Mechatronics

PLC Applications
Traffic Lights

Job Sheets - Courseware Sample

85249-F0

Order no.: 85249-30
First Edition
Revision level: 06/2015

By the staff of Festo Didactic

Internet: www.festo-didactic.com
e-mail: did@de.festo.com

Printed in Canada

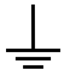
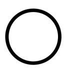
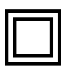
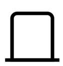
ISBN 978-2-89640-250-2 (Printed version)
Legal Deposit – Bibliothèque et Archives nationales du Québec, 2008
Legal Deposit – Library and Archives Canada, 2008

Safety and Common Symbols

The following safety and common symbols may be used in this manual and on the equipment:

Symbol	Description
⚠ DANGER	**DANGER** indicates a hazard with a high level of risk which, if not avoided, will result in death or serious injury.
⚠ WARNING	**WARNING** indicates a hazard with a medium level of risk which, if not avoided, could result in death or serious injury.
⚠ CAUTION	**CAUTION** indicates a hazard with a low level of risk which, if not avoided, could result in minor or moderate injury.
CAUTION	**CAUTION** used without the *Caution, risk of danger* sign , indicates a hazard with a potentially hazardous situation which, if not avoided, may result in property damage.
	Caution, risk of electric shock
	Caution, hot surface
	Caution, risk of danger
	Caution, lifting hazard
	Caution, hand entanglement hazard
	Notice, non-ionizing radiation
	Direct current
	Alternating current
	Both direct and alternating current
	Three-phase alternating current

Safety and Common Symbols

Symbol	Description
	Earth (ground) terminal
	Protective conductor terminal
	Frame or chassis terminal
	Equipotentiality
	On (supply)
	Off (supply)
	Equipment protected throughout by double insulation or reinforced insulation
	In position of a bi-stable push control
	Out position of a bi-stable push control

We invite readers of this manual to send us their tips, feedback, and suggestions for improving the book.

Please send these to did@de.festo.com.

The authors and Festo Didactic look forward to your comments.

Table of Contents

To the Instructor

Lab-Volt's *Programmable Logic Controller, Basic Principles Using the Programming Software* student manual (P/N 36017) allowed familiarization with PLCs and ladder programming. This was accomplished with the help of Lab-Volt's Programmable Logic Controller Training System (Model 3240).

The aim of the present series of PLC applications is to integrate the basic principles previously acquired by designing small-scale systems that can be found in the real world. Through practical examples, students will gain a strong know-how of the PLC field of study.

Each manual of the PLC applications series concentrates on a specific example of PLC application that evolves along a path of increasing complexity. With each manual, new components are added to the PLC module to create different opportunities to learn.

This Instructor Guide provides the instructor with complete solutions to all procedure questions found in the corresponding student manual. The instructor should be familiar with PLCs to recognize erroneous results. It is advised that a complete runthrough of each job sheet be included in the instructor's preparation for class. Each Job Sheet has several performance objectives. The instructor should ensure that each student understands them.

Sample

Extracted from

Job Sheets - Student

Basic Traffic Light System

On locations where roads intersect, ways to regulate the traffic are established so that vehicles and pedestrians can circulate as smoothly and securely as possible. Traffic lights, roundabouts, and signs (e.g. yield or stop) are some of the most popular methods employed. A particular method is chosen in favor of others depending on the type of traffic, the site configuration, and the local customs.

A typical intersection controlled by traffic lights is depicted in Figure 1-1. On this picture, two roads intersect at 90 degrees and pedestrians can get from one walkway to another. A view of the Traffic Lights module is also presented. It is assumed that Side 1 of the Traffic Lights module corresponds to the East-West direction whereas Side 2 relates to the South-North direction.

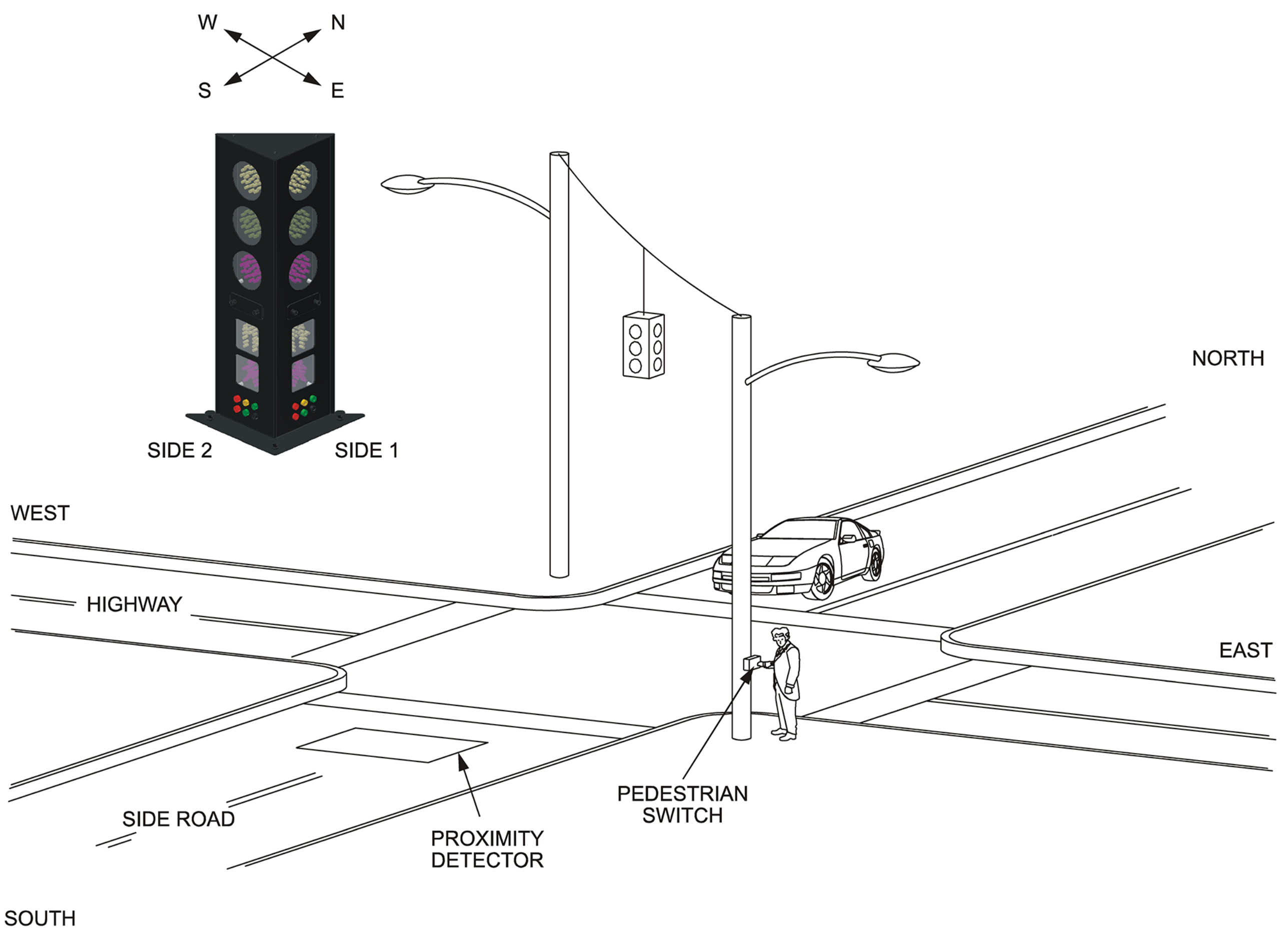


Figure 1-1. Typical Two-Road Intersection.

Often, you will find metal cabinets standing on such street corners (see Figure 1-2). These cabinets are where the lights controlling gear is located. A traffic lights controller key switch can be mounted in that same cabinet, thus permitting manual control of the intersection by the police force.

Proximity detectors can be installed under the surface of the road to detect the presence of vehicles. Likewise, pushbuttons and dedicated lights can be installed to manage pedestrian traffic. Note, however, that the traffic light control program gains in complexity with a greater number of variables. Other elements that can complicate the whole system are priority lights, traffic signal preemption for emergency vehicles, and coordinated signals with other street lights.

Figure 1-2. Traffic Lights Cabinet.

Traffic light bulbs are usually provided in two different diameters: 8 and 12 inches (20 and 30 cm). In recent years, incandescent bulbs are being replaced by **LEDs** (Light-Emitting Diodes), which are more energy efficient and durable.

In this manual, the metal cabinet on the street corner with the controlling equipment is replaced by a PLC module. The PLC is used to monitor input signals and control the traffic lights made of LEDs. Input signals from proximity detectors, pedestrian pushbuttons, or manual control by the police force can be simulated through pushbuttons and switches. The whole system is working on low-voltage provided by a 24 V dc power supply. The PLC is programmed and monitored using a computer running a ladder programming software.

Program Design

Different programming languages exist for PLCs. Some are graphical (ladder logic, function block diagram, sequential function chart) while others are textual (structured text, instruction list). This manual, like the Lab-Volt student manual *Programmable Logic Controller, Basic Principles Using the Programming Software* (P/N 36017-00), concentrates on ladder logic.

Before actually coding the ladder programs, it is well-advised to produce a draft of what the program should look like. Several tools can be employed for that purpose, including **state diagrams**, **state tables**, **flowcharts**, **boolean equations**, and **timing diagrams**.

Note: *Examples of how to employ those tools are detailed in Appendix B.*

Program Initialization

To avoid unexpected outcomes, it is good practice to start a program by setting data files to precise values. This is particularly important with sensitive files such as outputs or control bits.

A way to realize this in practice is to use a condition that is only true during the first processor **scan**. With Micrologix Controllers from Allen-Bradley, S:1/15 is a reserved status bit that is only actuated during the first pass. As an example, Figure 1-3 shows how all outputs of slot 0 can be reset at the beginning of the program.

Note: *The instruction CLR (**clear**) sets all the bits in a word to zero.*

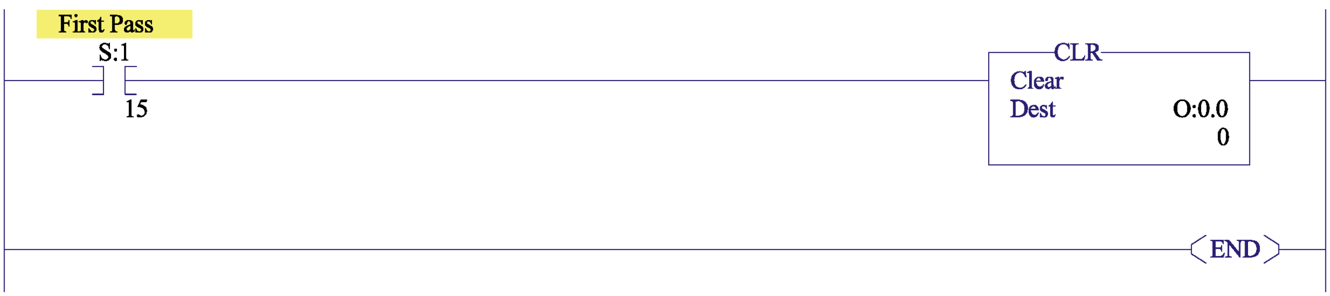


Figure 1-3. Outputs Reset Upon Initial Scan.

Comparison Instructions

Micrologix Controllers from Allen-Bradley permit the following comparison instructions:

- Equal (EQU);
- Not equal (NEQ);
- Less than (LES);
- Less than or equal (LEQ);
- Greater than (GRT);
- Greater than or equal (GEQ).

In order for a comparison to take place, two sources must be entered.

Source A: Must be a word **address** (e.g. timer accumulator T4:0.ACC).
Source B: Can be a word address or a constant (e.g. 56).

When the condition is met between the two arithmetical values, the instruction becomes true.

Timer On-Delay (TON) Instruction

Traffic lights use one or more timers to permit sequences of definite time durations. With Micrologix Controllers from Allen-Bradley, the TON instruction starts to count time base intervals when the **rung** in which it is contained goes from false to true. As long as the rung remains true, the TON instruction increases its **accumulated value** until it reaches the **preset** value. If the rung becomes false, the accumulated value is reset to zero regardless of whether or not the preset value has been reached.

Here are the different timer status bits that you can use to control relay-type (XIO and XIC) instructions in your program (timer T4:0 given as an example):

- **Done bit (T4:0/DN):** Set to logic state 1 when the accumulated value becomes equal to the preset value, and reset to logic state 0 when the rung becomes false.
- **Timer timing bit (T4:0/TT):** Set to logic state 1 when the TON rung is true and the accumulated value is less than the preset value.
- **Timer enable bit (T4:0/EN):** Set to logic state value 1 when the TON rung is true.

Here are the timer status words that you can use in your program's comparison instructions (Timer T4:0 given as an example):

- **Preset value word (T4:0.PRE):** Final value the accumulated value must reach to stop timing and set the done bit.
- **Accumulated value word (T4:0.ACC):** Value accumulated since the timer was last reset to zero.

Basic Traffic Light System

OBJECTIVE

- Create a ladder program permitting sequenced control of a traffic light.
- Use timing diagram, state diagram, state table, flowchart, and boolean equations.
- Use timer and comparison instructions.

PROCEDURE

☐ 1. The traffic light ladder program that you are about to write must respect the following conditions:

- The first rung features a TON instruction that resets after every cycle.
- The traffic lights sequence is always green - yellow - red - green.
- This sequence must repeat continuously.
- Green lights stay on for ten seconds.
- Yellow lights stay on for three seconds.
- There is always a light on in each direction, including one red light.
- "Walk" and "Don't Walk" lights are not used.
- Each one of the six outputs (see Table 1-1) has a dedicated rung.

The ladder program uses only the following RSLogix instructions:

XIC - XIO - TON - GEQ - LES - OTE

☐ 2. Make these output connections:

Address	Output
O:0/0	Red Light 1 (R1)
O:0/1	Yellow Light 1 (Y1)
O:0/2	Green Light 1 (G1)
O:0/3	Red Light 2 (R2)
O:0/4	Yellow Light 2 (Y2)
O:0/5	Green Light 2 (G2)

Table 1-1. Output Connections of Job Sheet 1.

☐ 3. Draw the timing diagram. Refer to Appendix B if necessary.

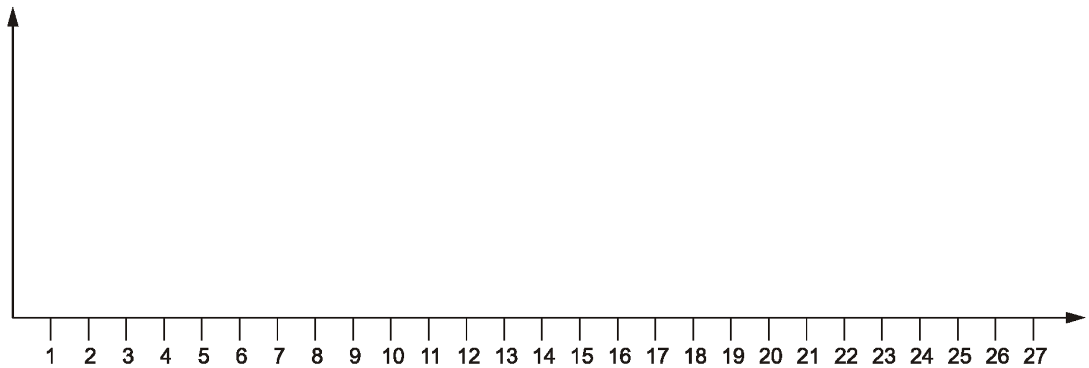


Figure 1-4. Timing Diagram of Job Sheet 1.

How many seconds does one complete system cycle last?

How many different states does the system go through during one cycle?

☐ 4. Draw the state diagram. Refer to Appendix B if necessary.

How many transitions do you observe (excluding the first pass)?

Define the conditions of the transitions.

(e.g.: T1: FROM State S1 TO State S2 UPON TIMER.ACC < 20s).

☐ 5. Fill up the state table. Refer to Appendix B if necessary.

State	G2	Y2	R2	G1	Y1	R1

Table 1-2. State table of Job Sheet 1.

☐ 6. Draw the flowchart. Refer to Appendix B if necessary.

☐ 7. Write down the boolean equations. Refer to Appendix B if necessary.

____

____

____

____

____

____

____

____

____

☐ 8. Start RSLogix500 and translate the equations into a ladder program.

☐ 9. Test and comment your ladder.

☐ 10. Demonstrate the operation of the traffic light system to your instructor.

☐ 11. Print your ladder.

☐ 12. Close RSLogix 500 and turn off the PLC.

Name: ______________________ Date: ______________

Instructor's approval: ______________________________

DRAWINGS AND CALCULATIONS

Sample

Extracted from

Job Sheets - Instructor

Basic Traffic Light System

ANSWERS TO PROCEDURE STEP QUESTIONS

☐ 3.

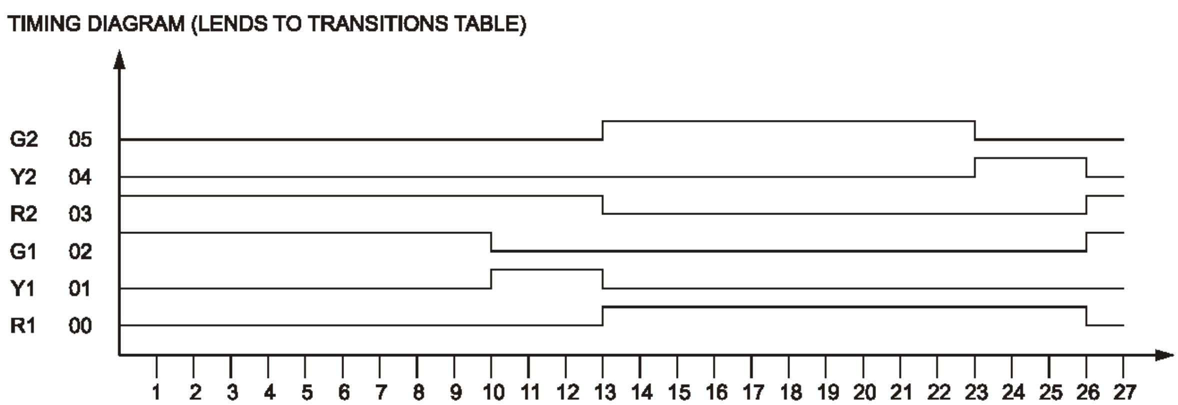


Figure 1-1. Timing Diagram of Job Sheet 1.

26 seconds

4 states

☐ 4.

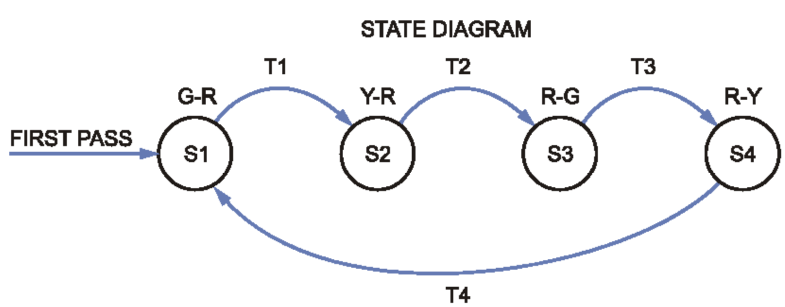


Figure 1-2. State Diagram of Job Sheet 1.

4 transitions

T1: FROM State 1 TO State 2 UPON TIMER.ACC ≥ 10s

T2: FROM State 2 TO State 3 UPON TIMER.ACC ≥ 13s

T3: FROM State 3 TO State 4 UPON TIMER.ACC ≥ 23s

T4: FROM State 4 TO State 1 UPON TIMER.ACC ≥ 26s

☐ 5.

State	G2	Y2	R2	G1	Y1	R1
1	0	0	1	1	0	0
2	0	0	1	0	1	0
3	1	0	0	0	0	1
4	0	1	0	0	0	1

Table 1-2. State table of Job Sheet 1.

☐ 6.

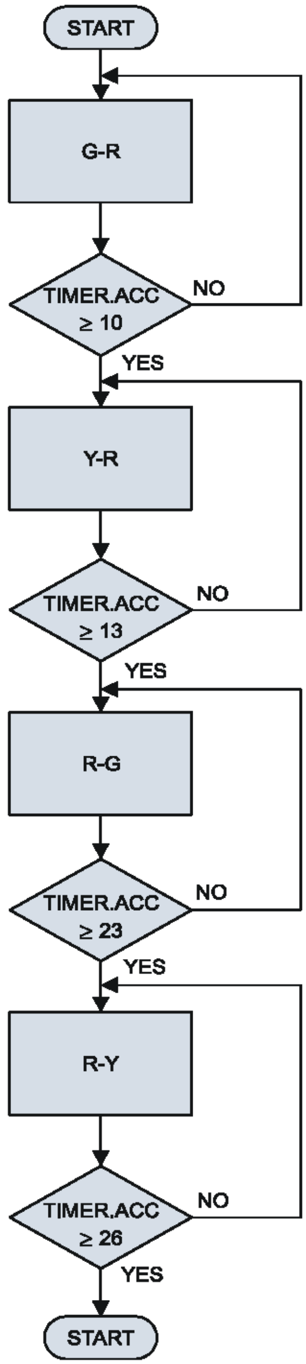


Figure 1-3. Flowchart of Job Sheet 1.

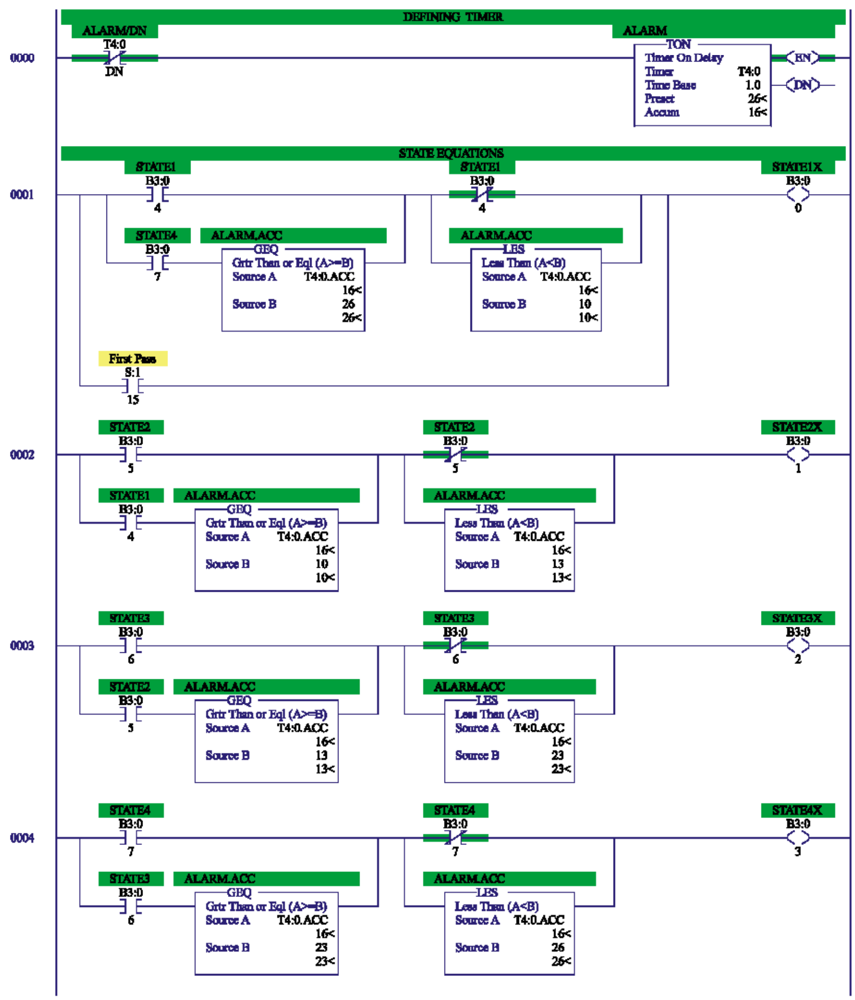
☐ 7.

$$S1=(S1+S4\cdot T4)\cdot\overline{S1\cdot T1}=(S1+S4\cdot(TIMER.ACC\geq 26))\cdot(\overline{S1}+(TIMER.ACC<10))+First\ Pass$$
$$S2=(S2+S1\cdot T1)\cdot\overline{S2\cdot T2}=(S2+S1\cdot(TIMER.ACC\geq 10))\cdot(\overline{S2}+(TIMER.ACC<13))$$
$$S3=(S3+S2\cdot T2)\cdot\overline{S3\cdot T3}=(S3+S2\cdot(TIMER.ACC\geq 13))\cdot(\overline{S3}+(TIMER.ACC<23))$$
$$S4=(S4+S3\cdot T3)\cdot\overline{S4\cdot T4}=(S4+S3\cdot(TIMER.ACC\geq 23))\cdot(\overline{S4}+(TIMER.ACC<26))$$

Notes: • $\overline{TIMER.ACC\geq x}=TIMER.ACC<x$

• Putting the "First Pass" variable in the first state equation is equivalent to setting the system in state 1 during initialization.

☐ 10.



(a)

Figure 1-4a. Ladder Program of Job Sheet 1.

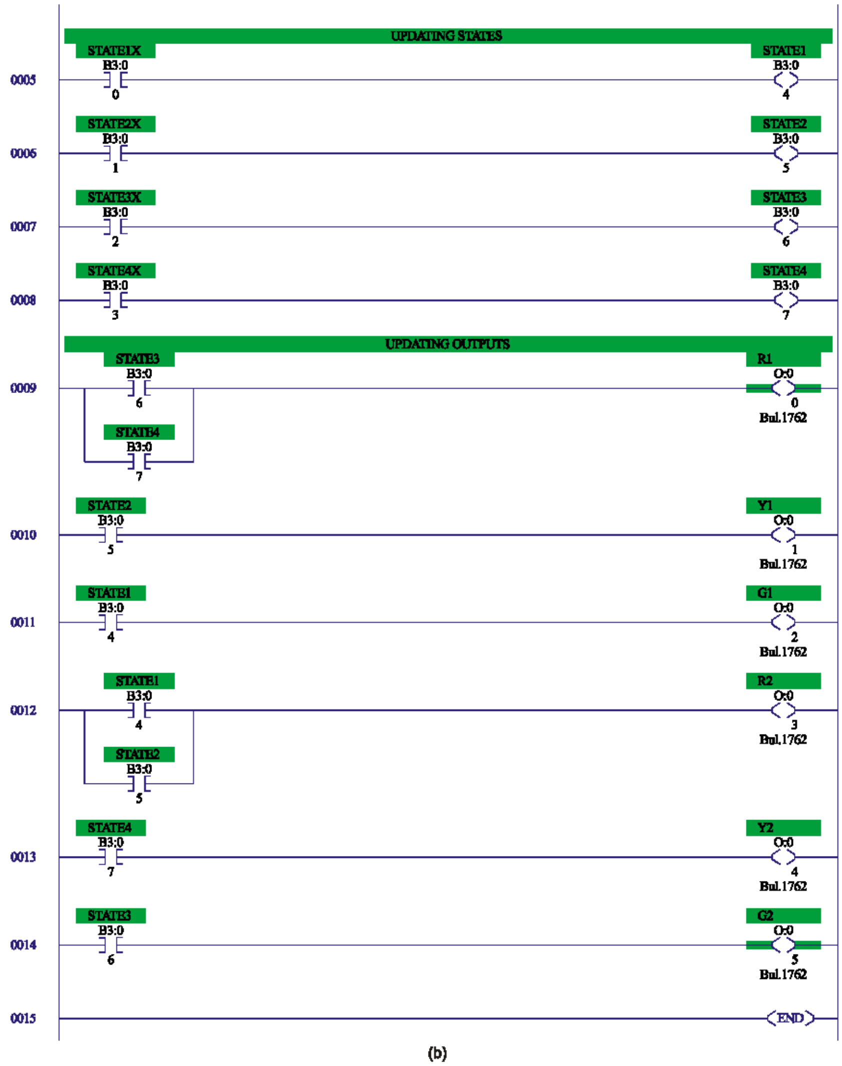


Figure 1-4b. Ladder Program of Job Sheet 1 (cont'd).