Bulletin 1745
SLC™ Programmable Controllers

User's Manual

Publication 1745-800 – November, 1987

Also contains the

Self-Teach Manual

Publication 1745-800A – March, 1988

[redacted]

ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

User's Manual

[redacted]

[redacted]

Publication 1745-800 – November, 1987
40065-236-01(C)
Supersedes Publication 1745-800 Dated September, 1985

How to Use This Manual

The User's Manual

The User's Manual contains the necessary information to install, program, and operate the Bulletin 1745 SLC Programmable Controllers. The manual has been written with the assumption that the reader has a basic knowledge of industrial relay control techniques. On this basis, understanding and applying the controller should be simple and straightforward.

We have divided the manual into three sections:

Section A – Becoming Familiar with the Equipment

Introduces you to the controller components, discusses programming basics as they apply specifically to SLC controllers, then shows you how to use the pocket programmer. You will enter a simple program into memory, in preparation for the next section.

Section B – The Instruction Set, Programming Techniques

Introduces you to the various instructions, beginning with relay-type instructions, timers, and counters. It then moves on to the more complex instructions, some of which combine two or three of the basic instructions. Keystroke examples are included to help you gain an understanding of how to apply the instructions to your particular needs. The section ends with a summary of program editing techniques and on-line data control. A chapter on using the optional EEPROM module is also included.

Section C – Installing and Maintaining the System

Contains the controller specifications and important information on system layout, installation, and start-up. We have also included a chapter on maintenance and troubleshooting.

The Self-Teach Manual

This companion to the User's Manual is designed to

- Help the first-time PC user gain an understanding of Bulletin 1745 SLC 100 and SLC 150 Programmable Controllers.
- Supplement the programming portion of the User's Manual, providing additional examples and exercises.

The Self-Teach Manual consists of 16 Question/Exercise (Q/E) units, corresponding to Chapters 2 thru 17 of the User's Manual. We suggest that after you read a chapter of the User's Manual, read it a second time, then refer to the corresponding Question/Exercise Unit in the Self-Teach Manual, where you will be asked questions, shown examples, and practice programming.

We believe this to be an efficient technique in gaining an understanding of Bulletin 1745 SLC Programmable Controllers.

Related Documentation

The following publications will be helpful to you in applying Bulletin 1745 SLC Programmable Controllers.

	Publication
Application Considerations for Solid State Controls – Describes some important differences between solid state programmable controller products and hard-wired electromechanical devices.	SGI-1.1
Allen-Bradley Programmable Controller Grounding and Wiring Guidelines	1770-4.1
Product Data – SLC 100 Processor Unit	1745-2.1
Product Data – SLC 100 Expansion Units	1745-2.2
Product Data – Pocket Programmer	1745-2.3
Product Data – EEPROM Memory Module	1745-2.4
Product Data – SLC 150 Processor Unit	1745-2.5
Product Data – SLC 150 Expansion Unit	1745-2.6
Product Data – SLC 150 High Speed Input Module	1745-2.7
Product Data – SLC 100/150 Communications Protocol	1745-2.11
Product Data – Personal Computer Interface Kit	1745-2.15
Quick Reference Guide	1745-801
Demonstrator Exercises	1745-802
SLC Personal Computer Software User's Manual	1745-825
TCAT User's Manual	1745-850
TCAT Demonstrator Exercises	1745-852

Contents

A Becoming Familiar with the Equipment

A Becoming Familiar with the Equipment

Chapter 1 Introduction

These are microprocessor-based programmable controllers. They make up a technologically advanced control system having the inherent flexibility and advantages characteristic of other programmable controllers–but with an important difference: Simplicity. They were designed with the first-time user in mind.

You will find these controllers easy to program, operate, and maintain, allowing you to take full advantage of their capabilities in the shortest possible time.

Hard-Wired Control Versus the Bulletin 1745

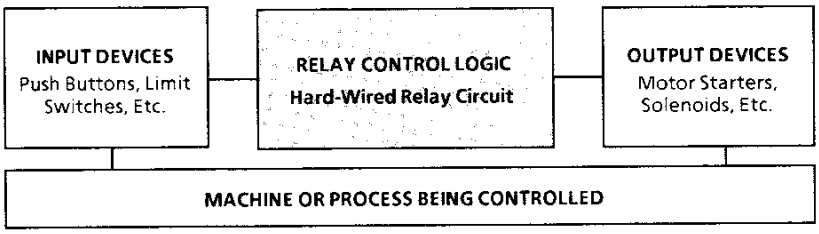
A comparison of a hard-wired relay logic system and a system using the SLC Programmable Controller will show the similarities which make the Bulletin 1745 controller so easy to apply.

The following figure shows a hard-wired relay control logic system. For purposes of comparison, it is divided into sections, consisting of:

Input Devices, which include devices operated manually (push buttons) and devices operated automatically (limit switches) by the machine or process being controlled.

Relay Control Logic, consisting of relays, timers, etc. interconnected to energize or de-energize output devices in response to the status of the input devices, and in accordance with the logic designed into the circuit.

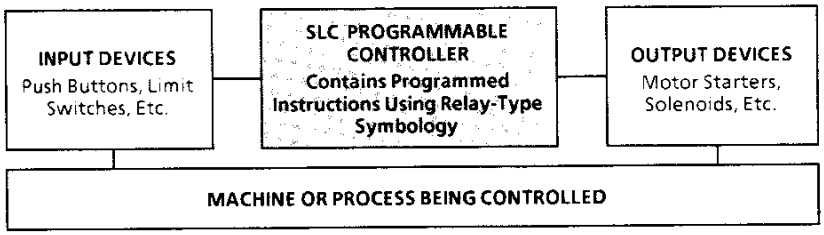
Output Devices, consisting of motor starters, solenoids, etc. which control the machine or process.



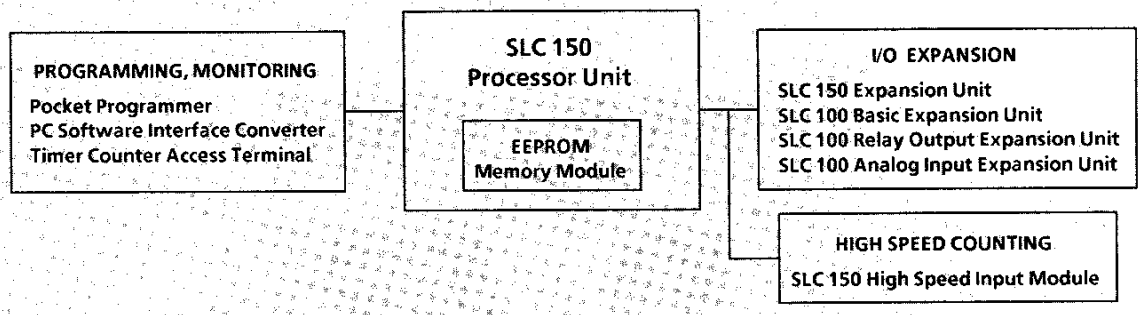
A similar control system, using the Bulletin 1745 controller is shown below. The only difference in this figure is that the relay control logic section is replaced by the SLC Programmable Controller.

In place of hard-wired relay circuitry, the SLC Programmable Controller uses programmed instructions,with **relay-type symbology**.

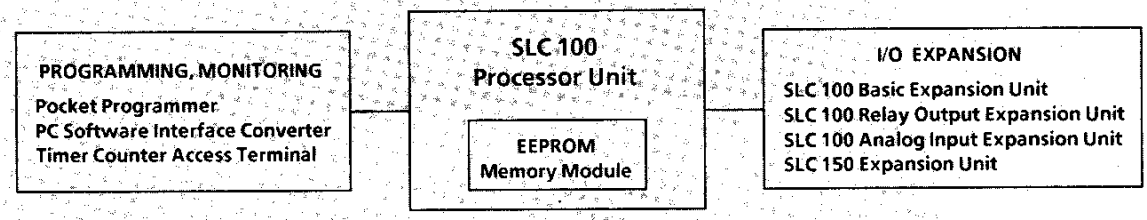
With the Bulletin 1745, you are still employing familiar relay logic methods–but in a new way, which provides a great deal more flexibility.



Chapter 2 Equipment Overview



A – Programmable Controller using the SLC 150 Processor Unit



B – Programmable Controller using the SLC 100 Processor Unit

General

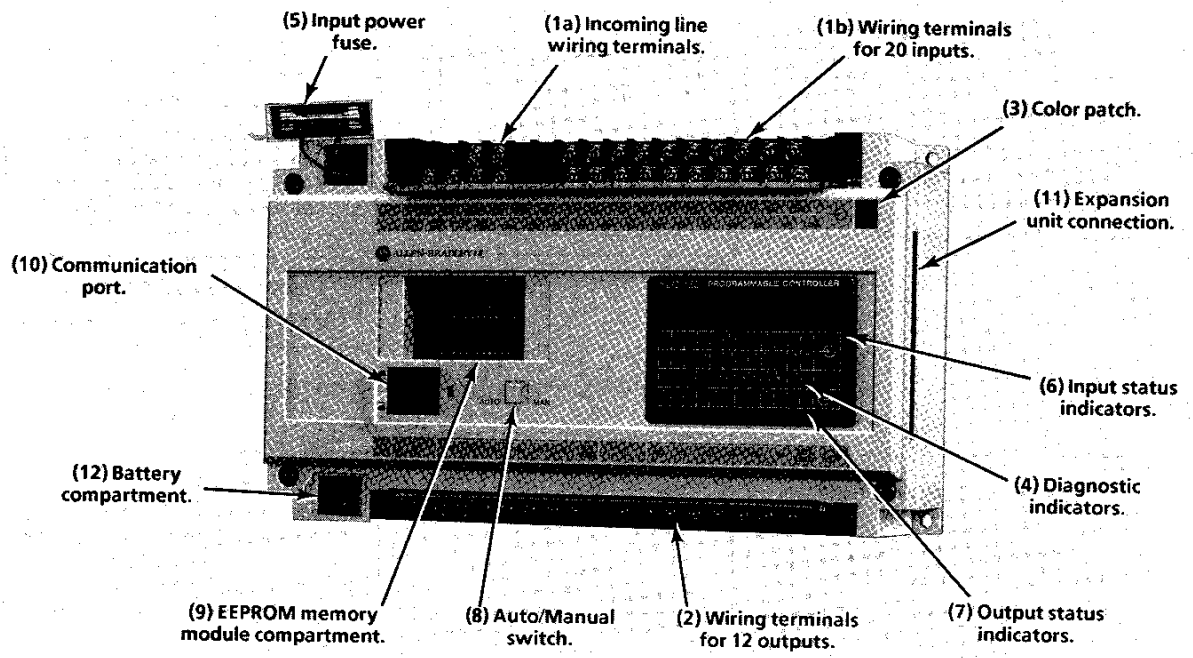
The diagrams above list the various SLC Programmable Controller components. Diagram A indicates the components you can use with the SLC 150 Processor Unit. Programming and monitoring devices, shown at the left, include the pocket programmer, PC software interface converter, and timer counter access terminal (TCAT). These devices plug into the communication port of the processor.

I/O expansion units, shown at the right, are connected by cable to the processor unit. They can be used in various combinations to obtain the I/O configuration you require. If your application involves high speed counting, you can also add high speed input modules.

The EEPROM memory module, shown in the center, plugs directly into the processor unit.

Diagram B indicates the components you can use with the SLC 100 Processor Unit. These are the *same* components used with the SLC 150, with one exception. The SLC 150 high speed input module can be used with the SLC 150 processor unit only.

Important features are pointed out on the following pages:



SLC 150 Processor Unit

The following features are pointed out in the illustration above.

1. a) Incoming line wiring terminals. b) Wiring terminals for 20 inputs. Self-lifting pressure plates allow for easy wire insertion and secure connections. Terminals accept two #14 AWG wires. The hinged cover (shown in the open position) has write-on areas for identification of external circuits. The terminal block is removable for easy processor unit replacement.
2. Wiring terminals for 12 outputs. The removable terminal block has the same construction as the line-input terminal block. Hinged cover (shown in the open position) has write-on areas.
3. Color patch. Red, black, blue, green, purple, or yellow. Identifies the 6 processor unit versions. See General Specifications, Chapter 18.
4. Five red LED diagnostic indicators:

   DC POWER – Indicates that the processor unit is energized and DC power is being supplied.

   PC RUN – Indicates the processor unit is in the Run mode.

   CPU FAULT – Indicates the processor has detected an error in the CPU, expansion units, or memory. Outputs are turned off and operation is automatically stopped.

   BATTERY LOW – An optional battery provides back-up power for the CMOS RAM memory. This LED alerts you when the battery voltage level has fallen below a threshold level.

   FORCED I/O – Indicates that one or more input or output addresses have been forced to an ON or OFF state.

SLC 150 Processor Unit *(continued)*

5. Input power fuse compartment. If line terminal voltage is present but the DC POWER LED is not lit, the fuse may be blown. Refer to Chapter 22 for fuse replacement procedure.

6. Input status indicators. Twenty red LEDs, identified with address numbers 1 thru 10 and 101 thru 110, corresponding to numbers below the input wiring terminals. When an input circuit is energized, the corresponding status indicator will be lit.

7. Output status indicators. Twelve red LEDs, identified with address numbers 11 thru 16 and 111 thru 116, corresponding to numbers above the output wiring terminals. When a programmed output instruction is TRUE, the corresponding output status indicator will be lit, and the corresponding output circuit will be energized.

8. Auto/Manual switch. This switch controls restarting of the processor unit after a power loss, brown-out, or correction of a CPU fault.

   Auto – On power-up, the processor runs thru its normal diagnostic tests and then automatically enters the Run mode (if it was in the Run mode at the last power-down).

   Manual – On power-up, the processor runs thru its diagnostic tests but will not enter the Run mode. To enter the Run mode, you must move the switch to the auto position or use the pocket programmer (or personal computer).

9. EEPROM memory module compartment. The optional memory module can be plugged into the processor. (Cat. No. is 1745-M1. SLC 150 processor requires a Series B module or a Series A module having a 28-pin chip.) The pocket programmer or personal computer software allows you to store your processor RAM program in the EEPROM. You can also load a program from the EEPROM into the processor RAM. In addition, the processor unit has an Auto-Load feature.

   Further details on using the EEPROM module appear in Chapter 17.

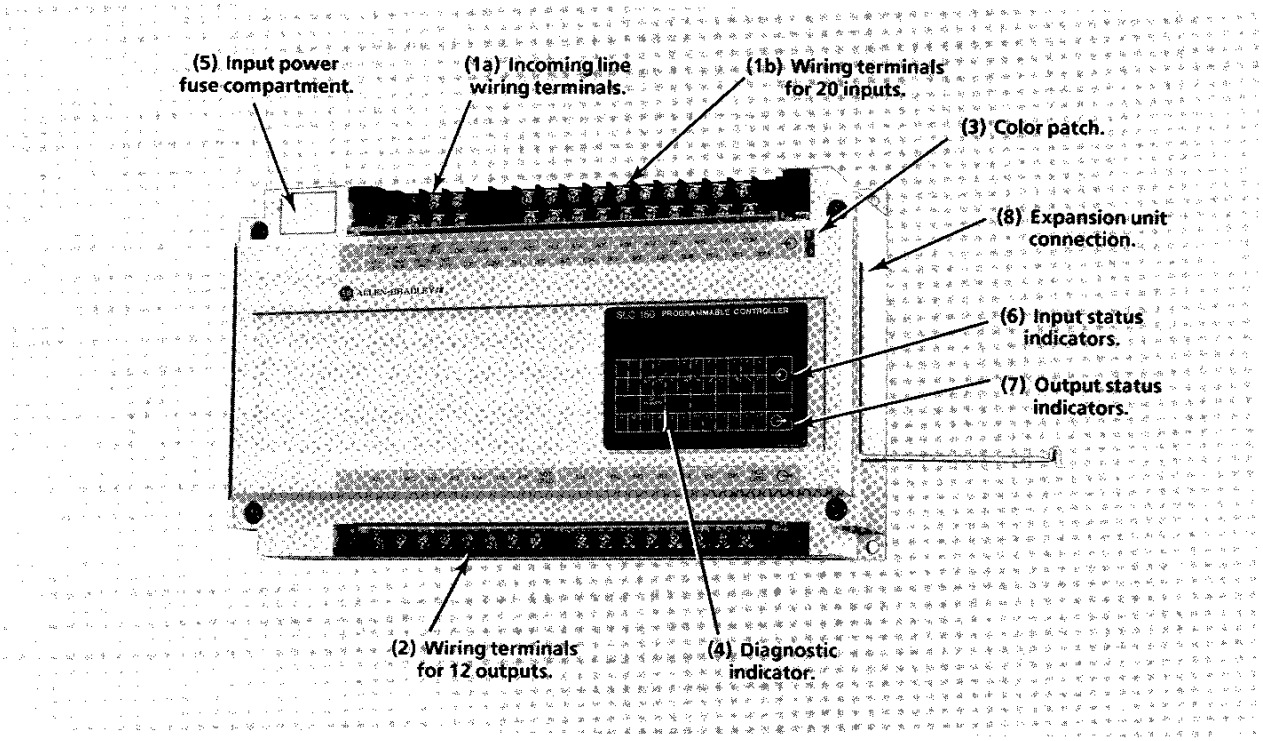
> **We recommend that you install an EEPROM memory module.** This will provide maximum protection against user program loss or program alteration due to capacitor back-up drain, battery back-up drain, processor malfunction, or excessive noise.

10. Communication port. The pocket programmer, interface converter, or TCAT cable is plugged into this socket.

11. Socket for connecting an expansion unit or high speed input module. We've included a 20-pin to 10-pin ribbon cable with the processor unit. **Save this cable.** You will need it if you want to connect an I/O expansion unit to the processor unit or if you want to connect both a high speed input module and an expansion unit.

12. Battery compartment. An optional battery assembly can be installed in this compartment. This will provide a typical 2-3 year back-up power for the CMOS RAM memory.

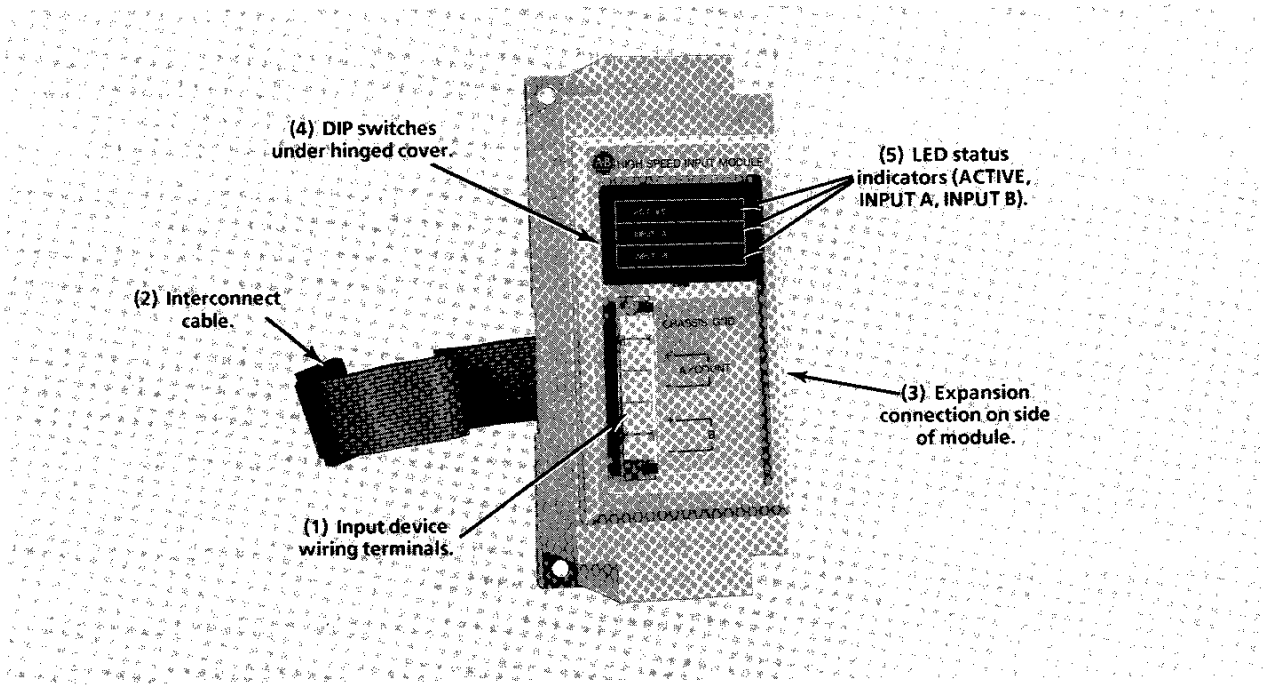
    Standard back-up power (1-2 weeks) is provided by a capacitor. Under certain conditions, battery back-up is preferable. Refer to General Specifications (Chapter 18) for further information.



SLC 150 Expansion Unit

This unit can be connected to an SLC 150 or SLC 100 processor unit. The following features are pointed out in the illustration above.

1. a) Incoming line wiring terminals. b) Wiring terminals for 20 inputs.
2. Wiring terminals for 12 outputs.
3. Color patch. Identifies the 6 expansion unit versions.
4. Diagnostic indicator: **DC POWER** – Indicates that the expansion unit is energized and DC power is being supplied.
5. Input power fuse compartment. If line terminal voltage is present but the DC POWER LED is not lit, the fuse may be blown. Refer to Chapter 22 for fuse replacement procedure.
6. Input status indicators. Twenty red LEDs, identified with address numbers A01 thru A10 and B01 thru B10, corresponding to numbers below the input wiring terminals. When an input circuit is energized, the corresponding status indicator will be lit.
7. Output status indicators. Twelve red LEDs, identified with address numbers A11 thru A16 and B11 thru B16, corresponding to numbers above the output wiring terminals. When a programmed output instruction is TRUE, the corresponding output status indicator will be lit, and the corresponding output circuit will be energized.
8. Expansion unit connection. Hinged cover is shown open. The expansion unit is interconnected with the processor unit and other expansion units via ribbon cable. See Chapter 20.



SLC 150 High Speed Input Module

The High Speed Input (HSI) module, illustrated above, is used with the SLC 150 processor unit for high speed counting applications. Up to 4 modules can be connected to the processor unit. HSI features:

1. Input device wiring terminals (under hinged cover). Self-lifting pressure plates allow for easy wire insertion and secure connections. Terminals accept #14 AWG wires. The hinged cover has write-on areas for identification of external circuits.
2. Interconnect cable. Plug this 3-inch cable into the expansion socket of the SLC 150 processor unit.
3. Expansion connection. An I/O expansion unit or another HSI module can be plugged into a socket on the side of the module. The end plug in this socket is to be removed *only* if you add an expansion unit or HSI module.
4. DIP switches. Hinged cover can be raised for access to DIP switches. Switches can be set for 24VDC, 12VDC, or 5VDC input circuit voltage. Switches are also used to select input filter delay time of 4 millisec. or 50 microsec.
5. LED status indicators. ACTIVE is On when HSI counter instruction is TRUE. INPUT A is On when HSI module receives signal from input connected to A/COUNT terminals. INPUT B is On when the B terminals of the HSI receive a signal from an input device or the B channel of a quadrature type encoder.

SLC 100 Processor Unit

The following features are pointed out in the illustration on Page 2-7.

1. Incoming line wiring terminals. Self-lifting pressure plates allow for easy wire insertion and secure connections. Terminals accept two #14 AWG wires. A hinged cover is provided (but not illustrated). The cover needn't be removed to gain access to the terminals.
2. Wiring terminals for 10 inputs. Same construction as line terminals. The hinged cover (not illustrated) has write-on areas for identification of external circuits. Cover is color-coded to identify the circuit voltage level. The cover needn't be removed to gain access to the terminals.
3. Wiring terminals for 6 outputs. Same construction as line terminals. Hinged cover (not illustrated) has write-on areas to identify external circuits. The cover needn't be removed to gain access to the terminals.

   The processor unit has relay (hard contact) output circuits.
4. Five LED diagnostic indicators:

   DC POWER (green) – Indicates that the processor unit is energized and DC power is being supplied.

   PC RUN (green) – Indicates the processor unit is in the Run mode.

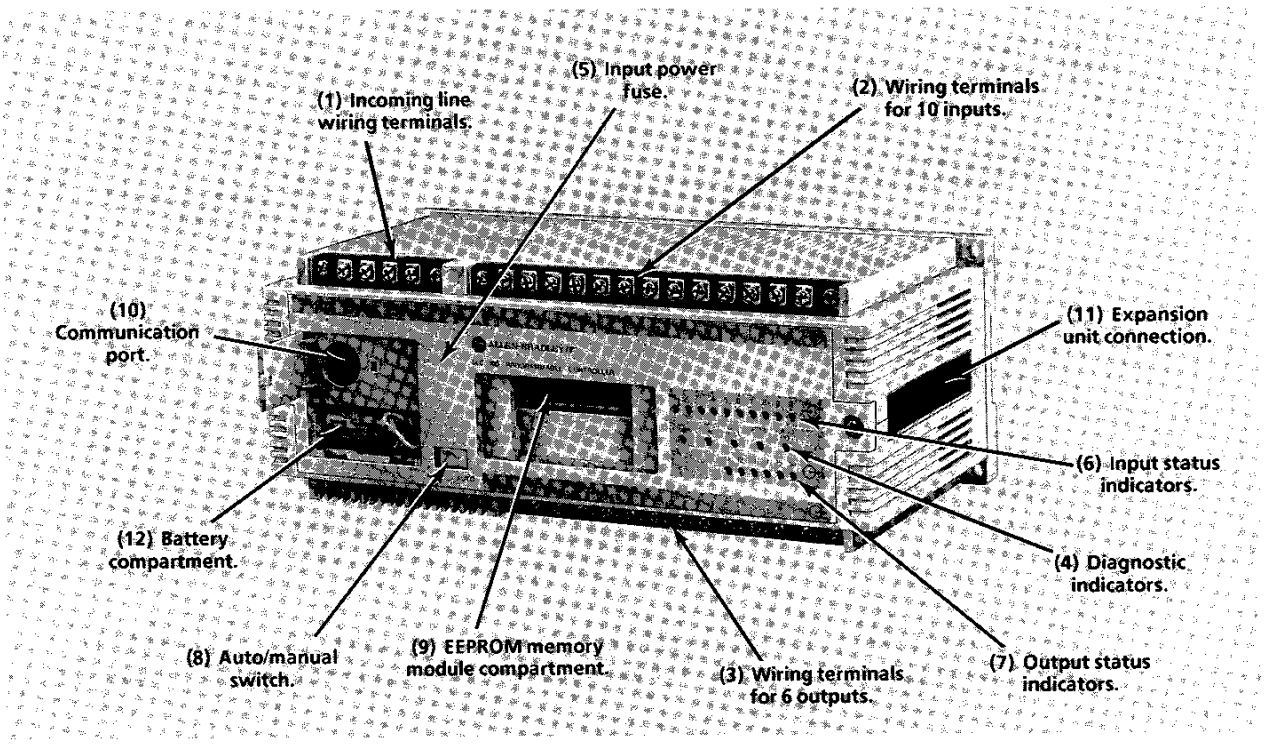
   CPU FAULT (red) – Indicates the processor has detected an error in either the CPU or memory. Operation is automatically stopped.

   BATTERY LOW (red) – A battery provides back-up power for the CMOS RAM memory. This LED alerts you when the battery voltage level has fallen below a threshold level.

   FORCED I/O (amber) – Indicates that one or more input or output addresses have been forced to an ON or OFF state.
5. Input power fuse (behind front cover). If line terminal voltage is present but the DC POWER LED is not lit, the fuse may be blown. Refer to Chapter 22 for fuse replacement procedure.
6. Input status indicators. Ten red LEDs, identified with address numbers 1 thru 10, corresponding to numbers 1 thru 10 on the input device wiring terminals. When an input circuit is energized, the corresponding status indicator will be lit.
7. Output status indicators. Six red LEDs, identified with address numbers 11 thru 16, corresponding to numbers 11 thru 16 on the output contact wiring terminals. When a programmed output instruction is TRUE, the corresponding output status indicator will be lit, and the corresponding output contact will close.
8. Auto/Manual switch. This switch controls restarting of the processor unit after a power loss, brown-out, or correction of a CPU fault.

   Auto – On power-up, the processor runs thru its normal diagnostic tests and then automatically enters the Run mode (if it was in the Run mode at the last power-down).

   Manual – On power-up, the processor runs thru its diagnostic tests but will not enter the Run mode. To enter the Run mode, you must move the switch to the auto position or use the pocket programmer (or personal computer).

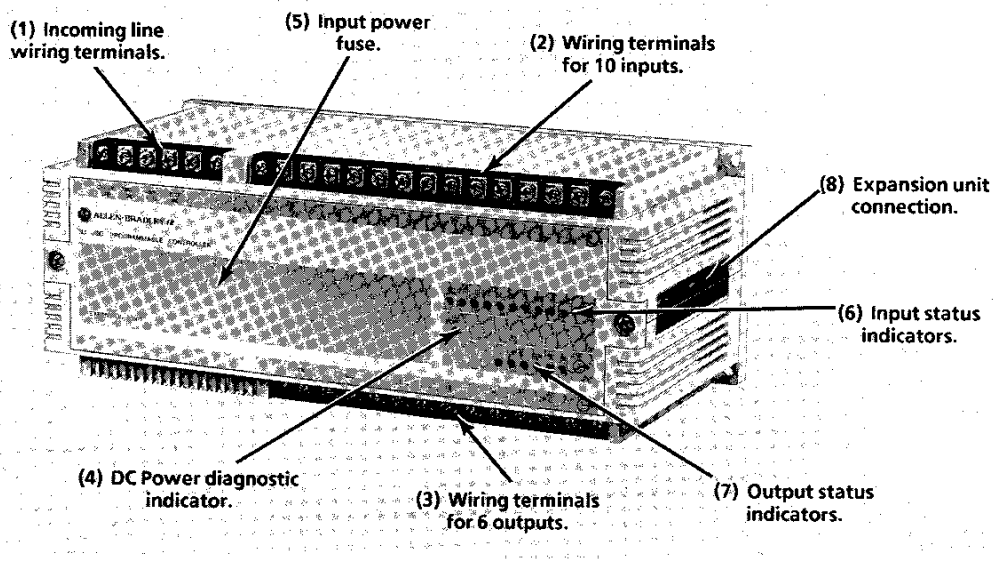


SLC 100 Processor Unit *(continued)*

9. EEPROM memory module compartment. The optional memory module can be plugged into the processor. The pocket programmer or personal computer software allows you to store your processor RAM program in the EEPROM. You can also load a program from the EEPROM into the processor RAM. Refer to Chapter 17.

> **We recommend that you install an EEPROM memory module.** This will provide maximum protection against user program loss or program alteration due to battery back-up drain, processor malfunction, or excessive noise.

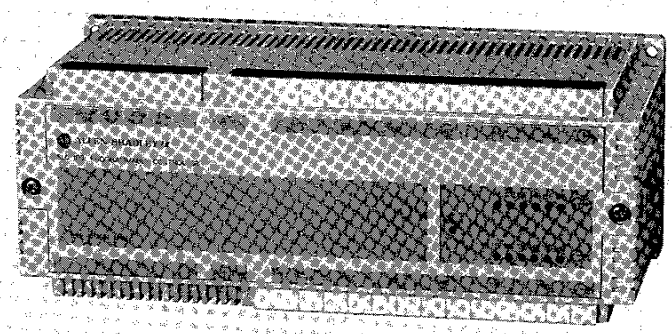
10. Communication port. The pocket programmer, interface converter, or TCAT cable is plugged into this socket.
11. Expansion unit connection. The expansion unit cable is plugged in this socket. SLC 100 and SLC 150 expansion units can be interconnected to increase the I/O capacity of the controller.
12. Battery compartment. Back-up power for the CMOS RAM is provided by a replaceable battery assembly, accessible from the front of the processor unit. The lithium battery provides back-up power for approx. 2-3 years. Battery replacement: Chapter 22.



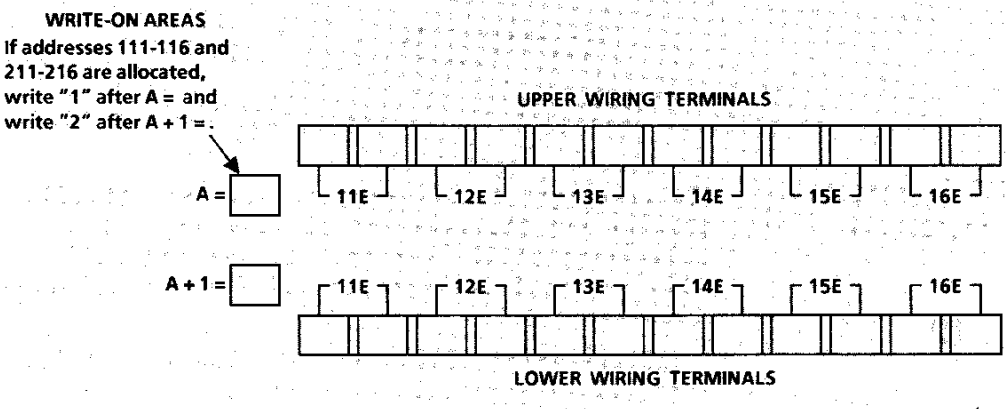
SLC 100 Basic Expansion Unit

The basic expansion unit can be used with the SLC 100 or SLC 150 processor. The following features are pointed out in the illustration above.

1. Incoming line wiring terminals. Self-lifting pressure plates allow for easy wire insertion and secure connections. Terminals accept two #14 AWG wires. A hinged cover is provided (but not illustrated). The cover needn't be removed to gain access to the terminals.
2. Wiring terminals for 10 inputs. Same construction as line terminals. The hinged cover (not illustrated) has write-on areas for identification of external circuits. Cover is color-coded to identify the circuit voltage level. The cover needn't be removed to gain access to the terminals.
3. Wiring terminals for 6 outputs. Same construction as line terminals. Hinged cover (not illustrated) has write-on areas to identify external circuits. The cover needn't be removed to gain access to the terminals.
4. Diagnostic indicator: **DC POWER** (green LED) – Indicates that the expansion unit is energized and DC power is being supplied.
5. Input power fuse (behind front cover). If line terminal voltage is present but the DC POWER LED is not lit, the fuse may be blown.
6. Input status indicators. Ten red LEDs, 1E thru 10E, correspond to input device wiring terminals 1E thru 10E. When an input circuit is energized, the corresponding status indicator will be lit.
7. Output status indicators. Six red LEDs, 11E thru 16E, correspond to output contact wiring terminals 11E thru 16E. When a programmed output instruction is TRUE, the corresponding output status indicator will be lit, and the corresponding output contact will close.
8. Expansion unit connection. The expansion unit is interconnected with the processor unit and other expansion units via ribbon cable. See Chapter 20.



Relay output expansion unit.



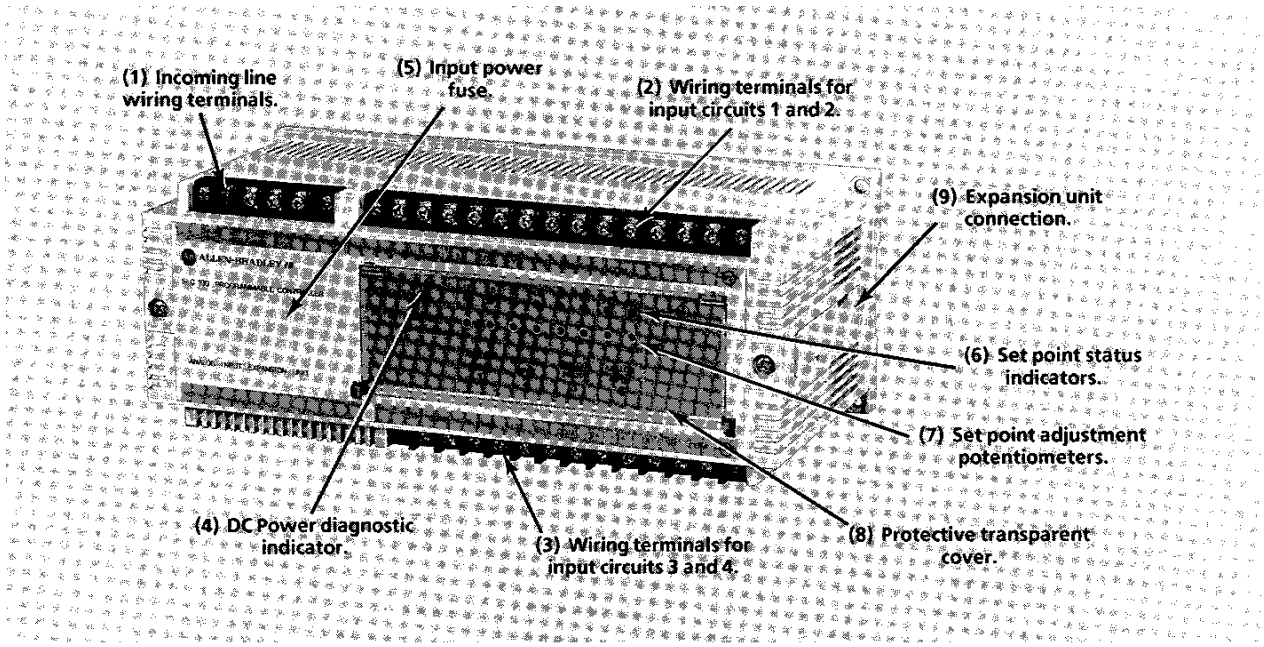
Wiring terminal identification.

SLC 100 Relay Output Expansion Unit

The relay output expansion unit can be connected to the SLC 100 or SLC 150 processor unit. It has the same construction features and dimensions as the basic expansion unit except that the 10 input circuits are replaced by 6 additional relay (hard contact) output circuits.

Relay output expansion units have wiring terminals for 6 output circuits on the upper edge of the unit and wiring terminals for 6 more output circuits on the lower edge of the unit. Write-on areas are provided for you to identify the first digit of the upper and lower addresses.

The upper group always uses the lower number address block. For example, if address blocks 2 and 3 apply to a unit, the upper group of addresses must be 111-116 (address block 2) and the lower group 211-216 (address block 3). This is illustrated in the diagram above.



SLC 100 Analog Input Expansion Unit

The analog input unit can be connected to the SLC 100 or SLC 150 processor unit. The analog unit, illustrated above, has 4 input circuits and no output circuits. Each input circuit has 2 adjustable set points which can be individually programmed. The following features are pointed out.

1. Incoming line wiring terminals. Self-lifting pressure plates allow for easy wire insertion and secure connections. Terminals accept two #14 AWG wires. A hinged cover is provided (but not illustrated). The cover needn't be removed to gain access to the terminals.

2 and 3. Wiring terminals for 4 input circuits. Same construction as line terminals. The hinged cover (not illustrated) has write-on areas for identification of external circuits. The cover needn't be removed to gain access to the terminals.

4. Diagnostic indicator: **DC POWER** (green LED) – Indicates that the expansion unit is energized and DC power is being supplied.

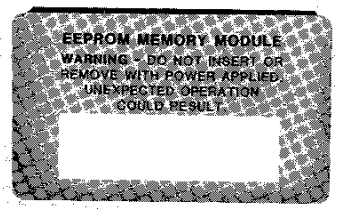
5. Input power fuse (behind front cover). If line terminal voltage is present but the DC POWER LED is not lit, the fuse may be blown.

6. Set point status indicators (red LEDs). The 8 set points (2 per input circuit) are labeled 1E thru 8E. When an input circuit set point is reached, the corresponding status indicator will be lit.

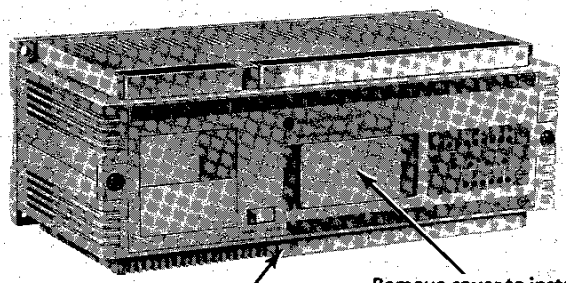
7. Set point adjustment potentiometers.

8. Protective transparent cover. Hinged to allow access to the adjustment potentiometers.

9. Expansion unit connection. The expansion unit is interconnected with the processor unit and other expansion units via ribbon cable. See Chapter 20.

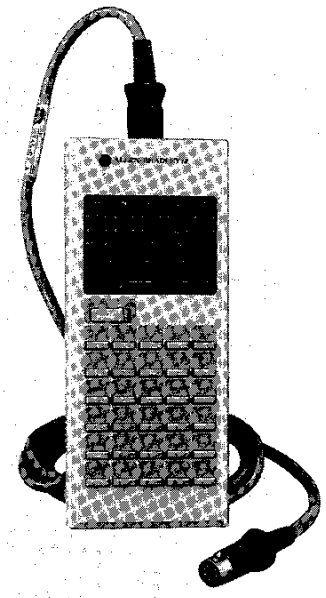


EEPROM memory module.



SLC 100 processor unit.

Remove cover to install EEPROM module.



Pocket programmer.

EEPROM Memory Module

This optional device plugs into the SLC 100 or SLC 150 processor unit. With the EEPROM module, you can:

- Save the contents of the processor RAM memory, for storage purposes.
- Load the contents of the EEPROM memory into the processor RAM.

As shown in the illustration above, a write-on area is provided for use in identifying memory contents. You will find detailed information on the EEPROM module in Chapter 17.

Pocket Programmer

The pocket programmer is used to program, edit, and monitor controller operation. This device, illustrated above, is easily connected to the communication port of the SLC 100 or SLC 150 processor unit when needed. You will find detailed information on the pocket programmer in Chapter 4.

PC Software Interface Converter

The interface converter, which plugs into the communication port of the processor unit, is part of the Personal Computer Software package. The software allows you to use an IBM or IBM-compatible personal computer in place of the pocket programmer. With the software, you can select functions equivalent to the operating modes discussed in this manual. Additional capabilities: off-line programming and program library development; ladder diagram display and enhancements; data display and cross reference table generation; program print-out. An RS-232-C/RS-422 interface converter is required for communication between the processor and computer.

Refer to the SLC Personal Computer Software User's Manual, Publication 1745-825, for application and operation information.

Timer Counter Access Terminal (TCAT)

The TCAT is used with the SLC programmable controllers to access programmed timer, counter, and sequencer data. I/O addresses can also be accessed. This allows production, supervisory, and maintenance people to monitor this data "on-line".

Refer to the TCAT User's Manual, Publication 1745-850, for application and operation information.

Chapter 3 Programming Basics

Ladder Diagram Format

The programming format for SLC controllers is the *ladder diagram*, which uses symbology similar to hard-wired relay ladder circuits. Figure 3.1 shows a relay ladder rung as used in hard-wired relay control systems and a similar PC ladder rung as used in programming the SLC controller. With the relay ladder rung, *electrical* continuity is required to energize the output, whereas in the PC ladder rung, *logical* continuity is required to energize the output.

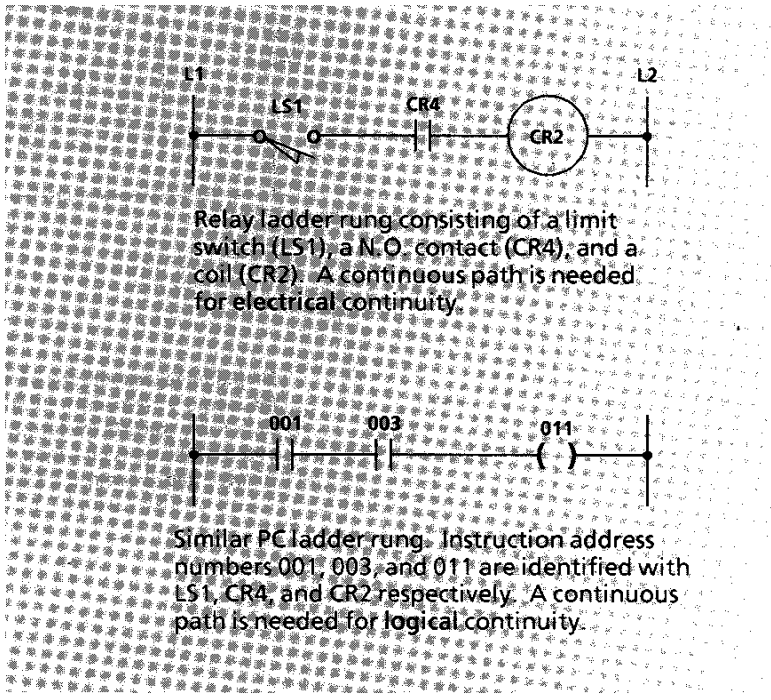
Instructions

In the PC ladder rung of Figure 3.1, the individual symbols represent *instructions*; the numbers 001, 003, and 011 are the instruction *addresses*. When programming the controller, these instructions are entered one-by-one into the processor memory from the pocket programmer keyboard. Instructions are stored in the *user program* portion of the processor memory.

Some of the instructions entered are used to represent the external input and output devices connected to the processor unit; other instructions are "internal", used to establish the exact conditions under which the processor will energize or de-energize output devices in response to the status of input devices.

The memory storage unit for instructions is the *word*, with most instructions occupying one word of memory storage space.

Figure 3.1



Relay ladder rung consisting of a limit switch (LS1), a N.O. contact (CR4), and a coil (CR2). A continuous path is needed for electrical continuity.

Similar PC ladder rung. Instruction address numbers 001, 003, and 011 are identified with LS1, CR4, and CR2 respectively. A continuous path is needed for logical continuity.

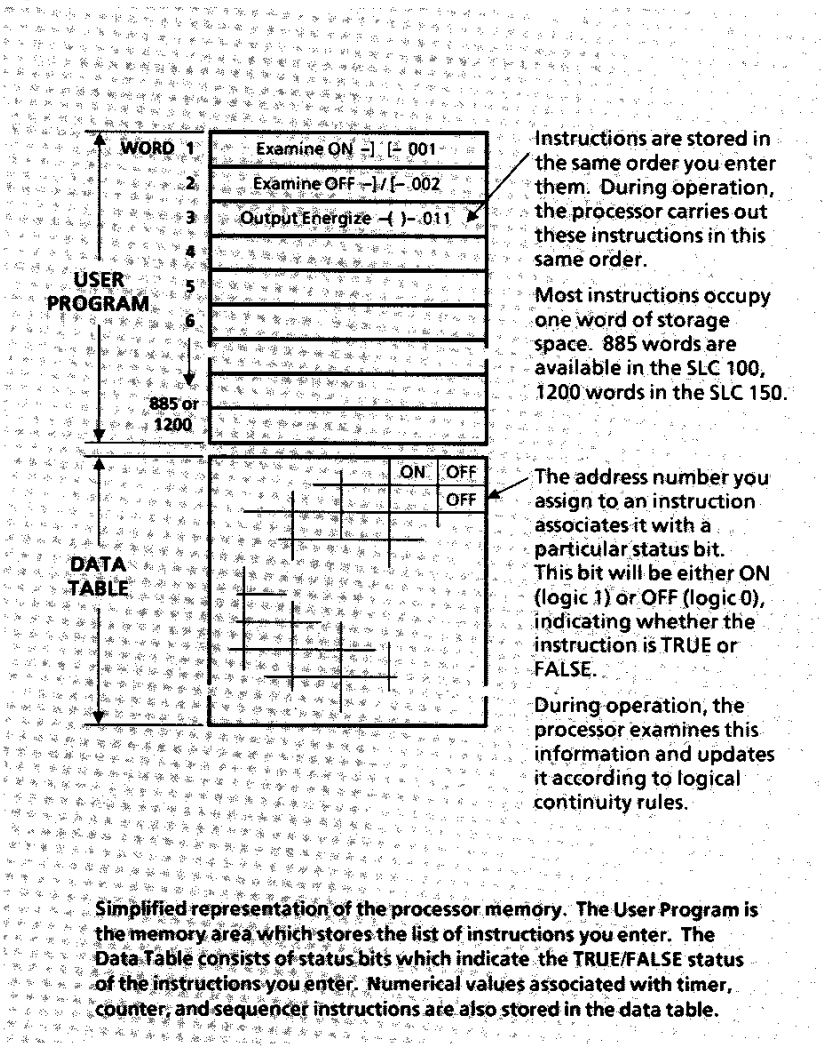
Instruction Addressing

To complete the entry of an instruction, you must assign an *address* number to it. This number identifies the function of an instruction and links it to a particular *status bit* in the *data table* portion of the memory.

The status bits in the data table can be either ON (logic 1 state) or OFF (logic 0 state), indicating the TRUE/FALSE status of the instructions they are associated with.

Figure 3.2 is a simplified representation of the user program and data table areas of the processor memory.

Figure 3.2



Simplified representation of the processor memory. The User Program is the memory area which stores the list of instructions you enter. The Data Table consists of status bits which indicate the TRUE/FALSE status of the instructions you enter. Numerical values associated with timer, counter, and sequencer instructions are also stored in the data table.

Internal Addresses

The table below lists the various *internal* instructions and the addresses you will be assigning to them in your user program. The instructions are discussed in detail in succeeding chapters.

Figure 3.3

Internal Instruction	Addresses		
Relay Type Instructions	701 thru 863 plus two addresses of each I/O address block (Page 3-5).	701 thru 866 plus two addresses of each I/O address block (Page 3-5).	5
Timer, Counter, Sequencer, and Reset	901 thru 932 and 951 thru 982	901 thru 932 and 951 thru 982	6, 7, 10, 11
HSI Counter①	901 thru 904 and 951 thru 954 (SQO instructions)	Not applicable	12
Fine Time Base	869 thru 873	869 thru 873	8
Long Time Base	874, 875	874, 875	8
Breakpoint	100	100	14
Programmable EEPROM Auto-Load①	864	Not applicable	17
Battery Status①	865	Not applicable	14
Triac Zero-Cross Turn-On Enable①	866	Not applicable	14
TCAT Power-Up	867	867	14
Program Initialization	868	868	14
Auto/Man Switch	876	876	14
MCR and ZCL	Addresses are not required		9
Shift Register	Combines Relay Type , ZCL, and Sequencer instructions		13

① Note that these instructions apply to the SLC 150 only, although the addresses apply to instructions used in the SLC 100 as well (relay type, timer, counter, sequencer). You must be cautious when interchanging programs between the SLC 150 and SLC 100:

> CAUTION: Interchanging programs between controllers could cause improper operation and damage equipment. You must be especially cautious when interchanging programs between SLC 150 and SLC 100 controllers, because of differences associated with certain internal addresses and other variations.

I/O Addresses

I/O addresses are used in your program to represent the input and output circuits connected to the processor unit and I/O expansion units. There are 112 I/O addresses available to you. These are divided into 7 *address blocks* (10 input addresses and 6 output addresses per block). The SLC 150 processor unit, having 32 I/O, uses blocks 1 and 2. The SLC 100 processor unit, having 16 I/O, uses block 1.

You can connect various combinations of SLC 150 and SLC 100 I/O expansion units to your processor unit. Assigning addresses is simple, once you understand how processor "address blocks" are allocated.

Figure 3.4 identifies the address blocks and summarizes the address block requirements for the various processor and expansion units.

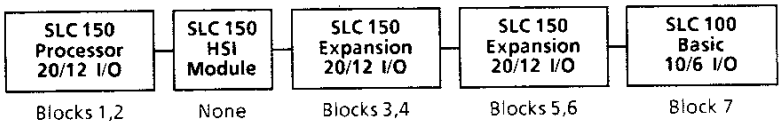
Figure 3.4

The 7 address blocks are identified in the table to the right. Address block requirements for the various processor and expansion units are shown below.

Address Block Number	Input Addresses (10)	Output Addresses (6)
1	1–10	11–16
2	101–110	111–116
3	201–210	211–216
4	301–310	311–316
5	401–410	411–416
6	501–510	511–516
7	601–610	611–616

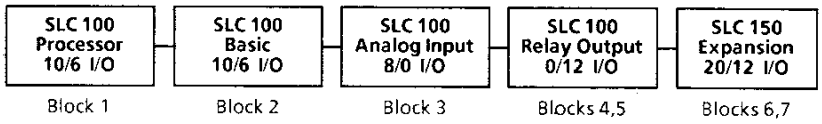
ADDRESS BLOCKS REQUIRED FOR PROCESSOR AND EXPANSION UNITS			
Type of Unit	I/O	Address Blocks Required	Addresses Used
SLC 150 Processor Unit	20/12	2	Block 1 and 2 addresses are used. Inputs: 1-10, 101-110. Outputs: 11-16, 111-116.
SLC 150 Expansion Unit	20/12	2	All addresses of 2 consecutive blocks are used.
SLC 150 High Speed Input Module	1/0	None	An internal SQO address is associated with the HSI module.
SLC 100 Processor Unit	10/6	1	Block 1 addresses are used. Inputs: 1-10. Outputs: 11-16.
SLC 100 Basic Expansion Unit	10/6	1	All addresses of the block are used.
SLC 100 Relay Output Expansion Unit	0/12	2	Output addresses of 2 consecutive blocks are used. Input addresses of the 2 blocks are bypassed.
SLC 100 Analog Input Expansion Unit	8/0	1	Upper two input addresses (_9, _10) are bypassed. All output addresses of the block are bypassed.

Connection Examples – The following examples indicate how you might interconnect expansion units with an SLC 150 processor unit and an SLC 100 processor unit. Note that example 1 includes an HSI module.



EXAMPLE 1: All input and output addresses of all 7 address blocks are used. The configuration provides 112 I/O (70 inputs, 42 outputs). It also provides 1 high speed input circuit.

I/O Addresses *(continued)*



EXAMPLE 2: All 7 address blocks are used. The configuration provides 84 I/O: 40 discrete inputs (blocks 1, 2, 6, 7 – addresses 1-10, 101-110, 501-510, 601-610); 8 analog set points (block 3 – addresses 201-208); and 36 outputs (blocks 1, 2, 4, 5, 6, 7 – addresses 11-16, 111-116, 311-316, 411-416, 511-516, 611-616).

I/O Terminal Identification – The table below indicates the terminal identification of the four types of I/O expansion units. The table also indicates the address blocks, input addresses, and output addresses that apply if you connect the expansion unit to an SLC 150 processor unit.

Expansion Unit	I/O Terminal Identification	[illegible]	[illegible]
SLC 150	Inputs: A01-A10 and B01-B10. Outputs: A11-A16 and B11-B16.	3 and 4	Terminals: A = 2 and B = 3. Input addresses: 201-210 and 301-310. Output addresses: 211-216 and 311-316.
SLC 100 Basic	Inputs: 1E-10E. Outputs: 11E-16E.	3	Input addresses: 201-210. Output addresses: 211-216.
SLC 100 Relay Output	Upper terminals (A): 11E-16E. Lower terminals (A + 1): 11E -16E.	3 and 4	Terminals: A = 2 and A + 1 = 3. Upper addresses: 211-216. Lower addresses: 311-316. Input addresses of blocks 3 and 4 are bypassed.
SLC 100 Analog Input	8 set points derived from the 4 input circuits: 1E-8E.	3	Set point addresses: 201-208. Output addresses of block 3 are bypassed.

Maximum I/O Configuration – The maximum I/O configuration is 112, using all available addresses. If you use SLC 100 relay output or analog input expansion units, the maximum configuration will be less, and varies with the particular combination of expansion units used. Keep in mind that when you've used all 7 address blocks, you've reached the maximum.

Internal Addresses Associated with I/O Address Blocks – The table below lists *internal* relay-type instruction addresses associated with I/O address blocks. You can use these addresses in your program in the same way as the relay-type instructions discussed in Chapter 5, but *only if expansion units using the associated address blocks are connected to the processor unit.*

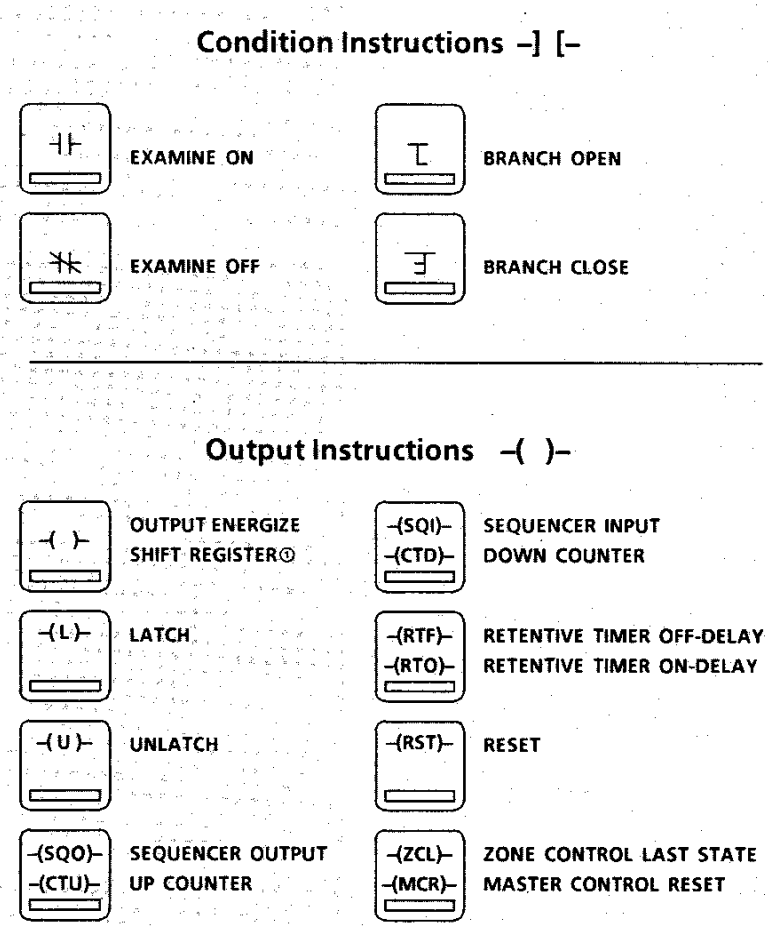
If you are using the SLC 100 processor unit, addresses 17 and 18 (address block 1) can always be used. If you are using the SLC 150 processor unit, addresses 17, 18, 117, and 118 (address blocks 1 and 2) can always be used.

Relay-Type Instruction Addresses	Associated Address Block Number
17, 18	1
117, 118	2
217, 218	3
317, 318	4
417, 418	5
517, 518	6
617, 618	7

Instruction Classifications

Instructions are classified as *condition* instructions and *output* instructions. The instruction *set* is shown in Figure 3.5.

Figure 3.5



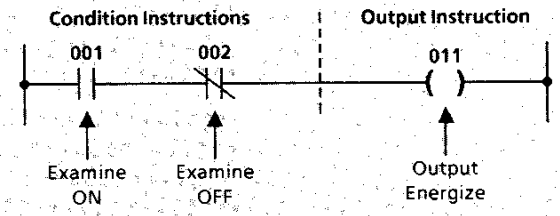
Instruction set of the SLC controller. Symbols are representations of the pocket programmer keys.

① The Shift Register symbol is –(SR)–. It does not appear on the programmer key.

Logical Continuity

An example of how condition and output instructions are used is shown in the ladder diagram of Figure 3.6. The *Examine ON* and *Examine OFF* instructions (conditional) are analogous to relay contacts, while the *Output Energize* instruction is analogous to a relay coil. However, this diagram must be evaluated in terms of *logical* continuity rather than *electrical* continuity.

Figure 3.6



Ladder diagram rung. A continuous path of TRUE condition instructions is required to make the output instruction TRUE.

As stated earlier, each instruction is linked to a status bit in the data table. The bit will be either ON or OFF to indicate the status of the instruction. Thus, with the Examine ON instruction we are asking the controller to "examine the status bit for an ON condition". If the status bit is ON, then the instruction is TRUE; if the bit is OFF, then the instruction is FALSE.

Similarly, the Examine OFF instruction means "examine the status bit for an OFF condition". If the status bit is OFF, the instruction is TRUE; if the bit is ON, the instruction is FALSE.

The Output Energize instruction asks the controller to "set the status bit of the addressed Output Energize instruction to ON when rung conditions are TRUE". Thus, in Figure 3.6, when both the Examine ON instruction and the Examine OFF instruction are TRUE, the status bit of the Output Energize instruction will be set to ON.

In terms of continuity: When there is a continuous path of TRUE *conditional* instructions in a rung, logical continuity exists; accordingly, the output instruction is TRUE and its status bit will be set ON. If any conditional instruction in the continuity path goes FALSE, logical continuity is lost; the output instruction is then FALSE and its status bit will be set to OFF.

External I/O Devices

The user program always includes instructions to represent external devices connected to the processor and expansion units. The processor examines the status of these external devices indirectly.

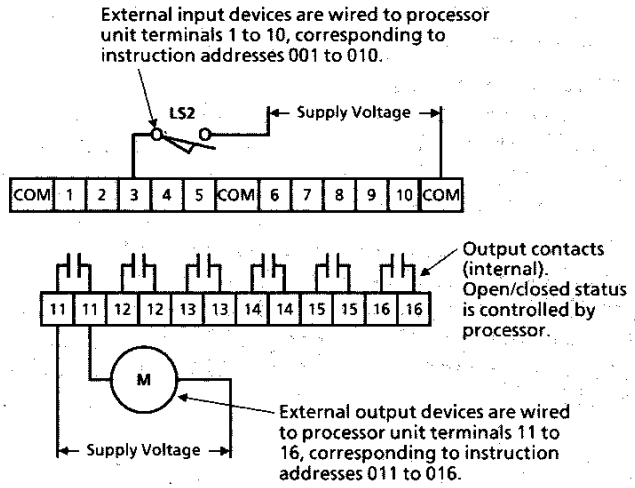
Thus, in Figure 3.7a, the limit switch contact connected to input terminal 3 causes a voltage to be present at the terminal when the switch is closed, and removes this voltage when the switch is opened. The processor senses these voltage levels, and sets the status bit of the instruction representing the limit switch to ON for the closed condition and OFF for the open condition.

If an Examine ON instruction is used to represent the limit switch contact, the open condition makes the instruction logically FALSE, and the closed condition makes it logically TRUE.

If an Examine OFF instruction is used to represent the limit switch contact, the open condition makes the instruction logically TRUE, and the closed condition makes it logically FALSE. Figure 3.7b summarizes both the Examine ON and Examine OFF cases.

Figure 3.7b also summarizes the Output Energize instruction as it applies to an external I/O address. In this case, if the status bit of the instruction is ON, the corresponding output contact (Figure 3.7a) closes, energizing the external device. If logical continuity of the rung is lost, the status bit is changed to OFF; the output contact opens, de-energizing the external device.

Figure 3.7



3.7a – External input and output device connections for the SLC 100 processor unit. Input wiring terminals 1-10 correspond to addresses 001-010. Output wiring terminals 11-16 correspond to addresses 011-016.

External Input Device	Input Terminal	Examine ON Instruction	Examine OFF Instruction
OPEN	OFF	FALSE Status Bit OFF	TRUE Status Bit OFF
CLOSED	ON	TRUE Status Bit ON	FALSE Status Bit ON

Output Energize Instruction	Output Contacts	External Output Device
Status Bit ON TRUE Rung Conditions	CLOSED	M ENERGIZED
Status Bit OFF FALSE Rung Conditions	OPEN	M NOT ENERGIZED

3.7b – Examine ON, Examine OFF, and Output Energize instructions, as they relate to external input and output devices.

Processor Operating Cycle

During each operating cycle, the processor examines the status of input devices, executes the user program, and changes outputs accordingly. For a typical 500 word program, the SLC 150 controller repeats this cycle about 500 times each second; with the SLC 100, the cycle is repeated about 67 times each second.

A single operating cycle or *scan* is shown in Figure 3.8. Note that the cycle is divided into two parts – the I/O scan and the program scan.

I/O Scan

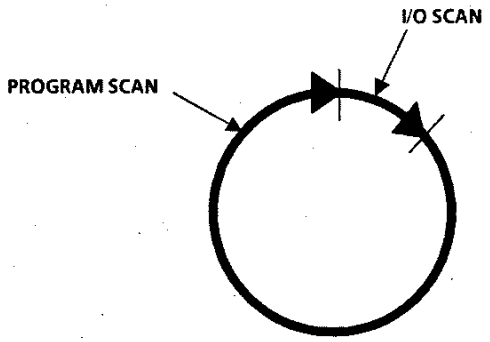
In the I/O scan, data associated with external outputs is transferred from the data table to the output terminals. (This data was up-dated during the preceding program scan.) In addition, input terminals are examined, and the associated status bits are changed accordingly.

Program Scan

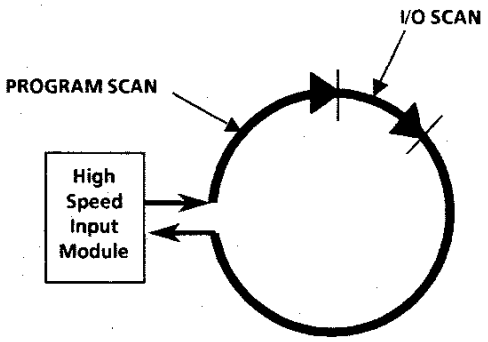
The updated status of the input devices is applied to the user program during this part of the cycle. The processor executes the entire list of instructions in the same order they were entered. Status bits are updated according to logical continuity rules as the program scan moves from instruction to instruction thru successive ladder rungs. (If you use an instruction with a unique address more than once in your program, the solution of the rung in which the instruction last appears takes precedence. An example appears in Chapter 9, Page 9-3.)

The I/O scan and program scan are separate, independent functions. Thus, any status changes occurring in external input devices during the program scan are not accounted for until the next I/O scan. Similarly, data changes associated with external outputs are not transferred to the outputs until the next I/O scan. **Exception:** If you are using the high speed input module (SLC 150 only), the operating cycle is interrupted to allow the processor unit to count high speed input signals and *immediately* set processor unit outputs associated with SQO instructions 901-904. This is illustrated below.

Figure 3.8



3.8a. Operating cycle of the processor.



3.8b. Operating cycle when using the HSI module. The SLC 150 processor interrupts the operating cycle to count high speed input signals and immediately set appropriate outputs.

Typical Scan Times

The table below lists typical scan times for the SLC 150 and SLC 100 Programmable Controllers. Approximate execution times for the individual instructions appear on Pages 18-13 and 18-14.

Note: Scan time can vary significantly depending on program content, program length, and expansion units connected. Also, with the SLC 150, scan time increases if you are using high speed input modules.

Scan times in excess of 100 milliseconds will be detected by the processor and cause system shutdown. You can measure your scan time using the techniques described on Page 8-4.

SLC 150	2.5 milliseconds	0.3 milliseconds plus 0.25 milliseconds for each expansion unit address block.	2 milliseconds
SLC 100	15 milliseconds	1.5 milliseconds plus 0.27 milliseconds for each expansion unit address block.	12.4 milliseconds

Chapter 4 Using the Pocket Programmer

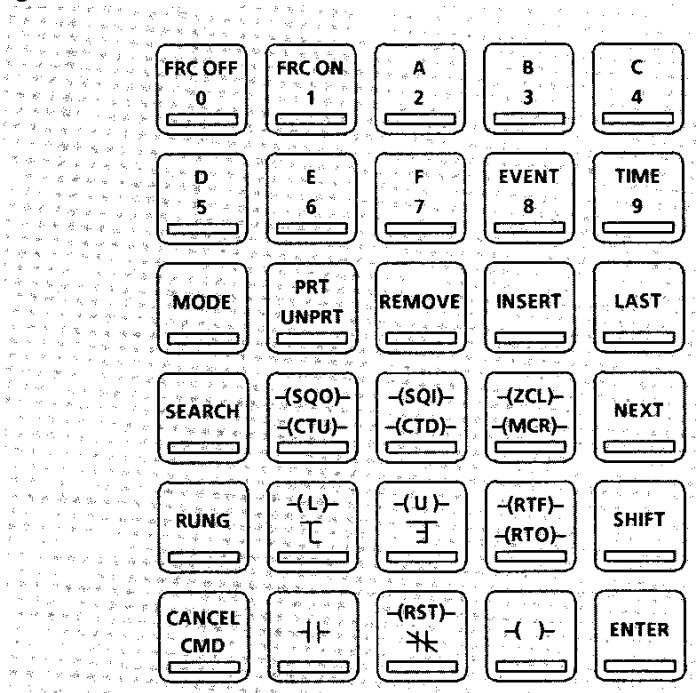
The Pocket Programmer

Instructions are entered into the processor memory with the pocket programmer. Becoming familiar with the features and basic operation of this device will prepare you for the specific programming procedures discussed in later chapters.

Keyboard

The pocket programmer keyboard is shown below. Upper case functions (FRC OFF, FRC ON, A, B, C, etc) are obtained by pressing the SHIFT key, then the desired function key.

Figure 4.1



Pocket programmer keyboard.

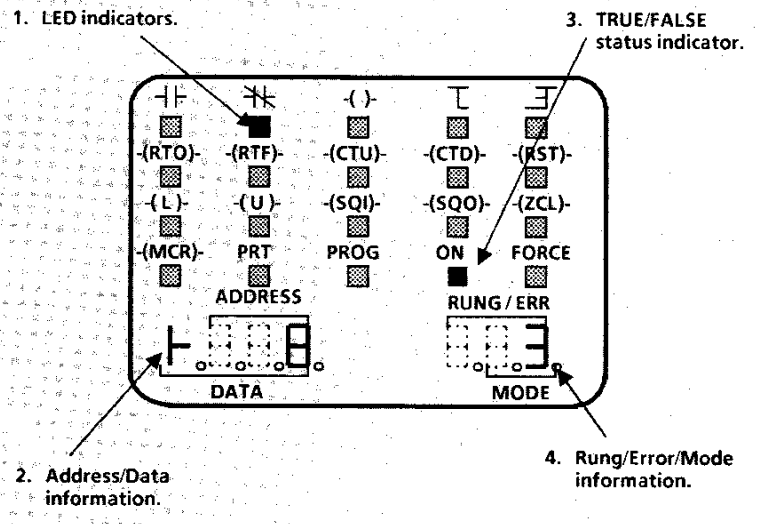
Abbreviations and Symbols			
FRC OFF	Force OFF	-(RTF)-	Retentive Timer Off-Delay
FRC ON	Force ON	-(RTO)-	Retentive Timer On-Delay
PRT	Protect	CANCEL CMD	Cancel Command
UNPRT	Not Protect	-(RST)-	Reset
-(SQO)-	Sequencer Output	⊤	Branch Open
-(SQI)-	Sequencer Input	⊣	Branch Close
-(CTU)-	Up Counter	⊣⊢	Examine ON
-(CTD)-	Down Counter	⊣/⊢	Examine OFF
-(ZCL)-	Zone Control Last State	-( )-	Output Energize
-(MCR)-	Master Control Reset		Shift Register (use shift key)
-(L)-	Latch		
-(U)-	Unlatch		

Display

The display window of the programmer is shown in Figure 4.2. The LEDs in the upper portion illuminate when instructions are entered, edited or monitored. Addresses, data, prompt messages, and other information is presented on the 7-segment displays located below the LEDs.

Figure 4.2

In this typical display, the cursor is located at an Examine OFF instruction, address 008. It is the first instruction in rung 3 of the program, and its current status is TRUE:



1. Illuminated LED indicates the cursor position, that is, the instruction being entered, edited or monitored. In this case, the Examine OFF LED is illuminated.
2. Four-character digital display shows instruction addresses and other information. The symbol at the left indicates that the cursored instruction is at the beginning of the rung. The number 8 at the right indicates the instruction address, 008.
3. TRUE/FALSE status indicator. It is illuminated, indicating that the Examine OFF instruction is TRUE. This indicator functions only in the Run or Test mode.
4. Three-character digital display shows rung, mode, or error code number. In this case, the display indicates that we are in rung 3 of the program.

After you enter a program, you can move thru it from one instruction to another by pressing the NEXT and LAST keys.

Modes of Operation

The pocket programmer is used to select the various *modes of operation*. A list of these modes appears on the back of the programmer. A detailed explanation is given below.

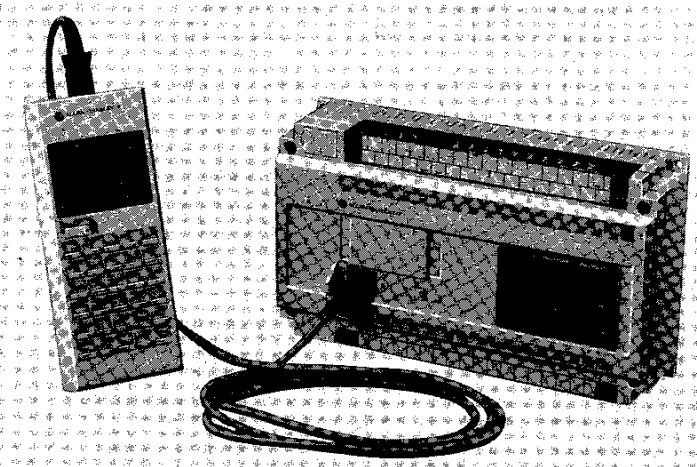
Figure 4.3

Mode #	Description	Display
1	CLEAR MEMORY: Selecting this mode erases the contents of the on-board RAM memory. Upon completion, the programmer automatically switches to mode 2, Program.	CLEr
2	PROGRAM: Used to enter a new program or up-date an existing one in the RAM memory.	prog
3	RUN: In this mode, the processor scans and executes the user program. Input devices are monitored and output devices are energized accordingly. In this mode, the programmer can be used to monitor the user program, force I/O, and change timer/counter preset and accumulated values. Sequencer preset values can also be changed.	run
4	TEST–SINGLE SCAN: This mode causes the processor to complete a single scan of the user program each time the ENTER key is pressed. No outputs will be energized. Timer and time-driven sequencer accumulated values will be incremented by 0.1 on each scan if rung conditions are correct. The programmer can be used to monitor the user program, force I/O, and change counter/timer/sequencer values.	SScn
5	TEST–CONTINUOUS SCAN: Causes the processor to operate from the user program without energizing any outputs. The programmer can be used to monitor the user program, force I/O, and change counter/timer/sequencer values.	CScn
6	STORE USER PROGRAM IN EEPROM MODULE: Allows you to *save* a program, that is, store a program contained in the on-board RAM memory in an EEPROM memory module.	SAUE
7	LOAD USER PROGRAM FROM EEPROM MODULE: This mode allows you to *read* a program into memory, that is, load a program contained in an EEPROM module into the on-board RAM memory. You can then remove the EEPROM module or leave it in place. The processor operates from the RAM only.	rEAd
8	ENTER/CHANGE ACCESS CODE: This mode allows you to enter or change an access code or *password*.	PASS
9	DIAGNOSTIC TEST–PROGRAMMER: A sequence of self-checking diagnostic tests. Refer to Page 22-15 (Maintenance and Troubleshooting) for details.	d IAG

Practice Setup

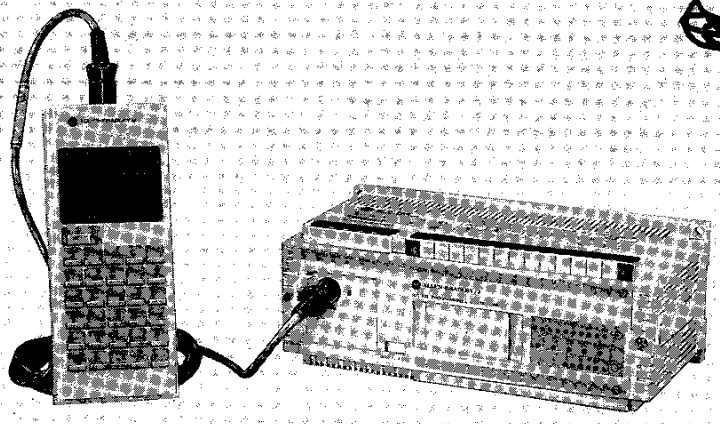
SLC 100 and 150 Demonstrator Units (Catalog Nos. 1745-DEMO1, -DEMO3) provide a convenient means for practicing the keystroke examples, monitoring techniques, and editing techniques shown in this manual. If you do not have access to a demonstrator, you can use another of the practice setups shown below.

Figure 4.4



SLC 150 processor unit with the pocket programmer plugged in. Connect a power cord to the processor as indicated in Figure 4.5.

SLC 100 demonstrator unit.

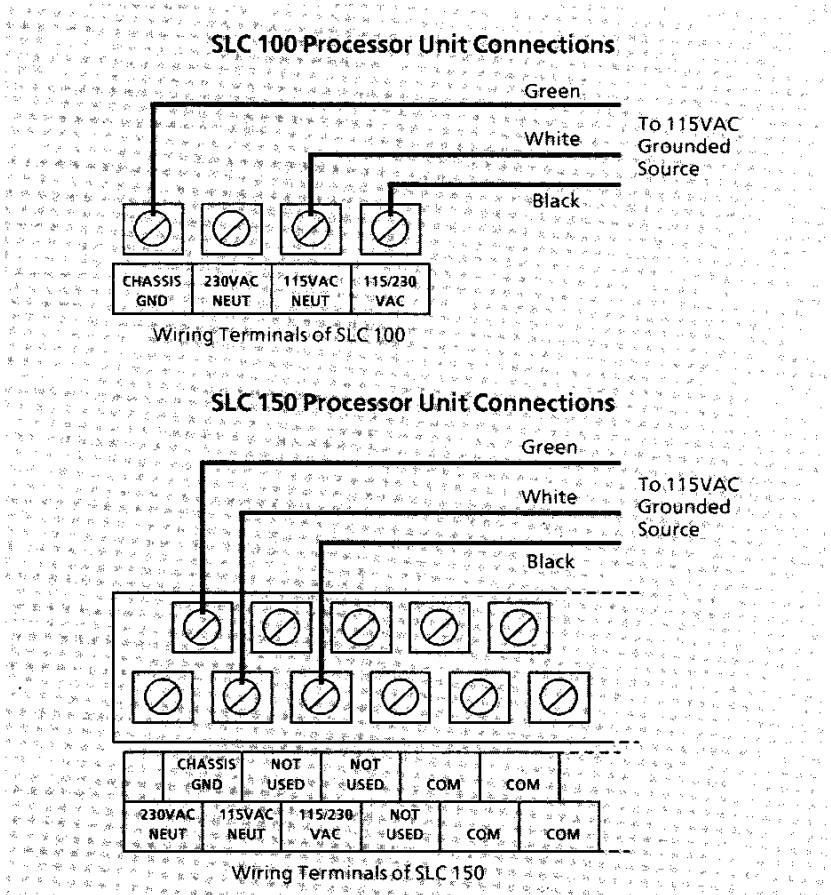


SLC 100 processor unit with the pocket programmer plugged in. Connect a power cord to the processor as indicated in Figure 4.5.

Practice Setup *(continued)*

The figure below shows the procedures for connecting an SLC 100 processor unit or an SLC 150 processor unit to a 115VAC power source.

Figure 4.5



Connection Procedure – 115VAC

1. Plug the programmer cable into the processor unit (see Figure 4.4). Cable connector is keyed to guard against improper insertion. Make sure the spring latch is engaged to secure the cable.
2. For 115VAC power connection, we suggest you use a flexible power cord equipped with a 3-prong plug. Connect the wires as shown above.

> **CAUTION:** Incorrect wire connections can cause damage to the processor unit power supply. Do not jumper 115VAC NEUT and 230VAC NEUT together. Do not jumper unused 115VAC NEUT or unused 230VAC NEUT to the CHASSIS GND terminal.

3. Plug the cord into a grounded 115 VAC outlet. If the outlet is not controlled by a nearby switch, power can be applied and removed by plugging and unplugging the power cord.

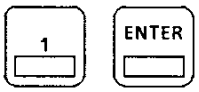
> **WARNING:** Contact with the source voltage can cause serious injury. Do not apply power until the wiring is completed and checked.

Practice Setup *(continued)*

When you apply power to the processor unit, the DC ON indicator should light, indicating that DC voltage is being supplied to the internal logic circuitry.

When you turn the pocket programmer switch on, the programmer display will show "SLC-100" and automatically go thru a series of diagnostic checks; it then displays the operating mode it was left in when last used. (If the programmer displays an error, follow the procedure described in "New Processors" at the bottom of this page.)

To be certain that you are beginning with the processor memory cleared, enter the Clear Memory mode. To do this, press



The display will show

SurE ?

It is asking whether you are SURE you want to clear the memory.

Press the ENTER key again. The display will briefly show

donE

indicating the memory is cleared. It will then show

885 End if you are using an SLC 100 processor or 1200 End if you are using an SLC 150 processor. This indicates that you are at the *end* of the program and there are 885 (or 1200) words *remaining* in memory. Since the memory capacity is 885 (or 1200) words, this also tells you that the program length is zero words.

Note that the LED labeled PROG is lit. This indicates you are in the Program mode.

If the controller is protected with an access code: When you turn the pocket programmer switch on, the programmer will go thru the diagnostic checks, then display

___ PAS

Enter the code and press ENTER. The processor will then enter the mode it was left in when last used.

New Processors

If this is the first time the processor unit is being used since it left the factory, the programmer will detect a series of errors in memory. Clear each error by pressing the CANCEL CMD key after the error is displayed.

After you clear the errors, you must clear the processor memory. Press MODE, 1, ENTER, ENTER. This clears the memory and places the processor in the Program mode.

Using an Access Code (Password)

You can restrict access to the processor memory by entering a 3-digit access code or "password". To do this, enter mode 8 (Enter/Change Access Code) by pressing MODE, 8, ENTER. Then enter any 3-digit number from 001 to 999 except 356, 712, or 940. Finally, press ENTER. The access code is now stored in the user program.

Each time you use the pocket programmer (*any* pocket programmer), you will be prompted for the access code with the display "PAS"; you must enter the correct code and press ENTER before the processor will enter the mode it was left in when last used.

The access code will also protect the contents of an EEPROM memory module. If an access code has been stored in an EEPROM, you will be required to enter the code before the processor will execute this program (Chapter 17).

If you want to change your code, enter your code as you normally would, then re-enter mode 8 and press the new code and ENTER when prompted for a code.

If you no longer want the processor to be protected by a code, re-enter mode 8 and press 0, ENTER when prompted for a code.

An access code can be overriden. To do this, press 9, 4, 0, ENTER, INSERT when prompted for a code. Important: The processor will no longer be protected by the access code. It must be re-entered.

Mode Selection

Operating modes are described in detail on Page 4-3. You can select a particular mode as follows:

1. Press the MODE key. The programmer will display the symbol and number of the current mode, for example
   prog 2
2. Press the numerical key corresponding to the particular mode you want. Your mode will be shown on the display.

   Alternately, you can press the NEXT or LAST key repeatedly until your mode appears on the display.
3. Press the ENTER key. If you selected mode 2, 3, 4, 5, or 9, you are now in that mode.

   If you selected mode 1, press ENTER a second time.

   Entering modes 6 and 7 are discussed in detail in Chapter 17.

   Entering mode 8 is discussed in the paragraph above, "Using an Access Code (Password)".

Note: When you enter the Run mode (3), outputs in your program are enabled. When you change from the Run mode to any other mode, outputs will be disabled.

Entering a Simple Program

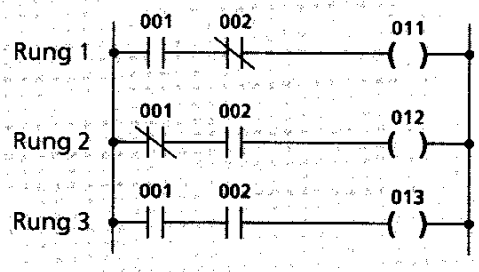
The ladder diagram of your program is your "road map" in entering and reviewing your program. It is important to check the diagram for accuracy before you enter it into memory. Number the rungs.

A simple 3-rung ladder diagram is shown below. External I/O addresses 001 and 002 are assigned to the conditional instructions (Examine ON, Examine OFF) in each of the three rungs. External I/O addresses 011, 012, and 013 are assigned to the three Output Energize instructions.

The step-by-step programming procedure indicates which keys you should press, and what you should observe in the programmer display window after pressing the keys. Note that the PROG LED is illuminated continuously while you are in the Program mode.

Figure 4.6

Keystroke Example – Entering a simple program:



-] [-	-] [-	_ _ _ 1	When you press the Examine ON key, the corresponding LED illuminates. The dashes are prompting you for an address. The "1" indicates you are in rung 1.
1	-] [-	_ _ 1 1	Address 001 is entered by pressing key 1. You do not enter the leading zeros of an address.
-]/[-	-]/[-	_ _ _ 1	The Examine OFF key is pressed. Dashes are prompting you for an address.
2	-]/[-	_ _ 2 1	The address has been entered. If you enter an incorrect address, you can delete it by pressing CANCEL CMD and then enter the correct address.
-( )-	-( )-	_ _ _ 1	The Output Energize key is pressed. Dashes are prompting you for an address.
1, 1	-( )-	_ 1 1 1	The address has been entered. This is the final instruction of rung 1.
ENTER		882 End	The rung is entered into memory by pressing the ENTER key. The display indicates that 882 words remain in memory.
Keys for Rungs 2, 3		876 End	Rungs 2 and 3 of the program are very similar to rung 1. Enter these rungs. At the end of rung 3 the display should indicate 876 words remain in memory.
NEXT and LAST keys	-] [-	⊢ 1 3	**NEXT and LAST keys:** You can review your program with the NEXT and LAST keys. Each time you press the NEXT or LAST key, the cursor moves to the following or previous instruction in the program, indicating its address and the rung number. The first instruction of each rung is indicated by the rung start symbol: ⊢ When you reach the first instruction of rung 3, the display at the left appears.

Display Symbols

To guide you thru programming and monitoring procedures, various abbreviations and symbols appear in the Address-Data and Rung Err-Mode display areas of the pocket programmer. These abbreviations and symbols are explained in Figure 4.7.

Figure 4.7

A	Accumulated (AC) value, sequencers	PASS	Mode 8, Enter/Change Access Code
Ac	Accumulated (AC) value, timers/counters	Pr	Preset (PR) value, timers and counters
bnd	Branch close	prog	Mode 2, Program
bro	Branch open	r	Shift right register
brP	Breakpoint instruction	r	Rung
C	Time-driven (Clock), sequencers	rAc	Reset Accumulated (RAC) value
C-E	Select time- or event-driven, sequencers	rE	Remove
CLEr	Mode 1, Clear Memory	rE F	Remove force
CScn	Mode 5, Test-Continuous Scan	rE r	Remove rung
d	Step data, sequencers	rE P	Remove faulty branch error
d IAG	Mode 9, Diagnostic Test - Programmer	rEAd	Mode 7, Load Program from EEPROM
donE	Done processing	run	Mode 3, Run
E	Event-driven, sequencers, or Error	rung	Rung
-EL-	Enter logic element	SAUE	Mode 6, Store Program in EEPROM
End	End	SLC-100	Start-up display
F	Forced I/O	Srch	Search
FAIL	Failed internal test or password	SScn	Mode 4, Test-Single Scan
Frn	Firmware revision number	StP	Prompt for step number
grP	Group number, sequencers	SurE P	Sure you want to continue?
in r	Insert rung	USE	Mask data, sequencers
in	Insert	ZCL	Zone Control Last State
L	Shift left register	•	Shift key in effect
Mcr	Master Control Reset	⊢	Start of rung
nF	Not found	-- / --- / ----	Prompt for data
P	Preset (PR) value, sequencers		

Error Codes –SLC 100 and SLC 150

The error code numbers which may appear on the programmer display are defined in Figure 4.8. This information also appears on the back of the pocket programmer.

Internal processor errors (codes 1-9) are those occurring in the processor circuitry or the memory. When any of these errors occur, the "CPU Fault" LED on the processor unit will light, and all outputs will be disabled.

Error codes 10-59 identify communication, expansion unit, EEPROM, and programmer hardware problems (code 27 applies to SLC 150 only).

Programming errors (codes 60-67) are user errors indicating that you have used an invalid procedure, entered incorrect or incomplete data, etc. In most cases, the errors are easily understood and remedied; occasionally, it may be necessary to review the proper keystroke sequence for the particular programming operation you are attempting.

The procedure for correcting programming errors is to 1) read the error description, 2) press the CANCEL CMD key, which returns the display to the point it was at before the error was made, then 3) take corrective action as determined by the nature of the error.

Figure 4.8

ERROR CODE	DESCRIPTION
	INTERNAL PROCESSOR ERRORS
1 thru 4	Processor hardware problem
5 thru 8	Processor memory problem
9	Processor scan time problem
10 thru 25	Communication problem
26	Expansion unit problem
27	SLC 150 only. See Figure 4.9
51	EEPROM module problem
52	CAUTION – Program changes not saved in EEPROM module
53 thru 59	Programmer hardware problem
	PROGRAMMING ERRORS
60	Incomplete rung (no output instruction)
61	Invalid address for instruction
62	User memory exceeded
63	Instruction cannot be forced
64	Branch error (short circuit)
65	Branch error (incomplete branch)
66	Invalid access code
67	Rung too large

Error Codes –SLC 150

Error codes 27-1, 27-2, and 27-5 are detected by the SLC 150 processor only. If any of these error conditions occur, the pocket programmer will initially display error code 27. You must then press the CANCEL CMD key to display the -1, -2, or -5 suffix.

Figure 4.9

ERROR CODE	DESCRIPTION
27-1	High speed input module problem
27-2	Zero-cross turn-on failure
27-5	ZCL programming error

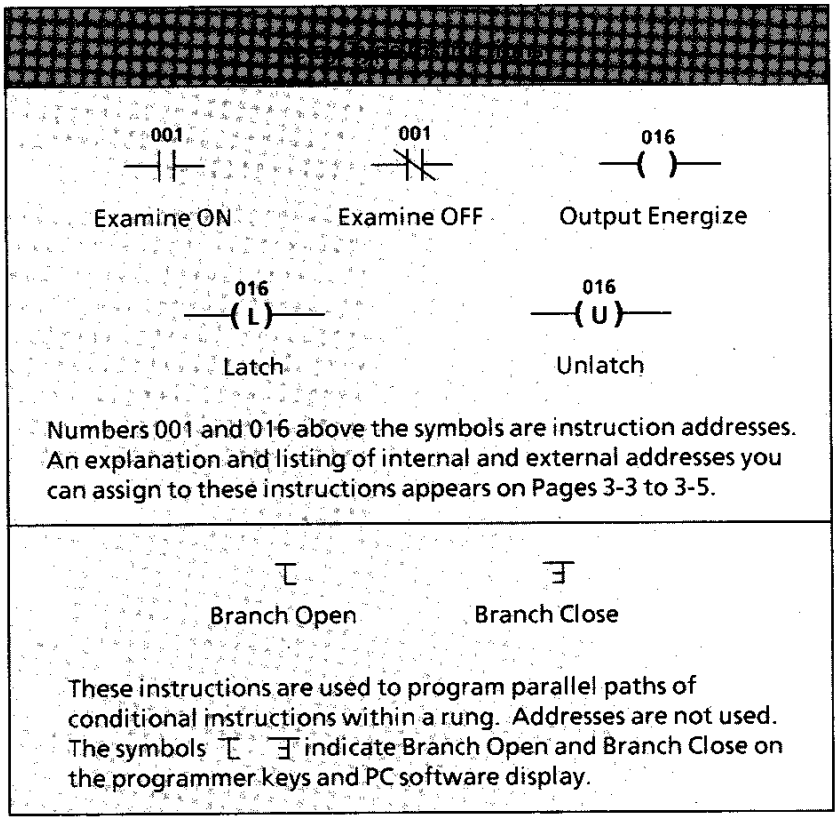
B The Instruction Set, Programming Techniques

Chapter 5 Relay Type Instructions

General

Relay type instructions include Examine ON, Examine OFF, and Output Energize instructions, already discussed in Chapters 3 and 4. This chapter goes into further detail and also introduces you to the Latch and Unlatch instructions (retentive output) and the Branch Open and Branch Close instructions.

The figure below summarizes relay type instructions.



Examine ON, Examine OFF, Output Energize

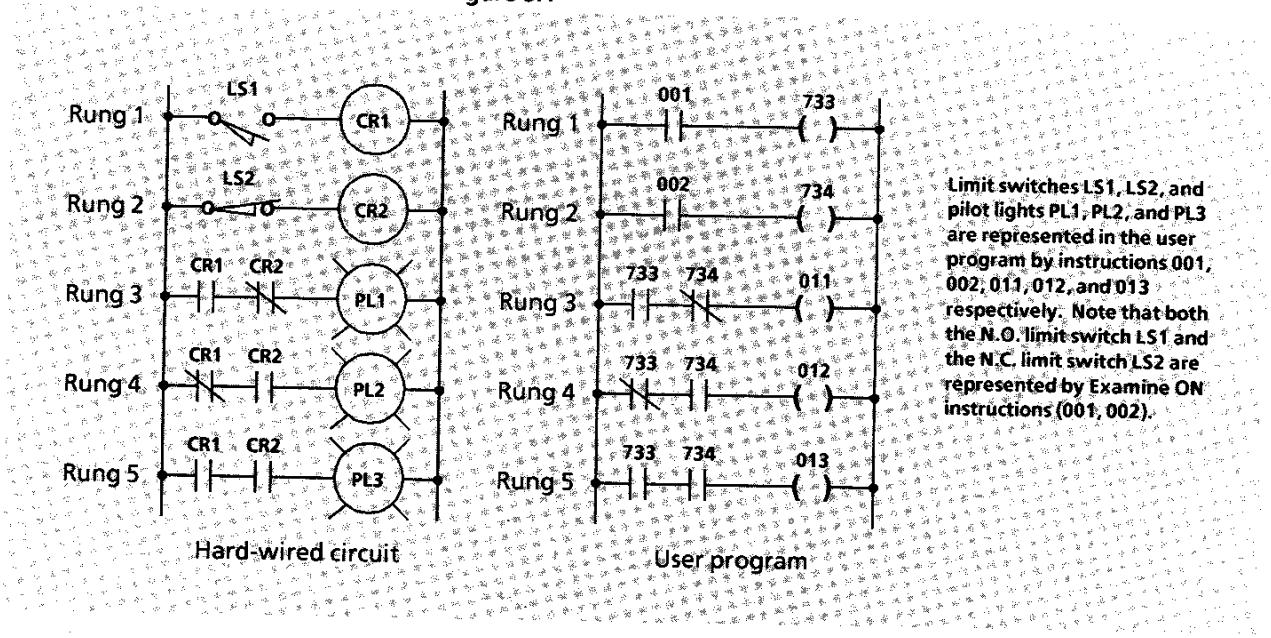
These instructions were already explained (Pages 3-6, 3-7). A user program applying them is shown in Figure 5.1. This figure shows a hard-wired circuit and a user program which provides the same results.

Programmed instructions with addresses 733 and 734 are *internal*. They do not represent external I/O devices. On the other hand, instructions 001, 002, 011, 012, and 013 represent limit switches LS1, LS2, and pilot lights PL1, PL2, PL3.

You will note similarities between the hard-wired circuit and the user program. Thus, when relay coil CR1 in the hard-wired circuit is energized, its normally open and normally closed contacts in rungs 3, 4, and 5 operate, or change state. Similarly, when Output Energize instruction 733 in the user program goes TRUE, its associated Examine ON and Examine OFF instructions in rungs 3, 4, and 5 change their TRUE/FALSE state.

In comparing the diagrams, you will also discover an apparent contradiction. That is, normally open limit switch LS1 and normally closed limit switch LS2 are *both* represented in the program by an Examine ON instruction. This makes sense when you consider that rungs 1 and 2 must accomplish the same thing: The rung must be TRUE when the external limit switch is closed, and FALSE when the limit switch is open. Using an Examine ON instruction to represent the limit switch satisfies these requirements. The N.O./N.C. mechanical action of the switch is not a consideration.

Figure 5.1

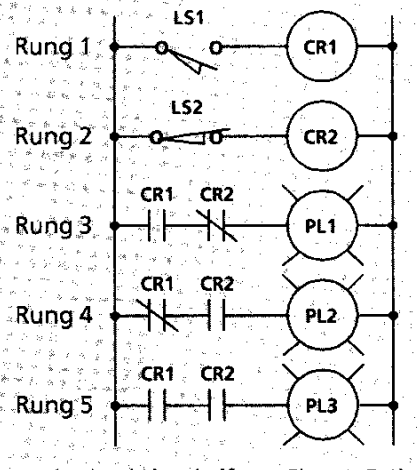


Equivalent User Programs

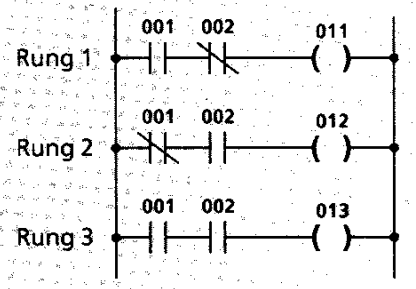
The function of the hard-wired circuit in Figure 5.1 can be achieved with another, simpler program–the 3-rung ladder diagram in Figure 5.2. Again, instructions 001 and 002 represent limit switches LS1 and LS2; instructions 011, 012, and 013 represent pilot lights PL1, PL2, and PL3.

The ladder diagram of Figure 5.2 does not use internal instructions to represent hard-wired relays CR1 and CR2. Instead, instruction addresses 001 and 002 are used repeatedly to obtain the logical continuity equivalent to the electrical continuity of rungs 3, 4, and 5 of the hard-wired circuit.

Figure 5.2



Hard-wired circuit (from Figure 5.1)



Equivalent 3-rung user program. (Compare this with the user program in Figure 5.1.)

Unconditional Output Energize

It is possible to enter a rung consisting of an Output Energize instruction only. Since the rung lacks a condition instruction, the output will be continuously TRUE. Although the unconditional Output Energize has a few specialized applications, it is generally an undesirable programming procedure which you should be careful to avoid.

Branch Instructions

These are used to create parallel paths of condition instructions, allowing more than one set of conditions (OR logic) to establish continuity in a rung. Figure 5.3 illustrates a branching condition. The rung will be TRUE if *either* instruction 001 *or* instruction 002 is TRUE.

To program branch instructions with the pocket programmer:

1) Press the Branch Open key before entering the first instruction of each parallel path.
2) Press the Branch Close key after entering the last instruction of the last branch.

The ladder diagram of Figure 5.3 illustrates the use of branch instructions. To program this rung, press the following keys:

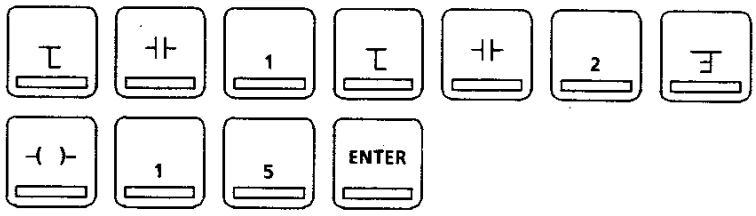
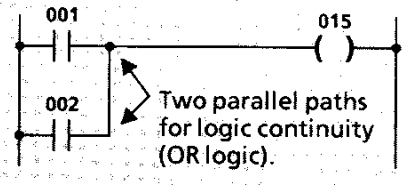


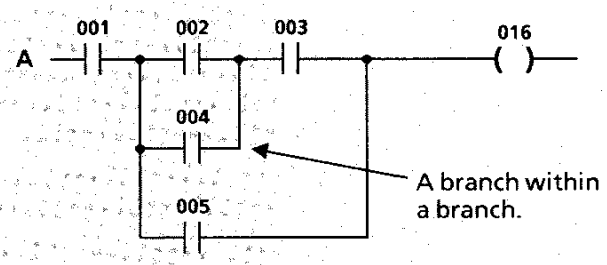
Figure 5.3



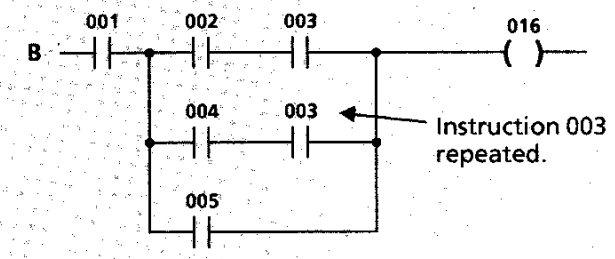
The Nested Branch

A nested branch (branch within a branch) cannot be programmed, but you can program an equivalent branching condition. This is shown in the figure below.

Figure 5.4



Rung **A** above *cannot* be programmed.



Rung **B**, equivalent to rung **A**, *can* be programmed.

Output Latch, Output Unlatch

These retentive instructions are used as a pair at the same address.

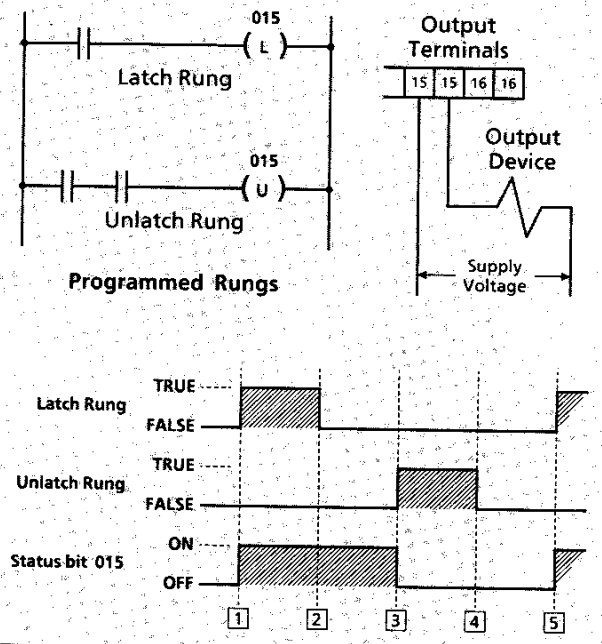
When a rung containing an Output Latch instruction goes TRUE, the Output Latch status bit is set to ON. The bit can only be turned OFF by an Output Unlatch instruction located in a separate rung. This is summarized in Figure 5.5.

The Latch/Unlatch instruction status is retained when you change from the Run mode to another mode. Thus, in the example of Figure 5.5, if you change from the Run mode to any other mode while status bit 015 is ON, the external output device will be de-energized, but the status bit will remain ON; when you return to the Run mode, the output device will again be energized.

The Latch/Unlatch instruction status is also retained on power loss, since the processor memory is provided with battery back-up.

Operation after CPU Faults: Latch instructions are reset to their initial conditions after a CPU fault is detected (error codes 1 thru 9, 26, and 51).

Figure 5.5



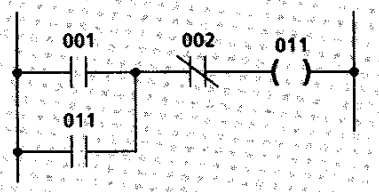
1. Latch rung is TRUE; Status bit 015 is ON; Output device is energized.
2. Latch rung goes FALSE; Status bit 015 remains ON; Output device remains energized.
3. Unlatch rung goes TRUE; Status bit 015 goes OFF; Output device is de-energized.
4. Unlatch rung goes FALSE; Status bit 015 remains OFF; Output device remains de-energized.
5. Latch rung goes TRUE; Status bit 015 goes ON; Output device is energized.

Timing Diagram

Hold-In Rungs – Operating Differences Series A vs Series B

Figure 5.6 shows a typical hold-in rung, analogous to the familiar electromechanical start-stop hold-in circuit. All versions of SLC 100 and SLC 150 processor units will function as described. However, there are two conditions under which there are operating differences. This is pointed out in the table.

Figure 5.6



A typical hold-in rung

All processor units:

When Examine ON instruction 001 goes TRUE, output instruction 011 goes TRUE.

When instruction 001 subsequently goes FALSE, output 011 remains TRUE, being "held-in" by Examine ON instruction 011.

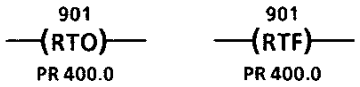
Output 011 will go FALSE when Examine OFF instruction 002 goes FALSE.

Event	Status of Non-Retentive Outputs in Hold-In Rungs	
	Series A SLC 150	Series B SLC 150 and Series A, B, C SLC 100
Power is cycled from Off to On.	Outputs return to their last state.	Outputs are Off.
Processor is switched from the Run mode to any other operating mode, then switched back to the Run mode.	Outputs return to their last state. However, if the Program mode was entered and a program change was made, outputs will be Off.	Outputs are Off.

Chapter 6 Timer Instructions

RTO and RTF Timer Characteristics

Timer instructions include the Retentive Timer On-Delay –(RTO)– and the Retentive Timer Off-Delay –(RTF)–. Both require the use of the Reset instruction –(RST)– . Timer instructions are represented by the following symbols:



Address: 901-932. Time base: 0.1 second. Range: 000.1 to 999.9 seconds.

Timers with long or fine time base: Special instructions can be used for time delays beyond 999.9 seconds and for timers with greater resolution than 0.1 second. Refer to Chapter 8.

AC, PR, and RAC Values

The timer instruction functions as an internal "clock", counting 0.1-second intervals. The number of intervals counted is called the *Accumulated value* (AC). Counting takes place under these TRUE/FALSE rung conditions:

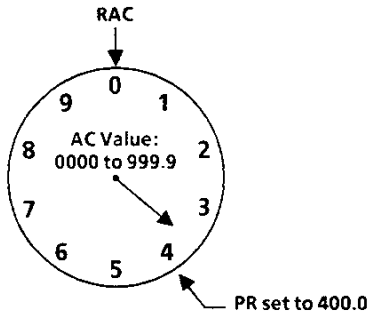
RTO Timer Rung Conditions		
TRUE	**FALSE**	**TRUE**
Timer is counting.	Counting stops. AC value retained.	Counting resumes.
AC value represents the cumulative time during which rung is TRUE.		

RTF Timer Rung Conditions		
FALSE	**TRUE**	**FALSE**
Timer is counting.	Counting stops. AC value retained.	Counting resumes.
AC value represents the cumulative time during which rung is FALSE.		

The clock figure below represents the AC value. The time delay is set by programming a *Preset value* (PR). In this case, the PR value is set at 400.0 seconds.

The figure also shows a *Reset Accumulated value* (RAC) at 0000. This is automatically set when you program the reset instruction. In most cases, RAC is left at 0000, but you can change it to any value up to 9999. If you program an RAC value, the time delay will equal PR minus RAC.

You will find that the programmer display does not show a decimal point for RAC values. Thus, if you program PR to 30.8 and RAC to 105, the time delay will be 30.8 minus 10.5 = 20.3 seconds.



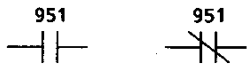
Status and Overflow Bits

The timer *status bit* has the same address as the timer instruction:

The RTO status bit is set ON when the AC value reaches the PR value. Examine ON instructions at the status bit address go TRUE; Examine OFF instructions go FALSE.

The RTF status bit is set ON instantaneously when the RTF instruction goes TRUE, and is set OFF when the AC value reaches the PR value. Examine ON instructions at the status bit address go FALSE; Examine OFF instructions go TRUE.

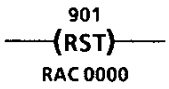
The *overflow bit* is given the timer address plus 50. (901 + 50 = 951):



The overflow bit is set to ON when the AC value "overflows" from 999.9 to 0000. Examine ON instructions at the overflow bit address go TRUE; Examine OFF instructions go FALSE. RTO and RTF status bits are not affected when the AC value overflows.

Reset Instruction

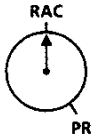
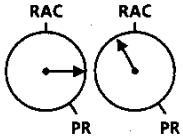
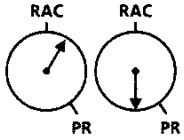
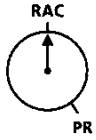
The reset (RST) instruction is given the same address as the timer instruction:



When the RST instruction goes TRUE, status bits and overflow bits are set to OFF, and the AC value is reset to the RAC value. The RST instruction must go FALSE again before the timer can resume counting.

Typical Timer Operation

The table below indicates how the status and overflow bits of timers change state as the AC value increments from 1) the reset state (AC = RAC), 2) up to and past the PR value, 3) beyond a value of 999.9 (overflow), and 4) back to the reset state.

Status Bit 901 / Overflow Bit 951 (Examine ON Instructions)	1. Timer Reset AC = RAC	2. RTO Rung TRUE or RTF Rung FALSE. AC Value Increments		3. AC Value Overflows Beyond 999.9		4. RST Rung TRUE. Timer Reset
						
	AC<PR	AC<PR	AC ≥ PR	AC<PR	AC ≥ PR	AC<PR
RTO Status Bit	OFF	OFF	ON	ON	ON	OFF
RTF Status Bit	OFF	ON	OFF	OFF	OFF	OFF
Overflow Bits	OFF	OFF	OFF	ON	ON	OFF

Power Down: The timer status is retentive. When you apply power after a power-down, the AC value and ON/OFF states of status and overflow bits will be the same as before the power-down. The status is also retained when going from the Run mode to another mode.

Operation after CPU Faults: Timer instructions and associated status bits and data values are reset to their initial conditions after a CPU fault is detected (error codes 1 thru 9, 26, and 51).

Monitoring and Changing Data: PR and AC values can be monitored and changed in the Run and Test modes. The PR value can be protected from changes in the Run and Test modes by using the UNPRT/PRT key. Refer to Page 15-9.

Retentive Timer On-Delay (RTO)

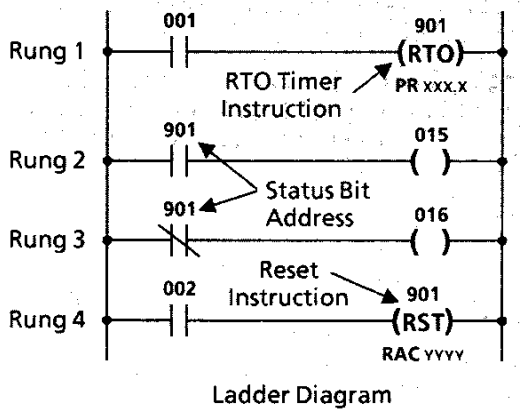
As pointed out on Page 6-1, the AC value of the RTO timer increments when the RTO rung is TRUE. This is illustrated in the figure below, which shows an RTO timer ladder diagram and timing diagram.

For illustrative purposes, both Examine ON and Examine OFF instructions are used at the timer status bit address, 901. These instructions are used to energize output 015 and de-energize output 016 when the AC value reaches the programmed PR value.

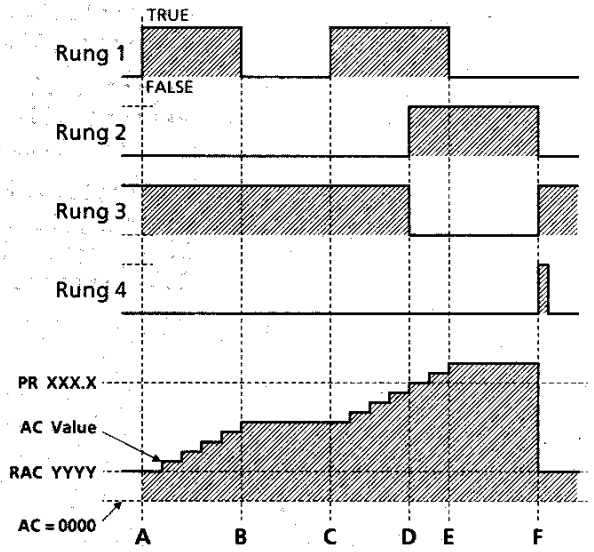
The timing diagram illustrates the retentive nature of the timer. In effect, the timer measures the cumulative periods during which the timer rung (1) is TRUE. The AC value will continue to increment during TRUE rung periods as long as the reset rung (4) remains FALSE. On the count after 999.9, an overflow bit (not used here) is set ON and the AC value continues to increment from 0000.

After the timer is reset and rung 1 goes TRUE again, the timing period will begin at AC = RAC. In most applications, the RAC value would be 0000. We chose to show a non-zero value for illustrative purposes.

Figure 6.1



Ladder Diagram



Timing Diagram

Letters A to F at the bottom of the timing diagram indicate the following events:

A. Rung 1 goes TRUE. The AC value increments, beginning at AC = RAC. Examine ON instruction 901 in rung 2 is FALSE. Examine OFF instruction 901 in rung 3 is TRUE.

B. Rung 1 goes FALSE. The AC value stops incrementing, but the existing value is retained.

C. Rung 1 goes TRUE. The AC value continues to increment from the point it left off.

D. The AC value has reached the PR value. Examine ON instruction 901 goes TRUE, making rung 2 TRUE. Examine OFF instruction 901 goes FALSE, making rung 3 FALSE. The AC value continues to increment.

E. Rung 1 goes FALSE. The AC value stops incrementing, but the existing value is retained. Rung 2 remains TRUE. Rung 3 remains FALSE.

F. Rung 4 goes TRUE. The AC value is reset to the programmed RAC value. Examine ON instruction 901 goes FALSE, making rung 2 FALSE. Examine OFF instruction 901 goes TRUE, making rung 3 TRUE.

While the RST instruction is TRUE, the timer is disabled. Timing can resume when rung 4 goes FALSE again.

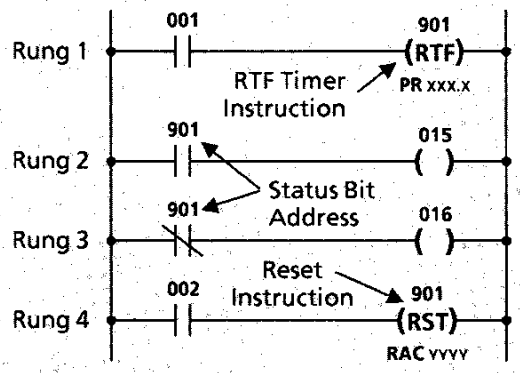
Retentive Timer Off-Delay (RTF)

The essential difference between the RTO and RTF timers is that with the RTF timer, the AC value increments when the timer rung goes FALSE rather than TRUE.

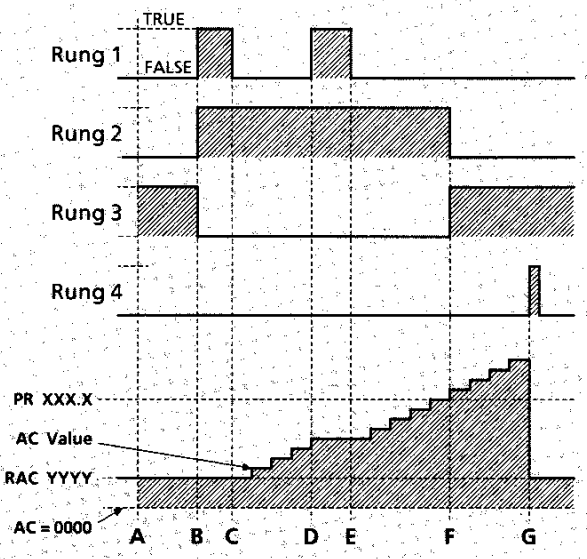
Figure 6.2 shows an RTF timer ladder diagram and the corresponding timing diagram. For illustrative purposes, both Examine ON and Examine OFF instructions are used at the timer status bit address, 901. These instructions are used to de-energize output 015 and energize output 016 when the AC value reaches the programmed PR value.

Here again, the timing diagram illustrates the retentive nature of the timer. After rung 1 makes a TRUE-FALSE transition, the timer measures the cumulative periods during which rung 1 is FALSE. The AC value will continue to increment during FALSE rung periods as long as the reset rung (4) remains FALSE. On the count after 999.9, an overflow bit (not used here) is set ON and the AC value continues to increment from 0000.

Figure 6.2



Ladder Diagram



Timing Diagram

Letters A to G at the bottom of the timing diagram indicate the following events:

A. The timer is reset, with AC = RAC. Rung 1 is FALSE. Examine ON instruction 901 in rung 2 is FALSE. Examine OFF instruction 901 in rung 3 is TRUE.

B. Rung 1 goes TRUE. Instantaneously, Examine ON instruction 901 goes TRUE and Examine OFF instruction 901 goes FALSE, making rung 2 TRUE and rung 3 FALSE.

C. Rung 1 goes FALSE. The AC value begins to increment.

D. Rung 1 goes TRUE. The AC value stops incrementing but the existing value is retained.

E. Rung 1 goes FALSE. The AC value continues to increment from the point it left off.

F. The AC value has reached the PR value. Examine ON instruction 901 goes FALSE, making rung 2 FALSE. Examine OFF instruction 901 goes TRUE, making rung 3 TRUE. The AC value continues to increment.

G. Rung 4 goes TRUE. The AC value is reset to the RAC value. Note that this is the same situation as event A, with the Examine ON instruction 901 FALSE and the Examine OFF instruction 901 TRUE.

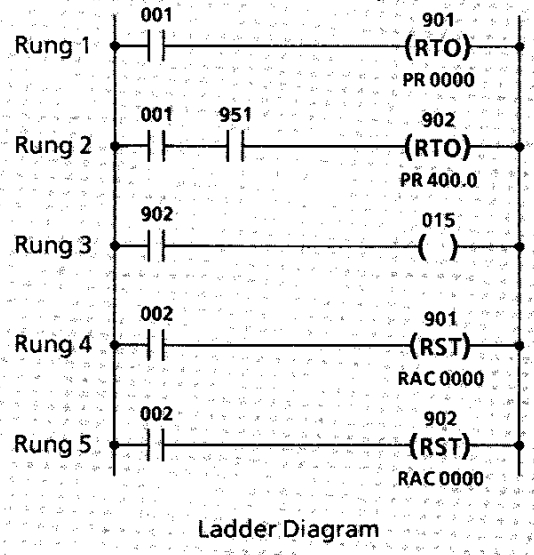
While the RST instruction is TRUE, the timer is disabled. After the RST instruction goes FALSE and rung 1 makes a TRUE to FALSE transition, the timing period will begin at AC = RAC.

Timing Beyond 999.9 Seconds

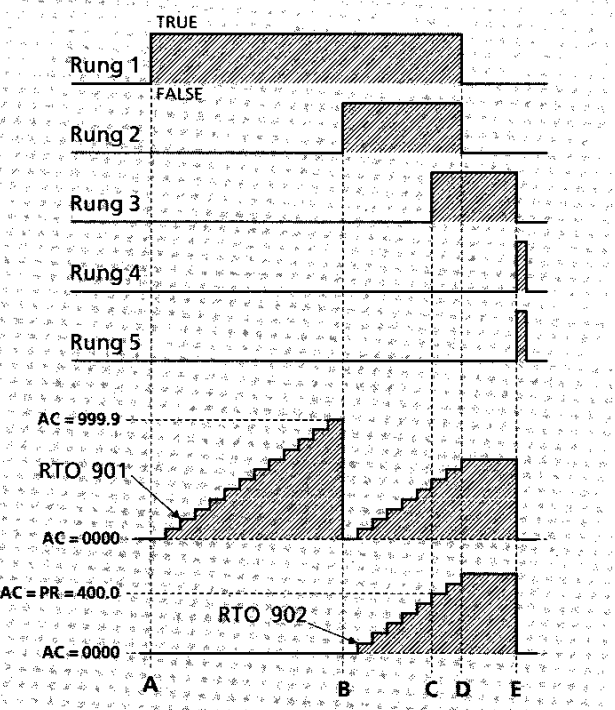
Special instructions at addresses 874 and 875 have time bases of 0.5 and 1.0 seconds, allowing you to program the equivalent of an RTO timer with time delays beyond 999.9 seconds. This is discussed in Chapter 8. You can also use the instructions discussed in this chapter to obtain time delays beyond 999.9 seconds. An example is shown below, where the overflow bit of one RTO timer initiates a second RTO timer to produce an output at 1400 seconds.

Timer 901 in rung 1 is used for the first part of the timing period and timer 902 in rung 2 is used for the last part. The Examine ON instruction 001 of rung 1 is repeated in rung 2, so that if timing is interrupted during the last part of the timing period, the AC value will stop incrementing, preventing an output in rung 3.

Figure 6.3



Ladder Diagram



Timing Diagram

Letters A to E at the bottom of the timing diagram indicate the following events:

A. Examine ON instructions 001 in rungs 1 and 2 go TRUE. Rung 1 goes TRUE. AC value of timer 901 increments from 0000.

B. AC value of timer 901 overflows beyond 999.9, setting the overflow bit (address 951) ON. Examine ON instruction 951 in rung 2 goes TRUE, making rung 2 TRUE. AC value of timer 902 increments from 0000.

1000 seconds have elapsed.

C. AC value of timer 902 reaches programmed PR value of 400.0, setting the status bit (address 902) ON. Examine ON instruction 902 in rung 3 goes TRUE, making rung 3 TRUE.

Total time elapsed: 1000 + 400 = 1400 seconds.

D. Instructions 001 in rungs 1 and 2 go FALSE, making rungs 1 and 2 FALSE. Rung 3 remains TRUE.

E. Reset rungs 4 and 5 go TRUE. Examine ON instructions 902 and 951 go FALSE. AC values are reset to 0000.

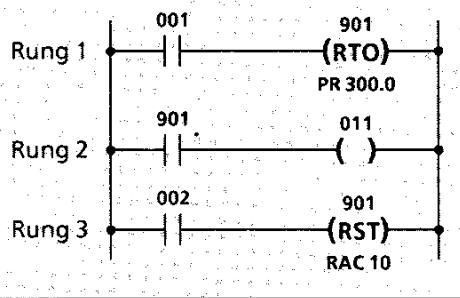
While the RST instructions are TRUE, the timer is disabled. Timing can resume when rungs 4 and 5 go FALSE again.

Timer Keystroke Example

Program this basic RTO timer. After you've programmed it, you may want to use the NEXT and LAST keys to review your work.

Figure 6.4

Keystroke Example – RTO Instruction



-] [-	-] [-	_ _ _ 1	Prompt for an address.
1	-] [-	_ _ 1 1	Address 001 is entered.
-(RTO)-	-(RTO)-	_ _ _ 1	Prompt for an address.
9, 0, 1	-(RTO)-	9 0 1 1	Address 901 is entered.
ENTER	-(RTO)-	_ _ _ _ Pr	Prompt for a preset (PR) value.
3, 0, 0, 0	-(RTO)-	3 0 0.0 Pr	PR value of 300.0 seconds is entered.
ENTER		8 8 3 End	This completes rung 1.
-] [-	-] [-	_ _ _ 2	Prompt for an address.
9, 0, 1	-] [-	9 0 1 2	Address 901 is entered.
-( )-	-( )-	_ _ _ 2	Prompt for an address.
1, 1	-( )-	_ 1 1 2	Address 011 is entered.
ENTER		8 8 1 End	This completes rung 2.
-] [-	-] [-	_ _ _ 3	Prompt for an address.
2	-] [-	_ _ 2 3	Address 002 is entered.
SHIFT	-] [-	_ _ 2 3.	The SHIFT key must be pressed before you press the -(RST)- key. The dot to the right of the rung number indicates you have pressed the SHIFT key.
-(RST)-	-(RST)-	_ _ _ 3	Prompt for an address.
9, 0, 1	-(RST)-	9 0 1 3	Address 901 is entered.
ENTER	-(RST)-	_ _ _ 0 rAc	The display indicates that the RAC value is set to zero. You can override this by entering an RAC value.
1, 0	-(RST)-	_ _ 1 0 rAc	The RAC value is specified as 10. **For timers, the actual RAC value is 0.1 of the value displayed** (in this case it is 1.0, not 10). The time delay (PR – RAC) is set at 300.0 – 1.0 = 299.0 seconds.
ENTER		8 7 8 End	End of program.

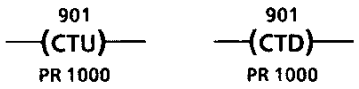
Programming an RTF Timer

Figure 6.4 can be easily converted to an RTF timer keystroke example. Just change the RTO instruction in the ladder diagram to an RTF instruction. Then, instead of pressing the –(RTO)– key in rung 1, press the SHIFT key and the –(RTF)– key.

Chapter 7 Counter Instructions

CTU and CTD Counter Characteristics

Counter instructions include the Up Counter –(CTU)– and the Down Counter –(CTD)–. Both are retentive, requiring the use of the Reset instruction –(RST)– . Counter instructions are represented by the following symbols:



Address: 901-932. Internal. Range: 1 to 9999 counts.

AC, PR, and RAC Values

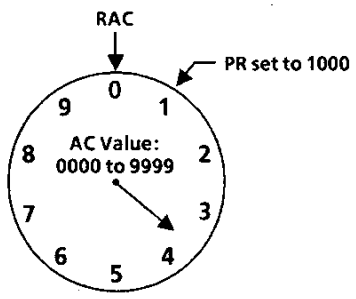
Counter instructions count successive FALSE-to-TRUE transitions of the rung containing the counter instruction. After each count, the rung must return to FALSE before another count can take place.

The existing count is called the *Accumulated value* (AC). For up counters, the AC value increases by 1 for each FALSE-TRUE transition. For down counters, the AC value decreases by 1 for each FALSE-TRUE transition.

Up-down counters have both an up counter rung and a down counter rung. Accordingly, the AC value both increases and decreases, responding to FALSE-TRUE transitions in both rungs.

The clock figure below represents the AC value. An output can be obtained at a particular count by programming a *Preset value* (PR). In this case, the PR value is set at 1000.

The figure also shows a *Reset Accumulated value* (RAC) at 0000. This is automatically set when you program the reset instruction. You can either leave it at 0000, or change it to any value up to 9999.



Setting a Counter

Up counters can be reset to begin counting at AC = RAC and produce an output at AC = PR. To obtain an output at 1000 counts, just set the PR value to 1000. Or, since RAC can be programmed to any value, you might set the PR value at 1500 and the RAC value at 0500, etc.

Up-down counters add and subtract counts, starting at AC = RAC, and produce an output at AC = PR. Thus, if you set RAC at 500 and PR at 1500, an output will be produced whenever the up counts exceed the down counts by 1000.

Status and Overflow/Underflow Bits

The counter *status bit* has the same address as the counter instruction:

The status bit is set ON when the AC value reaches the PR value. Examine ON instructions at the status bit address go TRUE; Examine OFF instructions go FALSE.

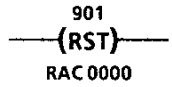
Overflow and *underflow bits* are assigned the counter address plus 50. (901 + 50 = 951):

The overflow bit is set ON when the AC value "overflows" from a count of 9999 to 0000. The underflow bit is set ON when the AC value "underflows" from 0000 to 9999. Examine ON instructions at the overflow/underflow bit address go TRUE; Examine OFF instructions go FALSE.

CTU status bits are not affected when the AC value overflows. CTD status bits *will be set ON* if the AC value underflows making AC > PR.

Reset Instruction

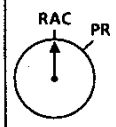
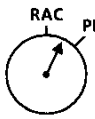
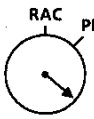
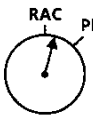
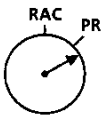
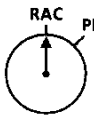
The reset (RST) instruction is given the same address as the counter instruction:



When the RST instruction goes TRUE, the status, overflow, and underflow bits are reset to OFF, and the AC value is reset to the RAC value. The RST instruction must go FALSE again before counting can resume.

Typical Up Counter Operation

The table below indicates how the status and overflow bits of counters change state as the AC value increments from 1) the reset state (AC = RAC), 2) up to and past the PR value, 3) beyond a value of 9999 (overflow), and 4) back to the reset state.

Status Bit 901 ⊣⊢ Overflow Bit 951 ⊣⊢ (Examine ON Instructions)	1. Counter Reset AC = RAC	2. Repeating FALSE-TRUE Rung Transitions. AC Value Increments		3. AC Value Overflows Beyond Count of 9999		4. RST Rung TRUE. Counter Reset.
						
	AC<PR	AC<PR	AC ≥ PR	AC<PR	AC ≥ PR	AC<PR
Status Bit	OFF	OFF	ON	ON	ON	OFF
Overflow Bit	OFF	OFF	OFF	ON	ON	OFF

Power Down: The counter status is retentive. When you apply power after a power-down, the AC value and the ON/OFF states of status and overflow/underflow bits will be the same as before power-down. The status is also retained when going from the Run mode to another mode.

Operation after CPU Faults: Counter instructions and associated status bits and data values are reset to their initial conditions after a CPU fault is detected (error codes 1 thru 9, 26, and 51).

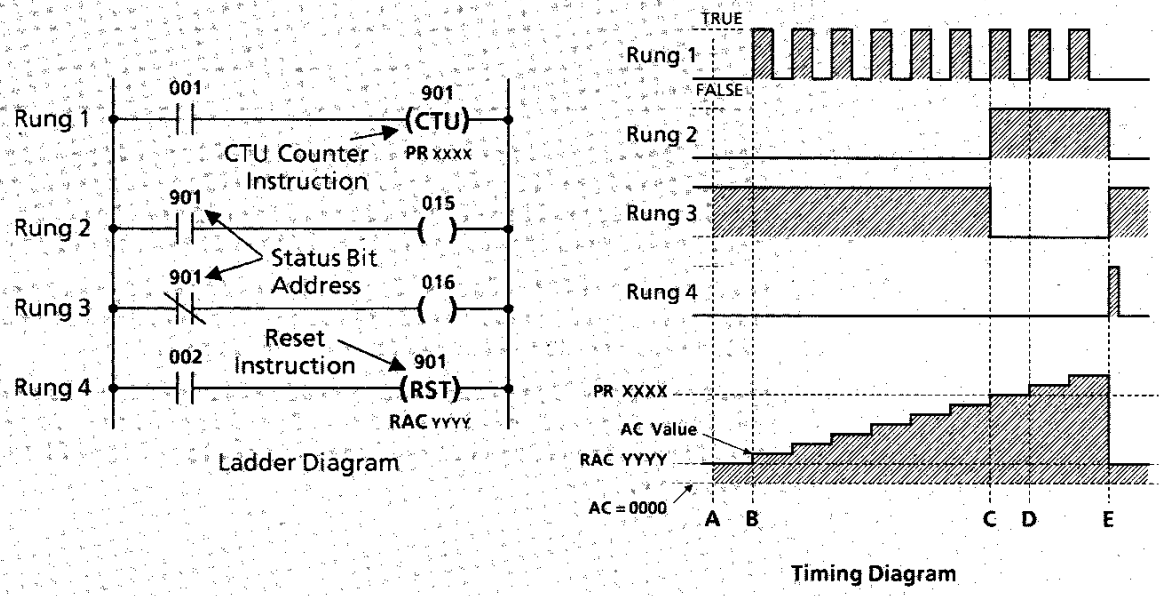
Monitoring and Changing Data: PR and AC values can be monitored and changed in the Run and Test modes. The PR value can be protected from changes in the Run and Test modes by using the UNPRT/PRT key. Refer to Page 15-9.

Up Counter (CTU)

As pointed out on Page 7-1, the AC value of the CTU counter increases by one for each FALSE-to-TRUE transition occurring in the CTU counter rung. This is illustrated in Figure 7.1, which shows a CTU counter ladder diagram and the corresponding timing diagram.

For illustrative purposes, both Examine ON and Examine OFF instructions are used at the counter status bit address, 901. These instructions are used to energize output 015 and de-energize output 016 when the AC value reaches the PR value.

Figure 7.1



Letters A to E at the bottom of the timing diagram indicate the following events:

A. The AC value equals the RAC value. Rung 1 is FALSE. Examine ON instruction 901 in rung 2 is FALSE. Examine OFF instruction 901 in rung 3 is TRUE.

B. Rung 1 goes from FALSE to TRUE, causing the AC value to increase by one. The rung then returns to FALSE. This cycle repeats again and again, with the AC value increasing by one count for each FALSE-TRUE transition.

C. The AC value has reached the programmed PR value. Examine ON instruction 901 goes TRUE, making rung 2 TRUE. Examine OFF instruction 901 goes FALSE, making rung 3 FALSE.

D. The AC value continues to increase for each FALSE-TRUE transition of rung 1. Rung 2 remains TRUE. Rung 3 remains FALSE.

E. Rung 4 goes TRUE. The AC value is reset to the programmed RAC value. Examine ON instruction 901 goes FALSE, making rung 2 FALSE. Examine OFF instruction 901 goes TRUE, making rung 3 TRUE.

While the RST instruction is TRUE, the counter is disabled. Counting can resume when rung 4 goes FALSE again.

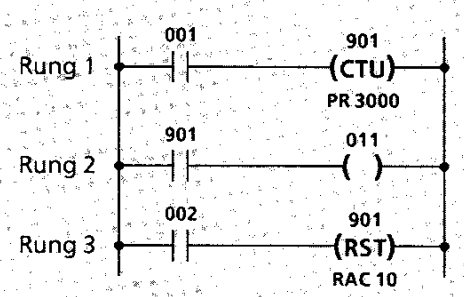
If the counter is not reset as done in event E, the AC value continues to increase for FALSE-TRUE counter rung transitions. On the count after 9999, an overflow bit is set ON and the AC value continues to increment from 0000.

Keystroke Example

Program this basic CTU counter. The programming steps are quite similar to those of the RTO timer example of Figure 6.4.

Figure 7.2

Keystroke Example – CTU Instruction



–] [–	–] [–	_ _ _	Prompt for an address.
1	–] [–	_ _ 1	Address 001 is entered.
–(CTU)–	–(CTU)–	_ _ _	Prompt for an address.
9, 0, 1	–(CTU)–	9 0 1	Address 901 is entered.
ENTER	–(CTU)–	_ _ _ _ Pr	Prompt for a preset (PR) value.
3, 0, 0, 0	–(CTU)–	3 0 0 0 Pr	PR value of 3000 is entered.
ENTER		8 8 3 End	This completes rung 1.
–] [–	–] [–	_ _ _ 2	Prompt for an address.
9, 0, 1	–] [–	9 0 1 2	Address 901 is entered.
–( )–	–( )–	_ _ _ 2	Prompt for an address.
1, 1	–( )–	_ 1 1 2	Address 011 is entered.
ENTER		8 8 1 End	This completes rung 2.
–] [–	–] [–	_ _ _ 3	Prompt for an address.
2	–] [–	_ _ 2 3	Address 002 is entered.
SHIFT	–] [–	_ _ 2 3.	The SHIFT key must be pressed before you press the –(RST)– key. The dot to the right of the rung number indicates you have pressed the SHIFT key.
–(RST)–	–(RST)–	_ _ _ 3	Prompt for an address.
9, 0, 1	–(RST)–	9 0 1 3	Address 901 is entered.
ENTER	–(RST)–	_ _ 0 rAc	The display indicates that the RAC value is set to zero. You can override this by entering an RAC value.
1, 0	–(RST)–	_ 1 0 rAc	The RAC value of 10 is entered. The count (PR – RAC) is 3000 – 10 = 2990.
ENTER		8 7 8 End	End of program.

Up-Down Counter

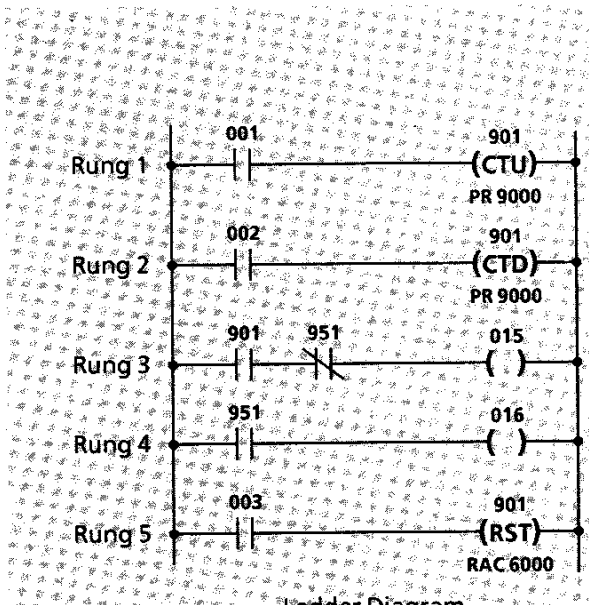
This counter includes a CTU instruction and a CTD instruction which share the same address (including status and overflow/underflow addresses). AC, PR, and RAC values are also shared. This is illustrated in Figure 7.3, which shows an up-down counter programmed to produce two outputs:

The first output is energized when the AC value reaches 9000, then de-energized when the AC value overflows beyond 9999. In terms of counts, this output is energized only when the net count (up counts minus down counts) is within the range of 3000 to 3999.

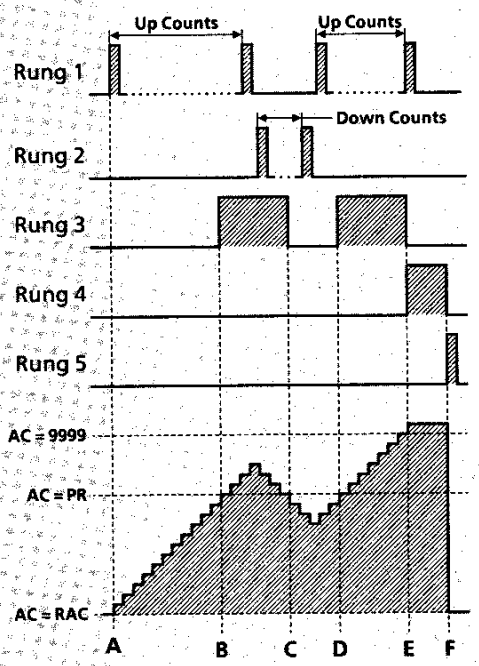
The second output is energized when the AC value overflows beyond 9999, that is, when the net count is 4000 or more.

We are assuming that the AC value does not underflow below 0000.

Figure 7.3



Ladder Diagram



Timing Diagram

Letters A to F at the bottom of the timing diagram indicate the following events:

A. CTU counter in rung 1 begins a series of up counts, beginning at AC = RAC = 6000. Examine ON instruction 901 in rung 3 is FALSE. Examine OFF instruction 951 in rung 3 is TRUE. Examine ON instruction 951 in rung 4 is FALSE.

B. The AC value has reached the programmed PR value of 9000, setting the status bit (address 901) ON. Examine ON instruction 901 in rung 3 goes TRUE, making rung 3 TRUE. The net count is 3000.

C. CTD counter in rung 2 is down counting. The AC value is now less than the PR value of 9000. Examine ON instruction 901 in rung 3 goes FALSE, making rung 3 FALSE. The net count has dropped below 3000.

D. Rung 1 is up counting, and the AC value has reached the PR value of 9000. Examine ON instruction 901 goes TRUE, making rung 3 TRUE. The net count is 3000 again.

E. The AC value overflows beyond 9999, setting the overflow bit (address 951) ON. Examine ON instruction 951 in rung 4 goes TRUE, making rung 4 TRUE. Examine OFF instruction 951 in rung 3 goes FALSE, making rung 3 FALSE. The net count is 4000.

F. Rung 5 goes TRUE. The AC value is reset to the programmed RAC value of 6000. Status bit and overflow bit are reset to OFF. Rung 4 goes FALSE.

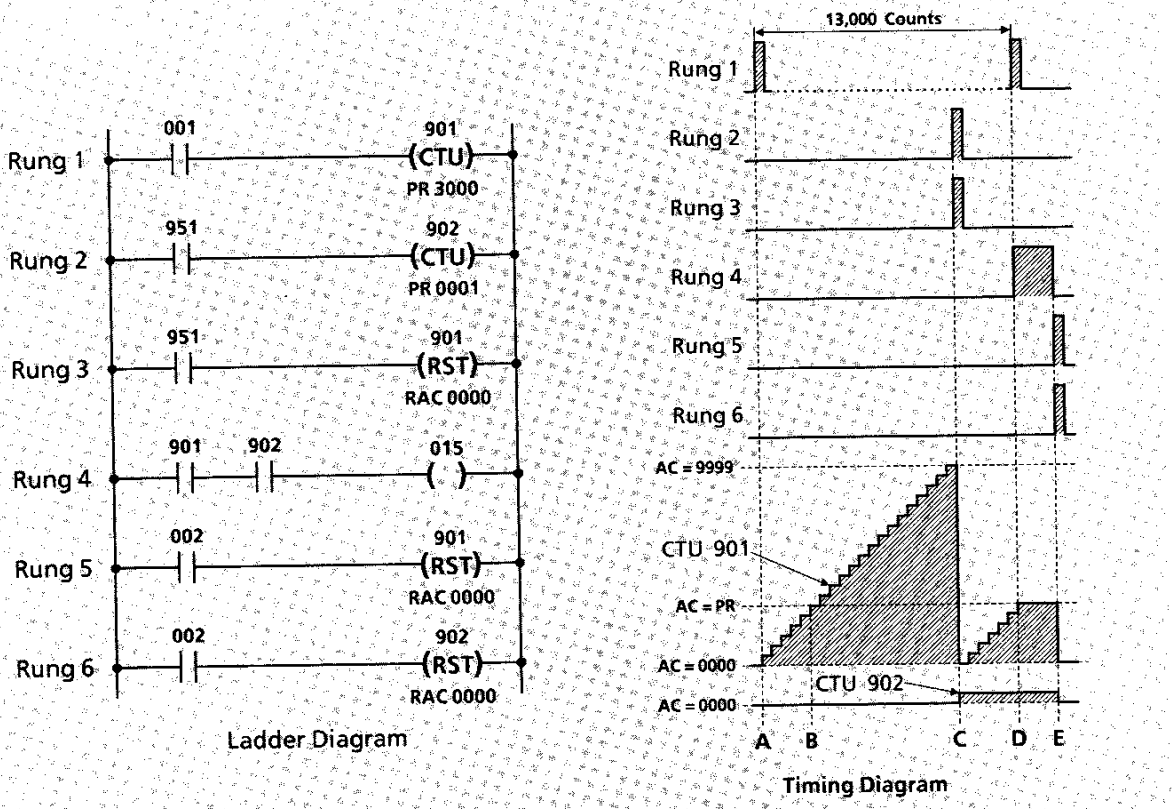
While the RST instruction is TRUE, the counter is disabled. Counting can resume when rung 5 goes FALSE again.

Counting Beyond 9999

When it is necessary to count beyond 9999, one way of accomplishing it is by programming the counter shown in Figure 7.4.

The overflow bit of counter 901 initiates a single count in counter 902 at a count of 10,000. Status bits of both counters are programmed in series to produce an output when an additional 3000 counts are made by counter 901.

Figure 7.4



Letters A to E at the bottom of the timing diagram indicate the following events:

A. Counter 901 begins a series of counts, beginning at AC = RAC = 0000. Instructions 951 in rungs 2 and 3 are FALSE. Examine ON instructions 901 and 902 in rung 4 are FALSE.

B. The AC value has reached counter 901 PR value of 3000. Examine ON instruction 901 in rung 4 goes TRUE, but rung 4 remains FALSE.

C. The AC value of counter 901 overflows beyond 9999, setting the overflow bit (address 951) ON. This causes the following:

Examine ON instruction 951 in rung 3 goes TRUE, resetting counter 901. Examine ON instruction 901 in rung 4 goes FALSE; Examine ON instruction 951 in rung 3 goes FALSE; Counter 901 continues to count from 0000.

Simultaneously, Examine ON instruction 951 in rung 2 has gone TRUE, then FALSE, causing the AC value of counter 902 to go from 0000 to the PR value of 0001. Examine ON instruction 902 in rung 4 goes TRUE. 10,000 counts have occurred.

D. The AC value of counter 901 has again reached the PR value of 3000. Examine ON instruction 901 in rung 4 goes TRUE, making rung 4 TRUE. 13,000 counts have occurred.

E. Rungs 5 and 6 go TRUE. The AC values of both counters are reset to the RAC value. Status bits and overflow bits are reset to OFF. Rung 4 goes FALSE.

While the RST instructions are TRUE, the counter is disabled. Counting can resume when rungs 5 and 6 go FALSE again.

Chapter 8 Timers with Long or Fine Time Base

General

Timers using the long time base and fine time base instructions have the following applications:

Long time base instructions

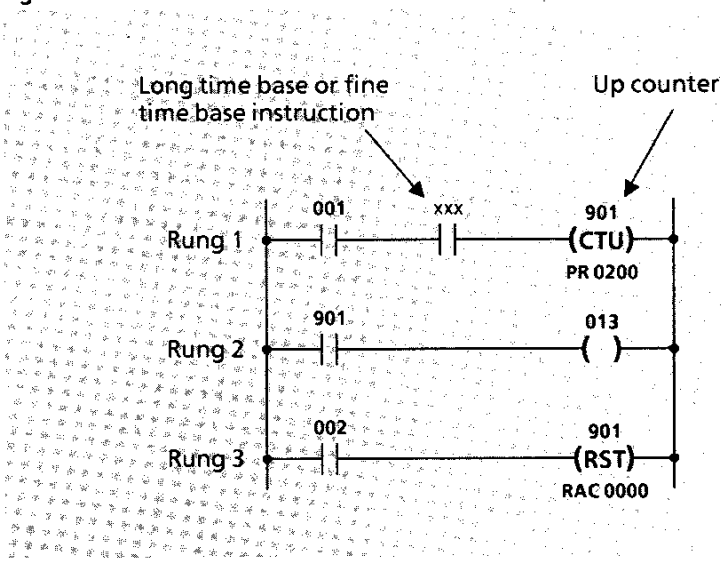
- Time bases of 0.5 sec and 1.0 sec. These instructions allow you to program counter-type timers with long time delays, up to 9999 seconds.
- With SLC 150 processor unit Catalog Nos. 1745-LP151 and 1745-LP152, the 1.0 sec time base is derived from the AC line frequency, useful in "real time" clock applications.

Fine time base instructions

- Choice of 4 time bases plus the scan rate time base. Allows you to program timers with greater resolution than the 0.1 sec provided by the standard timers.

Long time base and fine time base timers use a standard up counter instruction with a special examine instruction representing the time base. This provides the equivalence of an RTO timer. An example appears in Figure 8.1.

Figure 8.1



Long Time Base Instruction

Long time base instructions can be used with SLC 100 and SLC 150 processor units at addresses 874 and 875. This is summarized below.

Figure 8.2

Long Time Base Instructions –All SLC 100 Processors	
Internal Address	Time Increment in seconds
874	0.5 sec. Based on microprocessor clock
875	1.0 sec. Based on microprocessor clock

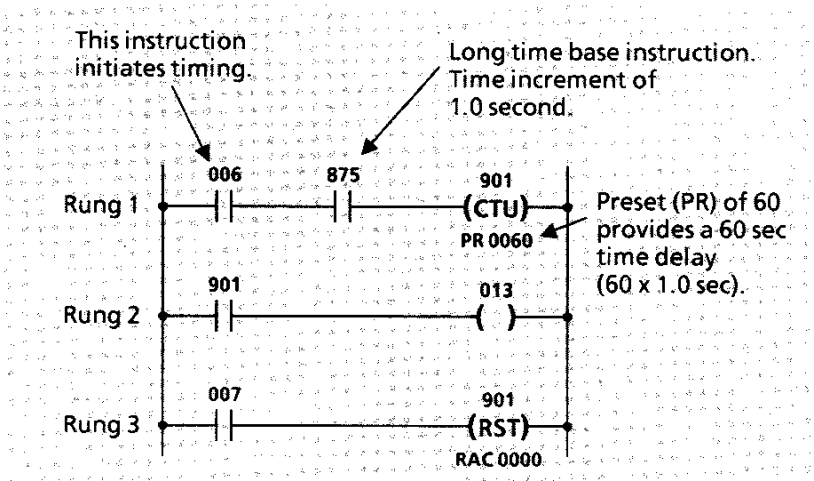
Long Time Base Instructions – SLC 150 Processor		
Processor Unit	Internal Address	Time Increment in seconds
All AC and DC versions	874	0.5 sec. Based on microprocessor clock
DC versions, Catalog Nos. 1745-LP153, -LP154	875	1.0 sec. Based on microprocessor clock
AC versions,Catalog Nos. 1745-LP151, -LP152	875	1.0 sec. Based on AC line frequency

Note: You do not have to measure your scan time in applying these instructions.

The timer in Figure 8.3 illustrates how the long time base instruction can be used. When instruction 006 in rung 1 is TRUE, Examine ON instruction 875 pulses ON and OFF, incrementing the AC value of counter 901. This provides the equivalence of an RTO timer.

Instruction 875 has a time base of 1.0 second. If an AC powered SLC 150 processor is used, timing will be synchronized with the AC line frequency.

Figure 8.3



A long time base timer.

Fine Time Base Instruction

The fine time base instruction allows you to program timers with greater resolution than the 0.1 sec provided by the standard timers. The special internal addresses and corresponding time increments (resolution) are indicated in Figure 8.4 (upper table). To use these addresses, you must first measure the scan time of your program, then refer to the lower table of Figure 8.4 and choose an address compatible with your application needs and the measured scan time.

For proper operation, do not use an address number lower (shorter time increment) than the lowest specified for your particular scan time. Exception: You can always use address 869, which pulses ON and OFF in time with your scan. This provides a time increment equal to twice your scan time. But keep in mind that the scan time of a particular program can vary, and the time increment using address 869 will vary accordingly.

> **Caution:** If you are using the Zero-Cross Turn-On instruction in your program, your scan time will be 8.3 msec (or some multiple of 8.3 msec) at 60 Hz, and 10 msec (or some multiple of 10 msec) at 50 Hz. This will be a deciding factor when applying the instructions listed in Figure 8.4.

Figure 8.4

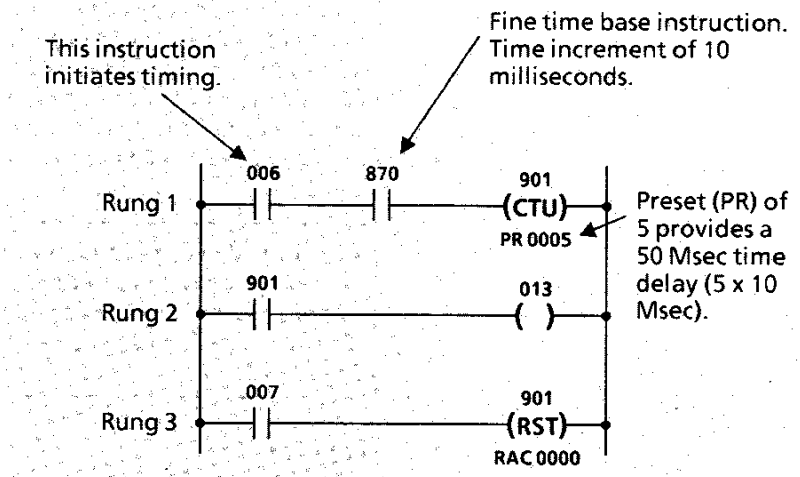
Internal Addresses and Corresponding Time Increments (Resolution)		
Internal Address	Time Increment in Milliseconds	
	SLC 100	SLC 150
869	Scan rate	Scan rate
870	10	10
871	20	20
872	40	50
873	80	100

Acceptable Addresses for your Measured Scan Time		
Measured Scan Time in Milliseconds		Acceptable Addresses
SLC 100	SLC 150	
Less than 5.0	Less than 5.0	870, 871, 872, 873
5.0 to 9.9	5.0 to 9.9	871, 872, 873
10.0 to 19.9	10.0 to 29.9	872, 873
20.0 to 39.9	30.0 to 49.9	873
Any value less than 100	Any value less than 100	869

Procedure for Using the Fine Time Base Instruction

Add the required fine time base timers to your program. An example appears in Figure 8.5 below. Since you do not as yet know the scan time of your program, use any of the special internal addresses, and leave the counter PR value at 0000.

Figure 8.5

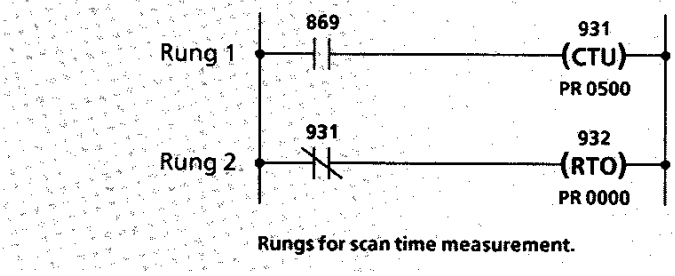


A fine time base timer.

Scan Time Measurement:

1. Add the following two rungs to the end of your program:

Figure 8.6



Rungs for scan time measurement.

Procedure for Using the Fine Time Base Instruction (continued)

2. Enter the Test-Continuous Scan mode (5), then immediately turn off the programmer (measurement will be more accurate if the programmer is off).
3. RTO 932 will measure the time it takes to perform 1000 scans. For a typical SLC 100 program, this would be 15 seconds (15 milliseconds x 1000); for the SLC 150, it would be 4 seconds (4 milliseconds x 1000). Leave the programmer off long enough to assure that scan time measurement is complete (2 minutes would be a safe off time). Now turn the programmer back on and monitor the AC value of the RTO instruction at address 932.
4. Take note of the final AC value displayed. Subtract 1.0 millisecond from this value (0.5 millisecond is added to your program by the scan time measurement rungs and another 0.5 millisecond is added because of Test mode sub-routines). The resultant is the scan time of your program, measured in milliseconds.

   Note: Your scan time will vary. Some conditions which increase scan time:
   - The pocket programmer is turned On.
   - The TCAT is connected.
   - Input pulses to an HSI module are being counted (SLC 150 only).
   - User program instructions are currently being executed.

   We recommend that you take several scan time measurements, under various conditions. This will give you an understanding of scan time variance for your application. When you repeat the scan time measurement, you must first reset counter 931 and timer 932 to zero.
5. Delete the scan time measurement rungs (Figure 8.6).

You can now change the address of the fine time base instruction to an address compatible with Figure 8.4 (lower table) on Page 8-3.

Now set the PR value of the up counter. The time delay in milliseconds will be the PR value multiplied by the time increment specified for the special internal address you selected. In the example of Figure 8.5, address 870 and a PR value of 5 are used, providing a time delay of 50 milliseconds (5 x 10 milliseconds). If you select address 869, the time delay in milliseconds will be your PR value times twice your measured scan rate.

Chapter 9 MCR and ZCL Instructions

MCR and ZCL Instruction Characteristics

The following are special relay-type instructions.

- Master Control Reset –(MCR)– (No address required).
- Zone Control Last State –(ZCL)– (No address required).

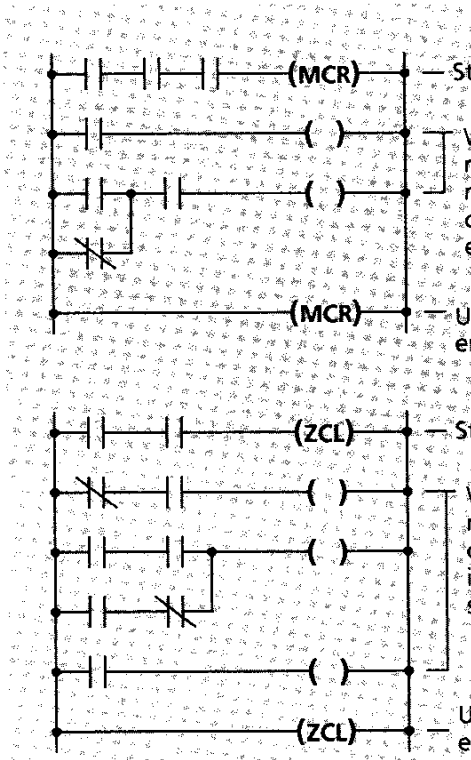
Figure 9.1a illustrates MCR and ZCL *zones*. Note that a zone begins with a start rung and ends with an end rung. The MCR and ZCL instructions are outputs in these rungs, and *do not require an address*.

The start rung contains the condition instructions which control the zone. The end rung contains the MCR or ZCL instruction only. Any number of rungs can be programmed between the start and end rungs.

When the start rung is TRUE, output instructions in the zone function normally. When the start rung goes FALSE, non-retentive outputs within an MCR zone are de-energized; non-retentive outputs in a ZCL zone remain in their last state. The effect on non-retentive and retentive instructions is summarized in Figure 9.1a.

Multiple MCR and ZCL zones can be programmed. With the SLC 150, ZCL zones can be nested (Page 9-4). MCR zones should not be nested.

Figure 9.1a



Instruction	MCR Zone Start Rung FALSE	ZCL Zone Start Rung FALSE
Non-Retentive Outputs	De-energized	Remain in last state
Latch/Unlatch	Remain in last state	
Counters, RTO Timers	AC value stops incrementing, value retained. Status, overflow, and underflow bits remain in last state.	
RTF Timers*	AC value increments	AC value stops incrementing, value retained. Status, overflow, and underflow bits remain in last state.
Sequencers	AC value stops incrementing, value retained. Step number retained. Bit addresses remain in last state.	
Reset	Remains in last state	
Shift Register	Outputs disabled. Shifting disabled.	Outputs remain in last state. Shifting disabled.

*CAUTION Since the RTF timer runs inside of a FALSE MCR zone, its accumulator, status, and overflow bits could change state while an MCR zone is FALSE. This could affect other outputs in your program.

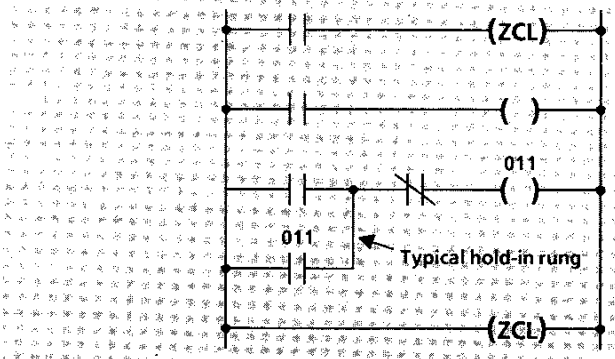
MCR and ZCL Instruction Characteristics *(continued)*

Figure 9.1b lists additional operating characteristics of non-retentive outputs inside a ZCL zone.

Figure 9.1b

The table below applies for either of the following conditions:

1. ZCL zone is FALSE. Non-retentive outputs inside ZCL zone are being held in last state.
2. ZCL zone is either TRUE or FALSE. Non-retentive output instruction inside ZCL zone is being held ON by a hold-in rung.



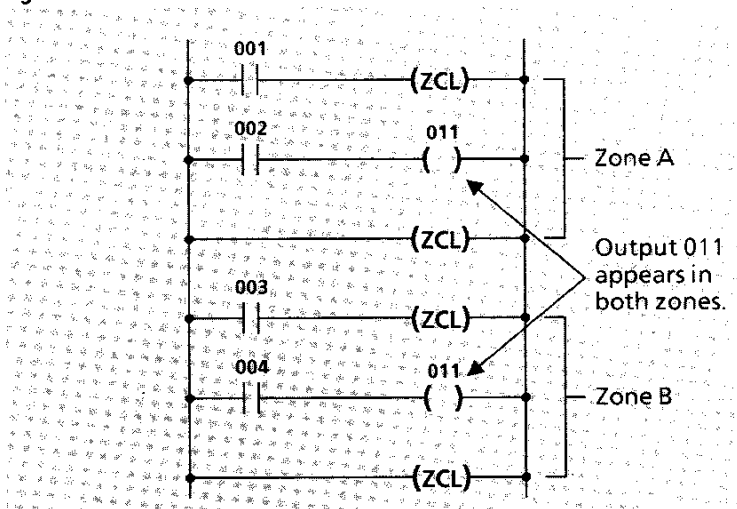
Event	Status of Outputs Within ZCL Zone	
	SLC 100 Series A, B, C and SLC 150 Series A	SLC 150 Series B
Power is cycled from Off to On.	Outputs return to their last state.	Outputs turned Off.
Processor is switched from the Run mode to any other operating mode, then switched back to the Run mode.	Outputs return to their last state. However, if the Program mode was entered and a program change was made, outputs will be Off.	Outputs turned Off.

ZCL Zones Using the Same Outputs

You may use two or more ZCL zones in your program. Multiple ZCL zones allow you to use the same output instruction several times in your program without experiencing contention between them. Figure 9.2 shows how this works.

If zone control were not included in Figure 9.2, the output instruction that appears last in the program would govern the state of the external output; using multiple zones avoids this. To avoid contention between the output instructions, only one ZCL zone should be TRUE at a time.

Figure 9.2



Zone A	Zone B	Control of Output 011
TRUE	FALSE	Output 011 is controlled by instruction 002 in zone A. Zone B is bypassed in the program scan and therefore has no effect.
FALSE	FALSE	When zone A goes FALSE, output 011 remains in its last state. Both zones are now bypassed in the program scan.
FALSE	TRUE	When zone B goes TRUE, output 011 is now controlled by instruction 004 in zone B. Zone A is FALSE and has no effect.
TRUE	TRUE	In this case, output 011 is controlled by instruction 004 in Zone B. This is because zone B is scanned *after* zone A and therefore takes precedence.

Nesting ZCL Zones

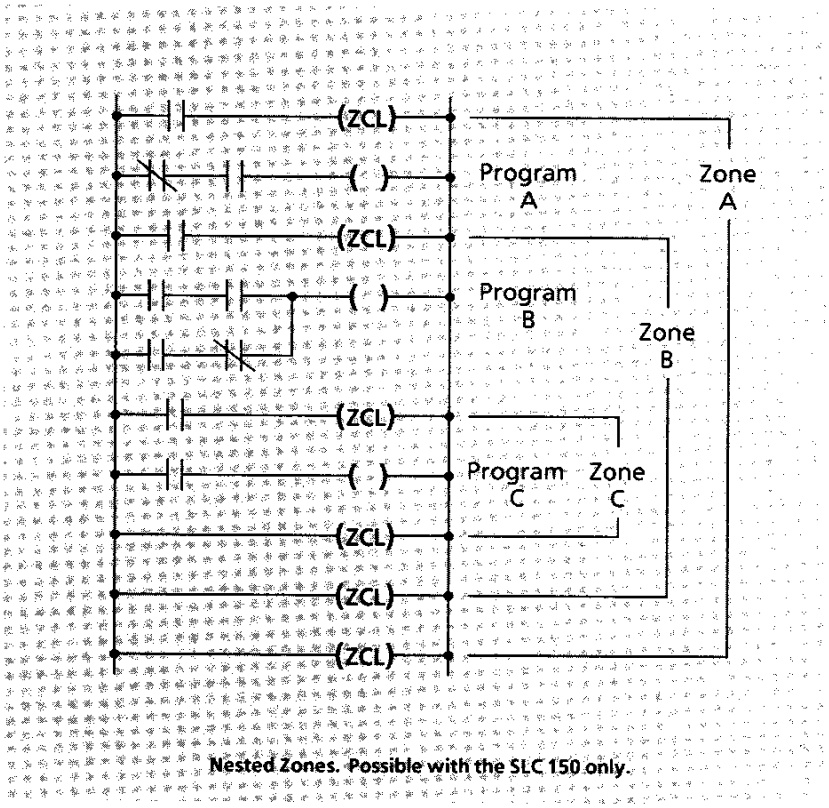
SLC 100: You cannot nest ZCL zones (place one zone within another) with the SLC 100 processor unit. It will produce undesired results.

SLC 150: Up to eight ZCL zones can be nested together. Figure 9.3 illustrates three nested zones. Note that each zone has a conditional start rung and an unconditional end rung. ZCL zone A controls outputs in programs A, B, and C. Zone B controls outputs in programs B and C. Zone C controls outputs in program C.

An outer zone must be TRUE in order for internal zones to control their outputs. If an outer zone is FALSE, all outputs within internal zones are held in their last state, regardless of the rung conditions of the internal zones.

Error code 27-5: If ZCL zones are nested deeper than eight zones, or if each start rung is not accompanied by an end rung, a processor fault will occur and error code 27-5 will be displayed on the pocket programmer. This prompts you to correct the condition.

Figure 9.3



Nested Zones. Possible with the SLC 150 only.

MCR and ZCL Instructions vs a Master Control Relay

In some respects, the function of MCR and ZCL instructions is similar to a hard-wired master control relay of a relay system, which can be wired to control the overall function of several operations, based on certain necessary conditions. An important distinction must be made, however:

The master control relay is a safety device, providing Emergency Stop capability to the controller operator; with MCR and ZCL instructions, this is **not** the case.

We strongly recommend that you install a hard-wired master control relay to provide emergency I/O power shut-down. Refer to Pages 19-4 and 19-5 for wiring details.

Chapter 10 Sequencer Instructions

SQO and SQI Sequencer Characteristics

Sequencer instructions include the Sequencer Output –(SQO)– and the Sequencer Input –(SQI)–. Both are retentive, requiring the use of the Reset instruction. You will find that sequencer instructions will be easier to understand if you first become familiar with timer and counter instructions.

Sequencer instructions are represented by the following symbols:

Address: 901-932 (internal). 100 steps maximum.

High Speed Input Module Applications (SLC 150 only): Sequencer Output instructions at addresses 901, 902, 903, and 904 are used in programming high speed input modules. Refer to Chapter 12 for programming details.

If you are not using high speed input modules, addresses 901, 902, 903, and 904 can be used for standard timer, counter, and sequencer programming.

Time- or Event-Driven

Sequencers can be *time-driven* or *event-driven.* Operation involves an Accumulated value (AC) and a Preset value (PR).

Time-driven sequencers count 0.1-second intervals while the sequencer rung is TRUE. When the AC value reaches the PR value, the sequencer advances to the next step and the AC value increments from 0000 again. After the final step, the sequencer continues with step 0.

Event-driven sequencers count FALSE-TRUE transitions of the sequencer rung. When the AC value reaches the PR value, the sequencer advances to the next step and the AC value increments from 0000 again. After the final step, the sequencer continues with step 0.

The Sequencer Output (SQO) instruction sets the ON/OFF status of up to 8 bit addresses for each step. This group of bit addresses can be selected as either 6 external output addresses plus 2 internal addresses, or as 8 internal addresses.

The Sequencer Input (SQI) instruction examines the ON/OFF status of up to 8 bit addresses for each step, setting an input-satisfied status bit ON when the status of the bit addresses matches programmed data. The group of bit addresses can be selected as either external input addresses or internal addresses.

Status and Completion Bits

The SQI sequencer *input-satisfied status bit* and the SQO sequencer *step completion bit* use the sequencer instruction address:

The SQI input-satisfied status bit is set ON when the ON/OFF status of the bit addresses matches the programmed data for the current step. The SQO step completion bit is set ON for a single program scan (approx. 15 msec) each time the SQO instruction completes a step. Examine ON instructions at the bit address go TRUE; Examine OFF instructions go FALSE.

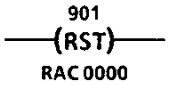
The SQO step completion bit is discussed further with shift registers, in Chapter 13.

The SQO *cycle completion bit* is assigned the sequencer address plus 50. (901 + 50 = 951):

The SQO cycle completion bit is set ON when the sequencer completes its final step. Examine ON instructions at the bit address go TRUE; Examine OFF instructions go FALSE.

Reset Instruction

The reset (RST) instruction is given the same address as the sequencer instruction:



When the RST instruction goes TRUE, the SQO cycle completion bit is reset to OFF, and the sequencer is reset to the step number corresponding to the RAC value. Outputs for the step are controlled as programmed.

The RST instruction must go FALSE again before sequencer operation can resume.

Sequencer Data

In programming the ON/OFF status of bit addresses for sequencer steps, 1's and 0's are used, where 1 = ON and 0 = OFF. To simplify programming, a 2-digit code is used to represent the status of the bit addresses for a particular step.

Group numbers define which addresses are involved in the sequencer data. Masking data (using the 2-digit code) indicates which of these addresses apply to the sequencer and which do not.

Additional details on sequencer data appear on Page 10-6.

Monitoring and Changing Data

The PR value, AC value, and current step number can be monitored in the Run and Test modes. The PR value of any step can be changed in the Run or Test modes. Use the PRT/UNPRT key to protect the PR value from being changed.

Power Down

The sequencer status is retentive. When you apply power after a power-down, the current step number, AC value, and the completion bit status will be the same as before power was removed. The status of the input-satisfied bit will also be the same, provided the status of the external inputs remain the same.

The sequencer status is also retained when going from the Run mode to another mode.

Operation after CPU Faults: Sequencer instructions and associated status bits and data values are reset to their initial conditions after a CPU fault is detected (error codes 1 thru 9, 26, and 51).

Basic Operation – Sequencer Output

The SQO instruction can control the ON/OFF status of up to 8 bit addresses for up to 100 steps. It can be time-driven or event-driven. With time-driven sequencers, each step functions similar to timer instructions, involving an AC value and a programmed PR value (*dwell time*). In the same way, the event-driven sequencer functions similar to the counter instruction.

Time-Driven SQO:

Step	Bit Addresses (Data Entry: ON = 1, OFF = 0)					Dwell Time (PR Value)
	A	B	C	D	E	
0	OFF	OFF	OFF	OFF	OFF	5 seconds
1	OFF	ON	OFF	ON	OFF	20 seconds
2	ON	ON	ON	ON	OFF	60 seconds
3	ON	OFF	OFF	ON	ON	10 seconds

Beginning at step 0: With rung conditions FALSE, the SQO instruction is waiting on step 0. Bit addresses A thru E are OFF in accordance with data for step 0. When the SQO instruction goes TRUE, step 0 is initiated; bit addresses A thru E remain OFF for a dwell time of 5 seconds (assuming SQO remains TRUE). Then step 1 begins; bit addresses B and D go ON. After 20 seconds, step 2 begins; bit addresses A and C go ON. After 60 seconds, step 3 begins; bit addresses B and C go OFF, E goes ON. After 10 seconds, a completion bit is set ON and the cycle repeats with step 0.

Event-Driven SQO:

Step	Bit Addresses (Data Entry: ON = 1, OFF = 0)					FALSE-TRUE Transitions (PR Value)
	A	B	C	D	E	
0	OFF	OFF	OFF	OFF	OFF	1
1	OFF	ON	OFF	ON	OFF	1
2	ON	ON	ON	ON	OFF	1
3	ON	OFF	OFF	ON	ON	1

The PR value is set at 1 for each step (the typical case). Beginning with step 0, bit addresses A thru E are OFF. After a FALSE-TRUE transition of the SQO instruction occurs, step 1 is in effect; bit addresses B and D go ON. After a 2nd transition, step 2 is in effect; A and C go ON. After a 3rd transition, step 3 is in effect; B and C go OFF, E goes ON. After a 4th transition, a completion bit is set ON and the cycle repeats with step 0.

Basic Operation – Sequencer Input

The SQI instruction differs from the SQO instruction in that it *examines* up to 8 bit addresses and sets an input-satisfied status bit ON when the ON/OFF status of these bit addresses matches the programmed data for a particular step. The SQI sequencer can be either time-driven or event-driven.

Time-Driven SQI:

Step	Bit Addresses (Data Entry: ON = 1, OFF = 0)					Dwell Time (PR Value)	Input-Satisfied Status Bit
	A	B	C	D	E		
0	OFF	OFF	OFF	OFF	OFF	120 seconds	Each step: This bit is ON only if inputs match programmed data.
1	OFF	ON	OFF	ON	OFF	60 seconds	
2	ON	ON	ON	ON	OFF	60 seconds	
3	ON	OFF	OFF	ON	ON	120 seconds	

This sequencer moves from step to step in the same way as the time-driven SQO sequencer.

During the time that a particular step of the SQI instruction is in effect, the input-satisfied status bit will be set ON only when the status of bit addresses A thru E matches the programmed data for that step.

Event-Driven SQI:

Step	Bit Addresses (Data Entry: ON = 1, OFF = 0)					FALSE-TRUE Transitions (PR Value)	Input-Satisfied Status Bit
	A	B	C	D	E		
0	OFF	OFF	OFF	OFF	OFF	1	Each step: This bit is ON only if inputs match programmed data.
1	OFF	ON	OFF	ON	OFF	1	
2	ON	ON	ON	ON	OFF	1	
3	ON	OFF	OFF	ON	ON	1	

This sequencer moves from step to step in the same way as the event-driven SQO sequencer.

During the time that a particular step of the SQI instruction is in effect, the input-satisfied status bit will be set ON only when the status of bit addresses A thru E matches the programmed data for that step.

Sequencer Data

Sequencer data includes:

- Sequencer classification: SQI or SQO; instruction address; time- or event-driven.
- Group number and corresponding 8 bit addresses.
- Mask data, indicating which of the 8 bit addresses will be used.
- Step data, indicating the ON/OFF status of the bit addresses for each step.
- Program codes for the mask and step data.
- Preset values for each step.

We recommend that you use the Sequencer Instruction Data Form to document this data. This form (included at the end of this manual) allows you to document sequencer data in an orderly, systematic way, reducing the chances for programming errors.

A sample data form is shown below.

CLASSIFICATION: ☐ -(SQI)- ☐ -(SQO)- ADDRESS: ____ ☐ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: ____

BIT ADDRESS DATA										PROGRAM CODE		PRESET VALUES
		B				A				Data B	Data A	
Bit Addresses →												
Mask Data →												
Step Data →	0											
	1											
	2											
	3											

Special Note for SLC Personal Computer Software Users

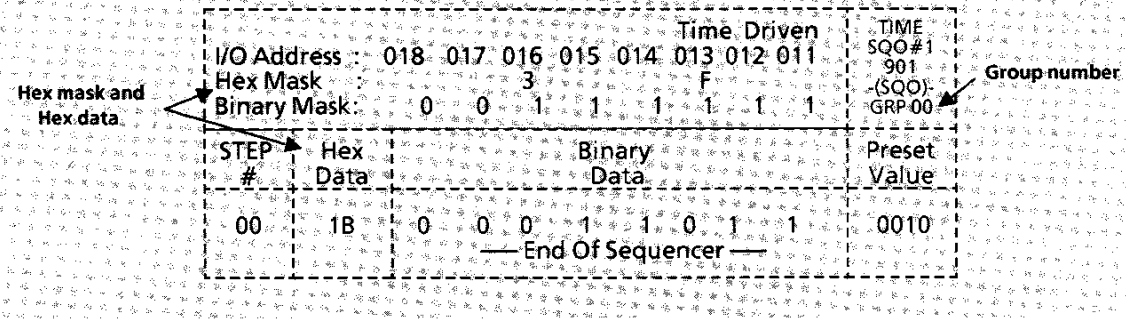
You will find that sequencer data and the sequencer data form differ slightly from the data and form discussed in this manual.

The sequencer data form as it applies to the SLC personal computer software is shown below. The data you must fill in from the tables on the following page are:

- Group number
- Hex mask
- Hex data

Example: In the form below, Group 0 has been specified, automatically entering I/O addresses 011 thru 018 in the form. Hex Mask 3F has been specified, automatically entering binary mask 0011 1111 in the form. Hex Data 1B has been specified for step 0, automatically entering binary data 0001 1011 in the form.

For "Special Sequencer Instructions", using group numbers 38 thru 69, the software calls for Decimal Data instead of Hex Data. The decimal data required are step numbers, as explained in Chapter 11.



Sequencer Data *(continued)*

Group numbers and program codes are used in filling out the table. They are listed below.

Figure 10.1

Group Numbers

External Output Bit Addresses and Internal Bit Addresses (SQO Sequencers)

Bit Addresses	Group Number	Bit Addresses	Group Number
011-016 (output) 017-018 (internal)	0	411-416 (output) 417-418 (internal)	4
111-116 (output) 117-118 (internal)	1	511-516 (output) 517-518 (internal)	5
211-216 (output) 217-218 (internal)	2	611-616 (output) 617-618 (internal)	6
311-316 (output) 317-318 (internal)	3		

External Input Bit Addresses (SQI Sequencers)

Bit Addresses	Group Number
001-008	7
101-108	8
201-208	9
301-308	10
401-408	11
501-508	12
601-608	13
- 009-010 109-110 209-210	14
309-310 409-410 509-510 609-610	15

Internal Bit Addresses (SQO or SQI Sequencers)

Bit Addresses	Group Number	Bit Addresses	Group Number
701-708	16	789-796	27
709-716	17	797-804	28
717-724	18	805-812	29
725-732	19	813-820	30
733-740	20	821-828	31
741-748	21	829-836	32
749-756	22	837-844	33
757-764	23	845-852	34
765-772	24	853-860	35
773-780	25	861-868	36*
781-788	26	869-876	37*

***NOTE**: The first two bit addresses are not used for Group 14.

***NOTE**: Bit addresses 864 thru 876 apply to special instructions, explained in the following chapters:

Addresses 869-875: Chapter 8.

Addresses 864-868, 876: Chapter 14.

Program Codes

(use Hex Mask and Hex Data codes for personal computer software data form)

Bit Address Data	Program Code
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Example - For data 0001 1111, enter code 1 for Data B, and code F for Data A.

Thus: 0001 1111

1 F

Hex Mask and Hex Data Codes

(for personal computer software data form)

Binary Value	Hex Value
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Note that the data in the "Binary Value" and "Hex Value" columns is equivalent to the data in the "Bit Address Data" and "Program Code" columns in the "Program Codes" table above.

Filling in the Sequencer Data Form

To illustrate how the form is used, we'll enter data for the following time-driven SQO sequencer (same sequencer as discussed on Page 10-4).

Time-Driven SQO:

Step	Bit Addresses (Data Entry: ON = 1, OFF = 0)					Dwell Time (PR Value)
	A	B	C	D	E	
0	OFF	OFF	OFF	OFF	OFF	5 seconds
1	OFF	ON	OFF	ON	OFF	20 seconds
2	ON	ON	ON	ON	OFF	60 seconds
3	ON	OFF	OFF	ON	ON	10 seconds

Follow these 3 steps (an explanation follows each data form figure):

Step 1: a) Enter sequencer classification. b) Enter group number, bit addresses.

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☒ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A				Data B	Data A	
Bit Addresses →	018	017	016	015	014	013	012	011			
Mask Data →											
Step Data → 0											
1											
2											
3											

Step 1: a) Enter the sequencer classification. We've indicated SQO, address 901, and time-driven in the heading of the data form.

b) Enter the group number and corresponding 8 bit addresses. Determine the sequencer group number from Figure 10.1, Page 10-7. We've selected group number "0", corresponding to bit addresses 011-018. Addresses 011 thru 016 apply to the external outputs of the processor unit; addresses 017 and 018 apply to internal relay-type instructions.

List these 8 bit addresses in *reverse* order, beginning with the highest number in the far left column under **B**.

Step 2: a) Enter mask data. b) Enter step data.

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☒ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	0	1	0	0	1	1	1	1			
Step Data → 0		0			0	0	0	0			
1		0			1	0	1	0			
2		1			1	1	1	0			
3		1			0	0	1	1			

Step 2: a) Enter mask data. Enter a 1 in the form under the addresses to be controlled by the sequencer. Enter a 0 under those excluded. Our sequencer has 5 outputs; we've excluded addresses 015, 016, and 018 from sequencer control. These addresses can be used elsewhere in your program.

Note that the sequencer is controlling 4 external output addresses (011 thru 014) and 1 internal address (017).

b) Enter step data. 1 corresponds to an ON condition and 0 corresponds to an OFF condition. Thus, for each step, the ON and OFF states of A, B, C, D, and E correspond to the 1's and 0's entered under addresses 017, 014, 013, 012, and 011 in the data form.

Step 3: a) Enter program entry codes for mask and step data. b) Enter preset values.

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☒ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	0	1	0	0	1	1	1	1	4	F	
Step Data → 0	⊗	0	⊗	⊗	0	0	0	0	0	0	5.0
1	⊗	0	⊗	⊗	1	0	1	0	0	A	20.0
2	⊗	1	⊗	⊗	1	1	1	0	4	E	60.0
3	⊗	1	⊗	⊗	0	0	1	1	4	3	10.0

⊗ Masked addresses are 0 for coding.

Step 3: a) Enter program codes for mask and step data. The program code is shown in Figure 10.1, Page 10-7. Applying this code to mask data 0100 under **B**, we've entered 4 under "Data **B**" in the form. Similarly, mask data 1111 under "Data **A**" is identified by code F.

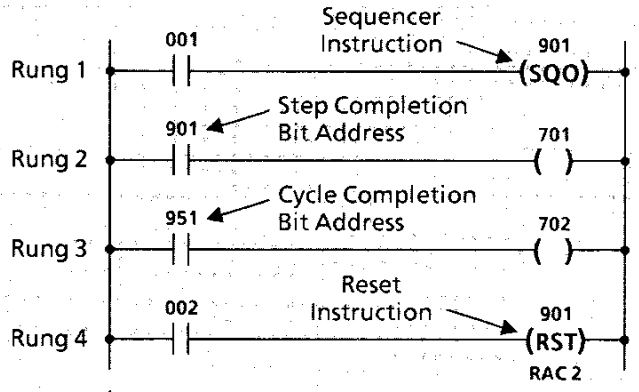
Note: For step data, masked addresses are considered as 0's. Thus, for step 3 under **B**, the step data is considered to be 0100. Code 4 is entered.

b) Enter Preset values. We've specified the PR values to the tenth of a second, just as with timers.

Sequencer Output (SQO)

Figure 10.2 shows a ladder diagram and a completed data form for an SQO instruction. This is a 10-step, time-driven sequencer controlling 4 external output addresses (012, 014, 015, 016) and 1 internal address (017). Operation is explained below the data form.

Figure 10.2



Ladder Diagram

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☒ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA										PROGRAM CODE		PRESET VALUES
		B				A						
Bit Addresses →		018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →		0	1	1	1	1	0	1	0	7	A	
Step Data →	0	⊗	1	0	1	0	⊗	1	⊗	5	2	5.0
	1	⊗	1	0	1	1	⊗	0	⊗	5	8	5.0
	2	⊗	0	1	0	0	⊗	0	⊗	2	0	2.5
	3	⊗	1	0	1	0	⊗	0	⊗	5	0	1.0
	4	⊗	0	0	0	1	⊗	1	⊗	0	A	1.0
	5	⊗	1	0	1	1	⊗	0	⊗	5	8	1.0
	6	⊗	1	1	0	0	⊗	0	⊗	6	0	0.5
	7	⊗	1	0	0	1	⊗	1	⊗	4	A	0.5
	8	⊗	0	1	1	1	⊗	0	⊗	3	8	1.5
	9	⊗	0	1	1	1	⊗	1	⊗	3	A	3.0

⊗ Masked address. Used as 0 for coding purposes.

Operation begins with rung 1 FALSE and the SQO instruction on step 0. External output addresses 12 and 15 are ON, and internal bit address 17 is ON. Now rung 1 goes TRUE. As long as the rung is TRUE, the sequencer advances from one step to the next each time the PR value of the current step is reached. At the beginning of each step, bit addresses 012, 014, 015, 016, and 017 are set ON or OFF as indicated by the 1's and 0's in the data form. (Addresses 011, 013, and 018 are not used.)

Rung 2 demonstrates how the step completion bit functions. Each time a step is completed, this bit is set to ON for a single program scan, during which time the Examine ON instruction in rung 2 will be TRUE. This bit is used in shift register applications.

Rung 3 demonstrates how the cycle completion bit functions. The first time the sequencer recycles from the final (9th) step to step 0, the cycle completion bit is set ON, causing the Examine ON instruction in rung 3 to go TRUE. The cycle completion bit remains ON until the sequencer is reset.

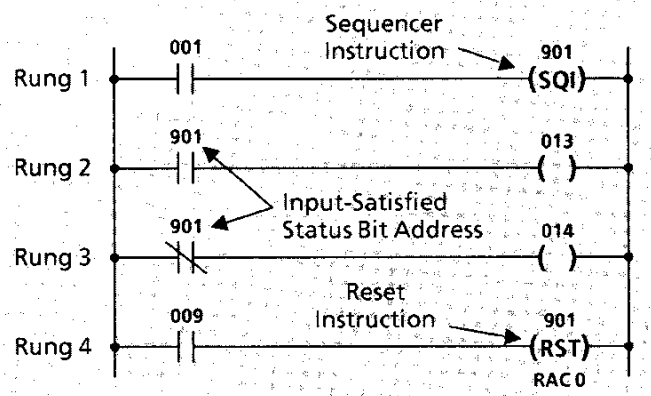
Rung 4 contains the reset instruction. When this rung goes TRUE, the sequencer is reset to step 2, corresponding to the programmed RAC value. Also, the cycle completion bit is reset to OFF.

While the RST instruction is TRUE, the sequencer is held on its RAC step with the corresponding outputs controlled as programmed. Operation can resume when rung 4 goes FALSE again.

Sequencer Input (SQI)

Figure 10.3 shows a ladder diagram and a completed data form for an SQI instruction. This 10-step, event-driven sequencer examines 7 external input addresses, 002-008. Bit address 001 is not used. Operation is explained below the data form.

Figure 10.3



Ladder Diagram

CLASSIFICATION: ☒ –(SQI)– ☐ –(SQO)– ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 7

	BIT ADDRESS DATA								PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	008	007	006	005	004	003	002	001	Data B	Data A	
Mask Data →	1	1	1	1	1	1	1	0	F	E	
Step Data → 0	1	1	0	0	1	1	1	⊗	C	E	1
1	0	0	0	1	1	0	0	⊗	1	8	1
2	0	0	1	1	1	1	1	⊗	3	E	1
3	1	1	0	0	1	1	1	⊗	C	E	1
4	1	1	0	0	0	0	0	⊗	C	0	1
5	1	0	0	0	0	0	0	⊗	8	0	1
6	0	1	0	0	0	1	0	⊗	4	4	1
7	0	0	1	0	0	0	0	⊗	2	0	1
8	1	0	1	0	1	0	1	⊗	A	A	1
9	0	1	0	0	1	1	1	⊗	4	E	1

⊗ Masked address. Used as 0 for coding purposes.

FALSE-TRUE transitions of rung 1 move the sequencer from one step to the next. In this case, a single FALSE-TRUE transition advances the sequencer, since all PR values are set at 1.

The ON/OFF states of the external inputs determine whether the input-satisfied status bit will be ON or OFF. That is, when the ON/OFF states of the external inputs match the data programmed for the current step, the input-satisfied status bit will be ON; when the external inputs do not match the data programmed for the current step, the status bit will be OFF.

Rungs 2 and 3 are output rungs, containing examine instructions at the input-satisfied status bit address. When the status bit is ON, rung 2 is TRUE and rung 3 is FALSE; when the status bit is OFF, rung 2 is FALSE and rung 3 is TRUE.

Rung 4 contains the reset instruction. When this rung goes TRUE, the sequencer is reset to step 0 (corresponds to RAC value).

While the RST instruction is TRUE, the sequencer is held on its RAC step and the SQI will continue to examine inputs. Operation can resume when rung 4 goes FALSE again.

Programming Sequencer Instructions

The keystroke example below shows you how to program rung 1 of the sequencer appearing on Page 10-10. This example illustrates how to program time- or event-driven operation, the group number, mask data, step data, and PR values. For practice, you may want to program all ten sequencer steps and all four rungs of the sequencer ladder diagram.

Figure 10.4

Keystroke Example – Sequencer Instruction.

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☒ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	0	1	1	1	1	0	1	0	7	A	
Step Data → 0	⊗	1	0	1	0	⊗	1	⊗	5	2	5.0
1	⊗	1	0	1	1	⊗	0	⊗	5	8	5.0
2	⊗	0	1	0	0	⊗	0	⊗	2	0	2.5

(Complete ladder diagram and data form appear on Page 10-10.)

Press these keys	This LED will be lit (in addition to PROG)	Display will show	Explanation
		Address / Data / Rung / Err / Mode: 885 End	Arbitrarily, we start with the display showing 885 words of memory (max for SLC 100).
–] [–, 1	–] [–	__ | 1	Examine ON instruction, address 001 is entered.
SHIFT, –(SQO)–	–(SQO)–	___ 1	Display is prompting for address of the SQO instruction.
9, 0, 1, ENTER	–(SQO)–	_ C-E	Address 901 is entered. Display shows **C – E**, prompting you to specify time-driven or event-driven.
SHIFT, TIME, ENTER	–(SQO)–	__ grp	We have specified time-driven (we could also have pressed C instead of TIME). Display is prompting you for the group number.
0, ENTER	–(SQO)–	__ USE	Group 0 is entered. Display shows **USE**, prompting you for mask data.
7, SHIFT, A, ENTER	–(SQO)–	__ d 0	Mask data 7, A is entered from the data form. Display is prompting you for step data of step 0.
5, 2, ENTER	–(SQO)–	____ P 0	Step data 5, 2 is entered from the data form. Display is prompting for step 0 PR value. **Note:** With event-driven sequencers, a PR value of 1 is entered automatically (you can change this).
5, 0, ENTER	–(SQO)–	__ d 1	PR value of 5.0 is entered from the data form. Display is prompting you for step data of step 1.
Keys for steps 1-9, ENTER	PROG	885 End	The display will continue prompting for step data and PR values. When entering the PR value of step 9, press ENTER a second time. This ends the data entry; the display at the left will appear.

Chapter 11 Special Sequencer Instruction Techniques

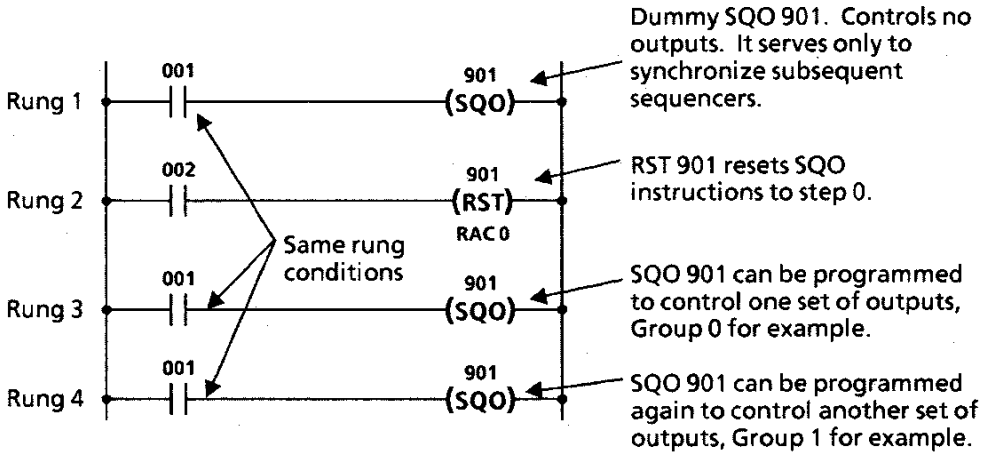
Cascading SQO Instructions

SQO instructions can be cascaded in order to control more than 8 bit addresses. An example is shown in Figure 11.1a. Rung 1 is a dummy sequencer. Rung 2 resets all sequencer instructions in the program. It is important that this reset rung follows the dummy sequencer rung so that the cascaded sequencers are reset properly. Rungs 3 and 4 are cascaded sequencers controlling 16 bit addresses.

With cascaded SQO instructions, bit addresses controlled by the 2nd, 3rd, and subsequent sequencers are set in the same I/O scan; bit addresses controlled by the 1st sequencer are set one scan later. This is the reason we've made the rung 1 sequencer a dummy. It controls no bit addresses. It serves only to synchronize the setting of bit addresses of the sequencers in rungs 3 and 4.

Sequencers in rungs 1, 3, and 4 have the same address and are operated by the same rung conditions. They are event-driven (could also be time-driven), and have the same presets and number of steps. The sequencer in rung 3 controls Group 0 outputs; the sequencer in rung 4 controls Group 1 outputs; the sequencer in rung 1 has Mask data of 00, so that it controls no outputs at all. Data forms are shown in Figure 11.1b on the next page.

Figure 11.1a



Cascading sequencers. See Figure 11.1b for data forms.

Cascading SQO Instructions (continued)

The data forms below apply to the cascaded sequencers in Figure 11.1a on the previous page.

Figure 11.1b

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 0

Rung 1 Dummy Sequencer

		BIT ADDRESS DATA								PROGRAM CODE		PRESET VALUES
		B				A						
Bit Addresses →		018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →		0	0	0	0	0	0	0	0	0	0	
Step Data →	0	1	1	0	1	0	0	0	1	D	1	1
	1	0	0	1	0	1	0	1	0	2	A	1
	2	1	1	0	0	0	1	0	0	C	4	1
	3	0	0	0	1	0	0	0	1	1	1	1

Same data as rung 3 except Mask is 00

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 0

Rung 3 Sequencer

		BIT ADDRESS DATA								PROGRAM CODE		PRESET VALUES
		B				A						
Bit Addresses →		018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →		1	1	1	1	1	1	1	1	F	F	
Step Data →	0	1	1	0	1	0	0	0	1	D	1	1
	1	0	0	1	0	1	0	1	0	2	A	1
	2	1	1	0	0	0	1	0	0	C	4	1
	3	0	0	0	1	0	0	0	1	1	1	1

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 1

Rung 4 Sequencer

		BIT ADDRESS DATA								PROGRAM CODE		PRESET VALUES
		B				A						
Bit Addresses →		118	117	116	115	114	113	112	111	Data B	Data A	
Mask Data →		1	1	1	1	1	1	1	1	F	F	
Step Data →	0	1	1	0	0	1	1	0	0	C	C	1
	1	1	0	0	0	1	0	0	0	8	8	1
	2	0	1	0	0	1	1	0	1	4	D	1
	3	1	0	1	0	0	1	0	0	A	4	1

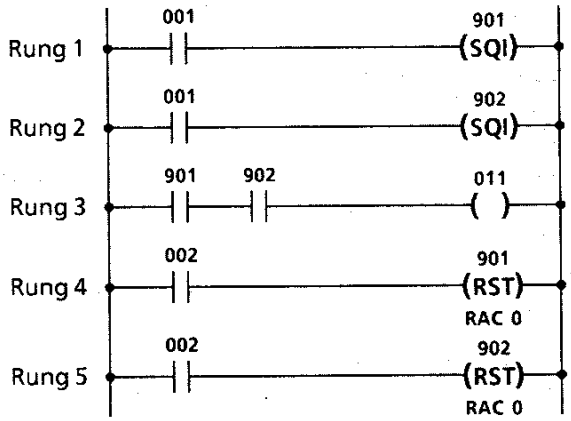
Note for SLC Personal Computer Software Users

The data form you are using prompts you for Hex Mask and Hex Data. This corresponds to the Program Code used in the forms above. A comparison of the sequencer data form you are using and the form shown above appears on Page 10-6.

Cascading SQI Instructions

SQI instructions can be cascaded in order to examine more than eight input addresses. The example below shows how this is done. (Note that a dummy sequencer is not required as with SQO instructions.)

Figure 11.2



Input addresses 001 thru 008 are examined by SQI 901. Addresses 009 and 010 are examined by SQI 902. Input conditions must be satisfied for both sequencers in order to energize instruction 011 in rung 3.

Both sequencers must have the same number of steps. In this example, examine instruction 002 resets both sequencers to step 0.

CLASSIFICATION: ☒ –(SQI)– ☐ –(SQO)– ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 7

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	008	007	006	005	004	003	002	001	Data B	Data A	
Mask Data →	1	1	1	1	1	1	1	1	F	F	
Step Data → 0	0	0	0	1	0	0	0	1	1	1	1
1	0	0	1	0	0	0	1	0	2	2	1
2	0	1	0	0	0	1	0	0	4	4	1
3	1	0	0	0	1	0	0	0	8	8	1
4	0	0	0	0	0	0	0	0	0	0	1

Sequencer data form for rung 1

CLASSIFICATION: ☒ –(SQI)– ☐ –(SQO)– ADDRESS: 902 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 14

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	210	209	110	109	010	009	–	–	Data B	Data A	
Mask Data →	0	0	0	0	1	1	0	0	0	C	
Step Data → 0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	0	0	1
4	0	0	0	0	1	1	0	0	0	C	1

Sequencer data form for rung 2

Reversing Sequencer Step Operation

The reversing sequencer operation allows you to operate a sequencer in a forward or reverse step order. An example is shown in Figure 11.3. Rung 1 is the forward operating sequencer. Rung 3, having identical rung conditions, implements reverse operation of the sequencer; note that this rung is within a ZCL zone.

Operation: When rung 1 is TRUE and rung 2 is FALSE, the sequencer operates in a forward step order (rung 3 is TRUE, but inoperative). When rung 2 goes TRUE, rung 3 becomes operative and the sequencer reverses the step order, beginning with the step in effect at the time. When rung 2 goes FALSE again, the sequencer operates in a forward step order again.

The data forms for rungs 1 and 3 are included in the figure. Data for rung 1 is entered using the program codes listed in Figure 10.1, Page 10-7. Enter rung 3 according to the following special procedure:

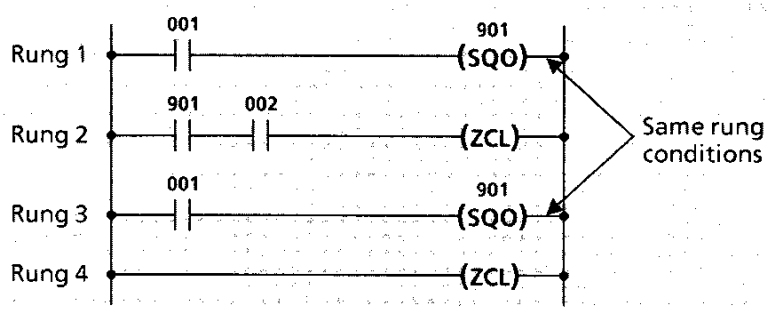
1. If you like, you can enter rung 3 by copying rung 1, then modifying the data. The SQO instruction address is 901, and it is time-driven.
2. Select a special group number from the table below. The number must correspond to the address number of the sequencer. In this case number 38 applies.

Address	Group No.	Address	Group No.	Address	Group No.	Address	Group No.
901	**38**	909	**46**	917	**54**	925	**62**
902	**39**	910	**47**	918	**55**	926	**63**
903	**40**	911	**48**	919	**56**	927	**64**
904	**41**	912	**49**	920	**57**	928	**65**
905	**42**	913	**50**	921	**58**	929	**66**
906	**43**	914	**51**	922	**59**	930	**67**
907	**44**	915	**52**	923	**60**	931	**68**
908	**45**	916	**53**	924	**61**	932	**69**

3. You must specify FF for mask data. This is required when you use any of special group numbers 38-69.
4. Enter the special step data for reverse step order, as directed below the rung 3 data form.
5. Preset values for both sequencers must be the same, in this case a value of 1.0.

Program Editing Note: When you program a reversing sequencer you must select a special group number from the table above. The step data you program is a decimal notation for the reversing step order, as opposed to the coded step data for a regular group number. If you edit a sequencer and change a group number from a special group number to a regular group number or vice versa, you must also edit the step data. Failure to do this could result in unexpected sequencer operation.

Figure 11.3



CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☒ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: 0

	BIT ADDRESS DATA								PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	1	1	1	1	1	1	1	1	F	F	
Step Data → 0	0	0	0	0	0	0	0	0	0	0	1.0
1	0	0	0	0	1	0	1	0	0	A	1.0
2	0	0	1	1	0	0	0	0	3	0	1.0
3	1	0	1	0	0	0	0	0	A	0	1.0
4	0	1	0	1	0	1	0	1	5	5	1.0
5	0	0	0	1	0	0	0	1	1	1	1.0
6	0	0	1	0	0	0	1	0	2	2	1.0
7	0	1	0	0	0	1	0	0	4	4	1.0
8	0	0	1	0	1	0	0	0	2	8	1.0
9	1	0	0	0	0	0	0	1	8	1	1.0

Sequencer data form for rung 1

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☒ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: 38

	BIT ADDRESS DATA								PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →									Data B	Data A	
Mask Data →									F	F	
Step Data → 0									0	8	1.0
1									0	9	1.0
2									0	0	1.0
3									0	1	1.0
4									0	2	1.0
5									0	3	1.0
6									0	4	1.0
7									0	5	1.0
8									0	6	1.0
9									0	7	1.0

SLC Personal Computer Software Users:

When you are prompted for "Dec Data", enter the step data as done here – 08, 09, 00, 01, 02, and so on.

Sequencer data form for rung 3

Rules for step data: Step 0: Enter second last step number (8 in this case.)
Step 1: Enter last step number (9 in this case.)
Steps 2 and higher: Enter step number minus 2 (2 minus 2 = 0, 3 minus 2 = 1, etc.)

Reversing sequencer

Sequencer Input Driving a Sequencer Output

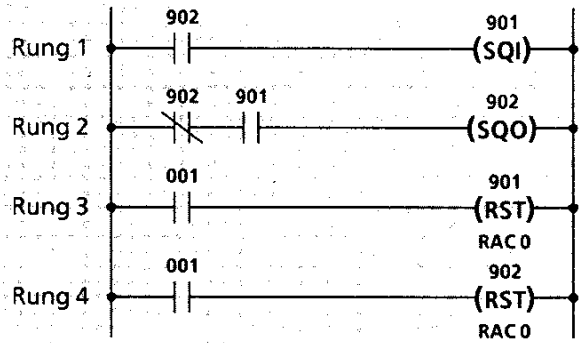
The input-satisfied status bit of an SQI instruction can be used to advance an SQO step number. Figure 11.4 shows how this is done.

The SQI and SQO instructions are assigned addresses 901 and 902. Both instructions are event-driven with a preset of 1 for each step. The input-satisfied status bit of the SQI instruction is used in rung 2 to advance the SQO step number. The SQO step completion bit is then used to advance the SQI step number.

In step 0, status bits corresponding to bit addresses 011 thru 018 are OFF, as specified in the SQO data table. Referring to step 0 of the SQI data table, note that when bit address 005 is ON and bit addresses 006, 007, and 008 are OFF, the programmed data for step 0 is matched, causing the input-satisfied status bit to go ON. This advances both the SQI and SQO instructions to step 1, causing outputs 012 and 014 to go ON. Operation continues from step to step in this manner.

Figure 11.4

SQI driving an SQO



CLASSIFICATION: ☒ -(SQI)- ☐ -(SQO)- ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 7

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	008	007	006	005	004	003	002	001	Data B	Data A	
Mask Data →	1	1	1	1	0	0	0	0	F	0	
Step Data → 0	0	0	0	1	⊗	⊗	⊗	⊗	1	0	1
1	0	0	1	0	⊗	⊗	⊗	⊗	2	0	1
2	0	1	0	0	⊗	⊗	⊗	⊗	4	0	1
3	1	0	0	0	⊗	⊗	⊗	⊗	8	0	1

⊗ Masked addresses are 0 for coding.

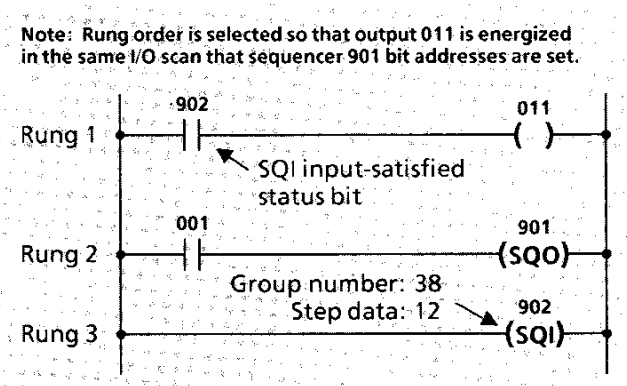
CLASSIFICATION: ☐ -(SQI)- ☒ -(SQO)- ADDRESS: 902 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	1	1	1	1	1	1	1	1	F	F	
Step Data → 0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	1	0	1	0	0	A	1
2	0	0	1	1	1	1	1	0	3	E	1
3	1	1	0	0	0	0	1	1	C	3	1

SQI Monitoring an SQO

Example 1 – SQI monitoring an SQO step. This is shown in Figure 11.5. The SQI instruction in rung 3 is monitoring the SQO instruction in rung 2. When the SQO instruction reaches step 12, the input-satisfied status bit of the SQI instruction (rung 1) is set to ON, energizing output 011.

Figure 11.5



SQI monitoring an SQO step

Program the SQO instruction as you ordinarily would. Program the SQI instruction according to the following special procedure:

1. Assign it an address between 901 and 932.
2. When prompted for time- or event-driven operation, select either one; operation will be the same regardless of your choice.
3. Select a group number from the table below. *It must correspond to the address of the SQO instruction you are monitoring.* For Figure 11.5, SQO 901 is monitored, requiring group number 38. You must program the mask data as FF (for any of group numbers 38-69).

Address	Group No.	Address	Group No.	Address	Group No.	Address	Group No.
901	**38**	909	**46**	917	**54**	925	**62**
902	**39**	910	**47**	918	**55**	926	**63**
903	**40**	911	**48**	919	**56**	927	**64**
904	**41**	912	**49**	920	**57**	928	**65**
905	**42**	913	**50**	921	**58**	929	**66**
906	**43**	914	**51**	922	**59**	930	**67**
907	**44**	915	**52**	923	**60**	931	**68**
908	**45**	916	**53**	924	**61**	932	**69**

4. The programmer will now prompt for step data. Enter the step number as data (12 for example). Any preset value can be entered.

   SLC Personal Computer Software Users: You will be prompted for "Dec Data". Enter the step number (12 for example).

   This completes the SQI instruction entry.

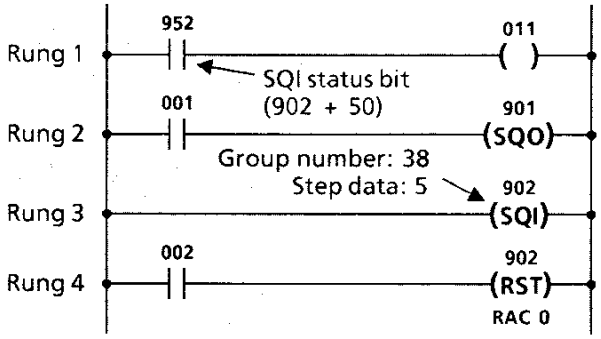
> **Program Editing Note:** When you program a sequencer input instruction to monitor another sequencer instruction, you must select a special group number from the table above. The step data you program is the step number you want to monitor as opposed to the coded step data for a regular group number. If you edit a sequencer and change a group number from a special group number to a regular group number or vice versa, you must also edit the step data. Failure to do this could result in unexpected sequencer operation.

SQI Monitoring an SQO (continued)

Example 2 – SQI monitoring an SQO for "greater than" (>) step. In Figure 11.6, SQI 902 is monitoring SQO 901. SQI 902 status bit 952 (902+ 50) is set ON when the SQO step number is greater than (>) 5. The status bit remains ON until SQI 902 is reset.

Figure 11.6

Note: Rung order is selected so that output 011 is energized in the same I/O scan that sequencer 901 bit addresses are set.



SQI monitoring an SQO for "greater than" (>) step

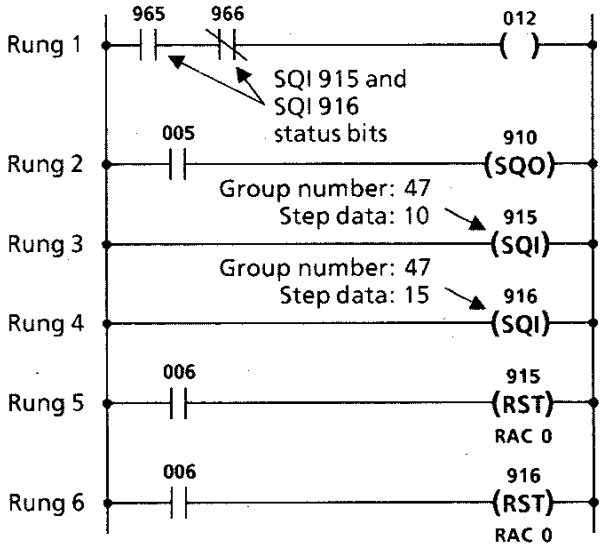
Program the SQO instruction as you ordinarily would. Program the SQI instruction according to the special procedure described in Example 1.

SQI Monitoring an SQO (continued)

Example 3 – SQI monitoring an SQO for step number range. In Figure 11.7, SQI 915 and SQI 916 are monitoring SQO 910 for a range of steps. Instruction 012 in rung 1 will be ON when the SQO 910 step is greater than 10 and less than or equal to 15. Reset instructions are used to set SQI status bits 965 and 966 back to their initial states.

Figure 11.7

Note: Rung order is selected so that output 012 is energized in the same I/O scan that sequencer 910 bit addresses are set.



SQI monitoring an SQO for step number range

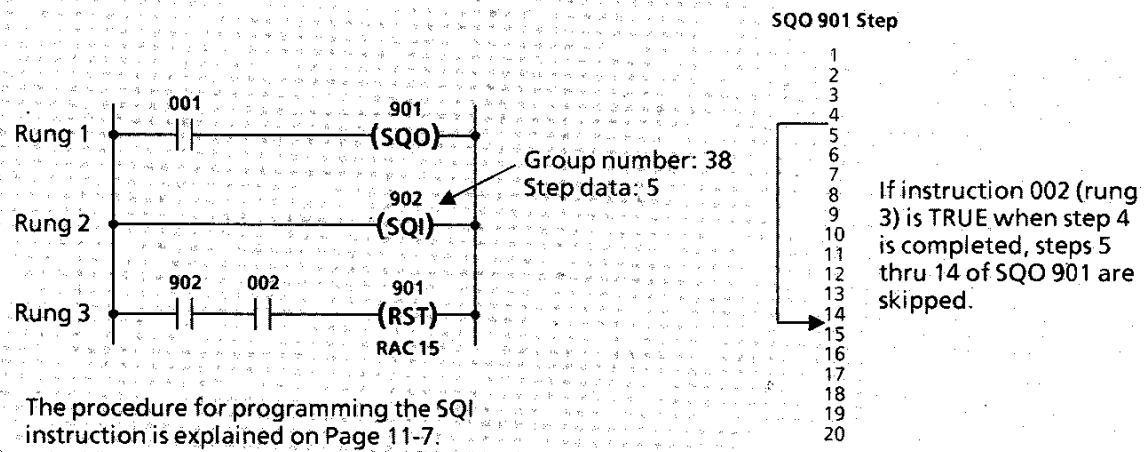
Program the SQO instruction as you ordinarily would. Program the SQI instructions according to the special procedure described in Example 1.

Sequencer Jump Operation

The jump operation allows you to either execute all steps of a sequencer instruction or to skip certain steps (jump). The sequencer input monitoring technique (Page 11-7, example 1) is used in the jump operation.

In Figure 11.8, SQI 902 is monitoring step 5 of SQO 901. If instruction 002 (rung 3) is TRUE and SQO 901 is on step 5, SQO 901 will be reset to step 15 by the RST 901 instruction. In effect, steps 5 thru 14 are skipped. On the other hand, if instruction 002 is FALSE when step 4 is completed, SQO 901 will continue with steps 5 thru 20.

Figure 11.8



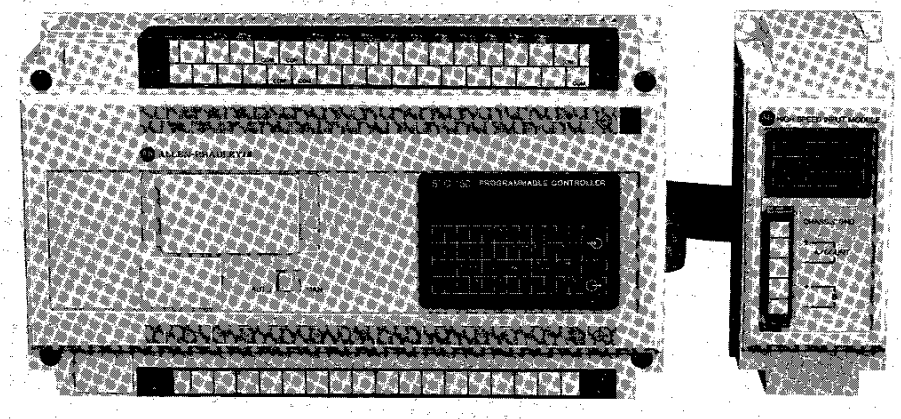
The procedure for programming the SQI instruction is explained on Page 11-7.

SLC Personal Computer Software Users: When you are prompted for "Dec Data", enter the step number, 05.

Program Editing Note: When you program a sequencer input instruction to monitor another sequencer instruction, you must select a special group number from the table on Page 11-7. The step data you program is the step number you want to monitor as opposed to the coded step data for a regular group number. If you edit a sequencer and change a group number from a special group number to a regular group number or vice versa, you must also edit the step data. Failure to do this could result in unexpected sequencer operation.

Sequencer jump operation

Chapter 12 High Speed Input Programming



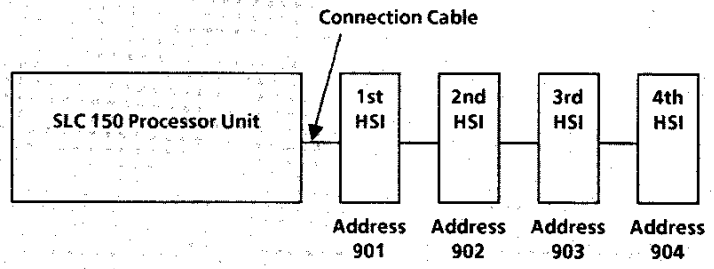
An HSI module (right) connected to an SLC 150 processor unit.
(The HSI module is for use with the SLC 150 processor only).

The High Speed Input (HSI) Module

The High Speed Input module facilitates the counting of rapid pulses produced by external input devices such as a solid state sensor or an encoder. Counting is accomplished in your program by a special event-driven SQO instruction. We will refer to this as an HSI counter instruction.

Your SLC 150 controller is designed to accommodate up to four HSI modules. Addresses for the corresponding four HSI counter instructions are 901, 902, 903, and 904. The figure below indicates how the addresses must be assigned.

Figure 12.1

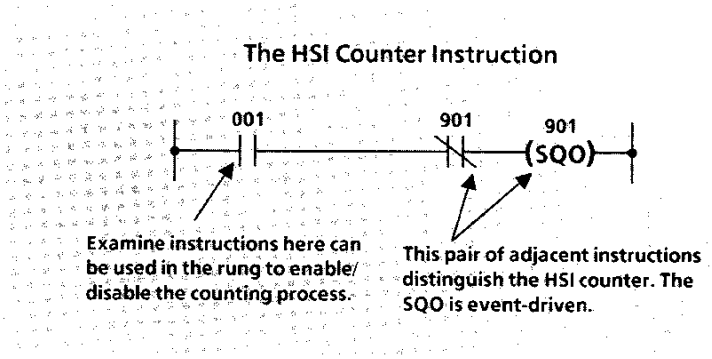


The physical location of the HSI module determines what address you must assign to it. The HSI connected directly to the processor must be assigned address 901, the HSI connected to the 1st HSI must be assigned address 902, etc. Unused addresses can be used for standard timer/counter/sequencer programming.

The HSI Counter Instruction

The HSI counter is distinguished by an event-driven SQO instruction immediately preceded by an Examine OFF instruction with the same address. Additional examine instructions can be used in the rung to enable or disable the counting process. This is illustrated below.

Figure 12.2



Operation is similar to event-driven sequencers discussed in Chapter 10. When instruction 001 in Figure 12.2 is TRUE, HSI counter 901 is enabled. Pulses fed into the HSI 901 module increase the AC value of SQO 901. When the AC value reaches the programmed PR value, the HSI counter advances to the next step, setting outputs according to programmed data.

Maximum Pulse Rate (Frequency) – If the maximum pulse rate is exceeded, Examine OFF instruction 901 (Figure 12.2) will go FALSE, disabling the HSI counter. The maximum pulse rate is 5 KHz when you are using one HSI module. If you use 2, 3, or 4 HSI modules, the maximum pulse rate is less. This is explained in the table below.

No. of modules connected	Maximum pulse rate for a single module	The sum of the pulse rates for all modules	Example
1	5 KHz	Cannot exceed 5 KHz	If you use 2 HSI modules, neither one can have a pulse rate greater than 4 KHz, and the sum of the pulse rates cannot exceed 5 KHz.
2	4 KHz		
3	3 KHz		
4	2 KHz		

Basic Operation

Figure 12.3 shows two HSI counters. Note that each counter is characterized by an event-driven SQO instruction preceded by an Examine OFF instruction with the same address.

Cycle Completion Bit (Rung 2): HSI counter 901 includes a cycle completion bit (901 + 50). Instruction 951 in rung 2 will go TRUE for one program scan when SQO 901 completes its last step.

Basic Operation
(continued)

Normal Reset (Rung 3): HSI counter 901 is reset normally. When instruction 002 goes TRUE, counting is disabled. SQO 901 is reset to step 0, applying step 0 outputs. SQO 901 remains at step 0 until rung 3 goes FALSE.

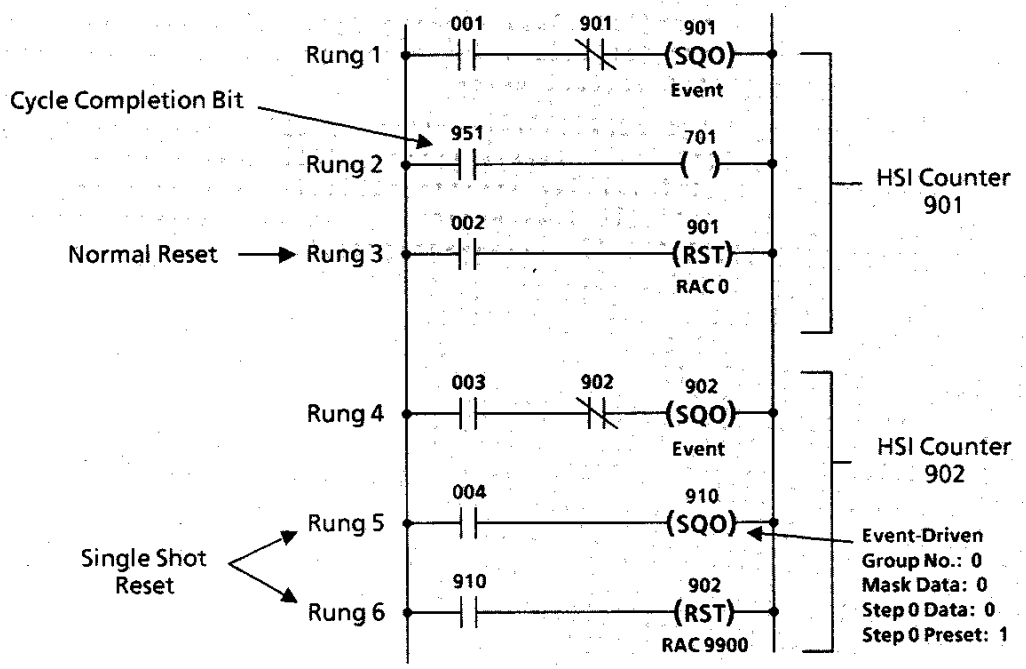
Single Shot Reset (Rungs 5 and 6): HSI counter 902 uses a single shot reset, which resets the counter to a specific step without losing any high speed input counts. The single shot is initiated by the step completion bit of an SQO instruction. Thus, when rung 5 goes TRUE, instruction 910 in rung 6 goes TRUE for one program scan. This momentarily sets rung 6 TRUE, resetting SQO 902 to step 0, where the counter resumes operation. Any high speed input pulses occurring during reset are applied to the AC value of step 0.

You must program a 4-digit RAC value for a single shot reset. The first two digits must be 99; the second two digits specify the step number you want the counter reset to.

Exceeding the Maximum Pulse Rate: If the maximum pulse rate is exceeded, Examine OFF instructions 901 and 902 (rungs 1 and 4) will go FALSE, disabling the counting process. The counters are held in their last states, with respective outputs applied. Instructions 901 and 902 remain FALSE until the over-speed condition is corrected and the counters are reset.

> **Caution:** Exceeding the maximum pulse rate will disable the HSI counter and hold outputs to the last state. Appropriate emergency stop switches should be used in your application to guard against unexpected machine operation.

Figure 12.3



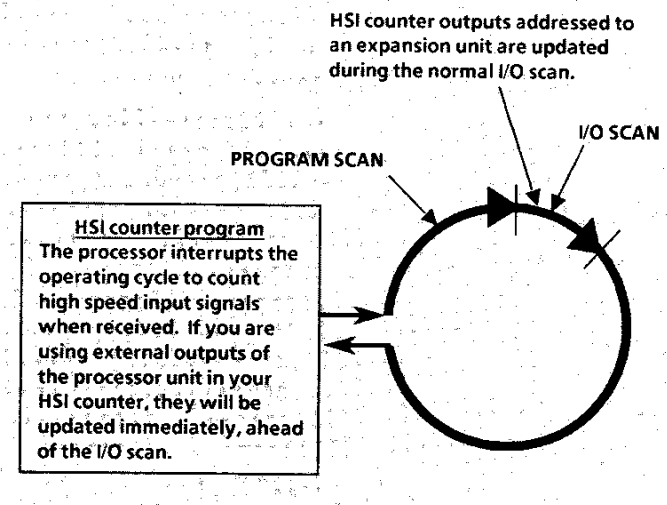
Special Application Considerations

Cascading: You cannot cascade sequencers used in HSI counters. Your HSI counter is limited to 8 bit addresses, 6 of which can be external outputs.

External Outputs: To obtain the fastest possible operation of your HSI counter, use the external outputs located on the processor unit. These outputs, corresponding to SQO sequencer group numbers 0 and 1, are updated *immediately*, ahead of the I/O scan. See Figure 12.4.

You can also use external outputs located on expansion units. These are updated during the normal I/O scan.

Figure 12.4



Scan Time: As the frequency of your HSI counter increases, your scan time also increases. Programming preset values of 1will further increase your scan time.

A scan time in excess of 100 milliseconds will be detected by the processor unit and cause system shutdown (error code 9). You should verify your scan time at the maximum HSI input rate for your application, to assure proper operation of your system. Refer to Page 8-4 for instructions on measuring scan time.

Special Application Considerations (continued)

Test Single Scan Mode: When the SLC 150 processor unit is placed in the Test-Single Scan mode and the HSI counter start rung is TRUE, the HSI will count in real time and the accumulator will increment as it would in the Run mode.

Disconnecting the HSI Module: The HSI module should not be connected or disconnected when power is applied to the SLC 150 processor unit.

> **WARNING:** Improper operation and damage to equipment or personnel could occur if the HSI module is connected or disconnected when power is applied to the SLC 150 processor unit.

Expansion Socket End Plug: The HSI module has an expansion socket for the purpose of adding another HSI module or expansion unit to your system. The HSI module is shipped with an end plug inserted in this expansion socket. Do not remove this plug unless you are adding another HSI module or an expansion unit to your system. Removing the plug will cause a CPU fault, shutting down the processor in the Run mode.

Application Example – Two Step Cut Off Operation

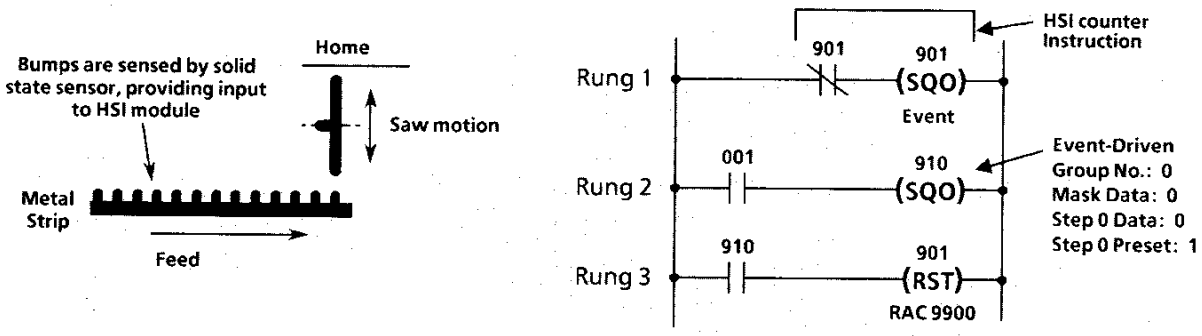
A high speed cut-off operation measures metal strips by counting preformed bumps as the metal strip is fed into the cut-off area. The bumps pass a solid state sensor at the rate of 1000 bumps per second. Every 2000 bumps, the strip is cut. The counting process then resumes.

In the program below, an HSI counter is used to perform the counting and cut-off operation. SQO 901 outputs at addresses 011 and 012 control the feed motor and the cut-off saw motor. The counter is enabled only when the cut-off saw is in the home position.

In step 0, the feed motor is energized, which feeds the metal strip. As the strip is fed, the bumps are counted by the HSI counter. When 2000 counts are reached, SQO 901 advances to step 1. This lowers the cut-off saw which cuts the strip and returns to the home position.

When the cut-off saw returns to its home position a limit switch at address 001 is closed, initiating a single shot reset that resets the process back to step 0. This single shot reset allows the feeding process to resume and high speed input signals to be counted immediately. The preset of 9999 for step 1 has been arbitrarily chosen and allows for overfeeding of the metal strip. The preset for step 1 could be set to any four digit number except zero since the process is reset when the saw returns home.

Figure 12.5



CLASSIFICATION: ☐ -(SQI)- ☒ -(SQO)- ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	0	0	0	0	0	0	1	1	0	3	
Step Data → 0	⊗	⊗	⊗	⊗	⊗	⊗	0	1	0	1	2 0 0 0
1	⊗	⊗	⊗	⊗	⊗	⊗	1	0	0	2	9 9 9 9

⊗ Masked addresses

Output address 011 controls feed

Output address 012 controls saw motion

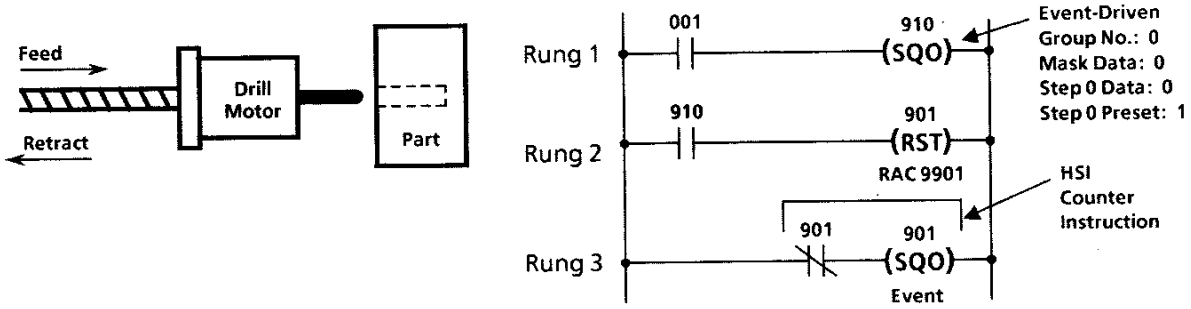
Application Example – Multiple Step Position Control

Drill position and hole depth are controlled by counting high speed input pulses from an encoder mounted on a feed motor. The feed motor turns a lead screw which moves the drill forward and backward. As the feed motor rotates, encoder pulses are fed into the HSI module at a rate of 5000 pulses per second.

Step 0 is the home position, with all outputs off. The preset of 9999 for step 0 is arbitrarily chosen and allows for encoder movement without advancing the sequencer step number. A push button at address 001 initiates the process by performing a single shot reset which moves the process to step 1 where the feed motor is energized. Encoder pulses are then counted by the HSI counter. After 9000 counts, the drill motor is turned on. After another 2000 pulses, the drill bit reaches its proper depth and the sequencer advances to step 3. Step 3 reverses the feed motor, retracting the drill. After 2000 pulses the drill motor is turned off. The drill continues to retract to its original position.

When the cycle is complete, the process returns to step 0 which is the home position with all outputs off.

Figure 12.6



CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	0	0	0	0	0	1	1	1	0	7	
Step Data → 0	⊗	⊗	⊗	⊗	⊗	0	0	0	0	0	9 9 9 9
1	⊗	⊗	⊗	⊗	⊗	0	0	1	0	1	9 0 0 0
2	⊗	⊗	⊗	⊗	⊗	1	0	1	0	5	2 0 0 0
3	⊗	⊗	⊗	⊗	⊗	1	1	0	0	6	2 0 0 0
4	⊗	⊗	⊗	⊗	⊗	0	1	0	0	2	9 0 0 0

⊗ Masked addresses

Output address 011 controls feed motor – forward

Output address 012 controls feed motor – reverse

Output address 013 controls drill motor

Chapter 13 Shift Register Instruction

General

With the shift register instruction, status data enters an 8-bit register and is automatically shifted thru the register from one bit address to the next on a time- or event-driven basis. A typical application is with conveyors, to monitor and control the flow of the individual parts. In general, it can be used in the control of machines or processes where parts are continually shifted from one position to the next.

You can program a shift register to shift status data either right or left. In operation, the shift register instruction is executed each program scan, and uses a ZCL zone to control the shifting process. A sequencer instruction is used to control the shift rate. The unlatch instruction can be used to set status bits from ON to OFF.

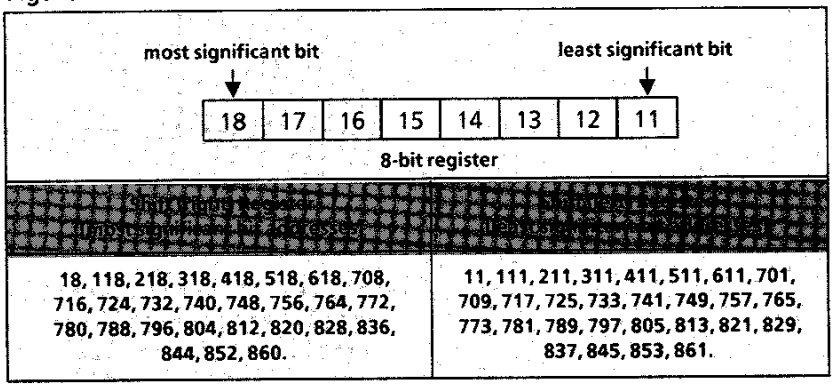
The shift register instruction is retentive. The ON/OFF status and the position of data is retained if power to the processor unit is removed or if the processor unit is switched from the Run mode to another mode.

Programming the Shift Register Instruction

Acceptable addresses for shift registers are shown below. A shift right register is assigned the address of its most significant bit. A shift left register is assigned the address of its least significant bit.

Valid addresses include external output addresses and internal addresses. Internal addresses can be used within your program or used to obtain outputs, as required.

Figure 13.1



The shift register instruction is represented by the symbol –(SR)–. To program this instruction, press SHIFT, –( )–, then the appropriate numeric keys for the address (selected from the table above).

Display symbol for a shift right register: ⌐

Display symbol for a shift left register: L

Display example: When the cursor is on a shift right register instruction, address 018, rung 3, the output energize LED will be lit and the display will show: ⌐ 18 3

Display example: When the cursor is on a shift left register instruction, address 011, rung 4, the output energize LED will be lit and the display will show: L 11 4

Event-Driven Shift Registers

Figure 13.2 is a ladder diagram of an event-driven shift right register. indicated in the figure. Rungs 2-3-4 consist of a shift right register within a ZCL zone.

Each time instruction 001 in rung 1 goes from FALSE to TRUE, step 0 of the sequencer is completed (AC = PR) and the ZCL zone is TRUE for one program scan. If instruction 002 is TRUE during this scan, rung 3 is TRUE, and a 1 is shifted into the most significant bit address (18) of the register; if instruction 002 is FALSE, a 0 is shifted into the register. Simultaneously, all other status data in the register is shifted right to the next address, with status data of bit 11 being shifted out of the register:

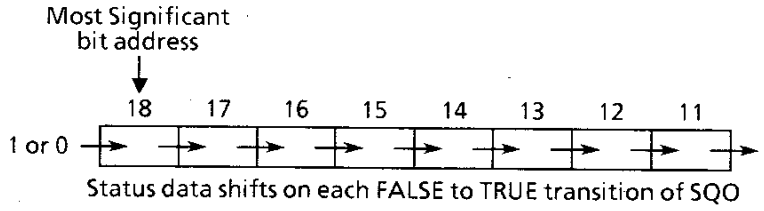


Figure 13.2

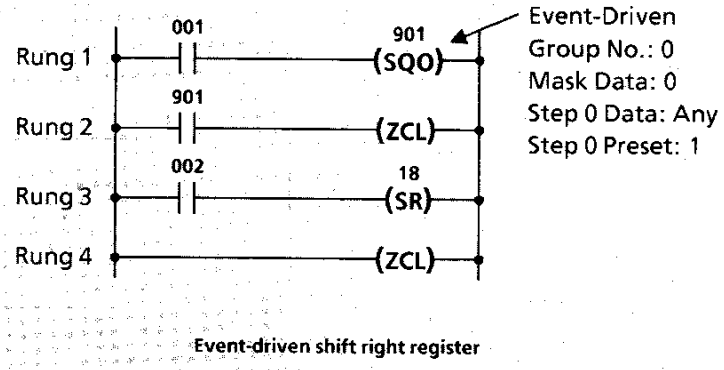
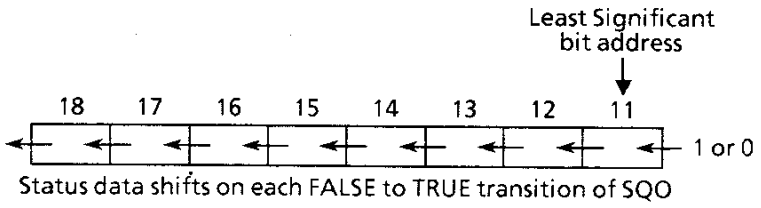


Figure 13.2 could be changed to an event-driven shift left register simply by assigning address 11 instead of 18 to the shift register instruction in rung 3. Operation would be similar, with status data shifted into the *least* significant bit address, and status data of bit 18 shifted out of the register:



Time-Driven Shift Registers

Figure 13.3 is a ladder diagram of a time-driven shift right register. Note the similarity to the event-driven shift register of Figure 13.2.

Rung 1 is a 1-step, time-driven sequencer. Programmed data is indicated in the figure; set the PR value to the desired shift rate. Rungs 2-3-4 consist of a shift right register within a ZCL zone.

Step 0 of the sequencer is completed each time the AC value reaches the PR value. The ZCL zone is then TRUE for one program scan. If instruction 001 is TRUE during this scan, rung 3 is TRUE, and a 1 is shifted into the most significant bit address (18) of the register; if instruction 001 is FALSE, a 0 is shifted into the register. Simultaneously, all other status data in the register is shifted right to the next address, with status data of bit 11 being shifted out of the register:

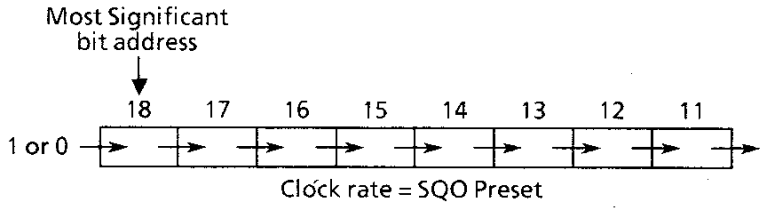
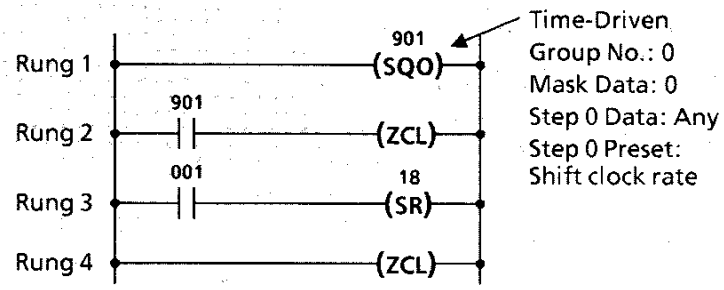


Figure 13.3

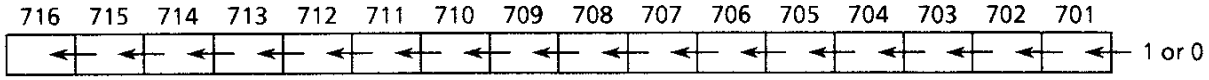


Time-driven shift right register

Figure 13.3 could be changed to a time-driven shift left register simply by assigning address 11 instead of 18 to the shift register instruction in rung 3. Operation would be similar, with status data shifted into the *least* significant bit address, and status data of bit 18 shifted out of the register:

Cascading Shift Registers

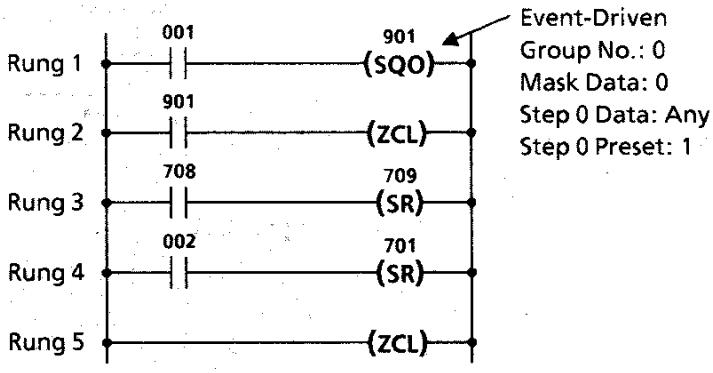
You can cascade shift register instructions in order to shift more than 8 bits of data. In the 16-bit shift left register of Figure 13.4, status data is entered into shift register 701. Bit 708 of this register shifts data into register 709:



Status data shifts on each FALSE to TRUE transition of SQO

Important: The rung containing the shift register into which data is entered must follow the rung containing the second shift register. Thus, shift register 701 (rung 4) follows shift register 709 (rung 3).

Figure 13.4



A 16-bit event-driven shift left register

Circulating Shift Register

Figure 13.5 is an example of a circulating time-driven shift left register. Status data is entered by instruction 001 (rung 3) and shifted left from bit address 701 to bit address 708; bit address 708 is used in rung 3 to shift the status data back into bit address 701, and the cycle is repeated:

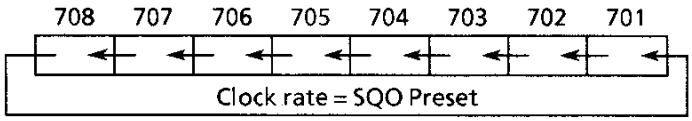
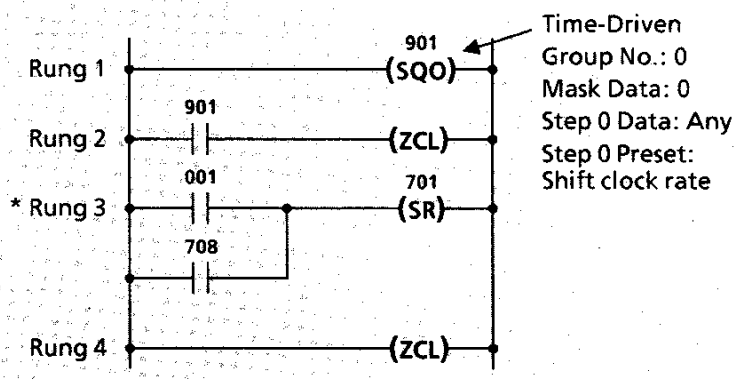


Figure 13.5



* Data circulated from bit address 708 to bit address 701 would be overridden by data shifted into bit address 701 by Examine ON instruction 001.

An 8-bit circulating shift register

Example: Using Shift Register Outputs

Rungs 6 thru 9 in Figure 13.6 show how you can use the various bit addresses within the shift register to produce external outputs. Note that these rungs are placed outside the ZCL zone.

Status data is entered by instruction 002 in rung 3. The 1's entered into the circulating register can be changed to 0's with instruction 003 in rung 5, which unlatches bit address 701.

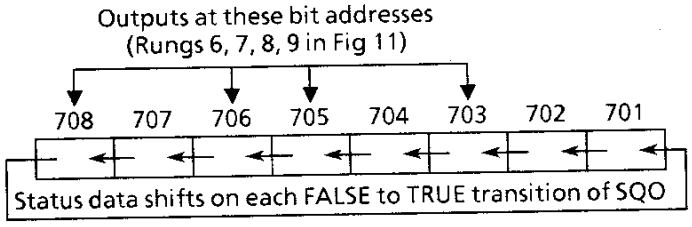
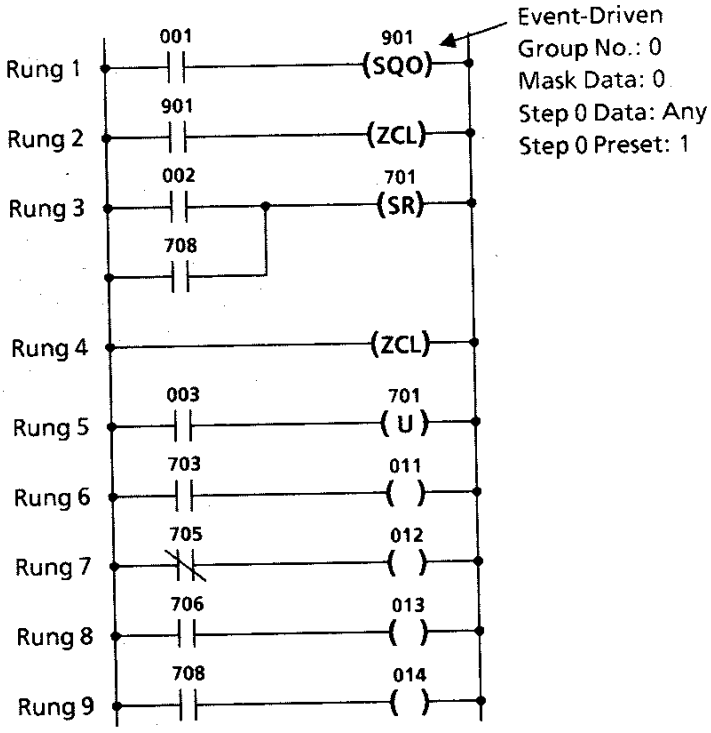


Figure 13.6



Circulating shift left register using outputs for bit addresses 703, 705, 706, and 708

Chapter 14 Special Internal Instructions

General

This chapter explains how to use the following special internal instructions:

Processor	Address	Function
SLC 100 and SLC 150	876	Auto/Man switch. This address bit is ON when the Auto/Man switch of the processor unit is in the Auto position.
	868	Program initialization instruction. Used with timers, counters, and latch instructions to set initial conditions on power-up.
	867	TCAT power-up. Used with the Timer Counter Access Terminal to display a selected timer/counter/sequencer address on power-up.
	100	Breakpoint instruction. Used for debugging and trouble-shooting. When a rung containing this instruction is TRUE, the processor will automatically halt its program scan, switch from the Run to the Test-Single Scan mode, and turn all outputs off.
SLC 150 only	864	Programmable EEPROM auto-load. Used to automatically load the EEPROM program in the processor RAM when the processor detects a memory checksum error (capacitor or battery back-up drain, or processor malfunction).
	865	Battery status. This bit is ON when the battery voltage is normal. The bit is set OFF when battery voltage drops below a certain threshold. If a battery is not used, the bit remains ON.
	866	Triac zero-cross turn-on. Output instruction, used to synchronize triac outputs with the AC line.

Auto-Man Switch – Address 876

This instruction allows you to examine the status of the Auto/Man switch of the processor:

TRUE when the switch is in the Auto position

FALSE when the switch is in the Auto position

Program Initialization – Address 868

This is an internal examine instruction you can use to initialize your program to a known state on power-up. It can be used to initialize timers, counters, sequencers, and latch instructions.

The bit address of the instruction is 868. This bit is set ON for the first *and only the first* program scan under either of the following conditions:

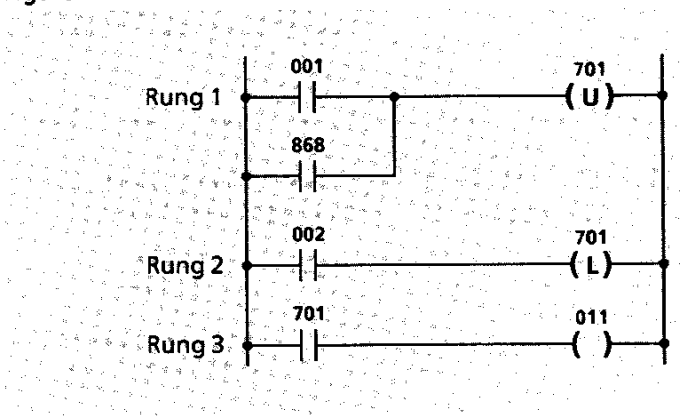
1. Power is switched on to the processor unit in the Run or Test modes.
2. The programmer is used to place the processor unit in the Run or Test modes.

The following examples show you how to use the program initialization instruction to program non-retentive timers and a non-retentive latch. You will also be shown how to initialize an up-down counter and a latch instruction. Always use instruction 868 in the first rung of your program so that program initialization will occur properly.

Non-Retentive Latch

A ladder diagram for a non-retentive latch instruction is shown in Figure 14.1. On power-up, instruction 868 in rung 1 will go TRUE for the first scan, so that instruction 701 will be unlatched initially. Examine ON instruction 001 in rung 1 is used to unlatch instruction 701 during program execution.

Figure 14.1

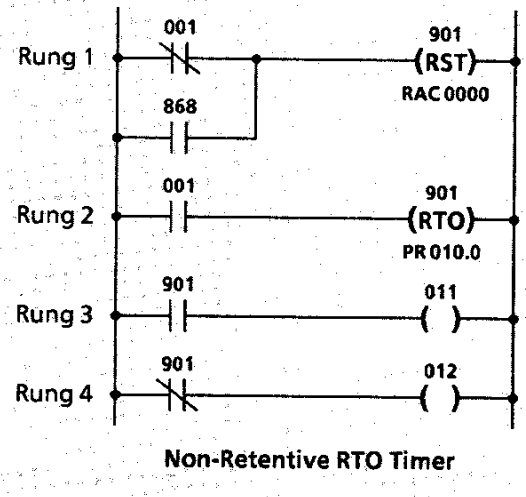


Non-Retentive Timers

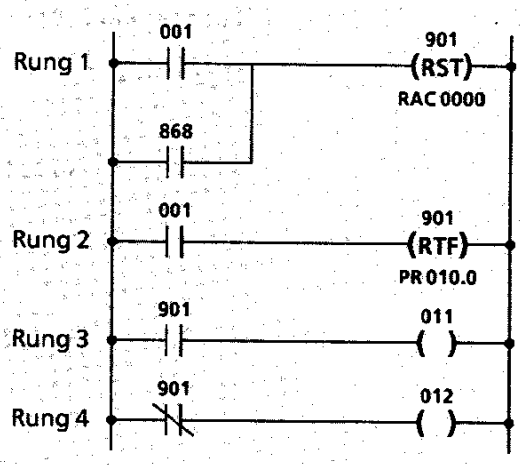
Figure 14.2 shows a non-retentive RTO timer. On power-up, instruction 868 in rung 1 will be TRUE for the first scan, so that the timer will be reset initially. Note that rung 1 also contains an Examine OFF instruction having the same address (001) as the Examine ON instruction in rung 2. This Examine OFF instruction resets the timer when rung 2 goes FALSE.

The figure also illustrates a non-retentive RTF timer. In this case, an Examine ON instruction (001) is used in the reset rung (1). Rung order is important with the non-retentive RTF timer. *The RST rung must precede the RTF rung.*

Figure 14.2



Non-Retentive RTO Timer

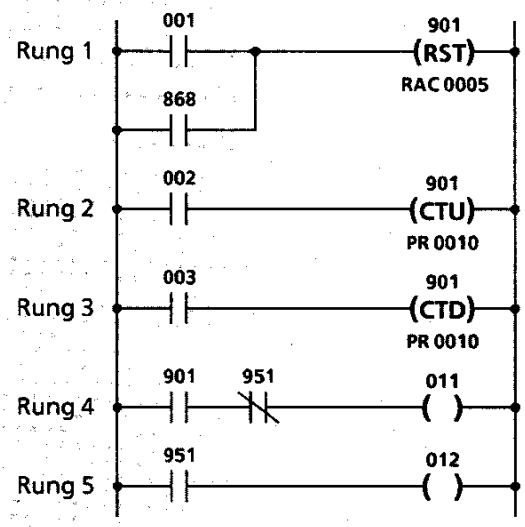


Non-Retentive RTF Timer

Initializing an Up-Down Counter

The figure below shows how you can initialize an up-down counter. On power-up, instruction 868 in rung 1 will go TRUE for the first scan, resetting the counter to 5, which is between the upper limit of 10 and the lower limit of 0.

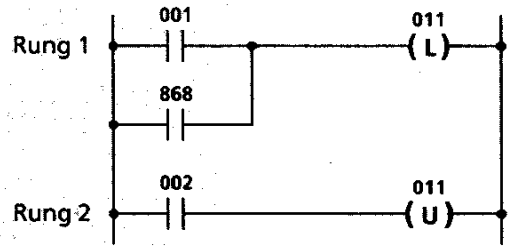
Figure 14.3



Initializing a Latch Instruction

The figure below shows how you can initialize a latch instruction. On power-up, instruction 868 in rung 1 will go TRUE for the first scan, latching output 011.

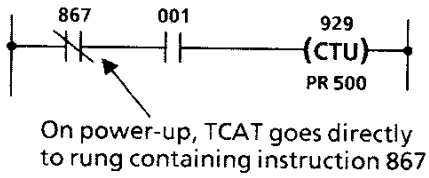
Figure 14.4



TCAT Power-Up – Address 867

If you are using a TCAT (Timer Counter Access Terminal), you can use this Examine OFF instruction to automatically display any timer/counter/sequencer address you choose on power-up (instead of scanning the timer/counter/sequencer addresses). An example is shown below.

Figure 14.5



PROGRAMMING NOTE – Instruction 867 should be an Examine OFF, programmed in series with the rung containing the instruction you want displayed on power-up. (You can program instruction 867 anywhere in the rung, to the left of the timer/counter/sequencer instruction.)

Breakpoint Instruction – Address 100

This is an output instruction at address 100. You can use it to pin-point suspected problem areas in your program during debugging and troubleshooting procedures.

When the rung containing the breakpoint output instruction is TRUE, the processor will halt the program scan, automatically switch from the Run mode to the Test-Single Scan mode, and turn all outputs OFF.

Use the breakpoint instruction as follows:

1. Decide on what point in your program you should insert the breakpoint instruction. This should be directly after the part of your ladder program that you suspect is not operating correctly.
2. Program the breakpoint rung. The conditions that make the breakpoint output instruction TRUE should be the conditions that you suspect would cause your program to malfunction. An example is shown on Page 14-6.
3. Place the processor in the Run mode and test your program. If the processor halts and outputs are disabled, the breakpoint output rung is TRUE.
4. Connect the pocket programmer. It will automatically identify the breakpoint rung number. (If you are using the SLC Personal Computer Software, it will display a message telling you that a breakpoint -(BPT)- has occurred at a specific rung number.)
5. Correct the program malfunction and delete the breakpoint rung.

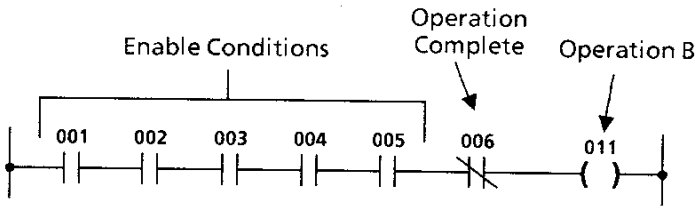
A breakpoint output rung can be programmed as many times as you need it in your program.

Important: **Do not** use the breakpoint output instruction for emergency stop situations. We strongly recommend that you use a hard-wired master control relay to provide emergency I/O power shutdown.

Breakpoint Instruction – Address 100 *(continued)*

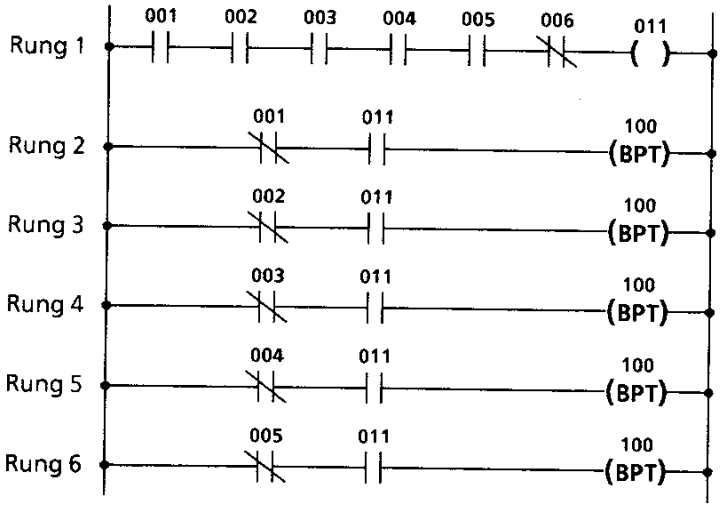
An example of how you might use the breakpoint instruction:

Operation B: Refer to the program below. When a series of five external inputs (represented in the program by instructions 001 thru 005) are closed, output instruction 011 is energized, causing assembly operation B to be performed. When assembly B is completed, instruction 006 goes FALSE, de-energizing output 011. The external inputs are then opened and the next operation in the assembly process is initiated.



The Problem: Occasionally, operation B is not performed because one of the external inputs does not remain closed for the duration of the assembly process. Determining which of the five external inputs is at fault is a difficult and time-consuming troubleshooting procedure.

The Solution: Breakpoint rungs 2 thru 6 are inserted into the user program:



During operation B, rung 1 will be TRUE and the breakpoint rungs will be FALSE. If an external input opens before assembly operation B is completed, one of the breakpoint rungs will go TRUE. This will cause the processor to switch from the Run mode to the Test Single Scan mode, disabling all outputs. The pocket programmer will identify the breakpoint rung number, revealing which external input is causing the problem.

If operation B is completed without a problem, program execution continues as normal.

Programmable EEPROM Auto-Load – SLC 150 Only

With the SLC 150 processor unit, you can use address 864 in your program to automatically load the EEPROM program in the processor RAM when the processor detects a memory checksum error. Such errors are detected if memory data is altered or lost due to capacitor back-up drain, battery back-up drain, or processor malfunction. It works like this:

1. Enter the following unconditional rung as Rung 1 of your program:

2. Store the program in the EEPROM. This is your "back-up" program.
3. If a memory problem occurs and the processor unit detects a memory error (codes 5 thru 8), recycle power. The EEPROM program will be automatically loaded in the processor RAM, clearing the error.

Refer to Chapter 17 for further information on the Auto-Load Procedure.

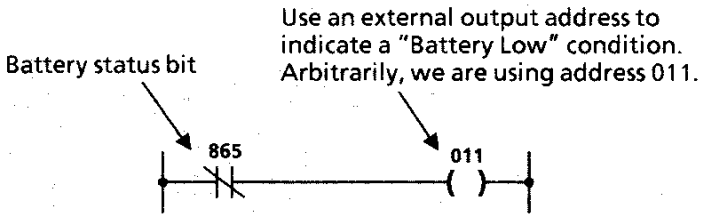
Important: Keep in mind that when the auto-load occurs, the I/O table data is cleared. All retentive instructions (latches, timers, counters, sequencers) will be reset to their initial states.

Battery Status – SLC 150 Only

If you are using the battery back-up option with the SLC 150, you can use the status bit at address 865 to energize a "Battery Low" indicator when the battery voltage falls below a certain threshold. This threshold voltage level will support the processor memory for approximately 2 weeks.

The figure below indicates how you might use this battery status instruction. Battery low indication is provided only when power is applied to the processor unit.

Figure 14.6



Rung is FALSE when battery voltage is normal. Rung goes TRUE when battery voltage drops below a certain threshold.

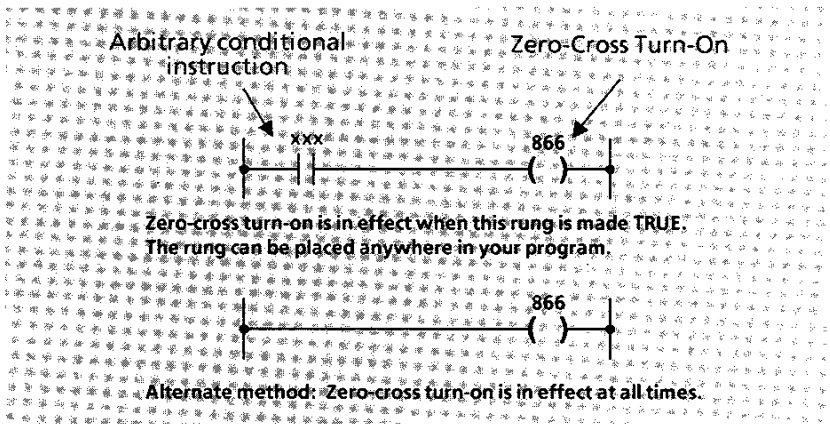
Triac Zero-Cross Turn-On – SLC 150 Only

This is an output energize instruction at address 866. It is used to synchronize triac outputs with the AC line to help minimize noise generated when switching loads. To achieve zero-cross turn-on, you must use a common power source for the processor unit power supply and the output circuits.

The figure below indicates how to use the triac zero-cross turn-on instruction. NOTE: When this instruction is in effect, your program scan will be synchronized with the AC line zero-cross. This means that your scan time will be 8.3 msec (or some multiple of 8.3 msec) at 60 Hz, and 10 msec (or some multiple of 10 msec) at 50 Hz.

> **Caution:** Make certain that the scan time resulting from using the zero-cross turn-on instruction will not adversely affect your program execution. This is especially important if you are using any of the fine time base instructions discussed in Chapter 8.

Figure 14.7



Chapter 15 Program Editing

General

In the following paragraphs, you will learn how to edit your program using the NEXT and LAST keys (cursor control), the SEARCH key, and the REMOVE and INSERT keys. The following keystroke examples do not cover all of the editing techniques. Refer to Pages 15-10, 15-11, and 15-12 for a comprehensive list.

Program Editing with the TCAT and PC Software: Editing techniques possible with the TCAT (Timer Counter Access Terminal) and the SLC Personal Computer Software are described in the respective User's Manuals, Publications 1745-850 and 1745-825.

Cursor Control

The programmer indicates the cursor location by the particular instruction LED lit, the address number, and the rung number. The cursor location is moved by pressing the NEXT key or the LAST key.

Each time you press the NEXT key, the cursor moves to the following instruction, that is, one instruction to the right. Each time you press the LAST key, the cursor moves to the preceding instruction, one instruction to the left. You can use this method to move from instruction to instruction thru the rungs of your program.

When the cursor is positioned on a timer, counter or sequencer instruction, pressing the NEXT key displays the data associated with that instruction (PR value, AC value, and other data), in the order in which it was entered.

You can move the cursor to the start of the next rung by pressing RUNG, NEXT, or to the start of the preceding rung by pressing RUNG, LAST.

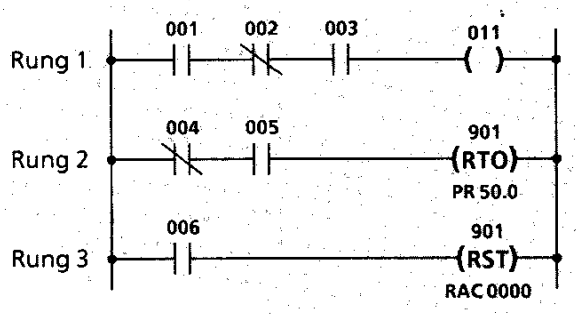
The keystroke example of Figure 15.1 (following page) illustrates use of the NEXT and LAST keys.

Keystroke Example – Cursor Control

Cursor control was explained on Page 15-2. The keystroke example below will show you how to use the NEXT and LAST keys to move the cursor thru the program and access data.

Figure 15.1

Keystroke Example – Program the ladder diagram below.



Ladder Diagram

LAST 7 times	-] [-	3 1	You have located the cursor on the Examine ON instruction in rung 1.
NEXT 2 times	-]/[-	4 2	The cursor is now on the Examine OFF instruction in rung 2. The start rung symbol shown at the left in the display indicates this is the first instruction in the rung.
RUNG, NEXT	-] [-	6 3	You have moved the cursor to the first instruction in rung 3.
RUNG, LAST	-]/[-	4 2	The cursor has returned to the first instruction in rung 2.
NEXT 2 times	-(RTO)-	901 2	You have moved the cursor to the RTO instruction in rung 2.
NEXT	-(RTO)-	50.0 Pr	The display is showing the PR value you entered earlier.
NEXT	-] [-	6 3	The cursor has moved to the next instruction in your program.
LAST	-(RTO)-	901 2	The cursor has moved back to the RTO instruction. Notice that you did not return to the PR value. You must use the NEXT key to access data.

Program editing techniques are summarized on Pages 15-10, 15-11, 15-12 .

Search Function

The search function can be used to locate:

1. A specific instruction.
2. A specific rung number.
3. The start or end of a program.
4. The start or end of a rung.
5. A specific sequencer step.

When searching for a specific instruction, the search begins at the cursor location; you can choose the start of the program, or any other point in the program. When the instruction is found, press SEARCH, ENTER to find the next occurrence of the instruction.

To move the cursor to a specific rung, you press SEARCH, RUNG, then the rung number. Specifying rung number 1 will locate the cursor at the beginning of the program; rung number 999 will locate the cursor at the end of the program.

To move to the start of a rung, press SEARCH, LAST. To move to the end of the rung, press SEARCH, NEXT.

To move to a specific sequencer step, locate the cursor on any sequencer data, then press SEARCH, the step number, ENTER.

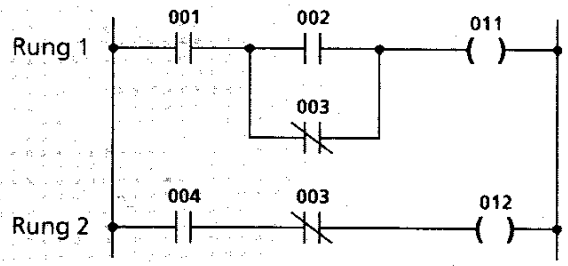
The keystroke example of Figure 15.2 illustrates the use of the SEARCH key.

Keystroke Example – Search Function

The search function was explained on Page 15-4. The keystroke example below will show you how to use the SEARCH key to find a specific instruction, the start of the program, and the end of the program.

Figure 15.2

Keystroke Example – Program the ladder diagram below.



Ladder Diagram

SEARCH		Srch	You are in a search operation.
–] / [–	–] / [–	___	An Examine OFF instruction is specified. Dashes are prompting you for an address.
3	–] / [–	__3	Address 003 is entered.
ENTER	–] / [–	3 1	Search begins at cursor location. The rung number is displayed when the instruction is found.
SEARCH, ENTER	–] / [–	3 2	Pressing SEARCH, ENTER will locate the next occurrence of the specified instruction.
SEARCH		Srch	You have initiated a new search operation.
RUNG, 1		__1	You are searching for rung 1, the start of your program.
ENTER	–] [–	⊢ 1 1	Search is completed. The cursor is on the first instruction of rung 1.
SEARCH, RUNG, 9, 9, 9		999	To find the end of the program, you enter a rung number which is certain to exceed the number of rungs in your program. 999 is a good choice.
ENTER		875 End	The cursor is at the end of your program.

Program editing techniques are summarized on Pages 15-10, 15-11, 15-12.

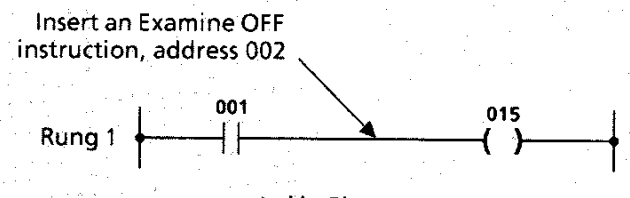
Insert and Remove Functions

While in the Program mode, you can insert and remove instructions, branches, rungs, or sequencer steps by using the INSERT and REMOVE keys. The keys also allow you to copy a rung and insert it anywhere in your program. These editing techniques are summarized on Pages 15-11 and 15-12.

The keystroke examples of Figures 15.3, 15.4, and 15.5 illustrate how to insert and remove instructions, branches, and rungs.

Figure 15.3

Keystroke Example – Program the ladder diagram below. The keystroke example will show you how to insert and remove an instruction.



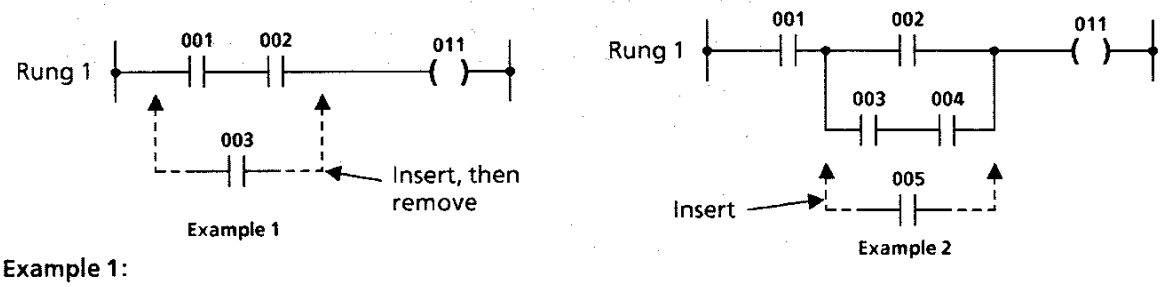
Ladder Diagram

LAST	–( )–	15	The cursor is positioned on the instruction following the point where the new instruction is to be.
INSERT		In	Display is prompting you to insert an instruction.
–]/[–	–]/[–	___	Prompt for an address.
2	–]/[–	__2	Address 002 is entered.
ENTER	–( )–	15	The instruction has been entered. The cursor has moved to the next instruction.
LAST	–]/[–	2	Cursor is positioned on the newly entered instruction. Now we'll remove it.
REMOVE	–]/[–	rE	Display indicates the remove function.
ENTER	–( )–	15	The instruction has been removed. The cursor has moved to the next instruction.
LAST	–] [–	⊢	The cursor has moved to the first instruction in the rung, verifying the removal of the Examine OFF instruction.

Program editing techniques are summarized on Pages 15-10, 15-11, 15-12.

Figure 15.4

Keystroke Example – Program these rungs. The keystroke example will show you how to insert and remove branches.



Example 1:

Press these keys	This LED will light	Display will show	Explanation
LAST 3 times	-] [-	⊢ | |	We must first insert a branch open instruction preceding instruction 001. Cursor is on instruction 001.
INSERT, ⊤	⊤	bro |	Display indicates **in**, Insert, then **bro**, branch open.
ENTER	-] [-	| |	Branch open instruction entered. Cursor moves to next instruction.
NEXT, NEXT	-( )-	| |	Cursor is now on the instruction following the point where the new branch will be inserted.
INSERT, ⊤	⊤	bro |	The branch open instruction is entered. We can now enter instruction 003. **Note:** Any number of examine instructions could be entered without pressing ENTER after each one.
-] [-, 3	-] [-	__3 |	Examine ON, address 003 has been entered.
⊥	⊥	bnd |	To complete the branch, we've pressed the branch close key. Display **bnd** indicates branch end.
ENTER	-( )-	| |	Cursor has moved to the instruction following the branch close.
LAST, LAST	-] [-	3 |	Cursor is on the examine instruction in the branch we just entered. We will now remove this branch.
REMOVE, ENTER	⊥	bnd |	The examine instruction is removed. Cursor is on the branch close. This instruction and the 2 branch open instructions in the program must be removed. We can do this very simply, with 2 keystrokes.
MODE	⊥	rE P E64	First, we attempt to move off the rung. The display indicates a branch error E64 (short circuit). It is asking whether we want to remove it.
ENTER		prog 2	Next, we press ENTER, which removes the branch instructions. We could also have removed the 3 branch instructions individually.

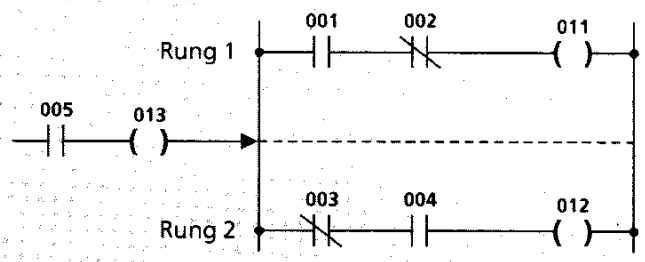
Example 2:

Press these keys	This LED will light	Display will show	Explanation
LAST, LAST	⊥	bnd |	In this case, the instruction beginning the new branch (003) is already preceded by a branch open instruction. Begin by moving the cursor to the instruction following the point where the branch will be inserted.
INSERT, ⊤	⊤	bro |	Display indicates branch open. We can now enter instruction 005.
-] [-, 5	-] [-	__5 |	Examine ON, address 005 has been entered.
ENTER	⊥	bnd |	New branch is entered. Cursor moves to the next instruction.

Insert and Remove Functions *(continued)*

Figure 15.5

Keystroke Example – Program the ladder diagram below. The keystroke example will show you how to insert and remove a rung.



Ladder Diagram

Keystroke	Instruction Cursor Is On	Display will show	Explanation
LAST	-( )-	12 2	You can move the cursor to any instruction in the rung which follows the location where you will insert the new rung. The cursor is located on the output instruction in rung 2.
INSERT, RUNG		In r	The display is prompting you to insert a rung.
-] [-, **5**	-] [-	--5 2	Examine ON instruction is inserted. This will be rung 2, as indicated.
-( )-, **1, 3**	-( )-	-13 2	The output instruction is entered.
NEXT	-]/[-	H 3 3	Cursor has moved to the first instruction in rung 3. This was formerly rung 2.
LAST	-( )-	13 2	Cursor is back to the output instruction of the newly entered rung. We'll remove this rung.
REMOVE, RUNG	-( )-	rE r	Display is prompting you to remove a rung.
ENTER	-]/[-	H 3 2	Rung is removed. Cursor has moved to the following rung, which is now rung 2 again.

Program editing techniques are summarized on Pages 15-10, 15-11, 15-12.

Changing Instructions in the Program Mode

With the CANCEL CMD key, you can easily change an instruction, address, or data while the cursor is still located on the instruction.

You can also change an instruction, address, or data after a rung or program is completed. Just locate the cursor on the instruction to be changed, then enter a new instruction, address, and data. Refer to the program editing summary on Pages 15-11 and 15-12.

Protecting the PR Value of Timers, Counters, and Sequencers

The PRT key allows you to protect the PR value. This prevents anyone from changing the PR value while the processor is in the Run or Test modes. In the case of timers and counters, neither the PR value nor the AC value can be changed while the processor is in the Run or Test modes.

You can change a PR value in the Program mode even if it is protected. But if you change a protected PR value, you must protect it again after changing it. Refer to the program editing techniques on Page 15-12.

Deleting an Entire Program

Selecting mode 1, Clear Memory, allows you to delete your entire program. Refer to the program editing summary on Page 15-11.

Program Editing Techniques

You can use the following procedures to move about your program while you are in the Program, Run or Test modes.

When you want to	Press these keys	Explanation
Move to the following instruction.	**NEXT**	Handy when you want to move to a nearby instruction or review your program instruction-by-instruction.
Move to the previous instruction.	**LAST**	
Move to the following rung.	**RUNG, NEXT**	If the cursor is on a branch or condition instruction, you will move to the start of the next rung. If the cursor is on the output, you will move to the output of the next rung.
Move to the previous rung.	**RUNG, LAST**	If the cursor is on a branch or condition instruction, you will move to the start of the last rung. If the cursor is on the output, you will move to the output of the last rung.
Move to the start of the rung.	**SEARCH, LAST**	This saves time when the rung is complex, where you would otherwise have to press LAST or NEXT many times to go thru the instructions and/or data.
Move to the end of the rung.	**SEARCH, NEXT**	
Move to a specific rung, number 5 for example. Two methods can be used.	**RUNG, 5, ENTER**	If the cursor is on a branch or condition instruction, you will move to the start of rung 5. If the cursor is on the output, you will move to the output of rung 5.
	SEARCH, RUNG, 5, ENTER	You will move to the start of rung 5.
Move to the start of the program.	**SEARCH, RUNG, 1, ENTER**	You will move to the start of rung 1.
Move to the end of the program.	**SEARCH, RUNG, 9, 9, 9, ENTER**	You will move to the end statement after the last rung.
Move to a specific instruction, –] [–, 001 for example.	**SEARCH, –] [–, 1, ENTER**	Search begins at cursor location. If the instruction is not found in the program, **n F** is displayed.
Move to next occurrence of the specific instruction.	**SEARCH, ENTER**	If the instruction occurs only once in the program, the display of the first occurrence reappears.

Program Editing Techniques (continued)

You can use the following procedures to add, delete, or change instructions and rungs while you are in the Program mode.

Correct an error while entering an instruction, address, or data.	You can correct errors as long as the cursor position has not been moved (pressing ENTER, NEXT, LAST, or another instruction key moves the cursor).	**CANCEL CMD** (as often as required)	Each time you press the key, a previous keystroke is "wiped out". An instruction and its address can be canceled and changed in this way.
Insert an instruction in a completed rung. Example: –] [–, 008	Locate the cursor on the instruction following the point where you want to add the new instruction to the rung.	**INSERT, –] [–, 8, ENTER**	Cursor moves to the instruction following the new instruction.
Delete an examine instruction.	Locate the cursor on the instruction to be deleted.	**REMOVE, ENTER**	Cursor moves to the instruction which followed the deleted instruction.
Change an instruction or its address in a completed rung. Example: –(RTO)–, 901 to –( )–, 012	Locate the cursor on the instruction to be changed.	**–( )–, 1, 2, ENTER**	We have changed the instruction and its address. Cursor moves to the following instruction.
Another example: –] [–, 002 to –] [–, 702	Locate the cursor on the instruction to be changed.	**–] [–, 7, 0, 2, ENTER**	We have changed the address only.
Insert or remove a branch.	Refer to the keystroke example of Figure 15.4 on Page 15-7.		
Insert a rung. Example: 003 016 —] [—( )—	Locate the cursor on any instruction in the rung which follows the point where you want to enter the new rung.	**INSERT, RUNG, –] [–, 3, –( )–, 1, 6, NEXT**	Rung is inserted. Cursor has moved to the next rung. Rung numbers have been automatically changed.
Add a rung to the end of the program.	Locate the cursor on the End display.	**–] [–, 3, –( )–, 1, 6, ENTER**	We have entered the example rung at the end of the program.
Delete a rung.	Locate the cursor on any instruction in the rung you want to delete.	**REMOVE, RUNG, ENTER**	Rung is removed. Cursor has moved to the following rung.
Copy a rung. Example: Insert a copy of rung 2 after rung 5.	Locate the cursor on any instruction in rung 2.	**REMOVE, RUNG, INSERT, 6, ENTER**	Rung 2 is repeated and placed after rung 5. The following rungs are re-numbered accordingly.
Delete your entire program. (Select Clear Memory, mode 1.)	We will assume you are entered in the Program, Run, or Test mode. Cursor can be on any instruction in the program.	**MODE, 1, ENTER**	The prompt message **Sure ?** gives you the opportunity to change your mind. You would press CANCEL CMD to avoid clearing your program.
		ENTER	You have automatically entered the Program mode. 885 words (SLC 100) or 1200 words (SLC 150) of memory are available to you.

Program Editing Techniques *(continued)*

Use the following procedures to change or protect timer/counter data while you are in the Program mode. The table also lists procedures for changing, protecting, adding and deleting sequencer data.

When you want to	Cursor location	Press these keys	Explanation
Change the PR value of a timer or counter. Example: change timer PR to 20.0.	Locate the cursor on the PR value.	**2, 0, 0, ENTER**	Value is changed to 20.0 sec. Cursor has moved to the next instruction. **NOTE:** If PR value was protected, you must protect it again after changing the value.
Protect the PR value of a timer or counter (this also protects AC value).	Locate the cursor on the PR value.	**SHIFT, PRT**	Cursor moves to next instruction. The PRT LED will be lit while cursor is on the PR value (Program, Run, Test modes) *and the AC value* (Run, Test modes).
Remove protection from the PR value.	Locate the cursor on the PR value.	**UNPRT**	PR and AC values have returned to the unprotected state.
Change the RAC value of the RST instruction. Example: change RAC to 9.	Locate the cursor on the RAC value.	**9, ENTER**	Value is changed. Cursor has moved to the next instruction.
Changing, protecting, adding, and deleting sequencer data while in the Program mode (Procedure for changing RAC value is the same as for timers and counters)			
When you want to	**Cursor location**	**Press these keys**	**Explanation**
Access data in a specific step. Example: access data in step 5.	Cursor can be located on sequencer group, mask, step, or PR data.	**SEARCH, 5, ENTER**	Cursor has moved to the step data of step 5.
Change group number, mask data, step data, or PR values. Example: change step data for step 5 to 5, 7.	Locate the cursor on step 5. Display shows current data and **d 5**.	**5, 7, ENTER**	Value is changed to 5, 7. Cursor has moved to PR value of step 5. Other data is changed in this same way. Just locate cursor on data, enter new data, and press ENTER.
Protect the PR value of a sequencer step.	Locate the cursor on the PR value of the step.	**SHIFT, PRT**	Cursor moves to next instruction. PRT LED will be lit while cursor is on protected PR value (Program, Run, Test modes).
Remove protection from the PR value.	Locate the cursor on the PR value of the step.	**UNPRT**	The PR value has returned to the unprotected state.
Add a step after the final step. Example: step data of 7, 3; PR of 1.0 second.	Locate the cursor on the PR data for the final step.	**INSERT, NEXT, 7, 3, ENTER, 1, 0, ENTER, ENTER**	New step is added. You can add as many steps as you want before pressing the final ENTER.
Insert a step somewhere before the final step.	Locate cursor on the PR data of the step following the point where you will insert a step.	**INSERT, 7, 3, ENTER, 1,0, ENTER**	We have added the same data as in the example above. Cursor has moved to the next step.
Delete a step.	Locate cursor on the step data or PR value of the step to be deleted.	**REMOVE, ENTER**	Step is removed. Following steps are re-numbered.

Chapter 16 On-Line Data Control

General

With the controller in the Run or Test mode, you can monitor, control, and change your program. This is referred to as On-Line Data Control. It includes:

- Forcing external I/O addresses.
- Monitoring timer, counter, and sequencer data.
- Changing PR, AC, and RAC values.

> **WARNING:** Before you force external I/O or change data, investigate the effects on machine operation to avoid possible personal injury and equipment damage.

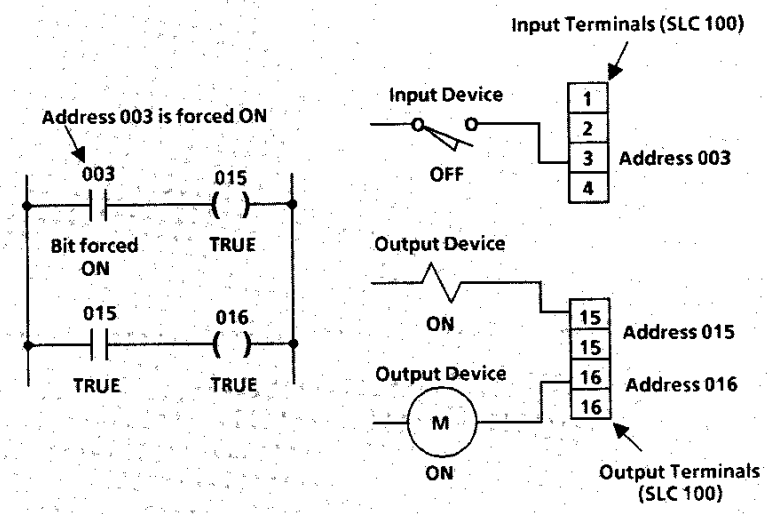
On-Line Data Control with the TCAT and PC Software: On-line data control techniques possible with the TCAT (Timer Counter Access Terminal) and the SLC Personal Computer Software are described in the respective User's Manuals, Publications 1745-850 and 1745-825.

Force Function

This function (FRC ON and FRC OFF keys) can be used in the Run and Test modes to force an external I/O address to an ON or OFF state regardless of its actual status. This function is very helpful in start-up and troubleshooting procedures.

Forcing an input address: When you force an input address, you are forcing the status bit of the instruction at the I/O address to an ON or OFF state. The program scan records this, and the program is executed with this forced status regardless of the actual ON/OFF status of the input device. This is illustrated in Figure 16.1.

Figure 16.1



Forcing an input address. The program is executed as if the input device were actually ON.

Force Function (continued)

Forcing an output address: In this case, we are forcing only the output terminal to an ON or OFF state. The status bit of the output instruction at the address is **not** affected. Refer to Figure 16.2.

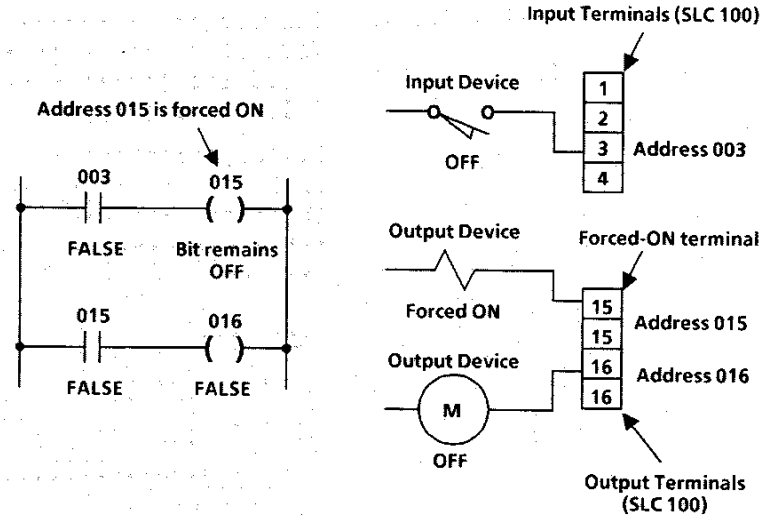
A forced I/O address will be retained in memory when any of the following occurs: 1) A power outage; 2) The mode is changed from Run; 3) The pocket programmer is disconnected. A forced I/O address will not be retained if you enter the Program mode and make a change, or if a processor fault is detected.

Figure 16.3 on the following page shows the keystrokes used for forcing I/O addresses.

Forcing sequencer I/O bit addresses: These programmed I/O addresses cannot be accessed in the user program, but they can be forced ON or OFF.

Example: To force sequencer output bit address 016 ON, press INSERT, –( )–, 1, 6, ENTER, SHIFT, FRC ON. To remove the force, press REMOVE, –( )–, 1, 6, SHIFT, FRC ON, ENTER. To remove all forces in the program, press REMOVE, ENTER.

Figure 16.2



Forcing an output address. The output device associated with the forced output address is turned ON. The status bit is not affected.

TRUE-FALSE Status Indicator

With the controller in the Run or Test mode, you can monitor the TRUE/FALSE status of an instruction. The ON LED of the programmer is lit when the cursored instruction is TRUE, and not lit when the instruction is FALSE. Remember that when you force an output address, only the output terminal is forced; the output instruction is not affected, and therefore the ON LED remains off.

Figure 16.3 indicates how the ON LED functions.

Figure 16.3

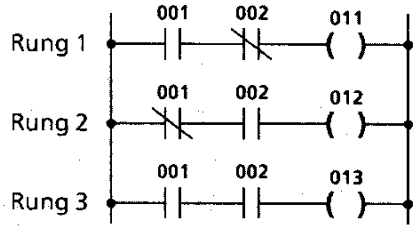
Keystroke Example – Forcing instruction addresses. Program the ladder diagram at the right. Read the following information before going thru the keystroke example. Note that the ON LED and the force function are effective in the Run and Test modes only.

ON LED: Indicates whether the cursored instruction is TRUE or FALSE. The LED is lit when the instruction is TRUE and not lit when it is FALSE. The LED is not lit when an output terminal is forced.

FORCE LED: This LED is continuously lit whenever one or more instructions in the program are forced ON or OFF.

FORCE ON function key: Pressing this key will force the status bit address of the cursored instruction to the ON state (except for output addresses). If the status bit was already ON, it will be held ON until the force is removed.

FORCE OFF function key: Pressing this key will force the status bit address of the cursored instruction to the OFF state. If the status bit was already OFF, it will be held OFF until the force is removed.



Ladder Diagram

Press these keys	The LED will be lit	Display will show	Explanation
	(no addition for PROG)	Address / Rung / Data / Mode	Arbitrarily, We are using an SLC 100. After you enter the ladder diagram there will be 876 words remaining.
		876 End	
MODE, 3, ENTER		876 End	You are in the Run mode. The PROG LED is no longer lit.
LAST 8 times	–]/[–, ON	2 1	You have moved thru your program to the Examine OFF instruction in rung 1. The ON LED is lit, indicating the instruction is TRUE (status bit OFF).
SHIFT, FRC ON	–]/[–, FORCE	F 2 1	Status bit address 002 has been forced ON. The letter F in the display indicates the address has been forced (either ON or OFF). The ON LED is not lit, indicating the instruction is FALSE. The FORCE LED is lit, and will stay lit as long as one or more instructions in the program have been forced ON or OFF.
NEXT 3 times	–] [–, ON, FORCE	F 2 2	Cursor is located on the Examine ON instruction in rung 2. The letter F in the display indicates bit address 002 has been forced. We did this in the previous step. The ON LED is lit, indicating the instruction is TRUE (status bit ON).
NEXT	–( )–, ON, FORCE	12 2	Cursor is located on the output instruction in rung 2. The ON LED is lit, indicating that the instruction is TRUE. By forcing status bit address 002 ON, we have made both condition instructions in rung 2 TRUE, making the rung TRUE. Notice that the letter F does not appear in the display. This is because we have not applied the force function to status bit address 012.
NEXT, SHIFT, FRC ON	–] [–, ON, FORCE	F 1 3	We have moved the cursor to the Examine ON instruction 001 in rung 3 and have forced bit address 001 ON. Now both bit addresses 001 and 002 are forced ON. (Rung 2 is now FALSE. Rung 3 is now TRUE).
NEXT	–] [–, ON, FORCE	F 2 3	Cursor is on the Examine ON instruction 002, rung 3. The next two steps show you how to remove a force from a particular bit address (002 in this case).
REMOVE	–] [–, ON, FORCE	rE F 3	Display indicates "remove force".
SHIFT, FRC ON	–] [–, FORCE	2 3	The Force ON is removed from all instructions at status bit address 002. Status bit address 001 is still forced ON, therefore the FORCE LED remains lit.
REMOVE, ENTER	–] [–	2 3	To remove forces from all forced addresses, press REMOVE, ENTER as we've done here. Cursor does not have to be on a forced instruction.

Changing and Monitoring Techniques

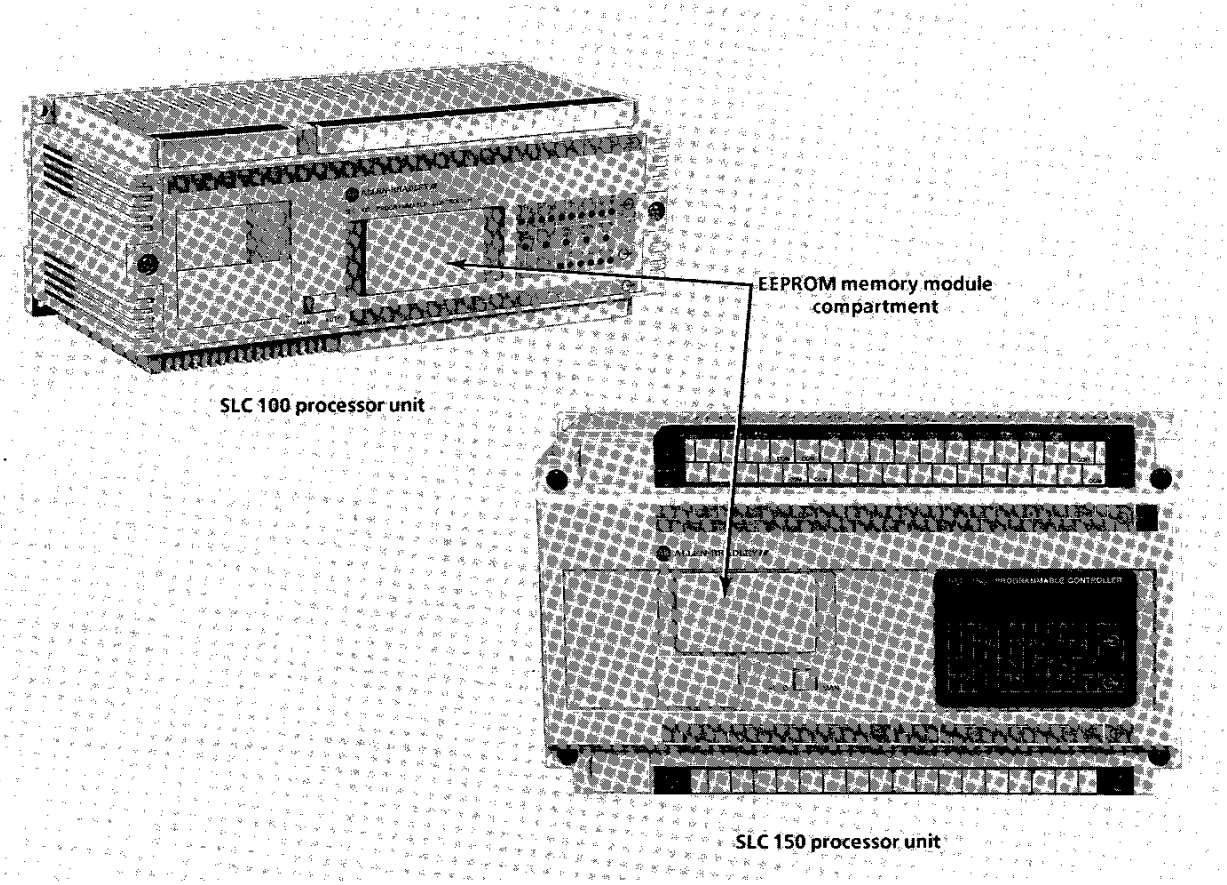
You can use the following procedures to change or monitor PR, AC, and RAC values of timers and counters while you are in the Run or Test mode.

The figure also lists procedures for changing and monitoring sequencer PR and RAC values.

Figure 16.4

When you want to:	Cursor location	Press these keys:	Explanation:
Change the PR value (if unprotected). Example: change PR to 200.	Locate the cursor on the PR value. (If the PRT LED is lit, value cannot be changed.)	**2, 0, 0, ENTER**	Value is changed. Cursor remains on the PR value. Press NEXT or LAST to move it.
Monitor and change an AC value. Assume this is a timer. Change the AC value to 10.0.	Locate the cursor on the AC value.	**1, 0, 0, ENTER**	The AC value is changed to 10.0 seconds. If the AC value is incrementing, it will continue to increment from 10.0. NOTE: If the PR value is protected, the PRT LED will be lit and you will not be able to change the AC value.
Change the RAC value. Example: change RAC to 5.	Locate the cursor on the RAC value.	**5, ENTER**	Value is changed. Cursor remains on the RAC value.
Changing/monitoring sequencer PR and RAC values while in the Run or Test mode:			
When you want to:	**Cursor location**	**Press these keys:**	**Explanation:**
Monitor step data, PR values, or AC values.	Locate the cursor on the mask data (**U S E**).	**NEXT**	Current step number and corresponding step data is displayed. If the sequencer is operating, the step number and data will be incrementing.
		NEXT	Current step number and corresponding PR value is displayed. If the sequencer is operating, the PR value and step number will be incrementing.
		NEXT	Current step number and corresponding AC value is displayed. If the sequencer is operating, the AC value and step number will be incrementing.
Change the PR value (unprotected) of a specific step. Example: change PR value of step 5 to 25.	Locate the cursor on the PR value. If the sequencer is operating, the PR value and step number will be incrementing.	**INSERT**	Display prompts for step number, **S t P.**
		5, ENTER	Current PR value of step 5 is displayed.
		2, 5, ENTER	New PR value of 25 is entered. If the sequencer is operating, the PR value and step number will be incrementing.
Change the RAC value. Example: change RAC to 2.	Locate the cursor on the RAC value.	**2, ENTER**	Value is changed. Cursor remains on the RAC value.

Chapter 17 Using the EEPROM Memory Module



The EEPROM Memory Module

The EEPROM is a non-volatile memory in convenient modular form, for use with SLC 100 and SLC 150 Programmable Controllers.

A Series B module or a Series A module having a 28-pin chip must be used with the SLC 150 processor unit. The majority of Series A modules sold in the past meet the 28-pin chip requirement.

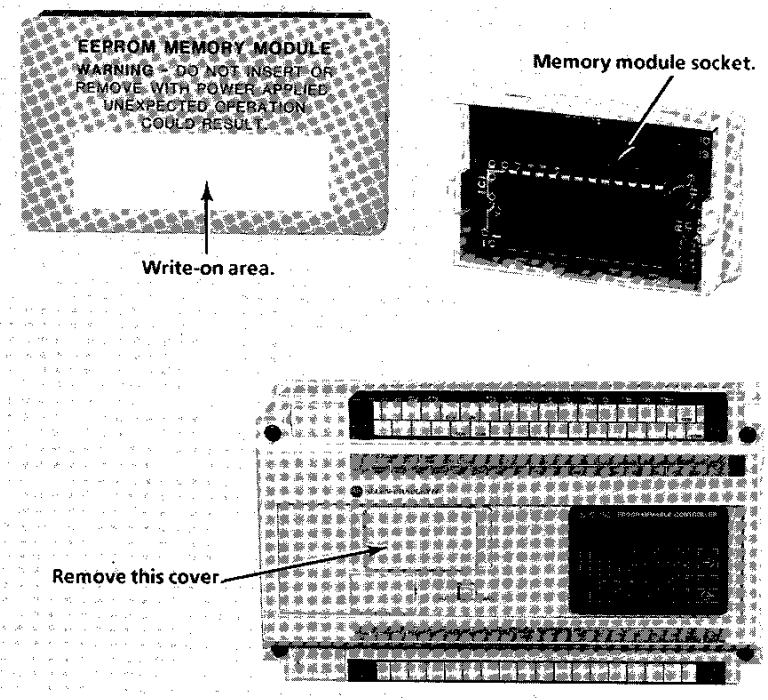
EEPROM Program Identification

Since EEPROM modules can be inserted in any SLC 100 or SLC 150 processor unit, it is important that you identify each EEPROM with the particular program it contains and the controller or controllers it is to be used with.

Caution: Interchanging programs between controllers could cause improper operation and equipment damage. You must be especially cautious when interchanging programs between SLC 150 and SLC 100 controllers, because of differences associated with certain internal addresses and other variations.

Installing the EEPROM

The procedure for installing the EEPROM in the processor unit is shown in the figure below.



SLC 150 processor unit. EEPROM socket cover on SLC 100 processor unit is similarly located.

Caution: Always turn off power to the processor before you insert or remove the EEPROM. This guards against possible damage to the EEPROM and also avoids undesired CPU faults.

INSTALLATION PROCEDURE:

1. Turn off power to the processor unit.
2. To insert the memory module, remove the cover from the front of the processor unit. Save the cover for use later.
3. Align the memory module socket with the receptacle inside the processor unit. Push *carefully* to seat the module.
4. Restore power to the processor.

Storing and Loading Procedures

The following paragraphs explain how to use the pocket programmer to store a processor RAM program in the EEPROM and to load an EEPROM program in the processor RAM. If you are using the personal computer software, refer to the SLC Personal Computer Software User's Manual (Pub. 1745-825) for further information on these procedures.

Storing a Program in the Memory Module

The following steps show you how to use the pocket programmer to store (save) the contents of the processor RAM in the EEPROM memory module. If you are using an access code, it will be stored in the EEPROM. We'll assume that the memory module is not plugged in.

1. Connect the programmer. Press MODE, 6, ENTER. Display will show **SurE ?**

   Note: You could press ENTER a second time, altho it isn't necessary. Error code E51 will appear; you can ignore the error code and perform step 2.

2. Disconnect power from the processor. Insert the memory module. Restore power.
3. Press ENTER, ENTER. After a brief time, the display will momentarily show **donE**. Any program previously stored in the memory module will be erased.
4. You can now disconnect power and remove the memory module (replace cover), or you can leave the module in place. The processor operates from the RAM only.

> **Caution:** It is important to enter mode 6 *before* installing the memory module. Reason: If the processor were in the Run mode when you disconnect power, insert the memory module, and restore power (step 2), your program could be *automatically erased* and replaced with the contents of the memory module. Refer to auto-load procedure on the following page.

Loading the EEPROM into the Processor RAM

The following steps show you how to use the pocket programmer to load (read) the contents of the EEPROM memory module into the processor RAM. If you know that the processor is in the Run mode, you may be able to use the auto-load procedure described on the following page. If the processor is in some other mode, follow these steps:

1. Disconnect power from the processor. Insert the memory module. Restore power.
2. Connect the programmer. Press MODE, 7, ENTER, ENTER. (If an access code is stored in the EEPROM, you will be requested to enter the code after you press ENTER the second time.)
3. Duplication of the EEPROM contents in the RAM occurs almost instantaneously. The previous program in the RAM is automatically erased.
4. You can now disconnect power and remove the memory module (replace cover), or you can leave the module in place. The processor operates from the RAM only.

Auto-Load Procedure

If you are using an SLC 100 processor unit: The auto-load procedure will load the contents of an EEPROM into the processor RAM without the use of the programmer. The only exception: If the EEPROM has an access code, you must use the programmer. (It doesn't matter whether the processor has the same code as the EEPROM, a different code, or no code.)

Follow these steps:

1. The processor must be in the Run mode initially.
2. If the processor is energized, disconnect power. Insert the EEPROM. **Caution:** To avoid automatically entering the Run mode when you restore power, set the Auto/Man switch to Man at this time.
3. Restore power. The EEPROM contents is automatically duplicated in the RAM. The previous program in the RAM is automatically erased.
4. When you are prepared to enter the Run mode, set the Auto/Man switch to Auto.

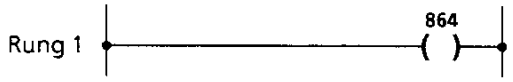
If you are using an SLC 150 processor unit: In this case, you can enter an access code in the processor RAM to protect the stored program against unauthorized EEPROM loading. The table below indicates the conditions under which the auto-load procedure will and will not function. If the auto-load is functional, follow steps 1 - 4 in the paragraph above.

[illegible]	[illegible]	[illegible]
Has access code	Has no access code or access code does not match code in processor	No. You must use programmer.
Has access code	Has same access code as the processor	Yes
Has no access code	May or may not have access code	Yes

Programmable Auto-Load – SLC 150 Processor Unit

With the SLC 150 processor unit, you can use address 864 in your program to automatically load the EEPROM program in the processor RAM when the processor detects a memory checksum error. Such errors are detected if memory data is altered or lost due to capacitor back-up drain, battery back-up drain, or processor malfunction. It works like this:

1. Enter the following unconditional rung as Rung 1 of your program:



Programmable Auto-Load – SLC 150 Processor Unit (continued)

2. Enter the rest of your program and store the program in the EEPROM. This is your "back-up" program.

3. If a memory problem occurs and the processor unit detects a memory error (codes 5 thru 8), recycle power. The EEPROM program will be automatically loaded in the processor RAM, clearing the error.

   An error 6 message will be displayed by your pocket programmer or computer following an auto-load error. This error message can be cleared by using the Cancel key on the pocket programmer or re-entering the Run/Mon/Test mode via SLC personal computer software.

Important: Keep in mind that when the auto-load occurs, the I/O table data is cleared. All retentive instructions (latches, timers, counters, sequencers) will be reset to their initial states.

Editing EEPROM Programs

You cannot edit the EEPROM program directly. You must first load the program into the processor RAM, if it is not already in the RAM. You then edit the RAM program and store the edited program in the EEPROM.

For example, suppose an EEPROM is installed, and its contents are loaded in the RAM. You can change the program in the EEPROM as follows:

1. The RAM program is a duplicate of the EEPROM program; edit the RAM program.
2. Press MODE, 6, ENTER, ENTER.
3. The edited program is now in the EEPROM (and in the RAM).

Changes not Saved in EEPROM (Error Code E52)

Error code E52 tells you that the processor RAM program does not match the EEPROM program. If an EEPROM is installed in the processor unit,this error will appear after you edit your program and attempt to enter the Run or Test modes. This reminds you that you should load the edited program in the EEPROM before going into the Run mode.

Error code E52 can be overridden by pressing ENTER again. Note: Error code E52 is not displayed when on-line data changes are made to the processor RAM.

C Installing and Maintaining the System

Chapter 18 Specifications

General Specifications – SLC 100 Processor Unit

Voltage Ranges (Incoming Power, Input Circuits, Output Circuits):

Catalog Number	Voltage Ranges (Incoming Power Connections)	External I/O Circuits	
		Input	Output (Hard Contact)
1745-LP101	85-132/170-265 VAC 50/60 Hz	85-132 VAC 50/60 Hz	10-250VAC/ 10-125VDC
1745-LP102	85-132/170-265 VAC 50/60 Hz	170-265 VAC 50/60 Hz	
1745-LP103	85-132/170-265 VAC 50/60 Hz	10-30V AC/DC	
1745-LP104	18-30 VDC	10-30V AC/DC	
	Full wave rectified AC voltage 18V RMS ± 20%, 30V peak		

Input Terminal Cover Color Code	
Volts	Color
85-132 VAC 50/60 Hz	Red
170-265 VAC 50/60 Hz	Black
10-30V AC/DC	Blue

Maximum Power Requirement:
1745-LP101, -LP102, -LP103: 20VA. 1745-LP104: 15VA.

Input Power Fuse Protection:
1745-LP101, -LP102, -LP103: 315mA/250V. 1745-LP104: 1.6 A/250V.
Fuse Types: SAN-O: SOC SD4. Bussman: MDL, or GDC (miniature).

Hold-Up Time: The processor can sustain operation for approx. 300 milliseconds in the event of power interruption or removal. It takes approx. 85 milliseconds for the processor to reach full power from an initial power on condition.

Specifications applying to all Catalog Numbers

Memory Type: CMOS RAM with battery back-up. Provision for optional EEPROM module (Cat. No. 1745-M1).

Battery Back-Up: Lithium battery, non rechargeable, 2-3 year life.

User Memory Size: 885 words maximum. (Most instructions require 1 word.)

Typical Scan Time: 15 msec (depends on program length).

I/O Capacity: 16 I/O (10 inputs, 6 outputs). Expandable to 112 I/O.
Input specifications: Page 18-3.
Output specifications: Page 18-5.

Internal Relay-Type Instructions: 181 max, regular or latched.

Timers/Counters/Sequencers: 32 max, any combination. Retentive.

Time Base: 0.1 sec. Fine time bases to 0.01 sec can be selected.

Timer Range: 0.1 to 999.9 seconds.

Counter Capacity: 9999 counts.

Sequencer Capacity: 8 bits x 100 steps.

Shift Register: 8-bit groups.

Noise Immunity: NEMA Standard ICS 2-230.

Vibration: DIN Rail Mounting: 0.006 inch peak to peak displacement, 1.0g peak (max) acceleration, 1 Hr/axis. Screw Fastener Mounting: 0.015 inch peak to peak displacement, 2.5g peak(max) acceleration, 1 Hr/axis.

Ambient Temperature Rating: 0° to 60° C (operating). – 40° to 85° C (storage).

Humidity Rating: 5 to 95% (without condensation).

Wiring: #14 - #24 AWG stranded. 3/64" insulation (max).

General Specifications – SLC 100 Expansion Units

General specifications for the basic, relay output, and analog input expansion units are shown below.

Voltage Ranges (Incoming Power, Input Circuits, Output Circuits):

Catalog Number	Voltage Ranges (Incoming Power Connections)	External I/O	
		Input Circuits	Output Circuits – Hard Contact
1745-E101	85-132/170-265 VAC 50/60 Hz	85-132 VAC 50/60 Hz	10-250VAC/10-125VDC
1745-E102	85-132/170-265 VAC 50/60 Hz	170-265 VAC 50/60 Hz	
1745-E103	85-132/170-265 VAC 50/60 Hz	10-30V AC/DC	
1745-E104	18-30 VDC	10-30V AC/DC	
	Full wave rectified AC voltage 18 volts RMS ± 20%, 30 volts peak		
1745-E105	85-132/170-265 VAC 50/60 Hz	N/A	
1745-E106	85-132/170-265 VAC 50/60 Hz	Analog Inputs 0-10 VDC or 0-20 mA	N/A
1745-E107	18-30 VDC		
	Full wave rectified AC voltage 18 volts RMS ± 20%, 30 volts peak		

Input Terminal Cover Color Code	
Volts	Color
85-132 VAC 50/60 Hz	Red
170-265 VAC 50/60 Hz	Black
10-30V AC/DC	Blue

Maximum Power Requirement:
1745-E101, -E102, -E103, -E105, -E106: 12VA.
1745-E104, -E107: 9VA.

Input Power Fuse Protection:
1745-E101, -E102, -E103: 315mA/250V.
1745-E104: 1.6A/250V.
1745-E105: 160mA/250V.
1745-E106: 200mA/250V.
1745-E107: 1A/250V.

Fuse Types: SAN-O: SOC SD4.
Bussman: MDL, or GDC1 (miniature).

I/O Capacity:
1745-E101, -E102, -E103, -E104:
16 I/O (10 inputs, 6 outputs).
1745-E105: 12 I/O (0 inputs, 12 outputs).
1745-E106, -E107: 4 analog input circuits, with 2 adjustable set points per input.

Input specifications: Pages 18-3 and 18-4.
Output specifications: Page 18-5.

Specifications applying to all Catalog Numbers

Noise Immunity: NEMA Standard ICS 2-230.

Vibration:
DIN Rail Mounting: 0.006 inch peak to peak displacement, 1.0g peak (max) acceleration, 1 Hr/axis. Screw Fastener Mounting: 0.015 inch peak to peak displacement, 2.5g peak(max) acceleration, 1 Hr/axis.

Ambient Temperature Rating:
0° to 60° C (operating).
– 40° to 85° C (storage).

Humidity Rating:
5 to 95% (without condensation).

Wiring: #14 - #24 AWG stranded.
3/64" insulation (max).

Input Specifications – SLC 100 Processor and Basic Expansion Units

Input specifications are shown below. The 2 mA OFF state leakage current specification allows direct interface to solid state sensing devices.

Note that the 2 mA OFF state leakage current for 24VAC circuits applies to inputs 1 and 2 only. When required, you can also achieve a 2 mA OFF state leakage current for inputs 3 thru 10, as indicated in the specifications.

All input circuits include optical isolation as well as filtering and surge suppression to guard against damage by transients from external input devices.

ON State Voltage Range and Frequency:
1745-LP101, -E101: 85-132 VAC, 50/60 Hz.
1745-LP102, -E102: 170-265 VAC, 50/60 Hz.
1745-LP103, -LP104, -E103, -E104: 10-30V, AC/DC.

Maximum OFF State Voltage:
1745-LP101, -E101: 35V.
1745-LP102, -E102: 50V.
1745-LP103, -LP104, -E103, -E104: 5V.

Maximum OFF State Leakage Current:
1745-LP101, -LP102, -E101, -E102: 2 mA.
1745-LP103, -LP104, -E103, -E104: 2mA, inputs 1 and 2 only. To achieve a 2 mA OFF state leakage current for inputs 3 thru 10, connect a 5.6KΩ, 1/2 watt (min) resistor from the input terminal to the common terminal.

Nominal Input Current:
1745-LP101, -LP102, -E101, -E102: 8 mA.
1745-LP103, -LP104, -E103, -E104: Inputs 1 and 2– 8 mA at 12V, 18 mA at 24V. Inputs 3 thru 10– 6 mA at 12V, 14 mA at 24V.

Specifications applying to all Catalog Numbers:

Input Filter Time Delay:
10-25 msec.

Electrical-Optical Isolation:
1500 volts between input voltage and control logic.

Input Specifications – SLC 100 Analog Input Expansion Unit

Input specifications for the analog input expansion unit are shown below.

Voltage Inputs

Minimum Isolation Between Input and Control Logic:
1500 volts.

Input Type: Differential.

Input Impedance:
+V to (–): > 1000 K ohms.
+V or (–) to COM: > 500 K ohms.

Trip Point Repeatability over Time and Temperature:
25-35°C: 0.25% of full scale or 25 millivolts (typical).
1.00% of full scale or 100 millivolts (worst case).
0-60°C: 0.50% of full scale or 50 millivolts (typical).
1.50% of full scale or 150 millivolts (worst case).

Hysteresis: 10 millivolts.

Maximum Operating Range:
+V to (–): 0 to +10 volts.
+V or (–) to COM: –35 to +35 volts.

Minimum Circuit Protection Range:
+V to (–), +V or (–) to COM: ± 50.0 volts.

Maximum Input Response Time:
3.5 milliseconds.

Current Inputs

Minimum Isolation Between Input and Control Logic:
1500 volts.

Input Type: Differential.

Input Impedance:
+I to (–): 250 ohms ± 1.0%.

Trip Point Repeatability over Time and Temperature:
25-35°C: 0.50% of full scale or 0.10mA (typical).
1.75% of full scale or 0.35mA (worst case).
0-60°C: 1.00% of full scale or 0.20mA (typical).
2.75% of full scale or 0.55mA (worst case).

Hysteresis: 40 microamps.

Maximum Operating Range:
+I to (–): 0 to 20 mA.
+I or (–) to COM: –35 to +35 volts.

Minimum Circuit Protection Range:
+I to (–): ± 40mA.
+I or (–) to COM: ± 50.0 volts.

Maximum Input Response Time:
3.5 milliseconds.

Output Specifications – SLC 100 Processor and Expansion Units

Output specifications for all versions of the processor unit and the basic and relay output expansion units are shown below.

Internal output circuitry includes surge suppression to guard against possible damage by transients from external output devices. We recommend that you also use some type of contact protection when switching inductive load devices. Refer to Page 19-9.

Voltage Range and Frequency:
10-250 VAC (50/60 Hz), 10-125 VDC.

Contact Ratings:

Maximum Volts	Amperes		Amperes Continuous	Voltamperes	
	Make	Break		Make	Break
240VAC 120VAC	7.5A 15A	0.75A 1.5A	2.5A	1800VA	180VA
125VDC	0.22A		1.0A	28VA	
24VDC	1.2A		2.5A	28VA	

Contact Resistance: 20 mΩ (typical).

Electrical Isolation: 2000 volts.

OFF State Leakage Current: 2 mA (AC voltage only).

Internal RC Network Values: R = 120 ohms, C = 0.022 microfarad.

General Specifications – SLC 150 Processor Unit

Voltage Ranges (Incoming Power, Input Circuits, Output Circuits):

Catalog Number	Voltage Ranges (Incoming Power Connections)	External I/O		
		Input Circuits – 20	Output Circuits – 12	Color Patch*
1745-LP151	85-132/170-265 VAC 50/60 Hz	85-132 VAC 50/60Hz Current Sinking	10 Triac – 85-265 VAC 50/60 Hz 2 Hard Contact – 10-250VAC/10-125VDC	Red
1745-LP152	85-132/170-265 VAC 50/60 Hz	170-265 VAC 50/60Hz Current Sinking	10 Triac – 85-265 VAC 50/60 Hz 2 Hard Contact – 10-250VAC/10-125VDC	Black
1745-LP153	85-132/170-265 VAC 50/60 Hz	10-30 VDC Current Sinking	12 Hard Contact – 10-250VAC/10-125VDC	Blue
1745-LP154	18-30 VDC	10-30 VDC Current Sinking	10 PNP Transistor – 10-50 VDC 2 Hard Contact – 10-250VAC/10-125VDC	Green
1745-LP156	85-132/170-265 VAC 50/60 Hz	10-30 VDC Current Sourcing	12 Hard Contact – 10-250VAC/10-125VDC	Purple
1745-LP157	18-30 VDC	10-30 VDC Current Sourcing	10 NPN Transistor – 10-50 VDC 2 Hard Contact – 10-250VAC/10-125VDC	Yellow

* Color patch appears in the upper right corner of the unit, above the cluster of LED status indicators.

Maximum Power Requirement:
1745-LP151, -LP152, -LP153, -LP156: 25 VA.
1745-LP154, -LP157: 15 VA.

Input Power Fuse Protection:
1745-LP151, -LP152, -LP153: 315mA/250V.
1745-LP154: 1.6A/250V.
Fuse Types: SAN-O: SOC SD4.
Bussman: MDL, or GDC (miniature).

Hold-Up Time:
The processor can sustain operation for a minimum of 25 milliseconds in the event of a power interruption.

I/O Capacity:
20 inputs and 12 outputs. See table above.

Input specifications: Page 18-8.
Output specifications: Pages 18-9 and 18-10.

Memory Type: CMOS RAM.

Standard Capacitor Back-up: Provides memory back-up for 2 weeks at 30°C and 1 week at 60°C. No maintenance required.

Note: If the processor unit is operated in temperatures above 50°C for extended periods of time, capacitor life cannot be guaranteed beyond 2 years. For these applications, we recommend that you install the Lithium battery for memory back-up.

Optional Battery Back-up: Catalog No. 1745-B1. Lithium battery, non-rechargeable. 2 to 3 year life. Actual life may vary, depending on controller environmental conditions.
Installation: Page 22-11.

User Memory Size: 1200 words max. Most instructions require 1 word.

Typical Scan Time:
4 msec (1000 word program).

Internal Relay Type Instructions:
177 max. (regular or latched).

Timers, Counters, Sequencers:
32 max, any combination, retentive.

Time Base: 0.1 sec. Fine time bases down to 0.01 second can be selected.

Timer Range:
0.1 to 999.9 seconds.

Sequencer Capacity:
8 bits x 100 steps.

Shift Register:
8 bit groups.

Noise Immunity:
NEMA Standard ICS 2-230.

Vibration:
0.015 inch peak to peak displacement, 2.5g peak(max) acceleration, 1 Hr/axis.

Ambient Temperature Rating:
0° to 60° C (operating).
– 40° to 85° C (storage).

Humidity Rating:
5 to 95% (without condensation).

Wiring: #14 - #24 AWG stranded.
3/64" insulation (max).

General Specifications – SLC 150 Expansion Unit

Voltage Ranges (Incoming Power, Input Circuits, Output Circuits):

Catalog Number	Voltage Ranges (Incoming Power Connections)	External I/O		
		Input Circuits – 20	Output Circuits – 12	Color Patch*
1745-E151	85-132/170-265 VAC 50/60 Hz	85-132 VAC 50/60Hz Current Sinking	10 Triac – 85-265 VAC 50/60 Hz 2 Hard Contact – 10-250VAC/10-125VDC	Red
1745-E152	85-132/170-265 VAC 50/60 Hz	170-265 VAC 50/60Hz Current Sinking	10 Triac – 85-265 VAC 50/60 Hz 2 Hard Contact – 10-250VAC/10-125VDC	Black
1745-E153	85-132/170-265 VAC 50/60 Hz	10-30 VDC Current Sinking	12 Hard Contact – 10-250VAC/10-125VDC	Blue
1745-E154	18-30 VDC	10-30 VDC Current Sinking	10 PNP Transistor – 10-50 VDC 2 Hard Contact – 10-250VAC/10-125VDC	Green
1745-E156	85-132/170-265 VAC 50/60 Hz	10-30 VDC Current Sourcing	12 Hard Contact – 10-250VAC/10-125VDC	Purple
1745-E157	18-30 VDC	10-30 VDC Current Sourcing	10 NPN Transistor – 10-50 VDC 2 Hard Contact – 10-250VAC/10-125VDC	Yellow

* Color patch appears in the upper right corner of the unit, above the cluster of LED status indicators.

Maximum Power Requirement:
1745-E151, -E152, -E153, -E156: 15 VA.
1745-E154, -E157: 9 VA.

Input Power Fuse Protection:
1745-E151, -E152, -E153: 315mA/250V.
1745-E154: 1.6A/250V.
Fuse Types: SAN-O: SOC SD4.
Bussman: MDL, or GDC (miniature).

I/O Capacity:
20 inputs and 12 outputs. See table above.

Input specifications: Page 18-8.
Output specifications: Pages 18-9 and 18-10.

Specifications applying to all Catalog Numbers

Noise Immunity: NEMA Standard ICS 2-230.

Vibration:
0.015 inch peak to peak displacement, 2.5g peak(max) acceleration, 1 Hr/axis.

Ambient Temperature Rating:
0° to 60° C (operating).
– 40° to 85° C (storage).

Humidity Rating:
5 to 95% (without condensation).

Wiring: #14 - #24 AWG stranded.
3/64" insulation (max).

Input Specifications – SLC 150 Processor and Expansion Units

Input specifications are shown below. All input circuits include optical isolation as well as filtering to guard against high voltage transients from external input devices.

ON State Voltage Range and Frequency:
1745-LP151, -E151: 85-132 VAC, 47-63 Hz.
1745-LP152, -E152: 170-265 VAC, 47-63 Hz.
1745-LP153, -LP154, -LP156, -LP157, -E153, -E154, -E156, -E157: 10-30VDC.

Maximum OFF State Voltage:
1745-LP151, -E151: 30V.
1745-LP152, -E152: 50V.
1745-LP153, -LP154, -LP156, -LP157, -E153, -E154, -E156, -E157: 4V.

Maximum OFF State Leakage Current:
1745-LP151, -LP152, -E151, -E152: 2 mA.
1745-LP153, -LP154, -LP156, -LP157, -E153, -E154, -E156, -E157: 1mA.

Nominal Input Current:
1745-LP151, -LP152, -E151, -E152: 8 mA.
1745-LP153, -LP154, -LP156, -LP157, -E153, -E154, -E156, -E157: 4 mA at 12VDC, 8mA at 24VDC.

Input Filter Time ON Delay:
1745-LP151, -LP152, -E151, -E152: 3 to 13 msec.
1745-LP153, -LP154, -LP156, -LP157, -E153, -E154, -E156, -E157: 4 to 8 msec.

Input Filter Time OFF Delay:
1745-LP151, -LP152, -E151, -E152: 9 to 18 msec.
1745-LP153, -LP154, -LP156, -LP157, -E153, -E154, -E156, -E157: 4 to 8 msec.

Electrical-Optical Isolation:
1500 volts between input voltage and control logic. Applies to all catalog numbers.

Output Specifications – SLC 150 Processor and Expansion Units

TRIAC OUTPUTS, 1745-LP151, -LP152, -E151, -E152
(Hard contact relay outputs: Page 18-10)

Output Voltage Range: 85-265 VAC.

Continuous Output Current per Circuit:
1A at 30°C, linearly derated to 0.5A at 60°C.

Continuous Output Current per Chassis:
10A at 30°C, linearly derated to 5A at 60°C.

Surge Current:
10A for 25 msec. Repeat once each second at 30°C, or once each 2 seconds at 60°C.

Minimum Load Current: 10 mA.

Maximum OFF State Leakage Current: 2 mA.

OFF to ON Response Time (non-zero cross): 0.1 msec (max).

Zero-Cross Turn-on Timing Accuracy: ± 500 microseconds.

Saturation Voltage Drop: 1.5 volts at 1.0A.

Electrical-Optical Isolation:
1500 volts between output voltage and control logic.

Recommended Output Fusing: San-O: SOC ST4-3A, Bussman: MSL-2A, or equivalent.

TRANSISTOR OUTPUTS, 1745-LP154, -LP157, -E154, -E157
(Hard contact relay outputs: Page 18-10)

Output Voltage Range: 10-50 VDC.

Continuous Output Current per Circuit:
1A at 30°C, linearly derated to 0.5A at 60°C.

Continuous Output Current per Chassis:
10A at 30°C. 5A at 60°C.

Surge Current:
3A for 20 msec. Repeat once each second at 30°C, or once each 2 seconds at 60°C.

Minimum Load Current: 1.0 mA.

Maximum OFF State Leakage Current: 0.1 mA.

OFF to ON Response Time: 100 microseconds.

Maximum ON State Voltage Drop:
1.5 volts at 1.0A. 0.8 volts at 0.5A.

Electrical-Optical Isolation:
1500 volts between output voltage and control logic.

Recommended Output Fusing: San-O: SOC ST4-2A, Bussman: MSL-2A, or equivalent.

Output Specifications – SLC 150 Processor and Expansion Units (continued)

Specifications for hard contact relay outputs are shown below. We recommend that you use some type of surge suppression when switching inductive load devices with hard contact outputs. Refer to Page 19-9.

HARD CONTACT RELAY OUTPUTS

Wiring Terminals, Internal Arc Suppression:

Catalog Number	Hard Contact Relay Outputs
1745-LP151 1745-LP152 1745-LP154 1745-LP157	Units have 2 hard contact relay outputs, at terminals 11 and 111. These outputs do *not* have internal arc suppression circuitry.
1745-LP153 1745-LP156	Units have 12 hard contact relay outputs. Relay outputs at terminals 12 thru 16 and 112 thru 116 have internal arc suppression circuitry. Relay outputs at terminals 11 and 111 do *not* have internal arc suppression circuitry.
1745-E151 1745-E152 1745-E154 1745-E157	Units have 2 hard contact relay outputs, at terminals A11 and B11. These outputs do *not* have internal arc suppression circuitry.
1745-E153 1745-E156	Units have 12 hard contact relay outputs. Relay outputs at terminals A12 thru A16 and B12 thru B16 have internal arc suppression circuitry. Relay outputs at terminals A11 and B11 do *not* have internal arc suppression circuitry.

Voltage Range and Frequency:
10-250 VAC (50/60 Hz), 10-125 VDC.

Contact Ratings:

Maximum Volts	Amperes		Amperes Continuous	Voltamperes	
	Make	Break		Make	Break
240VAC 120VAC	7.5A 15A	0.75A 1.5A	2.5A	1800VA	180VA
125VDC	0.22A		1.0A	28VA	
24VDC	1.2A		2.5A	28VA	

Contact Resistance: 20 mΩ (typical).

Electrical Isolation:
2000 volts between output contacts and control logic.

OFF State Leakage Current:
Processor unit outputs at terminals 11 and 111 (all catalog numbers); Also expansion unit outputs at terminals A11 and B11 (all catalog numbers): No leakage current.

Processor unit outputs at terminals 12 thru 16 and 112 thru 116 of 1745-LP153, -LP156; Also expansion unit outputs at terminals A12 thru A16 and B12 thru B16 of 1745-E153, -E156: 2mA (AC voltage only). To limit leakage current, use a loading resistor across the load as shown on Page 20-15.

Output Specifications – SLC 150 Processor and Expansion Units (continued)

Specifications for hard contact relay outputs are shown below. We recommend that you use some type of surge suppression when switching inductive load devices with hard contact outputs. Refer to Page 19-9.

HARD CONTACT RELAY OUTPUTS

Wiring Terminals, Internal Arc Suppression:

Catalog Number	Hard Contact Relay Outputs
1745-LP151 1745-LP152 1745-LP154 1745-LP157	Units have 2 hard contact relay outputs, at terminals 11 and 111. These outputs do *not* have internal arc suppression circuitry.
1745-LP153 1745-LP156	Units have 12 hard contact relay outputs. Relay outputs at terminals 12 thru 16 and 112 thru 116 have internal arc suppression circuitry. Relay outputs at terminals 11 and 111 do *not* have internal arc suppression circuitry.
1745-E151 1745-E152 1745-E154 1745-E157	Units have 2 hard contact relay outputs, at terminals A11 and B11. These outputs do *not* have internal arc suppression circuitry.
1745-E153 1745-E156	Units have 12 hard contact relay outputs. Relay outputs at terminals A12 thru A16 and B12 thru B16 have internal arc suppression circuitry. Relay outputs at terminals A11 and B11 do *not* have internal arc suppression circuitry.

Voltage Range and Frequency:
10-250 VAC (50/60 Hz), 10-125 VDC.

Contact Ratings:

Maximum Volts	Amperes		Amperes Continuous	Voltamperes	
	Make	Break		Make	Break
240VAC 120VAC	7.5A 15A	0.75A 1.5A	2.5A	1800VA	180VA
125VDC	0.22A		1.0A	28VA	
24VDC	1.2A		2.5A	28VA	

Contact Resistance: 20 mΩ (typical).

Electrical Isolation:
2000 volts between output contacts and control logic.

OFF State Leakage Current:
Processor unit outputs at terminals 11 and 111 (all catalog numbers); Also expansion unit outputs at terminals A11 and B11 (all catalog numbers): No leakage current.

Processor unit outputs at terminals 12 thru 16 and 112 thru 116 of 1745-LP153, -LP156; Also expansion unit outputs at terminals A12 thru A16 and B12 thru B16 of 1745-E153, -E156: 2mA (AC voltage only). To limit leakage current, use a loading resistor across the load as shown on Page 20-15.

Specifications – SLC 150 High Speed Input Module

ON State Voltage Range:

5VDC	2-7VDC
12VDC	5-15VDC
24VDC	10-30VDC

Maximum OFF State Voltage:

5VDC	0.8VDC
12VDC	2VDC
24VDC	4VDC

Maximum OFF State Leakage Current:
1 mA (all voltage levels)

Nominal Input Current:

5VDC	25mA
12VDC	17mA
24VDC	15mA

Maximum Turn ON and OFF Time:
Filter not selected – 50 microseconds ± 25%
Filter selected – 4 milliseconds ± 25%

Electrical-Optical Isolation:
1500 volts between input voltage and control logic.

Noise Immunity:
NEMA Standard ICS 2-230

Vibration:
0.015 inch peak to peak displacement, 2.5g peak (max) acceleration, 1Hr/axis

Ambient Temperature Rating:
0°C to 60°C (operating). –40°C to 85°C (storage)

Humidity Rating:
5 to 95% (without condensation)

Wiring:
#14 to #24 AWG stranded. 3/64" insulation (max)

Specifications – Pocket Programmer

Catalog No. 1745-PT1

Display:
Light emitting diodes (LEDs) and 7-segment LED readouts.

Keyboard:
Thirty momentary push keys with tactile feedback.

Interconnect Cable:
Six-foot (1.8 m) cable. Mates with processor unit.

Operating Power:
Supplied by processor unit.

Ambient Temperature Rating:
0° to 50° C (operating).
– 40° to 85° C (storage).

Overall Dimensions (Approx.):
7x3x1 inches (178x76x25 mm).

Specifications – EEPROM Memory Module

Catalog No. 1745-M1

Memory Type:
Electrically-Erasable Programmable Read-Only Memory (EEPROM).

Memory Size:
1200 words maximum. Most instructions require 1 word.

Ambient Temperature Range:
0° to 60° C (operating).
– 40° to 85° C (storage).

Instruction Execution Times and Word Usage – SLC 100

Approximate instruction execution times in microseconds:

–] [– , –] / [–	Examine ON, Examine OFF	25
–( )–	Output Energize	30
–(L)–, –(U)–	Latch, Unlatch	32
–(RTO)–	Retentive Timer On-Delay	208
–(RTF)–	Retentive Timer Off-Delay	208
–(CTU)–	Up Counter	196
–(CTD)–	Down Counter	156
–(ZCL)–	Zone Control Last State	37
–(MCR)–	Master Control Reset	34
⊥	Branch Open	30
⊣	Branch Close	32
–(RST)–	Reset	156
–(SQO)–	Sequencer Output, time-driven	370
–(SQO)–	Sequencer Output, event-driven	362
–(SQI)–	Sequencer Input, time-driven	390
–(SQI)–	Sequencer Input, event-driven	378
–(SR)–	Shift Register	24

The I/O Scan execution time is approximately 1500 microseconds, plus 270 microseconds for each expansion unit.

Note: Execution times are approximate. You can measure your total program scan time by programming the rungs shown in Figure 8.6, Page 8-4.

Instruction Word Usage: The instructions above require one word of memory, except for the Reset and all versions of the Sequencer. These require

–(RST)–: 2 words.

–(SQO)–, –(SQI)–: $\frac{7 + 3(\text{number of steps})}{2}$ words

Instruction Execution Times and Word Usage – SLC 150

Approximate instruction execution times in microseconds:

–] [–, –] / [–	Examine ON, Examine OFF	2
–( )–	Output Energize	3
–(L)–, –(U)–	Latch, Unlatch	1
–(RTO)–	Retentive Timer On-Delay	50
–(RTF)–	Retentive Timer Off-Delay	50
–(CTU)–	Up Counter	43
–(CTD)–	Down Counter	43
–(ZCL)–	Zone Control Last State	2
–(MCR)–	Master Control Reset	3 + 3 x (no. of rungs in zone)
⊤	Branch Open	0
⊣	Branch Close	0
–(RST)–	Reset	50
–(SQO)–	Sequencer Output, time-driven	113
–(SQO)–	Sequencer Output, event-driven	113
–(SQI)–	Sequencer Input, time-driven	113
–(SQI)–	Sequencer Input, event-driven	113
–(SR)–	Shift Register	5

The I/O Scan execution time is approximately 300 microseconds, plus 250 microseconds for each expansion unit.

Note: Execution times are approximate. You can measure your total program scan time by programming the rungs shown in Figure 8.6, Page 8-4.

Instruction Word Usage: The instructions above require one word of memory, except for the Reset and all versions of the Sequencer. These require

–(RST)–: 2 words.

–(SQO)–, –(SQI)–: $\frac{7 + 3(\text{number of steps})}{2}$ words

Auto-Baud Recognition – SLC 150

The SLC 150 processor unit automatically adjusts its communication baud rate to allow communication with external devices such as computers and modems. Acceptable baud rates are 300, 1200, 2400, 9600 and 19,200.

This auto-baud feature is effective in any mode except the Run mode. If the SLC 150 is in the Run mode and you want to communicate at a different baud rate, remove power from the processor unit or enter any other operating mode. Then connect the external device you want to communicate with. If you remove power from the processor unit in the Run mode, the Auto/Man switch must be set to the Man position in order for the baud rate to be adjusted on power-up. Once power is applied, you can return the switch to the Auto position.

Chapter 19 System Layout Recommendations

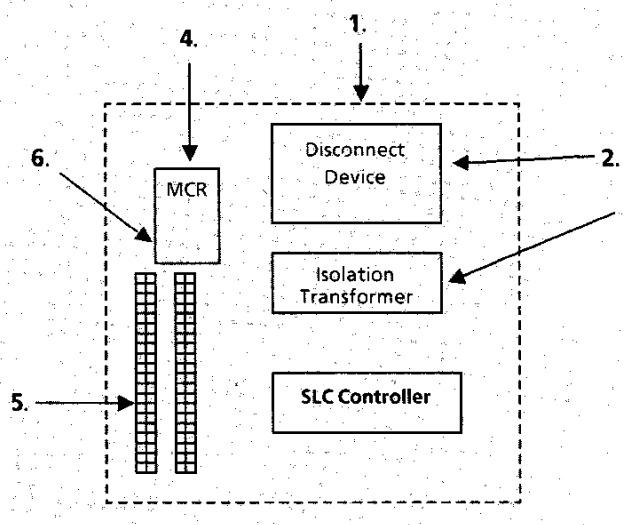
General

To help you install the controller as safely and securely as possible, we have set up a few specific recommendations and guidelines for you to follow. They are the result of product testing and Allen-Bradley's extensive experience with solid state industrial controls.

For general installation guidelines, also refer to article 70E of the National Fire Protection Association (NFPA). Article 70E describes electrical safety requirements for employee workplaces.

Figure 19.1 shows some of the components that make up a typical installation.

Figure 19.1



1. **NEMA rated enclosure suitable for your application and environment that will shield your controller from electrical noise and airborne contaminants.**
2. **Disconnect, to remove power from the system.**
3. **Fused isolation transformer or a constant voltage transformer, as your application dictates.**
4. **Master control relay/emergency stop circuit.**
5. **Terminal blocks or wiring ducts.**
6. **Suppression devices for limiting EMI generation.**

Enclosure Considerations

The enclosure protects the equipment from atmospheric contamination. Standards established by the National Electrical Manufacturer 's Association (NEMA) define enclosure types, based on the degree of protection an enclosure will provide. Select a NEMA rated enclosure that is suitable for your application and environment.

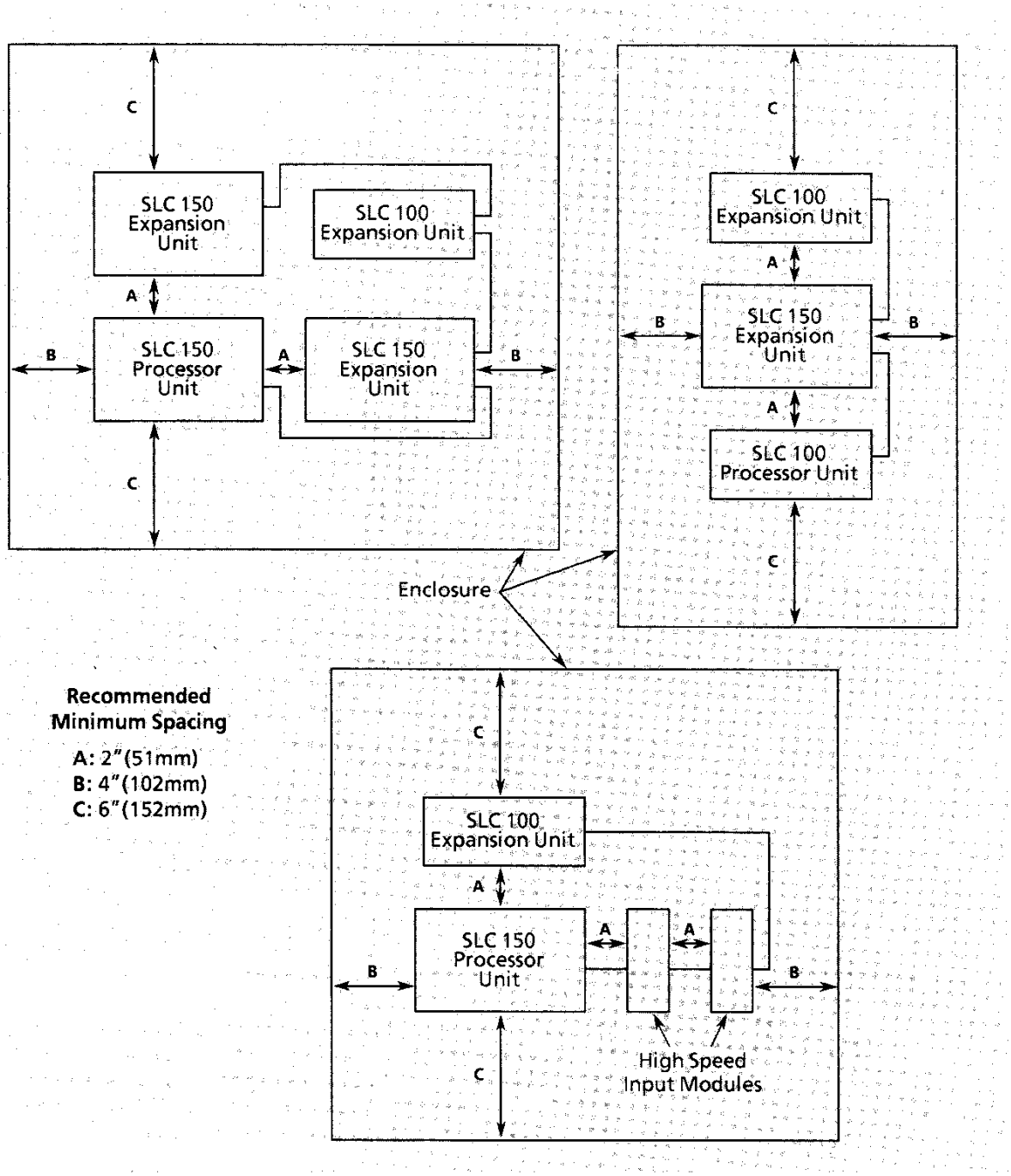
The enclosure must be equipped with a disconnect device.

Component Spacing

The figure below shows acceptable layouts. Follow the recommended minimum spacing to allow for convection cooling within the enclosure. Cooling air in the enclosure must be kept within a range of 0° to 60°C.

Note that SLC 100 and SLC 150 expansion units can be interconnected with either an SLC 100 or an SLC 150 processor unit. High speed input modules can be used with the SLC 150 processor unit only.

Figure 19.2



Recommended Minimum Spacing

A: 2"(51mm)
B: 4"(102mm)
C: 6"(152mm)

Excessive Heat

For most applications, normal convection cooling will keep the controller components within the specified operating range. Proper spacing of components within the enclosure is usually sufficient for heat dissipation.

Additional cooling provisions might be necessary when high ambient temperatures are encountered.

In some applications, a substantial amount of heat is produced by other equipment inside or outside the enclosure. In this case, blower fans may be placed inside the enclosure to assist in air circulation and to reduce "hot spots" near the controller. **Do not** bring in unfiltered outside air. It may introduce harmful contaminants or dirt that could cause improper operation or damage to components. In extreme cases, air conditioning may be required to protect against heat build-up within the enclosure.

Wiring Layout

Careful wire routing within the enclosure helps to cut down electrical noise between I/O lines. Follow these rules for routing your wires:

1. Route incoming power to the processor and expansion units by a separate path from wiring to I/O devices. Do **not** run signal wiring and power wiring in the same conduit! Where paths must cross, their intersection should be perpendicular.
2. If wiring ducts are used, allow at least 2 inches between I/O wiring ducts and the processor or expansion unit. If terminal strips are used for I/O wiring, allow at least 2 inches between the terminal strips and the processor or expansion unit.
3. Segregate I/O wiring by signal type. Bundle wiring with similar electrical characteristics together.

   Wiring with different signal characteristics should be routed into the enclosure by separate paths whenever possible.

Grounding

In solid state control systems, grounding helps limit the effects of noise due to electromagnetic induction (EMI). The grounding path for the controller and its enclosure is provided by the equipment grounding conductor.

> **WARNING:** The SLC controller, other control devices, and the enclosure must be properly grounded. All applicable codes and ordinances must be observed when wiring the controller system.

Ground connections should run from the CHASSIS GND terminal on each processor and expansion unit to the ground bus. Ground wires can be daisy-chained between units to minimize the number of ground wire connections. Exact connections will differ between applications. An authoritative source on grounding requirements for most installations is the National Electrical Code.

Grounding (continued)

In addition to the grounding required for the controller and its enclosure, you must also provide proper grounding for all controlled devices in your application. Care must be taken to provide each device with an acceptable grounding path.

Master Control Relay

A hard-wired master control relay (supplied by you) provides a convenient means for emergency controller shutdown. Since the master control relay allows the placement of several Emergency Stop switches in different locations, its installation is important from a safety standpoint. Overtravel limit switches or mushroom-head push buttons are wired in series so that when any of them opens, the master control relay is de-energized. This will remove power from the machine. Never alter these circuits to defeat their function, since serious injury or machine damage could result.

If you are using a DC power supply, interrupt the DC side rather than the AC side to avoid the additional delay of power supply turn-on and turn-off. As shown in Figure 19.3 on the following page, the DC power supply should receive its power directly from the fused secondary of the transformer. Power to the DC input and output circuits must be connected through a set of master control relay contacts.

The main power disconnect switch should be located where operators and maintenance personnel have quick and easy access to it. If a disconnect switch is to be mounted inside the controller enclosure, the switch operating handle should be on the outside of the enclosure, so that power can be disconnected without opening the enclosure.

Wiring the master control relay into a grounded system is shown in Figure 19.3. Whenever any of the Emergency Stop switches are opened, power to input and output devices is removed.

When the master control relay is used to remove power from the external I/O circuits, power continues to be provided to the controller's power supply so that diagnostic indicators on the processor can still be observed.

NOTE – The master control relay is **not** a substitute for a disconnect to the controller. It is intended for any situation where the operator must quickly de-energize I/O devices **only**. When inspecting or installing terminal connections, replacing output fuses, or working on equipment within the enclosure, the disconnect must be used to shut off power to the rest of the system.

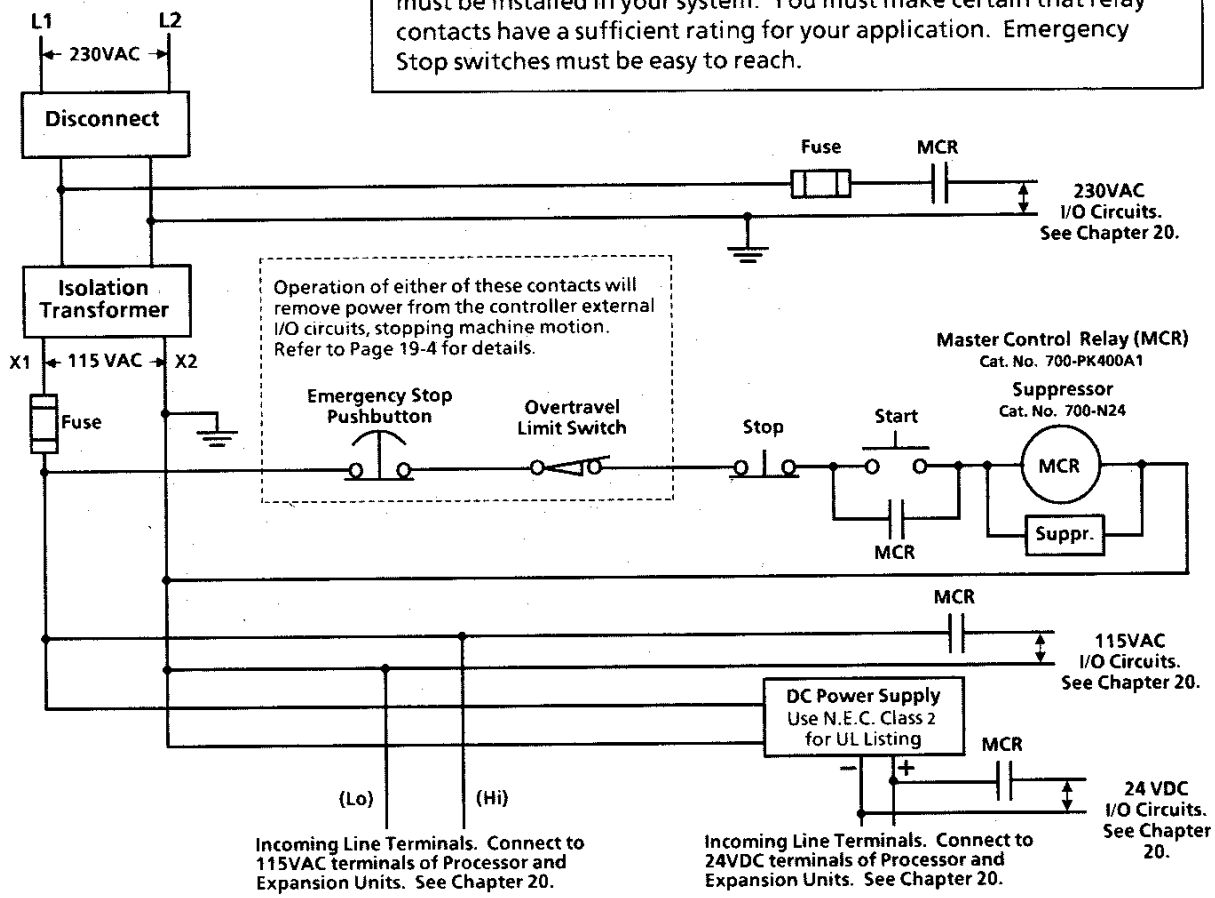
Master Control Relay *(continued)*

The master control relay must **not** be controlled by the SLC controller. The operator must be provided with the additional safety of a direct connection between an emergency stop switch and the master control relay.

> **WARNING:** Do **not** program Emergency Stop switches in the controller program. Any Emergency Stop switch should turn off all machine power by turning off the master control relay.
>
> Observe all applicable local codes concerning the placement and labeling of Emergency Stop switches.

Figure 19.3

> **CAUTION:** Emergency Stop switches and the master control relay must be installed in your system. You must make certain that relay contacts have a sufficient rating for your application. Emergency Stop switches must be easy to reach.



Refer to Chapter 20 for details on line wiring connections and I/O wiring connections to SLC components.

Common Power Source

We strongly recommend that the processor unit and all expansion units interconnected with it have the same power source as the input and output devices. This inhibits electrical interference.

The processor unit power supply monitors the incoming voltage. When it detects low line voltage, the monitor signals the processor to stop receiving input data. At the same time, the controller will turn all outputs OFF and the CPU FAULT LED will be lit.

In addition to monitoring incoming voltage at its power terminals, the processor unit detects the absence of power to any expansion unit it is interconnected with. If power to an expansion unit is lost (or not yet applied), the CPU fault LED will be lit and all controller outputs will be turned off.

This fault detection makes it necessary that you apply power to the expansion units *before* you apply power to the processor unit, to avoid an unwanted CPU fault. Of course, applying power in sequence is unnecessary if the processor and expansion units have a common power source.

Isolation Transformer

You may want to use an isolation transformer in the AC line to the controller. See Figure 19.3 (Page 19-5). This type of transformer provides isolation from your power distribution system and is often used as a "step down" transformer to reduce line voltage. Any transformer used with the controller must have a sufficient power rating for its load. This power rating is generally expressed in volt-amperes (VA).

To supply the worst-case power to the controller power supply, an isolation transformer should have a rating of approximately 3 times the maximum power requirement of the processor and expansion units. This allows ample power to be delivered to the power supply throughout the AC cycle. This also allows sufficient power for most types of input devices connected to the controller. If output devices are connected through the transformer as in Figure 19.3, their maximum VA requirements must be added to determine correct transformer size.

Special Considerations

The recommendations given previously will provide favorable operating conditions for most controller installations. Your application may involve one or more of the following adverse conditions. Additional measures can be taken to minimize the effect of these conditions.

Excessive Line Voltage Variations

The best solution to excessive line voltage variation is to correct any feeder problems in your distribution system. Where this does not solve the line variation problem, or in certain critical applications, a constant voltage transformer can be used. If a constant voltage transformer is required, it must be connected to the controller power supply **and** all input devices connected to the SLC controller. Output devices should be connected on the same power line, but their connection along the power line is normally made before the constant voltage transformer. A constant voltage transformer must have a sufficient power rating for its load.

Excessive Noise

When the SLC controller is operating in a "noise-polluted" industrial environment, special consideration should be given to possible electrical interference.

The effect of electrical interference is reduced by the following:

- SLC controller design features.
- Proper mounting of controller within an enclosure.
- Proper equipment grounding.
- Proper routing of wiring.

NOTE – To increase the operating noise margin, we recommend that you provide suppression for noise generating devices.

Excessive Noise (continued)

Potential noise generators include inductive loads, such as relays, solenoids, and motor starters when operated by "hard contacts" like push buttons or selector switches. Suppression may be necessary when such loads are connected as output devices or when connected to the same supply line that powers the controller. Figure 19.4 lists Allen-Bradley load devices and recommended suppressors. Figure 19.5 shows how suppressors are used.

If an SLC 150 triac output (1745-LP151, -LP152, -E151, -E152) is connected with a hard contact to control an inductive load, we recommend that you use varistors to suppress noise. Do not use suppressors having RC networks, since damage to triacs could occur. Allen-Bradley AC surge suppressors which are not recommended for use with triacs include Catalog Nos. 599-K04, 599-KA04, 199-FSMA1, 199-FSMA2, 1401-N10, and 700-N24.

Applications such as high frequency welding equipment and large AC motors generate excessively high levels of noise pollution. In these applications, all possible sources of noise should be suppressed. Best results are achieved when the noise suppressing networks are connected as closely as possible to the noise generating device.

Figure 19.4

Bulletin 509 Motor Starter Bulletin 509 Motor Starter	120 VAC 240 VAC	599-K04 599-KA04
Bulletin 100 Contactor Bulletin 100 Contactor	120 VAC 240 VAC	199-FSMA1 199-FSMA2
Bulletin 709 Motor Starter	120 VAC	1401-N10
Bulletin 700 Type R, RM Relays	AC coil	None required
Bulletin 700 Type R Relay Bulletin 700 Type RM Relay	12 VDC 12 VDC	700-N22 700-N28
Bulletin 700 Type R Relay Bulletin 700 Type RM Relay	24 VDC 24 VDC	700-N10 700-N13
Bulletin 700 Type R Relay Bulletin 700 Type RM Relay	48 VDC 48 VDC	700-N16 700-N17
Bulletin 700 Type R Relay Bulletin 700 Type RM Relay	115-125 VDC 115-125 VDC	700-N11 700-N14
Bulletin 700 Type R Relay Bulletin 700 Type RM Relay	230-250 VDC 230-250 VDC	700-N12 700-N15
Bulletin 700 Type N, P, or PK Relay	150V max, AC or DC	700-N24
Miscellaneous electromagnetic devices limited to 35 sealed VA		

Output Contact Protection

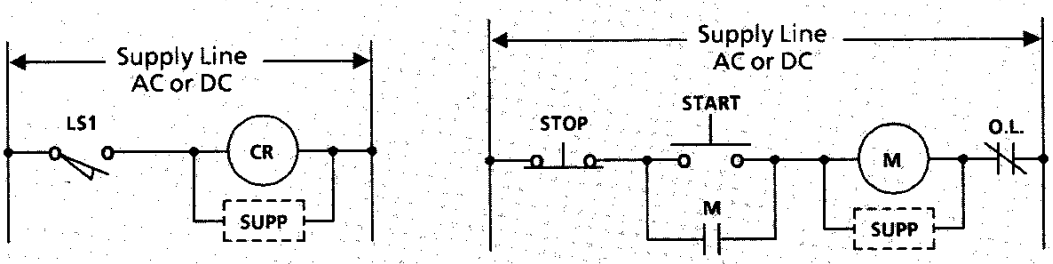
Inductive output devices such as motor starters and solenoids may require the use of some type of surge suppression to protect the controller output contacts. Examples are shown in Figure 19.6.

These surge suppression circuits are connected directly across the output device. The effect is to reduce arcing of the output contacts (arcing can be caused by the high transient voltage which occurs when switching off an inductive device). Suitable surge suppression methods for inductive AC output devices include a varistor, an RC network, and an Allen-Bradley surge suppressor (see Figure 19.4). These components must be appropriately rated to suppress the switching transient characteristic of the particular inductive device.

For inductive DC output devices, a diode is suitable. A 1N4004 diode is acceptable for most applications. A surge suppressor can also be used, as indicated in Figure 19.4.

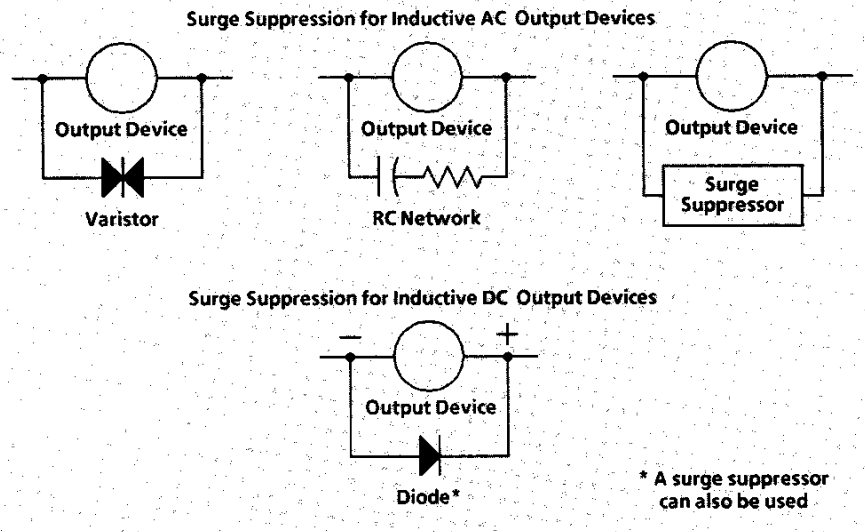
We recommend that you locate the suppression device as close as possible to the output device.

Figure 19.5



Examples of inductive loads operated by "hard contacts". Suppressors are connected directly across the coils. A list of Allen-Bradley suppressors appears in Figure 19.4.

Figure 19.6



Contact protection methods for inductive AC and DC output devices. A list of Allen-Bradley suppressors appears in Figure 19.4.

Chapter 20 Installation Procedure

General

The following paragraphs will show you how to

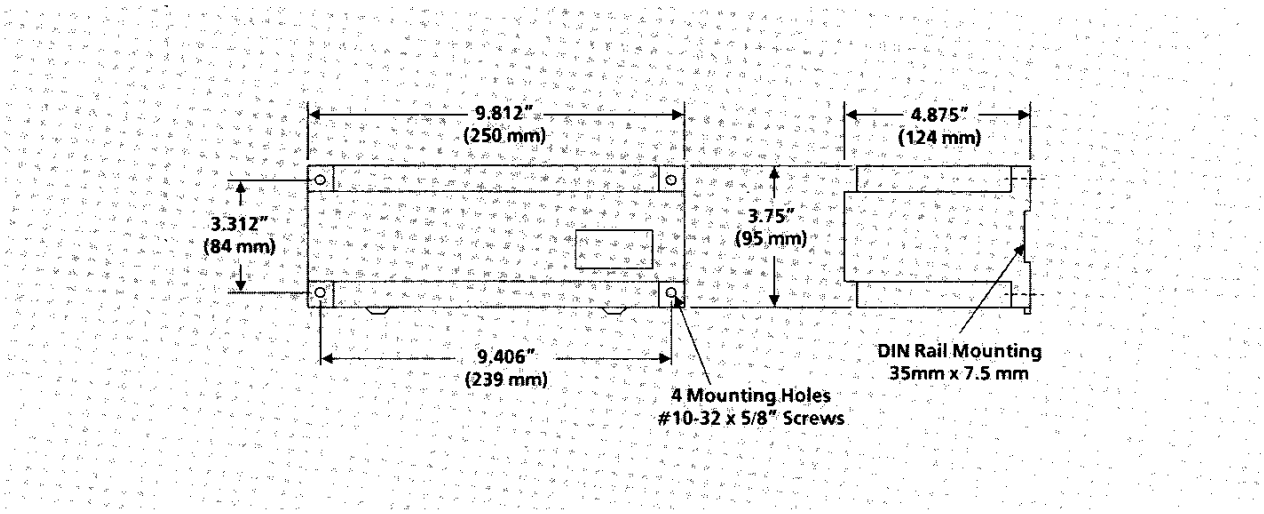
1. Mount the processor, expansion units, and high speed input module.
2. Install connection cables.
3. Wire input devices, output devices, and incoming power.

IMPORTANT: Before you begin, plan your system layout according to the guidelines in Chapter 19, System Layout Recommendations.

Mounting the SLC 100 Processor and Expansion Units

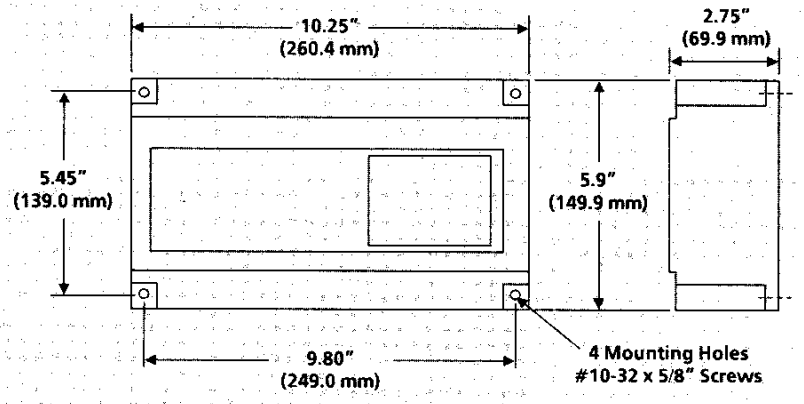
1. Screw mounting: The processor unit can be mounted directly to the back panel of your enclosure using four #10 screws. Hole locations are shown in the dimension drawing below.
2. DIN rail mounting: The processor unit can be mounted in your enclosure on a 35 mm by 7.5 mm DIN mounting rail (Catalog No. 199-1DR). Two DIN rail fasteners are provided on the processor unit.

   The DIN rail can be screwed, bolted, or welded to the enclosure back panel. Install the processor unit by hanging the unit on the top edge of the DIN rail, then pressing the unit toward the rail until it snaps into place. To remove the processor unit, pry open the fasteners.



Mounting the SLC 150 Processor and Expansion Units

Dimensions are shown below. Take note of the important mounting recommendations. If you are using a high speed input module with the processor unit, allow panel space to the right of the processor unit. Refer to Page 19-2 for layout recommendations.

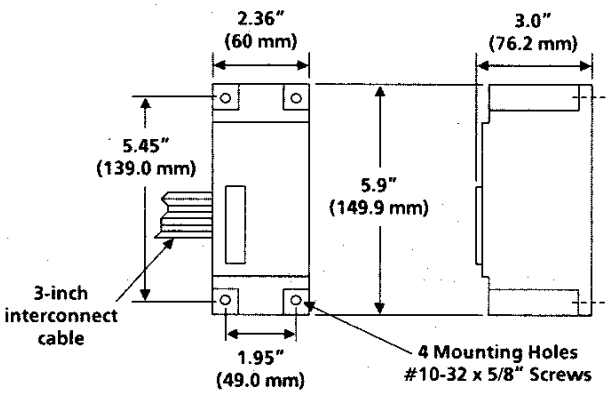


IMPORTANT: SLC 150 processor and expansion units utilize the mounting back panel as a heat sink for dissipating excess heat during operation. It is important that you mount the unit on a smooth metal back panel to provide good thermal conductivity. Back panel temperature specification: 60°C maximum.

Mounting on a sub panel – In high ambient temperatures (60°C), sub panels not exposed to air outside the enclosure could heat up beyond 60°C. In these situations, the sub panel temperature can be reduced by removing other heat generating equipment from the enclosure or by providing fans or air conditioning for cooling.

Mounting the SLC 150 High Speed Input Module

Mount directly to the back panel of your enclosure using four #10 screws. The module must be mounted directly to the right of the 150 processor unit or directly to the right of another high speed input module. Refer to Page 19-2 for layout recommendations.

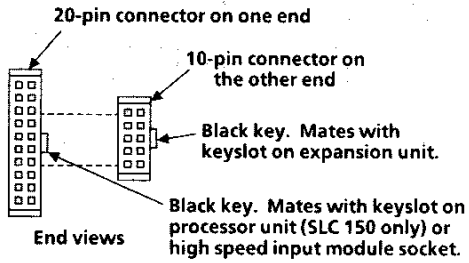


Connection Cables

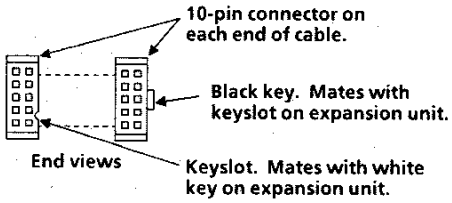
Two cables are used for interconnecting processor and expansion units: Interconnect cable 1745-C2 is a 10-pin to 10-pin cable supplied with each expansion unit. Interconnect cable 1745-C3 is a 20-pin to 10-pin cable supplied with the SLC 150 processor unit. Both cables are 18.5 inches (47 cm) long. See the figure below for cable connector details.

Important: Do not use cables longer than those provided. Longer cables could affect the integrity of data communications between the processor and expansion units, possibly causing unsafe operation.

1745-C3 interconnect cable supplied with SLC 150 processor unit



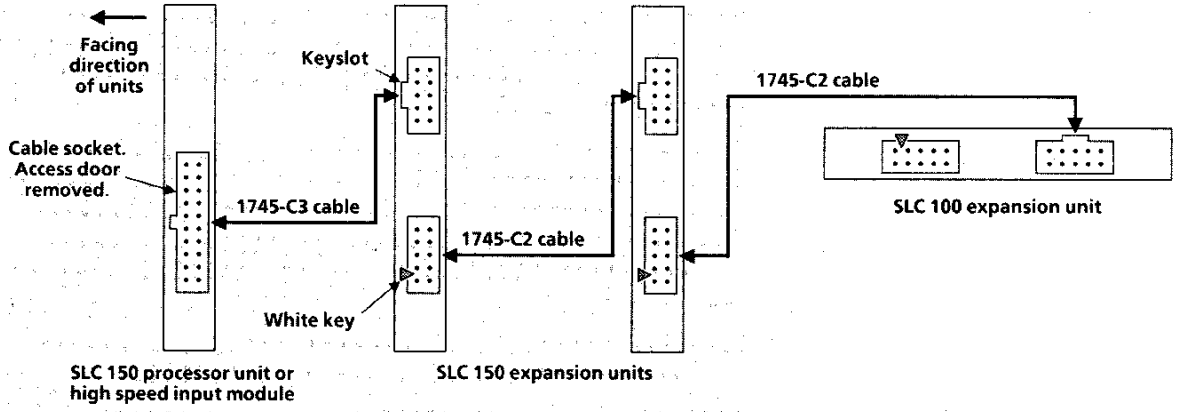
1745-C2 interconnect cable supplied with expansion units



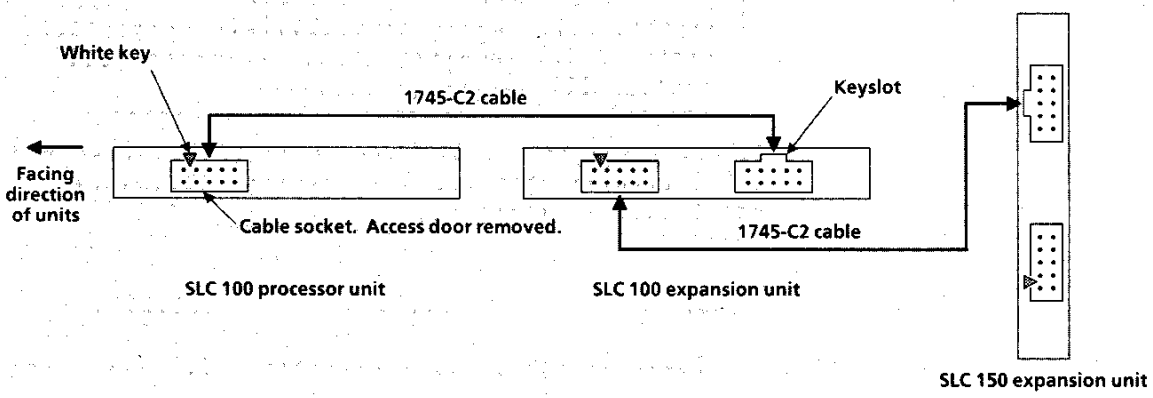
Connection Cables *(continued)*

The figure below explains how to install the interconnect cables.

A – Connecting expansion units to an SLC 150 processor unit



B – Connecting expansion units to an SLC 100 processor unit



Connecting an expansion unit to an SLC 150 processor unit:

Use cable 1745-C3, supplied with the SLC 150 processor unit.

1. Open the access doors.
2. Refer to illustration A. Align the 20-pin cable connector with the socket on the processor unit. Push gently; tabs will lock the connector in place.
3. Align the 10-pin connector with the expansion unit socket having the keyslot. Push gently; tabs will lock the connector in place.
4. Close the access doors.

Connecting an expansion unit to a high speed input module:

Same as above, but remove the plug from the high speed input module socket.

Connecting an expansion unit to an SLC 100 processor unit:

Use cable 1745-C2, supplied with the expansion unit.

1. Open the access doors.
2. Refer to illustration B. Align the cable connector having the keyslot with the socket on the processor unit. Push gently; tabs will lock the connector in place.
3. Align the connector having the black key with the expansion unit socket having the keyslot. Push gently; tabs will lock the connector in place.
4. Close the access doors.

Connecting one expansion unit to another:

Use cable 1745-C2, supplied with the expansion unit.

1. Open the access doors.
2. Refer to illustrations A and B. On the expansion unit already connected to another unit: Align the cable connector having the keyslot with the socket. Push gently; tabs will lock the connector in place.
3. On the other expansion unit: Align the connector having the black key with the socket having the keyslot. Push gently; tabs will lock the connector in place.
4. Close the access doors.

To remove cable: Move tabs on socket outward; connector will pop out.

Wiring I/O Devices and Incoming Power

General wiring recommendations:

- **Use acceptable wire gauge:** The I/O wiring terminals are designed to accept up to two #14 AWG stranded wires.
- **Label wires:** All wiring to I/O devices, power sources, and ground should be appropriately labeled. Tape, shrink-tubing, or other dependable means can be used for labeling purposes. In addition to labeling, insulation color may be used to identify wiring based on signal characteristics. For example, DC I/O wiring may be blue; AC I/O wiring may be red.
- **Bundle wires:** Wiring for each I/O device should be bundled together. If you use ducts, allow at least 2 inches between wiring bundles and ducts so there is sufficient room to wire the devices. Power and ground wires, of appropriate gauge, may be used as jumpers from one I/O terminal to the next, if the I/O terminals are the same voltage.
- **Identify terminals:** Terminal coverplates have a write-on area for each terminal. Use this area to identify your I/O devices.

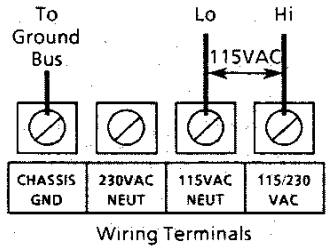
> **CAUTION:** Calculate the maximum possible current in each power and common wire. Observe all local electrical codes dictating the maximum current allowable for each wire size. Current above the maximum ratings may cause wiring to overheat which may damage equipment.

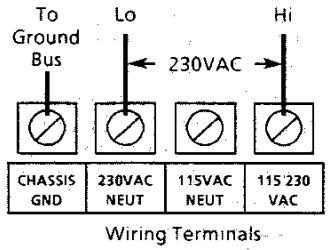
Before you install and wire the I/O devices, disconnect power from the controller and any other source to the I/O devices. Remove the protective terminal coverplates.

Line Wiring Connections – SLC 100

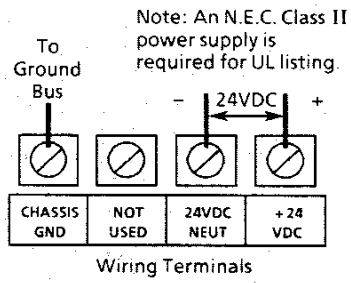
Make line connections to the processor and expansion units as follows:

> **CAUTION:** Incorrect wire connections can cause damage to the processor unit power supply. **Do not** jumper 115VAC NEUT and 230VAC NEUT together. **Do not** jumper unused 115VAC NEUT or unused 230VAC NEUT to the CHASSIS GND terminal.





Catalog Nos. 1745-LP101, -LP102, -LP103, -E101, -E102, -E103, -E105, -E106



Catalog Nos. 1745-LP104, -E104, -E107

Discrete Input Wiring Connections –SLC 100

SLC 100 input circuitry is current sinking. Circuitry includes optical isolation and surge suppression to guard against damage by transients from user input devices.

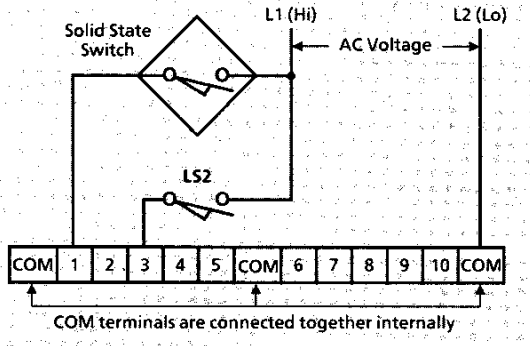
A direct interface to solid state sensing devices and SLC 100 controller output circuits is possible because of the 2 mA OFF state leakage current specification, indicated below. Note that the 2 mA value applies only to inputs 1 and 2 in some units. You can achieve a 2 mA current value for inputs 3 thru 10 by adding a resistor, as indicated.

Catalog Number		Maximum OFF State Leakage Current
Processor Unit	Expansion Unit	
1745-LP101 1745-LP102	1745-E101 1745-E102	2 mA, all inputs
1745-LP103 1745-LP104	1745-E103 1745-E104	2 mA, inputs 1 and 2 only ①

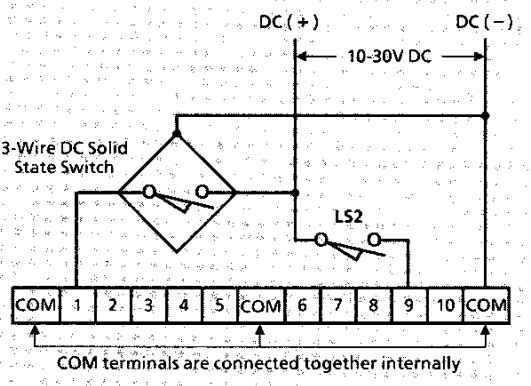
① To achieve a 2 mA OFF state leakage current for inputs 3 thru 10, connect a resistor from the input terminal to the COM terminal as shown in the wiring diagram.

Input terminal coverplates are color coded: 115VAC – red, 230VAC – black, 24VDC – blue. Wire input devices as shown below.

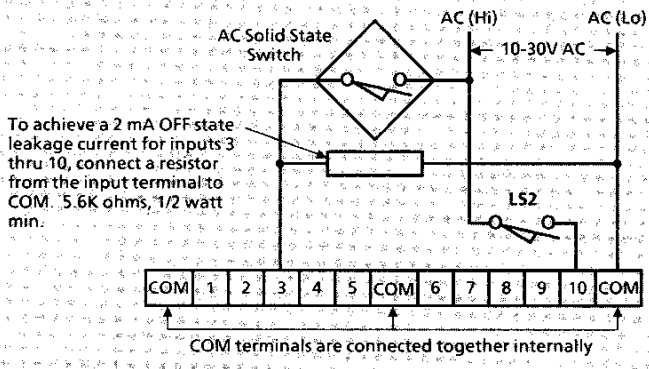
Input wiring terminal numbers in the diagrams (1 thru 10) apply to processor units. Terminals of expansion units are numbered 1E thru 10E.



1745-LP101, -LP102, -E101, -E102



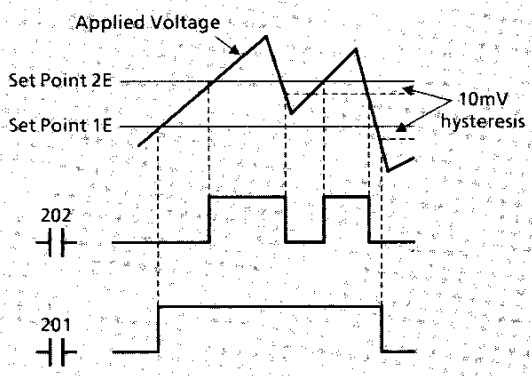
1745-LP103, -LP104, -E103, -E104 (DC input devices)



1745-LP103, -LP104, -E103, -E104 (AC input devices)

Analog Input Wiring Connections

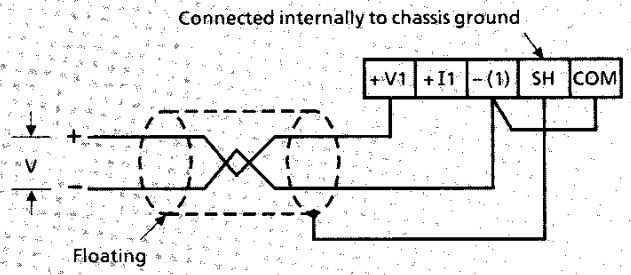
Wiring connections for the analog input expansion unit are shown below. These connection diagrams use a single analog input circuit, providing one or two programmable set points. If you require more than two set points per input circuit, use the connection diagrams on the next page.



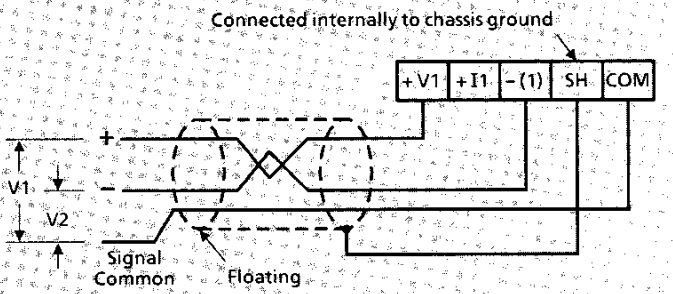
The following applies to the diagram above:

Analog input circuit number 1 is used, as in connection diagrams 1 and 2. Set points 1E and 2E apply. Arbitrarily, address block 3, addresses 201-208 is assigned. Examine ON instructions 201 and 202 are programmed for set points 1E and 2E.

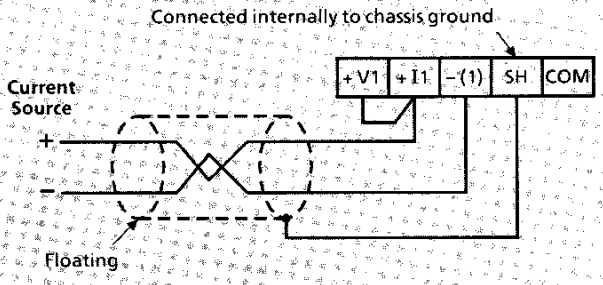
The "applied voltage" is V for connection diagram 1, or V1–V2 for diagram 2. When this voltage increases to set point 1E, instruction 201 goes TRUE. When the voltage increases to set point 2E, instruction 202 goes TRUE. When the voltage drops below these set point levels (note hysteresis), the respective instructions go FALSE.



1. Voltage source, single ended connection.



2. Voltage source, differential connection.



3. Current source. Terminal +V connected to terminal +1. COM not used.

APPLICATION NOTES

1. **+V must be 0-10 volts greater than (–).**
2. **+V and (–) must be kept within 35 volts of COM.**
3. **All COM terminals of the expansion unit are connected together internally.**
4. **All SH terminals are connected together internally, and are connected to chassis ground.**
5. **Important wiring recommendations will be found in Publication 1770-4.1, "Programmable Controller Grounding and Wiring Guidelines".**
6. **Adjustment: Use a screwdriver with a blade width of 0.1 inch or less. Procedure:**

   a) Apply the desired set point voltage or current to the input circuit. Measure the voltage or current, or simulate the condition for which the set point is to be adjusted.

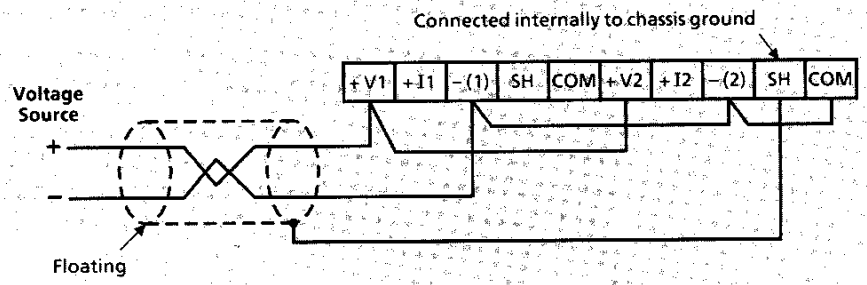
   b) Turn the adjustment potentiometer until the corresponding LED indicator lights. CW increases the setting. There are 18 turns between minimum and maximum setting.

   c) Verify the setting during controller Start Up procedures.

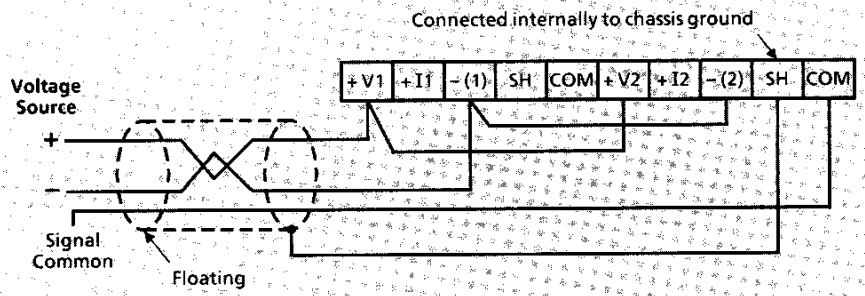
CAUTION: If an open circuit exists at the source end of the input circuit, the input could oscillate around the set point, adversely affecting user program operation.

Analog Input Wiring Connections *(continued)*

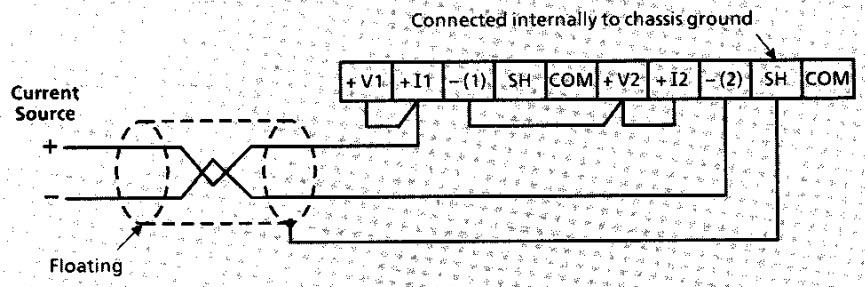
In these connection diagrams, two analog input circuits of the expansion unit are interconnected, providing four programmable set points (1E, 2E, 3E, and 4E in this case).



1. Voltage source, single ended connection.



2. Voltage source, differential connection.



3. Current source. COM terminal not used.

IMPORTANT: Refer to the APPLICATION NOTES and CAUTION box on Page 20-8.

Output Wiring Connections – SLC 100

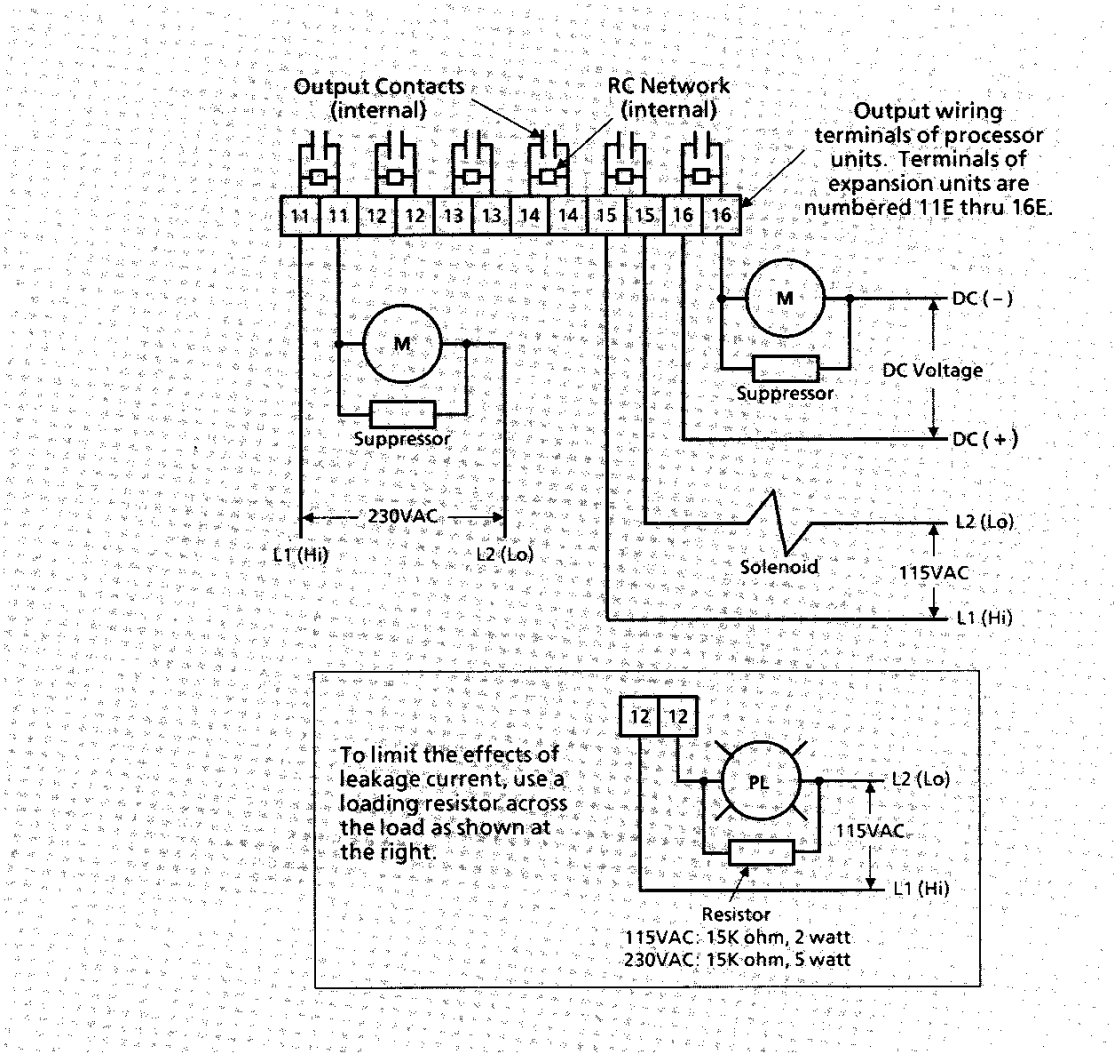
Wiring connections for output devices are shown below. Note that the diagram shows the internally-connected output contacts and parallel RC networks. The RC networks guard against possible damage by transients from external output devices.

External connections are shown for outputs 11, 15, and 16. We've added a suppressor in parallel with two of the external devices for the purpose of contact protection. Contact protection is discussed on Page 19-9.

In cases where you need to limit the effects of leakage current, add a resistor in parallel with the load as indicated in the diagram.

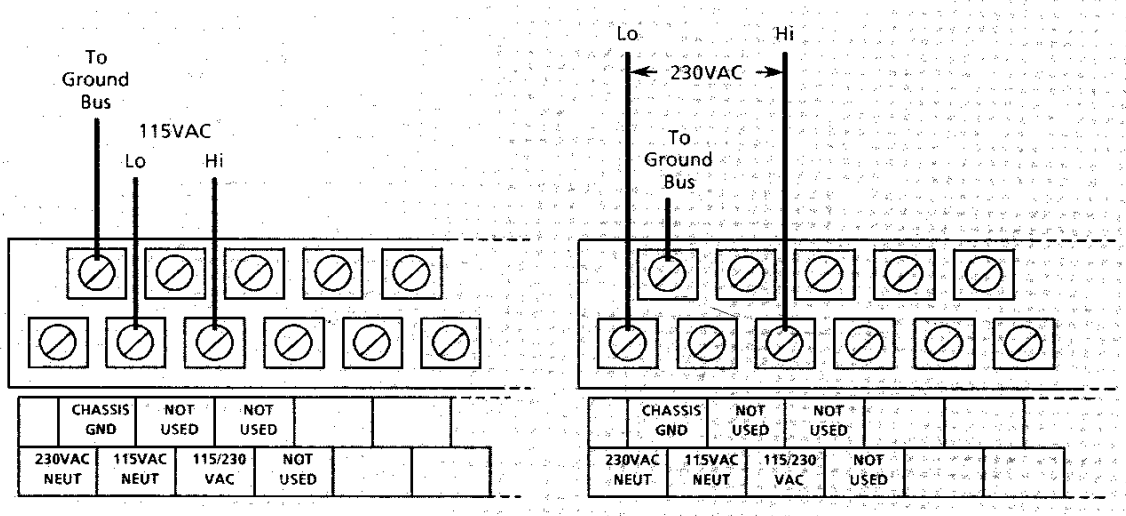
Since the output contacts are isolated from each other, each output circuit can be wired independently, with its own ground return. You can apply a different voltage in each output circuit, as your application might require. Power or ground wires can be jumpered between sets of terminals if desired.

You should provide appropriate fusing to protect the output devices and wiring from short circuits and overload conditions.

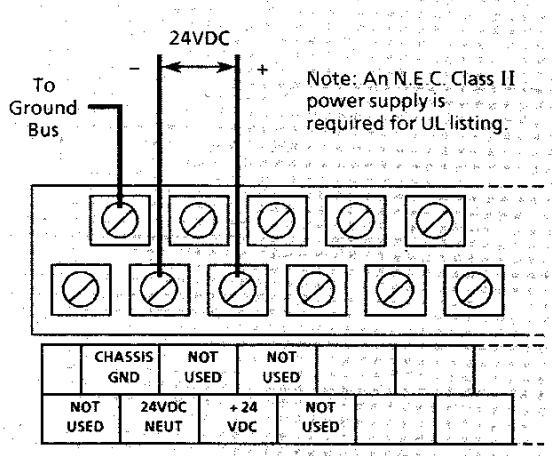


Line Wiring Connections – SLC 150

Make line connections to the processor and expansion units as follows.



Catalog Nos. 1745-LP151, -LP152, -LP153, -LP156, -E151, -E152, -E153, -E156

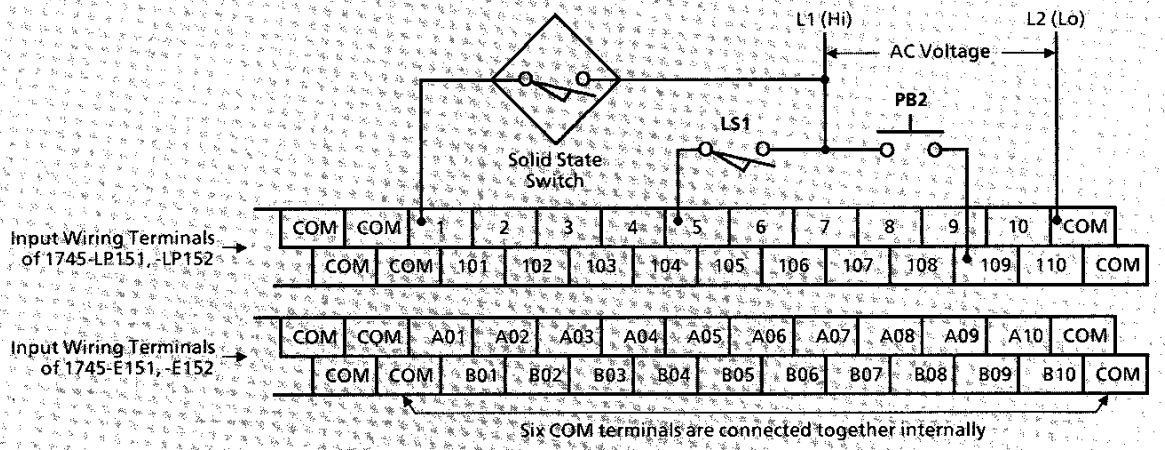


Catalog Nos. 1745-LP154, -LP157, -E154, -E157

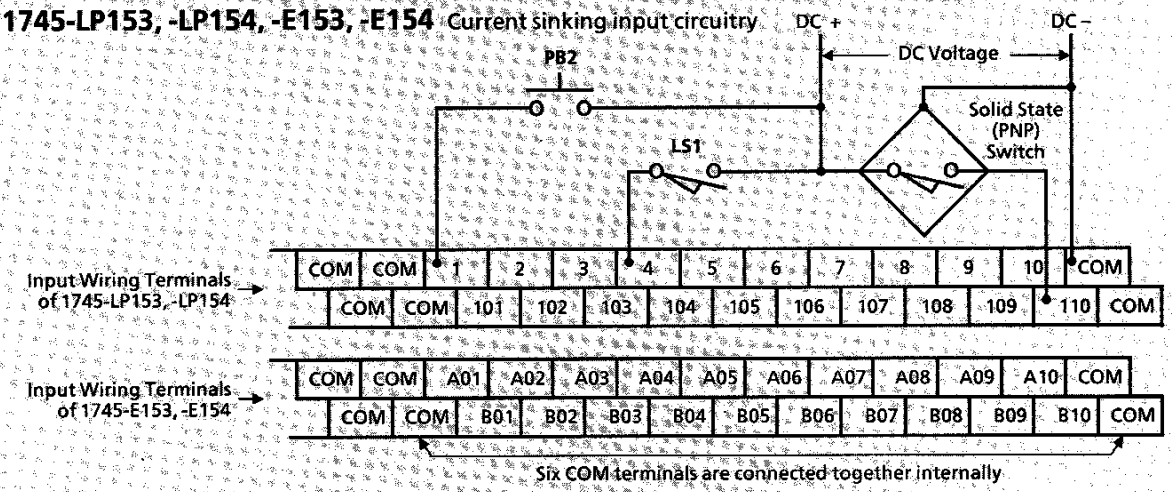
CAUTION: Incorrect wire connections can cause damage to the processor unit power supply. **Do not** jumper 115VAC NEUT and 230VAC NEUT together. **Do not** jumper unused 115VAC NEUT or unused 230VAC NEUT to the CHASSIS GND terminal.

Input Wiring Connections – SLC 150

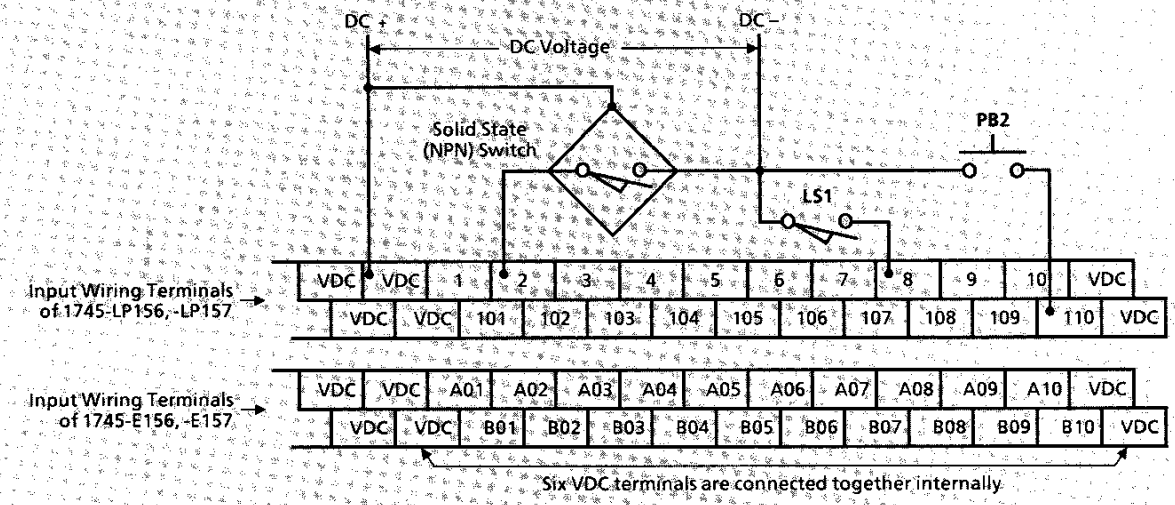
1745-LP151, -LP152, -E151, -E152 Current sinking input circuitry



1745-LP153, -LP154, -E153, -E154 Current sinking input circuitry



1745-LP156, -LP157, E156, -E157 Current sourcing input circuitry



Output Wiring Connections – SLC 150

Wiring connections are shown on Pages 20-14 and 20-15. Note that each processor and expansion unit has two isolated groups of outputs plus two additional isolated hard contact relay outputs.

Triac Outputs: Triac outputs include optical isolation as well as MOV protection to guard against possible damage by transients from external outputs. Triac output firing can be synchronized with the AC line to accomplish zero-cross turn-on and minimize noise generated when switching loads. This is accomplished by making instruction –(866)– TRUE in the user program. If this feature is used, your scan time will be 8.3 msec (or some multiple) at 60 Hz and 10 msec (or some multiple) at 50 Hz. A common power source **must be used** for the processor unit power supply and output circuits to achieve zero-cross turn on.

Since triacs turn off at AC line zero cross, it is not necessary to use external surge suppression when switching inductive loads. However, if hard contacts are connected with triacs to switch an inductive load, we recommend using varistors for external surge suppression. Do not use suppressors having RC networks, since damage to triacs could occur. Refer to Pages 19-8, 19-9 for further discussion on surge suppression.

Hard Contact Relay Outputs: Outputs at terminals 12-16 and 112-116 (A12-A16 and B12-B16) include arc suppression circuitry (RC networks) which protects contacts when switching inductive loads. We recommend that you also connect external surge suppression to protect the contacts from high transient voltage which occurs when an inductive device is switched off.

Hard contact relay outputs at terminals 11 and 111 (A11 and B11) do *not* include internal arc suppression. Contact protection/surge suppression: See Pages 19-8, 19-9.

Transistor Outputs: Catalog Numbers 1745-LP154 and 1745-E154 have current sourcing (PNP) transistor outputs. Catalog Numbers 1745-LP157 and 1745-E157 have current sinking (NPN) transisitor outputs.

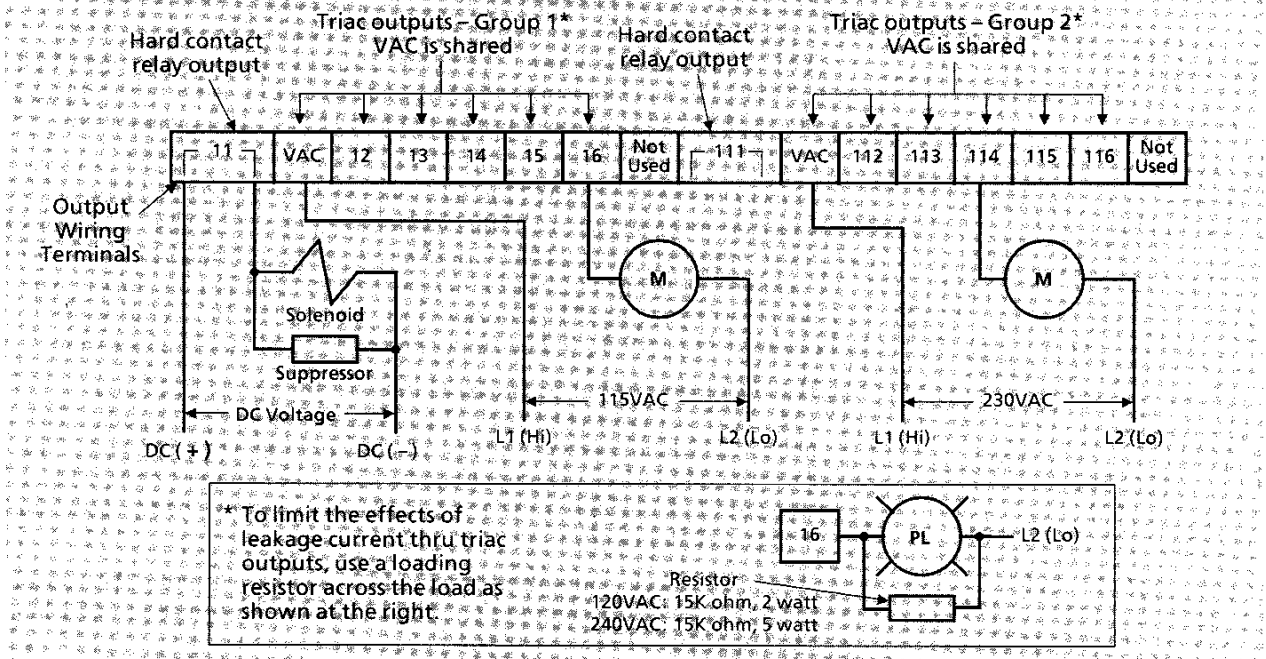
We recommend that you connect external surge suppression to protect transistors from the high transient voltage which occurs when an inductive device is turned off. An IN4004 diode is acceptable for most applications. Refer to Pages 19-8, 19-9 for further discussion on surge suppression.

Fusing: You should provide appropriate fusing to protect output devices and wiring from short circuits and overload conditions. Refer to Chapter 18, Page 18-9 for recommended fusing.

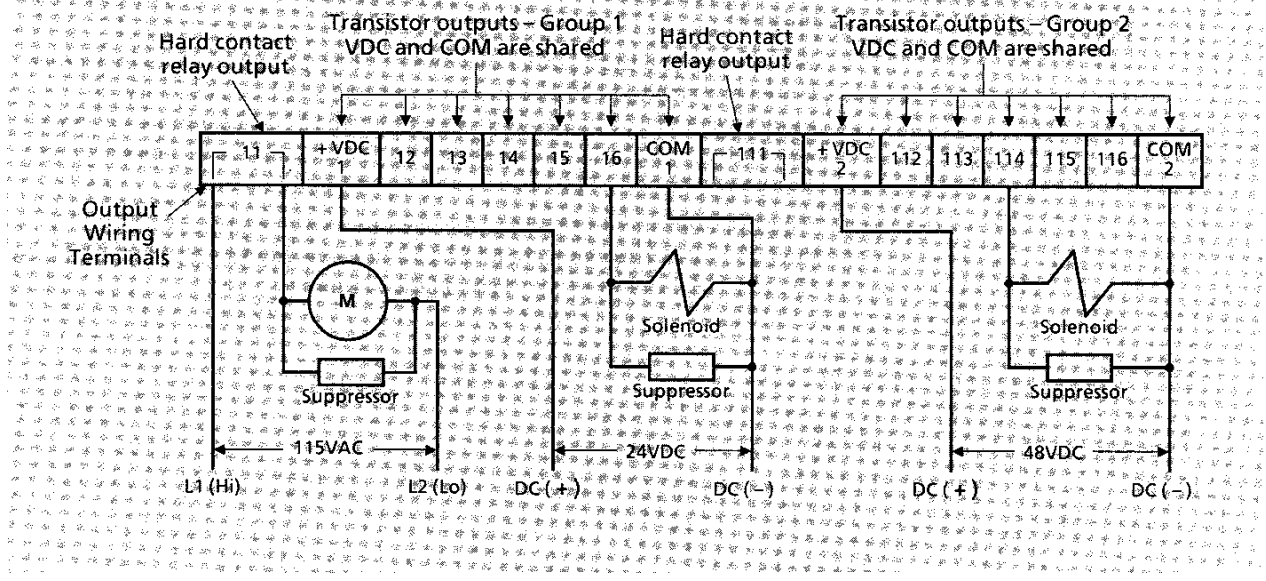
Output Wiring Connections – SLC 150 (continued)

Wiring connections for catalog numbers 1745-LP151, -LP152, -LP154, -E151, -E152, -E154 are shown below.

1745-LP151, -LP152, -E151, -E152 Wiring terminal numbers in the diagram (11 thru 16 and 111 thru 116) apply to 1745-LP151, -LP152. Terminal numbers of 1745-E151, -E152 are A11 thru A16 and B11 thru B16.



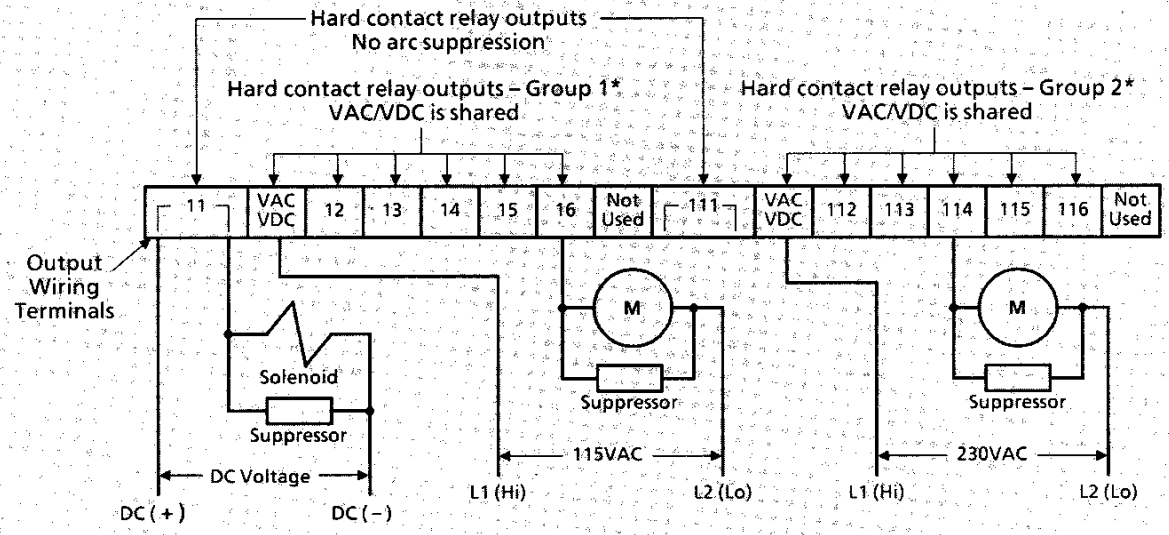
1745-LP154, -E154 Current sourcing (PNP) transistor outputs. Output wiring terminal numbers in the diagram (11 thru 16 and 111 thru 116) apply to 1745-LP154. Terminal numbers of 1745-E154 are A11 thru A16 and B11 thru B16.



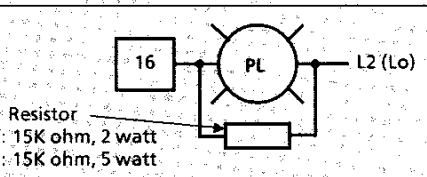
Output Wiring Connections – SLC 150 (continued)

Wiring connections for catalog numbers 1745-LP153, -LP156, -LP157, -E153, -E156, -E157 are shown below.

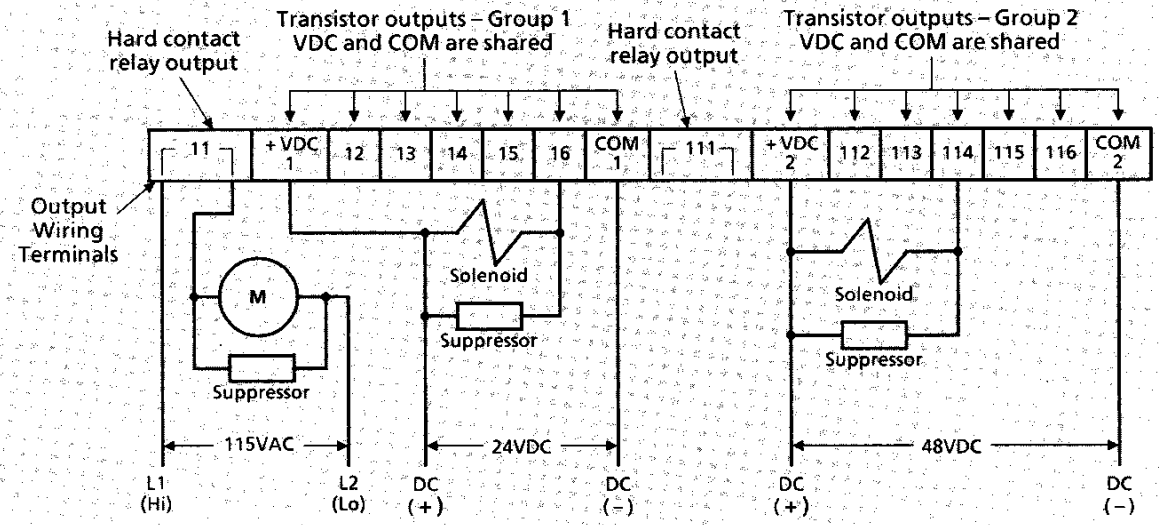
1745-LP153, -LP156, -E153, -E156 Output wiring terminal numbers in the diagram (11 thru 16 and 111 thru 116) apply to 1745-LP153, -LP156. Terminal numbers of 1745-E153, -E156 are A11 thru A16 and B11 thru B16.



* Group 1 and 2 hard contact relay outputs have built-in arc suppression (R = 120 ohms, C = 0.022 microfarad). To limit the effects of leakage current, use a loading resistor across the load as shown at the right.



1745-LP157, -E157 Current sinking (NPN) transistor outputs. Output wiring terminal numbers in the diagram (11 thru 16 and 111 thru 116) apply to 1745-LP157. Terminal numbers of 1745-E157 are A11 thru A16 and B11 thru B16.



Wiring Connections – SLC 150 High Speed Input Module

Typical connection diagrams are shown below and on the following page. DIP switches must be set for the input voltage range, input filter selection, and the encoder/count function. This is explained below.

DIP switches are located under the module cover, as shown at the right.

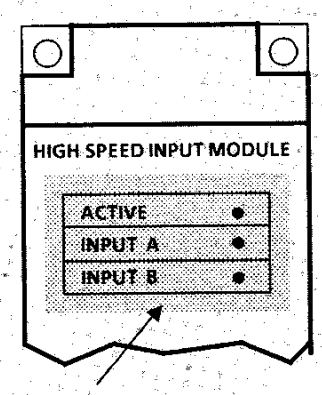
The tables below explain the DIP switch settings. The maximum input pulse rate is also listed.

Input	Input Voltage	Switch Settings
A	24VDC 12VDC 5VDC	1 OFF, 2 OFF 1 ON, 2 OFF 1 OFF, 2 ON
B	24VDC 12VDC 5VDC	3 OFF, 4 OFF 3 ON, 4 OFF 3 OFF, 4 ON

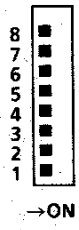
Input	Filter Selection	Switch Settings
A	4 millisec. 50 microsec.	6 ON 6 OFF
B	4 millisec. 50 microsec.	7 ON 7 OFF

Function	Switch Settings
Count Select this function to count high speed input signals received at terminal A/COUNT	8 ON
Encoder Select this function to interface to quadrature type encoder	8 OFF

Maximum Pulse Rate Note: If maximum or total frequencies are exceeded, status bit 901-904 will be set ON				
Number of HSI Modules Connected	1	2	3	4
Highest Single Frequency (KHz)	5	4	3	2
Highest Total Frequency (KHz)	5	5	5	5

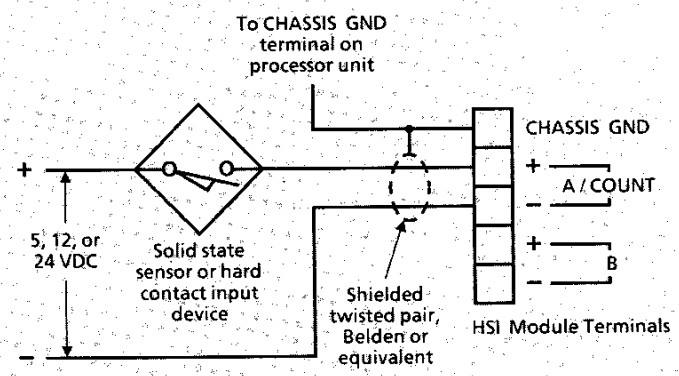


Lift here to gain access to the DIP switches.



DIP switches. The top switch is number 8 and the bottom switch is number 1. Switch number 5 is not used.

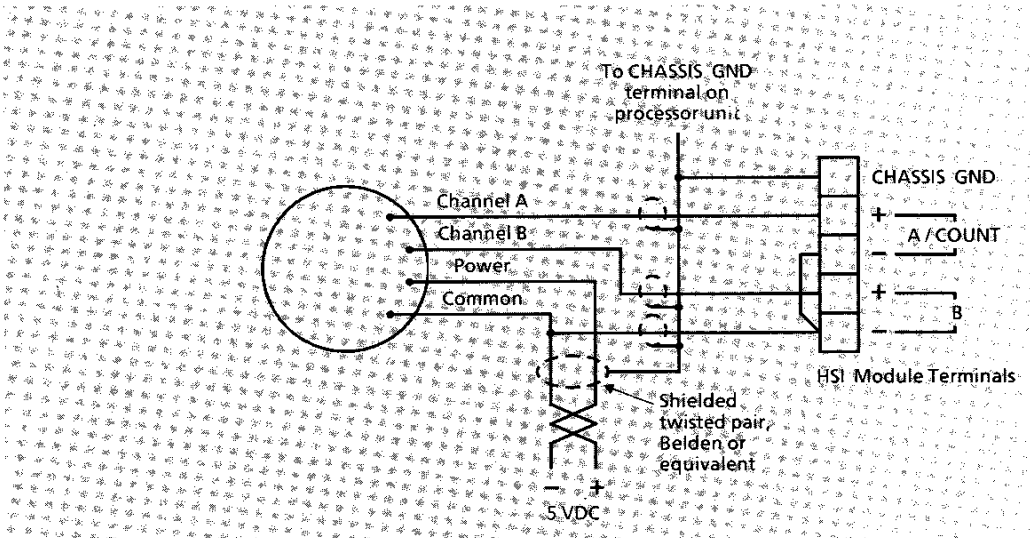
Move the switch to the right for the ON position.



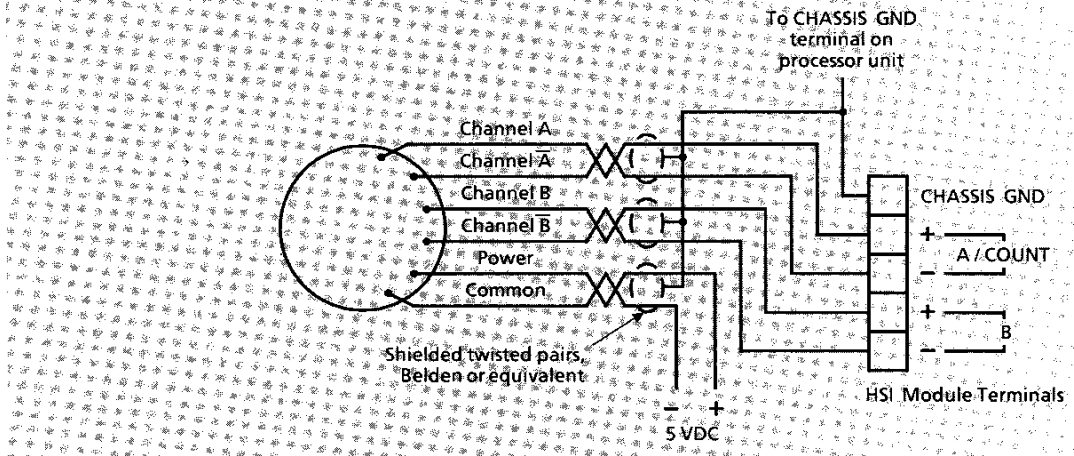
Connections for a solid state sensor or hard contact input device (Count application)

Wiring Connections – SLC 150 High Speed Input Module *(continued)*

The following wiring diagrams apply to single-ended and differential encoder applications. **Note:** The purpose of the B terminal of the HSI module is to indicate a direction change for input pulses. This function is not recognized by SLC 150 Series A processor units. In the diagrams below, the B channel of a quadrature type encoder is connected to the B terminals to help increase immunity to noise and shaft jitter.



Single ended wiring connections for a Bulletin 845N encoder



Differential wiring connections for a Bulletin 845N encoder

Chapter 21 Start-Up

General

Start-up involves the following procedures, to be carried out in sequence:

A. Inspect the installation before power is connected.

B. Disconnect motors and other devices which cause machine motion.

C. Test inputs.

D. Test outputs.

E. Enter and verify your program.

F. Test the system with motors and other motion-causing devices reconnected.

G. Go thru a dry run of the application.

The purpose of these procedures is to isolate such problems as wiring mistakes, equipment malfunction, and programming errors in a systematic, controlled manner.

We urge you to go thru these procedures very carefully. This will help avoid possible personal injury and equipment damage.

Do not attempt system start-up until you are thoroughly familiar with the controller components and programming/editing techniques discussed in earlier chapters. You must also be thoroughly familiar with the particular application.

Note: For general recommendations concerning installation safety requirements and safety related work practices, refer to NFPA 70E, Electrical Safety Requirements for Employee Workplaces.

SLC PERSONAL COMPUTER SOFTWARE USERS: This chapter is written with the assumption that you will be using a pocket programmer in the start-up procedures. If you are using the SLC Personal Computer Software, you will find that the multi-rung display and monitoring capabilities are valuable tools. However, the SLC Personal Computer Software does not allow you to use the FORCE function as we recommend in this chapter.

Inspecting the Installation (A)

You can often prevent serious problems in later test procedures by first making a thorough physical inspection. We recommend that you:

1. Make sure that the controller and all other devices in the system are securely mounted.
2. Check all wiring, including: connections from the main disconnect to the controller input; the master control relay/emergency stop circuit; input device circuits; output device circuits; expansion cables.

   Make certain that all wiring connections are correct and that there are no missing wires. Check the tightness of all terminals to make certain wires are secure.
3. Measure the incoming line voltage to be certain that it corresponds to controller requirements and that it falls within the specified voltage range. See specifications for input voltage ranges.

Disconnecting Motion-Causing Devices (B)

In the following test procedures, the controller will be energized. As a safety precaution, you must make certain that machine motion will not occur. The preferred way to disconnect a motion-causing device such as a motor is to disconnect the motor wires at the motor starter or the motor itself. In this way, you can test the operation of the starter coil, verifying that your output circuit is wired correctly and functioning. Similarly, the preferred way to disconnect a solenoid is to disengage the valve, leaving the coil connected.

In some instances, you may not be able to disconnect a device the preferred way. In this case, it will be necessary to open the output circuit at some convenient point.

For circuit testing purposes, it is best to open the circuit at a point as close as possible to the motion-causing device. For example, your output might be a relay coil which in turn energizes a motor starter; if it is impractical to disconnect the motor wires, the next best thing to do is to open the circuit at a point between the motor starter and the relay contact.

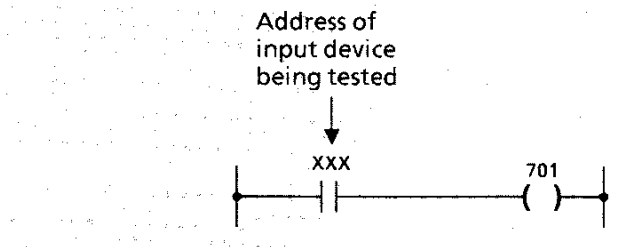
> **WARNING:** Machine motion during system checkout can be hazardous to personnel. During the following checkout procedures **C, D,** and **E,** you must disconnect all devices which, when energized, might cause machine motion.

Testing Inputs (C)

When you are certain that machine motion can not occur with the controller energized, you can begin testing inputs. Follow these steps:

1. Energize the controller.
2. Connect the pocket programmer and enter the Clear Memory mode (mode 1). Press ENTER when the display prompts you with SurE ? A clear memory is necessary for this test.
3. The controller will enter the Program mode automatically when the memory is cleared.

   Enter the test rung shown below. The Examine ON instruction should be assigned the address corresponding to the first input you will test. The Output instruction should be assigned an internal address, 701, to avoid involving output circuits at this time.



Input test rung

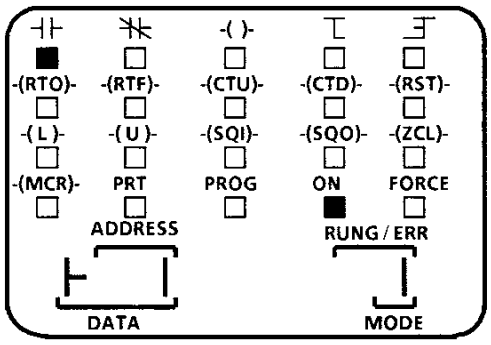
4. Place the controller in the Test–Continuous Scan mode (5). Cursor to the Examine ON instruction in the test rung with the programmer.

Testing Inputs (C)
(continued)

5. Manually close and open the addressed input device. Use a wooden stick for this purpose, as recommended below.

> **WARNING:** Never reach into a machine to actuate a switch, since unintentional machine motion can occur. Use a wooden stick or some other non-conductive device to guard against personal injury and electrical shock.

Observe the ON LED indicator on the pocket programmer and status LED on the processor or expansion unit. The LEDs should be ON (lit) when the input device is closed, and OFF when the input device is open. Say for example the first input device you wanted to test was an Examine ON instruction at address 001 in rung 1. When input 001 is ON, the display on the programmer would look like this:



Cursor to the output instruction at address 701. It should be ON when the input device is ON.

If the ON LED and test rung respond properly, testing of the first input is complete. Repeat this procedure for each input. To save time, begin at step 3, by entering the Program mode. Remove the Examine ON instruction of the test rung, and insert a new Examine ON instruction with the appropriate address.

We recommend that you do not program multiple rungs. This is a potential source of confusion and errors.

Troubleshooting: If a status indicator on the processor or expansion unit or the ON LED on the programmer fails to go ON when it should, check the following:

a) The input terminal power source.

b) The input circuit wiring.

c) The input device itself.

d) The expansion unit connection.

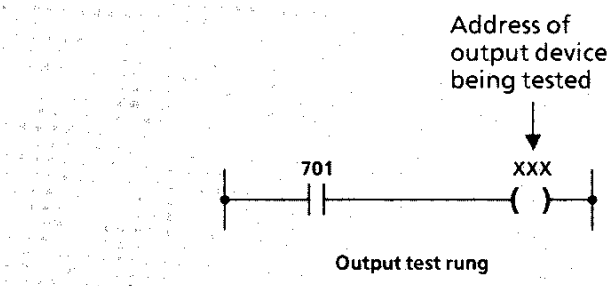
If the source of the problem is not found to be in a), b), c), or d), try connecting to another input terminal. If the problem still exists, the processor unit or pocket programmer may need replacement. Contact your local Allen-Bradley representative.

Testing Outputs (D)

After all inputs have been tested and are functioning properly, test the outputs. Once again, make certain that no machine motion can occur. Follow these steps:

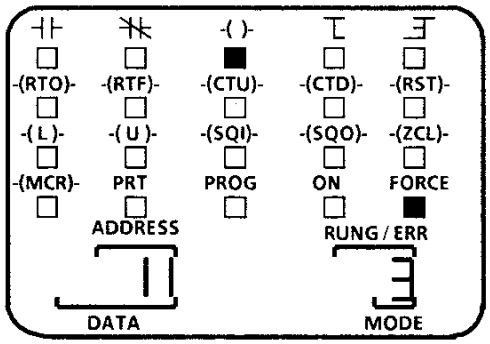
1. Energize the controller.
2. Connect the pocket programmer and enter the Clear Memory mode (mode 1). Press ENTER when the display prompts you with SurE ?
3. The controller will enter the Program mode (mode 2) automatically.

   Enter the test rung shown below. The Examine ON instruction is assigned an internal address, 701, to avoid involving external inputs. The Output instruction should be assigned the address corresponding to the first output device you will test.



Output test rung

4. Place the controller in the Run mode (mode 3). Display the test rung on the pocket programmer.
5. Move the cursor to the output instruction. Force this instruction address to the ON state, using the Force ON function. The FORCE LED on the programmer should be lit. Also, the FORCED I/O indicator on the processor unit should light. If for example your first output device to be tested were at address 011 in rung 3, the pocket programmer display should look like this:



   Now observe the output terminal status indicator LED and the output device. The LED should be ON and the output device should be energized (unless you had to purposely disconnect it).

6. Remove the Force ON condition, using the Force OFF function. The test rung should go FALSE, the output status indicator should go OFF, and the output device should be de-energized.

Testing Outputs (D) *(continued)*

If the LED status indicators, and output device functioned properly, testing of the first output is complete. Repeat this procedure for each output. To save time, begin at step 3, by entering the Program mode, deleting the test rung, and re-entering it with the appropriate output address.

We recommend that you do not program multiple rungs. This is a potential source of confusion and errors.

Troubleshooting: If the output terminal LED status indicator does not go ON and the output device does not energize, try another output terminal and check the expansion unit connection. If the problem still exists, the processor unit or pocket programmer may need replacement. Contact your local Allen-Bradley representative.

If the output terminal status indicator goes ON but the output device fails to operate, check:

a) The output terminal power source.

b) The output circuit wiring.

c) The output device itself.

If the source of the problem is not found to be in a), b), or c), the processor unit or pocket programmer may need replacement. Contact your local Allen-Bradley representative.

Entering and Testing Your Program (E)

After all inputs and outputs have been tested and are functioning properly, you can enter your program in the processor RAM memory. Before entering your program, enter mode 1 and follow the prompting message to clear the processor RAM memory. The processor will enter the Program mode automatically when memory is cleared. Enter your program using the keystroke sequences you have become familiar with in previous chapters. Careful program entering will reduce the chance of program related problems.

Program Verification – After the program has been entered, it must be verified. Each instruction and rung must be closely checked for errors before the controller is placed into operation. Check your written program, rung for rung, against the program you have just entered into the processor RAM. Use the NEXT and LAST keys to step through your program and check addresses and other information associated with your instructions. The most common mistakes in program entry are:

- Incorrect addressing of instructions.
- Omission of an instruction or rung.
- Mistaken entry of the same output instruction and address twice in a program. This will cause the first output instruction to be ignored.

Program Checkout Using the Test-Single Scan Mode – After the user program has been entered and verified, the Test-Single Scan mode (mode 4) can be used to verify correct operation of the user program. This mode allows you to step through the sequence of events that are to be executed by your program.

Entering and Testing Your Program (E) *(continued)*

When the controller is placed in the Test-Single Scan mode, the processor unit will complete one scan of the user program each time the ENTER key is pressed. Controller outputs are disabled in this mode, however execution of the user program can be monitored with the programmer. Timer and time driven sequencer accumulated values will be incremented by 0.1 for each scan of the user program if rung conditions are correct. To check out the user program using the Test-Single Scan mode follow the steps below.

1. Place the controller in the Test-Single Scan mode using the pocket programmer.
2. Use the cursor keys to display the output instruction or ladder rung that you wish to monitor.
3. Simulate the input conditions necessary to execute the first step of the user program. If it is not practical to manually activate the input device, use the FORCE function to simulate the proper condition.
4. As you simulate the input conditions that will make rung 1 TRUE, the status of your input devices changes state but you will not be able to see this change updated on the programmer until the ENTER key is pressed. Press the ENTER key and monitor the programmer display to verify correct operation of the user program.

   Say for example your first rung is an Examine ON instruction and an Output Energize instruction. You could simulate the ON condition necessary to make the Examine ON instruction TRUE. The ON LED for the examine instruction would not be lit until you press the ENTER key. When you turn the input device OFF, the Examine ON instruction will continue to remain TRUE (ON LED lit), until the ENTER key is pressed again.
5. Repeat steps 2 thru 4 for each step of your program to verify that the user program is executed correctly.

Motion Checkout (F)

Now that program execution has been verified, checkout of machine motion can begin. All persons involved with the programming, installation, layout design, machine or process design and maintenance should be involved in making decisions for determining the best and safest way to test the total system.

The following procedures are general in nature. Individual conditions may warrant their modification. The basic approach is to initiate testing with the least amount of machine motion. Only some outputs are allowed to generate machine motion. Then additional machine motion can be added gradually, thereby allowing any problems to be detected more easily under controlled conditions. The following procedure gives steps for testing machine motion using one output at a time.

Motion Checkout (F) *(continued)*

WARNING: During all phases of motion checkout, station a person ready to operate an Emergency Stop Switch if necessary. The Emergency Stop Switch will de-energize the master control relay and remove power from the machine. This circuit must be hard-wired only; it must not be programmed.

Procedures are as follows:

1. Identify the first output device to be tested and reconnect its wiring.

   WARNING: Contact with AC line potential may cause injury to personnel. When reconnecting wiring, make sure that AC power disconnect switch is opened.

2. Place the controller in the Run mode and observe the behavior of the output device. To do this, simulate the input conditions necessary to energize the output in the program. If it is not practical to manually activate an input device, use the FORCE function to simulate the proper input condition.

Repeat steps 1 and 2, testing each output device, one at a time.

Dry Run (G)

After thoroughly checking out the controller system and program, proceed with a dry run of the application with all output devices enabled. This dry run will vary with the application. A machine tool dry run would test the program with all outputs enabled but without tooling an actual piece part. After the entire system is checked out, and your dry run has been completed satisfactorily, we recommend that you load your program into an EEPROM memory module for back-up program storage. Refer to Page 17-3 for directions on loading the EEPROM from RAM. This step completes start-up procedures. Your SLC Programmable Controller is now ready for operation.

Chapter 22 Maintenance and Troubleshooting

General

The SLC controller has been designed to simplify maintenance and troubleshooting procedures. By observing the diagnostic indicators on the front of the processor unit, the majority of faults can be located and corrected. These indicators, along with error codes displayed on the programmer and PC software screen, help trace the source of the fault to the user's input/output devices, wiring, or the controller.

> **SLC PERSONAL COMPUTER SOFTWARE USERS: This chapter is written with the assumption that you will be using a pocket programmer in certain troubleshooting procedures. You can also use the SLC Personal Computer Software, except in instances where we suggest using the FORCE function. The SLC Personal Computer Software does not allow you to use the FORCE function.**

Safety Considerations

Safety considerations are an important element of proper troubleshooting procedures. Actively thinking about the safety of yourself and others, as well as the condition of your equipment, is of primary importance. Several safety areas are discussed below.

Power Supplies: Before working on a power supply, always remove the AC power source at the main power disconnect switch. When using more than one power supply, be sure to disconnect all of them.

Replacing Fuses: When replacing a fuse, be sure to remove all power from the system.

Main Power Disconnect: The main power disconnect switch should be located where operators and maintenance personnel have quick and easy access to it. Ideally, the disconnect switch is mounted on the outside of the enclosure, so that it can be accessed without opening the enclosure. In addition to disconnecting electrical power, all other sources of power (pneumatic and hydraulic) should be de-energized before working on a machine or process controlled by an SLC controller.

Activating Devices When Troubleshooting: When troubleshooting, never reach into the machine to actuate a device. Unexpected machine motion could occur. Use a wooden stick. A metal rod is more likely to damage the machine and could conduct electricity back to you.

Stand Clear of Machine: When troubleshooting any controller problem, have all personnel remain clear of the machine. The problem could be intermittent, and sudden unexpected machine motion could occur. Have someone ready to operate an emergency stop switch in case it becomes necessary to shut off power to the machine.

Program Alteration: There are several causes of alteration to the user program, including extreme environmental conditions, Electromagnetic Interference (EMI), improper grounding, improper wiring connections, and unauthorized tampering. If you suspect the memory has been altered, check the program against an approved version such as on an EEPROM memory module.

Safety Considerations (continued)

Hardware Redundancy: Circuits installed on the machine for safety reasons, like overtravel limit switches, stop push buttons, and interlocks, should always be hard-wired directly to the master control relay. These devices must be wired in series so that when any one device opens, the master control relay is de-energized thereby removing power to the machine. Never alter these circuits to defeat their function. Serious injury or machine damage could result.

Power Distribution: There are some points about power distribution that you should be aware of. First, the master control relay must be able to inhibit all machine motion by removing power to the machine I/O devices when the relay is de-energized.

Second, if you are using a DC power supply, interrupt the load side rather than the AC line power. This avoids the additional delay of power supply turn-on and turn-off. The DC power supply should be powered directly from the fused secondary of the transformer. Power to the DC input and output circuits is connected through a set of master control relay contacts.

Periodic Tests of Master Control Relay Circuit: Any part can fail, including the switches in a master control relay circuit. The failure of one of these switches would most likely cause an open circuit which would be a safe power-off failure. However, if one of these switches shorts out, it no longer provides any safety protection. These switches should be tested periodically to assure they will stop machine motion when needed.

Preventive Maintenance

The printed circuit boards of the controller must be protected from dirt, oil, moisture and other airborne contaminants. In order to protect these boards, the controller must be installed in an enclosure suitable for the environment. The interior of the enclosure should be kept clean and the enclosure door should be kept closed whenever possible.

Regularly inspect your terminal connections for tightness. Loose connections may cause improper functioning of the controller or damage the components of the system.

> **WARNING:** To ensure personal safety and to guard against damaging equipment, inspect connections with incoming power OFF.

The National Fire Protection Association (NFPA) gives recommendations for electrical equipment maintenance. Refer to article 70B of the NFPA for general requirements regarding safety related work practices.

Spare Parts

Stock a full set of spare parts to minimize down-time. We recommend that at least 10% (minimum of 1) of each SLC controller component be stocked as spare parts. Recommended spare parts are listed below.

	Catalog Number
SLC 100 Processor Unit	1745-LP101, -LP102, -LP103, -LP104
SLC 100 Basic Expansion Unit	1745-E101, -E102, -E103, -E104
SLC 100 Relay Output Expansion Unit	1745-E105
SLC 100 Analog Input Expansion Unit	1745-E106, -E107
SLC 150 Processor Unit	1745-LP151, -LP152, -LP153, -LP154, -LP156, -LP157
SLC 150 Expansion Unit	1745-E151, -E152, -E153, -E154, -E156, -E157
SLC 150 High Speed Input Module	1745-E155
Pocket Programmer	1745-PT1
EEPROM Memory Module	1745-M1
Processor/Programmer Interconnect Cable	1745-C1
Expansion Unit Interconnect Cable – 10 pin to 10 pin	1745-C2
Expansion Unit Interconnect Cable – 20 pin to 10 pin	1745-C3
Plastic Parts Kit – SLC 100	1745-N3
Plastic Parts Kit – SLC 150	1745-N4
Terminal Blocks for Power Supply, Inputs, and Outputs – SLC 150 Processor and Expansion Units	1745-N5
Relay Board – SLC 100 Processor Unit	1745-R1
Relay Board – SLC 100 Basic Expansion Unit	1745-R2
Relay Board – SLC 100 Relay Output Expansion Unit	1745-R3
Battery Assembly – SLC 100 and 150 Processor Units	1745-B1
Power Supply Fuses – 5 per package: SLC 100/150 AC Processor Units and AC Basic Expansion Units SLC 100/150 DC Processor Unit and DC Basic Expansion Units SLC 100 Relay Output Expansion Unit SLC 100 AC Analog Input Expansion Unit SLC 100 DC Analog Input Expansion Unit	 1745-F1 1745-F2 1745-F3 1745-F4 1745-F5

Terminal Block Removal – SLC 150 Processor and Expansion Units

The wiring terminal blocks can be removed to allow replacement of the processor unit without removing power supply, input, or output wiring. To remove a terminal block, back out the two screws located at the ends of the terminal block. Alternate between the two screws, backing out about five turns at a time. This will help avoid binding.

To replace the terminal block, align the terminal block screws with the holes on the chassis. Alternate between the two screws, as you did when removing the terminal block. Press on the center of the terminal block as you tighten the screws to help guard against an improper seat.

Troubleshooting

If installation and start-up procedures detailed in Chapters 19, 20, and 21 were followed closely, your SLC controller will give you reliable service. If a problem should occur, the first step in the troubleshooting procedure is to identify the problem and its source. Do this by observing your machine or process and by monitoring the diagnostic indicators on the controller. By doing this, the source of a problem can generally be narrowed down to the processor/expansion unit, wiring, or the input/output devices.

If a problem should occur within the processor/expansion unit, we recommend that you start troubleshooting the following areas.

Power Supply

The power supply is located in the processor/expansion unit. Its purpose is to convert incoming power to the logic level voltage required by the processor/expansion unit, programmer and EEPROM module. The DC POWER green LED indicator on the front of the processor/expansion unit is ON when the proper logic level voltage is at the output of the power supply. If the DC POWER indicator is not illuminated when incoming power is available, the following should be checked to identify the source of the problem:

1. Incoming power wiring and connections.
2. Power supply fuse. See Pages 22-5 and 22-6 for replacement procedures.
3. Incoming power voltage level. See following paragraph.

Incoming Voltage Monitor: The power supply monitors the voltage at the incoming power terminals:

Specified Voltage	Allowable Range
115VAC	85-132VAC
230VAC	170-265VAC
24VDC	18-30VDC

If the incoming voltage falls outside the allowable voltage range, the PC RUN indicator on the processor unit will extinguish and all outputs will be disabled. Note that the DC POWER indicator might not be extinguished because the voltage could be sufficient for the power supply to provide the necessary logic level voltage.

Processor Restart Switch

The restart switch (MAN/AUTO) on the processor can be used in conjunction with the incoming voltage monitor feature to aid in troubleshooting the controller. When the incoming voltage monitor detects abnormal line voltage, it shuts the processor down. The processor restart switch (MAN/AUTO) setting determines what happens when power is restored to acceptable levels. (The processor must have been in the Run mode when the outage occurred.)

If you place the restart switch in the Manual (MAN) position, the processor will **not** automatically enter the Run mode on power-up. Instead, the processor goes through its normal diagnostic tests, but the outputs remain disabled. To enter the Run mode, you must move the restart switch to the Automatic (AUTO) position or use the pocket programmer. This feature provides you with the additional safety of manually restarting the controller when you want.

In the AUTO position, upon power-up, the processor goes through its normal diagnostic tests then automatically enters the Run mode. (The processor must have been in the Run mode when power was removed.)

Power Supply Fuse Replacement

Under normal power-up conditions, the DC POWER indicator will illuminate. If a power supply fuse is blown, the DC POWER indicator will not illuminate. One of the following conditions could cause a blown power supply fuse:

- Excessive line voltage.
- Internal power supply malfunction.

CAUTION: Use only replacement fuses of the type and rating specified for the unit. Improper fuse selection may result in equipment damage.

WARNING: Contact with AC line potential may cause injury to personnel. Remove system power before attempting fuse replacement.

Power Supply Fuse Replacement (continued)

After the conditions causing the malfunction have been corrected, the fuse can be replaced. It is especially important to check the wiring. Replacement procedure:

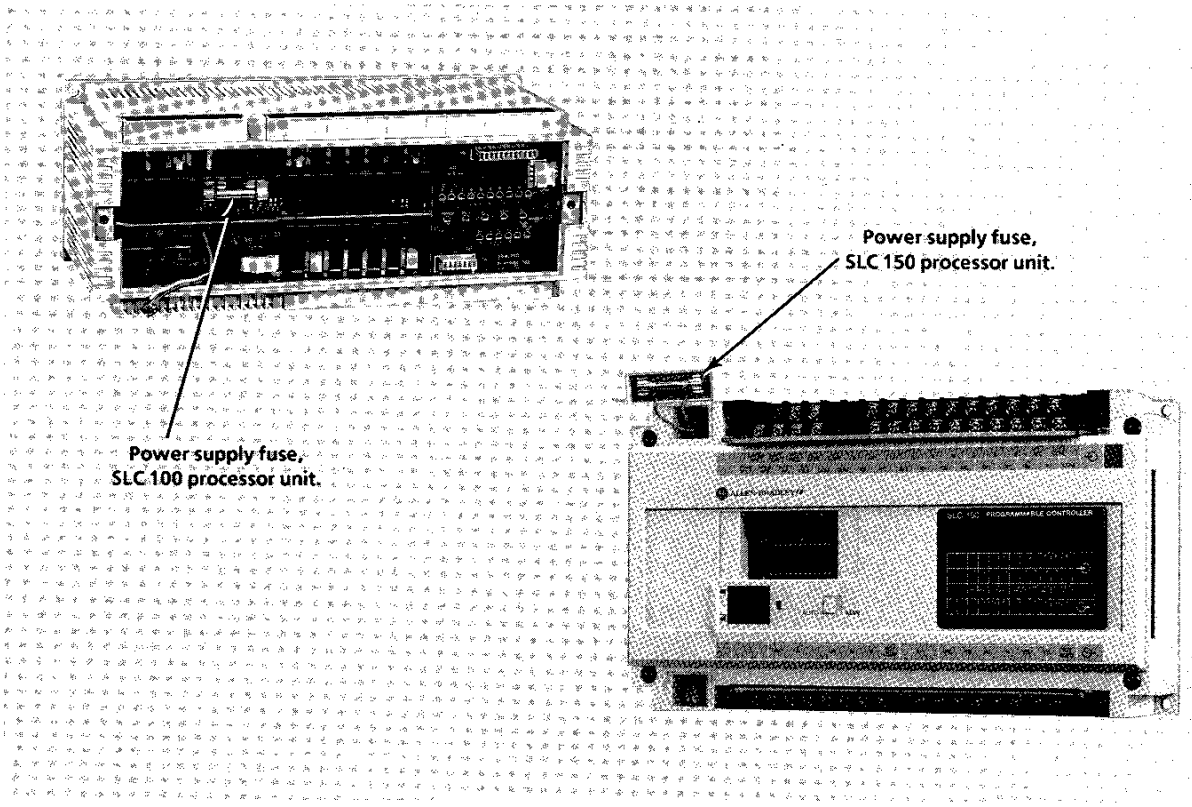
Replacement Procedure – SLC 100 Units: See Figure 22.1.

1. Disconnect line power to processor/expansion unit.
2. Remove the EEPROM module if you are replacing the processor unit fuse. Loosen the 2 cover screws. Move the cover aside to expose the fuse. The cover does not have to be removed.
3. Locate the fuse. Use a miniature fuse puller to grip the fuse and remove it from its holder. Discard this fuse and replace it with the recommended replacement fuse.
4. Replace the cover and tighten the screws. Do not over tighten.
6. Restore power. The DC POWER indicator should now be lit.

Replacement Procedure – SLC 150 Units: See Figure 22.1.

1. Disconnect line power to processor/expansion unit.
2. Remove the fuse compartment cover.
3. Remove fuse holder. Push the handle to the left, then pull outward.
4. Remove the fuse from its holder and replace it with a recommended replacement fuse.
5. Place the fuse holder back into its compartment. Push inward until it locks into place. You may first have to shift the position of the wires.
6. Replace the fuse compartment cover.
7. Restore power. The DC Power indicator should now illuminate.

Figure 22.1



Processor Fault Detection

The processor continually monitors itself for any problems that might cause the controller to improperly execute the user program. A built-in "watch-dog" timer circuit will time out if a processor hardware malfunction occurs. Errors in memory data are also detected through built-in diagnostic routines.

If a processor hardware malfunction or a memory data error is detected, the processor CPU FAULT indicator will light and outputs are disabled. To clear a processor hardware fault, follow this procedure:

1. Toggle the processor restart switch (MAN/AUTO). If this does not clear the fault, then:
2. Turn processor power OFF, then ON. If the fault still does not clear:
3. Connect the pocket programmer to display the error code. Processor error codes, fault descriptions and remedies are listed in Figure 22.2a and 22.2b.

Figure 22.2a

01	Processor hardware problem: malfunction has caused "watch-dog" timer to time-out.	Contact your A-B representative for repair or replacement.
02	Processor hardware problem: unable to write in RAM.	Contact your A-B representative for repair or replacement.
03	Processor hardware problem: internal communication error.	Contact your A-B representative for repair or replacement.
04	Processor hardware problem: unable to write in RAM.	Contact your A-B representative for repair or replacement.
05	Processor memory problem: incorrect checksum in user program.	Correct user program or delete program and re-enter.
06	Processor memory problem: parity error in user program op-code.	Correct user program or delete program and re-enter.
07	Processor memory problem: preset or accumulated values out of range of user program.	Correct user program or delete program and re-enter.
08	Processor memory problem: incorrect checksum in I/O table.	Correct user program or delete program and re-enter.
09	Processor scan time exceeds 100 ms.	Reduce size of user program.

Processor Fault Detection (continued)

Error Codes Detected by the SLC 150 Processor only: Error codes 27-1, 27-2, and 27-5 are detected by the SLC 150 processor only. If any of these error conditions occur, the pocket programmer will initially display error code 27. You must then press the CANCEL CMD key to display the -1, -2, or -5 suffix.

Figure 22.2b

27-1	High speed input module problem	• Check the high speed input module connection. • Determine whether you are using an insufficient number of high speed input modules to support your program. • Determine whether two high speed sequencer instructions have been assigned the same address.
27-2	Zero-cross turn-on failure	• Check to make sure that an AC powered unit is used when address 866 is used. Check the line frequency. Refer to note ① below. • If you are using an AC powered unit and the line frequency is within range listed in note ①, the problem is a hardware failure. Contact your local A-B representative for repair or replacement.
27-5	ZCL programming error	• Determine whether you have programmed more than 8 nested ZCL zones. • Make certain that each ZCL start rung has a ZCL end rung.

① **Error Code 27-2:** The processor measures the input power frequency each time you power-up. If the frequency falls outside the ranges of 57-63 Hz or 47-53 Hz, a processor fault may occur. The pocket programmer will display error 27. When this happens, press the cancel key; if the number 2 appears, the error is a zero-cross fault. (If you are using the SLC Personal Computer Software, a message indicating a Zero-Cross fault appears on the screen.)

Correct the fault as follows:

1. Remove input power to the processor.
2. Correct the input power problem so that the input frequency remains within a valid range.
3. Apply input power to the processor.

Note: When you perform the corrective action above, retentive outputs (timers, counters etc.) will remain in their last state when the Run mode is re-entered. If you want the retentive outputs to be reset to their Off states on power-up, you must clear the error message *before* performing the corrective steps above. Do this by pressing the cancel key until the message "Run 3" is displayed. (If you are using the SLC Personal Computer Software, press any key until the main menu is displayed.) Then correct the fault as indicated above.

Processor Fault Detection (continued)

For your convenience, a processor diagnostic guide is shown in Figure 22.3.

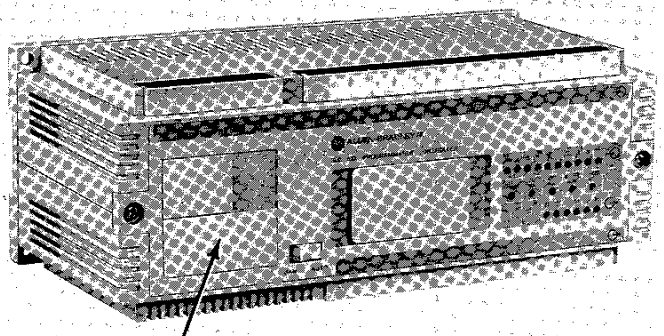
Figure 22.3

ON	ON	OFF	OFF	None; normal status indication in Run mode.
ON	OFF	OFF	OFF	None; normal status indication when not in Run mode.
ON	OFF (If controller is in Run mode.)	OFF	OFF	Processor restart switch in MAN position.
OFF	OFF	OFF	OFF	1. Incoming power absent. 2. Incoming power wiring /connection problem. 3. Blown power supply fuse. (To replace fuse, see Pages 22-5, 22-6.) 4. Power supply malfunction.
ON	OFF (If controller is in Run mode.)	ON	OFF	1. Processor hardware fault (error codes 1 – 4). See Figure 22.2a. 2. Processor memory fault (error codes 5 – 9). See Figure 22.2a. 3. Expansion unit problem (error code 26). See Page 22-16. 4. Incoming power below minimum voltage level (see Chapter 18, Specifications).
ON	ON (If controller is in Run mode.)	OFF	ON	Battery needs to be replaced.

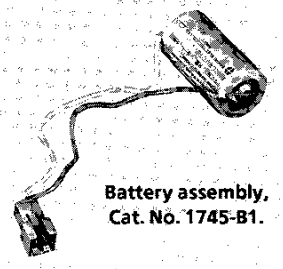
RAM Memory Battery Backup

The SLC 100 processor unit is supplied with battery backup as standard. The SLC 150 processor unit is supplied with capacitor back-up as standard, but can be equipped with battery backup as an option. Figure 22.4 shows the location of the battery compartment in the SLC 100 and SLC 150 processors.

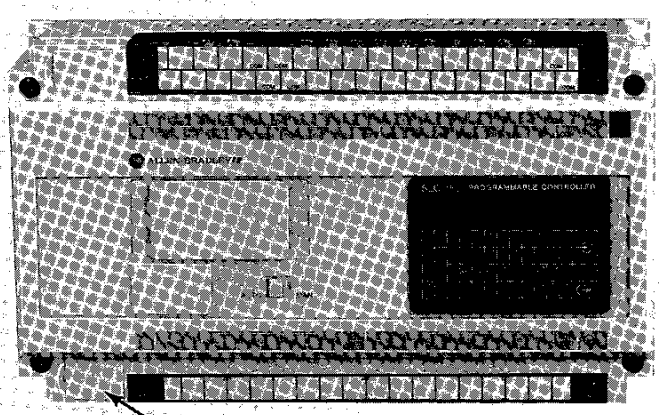
Figure 22.4



Battery compartment, SLC 100 processor unit.



Battery assembly, Cat. No. 1745-B1.



Battery compartment, SLC 150 processor unit.

Battery Handling Precautions

Lithium batteries should be handled and disposed of carefully. Follow the procedures below to ensure proper battery and controller operation, and to reduce hazards to personnel:

- **Only use the battery for its intended application.**
- **Do not attempt to charge the battery. An explosion could result, or the cell could overheat and cause burns to personnel.**
- **Do not open, puncture, crush or otherwise mutilate the battery. The possibility of an explosion exists and/or toxic, corrosive and flammable chemicals would be exposed.**
- **Do not incinerate or expose the battery to high temperatures. Do not solder the battery or its leads. An explosion could occur.**
- **Do not short together the positive and negative leads since the battery could overheat.**
- **Dispose of used batteries immediately. Do not store used batteries longer than 3 months.**

RAM Memory Battery Backup (continued)

Battery Replacement – SLC 100: A lithium battery is housed in a small compartment on the front of the processor unit. This battery provides backup power to the processor RAM when external power is removed. The battery provides backup power for a typical life span of 2-3 years. Actual battery life span may vary depending on controller environmental conditions.

The processor BATTERY LOW indicator will be lit when the battery voltage falls below a threshold voltage. Replace the battery at this time. You can replace the battery without disconnecting power or disturbing normal operation of the machinery. Procedure:

1. Remove small compartment door on front of processor.
2. Remove battery from compartment. Unplug lead wires from processor.
3. Connect new battery making sure that the slot on the battery lead is correctly aligned with the key of the processor unit's socket. Insert battery into compartment.
4. Replace compartment door.

Battery Installation or Replacement – SLC 150: You can install or replace the battery, Catalog No. 1745-B1, without disconnecting power or disturbing normal operation of your machinery. Procedure:

1. Remove battery compartment door from the front of the processor unit.
2. If you are **installing** a battery in a new processor unit (battery never installed before), unplug the battery connector and lead wires (red and white) by inserting your finger into the compartment and pulling up on the lead wires. If this cannot be done with a finger, you may wish to use a needle nose pliers to pull out the lead wires.

2a. If you are **replacing** an old battery, remove the battery assembly stored in the compartment. To do this, first pull up on the lead wires and remove them from the compartment. The battery is held in place by a small retainer clip on the right of the battery. Apply pressure on the left of the battery with a small screwdriver or other small tool to push the battery against the clip while pulling the lead wires up at the same time. This will allow the battery to be pulled straight out of the compartment. Unplug female from male end of connector to disconnect old battery.

3. Connect a new battery making sure that the slot on the battery assembly connector aligns with the key of the processor unit socket.
4. Insert battery into compartment minus side first (white lead wire). Push the retainer clip to the side with the battery while inserting so that the battery will fall to the bottom of the compartment and stay in place with the clip.
5. Place battery connector and lead wires in compartment.
6. Replace compartment door.

I/O Hardware

When troubleshooting, pay careful attention to these general warnings:

> **WARNING:** Have all personnel remain clear of the controller equipment when power is applied. The problem may be intermittent and sudden unexpected machine motion could occur and result in injury. Have someone ready to operate an emergency stop switch in case it becomes necessary to shut off power to the controller equipment. Also see NFPA 70E Part II for additional guidelines for safety related work practices.
>
> **WARNING:** Never reach into a machine to actuate a switch since unexpected machine motion can occur and cause injury. Use a wooden stick. A metal rod could damage the machine and/or conduct current to the person holding it.
>
> **WARNING:** Remove all electrical power at the main power disconnect switches before checking electrical connections or inputs / outputs causing machine motion.

If the controller is operating in the Run mode but output devices do not operate as programmed, most likely the problem is one of the following:

- I/O devices.
- Wiring between I/O devices and user power.
- User power.

Identifying one of the above as a problem can usually be narrowed down by comparing the actual status of the suspect I/O with controller status indicators. This comparison technique is similar to procedures given in Start-Up, Chapter 21.

Each I/O device has two status indicators. One of these indicators is located on the front of the processor/expansion unit and the other is the "ON" LED on the programmer. These two status LEDs indicate the logical (TRUE/FALSE) nature of programmed instructions corresponding to connected I/O devices.

Testing Outputs: If an output device does not energize as expected, observe the appropriate status indicator on the processor/expansion unit to see if it is lit.

If the status indicator is lit but the output device does not energize, check wiring connections between the wiring terminals and output device. If wiring connections appear correct, use a test light or voltmeter to check for power at the output wiring terminals. If there is power at the wiring terminals, check your output device.

If there is no power at the wiring terminals, the processor/expansion unit output relay may be faulty. Reconnect the output device to a spare output terminal. If none is available, replace the output relay board.

If the status indicator is not lit, the logical status of the corresponding programmed instruction is FALSE. Connect the pocket programmer and monitor each instruction in the programmed rung and observe the "ON" LED status indicator on the programmer. A continuous path of TRUE condition instructions must exist in order for the status indicator to be lit and the output device energized. If a continuous path of TRUE condition instructions does not exist, your problem might involve input devices.

I/O Hardware (continued)

If you want to make sure that your output can be energized before you start checking your input devices, you can use the Force function to energize the output. Refer to Page 16-1 for information on using the Force function. If the Force function energizes your output, your processor/expansion unit is functioning properly. Move on and test the input device that energizes the problem output.

A guide for troubleshooting output devices is shown in Figure 22.5.

Figure 22.5

Energized (ON)	ON	TRUE ▩	None. Correct status indication.
De-energized (OFF)	OFF	FALSE □	None. Correct status indication.
De-energized (OFF)	ON	TRUE ▩	1. Wiring to output device at fault. 2. Output device fault. 3. Processor/expansion unit output relay fault.
De-energized (OFF)	OFF	Rung conditions are TRUE but output instruction is FALSE. □	Programming error. Two outputs at the same address.
Energized (ON)	ON	Rung conditions are FALSE but output instruction is TRUE. ▩	Programming error. Two outputs at the same address.

Testing Inputs: If input hardware is suspected as being the source of a problem, follow the procedures given below to isolate the problem:

1. Using the programmer, place the controller in the TEST – Continuous Scan mode (mode 5).
2. The input status indicator located on the processor/expansion unit, illuminates when the input circuit is receiving power; that is, when the input device is ON. You should compare the actual ON/OFF status of the suspect input device with its corresponding input status indicator. To do this, manually turn the input device ON and OFF and observe the status indicator.

Procedure continues on next page.

I/O Hardware (continued)

Step 2, Testing Inputs, (continued):

> **WARNING:** Do not reach into a machine to actuate a switch by hand. Unexpected machine motion could cause injury. Stand clear of the machine and use a wooden stick to operate the switch.

If the input status indicator **does not illuminate** when the input device is ON, check the following:

- User power to the input device.
- Wiring between input device and user power.
- The input device.

> **CAUTION:** Remove system power before checking electrical connections or input devices.

If the input status indicator **illuminates**, go to step 3.

3. Using the programmer, place the controller in the Run or Test mode (mode 3, 4 or 5). Compare the programmer's ON status indicator with the actual ON/OFF status of the suspect input device.

   If the programmer does not show the correct status indication for the suspect condition instruction, the input circuitry may not be supplying the required logic level voltage. In this case, reconnect the input device to a spare input circuit. If this does not solve the problem, contact your A-B representative for repair or replacement of the processor or expansion unit.

A guide for troubleshooting input devices is shown in Figure 22.6.

Figure 22.6

Closed (ON)	ON	TRUE ▩	FALSE □	None. Correct status indication.
Open (OFF)	OFF	FALSE □	TRUE ▩	None. Correct status indication.
Closed (ON)	ON	FALSE □	TRUE ▩	1. Processor/expansion unit input circuitry fault. 2. Processor/programmer communication fault.
Closed (ON)	OFF	FALSE □	TRUE ▩	1. Wiring/power to processor/expansion unit input circuitry fault. 2. Processor/expansion unit input circuitry fault.
Open (OFF)	OFF	TRUE ▩	FALSE □	1. Programming error. 2. Processor/programmer communication fault.
Open (OFF)	ON	TRUE ▩	FALSE □	Short circuit in input device or wiring.

Pocket Programmer

The programmer can be used to program, edit, monitor and troubleshoot the controller.

If no display appears when the programmer is initially energized, check the following:

- DC POWER indicator on the processor unit: If the LED is OFF, power is not available for the programmer.
- Interconnect cable: Check connections at programmer and processor unit for proper fit.

If everything appears to be correct, replace the interconnect cable. If this does not solve the problem, contact your A-B representative for repair or replacement of the pocket programmer.

Error Code 55: If a programmer hardware malfunction is detected, error code 55 will be displayed by the programmer. If this happens, contact your A-B representative for repair or replacement of the programmer.

Diagnostic Test – Programmer, Mode 9

When mode 9 is selected, the programmer automatically runs through a series of diagnostic tests including:

1. Firmware revision number (displayed once at start of test).
2. LED indicators: Each LED is individually illuminated.
3. Digital display: Each segment of the digital display is individually illuminated.
4. LED indicators and digital display combined: All indicators and LED display segments illuminate.

In addition, the keyboard can be tested to verify that it is working correctly. When a key is pressed, a unique numeric value is displayed on the programmer.

You exit the diagnostic mode by pressing CANCEL CMD or by toggling the ON/OFF switch on the programmer.

Communication Problems (Error Codes 10 thru 25)

Error codes 10 through 25 will be displayed if communication problems are detected between the programmer and the processor unit. Follow the procedures given below to locate the problem:

1. Check the interconnect cable to insure it is securely connected at both the processor and programmer.
2. Recycle power to the programmer.
3. Recycle power to the processor unit.

If the error code still appears, contact your A-B representative for repair or replacement of the processor/expansion unit and/or pocket programmer.

Expansion Unit Problem (Error Code 26)

In the Run and Test modes, the processor unit will continually monitor the expansion units that are connected. If a problem is detected with any of the expansion units, the processor unit will shut down, all outputs will be turned OFF, and the CPU FAULT LED indicator will illuminate. Error code 26 will be displayed by the pocket programmer. If the above conditions exist, your controller should be checked for the following fault conditions.

1. Check program addressing and the number of expansion units connected. If you have addressed an expansion unit that is not connected, this error condition will occur.
2. Check the expansion unit interconnect cable for damage. Make sure that all expansion units are properly connected.
3. Check expansion unit power connections and hardware. Make sure that the DC power LED is lit on each expansion unit connected.

If items 1, 2, and 3 above appear to be O.K., and the CPU Fault still exists, the expansion unit may need to be replaced. Contact your local Allen-Bradley representative for repair or replacement.

EEPROM Module Problem (Error Codes 51 and 52)

If error code 51 is displayed by the programmer when programming the EEPROM module, a malfunction has occurred in either the processor unit or the EEPROM module. If this happens, try a different EEPROM module. If the problem persists after replacing the EEPROM module, contact your A-B representative for repair or replacement of the processor unit.

Error code 52 appears when the program in the processor unit's RAM does not match the program stored in the EEPROM module. The error code is a cautionary reminder that changes have been made to the program and the RAM and EEPROM no longer match. You can correct this error in two ways:

1. Select mode 6, Store Program in EEPROM, and transfer the new program into the EEPROM. The previous contents on the EEPROM module will be deleted.
2. Press ENTER a second time to override the error code.

Note: Error code 52 will not appear if on-line data changes are made while the EEPROM module is installed.

Subject Page

Subject Page

Subject *Page*

SEQUENCER INSTRUCTION DATA FORM

SEQUENCER CLASSIFICATION: ☐ –(SQI)– ☐ –(SQO)– ADDRESS: _____ ☐ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: _____

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES			
	B				A				Data B	Data A				
Bit Addresses →														
Mask Data →														
Step Data → 0														
1														
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														

ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

Industrial Control Group
Milwaukee, Wisconsin 53204

Publication 1745-800 – November, 1987
Supersedes Publication 1745-800 Dated September, 1985

40065-236-01(C)

ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

Self-Teach Manual

Bulletin 1745 SLC™ Programmable Controllers

Publication 1745-800A – March, 1988
Supersedes Publication 1745-800A Dated November, 1987

40065-237-01(D)

Table of Contents

Unit 1 How to Use This Manual

A Companion to the User's Manual

The Self-Teach Manual is designed for use with the SLC Programmable Controller User's Manual, Publication 1745-800.

This manual consists of Question/Exercise Units corresponding to Chapters 2 thru 17 of the User's Manual. Question/Exercise Units include questions, a summary of key terms and concepts, and in some cases additional programming details, examples, and exercises.

A 1-2-3 Procedure

We suggest you use the following procedure. You will need a simple controller set-up or a factory demonstrator to practice programming.

1. Read Chapter 2 of the User's Manual *twice*, then refer to Question-Exercise Unit 2 of this manual.
2. Read the "Key Terms and Concepts" in the Question-Exercise unit. Make sure you understand them. Review the referenced User's Manual pages if necessary.
3. Answer the questions, then check your answers with those in Unit 18. If you have answered any questions incorrectly, go back and review the text material associated with these questions.

Repeat steps 1, 2, and 3 for succeeding chapters (read the chapters in numerical sequence).

Set your own pace.

Q-E Unit 2 Equipment Overview

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

SLC 150 Processor Unit 2-2
Expansion Unit 2-4, 2-8, 2-9, 2-10
Pocket Programmer 2-11
EEPROM Memory Module 2-11
High Speed Input Module 2-5
PC Software Interface Converter 2-12
SLC 100 Processor Unit 2-6
Timer Counter Access Terminal 2-12

Question Group 2A

Answers in Unit 18

1. The pocket programmer must remain connected to an SLC processor unit in order to run your program. True or false?
2. The EEPROM memory module must be inserted in order to operate an SLC controller. True or false?
3. An SLC processor unit user program is retained when processor power is removed. True or false?
4. Which of the following statements are true?
   a. The pocket programmer plugs directly into the processor unit.
   b. The pocket programmer can communicate directly with the processor unit with no cable connections at distances up to 6 feet.
   c. The pocket programmer must be wired to the processor unit by means of a 3-wire power cord.
   d. The 6 ft. interconnect cable that is connected to the pocket programmer plugs in to the processor unit when needed.
5. The EEPROM memory module is used to:
   a. Store the contents of the processor RAM.
   b. Load the contents of EEPROM memory to the processor RAM.
   c. Provide back-up storage for your program.
   d. All of the above.
   e. None of the above.
6. The SLC 150 uses its back panel as a heat sink. True or false?
7. The incoming power terminals on SLC processor units have self-lifting pressure plates for easy wire insertion and secure connections. True or false?
8. There are 10 input circuits on the SLC 100, and 20 inputs on the SLC 150 processor unit. True or false?
9. There are 6 output circuits on the SLC 150 processor unit. True or false?
10. SLC 100 and SLC 150 processor and expansion units can be mixed and matched to suit your application. True or false?
11. Each I/O circuit on the processor and expansion unit is associated with an external I/O address. True or false?

Question Group 2A
(continued)

Answers in Unit 18

12. A snap-on cover protects the EEPROM port when the EEPROM module is not in use. True or false?
13. There are 21 LED diagnostic indicators on the processor unit that aid in monitoring and troubleshooting the SLC 100 Programmable Controller. True or false?
14. I/O terminal circuitry interfaces the logic level voltage of the processor with the voltage levels of external input and output devices. True or false?
15. Input terminal circuitry senses whether external input devices are ON or OFF and relays this information to the processor unit. True or false?
16. Output terminal circuitry controls the ON/OFF status of external output devices, based on commands received from the processor unit. True or false?
17. SLC 100 processor units have 10 LED status indicators that correspond to the 10 input circuits. An LED is lit when the corresponding input circuit is energized. True or false?
18. SLC 150 processor units have 6 LED status indicators that correspond to the 6 output circuits. An LED is lit when the corresponding output instruction is TRUE. True or false?
19. SLC 100 basic expansion units have 10 LED status indicators that correspond to the 10 input circuits and cannot be used with the SLC 150 controllers. True or false?
20. SLC 100 basic expansion units have 6 LED status indicators that correspond to the 6 output circuits. An LED is lit when the corresponding output instruction is TRUE. True or false?
21. SLC I/O wiring terminals have self-lifting pressure plates for easy wire insertion and secure connections. True or false?
22. SLC 100 input terminal covers are color-coded for easy identification. True or false?
23. The pocket programmer is your means of communicating with the processor unit. True or false?
24. The EEPROM memory module is an optional device that can be used to store the contents of the processor RAM. True or false?
25. You can load the contents of the EEPROM memory module to the processor RAM memory. True or false?
26. The EEPROM module has a write-on area for use in identifying its contents. True or false?
27. The EEPROM memory module requires a special mounting plate that will hold up to four memory modules and a reserve battery. True or false?
28. The relay output expansion unit has no inputs. True or false?
29. There are 12 hard contact output circuits in the relay output expansion unit. True or false?

Question Group 2A
(continued)

Answers in Unit 18

30. SLC personal computer software can also be used for programming SLC processor units. True or false?
31. Expansion units are available with AC, DC, triac and transistor outputs. True or false?
32. High speed input modules can be used only with the SLC 150 processor. True or false?
33. Relay expansion units have wiring terminals for 12 output circuits and can be used with either the SLC 100 or SLC 150. True or false?
34. The TCAT is used to access programmed timer data only. True or false?
35. The analog input expansion unit allows connection of up to 8 analog input signals. True or false?

Q-E Unit 3 Programming Basics

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

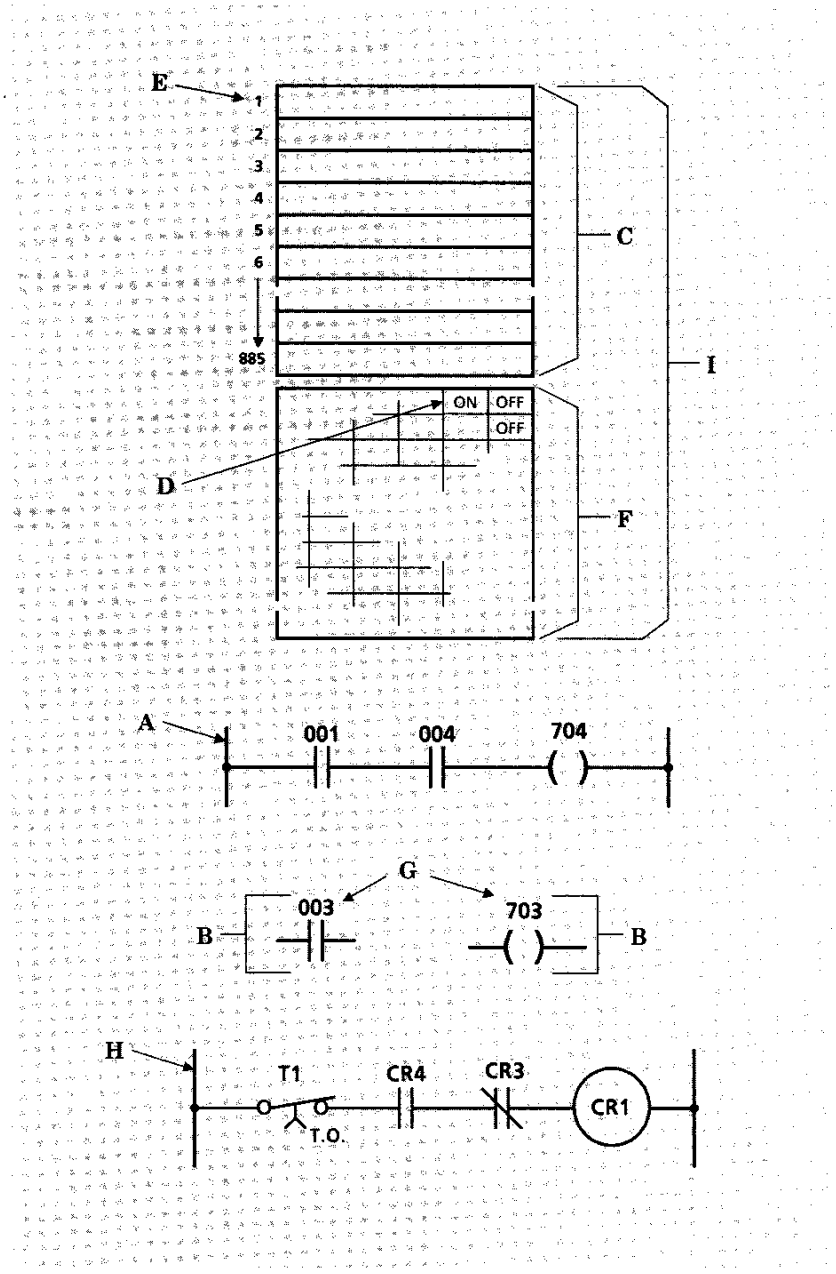
SYMBOLOGY

Matching Exercise

Answers in Unit 18

Each of the following terms can replace one of the callout letters in the figures below. Fill in the correct letter for each term.

1. Processor Memory _____
2. Address _____
3. User Program _____
4. Instruction _____
5. Data Table _____
6. Relay Ladder Rung _____
7. Status Bit _____
8. Word _____
9. PC Ladder Rung _____



Question Group 3A

Answers in Unit 18

1. Instructions are either:
   a) TRUE or FALSE. b) ON or OFF.
2. Status bits are either:
   a) TRUE or FALSE. b) ON or OFF.
3. Status bits tell us exactly where an instruction is located in the user program. True or false?
4. All SLC 150 programs are compatible with the SLC 100. True or false?
5. The address number of an instruction tells us:

   a) Whether the instruction is internal or external.

   b) Whether the instruction represents an input or an output device.

   c) a and b
6. An address number of 016 indicates that the address is associated with an internal instruction. True or false?
7. How many external addresses are available for programming?

   a) 32 b) 112 c) 885
8. The memory storage unit for instructions is called:

   a) the BYTE. b) the word. c) the bit.
9. The data table portion of memory stores the list of instructions you enter. True or false?
10. The instruction address links the instruction to a particular status bit in the data table portion of the memory. True or false?
11. Relay-type instructions are used internally and externally. True or false?
12. Address 868 has a function called _____ _____ instruction.
13. When developing an SLC program, you may use as many address blocks as you like, so long as you do not exceed 112 addresses. True or false?
14. When using the HSI, the operating cycle is interrupted to allow processor to count high speed input signals. True or false?
15. Triac zero cross turn on can be programmed in the SLC 150 by using address 864 in the user program. True or False?

Question Group 3B

Answers in Unit 18

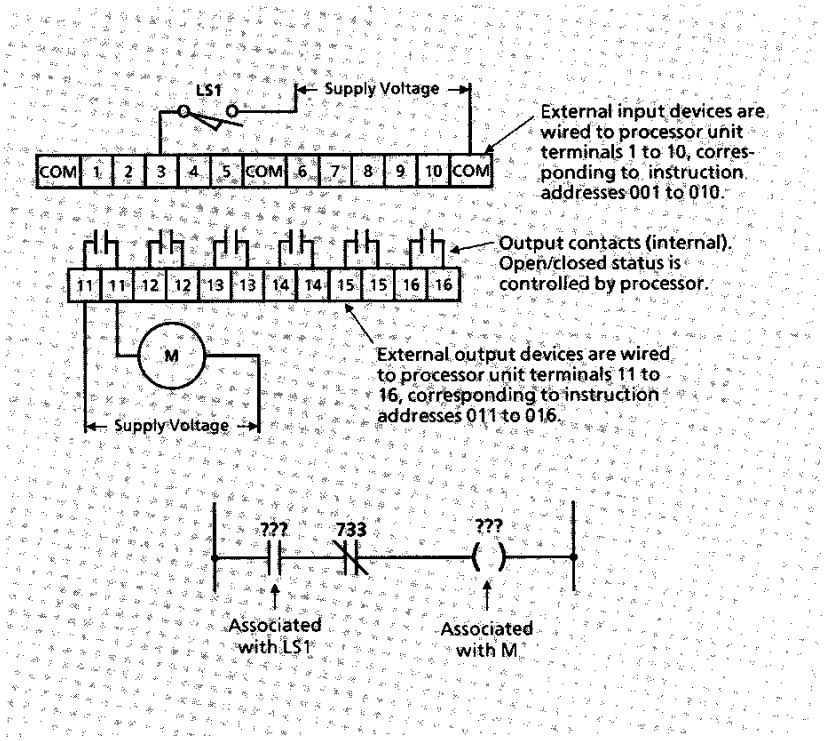
1. Instructions are classified into two groups, input instructions and output instructions. True or false?
2. The branching instructions are classified in the output instruction group. True or false?
3. Examine ON and Examine OFF instructions are classified as _____ instructions. Choose one of the following.

   a. input. b. output. c. condition. d. unconditional.
4. The Examine ON instruction means "examine the status bit for an ON condition". True or false?
5. With the Output Energize instruction, the status bit of the addressed Output Energize instruction is set ON when the conditional instructions in that rung are TRUE. If there are no conditional instructions, the status bit is always ON. True or false?
6. Logical continuity exists in a rung if:
   a. The Examine ON instructions in the rung are TRUE.
   b. The Examine OFF instructions in the rung are FALSE.
   c. A continuous path of TRUE conditional instructions exists.
   d. Only one conditional instruction in the rung is FALSE.
7. If an Examine ON instruction is used to represent a limit switch contact, a closed condition for the limit switch makes this instruction logically TRUE. True or false?
8. The controller operating cycle is divided into two parts, the I/O scan and the program scan. True or false?
9. If an external input device changes state during the program scan, which of the following is true?
   a. The change of state will be recognized immediately during that program scan.
   b. The change of state will be recognized during the next I/O scan.
   c. The user program scan will be reset to 0.
   d. The program scan will start over again.
10. For a typical 500 word program, the SLC 100 repeats its processor operating cycle 67 times each second. How many times does the SLC 150 controller repeat its cycle in one second:

    a. 200 b. 300 c. 400 d. 500 e. None of these.
11. If you are using a high speed input module, the SLC 150 operating cycle is interrupted to allow the processor unit to count high speed input signals and immediately set processor unit outputs. True or false?
12. The SLC 150 I/O scan is 300 microseconds plus 250 for each expansion unit address block used. How long is the I/O scan for the SLC 100?
    a. 1.5 msec plus .27 ms for each expansion unit.
    b. 1.5 sec plus .27 sec for each expansion unit.
    c. 6090 microseconds plus 500 microseconds for each expansion unit.
    d. none of these.
13. The typical SLC 100 operating cycle is 15 msec, whereas the typical SLC 150 operating cycle is only 2.5 msec. True or false?

Question Group 3C
Answers in Unit 18

In the figure of the SLC 100 processor unit I/O terminals below, a limit switch is wired to an input terminal and a starter coil is wired to an output terminal. Programmed instructions associated with these devices are shown in the ladder rung. Answer the following questions.



1. What is the 3-digit address number of the instruction associated with N.O. limit switch LS1?
2. Which of the two classifications does this instruction belong to?
3. What is the name of the instruction?
4. Assume that the controller is energized and operating (Run mode), and the limit switch is in the closed position. Will this instruction be TRUE or FALSE?
5. Will the corresponding status bit be ON or OFF?
6. What is the 3-digit address number of the instruction associated with coil M?
7. Which of the two classifications does this instruction belong to?
8. What is the name of the instruction?
9. What rung conditions are necessary to make this instruction TRUE?
10. When this instruction is TRUE, will the associated status bit be ON or OFF?
11. The instruction at address 733 is associated with an external I/O device True or false?
12. What is the name of the instruction?

Question Group 3C
(continued)

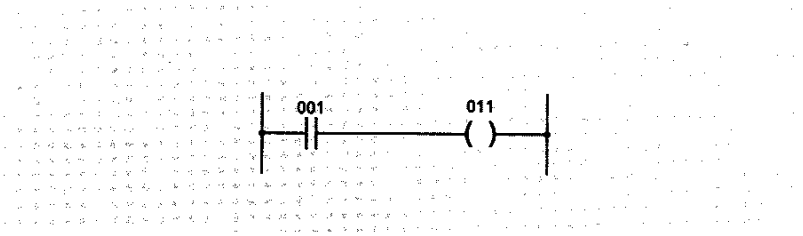
Answers in Unit 18

13. When this instruction is TRUE, will the corresponding status bit be ON or OFF?
14. When the rung is TRUE, the status bits associated with each of the instructions will be ON. True or false?
15. If a N.C. limit switch were substituted for LS1, the instruction representing it would have to be changed to an Examine OFF instruction in order to energize coil M for the closed position of the switch. True or false?
16. Suppose you had one expansion unit connected to your processor unit. If the limit switch and coil were connected to corresponding terminals on the expansion unit, what addresses would you assign to these instructions?

Question Group 3D

Answers in Unit 18

In the figure below, a simple programmed rung is shown. Answer the following questions.



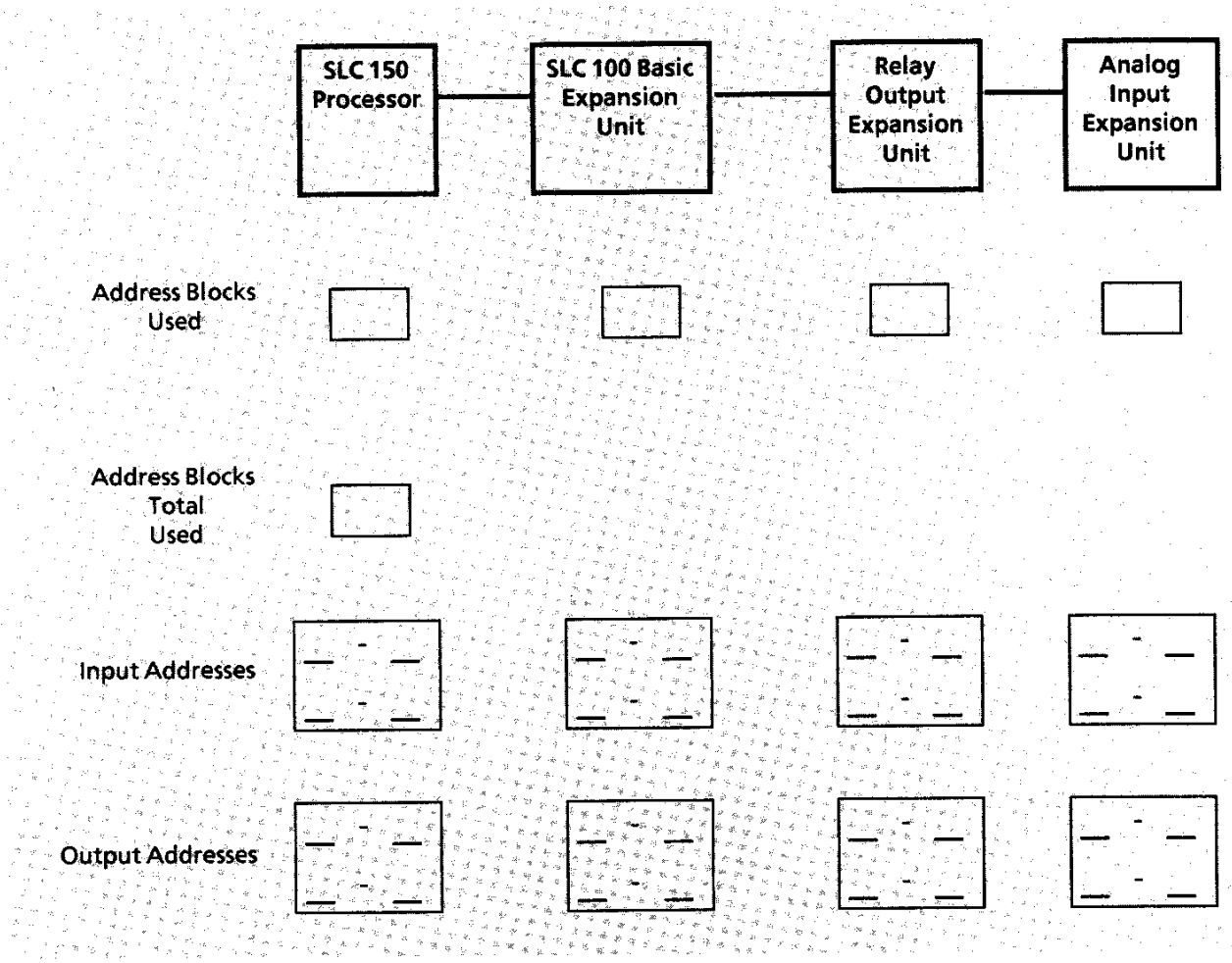
1. In what part of the operating cycle does the status bit of instruction 001 change from OFF to ON or vice versa?
2. In what part of the operating cycle does the status bit of instruction 011 change from OFF to ON or vice versa?
3. The output contact will go ON or OFF during the program scan. True or false?
4. The time period beginning when the input terminal representing instruction 001 goes ON and ending when the output terminal associated with instruction 011 goes ON is
   a) Fixed, at approximately 25 milliseconds.
   b) Variable, but never longer than one operating cycle.
   c) Variable, depending on the location of this rung in the program.
   d) Variable, depending on what point in the operating cycle the processor happens to be at when the input terminal goes ON.
   e) Variable, depending on program length (how many instructions in the program) and content (the various *types* of instructions in the program).

   Which of the above statements are true?

Question Group 3E

Answers in Unit 18

Assign address and determine the number of address blocks required for the SLC system below.

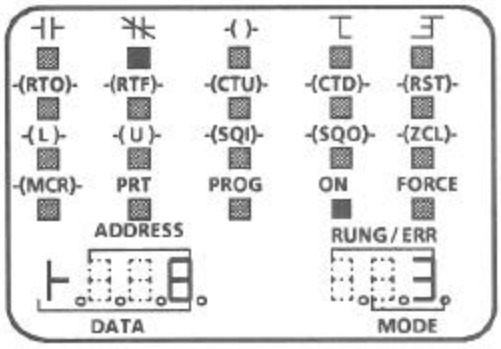


Q-E Unit 4 Using the Pocket Programmer

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

SYMBOLOGY



Question Group 4A

Answers in Unit 18

1. The pocket programmer is only used to enter your program. True or false?
2. The pocket programmer has 30 keys, many with upper case functions. True or false?
3. The pocket programmer has an ON/OFF switch located just below the display. True or false?
4. The pocket programmer display consists of two separate areas. The upper display consists of 20 square LEDs. An LED will be lit when you press the corresponding key on the keyboard. The lower display area consists of seven-segment LEDs which display various data and messages. True or false?
5. To select an upper case function on the keyboard, which key must be pressed first?

   a. RUNG. b. SEARCH. c. SHIFT. d. MODE. e. NEXT.

6. The pocket programmer can be password-protected to prevent unauthorized access to the user program. True or false?
7. What mode of operation must you enter to change or enter a password?

   a. Mode 2. b. Mode 4. c. Mode 6. d. Mode 8. e. Mode 10.

8. Error codes must be converted to binary code before they can be interpreted. True or false?
9. When any error code from 01 thru 08 appears on the display:

   a. An internal processor error has occurred.
   b. The processor CPU FAULT LED will light.
   c. All outputs will be disabled.
   d. All of the above.
   e. None of the above.

10. The processor unit will function in only two modes, Programming and Run. True or false?
11. There is a table fixed to the back of the pocket programmer with information on error codes, modes of operation and instruction addressing. True or false?
12. If you enter mode 9 on the pocket programmer the display will read ERROR 40. True or false?
13. If you enter mode 9, the pocket programmer will perform a diagnostic test on itself. True or false?
14. In order to clear the processor RAM memory you must enter mode 1. True or false?
15. Error codes 60 - 66 are user errors that indicate:

    a. Invalid procedure (s)
    b. Incorrect data
    c. Incomplete data
    d. All of the above

16. Error Code 27 is detected only by an SLC 150 Processor. True or false?

Question Group 4A
(continued)
Answers in Unit 18

17. The keyboard is used to enter your program information, and the display shows you this information and data as you enter it. True or false?

Examine the Programmer

1. Examine the pocket programmer keyboard and press one of the keys. As you press down on the key you can feel the key give under the pressure of your finger. As you let up on the key, it will return to its original position. This tactile feedback feature helps make it easy to program the SLC Programmable Controller. You can feel the key and know that the information is entered.
2. Turn the pocket programmer over so that you can see the table that is fixed to the back side. This table can be very useful whenever you are using the pocket programmer. Mode numbers and descriptions, acceptable addresses for programming various instructions, and error code descriptions are featured on this table. Refer to this table when you need information on any of these items.
3. Above the information table at the top of the programmer is the interconnect cable. The cable should remain connected to the programmer at all times. When you no longer need the programmer to communicate with the processor, disconnect the cable at the processor unit and leave the cable connected to the programmer.

Question Group 4B
Answers in Unit 18

1. SLC Programmable Controllers operate from a 5 VDC power source and should not be connected to 120 VAC power. True or false?
2. The pocket programmer operates from a lithium power battery and can communicate with the processor unit with no cable connection at a distance of up to 6 ft. True or false?
3. When power is applied to the processor unit, the DC POWER LED should light. True or false?
4. When power is first applied to the pocket programmer, it will automatically go thru a series of diagnostic checks. True or false?
5. The interconnect cable is equipped with spring latches that secure the cable to the programmer and processor unit. True or false?

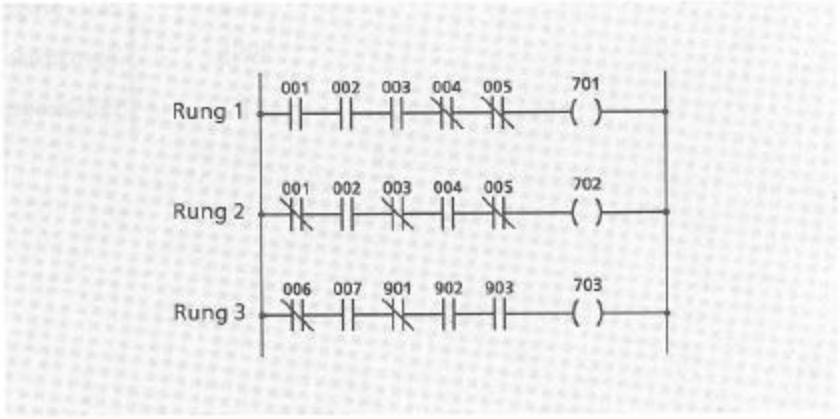
Question Group 4B
(continued)
Answers in Unit 18

6. When connecting a three-wire power cable to an SLC processor, connect the black wire to the terminal marked CHASSIS GND. True or false?
7. The interconnect cable should remain attached to the programmer at all times. When not needed, disconnect the programmer at the processor unit and leave the cable connected to the programmer. True or false?
8. To avoid accidental memory loss, the pocket programmer will ask you to verify that you want to clear memory when you enter mode 1. What prompting message does the display show you when you enter mode 1?
9. When the processor memory has been cleared on the SLC 100, there are 885 words of memory available for programming. True or false?
10. A typical rung has one or more condition instructions and end with a single output instruction. True or false?
11. If you enter the wrong address for an instruction, you can change it by pressing the CANCEL CMD key until the incorrect information is deleted. The correct address can then be entered. True or false?
12. It is possible to have two output instructions in the same rung. True or false?
13. The output instruction completes the ladder rung. After entering the output instruction and its address, pressing the ENTER key will enter the rung in memory. True or false?
14. Ladder rungs are numbered as they are entered in memory. While you are entering the rung, the display tells you what rung number you are entering. True or false?
15. The upper case functions are all color-coded blue for easy identification. True or false?
16. When you press an instruction key, the corresponding instruction LED will be lit in the upper part of the display. In the lower display area, three underlines (_ _ _) appear under the address line, which are actually prompting you for an address for the instruction you have just entered. True or false?
17. The instruction address is entered with the numeric keys that are located in the 2 top rows on the keyboard. True or false?
18. If you wanted to assign address 001 to an instruction, you need not enter the leading zeros of the address. Pressing numeric key 1 is all you must do to enter this address. True or false?
19. The SLC 150 can be programmed with the same pocket programmer used to program the SLC 100. True or false?

Question Group 4C

Answers in Unit 18

Program the rungs below and answer the questions that follow. Refer to the keystroke sequence in Figure 4.7 of the User's Manual if you need help entering this sample program. Make sure that you have cleared the processor memory before you begin to program these rungs.



1. After you have entered this program into an SLC 150 processor unit, there are _____ words of memory available.

   a. 1182 b. 1200 c. 867 d. None of these.

2. It is possible to examine an input address more than once in your program. True or false?

3. It is possible to examine an output address for an ON or OFF condition. True or false?

4. Enter the Run mode and cursor to the Examine ON instruction at address 003 in rung 1 using the NEXT and LAST keys. Is the ON LED in the display lit?

5. Cursor right to the Examine OFF instruction at address 004 with the NEXT key. Is the ON LED in the display lit?

6. If you have no external I/O devices connected, you cannot turn any inputs ON or OFF. Without using the Force function, all Examine ON instructions should be FALSE and Examine OFF instructions should be TRUE. True or false?

7. Using the NEXT key, cursor thru this program until the display reads

   1182 End

   Press the LAST key once and observe the display. Which square LEDs on the display are lit?

8. Press the LAST key three times. Which square LEDs on the display are lit?

9. Why is the ON LED lit?

Question Group 4D

Answers in Unit 18

Matching exercise – Match the SLC 150 error codes with their appropriate description.

Error Code	
27-2	ZCL programming error
27-5	Zero cross turn on failure
27-1	HSI module problem

Q-E Unit 5 Relay-Type Instructions

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

Hard-Wired Circuit 5-2
Examine ON 5-2
Examine OFF 5-2
Output Energize 5-2
Branch Instructions 5-4
Nested Branch 5-5
Output Latch 5-6
Output Unlatch 5-6

SYMBOLOGY

Question Group 5A

Answers in Unit 18

1. In a relay ladder rung, output devices are represented by relay coils or pilot lights. In a PC ladder rung, these output devices are represented by which instruction?
   a. Examine ON.
   b. Examine OFF.
   c. Output Energize.
   d. Branch Close.
2. Relay-type instructions are used for external I/O points only. True or false?
3. An Examine ON instruction examines a status bit for an ON condition. The instruction is TRUE when the status bit is ON. True or false?
4. Since the Examine ON instruction examines for an ON condition, it can be used to represent a normally open or a normally closed set of contacts if you want to know when your input device is ON. True or false?
5. Since the Examine OFF instruction examines for an OFF condition, it can be used to represent a normally open or a normally closed set of contacts if you want to know when your device is OFF. True or false?
6. It is possible to program a rung consisting only of an Output Energize instruction. True or false?

Question Group 5A
(continued)

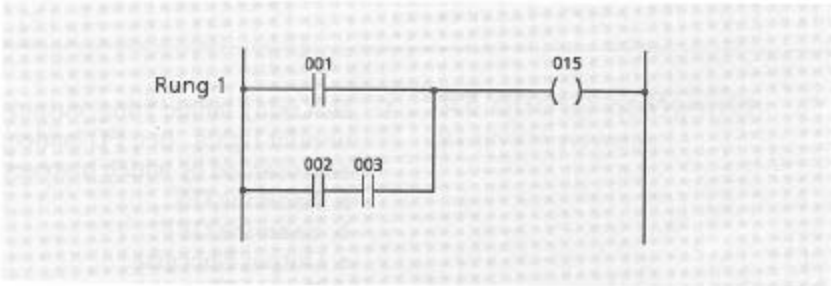
Answers in Unit 18

7. The instruction in question 6 will always be TRUE and the output will always be energized when in the Run mode. True or false?
8. The unconditional output energize instruction is generally an undesirable programming procedure. True or false?
9. If you have a normally open limit switch and you want a pilot light to go ON when the limit switch is open, programming a rung using an Output Energize instruction to represent the pilot light and an Examine OFF instruction to represent the limit switch could accomplish this. True or false?
10. A separate branch close instruction (⊣⊢) must be used for every rung of the parallel branch that you wish to close. True or false?

Question Group 5B

Answers in Unit 18

Study the rung below and answer the questions that follow.



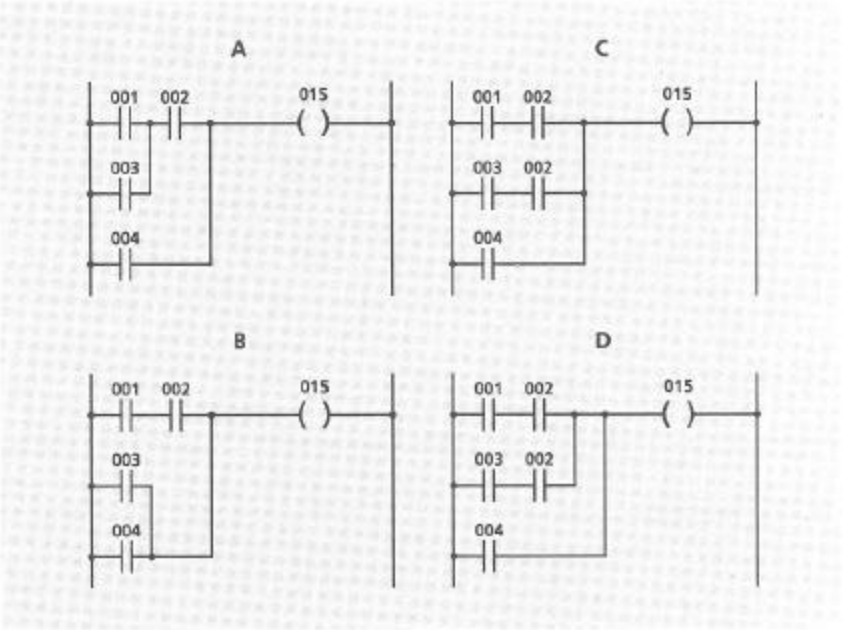
1. If the examine instruction at address 001 is TRUE, output 015 will also be TRUE. True or false?
2. If the examine instruction at address 002 is TRUE, output 015 will also be TRUE. True or false?
3. If the examine instructions at addresses 002 and 003 are TRUE, output 015 will also be TRUE. True or false?
4. If the examine instructions at addresses 001 and 003 are TRUE, output 015 will also be TRUE. True or false?
5. If the examine instructions at addresses 001 and 002 are TRUE, output 015 will also be TRUE. True or false?
6. If the examine instructions at addresses 001, 002 and 003 are all TRUE, the output at address 015 will be OFF (de-energized). True or false?

Question Group 5C

Answers in Unit 18

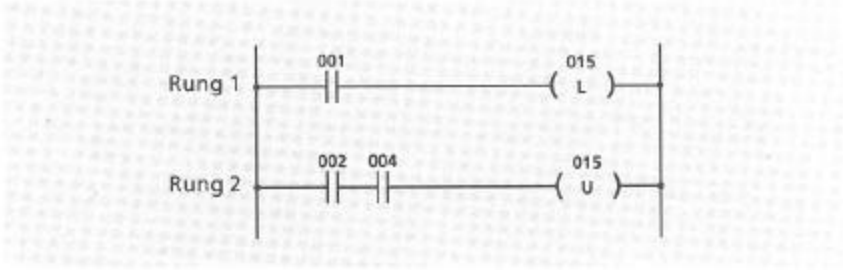
1. Which of the rungs in the following figure use the proper branching procedure for the Bulletin 1745 programming format? Enter the correct rung on the pocket programmer. If you need help, the correct key sequence for this problem is in the answer section.

Question Group 5C
(continued)



Question Group 5D
Answers in Unit 18

Study the rungs below and answer the questions that follow. Refer to Page 5-4 if you need help.



1. When input 001 is ON, output 015 is ON. True or false?
2. When input 001 makes a transition from TRUE to FALSE, output 015 remains ON. True or false?
3. When input 002 is ON, output 015 will unlatch and turn OFF. True or false?
4. When input 001 is ON output 015 is latched ON. Output 015 will unlatch only when inputs 002 and 004 are ON. True or false?
5. If output 015 is latched ON and power is removed, it will return to ON when power is restored. True or false?
6. If the Examine ON instructions at addresses 001, 002 and 004 are all TRUE, the output at address 015 will be OFF (de-energized). True or false?

Q-E Unit 6 Timer Instructions

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

SYMBOLOGY

Question Group 6A

Answers in Unit 18

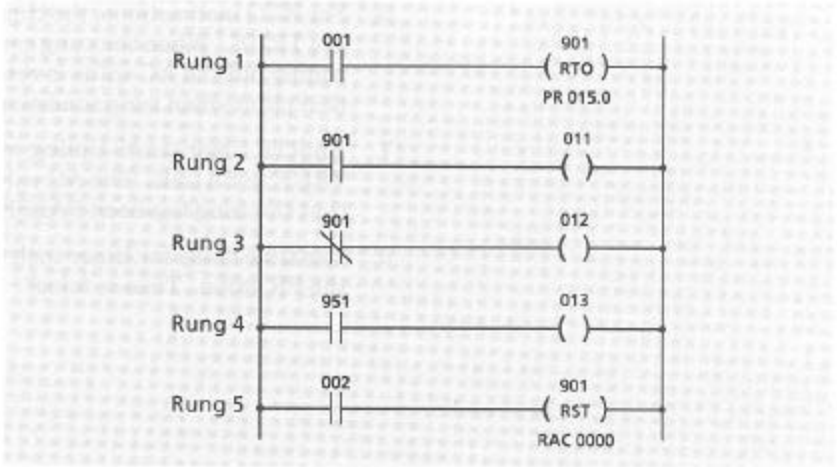
1. The RTO and RTF instructions both require a reset instruction. True or false?
2. Address assignments for timers must be from 901 thru 932. True or false?
3. The timer instruction counts 0.01 second time intervals. True or false?
4. The number of intervals counted while the timer is timing is called the accumulated value. True or false?
5. Because of the retentive nature of these timers, their AC value is retained when power is lost and when switching from the Run mode. True or false?
6. For RTO instructions, the timer starts timing when rung conditions are TRUE. When the rung conditions are FALSE, the timer stops timing but the AC value is retained so that timing is resumed from that point when rung conditions are TRUE again. True or false?
7. An RTF timer starts timing when rung conditions change from TRUE to FALSE. The AC value represents the cumulative time that the rung has been FALSE. True or false?
8. The time delay for timer instructions is actually the AC value minus the PR value. True or false?

Question Group 6A
(continued)
Answers in Unit 18

9. A PR value of 500 with an RAC value of 0 represents a time delay of:
   a. 500 seconds.
   b. 5 minutes.
   c. 5 hours.
   d. 50 seconds.
   e. none of the above.
10. Once a PR value has been set, it cannot be changed. True or false?
11. In order to change a PR value in the run mode, the PR value must be protected. True or false?
12. AC values cannot be changed. True or false?
13. The timing range for timer instructions is from 0.1 to 999.9 seconds. True or false?
14. Timer status bits can be assigned addresses from 951 thru 982. True or false?
15. Timer overflow status bits must be assigned an address that equals the timer address plus 50. True or false?
16. The overflow status bit is set ON when the AC value "overflows" from a count of 9999 to 0000. Examine ON instructions at the overflow address are then TRUE. True or false?
17. The RESET –(RST)– instruction is given an address that is the same as the timer address. True or false?
18. When the reset instruction is TRUE, status bits and overflow bits return to their initial state and the AC value is reset to the programmed RAC value. True or false?

Question Group 6B
Answers in Unit 18

Study the rungs below and answer the questions that follow. You may want to program these rungs and observe what happens on the pocket programmer.



Question Group 6B
(continued)
Answers in Unit 18

1. When input 001 is first turned ON, which of the following is correct?

   a. Rung 2 is TRUE.
   b. Rung 3 is TRUE.
   c. Rung 4 is TRUE.
   d. Rung 5 is TRUE.
   e. None of the above.

2. When the AC value of timer 901 reaches 015.0 and rung 1 remains TRUE, which other rungs are TRUE?

   a. 2. b. 3. c. 4. d. 2,4. e. 3,4.

3. Rung 4 will be TRUE as soon as the AC value exceeds the PR value. True or false?

4. When rung 5 is TRUE, what will the AC value be reset to?

   a. 015.0. b. 901. c. 0. d. AC value minus 50. e. None of these.

5. Address 951 is the address of:

   a. The examine instruction in rung 4.
   b. The overflow status bit of timer 901.
   c. Both of the above.
   d. None of the above.

6. Suppose that rung 1 is TRUE for ten seconds and then power is lost. What will the AC value of counter 901 be when power is restored?

   a. 110.0. b. 010.0. c. 001.0. d. 100.0. e. 000.1.

7. When will rung number 2 be TRUE?

   a. As soon as rung 1 is TRUE.
   b. When timer 901 has an AC value of 015.0 or more.
   c. Until timer 901 has an AC value of 015.0.
   d. While the RST instruction is TRUE.
   e. None of the above.

8. If the RTO instruction in rung 1 had an address of 925, what would the address of the Examine ON instruction in rung 4 have to be to examine the overflow status bit of timer 925?

   a. 901. b. 951. c. 925. d. 975. e. None of these.

9. When the AC value of timer 901 exceeds 9999, what output will be energized?

   a. 901. b. 011. c. 012. d. 013. e. None of these.

10. What initiates the timing function for timer 901?

    a. Examine ON at address 001 is TRUE.
    b. Examine ON at address 002 is TRUE.
    c. Examine ON at address 951 is TRUE.
    d. All of the above.
    e. None of the above.

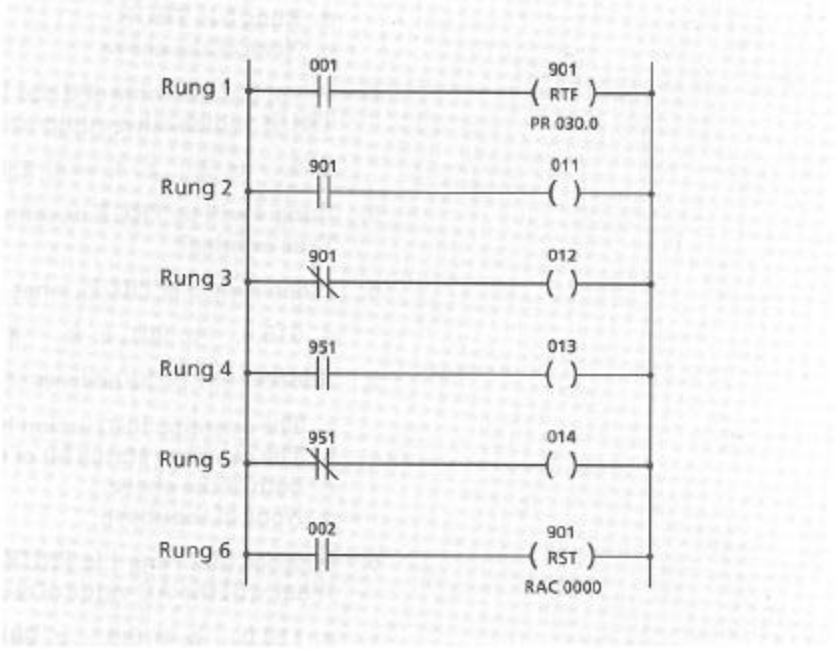
11. What is the address of the examine instruction that will reset timer 901?

    a. 001. b. 901. c. 951. d. 002. e. None of these.

Question Group 6C

Answers in Unit 18

Study the rungs below and answer the questions that follow. You may want to program these rungs and observe what happens on the pocket programmer.



1. When rung 1 goes TRUE, input 001 must be ON. True or false?
2. When rung 1 goes TRUE, timer 901 will start timing. True or false?
3. Rung 2 will be TRUE when:

   a. Rung 3 is TRUE.
   b. Rung 3 is FALSE.
   c. Input 002 is ON.
   d. Input 002 is OFF.
   e. None of the above.

4. Rung 4 will be TRUE as soon as the AC value exceeds the PR value. True or false?
5. Rung 3 is TRUE:

   a. When the AC value equals or exceeds 0300.
   b. Until the AC value equals or exceeds 0300.
   c. When rung 2 is TRUE.
   d. Upon power-up only.
   e. None of the above.

6. For an RTF timer, the AC value represents the cumulative time that the RTF rung has been FALSE. True or false?
7. If your accumulator value is up to 0250 and power to your system is lost, what will your accumulator value be when power is restored?

   a. 0000. b. 0300. c. 9999. d. 0250. e. None of these.

Question Group 6C
(continued)
Answers in Unit 18

8. When will output 012 be energized?

   a. Never.
   b. Always.
   c. When rung 3 is TRUE.
   d. When rung 1 is FALSE.
   e. None of the above.

9. When will output 013 be energized?

   a. Never.
   b. Always.
   c. When timer 901 AC value exceeds 9999.
   d. When input 002 is ON.
   e. Until timer 901 AC value exceeds 9999.

10. When will output 014 be energized?

    a. When timer 901 AC value exceeds 9999.
    b. When input 002 is ON.
    c. Until timer 901 AC value exceeds 9999.
    d. b and c.

11. It is possible for status bit 901 and status bit 951 to be ON at the same time. True or false?

12. When your AC value is at 500, status bit 901 will be OFF. True or false?

13. Suppose the AC value equals or exceeds the PR value and status bit 901 has gone OFF. If the reset instruction goes TRUE:

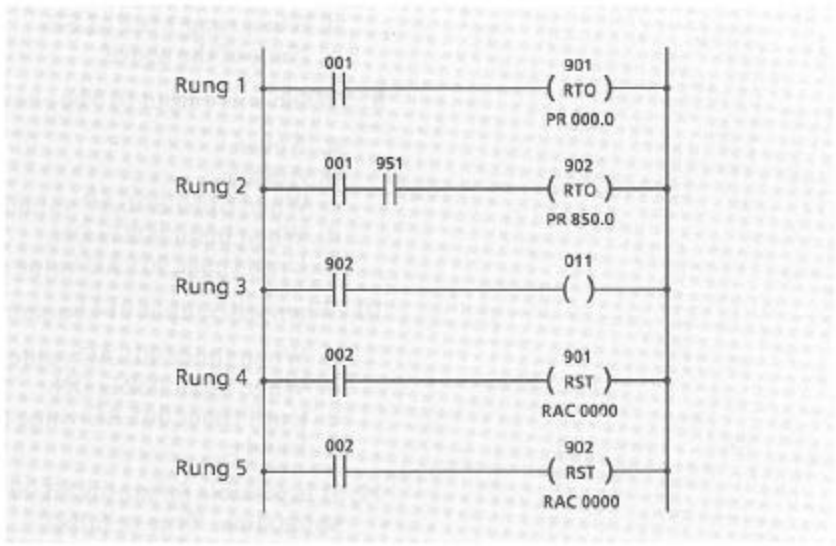
    a. The AC value will be reset to the RAC value.
    b. Status bit 901 will remain OFF.
    c. Status bit 901 will be reset to ON.
    d. Overflow status bit 951 will reset to OFF (if it has been set ON).
    e. a, b, and d above.

14. If RTF 901 is reset, status bit 901 will go ON after rung 1 makes a TRUE to FALSE transition. True or false?

Question Group 6D

Answers in Unit 18

Study the rungs below and answer the questions that follow. You may want to program these rungs and observe what happens on the pocket programmer.



1. After –] [– 001 goes TRUE, how much time must elapse before output 011 is energized?

   a. 850 seconds.
   b. 999.9 seconds.
   c. 1850 seconds.
   d. 1000 seconds.
   e. None of the above.

2. Which rung(s) must be TRUE in order for output 011 to be energized?

   a. 1. b. 1, 2. c. 1, 2, 4. d. 3. e. None of these.

3. When input 002 is ON, what will happen?

   a. Timer 901 will be reset.
   b. Timer 902 will be reset.
   c. Rung 4 will be TRUE.
   d. Rung 5 will be TRUE.
   e. All of the above.

4. After what time period will timer 902 start timing?

   a. Input 001 and status bit 901 must be ON. 850 seconds have elapsed.
   b. Input 001 and status bit 951 must be ON. 1000 seconds have elapsed.
   c. Input 002 must be ON.
   d. All of the above.
   e. None of the above.

5. Timer 901 starts timing when the rung conditions for rung 1 are FALSE. True or false?

Q-E Unit 7 Counter Instructions

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages/figures if necessary.

Preset Value (PR) 7-1
Reset Value (RAC) 7-1
Accumulated Value (AC) 7-1
Status Bit 7-2
Overflow Bit 7-2
Underflow Bit 7-2
Reset Instruction 7-2
FALSE –TRUE Transition Fig. 7.1
Up Counter –(CTU)– 7-4
Up-Down Counter 7-6

SYMBOLOGY

Question Group 7A

Answers in Unit 18

1. CTU counters start counting when rung conditions are TRUE. True or false?
2. CTU counters count FALSE to TRUE transitions of the rung containing the CTU instruction. True or false?
3. A CTU counter's AC value increases by 1 count for each FALSE to TRUE transition of the CTU rung. True or false?
4. If a CTU counter rung is FALSE and then goes TRUE, its AC value will increase by 1. When the rung goes FALSE again, the AC value will increase by 1 again. True or false?
5. CTD counters start counting when rung conditions are TRUE. True or false?
6. CTD counters count FALSE to TRUE transitions of the rung containing the CTD instruction. True or false?
7. A CTD counter's AC value will decrease by 1 for each FALSE to TRUE transition of the CTD rung. True or false?
8. The counter status bit is set ON when the AC value reaches the PR value. True or false?

Question Group 7A
(continued)
Answers in Unit 18

9. The counter status bit has an address that is:
   a. The same as the counter instruction.
   b. The counter address plus 50.
   c. The counter address minus 50.
   d. Automatically assigned when the instruction is entered.
   e. None of the above.

10. The counter overflow status bit has an address that is:
   a. The same as the counter instruction.
   b. The counter address plus 50.
   c. The counter address minus 50.
   d. Automatically assigned when the instruction is entered.
   e. None of the above.

11. The counter overflow status bit is set ON when:
   a. The AC value equals or exceeds the PR value.
   b. The AC value exceeds 9999.
   c. The AC value exceeds the PR value minus the RAC value.
   d. All of the above.
   e. None of the above.

12. The counter underflow bit is set ON when:
   a. The AC value exceeds 9999.
   b. The AC value goes below 0000.
   c. The AC value is less than the PR value.
   d. The AC value equals the PR value minus the RAC value.

13. The reset instruction is given the same address as the counter instruction. True or false?

14. When the reset instruction is TRUE, status, overflow and underflow bits are reset to OFF. True or false?

15. When the reset instruction is TRUE, the AC value is reset to the PR value. True or false?

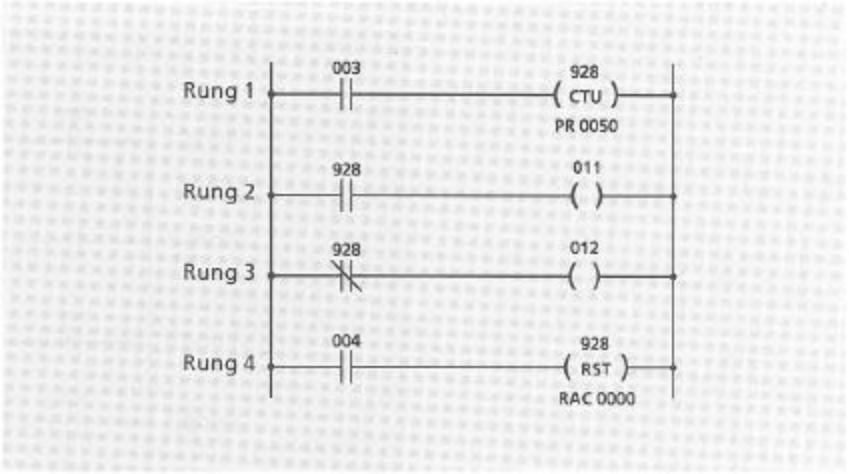
16. In the Run mode, when the cursor is positioned on the reset instruction and rung conditions are TRUE, the ON LED in the programmer display will be lit. True or false?

17. PR and AC values can be monitored and changed only in the Program mode. True or false?

18. Your PR values can be protected from changes in the Run and Test modes by using the PRT/UNPRT key. True or false?

Question Group 7B
Answers in Unit 18

Study the rungs below and answer the questions that follow. You may want to program these rungs and observe what happens.



1. The address for the CTU instruction in rung 1 is an acceptable address for a counter instruction. True or false?

2. If you wanted to examine the overflow status bit of the counter in rung 1, what address would you assign to the examine instruction?

   a. 901. b. 951. c. 952. d. 978. e. None of these.

3. When would output 011 be energized?
   a. Until the AC value equals the PR value.
   b. When the AC value equals or exceeds the PR value.
   c. Only when the AC value exceeds 9999.
   d. Only when the AC value goes below 0000.
   e. None of the above.

4. When would output 012 be energized?
   a. Until the AC value equals the PR value.
   b. When the AC value equals or exceeds the PR value.
   c. Only when the AC value exceeds 9999.
   d. Only when the AC value goes below 0000.
   e. None of the above.

5. Suppose your AC value is at 0048 and you lose AC line power to the controller. When power is restored to your controller, what will your AC value be?

   a. 0000. b. 0048. c. 0049. d. 0050. e. None of these.

6. Rung 4 goes TRUE and while it is TRUE rung 1 goes thru 5 FALSE to TRUE transitions. After that, rungs 1 and 4 go FALSE at the same time. What is the AC value of counter 928 at this time?

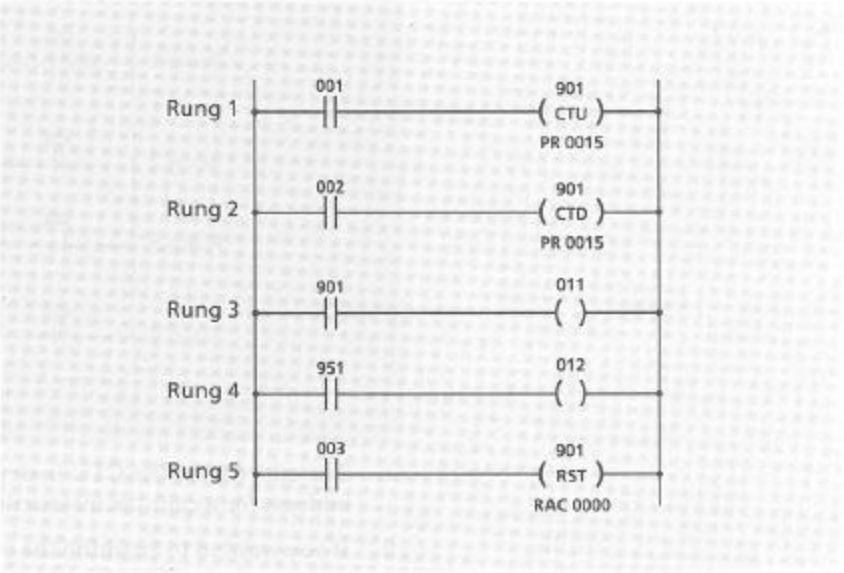
   a. 0005. b. 0004. c. 0000. d. 0001. e. None of these.

7. Suppose you were running a production line and you were using a counter to keep track of the number of parts produced. If you wanted to subtract the number of rejected parts, so your counter would count only good parts, you could program an up/down counter. True or false?

Question Group 7C

Answers in Unit 18

Study the rungs below and answer the questions that follow. You may want to program these rungs and observe what happens on the pocket programmer.



1. What is the address of the examine instruction that will cause the counter to increment?

   a. 001. b. 002. c. 901. d. 951. e. 003.

2. What is the address of the examine instruction that will cause the counter to decrement?

   a. 001. b. 002. c. 901. d. 951. e. 003.

3. What is the address of the examine instruction that will preset the counter to a count of zero when TRUE?

   a. 001. b. 002. c. 901. d. 951. e. 003.

4. When would output 011 be energized?
   a. When the up counter reaches a count of 0015.
   b. When the down counter goes below 0000.
   c. When status bit 901 is ON.
   d. All of the above.
   e. None of the above.

5. When would output 012 be energized?
   a. When the up counter exceeds a count of 9999.
   b. When the down counter goes below 0000.
   c. When status bit 951 is ON.
   d. All of the above.
   e. None of the above.

6. When input 003 is ON, which counter(s) would be reset?

   a. CTU 901. b. CTD 901. c. Both. d. None of these.

Question Group 7C
(continued)

Answers in Unit 18

7. Suppose rung 5 is TRUE and counters are reset. Inputs 001 and 002 are both actuated 5 times. What will the AC value of CTU 901 be?

   a. 0015. b. 0005. c. 0000. d. 0010. e. None of these.

8. What will the AC value of CTD 901 be in question 7?

   a. 0015. b. 0005. c. 0000. d. 0010. e. None of these.

9. From that point input 001 is actuated 9 times. What will the new AC value of CTU 901 be?

   a. 0001. b. 0019. c. 0009. d. 0011. e. None of these.

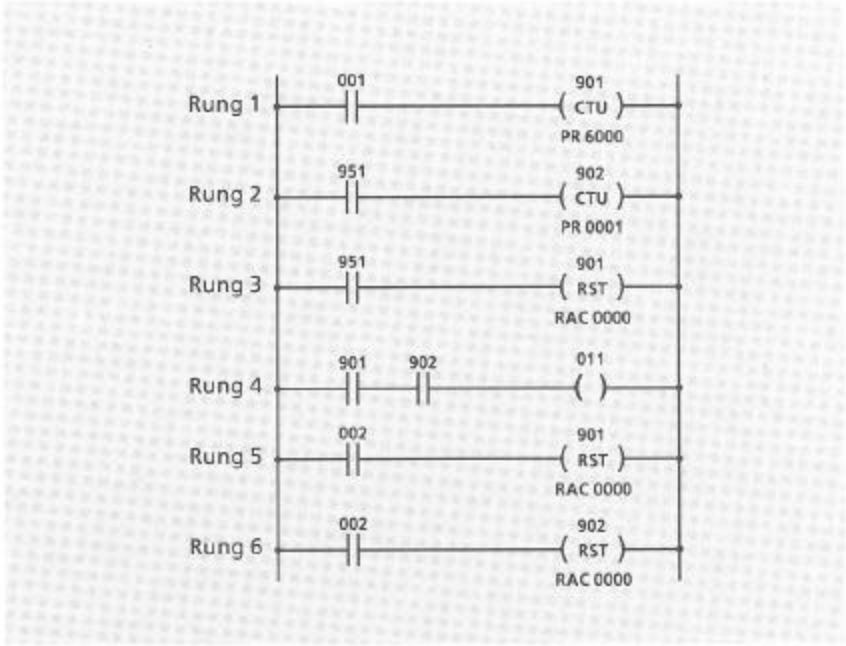
10. The AC value of CTU 901 will always be the same as CTD 901 during normal operation. True or false?

11. While the reset instruction is TRUE, both counters are disabled and the inputs 001 and 002 will not enable the counters. True or false?

Question Group 7D

Answers in Unit 18

Study the rungs below and answer the questions that follow.



1. It is possible to energize an output after a count of something greater than 9999 by cascading counters together. True or false?

2. When does counter 902 start counting?
   a. When the AC value of CTU 901 reaches 3000.
   b. When the AC value of CTU 901 reaches 6000.
   c. When the AC value of CTU 901 exceeds 9999.
   d. When the AC value of CTU 901 reaches 0001.
   e. None of the above.

Question Group 7D
(continued)
Answers in Unit 18

3. When will output 011 be energized?
   a. When the AC value of CTU 901 reaches 6000.
   b. When CTU 901 is reset.
   c. When CTU 902's AC value equals or exceeds 0001 *and* CTU 901's AC value equals or exceeds 6000.
   d. Then CTU 901's AC value exceeds 9999 *and* CTU 902's AC value equals or exceeds 0001.
   e. None of the above.

4. When output 011 is energized, how many counts have occurred?

   a. 13,000. b. 10,000. c. 0001. d. 16,000. e. None of these.

5. The AC value of counter 901 must reach 6,000 twice before output 011 will be energized. True or false?

6. If you wanted output 011 to go ON after a count of 10,500, what would you change the PR value of counter 901 to?

   a. 9999. b. 0500. c. 0501. d. 0001. e. None of these.

Q-E Unit 8 Timers with Long or Fine Time Base

Key Terms and Concepts

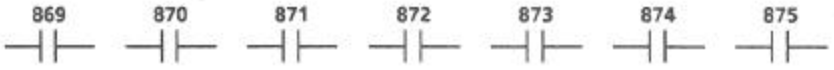
You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

Long Time Base Instructions 8-2

Fine Time Base Instructions 8-3

Scan Time Measurement 8-4

SYMBOLOGY



Question Group 8A

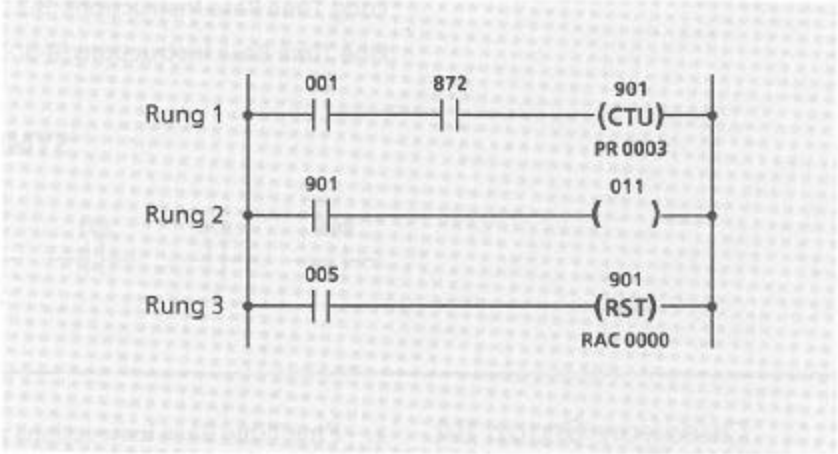
Answers in Unit 18

1. Fine Time Base instructions are bits that pulse ON and OFF at intervals of 10 milliseconds to one second. True or false?
2. If the scan time of your program is less than 10 milliseconds but more than 5 milliseconds, you could select a fine time base of 10 milliseconds using address 870. True or false?
3. The only way to determine the scan time of your program is to write down all the instructions you use in your program and add up the instruction execution times. True or false?
4. The pulse rate of the fine time base instruction at address 869 is the same as your measured scan time. True or false?
5. You can only use address 869 if your measured scan rate is less than 5.0 milliseconds. True or false?
6. By using a fine time base instruction with a counter instruction, you could program a time delay that is less than the 0.1 sec. provided by standard timers. True or false?
7. The rungs that you add to your program to measure scan time should be deleted from your program when you are ready to run. True or false?
8. Long time base instructions can be used with the SLC 100 and SLC 150 processors at addresses 875 and 874. True or false?
9. The long time base instruction allows you to program timers with greater resolution than the .1 second provided by the standard timers. True or false?
10. Fine time base instructions allow you to choose only one of four time bases. True or false?
11. 0.5 and 1.0 second time bases allow you to program timers with time delays up to 9999 seconds. True or false?
12. The SLC 150 1.0 second long time base is based on the processor scan cycle allowing you to program real time clock applications. True or false?

Question Group 8B

Answers in Unit 18

Study the SLC 100 program below and answer the questions that follow. You may want to program these rungs and observe what happens on the pocket programmer.



1. When input 001 is ON, rung 1 will be TRUE once every 40 milliseconds. True or false?

2. If input 001 stays ON, when will output 011 be energized?
   a. Immediately.
   b. After 12 seconds.
   c. After 120 milliseconds.
   d. Never.
   e. None of the above.

3. If you wanted a time delay of 240 milliseconds without changing the PR value of CTU 901, what address would you assign to the fine base instruction in rung 1?

   a. 870. b. 871. c. 873. d. 874. e. 875.

Q-E Unit 9 MCR and ZCL Instructions

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

Master Control Reset –(MCR)– 9-1
Zone Control Last State –(ZCL)– 9-1
Zone 9-1
Start Rung 9-1
End Rung 9-1
Non-retentive 9-1
Multiple ZCL Zones 9-3
Nesting ZCL Zones 9-4
Master Control Relay 9-5

SYMBOLOGY

Question Group 9A

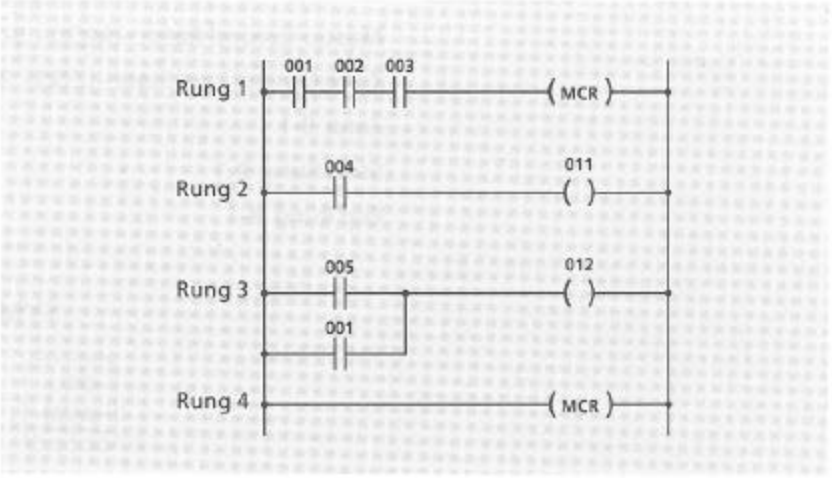
Answers in Unit 18

1. The master control reset instruction (MCR) allows you to use one set of condition instructions to control multiple outputs. True or false?
2. The master control reset instruction (MCR) consists of a start rung which contains the condition instructions for the zone, and an end rung which contains no conditional instructions and marks the end of the zone. True or false?
3. All non-retentive outputs between the MCR start and end rungs are controlled by the MCR instruction. True or false?
4. Any number of rungs may be programmed between the MCR start and end rungs providing memory space is available. True or false?
5. When the MCR start rung is TRUE, all non-retentive outputs within the zone are de-energized. True or false?
6. In an MCR zone when the start rung is FALSE, latch/unlatch instructions remain in their last state regardless of whether they were ON or OFF. True or false?
7. A timer or counter in an MCR zone with a FALSE start rung will have its AC value reset to the RAC value. True or false?
8. A sequencer instruction in an MCR zone with a FALSE start rung will have its step number retained. AC value stops incrementing and will be retained. Programmed outputs, completion and input satisfied bits will remain in their last states. True or false?
9. A reset instruction in an MCR zone with a FALSE start rung will remain in its last state. Instructions cannot be reset. True or false?
10. An MCR instruction is assigned an address that is between 033 and 099. True or false?
11. When the MCR rung is TRUE and you cursor to the MCR instruction on the display, the ON LED should be lit. True or false?

Question Group 9B

Answers in Unit 18

Study the rungs below and answer the questions that follow. You should program these rungs and observe how the MCR instruction works.



1. Turn inputs 1 through 5 ON. Outputs 11 and 12 will be ON. True or false?
2. Turn input 5 OFF. Outputs 11 and 12 will be ON. True or false?
3. Turn input 1 OFF. Outputs 11 and 12 will be OFF. True or false?
4. Turn input 1 back ON and input 2 OFF. Outputs 11 and 12 will be OFF. True or false?
5. Inputs 1, 2, and 3 are the condition instructions that control the MCR zone. True or false?
6. When inputs 1, 2, and 3 are ON, the outputs in the MCR zone operate normally. True or false?

Question Group 9C

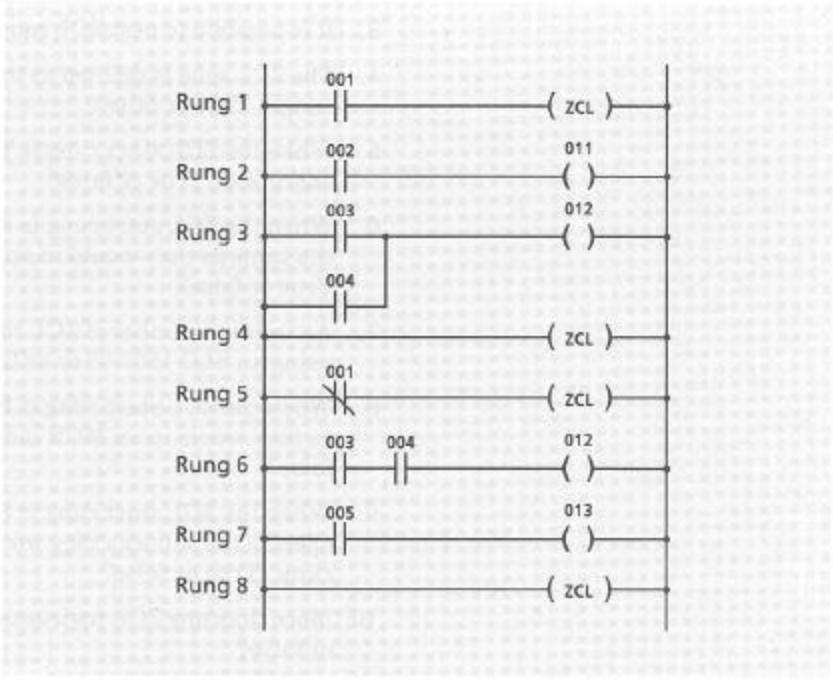
Answers in Unit 18

1. The ZCL instruction operates the same as the MCR instruction. True or false?
2. The ZCL instruction does not require an address. True or false?
3. It is possible to nest ZCL zones with the SLC 100? True or false?
4. The ZCL zone is defined as the rungs between the ZCL start and end rungs. True or false?
5. When the ZCL start rung is TRUE, non-retentive outputs are de-energized. True or false?
6. When the ZCL start rung is FALSE, timers and counters function the same as they would in an MCR zone with a FALSE start rung. True or false?
7. An infinite number of ZCL zones may be nested together when using an SLC 150? True or false?
8. When the ZCL start rung is FALSE, sequencers function the same as they would in an MCR zone with a FALSE start rung. True or false?
9. When the ZCL start rung is FALSE, latch/unlatch instructions operate the same as they would in an MCR zone with a FALSE start rung. True or false?
10. More than one ZCL instruction may be used in your program. True or false?
11. If two ZCL zones are used with an SLC 100, one zone must end before another can begin. True or false?
12. A ZCL or MCR instruction may be used instead of hard-wiring a Master Control Relay for emergency shut-down. True or false?
13. A Master Control Relay provides emergency shut-down capability for the operator of a machine. This system should be hard-wired to shut your controller down in the event of an emergency. A ZCL or MCR instruction cannot provide this degree of protection. True or false?
14. ZCL zones may not share common outputs. True or false?

Question Group 9D

Answers in Unit 18

Study the rungs below and answer the questions that follow.



1. It is possible to have two ZCL zones in a program as long as they don't overlap. True or false?
2. If input 001 is OFF, outputs in the lower ZCL zone will be held in their last state. True or false?
3. This program will not work properly because output 012 appears twice in the program. True or false?
4. If input 001 is ON, what must happen in order for output 012 to be ON?

   a. Rung 6 must be TRUE.
   b. Rung 3 must be TRUE.
   c. Input 2 must be OFF.
   d. Both a and b above.
   e. None of the above.

5. This program would work identically if MCR instructions were used in place of the ZCL instructions. True or false?

Q-E Unit 10 Sequencer Instructions

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

Time-Driven 10-1
Event Driven 10-1
SQI Input-Satisfied Status Bit 10-2
SQO Step Completion Bit 10-2
SQO Cycle Completion Bit 10-2
Reset Instruction –(RST)– 10-2
Sequencer Output 10-4
Sequencer Input 10-5
Sequencer Data 10-6
Group Numbers 10-7
Sequencer Data Form 10-8
Mask/Step Data 10-9

SYMBOLOGY

Question Group 10A

Answers in Unit 18

1. The Sequencer Output instruction (SQO) is retentive and requires a reset instruction, however the Sequencer Input instruction (SQI) is non-retentive and requires no reset instruction. True or false?
2. Sequencer instructions are typically used with machines or processes involving repeating operating cycles which can be segmented into steps. True or false?
3. Sequencer instructions must be assigned an address from_____thru _____.
4. Sequencers can have up to _____ steps and can be _____ driven or _____ driven.
5. When using a time-driven sequencer, what advances the sequencer to the next step?
   a. When the AC value equals 999.
   b. When the AC value equals the PR value.
   c. When the AC value equals the RAC value.
   d. When the input satisfied bit is ON.
   e. None of the above.
6. Time-driven sequencers count 1.0 second intervals. True or false?

Question Group 10A
(continued)

Answers in Unit 18

7. Event-driven sequencers count FALSE to TRUE transitions of the sequencer rung. True or false?
8. Event-driven sequencers move to the next step when their AC value equals the RAC value. True or false?
9. Sequencer output instructions set the ON/OFF status for one group of bit addresses for each step. True or false?
10. After the final step, the sequencer continues with step 0. True or false?
11. The sequencer input instruction examines the ON/OFF status of one group of bit addresses for each step. An input-satisfied status bit is set ON, when the status of the examined bit addresses matches programmed data. True or false?
12. The SQI sequencer input-satisfied status bit is assigned an address that is equal to the SQI instruction plus 50. True or false?
13. The SQO cycle completion bit is assigned the same address as the sequencer. True or false?
14. The SQO cycle completion bit goes ON when the sequencer completes its final step. True or false?
15. The reset (RST) instruction is given the same address as the sequencer. True or false?
16. When the reset instruction goes TRUE, which of the following statements are true?
    a. The SQI input-satisfied status bit is reset to OFF.
    b. The completion bit is reset to OFF.
    c. Sequencer instruction is reset to the programmed RAC step value.
    d. All of the above.
    e. None of the above.
17. The RST instruction must go FALSE before the sequencer can resume operation. True or false?
18. When programming sequencer data for the bit addresses to be controlled, a two-digit code is used to represent the ON/OFF status for each step. Instead of entering eight 1's or 0's only 2 digits are entered for each step. True or false?
19. If AC power is lost during sequencer operation and then restored, the sequencer will start over from step 0. True or false?
20. Sequencer status will be retained when switching from one mode to another. True or false?
21. Sequencer output instructions at addresses 901, 902, 903, 904 are used in programming high speed input modules. True or false?
22. SQI and SQO instructions can be time or event driven. True or false?
23. The SQI instruction is no different than the SQO instruction. True or false?

Question Group 10B

Answers in Unit 18

1. Sequencer Programming Example.
Use the Sequencer Instruction Data Form to document an SQO time-driven sequencer at address 901. You want to control outputs at bit addresses 011 through 016 for a simple 4-step sequence.

- Step 0 has no bits ON for 5 seconds.
- Step 1 has only bit 016 ON for 15 seconds.
- Step 2 has bits 011, 012, 013, and 014 ON for 30 seconds.
- Step 3 has bits 013, 014, 015, and 016 ON for 1 minute.

Try not to look at the answer until you have completed the form.

Use this form as a work sheet:

CLASSIFICATION: ☐ –(SQI)– ☐ –(SQO)– ADDRESS: ____ ☐ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: ____

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A				Data B	Data A	
Bit Addresses →											
Mask Data →											
Step Data → 0											
1											
2											
3											

Use this form for your final answer:

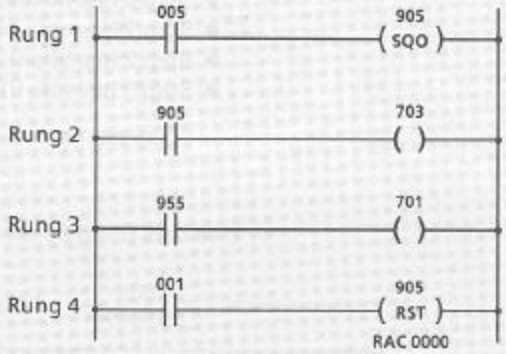
CLASSIFICATION: ☐ –(SQI)– ☐ –(SQO)– ADDRESS: ____ ☐ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: ____

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A				Data B	Data A	
Bit Addresses →											
Mask Data →											
Step Data → 0											
1											
2											
3											

Question Group 10C

Answers in Unit 18

Study the rungs below and the sequencer data form that follows. This information will help you answer the questions in group 10C.



CLASSIFICATION: ☐ -(SQI)- ☒ -(SQO)- ADDRESS: 905 ☒ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	0	0	1	1	1	1	1	1	3	F	
Step Data → 0	0	0	0	0	0	0	0	1	0	1	0.1
1	0	0	0	0	0	0	1	1	0	3	0.1
2	0	0	0	0	0	1	1	1	0	7	0.1
3	0	0	0	0	1	1	1	1	0	F	0.1
4	0	0	0	1	1	1	1	1	1	F	0.1
5	0	0	1	1	1	1	1	1	3	F	0.1
6	0	0	0	0	0	0	0	0	0	0	0.1
7	0	0	1	1	1	1	1	1	3	F	0.1
8	0	0	0	0	0	0	0	0	0	0	0.1
9	0	0	1	1	1	1	1	1	3	F	0.1

1. How many bit addresses are controlled by this sequencer?

   a. 4. b. 6. c. 8. d. 10. e. 12.

2. What is the mask data program code for this sequencer?

   a. 3F. b. FE. c. 0F. d. CE. e. 01.

3. Assuming normal operation, when input 005 is turned ON, the sequencer will perform the steps as you have programmed. When input 005 is turned OFF, the sequencer will stop at the step it was currently on and resume from that step when input 005 is turned ON again. True or false?

Question Group 10C (continued)

Answers in Unit 18

4. Suppose input 005 were turned OFF at step 3 in the Run mode. Inputs 011 through 014 would be energized. True or false?
5. If you wanted to control only outputs 011, 012, 013, and 014, what would the program entry code for your mask data be?

   a. C3. b. 0F. c. 2B. d. B2. e. None of these.
6. What is the preset value for each step?

   a. 10 sec. b. 0.1 sec. c. 1.0 sec. d. 100 sec. e. None of these.
7. When will rung 2 be TRUE?

   a. For 1.0 second after each step.
   b. For 10 seconds after each step.
   c. For one program scan after each step is completed.
   d. All of the above.
   e. None of the above.
8. For what step(s) of the sequence will all outputs be ON?

   a. 1. b. 5. c. 8. d. 7. e. 9.
9. For what step(s) of the sequence will all outputs be OFF?

   a. 1. b. 6. c. 8. d. 9. e. 16.
10. PR values can be changed in the Run and Test modes. True or false?
11. The sequencer RAC value should always be set to 0. True or false?
12. The reset instruction will always reset the sequencer to step 0 of the sequence. True or false?
13. When will bit address 703 be ON?
14. When input 001 is ON, the sequencer will be reset to step 0 and the completion bit will be reset to OFF. True or false?
15. While rung 4 is TRUE, the sequencer will be held on step 0. True or false?
16. If a bit address of the group controlled by your sequencer is masked, it may be used elsewhere in your program. True or false?

Question Group 10D

Answers in Unit 18

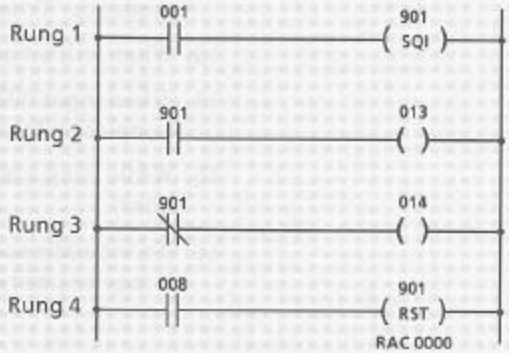
1. If you wish to energize an output only after a certain set of input conditions have been satisfied, a sequencer input instruction (SQI) can be used. True or false?
2. A typical application for a sequencer input instruction might be a packaging application where certain limit switches must be tripped before the conveyor can cycle to its next position. True or false?
3. How many bit addresses can the SQI instruction examine?

   a. 1. b. 2. c. 4. d. 6. e. 8.
4. What happens when all input conditions for a given step are satisfied?
   a. The sequencer advances to the next step.
   b. The sequencer is reset to step 1.
   c. The input-satisfied bit is set ON.
   d. All flags are reset to 0.
   e. None of the above.

Question Group 10E

Answers in Unit 18

Study the rungs below and the sequencer data form that follows. This information will help you answer the questions for group E.



CLASSIFICATION: ☒ -(SQI)- ☐ -(SQO)- ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 7

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	008	007	006	005	004	003	002	001	Data B	Data A	
Mask Data →	0	1	1	1	1	1	1	0	7	E	
Step Data → 0	0	0	1	1	1	0	0	0	3	8	1
1	0	0	0	0	0	1	1	0	0	6	1
2	0	0	0	1	1	0	0	0	1	8	1
3	0	1	1	0	0	0	0	0	6	0	2
4	0	1	1	1	1	1	1	0	7	E	1

Question Group 10E
(continued)

Answers in Unit 18

1. How many bit addresses are examined by this sequencer?
   a. 2. b. 4. c. 6. d. 8. e. None of these.
2. What is the bit address group number controlled by this sequencer?
   a. 1. b. 3. c. 5. d. 7. e. 9.
3. When will the input-satisfied bit be ON for step number 0?
   a. When inputs 3 and 8 are ON.
   b. When inputs 4,5, and 6 are ON.
   c. When rung 1 makes a FALSE -TRUE transition.
   d. When the sequencer advances to step 2.
   e. None of the above.
4. When will output 014 be energized?
   a. Always.
   b. Never.
   c. When the AC value exceeds 9999.
   d. Until the input satisfied conditions for a particular step are met.
   e. For the first second of each step.
5. The SQI sequencer will advance to the next step whenever the FALSE to TRUE transitions of the SQI rung match the programmed preset value. True or false?
6. If input conditions have not been met but the SQI rung goes thru a FALSE to TRUE transition, the sequencer will advance to the next step if the preset value for that step is 1. True or false?
7. The reset instruction can be used to reset the sequencer to any step. True or false?
8. The reset instruction will reset the sequencer to the programmed RAC step, reset the input-satisfied bit to 0, and reset the completion bit to 0. True or false?
9. When will the sequencer advance to step 4?
   a. When rung 1 goes thru one FALSE to TRUE transition at step 3.
   b. When rung 1 goes thru two FALSE to TRUE transitions at step 3.
   c. When the input-satisfied bit for step 3 goes ON.
   d. When rung 4 is TRUE.
   e. None of the above.
10. What conditions must be met for the input-satisfied bit to be ON for step 4?
    a. Inputs corresponding to program entry code 7E must be ON.
    b. Inputs 2 through 7 must be ON.
    c. All inputs controlled by this sequencer must be ON.
    d. All of the above.
    e. None of the above.

Q-E Unit 11 Special Sequencer Instruction Techniques

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

Cascading Sequencers 11-1
Reversing 11-4
SQI Driving an SQO 11-6
SQI Monitoring an SQO 11-7
Sequencer Jump Operation 11-10

SYMBOLOGY

Question Group 11A

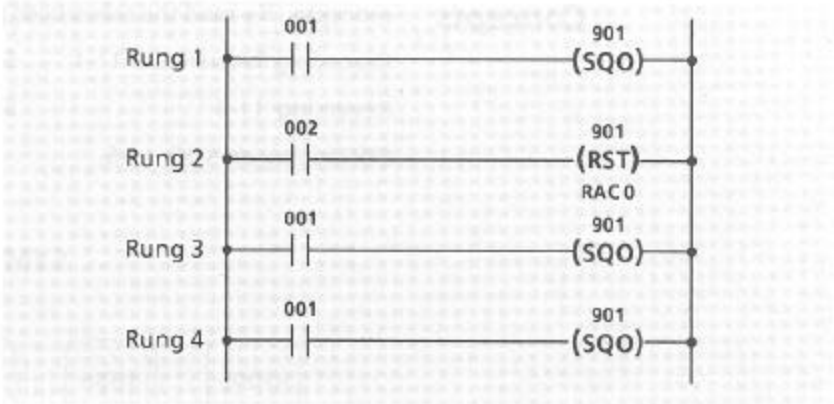
Answers in Unit 18

1. SQI instructions can be cascaded in order to monitor more than 8 input addresses. True or false?
2. SQO instructions can be cascaded in order to control a sequence of operations that energizes outputs in more than one group. True or false?
3. A reversing sequencer operation allows you to operate a sequencer in forward or reverse step order. True or false?
4. The reversing sequencer operation requires a ZCL zone to control reverse operation. True or false?
5. To program a reversing sequencer you must assign a special group number to the reversing SQO instruction. True or false?
6. If you had a sequencer at address 921, and you wanted to make it a reversing sequencer, what group number would you assign to the reversing sequencer instruction? _____
7. For a reversing sequencer operation, the preset values for both sequencers must be the same. True or false?
8. It is possible to program an SQO instruction that will energize outputs in response to the conditions satisfied in an SQI instruction. True or false?
9. With the SLC 100 you can't use an SQI instruction to monitor an SQO instruction. True or false?
10. If you don't want to use all the steps of a programmed SQO instruction, you can program a sequencer jump to skip the steps you don't want to use. True or false?
11. An SQI status bit can be used to advance an SQO to its next step. True or false?
12. The jump operation allows you to skip certain steps of your program if the proper conditions are met. True or false?

Question Group 11B

Answers in Unit 18

Study the rungs below and the sequencer data forms that follow. These will help you answer the questions for group 11B.



Question Group 11B
(continued)

Answers in Unit 18

Rung 1

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	0	0	0	0	0	0	0	0	0	0	
Step Data → 0	1	1	0	1	0	0	0	1	D	1	1
1	0	0	1	0	1	0	1	0	2	A	1
2	1	1	0	0	0	1	0	0	C	4	1
3	0	0	0	1	0	0	0	1	1	1	1

Rung 3

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	1	1	1	1	1	1	1	1	F	F	
Step Data → 0	1	1	0	1	0	0	0	1	D	1	1
1	0	0	1	0	1	0	1	0	2	A	1
2	1	1	0	0	0	1	0	0	C	4	1
3	0	0	0	1	0	0	0	1	1	1	1

Rung 4

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 1

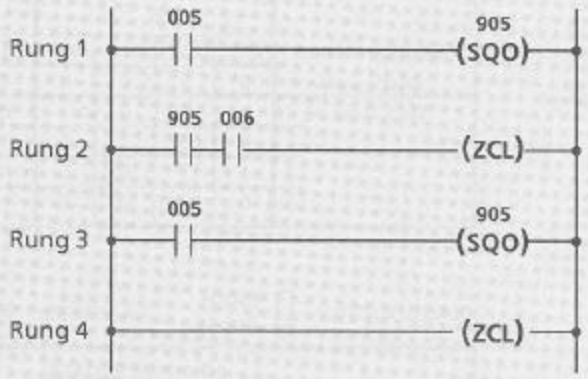
BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	118	117	116	115	114	113	112	111	Data B	Data A	
Mask Data →	1	1	1	1	1	1	1	1	F	F	
Step Data → 0	1	1	0	0	1	1	0	0	C	C	1
1	1	0	0	0	1	0	0	0	8	8	1
2	0	1	0	0	1	1	0	1	4	D	1
3	1	0	1	0	0	1	0	0	A	4	1

1. Rung 1 is used to synchronize the bit addresses of the sequencers in rungs 3 and 4. True or false?
2. Rung 2 will reset the SQO instructions to step 0. True or false?
3. The data in rungs 3 and 4 are very rarely the same when using cascaded SQO instructions. True or false?
4. The sequencers in rungs 1, 3 and 4 are time driven. True or false?
5. The sequencer in rung 4 controls Group 2 outputs. True or false?
6. The mask data of a Dummy rung is always Hexadecimal FF in the program code. True or false?

Question Group 11C

Answers in Unit 18

Study the rungs below and fill out the sequencer data form for SQO 905 in rung 3. You want to program a reversing sequencer so select the right group number, program code, and preset values. The correct answer is located in the answer section. Try not to look at the answer until you have completed the blank form below.



CLASSIFICATION: ☐ -(SQI)- ☒ -(SQO)- ADDRESS: 905 ☒ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	0	0	1	1	1	1	1	1	3	F	
Step Data → 0	0	0	0	0	0	0	0	0	0	0	0.5
1	0	0	0	0	0	0	0	1	0	1	0.5
2	0	0	0	0	0	0	1	0	0	2	0.5
3	0	0	0	0	0	1	0	0	0	4	0.5
4	0	0	0	0	1	0	0	0	0	8	0.5
5	0	0	0	1	0	0	0	0	1	0	0.5
6	0	0	1	0	0	0	0	0	2	0	0.5

CLASSIFICATION: ☐ -(SQI)- ☐ -(SQO)- ADDRESS: ____ ☐ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: ____

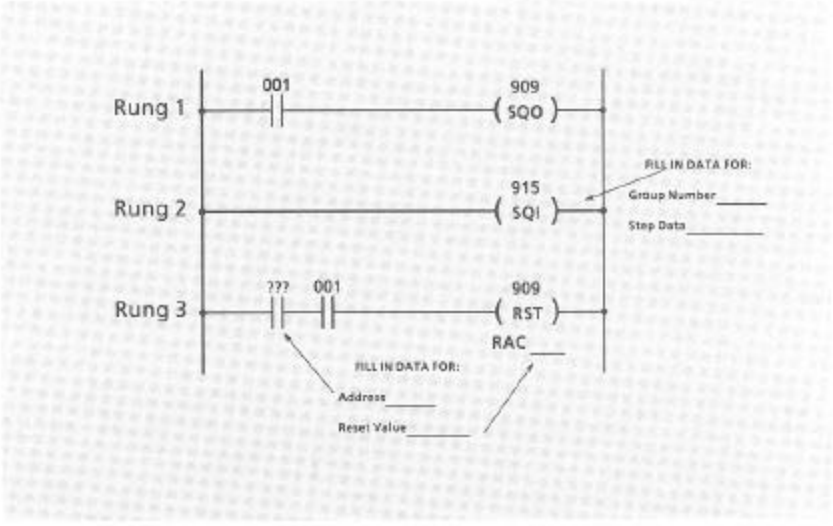
BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →									Data B	Data A	
Mask Data →											
Step Data → 0											
1											
2											
3											
4											
5											
6											

Question Group 11D
Answers in Unit 18

1. Suppose that you are controlling a long sequence of events with an SQO sequencer. You would like to know when the sequencer is on the last step of the sequence. You could program an SQI sequencer to monitor the SQO instruction for the last step. True or false?
2. If you want to monitor step 5 of an SQO instruction at address 925, you could program an SQI sequencer, select group number _____, and enter _____ for the step number.
3. If you wish to know when a sequence of events has reached or past a certain step in your application, you could program a
   a. An SQI monitoring an SQO
   b. An SQI instruction using a > (greater than) status bit
   c. An SQO instruction using a < (less than) status bit
   d. a and b
   e. a and c
4. If you wanted to know when your application had completed step 25 but not yet started step 30, you could program two SQI instructions to monitor an SQO instruction. The SQI instructions would have step data of:
   a. 25, 29
   b. 24, 28
   c. 25, 30
   d. 26, 30
   e. 26, 28

Question Group 11E
Answers in Unit 18

1. Suppose that you are controlling a bottling operation for a soft drink line. You bottle Brand A three days a week and Brand B 2 two days a week. Both processes are controlled by an SQO instruction. Brand A requires 15 steps of the sequencer operation but when you bottle Brand B, you can skip steps 5 through 9 of the sequencer operation. Given the ladder diagram below, fill in the missing information to complete the sequencer jump operation.



Q-E Unit 12 High Speed Input Programming

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

HSI Counter Instructions 12-2
Maximum Pulse Rate 12-2
Normal Reset 12-3
Single Shot Reset 12-3
Cascading 12-4
Scan Time 12-4
Socket End Plug 12-5

Question Group 12A

Answers in Unit 18

1. High speed counting is accomplished in your program by a(n)

   a. time driven SQO instruction.
   b. special event driven SQI instruction.
   c. special event driven SQO instruction.
   d. time driven SQO instruction.
   e. HSI instruction.

2. The controller is designed to handle up to _____ HSI module(s).

   a. 8
   b. 6
   c. 4
   d. 12
   e. 1

3. The corresponding HSI counter instruction addresses are

   a. 601-608.
   b. 601-604.
   c. 901-908.
   d. 901-904.
   e. 951-954.

4. You must program a 4-digit RAC value for a single shot reset. The first two digits must be 99. What do the other two digits represent?

   a. The instruction number.
   b. The counter location.
   c. The step number you want the sequencer set to.
   d. The SSR number.
   e. They have no significance.

5. What is the maximum pulse rate of one HSI module?

   a. 5MHz
   b. 5KHz
   c. 2KHz
   d. 10KHz

6. What happens when the maximum pulse rate is exceeded?

   a. Instruction 601 goes FALSE.
   b. The HSI counter is disabled.
   c. Instruction 901 goes FALSE.
   d. Both a and b.
   e. Both c and b.

Question Group 12A
(continued)
Answers in Unit 18

7. What happens when you remove the end plug, which is inserted in the expansion socket, while the processor is in the Run mode?

   a. A CPU fault will occur and the processor will always shut down.
   b. The processor may shut down.
   c. Nothing if you have a lithium battery back-up.
   d. The program will be temporarily suspended.

8. Is it possible to cascade sequencers used in HSI counters?

   a. Yes.
   b. No.
   c. Maybe.
   d. Only on the SLC 100.
   e. I don't know.

Question Group 12B
Answers in Unit 18

1. HSI counter instructions are very similar to some sequencer instructions described earlier. True or False?
2. To obtain the fastest possible output control when using your HSI counter, use the external outputs located on the expansion unit. True or false?
3. As the frequency of your HSI counter increases, your scan time also increases. Therefore, programming preset values of 1 will decrease your scan rate. True or false?
4. If the processor is placed in the Test-Single Scan mode and the HSI counter start rung is TRUE, will the HSI count in real time and increment the accumulator as it would in the Run mode? Yes or no?
5. The HSI module should not be connected or disconnected when power is applied to the processor unit. True or false?
6. If you use two or more HSI modules the maximum pulse rate, for each module, is less than if you were using only one module. True or false?
7. The single shot reset is used to reset the counter to a specified step without losing any high speed input counts. True or false?

Question Group 12C

Answers in Unit 18

An SLC 150 and HSI module is being used to count cans in a high speed packaging application. The cans pass a solid state sensor at a rate of 200 cans per second. The solid state sensor is connected to an HSI module. After 24 cans are fed into a case, the conveyor belt is stopped. The full case is removed and replaced with an empty one. The conveyor then carries the next 24 cans into the empty case. This process may be summarized in 2 steps.

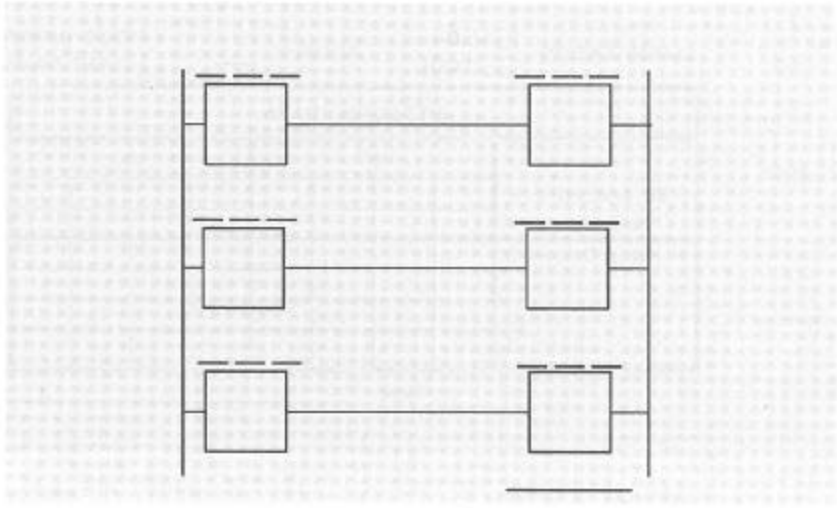
Step/operation 0 - Conveyor ON, fill case
Step/operation 1 - Conveyor OFF, eject case

The conveyor is controlled by SLC 150 output address 111.

The eject solenoid is controlled by SLC 150 output address 112.

A limit switch at address 001 senses when an empty case is in position. The empty case, once in position, will reset the process to step 0 and enable the conveyor. Hint: Use a single shot reset so the HSI module does not miss counting any cans.

Complete the ladder diagram and sequencer data table to perform the application in 12C.



Question Group 12C
(continued)

Answers in Unit 18

CLASSIFICATION: ☐ –(SQI)– ☐ –(SQO)– ADDRESS: ___ ☐ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: ___

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A				Data B	Data A	
Bit Addresses →											
Mask Data →											
Step Data → 0											
1											
2											
3											

Type: _____

Use this form for your final answer: nswer.

CLASSIFICATION: ☐ –(SQI)– ☐ –(SQO)– ADDRESS: ___ ☐ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: ___

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A				Data B	Data A	
Bit Addresses →											
Mask Data →											
Step Data → 0											
1											
2											
3											

Type: _____

Q-E Unit 13 Shift Register Instruction

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

Status Data 13-1
8-bit Register 13-1
Shift Right 13-1
Shift Left 13-1
Most Significant Bit 13-1
Least Significant Bit 13-1
Event-Driven 13-2
Time-Driven 13-3
Cascading 13-4
Circulating 13-5

SYMBOLOGY

Question Group 13A

Answers in Unit 18

1. Shift registers can be used to control machines or processes where parts are continually shifted from from one position to the next. True or false?
2. Address assignments are not important for shift register instructions. True or false?
3. When programming a shift register, a ZCL zone is used to control the shifting process and a sequencer instruction is used to control the shift rate. True or false?
4. If you were programming a shift register instruction and you wanted to shift data through the outputs of expansion unit number 2 from least significant bit to most significant bit, what address would you assign to the shift register instruction?

   a. 211. b. 218. c. 111. d. 118. e. None of the above.
5. If you needed a shift left register for the application in question 4, what address would you assign to the shift register instruction?

   a. 211. b. 218. c. 111. d. 118. e. None of the above.
6. You can program a shift register instruction to shift 1's or 0's through the register. True or false?
7. You can program shift register instructions to be either time or event-driven. True or false?

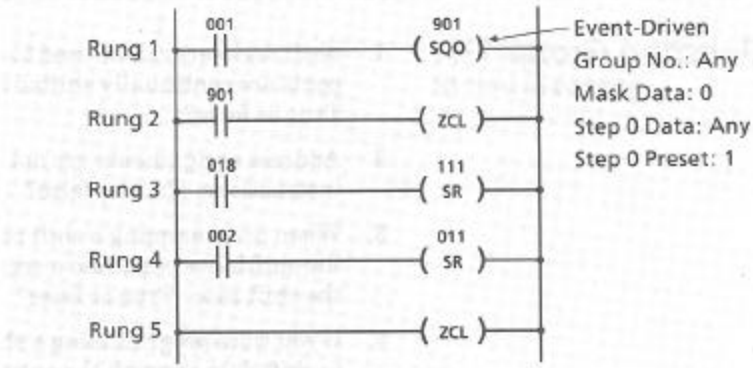
Question Group 13A
(continued)
Answers in Unit 18

8. An event driven shift register instruction shifts status data (either a 1 or a 0) on each FALSE-to-TRUE transition of the sequencer instruction that drives it. True or false?
9. A time-driven shift register instruction shifts status data at a clocked rate that equals the preset value you have programmed for the SQO instruction when rung conditions are TRUE. True or false?
10. What is the most significant bit address in group number 5? _____

    a. 118. b. 111. c. 518. d. 511. e. None of the above.

Question Group 13B
Answers in Unit 18

Study the rungs below and answer the questions that follow. You may want to program these rungs and observe what happens on the pocket programmer.



1. When input 002 is ON, a 1 is shifted into the 011 register every time rung 1 goes through a FALSE-to-TRUE transition. True or false?
2. If input 002 is OFF, a 0 is shifted into the register every time rung 1 goes through a FALSE-to-TRUE transition. True or false?
3. In rung 4, the shift register instruction is assigned the least significant bit address in output group number 0. If data is continually shifted through the register, what address will data be shifted to after address 018?

   a. 019. b. 011. c. 118. d. 111. e. None of the above.
4. When data leaves address 118, it is shifted back to address 011. True or false?
5. The order of rungs 3 and 4 is not critical to the shift register operation. True or false?

Question Group 13C

Answers in Unit 18

1. You can program a shift register instruction to shift data in a circulating manner so that data leaving the most significant bit address is shifted back into the least significant bit address or vice versa. True or false?
2. If you wanted to produce an external output when a certain bit in an internal shift register is ON, you could program a rung with an Examine ON instruction corresponding to the address of the shift register bit address. You could then assign an external output address to an Output Energize instruction. When the bit address in the shift register goes ON, the Examine ON instruction will be TRUE, causing the output to be energized. True or false?
3. If you program an internal shift register instruction, you can only shift data left. True or false?

Q-E Unit 14 Special Internal Instructions

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

Auto/Man Switch Instruction 14-1

Program Initialization Instruction 14-2

TCAT Power-up Instruction 14-5

Breakpoint Instruction 14-5

Programmable EEPROM Auto-Load 14-7

Battery Status Instruction 14-7

Triac Zero-Cross Turn-On 14-8

Question Group 14A

Answers in Unit 18

1. The program initialization instruction is a bit that pulses ON and OFF in accordance with the processor I/O scan. True or false?
2. The program initialization instruction has two hard-wired terminals on the processor so you can wire input devices directly. True or false?
3. The program initialization instruction is a bit in memory that is set ON for the first program scan under certain conditions. True or false?
4. If you use the pocket programmer to place the processor unit in the Run or Test modes, the program initialization instruction will go ON for the first program scan and then turn OFF. True or false?
5. You can use bit address 868 to program non-retentive timers, counters and non-retentive latch instructions. True or false?
6. The program initialization instruction is an internal examine instruction that you can use to initialize your program to a known state on power-up. True or false?
7. Breakpoint instructions can be programmed on both the SLC 150 and SLC 100. True or False?
8. Address 864 can be programmed in the SLC 150 to allow the EEPROM memory module to clear processor faults caused by memory errors. True or False?
9. The status of bit address 865 can be examined for battery status. True or False?
10. A breakpoint output instruction should be used in place of a hard-wired E-Stop. True or False?
11. Triac zero-cross turn ON instructions are used to minimize noise generated when switching loads. True or False?
12. To enable triac zero-cross turn ON, status bit address 866 must be set ON with:
    a. an unconditional rung.
    b. a conditional rung.
    c. neither a or b.
    d. either a or b.

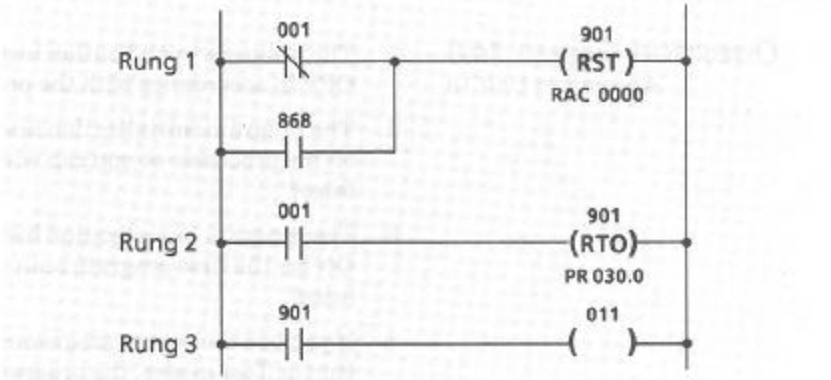
Question Group 14A
(continued)
Answers in Unit 18

13. Although special functionality of addresses 864, 865 and 866 are unique only to the SLC 150, these addresses are simple Relay type instructions in the SLC100. True or false?

14. Address 865 is used for the programmable EEPROM auto-load feature. True or false?

Question Group 14B
Answers in Unit 18

Study the rungs below and answer the questions that follow. You may want to program these rungs and observe what happens on the pocket programmer.



1. When power is first applied, the reset instruction in rung 1 is TRUE. True or false?

2. When input 001 is ON, timer 901 starts timing. When will status bit 901 be ON?

   a. As soon as input 001 is ON.
   b. After input 001 goes OFF.
   c. When rung 1 is TRUE.
   d. After 30 seconds.
   e. After 300 seconds.

3. What happens when rung 2 goes FALSE?

   a. Timer 901 stops timing.
   b. Timer 901 is reset.
   c. Input 001 is OFF.
   d. All of the above.
   e. None of the above.

4. Timer 901 starts timing when input 001 is ON. The accumulator reaches a value of 025.0 and power is lost. When power is restored, what will the AC value be?

   025.0. b. 010.0. c. 000.0. d. 999.9. e. None of these.

Question Group 14B
(continued)
Answers in Unit 18

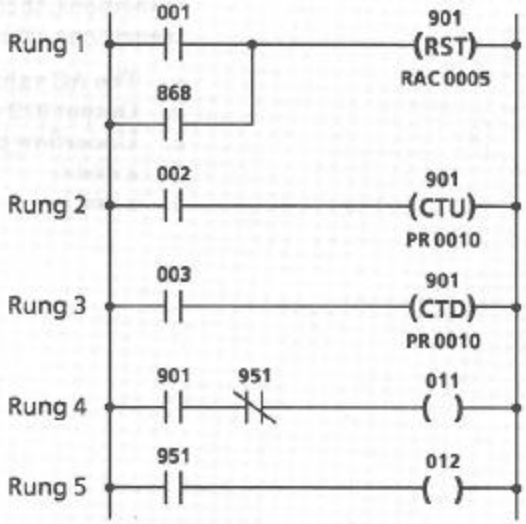
5. Timer 901 starts timing when input 001 is ON. The accumulator reaches a value of 025.0 and input 001 is turned OFF. When input 001 is turned ON again, what will the AC value be?

   a. 025.0. b. 010.0. c. 000.0. d. 999.9. e. None of these.

6. Since the AC value is reset to 0 when power is lost and restored, this type of timing example is classified as a(n) _____timer.

Question Group 14C
Answers in Unit 18

Study the rungs below and answer the questions that follow.



1. What is the address of the examine instruction that will cause the counter to increment?

   a. 001. b. 002. c. 901. d. 951. e. 003.

2. If power is lost and then restored, the AC values of the counters will be:

   a. The same as they were before power was lost.
   b. Equal to the programmed preset values.
   c. Reset to 5.
   d. Reset to 0000.
   e. None of the above.

3. What is the address of the examine instruction used to reset the counters during program execution?

   a. 003. b. 901. c. 001. d. 002. e. 868.

Question Group 14C
(continued)
Answers in Unit 18

4. Output address 011 will be turned ON when:

   a. The AC value is greater than or equal to 10.
   b. The AC value is less than or equal to 10.
   c. The AC value is less than 0.
   d. The AC value is greater than 9999.
   e. None of the above.

5. Output address 012 will be turned ON when

   a. The AC value is greater than or equal to 10.
   b. The AC value is less than or equal to 10.
   c. The AC value underflows below 0.
   d. The AC value is greater than 9999.
   e. None of the above.

6. If the AC value is 2, and input 003 makes three FALSE to TRUE transitions, then input 002 makes three FALSE to TRUE transitions, which of the following will be TRUE?

   a. The AC value would be 2.
   b. Output 012 would be ON.
   c. Underflow bit 951 would be OFF.
   d. a and c.
   e. a and b.

Q-E Unit 15 Program Editing

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

Cursor Control 15-2

Search Function 15-4

Remove / Insert Functions 15-6

Cancel Command 15-9

Clear Memory Mode 15-9

Protecting PR Values 15-9

Changing Instructions in the Program Mode 15-9

SYMBOLOGY

Question Group 15A

Answers in Unit 18

1. The RUNG, NEXT and LAST keys allow you to move through your program and stop at any instruction you wish to see on the display. True or false?
2. Since the pocket programmer can display only one instruction at a time, the cursor keys are necessary to help you move through your program for editing or troubleshooting purposes. True or false?
3. The cursor functions in which of the following modes of operation?

   a. Run.
   b. Test.
   c. Load EEPROM Module.
   d. Program.
   e. Clear Memory.

4. When you are cursored on the first instruction of a rung, the display will indicate this by displaying the symbol ⊢ in front of the address information. True or false?
5. While in that rung, if you cursor right (NEXT), the instruction to the immediate right of the first instruction will be displayed. True or false?
6. If you are cursored on the output instruction in the last rung of your program and you press the NEXT key, the pocket programmer will display:

   585 End

   Although your program may vary in length, this message is intended to tell you that you are at the end of your program and you have so many words of memory remaining. True or false?

Question Group 15A
(continued)
Answers in Unit 18

7. If you are cursored on the output instruction in the last rung of your program and you press RUNG, LAST, you will move to the first instruction in the first rung. True or false?
8. If you cursor to a sequencer instruction, the instruction LED will light and its address and rung number will be displayed. By using the NEXT key, you can step through the instruction and access other data associated with that instruction. True or false?
9. If you have a long program, the SEARCH function will help you save time finding a specific instruction. The SEARCH function can also help you find the start or end of rungs or the start or end of your program. True or false?
10. If you are cursored to a point in the middle of your program and are searching for a specific instruction, the search will wrap around and search your whole program until the instruction is found. True or false?
11. To locate the start of your program, the correct keystroke sequence is: SEARCH, RUNG, 1, ENTER. True or false?
12. If you are searching for a specific instruction and the pocket programmer displays the message

    nF

    (not found), the instruction you were searching for was not found in your program. True or false?
13. You can add and delete examine instructions and entire rungs in your program with the REMOVE and INSERT keys only in the Program mode. True or false?
14. If you want to add a rung to your program, you must cursor to the rung that will follow the rung you are about to insert and press INSERT, RUNG. After this key sequence you can enter the new rung. True or false?
15. A new rung may only be added in the Program mode. True or false?
16. When you remove or insert an examine instruction, this change is automatically recorded in RAM memory. True or false?
17. If you cursor to an output instruction in a rung of your program and remove that output instruction, the conditional instructions will be automatically removed so that the whole rung is removed. True or false?
18. By entering mode 1 and following the prompting messages on the pocket programmer, you can clear the processor RAM memory. True or false?
19. Mode 1 can also be used to clear the memory of an EEPROM module. True or false?
20. Once RAM memory is cleared, the controller automatically enters the Program mode. True or false?
21. Sequencer data can only be changed in the Program mode. True or false?
22. Adding or deleting steps of a sequencer instruction can be performed through an editing function in the Program mode. True or false?

Question Group 15A
(continued)
Answers in Unit 18

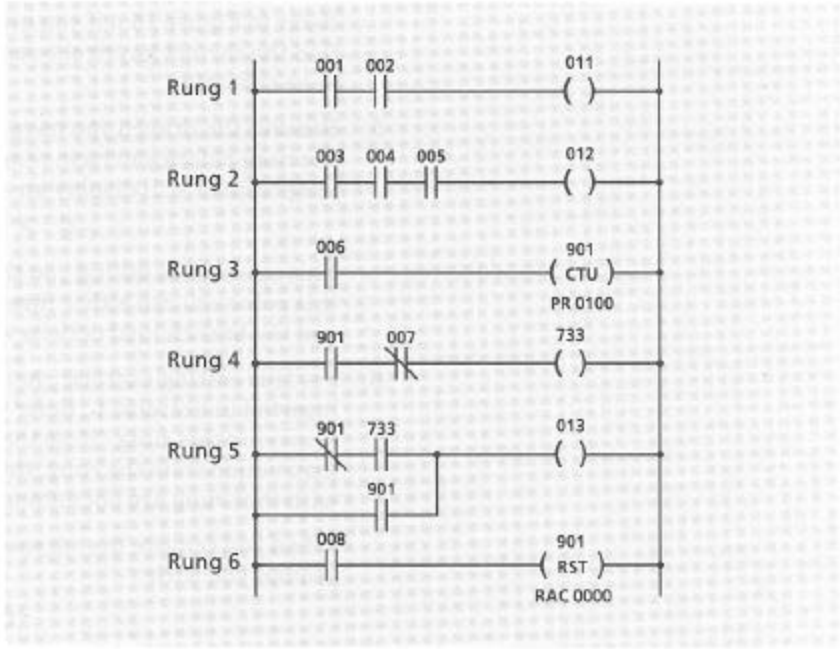
23. When you cursor to a sequencer instruction, each time you press the NEXT key, one of the values programmed as sequencer data will be displayed. True or false?

24. If you want to change a sequencer parameter in the Program mode you must cursor to the sequencer instruction and press the sequencer key associated with that instruction and then press the ENTER key. From that point you are able to change any data or preset value. True or false?

25. Changes that you make to sequencer data are entered into a temporary storage register. When you have finished editing the sequencer, you must press ENTER to register the last change you have made. These changes will not be permanently recorded in RAM memory until the ENTER key is pressed again. True or false?

Question Group 15B
Answers in Unit 18

Program the following rungs and complete the program editing exercises that follow.



1. After you have entered this program, stay in the Program mode. An additional rung must be entered in this program that will follow rung 4. The rung you want to insert will examine status bit 951 for an ON condition and energize output address 016 when rung conditions are TRUE. See Figure 15.5 if you need help inserting the new rung.

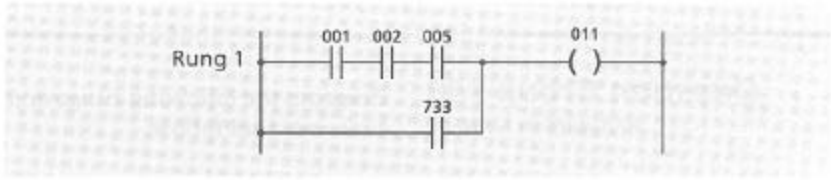
   After you have entered this rung, cursor through your program and verify that the rung you have just entered has been inserted between rungs 4 and 5 of your original program. Your edited program will now contain 7 rungs. Check your final program with the answer for Question / Exercise 1, group B, in the answer section. Verify that the rung you have just entered is now rung 5.

Question Group 15B
(continued)
Answers in Unit 18

2. For this exercise you will remove an examine instruction from rung 2. You must remove the instruction that examines address 005 for an ON condition. Refer to Figure 15.3 if you need help removing this examine instruction. Check the new configuration of rung 2 in the answer section and make sure that it matches your edited rung.

3. The Examine ON instruction that you removed in exercise 2 should now be placed in rung 1. Insert this examine instruction between the Examine ON at address 001 and the Examine ON at address 002. If you need help inserting this instruction, refer to Figure 15.3. Check the answer section for the correctly edited rung and compare this with the rung you have just edited.

4. Add a branch to the new rung 1 so that the ladder diagram looks like this:



Refer to Figure 15.4 if you need help inserting this branch. You will find the correct keystroke sequence for adding this branch in the answer section.

5. Remove the branch you have just entered. If you attempt to leave the rung before you have removed all of the branch instructions, the programmer will display an error message

rE P E64

This message is telling you that you have created a short circuit. At this point you can either insert a new instruction or, press ENTER, and the branch instructions will be automatically removed.

6. For this editing exercise, you will remove the rung that you have inserted in exercise 1. Refer to Figure 15.5 if you need help removing this rung. Note that when you remove the output instruction from the rung, the conditional instructions associated with that rung are not automatically removed.

When you remove the output instruction, the programmer will prompt you with

-EL-

You must either add another output instruction or remove all the remaining conditional instructions in that rung. The correct keystroke sequence for removing this rung is located in the answer section. Verify that there are only 6 rungs left in the modified program.

7. Before you move on to the questions and exercises in Q-E Unit 16, you must clear the processor RAM memory. Enter mode 1 and follow the prompting messages on the pocket programmer. If you need help, follow the keystroke example in the table on Page 15-8. After the memory is cleared, you will have 885 words of memory available for your next exercise.

Q-E Unit 16 On-Line Data Control

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

On-Line Data Control 16-1
Forcing I/O Addresses 16-1
FRC ON, FRC OFF 16-1
TRUE-FALSE Status Indicator 16-2
Monitoring Data 16-4

SYMBOLOGY

Question Group 16A

Answers in Unit 18

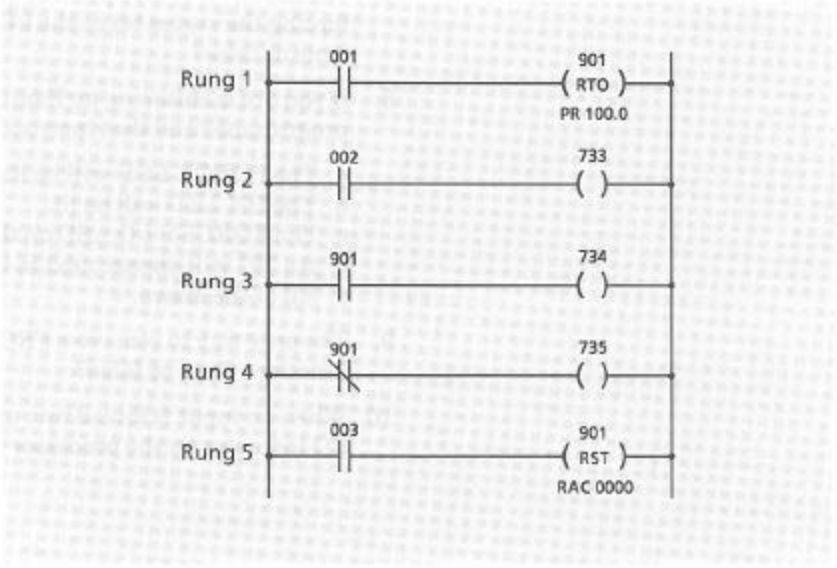
1. When the processor is in the Run mode, the pocket programmer can be used to change certain instructions in your program. True or false?
2. If left unprotected, timer and counter data can be changed in the Run mode with the pocket programmer. True or false?
3. Sequencer data may also be changed in the Run mode. True or false?
4. Timer, counter, and sequencer data may only be monitored in the Run mode. True or false?
5. By using the Force function, you can force an external I/O address into an ON or OFF state in the Run mode. True or false?
6. When you force an input device to an ON or OFF state, you are forcing the status bit at the I/O address and not the input terminal. True or false?
7. When you force an output device to an ON or OFF state, you are forcing the status bit at the I/O address and not the output terminal. True or false?
8. If you are monitoring an Examine ON instruction with the pocket programmer and that instruction is forced ON,
   a. The FORCE LED will be lit.
   b. The ON LED will be lit.
   c. The FORCED I/O LED on the processor will be lit.
   d. An "F" will appear next to the displayed instruction.
   e. All of the above.
9. The only way to remove a FRC ON function is with a FRC OFF function. True or false?
10. When an input is forced into an ON or OFF condition, the FORCED I/O indicator on the processor unit will be lit. True or false?

Question Group 16A
(continued)
Answers in Unit 18

11. When an input is forced ON, the program is executed as if the input device were actually ON. True or false?
12. A forced I/O address will not be retained if you enter the Program mode and make a change or if a processor fault is detected. True or false?
13. It is possible to remove all forces within your program without removing each forced instruction individually. True or false?
14. Forced I/O addresses may be removed individually by cursoring to the forced instruction and pressing REMOVE, ENTER. True or false?
15. The pocket programmer allows you to view timer, counter, and sequencer data in the Run and Test modes. True or false?
16. To view data associated with timer or counter instructions you must cursor to the instruction you wish to view. Once you are positioned on that instruction, pressing the NEXT key will display the PR value. Pressing the NEXT key again will display the AC value. Pressing the NEXT key one more time will display the PR value again. True or false?
17. If you are monitoring the AC value and you press the LAST key, the PR value will be displayed. True or false?
18. Timer and counter PR and AC values can only be changed in the Program mode. True or false?
19. If a PR value is protected, it cannot be unprotected in the Run mode. True or false?

Exercises

Program the following rungs and complete the exercises that follow. When programming the timer instruction, leave the PR value unprotected.



Exercises (continued)

1. Enter the Run mode and turn input 001 ON. Cursor to the timer instruction in rung 1 and observe the AC value as it increments. Refer to Figure 16.4 if you need help monitoring this timer data. As you cursor to the timer instruction, pressing the NEXT key will display the PR value. Pressing NEXT again will display the AC value.
2. While in the Run mode, change the PR value of RTO 901 to 0100. If you need help changing this PR value, refer to Figure 16.4.
3. The reset instruction should be changed to make the RAC value equal to 10 instead of 0. Refer to Figure 16.4 if you need help changing the RAC value.
4. Enter the Run mode and turn input 001 ON. Cursor to the RTO instruction and monitor the AC value as it increments. Next, cursor to the Examine On instruction at address 001 in that rung. Force input 001 to an OFF condition. If you need help implementing the FRC OFF condition, refer to Figure 16.3.

   When you have forced input 001 OFF, cursor to the RTO instruction and monitor the AC value. Verify that turning input 001 OFF and ON has no effect on the rung conditions. Remove the FRC OFF condition and verify that the rung will now function normally.
5. Cursor to the Examine ON instruction at address 003 in rung 5. Force input 003 to an ON condition. If you need help implementing the FRC ON function, refer to Figure 16.3.

   After you have forced input 003 ON, turn input device 003 ON and OFF and verify that the reset instruction remains TRUE even though the condition instruction that will enable the reset instruction is FALSE.

   Cursor to rung 1 and try to enable the RTO instruction by turning input 001 ON. Verify that the timer AC value is reset to the RAC value and will not increment even though the rung conditions are TRUE.
6. Remove all forces in this program before you move on to the next exercise.

Q-E Unit 17 Using the EEPROM Memory Module

Key Terms and Concepts

You should become thoroughly familiar with the following terms and concepts. Re-read the referenced pages if necessary.

EEPROM 17-1
Non-volatile 17-1
Store (Save) 17-2
Load (Read) 17-3
Auto-Load 17-4
Editing EEPROM Programs 17-5

SYMBOLOGY

Question Group 17A

Answers in Unit 18

1. The EEPROM memory module uses the same battery the processor uses to retain memory contents. True or false?
2. The EEPROM memory module has a memory capacity of 1200 words, so both SLC 100 and SLC 150 programs can be duplicated in the EEPROM module. True or false?
3. Before the EEPROM can be inserted into the processor module, AC line power to the processor must be removed. This helps to guard against possible damage to the EEPROM or unwanted CPU faults. True or false?
4. The EEPROM module must be used in conjunction with the pocket programmer or SLC computer software in order to load programs into the processor RAM. True or false?
5. The EEPROM is housed in a small plastic case with a large white write-on area to help you identify EEPROM contents. True or false?
6. To save or store the contents of the processor RAM in the EEPROM, you must use mode 6 (Save). True or false?
7. When you enter mode 6, prompting messages that appear on the pocket programmer display will guide you through the saving process. True or false?
8. In order to avoid accidental memory loss, the programmer will prompt you with the message **SurE ?** when you enter mode 6. True or false?
9. If a program is in the EEPROM module and you want to load a new program, you must erase the old program in the EEPROM module by entering mode 10. True or false?
10. If you clear memory (mode 1), and then enter mode 6 and go through the saving process, the contents of the empty RAM will be duplicated in the EEPROM module, leaving the EEPROM blank. True or false?
11. The programmable EEPROM auto-load instruction (864) can be used with the SLC 100 or SLC 150. True or false?

Question Group 17A
(continued)
Answers in Unit 18

12. The programmable EEPROM auto-load instruction can be used to clear a memory error and reload your user program into the processor RAM memory. True or false?
13. If you want to load the contents of the EEPROM into the processor RAM you could enter mode 7 (Read). True or false?
14. When the EEPROM contents are loaded to RAM memory, a program already in the processor RAM will be automatically erased. True or false?
15. When you enter mode 7, prompting messages on the pocket programmer will guide you through the copying process. True or false?
16. The auto-load procedure is a time saving convenience in applications where the same program is to be used in a number of processors or when quick program change-overs are necessary. True or false?
17. The program in the processor RAM will be erased when you follow the auto-load procedure. True or false?
18. The auto-load procedure is disabled if an access code is stored in the EEPROM. True or false?
19. The EEPROM modules are interchangeable with SLC 100 and SLC 150 programmable controllers. True or false?
20. SLC 100 programs can be easily run on the SLC 150 programmable controller. True or false?
21. All SLC 150 programs are compatible with SLC 100 controllers. True or false?

Unit 18 Answers

Q-E Unit 2

Group 2A:

1. False. 2. False. 3. True. 4. (d). 5. (d).

6. True. 7. True. 8. True. 9. False. 10. True.

11. True. 12. True. 13. True. 14. True. 15. True.

16. True. 17. True. 18. False. 19. False. 20. True.

21. True. 22. True. 23. True. 24. True. 25. True.

26. True.

27. False. A mounting plate is not needed for the EEPROM module. The EEPROM needs no battery.

28. True. 29. True. 30. True 31. True. 32. True.

33. True. 34. False. 35. False

Q-E Unit 3

Matching Exercise:

1. I. 2. G. 3. C. 4. B 5. F.

6. H. 7. D. 8. E. 9. A.

Group 3A:

1. (a). 2. (b).

3. False. Status bits will be ON or OFF, telling you whether the instruction is TRUE or FALSE.

4. False.

5. (c).

6. False. Addresses 701 thru 867 are for internal relay-type instructions.

7. (b). 8. (b). 9. False. 10 True. 11. True.

12. Program initialization. 13. False 14. True 15. True.

Q-E Unit 3 (continued)

Group 3B:

1. False. Instructions are classified as conditional and output instructions.

2. False. Branching instructions are classified as conditional.

3. (c). 4. True. 5. True. 6. (c). 7. True.

8. True. 9. (b). 10. (e) 11. True. 12. (A)

13. True

Group 3C:

1. 003. 2. Conditional. 3. Examine ON. 4. TRUE.

5. ON. 6. 011. 7. Output. 8. Output Energize.

9. A continuous path of TRUE condition instructions must exist. In this case, instructions 003 and 733 must be TRUE.

10. ON.

11. False. External I/O terminals have address numbers from 001 to 016, 101 to 116, 201 to 216, and 301 to 316.

12. Examine OFF. 13. OFF.

14. False. When the rung is TRUE, Examine ON instruction 003 is TRUE and its status bit is ON; Examine OFF instruction 733 is TRUE and its status bit is OFF; Output Energize instruction 011 is TRUE and its status bit is ON.

15. False. If the closed position of the switch is to energize coil M, the switch must be represented by an Examine ON instruction. This is true whether the switch is N.O. or N.C.

16. Address 103 for LS1. Address 111 for coil M.

Q-E Unit 3 (continued)

Group 3D:

1. I/O scan.

2. Program scan.

3. False. Output contact will go ON or OFF during I/O scan.

4. a) False. The time period is variable, as explained in d) and e).

   b) False. Time period can be longer than one cycle, as explained in d).

   c) False. After the input terminal goes ON (any point in the operating cycle), its corresponding status bit will not be set ON until the next I/O scan, *no matter where the rung is located in the program.* Similarly, the output terminal will not be set ON until the following I/O scan, *no matter where the rung is located in the program.*

   d) True. If the input terminal goes ON early in the program scan, it will take longer to reach the next I/O scan than if the input terminal went ON late in the program scan.

   Shortest time period: The input terminal goes ON just at the beginning of the I/O scan. The output terminal goes ON one operating cycle later.

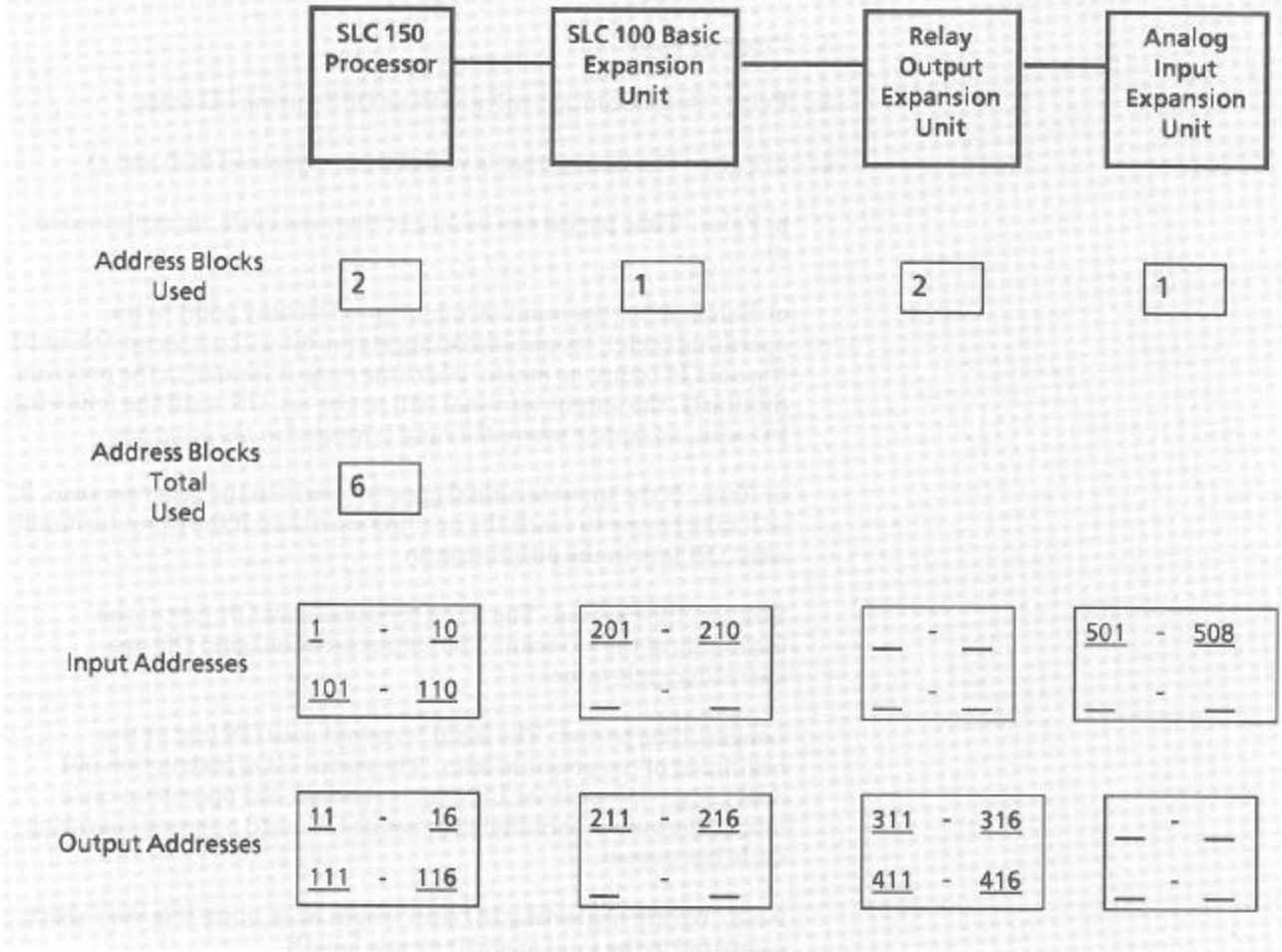
   Longest time period: The input terminal goes ON just at the beginning of the program scan. The status bit of instruction 001 does not go ON until the I/O scan is reached, one operating cycle later. The output terminal does not go ON until a second operating cycle has passed.

   Note: We are neglecting I/O scan length here, since it is very short compared to the total operating cycle length.

   e) True. Obviously, the more instructions in the program, the longer the program scan. This means that after the input terminal goes ON, it takes longer to get to the I/O scan (when the status bit of instruction 001 goes ON), and it takes longer to get to the next I/O scan (when the output terminal goes ON).

   Program content also has an effect on program scan length, since some instructions take a longer time to execute than others.

Q-E Unit 3 (continued) Group 3E:



Q-E Unit 4

Group 4A:

1. False. The pocket programmer can also be used for monitoring your program and troubleshooting.

2. True. 3. True. 4. True.

5. (c). 6. True. 7. (d).

8. False. There is a table fixed to the back of the pocket programmer with a description of the error codes.

9. (d).

10. False. The processor will function in all modes of operation.

11. True.

12. False. If you enter mode 9, the programmer will initiate a diagnostic test.

13. True. 14. True. 15. (d). 16. True. 17. True.

Group 4B:

1. False. The SLC 100 controller power supply will convert your incoming line power to the 5 VDC required.

2. False. The pocket programmer receives power from the processor unit by means of the interconnect cable.

3. True. 4. True. 5. True.

6. False. Connect the green wire to the terminal marked CHASSIS GRD.

7. True. 8. SurE? 9. True. 10. True. 11. True.

12. False. Only one output instruction per rung.

13. True. 14. True. 15. True. 16. True

17. True. 18. True. 19. True

Q-E Unit 4 (continued)

Group 4C:

1. (a). 2. True. 3. True. 4. No. 5. Yes.

6. True. 7. The Output Energize LED.

8. The Examine OFF LED and the ON LED.

9. The Examine OFF instruction is TRUE.

Group 4D:

27-5 ZCL Programming error.

27-2 Zero cross turn on failure

27-1 HSI module problem

Q-E Unit 5

Group 5A:

1. (c).

2. False. Relay type instructions are used internally also.

3. True. 4. True. 5. True. 6. True.

7. True. 8. True. 9. True. 10. False.

Group 5B:

1. True.

2. False. A continuous path of TRUE condition instructions does not exist. Input 001 *OR* inputs 002 *and* 003 must be ON for output 015 to be energized.

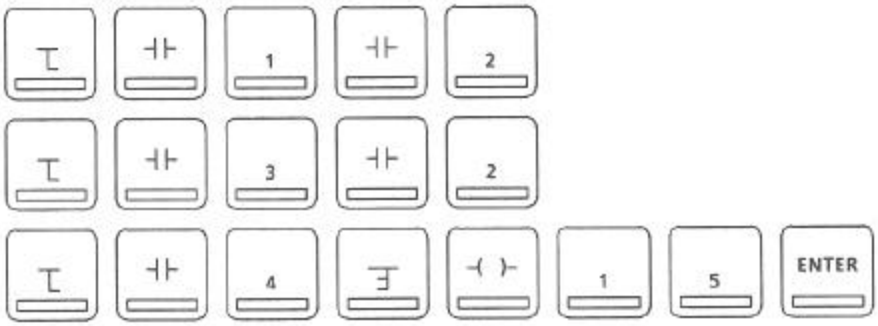
3. True. 4. True. 5. True.

6. False. Continuous path of TRUE instructions does exist.

Q-E Unit 5 (continued)

Group 5C:

1. (c). The keystroke sequence is as follows:



Group 5D:

1. True. 2. True. 3. False. 4. True. 5. True. 6. True.

Q-E Unit 6

Group 6A:

1. True. 2. True. 3. False. Timers count 0.1 second intervals.

4. True. 5. True. 6. True. 7. True.

8. False. The time delay is actually the PR value minus the RAC value.

9. (d). 500 x 0.1 = 50 seconds.

10. False. PR values can easily be changed in the Run or Program mode.

11. False.

12. False. AC values can be changed as easily as PR values.

13. True.

14. False. The address assignments from 951 to 982 are reserved for overflow status bits.

15. True. 16. True. 17. True. 18. True.

Q-E Unit 6 (continued)

Group 6B:

1. (b). The status bit does not go ON until the AC value reaches the PR value.

2. (a). 3. False. 4. (c). 5. (c). 6. (b).

7. (b). 8. (d). 925 + 50 = 975.

9. (d). The Examine ON instruction at address 951 examines the overflow status bit of timer 901 for an ON condition.

10. (a). 11. (d).

Group 6C:

1. True.

2. False. RTF timers start timing when their rung conditions change from TRUE to FALSE.

3. (b).

4. False. Rung 4 will be TRUE when the AC value exceeds 9999.

5. (a). 6. True. 7. (d). RTF timer status is retentive.

8. (c). 9. (c). 10. (d). 11. False.

12. True. Status bit 901 will be OFF when the AC value reaches 300.

13. (e). 14. True.

Group 6D:

1. (c). RTO 901 overflows from 999.9 to 1,000. RTO 902 times to an AC value of 850.0. (1000 + *850* = 1850 seconds.)

2. (d). Rungs 1 and 2 (or input -] [- 001) must be TRUE for 1,850 seconds before output 011 will be energized. After 1,850 seconds, output 011 will remain ON even if rungs 1 and 2 go FALSE. The only way to turn output 011 OFF is to reset.

3. (e). 4. (b).

5. False. Timing starts when rung conditions are TRUE.

Q-E Unit 7

Group 7A:

1. False. 2. True. 3. True.

4. False. AC value will increment when the rung makes the next transition to TRUE .

5. False. 6. True. 7. True. 8. True. 9. (a).

10. (b). 11. (b). 12. (b). 13. True. 14. True.

15. False. When the reset instruction is TRUE, the AC value is reset to the RAC value.

16. True.

17. False. Counter parameters can also be changed in the Run mode.

18. True.

Group 7B:

1. True. The address range for counters is from 901 thru 932.

2. (d). 928+50=978. 3. (b). 4. (a).

5. (b). Counter instructions are retentive.

6. (c). The counter is inactive while the reset instruction is TRUE.

7. True. An up-counter could be programmed to keep track of all parts made, and a down-counter could be programmed at the same address to subtract the bad parts from the total count.

Q-E Unit 7 (continued)

Group 7C:

1. (a). 2. (b). 3. (e). 4. (d). 5. (d).

6. (c). 7. (c).

8. (c). CTU has incremented 5 times but CTD has decremented 5 times, leaving the AC value at 0000.

9. (c).

10. True. Counters have the same address, therefore the same AC value.

11. True.

Group 7D:

1. True. 2. (c).

3. (c). Both of these conditions must be met. Counter 902 will not start counting until counter 901 has overflowed. When 901 overflows, the Examine instruction in rung 3 will reset counter 901 and it must reach the preset value again before the status bit is set ON again.

4. (d). 5. True. 6. (b).

Q-E Unit 8

Group 8A:

1. True.

2. False. You should select a time increment of 20, 40, or 80 milliseconds.

3. False. Although this method will work, adding the timing rungs to your program is faster and more accurate.

4. True.

5. False. You can always use address 869, since it is associated with your scan rate.

6. True. 7. True. 8. True 9. False 10. False 11. True

12. False.

Group 8B:

1. True. 2. (c). 3. (c).

Q-E Unit 9

Group 9A:

1. True. 2. True. 3. True. 4. True.

5. False. When the MCR start rung is FALSE, all non-retentive outputs within the zone are de-energized.

6. True.

7. False. A timer or counter functioning in an MCR zone with a FALSE start rung will have its status, underflow, and overflow bits held in their last state. The AC value will stop incrementing, but that value will be retained.

8. True. 9. True.

10. False. An MCR instruction is assigned no address at all.

11. False. The ON LED does not indicate when MCR or ZCL instructions are TRUE or FALSE.

Group 9B:

1. True. 2. True.

3. True. MCR start rung is now FALSE. All non-retentive outputs within the zone are de-energized. Outputs 011 and 012 are de-energized even though the condition instructions for their rungs are TRUE. These outputs are being controlled by the MCR zone.

4. True. All three conditional instructions in the MCR start rung must be TRUE for outputs in the zone to function normally.

5. True. 6. True.

Group 9C:

1. False. 2. True. 3. False. 4. True.

5. False. When the ZCL start rung is TRUE, all outputs within the zone function normally.

6. True. 7. False. 8. True. 9. True. 10. True. 11. True.

12. False. A hard-wired Master Control Relay must be used.

13. True. 14. False.

Q-E Unit 9 (continued)

Group 9D:

1. True. The two zones may not overlap. Overlapping zones may cause unexpected operation.

2. False.

3. False. The ZCL zone allows you to select which rung conditions you want to control output 012. If input 001 is ON, rung 3 controls output 012. If input 001 is OFF, rung 6 controls output 012.

4. (b).

5. False. The program would operate differently with MCR instructions. The lower MCR zone (rungs 5, 6, and 7) would always control output 012. Also, under the MCR zone, any outputs which were ON at the time the MCR took effect would be de-energized. In a ZCL zone, these outputs would remain in their last state.

Q-E Unit 10

Group 10A:

1. False. Both instructions are retentive.

2. True. 3. 901 thru 932.

4. Sequencers can have up to 100 steps and they can be time-driven or event-driven.

5. (b). 6. False. Time-driven sequencers count 0.1 second intervals.

7. True.

8. False. Event-driven sequencers move to the next step when their AC value matches their PR value.

9. True. 10. True. 11. True.

12. False. The SQI sequencer input-satisfied status bit has the same address as the SQI sequencer instruction.

13. False. The cycle completion bit is assigned the sequencer address plus 50.

Q-E Unit 10 (continued)

Group 10A:

14. True. 15. True. 16. (d). 17. True. 18. True.

19. False. Sequencer data is retentive. The sequencer will resume operation from the point where it left off.

20. True. 21. True. 22. False. 23. False

Group 10B:

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 901 ☒ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	18	17	16	15	14	13	12	11	Data B	Data A	
Mask Data →	0	0	1	1	1	1	1	1	3	F	
Step Data → 0	0	0	0	0	0	0	0	0	0	0	50
1	0	0	1	0	0	0	0	0	2	0	150
2	0	0	0	0	1	1	1	1	0	F	300
3	0	0	1	1	1	1	0	0	3	C	600

Group 10C:

1. (b). Mask data is 3F. Bit addresses 011 thru 016 are controlled by this sequencer.

2. (a). 3. True. 4. True. 5. (b). 6. (b).

7. (c). 8. (b), (d), and (e). 9. (b) and (c).

10. True. 11. False.

12. False. The reset instruction will reset the sequencer to the programmed RAC step value.

13. Bit address 703 will be ON for one program scan each time SQO completes a step.

14. True. 15. True. 16. True.

Q-E Unit 10 (continued)

Group 10D:

1. True. 2. True. 3. (e). 4. (c).

Group 10E:

1. (c). Inputs 001 and 008 are not examined by this sequencer.

2. (d). 3. (b). 4. (d). 5. True.

6. True. The input-satisfied status bit will not go ON for that step.

7. True. 8. True.

9. (b). The preset value for step 4 is 2 and so two FALSE-to-TRUE transitions of the rung must occur before the sequencer will advance to the next step.

10. (d).

Q-E Unit 11

Group 11A:

1. True. 2. True. 3. True. 4. True. 5. True. 6. 58.

7. True. 8. True 9. False. 10. True. 11. True. 12. True

Group 11B:

1. True. 2. True. 3. False. 4. False. 5. False. 6. False.

Group 11C:

CLASSIFICATION: ☐ –(SQI)– ☒ –(SQO)– ADDRESS: 905 ☒ TIME DRIVEN ☐ EVENT DRIVEN GROUP NUMBER: 42

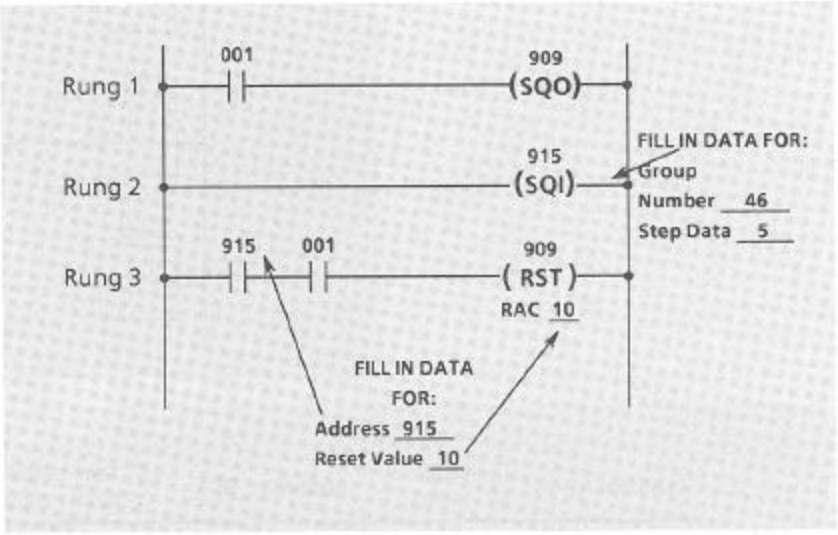
BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →									Data B	Data A	
Mask Data →									F	F	
Step Data → 0									0	5	0.5
1									0	6	0.5
2									0	0	0.5
3									0	1	0.5
4									0	2	0.5
5									0	3	0.5
6									0	4	0.5

Q-E Unit 11 (continued)

Group 11D:

1. True. 2. Group number 62 and step number 5. 3. (d) 4. (a)

Group 11E:



Q-E Unit 12

Group 12A:

1. (c). 2. (c). 3. (d). 4. (c).

5. (b). 6. (e). 7. (a). 8. (b).

Group 12B:

1. True.

2. False. You should use the external outputs on the processor.

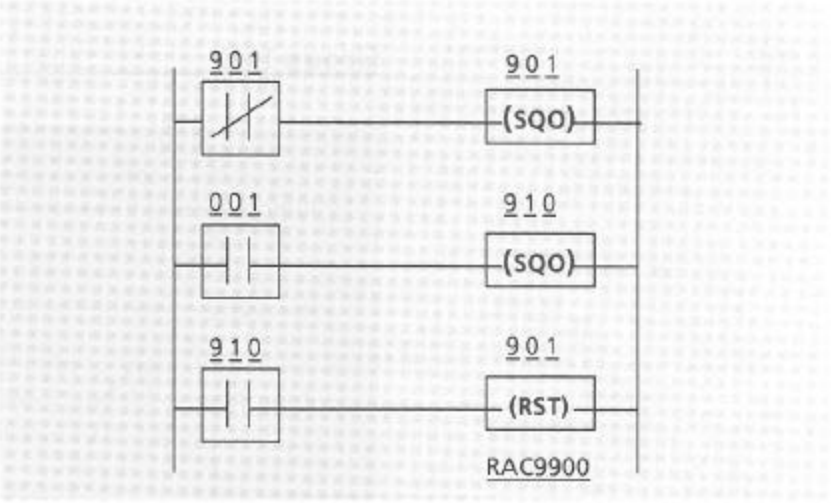
3. False. 4. Yes. 5. True.

6. True. Refer to the Maximum Pulse Rate table on page 12-2.

7. True.

Q-E Unit 12 (continued)

Group 12C:



CLASSIFICATION: ☐ -(SQI)- ☒ -(SQO)- ADDRESS: 901 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 1

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	118	117	116	115	114	113	112	111	Data B	Data A	
Mask Data →	0	0	0	0	0	0	1	1	0	3	
Step Data → 0	0	0	0	0	0	0	0	1	0	1	0 0 2 4
1	0	0	0	0	0	0	1	0	0	2	9 9 9 9
2											
3											

Type: HSI Sequencer

Use this form for your final answer: nswer.

CLASSIFICATION: ☐ -(SQI)- ☒ -(SQO)- ADDRESS: 910 ☐ TIME DRIVEN ☒ EVENT DRIVEN GROUP NUMBER: 0

BIT ADDRESS DATA									PROGRAM CODE		PRESET VALUES
	B				A						
Bit Addresses →	018	017	016	015	014	013	012	011	Data B	Data A	
Mask Data →	0	0	0	0	0	0	0	0	0	0	
Step Data → 0	0	0	0	0	0	0	0	0	0	0	0 0 0 1
1											
2											
3											

Type: Single Shot Reset Sequencer

Q-E Unit 13

Group 13A:

1. True. 2. False. 3. True. 4. (a). 5. (b).

6. True. 7. True. 8. True. 9. True. 10. (c).

Group 13B:

1. True. 2. True. 3. (d). 4. False. 5. False

Group 13C:

1. True. 2. True. 3. False.

Q-E Unit 14

Group 14A:

1. False. 2. False. 3. True. 4. True. 5. True. 6. True.

7. True. 8. True. 9. True. 10. False. 11. True. 12. (d).

13. True 14. False (864)

Group 14B:

1. True. Address 868 is the program initialization bit. This bit will be ON during the first program scan for power-up

2. (d). 3. (d).

4. (c). RTO timer status is retentive, but the program initialization bit is always ON for the first program scan on power-up. When this bit is ON, the Examine ON instruction at address 868 in rung 3 will be TRUE, enabling the reset instruction that resets the timer to 0.

5. (c). Since the AC value has not reached the PR value, status bit 901 has not been set ON. When input 001 is turned OFF, Examine OFF instruction 001 in rung 3 will reset timer 901 to the RAC value.

6. Non-retentive

Q-E Unit 14 (continued)

Group 14C:

1. (b).

2. (c). The counter status is retentive. However, the Examine ON instruction at address 868 is the program initialization instruction, which is TRUE for the first program scan and will reset the counters automatically upon power-up.

3. (c). 4. (a). 5. (c) and (d).

6. (e). Even though the current AC value of 0002 does not indicate an underflow condition, output 012 was turned ON when the AC value underflowed to 9999 (0002 minus 0003). Once set ON, overflow/ underflow bits can only be turned OFF with the reset instruction..

Q-E Unit 15

Group 15A:

1. True. 2. True. 3. (a), (b), and (d).

4. True. 5. True. 6. True.

7. False. The cursor will move to the output instruction in the rung directly above the last rung of your program.

8. True. 9. True. 10. True. 11. True. 12. True.

13. True. 14. True. 15. True. 16. True.

17. False. All conditional instructions must also be removed.

18. True.

19. False. Mode 1 has no effect on the EEPROM memory module.

20. True.

21. True. However sequencer presets can be changed in the Run, Test, or Program modes.

22. True. 23. True.

Q-E Unit 15 (continued)

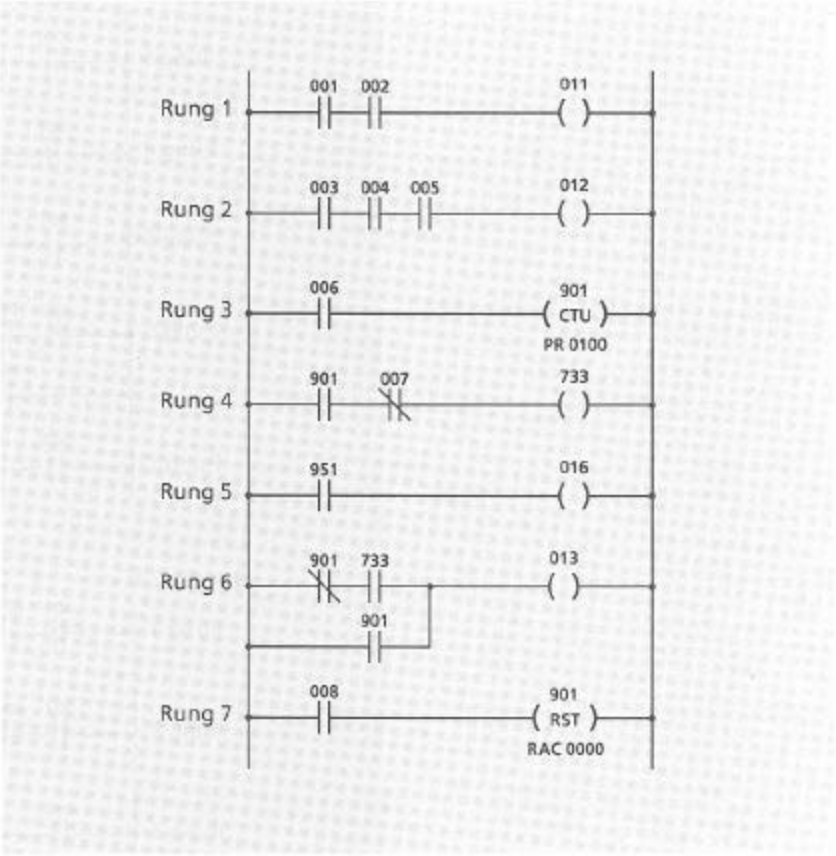
Group 15A:

24. False. Cursor to the value you want to change, enter the new value and press ENTER.

25. False. Changes are immediately entered into RAM memory.

Group 15B:

1. See diagram below.

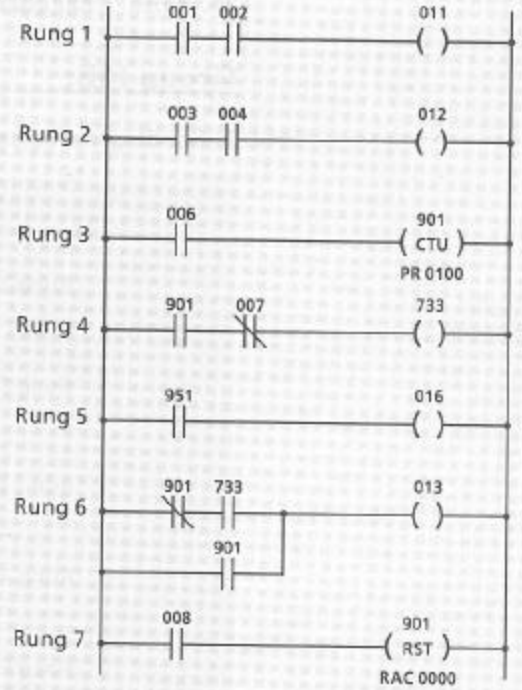


Q-E Unit 15 (continued)

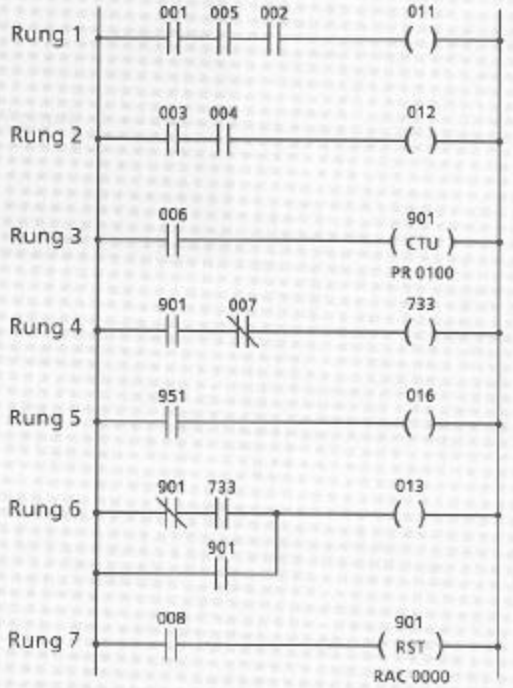
Group 15B:

2 and 3. See diagrams below.

Question 2:



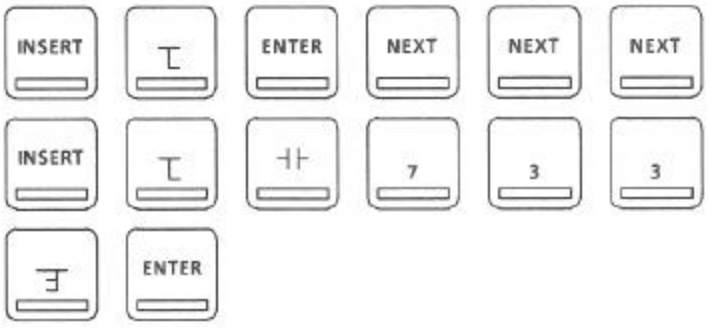
Question 3:



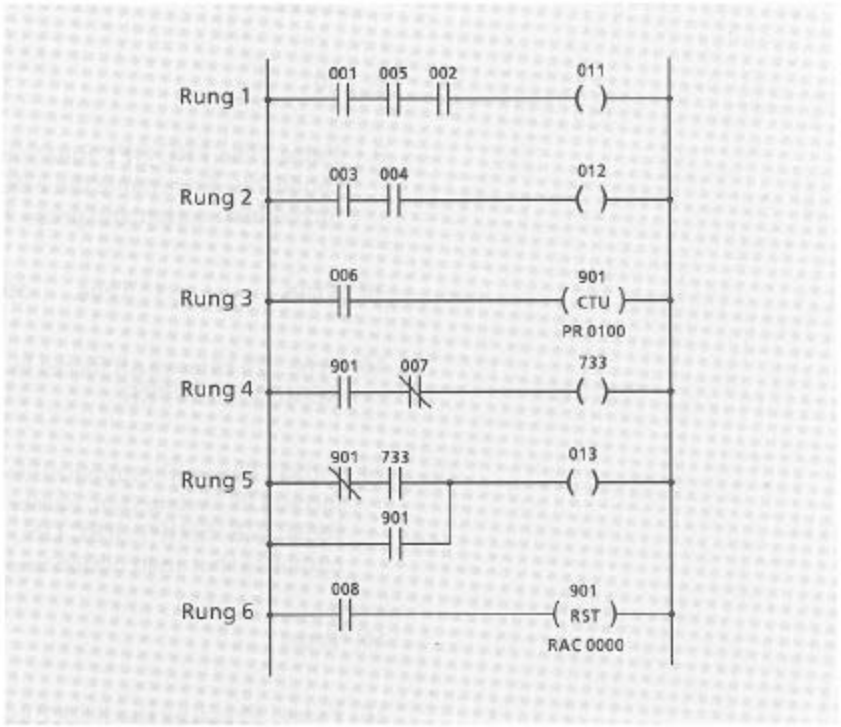
Q-E Unit 15 (continued)

Group 15B:

4. Correct keystroke sequence for adding the branch: Cursor to Examine ON instruction at address 001 in rung 1 and press



6. See diagram below.



Correct keystroke sequence for removing the rung in exercise 6: Cursor to any instruction in rung 5 and press

The entire rung will be removed from memory.

Q-E Unit 16 Group 16A:

1. True. Timer and counter parameters can be changed in the Run mode.

2. True.

3. False. Sequencer data may only be changed in the Program mode. Sequencer Preset values may be changed in the Run mode.

4. False. Data may also be monitored in the Test modes.

5. True. 6. True.

7. False. The status bit is not affected. Only the output terminal is forced.

8. (e).

9. False. The FRC OFF function will force the instruction to an OFF condition. To remove the FRC ON condition you must cursor to the forced instruction and press REMOVE, SHIFT, FRC ON.

10. True. 11. True. 12. True.

13. True. To remove all forces in your program you must press REMOVE, ENTER.

14. False. This will remove all forces in your program. To remove a forced ON instruction, cursor to that instruction and press REMOVE, SHIFT, FRC ON. To remove a forced OFF instruction, cursor to that instruction and press REMOVE, SHIFT, FRC OFF.

15. True.

16. False. Pressing the NEXT key one more time will take you to the next rung.

17. True.

18. False. Timer and counter PR and AC values can be changed in the Run mode also.

19. True. However you can switch to the Program mode and unprotect.

Q-E Unit 17 Group 17A:

1. False. The EEPROM requires no battery to retain memory contents.

2. True. 3. True.

4. False. The Auto-load procedure can be used. See Page 16-3 of the User's Manual for details.

5. True. 6. True. 7. True. 8. True.

9. False. The old program will be automatically erased when you load a new program.

10. True. 11. False, only SLC 150 12. True. 13. True.

14. True. 15. True. 16. True.

17. True.

18. True

19. True.

20. True. However caution should be used when using addresses that have unique functions with the SLC 150.

21. False. SLC 150 special instruction addresses and HSI counter instructions are not compatible with the SLC 100.

ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

Industrial Control Group
Milwaukee, Wisconsin 53204

Publication 1745-800A – March, 1988
Supersedes Publication 1745-800A Dated November, 1987

40065-237-01(D)