Capítulo 6

Projeto de automação utilizando o CLP

6.1 Definição dos pontos de entrada e saída

A primeira e talvez a mais importante etapa no processo de automação de um sistema é o levantamento dos pontos de monitoramento referentes à interligação entre o CLP, o equipamento, a máquina ou a planta a ser automatizada. Após a definição desses pontos de entrada e saída, devemos determinar o número de sensores e atuadores do projeto, para análise de sua viabilidade.

Uma vez de posse dessas informações e levando em conta que projetos novos sempre podem estar sujeitos a ajustes de último momento e a solicitações de contingência do cliente, costuma-se manter reserva de entradas e saídas além daquelas do projeto original. Tal adequação também possibilita compatibilizar diferentes modelos de cartões de CLPs comercialmente existentes. Por exemplo, em uma aplicação na qual são necessárias 30 entradas digitais, é comum utilizar cartões que tenham 32 entradas, garantindo duas entradas de reserva para o projeto.

A informação de quais sensores e quantas interfaces existem disponíveis nos módulos de CLP a serem utilizados servirá também como documentação prévia formatada ao programador. O programador deverá vincular cada sensor ou atuador utilizado a um endereço de entrada ou saída do CLP, definindo o endereço para a leitura e escrita de sensores e atuadores que vão compor o projeto.

A tabela 6.1 exemplifica um mapeamento de dispositivos de entrada e saída já definidos de acordo com a programação desenvolvida no projeto.

Tabela 6.1
Mapa de entradas e saídas

Endereço do CLP	Símbolo	Descrição
I0.0	B1	Chave liga – Tipo NA
I0.1	B2	Chave desliga – Tipo NF
I0.2	BE	Chave emergência – Tipo NF
I0.3	S7	Contato relé térmico – NF
Q0.2	M1	Contator – Motor M1

6.2 Descritivo de funcionamento

Com o mapa de variáveis de entrada e saída definido, o próximo passo é entender o funcionamento desejado do processo. O processo pode ser:

- **Descritivo** – O projeto é descrito função por função de forma textual para que todas as possibilidades de funcionamento da automação estejam previstas no documento.
- **Gráfico** – O projeto é descrito por fluxos de operação (fluxograma), definindo a sequência lógica na qual o programa deve ser executado. Em geral, é apresentado quando se sabe o objetivo a ser alcançado, mas não exatamente quais as formas de atuação e aquisição de informação da planta.

- **Diagrama elétrico de comandos** – Costuma ser utilizado em situações de *retrofiting*, ou seja, quando se tem uma máquina com uma automação baseada em comandos de relés e se deseja atualizar a automação com CLP. Algumas vezes, esquemas elétricos ainda existentes da máquina a ser atualizada possuem documentação em diagrama elétrico. Cabe à pessoa que assume o projeto executar a conversão do projeto em lógicas de programa.
- **Montado pelo projetista** – É usado quando não existe documentação da operação ou a documentação encontra-se incompleta. Nesse caso, é importante observar o funcionamento do equipamento, bem como conversar com os operadores, com o objetivo de entender não só o funcionamento em regime normal, mas também os possíveis erros de operação existentes e como corrigir essas falhas.

Em todos os procedimentos citados, deve-se elaborar uma documentação detalhada do processo antes de iniciar o trabalho de programação. Vale ressaltar que qualquer situação prevista durante a confecção do programa que, no entanto, não conste da documentação recebida deve ser esclarecida de forma oficial, ou seja, por meio de documento firmado entre as partes, para a decisão da situação. Com base nesse acordo, pode-se concluir a tarefa em aberto ou o projeto em questão.

6.3 Elaboração de programas

Diversas técnicas podem ser utilizadas para apresentar uma sequência lógica aos programadores que iniciam seus trabalhos com CLP. Esta seção apresenta duas formas para facilitar o entendimento do sequenciamento a ser seguido na elaboração de um programa.

6.3.1 Ligar, manter ligado e desligar

Trata-se de uma forma de elaborar programas utilizando selos lógicos em cada linha do programa. Esse procedimento facilita o trabalho dos programadores sem muita experiência em programação de CLPs, por ser de fácil entendimento e possibilitar a construção de lógicas simples.

De início, vamos levar em conta as seguintes variáveis:

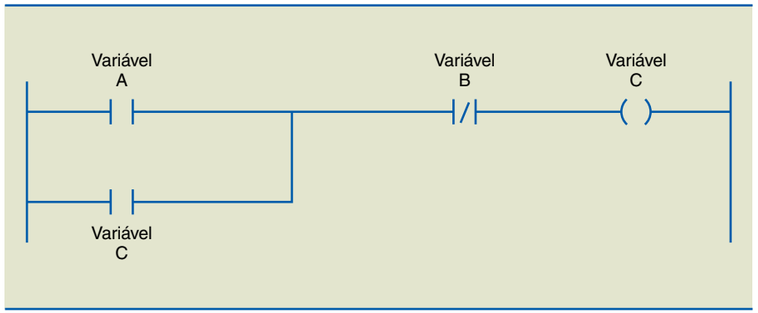
- Variável A: responsável por ligar determinada saída digital.
- Variável B: responsável por desligar determinada saída digital.
- Variável C: saída digital ligada por A e desligada por B.

Completando o descritivo, consideremos que as variáveis A e B são energizadas quando são pressionadas as botoeiras A e B, respectivamente.

A leitura do estado da variável C é justamente o que define se essa variável deve manter-se ativa ou não.

Apresentam-se, a seguir, a linha de programação (figura 6.1) que executa essa tarefa e sua análise.

Figura 6.1
Linha de programação básica.



Quando a botoeira A é acionada, considerando que a botoeira B encontra-se em repouso, a variável A torna-se verdadeira, garantindo a continuidade lógica para o acionamento da variável C. Dado que a variável C agora é verdadeira, sua leitura em paralelo com a variável A ainda garante a continuidade lógica na variável C. Essa situação agora se mantém independente do *status* da variável A. No entanto, basta que a variável B seja falsa, isto é, que a botoeira B seja acionada, para interromper a continuidade lógica na variável C e desligar todo o circuito. Para quem conhece o funcionamento de um selo lógico, não existe muita novidade nesse descritivo.

Agora analisemos essas variáveis de maneira mais crítica. Consideremos as condições para que a variável C não seja somente uma botoeira, mas sim um conjunto de operações, sensores ou estados de memória que, quando todos forem verdadeiros, devam executar determinada ação ou determinado acionamento. Um exemplo clássico é o acionamento de sistemas mecânicos como prensas e furadeiras.

Apresenta-se, a seguir, o descritivo de funcionamento de um projeto desse tipo de forma didática, não levando em conta detalhes mais aprofundados que devem ser observados em sistemas de segurança de máquinas operatrizes com operação humana.

O projeto deve obedecer às seguintes condições:

Para que a furadeira desça e execute sua função de furar a peça, ambas as mãos do operador devem estar acionando as botoeiras B1 e B2. O sensor de peça tem de ser acionado, indicando que existe uma peça a ser furada. A área de segurança protegida por uma cortina de luz, sensor S2, não pode ser invadida. O sensor de início de curso S4 deve estar acionado, indicando que a furadeira se encontra na posição inicial do processo. A botoeira de emergência e o sensor S7 de sobrecarga do motor não podem estar acionados. Nessas condições, a lâmpada L1 deve acender e o contator de acionamento de descida do motor deve ser acionado. Quando o sensor S3 é acionado, o contator que propicia a descida da furadeira deve ser desligado e o contator que propicia a subida do conjunto devr ser acionado. O motor da furadeira tem de ser desligado quando o sensor S4 for acionado novamente. A qualquer momento em que uma das botoeiras do operador for desacionada, S7 apresentar sobrecarga, S1 detectar a ausência de peça ou a cortina de luz for invadida, o processo de descida ou subida com o motor da furadeira acionado deve ser interrompido. Esse processo somente pode ser reiniciado com as condições iniciais de operação garantidas (subindo quando interrompido na subida e descendo quando interrompido na descida). A figura 6.2 mostra o esquema de um sistema de furação de peças em que tal conceito pode ser estudado.

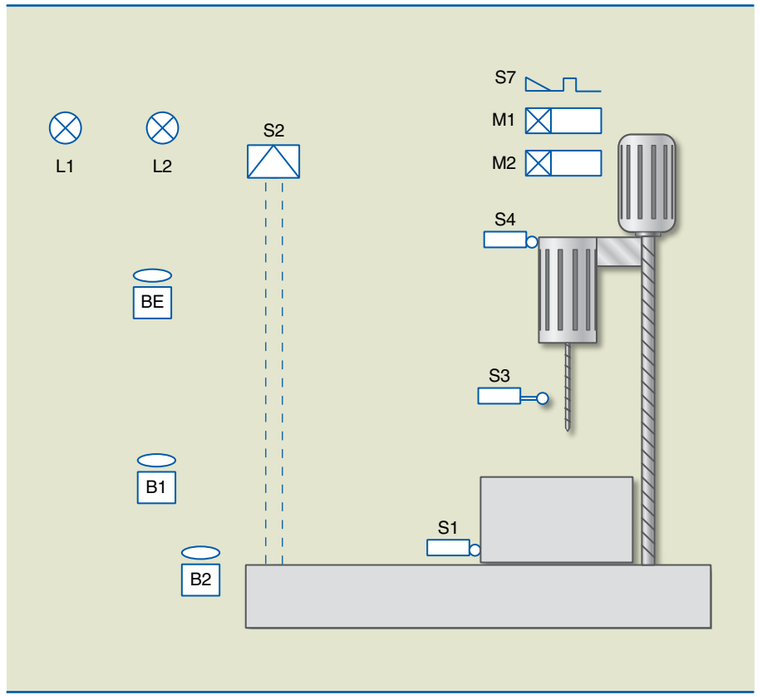


Figura 6.2
Sistema de furação de peças.

Podemos destacar como dispositivos de entrada:

B1 – Botoeira esquerda de acionamento.
B2 – Botoeira direita de acionamento.
S1 – Sensor de peça.
S2 – Cortina de segurança.
S3 – Sensor de fim de curso.
S4 – Sensor de início de curso.
BE – Botão de emergência.
M1 – Contator de avanço da furadeira.
M2 – Contator de recuo da furadeira.
S7 – Sensor de sobrecarga.
L1 – Lâmpada de indicação ligada.
L2 – Lâmpada de indicação desligada.

Apresentadas as variáveis de entrada e saída, o mapa de variáveis pode obedecer à descrição mostrada na figura 6.3.

Figura 6.3
Mapa de variáveis de entrada e saída.

Tabela de Simbolos

Endereço	Símbolo	Comentário
I0.0	BE	Botão Emergencia
I0.1	B1	Botão Esquerdo de Acionamento
I0.2	B2	Botão Direito de Acionamento
I0.3	SENSOR 1	Sensore de Peça
I0.4	SENS FIM	Sensor de Fim de Curso
I0.5	SENS INIC	Sensor de Inicio de Curso
Q0.0	MOT AVANÇ	Contator de Avanço do Motor
Q0.1	MOT RECUO	Contator de Recuo do Motor
I0.6	SENS SOBR	Sensor de Sobrecarga
Q0.2	LAMP 1	Lampada Sistema Ligado
Q0.3	LAMP 2	Lampada Sistema Desligado
I0.7	SENS CORT	Sensor da Cortina de Luz

Inserir todos | Apagar linha | Imprimir | Fechar

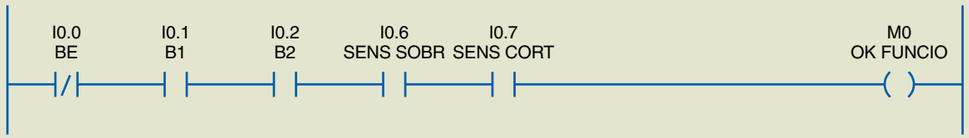
Uma vez configurado o mapa de entradas e saídas, pode-se iniciar a elaboração do programa. Vale ressaltar que existem várias soluções para o mesmo problema, e cada programador pode encontrar outras formas de elaboração. Seguindo a ideia do "ligar, manter ligado e desligar", as linhas de programa podem ser descritas como apresentado a seguir.

A primeira lógica desenvolvida (figura 6.4) tem a função de energizar a memória zero (M_0) quando todas as condições iniciais de funcionamento da ferramenta são verdadeiras, ou seja:

- Não existe um botão de emergência pressionado (BE).
- O operador está pressionando B1 e B2 simultaneamente.
- O sensor de sobrecarga não abriu, ou seja, não está em sobrecarga (SENS SOBR).
- A área do sensor de cortina de segurança S2 não foi invadida (SENS CORT).

Temos a seguinte leitura dessa lógica: todas as variáveis lidas para garantir a continuidade lógica do processo são essenciais, ou seja, a situação necessária para desligar a memória (M_0) é a própria ausência de qualquer um dos sinais lidos. As instruções apresentadas são necessárias para manter energizada a variável, não sendo preciso selar essa saída nem programar lógica específica para desativar o selo.

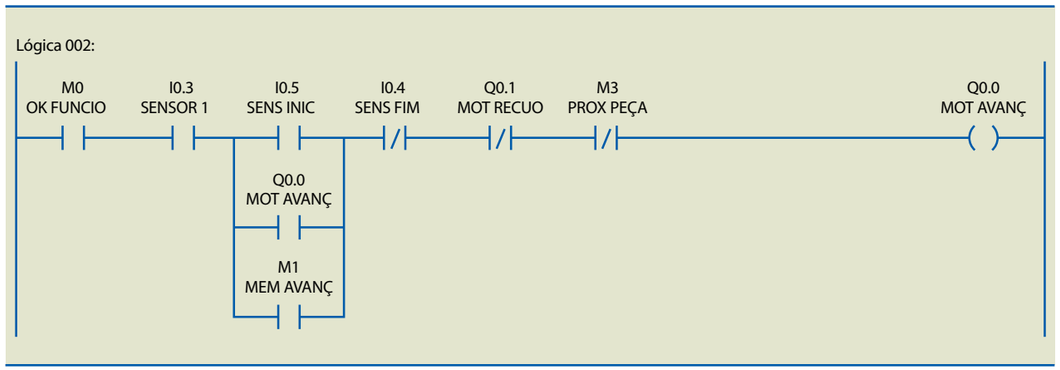
Figura 6.4
Primeira lógica: energizar M_0.



A lógica 2 (figura 6.5) trata do procedimento de descida do conjunto furadeira. Garantidas as condições iniciais pela memória (M0), existindo a peça a ser furada, detectada pelo sensor 1 (I0.3), o sensor de fim de curso (I0.4) não está acionado, pois não haverá movimento vertical do conjunto, uma vez que ele já se encontra na posição final e o contator ligado ao recuo (Q0.1 – MOT RECUO) não está acionado. A memória (M3), que indica próxima peça, não pode estar habilitada (essa variável será explicada na lógica 6). Garantidas essas condições, estando o mecanismo em sua posição inicial, indicada pelo sensor início (I0.5), a saída (Q0.0) será acionada, energizando o contator que propicia o avanço do conjunto e, em paralelo, o acionamento da furadeira.

Toda essa análise garante o acionamento dos motores de descida e da furadeira, mas não que eles permanecerão ligados. Dado o início da descida do conjunto, o sensor de início já não estará ativo, pois o conjunto não estará mais na posição inicial. Dessa maneira, é necessário criar uma lógica que faça com que, uma vez iniciado o processo de descida e furação da peça, ele continue nesse sentido até que o sensor de fim de curso seja acionado. O selo lógico, em paralelo com esse sensor feito com a própria saída (Q0.0), é que garante a continuidade do processo. Existe ainda uma terceira instrução, a memória (M1), comentada na lógica 4, que assegura o procedimento de descida em caso de interrupção momentânea do processo. O sensor de fim de curso (I0.4) exerce a função do "desligar" no esquema "ligar, manter ligado e desligar".

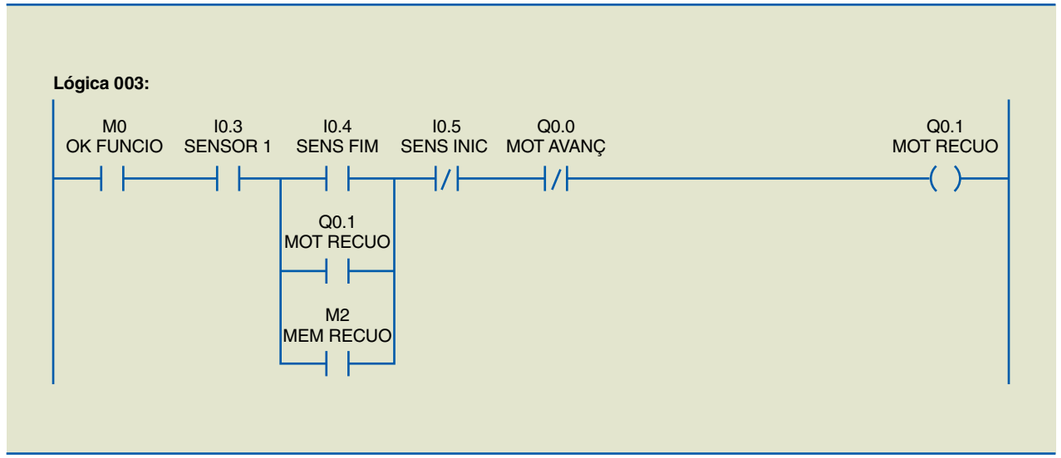
Figura 6.5
Procedimento de descida do conjunto furadeira.



A lógica 3 (figura 6.6) apresenta uma solução para o procedimento de subida do conjunto. Ainda garantidas as condições iniciais pela memória (M0) e existindo a peça a ser furada, detectada pelo sensor 1 (I0.3), a lógica aguarda que o sensor de fim de curso (I0.4) seja acionado. Quando acionado o sensor, essa instrução garante a interrupção da continuidade lógica na saída (Q0.0) do procedimento de descida do conjunto. Uma vez que a descida esteja desabilitada na lógica 2, a mesma instrução que examina se essa variável está em zero na lógica 3 assegura, com o acionamento do sensor de fim de curso, o acionamento do contator responsável pela subida do conjunto. Fazendo novamente a análise do "ligar, manter ligado e desligar", todas essas instruções garantem o início do processo de subida, mas não que ele se mantenha assim depois de iniciado, pois o sensor de fim de curso é automaticamente desabilitado quando esse processo se inicia. Por isso, é

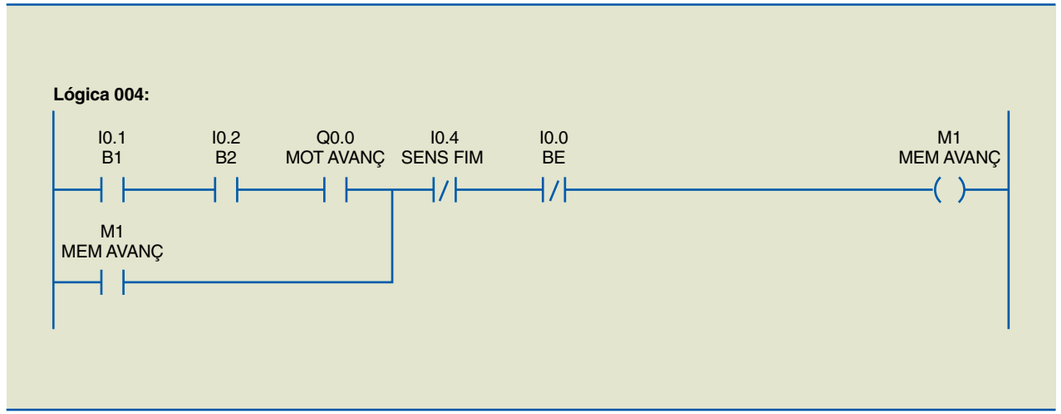
necessário que o selo lógico esteja em paralelo com esse sensor, fazendo com que o processo de subida continue até que a lógica seja interrompida com o acionamento do sensor de início de curso (I0.5). Esse sensor exerce a função do "desligar" no esquema "ligar, manter ligado e desligar".

Figura 6.6
Procedimento de subida do conjunto furadeira.



A lógica 4 (figura 6.7) tem a função específica de memorizar se o conjunto está na situação de descida no instante em que algum evento que interrompa o processo ocorrer. Se as botoeiras B1 e B2 forem acionadas e o contator de descida energizado, a memória de avanço (M1) será energizada. Tal sequência faz parte da lógica "ligar". Essa memória permanecerá nessa condição graças ao selo lógico feito pelo próprio endereço (M1), exercendo a função do "manter ligado" até que o sensor de fim de curso (I0.4) ou o botão de emergência (I0.0) seja acionado, ambos (sensor de fim de curso e botão de emergência) fazendo parte da lógica "desligar", da proposta inicial comentada. A memória (M1) garante que o procedimento de descida, uma vez interrompido na lógica 2, continue até que uma nova condição de parada ocorra ou até que o conjunto termine o procedimento, acionando, assim, o sensor de fim de curso.

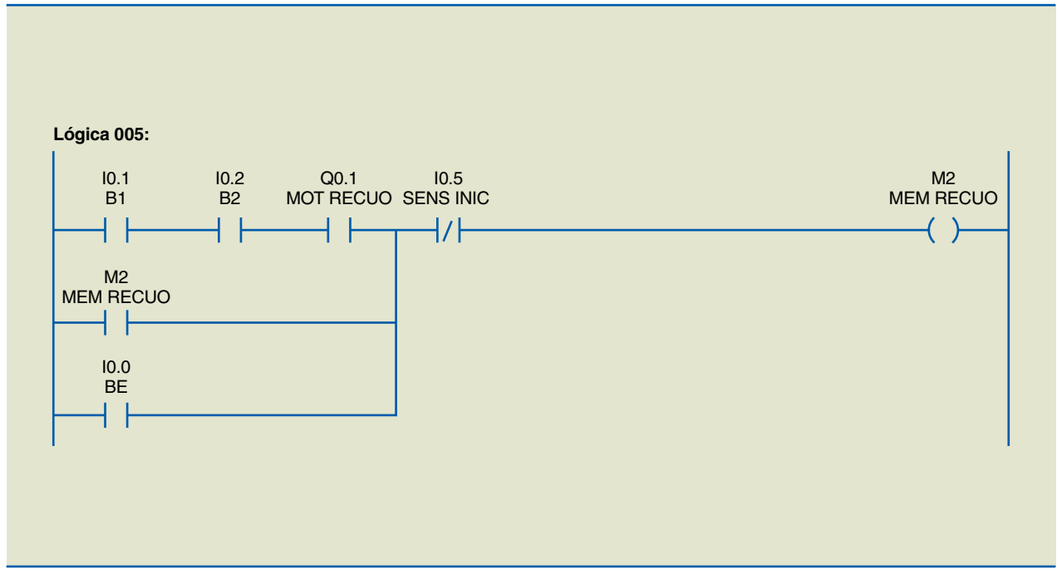
Figura 6.7
Memória de avanço (M1).



A descrição da lógica 5 (figura 6.8) é semelhante à da lógica 4, porém, faz com que o conjunto recue quando as condições de operação estiverem restabelecidas. A principal diferença entre as duas lógicas é que o botão de emergência garante, na 5, que a memória seja acionada e, na 4, que ela seja desligada.

É interessante que a máquina retorne à posição inicial mesmo que o processo ainda não tenha sido completado. Voltando à análise do "ligar, manter ligado e desligar", as botoeiras pressionadas com o acionamento do motor de recuo ou o acionamento do botão de emergência asseguram que a memória seja energizada. A própria memória de recuo faz com que ela se mantenha energizada, e o acionamento do sensor de início de curso garante o desligamento da memória.

Figura 6.8
Memória de recuo (M_2).



A lógica 6 (figura 6.9) bloqueia o processo de descida, depois que o conjunto desceu e subiu, executando a função de furar a peça. Conforme comentado na lógica 2, o endereço de memória (M_3) será verdadeiro se os botões B1 e B2 e também o contador de recuo estiverem acionados. Essas três instruções verdadeiras garantem continuidade lógica no contador (C01), predefinido para totalizar um pulso de entrada e acionar sua saída uma vez que a situação esteja atendida.

A saída *enable* do contador será verdadeira enquanto o sensor de início de processo não estiver acionado e as duas botoeiras liberadas, garantindo que o operador utilizará as mãos para substituir a peça que foi furada por outra que ainda não foi. Ou seja, a ausência das mãos do operador nas botoeiras B1 e B2, mais o fato de a máquina estar na posição inicial faz com que o *reset* do contador seja acionado e que a memória (M3) volte ao estado desligado, permitindo um novo ciclo de processo.

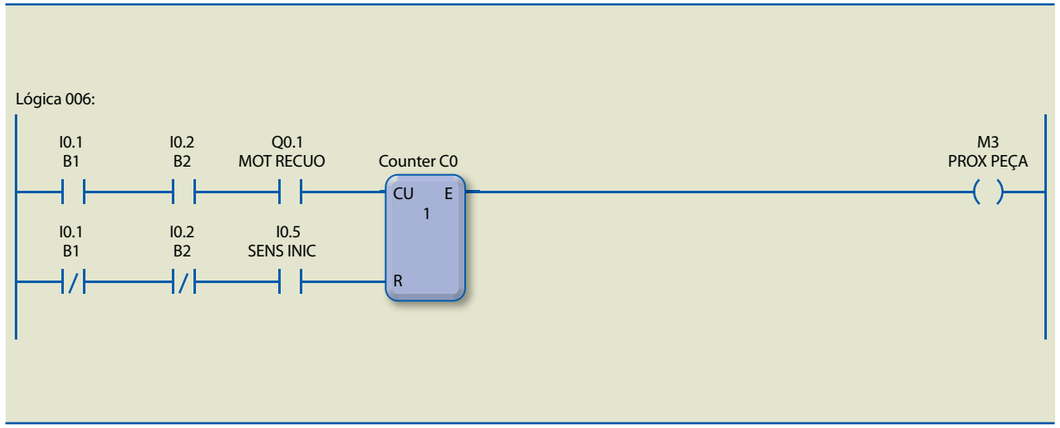


Figura 6.9
Bloqueio do processo de descida.

A lógica 7 (figura 6.10) é um simples complemento que garante o acionamento da lâmpada 1 (Q0.2) quando um dos contatores estiver acionado, seja na descida, seja na subida. Com a saída complementar (Q0.3) ligada à lâmpada 2, esta permanecerá desligada enquanto a lâmpada 1 estiver acionada. Se a lâmpada 1 estiver apagada, a lâmpada 2 será acionada, informando que a máquina está parada nesse instante.

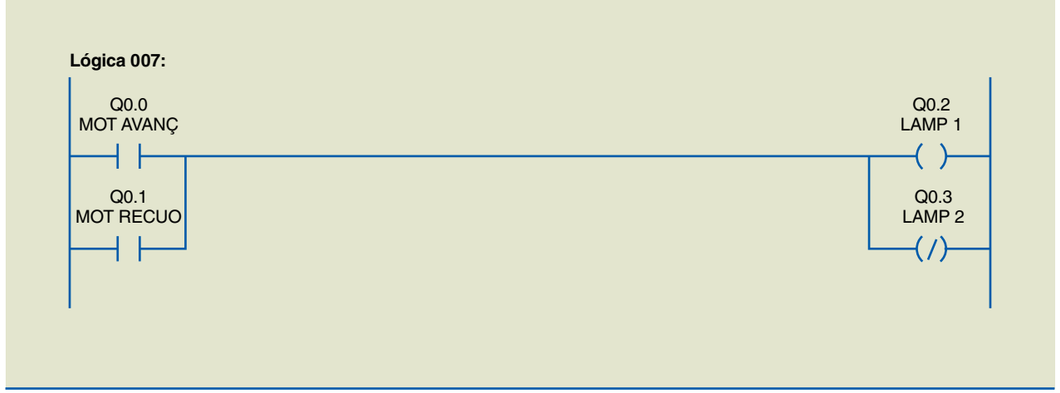


Figura 6.10
Acionamento das lâmpadas.

6.3.2 Passos e transições

A técnica de programação de passos e transições é muito útil na elaboração de programas de natureza essencialmente sequencial, ou seja, quando os acionamentos dependem do ponto da sequência em que o sistema se encontra. Essa técnica baseia-se no estudo do que deve ocorrer e por quanto tempo, caso o controlador receba alguma informação do processo. É recomendada quando a lógica com intertravamento puro se mostra muito extensa, exigindo a criação e manipulação de diversas memórias auxiliares, o que torna o programa como um todo difícil de entender e, não raras vezes, pouco confiável.

Essa técnica consiste basicamente em definir bits que correspondam aos passos da sequência de funcionamento ou estados do programa. Os passos são níveis lógicos que determinadas entradas ou saídas devem encontrar de modo a habilitar uma nova etapa do processo. Para que esse novo passo seja obtido, é necessário que ocorram transições, até que um novo passo seja alcançado.

Exemplificando de forma prática: a válvula que abastece um reservatório só deve ser desligada quando ele estiver cheio, ou seja, o primeiro passo é o reservatório vazio e a primeira transição é encher o reservatório abrindo a válvula. O segundo passo é o reservatório cheio, que provoca uma segunda transição, que é fechar a válvula.

Vários são os processos que podem ser montados com essa técnica, entre eles projetos de esteira transportadora de caixas com cilindros expulsores. Com o sistema descrito na figura 6.11, pode-se aprofundar a análise dessa técnica de programação.

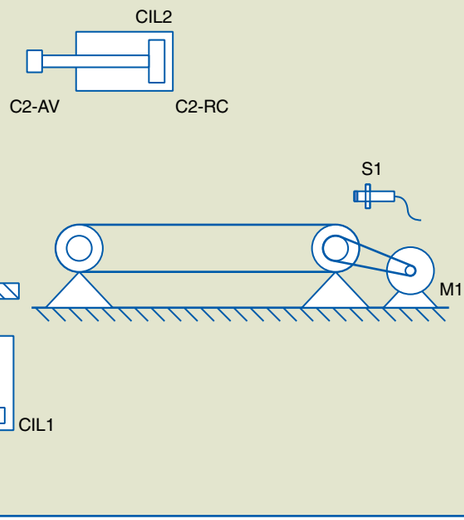


Figura 6.11
Esteira transportadora de caixas com cilindros expulsores.

Sequência de trabalho do sistema:

- Quando uma caixa for detectada pelo sensor S1, o motor M1 deve ser acionado e o transporte da caixa iniciado.
- Quando a caixa se aproximar do sensor S2, o motor deve ser desligado e o avanço do cilindro CIL1 acionado.
- Quando o cilindro CIL1 chegar à sua posição final, o avanço do cilindro CIL2 deve ser acionado.
- Quando o cilindro CIL2 chegar à sua posição final, os cilindros CIL1 e CIL2 devem retornar, habilitando o sistema para um novo ciclo de operação.

Para esse sistema, desenvolveremos a sequência de passos da tabela 6.2.

Passo	Acionamentos realizados	Mudança de passo (transição)	Passo seguinte	Descrição
1 (passo inicial)	Nenhum	Sensor S1 atuado	2	Aguarda caixa ser depositada na cabeceira da esteira.
2	Movimento da esteira – motor M1	Sensor S2 atuado	3	Movimenta a esteira e aguarda a caixa chegar à cabeceira esquerda, mantendo o movimento da esteira.
3	Avanço do cilindro elevador CIL1	Avanço do cilindro CIL1 concluído – sensor C1-AV atuado	4	Para o movimento da esteira e aciona a elevação da caixa.
4	Mantém avanço do cilindro elevador CIL1 Avanço do cilindro expulsor CIL2	Avanço do cilindro CIL1 concluído – sensor C1-AV atuado Avanço do cilindro CIL2 concluído – sensor C2-AV atuado	5	Mantém a elevação da caixa e aciona o cilindro de expulsão.
5	Recuo dos cilindros de elevação e expulsão – recuo dos cilindros CIL1 e CIL2	Recuo do cilindro CIL1 concluído – sensor C1-RC atuado Recuo do cilindro CIL2 concluído – sensor C2-RC atuado	1	Recua cilindros pneumáticos e retorna ao passo inicial.

Tabela 6.2
Sequência de passos

Pela sequência, é possível perceber que cada passo do programa corresponde a um ponto em que dada ação deve ser tomada, seja a execução ou a interrupção de uma instrução, seja a espera de determinada condição. Dessa maneira, durante todo o ciclo de funcionamento do programa, apenas um estado da sequência permanece ativo por vez, garantindo o comportamento preciso e confiável do sistema.

Podemos utilizar a definição de que cada passo do programa corresponde a um bit. Os passos são executados de forma sequencial; o primeiro deles deve ser o primeiro da sequência lógica definida no programa e pré-requisito de análise para os subsequentes. Nessa técnica, é comum a padronização com o uso das instruções *set* e *reset*, vistas no capítulo 5, ou seja, em cada transição, devemos ressetar o passo dado e setar o seguinte.

Como exemplo, implementaremos o programa da esteira transportadora de caixas. O objetivo é descrever o programa em Ladder do sistema, utilizando a técnica de passos e transições para o desenvolvimento da sequência de operações.

Cada passo do programa corresponderá a uma memória do CLP. Portanto, o passo inicial para programar é a alocação das memórias que corresponderão aos passos. No exemplo apresentado, há apenas cinco passos, logo serão necessárias apenas cinco memórias. Recomenda-se reservar maior quantidade de memórias que os passos do programa, mesmo que muitas delas não sejam utilizadas, pois servirão para futuras expansões no sistema. Reserva-se, assim, os primeiros 16 bits da memória para os passos do sistema. A tabela 6.3 mostra a alocação dos passos e das memórias.

Passo	Símbolo	Memória
1	PAS1	M0
2	PAS2	M1
3	PAS3	M2
4	PAS4	M3
5	PAS5	M4
...	...	...
14	PAS14	M13
15	PAS15	M14
16	PAS16	M15

Tabela 6.3
Alocação dos passos e das memórias do sistema

Uma vez definidos os sensores e atuadores do sistema, tal informação também deve estar definida no mapa de entradas e saídas, conforme mostra a tabela 6.4.

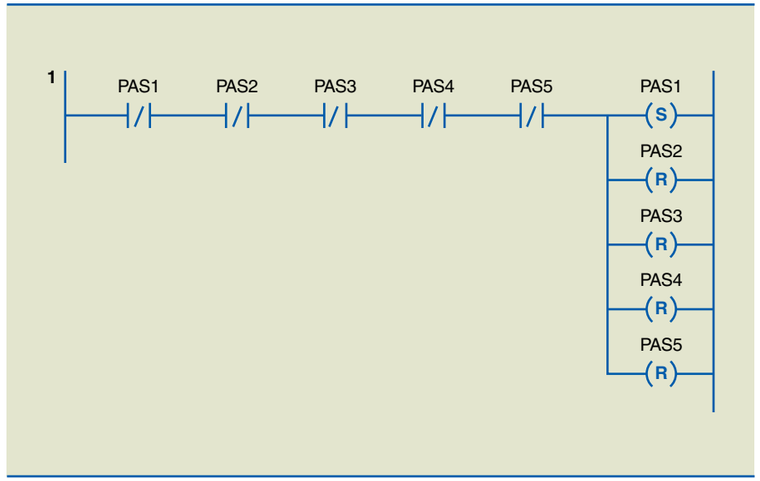
Tabela 6.4
Mapa de entradas e saídas (I/O)

Sensores	Descrição	Memória
S1	Sensor de caixa 1 (caixa presente = nível lógico 1)	I0.0
S2	Sensor de caixa 2 (caixa presente = nível lógico 1)	I0.1
C1-AV	Sensor cilindro 1 (posição máxima de avanço = nível lógico 1)	I0.2
C2-AV	Sensor cilindro 2 (posição máxima de avanço = nível lógico 1)	I0.3
C1-RC	Sensor cilindro 1 (posição máxima de recuo = nível lógico 1)	I0.4
C2-RC	Sensor cilindro 2 (posição máxima de recuo = nível lógico 1)	I0.5
Atuadores	**Descrição**	**Memória**
M1	Contator de acionamento do motor 1 da esteira (liga motor em 1)	Q0.0
CIL1	Válvula solenoide para o cilindro 1 (avanço em 1, recuo em 0)	Q0.1
CIL2	Válvula solenoide para o cilindro 2 (avanço em 1, recuo em 0)	Q0.2

Declaradas as variáveis e as memórias vinculadas aos passos do programa, pode-se começar a escrevê-lo.

A primeira linha (figura 6.12) é utilizada para fazer o programa assumir o passo inicial quando nenhum outro passo estiver selecionado. Portanto, quando nenhum passo estiver selecionado, aciona-se o passo 1.

Figura 6.12
Programa para assumir o passo inicial.

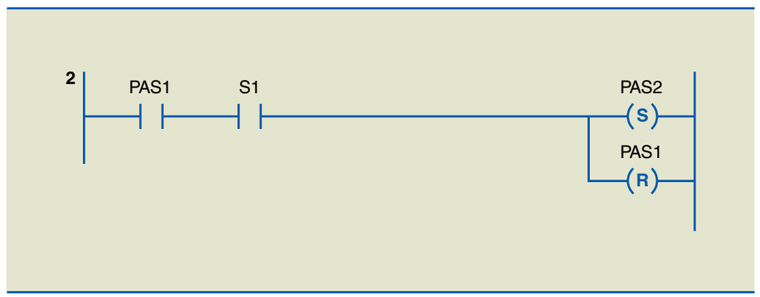


Geralmente, os sistemas possuem um botão de *reset* geral, que retorna o sistema à posição inicial. Esse comando de *reset* geral costuma ser colocado na linha acima, como condição para retorno dos passos ao ponto inicial, e em paralelo com o conjunto de contatos NF em série (PAS1 a PAS5).

Após essa linha de instruções, será iniciada a construção dos passos do programa propriamente ditos. Para simplificar o entendimento, vamos escrever as linhas dos passos na ordem: passo 1, passo 2, passo 3 etc.

O passo 1 (figura 6.13) apresenta a linha de transição do passo 1 para o próximo passo – no caso, passo 2. Tal transição é feita quando o sensor S1 está atuado. Portanto, o passo 2 é acionado quando o programa está no passo 1 e o sensor S1 ativo. O passo 2 é acionado e o passo 1 ressetado.

Figura 6.13
Transição do passo 1 para o passo 2.



A figura 6.14 ilustra a linha que representa a transição do passo 2 para o passo 3. Essa transição é iniciada quando, estando no passo 2, se detecta o acionamento do sensor S2. Lembre-se de que essa transição representa o abandono do passo 2 e o acionamento do passo 3.

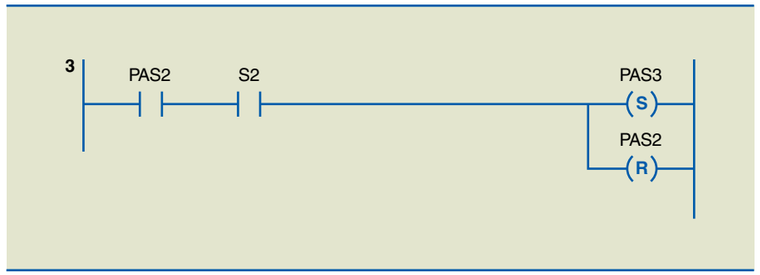


Figura 6.14
Transição do passo 2 para o passo 3.

Seguindo com a lógica, desenvolveremos a linha que representa a transição do passo 3 para o passo 4 (figura 6.15). Essa transição ocorre com a conclusão do movimento de avanço do cilindro CIL1 no passo 3, ou seja, quando o sensor C1-RC está desligado e o sensor C1-AV é acionado durante o passo 3.

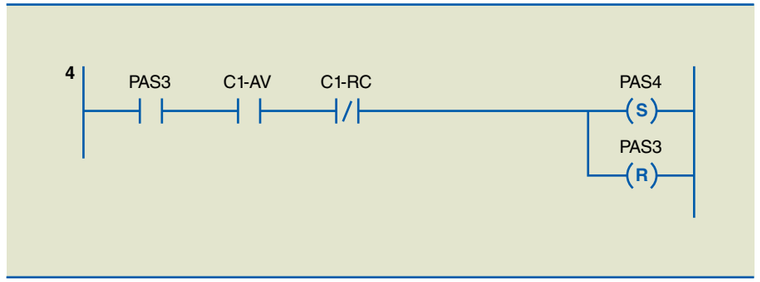


Figura 6.15
Transição do passo 3 para o passo 4.

A transição do passo 4 para o passo 5 (figura 6.16) é semelhante à última linha, mas envolve o avanço dos cilindros CIL1 e CIL2. Essa transição ocorre quando se tem a confirmação de que CIL1 e CIL2 estão avançados no passo 4. Considerando que o reconhecimento de avanço de determinado cilindro é dado pelo acionamento de seu sensor de avanço mais o desacionamento de seu sensor de recuo, a linha para a transição será representada como na figura 6.16.

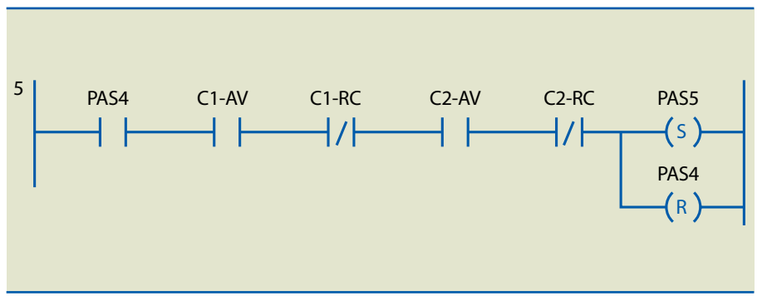
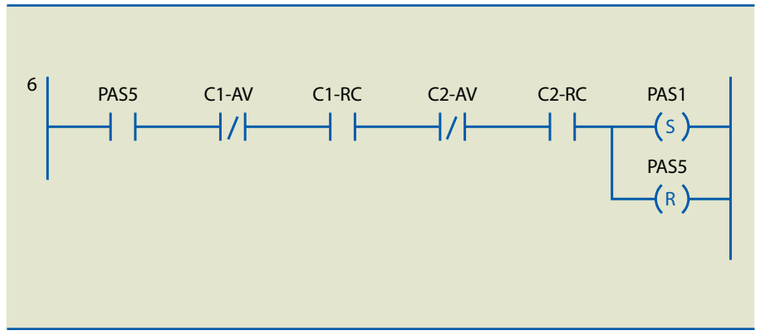


Figura 6.16
Transição do passo 4 para o passo 5.

Finalizadas as linhas de transição, deve-se realizar a transição do passo 5 de volta ao passo 1 (figura 6.17), a qual ocorre quando CIL1 e CIL2 estão recuados. Portanto, o acionamento do passo 1 pode ser escrito com o reconhecimento de ambos os cilindros recuados durante o passo 5, sendo o reconhecimento de recuo de um cilindro o acionamento de seu sensor de recuo com o desligamento de seu sensor de avanço. A figura 6.17 ilustra a linha para essa transição.

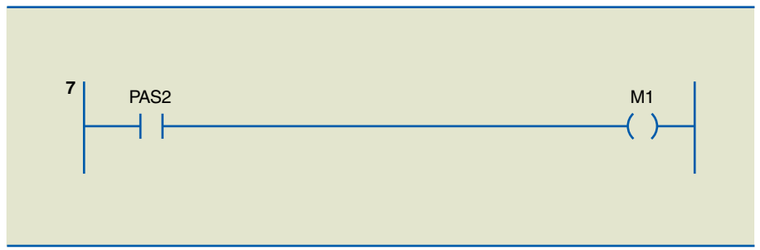
Figura 6.17
Transição do passo 5 para o passo 1.



Toda a sequência de transições é concluída com a linha da figura 6.17, restando somente elaborar as linhas de acionamento de cada dispositivo em seu respectivo passo.

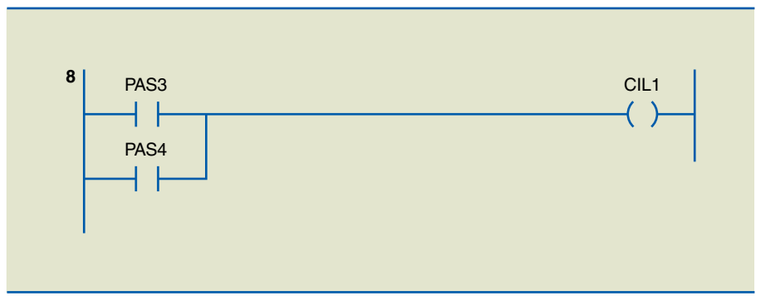
O motor M1 é acionado no passo 2, conforme mostra a figura 6.18.

Figura 6.18
Acionamento do motor M1.



O cilindro CIL1 é acionado nos passos 3 e 4, como ilustra a figura 6.19.

Figura 6.19
Acionamento do cilindro CIL1.



O cilindro CIL2 é acionado no passo 4, tendo como última linha a da figura 6.20.

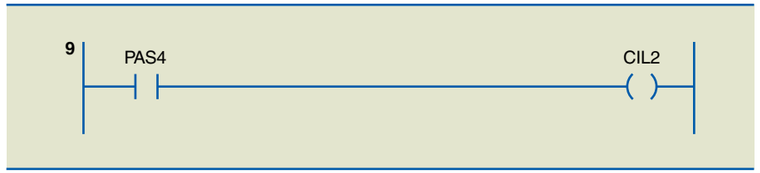


Figura 6.20
Acionamento do cilindro CIL2.

6.4 Testes, simulações e alterações

É grande o número de *softwares* de programação de CLPs que permitem a simulação dos programas elaborados, porém, são genéricos e limitados. No procedimento de teste e simulação de um programa, devem ser consideradas com rigor as características dos dispositivos de entrada que serão utilizados no projeto físico. Uma das causas de erros de programação é não levar em conta a variedade de tipos de dispositivos de entrada, como chaves e botões pulsantes, chaves e botões que possuem contatos normalmente fechados, chaves de duas posições (que se mantêm na posição) etc.

Os *softwares* de simulação normalmente disponibilizam chaves biestáveis para o acionamento de entradas digitais. Essas chaves são de operação retentiva e podem confundir os que se iniciam nessa tarefa de programação Ladder. Considerando que as chaves disponibilizadas pelos *softwares* são retentivas, a simulação de uma botoeira não retentiva, por exemplo, nada mais é do que o acionamento e o desacionamento da chave disponibilizada pelo *software*, ou seja, a geração de um pulso na entrada digital em teste.

Cabe alertar que ocorrem erros de simulação quando essa chave é acionada e esquecida pelo programador, simulando que o operador da máquina está o tempo todo com a botoeira pressionada. Essa situação pode camuflar possíveis erros de lógica, como a necessidade de inserção de selos lógicos na programação.

Outro erro comum é usar instruções invertidas quando se precisa elaborar um programa com chaves ou sensores NF. Para garantir a continuidade lógica de uma linha de programa quando o dispositivo NF está em situação normal de operação, deve-se utilizar a instrução NA (–| |–), pois é a chave que garante energia à entrada digital e, por consequência, torna a instrução NA verdadeira na lógica construída.

Simular um dispositivo NF em um *software* de simulação é iniciar o programa com a chave fechada. Quando pressionada, a chave NF se abre, interrompendo a continuidade lógica na linha de programação que utiliza uma instrução NA monitorando tal variável.

Simulações devem ser feitas com cautela e dificilmente substituem um teste final na liberação do projeto.

6.5 Liberação de projeto

O sucesso na liberação de um projeto está diretamente atrelado à qualidade da produção do projeto, no que se refere à identificação dos componentes, à sua descrição técnica, aos diagramas unifilares indicando claramente em que porta do CLP cada sensor ou atuador vai ser ligado etc. Todo o processo de teste, incluindo o monitoramento das variáveis, a energização de contatores e o reconhecimento de sensores, deve ser feito antes de o circuito de potência ser energizado. Tal processo pode durar dias, semanas ou meses, mas é de vital importância para garantir que todas as possíveis falhas sejam investigadas e, se existirem, corrigidas. Essa etapa garante a integridade física dos operadores e do processo durante os testes e durante a operação do sistema.

6.6 Projetos para estudo

6.6.1 Motor trifásico com reversão

Será elaborado um programa para o CLP que implemente a partida direta de um motor trifásico com reversão. O esquema elétrico é apresentado na figura 6.21, e a ligação no CLP, na figura 6.22. O mapeamento das entradas e saídas utilizado pode ser observado na figura 6.23.

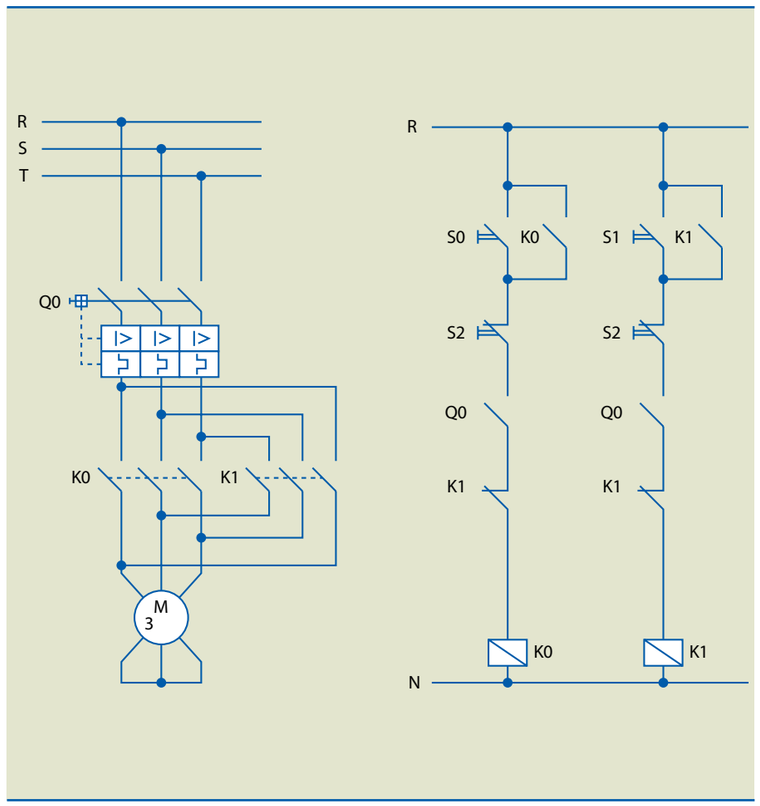


Figura 6.21
Esquema elétrico de partida de motor trifásico com reversão.

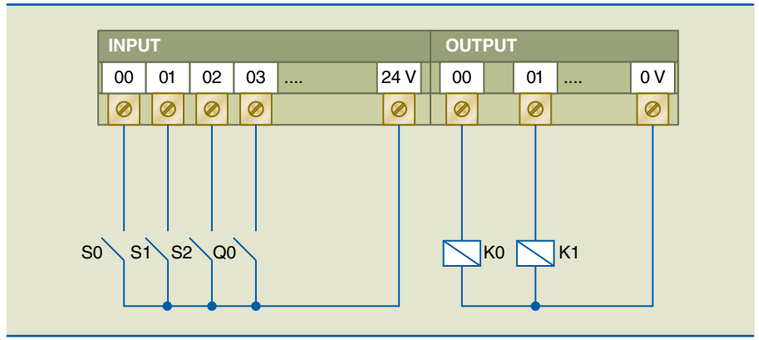


Figura 6.22
Esquema elétrico de ligação no CLP do projeto de motor trifásico com reversão.

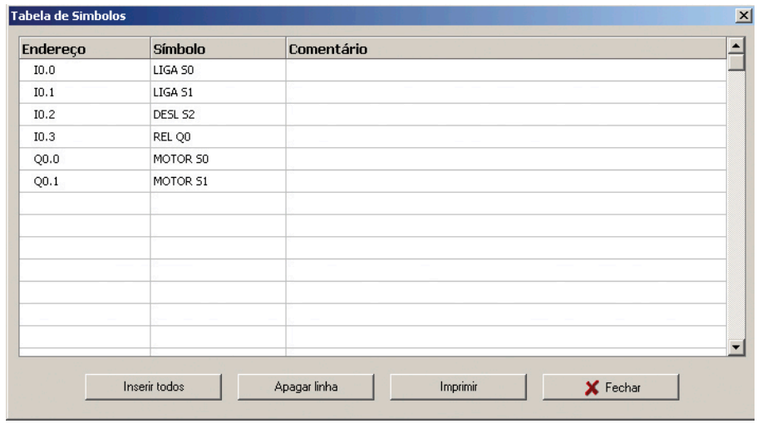


Figura 6.23
Mapeamento das entradas e saídas do projeto de motor trifásico com reversão.

A solução proposta é descrita a seguir.

Lógica 1 (figura 6.24) – (I0.0) inicia todo o processo de energizar a saída (Q0.0). Isso ocorrerá desde que: a chave S2 (tipo NF) não esteja pressionada; o relé térmico (I0.3) (NF) não esteja atuado; o botão S1 de reversão não esteja pressionado; e o contator (Q0.1) que aciona o motor no sentido anti-horário não esteja acionado. Uma vez atuada a saída (Q0.0), o selo lógico sobre a chave S0 é mantido até que uma das demais condições citadas não seja mais válida.

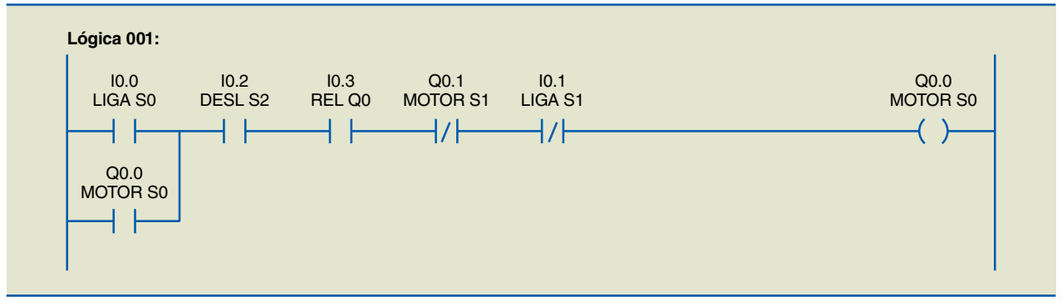
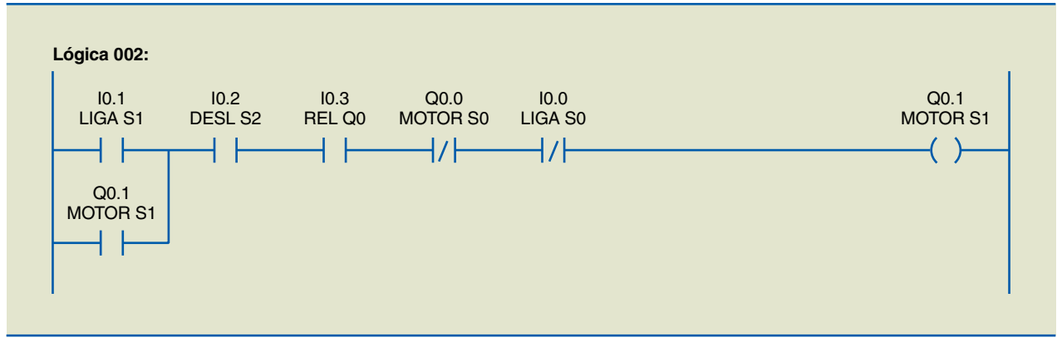


Figura 6.24
Acionamento do motor no sentido horário com S0.

Lógica 2 (figura 6.25) – De maneira análoga à lógica 1, o que define esse sentido de rotação é a chave S1 ligada ao endereço (I0.1). Isso ocorrerá desde que: a chave S2 (tipo NF) não esteja pressionada; o relé térmico (I0.3) não esteja atuado; o botão S0 de reversão não esteja pressionado; e o contator (Q0.0) que aciona o motor no sentido horário não esteja acionado. Uma vez atuada a saída (Q0.1), o selo lógico sobre a chave S1 é mantido até que uma das demais condições citadas não seja mais válida.

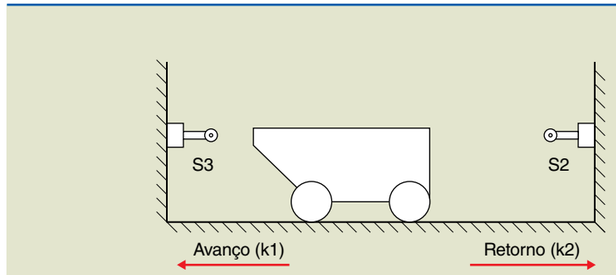
Figura 6.25
Acionamento do motor no sentido anti-horário com S1.



6.6.2 Carro transportador

Será desenvolvido o programa do CLP para controle de um carro transportador. O carro transportador deve funcionar da seguinte maneira: o operador pressiona o botão S1 para dar o comando de avanço do carro. Na posição inicial (recuada), o sensor fim de curso S2 permanece atuado. Uma vez pressionado o botão S1, o carro inicia seu movimento de avanço, por meio do contator K1. Ao atingir o fim da trajetória, o sensor fim de curso S3 é acionado, momento no qual o movimento de avanço é interrompido e o movimento de recuo é acionado pelo contator K2, automaticamente. O carro transportador continua seu movimento de retorno até que o sensor fim de curso S2 seja acionado de novo, quando o carro transportador deve parar. Caso o carro transportador encontre-se parado no meio da trajetória, o operador tem de pressionar o botão S1 para que ele recue. A representação do processo é ilustrada na figura 6.26; e a ligação dos dispositivos ao CLP, na figura 6.27. O mapeamento das entradas e saídas utilizado pode ser observado na figura 6.28.

Figura 6.26
Representação do processo.



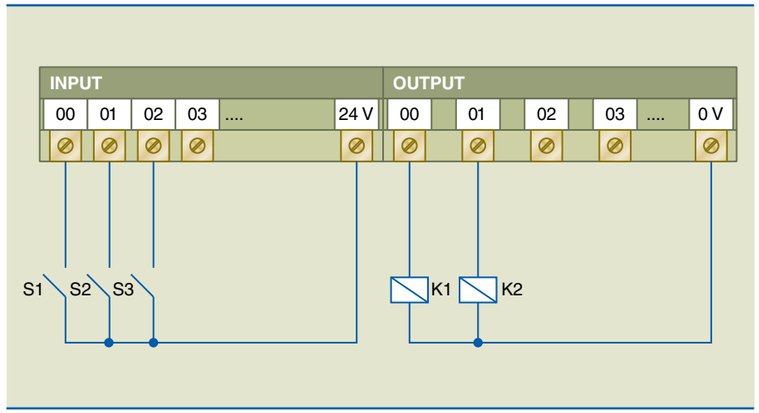


Figura 6.27
Esquema elétrico de ligação no CLP do projeto de carro transportador.

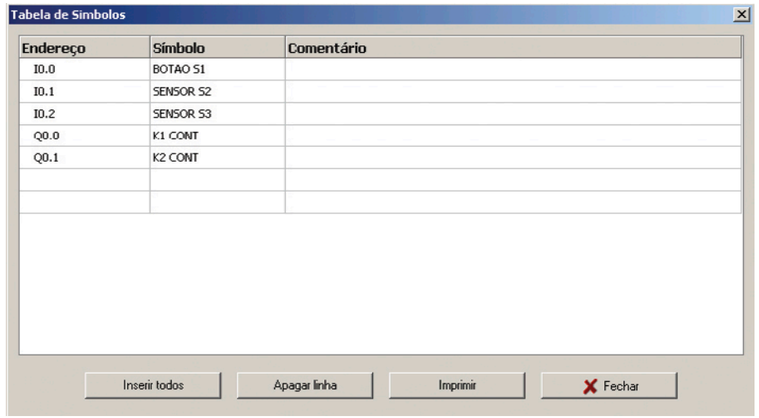
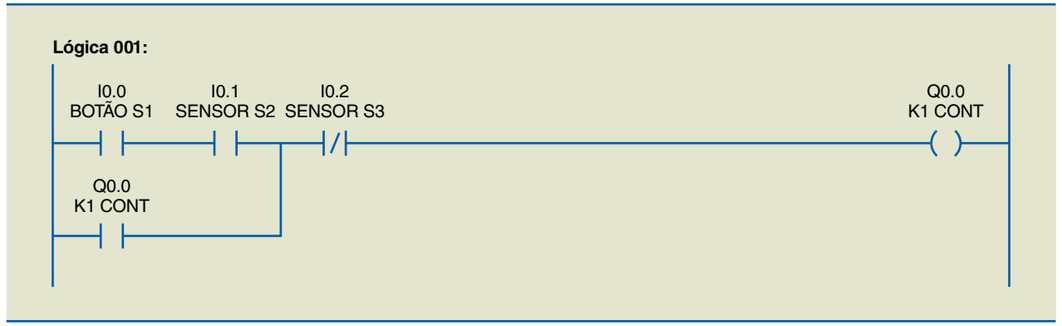


Figura 6.28
Mapeamento das entradas e saídas do projeto de carro transportador.

A solução proposta é descrita a seguir.

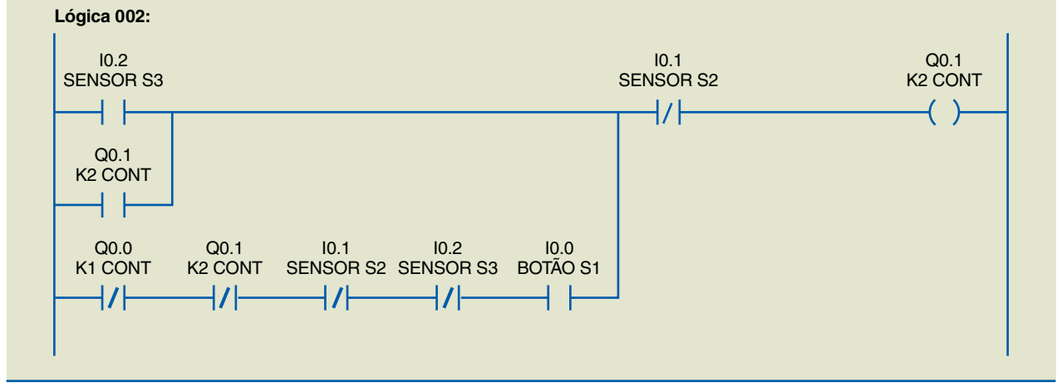
Lógica 1 (figura 6.29) – O acionamento do motor no sentido avanço ocorre quando o botão S1 é pressionado, desde que o sensor S2 esteja acionado, até o momento de acionamento de S3. O selo lógico é garantido por (Q0.0).

Figura 6.29
Lógica 1.



Lógica 2 (figura 6.30) – (Q0.1) o contator responsável pelo retorno do carro é acionado quando, em movimento, o carro aciona o sensor S3. Após interrupção de energia, quando o carro se encontra no meio do caminho, deve-se pressionar S1 para que ele retorne à posição inicial. O selo lógico é mantido pela própria saída (Q0.1), até que o carro atinja a posição inicial.

Figura 6.30
Lógica 2.



6.6.3 Semáforos

Será desenvolvido um programa no CLP para controle dos semáforos de um cruzamento de duas avenidas (figura 6.31). A mudança dos sinais dos semáforos é feita por temporização. Os semáforos permanecem em cada estado por um período de 5 segundos. Os estados dos sinais são apresentados na tabela 6.5; o esquema de ligação dos dispositivos no CLP, na figura 6.32; e o mapeamento das saídas utilizado, na figura 6.33.

Figura 6.31
Cruzamento e semáforos.

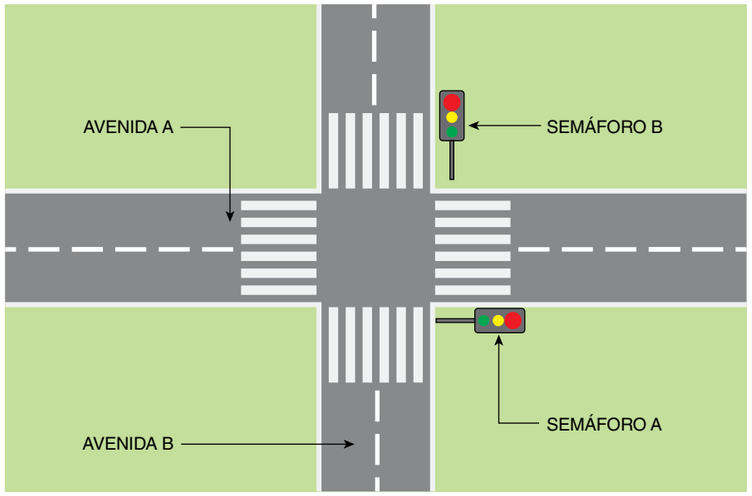


Tabela 6.5
Tabela de estados dos semáforos

Estado	Semáforo avenida A	Semáforo avenida B
1	Verde	Vermelho
2	Amarelo	Vermelho
3	Vermelho	Verde
4	Vermelho	Amarelo
Ao sair do estado 4, os semáforos retornam ao estado 1.		

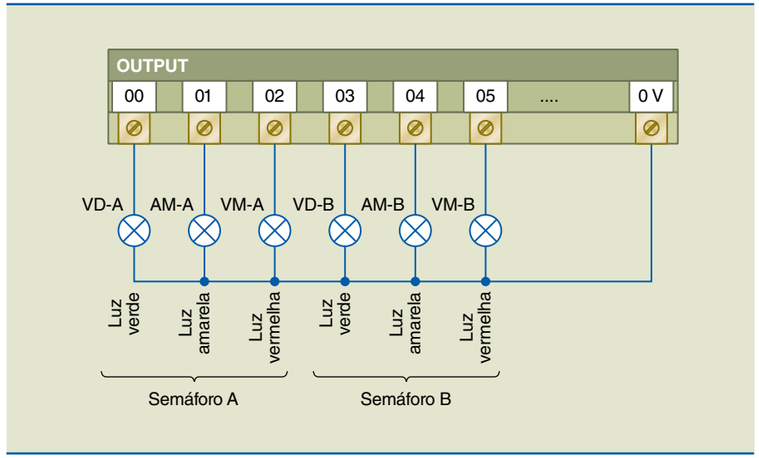


Figura 6.32
Esquema elétrico de ligação no CLP do projeto de semáforos.

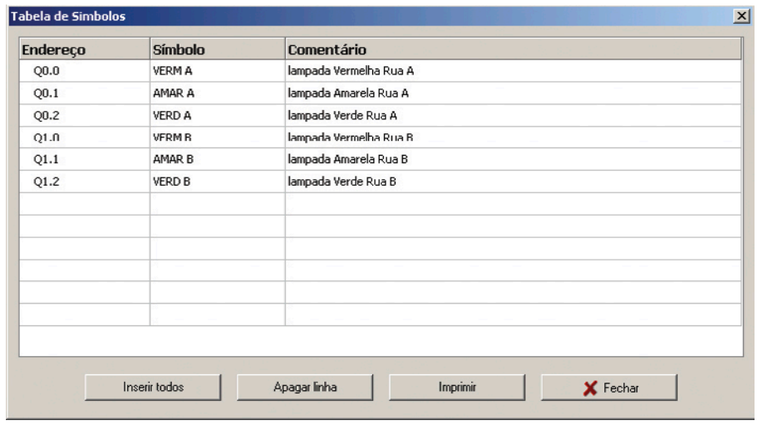
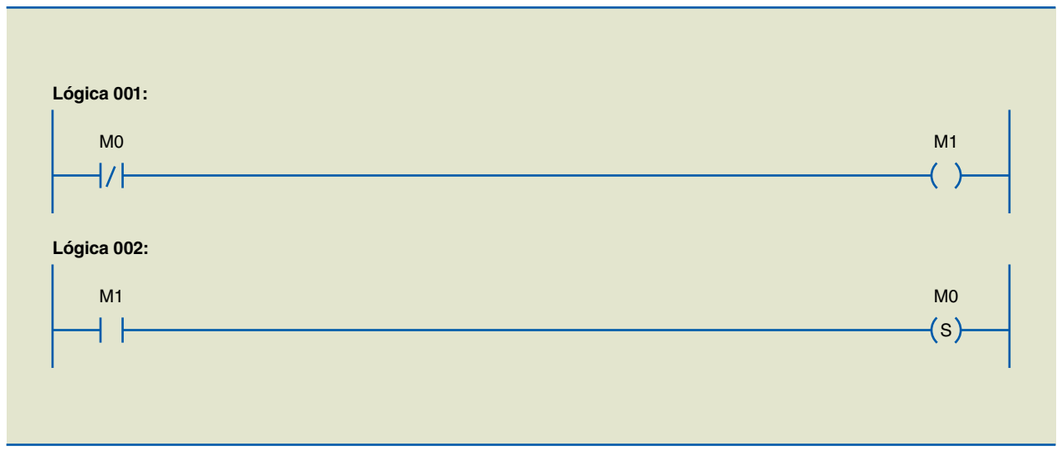


Figura 6.33
Mapeamento das saídas do projeto de semáforos.

A solução proposta é descrita a seguir.

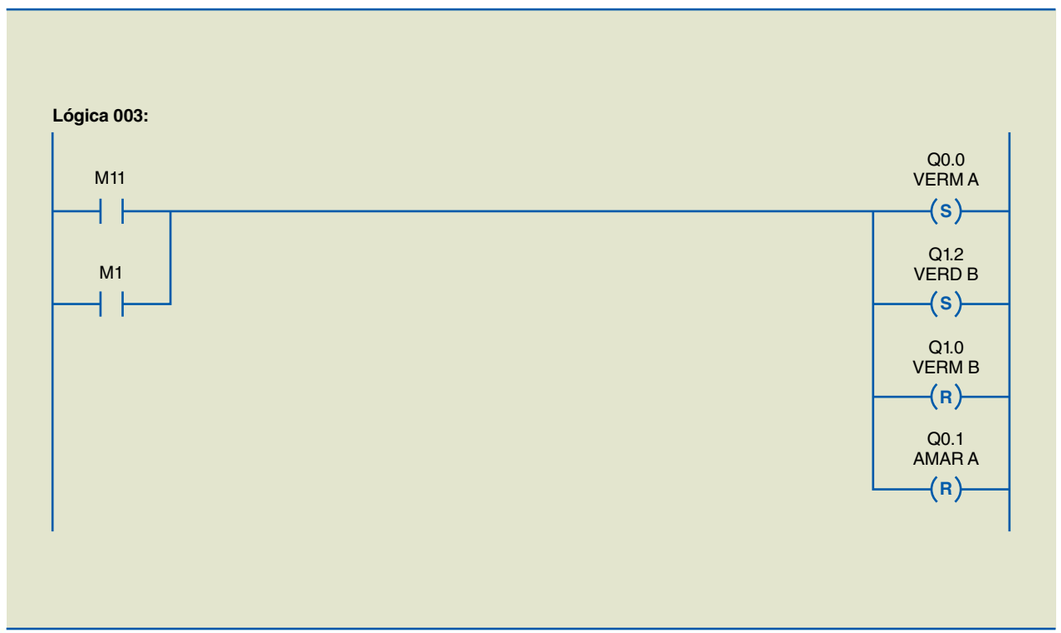
Lógicas 1 e 2 – Geram um pulso de início do processo em (M1) sempre que o CLP for iniciado em condições normais de operação (figura 6.34). Preveem situação de queda de energia, não sendo necessário um botão de *start* para o processo ser inicializado.

Figura 6.34
Lógicas 1 e 2.



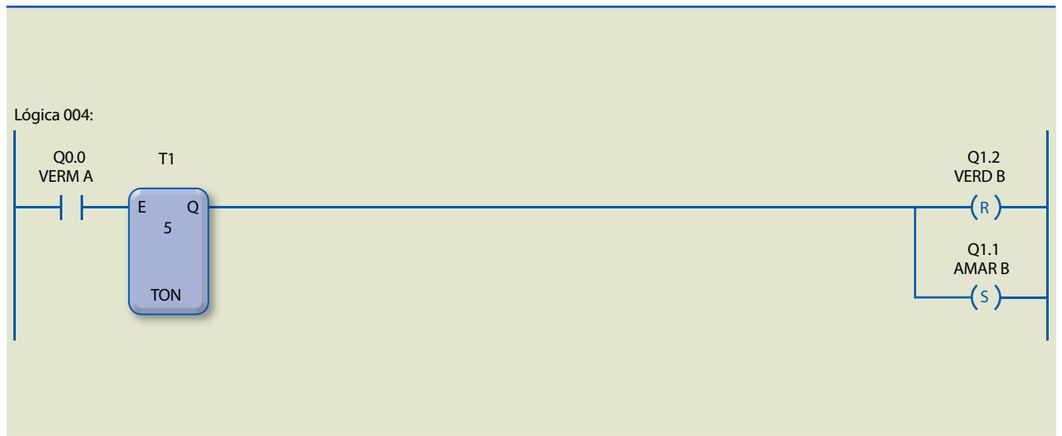
Lógica 3 – Uma vez que o processo seja iniciado por (M1) ou reiniciado por (M11), a lógica define a situação inicial de ligar a lâmpada vermelha do semáforo A e a verde do B, apagando a vermelha do B e a amarela do A (figura 6.35).

Figura 6.35
Lógica 3.



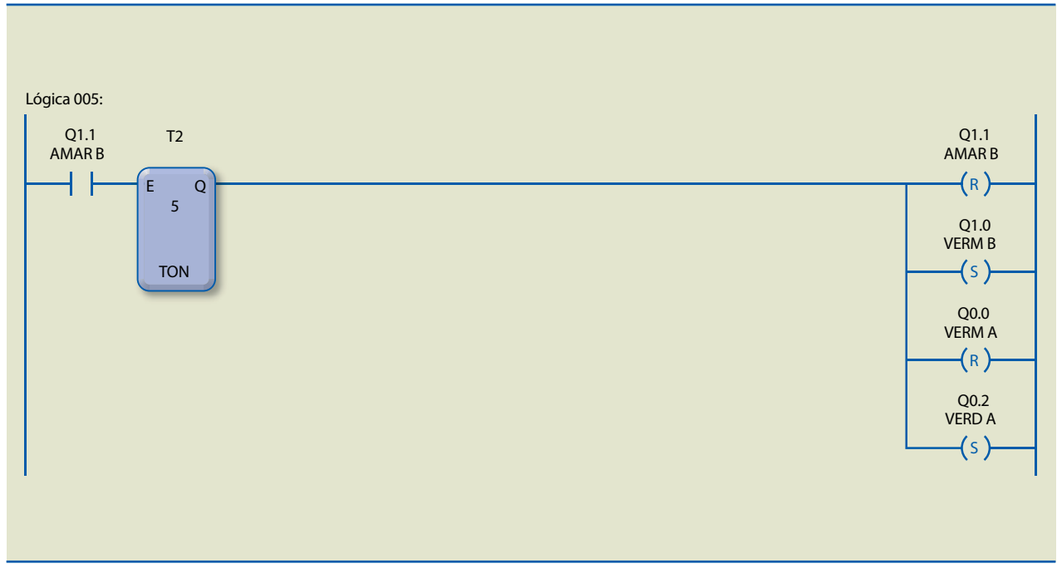
Lógica 4 – Uma vez acionada a lâmpada vermelha do semáforo A, contam-se 5 segundos para que a lâmpada verde do B se apague e a amarela do B seja acionada (figura 6.36).

Figura 6.36
Lógica 4.



Lógica 5 – Uma vez acionada a lâmpada amarela do semáforo B, contam-se 5 segundos e a lógica, então, segue apagando a lâmpada amarela do B e a vermelha do A e acendendo a vermelha de B e a verde de A (figura 6.37).

Figura 6.37
Lógica 5.



Lógica 6 – Uma vez acionada a lâmpada vermelha do semáforo B, contam-se 5 segundos para que a lâmpada verde do A se apague e a amarela do A seja acionada (figura 6.38).

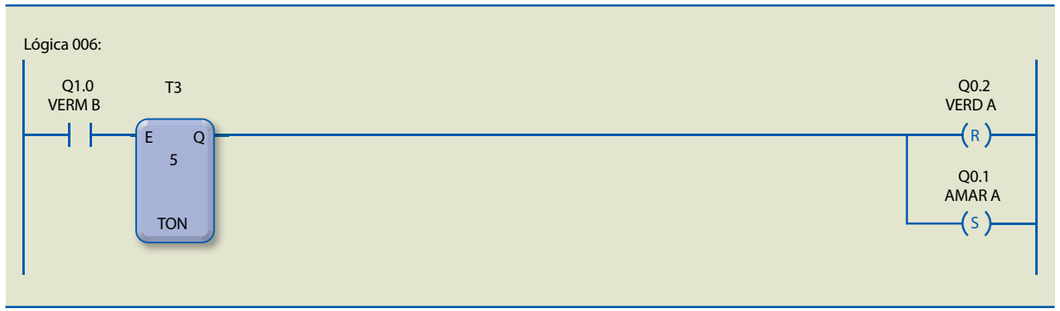


Figura 6.38
Lógica 6.

Lógica 7 – Ao final de um ciclo, após 5 segundos que a lâmpada amarela do semáforo A foi acionada, a memória (M11) aciona o processo da lógica 3, reiniciando o ciclo por tempo indeterminado (figura 6.39).

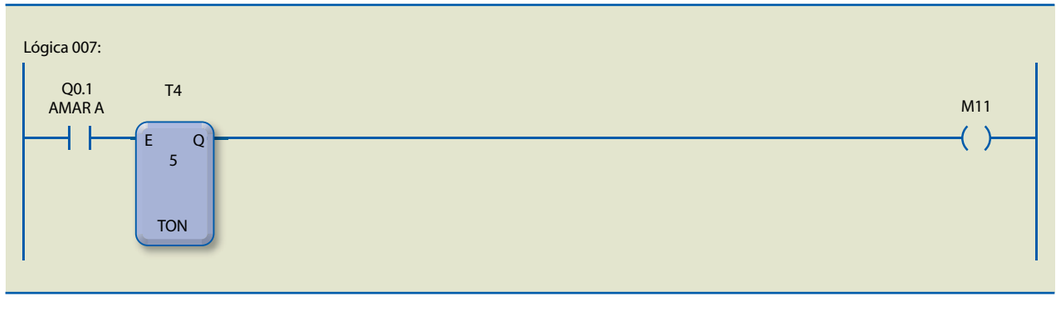
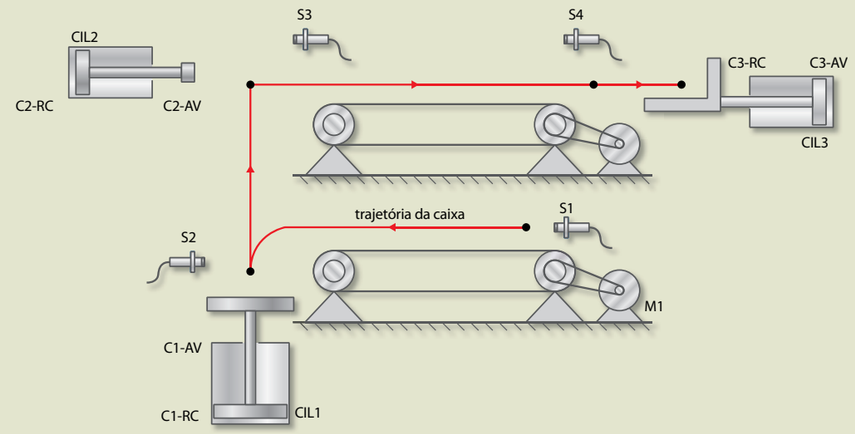


Figura 6.39
Lógica 7.

6.6.4 Transportadora versão 2

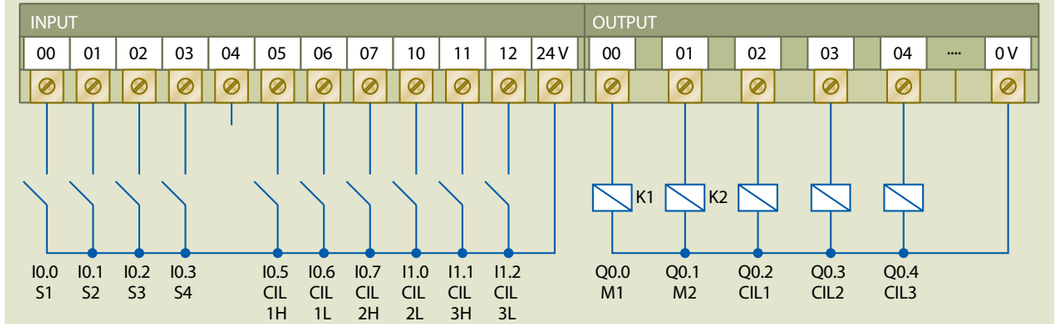
O sistema da figura 6.40, composto por duas esteiras transportadoras e três cilindros pneumáticos, foi projetado para o transporte de caixas.

Figura 6.40
Transportadora versão 2.



O esquema de ligação no CLP é apresentado na figura 6.41; e o mapeamento das entradas e saídas utilizado, na figura 6.42.

Figura 6.41
Esquema de ligação no CLP do projeto de transportadora versão 2.



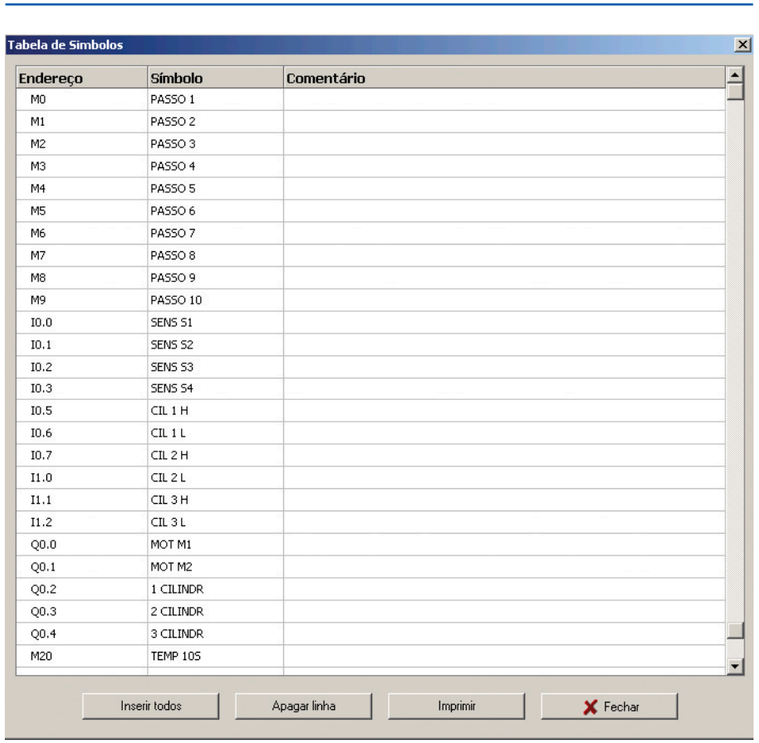


Tabela de Símbolos

Endereço	Símbolo	Comentário
M0	PASSO 1	
M1	PASSO 2	
M2	PASSO 3	
M3	PASSO 4	
M4	PASSO 5	
M5	PASSO 6	
M6	PASSO 7	
M7	PASSO 8	
M8	PASSO 9	
M9	PASSO 10	
I0.0	SENS S1	
I0.1	SENS S2	
I0.2	SENS S3	
I0.3	SENS S4	
I0.5	CIL 1 H	
I0.6	CIL 1 L	
I0.7	CIL 2 H	
I1.0	CIL 2 L	
I1.1	CIL 3 H	
I1.2	CIL 3 L	
Q0.0	MOT M1	
Q0.1	MOT M2	
Q0.2	1 CILINDR	
Q0.3	2 CILINDR	
Q0.4	3 CILINDR	
M20	TEMP 10S	

Inserir todos | Apagar linha | Imprimir | Fechar

Figura 6.42
Mapeamento das entradas e saídas do projeto de transportadora versão 2.

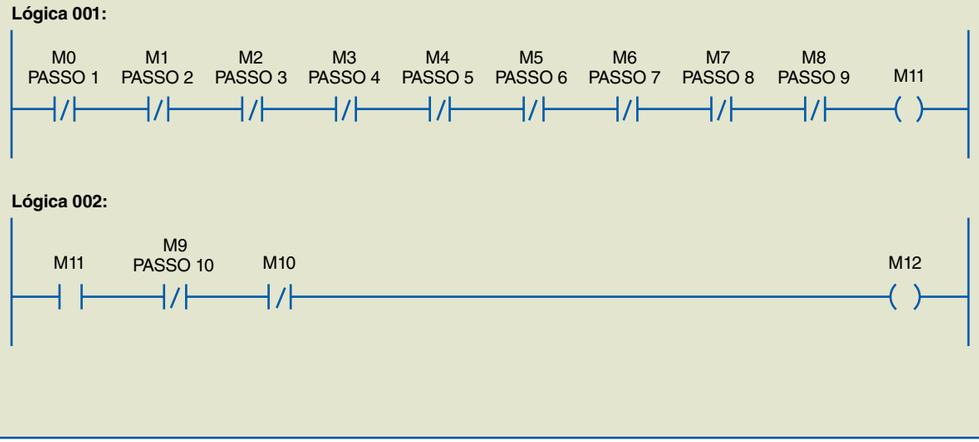
Resumo dos 10 passos e dos respectivos eventos de transição para o projeto transportadora versão 2:

10 – Aguarda sensor S1.
1 – Liga motor M1 e aguarda sensor S2.
2 – Desliga motor M1, avança cilindro CIL1 e aguarda cilindro CIL1 avançado.
3 – Mantém avanço de cilindro CIL1, avança cilindro CIL2, aguarda cilindro CIL2 avançado e aguarda sensor S3.
4 – Liga motor M2 e aguarda ausência de sensor S3.
5 – Desliga motor M2, recua cilindro CIL1, recua cilindro CIL2, aguarda cilindro CIL1 recuado e aguarda cilindro CIL2 recuado.
6 – Liga motor M2 e aguarda sensor S4.
7 – Desliga motor M2, avança cilindro CIL3, aguarda cilindro CIL3 avançado e aguarda ausência de sensor S4.
8 – Espera 10 segundos.
9 – Recua cilindro CIL3 e aguarda cilindro CIL3 recuado.
10 – Aguarda sensor S1 (e volta para 1 início).

Com essas informações, pode-se desenvolver o programa em Ladder utilizando como auxiliar o diagrama de funcionamento do sistema. O programa atende ao solicitado, de maneira que se observam os detalhes descritos a seguir.

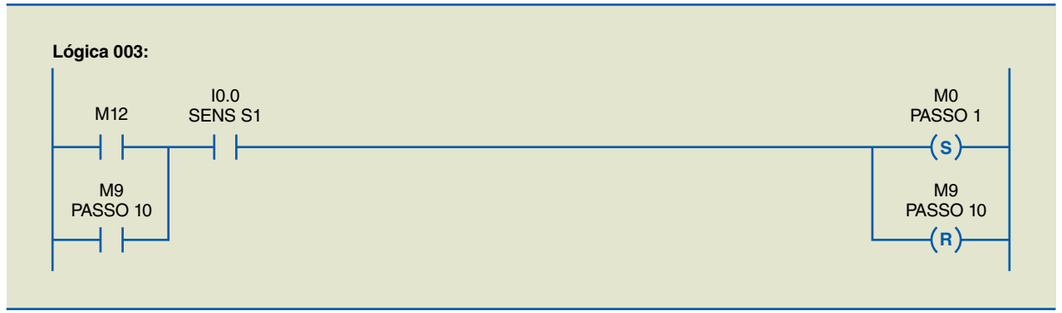
Lógicas 1 e 2 – Definem que, se nenhum passo está ativo, o primeiro passo do processo é o que deve entrar em execução. O que determina isso é a saída da memória (M12). Na figura 6.43 estão representadas duas linhas para essa função. É preciso ficar atento, pois, se o número de passos for superior ao número de instruções-limite por linha de lógica, será possível cascatear todas as instruções de uma primeira linha armazenando seu resultado em uma memória. Nessas condições, em uma segunda linha, adiciona-se essa memória como pré-requisito para o acionamento da segunda.

Figura 6.43
Lógicas 1 e 2.



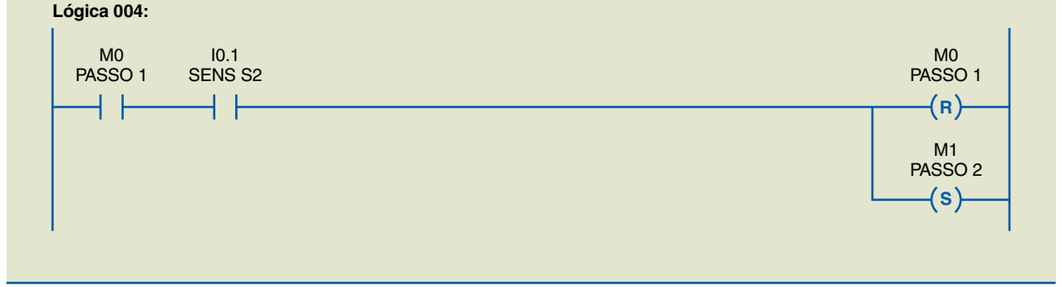
Lógica 3 – A transição para o passo 1 ocorrerá com S1 acionado, descartando a possibilidade de o passo 10 ser executado, desde que a memória (M12) de início do processo esteja ativa ou que o passo 10 tenha sido executado (figura 6.44).

Figura 6.44
Lógica 3.



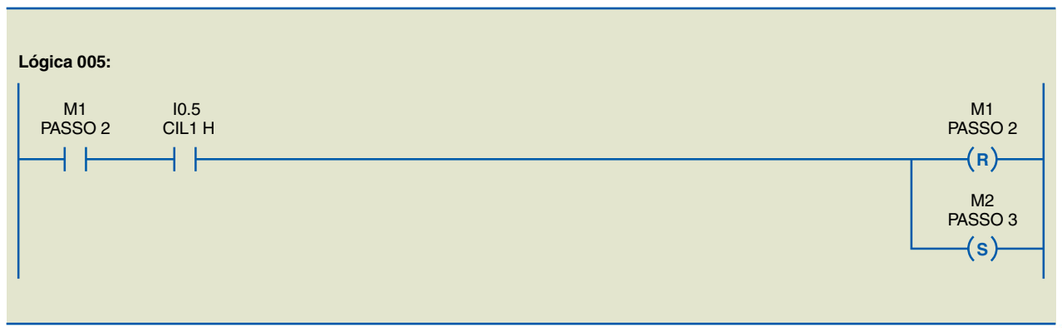
Lógica 4 – A transição para o passo 2 ocorrerá com S2 acionado, desde que o passo 1 esteja ativo (figura 6.45). Isso descartará ao final a possibilidade de execução do passo 2.

Figura 6.45
Lógica 4.



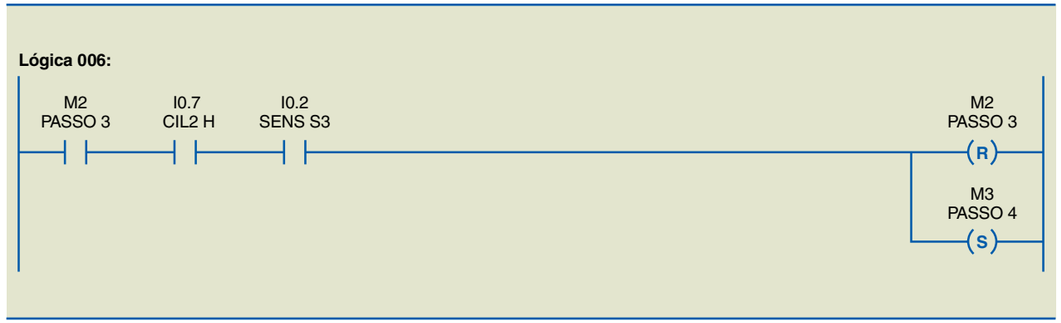
Lógica 5 – A transição para o passo 3 ocorrerá com CIL1 H acionado, desde que o passo 2 esteja ativo (figura 6.46). Isso descartará ao final a possibilidade de execução do passo 3.

Figura 6.46
Lógica 5.



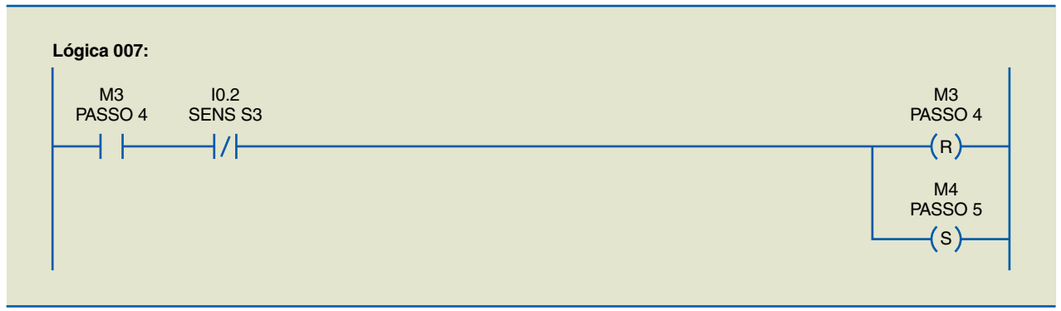
Lógica 6 – A transição para o passo 4 ocorrerá com CIL2 H e S3 acionados, desde que o passo 3 esteja ativo (figura 6.47). Isso descartará ao final a possibilidade de execução do passo 4.

Figura 6.47
Lógica 6.



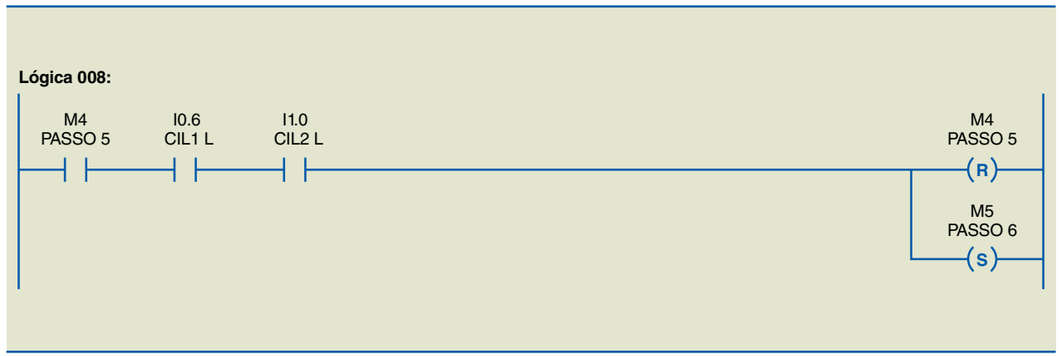
Lógica 7 – A transição para o passo 5 ocorrerá com S3 inativo, desde que o passo 4 esteja ativo (figura 6.48). Isso descartará ao final a possibilidade de execução do passo 5.

Figura 6.48
Lógica 7.



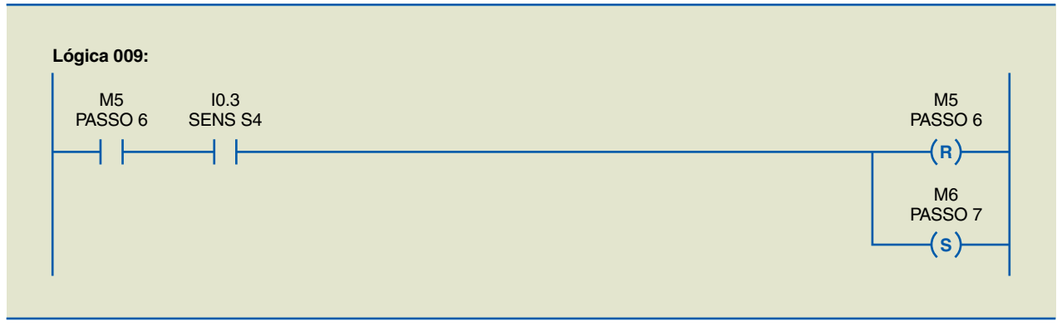
Lógica 8 – A transição para o passo 6 ocorrerá com CIL1 L e CIL2 L acionados, desde que o passo 5 esteja ativo (figura 6.49). Isso descartará ao final a possibilidade de execução do passo 6.

Figura 6.49
Lógica 8.



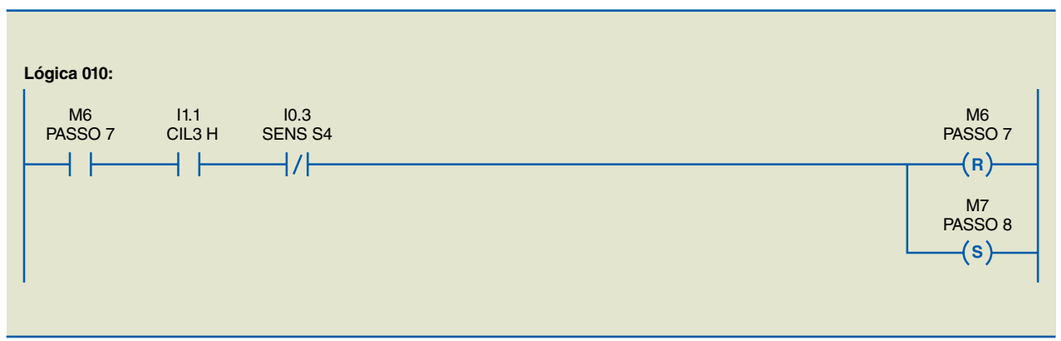
Lógica 9 – A transição para o passo 7 ocorrerá com S4 acionado, desde que o passo 6 esteja ativo (figura 6.50). Isso descartará ao final a possibilidade de execução do passo 7.

Figura 6.50
Lógica 9.



Lógica 10 – A transição para o passo 8 ocorrerá com CIL3 H ativo e S4 inativo, desde que o passo 7 esteja ativo (figura 6.51). Isso descartará ao final a possibilidade de execução do passo 8.

Figura 6.51
Lógica 10.



Lógica 11 – A transição para o passo 9 ocorrerá com TEMP 10S acionado, desde que o passo 8 esteja ativo (figura 6.52). Isso descartará ao final a possibilidade de execução do passo 9.

Figura 6.52
Lógica 11.

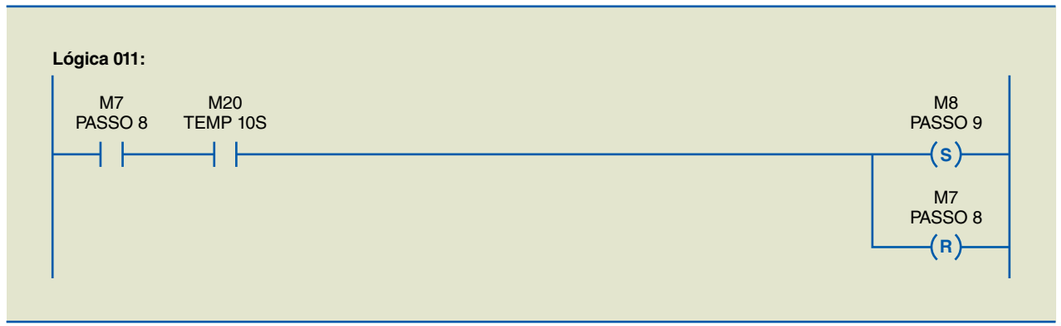


Figura 6.53
Lógica 12.

Lógica 12 – A transição para o passo 10 ocorrerá com CIL3 L acionado, desde que o passo 9 esteja ativo (figura 6.53). Isso descartará ao final a possibilidade de execução do passo 10 e permitirá que o processo seja reiniciado na lógica 3.

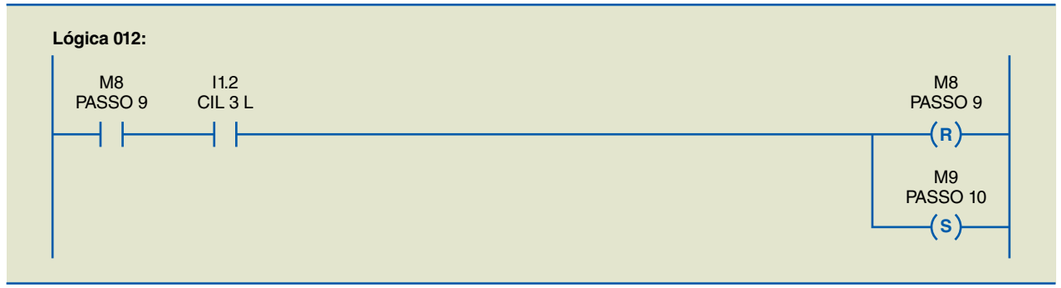


Figura 6.54
Lógica 13.

Lógica 13 – O passo 1 consiste em acionar o motor M1 (figura 6.54).

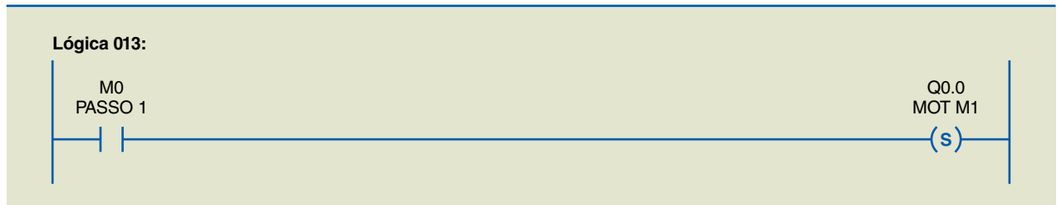


Figura 6.55
Lógica 14.

Lógica 14 – O passo 2 consiste em acionar o cilindro 1 e desligar o motor M1 (figura 6.55).

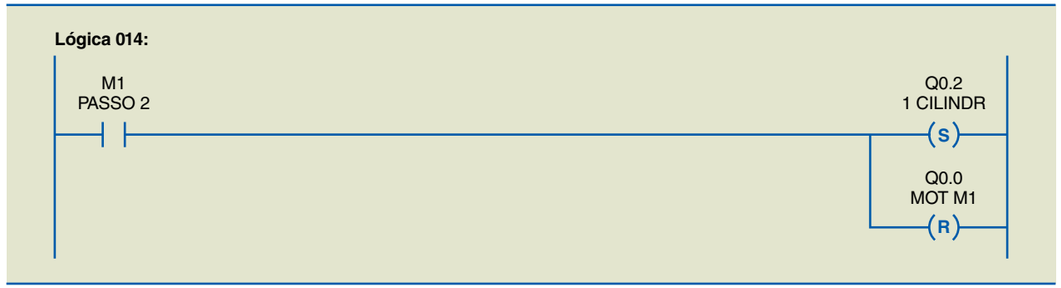


Figura 6.56
Lógica 15.

Lógica 15 – O passo 3 consiste em acionar o cilindro 2 (figura 6.56).

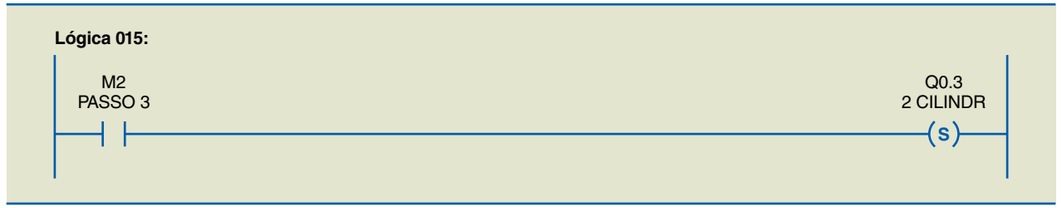


Figura 6.57
Lógica 16.

Lógica 16 – O passo 4 ou 6 consiste em acionar o motor M2 (figura 6.57).

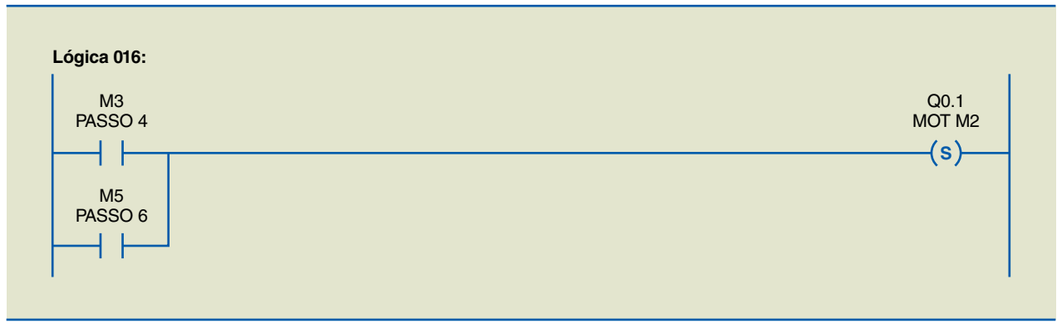


Figura 6.58
Lógica 17.

Lógica 17 – O passo 5 ou 7 consiste em desligar o motor M2, recuar o cilindro 1 e recuar o cilindro 2 (figura 6.58).

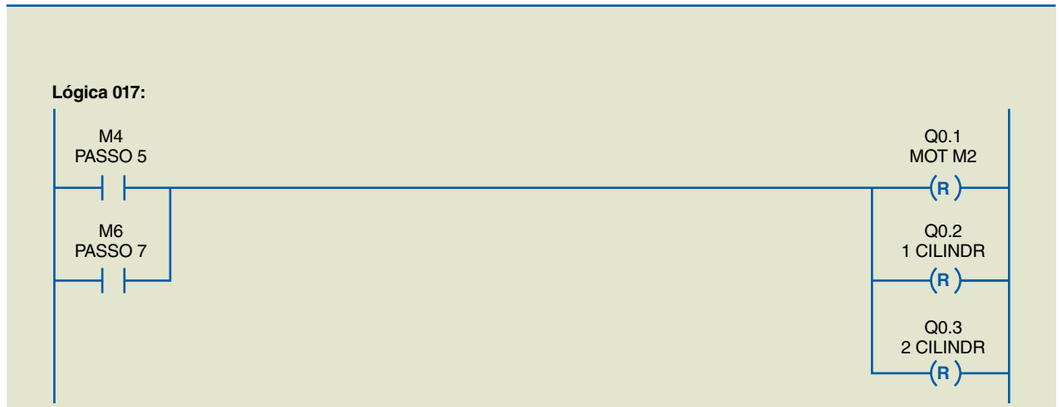


Figura 6.59
Lógica 18.

Lógica 18 – O passo 7 consiste em acionar o cilindro 3 (figura 6.59).