.

Os LEDs são disponibilizados na forma de indicadores luminosos simples, display alfanuméricos de sete e dezesseis segmentos, no formato de matriz de pontos e de *bar graph* (barra de LEDs em formatos diversos).

A Figura 6.18(a) mostra como sete LEDs, que formam um display de sete segmentos como o da Figura 6.16, podem ser conectados a um circuito de acionamento de modo que, quando a linha for acionada em nível baixo, uma tensão é aplicada e o LED conectado àquela linha é ligado. A tensão tem que estar acima de um valor mínimo de operação antes que o LED emita uma luz significativa; tipicamente esta tensão é de cerca de 1,5 V. Esta configuração é conhecida como **anodo comum**, visto que todos os anodos dos LEDs são conectados juntos. Uma configuração alternativa é o **catodo comum** (Figura 6.18(b)). Os elementos na configuração anodo comum são ativados através de uma entrada em nível baixo, e no tipo catodo comum, através de uma entrada em nível alto. Geralmente é escolhido o tipo anodo comum, pois a direção da corrente e a sua intensidade são mais apropriadas.

Exemplos destes tipos de displays de sete segmentos de 7,6 mm e 10,9 mm da Hewlett Packard estão disponíveis nas configurações anodo comum e catodo comum. Além dos sete segmentos que formam os caracteres, há ainda o ponto decimal* à esquerda ou à direita. Com a iluminação de segmentos variados podem-se formar todos os dígitos numéricos e uma pequena quantidade de caracteres do alfabeto.

Normalmente, a saída do circuito acionador não está no formato binário normal, mas em **decimal codificado em binário** (BCD – *Binary-Coded Decimal*)

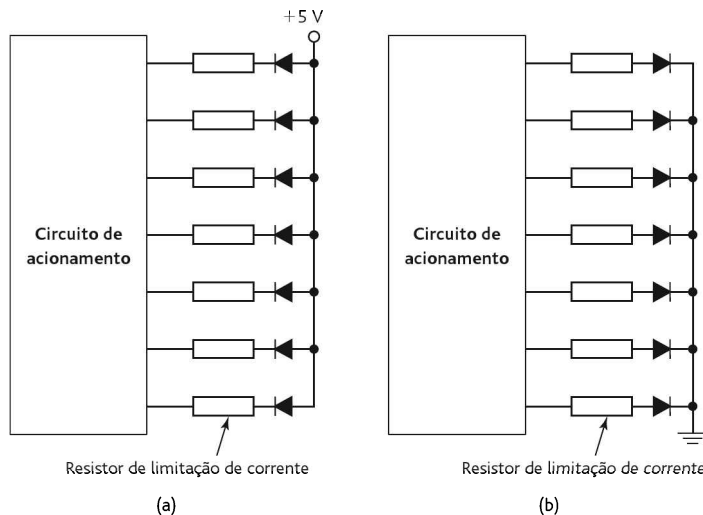


Figura 6.18 (a) Configuração anodo comum para LEDs, (b) catodo comum.

* N. de T.: Equivale à vírgula em português.

(veja o Apêndice B). Em BCD, cada dígito decimal é decodificado separadamente em binário. Por exemplo, o número decimal 15 tem o 1 codificado como 0001 e o 5, como 0101, resultando no código BCD 0001 0101. A saída do circuito acionador tem que ser decodificada no formato que o display de LEDs necessita. O 7447 é normalmente usado para decodificar e acionar displays (Figura 6.19).

A Figura 6.20 mostra a forma básica usada para um display de LEDs de matriz de pontos. O arranjo consiste de cinco conexões de coluna, sendo que cada uma reúne a conexão dos anodos de sete LEDs. Cada linha conecta os catodos de cinco LEDs. Para ligar um determinado LED, a alimentação é aplicada em sua coluna e sua linha é aterrada. Este tipo de display pode mostrar todos os caracteres ASCII.

6.5.2 Displays de cristal líquido

Os **displays de cristal líquido** não produzem luz própria, mas contam com a luz refletida ou transmitida. O material cristal líquido é composto por longas moléculas em forma de barra e colocado entre duas folhas de polímero contendo ranhuras microscópicas. As folhas inferior e superior são ranhuradas em direções que formam 90° uma em relação à outra. As moléculas do cristal líquido se alinham com as ranhuras no polímero e adotam uma suave rotação de 90° entre elas (Figura 6.21).

Quando uma luz polarizada plana incide no cristal líquido, seu plano de polarização é rotacionado conforme passa pelo material. Portanto, se o material está entre duas folhas do polarizador com suas direções de transmissão em ângulo reto, a rotação permite que a luz seja transmitida e assim o material se mostra claro.

Entretanto, se um campo elétrico for aplicado no material, as moléculas se tornam alinhadas com o campo e a luz passa pelo polarizador superior e, por não ser rotacionada, não pode passar pelo polarizador inferior, sendo absorvida. O material então se mostra escuro.

Esta estrutura é colocada entre duas folhas de vidro que possuem eletrodos transparentes na forma do display de sete segmentos, e desta forma, a aplicação de tensões nos elementos do display resulta em uma aparência escura em contraste com o display claro nas áreas onde não há campo elétrico. Este é o tipo de display usado em dispositivos que funcionam com bateria, como relógios e calculadoras. Também é disponível o tipo matriz de pontos de 5×7 .

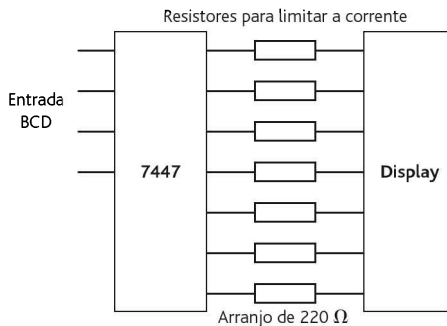


Figura 6.19 Decodificador com display de sete segmentos.

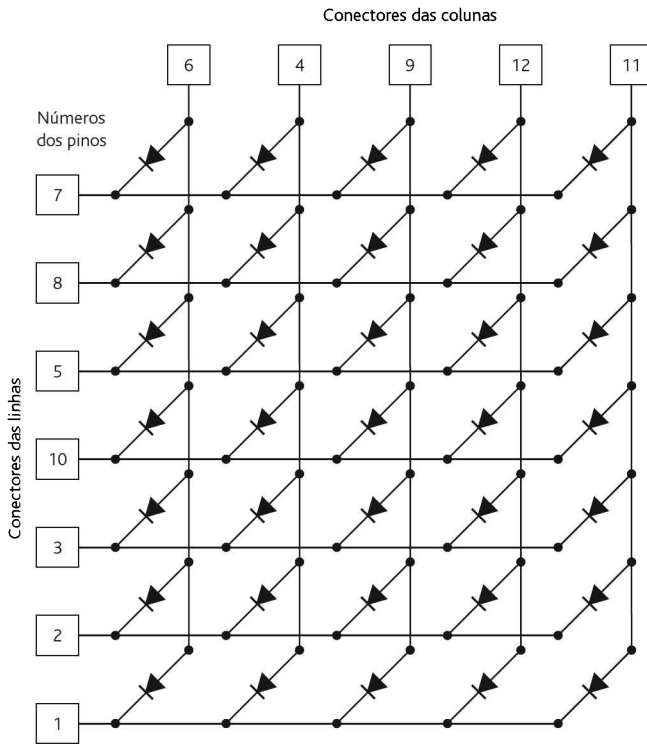


Figura 6.20 Display de matriz de pontos.

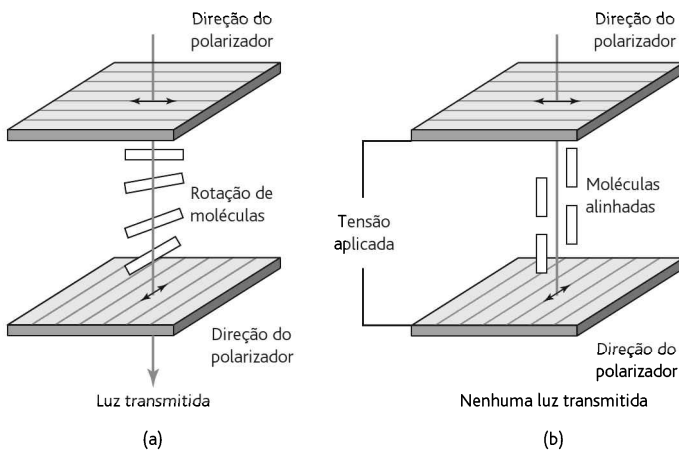


Figura 6.21 Cristal líquido: (a) sem campo elétrico, (b) com campo elétrico.

6.6 Sistemas de aquisição de dados

O termo **aquisição de dados** (DAQ – *Data Aquisition*) tende a ser usado frequentemente para sistemas nos quais as entradas dos sensores são convertidas no formato digital para processamento, análise e apresentação de resultados por um computador. O sistema contém: sensores, fiação para conectar os sensores ao condicionamento de sinal para realizar, talvez, filtragem e amplificação, hardware de aquisição de dados para realizar funções como a conversão da entrada para o formato digital e a conversão de sinais de saída para o formato analógico para os sistemas de controle, o computador e o software de aquisição de dados. Os softwares fazem a análise dos sinais de entrada digitais. Estes sistemas também são projetados para exercer igualmente funções de controle.

6.6.1 Computador com placas *plug-in*

A Figura 6.22 mostra os elementos básicos de um sistema de aquisição de dados usando placas *plug-in* com um computador para o hardware de aquisição de dados. O condicionamento de sinal anterior às entradas na placa depende dos sensores envolvidos, por exemplo, se forem termopares: amplificação, compensação de junção fria e linearização; se for para *strain gauge*: ponte de Wheatstone, fonte de alimentação para a ponte e linearização; se for para RTDs: fonte de corrente, circuito e linearização.

Para a escolha de placas DAQ, os seguintes critérios devem ser considerados:

1. Que tipo de sistema de software de computador será usado, por exemplo Windows, MacOS?
2. Que tipo de conector é disponibilizado para a conexão da placa, por exemplo, PCMCIA para laptops, NuBus para MacOS, PCI?
3. Quantas entradas analógicas serão necessárias e quais são as suas faixas de operação?
4. Quantas entradas digitais serão necessárias?
5. Que resolução será necessária?
6. Qual é a mínima taxa de amostragem necessária?
7. Serão necessários sinais de temporização ou contagem?

A Figura 6.23 mostra os elementos básicos de uma placa DAQ. Algumas placas são projetadas apenas para operar com entradas/saídas analógicas e outras para entradas/saídas digitais.

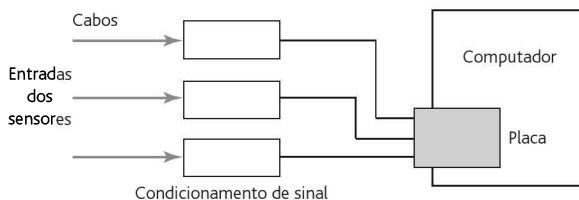


Figura 6.22 Sistema de aquisição de dados.

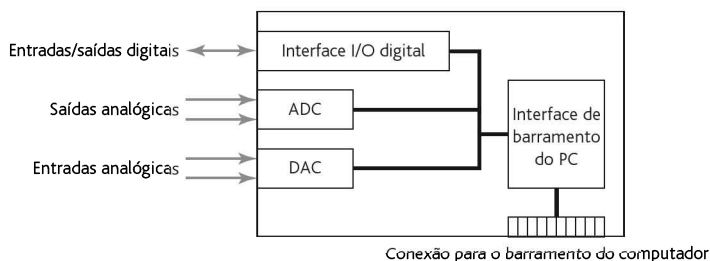


Figura 6.23 Elementos de uma placa DAQ.

Todas as placas DAQ usam **drivers**, que são softwares geralmente fornecidos pelo fabricante da placa juntamente com a placa para comunicar com o computador e informar a ele o que foi inserido e como ele pode se comunicar com a placa. Antes da placa ser usada, três parâmetros têm que ser definidos. Estes são os endereços dos canais de entrada e saída, o nível de interrupção e o canal que será usado para acesso direto à memória. Com as placas *plug and play* para uso com o software Windows, estes parâmetros são definidos pelo software; caso contrário, existem microchaves na placa para definição destes parâmetros conforme as instruções fornecidas no manual da placa.

Podem ser usados softwares aplicativos para ajudar no projeto de sistemas de medição e na análise dos dados. Um exemplo deste tipo de software disponível é o LabVIEW, que é um pacote de software de programação gráfica desenvolvido pela National Instruments para aquisição de dados e controle de instrumentos. Os programas LabVIEW, são denominados **instrumentos virtuais** porque na aparência e na operação eles imitam instrumentos reais. Um instrumento virtual tem três partes: um painel frontal, que é a interface interativa com o usuário e simula o painel frontal de um instrumento com os botões de controle, botões de pressão e os displays gráficos; um diagrama em bloco com o código fonte para o programa, com a programação sendo feita graficamente por meio do desenho de linhas entre os pontos de conexão nos ícones selecionados na tela do computador; e a representação como ícone e conector que pode fornecer uma representação gráfica do instrumento virtual caso seja usado em outros diagramas em bloco.

A Figura 6.24(a) mostra o ícone selecionado para um instrumento virtual onde uma amostra analógica é obtida do canal de entrada especificado. O ícone foi selecionado a partir da palheta de entrada analógica. O “Dispositivo” é o número dado ao dispositivo na placa DAQ, o “Canal” é a fonte dos dados, a “Amostra” é uma conversão analógico-digital, e “Limite superior” e “Limite inferior” são as tensões limites esperadas para o sinal (o default é +10 V e -10 V, e a mudança destes valores varia automaticamente o ganho do amplificador na placa DAQ).

Se quisermos uma forma de onda de cada canal em uma sequência desejada, então o ícone mostrado na Figura 6.24(b) pode ser selecionado. Para cada canal de entrada é adquirido um conjunto de amostras durante um período, a uma taxa de amostragem especificada, e a forma de onda de saída é mostrada como uma variável analógica no tempo.

Conectando outros ícones ao, digamos, ícone anterior, um diagrama em bloco pode ser construído, o qual pode obter as entradas a partir de diversos canais analó-

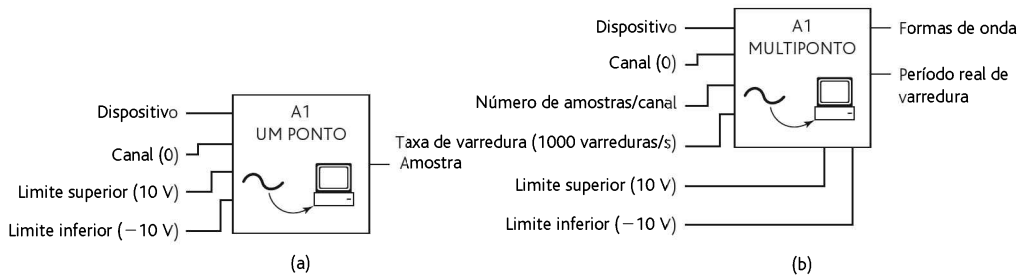


Figura 6.24 Ícone de entrada analógica: (a) entrada simples, (b) para a amostragem de mais de um canal.

gicos, amostrá-los em sequência e apresentar os resultados como uma sequência de gráficos. O tipo de display de painel frontal que podemos ter para um simples DAQ de amostras e display é mostrado na Figura 6.25. Os parâmetros podem ser alterados usando-se as setas para cima e para baixo e o resultado visualizado no display.

Os instrumentos virtuais têm uma grande vantagem sobre os tradicionais em que o fornecedor de um instrumento tradicional determina as características do instrumento e a interface, enquanto que no caso de um instrumento virtual isto pode ser definido pelo usuário e alterado rapidamente.

6.6.2 Data loggers

O termo *data logger* é associado a sistemas DAQ que podem ser usados sem a necessidade de estarem conectados a um computador. Uma vez definido o programa no computador, ele pode ser colocado em um cartão de memória que pode ser inse-

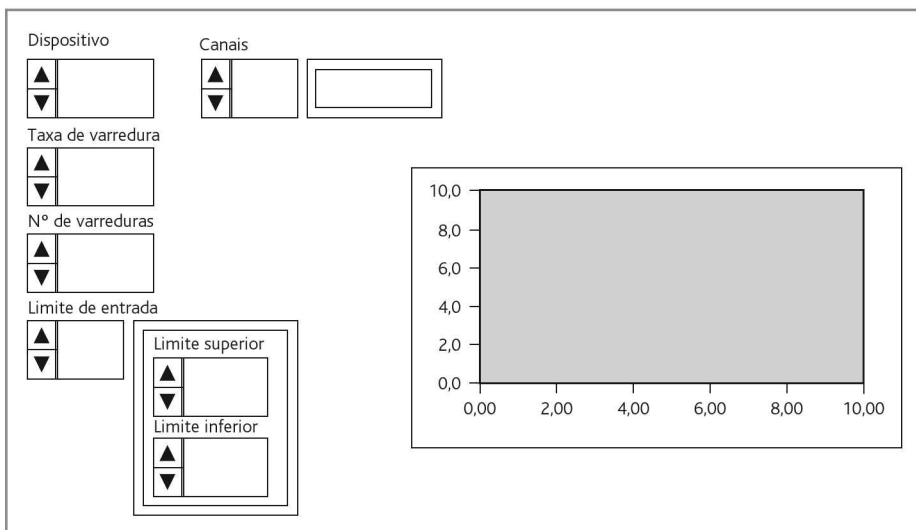


Figura 6.25 Painel frontal.

rido no *data logger*, ou então o programa pode ser transferido para ele a partir de um computador, possibilitando-lhe realizar as funções de um DAQ.

A Figura 6.26 mostra os elementos básicos de um *data logger*. Esta unidade pode monitorar as entradas de alguns sensores. As entradas dos sensores individuais, após um condicionamento de sinal adequado, são transferidas para um multiplexador. Este é usado para selecionar o sinal, que é então, após amplificação, transferido para o conversor analógico-digital. O sinal digital é então processado por um microprocessador. Este é capaz de realizar operações aritméticas simples, que podem ser a média de algumas medições. A saída do sistema pode ser mostrada em um medidor digital que indica a saída e o número do canal, pode ser registrada de forma permanente com uma impressora, pode ser armazenada em um disco flexível, ou pode ser transferida para um computador para análise.

Devido aos *data loggers* serem frequentemente usados com termopares, é comum a existência de entradas especiais para termopares com compensação de junção fria e linearização. O multiplexador pode ser comutado para cada sensor por vez, e assim a saída consiste de uma sequência de amostras. O escaneamento das entradas pode ser selecionado pela programação do microprocessador para comutar o multiplexador para amostrar apenas um canal, fazer uma varredura simples de todos os canais, uma varredura contínua de todos os canais, ou fazer uma varredura periódica de todos os canais a cada, digamos, 1, 5, 15, 30 ou 60 minutos.

Tipicamente um *data logger* pode operar de 20 a 100 entradas, embora alguns possam operar com um número consideravelmente maior, podendo ser 1.000. Ele pode ter um tempo de amostragem e conversão de 10 μ s e ser usado para realizar, por exemplo, 1.000 leituras por segundo. A precisão é tipicamente em torno de 0,01% da entrada de fundo de escala, e a linearidade é cerca de $\pm 0,005\%$ da entrada de fundo de escala. A interferência cruzada é tipicamente de 0,01% da entrada de fundo de escala para qualquer uma das entradas. O termo *cross-talk* é usado para descrever esta interferência que pode ocorrer quando um sensor é amostrado como resultado dos sinais de outros sensores.

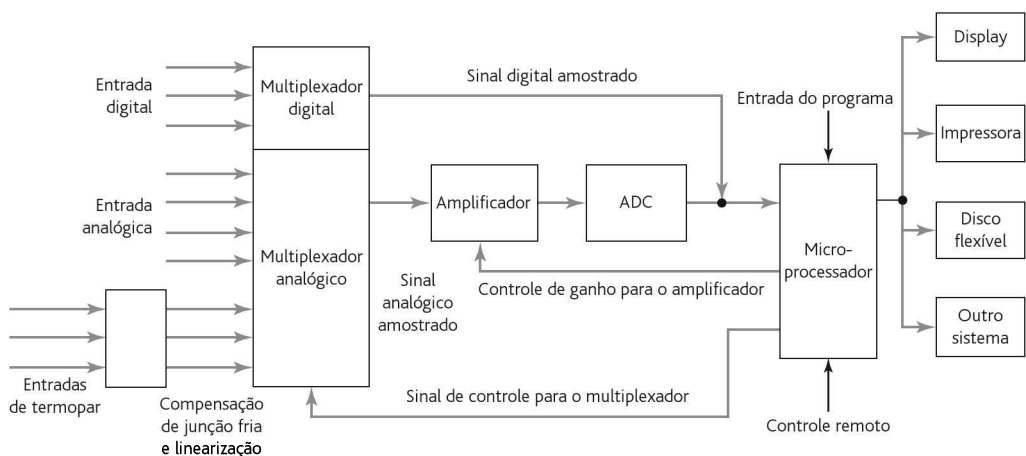


Figura 6.26 Sistema de um *data logger*.

6.7 Sistemas de medição

Os exemplos a seguir ilustram alguns dos aspectos envolvidos no projeto de sistemas de medição para aplicações particulares.

6.7.1 Célula de carga usada como elo de conexão no levantamento de carga

Uma célula de carga do tipo elo de conexão, da forma mostrada na Figura 6.27, tem quatro *strain gauges* fixados em sua superfície e pode ser inserida entre o cabo de levantamento da carga e a carga para fornecer uma medida da carga levantada. Dois dos *strain gauges* estão na direção longitudinal e dois, na transversal. Quando o elo de conexão é submetido a uma força de tração, os *strain gauges* axiais sofrerão força de tração e os transversais, de compressão. Suponha que temos o critério de projeto para a célula de carga com uma sensibilidade tal que uma saída de cerca de 30 mV é gerada quando o esforço aplicado ao elo de conexão for de 500 MPa. Consideraremos que os *strain gauges* podem ter fatores *gauge* de 2,0 e resistências de 100 Ω .

Quando uma carga F for aplicada ao elo de conexão, então, visto que o módulo de elasticidade E é esforço/deformação, sendo o esforço a força por unidade de área, o esforço no eixo longitudinal ε_l é F/AE e o esforço transversal ε_t é $-\nu F/AE$, onde A é a área da seção reta e ν é a razão de Poisson para o material do elo de conexão. As respostas dos *strain gauges* (veja Seção 2.3.2) para estes esforços são

$$\frac{\delta R_1}{R_1} = \frac{\delta R_4}{R_4} = G\varepsilon_l = \frac{GF}{AE}$$

$$\frac{\delta R_3}{R_3} = \frac{\delta R_2}{R_2} = G\varepsilon_t = -\frac{\nu GF}{AE}$$

A tensão de saída da ponte de Wheatstone (veja a Seção 3.5) é dada por

$$V_o = \frac{V_s R_1 R_4}{(R_1 + R_2)(R_3 + R_4)} \left(\frac{\delta R_1}{R_1} - \frac{\delta R_2}{R_2} - \frac{\delta R_3}{R_3} + \frac{\delta R_4}{R_4} \right)$$

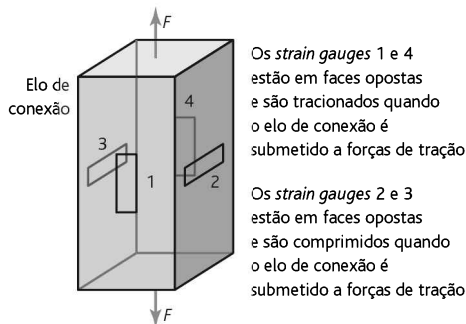


Figura 6.27 Célula de carga.

Com $R_1 = R_2 = R_3 = R_4 = R$, e com $\delta R_1 = \delta R_4$ e $\delta R_2 = \delta R_3$, então

$$V_o = \frac{V_s}{2R}(\delta R_1 - \delta R_2) = \frac{V_s GF}{2AE}(1 + \nu)$$

Vamos supor um elo de conexão de aço. Então as tabelas fornecem E como cerca de 210 GPa e ν como cerca de 0,30. Portanto, com um esforço ($=F/A$) de 500 MPa, temos, para *strain gauges* com um fator *gauge* de 2,0,

$$V_o = 3,09 \times 10^{-3} V_s$$

A tensão na ponte, com uma tensão de alimentação de 10 V, seria 30,9 mV. Nenhuma amplificação seria necessária se fosse apenas esse valor de carga; entretanto, esse é o valor máximo, e como queremos determinar cargas abaixo deste valor, podemos usar um amplificador diferencial. A saída pode ser mostrada em um voltímetro de alta resistência, para evitar problemas de efeito de carga. Um voltímetro digital pode ser adequado neste caso.

6.7.2 Sistema de alarme de temperatura

Um sistema de medição é necessário para acionar um alarme quando a temperatura de um líquido subir acima de 40°C. O líquido está normalmente em 30°C. A saída do sistema tem que ser um sinal de 1 V para ativar o alarme.

Visto que a saída é um sinal elétrico e uma velocidade adequada de resposta é provavelmente desejada, uma possibilidade óbvia é um elemento de resistência elétrica. Para gerar uma saída de tensão, o elemento de resistência deve ser usado com uma ponte de Wheatstone. A tensão de saída será provavelmente menor do que 1 V para uma variação de 30 para 40°C, mas um amplificador diferencial pode ser usado para se obter a tensão desejada. Um comparador pode então ser usado para comparar o valor com o valor ajustado para acionar o alarme.

Suponha que o elemento níquel seja usado. O níquel tem um coeficiente de temperatura de resistência de 0,0067/K. Portanto, se o elemento de resistência for de 100 Ω em 0°C, então sua resistência em 30°C será de

$$R_{30} = R_0(1 + \alpha t) = 100(1 + 0,0067 \times 30) = 120,1 \Omega$$

e em 40°C de

$$R_{40} = 100(1 + 0,0067 \times 40) = 126,8 \Omega$$

Portanto, há uma variação na resistência de 6,7 Ω . Se este elemento está em um ramo da ponte de Wheatstone que está equilibrada em 30°C, então a tensão de saída, V_o , é dada por (veja a Seção 3.5)

$$\delta V_o = \frac{V_s \delta R_1}{R_1 + R_2}$$

Com a ponte equilibrada em 30°C e, digamos, todos os ramos tendo o mesmo valor e uma tensão de alimentação de 4 V, então

$$\delta V_o = \frac{4 \times 6,7}{126,8 + 120,1} = 0,109 \text{ V}$$

Para amplificar este valor para 1 V, podemos usar um amplificador diferencial (veja a Seção 3.2.5):

$$V_o = \frac{R_2}{R_1}(V_2 - V_1)$$

$$1 = \frac{R_2}{R_1} \times 0,109$$

Portanto, $R_2/R_1 = 9,17$ e assim, se usarmos uma resistência de entrada de 1 k Ω , a resistência de realimentação deve ser 9,17 k Ω .

6.7.3 Posição angular de uma polia

Um potenciômetro é usado para monitorar a posição angular de uma polia. Considere os itens que podem ser necessários para que um circuito forneça uma saída de 10 mV por grau para um gravador se o potenciômetro tem uma rotação angular de fundo de escala de 320°.

Quando a tensão de alimentação V_s é conectada no potenciômetro, precisamos salvaguardá-lo de possíveis correntes altas, e por isto a resistência R_s é colocada em série com o potenciômetro R_p . A queda de tensão total no potenciômetro é então $V_s R_p / (R_s + R_p)$. Para um ângulo θ , sendo a deflexão de fundo de escala do potenciômetro θ_F , obtemos uma saída do potenciômetro de

$$V_\theta = \frac{\theta}{\theta_F} \frac{V_s R_p}{R_s + R_p}$$

Vamos considerar um potenciômetro com uma resistência de 4 k Ω e R_s de 2 k Ω . Então, para 1 mV por grau temos

$$0,01 = \frac{1}{320} \frac{4V_s}{4 + 2}$$

Portanto, precisamos de uma fonte de alimentação de 4,8 V. Para evitar o efeito de carga no potenciômetro provocado pela resistência de entrada do registrador, pode-se usar um circuito seguidor de tensão. Assim, o circuito tem a forma mostrada na Figura 6.28.

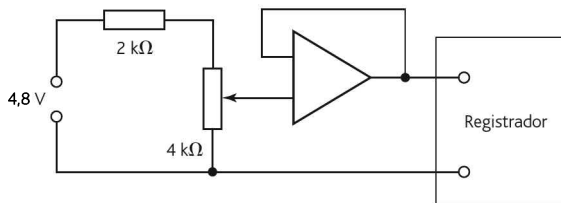


Figura 6.28 Monitor da polia.

6.7.4 Medição de temperatura com saída binária

Considere a necessidade de um sistema de medição de temperatura em uma faixa de 0 a 100°C, com uma saída binária de 8 bits e com uma variação de 1 bit correspondendo a uma variação de 1°C na temperatura. A saída é enviada a um microprocessador como parte do sistema de controle da temperatura.

Como é necessário um sensor de temperatura linear, o termotransistor LM35 pode ser usado (veja a Seção 2.9.4). Este dispositivo fornece uma saída de 10 mV/°C quando alimentado por uma tensão de 5 V. Se aplicarmos a saída de um LM35 a um conversor analógico-digital de 8 bits, então uma saída digital pode ser obtida. Precisamos que a resolução do ADC seja de 10 mV, de modo que cada degrau de 10 mV produza uma variação de 1 bit na saída. Suponha que usemos um ADC de aproximações sucessivas, como por exemplo o ADC0801; assim, será necessário uma tensão de referência, de entrada que, quando dividida por $2^8 = 256$ bits, forneça 10 mV por bit. Portanto, uma tensão de referência de 2,56 V é necessária. Para se obter este valor de tensão de referência, que para o ADC0801 tem que ser $V_{ref}/2$, é necessário uma tensão de entrada de precisão de 1,28 V. Esta tensão pode ser obtida usando-se um circuito com potenciômetro alimentado por 5 V e com um seguidor de tensão para evitar problemas de efeito de carga. Como a tensão tem que permanecer estável em 1,28 V, ainda que a tensão de alimentação de 5 V flutue em torno deste valor, podemos usar um regulador de tensão, por exemplo, o regulador de tensão ZN458/B de 2,45 V. Portanto, o circuito pode ser como o que é mostrado na Figura 6.29.

6.8 Teste e calibração

O teste de um sistema de medição pode ser dividido em três estágios:

1. Teste de pré-instalação

Este é o teste de cada instrumento para a correta calibração e operação antes de serem instalados.

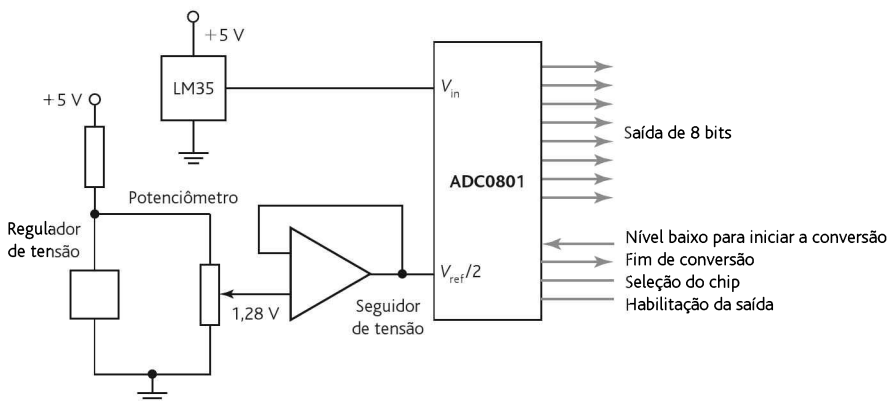


Figura 6.29 Sensor de temperatura.

2. *Teste da tubulação e do cabeamento*

No caso de linhas pneumáticas, este teste envolve, antes da conexão dos instrumentos, a limpeza dos tubos soprando-se ar seco por eles e um teste de pressão para garantir que não há vazamento na tubulação. Nos processos com tubulação, esta deve ser limpa com o uso de um fluido e testada antes da conexão dos instrumentos. No caso de instrumentos com cabos, deve-se verificar a continuidade e a resistência de isolamento antes de se conectar quaisquer instrumentos.

3. *Pré-teste do sistema*

Envolve o teste da instalação completa, em que todos os instrumentos operam integralmente quando interconectados e todos os painéis na sala de controle ou displays funcionam.

6.8.1 Calibração

A **calibração** consiste na comparação da saída de um sistema de medição, e seus subsistemas, com padrões de precisões conhecidas. Os padrões podem ser outros instrumentos dedicados especialmente para a função de calibração ou alguns meios de definição de valores padrão. Em muitas companhias, alguns instrumentos e itens, como resistores padrão e células, são mantidos no departamento de padrões da companhia e usados apenas para fins de calibração. A relação entre a calibração de um instrumento de uso diário e padrões nacionais é:

1. Padrões nacionais são usados para calibrar padrões de laboratórios de calibração.
2. Os padrões dos laboratórios de calibração são usados para calibrar padrões dos fabricantes de instrumentos.
3. Instrumentos calibrados são destinados a servirem de padrões dentro de companhias.
4. Os padrões dentro das companhias são usados para calibrar os instrumentos dos processos

Existe uma rede de rastreabilidade desde o instrumento usado em um processo até os padrões nacionais. Os exemplos a seguir são procedimentos de calibração que podem ser usados na companhia:

1. *Voltímetros*

Estes podem ser verificados na comparação com voltímetros padrão ou células padrão que fornecem fems padrão.

2. *Amperímetros*

Podem ser verificados na comparação com amperímetros padrão.

3. *Fator gauge de strain gauges*

Pode ser verificado tomando-se uma amostra de medidas a partir de um grupo e aplicando-se esforços conhecidos com eles montados em uma peça de teste. A resistência que varia pode ser medida para então se calcular o fator *gauge*.

4. *Circuitos em ponte de Wheatstone*

A saída de uma ponte de Wheatstone pode ser verificada quando uma resistência padrão é introduzida em um dos ramos.



Figura 6.30 Calibração de peso morto para medidores de pressão.

5. Células de carga

No caso de células de carga de baixa capacidade, cargas de peso morto usando pesos padrão podem ser usadas.

6. Sensores de pressão

Sensores de pressão podem ser calibrados usando-se um testador de peso morto (Figura 6.30). As pressões calibradas são geradas acrescentando-se pesos padrão W sobre uma bandeja do pistão. Após os pesos serem colocados na bandeja, um êmbolo de parafuso é forçado em um óleo hidráulico em uma câmara para elevar o pistão que recebeu os pesos. A pressão de calibração é então W/A , onde A é a área da seção reta do pistão. Alternativamente, o testador de peso morto pode ser usado para calibrar a medida de pressão, e esta medida pode ser usada para calibração de outros medidores.

7. Sensores de temperatura

Estes podem ser calibrados através da imersão em um metal puro derretido ou água. A temperatura da substância é então reduzida lentamente, sendo registrada a variação temperatura-tempo. Quando a substância muda do estado líquido para o sólido, a temperatura permanece constante. Este valor pode ser obtido de tabelas, e assim uma temperatura de referência de precisão é obtida. Alternativamente, a temperatura na qual um líquido entra em ebulição pode ser usada. Entretanto, o ponto de ebulição depende da pressão atmosférica, e correções têm que ser aplicadas se a mesma diferir da pressão atmosférica padrão. Alternativamente, as leituras na companhia de um sistema de medição podem ser comparadas com as dadas por um termômetro padrão.

Resumo

Um ponto geral a ser levado em consideração quando se interliga qualquer sistema de medição é o **efeito de carga**, ou seja, o efeito da conexão de uma carga nos terminais de saída de qualquer elemento de um sistema de medição.

Os **indicadores** fornecem uma indicação visual instantânea da variável medida, enquanto que um **registrador** registra o sinal de saída durante um tempo e fornece automaticamente um registro permanente.

O termo **aquisição de dados** (DAQ) tende a ser usado frequentemente em sistemas nos quais as entradas dos sensores são convertidos no formato digital para o processamento, análise e apresentação por meio de um computador. O termo *data logger* é usado para sistemas DAQ que são capazes de serem usados sem a conexão com um computador.

Os **instrumentos virtuais** são instrumentos gerados por software; eles imitam os instrumentos reais em aparência e operação.

Os testes na instalação de um sistema de medição se enquadram em três estágios: teste de pré-instalação, teste da tubulação e do cabeamento e pré-teste do sistema.

A **calibração** consiste em comparar a saída de um sistema de medição e os seus subsistemas com um padrão de precisão conhecida.

Problemas

- 6.1 Explique o significado dos seguintes termos que são especificações de sistemas de displays:
 - a. Registrador: banda morta de $\pm 0,2\%$ do span.
 - b. O disco rígido tem dois discos com quatro cabeçotes de leitura/gravação, um para cada superfície dos discos. Cada superfície tem 614 trilhas e cada trilha, 32 setores.
 - c. *Data logger*: número de entradas igual a 100, *cross-talk* em qualquer uma das entradas igual a 0,01% da entrada de fundo de escala.
 - d. Osciloscópio de duplo feixe: deflexão vertical com dois canais idênticos, largura de banda CC até 15 MHz, fator de deflexão de 10 mV/div a 20 V/div em 11 escalas, base de tempo de 0,5 $\mu\text{s}/\text{div}$ a 0,5 s/div em 19 escalas.
- 6.2 Explique os problemas do efeito de carga quando um sistema de medida é estruturado em sensor, condicionador de sinal e display.
- 6.3 Sugira uma unidade de apresentação de dados que poderia ser usada para:
 - a. Registro permanente da saída a partir de um termopar.
 - b. Mostrar a pressão do óleo em um sistema.
 - c. Gravar a saída digital de um microprocessador.
 - d. Mostrar as tensões transientes resultantes do monitoramento das cargas em uma aeronave durante uma turbulência simulada.
- 6.4 Uma célula de carga cilíndrica, da forma mostrada na Figura 2.32, tem quatro *strain gauges* fixados em sua superfície. Dois deles estão na direção circunferencial e dois, na longitudinal. Quando o cilindro é submetido a uma carga de compressão, os *strain gauges* axiais recebem esforços de compressão enquanto os circunferenciais são tensionados. Se o material do cilindro tem uma área de seção reta A e um módulo de elasticidade E , então a força F que age no cilindro resulta em uma ação de esforço nas medidas axiais de $-F/AE$ e nas medidas de circunferência de $+\nu F/AE$, onde ν é a relação de Poisson para o material. Projete um sistema de medida completo, usando células de carga, que poderia ser usado para monitorar a massa de água em um tanque. O tanque tem uma massa de 20 kg e a água, quando

está no nível desejado, 40 kg. A massa deve ser monitorada com uma precisão de $\pm 0,5$ kg. Os *strains gauges* têm um fator *gauge* de 2,1 e todos têm a mesma resistência, 120 Ω . Para todos os outros itens, especifique o que o seu projeto necessita. Se você usar aço doce para o material da célula, então o módulo de tração pode ser considerado como 210 GPa e a razão de Poisson como 0,30.

- 6.5 Projete um sistema de medida completo envolvendo o uso de um termopar para determinar a temperatura da água em um ebulidor e fornecer uma indicação visual de um medidor. A temperatura será na faixa de 0 a 100°C e o sistema necessita de uma precisão de $\pm 1\%$ da leitura de fundo de escala. Especifique o material a ser usado para o termopar e todos os outros itens necessários. Em defesa de seu projeto considere os problemas da junção fria e da não linearidade. Provavelmente você precisará consultar tabelas de termopares. Os dados a seguir foram obtidos de tais tabelas, estando a junção fria em 0°C, e podem ser usados como um guia:

Materiais	FEM em mV a				
	20°C	40°C	60°C	80°C	100°C
Cobre-constantan	0,789	1,611	2,467	3,357	4,277
Cromel-constantan	1,192	2,419	3,683	4,983	6,317
Ferro-constantan	1,019	2,058	3,115	4,186	5,268
Cromel-alumel	0,798	1,611	2,436	3,266	4,095
Platina-10%, Rh, Pt	0,113	0,235	0,365	0,502	0,645

- 6.6 Projete um sistema de medida que possa ser usado para monitorar as temperaturas, da ordem de 100°C, em diversos pontos espalhados em uma planta e apresente o resultado em um painel de controle.
- 6.7 Um projeto sugerido para a medida do nível de um líquido em um recipiente envolve uma boia que em seu movimento vertical dobra uma haste em balanço. O grau em que a haste é dobrada representa a medida do nível do líquido. Quando uma força F é aplicada na extremidade livre da haste de comprimento L , o esforço em sua superfície a uma distância x da extremidade fixa é dada por

$$\text{esforço} = \frac{6(L - x)}{wt^2E}$$

onde m é a largura da haste, t é a sua espessura e E é o módulo da elasticidade do material. Os *strain gauges* devem ser usados para monitorar a dobra da haste, com dois deles fixados longitudinalmente na superfície superior e dois longitudinalmente na superfície inferior. Os *strain gauges* são então incorporados nos quatro ramos de uma ponte de Wheatstone, e a tensão de saída, após uma possível amplificação, é considerada a medida do nível do líquido. Determine as especificações necessárias para os componentes deste sistema se a saída apresenta uma variação de 10 mV para uma variação de 10 cm no nível.

- 6.8 Projete um sistema de medida de pressão estática baseado em um sensor que envolve o uso de um diafragma de 40 mm que tem uma diferença de pressão máxima de 500 MPa. Para um diafragma onde a deflexão central y é muito menor do que a espessura t do diafragma,

$$y \approx \frac{3r^2 P(1 - \nu^2)}{16Et^3}$$

em que r é o raio do diafragma, P é a diferença de pressão, E é o módulo da elasticidade e ν é a razão de Poisson. Explique como a deflexão y será convertida em um sinal que pode ser mostrado em um medidor.

- 6.9 Sugira os elementos que podem ser considerados para um sistema de medidas usado para:
- Monitorar a pressão em uma linha de pressão de ar e apresentar o resultado em um mostrador, sem que uma grande precisão seja necessária.
 - Monitorar continuamente e registrar a temperatura de uma sala com uma precisão de $\pm 1^\circ\text{C}$.
 - Monitorar o peso de caminhões passando sobre uma balança de pesagem.
 - Monitorar a velocidade angular de rotação de um eixo.

7

Sistemas de Acionamento Pneumático e Hidráulico

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Interpretar os desenhos de sistemas hidráulicos/pneumáticos e projetar sistemas simples para controle sequencial usando válvulas de controle direcional e cilindros.
- Explicar os princípios das válvulas de controle de processos, suas características e dimensões.

7.1 Sistemas de acionamento

Os **sistemas de acionamento** são elementos dos sistemas de controle responsáveis pela transformação da saída de um microprocessador ou sistema de controle em uma ação de controle para uma máquina ou dispositivo. Portanto, por exemplo, podemos ter uma saída elétrica a partir de um controlador que tem que ser transformada em um movimento linear para deslocar uma carga. Outro exemplo pode ser a transformação de uma saída elétrica de um controlador em uma ação que controla a quantidade de líquido que passa em uma tubulação.

Neste capítulo estudaremos os sistemas de potência baseado em fluidos denominados sistemas de acionamento pneumático e hidráulico. O termo **pneumático** é usado em aplicações de ar comprimido, e o termo **hidráulico** em aplicações de líquido, tipicamente óleo. No Capítulo 8 estudaremos os sistemas mecânicos de acionamento, e no Capítulo 9, os sistemas elétricos de acionamento.

7.2 Sistemas hidráulicos e pneumáticos

Os sinais pneumáticos são frequentemente usados para controle dos elementos finais de controle, mesmo quando o sistema de controle for elétrico. Isto porque estes sinais podem ser usados para acionar válvulas grandes e outros dispositivos de

controle de potência alta e, desta forma, mover cargas significativas. Entretanto, a principal desvantagem dos sistemas pneumáticos é a compressibilidade do ar. Os sistemas hidráulicos podem ser usados em dispositivos de controle de potências ainda maiores, mas são mais caros do que os sistemas pneumáticos e existem perigos associados com vazamentos de óleo que não existem nos vazamentos de ar.

A pressão atmosférica varia com o local e com o tempo, mas é geralmente considerada como sendo de 10^5 Pa, que equivale a 1 bar, em sistemas pneumáticos.

7.2.1 Sistemas hidráulicos

Em um sistema hidráulico, o óleo pressurizado é produzido por uma bomba acionada por um motor elétrico. A bomba bombeia óleo a partir de um coletor de óleo através de uma válvula de retenção e um acumulador para o sistema, a partir do qual ele retorna para o coletor de óleo. A Figura 7.1(a) ilustra esta configuração. Uma válvula de alívio de pressão é incluída para alívio da pressão caso ela aumente acima de um valor seguro, a válvula de retenção é para evitar que o óleo retorne pela bomba, e o acumulador é para atenuar flutuações na pressão de saída do óleo. Essencialmente, o acumulador é apenas um recipiente no qual o óleo é mantido sob pressão por meio de uma força externa. A Figura 7.1(b) mostra o tipo geralmente usado, que é o de um gás pressurizado no interior de uma bexiga que se encontra dentro de uma câmara que contém um fluido hidráulico; um tipo mais antigo envolve um pistão acionado por mola. Se a pressão do óleo aumenta, então a bexiga é contraída, e o volume que o óleo ocupa aumenta, reduzindo assim a pressão. Se a pressão do óleo cair, a bexiga se expande para reduzir o volume ocupado pelo óleo e assim aumentar a sua pressão.

As bombas hidráulicas normalmente usadas são do tipo engrenagens, palhetas e pistão. A **bomba de engrenagens** consiste de duas rodas de engrenagens acopladas que giram em direções opostas (Figura 7.2(a)). O fluido é forçado através da bomba à medida que ele fica confinado entre os dentes da engrenagem que gira e a carcaça, transferindo assim o fluido da porta de entrada e descarregando-o na porta

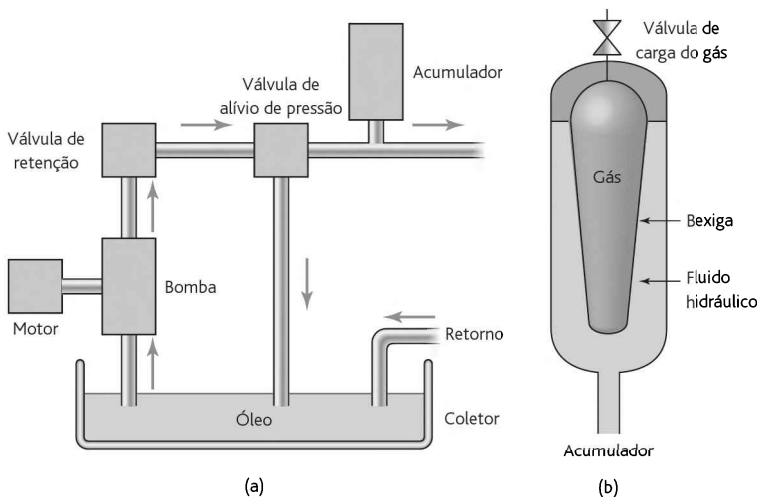


Figura 7.1 (a) Fonte de alimentação hidráulica, (b) acumulador.

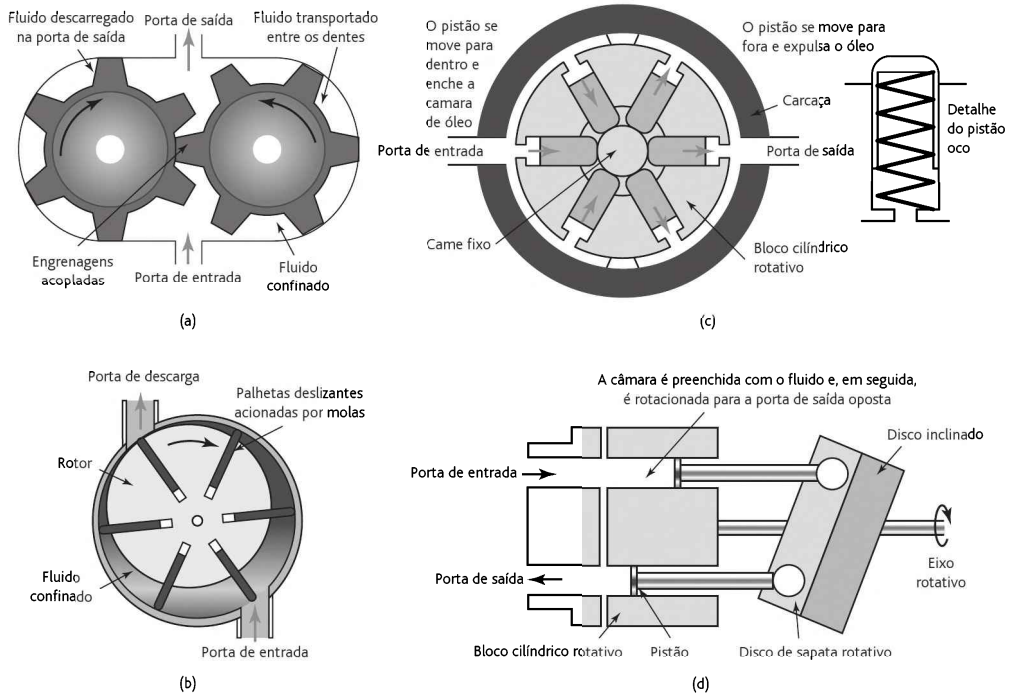


Figura 7.2 (a) Bomba de engrenagens, (b) bomba de palhetas, (c) bomba de pistão radial, (d) bomba de pistão axial com disco inclinado.

de saída. Estas bombas são bastante usadas, sendo de baixo custo e robustas. Elas geralmente operam em pressões abaixo de 15 MPa e a 2400 rotações por minuto. A vazão máxima é aproximadamente $0,5 \text{ m}^3/\text{min}$. Entretanto, ocorrem vazamentos entre os dentes e a carcaça e entre o travamento dos dentes, limitando a sua eficiência. A **bomba de palhetas** tem palhetas deslizantes com mola que são fixadas nas fendas de um rotor acionado (Figura 7.2(b)). À medida que o rotor gira, as palhetas seguem os contornos da carcaça da bomba. Isto faz com que o fluido preso entre as palhetas e a carcaça seja transportado da porta de entrada para a de saída. Os vazamentos são menores do que na bomba de engrenagens. As **bombas de pistão** usadas em sistemas hidráulicos têm formas variadas. No caso da **bomba de pistão radial** (Figura 7.2(c)), um bloco cilíndrico gira em torno de um came estacionário, provocando o movimento para dentro e para fora de êmbolos ocios que retornam por ação de mola. O resultado é que o fluido é puxado da porta de entrada, transportado com a rotação do mecanismo e ejetado na porta de saída. A **bomba de pistão axial** (Figura 7.2(d)) tem pistões que se movem axialmente em vez de radialmente. Os pistões são organizados axialmente em um bloco cilíndrico rotativo que gira por meio do contato com o disco inclinado. Este disco forma um ângulo com o eixo propulsor e, portanto, à medida que o eixo gira, ele move os pistões de modo que o fluido seja sugado para dentro quando um pistão está em oposição à porta de entrada e seja expelido quando está em oposição à porta de saída. As bombas de pistão têm uma alta eficiência e podem ser usadas em pressões hidráulicas maiores do que as bombas de engrenagens ou palhetas.

7.2.2 Sistemas pneumáticos

Em uma fonte de alimentação pneumática (Figura 7.3), um motor elétrico aciona um compressor de ar. O ar que entra no compressor é geralmente filtrado, e o nível de ruído é reduzido quando o ar passa por um silenciador. Uma válvula de alívio de pressão fornece proteção caso a pressão no sistema suba acima de um nível seguro. Visto que o compressor de ar faz aumentar a temperatura do ar, geralmente há um sistema de ventilação, e para remover contaminação e água do ar, um filtro detém a passagem da água. Um receptor de ar aumenta o volume de ar no sistema e atenua qualquer flutuação rápida de pressão.

Normalmente, nos compressores de ar, sucessivos volumes de ar são isolados e então comprimidos. A Figura 7.4(a) mostra um tipo básico de um compressor de ação simples, estágio simples, vertical e de alternância. No movimento de admissão

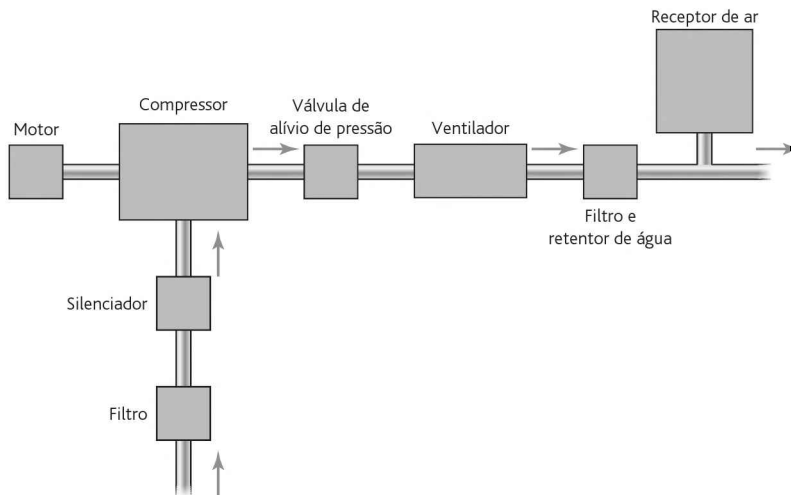


Figura 7.3 Fonte de alimentação pneumática.

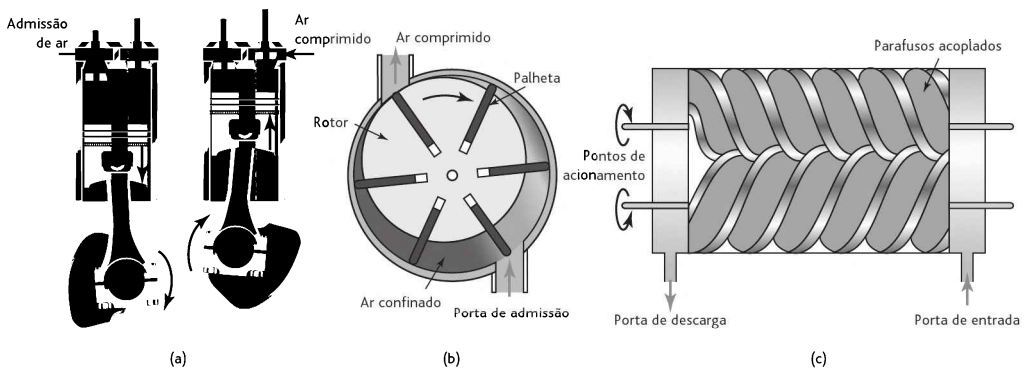


Figura 7.4 (a) Compressor de ação simples, estágio simples, vertical e de alternância, (b) compressor de palhetas rotativas, (c) compressor de parafuso.

do ar, a descida do pistão faz com que o ar seja aspirado para dentro da câmara através da válvula de admissão com mola, e quando o pistão inicia o movimento de subida novamente, o ar preso força a válvula de admissão a fechar, sendo então comprimido. Quando a pressão do ar é suficientemente elevada, a válvula de saída com mola se abre e o ar confinado flui para dentro do sistema de ar comprimido. Após o pistão ter alcançado o ponto morto superior, ele inicia então a descida e o ciclo se repete. Este compressor é denominado de **ação simples** porque um pulso de ar é produzido por movimento do pistão; os compressores de **ação dupla** são projetados para produzir pulsos de ar na subida e na descida do pistão. Ele também é denominado de **estágio simples** porque o compressor vai diretamente da pressão atmosférica para a pressão desejada em uma única operação. Para a produção de ar comprimido em mais do que alguns bars, são geralmente usados dois ou mais estágios. Normalmente, dois estágios são usados para pressões em torno de 10 a 15 bars, e mais estágios para pressões maiores. Portanto, com um compressor de dois estágios podemos ter o primeiro estágio tomando o ar na pressão atmosférica e o comprimindo até, digamos, 2 bar e, em seguida, o segundo estágio comprimindo este ar até, digamos, 7 bar. Compressores de pistão de alternância podem ser usados como um compressor de estágio simples para produzir pressões de ar de até 12 bar, e como um compressor de múltiplo estágio para até 140 bar. Tipicamente, a vazão de ar fornecida tende a variar de cerca de $0,02 \text{ m}^3/\text{min}$ a cerca de $600 \text{ m}^3/\text{min}$ de ar livre entregue; o termo “ar livre” é usado para o ar na pressão atmosférica normal. Um outro tipo de compressor é o de **palhetas rotativas**. Este tipo tem um rotor montado excentricamente em uma câmara cilíndrica (Figura 7.4(b)). O rotor tem palhetas fixadas em fendas radiais e livres para deslizar com a rotação, o que faz com que as palhetas sejam acionadas em movimento centrífugo contra as paredes do cilindro. Conforme o rotor gira, o ar é confinado em compartimentos formados pelas palhetas e, à medida que o rotor gira, estes compartimentos se tornam menores e o ar é comprimido. Os compartimentos com o ar comprimido são descarregados na porta de descarga. Os compressores de palhetas rotativas de estágio simples podem tipicamente serem usados para pressões de até cerca de 800 kPa com vazão da ordem de $0,3 \text{ m}^3/\text{min}$ a $30 \text{ m}^3/\text{min}$ fornecendo ar livre. Um outro tipo de compressor é o de **parafuso rotativo** (Figura 7.4(c)). Este tem dois parafusos acoplados que giram em sentidos opostos. À medida que os parafusos giram, o ar é puxado para dentro do compartimento através da porta de admissão e entra no espaço entre as roscas. Este ar confinado é movido ao longo do comprimento dos parafusos e comprimido conforme o espaço se torna progressivamente menor, emergindo pela porta de descarga. Tipicamente, um compressor de parafuso rotativo com um estágio simples pode ser usado para pressões de até cerca de 1.000 kPa com vazões entre 1,4 e $60 \text{ m}^3/\text{min}$ de ar livre entregue.

7.2.3 Válvulas

As válvulas são usadas em sistemas hidráulicos e pneumáticos para guiar e regular a vazão de um fluido. Basicamente, existem apenas dois tipos de válvulas: a de **posições finitas** e a de **posições infinitas**. As válvulas de posições finitas são aquelas em que a ação é apenas para permitir ou bloquear a passagem do fluido e, assim, podem ser usadas para ligar ou desligar atuadores. Elas podem ser usadas para controle direcional comutando o fluxo de uma via para outra e, desta forma, de

um atuador para outro. As válvulas de posições infinitas são capazes de controlar a vazão em qualquer valor entre a posição totalmente aberta e a totalmente fechada, sendo usadas para controlar as forças nos atuadores ou a vazão de um fluido em uma situação de controle de processo.

7.3 Válvulas de controle direcional

Os sistemas hidráulicos e pneumáticos usam válvulas de controle direcional para conduzir a vazão de um fluido através de um sistema. Elas não são usadas para variar a vazão do fluido, mas apenas abrem ou fecham completamente, ou seja, são dispositivos *on/off*. Estas válvulas *on/off* são muito usadas no desenvolvimento de sistemas de controle sequencial (assunto estudado posteriormente neste capítulo). Elas podem ser ativadas para comutar a direção da vazão do fluido através de sinais mecânico, elétrico ou de pressão de fluido.

Um tipo comum de válvula de controle direcional é a **válvula tipo carretel**. Um carretel se move horizontalmente dentro do corpo da válvula para controlar o fluxo. A Figura 7.5 mostra um tipo particular. Em (a) a fonte de ar é conectada na porta 1 e a porta 3 é fechada. Portanto, o dispositivo conectado na porta 2 pode ser pressurizado. Quando o carretel é movido para a esquerda (Figura 7.5(b)), a fonte de ar é desligada e a porta 2 é conectada na porta 3. Esta última é um ponto de saída de ar para a atmosfera e, desta forma, a pressão do ar do sistema que está na porta 2 é descarregada. Portanto, o movimento do carretel permitiu que o ar primeiro entrasse no sistema para depois sair. As **válvulas tipo carretel rotativo** têm um carretel girante que, ao girar, abre e fecha portas de forma similar.

Um outro tipo comum de válvula de controle direcional é a **válvula de gatilho**. A Figura 7.6 um tipo desta válvula. Ela está normalmente na condição fechada, não havendo conexão entre a porta 1, na qual é conectada a fonte de alimentação, e a porta 2, na qual é conectado o sistema. Nas válvulas de gatilho são usados esferas, discos ou cones na sede da válvula para controlar o fluxo. A figura mostra o uso de uma esfera. Quando o botão é pressionado, a bola é empurrada para fora da sua sede e ocorre um fluxo como resultado da porta 1 estar conectada com a 2. Quando o botão é liberado, a mola força a esfera a voltar para sua sede, desligando o fluxo.

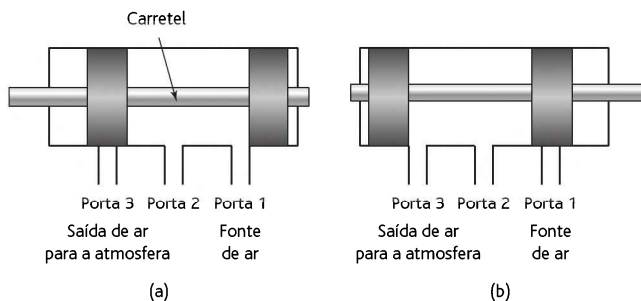


Figura 7.5 Válvula tipo carretel.

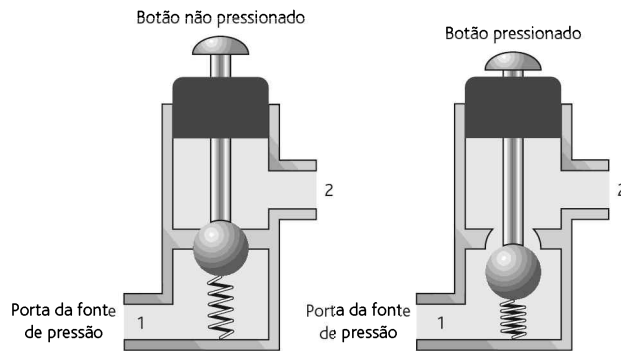


Figura 7.6 Válvula de gatilho.

7.3.1 Símbolos de válvulas

O símbolo usado para uma válvula de controle consiste de um quadrado para cada uma de suas posições de chaveamento. Portanto, para a válvula de gatilho mostrada na Figura 7.6, há duas posições: uma com o botão liberado e outra com ele pressionado. Assim, uma válvula de duas posições terá dois quadrados e uma de três posições, três quadrados. As linhas com seta (Figura 7.7(a)) são usadas para indicar o sentido do fluxo em cada uma das posições, com as linhas bloqueadas indicando linha de fluxo fechada (Figura 7.7(b)). A posição inicial da válvula tem as conexões às portas mostradas (Figura 7.7(c)); nesta figura, a válvula tem quatro portas. As portas são identificadas por números ou letras de acordo com suas funções. As portas são identificadas por 1 (ou P) no caso de fonte de alimentação, 3 (ou T) no caso de porta de retorno hidráulico, 3 ou 5 (ou R ou S) para portas de exaustão pneumática e 2 ou 5 (ou B ou A) para portas de saída.

A Figura 7.8(a) mostra exemplos de alguns dos símbolos que são usados para indicar as formas com que as válvulas podem ser atuadas. Mais de um destes símbolos podem ser usados com o símbolo de válvula. Como uma ilustração, a Figura 7.8 (b) mostra o símbolo para a válvula de gatilho de duas portas e duas posições da Figura 7.6. Note que uma válvula de duas portas e duas posições seria descrita como uma válvula 2/2, sendo que o primeiro número indica o número de portas e o segundo, o número de posições. A atuação da válvula é por botão e mola.

Como ilustração adicional, a Figura 7.9 mostra uma válvula de gatilho operada por solenoide e o seu símbolo. A válvula é acionada por uma corrente que passa em um solenoide e retorna à sua posição original por ação de mola.

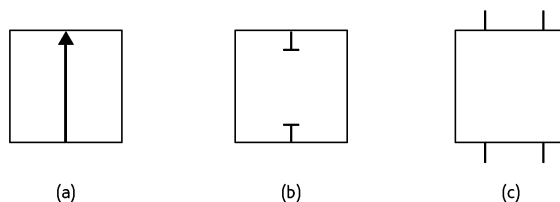


Figura 7.7 (a) Via de fluxo, (b) fluxo desligado, (c) conexões iniciais.

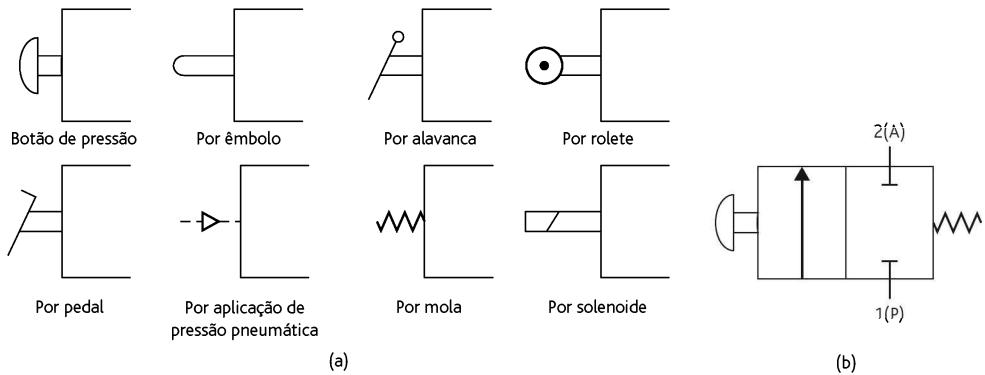


Figura 7.8 Símbolos de atuações de válvulas.

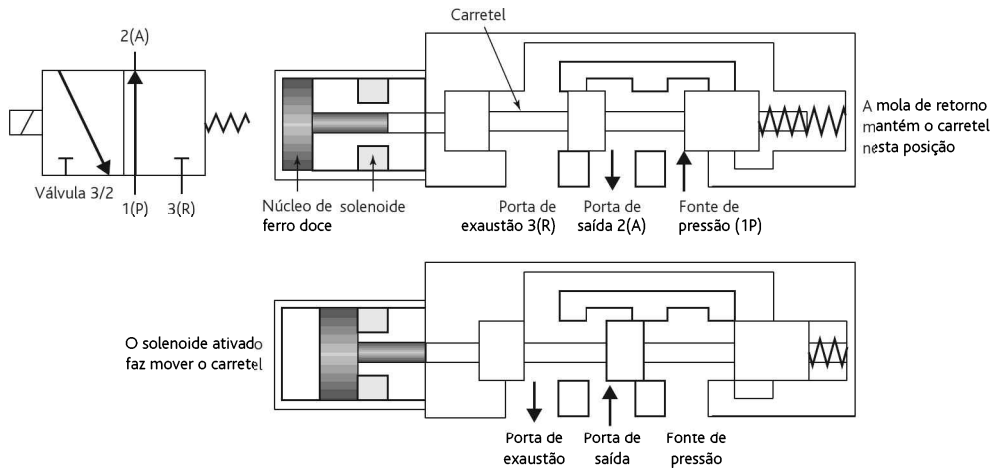


Figura 7.9 Válvula de um solenoide.

A Figura 7.10 mostra o símbolo de uma válvula 4/2. As conexões são mostradas para o estado inicial, ou seja, 1(P) é conectado em 4(A) e 3(R) em 2(B). Quando o solenoide é ativado, ele terá o estado indicado pelos símbolos usados no quadrado no qual está fixado, ou seja, agora temos 1(P) conectado em 2(B) e 4(A) conectado a 3(R). Quando a corrente no solenoide cessa, a mola empurra a válvula de volta à sua posição inicial. O movimento da mola proporciona o estado indicado pelos símbolos usados no quadrado no qual ela está fixada.

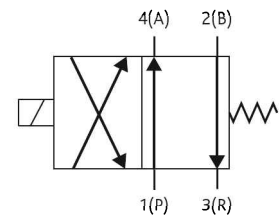


Figura 7.10 Uma válvula 4/2.

A Figura 7.11 mostra um exemplo simples de uma aplicação de válvulas em um sistema pneumático de levantamento. São usadas válvulas 2/2 com botão de pressão. Quando o botão PARA CIMA é pressionado, a carga é levantada. Quando o botão PARA BAIXO é pressionado, a carga é abaixada. Note que nos sistemas pneumáticos uma seta não preenchida é usada para indicar uma saída de ar para a atmosfera.

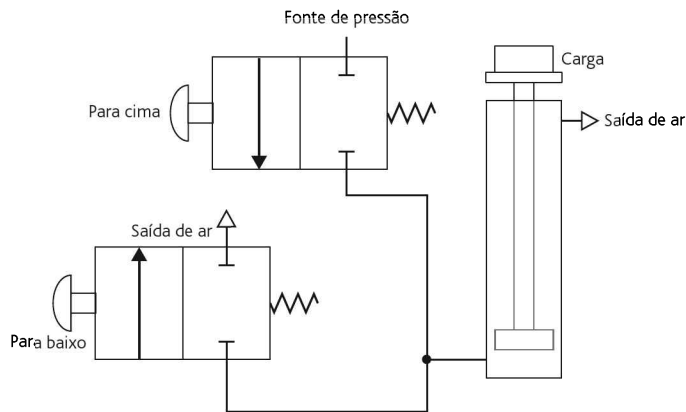


Figura 7.11 Sistema de levantamento de carga.

7.3.2 Válvulas piloto-operadas

A força necessária para mover a esfera ou o obturador em uma válvula pode ser em muitos casos muito grande para uma operação manual ou por solenoide. Para superar este problema, é usado um **sistema piloto-operado** onde uma válvula é usada para controlar uma segunda válvula. A Figura 7.12 ilustra isto. A válvula piloto é de pequena capacidade e pode ser acionada manualmente ou por solenoide. Ela é usada para permitir que uma válvula principal seja acionada por um sistema de pressão. A linha de pressão piloto é indicada por uma linha tracejada. As válvulas piloto e principal podem ser acionadas por duas válvulas em separado, mas elas geralmente são combinadas em uma única carcaça.

7.3.3 Válvulas direcionais

A Figura 7.13 mostra uma **válvula direcional** simples e o seu símbolo. A vazão livre pode ocorrer apenas em um sentido através da válvula: aquele que resulta na

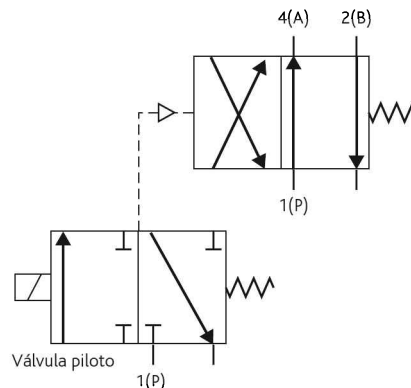


Figura 7.12 Sistema acionado por válvula piloto.

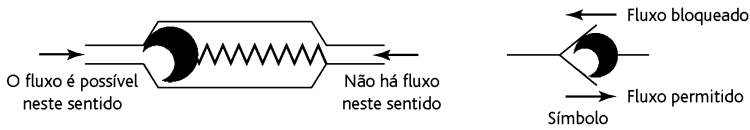


Figura 7.13 Válvula direcional.

esfera sendo pressionada contra uma mola. O fluxo no outro sentido é bloqueado pela mola que força a esfera contra sua sede.

7.4 Válvulas de controle de pressão

Existem três tipos principais de válvulas de controle de pressão:

1 *Válvula de regulação de pressão*

Estas são usadas para controlar a pressão de operação em um circuito e mantê-la em um valor constante.

2 *Válvulas de limitação de pressão*

Estas são usadas como dispositivos de segurança para limitar a pressão em um circuito abaixando-a para um valor seguro. A válvula abre dando saída do ar para a atmosfera, ou retornando-o para um coletor, se a pressão subir acima de um valor seguro. A Figura 7.14 mostra uma **válvula de alívio/limitação de pressão** que tem um orifício que é normalmente fechado. Quando a pressão de admissão supera a força exercida pela mola, a válvula abre e proporciona a saída do ar para a atmosfera, ou de volta para um coletor.

3 *Válvulas de sequência de pressão*

Estas válvulas são usadas para detectar a pressão de uma linha externa e enviar um sinal quando ela atingir um determinado valor ajustado. Com a válvula de limitação de pressão da Figura 7.15, a pressão de limitação é ajustada pela pressão na entrada de admissão da válvula. Podemos adaptar esta válvula para obtermos uma válvula de sequência. Isto pode ser usado para permitir que haja fluxo em uma parte do sistema quando a pressão subir ao nível desejado. Por exemplo, em uma máquina automática podemos necessitar que uma operação

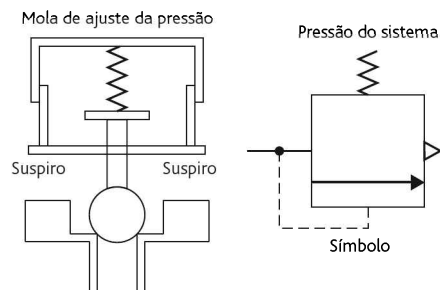


Figura 7.14 Válvula de limitação de pressão.

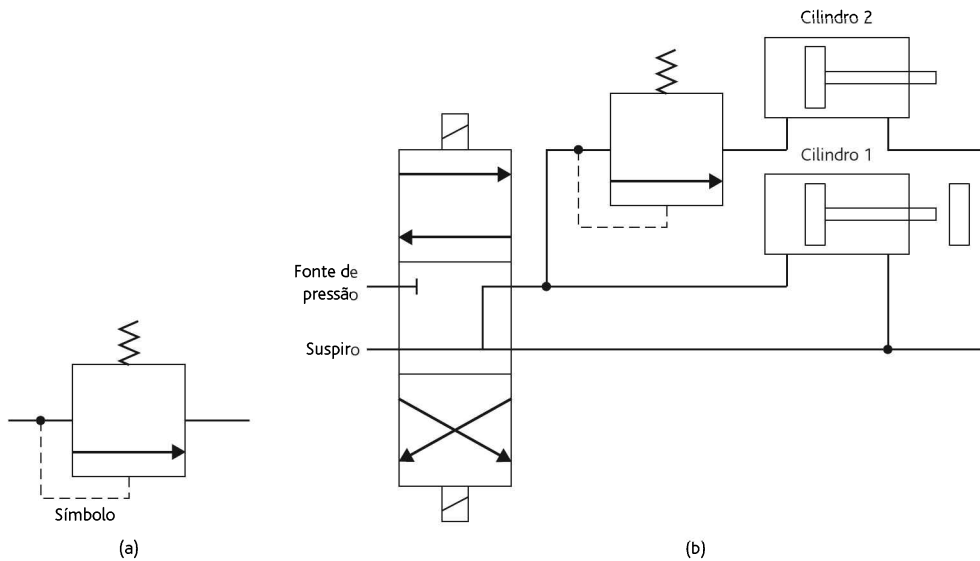


Figura 7.15 (a) Símbolo de uma válvula de sequência de pressão.

inicie quando a pressão de aperto aplicada a uma peça a ser trabalhada está em um determinado valor. A Figura 7.15(a) mostra o símbolo para uma válvula de sequência, que liga quando a pressão de admissão atinge um determinado valor e permitindo que a pressão seja aplicada ao sistema seguinte. A Figura 7.15(b) mostra um sistema em que esta válvula sequencial é usada. Quando a válvula 4/3 é acionada primeiro, a pressão é aplicada ao cilindro 1 e o seu êmbolo se move para a direita. Enquanto isto ocorre, a pressão é muito baixa para acionar a válvula de sequência e nenhuma pressão é aplicada ao cilindro 2. Quando o êmbolo do cilindro 1 atinge a extremidade e para, a pressão no sistema sobe e, em um nível apropriado, dispara a válvula de sequência para abrir e aplicar pressão no cilindro 2, iniciando o movimento do seu êmbolo.

7.5 Cilindros

O **cilindro hidráulico** ou **pneumático** é um exemplo de um atuador linear. Os princípios e tipos são os mesmos para as versões hidráulica e pneumática, sendo as diferenças apenas uma questão de tamanho como uma consequência das pressões maiores usadas em hidráulica. O cilindro consiste de um tubo cilíndrico ao longo do qual um pistão/êmbolo pode deslizar. Existem dois tipos básicos: os cilindros de ação simples e os de ação dupla.

O termo **ação simples** é usado quando a pressão de controle é aplicada apenas de um lado do pistão, que geralmente tem uma mola que fornece oposição ao seu movimento. O outro lado do pistão é aberto para a atmosfera. A Figura 7.16 mostra este tipo de cilindro com uma mola de retorno. O fluido é aplicado de um lado do pistão com um valor de pressão p , estando o outro lado na pressão atmosférica, produzindo

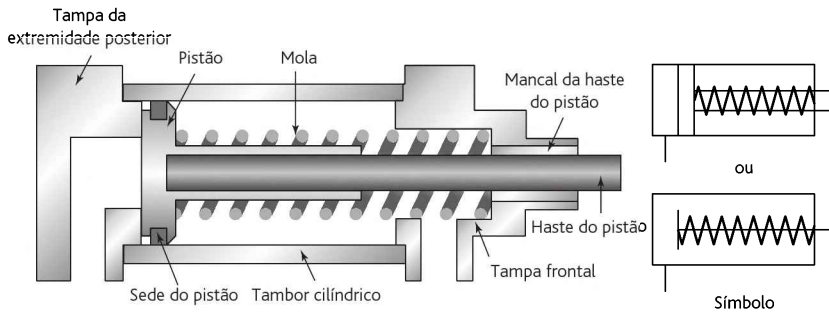


Figura 7.16 Cilindro de ação simples.

assim uma força no pistão de pA , em que A é a área do pistão. A força real que age na haste do pistão é menor do que esta por causa da fricção.

Para o cilindro de ação simples mostrado na Figura 7.17, quando passar uma corrente pelo solenoide, a válvula muda de posição e pressão é aplicada para mover o pistão ao longo do cilindro. Quando a corrente no solenoide cessa, a válvula inverte a sua posição inicial e o ar do cilindro sai para a atmosfera. Como consequência, a mola retorna o pistão ao longo do cilindro.

O termo ação dupla é usado quando as pressões de controle são aplicadas em cada lado do pistão (Figura 7.18). A diferença de pressão entre os dois lados resulta no movimento do pistão, sendo este capaz de se mover nos dois sentidos ao longo do cilindro como resultado dos sinais de pressão alta. Para o cilindro de ação dupla mostrado na Figura 7.19, a corrente em um solenoide faz com que o pistão se mova em um sentido, e a corrente no outro faz com que o movimento seja invertido.

A escolha de um cilindro é determinada pela força e a velocidade necessárias para mover a carga. Os cilindros hidráulicos são capazes de forças muito maiores do que os pneumáticos. Entretanto, os cilindros pneumáticos são capazes de velocidades maiores. A força produzida por um cilindro é igual à área da seção reta dele

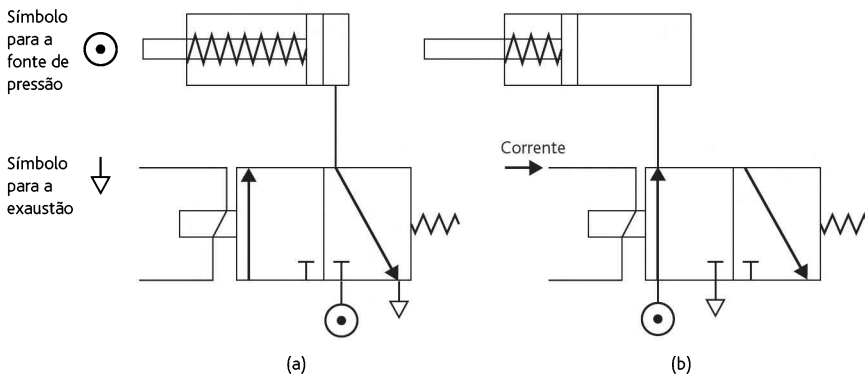


Figura 7.17 Controle de um cilindro de ação simples com (a) nenhuma corrente no solenoide, (b) uma corrente no solenoide.

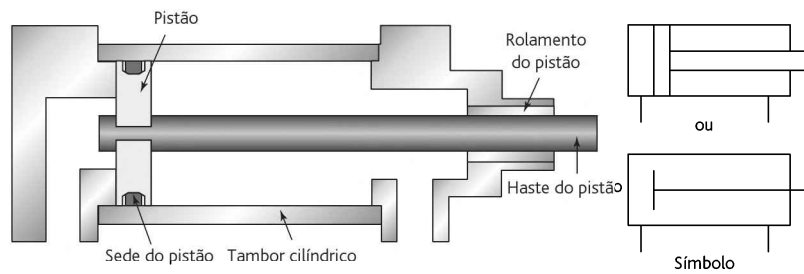


Figura 7.18 Cilindro de ação dupla.

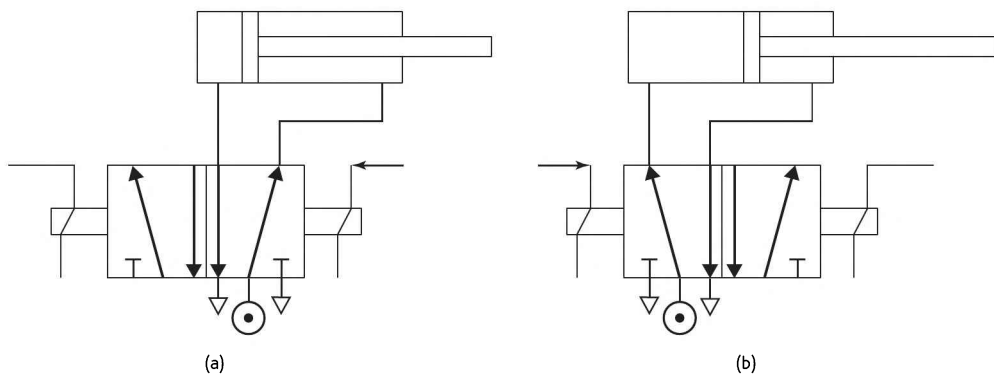


Figura 7.19 Controle de um cilindro de ação dupla com solenoide, (a) não ativado, (b) ativado.

multiplicada pela pressão de trabalho, ou seja, a diferença de pressão entre os dois lados do pistão no cilindro. Um cilindro usado com uma pressão de trabalho pneumática de 500 kPa e com um diâmetro de 50 mm fornece uma força de 982 N. Um cilindro hidráulico com o mesmo diâmetro e uma pressão de trabalho de 15.000 kPa fornece uma força de 29,5 kN.

Se a vazão do líquido hidráulico no cilindro for um volume de Q por segundo, então o volume arrastado para fora pelo pistão em um tempo de 1 s tem que ser Q . Porém, para um pistão com área de seção reta A , este é um movimento por uma distância correspondente à velocidade v em 1 s, onde temos $Q = Av$. Portanto, a velocidade v de um cilindro hidráulico é igual à vazão do líquido Q através do cilindro dividida pela área da seção reta A do cilindro. Portanto, para um cilindro hidráulico de 50 mm de diâmetro e um fluido hidráulico de $7,5 \times 10^{-3} \text{ m}^3/\text{s}$, a velocidade é 3,8 m/s. A velocidade de um cilindro pneumático não pode ser calculada desta forma, visto que sua velocidade depende da vazão do ar que pode ser liberado para a atmosfera posteriormente ao avanço do pistão. Uma válvula para ajustar isto pode ser usada para regular a velocidade.

Para ilustrar isto, considere o problema de um cilindro hidráulico usado para mover uma peça a ser trabalhada em uma operação da fabricação de um item ao longo de uma distância de 250 mm em 15 s. Se uma força de 50 kN for necessária para mover a peça, qual é a pressão de trabalho necessária e a vazão do líquido hidráulico se um cilindro com um pistão de 150 mm de diâmetro estiver disponível? A área da

seção reta do pistão é $\frac{1}{4}\pi \times 0,150^2 = 0,0177 \text{ m}^2$. A força produzida pelo cilindro é igual ao produto da área da seção reta do cilindro pela pressão de trabalho. Portanto, a pressão de trabalho é $50 \times 10^3 / 0,0177 = 2,8 \text{ MPa}$. A velocidade de um cilindro hidráulico é igual à vazão do líquido pelo cilindro dividida pela área da seção reta dele. Assim, a vazão necessária é $(0,250/15) \times 0,0177 = 2,95 \times 10^{-4} \text{ m}^3/\text{s}$.

7.5.1 Sequenciamento de cilindros

Muitos sistemas de controle empregam cilindros pneumáticos ou hidráulicos como os elementos de atuação e necessitam de uma sequência de extensões e retrações dos cilindros. Por exemplo, podemos ter dois cilindros A e B e necessitar que, quando o botão iniciar for pressionado, o pistão do cilindro A avance e então, quando ele estiver completamente estendido, o pistão do cilindro B seja estendido. Quando isto acontecer e os dois estiverem estendidos, podemos precisar que o pistão B retraia. Em discussões de controle sequencial com cilindros, é uma prática comum atribuir a cada cilindro as letras A, B, C, D, etc., e para indicar o estado de cada cilindro utiliza-se um sinal positivo (+) se ele estiver estendido e um sinal negativo (-) se estiver retraído. Portanto, esta sequência desejada de operações é A+, B+, A-, B-. A Figura 7.20 mostra um circuito que poderia ser usado para gerar esta sequência.

A sequência de operação é:

- 1 Inicialmente os dois cilindros têm os pistões retraídos. O botão iniciar na válvula 1 é pressionado. Com isto, uma pressão é aplicada na válvula 2, visto que a

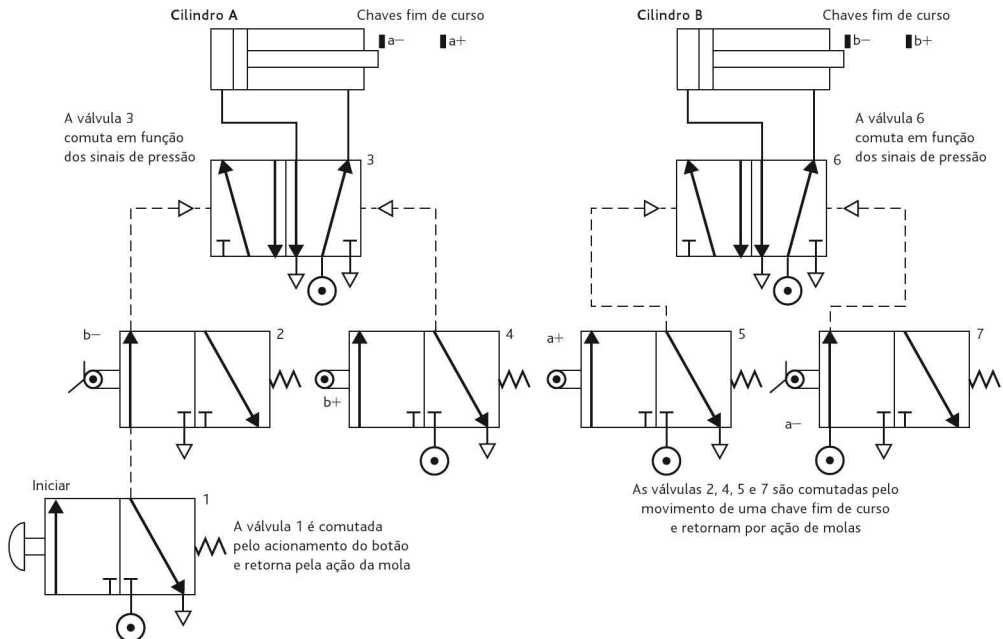


Figura 7.20 Operação sequencial com dois atuadores.

chave fim de curso $-b$ está inicialmente ativada, portanto, a válvula 3 é comutada aplicando uma pressão no cilindro A para estendê-lo.

- 2 O cilindro A estende, liberando o fim de curso $a-$. Quando o cilindro A estiver totalmente estendido, a chave fim de curso $a+$ é ativada. Isto comuta a válvula 5 e faz com que uma pressão seja aplicada na válvula 6, permitindo a aplicação de uma pressão no cilindro B e fazendo com que ele se estenda.
- 3 O cilindro B se estende, liberando a chave fim de curso $b-$. Quando o cilindro B estiver totalmente estendido, a chave fim de curso $b+$ é acionada. Isto comuta a válvula 4 e faz com que uma pressão seja aplicada na válvula 3, e assim aplica uma pressão no cilindro A para iniciar a retração do pistão.
- 4 O cilindro A retrai, liberando a chave fim de curso $a+$. Quando o cilindro A estiver totalmente retraído, a chave fim de curso $a-$ é ativada. Isto comuta a válvula 7 e faz com que uma pressão seja aplicada na válvula 5, e assim aplica uma pressão no cilindro B para iniciar a retração do seu pistão.
- 5 O cilindro B retrai, liberando a chave fim de curso $b+$. Quando o cilindro B estiver totalmente retraído, a chave fim de curso $b-$ é acionada, completando o ciclo.

O ciclo pode ser iniciado novamente pressionando-se o botão iniciar. Se quisermos que o sistema funcione continuamente, o último movimento da sequência tem que disparar o primeiro.

Uma forma alternativa de realizar esta sequência envolve uma fonte de ar a ser ligada e desligada nos grupos de válvulas, sendo denominada **controle em cascata**. Isto evita um problema que pode ocorrer com circuitos como o que é mostrado na Figura 7.20, do ar ficar confinado na linha de pressão para controlar uma válvula e assim evitar que a válvula comute. Com o controle em cascata, a sequência de operações é dividida em grupos com nenhuma letra de cilindro aparecendo mais do que uma vez em cada grupo. Portanto, para a sequência $A+, B+, B-, A-$, temos os grupos $A+, B+$ e $A-, B-$. Uma válvula é usada para comutar a fonte de ar entre os dois grupos, ou seja, ar para o grupo $A+B+$ e, em seguida, o ar é comutado para o grupo $A-B-$. Uma válvula iniciar/parar é incluída na linha que seleciona o primeiro grupo e, se a sequência for se repetir continuamente, a última operação tem que gerar um sinal para iniciar a sequência novamente. A primeira função em cada grupo é iniciada pelo grupo no qual a fonte está ligada; as ações posteriores dentro do grupo são controladas pelas válvulas acionadas por chaves e a última operação da válvula inicia o próximo grupo selecionado. A Figura 7.21 mostra o circuito pneumático.

7.6 Válvulas de controle proporcional e servo-assistida

Válvulas de controle proporcional e servo-assistida são válvulas de posições infinitas que proporcionam um deslocamento do carretel da válvula proporcional à corrente do solenoide. Basicamente, as válvulas servo-assistidas têm um motor de torque para mover o carretel dentro da válvula (Figura 7.22). Variando a corrente no motor de torque, uma armadura é defletida, isto move o carretel na válvula e obtemos um fluxo relacionado à corrente. As válvulas servo-assistida são de alta precisão, caras e geralmente são usadas em sistemas de controle em malha fechada.

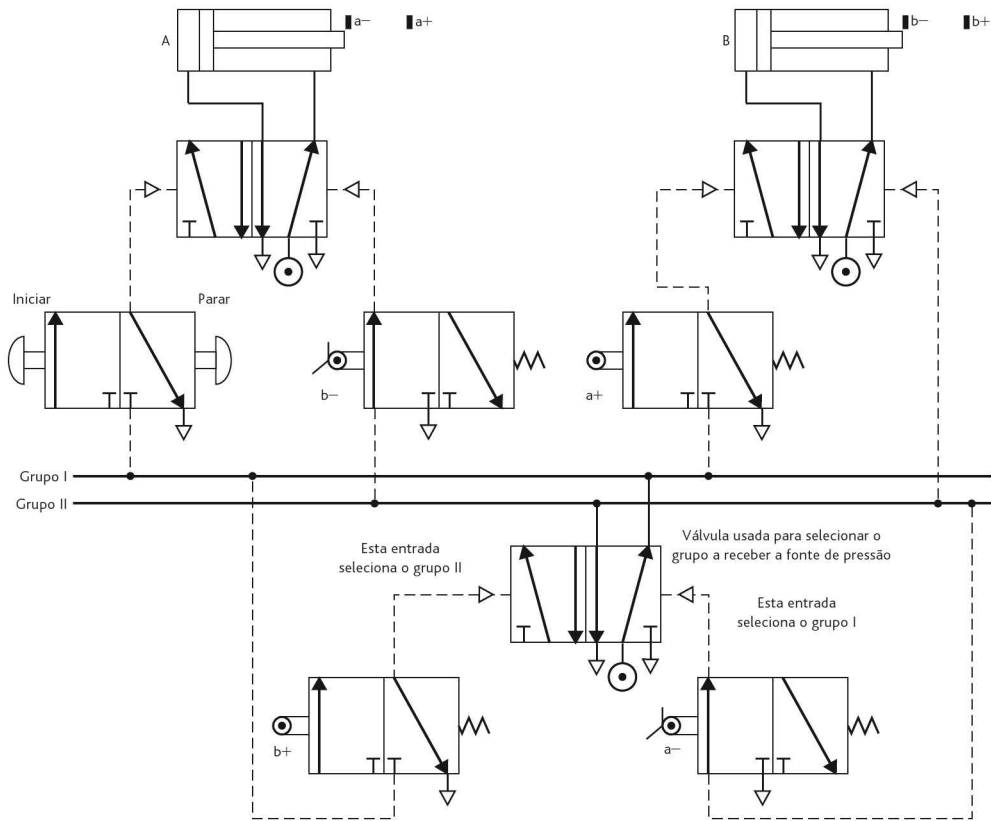


Figura 7.21 Controle em cascata usado para gerar A+, B+, B-, A-.

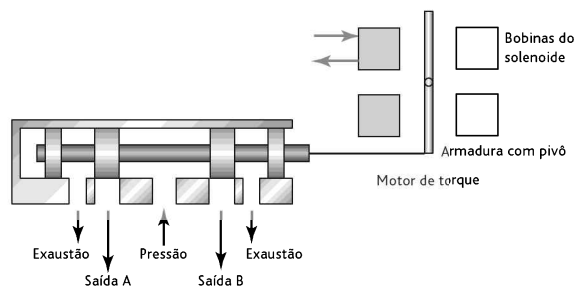


Figura 7.22 Forma básica de uma válvula servo-assistida.

As válvulas de controle proporcional são menos caras e basicamente têm a posição do carretel diretamente controlada pela intensidade da corrente no solenoide da válvula. Elas são frequentemente usadas em sistemas de controle em malha aberta.

7.7 Válvulas de controle de processos

As **válvulas de controle de processos** são usadas para controlar a vazão de um fluido e são usadas onde normalmente a vazão de um líquido que entra em um tanque tem que ser controlada. A base destas válvulas é um atuador que é usado para mover um obturador (*plug*) dentro de um tubo, alterando a seção reta do tubo pelo qual o fluido passa.

Uma forma comum de atuador pneumático usado em válvulas de controle de processos é o **atuador de diafragma**. Ele consiste essencialmente de um diafragma com o sinal de pressão de entrada proveniente do controlador colocado de um lado do diafragma e a pressão atmosférica, do outro. Esta diferença de pressão é denominada **pressão manométrica**. O diafragma é feito de uma borracha entre dois discos de aço no centro. O efeito da variação na pressão de entrada é mover a parte central do diafragma, conforme ilustra a Figura 7.23(a). Este movimento é comunicado ao elemento final de controle por um eixo fixado no diafragma, como, por exemplo, podemos ver na Figura 7.23(b).

A força F que age no eixo é a força que age no diafragma e é, portanto, a pressão manométrica P multiplicada pela área do diafragma A . A força de restauração é proporcionada por uma mola. Portanto, se o eixo se move em uma distância x , e

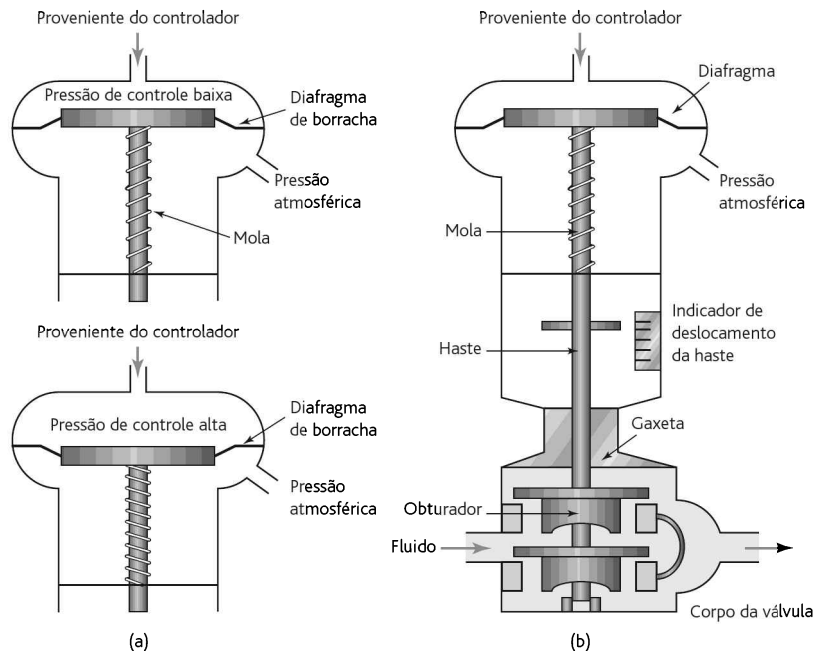


Figura 7.23 (a) Atuador com diafragma pneumático, (b) válvula de controle.

considerando que a compressão da mola é proporcional à força, ou seja, $F = kx$, sendo k uma constante, então $kx = PA$ e o deslocamento do eixo é proporcional à pressão manométrica.

Para ilustrar isto, considere o problema de um atuador de diafragma usado para abrir uma válvula de controle se uma força de 500 N tem que ser aplicada à válvula. Qual área do diafragma é necessária para uma pressão manométrica de controle de 100 kPa? A força F aplicada ao diafragma de área A pela pressão P é dada por $P = F/A$. Portanto, $A = 500/(100 \times 10^3) = 0,005 \text{ m}^2$.

7.7.1 Corpos e obturadores de válvulas

A Figura 7.23(b) mostra uma seção reta de uma válvula que controla a vazão de um fluido. A pressão que varia no atuador faz com que se mova o diafragma e, conseqüentemente, a haste da válvula. O resultado é o movimento do obturador dentro do corpo da válvula. O obturador restringe a passagem do fluido, e portanto sua posição determina a vazão.

Existem muitos tipos de corpos e obturadores de válvulas. A Figura 7.24 mostra alguns tipos. O termo **válvula de uma só sede** é usado para a válvula que tem apenas uma via interna para o fluido passar e portanto apenas um obturador é necessário para controlar este fluxo. O termo **válvula de dupla sede** é usado para a válvula em que o fluido que entra nela é dividido em dois, como mostra a Figura 7.23, em que cada um dos fluxos passa por um orifício controlado por um obturador. Então, existem dois obturadores neste tipo de válvula.

Uma válvula de uma só sede tem a vantagem de poder ser fechada mais firmemente do que uma de sede dupla, mas tem a desvantagem de que a força sobre o obturador devido ao fluxo é muito maior e o diafragma no atuador tem que exercer uma força consideravelmente maior na haste. Isto pode resultar em problemas no posicionamento preciso do obturador. As válvulas de dupla sede têm vantagem neste aspecto. A forma do corpo também determina se um aumento na pressão de ar resultará na abertura ou fechamento da válvula.

O formato do obturador determina a relação entre o movimento do eixo e o efeito sobre a vazão. A Figura 7.25(a) mostra três tipos normalmente usados e a

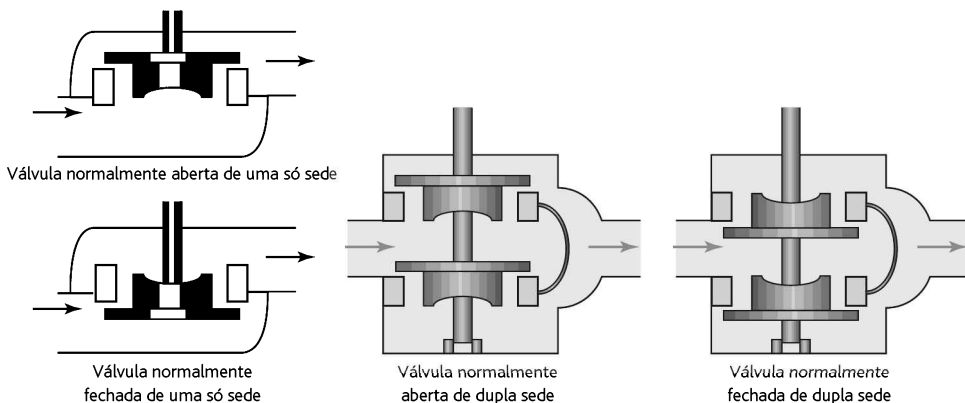


Figura 7.24 Corpos de válvulas.

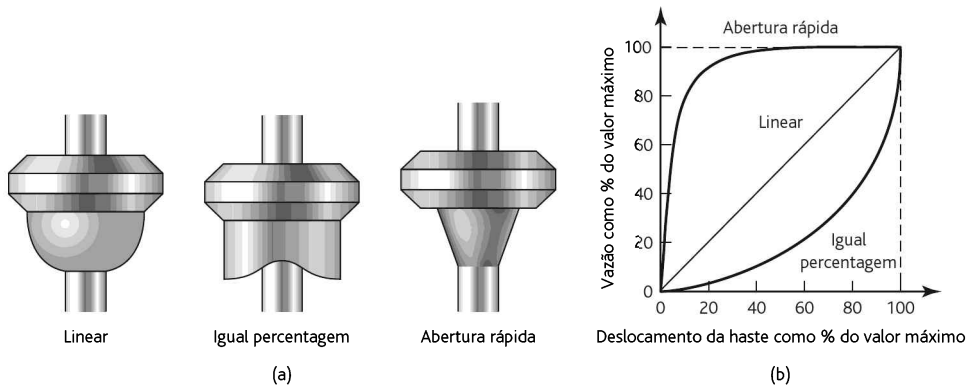


Figura 7.25 (a) Formatos de obturadores, (b) características de vazão.

Figura 7.25(b) mostra como a percentagem da vazão volumétrica está relacionada à percentagem do deslocamento da haste da válvula.

Com o tipo **abertura rápida**, pode ocorrer uma grande vazão para um pequeno movimento da haste da válvula. Este obturador é usado onde é necessário o controle *on/off* de vazão.

Com o tipo **linear**, a variação na vazão é proporcional à variação no deslocamento da haste da válvula, ou seja,

$$\text{variação na vazão} = k(\text{variação no deslocamento da haste})$$

em que k é uma constante. Se Q for a vazão para um deslocamento de haste da válvula S , e $Q_{\text{máx}}$ for a máxima vazão para um deslocamento de haste máximo $S_{\text{máx}}$, então temos

$$\frac{Q}{Q_{\text{máx}}} = \frac{S}{S_{\text{máx}}}$$

ou a percentagem de variação da vazão é igual à percentagem de variação do deslocamento da haste.

Para ilustrar isto, considere o problema de um atuador que tem um movimento de haste máximo de 30 mm. Ele é montado em uma válvula com obturador linear que tem uma vazão mínima de 0 e máxima de 40 m³/s. Qual será a vazão quando o movimento da haste for (a) 10 mm, (b) 20 mm? Como a percentagem da vazão é igual à percentagem do deslocamento da haste, então: (a) a percentagem do deslocamento da haste de 33% resulta em uma percentagem da vazão de 33%, ou seja, 13 m³/s; (b) uma percentagem de deslocamento da haste de 67% resulta em uma percentagem de vazão de 67%, ou seja, 27 m³/s.

Com o tipo de obturador de **igual percentagem**, a variação de percentagem igual na vazão ocorre para variações iguais na posição da haste, ou seja,

$$\frac{\Delta Q}{Q} = k\Delta S$$

em que ΔQ é a variação na vazão Q e ΔS é a variação na posição da válvula que resulta da variação na vazão. Se escrevermos esta expressão para pequenas variações e calcular a integral, teremos

$$\int_{Q_{\min}}^Q \frac{1}{Q} dQ = k \int_{S_{\min}}^S dS$$

$$\ln Q - \ln Q_{\min} = k(S - S_{\min})$$

Se considerarmos a vazão Q_{\max} que é dada por S_{\max} , então,

$$\ln Q_{\max} - \ln Q_{\min} = k(S_{\max} - S_{\min})$$

Eliminando k destas duas equações, obtemos

$$\frac{\ln Q - \ln Q_{\min}}{\ln Q_{\max} - \ln Q_{\min}} = \frac{S - S_{\min}}{S_{\max} - S_{\min}}$$

$$\ln \frac{Q}{Q_{\min}} = \frac{S - S_{\min}}{S_{\max} - S_{\min}} \ln \frac{Q_{\max}}{Q_{\min}}$$

e assim

$$\frac{Q}{Q_{\min}} = \left(\frac{Q_{\max}}{Q_{\min}} \right)^{(S - S_{\min}) / (S_{\max} - S_{\min})}$$

O termo **rangeabilidade** R é usado para a relação Q_{\max}/Q_{\min} .

Para ilustrar isto, considere o problema de um atuador que tem uma haste máxima de 30 mm. Ela é montada em uma válvula de controle que tem uma obturação de igual percentagem e que tem uma vazão mínima de $2 \text{ m}^3/\text{s}$ e máxima de $24 \text{ m}^3/\text{s}$. Qual é a vazão quando o movimento da haste for de (a) 10 mm, (b) 20 mm? Usando a equação

$$\frac{Q}{Q_{\min}} = \left(\frac{Q_{\max}}{Q_{\min}} \right)^{(S - S_{\min}) / (S_{\max} - S_{\min})}$$

temos para (a) $Q = 2 \times (24/2)^{10/30} = 4,6 \text{ m}^3/\text{s}$ e para (b) $Q = 2 \times (24/2)^{20/30} = 10,5 \text{ m}^3/\text{s}$.

A relação entre a vazão e o deslocamento da haste é a característica inerente da válvula. Na prática isto é válido apenas se as quedas de pressão no restante da tubulação, etc. forem desprezíveis em comparação com a queda de pressão na própria válvula. Se houver uma queda de pressão grande na tubulação de forma que, por exemplo, menos da metade da queda de pressão ocorre na válvula, então a característica linear pode se tornar quase uma característica de abertura rápida. A característica linear é muito usada quando é necessário uma resposta linear e a maior parte da pressão do sistema aparece como queda na válvula. O efeito de uma queda de pressão grande na tubulação com uma válvula de igual percentagem é torná-la mais parecida com a característica linear. Por esta razão, se for necessário uma resposta linear quando apenas uma pequena proporção da pressão do sistema aparece como queda na válvula, então um valor de igual percentagem pode ser usado.

7.7.2 Dimensionamento da válvula de controle

O termo dimensionamento da válvula de controle é usado para o procedimento de determinar a dimensão correta do corpo da válvula. A equação que relaciona a vazão do líquido Q através de uma válvula bem aberta e a sua dimensão é

$$Q = A_v \sqrt{\frac{\Delta P}{\rho}}$$

em que A_v é o coeficiente de vazão da válvula, ΔP é a queda de pressão na válvula e ρ é a densidade do fluido. Esta equação é escrita algumas vezes com as grandezas em unidades do SI, como

$$Q = 2,37 \times 10^{-5} C_v \sqrt{\frac{\Delta P}{\rho}}$$

em que C_v é o coeficiente de vazão da válvula. Alternativamente, ela pode ser escrita como

$$Q = 0,75 \times 10^{-6} C_v \sqrt{\frac{\Delta P}{G}}$$

em que G é a gravidade específica ou a densidade relativa. Estas duas últimas formas de equação são deduzidas de sua especificação original em termos de galões americanos. A Tabela 7.1 mostra alguns valores típicos de A_v e C_v e tamanhos de válvulas.

Para ilustrar isto, considere o problema de determinar o tamanho da válvula para uma válvula que precisa controlar a vazão de água quando a vazão máxima desejada for $0,012 \text{ m}^3/\text{s}$ e a queda de pressão permissível na válvula para esta vazão for 300 kPa . Usando a equação

$$Q = A_v \sqrt{\frac{\Delta P}{\rho}}$$

Então, visto que a densidade da água é 1.000 kg/m^3 ,

$$A_v = Q \sqrt{\frac{\rho}{\Delta P}} = 0,012 \sqrt{\frac{1000}{300 \times 10^3}} = 69,3 \times 10^{-5}$$

Portanto, usando a Tabela 7.1, o tamanho da válvula é 960 mm .

Tabela 7.1 Coeficientes de vazão e dimensões de válvulas

Coeficientes de fluxo	Dimensões de válvulas (mm)							
	480	640	800	960	1260	1600	1920	2560
C_v	8	14	22	30	50	75	110	200
$A_v \times 10^{-5}$	19	33	52	71	119	178	261	474

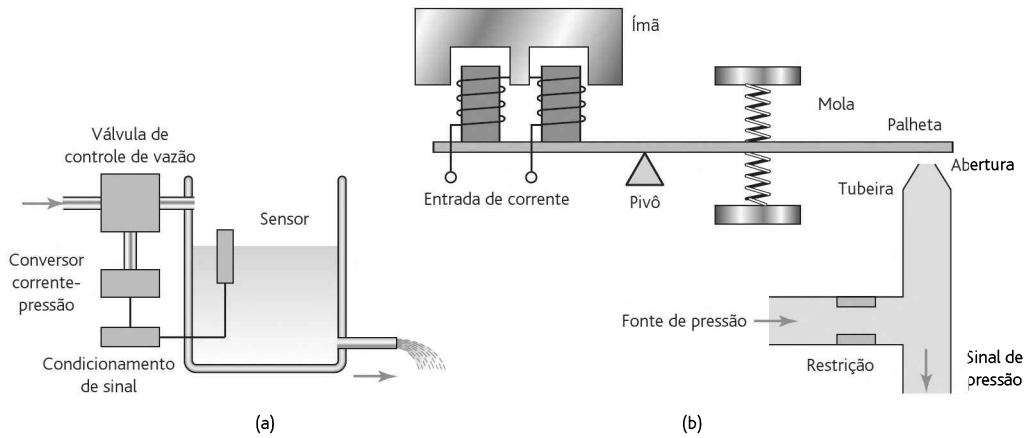


Figura 7.26 (a) Sistema de controle de fluido, (b) conversor corrente-pressão.

7.7.3 Exemplo de um sistema de controle de fluido

A Figura 7.26(a) mostra as características essenciais de um sistema para o controle de uma variável, como o nível de um líquido em um recipiente, controlando-se a vazão de líquido que entra nele. A saída do sensor de nível, após o condicionamento do sinal, é transmitida para um conversor corrente-pressão com entrada de 4 a 20 mA e saída de 20 a 100 kPa, que aciona uma válvula de controle pneumática e que, por sua vez, controla a vazão do líquido que é permitida entrar no recipiente.

A Figura 7.26(b) mostra o tipo do conversor corrente-pressão para este sistema. A corrente de entrada passa pelas bobinas montadas em um núcleo que é atraído por um ímã. A extensão da atração depende da intensidade da corrente. O movimento do núcleo provoca o movimento da alavanca em torno do pivô e, conseqüentemente, o movimento de uma palheta acima da tubeira. A posição da palheta em relação à tubeira determina a vazão de ar que pode escapar do sistema e, portanto, a pressão do ar no sistema. São usadas molas na palheta para ajustar a sensibilidade do con-

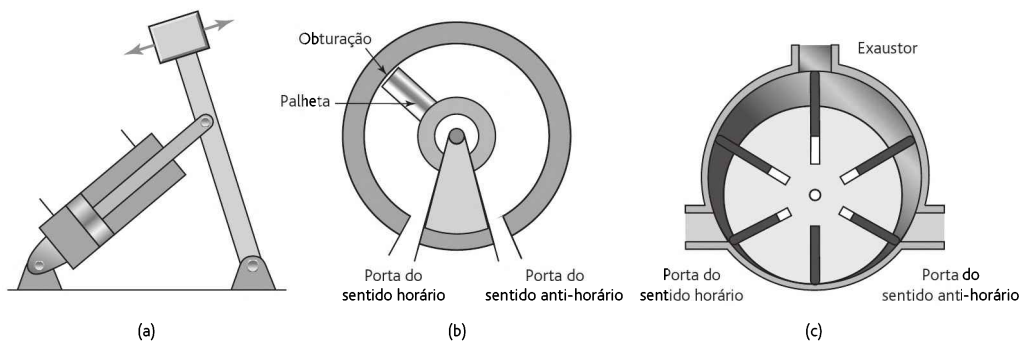


Figura 7.27 (a) Um cilindro linear usado para produzir rotação, (b) atuador semirrotativo do tipo palheta, (c) motor de palhetas.

versor de forma que as correntes de 4 a 20 mA produzam pressões de 20 a 100 kPa. Estes são valores padrão que são geralmente usados em sistemas como este.

7.8 Atuadores rotativos

Um cilindro linear pode, com as conexões mecânicas adequadas, ser usado para produzir movimento rotativo em ângulos menores do que 360°. A Figura 7.27(a) ilustra esta configuração. Uma alternativa é um **atuador semirrotativo** que envolve uma palheta (Figura 7.27(b)). Uma diferença de pressão entre as duas portas faz com que a palheta gire, fazendo girar o eixo que indica a medida da diferença de pressão. Dependendo da pressão, a palheta gira no sentido horário ou anti-horário.

Para a rotação de um ângulo maior do que 360°, pode ser usado um motor pneumático; um tipo é o **motor de palhetas** (Figura 7.27(c)). Um rotor excêntrico tem fendas nas quais as palhetas são fixadas e, por força centrífuga, são forçadas contra as paredes do cilindro. As palhetas dividem a câmara em compartimentos separados que aumentam de tamanho a partir da porta de admissão até a porta de exaustão. O ar que entra em um compartimento exerce uma força na palheta, fazendo o rotor girar. O motor pode ter o sentido de rotação invertido usando-se portas de admissão distintas.

Resumo

Os **sistemas pneumáticos** usam ar, e os **hidráulicos**, óleo. A principal desvantagem dos sistemas pneumáticos é a compressibilidade do ar. Os sistemas hidráulicos podem ser usados para dispositivos de controle de potência maior, mas são mais caros do que os sistemas pneumáticos, e existem perigos associados com vazamentos de óleo que não existem nos vazamentos de ar.

Os sistemas pneumáticos e hidráulicos usam **válvulas de controle direcional** para direcionar o fluxo do fluido pelo sistema. Estas válvulas são do tipo *on/off*. O símbolo usado para estas válvulas é um quadrado para cada uma de suas posições de chaveamento, com os símbolos usados em cada quadrado indicando as conexões feitas quando a posição de chaveamento é ativada.

O **cilindro hidráulico** ou **pneumático** consiste de um tubo cilíndrico ao longo do qual um pistão/êmbolo pode se deslocar. Existem dois tipos básicos: cilindros de **ação simples** e de **ação dupla**. Com o de ação simples, a pressão de controle é aplicada apenas de um lado do pistão. Geralmente é usada uma mola para oferecer oposição ao movimento do pistão. O outro lado do pistão é aberto para a atmosfera. O termo ação dupla é usado quando as pressões de controle são aplicadas em cada um dos lados do pistão.

As **válvulas de controle proporcional** e **servo-assistidas** são as válvulas de posições infinitas que proporcionam à válvula deslocamento do carretel proporcional à corrente fornecida ao solenoide.

As **válvulas de controle de processo** são usadas para controlar a vazão do fluido. A base destas válvulas é um atuador usado para mover um obturador (*plug*) na tubulação de vazão e assim alterar a seção reta da tubulação através da qual o

fluido passa. Existem vários tipos de corpos de válvulas e obturadores, os quais determinam como a válvula controla a vazão do fluido.

Problemas

- 7.1 Descreva os detalhes básicos de (a) uma válvula de gatilho, (b) uma válvula com obturador.
- 7.2 Explique o princípio de uma válvula operada por uma válvula piloto.
- 7.3 Explique como uma válvula sequencial pode ser usada para iniciar uma operação apenas quando uma outra operação tiver sido finalizada.
- 7.4 Desenhe os símbolos para (a) válvula de descarga da pressão, (b) uma válvula 2/2 que tem atuadores de um botão de pressão e uma mola, (c) uma válvula 4/2, (d) uma válvula direcional.

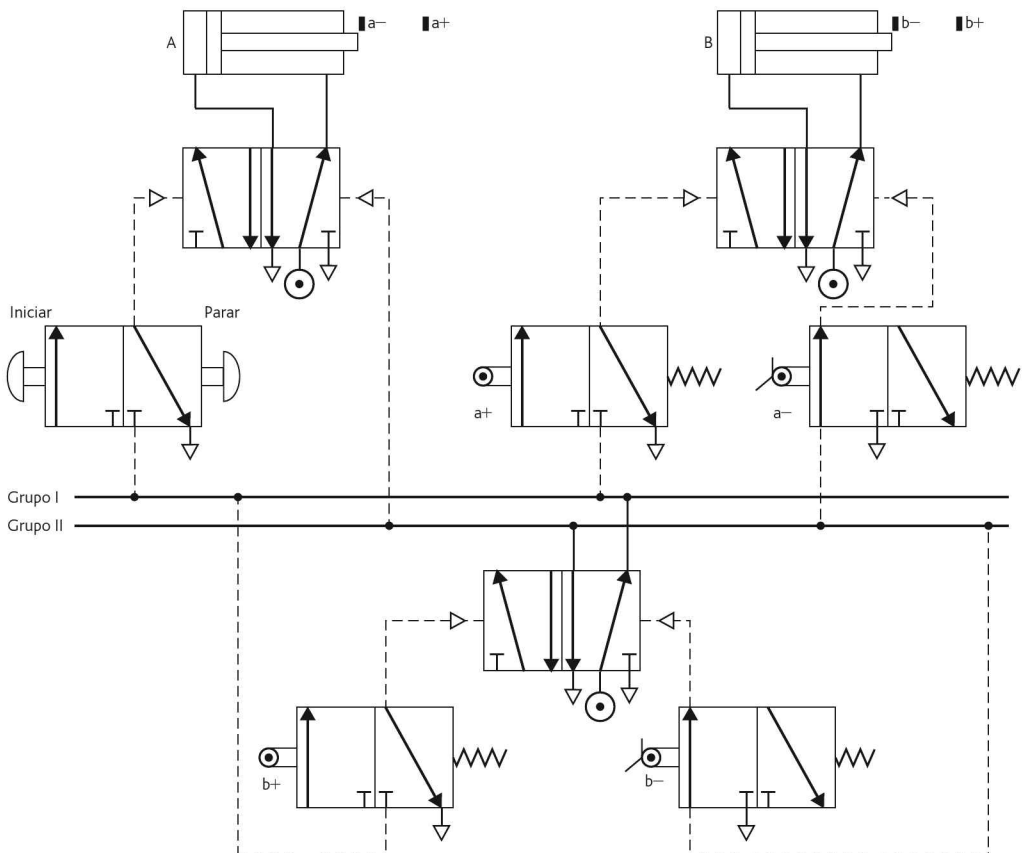


Figura 7.28 Problema 7.5.

- 7.5 Descreva a sequência de operações que ocorrerão para os cilindros A e B na Figura 7.28 quando o botão iniciar for pressionado. As chaves fim de curso a⁻, a⁺, b⁻ e b⁺ são detectadas quando os cilindros estão totalmente retraídos e totalmente estendidos.
- 7.6 Projete o circuito para uma válvula pneumática executar a sequência A⁺, seguida de B⁺ e então seguida simultaneamente por A⁻ e B⁻.
- 7.7 Uma força de 400 N é necessária para abrir uma válvula de controle de processo. Qual é a área do diafragma necessária com um atuador de diafragma para abrir a válvula com uma pressão manométrica de 70 kPa?
- 7.8 Um sistema pneumático é acionado com uma pressão de 1.000 kPa. Qual é o diâmetro que o cilindro deve ter para mover uma carga que requer uma força de 12 kN?
- 7.9 Um cilindro hidráulico é usado para mover uma peça a trabalhar em uma operação de manufatura por uma distância de 50 mm em 10 s. Uma força de 10 kN é necessária para mover a peça. Determine a pressão de trabalho necessária e a vazão do líquido se um cilindro com pistão de diâmetro de 100 mm for usado.
- 7.10 Um atuador tem um movimento de haste em que o curso total é de 40 mm. Ele é montado em uma válvula de controle de processo de obturador linear que tem uma vazão mínima de 0 e máxima de 0,20 m³/s. Qual será a vazão quando o movimento da haste for de (a) 10 mm, (b) 20 mm?
- 7.11 Um atuador tem um movimento de haste em que o curso total é de 40 mm. Ele é montado em uma válvula de controle de processo com um obturador de igual percentagem e que tem uma vazão mínima de 0,2 m³/s e máxima de 4,0 m³/s. Qual será a vazão quando o movimento da haste for de (a) 10 mm, (b) 20 mm?
- 7.12 Qual é o tamanho de uma válvula de controle de processo necessário para controlar a vazão de água quando a vazão máxima necessária for de 0,002 m³/s e a queda de pressão permitida na válvula para esta vazão for de 100 kPa? A densidade da água é de 1.000 kg/m³.

8

Sistemas de Acionamento Mecânico

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Determinar os possíveis sistemas de atuação mecânica para transmissão de movimento linear para rotativo, rotativo para rotativo, rotativo para linear e transmissões de movimentos cíclicos.
- Avaliar a capacidade de articulações, cames, engrenagens, catraca e lingueta, transmissões por correia e corrente e rolamentos para sistemas de acionamento.

8.1 Sistemas mecânicos

Este capítulo aborda **mecanismos**: mecanismos são dispositivos que podem ser considerados conversores de movimentos que transformam movimentos de uma forma em outra desejada. Eles podem, por exemplo, transformar movimento linear em rotacional, ou o movimento em uma direção para uma direção perpendicular, ou talvez o movimento linear de alternância em movimento rotacional, como ocorre em um motor de combustão interna em que o movimento de alternância dos pistões é convertido em rotação da manivela e, por sua vez, do eixo propulsor.

Os elementos mecânicos podem incluir o uso de sistemas de articulação, cames, engrenagens, cremalheira e pinhão, correntes, transmissão por correia, etc. Por exemplo, a cremalheira e o pinhão pode ser usada para converter um movimento rotacional em linear. Eixos de engrenagens paralelas podem ser usados para reduzir a velocidade de um eixo. Engrenagens cônicas podem ser usadas para a transmissão de movimento de rotação de 90° . Uma correia dentada ou uma transmissão a corrente podem ser usadas para transformar o movimento de rotação em torno de um eixo para outro. Cames e articulações podem ser usados para se obter movimentos que são definidos para variarem de uma forma particular. Este capítulo é uma consideração das características básicas de uma gama destes mecanismos.

Muitas das ações que antes eram obtidas com o uso de mecanismos estão, entretanto, sendo obtidas hoje em dia como resultados de abordagens de mecatrônica,

com o uso de sistemas microprocessados. Por exemplo, as cames em eixos rotativos eram usadas antes em máquinas de lavar domésticas para se obter uma sequência temporizada de ações, como abrir uma válvula para permitir a entrada de água no cesto, desligamento da água, acionamento de um aquecedor, etc. As máquinas de lavar modernas usam um sistema microprocessado programado para acionar saídas na sequência desejada. Um outro exemplo são as engrenagens e ponteiros de um relógio que foram amplamente substituídos por um circuito integrado com, geralmente, um display de cristal líquido. A abordagem da mecatrônica tem resultado em simplificação e, frequentemente, redução de custo.

Entretanto, os mecanismos ainda têm aplicações nos sistemas mecatrônicos. Por exemplo, os sistemas usados em uma câmera automática para ajuste do obturador para uma exposição correta envolvem um mecanismo para ajuste do tamanho do diafragma.

Embora a eletrônica possa ser usada frequentemente para muitas funções que antes eram executadas por mecanismos, estes ainda podem ser usados para proverem funções como:

- 1 Amplificação de força, que pode ser obtida com alavancas.
- 2 Mudança de velocidade, que pode ser obtida com engrenagens.
- 3 Transferência de rotação em torno de um eixo para outro, como por exemplo em uma correia dentada.
- 4 Tipos particulares de movimento, como o que é obtido com um mecanismo de retorno rápido.

O termo **cinemática** é usado para o estudo dos movimentos sem considerar as forças. Quando consideramos apenas os movimentos sem qualquer consideração de forças ou energias envolvidas, então estamos tratando de uma análise cinemática do mecanismo. Este capítulo é uma introdução a estas considerações.

8.2 Tipos de movimentos

O movimento de qualquer corpo rígido pode ser considerado uma combinação de movimentos de translação e rotação. Considerando as três dimensões do espaço, o **movimento de translação** pode ser decomposto em um ou mais dos três eixos (Figura 8.1(a)). Um **movimento de rotação** pode ser considerado como uma rotação que tem componentes de rotação em torno de um ou mais eixos (Figura 8.1(b)).

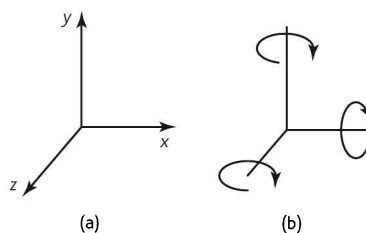


Figura 8.1 Tipos de movimento: (a) translacional, (b) rotacional.

Um movimento complexo pode ser uma combinação de movimentos de translação e rotação. Por exemplo, imagine o movimento necessário para você pegar um lápis na mesa. Este pode envolver o movimento da sua mão em um ângulo particular em direção à mesa, rotação da mão, e todos os movimentos associados com a abertura dos seus dedos e o movimento deles para a posição desejada para pegar o lápis. Esta é uma sequência de movimentos bastante complexos. Entretanto, podemos separar estes movimentos em uma combinação de movimentos de translação e rotação. Esta análise é particularmente relevante se não estamos movendo uma mão humana para pegar o lápis, mas instruindo um robô para que realize a tarefa. Então, realmente é necessário separar os movimentos em combinações de movimentos de translação e rotação de modo que possamos projetar mecanismos para realizar cada uma das partes do movimento. Por exemplo, uma sequência de sinais de controle enviados para um mecanismo pode ser que este grupo de sinais instrua a junção 1 a girar 20° e a articulação 2 a se estender por 4 mm para um movimento de translação.

8.2.1 Liberdades e restrições

Um aspecto importante no projeto de elementos mecânicos é a orientação e o arranjo dos elementos e partes. Um corpo livre no espaço pode se mover em três direções independentes e mutuamente perpendiculares e rotacionar de três formas em torno dessas direções (Figura 8.1). Diz-se que temos seis graus de liberdade. O número de **graus de liberdade** é o número de componentes do movimento que são necessárias para gerar o movimento. Se uma junção é limitada a se mover ao longo de uma linha, então os graus de liberdade de translação dela são reduzidos a um. A Figura 8.2(a) mostra uma junção com apenas um grau de liberdade de translação. Se uma junção for limitada a se mover em um plano, então ela tem dois graus de liberdade de translação. A Figura 8.2(b) mostra uma junção que tem um grau de liberdade de translação e um grau de liberdade de rotação.

O problema encontrado no projeto é frequentemente reduzir o número de graus de liberdade, e isto requer orientações e um número apropriado de restrições. Sem qualquer restrição, um corpo tem seis graus de liberdade. É necessário uma restrição para cada grau de liberdade que queremos evitar que ocorra. Desde que não tenhamos restrições redundantes, então o número de graus de liberdade será 6 menos o número de restrições. Entretanto, frequentemente ocorrem restrições redundantes, portanto temos a seguinte regra para restrições em um único corpo rígido:

$$6 - \text{número de restrições} = \text{número de graus de liberdade} \\ - \text{número de redundâncias}$$

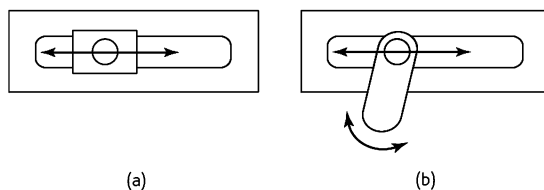


Figura 8.2 Junções com: (a) um grau de liberdade, (b) dois graus de liberdade.

Portanto, se for desejado que um corpo esteja fixo, ou seja, tenha zero graus de liberdade, então, sem restrições redundantes, o número de restrições necessária é 6.

Um conceito que é usado no projeto é o **princípio da restrição mínima**. Ele diz que, em um corpo fixo ou conduzido a um tipo particular de movimento, o número mínimo de restrições deve ser usado, ou seja, não deve haver redundâncias. Isto é frequentemente denominado **projeto cinemático**.

Por exemplo, para termos um eixo apenas com rotação (sem movimento de translação), temos que reduzir o número de graus de liberdade para 1. Assim, o número mínimo de restrições será 5. Qualquer restrição a mais leva a redundância. A montagem do eixo que pode ser usada tem um rolamento de esferas em uma extremidade e um rolamento de roletes na outra (Figura 8.3). Juntos, o par de rolamentos evita movimento de translação perpendicular ao eixo y e rotações em torno dos eixos y e z . O rolamento de esferas evita o movimento de translação ao longo dos eixos x e z . Então, existe um total de 5 restrições. Isto deixa apenas um grau de liberdade, a rotação desejada em torno do eixo x .

Se houvesse um rolamento de roletes em cada extremidade do eixo, eles evitariam o movimento de translação ao longo dos eixos x e z , e assim haveria redundância. Esta redundância poderia causar dano. Se fossem usados rolamentos de esferas nas duas extremidades do eixo, então, para evitar redundância, um dos rolamentos teria que ter o anel externo não fixado em sua carcaça, de modo que pudesse deslocar em alguma extensão a direção axial.

8.2.2 Efeito de carga

Os mecanismos são estruturas e, como tal, transmitem e suportam cargas. Então, uma análise é necessária para determinar as cargas a serem transportadas pelos elementos individuais. Assim, podem ser feitas considerações sobre as dimensões dos elementos de modo que seja possível, por exemplo, terem resistência suficiente e, talvez, rigidez para suportar estas cargas.

8.3 Cadeia cinemática

Quando consideramos os movimentos de um mecanismo sem qualquer referência às forças envolvidas, podemos tratar o mecanismo como sendo composto de uma série de articulações individuais. Cada parte do mecanismo que tem movimento relativo à outra parte é denominada **articulação**. Uma articulação não precisa necessariamente ser um corpo rígido, mas deve ser um corpo resistente capaz de transmitir

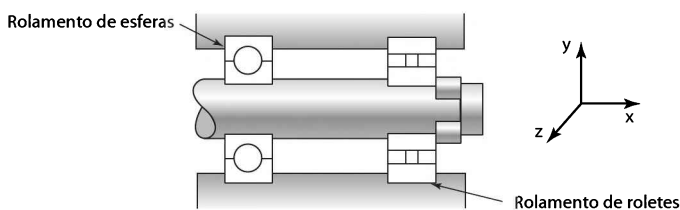


Figura 8.3 Eixo sem redundâncias.

a força desejada com uma deformação desprezível. Por esta razão ela é geralmente tida como representada por um corpo rígido que tem dois ou mais pontos de conexão com outras articulações, sendo denominados **nós**. Cada articulação é capaz de se mover relativamente às articulações vizinhas. A Figura 8.4 mostra exemplos de articulações com dois, três e quatro nós. Uma **junção** é uma conexão entre duas ou mais articulações em seus nós que permite movimento entre as articulações conectadas. Alavancas, manivelas, hastes de conexões e pistões, cursores, polias, correias e eixos são exemplos de articulações.

Uma sequência de junções e articulações é conhecida por **cadeia cinemática**. Para uma cadeia cinemática transmitir movimento, uma articulação tem que ser fixa. O movimento de uma articulação produz o movimento relativo previsível das outras. É possível obter de uma cadeia cinemática diferentes mecanismos tendo-se uma articulação diferente como a fixa.

Como ilustração de uma cadeia cinemática, considere o motor de um carro em que o movimento de alternância de um pistão é transformado em movimento de rotação de um eixo de manivelas em rolamentos montados em um quadro fixo (Figura 8.5(a)). Podemos representar isto como quatro articulações conectadas (Figura 8.5(b)). A articulação 1 é o eixo de manivelas, a 2, a haste de conexão, a 3, o quadro

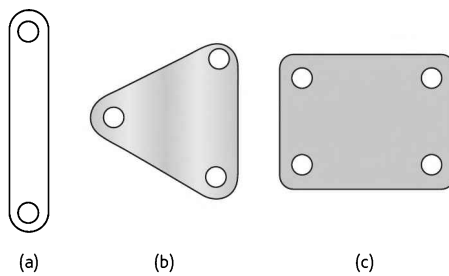


Figura 8.4 Articulações: (a) com dois nós, (b) com três nós, (c) com quatro nós.

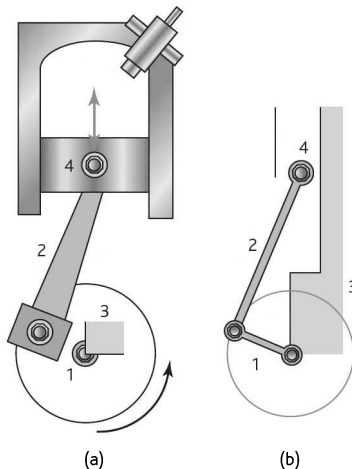


Figura 8.5 Mecanismo de um motor de combustão simples.

fixo e a 4, o cursor, ou seja, o pistão, que se move relativamente ao quadro fixo (veja a Seção 8.3.2 para um estudo mais amplo).

Os projetistas de mecanismos se baseiam em duas formas básicas de cadeia cinemática, a de quatro barras e a do cursor-manivela. As ilustrações a seguir são de algumas formas de cadeias.

8.3.1 Cadeia de quatro barras

A **cadeia de quatro barras** consiste de quatro articulações conectadas, resultando em quatro junções em torno das quais o giro ocorre. A Figura 8.6 mostra algumas formas de cadeias de quatro barras construídas com a alteração dos comprimentos relativos das articulações. Se a soma do comprimento da menor articulação com o da maior for menor ou igual à soma dos comprimentos das outras duas, então pelo menos uma articulação será capaz de realizar uma revolução completa em relação à articulação fixa. Se esta condição não ocorrer, então nenhuma articulação é capaz de fazer uma revolução completa. Isto é conhecido como condição de Grashof. Na Figura 8.6(a), a articulação 3 é fixa e os comprimentos relativos das articulações são de modo que as articulações 1 e 4 oscilam mas não giram. O resultado é um **mecanismo de alavanca dupla**. Diminuindo-se a articulação 4 em relação à 1, a articulação 4 pode girar (Figura 8.6(b)) com a articulação 1 oscilando, e o resultado é denominado **mecanismo alavanca-manivela**. Com as articulações 1 e 4 de comprimentos iguais e ambas capazes de girar (Figura 8.6(c)), então o resultado é um **mecanismo de manivela dupla**. Alterando a articulação fixa, as outras formas de mecanismo podem ser produzidas.

A Figura 8.7 ilustra como este mecanismo pode ser usado para girar o filme em um projetor de cinema. Conforme a articulação 1 gira, a extremidade da articulação 2 encaixa em um dos orifícios na lateral do filme, puxa-o para frente antes de liberá-lo e se mover para trás sobre ele para encaixar no próximo orifício.

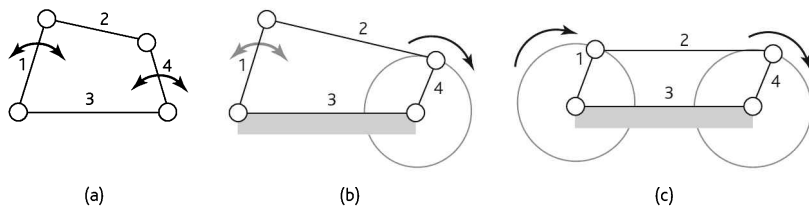


Figura 8.6 Exemplos de cadeias de quatro barras.

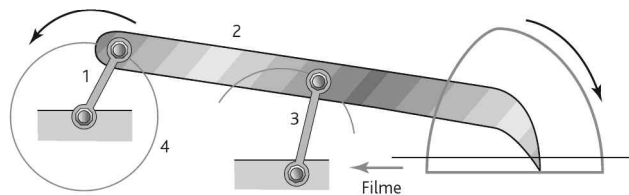


Figura 8.7 Mecanismo de avanço de um filme de cinema.

Alguns sistemas articulados têm **posições limites**. São posições em que esses sistemas não reagem a qualquer entrada a partir de suas articulações. A Figura 8.8 ilustra as posições limites de um sistema articulado, sendo o sistema usado para controlar o movimento da porta traseira de um caminhão de forma que quando a articulação 2 alcança a posição horizontal nenhuma carga a mais na articulação 2 provoca qualquer movimento. Existe uma outra posição limite para o sistema de articulação que ocorre quando as articulações 3 e 4 estão na vertical, assim como a tampa traseira.

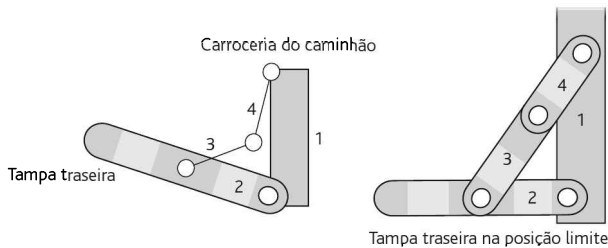


Figura 8.8 Sistema articulado com posições limites.

8.3.2 Mecanismo cursor-manivela

Esta forma de mecanismo consiste de uma manivela, uma haste de conexão e um cursor, e é o tipo de mecanismo descrito na Figura 8.5 que mostra o mecanismo simples de um motor de combustão. Nesta configuração, a articulação 3 é fixa, ou seja, não há movimento relativo entre o centro de rotação da manivela e o cilindro no qual o pistão desliza. A articulação 1 é a manivela que gira, a 2 é a haste de conexão, e a 4 é o cursor que se move relativamente à articulação fixa. Quando o pistão se move para trás e para frente, a manivela, articulação 1, é forçada a girar. Portanto, este mecanismo transforma uma entrada de movimento para frente e para trás em movimento rotacional.

A Figura 8.9 mostra uma outra forma deste tipo de mecanismo, um **mecanismo de retorno rápido**. Ele consiste de uma manivela girante, articulação AB, que gira em torno de um centro fixo, uma alavanca oscilante CD, cuja causa da oscilação em torno de C é o movimento deslizante do bloco em B ao longo de CD conforme AB gira, e a articulação DE que faz com que E se mova para frente e para trás. E pode ser o cabeçote de uma máquina que tem uma ferramenta de corte fixada a ele. O cabeçote estará nos extremos do seu movimento quando as posições da manivela forem AB_1 e AB_2 . Portanto, à medida que a manivela se move no sentido anti-horário de B_1 para B_2 , o cabeçote faz um percurso completo, o percurso de corte. Quando a manivela continua o seu movimento de B_2 para B_1 , no sentido anti-horário, o cabeçote faz novamente um percurso completo na direção oposta, o percurso de retorno. Com a rotação da manivela a uma velocidade constante, e devido ao ângulo de rotação da manivela necessário para o percurso de corte ser maior do que o ângulo para o percurso de

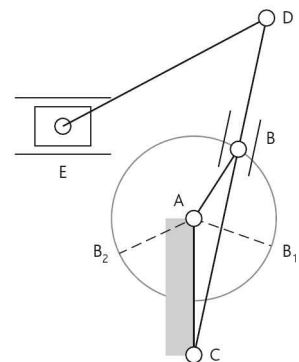


Figura 8.9 Mecanismo de retorno rápido.

retorno, o percurso de corte gasta mais tempo do que o de retorno. Por isto o termo “retorno rápido” para o mecanismo.

8.4 Cames

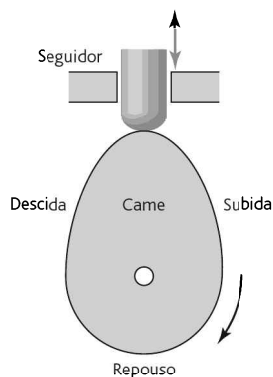


Figura 8.10 Cme e o seguidor de cme.

Uma **cme** é um corpo que gira ou oscila e, ao fazer isto, imprime um movimento de alternância ou de oscilação no outro corpo, denominado de **seguidor**, com o qual está em contato (Figura 8.10). Conforme a cme gira, faz o seguidor subir, repousar e descer, e o tempo gasto em cada uma dessas posições depende do formato da cme. A seção de subida da cme é a parte que eleva o seguidor, e o seu perfil determina a rapidez com que o seguidor da cme é erguido. A seção de descida da cme é a parte em que o seguidor abaixa, e a rapidez é determinada pelo seu perfil. A seção de repouso da cme é a parte que permite que o seguidor permaneça no mesmo nível por um período significativo do tempo. Esta seção é onde a cme é circular tendo um raio que não varia.

A produção de um movimento particular do seguidor depende do formato da cme e do tipo de seguidor usado. A Figura 8.11 mostra os tipos de diagramas de deslocamento de um seguidor que pode ser produzido para diferentes formatos de cames e seguidores com ponta ou gume de faca. A distancia radial do eixo de rotação da cme para o ponto de contato dela com o seguidor determina o deslocamento do seguidor em relação ao eixo de rotação da cme. As figuras mostram como as distâncias radiais, e portanto o deslocamento do seguidor, variam com o ângulo de rotação da cme.

A cme excêntrica (Figura 8.11(a)) é uma cme circular com um centro de rotação deslocado. Ela produz uma oscilação no seguidor que é um movimento harmônico simples e é frequentemente usada com bombas. A cme com formato de coração (Figura 8.11(b)) resulta em um deslocamento do seguidor que aumenta a uma taxa constante com o tempo antes de diminuir também a uma taxa constante, por isto a velocidade uniforme do seguidor. A cme com formato de pera (Figura 8.11(c)) resulta em um movimento do seguidor que é estacionário em torno da metade do movimento da cme e aumenta e diminui simetricamente durante os dois quartos restantes da revolução. Esta cme no formato de pera é usada no controle de válvulas de motor a combustão. O período de repouso mantém a válvula aberta enquanto a mistura ar/combustível enche o cilindro. Quanto mais longo o período de repouso, ou seja, maior o comprimento da cme com raio constante, mais tempo para o cilindro encher com o vapor inflamável.

A Figura 8.12 mostra alguns exemplos de diferentes tipos de seguidores de cme. Os seguidores de rolete são essencialmente rolamentos de esfera ou rolete. Eles têm a vantagem de ter uma fricção menor do que um de contato, porém podem ser mais caro. Os seguidores com uma face plana são frequentemente usados porque eles são baratos e podem ser construídos em tamanhos menores do que os de rolete. Estes seguidores são muito usados com as cames de válvulas de motor a combustão. Embora as cames possam rodar a seco, frequentemente são usadas com lubrificação e podem ser imersas em óleo.

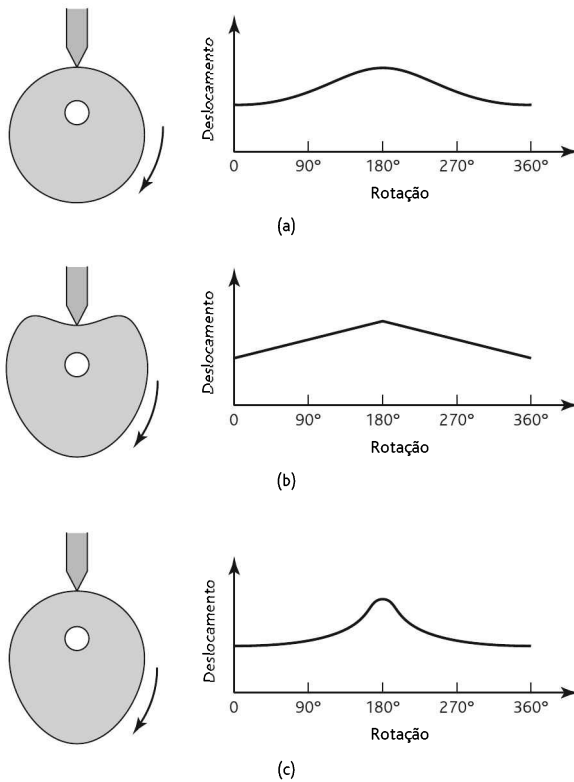


Figura 8.11 Cames: (a) excêntrico, (b) formato de coração, (c) formato de pera.

8.5 Engrenagens

Os **trens de engrenagens** são mecanismos bastante usados para transferir e transformar movimento rotacional. Eles são usados quando é necessário uma variação na velocidade ou no torque de um dispositivo. Por exemplo, a caixa de marchas de um carro permite ao motorista combinar a velocidade e o torque necessários para um tipo de terreno com a potência disponível do motor.

As engrenagens podem ser usadas para a transmissão de movimento de rotação entre eixos paralelos (Figura 8.13(a)) e para eixos inclinados entre si (Figura 8.13(b)).

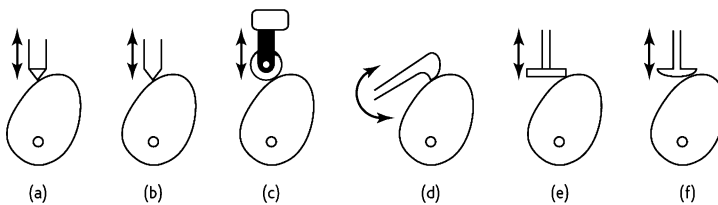


Figura 8.12 Seguidores de cames: (a) ponto, (b) faca, (c) rolete, (d) deslizante e oscilante, (e) plano, (f) cogumelo.

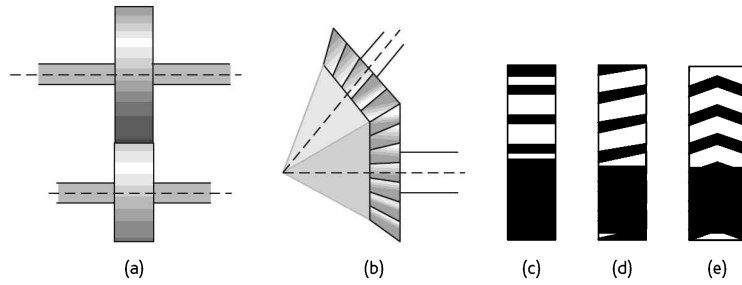


Figura 8.13 (a) Engrenagens de eixos paralelos, (b) eixos inclinados entre si, (c) dente axial, (d) dente helicoidal, (e) dente de dupla hélice.

O termo **engrenagens cônicas** é usado quando as linhas dos eixos se interceptam, como ilustrado na Figura 8.13(b). Quando duas engrenagens estão acopladas, a roda de engrenagem maior é frequentemente denominada **roda dentada** ou **coroa dentada** e a menor, **pinhão**. As engrenagens para uso com eixos paralelos podem ter dente axial com o corte do dente ao longo da linha axial em paralelo ao eixo (Figura 8.13(c)). Estas são denominadas **engrenagens dentadas**. Alternativamente, elas podem ter dente helicoidal com cortes na forma de hélice (Figura 8.13(d)) e ser denominadas **engrenagens helicoidais**. Estas engrenagens têm a vantagem de ter um endentamento gradual de qualquer dente individual e, conseqüentemente, há um acionamento suave, com uma vida útil geralmente prolongada. Entretanto, a inclinação do dente em relação ao eixo resulta em uma componente de força axial no rolamento do eixo. Isto pode ser superado usando-se um dente de dupla hélice (Figura 8.13(e)).

Considere duas rodas de engrenagens A e B acopladas (Figura 8.14). Se a roda A tiver 40 dentes e a B tiver 80, então a roda A gira duas voltas no mesmo tempo em que a B gira uma. A velocidade angular ω_A da roda A tem que ser duas vezes a ω_B da roda B, ou seja,

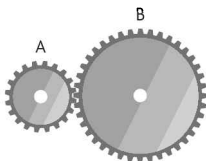


Figura 8.14 Duas engrenagens acopladas.

$$\frac{\omega_A}{\omega_B} = \frac{\text{Número de dentes em B}}{\text{Número de dentes em A}} = \frac{80}{40} = 2$$

Visto que o número de dentes na roda é proporcional ao diâmetro, podemos escrever

$$\frac{\omega_A}{\omega_B} = \frac{\text{Número de dentes em B}}{\text{Número de dentes em A}} = \frac{d_B}{d_A}$$

Portanto, para os dados que temos considerado, a roda B tem que ter duas vezes o diâmetro de A. O termo **relação de engrenagens** é usado para a relação das velocidades angulares de um par de engrenagens acopladas. A relação de engrenagens para este exemplo é 2.

8.5.1 Trem de engrenagens

O termo trem de engrenagens é usado para descrever uma série de engrenagens acopladas. O termo **trem de engrenagens simples** é usado para um sistema em

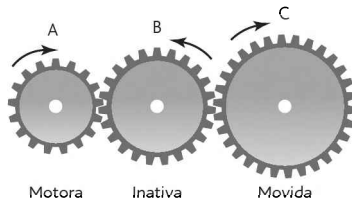


Figura 8.15 Trem de engrenagens simples.

que cada eixo tem apenas uma engrenagem, como na Figura 8.15. Para este trem de engrenagens, a relação de engrenagens geral é a relação de velocidades angulares nos eixos de entrada e saída e é ω_A/ω_C , ou seja,

$$G = \frac{\omega_A}{\omega_C}$$

Considere um trem de engrenagens simples que consiste das rodas A, B e C, como na Figura 8.15, com A tendo 9 e C tendo 27 dentes. Então, como a velocidade angular de uma roda é inversamente proporcional ao número de dentes na mesma, a relação de engrenagem é $27/9 = 3$. O efeito da roda B é puramente mudar o sentido de rotação da roda de saída em comparação com o que aconteceria se houvesse apenas as rodas A e C acopladas. A roda intermediária, B, é denominada **roda inativa**.

Podemos reescrever esta equação para a relação de engrenagens geral G como

$$G = \frac{\omega_A}{\omega_C} = \frac{\omega_A}{\omega_B} \times \frac{\omega_B}{\omega_C}$$

Mas ω_A/ω_B é a relação de engrenagens para o primeiro par e ω_B/ω_C é para o segundo par. A relação geral para um trem de engrenagens simples é o produto das relações de engrenagens para cada par sucessivo.

O termo **trem de engrenagens compostas** é usado para descrever um trem de engrenagens quando duas rodas são montadas em um eixo comum. A Figura 8.16(a) e (b) mostra dois exemplos de trem de engrenagens compostas. O da Figura 8.16(b) permite que os eixos de entrada e saída estejam em linha.

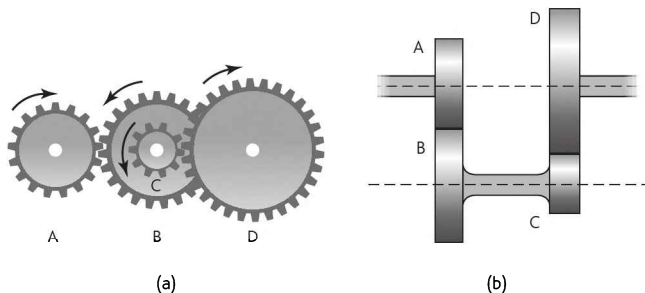


Figura 8.16 Trens de engrenagens compostas.

Quando duas rodas de engrenagens são montadas no mesmo eixo, elas têm a mesma velocidade angular. Portanto, para os dois trens de engrenagens compostas da Figura 8.16, $\omega_B = \omega_C$. Então, a relação de engrenagens geral é

$$G = \frac{\omega_A}{\omega_D} = \frac{\omega_A}{\omega_B} \times \frac{\omega_B}{\omega_C} \times \frac{\omega_C}{\omega_D} = \frac{\omega_A}{\omega_B} \times \frac{\omega_C}{\omega_D}$$

Para o arranjo mostrado na Figura 8.16(b), para que os eixos de entrada e saída estejam em linha, temos que ter também os raios das engrenagens

$$r_A + r_B = r_D + r_C$$

Considere um trem de engrenagens compostas da forma mostrada na Figura 8.16(a), com A, a primeira propulsora, tendo 15 dentes, B tendo 30, C tendo 18 e D, a roda final acionada, tendo 36 dentes. Visto que a velocidade angular de uma roda é inversamente proporcional ao número de dentes da roda, a relação de engrenagens geral é

$$G = \frac{30}{15} \times \frac{36}{18} = 4$$

Portanto, se a entrada na roda A tem uma velocidade angular de 160 rev/min, então a velocidade angular de saída da roda D é $160/4 = 40$ rev/min.

Um trem de engrenagens simples com dente reto, helicoidal ou engrenagens cônicas é geralmente limitado a uma relação de engrenagens geral de aproximadamente 10. Isto por causa da necessidade de manter o trem de engrenagens em um tamanho gerenciável se o número de dentes no pinhão for mantido acima de um número mínimo que geralmente está entre 10 e 20. Entretanto, relações de engrenagens maiores podem ser obtidas com trens de engrenagens compostas. Isto porque a relação de engrenagens é o produto das relações de engrenagens individuais de conjunto de engrenagens paralelas.

8.5.2 Transformação do movimento rotacional em translacional

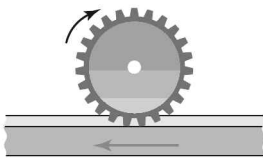


Figura 8.17 Cremalheira e pinhão.

A **cremalheira e pinhão** (Figura 8.17) é uma outra forma de engrenagem em que, essencialmente, há duas engrenagens acopladas, tendo uma delas raio infinito. Estas engrenagens podem ser usadas para transformar um movimento linear em rotacional e vice-versa.

Outro método usado para conversão de movimento rotacional em translacional é o sistema **porca-parafuso**. Com a forma convencional de porca e parafuso, a porca é girada e se move ao longo de um parafuso estacionário. Entretanto, se o parafuso girar, então a porca, que é fixada na parte acionada, se move ao longo do parafuso. Este arranjo é denominado de **parafuso de avanço**. Entretanto, existem problemas com o uso deste arranjo para converter um movimento rotacional em linear: as altas forças de fricção envolvidas e a falta de rigidez. Estes problemas podem ser superados usando-se um **fuso de esferas**. Este arranjo é usado em robôs em que o braço é movido por um fuso de esferas acionado por um motor CC com engrenagens (Figura 8.18). O motor gira o fuso, que move a porca para cima ou para baixo no fio da rosca. O movimento da porca é transmitido para o braço através de um sistema de articulações.

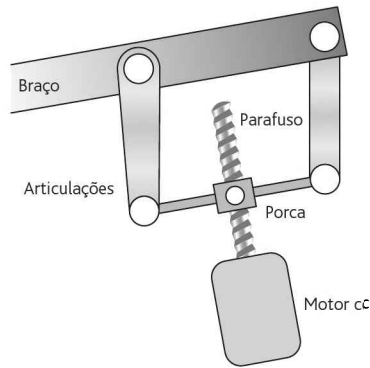


Figura 8.18 Fuso de esferas e sistema de articulações usado para mover o braço de um robô.

8.6 Catraca e lingueta

Catracas podem ser usados para travar um mecanismo quando ele está segurando uma carga. A Figura 8.19 mostra uma catraca e lingueta. O mecanismo consiste de uma roda, denominada **catraca**, com dente de serra que engata com um braço denominado **lingueta**. O braço é articulado e pode se mover para trás e, ao se mover para frente, engata a roda. O formato dos dentes determina que a rotação seja apenas em um sentido. A rotação da roda de catraca no sentido horário é evitada pela lingueta e pode ocorrer apenas quando ela é elevada. A lingueta é normalmente acionada por mola para garantir que ela engate automaticamente no dente da catraca.

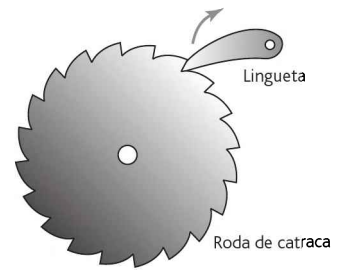


Figura 8.19 Catraca e lingueta.

8.7 Transmissão por correia e por corrente

As transmissões por correia são essencialmente um par de cilindros girantes em que o movimento de um cilindro é transferido para o outro através de uma correia (Figura 8.20). As transmissões por correia fazem uso da fricção que existe entre os tambores fixados nos eixos e as correias em torno do arco de contato para transmitir um torque. Como a transferência depende das forças de fricção, podem ocorrer escorregamentos. O torque transmitido é devido às diferenças na tensão que ocorrem na correia durante a operação. Esta diferença resulta em um lado tensionado da correia e um outro folgado. Se a tensão no lado tensionado for T_1 , e no lado folgado for T_2 , então, sendo o tambor A o propulsor, na Figura 8.20 temos

$$\text{torque em A} = (T_1 - T_2)r_A$$

em que r_A é o raio do tambor A. No tambor acionado B temos

$$\text{torque em B} = (T_1 - T_2)r_B$$

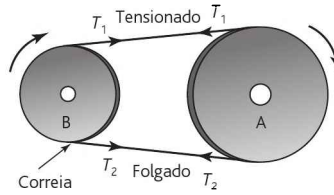


Figura 8.20 Transmissão por correia.

em que r_B é o raio do tambor B. Como a potência transmitida é o produto do torque pela velocidade angular, e como a velocidade angular é v/r_A para o tambor A e v/r_B para o tambor B, em que v é a velocidade da correia, então para qualquer tambor temos

$$\text{potência} = (T_1 - T_2)v$$

Como um método de transmitir potência entre dois eixos, a transmissão por correia tem a vantagem do comprimento da correia poder ser facilmente ajustado para corresponder a uma ampla gama de distâncias entre eixos e o sistema é automaticamente protegido contra sobrecarga porque ocorrem escorregamentos se a carga exceder a tensão máxima que pode ser mantida por forças de fricção. Se as distâncias entre eixos for grande, a transmissão por correia é mais indicada do que por engrenagens, porém para pequenas distâncias as engrenagens são mais indicadas. Podem ser usados tambores de tamanhos diferentes para proporcionar o efeito engrenagem. Entretanto, a relação de engrenagem é limitada a aproximadamente 3 por causa da necessidade de se manter um arco adequado ao contato entre a correia e os tambores.

A transmissão por correia mostrada na Figura 8.20 proporciona uma rotação da roda acionada na mesma direção da roda propulsora. A Figura 8.21 mostra dois tipos de transmissões invertidas. Nestes dois tipos de transmissão, os dois lados da correia fazem contato com as rodas e, desta forma, as correias em V, ou dentadas, não podem ser usadas.

8.7.1 Tipos de correias

Os quatro tipos principais de correias (Figura 8.22) são:

1 *Plana*

Esta correia tem uma seção reta retangular. A transmissão com este tipo tem uma eficiência em torno de 98% e produz pouco ruído. Ela pode transmitir potência através de distâncias longas entre os centros dos tambores. Tambores com coroa são usados para manter as correias presas aos tambores.

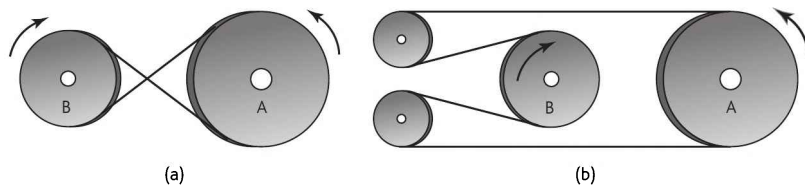


Figura 8.21 Transmissões por correia invertida: (a) correia cruzada, (b) correia aberta.

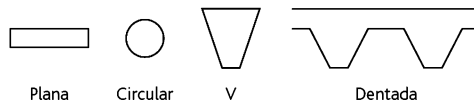


Figura 8.22 Tipos de correias.

2 Circular

Esta correia tem uma seção reta circular e é usada com tambores com ranhura.

3 Em V

As correias em V (trapezoidais) são usadas em tambores com ranhuras e são menos eficientes do que as planas, mas algumas delas podem ser usadas em uma única roda e proporcionar uma transmissão múltipla.

4 Dentada

As correias dentadas necessitam de rodas dentadas, cujos dentes se encaixam nas ranhuras das rodas. Esta correia, diferentemente das outras, não estica nem escorrega e, conseqüentemente, transmite potência a uma razão de velocidade angular constante. Os dentes tornam possível para a correia se movimentar em velocidades baixas ou altas.

8.7.2 Correntes

O escorregamento pode ser evitado usando-se correntes que travam nos dentes localizados nos cilindros girantes, o que equivale a um par de rodas com engrenagens acopladas. O mecanismo de transmissão usado em uma bicicleta é um exemplo de transmissão a corrente. As correntes permitem que eixos sejam acionados por uma simples roda que fornece uma transmissão múltipla. Elas não são tão silenciosas quanto as correias dentadas, mas podem ser usadas para torques maiores.

8.8 Rolamentos

Sempre que há um movimento relativo de uma superfície em contato com outra, por rotação ou por deslizamento, as forças de fricção resultantes geram calor, com desperdício de energia e desgaste. A função de um **rolamento** é guiar, com um mínimo de fricção e um máximo de precisão, o movimento de uma parte em relação à outra.

De importância particular é a necessidade de se conferir suporte adequado para a rotação dos eixos, ou seja, suporte a cargas radiais. O termo **rolamento de tração axial** é usado para rolamentos que são projetados para resistir a forças ao longo de um eixo quando o movimento relativo é principalmente de rotação. As seções a seguir resumem as características dos tipos mais comuns de rolamentos.

8.8.1 Rolamentos radiais planos

Os rolamentos radiais são usados para suportar eixos em rotação com carga na direção radial. O rolamento consiste basicamente na inserção de um material ade-

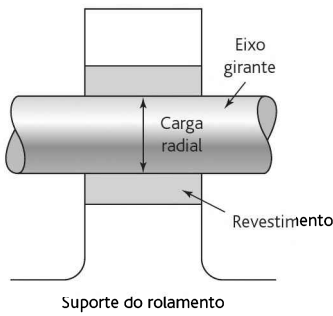


Figura 8.23 Rolamento radial plano.

quado que é colocado entre o eixo e o suporte (Figura 8.23). A rotação do eixo resulta no deslizamento de sua superfície sobre a superfície do rolamento. A inserção pode ser um metal branco, liga de alumínio, liga de cobre, bronze ou um polímero como o nylon ou PTFE. A inserção de um destes materiais fornece baixa fricção e menor desgaste do que se o eixo girasse apenas dentro de um furo no suporte. O rolamento pode ser seco ou lubrificado. Plásticos como o nylon e o PTFE são usados geralmente sem lubrificação, pois o coeficiente de fricção deles é extremamente baixo. Um outro material bastante usado é o bronze sinterizado: este bronze tem uma estrutura de poros que lhe permite ser impregnado por óleo, e assim o rolamento passa a ter uma lubrificação embutida.

O lubrificante pode ser:

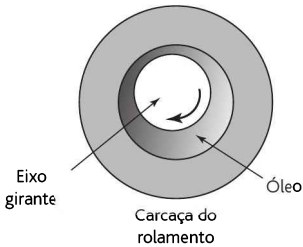


Figura 8.24 Rolamento radial hidrodinâmico.

1 Hidrodinâmico

O **rolamento radial hidrodinâmico** consiste na rotação contínua do eixo continuamente em óleo de forma que ele desliza sobre o óleo sem contato direto com o metal (Figura 8.24). A carga é suportada pela pressão gerada no óleo como resultado da rotação do eixo.

2 Hidrostático

Um problema da lubrificação hidrodinâmica é que o eixo desliza no óleo apenas quando está girando, e quando está em repouso há o contato de metal com metal. Para evitar desgaste excessivo na partida e quando há uma carga baixa, o óleo é bombeado na área do rolamento que recebe o esforço da carga em um pressão suficiente para elevar o eixo, separando-o do metal no repouso.

3 Filme sólido

Esta é uma cobertura de um material sólido como grafite ou dissulfeto de molibdênio.

4 Camada parcial

Esta é uma camada fina de lubrificante que adere à superfície do rolamento.

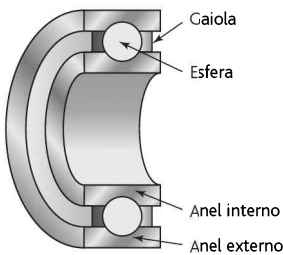


Figura 8.25 Elementos básicos de um rolamento de esferas.

8.8.2 Rolamentos de esferas e roletes

Neste tipo de rolamento, a carga principal é transferida do eixo girante para o seu suporte através de uma superfície de rolamento em vez de um contato deslizante. Um rolamento consiste de quatro elementos principais: anel interno, anel externo, elemento de rolamento (esferas ou roletes) e uma gaiola para manter os elementos do rolamento separados (Figura 8.25). Os anéis interno e externo contêm trilhas rígidas nas quais os elementos de rolamento rodam.

Existem alguns tipos de rolamentos de esferas:

1 Rolamento rígido de esferas (Figura 8.26(a))

Este tipo é bom para resistir a cargas radiais, porém é apenas moderadamente bom para cargas axiais. É um rolamento versátil que pode ser usado com uma ampla faixa de carga e velocidade.

2 Entalhado (Figura 8.26(b))

Este é capaz de resistir a cargas radiais maiores do que o tipo rolamento rígido de esferas equivalente, porém não pode ser usado quando há cargas axiais.

3 Contato angular (Figura 8.26(c))

Este tipo é bom para ser usado com cargas radial e axial e é melhor para cargas axiais do que o rolamento rígido de esferas equivalente.

4 Via dupla (Figura 8.26(d))

Os rolamentos de esferas de via dupla têm diversas variações e são capazes de resistir a cargas radial e axial maiores do que os equivalentes de via única. A figura mostra um rolamento rígido de esfera de via dupla, sendo que existem versões de dupla via de cada um dos tipos de via única.

5 Autoalinhado (Figura 8.26(e))

Os rolamentos de via única podem resistir a um pequeno desalinhamento de eixo, porém, onde o desalinhamento for severo, um rolamento autoalinhado é usado. Ele é capaz de resistir apenas a cargas radiais moderadas, não suportando cargas axiais.

6 Anel ranhurado (Figura 8.26(f))

Este tipo é projetado para resistir a cargas axiais, mas não é adequado para cargas radiais.

Existem também diversas formas de rolamentos de rolete. A seguir podemos ver os exemplos mais comuns:

1 Rolete cilíndrico (Figura 8.27(a))

Este é melhor para cargas radiais em comparação com rolamentos de esfera equivalentes, mas geralmente não é adequado para cargas axiais. Eles podem suportar uma carga maior do que os rolamentos de esfera de mesmo tamanho porque eles têm uma área de contato maior. Entretanto, eles não toleram desalinhamento.

2 Rolete cônico (Figura 8.27(b))

Este é bom para cargas radiais e bom para cargas axiais em uma direção.

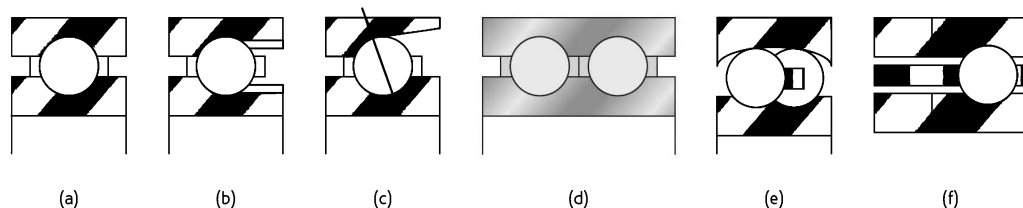


Figura 8.26 Tipos de rolamentos de esfera.

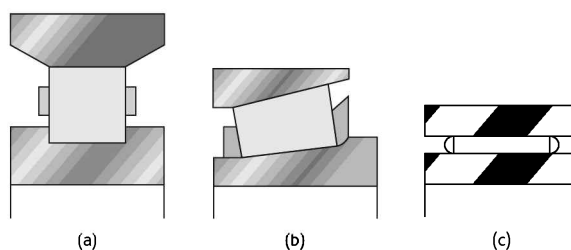


Figura 8.27 Rolamentos de rolete.

3 Rolete tipo agulha (Figura 8.27(c))

Este tem um rolete com uma alta relação comprimento/diâmetro e tende a ser usado em situações onde o espaço é insuficiente para o tipo de rolamento equivalente de esfera ou de rolete.

8.8.3 Seleção de rolamentos

Em geral, os rolamentos de deslizamento seco tendem a ser mais usados apenas para eixos com diâmetros menores e com cargas e velocidades menores; os de esferas e roletes, ou seja, os rolamentos que envolvem rolagem, são usados com uma faixa de diâmetros de eixos muito maiores e cargas e velocidades maiores; e os hidrodinâmicos são usados para altas cargas com diâmetros de eixo grandes.

8.9 Aspectos mecânicos da seleção de motores

Um sistema de acionamento por motor é mecanicamente necessário para girar um eixo com a carga acoplada a ele. Os fatores a serem considerados são o momento de inércia e o torque.

8.9.1 Momento de inércia

O torque necessário para fornecer uma aceleração α a uma carga com momento de inércia I_L é $I_L\alpha$. O torque necessário para acelerar o eixo do motor é $T_M = I_M\alpha_M$ e para acelerar a carga é $T_L = I_L\alpha_L$. O eixo do motor terá, na ausência de engrenagens, a mesma velocidade e aceleração angulares. A potência necessária para acelerar o sistema como um todo é $T_M\omega + T_L\omega$, em que ω é a velocidade angular. Portanto,

$$\text{potência} = (I_M + I_L) \alpha \omega$$

Esta potência é produzida pelo torque do motor, T_M , e assim a potência tem que ser igual a $T_M\omega$. Portanto,

$$T = (I_M + I_L)\alpha$$

O torque para se obter uma determinada aceleração angular será minimizado quando $I_M = I_L$. Então, para o melhor desempenho, o momento de inércia da carga deve ser similar ao do motor.

Considere um sistema de engrenagens com o eixo do motor girando em uma velocidade diferente do eixo de rotação da carga. A relação de engrenagens $G = \omega_L/\omega_M = \alpha_L/\alpha_M$, em que ω_L é a velocidade angular da carga, ω_M é a velocidade angular do motor, α_L é a aceleração angular da carga e α_M a aceleração angular do motor. O eixo da carga terá uma aceleração angular de $\alpha_L = G\alpha_M$. O torque necessário para acelerar o eixo do motor é $T_M = I_M\alpha_M$ e o necessário para acelerar a carga é $T_L = I_L\alpha_L$. A potência necessária para acelerar o sistema como um todo é $T_M\omega_M + T_L\omega_L$, em que ω é a velocidade angular. Porém, $G = \omega_L/\omega_M$, assim a potência é

$$\text{potência} = (I_M + G^2 I_L)\alpha_M\omega_M$$

Esta potência é produzido pelo torque do motor, T_M , e a potência tem que ser igual a $T_M\omega_M$. Portanto,

$$T_M = (I_M + G^2 I_L)\alpha_M$$

O efeito do uso de engrenagens é fornecer à carga um momento de inércia efetivo de $G^2 I_L$. O torque para fornecer uma determinada aceleração angular será minimizado quando $I_M = G^2 I_L$.

8.9.2 Torque

A Figura 8.28 mostra as curvas de operação para um motor típico. Em operação contínua, o valor do torque de parada não deve ser excedido. Este é o valor de torque máximo no qual não ocorre sobreaquecimento. No caso de uso intermitente, são possíveis torques maiores. Conforme a velocidade angular aumenta, a capacidade de torque do motor diminui. Portanto, se forem necessários velocidades e torques maiores, então precisa ser especificado um motor de potência maior.

Resumo

Os **mecanismos** são dispositivos que podem ser considerados conversores de movimento, pois eles transformam o movimento de uma forma em outra desejada.

O movimento de um corpo pode ser considerado uma combinação de movimentos de translação e rotação. O número de **graus de liberdade** é o número de componentes do movimento que são necessários para gerá-lo.

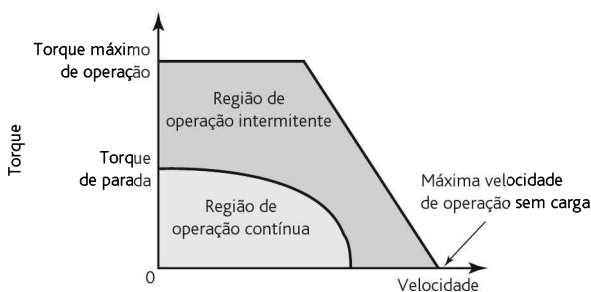


Figura 8.28 Gráfico de torque versus velocidade.

Cada parte de um mecanismo que tem movimento relativo a uma outra parte é denominado **articulação**. Os pontos de fixação de uma articulação em suas articulações vizinhas são denominados **nós**. Uma **junção** é uma conexão entre duas ou mais articulações em seus nós. Uma sequência de junções e articulações é denominada **cadeia cinemática**. Uma **cadeia de quatro barras** consiste de quatro articulações conectadas para formar quatro junções em torno das quais pode-se girar.

Uma **came** é um corpo que gira ou oscila e, ao fazer isto, comunica um movimento de alternância ou oscilatório para um segundo corpo, denominado **seguidor**, com o qual ela faz contato.

Engrenagens podem ser usadas para a transmissão de movimento de rotação entre eixos paralelos e inclinados entre si.

Os sistemas de **cremalheira e pinhão** e **porca e parafuso** podem ser usados para converter um movimento rotacional em translacional.

As **catracas** podem ser usadas para travar um mecanismo quando ele está segurando uma carga.

As **transmissões por correia** e **por corrente** podem ser usadas para transmitir movimento de rotação entre eixos em paralelo afastados entre si uma certa distância.

Os **rolamentos** são usadas para guiar, com o mínimo de fricção e o máximo de precisão, os movimentos de uma parte em relação a outra.

Considerando um **sistema de acionamento com motor**, deve-se levar em conta o momento de inércia e o torque.

Problemas

- 8.1 Explique os termos (a) mecanismo, (b) cadeia cinemática.
- 8.2 Explique o significado de cadeia de quatro barras.
- 8.3 Examinando os seguintes mecanismos, determine o número de graus de liberdade de cada um.
 - (a) Mecanismo de articulação do capô de um carro.
 - (b) Mecanismo da porta traseira de carro com carroceria.
 - (c) Mecanismo do limpador de para-brisa de um carro.
 - (d) Seu joelho.
 - (e) Seu tornozelo.
- 8.4 Analise os movimentos dos seguintes mecanismos e determine se eles envolvem somente rotação, somente translação ou uma mistura destes dois componentes.
 - (a) As teclas de um teclado de computador.
 - (b) A caneta em um ploter XY.
 - (c) A nossa mão ao ver as horas em um relógio de pulso.
 - (d) O ponteiro de um amperímetro de bobina móvel.
 - (e) Uma aparafusadeira elétrica.
- 8.5 Para o mecanismo mostrado na Figura 8.29, o braço AB gira em uma taxa constante. B e F são cursores que se movem ao longo de CD e AF. Descreva o comportamento deste mecanismo.

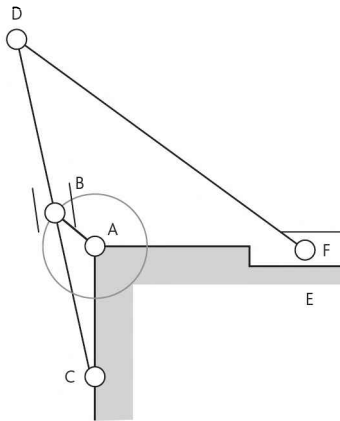


Figura 8.29 Problema 8.5.

- 8.6 Descreva como o deslocamento do seguidor de came mostrado na Figura 8.30 varia com o ângulo de rotação da came.
- 8.7 Um came circular de diâmetro 100mm tem um eixo de rotação excêntrico deslocado 30mm do centro. Quando utilizado com um seguidor tipo faca com sua linha de ação passando pelo centro de rotação, qual é a diferença entre os deslocamentos máximo e mínimo do seguidor?
- 8.8 Projete um sistema seguidor de came para gerar velocidades de seguidores constantes ao longo de deslocamentos do seguidor que variam de 40 a 100 mm.
- 8.9 Projete um sistema mecânico que possa ser usado para:
 - (a) Operar uma sequência de chaves fim de curso em uma sequência temporal.
 - (b) Mover uma ferramenta em uma taxa estacionária em um sentido e, em seguida, se mover de volta ao início do percurso.
 - (c) Transformar uma rotação em um movimento linear de vai e vem com um movimento harmônico simples.
 - (d) Transformar uma rotação de um certo ângulo em deslocamento linear.
 - (e) Transformar uma rotação de um eixo em rotação de outro que esteja em paralelo afastado de uma certa distância.
 - (f) Transformar uma rotação de um eixo em rotação de outro próximo que forma com ele um ângulo reto.
- 8.10 Um trem de engrenagens composto consiste de uma roda acionada final com 15 dentes que se acopla a uma segunda roda com 90 dentes. No mesmo eixo há uma segunda roda com 15 dentes. Esta se acopla a uma quarta roda, a propulsora, com 60 dentes. Qual é a relação de engrenagens geral?

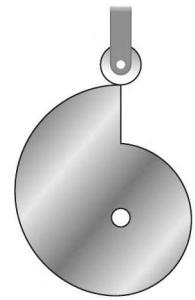


Figura 8.30 Problema 8.6.

9

Sistemas de Acionamento Elétrico

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Avaliar as características operacionais dos sistemas de acionamento elétrico: relés, chaves de estado sólido (tiristores, transistores bipolar e MOSFETs), solenoides, motores CC, motores CA e motores de passo.
- Explicar os princípios dos motores CC, incluindo os de ímã permanente, e como sua velocidade pode ser controlada.
- Explicar o princípio do motor CC de ímã permanente sem escovas.
- Explicar os princípios dos motores de passo de relutância variável, ímã permanente e híbrido, e como a sequência de passos pode ser gerada.

9.1 Sistemas elétricos

Qualquer estudo de sistemas elétricos usados como atuadores para controle deve incluir:

- 1 *Dispositivos de comutação* como chaves mecânicas, como por exemplo, relés, e chaves de estado sólido, entre as quais temos os diodos, tiristores e transistores, nos quais o sinal de controle liga ou desliga algum dispositivo elétrico, como, por exemplo, um aquecedor ou um motor.
- 2 *Dispositivos do tipo solenoide* nos quais a corrente através de um solenoide é usada para acionar um núcleo de ferro doce, como, por exemplo, em uma válvula hidráulica/pneumática acionada por solenoide no qual o controle da corrente no solenoide é usado para atuar na vazão.
- 3 *Sistemas de acionamento*, como, por exemplo, motores CC e CA, onde a corrente é usada para produzir rotação.

Este capítulo dá uma visão geral destes dispositivos e suas características.

9.2 Chaves mecânicas

As chaves mecânicas são elementos frequentemente utilizados como sensores de entrada para sistemas (veja a Seção 2.12), como, por exemplo, teclados. Neste capítulo consideramos o seu uso como atuadores que ligam, por exemplo, aquecedores ou motores elétricos, ou ainda ligam a corrente em válvulas de solenoide que controlam cilindros hidráulicos ou pneumáticos. O **relé** elétrico é um exemplo de uma chave mecânica usada em sistemas de controle com um atuador.

9.2.1 Relés

Os relés são chaves operadas eletricamente em que uma variação na corrente em um circuito faz ligar ou desligar outro circuito. Para o relé mostrado na Figura 9.1(a), quando há corrente no solenoide do relé, é produzido um campo magnético que atrai a armadura de ferro, que por sua vez empurra a barra, fechando o contato normalmente aberto (NA) e abrindo o contato normalmente fechado (NF).

Os relés geralmente são utilizados em sistemas de controle. A saída de um controlador é uma corrente relativamente pequena e, desta forma, é geralmente usada para acionar um transistor que liga a corrente no solenoide do relé e, assim, usa o relé para ligar uma corrente muito maior que liga ou desliga o elemento final de correção, como um aquecedor elétrico, em um sistema de controle de temperatura, ou um motor. A Figura 9.1(b) mostra um tipo de circuito que pode ser usado. Em função dos relés serem indutâncias, eles podem gerar tensão reversa quando a corrente de energização é desligada ou quando a entrada deles comutam de nível alto para baixo. Como resultado, podem ocorrer danos ao circuito de conexão. Para superar este problema, é conectado um diodo na bobina do relé. Quando ocorrer a FEM contrária, o diodo conduz e coloca a bobina em curto-circuito. Este diodo é denominado *flyback* ou *roda livre*.

Como ilustração das formas em que um relé pode ser usado em sistemas de controle, a Figura 9.2 mostra como dois relés podem ser usados para controlar a operação de válvulas pneumáticas, que por sua vez controlam os movimentos de pistões em três cilindros, A, B e C. A sequência de operação é:

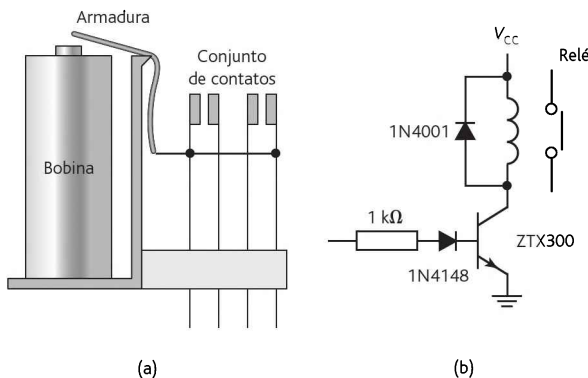


Figura 9.1 (a) Um relé e (b) circuito de acionamento.

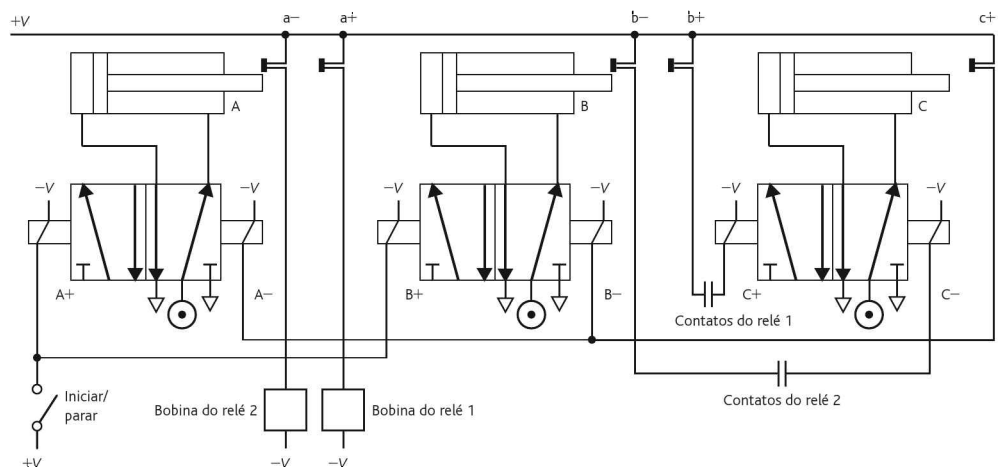


Figura 9.2 Sistema controlado por relé.

- 1 Quando a chave de partida é fechada, uma corrente é aplicada nos solenoides A e B, resultando no avanço de A e B, ou seja, A+ e B+.
- 2 As chaves fim de curso a+ e b+ são então fechadas; o fechamento de A+ resulta em uma corrente fluindo na bobina do relé 1, que então fecha os seus contatos e assim fornece corrente para o solenoide C, fazendo-o avançar, ou seja, C+.
- 3 Este avanço faz com que a chave fim de curso c+ feche e, assim, comute as válvulas de controle A e B, retraindo os cilindros A e B, ou seja, A- e B-.
- 4 Fechar a chave fim de curso a- faz passar uma corrente pela bobina do relé 2; seus contatos fecham, permitindo uma corrente na válvula C, e o cilindro C se retrai, ou seja, C-.

A sequência dada por este sistema é A+ e B+ concorrentemente, em seguida C+, depois A- e B- concorrentemente, e finalmente C-.

Os **relés de tempo** são relés de controle que têm uma ação de comutação atrasada. O atraso de tempo é geralmente ajustável e pode ser iniciado quando uma corrente passa pela bobina do relé ou quando ela cessa.

9.3 Chaves de estado sólido

Existem vários dispositivos de estado sólido que podem ser usados eletronicamente para comutar circuitos. Entre eles temos:

- 1 Diodos
- 2 Tiristores e triacs
- 3 Transistores bipolares
- 4 MOSFETs de potência

9.3.1 Diodos

O diodo tem as características mostradas na Figura 9.3(a), só permitindo a passagem de corrente quando polarizado diretamente, ou seja, com o anodo mais positivo em relação ao catodo. Se o diodo estiver suficientemente polarizado reversamente, ou seja, com uma tensão muito alta, ele pode se romper. Se uma tensão alternada for aplicada no diodo, ele pode ser considerado ligado quando a polaridade da tensão o torna polarizado diretamente, e desligado quando polarizado reversamente. O resultado é que a corrente no diodo é semirretificada, sendo constituída da metade positiva da tensão de entrada (Figura 9.3(b)), ou seja, o circuito liga apenas no semiciclo positivo.

9.3.2 Tiristores e triacs

O tiristor, ou retificador controlado de silício (SCR – *Silicon-Controlled Rectifier*), pode ser considerado como sendo um diodo com um gatilho que controla a condição na qual o diodo pode ser ligado. A Figura 9.4(a) mostra a curva característica de um tiristor. Com uma corrente de gatilho, ou porta, zero, pelo tiristor passa uma corrente insignificante quando polarizado reversamente (a menos que haja uma polarização reversa de centenas de volts, podendo ocorrer a ruptura dele). Quando polarizado diretamente, a corrente também é insignificante até que a tensão de ruptura seja atingida. Quando isso ocorre, a tensão no diodo cai para um nível baixo, aproximadamente de 1 a 2 V, e a corrente é limitada apenas pela resistência externa do circuito. Portanto, por exemplo, se a tensão de ruptura direta for de 300 V, então, quando esta tensão for atingida, o tiristor liga e a tensão nele cai para 1 ou 2 V. Se o tiristor estiver em série com uma resistência de, digamos, 20 Ω (Figura 9.4(b)), então antes da ruptura temos uma resistência muito alta em série com a de 20 Ω , e praticamente toda a tensão, 300 V, está sobre o tiristor e a corrente é desprezível. Quando a ruptura direta ocorre, a tensão no tiristor cai para, digamos, 2 V, e agora temos uma tensão de $300 - 2 = 298$ V no resistor de 20 Ω , elevando então a corrente para $298/20 = 14,9$ A. Uma vez ligado o tiristor, ele permanece nesta condição até que a corrente direta seja reduzida abaixo de um valor de alguns miliampere. A tensão na qual a ruptura direta ocorre é determinada pela corrente de porta: quanto maior essa corrente, menor a tensão de ruptura. A capacidade de potência de um tiristor

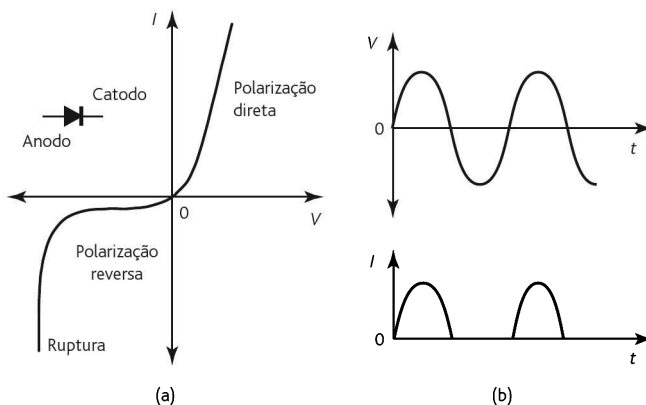


Figura 9.3 (a) Curva característica do diodo, (b) retificação de meia onda.

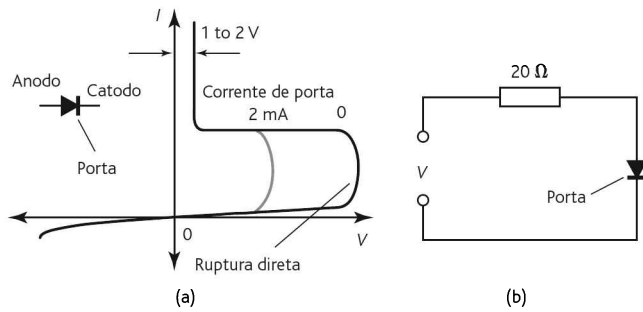


Figura 9.4 (a) Curva característica de um tiristor, (b) circuito com tiristor.

é alta, e por isso ele é muito usado em aplicações de alta potência. Como exemplo, o CF106D da Texas Instruments tem uma tensão máxima em estado desligado de 400 V e uma corrente máxima de porta de 0,2 mA.

O triac é similar ao tiristor e é equivalente a um par de tiristores conectados em antiparalelo no mesmo chip. O triac pode ser ligado direta ou reversamente. A Figura 9.5 mostra a sua curva característica. Como um exemplo, o triac MAC212-4 da Motorola tem uma tensão de estado desligado máxima de 200 V e uma corrente máxima em estado ligado de 12 A rms. Os triacs representam um método simples e relativamente barato de controle de potência ca.

A Figura 9.6 mostra o tipo de efeito que ocorre quando uma tensão alternada senoidal é aplicada (a) em um tiristor e (b) em um triac. A ruptura direta ocorre quando a tensão alcança o valor de ruptura, e a partir deste ponto a tensão no dispositivo permanece baixa.

Como um exemplo do uso destes dispositivos para fins de controle, a Figura 9.7 ilustra como um tiristor poderia ser usado para controlar uma tensão CC estacionária V . Neste caso o tiristor opera como uma chave, usando a porta para ligar o dispositivo. Usando uma técnica de desligamento do SCR, não ilustrada na figura, a tensão da fonte pode ser segmentada, produzindo uma tensão intermitente (*chop-*

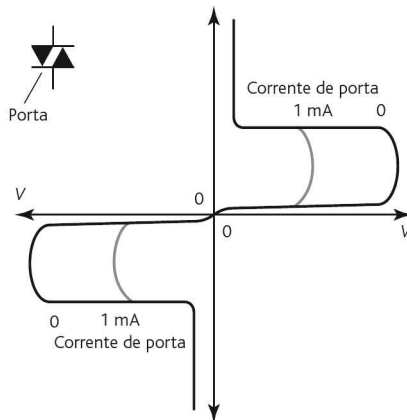


Figura 9.5 Curva característica de um triac.

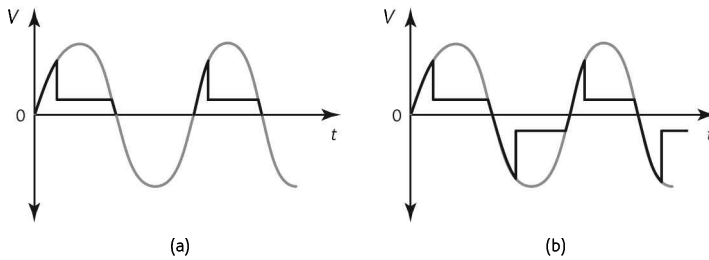


Figura 9.6 Controle de tensão: (a) tiristor, (b) triac.

per). O valor médio da tensão CC de saída pode ser controlado ao se variar o ciclo de trabalho desta forma de onda.

Outro exemplo de controle é o de corrente alternada em aquecedores elétricos, motores e *dimmers* (controle de luminosidade) para lâmpadas. A Figura 9.8(a) mostra um circuito de controle de fase de meia-onda e resistência variável. A corrente alternada é aplicada em uma carga, como, por exemplo, uma lâmpada em um circuito *dimmer*, em série com um tiristor. R_1 é um resistor de limitação de corrente e R_2 é um potenciômetro que ajusta o nível no qual o tiristor é disparado. O diodo evita que o ciclo negativo da tensão seja aplicado na porta. Pelo ajuste de R_2 o disparo do tiristor pode ser entre 0° e 90° do semiciclo positivo da tensão alternada aplicada. Quando o tiristor é disparado próximo ao início do ciclo, ou seja, 0° , ele conduz por todo o semiciclo positivo e a potência máxima é fornecida à carga. Conforme o disparo do tiristor é retardado no ciclo, a potência fornecida à carga é reduzida.

Quando uma tensão de alimentação é aplicada repentinamente em um tiristor, ou triac, com a porta desligada, o tiristor pode passar do estado desligado para o ligado. Uma taxa de variação de tensão típica que produz este efeito é da ordem de $50 \text{ V}/\mu\text{s}$. Se a fonte for uma tensão CC, o tiristor pode permanecer no estado de condução até que haja uma interrupção do circuito. Para evitar esta variação rápida na tensão de alimentação que produz este efeito, a taxa na qual a tensão varia no tempo, ou seja, dV/dt , é controlada com o uso de um **circuito snubber** (circuito de amortecimento). Este é constituído de um resistor em série com um capacitor e é colocado em paralelo com o tiristor (Figura 9.8(b)).

9.3.3 Transistores bipolares

Os **transistores bipolares** são de dois tipos, npn e pnp. A Figura 9.9(a) mostra o símbolo para cada um. No caso do transistor npn, a corrente principal entra no

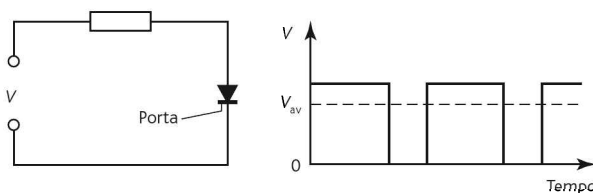


Figura 9.7 Controle CC com tiristor.

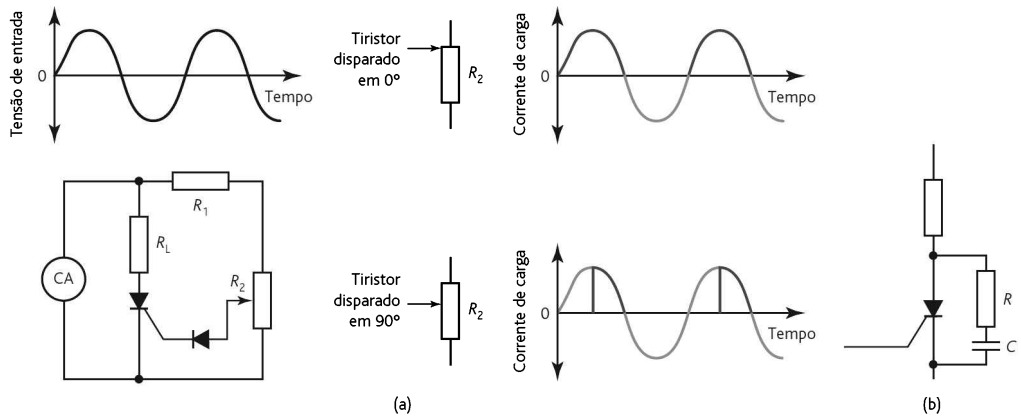


Figura 9.8 (a) Controle de fase, (b) circuito snubber.

coletor e sai no emissor (sentido convencional), com um sinal de controle aplicado na base. O transistor pnp tem uma corrente principal entrando no emissor e saindo no coletor, com um sinal de controle aplicado na base.

O transistor npn conectado como mostra a Figura 9.9(b), denominado circuito emissor comum, a relação entre a corrente de coletor I_C e a diferença de potencial entre coletor e emissor, V_{CE} , é descrita por uma série de curvas mostradas na Figura 9.9(c). Quando a corrente de base, I_B , é zero, o transistor está em corte; neste estado, as junções base-emissor e base-coletor estão polarizadas reversamente. Quando se

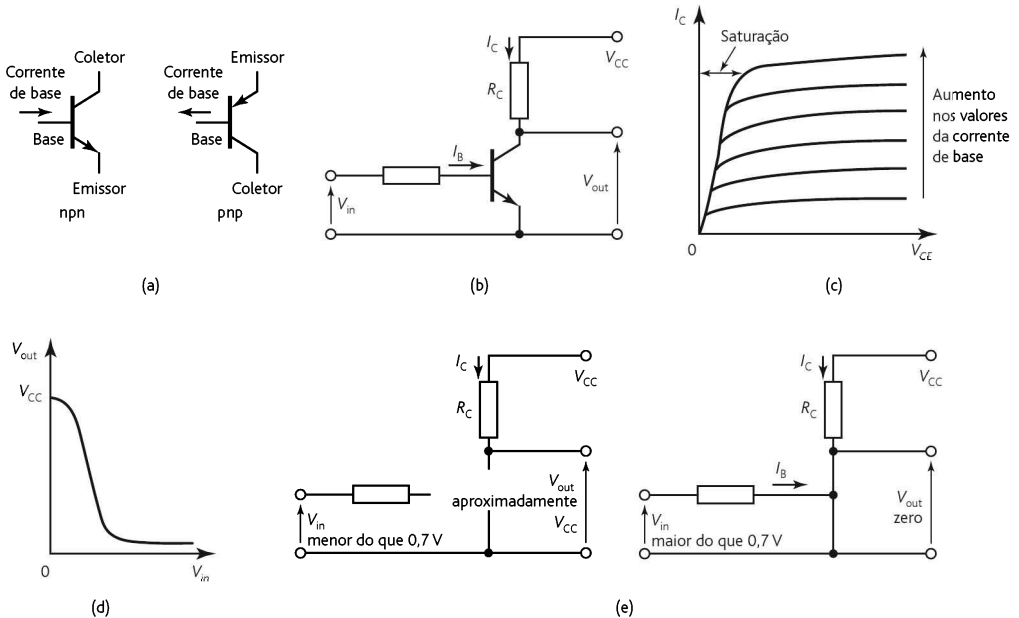


Figura 9.9 (a) Símbolos do transistor, (b), (c), (d), (e) chave com transistor.

aumenta a corrente de base, a corrente de coletor aumenta e V_{CE} diminui como resultado de uma maior queda de tensão em R_C . Quando V_{CE} alcança o valor de $V_{CE(sat)}$, a junção base-coletor torna-se polarizada diretamente e a corrente de coletor não aumenta mais, mesmo que a corrente de base tenha mais aumento. Isto é denominado **saturação**. Comutando-se a corrente de base entre 0 e um valor que coloca o transistor bipolar na saturação, este pode ser usado com chave. Quando não há tensão de entrada, V_{in} , então praticamente toda a tensão V_{CC} aparece na saída (Figura 9.9(d)). A Figura 9.9(e) resume o comportamento de chave de um transistor típico.

A relação entre a corrente de coletor e a corrente de base, I_B , para os valores abaixo da condição de saturação do transistor é

$$I_C = h_{FE} I_B$$

em que h_{FE} é o **ganho de corrente**. Na saturação, a corrente de coletor, $I_{C(sat)}$, é

$$I_{C(sat)} = \frac{V_{CC} - V_{CE(sat)}}{R_C}$$

Para garantir que o transistor seja colocado na saturação, a corrente de base tem que ser pelo menos

$$I_{B(sat)} = \frac{I_{C(sat)}}{h_{FE}}$$

Portanto, para um transistor com h_{FE} de 50 e $V_{CE(sat)}$ de 1 V, então, em um circuito com $R_C = 10 \Omega$ e $V_{CC} = 5 V$, a corrente de base tem que subir até pelo menos 8 mA.

Como a corrente de base necessária para acionar um transistor bipolar de potência é bastante grande, frequentemente é necessário usar um segundo transistor para permitir que a comutação seja obtida com correntes relativamente menores, por exemplo, como a que pode ser fornecida por um microprocessador. Portanto, o circuito de comutação pode ser como o que é mostrado na Figura 9.10(a). Uma combinação como esta de um par de transistores para permitir que correntes altas

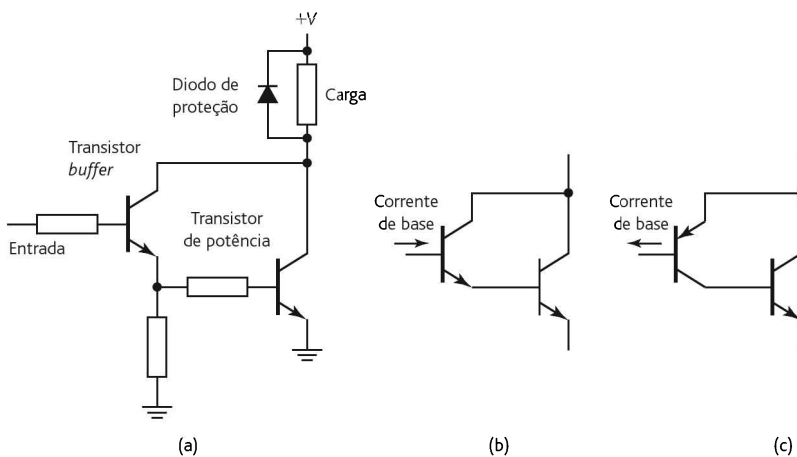


Figura 9.10 (a) Comutação de uma carga, (b) e (c) pares Darlington.

sejam comutadas por correntes pequenas é denominada **par Darlington**, sendo disponibilizada também em um único chip. Geralmente é conectado um **diodo de proteção** (diodo roda livre) em paralelo com a carga para evitar danos quando o transistor é desligado, visto que geralmente ele é usado com carga indutiva, podendo ocorrer tensões transientes grandes. O circuito integrado ULN2001N da SGS-Thomson contém sete pares Darlington separados, com cada par contendo um diodo de proteção. Cada par é especificado para uma corrente contínua de 500 mA e pode suportar surtos de até 600 mA.

A Figura 9.10(b) mostra as conexões Darlington quando um pequeno transistor é combinado com um transistor npn de maior capacidade, resultando em um transistor npn de grande capacidade com um grande fator de amplificação. A Figura 9.10(c) mostra as conexões Darlington para um pequeno transistor pnp com um transistor npn de maior capacidade, o que equivale a um único transistor de grande capacidade.

Ao usar acionadores transistorizados com um microprocessador, devemos dar atenção à amplitude da corrente de base necessária e o seu sentido. Esta corrente necessária pode ser muito alta e, desta forma, um *buffer* deve ser usado. O *buffer* aumenta a corrente de acionamento para o valor desejado. Ele também pode ser usado para inversão. A Figura 9.11 ilustra como um *buffer* pode ser usado quando uma chave transistorizada é usada para controlar, ligando e desligando, um motor CC. O *buffer* tipo 240 é inversor, enquanto que o 241 e 244 são não inversores. O *buffer* 74LS240 tem uma corrente de saída máxima em nível alto de 15 mA e uma corrente de saída em nível baixo de 24 mA.

A comutação com transistor bipolar é implementada por correntes de base e frequências de comutação maiores do que se consegue com tiristores. A capacidade de potência é um pouco menor em comparação com os tiristores.

9.3.4 MOSFETs

Os transistores de efeito de campo de óxido metálico, MOSFETs (*metal-oxide field-effect transistors*), são de dois tipos, o de canal n e o de canal p. A Figura 9.12(a) e (b) mostram os respectivos símbolos. A principal diferença entre o uso de um MOSFET e um transistor bipolar em comutação é que não há corrente na porta do

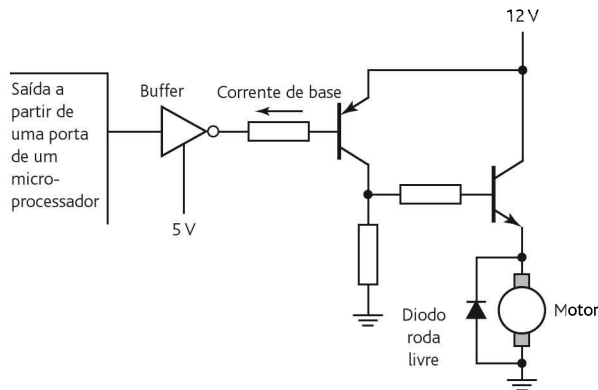


Figura 9.11 Controle de um motor CC.

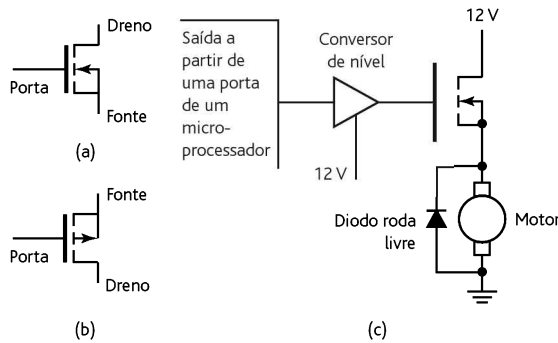


Figura 9.12 MOSFETs: (a) canal n, (b) canal p, (c) usado para controlar um motor CC.

MOSFET para exercer o controle. A tensão de porta é o sinal de controle. Portanto, o circuito de acionamento pode ser simplificado, pois não há a preocupação com a amplitude da corrente.

A Figura 9.12(c) ilustra o uso de um MOSFET como uma chave *on/off* para um motor; compare o circuito com o da Figura 9.11, que usa transistores bipolares. Um conversor de nível é indicado para elevar o nível de tensão ao valor necessário para o MOSFET.

Com o uso de MOSFETs é possível comutar frequências muito altas, até 1 MHz, e a interface com um microprocessador é mais simples em comparação com transistores bipolares.

9.4 Solenoides

Os solenoides podem ser usados como atuadores elétricos. As **válvulas solenoide** são exemplos destes dispositivos, sendo usadas para controlar vazão de fluido em sistemas hidráulicos e pneumáticos (veja a Figura 7.9). Quando uma corrente passa através de uma bobina, um núcleo de ferro doce é puxado para dentro da bobina e, desta forma, pode abrir ou fechar portas, permitindo ou impedindo a passagem de um fluido. A força exercida por um solenoide no núcleo é uma função da corrente na bobina e o comprimento do núcleo que está dentro da bobina. No caso das válvulas *on/off*, ou seja, as que usam controle direcional, a corrente na bobina é controlada ligando-se e desligando-se, fazendo com que o núcleo assuma uma de duas posições. No caso das válvulas de controle proporcional, a corrente na bobina é controlada de modo que o movimento do núcleo seja proporcional à amplitude da corrente.

9.5 Motores CC

Os motores elétricos são frequentemente usados como elemento final de controle em sistemas de controle de velocidade ou posição. Os motores podem ser classificados em duas categorias principais: CC e CA, sendo que os motores CC são os mais usados em sistemas de controle modernos. Estes motores podem ser divididos em dois

grupos principais: os que usam escovas para fazer contato com um anel comutador montado no rotor para comutar a corrente de uma bobina do rotor para a outra, e os que não usam escovas. No caso do motor de escovas, o rotor é constituído de uma bobina enrolada, e o estator pode ser um ímã permanente ou um eletroímã. No caso do motor sem escovas, o arranjo é invertido, sendo o rotor um ímã permanente e o estator constituído de uma bobina enrolada.

9.5.1 Motor CC do tipo escova

Um **motor CC do tipo escova** é essencialmente uma bobina de fio, que é denominada rotor, livre para girar no campo de um ímã permanente ou de um eletroímã, sendo este denominado estator, visto ser estacionário (Figura 9.13(a)). Quando uma corrente passa pela bobina, a força resultante age nas suas laterais em um ângulo de 90° com o campo, forçando a mesma a girar. Entretanto, para a rotação continuar, quando a bobina passa pela posição vertical, o sentido da corrente na bobina tem que ser invertido, e isso é obtido com o uso de escovas que fazem contato com o anel comutador dividido que gira com a bobina.

Em um motor CC convencional, as bobinas de fio são montadas nas ranhuras de um cilindro de material magnético denominado **armadura**. A armadura é montada em rolamentos e, portanto, é livre para girar. Esta fica dentro de um campo magnético produzido por **polos do campo**. Estes podem ser, no caso de motores pequenos, ímãs permanentes ou eletroímãs em que o magnetismo é produzido por uma corrente em **bobinas de campo**. A Figura 9.13(b) mostra o princípio básico de um motor CC de quatro polos com o campo magnético produzido por bobinas energizadas. As extremidades de cada bobina da armadura são conectadas em segmentos adjacentes de um anel, denominado comutador, com contatos elétricos feitos aos segmentos através de contatos de carbono denominados escovas. À medida que a armadura gira, o comutador inverte a corrente em cada bobina conforme elas se movem entre os polos do campo. Isto é necessário para que as forças que atuam na bobina permaneçam no mesmo sentido e a rotação continue. O sentido da rotação de um motor CC pode ser invertido invertendo-se a corrente na armadura ou a corrente de campo.

Considere um motor CC de ímã permanente com uma densidade de fluxo de valor constante. Para um condutor da armadura de comprimento L e percorrido por uma corrente i , a força resultante de uma densidade de fluxo magnético B em um ângulo de 90° com o condutor é BiL (Figura 9.14(a)). Com N condutores destes, a

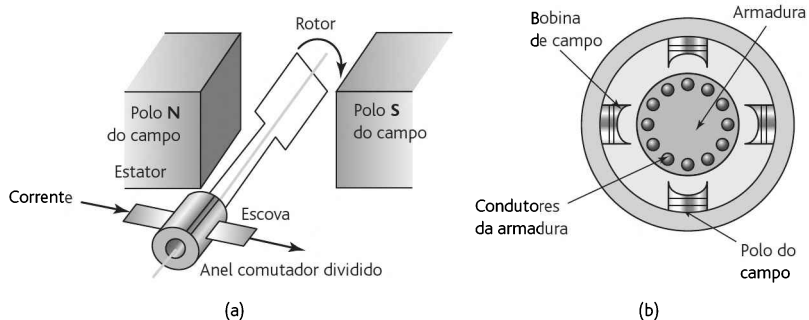


Figura 9.13 Motor CC: (a) básico, (b) com dois conjuntos de polos.

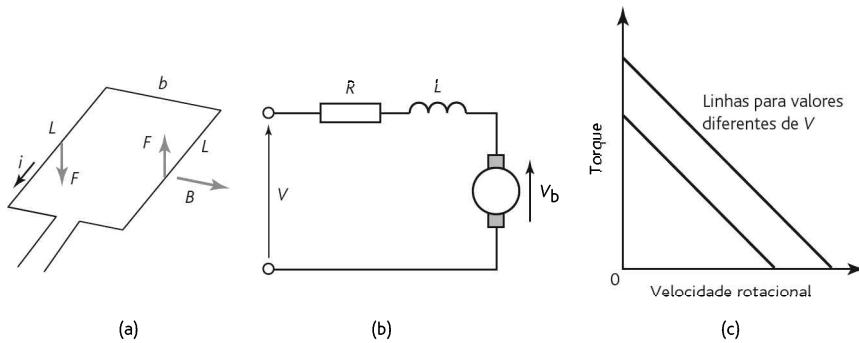


Figura 9.14 Motor CC: (a) forças na armadura, (b) circuito equivalente, (c) curvas características de torque-velocidade.

força é $NBiL$. A força resulta em um torque T no eixo da bobina de Fb , sendo b a largura da bobina. Portanto,

$$\text{torque } T = NBbLi = k_t i$$

em que k_t é o torque constante. Como a bobina da armadura gira no campo magnético, ocorre uma indução eletromagnética e uma FEM inversa é induzida (v_b). Esta é proporcional à taxa na qual o fluxo concatenado pela bobina varia e, portanto, para um campo magnético constante, é proporcional à velocidade angular ω da rotação. Portanto,

$$\text{velocidade rotacional } v_b = k_v \omega$$

em que k_v é uma FEM contrária constante.

Podemos considerar um motor CC como tendo o circuito equivalente mostrado na Figura 9.14(b), ou seja, a bobina da armadura sendo representada por um resistor R em série com uma indutância L em série com uma fonte de FEM contrária. Se desconsiderarmos a indutância da bobina da armadura, então a tensão que proporciona a corrente i na resistência é a tensão aplicada V menos a FEM contrária, ou seja, $V - v_b$. Portanto,

$$i = \frac{V - v_b}{R} = \frac{V - k_v \omega}{R}$$

O torque T é então

$$T = k_t i = \frac{k_t}{R} (V - k_v \omega)$$

Os gráficos do torque em relação à velocidade rotacional ω são uma série de linhas retas para diferentes valores de tensão (Figura 9.14(c)). O torque inicial, ou seja, o torque quando $\omega = 0$, é portanto proporcional à tensão aplicada, a velocidade sem carga é proporcional à tensão aplicada e o torque diminui com o aumento da velocidade.

A potência elétrica convertida em potência mecânica desenvolvida por um motor quando opera em condições de regime estacionário é o produto do torque

pela velocidade angular. A potência entregue pelo motor em regime estacionário é a soma da potência perdida na resistência da bobina da armadura e a potência mecânica desenvolvida.

Como um exemplo, considere o motor S6M41, produzido pela PMI Motors, que é um motor pequeno de ímã permanente com $k_t = 3,01 \text{ N cm/A}$, $k_v = 3,15 \text{ V por mil revoluções/min}$, uma resistência de terminal de $1,207 \Omega$ e uma resistência de armadura de $0,940 \Omega$.

9.5.2 Motores CC do tipo escova com bobinas de campo

Os **motores CC com bobinas de campo** são classificados como série, *shunt*, composto e excitado separadamente conforme as bobinas de campo e as de armadura são conectadas (Figura 9.15).

1 Motor com enrolamentos em série (Figura 9.15(a))

No motor com enrolamentos em série, as bobinas de armadura e campo estão em série. Este motor apresenta o maior torque de partida e tem a maior velocidade sem carga. Com pequenas cargas há o risco de o motor girar em uma velocidade muito alta. Inverter a polaridade da fonte nas bobinas não tem efeito no sentido de rotação do motor; ele continuará girando no mesmo sentido, pois as correntes de armadura e do campo são invertidas.

2 Motor shunt (Figura 9.15(b))

No motor *shunt*, as bobinas de armadura e de campo estão em paralelo. Ele proporciona o menor torque de partida, uma velocidade sem carga muito baixa e tem uma boa regulação de velocidade. Como a velocidade é quase

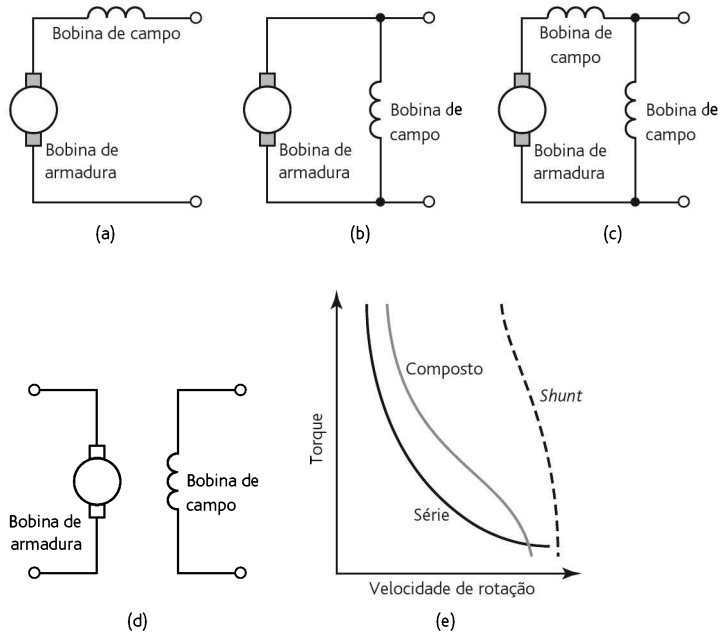


Figura 9.15 Motores CC: (a) série, (b) *shunt*, (c) composto, (d) enrolamentos em separado, (e) curvas características torque-velocidade.

constante, independentemente da carga, o motor *shunt* é bastante usado. Para inverter o sentido de rotação, a alimentação da armadura ou do campo deve ser invertida. Por essa razão, é preferível que os enrolamentos sejam excitados separadamente.

3 Motor composto (Figura 9.15(c))

O motor composto tem dois enrolamentos de campo, sendo um em série com a armadura e o outro em paralelo. O objetivo é reunir as melhores características dos motores série e *shunt*, ou seja, um alto torque de partida e uma boa regulação de velocidade.

4 Motor excitado separadamente (Figura 9.15(d))

O motor excitado separadamente tem o controle em separado das correntes de armadura e campo, e pode ser considerado um caso especial do motor *shunt*.

A velocidade deste motor CC pode ser alterada variando-se a corrente de armadura ou de campo. Geralmente é a corrente de armadura que é alterada. A escolha do motor depende da aplicação. Por exemplo, no caso de um robô manipulador, no pulso do robô pode-se usar um motor série porque a velocidade diminui com o aumento de carga. Um motor *shunt* pode ser usado onde é necessário uma velocidade constante, independentemente da carga.

9.5.3 Controle de motores CC tipo escova

A velocidade de um motor de ímã permanente depende da corrente na bobina de armadura. Em um motor com bobina de campo, a velocidade pode ser alterada variando-se a corrente da armadura ou do campo; geralmente é a corrente da armadura que é variada. Portanto, o controle de velocidade pode ser obtido controlando-se a tensão aplicada na armadura. Entretanto, como normalmente são usadas fontes de tensão fixa, uma tensão variável é obtida por meio de um circuito eletrônico.

A partir de uma fonte ca, o circuito com tiristor da Figura 9.4(b) pode ser usado para controlar a tensão média aplicada na armadura. Entretanto, geralmente ocorre dos sinais de controle de motores CC serem provenientes de um microprocessador. Nestes casos, geralmente é usada uma técnica conhecida como **modulação por largura de pulso (PWM – Pulse Width Modulation)**. Esta técnica basicamente envolve uma tensão de alimentação CC constante segmentada de modo que o seu valor médio seja variado (Figura 9.16).

A Figura 9.17(a) mostra como um PWM pode ser obtido através de um circuito básico com transistor. O transistor é ligado ou desligado através de um sinal aplicado em sua base. O diodo é usado para fornecer um caminho para a corrente que aumenta quando o transistor está desligado como resultado do motor agir como um gerador. Este circuito pode apenas ser usado para acionar o motor em um sentido de rotação; um outro circuito (Figura 9.17(b)) envolvendo quatro transistores, denominado circuito H, pode ser usado para permitir que o motor opere no sentido direto e reverso. Este circuito pode ser modificado com o uso de portas lógicas, de modo que uma entrada controla a comutação e a outra, o sentido de rotação (Figura 9.17(c)).

Os circuitos anteriores são exemplos de controle em malha aberta; assim, considera-se que as condições permanecerão constantes, como por exemplo, a tensão de alimentação e a carga acionada pelo motor. Os sistemas de controle em malha fechada usa a realimentação para corrigir a velocidade do motor caso as condições mudem. A Figura 9.18 mostra alguns métodos que podem ser empregados.

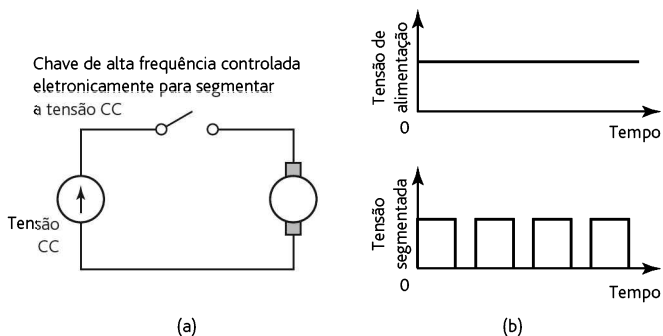


Figura 9.16 PWM: (a) princípios do circuito PWM, (b) variação da tensão de armadura segmentando-se a tensão CC.

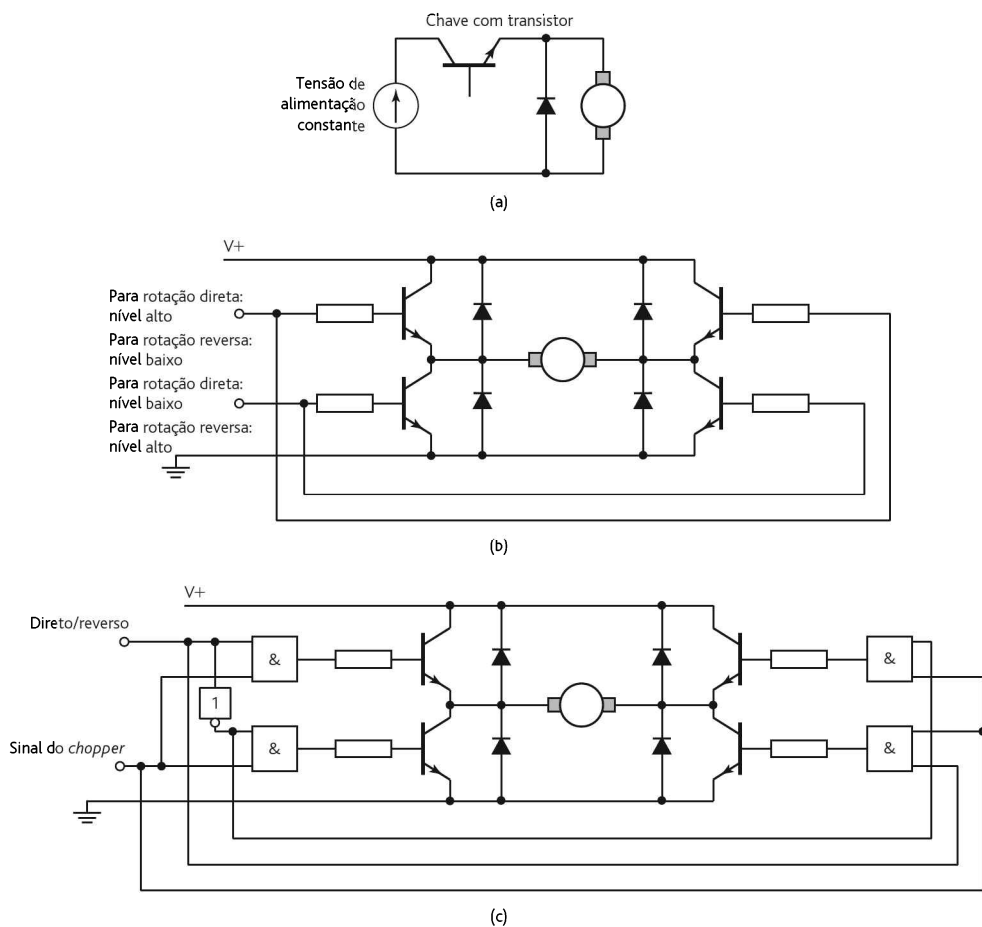


Figura 9.17 (a) Circuito básico com transistor, (b) circuito H, (c) circuito H com portas lógicas.

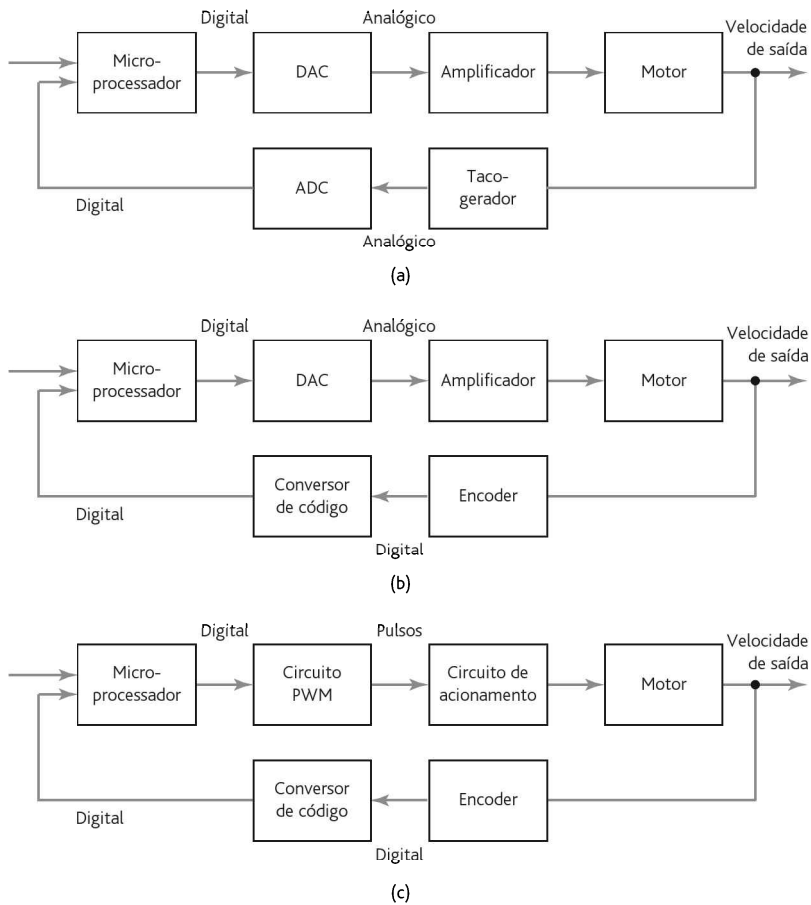


Figura 9.18 Controle de velocidade com realimentação.

Na Figura 9.18(a) o sinal de realimentação é fornecido por um tacogerador, que gera um sinal analógico que tem que ser convertido para o formato digital por um ADC para entrar no microprocessador. A saída do microprocessador é convertida em sinal analógico por um DAC e é usada para variar a tensão aplicada na armadura do motor CC. Na Figura 9.18(b) o sinal de realimentação é fornecido por um encoder, que fornece um sinal digital que, após a conversão de código, pode ser inserido no microprocessador. Como em (a), o sistema mostra uma tensão analógica que é variada para controlar a velocidade do motor. Na Figura 9.18(c) o sistema é completamente digital e o PWM é usado para controlar a tensão média aplicada na armadura.

9.5.4 Motores CC de ímã permanente sem escovas

Um problema que ocorre com os motores CC é que eles precisam de um comutador e escovas para inverter periodicamente a corrente em cada bobina da armadura. As escovas fazem um contato deslizante com o comutador e, como consequência,

ocorrem faíscas no salto entre dois segmentos do comutador e eles sofrem desgates. As escovas têm que ser substituídas periodicamente e o comutador, reconicionado. Para evitar estes problemas, os **motores sem escovas** foram projetados.

Essencialmente eles consistem de uma seqüência de bobinas no estator e um rotor de ímã permanente. Um condutor percorrido por uma corrente em um campo magnético sofre uma força; igualmente, como uma consequência da terceira lei de Newton do movimento, o ímã também sofre uma força igual e oposta. No caso dos motores CC convencionais, o ímã é fixo e os condutores percorridos por correntes podem se mover. No caso do motor CC de ímã permanente sem escovas, a situação é inversa: os condutores percorridos por correntes são fixos e o ímã permanente se move. O rotor é um ímã permanente de cerâmica ou ferrite. A Figura 9.19(a) mostra a forma básica deste motor. A corrente nas bobinas do estator é comutada eletronicamente por transistores em seqüência em torno das bobinas, a comutação sendo controlada através da posição do rotor de forma que sempre haja uma força agindo no ímã fazendo-o girar no mesmo sentido. Geralmente são usados sensores Hall para determinar a posição do rotor e iniciar a comutação através dos transistores, com os sensores posicionados em torno do estator.

A Figura 9.19(b) mostra os circuitos dos transistores de comutação que podem ser usados com o motor mostrado na Figura 9.19(a). Para comutar as bobinas em seqüência precisamos fornecer sinais para comutar os transistores na seqüência cor-

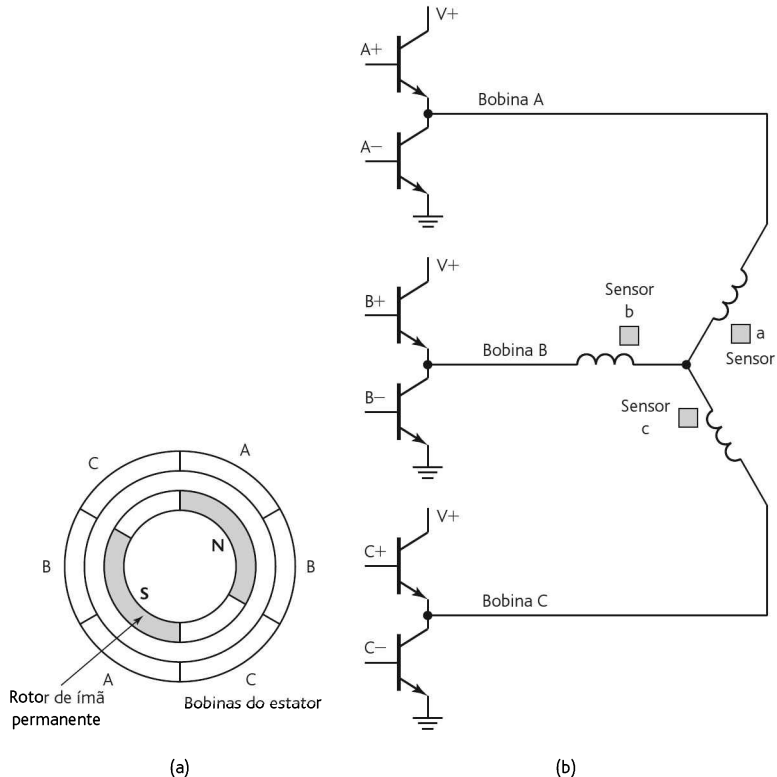


Figura 9.19 (a) Motor de ímã permanente sem escovas, (b) transistores de comutação.

reta. Isto é fornecido pelas saídas dos três sensores através de um circuito decodificador que fornece as correntes de base apropriadas. Portanto, quando o rotor estiver na posição vertical, ou seja, em 0° , existe saída do sensor c, o que não ocorre para os sensores a e b. Esta situação é usada para acionar os transistores A+ e B-. Para o rotor na posição 60° existem sinais dos sensores b e c e os transistores A+ e C- são acionados. A Tabela 9.1 mostra a sequência de comutação completa. O circuito completo para o controle deste motor está disponível em um único circuito integrado.

Os motores CC de ímã permanente sem escovas estão tendo uma utilização crescente em situações onde um alto desempenho juntamente com confiabilidade e baixo índice de manutenção são essenciais. Devido à falta de escovas, eles apresentam um funcionamento silencioso e capacidades de altas velocidades.

9.6 Motores CA

Os motores CA podem ser classificados em dois grupos, monofásicos e polifásicos, com cada grupo subdividido em motores de indução e síncronos. Os motores monofásicos tendem a ser usados para baixas potências e os polifásicos, para altas potências. Os motores de indução tendem a ser mais baratos que o síncronos e são bastante utilizados.

O motor de indução em gaiola de esquilo monofásico consiste de um rotor em gaiola de esquilo que é constituído de barras de cobre ou alumínio que se ajustam em fendas com anéis nas extremidades para formar circuitos elétricos completos (Figura 9.20(a)). Não há conexões elétricas externas ao rotor. O motor básico consiste deste rotor com um estator contendo um conjunto de enrolamentos. Quando uma corrente alternada passa nos enrolamentos do estator, é produzido um campo magnético alternado. Como resultado da indução eletromagnética, feds são induzidas nos condutores do rotor e correntes percorrem o rotor. Inicialmente, quando o rotor está parado, as forças nos condutores do rotor percorridos por correntes, e que se encontram no campo magnético do estator, resultam em um torque nulo. O motor não inicia o movimento por conta própria. Alguns métodos são usados para que isso aconteça, conferindo ao motor a capacidade de partir, ou seja, iniciar o movimento; um deles é usar um enrolamento de partida auxiliar para proporcionar ao motor o “empurrão” inicial. O rotor gira em uma velocidade determinada pela frequência da corrente alternada aplicada ao estator. Para uma fonte

Tabela 9.1 Sequência de comutação

Posição do rotor	Sinais dos sensores			Transistores ligados	
	a	b	c		
0°	0	0	1	A+	B-
60°	0	1	1	A+	C-
120°	0	1	0	B+	C-
180°	1	1	0	B+	A-
240°	1	0	0	C+	A-
360°	1	0	1	C+	B-

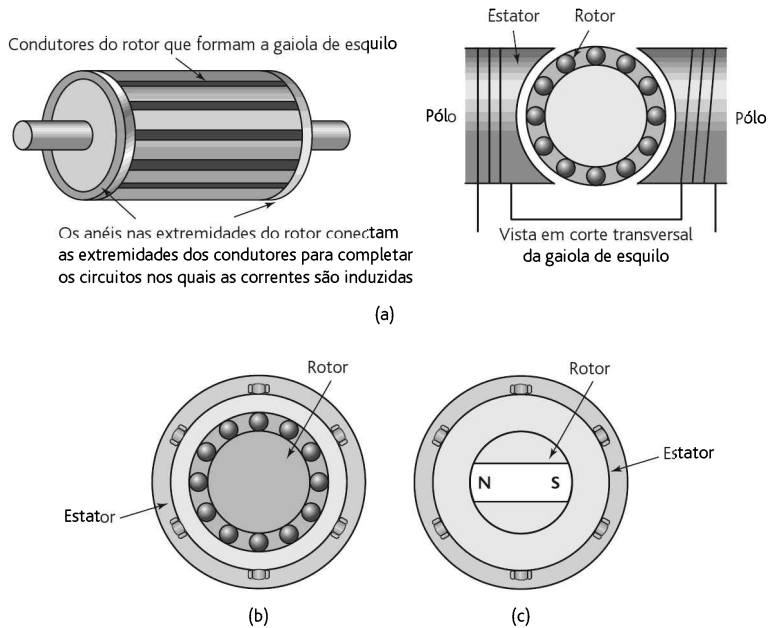


Figura 9.20 (a) Motor de indução monofásico, (b) motor de indução trifásico, (c) motor síncrono trifásico.

de alimentação de frequência constante que alimenta um motor monofásico de dois polos, o campo magnético alterna na mesma frequência. Essa velocidade de rotação do campo magnético é denominada **velocidade síncrona**. O rotor nunca gira exatamente nesta frequência, diferindo dela em cerca de 1 a 3%. Esta diferença é denominada **escorregamento**. Portanto, para uma fonte de 50 Hz, a velocidade de rotação do rotor será de quase 50 revoluções por segundo.

O **motor de indução trifásico** (Figura 9.20(b)) é similar ao equivalente monofásico, porém tem um estator com três enrolamentos situados a 120° uns dos outros, e cada um destes enrolamentos está conectado a uma das três linhas de alimentação. Como as três fases atingem o valor máximo de corrente em momentos diferentes, o campo magnético pode ser considerado girante em torno dos polos do estator, que completa uma rotação em um ciclo completo da corrente. A rotação do campo é muito mais suave que no motor monofásico. O motor trifásico tem uma grande vantagem sobre o monofásico: a partida automática. O sentido da rotação é invertido trocando-se duas conexões de linha quaisquer, pois isto inverte o sentido de rotação do campo magnético.

Os **motores síncronos** têm estatores similares aos descritos anteriormente para motores de indução, mas têm um rotor que é um ímã permanente (Figura 9.20(c)). O campo magnético produzido pelo estator gira e, desta forma, o ímã gira com ele. Com um par de polos por fase de alimentação, o campo magnético gira os 360° em um ciclo da alimentação e, portanto, a frequência de rotação é a mesma da fonte de alimentação. Os motores síncronos são usados quando é necessário uma velocidade precisa. Eles não têm partida automática e algum sistema deve promover a partida deles.

Os motores ac têm a grande vantagem em relação aos motores CC de serem mais baratos, além de mais robustos, confiáveis e não precisarem de manutenção. Entretanto, o controle de velocidade é geralmente mais complexo do que nos motores CC e, como consequência, um controlador CC de velocidade controlada é mais barato que o mesmo controlador ca, mas essa diferença está caindo continuamente em função dos desenvolvimentos tecnológicos e da redução no preço dos dispositivos de estado sólido (semicondutores). O controle de velocidade de motores CA é baseado em uma fonte de frequência variável, visto que a velocidade destes motores é determinada pela frequência da fonte de alimentação. O torque desenvolvido por um motor CA é constante quando a relação entre a tensão aplicada ao estator e a frequência for constante. Portanto, para manter o torque constante em diferentes velocidades quando a frequência é variada, a tensão aplicada ao estator tem que variar também. Em um dos métodos, a corrente alternada é primeiro retificada por um **conversor**, obtendo-se uma corrente contínua, seguido de uma **inversão**, em que a corrente volta a ser alternada, porém com uma frequência que pode ser selecionada (Figura 9.21). Um outro método que é muito usado em motores que operam em velocidades baixas é o uso de um **cicloconversor**. Este converte corrente alternada de uma frequência diretamente em corrente alternada de outra frequência, sem a conversão CC intermediária.

9.7 Motores de passo

O **motor de passo** é um dispositivo que produz rotação em movimentos de ângulos iguais, denominados de **passos**, para cada pulso digital em sua entrada. Portanto, por exemplo, se um pulso no motor produz uma rotação de 6° , então 60 pulsos produzem uma rotação de 360° . Existem alguns tipos de motores de passo:

1 Relutância variável

A Figura 9.22 mostra a forma básica do motor de passo de relutância variável. Neste caso, o rotor é feito de aço doce na forma cilíndrica com quatro polos, ou seja, menos polos do que no estator. Quando um par oposto de enrolamentos tem corrente circulando, um campo magnético é produzido com linhas de força que passam dos polos do estator para o conjunto de polos mais próximo do rotor. Visto que as linhas de força podem ser consideradas como se fossem fios elásticos e que sempre buscam ter o menor comprimento, o rotor se moverá até que os polos do rotor e estator se alinhem. Diz-se que esta posição tem relutância mínima. Este tipo apresenta ângulos com passos de $7,5^\circ$ ou 15° .

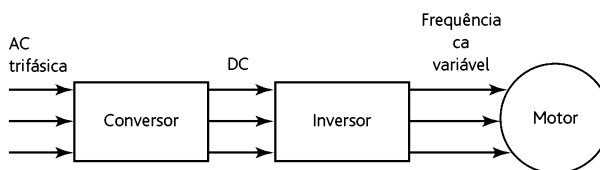


Figura 9.21 Motor CA de velocidade variável.

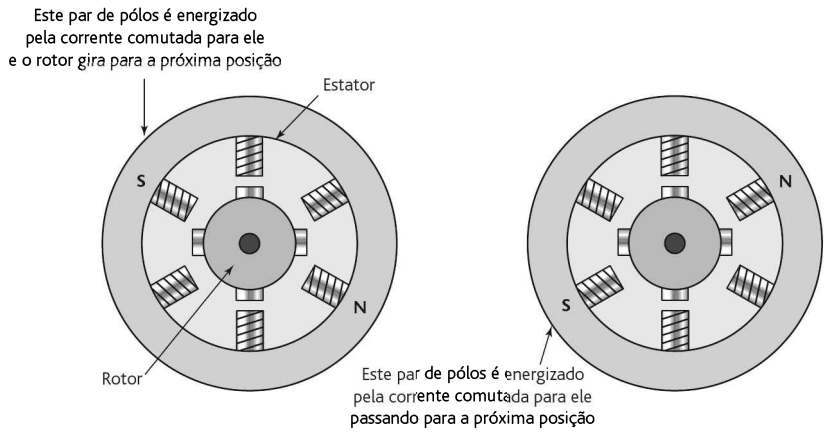


Figura 9.22 Motor de passo de relutância variável.

2 Ímã permanente

A Figura 9.23 mostra a forma básica de um motor de passo de ímã permanente. O motor mostrado tem um estator com quatro polos. Cada polo é enrolado com uma bobina de campo, com as bobinas em pares opostos de polos em série. A corrente é fornecida por uma fonte CC aos enrolamentos comutados. O rotor é um ímã permanente e, portanto, quando um par de polos do estator tem uma

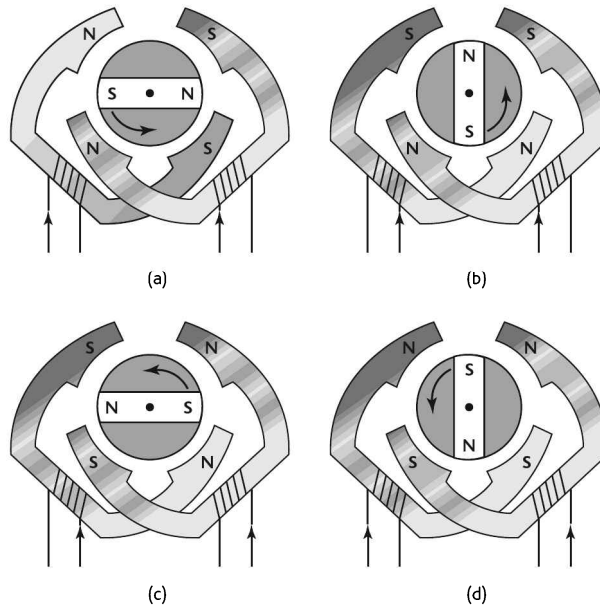


Figura 9.23 Motor de passo de ímã permanente com duas fases e passos de 90°. (a), (b), (c) e (d) mostram as posições do ímã do rotor conforme as bobinas são energizadas em diferentes sentidos.

corrente, o rotor se move para alinhar com ele. Assim, para as correntes dadas nas situações mostradas na figura, o rotor se move para a posição 45°. Se a corrente for comutada de forma que a polaridade seja invertida, o rotor gira mais 45° para se alinhar novamente. Portanto, comutando-se a corrente nas bobinas, o rotor gira em passos de 45°. Com este tipo de motor, os ângulos dos passos são normalmente de 1,8°, 7,5°, 15°, 30°, 34° ou 90°.

3 Híbrido

Os motores de passo híbrido combinam as características dos motores de relutância variável e ímã permanente, sendo constituídos de um ímã permanente encaixado em tampas de ferro que são cortadas formando dentes (Figura 9.24). O rotor se ajusta na posição de relutância mínima em resposta ao par de bobinas do estator que é energizado. Os ângulos de passo típicos são 0,9° e 1,8°. Se um motor tem n fases no estator e m dentes no rotor, o número total de passos por revolução é nm . Estes motores de passo são bastante usados em aplicações de posicionamento de alta precisão, como, por exemplo, em acionadores de disco rígido de computador.

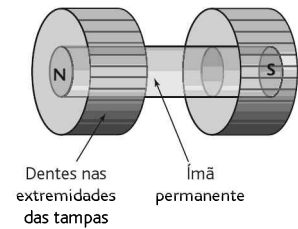


Figura 9.24 Motor de passo híbrido.

9.7.1 Especificações de motores de passo

Os termos a seguir são normalmente usados na especificação de motores de passo:

1 Fase

Este termo se refere ao número de enrolamentos independentes no estator, como, por exemplo, na descrição “motor de quatro fases”. A corrente necessária por fase e a sua resistência e indutância serão especificadas de modo que o controlador que comuta a saída seja especificado. Os motores de duas fases, como o da Figura 9.23, tendem a ser usados em aplicações de ciclos de trabalho pequenos, os motores de três fases tendem a ser de relutância variável, como o da Figura 9.22, e os motores de quatro fases tendem a ser usados em aplicações de potências maiores.

2 Ângulo de passo

Este é o ângulo em que o rotor gira para uma variação na comutação das bobinas do estator.

3 Torque de manutenção

Este é o torque máximo que pode ser aplicado a um motor energizado sem movê-lo de sua posição de repouso, ou seja, sem provocar uma rotação no eixo.

4 Torque pull-in

Este é o torque máximo vencido pelo motor na partida, para uma dada taxa de pulsos, alcançando o sincronismo sem perder um passo.

5 Torque pull-out

Este é o torque máximo que pode ser aplicado ao motor, que gira a uma taxa de passos dada, sem se perder o sincronismo.

6 Taxa pull-in

Esta é a taxa máxima de comutação na qual um motor com carga pode partir sem perder um passo.

7 Taxa pull-out

Esta é a taxa de comutação na qual um motor com carga permanece em sincronismo conforme a taxa de comutação é reduzida.

8 Faixa de deslocamento

Esta é a faixa da taxa de comutação entre *pull-in* e *pull-out* dentro da qual o motor gira em sincronismo mas não pode partir ou reverter.

A Figura 9.25 mostra a curva característica de um motor de passo.

9.7.2 Controle de motor de passo

A eletrônica de estado sólido é usada para comutar a alimentação CC entre os pares de enrolamentos do estator. Os motores de duas fases, como por exemplo o da Figura 9.23, são denominados **motores bipolares** quando têm quatro fios de conexão para os sinais para gerar a sequência de comutação (Figura 9.26(a)). Este motor pode ser acionado por um circuito H (veja a Figura 9.17 e a discussão relativa a ela); a Figura 9.26(b) mostra o circuito e a Tabela 9.2 mostra a sequência de comutação necessária para os transistores realizarem as quatro etapas, com a sequência sendo

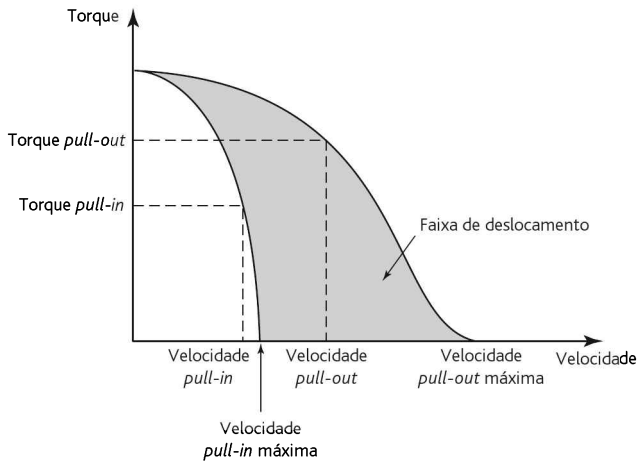


Figura 9.25 Curvas características de um motor de passo.

Tabela 9.2 Sequência de comutação para um motor de passo bipolar em um ciclo completo

Passo	Transistores			
	1 e 4	2 e 3	5 e 8	6 e 7
1	On	Off	On	Off
2	On	Off	Off	On
3	Off	On	Off	On
4	Off	On	On	Off

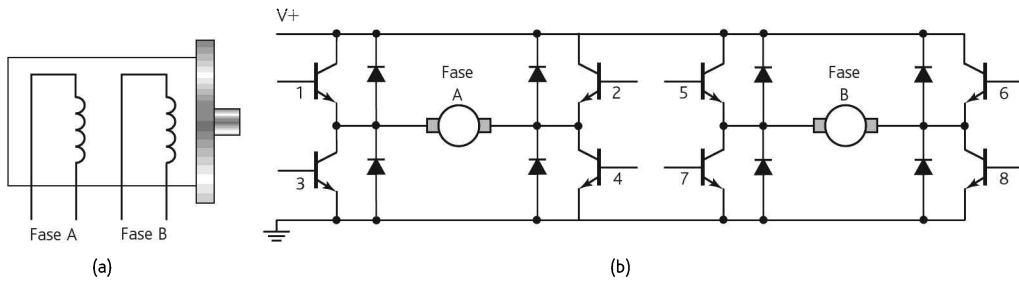


Figura 9.26 (a) Motor bipolar, (b) circuito H.

repetida em ciclos posteriores. Esta seqüência resulta em uma rotação em sentido horário; para uma rotação no sentido anti-horário, a seqüência é invertida.

Meio-passos, que significam resolução fina, são obtidos se, em vez de uma seqüência de passos inteiros em que é necessário implementar uma reversão de polo para ir de um passo para o próximo, as bobinas são comutadas de modo que o rotor pare em uma posição intermediária entre um passo e o próximo. A Tabela 9.3 mostra a seqüência de meio-passo para um motor de passo bipolar.

Os motores de duas fases são denominados **unipolares** quando têm seis fios de conexão para a geração da seqüência de comutação (Figura 9.27). Cada uma das bobinas tem uma derivação central. Com a derivação central das bobinas de fase interconectadas, esta forma de motor de passo pode ser comutada usando-se apenas quatro transistores. A Tabela 9.4 fornece a seqüência de comutação dos transistores para gerar os passos para uma rotação no sentido horário. Para gerar mais passos, a seqüência deve ser repetida. Para rotação no sentido anti-horário, a seqüência mostrada deve ser invertida. A Tabela 9.5 mostra a seqüência de meio-passo de acionamento do motor unipolar.

Existem circuitos integrados contendo todo o circuito de acionamento. A Figura 9.28 mostra as conexões com o circuito integrado SAA 1027 para um motor de fase de quatro

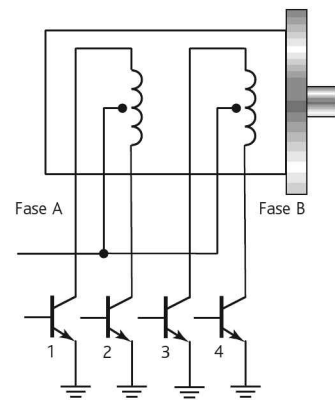


Figura 9.27 Motor unipolar.

Tabela 9.3 Meio-passos para motores de passo bipolares

Passo	Transistores			
	1 e 4	2 e 3	5 e 8	6 e 7
1	On	Off	On	Off
2	On	Off	Off	Off
3	On	Off	Off	On
4	Off	Off	Off	On
5	Off	On	Off	On
6	Off	On	Off	Off
7	Off	On	On	Off
8	Off	Off	On	Off

Tabela 9.4 Sequência de comutação para um motor de passo unipolar em passos inteiros

Passo	Transistores			
	1	2	3	4
1	On	Off	On	Off
2	On	Off	Off	On
3	Off	On	Off	On
4	Off	On	On	Off

Tabela 9.5 Meio-passos para motores de passo unipolares

Passo	Transistores			
	1	2	3	4
1	On	Off	On	Off
2	On	Off	Off	Off
3	On	Off	Off	On
4	Off	Off	Off	On
5	Off	On	Off	On
6	Off	On	Off	Off
7	Off	On	On	Off
8	Off	Off	On	Off

fases. As três entradas são controladas aplicando-se sinais de nível alto ou baixo nelas. Quando o terminal set for mantido em nível alto, a saída do circuito integrado muda de estado cada vez que o terminal disparo passar de nível baixo para alto. A sequência se repete a cada quatro passos, porém pode ser resetada para a condição zero a qualquer momento aplicando-se um sinal de nível baixo no terminal disparo. Quando a entrada de rotação for mantida em nível baixo, a rotação será no sentido horário, e, quando em nível alto, será no sentido oposto.

Algumas aplicações necessitam de ângulos de passo pequenos. Apesar do ângulo de passo poder ser feito pequeno aumentando-se o número de dentes e/ou o número de fases, geralmente não são usados mais que quatro fases e 50 a 100 dentes. Em vez disto, é usada uma técnica conhecida como **minipasso**. Isto envolve a divisão de cada passo em um determinado número de subpassos de tamanhos iguais. Isto é feito usando-se correntes diferentes nas bobinas de modo que o rotor se mova para posições intermediárias entre posições de passo normal. Portanto, por exemplo, um passo de $1,8^\circ$ pode ser subdividido em 10 passos iguais.

Os motores de passo podem ser usados para gerar passos de rotação controlada, mas também podem gerar rotação contínua com suas velocidades de rotação controladas pela taxa de controle na qual os pulsos são aplicados a eles para provocar o deslocamento em passos. Isto proporciona um útil motor de velocidade variável controlada que encontra várias aplicações.

Como as bobinas do motor de passo têm indutância, e cargas indutivas comutadas podem gerar grandes fems inversas, ao serem conectadas às portas de saída de

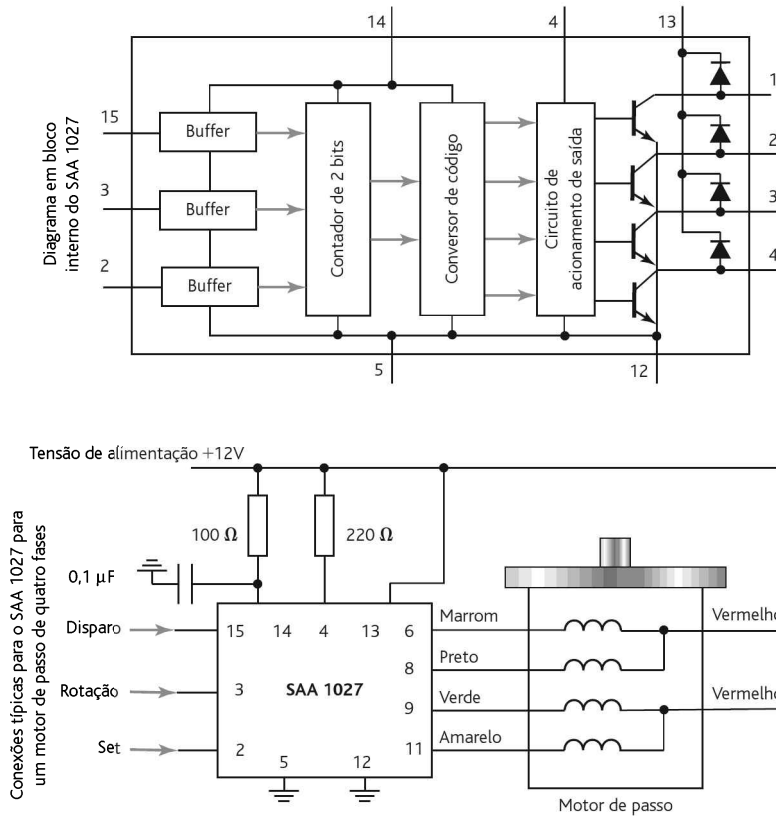


Figura 9.28 Circuito integrado SAA 1027 para motor de passo.

microprocessadores é necessário incluir proteção para evitar danos ao microprocessador. Esta proteção pode ser com resistores na linha para limitar a corrente, embora esses devam ter valores cuidadosamente escolhidos para proporcionar proteção e não limitar o valor da corrente necessária para comutar os transistores. A colocação de diodos nas bobinas protege evitando correntes reversas. Uma alternativa é o uso de optoisoladores (veja a Seção 3.3).

Resumo

Os **relés** são chaves operadas eletricamente em que a variação da corrente em um circuito liga ou desliga a corrente em um outro circuito.

Um **diodo** pode ser considerado um dispositivo que permite a passagem de corrente em apenas um sentido, apresentando uma resistência muito alta no outro sentido.

Um **tiristor** pode ser considerado como um diodo que tem uma porta que controla as condições nas quais o diodo pode ser ligado. Um **triac** é similar ao tiristor e é equivalente a um par de tiristores conectados em antiparalelo no mesmo chip.

Os **transistores bipolares** podem ser usados como chaves comutando a corrente de base entre zero e um valor que coloca o transistor na saturação. Os **MOS-FETs** são similares e também podem ser usados para comutação.

O princípio básico de um **motor CC** é uma espira de fio, a armadura, que é livre para girar no campo de um ímã como resultado de uma corrente que passa pela espira. O campo magnético pode ser gerado por um ímã permanente ou um eletroímã, ou seja, uma bobina de campo. A velocidade de um motor de ímã permanente depende da corrente na bobina de armadura; no caso do motor ter uma bobina de campo, ela dependerá da corrente na bobina de armadura ou na bobina de campo. Estes motores CC necessitam de um comutador e escovas para inverter periodicamente a corrente em cada bobina de armadura. Os **motores CC de ímã permanente sem escovas** têm um rotor de ímã permanente e uma sequência de bobinas no estator através das quais a corrente é comutada em sequência.

Os **motores CA** podem ser classificados em dois grupos, monofásicos e polifásicos, e cada um destes grupos é subdividido em motores de indução e síncrono. Os motores monofásicos tendem a ser usados para potências baixas, enquanto que os polifásicos são usados para potências altas. Os motores de indução tendem a ser mais baratos do que os síncronos e são mais usados.

O **motor de passo** é um dispositivo que produz rotação em movimentos de ângulos iguais, denominados **passos**, para cada pulso digital em sua entrada.

Problemas

- 9.1 Explique como o circuito mostrado na Figura 9.29 pode ser usado para retirar o efeito do repique de uma chave.

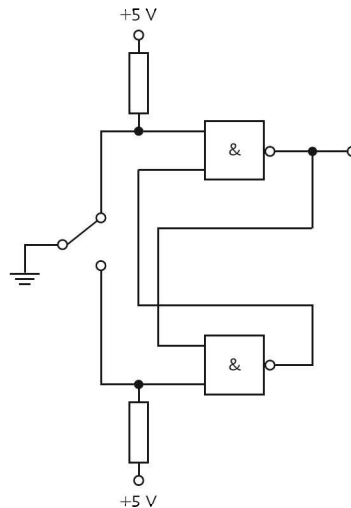


Figura 9.29 Problema 9.1.

- 9.2 Explique como um tiristor pode ser usado para controlar o nível de uma tensão CC segmentando a saída de uma fonte de tensão constante.
- 9.3 Necessita-se de um motor CC que tenha (a) um alto torque em velocidades baixas para acionar grandes cargas, (b) um torque que seja quase constante independentemente da velocidade. Sugira os tipos adequadas de motor.
- 9.4 Sugira os motores possíveis, CC ou CA, que podem ser considerados para aplicações em que (a) é necessário que o torque seja constante e o motor seja barato, (b) é necessário um alto controle de velocidade, (c) é necessário baixas velocidades, (d) é necessário manutenção mínima.
- 9.5 Explique o princípio do motor CC de ímã permanente sem escovas.
- 9.6 Explique o princípio de operação de um motor de passo de relutância variável.
- 9.7 Se um motor de passo tem um ângulo de passo de $7,5^\circ$, qual é a taxa de entrada digital para que ele produza uma rotação de 10 revoluções/s?
- 9.8 Qual será o ângulo de passo para um motor de passo híbrido com oito enrolamentos no estator e dez dentes no rotor?
- 9.9 Um motor CC de ímã permanente tem uma resistência de armadura de $0,5 \Omega$, e quando uma tensão de 120 V é aplicada ao motor, ele alcança a velocidade de rotação em estado estacionário de 20 revoluções/s e drena 40 A. Qual será (a) a potência de entrada do motor, (b) a perda de potência na armadura, (c) o torque gerado nesta velocidade?
- 9.10 Se um motor CC produz um torque de $2,6 \text{ N} \cdot \text{m}$ quando a corrente de armadura é 2 A, qual será o torque com uma corrente de 0,5 A?
- 9.11 Quantos pulsos/passos por segundo um microprocessador precisa gerar para um motor de passo se o motor deve ter uma saída de 0,25 revoluções/s e um ângulo de passo de $7,5^\circ$?
- 9.12 Um motor de passo é usado para girar uma polia de 240 mm de diâmetro e, portanto, uma correia que move uma massa de 200 kg. Se essa massa for acelerada uniformemente do repouso a 100 mm/s em 2 s e houver uma força de fricção constante de 20 N, qual será o torque necessário para o motor?

10

Modelos de Sistemas Básicos

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Explicar a importância de modelos na previsão do comportamento de sistemas.
- Idealizar modelos a partir de blocos construtivos básicos para os sistemas mecânico, elétrico, fluido e térmico.
- Reconhecer as analogias entre os sistemas mecânico, elétrico, fluido e térmico.

10.1 Modelos matemáticos

Considere a situação em que um microprocessador liga um motor. Qual será a rotação do eixo do motor em relação ao tempo? A velocidade não alcançará imediatamente o seu valor final, mas após algum tempo. Considere uma outra situação em que um sistema hidráulico é usado para abrir uma válvula que permite que a água entre no tanque para recuperar o nível de água desejado. Como o nível da água varia em relação ao tempo? O nível desejado de água no tanque não é alcançado imediatamente, mas após algum tempo.

Os **modelos matemáticos** são necessários para se entender o comportamento dos sistemas. Estes modelos são representações simplificadas de certos aspectos de um sistema real. Eles são criados usando-se equações para descrever a relação entre a entrada e a saída do sistema, podendo ser usados para prever o comportamento de um sistema sob determinadas condições, como, por exemplo, as saídas para um conjunto de entradas, ou as saídas se um determinado parâmetro for alterado. Ao projetar o modelo matemático de um sistema, é necessário fazer considerações e simplificações, fazendo-se uma escolha ponderada entre a simplicidade do modelo e a necessidade de representar o comportamento do mundo real. Por exemplo, podemos criar um modelo matemático para uma mola considerando que a extensão x é proporcional à força aplicada F , ou seja $F = kx$. Este modelo simplificado pode não propiciar uma previsão precisa do comportamento de uma mola real em que a extensão pode não ser

exatamente proporcional à força e onde não podemos aplicar este modelo independentemente da amplitude da força, visto que grandes forças deformam permanentemente a mola, podendo quebrá-la, e isto o modelo simples não permite prevê.

A base para qualquer modelo matemático é fornecida pelas leis fundamentais da física que governam o comportamento do sistema. Neste capítulo, uma gama de sistemas será considerada, incluindo exemplos de sistemas mecânico, elétrico, térmico e fluido.

Assim como uma criança constrói casas, carros, guindastes, etc. a partir de blocos construtivos básicos (blocos de montar que são encaixados uns nos outros), os sistemas podem ser construídos a partir de uma gama de blocos construtivos. Cada bloco construtivo é considerado como tendo uma propriedade ou função simples. Portanto, tomando um exemplo simples, um circuito elétrico pode ser feito a partir de blocos construtivos que representem os comportamentos de resistores, capacitores e indutores. O bloco construtivo resistor é considerado tendo a propriedade de uma resistência pura, assim como o capacitor como tendo capacitância pura e o indutor como tendo indutância pura. Combinando estes blocos de diferentes formas, podemos construir uma variedade de circuitos elétricos, com a relação entrada/saída do sistema obtida combinando-se de forma apropriada as relações dos blocos construtivos. Assim, podemos obter um modelo matemático para o sistema. Um sistema construído desta forma é denominado sistema de **parâmetros concentrados**. Isto porque cada parâmetro, ou seja, propriedade ou função, é considerado independentemente.

Existem semelhanças no comportamento de blocos construtivos usados nos sistemas mecânico, elétrico, térmico e fluido. Este capítulo aborda os blocos construtivos básicos e suas combinações para produzir modelos matemáticos para sistemas físicos reais. O Capítulo 11 aborda modelos mais complexos. É necessário enfatizar que tais modelos apenas auxiliam no projeto de sistemas. Os sistemas reais frequentemente apresentam características não lineares e podem se distanciar dos modelos ideais desenvolvidos neste capítulo. Esta questão é abordada no Capítulo 11.

10.2 Blocos construtivos de sistemas mecânicos

Os modelos usados para representar os sistemas mecânicos têm blocos construtivos básicos de molas, amortecedores e massas. As **molas** representam a rigidez de um sistema, os **amortecedores**, as forças que se opõem ao movimento, ou seja, fricção ou efeitos de amortecimento, e as **massas**, a inércia ou a resistência à aceleração (Figura 10.1). Os sistemas mecânicos não precisam ser realmente construídos de molas, amortecedores e massas, mas ter as propriedades de rigidez, amortecimento e inércia. Todos esses blocos construtivos básicos podem ser considerados como tendo uma força como entrada e um deslocamento como saída.

A rigidez de uma mola é descrita pela relação entre a força F usada para estender ou comprimir a mola e a extensão ou compressão resultante x (Figura 10.1(a)). No caso da mola em que a extensão ou a compressão é proporcional à força aplicada, ou seja, uma mola linear,

$$F = kx$$

em que k é uma constante. Quanto maior o valor de k , maior a força para estender ou comprimir a mola, e assim, maior a rigidez. O objeto que aplica a força para es-

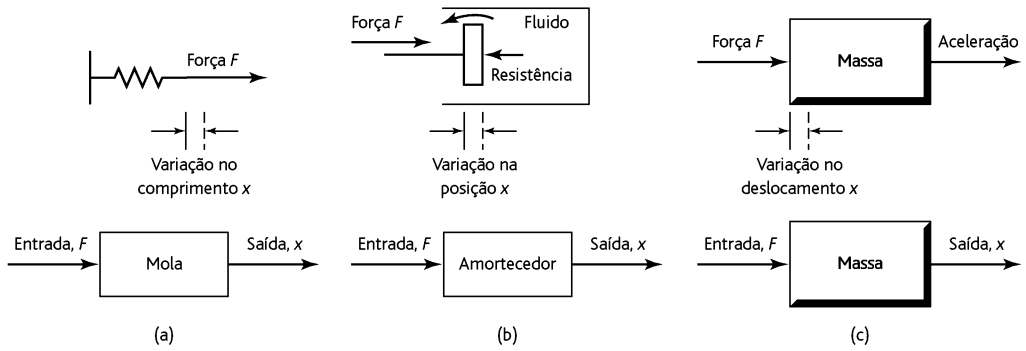


Figura 10.1 Sistemas mecânicos: (a) mola, (b) amortecedor, (c) massa.

tender a mola também sofre uma força exercida pela mola estendida (terceira lei de Newton). Esta força tem sentido oposto e mesma intensidade que a força usada para estender a mola, ou seja, kx .

O bloco amortecedor representa os tipos de forças experimentadas quando tentamos empurrar um objeto em um fluido ou movê-lo contra forças de fricção. Quanto mais rápido o objeto é empurrado, maiores as forças de oposição. O amortecedor que é usado ilustrativamente para representar estas forças de amortecimento que diminuem o movimento dos objetos consiste de um pistão se movendo em um cilindro fechado (Figura 10.1(b)). Para o movimento do pistão é necessário que o fluido entre de um lado dele ou do outro. Este fluxo produz uma força resistiva. Em um caso ideal, o amortecimento ou a força de resistência F é proporcional à velocidade v do pistão. Portanto,

$$F = cv$$

em que c é uma constante. Quanto maior o valor de c , maior a força de amortecimento para uma determinada velocidade. Visto que a velocidade é a taxa de variação do deslocamento x do pistão, ou seja, $v = dx/dt$, então

$$F = c \frac{dx}{dt}$$

Portanto, a relação entre o deslocamento x do pistão, ou seja, a saída, e a força de entrada é uma relação que depende da taxa de variação da saída.

O bloco construtivo massa (Figura 10.1(c)) apresenta a propriedade de que quanto maior a massa, maior a força necessária para fornecer a ela uma determinada aceleração. A relação entre a força F e a aceleração a (segunda lei de Newton) é $F = ma$, em que a constante de proporcionalidade entre a força e a aceleração é denominada massa (m). A aceleração é a taxa de variação da velocidade, ou seja, dv/dt , e a velocidade v é a taxa de variação do deslocamento x , ou seja, $v = dx/dt$. Portanto,

$$F = ma = m \frac{dv}{dt} = m \frac{d(dx/dt)}{dt} = m \frac{d^2x}{dt^2}$$

É necessária energia para esticar a mola, acelerar a massa e mover o pistão no amortecedor. Entretanto, no caso da mola e da massa podemos obter de volta a energia, o que não ocorre no caso do amortecedor. A mola armazena energia quando esticada e a libera quando volta ao comprimento anterior. A energia armazenada quando há uma extensão x é $\frac{1}{2}kx^2$. Visto que $F = kx$, a expressão anterior pode ser escrita como

$$E = \frac{1}{2} \frac{F^2}{k}$$

Existe também energia armazenada na massa quando ela se move com uma velocidade v , denominada energia cinética, que é liberada quando cessa o movimento:

$$E = \frac{1}{2} mv^2$$

Entretanto, não há energia armazenada no amortecedor. Ele não retorna para sua posição original quando não há força na entrada. Ele dissipa a energia em vez de armazená-la. A energia P dissipada depende da velocidade v e é dada por

$$P = cv^2$$

10.2.1 Sistemas rotacionais

A mola, o amortecedor e a massa são os blocos construtivos básicos para os sistemas mecânicos em que forças e deslocamentos em linha reta são envolvidos sem qualquer rotação. Se houver rotação, então os três blocos construtivos equivalentes serão a **mola de torção**, o **amortecedor de rotação** e o **momento de inércia**, ou seja, a inércia de uma massa em rotação. Com estes blocos construtivos, a entrada é o torque e a saída é o ângulo de rotação. Com uma mola de torção, o ângulo θ de rotação é proporcional ao torque T . Assim,

$$T = k\theta$$

Em um amortecedor rotativo, um disco é girado em um fluido e o torque resistivo T é proporcional à velocidade angular ω , e como esta velocidade é a taxa na qual o ângulo varia, ou seja, $d\theta/dt$,

$$T = c\omega = c \frac{d\theta}{dt}$$

O bloco construtivo momento de inércia tem a propriedade de que quanto maior o momento de inércia I , maior o torque necessário para produzir uma aceleração angular α :

$$T = I\alpha$$

Portanto, visto que a aceleração angular é a taxa de variação do deslocamento angular, ou seja, $d\omega/dt$, e a velocidade angular é a taxa de variação do deslocamento angular, então

$$T = I \frac{d\omega}{dt} = I \frac{d(d\theta/dt)}{dt} = I \frac{d^2\theta}{dt^2}$$

A mola de torção e a massa que gira armazenam energia; o amortecedor rotativo apenas dissipa energia. A energia armazenada na mola de torção quando torcida em um ângulo θ é $\frac{1}{2}k\theta^2$, e como $T = k\theta$, podemos escrever

$$E = \frac{1}{2} \frac{T^2}{k}$$

A energia armazenada por uma massa com uma velocidade angular ω é a energia cinética E , em que

$$E = \frac{1}{2}I\omega^2$$

A potência P dissipada pelo amortecedor rotativo, quando gira com uma velocidade angular ω , é

$$P = c\omega^2$$

A Tabela 10.1 resume as equações que definem as características dos blocos construtivos mecânicos quando há, no caso dos deslocamentos em linha reta (denominados translacionais), uma força de entrada F e um deslocamento x de saída, e, no caso da rotação, um torque T e um deslocamento angular θ .

10.2.2 Construção de um sistema mecânico

Muitos sistemas podem ser considerados como constituídos essencialmente de uma massa, uma mola e um amortecedor combinados na forma mostrada na Figura 10.2(a), sendo a força F a entrada e o deslocamento x a saída (Figura 10.2(b)). Para avaliar a relação entre a força e o deslocamento para o sistema, o procedimento a ser adotado é considerar apenas uma massa e uma força agindo no corpo. Um diagrama da massa e apenas das forças que atuam nela é denominado **diagrama de corpo livre** (Figura 10.2(c)).

Tabela 10.1 Blocos construtivos mecânicos

Blocos construtivos	Equação descritiva	Energia armazenada ou potência dissipada
<i>Translacional</i>		
Mola	$F = kx$	$E = \frac{1}{2} \frac{F^2}{k}$
Amortecedor	$F = c \frac{dx}{dt} = cv$	$P = cv^2$
Massa	$F = m \frac{d^2x}{dt^2} = m \frac{dv}{dt}$	$E = \frac{1}{2}mv^2$
<i>Rotacional</i>		
Mola	$T = k\theta$	$E = \frac{1}{2} \frac{T^2}{k}$
Amortecedor rotacional	$T = c \frac{d\theta}{dt} = c\omega$	$P = c\omega^2$
Momento de inércia	$T = I \frac{d^2\theta}{dt^2} = I \frac{d\omega}{dt}$	$E = \frac{1}{2}I\omega^2$

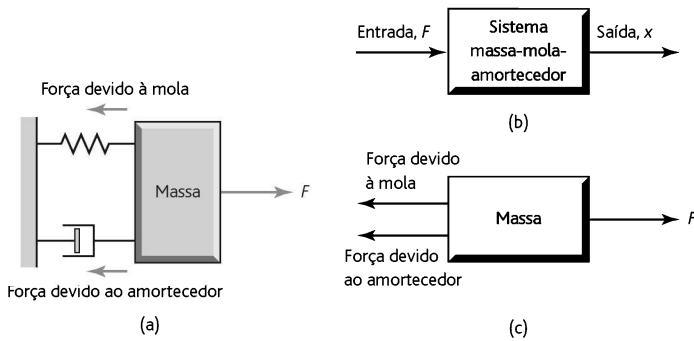


Figura 10.2 (a) Massa-mola-amortecedor, (b) sistema, (c) diagrama de corpo livre.

Quando algumas forças atuam concorrentemente em um corpo, o seu equivalente único resultante pode ser determinado pela adição de vetores. Se todas as forças atuam ao longo da mesma linha reta ou linhas paralelas, isto significa que a força resultante que age no bloco é uma soma algébrica. Portanto, para a massa na Figura 10.2(c)), se considerarmos apenas as forças que agem no bloco, então a força resultante aplicada na massa é a força aplicada F menos a força resultante de tração ou compressão da mola e menos a força do amortecedor. Assim,

$$\text{força resultante aplicada à massa } m = F - kx - cv$$

onde v é a velocidade na qual o pistão no amortecedor e, portanto, a massa move-se. Esta força resultante é a força aplicada na massa que provoca sua aceleração. Portanto,

$$\text{força resultante aplicada à massa} = ma$$

Então,

$$F - kx - c \frac{dx}{dt} = m \frac{d^2x}{dt^2}$$

ou, quando rearranjado

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = F$$

Esta equação, denominada **equação diferencial**, descreve a relação entre a força de entrada F no sistema e o deslocamento x de saída. Devido ao termo d^2x/dt^2 , a equação diferencial é de segunda ordem; uma equação de primeira ordem teria apenas o termo dx/dt .

Existem vários sistemas que podem ser construídos a partir de combinações adequadas dos blocos construtivos massa, mola e amortecedor. A Figura 10.3 ilustra alguns.

A Figura 10.3(a) mostra o modelo para uma máquina montada sobre o solo que poderia ser usado como base para o estudo dos efeitos dos distúrbios do solo sobre os deslocamentos da base da máquina. A Figura 10.3(b) mostra um modelo para a roda e a suspensão de um carro ou caminhão e pode ser usado para o estudo do comportamento esperado do veículo quando guiado por uma estrada com relevo acidentado e, assim, servir de base para o projeto da suspensão. A Figura 10.3(c) mostra como este modelo pode ser usado como parte de um modelo maior para prever como o motorista se sente

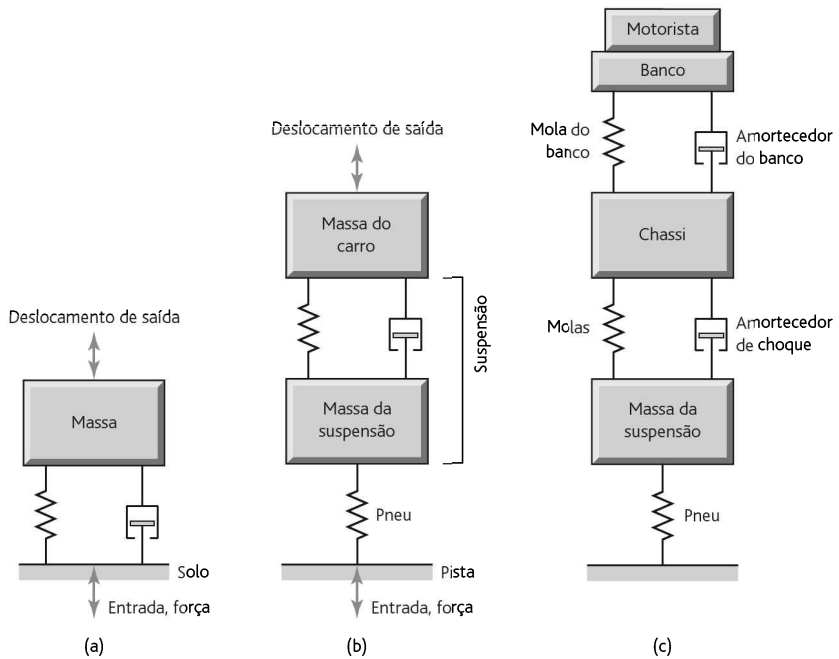


Figura 10.3 Modelos para (a) máquina montada sobre o solo, (b) o chassi de um carro como resultado de um roda que gira em uma pista, (c) o motorista de um carro dirigindo ao longo de uma pista.

quando dirige ao longo de uma pista. O procedimento a ser adotado para a análise destes modelos é exatamente o mesmo descrito antes para o modelo simples massa-mola-amortecedor. Um diagrama de corpo livre é desenhado para cada massa no sistema, e mostra cada massa independentemente e apenas as forças que atuam nela. Então, para cada massa, a resultante das forças que agem nela é igual ao produto da massa pela aceleração dela.

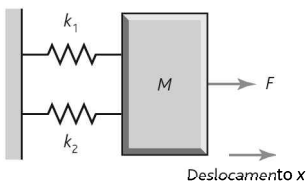


Figura 10.4 Exemplo.

Para ilustrar isto, considere a derivada da equação diferencial que descreve a relação entre a força F de entrada e o deslocamento x de saída para o sistema mostrado na Figura 10.4.

A força resultante aplicada à massa é F menos a força resistente exercida por cada uma das molas. Visto que estas são k_1x e k_2x , então

$$\text{força resultante} = F - k_1x - k_2x$$

Visto que a força resultante provoca a aceleração da massa, então

$$\text{força resultante} = m \frac{d^2x}{dt^2}$$

Portanto,

$$m \frac{d^2x}{dt^2} + (k_1 + k_2)x = F$$

O procedimento para a obtenção da equação diferencial que relaciona entradas e saídas para um sistema mecânico que consiste de vários componentes pode ser resumido como:

- 1 Isole os componentes no sistema e desenhe os diagramas de corpo livre para cada um.
- 2 Assim, com as forças identificadas para um componente, escreva a equação de modelamento para ele.
- 3 Combine as equações para os componentes do sistema para obter a equação diferencial dele.

Como uma ilustração, considere a derivada da equação diferencial que descreve o movimento da massa m_1 na Figura 10.5(a) quando uma força F é aplicada. Considere os diagramas de corpo livre (Figura 10.5(b)). Para a massa m_2 , estas são as forças F e a força exercida pela mola superior. A força exercida pela mola superior é devido a ela ser estendida por $(x_2 - x_3)$ e portanto é $k_2(x_3 - x_2)$. Portanto, a força resultante que age na massa é

$$\text{força resultante} = F - k_2(x_3 - x_2)$$

Esta força faz com que a massa acelere, e assim

$$F - k_2(x_3 - x_2) = m_2 \frac{d^2x_3}{dt^2}$$

No diagrama de corpo livre para a massa m_1 , a força exercida pela mola superior é $k_2(x_3 - x_2)$ e pela mola superior é $k_1(x_1 - x_2)$. Portanto, a força resultante que age na massa é

$$\text{força resultante} = k_1(x_2 - x_1) - k_2(x_3 - x_2)$$

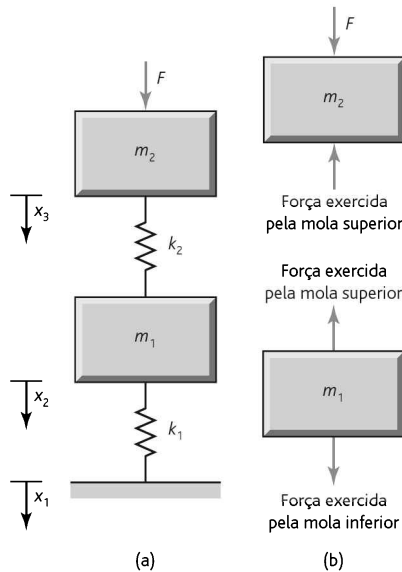


Figura 10.5 Sistema massa-mola.

Esta força faz com que a massa acelere, e assim

$$k_1(x_2 - x_1) - k_2(x_3 - x_2) = m_1 \frac{d^2 x_2}{dt^2}$$

Temos então duas equações diferenciais de segunda ordem simultâneas para descrever o comportamento do sistema.

Modelos similares podem ser construídos para sistemas rotativos. Para avaliar a relação entre o torque e o deslocamento angular para o sistema, o procedimento a ser adotado é considerar apenas um bloco de massa rotacional e apenas os torques que agem neste corpo. Quando vários torques agem em um corpo simultaneamente, a resultante equivalente pode ser determinada por adição em que os sentidos dos torques são considerados. Portanto, um sistema que envolve um torque usado para girar uma massa na extremidade de um eixo (Figura 10.6(a)) pode ser considerado representado pelos blocos construtivos rotacionais mostrados na Figura 10.6(b). Esta é uma situação comparável com a analisada antes (Figura 10.2) para deslocamentos lineares, e resulta em uma equação similar:

$$I \frac{d^2 \theta}{dt^2} + c \frac{d\theta}{dt} + k\theta = T$$

10.3 Blocos construtivos de sistemas elétricos

Os blocos construtivos básicos de sistemas elétricos são indutores, capacitores e resistores (Figura 10.7).

Para um **indutor**, a diferença de potencial v nele em qualquer instante depende da taxa de variação da corrente (di/dt) nele:

$$v = L \frac{di}{dt}$$

em que L é a indutância. O sentido da diferença de potencial é oposto ao da diferença de potencial usada para produzir a corrente no indutor, por isto o termo FEM inversa. A equação pode ser reescrita como

$$i = \frac{1}{L} \int v dt$$

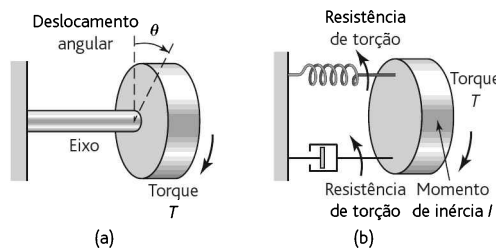


Figura 10.6 Giro de uma massa na extremidade de um eixo: (a) situação física, (b) modelo em blocos construtivos.

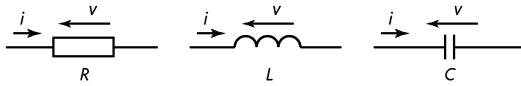


Figura 10.7 Blocos construtivos elétricos.

Para um **capacitor**, a diferença de potencial nele depende da carga q nas placas do capacitor no instante desejado:

$$v = \frac{q}{C}$$

em que C é a capacitância. Visto que a corrente i de carga ou descarga do capacitor é a taxa na qual a carga se move na carga ou descarga das placas, ou seja, $i = dq/dt$, então a carga total q nas placas é dada por

$$q = \int i dt$$

e assim

$$v = \frac{1}{C} \int i dt$$

Alternativamente, visto que $v = q/C$, então

$$\frac{dv}{dt} = \frac{1}{C} \frac{dq}{dt} = \frac{1}{C} i$$

e assim

$$i = C \frac{dv}{dt}$$

No caso de um **resistor**, a diferença de potencial v nele em qualquer instante depende de sua corrente i :

$$v = Ri$$

em que R é a resistência.

O indutor e o capacitor armazenam energia que pode ser liberada em um instante posterior. Um resistor não armazena energia, apenas a dissipa. A energia armazenada por um indutor quando percorrido por uma corrente i é

$$E = \frac{1}{2} Li^2$$

A energia armazenada por um capacitor quando há uma diferença de potencial v nele é

$$E = \frac{1}{2} Cv^2$$

A potência P dissipada por um resistor quando há uma diferença de potencial v nele é

$$P = iv = \frac{v^2}{R}$$

A Tabela 10.2 resume as equações que definem as características dos blocos construtivos elétricos quando a entrada é corrente e a saída é diferença de potencial. Compare-as com as equações dadas na Tabela 10.1 para os blocos construtivos de sistemas mecânicos.

10.3.1 Construção de um modelo para um sistema elétrico

As equações que descrevem como os blocos construtivos elétricos podem ser combinados são as **leis de Kirchhoff**. Estas podem ser expressas como:

- Lei 1: A corrente total que entra em um nó é igual à corrente total que sai deste nó, ou seja, a soma algébrica das correntes em um nó é zero.
- Lei 2: Em um circuito fechado ou malha, a soma algébrica das diferenças de potenciais em cada parte do circuito é igual à FEM aplicada.

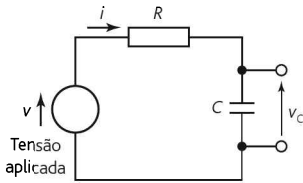


Figura 10.8 Sistema resistor-capacitor.

Agora considere um sistema elétrico simples que consiste de um resistor e um capacitor em série, como mostra a Figura 10.8. Aplicando a segunda lei de Kirchhoff a esta malha, obtemos

$$v = v_R + v_C$$

em que v_R é a diferença de potencial no resistor e v_C , no capacitor. Visto que o circuito tem apenas uma malha, a corrente i em todos os elementos do circuito é igual. Se a saída do circuito for a diferença de potencial no capacitor, v_C , então como $v_C = iR$ e $i = C(dv_C/dt)$,

$$v = RC \frac{dv_C}{dt} + v_C$$

Esta é a relação entre a saída v_C e a entrada v e é uma equação diferencial de primeira ordem.

A Figura 10.9 mostra um sistema resistor-indutor-capacitor. Se a segunda lei de Kirchhoff for aplicada a esta malha, teremos

$$v = v_R + v_L + v_C$$

em que v_R é a diferença de potencial no resistor, v_L , no indutor, e v_C , no capacitor. Como o circuito tem apenas uma malha, a corrente i é igual em todos os elementos

Tabela 10.2 Blocos construtivos elétricos

Blocos construtivos	Equação descritiva	Energia armazenado ou potência dissipada
Indutor	$i = \frac{1}{L} \int v dt$ $v = L \frac{di}{dt}$	$E = \frac{1}{2} L i^2$
Capacitor	$i = C \frac{dv}{dt}$	$E = \frac{1}{2} C v^2$
Resistor	$i = \frac{v}{R}$	$P = \frac{v^2}{R}$

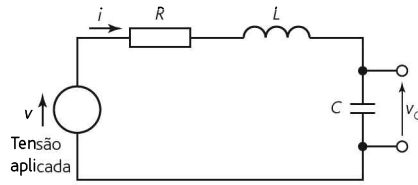


Figura 10.9 Sistema resistor-indutor-capacitor.

do circuito. Se a saída do circuito for a diferença de potencial no capacitor, v_C , então, como $v_R = iR$ e $v_L = L(di/dt)$,

$$v = iR + L \frac{di}{dt} + v_C$$

Porém, $i = C(dv_C/dt)$ e, deste modo,

$$\frac{di}{dt} = C \frac{d(dv_C/dt)}{dt} = C \frac{d^2v_C}{dt^2}$$

Portanto,

$$v = RC \frac{dv_C}{dt} + LC \frac{d^2v_C}{dt^2} + v_C$$

Esta é uma equação diferencial de segunda ordem.

Como mais uma ilustração, considere a relação entre a saída, que é a diferença de potencial no indutor (v_L), e a entrada v para o circuito mostrado na Figura 10.10. Aplicando a segunda lei de Kirchhoff a este circuito, temos

$$v = v_R + v_L$$

em que v_R é a diferença de potencial no resistor R e v_L , no indutor. Visto que $v_R = iR$,

$$v = iR + v_L$$

Como

$$i = \frac{1}{L} \int v_L dt$$

então a relação entre a entrada e a saída é

$$v = \frac{R}{L} \int v_L dt + v_L$$

Como mais um exemplo, considere a relação entre a saída, a diferença de potencial v_C no capacitor, e a entrada v para o circuito mostrado na Figura 10.11. Aplicando a primeira lei de Kirchhoff ao nó A, temos

$$i_1 = i_2 + i_3$$

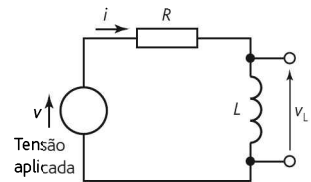


Figura 10.10 Sistema resistor-indutor.

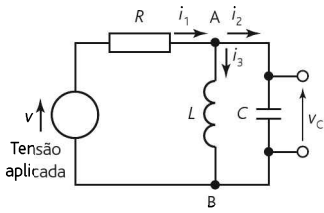


Figura 10.11 Sistema resistor-capacitor-indutor.

Porém,

$$i_1 = \frac{v - v_A}{R}$$

$$i_2 = \frac{1}{L} \int v_A dt$$

$$i_3 = C \frac{dv_A}{dt}$$

Portanto,

$$\frac{v - v_A}{R} = \frac{1}{L} \int v_A dt + C \frac{dv_A}{dt}$$

Mas $v_C = v_A$. Portanto, isolando v temos

$$v = RC \frac{dv_C}{dt} + v_C + \frac{R}{L} \int v_C dt$$

10.3.2 Analogias entre os sistemas elétrico e mecânico

Os blocos construtivos para os sistemas elétrico e mecânico têm muitas semelhanças (Figura 10.12). Por exemplo, o resistor elétrico não armazena energia, mas a dissipa, sendo a corrente i no resistor dada por $i = v/R$, em que R é uma constante, e a potência P dissipada dada por $P = v^2/R$. O sistema mecânico análogo ao resistor é o amortecedor. Ele também não armazena energia, mas a dissipa, sendo a força F relacionada à velocidade v por $F = cv$, em que c é uma constante, e a potência P dissipada dada por $P = cv^2$. Estes dois conjuntos de equações têm formas similares. Comparando-as, e tomando a corrente como sendo análoga à força, então a diferença de potencial é análoga à velocidade e a constante do amortecedor, c , ao inverso da resistência, ou seja, $(1/R)$. Estas analogias entre corrente e força, diferença de potencial e velocidade, são aplicáveis a outros blocos construtivos com a mola análoga à indutância e a massa, à capacitância.

O sistema mecânico visto na Figura 10.1(a) e o elétrico visto na Figura 10.1(b) têm relações de entrada/saída descritas por equações diferenciais similares:

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = F \quad \text{e} \quad RC \frac{dv_C}{dt} + LC \frac{d^2v_C}{dt^2} + v_C = v$$

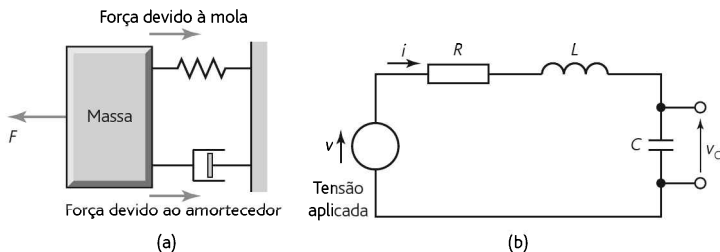


Figura 10.12 Sistemas análogos.

A analogia entre corrente e força é uma das mais usadas. Entretanto, outro conjunto de analogias pode ser obtido entre diferença de potencial e força.

10.4 Blocos construtivos de um sistema fluídico

Em um sistema de vazão de fluido existem três blocos construtivos básicos que podem ser considerados equivalentes à resistência elétrica, capacitância e indutância. Os sistemas fluídicos podem ser classificados em duas categorias: hidráulico, onde o fluido é um líquido e é considerado incompressível, e pneumático, onde se usa um gás que pode ser comprimido e, conseqüentemente, apresentar uma densidade variável.

A **resistência hidráulica** é a resistência à vazão que ocorre como resultado de um líquido passando por válvulas ou mudanças de diâmetro em tubulações (Figura 10.13(a)). A relação entre o volume da taxa de vazão do líquido q em um elemento de resistência e a diferença de pressão resultante $(p_1 - p_2)$ é

$$p_1 - p_2 = Rq$$

em que R é uma constante denominada resistência hidráulica. Quanto maior a resistência, maior a diferença de pressão para uma determinada vazão. Esta equação, assim como aquela para resistência elétrica e a lei de Ohm, tem uma relação linear. Esta resistência linear hidráulica ocorre com uma vazão ordenada pelos tubos capilares e obturadores permeáveis, porém resistências não lineares ocorrem com vazões através de orifícios com bordas acentuadas ou se a vazão é turbulenta.

A **capacitância hidráulica** é o termo usado para descrever a energia armazenada com um líquido onde ela é armazenada na forma de energia potencial. A altura do líquido em um recipiente (Figura 10.13(b)), ou seja, a altura de carga, como é mais conhecida, é uma forma deste armazenamento. Para esta capacitância, a taxa de variação do volume V no recipiente, ou seja, dV/dt , é igual à diferença entre a taxa volumétrica na qual o líquido entra no recipiente q_1 e a taxa na qual ele sai q_2 ,

$$q_1 - q_2 = \frac{dV}{dt}$$

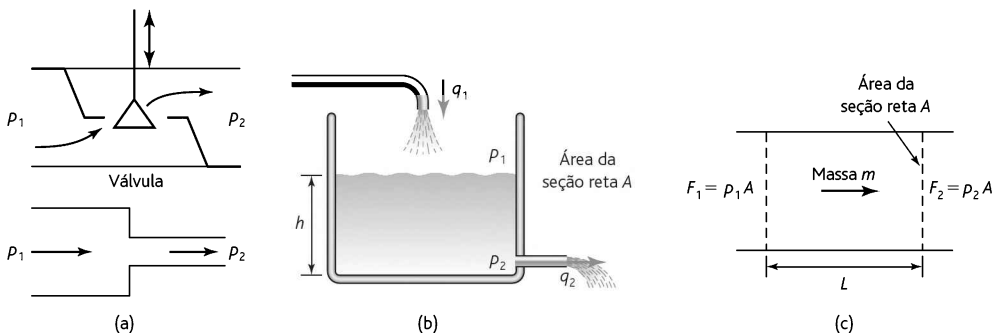


Figura 10.13 Exemplos hidráulicos: (a) resistência, (b) capacitância, (c) inertância.

Porém, $V = Ah$, em que A é a área da seção reta do recipiente e h é a altura do líquido nele. Assim,

$$q_1 - q_2 = \frac{d(Ah)}{dt} = A \frac{dh}{dt}$$

Porém, a diferença de pressão entre entrada e saída é p , em que $p = h\rho g$, sendo ρ a densidade do líquido e g a aceleração da gravidade. Portanto, se o líquido for considerado incompressível, ou seja, sua densidade não varia com a pressão,

$$q_1 - q_2 = A \frac{d(p/\rho g)}{dt} = \frac{A}{\rho g} \frac{dp}{dt}$$

A capacitância hidráulica C é definida como sendo

$$C = \frac{A}{\rho g}$$

Portanto,

$$q_1 - q_2 = C \frac{dp}{dt}$$

Integrando esta equação, obtemos

$$p = \frac{1}{C} \int (q_1 - q_2) dt$$

A **inertância hidráulica** é o equivalente da indutância nos sistemas elétricos ou a uma mola nos sistemas mecânicos. Para acelerar um fluido e, desta forma, aumentar sua velocidade, é necessário uma força. Considere um bloco de um líquido de massa m (Figura 10.13(c)). A força resultante que age no líquido é

$$F_1 - F_2 = p_1 A - p_2 A = (p_1 - p_2) A$$

em que $(p_1 - p_2)$ é a diferença de pressão e A é a área da seção reta. Esta força resultante faz com que a massa acelere com um valor a , e assim

$$(p_1 - p_2) A = ma$$

Mas a é a taxa de variação da velocidade dv/dt , portanto,

$$(p_1 - p_2) A = m \frac{dv}{dt}$$

Porém, a massa do líquido em questão tem um volume de AL , em que L é o comprimento do bloco de líquido ou a distância entre os pontos do líquido em que as pressões p_1 e p_2 foram medidas. Se o líquido tem uma densidade ρ , então, $m = AL\rho$, e assim

$$(p_1 - p_2) A = AL\rho \frac{dv}{dt}$$

Mas o volume da taxa de vazão $q = Av$, portanto

$$(p_1 - p_2)A = L\rho \frac{dq}{dt}$$

$$p_1 - p_2 = I \frac{dq}{dt}$$

em que a inercância hidráulica I é definida como

$$I = \frac{L\rho}{A}$$

Nos sistemas pneumáticos, os três blocos construtivos básicos são, assim como nos sistemas hidráulicos, a resistência, a capacitância e a inercância. Entretanto, os gases diferem dos líquidos por serem compressíveis, ou seja, uma variação na pressão provoca uma variação no volume e, desta forma, na densidade. A **resistência pneumática** R é definida em termos da massa da taxa de vazão dm/dt (observe que este termo é frequentemente escrito como um m com um ponto sobre a letra para indicar que o símbolo se refere à taxa de vazão da massa e não apenas à massa) e a diferença de pressão ($p_1 - p_2$) como

$$p_1 - p_2 = R \frac{dm}{dt} = R\dot{m}$$

A **capacitância pneumática** C é resultante da compressibilidade do gás, e é comparável com a forma com que a compressão de uma mola armazena energia. Se há uma taxa de vazão de massa dm_1/dt entrando em um recipiente de volume V e uma taxa de vazão de massa dm_2/dt saindo dele, então a taxa na qual a massa no recipiente varia é $(dm_1/dt - dm_2/dt)$. Se o gás no recipiente tiver uma densidade ρ , então a taxa de variação da massa no recipiente é

$$\text{taxa de variação da massa no recipiente} = \frac{d(\rho V)}{dt}$$

Porém, como o gás pode ser comprimido, ρ e V podem variar com o tempo. Assim,

$$\text{taxa de variação da massa no recipiente} = \rho \frac{dV}{dt} + V \frac{d\rho}{dt}$$

Visto que $(dV/dt) = (dV/dp)(dp/dt)$ e, para um gás ideal, $pV = mRT$ com consequentemente $p = (m/V)RT = \rho RT$ e $d\rho/dt = (1/RT)(dp/dt)$, então

$$\text{taxa de variação da massa no recipiente} = \rho \frac{dV}{dp} \frac{dp}{dt} + \frac{V}{RT} \frac{dp}{dt}$$

em que R é a constante do gás e T é a temperatura, que é considerada constante, na escala Kelvin. Portanto,

$$\frac{dm_1}{dt} - \frac{dm_2}{dt} = \left(\rho \frac{dV}{dp} + \frac{V}{RT} \right) \frac{dp}{dt}$$

A capacitância pneumática resultante da variação no volume do recipiente C_1 é definida como

$$C_1 = \rho \frac{dV}{dp}$$

e a capacitância pneumática resultante da compressibilidade do gás C_2 é definida como

$$C_2 = \frac{V}{RT}$$

Portanto,

$$\frac{dm_1}{dt} - \frac{dm_2}{dt} = (C_1 + C_2) \frac{dp}{dt}$$

ou

$$p_1 - p_2 = \frac{1}{C_1 + C_2} \int (\dot{m}_1 - \dot{m}_2) dt$$

A **inertância pneumática** é resultante da queda de pressão necessária para acelerar um bloco de gás. De acordo com a segunda lei de Newton, a força resultante é $ma = d(mv)/dt$. Visto que a força é proveniente da diferença de pressão ($p_1 - p_2$), então, se A é a área da seção reta do bloco de gás acelerado,

$$(p_1 - p_2)A = \frac{d(mv)}{dt}$$

Mas m , a massa do gás acelerado, é igual a ρLA , sendo ρ a densidade do gás e L o comprimento do bloco de gás acelerado. O volume da taxa de vazão $q = Av$, em que v é a velocidade. Portanto,

$$mv = \rho LA \frac{q}{A} = \rho Lq$$

e assim

$$(p_1 - p_2)A = L \frac{d(\rho q)}{dt}$$

Mas $\dot{m} = \rho q$ e, assim,

$$p_1 - p_2 = \frac{L}{A} \frac{d\dot{m}}{dt}$$

$$p_1 - p_2 = I \frac{d\dot{m}}{dt}$$

com a inertância pneumática I sendo $I = L/A$.

A Tabela 10.3 mostra as características básicas do bloco construtivo fluido, hidráulico e pneumático.

No sistema hidráulico, a taxa volumétrica da vazão, e, no pneumático, a taxa de vazão da massa são análogas à corrente no sistema elétrico. Nos sistemas hidráulico

Tabela 10.3 Blocos construtivos de sistemas hidráulico e pneumático

Bloco construtivo	Equação descritiva	Energia armazenada ou potência dissipada
<i>Hidráulica</i>		
Inertância	$q = \frac{1}{L} \int (p_1 - p_2) dt$ $p = L \frac{dq}{dt}$	$E = \frac{1}{2} L q^2$
Capacitância	$q = C \frac{d(p_1 - p_2)}{dt}$	$E = \frac{1}{2} C (p_1 - p_2)^2$
Resistência	$q = \frac{p_1 - p_2}{R}$	$P = \frac{1}{R} (p_1 - p_2)^2$
<i>Pneumático</i>		
Inertância	$\dot{m} = \frac{1}{L} \int (p_1 - p_2) dt$	$E = \frac{1}{2} L \dot{m}^2$
Capacitância	$\dot{m} = C \frac{d(p_1 - p_2)}{dt}$	$E = \frac{1}{2} C (p_1 - p_2)^2$
Resistência	$\dot{m} = \frac{p_1 - p_2}{R}$	$P = \frac{1}{R} (p_1 - p_2)^2$

e pneumático, a diferença de pressão é análoga à diferença de potencial no sistema elétrico. Compare a Tabela 10.3 com a Tabela 10.2. As inertâncias e capacitâncias hidráulica e pneumática são elementos armazenadores de energia; as resistências hidráulica e pneumática são dissipadores de energia.

10.4.1 Construção de um modelo para um sistema fluido

A Figura 10.14 mostra um sistema hidráulico simples, que é um líquido que entra e sai de um recipiente. Este sistema pode ser considerado como sendo constituído de um capacitor, o líquido no recipiente, com um resistor, a válvula.

A inertância pode ser ignorada, visto que a vazão varia muito lentamente. Para a capacitância, podemos escrever

$$q_1 - q_2 = C \frac{dp}{dt}$$

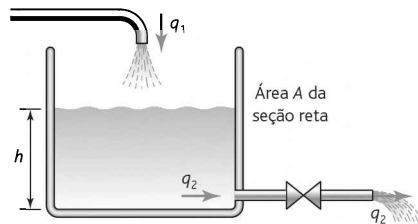


Figura 10.14 Um sistema fluido.

A taxa na qual o líquido sai do recipiente q_2 é igual à taxa na qual ele sai da válvula. Portanto, para o resistor

$$p_1 - p_2 = Rq_2$$

A diferença de pressão ($p_1 - p_2$) é a pressão resultante da altura da coluna de líquido no recipiente, que é $h\rho g$. Portanto, $q_2 = h\rho g / R$ e, substituindo q_2 na primeira equação, temos

$$q_1 - \frac{h\rho g}{R} = C \frac{d(h\rho g)}{dt}$$

e, como $C = A / \rho g$,

$$q_1 = A \frac{dh}{dt} + \frac{\rho gh}{R}$$

Esta equação descreve como a altura da coluna do líquido no recipiente depende da taxa de entrada do líquido no recipiente.

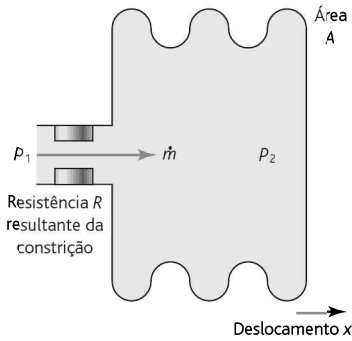


Figura 10.15 Sistema pneumático.

Um fole é um exemplo de um sistema pneumático simples (Figura 10.15). A resistência é resultante da uma constrição que restringe a taxa de vazão do gás para dentro do fole, e a capacitância é resultante do próprio fole. A inertância pode ser desprezada, visto que a taxa de vazão varia apenas lentamente.

A taxa de vazão da massa para dentro do fole é dada por

$$p_1 - p_2 = R\dot{m}$$

em que p_1 é a pressão antes da constrição e p_2 é a pressão após a constrição, ou seja, a pressão dentro do fole. Todo o gás que flui para dentro do fole permanece dentro dele sem sair. A capacitância do fole é dada por

$$\dot{m}_1 - \dot{m}_2 = (C_1 + C_2) \frac{dp_2}{dt}$$

A taxa de vazão da massa que entra no fole é dada pela equação da resistência, e a massa que sai do fole é zero. Portanto,

$$\frac{p_1 - p_2}{R} = (C_1 + C_2) \frac{dp_2}{dt}$$

Assim,

$$p_1 = R(C_1 + C_2) \frac{dp_2}{dt} + p_2$$

Esta equação descreve como a pressão dentro do fole, p_2 , varia com o tempo quando há uma pressão de entrada p_1 .

O fole expande ou contrai como resultado da variação da pressão dentro dele. O fole tem o formato de uma mola e, assim, podemos escrever $F = kx$ para a relação

entre a força F , que provoca a expansão ou a contração, e o deslocamento x resultante, em que k é a constante da mola para um fole. Porém, a força F depende da pressão p_2 , com $p_2 = F/A$, em que A é a área da seção reta do fole. Portanto, $p_2 A = F = kx$. Assim, substituindo p_2 na equação anterior, temos

$$p_1 = R(C_1 + C_2) \frac{k}{A} \frac{dx}{dt} + \frac{k}{A} x$$

Esta equação, que é diferencial de primeira ordem, descreve como a extensão ou a contração x das variações do fole muda com o tempo quando há uma pressão de entrada p_1 . A capacitância pneumática devido à variação no volume do recipiente C_1 é $\rho dV/dp_2$, e como $V = Ax$, C_1 é $\rho A dx/dp_2$, mas para o fole $p_2 A = kx$, portanto,

$$C_1 = \rho A \frac{dx}{d(kx/A)} = \frac{\rho A^2}{k}$$

C_2 , a capacitância pneumática devido à compressibilidade do ar, é $V/RT = Ax/RT$.

A seguir temos uma ilustração de como, no sistema hidráulico da Figura 10.16, podem ser derivadas relações que descrevem como as alturas da coluna de líquido em dois recipientes variam com o tempo. Neste modelo a inércia é desprezada. O recipiente 1 é um capacitor e, portanto,

$$q_1 - q_2 = C_1 \frac{dp}{dt}$$

em que $p = h_1 \rho g$ e $C_1 = A_1/\rho g$, e então

$$q_1 - q_2 = A_1 \frac{dh_1}{dt}$$

A taxa na qual o líquido sai do recipiente 1, q_2 , é igual à taxa na qual ele sai da válvula R_1 . Portanto, para o resistor,

$$p_1 - p_2 = R_1 q_2$$

As pressões são $h_1 \rho g$ e $h_2 \rho g$. Portanto,

$$(h_1 - h_2) \rho g = R_1 q_2$$

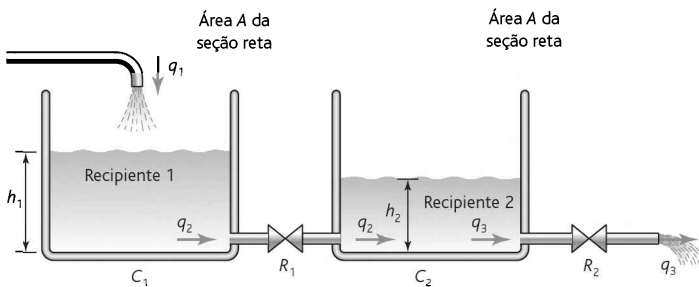


Figura 10.16 Um sistema flúidico.

Usando o valor de q_2 dado por esta equação e substituindo-o na equação anterior, obtemos

$$q_1 - \frac{(h_1 - h_2)\rho g}{R_1} = A_1 \frac{dh_1}{dt}$$

Esta equação descreve como a altura da coluna do líquido no recipiente 1 depende da vazão de entrada.

Pode-se deduzir um conjunto similar de equações para o recipiente 2. Assim, para o capacitor C_2 ,

$$q_2 - q_3 = C_2 \frac{dp}{dt}$$

em que $p = h_2\rho g$ e $C_2 = A_2/\rho g$, e assim

$$q_2 - q_3 = A_2 \frac{dh_2}{dt}$$

A taxa na qual o líquido sai do recipiente 2, q_3 , é igual à taxa na qual ele sai da válvula R_2 . Portanto, para o resistor,

$$p_2 - 0 = R_2 q_3$$

Esta equação considera que existe líquido na atmosfera. Portanto, usando o valor de q_3 dado por esta equação e substituindo-o na equação anterior, obtemos

$$q_2 - \frac{h_2\rho g}{R_2} = A_2 \frac{dh_2}{dt}$$

Substituindo q_2 nesta equação usando o valor dado pela equação deduzida para o primeiro recipiente, obtemos

$$\frac{(h_1 - h_2)\rho g}{R_1} - \frac{h_2\rho g}{R_2} = A_2 \frac{dh_2}{dt}$$

Esta equação descreve como a altura da coluna de líquido no recipiente 2 varia.

10.5 Blocos construtivos de um sistema térmico

Existem apenas dois blocos construtivos básicos para os sistemas térmicos: resistência e capacitância. Há um fluxo resultante de calor entre dois pontos se houver uma diferença de temperatura entre eles. O equivalente elétrico disto é que só existe uma corrente resultante i entre dois pontos se houver uma diferença de potencial v entre eles, sendo $i = v/R$ a relação entre a corrente e a diferença de potencial, em que R é a resistência elétrica entre os pontos. Uma relação similar pode ser usada para definir a **resistência térmica** R . Se q for a taxa de fluxo de calor e $(T_1 - T_2)$ for a diferença de temperatura, então

$$q = \frac{T_2 - T_1}{R}$$

O valor da resistência depende do modo da transferência de calor. No caso da condução de calor em sólidos, para uma condução unidirecional

$$q = Ak \frac{T_1 - T_2}{L}$$

em que A é a área da seção reta do material no qual o calor é conduzido e L é o comprimento do material entre os pontos nos quais as temperaturas são T_1 e T_2 ; k é a condutividade térmica. Portanto, com este modo de transferência de calor,

$$R = \frac{L}{Ak}$$

Quando o modo de transferência de calor é por convecção, como ocorre em líquidos e gases, então

$$q = Ah(T_2 - T_1)$$

em que A é a área da superfície na qual há a diferença de temperatura e h é o coeficiente de transferência de calor. Portanto, com este modo de transferência de calor,

$$R = \frac{1}{Ah}$$

A **capacitância térmica** é uma medida do armazenamento de energia em um sistema. Portanto, se a taxa de fluxo de calor para dentro de um sistema for q_1 e a taxa de fluxo de calor para fora for q_2 , então

$$\text{taxa de variação da energia interna} = q_1 - q_2$$

Um aumento da energia interna significa um aumento da temperatura. Visto que

$$\text{variação da energia interna} = mc \times \text{variação na temperatura}$$

em que m é a massa e c é a capacidade de calor específica, então

$$\text{taxa de variação da energia interna} = mc \times \text{taxa de variação da temperatura}$$

Portanto,

$$q_1 - q_2 = mc \frac{dT}{dt}$$

em que dT/dt é a taxa de variação da temperatura. Esta equação pode ser escrita como

$$q_1 - q_2 = C \frac{dT}{dt}$$

em que C é a capacitância térmica, e portanto $C = mc$. A Tabela 10.4 apresenta um resumo dos blocos construtivos térmicos.

10.5.1 Modelamento de um sistema térmico

Considere um termômetro à temperatura T que acaba de ser inserido em um líquido à temperatura T_L (Figura 10.17).

Tabela 10.4 Blocos construtivos térmicos

Bloco construtivo	Equação descritiva	Energia armazenada
Capacitância	$q_1 - q_2 = C \frac{dT}{dt}$	$E = CT$
Resistência	$q = \frac{T_1 - T_2}{R}$	

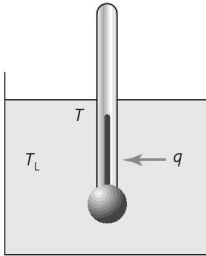


Figura 10.17 Um sistema térmico.

Se a resistência térmica ao fluxo de calor do líquido para o termômetro for R , então

$$q = \frac{T_L - T}{R}$$

em que q é a taxa resultante do fluxo de calor do líquido para o termômetro. A capacitância térmica C do termômetro é dada pela equação

$$q_1 - q_2 = C \frac{dT}{dt}$$

Visto que há apenas um fluxo de calor resultante do líquido para o termômetro, $q_1 = q$ e $q_2 = 0$. Portanto,

$$q = C \frac{dT}{dt}$$

Substituindo este valor de q na equação anterior, temos

$$C \frac{dT}{dt} = \frac{T_L - T}{R}$$

Isolando T_L , obtemos

$$RC \frac{dT}{dt} + T = T_L$$

Esta equação, uma equação diferencial de primeira ordem, descreve como a temperatura T , indicada pelo termômetro, varia com o tempo quando o termômetro é inserido no líquido quente.

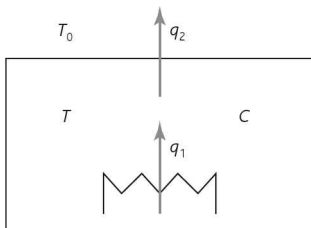


Figura 10.18 Sistema térmico.

Neste sistema térmico, os parâmetros foram considerados concentrados. Isto significa, por exemplo, que foram consideradas apenas as temperaturas do termômetro e do líquido, ou seja, as temperaturas são apenas funções do tempo, e não da localização do termômetro dentro do líquido.

Para ilustrar isto, considere a Figura 10.18, que mostra um sistema térmico que consiste em um fio elétrico em uma sala. O fio emite calor a uma taxa q_1 e a sala perde calor a uma taxa q_2 . Considerando que o ar dentro da sala está a uma temperatura uniforme T e que não há armazenamento de calor nas

paredes da sala, deduza uma equação que descreva como a temperatura da sala varia com o tempo.

Se o ar dentro da sala tem uma capacidade térmica C , então

$$q_1 - q_2 = C \frac{dT}{dt}$$

Se a temperatura interna da sala for T e a externa, T_0 , então

$$q_2 = \frac{T - T_0}{R}$$

em que R é a resistividade das paredes. Substituindo q_2 , temos

$$q_1 - \frac{T - T_0}{R} = C \frac{dT}{dt}$$

Portanto,

$$RC \frac{dT}{dt} + T = Rq_1 + T_0$$

Resumo

Um **modelo matemático** de um sistema é uma descrição deste em termos de equações que relacionam entradas e saídas de modo que as saídas podem ser determinadas a partir das entradas.

Os sistemas mecânicos podem ser considerados como constituídos de massas, molas e amortecedores, ou momentos de inércia, molas e amortecedores rotacionais, se os sistemas forem rotacionais. Os sistemas elétricos podem ser considerados constituídos de resistores, capacitores e indutores, os hidráulicos e pneumáticos constituídos de resistência, capacitância e inércia, e os térmicos constituídos de resistência e capacitância.

Existem muitos elementos nos sistemas mecânico, elétrico, fluídico e térmico que têm comportamentos similares. Portanto, por exemplo, a massa, nos sistemas mecânicos, tem propriedades similares à capacitância nos sistemas elétricos, à capacitância nos sistemas fluídicos e à capacitância nos sistemas térmicos. A Tabela 10.5 mostra uma comparação dos elementos em cada um destes sistemas e suas equações de definição.

Problemas

- 10.1 Deduza uma equação que relacione a entrada, força F , com a saída, deslocamento x , para o sistema descrito pela Figura 10.19.

Tabela 10.5 Elementos de sistemas

	Mecânico (translacional)	Mecânico (rotacional)	Elétrico	Fluídico (hidráulico)	Térmico
Elemento	Massa	Momento de inércia	Capacitor	Capacitor	Capacitor
Equação	$F = m \frac{d^2x}{dt^2}$ $F = m \frac{dv}{dt}$	$T = I \frac{d^2\theta}{dt^2}$ $T = I \frac{d\omega}{dt}$	$i = C \frac{dv}{dt}$	$q = C \frac{d(p_1 - p_2)}{dt}$	$q_1 - q_2 = C \frac{dT}{dt}$
Energia	$E = \frac{1}{2}mv^2$	$E = \frac{1}{2}I\omega^2$	$E = \frac{1}{2}Cv^2$	$E = \frac{1}{2}C(p_1 - p_2)^2$	$E = CT$
Elemento	Mola	Mola	Indutor	Inertância	Nenhum
Equação	$F = kx$	$T = k\theta$	$v = L \frac{di}{dt}$	$p = L \frac{dq}{dt}$	
Energia	$E = \frac{1}{2} \frac{F^2}{k}$	$E = \frac{1}{2} \frac{T^2}{k}$	$E = \frac{1}{2}Li^2$	$E = \frac{1}{2}Iq^2$	
Elemento	Amortecedor	Amortecedor rotacional	Resistor	Resistência	Resistência
Equação	$F = c \frac{dx}{dt} = c v$	$T = c \frac{d\theta}{dt} = c \omega$	$i = \frac{v}{R}$	$q = \frac{p_1 - p_2}{R}$	$q = \frac{T_1 - T_2}{R}$
Potência	$P = cv^2$	$P = c\omega^2$	$P = \frac{v^2}{R}$	$P = \frac{1}{R}(p_1 - p_2)^2$	

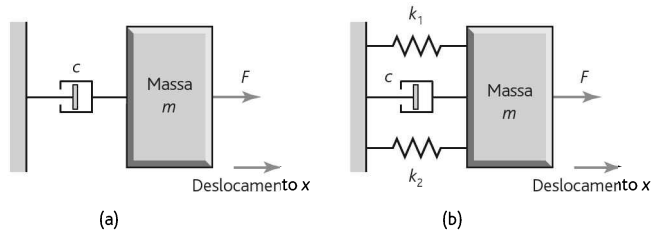


Figura 10.19 Problema 10.1.

10.2 Proponha um modelo para a roda metálica de um veículo ferroviário se deslocando na via férrea.

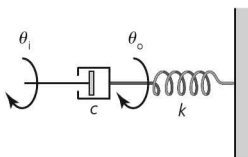


Figura 10.20 Problema 10.3.

10.3 Deduza uma equação que relacione o deslocamento angular de entrada θ_i com o deslocamento angular de saída θ_o para o sistema rotacional mostrado na Figura 10.20.

10.4 Proponha um modelo para um eixo de um motor de passo (ou seja, um eixo em que há uma variação em degrau no movimento angular) usado para girar uma massa e deduza uma equação que relacione o torque de entrada e a rotação angular. Você pode ignorar o amortecimento.

- 10.5 Deduza a relação entre a saída, a diferença de potencial no resistor R (v_R) e a entrada v para o circuito mostrado na Figura 10.21 o qual tem um resistor em série com um capacitor.

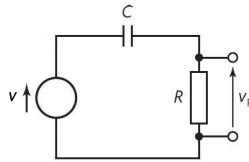


Figura 10.21 Problema 10.5.

- 10.6 Deduza a relação entre a saída, a diferença de potencial no resistor R (v_R) e a entrada v para o circuito RLC em série mostrado na Figura 10.22.

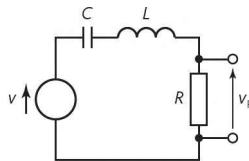


Figura 10.22 Problema 10.6.

- 10.7 Deduza a relação entre a saída, a diferença de potencial no capacitor C (v_C) e a entrada v para o circuito mostrado na Figura 10.23.

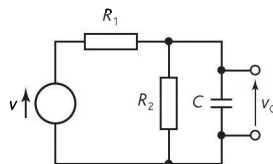


Figura 10.23 Problema 10.7.

- 10.8 Deduza a relação entre a altura h_2 e o tempo para o sistema hidráulico mostrado na Figura 10.24. Ignore a inércia.

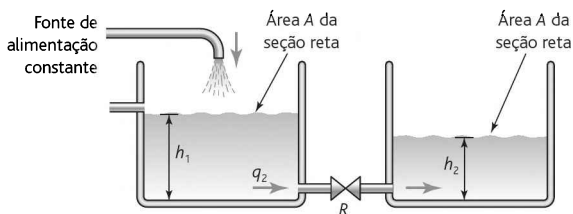


Figura 10.24 Problema 10.8.

- 10.9 Um objeto quente, de capacitância C e temperatura T , esfria em uma sala ampla que se encontra à temperatura T_r . Se o sistema térmico tiver uma resistência R , deduza uma equação que descreva como a temperatura do objeto quente varia com o tempo e determine o análogo elétrico deste sistema.
- 10.10 A Figura 10.25 mostra um sistema térmico que envolve dois compartimentos, sendo que um deles contém um aquecedor. Se a temperatura do compartimento que contém o aquecedor for T_1 , a temperatura do outro compartimento for T_2 e a temperatura externa em torno dos compartimentos for T_3 , desenvolva as equações que descrevem como as temperaturas T_1 e T_2 variam com o tempo. Todas as paredes dos recipientes têm a mesma resistência, e a capacitância pode ser desprezada. Os dois recipientes têm a mesma capacitância C .

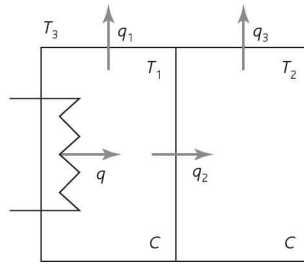


Figura 10.25 Problema 10.10.

- 10.11 Deduza a equação diferencial que relaciona a pressão de entrada p sobre o diafragma (como na Figura 7.23) ao deslocamento x da haste.
- 10.12 Deduza a equação diferencial para um motor que aciona uma carga através de um sistema de engrenagens (Figura 10.26) que relaciona o deslocamento angular da carga com o tempo.

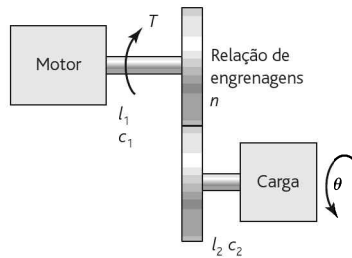


Figura 10.26 Problema 10.12.

11

Modelos de Sistemas

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Projetar modelos para sistemas mecânico-hidráulico e eletromecânico rotacional-translacional.
- Linearizar relações não lineares para gerar modelos lineares.

11.1 Sistemas de engenharia

No Capítulo 10, os blocos construtivos básicos dos sistemas mecânico translacional, mecânico rotacional, elétrico, fluídico e térmico foram estudados separadamente. Entretanto, muitos sistemas encontrados em engenharia envolvem aspectos de mais que uma destas disciplinas. Por exemplo, um motor elétrico envolve elementos elétricos e mecânicos. Este capítulo aborda como blocos construtivos de apenas uma área de conhecimento podem ser combinados para produzir modelos e sistemas multidisciplinares, e também discute a questão de alguns componentes não serem lineares. Por exemplo, ao considerar uma mola, um modelo simples parte do princípio de que a força e a extensão são proporcionais, independentemente da intensidade da força. O modelo matemático pode ser então uma simplificação de uma mola real. Entretanto, os modelos não lineares são de manipulação muito mais difícil, de forma que os engenheiros procuram evitá-los, e um sistema não linear pode ser aproximado por um modelo linear.

11.2 Sistemas rotacional-translacional

Existem muitos mecanismos que envolvem a conversão do movimento rotacional para o movimento translacional ou vice-versa. Como exemplo temos cremalheira e pinhão, eixos com parafuso de avanço, sistemas de polias e cabos, etc.

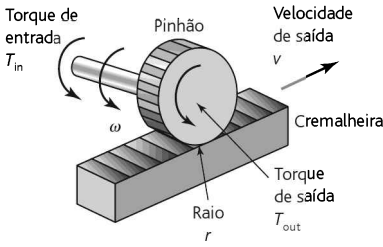


Figura 11.1 Cremalheira e pinhão.

Para ilustrar como estes sistemas podem ser analisados, considere o sistema de cremalheira e pinhão mostrado na Figura 11.1. O movimento rotacional do pinhão é transformado em movimento de translação da cremalheira. Considere primeiro o elemento pinhão. A ação de torque resultante nele é $(T_{in} - T_{out})$. Portanto, considerando o momento do elemento de inércia, e desprezando o amortecimento,

$$T_{in} - T_{out} = I \frac{d\omega}{dt}$$

em que I é o momento de inércia do pinhão e ω é a velocidade angular. A rotação do pinhão resulta em uma velocidade de translação v da cremalheira. Se o pinhão tem um raio r , então $v = r\omega$. Portanto, podemos escrever

$$T_{in} - T_{out} = \frac{I}{r} \frac{dv}{dt}$$

Agora considere o elemento cremalheira. Existe uma força de T/r agindo nela devido ao movimento do pinhão. Se existir uma força de fricção cv , então a força resultante é

$$\frac{T_{out}}{r} - cv = m \frac{dv}{dt}$$

Eliminando T_{out} das duas equações, obtemos

$$T_{in} - rcv = \left(\frac{I}{r} + mr \right) \frac{dv}{dt}$$

e assim

$$\frac{dv}{dt} = \left(\frac{r}{1 + mr^2} \right) (T_{in} - rcv)$$

O resultado é uma equação diferencial de primeira ordem que descreve como a saída está relacionada à entrada.

11.3 Sistemas eletromecânicos

Os dispositivos eletromecânicos, como os potenciômetros, motores e geradores, transformam sinais elétricos em movimento rotacional e vice-versa. Esta seção apresenta uma discussão de como podemos deduzir modelos para tais sistemas. Um potenciômetro tem uma entrada na forma de uma rotação e uma saída na forma de uma diferença de potencial. Um motor elétrico tem uma entrada na forma de diferença de potencial e uma saída na forma de rotação de um eixo. Um gerador tem uma entrada na forma de rotação de um eixo e uma saída na forma de diferença de potencial.

11.3.1 Potenciômetro

O **potenciômetro de rotação** (Figura 11.2) é um divisor de potencial, e portanto

$$\frac{v_o}{V} = \frac{\theta}{\theta_{\text{máx}}}$$

em que V é a diferença de potencial no comprimento máximo da trilha do potenciômetro e $\theta_{\text{máx}}$ é o ângulo total de giro do cursor de uma extremidade a outra da trilha. A saída é v_o para a entrada θ .

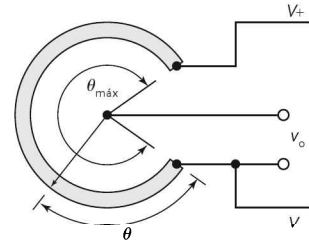


Figura 11.2 Potenciômetro rotativo.

11.3.2 Motor CC

Em um motor CC que é usado para converter um sinal de entrada elétrico em um sinal de saída mecânico, uma corrente na bobina de armadura do motor resulta em uma rotação do eixo e, portanto, na rotação da carga (Figura 11.3).

O motor consiste basicamente de uma bobina, a de armadura, a qual é livre para girar. Esta bobina situa-se em um campo magnético gerado por uma corrente nas bobinas de campo ou por um ímã permanente. Quando uma corrente i_a circula pela bobina de armadura, então, devido ao seu campo magnético, forças agem na bobina, fazendo-a girar (Figura 11.4). A força F que age nos fios de comprimento L percorridos pela corrente i_a em um campo magnético de densidade B e que forma um ângulo reto com o fio é dada por $F = Bi_aL$, e para N fios é $F = NBi_aL$. As forças nos fios da bobina da armadura resulta em um torque T , onde $T = Fb$, em que b é a largura da bobina. Portanto,

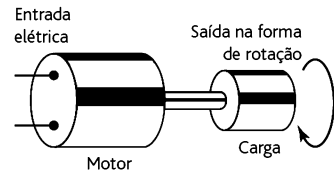


Figura 11.3 Motor acionando uma carga.

$$T = NBi_aLb$$

Portanto, o torque resultante é proporcional a (Bi_a) , sendo os outros fatores constantes. Assim, podemos escrever

$$T = k_1Bi_a$$

Como a armadura é uma bobina que gira em um campo magnético, uma tensão é induzida nela como consequência de indução eletromagnética. Esta tensão tem um sentido oposto à variação que a produziu, e é denominada FEM contrária (v_b). Ela é proporcional à variação ou à rotação da armadura e o fluxo concatenado pela bobina, daí a densidade de fluxo B . Portanto,

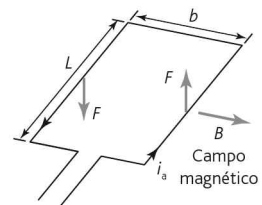


Figura 11.4 Um fio da bobina da armadura.

$$v_b = k_2B\omega$$

em que ω é a velocidade angular do eixo e k_2 é uma constante.

Considere um motor CC que tem bobinas de armadura e campo com excitação em separado. Em um **motor controlado pela armadura**, a corrente de campo i_f é mantida constante e o motor é controlado pelo ajuste da tensão de armadura (v_a).

Uma corrente de campo constante significa uma densidade de fluxo B constante para a bobina de armadura. Portanto,

$$v_b = k_2 B \omega = k_3 \omega$$

em que k_3 é uma constante. O circuito da armadura pode ser considerado como uma resistência R_a em série com uma indutância L_a (Figura 11.5).

Se v_a é a tensão aplicada no circuito da armadura, então, como há uma FEM contrária (v_b), temos

$$v_a - v_b = L_a \frac{di_a}{dt} + R_a i_a$$

Podemos pensar nesta equação em termos do diagrama em bloco mostrado na Figura 11.6(a). A entrada para o motor no sistema é v_a , e esta é somada com o sinal de realimentação da FEM contrária (v_b), gerando um sinal de erro que é a entrada do circuito de armadura. A equação anterior descreve a relação entre a entrada

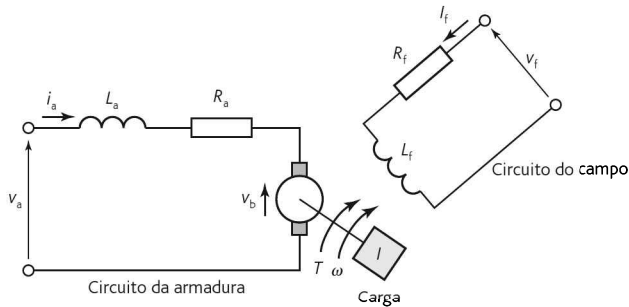


Figura 11.5 Circuitos em um motor CC.

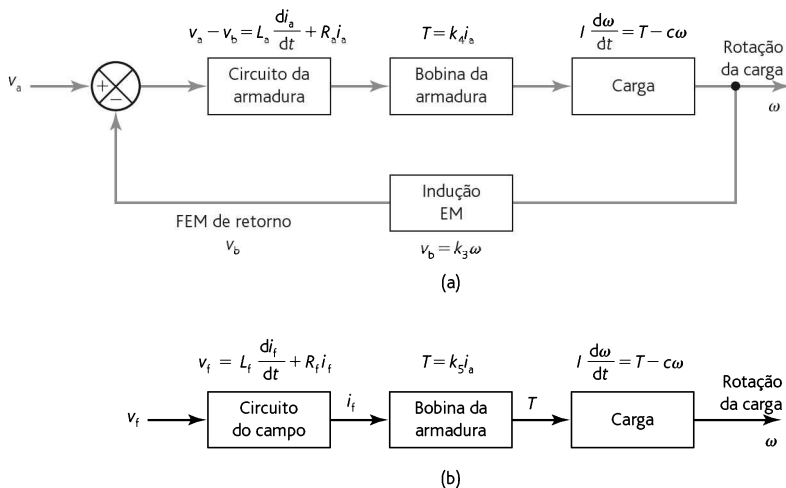


Figura 11.6 Motores CC: (a) controlado pela armadura, (b) controlado pelo campo.

do sinal de erro para a bobina de armadura e a saída da corrente de armadura (i_a). Substituindo v_b ,

$$v_a - k_3\omega = L_a \frac{di_a}{dt} + R_a i_a$$

A corrente i_a na armadura gera um torque T . Visto que, para um motor controlado pela armadura, B é constante, temos

$$T = k_1 B i_a = k_4 i_a$$

em que k_4 é uma constante. Este torque torna-se então a entrada para o sistema da carga. O torque resultante que age na carga é

$$\text{torque resultante} = T - \text{torque de amortecimento}$$

O torque de amortecimento é $c\omega$, em que c é uma constante. Portanto, se quaisquer efeitos devidos à elasticidade da torção do eixo forem ignorados,

$$\text{torque resultante} = k_4 i_a - c\omega$$

Isto provoca uma aceleração angular de $d\omega/dt$, portanto

$$I \frac{d\omega}{dt} = k_4 i_a - c\omega$$

Temos então duas equações que descrevem as condições que ocorrem para um motor controlado pela armadura, isto é,

$$v_a - k_3\omega = L_a \frac{di_a}{dt} + R_a i_a \quad \text{e} \quad I \frac{d\omega}{dt} = k_4 i_a - c\omega$$

Podemos então obter a equação resultante da saída ω com a entrada v_a para o sistema eliminando i_a . Veja uma breve discussão da transformada de Laplace no Capítulo 13, ou veja o Apêndice A para os detalhes de como isto pode ser feito.

No caso de um **motor controlado pelo campo**, a corrente de armadura é mantida constante e o motor é controlado pela variação da tensão de campo. No circuito do campo (Figura 11.5) há essencialmente apenas a indutância L_f em série com a resistência R_f . Portanto, para este circuito,

$$v_f = R_f i_f + L_f \frac{di_f}{dt}$$

Podemos pensar no motor controlado pelo campo em termos do diagrama em bloco mostrado na Figura 11.6(b). A entrada do sistema é v_f . O circuito de campo converte esta tensão em uma corrente i_f , sendo que a relação entre v_f e i_f é a equação anterior. Esta corrente conduz à produção de um campo magnético e, portanto, uma ação de torque na bobina de armadura, conforme dado por $T = k_1 B i_a$. Mas a densidade de fluxo B é proporcional à corrente de campo (i_f) e i_a é constante, assim,

$$T = k_1 B i_a = k_5 i_f$$

em que k_5 é uma constante. Este torque de saída é então convertido pelo sistema da carga em velocidade angular (ω). Como anteriormente, a ação de torque resultante na carga é

$$\text{torque resultante} = T - \text{torque de amortecimento}$$

O torque de amortecimento é $c\omega$, em que c é uma constante. Portanto, se algum efeito devido à elasticidade da torção do eixo for ignorado,

$$\text{torque resultante} = k_s i_f - c\omega$$

Isto provoca uma aceleração angular $d\omega/dt$, assim,

$$I \frac{d\omega}{dt} = k_s i_f - c\omega$$

As condições que ocorrem para um motor controlado pelo campo são descritas pelas equações

$$v_f = R_f i_f + L_f \frac{di_f}{dt} \quad \text{e} \quad I \frac{d\omega}{dt} = k_s i_f - c\omega$$

Podemos então obter a equação que relaciona a saída ω com a entrada v_f para o sistema eliminando i_f . Veja a breve discussão da transformada de Laplace no Capítulo 13, ou o Apêndice A para mais detalhes de como isto pode ser feito.

11.4 Linearidade

Ao combinar blocos para criar modelos de sistemas, estamos considerando que a relação para cada bloco é linear. A abordagem a seguir é uma breve discussão sobre a linearidade e como precisamos fazer uma aproximação linear para um item não linear, pois muitos deles são não lineares.

A relação entre a força F e a extensão x produzida por uma mola ideal é linear, sendo dada por $F = kx$. Isto significa que se a força F_1 produz uma extensão x_1 , e uma força F_2 produz uma extensão x_2 , uma força igual a $(F_1 + F_2)$ produzirá uma extensão $(x_1 + x_2)$. Este é o **princípio da superposição** e é uma condição necessária para um sistema poder ser denominado **sistema linear**. Outra condição para um sistema ser linear é que se uma entrada F_1 produz uma extensão x_1 , então uma entrada cF_1 produzirá uma saída cx_1 , em que c é uma constante multiplicadora.

O gráfico da força F pela extensão x é uma linha reta que passa na origem quando a relação é linear (Figura 11.7(a)). As molas reais, assim como quaisquer outros componentes reais, não são perfeitamente lineares (Figura 11.7(b)). Entretanto, geralmente existe uma faixa de operação que pode ser considerada linear. Assim, para uma mola que tem um gráfico como o da Figura 11.7(b), a linearidade da mola pode ser considerada se a mesma for usada apenas na parte central do gráfico. Para muitos componentes de sistemas, a linearidade pode ser considerada para operações dentro de uma faixa de valores da variável em torno de algum ponto de operação.

Para alguns componentes de sistemas, a relação é não linear (Figura 11.8(a)). Para estes componentes, o melhor que se pode fazer para se obter uma relação linear é trabalhar com a linha reta com a inclinação do gráfico no ponto de operação.

Portanto, para a relação entre y e x na Figura 11.8(a), no ponto de operação P onde a inclinação tem valor m , temos

$$\Delta y = m \Delta x$$

em que Δy e Δx são variações pequenas nos sinais de entrada e saída no ponto de operação.

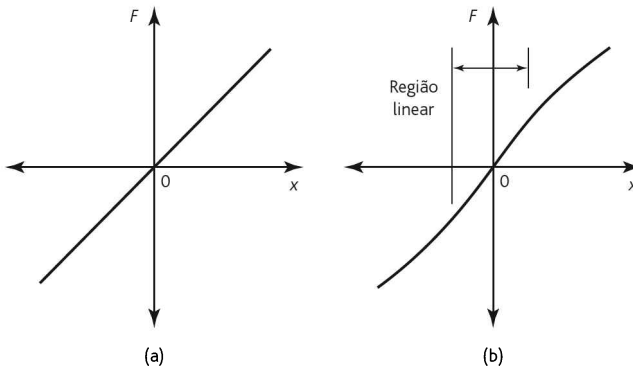


Figura 11.7 Molas: (a) ideal, (b) real.

Por exemplo, a taxa de vazão q de um líquido ao passar em um orifício é dada por

$$q = c_d A \sqrt{\frac{2(p_1 - p_2)}{\rho}}$$

em que c_d é uma constante denominada coeficiente de descarga, A é a área da seção reta do orifício, ρ é a densidade do fluido e $(p_1 - p_2)$ é a diferença de pressão. Para uma densidade e uma área de seção reta constantes, a equação pode ser escrita como

$$q = C \sqrt{p_1 - p_2}$$

em que C é uma constante. Esta é uma relação não linear entre a taxa de vazão e a diferença de pressão. Podemos obter uma relação linear considerando a linha reta que representa a inclinação do gráfico taxa de vazão–diferença de pressão (Figura 11.8(b)) no ponto de operação. A inclinação m é $dq/d(p_1 - p_2)$ e vale

$$m = \frac{dq}{d(p_1 - p_2)} = \frac{C}{2 \sqrt{p_{o1} - p_{o2}}}$$

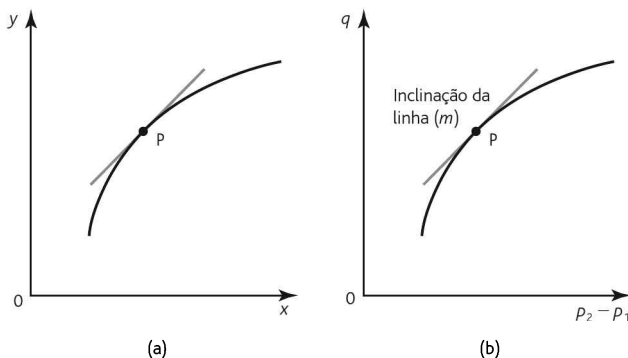


Figura 11.8 Uma relação não linear.

em que $(p_{o1} - p_{o2})$ é o valor no ponto de operação. Para variações pequenas em torno do ponto de operação, vamos assumir a substituição do gráfico não linear pela reta de inclinação m , e portanto podemos escrever $m = \Delta q / \Delta(p_1 - p_2)$. Assim,

$$\Delta q = m \Delta(p_1 - p_2)$$

Portanto, se tivermos $C = 2 \text{ m}^3/\text{s}$ por kPa, ou seja, $q = 2(p_1 - p_2)$, então, para um ponto de operação de $(p_1 - p_2) = 4 \text{ kPa}$ com $m = 2 / (2 \sqrt{4}) = 0,5$, a versão linearizada da equação seria

$$\Delta q = 0,5 \Delta(p_1 - p_2)$$

Modelos matemáticos linearizados são usados porque a maioria das técnicas dos sistemas de controle são baseadas em relações lineares entre os elementos destes sistemas. Além disto, como a maioria dos sistemas de controle mantêm uma saída igual a algum valor de referência, as variações a partir deste valor tendem a ser pequenas e, desta forma, o modelo linearizado é perfeitamente apropriado.

11.5 Sistemas hidráulico-mecânico

Os conversores hidráulico-mecânico envolvem a transformação de sinais hidráulicos em movimento de translação ou de rotação, ou vice-versa. Portanto, por exemplo, o movimento de um pistão em um cilindro, como resultado de uma pressão hidráulica, envolve a transformação de uma pressão hidráulica na entrada do sistema em um movimento de translação na saída.

A Figura 11.9 mostra um sistema hidráulico no qual uma entrada de deslocamento x_i é, após passar pelo sistema, transformada em um deslocamento x_o de uma carga. O sistema consiste de uma válvula tipo carretel e um cilindro. O deslocamento de entrada x_i para a esquerda resulta em uma pressão de alimentação do fluido hidráulico p_s que faz com que o fluido flua da esquerda para a direita no cilindro. Isto empurra o pistão para a direita no cilindro e expulsa o fluido da direita para a esquerda na câmara através da porta de saída na extremidade direita da válvula tipo carretel.

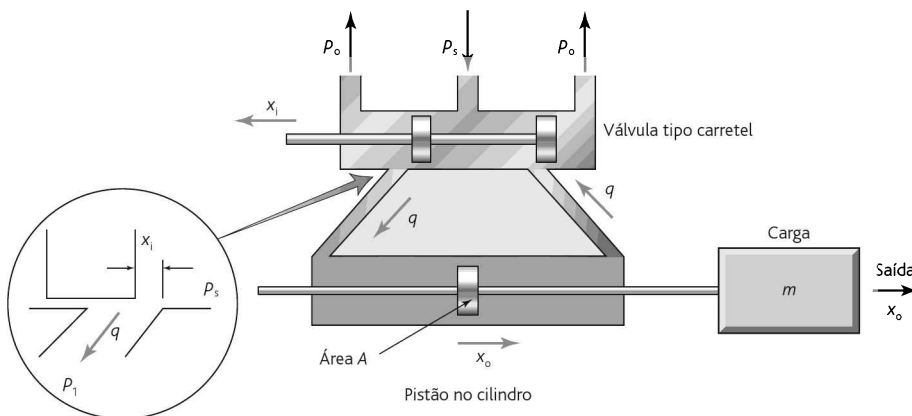


Figura 11.9 Sistema hidráulico com carga.

A taxa da vazão do fluido para dentro e para fora da câmara depende da extensão na qual o movimento de entrada descobre as portas permitindo que o fluido entre ou saia da válvula tipo carretel. Quando o deslocamento de entrada x_i for para a direita, a válvula tipo carretel permite que o fluido se mova para a extremidade direita do cilindro e, desta forma, o pistão no cilindro se move para a esquerda.

A vazão q do fluido através do orifício, que é o que as portas da válvula tipo-carretel são, é uma relação não linear que depende da diferença de pressão entre os dois lados do orifício e da área A de sua seção reta. Entretanto, uma versão linearizada da equação pode ser usada (veja a seção anterior para sua dedução).

$$\Delta q = m_1 \Delta A + m_2 \Delta (\text{diferença de pressão})$$

em que m_1 e m_2 são constantes no ponto de operação. Para o fluido que entra na câmara, a diferença de pressão é $(p_s - p_1)$ e para a saída é $(p_2 - p_0)$. Se o ponto de operação em torno do qual a equação é linearizada for o centro da válvula tipo carretel, e as portas de conexão com o cilindro estiverem fechadas, então nesta condição q é zero, e desta forma $\Delta q = q$, A é proporcional a x_s se este for medido a partir da posição central, e a variação na pressão no lado de entrada do pistão é $-\Delta p_1$ em relação a p_s e no lado de saída é Δp_2 em relação a p_0 . Portanto, para a porta de entrada, a equação pode ser escrita como

$$q = m_1 x_i + m_2 (-\Delta p_1)$$

e, para a porta de saída, como

$$q = m_1 x_i + m_2 \Delta p_2$$

Somando estas duas equações, obtemos

$$2q = 2m_1 x_i - m_2 (\Delta p_1 - \Delta p_2)$$

$$q = m_1 x_i - m_3 (\Delta p_1 - \Delta p_2)$$

em que $m_3 = m_2/2$.

Para o cilindro, a variação no volume do fluido que entra na câmara pela esquerda, ou a deixa pela direita, quando o pistão é movido uma distância x_o é Ax_o , em que A é a área da seção reta do pistão. Portanto, a taxa na qual o volume varia é $A(dx_o/dt)$. A taxa na qual o fluido entra à esquerda no cilindro é q . Entretanto, como existe algum vazamento do fluido de um lado do pistão para o outro,

$$q = A \frac{dx_o}{dt} + q_L$$

em que q_L é a taxa de vazamento. Substituindo q , obtemos

$$m_1 x_i - m_3 (\Delta p_1 - \Delta p_2) = A \frac{dx_o}{dt} + q_L$$

A taxa de vazamento q_L é uma vazão através de um orifício, o espaço (*gap*) entre o pistão e o cilindro. Este tem uma seção reta constante e tem uma diferença de pressão $(\Delta p_1 - \Delta p_2)$. Portanto, usando uma equação linearizada para esta vazão,

$$q_L = m_4 (\Delta p_1 - \Delta p_2)$$

Assim, usando esta equação para substituir q_1 ,

$$m_1 x_i - m_3 (\Delta p_1 - \Delta p_2) = A \frac{dx_o}{dt} + m_4 (\Delta p_1 - \Delta p_2)$$

$$m_1 x_i - (m_3 + m_4) (\Delta p_1 - \Delta p_2) = A \frac{dx_o}{dt}$$

A diferença de pressão no pistão resulta em uma força exercida na carga de $(\Delta p_1 - \Delta p_2)A$. Entretanto, há algum amortecimento do movimento, ou seja, fricção, da massa. Este é proporcional à velocidade da massa, ou seja, (dx_o/dt) . Portanto, a força resultante que age na carga é

$$\text{força resultante} = (\Delta p_1 - \Delta p_2)A - c \frac{dx_o}{dt}$$

Esta força resultante faz com que a massa acelere, sendo este valor $(d^2 x_o/dt^2)$. Portanto,

$$m \frac{d^2 x_o}{dt^2} = (\Delta p_1 - \Delta p_2)A - c \frac{dx_o}{dt}$$

Isolando a diferença de pressão, obtemos

$$\Delta p_1 - \Delta p_2 = \frac{m}{A} \frac{d^2 x_o}{dt^2} + \frac{c}{A} \frac{dx_o}{dt}$$

Usando esta equação para substituir a diferença de pressão na equação anterior, temos

$$m_1 x_i - (m_3 + m_4) \left(\frac{m}{A} \frac{d^2 x_o}{dt^2} + \frac{c}{A} \frac{dx_o}{dt} \right) = A \frac{dx_o}{dt}$$

Rearranjando a equação, temos

$$\frac{(m_3 + m_4)m}{A} \frac{d^2 x_o}{dt^2} + \left(A + \frac{c(m_3 + m_4)}{A} \right) \frac{dx_o}{dt} = m_1 x_i$$

e rearranjando novamente obtemos

$$\frac{(m_3 + m_4)m}{A^2 + c(m_3 + m_4)} \frac{d^2 x_o}{dt^2} + \frac{dx_o}{dt} = \frac{Am_1}{A^2 + c(m_3 + m_4)} x_i$$

Esta equação pode ser simplificada introduzindo-se duas constantes, k e τ , sendo esta última denominada constante e tempo (veja o Capítulo 12). Portanto,

$$\tau \frac{d^2 x_o}{dt^2} + \frac{dx_o}{dt} = k x_i$$

Assim, a relação entre entrada e saída é descrita por uma equação diferencial de segunda ordem.

Resumo

Muitos sistemas encontrados na engenharia envolvem aspectos que abrangem mais de uma disciplina, e estes podem ser considerados pela análise de como o sistema pode ser construído a partir de blocos construtivos de uma única disciplina.

Diz-se que um sistema é linear quando suas equações básicas, algébricas ou diferenciais, têm saídas diretamente proporcionais às entradas. Para uma equação algébrica, isto significa que o gráfico da saída versus entrada é uma linha reta que passa na origem. Assim, dobrando-se a entrada, a saída é dobrada. Para um sistema linear, podemos obter a saída dele para várias entradas somando as saídas do sistema em resposta a cada entrada individual considerada separadamente. Este procedimento é denominado **princípio da superposição**.

Problemas

- 11.1 Deduza uma equação diferencial que relacione a tensão de entrada de um motor CC servo-assistido e a velocidade angular de saída, considerando que o motor seja controlado pela armadura e que o circuito equivalente para o motor tenha uma armadura com resistência apenas (sua indutância é ignorada).
- 11.2 Deduza as equações diferenciais para um gerador CC. Este pode ser considerado como tendo um campo magnético constante. O circuito da armadura tem uma bobina com resistência e indutância em série com a carga. Considere que a carga tem também resistência e indutância.
- 11.3 Deduza as equações diferenciais para o motor CC de ímã permanente.

12

Resposta Dinâmica de Sistemas

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Modelar sistemas dinâmicos pelo método de equações diferenciais.
- Determinar as saídas de sistemas de primeira ordem a partir das entradas e também suas constantes de tempo.
- Determinar as saídas de sistemas de segunda ordem a partir das entradas e identificar as condições de subamortecido, criticamente amortecido e sobreamortecido.
- Descrever as características de resposta de um sistema de segunda ordem em termos do tempo de subida, *overshoot*, razão de amortecimento, decremento e tempo de acomodação.

12.1 Modelagem de sistemas dinâmicos

A função mais importante de um modelo idealizado para sistemas de controle ou medição é a capacidade de prever qual será a saída para uma determinada entrada. Não estamos nos referindo a uma situação estática, ou seja, em que, após algum tempo, quando o estado estacionário for atingido, uma saída x corresponde a uma entrada y . Estamos considerando como a saída irá variar com o tempo quando houver uma variação na entrada ou quando a entrada varia com o tempo. Por exemplo, como a temperatura de um sistema com temperatura controlada varia com o tempo quando o termostato é ajustado para um novo valor de temperatura? Para um sistema de controle, como a saída varia com o tempo quando um novo valor é ajustado ou aumentado a uma taxa constante?

Os Capítulos 10 e 11 abordaram os modelos de sistemas em que as entradas variam com o tempo, com os resultados expressos em termos de equações diferenciais. Este capítulo aborda como podemos usar estes modelos para fazer previsões de como as saídas variam com o tempo quando as entradas também variam.

12.1.1 Equações diferenciais

Para descrever a relação entre a entrada de um sistema e sua saída, devemos descrever a relação entre entradas e saídas que são funções do tempo. Precisamos então definir a equação que indica como a saída do sistema varia com o tempo quando a entrada varia também. Isto pode ser feito usando-se uma equação diferencial. Esta equação inclui derivadas em relação ao tempo e, desta forma, fornece informações sobre como a resposta de um sistema varia com o tempo. Uma derivada dx/dt descreve a taxa na qual x varia com o tempo; a derivada d^2x/dt^2 informa como dx/dt varia com o tempo. As equações diferenciais podem ser classificadas como de primeira ordem, segunda ordem, terceira ordem, etc. de acordo com a mais alta ordem da derivada na equação. Para uma equação de primeira ordem, a maior ordem é dx/dt ; para uma de segunda ordem, d^2x/dt^2 ; para uma de terceira ordem, d^3x/dt^3 ; para uma de ordem n , $d^n x/dt^n$.

Este capítulo aborda os tipos de respostas que podemos esperar de sistemas de primeira e segunda ordens e a solução destas equações diferenciais para que a resposta do sistema a diferentes tipos de entrada possa ser obtida. Este capítulo usa o método de “testar uma solução” para resolver a equação; o método da transformação de Laplace é apresentado no Capítulo 13.

12.2 Terminologia

Nesta seção focamos alguns termos que são usados ao se descrever a resposta dinâmica dos sistemas.

12.2.1 Respostas natural e forçada

O termo **resposta natural** é usado para um sistema quando não há entrada no sistema forçando a variável mudar, mas ela varia apenas naturalmente. Como exemplo, considere o sistema de primeira ordem em que a água sai naturalmente de um tanque (Figura 12.1(a)).

Para este sistema, temos

$$p_1 - p_2 = Rq$$

em que R é a resistência hidráulica. Porém, $p_1 - p_2 = h\rho g$, em que ρ é a densidade da água e q é a taxa na qual a água sai do tanque, que é igual a $-dV/dt$, em que V é

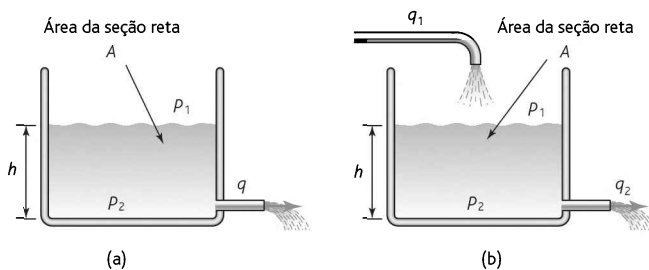


Figura 12.1 Água saindo de um tanque: (a) naturalmente sem entrada, (b) com uma entrada forçando.

o volume da água no tanque é igual a Ah . Portanto, $q = -d(Ah)/dt = -Adh/dt$ e, assim, a equação anterior pode ser escrita como

$$h\rho g = -RA \frac{dh}{dt}$$

Esta é a resposta natural para a qual não há entrada no sistema forçando a variável h a mudar; ela varia naturalmente com o tempo. Podemos chamar a atenção para isto escrevendo a equação diferencial com todos os termos de saída, ou seja, h , do mesmo lado da igualdade, e o termo de entrada, zero, do outro lado, ou seja,

$$RA \frac{dh}{dt} + (\rho g)h = 0$$

Na Seção 10.4.1, a equação diferencial foi deduzida para um tanque de água a partir do qual havia uma vazão de água saindo, porém havia também uma vazão de água entrando no tanque (Figura 12.1(b)). Esta equação tem uma função de entrada forçada, q_1 , e pode ser escrita como

$$RA \frac{dh}{dt} + (\rho g)h = q_1$$

Como outro exemplo, considere um termômetro colocado em um líquido quente em uma temperatura T_L . A taxa na qual a leitura do termômetro T varia com o tempo foi deduzida na Seção 10.5.1 como sendo dada pela equação diferencial

$$RC \frac{dT}{dt} + T = T_L$$

Esta equação tem uma entrada forçada T_L .

12.2.2 Respostas transiente e estacionária

A resposta total de um sistema de controle, ou de um elemento do sistema, pode ser considerada como sendo constituída de duas partes, a de estado estacionário e a transiente. A **resposta transiente** é a parte da resposta de um sistema que ocorre como resultado de uma variação na entrada e que termina após um curto intervalo de tempo. A **resposta de estado estacionário** é aquela que permanece após a resposta transiente ter terminado.

Considere, como uma simples ilustração disto, uma mola suspensa verticalmente (Figura 12.2) e o que acontece quando, de repente, um peso passa a ser sus-

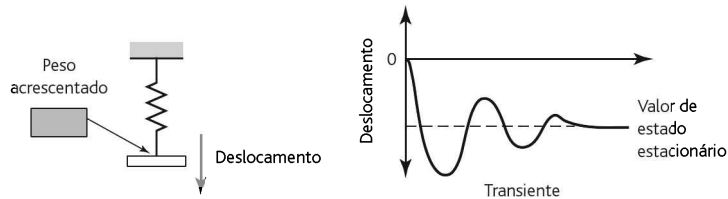


Figura 12.2 Resposta transiente e de estado estacionário de um sistema mola.

penso por ela. A deflexão da mola aumenta abruptamente e ela pode passa a oscilar por um algum tempo, após o que se estabiliza em um valor estacionário. Este valor é a resposta de estado estacionário do sistema mola; a oscilação que ocorre antes é a resposta transiente.

12.2.3 Formas de entradas

A entrada do sistema mola anterior, o peso, é uma quantidade que varia com o tempo. Até um determinado instante de tempo não há o acréscimo do peso, ou seja, não há entrada, então após este instante há uma entrada que permanece pelo resto do tempo. Este tipo de entrada é conhecida como **entrada em degrau** e é a forma mostrada na Figura 12.3(a).

O sinal de entrada para os sistemas pode tomar outras formas, como, por exemplo, o impulso, a rampa e os sinais senoidais. Um **impulso** é uma entrada de duração muito curta (Figura 12.3(b)); uma **rampa** é uma entrada que aumenta de forma constante (Figura 12.3(c)) e pode ser descrita por uma equação na forma $y = kt$, em que k é uma constante; uma entrada **senoidal** pode ser descrita por uma equação da forma $y = k \text{ sen } \omega t$, sendo ω denominado frequência angular e igual a $2\pi f$, em que f é a frequência.

Tanto a entrada quanto a saída são funções do tempo. Uma forma de indicar isto é escrevendo-as na forma $f(t)$, em que f é a função e (t) indica que o valor depende do tempo t . Portanto, para o peso W de entrada para o sistema mola poderíamos escrever $W(t)$ e para a deflexão de saída d escreveríamos $d(t)$. Normalmente utiliza-se $y(t)$ para uma entrada e $x(t)$ para uma saída.

12.3 Sistemas de primeira ordem

Considere um sistema de primeira ordem com $y(t)$ como entrada e $x(t)$ como saída, o qual tem uma entrada forçada $b_0 y$ e pode ser descrito por uma equação diferencial da forma

$$a_1 \frac{dx}{dt} + a_0 x = b_0 y$$

em que a_1 , a_0 e b_0 são constantes.

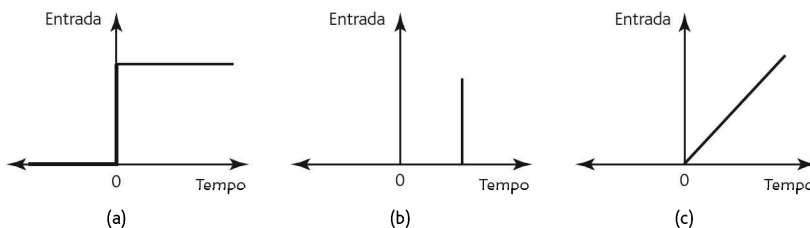


Figura 12.3 Entradas: (a) degrau no tempo, (b) impulso em algum instante, (c) rampa começando no instante zero.

12.3.1 Resposta natural

A entrada $y(t)$ pode tomar muitas formas. Considere primeiro a situação em que a entrada é zero. Como não há entrada no sistema, não temos sinal forçando o sistema a responder de uma forma diferente da natural, na qual não há entrada. A equação diferencial é então

$$a_1 \frac{dx}{dt} + a_0 x = 0$$

Podemos resolver esta equação usando a técnica denominada **separação de variáveis**. A equação pode ser escrita com todas as variáveis x de um lado e todas as variáveis t do outro:

$$\frac{dx}{x} = -\frac{a_0}{a_1} dt$$

Integrando ela entre os valores iniciais de $x = 1$ em $t = 0$, ou seja, uma entrada em degrau unitário, e x em t , obtemos

$$\ln x = -\frac{a_0}{a_1} t$$

e portanto temos

$$x = e^{-a_0 t/a_1}$$

Entretanto, poderíamos ter reconhecido que a equação diferencial seria uma solução da forma $x = Ae^{st}$, em que A e s são constantes. Temos em seguida $dx/dt = sAe^{st}$, e assim, quando estes valores são substituídos na equação diferencial, obtemos

$$a_1 s A e^{st} + a_0 A e^{st} = 0$$

e assim $a_1 s + a_0 = 0$ e $s = -a_0/a_1$. Portanto, a solução é

$$x = Ae^{-a_0 t/a_1}$$

Esta é denominada resposta natural, visto que não há nenhuma função forçando uma resposta. Podemos determinar o valor da constante A fornecendo alguma condição inicial (condição de contorno). Portanto, se $x = 1$ quando $t = 0$, então $A = 1$. A Figura 12.4 mostra a resposta natural, ou seja, um decaimento exponencial:

$$x = e^{-a_0 t/a_1}$$

12.3.2 Resposta com uma entrada forçada

Agora considere a equação diferencial quando há uma **função forçada**, ou seja,

$$a_1 \frac{dx}{dt} + a_0 x = b_0 y$$

Considere a solução para esta equação como tendo duas partes, ou seja, $x = u + v$. Uma parte representa a parte transiente da solução e a outra, a parte de estado estacionário. Substituindo isto na equação diferencial, temos

$$a_1 \frac{d(u + v)}{dt} + a_0(u + v) = b_0 y$$

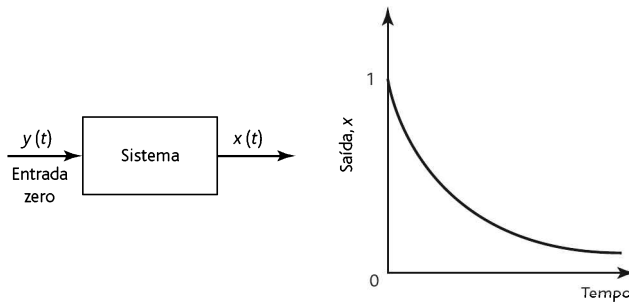


Figura 12.4 Resposta natural de um sistema de primeira ordem.

Rearranjando, temos

$$\left(a_1 \frac{du}{dt} + a_0 u \right) + \left(a_1 \frac{dv}{dt} + a_0 v \right) = b_0 y$$

Se fizermos

$$a_1 \frac{dv}{dt} + a_0 v = b_0 y$$

então teremos

$$a_1 \frac{du}{dt} + a_0 u = 0$$

e teremos duas equações diferenciais, uma contendo a função forçada e a outra, apenas a equação da resposta natural. Esta última equação é exatamente a equação natural que resolvemos anteriormente nesta seção e portanto tem uma solução da forma

$$u = Ae^{-a_0 t/a_1}$$

A outra equação diferencial contém a função forçada y . Para esta equação diferencial, a forma da solução depende da forma do sinal de entrada y . Para uma entrada em degrau quando y é constante para todos os instantes maiores do que 0, ou seja, $y = k$, podemos testar a solução $v = A$, em que A é uma constante. Se tivermos um sinal de entrada da forma $y = a + bt + ct^2 + \dots$, em que a , b e c são constantes que podem ser nulas, então podemos testar a solução da forma $v = A + Bt + Ct^2 + \dots$. Para um sinal senoidal, podemos testar a solução da forma $v = A \cos \omega t + B \sin \omega t$.

Para ilustrar isto, considere uma entrada em degrau no instante $t = 0$, sendo k a amplitude do degrau (Figura 12.5(a)). Então, teste a solução da forma $v = A$. A diferenciação de uma constante é zero; portanto, quando esta solução é substituída na equação diferencial, obtemos $a_0 A = b_0 k$ e, assim, $v = (b_0/a_0)k$.

A solução completa é dada por $y = u + v$ e, desta forma, temos

$$y = Ae^{-a_0 t/a_1} + \frac{b_0}{a_0} k$$

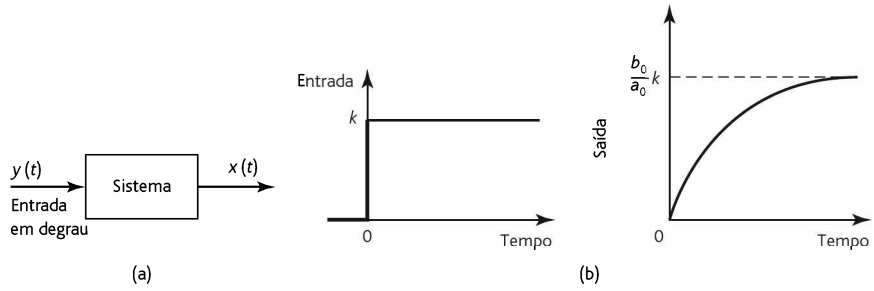


Figura 12.5 (a) Entrada em degrau, (b) saída resultante.

Podemos determinar o valor da constante A dando alguma condição inicial (condição de contorno). Portanto, se a saída $y = 0$ quando $t = 0$, então

$$0 = A + \frac{b_0}{a_0} k$$

Portanto, $A = -(b_0/a_0)k$. A solução então se torna

$$x = \frac{b_0}{a_0} k(1 - e^{-a_0 t/a_1})$$

Quando $t \rightarrow \infty$, o termo exponencial tende a zero. O termo exponencial nos fornece então parte da resposta que é a solução transiente. A resposta em estado estacionário é o valor de x quando $t \rightarrow \infty$ e, assim, é $(b_0/a_0)k$. Portanto, a equação pode ser escrita como

$$x = \text{valor de estado estacionário} \times (1 - e^{-a_0 t/a_1})$$

A Figura 12.5(b) mostra um gráfico de como a saída x varia com o tempo para uma entrada em degrau.

12.3.3 Exemplos de sistemas de primeira ordem

Exemplificando mais uma vez o que foi abordado anteriormente, considere os seguintes exemplos de sistemas de primeira ordem.

Um sistema de um transdutor elétrico consiste de uma resistência em série com um capacitor, e quando submetido a uma entrada em degrau de amplitude V fornece uma saída que é a diferença de potencial no capacitor, v , dada pela equação diferencial

$$RC \frac{dv}{dt} + v = V$$

Comparando a equação diferencial com a equação resolvida anteriormente: $a_1 = RC$, $a_0 = 1$ e $b_0 = 1$. Portanto, a solução é da forma

$$v = V(1 - e^{-t/RC})$$

Agora considere um circuito elétrico que consiste de uma resistência de $1 \text{ M}\Omega$ em série com uma capacitância de $2 \text{ }\mu\text{F}$. No instante $t = 0$ o circuito é submetido a uma tensão em rampa de $4t \text{ V}$, ou seja, a tensão aumenta a uma taxa de 4 V a cada 1

s. A equação diferencial tem forma similar à dada no exemplo anterior, porém com a tensão em degrau V sendo substituída pela tensão em rampa $4t$, ou seja,

$$RC \frac{dv}{dt} + v = 4t$$

Portanto, usando os valores dados,

$$2 \frac{dv}{dt} + v = 4t$$

Fazendo $v = v_n + v_f$, ou seja, a soma das respostas natural e forçada, temos para a resposta natural

$$2 \frac{dv_n}{dt} + v_n = 0$$

e para a resposta forçada

$$2 \frac{dv_f}{dt} + v_f = 4t$$

Para a equação diferencial da resposta natural podemos testar a solução da forma $v_n = Ae^{st}$. Portanto, usando este valor,

$$2Ase^{st} + Ae^{st} = 0$$

Assim, $s = -\frac{1}{2}$ e, portanto, $v_n = Ae^{-t/2}$. Para a equação diferencial da resposta forçada, visto que o membro direito da equação é $4t$, podemos testar a solução da forma $v_f = A + Bt$. Usando este valor, obtemos $2B + A + Bt = 4t$. Portanto, temos que $B = 4$ e $A = -2B = -8$. Portanto, a solução é $v_f = -8 + 4t$. Assim, a solução completa é

$$v = v_n + v_f = Ae^{-t/2} - 8 + 4t$$

Visto que $v = 0$ quando $t = 0$, temos $A = 8$. Portanto,

$$v = 8e^{-t/2} - 8 + 4t$$

Como mais um exemplo, considere um motor quando a relação entre a velocidade angular de saída ω e a tensão de entrada v para o motor é dada por

$$\frac{IR}{k_1 k_2} \frac{d\omega}{dt} + \omega = \frac{1}{k_1} v$$

Comparando a equação diferencial com a equação resolvida antes, então $a_1 = IR/k_1 k_2$, $a_0 = 1$ e $b_0 = 1/k_1$. O valor em estado estacionário para uma entrada em degrau de amplitude 1 V é, portanto, $(b_0/a_0) = 1/k_1$

12.3.4 Constante de tempo

Para um sistema de primeira ordem submetido a uma entrada em degrau de amplitude k , temos uma saída y que varia com o tempo t de acordo com

$$x = \frac{b_0}{a_0} k(1 - e^{-a_0 t/a_1})$$

ou

$$x = \text{valor de estado estacionário} \times (1 - e^{-a_0 t/a_1})$$

Quando o tempo $t = (a_1/a_0)$, então o termo exponencial tem o valor $e^{-1} = 0,37$ e

$$x = \text{valor de estado estacionário} \times (1 - 0,37)$$

Neste instante, a saída aumentou para 0,63 do seu valor de estado estacionário. Este tempo é denominado **constante de tempo** τ :

$$\tau = \frac{a_1}{a_0}$$

Em um tempo de $2(a_1/a_0) = 2\tau$, o termo exponencial torna-se $e^{-2} = 0,14$, e então

$$x = \text{valor de estado estacionário} \times (1 - 0,14)$$

Neste instante, a saída aumentou para 0,86 do seu valor de estado estacionário. De forma similar, podemos calcular os valores para a saída após $3\tau, 4\tau, 5\tau$, etc. A Tabela 12.1 mostra os resultados destes cálculos, e a Figura 12.6 mostra o gráfico de como a saída varia com o tempo para uma entrada em degrau.

Em termos de constante de tempo τ , podemos escrever a equação que descreve a resposta de um sistema de primeira ordem como

$$x = \text{valor de estado estacionário} \times (1 - e^{-t/\tau})$$

A constante de tempo τ é (a_1/a_0) , portanto podemos escrever a nossa forma geral da equação diferencial de primeira ordem como

$$a_1 \frac{dx}{dt} + a_0 x = b_0 y$$

enquanto

$$\tau \frac{dx}{dt} + x = \frac{b_0}{a_0} y$$

Mas b_0/a_0 é o fator pelo qual a entrada y é multiplicada para obtermos o valor de estado estacionário. Podemos denominá-lo **ganho de estado estacionário**, visto que ele é o fator que determina quantas vezes a saída é maior que a entrada em con-

Tabela 12.1 Resposta de um sistema de primeira ordem a uma entrada em degrau

Tempo t	Fração da saída em estado estacionário
0	0
1τ	0,63
2τ	0,86
3τ	0,95
4τ	0,98
5τ	0,99
∞	1

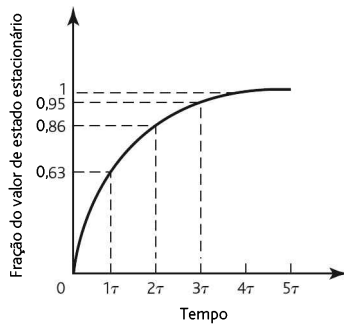


Figura 12.6 Resposta de um sistema de primeira ordem a uma entrada em degrau.

dições de estado estacionário. Portanto, se indicarmos este ganho por G_{SS} , então a equação diferencial pode ser escrita na forma

$$\tau \frac{dx}{dt} + x = G_{SS}y$$

Para ilustrar isto, considere a Figura 12.7, que mostra como a saída v_o de um sistema de primeira ordem varia com o tempo quando submetido a uma entrada em degrau de 5 V. A constante de tempo é o tempo gasto para que a saída de um sistema de primeira ordem varie de 0 a 0,63 do seu valor final de estado estacionário. Neste caso, este tempo é cerca de 3 s. Podemos verificar este valor, e que o sistema é de primeira ordem, determinando o valor para 2, ou seja, 6 s. Em um sistema de primeira ordem, ele deveria ser 0,86 do valor de estado estacionário. Neste caso, este é o valor encontrado. A saída de estado estacionário é 10 V. Portanto, o ganho de estado estacionário, G_{SS} , é (saída de estado estacionário/entrada) = 10/5 = 2. A equação diferencial para um sistema de primeira ordem pode ser escrita como

$$\tau \frac{dx}{dt} + x = G_{SS}y$$

Portanto, para este sistema, temos

$$3 \frac{dv_o}{dt} + v_o = 2v_i$$

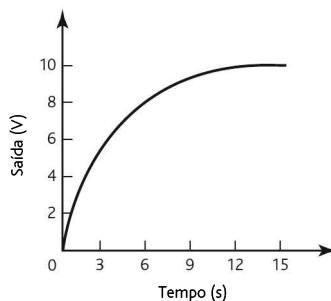


Figura 12.7 Exemplo.

12.4 Sistemas de segunda ordem

Muitos sistemas de segunda ordem podem ser considerados análogos a uma mola tensionada com uma massa e algum mecanismo que provê amortecimento. A Figura 12.8 mostra a base deste sistema.

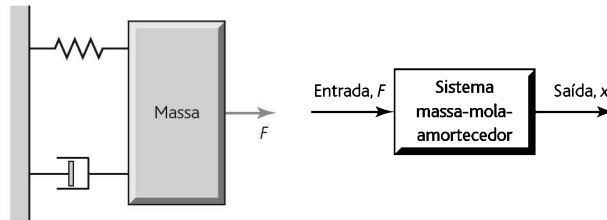


Figura 12.8 Sistema massa-mola-amortecedor.

Este sistema foi analisado na Seção 10.2.2. A equação que descreve a relação entre a força de entrada F e o deslocamento de saída x é

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = F$$

em que m é a massa, c é a constante de amortecimento e k é a constante da mola.

A forma na qual o deslocamento x varia com o tempo depende da quantidade de amortecimento no sistema. Portanto, se a força for aplicada como uma entrada em degrau e não houver amortecimento algum, então a massa oscila livremente sobre a mola e as oscilações continuam indefinidamente. Nenhum amortecimento significa $c = 0$ e, assim, o termo dx/dt é zero. Entretanto, o amortecimento faz com que as oscilações terminem antes que o deslocamento final da massa seja obtido. Se o amortecimento for suficientemente alto, não haverá oscilações e o deslocamento da massa aumentará vagarosamente com o tempo, e a massa se moverá gradualmente em direção à posição final (estado estacionário). A Figura 12.9 mostra a forma geral em que o deslocamento, para uma entrada em degrau, varia com o tempo para diferentes graus de amortecimento.

12.4.1 Resposta natural

Considere uma massa na extremidade de uma mola. Na ausência de qualquer amortecimento e deixando-a oscilar livremente sem forçar, a saída do sistema de segunda ordem é uma oscilação contínua (movimento harmônico simples). Portanto, considere que a oscilação seja descrita pela equação

$$x = A \text{ sen } \omega_n t$$

em que x é o deslocamento no instante t , A é a amplitude da oscilação e ω_n é a frequência angular das oscilações sem amortecimento. Derivando esta equação, temos

$$\frac{dx}{dt} = \omega_n A \cos \omega_n t$$

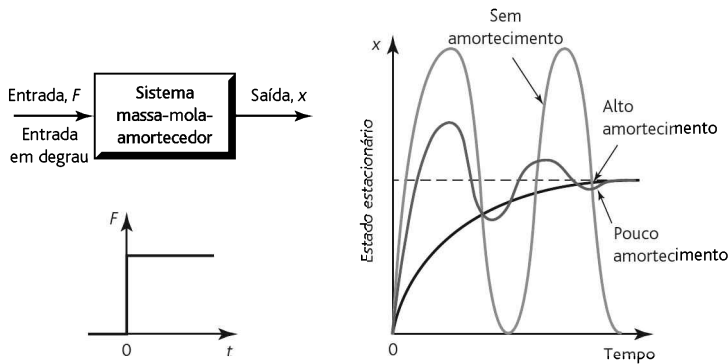


Figura 12.9 Efeito do amortecimento em um sistema de segunda ordem.

Derivando pela segunda vez, temos

$$\frac{d^2x}{dt^2} = -\omega_n^2 A \text{sen } \omega_n t = -\omega_n^2 x$$

Esta equação pode ser reorganizada para obtermos a equação diferencial

$$\frac{d^2x}{dt^2} + \omega_n^2 x = 0$$

Porém, para uma massa m em uma mola de rigidez k temos uma força de restabelecimento de kx e, portanto,

$$m \frac{d^2x}{dt^2} = -kx$$

Isto pode ser escrito como

$$\frac{d^2x}{dt^2} + \frac{k}{m} x = 0$$

Portanto, comparando as duas equações diferenciais, temos

$$\omega_n^2 = \frac{k}{m}$$

e $x = A \text{sen } \omega_n t$ é a solução da equação diferencial.

Agora considere o caso em que temos amortecimento. O movimento da massa é então descrito por

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0$$

Para resolver esta equação, podemos testar a solução da forma $x_n = Ae^{st}$. Isto nos dá $dx_n/dt = Ase^{st}$ e $d^2x_n/dt^2 = As^2e^{st}$. Portanto, substituindo estes valores na equação diferencial, temos

$$mAs^2e^{st} + cAse^{st} + kAe^{st} = 0$$

$$ms^2 + cs + k = 0$$

Portanto, $x_n = Ae^{st}$ apenas pode ser uma solução contanto que a equação anterior seja igual a zero. Esta equação é denominada **equação auxiliar**. As raízes desta equação podem ser obtidas fatorando-se ou usando-se a fórmula para as raízes de uma equação quadrática. Portanto,

$$s = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m} = -\frac{c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \frac{k}{m}}$$

$$= -\frac{c}{2m} \pm \sqrt{\frac{k}{m}\left(\frac{c^2}{4mk}\right) - \frac{k}{m}}$$

Mas $\omega_n^2 = k/m$, e assim, se fizermos $\zeta^2 = c^2/4mk$, podemos escrever a equação anterior como

$$s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

ζ é denominado **fator de amortecimento**.

O valor de s obtido da equação anterior depende muito do valor do termo quadrático. Portanto, quando ζ^2 for maior do que 1, o termo da raiz quadrada será a raiz quadrada de um número positivo, e quando ζ^2 for menor do que 1, temos a raiz quadrada de um número negativo. O fator de amortecimento determina se o termo raiz quadrada é um número positivo ou negativo e, assim, a forma da saída do sistema:

1 Sobreamortecido

Com $\zeta > 1$ existem duas raízes reais diferentes, s_1 e s_2 :

$$s_1 = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1}$$

$$s_2 = -\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1}$$

assim, a solução geral para x_n é

$$x_n = Ae^{s_1t} + Be^{s_2t}$$

Para estas condições diz-se que o sistema é **sobreamortecido**.

2 Criticamente amortecido

Quando $\zeta = 1$ existem duas raízes iguais, $s_1 = s_2 = -\omega_n$. Para esta condição, que é denominada **criticamente amortecida**,

$$x_n = (At + B)e^{-\omega_n t}$$

Pode parecer que a solução para este caso seria $x_n = Ae^{st}$, mas é necessário duas constantes e, assim, a solução é desta forma.

3 Subamortecido

Com $\zeta < 1$ existem duas raízes complexas, pois as raízes envolvem a raiz quadrada de (-1):

$$s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} = -\zeta\omega_n \pm \omega_n\sqrt{-1}\sqrt{1 - \zeta^2}$$

assim, escrevendo j para $\sqrt{-1}$,

$$s = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$$

Se fizermos

$$\omega = \omega_n \sqrt{1 - \zeta^2}$$

então podemos escrever $s = -\zeta\omega_d \pm j\omega$, e assim as duas raízes são

$$s_1 = -\zeta\omega_d + j\omega \text{ e } s_2 = -\zeta\omega_d - j\omega$$

O termo ω é a frequência angular do movimento quando ele está na condição amortecida especificada por ζ . A solução sob estas condições é portanto

$$x_n = Ae^{(-\zeta\omega_n + j\omega)t} + Be^{(-\zeta\omega_n - j\omega)t} = e^{-\zeta\omega_n t}(Ae^{j\omega t} + Be^{-j\omega t})$$

Mas $e^{j\omega t} = \cos \omega t + j \sin \omega t$ e $e^{-j\omega t} = \cos \omega t - j \sin \omega t$. Portanto,

$$\begin{aligned} x_n &= e^{-\zeta\omega_n t}(A \cos \omega t + jA \sin \omega t + B \cos \omega t - jB \sin \omega t) \\ &= e^{-\zeta\omega_n t}[(A + B) \cos \omega t + j(A - B) \sin \omega t] \end{aligned}$$

Se substituirmos as constantes P e Q para $(A + B)$ e $j(A - B)$, então

$$x_n = e^{-\zeta\omega_n t}(P \cos \omega t + Q \sin \omega t)$$

Para estas condições, diz-se que o sistema é **subamortecido**.

12.4.2 Resposta com uma entrada forçada

Quando temos uma entrada forçada F , a equação diferencial torna-se

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = F$$

Podemos resolver essa equação diferencial de segunda ordem pelo mesmo método usado anteriormente para a de primeira ordem e considerar a solução constituída de duas partes, a resposta transiente (natural) e a forçada, ou seja, $x = x_n + x_f$. Substituindo x na equação anterior, temos

$$m \frac{d^2(x_n + x_f)}{dt^2} + c \frac{d(x_n + x_f)}{dt} + k(x_n + x_f) = F$$

Se fizermos

$$m \frac{d^2x_n}{dt^2} + c \frac{dx_n}{dt} + kx_n = 0$$

então devemos ter

$$m \frac{d^2x_f}{dt^2} + c \frac{dx_f}{dt} + kx_f = F$$

A seção anterior apresentou as soluções para a parte natural da solução. Para resolver a equação forçada,

$$m \frac{d^2x_f}{dt^2} + c \frac{dx_f}{dt} + kx_f = F$$

precisamos considerar uma forma particular de sinal de entrada e então testar a solução. Portanto, para uma entrada em degrau de amplitude F no instante $t = 0$, podemos testar a solução $x_f = A$, em que A é uma constante (veja a Seção 12.3.2 sobre equações diferenciais de primeira ordem para uma discussão da escolha de soluções). Portanto, $dx_f/dt = 0$ e $d^2x_f/dt^2 = 0$. Portanto, quando estas substituições são feitas na equação diferencial, $0 + 0 + kA = F$ e, portanto, $A = F/k$ e $x_f = F/k$. A solução completa, a soma das soluções natural e forçada, é portanto, para o sistema sobre-amortecido,

$$x = Ae^{s_1t} + Be^{s_2t} + \frac{F}{k}$$

para o sistema criticamente amortecido

$$x = (At + B)e^{-\omega_n t} + \frac{F}{k}$$

e para o sistema subamortecido

$$x = e^{-\zeta\omega_n t}(P \cos \omega t + Q \sin \omega t) + \frac{F}{k}$$

Quando $t \rightarrow \infty$ as três equações anteriores levam à solução $x = F/k$. Esta é a **condição de estado estacionário**.

Portanto, uma equação diferencial de segunda ordem da forma

$$a_2 \frac{d^2x}{dt^2} + a_1 \frac{dx}{dt} + a_0x = b_0y$$

tem uma frequência natural dada por

$$\omega_n^2 = \frac{a_0}{a_2}$$

e um fator de amortecimento dado por

$$\zeta^2 = \frac{a_1^2}{4a_2a_0}$$

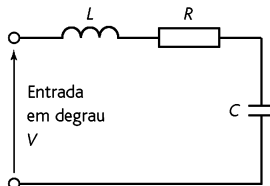


Figura 12.10 Sistema RLC.

12.4.3 Exemplos de sistemas de segunda ordem

Os exemplos a seguir ilustram o que foi ressaltado anteriormente.

Considere um circuito RLC em série (Figura 12.10) com $R = 100 \Omega$, $L = 2,0 \text{ H}$ e $C = 20 \mu\text{F}$. Quando há uma entrada em degrau V , a corrente i no circuito é dada por (veja o texto associado com a Figura 10.8)

$$\frac{d^2i}{dt^2} + \frac{R}{L} \frac{di}{dt} + \frac{1}{LC} i = \frac{V}{LC}$$

Se compararmos a equação com a equação diferencial de segunda ordem geral de

$$a_2 \frac{d^2x}{dt^2} + a_1 \frac{dx}{dt} + a_0x = b_0y$$

então a frequência angular natural é dada por

$$\omega_n^2 = \frac{1}{LC} = \frac{1}{2,0 \times 20 \times 10^{-6}}$$

e assim $\omega_n = 158$ Hz. A comparação com a equação de segunda ordem geral resulta em

$$\zeta^2 = \frac{(R/L)^2}{4 \times (1/LC)} = \frac{R^2C}{4L} = \frac{100^2 \times 20 \times 10^{-6}}{4 \times 2,0}$$

Portanto, $\zeta = 0,16$. Como ζ é menor do que 1, o sistema é subamortecido. A frequência de oscilação subamortecida ω é

$$\omega = \omega_n \sqrt{1 - \zeta^2} = 158 \sqrt{1 - 0,16^2} = 156 \text{ Hz}$$

Como o sistema é subamortecido, a solução terá a mesma forma que

$$x = e^{-\zeta\omega_n t} (P \cos \omega t + Q \sen \omega t) + \frac{F}{k}$$

e assim

$$i = e^{-0,16 \times 158 t} (P \cos 156 t + Q \sen 156 t) + V$$

Como $i = 0$ quando $t = 0$, então $0 = 1(P + 0) + V$. Portanto, $P = -V$. Como $di/dt = 0$ quando $t = 0$, então, derivando a equação anterior e igualando-a a zero, obtemos

$$\frac{di}{dt} = e^{-\zeta\omega_n t} (\omega P \sen \omega t - \omega Q \cos \omega t) - \zeta \omega_n e^{-\zeta\omega_n t} (P \cos \omega t + Q \cos \omega t)$$

Portanto, $0 = 1(0 - \omega Q) - \zeta \omega_n (P + 0)$, e assim

$$Q = \frac{\zeta \omega_n P}{\omega} = -\frac{\zeta \omega_n V}{\omega} = -\frac{0,16 \times 158 V}{156} \approx -0,16 V$$

Portanto, a solução da equação diferencial é

$$i = V - V e^{-25,3 t} (\cos 156 t + 0,16 \sen 156 t)$$

Agora considere o sistema mostrado na Figura 12.11. A entrada, um torque T , é aplicada a um disco com um momento de inércia I em torno do eixo. O eixo é livre para girar na extremidade do disco, porém é fixo na outra extremidade. A rotação do eixo tem como oposição a sua rigidez

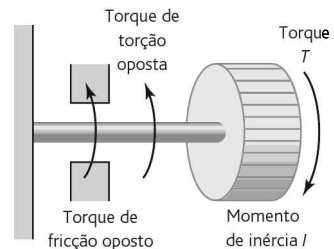


Figura 12.11 Sistema de torção.

a torção, com um torque oposto de $k\theta_0$ que acontece para uma rotação de entrada θ_0 . k é uma constante. As forças de fricção amortecem a rotação do eixo e proporcionam um torque oposto de $c \, d\theta_0/dt$, em que c é uma constante. Suponha que precisemos determinar a condição deste sistema para que seja criticamente amortecido.

Precisamos primeiro obter a equação diferencial do sistema. O torque resultante é

$$\text{torque resultante} = T - c \frac{d\theta_0}{dt} - k\theta_0$$

O torque resultante é $I \, d^2\theta_0/dt^2$, portanto

$$I \frac{d^2\theta_0}{dt^2} = T - c \frac{d\theta_0}{dt} - k\theta_0$$

$$I \frac{d^2\theta_0}{dt^2} + c \frac{d\theta_0}{dt} + k\theta_0 = T$$

A condição para o amortecimento crítico é dada quando a razão de amortecimento ζ for igual a 1. Comparando a equação diferencial anterior com a forma geral da equação diferencial de segunda ordem, temos

$$\zeta^2 = \frac{a_1^2}{4a_2a_0} = \frac{c^2}{4Ik}$$

Portanto, para um amortecimento crítico temos que ter $c = \sqrt{4Ik}$.

12.5 Medidas de desempenho para sistemas de segunda ordem

A Figura 12.12 mostra a forma típica da resposta de um sistema de segunda ordem subamortecido para uma entrada em degrau. Certos termos são usados para especificar este desempenho.

O **tempo de subida** (t_r) é o tempo gasto para a resposta x aumentar de 0 ao valor de estado estacionário x_{ss} e é uma medida da rapidez com que o sistema responde à entrada. Este é o tempo que a oscilação da resposta gasta para completar um quarto do ciclo, ou seja, $\frac{1}{2}\pi$. Portanto,

$$\omega t_r = \frac{1}{2}\pi$$

O tempo de subida é especificado algumas vezes como o tempo gasto para que a resposta aumente de 10 a 90%, por exemplo, do valor de estado estacionário.

O **tempo de pico** (t_p) é o tempo gasto para a resposta aumentar de 0 ao primeiro valor de pico. Este é o tempo que a oscilação da resposta gasta para completar um semiciclo, ou seja, π . Portanto,

$$\omega t_p = \pi$$

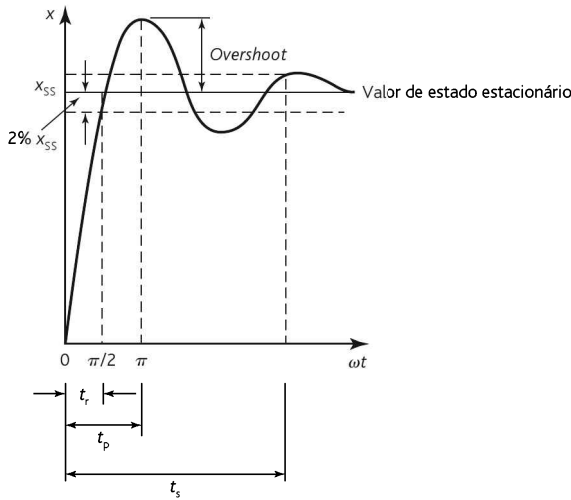


Figura 12.12 Resposta ao degrau de um sistema subamortecido.

O **overshoot** (sobressinal) é o valor máximo em que a resposta supera o valor de estado estacionário. Este valor corresponde à amplitude do primeiro pico. O *overshoot* é expresso muitas vezes como uma percentagem do valor de estado estacionário. Para as oscilações subamortecidas de um sistema podemos escrever

$$x = e^{-\zeta\omega_n t} (P \cos \omega t + Q \sen \omega t) + \text{valor de estado estacionário}$$

Visto que $x = 0$ quando $t = 0$, então $0 = 1(P + 0) + x_{SS}$ e, assim, $P = -x_{SS}$. O *overshoot* ocorre em $\omega t = \pi$, e portanto

$$x = e^{-\zeta\omega_n\pi/\omega}(P + 0) + x_{SS}$$

O *overshoot* é a diferença entre a saída neste instante e o valor de estado estacionário. Portanto,

$$\text{overshoot} = x_{SS} e^{-\zeta\omega_n\pi/\omega}$$

Visto que $\omega = \omega_n\sqrt{1 - \zeta^2}$, podemos então escrever

$$\text{overshoot} = x_{SS} \exp\left(\frac{-\zeta\omega_n\pi}{\omega_n\sqrt{1 - \zeta^2}}\right) = x_{SS} \exp\left(\frac{-\zeta\pi}{\sqrt{1 - \zeta^2}}\right)$$

Expresso como uma percentagem de x_{SS} ,

$$\text{overshoot em percentagem} = \exp\left(\frac{-\zeta\pi}{\sqrt{1 - \zeta^2}}\right) \times 100\%$$

A Tabela 12.2 mostra os valores de *overshoot* em percentagem para razões de amortecimento específicas.

Uma indicação de como oscilações rápidas diminuem é fornecida pela **razão de amortecimento** ou **decremento**. Este valor corresponde à amplitude do se-

Tabela 12.2 Percentagem de pico do *overshoot*

Razão de amortecimento	<i>Overshoot</i> em percentagem
0,2	52,7
0,4	25,4
0,6	9,5
0,8	1,5

gundo *overshoot* dividida pela do primeiro. O primeiro *overshoot* ocorre quando temos $\omega t = \pi$, o segundo ocorre quando $\omega t = 3\pi$. Portanto,

$$\text{primeiro } overshoot = x_{SS} \exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

$$\text{segundo } overshoot = x_{SS} \exp\left(\frac{-3\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

e assim

$$\text{razão de amortecimento} = \frac{\text{segundo } overshoot}{\text{primeiro } overshoot} = \exp\left(\frac{-2\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

O **tempo de acomodação** (t_s) é usado como uma medida do tempo gasto para a oscilação terminar. Este é o tempo em que a resposta cai e permanece em uma percentagem específica, por exemplo, 2%, do valor de estado estacionário (veja a Figura 12.12). Isto significa que a amplitude das oscilações deve ser menor do que 2% de x_{SS} . Temos então

$$x = e^{-\zeta\omega_n t} (P \cos \omega t + Q \sin \omega t) + \text{valor de estado estacionário}$$

e, conforme deduzido antes, $P = -x_{SS}$. A amplitude da oscilação é $(x - x_{SS})$, quando x é um valor máximo. Os valores máximos ocorrem quando ωt é um múltiplo de π e, portanto, temos $\cos \omega t = 1$ e $\sin \omega t = 0$. Para um tempo de acomodação de 2%, a amplitude máxima da resposta é 2% de x_{SS} , ou seja, $0,02x_{SS}$. Portanto,

$$0,02x_{SS} = e^{-\zeta\omega_n t_s} (x_{SS} \times 1 + 0)$$

Aplicando o logaritmo, temos $\text{Ln } 0,02 = -\zeta\omega_n t_s$, e como $\text{Ln } 0,02 = -3,9$ ou aproximadamente -4 , então

$$t_s = \frac{4}{\zeta\omega_n}$$

Este é o valor do tempo de acomodação se a percentagem especificada for 2%. Se a percentagem for 5%, a equação torna-se

$$t_s = \frac{3}{\zeta\omega_n}$$

Visto que o tempo de um ciclo, ou seja, o período, é $1/f$, em que f é a frequência, e como $\omega = 2\pi f$, então o tempo de um ciclo é $2\pi/\omega$. Em um tempo de ajuste t_s , o número de oscilações que ocorrem é

$$\text{número de oscilações} = \frac{\text{tempo de ajuste}}{\text{período}}$$

e, portanto, para um tempo de ajuste definido para 2% do valor de estado estacionário,

$$\text{número de oscilações} = \frac{4/\zeta\omega_n}{2\pi/\omega}$$

Como $\omega = \omega_n\sqrt{1-\zeta^2}$, então

$$\text{número de oscilações} = \frac{2\omega_n\sqrt{1-\zeta^2}}{\pi\zeta\omega_n} = \frac{2}{\pi}\sqrt{\frac{1}{\zeta^2}-1}$$

Para ilustrar isto, considere um sistema de segunda ordem que tem uma frequência angular de 2,0 Hz e uma frequência amortecida de 1,8 Hz. Como $\omega = \omega_n\sqrt{1-\zeta^2}$, então o fator de amortecimento é dado por

$$1,8 = 2,0\sqrt{1-\zeta^2}$$

e $\zeta = 0,44$. Como $\omega t_r = \frac{1}{2}\pi$, então 100% do tempo de subida é dado por

$$t_r = \frac{\pi}{2 \times 1,8} = 0,87 \text{ s}$$

O *overshoot* em percentagem é dado por

$$\begin{aligned} \text{overshoot em percentagem} &= \exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right) \times 100\% \\ &= \exp\left(\frac{-0,44\pi}{\sqrt{1-0,44^2}}\right) \times 100\% \end{aligned}$$

O *overshoot* em percentagem é, portanto, 21%. O tempo de acomodação de 2% é dado por

$$t_s = \frac{4}{\zeta\omega_n} = \frac{4}{0,44 \times 2,0} = 4,5 \text{ s}$$

O número de oscilações que ocorrem dentro do tempo de acomodação de 2% é dado por

$$\text{número de oscilações} = \frac{2}{\pi}\sqrt{\frac{1}{\zeta^2}-1} = \frac{2}{\pi}\sqrt{\frac{1}{0,44^2}-1} = 1,3$$

12.6 Identificação de sistemas

Nos Capítulos 10 e 11, os modelos foram idealizados para sistemas considerando-os constituídos de elementos simples. Uma forma alternativa de desenvolver um modelo para um sistema real é fazer testes para determinar sua resposta a algumas entradas, como, por exemplo, uma entrada em degrau, e determinar o modelo que se ajusta à resposta. Este processo de determinação de um modelo matemático é conhecido como **identificação de sistemas**. Portanto, se obtermos uma resposta a uma entrada em degrau da forma mostrada na Figura 12.5, podemos considerar que ele é um sistema de primeira ordem e determinar a constante de tempo a partir da curva de resposta. Por exemplo, suponha que a resposta leva um tempo de 1,5 s para atingir 0,63 de sua amplitude final, que é cinco vezes a amplitude da entrada em degrau. A Tabela 12.1 indica uma constante de tempo de 1,5 s, e a equação diferencial que descreve o modelo é

$$1,5 \frac{dx}{dt} + x = 5y$$

Um sistema de segunda ordem subamortecido apresenta uma resposta a uma entrada em degrau da forma mostrada na Figura 12.12. A razão de amortecimento pode ser determinada a partir das medidas do primeiro e segundo *overshoots* com a razão destes *overshoots*, ou seja, a razão de amortecimento, fornecendo a razão de amortecimento. A frequência natural pode ser determinada a partir do tempo entre sucessivos *overshoots*. Então podemos usar estes valores para determinar as constantes na equação diferencial de segunda ordem.

Resumo

A **resposta natural** de um sistema é quando não há entrada para forçar a variável do sistema a mudar, mas ela varia naturalmente. A **resposta forçada** de um sistema é quando há uma entrada no sistema forçando-o a mudar.

Um sistema de primeira ordem sem uma entrada forçada tem uma equação diferencial da forma

$$a_1 \frac{dx}{dt} + a_0x = 0$$

e tem a solução $x = e^{-a_0t/a_1}$.

Quando há uma **função forçada**, a equação diferencial é da forma

$$a_1 \frac{dx}{dt} + a_0x = b_0y$$

e a solução é $x = \text{valor de estado estacionário} \times (1 - e^{-a_0t/a_1})$.

A **constante de tempo** τ é o tempo que a saída gasta para aumentar para 0,63 do seu valor de estado estacionário, e é (a_1/a_0) .

Um sistema de segunda ordem sem entrada forçada tem uma equação diferencial da forma

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0$$

A frequência angular natural é dada por $\omega_n^2 = k/m$, e a constante de amortecimento, por $\zeta^2 = c^2/4mk$. O sistema é **sobreamortecido** quando temos $\zeta > 1$ e a solução geral para x_n é

$$x_n = Ae^{s_1 t} + Be^{s_2 t} \quad \text{com } s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

Quando $\zeta = 1$, o sistema é **criticamente amortecido** e

$$x_n = (At + B)e^{-\omega_n t}$$

e com $\zeta < 1$ o sistema é **subamortecido** e

$$x_n = e^{-\zeta\omega_n t} (P \cos \omega t + Q \sen \omega t)$$

Quando temos uma entrada forçada F , a equação diferencial de segunda ordem se torna

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = F$$

e para o sistema sobreamortecido

$$x = Ae^{s_1 t} + Be^{s_2 t} + \frac{F}{k}$$

para o sistema criticamente amortecido

$$x = (At + B)e^{-\omega_n t} + \frac{F}{k}$$

e para o sistema subamortecido

$$x = e^{-\zeta\omega_n t} (P \cos \omega t + Q \sen \omega t) + \frac{F}{k}$$

O **tempo de subida**, t_r , é o tempo que a resposta x gasta para aumentar de 0 para o valor de estado estacionário, x_{SS} , é uma medida da rapidez de resposta de um sistema a uma entrada e é dado por $\omega t_r = \frac{1}{2}\pi$. O **tempo de pico**, t_p , é o tempo que a resposta gasta para aumentar de 0 para o primeiro pico e é dado por $\omega t_p = \pi$. O **overshoot** é o valor máximo em que a resposta ultrapassa o valor de estado estacionário e é

$$\text{overshoot} = x_{SS} \exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

A **razão de amortecimento** ou **decremento** é a amplitude do segundo *overshoot* dividida pela do primeiro e vale

$$\text{razão de amortecimento} = \exp\left(\frac{-2\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

O **tempo de acomodação**, t_s , é o tempo que a resposta gasta para cair e permanecer dentro de uma percentagem especificada, como, por exemplo, 2%, do valor de estado estacionário, sendo dado por

$$t_s = \frac{4}{\zeta\omega_n}$$

Problemas

- 12.1 Um sistema de primeira ordem tem uma constante de tempo de 4 s e uma função de transferência de estado estacionário de 6. Qual é a forma da equação diferencial deste sistema?
- 12.2 Um termômetro de mercúrio em tubo de vidro tem uma constante de tempo de 10 s. Se ele, estando em 20°C, for repentinamente levado para uma água quente a 80°C, qual será a temperatura indicada pelo termômetro após (a) 10 s, (b) 20 s?
- 12.3 Um circuito consiste de um resistor R em série com um indutor L . Quando submetido a uma tensão em degrau de entrada V no instante $t = 0$, a equação diferencial do sistema é

$$\frac{di}{dt} + \frac{R}{L}i = \frac{V}{L}$$

Qual é (a) a solução para esta equação diferencial, (b) a constante de tempo, (c) a corrente i de estado estacionário?

- 12.4 Descreva a forma da variação de saída com o tempo para uma entrada em degrau em um sistema de segunda ordem com um fator de amortecimento de (a) 0, (b) 0,5, (c) 1,0, (d) 1,5.
- 12.5 Um circuito RLC tem uma corrente i que varia com o tempo t quando submetido a uma entrada em degrau de V e é descrito por

$$\frac{d^2i}{dt^2} + 10\frac{di}{dt} + 16i = 16V$$

Qual é (a) a frequência amortecida, (b) a razão de amortecimento, (c) a solução da equação se $i = 0$ quando $t = 0$ e $di/dt = 0$ quando $t = 0$?

- 12.6 Um sistema tem uma saída x que varia com o tempo t quando submetido a uma entrada em degrau de y e é descrito por

$$\frac{d^2x}{dt^2} + 10 \frac{dx}{dt} + 25x = 50y$$

Qual é (a) a frequência amortecida, (b) a razão de amortecimento, (c) a solução da equação se $x = 0$ quando $t = 0$ e $dx/dt = -2$ quando $t = 0$ e há uma entrada em degrau de amplitude de 3 unidades?

- 12.7 Um acelerômetro (um instrumento para medir aceleração) tem uma frequência angular amortecida de 100 Hz e um fator de amortecimento de 0,6. Qual será (a) a percentagem máxima do *overshoot* e (b) o tempo de subida quando houver uma variação repentina na aceleração?
- 12.8 Qual será (a) a frequência angular não amortecida, (b) o fator de amortecimento, (c) a frequência angular amortecida, (d) o tempo de subida, (e) o *overshoot* máximo em percentagem e (f) o tempo de acomodação de 2% para um sistema que tem a seguinte equação diferencial para uma entrada em degrau de y ?

$$\frac{d^2x}{dt^2} + 5 \frac{dx}{dt} + 16x = 16y$$

- 12.9 Quando uma tensão de 10 V é aplicada repentinamente a um voltímetro de bobina móvel, observa-se que o ponteiro sobe até 11 V antes de estabilizar em 10 V. Qual é (a) o fator de amortecimento e (b) o número de oscilações do ponteiro antes de entrar na faixa de 2% do valor de seu estado estacionário?
- 12.10 Um sistema de segunda ordem é descrito pela equação diferencial

$$\frac{d^2x}{dt^2} + c \frac{dx}{dt} + 4x = F$$

Qual valor seria necessário para a constante de amortecimento c para que a percentagem do *overshoot* fosse menor do que 9,5%?

- 12.11 A observação das oscilações de um sistema amortecido quando ele responde a uma entrada indica que o deslocamento máximo durante o segundo ciclo é 75% do primeiro. Qual é o fator de amortecimento do sistema?
- 12.12 Verifica-se que um sistema de segunda ordem tem um tempo de 1,6 s entre o primeiro *overshoot* e o segundo. Qual é a frequência natural do sistema?

13

Funções de Transferência de Sistemas

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Definir a função transferência e determiná-la a partir das equações diferenciais de sistemas de primeira e segunda ordens.
- Determinar as funções de transferência para sistemas com malhas de realimentação.
- Determinar, usando transformadas de Laplace, a resposta de sistemas de primeira e segunda ordens para entradas simples.
- Determinar o efeito da localização dos polos na resposta dos sistemas.

13.1 Função de transferência

Para um sistema amplificador, é comum falar em **ganho** do amplificador. Isto informa o quanto o sinal de saída é maior do que o de entrada. Isto permite que a saída seja determinada para entradas específicas. Portanto, por exemplo, um amplificador com um ganho de tensão de 10 fornecerá, para uma entrada de tensão de 2 mV, uma saída de 20 mV; ou, se a entrada for 1 V, a saída será 10 V. O ganho informa a relação matemática entre a saída e a entrada do bloco. Podemos indicar quando um sinal está no domínio do tempo, ou seja, quando é uma função do tempo, escrevendo-o como $f(t)$. Portanto, para uma entrada de $y(t)$ e uma saída de $x(t)$,

$$\text{ganho} = \frac{\text{saída}}{\text{entrada}} = \frac{x(t)}{y(t)}$$

Entretanto, para muitos sistemas a relação entre a saída e a entrada é na forma de uma equação diferencial, não sendo possível declarar uma função como apenas um número, como um ganho de 10. Não podemos simplesmente dividir a saída pela entrada porque a relação é uma equação diferencial e não uma simples equação algé-

brica. Entretanto, podemos transformar uma equação diferencial em uma equação algébrica usando a denominada **transformada de Laplace**. As equações diferenciais descrevem como os sistemas se comportam com o tempo e são transformadas pelo método da transformada de Laplace em equações algébricas simples, que não envolvem o tempo, com as quais podemos realizar manipulações algébricas normais. Estamos falando de um comportamento no domínio do tempo sendo transformado para o **domínio s** . $F(s)$ é a forma de representar uma função no domínio s . Geralmente usamos a letra maiúscula F para a transformada de Laplace e a letra minúscula f para a função que varia no tempo $f(t)$.

Definirmos a relação entre a saída e a entrada em termos de uma **função de transferência** significa estabelecer a relação entre a transformada de Laplace da saída e a da entrada. Suponha que a entrada de um sistema linear tenha uma transformada de Laplace de $Y(s)$ e a transformada de Laplace da saída seja $X(s)$. A função de transferência $G(s)$ do sistema é então definida como

$$\text{função de transferência} = \frac{\text{transformada de Laplace da saída}}{\text{transformada de Laplace da entrada}}$$

$$G(s) = \frac{X(s)}{Y(s)}$$

com todas as condições iniciais em zero, ou seja, considerando uma saída zero quando a entrada for zero e taxa zero de variação da saída com o tempo quando a taxa de variação da entrada com o tempo for zero. Portanto, a transformada da saída é $X(s) = G(s)Y(s)$, ou seja, o produto da transformada da entrada pela função de transferência. Se representarmos um sistema no domínio s por um diagrama em bloco (Figura 13.1(b)) então $G(s)$ é a função do bloco que recebe uma entrada $Y(s)$ e a converte em uma saída $X(s)$.

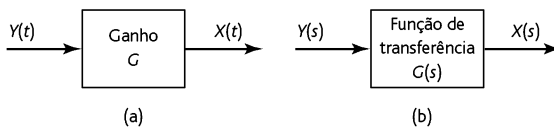


Figura 13.1 Diagramas em bloco: (a) no domínio do tempo, (b) no domínio s .

13.1.1 Transformadas de Laplace

Para obter a transformada de Laplace de uma equação diferencial que inclui grandezas que são funções do tempo podemos usar tabelas juntamente com algumas regras básicas (o Apêndice A inclui uma tabela desta e fornece detalhes das regras). A Figura 13.2 mostra as transformadas básicas de formas comuns de entradas.

As regras a seguir são algumas das que estão envolvidas no uso de transformadas de Laplace:

- 1 Se uma função do tempo for multiplicada por uma constante, então a transformada de Laplace é multiplicada pela mesma constante, ou seja,

$$af(t) \text{ tem a transformada de } aF(s)$$

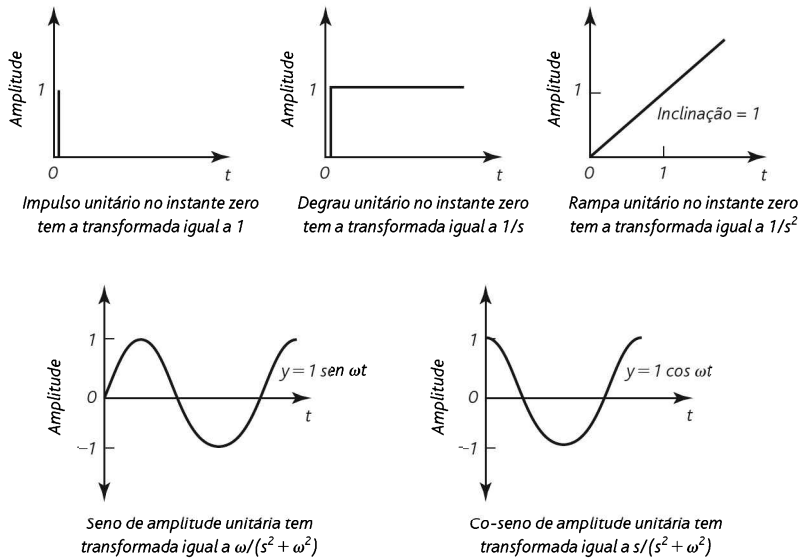


Figura 13.2 Transformadas de Laplace para entradas comuns.

Por exemplo, a transformada de Laplace de uma entrada em degrau de 6 V para um sistema elétrico é exatamente seis vezes a transformada do degrau unitário, portanto 6s.

- 2 Se uma equação inclui a soma de, digamos, duas grandezas separadas que são funções do tempo, então a transformada da equação é a soma das duas transformadas de Laplace em separado, ou seja,

$$f(t) + g(t) \text{ tem a transformada de } F(s) + G(s)$$

- 3 A transformada de Laplace da derivada primeira de uma função é

$$\text{a transformada de } \left\{ \frac{d}{dt} f(t) \right\} = sF(s) - f(0)$$

onde $f(0)$ é o valor inicial de $f(t)$ quando $t = 0$. Entretanto, quando estamos lidando com uma função de transferência, temos todas as condições iniciais nulas.

- 4 A transformada de Laplace da derivada segunda de uma função é

$$\text{a transformada de } \left\{ \frac{d^2}{dt^2} f(t) \right\} = s^2 F(s) - sf(0) - \frac{d}{dt} f(0)$$

em que $df(0)/dt$ é o valor inicial da derivada primeira de $f(t)$ quando temos $t = 0$. Entretanto, quando estamos lidando com uma função de transferência, temos todas as condições iniciais nulas.

- 5 A transformada de Laplace da integral de uma função é

$$\text{a transformada de } \left\{ \int_0^t f(t) dt \right\} = \frac{1}{s} F(s)$$

Portanto, na obtenção das transformadas de equações diferenciais e integrais, quando todas as condições iniciais são nulas, temos:

Substitua a função do tempo $f(t)$ por $F(s)$.

Substitua a derivada primeira $df(t)/dt$ por $sF(s)$.

Substitua a derivada segunda $d^2f(t)/dt^2$ por $s^2F(s)$.

Substitua a integral $\int f(t) dt$ por $F(s)/s$.

Quando ocorrem manipulações algébricas no domínio s , então o resultado pode ser transformado de volta para o domínio do tempo usando-se a tabela de transformadas de forma inversa, ou seja, encontrando a função no domínio do tempo que corresponde ao resultado no domínio s . Frequentemente a transformada tem que ser rearranjada para ser colocada na forma dada na tabela. As inversões a seguir são úteis; para conhecer mais inversões, consulte o Apêndice A:

	<i>Transformada de Laplace</i>	<i>Função do tempo</i>
1	$\frac{1}{s+a}$	e^{-at}
2	$\frac{a}{s(s+a)}$	$(1 - e^{-at})$
3	$\frac{b-a}{(s+a)(s+b)}$	$e^{-at} - e^{-bt}$
4	$\frac{s}{(s+a)^2}$	$(1 - at)e^{-at}$
5	$\frac{a}{s^2(s+a)}$	$t - \frac{1 - e^{-at}}{a}$

As seções a seguir ilustram estas aplicações em sistemas de primeira e segunda ordens.

13.2 Sistemas de primeira ordem

Considere um sistema onde a relação entre a entrada e a saída está na forma de uma equação diferencial. A equação diferencial de um sistema de primeira ordem é da forma

$$a_1 \frac{dx}{dt} + a_0 x = b_0 y$$

em que a_1, a_0, b_0 são constantes, y é a entrada e x é a saída, sendo ambas funções do tempo. A transformada de Laplace desta equação, com todas as condições iniciais nulas, é

$$a_1 s X(s) + a_0 X(s) = b_0 Y(s)$$

e assim podemos escrever a função de transferência $G(s)$ como

$$G(s) = \frac{X(s)}{Y(s)} = \frac{b_0}{a_1 s + a_0}$$

Isto pode ser rearranjado para obtermos

$$G(s) = \frac{b_0/a_0}{(a_1/a_0)s + 1} = \frac{G}{\tau s + 1}$$

em que G é o ganho do sistema nas condições de estado estacionário, ou seja, não há o termo dx/dt . (a_1/a_0) é a constante de tempo τ do sistema (veja a Seção 10.2.3).

13.2.1 Sistema de primeira ordem com entrada em degrau

Quando um sistema de primeira ordem é submetido a uma entrada de degrau unitário, então $Y(s) = 1/s$ e a transformada de saída $X(s)$ é

$$X(s) = G(s)Y(s) = \frac{G}{s(\tau s + 1)} = G \frac{(1/\tau)}{s(s + 1/\tau)}$$

Portanto, como temos a transformada na forma $a/s(s + a)$, usando a transformada inversa listada como item 2 na seção anterior, temos

$$x = G(1 - e^{-t/\tau})$$

13.2.2 Exemplos de sistemas de primeira ordem

Os exemplos a seguir ilustram os pontos levantados anteriormente na consideração da função de transferência de um sistema de primeira ordem e o seu comportamento quando submetido a uma entrada em degrau:

- 1 Considere um circuito que tem uma resistência R em série com uma capacitância C . A entrada do circuito é v e a saída é a diferença de potencial v_C no capacitor. A equação diferencial que relaciona a entrada e a saída é

$$v = RC \frac{dv_C}{dt} + v_C$$

Determine a função de transferência.

Tomando a transformada de Laplace, com todas as condições iniciais nulas, então

$$V(s) = RCsV_C(s) + V_C(s)$$

Portanto, a função de transferência é

$$G(s) = \frac{V_C(s)}{V(s)} = \frac{1}{RCs + 1}$$

- 2 Considere um termopar que tem uma função de transferência que relaciona sua tensão de saída V e a temperatura de entrada como

$$G(s) = \frac{30 \times 10^{-6}}{10s + 1} \text{ V/}^\circ\text{C}$$

Determine a resposta do sistema quando submetido a uma entrada em degrau de amplitude 100°C e, conseqüentemente, o tempo gasto para atingir 95% do valor de estado estacionário.

Visto que a transformada da saída é igual ao produto da função de transferência pela transformada da entrada, então

$$V(s) = G(s) \times \text{entrada}(s)$$

Uma entrada em degrau de amplitude 100°C , que significa que a temperatura do termopar é aumentada abruptamente para 100°C , é $100/s$. Portanto,

$$\begin{aligned} V(s) &= \frac{30 \times 10^{-6}}{10s + 1} \times \frac{100}{s} = \frac{30 \times 10^{-4}}{10s(s + 0,1)} \\ &= 30 \times 10^{-4} \frac{0,1}{s(s + 0,1)} \end{aligned}$$

A parte fracionária é da forma $a/s(s+a)$, e assim a transformada inversa é

$$V = 30 \times 10^{-4} (1 \times e^{-0,1t}) \text{ V}$$

O valor final, ou seja, o valor de estado estacionário, é quando $t \rightarrow \infty$ e, desta forma, é quando o termo exponencial é zero. Portanto, o valor final é $30 \times 10^{-4} \text{ V}$. O tempo gasto para atingir, digamos, 95% disto é dado por

$$0,95 \times 30 \times 10^{-4} = 30 \times 10^{-4} (1 \times e^{-0,1t})$$

Então, $0,05 = e^{-0,1t}$ e $\ln 0,05 = -0,1t$. Este tempo é 30 s.

- 3** Considere uma entrada em rampa de $5t^\circ\text{C}$ para o sistema anterior do termopar, ou seja, a temperatura é elevada em 5°C a cada segundo. Determine como a tensão do termopar varia com o tempo e a tensão após 12 s.

A transformada do sinal da rampa é $5/s^2$. Portanto,

$$V(s) = \frac{30 \times 10^{-6}}{10s + 1} \times \frac{5}{s^2} = 150 \times 10^{-6} \frac{0,1}{s^2(s + 0,1)}$$

A transformada inversa pode ser obtida usando-se o item 5 na lista dada na seção anterior. Portanto,

$$V = 150 \times 10^{-6} \left(t - \frac{1 - e^{-0,1t}}{0,1} \right)$$

Após um tempo de 12 s temos $V = 7,5 \times 10^{-4} \text{ V}$.

- 4** Considere uma entrada impulso de amplitude 100°C , ou seja, o termopar é submetido a um aumento momentâneo de temperatura de 100°C . Determine como a tensão do termopar varia com o tempo e, portanto, a tensão após 2 s.

O impulso tem uma transformada de 100. Portanto,

$$V(s) = \frac{30 \times 10^{-6}}{10s + 1} \times 100 = 3 \times 10^{-4} \frac{1}{s + 0,1}$$

Assim, $V = 3 \times 10^{-4} e^{-0,1t} \text{ V}$. Após 2 s, a tensão no termopar é $V = 1,8 \times 10^{-4} \text{ V}$.

13.3 Sistemas de segunda ordem

Para um sistema de segunda ordem, a relação entre a entrada y e a saída x é descrita por uma equação diferencial da forma

$$a_2 \frac{d^2x}{dt^2} + a_1 \frac{dx}{dt} + a_0x = b_0y$$

em que a_2, a_1, a_0 e b_0 são constantes. A transformada de Laplace desta equação, com todas as condições iniciais nulas, é

$$a_2s^2X(s) + a_1sX(s) + a_0X(s) = b_0Y(s)$$

Portanto,

$$G(s) = \frac{X(s)}{Y(s)} = \frac{b_0}{a_2s^2 + a_1s + a_0}$$

Uma forma alternativa de escrever a equação diferencial para um sistema de segunda ordem é

$$\frac{d^2x}{dt^2} + 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2x = b_0\omega_n^2y$$

em que ω_n é a frequência angular natural na qual o sistema oscila e ζ é a razão de amortecimento. A transformada de Laplace desta equação fornece

$$G(s) = \frac{X(s)}{Y(s)} = \frac{b_0\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Esta equação é a forma geral da função de transferência para um sistema de segunda ordem.

13.3.1 Sistema de segunda ordem com entrada em degrau

Quando um sistema de segunda ordem é submetido a uma entrada de degrau unitário, ou seja, $Y(s) = 1/s$, então a transformada da saída é

$$X(s) = G(s)Y(s) = \frac{b_0\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)}$$

Esta pode ser rearranjada como

$$X(s) = \frac{b_0\omega_n^2}{s(s + p_1)(s + p_2)}$$

em que p_1 e p_2 são as raízes da equação

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

Portanto, usando a equação para as raízes de uma equação quadrática,

$$p = \frac{-2\zeta\omega_n \pm \sqrt{4\zeta^2\omega_n^2 - 4\omega_n^2}}{2}$$

e as duas raízes p_1 e p_2 são

$$p_1 = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1} \quad p_2 = -\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1}$$

Com $\zeta > 1$, o termo da raiz quadrada é real e o sistema é sobreamortecido. Para determinar a transformada inversa, podemos usar o método de frações parciais (veja o Apêndice A) para dividir a expressão em frações simples ou usar o item 14 na tabela de transformadas no Apêndice A; o resultado em qualquer caso é

$$x = \frac{b_0\omega_n^2}{p_1p_2} \left[1 - \frac{p_2}{p_2 - p_1} e^{-p_2t} + \frac{p_1}{p_2 - p_1} e^{-p_1t} \right]$$

Com $\zeta = 1$, o termo da raiz quadrada é zero e $p_1 = p_2 = -\omega_n$. O sistema é criticamente amortecido. A equação torna-se então

$$X(s) = \frac{b_0\omega_n^2}{s(s + \omega_n)^2}$$

Esta equação pode ser expandida pelo método de frações parciais (veja o Apêndice A), obtendo-se

$$Y(s) = b_0 \left[\frac{1}{s} - \frac{1}{s + \omega_n} - \frac{\omega_n}{(s + \omega_n)^2} \right]$$

Portanto

$$x = b_0[1 - e^{-\omega_n t} - \omega_n t e^{-\omega_n t}]$$

Com $\zeta < 1$, temos

$$x = b_0 \left[1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1 - \zeta^2}} \text{sen}(\omega_n \sqrt{1 - \zeta^2} t + \phi) \right]$$

em que $\cos \phi = \zeta$. Esta é uma oscilação subamortecida.

13.3.2 Exemplos de sistemas de segunda ordem

Os exemplos a seguir ilustram os casos citados anteriormente:

- 1 Qual é o estado de amortecimento de um sistema que tem a seguinte função de transferência e está sujeito a uma entrada em degrau?

$$G(s) = \frac{1}{s^2 + 8s + 16}$$

Para uma entrada degrau unitário $Y(s) = 1/s$, a transformada da saída é

$$X(s) = G(s)Y(s) = \frac{1}{s(s^2 + 8s + 16)} = \frac{1}{s(s + 4)(s + 4)}$$

As raízes de $s^2 + 8s + 16$ são então $p_1 = p_2 = -4$. As duas raízes são reais e iguais e o sistema é criticamente amortecido.

2 Um braço de robô tem a seguinte função de transferência submetida a uma entrada rampa unitária. Qual será a saída?

$$G(s) = \frac{K}{(s + 3)^2}$$

A transformada da saída $X(s)$ é

$$X(s) = G(s)Y(s) = \frac{K}{(s + 3)^2} \times \frac{1}{s^2}$$

Usando frações parciais (veja Apêndice A), temos

$$X(s) = \frac{K}{9s^2} - \frac{2K}{9(s + 3)} + \frac{K}{9(s + 3)^2}$$

Portanto, a transformada inversa é

$$x = \frac{1}{9}Kt - \frac{2}{9}Ke^{-3t} + \frac{1}{9}Kte^{-3t}$$

13.4 Sistemas em série

Se um sistema consiste de subsistemas em série, como na Figura 13.3, então a função de transferência $G(s)$ do sistema é dada por

$$\begin{aligned} G(s) &= \frac{X(s)}{Y(s)} = \frac{X_1(s)}{Y(s)} \times \frac{X_2(s)}{X_1(s)} \times \frac{X(s)}{X_2(s)} \\ &= G_1(s) \times G_2(s) \times G_3(s) \end{aligned}$$

A função de transferência do sistema como um todo é o produto das funções de transferência dos elementos em série.

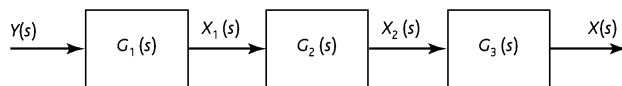


Figura 13.3 Sistemas em série.

13.4.1 Exemplos de sistemas em série

Os exemplos a seguir ilustram isto. Considere que, quando sistemas são interconectados, nenhuma interação ocorre entre os blocos que resulte em alteração de suas funções de transferência, por exemplo, os circuitos elétricos podem ter problemas quando partes do circuito interagem com a carga.

1 Qual é a função de transferência para um sistema que consiste de três elementos em série, sendo as funções de transferência dos elementos 10 , $2/s$ e $4/(s + 3)$?

Usando a equação desenvolvida antes,

$$G(s) = 10 \times \frac{2}{s} \times \frac{4}{s + 3} = \frac{80}{s(s + 3)}$$

2 Um motor CC controlado pelo campo consiste de três subsistemas em série: o circuito do campo, a bobina de armadura e a carga. A Figura 13.4 ilustra o arranjo das funções de transferência dos subsistemas. Determine a função de transferência geral?

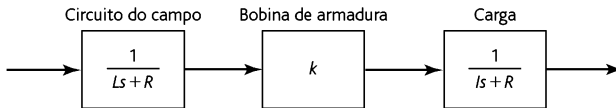


Figura 13.4 Motor CC controlado pelo campo.

A função de transferência geral é o produto das funções de transferência dos elementos em série. Portanto,

$$G(s) = \frac{1}{Ls + R} \times k \times \frac{1}{Is + c} = \frac{k}{(Ls + R)(Is + c)}$$

13.5 Sistemas com malhas de realimentação

A Figura 13.5 mostra um sistemas simples contendo uma realimentação negativa. Com a **realimentação negativa**, os sinais de entrada do sistema e de realimentação são subtraídos no ponto de soma. O termo **caminho direto** é usado para o caminho que tem a função de transferência $G(s)$ na figura e **caminho de realimentação**, para o que tem a função $H(s)$. O sistema completo é denominado **sistema em malha fechada**.

Para o sistema com realimentação negativa, a entrada do subsistema que tem a função de transferência $G(s)$ é $Y(s)$ menos o sinal de realimentação. A malha de realimentação tem uma função de transferência de $H(s)$ e tem uma entrada $X(s)$, assim, o sinal de realimentação é $H(s)X(s)$. Portanto, o elemento $G(s)$ tem uma entrada de $Y(s) - H(s)X(s)$ e uma saída de $X(s)$, e assim

$$G(s) = \frac{X(s)}{Y(s) - H(s)X(s)}$$

Esta pode ser rearranjada para obtermos

$$\frac{X(s)}{Y(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

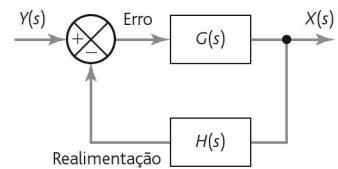


Figura 13.5 Sistema de realimentação negativa.

Portanto, a função de transferência geral para um sistema de realimentação negativa $T(s)$ é

$$T(s) = \frac{X(s)}{Y(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

13.5.1 Exemplos de sistemas com realimentação negativa

Os exemplos a seguir ilustram o que foi dito antes:

- 1 Qual é a função de transferência geral para um sistema em malha fechada com uma função de transferência do caminho direto de $2/(s + 1)$ e um caminho de realimentação negativa de $5s$?

Usando a equação desenvolvida antes,

$$T(s) = \frac{G(s)}{1 + G(s)H(s)} = \frac{2/(s + 1)}{1 + [2/(s + 1)]5s} = \frac{2}{11s + 1}$$

- 2 Considere um motor CC controlado pela armadura (Figura 13.6). Este tem um caminho direto que consiste de três elementos: o circuito de armadura com uma função de transferência de $1/(Ls + R)$, a bobina de armadura com uma função de transferência k , e a carga com uma função de transferência $1/(Is + c)$. Existe um caminho de realimentação negativa com uma função de transferência K . Determine a função de transferência geral do sistema.

A função de transferência do caminho direto para os elementos em série é o produto das funções de transferência destes elementos, ou seja,

$$G(s) = \frac{1}{Ls + R} \times k \times \frac{1}{Is + c} = \frac{k}{(Ls + R)(Is + c)}$$

O caminho de realimentação tem uma função de transferência de K . Portanto, a função de transferência geral é

$$\begin{aligned} T(s) &= \frac{G(s)}{1 + G(s)H(s)} = \frac{\frac{k}{(Ls + R)(Is + c)}}{1 + \frac{kK}{(Ls + R)(Is + c)}} \\ &= \frac{k}{(Ls + R)(Is + c) + kK} \end{aligned}$$

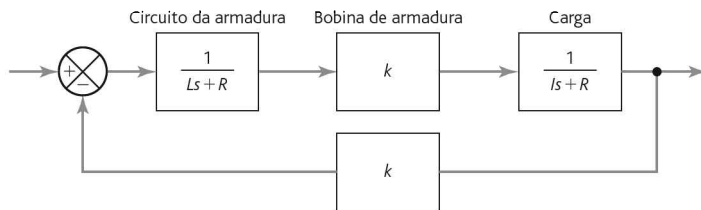


Figura 13.6 Motor CC controlado pela armadura.

13.6 Efeito da localização dos polos na resposta transiente

Podemos definir um sistema como sendo **estável** se, quando ele recebe uma entrada, o seu transiente termina com o tempo deixando o sistema na condição de estado estacionário. Diz-se que um sistema é **instável** se o transiente não termina com o tempo, mas aumenta sua amplitude, e a condição de estado estacionário nunca é atingida.

Considere uma entrada como sendo um impulso unitário em um sistema de primeira ordem com uma função de transferência de $G(s) = 1/(s + 1)$. A saída do sistema $X(s)$ é

$$X(s) = \frac{1}{s + 1} \times 1$$

e, portanto, $x = e^{-t}$. Conforme t aumenta, a saída diminui, se tornando ao fim zero. Agora considere a entrada impulso unitário para um sistema com a função de transferência $G(s) = 1/(s - 1)$. A saída é

$$X(s) = \frac{1}{s - 1} \times 1$$

e, portanto, $x = e^t$. Conforme t aumenta, a saída aumenta. Assim, um impulso momentâneo no sistema resulta em uma saída cada vez maior; este sistema é instável.

Para uma função de transferência, os valores de s que tornam a função de transferência infinita são denominados seus polos; eles são as raízes da equação característica. Assim, para $G(s) = 1/(s + 1)$, existe um polo de $s = -1$. Para $G(s) = 1/(s - 1)$, existe um polo de $s = +1$. Portanto, um sistema de primeira ordem é estável se o polo é negativo, e instável se ele for positivo (Figura 13.7).

Para um sistema de segunda ordem com função de transferência

$$G(s) = \frac{b_0 \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2}$$

quando submetido a uma entrada de impulso unitário,

$$X(s) = \frac{b_0 \omega_n^2}{(s + p_1)(s + p_2)}$$

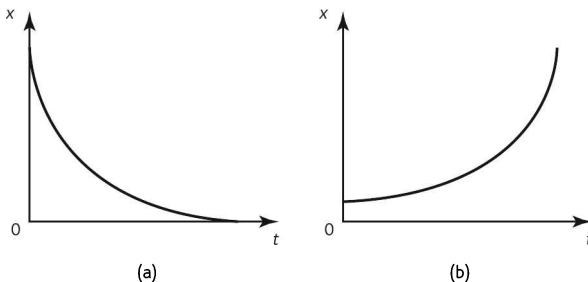


Figura 13.7 Sistemas de primeira ordem: (a) polo negativo, (b) polo positivo.

em que p_1 e p_2 são as raízes da equação

$$s^2 + 2\zeta\omega_n s + \omega_n = 0$$

Usando a equação para o cálculo das raízes de uma equação quadrática, temos

$$p = \frac{-2\zeta\omega_n \pm \sqrt{4\zeta^2\omega_n^2 - 4\omega_n^2}}{2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

Dependendo do valor do fator de amortecimento, o termo sob o sinal da raiz quadrada pode ser real ou imaginário. Quando há um termo imaginário, a saída envolve oscilações. Por exemplo, suponha que temos um sistema de segunda ordem com a função de transferência

$$G(s) = \frac{1}{[s - (-2 + j1)][s - (-2 - j1)]}$$

ou seja, $p = -2 \pm j1$. Quando submetido a uma entrada de impulso unitário, a saída é $e^{-2t} \sin t$. A amplitude da oscilação, ou seja, e^{-2t} , termina à medida que o tempo aumenta, e o efeito do impulso é uma oscilação que decai gradualmente (Figura 13.8(a)). O sistema é estável.

Entretanto, suponha que temos um sistema com a função de transferência

$$G(s) = \frac{1}{[s - (2 + j1)][s - (2 - j1)]}$$

ou seja, $p = +2 \pm j1$. Quando submetido a um entrada de impulso unitário, a saída é $e^{2t} \sin t$. A amplitude da oscilação, ou seja, e^{2t} , aumenta à medida que o tempo aumenta (Figura 13.8(b)). O sistema é instável.

Em geral, quando um impulso é aplicado a um sistema, a saída é uma soma de termos exponenciais. Se apenas um destes termos for uma exponencial crescente, então a saída continua a crescer e o sistema é instável. Quando existe um par de polos envolvendo termos imaginários positivos ou negativos, então a saída é oscilante.

Um sistema é estável se a parte real de todos os polos for negativa.

Um sistema é instável se a parte real de qualquer dos seus polos for positiva.

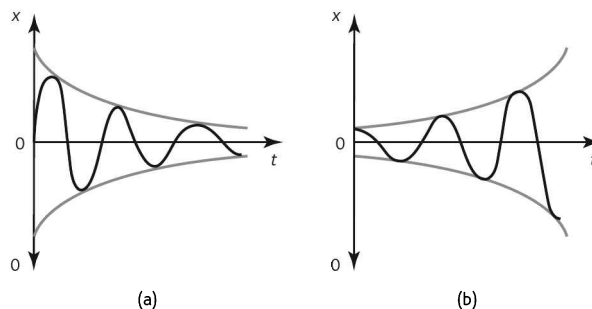


Figura 13.8 Sistemas de segunda ordem.

13.6.1 Plano s

Podemos representar graficamente as posições dos polos de um sistema colocando no eixo x as partes reais e no y as imaginárias. Este gráfico é denominado **plano s** . A localização dos polos no plano determina a estabilidade do sistema. A Figura 13.9 mostra este plano e como a localização das raízes afeta a resposta de um sistema.

13.6.2 Compensação

A saída de um sistema pode ser instável ou talvez a resposta seja muito lenta ou haja muito *overshoot*. Os sistemas podem ter suas respostas às entradas alteradas incluindo-se **compensadores**. Um compensador é um bloco que é incorporado no sistema de modo que altere a função de transferência geral dele para obter a características desejadas.

Como uma ilustração do uso de um compensador, considere um sistema de controle de posição que tem um caminho de realimentação negativa com uma função de transferência de 1 e dois subsistemas em seu caminho direto: um compensador com uma função de transferência de K e um sistema motor/acionamento com uma função de transferência de $1/s(s + 1)$. Qual valor de K é necessário para o sistema ser criticamente amortecido? O caminho direto tem uma função de transferência de $K/s(s + 1)$ e o caminho de realimentação tem uma função de transferência de 1. A função de transferência deste sistema é

$$T(s) = \frac{G(s)}{1 + G(s)H(s)} = \frac{\frac{K}{s(s + 1)}}{1 + \frac{K}{s(s + 1)}} = \frac{K}{s(s + 1) + K}$$

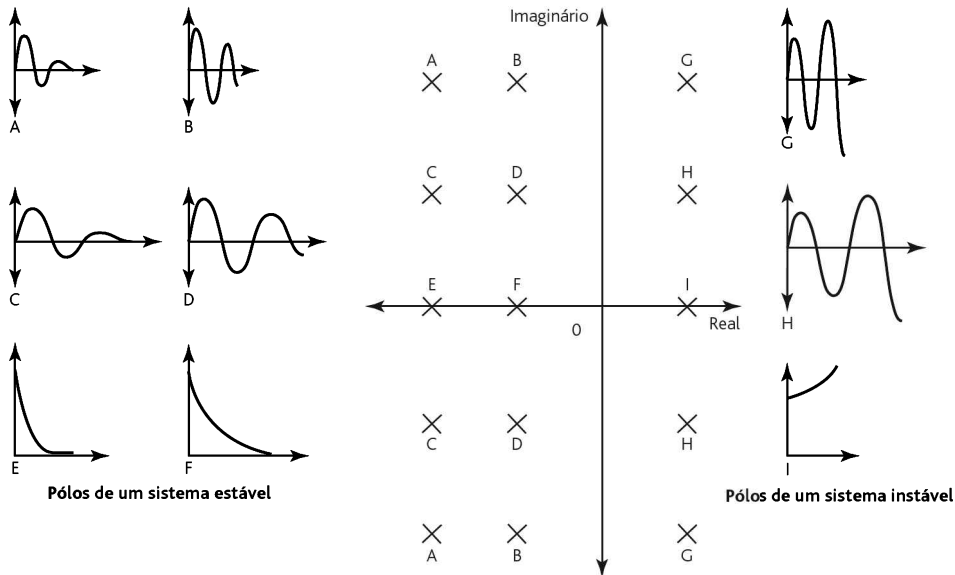


Figura 13.9 O plano s .

O denominador é $s^2 + s + K$. As raízes são

$$s = \frac{-1 \pm \sqrt{1 - 4K}}{2}$$

Para ser criticamente amortecido, temos que ter $1 - 4K = 0$ e o compensador tem que ter ganho proporcional de $K = \frac{1}{4}$.

Resumo

A **função de transferência** $G(s)$ de um sistema é (transformada de Laplace da saída)/(transformada de Laplace da entrada). Para obter as transformadas de equações diferenciais ou integrais quando todas as condições iniciais são nulas: substituímos uma função do tempo $f(t)$ por $F(s)$, substituímos a derivada primeira $df(t)/dt$ por $sF(s)$, substituímos a derivada segunda $d^2f(t)/dt^2$ por $s^2F(s)$ e substituímos a integral $\int f(t) dt$ por $F(s)/s$.

Um **sistema de primeira ordem** tem uma função de transferência da forma $G/(ts + 1)$, em que τ é a constante de tempo. Um **sistema de segunda ordem** tem uma função de transferência da forma

$$G(s) = \frac{b_0\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

em que ζ é o fator de amortecimento e ω_n é a frequência angular natural.

Os valores de s que tornam a função de transferência infinita são denominados polos; eles são as raízes da equação característica. Um sistema é estável se a parte real de todos os seus polos for negativa, e instável se a parte real de qualquer um dos polos for positiva.

Problemas

13.1 Quais são as funções de transferência para os sistemas dados pelas seguintes relações de entrada/saída?

(a) Um sistema hidráulico tem uma entrada q e uma saída h em que

$$q = A \frac{dh}{dt} + \frac{\rho gh}{R}$$

(b) Um sistema massa-mola-amortecedor com uma entrada F e uma saída x , em que

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = F$$

(c) Um circuito RLC com uma entrada v e uma saída v_C , em que

$$v = RC \frac{dv_C}{dt} + LC \frac{d^2v_C}{dt^2} + v_C$$

- 13.2 Quais são as constantes de tempo dos sistemas seguintes dadas as funções de transferência (a) $G(s) = 5/(3s + 1)$, (b) $G(s) = 2/(2s + 3)$?
- 13.3 Determine como as saídas dos seguintes sistemas variam com o tempo quando submetidos a uma entrada de degrau unitário no instante $t = 0$: (a) $G(s) = 2/(s + 2)$, (b) $G(s) = 10/(s + 5)$

- 13.4 Qual é o estado de amortecimento dos sistemas que têm a seguintes funções de transferência?

(a) $G(s) = \frac{5}{s^2 - 6s + 16}$, (b) $G(s) = \frac{10}{s^2 + s + 100}$,
 (c) $G(s) = \frac{2s + 1}{s^2 + 2s + 1}$, (d) $G(s) = \frac{3s + 20}{s^2 + 2s + 20}$

- 13.5 Qual é a saída de um sistema com a função de transferência $s/(s + 3)^2$ e submetido a uma entrada em degrau no instante $t = 0$?

- 13.6 Qual é a saída de um sistema que tem a função de transferência $G = 2/[(s + 3) \times (s + 4)]$ e é submetido a um impulso unitário?

- 13.7 Quais são as funções de transferência gerais dos seguintes sistemas com realimentação negativa?

Caminho direto

Caminho de realimentação

(a) $G(s) = \frac{4}{s(s + 1)}$

$H(s) = \frac{1}{s}$

(b) $G(s) = \frac{2}{s + 1}$

$H(s) = \frac{1}{s + 2}$

(c) $G(s) = \frac{4}{(s + 2)(s + 3)}$

$H(s) = 5$

(d) dois elementos em série $G_1(s) = 2/(s + 2)$ e $G_2(s) = 1/s$ $H(s) = 10$

- 13.8 Qual é a função de transferência geral para um sistema em malha fechada que tem uma função de transferência no caminho direto de $5/(s + 3)$ e uma função de transferência no caminho de realimentação de 10?

- 13.9 Um sistema em malha fechada tem um caminho direto com dois elementos em série com funções de transferência 5 e $1/(s + 1)$. Se o caminho de realimentação tem uma função de transferência $2/s$, qual é a função de transferência geral do sistema?

- 13.10 Um sistema em malha fechada tem um caminho direto com dois elementos em série com funções de transferência 2 e $1/(s + 1)$. Se o caminho de realimentação tem uma função de transferência s , qual é a função de transferência geral do sistema?

- 13.11 Um sistema tem uma função de transferência de $1/[(s + 1)(s + 2)]$. Quais são seus polos?

- 13.12 Dos sistemas a seguir, quais são estáveis e quais são instáveis?

(a) $G(s) = 1/[(s + 5)(s + 2)]$,

(b) $G(s) = 1/[(s - 5)(s + 2)]$,

(c) $G(s) = 1/[(s - 5)(s - 5)]$,

(d) $G(s) = 1/(s^2 + s + 1)$,

(e) $G(s) = 1/(s^2 - 2s + 3)$.

14

Resposta de Frequência

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Explicar o significado da função resposta em frequência.
- Analisar a resposta em frequência de sistemas submetidos a entradas senoidais.
- Desenhar e interpretar gráficos de Bode.
- Usar os gráficos de Bode para identificação de sistemas.
- Explicar o termo largura de banda.
- Explicar como as margens de ganho e de fase podem ser usadas para indicar a estabilidade de um sistema.

14.1 Entrada senoidal

Os dois últimos capítulos abordaram a resposta dos sistemas às entradas em degrau, impulso e rampa. Este capítulo estende esta análise para o caso da entrada senoidal. Para muitos sistemas de controle, esta não é uma entrada comum, porém ela pode ser um teste útil, visto que a forma como os sistemas respondem a ela representa uma fonte muito útil de informações que ajudam no projeto e análise de sistemas. Além disso, muitos outros sinais podem ser considerados como a soma de sinais senoidais. Em 1822, Jean Baptiste Fourier propôs que qualquer forma de onda periódica, como, por exemplo, uma forma de onda quadrada, poderia ser obtida a partir de uma combinação de formas de onda senoidais, e considerando o comportamento de um sistema em resposta a cada uma dessas senoides individualmente, é possível determinar a resposta a formas de onda mais complexas.

14.1.1 Resposta de um sistema a uma entrada senoidal

Considere um sistema de primeira ordem o qual é descrito pela seguinte equação diferencial

$$a_1 \frac{dx}{dt} + a_0 x = b_0 y$$

em que y é a entrada e x é a saída. Suponha que temos uma entrada senoidal de amplitude unitária $y = \text{sen } \omega t$. Qual será a saída? Acabamos obtendo a senoide $b_0 \text{sen } \omega t$ quando somamos $a_1 dx/dt$ e $a_0 x$. Porém senoídes têm a propriedade de que, quando derivadas, o resultado também é uma senoide e com a mesma frequência (um cosseno é uma senoide da forma $\text{sen}(\omega t + 90^\circ)$). Isto se aplica não importa quantas vezes realizamos a derivada. Portanto, esperamos que a resposta de estado estacionário x será senoidal e com a mesma frequência. Entretanto, a saída pode ser diferente em amplitude e fase em relação à entrada.

14.2 Fasores

Na análise de sinais senoidais, é conveniente o uso de **fasores**. Considere uma senoide descrita pela equação $v = V \text{sen}(\omega t + \phi)$, em que V é a amplitude, ω é a frequência angular e ϕ é o ângulo de fase. O fasor pode ser representado por uma linha de comprimento $|V|$ que faz um ângulo ϕ com o eixo de referência do fasor (Figura 14.1). As linhas $||$ são usadas para indicar que estamos interessados na magnitude ou amplitude da grandeza quando especificamos seu comprimento. Para especificar completamente uma grandeza fasorial, é necessário determinar uma amplitude e um ângulo. A convenção geralmente adotada é escrever o fasor em tipo negrito, não itálico, como, por exemplo, V . Quando este símbolo é visto, interpretamos como uma grandeza tendo um módulo e um ângulo.

Este fasor pode ser descrito por uma notação de número complexo. Um valor complexo pode ser representado por $(x + jy)$, em que x é a parte real e y é a parte imaginária do número complexo. Em um gráfico com a componente imaginária no eixo y e a parte real no eixo x , x e y são as coordenadas cartesianas do ponto que representa o número complexo (Figura 14.2(a)).

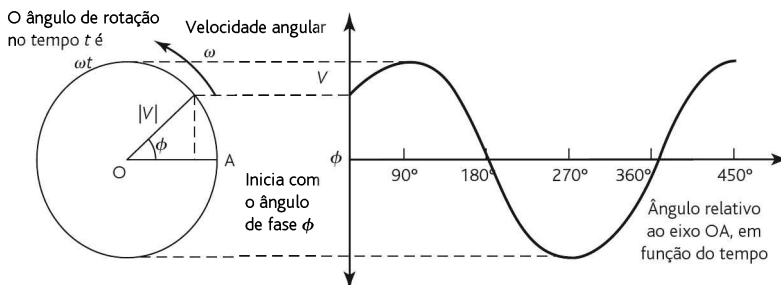


Figura 14.1 Representação de um sinal senoidal por um fasor.

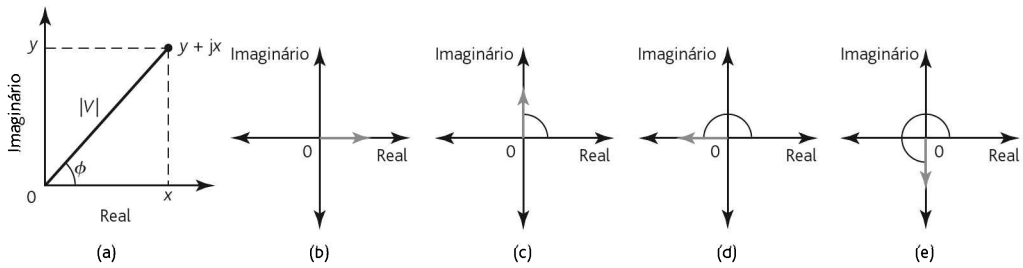


Figura 14.2 (a) Representação complexa de um fasor, (b) 0°, (c) 90°, (d) 270°, (e) 360°.

Se traçarmos uma linha reta deste ponto à origem do gráfico para representar o fasor, então teremos o ângulo de fase ϕ do fasor representado por

$$\text{tg } \phi = \frac{y}{x}$$

e o seu comprimento, fazendo uso do teorema de Pitágoras, é

$$\text{comprimento do fasor } |V| = \sqrt{x^2 + y^2}$$

Visto que $x = |V| \cos \phi$ e $y = |V| \text{sen } \phi$, então podemos escrever

$$V = x + jy = |V| \cos \theta + j|V| \text{sen } \theta = |V|(\cos \theta + j \text{sen } \theta)$$

Portanto, com a especificação das partes real e imaginária de um número complexo, o fasor é especificado

Considere um fasor de comprimento 1 e ângulo de fase 0° (Figura 14.2(b)). Ele tem uma representação complexa de $1 + j0$. Agora considere um fasor de mesmo comprimento, mas com um ângulo de fase de 90° (Figura 14.2(c)). Ele tem uma representação complexa de $0 + j1$. Portanto, a rotação de 90° em um fasor no sentido anti-horário corresponde a multiplicar o fasor por j . Se agora girarmos este fasor por mais 90° (Figura 14.2(d)), então, seguindo a mesma regra de multiplicação, temos o fasor original multiplicado por j . Porém o fasor é o mesmo que o original em sentido oposto, ou seja, ele foi apenas multiplicado por -1 . Desta forma, $j^2 = -1$ e assim $j = \sqrt{-1}$. A rotação do fasor original em um ângulo de 270°, ou seja, $3 \times 90^\circ$, é equivalente a multiplicar o fasor original por $j^3 = j(j^2) = -j$.

Para ilustrar isto, considere uma tensão v que varia senoidalmente com o tempo de acordo com a equação

$$v = 10 \text{ sen}(\omega t + 30^\circ) \text{ V}$$

Quando representada por um fasor, quais são (a) seu comprimento, (b) seu ângulo em relação ao eixo de referência, (c) suas partes real e imaginária quando representado por um número complexo?

- (a) O fasor tem um comprimento que representa a amplitude da senoide, que é 10 V.
- (b) O ângulo do fasor em relação ao eixo de referência é igual ao ângulo de fase, que vale 30°.
- (c) A parte real é dada pela equação $x = 10 \cos 30^\circ = 8,7 \text{ V}$ e a parte imaginária é dada por $y = 10 \text{ sen } 30^\circ = 5,0 \text{ V}$. Portanto, o fasor é especificado por $8,7 + j5,0 \text{ V}$.

14.2.1 Equações fasoriais

Considere um fasor como sendo a representação de uma senoide de amplitude unitária $x = \text{sen } \omega t$. A derivada de uma senoide é $dx/dt = \omega \cos \omega t$. Porém, podemos escrever também na forma $dx/dt = \omega \text{sen}(\omega t + 90^\circ)$. Em outras palavras, a derivada resulta apenas em um fasor de comprimento aumentado por um fator ω e rotacionado 90° em relação ao fasor original. Portanto, em notação complexa, multiplicamos o fasor original por $j\omega$, visto que a multiplicação por j é equivalente a uma rotação de 90° .

Portanto, a equação diferencial

$$a_1 \frac{dx}{dt} + a_0 x = b_0 y$$

pode ser escrita, em notação complexa, como uma **equação fasorial** da forma

$$j\omega a_1 \mathbf{X} + a_0 \mathbf{X} = b_0 \mathbf{Y}$$

em que as letras em negrito, não itálico, indicam que as variáveis se referem a fasores. Podemos dizer que a equação diferencial, que era uma equação no domínio do tempo, foi transformada em uma equação no **domínio da frequência**. Esta equação pode ser reescrita como

$$(j\omega a_1 + a_0) \mathbf{X} = b_0 \mathbf{Y}$$

$$\frac{\mathbf{X}}{\mathbf{Y}} = \frac{b_0}{j\omega a_1 + a_0}$$

Porém, na Seção 13.2, quando a mesma equação diferencial foi escrita no domínio s , tínhamos

$$G(s) = \frac{X(s)}{Y(s)} = \frac{b_0}{a_1 s + a_0}$$

Se substituirmos s por $j\omega$, teremos a mesma equação. Sempre podemos fazer isto para converter do domínio s para o domínio da frequência. Isto nos leva à definição da **função resposta de frequência** ou **função de transferência de frequência** $G(j\omega)$, para o estado estacionário, como

$$G(j\omega) = \frac{\text{fasor de saída}}{\text{fasor de entrada}}$$

Para ilustrar isto, considere a determinação da função resposta de frequência para um sistema com a seguinte função de transferência

$$G(s) = \frac{1}{s + 1}$$

A função resposta de frequência é obtida substituindo-se s por $j\omega$. Assim,

$$G(j\omega) = \frac{1}{j\omega + 1}$$

14.3 Resposta de frequência

O procedimento para determinação da resposta de frequência de um sistema é:

- 1 Substitua s na função de transferência por $j\omega$ para obter a função resposta de frequência.
- 2 A razão de amplitude entre a saída e a entrada é o módulo da função resposta de frequência complexa, ou seja, $\sqrt{x^2 + y^2}$.
- 3 O ângulo de fase entre a saída e a entrada é dado por $\phi = y/x$ ou a razão entre as partes real e imaginária do número complexo que representa a função resposta de frequência.

14.3.1 Resposta de frequência para um sistema de primeira ordem

Um sistema de primeira tem uma função de transferência que pode ser escrita como

$$G(s) = \frac{1}{1 + \tau s}$$

em que τ é a constante de tempo do sistema (veja a Seção 13.2). A função resposta de frequência $G(j\omega)$ pode ser obtida substituindo-se s por $j\omega$. Portanto,

$$G(j\omega) = \frac{1}{1 + j\omega\tau}$$

Podemos colocar esta equação em uma forma mais conveniente multiplicando o numerador e o denominador da expressão por $(1 - j\omega\tau)$ para obtermos

$$G(j\omega) = \frac{1}{1 + j\omega\tau} \times \frac{1 - j\omega\tau}{1 - j\omega\tau} = \frac{1 - j\omega\tau}{1 + j^2\omega^2\tau^2}$$

Mas $j^2 = -1$, assim

$$G(j\omega) = \frac{1}{1 + \omega^2\tau^2} - j \frac{\omega\tau}{1 + \omega^2\tau^2}$$

Este é o formato do $x + jy$ e, como $G(j\omega)$ é o fasor de saída dividido pelo de entrada, temos a amplitude do fasor de saída maior do que a do de entrada por um fator que pode ser escrito como $|G(j\omega)|$, com

$$|G(j\omega)| = \sqrt{x^2 + y^2} = \sqrt{\left(\frac{1}{1 + \omega^2\tau^2}\right)^2 + \left(\frac{\omega\tau}{1 + \omega^2\tau^2}\right)^2} = \frac{1}{\sqrt{1 + \omega^2\tau^2}}$$

$|G(j\omega)|$ informa-nos o quanto a amplitude da saída é maior do que a de entrada. Este parâmetro é geralmente conhecido como **magnitude** ou **ganho**. A diferença de fase ϕ entre o fasor de saída e o de entrada é dada por

$$\text{tg } \phi = \frac{y}{x} = -\omega\tau$$

O sinal negativo indica que o fasor de saída está atrasado em relação ao de entrada por este ângulo.

Os exemplos a seguir ilustram isto:

- 1 Determine a função resposta de frequência, o módulo e a fase de um sistema (um circuito elétrico com um resistor em série com um capacitor em que a saída é obtida deste último) que tem uma função de transferência de

$$G(s) = \frac{1}{RCs + 1}$$

A função resposta de frequência pode ser obtida substituindo-se $j\omega$ por s , e assim

$$G(j\omega) = \frac{1}{j\omega RC + 1}$$

Podemos multiplicar numerador e denominador desta equação por $1 - j\omega RC$, e em seguida, rearranjando a equação, obtemos

$$G(j\omega) = \frac{1}{1 + \omega^2(RC)^2} - j \frac{\omega(RC)}{1 + \omega^2(RC)^2}$$

Portanto,

$$|G(j\omega)| = \frac{1}{\sqrt{1 + \omega^2(RC)^2}}$$

e $\text{tg } \phi = -\omega RC$.

- 2 Determine o módulo e a fase da saída de um sistema quando submetido a uma entrada senoidal de $2 \text{ sen}(3t + 60^\circ)$ se ele tem uma função de transferência de

$$G(s) = \frac{4}{s + 1}$$

A função resposta de frequência é obtida substituindo-se s por $j\omega$. Assim,

$$G(j\omega) = \frac{4}{j\omega + 1}$$

Multiplicando o numerador e o denominador desta equação por $(-j\omega + 1)$, temos

$$G(j\omega) = \frac{-j4\omega + 4}{\omega^2 + 1} = \frac{4}{\omega^2 + 1} - j \frac{4\omega}{\omega^2 + 1}$$

O módulo é

$$|G(j\omega)| = \sqrt{x^2 + y^2} = \sqrt{\frac{4^2}{(\omega^2 + 1)^2} + \frac{4^2\omega^2}{(\omega^2 + 1)^2}} = \frac{4}{\sqrt{\omega^2 + 1}}$$

e o ângulo de fase é dado por $\text{tg } \phi = y/x$, e assim

$$\text{tg } \phi = -\omega$$

Para a entrada especificada, temos $\omega = 3$ rad/s. O módulo é, portanto,

$$|G(j\omega)| = \frac{4}{\sqrt{3^2 + 1}} = 1,3$$

e a fase é dada por $\tan \phi = -3$. Assim, $\phi = -72^\circ$. Este é o ângulo de fase entre a entrada e a saída. Portanto, a saída é $2,6 \sin(3t - 12^\circ)$

14.3.2 Resposta de frequência para um sistema de segunda ordem

Considere um sistema de segunda ordem com a função de transferência (veja a Seção 13.3)

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

em que ω_n é a frequência angular natural e ζ é a razão de amortecimento. A função resposta de frequência é obtida substituindo-se s por $j\omega$. Assim,

$$\begin{aligned} G(j\omega) &= \frac{\omega_n^2}{-\omega^2 + j2\zeta\omega\omega_n + \omega_n^2} = \frac{\omega_n^2}{(\omega_n^2 - \omega^2) + j2\zeta\omega\omega_n} \\ &= \frac{1}{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right] + j2\zeta\left(\frac{\omega}{\omega_n}\right)} \end{aligned}$$

Multiplicando o numerador e o denominador da expressão por

$$\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right] - j2\zeta\left(\frac{\omega}{\omega_n}\right)$$

obtemos

$$G(j\omega) = \frac{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right] - j2\zeta\left(\frac{\omega}{\omega_n}\right)}{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left[2\zeta\left(\frac{\omega}{\omega_n}\right)\right]^2}$$

Esta é a forma $x + jy$, e assim, visto que $G(j\omega)$ é o fasor de saída dividido pelo de entrada, temos a amplitude ou módulo do fasor de saída maior do que o de entrada por um fator que é dado por $\sqrt{x^2 + y^2}$ como

$$|G(j\omega)| = \frac{1}{\sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left[2\zeta\left(\frac{\omega}{\omega_n}\right)\right]^2}}$$

A diferença de fase ϕ entre entrada e saída é dada por $\text{tg } \phi = x/y$, e assim

$$\text{tg } \phi = -\frac{2\zeta\left(\frac{\omega}{\omega_n}\right)}{1 - \left(\frac{\omega}{\omega_n}\right)^2}$$

A existência do sinal negativo é porque a saída está atrasada em relação à entrada.

14.4 Gráficos de Bode

A resposta de frequência de um sistema é o conjunto de valores do módulo $|G(j\omega)|$ e o ângulo de fase ϕ que ocorre quando um sinal de entrada senoidal varia ao longo de uma faixa de frequências. Isto pode ser expresso como dois gráficos, sendo um do módulo $|G(j\omega)|$ em função da frequência angular ω e o outro da fase ϕ em função de ω . O gráfico do módulo e da frequência angular usa escalas logarítmicas. Este par de gráficos é denominado **gráfico de Bode**.

O módulo é expresso em unidade de decibel (dB):

$$|G(j\omega)| \text{ em dB} = 20 \lg_{10}|G(j\omega)|$$

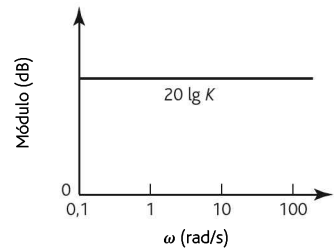
Portanto, por exemplo, um módulo de 20 dB significa que

$$20 = 20 \lg_{10}|G(j\omega)|$$

assim, $1 = \lg_{10}|G(j\omega)|$ e $10^1 = |G(j\omega)|$. Portanto, um módulo de 20 dB significa que o módulo é 10, e portanto a amplitude da saída é 10 vezes a da entrada. Um módulo de 40 dB tem um módulo de 100 e a amplitude de saída é 100 vezes a de entrada.

14.4.1 Gráfico de Bode para $G(s) = K$

Considere o gráfico de Bode para um sistema que tem a função de transferência $G(s) = K$, em que K é uma constante. A função resposta de frequência é $G(j\omega) = K$. O módulo $|G(j\omega)| = K$ e, em decibéis, $|G(j\omega)| = 20 \lg K$. O gráfico do módulo é uma linha de módulo constante; a variação de K meramente desloca a linha do módulo para cima ou para baixo em certo número de decibéis. A fase é zero. A Figura 14.3 mostra o gráfico de Bode.



14.4.2 Gráfico de Bode para $G(s) = 1/s$

Considere o gráfico de Bode para um sistema que tem uma função de transferência $G(s) = 1/s$. A função resposta de frequência de $G(j\omega)$ é $1/j\omega$. Multiplicando isto por j/j , temos $G(j\omega) = -j/\omega$. O módulo de $|G(j\omega)|$ é $1/\omega$. Em decibéis, isto vale $20 \lg(1/\omega) = -20 \lg \omega$. Quando $\omega = 1 \text{ rad/s}$, o módulo é zero. Quando $\omega = 10 \text{ rad/s}$, ele é -20 dB . Quando $\omega = 100 \text{ rad/s}$ ele é -40 dB . A cada aumento da frequência angular de 10 vezes, o módulo cai -20 dB . O gráfico do módulo é uma

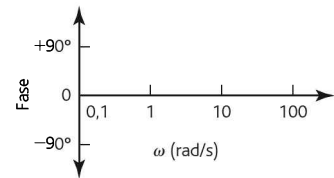


Figura 14.3 Gráfico de Bode para $G(s) = K$.

linha reta de inclinação -20 dB por década da frequência que passa por 0 dB a $\omega = 1$ rad/s. A fase deste sistema é dada por

$$\operatorname{tg} \phi = \frac{-1}{\omega} = -\infty$$

Portanto, $\phi = -90^\circ$ para todas as frequências. A Figura 14.4 mostra o gráfico de Bode.

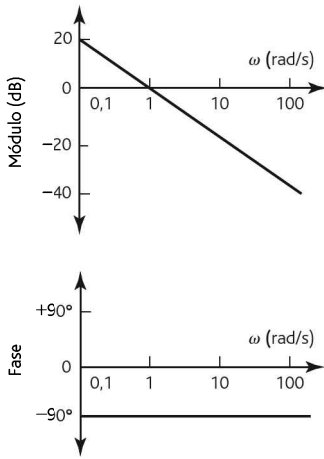


Figura 14.4 Gráfico de Bode para $G(s) = 1/s$.

14.4.3 Gráfico de Bode para um sistema de primeira ordem

Considere o gráfico de Bode para um sistema de primeira ordem em que a função de transferência é dada por

$$G(s) = \frac{1}{\tau s + 1}$$

A função resposta de frequência é

$$G(j\omega) = \frac{1}{j\omega\tau + 1}$$

O módulo (veja a Seção 14.2.1) é

$$|G(j\omega)| = \frac{1}{\sqrt{1 + \omega^2\tau^2}}$$

Em decibéis, ele vale

$$20 \operatorname{lg} \left(\frac{1}{\sqrt{1 + \omega^2\tau^2}} \right)$$

Quando $\omega \ll 1/\tau$, então $\omega^2\tau^2$ é desprezível em comparação com 1 e, assim, o módulo é $20 \operatorname{lg} 1 = 0$ dB. Portanto, em baixas frequências há uma linha reta no gráfico de módulo no valor constante de 0 dB. Para frequências maiores, quando $\omega \gg 1/\tau$, $\omega^2\tau^2$ é muito maior do que 1, sendo este valor desprezado. O módulo é então $20 \operatorname{lg}(1/\omega\tau)$, ou seja, $-20 \operatorname{lg} \omega\tau$. Esta é uma linha reta de inclinação -20 dB por década de frequência que intercepta a linha de 0 dB quando $\omega\tau = 1$, ou seja, quando $\omega = 1/\tau$. A Figura 14.5 mostra estas linhas para frequências baixas e altas com a interseção delas, ou **frequência de quina, de corte** ou **de canto**, em $\omega = 1/\tau$. As duas linhas retas são denominadas aproximações assintóticas do gráfico real. Este passa próximo da interseção das duas linhas. A diferença entre o gráfico real e a aproximação tem um máximo de 3 dB na frequência de corte.

A fase para um sistema de primeira ordem (veja a Seção 14.2.1) é dada por $\operatorname{tg} \phi = -\omega\tau$. Em baixas frequências, quando ω for menor do que aproximadamente $0,1/\tau$, a fase é praticamente 0° . Em altas frequências, quando ω for maior do que aproximadamente $10/\tau$, a fase é praticamente -90° . Entre estes dois extremos o ângulo de fase pode ser considerado dado aproximadamente pela linha reta no gráfico de Bode (Figura 14.5). O erro máximo ao se considerar essa linha reta é $5,5^\circ$.

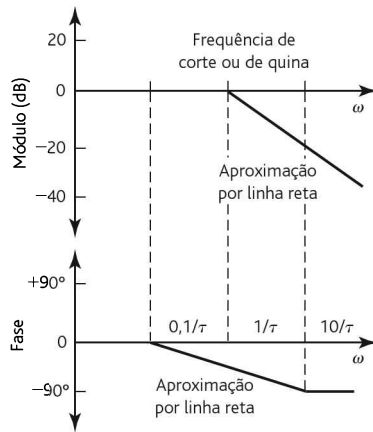


Figura 14.5 Gráfico de Bode para um sistema de primeira ordem.

Um exemplo deste sistema é um filtro RC (veja a Seção 13.2.2), ou seja, uma resistência R em série com uma capacitância C , sendo a saída a tensão no capacitor. Este sistema tem uma função de transferência de $1/(RCs + 1)$ e uma função resposta de frequência de $1/(j\omega\tau + 1)$ em que $\tau = RC$. O gráfico de Bode é mostrado na Figura 14.5.

14.4.4 Gráfico de Bode para um sistema de segunda ordem

Considere um sistema de segunda ordem com uma função de transferência de

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

A função resposta de frequência é obtida substituindo-se s por $j\omega$:

$$G(j\omega) = \frac{\omega_n^2}{-\omega^2 + j2\zeta\omega_n\omega + \omega_n^2}$$

O módulo é (veja a Seção 14.2.2)

$$|G(j\omega)| = \frac{1}{\sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left[2\zeta\left(\frac{\omega}{\omega_n}\right)\right]^2}}$$

Portanto, o módulo em decibéis é

$$\begin{aligned} & 20 \lg \frac{1}{\sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left[2\zeta\left(\frac{\omega}{\omega_n}\right)\right]^2}} \\ &= -20 \lg \sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left[2\zeta\left(\frac{\omega}{\omega_n}\right)\right]^2} \end{aligned}$$

Para $(\omega / \omega_n) \ll 1$, o módulo se aproxima de $-20 \log 1$ ou 0 dB, e para $(\omega / \omega_n) \gg 1$ o módulo se aproxima de $-20 \log (\omega / \omega_n)^2$. Portanto, quando ω aumenta em um fator de 10, o módulo aumenta em um fator de $-20 \log(100)^2$ ou -40 dB. Então, em baixas frequências o gráfico do módulo é uma linha reta em 0 dB, enquanto que em altas frequências ele é uma linha reta de -40 dB por década de frequência. A interseção destas duas linhas, ou seja, no ponto de corte, ocorre em $\omega = \omega_n$. O gráfico do módulo é representado aproximadamente por estas duas linhas assintóticas. Entretanto, o valor real depende da razão de amortecimento ζ . A Figura 14.6 mostra as duas linhas assintóticas e o gráfico real para alguns valores de razão de amortecimento.

A fase é dada por (veja a Seção 14.2.2)

$$\text{tg } \phi = -\frac{2\zeta\left(\frac{\omega}{\omega_n}\right)}{1 - \left(\frac{\omega}{\omega_n}\right)^2}$$

Para $(\omega / \omega_n) \ll 1$, como, por exemplo, $(\omega / \omega_n) = 0,2$, então, $\text{tg } \phi$ é aproximadamente 0 e $\phi = 0^\circ$. Para $(\omega / \omega_n) \gg 1$, como, por exemplo, $(\omega / \omega_n) = 5$, $\text{tg } \phi$ é aproximadamente $-\infty$ e $\phi = -180^\circ$. Quando $\omega = \omega_n$, temos $\text{tg } \phi = -\infty$ e $\phi = -90^\circ$. Uma aproximação razoável é dada pela linha reta em -90° para $\omega = \omega_n$ e o ponto de 0° para $(\omega / \omega_n) = 0,2$ e -180° para $(\omega / \omega_n) = 5$. A Figura 14.6 mostra o gráfico.

14.4.5 Construção de gráficos de Bode

Considere um sistema envolvendo elementos em série. A função de transferência do sistema completo é dada por (veja a Seção 13.4)

$$G(s) = G_1(s)G_2(s)G_3(s) \dots$$

Portanto, a função resposta de frequência para um sistema de dois elementos, quando s for substituído por $j\omega$, é

$$G(j\omega) = G_1(j\omega)G_2(j\omega)$$

Podemos escrever a função de transferência $G_1(j\omega)$ como um número complexo (veja a Seção 12.2)

$$x + jy = |G_1(j\omega)| (\cos \phi_1 + j \text{sen } \phi_1)$$

em que $|G_1(j\omega)|$ é o módulo e ϕ_1 é a fase da função resposta de frequência. De forma similar, podemos escrever $G_2(j\omega)$ como

$$|G_2(j\omega)| (\cos \phi_2 + j \text{sen } \phi_2)$$

Portanto,

$$\begin{aligned} G(j\omega) &= |G_1(j\omega)| (\cos \phi_1 + j \text{sen } \phi_1) \times |G_2(j\omega)| (\cos \phi_2 + j \text{sen } \phi_2) \\ &= |G_1(j\omega)| |G_2(j\omega)| [\cos \phi_1 \cos \phi_2 \\ &\quad + j(\text{sen } \phi_1 \cos \phi_2 + \cos \phi_1 \text{sen } \phi_2) + j^2 \text{sen } \phi_1 \text{sen } \phi_2] \end{aligned}$$

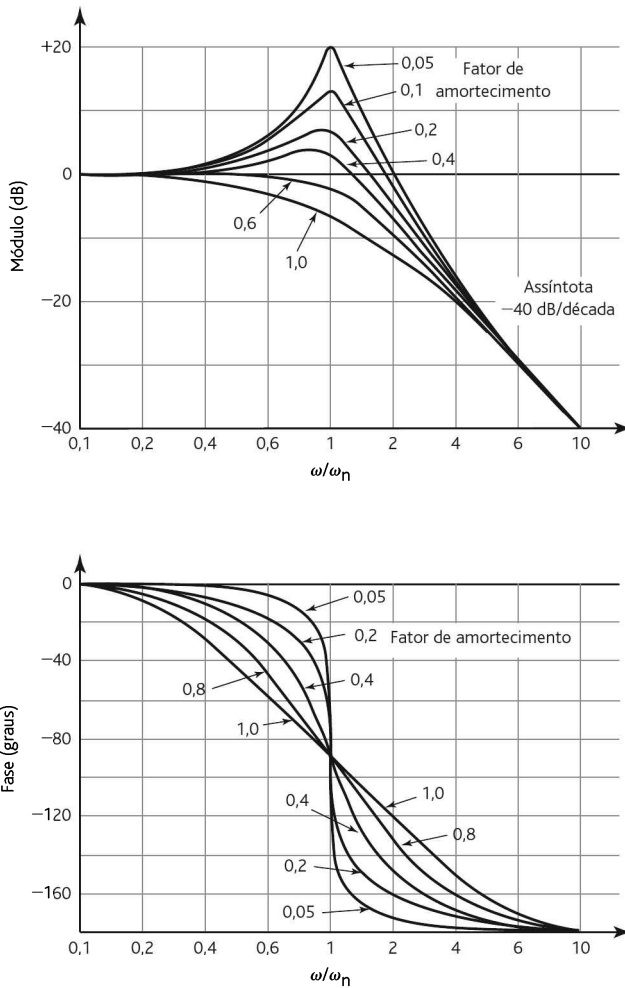


Figura 14.6 Gráfico de Bode para um sistema de segunda ordem.

Mas $j^2 = -1$ e, visto que $\cos \phi_1 \cos \phi_2 - \text{sen } \phi_1 \text{ sen } \phi_2 = \cos(\phi_1 + \phi_2)$ e $\text{sen } \phi_1 \cos \phi_2 + \cos \phi_1 \text{ sen } \phi_2 = \text{sen}(\phi_1 + \phi_2)$, então

$$G(j\omega) = |G_1(j\omega)| |G_2(j\omega)| [\cos(\phi_1 + \phi_2) + j \text{sen}(\phi_1 + \phi_2)]$$

A função resposta de frequência do sistema tem um módulo que é o produto dos módulos dos termos em separado e uma fase que é a soma das fases dos termos em separado, ou seja,

$$|G(j\omega)| = |G_1(j\omega)| |G_2(j\omega)| |G_3(j\omega)| \dots$$

$$\phi = \phi_1 + \phi_2 + \phi_3 + \dots$$

Agora, considerando o gráfico de Bode onde o logaritmo dos módulos é registrado no gráfico,

$$\lg |G(j\omega)| = \lg |G_1(j\omega)| + \lg |G_2(j\omega)| + \lg |G_3(j\omega)| + \dots$$

Assim, podemos obter o gráfico de Bode de um sistema somando os gráficos dos módulos dos termos constituintes. Da mesma forma, o gráfico de fase é obtido somando-se as fases dos termos constituintes.

Usando alguns termos básicos, os gráficos de Bode para uma ampla faixa de sistemas podem ser facilmente obtidos. Os termos básicos usados são:

- 1 $G(s) = K$ resulta no gráfico de Bode mostrado na Figura 14.3.
- 2 $G(s) = 1/s$ resulta no gráfico de Bode mostrado na Figura 14.4.
- 3 $G(s) = s$ resulta no gráfico de Bode que é uma imagem de espelho da Figura 14.4. $|G(j\omega)| = 20$ dB por década de frequência, passando por 0 dB em $\omega = 1$ rad/s. ϕ é constante para 90° .
- 4 $G(s) = 1/(\tau s + 1)$ resulta no gráfico de Bode mostrado na Figura 14.5.
- 5 $G(s) = \tau s + 1$ resulta no gráfico de Bode que é uma imagem de espelho da Figura 14.5. Para o gráfico de módulo, o ponto de corte ocorre em $1/\tau$ com a linha anterior a este ponto situada em 0 dB e após ele com uma inclinação de 20 dB por década de frequência. A fase é zero para $0,1/\tau$ e aumenta para $+90^\circ$ em $10/\tau$.
- 6 $G(s) = \omega_n^2/(s^2 + 2\zeta\omega_n s + \omega_n^2)$ resulta no gráfico de Bode mostrado na Figura 14.6.
- 7 $G(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2)/\omega_n^2$ resulta no gráfico de Bode que é uma imagem de espelho da Figura 14.6.

Para ilustrar isto, considere o desenho das assíntotas do gráfico de Bode para um sistema que tem a função de transferência

$$G(s) = \frac{10}{2s + 1}$$

A função de transferência é constituída de dois termos, um com uma função de transferência de 10 e outro com a função de transferência $1/(2s + 1)$. Os gráficos de Bode podem ser desenhados para cada uma destas funções e somados para se obter o gráfico desejado. O gráfico de Bode para a função de transferência 10 tem a forma mostrada na Figura 14.3 com $K = 10$, e para $1/(2s + 1)$, como o gráfico mostrado na Figura 14.5 com $t = 2$. O resultado é mostrado na Figura 14.7.

Como outro exemplo, considere o desenho das assíntotas do gráfico de Bode para um sistema que tem a função de transferência

$$G(s) = \frac{2,5}{s(s^2 + 3s + 25)}$$

A função de transferência é constituída de três termos: um é a função de transferência de 0,1, o segundo é $1/s$ e o terceiro é $25/(s^2 + 3s + 25)$. A função de transferência 0,1 resulta num gráfico de Bode como o da Figura 14.3 com $K = 0,1$. A função de transferência $1/s$ resulta num gráfico de Bode como o da Figura 14.4. A função de

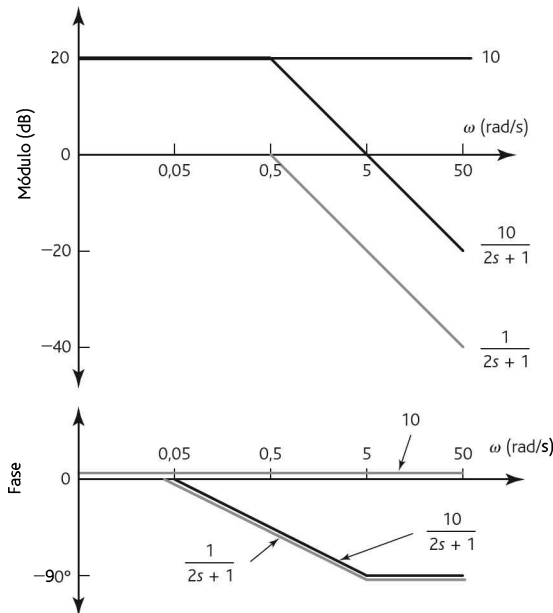


Figura 14.7 Construção de um diagrama de Bode.

transferência $25/(s^2 + 3s + 25)$ pode ser representada como $\omega_n^2/(s^2 + 2\zeta\omega_n s + \omega_n^2)$ com $\omega_n = 5$ rad/s e $\zeta = 0,3$. O ponto de corte ocorre quando $\omega = \omega_n = 5$ rad/s. A assíntota para a fase passa em -90° no ponto de corte, em 0° quando temos $(\omega/\omega_n) = 0,2$ e em -180° quando $(\omega/\omega_n) = 5$. A Figura 14.8 mostra o gráfico resultante.

Este método de obtenção do gráfico de Bode a partir dos termos constituintes, usando as aproximações de linha reta, foi amplamente usado, mas atualmente é pouco aplicado em função do uso crescente de computadores.

14.4.6 Identificação de sistemas

Se determinarmos experimentalmente o diagrama de Bode para um sistema considerando sua resposta a uma entrada senoidal, então podemos obter a função de transferência para o sistema. Basicamente desenhamos as assíntotas no gráfico de Bode do módulo e consideramos seus gradientes. A curva do ângulo de fase é usada para verificar o resultado obtido a partir da análise do módulo.

- 1 Se o gradiente para baixas frequências antes da primeira frequência de corte for zero, não haverá um termo em s ou $1/s$ na função de transferência.
- 2 Se o gradiente inicial para baixas frequências for -20 dB/década, então a função de transferência tem uma componente $1/s$.
- 3 Se o gradiente se torna mais negativo na frequência de corte em 20 dB/década, há um termo $(1 + s/\omega_c)$ no denominador da função de transferência, com ω_c

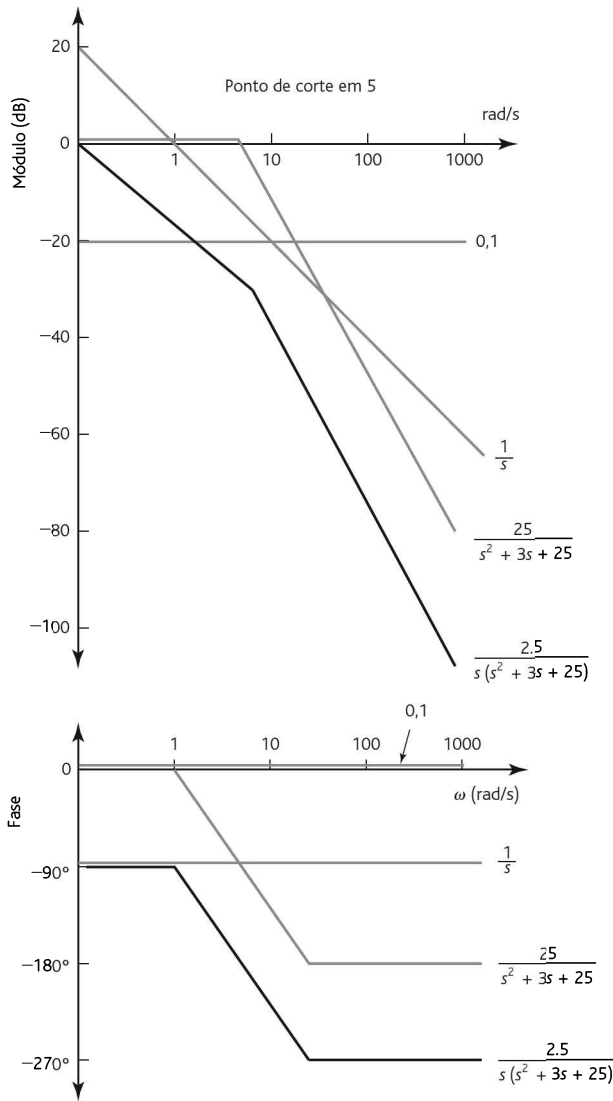


Figura 14.8 Construção de um gráfico de Bode.

sendo a frequência de corte na qual a mudança ocorre. Estes termos podem ocorrer em mais de uma frequência de corte.

- 4 Se o gradiente se torna mais positivo na frequência de corte em 20 dB/década, há um termo $(1 + s/\omega_c)$ no numerador da função de transferência, com ω_c sendo a frequência de corte na qual a mudança ocorre. Estes termos podem ocorrer em mais de uma frequência de corte.
- 5 Se o gradiente se torna mais negativo na frequência de corte em 40 dB/década, há um termo $(s^2/\omega_c^2 + 2\zeta s/\omega_c + 1)$ no denominador da função de transferên-

cia. A razão de amortecimento ζ pode ser determinada considerando-se o detalhe do gráfico de Bode na frequência de corte, como na Figura 14.6.

- 6 Se o gradiente se torna mais positivo na frequência de corte em 40 dB/década, há um termo $(s^2/\omega_c^2 + 2\zeta s/\omega_c + 1)$ no numerador da função de transferência. A razão de amortecimento ζ pode ser determinada considerando-se o detalhe do gráfico de Bode na frequência de corte, como na Figura 14.6.
- 7 Se o gradiente para baixas frequências não for zero, o termo K no numerador da função de transferência pode ser determinado considerando-se o valor da assíntota para baixa frequência. Em frequências baixas, muitos termos na função de transferência podem ser desprezados e o ganho em dB é aproximado para $20 \lg(K/\omega^2)$. Portanto, para $\omega = 1$ o ganho em dB se aproxima de $20 \lg K$.

Como ilustração disto, considere o gráfico de Bode do módulo mostrado na Figura 14.9. O gradiente inicial é 0 e, assim, não há o termo $1/s$ ou s na função de transferência. O ganho inicial é 20 e, assim, $20 = 20 \lg K$ e $K = 10$. O gradiente muda -20 dB/década para uma frequência de 10 rad/s. A função de transferência é então $10/(1 + 0,1s)$.

Como mais uma ilustração, considere a Figura 14.10. Há uma inclinação inicial de -20 dB/década e, assim, um termo $1/s$. Na frequência de corte 1,0 rad/s, há uma variação de -20 dB/década no gradiente e, portanto, um termo $1/(1 + s/1)$. Na frequência de corte 10 rad/s há mais uma variação de -20 dB/década no gradiente e, portanto, um termo $1/(1 + s/10)$. Em $\omega = 1$, o módulo é 6 dB e, assim, $6 = 20 \lg K$ e $K = 10^{6/20} = 2,0$. A função de transferência é, portanto, $2,0/s(1 + s)$ ($1 + 0,1s$).

Como mais uma ilustração, a Figura 14.11 mostra um gráfico de Bode com um gradiente inicial zero que varia -40 dB/década em 10 rad/s. O módulo inicial é 10 dB e, assim, $10 = 20 \lg K$ e $K = 10^{0,5} = 3,2$. A variação de -40 dB/década em 10 rad/s significa que há um termo $(s^2/10^2 + 2\zeta s/10 + 1)$ no denominador. A função de transferência é, portanto, $3,2/(0,01s^2 + 0,2\zeta s + 1)$. O fator de amortecimento pode ser obtido comparando-se o gráfico de Bode nas frequências de corte com a Figura 14.6. Ele aumenta cerca de 6 dB acima da frequência de corte, e isto corresponde a um fator de amortecimento de cerca de 0,2. A função de transferência é, portanto, $3,2/(0,01s^2 + 0,04s + 1)$.

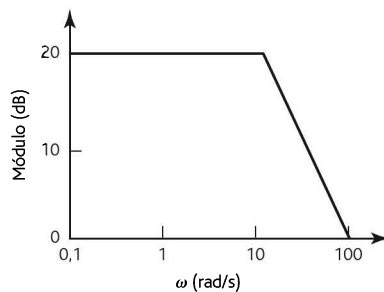


Figura 14.9 Gráfico de Bode.

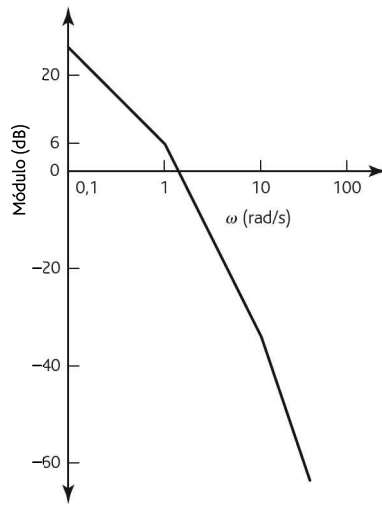


Figura 14.10 Gráfico de Bode.

14.5 Especificação de desempenho

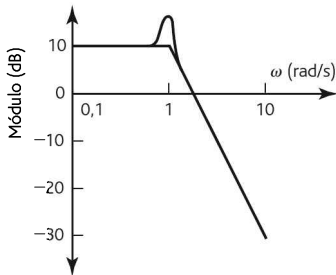


Figura 14.11 Gráfico de Bode.

Os termos usados para descrever o desempenho de um sistema quando submetido a uma entrada senoidal são o pico de ressonância e a largura de banda. O **pico de ressonância** M_p é definido como sendo o valor máximo do módulo (Figura 14.12). Um valor grande do pico de ressonância corresponde a um valor grande do *overshoot* máximo de um sistema. Para um sistema de segunda ordem, ele pode ser diretamente relacionado à razão de amortecimento comparando-se a resposta com o gráfico de Bode da Figura 14.6, em que uma razão de amortecimento baixa corresponde a um pico de ressonância alto.

A **largura de banda** é definida como a faixa de frequência entre os módulos que não caem abaixo de -3 dB, que são as frequências denominadas frequências de corte. Com o módulo expresso em unidades de decibel (dB),

$$|G(j\omega)| \text{ em dB} = 20 \lg_{10} |G(j\omega)|$$

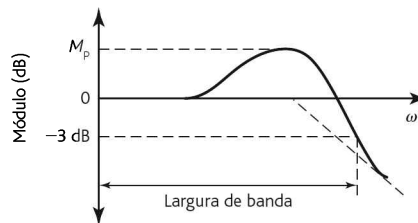


Figura 14.12 Especificações de desempenho.

e portanto

$$-3 = 20 \log_{10} |G(j\omega)|$$

e $|G(j\omega)| = 0,707$, que corresponde a uma queda da amplitude para 0,707 do seu valor inicial. Visto que a potência de uma forma de onda senoidal é o quadrado de sua amplitude, então a potência cai para $0,707^2 = 0,5$ do seu valor inicial. Portanto, o ponto de corte em -3 dB é o valor em decibel no qual a potência do sinal de entrada é atenuado para a metade do valor de entrada. Para o sistema representado pelo gráfico de Bode na Figura 14.12, a largura de banda está entre zero e a frequência na qual o módulo cai -3 dB. Este é o comportamento típico de sistemas de medição; eles normalmente não apresentam atenuação em baixas frequências, e o módulo diminui apenas em altas frequências.

Como uma ilustração, para o exemplo descrito na Seção 13.2.2, item 1, o módulo de um sistema (um circuito elétrico com um resistor em série com um capacitor, sendo a saída obtida deste) com a função de transferência

$$G(s) = \frac{1}{RCs + 1}$$

foi determinado como sendo

$$|G(j\omega)| = \frac{1}{\sqrt{1 + \omega^2(RC)^2}}$$

Para esta razão de módulo ser 0,707, a frequência de corte ω_c tem que ser dada por

$$0,707 = \frac{1}{\sqrt{1 + \omega_c^2(RC)^2}}$$

$$1 + \omega_c^2(RC)^2 = (1/0,707)^2 = 2$$

Portanto, $\omega_c = 1/RC$. Este circuito é denominado filtro passa-baixas, visto que as frequências baixas passam para a saída com pequena atenuação e as altas são atenuadas.

14.6 Estabilidade

Quando um sistema tem uma entrada senoidal, a saída dele é senoidal com a mesma frequência angular, porém com amplitude e fase de saída diferentes. Considere um sistema em malha fechada com realimentação negativa (Figura 14.13) e sem sinal de entrada. Suponha, de certa forma, que temos um pulso senoidal retificado em meia onda como o sinal de erro no sistema e que ele passa para a saída e é realimentado através de um comparador que mantém a amplitude inalterada mas insere um atraso de meio ciclo, ou seja, uma mudança de fase de 180° , como mostrado na figura. Quando este sinal é subtraído do sinal de entrada, o sinal de erro resultante continua com o pulso retificado em meia onda. Em seguida, este pulso retorna pela malha de realimentação e mais uma vez chega a tempo para prolongar o sinal. Assim, temos uma oscilação autossustentada.

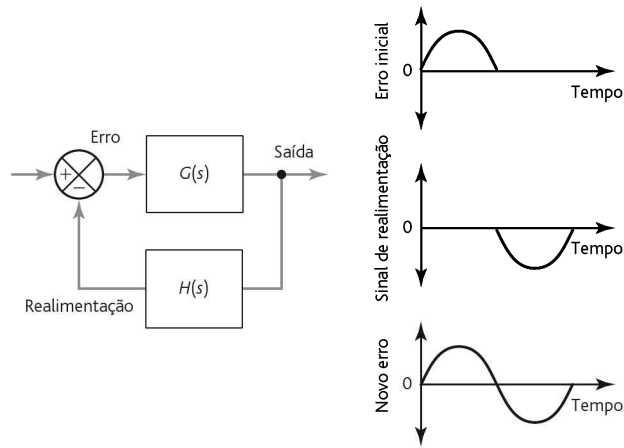


Figura 14.13 Oscilações autossustentadas.

Para oscilações autossustentadas ocorrerem, temos que ter um sistema que tem uma função resposta de frequência com um módulo de 1 e uma fase de -180° . O sistema pelo qual o sinal passa é $G(s)$ em série com $H(s)$. Se o módulo for menor do que 1, então cada pulso de meia onda subsequente é menor em amplitude, e portanto a oscilação se extingue. Se o módulo for maior do que 1, então cada pulso sucessivo é maior do que o anterior e, assim, a onda aumenta e o sistema é instável.

- 1 Um sistema de controle oscila com uma amplitude constante se o módulo resultante do sistema $G(s)$ em série com $H(s)$ for 1 e a fase for -180° .
- 2 Um sistema de controle oscila com uma oscilação decrescente se o módulo resultante do sistema $G(s)$ em série com $H(s)$ for menor do que 1 e a fase for -180° .
- 3 Um sistema de controle oscila com uma oscilação crescente se o módulo resultante do sistema $G(s)$ em série com $H(s)$ for maior do que 1 e a fase for -180° .

Um bom sistema de controle estável geralmente necessita que a amplitude de $G(s)H(s)$ seja significativamente menor do que 1. Tipicamente é usado um valor entre 0,4 e 0,5. Além disto, o ângulo de fase tem que estar entre cerca de -115° e -125° . Estes valores produzem um sistema de controle ligeiramente subamortecido que fornece, com uma entrada em degrau, um *overshoot* de 20 a 30% com uma razão de amortecimento de 3 a 1 (veja a Seção 12.5 para uma explicação destes termos).

O que importa em um sistema de controle é o quanto ele é estável e, portanto, que não oscile como resultado de algum distúrbio pequeno. O termo **margem de ganho** é usado para o fator pelo qual a razão de módulo é multiplicada quando a fase é -180° para fazer com que ela tenha o valor 1 e, assim, estar na vizinhança da instabilidade. O termo **margem de fase** é usado para o número de graus pelo qual o ângulo de fase é numericamente menor do que -180° quando o módulo for 1. Dentro destas regras, uma margem de ganho entre 2 e 2,5 e uma margem de fase entre 45° e 65° significam um sistema de controle estável.

Resumo

Podemos converter uma equação no domínio s para o **domínio da frequência** substituindo s por $j\omega$. A **função resposta de frequência** é a função de transferência quando transformada para o domínio da frequência.

A resposta de frequência de um sistema é o conjunto de valores do módulo $|G(j\omega)|$ e do ângulo de fase ϕ que ocorre quando um sinal senoidal de entrada varia ao longo de uma faixa de frequências. Isto pode ser expresso como dois gráficos, um do módulo $|G(j\omega)|$ em função da frequência angular ω e o outro da fase ϕ em função de ω . O módulo e a frequência angular são registrados num gráfico que usa escalas logarítmicas. Este par de gráficos é conhecido com **gráficos de Bode** (ou diagrama de Bode).

Podemos obter o gráfico de Bode de um sistema somando os gráficos de Bode dos módulos dos termos constituintes. Da mesma forma, o gráfico da fase é obtido somando-se as fases dos termos constituintes.

O **pico de ressonância** M_p é o valor máximo do módulo. A largura de banda é a faixa de frequência em que o módulo não cai abaixo de -3 dB, e as frequências em que isto ocorre são denominadas frequências de corte.

Para que oscilações autossustentadas ocorram com um sistema realimentado, ou seja, com o sistema na vizinhança da **instabilidade**, o sistema tem que ter uma função resposta de frequência com um módulo de 1 e uma fase de -180° . A **margem de ganho** é o fator pelo qual a razão de módulo tem que ser multiplicada quando a fase for -180° para fazer com que ela tenha valor 1 e se situe na vizinhança da instabilidade. A **margem de fase** é o número de graus pelo qual o ângulo de fase é numericamente menor do que -180° quando o módulo é 1.

Problemas

- 14.1 Quais são os módulos e as fases dos sistemas com as seguintes funções de transferência?

$$(a) \frac{5}{s+2}, \quad (b) \frac{2}{s(s+1)}, \quad (c) \frac{1}{(2s+1)(s^2+s+1)}$$

- 14.2 Qual deve ser a resposta de estado estacionário de um sistema com a função de transferência $1/(s+2)$ quando submetido à entrada senoidal $3 \sin(5t + 30^\circ)$?
- 14.3 Qual deve ser a resposta de estado estacionário de um sistema com a função de transferência $5/(s^2 + 3s + 10)$ quando submetido à entrada senoidal $2 \sin(2t + 70^\circ)$?
- 14.4 Determine os valores dos módulos e da fase para as frequências angulares de (i) 0 rad/s, (ii) 1 rad/s, (iii) 2 rad/s, (iv) ∞ rad/s para o sistema com as funções de transferência (a) $1/[s(2s+1)]$, (b) $1/(3s+1)$.
- 14.5 Desenhe as assíntotas do gráfico de Bode para os sistemas com as seguintes funções de transferência: (a) $10/[s(0,1s+1)]$, (b) $1/[(2s+1)(0,5s+1)]$.

14.6 Obtenha as funções de transferência dos sistemas dados pelos gráficos de Bode na Figura 14.14.

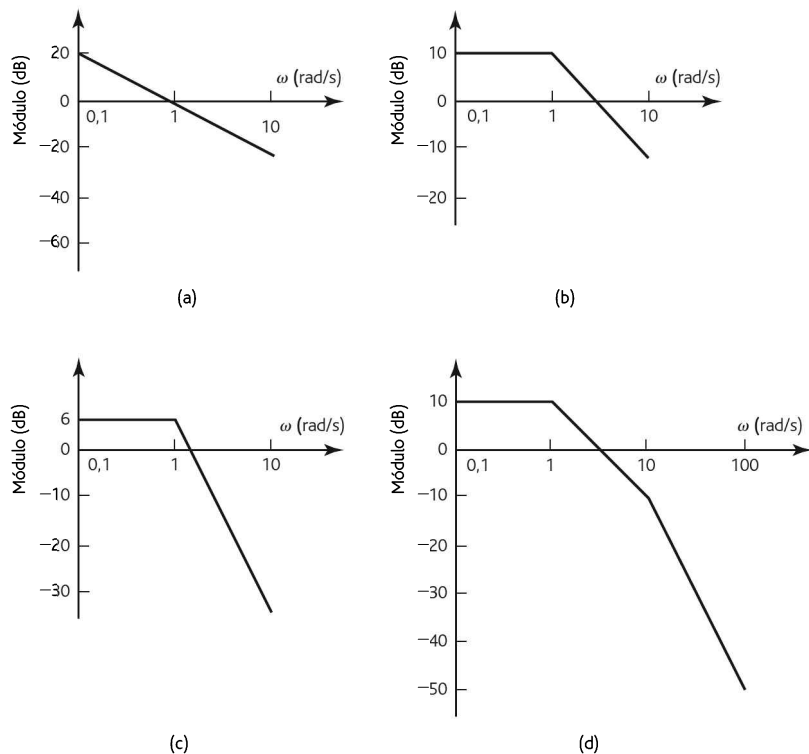


Figura 14.14 Problema 14.6.

15

Controladores em Malha Fechada

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Explicar o termo erro de estado estacionário.
- Explicar a operação do modo de controle *on/off*.
- Prever o comportamento de sistemas com os controles proporcional (P), integral (I), derivativo (D), proporcional mais integral (PI), proporcional mais derivado (PD) e PID.
- Descrever como funcionam os controladores digitais.
- Explicar como os controladores podem ser sintonizados.

15.1 Processos de controle contínuo e discreto

O **controle em malha aberta** é muitas vezes apenas um controle do tipo *on/off* (liga/desliga), como, por exemplo, um aquecedor elétrico que é ligado ou desligado para aquecer um ambiente. No caso dos **sistemas de controle em malha fechada**, é usado um controlador para comparar continuamente a saída de um sistema com a condição desejada e converter o erro em uma ação de controle com a finalidade de reduzir o erro. O erro pode aumentar como resultado de alguma alteração nas condições do sistema controlado ou por causa de uma mudança no valor desejado, ou seja, um degrau na entrada muda o valor desejado para o sistema de um nível para outro. Neste capítulo vamos nos ater às formas nas quais os controladores podem reagir aos sinais de erro, ou seja, os **modos de controle**, como são denominados, que ocorrem em processos contínuos. Estes controladores podem, por exemplo, ser sistemas pneumáticos ou com amplificadores operacionais. Entretanto, os sistemas de computador estão substituindo rapidamente muitos deles. O termo **controle digital direto** é usado quando o computador está em malha fechada e exercendo o controle desta forma. Este capítulo aborda o controle em malha fechada.

Muitos processos não envolvem apenas o controle de alguma variável, como, por exemplo, a temperatura, em um valor desejado, mas também uma sequência de operações. Um exemplo é uma máquina doméstica de lavar roupa (veja a Seção 1.5.5) em que várias ações têm que ser executadas em uma determinada sequência. Outro exemplo é a fabricação de um produto que envolve a montagem de um conjunto de peças em uma sequência específica por algum sistema controlado. A sequência de operações pode ser **baseada em um sinal de clock** ou **em eventos**, ou ainda em uma combinação desses dois. No caso de sistemas baseados em sinal de clock, as ações são executadas em momentos específicos; nos sistemas baseados em eventos, as ações são executadas quando há uma realimentação indicando que um evento em particular ocorreu.

Em muitos processos há uma mistura de controle discreto com contínuo. Por exemplo, em uma máquina doméstica de lavar roupa há um controle sequencial para as diversas fases do ciclo de lavagem com controle por realimentação de temperatura e do nível da água.

15.2 Terminologia

Os termos a seguir são usuais nas discussões sobre controladores em malha fechada.

15.2.1 Atraso (*lag*)

Em qualquer sistema de controle existem atrasos (*lags*). Portanto, por exemplo, uma variação na condição controlada não produz imediatamente uma resposta de correção a partir do sistema de controle. Isto ocorre porque é necessário um tempo para que o sistema produza as respostas necessárias. Por exemplo, no controle de temperatura de um ambiente por meio de um sistema central de aquecimento, ocorrerá um atraso entre a diminuição da temperatura, identificada pelo controlador, e a resposta dele para ligar o aquecedor. Este não é o único atraso. Mesmo quando o sistema de controle responde, ainda há um atraso na resposta da temperatura do ambiente conforme o tempo necessário para que o calor seja transferido do aquecedor para o ar dentro do ambiente.

15.2.2 Erro de estado estacionário

Um sistema de controle em malha fechada usa a medida da saída do sistema, que é comparada com o valor desejado, gerando um sinal de erro. Um controlador pode receber um sinal de erro como resultado de uma mudança no valor da variável controlada ou no valor desejado (*set point*). Por exemplo, a entrada do sistema pode ser uma rampa com a finalidade de que a variável controlada varie de forma constante com o tempo. Quando uma mudança ocorre, existem naturalmente alguns efeitos transientes; entretanto, eles se extinguem com o tempo. O termo **erro de estado estacionário** é usado para a diferença entre o valor desejado de entrada e a saída após todos os transientes terem cessado. É, portanto, uma medida da precisão do sistema de controle na busca do valor desejado. Sempre que houver um erro, é porque a saída *não* está no valor desejado.

Considere um sistema de controle que tem uma realimentação unitária (Figura 15.1). Quando houver uma entrada de referência $R(s)$, existirá uma saída $X(s)$. O

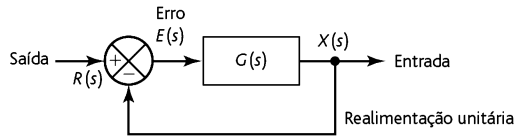


Figura 15.1 Realimentação unitária.

sinal de realimentação é $X(s)$ e, assim, o sinal de erro será $E(s) = R(s) - X(s)$. Se $G(s)$ for o caminho direto na função de transferência, então, para um sistema com realimentação unitária, teremos:

$$\frac{X(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} = \frac{G(s)}{1 + G(s)}$$

Portanto,

$$E(s) = R(s) - X(s) = R(s) - \frac{G(s)R(s)}{1 + G(s)} = \frac{1}{1 + G(s)}R(s)$$

Então, o erro depende de $G(s)$.

Para determinar o erro de estado estacionário podemos determinar o erro e como uma função do tempo e , em seguida, determinar o valor do erro quando todos os transientes tiverem acabado, ou seja, quando o tempo t tende para um valor infinito. Em vez de determinarmos o inverso de $E(s)$ e em seguida o seu valor quando $t \rightarrow \infty$, existe um método mais simples que usa o **teorema do valor final** (veja o Apêndice A); esta operação evolva a determinação de $sE(s)$ conforme s tende para o valor zero:

$$e_{SS} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s)$$

Para ilustrar isto, considere um sistema com realimentação unitária com a função de transferência no caminho direto de $k/(\tau s + 1)$ e submetido a uma entrada em degrau de $1/s$:

$$e_{SS} = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} \left[s \frac{1}{1 + k/(\tau s + 1)} \frac{1}{s} \right] = \frac{1}{1 + k}$$

Portanto, há um erro de estado estacionário; a saída do sistema nunca alcança o valor desejado (*set point*). Aumentando-se o ganho k do sistema, o erro de estado estacionário pode ser reduzido.

O caminho direto pode ser controlado com um ganho de k e um sistema com uma função de transferência $1/(\tau s + 1)$. Este ganho de controlador é denominado controlador proporcional. O erro de estado estacionário neste caso é normalmente denominado *offset*. Ele pode ser minimizado aumentando-se o ganho.

Entretanto, se o sistema com realimentação unitária tiver uma função de transferência no caminho direto de $k/s(\tau s + 1)$ e for submetido a uma entrada em degrau, então o erro de estado estacionário será

$$e_{SS} = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} \left[s \frac{1}{1 + k/s(\tau s + 1)} \frac{1}{s} \right] = 0$$

Este sistema não tem erro de estado estacionário. Neste caso, o caminho direto pode ser um controlador com um ganho de k/s e um sistema com uma função de transferência $1/(\tau s + 1)$. Este ganho de controlador é denominado controlador integral e não gera *offset*. Portanto, combinando-se um controlador proporcional com um integral é possível eliminar o *offset*. Acrescentar um controlador derivativo permite ao controlador responder mais rapidamente às variações.

15.2.3 Modos de controle

Existem algumas formas em que uma unidade de controle pode reagir a um sinal de erro e gerar uma saída para os elementos de correção:

- 1 O *modo on/off*, em que o controlador é essencialmente apenas uma chave que é ativada pelo sinal de erro e fornece apenas um sinal de correção do tipo *on/off*.
- 2 O *modo proporcional* (P), que produz uma ação de controle que é proporcional ao erro. Portanto, quanto maior o erro, maior o sinal de correção. Conforme o erro é reduzido, a intensidade da correção é reduzida e o processo de correção vai diminuindo.
- 3 O *modo derivativo* (D), que produz uma ação de controle que é proporcional à taxa na qual o erro varia. Quando há uma variação rápida no sinal de erro, o controlador gera um sinal de correção grande; quando há uma variação gradual, é produzido um sinal de erro pequeno. O controle derivativo pode ser considerado uma forma de controle antecipativo no qual a taxa de variação do erro é medida, um erro grande que está para chegar é antecipado e uma correção é aplicada antes da chegada dele. Este controle não é usado sozinho, mas em conjunto com o controle proporcional e, frequentemente, com o controle integral.
- 4 O *modo integral* (I), que produz uma ação de controle proporcional à integral do erro com o tempo. Portanto, um sinal de erro constante produz um sinal de correção crescente. A correção continuará aumentando enquanto o sinal de erro persistir. O controlador integral pode ser considerado como estando “olhando para trás”, somando todos os erros e respondendo às variações que ocorreram.
- 5 *Combinações de modos*: são os modos proporcional mais derivativo (PD), proporcional mais integral (PI) e proporcional mais integral mais derivativo (PID). O termo **controlador de três termos** é usado para o controle PID.

Estes cinco modos de controle são discutidos nas seções a seguir deste capítulo. Um controlador pode realizar estes modos de controle por meio de circuitos pneumáticos, circuitos de eletrônica analógica que envolvem amplificadores operacionais ou por meio da programação de um microprocessador ou um computador.

15.3 Modo *on/off*

Um exemplo de um **modo *on/off*** de controle é um termostato bimetálico (Figura 2.46) que pode ser usado em um sistema de controle de temperatura simples. Trata-se apenas de uma chave que é ligada (*on*) ou desligada (*off*) de acordo com a temperatura. Se a temperatura do ambiente estiver acima da desejada, o bimetálico está na posição *off* e o aquecedor está desligado. Se a temperatura da sala cair abaixo da desejada, o bimetálico se move para a posição *on*, ligando o aquecedor. Neste caso, o controlador pode estar apenas em duas posições, *on* ou *off*, conforme indicado pela Figura 15.2(a).

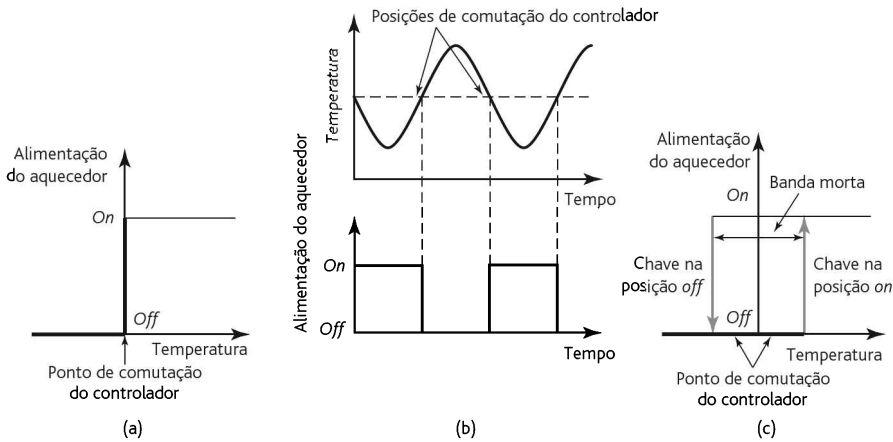


Figura 15.2 Controle on/off.

No modo de controle *on/off*, a ação de controle é descontínua. Uma consequência disto são as oscilações que ocorrem na variável controlada em torno da condição desejada. Isto ocorre por causa dos atrasos no sistema de controle e na resposta do processo. Por exemplo, no caso do controle de temperatura de um sistema central de aquecimento residencial, quando a temperatura do ambiente cai abaixo do nível desejado, o tempo decorrido antes do sistema de controle responder e ligar o aquecedor pode ser muito pequeno em comparação com o tempo decorrido antes do aquecedor começar a produzir um efeito na temperatura do ambiente. Neste meio tempo, a temperatura cai ainda mais. A situação inversa ocorre quando a temperatura sobe acima da desejada. Visto o tempo decorrido antes do sistema de controle reagir desligando o aquecedor somado ao tempo levado para esfriar e parar de aquecer o ambiente, a temperatura do ambiente vai além do valor desejado. O resultado é que a temperatura do ambiente oscila acima e abaixo da desejada (Figura 15.2(b)).

Com o sistema *on/off* descrito acima, existe o problema de que quando a temperatura do ambiente estiver em torno do valor ajustado no termostato, pode ocorrer uma operação de liga/desliga continuamente, em reação a pequenas variações na temperatura. Isto pode ser evitado se, em vez de se ter apenas um valor de temperatura na qual o controlador liga ou desliga o aquecedor, forem usados dois valores e o aquecedor for ligado na temperatura superior e desligado na inferior (Figura 15.2(c)). O termo **banda morta** é usado para os valores entre os valores *on* e *off*. Uma banda morta grande resulta em grandes flutuações da temperatura em torno do valor ajustado; uma banda morta pequena resulta em um aumento na frequência de comutação. O elemento bimetálico mostrado na Figura 2.46 tem um ímã permanente no contato da chave; ele tem o efeito de produzir a banda morta.

A ação de controle *on/off* tende a ser usada onde as variações acontecem muito lentamente, ou seja, em um processo com uma grande capacitância. Portanto, no caso do aquecimento de um ambiente, o efeito na temperatura do ambiente com a comutação do aquecedor é uma variação apenas lenta. O resultado disto é uma

oscilação com um período de tempo longo. O controle *on/off* não é preciso, mas ele envolve dispositivos simples, resultando em um sistema muito barato. Este tipo de controle não se restringe a chaves mecânicas como um bimetálico ou um relé; uma comutação rápida pode ser obtida com o uso de circuitos com tiristor (veja a Seção 9.3.2); este circuito pode ser usado no controle de velocidade de um motor e amplificadores operacionais.

15.4 Modo proporcional

No caso do método de controle *on/off*, a saída do controlador é um sinal *on* ou *off*, independentemente da amplitude do erro. No caso do **modo proporcional**, a amplitude da saída do controlador é proporcional à amplitude do erro: quanto maior o erro, maior a saída do controlador. Isto significa que o elemento de correção do sistema de controle, ou seja, a válvula, receberá um sinal que é proporcional à amplitude da correção necessária. Portanto,

$$\text{saída do controlador} = K_p e$$

em que e é o erro e K_p é uma constante. Aplicando a transformada de Laplace,

$$\text{saída do controlador (s)} = K_p E(s)$$

e, assim, K_p é a função de transferência do controlador.

15.4.1 Controlador proporcional eletrônico

Um amplificador operacional somador com um inversor pode ser usado como um controlador proporcional (Figura 15.3). Em um amplificador somador temos (veja a Seção 3.2.3)

$$V_{out} = -R_f \left(\frac{V_0}{R_2} + \frac{V_c}{R_1} \right)$$

A entrada do amplificador somador através de R_2 é V_0 , o valor de tensão de erro zero, ou seja, o valor ajustado (*set point*), e a entrada através de R_1 é o sinal de erro

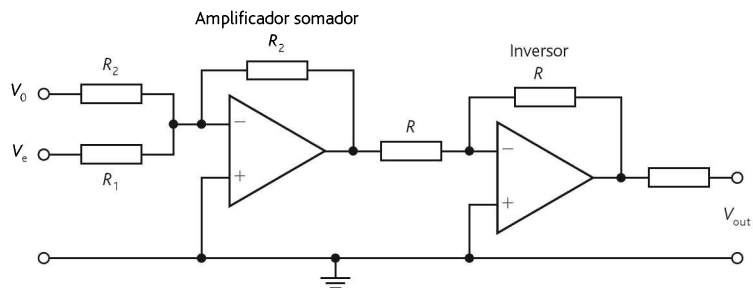


Figura 15.3 Controlador proporcional.

V_e . Porém, quando o resistor de realimentação for $R_f = R_2$, então a equação se torna

$$V_{out} = -\frac{R_2}{R_1} V_e - V_0$$

Se a saída do amplificador somador passar por um inversor, ou seja, um amplificador operacional com uma resistência de realimentação igual à resistência de entrada, então

$$V_{out} = \frac{R_2}{R_1} V_e + V_0$$

$$V_{out} = K_p V_e + V_0$$

em que K_p é a constante de proporcionalidade. O resultado é um controlador proporcional.

Como ilustração, a Figura 15.4 mostra um exemplo de um sistema de controle proporcional usado no controle da temperatura de um líquido em um recipiente à medida que o líquido é bombeado para dentro dele.

15.4.2 Resposta do sistema

O controle proporcional possui um elemento de ganho com função de transferência K_p em série com o elemento no caminho direto $G(s)$ (Figura 15.5). Portanto, o erro é

$$E(s) = \frac{K_p G(s)}{1 + K_p G(s)} R(s)$$

e, assim, para uma entrada em degrau, o erro de estado estacionário é

$$e_{SS} = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} \left[s \frac{1}{1 + 1/K_p G(s)} \frac{1}{s} \right]$$

Este resultado terá um valor finito e, desta forma, sempre haverá um erro de estado estacionário. Valores baixos de K_p resultam em grandes erros de estado estacionário, porém, respostas estáveis. Valores altos de K_p resultam em erros de estado estacionário menores, porém com uma grande tendência a instabilidade.

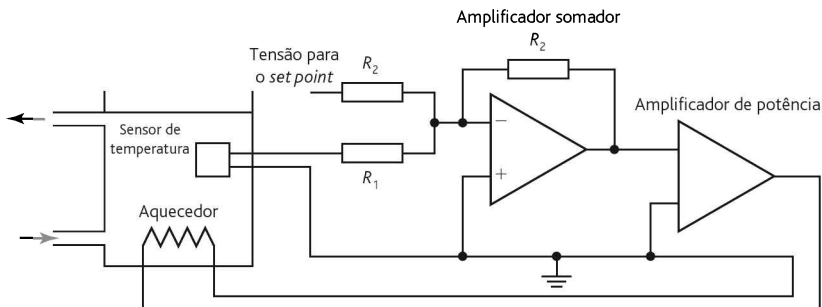


Figura 15.4 Controlador proporcional para controle de temperatura.

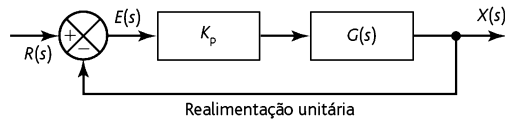


Figura 15.5 Sistema com controle proporcional.

15.5 Controle derivativo

No caso do controle em **modo derivativo**, a saída do controlador é proporcional à taxa de variação com o tempo do sinal de erro. Isto pode ser representado pela equação

$$\text{saída do controlador} = K_D \frac{de}{dt}$$

K_D é a constante de proporcionalidade. A função de transferência é obtida aplicando-se a transformada de Laplace, portanto

$$\text{saída do controlador } (s) = K_D s E(s)$$

Portanto, a função de transferência é $K_D s$.

No controle derivativo, logo que o sinal começa a variar, a saída do controlador pode ser bem grande, visto que ela é proporcional à taxa de variação do sinal de erro, e não ao valor dele. Então, ocorrem respostas iniciais rápidas aos sinais de erro. A Figura 15.6 mostra a saída do controlador quando existe uma taxa constante de variação do sinal de erro com o tempo. A saída do controlador é constante porque a taxa de variação é constante e a derivada deste sinal é determinada imediatamente. Entretanto, os controladores derivativos não respondem aos sinais de erro de estado estacionário, visto que neste caso a variação do erro com o tempo é zero. Por isto, o controle derivativo é usado sempre de forma combinada com o controle proporcional; a parte proporcional gera resposta para todos os sinais de erro, inclusive os de estado estacionário, enquanto que a parte derivativa responde à taxa de variação. A ação derivativa também pode ser um problema se a medida do sinal de processo apresentar ruído. As variações rápidas do ruído resultam em saídas que serão vistas pelo controlador como variações rápidas no sinal de erro e, assim, resultar em aumentos significativos na saída do controlador.

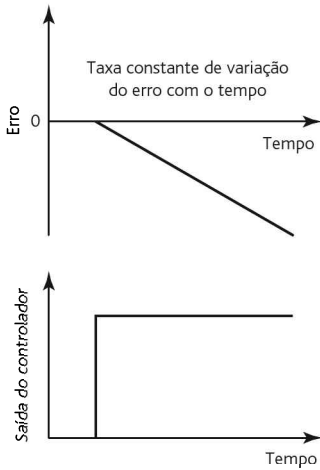


Figura 15.6 Controle derivativo.

A Figura 15.7 mostra a forma de um circuito controlador derivativo eletrônico. Este circuito envolve o uso de um amplificador operacional conectado como um diferenciador seguido por outro amplificador operacional conectado como um inversor. O tempo de derivação K_D é $R_2 C$.

15.5.1 Controle proporcional mais derivativo (PD)

O controle derivativo nunca é usado sozinho porque ele não é capaz de proporcionar uma saída quando houver um erro de estado estacionário e, assim, não haveria ne-

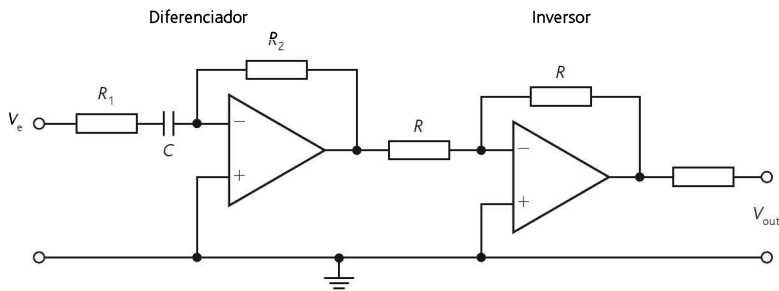


Figura 15.7 Controlador derivativo.

nhuma correção. Portanto, ele é usado invariavelmente em conjunto com o controle proporcional, de modo que este problema pode ser resolvido.

Em um controle proporcional mais derivativo a saída do controlador é dada por

$$\text{saída do controlador} = K_p e + K_D \frac{de}{dt}$$

em que K_p é a constante de proporcionalidade e K_D é a constante derivativa, de/dt é a taxa de variação do erro. O sistema tem uma função de transferência dada por

$$\text{saída do controlador } (s) = K_p E(s) + K_D s E(s)$$

Portanto, a função de transferência é $K_p + K_D s$. Esta função normalmente é escrita como

$$\text{função de transferência} = K_D \left(s + \frac{1}{T_D} \right)$$

em que $T_D = K_D/K_p$ é denominada **constante de tempo derivativa**.

A Figura 15.8 mostra como a saída do controlador pode variar quando há um erro variando constantemente. Existe uma variação rápida inicial na saída do controlador por causa da ação derivativa, seguida de uma variação gradual devido à ação proporcional. Portanto, esta forma de controle pode lidar com processos com variações rápidas.

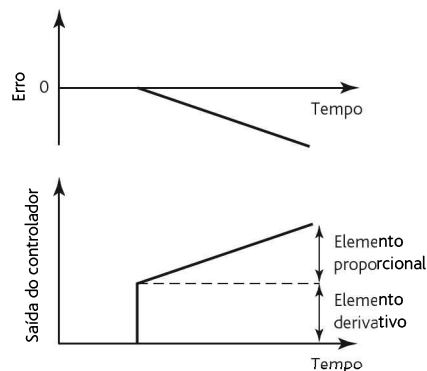


Figura 15.8 Controle PD.

15.6 Controle integral

O modo integral de controle é aquele em que a taxa de variação da saída de controle I é proporcional ao sinal de erro e de entrada:

$$\frac{dI}{dt} = K_I e$$

K_I é a constante de proporcionalidade e tem unidade de $1/s$. Integrando a equação acima, obtemos

$$\int_{I_0}^{I_{out}} dI = \int_0^t K_I e dt$$

$$I_{out} - I_0 = \int_0^t K_I e dt$$

I_0 é a saída do controlador no instante zero, I_{out} é a corrente de saída no instante t .

A função de transferência é obtida aplicando-se a transformada de Laplace. Portanto,

$$(I_{out} - I_0)(s) = \frac{1}{s} K_I E(s)$$

e assim

$$\text{função de transferência} = \frac{1}{s} K_I$$

A Figura 15.9 ilustra a ação de um controlador integral quando há uma entrada de erro constante no controlador. Podemos considerar o gráfico de duas formas. Quando a saída do controlador é constante, o erro é zero; quando a saída do controlador varia a uma taxa constante, o erro tem um valor constante. A forma alternativa de considerar o gráfico é em termos da área sob o gráfico de erro:

$$\text{área sob o gráfico de erro entre } t = 0 \text{ e } t = \int_0^t e dt$$

Portanto, até o momento em que o erro passa a existir, o valor da integral é zero. Portanto, $I_{out} = I_0$. Quando o erro ocorre, ele é mantido em um valor constante. Portanto, a área sob o gráfico aumenta com o tempo. Visto que a área aumenta a uma taxa constante, a saída do controlador aumenta a uma taxa constante.

A Figura 15.10 mostra a forma do circuito usada por um controlador integral eletrônico. Ele consiste de um amplificador operacional conectado como integrador e seguido de um outro amplificador operacional conectado como um somador que soma a saída do integrador com a saída do controlador no instante zero. K_I é $1/R_1 C$.

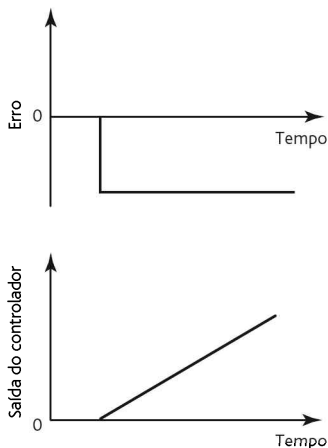


Figura 15.9 Controle integral.

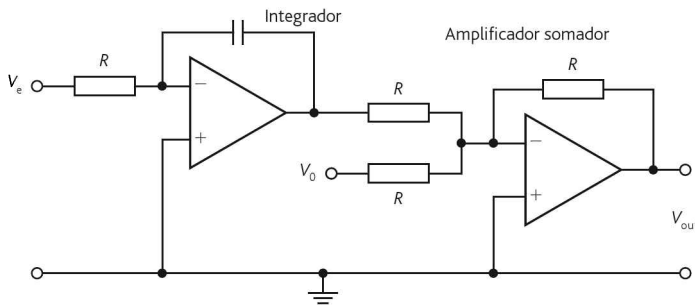


Figura 15.10 Controlador integral.

15.6.1 Controle proporcional mais integral (PI)

O modo integral de controle normalmente não é usado separadamente, mas em conjunto com o modo proporcional. Quando a ação integral é acrescentada a um sistema de controle proporcional, a saída do controlador é dada por

$$\text{saída do controlador} = K_p e + K_I \int e dt$$

em que K_p é a constante do controle proporcional, K_I é a constante do controle integral e e é o erro. A função de transferência é portanto

$$\text{função de transferência} = K_p + \frac{K_I}{s} = \frac{K_p}{s} \left(s + \frac{1}{T_I} \right)$$

em que $T_I = K_p/K_I$ e é a **constante de tempo da integral**.

A Figura 15.11(a) mostra como o sistema reage quando há uma mudança abrupta para um valor de erro constante. O erro faz com que a saída do controlador proporcional aumente e permanece constante, visto que o erro não varia. Há uma sobreposição desta resposta com a saída do controlador que aumenta de forma constante devido à ação integral. A Figura 15.11(b) mostra os efeitos das ações proporcional e integral se criarmos um sinal de erro que aumenta a partir do valor zero, retornando em seguida. Com a ação proporcional somente o controlador copia, como em um espelho, a variação e termina voltando ao seu ponto de operação original. Com a ação integral, a saída do controlador aumenta na proporção em que a área sob o gráfico erro versus tempo aumenta, e visto que ainda há um valor de área, mesmo quando o erro retorna para zero, há uma mudança na saída do controlador, que continua após o erro ter cessado.

15.7 Controlador PID

Combinar os três modos de controle (proporcional, integral e derivativo) resulta em um controlador conhecido **controlador de três modos** ou **controlador PID**. A equação que descreve sua ação pode ser escrita como

$$\text{saída do controlador} = K_p e + K_I \int e dt + K_D \frac{de}{dt}$$

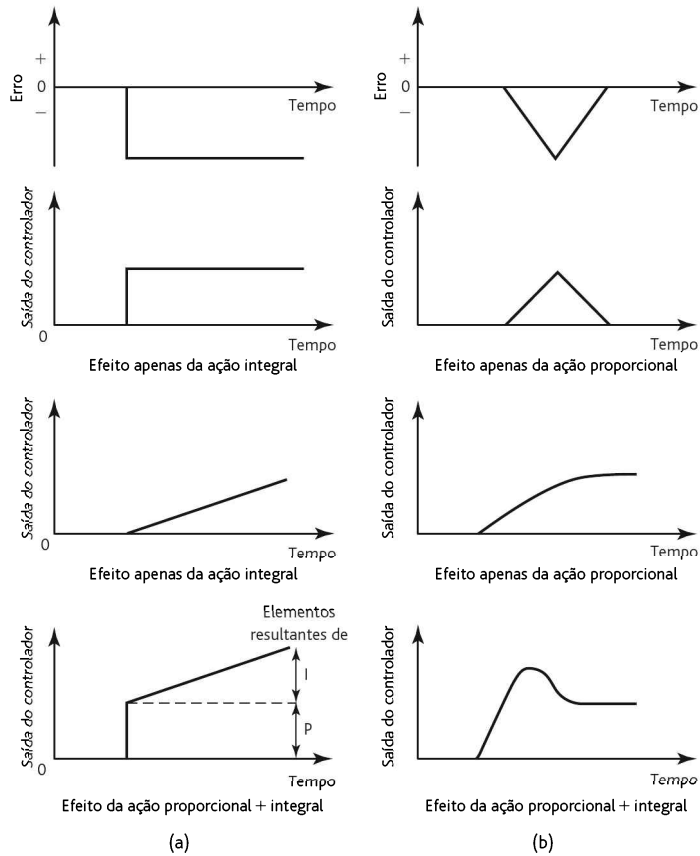


Figura 15.11 Controle PI.

em que K_p é a constante de proporcionalidade, K_I é a constante de integração e K_D é a constante de derivação. Aplicando a transformada de Laplace, obtemos

$$\text{saída do controlador} = K_p E(s) + \frac{1}{s} K_I E(s) + s K_D E(s)$$

e assim

$$\text{função de transferência} = K_p e + \frac{1}{s} K_I + s K_D = K_p \left(1 + \frac{1}{T_I s} + T_D s \right)$$

15.7.1 Circuitos PID com amplificador operacional

Um controlador PID pode ser construído a partir da combinação dos circuitos descritos neste capítulo para os modos proporcional, integral e derivativo em separado. Entretanto, um controlador mais prático pode ser construído com um único amplificador operacional. A Figura 15.12 mostra este circuito. A constante proporcional K_p é $R_I / (R + R_D)$, a constante derivativa K_D é $R_D C_D$ e a constante integral K_I é $1 / R_I C_I$.

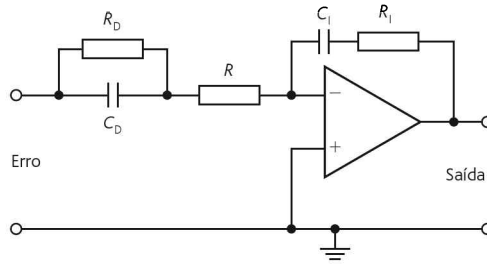


Figura 15.12 Circuito PID.

15.8 Controladores digitais

A Figura 15.13 mostra a base de um sistema de controle digital direto que pode ser usado em processos contínuos; o termo controle digital direto é usado quando o controlador digital, basicamente um microprocessador, controla o sistema de controle em malha fechada. O controlador recebe entradas a partir dos sensores, executa os programas de controle e fornece a saída para os elementos de correção. Estes controladores necessitam de entradas que são digitais, processam as informações na forma digital e fornecem uma saída em formato digital. Como muitos sistemas de controle têm medidas analógicas, é usado um conversor analógico-digital (ADC) nas entradas. Um circuito de clock fornece pulsos em intervalos regulares de tempo e determina quando são obtidas, pelo ADC, as amostras da variável controlada. Estas amostras são convertidas para sinais digitais, que são comparados pelo microprocessador com o valor desejado (*set point*), gerando o sinal de erro. O microprocessador pode então iniciar um modo de controle para processar o sinal de erro e fornecer uma saída digital. O modo de controle usado pelo microprocessador é determinado pelo programa usado pelo microprocessador para processar os sinais digitais, ou seja, o software. A saída digital, que geralmente passa por um conversor digital-analógico, visto que os elementos de correção geralmente requerem sinais analógicos, pode ser usada para iniciar a ação de correção.

Um controlador digital opera basicamente seguindo o seguinte ciclo de eventos:

- 1 Amostrar o valor medido.
- 2 Compará-lo ao valor desejado (*set point*) e estabelecer o erro.

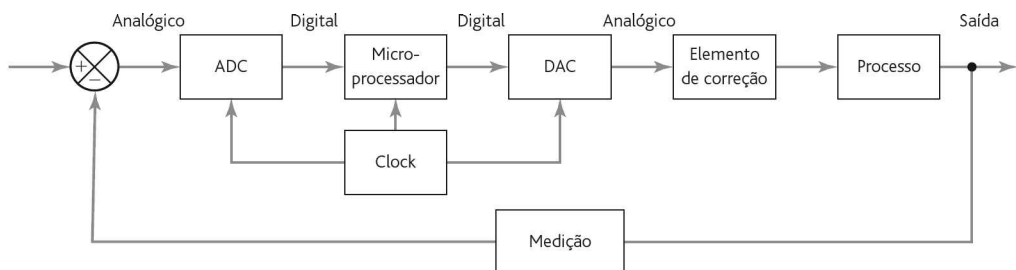


Figura 15.13 Sistema de controle digital em malha fechada.

- 3 Executar cálculos baseados no valor do erro e armazenar os valores das entradas e saídas anteriores para obter o sinal de saída.
- 4 Enviar o sinal de saída para o DAC.
- 5 Esperar decorrer o tempo do intervalo entre amostras antes de repetir o ciclo.

Os controladores microprocessados têm vantagem sobre os analógicos em que a forma da ação de controle, ou seja, os modos proporcional ou PID, podem ser alterados simplesmente mudando-se o software do computador. Nenhuma mudança no hardware ou conexões elétricas é necessária. Na verdade, a estratégia de controle pode ser alterada pelo programa do computador durante a ação de controle em resposta à situação existente.

Eles também têm outras vantagens. No caso do controle analógico, são necessários controladores analógicos em separado para cada processo controlado. Usando um microprocessador, muitos processos em separado podem ser controlados através de um processo de amostragem com o uso de multiplexador (veja a Seção 4.4). O controle digital proporciona melhor precisão que o analógico porque os amplificadores e outros componentes usados nos sistemas analógicos mudam suas características com o tempo e a temperatura, apresentando assim deriva (*drift*), enquanto que o controle analógico, que opera com sinais apenas no modo *on/off*, não sofre de deriva da mesma forma.

15.8.1 Implementação dos modos de controle

Para produzir um controlador digital para responder a um determinado modo de controle, é necessário produzir um programa adequado para o controlador. O programa tem que indicar como o sinal de erro digital em um determinado instante tem que ser processado para obter a saída necessária para o elemento de correção que vem em seguida. O processamento pode envolver a entrada atual juntamente com entradas e saídas anteriores. O programa faz com que o controlador execute uma equação de diferença (veja a Seção 4.6).

A função de transferência para um controlador analógico PID é

$$\text{função de transferência} = K_P + \frac{1}{s}K_I + sK_D$$

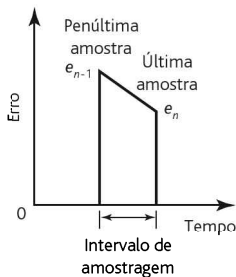


Figura 15.14 Sinais de erro.

Multiplicar por s equivale à diferenciação. Entretanto, podemos considerar o gradiente da resposta temporal para o sinal de erro no instante atual como sendo (a última amostra do erro e_n menos a penúltima amostra do erro e_{n-1})/(intervalo de amostragem T_s)(Figura 15.14).

A divisão por s é equivalente à integração. Entretanto, podemos considerar a integral do erro no final do período de amostragem como sendo a área sob o gráfico erro versus tempo durante o último intervalo de amostragem mais a soma das áreas sob o gráfico para todas as amostras prévias (Int_{prev}). Se o período de amostragem for curto em relação ao tempo envolvido, então a área durante o último intervalo de amostragem é aproximadamente $\frac{1}{2}(e_n + e_{n-1})/T_s$ (veja a Seção 4.6 para uma outra aproximação conhecida como aproximação de Tustin). Portanto, podemos escrever para a saída do controlador x_n para um determinado instante a função de transferência equivalente como

$$x_n = K_P e_n + K_I \left(\frac{(e_n + e_{n-1})T_s}{2} + \text{Int}_{\text{prev}} \right) + K_D \frac{e_n - e_{n-1}}{T_s}$$

Podemos reorganizar esta equação para obter

$$x_n = Ae_n + Be_{n-1} + C(\text{Int}_{\text{prev}})$$

em que $A = K_p + 0,5K_1T_s + K_D/T_s$, $B = 0,5K_1T_s - K_D/T_s$ e $C = K_1$.

O programa para o controle PID se torna então:

- 1 Determinar os valores de K_p , K_1 e K_D .
- 2 Determinar os valores iniciais de e_{n-1} , Int_{prev} e o intervalo de amostragem T_s .
- 3 Resetar o temporizador do intervalo de amostragem.
- 4 Obter o erro e_n .
- 5 Calcular y_n usando a equação anterior.
- 6 Atualizar, preparando para o próximo cálculo, o valor da área anterior para $\text{Int}_{\text{prev}} + 0,5(e_n + e_{n-1})T_s$.
- 7 Atualizar, preparando para o próximo cálculo, o valor do erro, fazendo e_{n-1} igual a e_n .
- 8 Esperar decorrer o tempo de amostragem.
- 9 Ir para o passo 3 e repetir o *loop*.

15.8.2 Taxa de amostragem

Quando um sinal contínuo é amostrado, para os valores amostrados refletirem precisamente o sinal contínuo, eles têm que ser suficientemente próximos entre si no tempo para que o sinal não tenha uma variação significativa entre amostras. Durante um intervalo de amostragem, nenhuma informação é realimentada para o controlador em relação a variações na saída. Na prática, isto quer dizer que as amostras têm que ser obtidas a uma taxa maior que duas vezes a maior componente frequencial no sinal contínuo. Esta regra é denominada teorema da amostragem de Shannon (veja a Seção 4.2.1). Em sistemas de controle digital, a taxa de amostragem é geralmente muito maior do que isto.

15.8.3 Sistema de controle computadorizado

Tipicamente um sistema de controle computadorizado consiste nos elementos mostrados na Figura 15.13 com o *set point* e os parâmetros de controle inseridos a partir de um teclado. O software para uso com este sistema provê o programa de instruções necessário, por exemplo, para o computador implementar o modo de controle PID, provê o display do operador, reconhece e processa as instruções inseridas pelo operador, provê informações sobre o sistema, provê instruções de iniciar e paralisar a operação e gera informações de clock/calendário. O display do operador mostra informações como o *set point*, o valor medido atual, o intervalo de amostragem, o erro, os parâmetros do controlador e o estado do elemento de correção. O display deve ser atualizado periodicamente a cada poucos segundos.

15.9 Desempenho dos sistemas de controle

A função de transferência de um sistema de controle é afetada pelo modo escolhido para o controlador. Portanto, a resposta de um sistema a, por exemplo, uma entrada é afetada. Considere o sistema simples mostrado na Figura 15.15.

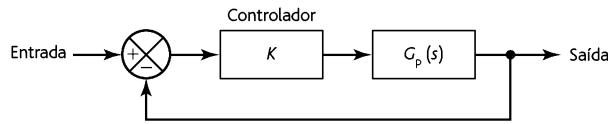


Figura 15.15 Sistema de controle.

No controle proporcional a função de transferência do caminho direto é $K_p G_p(s)$, e assim a função de transferência do sistema realimentado $G(s)$ é

$$G(s) = \frac{K_p G_p(s)}{1 + K_p G_p(s)}$$

Suponha que temos um processo que é de primeira ordem com uma função de transferência de $1/(\tau s + 1)$, em que τ é a constante de tempo (pode representar um motor CC, geralmente modelado como um sistema de primeira ordem – veja a Seção 13.5.1). Com controle proporcional, e realimentação unitária, a função de transferência do sistema de controle se torna

$$G(s) = \frac{K_p/(\tau s + 1)}{1 + K_p/(\tau s + 1)} = \frac{K_p}{\tau s + 1 + K_p}$$

O sistema de controle permanece um sistema de primeira ordem. O controle proporcional tem o efeito de alterar apenas a forma da resposta de primeira ordem do processo. Sem o controlador, a resposta à entrada de degrau unitário era (veja a Seção 12.3.2)

$$y = 1 - e^{-t/\tau}$$

Agora ela é

$$y = K_p \left(1 - e^{-t/(\tau/(1+K_p))} \right)$$

O efeito do controle proporcional foi reduzir a constante de tempo de τ para $\tau/(1 + K_p)$, respondendo de forma mais rápida quanto maior for o valor de K_p . Ele também diminui o erro de estado estacionário.

No controle integral, temos uma função de transferência no caminho direto de $K_I G_p(s)/s$, e assim a função de transferência do sistema é

$$G(s) = \frac{K_I G_p(s)}{s + K_I G_p(s)}$$

Portanto, se agora temos um processo que é de primeira ordem com uma função de transferência $1/(\tau s + 1)$, com controle proporcional e realimentação unitária a função de transferência do sistema de controle se torna

$$G(s) = \frac{K_I/(\tau s + 1)}{s + K_I/(\tau s + 1)} = \frac{K_I}{s(\tau s + 1) + K_I} = \frac{K_I}{\tau s^2 + s + K_I}$$

Agora o sistema de controle é de segunda ordem. Com uma entrada em degrau, o sistema fornecerá uma resposta de segunda ordem em vez de primeira ordem.

Em um sistema com controle derivativo, a função de transferência no caminho direto é $sK_D G(s)$ e assim, com uma realimentação unitária, a função de transferência do sistema é

$$G(s) = \frac{sK_D G_p(s)}{1 + K_D G_p(s)}$$

Em um processo de primeira ordem com uma função de transferência $1/(\tau s + 1)$, o controle derivativo resulta em uma função de transferência geral de

$$G(s) = \frac{sK_D/(\tau s + 1)}{1 + sK_D/(\tau s + 1)} = \frac{sK_D}{\tau s + 1 + sK_D}$$

15.10 Sintonia do controlador

O termo **sintonia** é usado para descrever o processo de seleção do melhor ajuste do controlador. Em um controlador proporcional, isto significa selecionar o valor de K_p ; em um controlador PID, as três constantes K_p , K_i e K_D têm que ser ajustadas. Existem vários métodos para fazer isto; discutiremos aqui apenas dois métodos, ambos de Ziegler e Nichols. Eles consideraram que quando o sistema controlado estivesse em malha aberta, uma aproximação razoável para o comportamento dele seria um sistema de primeira ordem com um atraso de tempo embutido. Baseado nisto, eles deduziram os parâmetros para um desempenho otimizado.

15.10.1 Método de reação de processo

A malha de controle do processo é aberta, geralmente entre o controlador e a unidade de correção, de forma que nenhuma ação de controle ocorre. Então é aplicado um sinal de teste na entrada para a unidade de correção, e a resposta da variável controlada é determinada. O sinal de teste deve ser tão pequeno quanto possível. A Figura 15.16 mostra a forma do sinal de teste e a resposta típica. O sinal de teste é um sinal em degrau com a amplitude expressa como uma porcentagem da variação de P na unidade de correção. O gráfico da variável medida em função do tempo é denominado **curva de reação do processo**. A variável medida é expressa como uma porcentagem do valor de fundo de escala.

Uma linha tangente é desenhada para se obter o gradiente máximo no gráfico. Na Figura 15.16, o gradiente máximo R é M/T . O tempo entre o início do sinal de teste e o ponto no qual esta tangente intercepta o eixo do tempo no gráfico é denominado atraso L . A Tabela 15.1 fornece os critérios de Ziegler e Nichols para o controle feito baseado nos valores de P , R e L .

Considere o seguinte exemplo. Determine a configuração desejada para um controlador PID que apresentou a curva de

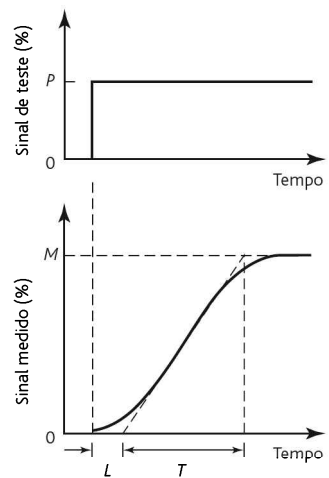


Figura 15.16 Curva de reação do processo.

Tabela 15.1 Critério da curva de reação do processo

Modo de controle	K_p	T_I	T_D
P	P/RL		
PI	$0,9P/RL$	$3,33L$	
PID	$1,2P/RL$	$2L$	$0,5L$

reação do processo mostrada na Figura 15.17 quando o sinal de teste variou 6% no posicionamento da válvula de controle. Desenhando a tangente para a parte de gráfico com gradiente máximo, obtemos um atraso L de 150 s e um gradiente R de $5/300 = 0,017/s$. Portanto,

$$K_p = \frac{1,2P}{RL} = \frac{1,2 \times 6}{0,017 \times 150} = 2,82$$

$$T_I = 2L = 300 \text{ s}$$

$$T_D = 0,5L = 0,5 \times 150 = 75 \text{ s}$$

15.10.2 Método de ciclo máximo

Neste método, as ações integral e derivativa são reduzidas primeiro a seus valores mínimos. A constante proporcional K_p é ajustada em um valor baixo e gradualmente aumentada. Enquanto se faz isto, são aplicados distúrbios no sistema. Este processo continua até que ocorram oscilações contínuas. O valor crítico da constante proporcional K_{pc} na qual isto ocorre é anotado e o período das oscilações T_c é medido. A Tabela 15.2 mostra como os critérios recomendados por Ziegler e Nichols para sintonia de controladores estão relacionados a estes valores de K_{pc} . A banda proporcional crítica é $100/K_{pc}$.

Considere o seguinte exemplo. Na sintonia de um sistema de controle nos três modos através do método de ciclo máximo, verificamos que oscilações começam quando K_{pc} é 3,33. As oscilações têm um periodicidade de 500 s. Quais são os ajustes adequados do controlador? Usando os critérios dados na Tabela 15.2, temos $K_p = 0,6K_{pc} = 0,6 \times 3,33 = 2,0$, $T_I = T_c/2,0 = 500/2 = 2,5 \text{ s}$, $T_D = T_c/8 = 500/8 = 62,5 \text{ s}$.

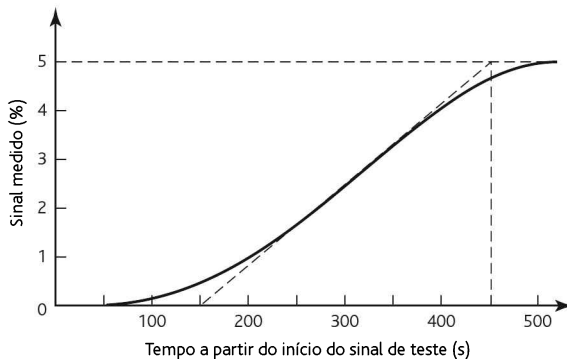


Figura 15.17 Exemplo de uma curva de processo.

Tabela 15.2 Critérios do ciclo máximo

Modo de controle	K_p	T_i	T_D
P	$0,5K_{pc}$		
PI	$0,45K_{pc}$	$T_d/1,2$	
PID	$0,6K_{pc}$	$T_d/2,0$	$T_d/8$

15.11 Controle de velocidade

Considere o problema do controle de movimento de uma carga através de um motor. Como o sistema do motor é provavelmente de segunda ordem, o controle proporcional fará com que a saída do sistema leve um tempo para alcançar o deslocamento desejado quando houver, por exemplo, uma entrada em degrau no sistema, e poderá oscilar em torno do valor desejado. O sistema levará um tempo para responder ao sinal de entrada. Uma velocidade maior de resposta, com poucas oscilações, pode ser obtida usando-se um controle PD em vez de apenas P. Entretanto, há uma alternativa para conseguir o mesmo efeito através do uso de uma segunda malha de realimentação que fornece uma medida relacionada à taxa na qual o deslocamento varia. Isto é denominado **realimentação de velocidade**. A Figura 15.18 mostra este sistema; a realimentação de velocidade pode envolver o uso de um tacogerador que fornece um sinal proporcional à velocidade de rotação do eixo do motor e, portanto, à taxa na qual o deslocamento varia, e o deslocamento pode ser monitorado usando-se um potenciômetro rotativo.

15.12 Controle adaptativo

Existem muitas situações de controle em que os parâmetros da planta variam com o tempo ou, talvez, com a carga, como, por exemplo, no caso de um robô que move cargas quando a carga varia. Se a função de transferência da planta muda, então é necessário uma nova sintonia do sistema para que valores otimizados sejam determinados para as constantes proporcional, derivativa e integral. Para os sistemas de controle considerados até aqui, consideramos que cada sistema, uma vez sintonizado, mantém os valores das constantes proporcional, derivativa e integral até que o operador decida fazer uma nova sintonia. Um sistema alternativo a este é o **sistema de controle adaptativo**, que se “adapta” às mudanças e altera seus parâmetros para se ajustar às circunstâncias em vigor.

O sistema de controle adaptativo é baseado no uso de um microprocessador que faz a função do controlador. Este dispositivo permite que o modo e os parâmetros de controle sejam adaptados para se ajustarem às circunstâncias, modificando-as conforme as circunstâncias mudam.

Um sistema de controle adaptativo pode ser considerado como tendo três estágios de operação:

- 1 Começa a operar com as condições definidas no controlador tendo como base uma condição inicial considerada.

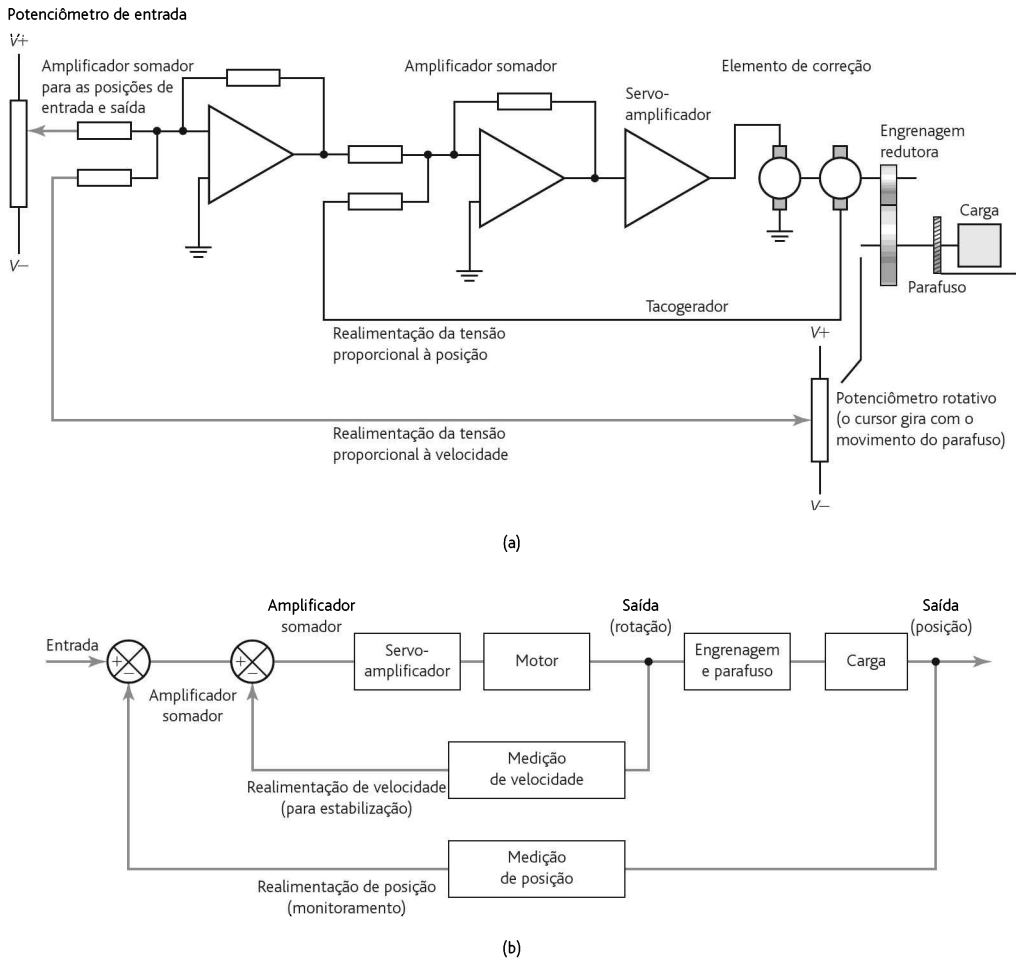


Figura 15.18 Sistema com realimentação de velocidade: (a) diagrama descritivo do sistema, (b) diagrama em bloco do sistema.

- 2 O desempenho desejado é comparado continuamente com o desempenho atual do sistema.
- 3 O modo de sistema de controle e os parâmetros são automática e continuamente ajustados para minimizar a diferença entre o desempenho atual do sistema e o desejado.

Por exemplo, com um sistema de controle operando no modo proporcional, a constante proporcional K_p pode ser automaticamente ajustada às circunstâncias, mudando conforme elas mudam. Os sistemas de controle adaptativo podem assumir algumas formas. As três formas mais comuns são:

- 1 Controle com ganho escalonado
- 2 Reguladores autoajustáveis
- 3 Sistemas adaptativos por modelo de referência

15.12.1 Controle com ganho escalonado

Com o **controle de ganho escalonado** ou, como algumas vezes é conhecido, **controle adaptativo pré-programado**, variações prévias nos parâmetros do controlador são feitas com base em uma medição auxiliar de uma variável de processo. A Figura 15.19 ilustra este método. O termo controle com ganho escalonado foi usado porque o único parâmetro originalmente ajustado é o ganho, ou seja, a constante de proporcionalidade K_p .

Por exemplo, para um sistema de controle usado para controlar alguma carga, os parâmetros do sistema poderiam ser calculados para diferentes valores de carga e armazenados em uma tabela na memória do controlador. Um célula de carga poderia ser usada para medir a carga atual e gerar um sinal para o controlador indicando o valor da massa que é usado pelo controlador para selecionar os parâmetros adequados.

Uma desvantagem deste sistema é que os parâmetros de controle têm que ser determinados para muitas condições de operação a fim de que o controlador possa selecionar um conjunto de parâmetros para se ajustar às condições predominantes. Entretanto, uma vantagem é que as mudanças nos parâmetros podem ser feitas rapidamente quando as condições mudam.

15.12.2 Reguladores autoajustáveis

Com o **controle autoajustável**, o sistema sintoniza continuamente seus próprios parâmetros baseado no monitoramento da variável que o sistema está controlando e da saída a partir do controlador. A Figura 15.20 ilustra as características deste sistema.

O autoajuste é frequentemente encontrado em controladores PID comerciais geralmente conhecidos como **autossintonizáveis**. Quando o operador pressiona um botão, o controlador injeta um pequeno distúrbio no sistema e mede a resposta. Esta resposta é comparada com uma resposta desejada e os parâmetros do controlador são ajustados pelo método modificado de Ziegler-Nichols, para que a resposta atual se aproxime da desejada.

15.12.3 Sistemas adaptativos por modelo de referência

Com o **sistema adaptativo por modelo de referência**, é desenvolvido um modelo de precisão do sistema. O valor ajustado é então usado como entrada para os sistemas real e modelo, e a diferença entre as saídas deles é comparada. A diferença

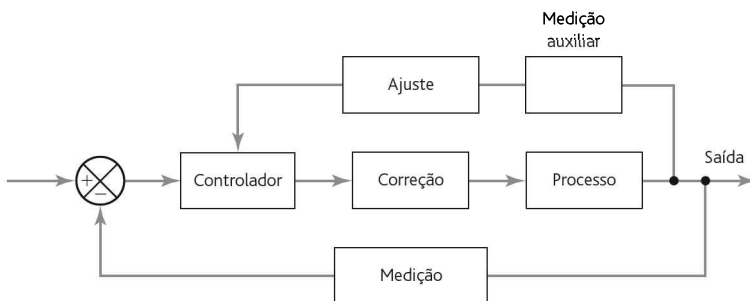


Figura 15.19 Controle de ganho escalonado.

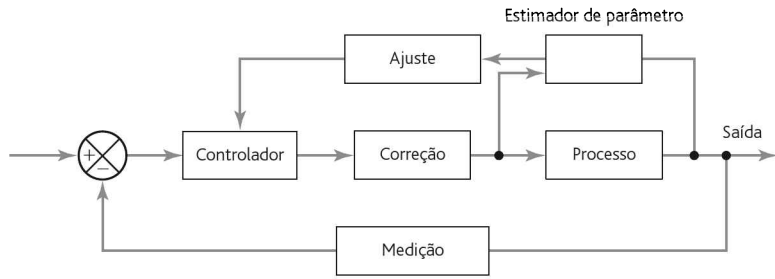


Figura 15.20 Regulador autoajustável.

nestes sinais é usada para ajustar os parâmetros do controlador para minimizar a diferença. A Figura 15.21 ilustra as características deste sistema.

Resumo

O **erro de estado estacionário** é a diferença entre o valor desejado (*set point*) e a saída após todos os transientes cessarem.

Os **modos de controle** são: **on/off**, no qual o controlador fornece um sinal de correção do tipo liga/desliga; **proporcional (P)**, no qual o sinal de correção é proporcional ao erro; **derivativo (D)**, no qual o sinal de correção é proporcional à taxa de variação do erro; e **integral (I)**, no qual o sinal de correção é proporcional à integral do erro no tempo. A função de transferência para um sistema PID é

$$\text{função de transferência} = K_{pe} + \frac{1}{s}K_I + sK_D = K_P \left(1 + \frac{1}{T_I s} + T_D s \right)$$

Um **controlador digital** basicamente opera amostrando os valores medidos, comparando-os com o valor ajustado e estabelecendo o erro, realizando cálculos baseado no valor do erro e nos valores anteriores armazenados de entradas e saídas para obter o sinal de saída, gerando uma nova saída e aguardando a próxima amostra.

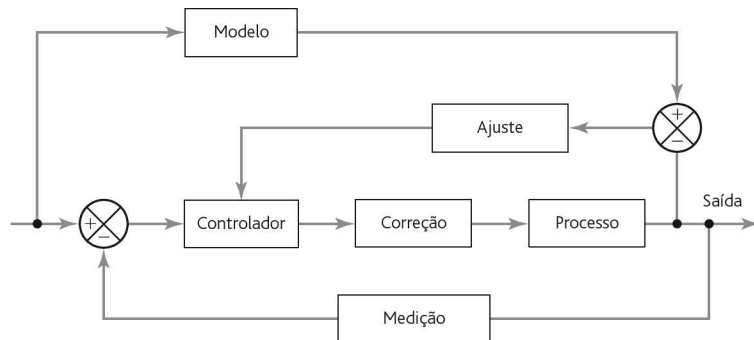


Figura 15.21 Controle por modelo de referência.

O termo **sintonia** é usado para descrever o processo de seleção dos melhores parâmetros do controlador, ou seja, os valores de K_p , K_i e K_D .

O termo **controle adaptativo** é usado para sistemas que se adaptam às mudanças e variações em seus parâmetros para se ajustar às circunstâncias dominantes. Três formas comuns são controle com ganho escalonado, autoajustável e sistema adaptativo por modelo de referência.

Problemas

- 15.1 Quais são as limitações do controle *on/off* e em que situações este sistema de controle é mais usado?
- 15.2 Um controlador em modo *on/off* liga um aquecedor em uma sala quando a temperatura cai para 20°C e o desliga quando ela alcança 24°C. Quando o aquecedor é ligado, o ar na sala aumenta de temperatura a uma taxa de 0,5°C por minuto; quando o aquecedor é desligado, ele esfria 0,2°C por minuto. Se os atrasos de tempo no sistema de controle forem desprezíveis, qual será o tempo gasto para (a) o aquecedor desligar, (b) o aquecedor ligar?
- 15.3 Um controlador *on/off* é usado para controlar o nível de água em um tanque abrindo ou fechado uma válvula que, na posição aberta, permite que a água entre no tanque a uma taxa de 0,4m³/s. O tanque tem uma área de seção reta de 12 m² e a água sai dele a uma taxa constante de 0,2m³/s. A válvula abre quando o nível de água alcança 4 m e fecha em 4,4 m. Qual será o tempo gasto para (a) a válvula fechar, (b) a válvula abrir?
- 15.4 Um controlador proporcional é usado para controlar a altura da água em um tanque onde o nível de água pode variar de 0 a 4,0 m. A altura desejada da água é 3,5 m e o controlador fecha a válvula completamente quando a água sobe para 3,9 m e abre a mesma quando a água cai para 3,1 m. Qual será a função de transferência necessária?
- 15.5 Descreva e compare as características do (a) controle proporcional, (b) do controle proporcional mais integral, (c) do controle proporcional mais derivativo.
- 15.6 Determine os valores de K_p , T_i e T_D necessários para um controlador PID que gera uma curva de reação de processo com um atraso L de 200 s e um gradiente R de 0,010%/s quando o sinal de teste for de 5% da variação no controle de posicionamento da válvula.
- 15.7 Ao se sintonizar um sistema de controle PID através do método de ciclo máximo, verificou-se que oscilações começavam quando o valor proporcional crítico era 5. As oscilações tinham período de 200 s. Quais são os valores adequados para K_p , T_i e T_D ?
- 15.8 Explique em que se baseiam as seguintes formas de funções de sistemas de controle adaptativo: (a) ganho escalonado, (b) autoajuste, (c) referência por modelo.

- 15.9 Um motor CC se comporta como um sistema de primeira ordem, com uma função de transferência relacionando a posição da saída na qual ele posicionou uma carga com o sinal de entrada $1/s(1 + s\tau)$. Se a constante de tempo τ for 1 s e o motor é usado em um sistema de controle de malha fechada, determine o valor da constante de proporcionalidade que proporciona uma resposta em malha fechada com um *overshoot* de 25%.
- 15.10 Um pequeno motor ultrassônico usado para mover as lentes de um câmera de foco automático (veja a Seção 24.2.3) aciona um anel com tão pouca inércia que a função de transferência que relaciona a posição angular com o sinal de entrada é representada por $1/cs$, em que c é a constante de proporcionalidade que relaciona o torque de fricção e a velocidade angular. Se o motor for controlado por um sistema de malha fechada com uma realimentação unitária, que tipo de comportamento pode ser esperado se o controle proporcional for usado?

16

Inteligência Artificial

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Explicar qual é o significado de máquina inteligente e as capacidades destas máquinas.
- Explicar o significado de redes neurais e a importância delas no reconhecimento de padrões.
- Explicar o termo lógica *fuzzy*.

16.1 Qual é o significado de inteligência artificial?

O que constitui uma máquina inteligente? Uma definição de um dicionário para a palavra “inteligente” poderia ser “dotado da habilidade da razão”. Quanto mais inteligente uma pessoa é, mais a consideramos capaz de aprender, generalizar a partir deste conhecimento adquirido, capaz de raciocinar e deduzir considerando as possibilidades, aprendendo a partir dos erros. Podemos aplicar os mesmos critérios para uma máquina: uma **máquina inteligente** é aquela que é dotada da habilidade da razão.

Um sistema de aquecimento central toma decisões dentro de sua capacidade de operação. Por exemplo, a caldeira deve ser ligada ou desligada como um resultado da informação de um termostato? Entretanto, este procedimento não é considerado resultado de inteligência porque a máquina não é capaz de raciocinar e tomar decisão entre uma ampla gama de condições. Por exemplo, ela não pode reconhecer um padrão de entrada a partir de um termostato e, assim, fazer previsões a respeito do clima para ligar ou desligar a caldeira. Ela apenas faz o que é determinado que seja feito. Ela não “pensa por conta própria”.

Neste capítulo faremos uma breve abordagem sobre os conceitos básicos associados a máquinas inteligentes.

16.1.1 Autorregulação

Podemos considerar os sistemas de malha fechada discutidos em capítulos anteriores como sendo sistemas autorregulados, pois regulam a saída do sistema no valor desejado. Portanto, um sistema de aquecimento central controlado termostaticamente é usado para manter a temperatura de um ambiente no valor ajustado no termostato. Entretanto, este sistema não pode ser considerado inteligente, pois ele faz meramente o que foi determinado.

16.2 Percepção e conhecimento

A **percepção** em um sistema inteligente é a coleção de informações usando sensores e a organização das informações obtidas de modo que decisões possam ser tomadas. Por exemplo, um sistema de controle usado em uma linha de produção pode ter uma câmera de vídeo para observar os componentes em uma correia transportadora. Os sinais recebidos da câmera permitem que uma representação computacional dos componentes seja feita, de forma que características sejam identificadas. Esta representação deve conter informações sobre elementos críticos dos componentes. Estes podem ser comparados com representações dos componentes, de modo que decisões possam ser tomadas pelo sistema de controle, como, por exemplo, se o componente está corretamente montado ou, talvez, identificar qual é o componente. Em seguida, uma ação pode ser tomada pelo sistema de controle que pode ser rejeitar componentes defeituosos ou desviar componentes específicos para boxes específicos.

Portanto, em um sistema mecatrônico, a percepção envolve o acúmulo de informações apropriadas sobre o sistema e o meio ambiente por meio de sensores, e a decodificação e o processamento delas para se obter informação útil que possa ser usada em outra parte do sistema para tomar decisões.

16.2.1 Conhecimento

Uma vez que a máquina coletou e organizou a informação, ela tem que tomar decisões sobre o que fazer com base nas informações coletadas. Isto é denominado **conhecimento**. Vital para a percepção e o conhecimento é o **reconhecimento de padrões**. Quais são os padrões nos dados coletados?

Os humanos são muito bons no reconhecimento de padrões. Imagine um guarda de segurança observando monitores de TV. Ele é capaz de observar os monitores e reconhecer padrões incomuns, como por exemplo, a existência de uma pessoa onde não deveria estar, um objeto que foi mexido etc. Estas capacidades são necessárias em máquinas inteligentes. Um sistema de piloto automático em aviões monitora uma grande quantidade de informações e, baseado em padrões obtidos dos dados, toma decisões sobre como ajustar os controles do avião.

O reconhecimento de padrões feito por máquinas pode ser conseguido por uma máquina que tem um conjunto de padrões em sua memória, sendo os padrões colecionados então comparados com os primeiros e verificada a semelhança. O número de padrões na memória pode aumentar a partir de modelos ou um processo de treinamento no qual dados são armazenados para uma gama de objetos ou situações e a estes são dados códigos de identificação. Por exemplo, para o reco-

nhcimento de moedas, informações como diâmetro e cor podem ser armazenadas. Portanto, uma moeda de 5 centavos de dólar pode ser classificada como tendo um diâmetro de 2,25 cm e uma cor com um grau específico de vermelho (é uma moeda de bronze). Entretanto, uma máquina inteligente precisa levar em consideração o desgaste e a sujeira das moedas e ainda ser capaz de reconhecer a moeda como sendo de 5 centavos.

16.2.2 Redes neurais

Neste exemplo das moedas, foram consideradas apenas duas dimensões, o diâmetro e a cor. Em situações mais complexas pode haver muito mais dimensões. O cérebro humano se depara com a escolha e a classificação de informações multidimensionais, e faz isto usando **redes neurais**. As redes neurais artificiais agora estão sendo usadas com máquinas inteligentes. Estas redes não precisam ser programadas, mas podem aprender e generalizar a partir de exemplos e treinamentos. Uma rede neural (Figura 16.1) é composta de um grande número de unidades de processamento interconectadas, ou seja, as saídas de algumas unidades são entradas de outras. Cada processador na rede recebe informações em suas entradas e multiplica cada uma por um fator de ponderação. Se a operação for uma AND, ela soma as entradas ponderadas e fornece uma saída de 1 se a soma exceder certo valor ou for positiva. Por exemplo, podemos ter uma entrada de 1 com um fator de ponderação de $-1,5$ e obter $-1,5$, uma outra entrada de 1 com um fator de ponderação de $1,0$ e obter $1,0$, e uma terceira entrada de 1 com um fator de ponderação de $1,0$ e obter $1,0$. A soma destas entradas ponderadas é $-1,5 + 1,0 + 1,0 = 0,5$ e, assim, uma saída de 1 se os valores são positivos para uma saída. Com as entradas sendo $1 \times -1,5$, $0 \times 1,0$ e $0 \times 1,0$, a soma ponderada é $-1,5$, obtendo-se assim uma saída de 0. A rede pode ser programada a partir de exemplos e ser capaz de aprender.

16.3 Raciocínio

O **raciocínio** é o processo de partir de alguma coisa conhecida para outra não conhecida. Existem vários mecanismos para realizar o raciocínio.

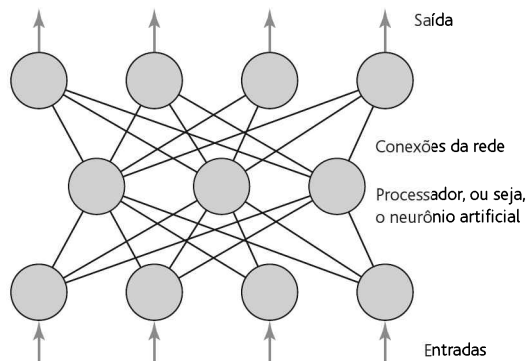


Figura 16.1 Rede neural.

16.3.1 Mecanismos de raciocínio

Um exemplo de **raciocínio determinístico** é o uso da regra “se-então”. Portanto, podemos deduzir que *se* uma moeda tem um diâmetro de 1,25 cm *então* ela é uma moeda de 5 centavos de dólar. Se a primeira parte da afirmativa é verdadeira, então a segunda é verdadeira. Nesta forma de raciocínio, temos uma situação *verdadeiro-falso*, e considera-se que não existe um conhecimento *default*, de modo que quando a dedução é feita, não existem exceções. Portanto, neste exemplo não estamos levando em conta a existência de uma moeda de um outro país com o mesmo diâmetro.

O **raciocínio não determinístico** nos permite fazer previsões baseadas em probabilidade. Se lançarmos uma moeda, existem duas formas da moeda cair: com a face voltada para cima ou para baixo. Dentre estas duas formas, só existe uma na qual a face cai voltada para cima. Portanto, a probabilidade da moeda cair com a face voltada para cima é de 1 em 2 ou 1/2. Uma forma alternativa de se chegar a este valor é lançar uma moeda repetidas vezes e, após uma longa série, em 1/2 das vezes ela cairá com a face voltada para cima. A Figura 16.2(a) mostra como podemos representar isto como uma árvore de probabilidades. Se lançarmos um dado de seis faces, a probabilidade de ele cair com uma das faces voltada para cima é 1/6. A Figura 16.2(b) mostra como podemos representar isto na forma de uma árvore de probabilidades. Cada ramo da árvore tem a probabilidade escrita. A chance de uma moeda cair com a cara ou a coroa voltada para cima é 1. Portanto, para uma árvore, a probabilidade total será 1.

Portanto, no exemplo da moeda de 5 centavos podemos querer considerar que existe uma probabilidade de 0,9 de uma moeda com diâmetro de 1,25 cm ser uma moeda de 5 centavos. No caso de um sistema mecatrônico, podemos monitorar a temperatura por, digamos, 1.000 horas e verificar que neste período a temperatura esteve alta durante 3 horas. Podemos então dizer que a probabilidade da temperatura ser alta é $3/1.000 = 0,003$.

Algumas vezes podemos conhecer a probabilidade de um evento ocorrer e querer estabelecer a probabilidade dele influenciar outro evento. Portanto, em um sistema mecatrônico pode ser necessário saber quais são as chances de, por exemplo, quando um sensor detectar uma pressão baixa, o sistema sobreaquecer, tendo em mente que poderia haver outras razões para a alta temperatura. Podemos representar isto como a árvore mostrada na Figura 16.3.

A **regra de Bayes** pode ser usada para resolver este problema. Isto pode ser expresso como

$$p(A|B) = \frac{p(B|A) \times p(A)}{p(B)}$$

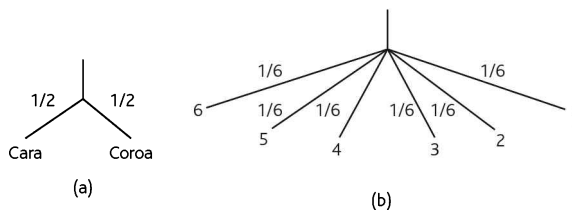


Figura 16.2 Árvores de probabilidades.

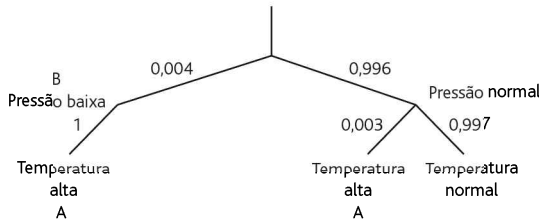


Figura 16.3 Uma árvore de probabilidades condicionais.

em que $p(A|B)$ é a probabilidade de A acontecer dado que B aconteceu, $p(B|A)$ é a probabilidade de B acontecer dado que A aconteceu, $p|A|$ é a probabilidade de A acontecer, $p|B|$ é a probabilidade de B acontecer. Portanto, se a probabilidade da temperatura alta ocorrer no sistema $p|A|$ for 0,003, ou seja, ocorre uma temperatura alta 3 vezes em 1.000, e a probabilidade de estar ocorrendo um pressão baixa $p|B|$ for 0,004, ou seja, ocorre uma pressão baixa 4 vezes em 1.000, então, como podemos estar certos de que o sistema irá sobreaquecer se a pressão for baixa, ou seja, $p(B|A)$ é 1, temos que ter uma probabilidade condicional de $(1 \times 0,003)/0,004 = 0,75$ de que o sistema sobreaquecerá quando uma pressão baixa for detectada.

16.3.2 Raciocínio baseado em regras

A parte principal de um **sistema baseado em regras** é o seu conjunto de regras. Estas, quando combinadas com fatos, que em mecatrônica poderiam ser as entradas a partir dos sensores e usuários, permite que inferências sejam feitas, as quais são usadas para acionar atuadores e saídas de controle. A Figura 16.4 ilustra esta sequência. A combinação dos bancos de dados de fatos e regras é conhecida como base de conhecimento para a máquina. A inferência ocorre como resultado do raciocínio sobre os fatos de entrada combinados com as regras, e as decisões são enviadas aos atuadores.

As regras usadas são geralmente afirmativas da forma “se-então” (*if-then*). Portanto, podemos ter um grupo de regras para um sistema central de aquecimento da forma:

Se o aquecedor está ligado
Então a bomba é ligada

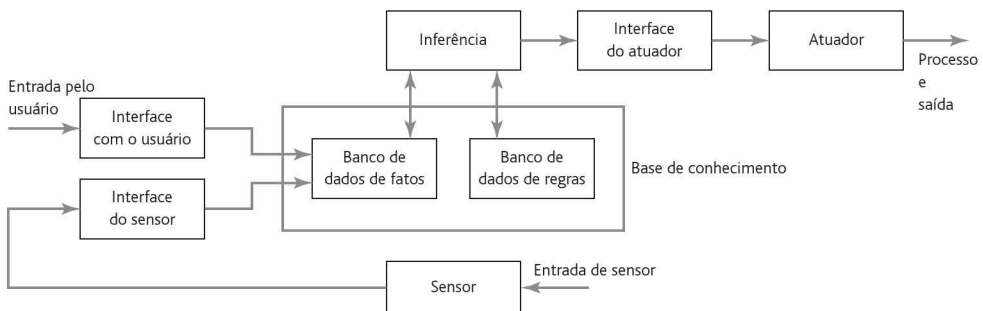


Figura 16.4 Sistema baseado em regras.

Se a bomba está ligada AND a temperatura da sala é menor do que 20°C
Então abra a válvula

Se o aquecedor estiver desligado
Então a bomba está desligada

etc.

O banco de dados de fatos com um sistema como este conteria os fatos:

- Temperatura da sala < 20°C
- Temporizador ligado
- Válvula aberta
- Aquecedor ligado
- Bomba ligada

As regras também podem ser na forma de proposições que envolvem declarações de probabilidades ou lógica *fuzzy*.

Lofti Zadeh propôs em 1965 uma forma de raciocínio que ficou conhecida como **lógica fuzzy**. Uma de suas principais ideias é que as proposições não necessitam ser classificadas com verdadeiras ou falsas, mas a veracidade ou a falsidade pode ser ponderada de forma que ela pode ser classificada entre as duas em uma escala. Uma **função de pertinência** é definida para um valor como se fosse membro de um grupo em particular. Portanto, podemos definir um conjunto de valores de temperatura como sendo de 0 a 20°C e um outro como sendo de 20 a 40°C. Se a temperatura for, por exemplo, 18°C, então a pertinência no conjunto de 0 a 20°C é 1 e no conjunto de 20 a 40°C é 0. Entretanto, com a lógica fuzzy definimos conjuntos sobrepostos, como por exemplo, FRIO (0 a 20°C), MORNO (10 a 30°C) e QUENTE (20 a 40°C). Uma temperatura de 18°C é portanto um membro de dois conjuntos. Se as funções de participação de conjuntos *fuzzy* forem definidas como mostra a Figura 16.5, então 18°C tem uma pertinência no conjunto FRIO de 0,2, no MORNO de 0,8 e no QUENTE de 0. Com base nestes dados, regras podem ser projetadas para disparar ações apropriadas. Por exemplo, a pertinência em FRIO de

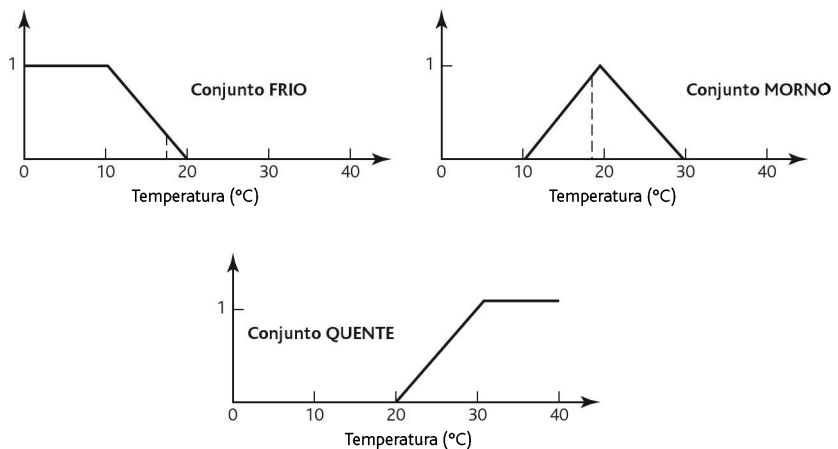


Figura 16.5 Funções de pertinência em conjuntos *fuzzy*.

0,2 pode ter o aquecedor ligado em BAIXO, mas uma pertinência em FRIO de 0,6 pode ter ele ligado em ALTO.

A lógica *fuzzy* está sendo usada atualmente em muitos produtos normalmente encontrados. Por exemplo, máquinas de lavar domésticas que podem detectar o tipo de tecido, de sujeira e definir os ajustes para o ciclo de lavagem adequado.

16.4 Aprendizado

As máquinas que podem aprender e estender sua base de conhecimento têm grandes vantagens em comparação com máquinas sem esta característica. O **aprendizado** pode ser pensado como uma adaptação ao ambiente baseada na experiência. No caso das máquinas, o aprendizado pode ser realizado de formas diferentes.

Um método simples de aprendizado é através da entrada de novos dados acumulados na memória. As máquinas também podem aprender através de dados que elas recebem que são usados para modificar parâmetros na máquina.

Um outro método de aprendizado que pode ser usado é quando o raciocínio é definido em termos de probabilidades que são usadas para atualizar as probabilidades usadas em função do que acontece. Podemos pensar nisto em termos de um exemplo simples. Considere que temos 10 bolas coloridas em uma sacola sendo todas vermelhas exceto uma. Quando retiramos uma bola da sacola, a probabilidade de retirar a bola preta é $1/10$. Se retirarmos uma bola vermelha, então na próxima vez que retirarmos uma bola a probabilidade dela ser preta é $1/9$. A nossa “máquina” pode aprender a partir da primeira bola vermelha retirada ajustando o valor da probabilidade para a bola preta. A regra de Bayes dada na Seção 16.3.1 pode ser usada para atualizar a máquina, passando a ser escrita como

$$p(H|E) = \frac{p(E|H) \times p(H)}{p(E)}$$

em que H é a hipótese de partida e E é o exemplo agora encontrado. Então $p(H|E)$ é a probabilidade da hipótese H ser verdadeira dado que o exemplo E aconteceu, $p(E|H)$ é a probabilidade do exemplo E acontecer dado que a hipótese H é verdadeira, $p|E|$ é a probabilidade do exemplo E acontecer, $p|H|$ é a probabilidade da hipótese H ser verdadeira. Isto permite à máquina atualizar a probabilidade de H cada vez que uma nova informação chega.

Em outro método, uma máquina pode aprender a partir de exemplos. Isto ocorre quando uma máquina generaliza a partir de um conjunto de exemplos. Isto pode ser o resultado de um treinamento com exemplos fornecidos à máquina de modo que ela possa incrementar suas regras ou como uma consequência de eventos com que ela se depara. O reconhecimento de padrão geralmente envolve esta forma de aprendizado. Portanto, dado um exemplo do número 2 em um arranjo de pixels, a máquina pode aprender a reconhecer o número 2. As redes neurais (Seção 16.2.2) também envolvem o aprendizado por exemplos.

Uma máquina pode aprender também fazendo analogias entre um problema resolvido antes e um novo problema.

Resumo

Uma **máquina inteligente** é aquela dotada da habilidade da razão. A **percepção** em um sistema inteligente é a coleção de informações usando sensores e a organização da informação obtida de forma que decisões possam ser tomadas. O **raciocínio** é o processo em que, a partir de situações conhecidas, a máquina lida com situações não conhecidas. Um exemplo de **raciocínio determinístico** é o uso da regra “se-então” (*if-then*). O **raciocínio não determinístico** nos permite fazer previsões baseadas em possibilidades. Com a **lógica fuzzy** as proposições não precisam ser classificadas como verdadeiras ou falsas, mas a veracidade ou a falsidade pode ser ponderada de modo que ela pode ser classificada entre duas opções em uma escala. O **aprendizado** pode ser pensado como uma adaptação ao ambiente com base na experiência.

Problemas

- 16.1 Examine os tipos de moedas do seu país e produza uma tabela de padrão de reconhecimento.
- 16.2 Qual é a probabilidade de (a) termos um seis no lançamento de um dado de seis faces, (b) termos um seis no lançamento de dois dados, (c) retirar uma bola preta de uma sacola que contém nove bolas vermelhas e uma bola preta?
- 16.3 Se a probabilidade de um sistema mecatrônico apresentar uma temperatura alta é 0,01, qual é a probabilidade dele não apresentar uma temperatura alta?
- 16.4 Uma máquina foi monitorada durante 2.000 horas, e neste período o sistema de resfriamento apresentou perdas por 4 horas. Qual é a probabilidade de ocorrerem perdas?
- 16.5 A probabilidade de um sistema de resfriamento de uma máquina ter perda foi determinado como sendo de 0,005, e a probabilidade do sistema apresentar uma temperatura alta é de 0,008. Se houver uma perda, certamente provocará uma temperatura alta. Qual é a probabilidade de uma temperatura alta ser provocada por uma perda no sistema de resfriamento?
- 16.6 A probabilidade de uma máquina funcionar incorretamente em relação a três componentes (A, B e C) é de 0,46. Se a probabilidade do componente A estar ativo é de 0,50 e a probabilidade de ocorrer um funcionamento incorreto com A é de 0,70, qual é a probabilidade de que A seja o responsável pelo funcionamento incorreto?
- 16.7 Proponha regras do tipo “se-então” para um controlador de temperatura que é usado para controlar um aquecedor e tem uma válvula que permite que a água circule pelo radiador do aquecedor central quando ele opera apenas em certo período de tempo.

17

Microprocessadores

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Descrever a estrutura básica de um sistema microprocessado.
- Descrever a arquitetura de microprocessadores comuns e como eles podem ser incorporados em sistemas microprocessados.
- Descrever a estrutura básica de microcontroladores e como os seus registradores podem ser configurados para executar tarefas.
- Explicar como os programas podem ser desenvolvidos usando-se fluxogramas e pseudocódigo.

17.1 Controle

Se analisarmos um simples problema de controle, como, por exemplo, o sequenciamento das luzes vermelha, amarela e verde de um semáforo em um cruzamento, é possível que resolvamos o problema usando um sistema de controle eletrônico que envolve circuitos integrados de lógica combinacional e sequencial. Entretanto, para uma situação mais complexa haverá muito mais variáveis para controlar em uma sequência de controle mais complexa. A solução mais simples neste caso não seria a construção de um sistema baseado em conexões de circuitos integrados de lógica combinacional e sequencial, mas o uso de um microprocessador associado a um software para fazer as “interconexões”.

O sistemas microprocessados a que nos referimos neste livro são para uso em sistemas de controle e são denominados **microprocessadores de sistemas embarcados**. Esta denominação se deve a estes microprocessadores serem dedicados ao controle de funções específicas e de inicialização automática, não necessitando da intervenção humana e contendo os próprios programas de operação. Para as pessoas leigas, não é aparente que o sistema é microprocessado. Portanto, uma máquina de lavar roupa moderna contém um microprocessador, mas tudo o que o operador

necessita fazer é selecionar o tipo de lavagem desejada pressionando os botões apropriados ou girando uma chave rotativa e, em seguida, pressionando o botão de início (*start*).

Neste capítulo fazemos uma abordagem geral da estrutura dos microprocessadores e microcontroladores, nos próximos dois capítulos discutimos a programação e no Capítulo 20, o interfacimento.

17.2 Sistemas microprocessados

Os sistemas que usam microprocessadores têm basicamente três partes: a **unidade central de processamento (CPU – Central Processing Unit)** para reconhecer e executar instruções do programa (esta é a parte que usa o microprocessador), **interfaces de entrada e saída** para estabelecer comunicação entre o microprocessador e o mundo externo (o termo **porta** é usado para interface) e **memória** para armazenar as instruções do programa e os dados. A Figura 17.1 ilustra a configuração geral de um sistema microprocessado.

Os microprocessadores que têm memória e várias configurações de entrada e saída no mesmo chip são denominados **microcontroladores**.

17.2.1 Barramentos

Os sinais digitais se movem de uma seção para outra ao longo de vias denominadas **barramentos**. Um barramento é, no sentido físico, apenas um conjunto de condutores paralelos ao longo dos quais os sinais elétricos podem ser transportados, e são também vias que podem ser compartilhadas por todos os chips no sistema. Isto ocorre porque se existissem vias individuais entre os chips, haveria um grande

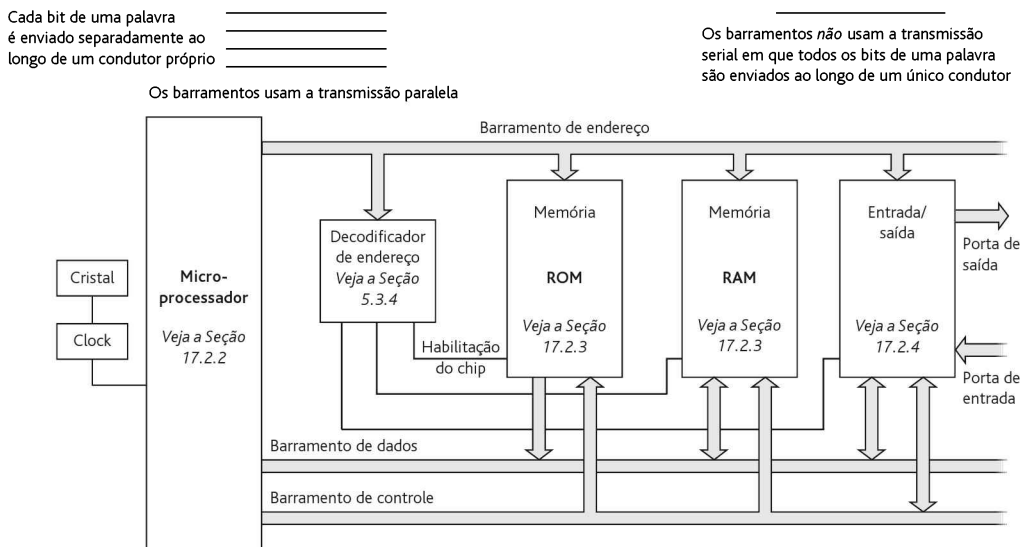


Figura 17.1 Aspecto geral de um sistema microprocessado e seus barramentos.

número de condutores interconectando-os. Com o uso de barramentos de conexões compartilhados, quando um chip coloca dados no barramento, os outros têm que esperar até que a transferência de dados seja finalizada antes que um deles possa colocar os próprios dados no barramento. Tipicamente um barramento tem 16 ou 32 vias em paralelo, de forma que cada uma transporta simultaneamente 1 bit de uma palavra de dados. Isto resulta em uma transmissão mais rápida do que no caso de uma via serial, na qual toda a palavra é enviada em uma sequência de bits ao longo do condutor da via.

Existem três tipos de barramentos em um sistema microprocessado:

1 *Barramento de dados*

Os dados associados com a função de processamento da CPU são transportados pelo **barramento de dados**. Portanto, ele é usado para transportar uma palavra entre CPU e memória ou interfaces de entrada/saída. Cada fio no barramento transporta um sinal binário, ou seja, 0 ou 1. Assim, em um barramento de quatro fios podemos ter, por exemplo, a palavra 1010 transportada, sendo que cada bit é transportado por um fio do barramento como a seguir:

Palavra	Fio do barramento
0 (bit menos significativo)	Primeiro fio do barramento de dados
1	Segundo fio do barramento de dados
0	Terceiro fio do barramento de dados
1 (bit mais significativo)	Quarto fio do barramento de dados

Quanto mais fios de dados houver no barramento, mais extenso será o comprimento da palavra que poderá ser usada. A faixa de valores que um único item de dado pode ter é definido pelo comprimento da palavra. Portanto, com uma palavra de 4 bits, o número de valores é $2^4 = 16$. Assim, se o dado representa, por exemplo, uma temperatura, então a faixa de temperatura possível tem que ser dividida em 16 segmentos se estivermos representando esta faixa por uma palavra de 4 bits. O primeiro microprocessador foi um dispositivo de 4 bits (comprimento da palavra), e ainda é muito usado em dispositivos como brinquedos, máquinas de lavar roupa e controladores de centrais de aquecimento domésticas. Estes microprocessadores foram seguidos de microprocessadores de 8 bits, como por exemplo, o 6800 da Motorola, o 8085A da Intel e o Z80 da Zilog. Atualmente temos disponíveis microprocessadores de 16, 32 e 64 bits; entretanto, os microprocessadores de 8 bits ainda são muito usados em controladores.

2 *Barramento de endereço*

O **barramento de endereço** transporta sinais que indicam onde os dados são encontrados e, assim, selecionam posições específicas da memória ou de portas de entrada/saída. Cada posição de armazenamento dentro de um dispositivo de memória tem uma única identificação, denominada endereço, de modo que o sistema seja capaz de selecionar uma instrução em particular ou um dado na memória. Cada interface de entrada/saída também tem um endereço. Quando um determinado endereço é selecionado pelo código colocado no barramento de endereço, apenas este local da memória tem comunicação com a CPU. Esta é então capaz de se comunicar com apenas uma localização de cada vez. Um

computador com um barramento de dados de 8 bits tem tipicamente um barramento de endereço de 16 bits, ou seja, 16 vias. Este tamanho de barramento de endereço permite que 2^{16} posições sejam endereçadas; 2^{16} é 65.536 posições e é geralmente escrito como 64K, em que o K é igual a 1024. Quanto mais memória pode ser endereçada, maior o volume de dados que pode ser armazenado e maiores e mais sofisticados podem ser os programas.

3 Barramento de controle

Os sinais relacionados às ações de controle são transportados pelo **barramento de controle**. Por exemplo, este barramento é necessário para que o microprocessador informe aos dispositivos de memória se os dados serão lidos de um dispositivo de entrada ou escritos em um dispositivo de saída. O termo leitura (READ) é usado para o recebimento de um sinal e escrita (WRITE), para o envio de um sinal. O barramento de controle também é usado para transportar os sinais de clock do sistema; estes fazem o sincronismo de todas as ações do sistema microprocessado. O clock é gerado por um circuito oscilador controlado por um cristal que produz pulsos em intervalos regulares.

17.2.2 O microprocessador

O microprocessador geralmente é conhecido como unidade central de processamento (CPU). Ele é a parte do sistema que processa os dados, busca instruções na memória, decodifica-as e as executa. A estrutura interna, também denominada arquitetura, depende do tipo de microprocessador. A Figura 17.2 mostra, de forma simplificada, a arquitetura geral de um microprocessador.

As funções a seguir são das partes constituintes de um microprocessador:

1 Unidade lógica e aritmética (ALU – Arithmetic and Logic Unit)

Esta unidade é responsável pela manipulação dos dados.

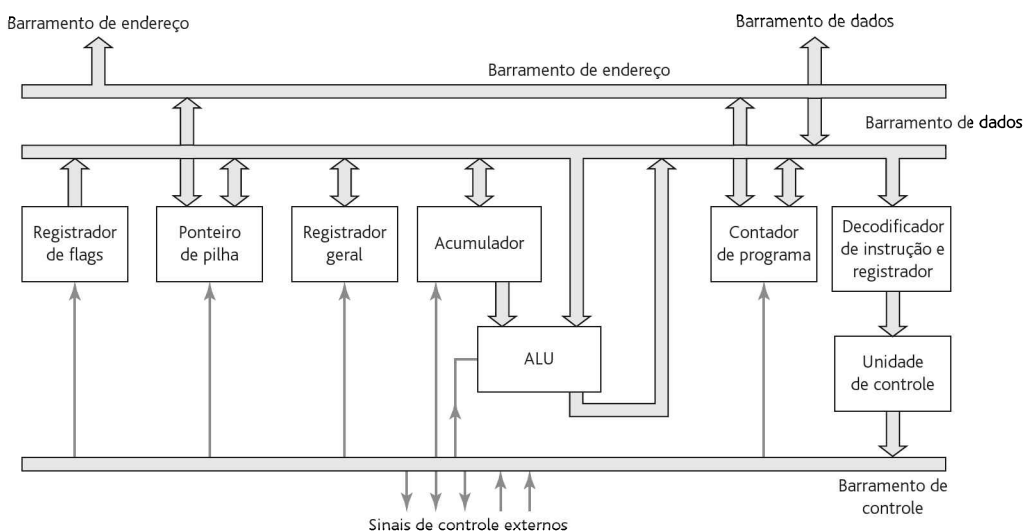


Figura 17.2 Arquitetura interna geral de um microprocessador.

2 Registradores

Os dados internos que a CPU usa em um determinado momento são armazenados temporariamente em grupos de **registradores** enquanto as instruções são executadas. Estes são como posições de memória dentro do microprocessador e são usados para armazenar informações envolvidas na execução do programa. Um microprocessador contém um grupo de registradores, e cada tipo tem uma função diferente.

3 Unidade de controle

A **unidade de controle** determina a temporização e a sequência de operações. Ela gera os sinais de temporização usados para buscar uma instrução do programa na memória e executá-la. O 6800 da Motorola usa um clock com uma frequência máxima de 1 MHz, ou seja, um clock com um período de 1 μ s, e as instruções dele necessitam de dois a doze ciclos de clock. As operações que envolvem o microprocessador são medidas em função do número de ciclos de clock que elas gastam.

Os tamanhos e os tipos de registradores variam de um microprocessador para outro. Os registradores a seguir são os tipos mais comuns:

1 Registrador acumulador

O registrador acumulador (A) armazena temporariamente os dados destinados à unidade lógica e aritmética. Para que a CPU seja capaz de acessar, ou seja, ler, instruções e dados na memória, ela tem que gerar o endereço da palavra desejada na memória usando o barramento de endereço. Quando isto é feito, as instruções ou os dados desejados podem ser lidos pela CPU usando o barramento de dados. Como apenas uma posição de memória pode ser endereçada de cada vez, o armazenamento temporário tem que ser usado quando, por exemplo, números são combinados. Por exemplo, na adição de dois números, um deles é buscado de um endereço e colocado no registrador acumulador enquanto que a CPU busca o outro em uma outra posição de memória. Em seguida, os dois números podem ser processados pela ALU da CPU. O resultado é então transferido de volta para o registrador acumulador. Este registrador mantém temporariamente o dado a ser operado pela ALU e também, após a operação, este registrador armazena o resultado. Ele está portanto envolvido em todas as transferências de dados associadas às operações na ALU.

2 Registrador de status, ou do código de condição ou de flags

Este registrador contém informações referentes ao resultado da última operação realizada pela ALU. Ele contém bits individuais em que cada um tem um significado especial. Estes bits são denominados *flags* (sinalizadores de estado). O status da última operação é indicado por cada *flag*, que pode ser setado ou resetado para indicar um status específico. Por exemplo, eles podem ser usados para indicar se o resultado da última operação foi negativo, nulo, apresentou um *carry* de saída (por exemplo, a soma dos números binários 1010 e 1100 é (1)0110, que pode ser maior do que o tamanho da palavra do microprocessador, havendo assim um *carry* de 1), um *overflow* ocorreu ou o programa está habili-

tado a ser interrompido para permitir que um evento externo ocorra. Os flags a seguir são comuns:

Flag	Setado, ou seja, 1	Resetado, ou seja, 0
Z	O resultado é zero	Resultado diferente de zero
N	O resultado é negativo	Resultado não negativo
C	Carry é gerado	Carry não gerado
V	Ocorre <i>overflow</i>	Não ocorre <i>overflow</i>
I	Interrupção ignorada	Interrupção processada normalmente

Como ilustração, considere o estado dos flags Z, N, C e V para a operação de soma dos números hexadecimais 02 e 06. O resultado é 08. Visto que ele não é zero, então Z é 0. O resultado é positivo, então N é 0. Não há *carry*, então C é 0. O resultado não sinalizado está dentro da faixa de -128 a $+127$ e, assim, não há *overflow* e V é 0. Agora considere os flags quando os números hexadecimais somados são F9 e 08. O resultado é (1)01. Este resultado não é zero, assim Z é 0. Como ele é positivo, N é 0. O resultado não sinalizado tem um *carry* e, assim, C é 1. O resultado não sinalizado está dentro da faixa de -128 a $+127$, e então V é 0.

3 Registrador contador de programa (PC – program counter) ou ponteiro de instruções (IP – instruction pointer)

Este registrador é usado para permitir que a CPU não perca a posição do programa. Este registrador contém o endereço da posição de memória que contém a próxima instrução do programa. Quando cada instrução é executada, o PC é atualizado de forma a conter o endereço da posição da memória onde está a próxima instrução a ser executada. O PC é incrementado cada vez que a CPU executa instruções sequencialmente, a menos que uma instrução, como um salto (JUMP) ou desvio (BRANCH), altere o PC, fazendo ele sair da sequência.

4 Registrador de endereço da memória (MAR – memory address register)

Este contém o endereço do dado. Portanto, por exemplo, na soma de dois números o MAR é carregado com o endereço do primeiro número. O dado neste endereço é movido para o acumulador. O endereço de memória do segundo número é carregado no MAR. O dado deste endereço é somado ao do acumulador. O resultado é armazenado no endereço de memória apontado pelo MAR.

5 Registrador de instrução (IR)

Este registrador armazena uma instrução. Após a busca de uma instrução da memória via barramento de dados, a CPU armazena esta informação no registrador de instrução. Após cada busca, o microprocessador incrementa em uma unidade o registrador PC, fazendo com que o PC aponte para a próxima instrução a ser buscada na memória. A instrução pode então ser decodificada e executada. Esta sequência é conhecida como **ciclo de busca-execução**.

6 Registradores de propósito geral

Estes servem como armazenamento temporário de dados ou endereços e são usados em operações que envolvem transferência entre outros registros.

7 Ponteiro de pilha (SP – stack pointer)

O conteúdo deste registrador representa um endereço que define o topo de uma pilha na RAM. A área desta pilha é uma área especial da memória na qual

o conteúdo do contador de programa pode ser armazenado quando uma sub-rotina, que é parte de uma programa, for executada.

A quantidade e o tipo de registradores dependem do modelo do microprocessador. Por exemplo, o microprocessador 6800 da Motorola (Figura 17.3) tem dois registradores acumuladores, um registrador de status, um registrador de índice, um registrador SP e um registrador PC. O registrador de status tem bits *flag* para sinalizar negativo, zero, *carry*, *overflow*, *half-carry* (*carry* auxiliar) e interrupção. O 6802 da Motorola é similar, mas inclui uma pequena quantidade de RAM e um gerador de clock interno.

O microprocessador 8085A da Intel é um desenvolvimento do processador anterior, o 8080; o 8080 precisa de um gerador de clock externo, ao passo que o 8085A tem este circuito implementado internamente. Os programas escritos para o 8080 podem rodar no 8085A. Este tem seis registradores de propósito geral, B, C, D, E, H e L, um ponteiro de pilha, um contador de programa, um registrador de *flag* e dois registradores temporários. Os registradores de propósito geral podem ser usados como registradores de 8 bits ou aos pares BC, DE e HL como registradores de 16 bits. A Figura 17.4 mostra um diagrama em bloco da arquitetura.

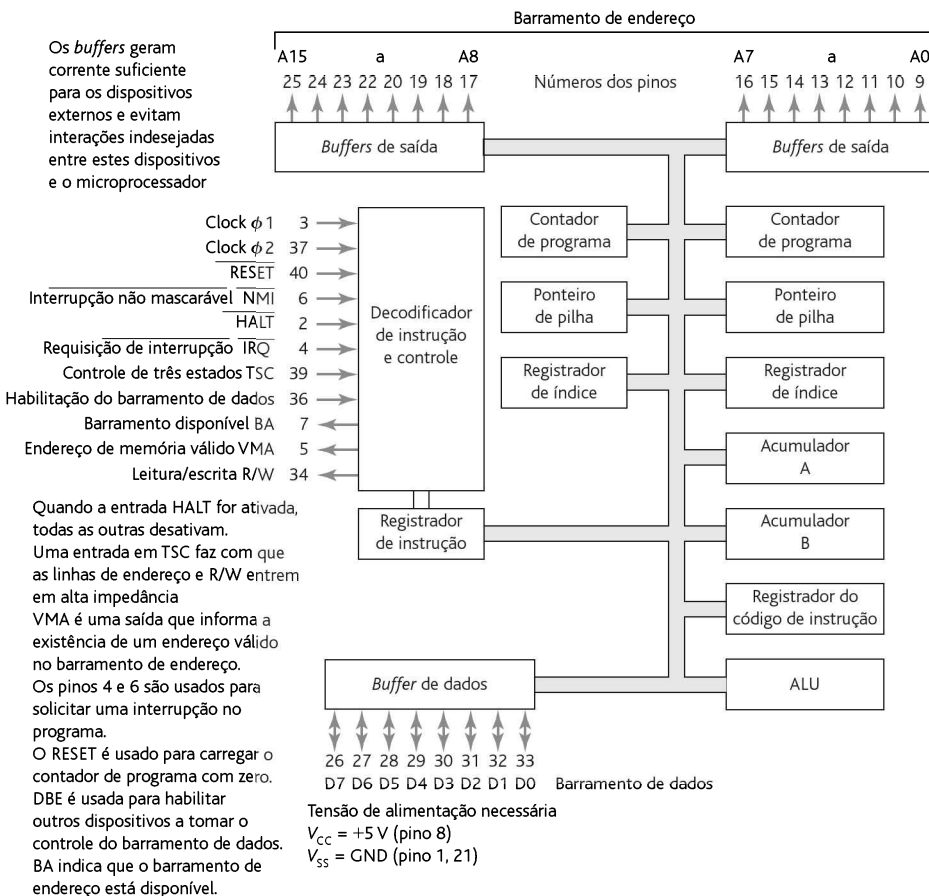


Figura 17.3 Arquitetura do 6800 da Motorola.

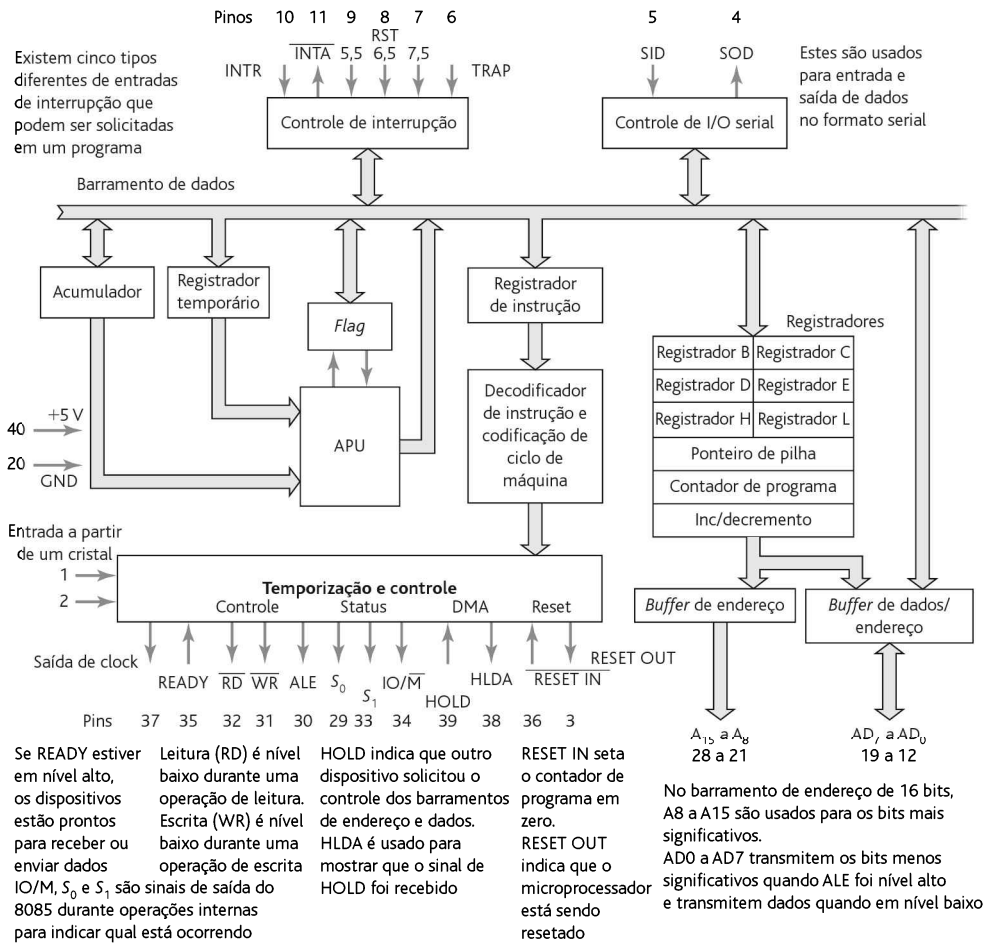


Figura 17.4 Arquitetura do 8085 da Intel.

Como mostram as Figuras 17.3 e 17.4, os microprocessadores têm uma série de entradas e saídas de controle e temporização. Isto provê saídas quando o microprocessador está executando determinadas operações e entradas que influenciam as operações de controle. Além disso, existem entradas relacionadas ao controle de interrupções. Estas entradas foram projetadas para permitir que uma operação do programa seja interrompida como resultado de algum evento externo.

17.2.3 Memória

A unidade de memória em um sistema microprocessado armazena dados binários e é constituída de um ou mais circuitos integrados. Os dados podem ser códigos de instruções do programa ou números que são manipulados.

O tamanho da memória é determinado pelo número de fios no barramento de endereço. Os elementos de memória em uma unidade consistem de um grande número de células de armazenamento em que cada célula é capaz de armazenar um

bit 0 ou 1. Estas células são agrupadas formando posições de memória em que cada posição é capaz de armazenar uma palavra. Para acessar uma palavra armazenada, cada posição é identificada por um único endereço. Portanto, com um barramento de endereço de 4 bits podemos ter 16 endereços diferentes, sendo que cada um é capaz de armazenar, por exemplo, 1 byte, ou seja, um grupo de 8 bits (Figura 17.5).

O tamanho de uma unidade de memória é especificado em termos do número de posições, de armazenamento disponíveis; 1K é $2^{10} = 1024$ posições, e uma memória de 4K tem 4096 posições.

Existem alguns tipos de unidades de memória:

1 ROM

Dispositivo de memória usado para armazenar dados permanentemente conhecido como **memória apenas de leitura (ROM – Read-Only Memory)**. As ROMs são programadas com o conteúdo desejado durante a fabricação do circuito integrado. Nenhum dado pode ser escrito nesta memória quando o CI estiver no computador. Os dados neste tipo de memória podem apenas ser lidos, e ela é usada para armazenar programas que não são alterados como os sistemas operacionais de computadores e programas para aplicações dedicadas de microprocessadores. Os dados desta memória não são perdidos quando a tensão de alimentação é removida. A Figura 17.6(a) mostra as conexões dos pinos de um chip de ROM típico que é capaz de armazenar 1K × 8 bits.

2 PROM

O termo **ROM programável (PROM)** é usado para chips de ROM que podem ser programados pelo usuário. Inicialmente todas as células de memória têm um conexão com fusível que pode representar um 0 armazenado. Um 0 é permanentemente alterado para 1 com o envio de uma corrente através do fusível para abri-lo permanentemente. Uma vez aberta a conexão de fusível, o dado é permanentemente gravado na memória e não pode mais ser alterado.

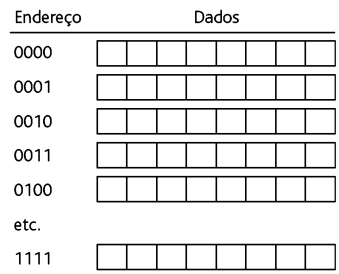


Figura 17.5 Tamanho do barramento de endereço.

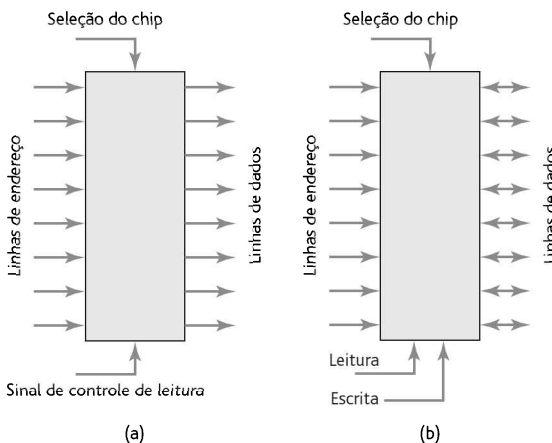


Figura 17.6 (a) Chip de ROM, (b) chip de RAM.

3 EPROM

O termo **ROM programável e apagável** (EPROM) é usado para ROMs que podem ser programadas e ter os conteúdos alterados. Um chip de EPROM típico contém uma série de pequenos circuitos eletrônicos, denominados células, que podem armazenar carga elétrica. O programa é armazenado aplicando-se tensões em determinados pinos do circuito integrado, produzindo um padrão de células com carga ou sem carga. Este padrão se mantém permanentemente no chip até que seja apagado por meio do uso de luz ultravioleta que passa através de uma janela de quartzo na parte superior do dispositivo. Isto faz com que todas as células sejam descarregadas (apagadas). O chip pode então ser reprogramado. A EPROM 2716 da Intel tem 11 pinos de endereços e um único pino de habilitação do chip que é ativado em nível baixo.

4 EEPROM

Uma **PROM apagável eletricamente** (EEPROM) é similar a uma EPROM. O apagamento no primeiro caso é feito por meio de uma tensão relativamente alta em vez do uso de luz ultravioleta.

5 RAM

Os dados temporários, ou seja, aqueles que frequentemente são alterados, são armazenados em uma memória de leitura/escrita denominada **memória de acesso aleatório** (RAM – *Random-Access Memory*). Esta memória pode ser lida ou escrita. A Figura 17.6(b) mostra os pinos de conexões típicos para um chip de RAM de $1K \times 8$ bits. A RAM 6810 da Motorola tem 7 pinos de endereço e 6 pinos de seleção de chip, dos quais 4 são ativos em nível baixo e 2, em nível alto, e todos têm que estar ativos simultaneamente para habilitar a RAM.

Quando uma ROM é usada para armazenar um programa, este está pronto para ser acessado quando o sistema é ligado. Os programas armazenados em ROM são denominados *firmware*. Algum *firmware* sempre deve estar presente. Quando uma RAM é usada para armazenar programas, estes são denominados software. Quando o sistema é ligado, o software pode ser carregado em uma RAM a partir de algum outro equipamento periférico, como um teclado, disco rígido, disco flexível ou CD.

17.2.4 Entrada/saída

A operação de entrada/saída (I/O) é definida como a transferência de dados entre o microprocessador e o mundo externo. O termo **dispositivo periférico** é usado para partes de equipamentos que trocam dados com o sistema microprocessado. Devido a velocidades e características de dispositivos periféricos que podem diferir significativamente das do microprocessador, eles são conectados via chips de interface. Uma das funções principais de um chip de interface é sincronizar as transferências de dados entre o microprocessador e o dispositivo periférico. Nas operações de entrada, o dispositivo de entrada coloca os dados no registrador de dados do chip de interface; isto mantém os dados até que seja feita a leitura deles pelo microprocessador. Nas operações de saída, o microprocessador coloca o dado no registrador até que ele seja lido pelo periférico.

Para o microprocessador receber dados válidos de um dispositivo de entrada, ele precisa estar certo de que o chip de interface armazenou corretamente o dado de entrada. Isto pode ser feito por **consulta** ou por **interrupção**. No caso de consulta, o chip de interface usa um bit de status em 1 para indicar quando o dado é

válido. O microprocessador fica verificando o chip de interface até detectar um bit de status em 1. O problema com este método é que o microprocessador tem que ficar esperando por este bit de status. No caso do método da interrupção, o chip de interface envia um sinal de interrupção para o microprocessador quando ele tem um dado válido; o microprocessador suspende então a execução do programa principal e executa a rotina associada com a interrupção para ler o dado.

17.2.5 Exemplos de sistemas

A Figura 17.7 mostra um exemplo de um sistema microprocessado usando o microprocessador 8085A da Intel. Ele tem um *latch* de endereço, o CI 74LS373, um decodificador de endereço de 3 para 8 linhas, o CI 74LS138, dois chips de RAM 2114 de $1K \times 4$, um chip de EPROM 2716 de $2K \times 8$ e chips de interface de saída 74LS244 e 74LS374.

1 *Latch de endereço*

A saída ALE (habilitação do *latch* de endereço) provê uma saída para o hardware externo para indicar quando as linhas de AD0 a AD7 contêm endereço e quando elas contêm dados. Quando o pino ALE for nível alto, ele ativa o *latch*, e pelas linhas de A0 a A7 passa a parte inferior do endereço onde será armazenado. Portanto, quando ALE retorna para nível baixo, os dados podem ser colocados nestas linhas pelo microprocessador e a referida parte do endereço permanece armazenada no *latch* 74LS373. A parte superior do endereço é fornecida em A8 a A15, permanecendo sempre válida no ciclo, e o endereço completo é dado pela parte inferior proveniente do *latch* e a parte superior a partir do barramento de endereço do microprocessador.

2 *Decodificador de endereço*

O CI 74LS138 é um decodificador de 3 para 8 linhas e gera um nível baixo em uma das oito saídas; a saída selecionada depende dos sinais em suas três linhas de entrada (A, B e C). Antes do CI poder fazer uma seleção como esta, ele tem que ser habilitado através de um nível baixo nas entradas de habilitação 1 e 2 e um nível alto na 3.

3 *EPROM*

Os bits de endereço A11, A12, A13 e A14 são usados para selecionar qual dispositivo será endereçado. Com isto, os bits de endereço de A0 a A10 ficam para endereçamento. Portanto, a EPROM pode ter $2^{11} = 2048$ endereços. Esta é a capacidade de armazenamento da EPROM 2716 da Intel. A EPROM é selecionada sempre que o microprocessador lê um endereço entre 0000 e 07FF e disponibiliza na saída (O0 a O7) o conteúdo de 8 bits no barramento de dados. A linha de habilitação de saída (OE) é conectada à saída do sinal de leitura do microprocessador para garantir que a EPROM seja apenas lida.

4 *RAM*

A figura mostra dois chips de RAM, sendo que cada um é de $1K \times 4$. Juntos eles formam uma memória com uma palavra de tamanho 8 bits. Os dois chips usam os mesmos bits de endereço (A0 a A9) para a seleção da memória, com um chip sendo responsável pelos bits de D0 a D3 e o outro pelos bits de D4 a D7. Com 10 bits de endereço podemos endereçar $2^{10} = 1024$ endereços diferentes (2000 a 23FF). A entrada de habilitação de escrita (WE) é usada pela RAM para determinar se a

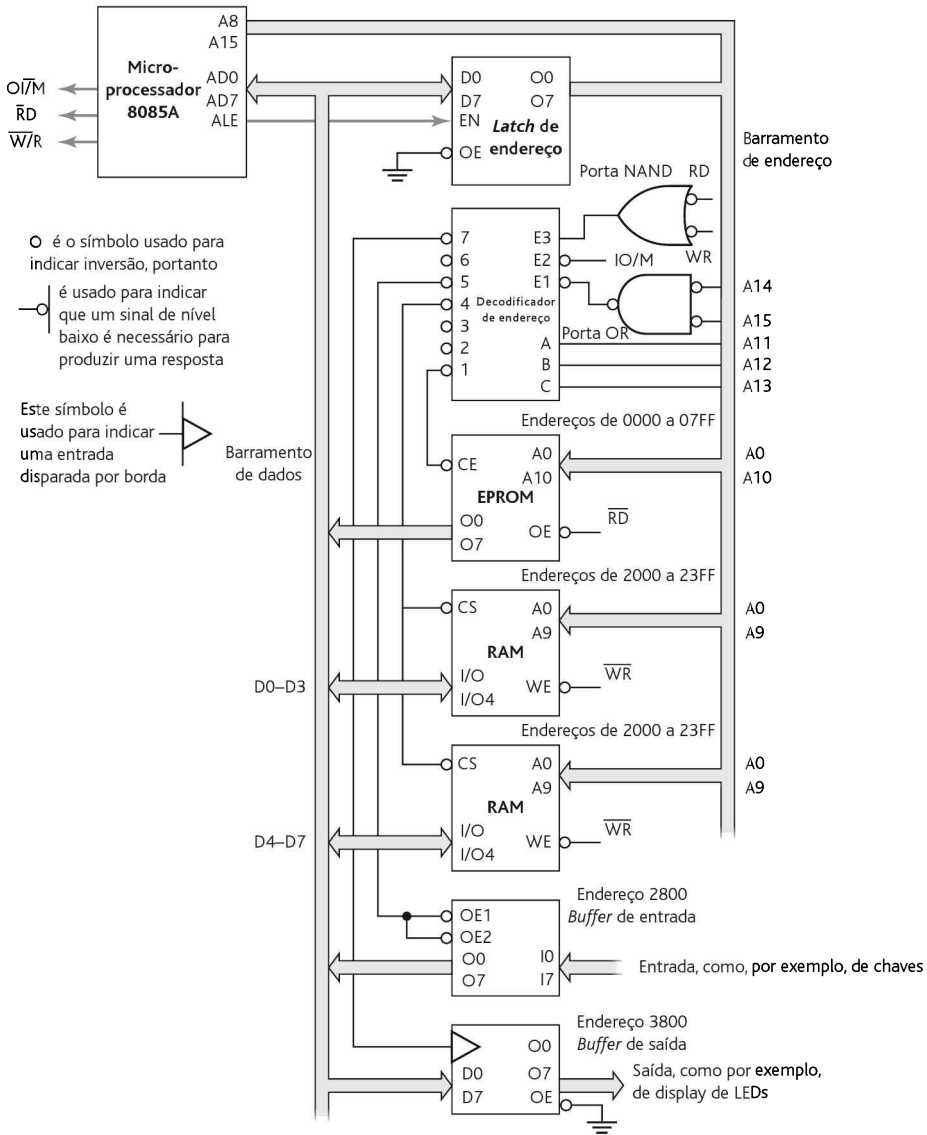


Figura 17.7 Sistema com o 8085A da Intel.

RAM está sendo escrita ou lida. Se esta entrada for nível baixo, então o endereço da RAM está sendo escrito, e se ela for nível alto, a RAM está sendo lida.

5 Buffer de entrada

O *buffer* de entrada 74LS244 é usado para passar o valor binário de entrada para o barramento de dados sempre que OE1 e OE2 estiverem em nível baixo. Isto é obtido para qualquer endereço de 2800 a 2FFF, assim podemos usar o endereço 2800. O *buffer* é usado para garantir que as entradas representem um pequeno efeito de carga no microprocessador.

6 *Latch de saída*

O 74LS374 é um *latch* de saída. Ele armazena a saída do microprocessador de forma que os dispositivos de saída tenham tempo para fazer a leitura enquanto o microprocessador pode buscar outra instrução do programa. O latch de saída pode ser acessado na faixa de endereço de 3800 a 3FFF e, assim, pode ser acessado usando-se o endereço 3800.

A Figura 17.8 mostra um exemplo de um sistema baseado no microprocessador 6800 da Motorola e tendo apenas um chip de RAM, um de ROM e um dispositivo de entrada/saída programável (PIA). Com este sistema não é necessária decodificação de endereço por causa do pequeno número de dispositivos envolvidos. Para entradas/saídas em paralelo é usado o adaptador de interface programável (PIA) (veja a Seção 20.4), e para entradas/saídas em série é usado um adaptador de interface assíncrono (ACIA) (veja a Seção 20.5). Estes dispositivos podem ser programados para lidar com entradas e saídas e prover a função de buffer necessária.

1 *RAM*

As linhas de endereço A14 e A15 são conectadas às entradas de habilitação (\bar{E}) do chip de RAM. Quando estas duas linhas estiverem em nível baixo, o chip de RAM poderá trocar dados com o microprocessador.

2 *ROM*

As linhas de endereço A14 e A15 são conectadas às entradas de habilitação (E) do chip de ROM, e quando os sinais nestas duas linhas estiverem em nível alto, este chip é endereçado.

3 *Entradas/saídas (I/O)*

As linhas de endereço A14 e A15 são conectadas nas entradas de habilitação dos dispositivos PIA e ACIA. Quando o sinal na linha 15 for nível baixo e aquele na linha A14 for alto, as interfaces de entrada/saída (I/O) são endereçadas. Para indicar qual dos dispositivos está sendo habilitado, a linha de endereço A2 é colocada em nível alto para o PIA e a linha de endereço A3 é colocada em nível alto para o ACIA.

17.3 Microcontroladores

Para que um microprocessador constitua um sistema que pode ser usado para controle, são necessários chips adicionais, como, por exemplo, memória para armazenamento de programa e dados e portas de entrada/saída para permitir a comunicação com o mundo externo e receber sinais. O microcontrolador é a integração do microprocessador com memória e interfaces de entrada/saída (I/O) e outros periféricos, como temporizadores, em um único chip. A Figura 17.9 mostra um diagrama em bloco geral de um microcontrolador.

Um microcontrolador geral tem pinos para conexões externas de entradas e saídas, alimentação, clock e sinais de controle. Os pinos para as entradas e saídas são agrupados em unidades denominadas portas. Geralmente estas portas têm oito linhas para serem capazes de transferir palavras de 8 bits. Duas portas podem ser usadas para palavras de 16 bits, uma para transmitir os 8 bits inferiores e a outra

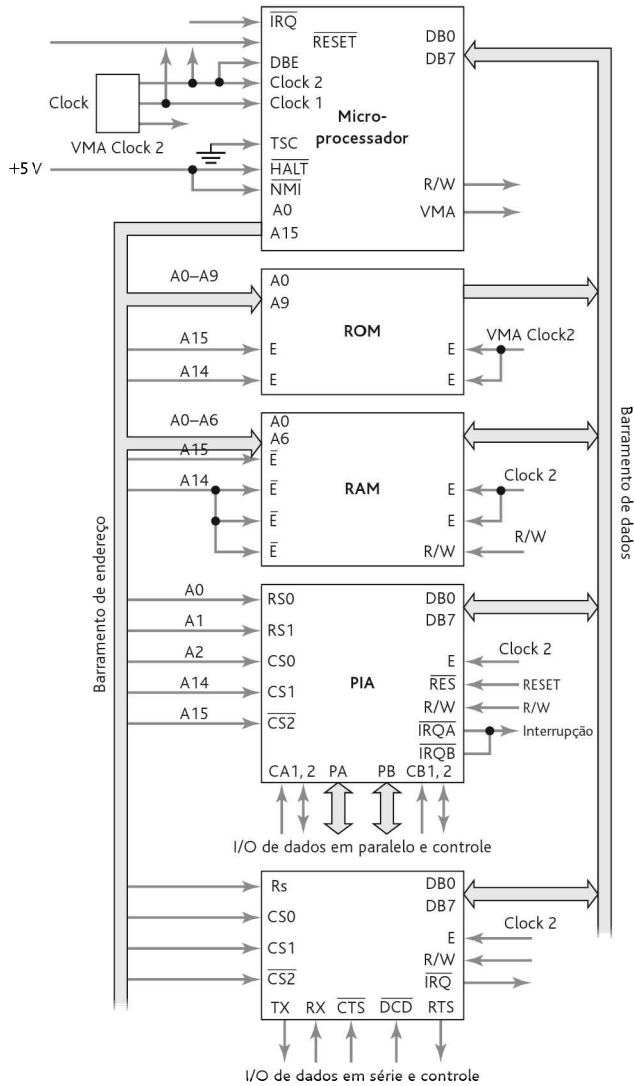


Figura 17.8 Sistema com o M6800.

para os 8 bits superiores. As portas podem ser apenas de entrada, apenas de saída ou programáveis como entrada ou saída.

O 68HC11 da Motorola, o 8051 da Intel e o PIC16C6x/7x da Microchip são exemplos de microcontroladores de 8 bits nos quais a via de dados tem uma extensão de 8 bits. O 68HC16 da Motorola é um exemplo de um microcontrolador de 16 bits e o 68300, também da Motorola, é um microcontrolador de 32 bits. Os microcontroladores têm uma quantidade de RAM e ROM limitada e são muito usados em sistemas de controle embarcados. Um sistema microprocessado com chips separados de memória e I/O é mais adequado para o processamento de informações em um sistema de computador.

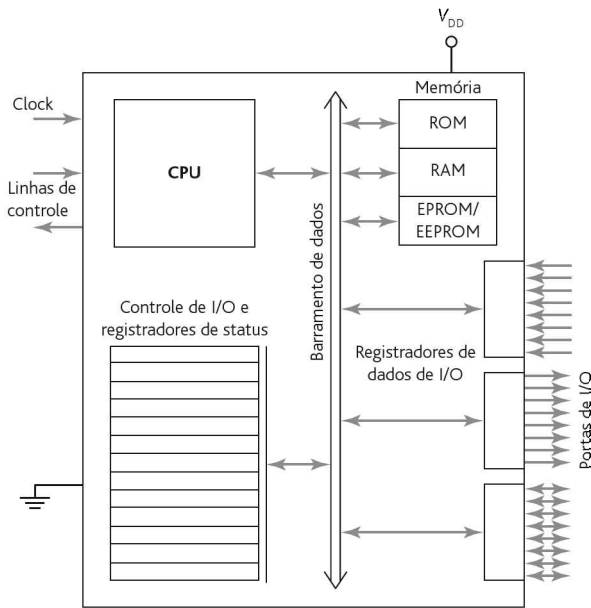


Figura 17.9 Diagrama em bloco de um microcontrolador.

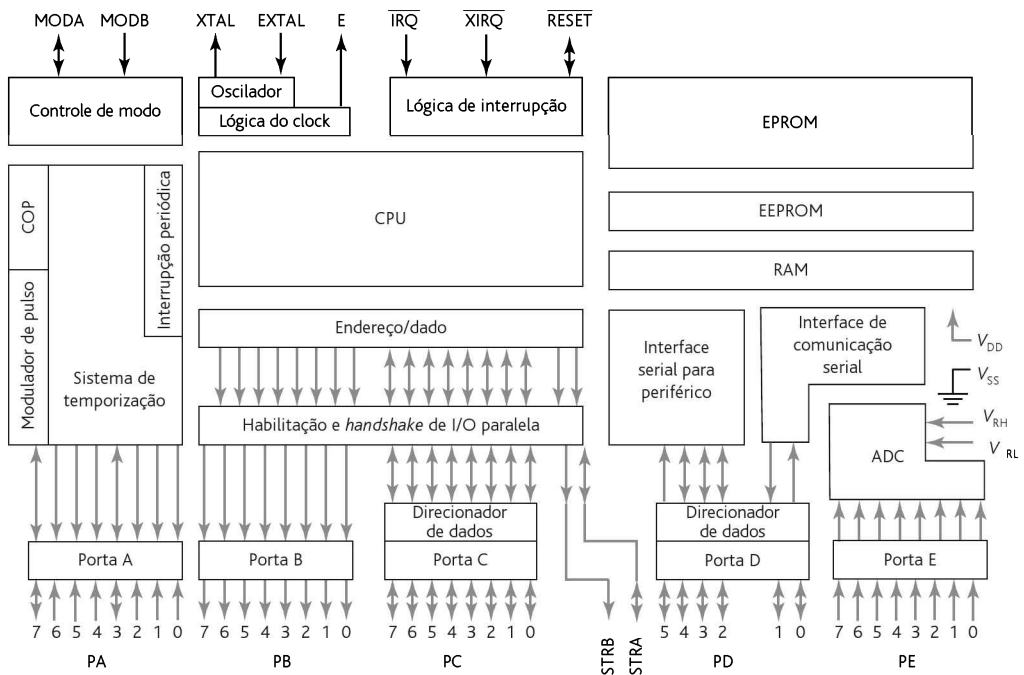


Figura 17.10 Diagrama em bloco do M68HC11.

17.3.1 68HC11 da Motorola

A Motorola oferece duas famílias básicas de microcontroladores de 8 bits, a 68HC05, que é mais barata, e a 68HC11, de alto desempenho. A família M68HC11 (Figura 17.10), baseada no microprocessador 6800, é muito usada em sistemas de controle.

Existem versões nesta família que se diferenciam na incorporação de RAM, ROM, EPROM, EEPROM e configurações de registradores. Por exemplo, a versão 68HC11A8 tem 8K de ROM, 512 bytes de EEPROM, 256 bytes de RAM, um temporizador de 16 bits, uma interface serial síncrona, uma interface serial de comunicação assíncrona sem retorno para zero (NRZ), um conversor analógico-digital de 8 canais e 8 bits cada canal, e 5 portas (A, B, C, D e E).

1 Porta A

A porta A tem apenas 3 linhas de entrada, 4 linhas de saída e 1 linha que serve como entrada ou saída. O registrador de dados da porta A está localizado no endereço \$1000 (Figura 17.11) com um registrador de controle do acumulador de pulsos (Figura 17.12) localizado no endereço \$1026 que é usado para controle das funções de cada bit na porta A. Esta porta também provê acesso ao temporizador interno do microcontrolador, aos bits PAMOD, PEDGE, RTR1 e RTRO que controlam o acumulador de pulsos e o clock.

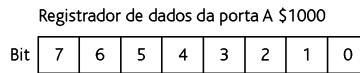


Figura 17.11 Registrador da porta A.

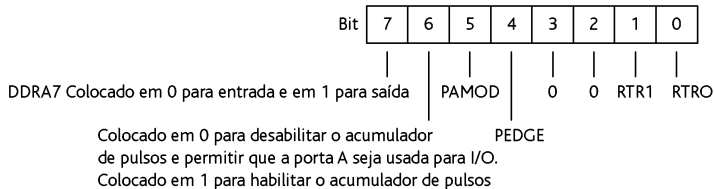


Figura 17.12 Registrador de controle do acumulador de pulsos.

2 Porta B

A porta B é apenas saída e tem 8 linhas de saída (Figura 17.13). Dados de entrada não podem ser colocados nos pinos da porta B. Este registrador de dados está no endereço \$1004, e para a saída de dados tem que haver uma escrita nesta posição de memória.

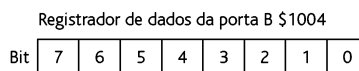


Figura 17.13 Registrador da porta B.

3 Porta C

A porta C pode ser entrada ou saída, sendo os dados escritos ou lidos do registrador no endereço \$1003 (Figura 17.14). A direção é controlada pelo registrador de direção de dados da porta no endereço \$1007. Os 8 bits neste registrador correspondem aos bits individuais na porta C e determinam se as linhas são entradas ou saídas; quando o registrador de direção de dados é colocado em 0, ele é uma entrada, e quando colocado em 1, ele é uma saída. As linhas STRA e STRB (quando se opera no modo de único chip) estão associadas às portas B e C e são usadas para sinais de *handshake* com estas portas. Estas linhas controlam a temporização da transferência de dados. O registrador de controle de I/O paralela PIOC no endereço \$1002 contém bits para definir o modo de controle e a polaridade e as bordas ativas dos sinais de controle.

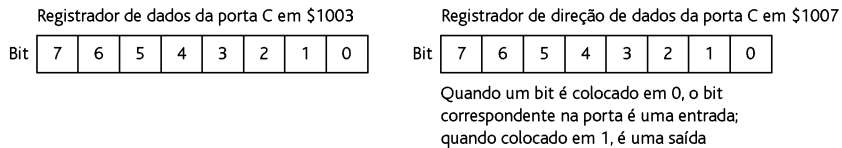


Figura 17.14 Registradores da porta C.

4 Porta D

A porta D contém apenas seis linhas; elas podem ser entrada ou saída e têm um registrador de dados no endereço \$1008 (Figura 17.15); as direções são controladas por um registrador de direção de dados da porta no endereço \$1009, com o bit correspondente sendo colocado em 0 para entrada e em 1 para saída. A porta D também serve como a conexão a dois subsistemas seriais do microcontrolador. A interface de comunicação serial é um sistema assíncrono que prevê comunicação serial que é compatível com modems e terminais. A interface serial para periférico é um sistema síncrono de alta velocidade que foi projetado para estabelecer comunicação entre o microcontrolador e componentes periféricos que podem operar nestas taxas.

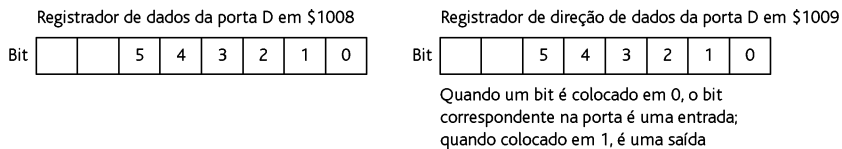


Figura 17.15 Registradores da porta D.

5 Porta E

A porta E é uma porta apenas de entrada de 8 bits (Figura 17.16) que pode ser usada como uma porta de entrada de propósito geral ou para entradas analógicas para o conversor analógico-digital interno. As duas entradas V_{RH} e V_{RL} proveem tensões de referência para o ADC. O registrador de dados da porta E está no endereço \$1002.

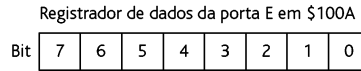


Figura 17.16 Registradores da porta E.

O 68HC11 tem um conversor analógico-digital (A/D) interno; os bits 0, 1, 2, 3, 4, 5, 6 e 7 são os pinos de entrada analógica. As duas linhas V_{RH} e V_{LH} proveem as tensões de referência usadas pelo ADC; a tensão de referência alta não deve ser menor do que V_{DD} , ou seja, 5 V, e a tensão de referência baixa não deve ser menor do que V_{SS} , ou seja, 0 V. O ADC tem que ser habilitado antes de ser usado. Isto é feito setando-se o bit de controle de ativação do A/D (ADPU) no registrador OPTION (Figura 17.17), que é o bit 7. O bit 6 seleciona a fonte de clock do ADC. É necessário um atraso de pelo menos 100 μ s após a ativação para permitir que o sistema estabilize.

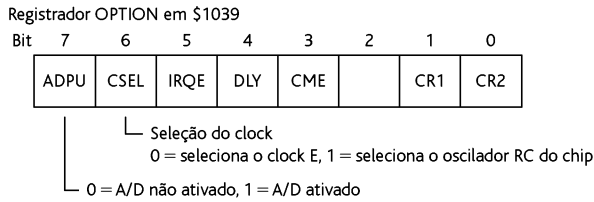


Figura 17.17 Registrador OPTION.

A conversão analógico-digital inicia com a escrita no registrador de controle/status do A/D (ADCTL) após os tempo de atraso para estabilização (Figura 17.18). Isto envolve a seleção de canais e modos de operação. A conversão começa após um ciclo do clock. Por exemplo, se o modo único canal for selecionado fazendo-se $MULT = 0$, então ocorrerão 4 conversões sucessivas do canal selecionado do A/D pelos bits de CD a CA. Os resultados das conversões são colocados nos registradores de resultado do A/D, ADR1 a ADR4.

6 Modos

MODA e MODB são dois pinos que podem ser usados para forçar o microcontrolador em um dos quatro modos quando o sistema é ligado, sendo os modos: inicialização especial, teste especial, único chip e expandido.

MODB	MODA	Modo
0	1	Inicialização especial
0	1	Teste especial
1	0	Único <i>chip</i>
1	1	Expandido

No modo único chip, o microcontrolador opera sozinho, com a exceção dos circuitos de clock externo e reset. Com este modo, o microcontrolador pode

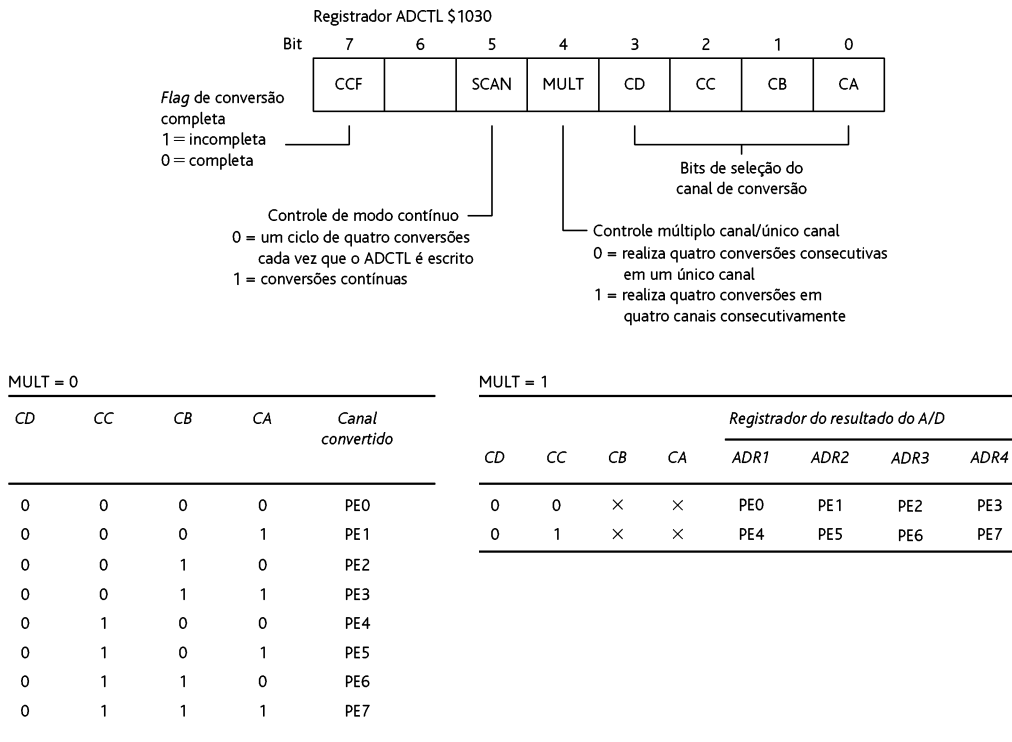


Figura 17.18 Registrador ADCTL.

não ter recursos suficientes, como, por exemplo, memória, para algumas aplicações, e neste caso pode ser usado o modo expandido para aumentar o número de endereços. As portas B e C proverão os barramentos de dados, endereço e controle. A porta B funciona como os pinos dos 8 bits superiores de endereço e a porta C, como os pinos multiplexados de dados e a parte inferior de endereço. O modo de inicialização permite ao fabricante carregar programas especiais em uma ROM especial para um usuário do M68HC11. Quando o microcontrolador é colocado neste modo, o programa especial é carregado. O modo de teste especial é usado principalmente durante o teste de produção interno da Motorola.

Após o modo ser selecionado, o pino MODA se torna um pino que pode ser usado para determinar se uma instrução começa a ser executada. O pino MODB tem outra função, em que a RAM interna do chip pode ser alimentada quando a alimentação regular é removida.

7 Pinos do oscilador

Os pinos XTAL e EXTAL do sistema oscilador são as conexões que acessam o oscilador interno. A Figura 17.19 mostra um circuito externo que pode ser usado. O pino E é o clock do barramento e tem uma frequência de um quarto da frequência do oscilador, e pode ser usado para sincronismo de eventos externos.

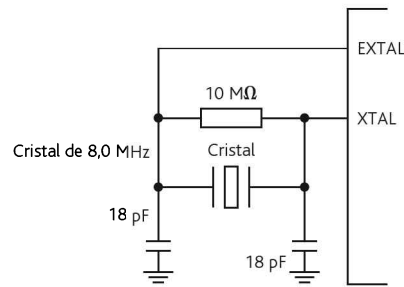


Figura 17.19 Saída do oscilador.

8 Controlador de interrupção

O controlador de interrupção habilita o microcontrolador a interromper um programa (veja a Seção 20.3.3). Uma interrupção é um evento que necessita que a CPU pare a execução do programa normal e execute algum serviço relacionado ao evento. As linhas IRQ e XIRQ são entradas externas dos sinais de interrupção. A entrada RESET é para resetar o microcontrolador e permitir a inicialização do sistema. O estado deste pino pode ser definido internamente ou externamente. Quando uma condição de reset é detectada, o sinal no pino é colocado em nível baixo durante quatro ciclos do clock. Se após mais dois ciclos ele ainda estiver em nível baixo, então é considerado que ocorreu um reset externo. Se uma transição positiva é detectada na entrada de alimentação VDD, ocorreu um reset ao energizar. Isto gera um atraso de tempo de 4064 ciclos. Se o pino de reset estiver em nível baixo no final do tempo de atraso na energização, o microcontrolador permanece na condição de reset até que passe para nível alto.

9 Temporizador

O M68HC11 contém um sistema temporizador. Ele tem um contador autônomo (*free-running*), cinco saídas com função de comparação, a capacidade de capturar o tempo quando um evento externo ocorre, uma interrupção periódica de tempo real e um contador, denominado acumulador de pulsos, para eventos externos. O contador autônomo, denominado TCNT, é um contador de 16 bits que inicia a contagem em 0000 quando a CPU é resetada e conta continuamente, não podendo ser resetado pelo programa. O seu valor pode ser lido em qualquer momento. A fonte de pulsos do contador é o clock do barramento, e sua saída pode passar por um *prescaler* (divisor de frequência programável), cujo fator é determinado pelos bits PR0 e PR1 no registrador TMSK2 no endereço \$1024 (Figura 17.20).

As funções de comparação de saída possibilitam temporizações, ou seja contagem de tempo, especificadas para ocorrerem quando a contagem pré-estabelecida for alcançada. O sistema de captura de entrada captura o valor do contador quando uma entrada ocorre e, assim, o instante exato no qual a entrada ocorre é capturado. O acumulador de pulsos pode ser configurado para operar como um contador de eventos e contar os pulsos de clock externos, ou como um acumulador com janela de tempo e armazenar o número de pulsos que ocorreram em um determinado intervalo de tempo como resulta-

Registrador de interrupção do
temporizador 2 no endereço \$1024

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---

PR1 PRO

Fatores do *prescaler*

PR1	PRO	Fatores do <i>prescaler</i>	<u>Uma contagem</u>	
			<u>Frequência do barramento</u>	
			2 MHz	1 MHz
0	0	1	0,5 ms	1 ms
0	1	4	2 ms	4 ms
1	0	8	4 ms	8 ms
1	1	16	8 ms	16 ms

Figura 17.20 Registrador TMSK2.

do de um contador que é habilitado e, algum tempo depois, desabilitado. O registrador de controle do acumulador de pulsos PACTL (Figura 17.12) no endereço \$1026 é usado para selecionar o modo de operação. O bit PAEN é colocado em 0 para desabilitar o acumulador de pulsos e em 1 para habilitá-lo. O bit PAMOD é colocado em 0 para o modo contador de eventos e em 1 para o modo de janela de tempo. O bit PEDGE é colocado em 0 para o acumulador de pulsos responder a uma borda de descida, ou negativa, quando estiver no modo de contagem de eventos, e colocado em 1 para responder a uma borda de subida, ou positiva, neste modo. No modo de janela de tempo, o bit PEDGE em 0 faz com que o contador seja desabilitado quando o bit 7 da porta A for 0 e que acumule pulsos quando for 1, e quando o bit PEDGE é colocado em 1 neste modo, o contador é desabilitado quando o bit 7 da porta A for 1 e habilitado quando for 0.

10 COP

Outro temporizador é o COP, com a função de indicar que o computador está operando adequadamente. Este é um temporizador que, ao terminar a temporização, reseta o sistema se uma operação não for concluída em um tempo considerado razoável (veja a Seção 23.2). Este temporizador é normalmente denominado **temporizador**.

11 PWM

A modulação por largura de pulso (PWM – *Pulse Width Modulation*) é usada para o controle de velocidade de motores CC (veja as Seções 3.6 e 9.5.3) usando um sinal de onda quadrada em que o seu valor médio varia através da variação do tempo em que o sinal está ligado. Uma onda quadrada pode ser gerada por um microcontrolador fazendo-se com que uma saída seja ativada a cada semiciclo. Entretanto, algumas versões do M68HC11 têm um módulo PWM que, após ser inicializado e habilitado, gera formas de onda PWM.

Conforme visto até aqui, antes que um microcontrolador possa ser usado é necessário configurá-lo, ou seja, definir os bits nos registros apropriados, de forma que ele opere conforme desejado.

17.3.2 8051 da Intel

Uma outra família de microcontroladores é a do 8051 da Intel. A Figura 17.21 mostra os pinos de conexão e a arquitetura interna. O 8051 tem 4 portas de entrada/saída (I/O) denominadas 0, 1, 2 e 3. As portas 0, 2 e 3 também têm funções alternativas. A versão 8051AH tem 4K de ROM, 128 bytes de RAM, 2 temporizadores e controle de interrupção para cinco fontes de interrupção.

1 Portas de I/O

A porta 0 está no endereço 80H, a 1 no endereço 90H, a 2 no endereço A0H e a 3 no endereço B0H (observe o uso da letra H, ou h, após o endereço para indicar que o número é em hexa). Quando uma porta é usada como saída, o dado é colocado no registro de função especial correspondente. Quando uma porta é usada como entrada, o valor FFH tem que ser escrito primeiro nela. Todas as portas são bit-endereçáveis. Portanto, podemos usar, por exemplo, apenas o bit 6 na porta 0 para ligar ou desligar um motor e o bit 7 para ligar ou desligar uma bomba hidráulica.

A porta 0 pode ser usada como uma porta de entrada ou saída. Alternativamente, ela pode ser usada como barramento multiplexado de endereço/dado para acessar memória externa. A porta 1 pode ser usada como entrada ou saída. A porta 2 pode ser usada como entrada ou saída. Alternativamente, ela pode ser usada para a parte superior do barramento de endereço para acesso à memória externa. A porta 3 pode ser usada como entrada ou saída. Alternativamente,

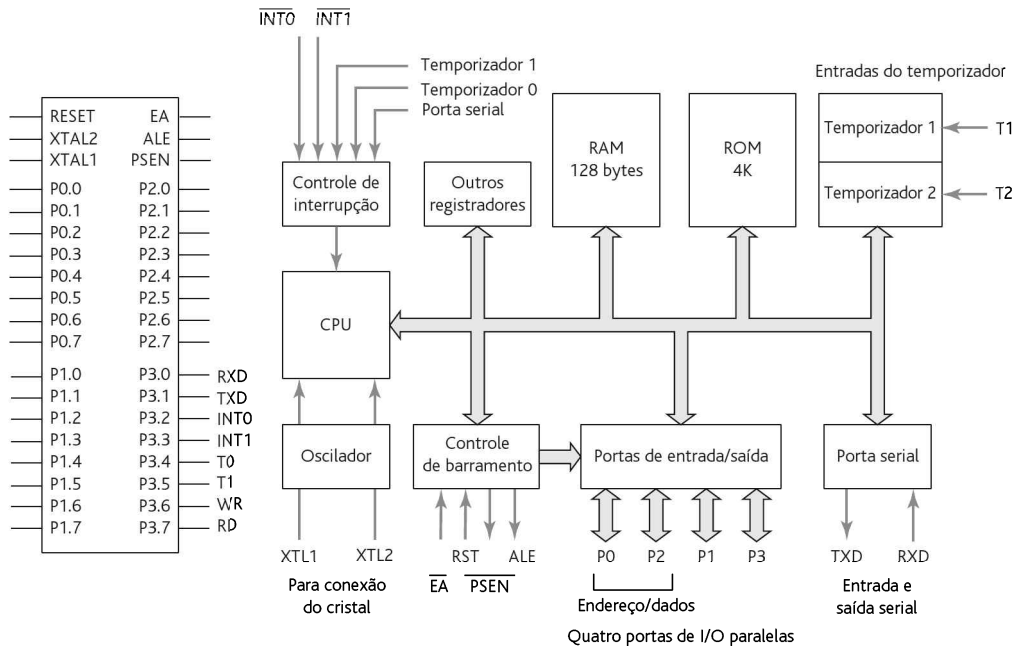


Figura 17.21 8051 da Intel.

ela pode ser usada como uma porta de entrada/saída de finalidade especial. As funções alternativas da porta 3 incluem interrupção e saídas do temporizador, entrada e saída da porta serial e sinais de controle para interfaceamento com memória externa. RXD é a porta de entrada serial, TXD é a porta de saída serial, INT0 é a interrupção externa 0, INT1 é a interrupção externa 1, T0 é a entrada externa do temporizador/contador 0, T1 é a entrada externa do temporizador/contador 1, WR é a autorização de escrita para a memória externa e RD é a autorização de leitura para a memória externa. O termo **habilitação** (*strobe*) descreve uma conexão usada para habilitar ou desabilitar uma função particular. A porta 0 pode ser usada como entrada ou saída. Alternativamente, ela pode ser usada para acessar memória externa.

2 ALE

O pino de habilitação do *latch* de endereço (ALE) fornece um pulso de saída para o *latch* (memorização) do byte inferior de endereço durante um acesso à memória externa. Isto permite que um endereçamento de 16 bits seja usado. A Figura 17.22 ilustra isto.

3 PSEN

O pino PSEN é o sinal de leitura para a memória de programa externa e é ativado em nível baixo. Ele é conectado no pino de habilitação de saída da ROM ou EPROM externa.

4 EA

O pino de acesso externo (EA) é levado ao nível baixo pelo microprocessador para acesso apenas ao código de programa externo; quando em nível alto, ele automaticamente acessa a memória interna ou externa, dependendo do código de endereço. Portanto, quando o 8051 é resetado, o contador de programa (PC) inicia em \$0000 e aponta para a primeira instrução do programa na memória de código interna, a menos que o pino EA esteja em nível baixo. Neste caso, a CPU gera um nível baixo em PSEN para habilitar a memória de código externa. Este pino também é usado em um microcontrolador com EPROM para receber a tensão de programação da EPROM.

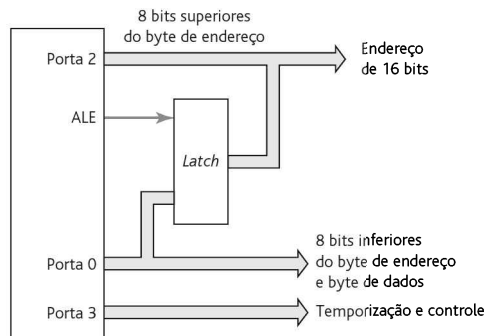


Figura 17.22 Uso do ALE.

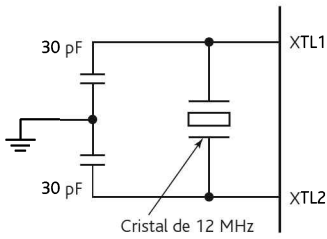


Figura 17.23 Conexão do cristal.

5 XTAL1, XTAL2

Estes são os pinos de conexão do cristal ou do oscilador externo. A Figura 17.23 ilustra como eles são usados com um cristal. O cristal mais usado tem frequência de 12 MHz.

6 RESET

Um sinal de nível alto neste pino com uma duração de pelo menos dois ciclos de máquina reseta o microcontrolador, ou seja, coloca-o na condição de inicialização do sistema.

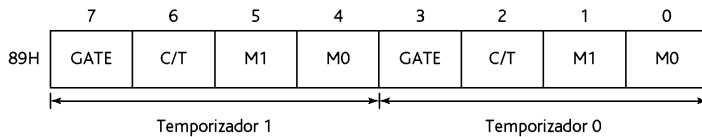
7 Entrada/saída serial

A escrita no buffer de dados serial SBUF, no endereço 99H, carrega o dado para transmissão; a leitura do SBUF acessa o dado recebido. O registrador, bit-endereçável, de controle da porta serial, SCON, no endereço 98H é usado para controlar os modos de operação.

8 Temporização

O registrador de modo do temporizador, TMOD, situa-se no endereço 89H e é usado para definir os modos de operação dos temporizadores 0 e 1 (Figura 17.24). Ele é carregado como uma unidade, não sendo bit-endereçável. O registrador de controle do temporizador, TCON (Figura 17.25), contém os bits de status e controle dos temporizadores 0 e 1. Os 4 bits superiores são usados para ligar e desligar os temporizadores ou sinalizar o *overflow* do temporizador. Os 4 bits inferiores não têm nada a ver com os temporizadores e são usados para detectar e iniciar interrupções externas.

A fonte dos bits contados por cada temporizador é determinada pelo bit C/T; se este bit for nível baixo, a fonte é o clock do sistema dividido por 12, caso contrário, se este bit for nível alto, a fonte é externa. Os temporizadores podem ser disparados carregando-se TR0 ou TR1 com 1 e paralisados fazendo-se este bit igual a 0. Um outro método de controlar o temporizador é carregando GATE com 1, permitindo assim que um temporizador seja controlado pelo pino INT0



Gate: 0 = temporizador funciona quando TR0/TR1 for setado
 1 = temporizador funciona apenas quando INTO/INT1 for nível alto juntamente com TR0/TR1

C/T: seleção contador/temporizador
 0 = entrada a partir do clock do sistema, 1 = entrada a partir de TX0/TX1

M0 e M1 define o modo

M1	M0	Modo
0	0	0 Contador de 13 bits – 5 bits inferiores de TL0 e todos os 8 bits de TH0
0	1	1 Contador de 16 bits
1	0	2 Temporizador/contador de 8 bits autorrecarregável
1	1	3 TL0 é um temporizador/contador de 8 bits controlado pelos bits de controle do temporizador 0. TH0 é um temporizador/contador de 8 bits controlado pelos bits de controle do temporizador 1. Temporizador 1 para de funcionar

Figura 17.24 Registrador TMOD.

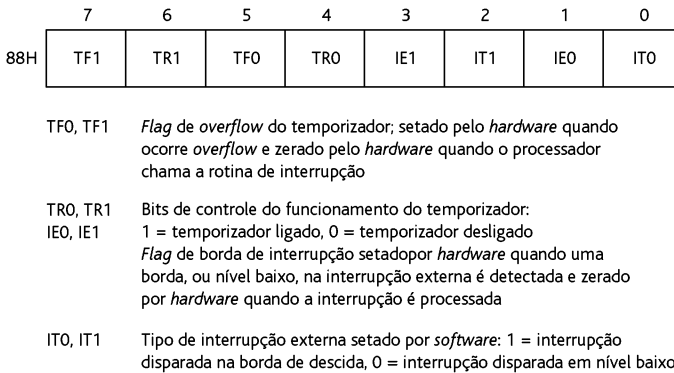


Figura 17.25 Registrador TCON.

ou INT1 do microcontrolador quando for para 1. Desta forma, um dispositivo externo conectado em um destes pinos pode controlar (*on/off*) o contador.

9 Interrupções

As interrupções forçam o programa a chamar uma sub-rotina localizada em um endereço específico na memória; elas são habilitadas através do registrador de habilitação de interrupção (IE) no endereço A8H (Figura 17.26).

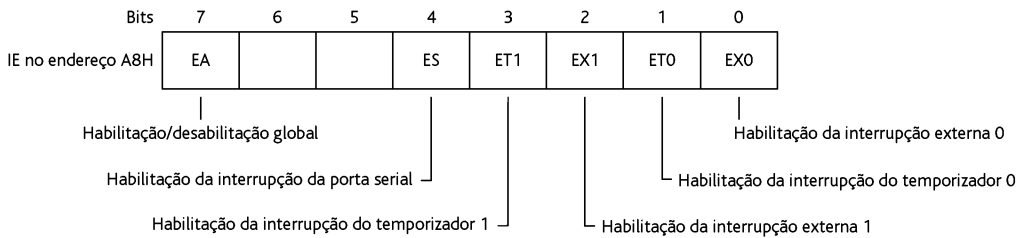


Figura 17.26 Registrador IE.

O termo **registradores de função especial** é usado para registradores de controle de entrada/saída (Figura 17.27), como o registrador IE visto antes, e estes estão localizados entre os endereços 80H e FFH. O acumulador A (ACC) é o registrador principal usado para operações com dados; o registrador B é usado para multiplicação e divisão. P0, P1, P2 e P3 são os registradores *latch* para as portas 0, 1, 2 e 3.

17.3.3 Microcontroladores da Microchip™

Uma outra família de microcontroladores de 8 bits bastante usada é fornecida pela Microchip™. O termo PIC (*Peripheral Interface Controller*) é usado para estes microcontroladores de pastilha única. Eles usam a chamada **arquitetura Harvard**. Com esta

8D	TH1	F0	B
8C	TH0	E0	ACC
8B	TL1	D0	PSW
8A	TL0	B8	IP
89	TMOD	B0	P3
88	TCON	A8	IE
87	PCON	A0	P2
83	DPH	99	SBUF
82	DPL	98	SCON
81	SP	90	P1
80	P0		

Figura 17.27 Registradores.

arquitetura, as instruções são buscadas em uma memória de programa usando barramentos que são distintos dos usados para acessar variáveis (Figura 17.28). Nos outros microcontroladores discutidos neste capítulo, não são usados barramentos separados e, portanto, a busca do programa tem que esperar que as operações de leitura/escrita e de entrada/saída sejam completadas antes que a próxima instrução possa ser buscada na memória. Com a arquitetura Harvard, as instruções podem ser buscadas a cada ciclo sem espera, sendo a execução realizada no próximo ciclo de busca. Esta arquitetura permite velocidades de execução maiores para uma dada frequência de clock. A Figura 17.29 mostra os pinos de conexão para a versão PIC16C74A e 16F84 do microcontrolador, e a Figura 17.30 mostra a forma básica da arquitetura.

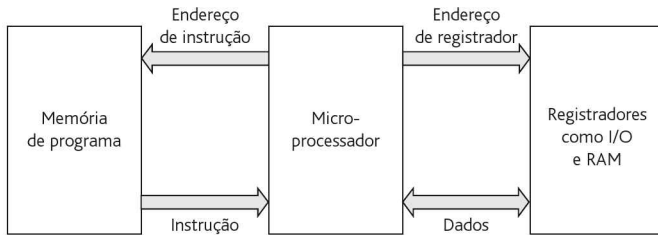


Figura 17.28 Arquitetura Harvard.

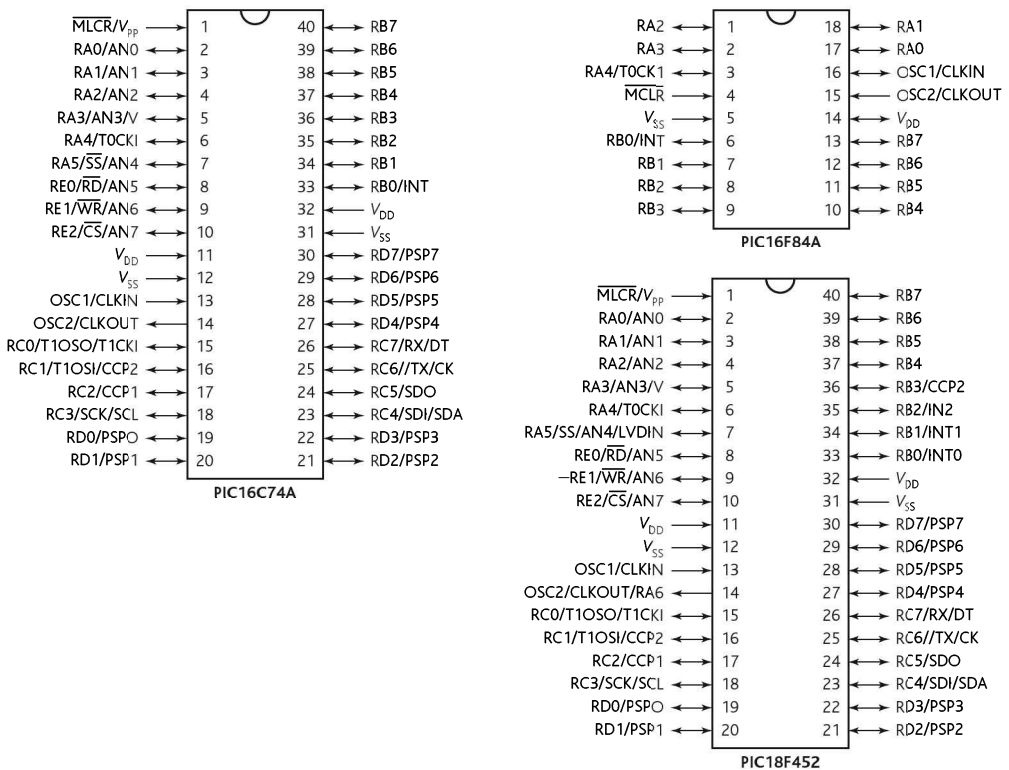


Figura 17.29 Diagramas de pinos de três versões de PIC.

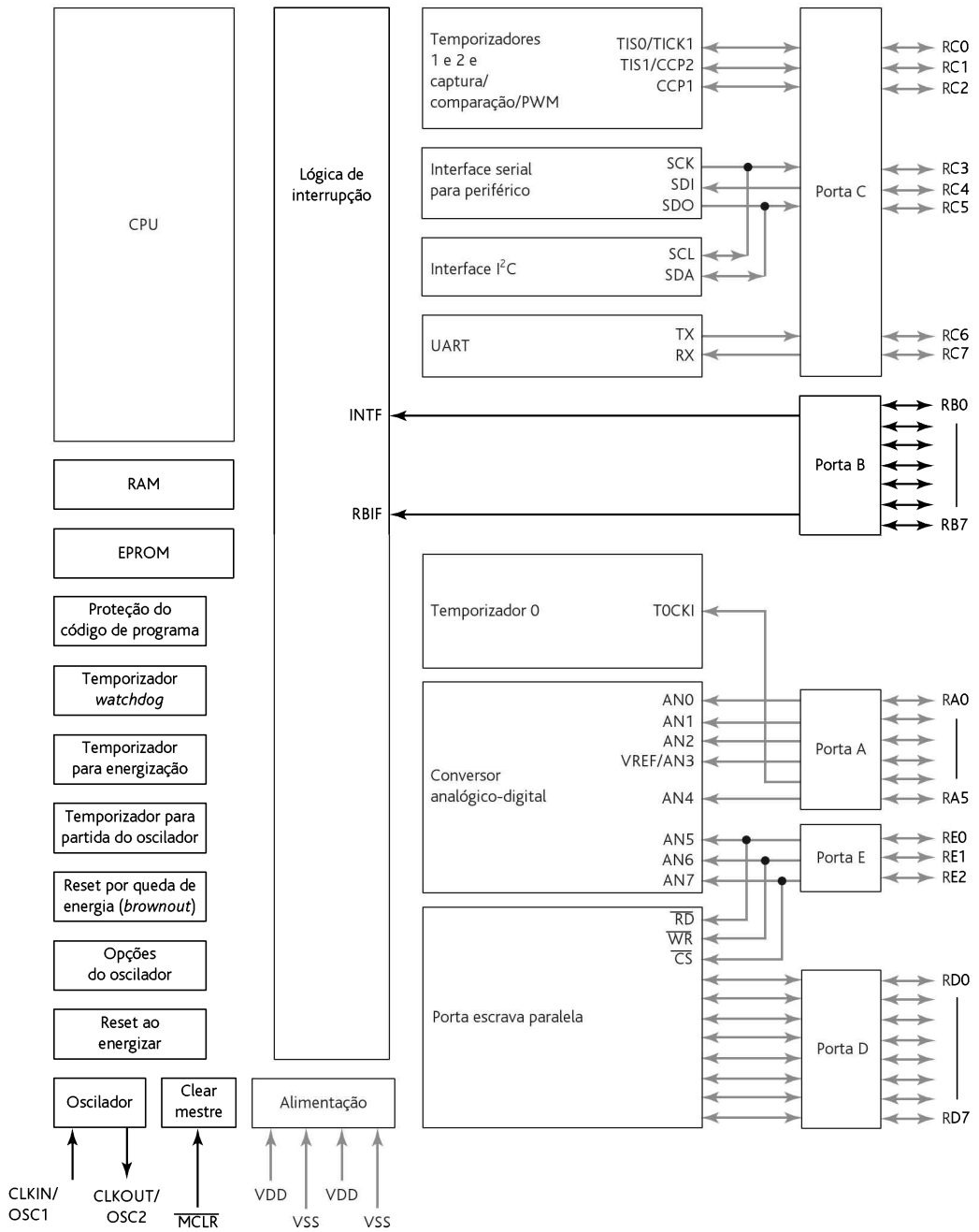


Figura 17.30 PIC 16C74/74A.

As características básicas do microcontrolador 16C74, que são similares às funções de outros PIC, são:

1 Portas de entrada/saída

Os pinos 2, 3, 4, 5, 6 e 7 são para a porta A de entrada/saída bidirecional. Assim como outras portas bidirecionais, os sinais são lidos ou escritos via registradores da porta. A direção dos sinais é controlada pelo registrador de direção TRIS. Existe um registrador TRIS para cada porta. TRIS é feito igual a 1 para leitura e igual a 0 para escrita (Figura 17.31).

Os pinos 2, 3, 4 e 5 também podem ser usados como entradas analógicas, o pino 6 como entrada de clock para o temporizador 0; o pino 7 também pode ser a seleção de escravo para a porta serial síncrona (veja mais adiante nesta seção).

Os pinos 33, 34, 35, 36, 37, 38, 39 e 40 são para a porta B de entrada/saída bidirecional; a direção dos sinais é controlada pelo registrador de direção TRIS correspondente. O pino 33 também pode ser o pino de interrupção externa. Os pinos 37, 38, 39 e 40 também podem ser a interrupção por variação nos pinos. O pino 39 também pode ser o clock de programação serial e o pino 40, o dado de programação serial.

Os pinos 15, 16, 17, 18, 23, 24, 25 e 26 são para a porta C de entrada/saída bidirecional; a direção dos sinais é controlada pelo correspondente registrador TRIS. O pino 15 também pode ser a saída do temporizador 1 ou a entrada do clock do temporizador 1. O pino 16 também pode ser a entrada do oscilador do temporizador 1 ou a entrada de captura 2/saída de comparação 2/saída do PWM.

Os pinos 19, 20, 21, 22, 27, 28, 29 e 30 são para a porta D de entrada/saída bidirecional; a direção dos sinais é controlada pelo registrador de direção TRIS correspondente.

Os pinos 8, 9 e 10 são para a porta E de entrada/saída bidirecional; a direção dos sinais é controlada pelo registrador de direção TRIS correspondente. O pino 8 também pode ser o controle de leitura para a porta escrava paralela ou a entrada analógica 5. A porta escrava paralela é uma característica que facilita o projeto de circuitos de interface para computadores pessoais; neste caso, os pinos das portas D e E são dedicados a esta operação.

2 Entradas analógicas

Os pinos 2, 3, 4, 5 e 7 da porta A e os pinos 8, 9 e 10 da porta E também podem ser usados para entradas analógicas para o ADC interno. Os registradores usados para entradas analógicas ADCON1 e TRISA para a porta A (TRISE para a porta E) têm que ser inicializados para selecionar a tensão de referência a ser usada para a conversão e selecionar os canais como entradas. Então, ADCON0 tem que ser inicializado usando-se os parâmetros indicados a seguir:

	7	6	5	4	3	2	1	0
Bits da porta								
TRIS para 5 saídas e 3 entradas	1	1	1	0	0	0	0	0

Figura 17.31 Direção de porta.

Bits do ADCON0			
5	4	3	Para as entradas analógicas
0	0	0	Porta A, bit 0
0	0	1	Porta A, bit 1
0	1	0	Porta A, bit 2
0	1	1	Porta A, bit 3
1	0	0	Porta A, bit 5
1	0	1	Porta E, bit 0
1	1	0	Porta E, bit 1
1	1	1	Porta E, bit 2

3 Temporizadores

Este microcontrolador tem 3 temporizadores: temporizador 0, temporizador 1 e temporizador 2. O temporizador 0 é um contador de 8 bits que pode ser escrito ou lido e pode ser usado para contar as transições de um sinal externo, gerando uma interrupção quando o número desejado de eventos tiver ocorrido. A fonte para a contagem pode ser o sinal de clock do barramento interno ou um sinal digital externo. A escolha da fonte para contagem é feita através do bit TOCS no registrador OPTION (Figura 17.32).

Se o *prescaler* não for selecionado, então a contagem é incrementada a cada dois ciclos da fonte de entrada. Um *prescaler* pode ser usado de modo que os sinais apenas passem para o contador após algum outro número fixo de ciclos de clock. A tabela a seguir mostra as taxas possíveis de escala. WDT fornece os fatores de escala selecionados quando o temporizador *matchdog* está habilitado. Ele é usado para, ao terminar a temporização, resetar o sistema se uma operação não é concluída em um tempo razoável; o tempo *default* é nominalmente 18 ms.

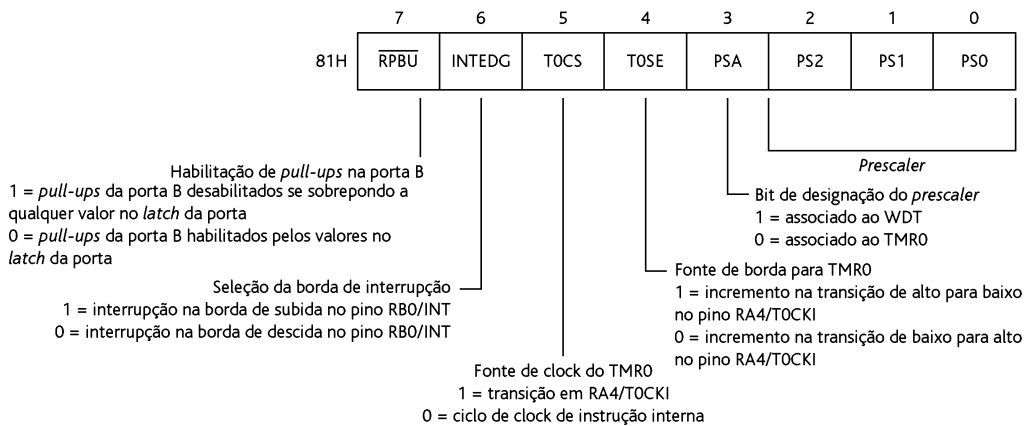


Figura 17.32 Registrador OPTION.

Valores dos bits do <i>prescaler</i>			Taxa do TMRO	Taxa do WDT
PS2	PS1	PS0		
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

O temporizador 1 é o mais versátil dos temporizadores e pode ser usado para monitorar o tempo entre transições de sinais em um pino de entrada ou controlar o tempo preciso das transições em um pino de saída. Quando usado com os modos de captura ou comparação, ele permite ao microcontrolador controlar a temporização de uma saída no pino 17.

O temporizador 2 pode ser usado para controlar o período de uma saída PWM. As saídas PWM são fornecidas nos pinos 16 e 17.

4 *Entrada/saída serial*

O microcontrolador PIC inclui uma porta serial síncrona (SSP) e uma interface de comunicação serial (SCI). O pino 18 tem as funções alternativas de entrada de clock serial síncrona ou saída para o modo de interface serial para periférico SPI e o modo I²C. O barramento I²C provê uma interface bidirecional de dois fios que pode ser usada com outros chips; ele também pode ser usado para conexão de um microcontrolador mestre com microcontroladores escravos. O UART, ou seja, o transmissor/receptor assíncrono universal, pode ser usado para criar uma interface serial com um computador pessoal (PC).

5 *Porta escrava paralela*

A porta escrava paralela usa as portas D e E e habilita o microcontrolador a prover uma interface com um PC.

6 *Entrada do cristal*

O pino 13 é para entrada do cristal para o oscilador ou a entrada de uma fonte de clock externa; o pino 14 é para a saída do oscilador a cristal. A Figura 17.33(a) mostra a configuração que pode ser usada para um controle de frequência preciso. A Figura 17.33(b) mostra que pode ser usado um controle de frequência de baixo custo; para uma frequência de 4 MHz, podemos ter $R = 4,7 \text{ k}\Omega$ e $C = 33 \text{ pF}$. A taxa de clock interno é a frequência do oscilador dividida por 4.

7 *Clear e reset mestres*

O pino 1 é o *clear* mestre MCLR, ou seja, a entrada de reset, e é colocado em nível baixo para resetar o dispositivo quando necessário e promover a inicialização. Quando é detectada a subida de V_{DD} , o pulso de reset ao energizar (POR) é gerado para prover um atraso de tempo fixo e manter o processador no estado de reset. Se a tensão V_{DD} cair abaixo de um valor específico por mais do que

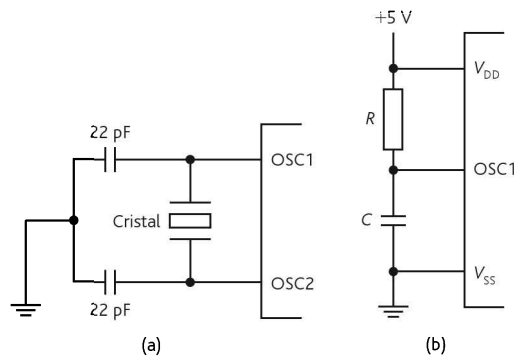


Figura 17.33 Controle de frequência.

certo tempo, um reset por queda de energia (*brownout*) é ativado. O temporizador *matchdog* é outra forma de reset que pode ocorrer. Este é um temporizador que, ao final da temporização, reseta o microcontrolador se uma operação não for concluída num tempo considerado adequado.

Os **registradores de função especial** (Figura 17.34) são usados para controle de entrada/saída, conforme ilustrado antes para alguns destes registradores. Os registradores do PIC16C73/74 são organizados em dois bancos e, antes que um determinado registrador seja acessado, o banco tem que ser escolhido por meio de um bit no registrador de status (Figura 17.35).

17.3.4 Seleção de um microcontrolador

Deve-se considerar os seguintes fatores na seleção de um microcontrolador:

1 *Número de pinos de entrada/saída*

Quantos pinos de entrada/saída são necessários para o projeto do sistema?

2 *Interfaces necessárias*

Quais interfaces são necessárias? Por exemplo, é necessário um PWM? Muitos microcontroladores têm saídas PWM, por exemplo, o PIC17C42 tem duas.

3 *Requisitos de memória*

Qual é o tamanho da memória necessária para o sistema?

4 *O número de interrupções necessárias*

Quantos eventos necessitarão de interrupções?

5 *Requisitos de velocidade de processamento*

O microprocessador gasta tempo para executar instruções (veja a Seção 18.2.2), e este tempo é determinado pelo clock do processador.

Como ilustração das variações disponíveis de microcontroladores, a Tabela 17.1 mostra detalhes de membros da família 8051 da Intel, a Tabela 17.2 mostra a família PIC16Cxx e a Tabela 17.3 mostra a família M68HC11.

Endereço dos registradores de arquivo (f)	Banco 0	Banco 1	Endereço dos registradores de arquivo (f)
00h	INDF	INDF	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	PORTC	TRISC	87h
08h	PORTD	TRISD	88h
09h	PORTE	TRISE	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh	PIR2	PIE2	8Dh
0Eh	TMR1L	PCON	8Eh
0Fh	TMR1H		8Fh
10h	T1CON		90h
11h	TMR2		91h
12h	T2CON	PR2	92h
13h	SSPBUF	SSPADD	93h
14h	SSPCON	SSPSTAT	94h
15h	CCPR1L		95h
16h	CCPR1H		96h
17h	CCP1CON		97h
18h	RCSTA	TXSTA	98h
19h	TXREG	SPBRG	99h
1Ah	RCREG		9Ah
1Bh	CCPR2L		9Bh
1Ch	CCPR2H		9Ch
1Dh	CCP2CON		9Dh
1Eh	ADRES		9Eh
1Fh	ADCON0	ADCON1	9Fh
20h	Registradores de propósito geral	Registradores de propósito geral	A0h
7Fh			FFh

Figura 17.34 Registradores de propósito especial.

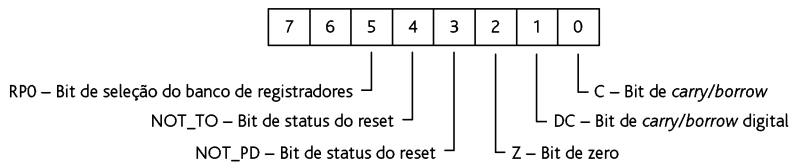


Figura 17.35 Registrador de STATUS.

Tabela 17.1 Membros da família 8051 da Intel

	ROM	EPROM	RAM	Temporizadores	Portas de I/O	Interrupções
8031AH	0	0	128	2	4	5
8051AH	4K	0	128	2	4	5
8052AH	8K	0	256	3	4	6
8751AH	0	4K	128	2	4	5

Tabela 17.2 Membros da família PIC16C

	I/O	EPROM	RAM	Canais do ADC	USART	Módulos CCP
PIC16C62A	22	2K	128	0	0	1
PIC16C63	22	4K	192	0	1	2
PIC16C64A	33	2K	128	0	0	1
PIC16C65A	33	4K	192	0	1	2
PIC16C72	22	2K	128	5	0	1
PIC16C73A	22	4K	192	5	1	2
PIC16C74A	33	4K	192	8	1	3

Tabela 17.3 Membros da família M68HC11

	ROM	EEPROM	RAM	ADC	Tempo- rizador	PWM	I/O	Serial	E-clock MHZ
68HC11AO	0	0	256	8 canais, 8 bits	(1)	0	22	SCI, SPI	2
68HC11A1	0	512	256	8 canais, 8 bits	(1)	0	22	SCI, SPI	2
68HC11A7	8K	0	256	8 canais, 8 bits	(1)	0	38	SCI, SPI	3
68HC11A8	8K	512	256	8 canais, 8 bits	(1)	0	38	SCI, SPI	3
68HC11C0	0	512	256	4 canais, 4 bits	(2)	2 canais, 8 bits	36	SCI, SPI	2
68HC11D0	0	0	192	Nenhum	(2)	0	14	SCI, SPI	2

Temporizador: (1) captura com três entradas, comparação com cinco saídas, temporizador *watchdog*, acumulador de pulsos, (2) captura com tres ou quatro entradas, comparação com cinco ou quatro saídas, interrupção em tempo real, temporizador *watchdog*, acumulador de pulsos. Serial: SCI é interface de comunicação serial assíncrona, SPI é interface serial síncrona para periférico.

17.4 Aplicações

Apresentamos a seguir dois exemplos de como os microcontroladores são usados; mais estudos de casos são apresentados no Capítulo 24.

17.4.1 Sistema de medição de temperatura

Em uma breve descrição de como um microcontrolador pode ser usado, a Figura 17.36 mostra os principais elementos de um sistema de medição de temperatura usando um M68HC11. O sensor de temperatura fornece uma tensão proporcional à temperatura

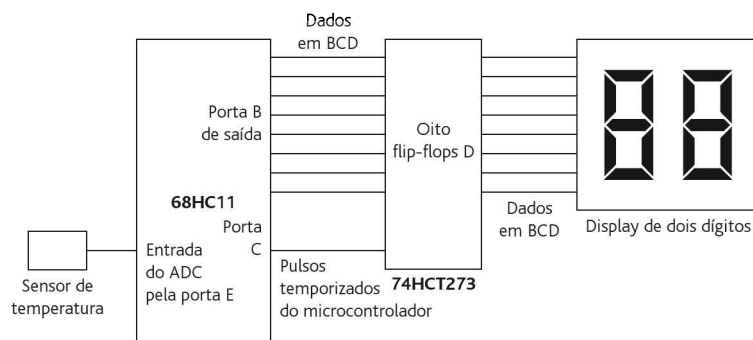


Figura 17.36 Sistema de medição de temperatura.

(por exemplo, um termotransistor como o LM35; veja a Seção 2.9.4). A saída do sensor de temperatura está conectada na linha de entrada do ADC do microcontrolador. Este é programado para converter a temperatura em uma saída BCD que pode ser usada para ligar os elementos de um display de sete segmentos e dois dígitos. Entretanto, como a temperatura pode flutuar, é necessário usar um registrador de armazenamento que pode manter o dado por um tempo suficiente para que o display seja lido. Este registrador, 74HCT273, é constituído de um conjunto de 8 flip-flops do tipo D que é resetado na próxima borda positiva na entrada de clock proveniente do microcontrolador.

17.4.2 Máquina de lavar roupa doméstica

A Figura 17.37 mostra como um microcontrolador pode ser usado como o controlador de uma máquina de lavar roupa doméstica. O microcontrolador geralmente usado é o Motorola M68HC05B6; esta versão é mais simples e mais barata do que o M68HC11 discutido anteriormente neste capítulo e é bastante usada em aplicações de baixo custo.

As entradas dos sensores para temperatura da água e velocidade do motor são feitas via porta de entrada do conversor analógico-digital. A porta A provê as saídas para os atuadores usados para controlar a máquina e também a entrada para a chave do nível de água. A porta B fornece as saídas para o display. A porta C fornece saídas para o display e também recebe as entradas do teclado usado para a seleção dos programas na máquina. A seção PWM do temporizador fornece um sinal PWM para controlar a velocidade do motor. Toda a operação da máquina é interrompida e paralisada se a porta dela for aberta.

17.5 Programação

Um método normalmente usado para desenvolvimento de programas segue os seguintes passos:

- 1 Defina o problema, descrevendo de forma bem clara quais são as funções que o programa tem que realizar, as entradas e saídas necessárias, quaisquer limitações em relação a velocidade de operação, precisão, tamanho da memória, etc.

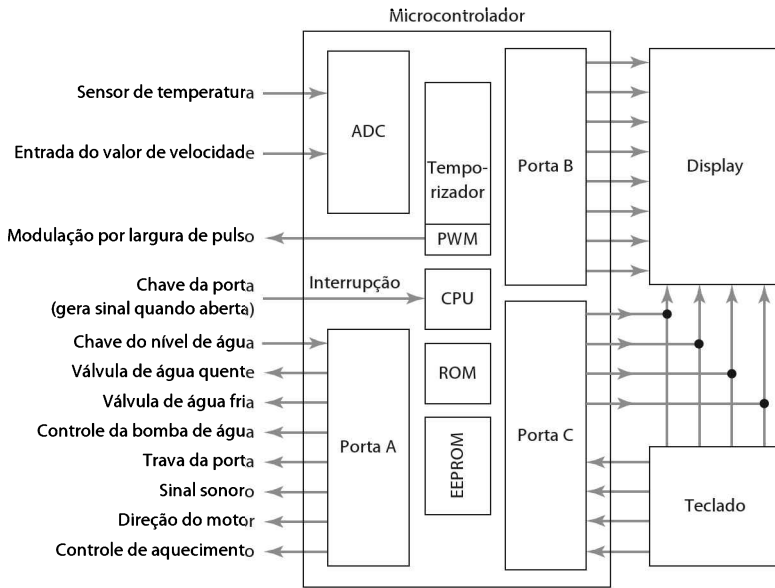


Figura 17.37 Máquina de lavar roupa.

- 2 Defina o algoritmo a ser usado. Um **algoritmo** é a sequência de passos que define um método de solução do problema.
- 3 Para um sistema com poucas centenas de instruções, a representação do algoritmo através de um **fluxograma** pode ser útil. A Figura 17.38(a) mostra os símbolos padrão usados na elaboração de um fluxograma. Cada passo de um algoritmo é representado por um ou mais destes símbolos, que são interconectados por linhas que representam o fluxo do programa. A Figura 17.38(b) mostra parte de um fluxograma em que, seguindo-se o programa a partir do início, há uma operação A seguida de uma ramificação para uma operação B ou C que depende do resultado da pergunta, que é sim ou não. Uma outra ferramenta de projeto útil é o **pseudocódigo**, que é um jeito informal de descrever os passos de um algoritmo para posteriormente ser traduzido em um programa (veja a próxima seção).
- 4 Traduza o fluxograma/algoritmo em instruções que o processador possa executar. Isto pode ser feito escrevendo-se as instruções em uma linguagem, como por exemplo, o assembly ou ainda o C, e, em seguida, convertê-las, manualmente ou através de um programa de computador, para o código aceito pelo microprocessador, ou seja, o código de máquina.
- 5 Teste e depure o programa. Os erros em um programa são denominados *bugs*, e o processo de busca deles é denominado *debugging*.

17.5.1 Pseudocódigo

O pseudocódigo é bastante semelhante a um fluxograma e envolve a escrita de um programa como uma sequência de funções ou operações com o elemento de decisão IF-THEN-ELSE e o elemento de repetição WHILE-DO.

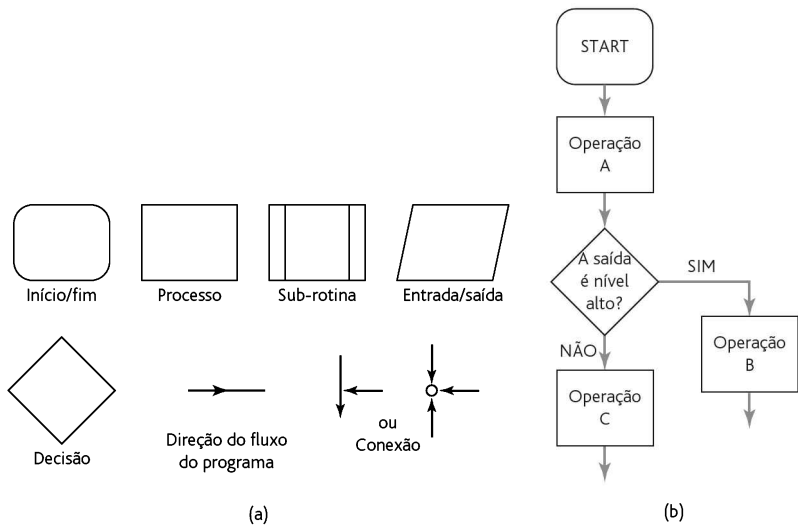


Figura 17.38 Fluxograma: (a) símbolos, (b) exemplo.

Uma seqüência (Figura 17.39(a)) poderia ser escrita assim:

```
BEGIN A
...
END A
...
BEGIN B
...
END B
```

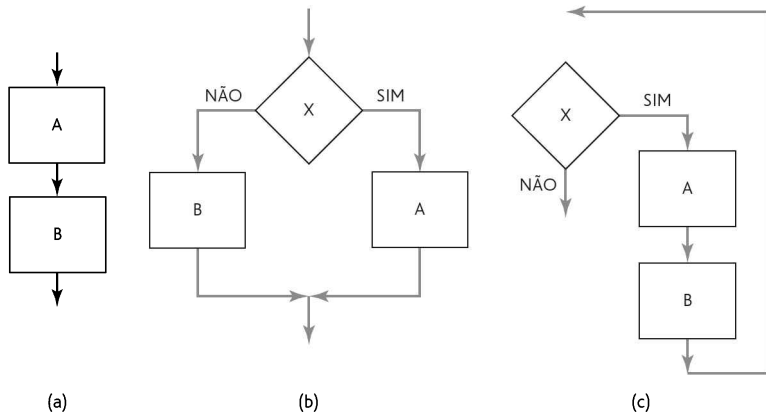


Figura 17.39 (a) Seqüência, (b) IF-THEN-ELSE, (c) WHILE-DO.

e uma decisão como:

```
IF X
THEN
  BEGIN A
  ...
END A
ELSE
  BEGIN B
  ...
END B
ENDIF X
```

A Figura 17.39(b) mostra esta decisão na forma de fluxograma. Uma repetição é escrita como:

```
WHILE X
DO
  BEGIN A
  ...
END A
  BEGIN B
  ...
END B
ENDO WHILE X
```

A Figura 17.39(c) mostra o WHILE-DO na forma de fluxograma. Um programa escrito desta forma poderia ter o seguinte aspecto:

```
BEGIN PROGRAM
  BEGIN A
    IF X
      BEGIN B
      END B
    ELSE
      BEGIN C
      END C
    ENDIF X
  END A
  BEGIN D
    IF Z
      BEGIN E
      END E
    ENDIF Z
  END D
END PROGRAM
```

O Capítulo 18 mostra como programas podem ser escritos em linguagem assembly e o Capítulo 19, em linguagem C.

Resumo

Basicamente, os sistemas que envolvem **microprocessadores** têm três partes: uma unidade central de processamento (CPU), interfaces de entrada e saída, e memória. Dentro do microprocessador, os sinais digitais se movem em barramentos, que são trilhas em paralelo para transmissão de dados em paralelo em vez de em série.

Os **microcontroladores** são a integração em um único chip de um microprocessador com memória, interfaces de entrada/saída e outros periféricos, como temporizadores.

Um **algoritmo** é a sequência de passos que define um método de solução de um problema. O **fluxograma** e o **pseudocódigo** são dois métodos de descrição destes passos.

Problemas

- 17.1 Explique, considerando um microprocessador, o papel (a) do registrador acumulador, (b) do registrador de status, (c) de um endereço de memória, (d) do registrador contador de programa.
- 17.2 Um microprocessador usa oito linhas de endereços para acessar memória. Qual é o número máximo de posições de memória que podem ser endereçadas?
- 17.3 Um chip de memória tem 8 linhas de dados e 16 linhas de endereços. Qual é a capacidade dele?
- 17.4 Em que um microcontrolador difere de um microprocessador?
- 17.5 Desenhe um diagrama em bloco de um controlador básico e explique o funcionamento deste subsistema.
- 17.6 Qual das portas de um M68HC11 é usada para (a) o ADC, (b) uma porta bidirecional, (c) entrada/saída serial, (d) uma porta de saída de apenas 8 bits?
- 17.7 Quantos bytes de memória o M68HC11A7 tem para memória de dados?
- 17.8 No M68HC11 da Motorola, a porta C é bidirecional. Como ela é configurada para ser (a) uma entrada, (b) uma saída?
- 17.9 O M68HC11 da Motorola pode ser operado em modo de chip único e expandido. Que modos são estes?
- 17.10 Qual é a finalidade do pino ALE no 8051 da Intel?
- 17.11 Qual entrada é necessária ativar para resetar um microcontrolador 8051 da Intel?
- 17.12 Escreva um pseudocódigo para representar o seguinte:
 - (a) Se (*if*) A for sim, então B, caso contrário (*else*) C.
 - (b) Enquanto (*while*) A for sim, faça (*do*) B

18

Linguagem Assembly

OBJETIVOS

Ao final do estudo deste capítulo o leitor deverá ser capaz de:

- Usar a linguagem assembly para escrever programas envolvendo transferência de dados, operações aritméticas e lógicas, saltos, desvios, sub-rotinas, atrasos de tempo e listagens (LTU – *look-up table*).

18.1 Linguagens

Software é o termo usado para as **instruções** que “dizem” ao microprocessador ou microcontrolador o que fazer. A coleção de instruções que um microprocessador reconhece é o seu **conjunto de instruções**. A forma deste conjunto de instruções varia de um microprocessador para outro. A série de instruções usadas para executar uma tarefa em particular é denominada **programa**.

Os microprocessadores operam com código binário. As instruções escritas em código binário são denominadas **código de máquina**. Escrever um programa neste código é um processo muito especializado e enfadonho. Ele é propenso a erros porque o programa é constituído apenas de uma sequência de zeros e uns e as instruções não são facilmente interpretadas apenas olhando-se o código. Uma alternativa é o uso compreensivelmente mais fácil de um código compacto para os padrões de zeros e uns. Por exemplo, a operação da soma de um dado com o acumulador pode ser representada apenas por ADDA. Este código abreviado é denominado **código mnemônico**, e é mais fácil de memorizar. O termo **linguagem assembly** é usado para este código. Escrever um programa usando mnemônicos é mais fácil porque eles são uma versão abreviada da operação realizada através da instrução. Além disto, como as instruções descrevem as operações do programa e podem ser compreendidas mais facilmente, elas são menos propensas a erros do que os padrões binários dos programas em código de máquina. Entretanto, o programa em assembly ainda tem que ser convertido em código de máquina, pois este é o que

o microprocessador é capaz de reconhecer. Esta conversão pode ser feita manualmente usando-se as folhas de dados do fabricante contendo os códigos de máquina de cada mnemônico. Entretanto, existem programas de computador disponíveis para esta conversão denominados assembladores.

As linguagens de alto nível, como, por exemplo, BASIC, C, FORTRAN e PASCAL, representam um tipo de linguagem de programação que descreve em uma linguagem compreensivelmente mais fácil os tipos de operações desejadas. Entretanto, estas linguagens têm que ser convertidas em código de máquina, através de um programa de computador, para que o microprocessador seja capaz de usá-las. Este capítulo é uma abordagem de como os programas podem ser escritos em linguagem assembly, e o Capítulo 19 aborda a linguagem C.

18.2 Conjuntos de instruções

As instruções a seguir são normalmente usados por microprocessadores. A lista completa de instruções é denominada conjunto de instruções. O Apêndice C apresenta os conjuntos de instruções de três tipos de microcontroladores normalmente encontrados. O conjunto de instruções difere de um microprocessador para outro. A seguir são apresentadas algumas das instruções mais comuns:

Transferência/movimentação de dados

1 *Load*

Esta instrução lê o conteúdo de uma posição específica da memória e copia-a para um registrador especificado da CPU, e é tipicamente usada em microprocessadores Motorola, como por exemplo, LDAA \$0010:

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Dado na posição 0010	Dado na posição 0010
	Dado da posição 0010 no acumulador A

2 *Store*

Esta instrução copia o conteúdo atual de um registrador especificado em uma posição de memória específica e é tipicamente usada em microprocessadores Motorola, como, por exemplo, STA \$0011:

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Dado no acumulador A	Dado no acumulador A
	Dado copiado para a posição 0011

3 *Move*

Esta instrução é usada para mover um dado para um registrador ou copiá-lo de um registrador para outro, e é usada em microprocessadores PIC e Intel, como, por exemplo, em um PIC, MOV R5,A:

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Dado no registrador A	Dado no registrador A
	Dado copiado para o registrador R5

4 *Clear*

Esta instrução reseta todos os bits para zero, como, por exemplo, CLRA para zerar o acumulador A (microprocessador Motorola); CLRF 06 para zerar o registrador de arquivo 06 (microcontrolador PIC).

Aritmética

5 *Add*

Esta instrução soma um número ao dado de um registrador, como, por exemplo, ADD A, #10h (microprocessador Intel):

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Acumulador A com dado	Acumulador A mais 10 hexa

e em um Motorola, ADDD \$0020:

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Acumulador D com dado	Acumulador D mais o conteúdo da posição 0020

ou o conteúdo de um registrador ao dado em um registrador, como, por exemplo, ADD A, @R1 (microprocessador Intel):

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Acumulador A com dado	Acumulador A mais o conteúdo da posição R1

e em um PIC, addwf 0C:

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Registrador 0C com dado	Registrador 0C mais conteúdo da posição w

6 *Decrement*

Esta instrução subtrai 1 do conteúdo de uma posição especificada. Por exemplo, podemos ter o registrador 3 como a posição especificada e assim, em um Intel, DEC R3:

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Registrador R3 com dado 0011	Registrador R3 com dado 0010

7 *Incremento*

Esta instrução soma 1 ao conteúdo de uma posição especificada, como, por exemplo, INCA, em um Motorola, para incrementar em 1 o dado no acumulador A; incf 06, em um PIC, para incrementar em 1 o dado no registrador 06.

8 *Compare*

Esta instrução indica se o conteúdo de um registrador é maior, menor ou igual ao conteúdo de uma posição de memória especificada. O resultado aparece no registrador de status como um *flag*.

Lógica

9 *AND*

Esta instrução executa a operação lógica AND entre o conteúdo da posição de memória especificada e o dado de um registrador. A operação AND é feita bit a bit, como, por exemplo, ANDA %1001 (Motorola):

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Acumulador A com o dado 0011	Acumulador A com o dado 0001
Posição de memória com o dado 1001	

Apenas no bit menos significativo destes dados há um bit 1 nos dois conjuntos de dados, e a operação AND produz um resultado 1 no bit menos significativo. Em um PIC, ANDLW 01 faz a soma lógica (AND) entre o número 01 e o número em W, e se o bit menos significativo for, por exemplo, 0, então o resultado é 0.

10 OR

Esta instrução executa, bit a bit, a operação lógica OR entre o conteúdo de uma posição de memória específica e o dado em um registrador, como, por exemplo, ORL A,#3Fh (Intel), que faz a operação OR entre o conteúdo do registrador A e o número hexa 3F.

11 EXCLUSIVE-OR

Esta instrução executa, bit a bit, a operação lógica EX-OR com o conteúdo de uma posição de memória especificada e o dado de um registrador, como, por exemplo, xorlw 81h (em binário 10000001) em um PIC:

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Registrador w com 10001110	Registrador w com 00001111

Uma operação EX-OR de um bit com um 0 resulta no mesmo valor do bit, e no caso da operação ser com 1, resulta no inverso do bit.

12 Deslocamento lógico (esquerda ou direita)

As instruções de deslocamento lógico envolvem a movimentação de um padrão de bits em um registrador um bit à esquerda ou à direita com a inserção de um 0 na extremidade do número. Por exemplo, para o deslocamento lógico à direita, um 0 é deslocado para o bit mais significativo, e o menos significativo é movido para o *flag* de *carry* no registrador de status. A instrução pode ser LSRA (Motorola) para o deslocamento à direita e LSLA para à esquerda.

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Acumulador com 0011	Acumulador com 0001
	Registrador de status indica Carry igual a 1

13 Deslocamento aritmético (esquerda ou direita)

As instruções de deslocamento aritmético envolvem a movimentação de um padrão de bits em um registrador deslocando para a esquerda ou para a direita uma posição correspondente a um bit, mas preservando o bit de sinal na extremidade esquerda do número, como, por exemplo, para um deslocamento aritmético à direita com a instrução ASRA (Motorola):

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Acumulador com 1011	Acumulador com 1001
	Registrador de status indica Carry igual a 1

14 Rotação (esquerda ou direita)

As instruções de rotação envolvem a movimentação do padrão de bits no registrador uma posição para a esquerda ou direita e o bit que sai é escrito na outra extremidade, ou seja, para uma rotação à direita, a instrução Intel RR A produz:

<i>Antes da instrução</i>	<i>Depois da instrução</i>
Acumulador em 0011	Acumulador com 1001

Controle do programa

15 Salto ou desvio

Esta instrução altera a sequência na qual o programa é executado. Normalmente o contador de programa faz com que o programa seja executado sequencialmente em uma determinada sequência numérica. Entretanto, a instrução de salto (*jump*) faz com que o contador de programa salte para uma outra posição especificada no programa (Figura 18.1(a)). Saltos incondicionais ocorrem sem que o programa esteja testando a ocorrência de uma condição. Portanto, podemos ter, no caso de um dispositivo Intel, LJMP POINTA para o programa saltar para a linha do programa rotulada de POINTA. No caso de um dispositivo Motorola, a instrução seria JMP POINTA e, em um PIC, seria GOTO POINTA. Saltos condicionais ocorrem se alguma condição for satisfeita (Figura 18.1(b)). No caso de um dispositivo Intel, podemos ter JNZ POINTA para o programa saltar para a linha no programa rotulada de POINTA se qualquer bit do acumulador não for zero, caso contrário, a execução do programa continua na próxima linha. A instrução JZ POINTA é executada se todos os bits do acumulador forem zero. No caso do PIC, um salto condicional pode envolver duas linhas de código: BTFC 05,1 para “teste o bit e salte”, ou seja, teste se o bit 1 do registrador de file 5 e, se o resultado for 0, então passe para a próxima linha do programa; se for 1, execute o salto da instrução. A próxima linha é GOTO POINTA. Um desvio, para um dispositivo Motorola, é uma instrução *jump* (salto) condicional para o programa determinar que desvio será seguido se as condições especificadas ocorrerem. Por exemplo, o dispositivo Motorola usa BEQ para desviar se igual a zero, BGE para desviar se maior ou igual a zero, BLE para desviar se menor ou igual a zero.

16 Halt/stop

Esta instrução para a operação do microprocessador.

Os dados numéricos podem ser em binário, octal, hexa ou decimal. Geralmente, na ausência de um indicador o assembler assume que o número é decimal. Para um Motorola, um número é indicado pelo prefixo #; um número binário é precedido por

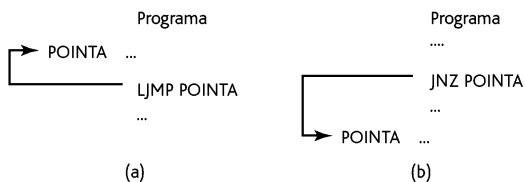


Figura 18.1 (a) Salto incondicional, (b) salto condicional.

% ou seguido por B; um número octal é precedido por @ ou seguido por O; um número hexa é precedido por \$ ou seguido por H; e um número decimal não necessita de letra ou símbolo indicativo. Para um dispositivo Intel, os valores numéricos têm que ser precedido por # para indicar um número e por B para binário, O ou Q para octal, H ou h para hexa e D ou nenhuma indicação para decimal. No caso dos microcontroladores PIC, o arquivo de cabeçalho tem R = DEC para decimal como *default*. Então, para binário, os números são colocados entre aspas e precedido por B, e para hexa, por H.

18.2.1 Endereçamento

Quando um mnemônico, como LDA, é usado para especificar uma instrução, ele tem que ser acompanhado de uma informação adicional para especificar a fonte e o destino do dado que a instrução necessita. O dado que acompanha a instrução é denominado **operando**.

Existem diferentes métodos que são usados para especificar as posições dos dados, ou endereçamento, e portanto a forma na qual o microprocessador obtém suas instruções ou dados. Microprocessadores diferentes têm modos de endereçamento diferentes. O 68HC11 da Motorola tem seis modos de endereçamento: imediato, direto, estendido, indexado, inerente e relativo; o 8051 da Intel tem cinco modos: imediato, direto, por registrador, indireto e indexado; o microcontrolador PIC tem três modos: imediato, direto e indireto, sendo que o modo indireto permite a indexação. A seguir apresentamos os modos comuns de endereçamento:

1 *Endereçamento imediato*

O dado que acompanha o mnemônico é o valor a ser operado e é usado para carregar um valor predeterminado em um registrador ou posição de memória. Por exemplo, no caso do código Motorola, LDA B #25 significa carregar o número 25 no acumulador B. O símbolo # significa modo de endereçamento imediato, e o símbolo \$ significa que o número é hexa. No código Intel, temos MOV A, #25H, que move o número 25 para o acumulador A. O símbolo # indica um número e o H indica um número hexa. No código PIC, temos movlw H'25', que carrega o número 25 no registrador de trabalho w, e H indica que o número é hexa.

2 *Endereçamento direto, absoluto, estendido ou de página zero*

Nesta forma de endereçamento, o byte de dado que acompanha o código de operação fornece diretamente um endereço que define a posição do dado a ser usado na instrução. Em um Motorola, o termo **endereçamento direto** é usado quando o endereço dado tem apenas 8 bits de extensão; o termo **endereçamento estendido** é usado quando ele é de 16 bits de extensão. Por exemplo, no código Motorola, LDAA \$25 quer dizer carregar o acumulador com o conteúdo da posição de memória 0025; o 00 é um valor considerado que complementa o endereço. No código Intel, para a mesma operação, podemos ter a instrução de endereçamento direto MOV A, 20H para copiar o dado do endereço 20 no acumulador A. No código do PIC, podemos ter movwf Reg1 para copiar o conteúdo do Reg1 para o registrador de trabalho, com o endereço de Reg1 sendo definido previamente.

3 *Endereçamento implícito ou inerente*

Com este modo de endereçamento, o endereço é implícito na instrução. Por exemplo, no código Motorola, CLR A significa zerar o acumulador A. No código PIC, clrw significa zerar o registrador de trabalho.

4 Por registrador

Com esta forma de endereçamento, o operando é especificado como o conteúdo de um dos registradores interno. Por exemplo, com um Intel, `ADD R7,A` soma o conteúdo do acumulador ao registro R7.

5 Indireto

Esta forma de endereçamento significa que o dado é encontrado na posição de memória na qual o endereço é dado pela instrução. Por exemplo, em um sistema com PIC, INDF e os registradores FSR são usados. O endereço é escrito primeiro no registro FSR, e este serve como um ponteiro de endereço. O subsequente acesso direto de INDF com a instrução `movf INDF,w` carrega o registrador de trabalho w usando o conteúdo de FSR como um ponteiro da posição do dado.

6 Endereçamento indexado

O endereçamento indexado significa que o dado está em uma posição de memória cujo endereço é mantido em um registrador de índice. O primeiro byte da instrução contém o código de operação, e o segundo byte contém o *offset* (deslocamento); o *offset* é somado ao conteúdo do registrador de índice para determinar o endereço do operando. Uma instrução de um Motorola pode ser da forma `LDA A $FF,X`; isto significa carregar o acumulador A com o dado no endereço dado pela soma do conteúdo do registrador de índice com FF. Um outro exemplo é `STA A $05,X`; esta instrução significa armazenar o conteúdo do acumulador A no endereço dado pelo registrador de índice somado com 05.

7 Endereçamento relativo

Este é usado com instruções de desvio. O código de operação é acompanhado de um byte denominado endereço relativo. Isto indica que o deslocamento no endereço tem que ser somado ao contador de programa se o desvio ocorrer. Por exemplo, o código Motorola `BEQ $F1` indica que se o dado for igual a zero, então o próximo endereço no programa é F1 endereços à frente. O endereço relativo de F1 é somado ao endereço da próxima instrução.

Como ilustração, a Tabela 18.1 mostra algumas instruções com os modos de endereçamento usados em sistemas Motorola.

Tabela 18.1 Exemplos de endereçamento

Modo de endereçamento	Instrução	
Imediato	<code>LDA A #\$F0</code>	Carrega o acumulador A com o dado F0
Direto	<code>LDA A \$50</code>	Carrega o acumulador A com o dado do endereço 0050
Estendido	<code>LDA A \$0F01</code>	Carrega o acumulador A com o dado do endereço 0F01
Indexado	<code>LDA A \$CF,X</code>	Carrega o acumulador A com o dado do endereço dado pelo registrador de índice mais CF
Inerente	<code>CLR A</code>	Zera o acumulador A
Estendido	<code>CLR \$2020</code>	Zera o endereço 2020, ou seja, armazena zero em todos os bits do endereço 2020
Indexado	<code>CLR \$10,X</code>	Zera o endereço dado pelo registrador de índice mais 10, ou seja, armazena zero em todos os bits deste endereço

18.2.2 Movimentação de dados

O exemplo a seguir é obtido das folhas de dados do fabricante do 6800:

Operação	Mnemônico	Modo de endereçamento					
		IMEDIATO			DIRETO		
		OP	~	#	OP	~	#
Add	ADDA	8B	2	2	9B	3	2

~ é o número de ciclos do microprocessador necessário e # é o número de bytes de programa necessário.

Isto significa que quando se usa o modo de endereçamento imediato com este processador, a operação de soma é representada pelo mnemônico ADDA. Quando a forma imediata de endereçamento é usada, o código de máquina desta instrução é 8B e ela gasta dois ciclos para ser executada completamente. Esta operação necessita de 2 bytes no programa. O **código de operação** (*op-code*) é o termo usado para a instrução que diz o que o microprocessador fará e é expresso em hexadecimal. Um byte é um grupo de oito dígitos binários reconhecido pelo microcomputador como uma palavra. Portanto, são necessárias duas palavras. Com o modo de endereçamento direto, o código de máquina é 9B e gasta três ciclos e dois bytes de programa.

Para ilustrar como a informação trafega entre a memória e o microprocessador, considere as seguintes tarefas. Os endereços na RAM nos quais um novo programa pode ser colocado são apenas uma questão de conveniência. Nos exemplos a seguir, o endereço inicial é 0010. Para o endereçamento direto ser usado, o endereço tem que estar na página zero, ou seja, entre 0000 e 00FF. Os exemplos são baseados no uso do conjunto de instruções do microprocessador M6800.

Tarefa: Inserir zeros em todo o acumulador

Endereço de memória	Código de operação
0010	8F CLR A

O próximo endereço de memória que pode ser usado é 0011 porque CLR A ocupa apenas um byte no programa. Este é o modo de endereçamento inerente.

Tarefa: Somar o conteúdo do acumulador A com o dado 20.

Endereço de memória	Código de operação
0010	8B 20 ADD A #20

Esta instrução usa a forma de endereçamento imediato. O próximo endereço de memória que pode ser usado é 0012 porque, nesta forma de endereçamento, ADD A ocupa dois bytes no programa.

Tarefa: Carregar o acumulador A com o dado contido no endereço de memória 00AF.

Endereço de memória	Código de operação
0010	B6 00AF LDA A \$00AF

Esta instrução usa a forma de endereçamento absoluto. O próximo endereço de memória que pode ser usado é 0013 porque, nesta forma de endereçamento, LDA A ocupa três bytes no programa.

Tarefa: Rotacionar à esquerda o dado contido na posição de memória 00AF.

Endereço de memória	Código de operação
0010	79 00AF ROL \$00AF

Esta instrução usa a forma de endereçamento absoluto. O próximo endereço de memória que pode ser usado é 0013, visto que ROL, neste modo, ocupa três bytes no programa.

Tarefa: Armazenar na posição 0021 da memória o dado contido no acumulador A.

Endereço de memória	Código de operação
0010	D 721 STA A \$21

Esta instrução usa o modo de endereçamento direto. O próximo endereço de memória que pode ser usado é 0012, visto que STA A, neste modo, ocupa dois bytes no programa.

Tarefa: Saltar quatro endereços à frente se o resultado da instrução anterior for zero.

Endereço de memória	Código de operação
0010	27 04 BEQ \$04

Esta instrução usa o modo de endereçamento relativo. O próximo endereço de memória a ser usado se o resultado não for zero é 0012, visto que BEQ, neste modo, ocupa dois bytes no programa. Se o resultado for zero, o próximo endereço será $0012 + 4 = 0016$.

18.3 Programas em linguagem assembly

Um programa em linguagem assembly deve ser considerado como uma série de instruções a partir das quais o assembler (*assembler*) produzirá o programa em código de máquina. Um programa escrito em linguagem assembly consiste de uma sequência de instruções, sendo uma por linha. Uma instrução contém de uma a quatro seções, ou **campos**, a saber:

Rótulo (*label*) Código de operação (*op-code*) Operando Comentário

Um símbolo especial é usado para indicar o início ou o final de um campo. Os símbolos usados dependem do assembler que gera o código de máquina do microprocessador. No caso do 6800 da Motorola, são usados espaços. No caso do 8080 da Intel, existe uma vírgula depois de um rótulo, um espaço depois de um código de operação, vírgulas entre os apontamentos no campo de endereço e um ponto-e-vírgula antes de um comentário. Em geral, é usado um ponto e vírgula para separar comentários de operandos.

O **rótulo** é o nome pelo qual uma entrada situada na memória é denominada. Os rótulos podem ser constituídos de letras, números e outros caracteres. No 6800 da

Motorola, os rótulos são restritos entre um e seis caracteres, sendo o primeiro obrigatoriamente uma letra e não podendo ser apenas uma das letras A, B ou X, visto que estas são reservadas para indicar acumulador e registrador de índice. No 8080 da Intel, são permitidos cinco caracteres, sendo o primeiro uma letra, @ ou ?. O rótulo não pode usar quaisquer dos nomes reservados para registradores, códigos de instruções ou pseudo-operações (veja a seguir nesta seção). Todos os rótulos em um programa têm que ser únicos. Se não houver um rótulo, um espaço deve ser inserido no campo do rótulo. No 6800 da Motorola, um asterisco (*) no campo do rótulo indica que toda a declaração é um comentário, ou seja, um comentário incluído com a finalidade de tornar clara a interpretação do programa. Desta forma, o comentário será ignorado pelo assembler durante o processo de gerar o código de máquina do programa.

O código de operação (*op-code*) especifica como os dados devem ser manipulados de acordo com o mnemônico, como, por exemplo, LDA A. Este é o único campo que não pode ficar vazio. Além disto, ele pode conter diretivas para o assembler. Estas são denominadas **pseudo-operações**, visto que elas aparecem no campo código de operação mas não são traduzidas em código de máquina. Elas podem definir símbolos, definir programas e dados em determinadas áreas de memória, gerar tabelas fixas e dados, indicar o fim do programa, etc. A seguir mostramos algumas diretivas comuns do assembly:

Carrega o contador de programa

ORG Define o endereço de memória inicial da parte do programa que vem a seguir. Um programa pode ter várias origens.

Define símbolos

QUE, SET, DEF Iguala/carrega/define um símbolo para um valor numérico, um outro símbolo ou expressão.

Reserva posições de memória

RMB, RES Reserva bytes/espaco de memória

Define constantes na memória

FCB Constante na forma de byte.

FCC Constante na forma de string de caracteres.

FDB Constante na forma de byte duplo.

BSW Armazena blocos de zeros.

A informação incluída no campo do **operando** depende do mnemônico precedente e do modo de endereçamento. Ela fornece o endereço do dado a ser operado pelo processo especificado através do código de operação. Este campo também é denominado **campo de endereço**. Este campo pode estar vazio se as instruções dadas pelo código de operação não necessitarem de qualquer dado ou endereço. Os dados numéricos neste campo podem ser em hexa, decimal, octal ou binário. O assembler considera que os números são decimais a menos que haja especificação em contrário. No 6800 da Motorola, um número hexa é precedido por \$ ou seguido de H, um número octal é precedido por @ ou seguido de O ou Q, um número binário é precedido por % ou seguido de B. No 8080 da Intel, um número hexa é seguido de H, um número octal por O ou Q, um número binário por B. Os números em hexa têm que iniciar com um caractere numérico, ou seja, 0 a 9, para evitar confusão com nomes. No 6800 da Motorola, o modo imediato de endereço pode ser indicado por # antes do operando, o modo de endereçamento indexado envolvendo o operando seguido por X. Nos modos de endereçamento direto ou estendido, não são usados símbolos especiais. Se o endereço estiver na página

zero, ou seja, FF ou menos, o assembler associa automaticamente o modo direto. Se o endereço for maior do que FF, o assembler associa o modo estendido.

O campo de comentário é opcional e permite ao programador incluir qualquer comentário que possa tornar o programa mais fácil de ser entendido pelo leitor. Este campo é ignorado pelo assembler durante a montagem do código de máquina do programa.

18.3.1 Exemplos de programas em linguagem assembly

Os exemplos a seguir ilustram como alguns programas simples podem ser desenvolvidos.

Problema: Somar dois números de 8 bits situados em diferentes endereços de memória e armazenar o resultado de volta na memória.

O algoritmo é:

- 1 Início
- 2 Carregar o acumulador com o primeiro número. O acumulador é onde os resultados das operações lógicas e aritméticas são acumulados. Ele é um registrador de trabalho, ou seja, como um bloco de notas no qual os cálculos são realizados antes do resultado ser transferido para outro local. Portanto, podemos copiar o dado para o acumulador antes de executar a operação aritmética. No PIC é usado o termo registrador de trabalho (w).
- 3 Somar com o segundo número.
- 4 Armazenar a soma na posição de memória especificada.
- 5 Parar.

A Figura 18.2 mostra estes passos representados como um fluxograma.

Os programas a seguir foram feitos para três microcontroladores diferentes. Em cada programa, a primeira coluna é o rótulo, a segunda é o código de operação, a terceira é o operando e a quarta é o comentário. Note que todos os comentários são precedidos de ponto e vírgula.

Programa para o M68HC11

```

; Adição de dois números

NUM1 EQU $00 ; posição do número 1
NUM2 EQU $01 ; posição do número 2
SUM EQU $02 ; posição da soma

START ORG $C000 ; endereço de início da RAM do usuário
      LDAA $NUM1 ; carrega o 1º número no acc A
      ADDA $NUM2 ; soma o 2º número ao A
      STAA SUM ; salva a soma em $02
      END

```

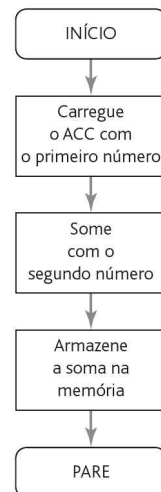


Figura 18.2 Fluxograma para a soma de dois números.

A primeira linha do programa especifica o endereço do primeiro número a ser somado. A segunda linha especifica o endereço do segundo número a ser somado com o primeiro. A terceira linha especifica onde a soma deve ser co-

locada. A quarta linha especifica o endereço de memória no qual o programa deve começar. O uso de rótulos significa que o operando envolve um dado que não tem um endereço específico, mas apenas rótulos.

O mesmo programa para o 8051 da Intel vem a seguir.

Programa para o 8051

```

; Adição de dois números

NUM1 EQU 20H ; posição do número 1
NUM2 EQU 21H ; posição do número 2
SUM EQU 22H ; posição da soma

START ORG 800H ; endereço de início da RAM do usuário
MOV A,NUM1 ; carrega o 1º número no acc A
ADD A,NUM2 ; soma o 2º número ao A
MOV SUM,A ; salva a soma no endereço 22H
END
    
```

A seguir, o mesmo programa para o PIC.

PIC program

```

; Adição de dois números

Num1 equ H'20' ; posição do número 1
Num1 equ H'21' ; posição do número 2
Sum equ H'22' ; posição da soma

Start org H'000' ; endereço de início da RAM do usuário
movlw Num1 ; carrega o 1º número em w
addlw Num2 ; soma o 2º número ao conteúdo de w
movwf Sum ; salva a soma em H'22'
End
    
```



Figura 18.3 Um loop de programação.

Em muitos programas pode haver a necessidade de uma tarefa ser executada sucessivamente um determinado número de vezes. Nestes casos, o programa pode ser projetado de modo que a operação passe pela mesma seção do programa um determinado número de vezes. Isto é denominado **loop de programação**. Um **loop** é uma seção de um programa que é repetida um determinado número de vezes. A Figura 18.3 mostra um fluxograma de um **loop**. Com este loop, certa operação tem que ser realizada um determinado número de vezes antes do programa seguir. O programa a seguir usa um **loop** de programação:

Problema: Somar os números situados em 10 endereços diferentes (estes podem ser o resultado de entrada de 10 sensores diferentes que foram amostrados).

O algoritmo poderia ser este a seguir:

- 1 Início
- 2 Carregue o contador com 10.
- 3 Aponte para o número do endereço inferior.

- 4 Some o número do endereço inferior.
- 5 Decremente 1 no contador.
- 6 Some 1 ao ponteiro de endereço.
- 7 O contador é 0? Se não, desvie para 4. Se sim, continue.
- 8 Armazene a soma.
- 9 Pare.

A Figura 18.4 mostra o fluxograma.

O programa é:

```

COUNT EQU $0010
POINT EQU $0020
RESULT EQU $0050
ORG $0001
LDA B COUNT ; Carrega contador
LDX POINT ; Inicializa registrador de índice com
           : endereço inicial dos números
SUM ADD A X ; Soma parcela
INX ; Soma 1 ao registrador de índice
DEC B ; Subtrai 1 do acumulador B
BNE SUM ; Desvie para sum
STA A RESULT ; Armazena
WAI ; Para o programa

```

A contagem com o número 10 é carregada no acumulador B. O registrador de índice fornece o endereço inicial do dado a ser somado. O primeiro passo do somatório é a soma do conteúdo da posição de memória endereçada pelo registrador de índice ao conteúdo do acumulador, considerado inicialmente zero (uma instrução CLR A deve ser usada para zerar o acumulador primeiro). A instrução INX soma 1 ao registrador de índice de forma que o próximo endereço a ser acessado seja 0021. DEC B subtrai 1 do conteúdo do acumulador B, que passa a indicar uma contagem 9. BNE é a instrução que desvia para SUM se não for igual a zero, ou seja, se o *flag Z* tiver valor 0. O programa entra em *loop* até ACC B ser zero.

Problema: Determine o maior número de uma lista (que pode ser, por exemplo, o maior valor de temperatura fornecido por sensores de temperatura).

O algoritmo pode ser:

- 1 Zere o conteúdo do endereço da resposta.
- 2 Carregue o ponteiro com o endereço inicial da lista.
- 3 Carregue o número a partir do endereço acessado na lista.
- 4 Compare o número da lista com o número do endereço da resposta.
- 5 Armazene o número se for maior do que a resposta.
- 6 Caso contrário, manter o número da resposta.
- 7 Incremente em 1 o ponteiro que endereça a lista.
- 8 Desvie para o passo 3 se não for o último endereço.
- 9 Pare.

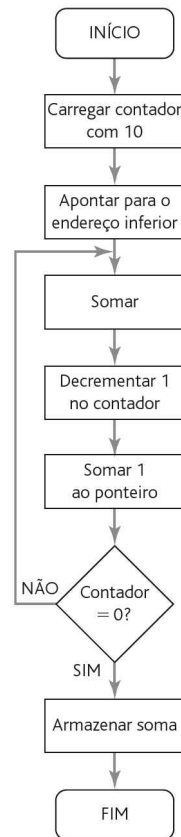


Figura 18.4 Fluxograma para a soma de 10 números.

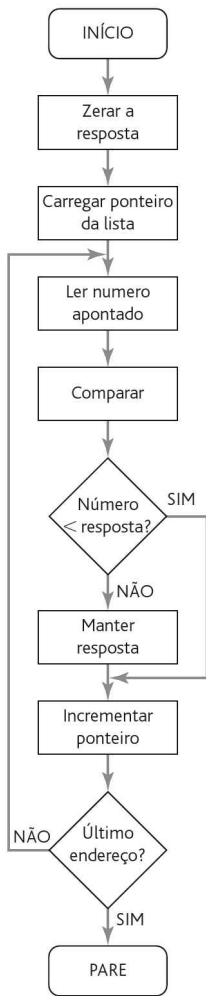


Figura 18.5 Fluxograma para encontra maior número.

A Figura 18.5 mostra o fluxograma. O programa é mostrado a seguir:

FIRST	EQU	\$0030	
LAST	EQU	\$0040	
ANSW	EQU	\$0041	
	ORG	\$0000	
	CLR	ANSW	; Zera resposta
	LDX	FIRST	; Carrega ponteiro com o 1º endereço da lista
NUM	LDA A	\$30,X	; Lê número apontado
	CMP A	ANSW	; Compara com resposta
	BLS	NEXT	; Desvia para próximo endereço se menor ou igual
	STA A	ANSW	; Armazenar resposta
NEXT	INX		; Incrementa registrador de índice
	CPX	LAST	; Compara registrador de índice com LAST
	BNE	NUM	; Desvia se diferente de zero
	WAI		; Fim do programa

Neste programa, primeiro o endereço da resposta é zerado. O primeiro endereço é carregado e o número deste endereço é colocado no acumulador A. A instrução LDA A \$30,X significa carregar o acumulador A com o dado do endereço contido no registrador de índice mais 30. Compara o número com a resposta, mantendo a resposta se ela for maior do que o número no acumulador, caso contrário, desvia para repetir o loop com o próximo número da lista.

18.4 Sub-rotinas

É comum situações em que um bloco de programa, uma sub-rotina, precisa ser executada um determinado número de vezes em um programa. Por exemplo, pode ser necessário um bloco de programa para gerar um atraso de tempo. Seria possível copiar o programa da sub-rotina o número de vezes necessário no programa principal. Entretanto, seria um uso ineficiente da memória. Alternativamente, podemos ter na memória uma única cópia da sub-rotina e desviar ou saltar para ela todas as vezes que for necessário executá-la. Entretanto, isto apresenta o problema de saber, após a execução da sub-rotina, o ponto no qual o programa principal tem que ser retomado. Neste caso, há a necessidade de um mecanismo de retorno ao programa principal no ponto em que o mesmo foi deixado para a execução da sub-rotina. Para fazer isto precisamos armazenar o conteúdo do contador de programa no instante do desvio para a sub-rotina de modo que este valor possa ser recarregado no contador de programa quando a sub-rotina for finalizada. As duas instruções existentes no conjunto de instruções da maioria dos microprocessadores para permitir o uso de sub-rotinas são deste tipo:

- 1 JSR (*jump to routine*) ou CALL, que permite que uma sub-rotina seja chamada.
- 2 RTS (*return from subroutine*) ou RET (*return*), que é usada como a última instrução da sub-rotina e retorna para o ponto correto do programa que chamou a sub-rotina.

As sub-rotinas podem ser chamadas a partir de qualquer ponto do programa. Para isto é necessário armazenar o conteúdo do contador de programa na forma de uma LIFO (*last-in-first-out*) em que o último a entrar será o primeiro a sair. Este armazenamento é denominado **pilha**. Ele é análogo a uma pilha de pratos em que o primeiro prato a ser removido desta pilha é sempre o de cima e, portanto, o último prato a ser acrescentado à pilha. A memória de pilha pode ser um bloco de registradores em um microprocessador ou, como é mais comum, usando uma seção da RAM. Um registrador especial dentro do microprocessador, denominado ponteiro de pilha (*stack pointer*), é usado para apontar para o próximo endereço livre na área de RAM a ser usado para a pilha.

Além do uso automático da pilha quando se usam sub-rotinas, um programador pode fazer um programa que envolve o uso da pilha para armazenamento temporário de dados. As duas instruções usadas são:

- 1 PUSH, que faz com que dados de registradores especificados sejam salvos no próximo espaço livre na pilha.
- 2 PULL, ou POP, que faz com que os dados sejam recuperados da última posição da pilha, sendo transferidos para o registrador especificado.

Por exemplo, antes de alguma sub-rotina, os dados de alguns registradores podem ser salvos e, após a sub-rotina, ser recuperados. A parte do programa que faz isto no 6800 da Motorola é exemplificada a seguir:

```

SAVE      PSH A   ; Salva o acumulador A na pilha
          PSH B   ; Salva o acumulador B na pilha
          TPA     ; Transfere o registrador de status para o acumulador A
          PSH A   ; Salva o registrador de status na pilha
; Sub-rotina
RESTORE   PUL A   ; Retorna o dado da pilha para o acumulador A
          TAP     ; Transfere dado de A para registrador de status
          PUL B   ; Retorna da pilha para o acumulador B
          PUL A   ; Retorna da pilha para o acumulador A

```

18.4.1 Sub-rotina de atraso de tempo

Os **loops de atraso de tempo** (*delay*) são frequentemente necessários quando o microprocessador tem uma entrada a partir de um dispositivo, como um conversor analógico-digital. Geralmente há a necessidade de enviar um sinal para um conversor iniciar a conversão e esperar um tempo fixo antes de se ler o dado do conversor. Isto pode ser feito criando-se um *loop* que faz com que o microprocessador execute instruções antes de continuar o programa. Um programa simples de atraso de tempo pode ser:

```

DELAY    LDA A   #05      ; Carrega o acumulador A com 05
LOOP     DEC A           ; Decrementa o acumulador A de 1
          BNE    LOOP     ; Desvia se não for igual a zero
          RTS           ; Retorna da sub-rotina

```

Para cada execução deste *loop*, um determinado número de ciclos de máquina estão envolvidos. No programa de atraso, ao percorrer o *loop* cinco vezes, temos:

Instrução	Ciclos	Total de ciclos
LDA A	2	2
DECA	2	10
BNE	4	20
RTS	1	1

O atraso total é, portanto, de 33 ciclos de máquina. Se cada ciclo de máquina gasta, por exemplo, 1 μ s, então o atraso total é de 33 μ s. Para um atraso maior, um número maior pode ser colocado no acumulador A.

Um exemplo de uma sub-rotina de atraso de tempo para um microcontrolador PIC é:

```

Delay      movlw   Value      ; carrega o valor de contagem necessário
          movwf  Count    ; contador de loop
          decfsz Count    ; decrementa o contador
          goto   Delay    ; loop
    
```

A instrução *decfsz* gasta um ciclo, e a instrução *goto* gasta dois. Este *loop* é repetido (*count* - 1) vezes. Além disso, temos as instruções *movlw* e *movwf*, cada uma de um ciclo, e quando o contador for igual a 1 teremos *decfsz*, que gasta mais dois ciclos. Portanto, o número total de ciclos é:

$$\text{números de ciclos de instrução} = 3(\text{count} - 1) + 4$$

Cada ciclo de instrução gasta 4 ciclos do clock, e assim o número, de ciclos de atraso introduzidos pela sub-rotina é:

$$\text{número de ciclos de clock} = 4[2(\text{count} - 1) + 4]$$

Com um clock de 4 MHz, cada ciclo de clock gasta $1/(4 \times 10^6)$ s.

Frequentemente os atrasos obtidos com o uso de um único *loop*, como apresentado, não são suficientes. Uma forma de se obter um atraso maior é usar uma associação de *loops*. A Figura 18.6 mostra o fluxograma para um atraso usando *loops* associados. O *loop* interno é o mesmo descrito antes para um programa com *loop* único. Ele decrementa o registrador E 255 vezes antes que o *loop* seja completado e o *flag* de zero seja setado. O *loop* externo faz com que a rotina do *loop* interno seja executada repetidamente à medida que o registrador D é decrementado até zero. Portanto, com o registrador D carregado inicialmente com uma contagem de, por exemplo, 140, então o atraso de tempo será $140 \times 2,298 = 321,71$ ms.

O programa é então:

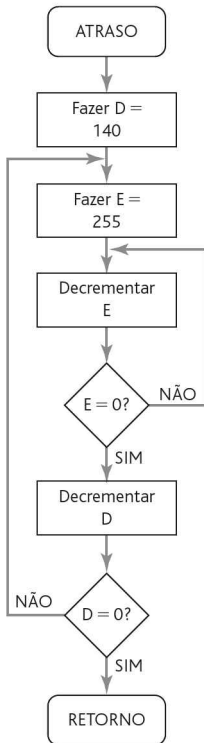


Figura 18.6 Atraso de tempo com *loops* associados.

DELAY	MOV	D,8CH	; carrega D com 8CH, ou seja, 140
OLOOP	MOV	E,FFH	; carrega E com FFH, ou seja, 255
ILOOP	DEC	E	; decreenta E, ou seja, contador do <i>loop</i> interno
	JNZ	ILOOP	; repete ILOOP 255 vezes
	DEC	D	; decreenta D, ou seja, contador do <i>loop</i> externo
	JNZ	OLOOP	; repete OLOOP 140 vezes

A seguir apresentamos exemplos de programas nos quais uma sub-rotina de atraso de tempo é usada:

1 *Problema*: Ligar e desligar um LED repetidamente.

Neste problema, a sub-rotina DELAY é usada com *loop* para gerar atrasos de tempo; o microprocessador gasta uma quantidade finita de tempo para processar as instruções no *loop*. A estrutura do programa é a seguinte:

- 1 Se LED ligado
 - Desligar LED
 - Enquanto LED desligado, faça a sub-rotina TIME_DELAY
- 2 Se não
 - Ligar o LED
 - Executar a sub-rotina TIME_DELAY

Sub-rotina: TIME_DELAY

A execução de uma instrução, ou instruções, ou um *loop*, ou um *loop* duplo depende do tempo de atraso necessário.

Em função do atraso de tempo que é necessário, é usado um *loop* duplo para gerar o atraso de tempo. No caso do 8051 da Intel, é possível usar a instrução DJNZ, decrementar e saltar se o resultado não for zero. Ela decreenta a posição indicada pelo primeiro operando e salta para o segundo operando se o resultado não for zero. O LED está conectado ao bit 0 da porta 1 do microcontrolador. O programa em linguagem assembly para o 8051 pode ser o seguinte:

FLAG	EQU	0FH	; flag setado (em 1) quando LED ligado
	ORG	8000H	
START	JB	FLAG,LED_OFF	; salta para LED_OFF se LED ligado
	SETB	FLAG	; se não, seta o bit FLAG
	CLR	P1.0	; liga o LED
	LCALL	DELAY	; chama a sub-rotina de atraso (DELAY)
	SJMP	START	; salta para START
LED_OFF	CLR	FLAG	; zera o flag de LED ligado para indicar que o LED está desligado
	SETB	P1.0	; desliga o LED
	LCALL	DELAY	; valor do atraso para o <i>loop</i> externo
	LJMP	START	; valor do atraso para o <i>loop</i> interno
DELAY	MOV	R0,#OFFH	; espera no <i>loop</i> interno
ILOOP	MOV	R1,#OFFH	; espera no <i>loop</i> externo
OLOOP	DJNZ	R1,ILOOP	; retorno da sub-rotina
	DJNZ	R0,OLOOP	
	RET		
	END		

2 Problema: Ligar oito LEDs em sequência.

A instrução de rotação pode ser usada sucessivamente para ligar LEDs de forma que tenhamos inicialmente a sequência de bits 0000 0001, que é então rotacionada, resultando em 0000 0011, em seguida passa a ser 0000 0111, e assim por diante. Vemos as seguir um programa em linguagem assembly para o 68HC11 da Motorola que pode ser usado. Os LEDs estão conectados na porta B; um pequeno atraso é incorporado ao programa:

```

COUNT    EQU    8                ; COUNT representa o número de loops
                                                ; desejados, ou seja, o número de
                                                ; bits a serem ligados
FIRST     EQU    %00000001        ; para ligar o bit 0
PORTB     EQU    $1004            ; endereço da porta B
                                                ORG    $C000
LDAA      #FIRST                  ; carrega valor inicial
LDAB      #COUNT                ; carrega contador
LOOP      STAA  PORTB             ; liga o bit 1 que corresponde ao LED 1
          JSR   DELAY             ; salta para sub-rotina DELAY
          SEC                      ; seta o bit carry para girar através do
                                                ; bit menos significativo para mantê-lo 1
                                                ROLA                      ; rotaciona à esquerda
          DECB                    ; decrementa o contador
          BNE  LOOP              ; desvia para loop oito vezes
DELAY     RTS                     ; atraso curto simples
          END

```

18.5 Listagens (LUTs)

O endereçamento indexado pode ser usado para habilitar um programa a buscar valores em uma tabela. Por exemplo, na determinação de quadrados de inteiros, um método possível é buscar os valores correspondentes a um inteiro em particular em uma tabela de quadrados, em vez de fazer a operação aritmética para determinar o quadrado. As listagens (LUTs – *Look Up-Tables*) são particularmente úteis quando a relação não é linear e não é descrita por uma equação aritmética simples, como, por exemplo, no gerenciamento de um motor de combustão descrito na Seção 1.7.2, em que o tempo de ignição é uma função do ângulo do virabrequim e da pressão de admissão. Neste caso, o microcontrolador tem que gerar um sinal temporizado que depende dos sinais de entrada do sensor de velocidade e dos sensores do virabrequim.

Para ilustrar como as tabelas de busca são usadas, considere o problema na determinação de quadrados de inteiros. Podemos colocar uma tabelas de quadrados dos inteiros 0, 1, 2, 3, 4, 5, 6, ... na memória de programa e os quadrados 0, 1, 4, 9, 16, 25, 36, ... em endereços sucessivos. Se o número a ser elevado ao quadrado for 4, então ele se torna índice do endereço indexado do dado na tabela, estando a primeira linha da tabela associada ao índice 0. O programa soma o índice ao endereço

base da tabela para determinar o endereço da entrada correspondente ao inteiro. Portanto, temos:

Índice	0	1	2	3	4	5	6
Entrada da tabela	0	1	4	9	16	25	36

Por exemplo, com o microcontrolador 68HC11 da Motorola, podemos fazer o seguinte programa de busca para encontrar o quadrado:

```

REGBAS EQU    $B600      ; endereço base da tabela
          ORG    $E000
          LDAB   $20      ; carrega o acc. B com o inteiro a ser elevado ao quadrado
          LDX   #REGBAS  ; aponta para a tabela
          ABX    ; soma o conteúdo de acc. B ao do registrador de índice X
          LDAA  $00,X    ; carrega o acc. A com o valor indexado

```

e poderíamos ter carregado a Tabela na memória usando a pseudo-operação FDB:

```

ORG    $B600      ; define os valores para o bloco de memória reservado
FDB    $00,$01,$04,$09 ; block

```

Com o microcontrolador 8051 da Intel a instrução `MOVC A,@A+DPTR` busca o dado na posição de memória apontada pela soma do DPTR com o acumulador A e o armazena no acumulador. Esta instrução pode ser usada para buscar um dado em uma tabela em que o ponteiro de dados DPTR é inicializado com o primeiro endereço da tabela. Como ilustração, suponha que queremos usar uma tabela para a conversão de temperatura na escala Celsius para a Fahrenheit. O programa envolve a passagem de parâmetro, para a conversão da temperatura desejada, para uma sub-rotina, de modo que ela pode incluir as seguintes instruções:

```

          MOV    A,#NUM    ; carrega o valor a ser convertido
          CALL  LOOK_UP   ; chama a sub-rotina LOOK_UP

LOOK_UP  MOV    DPTR,#TEMP ; aponta para a tabela
          MOVC  A,@A+DPTR ; lê o valor na tabela
          RET    ; retorna da sub-rotina

TMP      DB    32, 34, 36, 37 ; carrega os valores na tabela
          39, 41, 43, 45

```

Um outro exemplo do uso de uma tabela é o acionamento sequencial de saídas. Isto pode ser uma operação sequencial de um semáforo que gera a sequência vermelho, vermelho mais amarelo, verde e amarelo. A luz vermelha é ativada quando é ativada a saída RD0, a amarela é ativada por RD1 e a verde, por RD2. Os dados podem ser como segue:

	Vermelho	Vermelho + amarelo	Verde	Amarelo
Índice	0	1	2	3
	0000 0001	0000 0011	0000 0100	0000 0010

18.5.1 Atraso com um motor de passo

Na operação de um motor de passo devem ser usados atrasos que têm que ser inseridos entre cada instrução usada para avançar um passo a fim de dar um tempo para que o movimento de um passo ocorra antes da próxima instrução. Veja a seguir um algoritmo para gerar uma sequência contínua de passos:

- 1 Início.
- 2 Determine a sequência necessária na saída para obter a sequência de passos desejada.
- 3 Coloque na posição inicial.
- 4 Avance um passo.
- 5 Salte para a rotina de atraso.
- 6 É o último passo da sequência de uma rotação completa? Se não, continue com o próximo passo; se sim, volte ao passo 3.
- 7 Continue indefinidamente.

O programa a seguir é para um motor de passo na configuração passo completo e controlado pelo microcontrolador M68HC11 através das saídas PB0, PB1, PB2 e PB3. É usada uma LTU para se obter a sequência do código de saída necessário a partir do acionador de motor de passo. A seguir também é mostrada a LUT usada.

A sequência de código necessária para a operação de um motor de passo de passo completo é A, 9, 5, 6, A, e estes valores constituem a sequência que o ponteiro tem que buscar na tabela. FCB é o código de operação para “*form constant byte*” e é usado para definir os bytes de dados da tabela.

As saídas necessárias a partir da porta B

Passo	PB0	PB1	PB2	PB3	Código
1	1	0	1	0	A
2	1	0	0	1	9
3	0	1	0	1	5
4	0	1	1	0	6
1	1	0	1	0	4

BASE	EQU	\$1000	
PORTB	EQU	\$4	; Porta de saída
TFLG1	EQU	\$23	; Registrador do <i>flag</i> de interrupção do temporizador 1
TCNT	EQU	\$0E	; Registrador de contagem do temporizador
TOC2	EQU	\$18	; Registrador de comparação de saída 2
TEN_MS	EQU	20000	; 10 ms baseado no clock
	ORG	\$0000	
STTBL	FCB	\$A	; Esta é a LUT
	FCB	\$9	
	FCB	\$5	
	FCB	\$6	
ENDTBL	FCB	\$A	; Fim da LUT

```

        ORG    $C000
        LDX    #BASE
        LDAA   #$80
        STAA   TFLG1,X      ; Zera flag
START   LDY    #STTBL
BEG     LDAA   0,Y          ; Começa com a primeira posição na tabela
        STAA   PORTB,X
        JSR    DELAY        ; Salta para DELAY
        INY    ; Próxima posição da tabela
        CPY    #ENTBL      ; É o final da tabela?
        BNE   BEG          ; Se não, desvia para BEG
        BRA   START        ; Se sim, vai para START novamente

        DELAY   LDD    TCNT,X
        ADDD   #TEN_MS      ; Soma um atraso de tempo de 10 ms
        STD    TOC2,X
HERE    BRCLR  TFLG1, X, $80, HERE ; Espera até que o atraso de tempo tenha
                                   decorrido
        LDAA   #$80
        STAA   TFGL1        Zera flag
        RTS

```

Note que, neste programa, o rótulo `TEM_MS` tem um espaço sublinhado (*underline*) para indicar que `TEM` e `MS` são partes do mesmo rótulo.

O atraso neste programa é obtido usando-se um bloco de temporizador no microcontrolador. Este temporizador é programado para um atraso de tempo de 10 ms. Para um sistema microcontrolado com um clock de 2 MHz, um atraso de 10 ms corresponde a 20.000 ciclos de clock. Portanto, para obter este atraso, o valor atual do registrador `TCNT` é determinado, 20.000 ciclos são somados a ele e o registrador `TOC2` é carregado com este valor.

18.6 Sistemas embarcados

Os microprocessadores e microcontroladores são frequentemente “embarcados” nos sistemas de forma que possa ser exercido controle. Por exemplo, uma máquina de lavar roupa doméstica tem um microcontrolador interno programado com as diferentes programações de lavagem; tudo o que o operador desta máquina tem que fazer é selecionar o programa de lavagem desejado através de uma chave e o programa solicitado é executado. O operador não tem que programar o microcontrolador. O termo **sistema embarcado** é usado para um sistema microprocessado que é projetado para controlar uma ou mais funções e não é projetado para ser programado pelo usuário do sistema. A programação é feita pelo fabricante e é “gravada” no sistema de memória, não podendo ser alterada pelo usuário do sistema.

18.6.1 Programas de sistemas embarcados

Em um sistema embarcado, o fabricante coloca o programa em uma memória ROM. Isto é viável economicamente se houver um grande número destes chips. Alterna-

tivamente, para protótipos e aplicações de poucas unidades, o programa pode ser carregado em uma EPROM/EEPROM no *hardware* da aplicação. A seguir apresentamos uma ilustração de como a EPROM/EEPROM de microcontroladores pode ser programada.

Por exemplo, para programar a EPROM do microcontrolador 8051 da Intel, é necessária a configuração mostrada na Figura 18.7(a). Na entrada do oscilador deve haver de 4 a 6 MHz. O procedimento é:

- 1 O endereço de uma posição da EPROM, a ser programada na faixa de 0000H a 0FFFH, é aplicado na porta 1 e os pinos P2.0 e P2.1 da porta 2; no mesmo instante, o byte do código a ser programado neste endereço é aplicado na porta 0.
- 2 Os pinos P2.7, RST e ALE devem ser mantidos em nível alto e os pinos P2.6 e PSEN, em nível baixo. Para os pinos P2.4 e P2.5 não importa se eles estão em nível alto ou baixo.
- 3 O pino EA/ V_{pp} é mantido no nível lógico alto até antes de ALE ser pulsado, então ele é elevado para +21 V, ALE é pulsado em nível baixo por 50 ms para programar o byte do código na posição endereçada, e então EA é retornado para nível alto.

A verificação do programa, ou seja, a leitura do programa, é conseguida com a configuração mostrada na Figura 18.7(b).

- 1 O endereço da posição do programa a ser lido é aplicado na porta 1 e os pinos de P2.0 a P2.3 da porta 2.
- 2 Os pinos EA/ V_{pp} , RST e ALE devem ser mantidos em nível alto; os pinos P2.7, P2.6 e PSEN, em nível baixo. Para os pinos P2.4 e P2.5 não importa se eles estão em nível alto ou baixo.
- 3 O conteúdo da posição endereçada sai pela porta 0.

Um bit de segurança pode ser programado para impedir o acesso elétrico por qualquer meio externo à memória de programa no chip. Uma vez que este bit foi programado, ele só pode ser apagado com o apagamento da memória de programa. A mesma configuração para programação (Figura 18.7(a)) é usada, mas P2.6 é man-

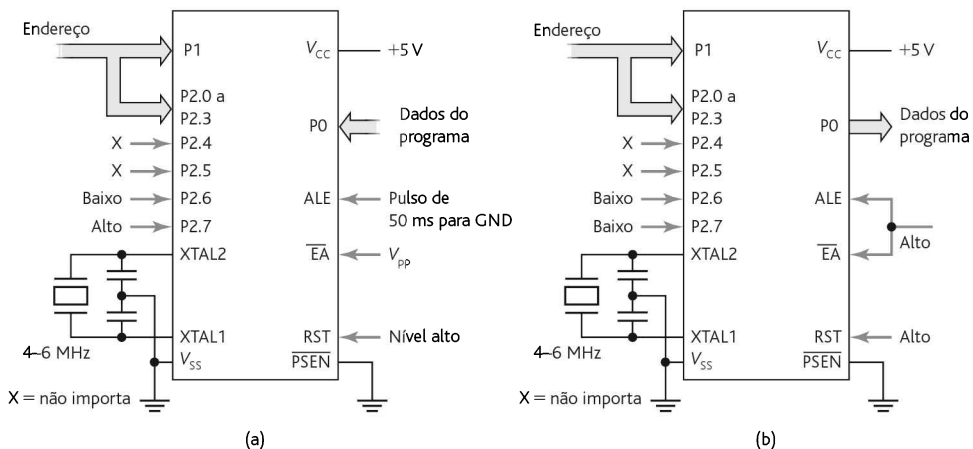


Figura 18.7 O 8051 da Intel: (a) programação, (b) verificação.

tido em nível alto. O apagamento é feito por exposição à luz ultravioleta. Como a luz solar e de lâmpadas fluorescentes contém luz ultravioleta, deve ser evitada exposição prolongada (cerca de uma semana para a luz solar ou 3 anos para uma lâmpada fluorescente em uma sala) e deve-se proteger a janela do chip com uma etiqueta opaca.

O microcontrolador 68HC11 da Motorola é oferecido com uma memória apenas de leitura programável e apagável eletricamente (EEPROM). Esta memória está localizada entre os endereços \$B600 e \$B7FF. Assim como uma EPROM, um byte é apagado quando todos os bits são 1 e a programação envolve a inserção de bits 0. A EEPROM é habilitada setando-se o bit EEON no registrador CONFIG (Figura 18.8) em 1 e desabilitada em 0. A programação é controlada pelo registrador de programação da EEPROM (PPROG) (Figura 18.8).

O procedimento para programação é:

- 1 Escreva no registrador PPROG setando o bit EELAT em 1 para programação.
- 2 Escreva o dado no endereço da EEPROM selecionada. Isto armazena o endereço e o dado a ser programado.
- 3 Escreva no registrador PPROG setando o bit EEPGM em 1 para ligar a tensão de programação.
- 4 Aguarde por 10 ms (*delay*).
- 5 Escreva no registrador PPROG para zerar todos os bits.

A seguir temos uma sub-rotina de programação em linguagem assembly usada no MC68HC11:

```
EELAT EQU %00000010 ; Bit EELAT
EEPGM EQU %00000001 ; Bit EEPGM
PPROG EQU $1028 ; endereço do registrador PPROG
```

EEPROM

```
PSHB
LDAB #EELAT
STAB PPROG ; seta EELAT = 1 e EEPGM = 0
STAA 0,X ; armazena o dado X no endereço da EEPROM
LDAB #%00000011
```

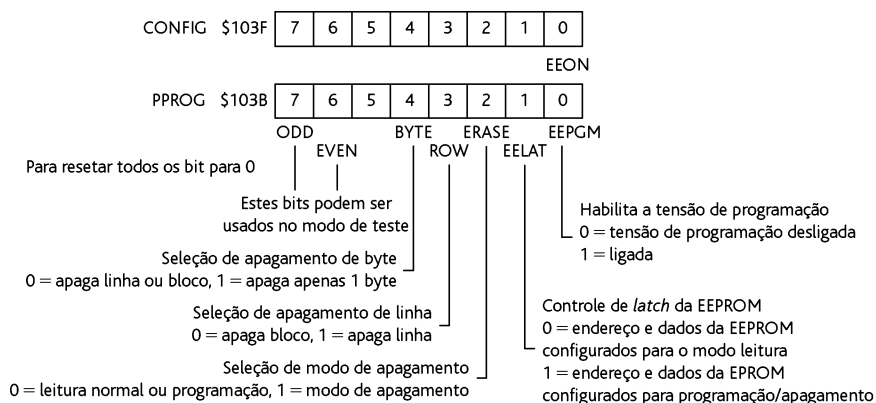


Figura 18.8 Registradores CONFIG e PPROG.

```

STAB  PPROG      ; seta EELAT = 1 e EEPGM = 1
JSR   DELAY_10  ; salta para a sub-rotina de atraso DELAY_10
CLR   PPROG      ; zera todos os bits de PPROG e retorna para
                  o modo leitura

PULB
RTS

```

; Sub-rotina para um atraso de aproximadamente 10 ms

```

DELAY_10
      PSHX
      LDX  #2500      ; conta 20.000 ciclos
DELAY  DEX
      BNE  DELAY
      PULX
      RTS

```

O procedimento para apagamento é:

- 1 Escreva no registrador PPROG para selecionar o apagamento de um byte, uma linha ou toda a EEPROM.
- 2 Escreva no endereço da EEPROM dentro da faixa a ser apagada.
- 3 Escreva um 1 no registrador PPROG para ligar o bit EEPGM e, portanto, a tensão de apagamento.
- 4 Aguarde por 10 ms.
- 5 Escreva zeros no registrador PPROG para desligar todos os bits.

No caso do microcontrolador PIC com EEPROM interna, um programa para escrever dados nela é (Figura 18.9):

```

      bcf   STATUS, RPO      ; Mudar para o Banco 0 de dados
      mov.f Data, w         ; Carregar o dado a ser escrito
      movwf EEDATA
      movf  Addr, w         ; Carregar o endereço do dado a ser escrito
      movwf EEADR
      bsf   STATUS, RPO      ; Mudar para o Banco 1
      bcf   INTCON, GIE     ; Desabilitar interrupções
      bsf   EECON1, WREN    ; Habilitar a escrita
      movlw 55h             ; Sequência especial para habilitar
                          ; a escrita

      movwf EECON2
      movlw 0AAh
      movwf EECON2
      bsf   EECON1, WR      ; Início do ciclo de escrita
      bsf   INTCON, GIE     ; Habilitar interrupções
EE_EXIT btsf  EECON, WR      ; Verifica se a escrita está completa
      goto EE_EXIT         ; Se não, repete
      bsf   EECON, WREN    ; A escrita na EPROM está completa

```

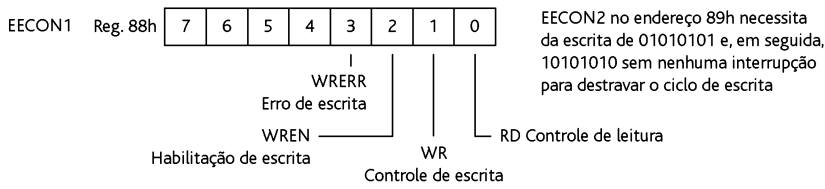



Figura 18.9 Registradores EECON.

Resumo

A coleção de instruções reconhecidas por um microprocessador é o seu **conjunto de instruções**. A série de instruções que são necessárias para executar uma determinada tarefa é denominada **programa**.

O microprocessador opera com o código binário. As instruções escritas no código binário são denominadas **código de máquina**. Um código abreviado de compreensão mais simples que o código binário é denominado **código mnemônico**. Este código é mais fácil de memorizar. Este código é denominado de **linguagem assembly**. Os programas nesta linguagem consistem de uma sequência de instruções, uma por linha, com cada uma contendo de um a quatro campos: rótulo (*label*), código de operação (*op-code*), operando e comentário. O **rótulo** é o nome pelo qual uma posição em particular da memória é denominada. O **código de operação** especifica como os dados são manipulados. O **operando** contém o endereço do dado a ser manipulado. O campo de **comentários** possibilita ao programador incluir comentários que podem tornar o programa mais fácil de ser entendido pelo leitor.

Problemas

- 18.1 Usando a seguinte parte do conjunto de instruções de um fabricante (6800), determine os códigos de máquina necessários para fazer a operação de adição com *carry* no modo de endereçamento (a) imediato, (b) direto.

Operação	Mnemônico	Modo de endereçamento					
		IMEDIATO			DIRETO		
		OP	~	#	OP	~	#
Soma com <i>carry</i>	ADC A	88	2	2	99	3	2

- 18.2 A operação clear, dentro do conjunto de instruções do processador 6800 da Motorola, tem uma entrada apenas na coluna do modo de endereçamento implícito. Qual é o significado disto?

- 18.3** Quais são os mnemônicos do, por exemplo, 6800 da Motorola para (a) zerar o registro A, (b) armazenar o acumulador A, (c) carregar o acumulador A, (d) comparar os acumuladores, (e) carregar o registrador de índice?
- 18.4** Escreva uma linha de programa assembly para (a) carregar o acumulador com 20 (hex), (b) decrementar o acumulador A, (c) zerar o endereço \$0020, (d) somar ao acumulador o número do endereço \$0020.
- 18.5** Explique as operações especificadas pelas seguintes instruções: (a) STA B \$35, (b) LDA A #\$F2, (c) CLC, (d) INC A, (e) CMP A #\$C5, (f) CLR \$2000, (g) JMP 05,X.
- 18.6** Escreva programas em linguagem assembly para:
- (a) Subtrair um número hexa no endereço de memória 0050 de um número hexa na posição 0060 da memória e armazenar o resultado na posição 0070.
 - (b) Multiplicar dois números de 8 bits, localizados nos endereços 0020 e 0021, e armazenar o produto, de 8 bits, na posição 0022.
 - (c) Armazenar os números hexa de 0 a 10 em posições de memória começando no endereço 0020.
 - (d) Mover um bloco de 32 números que inicia no endereço \$2000 para uma nova área que inicia no endereço \$3000.
- 18.7** Escreva, em linguagem assembly, uma sub-rotina que pode ser usada para produzir um atraso de tempo e que possa ser ajustada para qualquer valor de tempo dentro de uma faixa.
- 18.8** Escreva, em linguagem assembly, uma rotina que pode ser usada de modo que, se a entrada de um sensor no endereço 2000 for nível alto, o programa salta para uma rotina que começa no endereço 3000, e se for nível baixo, o programa continua.

19

Linguagem C

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Compreender as principais características dos programas em C.
- Usar a linguagem C para escrever programas simples para microcontroladores.

19.1 Por que C?

Este capítulo tem como objetivo fornecer uma introdução à linguagem C e à elaboração de programas. C é uma linguagem de alto nível que é muito usada no lugar da linguagem assembly (veja Capítulo 18) na programação de microprocessadores. Ela tem vantagens quando comparada com a linguagem assembly por ser mais fácil de ser usada e o mesmo programa poder ser usado em diferentes microprocessadores; tudo o que é necessário para isto é usar o compilador apropriado para traduzir o programa em C na linguagem de máquina do microprocessador. A linguagem assembly é diferente para diferentes microprocessadores, enquanto que a linguagem C é padronizada pela ANSI (*American National Standards Institute*).

19.2 Estrutura de um programa

A Figura 19.1 mostra uma visão geral dos principais elementos de um programa em C. Existe um comando de pré-processador que evoca um arquivo padrão, seguido da função principal. Dentro desta função principal existem outras funções que são acessadas como sub-rotinas. Cada função contém diversas declarações.

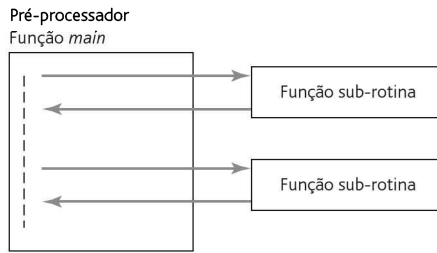


Figura 19.1 Estrutura de um programa em C.

19.2.1 Características importantes

A seguir abordamos as características importantes dos programas escritos em linguagem C. Note que, nestes programas, espaços e mudanças de linha (enter) são ignorados pelo compilador e são usados simplesmente por conveniência do programador para facilitar a leitura do programa.

1 Palavras-chave

Em C, certas palavras são reservadas como palavras-chave com significados específicos. Por exemplo, *int* é usada para indicar que se deseja fazer uso de um valor inteiro; *if* é usada quando um programa pode mudar de direção com base na decisão, que pode ser falsa ou verdadeira. A linguagem C necessita que todos estas palavras sejam escritas em letras minúsculas não podem ser usadas para outra finalidade no programa. Vemos a seguir as palavras-chave no padrão ANSI C:

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

2 Declarações

Estas são entradas com as quais um programa é construído. Cada declaração é finalizada com um ponto e vírgula. As declarações podem ser agrupadas em blocos delimitados por chaves, ou seja, { }. Portanto, para um grupo com duas declarações, temos:

```
{
    statement 1;
    statement 2;
}
```

3 Funções

O termo **função** é usado para um bloco de código de programa que realiza um conjunto específico de ações e tem um nome pelo qual pode ser identificado (como uma sub-rotina em um programa em linguagem assembly). Uma função

é escrita como um nome seguido por parênteses, ou seja, nome(). Os parênteses encerram os argumentos; o argumento de uma função é um valor que é passado para a função quando ela é chamada. Uma função é executada ao ser invocada pelo nome na declaração do programa. Por exemplo, podemos ter a declaração:

```
printf("Mecatrônica");
```

Isto significa que a palavra Mecatrônica é passada para a função printf(), que é uma função existente chamada pelo comando do pré-processador, e, como resultado, a palavra é mostrada na tela. Para indicar que os caracteres formam uma sequência, como, por exemplo, os que formam a palavra Mecatrônica, eles são colocados entre aspas.

4 Retorno

Uma função pode retornar um valor ao ser executada. O **tipo de retorno** aparece em frente ao nome da função, especificando o tipo do valor a ser retornado quando a execução da função é concluída. Por exemplo, int main() é usada para o retorno de um inteiro a partir da função principal. O tipo de retorno pode ser especificado como *void* (vazio) se a função não retornar um valor, como, por exemplo, void main(void). Frequentemente um arquivo de cabeçalho contém esta informação de retorno e, desta forma, ela não precisa ser especificada para as funções definidas pelo arquivo de cabeçalho.

Para retornar um valor a partir de uma função que volta no ponto de chamada dela, a palavra-chave return é usada, como, por exemplo, para retornar o resultado:

```
return result;
```

A declaração return termina a função.

5 Funções de biblioteca padrão

Pacotes em C são fornecidos com bibliotecas que contêm um grande número de funções predefinidas, poupando esforços por não se ter que escrevê-las. Elas podem ser chamadas pelo nome delas. Para usar o conteúdo de qualquer biblioteca em particular, ela tem que ser especificada no arquivo de cabeçalho. Como exemplos destas bibliotecas, citamos as seguintes:

```
math.h para funções matemáticas
stdio.h para funções de entrada e saída
time.h para funções de data e tempo
```

Por exemplo, a função printf() é uma função que pode ser chamada a partir da biblioteca stdio.h e é a função que imprime na tela do monitor. Uma outra função é scanf(), que pode ser usada para ler dados a partir do teclado.

6 Pré-processador

O **pré-processador** é um programa que é identificado por **comandos pré-processadores** de modo que ele é executado antes da compilação. Todos estes comandos são identificados por terem o símbolo # no início da linha. Portanto, podemos ter:

```
# include < >
```

para incluir o nome do arquivo entre os sinais de menor e maior (< >). Quando este comando é executado, o arquivo especificado é inserido no programa. Ele é

usado frequentemente para acrescentar os conteúdos de arquivos de cabeçalho padrão, fornecendo declarações e definições para habilitar funções de biblioteca padrão a serem usadas. A entrada deveria ser:

```
# include <stdio.h>
```

Como ilustração, considere o seguinte programa simples:

```
# include <stdio.h>
main( )
{
    printf("Mecatrônica");
}
```

Antes de iniciar o programa principal o arquivo `stdio.h` é acrescentado. Portanto, quando o programa principal iniciar poderemos acessar a função `printf()` que resulta na palavra `Mecatrônica` escrita na tela.

Um outro tipo de comando do pré-processador é:

```
# define pi 3.14
```

isto pode ser usado para definir valores que poderão ser inseridos sempre que um símbolo em particular for encontrado no programa. Portanto, sempre que `pi` for encontrado, o valor `3,14` será usado.

```
# define square(x) (x)*(x)
```

substitui o termo `square` no programa por `(x)*(x)`.

7 Função *main*

Cada programa em C tem que ter uma função chamada `main()`. Esta função é a que exerce controle quando o programa é executado e é a primeira função a ser chamada. A execução começa com esta primeira declaração. Outras funções podem ser chamadas dentro dela, sendo uma de cada vez executada e retornando o controle para a função `main`. A declaração:

```
void main(void)
```

indica que nenhum resultado será retornado para o programa `main` e não há argumento. Por convenção, um valor de retorno `0` a partir de `main()` é usado para indicar o término normal do programa, ou seja, a entrada:

```
return 0;
```

8 Comentários

`/* e */` são usados para encerrar comentários. Portanto, podemos ter uma entrada como:

```
/* Programa principal a seguir */
```

Os comentários são ignorados pelo compilador e são usados apenas para permitir ao programador compreender um programa mais facilmente. Os comentários podem ter uma extensão maior do que uma linha, como por exemplo:

```
/* Um exemplo de um programa usado para
ilustrar a programação */
```

9 Variáveis

Uma **variável** é um nome dado a uma posição de memória que pode armazenar valores. As variáveis que podem armazenar caracteres são especificadas com a palavra chave *char*. Esta variável tem uma extensão de 8 bits e geralmente é usada para armazenar um único caractere. Inteiros sinalizados, ou seja, sem partes fracionárias e que são sinalizados para indicar positivo ou negativo, são especificados usando-se a palavra chave *int*. A palavra chave *float* é usada para números de ponto flutuante. Estes números têm uma parte fracionária. A palavra chave *double* também é usada para números de ponto flutuante, mas com o dobro da quantidade de dígitos significativos de *float*. Para declarar uma variável, o tipo é inserido antes do nome da variável, como, por exemplo:

```
int counter;
```

Este exemplo declara a variável “counter” como sendo do tipo inteiro. Como outro exemplo, podemos ter:

```
float x, y;
```

Este exemplo indica que as variáveis *x* e *y* são números de ponto flutuante.

10 Atribuições

Uma declaração de atribuição atribui o valor da expressão à direita do sinal de = à variável à esquerda. Por exemplo, $a = 2$ atribui o valor 2 à variável *a*.

11 Operadores aritméticos

Os operadores aritméticos usados são: adição (+), subtração (−), multiplicação (*), divisão (/), módulo (%), incremento (++) e decremento (− −). Os operadores de incremento aumentam o valor da variável em 1 e os de decremento a diminuem em 1. As regras normais da aritmética são mantidas para a precedência de operações. Por exemplo, $2*4 + 6/2$ resulta em 11. Um exemplo de um programa envolvendo operadores aritméticos é mostrado a seguir:

```
/*programa para determinar a área de um círculo*/

#include <stdio.h> /*identifica a biblioteca de função*/

int raio, área /*as variáveis raio e área são inteiros*/

int main(void) /*começa o programa principal, int especifica
que um valor inteiro será retornado, void indica
que main( ) não tem parâmetros*/
{
    printf("Inserir raio:"); /*"Inserir raio" na tela*/
    scanf("%d", &raio); /*Lê um inteiro do
teclado e o associa à variável raio*/
    área=3.14 *raio *raio; /*Calcula a área*/
    printf("\nArea = %d", área); /*Imprime em uma nova linha Área
= e coloca o valor numérico de área*/
    return 0; /*retorna ao ponto da chamada*/
}
```

12 Operadores relacionais

Os operadores relacionais são usados para comparar expressões, fazer perguntas como “x é igual a y?” ou “x é maior do que 10?”. Os operadores relacionais são: igual a (==), é diferente de (!=), é menor do que (<), é menor ou igual a (<=), é maior do que (>), é maior ou igual a (>=). Note que == tem que ser usado quando perguntamos se duas variáveis são iguais, e = é usado em atribuições quando dizemos que elas são iguais. Por exemplo, podemos ter a pergunta “x é igual a 2?” e representar isto por (a==2).

13 Operadores lógicos

Os operadores lógicos são:

Operador	Símbolo
AND	&&
OR	
NOT	!

Note que, em C, o resultado é igual a 1 se verdadeiro e igual a 0 se falso.

14 Operações binárias

Os operadores binários tratam seus operandos como uma série de bits individuais em vez de valores numéricos, comparando os bits correspondentes em cada operando, e trabalhando apenas com variáveis do tipo inteiro. Estes operadores são:

Operação binária	Símbolo
AND	&
OR	
EX-OR	^
NOT	~
Deslocamento à direita	➤
Deslocamento à esquerda	➤

Portanto, por exemplo, podemos ter a declaração:

```
portA = portA | 0x0c;
```

O prefixo 0x é usado para indicar que o 0c é um valor hexa, sendo 0000 1100 em binário. A operação OR com port A é portanto um número binário que força os bits 2 e 3 em nível alto, sendo que todos os outros bits permanecem inalterados.

```
portA = portA ^ 1;
```

Esta declaração faz com que todos os bits, exceto o bit 1 de port A, permaneçam inalterados. Se o bit 0 for 1 em port A, a EX-ORO força para 0, e se ele for 0, força-o para 1.

15 String

Uma sequência de caracteres entre aspas, ou seja, “ ”, é denominada string. Conforme o termo diz, os caracteres entre aspas são tratados como uma entidade conjugada. Por exemplo, podemos ter:


```
printf("Sum = %d", x)
```

O argumento entre () especifica o que é passado para a função print. Existem dois argumentos, sendo os dois separados por vírgula. O primeiro argumento é a string entre aspas e especifica como a saída será apresentada, e o %d especifica que a variável é para ser mostrada como inteiro decimal. Outros formatos de especificação são:

%c	caractere
%d	inteiro decimal sinalizado
%e	notação científica
%f	decimal de ponto flutuante
%o	octal não sinalizado
%s	string de caracteres
%u	inteiro decimal não sinalizado
%x	hexadecimal não sinalizado
%%	imprime um sinal %

O outro argumento x especifica o valor a ser mostrado.

Como outro exemplo, considere a declaração:

```
scanf("%d", &x);
```

lê um inteiro decimal do teclado e o atribui a uma variável x inteiro. O símbolo & antes de x é o operador “endereço de”. Quando colocado antes do nome de uma variável, retorna o endereço da variável. Portanto, o comando busca pelo dado e armazena o item usando o endereço dado.

16 Sequências de escape

Sequências de escape são caracteres que “escapam” de uma interpretação padrão de caracteres e são usados para controlar a posição das saídas no display movendo o cursor na tela ou indicando tratamentos especiais. Portanto, podemos ter:

```
printf("\nSum = %d", d)
```

com o \n indicando que uma nova linha deve ser usada quando o item for impresso na tela. As sequências de escape normalmente usadas são:

\a	campainha (<i>bell</i>)
\b	retrocesso (<i>backspace</i>)
\n	nova linha (<i>new line</i>)
\t	tabulação horizontal
\\	barra invertida
\?	sinal de interrogação
\'	apóstrofo

19.2.2 Exemplo de um programa em C

Eis a seguir um exemplo de um programa simples que ilustra o uso de alguns dos termos vistos:

```
/*Um programa simples em C*/
# include <stdio.h>
```

```

void main(void)
{
int a, b, c, d; /*a, b, c são inteiros*/
a = 4; /*a a é atribuído o valor 4*/
b = 3; /*a b é atribuído o valor 3*/
c = 5; /*a c é atribuído o valor 5*/
d = a * b * c; /*a d é atribuído o valor de a * b * c*/
printf("a * b * c = %d\n", d);
}

```

A declaração `int a, b, c, d;` declara as variáveis `a, b, c` e `d` como do tipo inteiro. As declarações `a = 4`, `b = 3`, `c = 5` atribuem valores iniciais às variáveis, com o sinal = sendo usado para indicar atribuição. A declaração `d = a*b*c` indica que `a` é multiplicada por `b` e, em seguida, por `c` e armazenada em `d`. O comando `printf` na declaração `printf("a*b*c=%d\n",d)` é a função de apresentação na tela. O argumento contém `%d`, que indica que ela é convertida em um valor decimal para ser apresentada. Portanto, ela imprime `a*b*c = 60`. O caractere `\n` no final da string indica que uma nova linha deve ser inserida neste ponto.

19.3 Desvios e loops

Declarações para permitir desvios e *loops* em programas incluem *if*, *if/else*, *for*, *while* e *switch*.

1 If

A declaração *if* permite desvios (Figura 19.2(a)). Por exemplo, se uma expressão for verdadeira, então a declaração é executada; se não for verdadeira, o programa continua na próxima declaração. Portanto, podemos ter declarações da forma:

```

If (condição 1 == condição 2);
printf ("\nCondição OK.");

```

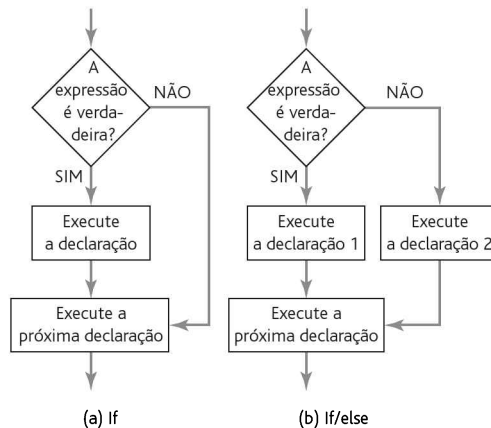


Figura 19.2 (a) If, (b) if/else.

Um exemplo de um programa envolvendo a declaração `if` é o seguinte:

```
#include <studio.h>

int x, y;
main()
{
    printf("\nInserir um valor inteiro para x:");
    scanf("%d", &x);
    printf("\nInserir um valor inteiro para y:");
    scanf("%d", &y);
    if (x == y)
        printf("x é igual a y");
    if(x > y)
        printf("x é maior do que y");
    if(x < y)
        printf("x é menor do que y");
    return 0;
}
```

A tela mostra Inserir um valor inteiro para x: e, em seguida, um valor é inserido pelo teclado. A tela mostra depois Inserir um valor inteiro para y: e, em seguida, um valor é inserido pelo teclado. A sequência *if* determina então se os valores inseridos são iguais ou qual é o maior e mostra o resultado na tela.

2 *If/else*

A declaração *if* pode ser combinada com a declaração *else*. Isto permite que uma declaração seja executada se o resultado for sim e o outro se for não (Figura 19.2(b)). Portanto, podemos ter:

```
#include <studio.h>

main()
{
    int temp;
    if(temp > 50)
        printf("Advertência");
    else
        printf("Sistema OK");
}
```

3 *For*

O termo *loop* é usado para a execução de uma sequência de declarações até que uma determinada condição desejada seja verdadeira, ou falsa. A Figura 19.3(a) ilustra isto. Uma forma de escrever declarações para um *loop* é usando a função *for*. A forma geral desta declaração é:

```
for(expressão de inicialização; expressão de teste; expressão de incremento)
    declaração do loop;
```

Portanto, podemos ter:

```
#include <studio.h>
```

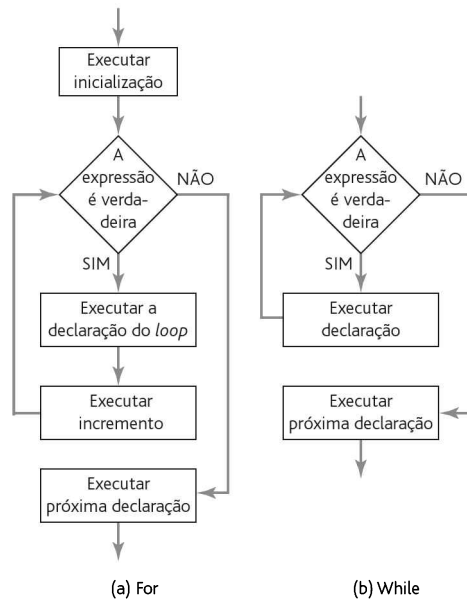


Figura 19.3 (a) For, (b) while.

```

int count

main()
{
    for(count = 0; count < 7; count ++)
        printf("\n%d", count);
}
  
```

Inicialmente o contador (count) é 0 e passa a ser incrementado de 1, e então o programa entra em *loop* repetindo a declaração for enquanto o count for menor do que 7. O resultado é que a tela mostra 0 1 2 3 4 5 6, sendo cada número em uma linha.

4 *While*

A declaração *while* permite que o *loop* seja repetido continuamente enquanto a expressão for verdadeira (Figura 19.3(b)). Quando a expressão passa a ser falsa, o programa continua com a declaração seguinte ao *loop*. Como uma ilustração, podemos ter o seguinte programa em que a declaração *while* é usada para contar enquanto o número é menor do que 7, mostrando os resultados:

```

#include <studio.h>

int count;
int main( );
{
  
```

```

count = 1;
while(count < 7)
{
    printf("\n%d", count);
    count + +;
}
return 0;
}

```

A apresentação na tela é 1 2 3 4 5 6, com cada número em uma linha separada.

5 Switch

A declaração *switch* permite a seleção entre várias alternativas com a condição de teste entre parênteses. As escolhas possíveis são identificadas por rótulos *case*, que especificam o valor esperado da condição de teste. Por exemplo, podemos ter a situação em que se o *case 1* ocorrer, executamos a declaração 1, se o *case 2* ocorrer, executamos a declaração 2, etc. Se a expressão não for igual a qualquer um dos *cases*, então a declaração *default* é executada. Após cada declaração de *case* há normalmente uma declaração *break* para transferir a execução para a declaração após o *switch* e parar a continuação da lista de *cases* dentro do *switch*. A sequência é portanto (Figura 19.4):

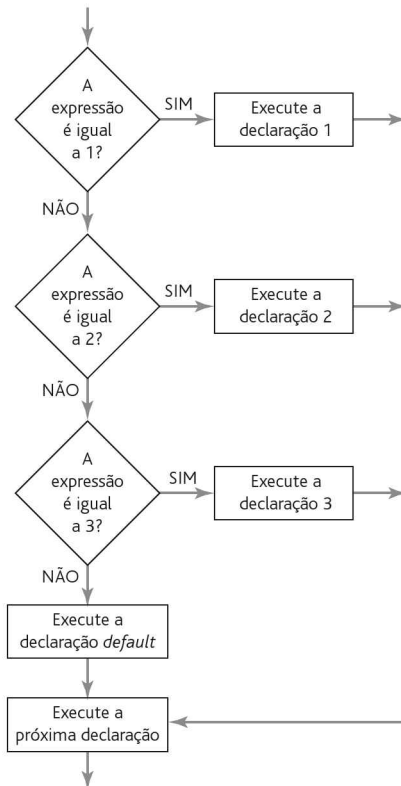


Figura 19.4 Switch.

```
switch(expressão)
{
    case 1;
        declaração 1;
        break;
    case 2;
        declaração 2;
        break;
    case 3;
        declaração 3;
        break;
    default;
        declaração default;
}
próxima declaração
```

Veja a seguir um exemplo de programa que reconhece os números 1, 2 e 3 e mostra qual número foi digitado no teclado:

```
#include <stdio.h>

int main ( );
{
    int x;

    printf("Digite o número 0, 1, 2 ou 3: ");
    scanf("%d", &x);

    switch (x)
    {
        case 1:
            printf("Um");
            break;
        case 2:
            printf("Dois");
            break;
        case 3:
            printf("Três");
            break;
        default;
            printf("Não é 1, 2 ou 3");
    }
    return 0;
}
```

19.4 Arranjos

Suponha que desejamos registrar a temperatura ao meio-dia por uma semana e depois podermos determinar a temperatura que corresponde a um determinado dia. Podemos fazer isto usando um arranjo (*array*). Um **arranjo** é uma coleção de posi-

ções de armazenamento de dados em que cada uma tem o mesmo tipo de dado e são referenciadas pelo mesmo nome. Para declarar um arranjo com o nome Temperatura para armazenar valores do tipo float, usamos a declaração:

```
float Temperatura[7];
```

O tamanho do arranjo é indicado entre colchetes imediatamente após o nome do arranjo. Neste caso, 7 foi usado para os dados dos sete dias da semana. Os elementos individuais no arranjo são referenciados por um valor de índice. O primeiro elemento tem o número 0, o segundo 1, e assim por diante até o último elemento em uma sequência n que será $n - 1$. A Figura 19.5(a) mostra a forma de um arranjo sequencial. Para armazenar valores no arranjo, podemos escrever:

```
temperatura[0] = 22.1;
temperatura[1] = 20.4;
etc.
```

Se você quiser usar `scanf()` para entrar com um valor em um elemento do arranjo, coloque `&` em frente do nome do arranjo, como, por exemplo:

```
scanf("%d", &temperatura [3]);
```

Veja a seguir um exemplo de um programa simples para armazenar e mostrar os quadrados (sqrs) dos números 0, 1, 2, 3 e 4:

```
#include <stdio.h>

int main(void)
{
    int sqrs[5];
    int x;

    for(x = 1; x<5; x++)
        sqrs[x - 1] = x * x;
    for(x = 0; x < 4; x++)
        printf("%d", sqrs[x]);

    return 0;
}
```

Aos arranjos podem ser dados valores iniciais quando declarados pela primeira vez, como, por exemplo:

```
int array[7] = {10, 12, 15, 11, 10, 14, 12};
```

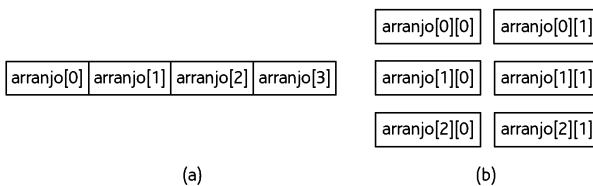


Figura 19.5 (a) Um arranjo sequencial de quatro elementos, (b) um arranjo de duas dimensões.

Se omitirmos o tamanho do arranjo, o compilador criará um grande o suficiente para gravar os valores de inicialização:

```
int array[] = {10, 12, 15, 11, 10, 14, 12};
```

Arranjos multidimensionais podem ser usados. Por exemplo, uma tabela de dados é um arranjo de duas dimensões (Figura 19.5(b)), onde x representa a linha e y , a coluna, e é escrito como:

```
array[x][y];
```

19.5 Ponteiros

Cada posição de memória tem um único endereço, que é o meio pelo qual o dado armazenado em uma posição pode ser acessado. Um **ponteiro** é um tipo especial de variável que pode armazenar o endereço de uma outra variável. Portanto, se uma variável denominada p contém o endereço de uma outra denominada x , então p é denominada *ponteiro* de x . Portanto, se x estiver no endereço 100 na memória, então p deve ter o valor 100. Um ponteiro é uma variável e, assim como todas as outras variáveis, tem que ser declarado antes de ser usado. A declaração de um ponteiro tem a seguinte forma:

```
type *name;
```

O $*$ indica que o nome (*name*) se refere a um ponteiro. Normalmente os nomes usados para ponteiros são escritos com o prefixo p , ou seja, da forma $pname$. Portanto, podemos ter:

```
int *pnumber;
```

Para inicializar um ponteiro dando a ele um endereço para o qual apontará, podemos usar $&$, que é o endereço do operador, em uma declaração do tipo:

```
pointer = &variable;
```

O seguinte programa ilustra isto:

```
#include <stdio.h>

int main(void)
{
    int *p, x;
    x = 12;
    p = &x; /*atribui a p o endereço de x*/
    printf("%d", *p); /*mostra o valor de x usando ponteiro*/

    return 0;
}
```

Portanto, este programa mostra o número 12 na tela. O acesso ao conteúdo de uma variável usando-se um ponteiro, como neste programa, é denominado de **acesso indireto**. O processo de acesso a dados em uma variável apontada por um ponteiro é denominado **derreferenciando** o ponteiro.

19.5.1 Aritmética com ponteiros

As variáveis ponteiro podem ter os operadores aritméticos $+$, $-$, $++$ e $--$ aplicados a elas. O incremento ou decremento de um ponteiro faz ele apontar para o próximo, ou o anterior, elemento do arranjo. Portanto, para incrementar um ponteiro para o próximo item de um arranjo, podemos usar:

```
pa++; /*usando o operador de incremento de 1*/
```

ou

```
pa = pa + 1; /*soma 1*/
```

19.5.2 Ponteiros e arranjos

Os ponteiros podem ser usados para acessar elementos individuais em um arranjo. O programa a seguir mostra este tipo de acesso:

```
#include <stdio.h>

int main(void)
{
    int x[5] = (0, 2, 4, 6, 8);
    int *p;
    p = x; /*atribui a p o endereço inicial de x*/
    printf("%d %d", x[0], x[2]);

    return 0;
}
```

A declaração `printf("%d %d", x[0], x[2]);` resulta em apontar para o endereço dado por `x`, e portanto os valores nos endereços `[0]` e `[2]` são mostrados, ou seja, 0 e 4, em linhas separadas.

19.6 Desenvolvimento de programas

No desenvolvimento de programas, o objetivo é chegar a um conjunto de instruções em linguagem de máquina que pode ser usado para operar um sistema microprocessado/microcontrolado. Essas instruções são denominadas **arquivo executável**. Para chegar a este arquivo a seguinte sequência de eventos ocorre:

1 Criação do código fonte

Corresponde à escrita de uma sequência de declarações em C que se constituirão no programa. Muitos compiladores agregam um editor, e assim o programador pode simplesmente digitar o código fonte a partir do teclado. Caso contrário, pode ser usado um programa como o Bloco de Notas do Microsoft Windows. O uso de um processador de texto pode apresentar problemas por causa da informação adicional de formatação que é incluída, podendo impedir a compilação a menos que o arquivo seja salvo sem formatação.

2 *Compilação do código fonte*

Uma vez escrito o código fonte, o programador pode compilar o arquivo diretamente traduzindo-o para o código de máquina. Antes de iniciar o processo de compilação, todos os comandos do pré-processador são executados. O compilador pode detectar diferentes tipos de erro durante a tradução e gerar mensagens indicando os erros. Algumas vezes, um único erro pode resultar em uma sequência de erros. Os erros geralmente envolvem o retorno ao estágio de edição do código fonte. O compilador armazena o código de máquina resultante em um outro arquivo.

3 *Criação do arquivo executável*

O compilador é usado para reunir (*link*) o código gerado com as funções de biblioteca para a obtenção de um único arquivo executável. O programa é armazenado como o arquivo executável.

19.6.1 Arquivos de cabeçalho

Os comandos do pré-processador são usados no início de um programa para definir as funções usadas nele; desta forma eles podem ser referenciados simplesmente por rótulos. Entretanto, para evitar escrever longas listas de funções padrão para cada programa, uma instrução do pré-processador pode ser usada para indicar que um arquivo, o qual inclui as funções padrão relevantes, pode ser usado. Tudo o que é necessário fazer é indicar qual arquivo de funções padrão deve ser usado pelo compilador; este arquivo é um **cabeçalho**, visto que ele está no início do programa. Por exemplo, `<stdio.h>` contém funções de entrada e saída padrão, como, por exemplo, `gets` (entra, ou seja, lê uma linha de um dispositivo), `puts` (saída, ou seja, escreve uma linha no dispositivo) e `scanf` (lê dados); `<math.h>` contém funções matemáticas como `cos` (co-seno), `sin` (seno), `tan` (tangente), `exp` (exponencial) e `sqrt` (raiz quadrada).

Arquivos de cabeçalho também estão disponíveis para definir os registradores e portas do microcontrolador e evitam que o programador tenha que definir cada registrador e porta escrevendo linhas do pré-processador para cada um deles. Portanto, para um microcontrolador 8051 da Intel, podemos ter o cabeçalho `<reg.51.h>`; isto define os registradores, como, por exemplo, as portas P0, P1, P2 e P3, e os bits individuais dos registradores bit-endereçáveis, como, por exemplo, os bits TF1, TR1, TF0, TR0, IE1, IT1, IE0 e IT0 no registrador TCON. Portanto, podemos escrever instruções com referência às entradas/saídas da porta 0 simplesmente usando o rótulo P0 ou TF1 em referência ao bit TF1 no registrador TCON. De forma similar, para o M68HC11E9 da Motorola, o cabeçalho `<hc11e9.h>` define os registradores, como, por exemplo, PORTA, PORTB, PORTC e PORTD, e bits individuais nos registradores bit-endereçáveis, como, por exemplo, os bits STAF, STAI, CWOM, HNDS, OIN, PLS, EGA e INVB no registrador PIOC. Portanto, podemos, por exemplo, escrever instruções com referência à porta de entrada/saída A simplesmente usando o rótulo PORTA. As bibliotecas também podem fornecer rotinas para auxiliar no uso de dispositivos periféricos, como teclados e displays de cristal líquido (LCDs).

O programa principal escrito para um microcontrolador em específico pode, como resultado da mudança dos arquivos de cabeçalho, ser facilmente adaptado para rodar em outro microcontrolador. As bibliotecas fazem os programas em C serem altamente portáteis.

19.7 Exemplos de programas

Apresentamos a seguir exemplos de programas escritos em C para sistemas micro-controlados.

19.7.1 Ligando e desligando um motor

Considere a programação de um microcontrolador M68HC11 para ligar e desligar um motor CC. A porta C é usada como entrada e a porta B como saída para o motor, através de um driver adequado (Figura 19.6). O botão de iniciar é conectado em PC0 para mudar a entrada de 1 para 0 quando o motor for acionado. O botão de parar é conectado em PC1 para mudar a entrada de 1 para 0 quando o motor for parado. O registrador de direção dos dados na porta C, DDRC, tem que ser colocado em 0 de forma que a porta C seja configurada para entradas.

Um programa pode ser:

```
#include <hc11e9.h> /*inclui o arquivo de cabeçalho*/

void main(void)
{
    PORTB.PB0 &=0; /*inicialmente garante o motor desligado*/
    DDRC = 0; /*configura a porta C como entrada*/
    while (1) /*repete enquanto esta condição se mantém*/
    {
        if (PORTC.PC0 == 0) /*o botão de início está pressionado?*/
            PORTB.PB0 |=1; /*ativa a saída se pressionado*/
        else if(PORTC.PC1 == 0) /*o botão de parar está pressionado?*/
            PORTB.PB0 &=0; /*desativa a saída se pressionado*/
    }
}
```

Note que `|=` é o operador OR e faz com que o bit do resultado seja 0 apenas se os bits correspondentes nos dois operandos forem 0, caso contrário ele será 1. Ele é usado para ligar, ou setar, um ou mais bits em um valor (0 ou 1). Portanto, `Port B.PB0 |=1`

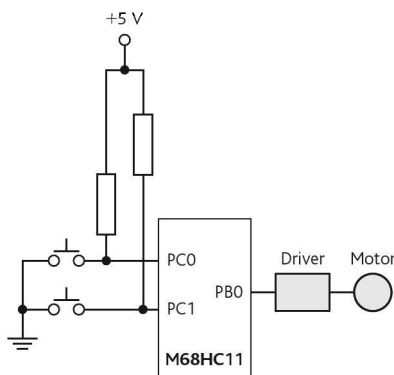


Figura 19.6 Controle de motor.

é a operação OR entre 1 e o valor de PB0 que liga o motor. Esta é uma forma útil de acionar bits simultaneamente em uma porta. O $\&$ em $\text{PORTB.PB0} \&=0$ é usado para fazer a operação AND entre o bit PB0 e 0 e, como PB0 é sempre 1, o valor 0 é associado a PORTB.PB0 .

19.7.2 Leitura de um canal ADC

Considere a tarefa de programar um microcontrolador (M68HC11) de modo que um único canal do ADC possa ser lido. O M68HC11 contém um ADC de 8 bit com 8 canais multiplexados com entradas via porta E (Figura 19.7). O registrador de controle/status do ADC, ADCTL, contém o *flag* de conversão finalizada, CCF, no bit 7 e outros bits para controlar o multiplexador e a varredura de canais. Quando $\text{CCF} = 0$, a conversão ainda não está completa, e quando igual a 1, significa que está completa. A conversão analógico-digital inicia ao se escrever 1 no bit ADPU no registrador OPTION. Entretanto, o ADC tem que ser ligado pelo menos 100 μs antes da leitura de um valor.

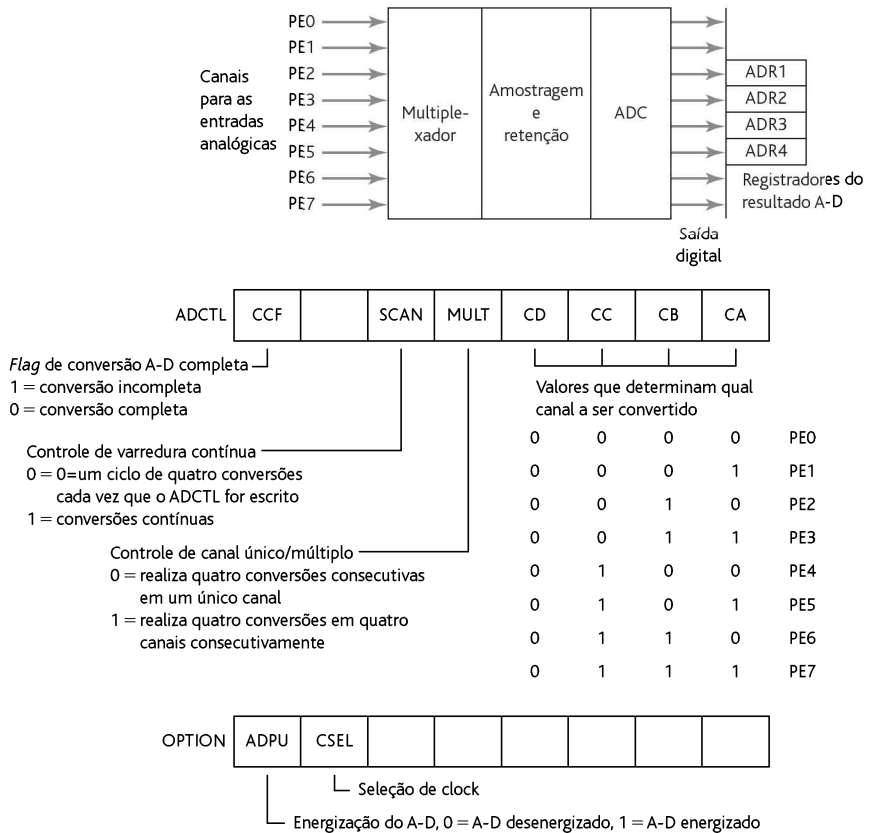


Figura 19.7 ADC.

Para converter a entrada analógica em PE0, os primeiros 4 bits no registrador ADCTL, ou seja, CA, CB, CC e CD, têm que ser setados em 0. Quando a operação for para converter apenas um canal, o bit 5, SCAN, pode ser setado em 0 e o bit 4, MULT, em 0. Um programa simples para ler um determinado canal pode envolver, após a energização do ADC, colocar todos os bits do ADCTL em zero, colocar o número do canal e ler a entrada enquanto CCF for 0. O programa pode ser como a seguir:

```
#include <hc11e9.h> /*o arquivo de cabeçalho*/

void main(void)
{
    unsigned int k; /*número do canal de entrada*/

    OPTION=0; /*esta e a próxima linha ligam o ADC*/
    OPTION.ADPU=1;
    ADCTL &=~0x7; /*zera os bits*/
    ADCTL |=k; /*coloca o número do canal a ser lido*/
    while (ADCTL.CCF==0);
    return ADR1; /*retorna o valor convertido para o endereço 1*/
}
```

Note que `~` é o operador de complemento, e sua ação é inverter cada bit no operando, trocando zeros por uns e vice-versa. Portanto, o bit 7 é setado. `|` é o operador OR e faz um bit no resultado ser 0 apenas se os bits correspondentes dos dois operandos forem 0; caso contrário, ele é feito igual a 1. Ele é usado para ligar, ou setar, um ou mais bits no valor. Neste caso, sendo `k = 1`, ele seta apenas CA para 1. Uma sub-rotina de atraso pode ser incluída para garantir que, após a energização, o valor não será lido tão rapidamente.

Resumo

C é uma linguagem de alto nível que tem vantagens quando comparada com a linguagem assembly por ser mais fácil de ser usada e o mesmo programa poder ser usado com diferentes microprocessadores; tudo o que é necessário para isto é usar o compilador apropriado para traduzir o programa em C para a linguagem de máquina do microprocessador em questão. A linguagem assembly é diferente para diferentes microprocessadores, enquanto que a linguagem C é padronizada.

Os pacotes em C são fornecidos com bibliotecas que contêm um grande número de funções predefinidas em C. Para usar o conteúdo de uma biblioteca em particular, esta tem que ser especificada em um arquivo de cabeçalho. Cada programa em C tem que ter uma função denominada `main()`; esta exerce o controle quando o programa é executado e é a primeira função a ser chamada. Um programa é constituído de declarações, com cada uma terminada por ponto e vírgula. As declarações podem ser agrupadas em blocos colocando-as entre chaves, ou seja, `{ }`.

Problemas

19.1 As questões a seguir são todas relacionadas a componentes de programas.

(a) Descreva o significado de int na declaração:

```
int counter;
```

(b) Descreva o significado da seguinte declaração:

```
num = 10
```

(c) Descreva qual é o resultado da seguinte declaração:

```
printf("Nome");
```

(d) Descreva qual é o resultado da seguinte declaração:

```
printf("Number %d", 12);
```

(e) Descreva qual é o efeito de:

```
#include <stdio.h>
```

19.2 Para o programa a seguir, quais são as razões para a inclusão da linha (a) #include<stdio.h>, (b) os símbolos { e }, (c) o /d, e (d) o que aparecerá na tela quando o programa for executado?

```
#include <stdio.h>
```

```
main( )  
{  
    printf(/d"problema 3");  
}
```

19.3 Considerando o programa a seguir, o que será mostrado na tela?

```
#include <stdio.h>
```

```
int main(void);  
{  
    int num;  
    num = 20;  
  
    printf("O número é %d", num);  
    return 0;  
}
```

19.4 Escreva um programa que calcule a área de um retângulo dados o seu comprimento e a sua largura no prompt da tela e que, em seguida, mostre a resposta precedida das palavras "A área é"

19.5 Escreva um programa que mostre os números de 1 a 15, cada um em uma linha.

- 19.6 Explique os motivos das declarações no programa a seguir que faz a divisão entre dois números:

```
#include <stdio.h>

int main(void);
{
    int num1, num2;

    printf("Inserir o primeiro número:");
    scanf("%d", &num1);

    printf("Inserir o segundo número:");
    scanf("%d", &num2);

    if(num2 == 0)
        print f("Não existe divisão por zero")
    else
        printf("A resposta é: %d", num1/num2);

    return 0;
}
```

20

Sistemas de Entrada/Saída

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Identificar os requisitos de interface e como eles podem ser realizados: *buffers*, *handshaking*, varredura e interface serial.
- Explicar como as interrupções são usadas em microcontroladores.
- Explicar a função de adaptadores de interface de periféricos e ser capaz de programá-los em situações particulares.
- Explicar a função de adaptadores de interface de comunicação síncrona.

20.1 Interfaceamento

Quando um microprocessador é usado para controlar algum sistema, ele tem que receber informações de entrada, responder a elas e produzir sinais de saída para implementar a ação de controle necessária. Portanto, podem existir entradas a partir de sensores para fornecer dados e saídas para dispositivos externos como relés e motores. O termo **periférico** é usado para um dispositivo, como um sensor, teclado, atuador, etc., que é conectado a um microprocessador. Entretanto, normalmente não é possível conectar um dispositivo periférico diretamente ao sistema de barramento de um microprocessador devido a falta de compatibilidade nos formatos e níveis dos sinais. Por causa desta incompatibilidade, é usado um circuito conhecido com interface entre os periféricos e o microprocessador. A Figura 20.1 ilustra a configuração. Com a interface, a incompatibilidade é resolvida.

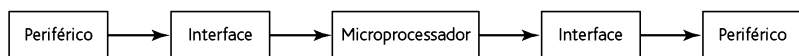


Figura 20.1 Utilização de interfaces.

Este capítulo discute a necessidade das interfaces e o Adaptador de Interface de Periférico MC6820 da Motorola, bastante usado, e o MC6850, que é um Adaptador de Interface para Comunicações Assíncronas.

20.2 Endereçamento de entrada/saída

Existem duas formas do microprocessador selecionar um dispositivo de entrada/saída (I/O). Alguns microprocessadores, como, por exemplo, o Z80 da Zilog, têm **entrada/saída isolada**, e instruções de entrada especiais com IN são usadas para ler um dispositivo de entrada, e instruções de saída especiais como OUT são usadas para escrever em dispositivos e saída. Por exemplo, com o Z80 podemos ter:

```
IN A,(B2)
```

para ler o dispositivo de entrada B2 e colocar o dado no acumulador A. Uma instrução de saída pode ser:

```
OUT (C), A
```

para escrever o dado do acumulador A na porta C.

O mais comum é os microprocessadores não possuírem instruções separadas para entrada e saída, mas usar as mesmas instruções de leitura e escrita na memória. Isto é denominado **entrada/saída mapeada na memória**. Com este método, cada dispositivo de entrada/saída tem um endereço, assim como uma posição de memória. O 68HC11 da Motorola, o 8051 da Intel e os microcontroladores PIC não possuem instruções de entrada/saída separadas e usam mapeamento na memória. Portanto, com o mapeamento podemos usar:

```
LDAA $1003
```

para ler a entrada de dados no endereço \$1003, e:

```
STAA $1004
```

para escrever o dado na saída no endereço \$1004.

Os microprocessadores usam portas paralelas para entrada ou saída de bytes de dados. Muitos periféricos necessitam normalmente de mais de uma porta. Isto pode ser porque a palavra de dados do periférico é maior do que a da CPU. Neste caso, a CPU tem que transferir os dados por segmentos. Por exemplo, se precisarmos de uma saída de 16 bit com uma CPU de 8 bits, o procedimento será:

- 1 A CPU prepara os oito bits mais significativos dos dados.
- 2 A CPU envia estes oito bits para a primeira porta.
- 3 A CPU prepara os oito bits menos significativos dos dados.
- 4 A CPU envia estes oito bits para a segunda porta.
- 5 Portanto, após algum atraso, todos os 16 bits estarão disponíveis para o periférico.

20.2.1 Registradores de entrada/saída

O microcontrolador 68HC11 da Motorola tem cinco portas (A, B, C, D e E) (veja a Seção 17.3.1). As portas A, C e D são bidirecionais e podem ser usadas para

entrada ou saída. A porta B é apenas saída e a porta E é apenas entrada. Se uma porta bidirecional será usada como entrada ou como saída depende dos valores dos bits no registrador de controle. Por exemplo, a porta A no endereço \$1000 é controlada pelo registrador de controle do acumulador de pulsos PACTL no endereço \$1026. Para configurar a porta A como entrada, é necessário que o bit 7 seja 0; para ser configurada com saída, este bit tem que ser 1 (veja a Figura 17.12). A porta C é bidirecional, e os 8 bits no seu registrador no endereço \$1003 são controlados pelos bits correspondentes no seu registrador de direção de dados no endereço \$1007. Quando o bit de direção de dado correspondente for 0, ela é uma entrada; quando for 1, ela é uma saída. A porta D é bidirecional e contém apenas seis linhas de entrada/saída no endereço \$1008. Ela é controlada por um registrador de direção da porta no endereço \$1009. A direção de cada linha é controlada pelo bit correspondente no registrador de controle; ele é colocado em 0 para configurar a porta como entrada, e em 1 para saída. Algumas portas também podem ser configuradas para executar outras funções através de outros bits no registrador de controle.

Para uma porta com direção fixa, como, por exemplo, a porta B no 68HC11 da Motorola, que é apenas saída, as instruções necessárias para transferir para a saída um determinado valor, como, por exemplo, \$FF, são simplesmente as que são necessárias para carregar o dado no endereço da saída. As instruções podem ser:

REGBAS	EQU	\$1000	; endereço base dos registradores de I/O
PORTB	EQU	\$04	; <i>offset</i> da porta B a partir de REGBAS
	LDX	#REGBAS	; carrega registrador de índice X
	LDAA	#\$FF	; carrega \$FF no acumulador
	STAA	PORTB,X	; armazena o valor no endereço da porta B

Para a porta E, que é apenas entrada, a instrução que lê um byte dela pode ser:

REGBAS	EQU	\$1000	; endereço base dos registradores de I/O
PORTE	EQU	\$0A	; <i>offset</i> da porta E a partir de REGBAS
	LDAA	PORTE,X	; carrega o valor da porta E no acumulador

Para uma porta bidirecional como a C, antes de podermos usá-la para uma entrada temos que configurá-la para que atue como entrada. Isto significa colocar todos os bits em 0. Portanto, podemos ter:

REGBAS	EQU	\$1000	; endereço base dos registradores de I/O
PORTC	EQU	\$03	; <i>offset</i> da porta C a partir de REGBAS
DDRC	EQU	\$07	; <i>offset</i> do registrador direcionador de dados
			; a partir de REGBAS
	CLR	DDRC,X	; zera todos os bits de DDRC

No microcontrolador 8051 da Intel (veja a Seção 17.3.2), há quatro portas de entrada/saída bidirecional paralela. Quando um bit da porta é usado como saída, o dado é apenas colocado no bit do registrador de função especial correspondente; quando ele é usado como entrada, um bit 1 tem que ser escrito em cada bit de interesse, portanto, FFH poderia ser escrito em toda a porta. Considere um exemplo de instruções do 8051 da Intel que ligam um LED quando um botão é pressionado. O botão tem entrada em P3.1 e saída em P3.0; o botão coloca nível baixo na entrada quando pressionado:

	SETB	P3.1	; escreve 1 em P3.1, configurando-a como ; entrada
LOOP	MOV	C,P3.1	; lê o estado do botão ; e o armazena no <i>flag de carry</i>
	CPL	C;	; complementa o <i>flag de carry</i>
	MOV	P3.0, C	; copia o estado do <i>carry</i> na saída
	SJMP	LOOP	; repete continuamente a sequência

No microcontrolador PIC, a direção dos sinais em suas portas bidirecionais é definida pelos registradores de direção TRIS (veja a Seção 17.3.3). Este registrador é setado em 1 para leitura e em 0 para escrita. Os registradores do PIC16C73/74 são organizados em dois bancos, e antes que um determinado registrador seja selecionado, o banco tem que ser escolhido setando-se o bit 5 no registrador STATUS. Este registrador está nos dois bancos, assim, não é necessário selecionar o banco para acessá-lo. Os registradores TRIS estão no banco 1 e os registradores PORT, no banco 0. Portanto, para configurar a porta B como saída, primeiro temos que selecionar o banco 1 e, em seguida, carregar TRISB com 0. Podemos selecionar o banco 0 e escrever em PORTB. O banco é selecionado setando-se o bit no registrador STATUS. As instruções para configurar a porta B como saída são:

Output	clrf	PORTB	; zerar todos os bits na porta B
	bsf	STATUS,RPO	; usa o registrador status para selecionar ; o banco 1 setando RPO em 1
	clrf	TRISB	; zerar os bits para configurar saída
	bcf	STATUS,RPO	; usa o registrador status para selecionar o ; banco 0 ; a porta B agora é uma saída com os bits em 0

20.3 Requisitos de interfaces

Apresentamos a seguir algumas características necessárias em um circuito de interface:

1 Buffer/isolação elétrica

Isto é necessário quando o periférico opera em uma tensão ou corrente diferente daquela em que o sistema de barramento do microprocessador opera ou há referências de terra diferentes. O termo *buffer* é usado para um dispositivo que fornece isolação ou amplificação de corrente ou tensão. Por exemplo, se a saída de um microprocessador estiver conectada na base de um transistor, a corrente de base necessária para comutar o transistor é maior do que a fornecida pelo microprocessador, exigindo que um *buffer* seja usado para elevar a corrente. Frequentemente há também a necessidade de isolação entre o microprocessador e um sistema de potência maior.

2 Controle de temporização

O controle de temporização é necessário quando a taxa de transferência de dados entre o periférico e o microprocessador são diferentes, como, por exemplo, na interface entre um microprocessador e um periférico mais lento. Isto é conseguido usando-se linhas especiais entre o microprocessador e o periférico para

controlar a temporização na transferência de dados. Estas linhas são denominadas **linhas de handshake** e o processo é o **handshaking**.

3 Conversão de código

Isto é necessário quando os códigos usados pelos periféricos são diferentes dos utilizados pelo microprocessador. Por exemplo, um display de LED pode necessitar de um decodificador para converter a saída BCD de um microprocessador em um código necessário para ativar os sete segmentos do display.

4 Alteração no número de linhas

Os microprocessadores operam com um tamanho fixo de palavra que pode ser 4, 8 ou 16 bits. Isto determina o número de linhas no barramento de dados do microprocessador. O equipamento periférico pode ter um número diferente de linhas, necessitando talvez de uma palavra maior do que a do microprocessador.

5 Transferência de dados de serial para paralelo e vice-versa

Em um microprocessador de 8 bit, os dados geralmente são manipulados de uma vez em grupos de 8 bits. Para transferir 8 bits simultaneamente para um periférico, é necessário um barramento de 8 bits. Esta forma de transferência de dados é denominada **transferência de dados em paralelo**. Entretanto, nem sempre é possível transferir dados desta forma. Por exemplo, os dados transferido pelo sistema de telefonia pública fazem uso apenas de uma via de dados. Os dados têm que ser transferidos sequencialmente 1 bit de cada vez. Esta forma de transferência é denominada **transferência de dados em série**. Esta transferência é mais lenta do que a paralela. Portanto, se a transferência de dados em série for usada, haverá a necessidade de converter os dados seriais que chegam no formato paralelo para o microprocessador e vice-versa para a saída a partir do microprocessador.

6 Conversão analógico-digital e vice-versa

A saída de sensores é geralmente analógica, e isto requer conversão para o formato digital em função do microprocessador ser digital. A saída de um microprocessador é digital, e pode ser necessária uma conversão para analógico para operar um atuador. Muitos microcontroladores têm conversores analógico-digital internos, como, por exemplo, o PIC 16C74/74A (veja a Figura 17.30) e o M68HC11 da Motorola (veja a Figura 17.10), que podem operar com entradas analógicas. Entretanto, quando há a necessidade de saídas analógicas, o microcontrolador geralmente passa esta tarefa para um conversor digital-analógico externo (como exemplo, veja a Seção 20.6.2).

20.3.1 Buffers

Um *buffer* é um dispositivo que é conectado entre duas partes de um sistema para evitar interferência indesejada entre elas. Uma aplicação importante de um *buffer* é na porta de entrada de um microprocessador para isolar a entrada de dados do barramento de dados do microprocessador até que o microprocessador necessite dela. O *buffer* mais usado é o *buffer tristate*. Este *buffer* é habilitado por um sinal de controle para fornecer um nível lógico 0 ou 1 na saída, e quando desabilitado apresenta uma alta impedância, se comportando efetivamente como um circuito aberto. A Figura 20.2 mostra os símbolos para *buffers tristate* e as condições nas quais ele é habilitado. A Figura 20.2(a) e (b) mostram o símbolo para *buffers* que não mudam o nível lógico de entrada e a Figura 20.2(c) e (d) mostra *buffers* inversores.

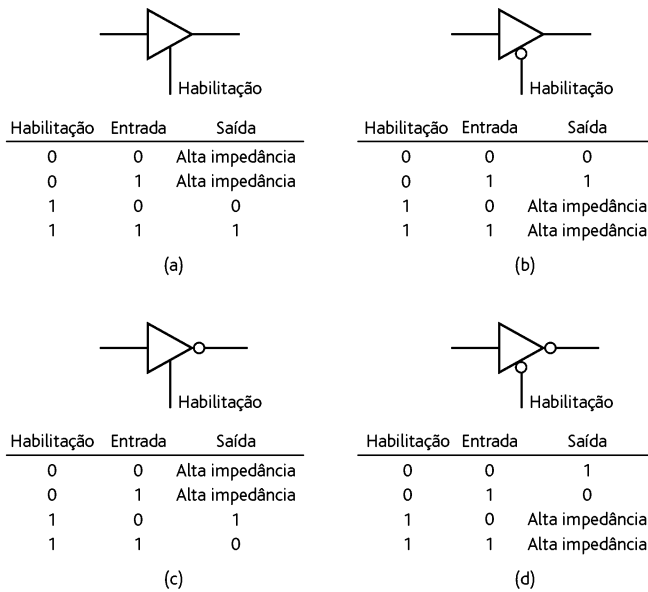


Figura 20.2 Buffers: (a) sem mudança do nível lógico de entrada e habilitado em nível 1, (b) sem mudança do nível lógico de entrada e habilitado em 0, (c) com inversão do nível lógico de entrada e habilitado em 1, (d) com inversão do nível lógico de entrada e habilitado em 0.

Nos microcontroladores PIC (veja a Seção 17.3.3), o bit TRIS é conectado à entrada de habilitação de um *buffer tristate*. Se o bit for 0, o *buffer tristate* é habilitado a passar o valor de entrada para a saída. Se ele for 1, o *buffer tristate* é desabilitado e a saída se torna uma alta impedância (como na Figura 20.2(b)).

Estes *buffers tristate* são usados quando mais de um dispositivo periférico tem que compartilhar as mesmas linhas de dados do microprocessador, ou seja, eles são conectados ao barramento de dados, e desta forma há a necessidade do microprocessador habilitar apenas um dos dispositivos de cada vez, estando os outros desabilitados. A Figura 20.3 mostra como estes buffers podem ser usados. Eles são disponibilizados como circuitos integrados, como, por exemplo, o 74125, que tem quatro *buffers* não inversores ativos em nível baixo, e o 74126, que tem quatro *buffers* não inversores ativos em nível alto.

20.3.2 Handshaking

A menos que dois dispositivos possam enviar e receber dados na mesma taxa, o *handshaking* é necessário para a troca de dados. Com o *handshaking*, o dispositivo mais lento controla a taxa na qual os dados são transferidos. Na transferência paralela de dados, o *handshaking* de **habilitação-reconhecimento** é normalmente usado. O periférico envia um sinal DATA REDY (dados prontos) para a seção de entrada/saída. A CPU identifica que este sinal está ativo e lê em seguida os dados da seção de entrada/saída e envia um sinal INPUT ACKNOWLEDGED (reconhecimento de entrada) para o periférico. Este sinal indica que a transferência foi completada e,

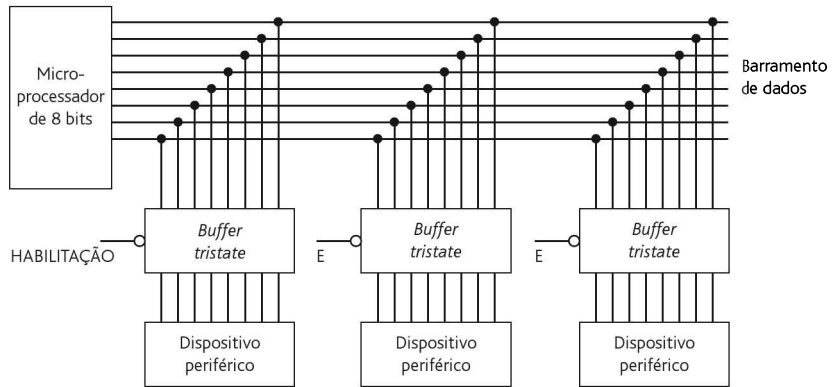


Figura 20.3 Buffer tristate.

portanto, o periférico pode enviar mais dados. Para uma saída, o periférico envia um sinal OUTPUT REQUEST (solicitação de saída) ou PERIPHERAL READY (periférico pronto) para a seção de entrada/saída. A CPU determina que o sinal PERIPHERAL READY é ativado e envia o dado para o periférico. O próximo sinal PERIPHERAL READY pode ser usado para informar à CPU que a transferência foi completada.

No microcontrolador MC68HC11, a operação básica de habilitação de entrada/saída é abordada a seguir. O sinal de controle de *handshaking* usa os pinos STRA e STRB (Figura 20.4(a). Veja também a Figura 17.10 com o diagrama em bloco completo), a porta C é usada para a habilitação da entrada e a porta B, para a saída. Quando o dado está pronto para ser enviado pelo microcontrolador, um pulso é produzido em STRA e enviado para o dispositivo periférico. Quando o microcontrolador recebe uma borda de subida ou descida em STRB, a porta de saída relevante do microcontrolador envia o dado para o periférico. Quando o dado está pronto para ser transmitido para o microcontrolador, o periférico envia um sinal para STRA informando que ele está pronto e, em seguida, envia uma borda de descida ou subida em STRB para indicar que está pronto para receber. Antes que o *handshaking* aconteça, o registrador de entrada/saída PIOC no endereço \$1002 tem

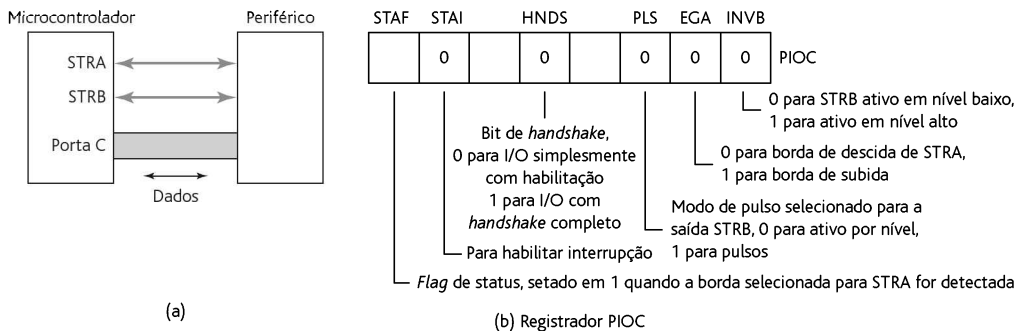


Figura 20.4 Controle de *handshaking*: habilitação e reconhecimento.

que ser configurado primeiro. A Figura 20.4(b) mostra os estados necessários dos bits relevantes no registrador.

O modo de **entrada/saída com handshake completo** envolve dois sinais enviados em STRB, sendo o primeiro para indicar que está pronto para receber dados e o seguinte, que o dado foi lido. Este modo de operação necessita que em PIOC o bit HNDS seja 1, e se PLS for 0, o *handshake* completo é dito pulsado; se for 1, intertravado. Com a operação pulsada, um pulso é enviado como reconhecimento; com intertravamento, STRB é um reset (Figura 20.5).

20.3.3 Consulta e interrupção

Considere a situação em que todas as transferências de entrada/saída de dados são controladas pelo programa. Quando os periféricos precisam de atenção, eles sinalizam para o microprocessador mudando o nível de tensão em uma linha de entrada. O microprocessador pode responder saltando para uma rotina de serviço para o dispositivo. Ao final desta rotina, ocorre um retorno para o programa principal. O controle por programa de entradas/saídas é um *loop* com leitura das entradas e atualização das saídas continuamente, com saltos para rotinas de serviços conforme necessário. Este processo de verificar repetidamente cada dispositivo periférico para saber se ele está pronto para enviar ou receber um novo byte de dados é denominado *consulta (polling)*.

Uma alternativa ao controle por programa é o **controle por interrupção**. Uma interrupção envolve um dispositivo periférico que ativa uma linha de requisição de interrupção. A recepção de uma interrupção resulta no microprocessador suspender a execução do programa principal e saltar para uma rotina de serviço para o periférico. A interrupção não pode levar a perda de dados, e uma rotina de interrupção tem que ser incorporada ao software de modo que o conteúdo dos registradores do processador e o último endereço acessado no programa principal sejam armazenados em um local dedicado na memória. Após a rotina de serviço da interrupção, o conteúdo do local dedicado na memória é recuperado e o microprocessador pode continuar a execução do programa principal a partir do ponto no qual foi interrompido (Figura 20.6).

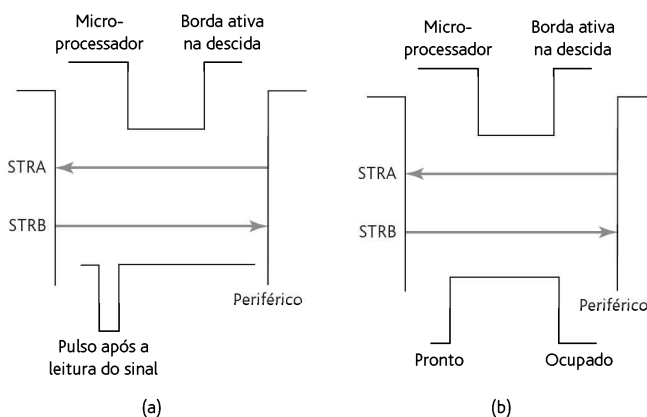


Figura 20.5 Handshaking completo: (a) pulsado, (b) intertravado.

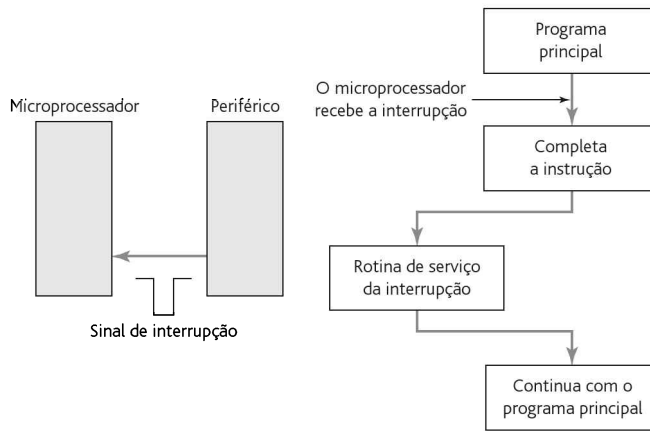


Figura 20.6 Controle de interrupção.

Portanto, quando ocorre uma interrupção:

- 1 A CPU espera até que termine a instrução em curso antes de atender à interrupção.
- 2 Todos os registradores da CPU são armazenados na pilha e um bit setado impede um posterior pedido de interrupção durante o atendimento do pedido anterior. A pilha é uma área especial da memória na qual o valor do contador de programa pode ser armazenado quando uma sub-rotina é executada. O contador de programa fornece o endereço da próxima instrução do programa, sendo que o armazenamento deste valor permite que o programa continue a ser executado após a execução da rotina de interrupção.
- 3 A CPU determina então o endereço da rotina de serviço da interrupção a ser executada. Alguns microprocessadores têm pinos de interrupção dedicados, e o pino escolhido determina qual endereço será usado. Outros microprocessadores têm apenas um pino de interrupção, e o dispositivo que gera a interrupção tem que fornecer um dado ao microprocessador que diz onde se encontra a sub-rotina de serviço da interrupção. Certos microprocessadores têm os dois tipos de entrada de interrupção. O endereço inicial da rotina de serviço da interrupção é denominado **vetor de interrupção**. O bloco de memória associado ao armazenamento de vetores é denominado **tabela de vetores**. Os endereços dos vetores são determinados pelo fabricante do chip.
- 4 A CPU desvia para a rotina de serviço da interrupção.
- 5 Ao final desta rotina, os registradores da CPU retornam da pilha e o programa principal continua a partir do ponto no qual foi interrompido.

Diferentemente de uma chamada de sub-rotina, que é inserida em um ponto específico do programa, uma interrupção pode ser chamada a partir de qualquer ponto do programa. Note que o programa não controla quando uma interrupção ocorre; o controle é do evento de interrupção.

As operações de entrada/saída frequentemente usam interrupções, visto que normalmente o *hardware* não pode esperar. Por exemplo, um teclado pode gerar um sinal de entrada de interrupção quando uma tecla é pressionada. O microprocessa-

dor então suspende o programa principal para tratar do acionamento do teclado; ele processa a informação e então retorna ao programa principal para continuar a partir do ponto em que foi paralisado. Esta habilidade de codificar uma tarefa como uma rotina de serviço de interrupção e associá-la a um sinal externo simplifica muitas tarefas de controle, permitindo que elas sejam tratadas sem atrasos. Para algumas interrupções, é possível programar o microprocessador para ignorar o sinal de interrupção a menos que um bit de habilitação seja setado. Estas interrupções são denominada **mascaráveis**.

O 68HC11 da Motorola tem duas entradas de interrupção externas. XIRQ é uma interrupção não mascarável e é executada sempre ao término da instrução em curso. Quando a interrupção XIRQ ocorre, a CPU salta para a rotina de serviço de interrupção cujo vetor de interrupção está no endereço \$FFF4/5 (os bytes inferior e superior do endereço). IRQ é uma interrupção mascarável. Quando o microcontrolador recebe um sinal no pino de requisição de interrupção IRQ através de uma transição para nível baixo, o microcontrolador salta para a rotina de serviço da interrupção indicada pelo vetor de interrupção \$FFF2/3. IRQ pode ser mascarada (desabilitada) por meio da instrução que seta a máscara de interrupção, SEI, e não mascarada (habilitada) pela instrução que zera a máscara de interrupção, CLI. No final da rotina de serviço da interrupção, a instrução RTI é usada para retornar ao programa principal.

No 8051 da Intel, as fontes de interrupção são habilitadas ou desabilitadas individualmente através do registrador bit-endereçável IE (*interrupt enable*) no endereço 0A8H (veja a Figura 17.26). A interrupção é desabilitada com 0 e habilitada com 1. Além disto, existe um bit de habilitação/desabilitação global no registrador IE que é setado para habilitar todas as interrupções externas ou zerado para desabilitar todas elas. O registrador TCON (Figura 17.25) é usado para determinar o tipo do sinal de entrada de interrupção que iniciará uma interrupção.

Nos microcontroladores PIC, a interrupções são controladas pelo registrador INTCON (Figura 20.7). Para usar o bit 0 da porta B como uma interrupção, ele tem que ser setado como uma entrada e o registrador INTCON tem que ser inicializado com 1 em INTE e 1 em GIE. Se a interrupção é para ocorrer na borda de subida, então INTEDG (bit 6) no registrador OPTION (veja a Figura 17.32) tem que ser 1; se for na borda de descida, ele tem que ser 0. Quando uma interrupção ocorrer, INTF é setado. Ele pode ser zerado através da instrução `bcf INTCON,INTF`.

Como ilustração de um programa que envolve interrupção externa, considere um programa de controle *on/off* simples para um sistema de aquecimento central envolvendo o microcontrolador 8051 da Intel (Figura 20.8). O aquecedor é controlado pela saída P1.7 e são usados dois sensores de temperatura, um para determinar quando a temperatura cai abaixo, por exemplo, de 20,5°C, e o outro



Figura 20.7 INTCON.

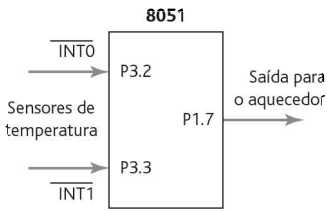


Figura 20.8 Sistema de aquecimento central.

quando ela sobe acima de 21,0°C. O sensor para a temperatura de 21,0°C é conectado na interrupção INT0, porta 3.2, e o sensor para a temperatura 20,5°C é conectado na interrupção INT1, porta 3.3. Selecionando-se o bit IT1 para que seja 1 no registrador TCON, as interrupções externas serão disparadas por borda, ou seja, ativadas quando houver uma transição de 1 para 0, e a interrupção é ativada para executar a instrução CLR P1.7, que zera esta saída, desligando o aquecedor. Quando a temperatura cai para 20,5°C, a interrupção externa INT1 tem uma entrada que varia de 0 para 1 e a interrupção é ativada fazendo executar a instrução

SETB P1.7, fazendo com que esta saída seja 1, ligando o aquecedor. O programa MAIN apenas configura e habilita as interrupções, estabelece a condição inicial de aquecedor ligado se a temperatura for menor do que 21,0°C, ou desligado se abaixo, e esperar sem fazer nada até que uma interrupção ocorra. Neste programa, foi considerado um arquivo de cabeçalho:

```

                ORG    0
                LJMP   MAIN

                ORG    0003H           ; fornece o endereço de entrada para ISR0
ISR0            CLR    P1.7           ; rotina de serviço da interrupção para
                RETI                ; desligar o aquecedor
                RETI                ; retorno da interrupção

                ORG    0013H           ; fornece o endereço de entrada para ISR1
ISR1            SETB   P1.7           ; rotina de serviço da interrupção para ligar o
                RETI                ; aquecedor
                RETI                ; retorno da interrupção

                ORG    30H
MAIN            SETB   EX0            ; para habilitar a interrupção externa 0
                SETB   EX1            ; para habilitar a interrupção externa 1
                SETB   ITO            ; seta o disparo quando muda de 1 para 0
                SETB   IT1           ; set to trigger when change from to 0
                SETB   P1.7           ; liga o aquecedor
                JB     P3.2,HERE       ; se a temperatura for maior do que 21,0°C
                SJMP  HERE            ; salte para HERE e deixe o aquecedor ligado
                CLR    P1.7           ; desliga o aquecedor
HERE            SJMP  HERE            ; apenas não faz nada até que ocorra uma
                RETI                ; interrupção

                END
    
```

Além da solicitação de interrupção, os microcontroladores têm uma interrupção de reset e uma interrupção, não mascarável. A interrupção de reset é um tipo especial de interrupção e quando ela ocorre, o sistema reseta; portanto, quando ela é ativada, toda a atividade no sistema para, o endereço inicial do programa principal é carregado e a rotina de *start-up* (inicialização) é executada. O microcontrolador M68HC11 tem um temporizador *matchdog* que sinaliza a operação adequada do computador (COP). A intenção é detectar erros no processamento do software quando a CPU não estiver executando certas seções do código dentro de um tempo

especificado. Quando isto ocorre, o temporizador COP finaliza a temporização, iniciando um reset no sistema.

A interrupção não mascarável não pode ser mascarada e, assim, não há nenhum método para evitar que a rotina de serviço desta interrupção seja executada quando esta linha for ativada. Uma interrupção deste tipo é geralmente reservada para rotinas de emergência como a que é necessária quando há uma falha de alimentação, como, por exemplo, a comutação para uma fonte *back-up* (sobressalente) de alimentação.

20.3.4 Interface serial

Na transmissão de dados em paralelo, uma linha é usada para cada bit; entretanto, os sistemas seriais usam um única linha para transmitir dados em bits sequenciais. Existem dois tipos básicos de transferência de dados seriais: assíncrona e síncrona.

Na **transmissão assíncrona**, o receptor e o transmissor usam sinais de clock próprios, de modo que não é possível para o receptor saber quando a palavra inicia ou termina. Portanto, é necessário que cada palavra de dado transmitida tenha bits de início e término próprios, de modo que seja possível para o receptor dizer onde uma palavra termina e a outra começa (Figura 20.9). Com este modo de transmissão, o transmissor e o receptor são tipicamente remotos (veja o Capítulo 22 para detalhes de interfaces padrão). Na transmissão síncrona, o transmissor e o receptor têm um sinal de clock comum e, portanto, a transmissão e a recepção podem ser sincronizadas.

O microcontrolador MC68HC11 (veja a Figura 17.10) tem uma interface de comunicação serial (SCI) que pode ser usada para transmissão assíncrona e, assim, pode ser usada para comunicação com dispositivos periféricos remotos. A SCI usa o pino PD1 da porta D como uma linha de transmissão e PD0 como linha de recepção. Estas linhas podem ser habilitadas ou desabilitadas através do registrador de controle SCI. O microcontrolador tem também uma interface periférica serial (SPI) para transmissão síncrona. Esta interface pode ser usada para comunicação serial local. A terminologia local significa essencialmente dentro da máquina na qual o chip está localizado.

20.4 Adaptadores de interface para periférico

As interfaces podem ser especificamente projetadas para entradas/saídas particulares; entretanto, existem disponíveis dispositivos de interface de entrada/saída programáveis que permitem diferentes opções de entrada e saída selecionadas por meio de software. Estes dispositivos são conhecidos como **adaptadores de interfaces periféricas (PIAs)**.

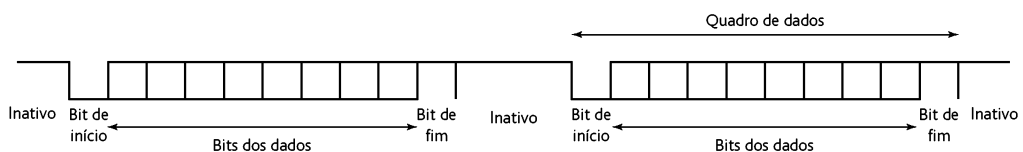


Figura 20.9 Transmissão assíncrona.

Uma interface paralela PIA normalmente usada é a MC6821 da Motorola. Ela faz parte da família MC6800 e pode ser conectada diretamente aos barramentos do MC6800 e do MC68HC11. O dispositivo pode ser considerado essencialmente como duas portas de entrada/saída paralelas, com lógica de controle, para conexão com o microprocessador mestre. A Figura 20.10 mostra a estrutura básica do PIA MC6821 e os seus pinos de conexão.

O PIA contém duas portas paralelas de 8 bits, denominadas A e B. Cada porta tem:

- 1 *Um registrador de interface periférica.* Uma porta de saída opera de forma diferente de uma porta de entrada porque os dados têm que ser armazenados para o periférico. Portanto, é usado um registrador para a saída armazenar temporariamente os dados. O registrador apresenta a função *latch* apenas quando a porta é configurada como saída.
- 2 *Um registrador de direção de dados* que determina se as linhas de entrada/saída são entradas ou saídas.
- 3 *Um registrador de controle* que determina as conexões lógicas ativas no periférico.
- 4 *Dois linhas de controle, CA1 e CA2 ou CB1 e CB2.*

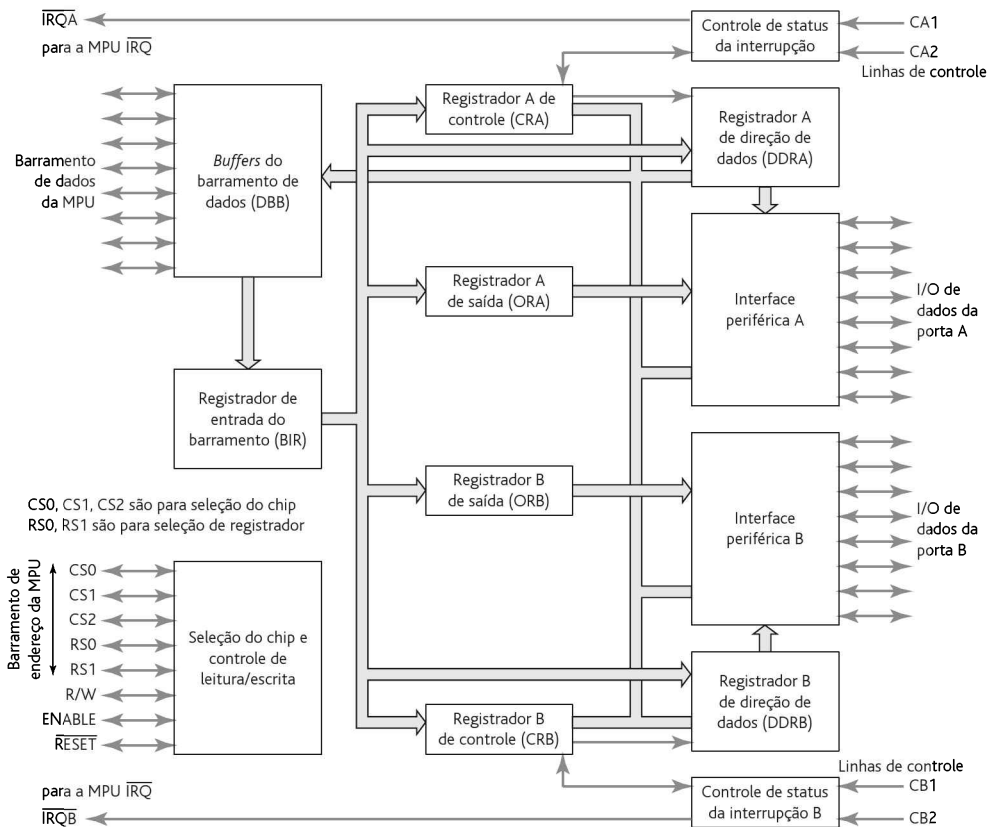


Figura 20.10 PIA MC6821.

Duas linhas de endereço do microprocessador são conectadas diretamente à PIA através das duas linhas de seleção de registrador, RS0 e RS1. Isto produz quatro endereços para os seis registradores da PIA. Quando RS1 está em nível baixo, o lado A é endereçado, e quando ele está em nível alto, é o lado B. RS0 endereça os registros de um determinado lado, ou seja, A ou B. Quando RS0 está em nível alto, o registrador de controle é endereçado; quando está em nível baixo, é o registrador de dados ou o registrador de direção dos dados. Para um determinado lado, o registrador de dados e o registrador de direção dos dados têm o mesmo endereço. Qual deles é endereçado é determinado pelo bit 2 do registrador de controle (veja a seguir).

Cada um dos bits nos registradores de controle A e B está relacionado com algumas características da operação das portas. Portanto, para o registrador de controle A, temos os bits mostrados na Figura 20.11. Um padrão similar é usado para o registrador de controle B.

Bits 0 e 1

Os dois primeiros bits controlam a forma em que as linhas de controle de entrada CA1 ou CB1 operam. O bit 0 determina se a saída de interrupção é habilitada. B0 = 0 desabilita a interrupção IRQA(B) do microprocessador, B0 = 1 habilita a interrupção. CA1 e CB1 não são ativadas por nível, mas por borda, ou seja, por uma variação no sinal. O bit 1 determina se o bit 7 é setado por uma transição de alto para baixo (borda posterior) ou uma transição de baixo para alto (borda anterior). B1 = 0 seta a transição de alto para baixo, B1 = 1 seta a transição de baixo para alto.

Bit 2

O bit 2 determina se os registradores de direção dos dados ou os registradores de dados do periférico são endereçados. Com B2 em 0, os registradores de direção dos dados são endereçados; com B2 em 1, os registradores de dados do periférico são selecionados.

Bits 3, 4 e 5

Estes bits permitem que a PIA realize uma variedade de funções. O bit 5 determina se a linha de controle 2 é uma entrada ou uma saída. Se o bit 5 for 0, a linha de controle 2 é uma entrada; se for 1, será uma saída. No modo de entrada, CA2 e CB2 operam da mesma forma. Os bits 3 e 4 determinam se a saída de interrupção é ativa conforme o tipo de transição definida no bit 6.

Com B5 = 0, ou seja, CA2(CB2) setado como entrada: B3 = 0 desabilita a interrupção IRQA(B) do microprocessador através de CA2(CB2); B3 = 1 habilita a interrupção IRQA(B) do microprocessador através de CA2(CB2); B4 = 0 determina que o *flag* de interrupção IRQA(B), bit 6, é setado por uma transição de alto para baixo em CA2(CB2); B4 = 1 determina que ele é setado por uma transição de baixo para alto.

B5 = 1 seta CA2(CB2) como saída. No modo saída, CA2 e CB2 se comportam diferentemente. Para CA2: com B4 = 0 e B3 = 0, CA2 vai para nível

B7	B6	B5	B4	B3	B2	B1	B0
IRQA1	IRQA2	Controle CA2		Acesso DDRA		Controle CA1	

Figura 20.11 Registrador de controle.

baixo na primeira transição de alto para baixo em ENABLE (E) após uma leitura do microprocessador no registrador de dados A do periférico e retorna para nível alto na próxima transição de CA1; com $B4 = 0$ e $B3 = 1$, CA2 vai para nível baixo na primeira transição de alto para baixo em ENABLE após uma leitura do microprocessador no registrador de dados A do periférico e retorna para nível alto na próxima transição de alto para baixo em ENABLE. Para CB2: com $B4 = 0$ e $B3 = 0$, CB2 vai para nível baixo na primeira transição de baixo para alto em ENABLE após uma escrita do microprocessador no registrador de dados B e retorna para nível alto na próxima transição em CB1; com $B4 = 0$ e $B3 = 1$, CB2 vai para nível baixo na primeira transição de baixo para alto em ENABLE após uma escrita do microprocessador no registrador de dados B e retorna para nível alto na próxima transição de baixo para alto em ENABLE. Com $B4 = 1$ e $B3 = 0$, CA2(CB2) vai para nível baixo enquanto o microprocessador escreve $B3 = 0$ no registrador de controle. Com $B4 = 0$ e $B3 = 1$, CA2(CB2) vai para nível alto enquanto o microprocessador escreve $B3 = 1$ no registrador de controle.

Bit 6

Este é o *flag* de interrupção de CA1(CB1), que é setado pela transição em CA2(CB2). Com CA2(CB2) como entrada ($B5 = 0$), ele é zerado por uma leitura do microprocessador no registrador de dados A(B). Com CA2(CB2) como saída ($B5 = 1$), o *flag* é 0 e não é afetado pelas transições em CA2(CB2).

Bit 7

Este é o *flag* da interrupção de CA1(CB1), sendo zerado por uma leitura do microprocessador no registrador de dados A(B).

O processo de seleção das opções a serem usadas é denominado **configuração** ou **inicialização** da PIA. A conexão de RESET é usada para zerar todos os registros da PIA. Em seguida, a PIA tem que ser inicializada.

20.4.1 Inicialização da PIA

Antes da PIA ser usada, deve ser executado um programa em que as condições desejadas para o fluxo de dados no periférico sejam definidas. O programa da PIA é colocado no início do programa principal de modo que, deste ponto em diante, o microprocessador pode ler dados no periférico. O programa de inicialização é executado apenas uma vez.

Este programa que define as portas como entrada ou saída pode ter os seguintes passos:

- 1 Zerar o bit 2 de cada registrador de controle através de um reset, assim os registradores de direção de dados são endereçados. O registrador de direção de dados A é endereçado como XXX0 e o registrador de direção de dados B, como XXX2.
- 2 Para A ser uma porta de entrada, carregue zeros no registrador de direção A.
- 3 Para B ser uma porta de saída, carregue uns no registrador de direção B.
- 4 Carregue 1 no bit 2 dos dois registradores de controle. O registrador A de dados agora é endereçado como XXX0 e o B, como XXX2.

Portanto, um exemplo de programa de inicialização em assembly que configura o lado A como entrada e o B como saída, logo após um reset, é mostrado a seguir:

```

INIT   LDAA   #$00   ; Carrega zeros
        STAA  $2000  ; Configura o lado A como porta de entrada
        LDAA  #$FF   ; Carrega uns
        STAA  $2000  ; Configura o lado B como porta de saída
        LDAA  #$04   ; Carrega 1 no bit 2, ficando os outros bits em 0
        STAA  $2000  ; Seleciona o registrador de dados da porta A
        STAA  $2002  ; Seleciona o registrador de dados da porta B
    
```

Agora os dados do periférico podem ser lidos a partir da porta A de entrada com a instrução LDAA 2000 e o microprocessador pode escrever dados no periférico a partir da porta B de saída com a instrução STAA 2002.

20.4.2 Conexão dos sinais de interrupção via PIA

A PIA MC6821 da Motorola (Figura 20.12) tem as conexões IRQA e IRQB através das quais os sinais de interrupção podem ser enviados ao microprocessador de modo que as requisições de interrupção a partir de CA1 e CA2 ou CB1 e CB2 podem acionar o pino IRQ do microprocessador para ativar o estado baixo. Quando o programa de inicialização para a PIA foi considerado na seção anterior, apenas o bit 2 do registrador de controle foi carregado com 1, e os outros bits, com 0. Estes zeros desabilitam as entradas de interrupção. Para usar as interrupções, o passo da inicialização que armazena \$04 no registrador de controle tem que ser modificado.

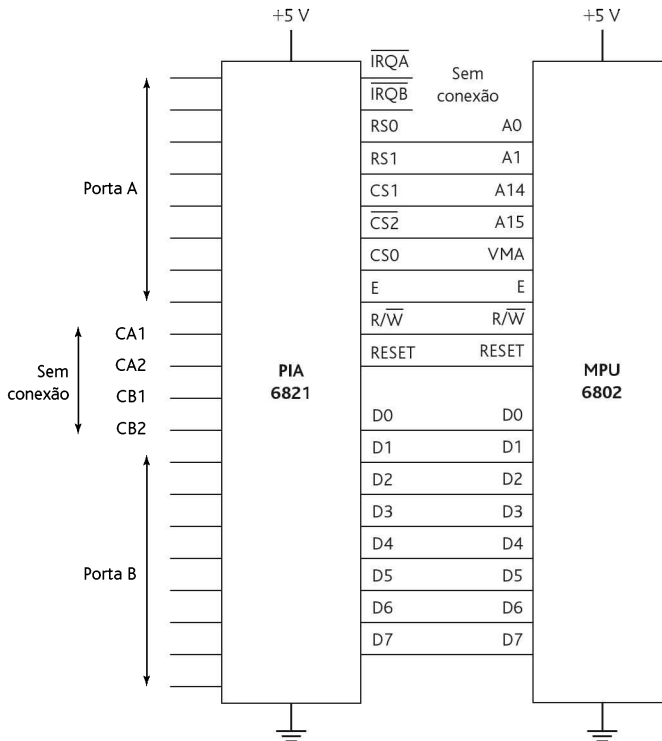


Figura 20.12 Interfaceamento com a PIA.

A forma da modificação dependerá do tipo de variação na entrada que ativar a interrupção.

Suponha, por exemplo, que queremos que CA1 habilite uma interrupção quando houver uma transição de nível alto para baixo, com CA2 e CBI não utilizadas e CB2 habilitada e usada para uma saída de *set/reset*. O formato do registrador de controle para esta especificação é, para CA:

B0 é 1 para habilitar a interrupção em CA1.

B1 é 0 de modo que o *flag* de interrupção IRQA1 é setado através de uma transição de nível alto para baixo em CA1.

B2 é 1 para que o registrador de dados possa ser acessado.

B3, B4 e B5 são 0 porque CA2 está desabilitada.

B6 e B7 são *flags* apenas de leitura e, assim, pode ser usado 0 ou 1.

Portanto, o formato para CA1 pode ser 00000101, que é 05 em hexadecimal. O formato do registrador de controle para CB2 é:

B0 é 0 para desabilitar CBI.

B1 pode ser 0 ou 1 visto que CBI está desabilitada.

B2 é 1 para que o registrador de dados possa ser acessado.

B3 é 0, B4 é 1 e B5 é 1, para selecionar o *set/reset*.

B6 e B7 são *flags* apenas de leitura e, assim, pode ser usado 0 ou 1.

Portanto, o formato para CA1 pode ser 00110100, que é 34 em hexadecimal. O programa de inicialização pode então ser:

INIT	LDAA	#\$00	; Carrega zeros
	STAA	\$2000	; Configura o lado A como porta de entrada
	LDAA	#\$FF	; Carrega uns
	STAA	\$2000	; Configura o lado B como porta de saída
	LDAA	#\$05	; Carrega o formato do registrador de controle desejado
	STAA	\$2000	; Seleciona o registrador de dados da porta A
	LDAA	#\$34	; Carrega o formato do registrador de controle desejado
	STAA	\$2002	; Seleciona o registrador de dados da porta B

20.4.3 Um exemplo de interfaceamento com uma PIA

Como exemplo de interfaceamento com uma PIA, a Figura 20.13 mostra um circuito que pode ser usado com um motor de passo unipolar (veja a Seção 9.7.2). Os enrolamentos indutivos podem gerar uma grande FEM reversa quando comutados, de modo que deve haver alguma forma de isolamento entre as bobinas e a PIA. Podem ser usados optoisoladores, diodos ou resistores. Os diodos constituem uma interface simples e barata, e os resistores não isolam completamente a PIA.

20.5 Interface de comunicação serial

O **transmissor/receptor assíncrono universal** (UART – *Universal Asynchronous Receiver/Transmitter*) é o elemento essencial de um sistema de comunicação serial com a função de transformar dados em formato serial para paralelo para a entra-

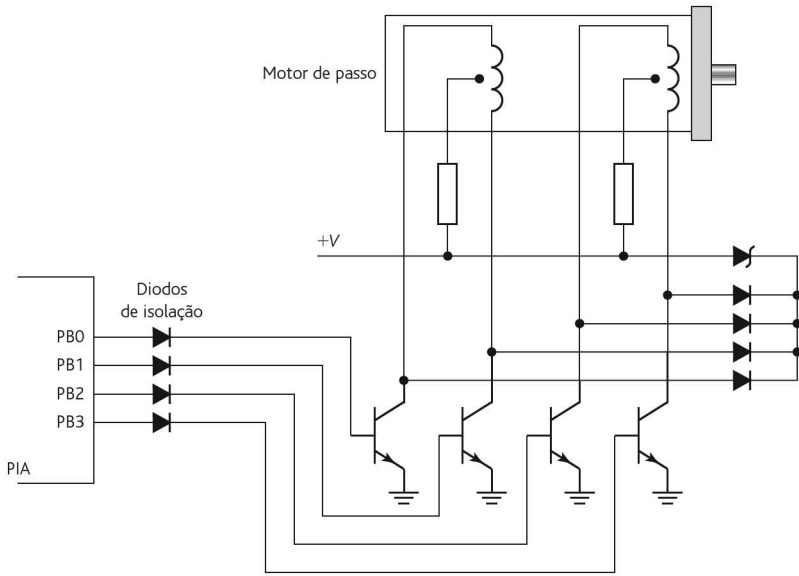


Figura 20.13 Interfacemento com um motor de passo.

da, e dados do formato paralelo para serial para a saída. Um tipo de UART comum programável é o **adaptador de interface de comunicação assíncrona (ACIA)** da Motorola (MC6850); a Figura 20.14 mostra um diagrama em bloco interno dele.

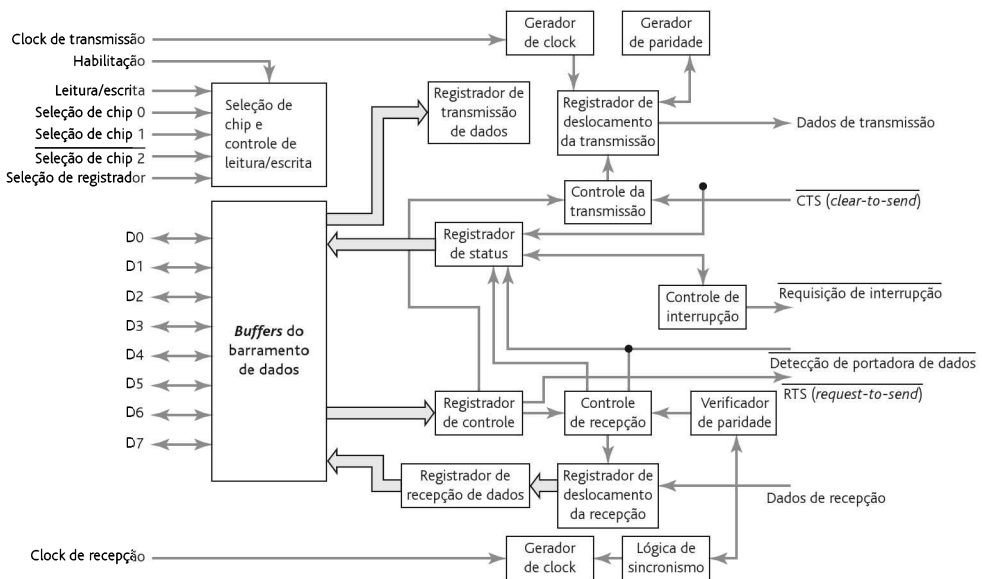


Figura 20.14 ACIA MC6850.

Os dados fluem entre o microprocessador e a ACIA via 8 linhas bidirecionais, D0 a D7. A direção dos dados é controlada pelo microprocessador através da entrada de leitura/escrita da ACIA. As três linhas de seleção de chip são usadas para endereçamento de uma determinada ACIA. A linha de seleção de registrador é usada para selecionar determinados registradores dentro da ACIA; se a linha de seleção de registro for nível alto, os registros transmissor e receptor de dados são selecionados. O registrador de status contém informações sobre o status da transferência serial de dados conforme elas ocorrem, e é usado para ler as linhas de detecção de portadora de dados e de CTS (*clear-to-send*). O registrador de controle é usado inicialmente para resetar a ACIA e subsequentemente definir a taxa de transferência serial de dados e o formato do dado.

O lado da ACIA conectado ao periférico inclui duas linhas de dados seriais e três linhas de controle. Os dados são enviados através da linha de dados do transmissor e recebidos pela linha de dados do receptor. Os sinais de controle são fornecidos pelo CTS, pelo detector de portadora de dados e pelo RTS. A Figura 20.15 mostra o formato dos bits de controle, e a Figura 20.16 mostra o registrador de status.

A transferência assíncrona serial de dados é geralmente usada para comunicações entre dois computadores, com ou sem modem, ou entre um computador e uma impressora (veja o Capítulo 22 para mais detalhes).

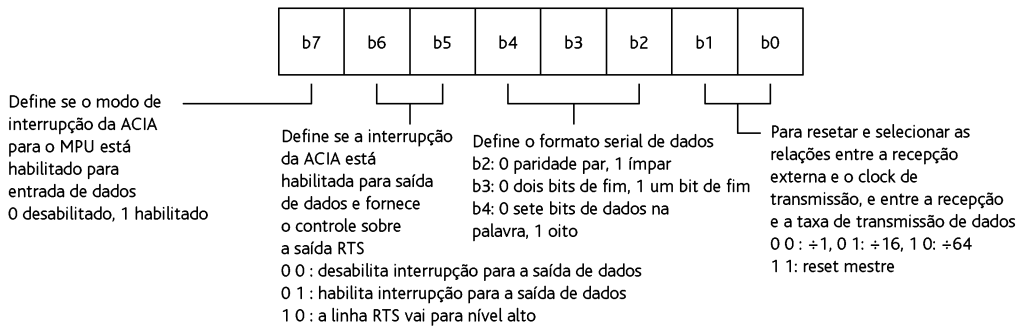


Figura 20.15 Registrador de controle.

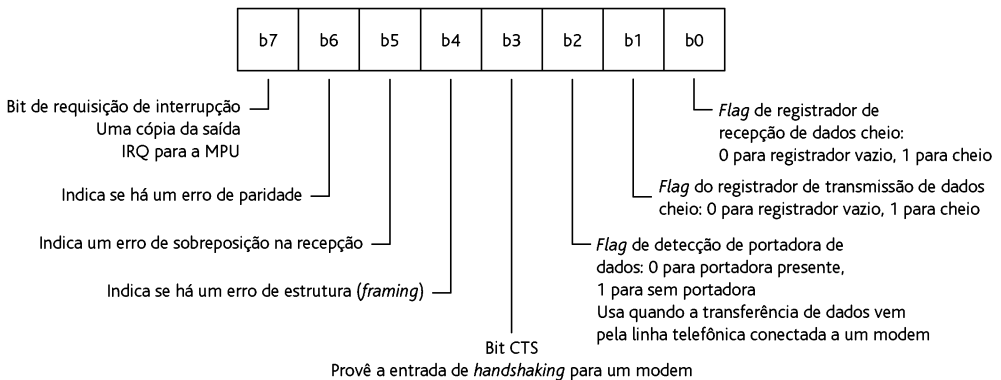


Figura 20.16 Registrador de status.

20.5.1 Interfaces seriais de microcontroladores

Muitos microcontroladores têm interfaces seriais, ou seja, UARTs internas. Por exemplo, o M68HC11 da Motorola tem uma interface de periférico serial (SPI), que é uma interface assíncrona, e uma interface de comunicação serial (SCI), que é uma interface assíncrona (veja a Figura 17.10). O SPI necessita do mesmo sinal de clock usado pelo microcontrolador e os dispositivos conectados externamente (Figura 20.17(a)). O SPI permite, com esta facilidade, a interconexão de mais de um microcontrolador. O SCI é uma interface assíncrona que permite que sinais de clock diferentes sejam usados por um sistema SCI e o dispositivo conectado externamente (Figura 20.17(b)). Os microprocessadores de propósito geral não têm uma SCI ou uma UART, como, por exemplo, o MC6850 da Motorola, que tem que ser usado para permitir uma comunicação serial. Em algumas situações, é necessário usar mais de um SCI e, assim, o microcontrolador M68HC11 precisará de mais uma UART.

A SPI é inicializada através de bits no registrador de controle do SPI (SPCR) e no registrador de controle de direção da porta D (DDRD). O registrador de status da SPI contém os bits de status e erro. A SCI é inicializada através dos registradores 1 e 2 de controle da SCI e do registrador de controle da taxa *baud*. Os *flags* de status estão no registrador de status da SCI.

O 8051 da Intel tem uma interface serial interna com quatro modos de operação, os quais são selecionados ao se escrever uns ou zeros nos bits SM0 e SM1 no registrador SCON (controle da porta serial) no endereço 98H (Figura 20.18) (Tabela 20.1).

No modo 0, os dados seriais entram e saem através de RXD. O pino TXD fornece o clock de deslocamento que é usado para sincronizar a transmissão e a recepção de dados. A recepção é iniciada quando REN for 1 e R1 for 0. A transmissão

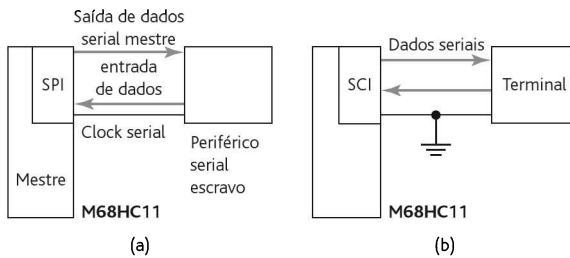


Figura 20.17 (a) SPI, (b) SCI.

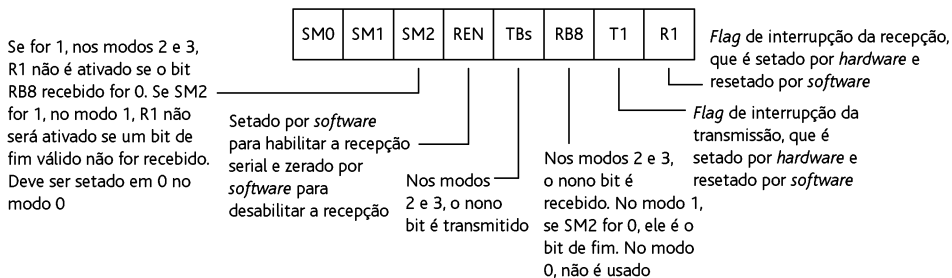


Figura 20.18 Registrador SCON.

Tabela 20.1 Modos da porta serial do 8051 da Intel

SMO	SM1	Modo	Descrição	Taxa em <i>baud</i>
0	0	0	Registrador de deslocamento	Freq. osc. /12
0	1	1	UART de 8 bits	Variável
1	0	2	UART de 9 bits	Freq. osc. /12 ou 64
1	1	3	UART de 9 bits	Variável

é iniciada quando qualquer dado for escrito no SBUF, que é o *buffer* da porta serial no endereço 99H. No modo 1, 10 bits são transmitidos em TXD ou recebidos em RXD; estes são o bit de início, que é 0, oito bits de dados e um bit de fim, que é 1. A transmissão é iniciada através de uma escrita no SBUF, e a recepção é iniciada através de uma transição de 1 para 0 em RXD. Nos modos 2 e 3, 11 bits são transmitidos em TXD ou recebidos em RXD.

Os microcontroladores PIC têm uma SPI (veja a Figura 17.30) que pode ser usada para comunicação serial síncrona. Quando os dados são escritos no registrador SSBUF, eles são deslocados para fora por meio do pino SDO em sincronismo com o sinal de clock em SCK e também por meio do pino RC5 como um sinal serial com o bit mais significativo aparecendo primeiro e o sinal de clock através de RC3. A entrada para o registrador SSBUF é via RC4. Muitos microcontroladores PIC também têm uma UART para criar uma interface serial para ser usada na transferência de dados seriais de forma assíncrona. Na transmissão, cada byte é delimitado por um bit de início (START) e um bit de fim (STOP). Quando o bit START é transmitido, a linha RX cai para nível baixo e o receptor pode então sincronizar com esta transição de alto para baixo. Em seguida, o receptor lê os 8 bits de dados serial.

20.6 Exemplos de interfaceamento

Os itens a seguir são exemplos de interfaceamento.

20.6.1 Interfaceamento com um display de sete segmentos com um decodificador

Considere que a saída de um microcontrolador seja usada para acionar um display de LEDs de sete segmentos (veja a Seção 6.5). Um único LED é um indicador *on/off*, no caso de um display, o número indicado depende de quais LEDs estão ligados. A Figura 20.19 mostra como podemos usar um microcontrolador para acionar um display do tipo anodo comum usando um decodificador acionador, que é capaz de receber uma entrada BCD e convertê-la no código apropriado para o display.

Para o decodificador 7447, os pinos 7, 1, 2 e 6 são as entradas BCD e os pinos 13, 12, 11, 10, 9, 15 e 14 são as saídas para os segmentos do display. O pino 9 do display é o ponto decimal. A Tabela 20.2 mostra os sinais de entrada e saída para o decodificador.

O **apagamento** ocorre quando nenhum dos segmentos é ativado. Esta função é usada para evitar que apareça um zero no dígito mais significativo quando temos, por exemplo, três displays e queremos mostrar apenas 10 em vez de 010, e assim

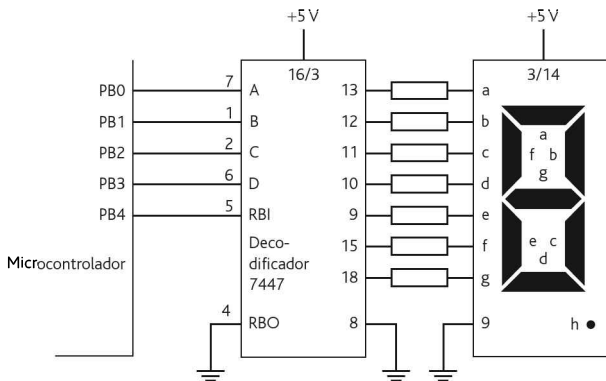


Figura 20.19 Acionamento de um display.

apagamos o 0 mais significativo para evitar que ele apareça. Consegue-se isto através da entrada de apagamento (RBI – *Ripple Blanking Input*) ativa em nível baixo. Quando RBI estiver em nível baixo e as entradas BCD estiverem todas em nível baixo, a saída é apagada. Se a entrada não for zero, a saída de apagamento (RBO – *Ripple Blanking Output*) será nível alto independentemente do status de RBI. A saída RBO do primeiro dígito pode ser conectada na entrada RBI do segundo, e o RBO deste no RBI do terceiro, permitindo assim que os zeros à esquerda do dígito mais significativo diferente de 0 sejam apagados (Figura 20.20).

Em displays com muitos dígitos, em vez de usar um decodificador para cada dígito, é usada a multiplexação com um único decodificador. A Figura 20.21 mostra o circuito para a multiplexação de um display do tipo catodo comum de 4 dígitos. O dado BCD sai pela porta A para o decodificador. A saída deste vai para todos os displays. Cada display tem seu catodo comum conectado a um transistor para GND. O display é ligado quando o transistor é ligado através da saída da porta B. Portanto, comutando entre PB0, PB1, PB2 e PB3, a saída da porta A pode ser comutada para o display apropriado. Para manter uma imagem constante no display, este é repeti-

Tabela 20.2 O decodificador BCD 7447 para um display de sete segmentos

Display	Pinos de entrada				Pinos de saída						
	6	2	1	7	13	12	11	10	9	15	14
0	L	L	L	L	ON	ON	ON	ON	ON	ON	OFF
1	L	L	L	H	OFF	ON	ON	OFF	OFF	OFF	OFF
2	L	L	H	L	ON	ON	OFF	ON	ON	OFF	ON
3	L	L	H	H	ON	ON	ON	ON	OFF	OFF	ON
4	L	H	L	L	OFF	ON	ON	OFF	OFF	ON	ON
5	L	H	H	L	ON	OFF	ON	ON	OFF	ON	ON
6	L	H	H	L	OFF	OFF	ON	ON	ON	ON	ON
7	L	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF
8	H	L	H	H	ON	ON	ON	OFF	OFF	OFF	OFF
9	H	L	H	L	ON	ON	ON	OFF	OFF	OFF	OFF

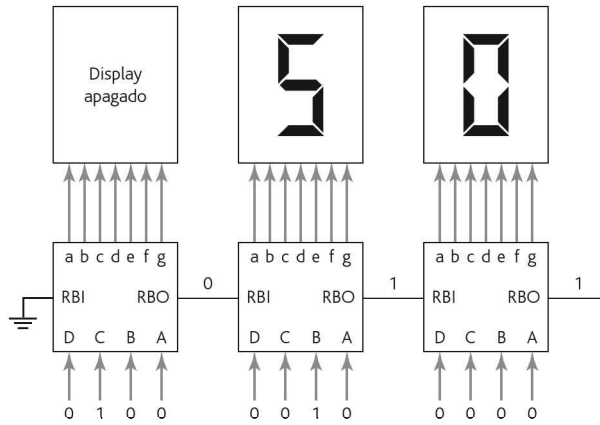


Figura 20.20 Apagamento do zero mais significativo.

damente ligado suficientemente rápido para que não apareçam cintilações (*flickers*). A multiplexação por divisão do tempo pode ser usada para permitir que mais de um dígito seja mostrado de cada vez.

20.6.2 Interface analógica

A conversão digital-analógica é necessária quando é necessário que a saída de um microprocessador ou microcontrolador forneça um sinal analógico. Por exemplo, o DAC

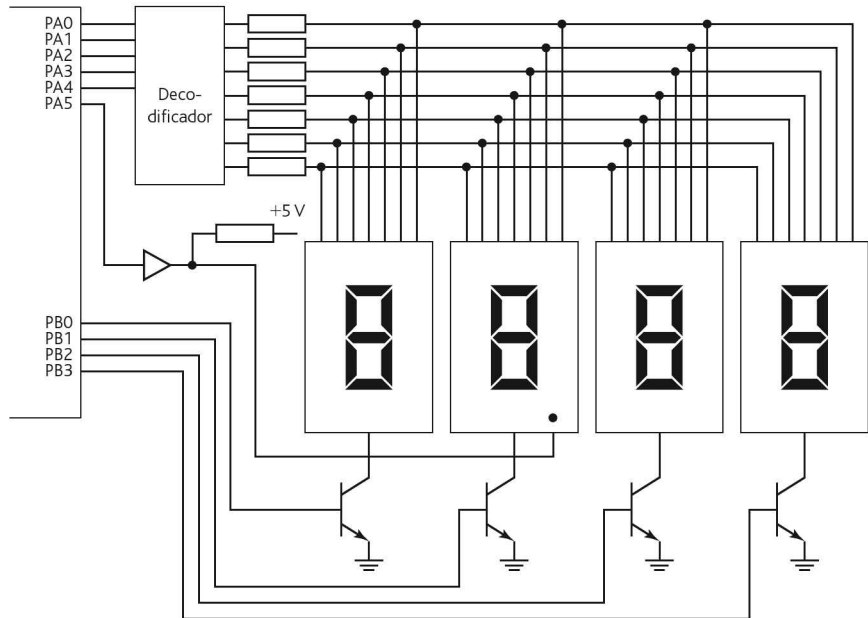


Figura 20.21 Multiplexação de 4 displays.

AD557 da Analog Devices pode ser usado para esta finalidade. Ele produz uma tensão de saída proporcional à sua entrada digital, contendo *latches* de entrada no interfaceamento com o microprocessador. Se os *latches* não forem necessários, os pinos 9 e 10 são conectados em GND. Os dados são armazenados nos *latches* quando há uma borda positiva, ou seja, uma transição de nível baixo para alto, na entrada do pino 9 ou do pino 10. O dado é mantido até que estes dois pinos retornem para nível baixo. Quando isto acontece, o dado é transferido do *latch* do DAC para a conversão em tensão analógica.

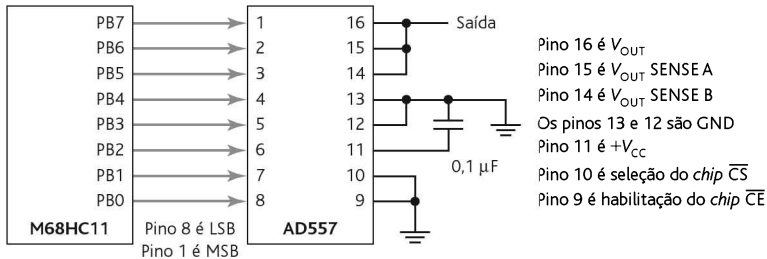


Figura 20.22 Geração de forma de onda.

A Figura 20.22 mostra o AD557 sem o uso do *latch* e conectado ao M68HC11 da Motorola de modo que, quando o programa a seguir é executado, ele gera uma tensão de saída na forma de dente de serra. Outras formas de onda de tensão podem ser geradas alterando-se o programa:

```

BASE EQU $1000      ; Endereço base de registradores I/O
PORTB EQU $04       ; Offset de PORTB a partir de BASE

                ORG $C000
                LDX #BASE      ; Aponta X para o registrador base
                CLR PORTB,X     ; Envia 0 para o DAC
AGAIN          INC PORTB,X     ; Incrementa de 1
                BRA AGAIN      ; Repete
                END
  
```

Resumo

Os **requisitos de interface** geralmente são *isolamento/buffer* elétrico, controle de temporização, conversão de código, mudança no número de linhas, conversão de serial para paralelo e vice-versa, conversão analógico-digital e vice-versa. A menos que dois dispositivos possam enviar e receber dados em taxas idênticas, o *handshaking* é necessário.

A **consulta** é uma técnica de programação de controle de entradas/saídas na qual um *loop* é usado continuamente para ler entradas e atualizar saídas, com saltos para rotinas de serviços conforme necessário, ou seja, um processo de repetidas verificações em cada dispositivo periférico para saber se ele está pronto para enviar ou receber um novo byte de dados. Um programa de controle alternativo é o **controle por interrupção**. Uma interrupção envolve um dispositivo periférico que

ativa uma linha de requisição de interrupção. A recepção de uma interrupção faz com que o microprocessador suspenda a execução do programa principal e salte para a rotina de serviço para o periférico. Após a rotina do serviço da interrupção, os conteúdos da memória são recuperados e o microprocessador pode continuar a execução do programa principal a partir do ponto em que foi interrompido.

Existem dois tipos básicos de transferência serial de dados: assíncrona e síncrona. Na **transmissão assíncrona**, o receptor e o transmissor usam seus próprios sinais de clock de modo que não é possível para o receptor saber quando uma palavra começa ou termina. Portanto, é necessário para cada palavra de dado transmitida a inserção de bits de início e de fim de modo que seja possível para o receptor saber quando uma palavra termina e a outra começa. Na **transmissão síncrona**, o transmissor e o receptor têm um sinal de clock comum e, assim, a transmissão e a recepção podem ser sincronizadas.

Os **adaptadores de interface de periféricos (PIAs)** são dispositivos de interface programáveis que permitem que várias opções de entrada e saída diferentes sejam selecionadas por meio do software.

O **transmissor/receptor assíncrono universal (UART)** é o dispositivo essencial de um sistema de comunicação serial que tem a função de converter dados de série para paralelo na entrada e de paralelo para série na saída. Um tipo comum de UART é o **adaptador de interface de comunicação assíncrona (ACIA)**.

Problemas

- 20.1 Descreva as funções que podem ser necessárias em uma interface.
- 20.2 Explique a diferença entre uma interface paralela e uma serial.
- 20.3 Explique o significado de sistema de I/O mapeado na memória.
- 20.4 Qual é a função de um adaptador de interface de periférico?
- 20.5 Descreva a arquitetura da PIA MC6821 da Motorola.
- 20.6 Explique a função de um programa de inicialização de uma PIA.
- 20.7 Quais são as vantagens do uso de interrupções externas em vez da consulta por software como meio de comunicação com periféricos?
- 20.8 Para a PIA MC6821 da Motorola, qual valor deve ser armazenado no registrador de controle se CA1 for desabilitada, CB1 for a entrada de interrupção habilitada e setada para uma transição de baixo para alto, CA2 é habilitada e usada com saída *set/reset*, e CB2 é habilitada e vai para nível baixo na primeira transição de baixo para alto em E após uma escrita do microprocessador no registrador B de dados do periférico e retorna para nível alto na próxima transição de baixo para alto em E?
- 20.9 Escreva, em linguagem assembly, um programa para inicializar a PIA MC6821 da Motorola para obter a especificação dada no problema 20.8.
- 20.10 Escreva, em linguagem assembly, um programa para inicializar a PIA MC6821 da Motorola para ler os 8 bits de dados da porta A.

21

Controladores Lógicos Programáveis

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Descrever a estrutura básica de CLPs e suas operações.
- Desenvolver programas em *ladder* para um CLP envolvendo funções lógicas, *latch*, relés internos e sequenciamento.
- Desenvolver programas envolvendo temporizadores, contadores, registradores de deslocamento, relés mestre, saltos e manipulação de dados.

21.1 Controlador lógico programável

Um **controlador lógico programável** (CLP) é um equipamento eletrônico que usa uma memória programável para armazenar instruções e implementar funções como lógica, sequenciamento, temporização, contagem e aritméticas para o controle de máquinas e processos, e foi especialmente projetado para tornar fácil a programação. O termo lógica é usada porque a programação é principalmente voltada para implementações lógicas e operações de comutação. Os dispositivos de entrada, como, por exemplo, chaves, e dispositivos de saída, como, por exemplo, motores, que são controlados são conectados no CLP, e o controlador monitora as entradas e saídas de acordo com o programa armazenado no CLP pelo operador para controlar as máquinas ou processos. Originalmente, os CLPs foram projetados como uma substituição aos sistemas que usavam relés interconectados (como por exemplo na Figura 9.2) e sistemas de controle lógico de temporização. Os CLPs têm a grande vantagem da possibilidade de modificação do sistema de controle sem ter que refazer a conexões nos dispositivos de entrada e saída. A única coisa que o operador tem que fazer é inserir um conjunto de instruções diferentes. Os sistemas com CLPs são muito mais rápidos do que os sistema com relé.

O resultado é um sistema flexível que pode ser usado para sistemas de controle que variam totalmente em sua natureza e complexidade. Estes sistemas são muito usados na implementação de funções de controle lógico porque eles são fáceis de serem usados e programados

Os CLPs são similares aos computadores mas têm certas características que são específicas do seu uso como controladores. Estas características são:

- 1 Eles são robustos e projetados para resistir a vibrações, temperaturas, umidade e ruído.
- 2 O interfaceamento de entradas e saídas é interno ao controlador.
- 3 Eles são facilmente programáveis.

21.2 Estrutura básica de um CLP

A Figura 21.1 mostra a estrutura interna básica de um CLP. Ele consiste essencialmente de uma unidade central de processamento (CPU), memória e interfaces de entrada/saída. Todos os processos e controles da CPU são internos ao CLP. Ele possui um clock com uma frequência típica entre 1 e 8 MHz. Esta frequência determina a velocidade de operação do CLP e fornece a temporização e sincronismo para todos os dispositivos do sistema. Um sistema de barramento transporta informações e dados que entram e saem da CPU, memória e unidades de entrada/saída. Existe mais de um tipo de memória: um sistema de ROM para armazenamento permanente do sistema de operação e dados fixos, RAM para o programa do usuário, e *buffers* temporários para armazenar os canais de entrada/saída.

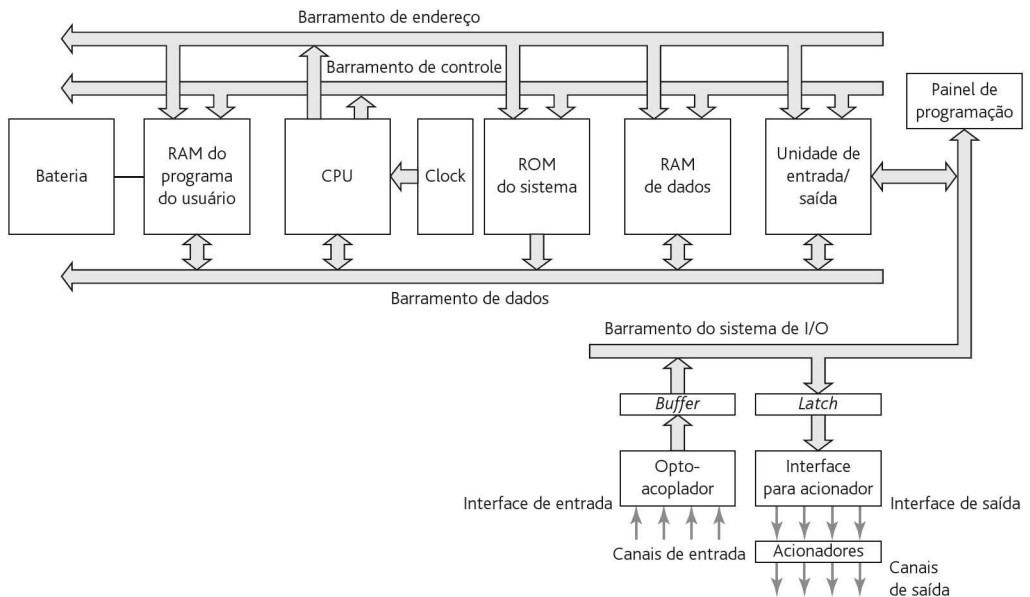


Figura 21.1 Arquitetura de um CLP.

21.2.1 Entrada/saída

As unidades de entrada e saída fornecem a interface entre o sistema e o mundo externo, e são por onde o processador recebe informações dos dispositivos externos e envia informações para os mesmos. As interfaces de entrada/saída fornecem isolamento e funções de condicionamento de sinal, de modo que os sensores e atuadores possam normalmente ser conectados diretamente a elas sem a necessidade de outros circuitos. As entradas podem ser desde chaves fim de curso que são ativadas quando algum evento ocorre a outros sensores, como de temperatura ou vazão. As saídas podem ser as bobinas de um motor, válvulas solenoides, etc. A isolamento elétrica do mundo externo é feita geralmente por meio de optoisoladores (veja a Seção 3.3).

A Figura 21.2 mostra a forma básica de um canal de entrada. O sinal digital que é geralmente compatível com o microprocessador no CLP é 5 V CC. Entretanto, o condicionamento, com isolamento, de sinal no canal de entrada permite uma ampla faixa de sinais que a entrada pode receber. Portanto, com um CLP maior podemos ter possíveis tensões de entrada de 5 V, 24 V, 110 V e 240 V. Um CLP pequeno pode ter apenas um valor de tensão de entrada, como, por exemplo, 24 V.

Uma unidade de saída digital tem nível de 5 V. As saídas são especificadas como de tipo relé, transistor ou *triac*. No caso do tipo relé, o sinal da saída do CLP é usado internamente para operar um relé, e assim é capaz de comutar correntes da ordem de alguns amperes no circuito externo. O relé isola o CLP do circuito externo e pode ser usado na comutação de sinais CC e ca. Entretanto, os relés são relativamente lentos na resposta. A saída do tipo transistor usa um transistor para comutar a corrente no circuito externo. Este tipo proporciona uma ação de comutação mais rápida. Os optoisoladores são usados com chaves de transistor para proporcionar isolamento entre o circuito externo e o CLP. A saída com transistor é usada apenas em comutação CC. As saídas com triac podem ser usadas para controlar cargas externas conectadas em fontes de alimentação ca. Os optoisoladores também são usados para fornecer isolamento. Portanto, podemos ter saídas de canais de saída que podem ser um sinal de comutação de 24 V, 100 mA, uma tensão CC de 110 V, 1 A ou ainda 240 V, 1 A ca, ou 240 V, 2 A CA a partir de um canal de saída com triac. Em um CLP pequeno, todas as saídas podem ser de um tipo, como, por exemplo, 240 V ca, 1 A. Entretanto, com CLPs modulares, uma faixa de saídas pode ser acomodada selecionando-se os módulos a serem usados.

Os termos **fornecimento de corrente** e **absorção de corrente** são usados para descrever a maneira na qual dispositivos CC são conectados a um CLP. No caso do fornecimento de corrente, usando-se o sentido convencional, que é do po-

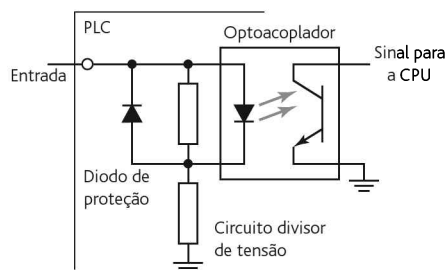


Figura 21.2 Canal de entrada.

sitivo para o negativo, um dispositivo de entrada recebe corrente do módulo de entrada (Figura 21.3(a)). Se a corrente flui de um módulo de saída para uma carga, então o módulo de saída fornece corrente (Figura 21.3(b)). No caso de absorção de corrente, um dispositivo de entrada fornece corrente para o módulo de entrada (Figura 21.3(c)). Se a corrente flui de uma carga para um módulo de saída, então este absorve corrente (Figura 21.3(d)).

A unidade de entrada/saída fornece a interface entre o sistema e o mundo externo, permitindo que conexões sejam feitas através dos canais de entrada/saída com dispositivos de entrada como sensores e dispositivos de saída como motores e solenoides. É também através de unidades de entrada/saída que os programas são inseridos a partir de um painel de programação. Cada ponto de entrada/saída tem um único endereço que pode ser usado pela CPU. É como se fosse uma fila de casas ao longo de uma rua: o número 10 pode ser a “casa” usada como entrada por um determinado sensor, enquanto que o número 45 pode ser a “casa” usada por uma saída de um determinado motor.

21.2.2 Entrada de programas

Os programas são inseridos na unidade de entrada/saída a partir de um pequeno dispositivo manual de programação, consoles desktop com uma unidade de display visual (VDU), teclado e display, ou através de uma conexão com um computador pessoal (PC) que é carregado com um pacote de software apropriado. Apenas após o programa ser desenvolvido no dispositivo de programação e estiver pronto é que ele é transferido para a unidade de memória do CLP.

Os programas em RAM podem ser alterados pelo usuário. Entretanto, para evitar a perda destes programas quando a fonte de alimentação é desligada, pode ser usada uma bateria no CLP para manter o conteúdo da RAM por um período de tempo. Após um programa ser desenvolvido em RAM, ele pode ser carregado em uma EPROM, passando assim a ser permanente. Nas especificações de um CLP pequeno, geralmente há uma especificação relativa à capacidade da memória de programa em termos do número de passos de programa que pode ser armazenado. Um passo de programa é uma instrução para a ocorrência de algum evento. Uma tarefa

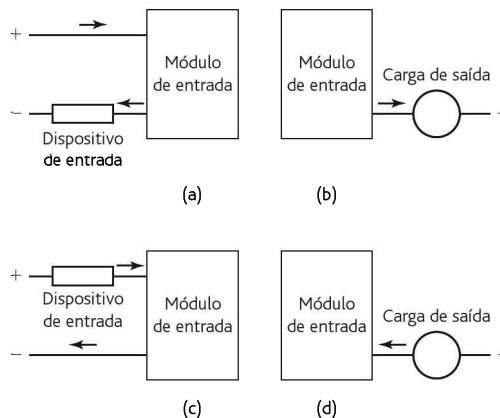


Figura 21.3 (a), (b) Fornecimento de corrente, (c), (d) absorção de corrente.

de programa pode consistir em vários passos que podem ser, por exemplo: examinar o estado da chave A, examinar o estado da chave B, se A e B estiverem fechadas, então energizar o solenoide P, o que pode resultar na operação de algum atuador. Quando isto acontece, uma outra tarefa pode ser iniciada. Tipicamente, o número de passos que podem ser manipulados por um CLP pequeno é da ordem de 300 a 1.000, o que geralmente é adequado para a maioria das situações de controle.

21.2.3 Tipos de CLPs

Os CLPs foram idealizados em 1968. Atualmente eles são muito usados, sendo encontrados desde pequenas unidades de módulo único, para uso com, por exemplo, 20 entradas/saídas digitais, até sistemas montados em rack que podem ser usados por um grande número de entradas/saídas, que podem ser digitais e analógicas e possuem também modos de controle PID. O tipo de módulo único é normalmente usado para pequenos controladores programáveis e é fornecido como um módulo compacto completo com fonte de alimentação, processador, memória e unidades de entrada/saída. Tipicamente, estes CLPs podem ter 6, 8, 12 ou 24 entradas, 4, 8 ou 16 saídas e uma memória que pode armazenar de 300 a 1.000 instruções. Por exemplo, o MELSEC FX3U tem modelos que têm 6, 8, 12 ou 24 entradas, 4, 8 ou 16 saídas a relé e uma memória que pode armazenar de 300 a 1.000 instruções. Alguns sistemas são capazes de serem estendidos para lidar com mais entradas e saídas através de uma conexão de módulos de entrada/saída a eles.

Os sistemas com um número maior de entradas e saídas são modulares e projetados para serem instalados em racks. Estes tipos consistem de módulos separados para fonte de alimentação, processador, entrada/saída, etc., e são montados em trilhos dentro de um gabinete metálico. O tipo rack pode ser usado por todos os tamanhos de controladores programáveis e tem diversas unidades funcionais em módulos individuais que podem ser conectados em soquetes na base do rack. A combinação de módulos necessária para uma finalidade em particular é decidida pelo usuário e inserida no rack; assim, o número de conexões de entrada/saída pode ser ampliado apenas acrescentando-se mais módulos. Por exemplo, o CLP SIMATIC S7-300/400 é montado em rack com componentes para fonte de alimentação, CPU, módulos de interface de entrada/saída, módulos de sinal que podem ser usados para prover condicionamento de sinal para entradas ou saídas, e módulos de comunicação que podem ser usados para conectar CLPs entre si ou com outros sistemas.

21.3 Processamento de entrada/saída

Um CLP executa continuamente seu programa e o atualiza como resultado dos sinais de entrada. Cada *loop* é denominado **ciclo**. Existem dois métodos que podem ser usados para o processamento de entrada/saída: atualização contínua e cópia em massa de entrada/saída.

21.3.1 Atualização contínua

A atualização contínua envolve a varredura da CPU nos canais de entrada conforme as instruções do programa assim determinem. Cada ponto de entrada é examinado

individualmente e o seu efeito no programa é determinado. Existe um atraso intrínseco, tipicamente de cerca de 3 ms, quando cada entrada é examinada para garantir que apenas sinais de entrada válidos sejam lidos pelo microprocessador. Este atraso faz com que o microprocessador evite contar um sinal de entrada duas vezes, ou com mais frequência, se houver o repique (*bounce*) no contato de uma chave. Um determinado número de entradas pode ter que passar por um processo de leitura sequencial (varredura), cada uma com um atraso de 3 ms, antes do programa ter a instrução para que uma operação lógica seja executada e uma saída seja atualizada. As saídas passam por um *latch*, de modo que elas retêm os seus status até que a próxima atualização ocorra.

21.3.2 Cópia em massa de entrada/saída

Como existe um atraso de 3 ms em cada entrada, no caso da atualização contínua, o tempo gasto para examinar algumas centenas de pontos de entrada/saída pode se tornar comparativamente longo. Para possibilitar uma rapidez maior na execução de um programa, uma área específica da RAM é usada como um *buffer* entre a lógica de controle e a unidade de entrada/saída. Cada unidade de entrada/saída tem um endereço na memória. No início de cada ciclo do programa, a CPU lê todas as entradas (varredura) e copia o status delas para os endereços de entrada/saída da RAM. Conforme o programa é executado, o dado de entrada é lido, conforme necessário, a partir da RAM e a operação lógica é executada. Os sinais de saída resultantes são armazenados na seção de entrada/saída reservada na RAM. No final de cada ciclo do programa, todas as saídas são transferidas da RAM para os canais de saída. As saídas são provenientes de *latches*, de modo que elas retêm seus status até a próxima atualização. A sequência é:

- 1 Ler todas as entradas e copiá-las para a RAM.
- 2 Buscar, decodificar e executar todas as instruções do programa na sequência, copiando as instruções de saída para a RAM.
- 3 Atualizar todas as saídas.
- 4 Repetir a sequência.

Um CLP gasta tempo para completar um ciclo de varredura nas entradas e atualização das saídas de acordo com as instruções do programa e, assim, as entradas não são observadas todo o tempo, mas apenas examinadas periodicamente. O tempo de um ciclo de CLP típico é da ordem de 10 a 50 ms e, assim, as entradas e saídas são atualizadas a cada 10 ou 50 ms. Isto significa que se uma entrada for ativada rapidamente no momento errado do ciclo, ela pode não ser detectada. Portanto, para um CLP com um ciclo de 40 ms, a frequência máxima dos impulsos digitais que pode ser detectada será se um pulso ocorrer a cada 40 ms. O CLP compacto MELSEC FX3U da Mitsubishi tem um tempo de ciclo de programa de 0,065 μ s por instrução lógica e, desta forma, quanto mais complexo o programa, maior o tempo do ciclo.

21.3.3 Endereços de I/O

O CLP tem que ser capaz de identificar cada entrada e saída em particular associando um endereço a cada uma, assim como as casas em uma cidade têm endereços para possibilitar que a correspondência seja entregue para as pessoas certas. Em

um CLP pequeno, os endereços são apenas números precedidos por uma letra para indicar se eles são entradas ou saídas. Por exemplo, nos CLPs Mitsubishi e Toshiba as entradas são identificadas como X400, X401, X402, etc., e as saídas, como Y430, Y431, etc. No caso de CLPs maiores contendo alguns racks de canais de entrada e saída e vários módulos em cada rack, os racks e os módulos são numerados, assim, uma entrada ou saída é identificada pelo número do seu rack seguido do número do módulo no rack e, em seguida, um número que mostra o número do seu terminal no módulo. Por exemplo, o CLP-5 da Allen-Bradley tem I:012/03 para indicar uma entrada no rack 01 no módulo 2 e no terminal 03.

21.4 Programação em ladder

O tipo de programação normalmente usada nos CLPs é a **ladder***. Cada tarefa (lógica de controle) do programa é especificada como os degraus de uma escada, ou seja, os ramos. Portanto, este ramo pode especificar o estado das chaves A e B, as entradas, a serem examinadas, e se A e B estiverem fechadas, um solenoide, a saída, é energizado. A Figura 21.4 ilustra esta ideia fazendo uma comparação com um circuito elétrico.

A sequência seguida por um CLP quando executa um programa pode ser resumida como:

- 1 Ler as entradas associadas com um ramo do programa ladder.
- 2 Resolver a operação lógica que envolve estas entradas.
- 3 Setar/resetar as saídas no ramo.
- 4 Mover para o próximo ramo e repetir as operações 1, 2 e 3.
- 5 Mover para o próximo ramo e repetir as operações 1, 2 e 3.
- 6 Mover para o próximo ramo e repetir as operações 1, 2 e 3.
- 7 E assim por diante até o final do programa, com cada ramo do programa ladder percorrido por sua vez. O CLP retorna então para o início do programa e inicia novamente.

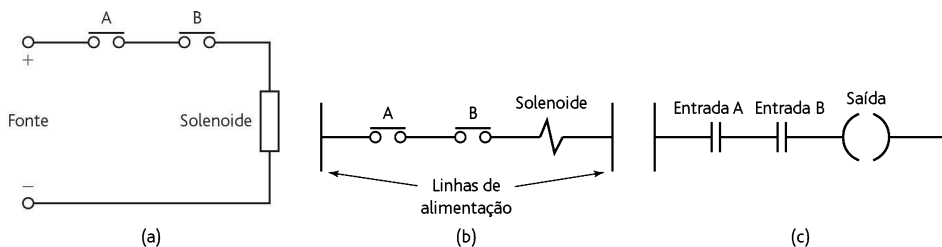


Figura 21.4 (a), (b) Formas alternativas de se desenhar um circuito elétrico, (c) comparável a um ramo em um programa ladder.

* N. de T.: O nome ladder foi dado devido a representação gráfica da linguagem ser semelhante a uma escada (*ladder*). Duas barras verticais paralelas são interligadas pela lógica de controle ou ramos, que são os degraus (*rungs*) da escada. Veja a Figura 21.5.

A programação de um CLP baseada no uso de **diagramas ladder** envolve a escrita de um programa de forma similar ao desenho de um circuito de comutação. O diagrama ladder consiste de duas linhas verticais representando os trilhos, ou as linhas, de alimentação. Os circuitos são conectados como linhas horizontais, ou seja, os ramos (lógica de controle) do ladder, entre estas duas linhas verticais. A Figura 21.5 mostra os símbolos básicos padrão que são usados e um exemplo de ramos em um diagrama ladder. Ao desenhar uma linha de circuito para um ramo, as entradas sempre têm que preceder as saídas e tem que existir pelo menos uma saída em cada linha. Cada ramo tem que iniciar com uma entrada ou uma série de entradas e terminar com uma saída.

Para ilustrar o desenho de um diagrama ladder, considere uma situação onde a saída do CLP é para energizar um solenoide quando uma chave de partida normalmente aberta conectada na entrada é ativada quando fechada (Figura 21.6(a)). O programa desejado é mostrado na Figura 21.6(b)). Começando pela entrada, temos o símbolo normalmente aberto $| |$. Esta entrada pode ter o endereço X400. A linha termina com a saída, o solenoide, com o símbolo $()$. Esta saída pode ter o endereço Y430. Para indicar fim do programa, o ramo FIM é indicado. Quando a chave é fechada, o solenoide é ativado. Esta saída poderia ser, por exemplo, o solenoide de uma válvula que abre para permitir que a água entre em um recipiente.

Um outro exemplo poderia ser um controle *on/off* de temperatura (Figura 21.7(a)) no qual a entrada vai de nível baixo para alto quando o sensor de temperatura alcança a temperatura desejada. A saída muda então do estado ligado para desligado. O sensor de temperatura mostrado na figura é um termistor conectado na configuração de uma ponte com a saída conectada a um amplificador operacional

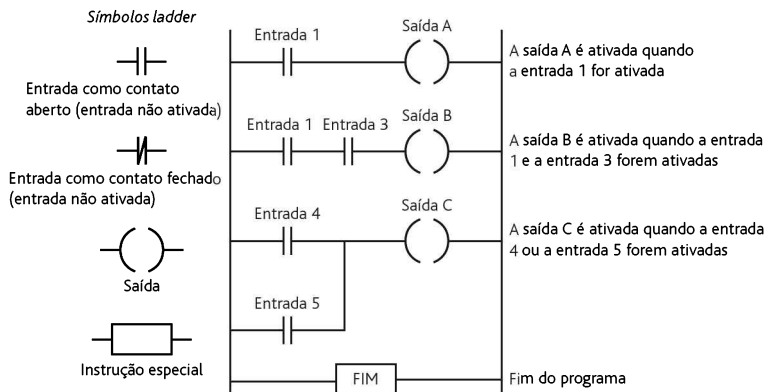


Figura 21.5 Programa ladder.

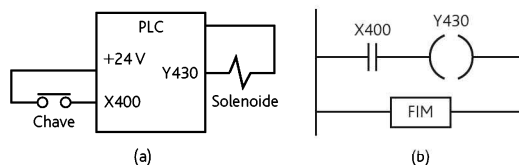


Figura 21.6 Uma chave controlando um solenoide.

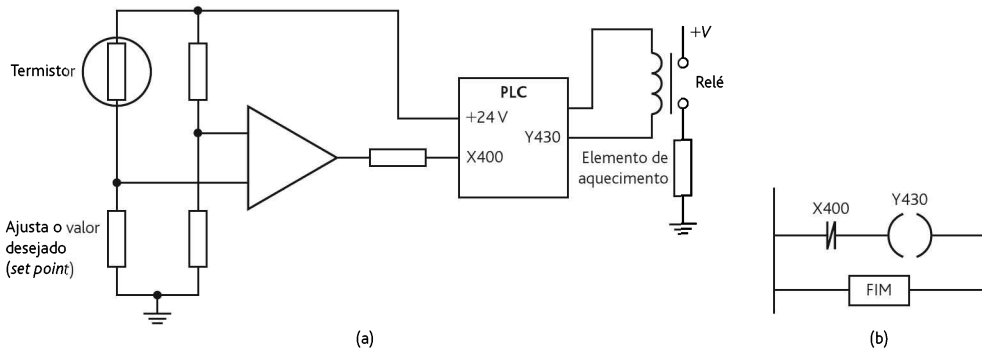


Figura 21.7 Sistema de controle de temperatura.

configurado como um comparador (veja a Seção 3.2.7). O programa (Figura 21.7(b)) mostra a entrada como um par de contatos normalmente fechados ativando, assim, a saída. Quando os contatos são abertos, a saída é desligada.

21.4.1 Funções lógicas

As funções lógicas podem ser obtidas combinando-se as chaves (veja a Seção 5.2), e a seguir mostramos como podemos escrever os programas ladder com estas combinações (Figura 21.8):

1 AND

A Figura 21.8(a) mostra uma situação onde uma bobina não é energizada a menos que duas chaves, normalmente abertas, sejam ambas fechadas. As chaves A e B têm que ser fechadas, o que representa uma situação lógica AND. O

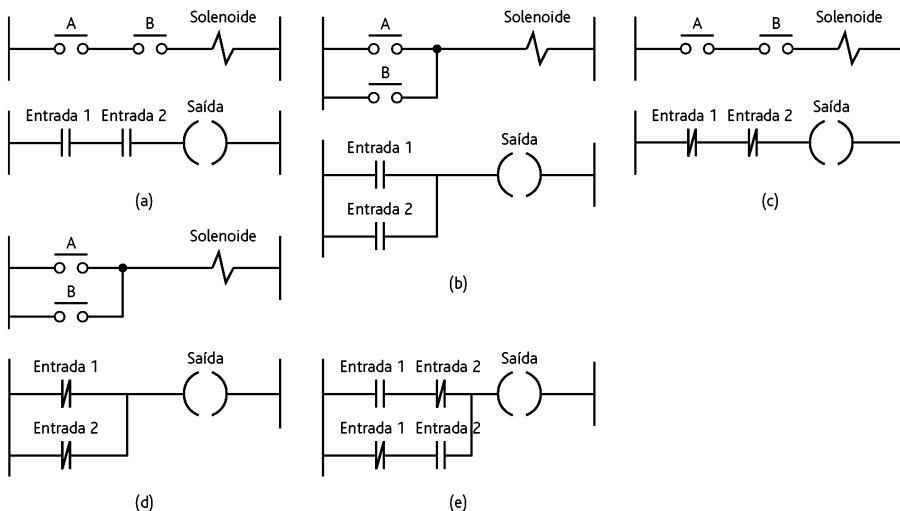


Figura 21.8 (a) AND, (b) OR, (c) NOR, (d) NAND, (e) EX-OR.

diagrama ladder equivalente começa com ||, nomeado como Entrada 1, para representar a chave A e em série com ela aparece ||, nomeado de Entrada 2, para representar a chave B. A linha termina então com () para representar a saída.

2 OR

A Figura 21.8(b) mostra uma situação em que uma bobina não é energizada até que a chave A, normalmente aberta, ou a chave B, também normalmente aberta, for fechada. A situação corresponde à da porta lógica OR. O diagrama ladder equivalente começa com ||, nomeado de Entrada 1, para representar a chave A e em paralelo com ||, nomeado de Entrada 2, para representar a chave B. A linha termina então com () para representar a saída.

3 NOR

A Figura 21.8(c) mostra como podemos representar a linha do programa ladder para a porta NOR. Como deve haver uma saída quando nem A, nem B têm uma entrada, e quando há uma entrada em A ou B a saída cessa, o programa ladder mostra a Entrada 1 em paralelo com a Entrada 2, sendo ambas representadas por contatos normalmente fechados.

4 NAND

A Figura 21.8(d) mostra uma porta NAND. Não há saída quando ambas as entradas A e B tiverem entrada. Portanto, para a linha e programa ladder obter uma saída, não podemos ter nenhuma das entradas (1 e 2).

5 EXCLUSIVE-OR (EX-OR)

A Figura 21.8(e) mostra como podemos desenhar a linha de programa ladder para uma porta EX-OR, em que não há saída quando não houver nenhuma das entradas (1 e 2) e quando houver as duas entradas (1 e 2). Note que representamos cada entrada por dois conjuntos de contatos, um normalmente aberto e outro normalmente fechado.

Considere uma situação em que a chave normalmente aberta A tem que ser ativada e uma das outras duas chaves normalmente abertas B e C tem que ser ativadas para que uma bobina seja energizada. Podemos representar esta configuração de chaves como a chave A em série com duas chaves em paralelo (B e C) (Figura 21.9(a)). Para a bobina ser energizada, precisamos fechar A, e B ou C. A chave A, quando considerada com as chaves em paralelo, resulta em uma situação lógica AND. As duas chaves em paralelo resultam em uma lógica OR. Então, temos a combinação de duas portas. A tabela-verdade é:

Entradas			Saída
A	B	C	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

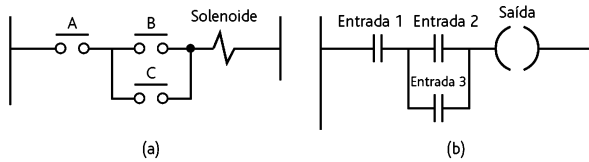


Figura 21.9 Chaves controlando um solenoide.

Para o diagrama ladder, começamos com ||, nomeado de Entrada 1, para representar a chave A. Esta está em série com dois || em paralelo, nomeados de Entrada 1 e Entrada 2, para as chaves B e C. A linha termina então com () para representar a saída, a bobina. A Figura 21.9(b) mostra a linha.

Como um simples exemplo de um programa usando portas lógicas, considere os requisitos para que a saída do solenoide controle a válvula que abre a porta da loja quando o lojista fecha uma chave para abrir a loja e o cliente se aproxima da porta e é detectado por um sensor que, ativado, fornece um nível alto. A tabela-verdade para este sistema é portanto:

Chave que abre a loja	Sensor de aproximação para clientes	Saída para o solenoide
Off	Off	Off
Off	On	Off
On	Off	Off
On	On	On

Esta tabela-verdade é a de uma porta AND e, portanto, o programa para um CLP controlar a porta é mostrado na Figura 21.10.

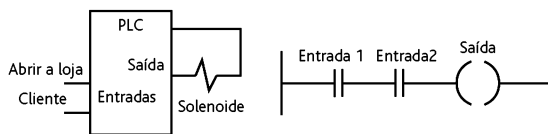


Figura 21.10 Sistema de controle da porta de uma loja.

21.5 Lista de instruções

Cada ramo horizontal no diagrama ladder representa uma linha no programa e o ladder inteiro fornece o programa completo na linguagem ladder. O programador pode inserir o programa no CLP usando um teclado com símbolos gráficos para os elementos ladder, ou usando a tela de um computador e um mouse para selecionar os símbolos, e um painel de programação ou um computador traduz estes símbolos em linguagem de máquina que pode ser armazenada na memória do CLP. Existe uma forma alternativa de inserir um programa, e que é a tradução do programa ladder em lista de instruções e então sua inserção no painel de programação ou no computador.

Tabela 21.1 Mnemônicos do código de instrução

IEC 1131-3	Mitsubishi	OMRON	Siemens	Operação	Diagrama ladder
LD	LD	LD	A	Carrega o operando no registrador de resultado	Começa um ramo com contatos abertos
LDN	LDI	LD NOT	AN	Carrega o operando negativo no registrador de resultado	Começa um ramo com contatos fechados
AND	AND	AND	A	Carrega o operando negativo no registrador de resultado	Um elemento em série com contatos abertos
ANDN	ANI	AND NOT	AN	AND booleana com operando negativo	Um elemento em série com contatos fechados
OR	OR	OR	O	OR booleana	Um elemento em paralelo com contatos abertos
ORN	ORI	OR NOT	ON	OR booleana com operando negativo	Um elemento em paralelo com contatos fechados
ST	OUT	OUT	=	Armazena o registrador de resultado no operando	Uma saída de um ramo

As listas de instruções consistem em uma série de instruções em que cada instrução inicia em uma linha separada. Uma instrução consiste de um operador seguido de um ou mais operandos, ou seja, o argumento do operador. Em termos de programas ladder, cada operador em um programa pode ser considerado como um elemento ladder. Portanto, podemos ter como equivalente a uma entrada em um programa ladder:

LD A (*Carrega a entrada A*)

O operador é LD para carga, o operando A é o objeto a ser carregado, e as palavras entre asteriscos (*) dentro dos parênteses são comentários que explicam o que é a operação e não fazem parte das instruções do programa para o CLP, mas ajudam o leitor a compreender o programa.

Os códigos mnemônicos usados por diferentes fabricantes de CLP diferem entre si, mas o padrão internacional IEC 1131-3 proposto é bastante usado. A Tabela 21.1 mostra mnemônicos comuns. Nos exemplos discutidos no restante do capítulo, onde descrições gerais não são usadas, os mnemônicos da Mitsubishi são usados. Entretanto, os que são usados por outros fabricantes não são muito diferentes destes, e os princípios envolvidos são os mesmos.

21.5.1 Listas de instruções e funções lógicas

A seguir mostramos como ramos individuais em um diagrama ladder são inseridos usando-se os mnemônicos da Mitsubishi onde funções lógicas são envolvidas (Figura 21.11).

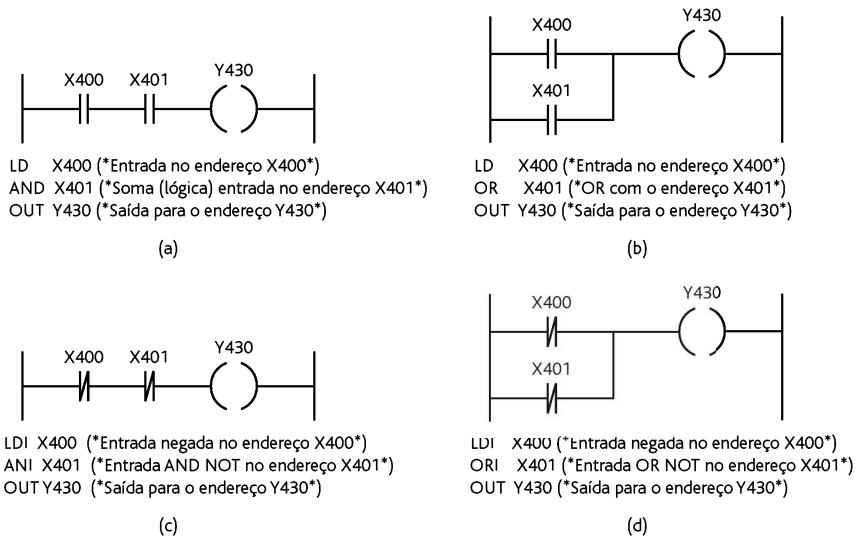


Figura 21.11 (a) AND, (b) OR, (c) NOR, (d) NAND.

21.5.2 Listas de instruções e desvios

A porta EX-OR mostrada na Figura 21.12 tem dois ramos com uma operação AND em cada ramo. Neste caso, o CLP da Mitsubishi (Figura 21.12(a)) usa uma instrução ORB para indicar “OR juntas em ramos paralelos”. A primeira instrução é para o par de contatos normalmente aberto X400, a próxima instrução é para o conjunto de contatos normalmente fechado X401 (ANI 401). A terceira instrução descreve uma nova linha, que é reconhecida como uma nova linha porque inicia com LDI (toda nova linha começa com LD ou LDI). Como a primeira linha não foi finalizada atra-

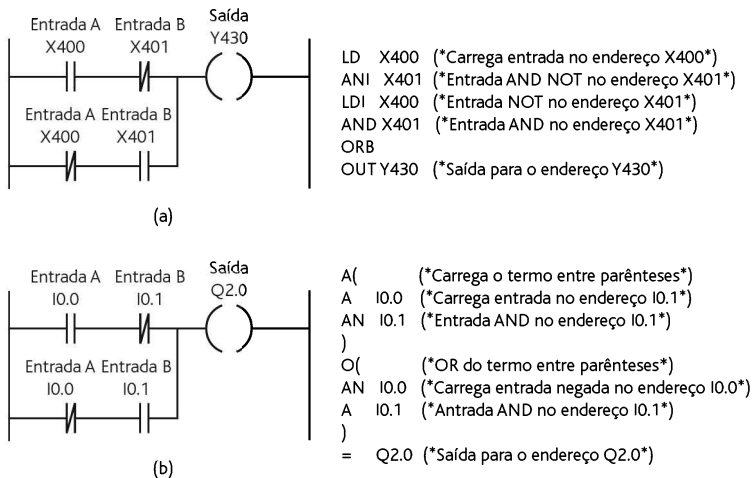


Figura 21.12 EX-OR.

vés de uma saída, o CLP reconhece que uma segunda linha em paralelo está envolvida e lê juntos os elementos listados até encontrar a instrução ORB. Esta instrução indica ao CLP que ele deve fazer uma OR do resultado da primeira e da segunda instruções com o novo ramo com a terceira e quarta instruções. A lista finaliza com a saída OUT Y430. A Figura 21.12(b) mostra a versão Siemens da porta EX-OR. Os parênteses são usados para indicar que certas instruções são para serem executadas como um bloco e são usados da mesma forma que os parênteses em qualquer equação matemática. Por exemplo, $(1 + 2)/4$ significa que 1 e 2 têm que ser somados antes da divisão por 4. Portanto, com a lista de instruções do Siemens, o A (significa que a instrução de carga A é aplicada apenas após os passos entre parênteses terem sido finalizados) é alcançado. O padrão IEC 1131-3 para esta programação usa parênteses da mesma forma usada no exemplo do CLP Siemens anterior.

21.6 Operação *latch* e relés internos

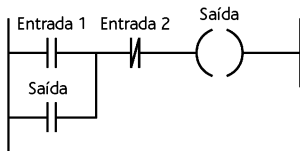


Figura 21.13 Um circuito *latch*.

Frequentemente existem situações onde é necessário manter uma bobina energizada, mesmo quando a entrada que foi energizada cessar. O termo **circuito *latch*** é usado para o circuito que realiza esta operação. Este é um circuito que, após energizado, mantém o estado até receber uma nova entrada. Ele “lembra” do seu último estado. Um exemplo de um circuito *latch* é mostrado na Figura 21.13. Quando a Entrada 1 é energizada e fechada, haverá uma saída. Entretanto, quando há uma saída, um conjunto de contatos associados com a saída é energizado e fechado. É realizada uma operação OR entre estes contatos e os contatos da Entrada 1. Portanto, mesmo se os contatos da Entrada 1 abrirem, o circuito ainda manterá a saída energizada. A única forma de liberar a saída é acionando a Entrada 2, que é um contato normalmente fechado.

Como um exemplo do uso de um circuito *latch*, considere a necessidade de um CLP controlar um motor de forma que quando o botão de partida é momentaneamente acionado, o motor parte, e quando a chave pare é usada, o motor é desligado. A segurança tem que ser uma prioridade em um projeto com CLP, de modo que os botões de pare têm que ser conectados de forma a não depender do software do CLP, pois no caso de uma falha da chave de parada ou do CLP, o sistema é automaticamente seguro. Em um sistema com CLP, o sinal de parada pode ser fornecido por uma chave como mostrado na Figura 21.14(a). Para dar a partida, pressionamos momentaneamente o botão de partida e o relé de controle interno do motor faz o *latch* (memoriza) deste acionamento e a saída permanece ligada. Para parar, abrimos momentaneamente a chave de parada, e isto desativa o *latch* da chave de partida. Entretanto, se a chave de parada não puder ser acionada, então não podemos parar o sistema. Portanto, este sistema não pode ser usado por ser inseguro, porque se houver uma falha e a chave não puder ser operada, então não será fornecido nenhum sinal de parada. O que precisamos é de um sistema que ainda possa parar mesmo que ocorra uma falha no botão de parada. A Figura 21.14(b) mostra este sistema. Este programa agora tem o botão de parada como contatos abertos. Entretanto, como a conexão do botão pare tem normalmente contatos fechados, então o programa recebe o sinal para fechar os contatos. Pressionando o botão pare, os contatos no programa são abertos e o sistema para.

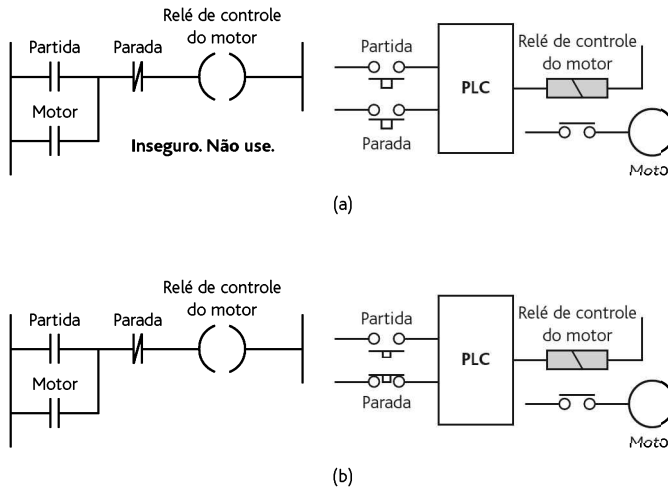


Figura 21.14 Sistema de parada: (a) inseguro, (b) seguro.

21.6.1 Relés internos

O termo **relé interno**, **relé auxiliar** ou **marcador** é usado para o que pode ser considerado com um relé interno do CLP. Estes se comportam como relés com seus contatos associados, mas na realidade não simulações de relés feitas pelo software do CLP. Alguns têm baterias de *back-up* de modo que podem ser usados em circuitos para garantir um desligamento seguro da planta no caso de falha de alimentação. Os relés internos podem ser muito úteis na implementação de seqüências de comutação.

Os relés internos são usados frequentemente quando existem programas com condições de entrada múltiplas. Considere a situação em que o acionamento de uma saída depende de duas configurações de entrada diferentes. A Figura 21.15(a) mostra como podemos desenhar um diagrama ladder usando relés internos. O primeiro ramo mostra uma configuração de entrada usada para controlar a bobina do relé interno IR1. O segundo ramo mostra a outra configuração de entrada controlando a bobina do relé interno IR2. Os contatos dos dois relés são então submetidos a uma operação OR para controlar a saída.

Um outro uso dos relés internos é no acionamento de múltiplas saídas. A Figura 21.15(b) mostra este programa ladder. Quando os contatos do botão de partida são fechados, o relé interno é ativado e faz o *latch* da entrada. Ele também ativa a Saída 1 e torna possível a ativação das Saídas 2 e 3.

Um outro exemplo do uso dos relés internos é no reset de um *latch*. A Figura 21.16 mostra o diagrama ladder. Quando os contatos da Entrada 1 são pressionados momentaneamente, a saída é energizada. Os contatos da saída são fechados, promovendo o *latch* da saída, ou seja, mantendo ela acionada mesmo quando os contatos da entrada não estiverem mais fechados. A saída pode ser “desativada” pela abertura dos contatos do relé interno. Isto ocorre se a Entrada 2 for fechada e energizar a bobina do relé interno.

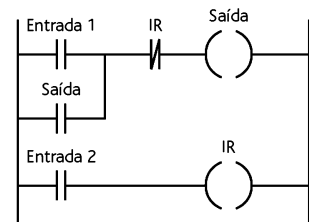


Figura 21.16 Resetando um *latch*.

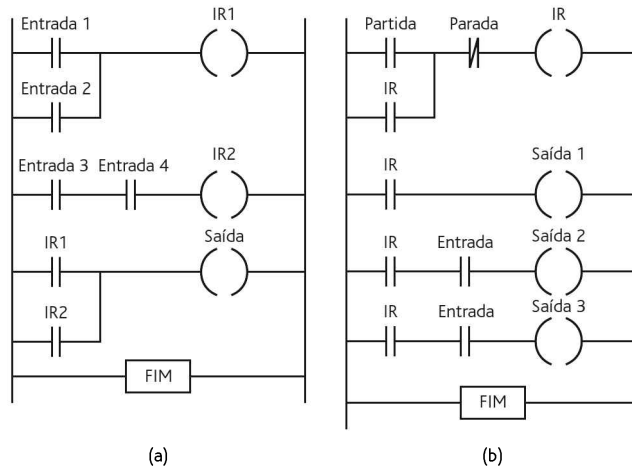


Figura 21.15 (a) Configuração de uma saída controlada por duas entradas, (b) acionamento de múltiplas saídas.

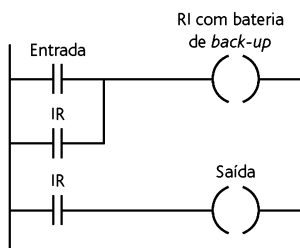


Figura 21.17 Uso de um relé interno com bateria de *back-up*.

Um exemplo do uso de um relé interno com bateria de *back-up* é mostrado na Figura 21.17. Quando os contatos de Entrada 1 fecham, a bobina do relé interno com bateria de *back-up* é energizada. Isto fecha os contatos do relé interno, e mesmo que os contatos da entrada abram como resultado de uma falha na alimentação, os contatos do relé interno permanecem fechados. Isto significa que a saída controlada pelo relé interno permanece energizada, mesmo quando há uma falha na alimentação.

21.7 Sequenciamento

Freqüentemente existem situações de controle em que seqüências de saídas são necessárias, com a comutação de uma saída para a outra controlada por sensores. Considere a necessidade de um programa ladder para um sistema pneumático (Figura 21.18) com válvulas de duplo solenoide controlando dois cilindros de ação dupla A e B se as chaves fim de curso a-, a+, b-, b+ forem usadas para detectar os limites dos movimentos da haste do pistão nos cilindros e a seqüência de atuação dos cilindros A+, B+, A-, B- for necessária. Um programa possível é mostrado na figura. Uma entrada com chave de partida foi incluída no primeiro ramo. Portanto, a extensão do cilindro A, ou seja, o solenoide A+ energizado, ocorre apenas quando a chave de partida é fechada e a chave b- é fechada, indicando que o cilindro B está retraído. Quando o cilindro A está estendido, a chave a+, que indica a extensão de A, é ativada. Isto leva uma saída para o solenoide B+ que resulta na extensão de B. Isto fecha a chave que indica a extensão de B, ou seja, a chave b+, e leva a saída para o solenoide A- e à retração do cilindro A. Esta retração fecha a chave fim de curso a- e produz a saída para o solenoide B- que resulta na retração de B. Isto conclui o ciclo do pro-

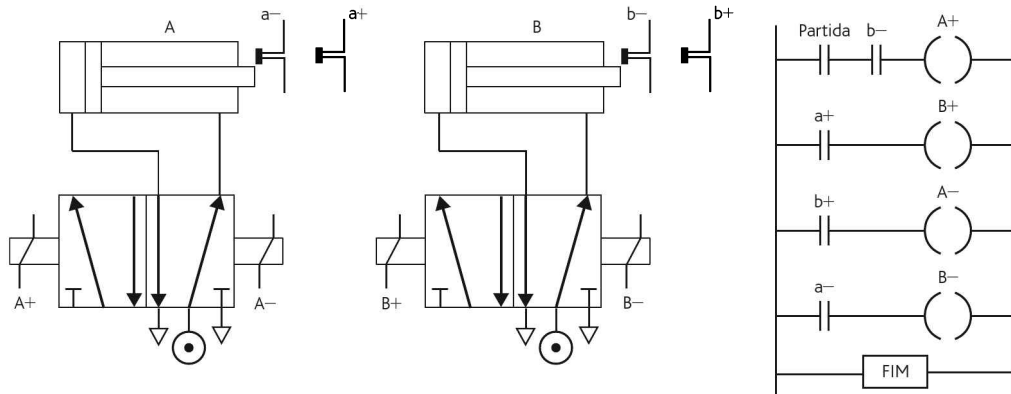


Figura 21.18 Sequenciamento de cilindros.

grama e conduz ao primeiro ramo novamente, que aguardará o fechamento da chave de partida antes de repetir esta seqüência.

Como mais uma ilustração, considere o problema de implementar um programa ladder para controlar um sistema pneumático com válvulas controladas por duplo solenoide e dois cilindros A e B se as chaves fim de curso a-, a+, b- e b+ são usadas para detectar os limites do movimento do pistões nos cilindros e a seqüência necessária é para o pistão A estender, seguido pela extensão do pistão B, para então o pistão B retrair e finalmente completar o ciclo com a retração do pistão A. Pode ser usado um relé interno para comutar entre os grupos de saídas a fim de produzir a forma de controle para os cilindros pneumáticos, o que é denominado **controle em cascata** (veja a Seção 7.5). A Figura 21.19 mostra um programa possível. Quando a chave de partida é fechada, o relé interno é ativado. Isto energiza o solenoide A+, resultando na extensão do cilindro A. Quando estendido, ele fecha a chave fim de

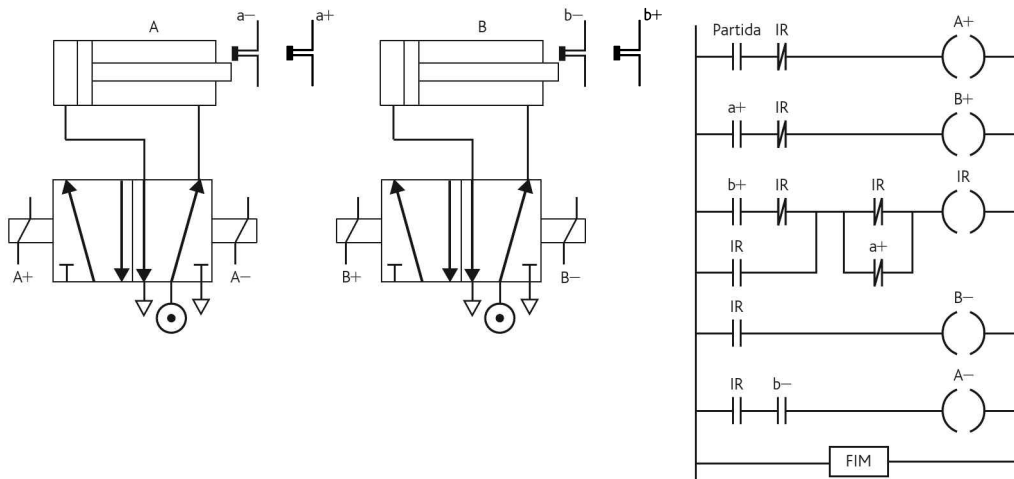


Figura 21.19 Sequenciamento de cilindros.

curso a+ e o pistão no cilindro B é estendido. Quando isto ocorre, ele fecha a chave fim de curso b+. Isto ativa o relé. Como resultado, o solenoide B- é energizado e o pistão em B se retrai. Quando isto fecha a chave fim de curso b-, o solenoide A- é energizado e o pistão no cilindro A se retrai.

21.8 Temporizadores e contadores

As seções anteriores neste capítulo abordaram tarefas que necessitam de conexões de contatos em série e em paralelo. Entretanto, existem tarefas que podem envolver atrasos de tempo e contagem de eventos. Estes requisitos podem ser atendidos com o uso de temporizadores e contadores que são fornecidos como características de CLPs. Eles podem ser controlados por instruções lógicas e representados em diagramas ladder.

21.8.1 Temporizadores

Uma abordagem comum usada por fabricantes de CLPs é considerar temporizadores que se comportam como relés com bobinas que, quando energizadas, resultam no fechamento ou abertura de contatos após um tempo determinado. O temporizador é tratado como uma saída para um ramo com controle exercido sobre pares de contato situados em outra parte (Figura 21.20(a)). Outros CLPs consideram um temporizador como um bloco de atraso em um ramo que atrasa os sinais no ramo que chegam na saída (Figura 21.20(b)).

Os CLPs geralmente possuem apenas um temporizador para ligar (TON), sendo que os menores provavelmente tenham apenas este tipo. Este temporizador espera por um período de tempo de atraso para ligar (Figura 21.20(c)), como, por exemplo, um período que pode ser definido entre 0,1 e 999 s em passos de 0,1 s. Podem existir outras faixas e resoluções.

Como uma ilustração do uso de um temporizador para sequenciamento, considere o diagrama ladder mostrado na Figura 21.21(a) ou (b). Quando a entrada In 1 é ligada, a saída Out 1 é ligada. Os contatos associados com esta saída fazem iniciar o temporizador. Os contatos do temporizador fecharão após o atraso de tempo ajustado. Quando isto acontece, a saída Out 2 é ligada.

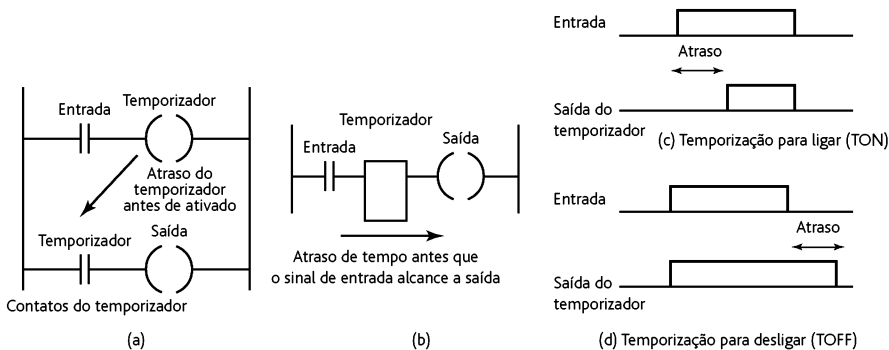


Figura 21.20 (a), (b) Temporização para ligar, (c) temporização com atraso para ligar, (d) temporização com atraso para desligar.

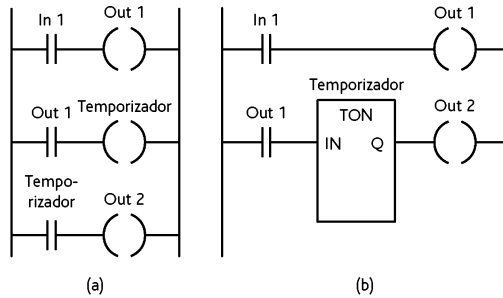


Figura 21.21 Sequência temporizada.

Os temporizadores podem ser conectados juntos, ou em **cascata**, para gerar atrasos maiores do que seria possível com apenas um temporizador. A Figura 21.22 mostra esta configuração. Quando os contatos de entrada fecham, o Temporizador 1 é iniciado. Após o seu atraso de tempo, seus contatos fecham e o Temporizador 2 é iniciado. Após o atraso de tempo deste, seus contatos se fecham e a saída é ativada.

A Figura 21.23 mostra um programa que pode ser usado para fazer com que uma saída seja ligada por 0,5 s, em seguida desligada por 0,5 s, novamente ligada por 0,5 s, e novamente desligada por 0,5 s e assim por diante. Quando os contatos de entrada se fecham, o Temporizador 1 é iniciado e liga após 0,5 s, que é o tempo para o qual foi ajustado. Após este tempo os contatos do Temporizador 1 fecham e iniciam o Temporizador 2. Este liga após 0,5 s, que é o tempo ajustado para ele, e abre os seus contatos. Isto faz com que o Temporizador 1 seja desligado. Com isto, os seus contatos se abrem e desligam o Temporizador 2. Isto então fecha os seus contatos e inicia todo o ciclo novamente. O resultado é que os contatos do temporizador para o Temporizador 1 são ligados por 0,5 s, em seguida desligados por 0,5 s, ligados por 0,5 s, e assim por diante. Portanto, a saída é ligada por 0,5 s, desligada por 0,5 s, ligada por 0,5 s, e assim por diante.

A Figura 21.24 mostra como um temporizador com atraso para desligar, ou seja, um temporizador que desliga uma saída após um atraso de tempo depois de ser energizado, pode ser projetado. Quando os contatos de entrada são momentaneamente fechados, a saída é energizada e o temporizador é iniciado. Os contatos de saída fazem o *latch* da entrada e mantêm a saída ligada. Após o tempo programado no temporizador, ele liga e desfaz o circuito *latch*, fazendo com que a saída seja desligada.

21.8.2 Contadores

Os contadores são usados quando há a necessidade de contar um número especificado de operações de um contato, como, por exemplo, onde os itens passam ao longo de uma correia

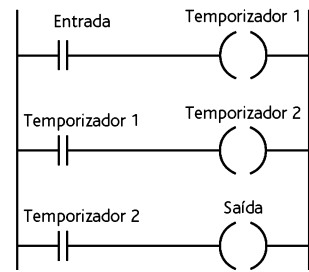


Figura 21.22 Temporizadores em cascata.

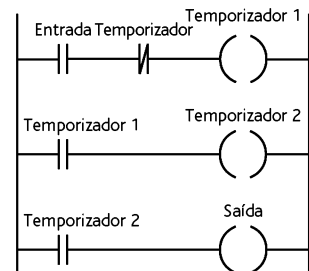


Figura 21.23 Temporizador cíclico on/off.

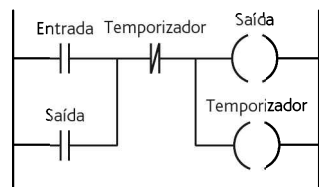


Figura 21.24 Temporizador com atraso para desligar.

transportadora para dentro de caixas, e quando o número especificado de itens tiver entrado em uma caixa, o próximo item é direcionado para outra caixa. Os circuitos contadores são apresentados como característica interna de CLPs. Na maioria dos casos, os contadores operam como **contadores decrescentes**. Isto significa que eles contam decrescentemente do valor atual até zero, ou seja, eventos são subtraídos do valor ajustado. Quando chegam ao zero, os contatos do contador mudam de estado. Um **contador crescente** conta para cima a partir do valor ajustado, ou seja, os eventos são somados até que o número chegue ao valor ajustado. Quando este valor é alcançado, os contatos do contador mudam de estado.

Diferentes fabricantes de CLPs lidam com contadores de formas diferentes. Alguns consideram o contador como sendo constituído de dois elementos básicos: uma bobina de saída para que o contador possa ser usado em outros ramos, como, por exemplo, nos CLPs Mitsubishi e Allen-Bradley. Outros tratam os contadores como um bloco intermediário em um ramo a partir do qual emanam sinais quando o contador é acessado, como, por exemplo, nos CLPs Siemens. Como uma ilustração, a Figura 21.25 mostra um circuito básico de contagem. Quando há um pulso na entrada In 1, o contador é resetado. Quando há um pulso na entrada In 2, o contador inicia a contagem. Se o contador for ajustado para, por exemplo, 10 pulsos, então, quando entrarem 10 pulsos em In 2, os contatos do contador se fecham e a saída Out 1 será ativada. Se em qualquer momento durante a contagem houver uma entrada In 1, o contador será resetado, iniciando tudo novamente, e o contador contará 10 pulsos.

Como uma ilustração do uso de um contador, considere o problema do controle para uma máquina que necessita receber 6 itens ao longo do percurso para serem empacotados em uma caixa, e em seguida 12 itens ao longo de outro percurso para serem empacotados em outra caixa. A Figura 21.26 mostra o programa que poderia ser usado. Ele envolve dois contadores, um ajustado para a contagem 6 e o outro,

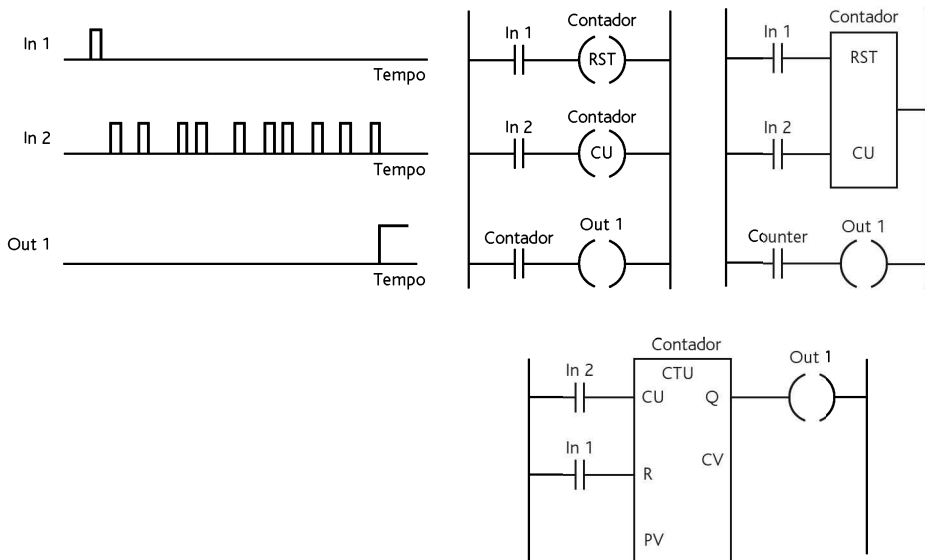


Figura 21.25 As entradas e saída de um contador e algumas formas de representar o mesmo programa.

para a 12. Os contatos da Entrada 1 são fechados momentaneamente para iniciar o ciclo de contagem, resetando os contadores. Os contatos na Entrada 2 poderiam ser ativados por uma chave fim de curso que é acionada a cada vez que um item passa na junção dos dois caminhos. O Contador 1 conta 6 itens e então fecha o seu contato. Isto ativa a saída, que pode ser um solenoide usado para ativar uma aleta que fecha um caminho e abre o outro. O Contador 1 também tem contatos que fecham e habilitam o Contador 2 para iniciar a contagem. Quando este conta 12 itens, reseta os dois contadores e abre os contatos do Contador 1, desativando a saída, e os itens não mais irão para a caixa que contém 12 itens.

21.9 Registradores de deslocamento

Relés internos podem ser agrupados juntos para formar um registrador que pode fornecer uma área de armazenamento para uma sequência de bits individuais. Um registrador de 4 bits pode ser formado usando-se quatro relés internos, e um de oito 8, usando-se oito. O termo registrador de deslocamento é usado porque os bits podem ser deslocados em 1 bit quando houver uma entrada adequada no registrador. Por exemplo, com um registrador de 8 bits, podemos ter inicialmente:

1	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

Em seguida há uma entrada de um pulso de deslocamento 0:

0	→	0	1	0	1	1	0	1	0	→	1
0	1	0	1	1	0	1	0				

em que todos os bits se deslocam ao longo de uma posição e o último bit “transborda”.

O agrupamento de registradores auxiliares para formar um registrador de deslocamento é feito automaticamente por um CLP quando a função registrador de deslocamento é selecionada no painel de controle. No CLP Mitsubishi, isto é feito usando-se a função de programação SFT (*shift*) junto ao número do relé auxiliar que é o primeiro no arranjo de registradores. Isto faz com que o bloco de relés, que começa a partir de um número inicial, seja reservado para o registrador de deslocamento. Portanto, se selecionarmos M140 para ser o primeiro relé, então o registrador de deslocamento consistirá de M140, M141, M142, M143, M144, M145, M146 e M147.

Os registradores de deslocamento têm três entradas: uma para carregar o dado no primeiro elemento (OUT), uma como o comando de deslocamento (SFT) e uma para reset (RST). Em OUT, é carregado um nível lógico 0 ou 1 no primeiro elemento do registrador de deslocamento. Em SFT, um pulso move o conteúdo do registrador 1 bit de cada vez, e o bit final que “transborda” é perdido. Em RST, um pulso do fechamento de um contato reseta o conteúdo do registrador preenchendo-o com zeros.

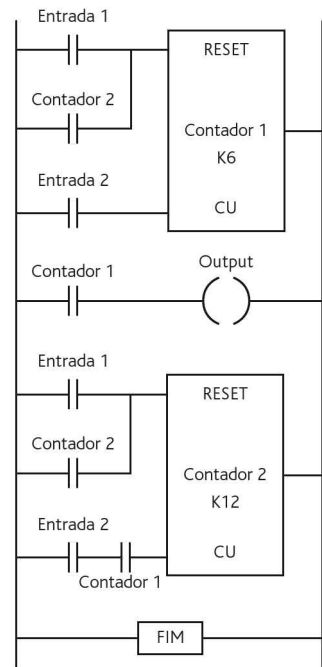


Figura 21.26 Contador.

A Figura 21.27 fornece um diagrama ladder que envolve um registrador de deslocamento quando a notação Mitsubishi é usada; entretanto, o princípio é o mesmo com outros fabricantes. M140 foi designado como o primeiro relé do registrador. Quando X400 é ligado, um nível lógico 1 é acrescentado ao primeiro elemento do registrador de deslocamento, ou seja, M140. Agora temos o registrador com 10000000. O circuito mostra que cada elemento do registrador de deslocamento foi conectado como um contato no circuito. Assim, o contato M140 fecha e Y430 é ligada. Quando o contato X401 é fechado, então os bits no registrador são deslocados ao longo do registrador em uma posição resultando em 11000000, e um 1 é deslocado para dentro de registrador porque X400 ainda está ligado. O contato M141 fecha e Y430 é ligada. Conforme cada bit é deslocado, as saídas são energizadas. Os registradores de deslocamento podem ser usados para eventos em sequência.

21.10 Controles mestre e de salto

Um bloco inteiro de saídas pode ser desligado ou ligado simultaneamente usando-se os mesmos contatos de relé interno em cada ramo de saída, de modo que a operação de ligar ou desligar afete cada uma nos ramos. Uma forma alternativa de programação para obter o mesmo efeito é usar um **relé mestre**. A Figura 21.28 ilustra o seu uso. Podemos considerá-lo como controlando a alimentação de parte de um trilho vertical no diagrama ladder. Quando há uma entrada que fecha os contatos da Entrada 1, o relé mestre MC1 é ativado e o bloco de ramos do programa controlado por este relé é energizado. A extremidade final de uma seção controlado por um relé

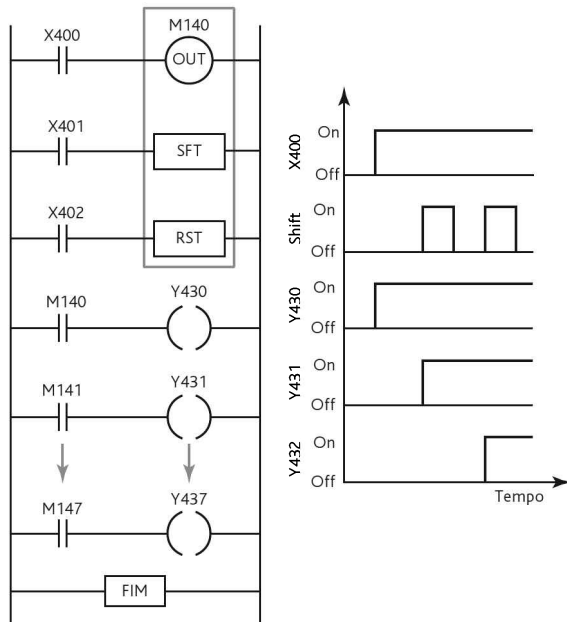


Figura 21.27 Registrador de deslocamento.

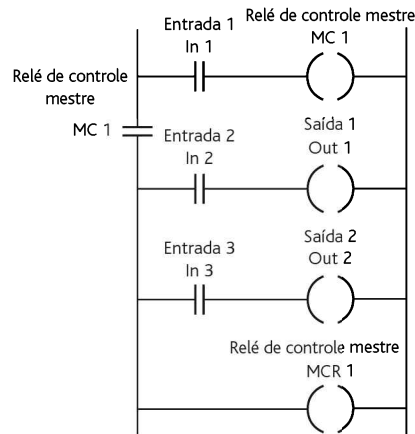


Figura 21.28 Relé de controle mestre.

mestre é indicada pelo reset (MCR). Este é um ramo do programa no qual se houver a Entrada 1, então o ramo após MC1 é um caminho controlado por MC1; caso contrário, o programa continua após o resete ignorando o ramo.

Em um CLP Mitsubishi, um relé interno pode ser designado como um relé de controle mestre através de uma programação adequada. Portanto, para programar um relé interno M100 como um relé de controle mestre, a instrução do programa é:

```
MC M100
```

Para indicar a extremidade final da seção controlada pelo relé de controle mestre, a instrução do programa é:

```
MCR M100
```

21.10.1 Saltos

Uma função que normalmente existe nos CLPs é a função de salto condicional. Esta função habilita que programas sejam projetados de modo que se uma determinada condição existe, então uma seção do programa é saltada. A Figura 21.29 ilustra isto em um fluxograma e com uma seção de um programa ladder. Em seguida à seção do programa, A, o ramo do programa existente é a Entrada 1 e o relé de salto condicional CJP. Se Entrada 1 ocorrer, então o programa salta o ramo com a extremidade final da bobina do relé de salto EJP e continua com esta seção do programa rotulada de C, caso contrário ele continua com os ramos do programa rotulados de B.

21.11 Manipulação de dados

Com exceção do registrador de deslocamento, as partes anteriores deste capítulo abordaram a manipulação de bits individuais de informação, como, por exemplo, uma chave fechada ou não. Entretanto, existem algumas tarefas de controle onde é útil lidar com grupos de bits, como por exemplo, um bloco de oito entradas com o qual se faz uma operação como uma palavra de dados. Esta situação pode ocorrer

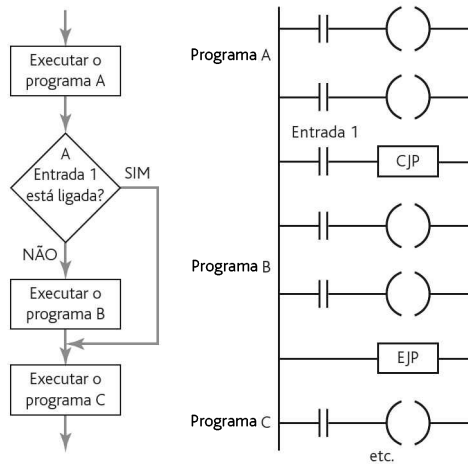


Figura 21.29 Salto.

quando um sensor fornecer um sinal analógico que é convertido para, por exemplo, uma palavra de 8 bits antes de entrar no CLP.

As operações sobre palavras de dados que podem ser executadas em um CLP normalmente incluem:

- 1 Movimentação de dados.
- 2 Comparação de magnitudes de dados, ou seja, maior que, igual a ou menor que.
- 3 Operações aritméticas como adição e subtração.
- 4 Conversões entre decimal codificado em binário (BCD), binário e octal.

Conforme discutido antes, os bits individuais são armazenados em posições de memória especificadas por endereços únicos. Por exemplo, no CLP Mitsubishi, os endereços de memórias de entrada são precedidos por A, os de saída por Y, os temporizadores por T, os relés auxiliares por M, etc. As instruções de dados também necessitam de endereços de memória, e as posições na memória do CLP alocadas para dados são denominadas **registradores de dados**. Cada registrador de dados pode armazenar uma palavra binária de, geralmente, 8 ou 16 bits em endereços como D0, D1, D2, etc. Uma palavra de 8 bits significa uma quantidade especificada com precisão de 1 em 256; uma palavra de 16 bits tem uma precisão de 1 em 65.536.

Cada instrução tem que especificar a forma da operação, a fonte de dados usada em termos de registrador de dados e o registrador de dados de destino.

21.11.1 Movimentação de dados

Para a movimentação de dados, o comando contém a instrução de movimentação de dados, o endereço da fonte de dados e o endereço de destino. Portanto, o ramo do ladder poderia ser da forma mostrada na Figura 21.30.

Esta transferência de dados pode mover uma constante para um registrador de dados, um valor de tempo ou contagem para um registrador de dados, dados de um registrador

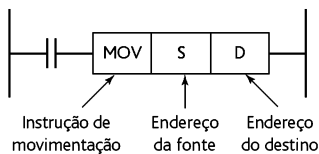


Figura 21.30 Movimentação de dados.

de dados para um temporizador ou contador, dados de um registrador de dados para uma saída, dados de entrada para um registrador de dados, etc.

21.11.2 Comparação de dados

Geralmente os CLPs podem realizar comparações entre dados como *menor que* (geralmente indicada por $<$ ou LES), *igual a* ($=$ ou EQU), *menor que ou igual a* (\leq ou \leq ou LEQ), *maior que* ($>$ ou GRT), *maior que ou igual a* (\geq , \geq ou GEQ) e *diferente de* (\neq ou \diamond ou NEQ). Para comparar dados, o comando de comparação contém a instrução de comparação, o endereço da fonte do dado e o endereço de destino. Portanto, para comparar o dado no registrador de dados D1 para saber se ele é maior que o dado no registrador de dados D2, o ramo do programa ladder seria da forma mostrada na Figura 21.31.

Esta comparação pode ser usada quando os sinais de dois sensores são comparados pelo CLP antes que uma ação seja tomada. Por exemplo, pode ser necessário que um alarme seja acionado se um sensor indicar uma temperatura acima de 80°C e permanecer acionado até que a temperatura caia abaixo de 70°C . A Figura 21.32 mostra o programa ladder que poderia ser usado. O dado da temperatura de entrada é inserido no endereço fonte e o endereço destino contém o valor desejado. Quando a temperatura sobe para 80°C , ou mais, o valor do dado no endereço fonte se torna \geq ao valor do endereço destino e há uma saída para o alarme que memoriza a entrada. Quando a temperatura cai para 70°C , ou menos, o valor do dado no endereço fonte se torna \leq ao valor no endereço destino e há uma saída para o relé, que então abre seus contatos, desligando o alarme.

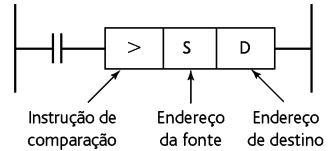


Figura 21.31 Comparação de dados.

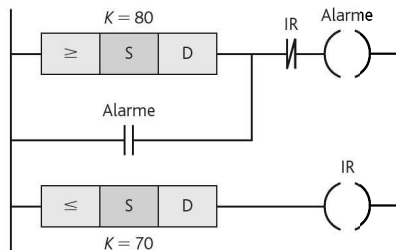


Figura 21.32 Alarme de temperatura.

21.11.3 Operações aritméticas

Alguns CLPs podem executar apenas as operações aritméticas de soma e subtração, já outros têm mais funções aritméticas. A instrução para somar ou subtrair expressa geralmente o registro que contém o endereço do valor do primeiro operando a ser somado ou subtraído, o endereço do segundo operando e o registrador onde o resultado deve ser armazenado. A Figura 21.33 mostra a forma usada para o símbolo ladder para a adição em um CLP OMRON.

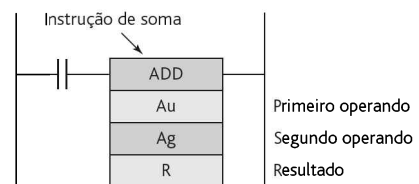


Figura 21.33 Soma de dados.

A adição ou subtração pode ser usada para alterar o valor de entrada de algum sensor, podendo ser uma correção ou *offset*, ou pode ainda alterar os valores ajustados em temporizadores ou contadores.

21.11.4 Conversões de código

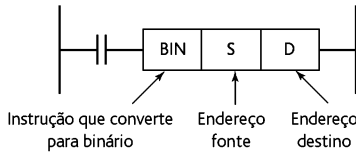


Figura 21.34 BCD para binário.

Todas as operações internas na CPU de um CLP são realizadas usando-se números binários. Portanto, quando a entrada é um sinal que é decimal, é usada a conversão para BCD. Da mesma forma, onde é necessário uma saída decimal, tem que ser feita a conversão para decimal. Estas conversões são realizadas pela maioria dos CLPs. Por exemplo, no CLP Mitsubishi, o ramo do ladder que converte de BCD para binário

é da forma mostrada na Figura 21.34. Os dados no endereço fonte estão em BCD e são convertidos para binário e colocados no endereço destino.

21.12 Entrada/saída analógica

Muitos sensores geram sinais analógicos e muitos atuadores necessitam de sinais analógicos. Portanto, alguns CLPs podem ter um módulo conversor analógico-digital instalado nos canais de entrada e um conversor digital-analógico instalado nos canais de saída. Um exemplo de onde este item pode ser usado é no controle de velocidade de um motor, de modo que ele atinja a velocidade de estado estacionário a uma taxa constante (Figura 21.35). A entrada é uma chave *on/off* para operação de partida. Com a chave de partida aberta, o registrador de dados é carregado com zero. Portanto, a saída do controlador é zero e, conseqüentemente, a velocidade do motor é zero. Ao fechar a chave de partida, as saídas DAC e registrador de dados são ativadas. Cada vez que o programa executa um ciclo, o registrador de dados é incrementado de 1, o sinal analógico é aumentado e, portanto, a velocidade do motor. A velocidade máxima é atingida quando a saída do registrador de dados contiver a palavra 11111111. A função temporizador do CLP pode ser usada para incorporar um atraso entre cada um dos sinais de bit de saída.

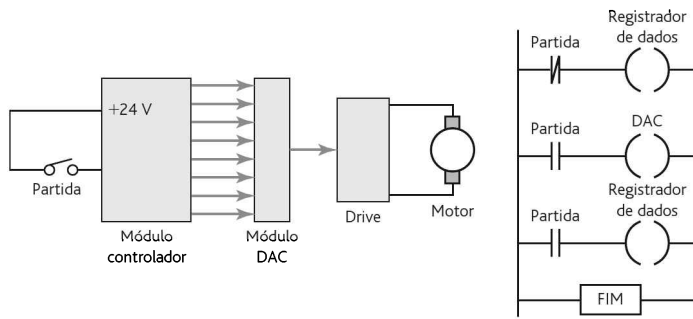


Figura 21.35 Rampa de velocidade de um motor.

Um CLP equipado com canais de entrada analógica pode ser usado para executar uma função de controle contínuo, ou seja, um controle PID (veja a Seção 15.7). Portanto, por exemplo, para realizar o controle proporcional a uma entrada analógica, os seguintes passos podem ser seguidos:

- 1 Converta a saída do sensor em um sinal digital.
- 2 Compare a saída atual do sensor com o valor do sensor desejado, ou seja, a *set point*, e obtenha a diferença. Esta diferença é o erro.
- 3 Multiplique o erro pela constante proporcional K_p .
- 4 Mova este resultado para a saída do DAC e use o resultado como o sinal de correção do atuador.

Um exemplo de onde esta ação de controle poderia ser usada é com um controlador de temperatura. A Figura 21.36 mostra uma possibilidade. A entrada poderia ser de um termopar que, após uma amplificação, é inserida no ADC conectado a um CLP. Este é programado para gerar uma saída proporcional ao erro entre a entrada do sensor e a temperatura desejada. A palavra de saída é então enviada ao DAC conectado ao atuador, que é um aquecedor, para reduzir o erro.

No programa ladder mostrado, o ramo 0 lê o ADC e armazena o valor de temperatura no registrador de dados DR1. No ramo 1, o registrador de dados DR2 é usado para armazenar o *set point* de temperatura. O ramo 2 usa a função de subtração para subtrair os valores dos registradores de dados DR1 e DR2 e armazena o resultado no registrador de dados DR3, ou seja, este registrador guarda o valor do erro. No ramo 3, é usada a função de multiplicação, que neste caso multiplica o valor do registrador de dados DR3 por um ganho proporcional de 4. O ramo 4 usa um relé interno que pode ser programado para desligar DR3 se ele se tornar um valor

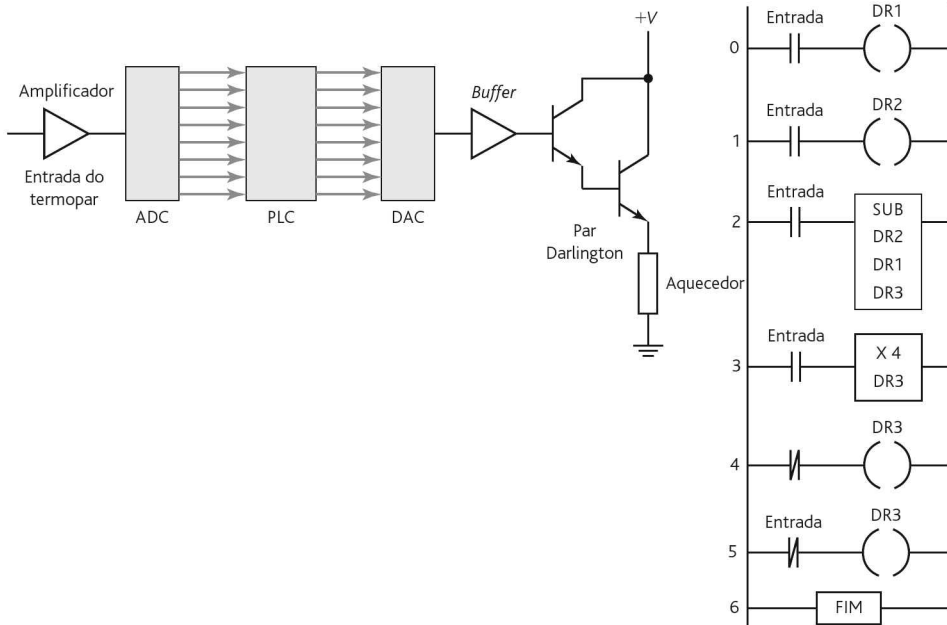


Figura 21.36 Controle proporcional de temperatura.

negativo. No ramo 5, o registrador de dados DR3 é resetado para zero quando a entrada é desligada. Alguns CLPs têm módulos que podem ser acrescentados os quais permitem que este controle seja feito mais facilmente sem a necessidade de escrever listas de instruções, como neste exemplo.

Resumo

Um **controlador lógico programável (CLP)** é um equipamento eletrônico que usa uma memória programável para armazenar instruções e implementar funções como lógica, sequenciamento, temporização, contagem e aritméticas para o controle de máquinas e processos, e foi especialmente projetado para tornar fácil a programação.

Um CLP executa o programa continuamente, fazendo atualizações em resposta aos sinais de entrada. Cada *loop* é denominado **ciclo**. O tipo de programação normalmente usada nos CLPs é a **ladder**. Cada tarefa (lógica de controle) do programa é especificada como os degraus (ramos) de uma escada (*ladder*). Existe uma forma alternativa de inserir um programa que é denominada **lista de instruções**. O programa ladder é traduzido em uma lista de instruções. Esta lista consiste de uma série de instruções em que cada uma usa uma linha em separado. Uma instrução consiste de um operador seguido por um ou mais operandos, ou seja, os dados sujeitos à operação especificada pelo operador.

Um **circuito latch** é aquele que, após ser energizado, mantém o seu estado até que uma outra entrada seja recebida. O termo **relé interno, relé auxiliar** ou **marcador** é usado para o que pode ser considerado como um relé interno em um CLP, com comportamento de relé com contatos associados. Os **temporizadores** podem ser considerados como tendo comportamento de relés com bobinas que, quando energizadas, resultam no fechamento ou abertura de contatos após algum tempo definido ou como um bloco de atraso em um ramo que atrasa o sinal que chega na saída. Os **contadores** são usados para contar um número especificado de operações de um contato, sendo considerado uma bobina de saída para contar pulsos de entrada com uma bobina para resetar o contador, e os contatos associados do contador são usados em outros ramos ou como um bloco intermediário em um ramo do qual emanam sinais quando o contador é acessado. Um **registrador de deslocamento** é um número de relés internos que são agrupados para formar um registrador com uma sequência de bits individuais em série. Um relé mestre permite que um bloco inteiro de saídas seja ligado ou desligado simultaneamente. A função de **salto condicional** permite que uma seção de um programa seja saltada se ocorrer uma determinada condição. As operações que podem ser executadas com palavras de dados incluem movimentação de dados, comparação de magnitudes de dados, operações aritméticas e conversões entre decimal codificado em binário (BCD), binário e octal.

Problemas

- 21.1 Quais são as funções lógicas usada para o caso de chaves (a) em série, (b) em paralelo?

- 21.2 Desenhe os ramos de um ladder para representar:
- (a) Duas chaves normalmente abertas devem ser fechadas para um motor operar.
 - (b) Uma, dentre duas chaves normalmente abertas, tem que ser fechada para que uma bobina seja energizada e opere um atuador.
 - (c) Um motor é ligado ao se pressionar uma chave de partida do tipo *push-button* com mola e retorno, e o motor permanece ligado até que uma outra chave do mesmo tipo seja pressionada.

21.3 Escreva as instruções do programa que corresponde ao programa *latch* mostrado na Figura 21.37.

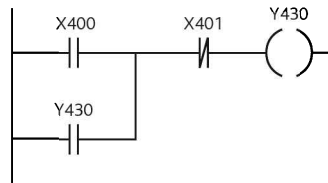


Figura 21.37 Problema 21.3.

21.4 Escreva as instruções do programa do diagrama da Figura 21.38 e diga como a saída varia com o tempo.

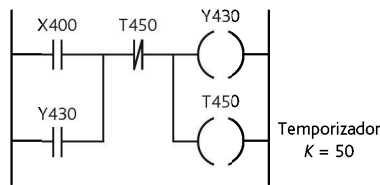


Figura 21.38 Problema 21.4.

21.5 Escreva as instruções do programa que correspondem ao diagrama mostrado na Figura 21.39 e determine os resultados das entradas do CLP.

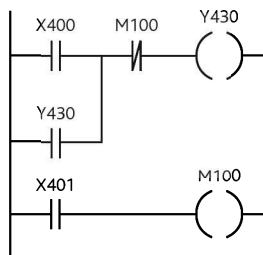


Figura 21.39 Problema 21.5.

21.6 Projete um circuito de temporização que liga uma saída por 1 s, desliga por 20 s, liga por 1s, desliga por 20 s, e assim por diante.

- 21.7 Projete um circuito de temporização que liga uma saída por 10 s, desligando-a em seguida.
- 21.8 Projete um circuito que pode ser usado para dar partida em um motor e, após um atraso de 100 s, dar partida em uma bomba. Quando o motor é desligado, deve haver um atraso de 10 s antes da bomba ser desligada.
- 21.9 Projete um circuito que possa ser usado com uma máquina da lavar roupa doméstica para ligar uma bomba que bombeia água por 100 s para dentro da máquina. Em seguida, a bomba é desligada e um aquecedor é ligado por 50 s para esquentar a água. O aquecedor é então desligado e uma outra bomba, que retira a água da máquina, é ligada por 100 s.
- 21.10 Projete um circuito que possa ser usado com uma correia transportadora que move um item para uma estação de trabalho. A presença do item na estação de trabalho é detectada pela interrupção de um feixe de luz de um fotossensor. O item deve permanecer parado por 100 s para que uma operação seja executada antes de se ligar e desligar o transportador. O motor da correia é acionado por uma chave de partida normalmente aberta e parado por uma chave normalmente fechada.
- 21.11 Qual seria o padrão de temporização para o registrador de deslocamento da Figura 21.27 mudar se o dado da entrada X400 tem a forma mostrada na Figura 21.40?

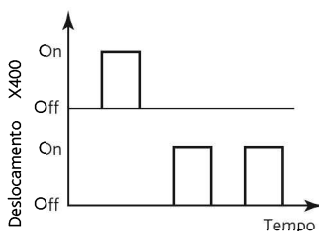


Figura 21.40 Problema 21.11.

- 21.12 Explique como um CLP pode ser usado para manipular uma entrada analógica.
- 21.13 Projete um sistema, usando um CLP, que possa ser usado para controlar o movimento de um pistão em um cilindro de modo que quando uma chave é momentaneamente pressionada, o pistão se move em uma direção, e quando uma segunda chave é momentaneamente pressionada, o pistão se move na outra direção. Sugestão: você pode considerar o uso de uma válvula 4/2 controlada por solenoide.
- 21.14 Projete um sistema, usando um CLP, que possa ser usado para controlar o movimento de um pistão em um cilindro usando uma válvula piloto 4/2 controlada por solenoide. O pistão deve se mover em uma direção quando um sensor de proximidade em uma extremidade do percurso fecha os contatos, e na outra direção quando um sensor de proximidade na outra extremidade do percurso indica a chegada dele.

22

Sistemas de Comunicação

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Descrever os sistemas de controle centralizado, hierárquico e distribuído, as configurações de rede, e métodos de transmissão de dados e protocolos usados.
- Descrever o modelo de comunicação denominado Interconexão de Sistemas Abertos (OSI).
- Descrever as interfaces de comunicação normalmente usadas: RS-232, IEEE 488, *loop* de corrente de 20 mA, barramentos I²C e CAN.

22.1 Comunicações digitais

Um **barramento externo** é um conjunto de linhas de sinais que interconectam microprocessadores, microcontroladores, computadores e CLPs e também conectam estes com equipamentos periféricos. Portanto, um computador necessita ter um barramento que o conecte a uma impressora se a sua saída for destinada a impressão. Os sistemas multiprocessadores são bastante comuns. Por exemplo, em um carro normalmente existem vários microprocessadores, cada um controlando uma parte do sistema, como, por exemplo, o gerenciamento do motor, frenagem e painel de instrumentos, sendo necessária a comunicação entre eles. Em uma planta automatizada não existe apenas a necessidade dos dados trafegarem entre controladores lógicos programáveis, displays, sensores e atuadores e de se possibilitar a inserção de dados e programas pelo operador, mas também pode ocorrer a comunicação de dados com outros computadores. Por exemplo, pode existir a necessidade da conexão de um CLP a um sistema de controle que envolve outros CLPs e computadores. A manufatura integrada por computador (CIM – *Computer Integrated Manufacturing*) é um exemplo de uma rede grande que pode envolver um grande número de máquinas interconectadas. Este capítulo aborda como a comunicação de dados entre computadores pode ocorrer, seja

apenas uma simples comunicação máquina a máquina ou uma grande rede envolvendo um grande número de máquinas interconectadas, e as formas de interfaces de comunicação padrão.

22.2 Controle centralizado, hierárquico e distribuído

O **controle computacional centralizado** envolve o uso de um computador central para controlar um planta inteira. O problema deste tipo de controle é que uma falha no computador resulta na perda do controle de toda a planta. Isto pode ser evitado com o uso de dois sistema de computador. Se um comutador falhar, o outro assume o controle. Estes sistemas centralizados foram comuns nos anos de 1960 e 1970. O desenvolvimento dos microprocessadores e a redução de custos dos computadores fizeram com que os sistemas multicomputador se tornassem cada vez mais comuns e levaram ao desenvolvimento dos sistemas hierárquico e distribuído.

No **sistema hierárquico**, há uma hierarquia de computadores de acordo com a tarefa executada por eles. Os computadores que manipulam as tarefas mais rotineiras são supervisionados por comutadores que fazem o papel de tomar decisões. Por exemplo, os computadores que são usados para o controle digital direto dos sistemas são subservientes a um computador que realiza o controle supervisorio do sistema inteiro. O trabalho é dividido entre os computadores de acordo com a função envolvida. Existem computadores especializados que apenas recebem algumas informações e outros que recebem outras.

No **sistema distribuído**, cada sistema de computador executa essencialmente tarefas similares a todos os outros sistemas de computador. No caso de um falhar ou de um computador estar sobrecarregado, o trabalho pode ser transferido para outros computadores. O trabalho é dividido entre todos os computadores e não destinado a computadores específicos de acordo com a função envolvida. Não há computadores especializados. Cada computador precisa acessar todas as informações no sistema.

Na maioria dos sistemas modernos, há geralmente uma mistura dos sistemas hierárquico e distribuído. Por exemplo, o trabalho de medir e atuar pode ser distribuído entre os microcontroladores/computadores que estão interconectados e fornecem os dados para a planta. Estes podem ser supervisionados por um computador usado para o controle digital direto ou o sequenciamento, que por sua vez pode ser supervisionado por um usado para controle supervisorio da planta como um todo. Os níveis típicos neste esquema são:

- Nível 1 Medição e atuação
- Nível 2 Controle digital direto e sequencial
- Nível 3 Controle supervisorio
- Nível 4 Controle de gerenciamento e projeto

Os sistemas distribuído/hierárquico têm a vantagem de permitir que tarefas de varredura de medições e condicionamento de sinal em sistemas de controle sejam partilhadas entre microprocessadores. Isto pode envolver um grande número de sinais com uma alta frequência de varredura. Se forem necessários mais *loops* de medição, basta aumentar a capacidade do sistema acrescentando-se microprocessadores. As unidades podem estar bastante dispersas, estando localizada próximo à fonte das medições. A falha de uma unidade não resulta em falha do sistema inteiro.

22.2.1 Transmissão de dados em paralelo e em série

A transmissão de dados pode ser de forma paralela ou serial.

1 *Transmissão de dados em paralelo*

Dentro dos computadores, a transmissão de dados geralmente é feita em **vias paralelas de dados**. Os barramentos de dados em paralelo transmitem 8, 16 ou 32 bits simultaneamente, tendo um fio do barramento em separado para cada bit de dado e sinal de controle. Portanto, se existem oito bits de dados a serem transmitidos, como, por exemplo, 11000111, então são necessários oito fios em paralelo. As linhas de *handshaking* (veja a Seção 20.3.2) também são necessárias, sendo o *handshaking* feito para cada caractere transmitido e estas linhas usadas para indicar que o dado está disponível para transmissão e que o terminal de recepção está pronto para receber. A transmissão de dados em paralelo permite uma alta taxa de transferência de dados, mas é cara por causa do cabeamento e dos circuitos de interface necessários. Ela é usada normalmente apenas em distâncias curtas ou onde é essencial uma alta taxa de transferência.

2 *Transmissão de dados em série*

Envolve a transmissão de dados que, em conjunto com os sinais de controle, são enviados bit a bit em sequência ao longo de uma única linha. Apenas dois condutores são necessários, um para transmitir e outro para receber os dados. Como os bits de uma palavra são transmitidos sequencialmente e não simultaneamente, a taxa de transferência de dados é considerada menor do que na transmissão em paralelo. Entretanto, é mais barato, visto que são necessários bem menos condutores. Por exemplo, quando é usado um determinado número de microcontroladores em um carro, as conexões entre eles são feitas para transmissão serial. Sem o uso da transmissão serial, o número de fios envolvidos seria considerável. Em geral, a transmissão de dados em série é usada por todos os dispositivos, porém em conexões de periféricos mais próximos.

Considere o problema de enviar uma sequência de caracteres ao longo de um enlace serial. O receptor precisa saber onde começam e terminam os caracteres. A transmissão de dados em série pode ser assíncrona ou síncrona. A **transmissão assíncrona** implica que os computadores transmissor e receptor não são sincronizados, ou seja, cada um tem o seu próprio sinal de clock. O tempo entre os caracteres transmitidos é arbitrário. Cada caractere transmitido em um enlace é precedido por um bit de início (*start bit*) para indicar ao receptor o início do caractere, que é seguido de um bit de fim (*stop bit*) para indicar o término do caractere. Este método tem a desvantagem da necessidade de transmitir bits extra juntamente com cada caractere, reduzindo assim a eficiência da linha para a transmissão de dados. Na **transmissão síncrona**, não há a necessidade de bits de início e de fim, visto que o transmissor e o receptor têm um sinal de clock comum e, portanto, os caracteres iniciam e terminam sempre no mesmo instante em cada ciclo.

A taxa de transmissão de dados é medida em bits por segundo. Se um grupo de n bits forem um único símbolo a ser transmitido e este símbolo tem uma duração de T segundos, então a taxa de dados da transmissão é n/T . A unidade usada é o **baud**. A taxa em **baud** é o mesmo que o número de bits por segundo transmitidos se cada caractere for representado por apenas um símbolo. Portanto, um sistema que não usa bits de início e de fim tem uma taxa em baud igual à taxa em bits, mas este não é o caso quando estes bits existem.

22.2.2 Modos de comunicação de dados em série

A transmissão de dados em série é realizada em um dos três modos a seguir:

1 *Simplex*

A transmissão é possível apenas em uma direção, do dispositivo A para o B, em que o dispositivo B não é capaz de transmitir de volta para A (Figura 22.1(a)). Podemos pensar em uma conexão entre dois dispositivos semelhante a uma rua de mão única. Este método é usado geralmente apenas para transmissão para dispositivos, como impressoras, que nunca transmitem informações.

2 *Half-duplex*

Os dados são transmitidos em um sentido de cada vez, ou seja, o sentido pode ser mudado (Figura 22.1(b)). Os terminais em cada extremidade do enlace podem ser comutados de transmissor para receptor. Portanto, o dispositivo A pode transmitir para o dispositivo B e este para o A, mas não ao mesmo tempo. Esta situação é análoga a uma rodovia de mão dupla que está em reparo e com tráfego em meia pista. O controle do tráfego permite a passagem dos carros em um sentido durante um tempo, interrompe este fluxo e, em seguida, permite a passagem dos carros que vão em sentido oposto. Um outro exemplo deste modo é a comunicação de rádio de faixa do cidadão (*Citizens Band – CB*); uma pessoa pode receber ou falar, mas não simultaneamente.

3 *Full-duplex*

Os dados podem ser transmitidos simultaneamente nos dois sentidos entre os dispositivos A e B (Figura 22.1(c)). É semelhante a uma rodovia em que o tráfego ocorre normalmente nos dois sentidos simultaneamente. O sistema telefônico é um exemplo do modo *full-duplex* em que uma pessoa pode falar e receber em qualquer instante.

22.3 Redes

O termo rede (*network*) é usado para um sistema que permite que dois ou mais computadores/microprocessadores sejam interconectados para troca de dados. A forma lógica das conexões é denominada **topologia** da rede. O termo **nó** é usado para um ponto de uma rede em que uma ou mais linhas de comunicação terminam ou uma unidade é conectada nas linhas de comunicação. Os tipos normalmente usados são:

1 *Barramento de dados*

Este tem um barramento linear (Figura 22.2(a)) no qual todas as estações são conectadas. Este sistema é usado frequentemente por um grupo de terminais

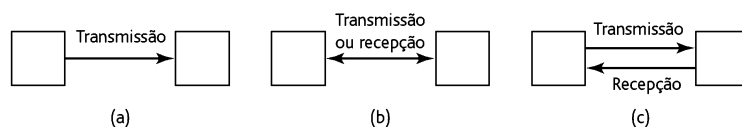


Figura 22.1 Modos de comunicação.

multiponto. Este é o método geralmente preferido para distâncias entre nós maiores do que 100 m.

2 Estrela

Este tem canais dedicados entre cada estação e um *hub* de comutação central (Figura 22.2(b)) através do qual todas as comunicações têm que passar. Este é o tipo de rede usado em sistemas de telefonia (troca automática de ramais privados [PBXs – *Private Branch Exchanges*]) em muitas companhias em que todas as linhas passam por uma central de ramais. Este sistema também é usado com frequência para conectar terminais remotos e locais a um computador central de grande porte (*mainframe*). O principal problema que há neste sistema é que se o *hub* central falhar, todo o sistema fica inoperante.

3 Hierárquico ou árvore

Este consiste de uma série de ramos que convergem indiretamente para um ponto no topo da árvore (Figura 22.2(c)). Neste sistema há apenas um percurso entre duas estações quaisquer. Esta configuração pode ser formada a partir de uma quantidade de sistemas de barramento de dados conectados. Assim como o método de barramento, ele é usado frequentemente para distâncias entre nós maiores do que 100 m.

4 Anel

Este é um método muito popular para redes locais em que cada estação é conectada formando um anel (Figura 22.2(d)). A distância entre os nós são geralmente menores do que 100 m. Os dados colocados no sistema em anel continuam circulando até que algum sistema os remova. Os dados estão disponíveis para todas as estações.

5 Malha

Este método (Figura 22.2(e)) não apresenta uma forma padrão de conexão entre as estações, e há muitos percursos múltiplos de dados entre elas.

O termo **rede local** (LAN – *Local Area Network*) é usado para uma rede localizada em uma área geográfica, como a de edifício ou um grupo de edifícios em um local. A topologia usada é normalmente a de barramento, estrela ou anel. Uma **rede de longa distância** (WAN – *Wide Area Network*) é aquela que interconecta computadores, terminais e redes locais em distâncias de nível nacional e internacional.

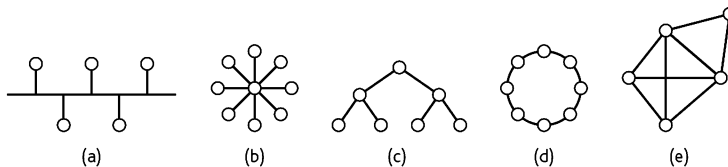


Figura 22.2 Topologias de rede: (a) barramento de dados, (b) estrela, (c) hierárquico, (d) anel, (e) malha.

22.3.1 Controle de acesso à rede

Os métodos de controle de acesso são necessários em uma rede para garantir que apenas um usuário da rede seja capaz de transmitir em qualquer momento. A seguir abordamos os métodos usados.

Os dois métodos normalmente usados em redes locais baseadas em anel são:

1 *Passagem de token*

Com este método, um *token* (ficha), um padrão de bit especial, circula na rede. Quando uma estação deseja transmitir, espera até a recepção do *token*, então transmite o dado com o *token* anexado à extremidade dele. Uma outra estação que deseja transmitir remove o *token* do pacote de dados e transmite o seu próprio dado com o *token* anexado à extremidade dele.

2 *Passagem de slot*

Este método envolve *slots* vazios circulando na rede. Quando uma estação deseja transmitir dados, ela deposita o dado no *slot* vazio que chega.

Na redes de barramento ou árvore, um método frequentemente usado é:

3 *Acesso múltiplo com detecção de portadora e de colisão (CSMA/CD)*

Este método é geralmente identificado com a **Ethernet LAN bus**. Com o método CSMA/CD, as estações têm que “ouvir” as outras transmissões antes de transmitir, sendo que cada estação pode ser capaz de obter o controle da rede e transmitir, por isto o termo acesso múltiplo. Se não for detectada nenhuma atividade, então a transmissão pode ocorrer. Se houver atividade, então o sistema tem que esperar até detectar a ausência de atividade. Apesar de “ouvirem” antes de transmitir, é possível que dois ou mais sistemas iniciem uma transmissão ao mesmo tempo. Se esta situação for detectada, as estações param de tentar a transmissão e esperam por um tempo aleatório antes de tentar retransmitir.

22.3.2 *Broadband* e banda base

O termo **transmissão *broadband*** é usado para uma rede na qual a informação é modulada em uma portadora na frequência de rádio que passa pelo meio de transmissão, como, por exemplo, um cabo coaxial. Tipicamente, a topologia de uma rede local é um barramento com ramificações. A transmissão *broadband* permite o uso simultâneo de portadoras na frequência de rádio moduladas, oferecendo assim uma capacidade multicanal. O termo **transmissão em banda base** é usado quando uma informação digital é passada diretamente através do meio de transmissão. As redes de transmissão em banda base podem suportar apenas um sinal de informação de cada vez. Uma LAN pode ser uma rede de *broadband* ou de banda base.

22.4 Protocolos

Os dados a serem transmitidos contêm dois tipos de informação. Uma é o dado que um computador deseja enviar para outro, a outra é a informação denominada **protocolo** e é usada pela interface entre um computador e a rede para controlar a transferência de dados entre a rede e o computador. Um protocolo é um conjunto formal de regras que governam o formato, temporização, sequência, controle de acesso e controle de erro dos dados. Os três elementos de um protocolo são:

- 1 *Sintaxe*, que define o formato, codificação e níveis de sinais dos dados.
- 2 *Semântica*, que trata da sincronização, controle e manipulação de erro sobre os dados.
- 3 *Temporização*, que trata da sequência dos dados e a escolha da taxa de dados.

Quando um remetente comunica com um receptor, ambos empregam o mesmo protocolo, como, por exemplo, dois microcontroladores com dados a serem transmitidos serialmente entre eles. Na comunicação *simplex*, o bloco de dados pode ser apenas enviado do remetente para o receptor. Entretanto, na *half-duplex*, cada bloco de dados transmitido, sendo válido, tem que ser reconhecido (ACK) pelo receptor antes do próximo bloco de dados poder ser enviado (Figura 22.3(a)); se for inválido, é enviado um sinal de reconhecimento negativo (NAK). Portanto, um fluxo contínuo de dados não pode ser transmitido. Os **bits de verificação de redundância cíclica (CRC – Cyclic Redundancy Check)** fazem parte de um mecanismo de detecção de erro e são transmitidos imediatamente após um bloco de dados. O dado é transmitido com um número binário e , no transmissor, é dividido por um número; o resto é usado como um código de verificação cíclica. Os dados que chegam no receptor, incluindo o CRC, é dividido pelo mesmo número, e o resto será zero se o sinal não contiver erro. No modo *full-duplex* (Figura 20.3(b)), os dados podem ser enviados e recebidos continuamente.

Em um pacote enviado, há a necessidade de se incluir a informação de protocolo. Por exemplo, na transmissão assíncrona, há caracteres para indicar o início e o fim dos dados. Com a transmissão síncrona e o protocolo Bisync, um bloco de dados é precedido por uma sequência de bits de sincronismo, geralmente o caractere ASCII SYN (Figura 22.4(a)). Os caracteres SYN são usados pelo receptor para conseguir a sincronização de caractere, preparando o receptor para receber grupos

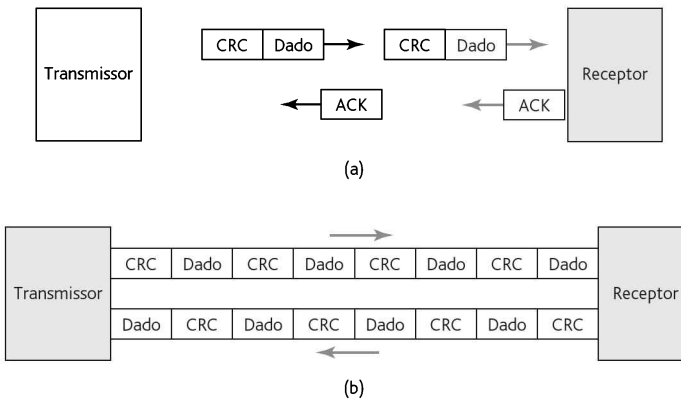


Figura 22.3 Protocolos: (a) *half-duplex*, (b) *full-duplex*.

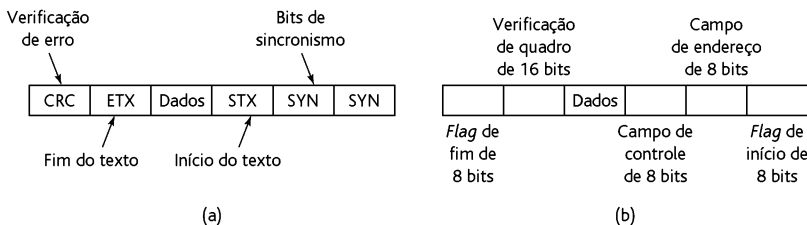


Figura 22.4 Protocolos: (a) Bisync, (b) HDLC.

de dados de 8 bits. O MC6852 da Motorola é um adaptador de dados serial síncrono (SSDA) que foi projetado para uso com os microprocessadores 6800 para fornecer uma interface de comunicação serial síncrona usando o protocolo Bisync. Ele é similar ao adaptador de interface de comunicação serial assíncrona descrito na Seção 20.5. Um outro protocolo usado é o HDLC (*High-level Data Link Control*). Este é um protocolo *full-duplex* com o início e o final da mensagem indicados pelo padrão de bits 01111110. Os campos de endereço e controle seguem o *flag* de início. O campo de endereço identifica o endereço da estação de destino, o campo de controle define se o quadro é supervisorio, informação ou não numerado. Após a mensagem há uma sequência de verificação de quadro de 16 bits que é usada para fornecer um CRC. O 6854 da Motorola é um exemplo de um adaptador de interface serial usando o protocolo HDLC.

22.5 Modelo de comunicação por interconexão de sistemas abertos

Os protocolos de comunicação devem existir em um determinado número de níveis. A organização internacional para padronização (ISO – *International Organization for Standardization*) definiu um sistema de protocolo padrão de sete camadas conhecido como modelo de **Interconexão de Sistemas Abertos** (OSI – *Open Systems Interconnection*). Este modelo é uma estrutura para o desenvolvimento de padrões coordenados. As camadas são:

1 Física

Esta camada descreve os meios para a transmissão de bits entre os componentes físicos da rede. Ela estabelece definições para a *hardware*, como, por exemplo, os tipos de cabos e conectores a serem usados, sincronização na transferência de dados e níveis de sinais. Os sistemas de LAN usados normalmente na camada física são *Ethernet* e *token ring*.

2 Enlace de dados

Esta camada define os protocolos de envio e recebimento de mensagens, detecção e correção de erros e o sequenciamento adequado dos dados transmitidos. Ela cuida do empacotamento dos dados, da inserção e da retirada deles no meio. Também para este nível são definidos os sistemas *Ethernet* e *token ring*.

3 Rede

Esta camada lida com os caminhos de comunicação e o endereçamento, roteamento e controle de mensagens na rede, e portanto cuida para que a mensagem chegue ao destino certo. A camada de rede usa normalmente protocolos como o Internet Protocol (IP) e o protocolo da Novell de troca de pacotes entre redes (IPX – *Internetwork Packet Exchange*)

4 Transporte

Esta fornece um transporte de mensagem fim a fim seguro. Ela se preocupa em estabelecer e manter a conexão entre transmissor e receptor. Os protocolos da camada de transporte normalmente usados são o protocolo de controle de transmissão da Internet (TCP – *Transmission Control Protocol*) e o protocolo da Novell de troca de pacotes sequenciais (SPX – *Sequenced Packet Exchange*).

5 Sessão

Esta camada cuida de estabelecer diálogo entre os processos de aplicação que são conectados juntos através da rede. Ela se responsabiliza pela determinação de quando ligar ou desligar uma comunicação entre duas estações.

6 Apresentação

Esta camada cuida de permitir que os dados codificados transmitidos sejam apresentados de forma adequada para utilização pelo usuário.

7 Aplicação

Esta camada fornece a função de processamento de informação do usuário presente e serviços de aplicações específicas. Ela fornece informações, como transferência de arquivos ou correio eletrônico, que uma estação pode usar para comunicar com outros sistemas na rede.

22.5.1 Padrões de rede

Existem vários padrões de rede, baseados no modelo OSI, que são normalmente usados. Os exemplos a seguir tratam de padrões de redes.

Nos Estados Unidos, a General Motors se deu conta de que a automação dos seus processos de manufatura fez surgir um problema de equipamento fornecidos com uma variedade de protocolos não padronizados. Então a GM desenvolveu um sistema de comunicação padrão para aplicações de automação de fábrica. O padrão é conhecido como **Protocolo de Automação da Manufatura** (MAP – *Manufacturing Automation Protocol*) (Figura 22.5). A escolha de protocolos em camadas diferentes reflete a necessidade dos sistemas de se ajustarem ao ambiente da manufatura. As camadas 1 e 2 são implementadas no *hardware* e os níveis 3 a 7 usam software. Na camada física é usada a transmissão *broadband*. Este método permite que o sistema seja usado para serviços além dos necessários para as comunicações MAP. Para a camada de enlace de dados, o sistema *token* com um barramento é usado com o controle de enlace lógico (LLC – *Logical Link Control*) para implementar funções como verificação de erro, etc. Para as outras camadas, são usados os padrões ISO. Na camada 7, o MAP inclui os serviços de mensagem da manufatura (MMS – *Manufacturing Message Services*), uma aplicação relevante para comunicações de chão de fábrica que define interações entre controladores lógicos programáveis e máquinas de controle numérico ou robôs.

O **Protocolo Técnico e de Escritório** (TOP – *Technical and Office Protocol*) é um padrão desenvolvido pela empresa Boeing (Boeing Computer Services). Ele tem muito em comum com o MAP, mas pode ser implementado a um custo menor porque é um sistema de banda base. Ele difere do MAP nas camadas 1 e 2, usando o

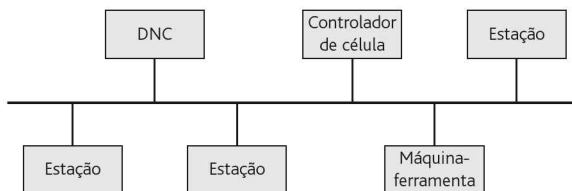


Figura 22.5 MAP.

token com um anel ou o método CSMA/CD com uma rede de barramento. O TOP tem também, na camada 7, protocolos de aplicações específicas voltadas para necessidades administrativas, em vez de requisitos de chão de fábrica. Com o método CSMA/CD, as estações têm que “ouvir” as outras transmissões antes de transmitir. As redes TOP e MAP são compatíveis podendo ser interconectadas com um dispositivo *gateway*. Este dispositivo executa as conversões de endereço e mudanças de protocolos apropriadas.

A **arquitetura de rede de sistemas (SNA – Systems Network Architecture)** é um sistema desenvolvido pela IBM com um projeto padrão para produtos IBM. A SNA é dividida em sete camadas; entretanto, difere, em alguma extensão, das camadas no modelo OSI (Figura 22.6(a)). A camada de controle de conexão de dados provê suporte para *token ring* para LANs. Cinco das camadas SNA são consolidadas em 2 pacotes: a rede de controle de caminho para as camadas 2 e 3 e a unidade endereçável da rede para as camadas 4, 5 e 6.

A Novell Netware descreve uma série de sistemas de operação da LAN; isto inclui a Netware Lite para redes *peer-to-peer* com dois a cerca de duas dúzias de usuários, Netware 3.x para LANs com um único servidor suportando centenas de usuários e Netware 4.x, que é o esquema de operação de uma rede da empresa. A Figura 22.6(b) mostra como uma série de protocolos relaciona as camadas OSI. O protocolo do núcleo Netware define os procedimentos usados para a transmissão de informação para um servidor e seus clientes na LAN Netware. Os *pipes* nomeados fornecem uma interface entre processos executados em computadores diferentes. O sistema de entrada e saída básico da rede (NetBIOS – *Network Basic Input Output System*) é uma interface do programa de aplicação. A troca de pacotes sequenciais (SPX) define os protocolos para envio de mensagens de uma extremidade da rede para outra. A troca de pacotes entre redes (IPX) é um protocolo de rede *peer-to-peer*. A interface aberta de enlace de dados (ODI) e as especificações de interface do driver de rede (NDIS) definem protocolos para enviar e receber informações entre unidades que são conectadas diretamente umas às outras. Na camada física, a Netware suporta estas tecnologias como a Ethernet e *token ring*.

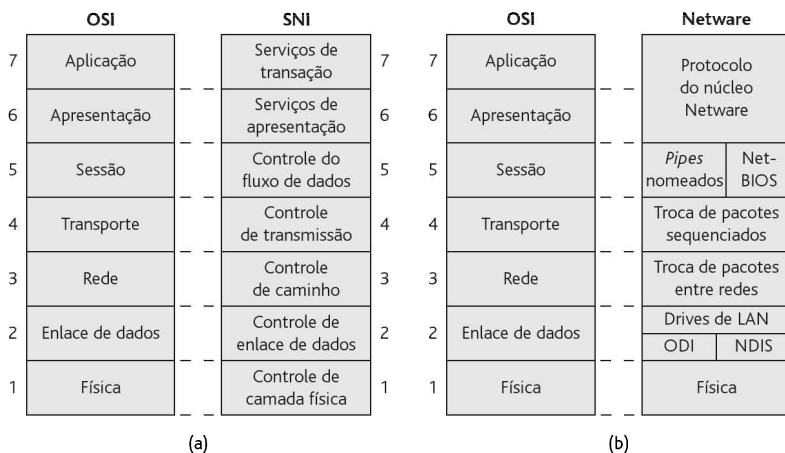


Figura 22.6 (a) SNA, (b) Netware.

Nos sistemas de CLP, é bastante comum os sistemas usados serem registrados pelo fabricante do CLP. Por exemplo, a Allen-Bradley tem o **Allen-Bradley data highway** que usa a passagem de *token* para controlar a transmissão de mensagens; a Mitsubishi tem o Melsec-Net, e a Texas Instruments tem o TIWAY. Um sistema normalmente usado em sistemas com rede de CLP é a Ethernet. Este é um sistema de barramento simples com CSMA/CD usado para controlar o acesso, e é muito usado com sistemas que envolvem CLPs que se comunicam com computadores. O problema com o uso do CSMA/CD é que, enquanto que este método funciona bem quando o tráfego não é intenso, conforme o tráfego da rede aumenta o número de colisões e as consequentes retransmissões aumentam. O *throughput* (vazão) da rede pode diminuir bastante.

22.6 Interfaces de comunicação serial

A interface serial pode envolver protocolos síncronos ou assíncronos. Normalmente, as interfaces assíncronas usadas são RS-232 e suas versões posteriores, o *loop* de corrente de 20 mA, I²C, Xn W USB.

22.6.1 RS-232

A interface serial mais popular é a **RS-232**; ela foi a primeira interface definida pela associação das indústrias de eletrônica americanas (EIA – American Electronic Industries Association), em 1962. O padrão relaciona o equipamento terminal de dados (DTE – *Data Terminal Equipment*) e o equipamento de comunicação de dados (DCE – *Data Communication Equipment*). O DTE pode enviar ou receber dados via a interface, como, por exemplo, um microcontrolador. O DCE é o dispositivo que facilita a comunicação; um exemplo típico é o modem. Este é a conexão essencial entre um microcomputador e uma linha telefônica analógica convencional.

Os sinais RS-232 podem ser agrupados em três categorias:

1 *Dados*

A RS-232 fornece dois canais de dados seriais independentes, denominados primário e secundário. Estes canais são usados em operações *full-duplex*.

2 *Controle de handshake*

Os sinais de *handshaking* são usados para controlar o fluxo de dados em série pela via de comunicação.

3 *Temporização*

Para a operação de sincronismo, é necessário passar sinais de clock entre transmissores e receptores.

A Tabela 22.1 fornece a numeração dos pinos do conector RS-232C e os sinais em cada um; nem todos os pinos e sinais são necessariamente usados em uma configuração particular. O fio de GND do sinal permite um caminho de retorno. O conector para a porta RS-232C é um conector tipo D em uma via de 25 pinos; geralmente é usado um conector macho nos cabos e um fêmea no DCE ou DTE.

No enlace bidirecional mais simples, são necessárias apenas as linhas 2 e 3 para a transmissão e recepção de dados, com o sinal GND (7) sendo o caminho de re-

Tabela 22.1 Designações dos pinos de uma RS-232

Pino	Abreviação	Direção: Para	Sinal/função
1	FG		Terra de proteção
2	TXD	DCE	Dado transmitido
3	RXD	DTE	Dado recebido
4	RTS	DCE	Solicitação de envio
5	CTS	DTE	Liberado para envio
6	DSR	DTE	DCE pronto
7	SG		Terra do sinal/retorno comum
8	DCD	DTE	Detector de linha recebida
12	SDCD	DTE	Detector de sinal de linha recebida secundária
13	SCTS	DTE	CTS secundário
14	STD	DCE	Dado transmitido secundário
15	TC	DTE	Temporização do sinal de transmissão
16	SRD	DTE	Dado recebido secundário
17	RC	DTE	Temporização do sinal recebido
18		DCE	Loop-back local
19	SRTS	DCE	RTS secundário
20	DTR	DCE	Detector de qualidade de loop-back/sinal remoto
21	SQ	DEC/DTE	Terminal de dados pronto
22	RI	DTE	Indicador de chamada
23		DEC/DTE	Seletor de taxa de sinal de dados
24	TC	DCE	Temporização do sinal de transmissão
25		DTE	Modo de teste

torno destes sinais (Figura 22.7(a)). Portanto, as conexões mínimas usam um cabo de 3 fios. Para uma configuração simples envolvendo um computador pessoal (PC) conectado a uma unidade de display visual (VDU), os pinos 1, 2, 3, 4, 5, 6, 7 e 20 es-

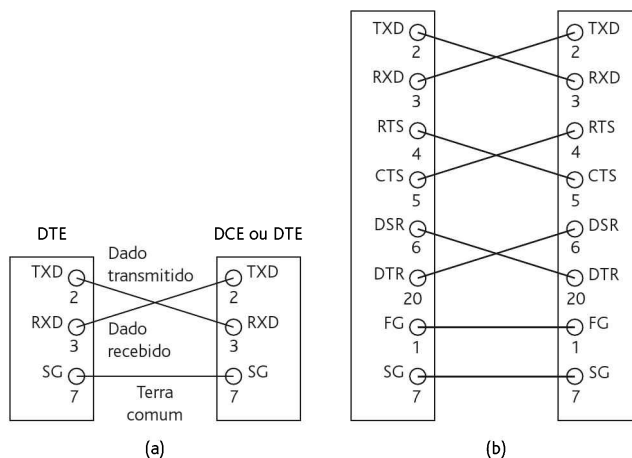


Figura 22.7 Conexões RS-232: (a) configuração mínima, (b) conexão com PC.

tão envolvidos (Figura 20.7(b)). Os sinais enviados pelos pinos 4, 5, 6 e 20 são usados para verificar que a recepção está pronta para receber um sinal, a transmissão está pronta para enviar e o dado está pronto para ser enviado.

A RS-232 é limitada em relação à distância na qual ela pode ser usado à medida que o ruído limita a transmissão de um alto número de bits por segundo quando o comprimento do cabo é maior do que cerca de 15 m. A taxa de dados máxima é cerca de 20 kbits/s. Outros padrões similares ao RS-232 são o RS-422 e o RS-485, que podem ser usados para transmissões em taxas e distâncias maiores.

A RS-422 usa um par de linhas para cada sinal e pode operar até cerca de 1220 m ou em uma velocidade de transmissão maior de até 100 bits/s e em ambientes com ruído; entretanto, a velocidade e a distância máximas não podem ser atingidas simultaneamente. A RS-485 pode ser usada em até cerca de 1220 m com velocidades de 100 kbits/s

As interfaces de comunicação serial do microcontrolador MC68HC11 da Motorola são capazes de comunicações *full-duplex* em uma variedade de taxas baud. Entretanto, as entradas e saídas deste sistema usam a lógica transistor-transistor (TTL), para a qual o nível lógico 0 é 0 V e o 1 é 5 V. O padrão RS-232C usa +12 V para o nível lógico 0 e -12 V para o nível lógico 1. Portanto, é necessário uma conversão nos níveis dos sinais. Isto pode ser feito usando-se circuitos integrados, como o MC1488 para a conversão de TTL para RS-232C e o MC1489 para conversão de RS-232C para TTL (Figura 22.8).

22.6.2 Loop de corrente de 20 mA

Uma outra técnica baseada no RS-232, mas que não é parte do padrão, é o *loop de corrente de 20 mA* (Figura 22.9). Esta técnica usa um sinal de corrente em vez de

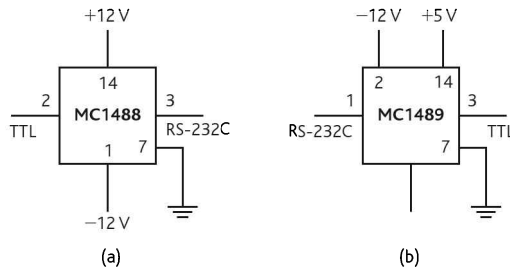


Figura 22.8 Conversão de nível: (a) MC1488, (b) MC1489.

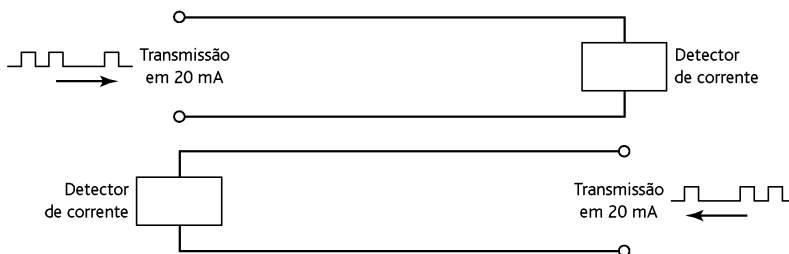


Figura 22.9 Loop de corrente de 20 mA.

tensão. É usado um par de fios separados para os *loops* de transmissão e recepção, com uma corrente no nível de 20 mA que é usada para indicar nível lógico 1 e 0 mA para nível lógico 0. Os dados seriais são codificados com um bit de início, oito bits de dados e dois bits de fim. Estes sinais de corrente permitem uma distância bem maior, alguns quilômetros, entre transmissor e receptor do que com as conexões de tensão do padrão RS-232.

22.6.3 Barramento I²C

O **barramento de comunicação entre CIs**, conhecido como barramento I²C, é um barramento de dados serial projetado pela Philips para uso em comunicações entre circuitos integrados ou módulos. Este barramento permite que dados e instruções sejam trocados entre dispositivos através de dois fios apenas. Isto resulta em uma considerável simplificação de circuitos.

As duas linhas são: uma de dados bidirecional (SDA) e uma de clock (SCL). As duas linhas são conectadas à fonte de alimentação positiva via resistores (Figura 22.10). O dispositivo que gera a mensagem é o transmissor, e o que recebe a mensagem é o receptor. O dispositivo que controla a operação do barramento é o mestre, e os que são controlados pelo mestre são os escravos.

A seguir é discutido o protocolo usado: uma transferência de dados só pode iniciar quando o barramento não estiver ocupado e, durante a transferência dos dados, quando a linha de clock for nível alto, a linha de dados tem que permanecer. Mudanças na linha de dados quando a linha de clock estiver em nível alto são interpretadas como sinal de controle.

- 1 Quando as linhas de clock e dados estiverem em nível alto, o barramento não está ocupado.
- 2 Uma mudança no estado da linha de dados de nível alto para baixo enquanto o clock estiver em nível alto é usada para definir o início da transferência de dados.
- 3 Uma mudança no estado da linha de dados de nível baixo para alto enquanto o clock for nível alto define o fim da transferência de dados.
- 4 Os dados são transferidos entre as condições de início e fim.
- 5 Após um início na transferência de dados, a linha de dados é estável durante os períodos de nível alto do sinal de clock, podendo mudar durante os períodos de nível baixo do sinal de clock.

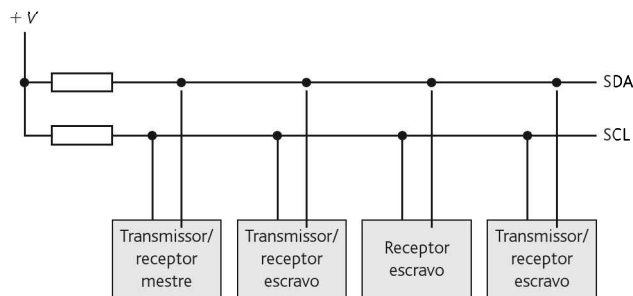


Figura 22.10 Barramento I²C.

- 6 Existe um pulso de clock por bit de dado transferido sem limite do número de bytes de dados que podem ser transferidos entre as condições de início e fim; após cada byte de dados, o receptor reconhece com um nono bit.
- 7 O bit de reconhecimento é um nível alto colocado no barramento pelo transmissor, e um nível baixo pelo receptor.

A Figura 22.11 ilustra isto mostrando a forma de onda do sinal de clock e as saídas de um transmissor e um receptor.

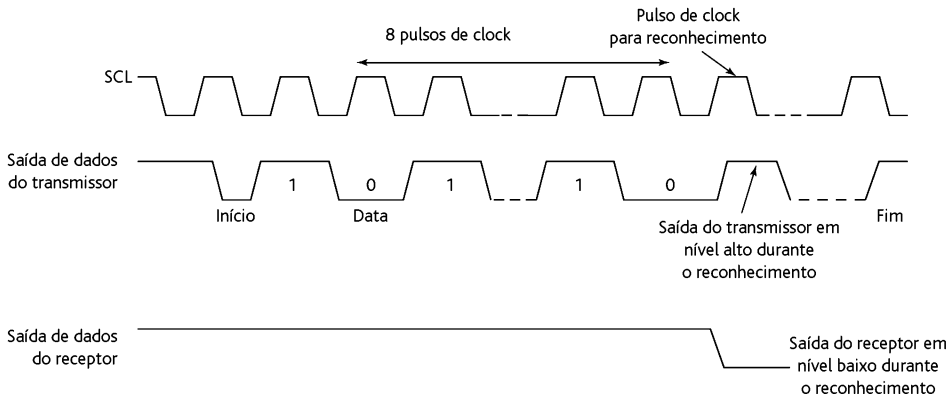


Figura 22.11 Condições do barramento.

22.6.4 Barramento CAN

Os carros modernos têm sistemas de gerenciamento do motor, freios antitravamento (ABS), controle de tração, suspensão ativa e muitos outros sistemas que necessitam conexão com sensores e a interconexão destes sistemas. Isto pode envolver a utilização de muito fio. Entretanto, uma abordagem alternativa é o uso de um barramento de dados comum, com os dados sendo transmitidos por ele e disponibilizados em todos os sistemas do carro. A Bosch desenvolveu um protocolo conhecido como **CAN** (*Controller Area Network*). A comunicação na rede CAN, para transmissão serial ao longo do barramento, consiste de uma sequência de bits com um comprimento total de 44 a 108 bits. Eles são o bit de início, os bits para identificação do destino da comunicação e sua prioridade, bits de controle, os dados a serem transmitidos, um CRC para detecção de erro, um sinal para confirmação e, finalmente, os bits de fim. Quando um transmissor coloca um nome no barramento, ele também lê o nome de volta do barramento, e se o nome não for o mesmo, então outro transmissor, que tem uma prioridade maior, deve estar em operação.

22.6.5 USB

O **barramento serial universal** (USB – *Universal Serial Bus*) foi projetado para permitir que monitores, impressoras, modems e outros dispositivos de entrada sejam conectados facilmente em um PC. O USB usa uma topologia estrela em camadas (*tired star*). Portanto, apenas um dispositivo necessita ser conectado a um PC, com outros dispositivos podendo ser conectados no *hub* resultante.

22.6.6 Firewire

O *firewire* é um barramento serial desenvolvido pela Apple Computers, sendo a especificação dada pelo IEEE 1394. Ele oferece possibilidade de conexão *plug-and-play* e é usado em aplicações como acionadores de disco, impressoras e câmeras.

22.7 Interfaces de comunicação paralela

No caso da interface paralela com uma impressora, normalmente é usada a interface Centronics. Entretanto, no caso de instrumentação, a interface paralela de comunicação mais usada é o **barramento de instrumentação de propósito geral** (GPIB – *General Purpose Instrument Bus*), o padrão IEEE 488, originalmente desenvolvido pela Hewlett Packard para interconectar computadores e instrumentos e que ficou conhecido como **Barramento de Instrumentação da Hewlett Packard**. Cada um dos dispositivos conectados ao barramento é denominado “ouvinte”, “falante” ou controlador. Os “ouvintes” são dispositivos que aceitam dados do barramento, os “falantes” colocam dados no barramento, quando solicitados, e os controladores gerenciam o fluxo de dados no barramento enviando comandos aos “falantes” e “ouvintes” e executando consultas para saber quais dispositivos estão ativos (Figura 22.12(a)).

A interface tem um total de 24 linhas:

- 1 Oito linhas bidirecionais para transportar dados e comandos entre os diversos dispositivos conectados no barramento.
- 2 Cinco linhas para controle e sinais de status.
- 3 Três linhas para *handshaking* entre dispositivos.
- 4 Oito linhas são de retorno de terra.

A Tabela 22.2 lista as funções das linhas e a numeração dos pinos em um conector tipo D de 25 vias. Até 15 dispositivos podem ser conectados ao barramento em qualquer momento, cada dispositivo tendo o seu próprio endereço.

O barramento de dados em paralelo de 8 bits pode transmitir dados na forma de bytes. Cada vez que um byte é transferido, o barramento entra em um ciclo de

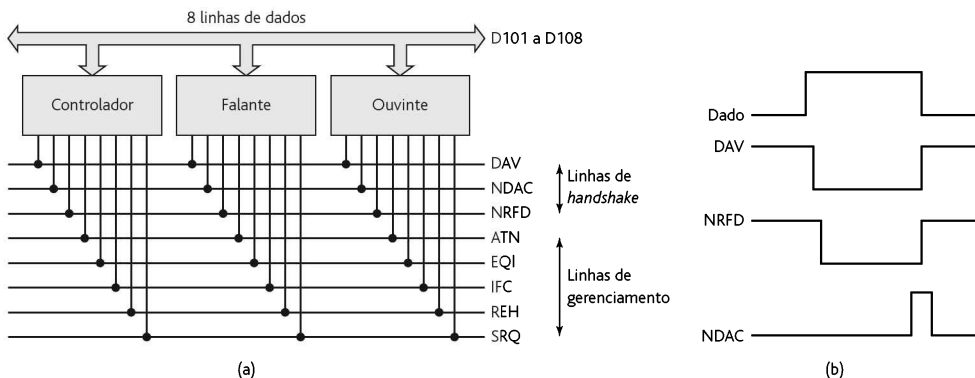


Figura 22.12 Barramento GPIB: (a) estrutura, (b) *handshaking*.

Tabela 22.2 Sistema de barramento IEEE 488

Pino	Grupo de sinais	Abreviação	Função
1	Dado	D101	Linha de dados 1
2	Dado	D102	Linha de dados 2
3	Dado	D103	Linha de dados 3
4	Dado	D104	Linha de dados 4
5	Gerenciamento	EOI	Fim ou identificação. Usado para identificar o fim de uma sequência de mensagem do dispositivo "falante" ou usado pelo controlador para solicitar a um dispositivo que se identifique.
6	<i>Handshake</i>	DAV	Dado válido. Quando o nível for baixo nesta linha, a informação no barramento de dados é válida e aceitável.
7	<i>Handshake</i>	NRFD	Não pronto para dados. Esta linha é usada pelos dispositivos ouvintes. Em nível alto, indica que eles estão prontos para aceitar dados.
8	<i>Handshake</i>	NDAC	Dado não aceito. Esta linha é usada pelos ouvintes. Em nível alto indica que os dados são aceitos.
9	Gerenciamento	IFC	Inicializa interface. Usada pelo controlador para resetar todos os dispositivos do sistema para colocá-los no estado inicial.
10	Gerenciamento	SRQ	Requisição de serviço. Usada pelos dispositivos para sinalizar ao controlador que eles precisam de atenção.
11	Gerenciamento	ATN	Atenção. Usada pelo controlador para sinalizar que ele está colocando um comando nas linhas de dados.
12		SHIELD	Blindagem
13	Dado	D105	Linha de dados 5
14	Dado	D106	Linha de dados 6
15	Dado	D107	Linha de dados 7
16	Dado	D108	Linha de dados 8
17	Gerenciamento	REN	Habilitação remota. Habilita um dispositivo para indicar que ele será selecionado para controle remoto em vez do seu próprio painel de controle.
18		GND	Terra/comum (par trançado com DAV)
19		GND	Terra/comum (par trançado com NRFD)
20		GND	Terra/comum (par trançado com NDAC)
21		GND	Terra/comum (par trançado com IFC)
22		GND	Terra/comum (par trançado com SRQ)
23		GND	Terra/comum (par trançado com ATN)
24		GND	Terra do sinal

handshake. Cada dispositivo no barramento tem o seu próprio endereço. Os comandos do controlador são sinalizados colocando-se a linha de atenção (ATN) em nível baixo. Os comandos são então direcionados aos dispositivos individualmente colocando-se os endereços nas linhas de dados; os endereços dos dispositivos são enviados via linhas de dados como uma palavra de 7 bits, sendo que os 5 bits menos significativos fornecem o endereço do dispositivo e os outros 2 bits são para informações de controle. Se estes dois bits estiverem em 0, então os comandos são enviados a todos os endereços; se o bit 6 for 1 e o 7 for 0, então o dispositivo é comutado para ser "ouvinte"; se o bit 6 for 0 e o 7 for 1, então o dispositivo é comutado para ser um "falante".

Para o *handshaking*, são usadas as linhas DAV, NRFD e NDAC, que são as três linhas que garantem que o “falante” falará apenas quando for escutado pelos “ouvintes” (Figura 22.12(b)). Quando um “ouvinte” está pronto para receber dados, a linha NRFD é feita nível alto. Quando os dados são colocados na linha, a linha DAV é feita nível baixo para notificar os dispositivos que os dados estão disponíveis. Quando um dispositivo aceita uma palavra de dados, ele coloca NDAC em nível alto para indicar que ele aceitou o dado e coloca NRFD em nível baixo para indicar que ele está pronto para receber dados. Quando todos os “ouvintes” colocam NDAC em nível alto, então o “falante” cancela o sinal de dado válido colocando DAV em nível alto. Isto resulta em NDAC colocada em nível baixo. Todo este processo pode ser repetido para uma nova palavra colocada no barramento de dados.

A GPIB é um barramento que é usado para fazer a interface de uma ampla gama de instrumentos, como, por exemplo, multímetros e osciloscópios digitais, via placas *plug-in* (Figura 22.13) para computadores com cabos padrão usados para conectar as placas com os instrumentos via interfaces.

22.7.1 Outros barramentos

Os barramentos usados para conectar a CPU em portas de entrada/saída ou outros dispositivos incluem:

- 1 O *barramento de computador XT* foi introduzido em 1983 para transferência de dados de 8 bits entre o IBM PC/XT e computadores compatíveis.
- 2 O *barramento AT*, também conhecido como barramento da arquitetura padrão da indústria (ISA – *Industry Standard Architecture*), foi introduzido mais tarde para uso com o IBM PC de 16 bits e outros computadores compatíveis usando os microprocessadores 80286 e 80386. O barramento AT é compatível com o XT, de forma que as placas XT *plug-in* podem ser usadas nos slots do barramento AT.
- 3 O *barramento de arquitetura padrão da indústria estendido* (EISA – *Extended Industry Standard Architecture*) foi desenvolvido para lidar com transferências de dados de 32 bits com o IBM PC e outros computadores compatíveis usando os microprocessadores 80386 e 80486.
- 4 O *barramento de arquitetura de microcanal* (MCA – *Micro Channel Architecture*) é um barramento de transferência de dados de 16 ou 32 bits projetado para uso com os computadores IBM Personal System/1 (PS/2). As placas para uso com este barramento não são compatíveis com as placas PC/XT/AT.

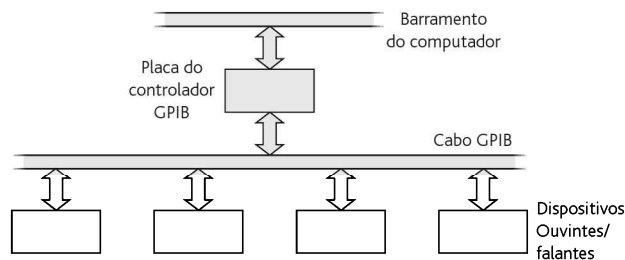


Figura 22.13 Hardware GPIB.

- 5 O barramento *NuBus* de 32 bits é usado em computadores Macintosh II da Apple.
- 6 O barramento *S-bus* de 32 bits é usado em estações SPARC da Sun Microsystems.
- 7 O *TURBOchannel* é um barramento de 32 bits usado em estações de trabalho DECstation 5000.
- 8 O barramento *VME* foi projetado pela Motorola para ser usado no sistema baseado no microprocessador 68000 de 32 bits. Entretanto, este barramento agora é amplamente usado com outros sistemas de computador como barramento para uso com sistemas de instrumentação.

Estes são os chamados **barramentos *backplane*** (apresentam estrutura de interconexões no chassi). Este termo se refere à placa (Figura 22.14) na qual os conectores são montados e na qual são conectadas as placas de circuito impresso com funções específicas, como, por exemplo, memória. O *backplane* provê os barramentos de dados, endereço e sinais de controle para cada placa, permitindo que os sistemas sejam facilmente expandidos com o uso de placas *off-the-shelf*. É nestes barramentos de computadores que as placas de aquisição de dados, de instrumentos e outros periféricos têm que interfacear. As placas de aquisição de dados e instrumentos são geralmente disponíveis em várias configurações, dependendo do computador com o qual elas serão usadas.

O VXIbus (extensão do VME para instrumentação) é uma extensão da especificação do VMEbus que foi projetada para aplicações de instrumentação, como equipamentos de teste automático, que necessitam de comunicações de velocidades maiores que as que o barramento GPIB pode proporcionar. Ele também fornece uma melhor sincronização e disparo e foi desenvolvido por um consórcio de fabricantes de instrumentos, de modo que é possível a interoperabilidade entre os produtos de diferentes companhias. O sistema envolve placas VXI conectadas em *mainframe*. A Figura 22.15 mostra alguns sistemas de configurações possíveis que podem ser usados. Na Figura 22.15(a) um *mainframe* VXI é conectado a um controlador externo, um computador, via conexão GPIB. O controlador “conversa” através desta conexão usando um protocolo GPIB para uma placa de interface no chassi que traduz o protocolo GPIB para VXI. Isto faz com que os instrumentos apareçam para o controlador como sendo GPIB e permite que eles sejam programados usando-se métodos GPIB. A Figura 22.15(b) mostra o computador completo embutido em um chassi VXI. Esta opção oferece o menor tamanho físico possível para o sistema e permite que o computador use diretamente o barramento *backplane* VXI. A Figura 22.15(c) usa um sistema de barramento com cabo especial de alta velocidade, o MXIbus, para conectar um computador e o chassi VXI, sendo o MXI 20 vezes mais rápido que a GPIB.

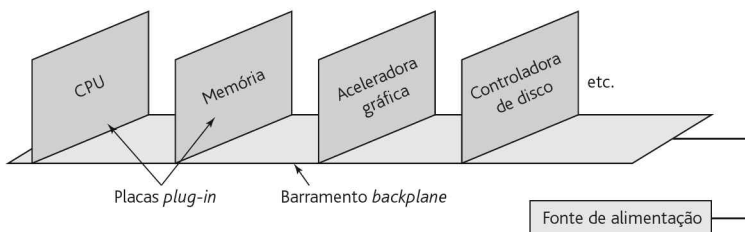


Figura 22.14 Barramento *backplane*.

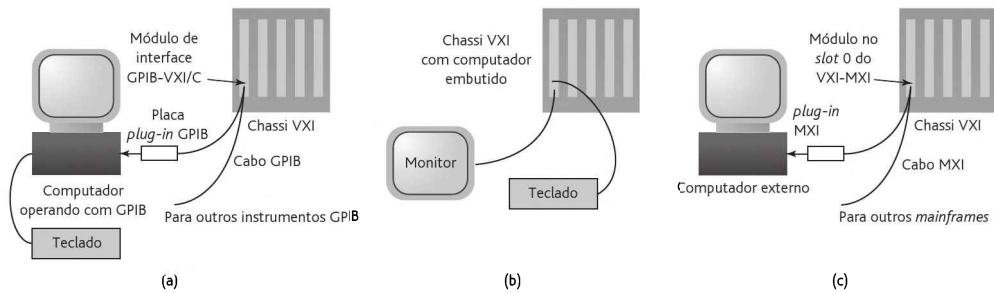


Figura 22.15 Opções VXI.

22.8 Protocolos *wireless*

O padrão IEEE 802.11 é uma proposta para LANs *wireless* (sem fio) que especifica as camadas física (PHY) e a de controle de acesso ao meio (MAC) da rede. A camada MAC especifica um protocolo de acesso múltiplo com detecção de portadora e prevenção de colisão (CSMA/CA). Com isto, quando um nó tem um pacote pronto para transmissão, ele primeiro “escuta” o meio para garantir que nenhum outro nó está transmitindo e, em caso afirmativo, transmite. Caso contrário, espera e tenta novamente. Quando um pacote é transmitido, o nó de transmissão primeiro envia um pacote RTS contendo informação sobre o comprimento do pacote e, em seguida, envia o pacote. Quando o pacote é recebido com sucesso, o nó de recepção transmite um pacote ACK.

Bluetooth é um padrão global para transmissões de rádio de curto alcance. Quando dois dispositivos equipados com Bluetooth estão distanciados um do outro até 10 m, a conexão pode ser estabelecida. Este padrão é amplamente usado em telefonia móvel e PCs.

Resumo

Um **barramento externo** é um conjunto de linhas de sinal que interconecta microprocessadores, microcontroladores, computadores e CLPs e também conecta estes com equipamentos periféricos.

O **controle computacional centralizado** envolve o uso de um computador central para controlar toda a planta. No sistema hierárquico, há um sistema de hierarquia de computadores de acordo com as tarefas que eles executam. Em um sistema distribuído, cada sistema de computador executa essencialmente tarefas similares a todos os outros sistemas de computador.

A comunicação de dados pode ser via conexões de **transmissão paralela** ou **serial**. A transmissão de dados em série pode ser assíncrona ou síncrona. A **transmissão assíncrona** implica que os computadores de transmissão e recepção não são sincronizados, tendo cada um o seu próprio sinal de clock. A transmissão de dados em série ocorre em um dos três modos: *simplex*, *half-duplex* e *full-duplex*.

O termo **rede** é usado para um sistema que permite que dois ou mais computadores/microprocessadores sejam interconectados para troca de dados. Normalmente, os tipos de redes usadas são de barramento, estrela, hierarquia/árvore, anel e malha. O controle de acesso a redes é necessário para garantir que apenas um usuário seja capaz de transmitir em um determinado momento; nas redes baseadas em anel, os métodos usados são passagem de *token* e passagem de *slot*, enquanto que o acesso múltiplo com detecção de portadora e de colisão é usado nas redes de barramento ou hierárquica. Um **protocolo** é um conjunto formal de regras que define o formato, temporização, sequência, controle de acesso e controle de erro dos dados.

A Organização Internacional para Padronização (ISO – *International Organization for Standardization*) definiu um sistema de protocolo padrão de sete camadas denominado **OSI** (*Open System Interconnection*).

As **interfaces de comunicação serial** incluem a RS-232 e suas versões posteriores, I²C e CAN. As **interfaces de comunicação paralela** incluem a GPIB (*General Purpose Instruments Bus*).

Problemas

- 22.1 Explique a diferença entre sistemas de comunicação centralizado e distribuído.
- 22.2 Explique os tipos de barramentos/árvore e redes em anel.
- 22.3 Com uma distância entre os nós da rede maior do que 100 m, utiliza-se uma LAN. A topologia escolhida deve ser de anel ou barramento?
- 22.4 Uma determinada rede tem que ser uma LAN multicanal. A escolha para o tipo de transmissão deve ser a de banda base ou a *broadband*?
- 22.5 O que são MAP e TOP?
- 22.6 Explique o que significa protocolo de comunicação.
- 22.7 Explique de forma resumida os dois tipos de controle de acesso múltiplo usado em LANs.
- 22.8 Um microcontrolador M68HC11 é um “ouvinte” a ser conectado a um “falante” via um barramento GPIB. Indique as conexões a serem feitas se o *handshaking* completo é usado.
- 22.9 Que tipo de problema tem que ser superado antes que a interface de comunicação serial de dados do microcontrolador M68HC11 possa enviar dados através de uma interface RS-232?
- 22.10 O que é um barramento *backplane*?

23

Identificação de Falhas

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Reconhecer as técnicas usadas para identificar falhas em sistemas microprocessados, incluindo *hardware* e *software*.
- Explicar o uso da emulação e da simulação.
- Explicar como a identificação de falhas pode ser feita em sistemas de CLP.

23.1 Técnicas de detecção de falhas

Este capítulo apresenta uma breve consideração dos problemas de detecção de falhas em sistemas de medição, controle e comunicação de dados. Para a obtenção de detalhes sobre a pesquisa de falhas em componentes ou sistemas específicos, deve-se usar o manual do fabricante.

A seguir apresentamos algumas técnicas de detecção de defeitos:

1 *Verificação por replicação*

Esta técnica envolve a duplicação ou replicação de uma atividade e a comparação dos resultados. Na ausência de falhas, considera-se que os resultados devem ser iguais. Ela pode significar apenas a repetição, com erros transientes, de uma operação e a comparação dos resultados, ou pode envolver um sistema duplicado e a comparação dos resultados dados por cada um dos dois. Esta pode ser uma opção cara.

2 *Verificação por valor esperado*

Os erros em softwares são normalmente detectados verificando-se se um valor esperado é obtido quando uma entrada numérica específica ocorre. Se o valor esperado não for obtido, então há uma falha.

3 Verificação por temporização

Esta envolve o uso de alguma função executada dentro de um tempo especificado. Esta verificação é normalmente conhecida como **temporizador *watchdog***. Por exemplo, em um CLP, quando uma operação inicia, um temporizador também é iniciado, e se a operação não for completada dentro do tempo especificado, considera-se que ocorreu uma falha. O temporizador *watchdog* dispara, desliga um alarme e desliga uma parte da planta ou toda ela.

4 Verificação reversa

Onde há uma relação direta entre os valores de entrada e saída, o valor de entrada pode ser calculado a partir do valor de saída. Este valor calculado pode ser comparado com o valor real de entrada.

5 Verificação por paridade e codificação de erro

Esta forma de verificação é normalmente usada para detectar erros de memória e transmissão de dados. Os canais de comunicações estão frequentemente sujeitos a interferências que podem afetar os dados transmitidos. Para detectar se os dados foram corrompidos, é acrescentado um bit de paridade à palavra de dados transmitida. O bit de paridade é escolhido para tornar o número resultante de uns no grupo ímpar (paridade ímpar) ou par (paridade par). No caso da paridade ímpar, a palavra pode ser verificada após a transmissão para ver se ainda permanece com o número de uns ímpar. Outras formas de verificação envolvem o acréscimo de códigos aos dados transmitidos para detectar bits corrompidos.

6 Verificação por diagnóstico

As verificações por diagnóstico são usadas para testar o comportamento dos componentes em um sistema. Entradas são aplicadas em um componente e as saídas são comparadas com as que devem ocorrer.

23.2 Temporizador *watchdog*

Um temporizador *watchdog* é basicamente um temporizador que o sistema tem que resetar antes que ele termine a temporização. Se o temporizador não for resetado a tempo, considera-se que ocorreu um erro.

A Figura 23.1 mostra uma ilustração deste temporizador através de um programa ladder simples que pode ser usado para prover um CLP com um temporizador *watchdog* para a operação que envolve a movimentação de um pistão em um cilindro. Quando a chave de partida é fechada, o solenoide A+ é ativado e começa o movimento do seu pistão no cilindro. Ele também inicia um temporizador. Quando o pistão estiver totalmente estendido, ele abre a chave fim de curso a+. Isto para o temporizador. Entretanto, se a+ não abrir antes do temporizador finalizar, os contatos do temporizador fecham e um alarme é ativado. Portanto, o temporizador pode ser setado, por exemplo, com 4 s, considerando que o pistão esteja totalmente estendido dentro deste tempo. Entretanto, se ele fica parado e não atinge a extensão total, o alarme é ativado.

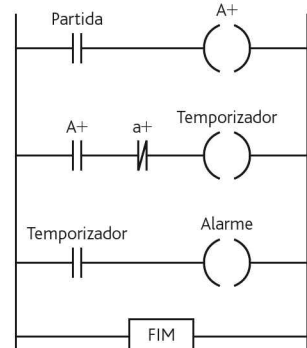


Figura 23.1 Temporizador *watchdog*.

Quando um microprocessador executa instruções a partir de sua memória, um distúrbio elétrico próximo ao sistema pode interferir momentaneamente no barramento de dados do processador, fazendo com que bytes de informações apresentem erros quando acessados. Alternativamente, um problema de software pode resultar em um erro quando o processador retorna de uma sub-rotina. Uma consequência desta falha é que o sistema pode gerar uma situação de risco em atuadores controlados pelo microprocessador. Para evitar que isto aconteça em sistemas críticos, é utilizado um temporizador *matchdog* para resetar o microprocessador.

Como uma ilustração de um sistema microprocessador com temporizador *matchdog*, considere o microcontrolador MC68HC11, que inclui um temporizador *matchdog* interno denominado **Computer Operating Properly (COP)**, que detecta erro no processamento de softwares. Quando o temporizador COP é disparado, é necessário que o programa principal resete o COP periodicamente antes que ele atinja o final da temporização. Se este evento ocorrer, então acontecerá um reset de falha do COP. O temporizador COP pode ser resetado para zero escrevendo-se \$55 (0x55 em linguagem C) no registrador de reset do COP (COPRST) no endereço \$103A (0x103A) seguido depois no programa da escrita de \$AA (0xAA) para limpar o temporizador COP. Se o programa estiver “preso” entre duas instruções, o temporizador atinge o final da temporização e ocorre a execução da rotina de reset de falha do COP. O programa, em linguagem assembly, tem as seguintes linhas:

```
LDAA    #$55    ; reseta o temporizador
STAA    $103A   ; escreve $55 em COPRST
          ; outras linhas do programa
LDAA    #$AA    ; limpa o temporizador
STAA    $103A   ; escreve $AA em COPRST
```

O período de operação do COP é definido através de CR1 e CR2 no registrador de OPTION, no endereço \$1039 (0x1039), que podem ser 0 ou 1. Por exemplo, com CR1 em 0 e CR2 também, é configurada uma temporização de 16,384 ms, ao passo que se CR1 for 1 e CR2 for 0, é configurada uma temporização de 262,14 ms.

23.3 Verificação por paridade e por codificação de erro

Para detectar quando um sinal de dados foi corrompido por um ruído, são usadas técnicas de detecção de erro como a verificação por paridade.

A Seção 4.5.2 apresenta uma descrição resumida do método de paridade na detecção de erro. Este método, um bit extra é acrescentado à mensagem para tornar o número total de uns par, quando se usa a paridade par, ou ímpar, quando se usa a paridade ímpar. Por exemplo, o caractere 1010000 tem como bit de paridade par um 0 colocado após o bit mais significativo, ou seja, 01010000, ou um 1 no caso de paridade ímpar, ou seja, 11010000.

Este método pode detectar a presença de um único erro na mensagem, mas não a presença de dois erros, o que não resultaria em alteração da paridade, como, por exemplo, com a paridade par e o número acima apresentando um único erro, digamos, no quarto bit (01011000), este seria detectado na verificação de paridade, mas

o mesmo não ocorreria se houvesse um erro no terceiro bit também (01011100), pois o bit de paridade estaria correto.

Se não for detectado erro, o sinal é reconhecido como correto através do retorno do caractere ACK para o terminal que enviou; se for detectado erro, o sinal NAK é usado. Isto é denominado **requisição de repetição automática (ARQ)**. O sinal NAK resulta na retransmissão da mensagem.

A eficiência na detecção de erro pode ser aumentada com o uso da **paridade de bloco**. A mensagem é dividida em um número de blocos e cada bloco tem um caractere de verificação do bloco acrescentado no final do bloco. Por exemplo, com o bloco a seguir, é colocado um bit de verificação para paridade par no final de cada linha e um bit de verificação adicional na base de cada coluna:

	Bits da informação				Bit de verificação
Primeiro símbolo	0	0	1	1	0
Segundo símbolo	0	1	0	0	1
Terceiro símbolo	1	0	1	1	1
Quarto símbolo	0	0	0	0	0
Bits de verificação do bloco	1	1	0	0	0

No receptor, a paridade de cada linha e cada coluna é verificada e qualquer erro simples é detectado pela interseção da linha e da coluna que contém o bit de verificação de erro.

Outra forma de detecção de erro é a **verificação de redundância cíclica (CRC)**. No terminal de transmissão, o número binário que representa o dado transmitido é dividido por um determinado número usando-se a aritmética de módulo 2. O resto da divisão é o caractere CRC, que é transmitido com o dado. No receptor, o dado mais o caractere CRC são divididos pelo mesmo número. Se não ocorrer erro na transmissão, o resto será zero.

Um código CRC comum é o CRC-16, que usa 16 bits para a sequência de verificação. Os 16 bits são considerados os coeficientes de um polinômio com o número de bits igual à maior potência do polinômio. O bloco de dados é multiplicado primeiro pela maior potência do polinômio, ou seja, x^{16} , e então dividido pelo polinômio CRC

$$x^{16} + x^{12} + x^5 + 1$$

usando-se a aritmética de módulo 2, ou seja, $x = 2$ no polinômio. O polinômio CRC é então 10001000000100001. O resto da divisão por este polinômio é o CRC.

Como uma ilustração, suponha que temos o dado 10110111 ou o polinômio

$$x^7 + x^5 + x^4 + x^2 + x^1 + 1$$

e um polinômio CRC de

$$x^5 + x^4 + x^1 + 1$$

ou 110011. O polinômio do dado é multiplicado primeiro por x^5 , resultando em

$$x^{12} + x^{10} + x^9 + x^7 + x^6 + x^5$$

e, portanto, 1011011100000. Dividindo isto pelo polinômio CRC, obtemos

$$\begin{array}{r}
 \underline{1101011} \\
 110011 \mid 1011011100000 \\
 \underline{110011} \\
 110011 \\
 \underline{110011} \\
 100100 \\
 \underline{110011} \\
 101110 \\
 \underline{110011} \\
 111010 \\
 \underline{110011} \\
 01001
 \end{array}$$

e, assim, um resto de 01001, que se torna o código CRC transmitido com o dado.

23.4 Falhas comuns de *hardware*

Algumas das falhas que podem ocorrer com tipos específicos de componentes e sistemas são abordadas a seguir.

23.4.1 Sensores

Se existem falhas no sistema de medida, então a falha pode estar no sensor. Um teste simples é substituir o sensor por um novo e verificar o resultado apresentado pelo sistema. Se o resultado mudar, então o sensor original provavelmente está com defeito; se o resultado não mudar, então o defeito está em outra parte do sistema. É possível verificar também se as fontes de tensão/corrente estão fornecendo os valores corretos, se há continuidade nos fios de conexão, se o sensor está montado corretamente e sendo usado sob as condições especificadas nas folhas de dados do fabricante, etc.

23.4.2 Chaves e relés

Sujeira e partículas de material desgastado entre os contatos da chave são fontes comuns de funcionamento incorreto de chaves mecânicas. Com o uso de um multímetro sobre uma chave deve-se constatar a tensão aplicada quando os contatos da chave estão abertos e uma tensão muito próxima de zero quando estiverem fechados. As chaves mecânicas usadas para detectar a posição de algum item, como, por exemplo, a presença de uma peça a trabalhar sobre um transportador, pode não conseguir fornecer a resposta correta se o alinhamento estiver incorreto ou a alavanca de atuação estiver encurvada.

A inspeção de um relé pode revelar evidências de arco ou solda de contato. O relé deve ser substituído nestes casos. Se um relé falhar na sua operação, então pode ser feita uma verificação na tensão de sua bobina. Se a tensão estiver correta, então a continuidade da bobina pode ser verificada com um ohmímetro. Se não houver tensão na bobina, então a falha pode estar no transistor de comutação usado com o relé.

23.4.3 Motores

A manutenção de motores CC e CA envolve a lubrificação correta. Nos motores CC, as escovas se desgastam e precisam ser substituídas. O conjunto de escovas novas precisa estar de acordo com as especificações do fabricante. Um motor monofásico com partida a capacitor que esteja lento na partida provavelmente está precisando de um novo capacitor de partida. Um motor de indução trifásico não tem escovas, comutador, anéis coletores ou capacitor de partida e, exceto no caso de forte sobrecarga, a única manutenção exigida regularmente é a lubrificação periódica.

23.4.4 Sistemas hidráulicos e pneumáticos

Uma causa comum de falhas em sistemas pneumáticos e hidráulicos é a sujeira. Pequenas partículas de sujeira podem danificar vedações, bloquear orifícios, emperrar carretéis de válvulas, etc. Os filtros devem ser regularmente verificados e limpos, os componentes devem ser desmontados apenas quando limpos, e o óleo deve ser regularmente verificado e trocado. Um método comum de teste de circuitos elétricos é medir as tensões nos pontos de testes. Da mesma forma, nos sistemas hidráulicos e pneumáticos há a necessidade de pontos nos quais as pressões podem ser medidas. Os danos em vedações podem resultar em vazamentos em cilindros hidráulicos e pneumáticos além do que pode ser considerado normal, resultando em uma queda de pressão no sistema quando o cilindro é atuado. Isto pode ser remediado substituindo-se as vedações dos cilindros. As palhetas em motores do tipo palheta estão sujeitas a desgaste e podem falhar na vedação com a carcaça do motor, tendo como resultado uma perda da potência do motor. As palhetas podem ser substituídas. Vazamentos em mangueiras, tubos e conexões são falhas comuns.

23.5 Sistemas microprocessados

As falhas típicas nos sistemas microprocessados são:

1 *Falha no chip*

Os chips são resistentes a falhas, mas ocasionalmente podem apresentar algum defeito.

2 *Falha em componente passivo*

Os sistemas microprocessados geralmente incluem componentes passivos, como resistores e capacitores. Falhas em qualquer um destes podem provocar mau funcionamento no sistema.

3 *Circuitos abertos*

Um circuito aberto pode resultar em uma ruptura no percurso de um sinal ou na linha de alimentação. As razões típicas para estas falhas são junções dessoldadas ou com defeitos na solda, ruptura na trilha da placa de circuito impresso, uma conexão com defeito em um conector e ruptura em cabos.

4 *Curto-circuito*

Um curto-circuito entre pontos em uma placa que não deveriam estar conectados frequentemente surge como resultado de excesso de solda que faz uma ponte entre trilhas ou *pads* (ilhas) vizinhas.

5 Interferência introduzida externamente

Pulsos introduzidos externamente afetam a operação do sistema, visto que eles são interpretados como sinais digitais válidos. Estas interferências podem ser originadas de uma fonte de alimentação que contém *spikes* como resultado de outros equipamentos compartilhando o mesmo circuito principal e sendo ligados e desligados. Podem ser usados filtros, na fontes principais, para remover estes *spikes*.

6 Falhas nos softwares

Apesar de passarem por testes extensivos, ainda é comum que os softwares contenham erros (*bugs*) e que, para uma determinada condição particular de entrada ou saída, este *bug* apareça.

23.5.1 Técnicas de detecção de falhas

As técnicas de detecção de falhas que são usadas em sistemas microprocessados incluem:

1 Inspeção visual

Uma observação cuidadosa de um sistema com defeito pode revelar a fonte do problema, como, por exemplo, um circuito integrado solto do seu soquete ou um excesso de solda que faz uma ponte entre trilhas de uma placa de circuito.

2 Multímetro

Este tem uso limitado em sistema microprocessados, mas também pode ser usado para verificar conexões de curto-circuito ou circuito aberto e fontes de alimentação.

3 Osciloscópio

O osciloscópio é essencialmente limitado a situações onde ocorrem sinais repetitivos, e o sinal mais óbvio é o de clock. A maioria dos outros sinais em um sistema microprocessado não é repetitiva e depende do programa em execução.

4 Ponta de prova lógica

A ponta de prova lógica é um dispositivo manual (Figura 23.2(a)), semelhante a uma caneta, que pode ser usado para determinar o nível lógico em qualquer

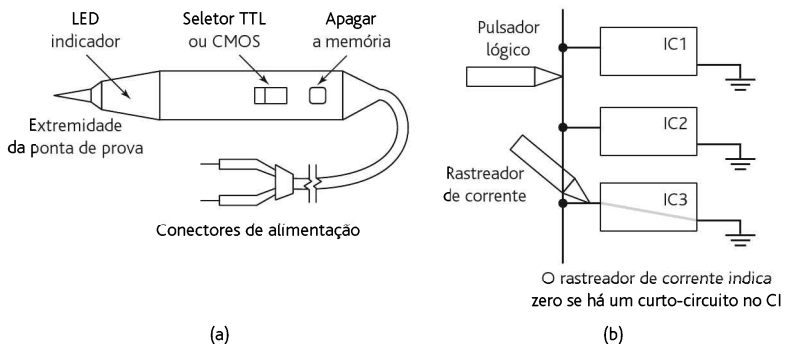


Figura 23.2 (a) Ponta de prova lógica, (b) rastreador de corrente.

ponto do circuito no qual está conectada. A chave seletora é posicionada em TTL ou CMOS e, quando a ponta de prova toca o ponto desejado, um LED indica se o ponto está abaixo do limiar do nível lógico 0, acima do limiar do nível lógico 1 ou se há um sinal pulsante. Normalmente é incluído na ponta de prova lógica um circuito extensor de pulso para estender a duração de um pulso para que dure tempo suficiente para que o efeito de acionamento do LED seja notado. Um circuito de memória pode ser usado para detectar um único pulso. Um botão para apagar a memória é pressionado para desligar o LED e permitir que qualquer variação no nível lógico seja registrada e indicada pelo LED.

5 *Pulsador lógico*

O pulsador lógico é um gerador de pulsos manual, similar a uma caneta, que é usado para injetar pulsos controlados em circuitos. A ponta do pulsador é pressionada contra um nó no circuito, e um botão no pulsador é pressionado para gerar um pulso. Geralmente ele é usado com a ponta de prova lógica para verificar as funções de portas lógicas.

6 *Rastreador de corrente*

O rastreador de corrente é similar à ponta de prova lógica, mas ele detecta pulsos de corrente em um circuito em vez de níveis de tensão. A ponta do rastreador de corrente é magneticamente sensível e é usada para detectar variações de campo magnético próximo a um condutor que transporta um pulso de corrente. A ponta do rastreador de corrente é movimentada ao longo da trilha na placa de circuito impresso para rastrear os percursos de baixa impedância nos quais a corrente está fluindo (Figura 23.2(b)).

7 *Clip lógico*

Um clip lógico é um dispositivo que se encaixa sobre o circuito integrado, fazendo contatos com cada pino do circuito integrado. O estado lógico de cada pino é mostrado por LEDs, havendo um para cada pino.

8 *Comparador lógico*

O comparador lógico testa circuitos integrados comparando-os a um circuito integrado de referência em bom estado (Figura 23.3). Sem remover o circuito integrado sob teste, cada pino de entrada é conectado em paralelo com o pino de entrada correspondente do circuito integrado de referência; da mesma forma, cada pino de saída é conectado ao pino de saída correspondente do CI de referência. As duas saídas são comparadas com uma porta EX-OR, que fornece uma saída quando as duas saídas são diferentes. O extensor de pulso é usado para estender a duração do sinal que aciona o indicador de modo que pulsos de duração muito curta possam ser notados através do indicador.

9 *Analisador de assinatura*

Em sistemas analógicos, a análise de falha geralmente envolve o rastreamento do circuito e o exame das formas de onda em diversos nós, comparando as formas de onda com aquelas esperadas, permitindo que o defeito seja identificado e localizado. Em sistemas digitais, o procedimento é mais complexo, visto que trens de pulsos em nós são muito similares. Para identificar se há um defeito, a sequência de pulsos é convertida em uma forma mais rapidamente identificável, como, por exemplo, 258F, que é denominada **assinatura**. A assinatura obtida

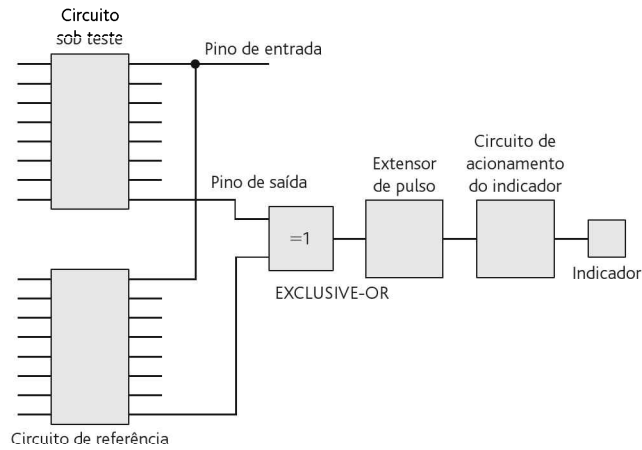


Figura 23.3 Comparador lógico.

em um nó pode ser comparada com a esperada. Quando se usa um analisador de assinatura em um circuito, geralmente é necessário que o circuito tenha sido projetado de modo que as vias de realimentação do barramento de dados sejam interrompidas facilmente para o teste parar a sequência digital com defeito que é realimentada durante o teste. Um pequeno programa, que é armazenado em ROM, é ativado para simular nós e permitir que assinaturas sejam obtidas. O próprio microprocessador pode ser testado se o barramento de dados for interrompido para isolá-lo da memória e fazê-lo operar “livre”, executando instruções “no operation” (NO). As assinaturas no barramento do microprocessador neste estado podem ser comparadas com as esperadas.

10 Analisador lógico

O analisador lógico é usado para amostrar e armazenar simultaneamente em uma memória FIFO (*first-in-first-out*) os níveis lógicos do barramento e os sinais de controle na unidade sob teste. O ponto no programa em que a captura dos dados começa ou termina é selecionado com o uso de uma “palavra de disparo”. O analisador compara a palavra de disparo com a palavra obtida e apenas começa a armazenar dados quando a palavra ocorre no programa. A captura de dados continua por um número pré-determinado de pulsos de clock, parando em seguida. Os dados armazenados podem então ser mostrados como uma lista de códigos binário, octal, decimal ou hexadecimal, como um display temporal em que as formas de onda são mostradas como uma função do tempo, ou como um display de mnemônicos.

23.5.2 Métodos sistemáticos de localização de falhas

Os métodos sistemáticos de localização de falhas são:

1 Entrada para saída

Um sinal de entrada adequado é injetado no primeiro bloco do sistema e medições são feitas em sequência, começando pelo primeiro bloco, na saída de cada bloco até que o bloco com defeito seja encontrado.

2 Saída para a entrada

Um sinal de entrada adequado é injetado no primeiro bloco do sistema e medições são realizadas na sequência, começando pelo último bloco, na saída de cada bloco até que o bloco com defeito seja encontrado.

3 Divisão pela metade

Um sinal de entrada adequado é injetado no primeiro bloco do sistema. Os blocos que constituem o sistema são divididos em duas partes, e cada uma é testada para determinar qual delas está com defeito. A metade que estiver com defeito é então dividida em duas partes e o procedimento se repete.

23.5.3 Autoteste

Os softwares podem ser usados por um sistema microprocessador para estabelecer um programa de autoteste para o funcionamento correto. Estes programas são normalmente inicializados durante a sequência de *start-up* (inicialização) de um sistema quando ele é ligado pela primeira vez. Por exemplo, as impressoras incluem microprocessadores em seus circuitos de controle, e geralmente os programas de controle que são armazenados em ROM também incluem rotinas de testes. Portanto, quando energizada, eles passam por estas rotinas de testes sem poder receber dados até que todos os testes indiquem que o sistema não apresenta defeito.

Um teste de ROM básico envolve a soma de todos os bytes de dados armazenados em cada posição de memória e a comparação com a soma do que foi armazenado (conhecido como **teste checksum**). Se houver uma diferença entre estas somas, então a ROM está com defeito; se forem iguais, ela é considerada sem defeito. Um teste básico em uma RAM envolve o armazenamento de padrões de dados nos quais os bits adjacentes são de níveis lógicos opostos, como, por exemplo, 55 em hexa e AA, em cada posição de memória e a leitura de cada valor armazenado para verificar que ele corresponde ao dado escrito (conhecido como **teste de padrão xadrez** [*checkerboard*]).

23.6 Emulação e simulação

Um emulador é uma placa de teste que pode ser usada para testar um microcontrolador e o seu programa. A placa contém:

- 1 O microcontrolador
- 2 Os chips de memória para o microprocessador usar como memória de dados e programa.
- 3 Uma porta de entrada/saída para permitir conexões com o sistema sob teste.
- 4 Uma porta de comunicação para permitir que o código de programa seja transferido a partir do computador e a operação do programa seja monitorada.

O código do programa pode ser escrito em um computador *host* e então transferido através de uma conexão serial ou paralela para a memória na placa. O microcontrolador opera então conforme o programa contido em sua memória interna. A Figura 23.4 mostra a configuração geral.

As linhas de entrada/saída do microcontrolador são conectadas via porta de entrada/saída na placa em um dispositivo *plug-in* no sistema sob teste, de modo que ele opera conforme determinado pelo microcontrolador conectado a ele. A placa já é

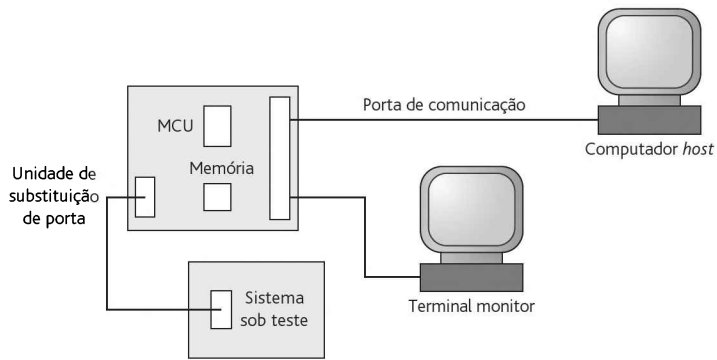


Figura 23.4 Uso de um emulador.

programada com o sistema de monitoração que permite que a operação do programa seja monitorada e os conteúdos da memória, dos registradores e das portas de entrada/saída sejam verificados e modificados.

A Figura 23.5 mostra os elementos básicos da placa de avaliação MC68HC11E-VB que é fornecida pela Motorola. Ela usa um programa monitor denominado *Bit User Fast Friendly Aid to Logical Operations* (Buffalo). A EPROM de 8K contém o programa monitor do Buffalo. É usado um adaptador de interface de comunicação assíncrona (ACIA) MC6850 (veja a Seção 20.5) para interface entre as linhas paralelas e seriais. Uma interface RS-232 parcial é fornecida com duas portas seriais para conexão do computador *host* e o terminal de monitoração.

23.6.1 Simulação

Em vez de testar um programa executando-o com um microcontrolador real, podemos testá-lo executando-o através do uso de um programa de computador que *simula* o microcontrolador. Esta simulação pode auxiliar na depuração (*debugging*)

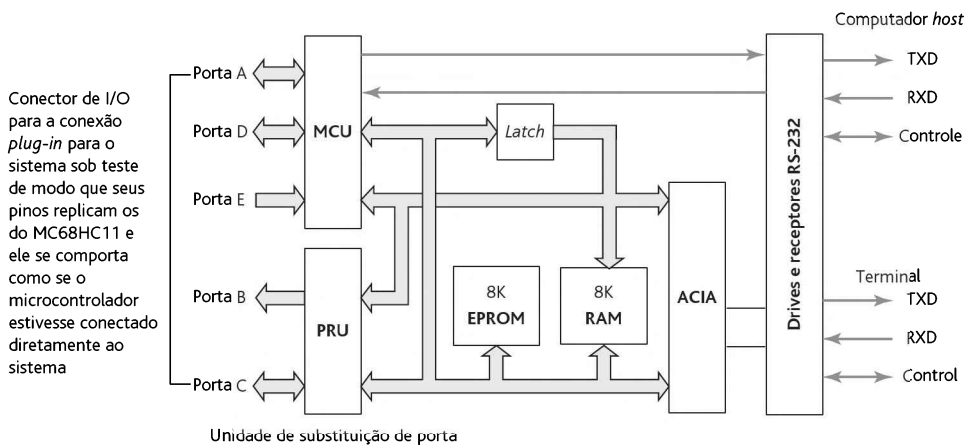


Figura 23.5 MC68HC11EVB.

do código do programa. A tela do display pode ser dividida em janelas nas quais as informações são mostradas, como o código fonte à medida que ele é executado, os registradores da CPU e os *flags* com seus estados atuais, as portas de entrada/saída, registradores e temporizadores e a situação da memória.

23.7 Sistemas de CLP

Os CLPs têm uma alta confiabilidade. Eles são isolados eletricamente através de optoisoladores ou relés para proteger as portas de entrada/saída contra tensões e correntes potencialmente perigosas; uma RAM com bateria como fonte de energia secundária protege o software de aplicação contra falhas na alimentação primária; e a construção é projetada de tal forma que os CLPs podem operar confiavelmente em condições industriais por longos períodos. Geralmente, os CLPs têm procedimentos internos para situações de falhas. As falhas críticas fazem com que a CPU pare, enquanto que outros defeitos menos críticos fazem com que a CPU continue executando o programa, mas mostre um código de falha no display. O manual do CLP indicará a ação que soluciona o problema quando um código de falha for mostrado.

23.7.1 Teste de um programa

O software que verifica o programa busca através do programa ladder endereços de dispositivos incorretos e fornece uma lista na tela, ou impressa, de todos os pontos de entrada/saída usados, ajustes de contadores e temporizadores, etc., com quaisquer erros detectados. Portanto, o procedimento pode envolver:

- 1 Abrir e mostrar o programa ladder de interesse.
- 2 Selecionar, a partir do menu na tela, o Teste Ladder.
- 3 A tela pode mostrar a mensagem: Começar do início do programa (S/N)?
- 4 Digitar S e pressionar Enter.
- 5 Qualquer mensagem de erro ou então a mensagem “Nenhum erro encontrado” é mostrada.

Por exemplo, pode haver uma mensagem para um determinado endereço de saída que é usado em mais de uma saída do programa, um temporizador ou um contador usado sem ser carregado com o valor desejado, um contador usado sem um reset, a ausência da instrução de fim, etc. Como resultado deste teste, pode ser necessário fazer alterações no programa. A retificação do programa pode ser feita selecionando-se a mudança a partir do menu mostrado na tela e seguindo-se o conjunto de mensagens na tela.

23.7.2 Teste de entradas e saídas

A maioria dos CLPs tem o recurso de testar entradas e saídas por meio de uma operação denominada **forçar**. Por meio do software as saídas ou entradas são “forçadas” em *on* ou *off*. Para forçar entradas ou saídas, um CLP tem que ser comutado para o modo forçar ou monitor, podendo ser através de um tecla denominada FORCE ou selecionando-se o modo MONITOR na tela. Como resultado de uma entrada forçada, podemos constatar que a ação consequente ocorre. O programa instalado pode ser executado e as entradas e saídas simuladas de modo que elas, e todos os

valores carregados, podem ser verificados. Entretanto, seja muito cuidadoso com o uso da operação forçar, pois uma saída forçada pode resultar no movimento de um elemento do *hardware* de uma forma inesperada e perigosa.

Como uma ilustração do tipo de símbolos mostrados ao se usar a operação forçar, a Figura 23.6 mostra como as entradas podem aparecer no programa ladder quando abertas e fechadas, e as saídas, quando estão energizadas e não energizadas. A Figura 23.7(a) mostra uma parte de um programa ladder e a Figura 23.7(b) mostra o que acontece quando se usa a operação forçar. Inicialmente, a Figura 23.7(a) mostra o ramo 11 com as entradas X400, X401 e M100 acionadas, mas não a X402, e sem a ativação da saída Y430. No ramo 12, os contatos do temporizador T450 estão fechados, o display na parte de baixo da tela indicando que não há tempo para ativar T450. Como Y430 não está energizada, os contatos de Y430 estão abertos e, assim, a saída Y431 não está ativa. Se forçarmos a entrada X402, a tela passa a ser como mostra a Figura 23.7(b) e Y430 é energizada, e conseqüentemente Y431 também.

23.7.3 CLP como monitor de sistemas

O CLP também pode ser usado como um monitor do sistema controlado. Ele pode ser usado para soar um alarme ou acender uma luz vermelha se as entradas se moverem fora dos limites estabelecidos, usando as funções “maior que”, “igual a” ou “menor que”, ou se suas operações gastarem mais tempo do que o determinado. Veja a Figura 23.1 para uma ilustração de como um programa ladder de um CLP pode ser usado como um temporizador *matchdog* para alguma operação.

Freqüentemente nos sistemas com CLP, são usadas lâmpadas de status para indicar a última saída setada durante um processo, e assim, se o sistema parar, identificamos onde a falha ocorreu. As lâmpadas são ativadas pelo programa de modo que, à medida que cada saída ocorre, uma lâmpada é ligada e a anterior é desligada. A Figura 23.8 ilustra isto.

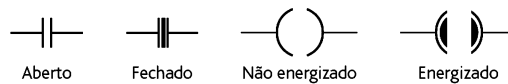


Figura 23.6 Símbolos no modo monitor.

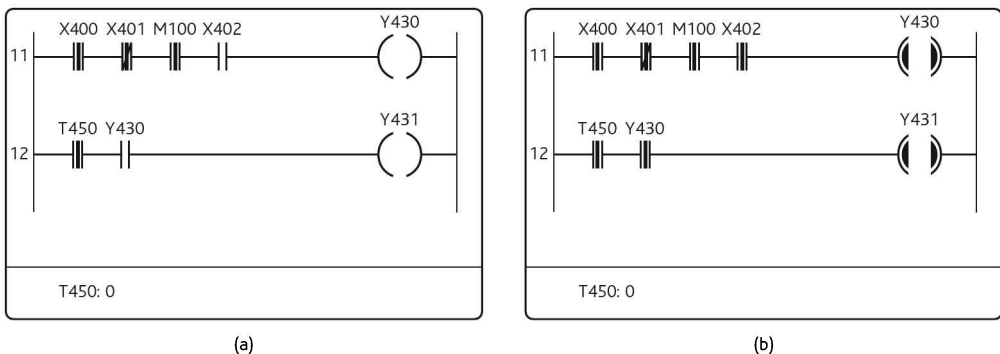


Figura 23.7 Forçando uma entrada.

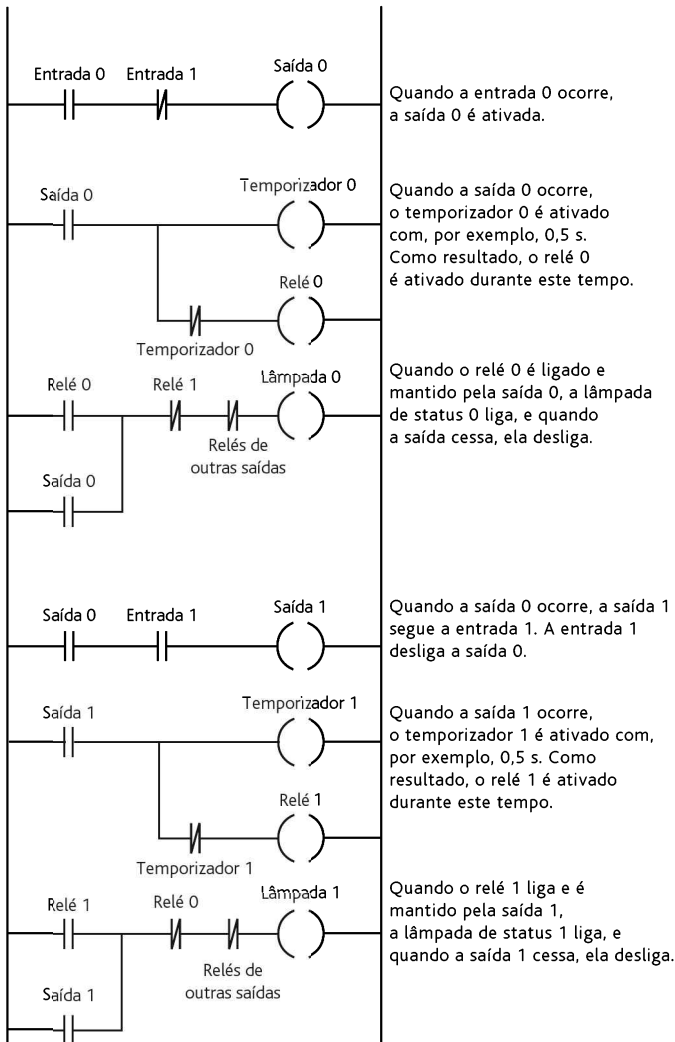


Figura 23.8 A última saída seta o programa de diagnóstico.

Resumo

As técnicas usadas para detectar falhas são verificação por replicação, verificação por valor esperado, verificação por temporização, ou seja, os temporizadores *matchdog*, verificação reversa, verificação por paridade e codificação de erro, e verificação por diagnóstico.

Um **temporizador *matchdog*** é basicamente um temporizador que o sistema tem que resetar antes do tempo terminar. Se o temporizador não for resetado em tempo, considera-se que ocorreu um erro. A **verificação de paridade** envolve um bit extra sendo acrescentado à mensagem para tornar o número de uns par quando for usada a paridade par ou torná-lo ímpar quando for usado a paridade ímpar. A

eficiência da detecção de erro pode ser aumentada pelo uso de **paridade de bloco**, na qual a mensagem é dividida em blocos e cada bloco tem um caractere de verificação acrescentado no final do bloco. A **verificação de redundância cíclica (CRC)** envolve os números binários que representam os dados transmitidos sendo divididos por um número pré-determinado usando a aritmética de módulo 2. O resto da divisão é o caractere CRC que é transmitido com os dados. No receptor, os dados mais o caractere CRC são divididos pelo mesmo número. Se não ocorrer nenhum erro na transmissão, o resto será zero.

Os softwares podem ser usados por um sistema microprocessado para instituir um programa de autoteste para corrigir o funcionamento. Um **emulador** é uma placa de teste que pode ser usada para testar um microcontrolador e o seu programa. Em vez de testar um programa executando-o em um microcontrolador real, ele pode ser testado através de um programa de computador que *simula* o microcontrolador.

Geralmente, os CLPs têm vários procedimentos internos de identificação de defeitos. Os defeitos críticos fazem com que a CPU pare, enquanto que outros menos críticos fazem com que a CPU continue executando o programa, mas mostram o código da falha no display. A maioria dos CLPs tem o recurso de testar entradas e saídas pelo que é denominado de operação **forçar**. O software é usado para “forçar” entradas e saídas em *on* ou *off*.

Problemas

- 23.1 Explique o que quer dizer (a) verificação por replicação, (b) verificação por valor esperado, (c) verificação reversa, (d) verificação por paridade.
- 23.2 Explique como um temporizador *matchdog* pode ser usado com uma planta controlada por CLP para indicar a presença de falhas.
- 23.3 Explique a função do COP no microcontrolador MC68HC11.
- 23.4 A série F2 de CLP da Mitsubishi tem a seguinte especificação:
Diagnóstico: Verificação de programação (soma, sintaxe, verificação de circuito), temporizador *matchdog*, tensão de bateria, tensão de alimentação
Explique o significado destes termos.
- 23.5 Explique como um auto-teste pode ser usado por um sistema microprocessado para verificar suas memórias RAM e ROM.

24

Sistemas Mecatrônicos

OBJETIVOS

Ao final do estudo deste capítulo, o leitor deverá ser capaz de:

- Desenvolver soluções possíveis para problemas de projeto considerando o ponto de vista da mecatrônica.
- Analisar estudos de caso de soluções mecatrônicas.

24.1 Projetos de mecatrônica

Este capítulo reúne muitos dos tópicos discutidos neste livro na consideração de soluções mecatrônicas para problemas de projeto e aborda estudos de casos.

24.1.1 Chave temporizada

Considere um simples requisito para um dispositivo que liga um atuador, como, por exemplo, um motor, por um determinado tempo. As soluções possíveis podem envolver:

- 1 Um came rotativo
- 2 Um CLP
- 3 Um microprocessador
- 4 Um microcontrolador
- 5 Um temporizador, como, por exemplo, o 555.

Uma solução mecânica poderia envolver um came rotativo (Figura 24.1) (veja a Seção 8.4). O came pode ser girado a uma taxa constante e o seguidor de came pode ser usado para acionar uma chave. O tempo no qual a chave é fechada depende do formato do came. Esta é uma solução que foi muito usada no passado.

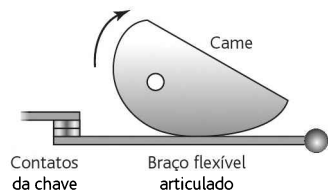


Figura 24.1 Chave acionada com came.

Uma solução via CLP pode envolver a configuração mostrada na Figura 24.2 com o programa ladder dado. Esta solução tem a vantagem em relação ao came rotativo de ter temporizadores para ligar e desligar que podem ser ajustados apenas alterando-se, no programa, o valor carregado no temporizador, ao passo que um came diferente é necessário se o tempo tiver que ser alterado no caso da solução mecânica. A solução por software é muito mais fácil de ser implementada do que por *hardware*.

Uma solução baseada em um microprocessador pode envolver um microprocessador combinado com um chip de memória e interfaces de entrada/saída. O programa é usado para ligar uma saída e, depois de um tempo, desligá-la, sendo que este tempo é gerado por um bloco de programa no qual existe um *loop* de tempo. Isto gera um atraso de tempo ao executar o *loop* o determinado número de vezes que é necessário para gerar o tempo desejado. Portanto, na linguagem assembly, podemos ter:

```

DELAY   LDX   #F424   ; F424 é o número de loops
LOOP    DEX
        BNE   LOOP
        RTS
    
```

DEX decrementa o registrador de índice, e BNE salta se não for igual. Como cada uma tem 4 ciclos, o *loop* tem 8 ciclos e serão executados *n loops* destes até $8n + 3 + 5$ resultar no número F424 (LDX tem 3 ciclos e RTS tem 5). Na linguagem C podemos escrever as linhas do programa usando a função WHILE.

Uma outra possibilidade é usar o temporizador do microcontrolador, como no MC68HC11. Este microcontrolador possui um contador de 16 bits, TCNT, que opera a partir do sinal E-clock (Figura 24.3(a)). Este sistema de clock pode fazer uso do *prescaler* setando-se os bits no registrador 2 de máscara de interrupção do temporizador (TMSK2), cujo endereço é \$1024 (Figura 24.3(b)). O registrador TCNT começa com \$0000 quando o processador é resetado e conta continuamente até alcançar a contagem máxima de \$FFFF. No próxima pulso ele sofre um overflow e volta para \$0000 novamente. Quando isto ocorre, ele seta o *flag de overflow* do

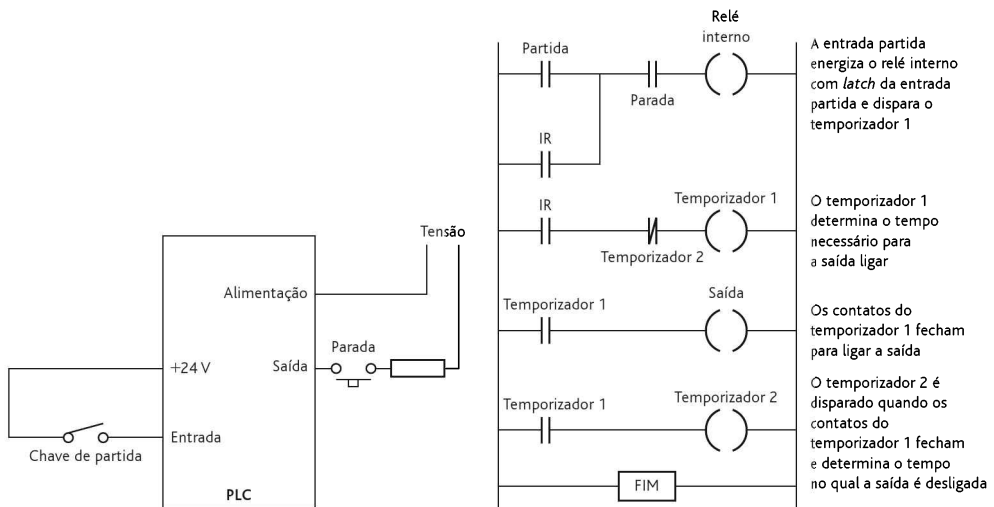


Figura 24.2 Sistema de temporização com CLP.

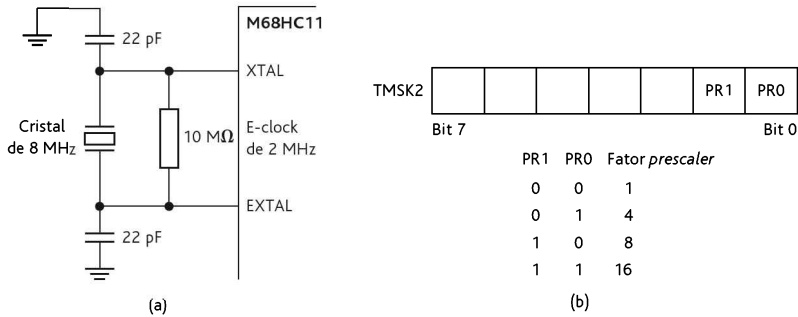


Figura 24.3 (a) Geração de clock interno de 2 MHz, (b) fator *prescaler*.

temporizador, TOF, (bit 7 no registrador 2 de *flags* de interrupção do temporizador, TFLG2 no endereço \$1025). Portanto, com um fator *prescaler* de 1 e um E-clock de 2 MHz, o *overflow* ocorre após 32,768 ms.

Uma forma de realizar temporização é por meio de uma consulta no flag TOF. Quando o flag é setado, o programa incrementa seu contador. Então, o programa reseta o flag escrevendo 1 no bit 7 do registrador TFLG2. Portanto, a operação de temporização consiste apenas no programa esperar pelo número desejado de ativações do flag de overflow.

Uma melhor forma de implementar uma temporização é usar a função de comparação de saída. A Porta A do microcontrolador pode ser usada para entradas ou saídas gerais ou para funções de temporização. O temporizador tem os pinos de saída OC1, OC2, OC3, OC4 e OC5 com os registradores internos TOC1, TOC2, TOC3, TOC4 e TOC5. Podemos usar a função de comparação de saída para comparar os valores nos registradores TOC1 a TOC5 com os valores do contador autônomo TCNT. Este contador começa em 0000 quando a CPU é resetada e, em seguida, conta continuamente. Quando ocorre uma igualdade entre um registrador e o contador, o *flag* OCx correspondente é setado e ocorre uma saída através dos pinos de saída relevantes. A Figura 24.4 ilustra isto. Portanto, programando-se os registradores TOCx, os momentos em que as saídas ocorrem podem ser setados. A função de comparação de saída pode gerar atrasos de temporização com uma precisão muito maior do que com o *flag* de *overflow* do temporizador.

O programa a seguir ilustra como a comparação de saída pode ser usada para gerar atrasos de tempo. O atraso de tempo mais longo que pode ser gerado em uma operação de comparação de saída é 32,7 ms quando o E-clock for de 2 MHz. Para gerar atrasos maiores, são necessárias operações múltiplas de comparação de saída.

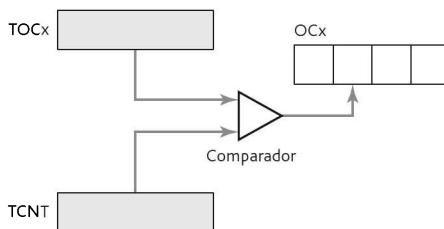


Figura 24.4 Comparação da saída.

Portanto, podemos ter cada operação de comparação de saída produzindo um atraso de 25 ms, e repetindo isto 40 vezes obtemos um atraso total de 1 s.

REGBAS	EQU	\$1000	; Endereço base dos registradores
TOC2	EQU	\$18	; <i>Offset</i> de TOC2 para REGBAS
TCNT	EQU	\$0E	; <i>Offset</i> de TCNT para REGBAS
TFLG1	EQU	\$23	; <i>Offset</i> de TFLG1 para REGBAS
OC1	EQU	\$40	; Máscara para zerar o pino OC1 e o <i>flag</i> OC1F
CLEAR	EQU	\$40	; Zerar o <i>flag</i> OC2F
D25MS	EQU	50000	; Número de ciclos do E-clock para gerar um atraso de 25 ms
NTIMES	EQU	40	; Número de operações de comparação de saída necessário para gerar um atraso de 1 s
	ORG	\$1000	
COUNT	RMB	1	; Posição de memória para guardar o número de operações de comparação de saída a serem executadas
	ORG	\$C000	; Endereço de início do programa
	LDX	#REGBAS	
	LDA	#OC1	; Zera <i>flag</i> OC1
	STAA	TFLG1,X	
	LDA	#NTIMES	; Inicializa o contador de comparações de saída
	STAA	COUNT	
	LDD	TCNT,X	
WAIT	ADDD	#D25MS	; Soma atraso de 25 ms
	STD	TOC2,X	; Começa a operação de comparação de saída
	BRCLR	TFLG1,X OC1	; Espera até que o <i>flag</i> OC1 seja setado
	LDA	#OC1	; Zera <i>flag</i> OC1
	STAA	TFLG1,X	
	DEC	COUNT	; Decrementa o contador de comparação de saída
	BEQ	OTHER	; Desvia para OTHER se decorreu 1 s
	LDD	TOC2,X	; Prepara para iniciar a próxima operação de comparação
	BRA	WAIT	
OTHER			; As outras operações do programa que ocorrem após o atraso de 1 s

Outro método possível de produzir um sinal de saída temporizado é usar um módulo temporizador, como, por exemplo, o 555. Com este temporizador, os intervalos de tempo são definidos por resistores e capacitores externos. A Figura 24.5 mostra o temporizador e o circuito externo necessário para gerar uma saída quando disparado, sendo a duração da saída $1,1RC$. Para tempos grandes são necessários valores grandes de R e C . R é limitado a cerca de $1\text{ M}\Omega$, caso contrário as fugas se tornam um problema, e C é limitado a cerca de $10\text{ }\mu\text{F}$ para que capacitores eletrolíticos com os problemas de fuga e baixa precisão sejam evitados. Portanto, o circuito mostrado é limitado a tempos menores do que cerca de 10 s. O limite inferior é cerca de $R = 1\text{ k}\Omega$ e $C = 100\text{ pF}$, ou seja, tempos de uma fração de milissegundo. Para tempos maiores, de 16 ms a dias, uma alternativa é usar o temporizador ZN1034E.

24.1.2 Movimento do limpador de para-brisa

Considere os requisitos para um dispositivo que oscila um braço para frente e para trás em um arco como um limpador de para-brisa. As soluções possíveis podem ser:

- 1 Conexão mecânica e um motor CC
- 2 Um motor de passo

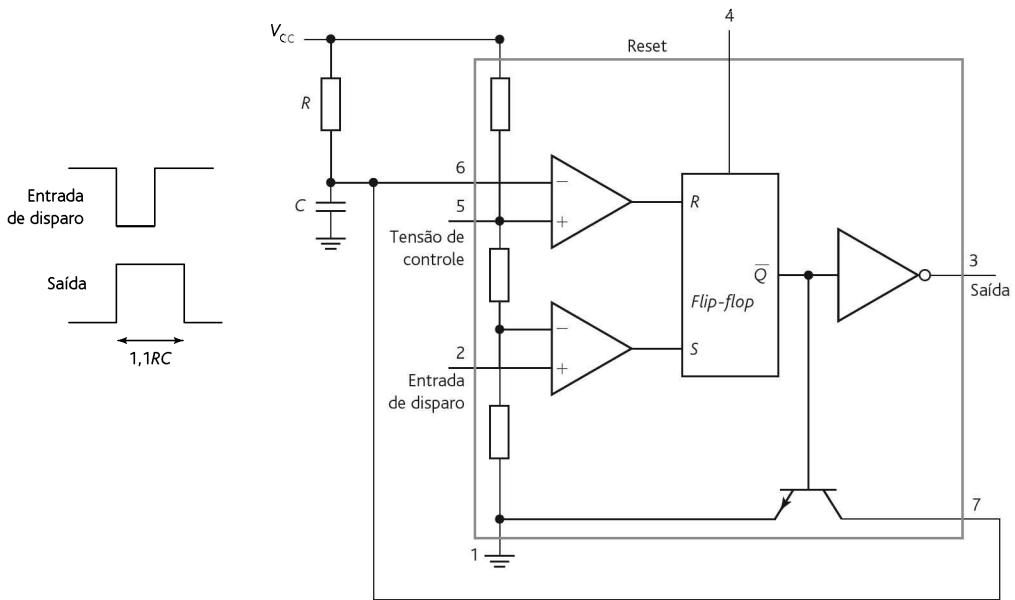


Figura 24.5 Temporizador 555.

Uma solução mecânica é mostrada na Figura 24.6. A rotação do braço 1 por meio de um motor faz com que o braço 2 comunique um movimento oscilatório ao braço 3. Geralmente, o limpador de para-brisa usa este mecanismo com um motor CC de ímã permanente.

Uma solução alternativa é o uso de um motor de passo. A Figura 24.7 mostra como um microprocessador com uma PIA, ou um microcontrolador, pode ser usado com um motor de passo. A entrada do motor de passo é necessária para fazer com que ele gire um número de passos em uma direção e então inverta para girar o mesmo número de passos na outra direção.

Se o motor de passo estiver na configuração “passo completo”, então as saídas precisam ser conforme a Tabela 24.1(a). Portanto, girar o motor no sentido direto envolve a sequência A, 9, 5, 6 e retornar ao passo 1 novamente. Para inverter, usamos a sequência 6, 5, 9, A, retornando então ao 6. Se for usada a configuração “meio-passo”, as saídas precisam ser como mostra a Tabela 24.1(b). O movimento no sentido direto envolve a sequência A, 8, 9, 1, 5, 4, 6, 2 e retornar ao A. O sentido reverso é 2, 6, 4, 5, 1, 9, 8, A e de volta ao 2.

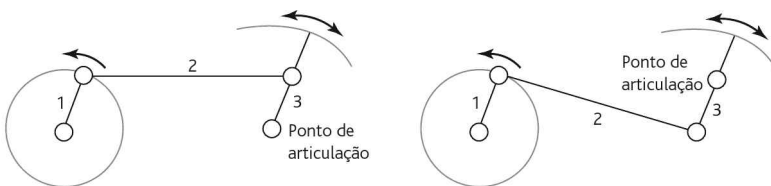


Figura 24.6 Mecanismo do limpador.

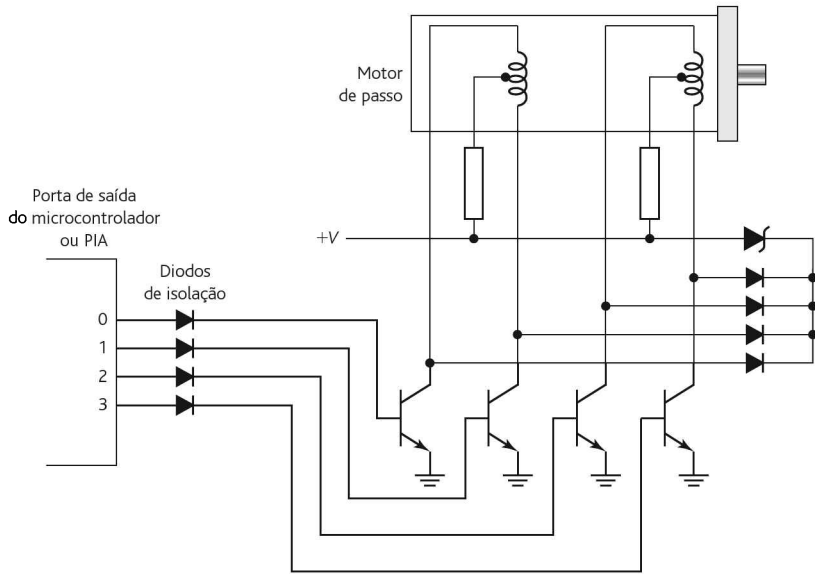


Figura 24.7 Interface com um motor de passo.

Tabela 24.1 Configurações: (a) passo completo, (b) meio-passo

(a)						(b)				
Passo	Bit 3	Bit 2	Bit 1	Bit 0	Código	Passo	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	1	0	A	1	1	0	1	0
2	1	0	0	1	9	2	1	0	0	0
3	0	1	0	1	5	3	1	0	0	1
4	0	1	1	0	6	4	0	0	0	1
1	1	0	1	0	A	5	0	1	0	1
						6	0	1	0	0
						7	0	1	1	0
						8	0	0	1	0
						1	1	0	1	0

Os elementos básicos de um programa poderiam ser:

Avançar um passo

Saltar para uma rotina de atraso de tempo para que o passo seja finalizado

Repetir os itens acima até que o número de passos desejado ocorra no sentido direto

Inverter sentido

Repetir os itens acima para o mesmo número de passos no sentido reverso

Este programa em C para três meios-passos no sentido direto e três no sentido reverso, e com a inclusão de um arquivo de cabeçalho adequado, pode ter a seguinte forma:


```

main ( )
{
  portB = 0xa; /*primeiro passo*/
  delay ( ); /*incorpora rotina de atraso para, digamos, 20 ms*/
  portB = 0x8; /*segundo passo*/
  delay ( ); /*incorpora rotina de atraso para, digamos, 20 ms*/
  port B = 0x9; /*terceiro passo*/
  delay ( ); /*incorpora rotina de atraso para, digamos, 20 ms*/
  port B = 0x8; /*retornar um passo*/
  delay ( ); /*incorpora rotina de atraso para, digamos, 20 ms*/
  port B = 0xa; /*retornar mais um passo*/
  delay ( ); /*incorpora rotina de atraso para, digamos, 20 ms*/
  port B = 0x2; /*retornar ao ponto de partida do motor*/
  delay ( ); /*incorpora rotina de atraso para, digamos, 20 ms*/
}

```

Onde há muitos passos envolvidos, um programa mais simples faz uso do incremento de um contador para cada passo até que o contador alcance o valor desejado. Este programa pode ter a seguinte forma básica:

Avançar um passo
 Saltar para uma rotina de atraso de tempo para que o passo seja finalizado
 Incrementar o contador
 Repetir os itens acima em passos sucessivos até que o contador inidique o número desejado de passos completados no sentido direto
 Inverter sentido
 Repetir os itens acima para o mesmo número de passos no sentido reverso

Os circuitos integrados disponíveis para controle de motor de passo podem simplificar a interface e o software. A Figura 24.8 mostra como um CI destes pode ser usado. Tudo que é necessário é o número desejado de pulsos na entrada do disparo, o motor de passo sofrendo uma transição de baixo-para-alto para um pulso alto-baixo-alto. Um nível alto na entrada rotação faz com que o motor se movimente no sentido anti-horário, enquanto que um nível baixo nesta entrada faz o motor girar no sentido horário. Portanto, precisamos apenas uma saída do microcontrolador para gerar pulsos de disparo e uma para escolher o sentido de rotação. Um sinal em SET é usado para levar o motor de volta para a sua posição original.

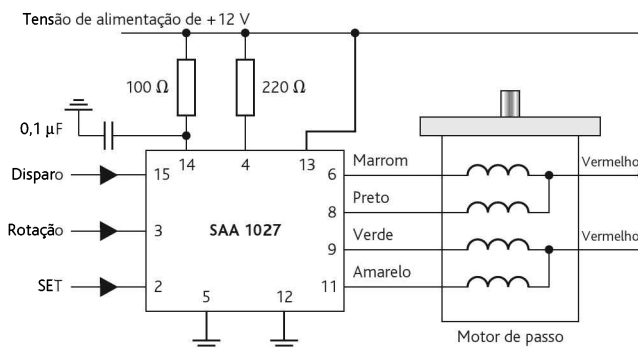


Figura 24.8 Circuito integrado SAA 1027 para acionamento de motor de passo.

Esta abordagem indica como podemos usar um motor de passo para efetuar uma rotação angular. Mas como um motor de passo se comporta quando lhe é dado um sinal de tensão de entrada? Podemos esperar que ele gire diretamente para o ângulo desejado sem *overshoot* e sem oscilação antes de estabilizar no ângulo desejado? Como uma ilustração de como podemos desenvolver um modelo para um sistema de motor de passo, e assim prever o seu comportamento, considere a seguinte análise simplificada (para uma análise mais detalhada, veja *Stepping Motors and their Microprocessor Controls* por T. Kenjo [Oxford University Press, 1984]).

O sistema que envolve um motor de passo acionado por pulsos de um microcontrolador é um sistema de controle em malha aberta. O motor de passo de ímã permanente (veja a Seção 9.7) tem um estator com um número de polos que são energizados por correntes que passam nas bobinas deles. Podemos determinar um modelo de como o rotor gira quando há um pulso de tensão de entrada considerando, para maior simplicidade, um motor de passo com apenas um par de polos e tratando-o da mesma forma que o motor CC que foi analisado na Seção 11.3.2. Se v for a tensão de alimentação do par de bobinas do motor e v_b for a FEM reversa, então

$$v - v_b = L \frac{di}{dt} + Ri$$

onde L é a indutância do circuito, R é a resistência e i é a corrente no circuito. Podemos considerar a simplificação de que a indutância não varia significativamente, e assim L é uma constante.

A FEM reversa será proporcional à taxa na qual o fluxo magnético varia no par de bobinas. Isto dependerá do ângulo θ do rotor em relação aos polos considerados. Portanto, podemos escrever

$$v_b = -k_b \frac{d\theta}{dt} \cos \theta = k_b \sin \theta \frac{d\theta}{dt}$$

em que k_b é uma constante. Portanto,

$$v - k_b \sin \theta \frac{d\theta}{dt} = L \frac{di}{dt} + Ri$$

Determinando a transformada de Laplace desta equação, obtemos

$$V(s) - k_b s \sin \theta \theta(s) = sL I(s) + R I(s) = (sL + R) I(s)$$

Assim como no motor CC, a corrente no par de bobinas gera um torque (o torque no ímã, ou seja, o rotor, é a reação resultante do torque exercido nas bobinas – terceira lei de Newton). O torque é proporcional ao produto da densidade de fluxo nas espiras da bobina e à corrente nelas. A densidade de fluxo depende da posição angular do rotor, e assim podemos escrever

$$T = k_t i \sin \theta$$

em que k_t é uma constante. Este torque provoca uma aceleração angular α e, visto que $T = J\alpha$, em que J é o momento de inércia do rotor,

$$T = J \frac{d^2\theta}{dt^2} = k_t i \sin \theta$$

Determinando a transformada de Laplace desta equação, obtemos

$$s^2 J \theta(s) = k_t \sin \theta I(s)$$

E assim podemos escrever

$$V(s) - k_b s \sin \theta \theta(s) = (sL + R)(s^2 J \theta(s)/k_t \sin \theta)$$

A função de transferência entre a tensão de entrada e o deslocamento angular resultante é

$$\begin{aligned} G(s) &= \frac{\theta(s)}{V(s)} = \frac{k_t \sin \theta}{J(sL + R)s^2 + k_b k_t s \sin^2 \theta} \\ &= \frac{1}{s} \times \frac{k_t \sin \theta}{JLs^2 + JRs + k_b k_t \sin^2 \theta} \end{aligned}$$

Quando há uma tensão de alimentação nas bobinas do motor, visto que um impulso unitário $V(s) = 1$,

$$\begin{aligned} \theta(s) &= \frac{1}{s} \times \frac{k_t \sin \theta}{JLs^2 + JRs + k_b k_t \sin^2 \theta} \\ &= \frac{1}{s} \times \frac{(k_t \sin \theta)/JL}{s^2 + (R/L)s + (k_b k_t \sin^2 \theta)/JL} \end{aligned}$$

A equação quadrática em s é da forma $s^2 + 2\zeta\omega_n s + \omega_n^2$ (veja a Seção 13.3.1) e, portanto, tem uma frequência natural ω_n de $\sqrt{(k_b k_t \sin^2 \theta)/JL}$ e um fator de amortecimento ζ de $(R/L)/2\omega_n$. O rotor gira para um certo ângulo e oscila em torno deste ângulo com oscilações amortecidas que se extinguem com o tempo.

24.1.3 Balanças de banheiro

Considere o projeto de uma máquina de pesar simples, como, por exemplo, uma balança de banheiro. Os principais requisitos são que a pessoa fique de pé sobre uma plataforma e o peso desta pessoa será mostrado. O peso deve ser dado com uma velocidade e precisão razoável e ser independente do local da plataforma em que a pessoa está de pé. As soluções possíveis podem envolver:

- 1 Um sistema mecânico puro baseado em uma mola e uma engrenagem.
- 2 Uma célula de carga e um sistema microprocessado/microcontrolado.

Uma solução possível é usar o peso da pessoa sobre a plataforma para defletir um arranjo de duas molas de folhas paralelas (Figura 24.9(a)). Com este arranjo, a deflexão é virtualmente independente do ponto da plataforma em que a pessoa está de pé. A deflexão pode ser transformada no movimento de um ponteiro sobre uma escala usando-se a configuração mostrada na Figura 24.9(b). Um arranjo de cremalheira e pinhão é usado para transformar o movimento linear em um movimento circular em torno de um eixo horizontal. Este movimento é transformado em uma

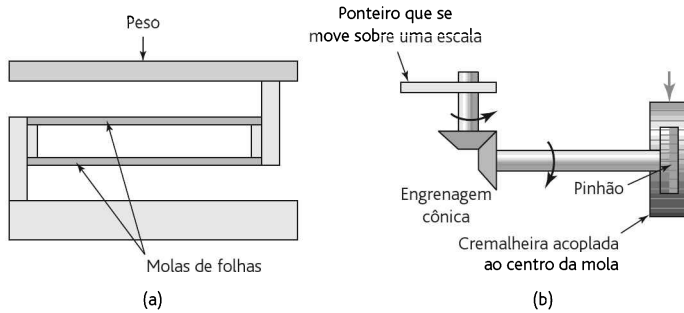


Figura 24.9 Balança de banheiro.

rotação de um eixo vertical, e portanto o movimento de um ponteiro sobre uma escala, por meio de uma engrenagem cônica.

Uma outra solução possível envolve o uso de um microprocessador. A plataforma pode ser montada sobre células de carga que empregam resistências *strain gauge*. Quando a pessoa está de pé sobre a plataforma, o *strain gauge* sofre um esforço variando a resistência. Se os *strain gauges* são montados nos quatro ramos de uma ponte de Wheatstone, então o desequilíbrio da tensão de saída da ponte é a medida do peso da pessoa. Este sinal pode ser amplificado por um amplificador operacional diferencial. O sinal analógico resultante pode ser entregue a um conversor analógico-digital cuja saída é enviada a um microprocessador, como, por exemplo, o 6820 da Motorola. A Figura 24.10 mostra a interface de entrada. É necessário também uma memória não apagável, que pode ser uma EPROM, como, por exemplo, a 2716 da Motorola. A saída para o display pode ser obtida de uma PIA, como, por exemplo, o 6821 da Motorola.

Entretanto, se for usado um microcontrolador, a memória faz parte do mesmo chip, e por meio de uma escolha adequada do microcontrolador, como, por exemplo, o M68HC11, teremos uma entrada para conversão analógico-digital. O sistema pas-

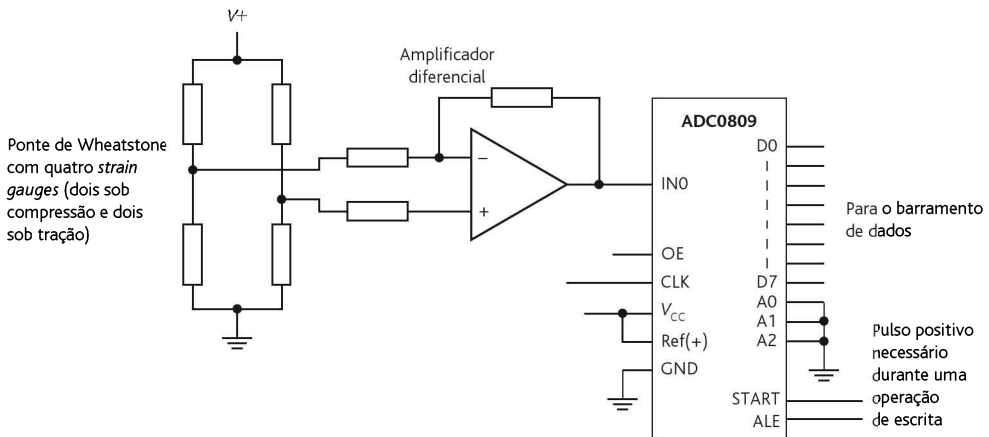


Figura 24.10 Interface de entrada.

sa a ser: *strain gauges* que, através de um amplificador operacional, enviam tensão para a porta E (a entrada o ADC) do microcontrolador, cuja saída passa por acionadores adequados, passando pelas portas B e C conectadas a decodificadores para displays de LEDs (Figura 24.11).

O programa pode ser como a seguir:

Inicialização zerando os displays de LEDs e a memória

Início

Alguém está sobre a balança? Se não, mostre 000

Se sim

receba a informação do peso

converta a informação do peso em um formato de saída adequado

envie esta informação para o decodificador do display de LEDs

atraso de tempo para reter a informação no display

Repita a partir do início para obter um novo peso

Ao considerar o projeto das partes mecânicas da balança de banheiro, precisamos considerar o que acontece quando alguém fica de pé na balança. Temos um sistema massa-mola-amortecedor comparável com o que está descrito na Figura 10.3(a) (veja a Seção 10.2.2), e assim podemos descrever o seu comportamento por meio de

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = F$$

em que x é a deflexão vertical da plataforma quando uma força F é aplicada. Determinando a transformada de Laplace, obtemos

$$ms^2 X(s) + cs X(s) + kX(s) = F(s)$$

assim o sistema pode ser descrito por uma função de transferência da forma

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{ms^2 + cs + k}$$

Podemos considerar uma pessoa de peso W de pé sobre a plataforma como uma entrada em degrau, e assim

$$X(s) = \frac{1}{ms^2 + cs + k} \times \frac{W}{s}$$

O termo quadrático é da forma $s^2 + 2\zeta\omega_n s + \omega_n^2$ (veja a Seção 13.3.1) e tem uma frequência natural ω_n de $\sqrt{k/m}$ e um fator de amortecimento ζ de $c/(2\sqrt{mk})$.

Quando uma pessoa fica de pé sobre a balança, ela quer que a indicação do peso seja rápida e não oscile por um longo tempo em torno do valor. Se o amortecimento for ajustado como crítico, a balança levará um longo tempo para indicar o peso, e desta forma é necessário que o amortecimento seja ajustado para permitir poucas oscilações que sejam rapidamente amortecidas. Podemos decidir que um tempo de estabilização (t_s) de 2% (veja a Seção 12.4) de, digamos, 4 s seja adequado. Visto que $t_s = 4/\zeta\omega_n$, então queremos que $\zeta\omega_n = 1$ e, assim, $\zeta = \sqrt{m/k}$. Uma forma simples de alterar o amortecimento é alterando a massa.

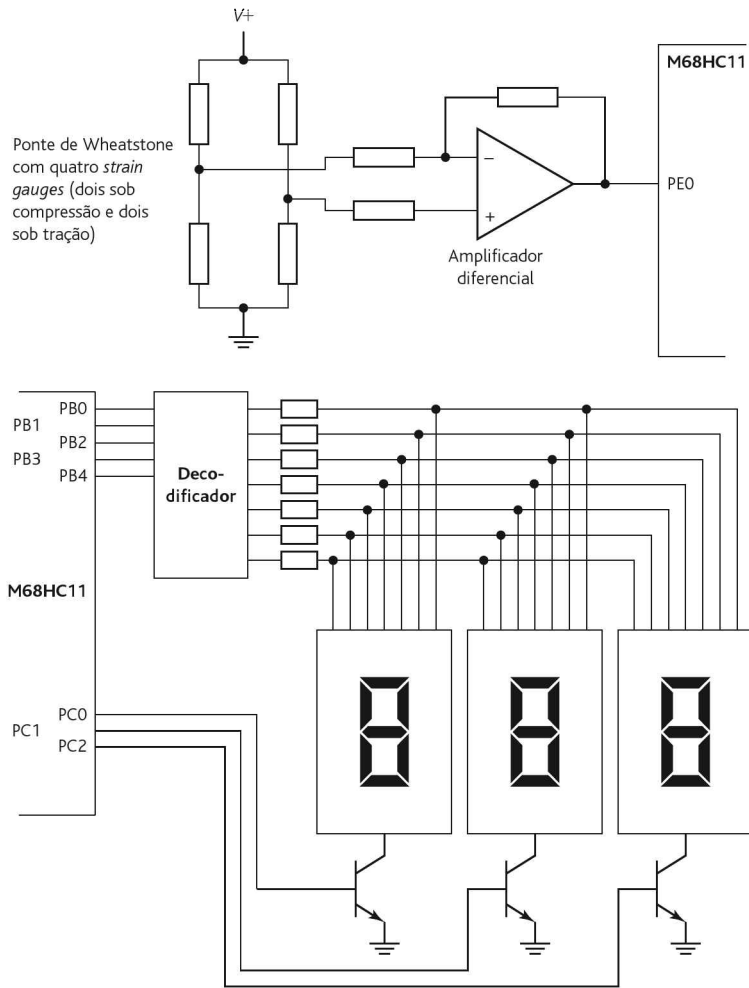


Figura 24.11 Um circuito de balança de banheiro.

Este exemplo indica como podemos usar um modelo matemático para prever o comportamento de um sistema e que fatores podemos alterar para melhorar seu desempenho.

24.2 Estudos de caso

A seguir abordamos de forma resumida exemplos de sistemas mecatrônicos.

24.2.1 Robô de manipulação

A Figura 24.12(a) mostra a forma básica de um robô de manipulação. Este robô tem três eixos em torno dos quais os movimentos podem ocorrer: rotação no sentido

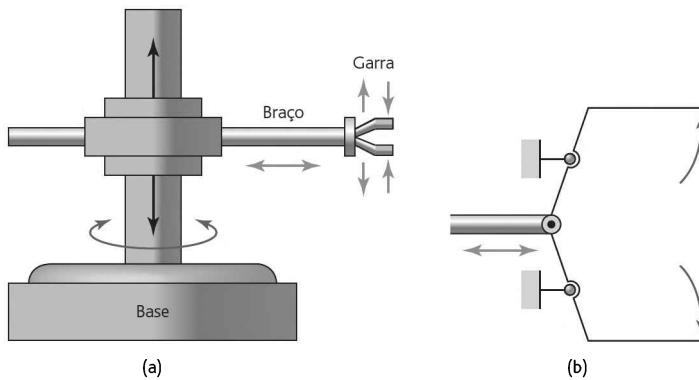


Figura 24.12 (a) Robô de manipulação, (b) garra.

horário ou anti-horário da unidade em sua base, avanço ou recuo e subida ou descida do braço; além de poder abrir e fechar uma garra. Estes movimentos podem ser acionados com o uso de cilindros pneumáticos ativados por solenoides controlados por válvulas com chaves fim de curso para indicar quando o movimento termina. Portanto, a rotação no sentido horário da base pode resultar de um pistão em um cilindro sendo estendido, e o movimento no sentido anti-horário, como resultado da retração do mesmo. Da mesma forma, o movimento do braço para cima pode resultar de um pistão em um cilindro linear sendo estendido, e o movimento para baixo, de quando o pistão é retraído; o avanço do braço pode ser resultado de um pistão em outro cilindro que é estendido, e o recuo, resultado da retração do mesmo. O movimento de abrir e fechar a garra pode ser o resultado da extensão e retração de um pistão em um cilindro linear. A Figura 24.12(b) mostra um mecanismo básico que poderia ser usado.

Um exemplo de um programa típico para este robô pode ser este:

- 1 Fechar a garra sobre um componente suspendendo-o a partir da parte superior no alimentador
- 2 Contrair o braço de modo que o componente seja retirado do alimentador.
- 3 Girar o braço no plano horizontal de modo que ele seja posicionado na direção da peça a ser trabalhada.
- 4 Avançar o braço de modo que a garra fique sobre a peça a ser trabalhada.
- 5 Girar o pulso de modo que o componente suspenso possa descer a partir da garra.
- 6 Liberar a garra de modo que o componente caia na posição desejada.
- 7 Girar a garra para a posição vertical.
- 8 Recuar o braço.
- 9 Girar o braço para apontar na direção do alimentador.

Repetir a sequência para o próximo componente.

A Figura 24.13 mostra como um microcontrolador poderia ser usado para controlar válvulas solenoide e, portanto, os movimentos do robô.

Êmbolos pneumáticos e hidráulicos são bastante usados para acionar braços de robôs, visto que eles podem ser facilmente controlados para mover as partes móveis em velocidades relativamente baixas, enquanto que os motores elétricos necessitariam operar através de caixas de engrenagens redutoras.

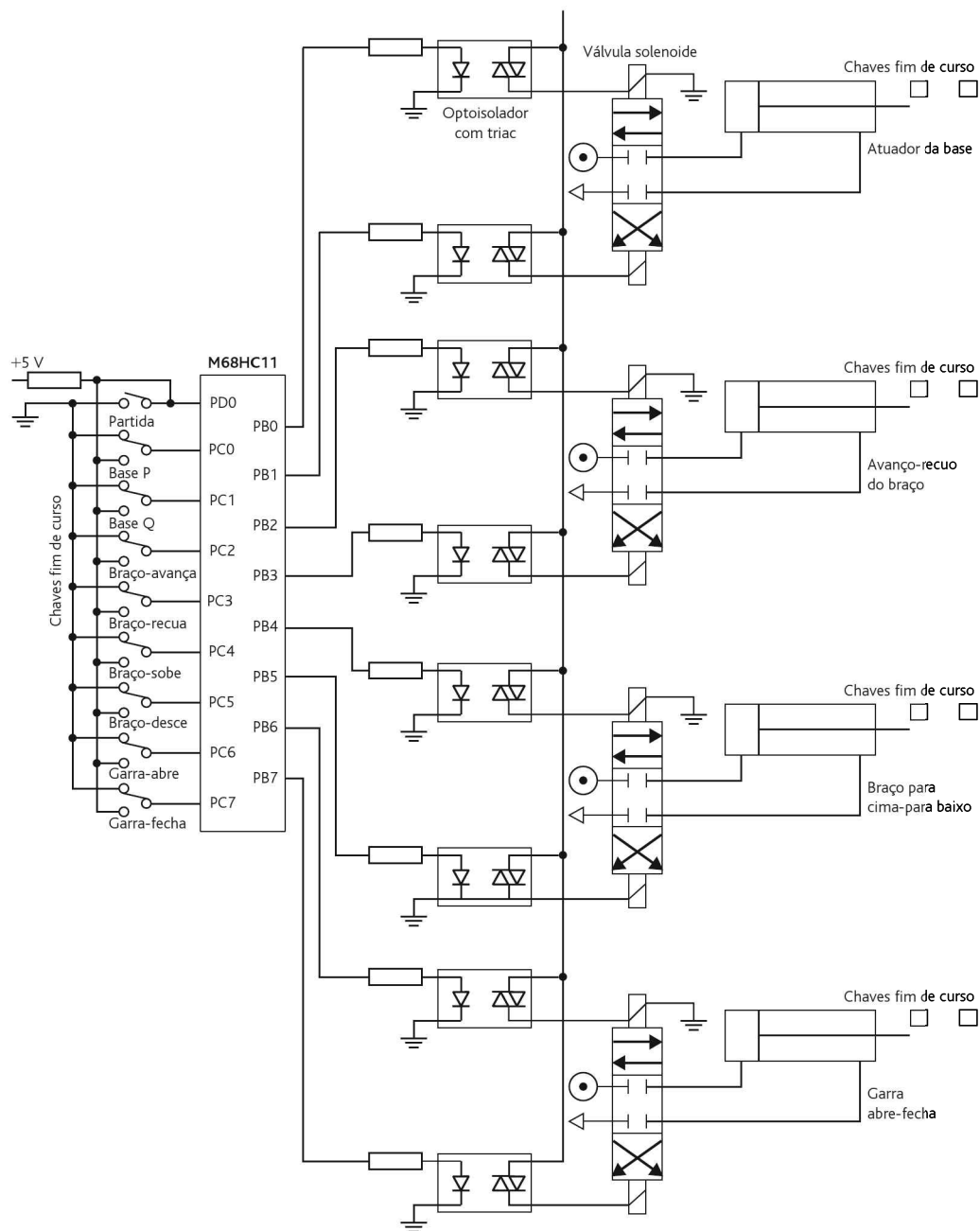


Figura 24.13 Controle do robô.

As posições do braço e da garra na Figura 24.13 são determinadas pelas chaves fim de curso. Isto significa que apenas duas posições podem ser atingidas com precisão em cada atuador e as posições não podem ser prontamente alteradas sem que ocorra o acionamento físico das chaves. A configuração é de um sistema de controle em malha aberta. Em algumas aplicações, isto pode não ser um problema.

Entretanto, é mais comum usar controle em malha fechada com as posições do braço e da garra monitoradas por sensores e realimentadas para serem comparadas no controlador com as posições desejadas. Quando há diferenças a partir das posições desejadas, o controlador opera o atuadores para reduzir o erro. A posição angular de uma articulação é geralmente monitorada usando-se um encoder (veja a Seção 2.3.7), que é capaz de proporcionar alta precisão. A Figura 24.14 mostra uma configuração em malha fechada que pode ser usada para o movimento linear do braço de um robô.

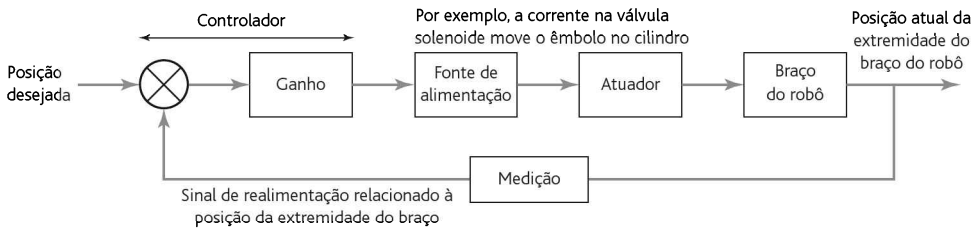


Figura 24.14 Controle de um braço de robô em malha fechada.

A saída do atuador é uma força F aplicada para mover a extremidade do braço robô. Sendo a posição desejada y_s e a posição atual y , o sinal de erro é $y_s - y$, considerando, para maior simplicidade, que o sistema de medição tem um ganho de 1. Se considerarmos o controlador com ganho G_c e o atuador com ganho G_a , então $F = G_c G_a (y_s - y)$. As massas a serem aceleradas por esta força são as massas da carga que o braço transporta, do braço e das partes móveis do atuador. Sendo esta massa total m , então a lei de Newton resulta em $F = ma$, em que a aceleração a pode ser escrita como d^2y/dt^2 . Entretanto, isto não leva em conta o atrito, e como podemos ter a força de atrito proporcional à velocidade, esta força é $k dy/dt$. Portanto, podemos escrever

$$F = G_c G_a (y_s - y) = m \frac{d^2y}{dt^2} + k \frac{dy}{dt}$$

e assim

$$y_s = \frac{m}{G_c G_a} \frac{d^2y}{dt^2} + \frac{k}{G_c G_a} \frac{dy}{dt} + y$$

Esta é uma equação diferencial de segunda ordem, e portanto a deflexão y será como descrita na Seção 13.3.1 e a sua forma dependerá do fator de amortecimento. Um sistema sub-amortecido tem uma frequência angular natural ω_n dada por

$$\omega_n = \sqrt{\frac{G_c G_a}{m}}$$

Esta frequência angular determina a rapidez com que o sistema responde à mudanças (veja a Seção 12.5): quanto maior a frequência angular, mais rápido o sistema responde (o tempo de subida é inversamente proporcional à frequência angular). Isto significa que, aumentando-se o ganho do controlador ou diminuindo-se a massa, pode-se aumentar a velocidade de resposta. O fator de amortecimento ζ é dado a partir da equação diferencial como

$$\zeta = \frac{k}{2\sqrt{G_c G_a m}}$$

O tempo que as oscilações gastam para se extinguir, ou seja, o tempo de estabilização (veja a Seção 12.5), é inversamente proporcional ao fator de amortecimento e, por exemplo, aumentar qualquer parte da massa resulta em uma diminuição no fator de amortecimento, e assim as oscilações gastam mais tempo para se extinguir.

24.2.2 Cancela em estacionamento de carros

Como uma ilustração do uso de um CLP, considere a cancela acionada por moeda em um estacionamento de carros. A cancela de entrada deve abrir quando o valor correto em dinheiro for inserido na caixa coletora, e a cancela de saída deve abrir quando um carro for detectado na saída do estacionamento. A Figura 24.15 mostra os tipos de sistemas de válvulas que podem ser usados para levantar e abaixar as cancelas articuladas.

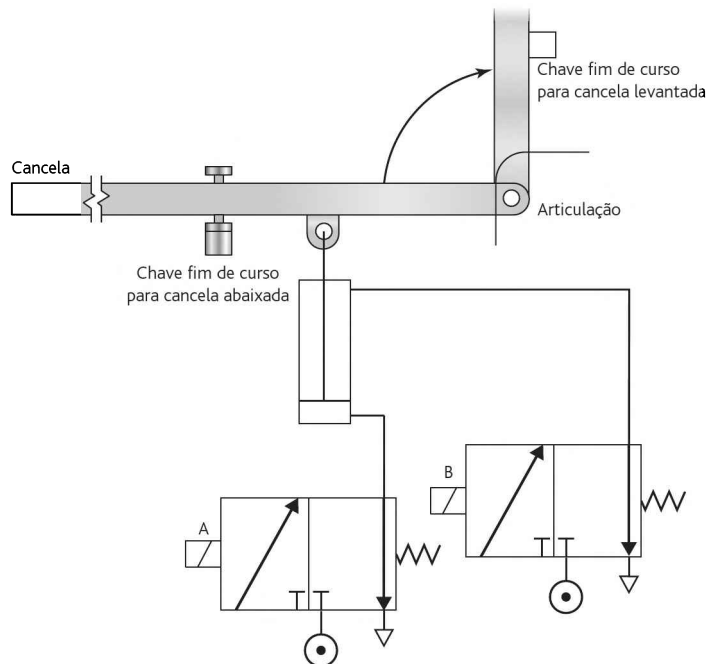


Figura 24.15 Sistema para levantar e abaixar um cancela.

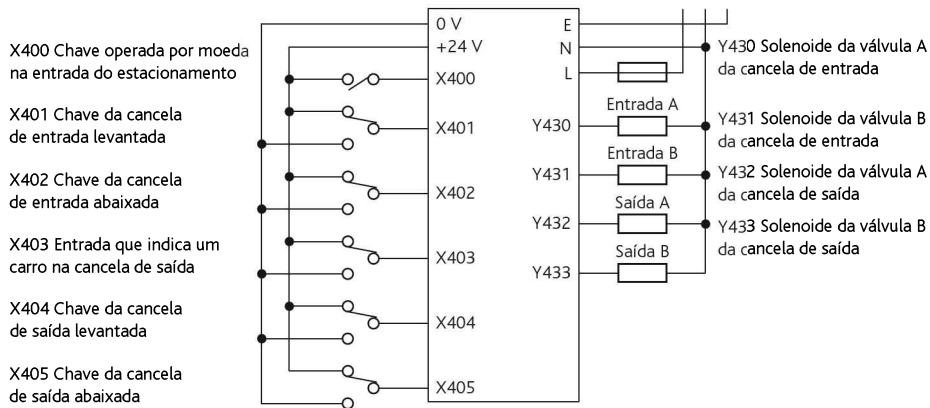


Figura 24.16 Conexões do CLP.

Quando uma corrente flui através do solenóide da válvula A, o pistão no cilindro se move para cima e faz com que a cancela gire em torno da articulação e levante para permitir que o carro passe. Quando a corrente no solenóide da válvula A cessa, a mola de retorno da válvula faz com que a válvula volte para a posição original. Quando uma corrente passa pelo solenóide B, a pressão é aplicada para abaixar a cancela. As chaves fim de curso são usadas para detectar quando a cancela está totalmente abaixada, bem como quando está totalmente levantada.

Com dois sistemas como o que é mostrado na Figura 24.15, um para a cancela de entrada e outro para a de saída, e as conexões de entrada e saída do CLP mostradas na Figura 24.16, o programa ladder pode ser como o mostrado na Figura 24.17.

24.2.3 Câmera digital

Uma câmera digital captura imagens e as armazena no formato digital em um cartão de memória, diferentemente das câmeras de filme anteriores, em que as imagens eram armazenadas no formato analógico como uma variação química sobre um filme. A Figura 24.18 mostra os elementos básicos de uma câmera digital relativamente barata.

Quando o fotógrafo pressiona o botão do obturador no primeiro estágio, o de pressão parcial, um microcontrolador calcula a velocidade do obturador, seta a abertura a partir da entrada do sensor de distância e a indica na tela de LCD. No mesmo instante, o microcontrolador processa a entrada a partir do sensor de alcance e envia sinais para acionar um motor que ajusta o foco das lentes. Quando o fotógrafo pressiona o obturador até o segundo estágio, o de pressão total, o microcontrolador emite sinais para mudar a abertura para a desejada, abre o obturador por um tempo de exposição desejado, e então, quando o obturador fecha, processa a imagem recebida pelo sensor de imagem e a armazena no cartão de memória. Além disto, quando o botão do obturador é parcialmente pressionado, o sistema de controle do foco automático é usado para mover as lentes de modo que a imagem seja focalizada (veja a Seção 1.7.1 para ver detalhes dos mecanismos do foco automático e, mais adiante nesta seção, veja uma discussão do motor usado para mover as lentes).

A luz proveniente do objeto fotografado passa pelo sistema de lentes e é focalizada no sensor de imagem. Este é tipicamente um dispositivo de acoplamento de carga (CCD – *Charge-Coupled Device*) (veja a Seção 2.10) que consiste de um

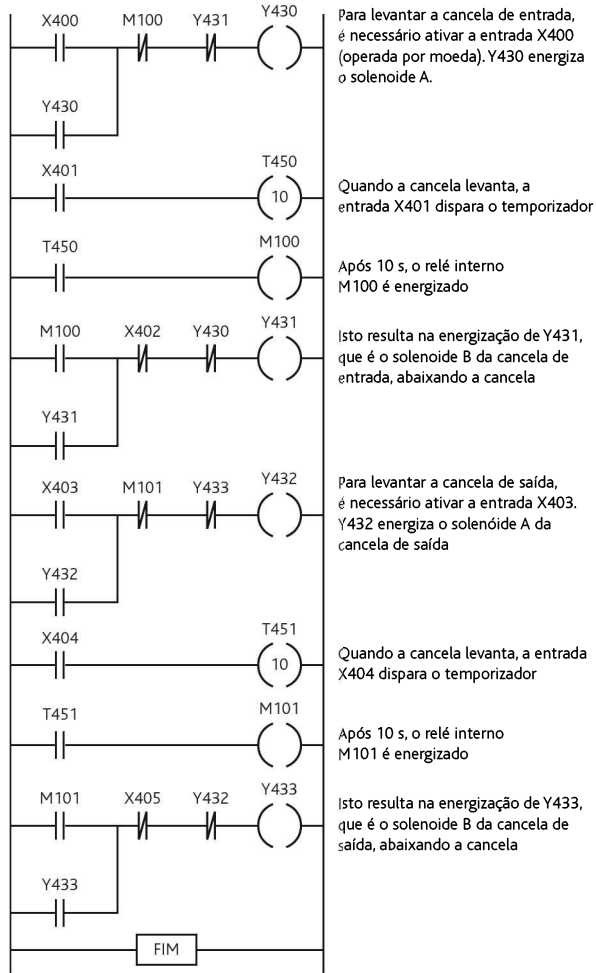


Figura 24.17 Programa ladder.

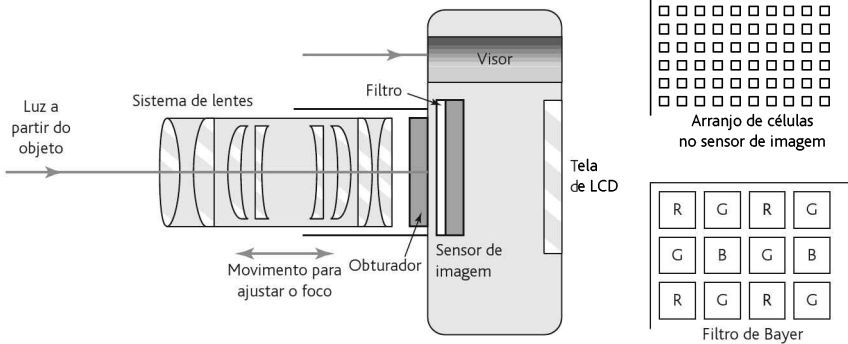


Figura 24.18 Elementos básicos de uma câmera digital.

arranjo de células sensitivas muito pequenas, denominadas pixels, que são expostas à luz que passa pelas lentes quando o obturador eletromecânico é aberto por um breve intervalo de tempo. A luz que incide nas células é convertida em uma pequena quantidade de carga elétrica que, quando a exposição é finalizada, é lida e armazenada em um registro antes de ser processada e armazenada no cartão de memória.

Os sensores são de tonalidade escura, e assim, para que as fotos coloridas possam ser produzidas, uma matriz de filtro de cor está situada antes do arranjo de células. Existem filtros separados, azul, verde ou vermelho, para cada célula. O projeto de matriz mais comum é o arranjo de Bayer. Ele tem as três cores organizadas em um padrão, de forma que dois filtros da mesma cor ficam próximos um do outro e existem duas vezes mais filtros verdes do que vermelhos ou azuis. Isto porque o verde está mais ou menos no centro do espectro visível e apresenta mais detalhes. O resultado é que este estágio é um mosaico de pixels vermelhos, verdes e azuis. Os arquivos dos resultados dos pixels neste estágio são denominados arquivos RAW (crus), já que nenhum processamento foi feito neles. Para fornecer o espectro total de cores para um pixel em particular, é usado um algoritmo em que a cor a ser alocada para um pixel em particular é determinada levando-se em conta as intensidades das cores nos pixels vizinhos.

O próximo estágio no processamento do sinal é a compressão dos arquivos de modo que eles ocupem o menor espaço de memória possível. Desta forma, mais dados podem ser armazenados no cartão de memória em comparação com os arquivos RAW. Geralmente, o arquivo comprimido é no formato JPEG, que são as iniciais de Joint Photographic Experts Group. A compressão JPEG usa o princípio de que, em muitas fotografias, muitos dos pixels na mesma área são idênticos, e assim, em vez de armazenar a mesma informação para cada um, basta armazenar efetivamente uma e informar que os outros são repetidos.

A exposição necessária é determinada por um microcontrolador em resposta à saída de um sensor, como um fotodiodo que detecta a intensidade de luz. Ele fornece as saídas que são usadas para controlar a abertura e a velocidade do obturador. O sistema de acionamento da abertura em uma câmera digital pode ser um motor de passo que abre ou fecha um conjunto de palhetas do diafragma de acordo com

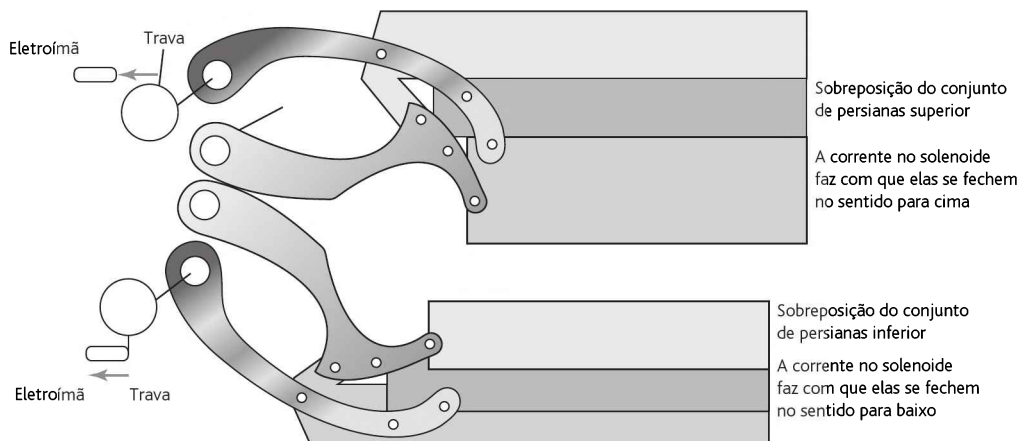


Figura 24.19 Mecanismo do obturador.

o sinal recebido do microcontrolador. O mecanismo do obturador usado em uma câmera digital é geralmente da forma mostrada na Figura 24.19. O obturador envolve dois conjuntos de cortinas, cada uma controlada por uma trava com mola. Na ausência de uma corrente no eletroímã, a mola força a trava para a posição na qual o conjunto superior de cortinas se sobrepõe ao conjunto inferior. Quando uma corrente passa no eletroímã, ela faz com que a trava gire, levantando o conjunto superior de cortinas. O conjunto inferior de cortinas é mantido inicialmente em baixo através do eletroímã que mantém a trava. Quando a corrente através da trava da cortina inferior é desligada, a cortina sobe. Portanto, a abertura do orifício sobre o sensor de imagem é determinada pelo tempo entre a comutação de uma corrente na trava superior e o desligamento dela na trava inferior.

O foco necessita de um mecanismo para mover as lentes. Este mecanismo frequentemente é um motor ultrassônico que consiste de uma série de elementos piezoelétricos, como o titanato zirconato de chumbo (PZT – *lead zirconium titanate*). Quando uma corrente é fornecida a um elemento piezoelétrico, este expande ou contrai de acordo com a polaridade da corrente (Figura 24.20(a)). Os elementos PZT são fixados nos dois lados de uma fina tira de aço com ação de mola, e quando uma diferença de potencial é aplicada na tira, a única forma do elemento PZT expandir ou contrair é dobrando a tira metálica (Figura 24.20(b)). Quando é aplicada a polaridade oposta para alternar os elementos, eles são forçados a dobrar em sentidos opostos (Figura 24.20(c)). Portanto, usando uma corrente alternada com uma sequência destes elementos em torno de um anel, uma onda pode se deslocar em torno do anel piezoelétrico de elementos no sentido horário ou anti-horário. A amplitude desta onda de deslocamento é de apenas cerca de 0,001 mm. Existe uma tira de material com minúsculos dentes fixados na parte externa dos elementos PZT, e quando a onda de deslocamento se move em torno dos elementos PZT, eles são capazes de empurrar as lentes montadas (Figura 24.20(d)) e, assim, acionar o elemento de foco.

Como uma aplicação do uso das técnicas de modelamento discutidas nos capítulos anteriores, considere este motor ultrassônico. O torque T gerado pelo motor é necessário para girar o anel dele em alguma posição angular θ . O anel é muito leve, e assim podemos desprezar sua inércia em comparação com o atrito entre os anéis.

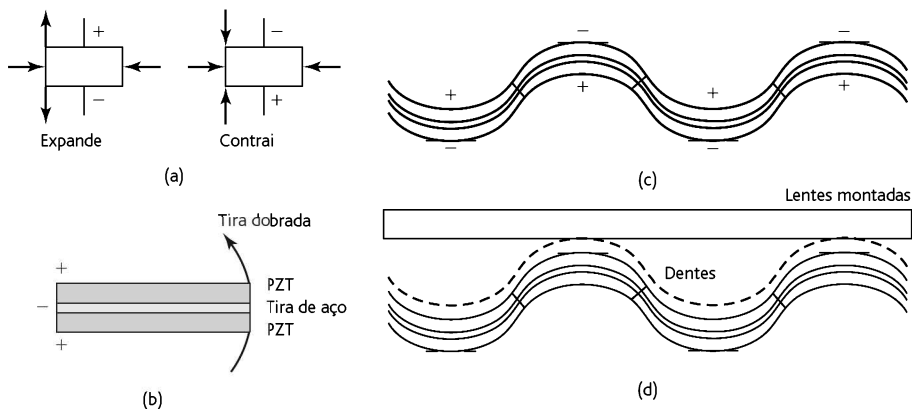


Figura 24.20 Motor piezoelétrico.

Considerando que a força de atrito é proporcional à velocidade angular ω , então $T = c\omega = c \, d\theta/dt$, em que c é uma constante de atrito. A integração resulta em

$$\theta = \frac{1}{c} \int dt$$

e assim em uma função de transferência $G(s)$ de $1/cs$.

O sistema de controle para o motor ultrassônico é da forma mostrada na Figura 24.21. y_n é o pulso de entrada de ordem n e x_n é o pulso de saída de ordem n . Com um microprocessador executando o controle proporcional de ganho K , a entrada dele é $y_n - x_n$ e a saída é $K(y_n - x_n)$. Este sinal passa pelo DAC para gerar um sinal analógico de saída que consiste de um número de degraus (Figura 24.21). O motor age como um integrador, e assim sua saída é $1/c$ vezes a soma progressiva das áreas sob os degraus (Figura 24.21). Cada degrau tem uma área de (variação do DAC na saída para o degrau) $\times T$. Portanto,

$$x_n - x_{n-1} = (\text{saída do DAC para } x_{n-1})T/c = K(y_{n-1} - x_{n-1})T/c$$

Assim,

$$x_n = [1 - (KT/c)]x_{n-1} + (KT/c)y_{n-1}$$

Suponha que temos $K/c = 5$ e um intervalo de amostragem de 0,1 s. Portanto,

$$x_n = 0,5y_{n-1} + 0,5x_{n-1}$$

Se houver uma entrada para o sistema de controle do foco como uma seqüência de pulsos de amplitude constante 1, antes disto não havendo entrada, ou seja, $y_0 = 1$, $y_1 = 1, y_2 = 1, \dots$, então

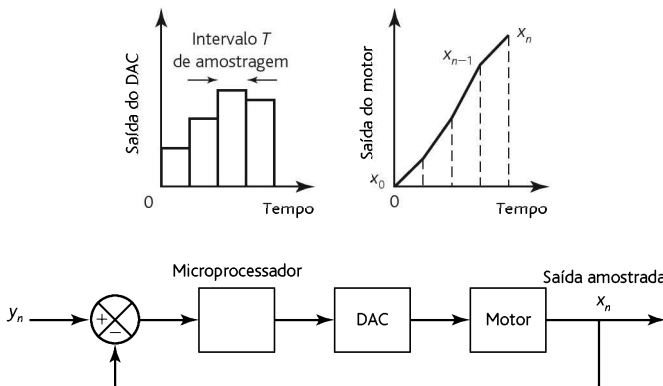


Figura 24.21 Sistema de controle.

$$\begin{aligned}
 x_0 &= 0 \\
 x_1 &= 0,5 \times 0 + 0,5 \times 1 = 0,5 \\
 x_2 &= 0,5 \times 0,5 + 0,5 \times 1 = 0,75 \\
 x_3 &= 0,5 \times 0,75 + 0,5 \times 1 = 0,875 \\
 x_4 &= 0,5 \times 0,875 + 0,5 \times 1 = 0,9375 \\
 x_5 &= 0,5 \times 0,9375 + 0,5 \times 1 = 0,96875 \\
 x_6 &= 0,5 \times 0,96875 + 0,5 \times 1 = 0,984375 \\
 x_7 &= 0,5 \times 0,984365 + 0,5 \times 1 = 0,9921875
 \end{aligned}$$

E assim por diante

Portanto, a saída leva cerca de sete períodos de amostragem, ou seja, 0,7 s, para que o foco seja alcançado. Este tempo é muito longo. Entretanto, suponha que escolhamos valores de forma que $KT/c = 1$. A equação de diferença torna-se então $x_n = y_{n-1}$. Então temos

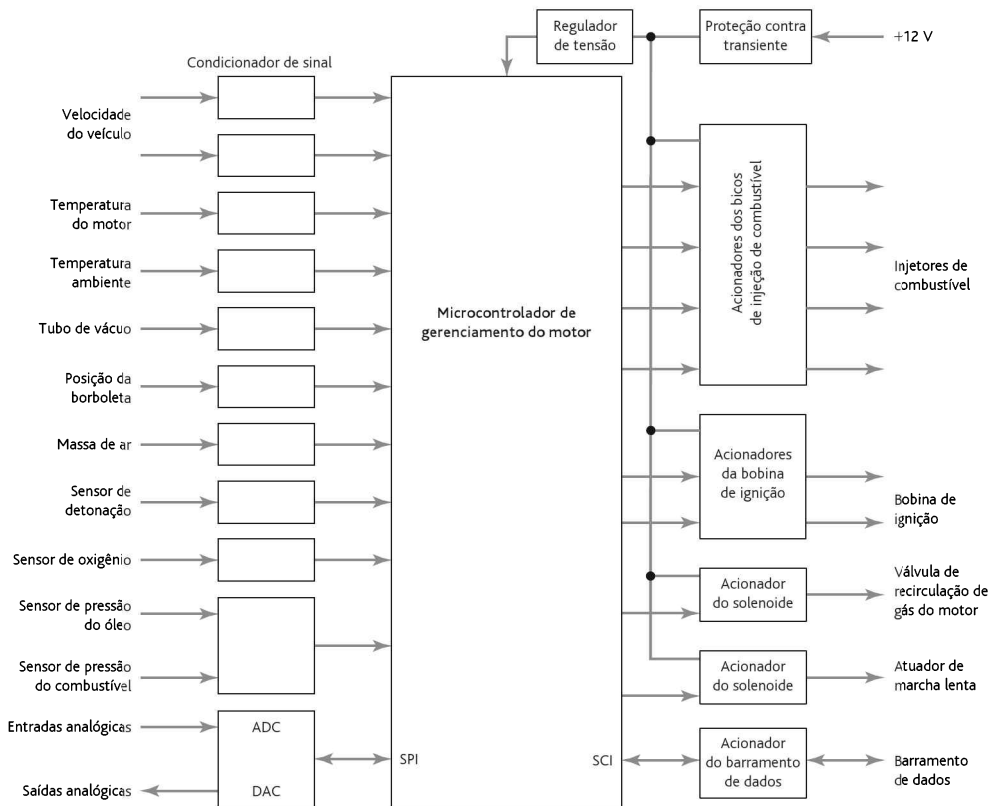


Figura 24.22 Sistema de gerenciamento do motor.

$$\begin{aligned}x_0 &= 0 \\x_1 &= 1 \\x_2 &= 1 \\x_3 &= 1 \\&\dots\end{aligned}$$

Isto significa que a saída alcança a posição desejada após uma amostra apenas. Esta resposta é bem mais rápida. Usando-se uma taxa de amostragem alta, pode ser conseguida uma resposta muito rápida. Esta forma de resposta é denominada **resposta aperiódica**.

24.2.4 Gerenciamento do motor de um carro

Os carros modernos estão incorporando muitos sistemas de controle eletrônico que envolvem microcontroladores, como no sistema de controle do motor, com o objetivo de garantir que o motor opere na melhor condição. A Figura 24.22 mostra um diagrama em bloco genérico de um sistema deste. O sistema consiste de sensores gerando, após adequado condicionamento do sinal, os sinais de entrada para o microcontrolador e este provendo os sinais de saída via circuitos acionadores para os atuadores. A Figura 24.23 mostra alguns destes elementos em um sistema de gerenciamento do motor, sendo que é mostrado apenas um cilindro.

O sensor de velocidade do motor é indutivo e consiste de uma bobina na qual a indutância varia conforme os dentes de uma roda passam em frente ao sensor,

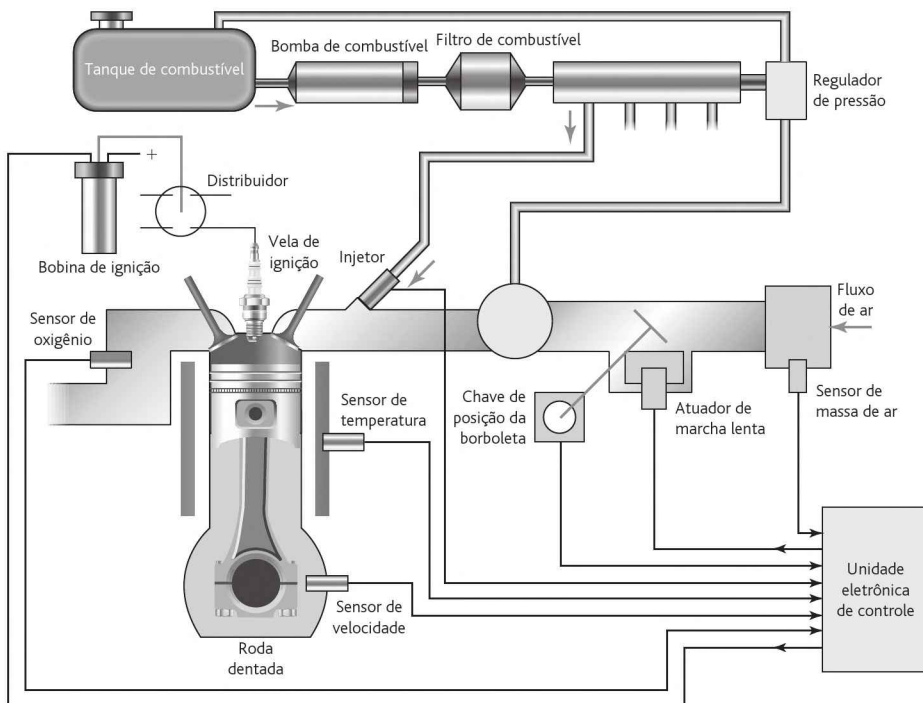


Figura 24.23 Sistema de gerenciamento de um motor de combustão.

gerando uma tensão oscilante. O sensor de temperatura é geralmente um termistor. O sensor de massa de ar pode ser um fio aquecido; à medida que o ar passa pelo fio aquecido, ele esfria, e o quanto ele esfria depende da velocidade da massa de ar. O sensor de oxigênio é geralmente um tubo com uma extremidade fechada feito de óxido de zircônio com eletrodos de platina porosos nas superfícies interna e externa. Acima de cerca de 300°C, o sensor se torna permeável aos íons de oxigênio, fornecendo assim uma tensão entre seus eletrodos.

24.2.5 Leitor de códigos de barras

Uma cena familiar no caixa de um supermercado é ver as mercadorias sendo passadas em frente a um feixe de luz ou então a caixa usando um equipamento manual que ela passa sobre as mercadorias, de forma que o código de barras possa ser lido e assim, a natureza da mercadoria, e por consequência o seu preço possa ser automaticamente determinado. Este código consiste de uma série de barras pretas e brancas de larguras variadas. Por exemplo, existe um código de barras como este na parte posterior deste livro.

A Figura 24.24 mostra a forma básica do código de barras usado no comércio varejista. O código de barras representa uma série de números. Existe um prefixo que identifica o esquema do código usado; este é um único dígito para o esquema de codificação de produtos universal (UPC – *Universal Product Coding*) usado nos Estados Unidos e dois dígitos para o esquema do número de artigo europeu (EAN – *European Article Number*) usado na Europa. O código UPC usa um prefixo 0 para artigos de mercearia e um 3 para artigos farmacêuticos. O prefixo EAN vai de 00 a 09 e, é de tal forma que o código UPC, pode ser lido dentro do código EAN. Isto é seguido por cinco dígitos para representar o fabricante, sendo cada fabricante associado a um único número. Neste ponto temos o centro do código, que é identificado por duas barras maiores. Os cinco dígitos posteriores representam o produto. O número final é um dígito de verificação que é usado para verificar que o código foi corretamente lido. Duas barras maiores são usadas no início e final do padrão de barras para delimitar o código.

Cada número é codificado como sete dígitos 0 ou 1. Os códigos usados em cada lado da linha central são diferentes, de modo que a direção da varredura pode ser determinada. Os caracteres à direita têm um número par de uns e, assim, uma paridade par, e, para o UPC, têm à esquerda um número ímpar de uns e, assim, uma paridade ímpar. Já o código EAN para a esquerda é uma mistura. A Tabela 24.2



Figura 24.24 Código de barras.

Tabela 24.2 Codificação UPC e EAN

Número decimal	Caracteres A à esquerda	Caracteres B à esquerda	Caracteres à direita
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	0011100
5	0110001	0111001	0001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

mostra os códigos UPC e EAN, sendo o UPC uma codificação A à esquerda e o EAN usando códigos de caracteres A à esquerda e B à esquerda.

Cada 1 é inserido como uma barra escura, e portanto à direita do caractere 2 seria representado por 1101100. Com as barras escuras adjacentes, a aparência é uma barra escura com largura dupla seguida de um espaço estreito e então uma outra barra escura dupla seguida por um espaço duplo. Esta representação é ilustrada na Figura 24.25. O padrão que delimita as extremidades do código representa 101 e a banda central de barras é 0100.

O código de barras mostrado na Figura 24.24 é o da primeira edição deste livro. Ele usa o código EAN e tem o prefixo 97 para identificá-lo como uma publicação, 80582 para identificar a editora, 25634 para identificar este livro em particular e o dígito de verificação 7. Note que o código de barras contém as partes relevantes do número ISBN, que identifica a editora e o livro.

O procedimento para o uso do dígito do código de verificação é:

- 1 Começando à esquerda, some todos os caracteres, excluindo o dígito de verificação, nas posições ímpares, ou seja, primeira, terceira, quinta, etc., e então multiplique a soma por 3.
- 2 Começando à esquerda, some todos os caracteres nas posições pares.
- 3 Some os resultados dos passos 1 e 2. O caractere de verificação é o menor número que, quando somado a este resultado, produz um múltiplo de 10.

Como uma ilustração do uso do dígito verificador, considere o código de barras 9780582256347. Para os caracteres ímpares, temos $9 + 8 + 5 + 2 + 5 + 3 = 32$, e quando multiplicado por 3, temos 96. Para os caracteres pares, temos $7 + 0 + 8 + 2 + 6 + 4 = 27$. A soma é 123 e o dígito verificador deve ser 7.

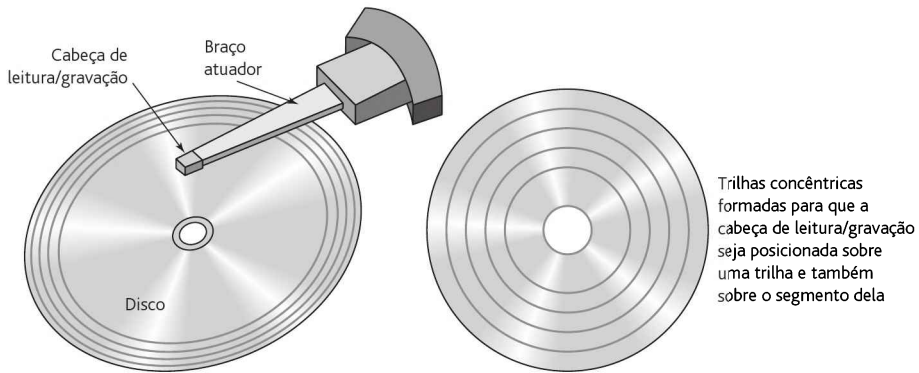
A leitura do código de barras envolve a determinação da largura das listas escuras e claras. Isto pode envolver um feixe estreito de um laser de estado sólido sendo usado para incidir sobre o código e a luz refletida sendo detectada por uma fotocélula. Geralmente a versão de scanner usada no supermercado é fixa e um espelho de rotação é usado para direcionar a luz sobre o código de barras e percorrer todas as barras. O con-



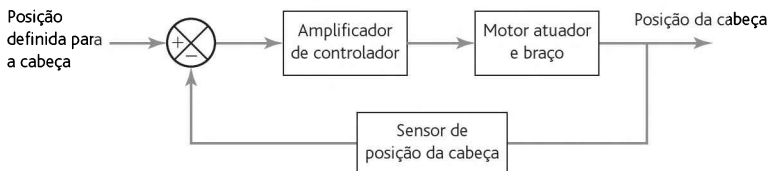
Figura 24.25 Código de barras à direita de 2.

dicionamento de sinal envolve a amplificação da saída da fotocélula usando amplificadores operacionais e usando um circuito com amplificador operacional como comparador para gerar um nível alto, ou seja, 1, na saída quando uma barra escura é identificada, e um nível baixo, ou seja, 0, quando for um espaço em branco. A sequência de zeros e uns pode então ser inserida, por exemplo, em uma PIA conectada a um microprocessador 6800 da Motorola. Em termos gerais, o programa do microprocessador é:

- 1 Inicializar zerando as diversas posições de memória.
- 2 Recuperar os dados da entrada. Isto envolve testar continuamente a entrada para determinar se ela é 0 ou 1.
- 3 Processar os dados para obter os caracteres no formato binário. A entrada é um sinal serial que consiste de zeros e uns de durações diferentes que dependem da largura das barras de espaços das listas escuras. O sistema microprocessado é programada para determinar a largura de tempo do módulo dividindo o tempo de varredura entre as barras das extremidades pelo número de módulos. Um módulo é uma lista clara ou escura que representa um único 0 ou 1. O programa pode então determinar se um lista escura ou clara é um único dígito ou mais de um e, portanto, interpretar o sinal lido.
- 4 Processar o resultado binário em uma descrição do item comprado e o preço.



(a)



(b)

Figura 24.26 Disco rígido: (a) constituição simplificada, (b) sistema de controle em malha fechada simplificado para o posicionamento da cabeça de leitura/gravação.

24.2.6 Acionador de disco rígido

A Figura 24.26(a) mostra a constituição simplificada do acionamento de um disco rígido. Ele consiste de um disco coberto com uma camada de metal que pode ser magnetizado. O espaço entre a cabeça de leitura/gravação e a superfície do disco é muito pequeno, cerca de $0,1 \mu\text{m}$. Os dados são armazenados na camada de metal como uma sequência de células de bit (veja a Seção 6.3.2). O disco é rotacionado por um motor em velocidades típicas de 3600, 5400 e 7200 ver/min, e um braço atuador é posicionado de forma que as trilhas concêntricas relevantes e as partes relevantes das trilhas fiquem sob a cabeça de leitura/gravação na extremidade do braço. A cabeça é controlada por um sistema de malha fechada (Figura 24.26(b)) para posicioná-la. As informações de controle são escritas no disco durante o processo de formatação, o qual permite que cada trilha e setor da trilha seja identificado. O processo de controle envolve o uso desta informação para posicionar a cabeça de leitura/gravação na parte desejada do disco.

O movimento do atuador geralmente envolve um atuador de bobina de voz (Figura 24.27) para girar o braço. Este atuador de bobina de voz é essencialmente uma bobina, montada em um núcleo de ferro de modo que, quando uma corrente passa na bobina, ela se move de forma semelhante ao movimento da bobina de um alto-falante, e assim ela é capaz de mover o braço atuador para posicionar a cabeça de leitura/gravação sobre a trilha desejada. A cabeça lê o campo magnético no disco e fornece um sinal de realimentação para o amplificador de controle.

O atuador da bobina de voz é uma forma de motor CC de ímã permanente controlado pelo campo e tem uma função de transferência da mesma forma (veja a Seção 13.5). Portanto, como estamos interessados na função de transferência que relaciona o movimento com o tempo, ou seja, a integral da velocidade em função do tempo dada na Seção 13.5, o atuador da bobina de voz tem uma função de transferência da forma

$$G(s) = \frac{k}{s(Ls + R)(Is + c)} = \frac{k/Rc}{s(\tau_I s + 1)(\tau_s + 1)}$$

O termo $(\tau_s + 1)$ é geralmente próximo de 1, e assim a função de transferência se aproxima de

$$G(s) = \frac{k/Rc}{s(\tau_I s + 1)}$$

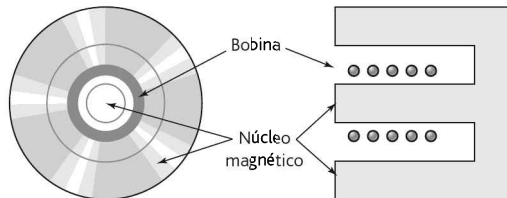


Figura 24.27 Atuador da bobina de voz.

Portanto, o sistema de controle em malha fechada na Figura 24.26(a), com um amplificador de controle que tem um ganho proporcional de K_a e a transferência de posicionamento da cabeça tendo um ganho de 1, deve ter uma função de transferência geral que fornece a relação entre o sinal de saída $X(s)$ e o sinal desejado de entrada $R(s)$ de

$$\frac{X(s)}{R(s)} = \frac{K_a G(s)}{1 + K_a G(s)}$$

Portanto, se tivermos, por exemplo, $G(s) = 0,25/s(0,05s + 1) = 5/s(s + 20)$ e $K_a = 40$, então

$$X(s) = \frac{200}{s^2 + 20s + 200} R(s)$$

Portanto, para uma entrada degrau unitário, ou seja, $R(s) = 1/s$, a saída será descrita por

$$X(s) = \frac{200}{s(s^2 + 20s + 200)}$$

O termo quadrático é da forma $s^2 + 2\zeta\omega_n s + \omega_n^2$ (veja a Seção 13.3.1) e tem uma frequência natural ω_n de $\sqrt{200}$ e um fator de amortecimento ζ de $10/\sqrt{(200)}$. Portanto, podemos determinar qual será a resposta deste sistema de segunda ordem para sinais de entrada em degrau e quanto tempo o sistema precisa para estabilizar; por exemplo, o tempo de estabilização de 2% (veja a Seção 12.5) é $4/\zeta\omega_n$ e, portanto, $4/10 = 0,4$ s. Este tempo é relativamente longo, e precisaríamos considerar como ele poderia ser reduzido para milissegundos. Podemos considerar a substituição do amplificador com seu ganho proporcional por um controle PD.

Resumo

A **mecatrônica** é o resultado da integração da engenharia mecânica com a eletrônica e o controle computacional inteligente de forma coordenada e concorrente no projeto e manufatura de produtos. Ela envolve o desenvolvimento e a solução integrada, em vez de uma junção de conhecimentos. No desenvolvimento de soluções, é necessária a consideração de modelos para se poder fazer previsões de como as soluções devem funcionar.

Problemas

- 24.1 Para as situações a seguir, apresente o esboço de possíveis soluções de projeto:
- Controlador de temperatura de um forno.

- (b) Um mecanismo para selecionar objetos pequenos, médios e grandes que se movem ao longo de uma esteira rolante de modo que eles sejam desviados para rampas para serem empacotados.
- (c) Um *plotter x-y* (uma máquina que traça gráficos mostrando como uma entrada x varia de acordo com uma outra y).

Atividades de pesquisa

- 24.2 Pesquise sobre o sistema utilizado em carros que evita o travamento das rodas na frenagem (ABS) e descreva os princípios de operação dele.
- 24.3 Pesquise o mecanismo usado por impressoras de matriz de pontos e descreva o princípio de operação dele.
- 24.4 Pesquise sobre o protocolo da rede de campo CAN usada em carros.

Atividades de projeto

- 24.5 Projete um sistema de termômetro digital que mostre a temperatura entre 0 e 99°C. Você pode considerar uma solução baseada no uso de um microprocessador com chips de RAM e ROM ou então o uso de um microcontrolador.
- 24.6 Projete um ohmímetro digital que apresente em um display a resistência de um resistor medindo entre seus terminais. Você pode basear sua solução em um multivibrador monoestável, como, por exemplo, o CI 74121, que fornece um pulso com uma largura relacionada a uma constante RC do circuito conectado ao CI.
- 24.7 Projete um barômetro digital que apresente em um display a pressão atmosférica. Você pode basear sua solução no uso do sensor de pressão MPX2100AP.
- 24.8 Projete um sistema que pode ser usado para controlar a velocidade de um motor CC. Você pode considerar o uso da placa de desenvolvimento M68HC11.
- 24.9 Projete um sistema envolvendo um CLP para colocar sobre uma correia transportadora caixas em grupos de quatro.

Apêndice A: Transformada de Laplace

A.1 Transformada de Laplace

Considere uma grandeza que seja função do tempo. Podemos expressar esta grandeza estando no **domínio do tempo** e representar esta função como $f(t)$. Em muitos problemas, estamos interessados apenas nos valores de tempo maiores ou iguais a 0, ou seja, $t \geq 0$. Para obter a transformada de Laplace desta função, multiplicamos ela por e^{-st} e, em seguida, integramos em relação ao tempo de zero ao infinito. Neste caso, s é uma constante em que a unidade é 1/tempo. Este resultado é a **transformada de Laplace**, e diz-se que a equação está no **domínio s** . Portanto, a transformada de Laplace da função do tempo $f(t)$, que é escrita como $\mathcal{L}\{f(t)\}$, é dada por

$$\mathcal{L}\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt$$

A transformada é **parcial** porque os valores são considerados apenas entre 0 e $+\infty$, e não ao longo da faixa total de tempo de $-\infty$ a $+\infty$.

Podemos realizar operações algébricas com quantidades no domínio s , ou seja, soma, subtração, multiplicação e divisão, da mesma forma que fazemos com quantidades algébricas. Não devemos fazer estas operações no domínio do tempo, considerando que as equações são diferenciais. Isto significa que podemos obter uma expressão consideravelmente simplificada no domínio s . Se desejarmos ver como a grandeza varia com o tempo no domínio do tempo, fazemos a transformação inversa. Isto envolve determinar a função no domínio do tempo que corresponde à expressão simplificada no domínio s .

Quando se trabalha no domínio s , a função é escrita como uma função de s , ou seja, $F(s)$. É comum usar a letra maiúscula F para a transformada de Laplace e a letra minúscula f para uma função que varia no tempo, $f(t)$. Portanto,

$$\mathcal{L}\{f(t)\} = F(s)$$

Para a operação inversa, quando a função do tempo é obtida a partir da transformada de Laplace, podemos escrever

$$f(t) = \mathcal{L}^{-1}\{F(s)\}$$

Esta equação é lida como: $f(t)$ é a transformada inversa da transformada de Laplace $F(s)$.

A.1.1 Princípio da transformada de Laplace

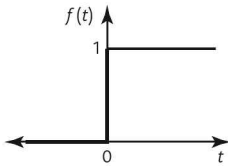


Figura A.1 Função degrau unitário.

Para ilustrar a transformação de uma grandeza no domínio do tempo para o domínio s , considere uma função que tem um valor constante de 1 para todos os valores de tempo maiores do que 0, ou seja, $f(t) = 1$ para $t \geq 0$. Isto descreve uma função degrau unitário e é mostrado na Figura A.1.

A transformada de Laplace é então

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^{\infty} 1e^{-st} dt = -\frac{1}{s}[e^{-st}]_0^{\infty}$$

Visto que com $t = \infty$ o valor de e é 0, e com $t = 0$ o valor de e^{-0} é -1 , então

$$F(s) = \frac{1}{s}$$

Como um outro exemplo, a equação a seguir mostra a determinação da transformada de Laplace da função e^{at} , em que a é uma constante. A transformada de Laplace de $f(t) = e^{at}$ é

$$F(s) = \int_0^{\infty} e^{at}e^{-st} dt = \int_0^{\infty} e^{-(s-a)t} dt = -\frac{1}{s-a}[e^{-(s-a)t}]_0^{\infty}$$

Quando $t = \infty$, o termo entre parênteses se torna 0, e quando $t = 0$, ele se torna -1 . Portanto,

$$F(s) = \frac{1}{s-a}$$

A.2 Funções degrau unitário e impulso

As funções degrau unitário e impulso são funções de entrada comuns em sistemas. A seguir vemos como obter a transformada de Laplace destas funções.

A.2.1 Função degrau unitário

A Figura A.1 mostra um gráfico de uma função degrau unitário. Esta função, quando o degrau ocorre em $t = 0$, tem a equação

$$\begin{aligned} f(t) &= 1 \text{ para todos os valores de } t \text{ maiores do que } 0 \\ f(t) &= 0 \text{ para todos os valores de } t \text{ menores do que } 0 \end{aligned}$$

A função degrau descreve uma mudança abrupta em uma grandeza de zero para um valor de estado estacionário, como, por exemplo, uma variação na tensão aplicada em um circuito quando a chave de alimentação é ligada.

A função degrau unitário não pode ser descrita por $f(t) = 1$, visto que isto implicaria que a função teria um valor constante de 1 para todos os valores de t , tanto positivos quanto negativos. A função degrau unitário que comuta de 0 para +1 em $t = 0$ é convencionalmente descrita pelo símbolo $u(t)$ ou $H(t)$, em que o H é em homenagem ao seu criador, O. Heaviside. Às vezes ela é denominada **função de Heaviside**.

A transformada de Laplace desta função degrau é, conforme deduzida na seção anterior,

$$F(s) = \frac{1}{s}$$

A transformada de Laplace da função degrau de amplitude a é

$$F(s) = \frac{a}{s}$$

A.2.2 Função impulso

Considere um pulso retangular de amplitude $1/k$ que ocorre em $t = 0$ e que tem uma largura de pulso de k , ou seja, a área do pulso é 1. A Figura A.2(a) mostra este pulso. O pulso pode ser descrito como

$$f(t) = \frac{1}{k} \text{ para } 0 \leq t < k$$

$$f(t) = 0 \text{ para } t > k$$

Se mantermos este pulso com área 1 e diminuirmos a largura (reduzir k), a amplitude aumenta. Portanto, no limite para o qual $k \rightarrow 0$, teremos apenas uma linha vertical em $t = 0$, com a amplitude indo para o infinito. O resultado é um gráfico que é zero exceto em um único ponto onde há um *spike* infinito (Figura A.2(b)). Este gráfico pode ser usado para representar um impulso. Diz-se que o impulso é unitário porque a área dele é 1. Esta função é representada por $\delta(t)$, que é a **função impulso unitário** ou **delta de Dirac**.

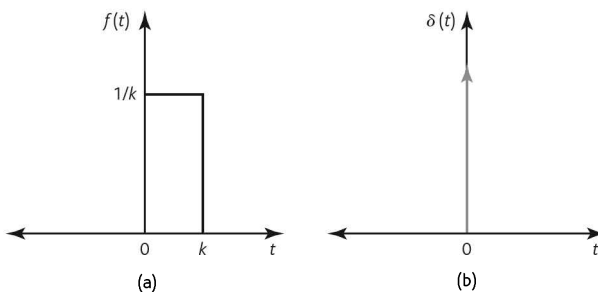


Figura A.2 (a) Pulso retangular, (b) impulso.

A transformada de Laplace para o pulso retangular de area unitária da Figura A.2(a) é dada por

$$\begin{aligned} F(s) &= \int_0^{\infty} f(t)e^{-st} dt = \int_0^k \frac{1}{k} e^{-st} dt + \int_k^{\infty} 0 e^{-st} dt \\ &= \left[-\frac{1}{sk} e^{-st} \right]_0^k = -\frac{1}{sk} (e^{-sk} - 1) \end{aligned}$$

Para obter a transformada de Laplace para o impulso unitário, precisamos determinar o valor desta equação no limite em que $k \rightarrow 0$. Podemos fazer isto expandindo o termo exponencial como uma série. Portanto,

$$e^{-sk} = 1 - sk + \frac{(-sk)^2}{2!} + \frac{(-sk)^3}{3!} + \dots$$

e assim podemos escrever

$$F(s) = 1 - \frac{sk}{2!} + \frac{(sk)^2}{3!} + \dots$$

Portanto, no limite em que $k \rightarrow 0$, a transformada de Laplace tende para o valor 1:

$$\mathcal{L}\{\delta(t)\} = 1$$

Como a área do impulso anterior é 1, podemos definir a amplitude deste impulso como sendo 1. Portanto, a equação anterior fornece a transformada de Laplace para um impulso unitário. Um impulso de tamanho a é representado por $a\delta(t)$, e a transformada de Laplace é

$$\mathcal{L}\{a\delta(t)\} = a$$

A.3 Transformadas de Laplace padrão

Na determinação da transformada de Laplace de funções, não é necessário calcular as integrais, visto que existem tabelas disponíveis que fornecem as transformadas de Laplace das funções mais comuns. Estas funções, combinadas com o uso de propriedades das transformadas (veja a próxima seção), permitem que os problemas mais comuns sejam resolvidos. A Tabela A.1 lista algumas das funções do tempo comuns e suas transformadas de Laplace. Observe na tabela que $f(t) = 0$ para todos os valores negativos de t e que o termo $u(t)$ foi omitido na maioria das funções do tempo, devendo ser considerado.

A.3.1 Propriedades da transformada de Laplace

Nesta seção são abordadas propriedades básicas da transformada de Laplace. Estas propriedades permitem o uso de tabelas de transformadas padrão em uma variedade de situações.

Tabela A.1 Transformadas de Laplace

Função do tempo $f(t)$	Transformada de Laplace $f(s)$
1 $\delta(t)$, impulso unitário	1
2 $\delta(t - T)$, impulso unitário atrasado	e^{-sT}
3 $u(t)$, degrau unitário	$\frac{1}{s}$
4 $u(t - T)$, degrau unitário com atraso	$\frac{e^{-sT}}{s}$
5 t , uma rampa unitária	$\frac{1}{s^2}$
6 t^n , rampa de ordem n	$\frac{n!}{s^{n+1}}$
7 e^{-at} , decaimento exponencial	$\frac{1}{s + a}$
8 $1 - e^{-at}$, crescimento exponencial	$\frac{a}{s(s + a)}$
9 te^{-at}	$\frac{1}{(s + a)^2}$
10 $t^n e^{-at}$	$\frac{n!}{(s + a)^{n+1}}$
11 $t - \frac{1 - e^{-at}}{a}$	$\frac{a}{s^2(s + a)}$
12 $e^{-at} - e^{-bt}$	$\frac{b - a}{(s + a)(s + b)}$
13 $(1 - at)e^{-at}$	$\frac{s}{(s + a)^2}$
14 $1 - \frac{b}{b - a}e^{-at} + \frac{a}{b - a}e^{-bt}$	$\frac{ab}{s(s + a)(s + b)}$
15 $\frac{e^{-at}}{(b - a)(c - a)} + \frac{e^{-bt}}{(c - a)(a - b)} + \frac{e^{-ct}}{(a - c)(b - c)}$	$\frac{1}{(s + a)(s + b)(s + c)}$
16 $\text{sen } \omega t$, uma onda senoidal	$\frac{\omega}{s^2 + \omega^2}$
17 $\text{cos } \omega t$, uma onda co-senoidal	$\frac{s}{s^2 + \omega^2}$
18 $e^{-at} \text{sen } \omega t$, uma onda senoidal amortecida	$\frac{\omega}{(s + a)^2 + \omega^2}$
19 $e^{-at} \text{cos } \omega t$, uma onda co-senoidal amortecida	$\frac{s + a}{(s + a)^2 + \omega^2}$
20 $1 - \text{cos } \omega t$	$\frac{\omega^2}{s(s^2 + \omega^2)}$
21 $t \text{cos } \omega t$	$\frac{s^2 - \omega^2}{(s^2 + \omega^2)^2}$
22 $t \text{sen } \omega t$	$\frac{2\omega s}{(s^2 + \omega^2)^2}$

(continua)

Tabela A.1 (continuação)

Função do tempo $f(t)$	Transformada de Laplace $f(s)$
23 $\text{sen}(\omega t + \theta)$	$\frac{\omega \cos \theta + s \text{sen } \theta}{s^2 + \omega^2}$
24 $\text{cos}(\omega t + \theta)$	$\frac{s \cos \theta - \omega \text{sen } \theta}{s^2 + \omega^2}$
25 $\frac{\omega}{\sqrt{1 - \zeta^2}} e^{-\zeta \omega t} \text{sen } \omega \sqrt{1 - \zeta^2} t$	$\frac{\omega^2}{s^2 + 2\zeta \omega s + \omega^2}$
26 $1 - \frac{1}{\sqrt{1 - \zeta^2}} e^{-\zeta \omega t} \text{sen } (\omega \sqrt{1 - \zeta^2} t + \phi), \text{cos } \phi = \zeta$	$\frac{\omega^2}{s(s^2 + 2\zeta \omega s + \omega^2)}$

Linearidade

Se duas funções no tempo, por exemplo, $f(t)$ e $g(t)$, têm transformadas de Laplace, a transformada da soma destas funções do tempo é a soma das duas transformadas em separado:

$$\mathcal{L}\{af(t) + bg(t)\} = a\mathcal{L}f(t) + b\mathcal{L}g(t)$$

a e b são constantes. Portanto, por exemplo, a transformada de Laplace de $1 + 2t + 4t^2$ é dada pela transformada dos termos individuais na expressão. Portanto, usando os itens 1, 5 e 6 na Tabela A.1,

$$F(s) = \frac{1}{s} + \frac{2}{s^2} + \frac{8}{s^3}$$

Deslocamento no domínio s

Esta propriedade é usada para determinar a transformada de Laplace de funções que têm um fator exponencial, e é algumas vezes denominada **propriedade do primeiro deslocamento**. Se $F(s) = \mathcal{L}\{f(t)\}$, então

$$\mathcal{L}\{e^{at}f(t)\} = F(s - a)$$

Por exemplo, a transformada de Laplace de $e^{at}t^n$, visto que a transformada de Laplace de t^n é dada pelo item 6 da Tabela A.1 como $n!/s^{n+1}$, é dada por

$$\mathcal{L}\{e^{at}t^n\} = \frac{n!}{(s - a)^{n+1}}$$

Deslocamento no domínio do tempo

Se um sinal é atrasado por um tempo T , então sua transformada de Laplace é multiplicada por e^{-sT} . Se $F(s)$ for a transformada de Laplace de $f(s)$, então

$$\mathcal{L}\{f(t - T)u(t - T)\} = e^{-sT}F(s)$$

Este atraso de um tempo T no sinal é conhecido como **segundo teorema do deslocamento**.

A propriedade do deslocamento no domínio do tempo pode ser aplicada a todas as transformadas de Laplace. Portanto, para um impulso $\delta(t)$ que é atrasado por um tempo T para obtermos a função $\delta(t - T)$, a transformada de Laplace de $\delta(t)$, isto é, 1, é multiplicada por e^{-sT} , resultando em le^{-sT} como a transformada para a função atrasada.

Funções periódicas

Para uma função $f(t)$ que é uma função periódica de período T , a transformada de Laplace desta função é

$$\mathcal{L}f(t) = \frac{1}{1 - e^{-sT}} F_1(s)$$

em que $F_1(s)$ é a transformada de Laplace da função para o primeiro período. Portanto, por exemplo, considere a transformada de Laplace de uma sequência de pulsos retangulares periódicos de período T , como mostra a Figura A.3. A transformada de Laplace de um único pulso retangular é dada por $(1/s)(1 - e^{-sT/2})$. Portanto, usando a equação anterior, a transformada de Laplace é

$$\frac{1}{1 - e^{-sT}} \times \frac{1}{s}(1 - e^{-sT/2}) = \frac{1}{s(1 + e^{-sT/2})}$$

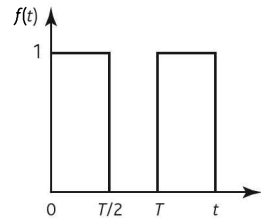


Figura A.3 Pulsos retangulares.

Teoremas do valor inicial e do valor final

O teorema do valor inicial pode ser enunciado como: se uma função do tempo $f(t)$ tem uma transformada de Laplace $F(s)$, então, no limite para o qual o tempo tende a zero, o valor da função é dado por

$$\lim_{t \rightarrow 0^+} f(t) = \lim_{s \rightarrow \infty} sF(s)$$

Por exemplo, o valor inicial da função que resulta na transformada de Laplace $3/(s + 2)$ no valor limite de $3s/(s + 2) = 3/(1 + 2/s)$ conforme s tende ao infinito é 3.

O teorema do valor final pode ser enunciado como: se uma função do tempo $f(t)$ tem uma transformada de Laplace $F(s)$, então, no limite para o qual o tempo tende ao infinito, o valor da função é dado por

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s)$$

Derivadas

A transformada de Laplace da derivada de uma função $f(t)$ é dada por

$$\mathcal{L}\left\{\frac{d}{dt}f(t)\right\} = sF(s) - f(0)$$

em que $f(0)$ é o valor da função quando $t = 0$. Por exemplo, a transformada de Laplace de $2(dx/dt) + x = 4$ é $2[sX(s) - x(0)] + X(s) = 4/s$, e se temos $x = 0$ em $t = 0$, então ela é $2sX(s) + X(s) = 4/s$ ou $X(s) = 4/[s(2s + 1)]$.

Para a segunda derivada

$$\mathcal{L}\left\{\frac{d^2}{dt^2}f(t)\right\} = s^2F(s) - sf(0) - \frac{d}{dt}f(0)$$

em que $df(0)/dt$ é o valor da primeira derivada em $t=0$.

Integrais

A transformada de Laplace da integral de uma função $f(t)$ que tem uma transformada de Laplace $F(s)$ é dada por

$$\mathcal{L}\left\{\int_0^t f(t) dt\right\} = \frac{1}{s}F(s)$$

Por exemplo, a transformada de Laplace da integral da função e^{-t} entre os limites 0 e t é dada por

$$\mathcal{L}\left\{\int_0^t e^{-t} dt\right\} = \frac{1}{s}\mathcal{L}\{e^{-t}\} = \frac{1}{s(s+1)}$$

A.4 Transformada inversa

A transformada de Laplace inversa é a conversão de uma transformada de Laplace $F(s)$ em uma função do tempo $f(t)$. Esta operação pode ser escrita como

$$\mathcal{L}^{-1}\{F(s)\} = f(t)$$

Geralmente a operação inversa pode ser executada usando-se a Tabela A.1. A propriedade da linearidade das transformadas de Laplace diz que se temos uma transformada como a soma de dois termos separados, então podemos obter a inversa de cada um separadamente, e a soma das duas transformadas inversas é a transformada inversa desejada:

$$\mathcal{L}^{-1}\{aF(s) + bG(s)\} = a\mathcal{L}^{-1}F(s) + b\mathcal{L}^{-1}G(s)$$

Portanto, para ilustrar como o rearranjo de uma função pode frequentemente colocá-la em uma forma padrão mostrada na tabela, a transformada inversa de $3/(2s+1)$ pode ser obtida rearranjando-a como

$$\frac{3(1/2)}{s + (1/2)}$$

A tabela (item 7) contém a transformada $1/(s+a)$ com a inversa de e^{-at} . Portanto, a transformada inversa é simplesmente esta multiplicada pela constante $(3/2)$ com $a = (1/2)$, ou seja, $(3/2)e^{-t/2}$.

Como um outro exemplo, considere a transformada de Laplace inversa de $(2s+2)/(s^2+1)$. Esta expressão pode ser rearranjada como

$$2\left[\frac{s}{s^2+1} + \frac{1}{s^2+1}\right]$$

O primeiro termo dentro dos colchetes tem como transformada inversa $\cos t$ (item 17 na Tabela A.1) e o segundo termo $\sin t$ (item 16 na Tabela A.1). Portanto, a transformada inversa da expressão é $2 \cos t + 2 \sin t$.

A.4.1 Frações parciais

Frequentemente $F(s)$ é uma razão entre dois polinômios e não pode ser facilmente identificada com uma transformada padrão da Tabela A.1. Ela tem que ser convertida em termos de frações simples antes que as transformadas padrão possam ser usadas. O processo de converter uma expressão em termos de frações simples é denominado **decomposição em frações parciais**. Esta técnica pode ser usada desde que o grau do numerador seja menor do que o do denominador. O grau de um polinômio é a maior potência de s na expressão. Quando o grau do numerador é igual ou maior do que o do denominador, o denominador tem que ser dividido pelo numerador até que o resultado seja a soma dos termos com o termo fracionário restante tendo um numerador de grau menor do que o denominador.

Podemos considerar que existem basicamente três tipos de frações parciais:

- 1 O denominador contém fatores que são apenas da forma $(s + a)$, $(s + b)$, $(s + c)$, etc. A expressão é da forma

$$\frac{f(s)}{(s + a)(s + b)(s + c)}$$

e as frações parciais são

$$\frac{A}{(s + a)} + \frac{B}{(s + b)} + \frac{C}{(s + c)}$$

- 2 Existem fatores $(s + a)$ repetidos no denominador, ou seja, o denominador contém potências deste fator, e a expressão é da forma

$$\frac{f(s)}{(s + a)^n}$$

Esta, dividida em frações parciais, resulta em

$$\frac{A}{(s + a)^1} + \frac{B}{(s + a)^2} + \frac{C}{(s + a)^3} + \dots + \frac{N}{(s + a)^n}$$

- 3 O denominador contém fatores quadráticos, e estes não são fatorados sem termos imaginários. Para uma expressão da forma

$$\frac{f(s)}{(as^2 + bs + c)(s + d)}$$

as frações parciais são

$$\frac{As + B}{as^2 + bs + c} + \frac{C}{s + d}$$

Os valores das constantes A , B , C , etc. podem ser determinados fazendo-se uso do fato de que qualquer uma das igualdades entre as expressões e as frações parciais tem que ser verdadeira para todos os valores de s ou que os coeficientes de s^n

na expressão têm que ser iguais aos de s^n na expansão das frações parciais. O uso do primeiro método é ilustrado pelo exemplo a seguir, em que as frações parciais de

$$\frac{3s + 4}{(s + 1)(s + 2)}$$

são

$$\frac{A}{s + 1} + \frac{B}{s + 2}$$

Então, para que as expressões sejam iguais, temos que ter

$$\frac{3s + 4}{(s + 1)(s + 2)} = \frac{A(s + 2) + B(s + 1)}{(s + 1)(s + 2)}$$

e conseqüentemente $3s + 4 = A(s + 2) + B(s + 1)$. Isto tem que ser verdadeiro para todos os valores de s . O procedimento é obter valores de s que permitem que alguns dos termos que envolvem constantes se tornem zero e, assim, permitam que outras constantes sejam determinadas. Portanto, se fizermos $s = -2$, temos $3(-2) + 4 = A(-2 + 2) + B(-2 + 1)$ e, assim, $B = 2$. Se agora fizermos $s = -1$, então $3(-1) + 4 = A(-1 + 2) + B(-1 + 1)$ e, assim, $A = 1$. Portanto,

$$\frac{3s + 4}{(s + 1)(s + 2)} = \frac{1}{s + 1} + \frac{2}{s + 2}$$

Problemas

- A.1** Determine as transformadas de Laplace de (a) $2t$, (b) $\sin 2t$, (c) um impulso unitário no instante $t = 2$ s, (d) $4 dx/dt$ em que $x = 2$ em $t = 0$, (e) $3 d^2x/dt^2$ quando $x = 0$ e $dx/dt = 0$ em $t = 0$, (f) a integral entre t e 0 de e^{-t} .
- A.2** Determine as transformadas de Laplace inversas de
- $1/s^2$
 - $5s/(s^2 + 9)$
 - $(3s - 1)/[s(s - 1)]$
 - $1/(s + 3)$
- A.3** Determine o valor inicial da função com transformada de Laplace $5/(s + 2)$.

Apêndice B:

Sistemas de Numeração

B.1 Sistemas de numeração

O **sistema decimal** é baseado no uso de 10 símbolos ou dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Quando um número é representado por este sistema, a posição do dígito no número indica que o peso associado a cada dígito aumenta por um fator de 10 à medida que se desloca da direita para a esquerda:

...	10^3	10^2	10^1	10^0
	milhares	centenas	dezenas	unidades

O **sistema binário** é baseado em apenas dois símbolos ou estados: 0 e 1. Estes são denominados de dígitos binários (*binary digits*) ou bits. Quando um número é representado neste sistema, a posição do dígito no número indica que o peso associado a cada dígito aumenta por um fator de 2 à medida que se desloca da direita para a esquerda:

...	2^3	2^2	2^1	2^0
	bit 3	bit 2	bit 1	bit 0

Por exemplo, o número decimal 15 no sistema binário é 1111. Em um número binário, o bit 0 é denominado de **bit menos significativo** (LSB – *Least Significant Bit*) e o maior bit é o **bit mais significativo** (MSB – *Most Significant Bit*).

O **sistema octal** é baseado em oito dígitos: 0, 1, 2, 3, 4, 5, 6, 7. Quando um número é representado neste sistema, a posição do dígito no número indica que o peso associado a cada dígito aumenta por um fator de 8 à medida que se desloca da direita para a esquerda:

...	8^3	8^2	8^1	8^0
-----	-------	-------	-------	-------

Por exemplo, o número decimal 15 no sistema octal é 17.

O **sistema hexadecimal** é baseado em 16 dígitos/símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Quando um número é representado neste sistema, a posição

do dígito no número indica que o peso associado a cada dígito aumenta por um fator de 16 à medida que se desloca da direita para a esquerda:

$$\dots \quad 16^3 \quad 16^2 \quad 16^1 \quad 16^0$$

Por exemplo, o número decimal 15 é F no sistema hexadecimal. Este sistema é geralmente usado na escrita de programas para sistemas microprocessados, visto que ele representa um método muito compacto de inserção de dados.

O **sistema decimal codificado em binário** (BCD – *Binary-Coded Decimal*) é o mais usado nos computadores. Cada dígito decimal é codificado separadamente em binário. Por exemplo, o número decimal 15 em BCD é 0001 0101. Este código é usado nas saídas de sistemas microprocessados, que precisam acionar um display decimal. Cada dígito decimal no display recebe dados do microprocessador com seu próprio código binário.

A Tabela B.1 fornece exemplos de números nos sistemas decimal, binário, octal e hexadecimal.

Tabela B.1 Sistemas de numeração

Decimal	Binário	BCD	Octal	Hexadecimal
0	0000	0000 0000	0	0
1	0001	0000 0001	1	1
2	0010	0000 0010	2	2
3	0011	0000 0011	3	3
4	0100	0000 0100	4	4
5	0101	0000 0101	5	5
6	0110	0000 0110	6	6
7	0111	0000 0111	7	7
8	1000	0000 1000	10	8
9	1001	0000 1001	11	9
10	1010	0001 0000	12	A
11	1011	0001 0001	13	B
12	1100	0001 0010	14	C
13	1101	0001 0011	15	D
14	1110	0001 0100	16	E
15	1111	0001 0101	17	F

B.2 Matemática binária

A soma de números binários é baseada nas seguintes regras:

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 + 0 = 1 \\ 1 + 1 &= 10 \quad \text{ou seja } 0 + \text{carry } 1 \\ 1 + 1 + 1 &= 11 \quad \text{ou seja } 1 + \text{carry } 1 \end{aligned}$$

Em números decimais, a adição de 14 com 19 resulta em 33. Em números binários, esta adição é feita assim:

Primeira parcela	01110
Segunda parcela	10111
Soma	100001

Para o bit 0, $0 + 1 = 1$. Para o bit 1, $1 + 1 = 10$, e assim fica o 0 e transporta-se o 1 (*carry*) para a próxima coluna. Para o bit 3, $1 + 0 + \text{carry } 1 = 10$. Para o bit 4, $1 + 0 + \text{carry } 1 = 10$. Continuamos esse processo até acabarmos com a soma mais o *carry* 1. Portanto, o número final é 100001. Quando somamos os números binários A e B para obter C , ou seja, $A + B = C$, A é a **primeira parcela**, B é a **segunda parcela** e C é a **soma**.

A subtração de números binários é baseada nas seguintes regras:

$$\begin{aligned}
 0 - 0 &= 0 \\
 1 - 0 &= 1 \\
 1 - 1 &= 0 \\
 0 - 1 &= 10 - 1 + \textit{borrow} = 1 + \textit{borrow}
 \end{aligned}$$

Quando fazemos o cálculo $0 - 1$, um 1 é emprestado (*borrow*) da próxima coluna à esquerda que contém um 1. O exemplo a seguir ilustra isto. Em números decimais, a subtração de 14 a partir de 27 resulta em 13.

Minuendo	11011
Subtraendo	01110
Diferença	01101

Para o bit 0, temos $1 - 0 = 1$. Para o bit 1, temos $1 - 1 = 0$. Para o bit 2, temos $0 - 1$. Agora pegamos emprestado um 1 da próxima coluna à esquerda, de forma que agora temos $10 - 1 = 1$. Para o bit 3, temos $0 - 1$; lembre-se do empréstimo (*borrow*). Pegando emprestado novamente da próxima coluna à esquerda, temos $10 - 1 = 1$. Para o bit 4, temos $0 - 0 = 0$; lembre-se que foi emprestado um 1. Quando subtraímos os números binários A e B para obter C , ou seja, $A - B = C$, então A é denominado **minuendo**, B é o **subtraendo** e C é a **diferença**.

A subtração de números binários é mais fácil de ser realizada eletronicamente quando é usado um método alternativo. O exemplo anterior de subtração pode ser considerado uma adição de um número positivo com um número negativo. As técnicas a seguir indicam como podemos especificar os números negativos e, assim, transformar a subtração em adição. Ela nos permite lidar com números negativos em quaisquer circunstâncias.

Os números usados até aqui foram considerados **sem sinal**. Isto porque o número não contém um indicador informando se ele é positivo ou negativo. Um número é dito **signalizado** quando o bit mais significativo é usado para indicar o sinal do número, sendo o 0 usado para número positivo e o 1 para negativo. Quando temos um número positivo, o escrevemos de forma normal, com um 0 à esquerda. Portanto, o número binário positivo de 10010 seria escrito como 010010. O número negativo de 10010 seria escrito como 110010. Entretanto, esta não é a forma mais útil para a representação de números negativos para serem manipulados por computadores.

A forma mais útil de representar números negativos é usando o método do complemento de dois. Um número binário tem dois complementos, conhecidos como **complemento de um** e **complemento de dois**. O complemento de um de um número binário é obtido trocando-se todos os uns por zeros e vice-versa no

número não sinalizado. O complemento de dois é obtido somando-se 1 ao complemento de um. Quando temos um número negativo, obtemos o complemento de dois e depois sinalizamos ele com 1, e o número positivo é sinalizado com 0. Considere a representação do número decimal -3 como um número em complemento de dois sinalizado. Primeiro escrevemos o número binário para 3 não sinalizado como 0011. Depois obtemos o complemento de um (1100), somamos 1 para obtermos o complemento de dois (1101) e, finalmente, sinalizamos com um 1 para indicar um número negativo. O resultado é 11101. A seguir temos um outro exemplo em que obtemos um número de 8 bits sinalizado em complemento de dois para o decimal -6 :

Número binário não sinalizado	000 0110
Complemento de um	111 1001
Somando 1	1
Complemento de dois não sinalizado	111 1010
Complemento de dois sinalizado	1111 1010

Quando temos um número positivo, o escrevemos de forma normal com um 0 à esquerda. Portanto, o número binário positivo de 100 1001 seria escrito como 0100 1001. A Tabela B.2 mostra alguns exemplos de números neste sistema.

Tabela B.2 Números sinalizados

Número decimal	Número sinalizado		Número decimal	Número sinalizado	
+127	0111 1111	Número binário sinalizado apenas com um 0	-1	1111 1111	Complemento de dois sinalizado com um 1
...			-2	1111 1110	
+6	0000 0110		-3	1111 1101	
+5	0000 0101		-4	1111 1100	
+4	0000 0100		-5	1111 1011	
+3	0000 0011		-6	1111 1010	
+0	0000 0010	...			
+1	0000 0001		-127	1000 0000	
+0	0000 0000				

A subtração de um número positivo de outro envolve a obtenção do complemento de dois do subtraendo e a soma com o minuendo sinalizado. Portanto, para a subtração entre o número decimal 6 e o decimal 4, temos

Minuendo sinalizado	0000 0100
Subtraendo em complemento de dois sinalizado	1111 1010
Soma	1111 1110

O bit mais significativo do resultado é 1 e, portanto, o resultado é negativo. Esta é a representação do número em complemento de dois sinalizado -2 .

Considere um outro exemplo: a subtração de 43 a partir de 57. O número positivo sinalizado de 57 é 0011 1001. O complemento de dois sinalizado de -43 é dado por

Número binário não sinalizado para 43	010 1011
Complemento de um	101 0100
Soma com 1	<u>1</u>
Complemento de dois não sinalizado	101 0101
Complemento de dois sinalizado	1101 0101

Portanto, obtemos pela adição do número positivo sinalizado com o número em complemento de dois sinalizado

Minuendo sinalizado	0011 1001
Subtraendo em complemento de dois sinalizado	<u>1101 0101</u>
Soma	0000 1110 + <i>carry</i> 1

O *carry* 1 é ignorado. O resultado é 0000 1110 e, como o bit mais significativo é 0, o resultado é positivo. O resultado é o número decimal 14.

Se desejarmos somar dois números negativos, então temos que obter o complemento de dois de cada um deles e então somá-los. Sempre que um número for negativo, usamos o complemento de dois; quando for positivo, apenas sinalizamos ele.

B.3 Números de ponto flutuante

No sistema de numeração decimal, os números grandes como 120 000 são frequentemente escritos em **notação científica** como $1,2 \times 10^5$ ou, talvez, 120×10^3 , e os números pequenos como 0,000 120 na forma de $1,2 \times 10^{-4}$ em vez de representar o número com uma vírgula decimal (ou ponto decimal, que é a representação em inglês) fixa. Os números na forma de notação são escritos em termos de potências de 10. Podemos usar esta notação para os números binários, porém escritos em termos de potência de 2. Por exemplo, podemos ter 1010 escrito como $1,010 \times 2^3$ ou talvez como $10,10 \times 2^2$. Como a vírgula binária (ou ponto binário) pode ser movida para posições diferentes através da escolha da potência de 2, esta notação é denominada **ponto flutuante**.

Um número de ponto flutuante é da forma $a \times r^e$, em que a é denominado **mantissa**, r é a **base** e e é o **expoente** ou **potência**. No caso dos números binários a base é 2, ou seja, temos $a \times 2^e$. A vantagem do uso de números em ponto flutuante é que, em comparação com a representação de ponto fixo, uma faixa bem maior de números pode ser representada para um dado número de dígitos.

Por causa desta representação, um número pode ser armazenado de diferentes formas, como, por exemplo, $0,1 \times 10^2$ ou $0,01 \times 10^3$. Em sistemas de computação, estes números são **normalizados**, ou seja, eles são representados na forma $0,1 \times r^e$. Portanto, no caso de números binários, temos $0,1 \times 2^e$ e, assim, se tivermos 0,00001001, representamos como $0,1001 \times 2^{-4}$. Para levar em consideração o sinal de um número binário, acrescentamos um bit 0 para números positivos e um bit 1 para negativos. Portanto, o número $0,1001 \times 2^{-4}$ torna-se $1,1001 \times 2^{-4}$ se for negativo e $0,1001 \times 2^{-4}$ se positivo.

Se quisermos somar $2,01 \times 10^3$ com $10,2 \times 10^2$, temos que fazer com que as potências (o termo expoente também é usado) sejam iguais. Portanto, podemos escrever $2,01 \times 10^3 + 1,02 \times 10^3$. Podemos somar então dígito por dígito, considerando qualquer *carry*, para obter $2,03 \times 10^3$. Adotamos um procedimento similar para números binários de ponto flutuante. Portanto, se quisermos somar $0,101100 \times 2^4$ com $0,111100 \times 2^2$, primeiro ajustamos os números para que tenham o mesmo expoente, como, por exemplo, $0,101100 \times 2^4$ e $0,001111 \times 2^4$, e somando dígito por dígito obtemos $0,111011 \times 2^4$.

Da mesma forma, a subtração de números de ponto flutuante dígito a dígito pode ocorrer entre dois números apenas quando eles têm o mesmo expoente. Portanto, $0,1101100 \times 2^{-4}$ menos $0,1010100 \times 2^{-5}$ pode ser escrito como $0,01010100 \times 2^{-4} - 0,101010 \times 2^{-4}$, e o resultado é dado como $0,1000010 \times 2^{-4}$.

B.4 Código Gray

Considere dois números sucessivos em código binário (0001 e 0010) (2 e 3 em decimal); entre um número e o outro houve a mudança de dois bits. Portanto, se temos, por exemplo, um encoder absoluto (veja a Seção 2.3.7) e posições sucessivas atribuídas a números binários sucessivos, então duas alterações têm que ser feitas neste caso. Isto representa um problema porque duas mudanças têm que ocorrer no mesmo instante; se uma ocorrer um pouco antes da outra, teremos um outro número indicado momentaneamente. Portanto, na mudança de 0001 para 0010 podemos ter momentaneamente 0011 ou 0000. Portanto, um método alternativo deve ser usado.

O **código Gray** é um código em que: apenas 1 bit varia de um número para o próximo. Este código não é ponderado, pois as posições dos bits no grupo do código não apresentam pesos específicos atribuídos a elas. Portanto, não é apropriado para operações aritméticas, mas é bastante usado para dispositivos de entrada/saída, como encoders absolutos. A Tabela B.3 lista números decimais e seus valores em código binário e código Gray.

Tabela B.3 Código Gray

Número decimal	Código binário	Código Gray	Número decimal	Código binário	Código Gray
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Problemas

- B.1** Qual é o maior número decimal que pode ser representado por um número binário de 8 bits?
- B.2** Converta os seguintes números binários para decimal:
(a) 1011
(b) 10 0001 0001
- B.3** Converta para hexa os números decimais (a) 423, (b) 529.
- B.4** Converta para decimal os números BCD (a) 0111 1000 0001, (b) 0001 0101 011 1.
- B.5** Quais são as representações em complemento de dois para os números decimais (a) -90 , (b) -35 .
- B.6** Qual deve ser o bit de paridade a ser acrescentado a (a) 100 1000, (b) 100 1111?
- B.7** Subtraia os seguintes números decimais usando o complemento de dois:
(a) $21 - 13$
(b) $15 - 3$

Apêndice C: Álgebra Booleana

C.1 Leis da álgebra booleana

A álgebra booleana envolve os dígitos binários 1 e 0 e as operações \cdot , $+$ e a inversão. As leis desta álgebra são:

1 Qualquer operação OR de uma variável com ela mesma é igual à variável:

$$A + A = A.$$

2 Qualquer operação AND de uma variável com ela mesma é igual à variável:

$$A \cdot A = A.$$

3 Não importa a ordem das variáveis em uma operação OR ou AND, como, por exemplo:

$$A + B = B + A \quad \text{e} \quad A \cdot B = B \cdot A$$

4 Conforme a tabela-verdade a seguir indica, esta igualdade é verdadeira:

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

A	B	C	$B \cdot C$	$A + B \cdot C$	$A + B$	$A + C$	$(A + B) \cdot (A + C)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

5 Da mesma forma, podemos usar uma tabela-verdade para mostrar que podemos tratar os termos entre parênteses da mesma forma que na álgebra comum, como, por exemplo:

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

6 Qualquer operação OR de uma variável com a mesma variável negada é igual a 1:

$$A + \bar{A} = 1$$

7 Qualquer operação AND de uma variável com a mesma variável negada é igual a 0:

$$A \cdot \bar{A} = 0$$

8 Qualquer operação OR de uma variável com 0 é igual à própria variável; qualquer operação OR de uma variável com 1 é igual a 1. Portanto, $A + 0 = A$ e $A + 1 = 1$.

9 Qualquer operação AND de uma variável com 0 é igual a 0; qualquer operação AND de uma variável com 1 é igual a própria variável. Portanto, $A \cdot 0 = 0$ e $A \cdot 1 = A$.

Como uma ilustração do uso desta leis para simplificar expressões booleanas, considere a simplificação de

$$(A + B) \cdot \bar{C} + A \cdot C$$

Usando o item 5 para o primeiro termo, temos

$$A \cdot \bar{C} + B \cdot \bar{C} + A \cdot C$$

Podemos reagrupar e usar o item 6 para obter

$$A \cdot (\bar{C} + C) + B \cdot \bar{C} = A \cdot 1 + B \cdot \bar{C}$$

Portanto, usando o item 9, a expressão simplificada se torna

$$A + B \cdot \bar{C}$$

C.2 Teoremas de De Morgan

Conforme ilustrado antes, as leis da álgebra booleana podem ser usadas para simplificar expressões booleanas. Além delas, temos o que é conhecido como **teoremas de De Morgan**:

1 O inverso do resultado de uma operação OR entre A e B é o mesmo que uma operação AND entre A negado e B negado. A tabela-verdade a seguir mostra a validade deste teorema:

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

A	B	A + B	$\overline{A + B}$	\overline{A}	\overline{B}	$\overline{A \cdot B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

2 O inverso do resultado de uma operação AND entre A e B é o mesmo que uma operação OR entre A negado e B negado. A tabela-verdade a seguir mostra a validade deste teorema:

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

A	B	A · B	$\overline{A \cdot B}$	\overline{A}	\overline{B}	$\overline{A} + \overline{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Como uma ilustração do uso dos teoremas de De Morgan, considere a simplificação do circuito lógico mostrado na Figura C.1.

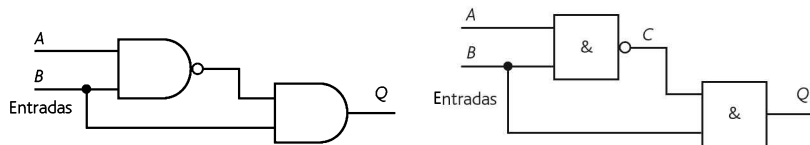


Figura C.1 Simplificação de um circuito.

A equação booleana para a saída em função das entradas é

$$Q = \overline{A \cdot B} \cdot B$$

Aplicando o segundo teorema, temos

$$Q = (\overline{A} + \overline{B}) \cdot B$$

Podemos escrever isto como

$$Q = \overline{A} \cdot B + \overline{B} \cdot B = \overline{A} \cdot B + 0 = \overline{A} \cdot B$$

Portanto, o circuito simplificado é como o mostrado na Figura C.2.

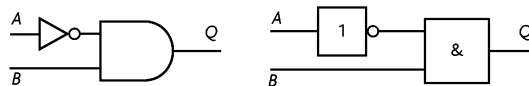


Figura C.2 Simplificação de circuito.

C.3 Generalização de função booleana a partir de tabelas-verdade

Dada uma situação em que os requisitos de um sistema podem ser especificados em termos de uma tabela-verdade, como um circuito lógico com um número mínimo de portas pode ser projetado de forma a corresponder à tabela-verdade dada?

A álgebra booleana pode ser usada para manipular funções de comutação de muitas formas equivalentes, algumas das quais utilizam muito mais portas lógicas do que outras; entretanto, a forma mínima é implementada com portas AND acionando uma porta OR ou vice-versa. Duas portas AND acionando uma única OR (Figura C.3(a)) resulta em

$$A \cdot B + A \cdot C$$

Esta forma é denominada **soma de produtos**.

Para duas portas OR acionando uma porta AND (Figura C.3(b)), temos

$$(A + B) \cdot (A + C)$$

Esta forma é conhecida como **produto de somas**. Portanto, ao considerar qual forma mínima corresponde a um dada tabela-verdade, o procedimento usual é determinar a soma de produtos ou o produto de somas que corresponde aos dados. Geralmente é usada a soma de produtos. O procedimento usado é considerar cada linha da tabela-verdade por sua vez e determinar o produto que corresponde à linha. O resultado final é a soma de todos estes produtos.

Suponha que temos uma linha na tabela-verdade em que

$$A = 1, B = 0 \text{ e saída } Q = 1$$

Quando A é 1 e B não é 1, então a saída é 1, e o produto que corresponde a esta linha é

$$Q = A \cdot \bar{B}$$

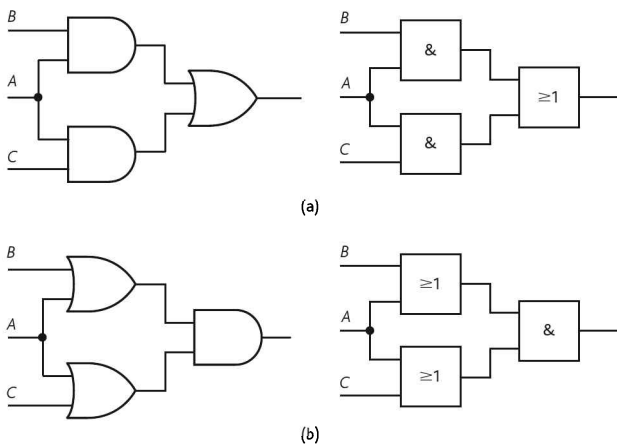


Figura C.3 (a) Soma de produtos, (b) produto de somas.

Podemos repetir esta operação para cada linha da tabela-verdade, conforme indicado na tabela a seguir:

A	B	Saída	Produtos
0	0	0	$\bar{A} \cdot \bar{B}$
0	1	0	$\bar{A} \cdot B$
1	0	1	$A \cdot \bar{B}$
1	1	0	$A \cdot B$

Entretanto, apenas as linhas da tabela-verdade que têm saída 1 precisam ser consideradas, visto que as linhas com saída 0 não contribuem para a expressão final; portanto, o resultado é

$$Q = A \cdot \bar{B}$$

O sistema de portas lógicas que corresponde a essa tabela-verdade é mostrado na Figura C.4.

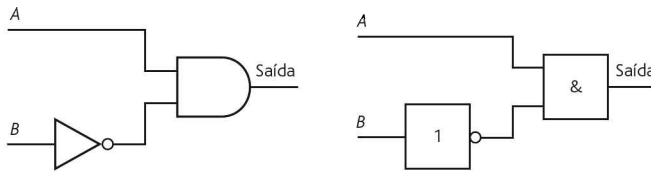


Figura C.4 Portas lógicas para a tabela-verdade anterior.

Como mais um exemplo, considere a seguinte tabela-verdade, em que apenas os termos produto com saída 1 foram incluídos:

A	B	C	Saída	Produtos
0	0	0	1	$\bar{A} \cdot \bar{B} \cdot \bar{C}$
0	0	1	0	
0	1	0	1	$\bar{A} \cdot B \cdot \bar{C}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	0	

Portanto, a soma de produtos que corresponde a esta tabela é

$$Q = \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot B \cdot \bar{C}$$

Isto pode ser simplificado, resultando em

$$Q = \bar{A} \cdot \bar{C} \cdot (\bar{B} + B) = \bar{A} \cdot \bar{C}$$

A mesma tabela-verdade também pode ser obtida através de uma porta NAND apenas.

C.4 Mapas de Karnaugh

O **mapa de Karnaugh** é um método gráfico que pode ser usado para produzir expressões booleanas de soma de produtos obtidas de uma tabelas-verdade. A tabela-verdade tem uma linha para o valor da saída de cada combinação dos valores de entrada. Para duas variáveis de entrada, a tabela apresenta quatro linhas; com três variáveis de entrada, a tabela tem oito linhas, e com quatro variáveis, dezesseis linhas. Com duas variáveis de entrada, existem quatro termos produto; com três variáveis, oito termos, e com quatro variáveis, dezesseis termos. O mapa de Karnaugh é desenhado como um arranjo retangular de células, em que cada célula corresponde a um termo produto em particular. Portanto, com duas variáveis, o mapa tem quatro células; com três variáveis, tem oito células, e com quatro variáveis, apresenta dezesseis células. Os valores de saída para as linhas são colocados nas células do mapa de Karnaugh, embora seja usual indicar apenas os valores em que a saída é 1 e deixar as células correspondentes aos zeros vazias.

A Figura C.5(a) mostra o mapa para duas variáveis de entrada. As células correspondem aos valores de saída para os seguintes produtos:

- a célula superior esquerda é $\bar{A} \cdot \bar{B}$
- a célula inferior esquerda é $A \cdot \bar{B}$
- a célula superior direita é $\bar{A} \cdot B$
- a célula inferior direita é $A \cdot B$

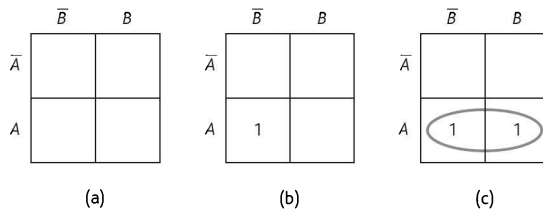


Figura C.5 Mapa para duas variáveis de entrada.

O arranjo dos quadrados do mapa é feito de forma que os quadrados adjacentes horizontalmente diferem entre si apenas em uma variável. O mesmo ocorre com os quadrados adjacentes verticalmente. Portanto, no mapa de duas variáveis, apenas a variável *A* difere horizontalmente e a variável *B* difere verticalmente.

Para a tabela-verdade a seguir, se colocarmos no mapa de Karnaugh os valores dados pelos produtos, indicando apenas as células que têm valor 1 e deixando em branco as outras com valor 0, obtemos o mapa mostrado na Figura C.5(b):

A	B	Saída	Produtos
0	0	0	$\bar{A} \cdot \bar{B}$
0	1	0	$\bar{A} \cdot B$
1	0	1	$A \cdot \bar{B}$
1	1	0	$A \cdot B$

Como existe apenas uma entrada 1 no quadrado inferior esquerdo, a tabela-verdade pode ser representada pela expressão Booleana

$$\text{saída} = A \cdot \bar{B}$$

Como mais um exemplo, considere a seguinte tabela-verdade:

A	B	Saída	Produtos
0	0	0	$\bar{A} \cdot \bar{B}$
0	1	0	$\bar{A} \cdot B$
1	0	1	$A \cdot \bar{B}$
1	1	1	$A \cdot B$

Ela corresponde ao mapa de Karnaugh mostrado na Figura C.5(c). Esta tem uma saída dada por:

$$\text{saída} = A \cdot \bar{B} + A \cdot B$$

Podemos simplificá-la como

$$A \cdot \bar{B} + A \cdot B = A \cdot (\bar{B} + B) = A$$

Quando duas células contêm valor 1, temos um agrupamento vertical que pode simplificar a expressão Booleana para uma variável apenas. Podemos fazer isto no mapa indicando quais células formam o grupo através de um laço, como podemos ver na Figura C.5(c).

A Figura C.6(a) mostra o mapa de Karnaugh para a tabela-verdade a seguir, que possui três variáveis:

A	B	C	Saída	Produtos
0	0	0	1	$\bar{A} \cdot \bar{B} \cdot \bar{C}$
0	0	1	0	$\bar{A} \cdot \bar{B} \cdot C$
0	1	0	1	$\bar{A} \cdot B \cdot \bar{C}$
0	1	1	0	$\bar{A} \cdot B \cdot C$
1	0	0	0	$A \cdot \bar{B} \cdot \bar{C}$
1	0	1	0	$A \cdot \bar{B} \cdot C$
1	1	0	0	$A \cdot B \cdot \bar{C}$
1	1	1	0	$A \cdot B \cdot C$

Assim como antes, usamos laços para simplificar a expressão Booleana resultante para apenas uma variável. O resultado é

$$\text{saída} = \bar{A} \cdot \bar{C}$$

A Figura C.6(b) mostra o mapa de Karnaugh para a seguinte tabela-verdade que contém quatro variáveis. O agrupamento simplifica a expressão Booleana resultante para

$$\text{saída} = \bar{A} \cdot \bar{C} \cdot D + A \cdot B \cdot C$$

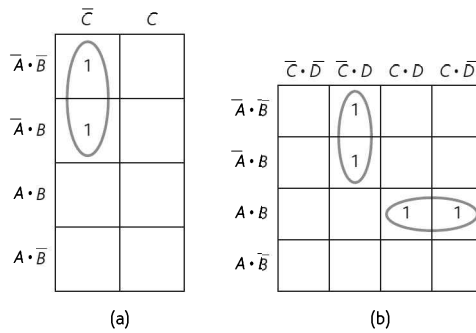


Figura C.6 (a) Mapa para três variáveis de entrada, (b) mapa para quatro variáveis de entrada.

A	B	C	D	Saída	Produtos
0	0	0	0	0	
0	0	0	1	1	$\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D$
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	$\bar{A} \cdot B \cdot \bar{C} \cdot D$
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	0	
1	1	0	1	0	
1	1	1	0	1	$A \cdot B \cdot C \cdot \bar{D}$
1	1	1	1	1	$A \cdot B \cdot C \cdot D$

Estas representações são exemplos simples do mapa de Karnaugh e do uso de agrupamentos. Observe que, no agrupamento, as células nas bordas inferior e superior e nas colunas à esquerda e à direita podem ser consideradas adjacentes. Imagine as bordas opostas do mapa justapostas. O agrupamento de um par de uns no mapa elimina a variável que aparece complementada e não complementada. Um agrupamento de quatro uns adjacentes elimina duas variáveis que aparecem, cada uma, complementada e não complementada. Um grupo constituído de um octeto de uns adjacentes elimina três variáveis que aparecem, cada uma, complementada e não complementada.

Como mais uma ilustração, considere uma máquina automática que parte apenas quando dois dentre três sensores (A , B e C) estão ativados. A tabela-verdade a seguir corresponde aos requisitos, e a Figura C.7(a) mostra o resultado do mapa de Karnaugh. A expressão Booleana que corresponde ao mapa e descreve o resultado de saída a partir da máquina é

$$\text{resultado} = A \cdot B + B \cdot C + A \cdot C$$

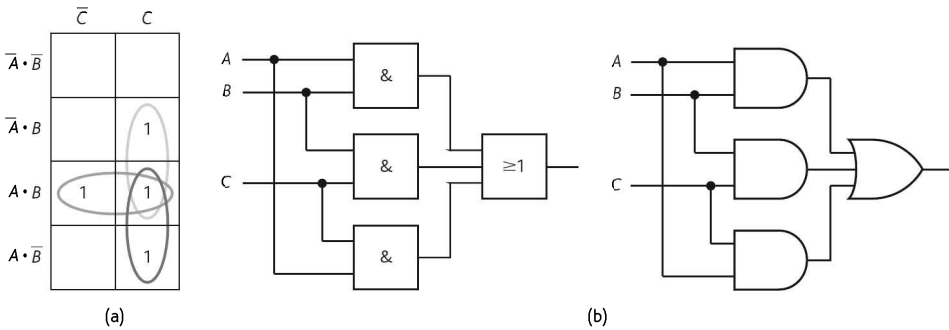


Figura C.7 Máquina automatizada.

A Figura C.7(b) mostra as portas lógicas que podem ser usadas para gerar esta expressão Booleana. $A \cdot B$ descreve uma porta AND com entradas A e B . Da mesma forma, $B \cdot C$ e $A \cdot C$ são outras duas portas AND. O sinal ≥ 1 indica que as saídas das três portas AND são entradas de uma porta OR.

A	B	C	Saída	Produtos
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{A} \cdot B \cdot C$
1	0	0	0	
1	0	1	1	$A \cdot \bar{B} \cdot C$
1	1	0	1	$A \cdot B \cdot \bar{C}$
1	1	1	1	$A \cdot B \cdot C$

Em alguns sistema lógicos, existem algumas combinações de variáveis de entrada para as quais a saída não é especificada. Essas combinações são denominadas “don’t care” (não importa). Quando elas entram no mapa, podem ser inseridas como 1 ou como 0, de modo que a expressão de saída seja a mais simplificada.

Problemas

- C.1 Determine as funções Booleanas que podem ser usada para descrever as seguintes situações:
- (a) Existe uma saída quando a chave A e a chave B ou a chave C é fechada.
 - (b) Existe uma saída quando a chave A ou a chave B é fechada e a chave C ou a chave D é fechada.
 - (c) Existe uma saída quando a chave A é aberta ou a chave B é fechada.
 - (d) Existe uma saída quando a chave A é aberta e a chave B é fechada.
- C.2 Determina as funções Booleanas para cada um dos circuitos mostrados na Figura C.8.

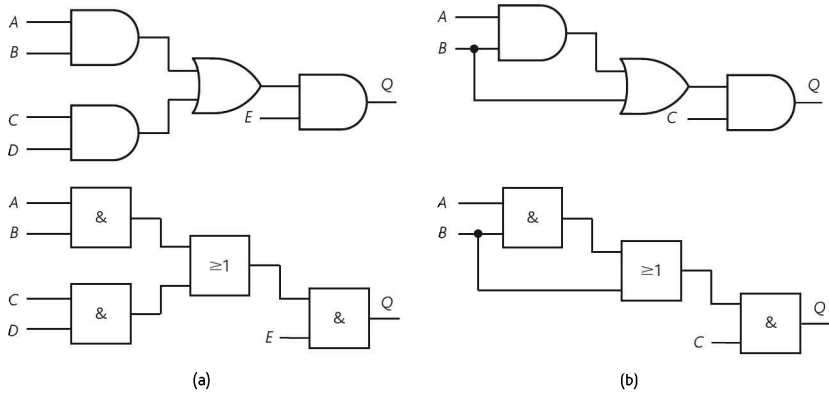


Figura C.8 Problema C.2.

C.3 Construa uma tabela-verdade para a equação Booleana $Q = (A \cdot C + B \cdot C) \cdot (A + C)$.

C.4 Simplifique as seguintes equações Booleanas:

(a) $Q = A \cdot C + A \cdot C \cdot D + C \cdot D$

(b) $Q = A \cdot \bar{B} \cdot D + A \cdot \bar{B} \cdot \bar{D}$

(c) $Q = A \cdot B \cdot C + C \cdot D + C \cdot D \cdot E$

C.5 Use os teoremas de De Morgan para mostrar que uma porta NOR com entradas invertida é equivalente a uma porta AND.

C.6 Desenhe os mapas de Karnaugh para as seguintes tabelas-verdade e determine as equações Booleanas simplificadas para as saídas:

(a)

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	1

(b)

A	B	C	Q
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

C.7 Simplifique as seguintes equações Booleanas através do uso de mapas de Karnaugh:

(a) $Q = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot B \cdot \bar{C}$

(b) $Q = \bar{A} \cdot B \cdot \bar{C} \cdot D + A \cdot \bar{B} \cdot \bar{C} \cdot D + \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D$
 $+ A \cdot B \cdot \bar{C} \cdot D + A \cdot B \cdot \bar{C} \cdot \bar{D} + A \cdot B \cdot C \cdot D$

C.8 Projete um sistema que permite que uma porta seja aberta apenas quando a combinação correta de quatro botões aconteça. Qualquer combinação errada ativa um alarme sonoro.

Apêndice D:

Conjuntos de Instruções

A seguir apresentamos os conjuntos de instruções usados nos microcontroladores M68HC11 da Motorola, 8051 da Intel e PIC16Cxx.

M68HC11

Instrução	Mnemônico	Instrução	Mnemônico
<i>Carga (movimentação)</i>		Troca os acumuladores com o registrador de índice X	XGDX
Carrega o acumulador A	LDAA	Troca os acumuladores com o registrador de índice Y	XGDY
Carrega o acumulador B	LDAB	<i>Decremento/incremento</i>	
Carrega duplo byte no acumulador	LDD	Subtrai 1 do conteúdo da memória	DEC
Carrega o ponteiro de pilha	LDS	Subtrai 1 do conteúdo do acumulador A	DECA
Carrega o registrador de índice X	LDX	Subtrai 1 do conteúdo do acumulador B	DECB
Carrega o registrador de índice Y	LDY	Subtrai 1 do ponteiro de pilha	DES
Retira dado da pilha e coloca no acc A	PULA	Subtrai 1 do registrador de índice X	DEX
Retira dado da pilha e coloca no acc B	PULB	Subtrai 1 do registrador de índice Y	DEY
Retira da pilha o conteúdo do registrador de índice X	PULX	Soma 1 ao conteúdo da memória	INC
Retira da pilha o conteúdo do registrador de índice Y	PULY	Soma 1 ao conteúdo do acumulador A	INCA
<i>Registradores de transferência</i>		Soma 1 ao conteúdo do acumulador B	INCB
Transfere do acumulador A para o B	TAB	Soma 1 ao ponteiro de pilha	INS
Transfere do acumulador B para o A	TBA	Soma 1 ao registrador de índice X	INX
Transfere do ponteiro de pilha para o registrador de índice X	TSX	Soma 1 ao registrador de índice Y	INY
Transfere do ponteiro de pilha para o registrador de índice X	TSY	<i>Rotação/deslocamento</i>	
Transfere do registrador de índice X para o ponteiro de pilha	TXS	Rotaciona bits na memória à esquerda	ROL
Transfere do registrador de índice Y para o ponteiro de pilha	TYS	Rotaciona bits no acumulador A à esquerda	ROLA
		Rotaciona bits no acumulador B à esquerda	ROLB
		Rotaciona bits na memória à direita	ROR

(continua)

Instrução	Mnemônico	Instrução	Mnemônico
Rotaciona bits no acumulador A à direita	RORA	Zerar B	CLRB
Rotaciona bits no acumulador B à direita	RORB	Zerar bits na memória	BCLR
Deslocamento aritmético de bits na memória à esquerda	ASL	Setar bits na memória	BSET
Deslocamento aritmético de bits no acc A à esquerda	ASLA	<i>Armazenamento de registradores</i>	
Deslocamento aritmético de bits no acc B à esquerda	ASLB	Armazena o conteúdo do acc A	STAA
Deslocamento aritmético de bits na memória à direita	ASR	Armazena o conteúdo do acc B	STAB
Deslocamento aritmético de bits no acc A à direita	ASRA	Armazena o conteúdo dos acumuladores	STD
Deslocamento aritmético de bits no acc B à direita	ASRB	Armazena o ponteiro de pilha	STS
Deslocamento lógico de bits na memória à esquerda	LSL	Armazena o registrador de índice X	STX
Deslocamento lógico de bits no acc A à esquerda	LSLA	Armazena o registrador de índice Y	STY
Deslocamento lógico de bits no acc B à esquerda	LSLB	Transfere dados do acc A para a pilha	PSHA
Deslocamento lógico de bits no registrador D à esquerda	LSLD	Transfere dados do acc B para a pilha	PSHB
Deslocamento lógico de bits na memória à direita	LSR	Transfere o conteúdo do registrador de índice X para a pilha	PSHX
Deslocamento lógico de bits no acc A à direita	LSRA	Transfere o conteúdo do registrador de índice Y para a pilha	PSHY
Deslocamento lógico de bits no acc B à direita	LSRB	<i>Lógicas</i>	
Deslocamento lógico de bits no registrador D à direita	LSRD	AND com o conteúdo do acc A	ANDA
<i>Teste de dados setando condições de código</i>		AND com o conteúdo do acc B	ANDB
Teste lógico AND entre acc A & memória	BITA	EX-OR com o conteúdo do acc A	EORA
Teste lógico AND entre acc B & memória	BITB	EX-OR com o conteúdo do acc B	EORB
Compara acc A com acc B	CBA	OR com o conteúdo do acc A	ORAA
Compara acc A com memória	CMPA	OR com o conteúdo do acc B	ORAB
Compara acc B com memória	CMPB	Faz o complemento de um na memória	COM
Compara os acumuladores (dois bytes)	CPD	Faz o complemento de um no acc A	COMA
Compara o registrador de índice X com memória	CPX	Faz o complemento de um no acc B	COMB
Compara o registrador de índice Y com memória	CPY	<i>Aritmética</i>	
Subtrai \$00 da memória	TST	Soma o conteúdo do acc A com acc B	ABA
Subtrai \$00 do acc A	TSTA	Soma o conteúdo do acc B com registrador de índice X	ABX
Subtrai \$00 do acc B	TSTB	Soma o conteúdo do acc B com registrador de índice Y	ABY
<i>Interrupção</i>		Soma memória ao acc A sem carry	ADDA
Zera máscara de interrupção	CLI	Soma memória ao acc B sem carry	ADDB
Seta máscara de interrupção	SEI	Soma memória aos dois acumuladores sem carry	ADDD
Interrupção por software	SWI	Soma memória ao acc A com carry	ADCA
Retorno da interrupção	RTI	Soma memória ao acc B com carry	ADCB
Espera por interrupção	WAI	Ajuste decimal	DAA
<i>Complemento e clear (zerar)</i>		Subtrai o conteúdo do acc B a partir do acc A	SBA
Zerar memória	CLR	Subtrai a memória a partir do acc A com carry	SBCA
Zerar A	CLRA	Subtrai a memória a partir do acc B com carry	SBCB
		Subtrai a memória a partir dos dois acumuladores	SUBA
		Faz o complemento de dois do acc A	SUBB
		Faz o complemento de dois do acc B	SUBD
		Multiplicação não sinalizada do acc A pelo acc B	MUL

Instrução	Mnemônico	Instrução	Mnemônico
Inteiro não sinalizado dividido pelo reg. índice X	IDIV	<i>Salto e desvio</i>	
Inteiro fracionário dividido pelo reg. índice X	FDIV	Saltar para o endereço	JMP
<i>Desvio condicional</i>		Saltar para a sub-rotina	JSR
Desvia se menos	BMI	Retornar da sub-rotina	RTS
Desvia se mais	BPL	Desviar para a sub-rotina	BSR
Desvia se <i>overflow</i> setado	BVS	Desviar sempre	BRA
Desvia se <i>overflow</i> zerado	BVC	Desviar nunca	BRN
Desvia se menor do que zero	BLT	Desviar bits set	BRSET
Desvia se maior ou igual a zero	BGE	Desviar bits clear	BRCLR
Desvia se menor ou igual a zero	BLE	<i>Código condicional</i>	
Desvia se maior do que zero	BGT	Zerar <i>carry</i>	CLC
Desvia se for igual	BEQ	Zerar <i>overflow</i>	CLV
Desvia se for diferente	BNE	Setar <i>carry</i>	SEC
Desvia se for maior	BHI	Setar <i>overflow</i>	SEV
Desvia se menor ou igual	BLS	Transferir do acc A para o reg. código de condição	TAP
Desvia se maior ou igual	BHS	Transferir do reg. código de condição para o acc A	TPA
Desvia se for menor	BLO	<i>Diversos</i>	
Desvia se <i>carry</i> for zero	BCC	Sem operação	NOP
Desvia se <i>carry</i> for um	BCS	Parar o processamento	STOP
		Modo de teste especial	TEST

Nota: O número de bits no registrador depende do processador. Um microprocessador de 8 bits geralmente tem registradores de 8 bits. Algumas vezes, dois registradores de dados podem ser usados juntos para dobrar o número de bits. Este registrador combinado é denominado *double register*.

Intel 8051

Instrução	Mnemônico	Instrução	Mnemônico
<i>Transferência de dados</i>			
Acumulador recebe data	MOV A, #data	Move o conteúdo de uma posição da memória RAM interna cujo endereço é direct para outra posição da RAM interna endereçada pelo conteúdo de RO ou R1	MOV @Ri, direct
Acumulador recebe o conteúdo do registrador	MOV A, Rn	Registrador recebe data	MOV Rn, #data
Acumulador recebe um byte da memória RAM interna cujo endereço é direct	MOV A, direct		MOV direct1, direct2
Acumulador recebe um dado da memória RAM interna endereçado pelo conteúdo de RO ou R1	MOV A, @Ri	O conteúdo da posição de memória RAM interna com endereço direct2 é copiado para outra posição da memória interna cujo endereço é direct1	
O conteúdo de A é copiado numa posição da memória RAM interna cujo endereço é direct	MOV direct, A	Uma posição de memória RAM interna, cujo endereço é direct, recebe um dado endereçado pelo conteúdo de RO ou R1	MOV direct, @Ri
O conteúdo do acumulador é copiado para uma posição de memória RAM interna cujo endereço é o conteúdo de RO ou R1	MOVX @Ri, A	O conteúdo de Rn é copiado numa posição de memória RAM interna cujo endereço é direct	MOV direct, Rn
Registrador recebe o conteúdo de A	MOV Rn, A		

(continua)

Instrução	Mnemônico	Instrução	Mnemônico
Uma posição de memória RAM interna, cujo endereço é direct, recebe data	MOV direct, #data	Desvia para a rotina se o conteúdo de Rn e data igual a data forem diferentes	CJNE Rn, #data, rel
data é copiado para uma posição de memória RAM interna cujo endereço é o conteúdo de R0 ou R1	MOV @Ri, #data	Desvia para a rotina se o conteúdo de Rn for diferente do conteúdo da posição de memória cujo endereço é o conteúdo de R1 ou R0	CJNE Rn, @Ri, #data, rel
Carrega o registrador DPTR com o valor data16	MOV DPTR, #data16	Decrementa o registrador e desvia para a rotina se o acumulador for diferente de zero	DJNZ Rn, rel
O conteúdo do acumulador é somado ao de DPTR, formando o endereço de um dado na memória de programa, sendo que tal dado é carregado no acumulador	MOVC A, @A+DPTR	Decrementa o conteúdo da posição de memória dada por direct. Se o resultado não for nulo, ocorre um desvio; caso contrário, a instrução seguinte é executada	DJNZ direct, rel
O conteúdo de DPTR é o endereço de um dado da RAM externa que é carregado no acumulador	MOVX A, @DPTR	Desvia se o <i>flag carry</i> estiver em 1	JC rel
O conteúdo do acumulador é copiado na posição de memória RAM externa cujo endereço é dado pelo conteúdo de DPTR	MOVX @DPTR, A	Desvia se o <i>flag carry</i> estiver em 0	JNC rel
Troca o conteúdo do acumulador com o conteúdo de uma posição de memória RAM interna cujo endereço é direct	XCH A, direct	Desvia se o bit estiver em 1	JB bit, rel
Troca o conteúdo do acumulador com o conteúdo de uma posição de memória RAM interna cujo endereço é Ri.	XCH A, @Ri	Desvia se o bit estiver em 0	JNB bit, rel
Troca o conteúdo do acumulador com o conteúdo do registrador	XCH A, Rn	Desvia se o bit estiver em 1 e automaticamente zera o bit	JBC bit, rel
Coloca na pilha o conteúdo da posição de memória RAM interna cujo endereço é direct	PUSH direct	<i>Chamada de sub-rotina</i> Desvia para uma sub-rotina. Essa sub-rotina deve estar no máximo a 1 Kbyte de distância do ponto de chamada	ACALL addr11
Retira da pilha um dado e armazena na posição de memória RAM interna cujo endereço é direct	POP direct	Desvia para uma sub-rotina	LCALL addr16
<i>Desvio</i>		Retorna de uma sub-rotina	RET
Desvia para uma rotina. Essa rotina deve estar no máximo a 1 Kbyte de distância do ponto de chamada	AJMP addr11	Retorna de uma sub-rotina de atendimento de Interrupção	RETI
Desvia para uma rotina	LJMP addr16	<i>Manipulação de bit</i> Zera o <i>flag carry</i> em PSW	CLR C
Desvia para uma rotina. Essa rotina deve estar no máximo a 128 bytes de distância do ponto de chamada	SJMP rel	Zera o bit em uma posição de memória RAM interna com bits endereçáveis	CLR bit
Desvia para uma rotina cujo endereço é dado pela soma do conteúdo de DPTR com o conteúdo de A	JMP @A + DPTR	Torna 1 o <i>flag carry</i> no PSW	SETB C
Desvia para a rotina se o acumulador for zero	JZ rel	Torna 1 o bit em uma posição de memória RAM interna com bits endereçáveis	SETB bit
Desvia para a rotina se o acumulador for diferente de zero	JNZ rel	Complementa o <i>flag carry</i> em PSW	CPL C
Desvia para a rotina se o conteúdo de A e direct forem diferentes	CJNE A, direct, rel	Lógica AND entre o <i>flag carry</i> e o bit em uma posição de memória RAM interna com bits endereçáveis	ANL C, bit
Desvia para a rotina se o conteúdo de A não for igual a data	CJNE A, #data, rel	Lógica AND entre o <i>flag carry</i> e o complemento do bit em uma posição de memória RAM interna com bits endereçáveis	ANL C, /bit
		Lógica OR entre o <i>flag carry</i> e o bit em uma posição de memória RAM interna com bits endereçáveis	ORL C, bit
		Lógica OR entre o <i>flag carry</i> e o complemento do bit em uma posição de memória RAM interna com bits endereçáveis.	ORL C, /bit

Instrução	Mnemônico	Instrução	Mnemônico
Acerta o <i>flag carry</i> com o valor do bit de uma posição de memória RAM interna com bits endereçáveis	MOV C,bit	Lógica EX-OR entre o conteúdo do acumulador e data	XRL A, #data
Acerta o bit de uma posição de memória RAM interna com bits endereçáveis com o <i>flag carry</i>	MOV bit,C	Lógica EX-OR entre o conteúdo do acumulador e o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i>	XRL A, direct
<i>Operações lógicas</i>			
Lógica AND entre o conteúdo de uma posição de memória RAM interna, cujo endereço é <i>direct</i> , e o conteúdo do acumulador	ANL direct, A	Lógica EX-OR entre o conteúdo do acumulador e uma posição de memória RAM interna cujo endereço é o conteúdo de Ri	XRL A,@Ri
Lógica AND entre o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i> e data	ANL direct, #data	Lógica EX-OR entre o conteúdo do acumulador e o conteúdo do registrador	XRL A, Rn
Lógica AND entre o conteúdo do acumulador e data	ANLA, #data	<i>Adição</i>	
Lógica AND entre o conteúdo do acumulador e o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i>	ANLA, direct	Ao conteúdo de A é somado o valor data	ADD A, #data
Lógica AND entre o conteúdo do acumulador e uma posição de memória RAM interna cujo endereço é <i>direct</i>	ANLA, @Ri	Ao conteúdo do acumulador é somado o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i>	ADD A, direct
Lógica AND entre o conteúdo do acumulador e uma posição de memória RAM interna cujo endereço é o conteúdo de Ri	ANLA, Rn	Ao conteúdo de A é somado o conteúdo de uma posição de memória RAM interna cujo endereço é o conteúdo de Ri	ADD A, @Ri
Lógica OR entre o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i> e o conteúdo do acumulador	ORL direct, A	Ao conteúdo do acumulador é somado o conteúdo do registrador	ADD A, Rn
Lógica OR entre o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i> e data	ORL direct, #data	Ao conteúdo de A são somados o valor data e o conteúdo do <i>flag carry</i>	ADDC A, #data
Lógica OR entre o conteúdo do acumulador e data	ORLA, #data	Ao conteúdo do acumulador são somados o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i> e o conteúdo do <i>flag carry</i> C	ADDC A, direct
Lógica OR entre o conteúdo do acumulador e o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i>	ORLA, direct	Ao conteúdo de A são somados o conteúdo de uma posição de memória RAM interna cujo endereço é o conteúdo de Ri e o conteúdo do <i>flag carry</i> C	ADDC A, @Ri
Lógica OR entre o conteúdo do acumulador e uma posição de memória RAM interna cujo endereço é o conteúdo de Ri	ORLA, @Ri	Ao conteúdo do acumulador são somados o conteúdo do registrador e o conteúdo do <i>flag carry</i> C	ADDC A, Rn
Lógica OR entre o conteúdo do acumulador e o conteúdo do registrador	ORLA, Rn	<i>Subtração</i>	
Lógica EX-OR entre o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i> e o conteúdo do acumulador	XRL direct,A	Do conteúdo de A são subtraídos o valor data e o <i>flag carry</i>	SUBB A, #data
Lógica EX-OR entre o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i> e o conteúdo do acumulador	XRL direct, #data	Do conteúdo do acumulador são subtraídos o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i> e o <i>flag carry</i>	SUBB A, direct
Lógica EX-OR entre o conteúdo de uma posição de memória RAM interna cujo endereço é <i>direct</i> e data	XRL direct, #data	Do conteúdo de A são subtraídos o conteúdo de uma posição de memória RAM interna cujo endereço é o conteúdo de Ri e o <i>flag carry</i>	SUBB A @Ri
		<i>Multiplicação e divisão</i>	
		O conteúdo do acumulador e do registrador B são multiplicados sem	MUL AB

Instrução	Mnemônico	Instrução	Mnemônico
<p>sinal. Da palavra de 16 bits resultante, os 8 bits LSB são armazenados em A e os 8 bits MSB são armazenados em B</p> <p>O conteúdo do acumulador é dividido (sem sinal) pelo conteúdo do registrador B. Em A é armazenada a parte inteira do quociente da divisão, e em B, o resto</p> <p><i>Operações matemáticas decimais</i></p> <p>Troca os 4 bits menos significativos do conteúdo de A com os 4 bits menos significativos do conteúdo da posição de memória cujo endereço é o conteúdo de Ri. Os outros bits ficam inalterados</p> <p>Troca os 4 bits menos significativos do conteúdo de A com o 4 bits mais significativos</p> <p>O conteúdo de A é convertido para um número decimal de dois dígitos de quatro bits cada, que são armazenados em A</p> <p><i>Incremento e decremento</i></p> <p>Ao conteúdo do acumulador é somado 1</p> <p>Ao conteúdo da posição de memória RAM interna cujo endereço é direct é somado 1</p> <p>Ao conteúdo da posição de memória RAM interna cujo endereço é o conteúdo de Ri é somado 1</p>	<p>DIV AB</p> <p>XCHD A, @Ri</p> <p>SWAP A</p> <p>DA A</p> <p>INC A</p> <p>INC direct</p> <p>INC @Ri</p>	<p>Ao conteúdo do registrador é somado 1</p> <p>Do conteúdo do acumulador é subtraído 1</p> <p>Do conteúdo da posição de memória RAM interna cujo endereço é direct é subtraído 1</p> <p>Do conteúdo da posição de memória RAM interna cujo endereço é o conteúdo de Ri é subtraído 1</p> <p>Do conteúdo do registrador é subtraído 1</p> <p>Ao conteúdo do registrador DPTR é somado 1</p> <p><i>Operações de clear e complemento</i></p> <p>Complementa o conteúdo do acumulador</p> <p>Zera o conteúdo do acumulador</p> <p><i>Operações de rotação</i></p> <p>Rotaciona o conteúdo do acumulador para a direita</p> <p>Rotaciona o conteúdo do acumulador para a direita através do carry</p> <p>Rotaciona o conteúdo do acumulador para a esquerda. O bit 7 é carregado no bit 0</p> <p>Rotaciona o conteúdo do acumulador para a esquerda através do carry</p> <p><i>Sem operação</i></p> <p>Não executa nada</p>	<p>INC Rn</p> <p>DEC A</p> <p>DEC direct</p> <p>DEC @Ri</p> <p>DEC Rn</p> <p>INC DPTR</p> <p>CPL A</p> <p>CLR A</p> <p>RR A</p> <p>RRC A</p> <p>RL A</p> <p>RLC A</p> <p>NOP</p>

Nota: Um valor precedido de # é um número, #data16 é uma constante de 16 bits; Rn se refere ao conteúdo de um registrador; @Ri se refere ao valor da memória para o qual o registro aponta; DPTR é o ponteiro de dados; direct é a posição de memória na qual o dado usado pela instrução é encontrado.

PIC16Cxx

Instrução	Mnemônico	Instrução	Mnemônico
Soma number com o número no registrador W e guarda o resultado em W	addlw number	Faz a operação AND entre o conteúdo de W e o conteúdo de FileReg e guarda o resultado em FileReg	andwf FileReg,f
Soma o número no registrador W ao número em FileReg e guarda o resultado em FileReg	addwf FileReg, f	Zera um bit em FileReg	bcf FileReg,bit
Soma o número no registrador W ao número em FileReg e guarda o resultado em W	addwf FileReg, w	Seta um bit em FileReg	bsf FileReg,bit
Faz a operação AND entre number e o conteúdo de W e guarda o resultado em W	andlw number	Testa um bit em FileReg e salta a próxima instrução se o bit for 0	btfsc FileReg, bit
		Testa um bit em FileReg e salta a próxima instrução se o bit for 1	btifss File Reg, bit
		Chama uma sub-rotina (AnySub)	call AnySub
		Zera o número em FileReg	clrf FileReg

Instrução	Mnemônico	Instrução	Mnemônico
Zera o conteúdo de W	clrw	Sem operação	nop
Zera o conteúdo do temporizador <i>watchdog</i>	clrwdt	Retorna de uma sub-rotina e habilita as interrupções globais	refie
Complementa o conteúdo em FileReg e deixa o resultado em FileReg	comf FileReg,f	Retorna de uma sub-rotina com number em W	retlw number
Decrementa FileReg	decf FileReg,f	Retorna de uma sub-rotina	return
Decrementa FileReg e, se o resultado for zero, salta a próxima instrução	decfsz FileReg,f	Rotaciona os bits no FileReg para a esquerda através do <i>carry</i>	rlf FileReg,f
Vai para o ponto rotulado por label	goto label	Rotaciona os bits no FileReg para a direita através do <i>carry</i>	rrf FileReg,f
Incrementa FileReg e coloca o resultado em FileReg	incf FileReg,f	Coloca o PIC em repouso, que é o modo de baixo consumo de energia	sleep
Faz a operação OR entre number e o conteúdo de W	iorlw number	Subtrai W de number	sublw number
Faz a operação OR entre os conteúdos de W e FileReg e coloca o resultado em FileReg	iorlw number	Subtrai W de FileReg e coloca o resultado em FileReg	subwf FileReg,f
Move (copia) o conteúdo de FileReg para W	movf FileReg,w	Troca os 4 bits mais significativos com os 4 bits menos significativos de FileReg	swapf FileReg,f
Move (copia) number para W	movlw number	Usa o número em W para especificar quais bits são entrada e quais são saída	tris PORTX
Move (copia) o conteúdo de W para FileReg	movwf FileReg	Faz a operação EX-OR de number com W	xorlf number
		Faz a operação EX-OR de W com FileReg e coloca o resultado em FileReg	xorwf FileReg,f

Nota: f é usado para registrador de arquivo, w para registrador de trabalho, e b para bit. Os mnemônicos indicam os tipos de operandos envolvidos, por exemplo, movlw indica a operação de movimentação com lw indicando que um valor literal, ou seja, um número (number), está envolvido no registrador de trabalho w; movwf indica a operação de movimentação quando o registrador de trabalho e um registrador de arquivo são envolvidos.

Apêndice E:

Funções de Biblioteca da Linguagem C

Apresentamos a seguir algumas das funções de biblioteca da linguagem C mais comuns. Esta não é uma lista completa de todas as funções de uma biblioteca ou uma lista completa de todas as bibliotecas disponíveis para um compilador qualquer.

<ctype.h>

isalnum	int isalnum(int ch)	Testa caracteres alfanuméricos, retornando um número diferente de zero se o argumento for uma letra ou um dígito, e retornando um zero se o caractere não for alfanumérico.
isalpha	int isalpha(int ch)	Testa caracteres alfabéticos, retornando valor diferente de zero se for uma letra do alfabeto; caso contrário, retorna 0.
iscntrl	int iscntrl(int ch)	Testa caractere de controle, retornando número diferente de zero se estiver entre 0 e 0x1F ou se for igual a 0x7F (DEL); caso contrário, retorna 0.
isdigit	int isdigit(int ch)	Testa caractere decimal, retornando um número diferente de 0 se um dígito (0 a 9); caso contrário, retorna 0.
isgraph	int isgraph(int ch)	Testa caractere imprimível (exceto espaço), retornando um número diferente de 0 se imprimível; caso contrário, retorna 0.
islower	int islower(int ch)	Testa caractere minúsculo, retornando um número diferente de 0 se minúsculo; caso contrário, retorna 0.
isprint	int isprint(int ch)	Testa caractere imprimível (incluindo espaço), retornando um número diferente de zero se imprimível; caso contrário, retorna 0.
ispunct	int ispunct(int ch)	Testa caractere de pontuação, retornando um número diferente de zero se for caractere de pontuação; caso contrário, retorna 0.

isspace	int isspace(int ch)	Testa caractere espaço, retornando um número diferente de zero se for espaço, tab, alimentação de formulário, retorno do carro ou nova linha; caso contrário, retorna 0.
isupper	int isupper(int ch)	Testa caractere maiúsculo; retornando um número diferente de 0 se maiúsculo, caso contrário, retorna 0.
isxdigit	int isxdigit(int ch)	Testa caractere hexadecimal, retornando um número diferente de 0 se for um dígito hexa; caso contrário, retorna 0.

<math.h>

acos	double acos(double arg)	Retorna o arco co-seno do argumento.
asin	double asin(double arg)	Retorna o arco seno do argumento.
atan	double atan(double arg)	Retorna o arco tangente do argumento. Requer um argumento.
atan2	double atan2(double y, double x)	Retorna o arco tangente de y/x
ceil	double ceil(double num)	Retorna o menor inteiro que não é menor do que o número.
cos	double cos(double arg)	Retorna o co-seno do argumento. O valor do argumento tem que ser em radianos.
cosh	double cosh(double arg)	Retorna o co-seno hiperbólico do argumento.
exp	double exp(double arg)	Retorna e^x , em que x é o argumento.
fabs	double fabs(double num)	Retorna o valor absoluto do número.
floor	double floor(double num)	Retorna o maior inteiro que não for maior do que o número.
fmod	double fmod(double x, double y)	Retorna o resto de ponto flutuante de x/y.
ldexp	double ldexp(double x, int y)	Retorna x vezes 2^y .
log	double log(double num)	Retorna o logaritmo natural do número.
log10	double log10(double num)	Retorna o logaritmo de base 10 do número.
pow	double pow(double base, double exp)	Retorna a base elevada ao expoente.
sin	double sin(double arg)	Retorna o seno do argumento.
sinh	double sinh(double arg)	Retorna o seno hiperbólico do argumento.
sqrt	double sqrt(double num)	Retorna a raiz quadrada do número.
tan	double tan(double arg)	Retorna a tangente do número.
tanh	double tanh(double arg)	Retorna a tangente hiperbólica do número.

<stdio.h>

getchar	int getchar(void)	Retorna o próximo caractere digitado no teclado.
gets	char gets(char *str)	Lê caracteres inseridos no teclado até um retorno do carro (enter) ocorrer e os armazena em um arranjo apontado por str.
printf	int printf(char *str, ...)	Mostra a string apontada por str.
puts	int puts(char *str)	Mostra a string apontada por str.
scanf	int scanf(char *str, ...)	Lê a informação da variável apontada pelo argumento que segue a string de controle.

<stdlib.h>

abort	void abort(void)	Provoca o término imediato do programa.
abs	int abs(int num)	Retorna o valor absoluto do número inteiro.
bsearch	void bsearch(const void *key, const void *base, size_t num, size_t size, int(*compare)(const void *, const void *))	Faz uma busca binária no arranjo escolhido apontado pela base e retorna um ponteiro do primeiro membro que coincide com a tecla apontada por key. O número de elementos no arranjo é especificado por num, e o tamanho em bytes de cada elemento é dado por size.
calloc	void *calloc(size_t num, size_t size)	Aloca memória suficiente para um arranjo de num objetos de tamanho dado por size, retornando um ponteiro para o primeiro byte da memória alocada.
exit	void exit(int status)	Provoca uma terminação normal imediata do programa. O valor do status é passado pelo processo de chamada.
free	void free(void *ptr)	Libera a memória alocada apontada por ptr.
labs	long labs(long num)	Retorna o valor absoluto do número long int.
malloc	void *malloc(size_t size)	Retorna um ponteiro para o primeiro byte da memória de tamanho dado por size que foi alocada.
qsort	void qsort(void *base, size_t num, size_t size, int(*compare)(const void *, const void *))	Seleciona o arranjo apontado por base. O número de elementos no arranjo é dado por num, e o tamanho em bytes de cada elemento, por size.
realloc	void *realloc(void *ptr, size_t size)	Altera o tamanho da memória alocada apontada por ptr para a especificada por size.

Nota: size_t é o tipo para variáveis "size of" e geralmente representa o tamanho de outro parâmetro ou objeto.

<time.h>

asctime	char *asctime(const struct tm *ptr)	Converte o tempo a partir de uma forma de estrutura para uma string de caracteres apropriada para display, retornando um ponteiro para a string.
clock	clock_t clock(void)	Retorna o número de ciclos de clock que ocorreram desde o início da execução do programa.
ctime	char *ctime(const time_t *time)	Retorna um ponteiro para uma string da forma dia mês data hora:minuto:segundo ano/n/0 apontando para os números de segundos decorridos desde 00:00:00 (tempo médio em Greenwich).
difftime	double difftime(time_t time 2, time_t time 1)	Retorna a diferença em segundos entre time 1 e time 2.
gmtime	struct tm *gmtime(const time_t *time)	Retorna um ponteiro para o tempo convertido da forma long int para uma forma de estrutura.

localtime	struct tm *localtime (const time_t *time)	Retorna um ponteiro para o tempo convertido da forma long int para uma forma de estrutura no tempo local.
time	time_t time(time_t *system)	Retorna o calendário atual do sistema.

Nota: time_t e clock_t são usados como o tipo para variáveis "tempo de" e "número de ciclos de".

Apêndice F: MATLAB e SIMULINK

F.1 MATLAB

Os softwares de computadores podem ser usados para auxiliar nos cálculos e no modelamento de sistemas; um programa que frequentemente é usado é o MATLAB. Neste apêndice, fazemos uma breve introdução ao MATLAB (marca registrada da Mathworks Inc.) versão 4.0 ou posterior. Para obter mais informações o leitor deve consultar o guia do usuário ou livros como *The MATLAB Handbook* por Eva Pärt-Enander, Anders Sjöberg, Bo Melin e Pernilla Isaksson (Addison-Wesley, 1996) e *Using MATLAB to Analyse and Design Control Systems*, 2ª edição, por Naomi Ehrlich Leonard e Willina S. Levine (Addison-Wesley, 1995).

Os comandos são inseridos digitando-os após o prompt e, em seguida, pressionando-se ENTER ou a tecla RETURN para que o comando seja executado. Na descrição dos comandos a seguir, o acionamento da tecla ENTER ou RETURN não é repetido, mas deve ser considerado em todos os casos. Para iniciar o MATLAB em sistemas Windows ou Macintosh, clique no ícone MATLAB; caso contrário, digite matlab. A tela apresentará o prompt do MATLAB (\gg). Para finalizar o MATLAB, digite *quit* ou *exit* após o prompt. Como o MATLAB faz distinção entre letras maiúsculas e minúsculas (*case sensitive*), as letras minúsculas devem ser usadas em todos os comandos.

Digitando *help* após o prompt, ou selecionando *help* a partir da barra de menu no topo da janela do MATLAB, é mostrada uma lista geral de tópicos. Para obter o *help* de um tópico em particular, como, por exemplo, *exponentials* (exponencial), digite *help exp*. Digitar *lookfor* mais um tópico faz com que o MATLAB pesquise por informações daquele tópico, por exemplo, *lookfor integ* mostrará os comandos que podem ser considerados para integração.

Em geral, as operações matemáticas são inseridas no MATLAB da mesma forma que escrevemos no papel. Por exemplo,

$$\gg a = 4/2$$

resulta na resposta

$$a = \\ 2$$

e

$$\gg a = 3*2$$

resulta na resposta

$$a = \\ 6$$

As operações são realizadas na seguinte ordem: operações de potência (^), multiplicação (*), divisão (/), adição (+), subtração (-). A precedência de operadores é da esquerda para a direita, mas os parênteses () podem ser usados para interferir na ordem. Por exemplo,

$$\gg a = 1 + 2^3/4*5$$

resulta na resposta

$$a = \\ 11$$

porque temos $2^3/4$ multiplicado por 5 e então somado com 1, ao passo que

$$\gg a = 1 + 2^3/(4*5)$$

resulta na resposta

$$a = \\ 1,4$$

porque temos 2^3 dividido pelo produto entre 4 e 5, para então ser somado com 1.

Mostramos a seguir algumas das funções matemáticas disponíveis no MATLAB:

abs(x)	fornece o valor absoluto de x , ou seja, $ x $
exp(x)	fornece a exponencial de x , ou seja, e^x
log(x)	fornece o logaritmo natural de x , ou seja, $\ln x$
log10(x)	fornece o logaritmo de base 10 de x , ou seja, $\log_{10}x$
sqrt(x)	fornece a raiz quadrada de x , ou seja, \sqrt{x}
sin(x)	fornece o sen x com x em radianos
cos(x)	fornece o cos x com x em radianos
tan(x)	fornece a tg x com x em radianos
asin(x)	fornece o arco seno de x , ou seja, $\sin^{-1}x$
acos(x)	fornece o arco co-seno de x , ou seja, $\cos^{-1}x$
atan(x)	fornece o arco tangente de x , ou seja, $\tan^{-1}x$
csc(x)	fornece $1/\sin x$
sec(x)	fornece $1/\cos x$
cot(x)	fornece $1/\tan x$

π é inserido digitando-se *pi*.

Em vez de escrever uma série de comandos no prompt, pode-se escrever um arquivo texto e então os comandos são executados no momento que o MATLAB acessar o arquivo. O termo M-file é usado, visto que estes arquivos texto que contêm uma lista de comandos consecutivos do MATLAB têm um sufixo .m. Ao escrever um

arquivo deste, a primeira linha tem que ser iniciada com a palavra *function* seguida da identificação do nome (*name*) da função e da entrada (*input*) e saída (*output*) na forma

$$\text{function [output] = function name [input]}$$

por exemplo, `function y = cotan(x)`, que é o arquivo usado para determinar o valor de y dado pela cotangente de x . Este arquivo pode ser chamado em alguma sequência de comandos do MATLAB escrevendo-se o nome seguido da entrada, como, por exemplo, `cotan(x)`. De fato, esta função já está incluída no MATLAB e é usada quando for solicitada a cotangente de x . Entretanto, o arquivo tem que ser escrito pelo usuário. Uma função que tem múltiplas entradas deve ter todas listadas na declaração da função. Da mesma forma, uma função que tem que retornar mais de um valor tem que ser listada todas as saídas.

As linhas que começam com `%` são de comentários; elas não são interpretadas pelo MATLAB como comandos. Por exemplo, suponha que escrevemos um programa para determinar os valores de raiz quadrada de uma única coluna de dados. O programa poderia ser como este:

```
function y = rms(x)
% rms – Root mean square (valor médio quadrático)
% rms(x) fornece o valor médio quadrático dos
% elementos da coluna x.
xs = x^2;
s = size(x);
y = sqrt(sum(xs)/s);
```

Fizemos com que `xs` seja os valores quadráticos de cada valor de x . O comando `s=size(x)` fornece o tamanho, ou seja, o número de entradas na coluna de dados. O comando `y=sqrt(sum(xs)/s (1))` obtém a raiz quadrada da soma de todos os valores `xs` dividido por `s`. O comando `;` é usado ao final de cada linha de programa.

O MATLAB fornece vários *toolboxes* contendo coleções de arquivos *M-files*. O *toolbox* mais relevante relacionado a este livro é o Control System (sistema de controle). Ele pode ser usado para mostrar a resposta de sistemas a impulsos, degraus, rampas, etc., juntamente com a análise de Bode e Nyquist, lugar das raízes, etc. Por exemplo, para fazer um gráfico de Bode de um sistema descrito pela função de transferência $4/(s^2 + 2s + 3)$, o programa é

```
%Gera o gráfico de Bode para  $G(s) = 4/(s^2 + 2s + 3)$ 
num 4
den [1 2 3];
bode(num,den)
```

O comando `bode(num,den)` produz o gráfico de Bode do ganho em dB em relação à frequência em rad/s em uma escala logarítmica e a fase em graus em relação à frequência em rad/s em uma escala logarítmica.

F.1.1 Fazendo gráficos

Os gráficos lineares bidimensionais podem ser gerados usando-se o comando `plot(x,y)`; este traça o gráfico com os valores de x e y . Por exemplo,

```
x=[0 1 2 3 4 5];
y=[0 1 4 9 16 25];
plot(x,y)
```

Para traçar o gráfico de uma função, seja ela padrão ou definida pelo usuário, usamos o comando `fplot(nome da função,lim)`, em que `lim` determina o intervalo do gráfico, ou seja, os valores mínimo e máximo de x .

O comando `semilogx(x,y)` gera um gráfico dos valores de x e y usando uma escala logarítmica para x e uma escala linear para y . O comando `semilogy(x,y)` gera um gráfico dos valores de x e y usando uma escala linear para x e uma escala logarítmica para y . O comando `loglog(x,y)` gera um gráfico dos valores de x e y usando uma escala logarítmica para x e para y . O comando `polar(theta,r)` traça um gráfico em coordenadas polares com θ sendo o argumento em radianos e r sendo a magnitude.

O comando `subplot` permite que uma janela de gráfico seja dividida em subjanelas e os gráficos sejam colocados um em cada janela. Por exemplo,

```
x=(0 1 2 3 4 5 6 7);
y=exp(x);
subplot(2,1,1);plot(x,y);
subplot(2,1,2);semilogy(x,y);
```

Os inteiros m , n , p são dados com o comando `subplot`; os dígitos m e n indicam que a janela do gráfico será dividida em $m \times n$ janelas menores, em que m é o número de linhas e n é o número de colunas, e o dígito p indica a janela a ser usada para o gráfico. As subjanelas são numeradas da esquerda para a direita e de cima para baixo. Portanto, a sequência anterior de comandos divide a janela em duas, com um gráfico acima do outro; o gráfico superior é linear e o inferior é semilogarítmico.

O número e o estilo das linhas de grade, a cor do gráfico e a adição de texto ao gráfico podem ser selecionados. O comando `print` é usado para imprimir uma cópia de um gráfico em um arquivo ou uma impressora. Isto pode ser feito selecionando-se o item na barra de menu do arquivo na janela e, em seguida, selecionando-se a opção `print`.

F.1.2 Funções de transferência

O programa MATLAB a seguir ilustra como uma função de transferência pode ser inserida e apresentada na tela:

```
% G(s)=4(s+10)/(s+5)(s+15)
num=4*[1 10];
den=conv([1 5],[1 15]);
printsys(num,den,'s')
```

O comando `num` é usado para indicar o numerador da função de transferência em ordem decrescente das potências de s . O comando `den` é usado para indicar o denominador em ordem decrescente das potências de s para cada um dos dois polinômios no denominador. O comando `conv` multiplica dois polinômios; neste caso, eles são $(s + 5)$ e $(s + 15)$. O comando `printsys` mostra a função de transferência com o numerador e o denominador especificados e escritos no domínio s .

Algumas vezes precisamos determinar os polos e zeros de uma função de transferência apresentada como uma razão de dois polinômios. Neste caso, podemos usar

```
% Determinando os polos e zeros de uma função de transferência
% G(s)=(5s^2 + 3s + 4)/(s^3 + 2s^2 + 4s + 7)
num=[5 3 4];
den=[1 2 4 7];
[z,p,k]=tf2zp(num,den)
```

`[z,p,k]=tf2zp(num,den)` é o comando para determinar e mostrar os zeros (z), polos (p) e ganho (k) da função de transferência inserida.

O MATLAB pode ser usado para gerar os gráficos que mostram a resposta de um sistema a diferentes entradas. Por exemplo, o programa a seguir apresenta a resposta de um sistema a uma entrada em degrau $u(t)$ com uma função de transferência específica:

```
% Mostra a resposta a uma entrada em degrau para um sistema com
% função de transferência G(s)=5/(s^2 + 3s + 12)
num=5;
den=[1 3 12];
step(num,den)
```

F.1.3 Diagramas em bloco

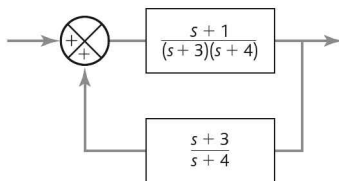


Figura F.1 Diagrama em bloco.

Frequentemente os sistemas de controle são representados como uma série de blocos interconectados, cada um tendo características específicas. O MATLAB permite que sistemas sejam construídos a partir de blocos interconectados. Os comandos usados são *loop* quando um bloco com uma função de transferência em malha aberta tem uma realimentação unitária. Se a realimentação não for unitária é usado o comando *feedback*; por exemplo, para a Figura F.1 temos o programa

```
% Sistema com malha de realimentação
ngo=[1 1];
dgo=conv([1 3],[1 4]);
nh=[1 3];
dh=[1 4];
[ngc2,dgc2]=feedback(ngo,dgo,nh,dh)
printsys(ngc2,dgc2,'s')
```

`ngo` e `dgo` indicam o numerador e o denominador da função de transferência em malha aberta $G_0(s)$, `nh` e `dh` são o numerador e o denominador da função de transferência com malha de realimentação $H(s)$. O resultado do programa é mostrar a função de transferência do sistema como um todo.

O comando *series* é usado para indicar que dois blocos estão em série em um determinado percurso; o comando *parallel* indica que eles estão em paralelo.

F.2 SIMULINK

O SIMULINK é usado em conjunto com o MATLAB para especificar sistemas através da “conexão” de blocos na tela em vez de, como anteriormente, escrever uma série de comandos para gerar a descrição do diagrama em blocos. Uma vez iniciado o MATLAB, o SIMULINK é executado através do comando »simulink. Isto faz abrir a janela de controle SIMULINK com seus ícones e menus *pull-down* na barra superior. Clique em *file*, em seguida clique em *new* a partir do menu *drop-down*. Isto abre uma janela na qual um sistema pode ser montado.

Para iniciar a montagem dos blocos desejados, volte à janela de controle e dê um duplo clique no ícone linear. Clique e arraste o ícone da função de transferência *Fcn* para a janela sem título. Se precisar de um bloco de ganho, clique e arraste o ícone *gain* para a janela sem título. Faça o mesmo para o ícone *sum* e, talvez, o ícone *integrator*. Assim, arraste todos os ícones desejados para a janela sem título. Agora dê um duplo clique no ícone *Sources* e selecione a fonte apropriada a partir do menu *drop-down*, como, por exemplo, a entrada *step*, e arraste-a para a janela sem título. Agora dê um duplo clique no ícone *sinks* e arraste o ícone *graph* para a janela sem título. Para conectar os ícones, pressione o botão esquerdo do mouse quando a seta estiver sobre o símbolo de saída de um ícone e arraste até o símbolo de entrada do ícone que deve ser conectado. Repita isto para todos os ícones até que o diagrama em bloco seja montado por completo.

Para atribuir uma função de transferência ao bloco *transfer Fcn*, dê um duplo clique nele. Isto fará aparecer uma caixa de diálogo na qual você pode usar comandos do MATLAB para numerador e denominador. Clique no numerador e digite [1 1] se for desejado ($s + 1$). Clique no denominador e digite [1 2 3] se for desejado ($s^2 + 2s + 3$). Em seguida, clique no ícone *done*. Dê um duplo clique no ícone *gain* e digite o valor do ganho. Dê um duplo clique no ícone *sum* e configure os sinais + ou - de acordo com o tipo de realimentação desejado (positiva ou negativa). Dê um duplo clique no ícone *graph* e ajuste os parâmetros para o gráfico. O diagrama de simulação está completo na tela. A Figura F.2 mostra o formato dele. Para apagar qualquer bloco ou conexão, selecione o item clicando nele e, em seguida, pressione a tecla DEL.

Para simular o comportamento do sistema, clique em *Simulation* para acionar o seu menu *pull-down*. Selecione *Parameters* e defina os tempos de início e fim da simulação. A partir do menu *Simulation*, selecione *Start*. O SIMULINK produzirá uma janela de gráfico e mostrará a saída do sistema. O arquivo pode ser salvo selecionando-se File e clicando-se em SAVE AS no menu *drop-down*. Digite o nome do arquivo na caixa de diálogo e clique em *Done*.

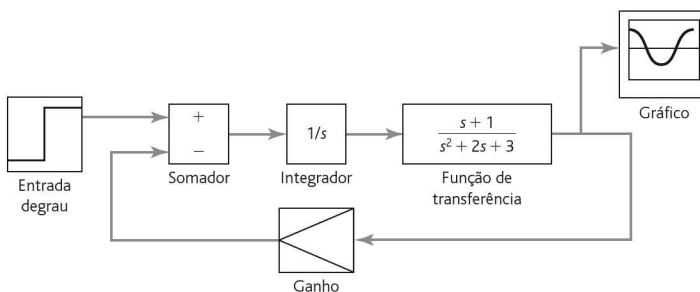


Figura F.2 Exemplo do uso do SIMULINK.

Informações Complementares

Sensores e condicionamento de sinais

- Bolton, W., *Newnes Instrumentation and Measurement*, Newnes 1991, 1996, 2000
Bolton, W., *Measurement and Instrumentation Systems*, Newnes 1996
Boyes, W., *Instrumentation Reference Book*, Newnes 2002
Clayton, G. B. and Winder, S., *Operational Amplifiers*, Newnes 2003
Figliola, R. S. and Beasley, D. E., *Theory and Design for Mechanical Measurements*, Wiley 2000
Fraden, J., *Handbook of Modern Sensors*, American Institute of Physics 1997
Gray, P. R. and Meyer, R. G., *Analysis and Design of Analog Integrated Circuits*, Wiley 1993
Holdsworth, B. and Woods, C., *Digital Logic Design*, Newnes 2002
Johnson, G. W., *LabVIEW Graphical Programming*, McGraw-Hill 1994
Morris, A. S., *Measurement and Instrumentation Principles*, 3rd edition, Newnes 2001
Park, J. and Mackay, S., *Data Acquisition for Instrumentation and Control Systems*, Elsevier 2003
Wells, L. K. and Travis, J., *LabVIEW for Everyone*, Prentice Hall 1997

Atuadores

- Bolton, W., *Mechanical Science*, Blackwell Scientific 1993, 1998, 2006
Bolton, W., *Pneumatic and Hydraulic Systems*, Butterworth-Heinemann 1997
Edwards, J. D., *Electrical Machines and Drives*, Macmillan 1991
Gottlieb, I. M., *Electric Motors and Control Techniques*, TAB Books, McGraw-Hill 1994
Kenjo, T., *Stepping Motors and their Microprocessor Controls*, OUP 1984
Kenjo, T., *Power Electronics for the Microprocessor Age*, OUP 1990
Miller, T. J. E., *Brushless Permanent-magnet and Reluctance Motor Drives*, OUP 1989
Norton, R. L., *Design of Machinery*, McGraw-Hill 1992
Pinches, M. J. and Callear, B. J., *Power Pneumatics*, Prentice Hall 1996
Rohner, P., *Industrial Hydraulic Control*, Wiley 1984, 1986, 1988, 1995
Rohner, P. and Smith, G., *Pneumatic Control for Industrial Automation*, Wiley 1987, 1990

Modelamento de sistemas

- Åström, K. J. and Wittenmark, B., *Adaptive Control*, Addison-Wesley 1995
- Bennett, S., *Real-time Computer Control*, Prentice Hall 1994
- Bolton, W., *Control Engineering*, Longman 1992, 1998
- Bolton, W., *Laplace and z-Transforms*, Longman 1994
- Bolton, W., *Control Systems*, Newnes 2002
- Dorf, R. C. and Bishop, H., *Modern Control Systems*, Addison-Wesley 1998
- Fox, H. and Bolton, W., *Mathematics for Engineers and Technologists*, Butterworth-Heinemann 2002
- Frederick, C., *Modelling and Analysis of Dynamic Systems*, Houghton Mifflin 1993
- Golten, J. and Verwer, A., *Control System Design and Simulation*, McGraw-Hill 1991
- Johnson, J. and Picton, P., *Designing Intelligent Machines, vol. 2. Concepts in Artificial Intelligence*, Butterworth-Heinemann 1995
- Leonard, N. E. and Levine, W. S., *Using MATLAB to Analyze and Design Control Systems*, Addison-Wesley 1995
- Pärt-Elander, E., Sjöberg, A., Melin, B. and Isaksson, P., *The MATLAB Handbook*, Addison-Wesley 1996
- Rzevski, G., Ed., *Designing Intelligent Machines, vol. 1. Perception, Cognition and Execution*, Butterworth-Heinemann 1994
- Shearer, J. L. and Kulakowski, B. T., *Dynamic Modelling and Control of Engineering Systems*, 2nd edition, Prentice Hall 1997

Sistemas microprocessados

- Barnett, R. H., *The 8051 Family of Microcontrollers*, Prentice Hall 1995
- Bates, M., *PIC Microcontrollers*, Newnes 2000, 2004, 2005
- Bolton, W., *Programmable Logic Controllers*, Newnes 1996, 2003, 2006
- Bolton, W., *Microprocessor Systems*, Longman 2000
- Bronson, G., *C for Engineers and Scientists*, West 1993
- Cady, F. M., *Software and Hardware Engineering: Motorola M68HC11*, OUP 1997
- Ibrahim, D., *PIC Basic: Programming and Projects*, Newnes 2001
- Lewis, R. W., *Programming Industrial Control Systems Using IEC 1131-3*, The Institution of Electrical Engineers 1998
- MacKenzie, I. Scott, *The 8051 Microcontroller*, Prentice Hall 1992, 1995, 1999
- Morton, J., *PIC: Your Personal Introductory Course*, Newnes 2001, 2005
- Parr, E. A., *Programmable Controllers*, Newnes 1993, 1999, 2003
- Pont, M. J., *Embedded C*, Addison Wesley 2002
- Predko, M., *Programming and Customizing the PIC Microcontroller*, McGraw-Hill 1998
- Rohner, P., *Automation with Programmable Logic Controllers*, Macmillan 1996
- Schultz, T. W., *C and the 8051, vol. 1 Hardware. Modular Programming and Multitasking*, Prentice Hall 1998; *vol. 2 Building Efficient Applications*, Prentice Hall 1999
- Spasov, P., *Microcontroller Technology: The 68HC11*, Prentice Hall 1992, 1996
- Vahid, F. and Givargis, T., *Embedded System Design*, Wiley 2002
- Van Sickle, T., *Programming Microcontrollers in C*, Newnes 2001
- Yeralan, S. and Ahluwalia, A., *Programming and Interfacing the 8051 Microcontroller*, Addison-Wesley 1993
- Zurrell, K., *C Programming for Embedded Systems*, R & D Books 2000

Respostas

A seguir são apresentadas as respostas dos problemas numéricos e, no caso dos problemas descritivos, as possíveis respostas são apresentadas de forma resumida.

Capítulo 1

- 1.1 (a) Sensor, mercúrio; condicionador de sinal, fina coluna de mercúrio; display, alcance da coluna de mercúrio na escala; (b) sensor, tubo encurvado; condicionador de sinal, mecanismo de movimento; display, ponteiro que se move sobre uma escala.
- 1.2 Veja o texto.
- 1.3 Comparação/controlador, termostato; correção, pode ser um relé; processo, aquecimento; variável, temperatura; medição, um dispositivo sensor de temperatura, talvez um bimetálico.
- 1.4 Veja a Figura P.1.

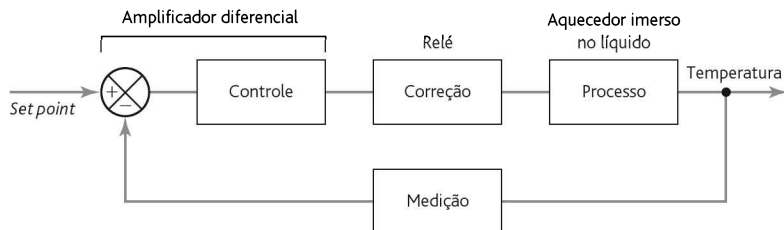


Figura P.1 Problema 1.4.

- 1.5 Veja o texto
- 1.6 Veja o texto

- 1.7 Por exemplo: entrada da água, lavagem, saída da água, entrada da água, aquecimento da água, lavagem, saída da água, entrada da água, lavagem, saída da água.
- 1.8 Tradicional: volumoso, funções limitadas, necessita “dar corda”. Mecatrônica: compacto, muitas funções, não necessita “dar corda”, barato.
- 1.9 Elemento bimetálico: lento, precisão limitada, funções simples, barato. Mecatrônica: rápido, preciso, muitas funções, tornando-se mais barato.

Capítulo 2

- 2.1 Veja o texto para explanação dos termos
- 2.2 $-3,9\%$
- 2.3 $67,5\text{ s}$
- 2.4 $0,73\%$
- 2.5 $0,105\ \Omega$
- 2.6 Incremental, ângulo a partir de uma referência, não absoluto; absoluto, identificação única de um ângulo
- 2.7 162
- 2.8 (a) $\pm 1,2^\circ$, (b) $3,3\text{ mV}$
- 2.9 Veja o texto
- 2.10 $2,8\text{ kPa}$
- 2.11 $19,6\text{ kPa}$
- 2.12 $-0,89\%$
- 2.13 $+1,54^\circ\text{C}$
- 2.14 Sim
- 2.15 $-9,81\text{ N}$, $-19,62\text{ N}$, por exemplo, *strain gauges*
- 2.16 Por exemplo, placa de orifício com célula de pressão diferencial
- 2.17 Por exemplo, célula de pressão diferencial
- 2.18 Por exemplo, sensor de deslocamento LVDT

Capítulo 3

- 3.1 Conforme a Figura 3.2, com $R_2/R_1 = 50$, por exemplo, $R_1 = 1\text{ k}\Omega$, $R_2 = 50\text{ k}\Omega$
- 3.2 $200\text{ k}\Omega$
- 3.3 A Figura 3.5 com duas entradas, por exemplo $V_A = 1\text{ V}$, $V_B = 0$ a 100 mV , $R_A = R_2 = 40\text{ k}\Omega$, $R_B = 1\text{ k}\Omega$
- 3.4 A Figura 3.11 com $R_1 = 1\text{ k}\Omega$ e $R_2 = 2,32\text{ k}\Omega$
- 3.5 $V = K\sqrt{I}$
- 3.6 $100\text{ k}\Omega$
- 3.7 80 db

- 3.8 Fusível para proteger contra alta corrente, resistor de limitação para reduzir correntes, diodo para retificar ca, circuito com diodo Zener para proteção de tensão e polaridade, filtro passa-baixas para remover ruído e interferência, optoisolador para isolar as tensões altas do microprocessador
- 3.9 0,059 V
- 3.10 $5,25 \times 10^{-5} \text{V}$
- 3.11 Conforme dado no problema

Capítulo 4

- 4.1 24,4 mV
- 4.2 9
- 4.3 0,625 V
- 4.4 1, 2, 4, 8
- 4.5 12 μs
- 4.6 Veja o texto
- 4.7 Buffer, conversor digital-analógico, proteção
- 4.8 0,33 V, 0,67 V, 1,33 V, 2,67 V
- 4.9 32 768R
- 4.10 15,35 ms
- 4.11 Fator de 315

Capítulo 5

- 5.1 Por exemplo: (a) bilhete selecionado AND dinheiro correto, dinheiro correto decidido por análise de portas OR dentre as possibilidades, (b) AND com a trava de segurança, lubrificante, resfriador, peça a ser trabalhada, fonte, etc., todos os itens de operação da máquina e ferramenta ou uma combinação de parte deles, c) Figura P.2, d) And
- 5.2 (a) Q, (b) P
- 5.3 AND
- 5.4 1 em A, 0 em B.
- 5.5 Veja a Figura P.3.
- 5.6 Veja a Figura P.4.
- 5.7 De acordo com o texto, Seção 5.3.1, as portas NOR têm acoplamento cruzado

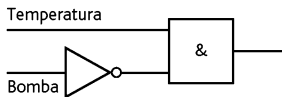


Figura P.2 Problema 5.1(c).

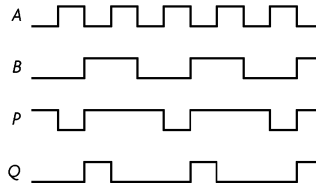


Figura P.3 Problema 5.5.

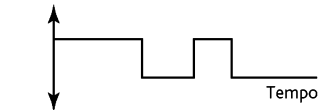
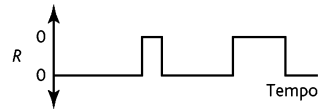
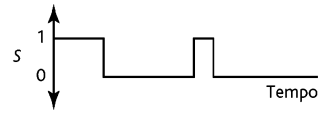


Figura P.4 Problema 5.6.

Capítulo 6

- 6.1 Veja o texto.
- 6.2 Veja a Seção 6.1.
- 6.3 Por exemplo: (a) um registrador, (b) um medidor de bobina móvel, (c) um disco rígido ou CD, (d) um osciloscópio digital ou um disco rígido ou um CD.
- 6.4 O projeto pode ser uma ponte com quatro braços ativos, amplificador operacional diferencial, display de um voltímetro. Os valores dos componentes dependem da espessura escolhida para o aço e do diâmetro da célula de carga. Podemos escolher montar o tanque sobre três células.
- 6.5 O projeto pode ser como na Figura 3.8, com compensação de junção fria por uma ponte (veja a Seção 3.5.2). A linearidade pode ser conseguida por meio de uma escolha adequada dos materiais do termopar.
- 6.6 O projeto pode ser feito com termistores com um elemento de amostragem e retenção seguido de um conversor digital-analógico para cada sensor. Este fornece um sinal digital para transmissão, reduzindo assim os efeitos de possíveis interferências. Podem ser usados optoisoladores para isolar tensões/correntes altas, seguido por um multiplexador que alimenta medidores digitais.
- 6.7 Este sistema se baseia no princípio de Arquimedes: a força para cima é igual ao peso do líquido deslocado pela boia.
- 6.8 No projeto pode ser usado um LVDT ou *strain gauges* com uma ponte de Wheatstone.
- 6.9 Por exemplo: (a) medidor tipo Bourdon, (b) termistores, registrador gráfico galvanométrico, (c) célula de carga com *strain gauge*, ponte de Wheatstone, amplificador diferencial, voltímetro digital, (d) tacogerador, condicionador de sinal para o formato de pulsos, contador.

Capítulo 7

- | | | | |
|-----|---|------|--|
| 7.1 | Veja a Seção 7.3 | 7.6 | Veja a Figura P.5 |
| 7.2 | Veja a Seção 7.3.2 | 7.7 | 0,0057 m ² |
| 7.3 | Veja a Seção 7.4 | 7.8 | 124 mm |
| 7.4 | Veja (a) Figura 7.14, (b) Figura 7.8
(b), (c) Figura 7.10, (d) Figura 7.13 | 7.9 | 1,27 Mpa, $3,9 \times 10^{-5}$ m ³ /s |
| 7.5 | A+, B+, A-, B- | 7.10 | (a) 0,05 m ³ /s, (b) 0,10 m ³ /s |
| | | 7.11 | (a) 0,42 m ³ /s, (b) 0,89 m ³ /s |
| | | 7.12 | 960 mm |

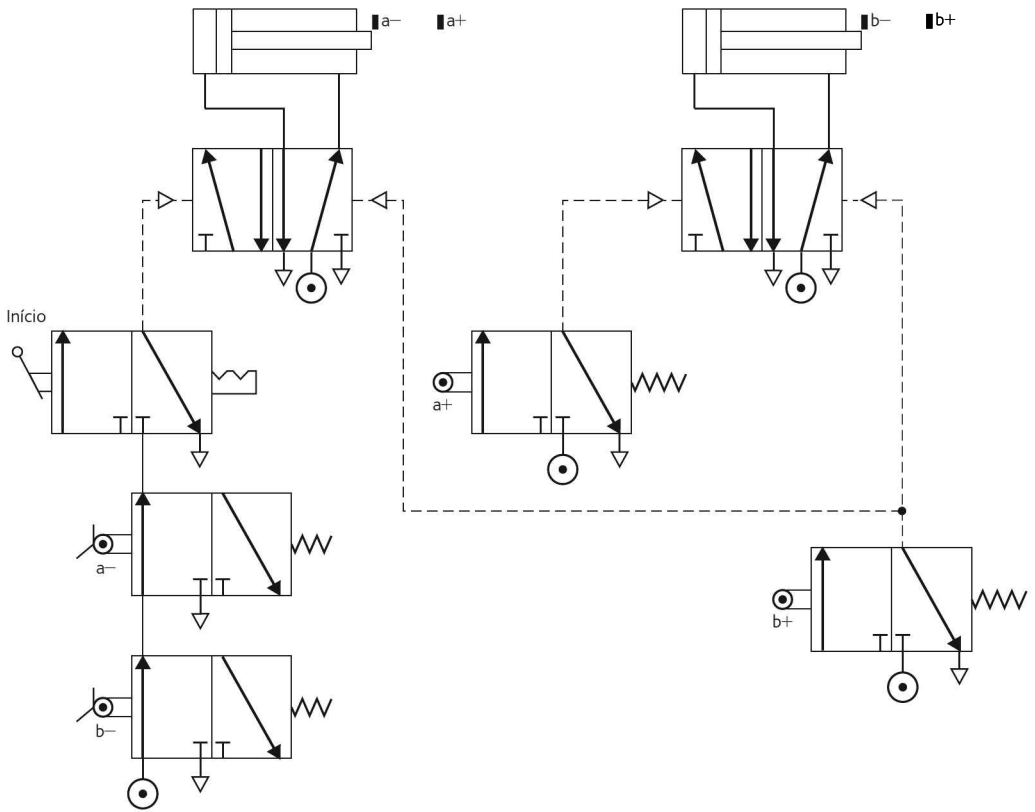


Figura P.5 Problema 7.6.

Capítulo 8

- 8.1 (a) Um sistema de elementos arranjados para transmitir movimento de uma forma para outra. (b) Uma seqüência de junções e clos para prover uma saída controlada em resposta a um movimento de entrada.

- 8.2 Veja a Seção 8.3.1.
- 8.3 (a) 1, (b) 2, (c) 1, (d) 1, (e) 3.
- 8.4 (a) Somente translação, (b) somente translação, (c) somente rotação, (d) somente rotação, (e) translação mais rotação.
- 8.5 Retorno rápido.
- 8.6 Queda repentina no deslocamento seguida por um aumento gradual e retornando novamente ao mesmo ponto de início.
- 8.7 60 mm
- 8.8 Seguidor no formato de coração com distância do eixo de rotação até a parte superior do coração de 40 mm e até a base de 100 mm. Veja a Figura 8.11(b).
- 8.9 Por exemplo: (a) cames em um eixo, (b) mecanismo de retorno rápido, (c) came excêntrica, (d) cremalheira e pinhão, (e) acionamento por correia, (f) engrenagens cônicas.
- 8.10 1/24

Capítulo 9

- 9.1 Ele funciona como um *flip-flop*.
- 9.2 Veja texto e Figura 9.7.
- 9.3 (a) Enrolamento série, (b) enrolamento *shunt*.
- 9.4 (a) Enrolamento *shunt* CC, (b) motor de indução ou síncrono com um inversor, (c) CC, (d) ca.
- 9.5 Veja a Seção 9.5.5.
- 9.6 Veja a Seção 9.7.
- 9.7 480 pulsos/s
- 9.8 9°
- 9.9 (a) 4 kW, (b) 800 W, (c) 31,8 N m
- 9.10 0,65 N
- 9.11 2
- 9.12 3,6 Nm

Capítulo 10

10.1 (a) $m \frac{d^2x}{dt^2} + c \frac{dx}{dt} = F$, (b) $m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + (k_1 + k_2)x = F$

10.2 Como na Figura 10.3(a).

10.3 $c \frac{d\theta_i}{dt} = c \frac{d\theta_o}{dt} + k\theta_o$

10.4 Duas molas de torção em série com um bloco de momento de inércia,

$$T = I \frac{d^2\theta}{dt^2} + k_1(\theta_1 - \theta_2) = m \frac{d^2\theta}{dt^2} + \frac{k_1 k_2}{k_1 + k_2} \theta_1$$

- 10.5 $v = v_R + \frac{1}{RC} \int v_R dt$
- 10.6 $v = \frac{L}{R} \frac{dv_R}{dt} + \frac{1}{CR} \int v_R dt + v_R$
- 10.7 $v = R_1 C \frac{dv_C}{dt} + \left(\frac{R_1}{R_2} + 1 \right) v_C$
- 10.8 $RA_2 \frac{dh_2}{dt} + h_2 \rho g = h_1$
- 10.9 $RC \frac{dT}{dt} + T = T_r$. Capacitor carregado descarregando através de um resistor.
- 10.10 $RC \frac{dT_1}{dt} = Rq - 2T_1 + T_2 + T_3, RC \frac{dT_2}{dt} = T_1 - 2T_2 + T_3$
- 10.11 $pA = m \frac{d^2x}{dt^2} + R \frac{dx}{dt} + \frac{1}{C} x$, R = resistência ao movimento da haste,
 c = capacitância da mola.
- 10.12 $T = \left(\frac{I_1}{n} + n \right) \frac{d^2\theta}{dt^2} + \left(\frac{c_1}{n} + nc_2 \right) \frac{d\theta}{dt} + \left(\frac{k_1}{n} + nk_2 \right) \theta$

Capítulo 11

- 11.1 $\frac{IR}{k_1 k_2} \frac{d\omega}{dt} + \omega = \frac{1}{k_2} v$
- 11.2 $(L_a + L_l) \frac{di_a}{dt} + (R_a + R_l) i_a - k_1 \frac{d\theta}{dt} = 0, I \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} + k_2 i_a = T$
- 11.3 A mesma que para um motor controlado pela armadura

Capítulo 12

- 12.1 $4 \frac{dx}{dt} + x = 6y$
- 12.2 (a) 59,9°C, (b) 71,9°C
- 12.3 (a) $i = \frac{V}{R} (1 - e^{-Rt/L})$, (b) L/R , (c) V/R
- 12.4 (a) Oscilações contínuas, (b) subamortecido, (c) criticamente amortecido, (d) superamortecido.
- 12.5 (a) 4 Hz, (b) 1,25, (c) $i = I \left(\frac{1}{3} e^{-8t} - \frac{4}{3} e^{-2t} + 1 \right)$
- 12.6 (a) 5 Hz, (b) 1,0, (c) $x = (-32 + 6t) e^{-3t} + 6$
- 12.7 (a) 9,5%, (b) 0,020 s
- 12.8 (a) 4 Hz, (b) 0,625, (c) 1,45 Hz, (d) 0,5 s, (e) 8,1%, (f) 1,4 s
- 12.9 (a) 0,59, (b) 0,87

- 12.10 2,4
 12.11 0,09
 12.12 3,93 rad/s, 0,63 Hz

Capítulo 13

- 13.1 (a) $\frac{1}{As + \rho g/R}$, (b) $\frac{1}{ms^2 + cs + k}$, (c) $\frac{1}{LCs^2 + RCs + 1}$
 13.2 (a) 3 s, (b) 0,67 s
 13.3 (a) $1 + e^{-2t}$, (b) $2 + 2e^{-5t}$
 13.4 (a) Sobreamortecido, (b) subamortecido, (c) criticamente amortecido, (d) subamortecido.
 13.5 $t e^{-3t}$
 13.6 $2e^{-4t} - 2e^{-3t}$
 13.7 (a) $\frac{4s}{s^2(s+1)+4}$, (b) $\frac{2(s+2)}{(s+1)(s+2)+2}$,
 (c) $\frac{4}{(s+2)(s+3)+20}$, (d) $\frac{2}{s(s+2)+20}$
 13.8 $5/(s+53)$
 13.9 $5s/(s^2+s+10)$
 13.10 $2/(3s+1)$
 13.11 $-1, -2$
 13.12 (a) Estável, (b) instável, (c) instável, (d) estável, (e) instável.

Capítulo 14

- 14.1 (a) $\frac{5}{\sqrt{\omega^2+4}}, \frac{\omega}{2}$, (b) $\frac{2}{\sqrt{\omega^4+\omega^2}}, \frac{1}{\omega}$,
 (c) $\frac{1}{\sqrt{4\omega^6-3\omega^4+3\omega^2+1}}, \frac{\omega(3-2\omega^2)}{1-3\omega^2}$
 14.2 $0,56 \text{ sen}(5t - 38^\circ)$
 14.3 $1,18 \text{ sen}(2t + 25^\circ)$
 14.4 (a) (i) $\infty, 90^\circ$, (ii) $0,44, 450^\circ$, (iii) $0,12, 26,6^\circ$, (iv) $0, 0^\circ$,
 (b) (i) $1, 0^\circ$, (ii) $0,32, -71,6^\circ$, (iii) $0,16, -80,5^\circ$, (iv) $0, -90^\circ$
 14.5 Veja a Figura P.6.
 14.6 $1/s$, (b) $3,2/(1+s)$, (c) $2,0/(s^2+2\zeta s+1)$, (d) $3,2/[1+s](0,01s^2+0,2\zeta s+1)$

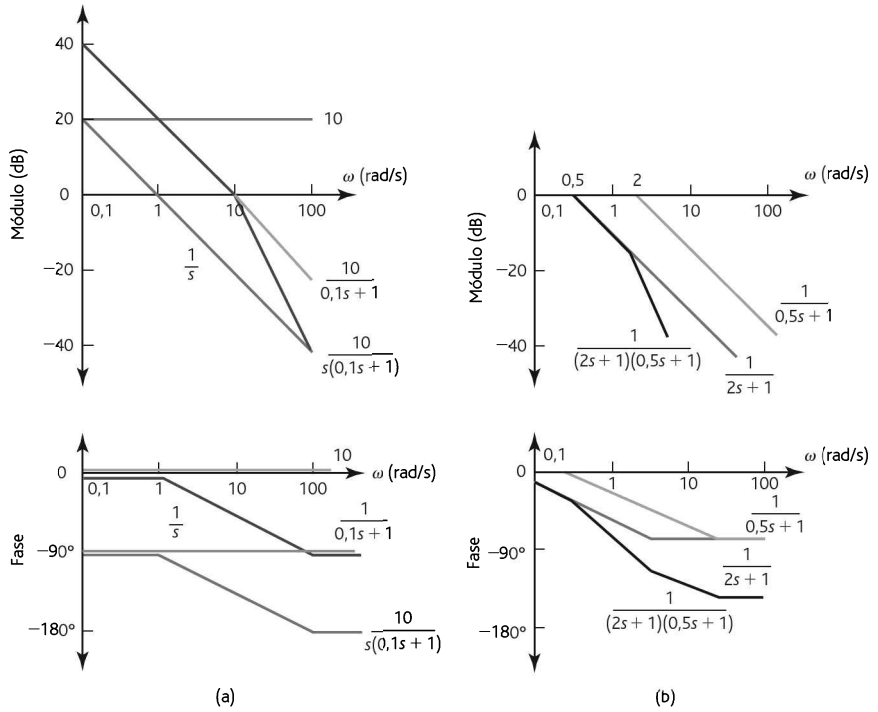


Figura P.6 Problema 14.5.

Capítulo 15

- 15.1 Veja a Seção 15.3.
- 15.2 (a) 8 minutos, (b) 20 minutos
- 15.3 (a) 12 s, (b) 24 s
- 15.4 5
- 15.5 Veja o texto. Em particular, veja o *offset* de P, e a ausência de *offset* em PI e PID.
- 15.6 3, 666 s, 100 s
- 15.7 3, 100 s, 25 s
- 15.8 Veja as Seções (a) 15.12.1, (b) 15.12.2, (c) 15.12.3
- 15.9 1,6
- 15.10 Resposta de primeira ordem com constante de tempo c/K_p

Capítulo 16

- 16.1 Por exemplo, experimente diferenciar diâmetros e tonalidades de vermelho. Você também pode considerar o peso. Os seus resultados devem permitir distinguir claramente entre os tipos de moedas, independente da condição delas.

- 16.2 (a) $1/6$, (b) $1/36$, (c) $1/10$
16.3 0,99
16.4 0,002
16.5 0,625
16.6 0,761
16.7 Por exemplo, se a temperatura da sala $< 20^{\circ}\text{C}$ e temporizador for ligado, então o aquecedor é ligado; se o aquecedor é ligado, então a bomba é ligada; se a bomba é ligada e a temperatura da sala $< 20^{\circ}\text{C}$, então a válvula é ligada; se o temporizador não é ligado, então o aquecedor não é ligado; se a temperatura da sala não for $< 20^{\circ}\text{C}$, então a válvula não é ligada; se o aquecedor não é ligado, então a bomba não é ligada. Você pode ainda refinar esta lógica considerando que há uma restrição em que o aquecedor opera abaixo de 60°C .

Capítulo 17

- 17.1 Veja a Seção 17.2.
17.2 256
17.3 $64\text{ K} \times 8$
17.4 Veja a Seção 17.3.
17.5 Veja a Figura 17.9 e o texto associado.
17.6 (a) E, (b) C, (c) D, (d) B
17.7 256
17.8 (a) 0, (b) 1
17.9 Veja a Seção 17.3.1, item 6.
17.10 Veja a Seção 17.3.2, item 5.
17.11 Nível alto no pino de reset.
17.12 (a) IF A
 THEN
 BEGIN B
 END B
 ELSE
 BEGIN C
 END C
 ENDIF A
(b) WHILE A
 BEGIN B
 END B
ENDWHILE A

Capítulo 18

- 18.1 (a) 89, (b) 99
- 18.2 Nenhum endereço tem que ser especificado, visto que o endereço está implícito no mnemônico.
- 18.3 (a) CLRA, (b) STAA, (c) LDAA, (d) CBA, (e) LDX
- 18.4 (a) LDAA \$20, (b) DECA, (c) CLR \$0020, (d) ADDA \$0020
- 18.5 (a) Armazena o valor do acumulador B no endereço 0035, (b) carrega o acumulador A com o dado F2, (c) zera o *flag carry*, (d) soma 1 ao valor no acumulador A, (e) compara C5 ao valor no acumulador A, (f) zera o endereço 2000, (g) salta para o endereço dado pelo registrador de índice mais 05.
- 18.6 (a)
- | | | | |
|-------|------|--------|------------------------|
| DATA1 | EQU | \$0050 | |
| DATA2 | EQU | \$0060 | |
| DIFF | EQU | \$0070 | |
| | ORG | \$0010 | |
| | LDAA | DATA1 | ; Lê o minuendo |
| | SUBA | DATA2 | ; Subtrai o subtraendo |
| | STAA | DIFF | ; Armazena a diferença |
| | SWI | | ; Fim do programa |
- (b)
- | | | | |
|-------|------|--------|---------------------------------|
| MULT1 | EQU | \$0020 | |
| MULT2 | EQU | \$0021 | |
| PROD | EQU | \$0022 | |
| | ORG | \$0010 | |
| | CLR | PROD | ; Zera o endereço do produto |
| | LDAB | MULT1 | ; Lê o primeiro número |
| SUM | LDAA | MULT2 | ; Lê o multiplicando |
| | ADDA | PROD | ; Soma o multiplicando |
| | STAA | PROD | ; Armazena o resultado |
| | DECB | | ; Decrementa acumulador B |
| | BNE | SUM | ; Desvia se a soma não terminou |
| | WAI | | ; Fim do programa |
- (c)
- | | | | |
|-------|------|--------|--------------------------------------|
| FIRST | EQU | \$0020 | |
| | ORG | \$0000 | |
| | CLRA | | ; Zera o acumulador |
| | LDX | #0 | |
| MORE | STAA | \$20,X | |
| | INX | | ; Incrementa o registrador de índice |
| | INCA | | ; Incrementa o acumulador |
| | CMPA | #\$10 | ; Compara com o número 10 |
| | BNE | MORE | ; Desvia se não for zero |
| | WAI | | ; Fim do programa |

- (d)
- ```

 ORG $0100
 LDX #$2000 ; Seta o ponteiro
LOOP LDA A $00,X ; Carrega o dado
 STA A $50,X ; Armazena o dado
 INX ; Incrementa o registrador de índice
 CPX $3000 ; Compara
 BNE LOOP ; Desvia
 SWI ; Fim do programa

```
- 18.7
- ```

YY     EQU    ??? ; Valor escolhido para proporcionar o atraso
       ; de tempo desejado

SAVEX  EQU    $0100
       ORG    $0010
       STA    SAVEX ; Salva o acumulador A
       LDAA   YY    ; Carrega o acumulador A
LOOP   DECA   ; Decrementa o acumulador A
       BNE    LOOP ; Desvia se não for zero
       LDA    SAVEX ; Recupera o acumulador
       RTS    ; Retorna da sub-rotina

```
- 18.8
- ```

LDA $2000 ; Lê dado de entrada
AND A #$01 ; Mascara todos os bits exceto o bit 0
BEQ $03 ; Se a chave for nível baixo, desvia o JMP
 ; que é 03 linhas de programa
JMP $3000 ; Se a chave for nível alto, não desvia e
 ; executa o JMP.

```
- (Continua)

## Capítulo 19

- 19.1 (a) A variável counter é um inteiro, (b) a variável num está associada ao valor 10, (c) a palavra name é mostrada, (d) o display mostra Number 12, (e) inclui o arquivo stdio.h
- 19.2 (a) Chama a biblioteca necessária para a função printf( ), (b) indica o início e o fim de um grupo de declarações, (c) começa uma nova linha, (d) problem 3
- 19.3 O número é 12
- 19.4 # include <stdio.h>
- ```

int main(void);
{
    int len, largura;
    printf("Enter comprimento: ");
    scanf("%d", &len);
    printf("Enter largura: ");
    scanf("%d", &largura);
    printf("Area is %d", len * largura);
    return 0;
}

```

- 19.5 Similar ao programa dado na Seção 19.3, item 4.
- 19.6 Divide o primeiro número pelo segundo, exceto se o segundo for 0

Capítulo 20

- 20.1 Veja a Seção 20.3.
- 20.2 Veja a Seção 20.3. Uma interface paralela tem o mesmo número de linhas de entrada/saída que o microprocessador. Uma interface serial tem apenas uma única linha de entrada/saída.
- 20.3 Veja a Seção 20.2.
- 20.4 Veja a Seção 20.4.
- 20.5 Veja a Seção 20.4 e a Figura 20.10.
- 20.6 Veja a Seção 20.4.1.
- 20.7 Veja a Seção 20.3.3. A consulta envolve a interrogação de todos os periféricos em intervalos regulares, mesmo que algum deles não esteja ativado. Esta operação gasta tempo. Uma interrupção é apenas iniciada quando um periférico solicita e é uma operação mais eficiente.
- 20.8 CRA 00110100, CRB 00101111
- 20.9 Conforme o programa em 18.4.2, com LDAA #05 substituída por LDAA #34 e LDAA #34 substituída por LDAA #2F.
- 20.10 Como o programa na Seção 20.4.2, seguido por
READ LDAA \$2000 ; Lê a porta A
Talvez após algum programa de atraso possa ser inserido
BRA READ

Capítulo 21

- 21.1 (a) AND, (b) OR
- 21.2 (a) Figura 21.9(b), (b) Figura 21.10(b), (c) um circuito *latch*, Figura 21.16, sendo a chave da partida na Entrada 1 e a de parada na Entrada 2
- 21.3 0 LD X400, 1 LD Y430, 2 ORB, 3 ANI X401, 4 OUT Y430
- 21.4 0 LD X400, 1 OR Y430, 3 OUT Y430, 4 OUT T450, 5 K 50; temporizador com atraso para ligar
- 21.5 0 LD X400, 1 OR Y430, 2 ANI M100, 3 OUT Y430, 4 LD X401, 5 OUT M100; *latch* de reset
- 21.6 Como na Figura 21.28, com o Temporizador 1 tendo $K = 1$ para 1 s e o Temporizador 2 tendo $K = 20$ para 20 s.
- 21.7 Figura P.7.
- 21.8 Figura P.8.

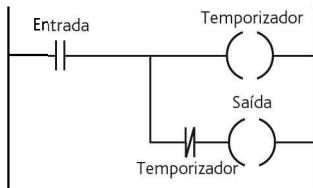


Figura P.7 Problema 21.7.

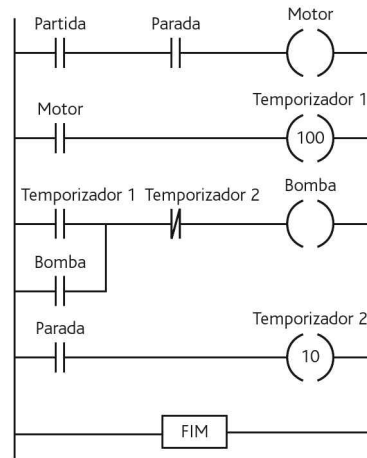


Figura P.8 Problema 21.8.

- 21.9 Figura P.9.
- 21.10 Figura P.10.
- 21.11 Uma saída pode ligar, como antes, mas desligar quando a próxima entrada ocorrer.
- 21.12 Veja a Seção 21.10.
- 21.13 Dois circuitos *latch*, como na Figura P.11.
- 21.14 Figura P.12.

Capítulo 22

- 22.1 Veja a Seção 22.2.
- 22.2 Veja a Seção 22.3.
- 22.3 Barramento
- 22.4 *Broadband*
- 22.5 Veja a Seção 22.5.1.
- 22.6 Veja a Seção 22.4.
- 22.7 Veja a Seção 22.3.1.
- 22.8 NRFD para PD0, DAV para STRA e IRQ, NDAC para STRB, dado para a Porta C
- 22.9 A conversão do nível de sinal de TTL para RS-232C
- 22.10 Veja a Seção 22.7.1.

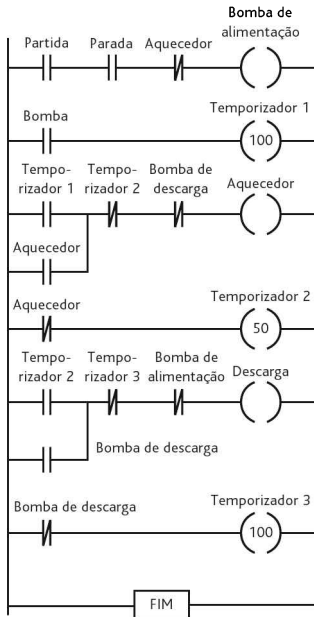


Figura P.9 Problema 21.9.

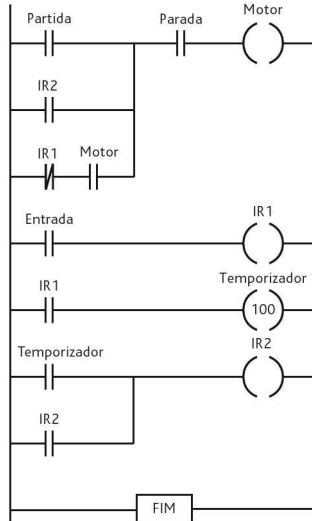


Figura P.10 Problema 21.10.

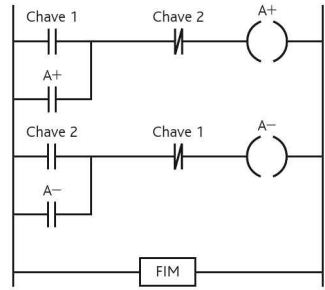


Figura P.11 Problema 21.13.

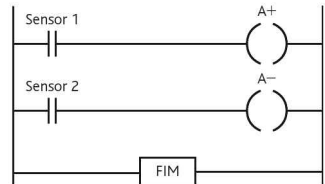


Figura P.12 Problema 21.14.

Capítulo 23

- 23.1 Veja a Seção 23.1.
- 23.2 Veja a Seção 23.2
- 23.3 Veja a Seção 23.2
- 23.4 Veja a Seção 23.5.3 para verificar programação e *checksum* e a Seção 23.2 para o temporizador *watchdog*.
- 23.5 Veja a Seção 23.5.3.

Capítulo 24

- 24.1 As soluções possíveis podem ser: (a) termopar, compensação de junção fria, amplificador, ADC, PIA, microprocessador, DAC, unidade de tiristor para controlar o elemento de aquecimento do forno, (b) sensores de feixe de luz, CLP, desviadores para rampas de distribuição operados por solenoides, (c) controle em malha fechada, em cada direção de movimento, tendo um motor CC como atuador para mover uma pena, microprocessador como comparador e controle, e realimentação a partir de um encoder óptico.

Atividades de pesquisa

A seguir apresentamos indicações resumidas do tipo de informação que a resposta pode conter:

- 24.2 Um sistema de frenagem com ABS típico tem sensores, do tipo indutivo, que geram uma informação de velocidade para cada roda do carro. Condicionadores de sinais convertem as informações dos sensores em pulsos de 5 V. Um microcontrolador com um programa calcula a velocidade das rodas e a taxa de desaceleração durante a frenagem, de modo que, quando o valor limite é excedido, o microcontrolador gera uma saída para válvulas solenoides da unidade moduladora hidráulica para evitar um aumento na força de frenagem ou, se necessário, para reduzi-la.
- 24.3 O motor do carro de impressão move a cabeça de impressão de lado enquanto a cabeça de impressão imprime os caracteres. Após a impressão de uma linha, o motor de movimentação do papel faz este avançar. A cabeça de impressão consiste de “agulhas” acionadas por solenoides, tipicamente uma coluna de 9, que impactam a fita tintada sobre o papel. Um microcontrolador pode ser usado para controlar as saídas. Para mais detalhes, veja *Microcontroller Technology: The 68HC11* por P. Spasov (Prentice Hall, 1992, 1996).
- 24.4 O barramento CAN opera com sinais que têm um bit de início, seguido pelo nome que indica o destino da mensagem e sua prioridade, seguido por bits de controle, seguido pelos dados enviados, seguido por bits CRC, seguido por uma confirmação de bits de recepção e finalizado por bits de fim.

Atividades de projeto

As seguintes indicações resumem as soluções possíveis:

- 24.5 Um termômetro digital que usa um microprocessador pode ter um sensor de temperatura como o LM35, um ADC, uma chip de ROM como o MCM6830 da Motorola ou o 8355 da Intel, um chip de RAM como o MCM6810 ou o 8156 da Intel, um microprocessador como o M6800 da Motorola ou o 8085A da Intel e um driver com display de LEDs. Com um microcontrolador como o MC68HC11 da Motorola ou o 8051 da Intel, pode ser necessário apenas mais um sensor, com condicionador de sinal e um driver para display de LED.
- 24.6 Um ohmímetro digital pode envolver um multivibrador monoestável que fornece um pulso com uma duração de $0,7RC$. Uma gama de diferentes valores de capacitores fixos pode ser usada para prover diferentes escalas de resistência. O intervalo de tempo pode ser determinado usando-se um microcontrolador ou um microprocessador mais memória e, em seguida, enviado diretamente para um driver adequado para display de LED.
- 24.7 Este projeto pode envolver um sensor de pressão, como, por exemplo, o transdutor MPX2100AP da Motorola, que é construído com semicondutor, um condicionador de sinal para converter o pequeno sinal diferencial do sensor para o nível apropriado, como, por exemplo, um amplificador de instrumentação que usa amplificadores operacionais, um microcontrolador, como, por exemplo, o MC68HC11, um driver de LCD, como, por exemplo, o MC145453, e um display LCD de quatro dígitos.

- 24.8 Este projeto pode ser implementado usando-se uma placa M68HC11EVM com uma saída PWM para o motor. Um encoder óptico pode ser usado na realimentação.
- 24.9 A configuração pode ter, para cada caixa a ser carregada, uma válvula solenoide que opera um cilindro pneumático para mover uma aleta e permitir que a caixa desça uma rampa. A caixa permanece na rampa fechada por uma aleta. A presença dela é detectada por um sensor que indica que a próxima caixa pode ser liberada para a rampa. Este procedimento continua até que as quatro caixas estejam na rampa. A aleta no final da rampa pode ser ativada por outra válvula solenoide que aciona um cilindro que permite que as caixas passem para a correia transportadora. O deslocamento das caixas na correia pode ser indicado por um sensor montado na extremidade da rampa. Isto pode então permitir que todo o processo seja repetido.

Apêndice A

- A.1 (a) $2/s^2$, (b) $2/(s^2 + 4)$, (c) e^{-2s} , (d) $sX(s) - 2$, (e) $3s^2X(s)$, (f) $1/[s(s + 1)]$
- A.2 (a) t , (b) $5 \cos 3t$, (c) $1 + 2e^t$, (d) e^{-3t}
- A.3 5

Apêndice B

- B.1 255
- B.2 (a) 11, (b) 529
- B.3 (a) 1A7, (b) 211
- B.4 (a) 781, (b) 157
- B.5 (a) 1010 0110, (b) 1101 1101
- B.6 (a) 0, (b) 1
- B.7 (a) 8, (b) 12

Apêndice C

- C.1 (a) $A \cdot (B + C)$, (b) $(A + B) \cdot (C + D)$, (c) $\bar{A} + B$, (d) $\bar{A} \cdot B$
- C.2 (a) $Q = (A \cdot B + C \cdot D) \cdot E$, (b) $Q = (A \cdot B + B) \cdot C$

C.3

A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

C.4 (a) $Q = C \cdot (A + D)$, (b) $Q = A \cdot B$, (c) $Q = A \cdot \bar{B} \cdot C + C \cdot D$

C.5 Conforme dado no problema.

C.6 (a) $Q = A + B$, (b) $Q = C + \bar{A} \cdot C$ C.7 (a) $Q = \bar{A} \cdot \bar{B} + \bar{A} \cdot \bar{C}$, (b) $Q = A \cdot B \cdot D + A \cdot B \cdot \bar{C} + \bar{C} \cdot D$ C.8 Portas AND de quatro entradas com duas portas NOT se a combinação correta for 1, 1, 0, 0: $Q = A \cdot B \cdot \bar{C} \cdot \bar{D}$

Índice

A

- abordagem sequencial para processos de projeto 14–15
- absorção de corrente em CLPs 491–492
- acesso a rede com passagem de slot 523–524
- acesso a rede com passagem de token 523–524
- acesso direto à memória (DMA) 114–116
- acesso indireto em C 456–457
- acesso múltiplo com detecção de portadora e de colisão (CSMA/CD) 523–524
- acesso múltiplo com detecção de portadora e prevenção de colisão (CSMA/CA) 537–538
- acionador de disco rígido 579–582
- acumulador 175–176
- adaptador de interface de comunicação assíncrona (ACIA) 480–482
- adaptadores de interface periféricas (PIAs) 475–482
 - inicialização 478–480
 - sinais de interrupção 478–480
- ADC em forma de rampa 110–111
- ADC flash 110–112
- álgebra Booleana 602–611
 - leis da 602–604
 - mapas de Karnaugh 606–611
 - tabelas-verdade 604–607
 - teoremas de De Morgan 603–605
- algoritmo 412–413
- aliasing 103–104
- Allen-Bradley data highway 529–530
- amortecedor 249–251
 - equações 252–253
- amortecedor rotativo 251–252
 - equações 252–253
- amplificador de diferença 82–87
- amplificador de instrumentação 84–85
- amplificador de integrador 81–83
- amplificador inversor 79–80
- amplificador logarítmico 86–87
- amplificador não inversor 80–81
- amplificador operacional 78–89
- amplificador somador 80–82
- amplificadores de amostragem e retenção 112–113
- apagamento de displays 484–486
- aprendizado 377
- apresentação de dados
 - displays para 144–146, 156–162
 - elementos de 145–151
 - impressoras 149–151
 - medidores analógicos e digitais 145–147
 - osciloscópio de raios catódicos 146–148
 - registradores gráficos analógicos 146–147
 - unidade de display visual 147–150
- gravação magnética 150–156
 - discos 154–156
 - gravação óptica 155–157
 - veja também* sistemas de medida
- aproximação de Tustin 117–118
- aproximações sucessivas em ADC 107–108
- aquisição de dados (DAQ) 113–116
 - detecção de erro 115–116
 - displays 161–165
 - data loggers 162–165
 - placas *plug-in* 161–164
 - precisão 114–116
- armadura em motores CC 229–230
- arquitetura 6800 da Motorola 385–387, 391–393
- arquitetura de Harvard 402–406
- Arquitetura de Rede de Sistemas (SNA) 527–528
- arquitetura do 8051 da Intel 386–392
- arquivos de biblioteca padrão em C 445–446
- arquivos de cabeçalho em C 457–459
- arranjo de Bayer 573–574
- arranjo multidimensional em C 454–456

- arranjos em C 454–456
 e ponteiros 456–458
 árvores de probabilidades 374–375
 atenuação 91–92
 atrasos com motores de passo 435–437
 atrasos em controladores em malha fechada 348–349
 atribuições em C 447–448
 atuador de diafragma 190–191
 atuadores rotativos 195–196
 atuadores semi-rotativos 195–196
 atualização contínua de entrada/saída em CLPs 493–494
 autorregulação 372
- B**
- balança de banheiro 563–567
 banda morta 351–352
 banda morta de transdutores 37–38
 barramento de Arquitetura de Micro Canal (MCA) 536–537
 barramento de Arquitetura Padrão da Indústria Estendido (EISA) 536–537
 barramento de computador AT 536–537
 barramento de computador XT 534–536
 barramento de controle 381–382
 barramento de dados 380–382
 barramento de endereço 381–382
 Barramento de Instrumentação da Hewlett Packard 533–534
 barramento externo 519–520
 barramento LAN Ethernet 523–524
 barramento S-bus 536–537
 barramento TURBOcanal 536–537
 barramento VME 536–537
 barramento VXI 536–538
 barramentos
 em sistemas microprocessados 380–382
 externo 519–520
 base em números de ponto flutuante 599–600
 BASIC 417–418
 bit de paridade 115–116
 bit mais significativo (MSB) 101–102, 595–596
 bit menos significativo (LSB) 101–102, 595–596
Bit User Fast Friendly Aid to Logical Operations (Buffalo) 549–550
 bits de verificação de redundância cíclica 525
 blocos construtivos de sistemas elétricos 256–261
 construção 257–260
 e analogias mecânicas 258–260
 equações 257–258
 blocos construtivos de sistemas fluidicos 260–269
 construção 264–269
 equações 265–266
 blocos construtivos de sistemas hidráulicos 260–263
 equações 265–266
 blocos construtivos de sistemas mecânicos 249–256
 construção 252–256
 e analogias elétricas 258–260
 equações 252–253
 sistemas rotacionais 251–253
 blocos construtivos de sistemas pneumáticos 263–265
 equações 265–266
 blocos construtivos de sistemas térmicos 268–271
 construção 269–271
 equações 269–270
Bluetooth 537–538
 bobinas de campo em motores CC 229–230
 boias 61–62
 bomba de engrenagem 175–177
 bomba de palhetas 175–177
 bomba de pistão 175–176
 bomba de pistão axial 175–177
 bomba de pistão radial 175–177
 buffer de entrada 389–392
 buffers 467–470
buffers tristate 468–469
- C**
- C 417–418
 arranjos 454–456
 e ponteiros 456–458
 características importantes
 arquivos de biblioteca padrão 445–446
 comentários 446–447
 declarações 447–448
 função main 446–447
 funções 444–445
 operações binárias 448–449
 operadores aritméticos 447–448
 operadores lógicos 448–449
 operadores relacionais 447–448
 pré-processador 445–447
 retorno 445–446
 sequências de escape 449
 strings 448–449
 teclados 444–445
 variáveis 446–448
 desenvolvimento de programas 457–459
 arquivos de cabeçalho 457–459
 desvios 450–451
 declaração *If* 450–451
 declarações *If/else* 451
 estrutura de programas 443–450
 funções de biblioteca 620–623
 leitura de um canal ADC 459–461
 ligando e desligando um motor 458–461
 loops 451–455
 declaração *for* 451–452
 declaração *switch* 452–455
 declaração *while* 452
 ponteiros 455–458
 aritmética com 456–457
 e arranjos 456–458
 programa exemplo 449–450
 vantagens do 443–444
 cadeias cinemáticas 202–206
 de quatro barras 203–205
 mecanismo cursor-manivela 204–206
 camada parcial em rolamentos radiais 214–215
 camadas no sistema OSI 525–527
 câmera digital 571–577
 câmera digital e foco automático 29–31
 cames 205–207
 e seguidores 205–206
 caminho de realimentação 318–319

- caminho direto 318–319
 campo de endereço 421–422, 425–426
 campo na linguagem assembly 424–425
 cancelas em estacionamento de carros 570–573
 capacitância hidráulica 260–261
 equações 265–266
 capacitância pneumática 263
 equações 265–266
 capacitância térmica 268–269
 equações 269–270
 capacitor 256–258
 equações 257–258
 características dinâmicas de transdutores 38–41
 características estáticas de transdutores 38–41
 catraca e lingueta 211–212
 célula de bit 152–153
 célula de carga com *strain gauge* 55–57
 células de carga 55–57
 calibração de 169–170
 em balanças de banheiro 563–564
 em sistemas de medida 164–166
 chave de proximidade indutiva 46–47
 chave fim de curso 49–50, 70–71
 em cancela de estacionamento para carro 570–571
 chave operada por came 27–28, 610–611
 chave temporizada 610–559
 chaves de estado sólido 222–229
 diodos 222–224
 MOSFETs 228–229
 tiristores e triacs 223–226
 transistores bipolares 225–229
 chaves de proximidade 49–51
 chaves mecânicas 70–71, 220–223
 estado sólido 222–229
 relés 221–223
 chaves, defeitos em 544–545
 chaves, entrada de dados com 70–74
 ciclo de busca-execução 384–385
 ciclo de programa em CLPs 493–494
 ciclo de trabalho na modulação por pulsos 98–99
 cilindro de duplo estágio 185–186
 cilindro hidráulico 184–185
 cilindro pneumático 184–185
 cilindros 184–188
 sequenciamento 186–188
 circuito de chave térmica 87–88
 circuito de controle de fase 225–226
 circuito *snubber* 225–226
 circuitos H 234, 241–243
 circuitos integrados 126–127
 circuitos latch em CLPs 502–503
 circuitos PID 357–359
 circuitos RLC 299–300
 clear e reset mestres em CLPs 408–409
 codificação de comprimento limitado (RLL) 153–156
 codificação de fase (PE) 153–154
 codificação NRZ 152–153
 codificador 128–130
 código de máquina 417–418
 código de operação 423–424
 código Gray 48–50, 599–601
 código mnemônico 417–418
 código padrão americano para troca de informações (ASCII) 148–149
 coeficiente Hall 51–52
 comparador 86–88
 comparador digital 127–129
 compensação em funções de transferência 322–324
 complemento de dois na matemática binária 597–598
 complemento de um em matemática binária 597–598
 compressores 177–179
 COP 398–399, 541–542
 comunicações digitais 519–520
 comutador em motores CC 230–231
 condicionador de sinal 16–17
 condicionamento de sinal 77–79
 condições de estado estacionário e resposta transiente 320–321
 em sistemas de segunda ordem 299–300
 condições de Grashof 204–205
 conexões 202–203
 conexões de anodo comum e catodo comum para LEDs 157–159
 conhecimento 372–373
 conjuntos de instruções 417–425
 8051 da Intel 615–618
 aritméticas 418–420
 controle de programa 420–422
 endereçamento 421–424
 lógicas 419–421
 M68HC11 da Motorola 613–615
 movimentação de dados 423–425
 PIC16Cxx 618–619
 transferência/movimentação de dados 418–419
 constante de tempo de transdutores 39–40
 constante de tempo derivativa 354–355
 constante de tempo em sistemas de primeira ordem 293–296
 constante de tempo integral 357–358
 consulta 388–389, 470–471
 contadores em CLPs 506–510
 controlador lógico programável (CLP) 28–30, 489–490
 arquitetura 490–491
 cancelas em estacionamento para carro 570–571
 circuitos latch 502–503
 como monitor de sistemas 552–553
 contadores 506–510
 detecção de falhas em 540–541, 551–553
 diagramas ladder 495–496
 em chave temporizada 556–557
 entrada/saída analógica 513–516
 estrutura 489–494
 entrada/saída 490–493
 formas de 492–493
 programas de entrada 492–493
 lista de instruções 499–502
 manuseio de dados 511–514
 comparação de dados 512–514
 conversões de código 513–514
 movimentação de dados 512–513
 operações aritméticas 513–514
 processamento de entrada/saída 493–495
 atualização contínua 493–494

- cópia em massa de entrada/saída 493–495
- endereços 494–495
- programação ladder 494–500
- funções lógicas 497–500
- registradores de deslocamento 509–511
- relés internos 503–504
- relés mestre 510–512
- saltos 511–512
- sequenciamento 504–506
- temporizadores 505–507
- controlador PID 421–422
- controlador proporcional eletrônico 352–353
- controladores de três termos (PID) 350–351, 357–359
- controladores digitais 359–362
- modos de controle 360–361
- controladores em malha aberta 347–348
- controladores em malha fechada 347–348
- atrasos 348–349
- autorregulação em 372
- controladores digitais 359–362
- controle adaptativo 364–368
- controle de velocidade 364–365
- desempenho 361–363
- erro de estado estacionário em 348–350
- modo derivativo 349–351, 353–357
- modo integral 350–351, 355–359
- modo on/off 349–352
- modo PID 350–351, 357–359
- modo proporcional 349–354
- resposta do sistema 353–354
- modos de controle 349–351
- para acionadores de disco rígido 579–581
- sintonia 362–365
- controladores PIC *veja Peripheral Interface Controller*
- controladores sequenciais 26–29
- controle com ganho escalonado 366–367
- controle computacional centralizado 519–520
- controle da velocidade da temperatura via CLP 514–516
- controle de acesso a redes 523–524
- controle de enlace lógico 526–527
- controle de interrupção 471–472
- controle de motor de passo 561–562
- controle de realimentação 19–20
- controle de robô 568
- controle de velocidade 364–365
- controle de velocidade do motor via CLPs 513–515
- controle digital 120–121
- controle digital direto 347–348
- controle e autossintonia 366–368
- controle em cascata de cilindros 187–188
- controle em cascata em CLPs 504–505
- conversão analógico-digital (ADC) 25–26, 107–112
- conversão de códigos em CLPs 513–514
- conversor analógico-digital (ADC) 101–105
- em controladores digitais 359–360
- taxa de amostragem 361–362
- teorema da amostragem 103–105
- conversor de código 129–135
- conversor de dupla rampa 110–112
- conversor digital-analógico (DAC) 25–26, 104–108
- interfaceamento 486–487
- conversor em motores CA 239–240
- cópia em massa de entrada/saída em CLPs 493–495
- coroa dentada 207–208
- correias, tipos 212–213
- critério de Nyquist 103–104
- curva de reação de processo 363–364
- D**
- 8051 da Intel 399–404
- acesso externo 400–401
- ALE 400–401
- conjunto de instruções 615–618
- entrada/saída serial 402–403
- interrupção 402–404
- portas de entrada/saída 399–401
- programa 428–429
- PSEN 400–401
- RESET 402–403
- temporização 402–403
- XTAL 402–403
- dado memorizado 138–139
- dados de protocolo 524–525
- data loggers* 162–165
- Decimal Codificado em Binário (BCD) 129–131
- declarações em C 444–445
- decodificação de dados 129–130
- decodificador 129–130
- decodificador de 3 para 8 linhas 129–130, 132–133
- decodificador de endereço 388–389
- decremento em sistemas dinâmicos 302–304
- dente de engrenagem axial 207–208
- dente de engrenagem helicoidal 207–208
- derivadas em transformadas de Laplace 591–592
- derreferenciação de ponteiros em C 456–457
- desvios e listas de instruções em CLPs 500–502
- desvios em C 450–451
- declaração *if* 450–451
- declaração *if/else* 451
- deteção de erro, método de paridade para 115–116
- deteção de falha com analisador de assinatura 547
- deteção de falha com multímetro 546
- deteção de falha com osciloscópio 546
- deteção de falha com ponta de prova lógica 546–547
- deteção de falha com pulsador lógico 547
- deteção de falha por inspeção visual 546
- deteção de falhas por clip lógico 547
- deteção de falhas por comparador lógico 547
- deteção de falhas por rastreador de corrente 546–547

- detectores de temperatura baseado em resistência (RDTs) 63–65
- diagrama de corpo livre 252–255
- diagramas ladder em CLPs 495–496
- diferença em matemática binária 597–598
- diodo de proteção 227–228
- diodo Zener 90–91
- diodos 221–225
- diodos emissores de luz (LEDs) 156–159
- diodos *flyback* ou roda livre 221–222
- discos magnéticos 154–156
- discos rígidos 155–156
- display de sete segmentos, decodificação de 483–487
- displays
- apagamento 484–486
 - efeito de carga 144–146
 - na aquisição de dados 161–165
 - data loggers* 162–165
 - placas plug-in 161–164
 - para apresentação de dados 144–146, 156–162
 - para sistemas de medição 144–145, 164–168
 - células de carga 164–166
 - medição de temperatura 167–169
 - posição angular 166–168
 - sistema de alarme de temperatura 165–167
- displays alfanuméricos 156–157
- displays de cristal líquido (LCDs) 156–162
- displays de lâmpada néon 156–157
- displays de lâmpadas incandescentes 156–157
- displays de matriz de pontos 156–160
- dispositivo de acoplamento de carga (CCD) 68–70, 571–573
- dispositivos periféricos 387–388
- domínio do tempo nas transformadas de Laplace 585–586
- domínio s em transformadas de Laplace 585–586
 - propriedade do deslocamento 590–591
- drivers para placas DAQ 161–162
- E**
- EEPROM (PROM apagável eletricamente) 387–388**
- efeito de carga em sistemas mecânicos 202–203
- efeito de carga, em displays 144–146
- efeito Hall 50–51
- eixos de engrenagens paralelas 207–208
- elemento de comparação de um sistema de malha fechada 21–22
- elemento de correção de sistemas de malha fechada 22–23
- elemento de medida de sistemas em malha fechada sistemas 22–23
- elemento de processo de sistemas em malha fechada 22–23
- elementos de controle de sistemas de malha fechada 21–23
- elementos PZT 574–575
- eliminação do efeito do ruído 71–73
- em robôs manipuladores 569
- encoders absoluto de 3 bits 48–49
- encoders absolutos 46–47
- encoders incrementais 46–47, 52–53
- encoders ópticos 46–50
- endereçamento de página zero 422–423
- endereçamento de registro 422–423
- endereçamento direto 422–424
- endereçamento estendido 422–424
- endereçamento imediata 421–424
- endereçamento implícito 422–423
- endereçamento indexado 422–424
- endereçamento indireto 422–423
- endereçamento inerente 422–424
- endereçamento relativo 422–423
- endereço absoluto 422–423
- engrenagem dentada helicoidal dupla 207–208
- engrenagens 207–211
- engrenagens cônicas 207–208
- engrenagens dentadas 207–208
- engrenagens helicoidais 207–208
- entrada do cristal em PICs 408–409
- entrada em degrau
 - em sistemas de primeira ordem 313–315
 - em sistemas de segunda ordem 316–317
 - em sistemas dinâmicos 288–290, 302–303
- entrada impulso em sistemas dinâmicos 288–290
- entrada senoidal
 - em sistemas dinâmicos 289–290
 - fasores em 237–330
 - equações 328–330
 - resposta de frequência para 236–237
- entrada/saída analógica
 - em CLPs 513–516
 - em PICs 406–407
- entrada/saída full-handshake 470–471
- entrada/saída isolada 464–465
- entrada/saída mapeada na memória 465–466
- entrada/saída serial em PICs 408–409
- EPROM (ROM apagável e programável) 386–389
- equação Booleana 121–122
- equações auxiliares 297–298
- equações diferenciais 253–254, 286–288
- erro de estado estacionário em controladores de malha fechada 348–350
- erro de histerese de transdutores 36–37
- erro de linearidade 110–112
- erro de não linearidade de transdutores 36–37
- erro de quantização 103–104
- erro de transdutores 35–36
- erros de amplificadores 88–89
- estabilidade de transdutores 37–38
- estágio conceitual do processo de projeto 13–14
- estrutura de um programa em C 443–450
- expoente em números de ponto flutuante 599–600

F

faixa de operação de transdutores 35–36
 falha em circuitos abertos 545
 falha no chip 545
 falhas de curto-circuito 545
 falhas de hardware 543–545
 falhas em software 546
 falhas por interferência 546
 fasores em entrada senoidal 237–330
 equações 328–330
 fator de amortecimento 297–298
 filme de fluoreto de polivinilideno (PVDF) 60–61
 filtragem 91–93
 filtro anti-aliasing 104–105
 filtro ativo 91–93
 filtro passa-altas 91–92
 filtro passa-baixas 91–92
 filtro passa-faixa 91–92
 filtro passivo 91–93
 filtro rejeita faixa 91–92
 firmware 387–388
 flip-flop 133–137
 flip-flop D 136–139
 flip-flop JK 136–138
 flip-flop SR 133–137
 flip-flop SR com portas lógicas 136–138
 fluxograma 412–414
 foco automático em câmeras digitais 29–31
 fole 266–267
 fontes em CLPs 491–492
 formatação de discos 155–156
 formato JPEG 573–574
 FORTRAN 417–418
 fotodiodos 68–70
 fotoresistor 68–70
 fototransistores 68–70
 fototransistores Darlington 68–70
 frações parciais entra transformadas de Laplace 592–594
 frequência de canto em gráficos de Bode 333–335
 frequência de corte 91–92
 função de impulso unitário em transformadas de Laplace 587–588

função de pertinência 375–377
 função de transferência de frequência 329–330
 função degrau unitário em transformadas de Laplace 586–588
 função delta de Dirac em transformadas de Laplace 587–588
 função impulso em transformadas de Laplace 587–589
 função lógica AND em CLPs 497–498
 função lógica EX-OR em CLPs 498
 função lógica NAND em CLPs 498
 função lógica NOR em CLPs 498
 função lógica OR em CLPs 498
 função resposta de frequência 329–330
 funções de Heaviside em transformadas de Laplace 586–587
 funções de transferência 310–314
 compensação 322–324
 em controladores digitais 360–361
 em sistemas de primeira ordem 313–316
 entrada em degrau 313–315
 exemplos de 313–316
 em sistemas de segunda ordem 315–318
 entrada em degrau 316–317
 exemplos de 317–318
 gráficos de Bode para 332–337, 341–342
 plano s 322–323
 polos 321–322
 sistemas em série 317–319
 exemplos de 318–319
 transformadas de Laplace 311–314
 funções de transferência de um braço de robô 317–318
 funções lógicas em CLPs 497–500
 e listas de instruções em CLPs 500–502
 funções periódicas em transformadas de Laplace 590–592
 fuso esfera 210–211

G

ganho de corrente em transistores bipolares 226–227
 ganho de estado estacionário em sistemas de primeira ordem 294–295
 ganho de tensão 78–79
 gerador CA 53–54
 gerador de bit de paridade 127–129
 gerenciamento do motor de um carro 576–578
 gráficos de Bode 332–342
 construção 336–340
 identificação de sistemas 339–342
 para $G(s) = 1/s$ 333–334
 para $G(s) = K$ 332–334
 para sistemas de primeira ordem 333–335
 para sistemas de segunda ordem 334–337
 graus de liberdade 201–202
 gravação magnética 150–156
 códigos 152–155

H

handshaking 467–471
 handshaking habilitação e reconhecimento 469–470
 HDLC 525–526
 hidráulico 174–175

I

identificação de falhas 540–541
 autoteste 549
 forçar 551–552
 métodos sistemáticos de localização de falhas 548–549
 placas emuladoras 549–550
 simulação 549–551
 técnicas de detecção de falhas 540–542
 impedância de entrada 78–79
 impedância de saída 78–79
 impedância de saída de transdutores 38–39
 impressoras 149–151
 impressoras de matriz de pontos 149–150

- impressoras jato de tinta 149–150
 impressoras laser 149–151
 indicadores 144–145
 indutores 256–258
 equações 257–258
 inércia, momentos de 215–217
 inércia hidráulica 261–262
 equações 265–266
 inércia pneumática 264–265
 equações 265–266
 instruções de computador 417–418
 instrumentos virtuais 162–163
 integrais em transformadas de Laplace 591–592
 inteligência artificial 371–372
 interconexão de sistemas abertos (OSI) 525–530
 interface aberta de enlace de dados (ODI) 527–528
 interface serial de microcontroladores 482–484
 interfaseamento 464–465
 analógico 486–487
 buffers 467–470
 consulta 470–471
 decodificação de um display de sete segmentos 483–487
 handshaking 467–471
 interrupções 471–475
 PIAs 475–482
 exemplos 480–482
 inicialização 478–480
 sinais de interrupção 478–480
 requisitos 467–476
 ADC e DAC 468–469
 buffer/isolação 467–468
 controle de temporização 467–468
 conversão de código 467–468
 serial para paralelo 467–469
 serial 475–476
 interfaces de comunicação paralela 533–538
 Barramento de Instrumentação Hewlett Packard 533–534
 interface GPIB 533–536
 interfaces de comunicação serial 480–484, 529–534
 barramento CAN 533–534
 barramento I²C 531–533
 interface *firewire* 533–534
 interface USB 533–534
 loop de corrente de 20 mA 531–532
 RS-232C 529–532
 interfaces de entrada 379–380
 em CLPs 490–493
 atualização contínua 493–494
 endereços 494–495
 entrada/saída analógica 513–516
 entrada/saída em massa
 cópia 493–495
 processamento 493–495
 endereçamento de entrada/saída 464–468
 portas em PICs 404–407
 registradores 465–468
 testes em CLPs 551–552
 interfaces de saída 379–380
 interrupções 388–389, 471–475
 interrupções mascaráveis 472–474
 intervalo de quantização 103–104
 inversor em motores CA 239–240
- J**
- junções 202–203
- L**
- Lab VIEW 162–163
 lâminas bimetálicas 62–65
 largura de banda 342–343
 lasers em leitor de código de barras 579–581
 latch de endereço 388–389
 latch de saída 391–392
 latência 155–156
 LEDs 158–159
 lei das temperaturas intermediárias 67–68
 lei dos metais intermediários 67–68
 leis de Kirchhoff 257–258
 leitor de código de barras 577–581
 liberdade
 de movimento 201–203
 graus de 201–202
 linearidade em sistemas de engenharia 279–283
 linguagem assembly 417–418
 exemplos 427–431
 LUTs 433–437
 programas 424–431
 sub-rotinas 430–434
 linguagens 417–418
 lingueta 211–212
 lista de instruções em CLPs 499–502
 e desvios 500–502
 e funções lógicas 500–502
 localização de falhas, métodos sistemáticos 548–549
 localização de polos e resposta transiente 320–321
 lógica combinacional 120–135
 lógica digital 120–121
 lógica fuzzy 375–377
 lógica sequencial 120–121, 133–141
 flip-flop 133–137
 registradores 139–140
 sistema de alarme 138–139
 sistemas síncronos 136–138
 temporizador 555 139–141
 loop de programação 428–429
 loops de atraso 431–432
 loops de realimentação 21–22, 318–321
 e autorregulação 372
 loops em C 451–455
 declaração *for* 451–452
 declaração *switch* 452–455
 declaração *while* 452
- M**
- M68HC11 da Motorola 391–400
 conjunto de instruções 613–615
 controlador de interrupção 397–398
 COP 398–399
 handshaking 469–471
 interface de comunicação serial 475–476
 interruptores 472–474
 modos 396–398
 pinos do oscilador 397–398
 programa 427–428
 PWM 398–400
 temporizador 397–399
 manipulação de sinal 78–79
 mantissa em números de ponto flutuante 599–600
 manuseio de dados em CLPs 511–514
 comparação de dados 512–514
 conversões de código 513–514

- movimentação de dados 512–513
- operações aritméticas 513–514
- mapas de Karnaugh 606–611
- máquina de lavar roupa doméstica 411–413
- margem de fase 344–345
- margem de ganho 344–345
- massa 249–251
 - equações 252–253
- matemática binária 596–599
- MATLAB 624–629
 - funções de transferência 627–629
 - gráficos 626–628
 - veja também SIMULINK
- mecanismo alavanca-manivela 204–206
- mecanismo de alavanca dupla 204–205
- mecanismo de manivela dupla 204–205
- mecanismo de retorno rápido 205–206
- mecanismos 199–200
- mecatrônica
 - e projetos tradicionais 14–15
 - origens 11–12
 - processo de projeto 13–15
 - projetos 610–567
 - balança de banheiro 563–567
 - chave temporizada 610–559
 - movimento do limpador de para-brisa 559–564
 - sistemas, exemplos de 12–13, 29–32
- medição de temperatura, saída binária a partir da 167–169
- medidor de bobina móvel 145–146
- medidores digitais na apresentação de dados 145–147
- memória 379–380
- memória em sistemas microprocessados 384–388
 - EEPROM 387–388
 - EPROM 386–388
 - PROM 386–387
 - RAM 387–388
 - ROM 385–387
- método de paridade para detecção de erro 115–116
- método de sintonia por reação de processo 363–364
- método do ciclo máximo para sintonia 364–365
- microcontroladores 12–13, 380–381, 391–411
 - 8051 da Intel 399–404
 - acesso externo 400–401
 - ALE 400–401
 - entrada/saída serial 402–403
 - interrupção 402–404
 - portas de entrada/saída 399–401
 - PSEN 400–401
 - RESET 402–403
 - temporização 402–403
 - XTAL 402–403
 - aplicações 409–413
 - máquina de lavar roupa 411–413
 - sistema de medição de temperatura 409–412
 - interface serial 482–484
 - M68HC11 da Motorola 391–400
 - controlador de interrupção 397–398
 - COP 398–399
 - modos 396–398
 - pinos do oscilador 397–398
 - PWM 398–400
 - temporizador 397–399
 - microcontroladores Microchip™ 402–410
 - seleção 409–411
 - sistemas embarcados 437–440–441
- microprocessadores 12–13, 379–380
- microprocessadores em sistemas embarcados 379–380
- minuendo em matemática binária 597–598
- modelo de referência em sistemas adaptativos 367–368
- modelos matemáticos 248–250
- modo de comunicação full-duplex 522–523
- modo de comunicação half-duplex 522–523
- modo de controle derivativo 349–351, 353–357
 - com modo proporcional 354–357
- modo de controle integral 350–351, 355–359
 - com modo proporcional 357–359
- modo de controle *on/off* 349–352
- modo PID de controle 350–351, 357–359
- modo proporcional de controle 349–354
 - com controle derivativo 354–357
 - com modo integral 357–359
 - resposta do sistema 353–354
- modos de comunicação de dados em série 521–523
 - modo *full-duplex* 522–523
 - modo *half-duplex* 522–523
 - modo *simplex* 521–522
- modos de controle 347–348
- modulação de frequência (FM)
 - codificação 153–154
- modulação de frequência modificada (MFM)
 - codificação 153–154
- modulação de pulso 97–99
- modulação por amplitude de pulso 97–98
- modulação por largura de pulsos (PWM) 97–98, 232–234
- molas 249–251, 280–281
 - equações 252–253
- molas de torção 251–252
- momento de pico em sistemas dinâmicos 302–304
- momentos de inércia 215–217, 251–252
 - em sistema pinhão-cremalheira 275–277
 - equações 252–253
- MOSFETs 228–229
- motor bipolar 242–243
- motor CA de indução em gaiola de esquilo monofásico 236–237
- motor de meio passo 244–245
- motor de palhetas 195–196
- motor de passo de ímã permanente 240–241
- motor de passo de relutância variável 239–241
- motor de passo híbrido 240–241
- motor unipolar 243–244
- motores
 - defeitos 544–545
 - seleção 215–217

motores CA 236–240
 motores CC 229–237
 como sistemas eletromecânicos 276–280
 funções de transferência 318–319
 motores CC com excitação em separado 232–233
 motores CC compostos 232–233
 motores CC de imã permanente sem escovas 235–237
 motores CC tipo escova 229–232
 com bobinas de campo 231–233
 controle de 232–236
 motores com enrolamentos em série 231–233
 motores controlados pela armadura 277–279
 realimentação negativa em 320–321
 motores controlados pelo campo 279–280
 funções de transferência 318–319
 motores de enrolamento shunt 232–233
 motores de passo 239–245
 atraso com 435–437
 características 242–243
 controle 241–245
 em câmeras digital 573–574
 em limpadores de pára-brisa 559–560
 especificações 241–242
 meio passo 244–245
 motores síncronos 237–239
 movimento de translação 200–201
 movimento do limpador de pára-brisa 559–564
 adaptador de interface periférica 559–560
 movimento rotacional 200–201
 movimento rotacional para translacional 210–211
 movimento, tipos de 200–203
 multiplexação por divisão de tempo 113–114
 multiplexadores 112–114
 multiplexadores digitais 112–114
 multivibrador monoestável 139–141

N

não linearidade em sistemas de engenharia 280–281
 níveis de quantização 103–104
 nível de sinal 78–79
 nós 202–203
 nós em redes 522–523
 NuBus 536–537
 números binários 101–102, 595–597
 números binários não-sinalizados 597–598
 números binários sinalizados 597–598
 números de ponto flutuante 599–600

O

obturador de igual percentagem 192–193
 obturador do tipo linear 191–192
 obturadores de abertura rápida 191–192
 operação latch de dados 138–139
 operação unipolar de DACs 106–107
 operações aritméticas em C 447–448
 operações aritméticas em CLPs 513–514
 operações baseadas em clock 347–348
 operações baseadas em evento 347–348
 operações binárias em C 448–449
 operador relacional em C 447–448
 operadores lógicos em C 448–449
 operando 421–422, 425–426
 optoisoladores 90–91
 oscilações autossustentadas 343–344
 osciloscópio de raios catódicos 146–148
overshoot em sistemas dinâmicos 302–304

P

palavras-chave em C 444–445
 parafuso de avanço 210–211
 parcelas em matemática binária 596–597

paridade ímpar 115–116
 PASCAL 417–418
 passa banda 91–92
 percepção 371–373
 periféricos 464–465
Peripheral Interface Controller (PIC) 402–411
 arquitetura Harvard 402–406
 características 404–409
 clear e reset mestres 408–409
 entrada do cristal 408–409
 entrada/saída serial 408–409
 entradas analógicas 406–407
 porta escrava paralela 408–409
 portas de entrada/saída 404–407
 temporizadores 406–408
 conjunto de instruções 618–619
 família 409–411
 registradores de propósito geral 408–410
 pico de ressonância 342–343
 pilha 430–431
 pinhão 207–208
 pixels 148–149
 placas *plug-and-play* 162–163
 placas *plug-in* para DAQs 161–164
 plano *s* em funções de transferência 323
 pneumático 174–175
 polos de campo em motores CC 229–230
 polos de chaves 70–71
 polos em funções de transferência 321–322
 ponte de Wheatstone 91–98
 calibração 169–170
 compensação de temperatura 93–97
 compensação de termopar 96–98
 em balanças de banheiro 564–566
 ponteiros em C 455–458
 aritmética com 456–457
 e arranjos 456–458
 pontos de corte em gráficos de Bode 333–334
 porta escrava paralela em PICs 408–409
 portas lógicas 120–121
 aplicações 127–135
 codificador 128–130
 comparador digital 127–129

- conversor de código 129–135
 geradores de bit de paridade 127–129
 combinação 125–128
 porta AND 120–123
 porta EX-OR 125–126
 porta NAND 123–127
 porta NOR 124–127
 porta NOT 123–124
 porta OR 122–124
 posição angular de uma polia 166–168
 posições de chaves 70–71
 posições limites 204–205
 potência em números de ponto flutuante 599–600
 potenciômetro rotativo 276–277
 precisão de transdutores 35–36
 pre-processador em C 445–447
 pressão manométrica 190–191
 primeira parcela em matemática binária 596–597
 primeira propriedade do deslocamento da transformada de Laplace 590–591
 princípio da restrição mínima 201–202
 princípio da superposição 85–86, 279–280
 processamento de sinais digitais 116–118
 processamento de sinais discretos no tempo 116–117
 produto de somas em álgebra Booleana 604–605
 programa 26–27, 417–418
 programação 412–415
 programação ladder em CLPs 494–500
 funções lógicas 497–500
 projeto cinemático 201–202
 PROM (memória apenas de leitura programável) 386–387
 propriedade da linearidade de transformadas de Laplace 590–591
 propriedade do deslocamento em transformadas de Laplace 590–591
 propriedade do deslocamento no domínio do tempo em transformadas de Laplace 590–591
 proteção no condicionamento de sinal 77–78, 88–92
 protocolo administrativo e técnico (TOP) 527–528
 protocolo Bisync 525–526
 Protocolo de Automação da Manufatura (MAP) 526–527
 protocolos para transmissão e dados 524–526
 protocolos wireless 537–538
 pseudocódigo 412–415
 pseudo-operações 425–426
- R**
- raciocínio 373–377
 baseado em regras 373–377
 mecanismo de 373–375
 RAM (memória de acesso aleatório) 387–393
 rampa em sistemas dinâmicos 288–290
 rangeabilidade de obturadores 193–194
 razão de amortecimento em sistemas dinâmicos 302–304
 razão de Poisson 95–96
 razão de rejeição em modo comum (CMRR) 84–85
 razão de transferência 90–91
 realimentação 17–21
 realimentação de velocidade 364–365
 realimentação negativa 21–22, 318–319
 exemplos de 320–321
 realimentação positiva 21–22
 reconhecimento de padrão 372–373, 377–378
 rede de longa distância (WAN) 523–524
 rede de resistores ponderados 105–106
 rede escada R-2R 105–106
 rede local (LAN) 523–524
 rede Novell 527–528
 redes 522–525
 padrões 526–530
 topologia 522–524
 redes neurais 372–374
reed smitch 50–51
 registrador acumulador 382–384
 registrador contador de programa 383–384
 registrador de código de condição 383–384
 registrador de endereço de memória 384–385
 registrador de flags 383–384
 registrador de instrução 384–385
 registrador de status 383–384
 registrador ponteiro de instrução 383–384
 registrador ponteiro de pilha 384–385
 registradores 144–145
 registradores de dados 139–140
 registradores de deslocamento de CLPs 509–511
 registradores de função especial 402–404
 registradores de propósito geral 384–385
 registradores gráficos analógicos 146–147
 registros em sistemas microprocessados 382–385
 regra de Byer 374–377
 rejeita faixa 91–92
 relés 221–223
 falhas em 544–545
 relés auxiliares em CLPs 503–504
 relés de atraso de tempo 222–223
 relés internos em CLPs 503–504
 relés mestre em CLPs 510–512
 repetibilidade em transdutores 36–38
 repique de chave 71–72
 reprodutibilidade de transdutores 36–38
 requisição de repetição automática (ARQ) 542–543
 resistência hidráulica 260–261
 equações 265–266
 resistência pneumática 263
 equações 265–266
 resistência térmica 268–269
 equações 269–270
 resistor 256–258
 equações 257–258

- resolução 103–104, 110–112
 resolução de transdutores 37–38
 resposta aperiódica 576–577
 resposta de frequência 329–333
 - em sistemas de primeira ordem 329–332
 - em sistemas de segunda ordem 331–333
 especificação de desempenho 342–344
 estabilidade 343–345
 gráficos de Bode 332–342
 - construção 336–340
 - identificação de sistemas 339–342
 - para $G(s) = 1/s$ 333–334
 - para $G(s) = K$ 332–334
 - para sistemas de primeira ordem 333–335
 - para sistemas de segunda ordem 334–337
 - para entrada senoidal 236–237
 resposta forçada em sistemas dinâmicos 287–289
 resposta natural
 - em sistemas de primeira ordem 289–291
 - em sistemas dinâmicos 287–289
 resposta transiente e localização de polos 320–321
 resposta transiente em sistemas dinâmicos 288–289
 respostas de estado estacionário em sistemas dinâmicos 288–289
 restrição mínima, princípio da 201–202
 retornos em C 445–446
 reversa em motores CC 230–231
 robô de manipulação 566–571
 roda inativa 208–210
 rolamento de esferas autoalinhas 214–215
 rolamento de esferas, tipo via dupla 214–215
 rolamento de filme sólido 214–215
 rolamento de rolete cônico 215–216
 rolamento de rolete tipo agulha 215–216
 rolamento radial hidrodinâmico 213–214
 rolamento radial hidrostático 213–214
 rolamento rígido de esferas 214–215
 rolamentos axiais 213–215
 rolamentos axiais planos 213–215
 rolamentos de esferas e roletes 214–215
 rolamentos de esferas, tipo entalhado 214–215
 rolamentos de rolete cilíndrico 215–216
 rolamentos de tração axial 213–214
 rolamentos, seleção de 215–216
 ROM (memória apenas de leitura) 385–387, 391–393
 rótulos em linguagem assembly 425–426
 ruído 78–79
- S**
- saída binária a partir de medição de temperatura 167–169
 saltos condicionais em CLPs 511–512
 saltos em CLPs 511–512
 saturação em transistores bipolar 226–227
 Schmitt trigger 71–73
 seguidor de tensão 80–81
 seguidores de came 205–207
 segundo teorema do deslocamento nas transformadas de Laplace 590–591
 sensibilidade de carga 59–60
 sensibilidade de tensão 59–60
 sensibilidade de transdutores 35–37
 sensor com medição de turbina 61–62
 sensor de pressão diferencial 62–63
 sensor de toque 60–61
 sensor indutivo 576–577
 sensor placa de orifício 61–62
 sensores 16–17, 34–36
 - falhas em 543–545
 - seleção de 68–71
 - terminologias de desempenho 35–41
 sensores baseado em potenciômetro 40–43
 sensores capacitivos 43–45
 sensores de deslocamento 40–41
 sensores de deslocamento push-pull 44–45
 sensores de efeito Hall 50–53, 235–236
 sensores de força 55–57
 - resposta forçada em sistemas de primeira ordem 290–292
 sensores de luz 68–70
 sensores de movimento 52–57
 sensores de nível de líquido 61–63
 sensores de posição 40–41
 sensores de pressão 55–63
 - calibração de 169–170
 sensores de pressão de fluido 55–61
 sensores de pressão de fluido de fole 57–58
 sensores de pressão de fluido em cápsula 57–58
 sensores de proximidade 40–41
 sensores de proximidade por corrente parasita 46–47
 sensores de temperatura 62–69
 - calibração de 169–170
 sensores de velocidade 52–57
 sensores e vazão de líquido 60–62
 sensores fotossensíveis 50–51
 sensores inteligentes 34–36
 sensores piezoelétricos 58–60
 sensores piroelétricos 53–57
 sensores piroelétricos duais 54–55
 sensores pneumáticos 49–50
 sensores *strain gauge* 42–44
 sequenciamento em PLCs 504–506
 sequências de escape em C 449
 SIMULINK 628–629
 sinais de clock 136–137
 sinais digitais 101–102
 sinal
 - filtragem 91–93
 - tipo retangular 77–78
 sintonia do controlador 362–365
 - método de reação de processo 363–364
 - método do ciclo máximo 364–365

- sistema de alarme 138–139
- sistema de alarme de temperatura 165–167
- sistema de articulação 204–205
- sistema de comunicação distribuído 520–521
- sistema de comunicação hierárquico 520–521
- sistema de controle de fluido 194–196
- sistema de controle digital 25–27
- sistema de entrada e saída básico (NetBIOS) 527–528
- sistema de gerenciamento de um motor 30–32
- sistema de medição de temperatura 409–412
- sistema de numeração decimal 595–597
- sistema de numeração hexadecimal 595–597
- sistema de numeração octal 595–597
- sistema decimal codificado em binário (BCD) 595–597
- sistema pinhão-cremalheira 210–211, 275–277
- sistema porca-parafuso 210–211
- sistemas 14–17
 - com malhas de realimentação 318–321
 - em série 317–319
 - exemplos de 318–319
- sistemas assíncronos 136–137
- sistemas com atuadores mecânicos 199–201
 - movimentos, tipos de 200–203
 - efeito de carga 202–203
 - liberdades e restrições 201–203
- sistemas com parâmetros concentrados 249–250
- sistemas conectados 16–17
- sistemas de amortecimento crítico 297–298
- sistemas de atuação elétricos 220–221
 - chaves 220–223
 - estado sólido 222–229
 - relés 221–223
 - motores CA 236–240
 - motores CC 229–237
 - motores de passo 239–245
 - solenoides 228–230
- sistemas de controle 17–29
 - realimentação 17–21
- sistemas de controle adaptativo 364–368
 - autossintonia 366–368
 - controle de ganho escalonado 366–367
- sistemas de controle adaptativo programáveis 366–367
- sistemas de controle analógico 25–27
- sistemas de controle contínuo 347–348
- sistemas de controle discreto 347–348
- sistemas de display 17–18
- sistemas de engenharia 275–276
 - linearidade 279–283
 - não linearidade 280–281
- sistemas eletromecânicos 276–280
- sistemas rotacional-translacional 275–277
- sistemas de malha aberta 20–22
- sistemas de malha fechada 20–22
 - elementos de 21–25
 - funções de transferência em 318–319
- sistemas de medida 15–18
 - calibração de 168–170
 - displays para 144–145, 164–168
 - células de carga em 164–166
 - medida de temperatura 167–169
 - posição angular 166–168
 - sistema de alarme de temperatura 165–167
 - teste 168–169
- sistemas de modelamento 15–17
- sistemas de numeração 595–601
- sistemas de primeira ordem
 - como sistemas dinâmicos 289–296
 - constante de tempo 293–296
 - exemplos de 291–294
 - resposta forçada 290–292
 - resposta natural 289–291
 - constante de tempo em 293–296
 - entrada em degrau em 313–315
- funções de transferência 313–316
 - entrada em degrau 313–315
 - exemplos 313–316
 - ganho de estado estacionário em 294–295
 - gráficos de Bode para 333–335
 - resposta de frequência em 329–332
 - resposta forçada nos 290–292
 - resposta natural em 289–291
 - transdutores 291–292
 - transformadas de Laplace em 313–314
- sistemas de segunda ordem
 - condições de estado estacionário 299–300
 - entrada em degrau 316–317
 - funções de transferência 315–318
 - exemplos 317–318
 - gráficos de Bode para 334–337
 - resposta de frequência 331–333
 - sistemas dinâmicos 295–303
 - exemplos 299–303
 - medidas de desempenho 302–306
 - resposta forçada 298–300
 - resposta natural 296–299
 - sistemas massa-mola-amortecedor 295–297
 - transformadas de Laplace 315–316
- sistemas de torção 301–302
- sistemas dinâmicos
 - entradas, formas de 288–290
 - equações diferenciais em 286–288
 - identificação de sistemas 305–306
 - modelamento 286–287
 - respostas natural e forçada 287–289
 - respostas transiente e de estado estacionário 288–289
 - sistemas de primeira ordem 289–296
 - constante de tempo 293–296
 - exemplos de 291–294
 - resposta forçada 290–292
 - resposta natural 289–291
 - sistemas de segunda ordem 295–303
 - exemplos de 299–303

- medidas de desempenho 302–306
 resposta forçada 298–300
 resposta natural 296–299
- sistemas eletromecânicos 276–280
 motor CC 276–280
 potenciômetro rotativo 276–277
- sistemas embarcados 12–14, 437–441
- sistemas estáveis 320–323
- sistemas hidráulicos 174–177
 defeitos em 544–545
 em robôs de manipulação 569
- sistemas instáveis 320–323
- sistemas massa-mola-amortecedor 253–256
 como sistema de segunda ordem 295–297
- sistemas mecatrônicos
 acionador de disco rígido 579–582
 câmera digital 571–577
 cancelas em estacionamento para carros 570–573
 gerenciamento do motor de um carro 576–578
 leitor de código de barras 577–581
 robô de manipulação 566–571
- sistemas microprocessados 379–393
 barramentos 380–382
 em balança de banheiro 563–564
 entrada/saída 387–389, 391–393
 exemplos 388–393
 falhas em 545–549
 memória 384–388
 microprocessador 381–385
 registradores 382–385
 unidade de controle 382–383
 unidade lógica e aritmética (ALU) 382–383
- sistemas não programáveis 22–23
- sistemas pneumáticos 176–179
 defeitos em 544–545
 em robôs de manipulação 569
- sistemas programáveis 22–23
- sistemas rotacionais 251–253
- sistemas rotacional-translacional 275–277
- sistemas síncronos 136–138
- sistemas subamortecidos 298–299, 302–303
- sistemas superamortecidos 297–298
- software 387–388, 417–418
- solenoides 228–230
- soma de produtos em álgebra Booleana 604–605
- soma em matemática binária 596–597
- span de transdutores 35–36
strain gauge 94–95
- strings em C 448–449
- sub-rotinas 430–434
- sub-rotinas de atraso 431–434
- subtraendo em matemática binária 597–598
- superposição, princípio da 85–86, 279–280
- ## T
- tabela de vetores 471–472
- tabelas de busca 433–437
- tabelas-verdade 121–127
 em álgebra Booleana 604–607
- tacogerador 52–54
- tacogerador de relutância variável 52–54
- teclados 72–74
- tempo de acomodação de transdutores 39–40
- tempo de acomodação em sistemas dinâmicos 304–305
- tempo de busca 155–156
- tempo de conversão 102–103, 110–112
- tempo de resposta de transdutores 38–40
- tempo de subida de transdutores 39–40
- tempo de subida em sistemas dinâmicos 302–303
- tempo morto de transdutores 37–38
- temporizador 555 139–141, 558–559
- temporizador *matchdog* 541–542
- temporizadores
 em CLPs 505–507
 em PICs 406–408
- temporizadores em cascata em CLPs 506–507
- tensão de offset 88–89
- tensão em modo comum 83–85
- tensão reversa em relés 221–222
- teorema da amostragem 103–105
- teorema da amostragem de Shannon 103–104
- teorema do valor final 349–350
- teorema do valor inicial nas transformadas de Laplace 591–592
- teoremas de De Morgan 603–605
- teoremas do valor final nas transformadas de Laplace 591–592
- termistores 63–66
 no gerenciamento do motor de um carro 576–577
- termiodiodos 64–66
- termopares 66–69
- termopilhas 68–69
- terra virtual 79–80
- teste *checksum* 548–549
- teste de programa em CLPs 551
- tiristor 223–226
- topologia de redes 522–524
- torque 215–217
- transdutor analógico 34–35
- transdutor digital 34–35
- transdutores 34–36
 como sistema de primeira ordem 291–292
- transferência paralela de dados 467–470
 habilitação e reconhecimento 469–470
- transformadas de Laplace 311–314, 585–594
 em sistemas de primeira ordem 313–314
 em sistemas de segunda ordem 315–316
 frações parciais 592–594
 função degrau unitário 586–588
 função impulso 587–589
 inversa 592–594
 padrão 588–590
 primeiro princípio 585–587
 propriedades 588–592
- transformadas de Laplace parciais 586–588
- transformador diferencial linear variável (LVDT) 44–47
 com foles 58–59

- transformador diferencial variável rotativo (RVDT) 46–47
transformadores diferenciais 44–47
transistor de efeito de campo de óxido metálico (MOSFET) 228–229
transistores 64–66
transistores bipolares 225–229
transistores buffer 227–228
transistores par Darlington 227–228
transmissão assíncrona 475–476, 521–522
transmissão *broadband* 523–525
transmissão de dados
 controle de acesso à rede 523–524
 em paralelo 520–522
 protocolos 524–526
 redes 522–525
 serial 521–523
 taxa de 521–522
transmissão em banda base 524–525
transmissão paralela de dados 520–522
transmissão por correia 211–213
 tipos 212–213
transmissão por corrente 213–214
transmissão síncrona 521–522
transmissor/receptor assíncrono universal (UART) 480–482
trens de engrenagens 207–211
- trens de engrenagens compostas 208–211
triacs 223–226
tubo de Bourdon 57–58
- U**
- unidade central de processamento (CPU) 379–385
unidade de amostragem e retenção 102–103
unidade de display visual (VDU) 147–150
- V**
- válvula de dupla sede 191–192
válvula de gatilho 179–180
válvula de sede simples 191–192
válvula de solenoide simples 181–182
válvulas 178–179
 símbolos 179–182
válvulas de carretel 178–180, 281–283
válvulas de controle de pressão 182–184
válvulas de controle de processo 190–196
 corpos de válvulas 191–194
 dimensionamento 193–196
 obturadores 191–194
válvulas de controle direcional 178–183
válvulas de controle proporcional 187–189
válvulas de posição finita 178–179
válvulas de posição infinita 178–179
válvulas direcionais 182–183
válvulas servo-controladas 187–189
válvulas-piloto operadas 181–183
variáveis em C 446–448
varredura entrelaçada em tubos de raios catódicos 147–149
varredura não entrelaçada em tubos de raios catódicos 147–149
verificação de paridade 155–156
verificação de redundância cíclica (CRC) 542–544
verificação por diagnóstico 541
verificação por paridade e codificação de erro 541
verificação por replicação 540
verificação por somas 115–116
verificação por temporização 541
verificação por temporizadores *matchdog* 540–543
verificação reversa 541
verificações por valor esperado 540
vetor de interrupção 471–472
voltímetro digital 146–147