

## 9

# Controle discreto utilizando controladores lógicos programáveis e computadores pessoais

## CONTEÚDO DO CAPÍTULO

- 9.1 Controle discreto de processos
  - 9.1.1 Controle lógico
  - 9.1.2 Sequenciamento
- 9.2 Diagramas de lógica ladder
- 9.3 Controladores lógicos programáveis (CLPs)
  - 9.3.1 Componentes dos CLP
  - 9.3.2 Ciclo de operação do CLP
  - 9.3.3 Capacidades adicionais do CLP
  - 9.3.4 Programando o CLP
- 9.4 Computadores pessoais utilizando lógica *soft*

O controle numérico (Capítulo 7) e a robótica industrial (Capítulo 8) preocupam-se, primeiramente, com o controle do movimento, pois as aplicações das máquinas-ferramenta e dos robôs envolvem movimentação de uma ferramenta de corte ou um efetuator, respectivamente. Uma categoria mais geral de controle é o controle discreto, definido na Seção

5.2.2. Neste capítulo, fazemos uma descrição mais completa do controle discreto e examinamos os dois principais controladores industriais utilizados na implementação do controle discreto: (1) controladores lógicos programáveis (do inglês, *programmable logic controllers* — CLPs ou PLCs,) e (2) computadores pessoais (PCs).

## 9.1 CONTROLE DISCRETO DE PROCESSOS

Os sistemas de controle discreto de processos lidam com parâmetros e variáveis que são discretos e alteram valores em momentos discretos do tempo. Dependendo da aplicação, os parâmetros e as variáveis costumam ser binários e podem assumir dois valores: 1 ou 0. Os valores significam ligado ou desligado, verdadeiro ou falso, objeto presente ou ausente, valor de tensão alto ou baixo. No controle discreto de processos, as variáveis binárias estão associadas a sinais de entrada, que chegam ao controlador, e a sinais de saída, que são enviados do controlador. Os sinais de entrada costumam ser gerados por sensores binários, tais como interruptores fim de curso ou fotossensores que fazem interface com o processo. Os sinais de saída são gerados pelo controlador para operar o processo em res-

posta aos sinais de entrada e como função do tempo. Esses sinais de saída ligam e desligam interruptores, motores, válvulas e outros atuadores binários relacionados ao processo. Na Tabela 9.1, apresentamos uma lista de sensores e atuadores binários juntamente à interpretação de seus valores, 0 e 1. O objetivo do controlador é coordenar as várias ações do sistema físico, tais como a transferência de peças para um sistema de fixação ou a alimentação do cabeçote de usinagem, dentre outras funções.

O controle de processos discreto pode ser dividido em duas categorias: (1) controle lógico, que se preocupa com alterações no sistema orientadas por eventos, e (2) sequenciamento, que se preocupa com alterações no sistema orientadas pelo tempo. Ambas referenciadas como *sistemas de comutação* (ou chaveamento), pois comutam seus valores de saída entre ligado e desligado em resposta às mudanças nos eventos ou no tempo.

Tabela 9.1 Sensores e atuadores binários utilizados no controle de processos discreto

Sensor	Interpretação 1/0	Atuador	Interpretação 1/0
Interruptor de fim de curso	Com contato/sem contato	Motor	Ligado/Desligado
Fotodetector	Ligado/Desligado	Relé de controle	Fechado/Aberto
Interruptor de botão de pressão	Ligado/Desligado	Luz	Ligado/Desligado
Temporizador	Ligado/Desligado	Válvula	Fechado/Aberto
Relé de controle	Fechado/Aberto	Embreagem	Conectado/Livre
Disjuntor	Fechado/Aberto	Solenóide	Energizado/Não energizado

### 9.1.1 Controle lógico

Também chamado de *controle lógico combinacional*, é um sistema de comutação cuja saída a qualquer instante é determinada exclusivamente pelos valores atuais das entradas. Um sistema de controle lógico não possui memória e não considera nenhum valor anterior de sinais de entrada na determinação do sinal de saída. Tampouco dispõe de características que sejam diretamente executadas como função do tempo.

Vamos usar um exemplo da robótica para ilustrar o controle lógico. Suponha que em uma aplicação de carregamento de uma máquina, o robô esteja programado para pegar uma peça em estado bruto em um ponto de parada conhecido, paralelo a um transportador, e posicioná-lo em uma prensa de forjamento. Três condições devem ser satisfeitas para que o ciclo de carregamento seja iniciado. Primeiro, a peça em estado bruto deve estar no ponto de parada; segundo, a prensa de forjamento deve ter concluído o processo na peça anterior; terceiro, a peça anterior

deve ter sido removida da matriz. A primeira condição pode ser indicada por meio de um interruptor simples, que percebe a presença da peça na parada do transportador e transmite o sinal LIGADO para o controlador do robô. A segunda condição pode ser indicada pela prensa de forjamento, que envia um sinal LIGADO após concluir o ciclo anterior. A terceira condição pode ser determinada por um fotodetector posicionado de forma a identificar a presença ou a ausência da peça na matriz de forjamento. Quando a peça acabada é removida da matriz, um sinal LIGADO é transmitido pela fotocélula. Esses três sinais devem ser recebidos pelo controlador do robô para que o ciclo de trabalho seguinte seja iniciado. Quando esses sinais de entrada forem recebidos pelo controlador, o ciclo de carregamento do robô é iniciado. Nenhuma condição ou nenhum histórico anterior é necessário.

**Elementos do controle lógico.** Os elementos básicos do controle lógico são as portas lógicas AND (E), OR (OU) e NOT (NÃO ou Inversora). Em cada caso, a

porta lógica é projetada de modo a oferecer um valor específico de saída com base nos valores de entrada. Tanto para as entradas como para as saídas, os valores podem ser de dois níveis: os valores binários 0 e 1. Para fins de controle industrial, definimos que 0 é DESLIGADO e 1 é LIGADO.

A porta AND fornece o valor de saída 1, se todas as entradas forem 1; caso contrário, o valor é 0. A Figura 9.1 ilustra a operação de uma porta lógica AND. Se ambos os interruptores X1 e X2 (representando as entradas) do circuito estiverem fechados, então a lâmpada Y (representando a saída) estará ligada. A porta AND deve ser usada em um sistema de produção automatizado para indicar que duas (ou mais) ações foram concluídas com sucesso, sinalizando, assim, que a próxima etapa do processo deve ser iniciada. O sistema de intertravamento no exemplo anterior de forjamento usando um robô ilustra uma porta AND. As três condições devem ser satisfeitas antes que o carregamento da prensa de forjamento possa acontecer.

A porta OR fornece uma saída de valor 1, se qualquer uma das entradas for 1; caso contrário, o valor é 0. A Figura

9.2 mostra como opera uma porta OR. Nesse caso, os dois sinais X1 e X2 estão organizados em um circuito paralelo de modo que, em se fechando qualquer um dos interruptores, a lâmpada Y seja ligada. Um uso possível da porta OR nos sistemas de manufatura é o monitoramento da segurança. Suponha que dois sensores sejam utilizados para monitorar a ocorrência de dois riscos diferentes na segurança. Quando qualquer um desses riscos acontecer, o sensor respectivo emitirá um sinal positivo, que soará a sirene de um alarme.

Tanto a porta AND como a porta OR podem ser utilizadas com duas ou mais entradas. A porta NOT possui somente uma entrada. Ela inverte o sinal de entrada: se ele for 1, a saída é 0; se ele for 0, a saída é 1. A Figura 9.3 mostra um circuito no qual o interruptor de entrada X1 está presente em um circuito paralelo com uma saída tal que a tensão flui pelo percurso inferior quando o interruptor está fechado ( $Y = 0$ ) e pelo percurso superior quando o interruptor está aberto ( $Y = 1$ ). A porta NOT pode ser usada para abrir um circuito após o recebimento de um sinal de controle.

Figura 9.1 Circuito elétrico ilustrando a operação de uma porta lógica AND

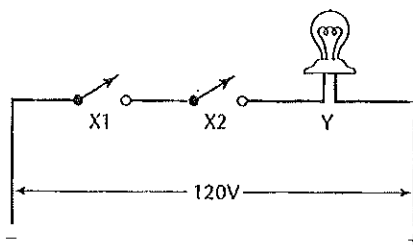


Figura 9.2 Circuito elétrico ilustrando a operação de uma porta lógica OR

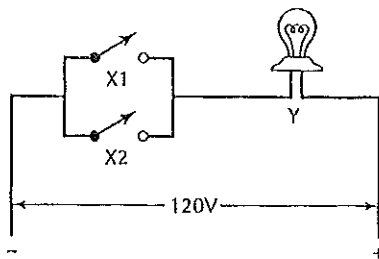
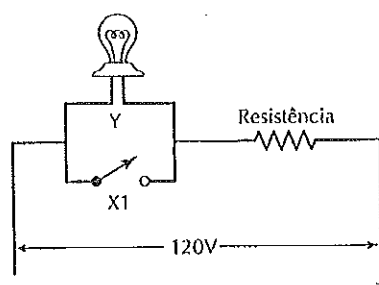


Figura 9.3 Circuito elétrico ilustrando a operação de uma porta lógica NOT



**Álgebra booleana e tabelas-verdade.** Os elementos lógicos formam a base de uma álgebra especial desenvolvida por volta de 1847 por George Boole e que leva seu nome. Seu propósito original era oferecer um meio simbólico de testar se declarações complexas de lógica eram VERDADEIRAS ou FALSAS. Na verdade, Boole a denominou *álgebra lógica*. Foi somente um século mais tarde que a álgebra booleana provou ser útil nos sistemas de lógica digital. Descrevemos brevemente alguns de seus fundamentos.

Na álgebra booleana, a função AND é representada como:

$$Y = X1 \cdot X2 \quad (9.1)$$

Essa expressão é chamada de produto lógico de X1 e X2. Como declaração lógica, significa: Y é verdadeiro se ambos X1 e X2 forem verdadeiros; senão, Y é falso. A tabela-verdade costuma ser sempre utilizada para apresentar a operação de sistemas lógicos. Uma *tabela-verdade* é uma tabulação de todas as combinações de valores de entrada correspondentes aos valores de saída. A tabela-verdade para a porta AND com quatro combinações possíveis para dois valores de entrada é apresentada na Tabela 9.2.

Na notação da álgebra booleana, a função OR é representada como:

$$Y = X1 + X2 \quad (9.2)$$

Essa expressão é chamada de soma lógica de X1 e X2. Na lógica, a declaração significa que Y é verdadeiro se um dos valores X1 ou X2 for verdadeiro; senão, Y é falso. As saídas da porta OR com quatro combinações possíveis para duas variáveis binárias de entrada são listados na Tabela 9.3.

A função NOT é conhecida como negação ou inversão da variável e indicada por uma barra acima da variável (por exemplo, X1). A tabela-verdade para a função NOT é apresentada na Tabela 9.4, e a equação booleana correspondente é a seguinte:

$$Y = \overline{X1} \quad (9.3)$$

Além dos três básicos, existem dois elementos que podem ser utilizados nos circuitos de comutação: as portas NAND e NOR. A porta NAND é formada pela combinação das portas AND e NOT em sequência, resultando na tabela-verdade mostrada na Tabela 9.5(a). Em forma de equação, tem-se:

$$Y = \overline{X1 \cdot X2} \quad (9.4)$$

A porta lógica NOR é formada pela combinação de uma porta OR seguida de uma porta NOT, resultando na tabela-verdade mostrada na Tabela 9.5(b). A equação da álgebra booleana para a porta NOR é a seguinte:

$$Y = \overline{X1 + X2} \quad (9.5)$$

Tabela 9.2 Tabela-verdade para a porta lógica AND

Saídas		Entradas
X1	X2	$Y = X1 \cdot X2$
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 9.3 Tabela-verdade para a porta lógica OR

Saídas		Entradas
X1	X2	$Y = X1 + X2$
0	0	0
0	1	1
1	0	1
1	1	1

Tabela 9.4 Tabela-verdade para a porta lógica NOT

Saídas	Entradas
X1	$Y = \overline{X1}$
0	1
1	0

Tabela 9.5 Tabelas-verdade para as portas lógicas (a) NAND e (b) NOR

(a) NAND			(b) NOR		
Saídas		Entradas	Saídas		Entradas
X1	X2	$Y = \overline{X1 \cdot X2}$	X1	X2	$Y = \overline{X1 + X2}$
0	0	1	0	0	1
0	1	1	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0

Diversas técnicas de diagramação foram desenvolvidas para a representação dos elementos lógicos e seus relacionamentos em um determinado sistema de controle lógico. O diagrama lógico é um dos métodos mais comuns. Os símbolos usados nesse tipo de diagrama são apresentados na Figura 9.4. Demonstramos o uso do diagrama lógico em diversos exemplos ao longo desta seção.

Existem certas leis e certos teoremas na álgebra booleana, citados na Tabela 9.6, que podem sempre ser aplicados à simplificação de circuitos lógicos e à redução do número de elementos necessários na implementação da lógica. O resultado é a economia de *hardware* e/ou de tempo de programação.

#### EXEMPLO 9.1

##### Carregamento da máquina pelo robô

O exemplo do carregamento da máquina pelo robô descrito no início da Seção 9.1.1 demandava o cumprimento de três condições antes que a sequência de carregamento pudesse ser iniciada. Determine a expressão da álgebra booleana e o diagrama lógico para esse sistema de intertravamento.

**Solução:** Seja X1 o valor que determina se a peça em estado bruto está presente no ponto de parada do transportador ( $X1 = 1$  para presente,  $X1 = 0$  para ausente); X2 o valor que determina se a peça anterior está concluída ( $X2 = 1$  para concluída,  $X2 = 0$  para ausente); e X3 o valor que determina se a peça anterior foi removida da

Figura 9.4 Símbolos usados para as portas lógicas: ANSI<sup>1</sup> e ISO<sup>2</sup>

	Simbologia ANSI	Simbologia ISO
AND		
OR		
NOT		
NAND		
NOR		

Tabela 9.6. Leis e teoremas da álgebra booleana

Lei comutativa:	Lei da absorção:
$X + Y = Y + X$	$X \cdot (X + Y) = X + X \cdot Y = X$
$X \cdot Y = Y \cdot X$	Teoremas de de Morgan:
Lei associativa:	$\overline{(X + Y)} = \overline{X} \cdot \overline{Y}$
$X + Y + Z = X + (Y + Z)$	$\overline{(X \cdot Y)} = \overline{X} + \overline{Y}$
$X + Y + Z = (X + Y) + Z$	Teoremas da consistência:
$X \cdot Y \cdot Z = X \cdot (Y \cdot Z)$	$X \cdot Y + X \cdot \overline{Y} = X$
$X \cdot Y \cdot Z = (X \cdot Y) \cdot Z$	$(X + Y) \cdot (X + \overline{Y}) = X$
Lei distributiva:	Teoremas da inclusão:
$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$	$X \cdot \overline{X} = 0$
$(X + Y) \cdot (Z + W) = X \cdot Z + X \cdot W + Y \cdot Z + Y \cdot W$	$X + \overline{X} = 1$

matriz ( $X3 = 1$  para removida,  $X3 = 0$  para não removida). Por fim, seja  $Y$  o valor que determina se a sequência de carregamento pode ser iniciada ( $Y = 1$  para iniciar,  $Y = 0$  para aguardar).

A expressão da álgebra booleana é  $Y = X1 \cdot X2 \cdot X3$ .

As três condições devem ser satisfeitas, portanto, utiliza-se a porta lógica AND. Para que  $Y$  seja igual a 1 e a sequência de carregamento seja iniciada, todos os valores  $X1$ ,  $X2$  e  $X3$  devem ser iguais a 1. O diagrama lógico para essa condição de intertravamento é apresentado na Figura 9.5.

#### EXEMPLO 9.2

##### Interruptor de pressão

Um interruptor de pressão utilizado para iniciar e parar motores e outros dispositivos elétricos é um componente de *hardware* comum nos sistemas de controle industrial.

Conforme mostrado na Figura 9.6(a), o interruptor é formado por uma caixa com dois botões, um para INICIAR e outro para PARAR. Quando o botão INICIAR é momentaneamente pressionado por um operador humano, energia é fornecida ao motor (ou outra carga) e lá mantida até que o botão PARAR seja pressionado.

<sup>1</sup> ANSI — American National Standard Institute (Instituto Nacional Americano de Padrões)

<sup>2</sup> ISO — International Organization for Standardization (Organização Internacional para Padronização)

Figura 9.5 Diagrama lógico para a máquina a ser carregada por um robô com o sistema de intertravamento descrito no Exemplo 9.1

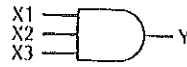
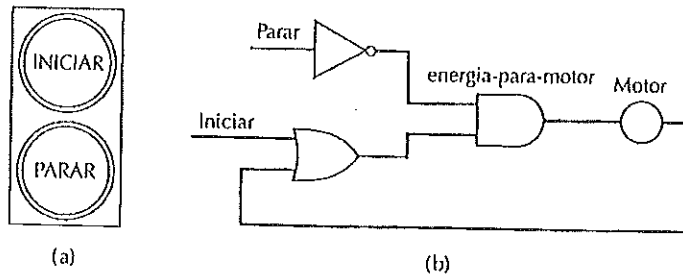


Figura 9.6 (a) Interruptor de pressão do Exemplo 9.2 e (b) seu diagrama lógico



“Energia-para-motor” é a saída do interruptor de pressão. O valor das variáveis pode ser definido da seguinte maneira:

INICIAR = 0 normalmente indica o estado contato aberto. INICIAR = 1 quando o botão INICIAR está pressionado para fazer contato.

PARAR = 0 normalmente indica o estado contato fechado. PARAR = 1 quando o botão PARAR está pressionado para parar o contato.

ENERGIA-PARA-MOTOR = 0 quando os contatos estão abertos.

ENERGIA-PARA-MOTOR = 1 quando os contatos estão fechados.

MOTOR = 0 quando desligado (não está em funcionamento). MOTOR = 1 quando ligado.

A tabela-verdade para o interruptor de pressão é apresentada na Tabela 9.7. A partir da condição inicial de desligado (MOTOR = 0), o motor é iniciado com o pressionamento do botão para iniciar (INICIAR = 1). Se o botão de parar estiver em sua condição normal de fechado (PARAR = 0), energia será fornecida ao motor (ENERGIA-PARA-MOTOR = 1). Enquanto o motor estiver funcionando (MOTOR = 1), ele pode ser parado por meio do pressionamento do botão de parar (PARAR = 1). O diagrama lógico correspondente é mostrado na Figura 9.6(b).

Tabela 9.7 Tabela-verdade para o interruptor de pressão do Exemplo 9.2

Iniciar	Parar	Motor	Energia para motor
0	0	0	0
0	1	0	0

1	0	0	1
1	1	0	0
0	0	1	1
0	1	1	0
1	0	1	1
1	1	1	0

De certa maneira, o interruptor de pressão do Exemplo 9.2 vai um pouco além de nossa definição de um sistema lógico puro, pois exibe características de memória. As variáveis MOTOR e ENERGIA-PARA-MOTOR são quase o mesmo sinal. As condições que determinam se a energia fluirá pelo motor são diferentes conforme o estado do motor (LIGADO/DESLIGADO). Compare as quatro primeiras linhas da tabela-verdade com as quatro últimas (Tabela 9.7). É como se o controle lógico precisasse lembrar se o motor está ligado ou desligado para decidir quais condições determinarão o valor do sinal de saída. Essa característica de memória é exibida pela malha de realimentação (parte inferior) no diagrama lógico da Figura 9.6(b).

### 9.1.2 Sequenciamento

Um sistema de sequenciamento utiliza dispositivos temporizadores internos para determinar quando iniciar as alterações nas variáveis de saída. Máquinas de lavar, secadoras, lavadoras de louça e aparelhos similares utilizam sistemas de sequenciamento para cronometrar o início e o fim dos elementos do ciclo. Existem muitas aplicações industriais de sistemas de sequenciamento.

Por exemplo, imagine que uma bobina de aquecimento por indução seja utilizada para aquecer uma peça em nosso exemplo anterior sobre uma operação de forjamento com a ajuda de um robô. Em vez de usar um sensor de temperatura, a bobina poderia ser configurada com um ciclo de aquecimento temporizado, de modo que energia suficiente fosse fornecida para aquecer a peça na temperatura desejada. O processo de aquecimento é suficientemente confiável e previsível e sabe-se que determinado tempo de energização da bobina de indução aquecerá de forma consistente a peça a uma determinada temperatura (com variação mínima).

Muitas aplicações da automação industrial exigem que o controlador ofereça um conjunto predefinido de valores LIGADOS/DESLIGADOS para as variáveis de saída. As saídas costumam ser geradas de uma única vez, o que significa que não existe verificação se a função de controle foi mesmo executada. Outra característica que exemplifica esse modo de controle é que a sequência de sinais de saída costuma ser cíclica: os sinais ocorrem no mesmo padrão repetido dentro de um ciclo regular. Temporizadores e contadores ilustram esse tipo de componente de controle.

Um *temporizador* é um dispositivo que alterna sua saída entre LIGADO ou DESLIGADO em determinados intervalos de tempo. Temporizadores comuns utilizados na indústria e em residências ligam quando são ativados e permanecem nesse estado durante um intervalo programado de tempo. Alguns temporizadores são ativados por meio do pressionamento de um botão para iniciar, como, por exemplo, os controles da bomba de água em uma banheira de hidromassagem. Outros operam com base em um relógio 24 horas. Por exemplo, alguns sistemas domésticos de segurança possuem características de temporização que ligam e apagam luzes durante o dia, dando a impressão de que há pessoas na casa.

Dois tipos adicionais de temporizadores são utilizados em sistemas de controle discreto: (1) temporizadores com atraso no desligamento (*delay-off timer*) e (2) temporizadores com atraso no acionamento (*delay-on timer*). Um *temporizador com atraso no desligamento* liga imediatamente a energia em resposta a um sinal para iniciar, e então a desliga após um tempo específico. Muitos carros estão equipados com esse tipo de dispositivo. Quando você sai do veículo, as luzes permanecem acesas durante determinado intervalo de tempo (por exemplo, 30 segundos), e então se apagam automaticamente. Um *temporizador com atraso no acionamento* espera um determinado intervalo de tempo antes de ligar a energia quando recebe um sinal para iniciar. Para programar um temporizador, o usuário deve especificar o tempo do intervalo de espera.

Um *contador* é um componente utilizado para contar pulsos elétricos e armazenar os resultados do procedimento de contagem (Seção 6.5.2). Os conteúdos instantâneos podem ser mostrados e/ou utilizados no algoritmo de controle do processo. Os contadores podem ser classificados como progressivos, regressivos, ou progressivos/regressivos. Um *contador progressivo* começa no zero e incrementa seu conteúdo (contagem total) em 1 como resposta a cada pulso. Quando determinado valor for alcançado, o contador progressivo voltará ao zero. Uma aplicação desse dispositivo é a contagem do número de garrafas de cervejas cheias em movimento sobre um transportador para encaixotamento. A cada conjunto de 24 garrafas, tem-se um engradado, e o contador retorna ao zero. Um *contador regressivo* inicia com determinado valor e decresce o total em 1 para cada pulso recebido. Ele poderia ser usado na mesma aplicação das garrafas, utilizando-se o valor inicial igual a 24. Um *contador progressivo/regressivo* combina as duas operações de contagem e pode ser útil no acompanhamento do número de garrafas restantes no *buffer* para armazenamento. Ele aumenta o número de garrafas entrando no *buffer* e diminui o número de garrafas saindo do *buffer* para obter um cálculo atual do conteúdo do *buffer*.

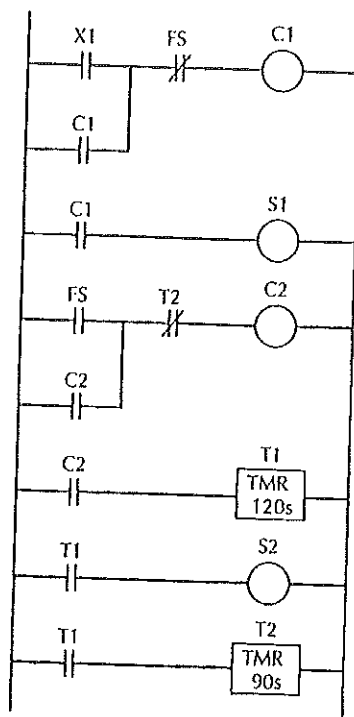
## 9.2 DIAGRAMAS DE LÓGICA LADDER

Os diagramas lógicos mostrados nas figuras 9.5 e 9.6(b) são úteis à exibição das relações entre os elementos lógicos. Outra técnica de diagramação que exibe a lógica e, de certo modo, o tempo e o sequenciamento do sistema é o diagrama de lógica ladder. O método gráfico apresenta uma virtude importante, pois é análogo aos circuitos elétricos usados para realizar a lógica e o controle da sequência. Além disso, os diagramas de lógica ladder são familiares ao pessoal de fábrica que deve construir, testar, manter e reparar os sistemas de controle discreto.

Em um diagrama de lógica ladder, vários elementos lógicos e outros componentes são dispostos ao longo de linhas horizontais ou degraus conectados em ambas as extremidades a dois trilhos verticais, conforme ilustrado na Figura 9.7. O diagrama tem aparência geral de escada (do inglês, *ladder*) e daí vem o nome. Os elementos e componentes são *contatos* (representando entradas lógicas) e *cargas*, também conhecidas como *bobinas* (representando saídas). As entradas incluem interruptores e contatos de relés, e as cargas são motores, lâmpadas e alarmes. A energia (por exemplo, 115 V de corrente alternada) para o componente é fornecida por dois trilhos verticais. Nos diagramas ladder, é comum posicionar as entradas à esquerda de cada degrau e as saídas à direita.



Figura 9.7 Diagrama de lógica ladder



Os símbolos usados nos diagramas ladder para os componentes lógicos e de sequenciamento comuns são apresentados na Figura 9.8. Existem dois tipos de contatos: normalmente abertos e normalmente fechados. Um *contato normalmente aberto* permanece aberto até que seja ativado. Quando ativado, ele fecha para permitir a passagem de corrente. Um *contato normalmente fechado* permanece fechado, permitindo que a corrente flua até que seja ativa-

do. Quando ativado, ele abre e impede, assim, o fluxo de corrente. Contatos normalmente abertos de um interruptor ou dispositivo semelhante são simbolizados por duas linhas verticais curtas ao longo de um degrau horizontal da escada, como mostra a Figura 9.8(a). Contatos normalmente fechados são mostrados nas mesmas linhas diagonais, mas com uma linha vertical sobre eles, como mostra a Figura 9.8(b). Ambos os tipos de contato são usados para re-

Figura 9.8 Símbolos para os elementos lógicos e de sequenciamento comuns em diagramas de lógica ladder

Símbolo ladder	Componente de hardware
(a)	Contatos normalmente abertos (interruptor, relé, outros dispositivos ON/OFF)
(b)	Contatos normalmente fechados (interruptor, relé etc.)
(c)	Cargas de saída (motor, lâmpada, solenoide, alarme etc.)
(d)	Temporizador
(e)	Contador

presentar entradas no circuito lógico do tipo LIGADO/DESLIGADO. Além dos interruptores, entradas incluem relés, sensores do tipo ligado/desligado (por exemplo, interruptores de fim de curso e fotodetectores), temporizadores e outros dispositivos binários de contato.

Cargas de saída tais como motores, luzes, alarmes, solenoides e outros componentes elétricos que são ligados e desligados pelo sistema de controle lógico são mostrados como círculos, como se vê na Figura 9.8(c). Temporizadores e contadores são simbolizados por quadrados (ou retângulos) com entradas e saídas apropriadas para o acionamento correto do dispositivo, conforme mostram as figuras 9.8(d) e (e). O temporizador simples requer a especificação do tempo de espera e a identificação do contato de entrada que ativa a espera. Quando o sinal de entrada é recebido, o temporizador aguarda o tempo de espera antes de ligar ou desligar o sinal de saída. O temporizador é reiniciado (volta ao valor inicial) quando o sinal de entrada é desligado.

Os contadores requerem duas entradas. A primeira é o trem de pulsos (séries de sinais ligado/desligado) que é contado pelo contador. A segunda é o sinal para reiniciar o contador e recomençar o procedimento de contagem. Reiniciar o contador significa zerar a contagem em um contador progressivo e definir um valor inicial em um contador regressivo. A contagem acumulada é mantida na memória para uso, se demandado pela aplicação.

#### EXEMPLO 9.3

##### Três simples circuitos com lâmpadas

Três portas lógicas básicas (AND, OR e NOT) podem ser simbolizadas em diagramas de lógica ladder. Crie diagramas para os três circuitos com lâmpadas ilustrados nas figuras 9.1, 9.2 e 9.3.

**Solução:** Os três diagramas ladder correspondentes a esses circuitos são apresentados nas figuras 9.9(a)-(c). Observe a semelhança entre os diagramas de circuito originais e os diagramas ladder mostrados aqui. Veja que o símbolo NOT é o mesmo que o do contato normalmente fechado, que é o inverso lógico de um contato normalmente aberto.

#### EXEMPLO 9.4

##### Interruptor de pressão

A operação do interruptor de pressão do Exemplo 9.2 pode ser representada em um diagrama de lógica ladder. A partir da Figura 9.6, vamos considerar que o INICIAR será representado por X1; PARAR, por X2; e MOTOR, por Y. Vamos criar o diagrama.

**Solução:** O diagrama ladder é apresentado na Figura 9.10. X1 e X2 são contatos de entrada, e Y é uma carga no diagrama. Observe como Y também serve como contato de entrada para oferecer a conexão ENERGIA-PARA-MOTOR.

#### EXEMPLO 9.5

##### Relé de controle

A operação de um relé de controle pode ser demonstrada por meio do diagrama de lógica ladder apresentado na Figura 9.11. Um relé pode ser utilizado para controlar a atuação ligado/desligado de um dispositivo elétrico em alguma localização remota. Também pode ser usado para definir decisões alternativas no controle lógico. Nosso diagrama ilustra ambos os usos. O relé é indicado pela carga C (para relé de controle), que controla a operação ligado/desligado de dois motores (ou outros tipos de cargas de saída) Y1 e Y2. Quando o interruptor de controle X está aberto, o relé fica sem energia e, assim, conecta a carga Y1 à linha de alimentação. Na verdade, o interruptor X aberto liga o motor Y1 por meio do relé.

Figura 9.9 Três diagramas de lógica ladder para os circuitos de lâmpada das figuras (a) 9.1, (b) 9.2 e (c) 9.3

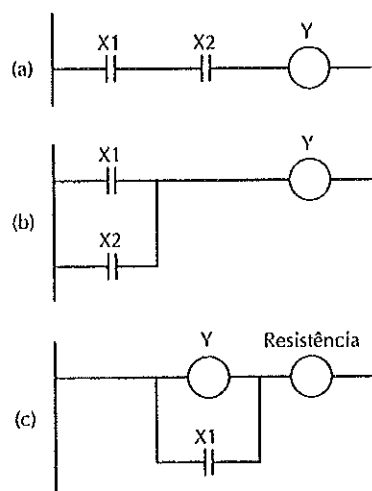
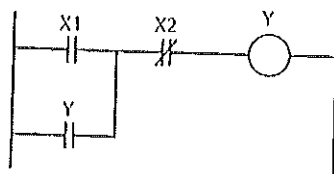


Figura 9.10 Diagrama de lógica ladder para o interruptor de pressão do Exemplo 9.4

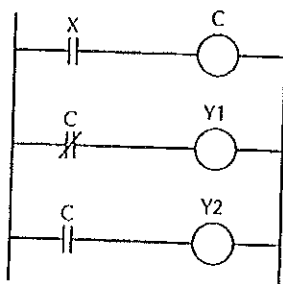


Quando o interruptor de controle está fechado, o relé fica energizado. Isso abre o contato normalmente aberto do segundo degrau da escada e fecha o contato normalmente aberto do terceiro degrau. Na verdade, a energia é desligada para a carga Y1 e ligada para a carga Y2.

O Exemplo 9.5 ilustra diversas características importantes de um diagrama de lógica ladder. Primeiro, a mesma entrada pode ser utilizada mais de uma vez no diagrama. No exemplo, o contato do relé C foi utilizado como entrada tanto no segundo como no terceiro degrau do diagrama. Como vamos ver na próxima seção, essa característica de utilização de um dado relé de contato em diversos degraus dife-

rentes do diagrama ladder para servir a múltiplas funções lógicas oferece uma vantagem substancial para o controlador programável, se comparado às unidades de controle com relés reais conectados entre si. Com relés reais, contatos separados teriam de ser construídos no controlador para cada função lógica. Uma segunda característica do Exemplo 9.5 é a possibilidade de que uma saída (carga) em um degrau do diagrama seja uma entrada (contato) em outro degrau. O relé C foi a saída do degrau superior na Figura 9.11, mas tal saída foi utilizada como entrada em outra parte do diagrama. Essa mesma característica foi apresentada no diagrama do interruptor de pressão do Exemplo 9.4.

Figura 9.11 Diagrama de lógica ladder para o relé de controle no Exemplo 9.5



#### EXEMPLO 9.6

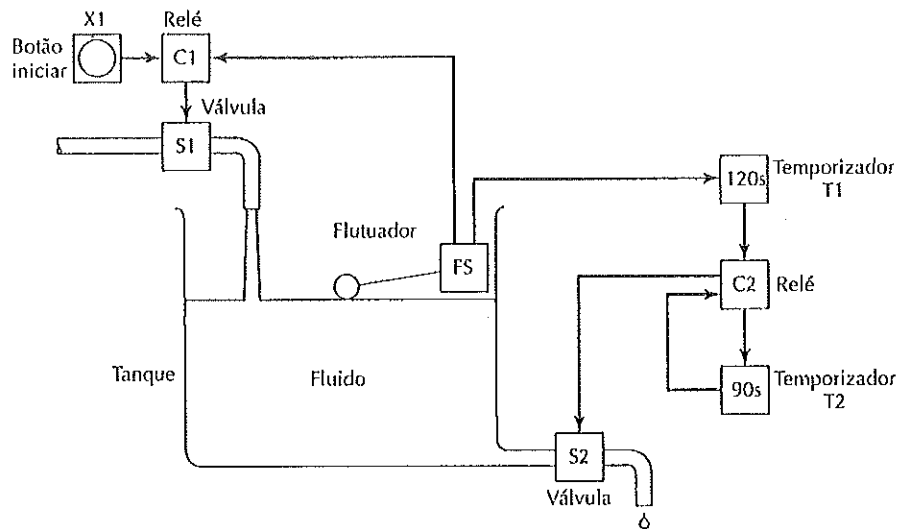
Considere o tanque de armazenamento de fluido ilustrado na Figura 9.12. Quando o botão para iniciar X1 é pressionado, o relé de controle C1 é energizado. Por sua vez, isso energiza o solenoide S1, que abre uma válvula que permite que o fluido passe para o tanque. Quando o tanque enche, o interruptor de boia FS se fecha, o que abre o relé C1 e faz com que o solenoide S1 deixe de receber energia e, assim, desligue o fluxo de entrada. O interruptor de boia FS também ativa o temporizador T1, que fornece um intervalo de 120 segundos para que determinada reação química aconteça no tanque. No fim do intervalo, o temporizador energiza um segundo relé C2, que controla dois dispositivos: (1) ele energiza o solenoide S2, que abre uma válvula que permite que o fluido saia do tanque; (2) ele inicia o temporizador T2, que aguarda 90 segundos para permitir que o conteúdo do tanque seja drenado. No fim dos 90 segundos, o temporizador interrompe a corrente e para de

energizar o solenoide S2, fechando assim a válvula de escoamento. O pressionamento do botão de iniciar X1 reinicia os temporizadores e abre seus respectivos contatos. Construa o diagrama de lógica ladder para o sistema.

**Solução:** O diagrama de lógica ladder é construído conforme se vê na Figura 9.7.

O diagrama de lógica ladder é uma excelente maneira de representar problemas de controle da lógica combinacional nos quais as variáveis de saída baseiam-se diretamente nos valores das entradas. Conforme indicado no Exemplo 9.6, ele também pode ser utilizado para demonstrar problemas de controle sequencial (temporizadores), embora, para esse propósito, o diagrama seja um tanto mais difícil de interpretar e analisar. O diagrama de lógica ladder é a principal técnica para configuração dos programas de controle nos controladores lógicos programáveis.

Figura 9.12 Operação de enchimento com fluido do Exemplo 9.6



### 9.3 CONTROLADORES LÓGICOS PROGRAMÁVEIS (CLPS)

Um controlador lógico programável (CLP) pode ser definido como baseado em microcomputador que usa instruções armazenadas em uma memória programável para implementar lógica, sequenciamento, temporização, contagem e funções aritméticas por meio de módulos de entrada/saída (E/S) digitais ou analógicos para controle de máquinas e processos. Aplicações de CLP são encontradas tanto em processos industriais como em manufatura discreta. Exemplos de aplicações nas indústrias de processos incluem processamento químico, operações em fábricas

de papel e produção alimentícia. CLPs estão primeiramente associados às indústrias de manufatura discreta para o controle de máquinas individuais, células de máquina, linhas de transferência, equipamentos para manuseio de materiais e sistemas automatizados de armazenamento. Antes do CLP ser lançado, por volta de 1970, controladores compostos por relés, bobinas, contadores, temporizadores e componentes semelhantes eram utilizados na implementação desse tipo de controle industrial (Nota histórica 9.1). Hoje, muitos equipamentos antigos estão sendo adaptados aos CLPs para substituir os controladores originais, em geral tornando o equipamento mais produtivo e confiável do que quando novo.

#### Nota histórica 9.1

#### Controladores lógicos programáveis [2], [5], [7], [8]

Em meados da década de 1960, Richard Morley era um dos sócios da Bedford Associates, empresa de consultoria de New England, especializada em sistemas de controle para empresas de máquinas-ferramenta. A maior parte do trabalho da empresa envolvia a substituição de relés por minicomputadores nos controles das máquinas-ferramenta. Em janeiro de 1968, Morley idealizou o primeiro controlador lógico programável e redigiu suas especificações.<sup>3</sup> Ele acabaria com algumas das limitações dos computado-

<sup>3</sup> Morley usou a abreviação PC para referir-se ao controlador programável. Esse termo foi utilizado por muitos anos até que a IBM começou a chamar seus computadores pessoais pela mesma sigla, no início da década de 1980. O termo CLP, amplamente utilizado hoje para referir-se ao controlador lógico programável, foi criado por Allen-Bradley, um importante fornecedor de CLPs.

res convencionais usados no controle de processos da época; especificamente, ele seria um processador em tempo real (Seção 5.3.1), seria previsível e confiável; modular e resistente. A programação seria baseada na lógica ladder, largamente utilizada para controles industriais. O controlador que surgia era chamado de Modicon Modelo 084. A palavra *modicon* era a abreviação de controlador digital modular (do inglês, *modular digital controller*). A nomenclatura do modelo 084 foi cunhada a partir do fato de o controlador ser o 84º produto criado pela Bedford Associates. Morley e seus associados decidiram criar uma nova empresa para a produção de comutadores, e o Modicon foi incorporado em outubro de 1968. Em 1977, foi vendido para a Gould e tornou-se a divisão de CLPs da empresa.

No mesmo ano em que Morley inventou o CLP, a Divisão Hidráulica da General Motors Corporation desenvolveu um conjunto de especificações para um CLP. Essas especificações eram motivadas pelo alto custo e pela falta de flexibilidade dos controladores eletromecânicos baseados em relés utilizados largamente na indústria automotiva para controlar linhas de transferência e outros sistemas mecânicos e automatizados. Os requisitos para o dispositivo eram que devia (1) ser programável e reprogramável, (2) ser projetado para operar em ambiente industrial, (3) aceitar sinais de 120 V ac enviados por interruptores-padrão de pressão e de fim de curso, (4) dispor de saídas projetadas para comutar e operar continuamente cargas como motores e relés de classificação 2-A e (5) possuir preço e custo de instalação competitivos se comparados aos dos relés e dos dispositivos lógicos de estado sólido usados na época. Além da Modicon, diversas outras empresas viram uma oportunidade comercial nas especificações da GM e desenvolveram versões de CLP.

As características dos primeiros CLPs eram semelhantes às dos controles por relés que substituíram. Eram limitados a controles do tipo ligado/desligado. Em cinco anos, as melhorias no produto incluíram melhores interfaces de operação, capacidade aritmética, manipulação de dados e comunicação com computadores. Ao longo dos cinco anos seguintes, as melhorias incluíram aumento da memória; controle analógico e de posicionamento e E/S remota (permitindo que dispositivos remotos fossem conectados a um subsistema de E/S-satélite multiplexado para o CLP usando par trançado). Grande parte do progresso baseava-se nos avanços na área de tecnologia de microprocessadores. Em meados da década de 1980, foi lançado o micro-CLP, um CLP de pequeno porte em tamanho (tamanho típico = 75 mm x 75 mm x 125 mm) e custo bastante inferior (menos de US\$ 500). Em meados da década de 1990, chega o nano-CLP, ainda menor e mais barato.

Existem vantagens significativas no uso do CLP em detrimento dos relés, temporizadores, contadores e outros componentes de controle convencionais. Essas vantagens incluem: (1) programar o CLP é mais fácil do que cabear o painel de controle do relé; (2) o CLP pode ser reprogramado, enquanto os controles convencionais devem ser recabeados e costumam ser destruídos nesse procedimento; (3) os CLPs ocupam menos espaço do que os painéis de controle de relés; (4) a confiabilidade é maior e a manutenção, mais fácil; (5) o CLP pode ser mais facilmente conectado aos sistemas de computador do que os relés; e (6) os CLPs podem executar uma variedade maior de funções de controle do que as dos controles de relés.

Nesta seção, descrevemos os componentes, a programação e a operação do CLP. Embora suas principais aplicações sejam no controle lógico e no sequenciamento (controle discreto), muitos CLPs também executam funções adicionais, descritas posteriormente nesta seção.

### 9.3.1 Componentes dos CLP

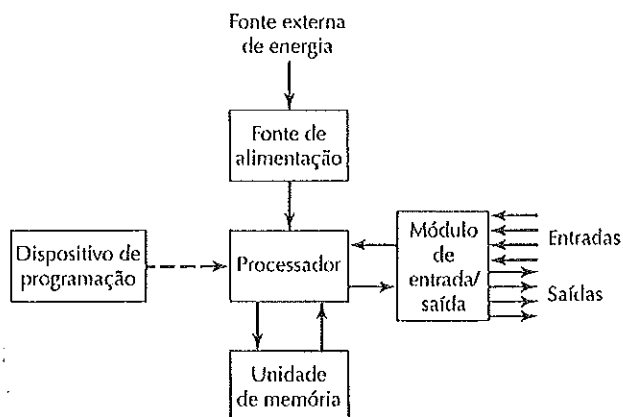
Um diagrama esquemático de um CLP é apresentado na Figura 9.13. Os componentes básicos do CLP são

os seguintes: (1) processador, (2) unidade de memória, (3) fonte de energia, (4) módulo de E/S e (5) dispositivo de programação. Esses componentes ficam abrigados em um espaço apropriado projetado para suportar o ambiente industrial.

O *processador* é a unidade central de processamento (do inglês, *central processing unit* — UCP ou CPU) do controlador programável. Com vistas a determinar os sinais de saída apropriados, ele executa várias funções lógicas e de sequenciamento por meio da manipulação das entradas do CLP. O ciclo típico de operação da UCP é descrito na Seção 9.3.2. A UCP consiste de um ou mais microprocessadores semelhantes aos utilizados nos PCs e em outros equipamentos de processamento de dados. A diferença é que possuem sistema operacional em tempo real e são programados de modo a facilitar as transações de E/S e a executar a função de lógica ladder. Além disso, os CLPs são robustos para que a UCP e outros componentes eletrônicos possam operar no ambiente eletricamente ruidoso da fábrica.

A *unidade de memória* está conectada à UCP. Contém os programas de lógica, sequenciamento e operações de E/S. Também mantém arquivos de dados associados a

Figura 9.13 Componentes de um CLP



esses programas, inclusive bits de estado de E/S, constantes de contadores e temporizadores, e valores de outros parâmetros e variáveis. Essa unidade de memória é tratada como memória do usuário ou da aplicação, pois seu conteúdo é informado pelo usuário. Além disso, o processador também contém a memória do sistema operacional, que direciona a execução do programa de controle e coordena as operações de E/S. O sistema operacional é gravado pelo fabricante e não pode ser acessado ou alterado pelo usuário.

Uma *fonte de alimentação* de 115 V ac costuma ser utilizada para acionar o CLP (algumas unidades operam em 230 V ac). A fonte de energia converte os 115 V ac em tensões de corrente contínua (cc) de  $\pm 5$  V. Essas baixas voltagens são usadas na operação de equipamentos que podem ter voltagem e potência muito superiores às do CLP. A fonte de energia costuma incluir uma bateria reserva que é acionada automaticamente no caso de falha externa de energia.

O *módulo de entrada/saída* oferece as conexões com os equipamentos ou os processos industriais a ser controlados. As entradas para o controlador são sinais de interruptores de fim de curso ou de pressão, sensores e outros dispositivos do tipo ligado/desligado. As saídas do controlador são sinais de ligado/desligado para operar motores, válvulas e outros dispositivos necessários à ativação do processo. Além disso, muitos CLPs são capazes de aceitar sinais contínuos oriundos de sensores analógicos e de gerar sinais adequados aos atuadores analógicos. O tamanho de um CLP costuma ser medido em termos do número de terminais de E/S, conforme indicado na Tabela 9.8.

O CLP é programado por meio do dispositivo de programação, que geralmente pode ser desacoplado do compartimento do CLP de modo a ser compartilhado entre muitos controladores. Diferentes fabricantes de CLP ofe-

recem diferentes dispositivos, variando de painéis de controle simples (do inglês, *teach-pendant*) semelhantes aos utilizados na robótica, a teclados e telas especiais de programação de CLPs. Um PC utilizado para esse fim algumas vezes permanece conectado ao CLP para servir como monitoramento de processos ou com função de supervisão e para aplicações convencionais de processamento de dados relacionados ao processo.

### 9.3.2 Ciclo de operação do CLP

Até onde sabe o usuário do CLP, as etapas no programa de controle são executadas simultânea e continuamente. Na verdade, é necessário um determinado tempo para que o processador do CLP execute o programa do usuário durante um ciclo de operação. O ciclo de operação típico do CLP, chamado de *varredura* (*scan*), é composto de três etapas: (1) varredura de entrada, (2) varredura do programa e (3) varredura de saída. Durante a *varredura de entrada*, as entradas do CLP são lidas pelo processador e o estado dessas entradas é armazenado na memória. Em seguida, o programa de controle é executado durante a *varredura do programa*. Os valores de entrada armazenados na memória são utilizados nos cálculos da lógica de controle para determinar os valores das saídas. Por fim, durante a *varredura de saída*, as saídas são atualizadas conforme os valores calculados. O tempo para execução da varredura é chamado de *tempo de varredura* e depende do número de entradas que devem ser lidas, da complexidade das funções que devem ser executadas e do número de saídas que devem ser alteradas. O tempo de varredura também depende da velocidade do relógio (*clock*) do processador e costuma variar entre 1 e 25 ms [3].

Um dos problemas potenciais que podem ocorrer durante o ciclo de varredura é que o valor de entrada

Tabela 9.8 Classificação típica de CLPs pelo número de terminais de entrada/saída

Tamanho do CLP	Quantidade de E/S
CLP grande	$\geq 1024$
CLP médio	$< 1024$
CLP pequeno	$< 256$
CLP micro	$\leq 32$
CLP nano	$\leq 16$

pode mudar imediatamente depois de ter sido lido. Como o programa usa o valor de entrada armazenado na memória, quaisquer valores de saída dependentes dessa entrada serão incorretamente calculados. Existe, obviamente, um risco potencial envolvido nesse modo de operação. Entretanto, o risco é minimizado porque o tempo entre as atualizações é tão curto que é pouco provável que o valor de saída permaneça incorreto por um tempo capaz de causar efeitos sérios sobre a operação do processo. O risco se torna mais significativo em processos nos quais o tempo de resposta é muito rápido e nos quais danos podem ocorrer durante o tempo de varredura. Alguns CLPs possuem características especiais para a realização de atualizações "imediatas" de sinais de saída quando se sabe que as alterações nas frequências das variáveis de entrada são mais rápidas do que o tempo de varredura.

### 9.3.3 Capacidades adicionais do CLP

É possível que as funções de controle lógico e sequenciamento descritas na Seção 9.1 representem as principais operações de controle realizadas pelo CLP. Essas são as funções para as quais o controlador programável foi originalmente criado. Entretanto, o CLP evoluiu e incluiu diversas possibilidades que vão além do controle lógico e do sequenciamento. Algumas dessas novas possibilidades disponíveis em muitos CLPs comerciais são:

- *Controle analógico.* O controle proporcional-integral-derivativo (PID) está disponível em alguns controladores programáveis. Esses algoritmos de controle foram tradicionalmente criados utilizando controladores analógicos. Hoje em dia, os esquemas de controle analógico são aproximados pelo uso do computador digital, seja com um CLP ou um controle de processos por computador.
- *Funções aritméticas.* Adição, subtração, multiplicação e divisão. O uso das mesmas permite que sejam desenvolvidos algoritmos de controle mais complexos

do que os que eram possíveis com os elementos convencionais da lógica e do sequenciamento.

- *Funções de matriz.* Alguns CLPs possuem capacidade de executar funções de matriz sobre valores armazenados na memória. Esse recurso pode ser utilizado para comparar os valores atuais do conjunto de entradas e saídas com os valores armazenados na memória do CLP de modo a determinar se algum erro ocorreu.
- *Processamento e relatório de dados.* Essas funções estão tipicamente associadas a aplicações de negócios de PCs (*business applications*). Os fabricantes de CLPs julgaram necessário incluir essas características em seus controladores à medida que desaparecia a diferença entre PCs e CLPs.

### 9.3.4 Programando o CLP

A programação é o meio pelo qual o usuário informa as instruções de controle ao CLP, a partir do dispositivo de programação. As instruções de controle mais básicas são as de comutação, lógica, sequenciamento, contagem e temporização. Quase todos os métodos de programação de CLPs oferecem conjuntos de instruções que incluem essas funções. Muitas aplicações de controle demandam instruções adicionais para a realização do controle analógico de processos contínuos, lógica de controle complexa, processamento e relatório de dados, além de outras funções avançadas que não são prontamente realizadas pelo conjunto básico de instruções. Graças às diferenças nos requisitos, diversas linguagens de programação para CLPs foram desenvolvidas. Um padrão para a programação de CLPs foi publicado pela International Electromechanical Commission, em 1992, e intitulado *International Standard for Programmable Controllers* (IEC 1131-3). Esse padrão especifica três linguagens gráficas e duas textuais para programação de CLPs, respectivamente: (1) diagramas de lógica ladder, (2) diagramas de blocos de funções, (3) diagramas de funções sequenciais, (4) lista de instruções e (5) texto estruturado. A Tabela 9.9 lista as cinco linguagens e



Tabela 9.9: Características das cinco linguagens de CLP especificadas pelo padrão IEC1131-3

Linguagem.	Abreviação	Tipo	Aplicações em que melhor se enquadram
Diagrama de lógica ladder	(LD)	Gráfica	Controle discreto
Diagrama de blocos de funções	(FBD)	Gráfica	Controle contínuo
Diagrama de funções sequenciais	(SFC)	Gráfica	Sequenciamento
Lista de instruções	(IL)	Textual	Controle discreto
Texto estruturado	(ST)	Textual	Lógica complexa, cálculos etc.

as aplicações mais comuns para cada uma delas. O padrão IEC 1131-3 também define que as cinco linguagens devem ser capaz de interagir entre si de modo a permitir todos os níveis possíveis de sofisticação do controle em qualquer aplicação.

**Diagrama de lógica ladder.** A linguagem de programação de CLPs mais utilizada atualmente envolve diagramas ladder (do inglês, *ladder diagrams* — LDs), que mostramos em várias figuras ao longo da seção anterior. Conforme indicado na Seção 9.2, os diagramas ladder são muito convenientes para o pessoal de chão de fábrica, que está familiarizado com diagramas ladder e de circuitos, mas que pode não conhecer computadores e sua programação. Para utilizar diagramas de lógica ladder, eles não precisam aprender uma linguagem de programação completamente nova.

A entrada direta do diagrama de lógica ladder na memória do CLP requer o uso de um teclado e de um monitor com capacidades gráficas para exibição de símbolos representando os componentes e suas inter-relações no diagrama de lógica ladder. Os símbolos são semelhantes aos apresentados na Figura 9.8; o teclado do CLP costuma ser projetado com teclas para cada um dos símbolos individuais; a programação é feita por meio da inserção do componente apropriado nos degraus (linhas) do diagrama ladder; os componentes são de dois tipos básicos: contatos e bobinas, conforme descrito na Seção 9.2; os contatos representam interruptores de entrada, contatos de relés e elementos semelhantes; as bobinas representam cargas, como motores, solenoides, relés, temporizadores e contadores. Na verdade, o programador insere, degrau por degrau, o circuito no diagrama de lógica ladder na memória do CLP e o monitor vai exibindo os resultados após cada verificação.

**Diagrama de blocos de funções.** Um diagrama de bloco de função (do inglês, *function block diagram* — FBD) oferece meios para a inserção de instruções de alto

nível compostas por blocos operacionais. Cada bloco possui uma ou mais entradas e uma ou mais saídas. Dentro de um bloco, certas operações acontecem sobre as entradas de modo a transformar os sinais nas saídas desejadas. Os blocos de funções incluem operações como temporizadores e contadores, cálculos de controle utilizando equações (por exemplo, controle proporcional-integral-derivativo), manipulação de dados e transferência de dados para outros sistemas baseados em computador. Deixamos uma descrição mais apurada sobre esses diagramas de funções para outras referências, como Hughes [3] e os manuais de operação para os CLPs comerciais disponíveis.

**Diagrama de funções sequenciais.** O diagrama de funções sequenciais (do inglês, *sequential function chart* — SFC, também chamado de método *Grafset*) exhibe graficamente as funções sequenciais de um sistema automatizado como uma série de etapas e transições de um estado do sistema para o estado seguinte. O diagrama de funções sequenciais é descrito em Boucher [1]. Ele se tornou um método-padrão para a documentação do controle lógico e do sequenciamento em grande parte da Europa. Entretanto, seu uso nos Estados Unidos é mais limitado e, portanto, sugerimos ao leitor a referência citada como fonte de mais detalhes sobre o método.

**Lista de instruções.** A programação por lista de instruções (do inglês, *instruction list* — IL) também oferece um modo de inserir o diagrama de lógica ladder na memória do CLP. Nesse método, o programador utiliza uma linguagem de programação de baixo nível para construir o diagrama de lógica ladder por meio da entrada de declarações que especificam os vários componentes e suas relações para cada degrau do diagrama ladder. Vamos explicar essa abordagem por meio da introdução de um conjunto hipotético de instruções CLP. Nossa "linguagem" CLP contém diversas linguagens de diferentes fabricantes e contém menos características do que a maioria dos CLPs disponíveis comercialmente. Assumimos que o dispositivo



de programação é composto por um teclado adequado à entrada dos componentes individuais em cada degrau do diagrama de lógica ladder. Um monitor capaz de exibir os degraus do diagrama (e talvez vários dos degraus que o

precedem) ajuda na verificação do programa. O conjunto de instruções do nosso CLP é apresentado na Tabela 9.10 com uma breve explicação sobre cada instrução. Vamos examinar o uso desses comandos em alguns exemplos.

Tabela 9.10 Conjunto típico de instruções de baixo nível para um CLP

STR	Armazena uma nova entrada e inicia um novo degrau do diagrama ladder.
AND	Operação lógica AND com o elemento anteriormente informado. É interpretada como um circuito em série relativo a ele.
OR	Operação lógica OR com o elemento anteriormente informado. É interpretada como um circuito paralelo relativo a ele.
NOT	Operação lógica NOT ou inversão do elemento informado.
OUT	Elemento de saída do degrau do diagrama ladder.
TMR	Elemento temporizador. Requer sinal de entrada para iniciar a sequência de temporização. Em relação à entrada, a saída é atrasada durante um tempo especificado pelo programador, em segundos. A restauração do temporizador é realizada por meio da interrupção (parada) do sinal de entrada.
CTR	Elemento contador. Requer duas entradas: uma é o trem de pulsos de entrada contador pelo elemento CRT; a outra é o sinal indicando a reinicialização do procedimento de contagem.

#### EXEMPLO 9.7

Comandos da linguagem para circuitos AND, OR e NOT. Utilizando o conjunto de comandos da Tabela 9.10, escreva os programas CLP para os três diagramas ladder da Figura 9.10 representando os circuitos AND, OR e NOT das figuras 9.1, 9.2 e 9.3.

**Solução:** Os comandos para os três circuitos estão listados a seguir, com explicações.

Comando	Comentário
(a) STR X1	Armazena a entrada X1
AND X2	Entrada X2 em série com X1
OUT Y	Fornece Y como saída
(b) STR X1	Armazena a entrada X1
OR X2	Entrada X2 paralela a X1
OUT Y	Fornece Y como saída
(c) STR NOT X1	Armazena o inverso de X1
OUT Y	Fornece Y como saída

#### EXEMPLO 9.8

Comandos da linguagem para relé de controle

Utilizando o conjunto de comandos da Tabela 9.10, escreva um programa CLP para o controle de relé mostrado no diagrama de lógica ladder da Figura 9.11.

**Solução:** Os comandos para controle do relé estão listados a seguir, com explicações.

Comando	Comentário
STR X	Armazena a entrada X
OUT C	Saída do contato do relé C
STR NOT C	Armazena o inverso da saída C
OUT Y1	Saída da carga Y1
STR C	Armazena a saída C
OUT Y2	Saída da carga Y2

As linguagens de dois níveis costumam estar limitadas aos tipos de funções lógicas e de sequenciamento que podem ser definidas em um diagrama de lógica ladder. Embora temporizadores e contadores não tenham sido apresentados nos dois exemplos anteriores, alguns dos exercícios no fim do capítulo vão demandar o uso dos mesmos.

**Texto estruturado.** O texto estruturado (do inglês, *structured text* — ST) é uma linguagem de alto nível semelhante à dos computadores, que pode, no futuro, tornar-se a forma mais comum de programação de CLPs e PCs para aplicações de controle e automação. A principal vantagem de uma linguagem de alto nível é a capacidade de executar processamento de dados e cálculos sobre valores que não sejam binários. Os diagramas ladder e as linguagens CLP de baixo nível costumam ser limitados em sua habilidade de operar sinais que não sejam do tipo ligado/desligado. A capacidade de executar pro-

cessamento de dados e cálculos permite o uso de algoritmos de controle mais complexos, a comunicação com outros sistemas baseados em computador, a exibição de dados em um monitor e a entrada de dados por meio de um operador humano. Outra vantagem é a relativa facilidade com que um programa de controle complicado pode ser interpretado por um usuário. Comentários explicativos podem ser inseridos no programa para facilitar a interpretação.

## 9.4 COMPUTADORES PESSOAIS UTILIZANDO LÓGICA *SOFT*

No início da década de 1990, os PCs começaram a invadir aplicações anteriormente dominadas pelos CLPs. Antigamente, recomendava-se CLPs para uso em fábricas porque eram projetados para operar em ambientes adversos, enquanto os PCs eram projetados para o uso em escritórios. Além disso, com sua interface E/S embutida e seus sistemas operacionais em tempo real, os CLPs poderiam ser prontamente conectados a um equipamento externo para processos de controle, enquanto os PCs demandavam cartões de E/S e programas especiais que viabilizassem essas funções. Por fim, os computadores pessoais às vezes travavam sem nenhuma causa aparente — em geral, paradas dessa natureza não podiam ser toleradas em aplicações de controle industrial. Os CLPs não estão propensos a esse tipo de mau funcionamento.

A despeito dessas vantagens do CLP, a evolução tecnológica dos controladores lógicos programáveis não acompanhou o desenvolvimento dos computadores pessoais, e novas gerações de PCs são lançadas com uma frequência muito maior do que a dos CLPs. Existem muito mais programas e arquiteturas proprietárias no CLP do que no PC, o que dificulta a mescla e a combinação de componentes de diferentes fornecedores. Ao longo do tempo, esses fatores resultaram na desvantagem do desempenho dos CLPs. O desempenho deles fica em torno de dois anos atrás do dos PCs, e a diferença continua a aumentar. As velocidades dos PCs estão dobrando a cada 18 meses, em média, enquanto as melhorias na tecnologia dos CLPs ocorrem muito mais lentamente e requerem que as empresas refaçam o projeto de seus programas e arquiteturas proprietárias a cada nova geração de microprocessadores.

Os PCs agora estão disponíveis em gabinetes mais robustos para o ambiente de fábrica sujo e ruidoso. Podem ser equipados com teclados do tipo membrana para proteção contra humidade, óleo e sujeira e ser adquiridos com placas de E/S e equipamentos que ofereçam os dispositivos necessários à conexão com os equipamentos e processos das fábricas. Os sistemas operacionais projetados para a implementação de aplicações de controle em tempo real podem ser instalados em conjunto com outros programas tradicionais de escritório. Os fabricantes de CLPs estão reagindo ao desafio imposto pelos PCs com a inclusão de componentes e características dos computadores em seus produtos controladores de modo a distingui-los dos CLPs convencionais. Ainda assim, o futuro está propenso a ver um número crescente de PCs sendo utilizados nas aplicações de controle de fábricas nas quais os CLPs costumavam ser utilizados.

Duas abordagens básicas são empregadas nos sistemas de controle baseados em PCs [10]: lógica *soft* e sistemas críticos de controle em tempo real. Na configuração da lógica *soft*, o sistema operacional do PC é o Windows e algoritmos de controle são instalados como programas de alta prioridade. Entretanto, é possível interromper as tarefas de controle para atender a determinadas funções do sistema no Windows, tais como comunicações de rede e acesso ao disco. Quando isso acontece, a função de controle aguarda, o que pode trazer consequências negativas ao processo. Assim, um sistema de controle com lógica *soft* não pode ser considerado um controlador em tempo real no sentido do CLP. Em aplicações de controle de alta velocidade ou processos voláteis, a falta do controle em tempo real é um perigo potencial. Em processos menos críticos, a lógica *soft* funciona bem.

Nos sistemas críticos de controle em tempo real, o sistema operacional do PC é o que opera em tempo real, e o *software* de controle tem prioridade sobre todos os outros programas. As tarefas do Windows são executadas com prioridade baixa em relação ao sistema operacional em tempo real. O Windows não pode interromper a execução do controlador em tempo real. Se o Windows travar, não há alterações nas operações do controlador. Além disso, o sistema operacional reside na memória ativa do PC e, portanto, uma falha no disco rígido não tem efeito algum sobre o sistema crítico de controle em tempo real.

- 9.4 Construa o diagrama de lógica ladder para (a) uma porta NAND e (b) uma porta NOR.
- 9.5 Construa os diagramas de lógica ladder para as seguintes expressões lógicas booleanas:  
 (a)  $Y = (X1 + X2) \cdot X3$ , (b)  $Y = (X1 + X2) \cdot (X3 + X4)$ ,  
 (c)  $Y = (X1 \cdot X2) + X3$ .
- 9.6 Utilizando o conjunto de instruções da Tabela 9.10, escreva as declarações de uma linguagem de baixo nível para o sistema de intertravamento do robô no Exemplo 9.1.
- 9.7 Utilizando o conjunto de instruções da Tabela 9.10, escreva as declarações de uma linguagem de baixo nível para a lâmpada e para o sistema do fotodetector no Problema 9.4.
- 9.8 Utilizando o conjunto de instruções da Tabela 9.10, escreva as declarações de uma linguagem de baixo nível para a operação de enchimento com fluido do Exemplo 9.6.
- 9.9 Utilizando o conjunto de instruções da Tabela 9.10, escreva as declarações de uma linguagem de baixo nível para as quatro partes do Problema 9.5.
- 9.10 Na operação de enchimento com fluido do Exemplo 9.6, imagine que um sensor (por exemplo, um interruptor de boia submersa) seja utilizado no lugar do temporizador T2 para determinar se o conteúdo do tanque foi evacuado. (a) Construa um diagrama de lógica ladder para esse novo sistema. (b) Com base no conjunto de instruções de CLP da Tabela 9.10, escreva as declarações da linguagem de baixo nível para o sistema.
- 9.11 No manual de operação de uma prensa de estampagem de metal laminado, costuma-se utilizar um sistema de intertravamento de segurança com dois botões para prevenir que o operador inicie inadvertidamente a prensa enquanto seu braço está na matriz. Ambos os botões devem ser pressionados para que seja iniciado o ciclo de estampagem. Nesse sistema, um botão está localizado em um lado da prensa e o outro está no lado oposto. Durante o ciclo de operação, o operador coloca a peça na matriz e pressiona os dois botões, utilizando as duas mãos. (a) Escreva a tabela-verdade para esse sistema de intertravamento. (b) Escreva a expressão lógica booleana para o sistema. (c) Construa o diagrama lógico para o sistema. (d) Construa o diagrama de lógica ladder para o sistema.
- 9.12 Um sistema de parada de emergência será projetado para determinada máquina de produção automática. Um único botão de iniciar é utilizado para ligar a energia para a máquina no início do dia. Além disso, existem três botões de parar localizados em diferentes posições da máquina e qualquer um pode ser pressionado para que se interrompa imediatamente a alimentação de energia da máquina. (a) Escreva a tabela-verdade para esse sistema de intertravamento. (b) Escreva a expressão lógica booleana para o sistema. (c) Construa o diagrama lógico para o sistema. (d) Construa o diagrama de lógica ladder para o sistema.
- 9.13 Um robô industrial executa as operações de carregamento e descarregamento de uma máquina. Um CLP é utilizado como controlador da célula do robô. A célula opera da seguinte maneira: (1) um trabalhador humano coloca uma peça em um compartimento, (2) o robô se movimenta, pega a peça e a coloca em uma bobina de aquecimento por indução, (3) a peça é aquecida por dez segundos, (4) o robô vai até a peça, pega a mesma e a coloca no transportador de saída. Na etapa (1), um interruptor de presença X1 (normalmente aberto) será usado no compartimento para indicar a presença da peça. Um contato de saída Y1 será utilizado para sinalizar para o robô a execução da etapa (2) do ciclo de trabalho. Ele é uma saída do CLP, mas uma entrada para o sistema de intertravamento para o controlador do robô. O temporizador T1 será usado para gerar um tempo de espera de dez segundos na etapa (3). O contato de saída Y2 será usado para sinalizar ao robô a execução da etapa (4). (a) Construa o diagrama de lógica ladder para o sistema. (b) Escreva as declarações em linguagem de baixo nível para o sistema utilizando o conjunto de instruções CLP da Tabela 9.10.
- 9.14 Um CLP é utilizado para controlar a sequência em uma operação automatizada de furação. Um operador humano carrega e aperta uma peça em estado bruto em um acessório da mesa da furadeira e pressiona um botão para iniciar o ciclo automático. O eixo-árvore da broca liga, desce em direção à peça a uma determinada profundidade (determinada pelo interruptor de contato) e se retrai. A broca então se move para uma segunda posição de furação e a operação de descida e retração da broca é executada. Após a segunda operação de furação, a broca é desligada e o acessório retorna à posição inicial. O trabalhador então retira a peça acabada e coloca outra em estado bruto. (a) Especifique as variáveis de entrada/saída para a operação desse sistema e defina símbolos para cada uma delas. (b) Construa o diagrama de lógica ladder para o sistema. (c) Escreva as declarações em linguagem de baixo nível para o sistema utilizando o conjunto de instruções CLP da Tabela 9.10.
- 9.15 Um forno industrial deve ser controlado da seguinte maneira: os contatos de uma faixa bimetalica dentro do forno fecham se a temperatura ficar abaixo do valor-alvo e abrem quando a temperatura está acima desse valor. Os contatos regulam o relé de controle que liga e

desliga os elementos de aquecimento do forno. Se a porta do forno for aberta, os elementos de aquecimento serão temporariamente desligados até que se feche a porta. (a) Especifique as variáveis de entrada/saída para a operação desse sistema e defina símbolos para cada

uma delas (por exemplo, X1, X2, C1, Y1 etc). (b) Construa o diagrama de lógica ladder para o sistema. (c) Escreva as declarações em linguagem de baixo nível para o sistema utilizando o conjunto de instruções CLP da Tabela 9.10.

## Referências

- [1] BOUCHER, T. O. *Computer automation in manufacturing*. Londres: Chapman & Hall, 1996.
- [2] CLEVELAND, P. "PLCs get smaller, adapt to newest technology". *Instrumentation and Control Systems*, p. 23-32, abr. 1997.
- [3] HUGHES, T. A. *Programmable controllers*, 2.ed. Research Triangle Park, NC: Instrument Society of America, 1997.
- [4] *INTERNATIONAL Standard for Programmable Controllers, Standard IEC 1131-3*. Geneva, Suíça: International Electrotechnical Commission, 1993.
- [5] JONES, T.; BRYAN, L. A. *Programmable Controllers*, International Programmable Controls, Inc. Atlanta, GA: IPC/ASTEC, 1983.
- [6] LAVALEE, R. "Soft logic's new challenge: Distributed machine control". *Control Engineering*, p. 51-8, ago. 1996.
- [7] *MICROMENTOR: Understanding and applying micro programmable controllers*. Milwaukee, WI: Allen-Bradley Company, Inc., 1995.
- [8] MORLEY, R. "The techy history of modicon". *Technology*, 1989. (Manuscrito.)
- [9] RIZZONI, G. *Principles and applications of electrical engineering*, 5. ed. Nova York: McGraw-Hill, 2007.
- [10] STENEROSON, J. *Fundamentals of programmable logic controllers, sensors, and communications*, 3. ed. Upper Saddle River, NJ: Pearson; Prentice Hall, 2004.
- [11] WEBB, J. W.; REIS, R. A. *Programmable logic controllers: Principles and applications*, 4. ed. Upper Saddle River, NJ: Pearson; Prentice Hall, 1999.
- [12] WILHELM, E. *Programmable controller handbook*. Hasbrouck Heights, NJ: Hayden Book Company, 1985.