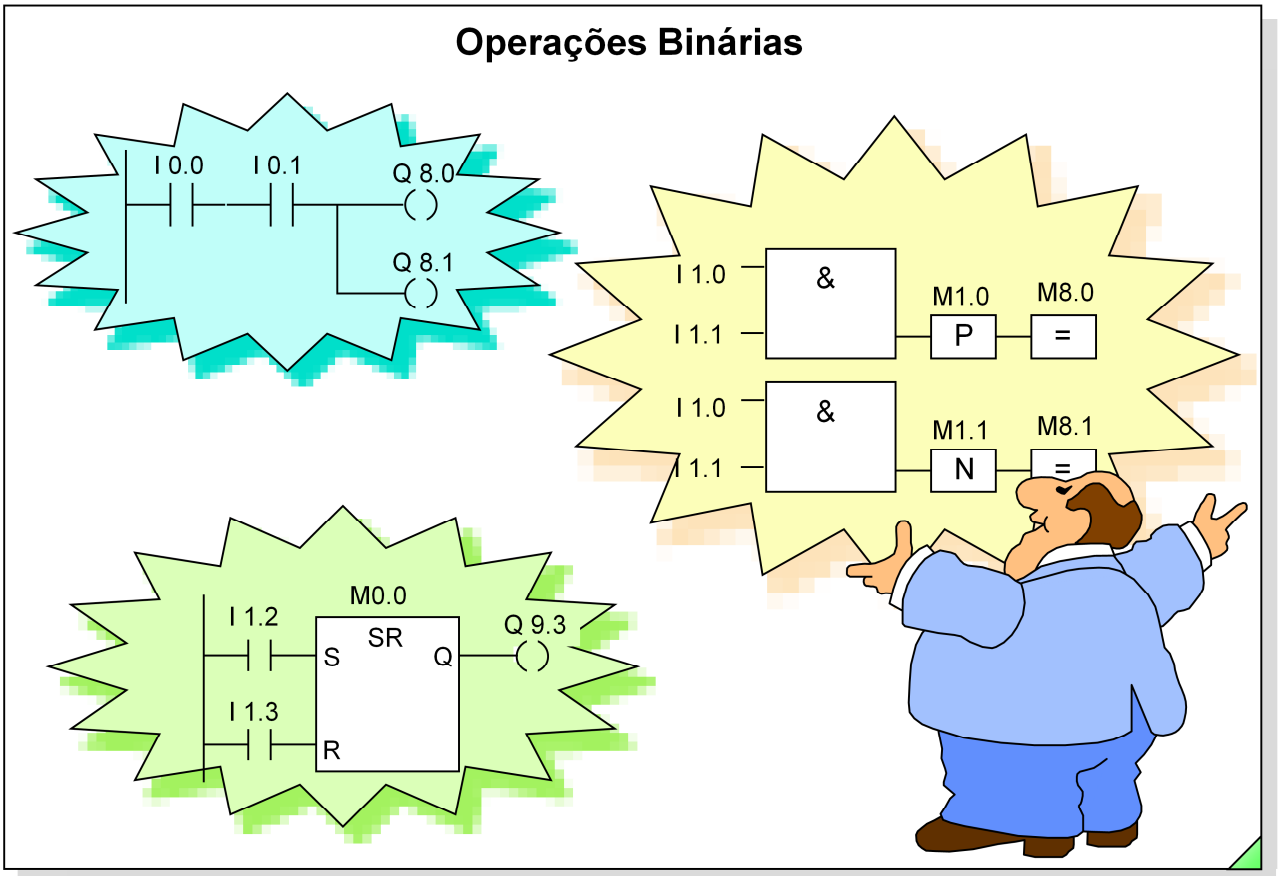
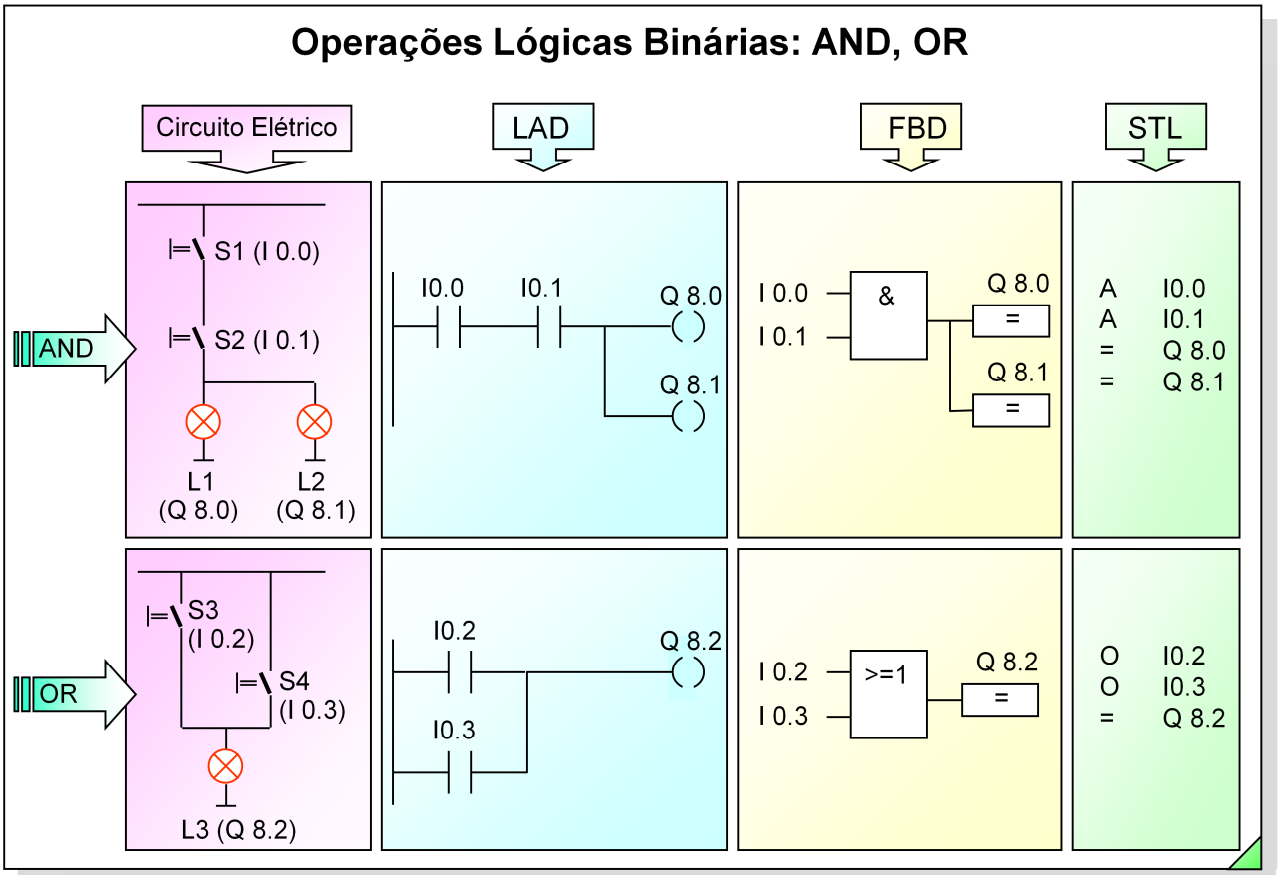


SIMATIC S7

Data: 26/08/2011
Arquivo: S7-Bas-06.1

sitrain

Conteúdo Página



SIMATIC S7

Data: 26/08/2011
Arquivo: S7-Bas-06.2

sitrain

Tabelas Lógicas

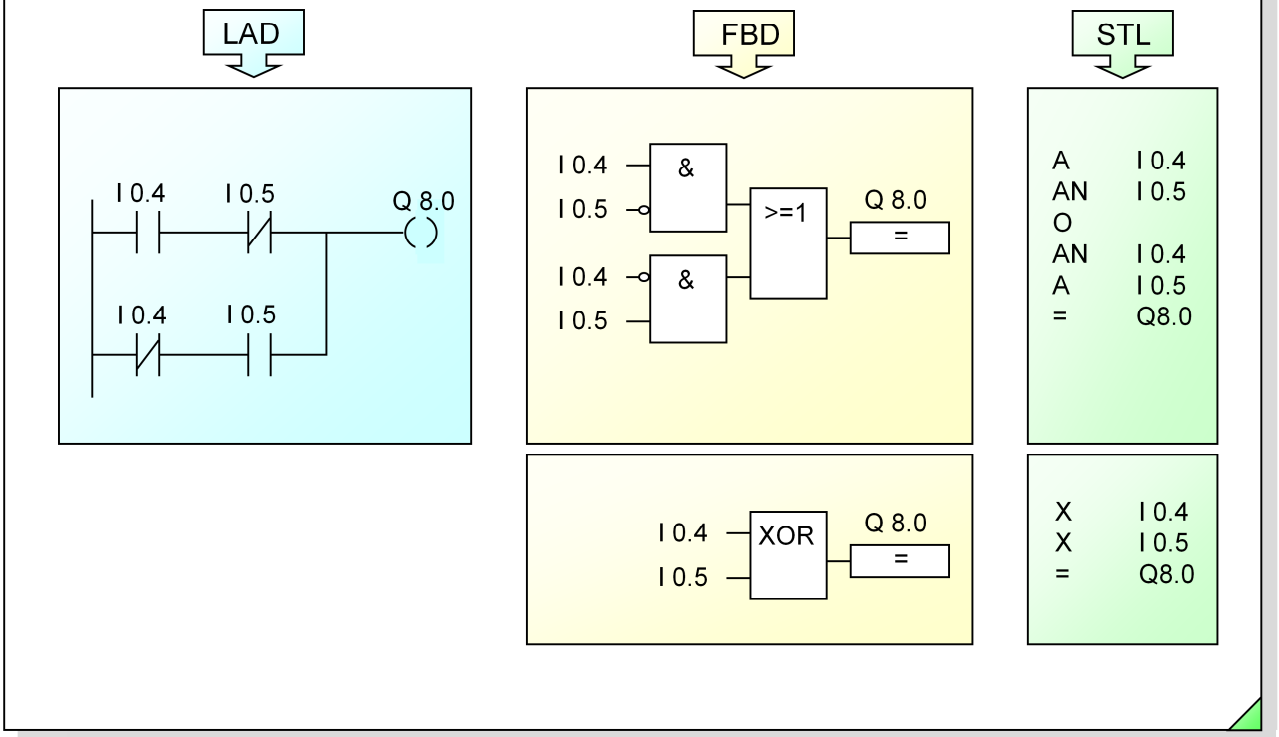
AND

I 0.0	I 0.1	Q 8.0
0	0	
0	1	
1	0	
1	1	

OR

I 0.2	I 0.3	Q 8.2
0	0	
0	1	
1	0	
1	1	

Operações Lógicas Binárias: OR Exclusivo (XOR)



SIMATIC S7

Data: 26/08/2011
Arquivo: S7-Bas-06.3

sitrain

Tabela Lógica

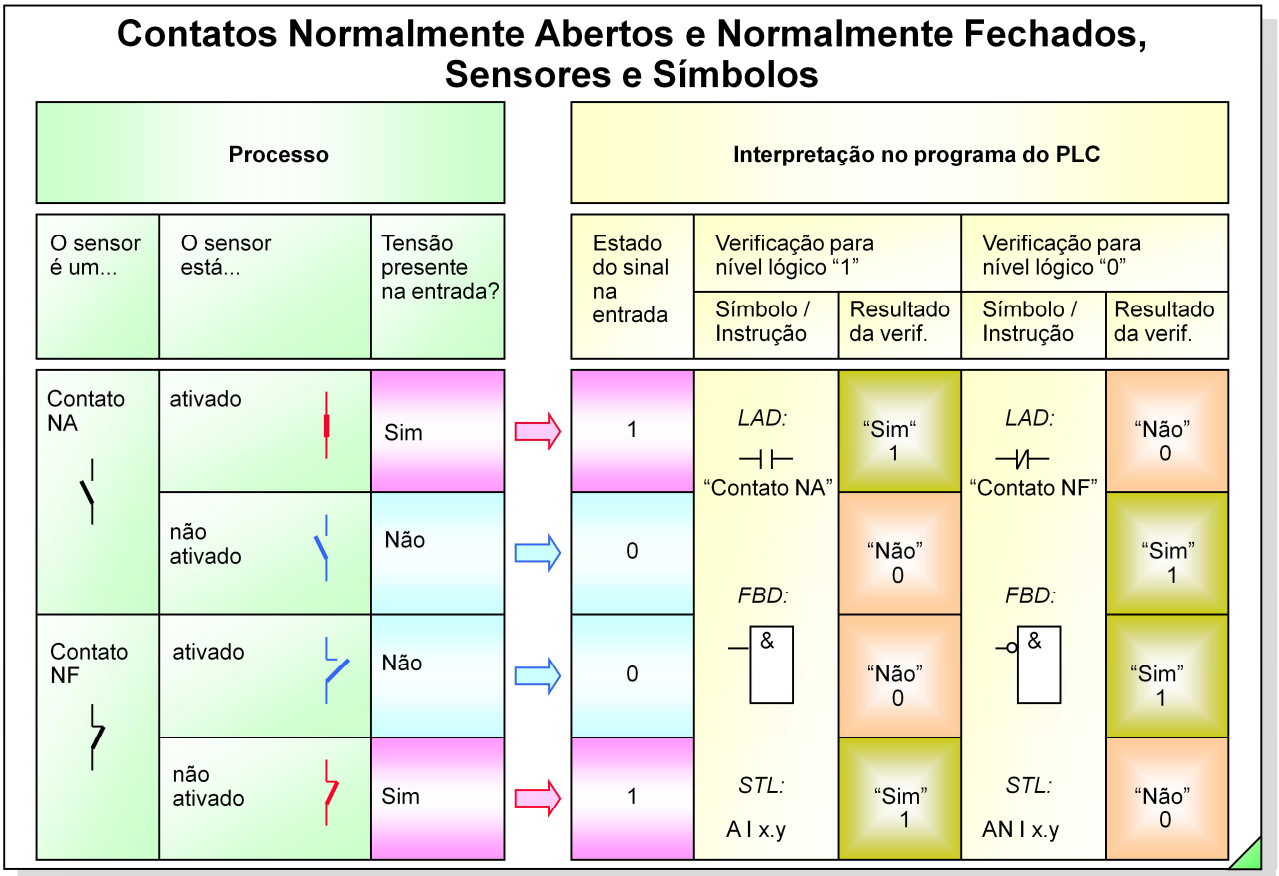
XOR

I 0.4	I 0.5	Q 8.0
0	0	
0	1	
1	0	
1	1	

Regra A regra seguinte é válida para a operação lógica XOR com duas entradas: a saída terá nível lógico "1" quando uma e somente uma das duas entradas estiver ativada.

Atenção! Essa regra não pode ser generalizada para “uma e apenas uma de n entradas" para a operação lógica XOR com várias entradas!

Se houver uma terceira instrução XOR, o RLO anterior é comparado na instrução XOR seguinte.



Processo

A utilização de contatos normalmente abertos ou normalmente fechados para os sensores em um processo controlado depende das regras de segurança do próprio processo.

Os contatos normalmente fechados são sempre utilizados para chaves de limite e interruptores de segurança, de forma que não apareçam situações de perigo se houver uma quebra de fio no circuito do sensor.

Os contatos normalmente fechados são também utilizados para desligar as máquinas, pela mesma razão.

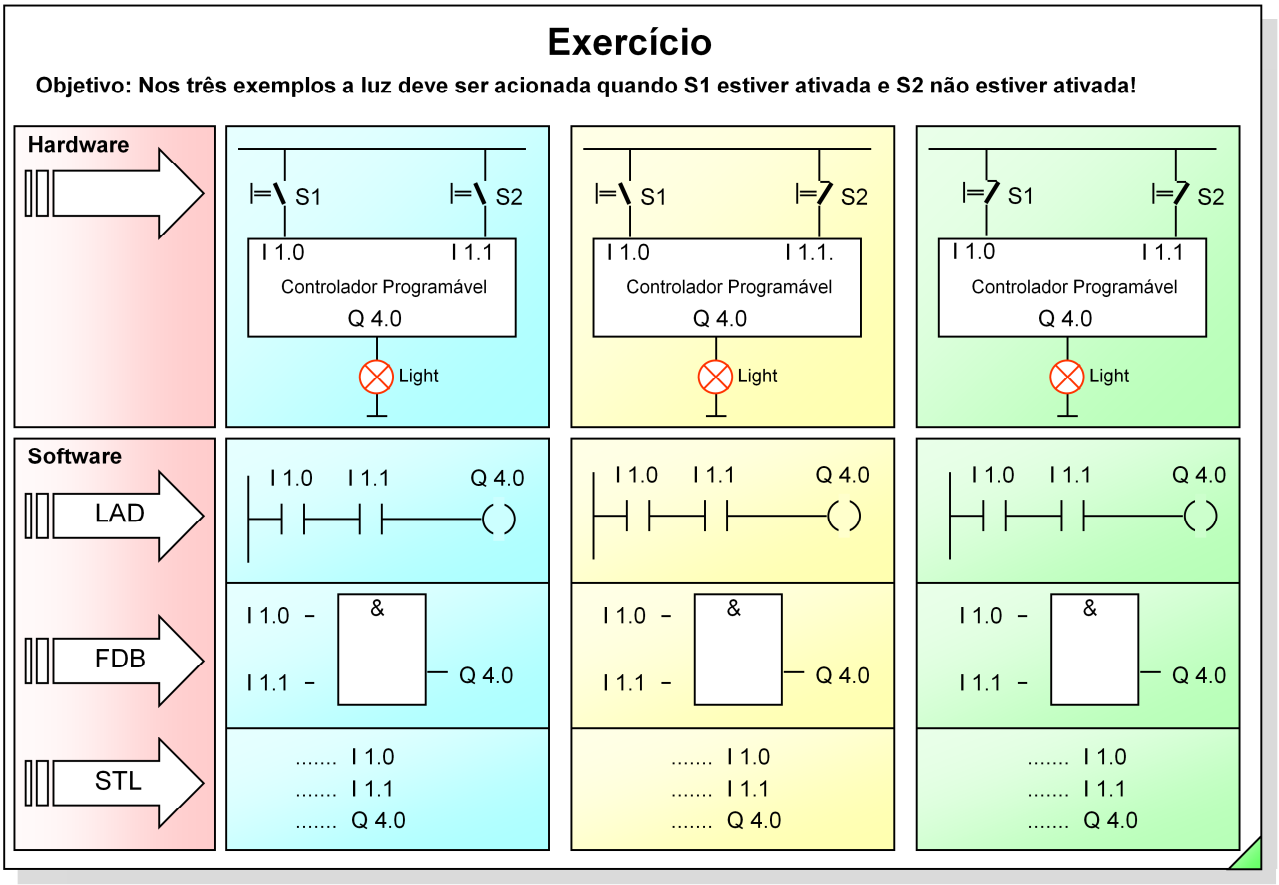
Símbolos

Em LAD, um símbolo com o nome ”Contato NA" é utilizado para fazer a verificação do estado lógico “1” e um símbolo com o nome “Contato NF” para verificar o estado lógico ”0”.

Não faz nenhuma diferença se o sinal “1” do processo é fornecido por um contacto NA ativado ou por um contacto NF não-ativado.

Exemplo

O resultado da verificação para o símbolo “Contato NA“ é “1“ se um contato NF na máquina não estiver ativado.



SIMATIC S7

Data: 26/08/2011
Arquivo: S7-Bas-06.5

sitrain

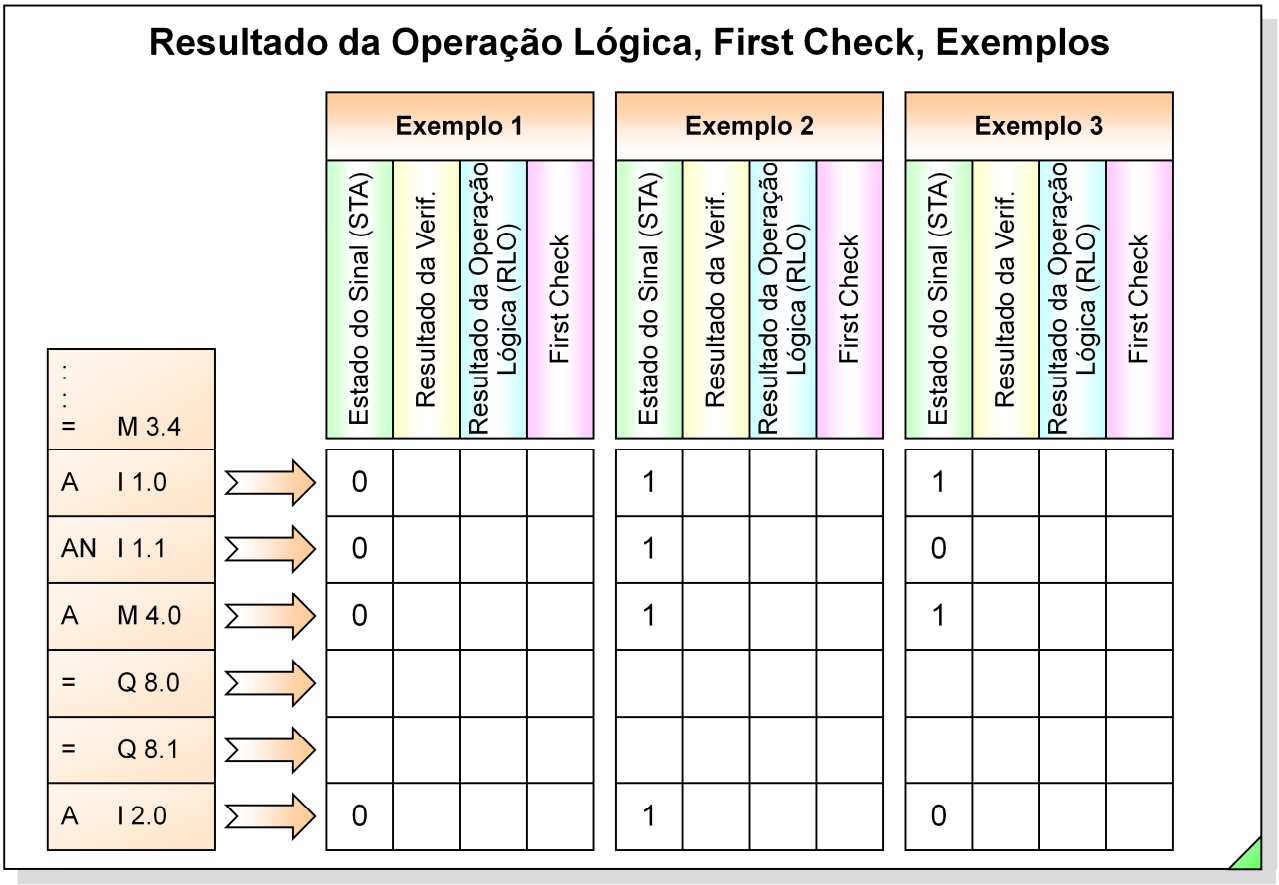
Exercício

Complete os programas acima ilustrados de forma a obter a seguinte funcionalidade: Quando o interruptor S1 estiver ativado e o interruptor S2 não estiver a luz deverá acender-se (nos três casos).

Nota !

Os termos ”Contato NA" e ”Contato NF" possuem diferentes significados, dependendo se são utilizados no contexto do hardware do processo ou como símbolos no software.

Resultado da Operação Lógica, First Check, Exemplos



SIMATIC S7

Data: 26/08/2011
Arquivo: S7-Bas-06.6

sitrain

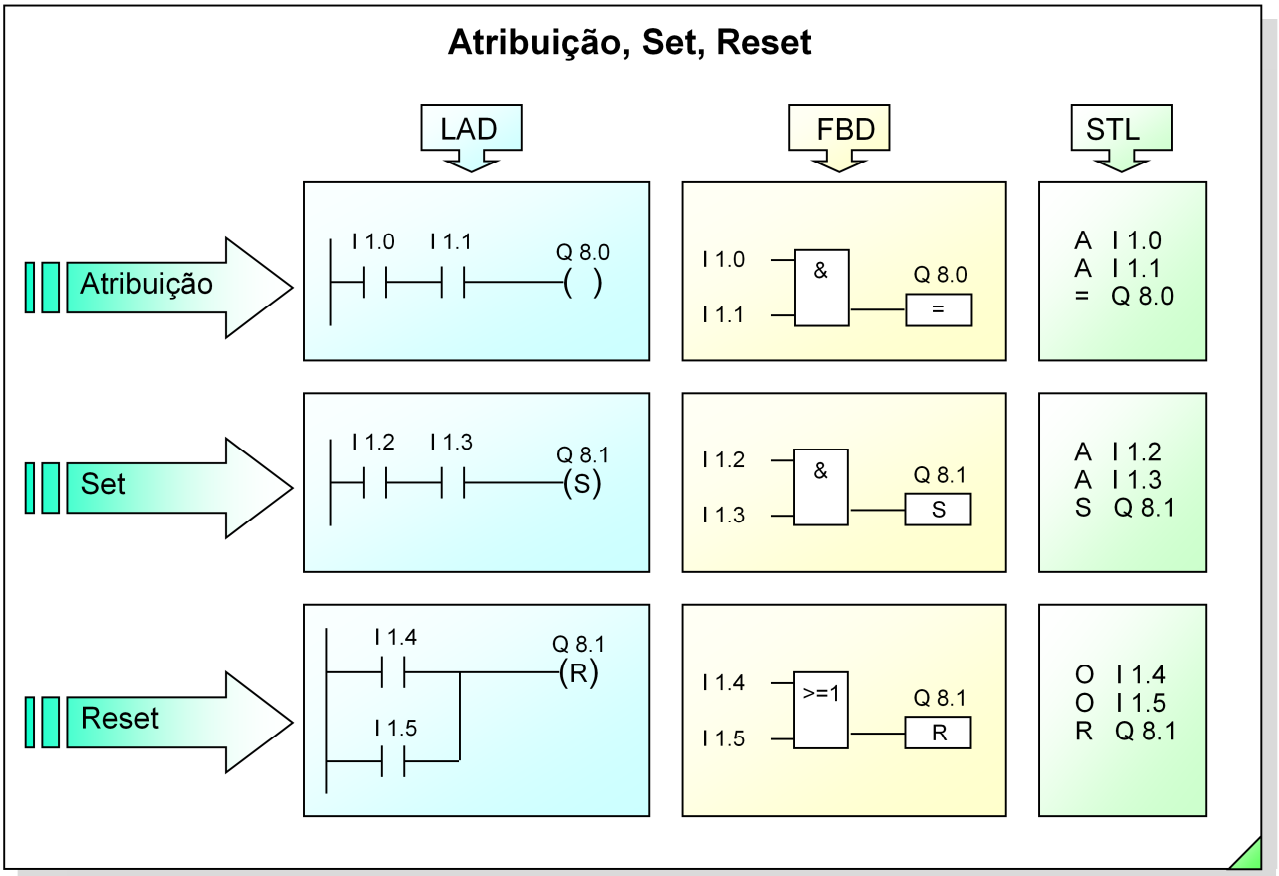
Estado do Sinal Uma operação lógica é constituída por um conjunto de instruções para verificar os estados dos sinais (entradas (**I**), saídas (**Q**), bits de memória (**M**), temporizadores (**T**), contadores (**C**) ou bits de dados (**D**)) e instruções para atribuir estados lógicos a Q, M, T ,C ou D.

Resultado da Verif. Quando o programa é executado, é obtido o resultado da verificação. Se a condição de verificação é verdadeira, o resultado da verificação é “1“. Se não for verdadeira, o resultado da verificação é “0“.

First Check O resultado da primeira verificação é armazenado como o resultado lógico da operação (RLO).

Resultado Lógico da Operação Quando a próxima verificação de instruções é executada, o resultado lógico da operação é associado ao resultado da verificação, e dessa forma é obtido o novo RLO.
Na última instrução de verificação de uma operação lógica o RLO permanece o mesmo. Sendo assim, o mesmo RLO pode ser utilizado nas instruções seguintes.

Nota O resultado da primeira verificação é armazenado sem ter sido submetido a uma operação lógica. Sendo assim não faz qualquer diferença se o programa faz a primeira verificação com um AND ou com um OR, em STL. Para que o programa possa ser convertido em uma das outras linguagens de programação, deve-se, contudo, programar utilizando sempre a instrução correta.

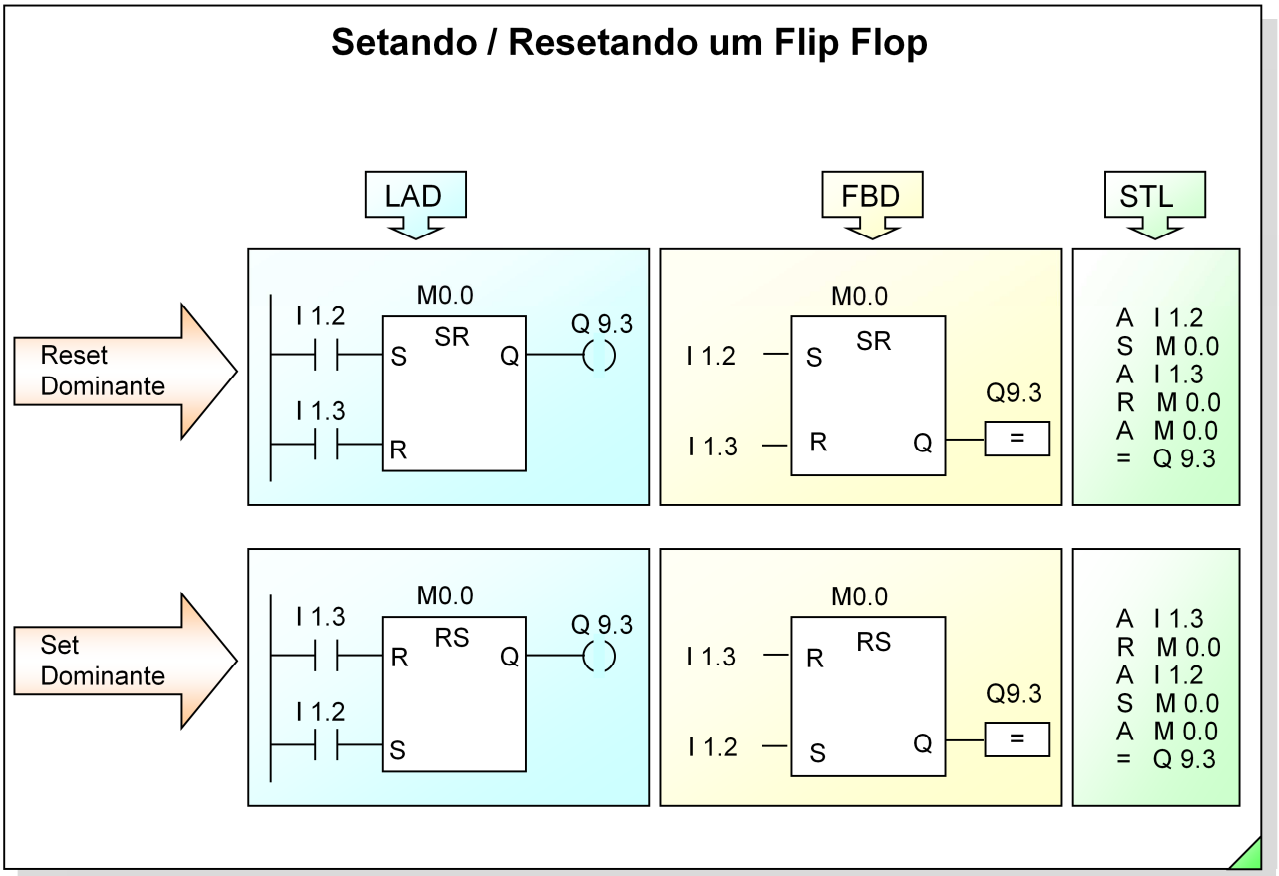


Data: 26/08/2011
Arquivo: S7-Bas-06.7

Atribuição Uma atribuição transfere o RLO para os endereços especificados (Q, M, D). Quando o RLO muda, o estado do sinal desse endereço também é alterado.

Set Se o RLO= "1", o endereço especificado é setado com nível lógico ”1”, e assim permanece até que seja feito um reset através de outra instrução.

Reset Se o RLO= "1", o endereço especificado é resetado para o nível lógico ”0” e assim permanece até que seja feito novamente um set através de outra instrução.



Flip Flop

Um flip flop possui uma entrada de Set e uma entrada de Reset. O bit de memória é setado ou resetado, dependendo da entrada que possuir RLO = ”1”.

Se por alguma razão ambas as entradas possuírem simultaneamente RLO=1, a prioridade deve ser determinada.

Prioridade

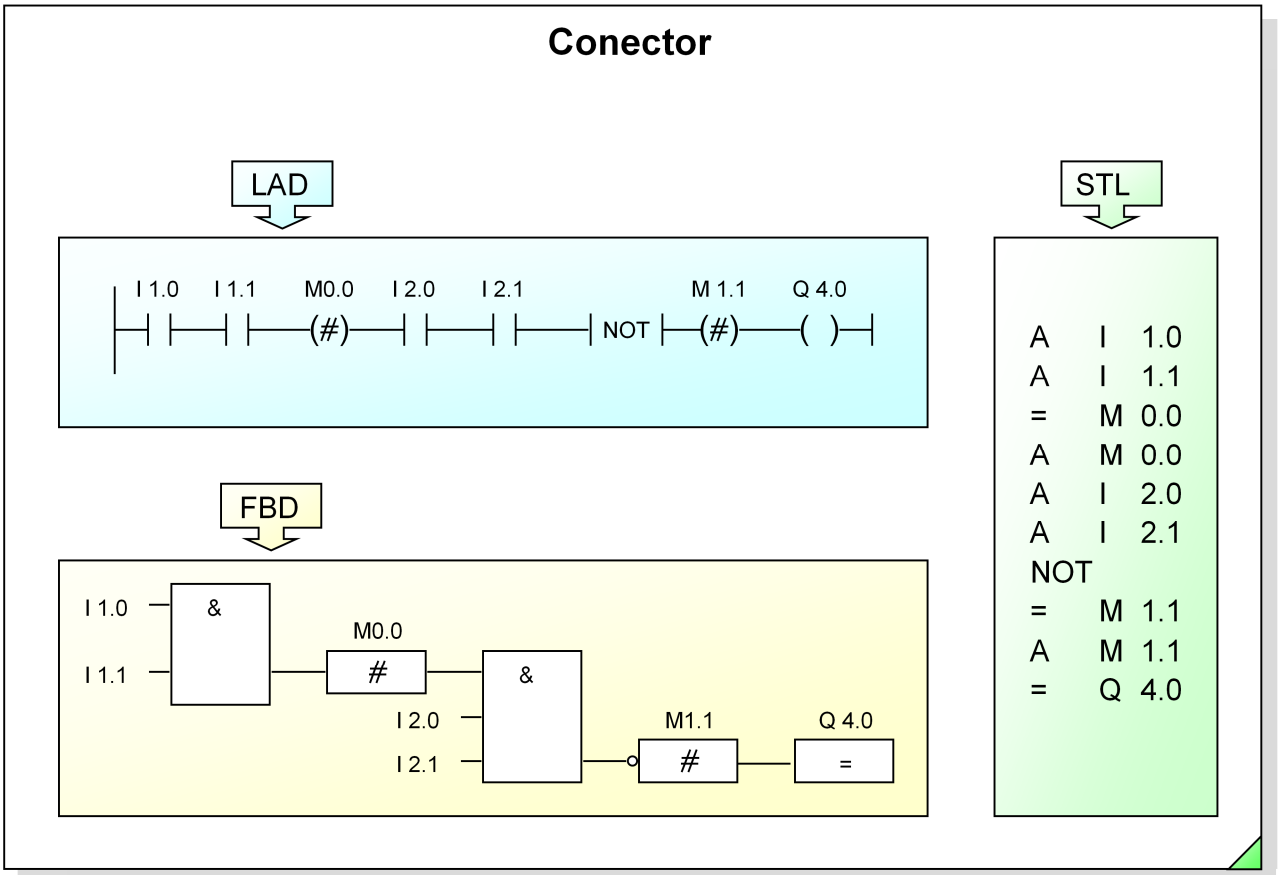
Em LAD e FBD existem diferentes símbolos para as funções Set Dominante e Reset Dominante.

Em STL, a instrução que for programada em último lugar tem prioridade.

Nota

Se uma saída é ativada através de uma instrução Set, ela é resetada num restart completo da CPU.

Se M 0.0 (no exemplo acima) tiver sido declarada como retentiva, ela permanecerá setada após um restart completo da CPU, e a saída Q 9.3 (que tinha sido desligada) terá novamente o estado “1”.



Conector

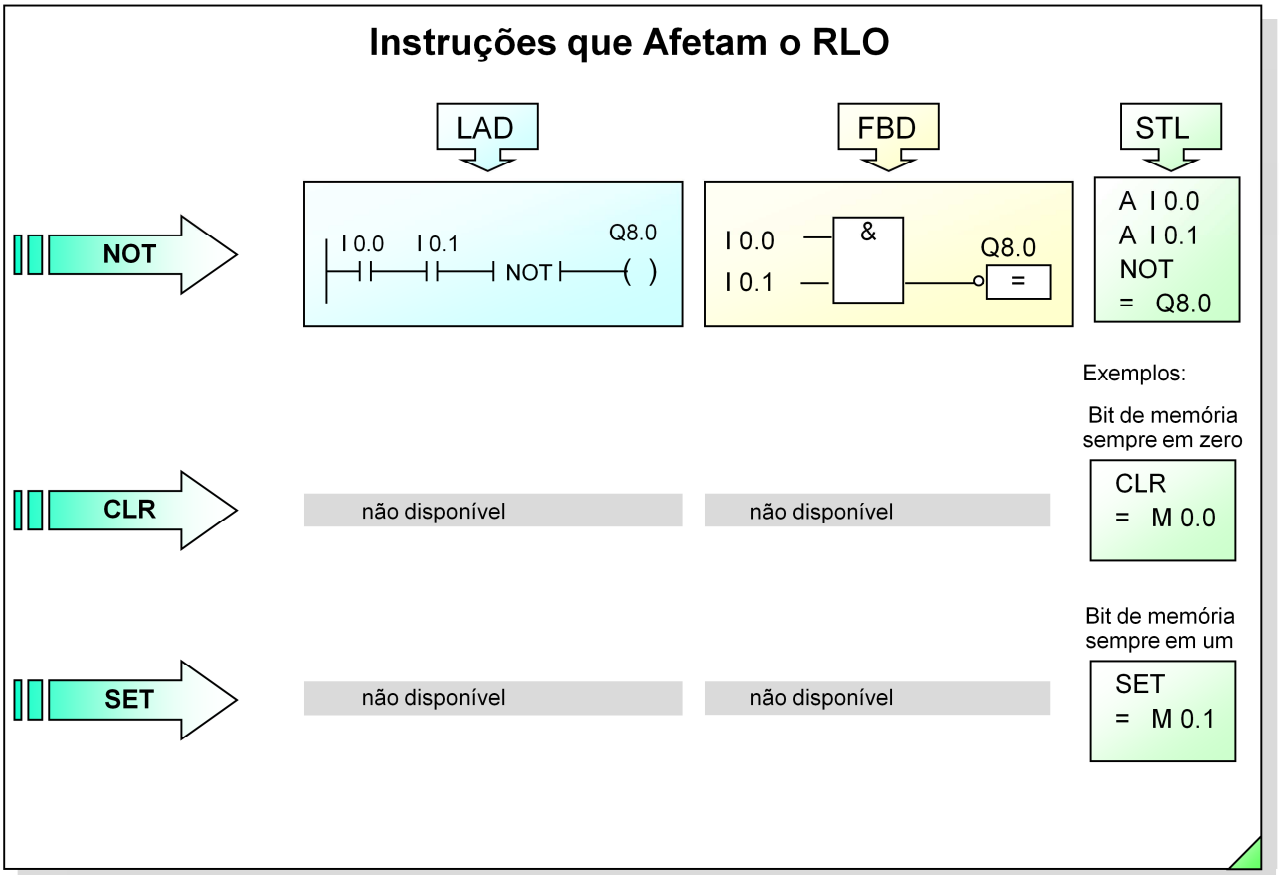
Um conector corresponde a um elemento intermediário de atribuição de sinal que armazena o RLO no endereço especificado.

Quando ligado em série com outros elementos, a instrução “Conector” é inserida da mesma forma que um contato normal.

Um conector nunca deve:

- ser conectado ao início de um ramo (LAD);
- seguir diretamente um ramo;
- ser usado no final de um ramo.

Pode-se programar um conector negado com um elemento “NOT”.



SIMATIC S7

Data: 26/08/2011
Arquivo: S7-Bas-06.10

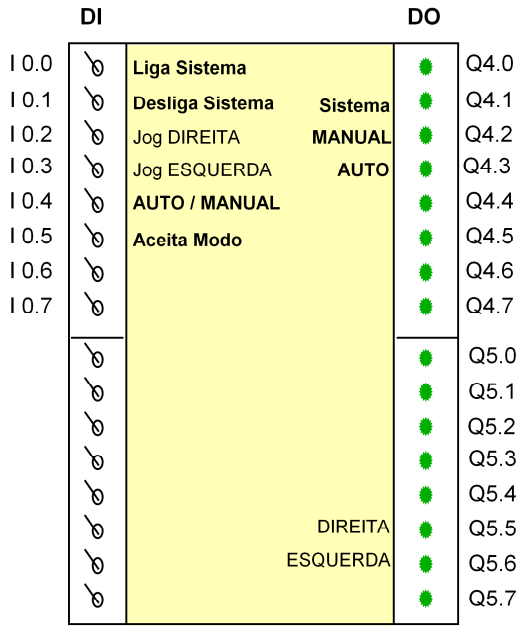
sitrain

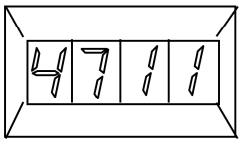
NOT A instrução NOT inverte o RLO.

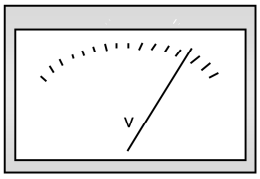
CLR A instrução CLEAR seta o RLO para "0" sem pré-condições (apenas disponível em STL atualmente).
A instrução CLR finaliza o RLO, isto é, a próxima instrução é tratada como verificação inicial (first check).

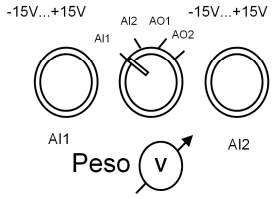
SET A instrução SET seta o RLO para "1" sem pré-condições (apenas disponível em STL atualmente).
A instrução SET finaliza o RLO, isto é, a próxima instrução é tratada como verificação inicial (first check).

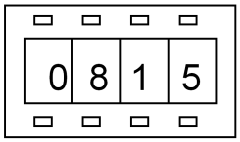
Exercício: Seleção de Modo do Transportador

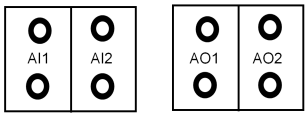












SIMATIC S7

Data: 26/08/2011
Arquivo: S7-Bas-06.11

sitrain

Objetivo

Programar a seleção de modo de operação do transportador no FC 15 e integrar o estado de modo MANUAL (Q 8.2 ou Q 4.2) como intertravamento no bloco FC 16.

Função de seleção de modo no FC 15:

- O sistema (LED Q 4.1) é acionado através da chave não-retentiva I 0.0, e desligado pela chave não-retentiva I 0.1.
- O modo MANUAL (LED Q 4.2) e o modo AUTO (LED Q 4.3) pode ser pré-selecionado através da chave I 0.4 como segue:
  - I 0.4 desligada (= ´0´): MANUAL selecionado;
  - I 0.4 ligada (= ´1´): AUTO selecionado.
- O modo de operação pré-selecionado pela chave I 0.4 é aceito (ou ligado) através da chave não-retentiva I 0.5.
- Os modos de operação são desligados quando a pré-seleção de modo é alterada (I 0.4) ou quando o sistema é desligado (Q 4.1 = ´0´).

Integrando o modo MANUAL (Q 4.2):

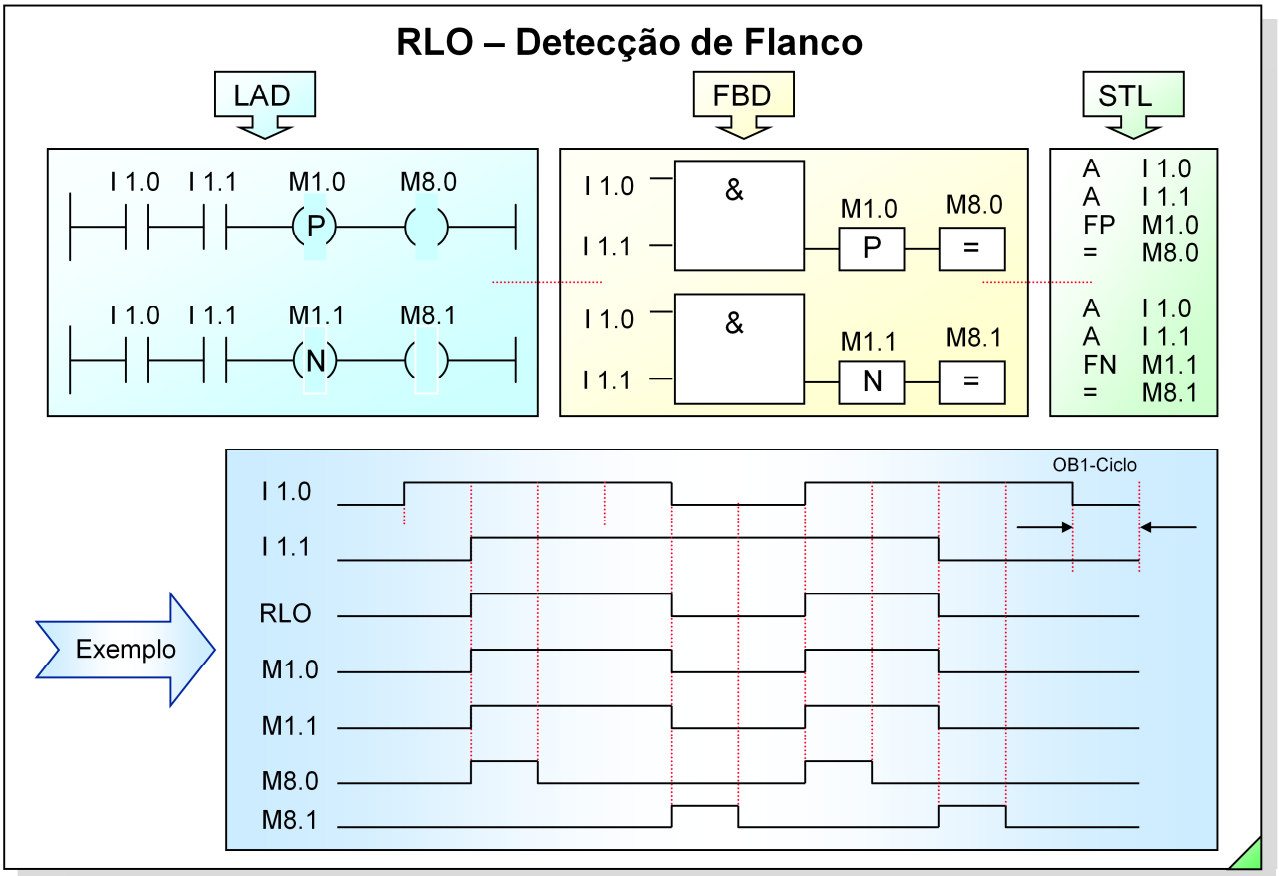
- O "Jog de Motor do Transportador" programado no FC 16 deve ser apenas possível quando o modo MANUAL é acionado. Programe o intertravamento relevante no FC 16.

O Que Fazer

- Insira o novo bloco FC 15 no "Programa_1" e programe-o de acordo com o especificado.
- Programe a chamada do bloco FC 15 no OB 1.
- Programe o intertravamento relevante no FC 16.
- Transfira todos os blocos para a CPU e teste o programa.

Nota de Solução

O acionamento / desligamento do sistema e os modos de operação são feitos com chaves não-retentivas. A forma mais simples de implementar as funções é através da utilização de memórias.



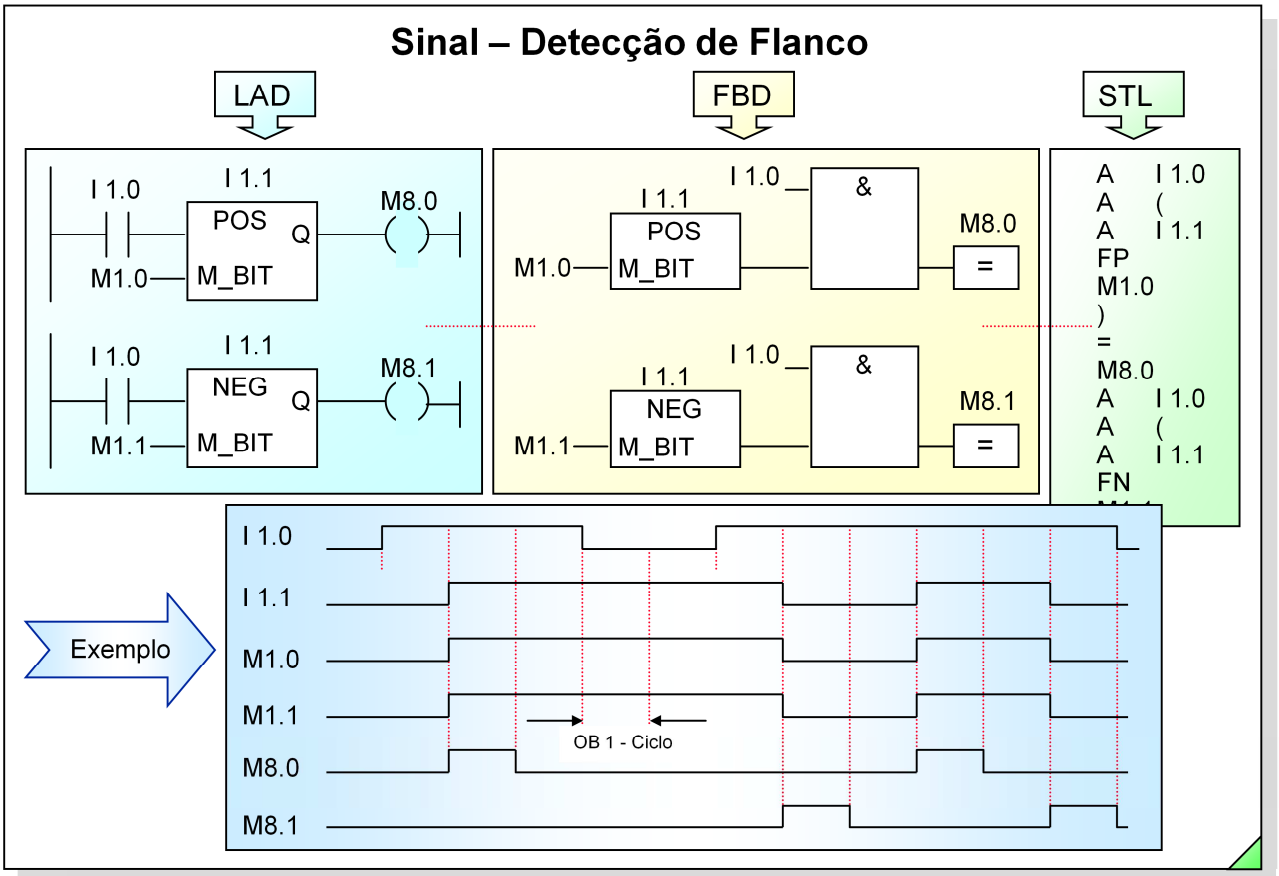
Flanco do RLO Um ”Flanco do RLO” acontece quando o resultado lógico de uma operação muda.

Flanco Positivo Quando o RLO muda de “0” para “1”, a instrução de detecção de flanco "FP" resulta no estado de sinal "1" (por ex. na M 8.0) durante um ciclo.

Para permitir que o sistema detecte a mudança de flanco, o RLO deve também ser salvo num bit de memória FP, ou bit de dados (por ex. M 1.0)

Flanco Negativo Quando o RLO muda de “1” para “0”, a instrução de detecção de flanco "FN" resulta no estado de sinal ”1” (por ex. M 8.1) durante um ciclo.

Para permitir que o sistema detecte a mudança de flanco, o RLO deve também ser salvo num bit de memória FN, ou bit de dados (por ex. M 1.1).



Data: 26/08/2011
Arquivo: S7-Bas-06.13
sitrain

Flanco de Sinal

Um "flanco de sinal" acontece quando o sinal muda o seu estado.

Exemplo

A entrada I 1.0 funciona como uma entrada estática de habilitação. A entrada I 1.1 é monitorada dinamicamente e cada mudança de sinal é detectada.

Flanco Positivo

Quando o estado do sinal I 1.1 passa de “0“ para “1“, a instrução de verificação “POS“ resulta no estado lógico “1“ na saída Q durante um ciclo, desde que I 1.0 tenha estado lógico “1“ (como no exemplo acima ilustrado).

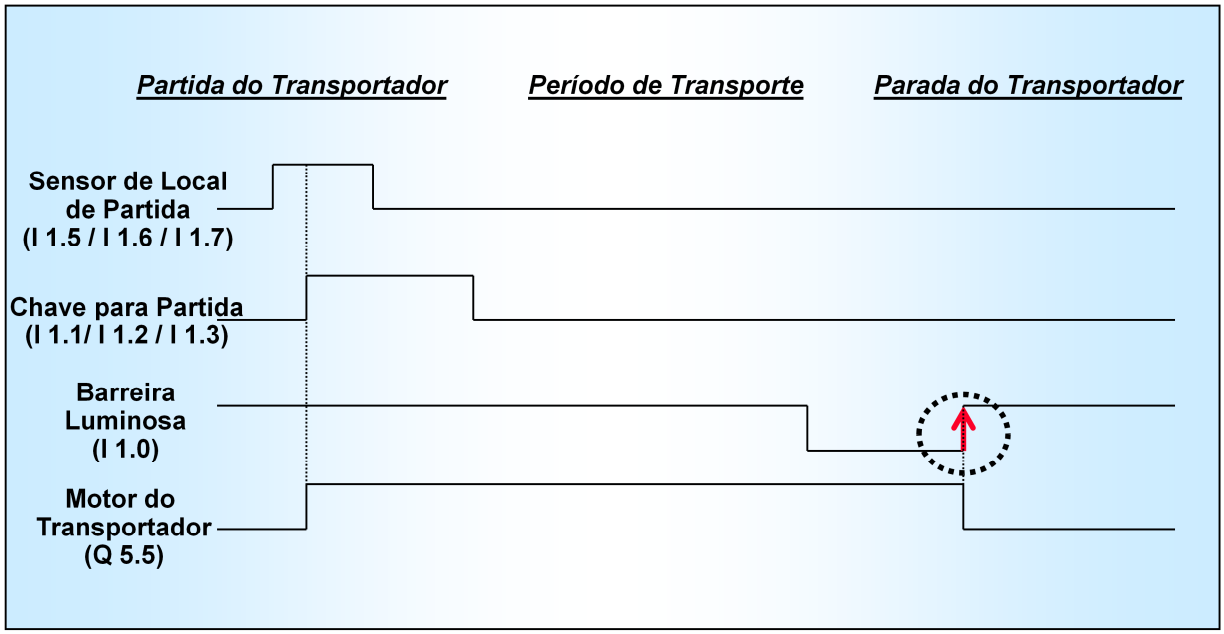
Para permitir que o sistema detecte a mudança de flanco, o estado de I 1.1 deve também ser salvo num M_BIT (bit de memória ou bit de dados – por exemplo M 1.0).

Flanco Negativo

Quando o estado de sinal I 1.1 passa de “1” para “0”, a instrução de verificação "NEG" resulta no estado lógico ”1” na saída Q durante um ciclo, desde que I 1.0 tenha estado lógico ”1” (como no exemplo acima ilustrado).

Para permitir que o sistema detecte a mudança de flanco, o estado de I 1.1 deve também ser salvo num M_BIT (bit de memória ou bit de dados – por exemplo M 1.1).

Exercício: Movimento do Transportador no Modo AUTO



SIMATIC S7

Data: 26/08/2011
Arquivo: S7-Bas-06.14

sitrain

Funcionamento Atual no FC 16

Quando o modo MANUAL (Q 4.2 = ´1´) é acionado, pode-se movimentar em jog o motor do transportador para a DIREITA e para a ESQUERDA usando a chave não-retentiva do simulador.

Objetivo:

Expandir o FC 16 de modo a controlar o motor do transportador como segue (observe também diagrama na figura):

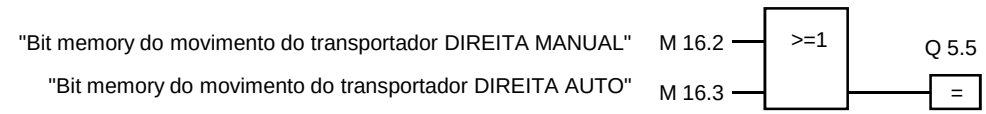
- Quando o modo AUTO (Q 4.3 = ´1´) é acionado, o motor do transportador parte para a DIREITA tão logo uma peça é colocada no Local 1, 2 ou 3 e a chave para partida associada é pressionada.
- O motor do transportador pára tão logo a peça atinge a posição final (Controle Final), ou seja, tão logo a peça **tiver atravessado** a barreira luminosa (-> necessária uma detecção de flanco, veja a figura) ou então se o modo AUTO for desligado.

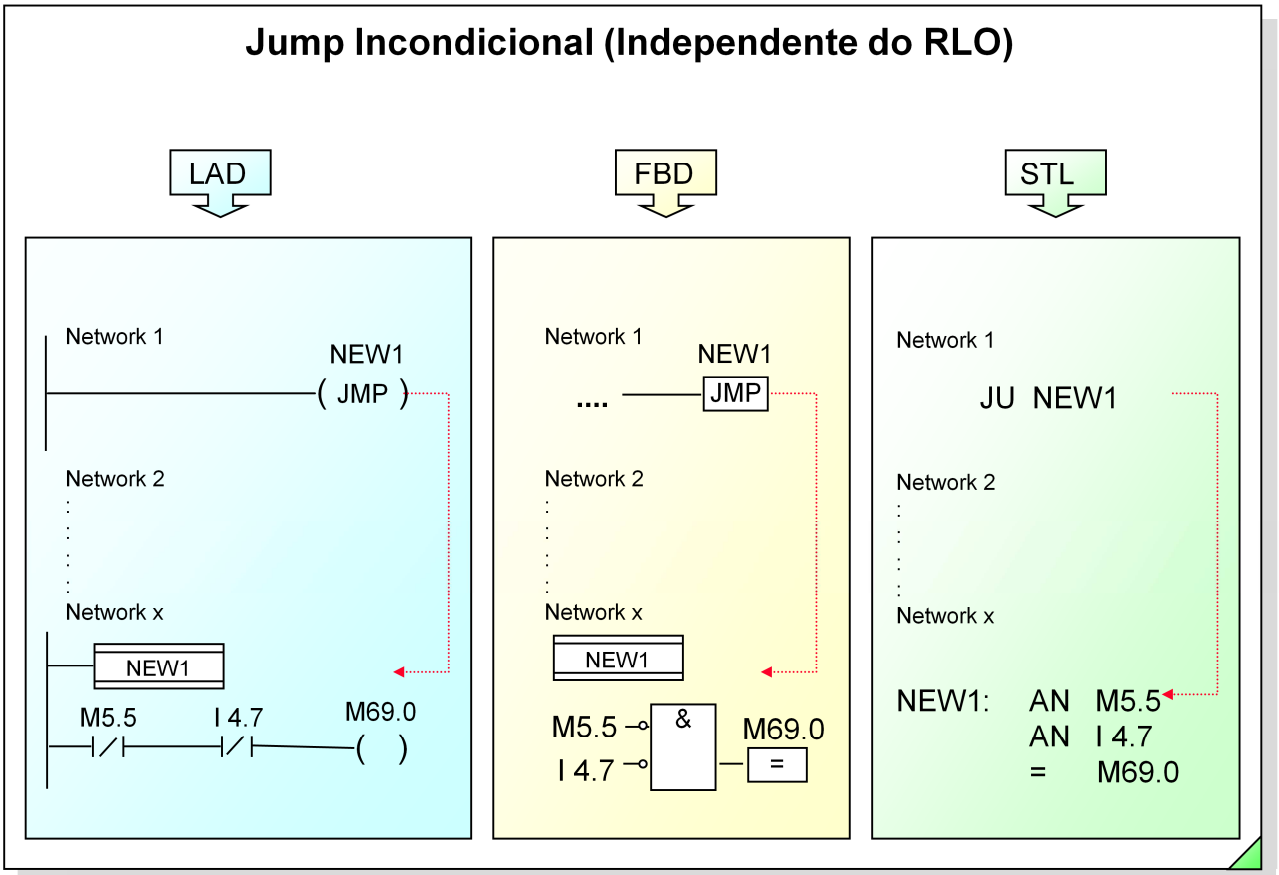
O Que Fazer:

1. Programe o movimento do transportador em AUTO no FC 16, onde o jog do motor do transportador no modo MANUAL está programado;
2. Transfira o bloco modificado FC 16 para a CPU;
3. Verifique se o programa realiza a função desejada!

Nota de Solução

O motor do transportador para a DIREITA (Q 5.5) deve ser acionado em duas condições: No modo MANUAL acionando o comando para a DIREITA OU no modo AUTO. Programe um bit memory para cada uma das duas condições e / ou armazene os resultados das operações lógicas em bit memories de modo a utilizá-los em um novo network para controlar o motor do transportador:





SIMATIC S7

Data: 26/08/2011
Arquivo: S7-Bas-06.15

sitrain

Instrução de Salto

Em LAD/FBD, o label é introduzido como um identificador acima do símbolo da saída, ou de atribuição. Em STL ele aparece depois da instrução de salto.

O label pode ter até quatro caracteres, sendo que o primeiro deve ser uma letra ou o caractere "_".

O label marca o ponto onde a execução do programa deve continuar. Nenhuma instrução ou segmento entre a instrução de salto e o label é executada.

Os saltos tanto podem ser feitos para a frente como para trás.

A instrução de salto e o destino do salto têm de estar programados no mesmo bloco (comprimento máx. do salto = 64Kbyte). O destino do salto só pode existir uma vez no bloco.

As instruções de salto podem ser utilizadas nos FBs, FCs e OBs.

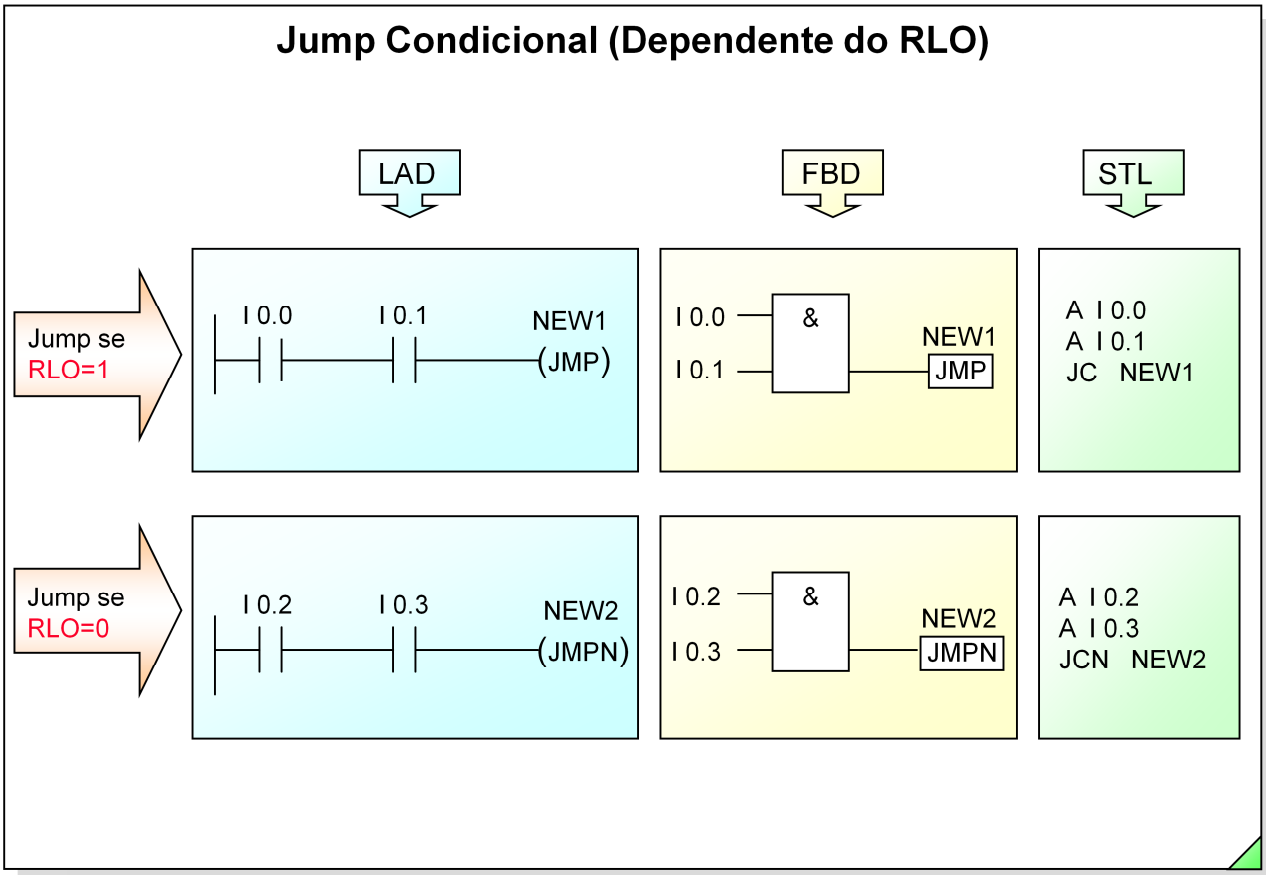
Inserindo um Label

Em LAD e FBD utiliza-se a seguinte seqüência de menus para inserir um label: *Program Elements -> Logic Control / Jump -> Label.*

Em STL, o label é introduzido do lado esquerdo da instrução na qual o programa deve continuar.

JU

Uma instrução de salto incondicional faz com que o programa salte para o label mencionado **independentemente do RLO**.



SIMATIC S7

Data: 26/08/2011
Arquivo: S7-Bas-06.16

sitrain

JC O salto condicional “JC” só é executado se o RLO for “1”.
Se o RLO for “0”, o salto não é executado, o RLO passa a ter o valor ”1” e a execução do programa continua na instrução seguinte.

JCN O salto condicional “JCN” só é executado se o RLO for "0".
Se o RLO for "1", o salto não é executado e a execução do programa continua na instrução seguinte.

Nota Em STL existem outras instruções de salto, que não são discutidas neste curso.