

4. ACIONAMENTOS VIA LÓGICA PROGRAMÁVEL

No final da década de 1960, a General Motors estava interessada na aplicação de computadores para a substituição de painéis de seqüenciamento a relés para o controle de linhas de montagem de veículos. Como resposta a esta demanda, em 1969 duas empresas (Modicon e Allen Bradley) produziram os primeiros controladores industriais de lógica programável para esta tarefa (CLPs ou PLCs em inglês). Desde então esta tecnologia evoluiu e ganhou bastante destaque nos projetos de automação.

Atualmente estes dispositivos empregam recursos linguagens de programação de alto nível e possuem alta confiabilidade operacional. Como resultado, seu uso simplifica o projeto, a manutenção e as alterações de quadros e painéis elétricos, pois toda a fiação de comando (que passa a ser realizado por software) fica resumida ao conjunto de entradas e saídas.

Estes dispositivos podem:

- realizar tarefas via acionamento de relés,
- realizar contagens, cálculos e comparações de variáveis analógicas,
- realizar alterações de lógicas de forma rápida,
- responder a estímulos externos em frações de segundos,
- comunicar-se em redes de dados,
- operar com auxílio de painéis de interface homem-máquina.

A chamada CPU com processamento eletrônico (microprocessador) que realiza a função de seqüenciamento de etapas, medição de estados de sensores de campo e ativações dos atuadores é projetada para operar em ambientes industriais agressivos. Tal dispositivo conecta-se à planta industrial (equipamentos produtivos) via cartões de entrada (aquisição) e de saída (atuação) de dados alocados em *racks* ou bastidores, de forma a se contar com uma estrutura modular e expansível segundo a demanda do projeto. Os tipos de cartão eram essencialmente quatro:

- cartões de entradas digitais,
- cartões de saídas digitais,
- cartões de entradas analógicas,

- cartões de saídas analógicas.

Cada cartão é fabricado com certo número de canais (entradas ou saídas – E/S), e pode ser empregado de forma a compor determinado número total de E/S para a unidade de processamento de forma escalonável (“ampliável”).

Na unidade de processamento eletrônico central do dispositivo, dotada de um microprocessador, então, são executadas as funções de seqüenciamento e de intertravamento tipicamente programáveis nos painéis de lógica a relés, conforme apresentado no capítulo anterior.

A figura adiante representa a estrutura de controladores industriais de lógica programável. Nesta ilustração, identifica-se uma unidade de processamento central, onde está localizado o microprocessador responsável por realizar a lógica de seqüenciamento e intertravamento do processo automatizado, 8 canais de entrada digital (IN0...IN7), alocados em um único cartão de entradas digitais (1-1IC1), 8 canais de saída digital (OUT0...OUT7), alocados em um único cartão de saídas digitais (1-1OC1), alguns dispositivos de entrada (chaves e sensores) conectados às entradas do controlador industrial de lógica programável e dois dispositivos de saída (bobinas solenóides) conectados ao cartão de saída. É importante notar que os cartões de entrada e saída digital possuem uma conexão denominada COM, ou “comum”, pois é comum a todos os canais do cartão e tem a função de fechar o circuito de alimentação dos dispositivos de entrada e dos dispositivos de saída respectivamente. Nesta ilustração ambos os circuitos de alimentação dos dispositivos de entrada e dos dispositivos de saída são energizados por fontes externas ao CLP de tensão contínua.

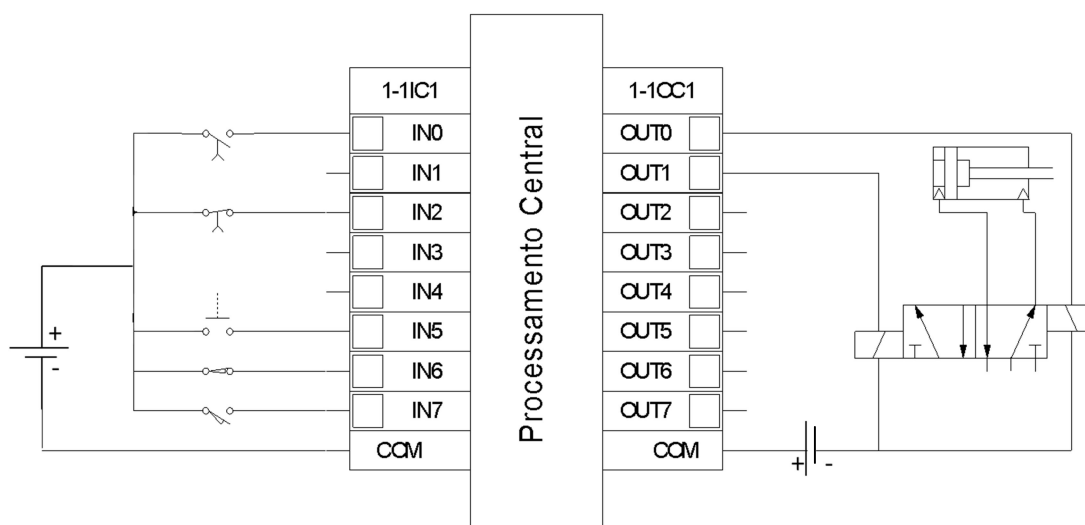


Figura 4.1. Uma estrutura com controlador por lógica programável (CLP)

1.1. ESTRUTURA DE CONTROLADORES PROGRAMÁVEIS

Tipicamente, um CLP moderno possui alguns componentes ou unidades básicas integradas (geralmente na forma de cartões) em um hardware resistente aos ambientes industriais, ou seja, imunes a temperaturas extremas, condições desfavoráveis de umidade e de poeiras, vibrações mecânicas e interferência eletromagnética:

- Unidade de processamento (CP ou CPU): cartão que contém o processador do CLP, interpreta os sinais externos e executa as rotinas de controle interno e o programa do usuário.

- Unidade fonte de energia: pode ser um cartão ou pode ser integrada à CPU, é alimentada externamente ($24V_{DC}$ ou $110/220V_{AC}$), condiciona e fornece a alimentação para os cartões do controlador: CP, módulos de comunicação, e cartões de entrada e de saída. Esta fonte de alimentação pode ainda fornecer uma conexão de saída para a eventual alimentação de dispositivos externos, por exemplo, sensores.

- Dispositivo de programação: computador ou outro tipo de dispositivo eletrônico externo utilizado pelo usuário para a elaboração do código de controle e para sua gravação no CLP. Em geral oferece recursos de monitoramento e operação do CLP (Figura 4.2).

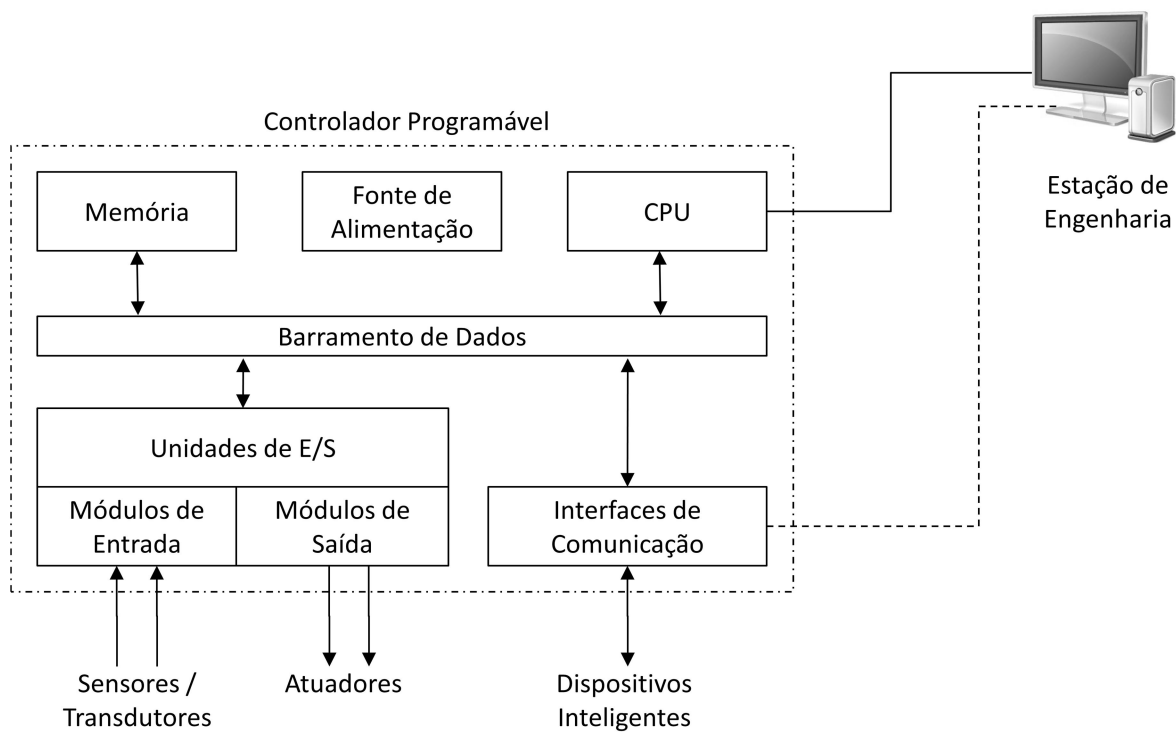


Figura 4.2. O sistema de módulos do CLP

- Memória: interna ao cartão CPU, utilizada para a gravação do sistema operacional do CLP (*firmware*), do programa do usuário e de todas as variáveis internas do controlador.

- Interfaces ou cartões de Entrada e de Saída (E/S): possuem os circuitos de condicionamento de sinais externos, de forma a interligar o CLP aos sensores e atuadores. São fornecidos em cartões que podem ser adicionados ao CLP, ou então os canais de E/S são fixos e já integrados à CPU nos chamados CLPs compactos.

- Interface de comunicação: cartão de rede ou porta de comunicação na CPU responsável por permitir a conexão do CLP com outros dispositivos de automação, com sistemas de supervisão ou com o dispositivo programador do CLP via uma rede digital de dados. As interfaces de comunicação seguem protocolos específicos de comunicação digital de “chão de fábrica”, abertos ou proprietários. A conexão típica de CLPs com sistemas de supervisão via redes digitais é apresentada na figura 4.3 a seguir.

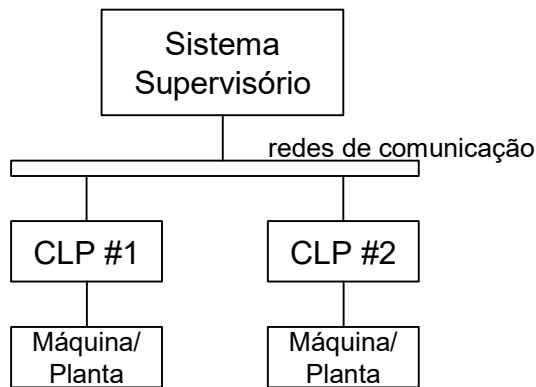


Figura 4.3. Conexão entre CLPs e sistema supervisório

Os CLPs são em geral dispositivos modulares de estrutura expansível, conforme a figura a seguir. O mecanismo utilizado para permitir a expansão do CLP em cartões ou módulos é através do uso de bastidores ou *racks* com capacidade de conexão (alimentação e comunicação) para um número determinado de módulos ou cartões (CPU, fonte, E/S). Ao se esgotar a capacidade de um rack, expande-se este através da conexão de um novo *rack* ao mesmo CLP.

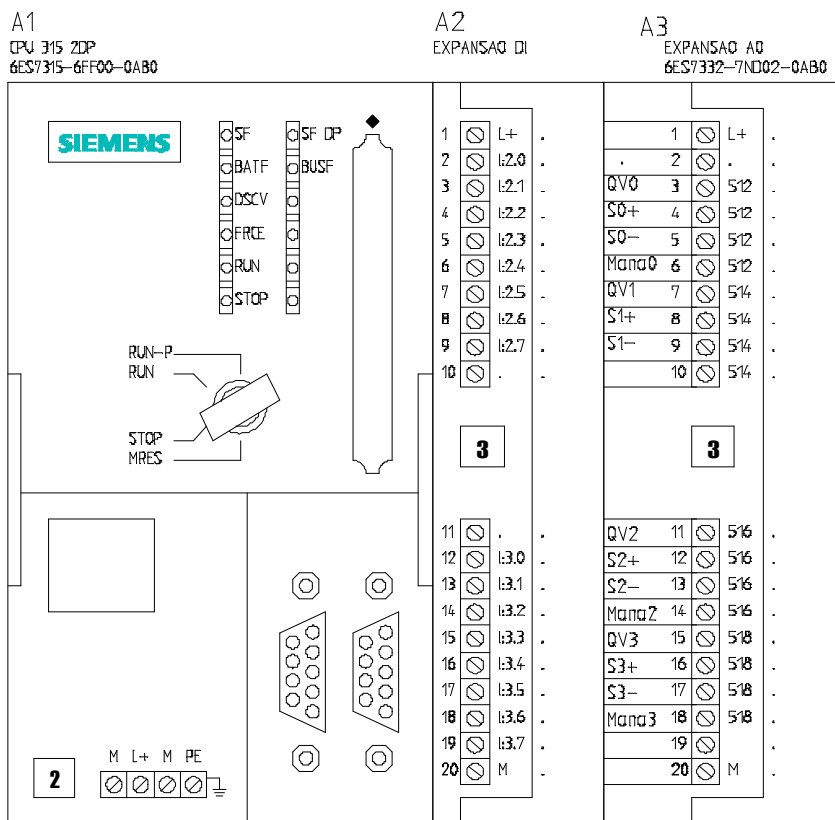


Figura 4.4. Exemplo de CLP modular

Quando não são modulares e expansíveis, o CLP é dito compacto, como na ilustração a seguir.

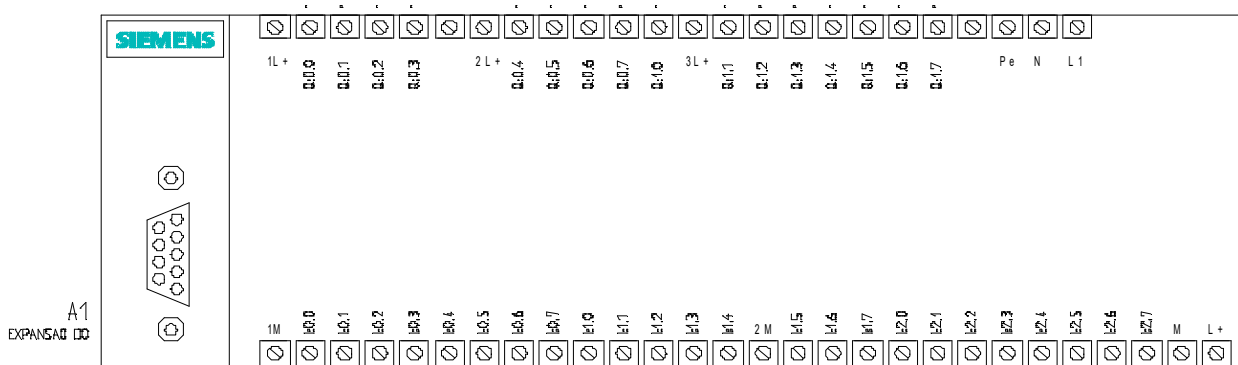


Figura 4.5. Exemplo de CLP compacto (não modular)

Internamente, a CP controla todos os módulos do CLP de acordo com uma arquitetura típica de sistemas microprocessados, com acesso a memórias e a dispositivos periféricos de acordo com um *clock* de operação (alguns MHz). Este clock determina a velocidade de operação do CLP e provê a temporização e o sincronismo temporal entre os módulos do equipamento. As trocas de informações entre os elementos do CLP são realizadas por meio de barramentos de dados, de endereços, de controle e de I/O, conforme a figura a seguir.

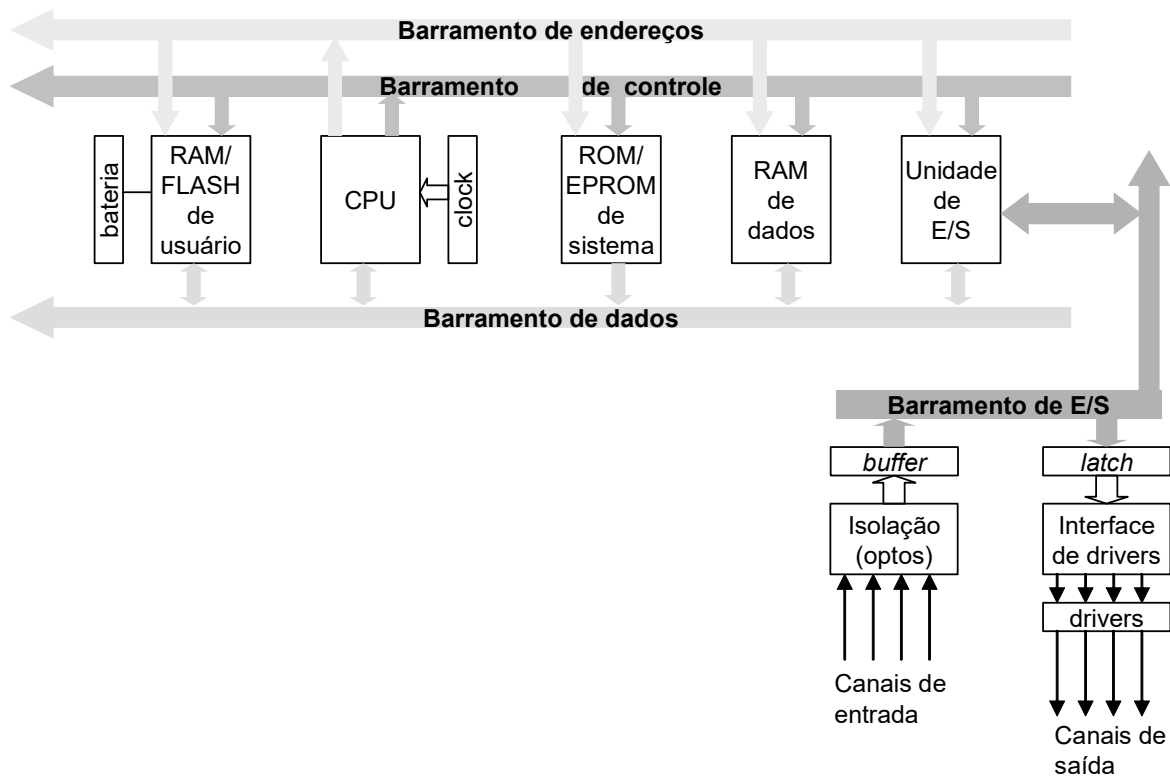


Figura 4.6. Arquitetura interna do CLP

Na arquitetura apresentada, nota-se que existem tipos diferentes de memórias para diferentes finalidades, tais como memória para o sistema operacional do CLP, ao qual o usuário não tem acesso; memória não volátil para o registro do programa de controle do usuário e variáveis importantes do programa ditas não voláteis e memória de registro dos status e valores de todas as entradas e saídas do CLP, bem como o registro de variáveis internas de diferentes tipos e de *timers* e contadores internos do CLP.

Tipicamente, os CLPs são classificados como pequenos quando usados com até 500 pontos de entradas ou saídas (E/S) digitais e analógicas, médios para número de E/S entre 500 e 5000 e grandes para sistemas com mais de 5000 pontos de E/S.

Tipos de Cartões de Entrada

A tensão de operação de circuitos eletrônicos é de $5V_{DC}$ geralmente, ao passo que a tensão de operação de dispositivos industriais (solenóides, contadores, chaves de fim de curso, etc.) pode ser de até $220V_{AC}$. Para que a junção destes dois níveis de tensão nos cartões de E/S do CLP não cause danos e interferências à operação do circuito eletrônico de processamento central, a alimentação do CLP é mantida separada da alimentação dos equipamentos da planta, ou seja, são fontes distintas.

Adicionalmente, todos os sinais elétricos externos de E/S ao CLP são isolados galvanicamente dos circuitos internos do controlador por meio de opto-acopladores nos cartões de entrada.

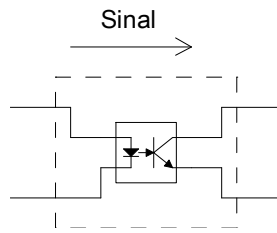


Figura 4.7. Isolação ótica de sinais de entradas

Os circuitos representativos, (não operacionais) típicos da eletrônica presente em cartões de entrada digital para sinais DC e AC em CLPs são apresentados a seguir. É importante notar que em ambos há a separação das fontes de alimentação dos dispositivos internos e externos do CLP, bem como o elemento de isolação do sinal.

Os cartões de entrada AC operam com um retificador (ponte de diodos) posicionado antes do elemento opto acoplador, conforme a ilustração adiante.

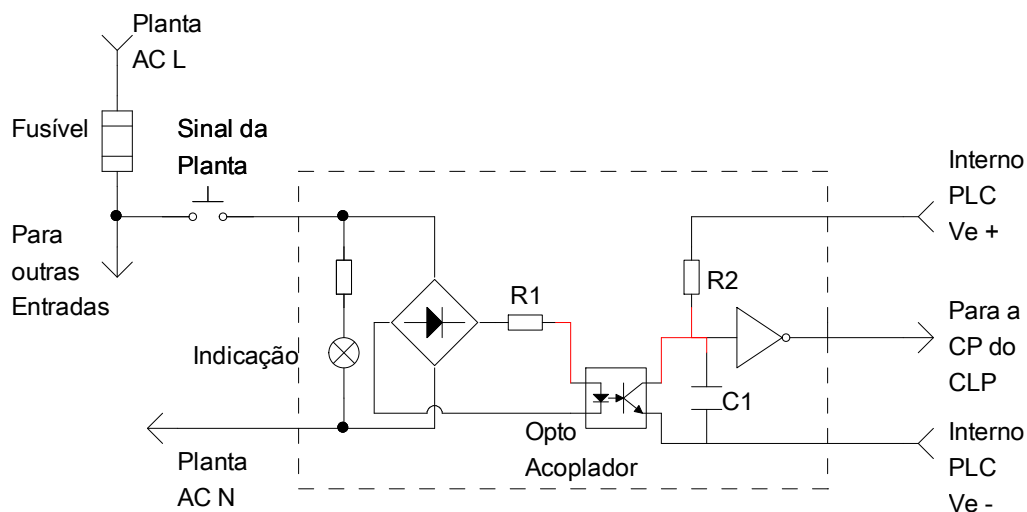


Figura 4.8. Cartão de Entrada AC

Já os cartões de entrada DC não possuem tal elemento retificador, o sinal DC polarizado da fonte externa é diretamente conduzido ao elemento opto acoplador, conforme a ilustração a seguir.

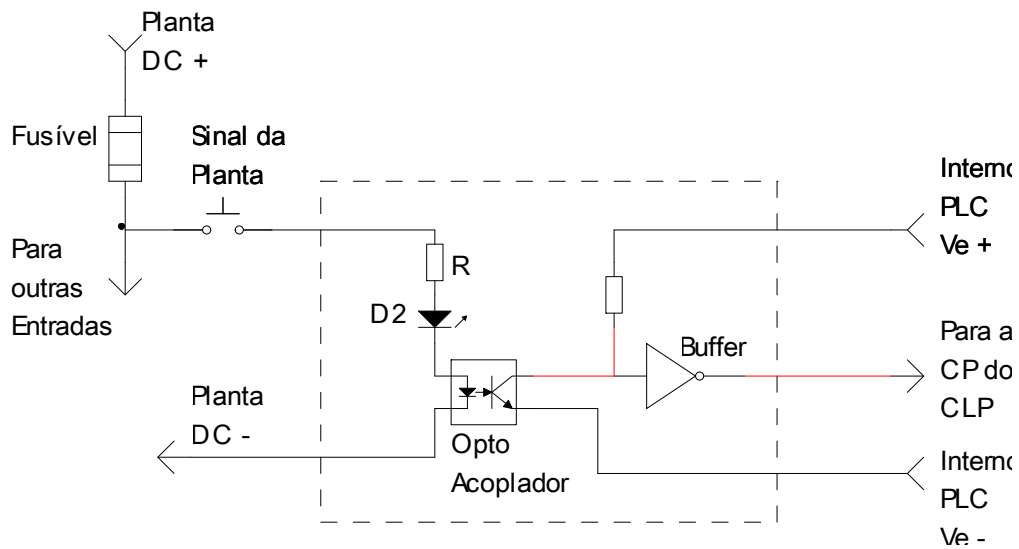
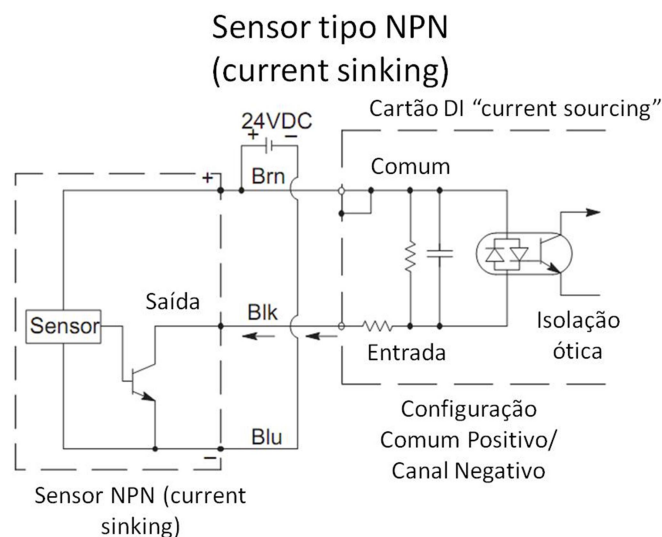


Figura 4.9. Cartão de Entrada DC

Os cartões de entrada DC podem ter configuração do tipo “current sourcing” e “current sinking”. Na configuração “current sourcing”, o canal de entrada fornece a corrente para o dispositivo externo, ou seja, o comum é ligado ao terminal positivo da fonte (comum positivo). Já no modo “current sinking”, a corrente flui do elemento externo para o canal de entrada do CLP, ou seja, o comum é conectado ao lado negativo da fonte (comum negativo). A seleção entre estas configurações deve ser realizada em função do tipo dos sensores utilizados, quando estes tiverem o estágio de saída a transistor (NPN ou PNP).

- Canal de entrada tipo N: A comutação é executada quando um dispositivo externo fornece 0 Vcc à entrada digital. Portanto, o comum das cargas deve estar ligado ao potencial de +24 Vcc da fonte de alimentação (comum positivo – cartões “current sourcing”).



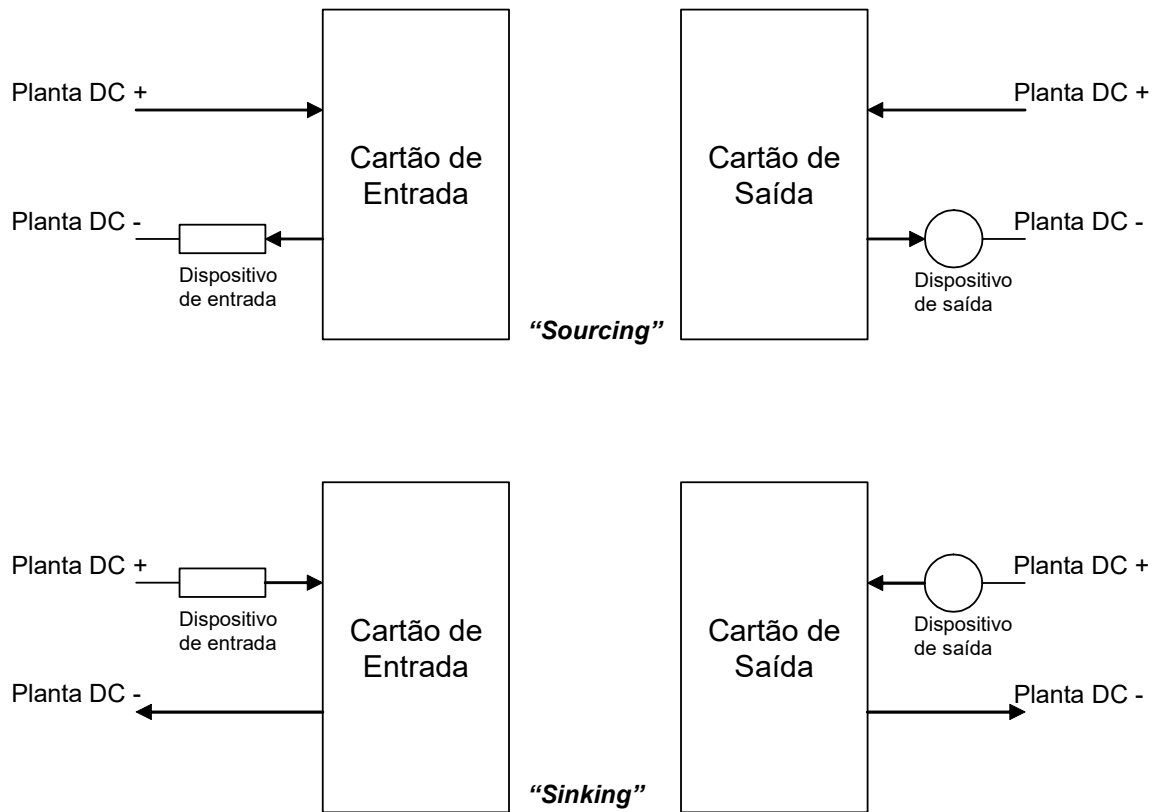


Figura 4.12. Configuração das conexões entre sensores e atuadores a cartões de entrada e de saída dos tipos "current sourcing" e "current sinking"

Em desenhos de engenharia, representam-se, de maneira simplificada as conexões de sensores e demais dispositivos de entrada a um cartão de entrada de CLP da maneira como indicam as ilustrações a seguir.

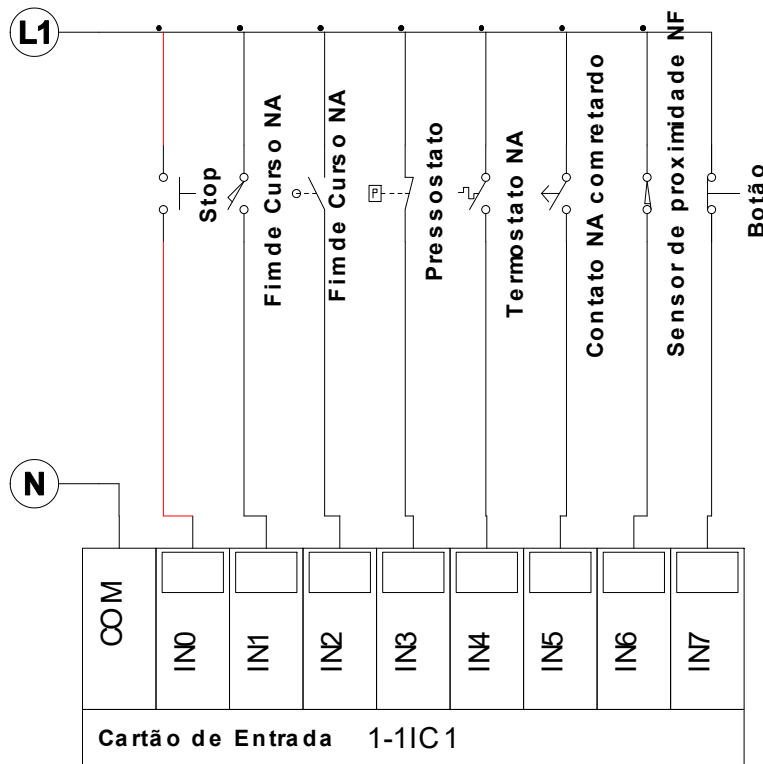


Figura 4.13. Representação simplificada de um cartão de entrada AC

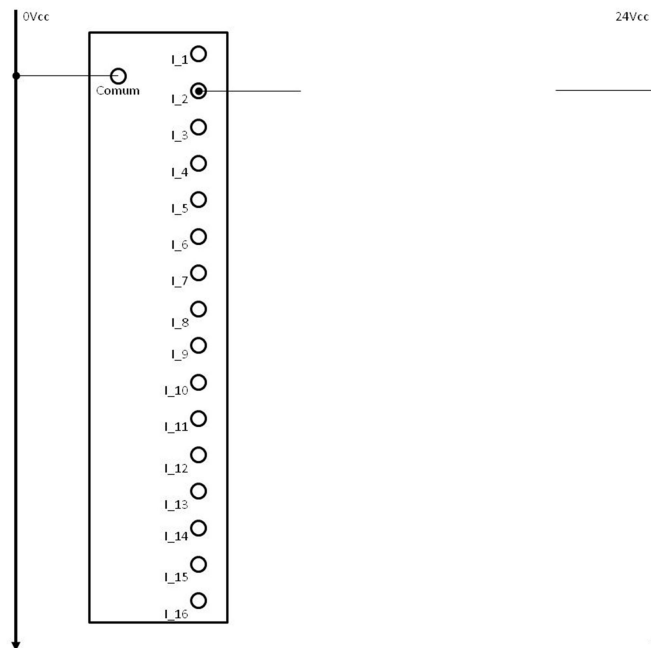


Figura 4.14. Representação simplificada de um cartão de entrada: alimentação e conexão com sensores e chaves

Tipos de Cartões de Saída

Para cartões de saída de CLPs, novamente é necessária a isolamento dos sinais internos do circuito eletrônico do CLP em relação aos circuitos externos. Interferências podem ser um problema mais crítico nos cartões de saída, visto que é comum se controlar o fluxo de altas correntes por cargas indutivas (ex. bobinas).

Os cartões de saída são usualmente de dois tipos: com alimentação comum a um grupo de canais ou com alimentação independente e isolada por canal. Nas figuras a seguir, esta diferença é evidenciada para cartões com saídas chaveadas por relés. Os cartões de saída com chaveamento a relés são comuns em todas as famílias e/ou modelos de diversos fabricantes de CLPs. Possuem capacidade de chaveamento relativamente alta (segundo especificação técnica de cada produto) em corrente AC ou DC indistintamente e possuem tempo de resposta lento e são sujeitos a desgaste mecânico nos elementos móveis dos relés.

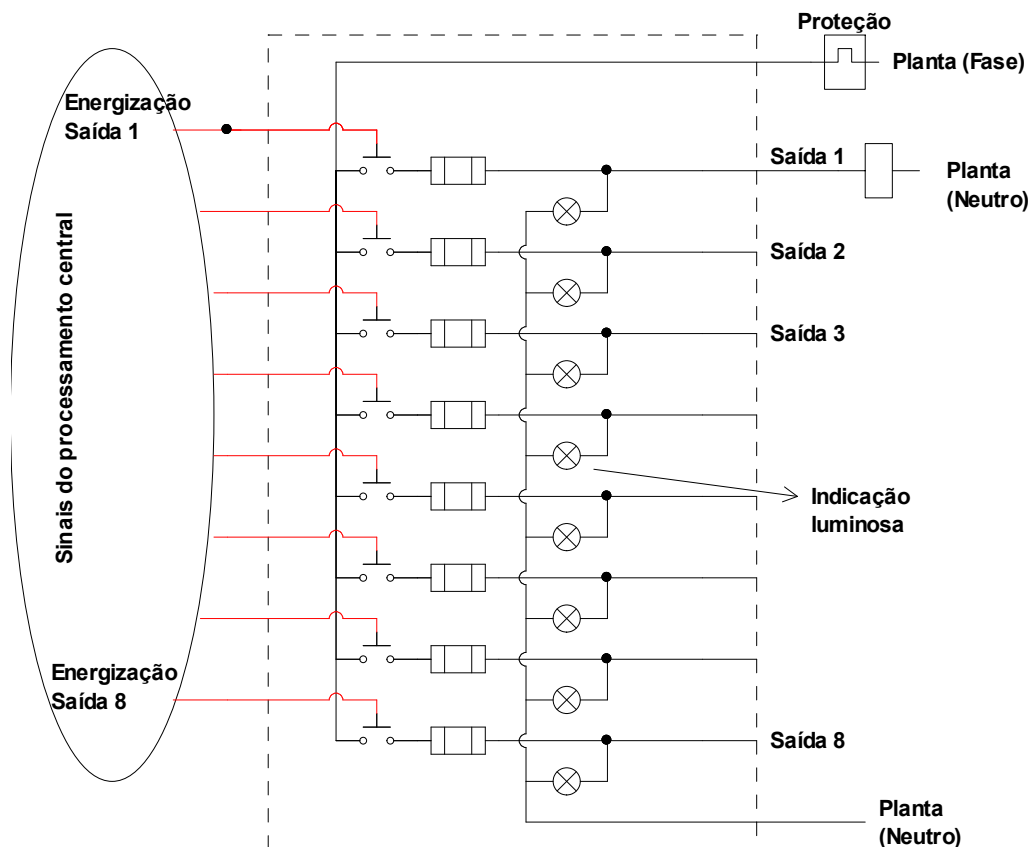


Figura 4.15. Representação de um cartão de saída a relés com alimentação comum

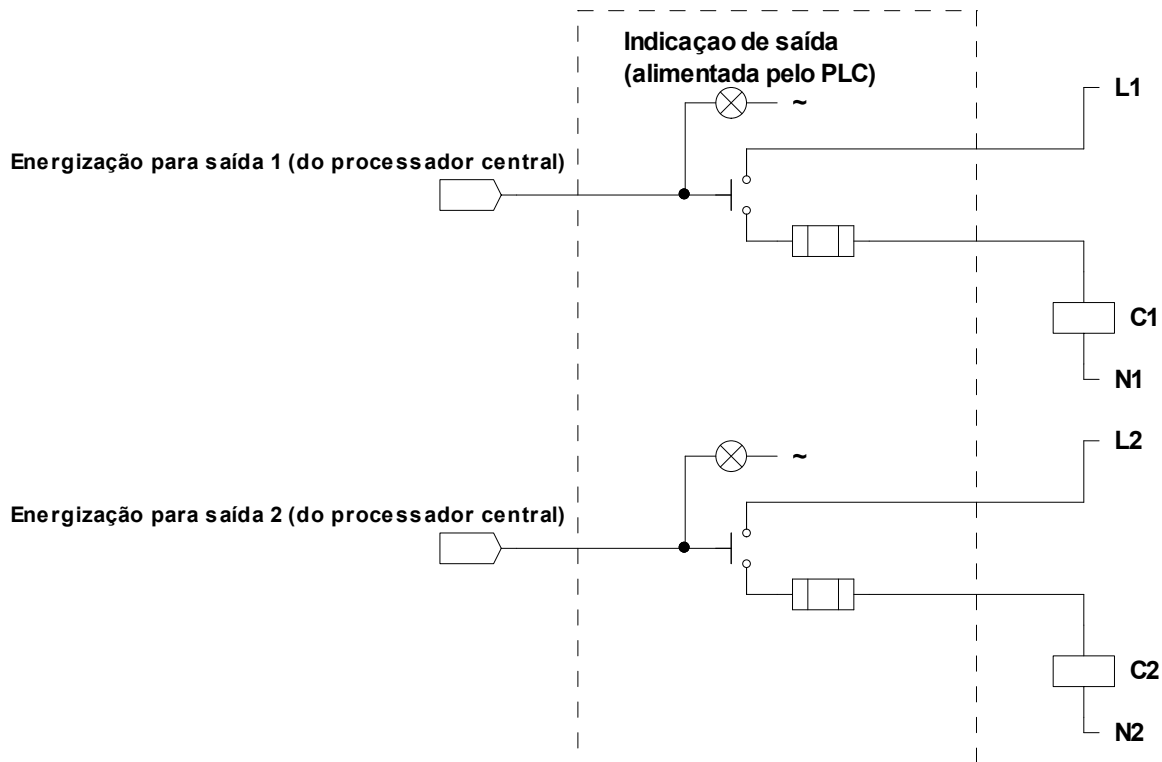


Figura 4.16. Representação de um cartão de saída a relés com alimentações separadas

Além dos cartões de saída a relés, existem também os cartões de saída baseados em transistores, conforme apresentados a seguir. Este tipo de cartão deve ser restrito a cargas DC, a polaridade na conexão com o processo deve ser observada conforme a configuração "current sourcing" ou "current sinking" explanada anteriormente. Possuem vida útil elevada e baixos tempos de resposta, todavia em geral chaveiam menores quantidades de corrente do que os relés.

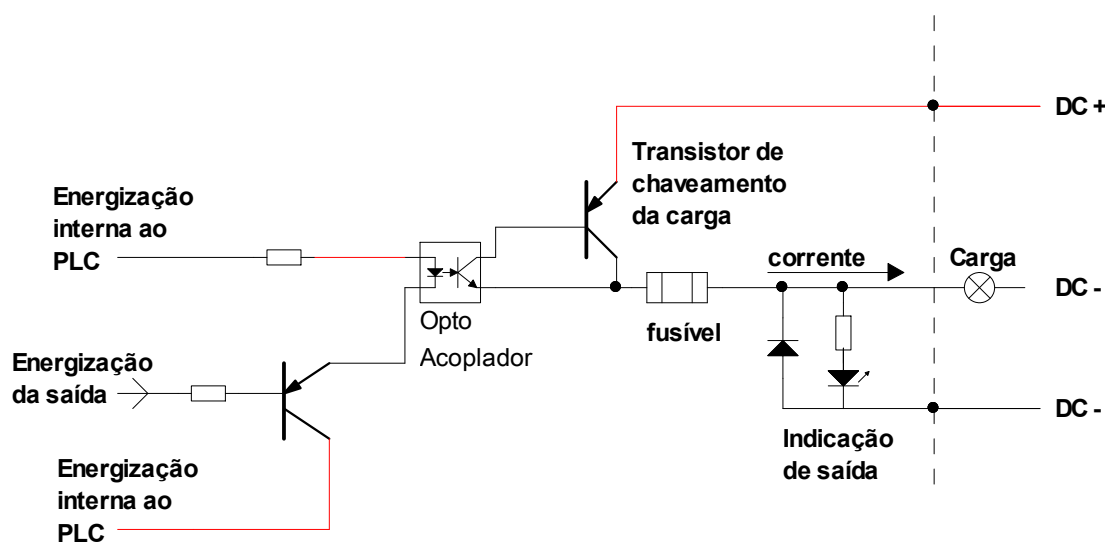


Figura 4.17. Representação de um cartão de saída a transistor para alimentação externa DC

Uma proteção externa ao CLP pode ser aplicada em circuitos de acionamento de cargas indutivas, protegendo-se desta forma o circuito de chaveamento interno no cartão de saída. Uma forma de realizar tal proteção é através da ligação de um diodo em paralelo com a carga indutiva, conforme a figura a seguir.

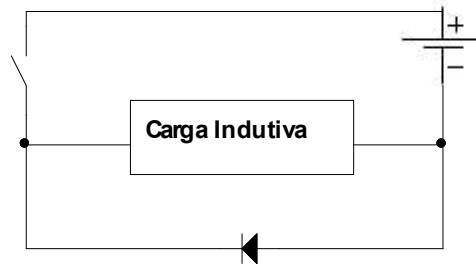


Figura 4.18. Diodo de supressão de pico ou arco em cargas indutivas

Nota-se que o diodo em paralelo com a Indicação de saída na figura 4.18 tem a função de supressão de picos de corrente no chaveamento da energização da carga quando esta tem natureza indutiva. A figura a seguir representa o efeito da operação deste supressor de pico para a proteção do equipamento.

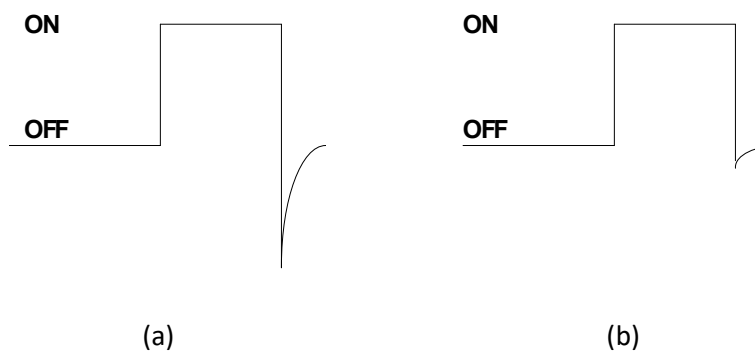


Figura 4.19. Efeito do diodo de supressão de pico de corrente em (b) versus sem a aplicação do mesmo em (a)

No diagrama da figura 4.17, o transistor utilizado no cartão do CLP é do tipo PNP, neste caso a saída é dita do tipo “*current sourcing*” ou o canal é dito positivo (P), pois a carga é conectada ao CLP pela sua conexão positiva (A1 para bobinas).

Caso se utilize um transistor do tipo NPN, então a saída seria do tipo “*current sinking*” ou o canal é dito negativo (N), pois a carga é conectada com CLP pela sua conexão negativa (A2 para bobinas), conforme a figura a seguir.

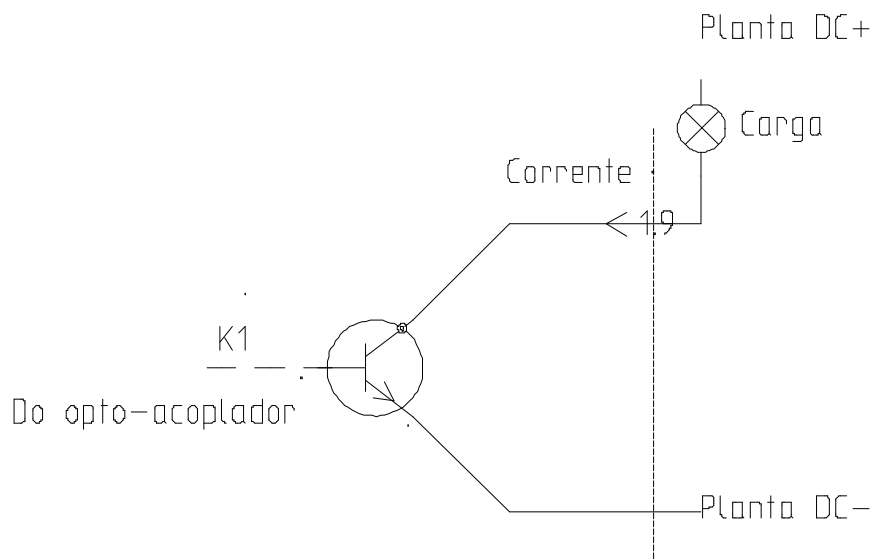


Figura 4.20. Saída “*current sinking*”

Cartões de saída a transistor para cargas AC utilizam Triacs, conforme a figura a seguir. Neste tipo de saída a denominação “source” ou “sinking” é irrelevante, pois a corrente flui em ambas as direções.

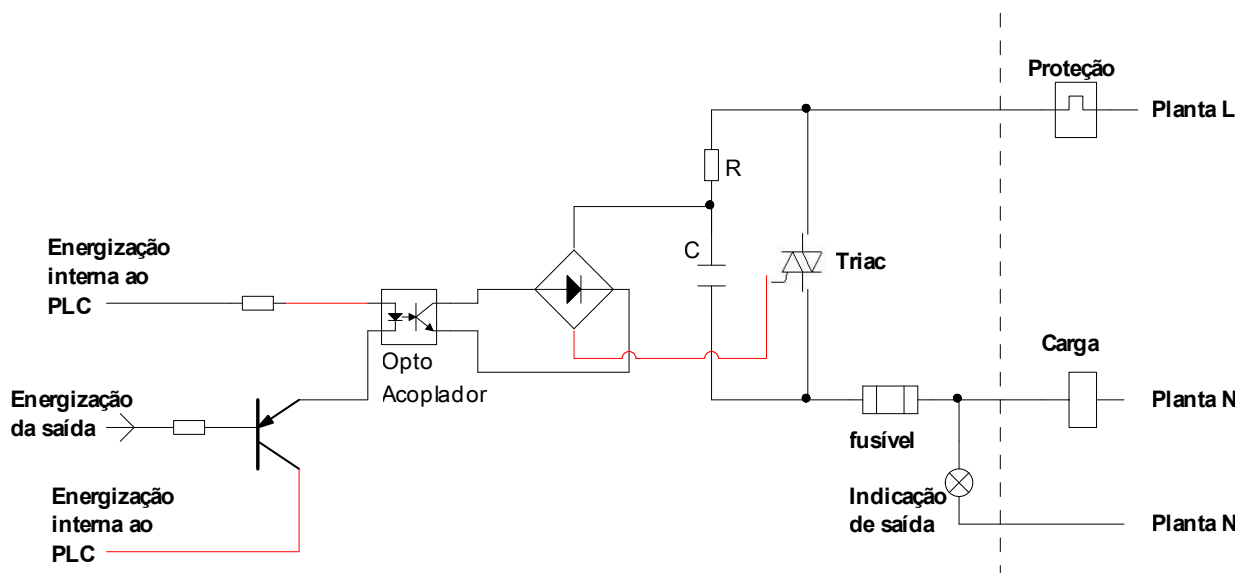


Figura 4.21. Cartão de saída AC a Triac

Com este tipo de dispositivo, pela natureza do elemento triac, a corrente de saída é levada a zero somente quando a corrente na carga atinge o valor zero na onda senoidal, eliminando a interferência quando o chaveamento é realizado em uma carga indutiva. Este efeito é representado na figura a seguir.

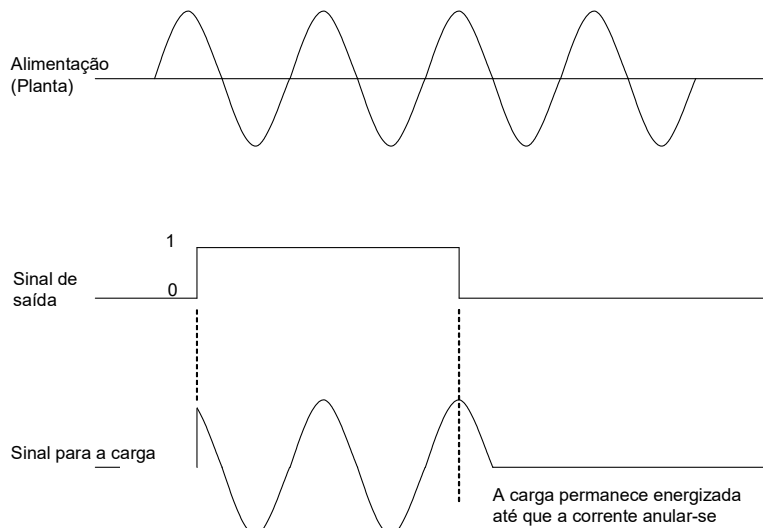


Figura 4.22. Efeito da saída a Triac

A representação em desenhos de engenharia de cartões de saída pode ser da seguinte forma:

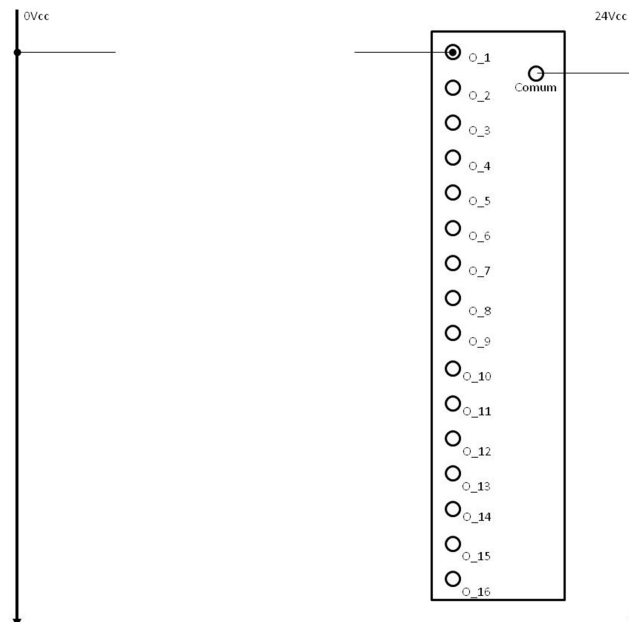


Figura 4.23. Representação simplificada de um cartão de saída: alimentação e conexão com sensores e chaves

5. PROGRAMAÇÃO E OPERAÇÃO DE CLPS

Um programa de controle executado em um CLP em modo de execução (modo RUN) basicamente pode ser considerado como em execução em “loop” ou de forma cíclica, ou seja, o programa é executado repetidamente até que o CLP saia do modo de execução para o modo de parada, ou STOP. Dentro deste padrão cíclico de execução, existe uma seqüência de tarefas, dentre as quais: a execução do programa ou da lógica de automação em si e a leitura das variáveis de entrada e a atualização das variáveis de saída, conforme a figura a seguir.

Com a adoção da norma internacional IEC 61131, o modo cíclico de operação descrito sofre modificações, conforme descrito no capítulo seguinte. Entretanto, para efeitos práticos, o modo de operação cíclica deve ser considerado como padrão.

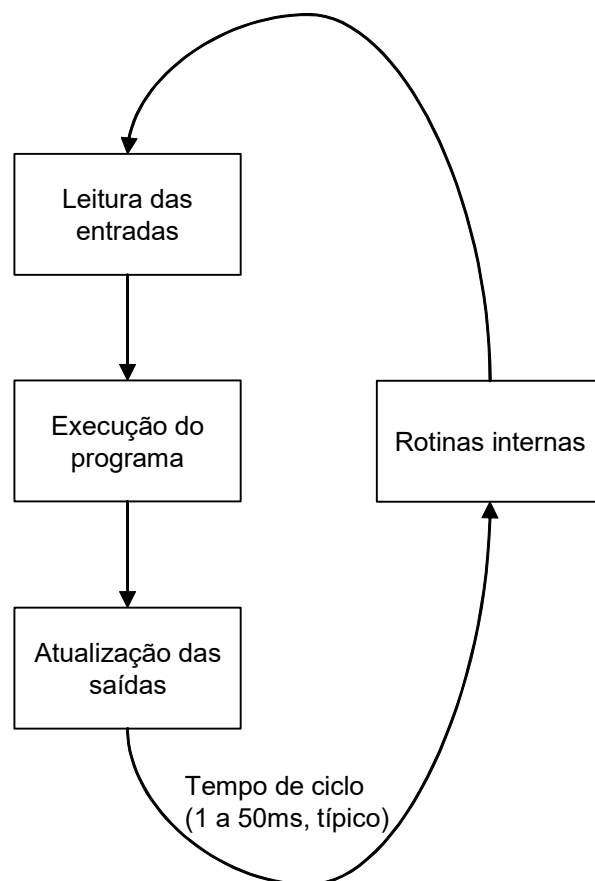


Figura 4.20. Modo de operação do CLP

Conseqüentemente, o CLP não se comunica constantemente com o a planta controlada (através dos sensores e atuadores), mas sim atualiza suas entradas e saídas periodicamente, em analogia à sucessivas “fotos” periódicas do processo. A ação cíclica de execução das tarefas do CLP no modo de operação é denominada “ciclo de *scan*”, e o período ou duração deste ciclo é chamado de tempo de *scan*, tipicamente da ordem de milissegundos. Este tempo determina o tempo de reação ou de resposta do CLP. Como os sistemas de controle industriais são sempre de tempo real, o tempo de resposta do CLP deve ser compatível com as dinâmicas dos processos ou dos equipamentos controlados. Tipicamente, segundo Astrom e Wittenmark (1990), os tempos de ciclo de *scan* que permitem o controle de variáveis de processo comum são:

Variável de Processo	Tempo de <i>scan</i> típico
Vazão	1 a 3 segundos
Pressão	1 a 5 segundos
Nível	5 a 10 segundos
Temperatura	10 a 20 segundos

Todavia, para aplicações em sistemas discretos, tais como acionamento de motores, linhas de montagens, automação de máquinas, os tempos de *scan* devem ser da ordem de milissegundos.

Neste modelo, nota-se que o CLP não lê as entradas ou atualiza as saídas sempre quando necessário, ele realiza a leitura de todas as entradas antes da execução do programa, e somente atualiza as saídas após o término da execução do programa.

O processo de acesso do programa às entradas e às saídas físicas do CLP se dá por meio de uma região especial da memória RAM de dados denominada imagem das entradas e imagem das saídas, conforme a figura a seguir.

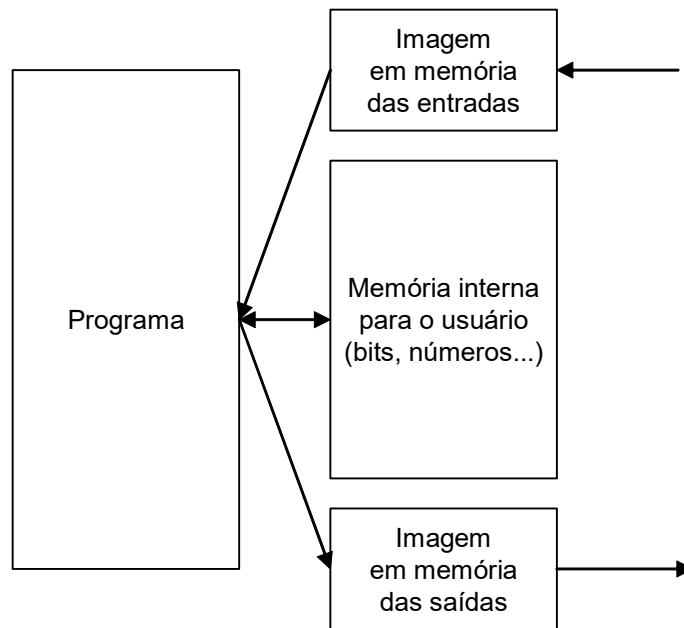


Figura 4.21. Imagem das entradas e das saídas em memória

Neste processo, em cada posição fixa de memória booleana (1 bit) mapeia-se um canal físico em um dos cartões de entrada ou de saída digital do controlador, conforme a figura a seguir.

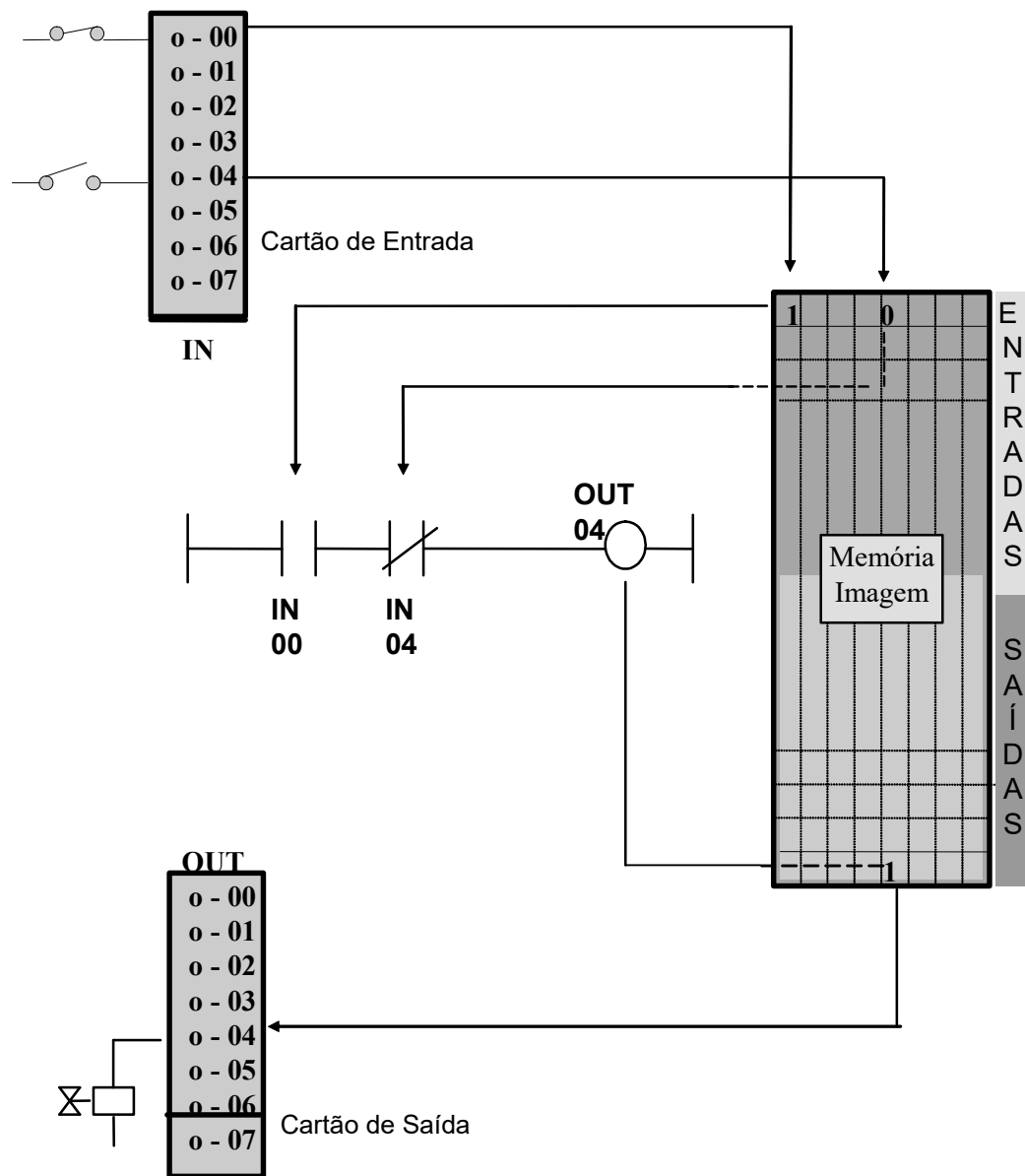


Figura 4.22. Mapeamento das entradas e das saídas em memória

No mapeamento indicado, nota-se que no programa do CLP, cada operando recebe uma descrição, ou *tag*, que indica em qual posição de memória deve-se referenciar seu valor. A linguagem de programação do programa do usuário na figura anterior é chamada "*ladder*", tradução ao inglês para escada. Esta linguagem é a mais difundida e utilizada em CLPs e guarda muita semelhança com a programação a lógica de relés. Para se entender um diagrama (programa) ladder, considere um circuito elétrico simples, conforme a figura a seguir.

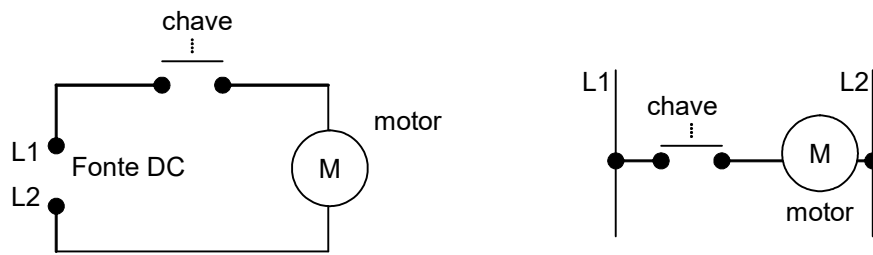


Figura 4.23. Duas formas de se representar um mesmo circuito elétrico

Na representação do circuito elétrico com duas barras de alimentação dispostas verticalmente, nota-se uma semelhança com uma escada, visto que a linha horizontal de energização do motor comandada pela chave pode ser comparada a um degrau da escada, que em um diagrama ladder é uma linha do programa. As barras verticais no ladder são as barras de energização dos circuitos, ou linhas.

Na representação de um diagrama ladder, certas convenções são adotadas:

- As barras verticais são as barras de energização, a fluxo de potência flui da barra da esquerda para a barra da direita através de uma linha horizontal.
- Cada linha define uma operação no programa de controle, comumente chamado de lógica ladder.
- Um diagrama ladder é avaliado da esquerda para a direita e de cima para baixo, conforme a figura a seguir. Deve-se lembrar que a execução do programa ladder somente acontece após a atualização de toda a imagem em memória das entradas.

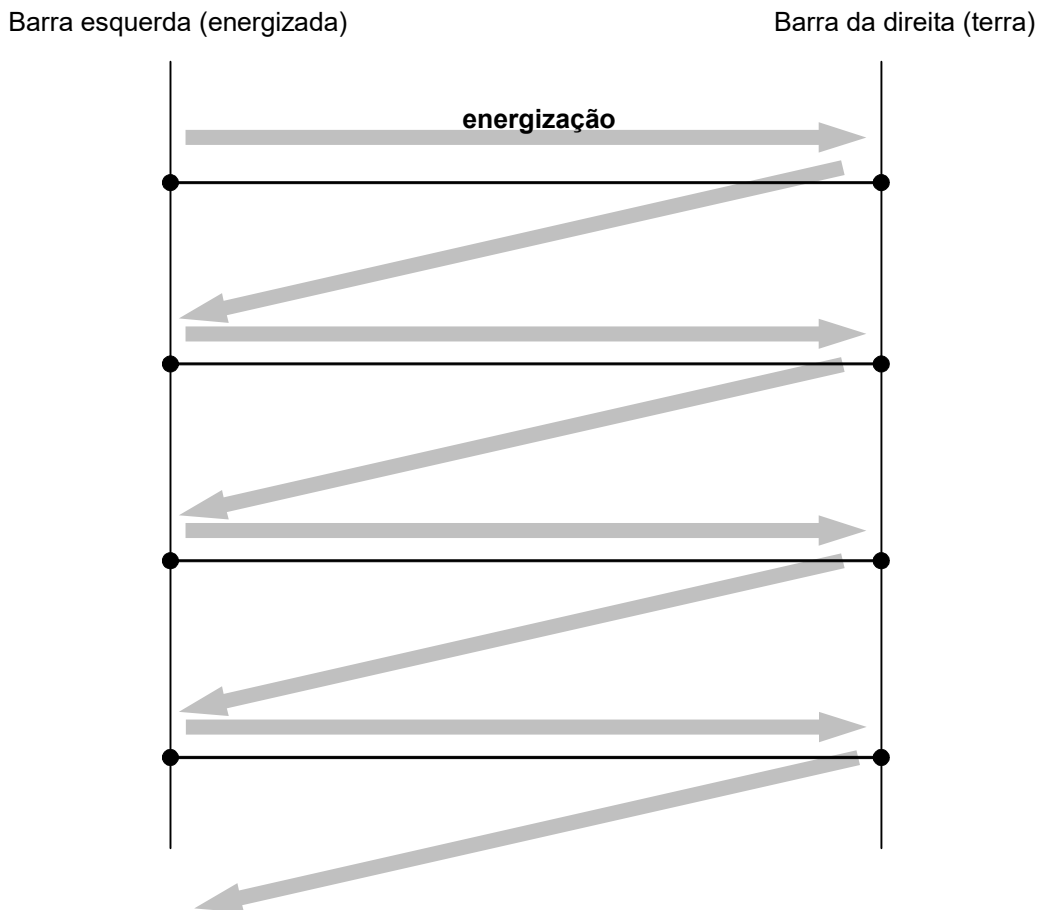


Figura 4.23. Avaliação do diagrama ou lógica ladder

Todas as linhas do programa são avaliadas da esquerda para a direita e todas as linhas são avaliadas na direção descendente até a última linha, quando o programa termina. Quando o CLP está em modo de execução, o programa é avaliado uma vez a cada ciclo de *scan*.

- Cada linha deve ser iniciada com entrada(s) e deve ser finalizada com saída(s). Os termos entrada e saída são utilizados no ladder para indicar elementos (ou operadores) do tipo contatos (de uma chave, ou sensor) e bobinas (solenóides, relés ou outros dispositivos de saída conectados ao CLP) respectivamente.

- Os operadores do ladder (entradas e saídas) são indicados em seus estados normais, ou seja, fechados ou abertos.

- Determinado operador pode aparecer em uma ou mais linhas do programa. Neste caso deve-se utilizar o mesmo nome (*label, tag*) ou descrição para todas as linhas onde o operador é empregado.

- As entradas e saídas físicas são indicadas por seus endereços na área de imagem em memória. Cada fabricante de CLP adota uma notação distinta para o endereçamento de entradas e de saídas.

- A cada ciclo de scan, nenhuma linha do programa é avaliada antes que todas as entradas do CLP tenham sido atualizadas.

- O ciclo de scan não se completa até que todas as saídas tenham sido definidas pela lógica. As saídas só mudarão de estado após todas as linhas do ladder terem sido executadas.

- A execução de um programa só estará concluída após a avaliação de todos os elementos (bobinas, blocos funcionais).

Para ilustrar o uso de lógicas ladder, considere as situações seguintes de energização de um dispositivo. Nestes diagramas apresenta-se a representação de contatos (entradas) e de bobinas (saídas) de tipo booleano.

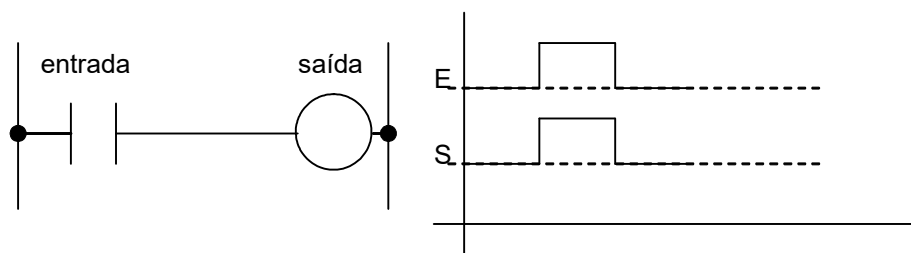


Figura 4.24. Energização de uma bobina (saída) por um contato normalmente aberto (NA) denominado entrada

Deve-se ter em mente que no caso da figura anterior, bem como em todos os casos correlatos, existe um atraso temporal “ t ” entre os sinais de entrada e de saída que é aproximadamente proporcional ao tempo de leitura do mapa de imagem de entradas e o tempo de execução do programa do usuário, conforme a ilustração a seguir.

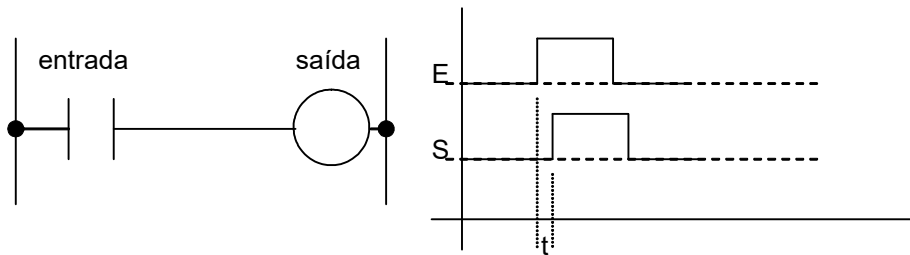


Figura 4.25. Energização de uma bobina denominada saída por um contato normalmente aberto (NA) denominado entrada

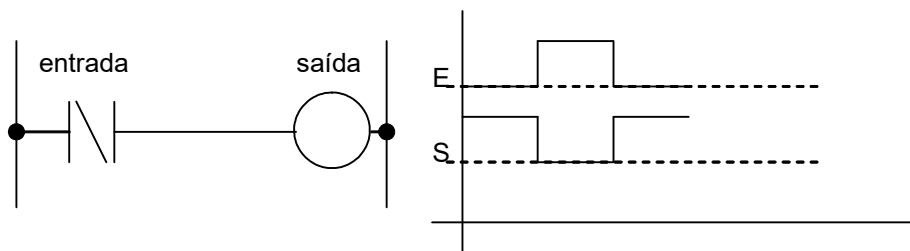


Figura 4.26. Energização de uma bobina denominada saída por um contato normalmente fechado (NF) denominado entrada

A notação de endereçamento de entradas e de saída em função dos módulos e dos canais à quais estão mapeadas, conforme mencionado anteriormente, é definida pelo fabricante do CLP. A seguir apresentam-se algumas notações comuns.

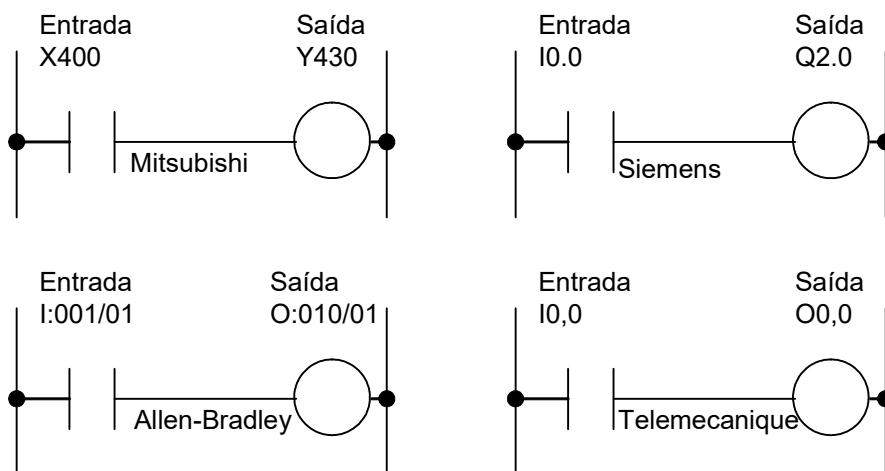


Figura 4.27. Notações de endereçamento de entradas e de saídas

Combinações lógicas entre entradas são possíveis de serem programadas em ladder, assim como em lógicas a relés. Algumas funções lógicas são apresentadas a seguir para ilustrar as funções fundamentais.

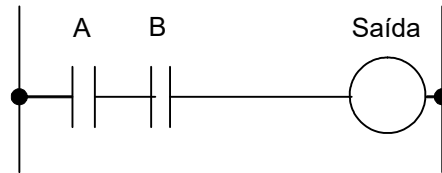


Figura 4.28. Saída = A "AND" B

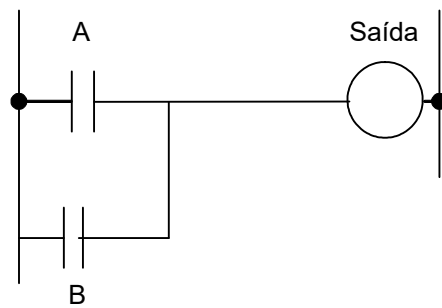


Figura 4.29. Saída = A "OR" B



Figura 4.30. Saída = "NOT" A

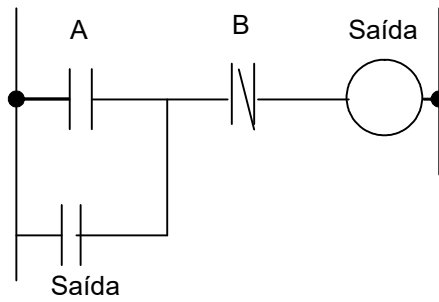


Figura 4.31. Entrada com selo

Em casos onde uma linha aciona mais de um contato, deve conectar os contatos em paralelo, conforme a figura a seguir.

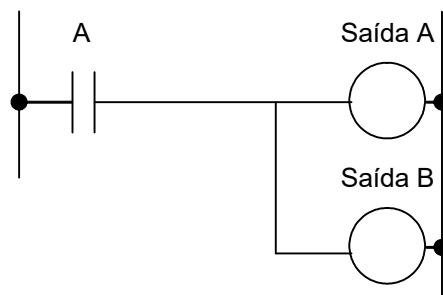


Figura 4.32. Saídas em paralelo

Além de bobinas “NF” e “NA”, podem-se utilizar bobinas do tipo “SET” e “RESET”. Nestas bobinas, somente a energização ou a desenergização é possível em uma linha de programa, conforme a ilustração a seguir. Note que a bobina de Reset é posicionada posteriormente para se configurar uma relação de prioridade para a desenergização.

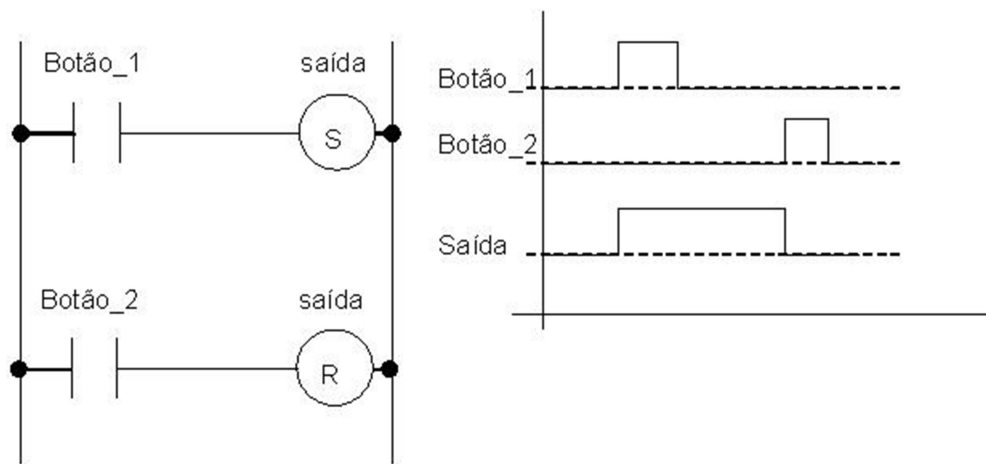


Figura 4.33. Diagrama de operação de bobinas SET e RESET

Um circuito equivalente ao do diagrama de operação de bobinas SET e RESET é possível pelo uso de bobinas NA e NF em arranjo similar ao selo de bobinas de circuitos de lógica a relé, conforme a figura a seguir.

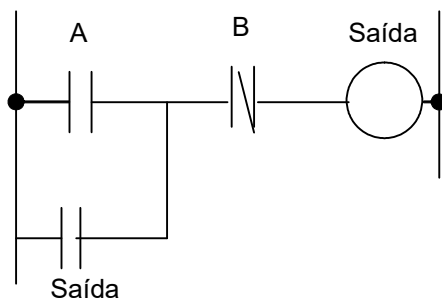


Figura 4.34. Diagrama de operação de um circuito "selo"

Neste caso, a ação do contato A seria a do *BOTÃO 1* e B do *BOTÃO 2*.

Conforme os exemplos apresentados, os contatos podem se normalmente abertos, normalmente fechados, set ou reset. Existem mais dois tipos de contatos disponíveis em lógicas ladder: os contatos de pulso sensíveis a bordas positivas ou de subida (P) e a bordas negativas de descida (N). A diferença entre estes e os contatos NA e NF é que a ativação do contato somente acontece durante um ciclo de varredura do CLP após a detecção da borda de subida ou de descida na linha em que o contato está posicionado.

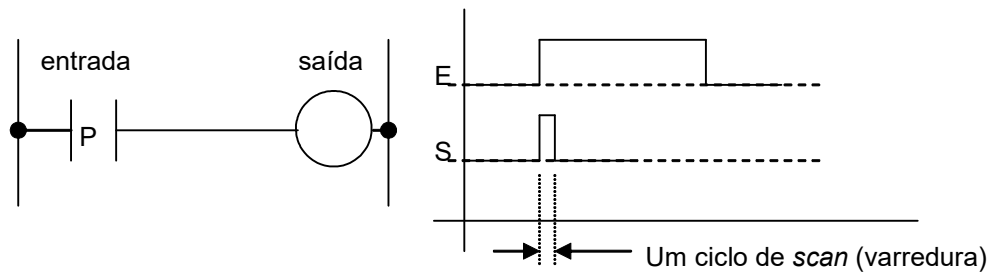


Figura 4.35. Contato tipo Pulso P

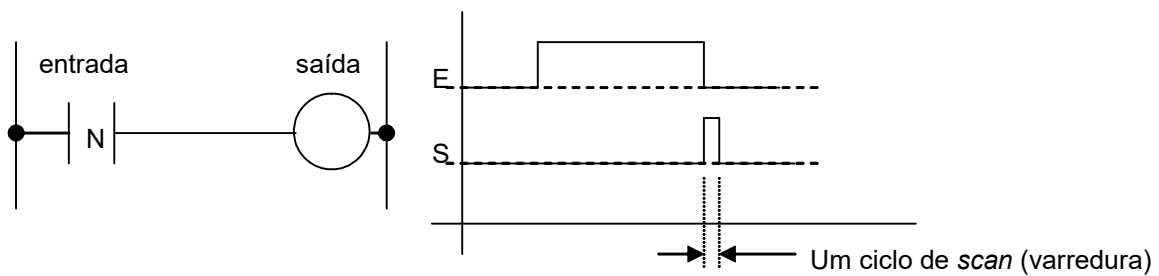


Figura 4.36. Contato tipo Pulso N

Uma aplicação possível com os contatos sensíveis a pulso é apresentada adiante. Neste circuito, um pulso em I0.0 pode tanto ligar como desligar a saída Q0.5.

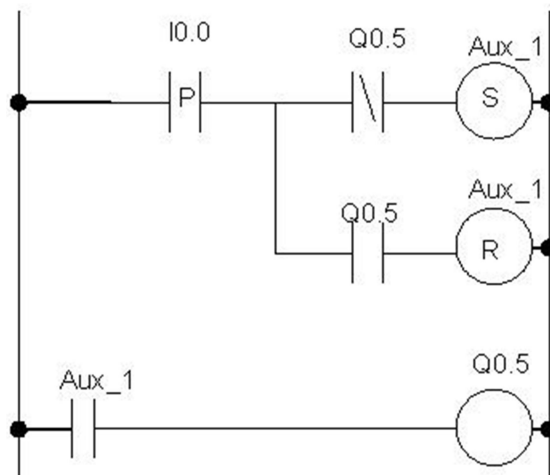


Figura 4.37. Lógica para ligar e desliga uma saída através de pulsos em uma entrada

Este tipo de operando sensível a bordas de subida ou de descida também é aplicado a bobinas, conforme a ilustração a seguir.

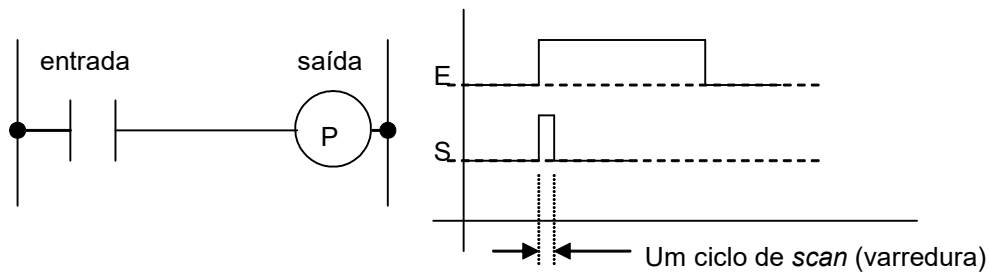


Figura 4.38. Bobina sensível à borda de subida

Os elementos sensíveis à borda de subida têm seu valor em 1 durante um único ciclo de varredura, quando seu fluxo de energização passa de 0 para 1. O diagrama ladder a seguir apresentado é análogo à aplicação do contato ou bobina sensível ao pulso P na borda de subida.

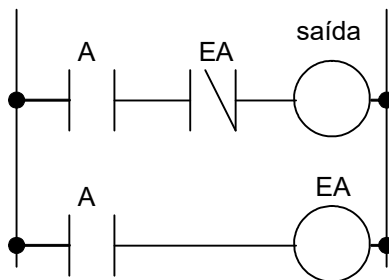


Figura 4.39. Diagrama equivalente à bobina sensível à borda de subida

De forma análoga, as bobinas sensíveis às bordas de descida operam segundo a ilustração a seguir.

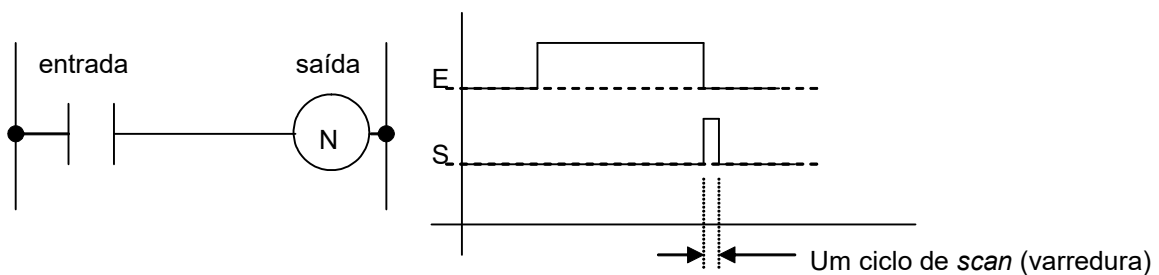


Figura 4.40. Diagrama equivalente ao de bobina sensível à borda de descida

Quando o estado de uma saída deve ser preservado em memória não volátil após a parada da execução de um programa para uma futura continuação da execução daquele programa, então pode-se armazenar a imagem correspondente em memória retentiva. Neste caso utilizam-se bobinas do tipo “M” (bobina retentiva), “SM” (bobina set retentiva) ou “RM” (bobina reset retentiva).

Alguns exemplos selecionados de conversão entre lógicas a relé e seus correspondentes diagrama ladders serão apresentados nas ilustrações a seguir.

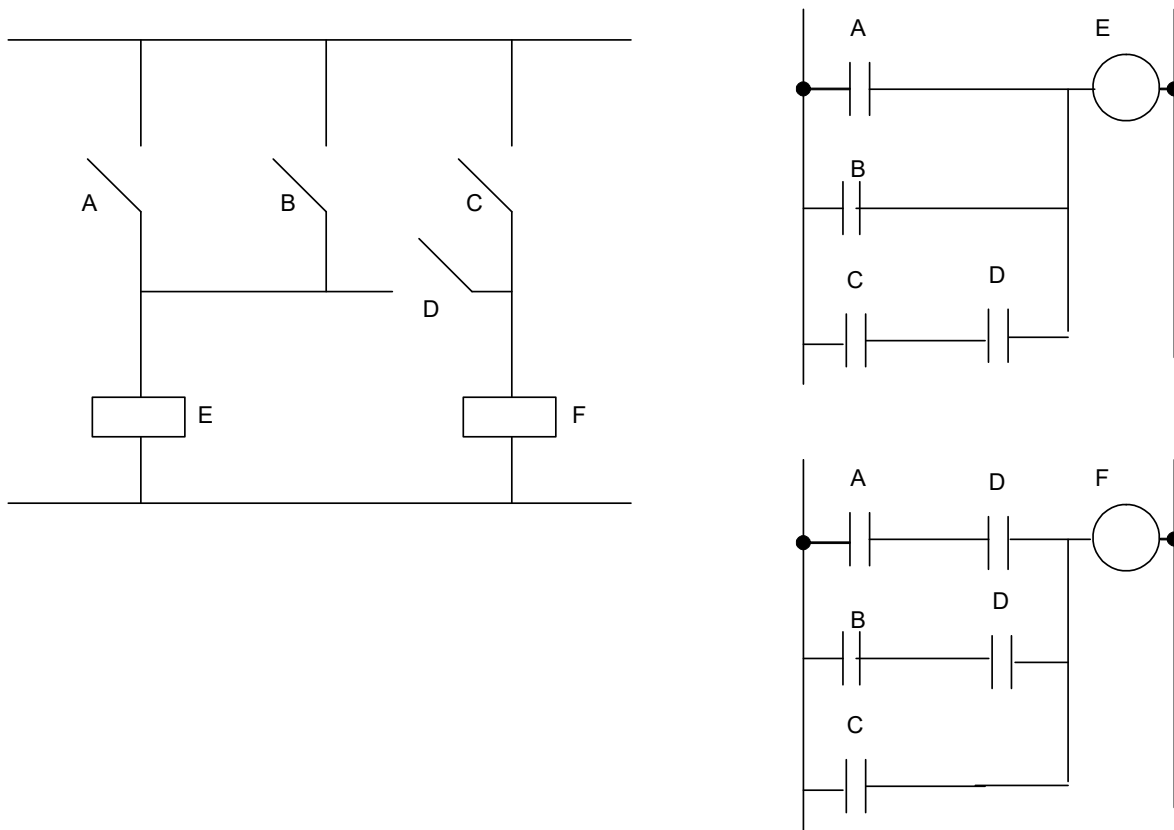


Figura 4.41. Circuito com ponte simples

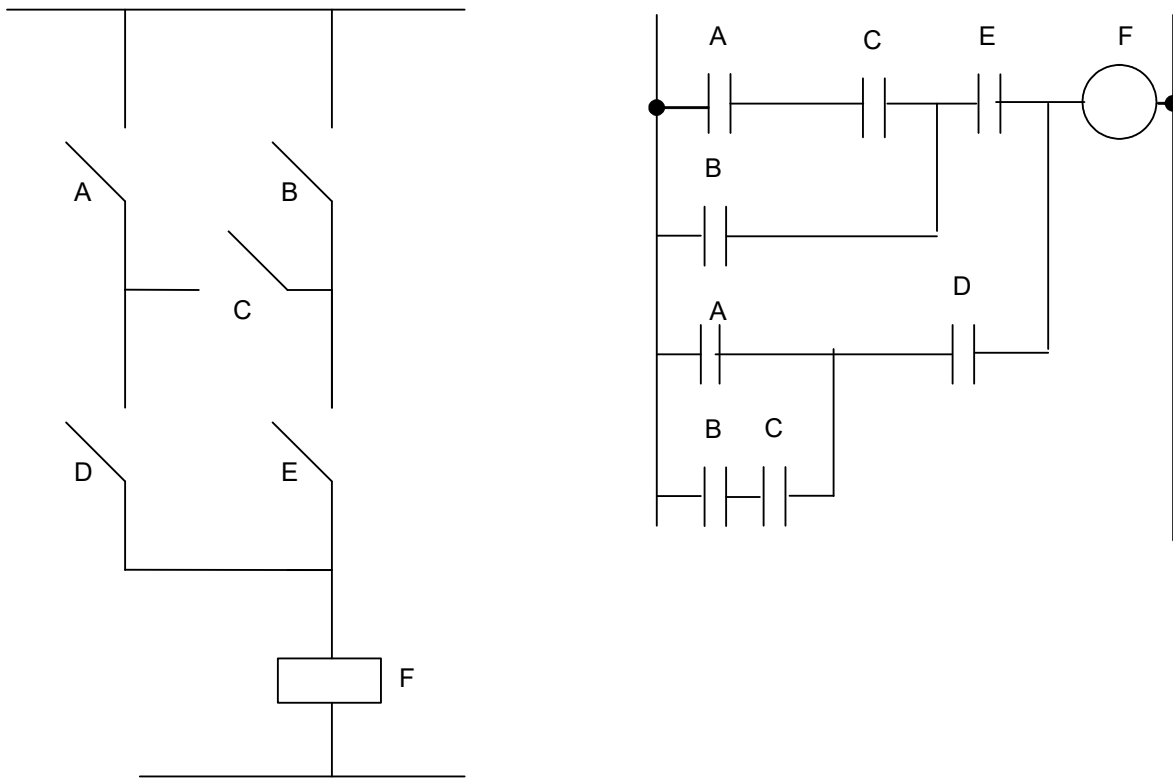


Figura 4.42. Circuito com ponte complexo

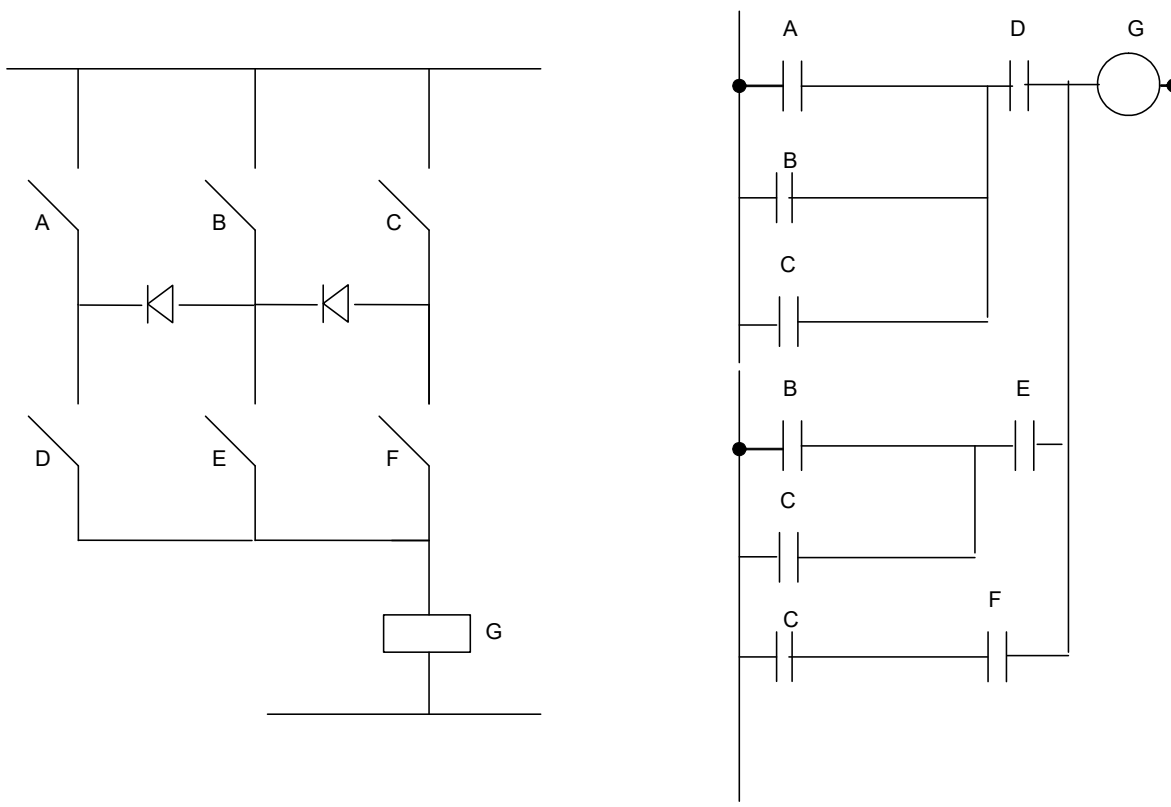


Figura 4.43. Circuito com diodos

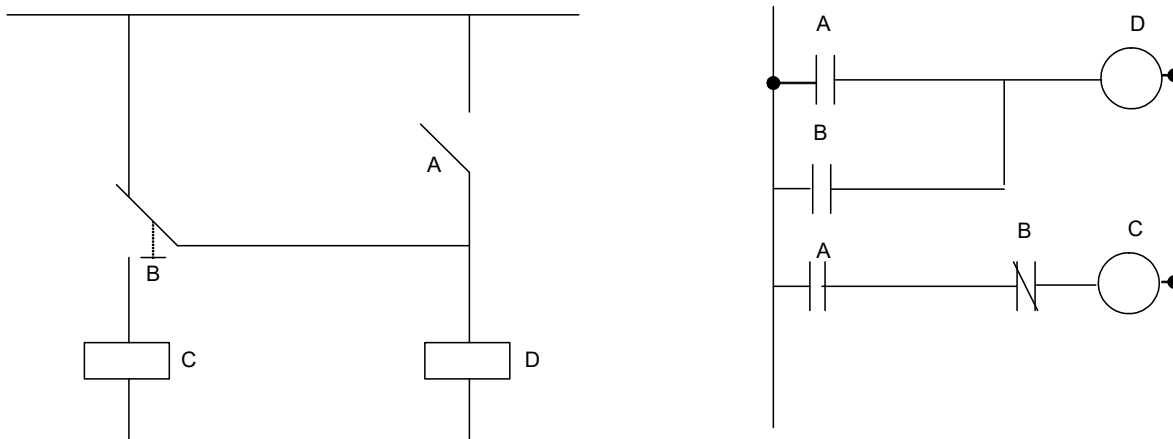


Figura 4.44. Circuito com chave seletora

Blocos funcionais são elementos que podem também ser utilizados em diagramas ladder.

Desde que tenham entradas e saídas booleanas, elas podem se associadas nas linhas do ladder, conforme as figuras a seguir.

Entradas ou saídas de blocos funcionais que não são variáveis do tipo booleano (ex. números inteiros, reais, datas) podem ser conectadas diretamente às entradas/saídas de outros blocos funcionais desde que tipo da variável seja compatível em ambos.

Alguns blocos têm entradas do tipo *enable* "EN" e saídas do tipo *enable output* "ENO", estas entradas e saídas especiais definem se o bloco será ou não executado naquele ciclo de *scan* em função da energização na entrada "EN", que é copiada adequadamente para a saída "ENO" pelo algoritmo do bloco.

A ilustração adiante representa uma linha ladder onde se utilizam blocos funcionais para uma operação com variáveis algébricas.

Cada família e tipo de CLP possuem bibliotecas de blocos funcionais que podem ser incluídos na lógica ladder. Tal recurso permite a um CLP executar não somente lógicas relativas a intertravamentos ou a seqüenciamento de processos, mas também lógicas que realizem controle dinâmico, contabilização de produção, fabricação de lotes em batelada, interfaces com o usuário ricas em recursos e manipulação de sinais de sensores e atuadores analógicos de toda a natureza, através de cartões de entrada e de saída analógica.

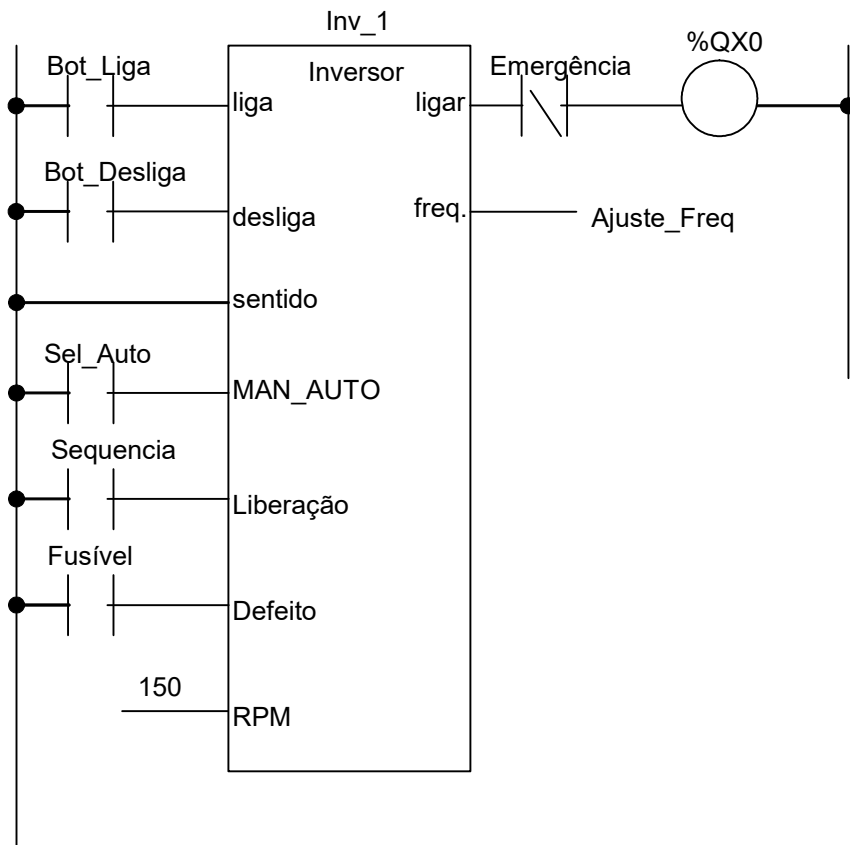


Figura 4.45. Exemplo de utilização de bloco funcional com ladder

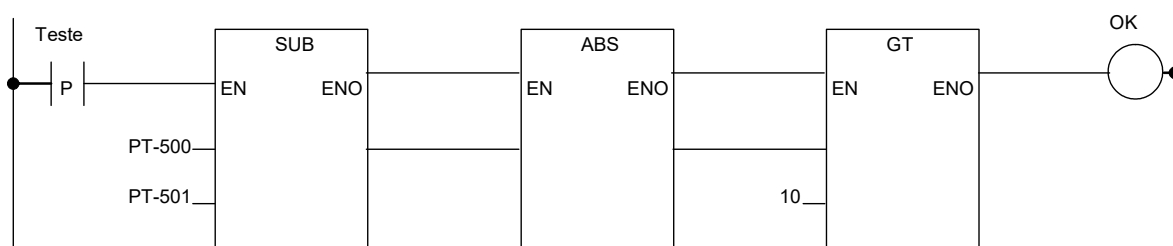
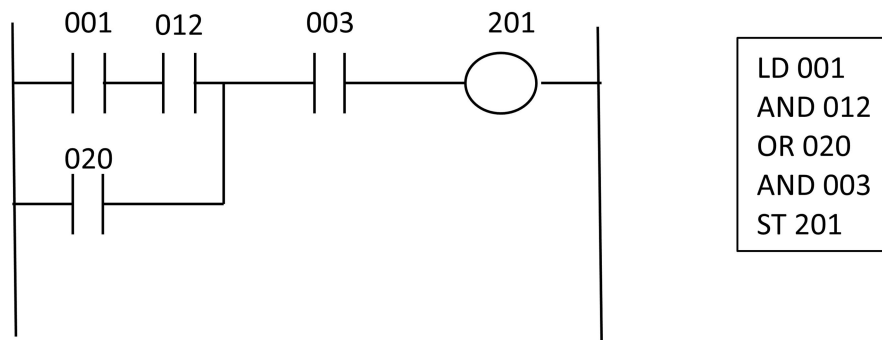


Figura 4.46. Exemplo de uso de “EN” e “ENO” em ladder

Além da linguagem ladder, outra linguagem bastante utilizada para a programação de CLPs é a lista de instruções. A seguir apresenta-se um exemplo desta linguagem.



Exercícios

4.1. Sejam A, B e C eventos de um processo automatizado, em que a saída S é habilitada quando:

- B ou C são verdadeiros e A é falso.
- B e A são falsos.
- Somente A é verdadeiro.

Esquematize uma tabela verdade, uma expressão booleana e o diagrama ladder correspondentes a cada caso.

4.2. Considere o sistema automatizado para carregamento de silos de um entreposto agropecuário, conforme descrito a seguir. Esquematize o diagrama elétrico da interligação dos sensores, atuadores, chaves liga/desliga e o controlador programável.

S_1 , S_2 e S_3 : Silos.

D_1 : Damper de descarregamento do silo S_1 .

D_2 : Damper de desvio da caixa desviadora de 2 vias (permite a mudança do fluxo do material para S_2 ou S_3).

m_1 e m_2 : Motores das correias transportadoras.

S_1 mín, S_2 mín e S_3 mín: Detectores de nível mínimo de material dos silos.

S_2 máx e S_3 máx: Detectores de nível máximo de material dos silos.

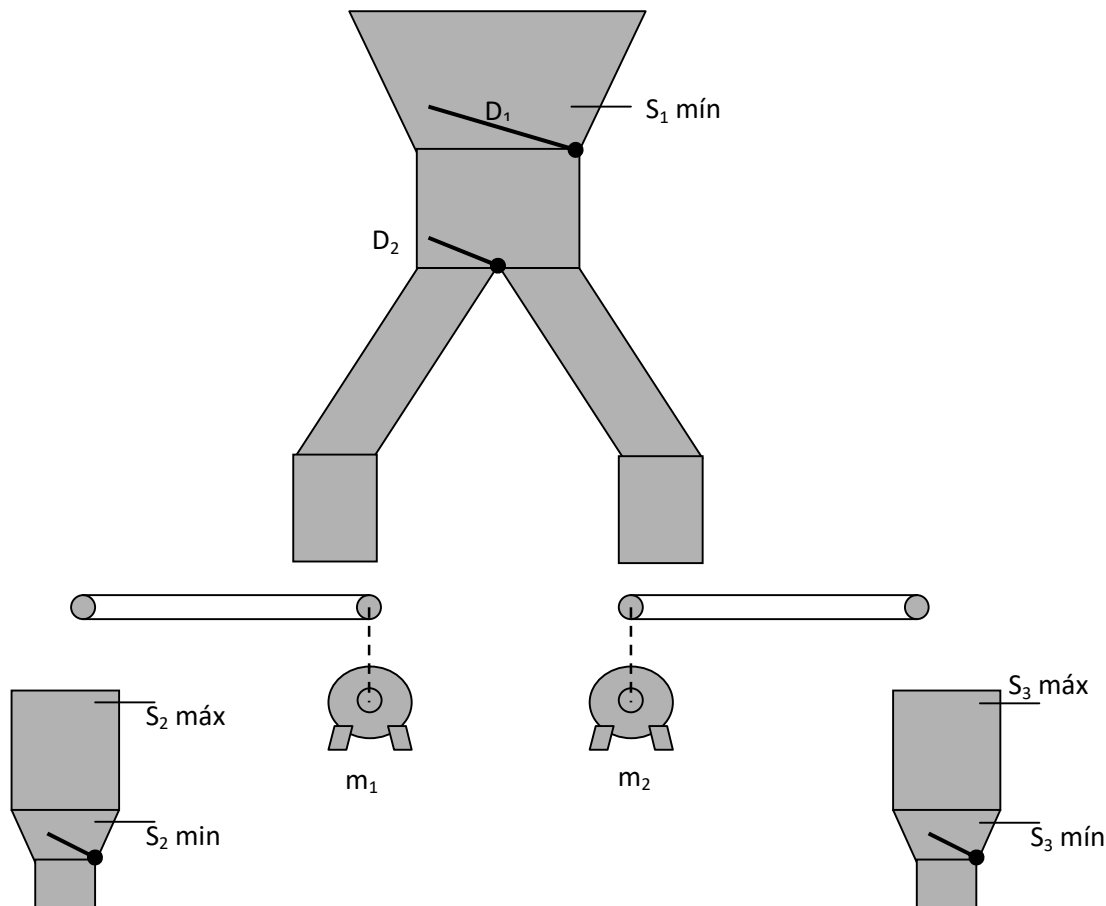
CS_1 e CS_2 : Chaves de fim-de-curso da caixa desviadora de 2 vias (indicam a posição da caixa).

Operação do sistema: os transportadores de correias são ligados e desligados automaticamente, funcionando corretamente. O damper D_1 do primeiro silo S_1 despeja o material no desviador até que o

detector de nível máximo de silo que está sendo carregado entre em ação e o desligue. O damper do primeiro silo é ligado novamente pela atuação do detector de nível mínimo de um dos dois silos.

O detector de nível mínimo do silo vazio também liga o damper de desvio D_2 da caixa, desviando o material adequadamente para esse respectivo silo.

Defina as variáveis de entrada e de saída no CLP e desenvolva o diagrama da fiação elétrica.



4.3. Que aspectos funcionais devem ser considerados na seleção de sensores de proximidade do tipo capacitivo e tipo indutivo?

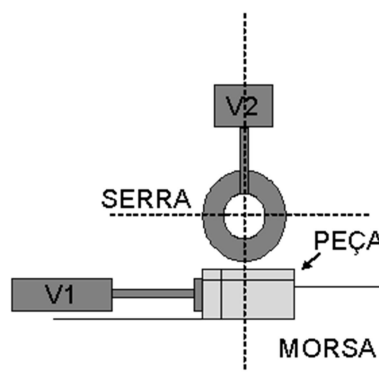
4.4. Uma prensa industrial quando aberta apresenta um vão de 1m de largura por 1,2m de altura, por onde o operador insere continuamente peças para serem estampadas. Como garantir que a prensa não seja acionada enquanto as mãos do operador estiverem na área de prensagem? Explique as características do equipamento e do software (ladder e/ou lógica a relé) utilizado neste projeto.

4.5. Em um entreposto de produtos industrializados é necessário identificar 4 tipos de produtos que são colocados no centro de uma esteira transportadora com uma distância mínima entre eles de 50cm. Temos disponíveis 3 sensores capacitivos e 1 sensor indutivo. Determine as posições onde serão montados os três sensores (indicar no desenho) e a lógica para a identificação dos tipos de produto. Considere quatro variáveis de saída do CLP ou quatro relés para a sinalização dos quatro produtos.

Produtos	Dimensões dos Produtos (cm)
Cerveja em garrafa de vidro	15 (diâmetro) x 30 (h)
Água em garrafa de plástico	15 (diâmetro) x 28 (h)
Caixa de madeira c/ peças de plástico	15 x 15 x 15
Caixa de papelão com colher de pau	15 x 15 x 30

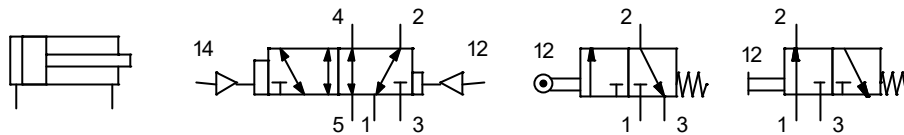
4.6. Esquematize o diagrama ladder correspondente à seguinte situação: Um ventilador é ligado e desligado por qualquer um de três locais definidos. Em cada um destes locais existe um botão “Liga” normalmente aberto (NA) e um botão “Desliga” normalmente fechado (NF).

4.7. Um pistão V1 comanda uma morsa e um pistão V2 comanda a descida de uma serra. A serra não pode trabalhar (descer ou subir) sem que a morsa esteja fechada e a haste de V1 não pode se retrair sem que a haste de V2 esteja retraída.



a) Indique o diagrama trajeto-passo do sistema;

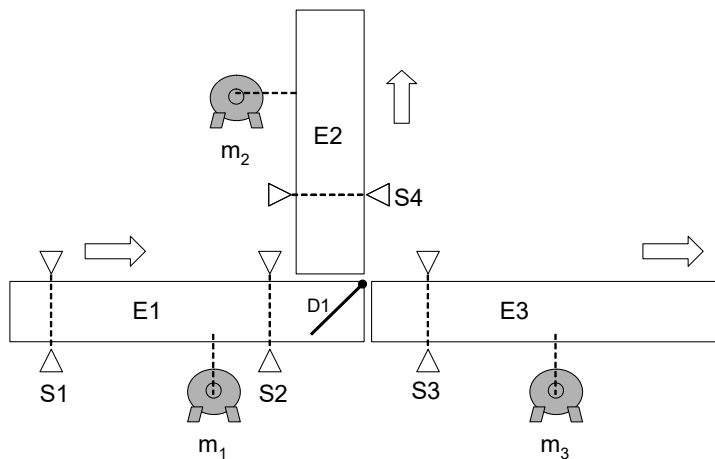
b) Proponha um circuito pneumático para o sistema, utilizando os elementos abaixo na quantidade que julgar necessária (cilindro de dupla ação, válvula direcional, chave de fim de curso e válvula com acionamento a botão para a partida do ciclo).



c) Proponha uma lógica ladder para comandar o mesmo sistema. Considere agora sinais elétricos para as chaves de fim de curso, para o botão de partida e para os pilotos das válvulas direcionais. Identifique e indique todos os sinais como entrada ou saída do CLP.

4.8. O que é o tempo de varredura de um CLP? De que ele depende?

4.9. Transporte de matéria-prima: Projete a automação de um sistema que se baseia em 3 esteiras transportadoras (acionadas por motores exclusivos), as quais dispõem caixas de tamanhos diferentes em locais diferentes, de acordo com a figura:



As caixas são colocadas por operários na esteira E1 e detectadas pelo sensor S1. Ao detectar as caixas, S1 aciona o motor M1 colocando E1 em funcionamento e, simultaneamente, inicia um temporizador que, ao término de 40s pára E1, desde que não haja outra caixa sobre ela.

Ao chegar ao final de E1, as caixas são detectadas pelo sensor S2, que as classifica em grandes e pequenas; logo após, o dispositivo D1 as separa. As caixas grandes são colocadas no começo da esteira E3 e as pequenas, na E2. Estas esteiras operam por 30s e param, a não ser que haja outras peças para transportar.

Para o projeto, detalhe:

Lista de variáveis de entrada e de saída

Diagrama das variáveis de entrada e de saída

Diagrama elétrico dos atuadores, sensores e clp

Programa em linguagem ladder com blocos funcionais (especificar blocos funcionais através do software Atos A1)

4.10. Estufa: Necessitamos desenvolver um programa de CLP para controlar o funcionamento de uma estufa para motores, sendo que a temperatura desejada será selecionada através de um potenciômetro entre 40°C e 180°C. O aquecimento será feito através de um grupo de resistências de 15KW, e um sistema de ventilação será responsável pela uniformização da temperatura no interior da estufa.

Especificação:

A estufa será ligada através de um botão de pulso NA e desligada através de um botão de pulso NF.

A partir de um pulso no botão “Liga”, será acionado o ventilador, que, através de um contato auxiliar, deverá afirmar ao CLP a sua ligação. Somente após a ligação do ventilador poderá ser acionado o conjunto de resistências

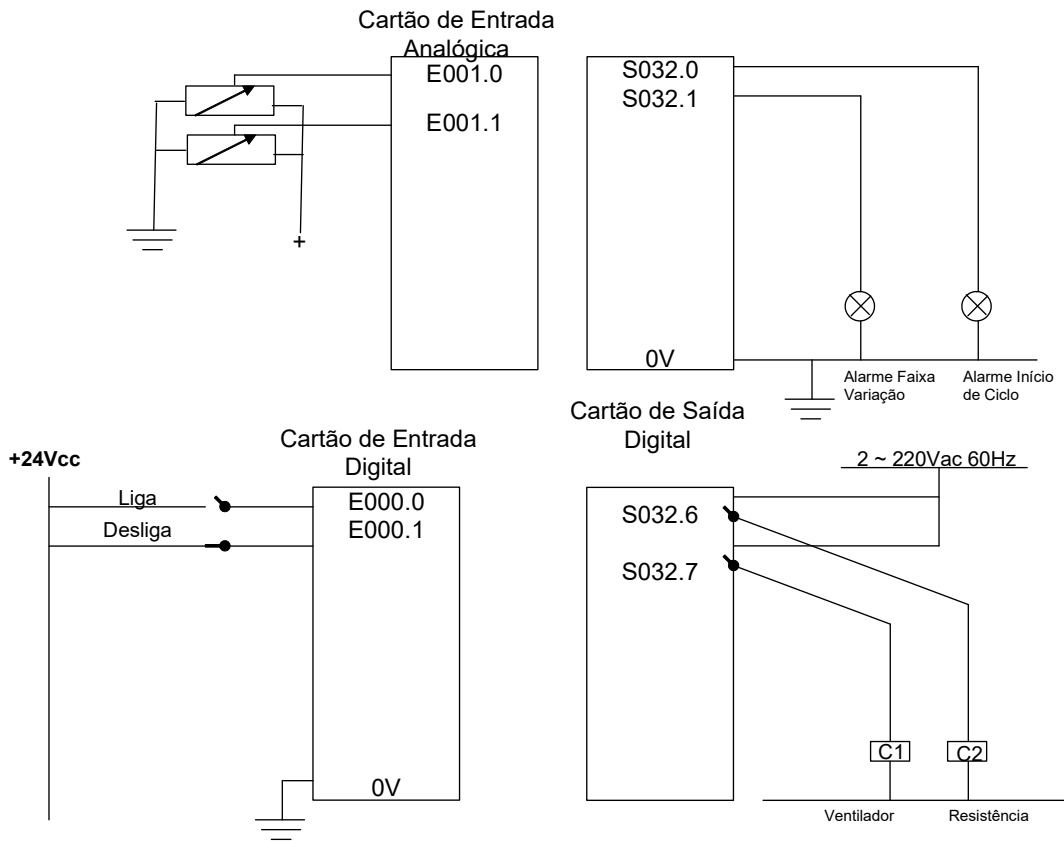
O ajuste de temperatura deverá ser realizado com a estufa desligada, após a ligação do sistema, não será possível regular a temperatura.

A medição da temperatura será feita através de um sistema analógico de 0 a 10Vcc, proporcional a uma temperatura de 0 a 200°C

A temperatura no interior da estufa admite uma variação máxima de 10°C, caso esta variação aumente, entrará em ação um alarme

Prever um sistema de alarme que entre em ação caso não seja atingida a temperatura necessária em até 10 minutos após a ligação, indicando defeito no sistema.

Obs.: Prever as interligações de E/S conforme o esquema e especificar os blocos funcionais.

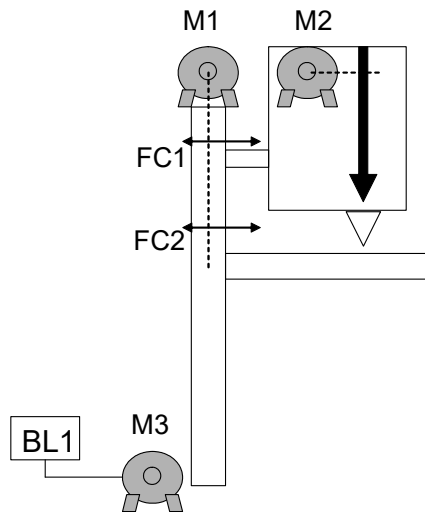


4.11. Criar um programa em Ladder e relacionar as variáveis de Entrada e Saída utilizadas para controlar a furadeira a seguir:

Com FC1 (I:3.0-0) acionado e um pulso no botão BL1 (I:3.0-4), deve-se ligar o motor de descida M1 (O:4.0-1) juntamente com o motor de giro M2 (O:4.0-2).

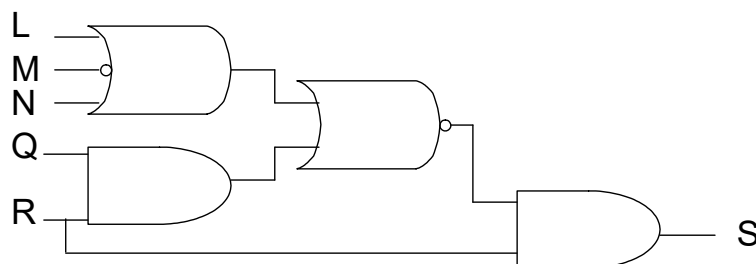
Quando o FC2 (I:3.0-1) for acionado deve-se desligar o motor M1, manter M2 ligado e ligar o motor de subida M3 (O:4.0-3).

Ao acionarmos o FC1, deve-se desligar os motores M2 e M3.

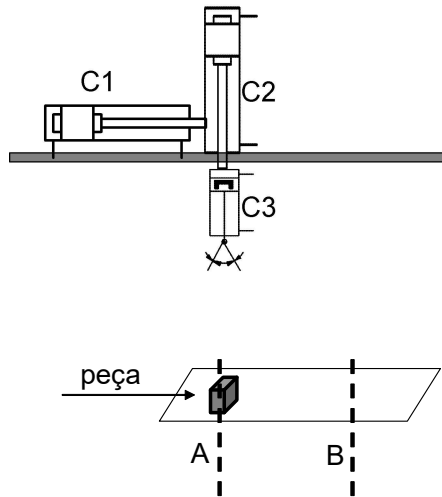


4.12. Dois sistemas de abastecimento de água são constituídos por duas torres (T1 e T2), cada uma com uma caixa d'água superior (CX1 e CX2), um reservatório no nível térreo (R1 e R2) e um poço artesiano único. Um motor M aciona uma bomba que tira água do poço para R1 e R2 através de válvulas V1 e V2 que desviam o fluxo para R1 e R2 quando abertas não simultaneamente. Dois motores M1 e M2 independentes recalcam água dos reservatórios para as caixas d'água. Considerando as caixas d'água e os reservatórios com sensores de nível mínimo e máximo, projete um sistema de automação baseado em um CLP programável via Ladder que procure manter cheias as caixas e os reservatórios, desligando o sistema quando não houver água no poço. Apresente a lista de Entradas e Saídas, o programa Ladder e o esquema elétrico.

4.13. Apresente um programa Ladder correspondente ao circuito apresentado.



4.14. Um equipamento manipulador composto por três atuadores pneumáticos de dupla ação C1, C2 e C3, responsáveis por movimentar peças em uma mesa sem arrastá-las. A posição inicial das peças é "A" e a posição final das peças é "B". A distância entre "A" e "B" é igual ao curso de C1. C3 possui uma garra adequada às peças em questão. Cada atuador do manipulador é equipado com dois sensores de fim de curso: na posição de avanço e na de retorno.



a) Apresente o diagrama de trajeto-passo para o ciclo do manipulador, considere o ciclo com início na posição “A” e fim (do ciclo) na mesma posição de início.

b) Liste os elementos pneumáticos necessários e projete o acionamento puramente pneumático do manipulador considerando-se um botão para partida do ciclo. Utilize o método de projeto passo a passo com identificação numérica adequada dos elementos. Apresente o circuito projetado.

c) Descreva formalmente a operação do manipulador com base no circuito projetado do item b.

d) Liste os elementos necessários para o acionamento eletro-pneumático do manipulador considerando-se um botão (NA) para partida do ciclo e um botão de parada (NF) de emergência que interrompe o ciclo na posição onde este esteja. Utilize um CLP para controlar o manipulador. Utilize válvulas direcionais com duplo piloto solenóide e sensores de fim de curso elétricos. Apresente a tabela de variáveis de entrada e de saída do CLP e sua lógica ladder. Identifique as variáveis auxiliares utilizadas. Apresente o circuito eletro-pneumático

e) Descreva formalmente a operação da lógica ladder com base no item d.

4.15. Alarme de alta pressão: O circuito faz soar uma buzina e acender uma lâmpada piloto quando a pressão atingir um valor alto perigoso. Depois que o alarme soa, o botão ACKN (conhecimento – NA) desliga a buzina e deixa a lâmpada acesa. Quando a pressão baixar para um valor seguro, a lâmpada se apaga.

4.16. Controle de bomba e duas lâmpadas piloto com chave de nível: A chave de nível opera o starter do motor da bomba. A bomba enche um tanque com água. Enquanto o nível do tanque receptor estiver baixo, a chave liga o motor da bomba e acende a lâmpada R. Quando o nível atingir o nível máximo (tanque cheio), a chave desliga o motor e a lâmpada R e acende a lâmpada A. Se o motor se sobrecarregar, o motor é desligado, mas a lâmpada R continua acesa.

4.17. Controle seqüencial de 3 motores: Ligar três motores, isoladamente e um após o outro. A parada desliga todos os motores. Qualquer sobrecarga desliga todos os motores.

4.18. Unidade de aquecimento de óleo

Motor M1 opera uma bomba de alta pressão, que injeta óleo em um queimador.

Motor M2 opera um soprador que força o ar para o queimador, quando o óleo estiver sendo queimado.

Motor M3 opera um soprador de ar de circulação.

Chave liga-desliga comanda o circuito

Termostato TS1 sente a temperatura do interior do ambiente

Termostato TS2 sente a temperatura do trocador de calor.

Quando a chave estiver ligada (ON) e a temperatura interna do ambiente for baixa, TS1 fecha e parte os motores M1 e M2.

Quando a temperatura do trocador de calor subir demais, TS2 fecha e parte M3. O soprador circula o ar dentro do ambiente através do trocador e aumenta a temperatura dentro do ambiente.

Quando a temperatura do ambiente subir muito, TS1 abre e desliga o motor da bomba M1 e o motor do soprador M2. O soprador de ar de circulação continua operando até que o trocador de calor seja resfriado a uma temperatura baixa, quando TS2 abre.

6. SEQUENTIAL FUNCTION CHARTS

Além da programação de CLPs pelo método intuitivo, existem algumas métodos formais muito uteis principalmente quando se programa aplicações grandes ou complexas:

- o crescimento da complexidade de sistemas automatizados requer a minimização do tempo de programação e, se possível, a possibilidade de reutilização de códigos para CLPs;

- em aplicações críticas relacionadas à sistemas de segurança, existe a necessidade de procedimentos de verificação e de validação das aplicações, ou seja, uma metodologia para se provar características estáticas e dinâmicas de programas de CLPs, tal como tempos de resposta.

A figura seguinte representa o processo de criação de um programa de controle para CLP. Sem o uso de técnicas formais, o processo seria limitado ao anel externo do diagrama. O programa é implementado diretamente da especificação informal do problema e validado informalmente com bases nestas especificações. Entende-se por especificação informal do problema a descrição do sistema automático em linguagem não formal (descrição do problema) e todo o conjunto de documentação técnica do sistema. A validação neste caso é possível posteriormente à implementação do projeto e quando se dispõe de uma equipe de testes, porém este é um processo lento e caro.

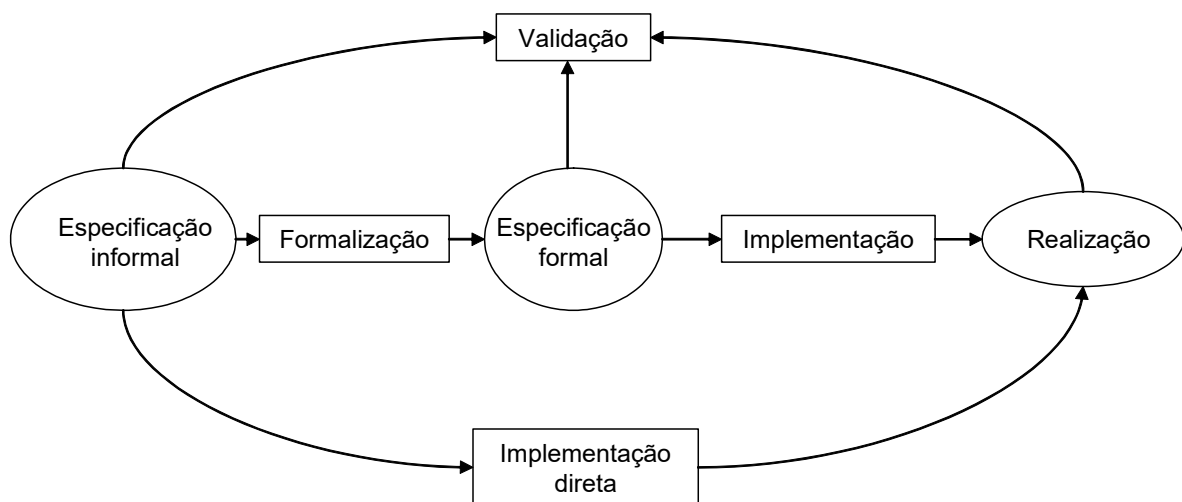


Figura 5.1. Processo de implementação de programas em CLPs

Como alternativa a esta metodologia de projeto, pode-se utilizar técnicas formais para a programação e a validação de programas em CLPs. Nesta metodologia, adicionam-se as etapas de formalização, especificação formal e implementação.

A formalização consiste da conversão da especificação informal em uma especificação formal, ou seja, que segue regras de representação e de comportamento. Esta etapa pode ter o auxílio de softwares nas nunca é automática, pois depende da interpretação do projetista. Uma especificação formal deve ser capaz de:

- 1 - capturar as características estáticas e dinâmicas do sistema;
- 2 - acomodar o estilo de modelagem dos estados e de atividades de uma forma natural;
- 3 - permitir a sua integração em qualquer outra linguagem de modelagem;
- 4 - ser capaz de representar sistemas sobre múltiplos níveis de abstração.

Já para o rastreamento de erros, verificação e validação, a técnica de especificação deve permitir simulação e animação e dispor de técnicas de análise eficientes para verificar as propriedades do modelo.

A implementação depende da plataforma tecnológica que se utiliza (software e hardware do CLP), neste caso pode-se contar com ferramentas de geração de código automático.

De acordo com a norma IEC 61131 descrita anteriormente, existem atualmente cinco linguagens padronizadas: IL (lista de instruções), ST (texto estruturado), FBD (diagramas de blocos funcionais), SFC (sequential function charts) e LD (diagramas ladder). No capítulo anterior foram apresentadas as linguagens LD e FBD, neste capítulo será apresentada a linguagem SFC.

Derivada das Redes de Petri (técnica descrita mais a diante neste capítulo) e do seu subconjunto destinado à modelagem de sistemas seqüenciais IEC 848, o denominado Grafcet (lançado em 1988), o SFC se presta a uma especificação formal de alto nível adequada para projetos de maior complexidade baseados em CLPs. A vantagem a destacar é a possibilidade de se traduzir um código SFC para as demais linguagens da norma.

Quando o projeto requer o processamento paralelo de duas ou mais tarefas em um controlador, ao contrario da programação direta em ladder (por exemplo), a técnica do SFC torna-se mais adequada, pois baseia-se em diagramas gráficos de estrutura seqüencial adequada à modelagem de sistemas

orientados a eventos, através de uma abordagem capaz de resolver problemas sequencias de grande porte ou complexidade com múltiplos processos (ou seqüências) paralelos.

Atualmente o SFC é adotado por alguns fabricantes de CLP como linguagem direta de programação. Para CLPs que não possuem essa característica, o SFC pode ser traduzido para ladder ou para qualquer outra linguagem IEC 1131, tornando-se assim uma ferramenta para elaboração de comandos seqüenciais, segundo uma abordagem *top down* para estruturação das ações do programa.

A sua filosofia consiste em partir da descrição informal das funções de automação a se projetar e decompô-las em passos e transições.

Nos passos e só neles são realizadas ações (por exemplo, ligar um contator de acionamento de um motor) ou eventualmente pode não se realizar qualquer ação (quando o controlador está em repouso). Em cada instante, numa dada seqüência só um passo está ativo.

Para haver a transição de um passo para outro é preciso que se verifique uma ou mais condições de transição. Por exemplo, para que um elevador em movimento do 2º para o 3º andar pare neste último, é preciso que um fim de curso indique a chegada da cabine a este andar.

Exemplo de representação de um SFC:

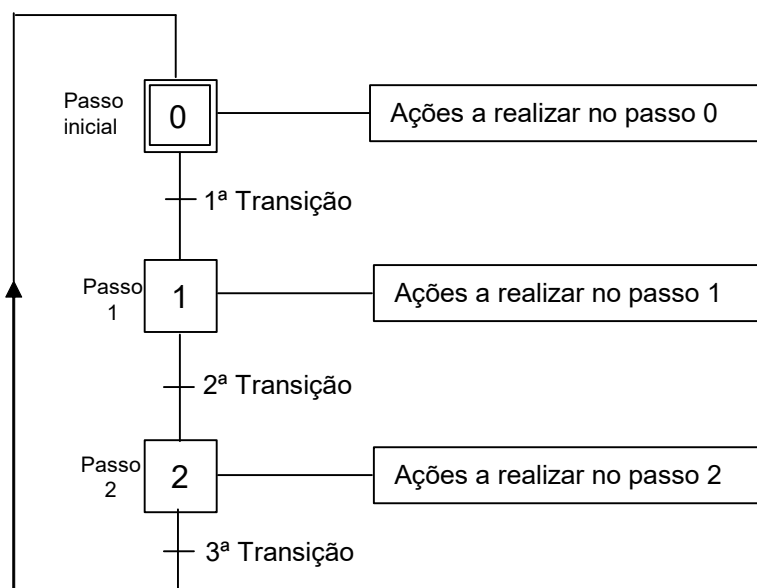
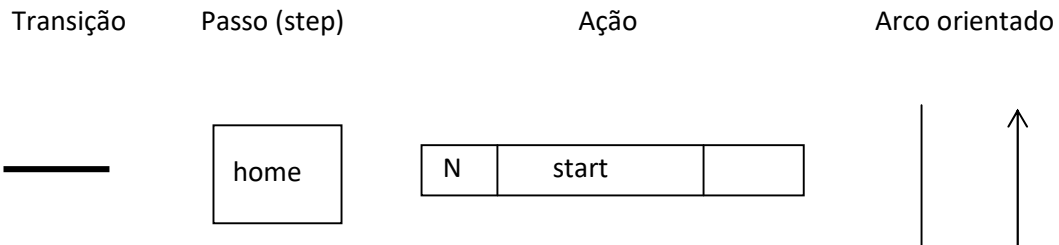


Figura 5.2. Exemplo de um SFC

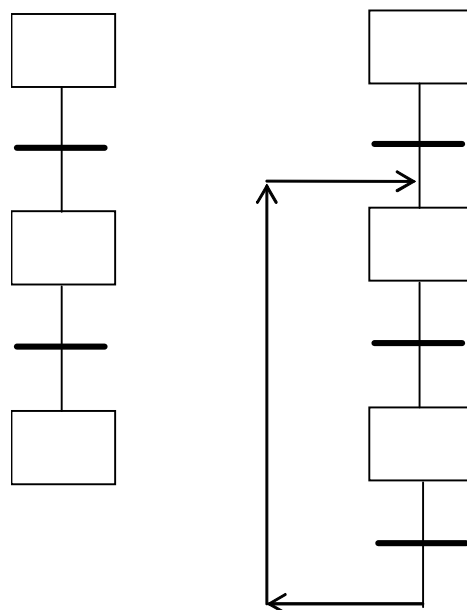
Elementos de um SFC

Os elementos de um SFC são passos, transições, arcos, ações e regras de evolução.

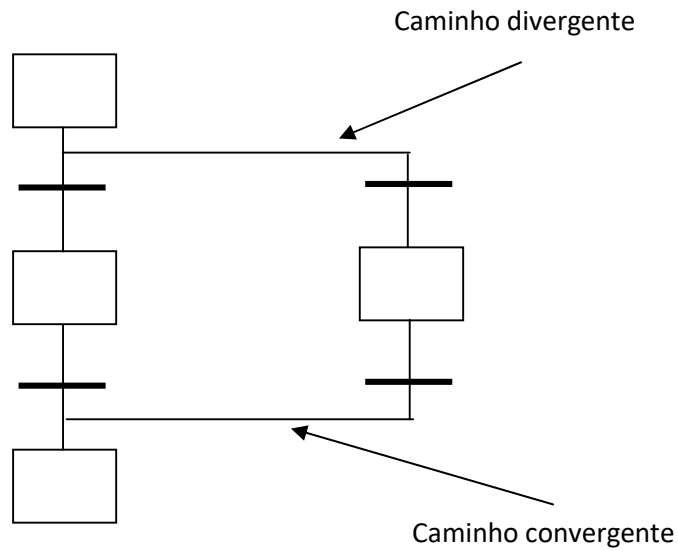


Um arco é representado por uma linha vertical e orientado de cima pra baixo (default). No caso em que se projeta um arco orientado de baixo para a cima, o sentido deve ser representado através uma seta.

Uma seqüência é definida por uma série de passos, transições e passos ligados por arcos orientados:



É possível realizar seqüências alternativas através de caminhos divergentes e convergentes. As seqüências são executadas em mútua exclusão.



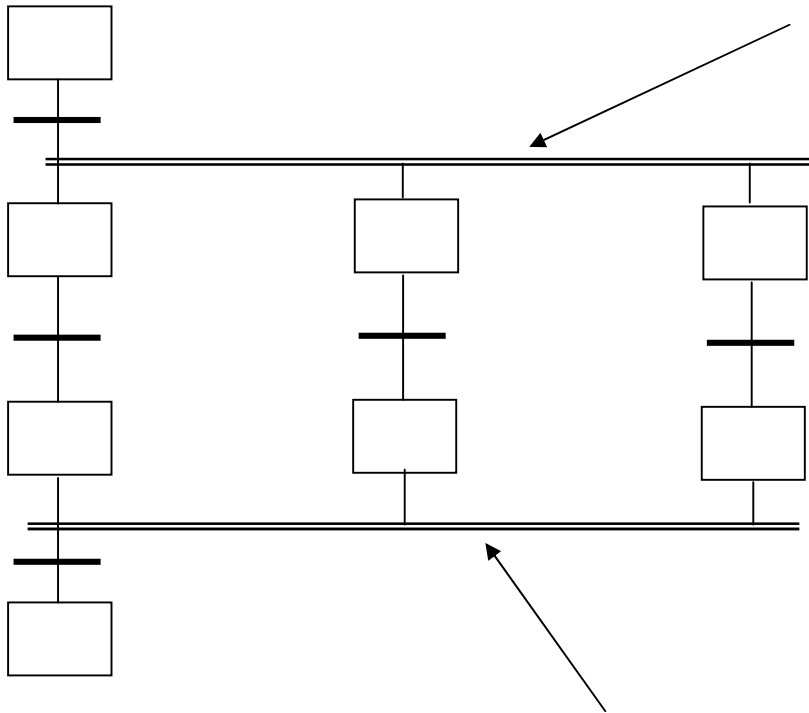
Regras de validação de seqüências alternativas:

As condições booleanas associadas a transições são estimadas da esquerda para direita.

Se uma ou mais transições tem condições TRUE, a precedência é garantida à transição mais a esquerda, é possível todavia modificar a regra de precedência, atribuindo uma prioridade cada transição. É boa prática associar a transições presentes em seqüências alternativas, condições lógicas mutuamente exclusivas

Seqüências Simultâneas: é possível realizar seqüências simultâneas, da seguinte forma:

Divergencia de seqüências simultâneas



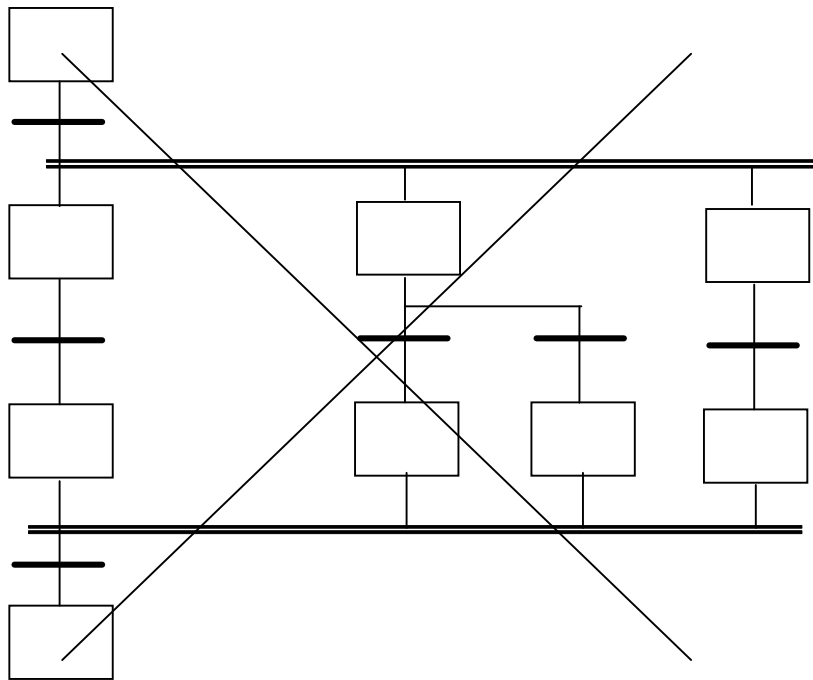
Convergencia de seqüências simultâneas

As seqüências simultâneas são consideradas todas em paralelo, pode-se por exemplo utilizar uma seqüência para o controle do processo e outra para realizar o monitoramento. Em cada seqüência simultânea, somente um passo de cada vez pode ser ativado e a convergência de seqüências simultâneas acontece somente quando todos os últimos passos de cada seqüência forem ativados.

Regras de Programação de uma seqüência simultânea:

É necessário que a convergência de seqüências simultâneas possa ser atuada, quando todos os passos que se inserem em tal convergência devem poder ficar todos ativos.

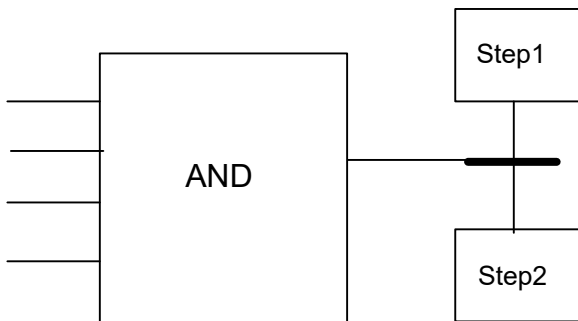
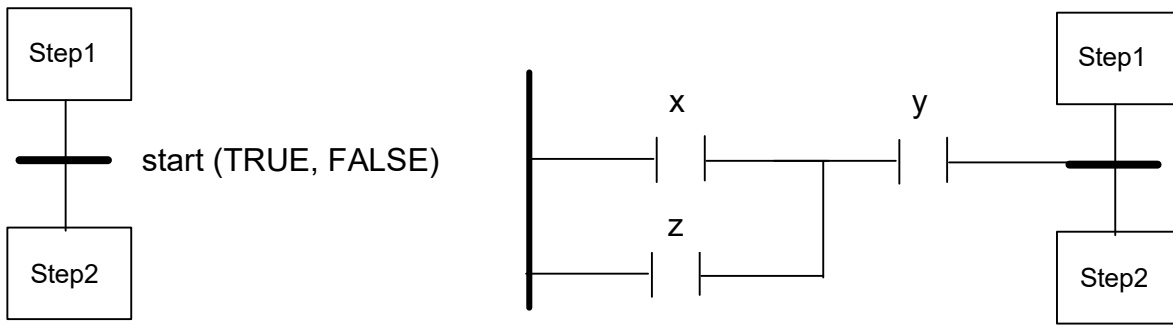
Exemplo de convergência de seqüências simultâneas errada:



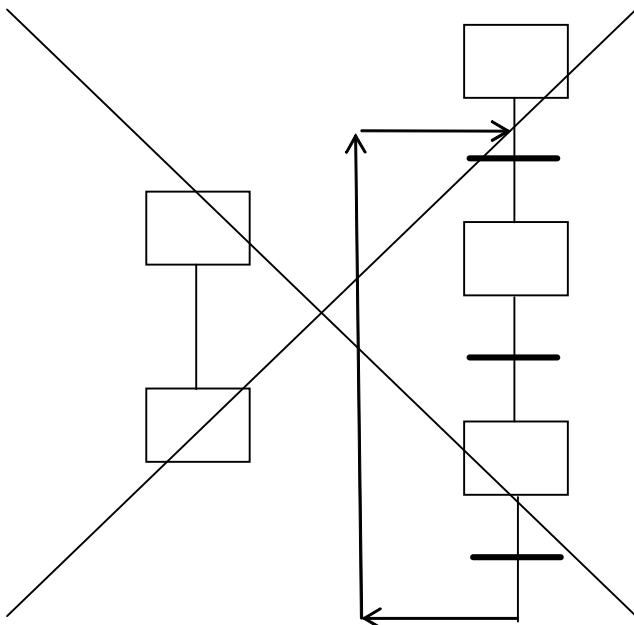
Uma transição é representada por uma barra horizontal, a qual é possível associar pelo menos um passo precedente e pelo menos um passo seguinte (os passos precedentes e seguintes podem ser mais de um) Para cada transição vem associada uma condição. As condições podem ser constituídas de:

- Uma variável booleana (TRUE, FALSE) ou,
- Uma linha em linguagem ladder. Se a linha for ativada, a condição associada à transição torna-se verdadeira, ou,
- Uma saída binária de qualquer FB. Se a saída é verdadeira então as condições associadas à transição tornam-se verdadeiras, ou
- Um nome de um programa. O programa deve ser definido separadamente e pode ser escrito em uma das linguagens IEC 1131-3. O programa deve fornecer como saída um valor booleano que deve ser associado ao nome do próprio programa.

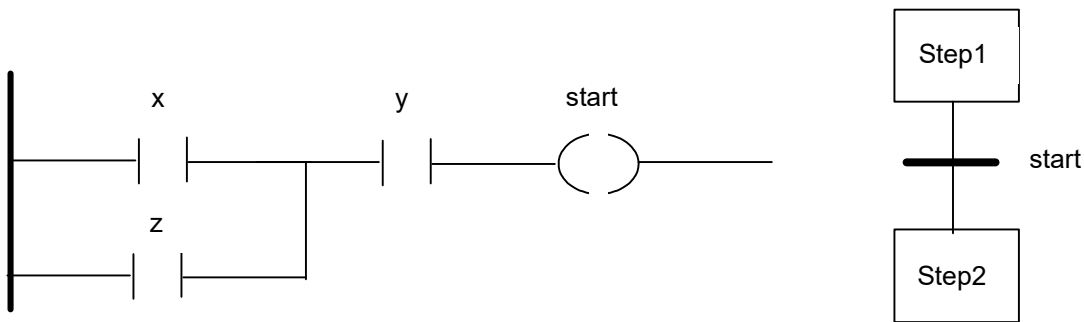
A condição associada a uma transição é estimada **SOMENTE** quando todos os passos precedentes a essa estiverem ativos. Quando todos os passos precedentes a uma transição estiverem ativos e a condição associada a esta é verdadeira, todos os passos precedentes desativam-se e os passos seguintes tornam-se ativos. Se a uma transição está associada a um nome, este deve ser singular (ou único) em todo o programa (variável local). Exemplos de Transição:



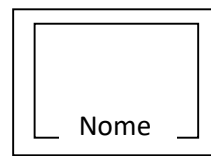
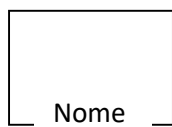
Exemplo de diagramas errados:



Programa "start" codificado em Ladder:



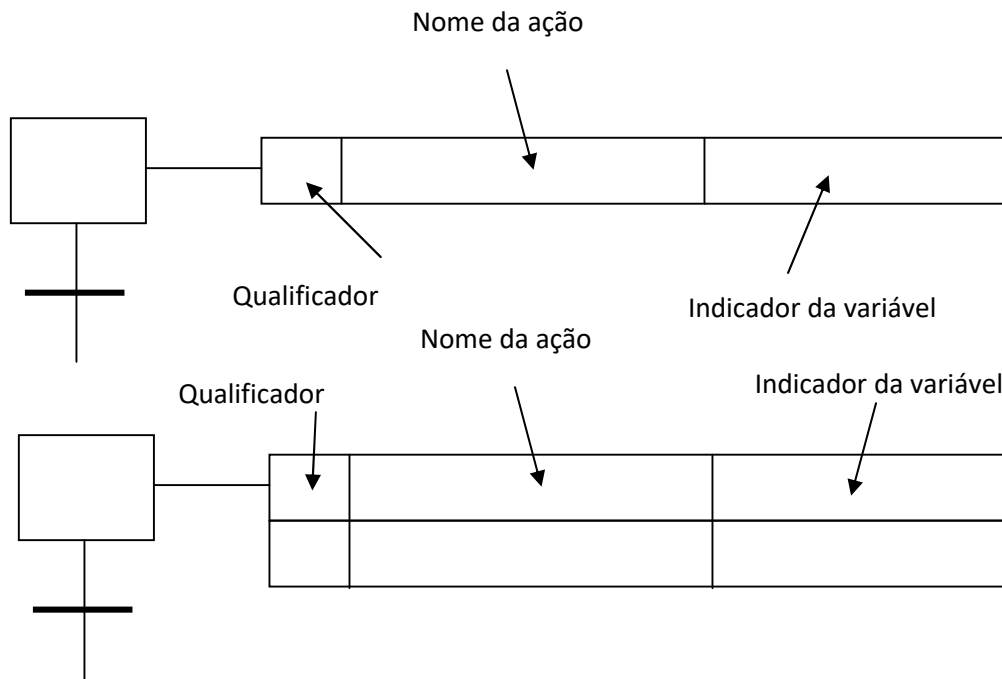
Tipos de passo: Passo Normal (caixa simples) e Passo Inicial (representado por caixa dupla)



Pode existir somente um Passo Inicial, que vai ser ativado em um *cold-start*. Cada passo tem associado um nome único em todo o programa (variáveis locais).

Para cada passo é associado um estado: ativo ou não ativo. O estado é ativo quando a transição precedente ao passo é "liberada", isso significa também que a condição associada a esta transição tornou-se verdadeira. O passo é desativado com a liberação da transição que o sucede, ou transição de saída.

A cada passo é possível se associar uma ou mais ações que descrevem o que se fazer quando o relativo passo é ativado. Cada ação é representada por um retângulo conectado ao passo:



Qualificador: Este parâmetro especifica as modalidades de execução da ação. Pode assumir os valores: N, S, R, L, D, P, SD, DS, SL

Nome da ação: Este parâmetro é único em todo o programa. Ele corresponde ao nome do programa que realiza a ação. O programa pode ser escrito em qualquer uma das linguagens IEC 1131

Indicador da variável: Este parâmetro é opcional e permite indicar a variável que é modificada pela ação e, quando modificada, indica o fim da execução.

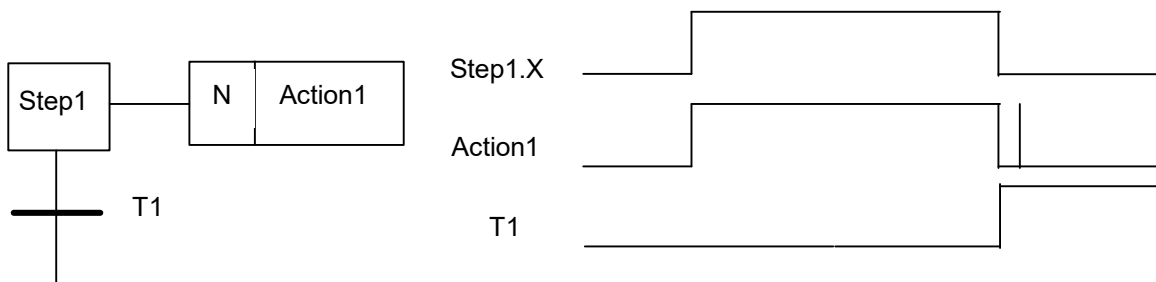
Uma mesma ação pode ser associada a mais de um passo.

É possível que a um passo não seja associada nenhuma ação. Nesse caso quando o passo fica ativo, nada é executado, e se espera que o passo seja desativado pela liberação da transição de saída.

Observação: A norma IEC 1131-3 assume que qualquer ação associada a um passo vai ser executada ainda mais uma vez após a desativação do passo. Pode-se ou não adotar tal hipótese.

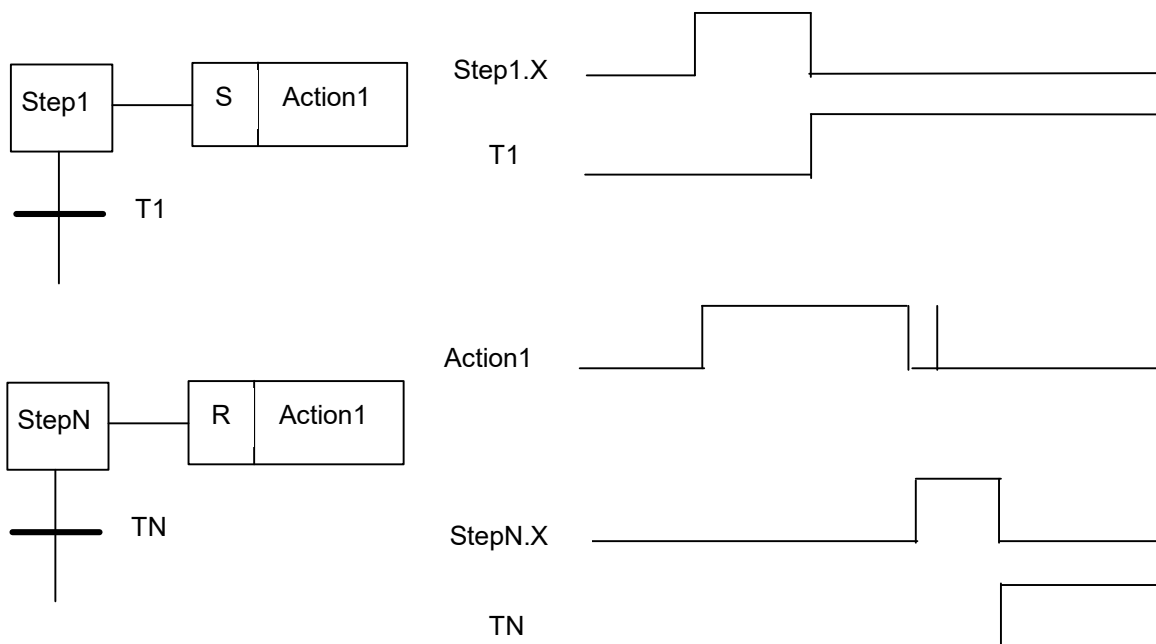
Vantagem: permite o reset de eventuais parâmetros ou variáveis usados durante a execução da ação.

Qualificador 'N' Non stored



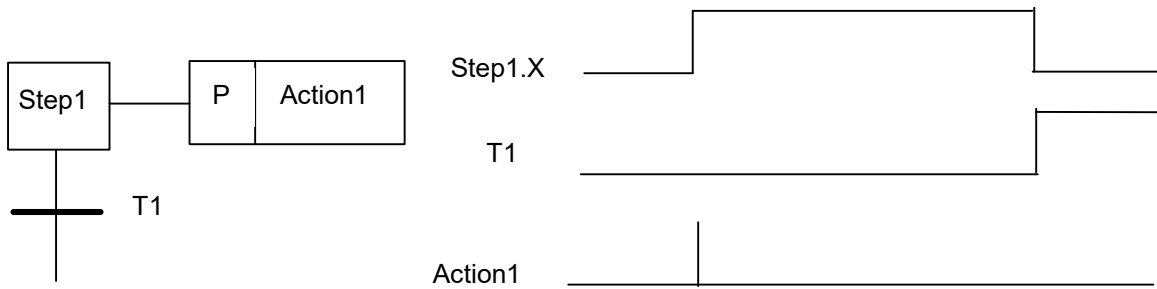
A ação "Action1" é executada continuamente enquanto o Step1 é ativo. Quando o Step1 é desativado, a ação é executada somente mais uma vez

Qualificador 'S', 'R', Set e Reset



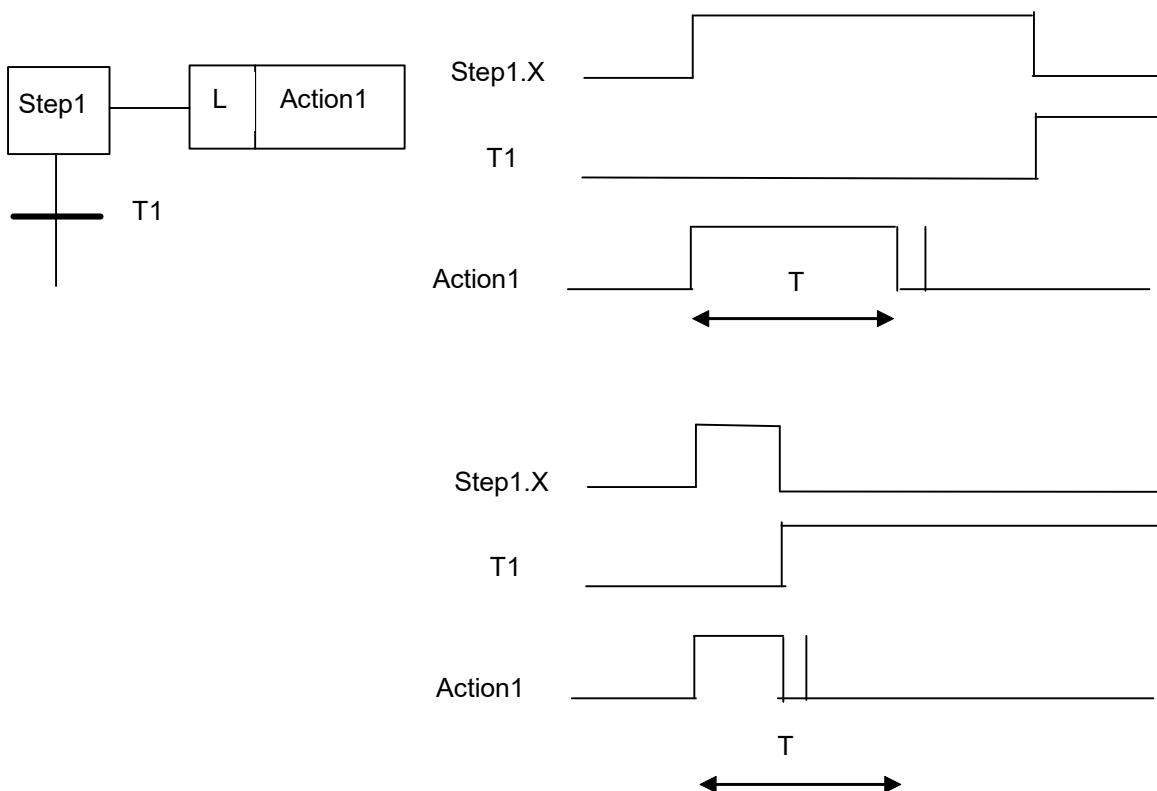
A ação "Action1" começa a ser executada quando o Step1 ficar ativo. A ação é memorizada e continua a ser executada enquanto o passo StepN ficar ativo. Neste caso a ação "Action1" é executada somente mais vez. No caso em que uma ação não seja resetada através do qualificador 'R', esta será executada indefinidamente.

Qualificador 'P' pulse



Quando o Step1 é ativado, a ação Action1 é executada somente uma vez, em algumas implementações do padrão, a execução depois da desativação do Step1 pode não ser prevista.

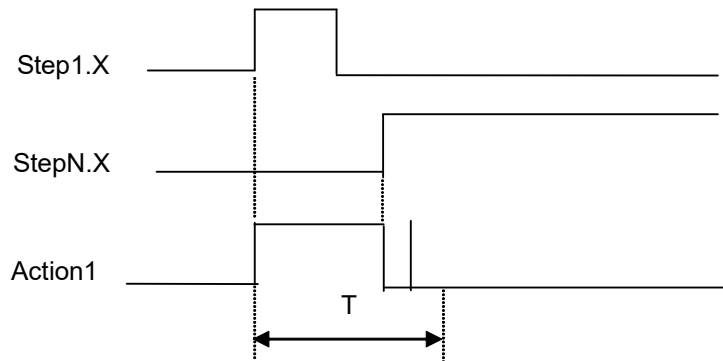
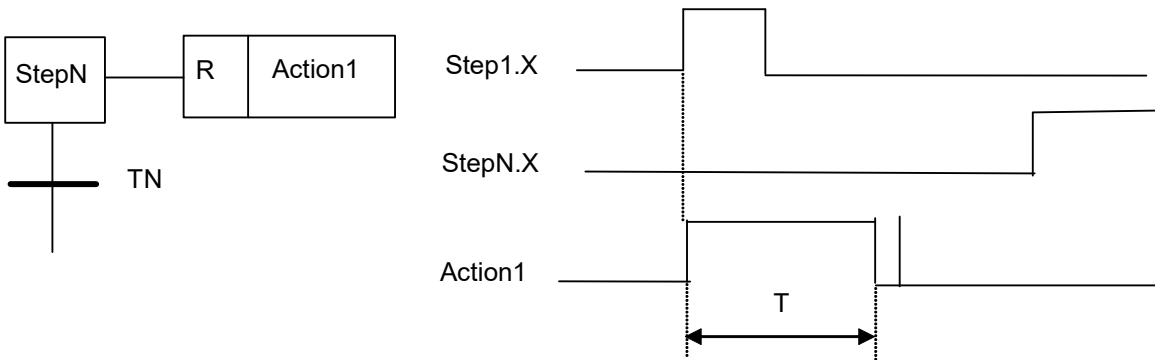
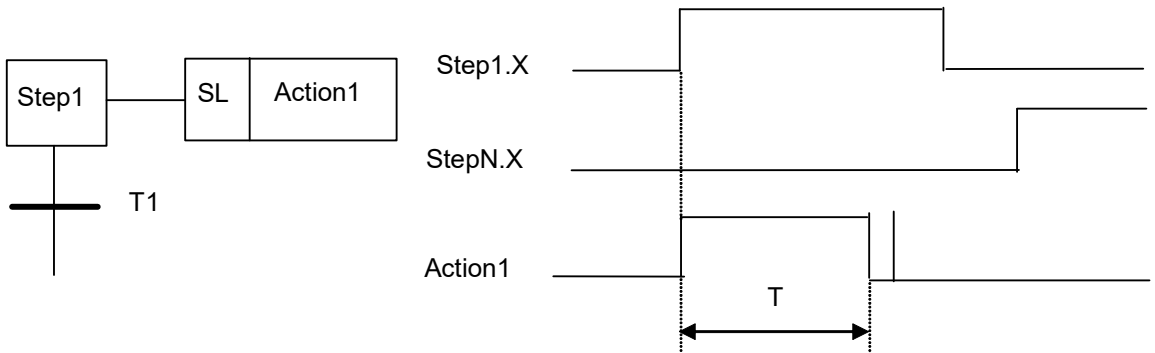
Qualificador 'L' Time Limited



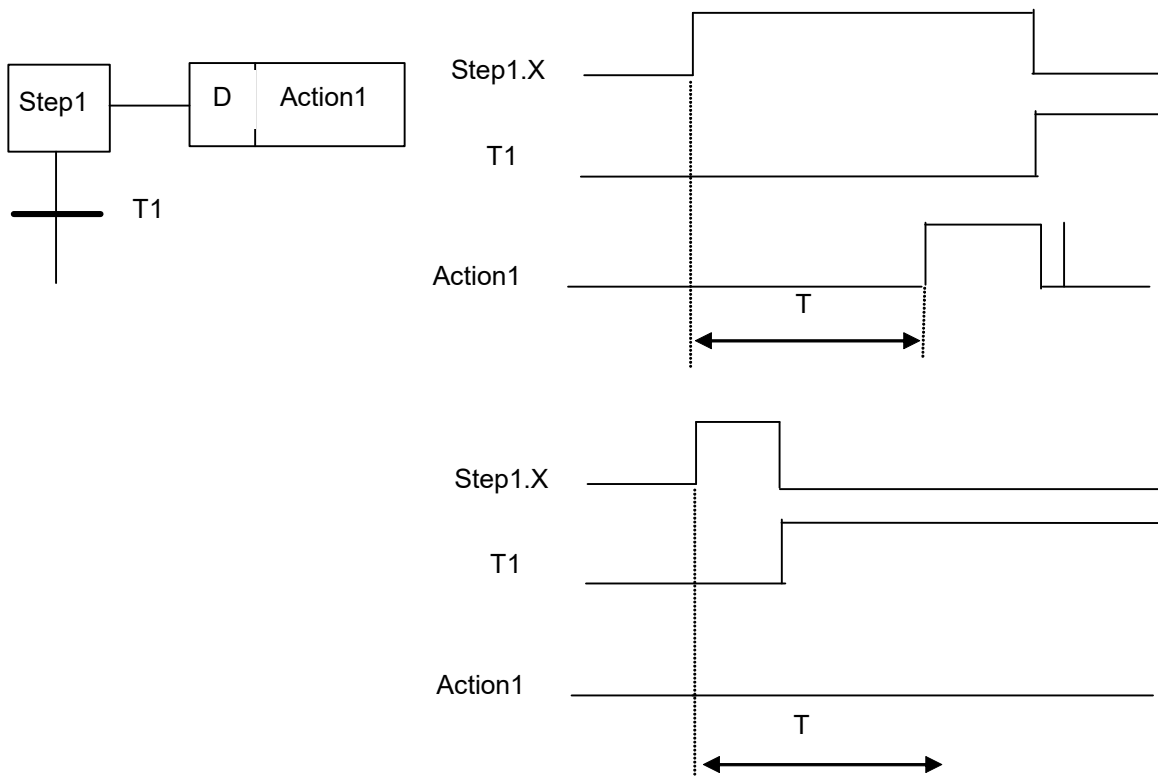
A ação "Action1" começa a ser executada quando o Step1 fica ativo. A ação é executada em um determinado intervalo de tempo T (associado ao qualificador L). Ao fim de tal intervalo, a ação é terminada. Se o passo é desativado antes que do final do tempo T, então a ação "Action1" é terminada.

Quando a ação é terminada, em ambos os casos, essa é executada pela a última vez e somente uma vez.

Qualificador 'SL' stored and time limited

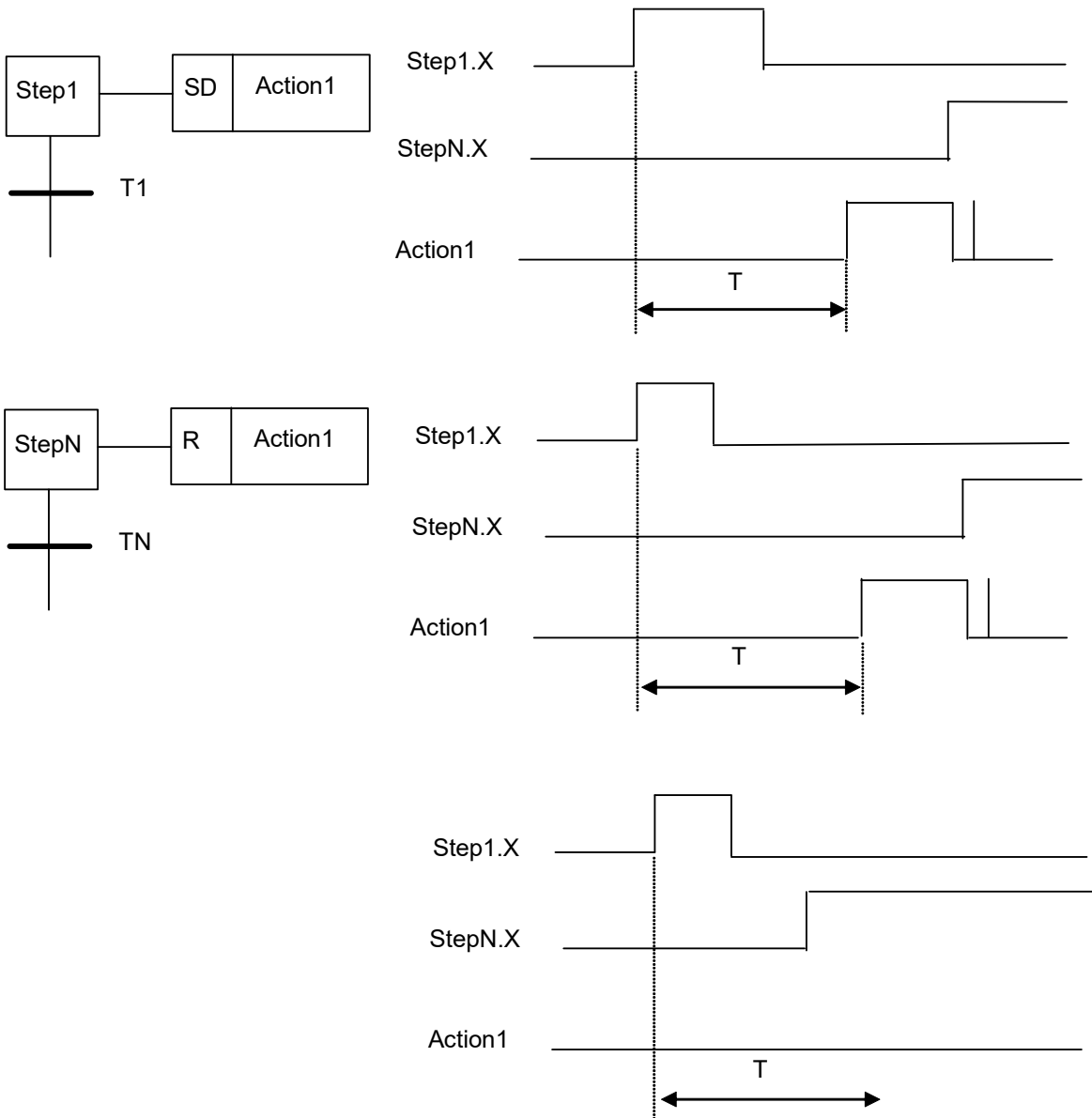


Qualificador 'D' time delayed



A ação "Action1" começa a ser executada depois do intervalo temporal T (associado ao qualificador D) a partir do instante no qual o Step1 fica ativo. A ação é executada enquanto o Step1 não é desativado. Nesse caso a ação "Action1" é executada somente uma vez. Se o Step1 não permanece ativo por um tempo pelo menos igual a T, a ação nunca é executada.

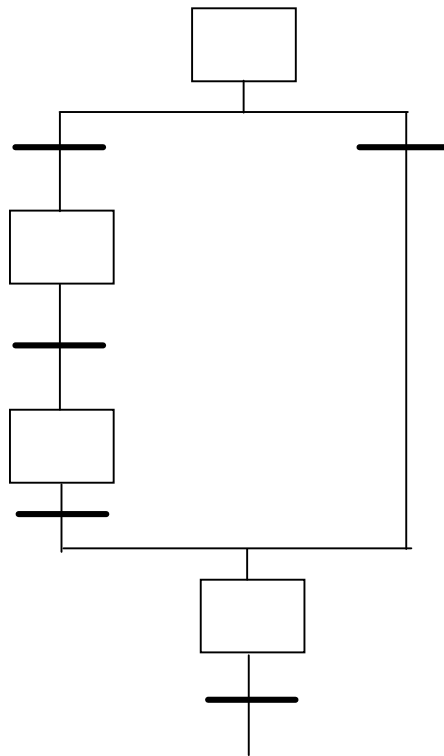
Qualificador 'SD' stored and time delayed



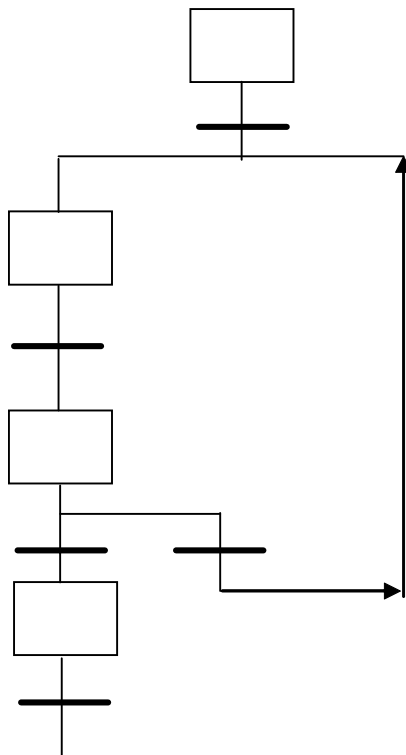
Quando o Step1 é ativo, a ação "Action1" é memorizada, mas não é executada até que o intervalo de duração T termine. A ação é executada enquanto não é resetada através do qualificador R. Se a ação "Action1" é chamada com o qualificador R antes que o intervalo T acaba-se, a ação não será executada.

Elementos Fundamentais de Controle através de SFC

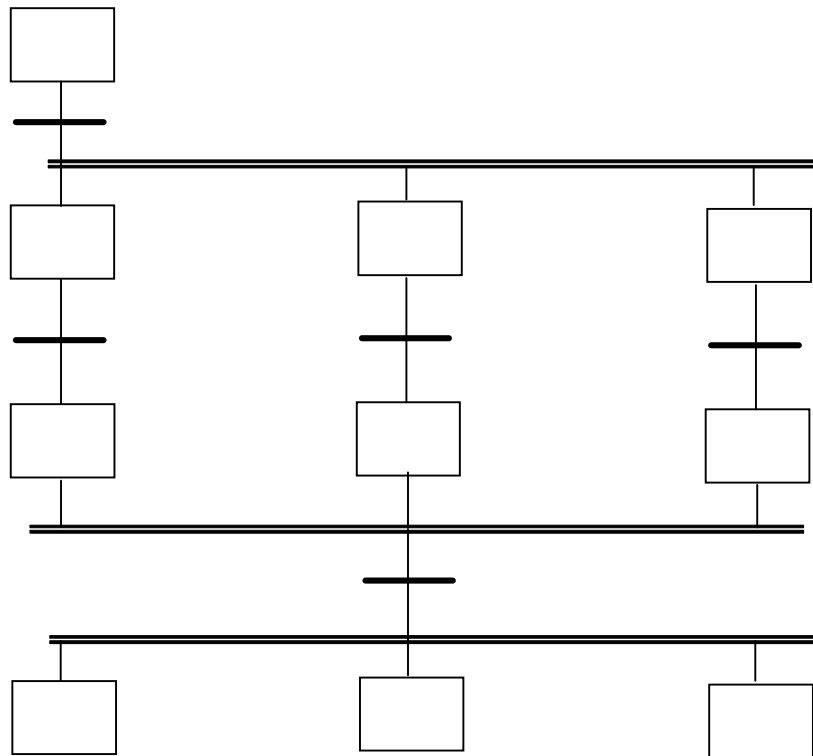
"Bypass" ou salto de uma seqüência:



Loop de uma sequencia



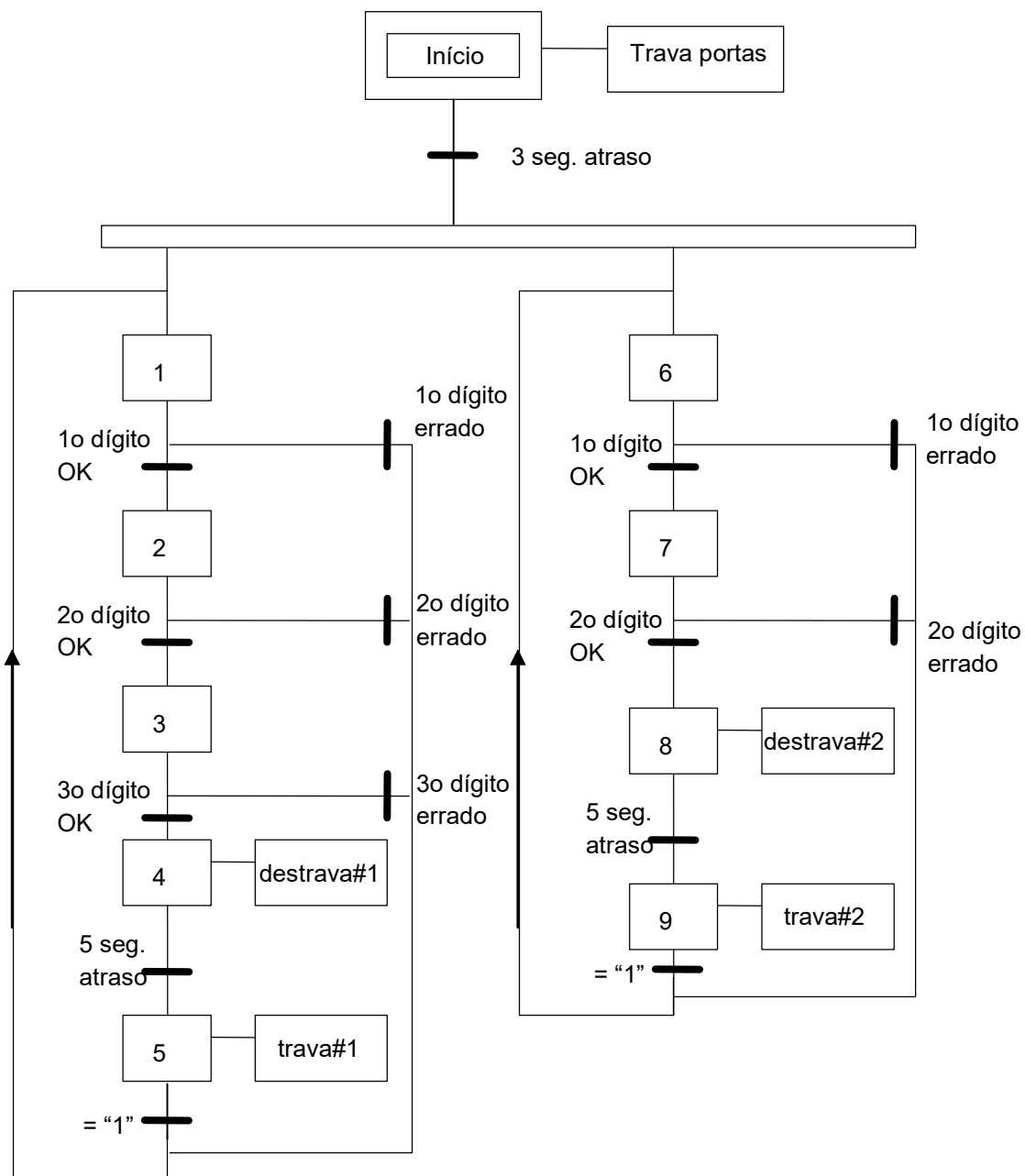
Rendezvous:



Exemplo: Controle de Acesso

O exemplo a seguir representa um SFC para controle de segurança de duas portas. Uma porta requer um código de acesso de dois dígitos, a segunda porta requer um código de três dígitos. A execução do sistema inicia-se no topo do diagrama, no passo inicial, quando o sistema é ligado. Existe uma ação associada ao passo inicial que trava a porta (Nota: na prática, em SFC se utiliza lógica ladder para acionamento de saídas e leitura de entradas, recurso não apresentado no diagrama).

Após a inicialização do diagrama, sua execução é dividida em dois processos e ambos os passos 1 e 6 são ativados. Quando uma condição lógica de digitação do código é satisfeita, a transição correspondente desativa o passo anterior e aciona o passo seguinte. Enquanto o passo 1 estiver ativo, duas transições podem ser disparadas. No caso da digitação correta do dígito, o passo 1 ficara inativo e o passo 2 ativo. Caso a digitação seja incorreta, então o a transição leva o sistema novamente para o passo 1. O passo 1 não tem nenhuma ação associada, portanto nada deve acontecer neste período. A lógica para ambas as portas deve se repetir após o final do ciclo de “combinação-abertura-atraso-travamento”.

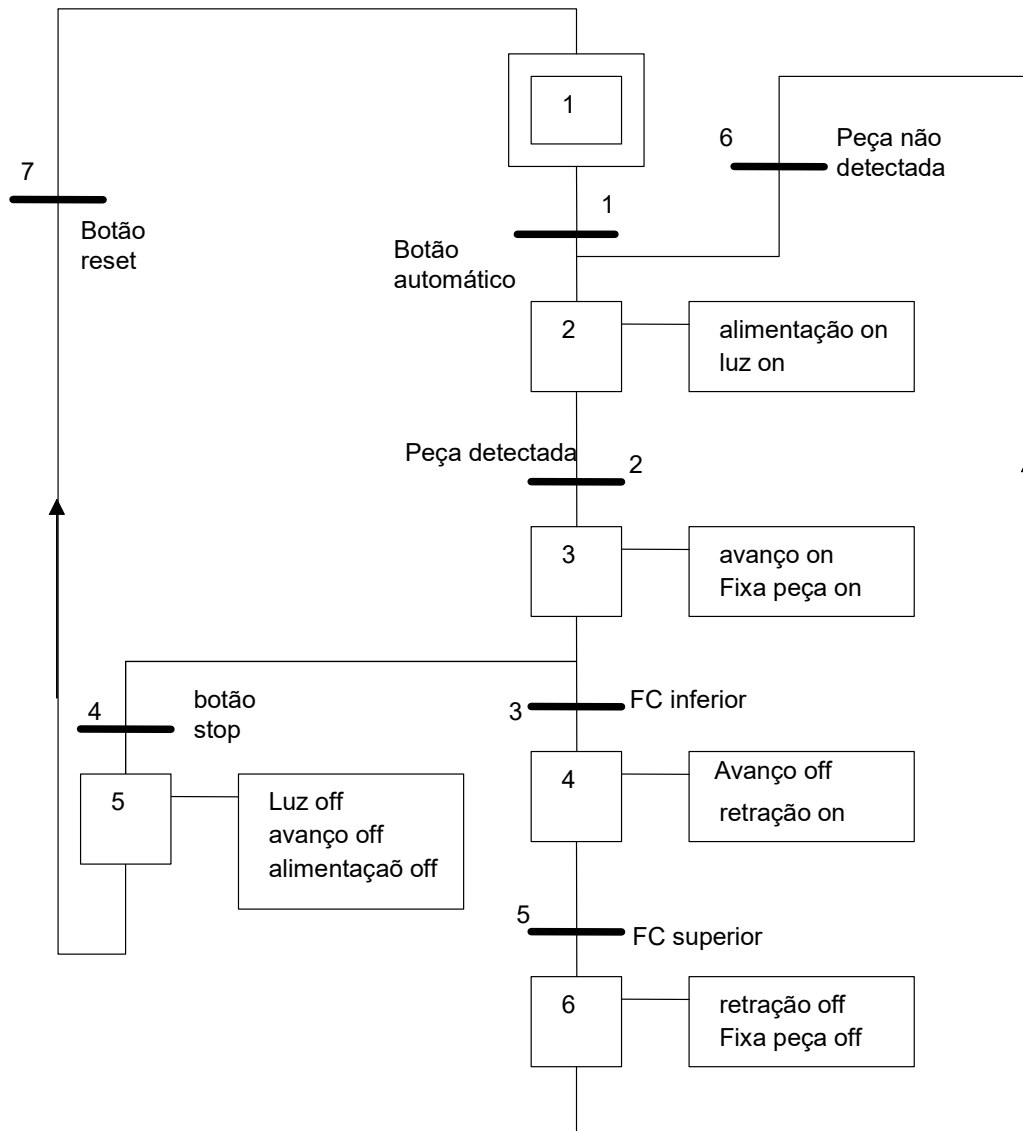


Exemplo: Controle de Prensa

Um SFC para o controle de uma prensa de estampagem é apresentado a seguir. Note que o processo é composto por uma única seqüência, portanto poderia ser implementado diretamente em ladder.

O processo da prensa se inicia em um estado ocioso, quando um botão “automático” é pressionado, a alimentação da prensa e uma luz de indicação são ligados. Frente à detecção de uma peça, o movimento é iniciado até a sensibilização de um fim de curso, então a prensa tem o movimento de retorno acionado até um fim de curso superior ser acionado. Um botão de parada pode ser acionado somente durante o movimento de descida da prensa (note que este tipo de mecanismo de segurança

pode ser ativo durante toda a operação). Quando o sistema é parado, deve-se pressionar um botão de reset antes de se iniciar novamente a operação automática. Após o passo 6, a prensa aguarda a não detecção da peça antes de aguardar por uma nova peça. Sem este mecanismo a prensa entraria em um ciclo interminável.



Exercícios

5.1. Crie um SFC para uma estação de trabalho para 2 pessoas. A estação tem duas prensas que podem ser utilizadas ao mesmo tempo. Cada prensa tem um botão de ciclo que inicia o avanço. Um fim

de curso inferior interrompe o avanço, e o cilindro retrai até o acionamento de um fim de curso superior.

5.2. Crie um SFC para o controle de um semáforo de trânsito. Devem-se considerar botões para a solicitação de travessia de pedestres em ambos os sentidos. Uma seqüência normal de operação deve ser: Sinal verde por 16 segundos, Amarelo por 4 segundos e Vermelho por 16 segundos. Se um dos botões de pedestres for acionado, uma luz de travessia de pedestre deve ser acesa por 10 segundos e o sinal verde estendido por 24 segundos.

5.3. Desenhe um SFC para o processo de estampagem onde o avanço e a retração são disparados por um botão de ciclo único.

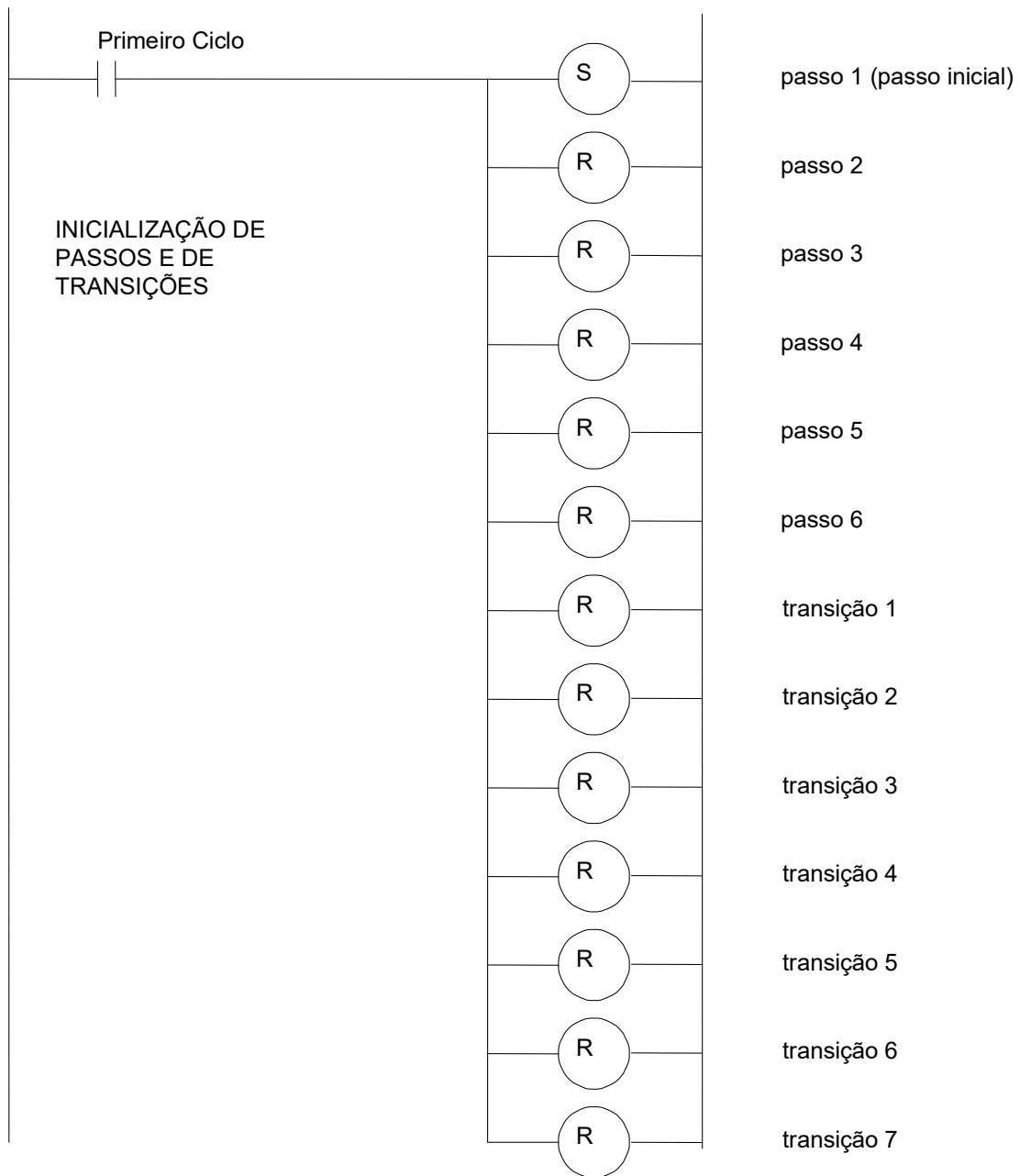
5.4. Projete o controle de um portão automático de garagem:

- existe um único botão na garagem e outro no controle remoto.
- quando o botão é pressionado, o portão abre ou fecha.
- se o botão é pressionado durante o movimento, o portão pára. Um Segundo acionamento do botão inicia o movimento no sentido contrário.
- existem fins de curso nas duas extremidades do movimento do portão.
- existe uma barreira ótica no curso do portão, que interrompe o fechamento e inicia a abertura quando o sinal ótico é bloqueado.
- existe uma sinalização luminosa que se ativa por 1 minuto após a abertura ou o fechamento do portão.

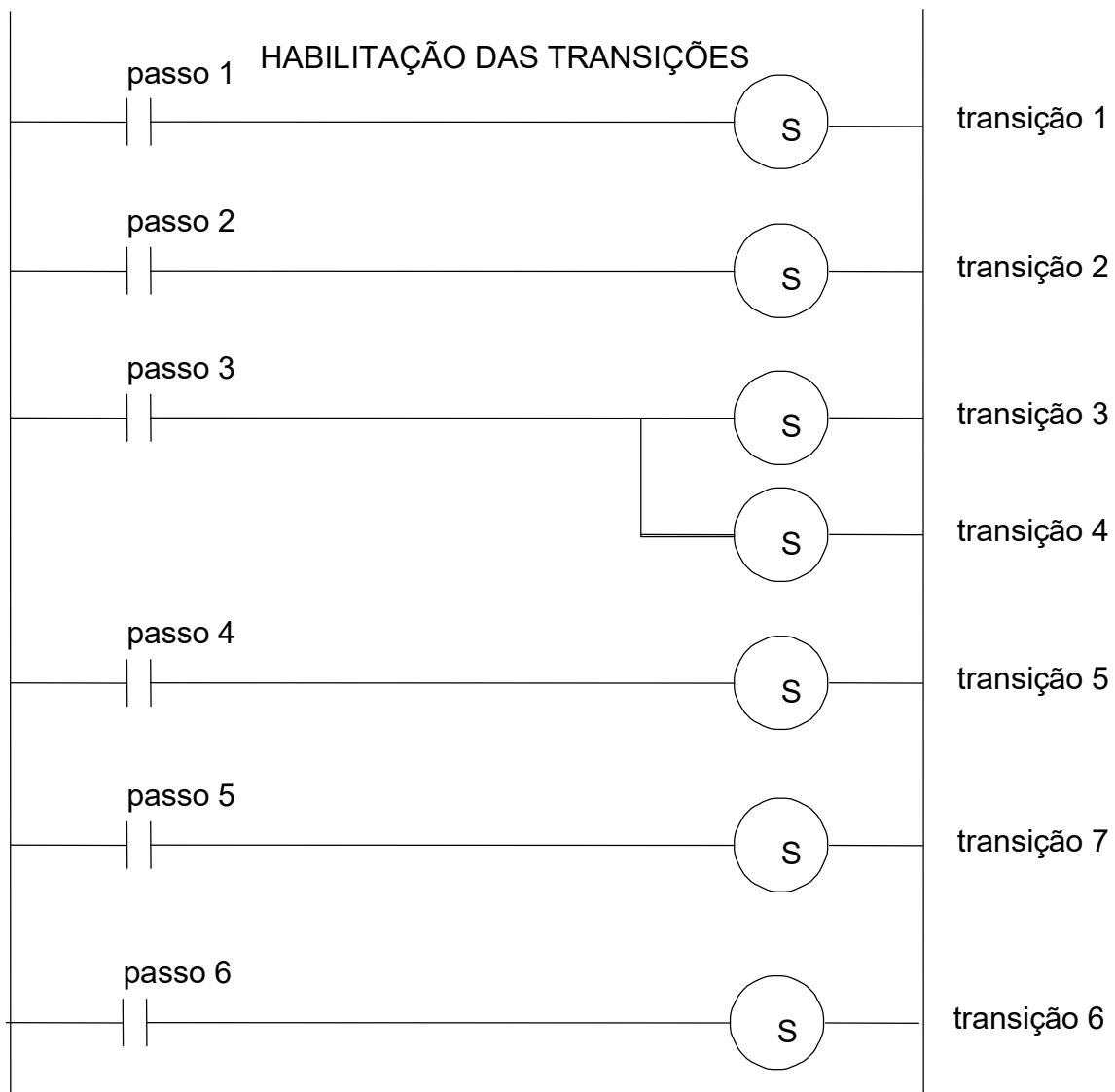
Conversão de diagramas SFC em linguagem Ladder

Um diagrama SFC pode ser convertido diretamente em ladder de algumas formas distintas. O método aqui apresentado deve ser utilizado em seqüência à elaboração do diagrama SFC. O programa ladder começa com uma seção de inicialização de passos e transições para valores únicos. Em uma segunda seção da lógica ladder ativam-se as transições subseqüentes aos passos. Em uma terceira seção, caso determinada condição de transição seja satisfeita, então o próximo passo é ativado e o passo precedente, bem como a própria transição em questão, são desativados. Segue-se a lógica ladder com a codificação da ação associada a cada passo, com as ativações de saídas e lógicas locais em uma última seção.

A seguir, demonstra-se a codificação em ladder do exemplo 5.2. Controle de Prensa.

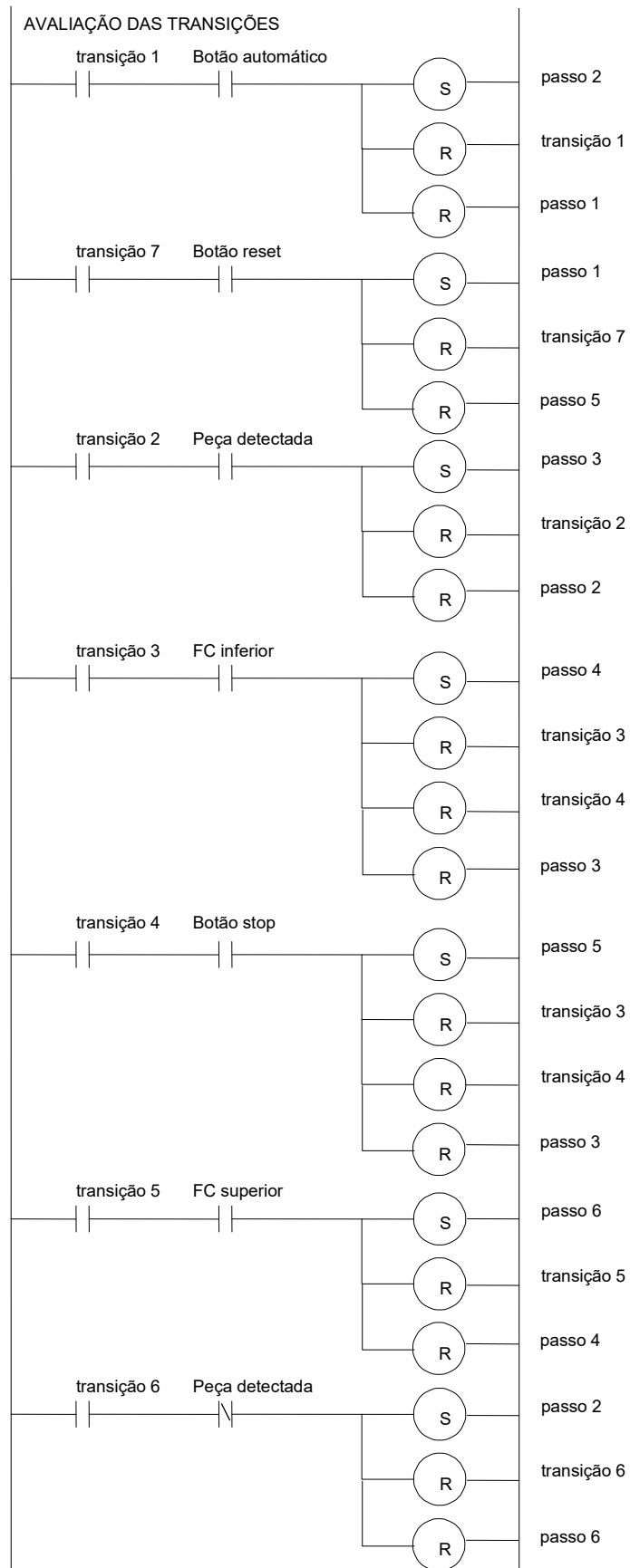


Nesta primeira seção do ladder, inicializam-se todas as variáveis relacionadas a passo e transições. Note que são seis passos e sete transições nomeados de forma única no programa. Deve-se sempre nomear passos e transições de forma única. Em geral os CLPs possuem uma variável de sistema que pode ser utilizada para a ativação das inicializações somente no primeiro ciclo de scan do programa. Caso o controlador não tenha este recurso, é necessária a utilização de uma lógica para realizar tal função.

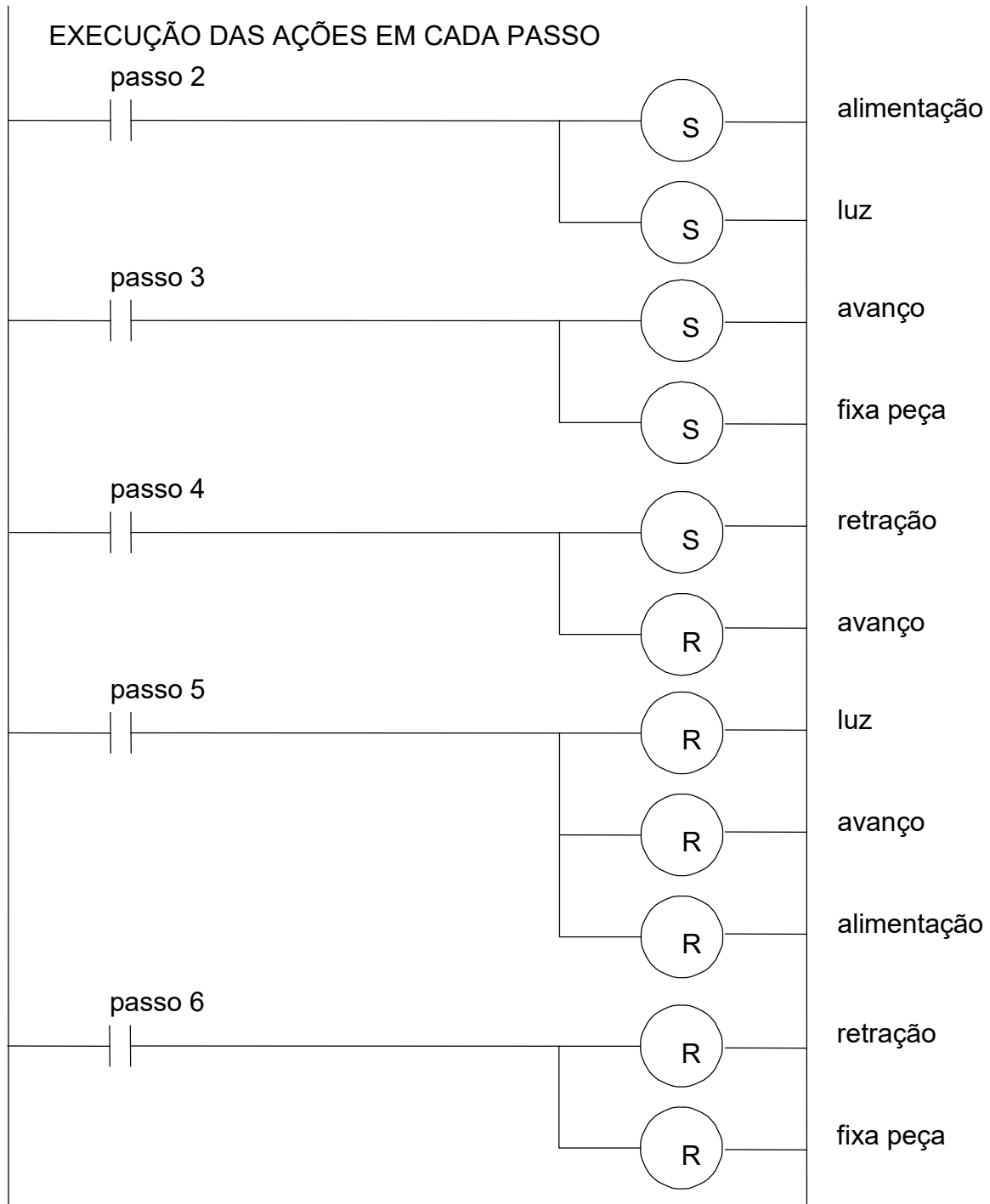


Na seção de habilitação das transições, deve-se reproduzir a estrutura do SFC em relação às seqüências passos-transições.

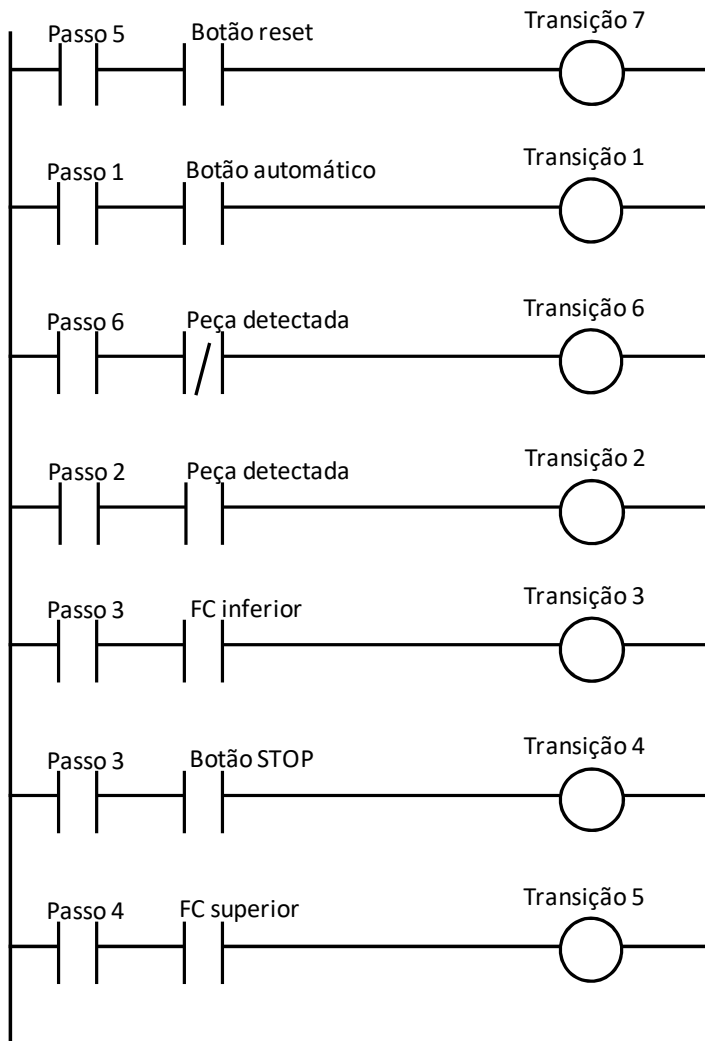
Na seção seguinte, cada transição deve ter sua condição de disparo avaliada e, caso a avaliação seja positiva, deve-se ativar os passos seguintes, desativar o passo anterior e desativar a própria transição em questão.

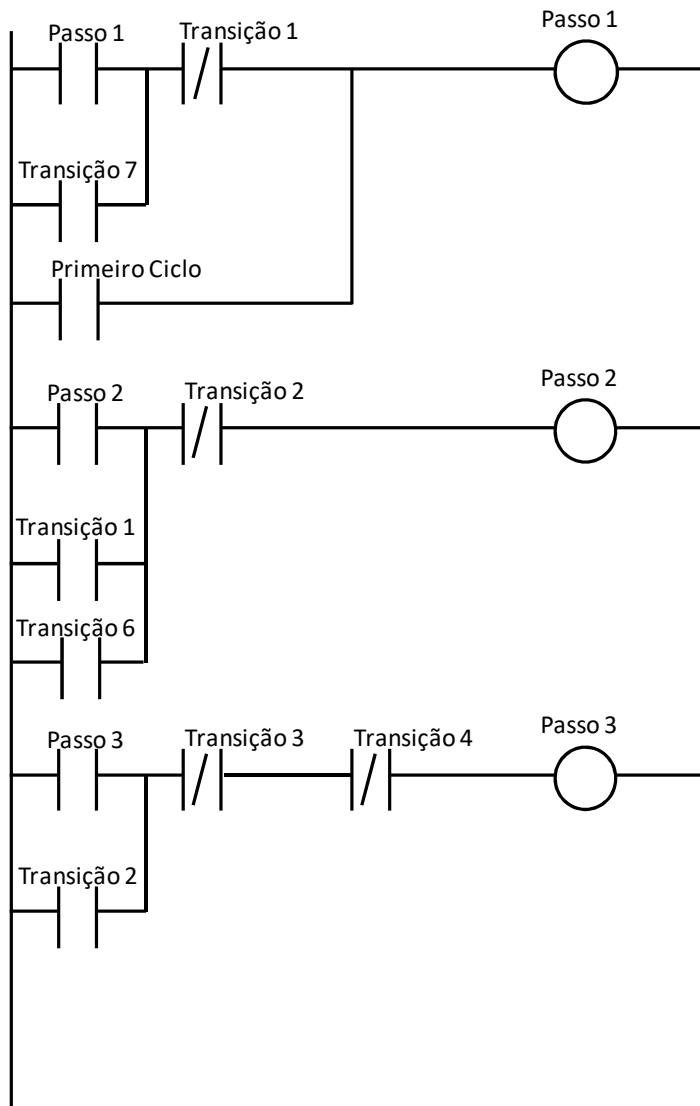


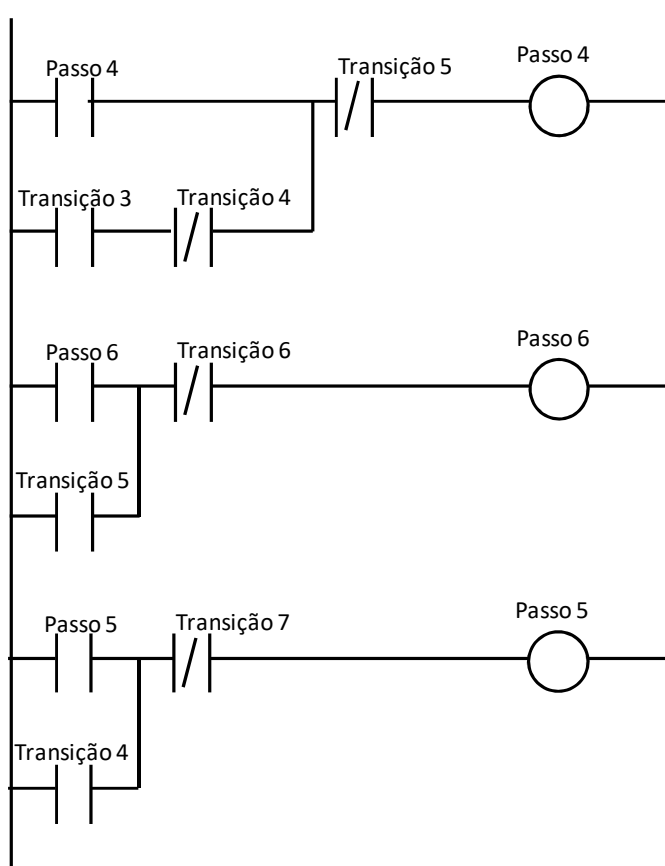
Por fim, executam-se as ações atribuídas a cada passo. Note que nesta seção do diagrama ladder o passo 1 não foi considerado por não ter nenhuma ação associada.



Pode-se alternativamente proceder com a conversão em ladder sem o uso de bobinas do tipo Set/Reset através do método exemplificado abaixo para o mesmo exemplo da prensa. Seguem as lógicas para a ativação das transições e para os passos.







Alguns CLPs permitem a programação gráfica diretamente do SFC. Mesmo assim, códigos em ladder, lista de instruções, blocos funcionais ou texto estruturado devem ser programados para cada transição ou passo. Cada código então deve ser separado em um programa distinto nesta lógica.

Exercícios

5.5. a 5.9. Codifique em ladder os SFC's projetados nos exercícios 5.1 a 5.4 e no exemplo 5.1 respectivamente.