

Lógica Combinatória e Linguagem FBD

8.0 Generalidades, 62

8.1 Operação Lógica OU-OR, 62

8.2 Operação Lógica E-AND, 63

8.3 Operação Lógica NÃO-NOT, 63

8.4 Operação Lógica NE-NAND, 63

8.5 Operação Lógica NOU-NOR, 64

8.6 Operação Lógica OU Exclusivo-OR Exclusivo, 64

8.7 Simples Conversão da Linguagem Ladder (KOP) em Linguagem FBD, 64

8.8 Operações Lógicas Combinatórias com a CPU S7-200, 67

8.9 Aplicação: Mascaramento das Entradas, 68

8.10 Aplicação: Comando de Três Saídas em Sequência Energizadas pelo Mesmo Botão, 70

8.11 Aplicação: Linha de Produção Discretizada, 72

8.0 Generalidades

Na programação dos controladores programáveis analisada nos capítulos anteriores, não foi necessária uma competência específica de circuitos digitais.

Todavia, para quem possui conhecimento e competência de eletrônica digital, é possível programar os PLCs com o uso de portas lógicas clássicas AND, OR e NOT, estudadas em todo curso de engenharia.

Em linhas gerais, qualquer programa escrito na linguagem Ladder pode se converter em uma linguagem a portas lógicas.

Lembramos que a linguagem a portas lógicas chama-se, segundo a norma IEC 61131-3, linguagem FBD (*Function Block Diagram*); na linguagem Siemens, chama-se FUP.

É importante dizer que nem todos os PLCs dispõem desse tipo de linguagem de programação.

Neste capítulo, abordaremos somente as bases das conversões entre a linguagem Ladder e a linguagem FBD, evidenciando a total dualidade dessas duas linguagens. Ou, melhor dizendo, são dois modos diferentes de se escrever em forma gráfica um mesmo programa.

Evitaremos entrar em detalhe da lógica booleana, até porque não é indispensável para a programação dos PLCs.

8.1 Operação Lógica OU-OR

A operação lógica OR é uma operação de soma lógica binária entre dois ou mais operandos, chamada às vezes de variável, e está reportada na tabela-verdade da Figura 8.1.

X	Y	S = X + Y
0	0	0
0	1	1
1	0	1
1	1	1

Figura 8.1

A tabela-verdade da Figura 8.1 mostra as possíveis combinações da operação de soma lógica binária e os resultados S (saída). Nessa tabela, X e Y possuem os estados "0" e "1" lógicos, nos quais a "0" lógico é associada uma chave aberta, e a "1" lógico, uma chave fechada. O conceito da operação lógica OR é:

É necessário que uma das variáveis X ou Y seja 1 para que o resultado S seja 1.

Essa situação na linguagem Ladder equivale a uma ligação de chaves em paralelo. Na Figura 8.2, são representados o esquema lógico, Ladder e FBD equivalente da função OR.

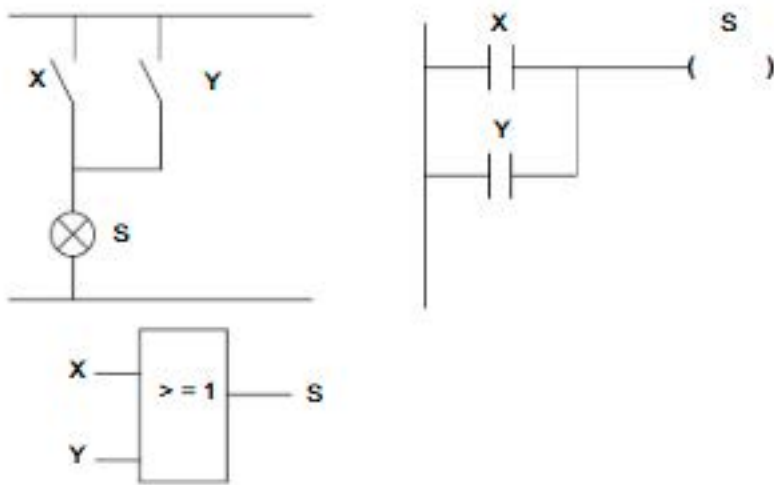


Figura 8.2

8.2 Operação Lógica E-AND

A operação lógica AND é uma operação de produto lógico binário entre dois ou mais operandos chamada variável. Está reportada na tabela-verdade da Figura 8.3.

X	Y	S = XY
0	0	0
0	1	0
1	0	0
1	1	1

Figura 8.3

O conceito da operação lógica AND diz que:

É suficiente que uma das variáveis X e Y seja 0 para que o resultado de S seja 0.

Essa situação, na linguagem Ladder, equivale a uma ligação de chaves em série. Na Figura 8.4 são representados o esquema lógico, Ladder e FBD equivalente da função AND.

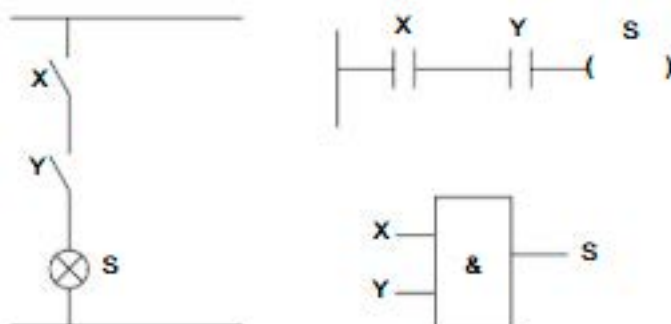


Figura 8.4

8.3 Operação Lógica NÃO-NOT

A operação lógica NOT é uma operação de complementação chamada também de inversão sobre uma só variável. Na Figura 8.5 são representados a tabela-verdade, o esquema lógico, Ladder e FBD equivalente à função NOT.

X	S
0	1
1	0

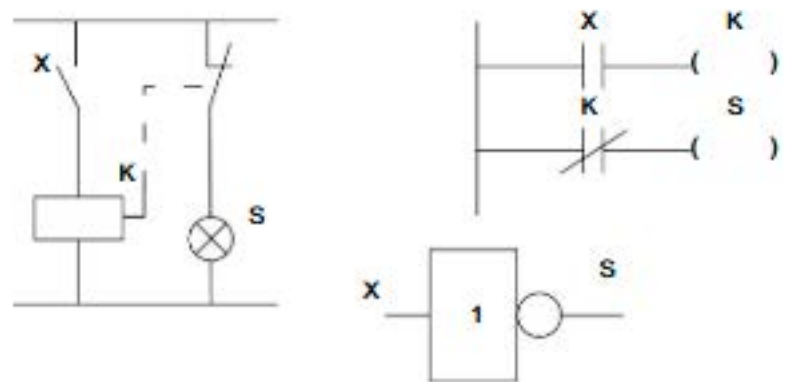


Figura 8.5

8.4 Operação Lógica NE-NAND

A operação lógica NAND é uma combinação em série de uma porta lógica AND com uma porta lógica NOT. Na Figura 8.6, são representados a tabela-verdade, o esquema lógico, Ladder e FBD equivalente da função NAND.

X	Y	S = XY
0	0	1
0	1	1
1	0	1
1	1	0

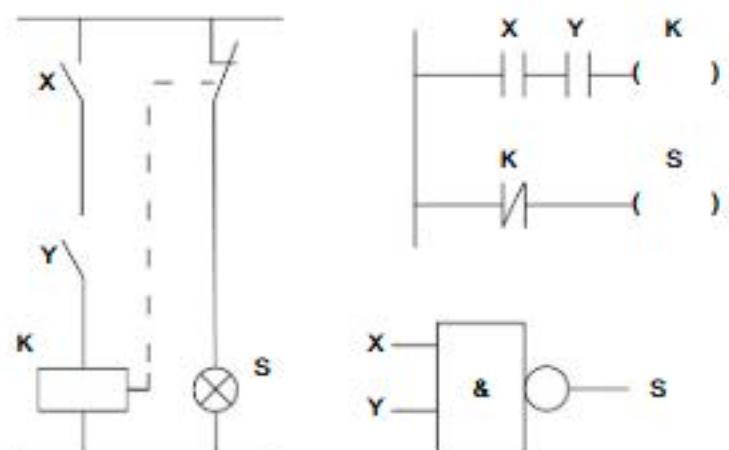


Figura 8.6

8.5 Operação Lógica NOU-NOR

A operação lógica NOR é uma combinação em série de uma porta lógica OR com uma porta lógica NOT. Na Figura 8.7, são representados a tabela-verdade, o esquema lógico, Ladder e FBD equivalente da função NOR.

X	Y	S = X+Y
0	0	1
0	1	0
1	0	0
1	1	0

$$S = \overline{xy} + \overline{xy}$$

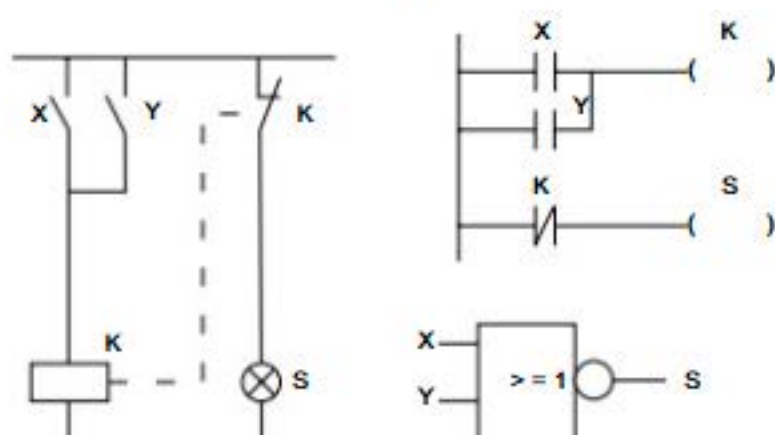


Figura 8.7

8.6 Operação Lógica OU Exclusivo-OR Exclusivo

A operação lógica OR Exclusivo é uma operação de combinação lógica segundo a seguinte tabela-verdade. Veja Figura 8.8.

X	Y	S
0	0	0
0	1	1
1	0	1
1	1	0

$$S = x\overline{y} + \overline{x}y$$

Figura 8.8

O conceito da operação lógica OR Exclusivo é:

A saída S é a "1" lógico quando somente uma das entradas X e Y estão ao nível lógico "1". Na Figura 8.9 são representados o esquema lógico, Ladder e FBD equivalente da função OR Exclusivo.

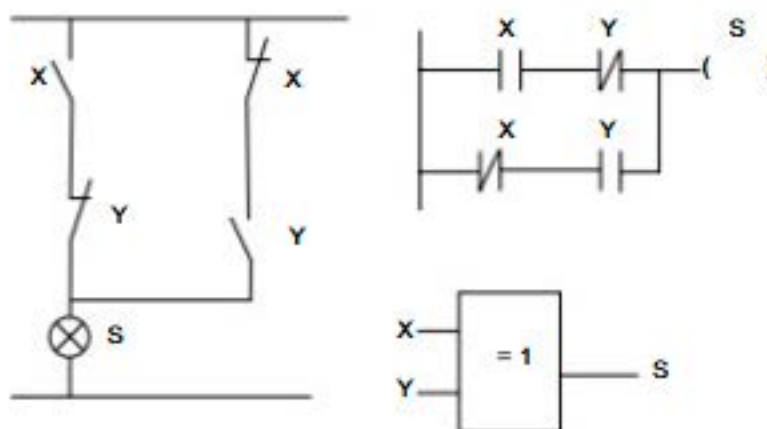


Figura 8.9

8.7 Simples Conversão da Linguagem Ladder (KOP) em Linguagem FBD

Nos exemplos a seguir ilustraremos uma simples conversão da linguagem Ladder (KOP) em equivalente FBD.

Lembramos que na linguagem Ladder (KOP) as chaves em série são equivalentes a uma operação lógica AND, enquanto as chaves em paralelo são equivalentes a uma operação lógica OR.

Uma chave normalmente fechada NF é equivalente a uma operação lógica NOT na linguagem FBD. Por convenção, é desenhado um pequeno círculo na entrada de uma operação lógica qualquer.

Exercício 1

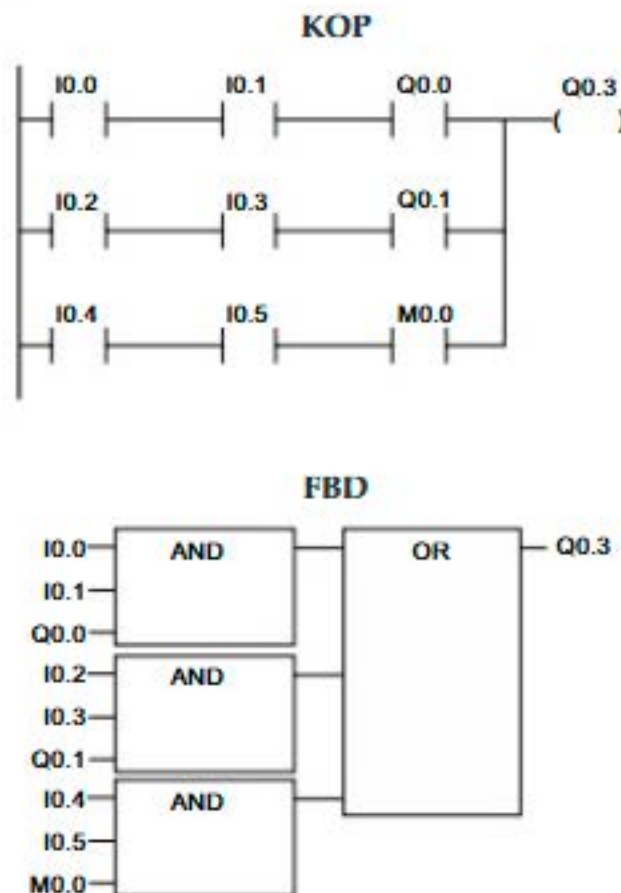


Figura 8.10

Exercício 2

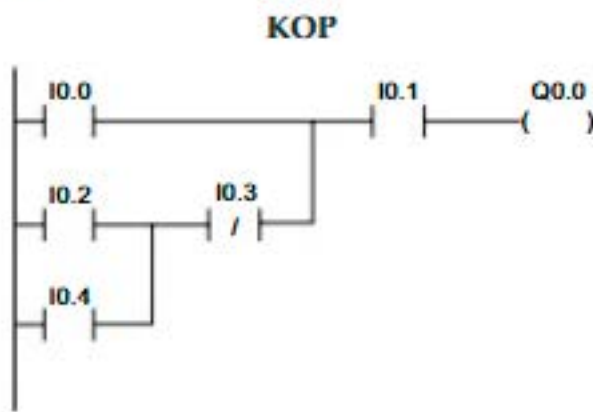


Figura 8.11A

Exercício 3

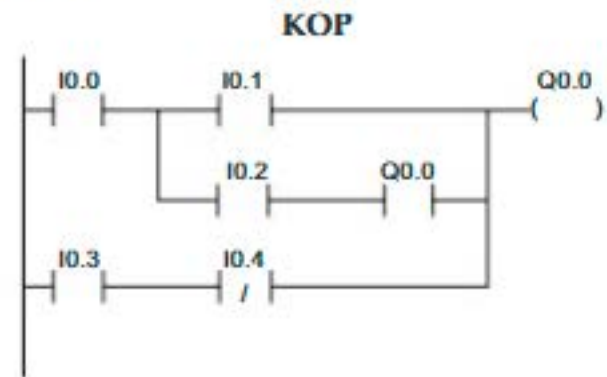


Figura 8.12A

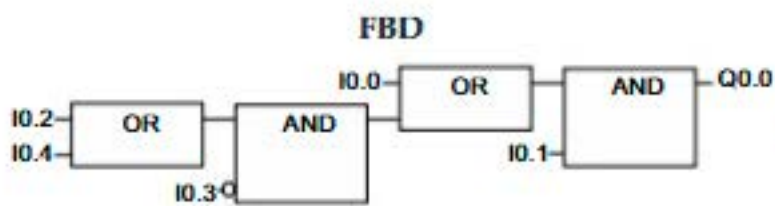


Figura 8.11B

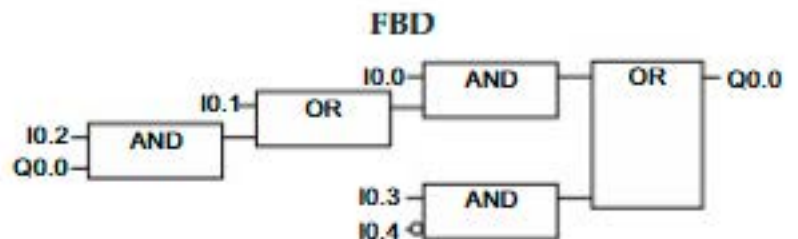


Figura 8.12B

Exercício 4: Chave reversora eletromecânica para MAT

Esquema elétrico funcional

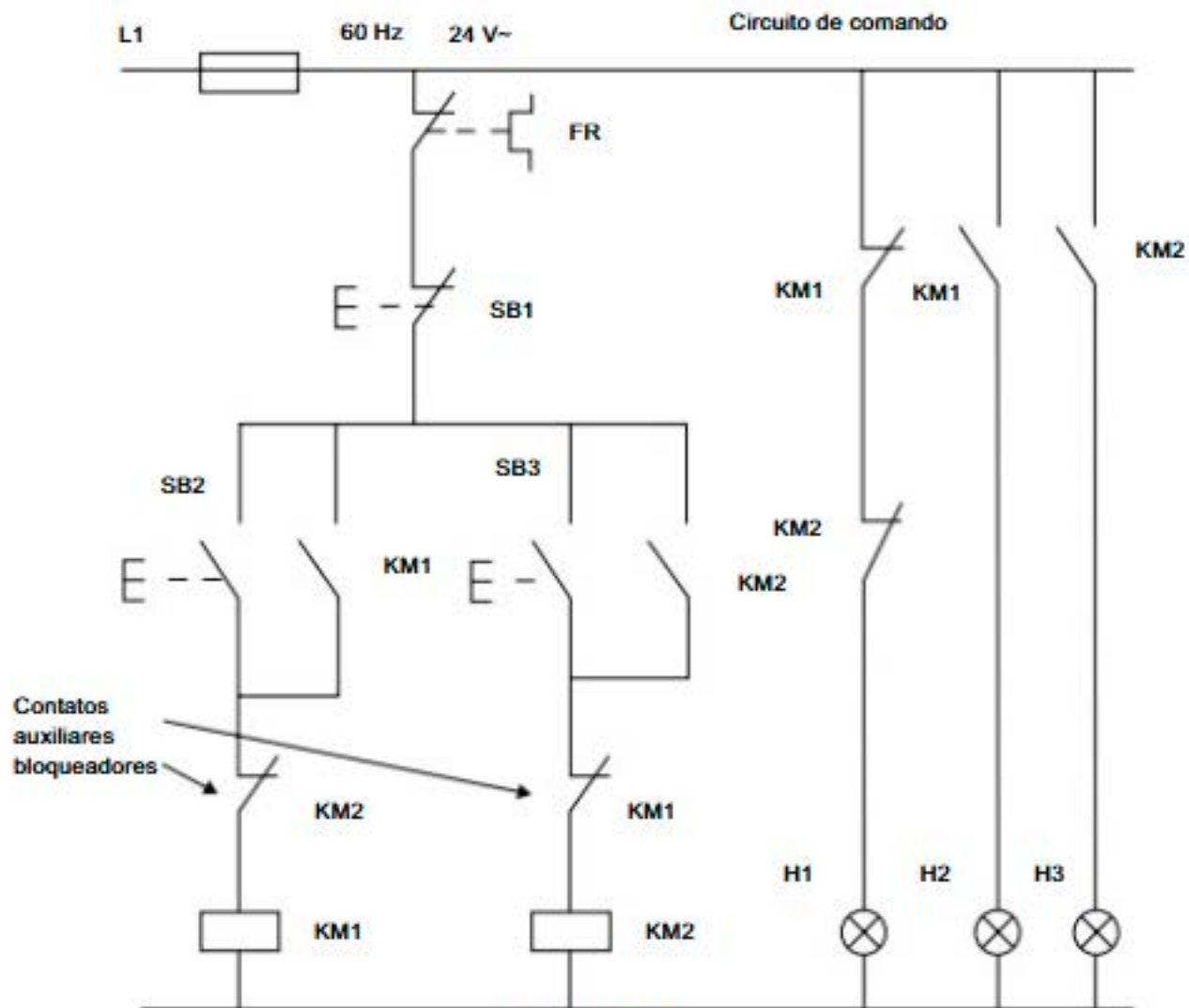


Figura 8.13A

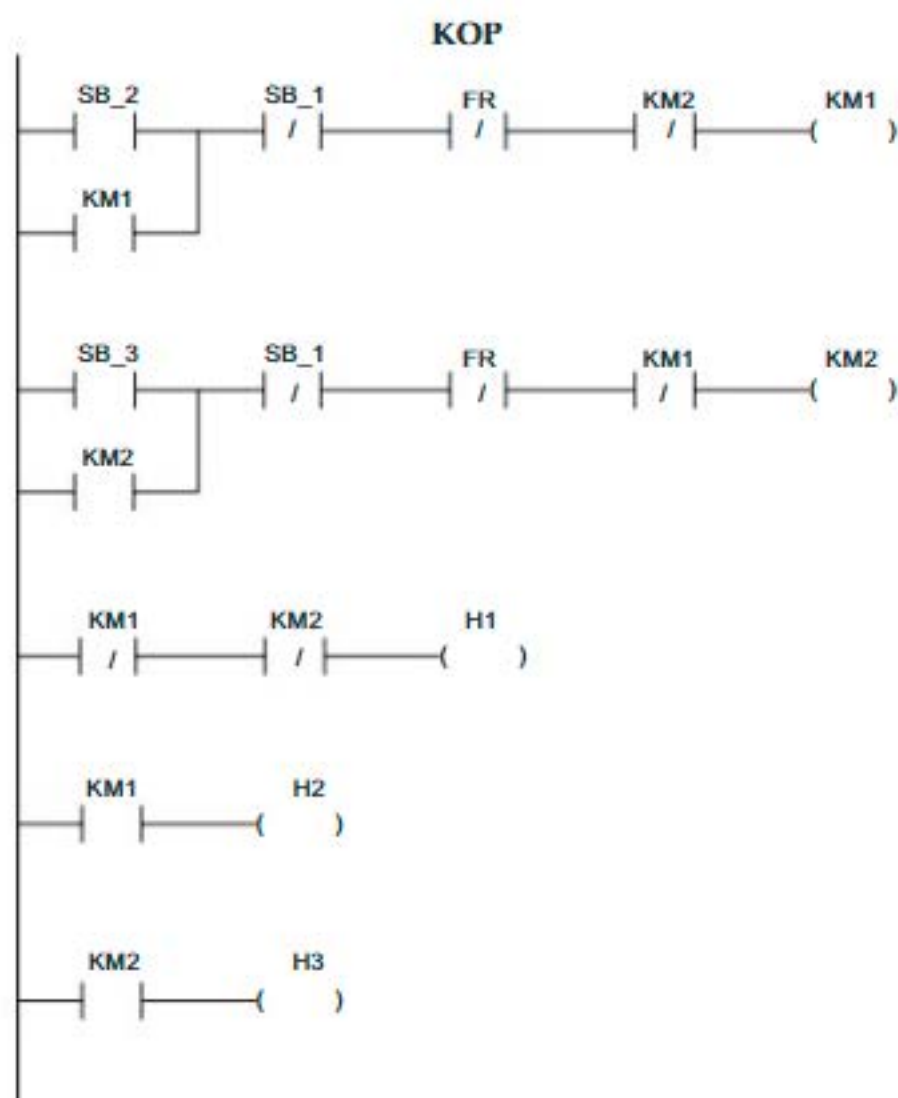


Figura 8.13B

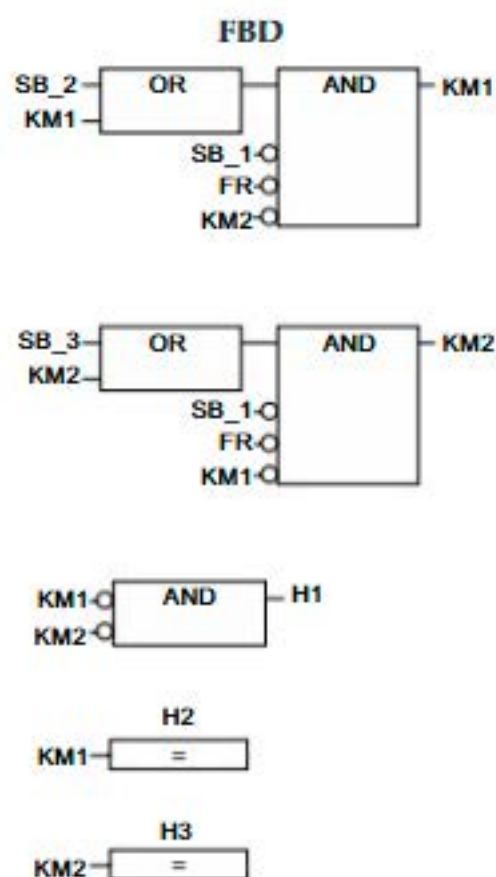
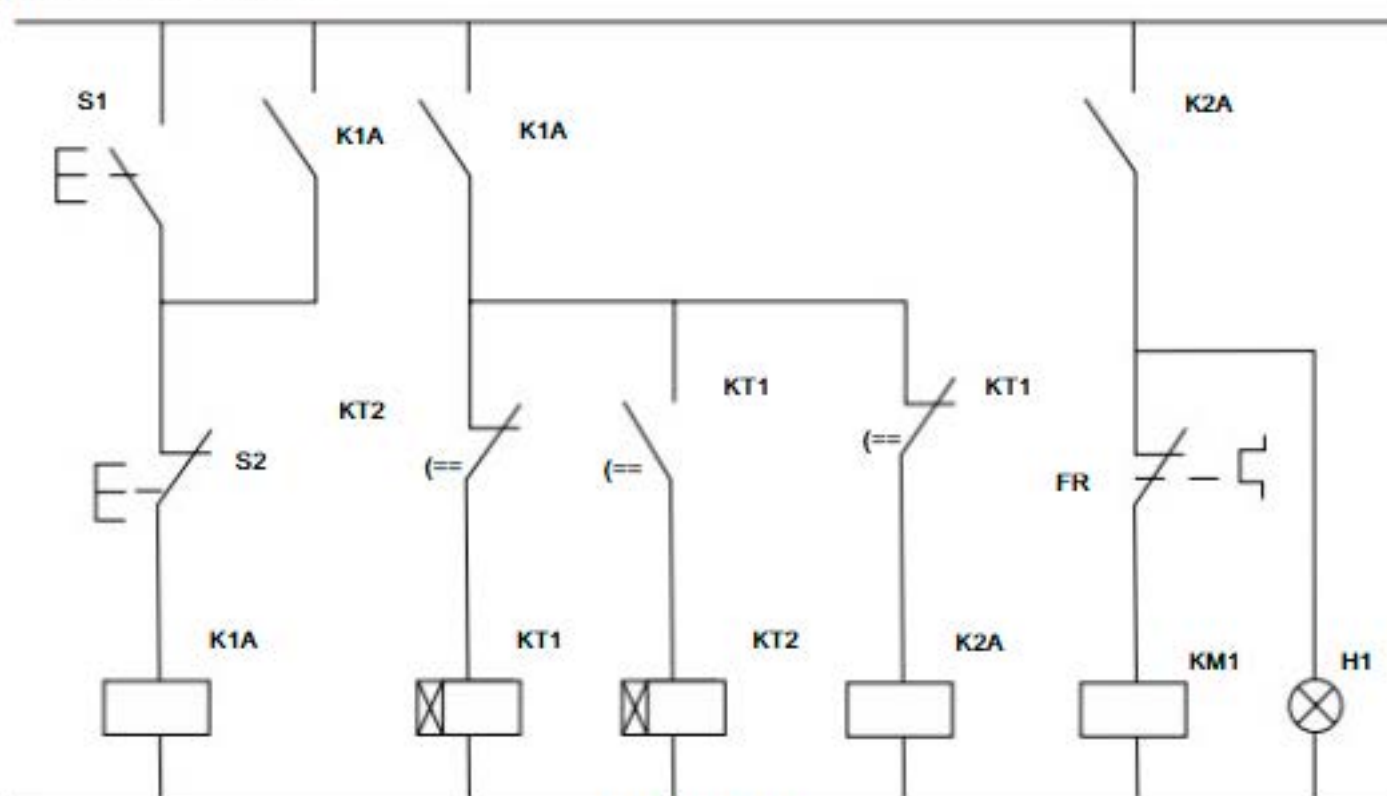


Figura 8.13C

Exercício 5: Pausa e trabalho de um MAT

Esquema elétrico funcional

**Figura 8.14A**

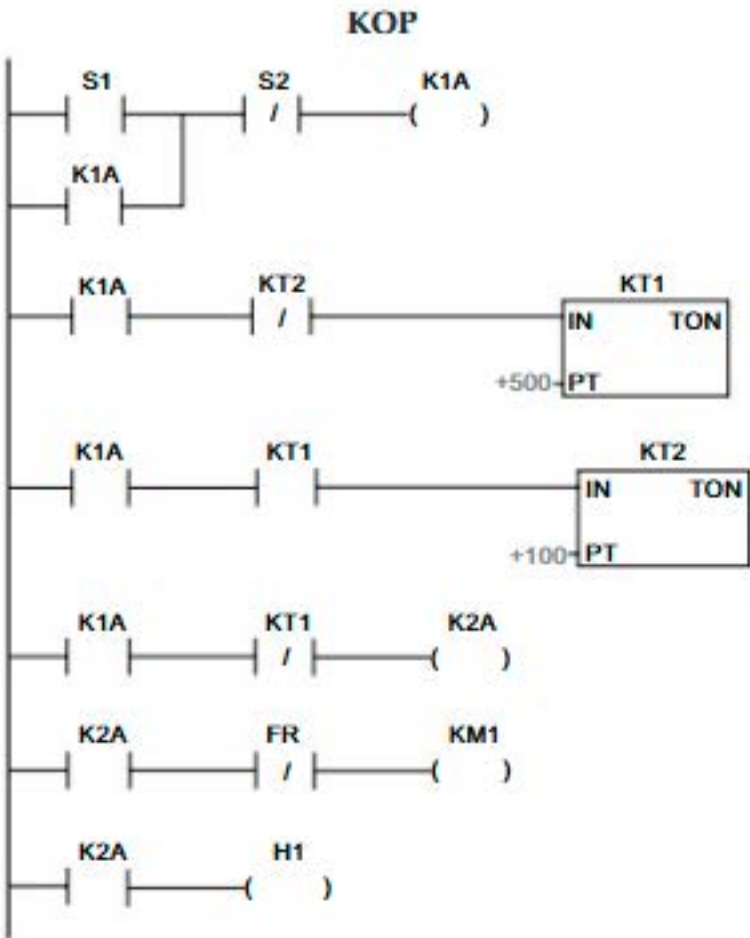


Figura 8.14B

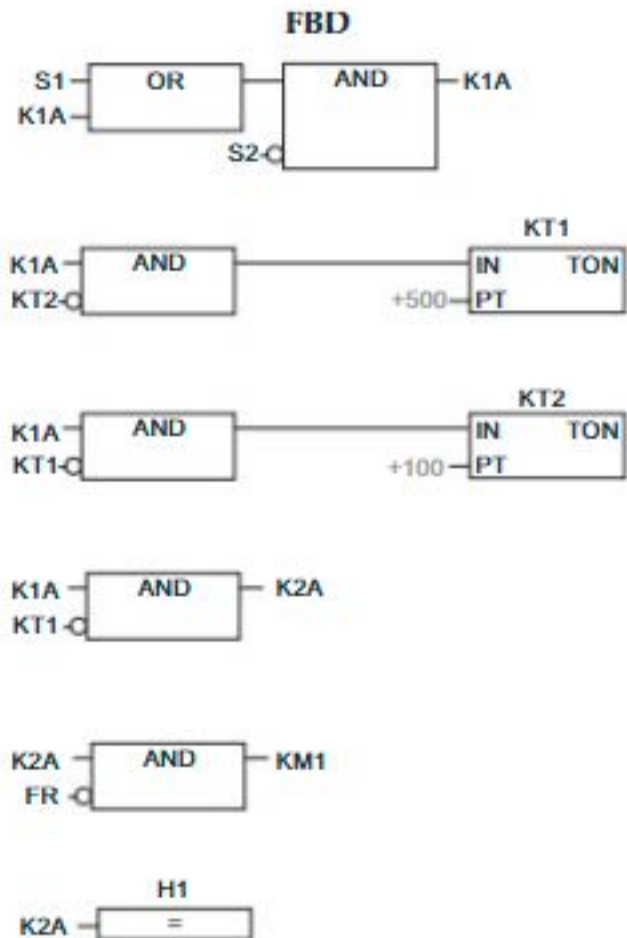


Figura 8.14C

8.8 Operações Lógicas Combinatórias com a CPU S7-200

As operações lógicas combinatórias nos PLCs são executadas entre bits individuais, Word ou double Word. É possível a combinação com operações lógicas tipo AND, OR, OR-Exclusivo. Na Tabela 8.1 são resumidas as principais operações lógicas combinatórias com a CPU S7-200.

Tabela 8.1

KOP	Função
WAND_W EN IN1 IN2 OUT	O boxe "combina Words em AND" (WAND_W) combina os bits correspondentes das Words de entrada IN1 e IN2 em AND e carrega o resultado (OUT) em uma Word.
WOR_W EN IN1 IN2 OUT	O boxe "combina Words em OR" (WOR_W) combina os bits correspondentes das Words de entrada IN1 e IN2 em OR, e carrega o resultado (OUT) em uma Word.
WXOR_W EN IN1 IN2 OUT	O boxe "combina Words em OR exclusivo" (WXOR_W) combina os bits correspondentes das Words de entrada IN1 e IN2 em OR exclusivo e carrega o resultado (OUT) em uma Word.

(continua)

Tabela 8.1 (Continuação)

KOP	Função
WAND_DW EN IN1 IN2 OUT	O boxe "combina double Word em AND" (WAND_DW) combina os bits correspondentes das double Words de entrada IN1 e IN2 em AND e carrega o resultado (OUT) em uma double Word.
WOR_DW EN IN1 IN2 OUT	O boxe "combina double Word em OR" (WOR_DW) combina os bits correspondentes das double Words de entrada IN1 e IN2 em OR e carrega o resultado (OUT) em uma double Word.
WXOR_DW EN IN1 IN2 OUT	O boxe "combina double Word em OR exclusivo" (WXOR_DW) combina os bits correspondentes das double Words de entrada IN1 e IN2 em OR exclusivo e carrega o resultado (OUT) em uma double Word.

O exemplo a seguir ilustra as operações lógicas combinatórias em linguagem Ladder com a utilização das Words. Veja Figura 8.15A.

Nas Figuras 8.15B, 8.15C e 8.15D são representados os estados dos registros seguidos de cada uma das operações lógicas combinatórias.

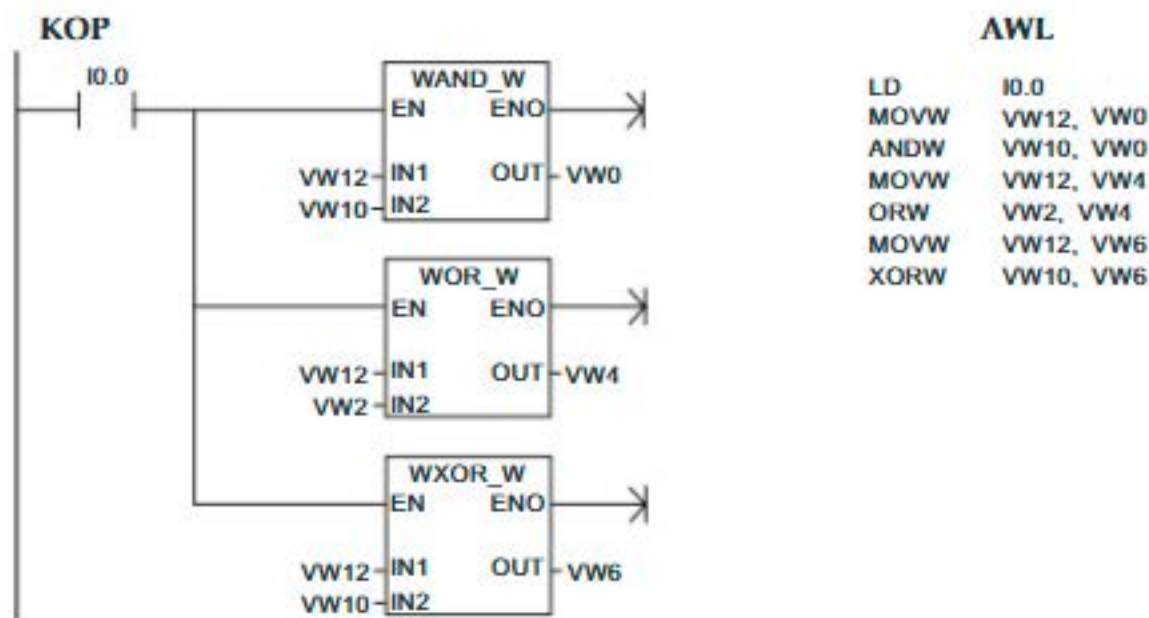


Figura 8.15A

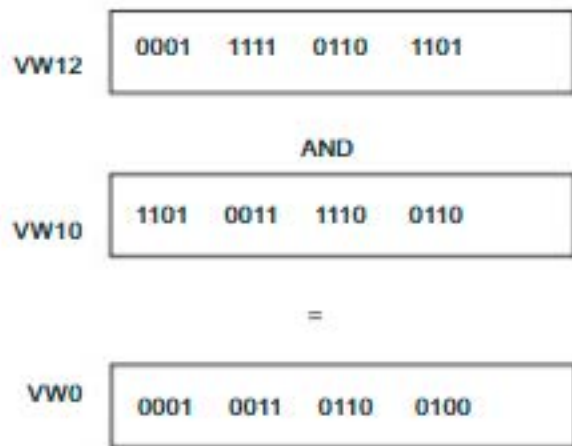


Figura 8.15B

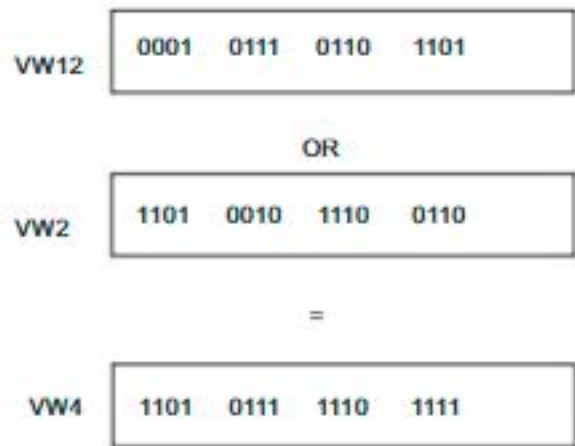


Figura 8.15C

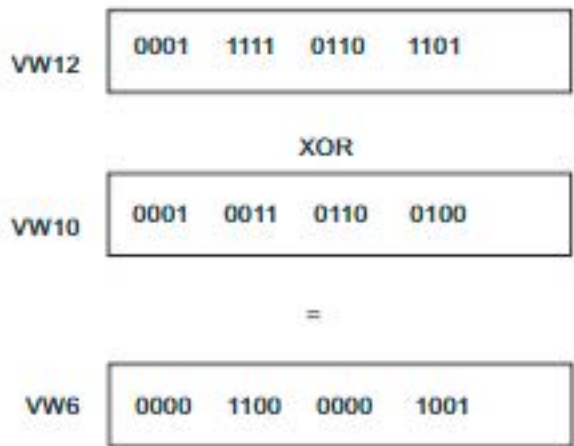


Figura 8.15D

8.9 Aplicação: Mascaramento das Entradas

Vamos mostrar agora um exemplo aplicativo simples da operação lógica combinatória à Word tipo WAND_W. O operador de um forno elétrico pode definir o tempo de aquecimento com o uso de chaves digitais. Na Figura 8.16, nota-se a presença de três chaves digitais codificadas em código BCD. Esse valor do tempo codificado em BCD é armazenado no registro das imagens

das entradas IW0, que possui 16 bits; os primeiros 4 bits à esquerda não são utilizados porque as chaves são 3. Os bits inutilizados, que vão de I0.4 até I0.7, deverão ser *mascarados*, ou seja, não devem ser considerados na lógica de controle do programa. Inicialmente se prevê a detecção de todas as entradas do PLC por meio do registro IW0. O *mascaramento* é a operação de AND lógico com os registros IW0 e máscara. O registro máscara é um registro no qual os

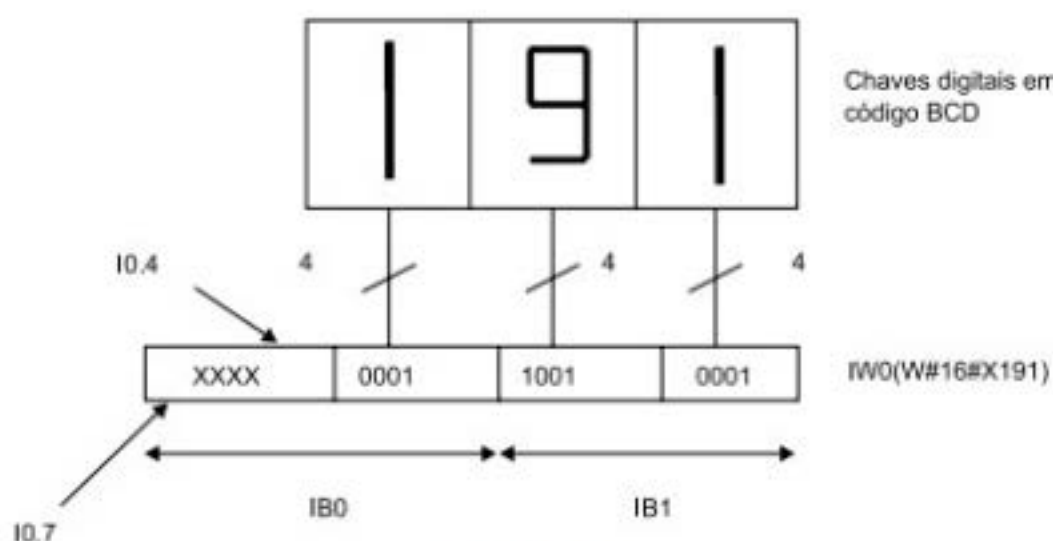


Figura 8.16

bits são todos forçados a ON ou "1" lógico em correspondência às entradas que se quer testar e forçados a OFF ou "0" lógico os bits que devem ser mascarados (no nosso caso de I0.4 até I0.7). A codificação desse registro em código hexadecimal é: W#16#0FFF.

Na Figura 8.17 temos o registro máscara armazenado com o valor 0FFF.

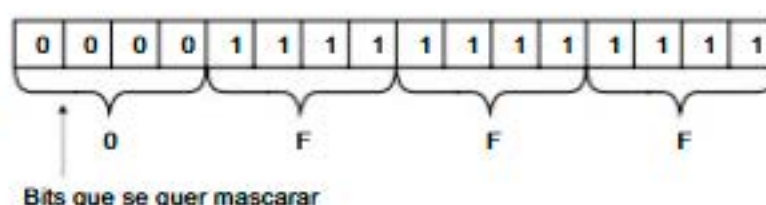


Figura 8.17

Esquema Ladder e AWL do mascaramento das entradas (Figura 8.18)

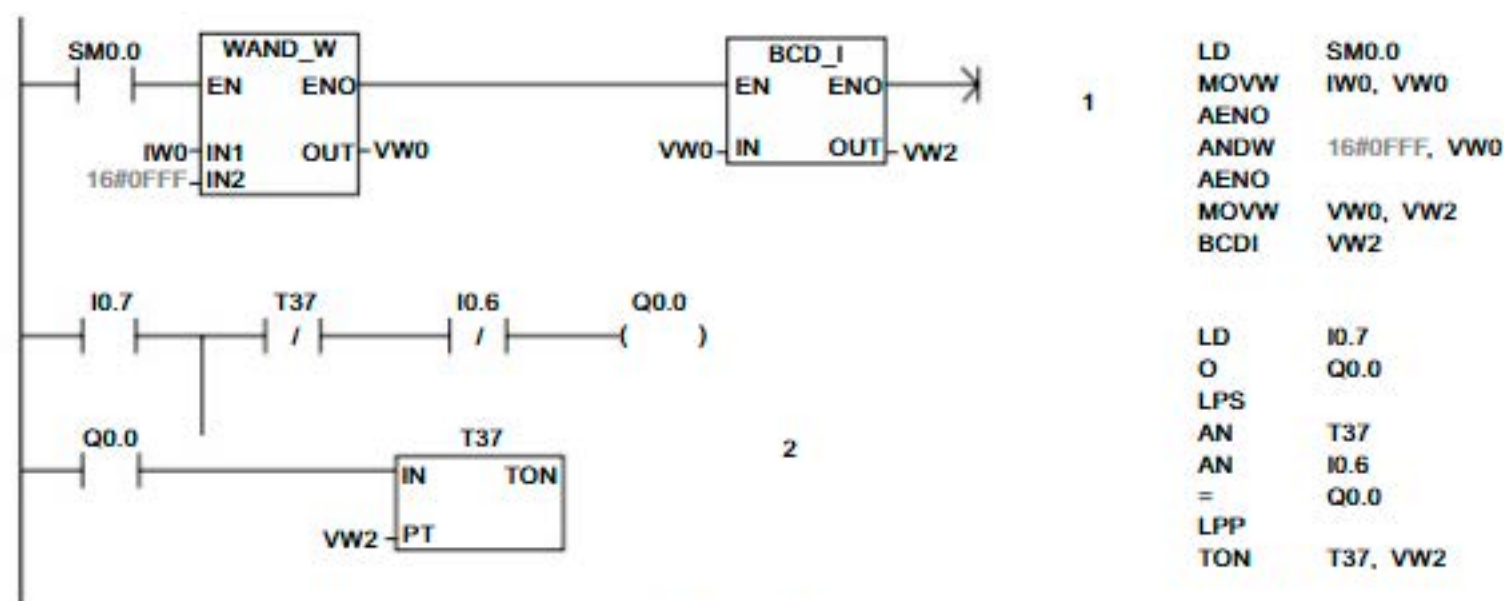


Figura 8.18

O esquema Ladder da Figura 8.18 é muito simples.

1. Na primeira linha de programa temos uma operação lógica combinatória a Word tipo AND entre o registro das imagens das entradas IWO e o registro com o valor máscara 0FFF. O resultado é carregado na variável Word VW0. Veja Figura 8.19.

O valor carregado na variável Word VW0 é convertido do código BCD em um valor inteiro e carregado na variável Word VW2.

2. Na segunda linha de programa temos o botão I0.7, que, pressionado, energiza a saída Q0.0 e em consequência se liga o forno; começa assim a contagem do timer T37, com o valor de preset definido pela variável Word VW2 (no nosso caso 19,1 segundos).

Terminada a contagem do tempo se desenergiza a saída Q0.0, o forno se desliga.

Pressionando o botão I0.6 se desliga o forno a qualquer momento.

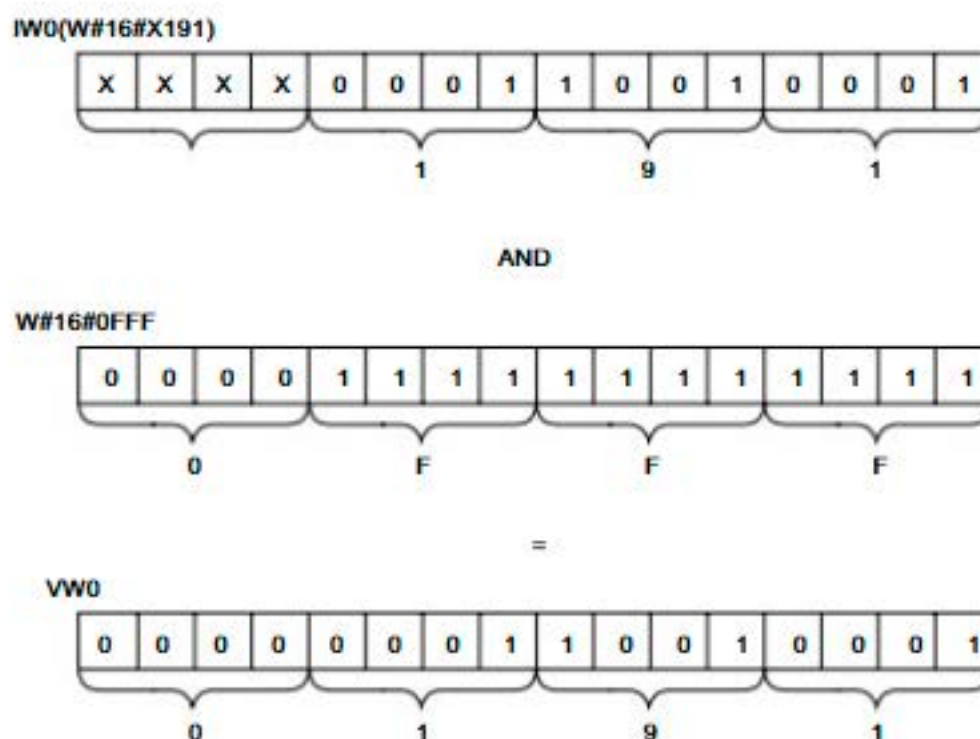


Figura 8.19

8.10 Aplicação: Comando de Três Saídas em Sequência Energizadas pelo Mesmo Botão

Essa aplicação consiste no comando em sequência de três saídas K1, K2, K3 energizadas pelo mesmo botão de partida S1.

O funcionamento é muito simples:

Pressionando o botão S1 se energiza a saída K1; pressionando novamente S1 se energiza a saída K2; pressionando novamente S1, se energiza a saída K3. O botão de parada S2 para tudo em qualquer momento. Veja Figura 8.20.

Tabela 8.2 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1	I0.1	Botão start
S2	I0.2	Botão stop
K1	Q0.0	Saída 1
K2	Q0.1	Saída 2
K3	Q0.2	Saída 3
K1A	M0.0	Merker
REG1	VW200	Word
REG2	VW300	Word

O esquema Ladder e AWL da Figura 8.21 resolve a aplicação.

É utilizado um shift register em combinação com uma operação de lógica combinatória do tipo WOR_W.

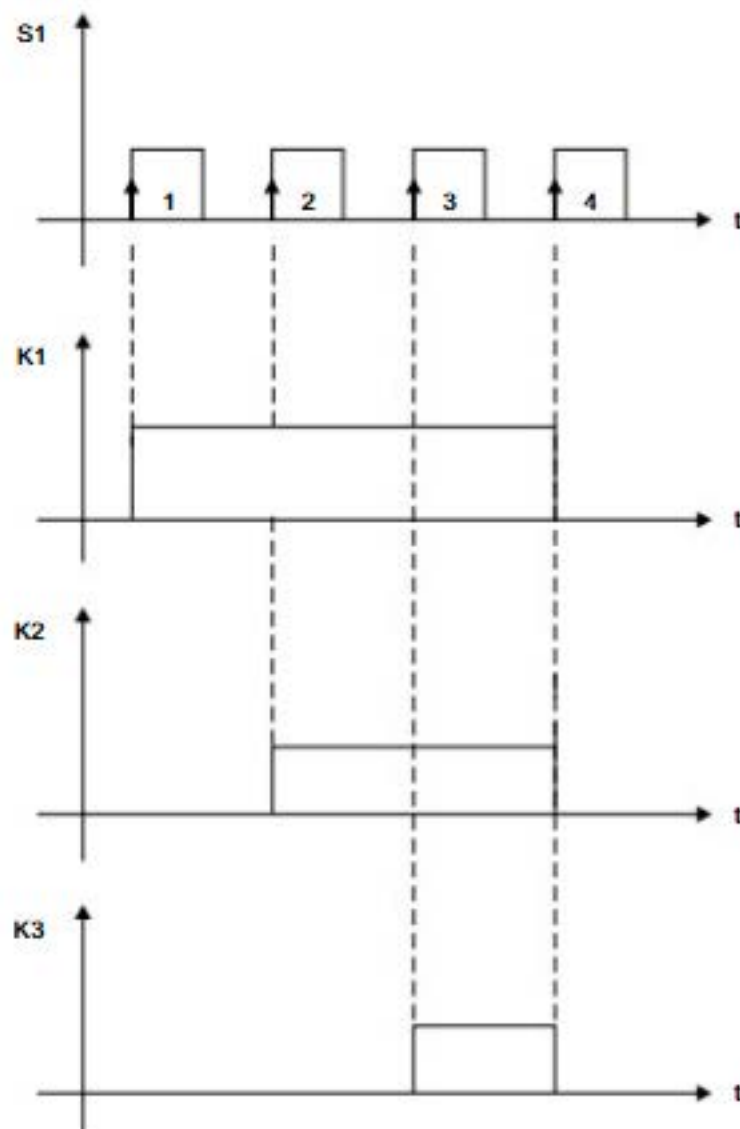
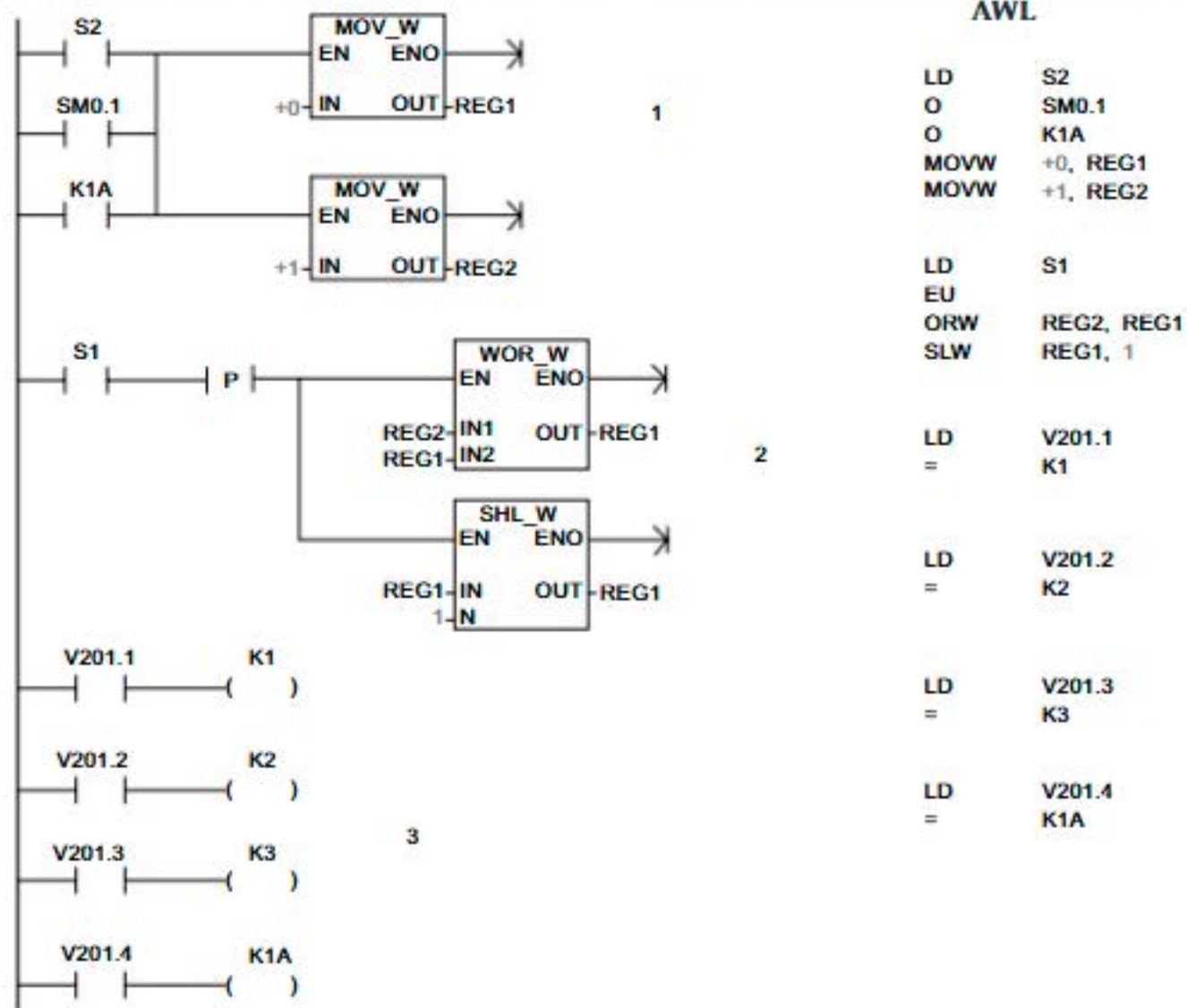


Figura 8.20

Esquema Ladder e AWL do Comando de Três Saídas em Sequência Energizadas do Mesmo Botão (Figura 8.21)

**Figura 8.21**

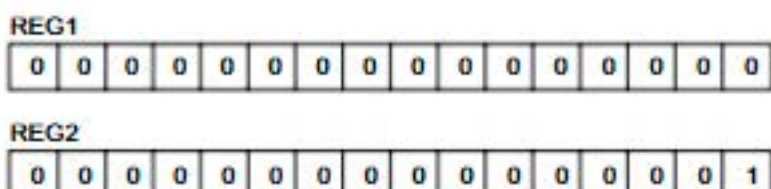
1. Na primeira linha de programa temos o bit especial SM0.1 ativo no primeiro ciclo de scan. Carrega-se o valor zero no registro REG1 e o valor 1 no registro REG2.
2. Na segunda linha de programa temos o botão de partida S1 com as instruções WOR W e SHL W.

Essa linha de programa precisa de algumas explicações.

O registro REG1 servirá como Word de amostra e será a Word em que os bits do shift register serão deslocados.

O registro REC2 servirá como Word de comparação.

Depois do reset inicial da primeira linha de programa, as Words dos registros REG1 e REG2 terão as seguintes configurações (Figura 8.22).

**Figura 8.22**

Com a programação da segunda linha de programa se efetua uma soma lógica (OR) entre o registro REG1 e o registro REG2 de modo a inserir o bit menos significativo LSB de REG1 no shift register (bit V201.0) (Figura 8.23).

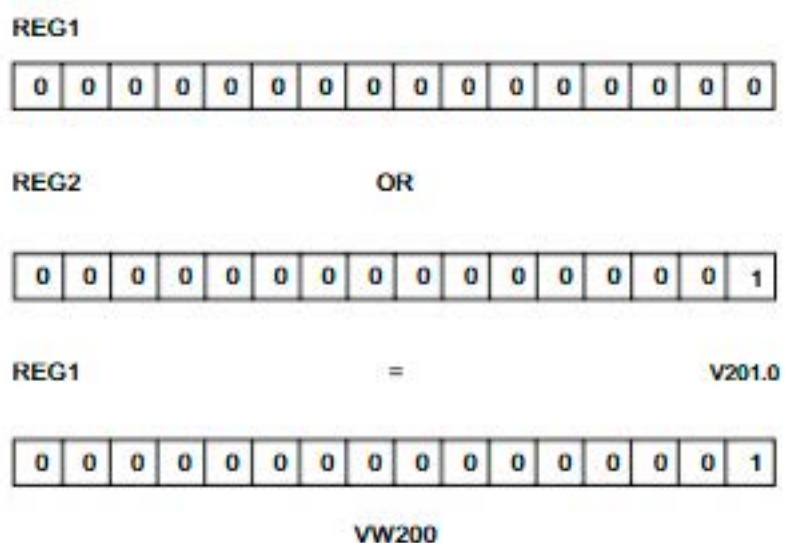


Figura 8.23

Pressionando o botão S1 (primeiro pulso), determina-se o deslocamento à esquerda de uma posição do bit que constitui a Word registro REG1. A nova configuração da Word registro REG1 será como na Figura 8.24.

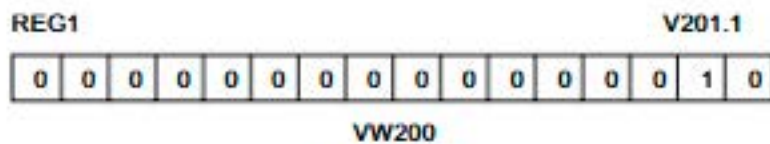


Figura 8.24

O bit V201.1 é em nível lógico "1" e energiza a saída K1.

Pressionando novamente o mesmo botão S1 (segundo pulso), a Word registro REG1 terá a seguinte configuração (Figura 8.25):

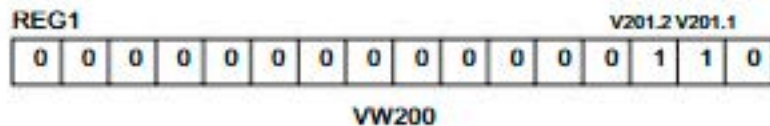


Figura 8.25

Os bits V201.1 e V201.2 serão dessa vez ao nível lógico "1" e energizarão as saídas K1 e K2, devido ao deslocamento de uma posição à esquerda do shift register.

Pressionando novamente o mesmo botão S1 (terceiro pulso), desloca-se novamente de uma posição à esquerda o bit no nível lógico "1"; dessa vez, os bits V201.3, V201.2 e V201.1 serão ao nível lógico "1" e energizarão as saídas K1, K2 e K3.

Pressionando novamente o mesmo botão S1 (quarto pulso), desloca-se novamente de uma posição à esquerda o bit no nível lógico "1"; dessa vez, o bit V201.4 é no nível lógico "1". A bobina Merker K1A será energizada.

Como se nota na primeira linha de programa, fechando-se o contato auxiliar de K1A se reseta novamente o registro REG1 e carrega o "1" no registro REG2, e assim o ciclo recomeça.

- Na terceira linha de programa temos as saídas dos bits V201.1, V201.2 e V201.3, da Word REG1 que energizarão as respectivas saídas do PLC em sequência.

8.11 Aplicação: Linha de Produção Discretizada

Nessa aplicação tomamos como referência a Seção 7.6 em que foi descrita, no nível teórico, uma linha de produção discretizada. Veremos agora a aplicação prática daquela linha de produção.

Uma estação de produção (Figura 8.26) trabalha no teste de motores elétricos.

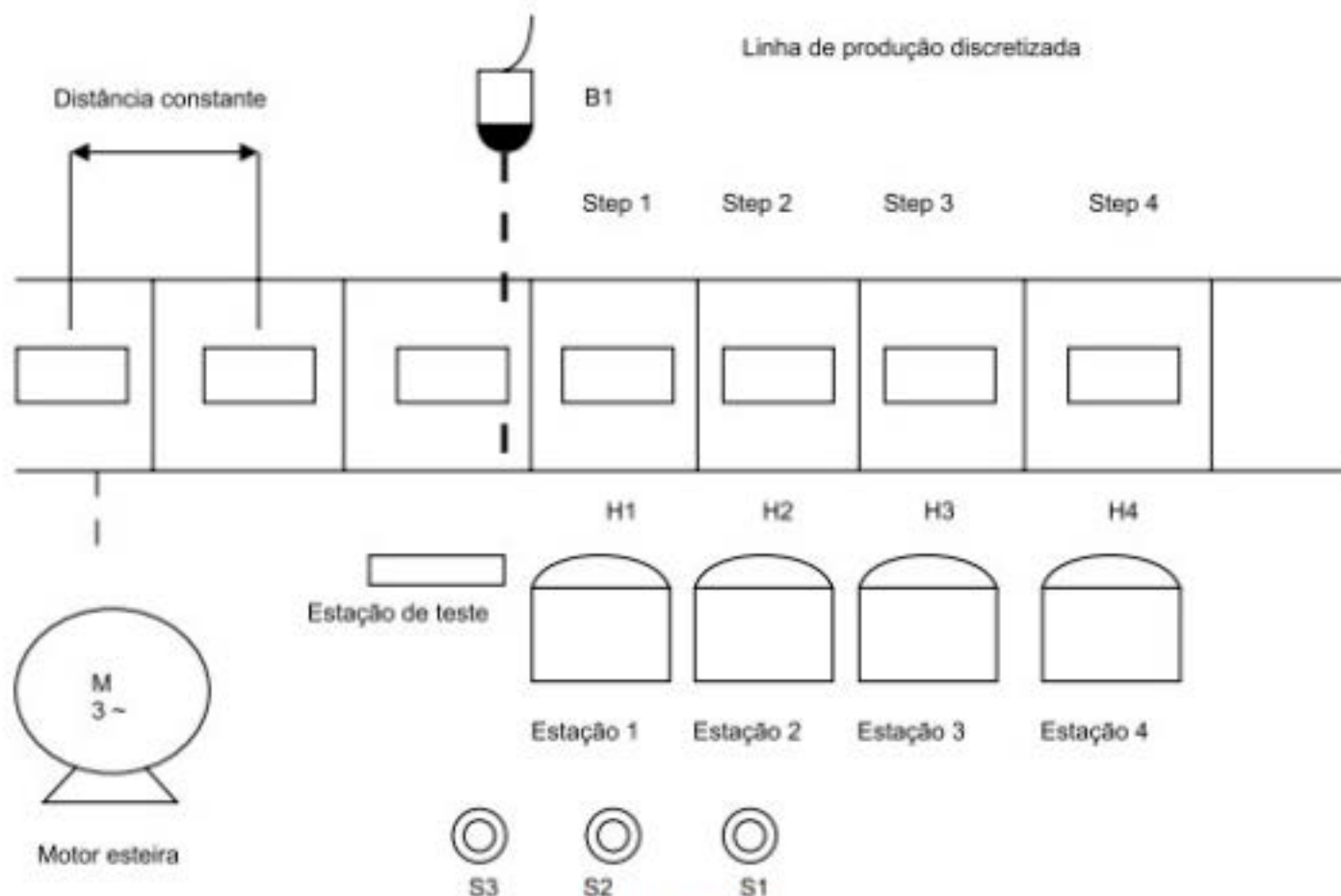


Figura 8.26

Um operador verifica o teste do motor por meio de um botão que sinaliza uma peça defeituosa, que não deve ser considerada na estação de produção sucessiva.

Nas 4 estações sucessivas, se o motor apresenta defeito; uma sinalização lampejante acusa o defeito e avisa que tal motor não deve ser trabalhado.

Uma fotocélula detecta o movimento dos motores sobre a esteira transportadora. Um botão reseta o sistema, e um botão de start/stop encaminha e para a esteira.

É importante ter uma perfeita correspondência da posição dos motores elétricos sobre a esteira a cada estação sucessiva. Ou seja, deve-se ter um motor para cada estação para que os produtos se desloquem simultaneamente.

A aplicação se resolve usando um shift register em combinação com uma operação de lógica combinatória do tipo WOR_W. Os motores elétricos sobre a esteira transportadora, uma vez testados por um operador, obterão uma informação discreta segundo a lógica.

0 = Motores testados bons

1 = Motores testados defeituosos

Tabela 8.3 Tabela dos Símbolos

Símbolos	Endereço	Comentário
S1	I0.0	Botão start-stop
S2	I0.1	Botão reset shift register
S3	I0.2	Botão de descarte peça defeituosa
B1	I0.3	Fotocélula detectar peça
KM	Q0.0	Contator esteira transportadora
H1	Q0.1	Sinalização peça defeituosa Estação 1
H2	Q0.2	Sinalização peça defeituosa Estação 2
H3	Q0.3	Sinalização peça defeituosa Estação 3
H4	Q0.4	Sinalização peça defeituosa Estação 4
REG1	VW200	Word
REG2	VW300	Word
K1A	M0.0	Merker
K2A	M0.1	Merker
K3A	M0.2	Merker
K4A	M0.3	Merker

O diagrama Ladder e AWL resolutivo é representado na Figura 8.27.

Esquema Ladder e AWL da Linha de Produção Discretizada (Figura 8.27)

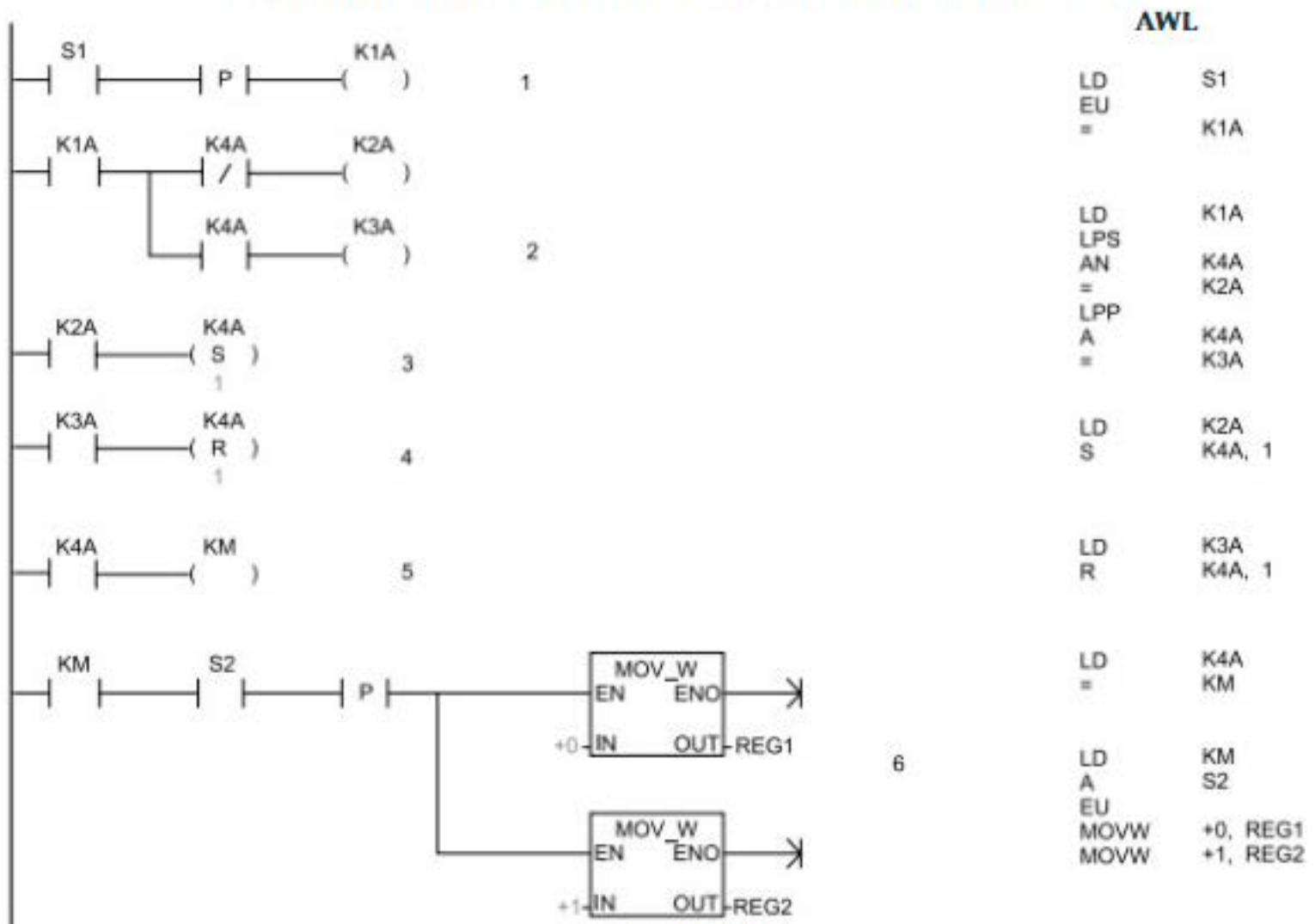


Figura 8.27

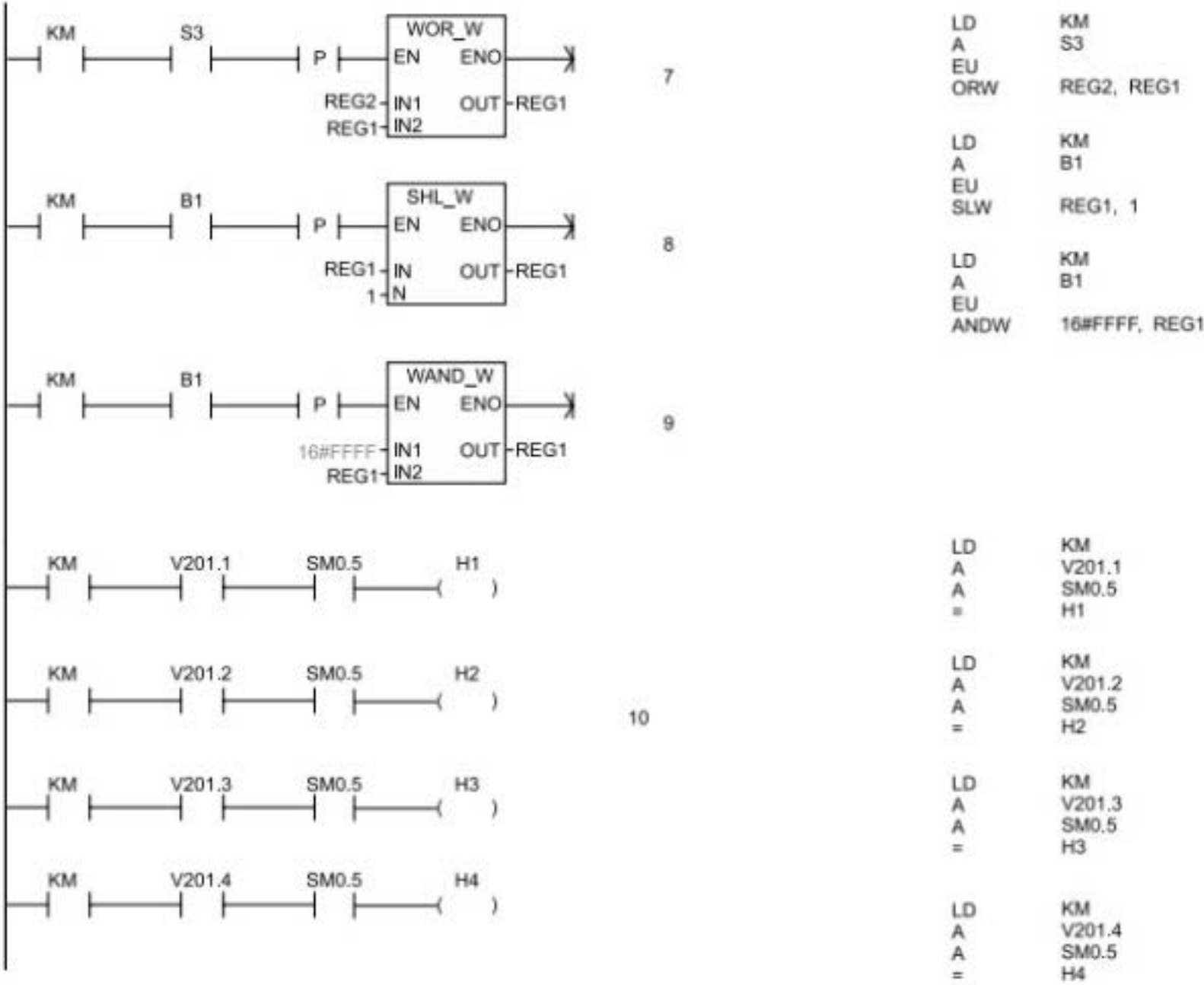


Figura 8.27 (Continuação)

- 1. Nas primeiras 5 linhas de programa temos um relé passo-passo com o botão S1, que comanda a esteira transportadora por meio do contator KM.
- Lembramos que com o relé passo-passo, pressionado o botão S1, a esteira parte. Pressionando novamente o mesmo botão S1, a esteira para.
- 2. Na sexta linha de programa temos o botão de reset S2 do sistema. Esse botão de reset carrega no início do ciclo os registros REG1 e REG2, que são as Words de amostra e comparação. A Figura 8.28 apresenta a configuração da Word de amostra REG1.

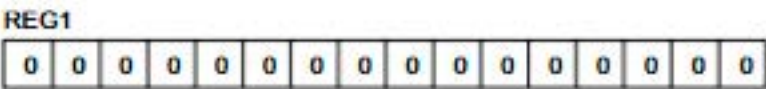


Figura 8.28

A Figura 8.29 apresenta a configuração da Word REG2 de comparação.

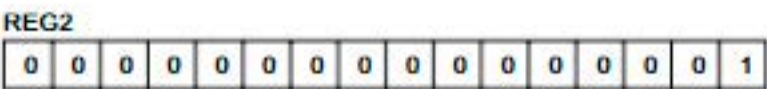


Figura 8.29

- 3. Na sétima linha de programa temos a instrução WOR_W em combinação lógica OR e a Word entre REG1 e REG2.
- Se na estação de teste surge um motor defeituoso, o operador pressiona o botão S3, que detecta o produto defeituoso, enviando, assim, um pulso na instrução WOR_W, que se habilita.
- Em seguida se efetua a comparação lógica OR entre REG1 e REG2.
- O estado das Words REG1 e REG2 é apresentado na Figura 8.30.
- O bit V201.0 está setado, o operador pressiona novamente o botão S1 e a esteira parte outra vez.
- O motor é assim detectado pela fotocélula B1.

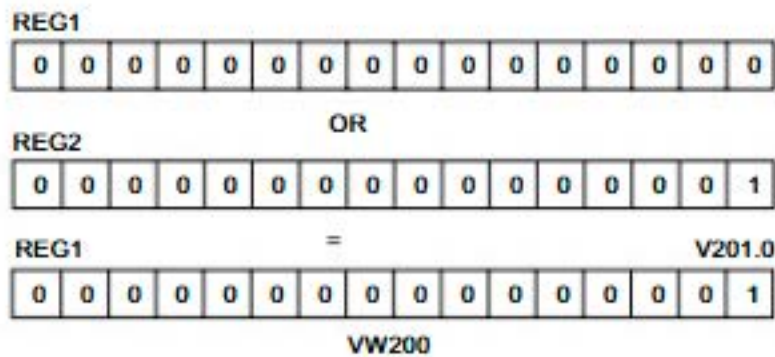


Figura 8.30

4. Na oitava linha de programa, com o pulso fornecido na fotocélula B1, o bit "1" do shift register se desloca de uma posição à esquerda. O estado da Word REG1 é mostrado na Figura 8.31.



Figura 8.31

Ao ser setado, o bit V201.1 provoca a ligação intermitente da lâmpada da estação H1 na passagem da peça defeituosa.

A movimentação de uma outra peça na esteira provoca a detecção das peças pela fotocélula B1. O pulso de B1 causa o deslocamento do bit "1" do shift register de mais uma posição à esquerda. O estado da Word REG1 é mostrado na Figura 8.32.

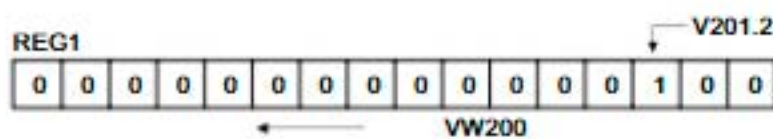


Figura 8.32

Ao ser setado, o bit V201.2 provoca a ligação intermitente da lâmpada da estação H2 na passagem da peça defeituosa.

Assim, sucessivamente a cada pulso fornecido pela fotocélula B1, devido ao movimento da esteira, o bit "1" do shift register se desloca à esquerda de uma posição.

5. Na nona linha de programa temos o mascaramento do registro REG1.

Com essa operação se evita que os bits que não são utilizados atrapalhem a operação de deslocamento.

O mascaramento se dá com uma operação de AND lógico entre o registro REG1 e o valor máscara 16#FFFF.

6. Na décima linha de programa, os bits ao nível "1" do registro REG1 (VW200) energizam a saída correspondente, ligando assim a lâmpada de sinalização lampejante por meio do bit especial SM0.5, alertan-

do assim o operador da estação subsequente que o motor elétrico não deve ser testado enquanto estiver defeituoso.

No caso de se apresentar sobre a esteira um outro motor elétrico defeituoso, o operador, pressionando o botão S3, carrega no registro REG1 um outro bit "1". O pulso fornecido pela fotocélula B1 desloca à esquerda o novo bit "1". Poderemos ter assim duas sinalizações luminosas lampejantes simultaneamente na esteira, como apresentado na Figura 8.33.

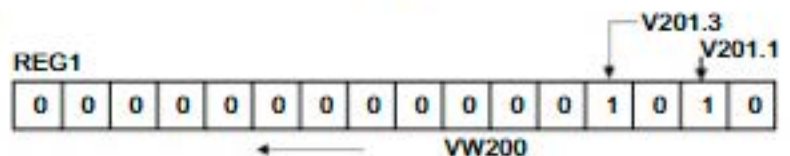


Figura 8.33

Temos assim a ligação das lâmpadas H1 e H3 simultaneamente.

Questões práticas

- Escreva um programa que energize em sequência quatro contadores K1M, K2M, K3M e K4M, ativados por um mesmo botão S0.

Segundo esta sequência (Tabela 8.4):

Tabela 8.4

	K1M	K2M	K3M	K4M
1 pulso S0	1	0	0	0
2 pulsos S0	1	1	0	0
3 pulsos S0	1	1	1	0
4 pulsos S0	1	1	1	1

1 = contator energizado

0 = contator desenergizado

- Codifique o seguinte esquema Ladder em linguagem FBD.

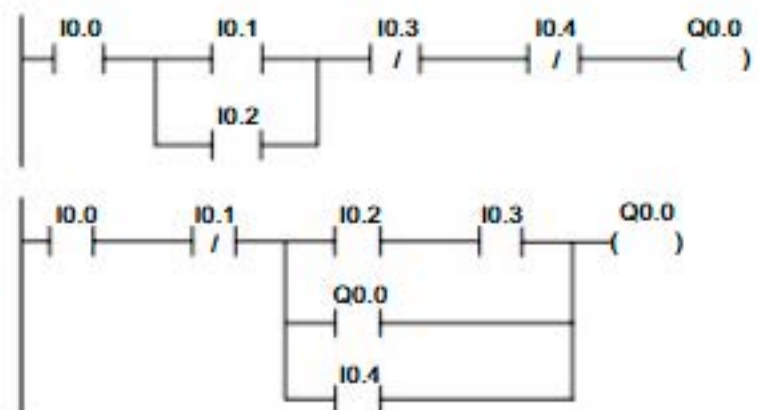


Figura 8.34

3. Codifique o seguinte esquema FBD em linguagem Ladder:

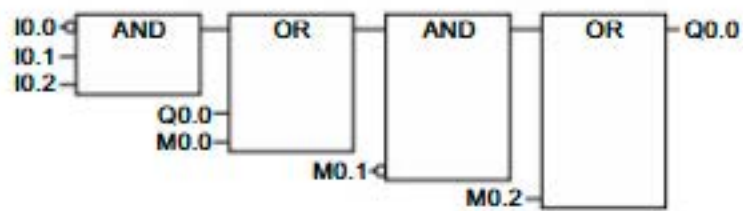
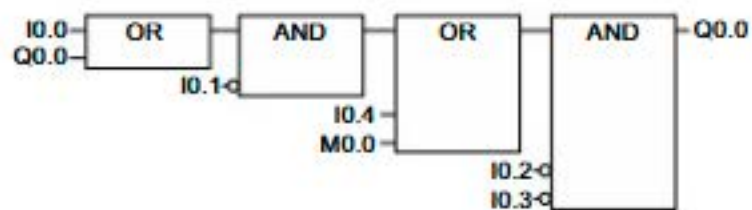
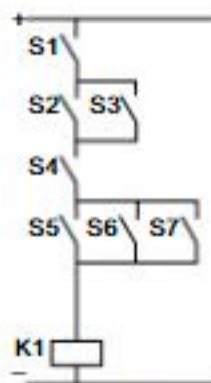
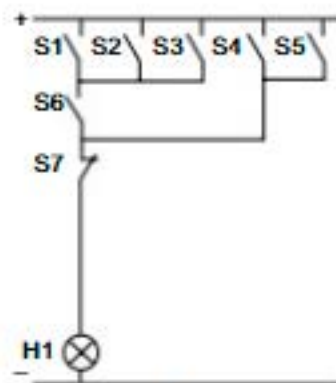


Figura 8.35

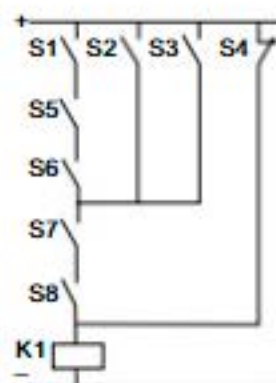
4. Explique a função do mascaramento das entradas nos PLCs.
5. Codifique os seguintes esquemas funcionais em FBD e linguagem Ladder.



a



b



c

Figura 8.36