Shark ™
XL II Series
Programmable Controller

Industrial
CONTROLS

Instruction Manual J-3809-2

RELIANCE
ELECTRIC

The information in this user's manual is subject to change without notice.

DANGER

ONLY QUALIFIED ELECTRICAL PERSONNEL FAMILIAR WITH THE CONSTRUCTION AND OPERATION OF THIS EQUIPMENT AND THE HAZARDS INVOLVED SHOULD INSTALL, ADJUST, OPERATE, OR SERVICE THIS EQUIPMENT. READ AND UNDERSTAND THIS MANUAL IN ITS ENTIRETY BEFORE PROCEEDING. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

DANGER

THE USER IS RESPONSIBLE FOR CONFORMING TO THE NATIONAL ELECTRIC CODE AND ALL OTHER APPLICABLE LOCAL CODES. WIRING, GROUNDING, DISCONNECTS, AND OVERCURRENT PROTECTION ARE PARTICULARLY IMPORTANT. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

DANGER

CHECK FOR THE PRESENCE OF VOLTAGE BEFORE YOU PERFORM ANY INSTALLATION OR MAINTENANCE OPERATIONS. THE I/O DEVICES CONNECTED TO THE FIELD WIRING MAY BE POWERED FROM AN EXTERNAL POWER SOURCE. DO NOT ASSUME THAT IT IS SAFE TO WORK ON THE MODULES AND WIRING WHEN JUST THE MAIN SOURCE OF POWER IS DISCONNECTED. DISCONNECT ALL SOURCES OF POWER BEFORE SERVICING THE EQUIPMENT. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

WARNING

THE USER MUST PROVIDE AN EXTERNAL, HARDWIRED EMERGENCY STOP CIRCUIT THAT WILL DISABLE THE SYSTEM IN CASE OF IMPROPER OPERATION. UNCONTROLLED MACHINE OPERATION MAY RESULT IF THIS IS NOT DONE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

Table of Contents

Appendices

List of Figures

List of Tables

1.0 INTRODUCTION

The products described in this manual are manufactured and/or distributed by Reliance Electric Industrial Company.

This manual describes the Shark XL II Series Programmable Controller. This manual tells you how to install, operate, and maintain your Shark XL II system.

For information on the Shark I/O system when it is used as remote I/O on an AutoMax Remote I/O network, refer to the Shark I/O System manual (J2-3018)

1.1 Manual Contents

- Chapter 1 - Introduction
  Summarizes the manual's contents and related publications.
- Chapter 2 - Shark XL II System Overview
  Describes the components that make up a Shark XL II system.
- Chapter 3 - Installation
  Describes how to install a rack, modules, and associated wiring.
- Chapter 4 - General Programming
  Explains ladder logic, I/O processing, and I/O addressing. Also describes bit, word, and byte processing.
- Chapter 5 - Operating Procedures
  Explains how to create, search, edit, and delete a program. Also explains how to force and monitor I/O.
- Chapter 6 - Instruction Set and Commands
  Lists the Shark XL II's instructions/commands and provides programming examples.
- Chapter 7 - Diagnostics and Troubleshooting
  Summarizes the Shark XL II's error codes and messages.
- Appendix A - Specifications
  Provides specifications on the Shark XL II system and modules.
- Appendix B - Instruction Set/Command Summary
  Summarizes the Shark XL II instructions and commands.
- Appendix C - Execution Times
  Lists the instruction/command execution times.
- Appendix D - Command Cross-Reference
  Provides a cross-reference between the Shark XL II commands and the function (FUN) numbers.

1.2 Related Publications

You must be familiar with all the instruction manuals that describe your system configuration. They may include, but are not limited to, the following:

- J-3800 Shark Universal Programmer Instruction Manual
- J-3801 Shark X Series Programmable Controller Instruction Manual
- J-3803 Shark Standard Programmer Instruction Manual
- J-3804 ReSource Shark Programming Executive
- J-3805 Shark Remote Master Module and Remote Target Module Instruction Manual
- J-3807 Shark Network Module Instruction Manual
- Your IBM-compatible personal computer and DOS operating system manuals.
- IEEE 518 Guide for the Installation of Electrical Equipment to Minimize Electrical Noise Inputs to the Controllers.

2.0 SHARK XL II SYSTEM OVERVIEW

This chapter identifies and summarizes the components that make up a Shark XL II programmable controller system.

2.1 System

The Shark XL II is a rack-based programmable controller. Rack sizes range from a three-slot unit to a seven-slot unit. The number of slots in a rack indicates the number of modules that can be plugged into the rack (exclusive of the power supply). For example, a three-slot rack can hold a processor module, two I/O modules, and a power supply module. Expansion racks can be added so that a single processor module can control up to ten I/O modules (slots). Using remote I/O communication, a Shark XL II processor can control up to 17 slots.

Each rack must contain a power supply module (either 115/230 VAC or 24 VDC). The power supply module must be located in the left-most slot in the rack. All main racks must contain a processor module in the slot immediately to the right of the power supply. Expansion and remote racks do not require a processor module. The slot normally occupied by the processor module can be used to hold an additional I/O module.

The Shark XL II system offers a number of modules that can accomodate a wide range of I/O applications:

- 115/230 VAC Inputs
- 24 VDC Inputs
- 115/230 VAC Outputs
- 5-27 VDC Outputs
- Relay Outputs
- Isolated High/Low Power Relay Outputs
- TTL Inputs/Outputs
- 4-20 mA Analog Inputs
- 0-10 VDC Analog Inputs
- 4-20 mA Analog Outputs
- 0-10 VDC Analog Outputs
- High Speed Counter
- Remote I/O Communication
- Network Communication

The Shark XL II can be programmed using either a:

- Standard Programmer
- Universal Programmer
- IBM-compatible personal computer with the SPX software.

The programmers connect to the processor through a 16-pin ribbon cable. Universal programmers have an RS-232 port that allows a printer to be attached for making program printouts. Epson printers are recommended.

The Shark XL II processor has an on-board serial port for connecting an IBM-compatible personal computer when using the SPX programming software.

Application programs can be stored either on floppy disks through the Shark Programming Executive (SPX) software (run on an IBM-compatible personal computer), on cassette tape, or in a memory pack. Tape recorder terminals are found on both programmers.

2.1.1 Racks

Five racks are available:

- M/N 45C910 Three-slot Rack
- M/N 45C911 Four-slot Rack
- M/N 45C912 Five-slot Rack
- M/N 45C913 Six-slot Rack
- M/N 45C914 Seven-slot Rack
- M/N 45C918 Nine-slot Rack

Figure 2.1 illustrates a typical rack. The number of slots that a rack has indicates the number of usable slots in addition to the slot occupied by the power supply module. For example, a three-slot rack has a slot for the power supply module and three additional slots for the processor and I/O modules. Rack mounting dimensions are given in chapter 3.

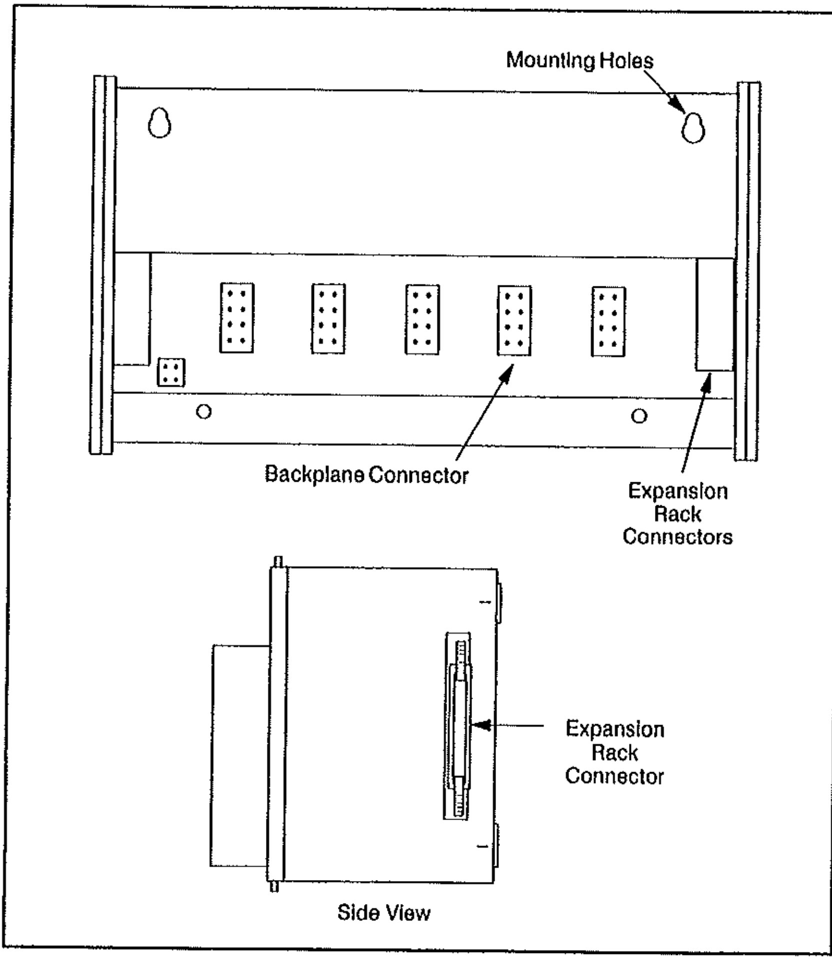


Figure 2.1 - Typical Rack

2.1.2 Power Supply Modules

Three power supply modules are available:

- M/N 45C920 115/230 VAC
- M/N 45C921 24 VDC
- M/N 45C922 115/230 VAC

Each rack requires a power supply module. As shown in figure 2.2, terminals are provided for connecting the incoming power wires. M/Ns 45C920 and 45C922 also have 24 VDC terminals for powering output circuitry. Power supply specifications are listed in Appendix A.

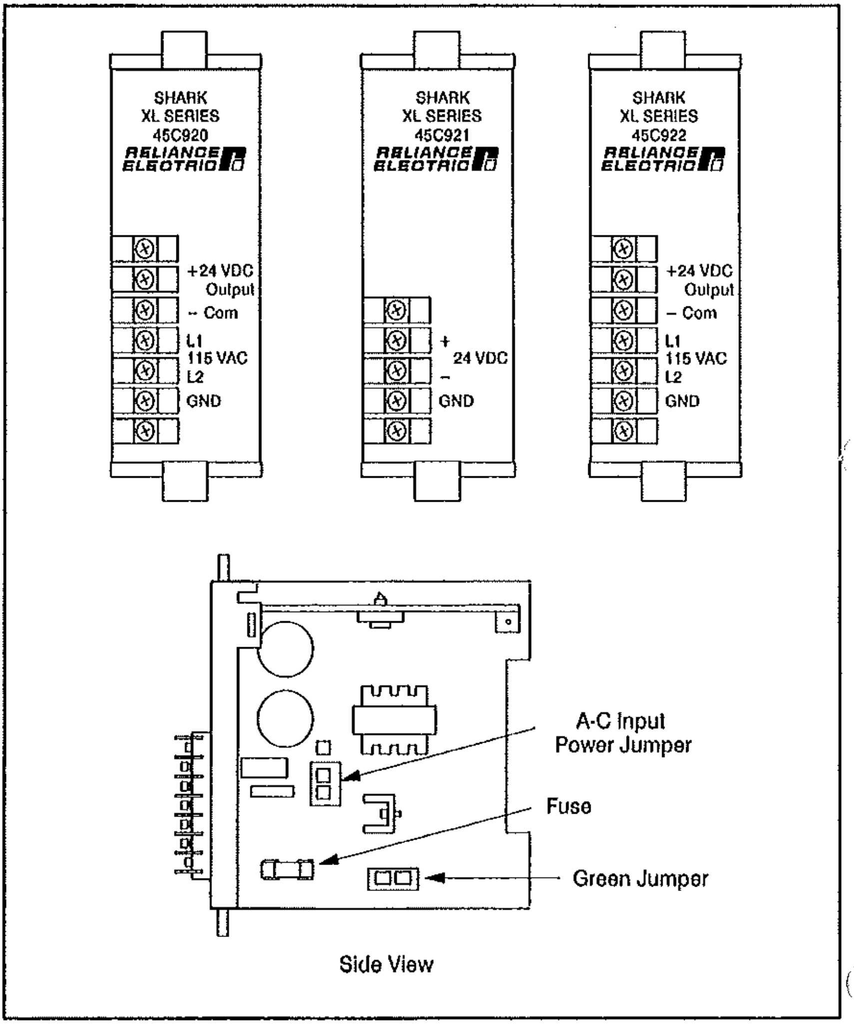


Figure 2.2 - Power Supply Modules

2.1.3 Processor Module

The processor module and EEPROM/EPROM memories are sold separately:

- M/N 45C901 Processor Module without Memory
- M/N 45C975 EEPROM Memory, 926 (1K) Words
- M/N 45C977 EEPROM Memory, 3997 (4K) Words
- M/N 45C978 EPROM Memory, 3997 (4K) Words

As shown in figure 2.3, the module's faceplate has three LEDs, a sixteen-pin connector for a hand-held programmer, an RS-232 port for connecting an IBM-compatible personal computer, and a communication port selector switch.

When the PWR LED is lit, incoming power is applied to the power supply module and the power supply is providing power to the backplane. When the RUN LED is lit, the processor is executing the application program. When the ERR LED is lit, a system error has been detected.

The PGM position of the communication selector switch selects the hand-held programmer (through the sixteen-pin connector) as the processor's programming device. The COM1 position enables the RS-232 port using the same protocol as the hand-hand programmers. The COM2 position enables the RS-232 port using the enhanced mode 2 protocol. This protocol provides better response with the connected devices. Select the COM2 position when using the Shark Programming Executive (SPX). Refer to the Shark Programming Executive Instruction Manual, J-3804, for additional information. The position of the communication selector switch is read upon power-up. Whenever you change the setting of the selector switch, you must cycle power to avoid an error condition. General processor specifications and RS-232 pinouts are given in Appendix A. An instruction set/command summary is given in Appendix B.

In addition to the XL II processor, a standard processor (Shark XL) is also available. Refer to the Shark XL Series Programmable Controller Instruction Manual (J-3802) for more information. Module specifications are in Appendix A.

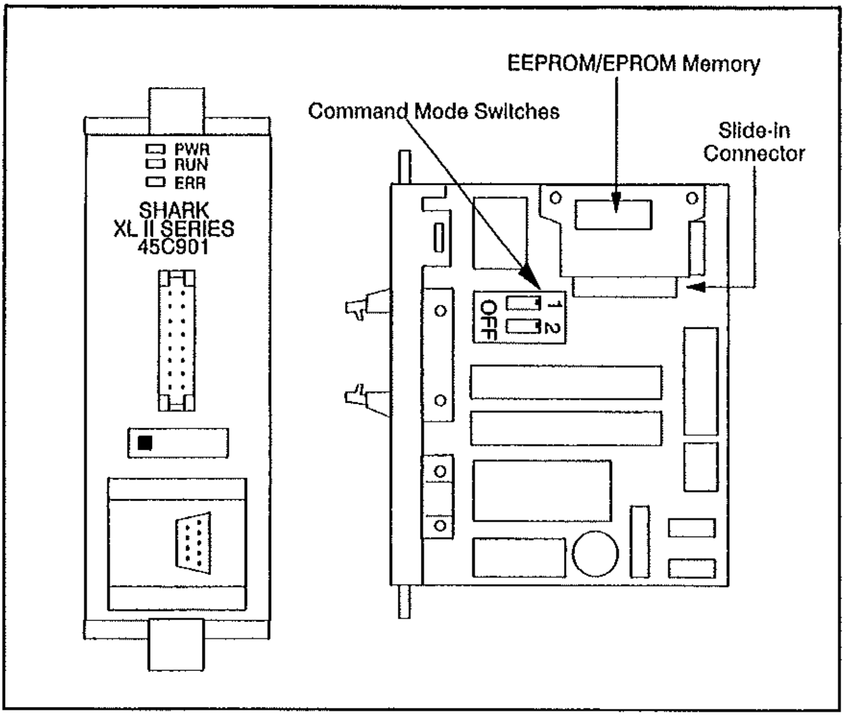


Figure 2.3 - Processor Module

2.1.4 Digital Input Modules

Six digital input modules are available:

- M/N 45C940 8-Channel 115/230 VAC Input
- M/N 45C941 16-Channel 115/230 VAC Input
- M/N 45C942 16-Channel 115/230 VAC Input with Removable Terminal Block
- M/N 45C944 8-Channel 24 VDC Input
- M/N 45C945 16-Channel 24 VDC Input
- M/N 45C946 16-Channel 24 VDC Input with Removable Terminal Block

As shown in figure 2.4, the input modules have LED status indicators (0-7) or (0-15) and terminals for connecting the external input devices. The C terminal is the common. On the 16-channel modules, terminals 0 to 7 use common C1 while terminals 8 to 15 use common C2. Terminals C1 and C2 are internally connected. When a status LED is lit, it indicates that the input device connected to the corresponding terminal is on. Module specifications are given in Appendix A.

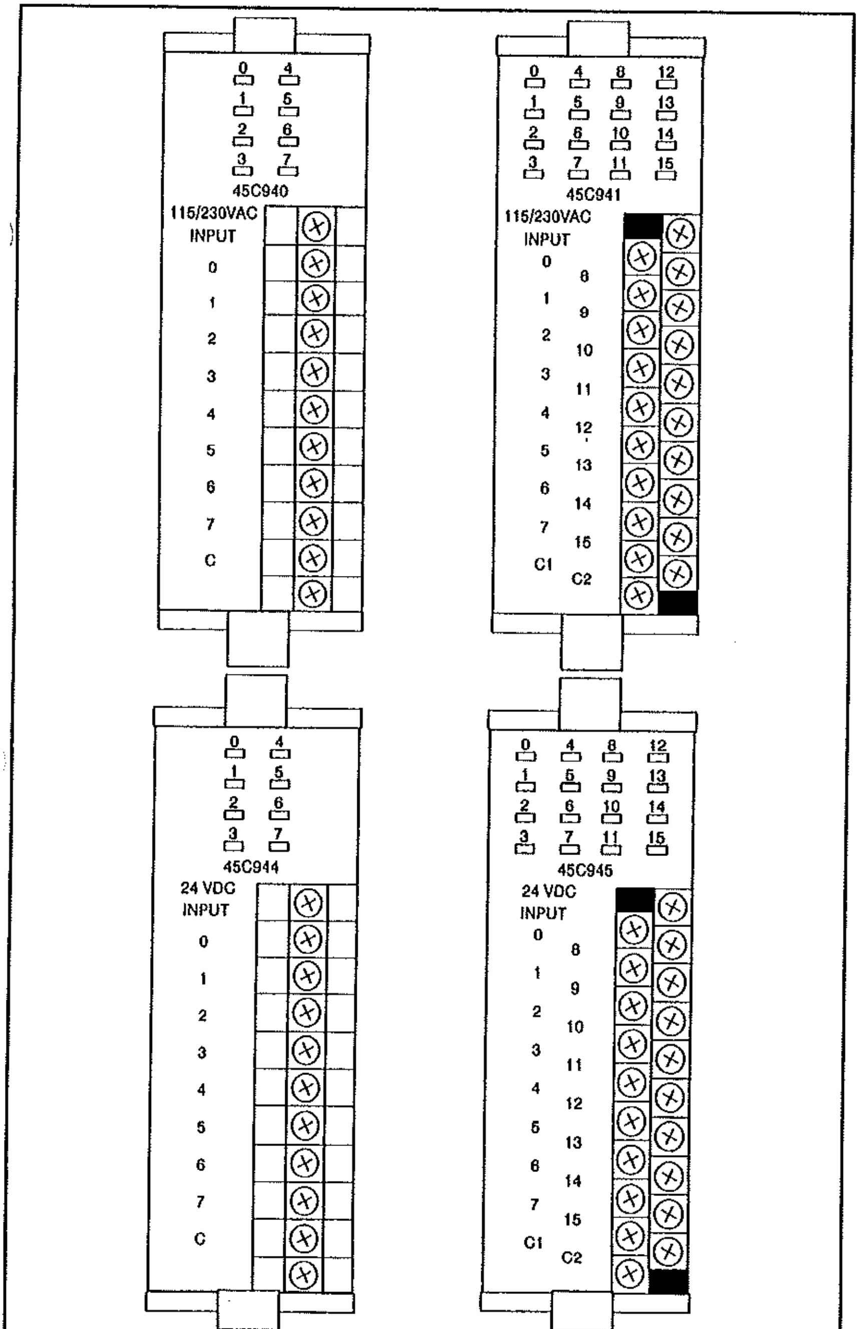


Figure 2.4 - Digital Input Modules

2.1.5 Digital Output Modules

Eleven digital output modules are available:

- M/N 45C959 16-Channel Relay Output with Removable Terminal Block
- M/N 45C960 8-Channel 115/230 VAC Output
- M/N 45C961 16-Channel 115/230 VAC Output
- M/N 45C962 8-Channel 5-27 VDC Output
- M/N 45C963 16-Channel 5-27 VDC Output
- M/N 45C964 16-Channel 24 VDC Output with Removable Terminal Block
- M/N 45C965 16-Channel 115/230 VAC Output with Removable Terminal Block
- M/N 45C966 8-Channel Relay Output
- M/N 45C967 16-Channel Relay Output
- M/N 45C968 8-Channel Isolated High Power Relay Output
- M/N 45C969 8-Channel Isolated Low Power Relay Output

As shown in figure 2.5, the output modules have LED status indicators (0-7) or (0-15) and terminals for connecting the external output devices. The C terminal is the common. On the 16-channel modules, terminals 0 to 7 use common C1 while terminals 8 to 15 use common C2. Terminals C1 and C2 are not internally connected. Terminals 0_a to 7_a are paired with terminals 0_b to 7_b on the 8-Channel Isolated Relay Output modules. When a status LED is lit, it indicates that the output device connected to the corresponding terminal has been instructed to turn on. Module specifications are given in Appendix A.

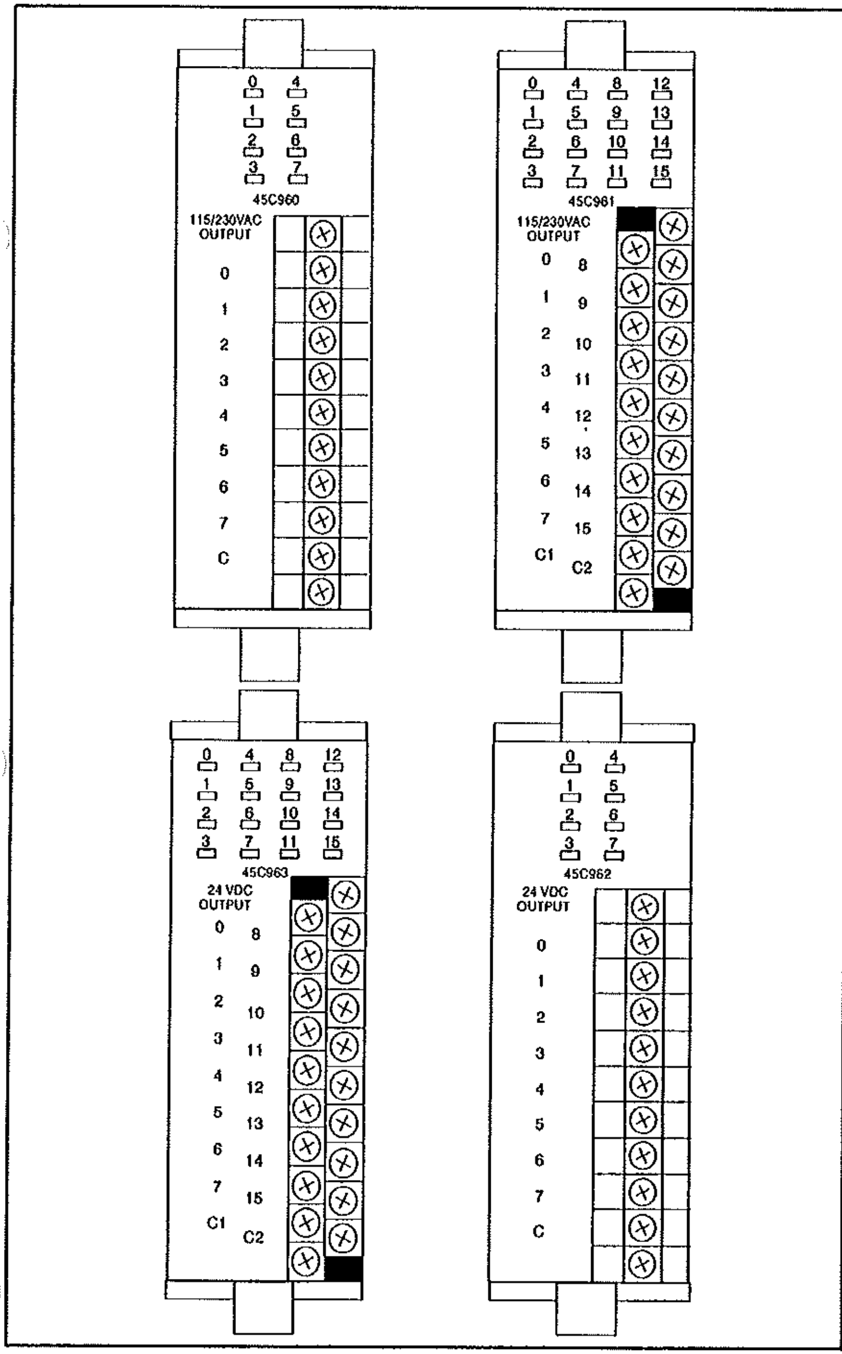


Figure 2.5 - Digital Output Modules

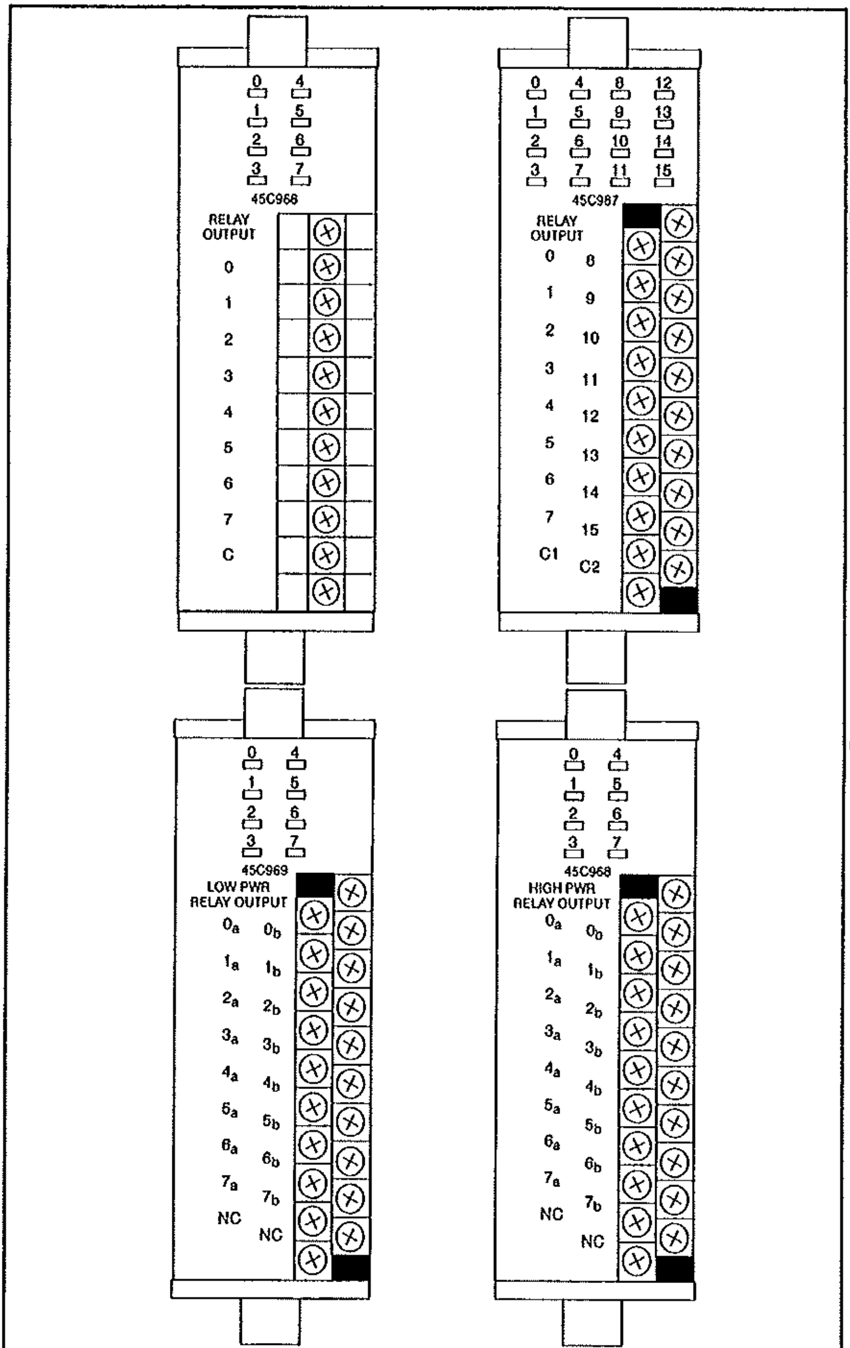


Figure 2.5 - Digital Output Modules (Continued)

2.1.6 Input/Output Modules

Two input/output modules are available:

- M/N 45C957 16-Channel TTL Input/Output
- M/N 45C958 8-Channel 24 VDC Input/Output

As shown in figure 2.6, the TTL I/O module uses the odd-numbered terminals for the input devices and the even-numbered terminals for the external output devices.

Terminal 1 is the common and terminal 3 is the positive terminal for the input devices. Terminals 21 and 23 have no connection. The odd-numbered terminals between 5 and 39 are for the actual input devices.

Terminal 2 is the common and terminal 4 is the positive terminal for the output devices that connect to the even-numbered terminals from 6 to 20. Terminal 22 is the common and terminal 24 is the positive terminal for the output devices that connect to the even-numbered terminals from 26 to 40.

Also as shown in figure 2.6, the 24 VDC I/O module uses terminals 0 to 7 for the input devices with terminal CI as the common. Terminals 8 to 15 are for the output devices with terminal C2 as the common.

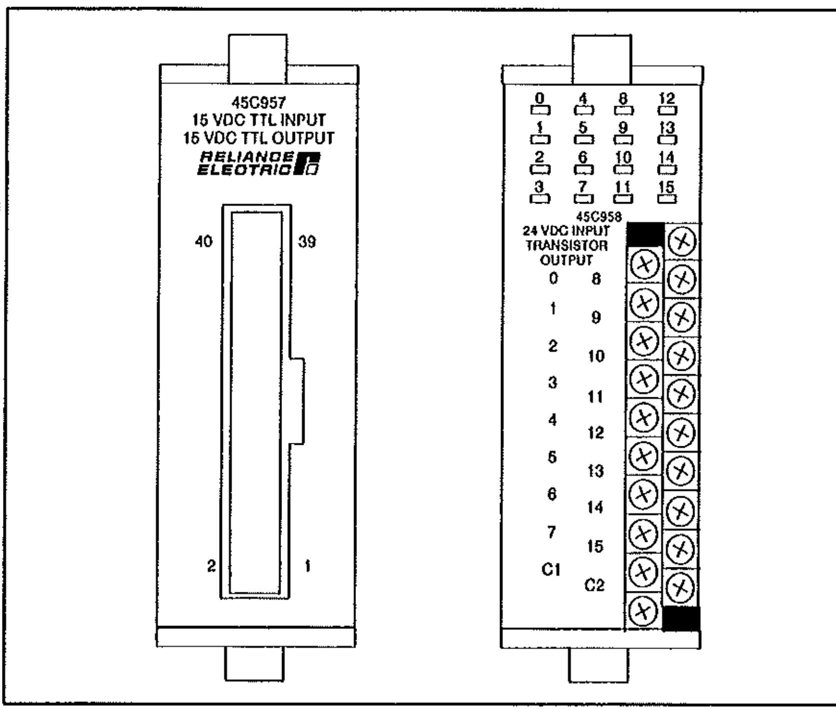


Figure 2.6 - Input/Output Modules

2.1.7 Analog Input Modules

Two analog input modules are available:

- M/N 45C990 8-Channel 4-20 mA Input
- M/N 45C992 8-Channel 0-10 VDC Input

As shown in figure 2.7, the input modules have terminals (0-7) for connecting the analog input devices. The C terminal is the common. Module specifications are given in Appendix A.

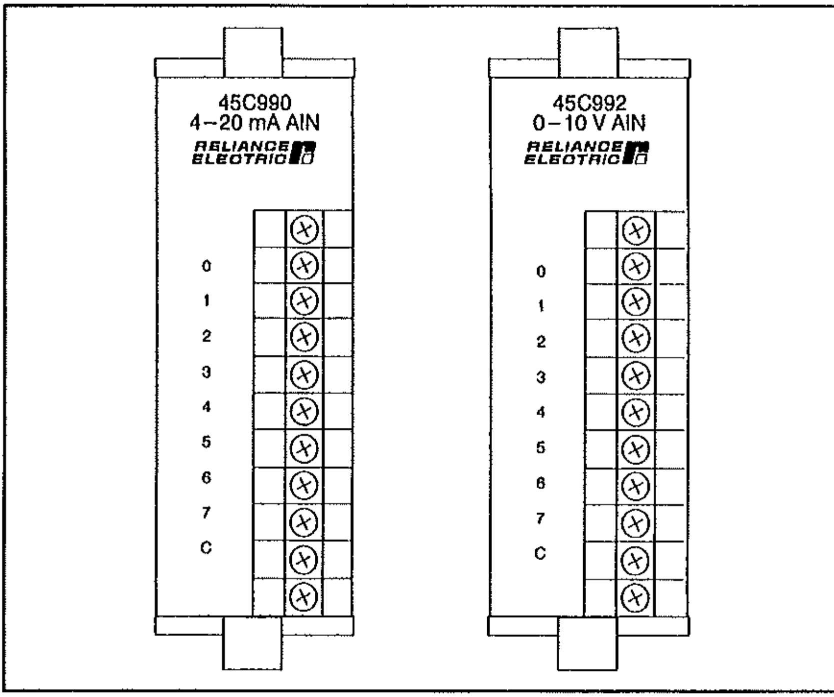


Figure 2.7 - Analog Input Modules

2.1.8 Analog Output Modules

Four analog output modules are available:

- M/N 45C993 2-Channel 4-20 mA Output
- M/N 45C994 2-Channel 0-10 VDC Output
- M/N 45C995 4-Channel 4-20 mA Output
- M/N 45C997 4-Channel 0-10 VDC Output

As shown in figure 2.8, the output modules have terminals for connecting the analog output devices. The C terminals are for the commons. Module specifications are given in Appendix A.

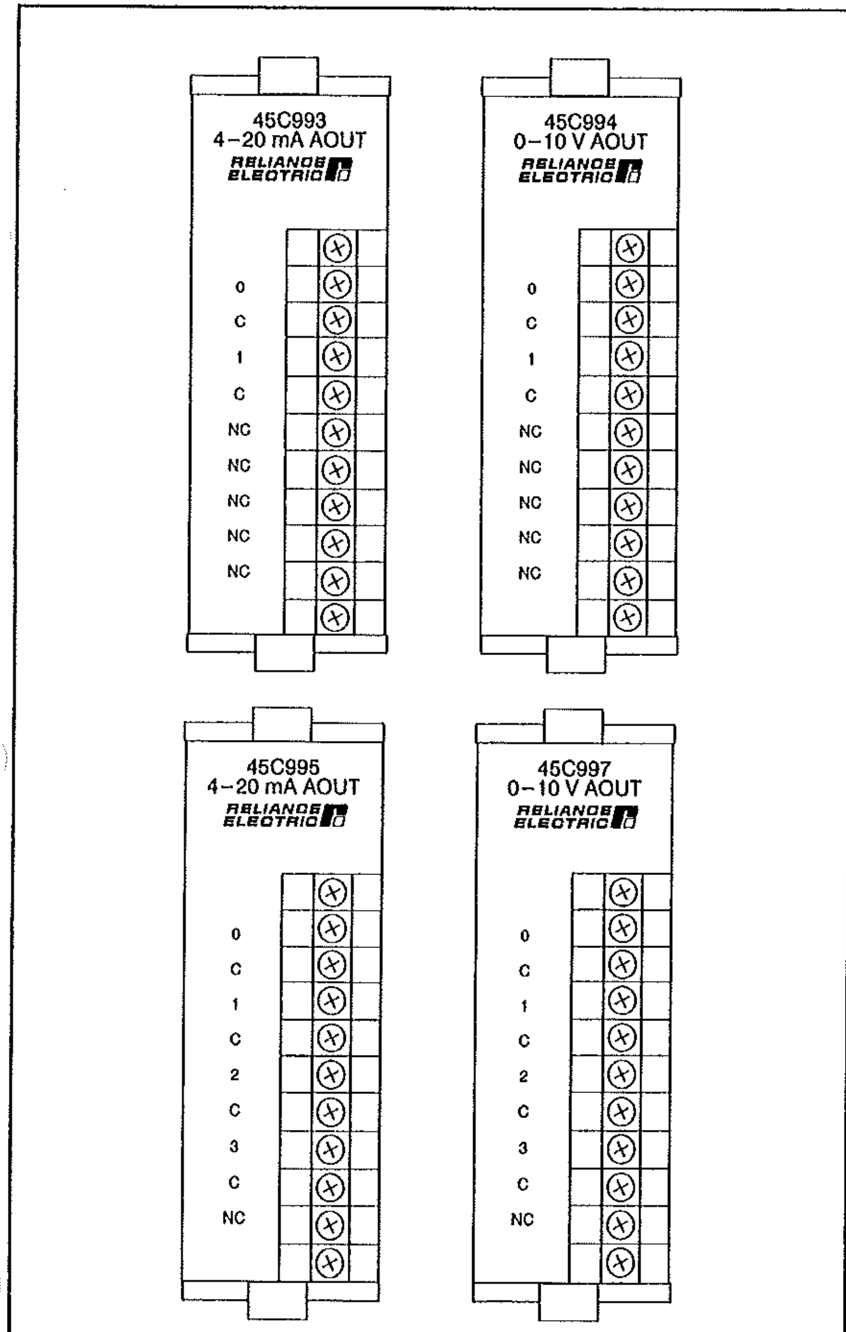


Figure 2.8 - Analog Output Modules

2.1.9 Counter Module

The 10kHz counter module has four software presets and four transistor outputs:

- M/N 45C982 Counter Module

As shown in figure 2.9, the module has connections for pulsetach inputs. Both quadrature and single-ended wiring configurations can be used. Module specifications are given in Appendix A.

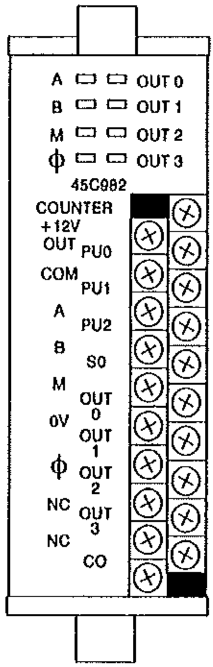


Figure 2.9 - Counter Module

2.1.10 Network Module

Network modules allow up to eight Shark XL or XL II systems to communicate together:

- M/N 45C987 Network Module

One module is required for each Shark XL or XL II system to be connected to the network. As shown in figure 2.10, the module has two rotary switches, a status LED, and terminals for the network communication wires. The rotary switches define the module's node number and the total number of nodes on the system. When the RUN LED is lit, it indicates that the module is transmitting data on the network. Module specifications are given in Appendix A. For additional information refer to the Network Module Instruction Manual (J-3807).

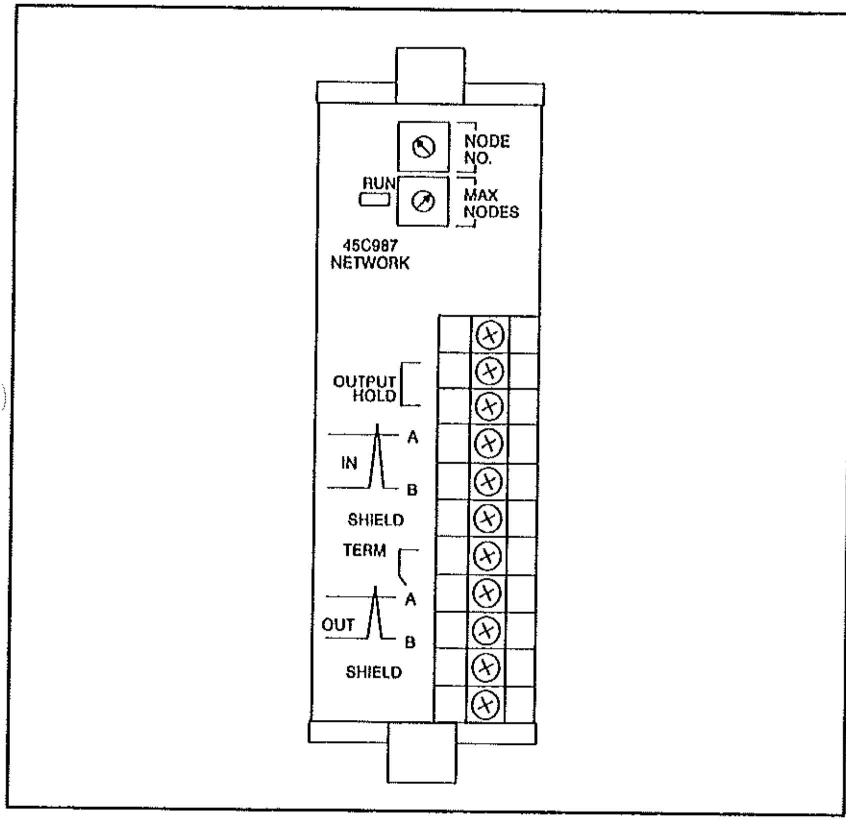


Figure 2.10 - Network Module

2.1.11 Remote Master Module

The remote master module along with remote target modules (see section 2.1.12) allow I/O to be in remote racks without processors:

- M/N 45C988 Remote Master Module

One remote master module per remote I/O network is required. As shown in figure 2.11, the module has a status LED and terminals for the remote I/O communication wires. When the RUN LED is lit, it indicates that the module is configured and operating correctly. Module specifications are given in Appendix A. For additional information refer to the Remote Master Module and Remote Target Module Instruction Manual (J-3805).

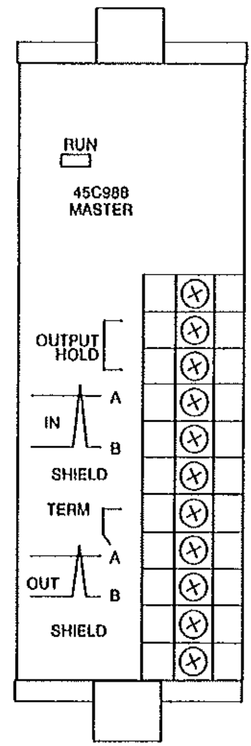


Figure 2.11 - Remote Master Module

2.1.12 Remote Target Module

Remote target modules along with a remote master module (see section 2.1.11) allow I/O to be in remote racks without processors:

- M/N 45C989 Remote Target

One module is required for each remote rack that is to be connected to the remote I/O network. As shown in figure 2.12, the module has two rotary switches, two status LEDs, and terminals for the remote I/O communication wires. The rotary switches define the slot address of the first remote module in the rack and the total number of slots addressed by the target module in the rack. When the PWR LED is lit, it indicates that power is applied to the rack's backplane. When the RUN LED is lit, it indicates that the module is configured and operating correctly. Module specifications are given in Appendix A. For additional information refer to the Remote Master Module and Remote Target Module Instruction Manual (J-3805).

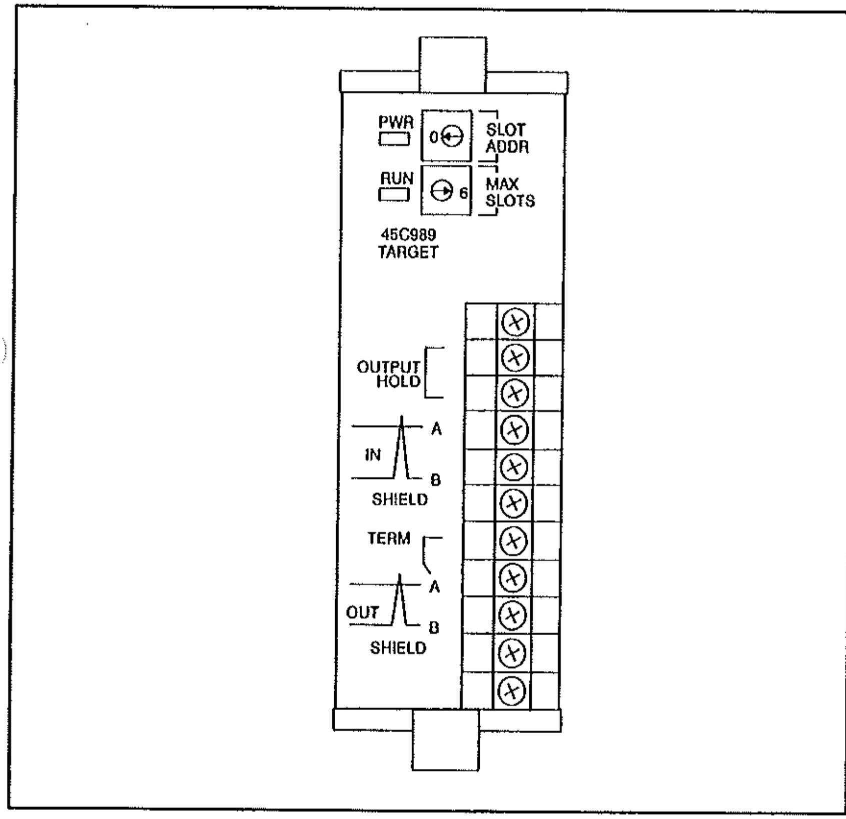


Figure 2.12 - Remote Target Module

2.1.13 AutoMax Remote I/O Shark Interface Module

The AutoMax Remote I/O Shark Interface module allows Shark I/O to communicate on an AutoMax remote I/O network:

- M/N 57C554 AutoMax Remote I/O Shark Interface Module

The Shark Interface module functions as an I/O scanner in a remotely-located XLII rack which is connected to an AutoMax remote I/O network. When a Shark rack containing I/O modules is to be used as a drop on an AutoMax remote I/O network, the Shark Interface module replaces the Processor module in the rack. For additional information refer to the Remote I/O Communications manual (J-3606-1).

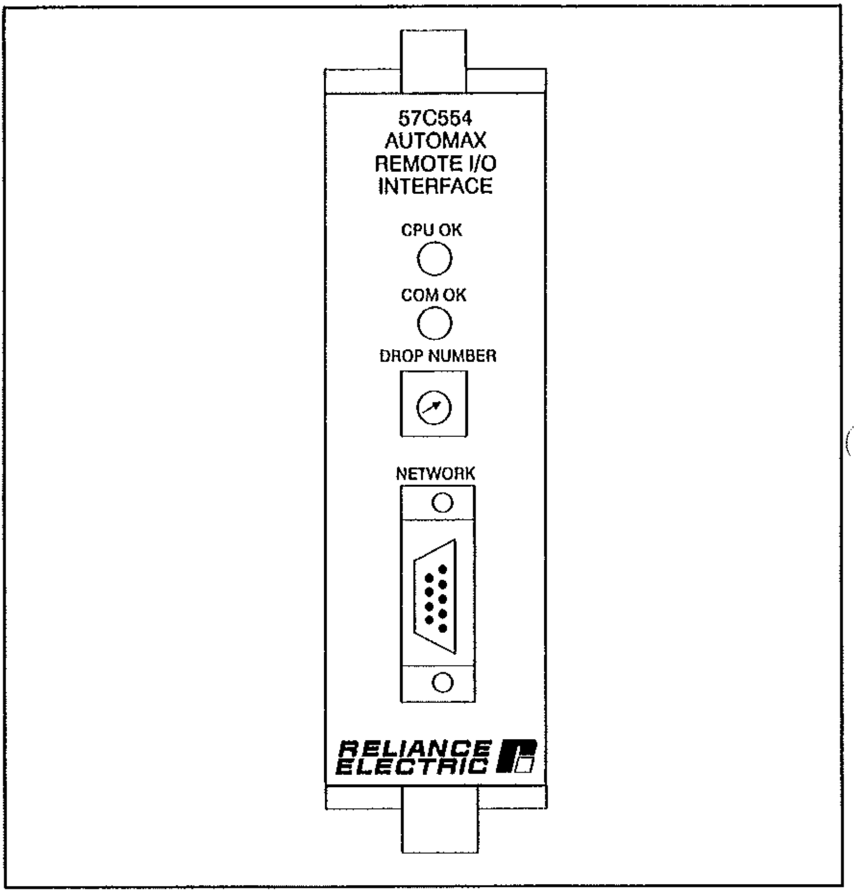


Figure 2.13 - AutoMax Remote I/O Shark Interface Module

2.1.14 Standard Programmer

This programmer allows application programs to be entered into memory. It also allows programs to be stored on cassette tape by connecting a cassette tape recorder to the MIC and EAR terminals:

- M/N 45C950 Standard Programmer

The programmer attaches to the processor's faceplate connector through a five-inch long sixteen-pin ribbon cable. See figure 2.14. An expansion cable (M/N 45C939) is available. The expansion cable is five feet long. A mounting bracket for the programmer (M/N 45C936) is also available. Installation instructions are given in chapter 3. For more information refer to the Standard Programmer Instruction Manual (J-3803).

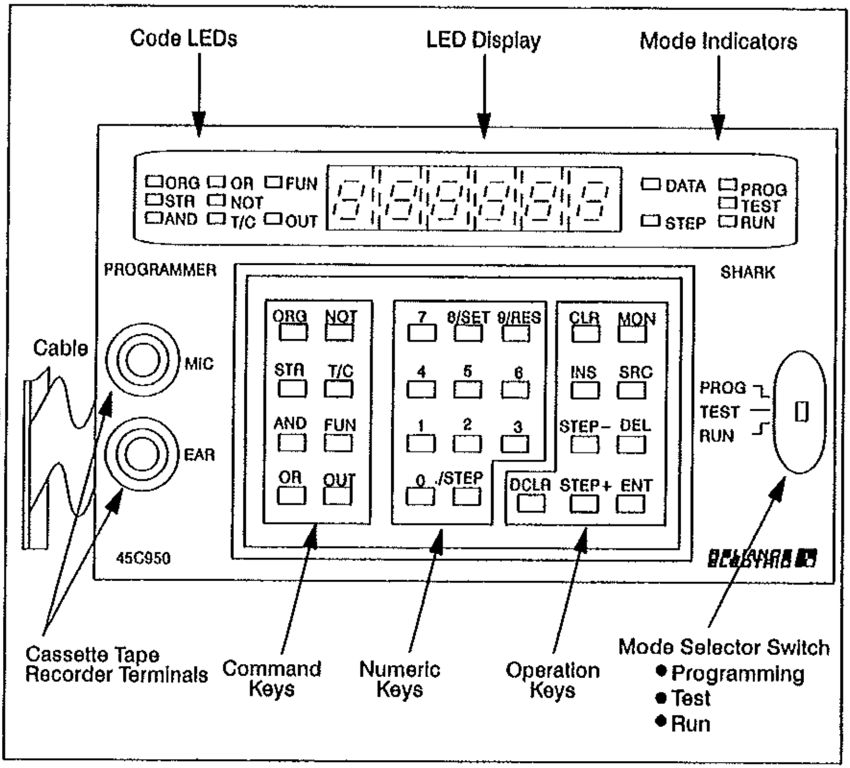


Figure 2.14 - Standard Programmer

2.1.15 Universal Programmer

This programmer has an EPROM burner and allows application programs to be entered into processor memory, stored on cassette tape, and printed out on a printer attached to the RS-232 connector:

- M/N 45C951 Universal Programmer

The programmer attaches to the processor's faceplate connector through a five-inch long sixteen-pin ribbon cable. See figure 2.15. An expansion cable (M/N 45C939) is available. The expansion cable is five feet long. A mounting bracket for the programmer (M/N 45C936) is also available. Installation instructions are given in chapter 3. For more information refer to the Universal Programmer Instruction Manual (J-3600).

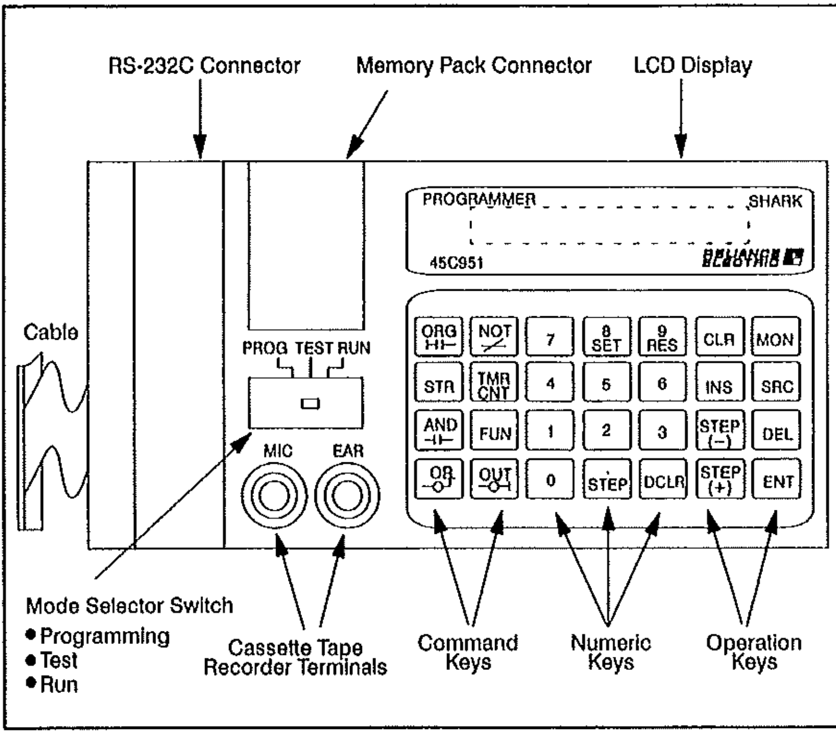


Figure 2.15 - Universal Programmer

2.1.16 ReSource Shark IBM Programming Software

The Shark XL II can be programmed using an IBM-compatible personal computer and the following software:

- M/N 45C152 ReSource Shark IBM Programming and Documentation Executive
- M/N 45C153 ReSource Shark IBM Programming and Documentation Executive Upgrade

The personal computer connects to the processor through the on-board RS-232 port.

3.0 INSTALLATION

This chapter describes how to install a Shark XL II rack and modules along with the required power, Emergency-Stop, and I/O wiring. Also included are rack and module replacement procedures.

> **DANGER**
> **ONLY QUALIFIED ELECTRICAL PERSONNEL FAMILIAR WITH THE CONSTRUCTION AND OPERATION OF THIS EQUIPMENT AND THE HAZARDS INVOLVED SHOULD INSTALL, ADJUST, OPERATE, OR SERVICE THIS EQUIPMENT. READ AND UNDERSTAND THIS MANUAL IN ITS ENTIRETY BEFORE PROCEEDING. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.**

> **DANGER**
> **THE USER IS RESPONSIBLE FOR CONFORMING TO THE NATIONAL ELECTRIC CODE AND ALL OTHER APPLICABLE LOCAL CODES. WIRING, GROUNDING, DISCONNECTS, AND OVERCURRENT PROTECTION ARE PARTICULARLY IMPORTANT. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.**

3.1 Initial Installation

The following sections describe how to install a rack, modules, and their associated wiring. Table 3.1 lists some general installation guidelines that you should follow as you design your Shark installation.

Table 3.1 - Installation Guidelines

- Do not mount the rack in direct sunlight.
- Keep the rack's operating temperature between 32 and 131°F (0 and 55°C). If the operating temperature falls below 32°F (0°C), install a heater. If the temperature exceeds 131° (55°C), install a fan to cool the unit.
- Maintain the rack's relative humidity at 30 to 90% (non-condensing).
- The rack should be located away from: combustible gases, dust, and metal particles.
- Isolate the rack from excessive vibration and shock.
- Allow at least 2 inches (5.0 cm) above and below the rack for ventilation. Allow at least 2 inches (5.0 cm) to either side of the rack for ventilation. If you mount an expansion rack next to the main rack, leave at least 2 inches (5.0 cm) of clearance between them.
- Do not place a heat source under the rack.
- Allow at least 8 inches (20.3 cm) between the rack and any high voltage (over 3000V) lines.
- Do not place I/O wires and power lines in the same conduit or wire tray.
- I/O lines should not exceed 95 feet (28 meters). If the lines exceed 95 feet, separate the input wires from the output wires.
- Do not exceed the power supply's current capacity. See Appendix A for module current consumption ratings.

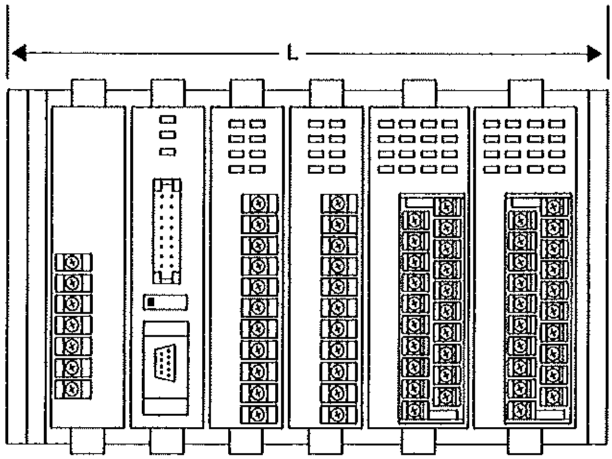
3.2 Rack Installation and Mounting

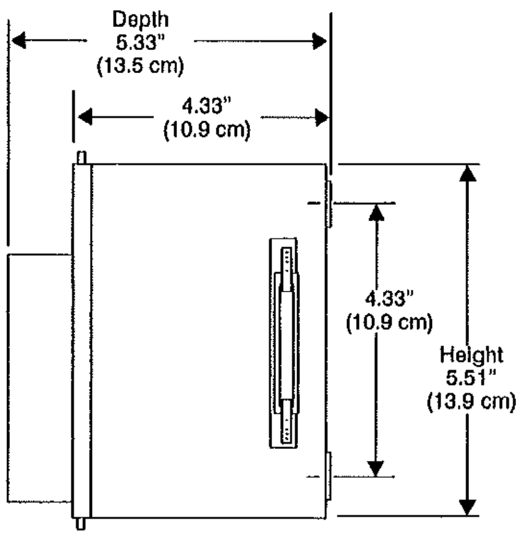
Use the following procedure to install a Shark XL II rack:

Step 1. Attach the rack to a flat mounting surface after referring to the guidelines in Table 3.1. Figures 3.1 to 3.3 show the mounting dimensions. An optional DIN (German Engineering Standard) mounting adapter (M/N 45C937) is available. Rack weights without modules range from 1 pound, 3 ounces (0.532 kg) for a 3-slot rack to 2 pounds, 10 ounces (1.176 kg) for a 7-slot rack. The racks are designed to be mounted using #8 or M4 panhead screws.

Note that if the rack is to be used as an expansion rack it must be mounted within 23" inches of the main rack because the expansion cable is 23" (0.6 m) long. See figure 3.4. Connect the cable from the right side of the main rack to the left side of the expansion rack.

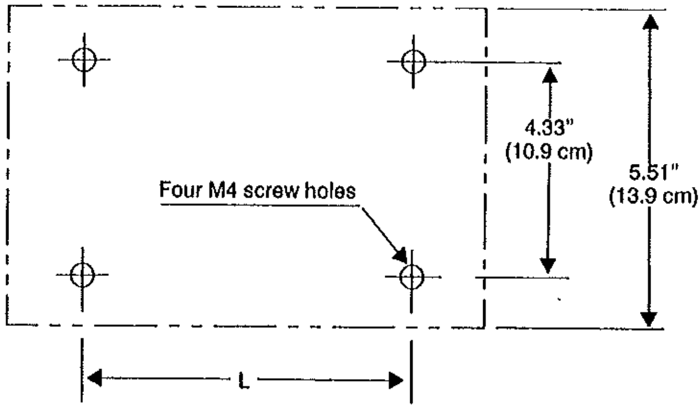
Step 2. If your installation requires that the Shark programmer be mounted next to the rack, use a programmer mounting bracket (M/N 45C396). See figures 3.5 to 3.7 for programmer mounting dimensions. The mounting bracket is designed to be mounted using #8 or M4 panhead or roundhead screws. The programmers slide down into the bracket which securely holds them to the mounting surface.





	Dimension L
3-Slot Rack	6.29 inches (15.9 cm)
4-Slot Rack	7.67 inches (19.4 cm)
5-Slot Rack	9.05 inches (22.9 cm)
6-Slot Rack	10.43 inches (26.4 cm)
7-Slot Rack	11.81 inches (29.9 cm)
9-Slot Rack	13.19 inches (33.4 cm)

Figure 3.1 - Rack Dimensions



	Dimension L
3-Slot Rack	3.14 inches (7.9 cm)
4-Slot Rack	4.72 inches (11.9 cm)
5-Slot Rack	6.29 inches (15.9 cm)
6-Slot Rack	7.87 inches (19.9 cm)
7-Slot Rack	9.44 inches (23.9 cm)
9-Slot Rack	12.58 inches (31.9 cm)

Figure 3.2 - Rack Mounting Holes

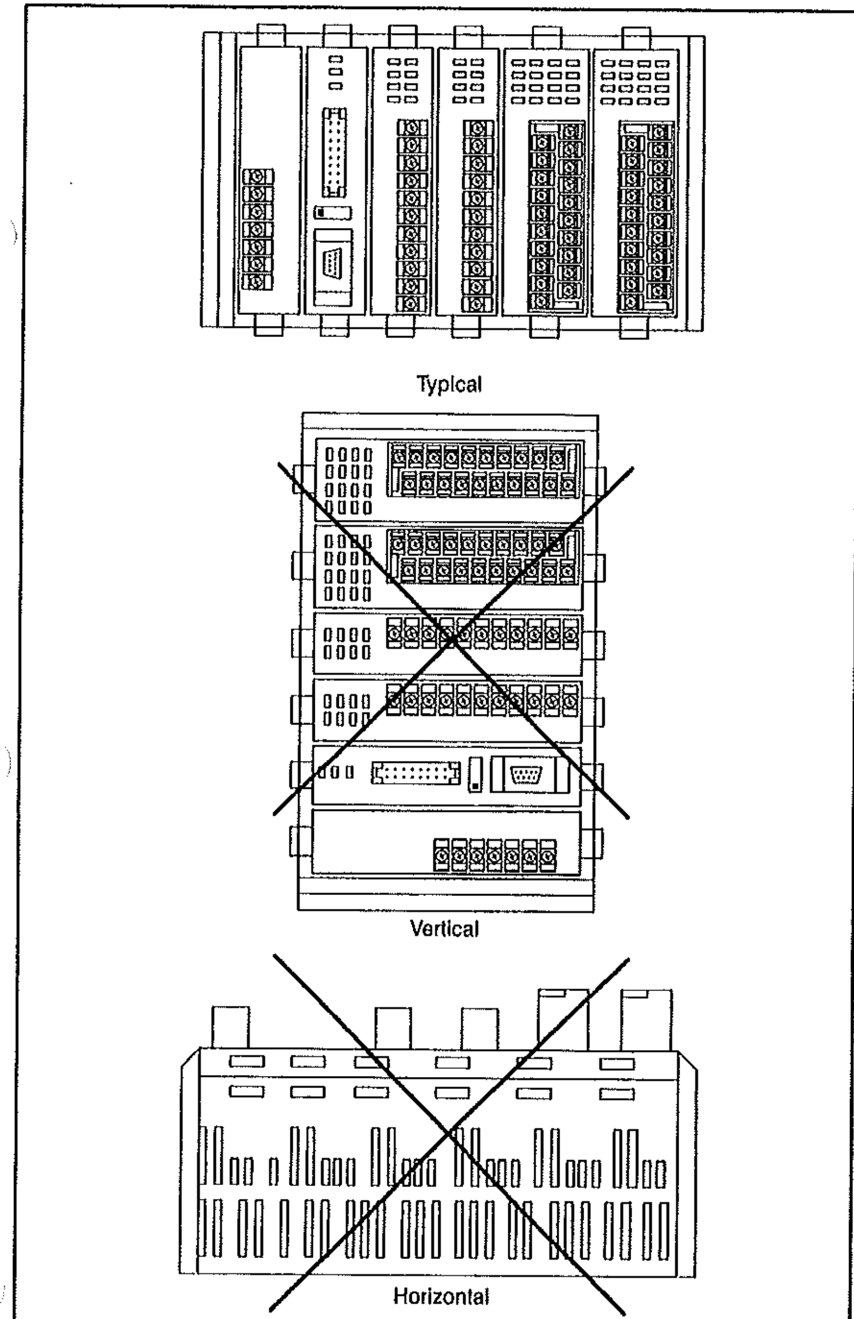


Figure 3.3 - Rack Mounting Positions

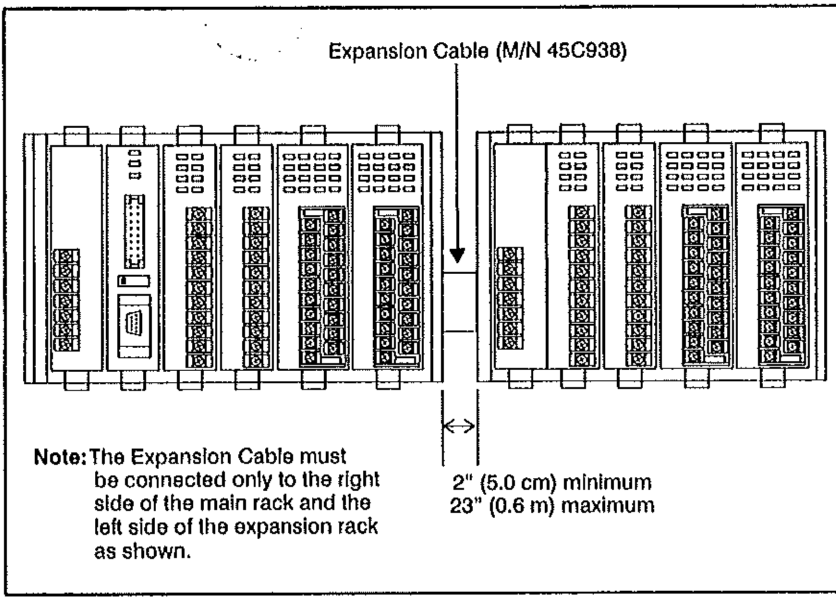


Figure 3.4 - Expansion Rack Mounting

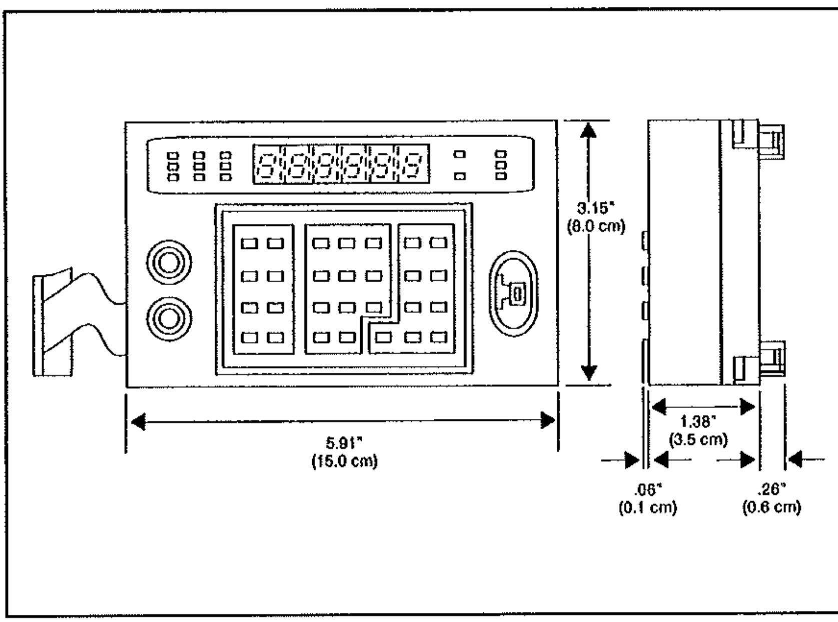


Figure 3.5 - Standard Programmer Dimensions

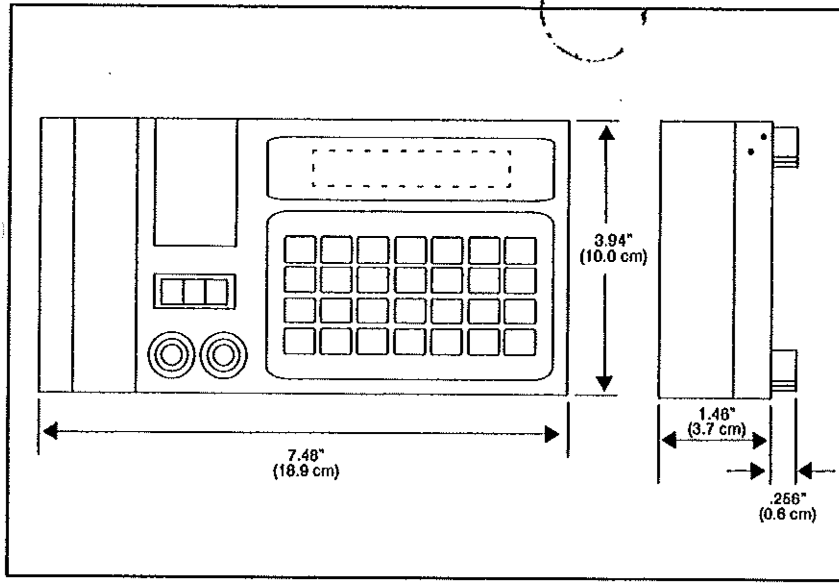


Figure 3.6 - Universal Programmer Dimensions

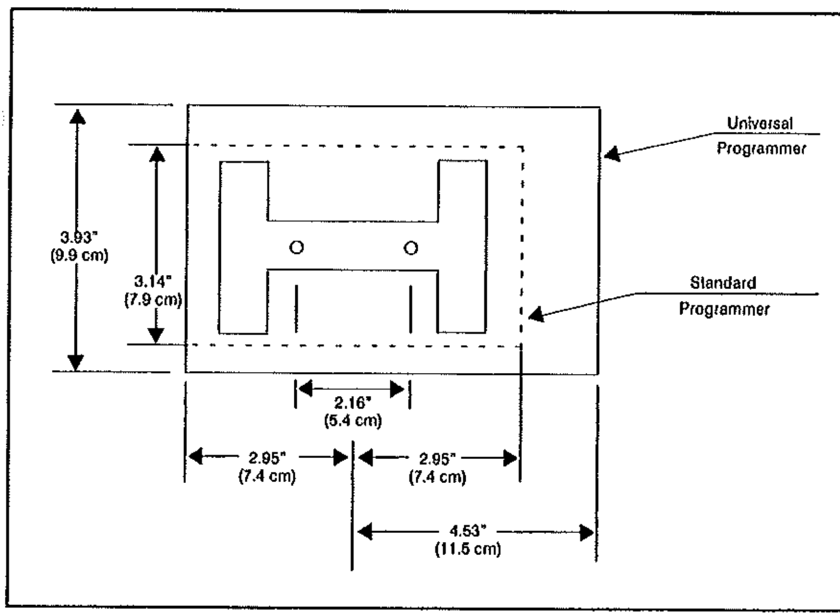


Figure 3.7 - Standard and Universal Programmer Mounting Dimensions

Step 3. Install the modules in the rack. Remove the modules from their shipping containers and insert them into the correct slots in the rack. Press the modules into the slots until the locking tabs on the top and bottom of the modules latch into the rack.

Before installing the processor module, you must install its memory. Follow the procedure in section 3.4 to install the processor's memory.

WARNING

DO NOT INSTALL OR REMOVE THE PROCESSOR MODULE WITH POWER APPLIED AS UNCONTROLLED MACHINE OPERATION MAY RESULT. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY OR DAMAGE TO EQUIPMENT.

You must also set the processor module's Command Mode switches as shown in Figure 2.3. Switch 1, the Transmission Speed Control switch, sets the baud rate of the RS-232-C port:

- Switch 1: ON position = 9600 baud (Factory Setting)
  Switch 1: OFF position = 4800 baud

Switch 2, the Operation Control Command switch, sets the processor's power-up mode of operation. When the switch is set to ON, the processor will retain its last mode of operation before power was removed. If the processor is in the run mode when power is removed, it will return to the run mode when power is restored. When the switch is set to OFF, the processor will not retain its last mode of operation when power is removed. When power is restored, the processor will go into the program mode:

- Switch 2: ON position = AutoRun Enable (Factory Setting)
  Switch 2: OFF position = AutoRun Disable

Note that the power supply module must be placed in the left-most slot in the rack. Three power supplies are available, 115/230 VAC or 24 VDC. Before placing an A-C power supply module in the rack the input power jumper must be set for the proper voltage. The A-C power supply modules are set at the factory for 115 VAC operation. If your installation uses 230 VAC, you will need to change the jumper setting. The jumper is located in the center of the module as shown in figure 2.2. Move the jumper as shown in figure 3.8 for 230 VAC operation. Then install the power supply module as described above.

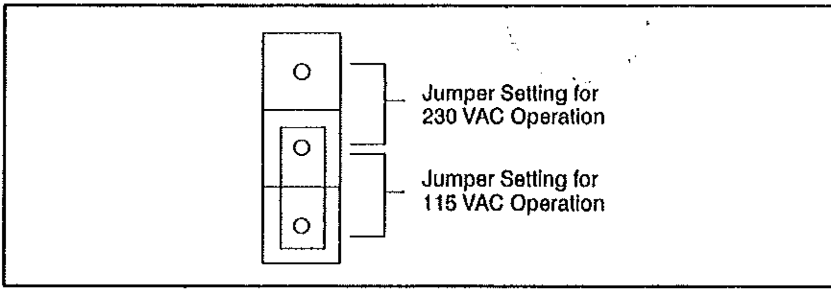


Figure 3.8 - Input Power Jumper

Step 4. Connect the power wiring as described in sections 3.3 and 3.3.1.

Step 5. Connect the E-Stop wiring as described in sections 3.3 and 3.3.2.

Step 6. Connect the input wiring as described in sections 3.3 and 3.3.3.

Step 7. Connect the output wiring as described in sections 3.3 and 3.3.4.

Step 8. Verify the proper operation of the system by using the handheld programmer or SPX software to monitor input signals and force output signals.

3.3 Wiring

> **WARNING**
> **THE USER MUST PROVIDE AN EXTERNAL, HARDWIRED EMERGENCY STOP CIRCUIT THAT WILL DISABLE THE SYSTEM IN CASE OF IMPROPER OPERATION. UNCONTROLLED MACHINE OPERATION MAY RESULT IF THIS IS NOT DONE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.**

The following sections provide information on power, Emergency-Stop, input, and output wiring.

The installation of wiring must conform to all applicable codes. To reduce the possibility of electrical noise interfering with controller operation, exercise care when installing wiring from the rack to the external devices. For more specific information, refer to IEEE Standard 518 (Guide for the Installation of Electrical Equipment to Minimize Electrical Noise Inputs to Controllers).

Route cables to protect them from abrasion, vibration, moving parts, and personnel traffic. Be sure the cables do not touch surfaces such as concrete which could damage the outer jacket.

Protect cables from: oil, grease, acids, caustics, and other hazardous chemicals that could damage the outer jacket or corrode metal connectors.

Isolate cables from electrical interference by routing them with cables of like voltages and power levels. This routing helps to reduce electrical noise.

There are four main wiring groups:

- Low Level

  Shark remote I/O and network signals, analog signals less than 50 VDC, signal commons, very low voltage signals from devices such as thermocouples and RTDs, operational amplifier signals, power amplifier signals, digital tachometer and encoder signals, and phone circuits.

- Medium Level

  Analog signals greater than 50 VDC with less than 28 VAC ripple, switching circuit signals greater than 24 VDC and less than 28 VDC, relays greater than 24 VDC, analog tachometer and pilot signals.

- High Level

  Switching signals greater than 28 VDC, analog signals greater than 50 VDC with more than 28 VAC ripple, regulating signals of less than 20 Amps (motor loops), fused 250 VDC control bus signals, A-C feeder signals of less than 20 Amps, A-C control signals of less than 20 Amps (pushbuttons, indicator lights, and limit switches), convenience outlets, and panel lighting.

- Power Level

  A-C and D-C signals greater than 20 Amps.

Table 3.2 can help you in determining the required spacing between the I/O wiring and the other cables in the installation.

Table 3.2 - I/O Cable Separation

	Separation In Inches		
	Tray to Tray	**Conduit to Tray**	**Conduit to Conduit**
Low Level Signals:	0	0	0
Medium Level Signals:	1	1	1
High Level Signals:	6	4	3
Power Level Signals:	26	18	12

3.3.1 Power Wiring

DANGER

THE USER IS RESPONSIBLE FOR CONFORMING TO THE NATIONAL ELECTRIC CODE AND ALL OTHER APPLICABLE LOCAL CODES. WIRING, GROUNDING, DISCONNECTS, AND OVERCURRENT PROTECTION ARE OF PARTICULAR IMPORTANCE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

The input power terminals are L1 and L2 on the power supply module. Before you connect incoming power, be sure the power supply's input power jumper is set correctly. See figure 3.8. A typical wiring example is shown in figure 3.9.

CAUTION

REMOVE ALL WIRING (INPUT POWER AND I/O) CONNECTED TO THE PROGRAMMABLE CONTROLLER BEFORE YOU RUN A HIGH POTENTIAL TEST. ALSO REMOVE THE GREEN JUMPER ON THE POWER SUPPLY. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN DAMAGE TO OR DESTRUCTION OF THE EQUIPMENT.

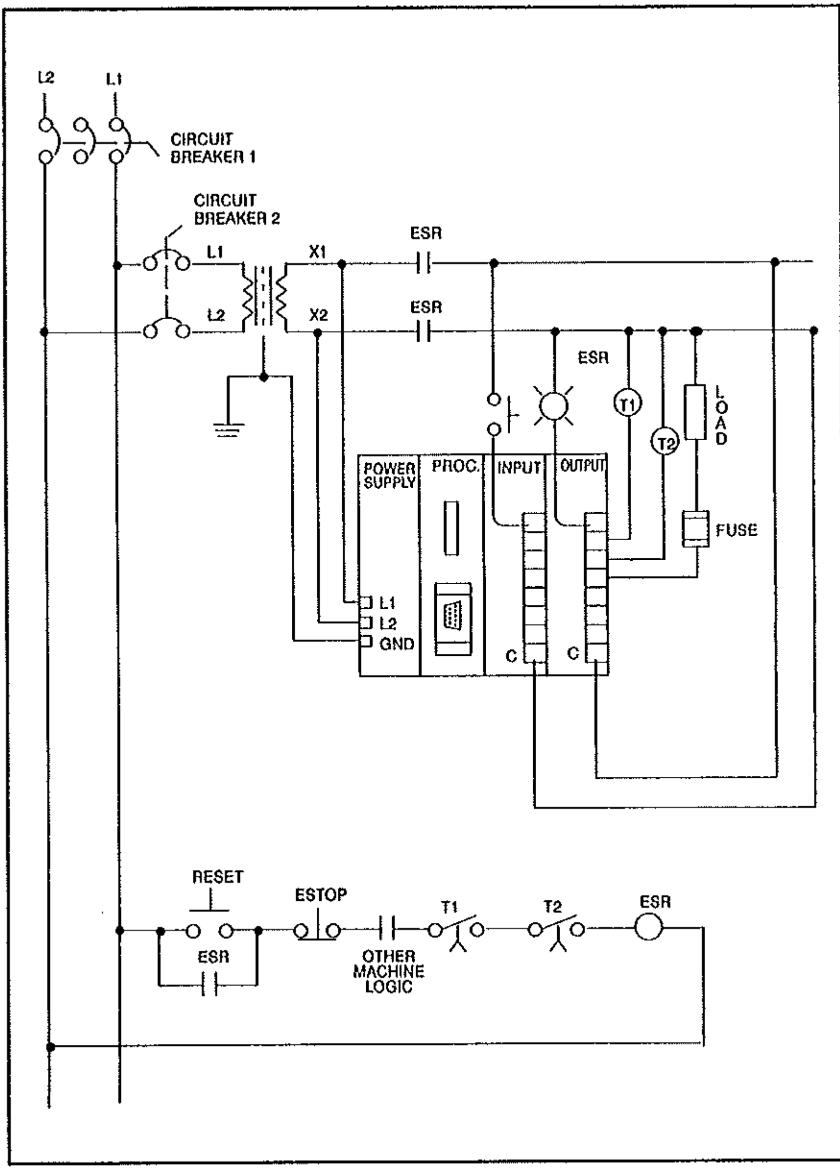


Figure 3.9 - Typical Input Power Wiring

3.3.2 Emergency-Stop Wiring

WARNING
YOU MUST PROVIDE AN EXTERNAL, HARDWIRED EMERGENCY STOP CIRCUIT THAT WILL DISABLE THE SYSTEM IN CASE OF IMPROPER OPERATION. UNCONTROLLED MACHINE OPERATION MAY RESULT IF THIS IS NOT DONE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

You must provide a hardwired emergency stop circuit outside of the Shark XL II programmable controller circuitry that will disable the system in case of improper operation. See figure 3.10. The hardwired circuit should include both an emergency stop relay (ESR) and watchdog timer contacts.

The ESR provides a way of removing power from all of the outputs at one time. This circuitry includes all hardwired logic to the safety circuits, permissives, and emergency stop pushbuttons. The ESR is energized only when all of the permissives (in series) are valid. Contacts from the ESR are used to power all of the I/O circuits. When the ESR is de-energized, all power to the I/O is removed.

Watchdog timers provide a test that the programmable controller is functioning properly and that it is correctly controlling its I/O. Timers T1 and T2 are external self-resetting timers. Logic in the Shark XL II toggles the two timers. For example, when timer T1 is timing, timer T2 is reset, and vice versa. The preset time of the timers should be at least 20% greater than the Shark XL II's I/O update time. For example, if timers T1 and T2 are cycling once per second, their preset times should be 1.2 seconds. If the Shark XL II's processor failed and stopped updating the system's I/O, one of the timers would time out and open the ESR, which would cause the system to shut down.

When the system shuts down it:

- turns all outputs off
- turns the processor's RUN LED off.

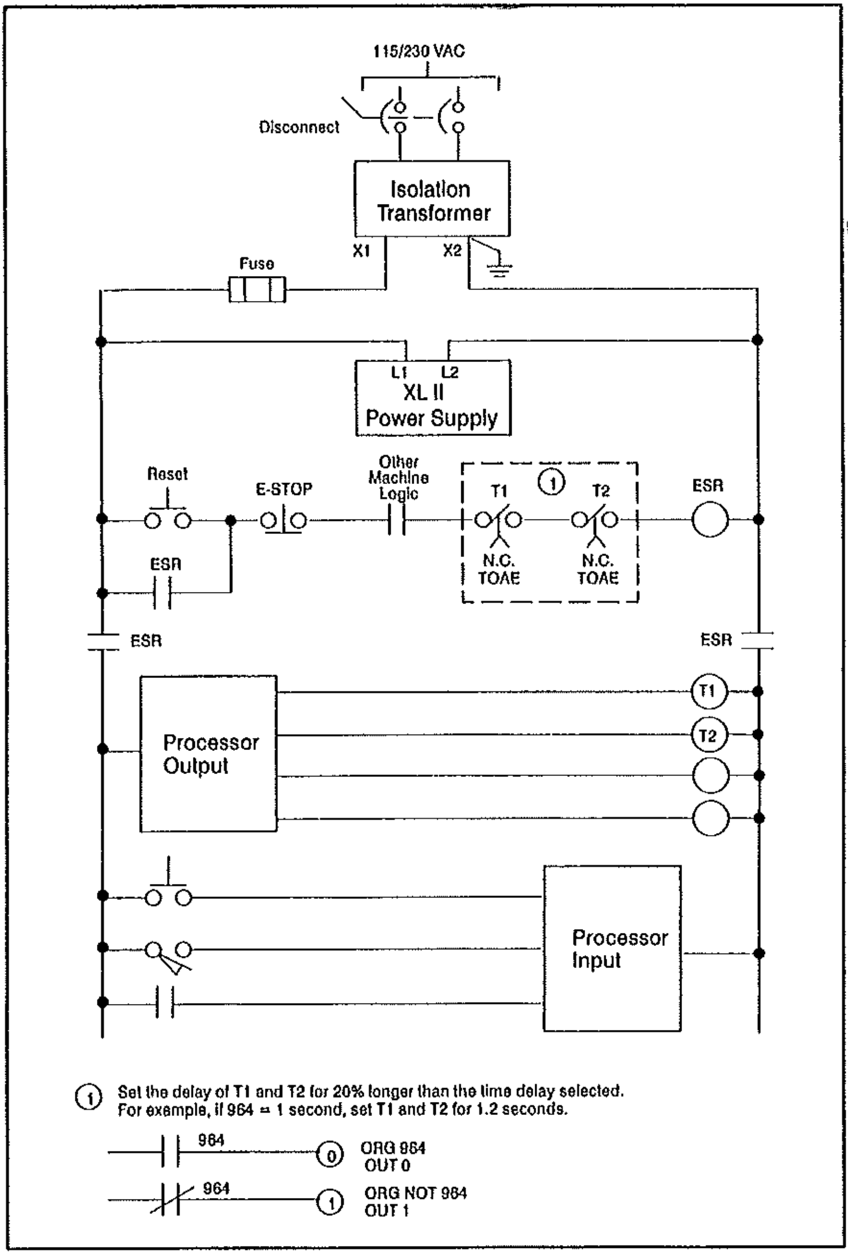


Figure 3.10 - Emergency Stop Wiring

3.3.3 Input Wiring

DANGER
HIGH VOLTAGE IS PRESENT ON THE INPUT TERMINALS. A PLASTIC COVER IS PROVIDED TO PROTECT AGAINST ACCIDENTAL ELECTRICAL SHOCK FROM THE INPUT TERMINALS. RE-INSTALL THE PLASTIC COVER AFTER THE WIRE CONNECTIONS HAVE BEEN MADE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

DANGER
EXTERNAL POWER WIRING MAY REMAIN ENERGIZED WHEN THE MAIN A-C POWER IS DISCONNECTED. IDENTIFY ALL SUCH EXTERNAL WIRING AND LOCK OUT OR TAG IT. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

DANGER
BEFORE CONNECTING THE INPUT WIRES TO THE MODULES, BE SURE POWER TO THE FIELD WIRING IS TURNED OFF. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

Use the following procedure to install the wiring between the input modules and field devices:

Step 1. Plan your input wiring connections. Note that input wires cannot be larger than 12 AWG or smaller than 22 AWG.

Step 2. Loosen the terminal screws on the input module to which the field wiring will be connected. See figures 3.11, 3.12, 3.13, 3.23, and 3.24 for input wiring examples.

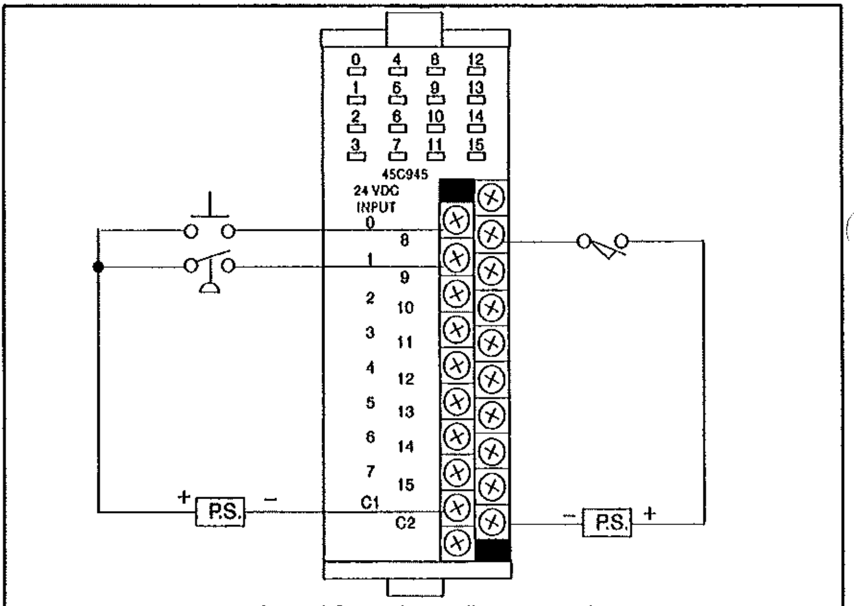
Step 3. For each input wire that is to be connected, bend the wire to form a neat wire bundle.

Step 4. Cut the wire. The wire should be 5/16 of an inch (0.7 cm) longer than is needed.

Step 5. Strip 5/16 of an inch of insulation from the end of the wire.

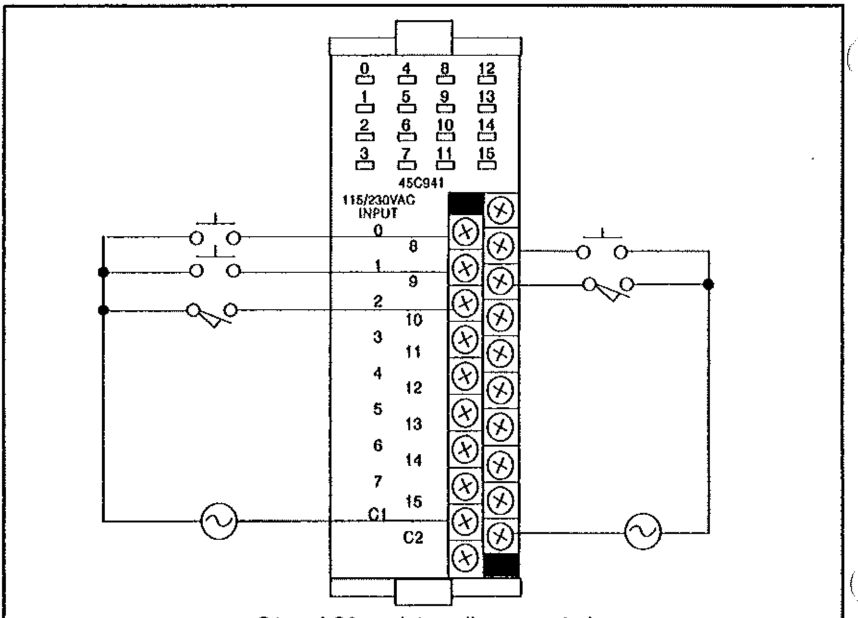
Step 6. Insert the wire under the terminal's pressure plate.

Step 7. Tighten the screw securely.



C1 and C2 are internally connected.

Figure 3.11 - 24 VDC Input Module Wiring (M/N 45C944, 45C945, and 45C946)



C1 and C2 are internally connected.

Figure 3.12 - 115/230 VAC Input Module Wiring (M/N 45C940, 45C941, and 45C942)

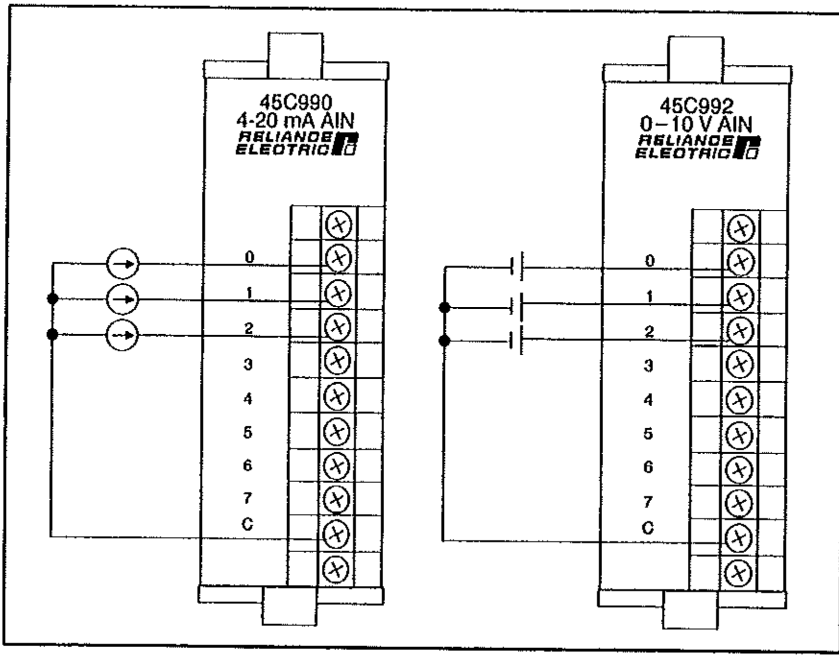


Figure 3.13 - Analog Input Module Wiring (M/N 45C990 and 45C992)

3.3.4 Output Wiring

DANGER
HIGH VOLTAGE IS PRESENT ON THE OUTPUT TERMINALS. A PLASTIC COVER IS PROVIDED TO PROTECT AGAINST ACCIDENTAL ELECTRICAL SHOCK FROM THE OUTPUT TERMINALS. RE-INSTALL THE PLASTIC COVER AFTER THE WIRE CONNECTIONS HAVE BEEN MADE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

DANGER
EXTERNAL POWER WIRING MAY REMAIN ENERGIZED WHEN THE MAIN A-C POWER IS DISCONNECTED. IDENTIFY ALL SUCH EXTERNAL WIRING AND LOCK OUT OR TAG IT. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

DANGER
BEFORE CONNECTING THE WIRING BETWEEN THE OUTPUT MODULES AND THE FIELD DEVICES, BE SURE POWER TO THE FIELD WIRING IS TURNED OFF. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

Use the following procedure to install the wiring between the output modules and the field devices:

Step 1. Plan your output wiring connections. Note that output wires cannot be larger than 12 AWG or smaller than 22 AWG. A surge suppressor is required in parallel with all inductive loads.

Step 2. Loosen the terminal screws on the output module to which the field wiring will be connected. See figures 3.14 to 3.24 for output wiring examples.

Step 3. For each output wire that is to be connected, bend the wire to form a neat wire bundle.

Step 4. Cut the wire. The wire should be 5/16 of an inch (0.7 cm) longer than is needed.

Step 5. Strip 5/16 of an inch of insulation from the end of the wire.

Step 6. Insert the wire under the terminal's pressure plate.

Step 7. Tighten the screw securely.

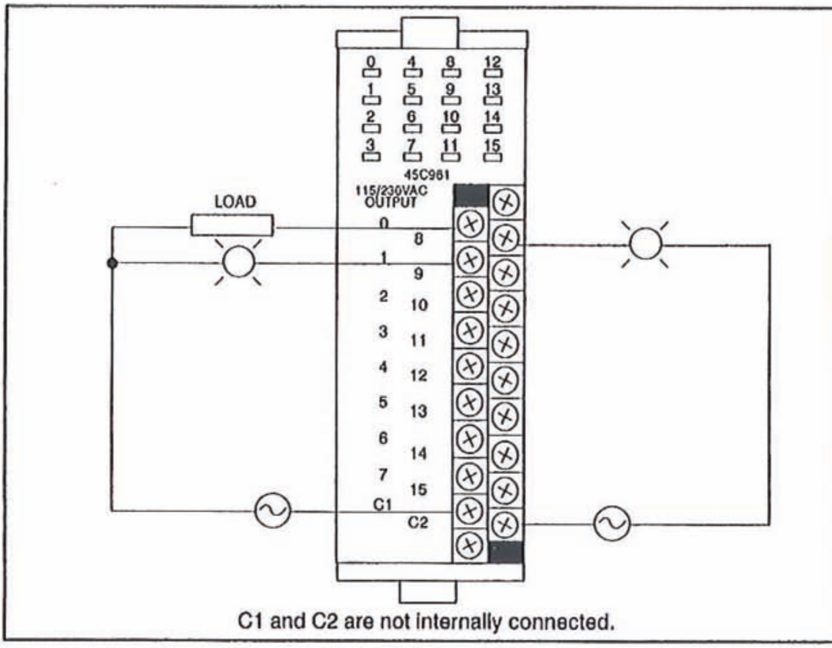


Figure 3.14 - 115/230 VAC Output Module Wiring (M/N 45C960, 45C961, and 45C965)

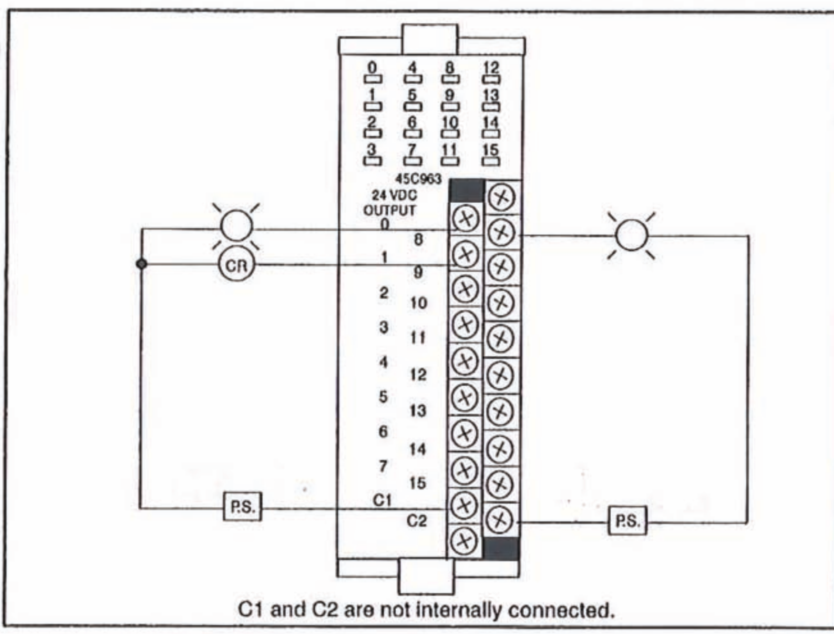


Figure 3.15 - 5-27 VDC Output Module Wiring (M/N 45C962, 45C963, and 45C964)

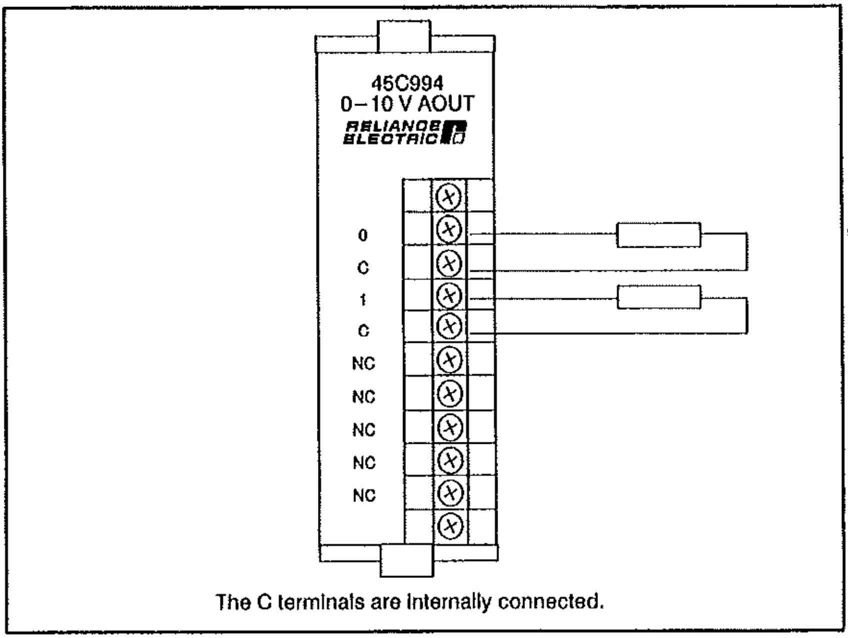


Figure 3.16 - Analog Output Module Wiring (M/N 45C997 and 45C994)

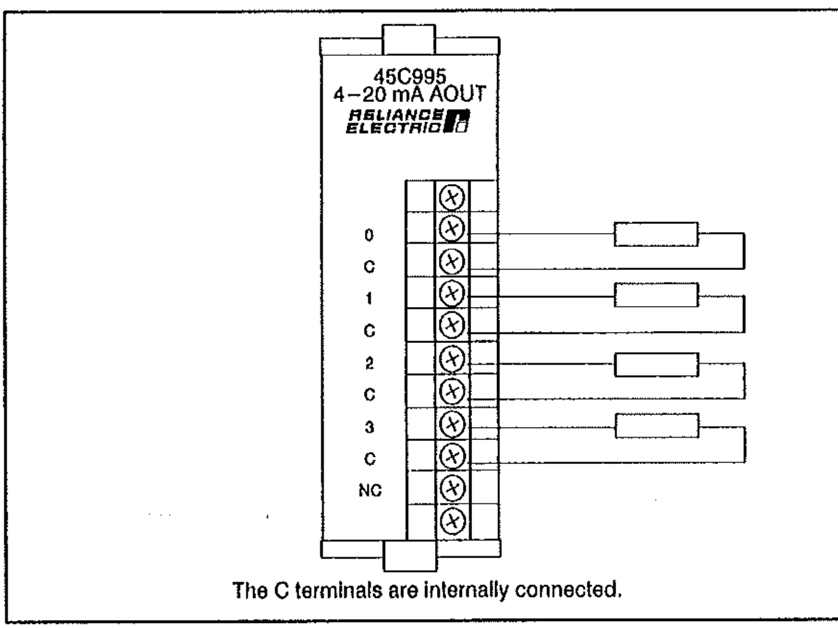


Figure 3.17 - Analog Output Module Wiring (M/N 45C995 and 45C993)

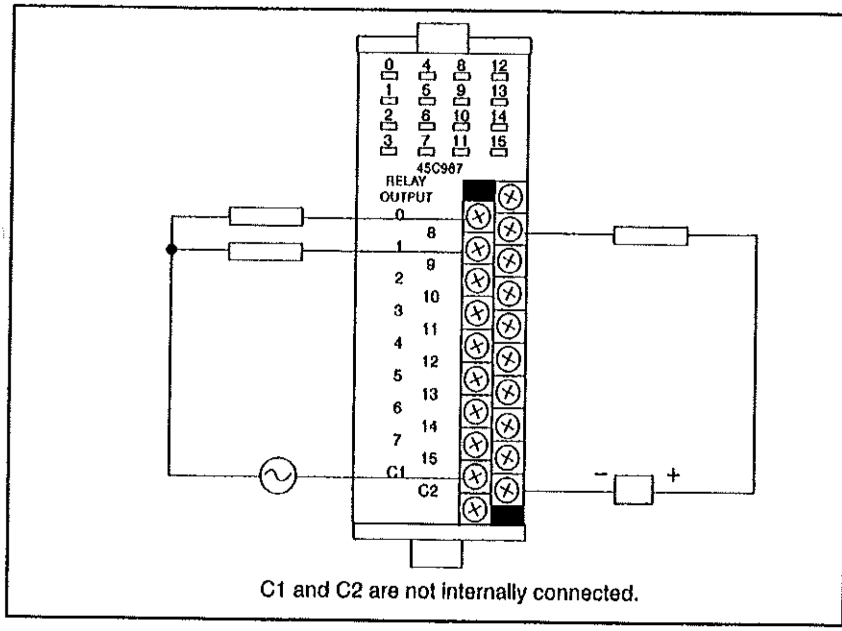


Figure 3.18 - Relay Output Module Wiring (M/N 45C966, 45C967, and 45C959)

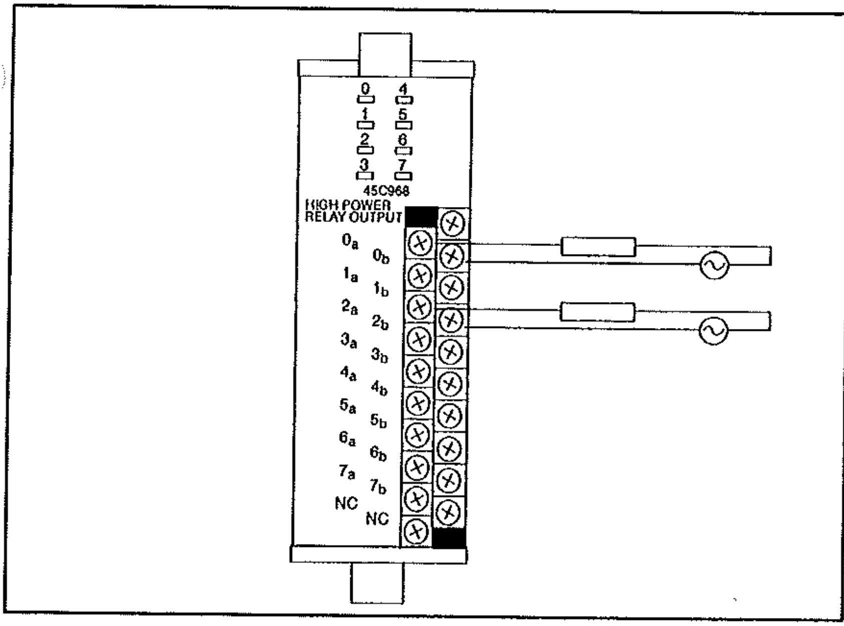


Figure 3.19 - Isolated Relay Output Module Wiring (M/N 45C968 and 45C969)

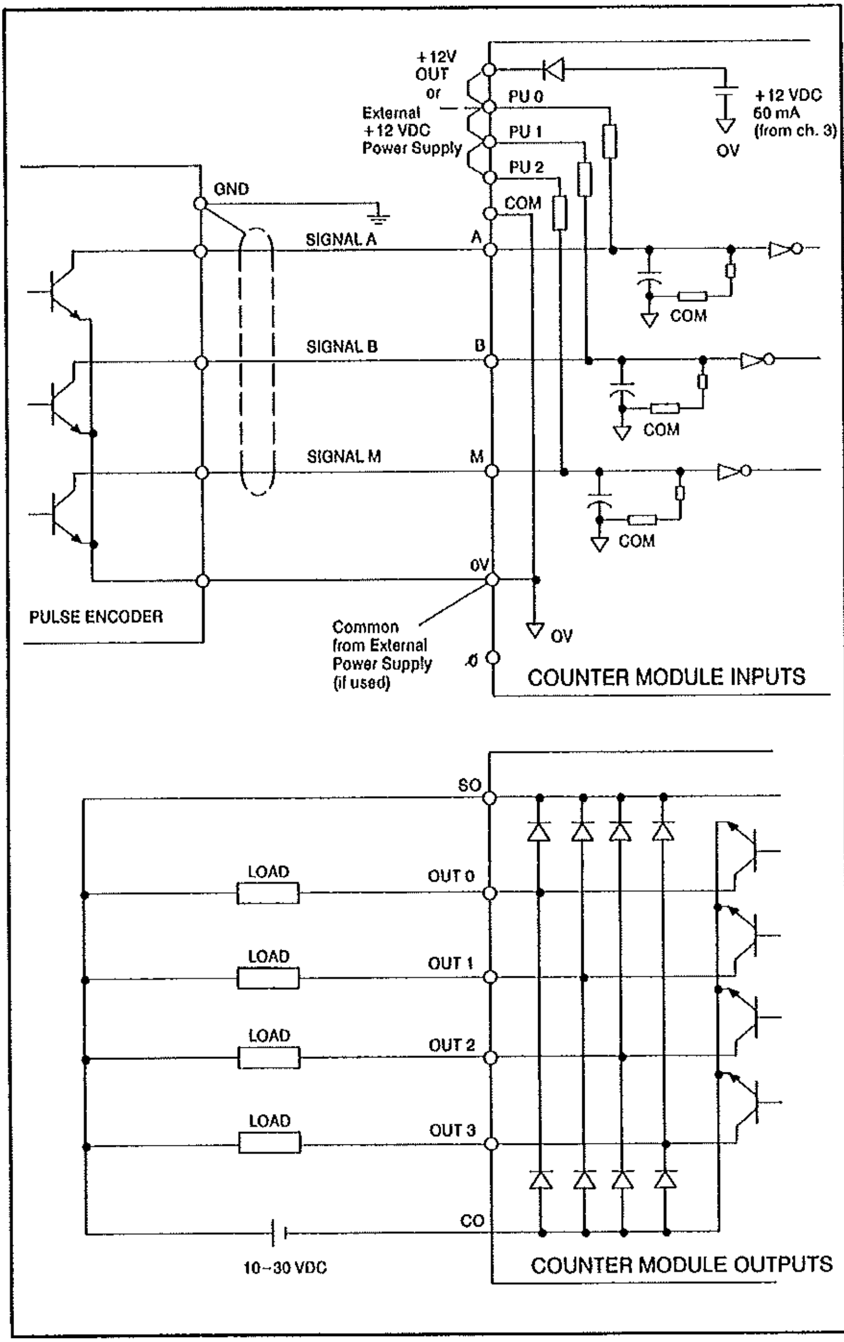


Figure 3.20 - Counter Module Wiring (M/N 45C982) - Single-Ended, Open-Collector, Quadrature with Marker Pulse Configuration

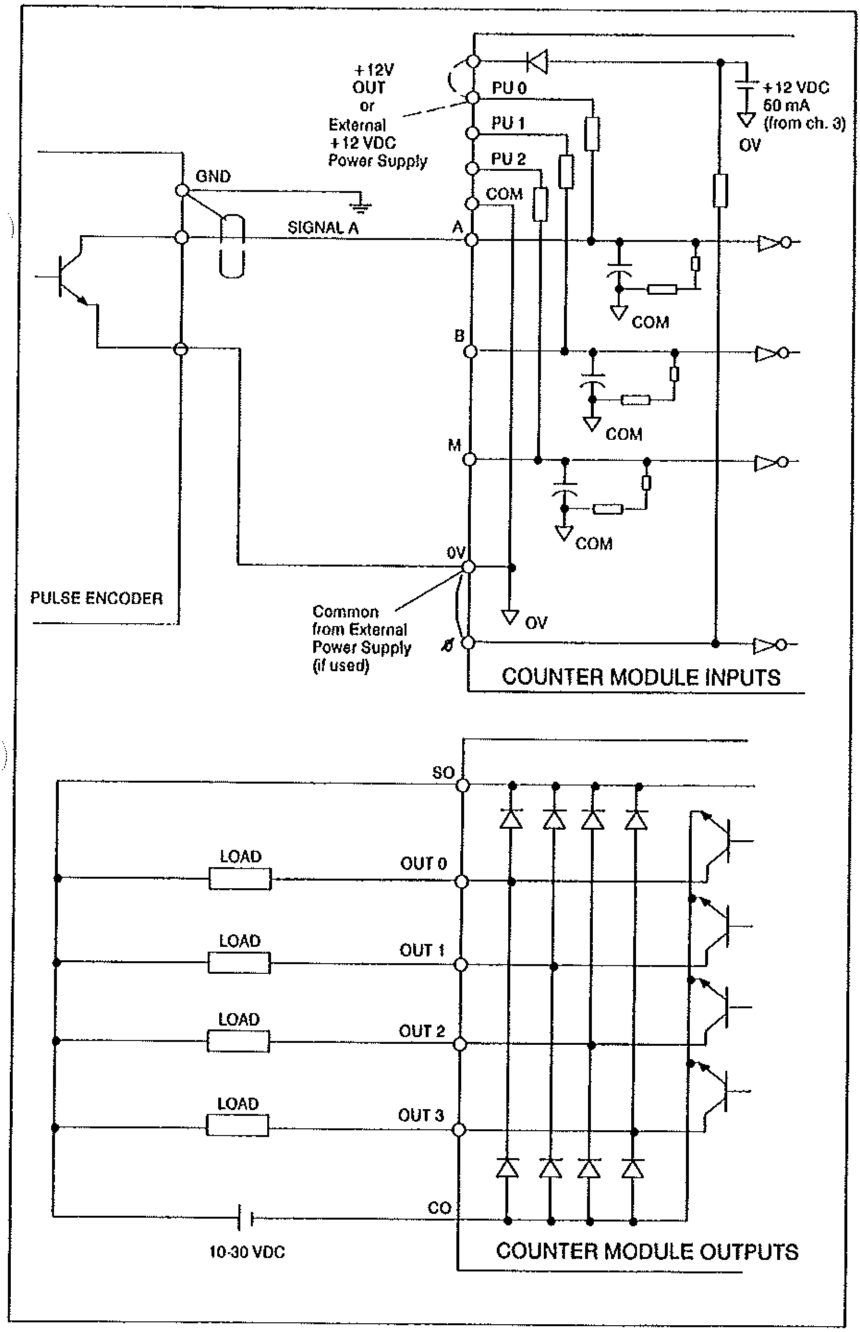


Figure 3.21 - Counter Module Wiring (M/N 45C982) - Single-Ended, Open-Collector, Single Channel Configuration

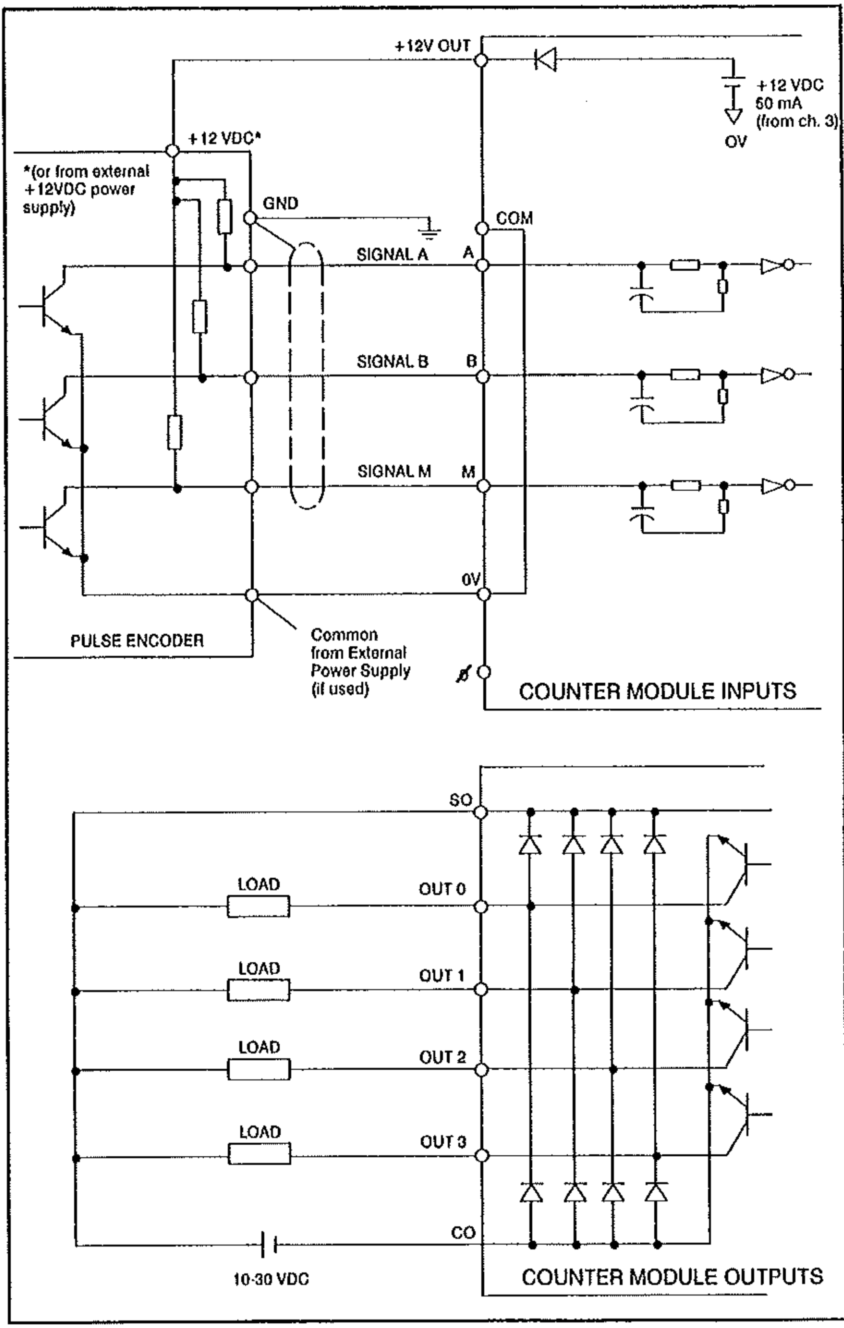
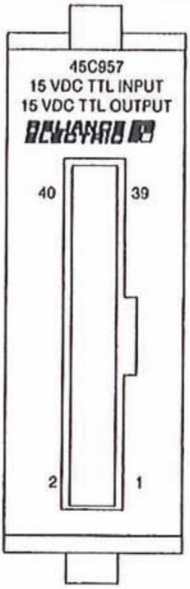


Figure 3.22 - Counter Module Wiring (M/N 45C982) - Single-Ended, Common Emitter Configuration



Pin 1	= Input Common 0	Pin 2	= Output Common 1
Pin 3	= Input Positive	Pin 4	= Output Positive for 0 to 7
Pin 5	= Input 0	Pin 6	= Output 0
Pin 7	= Input 1	Pin 8	= Output 1
Pin 9	= Input 2	Pin 10	= Output 2
Pin 11	= Input 3	Pin 12	= Output 3
Pin 13	= Input 4	Pin 14	= Output 4
Pin 15	= Input 5	Pin 16	= Output 5
Pin 17	= Input 6	Pin 18	= Output 6
Pin 19	= Input 7	Pin 20	= Output 7
Pin 21	= No Connection	Pin 22	= Output Common 2
Pin 23	= No Connection	Pin 24	= Output Positive for 8 to 15
Pin 25	= Input 8	Pin 26	= Output 8
Pin 27	= Input 9	Pin 28	= Output 9
Pin 29	= Input 10	Pin 30	= Output 10
Pin 31	= Input 11	Pin 32	= Output 11
Pin 33	= Input 12	Pin 34	= Output 12
Pin 35	= Input 13	Pin 36	= Output 13
Pin 37	= Input 14	Pin 38	= Output 14
Pin 39	= Input 15	Pin 40	= Output 15

Figure 3.23 - 16-Channel TTL Input/Output Module Wiring (M/N 45C957)

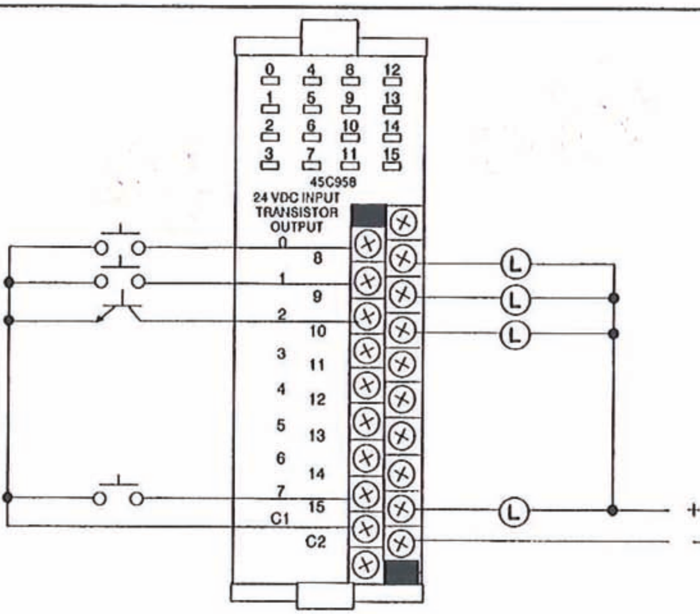


Figure 3.24 - 8-Channel 24 VDC Input/Output Module Wiring (M/N 45C958)

3.4 Processor Memory Installation

Use the following procedure to install a processor's EPROM or EEPROM memory:

Step 1. The EPROM/EEPROM memory connects to the processor through a slide-in connector. See figure 2.3. Install the EPROM/EEPROM memory module by placing it on the processor module and aligning it with the slide-in connector. Carefully push the memory module into the connector until the nylon standoffs have secured the module to the processor.

CAUTION

THIS MODULE CONTAINS STATIC-SENSITIVE COMPONENTS. CARELESS HANDLING CAN CAUSE SEVERE DAMAGE.

DO NOT TOUCH THE CONNECTORS ON THE BACK OF THE MODULE. WHEN NOT IN USE, THE MODULE SHOULD BE STORED IN AN ANTI-STATIC BAG. THE PLASTIC COVER SHOULD NOT BE REMOVED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN DAMAGE TO OR DESTRUCTION OF THE EQUIPMENT.

Step 2. Insert the processor module into the correct slot in the rack. Press the module into the slot until the locking tabs on the top and bottom of the module latch into the rack.

Step 3. Continue with the installation procedure in section 3.2.

3.5 Processor Memory Replacement

Use the following procedure to replace a processor's EPROM or EEPROM memory:

Step 1. Stop the application program from executing.

Step 2. Remove main input power from the rack. Remove power from the field and I/O wiring

DANGER

EXTERNAL POWER WIRING MAY REMAIN ENERGIZED WHEN THE MAIN A-C POWER IS DISCONNECTED. IDENTIFY ALL SUCH EXTERNAL WIRING AND LOCK OUT OR TAG IT. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

Step 3. Remove the processor from the rack. Press the tabs at the top and bottom of the module and carefully pull it out of the rack's backplane.

CAUTION
THIS MODULE CONTAINS STATIC-SENSITIVE COMPONENTS. CARELESS HANDLING CAN CAUSE SEVERE DAMAGE.

DO NOT TOUCH THE CONNECTORS ON THE BACK OF THE MODULE. WHEN NOT IN USE, THE MODULE SHOULD BE STORED IN AN ANTI-STATIC BAG. THE PLASTIC COVER SHOULD NOT BE REMOVED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN DAMAGE TO OR DESTRUCTION OF THE EQUIPMENT.

Step 4. The EPROM/EEPROM memory is connected to the processor through a slide-in connector. See figure 2.3. Remove the EPROM/EEPROM memory.

Step 5. Install the replacement EPROM/EEPROM memory module by placing it on the processor module and aligning it with the slide-in connector. Carefully push the memory module into the connector until the nylon standoffs have secured the module to the processor.

Step 6. Insert the processor module into the correct slot in the rack. Press the module into the slot until the locking tabs on the top and bottom of the module latch into the rack.

Step 7. Restore main input power to the rack. Restore power to I/O and field wiring.

3.6 Rack Replacement

Use the following procedure to replace a Shark XL II rack:

Step 1. Stop the application program from executing.

Step 2. Remove main input power from the rack. Remove power from field and I/O wiring.

DANGER
EXTERNAL POWER WIRING MAY REMAIN ENERGIZED WHEN THE MAIN A-C POWER IS DISCONNECTED. IDENTIFY ALL SUCH EXTERNAL WIRING AND LOCK OUT OR TAG IT. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

Step 3. Disconnect the input power wires from the power supply.

Step 4. Disconnect the I/O wiring from the input and output modules. Be sure to mark the wires so that they can be easily re-installed later.

If the module has a removable terminal strip (M/N 45C942, 45C946, 45C959, 45C964, and 45C965), you do not have to remove the wires attached to the terminal strip. Just remove the two screws that hold the terminal strip to the module's faceplate. See figure 3.25.

Step 5. Remove the modules from the rack. Press the tabs at the top and bottom of the modules and carefully pull them out of the rack's backplane.

CAUTION
THIS MODULE CONTAINS STATIC-SENSITIVE COMPONENTS. CARELESS HANDLING CAN CAUSE SEVERE DAMAGE.

DO NOT TOUCH THE CONNECTORS ON THE BACK OF THE MODULE. WHEN NOT IN USE, THE MODULE SHOULD BE STORED IN AN ANTI-STATIC BAG. THE PLASTIC COVER SHOULD NOT BE REMOVED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN DAMAGE TO OR DESTRUCTION OF THE EQUIPMENT.

Step 6. Remove the rack from its cabinet or enclosure.

Step 7. Install the replacement rack.

Step 8. Follow steps 1 to 5 in reverse order to re-install the modules and re-connect the input power and I/O wiring.

Step 9. Restore main input power to the rack. Restore power to the I/O and field wiring.

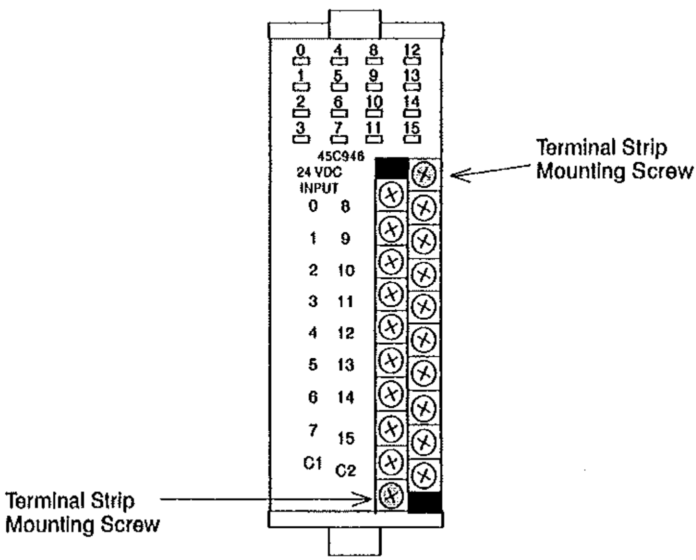


Figure 3.25 - Terminal Strip Mounting Screws on Modules with Removable Terminal Strips

3.7 Module Replacement

Use the following procedure to replace a module:

Step 1. Stop the application program from executing.

Step 2. Remove main input power from the rack. Remove power from the field and I/O wiring.

DANGER

EXTERNAL POWER WIRING MAY REMAIN ENERGIZED WHEN THE MAIN A-C POWER IS DISCONNECTED. IDENTIFY ALL SUCH EXTERNAL WIRING AND LOCK OUT OR TAG IT. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

Step 3. Disconnect the input power, I/O, or communication wires from the module to be replaced. Be sure to mark the wires so that they can be easily re-installed later.

If the module has a removable terminal strip (M/N 45C942, 45C946, 45C959, 45C964, and 45C965), you do not have to remove the wires attached to the terminal strip. Just remove the two screws that hold the terminal strip to the module's faceplate. See figure 3.25.

Step 4. Remove the module from the rack. Press the tabs at the top and bottom of the module and carefully pull it out of the rack's backplane.

CAUTION

THIS MODULE CONTAINS STATIC-SENSITIVE COMPONENTS. CARELESS HANDLING CAN CAUSE SEVERE DAMAGE.

DO NOT TOUCH THE CONNECTORS ON THE BACK OF THE MODULE. WHEN NOT IN USE, THE MODULE SHOULD BE STORED IN AN ANTI-STATIC BAG. THE PLASTIC COVER SHOULD NOT BE REMOVED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN DAMAGE TO OR DESTRUCTION OF THE EQUIPMENT.

Step 5. Place the module in the anti-static shipping bag that it came in. Place the module in its shipping container.

Step 6. Take the new module out of its anti-static bag.

Step 7. Insert the module into the correct slot in the rack. Press the module into the slot until the locking tabs on the top and bottom of the module latch into the rack.

Step 8. Re-connect the input power, I/O, or communication wires.

If the module has a removable terminal strip, simply re-attach the terminal strip to the module's faceplate with the two screws shown in figure 3.25.

Step 9. Restore main input power to the rack. Restore power to I/O and field wiring.

4.0 GENERAL PROGRAMMING

This chapter explains ladder logic, I/O processing, I/O addressing, and bit/word processing.

DANGER

ONLY QUALIFIED ELECTRICAL PERSONNEL FAMILIAR WITH THE CONSTRUCTION AND OPERATION OF THIS EQUIPMENT AND THE HAZARDS INVOLVED SHOULD INSTALL, ADJUST, OPERATE, OR SERVICE THIS EQUIPMENT. READ AND UNDERSTAND THIS MANUAL IN ITS ENTIRETY BEFORE PROCEEDING. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

4.1 Ladder Logic

A ladder diagram program is a series of sequences (also called rungs) that contain one or more inputs (either internal or external) in a series or parallel configuration that controls one or more outputs (either internal or external).

Each element of a sequence is an instruction to the processor to control a part of the electro-mechanical circuit. A sequence (or rung) consists of an output coil and its controlling input logic. A sequence may have up to 21 elements in series including the output coil and up to six elements in parallel. See figure 4.1.

External inputs may be devices such as: selector switches, proximity switches, pushbuttons, limit switches, or photo-electric switches. Inputs may be used within the program as many times as memory permits. The program uses the terminal number addresses to which these inputs are connected.

External outputs provide power to operate external devices such as indicator lights, solenoids, or electro-magnetic contactors. The program uses the terminal number addresses to which these outputs are connected. Outputs can also be referenced to contacts within the program. These internal outputs serve as auxiliary relays and provide special functions. Outputs may be used within the program as many times as memory permits.

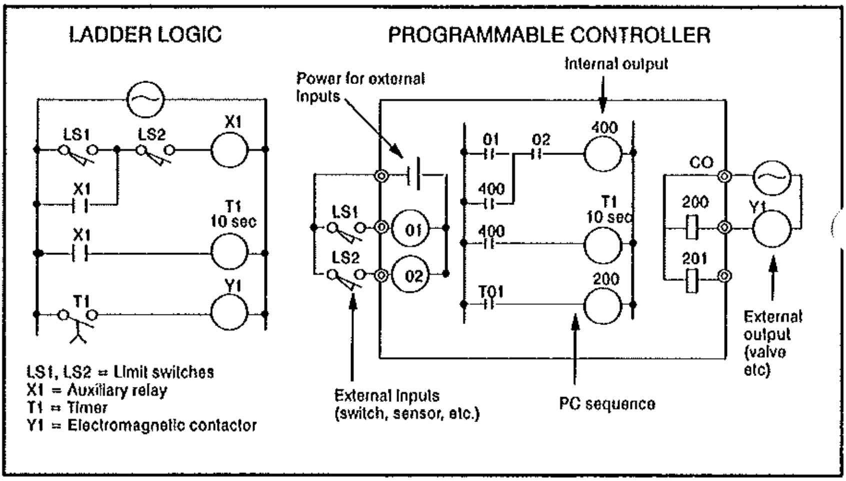


Figure 4.1 - Ladder Logic Sequence and Programmable Controller Ladder Diagram

4.1.1 Ladder Diagram Symbols

Typical ladder diagram symbols are shown below:

A normally-open contact (-[ ]-) represents an input to the sequence's logic. When its status is 0, it will remain open and not allow current to pass through it. When its status is 1, the contact will close and allow current to pass through it.

A normally-closed contact (-[/]-) also represents an input to the seqeuence's logic. When its status is 0, the contact will remain closed and allow current to pass through it. When its status is 1, the contact will open and not allow current to pass through it.

An output coil (-( )) is controlled by the input logic. If continuity exists along the path of the input circuit, the output coil will energize.

4.1.2 Ladder Diagram Format

Ladder diagram programs should conform to the following format rules:

- When using the IBM-compatible Shark Programming Executive (SPX) software, the maximum sequence or rung size is limited to 21 elements across (serially) including the output coil. The sequence can have up to six parallel input rows. See example 1 of figure 4.2. The rung elements can be normally-open/ normally-closed contacts or horizontal lines. Note that the handheld programmers do not limit a rung to 21 elements across and six down.
- There is no limit to the number of times a contact, output, or counter can be used in the program other than the size limitation of memory. See example 2 in figure 4.2.
- Output coils must be preceded by an ORG address. See example 3 of figure 4.2.

- Output coils must be placed on the right-hand side of the rung. Do not program a contact to the right of an output coil. See example 4 of figure 4.2.
- Parallel branches are entered below the line containing the output coil. See figure 4.3.

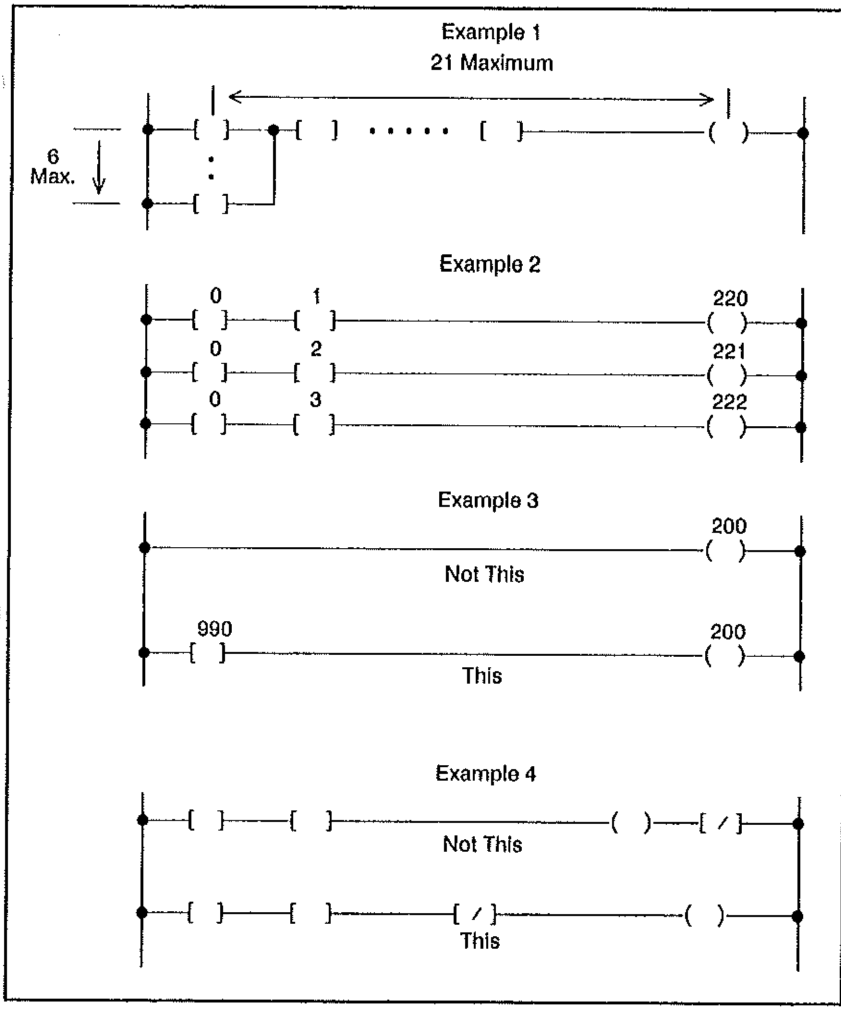


Figure 4.2 - Format Examples

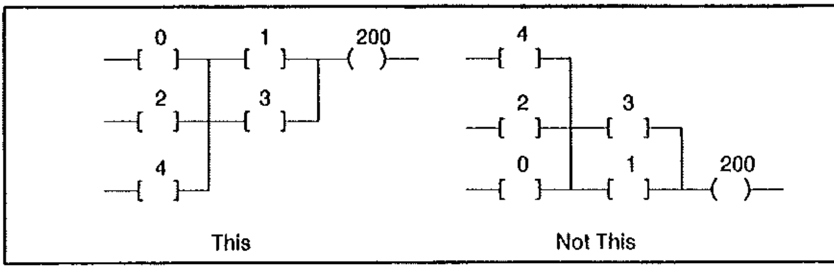


Figure 4.3 - Parallel Branch Example

- Power must flow from left to right. Circuits which allow power to flow from right to left cannot be programmed. See figure 4.4.

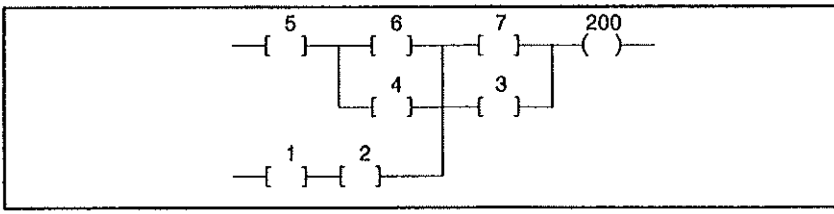


Figure 4.4 - Power Flow Example

The example in figure 4.4 is not allowed because current could flow right to left through contact 4 when contacts 1 and 2 are closed and contact 3 is open. This differs from electro-mechanical relay installations where power could flow in either direction through contact 4.

4.1.3 Order of Evaluation

The order of the ladder diagram program affects the way the program is executed. A typical scan of a program is as follows. See figure 4.5.

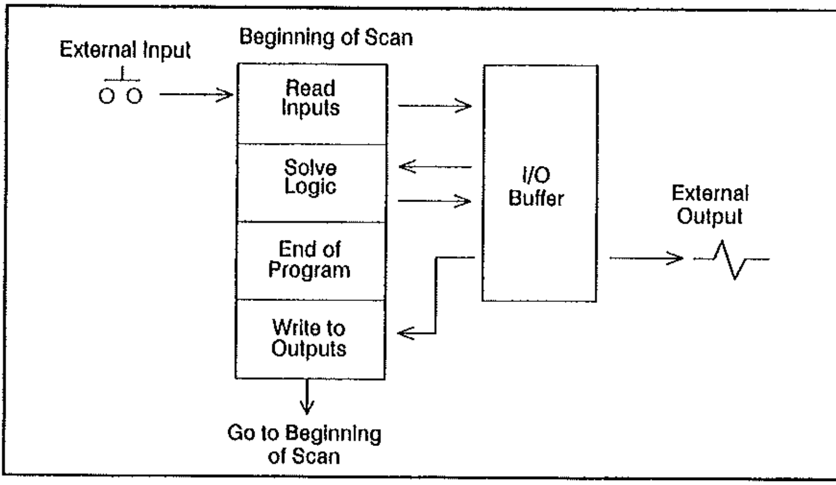


Figure 4.5 - Typical Program Scan

Data from the input modules is first read and then stored in the I/O buffer. The sequences of the program are then executed in order as they are stored in memory. As each sequence is executed, the appropriate data is read in from the I/O buffer, the logic is executed, and the output data (result) is written into the I/O buffer. When all of the sequences have been executed, the data in the I/O buffer is sent to the output modules. Data is again read in from the input modules and stored in the I/O buffer as the entire process repeats itself. Each of these cycles is referred to as a logic scan.

Note that the order of sequence execution will affect the speed of program response. See figure 4.6.

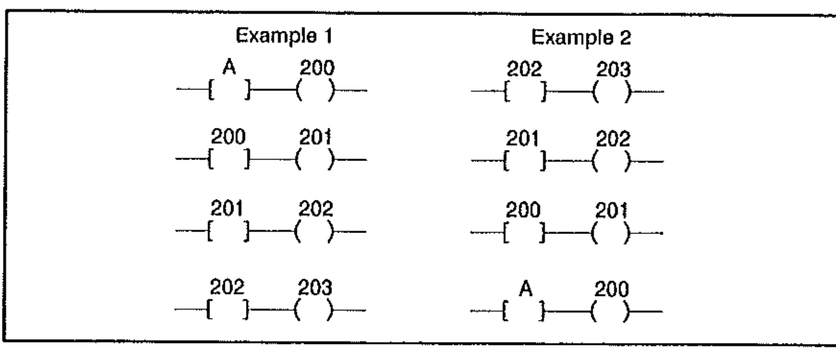


Figure 4.6 - Sequence Execution Examples 1 and 2

In example 1, of figure 4.6, if input A goes high, output 203 will go high during the same scan because each sequence reads the data from the buffer that the previous sequence just placed there.

In example 2, if A goes high, output 203 will go high four scans later because coil data is not read from the buffer until the next complete scan.

The order of sequence execution will also affect the program's logical results. See figure 4.7.

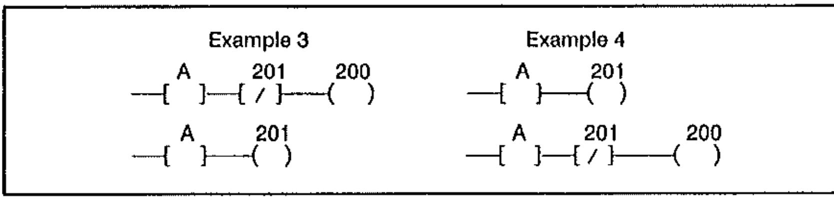


Figure 4.7 - Sequence Execution Examples 3 and 4

In example 3, when input A goes high, output 200 will go high for one scan and then go low again. This is an example of a one-shot application. Output 200 is on for one scan only. In example 4, when input A goes high, output 200 will not go high.

4.2 Digital I/O Addresses

The processor uses the following addresses when operating with digital devices. A summary of these addresses is shown in Table 4.1. The processor uses the hexadecimal (base sixteen) numbering system when addressing I/O. Internal, external, timer, and counter elements are addressed in decimal (base ten).

Table 4.1 - Digital I/O Addresses

External I/O (160 Points)					
Slot 1 Addresses		**Slot 2 Addresses**		**Slot 3 Addresses**	
Input Module	**Output Module**	**Input Module**	**Output Module**	**Input Module**	**Output Module**
0	200	20	220	40	240
1	201	21	221	41	241
2	202	22	222	42	242
3	203	23	223	43	243
4	204	24	224	44	244
5	205	25	225	45	245
6	206	26	226	46	246
7	207	27	227	47	247
8	208	28	228	48	248
9	209	29	229	49	249
10	210	30	230	50	250
11	211	31	231	51	251
12	212	32	232	52	252
13	213	33	233	53	253
14	214	34	234	54	254
15	215	35	235	55	255
Slot 4 Addresses		**Slot 5 Addresses**		**Slot 6 Addresses**	
Input Module	**Output Module**	**Input Module**	**Output Module**	**Input Module**	**Output Module**
60	260	80	280	100	300
61	261	81	281	101	301
62	262	82	282	102	302
63	263	83	283	103	303
64	264	84	284	104	304
65	265	85	285	105	305
66	266	86	286	106	306
67	267	87	287	107	307
68	268	88	288	108	308
69	269	89	289	109	309
70	270	90	290	110	310
71	271	91	291	111	311
72	272	92	292	112	312
73	273	93	293	113	313
74	274	94	294	114	314
75	275	95	295	115	315

Table 4.1 - Digital I/O Addresses (Continued)

External I/O (160 Points)					
Slot 7 Addresses		Slot 8 Addresses		Slot 9 Addresses	
Input Module	Output Module	Input Module	Output Module	Input Module	Output Module
120	320	140	340	160	360
121	321	141	341	161	361
122	322	142	342	162	362
123	323	143	343	163	363
124	324	144	344	164	364
125	325	145	345	165	365
126	326	146	346	166	366
127	327	147	347	167	367
128	328	148	348	168	368
129	329	149	349	169	369
130	330	150	350	170	370
131	331	151	351	171	371
132	332	152	352	172	372
133	333	153	353	173	373
134	334	154	354	174	374
135	335	155	355	175	375

Slot 10 Addresses	
Input Module	Output Module
180	380
181	381
182	382
183	383
184	384
185	385
186	386
187	387
188	388
189	389
190	390
191	391
192	392
193	393
194	394
195	395

Table 4.1 - Digital I/O Addresses (Continued)

Internal Non-Retentive Outputs (256 Points)					
400	446	492	538	584	633
401	447	493	539	585	634
402	448	494	540	586	635
403	449	495	541	587	636
404	450	496	542	588	637
405	451	497	543	589	638
406	452	498	544	590	639
407	453	499	545	591	640
408	454	500	546	592	641
409	455	501	547	593	642
410	456	502	548	594	643
411	457	503	549	596	644
412	458	504	550	597	645
413	459	505	551	600	646
414	460	506	552	601	647
415	461	507	553	602	648
416	462	508	554	603	649
417	463	509	555	604	650
418	464	510	556	605	651
419	465	511	557	606	652
420	466	512	558	607	653
421	467	513	559	608	654
422	468	514	560	609	655
423	469	515	561	610	
424	470	516	562	611	
425	471	517	563	612	
426	472	518	564	613	
427	473	519	565	614	
428	474	520	566	615	
429	475	521	567	616	
430	476	522	568	617	
431	477	523	569	618	
432	478	524	570	619	
433	479	525	571	620	
434	480	526	572	621	
435	481	527	573	622	
436	482	528	574	623	
437	483	529	575	624	
438	484	530	576	625	
439	485	531	577	626	
440	486	532	578	627	
441	487	533	579	628	
442	488	534	580	629	
443	489	535	581	630	
444	490	536	582	631	
445	491	537	583	632	

Table 4.1 - Digital I/O Addresses (Continued)

Internal Retentive Outputs (256 Points)					
700	746	792	838	884	933
701	747	793	839	885	934
702	748	794	840	886	935
703	749	795	841	887	936
704	750	796	842	888	937
705	751	797	843	889	938
706	752	798	844	890	939
707	753	799	845	891	940
708	754	800	846	892	941
709	755	801	847	893	942
710	756	802	848	894	943
711	757	803	849	896	944
712	758	804	850	897	945
713	759	805	851	900	946
714	760	806	852	901	947
715	761	807	853	902	948
716	762	808	854	903	949
717	763	809	855	904	950
718	764	810	856	905	951
719	765	811	857	906	952
720	766	812	858	907	953
721	767	813	859	908	954
722	768	814	860	909	955
723	769	815	861	910	
724	770	816	862	911	
725	771	817	863	912	
726	772	818	864	913	
727	773	819	865	914	
728	774	820	866	915	
729	775	821	867	916	
730	776	822	868	917	
731	777	823	869	918	
732	778	824	870	919	
733	779	825	871	920	
734	780	826	872	921	
735	781	827	873	922	
736	782	828	874	923	
737	783	829	875	924	
738	784	830	876	925	
739	785	831	877	926	
740	786	832	878	927	
741	787	833	879	928	
742	788	834	880	929	
743	789	835	881	930	
744	790	836	882	931	
745	791	837	883	932	

Table 4.1 - Digital I/O Addresses (Continued)

Internal Special Function Outputs (32 Points)						
960 961 962 963 964	965 966 967 968 969	970 971 972 973 974	975 976 977 978 979	980 981 982 983 984	985 986 987 988 989	990 991

Table 4.1 - Digital I/O Addresses (Continued)

Timers and Counters (96 Points)					
Coil Contacts		Current Values		Preset Values	
T/C 0	T/C 48	T/C 100	T/C 148	T/C 200	T/C 248
T/C 1	T/C 49	T/C 101	T/C 149	T/C 201	T/C 249
T/C 2	T/C 50	T/C 102	T/C 150	T/C 202	T/C 250
T/C 3	T/C 51	T/C 103	T/C 151	T/C 203	T/C 251
T/C 4	T/C 52	T/C 104	T/C 152	T/C 204	T/C 252
T/C 5	T/C 53	T/C 105	T/C 153	T/C 205	T/C 253
T/C 6	T/C 54	T/C 106	T/C 154	T/C 206	T/C 254
T/C 7	T/C 55	T/C 107	T/C 155	T/C 207	T/C 255
T/C 8	T/C 56	T/C 108	T/C 156	T/C 208	T/C 256
T/C 9	T/C 57	T/C 109	T/C 157	T/C 209	T/C 257
T/C 10	T/C 58	T/C 110	T/C 158	T/C 210	T/C 258
T/C 11	T/C 59	T/C 111	T/C 159	T/C 211	T/C 259
T/C 12	T/C 60	T/C 112	T/C 160	T/C 212	T/C 260
T/C 13	T/C 61	T/C 113	T/C 161	T/C 213	T/C 261
T/C 14	T/C 62	T/C 114	T/C 162	T/C 214	T/C 262
T/C 15	T/C 63	T/C 115	T/C 163	T/C 215	T/C 263
T/C 16	T/C 64	T/C 116	T/C 164	T/C 216	T/C 264
T/C 17	T/C 65	T/C 117	T/C 165	T/C 217	T/C 265
T/C 18	T/C 66	T/C 118	T/C 166	T/C 218	T/C 266
T/C 19	T/C 67	T/C 119	T/C 167	T/C 219	T/C 267
T/C 20	T/C 68	T/C 120	T/C 168	T/C 220	T/C 268
T/C 21	T/C 69	T/C 121	T/C 169	T/C 221	T/C 269
T/C 22	T/C 70	T/C 122	T/C 170	T/C 222	T/C 270
T/C 23	T/C 71	T/C 123	T/C 171	T/C 223	T/C 271
T/C 24	T/C 72	T/C 124	T/C 172	T/C 224	T/C 272
T/C 25	T/C 73	T/C 125	T/C 173	T/C 225	T/C 273
T/C 26	T/C 74	T/C 126	T/C 174	T/C 226	T/C 274
T/C 27	T/C 75	T/C 127	T/C 175	T/C 227	T/C 275
T/C 28	T/C 76	T/C 128	T/C 176	T/C 228	T/C 276
T/C 29	T/C 77	T/C 129	T/C 177	T/C 229	T/C 277
T/C 30	T/C 78	T/C 130	T/C 178	T/C 230	T/C 278
T/C 31	T/C 79	T/C 131	T/C 179	T/C 231	T/C 279
T/C 32	T/C 80	T/C 132	T/C 180	T/C 232	T/C 280
T/C 33	T/C 81	T/C 133	T/C 181	T/C 233	T/C 281
T/C 34	T/C 82	T/C 134	T/C 182	T/C 234	T/C 282
T/C 35	T/C 83	T/C 135	T/C 183	T/C 235	T/C 283
T/C 36	T/C 84	T/C 136	T/C 184	T/C 236	T/C 284
T/C 37	T/C 85	T/C 137	T/C 185	T/C 237	T/C 285
T/C 38	T/C 86	T/C 138	T/C 186	T/C 238	T/C 286
T/C 39	T/C 87	T/C 139	T/C 187	T/C 239	T/C 287
T/C 40	T/C 88	T/C 140	T/C 188	T/C 240	T/C 288
T/C 41	T/C 89	T/C 141	T/C 189	T/C 241	T/C 289
T/C 42	T/C 90	T/C 142	T/C 190	T/C 242	T/C 290
T/C 43	T/C 91	T/C 143	T/C 191	T/C 243	T/C 291
T/C 44	T/C 92	T/C 144	T/C 192	T/C 244	T/C 292
T/C 45	T/C 93	T/C 145	T/C 193	T/C 245	T/C 293
T/C 46	T/C 94	T/C 146	T/C 194	T/C 246	T/C 294
T/C 47	T/C 95	T/C 147	T/C 195	T/C 247	T/C 295

A total of 160 local external I/O are available. These may be a combination of inputs and/or outputs. If the Remote I/O modules (M/N 45C988 and 45C989) are used, an additional 112 external I/O are available. They may also be a combination of inputs and/or outputs.

4.2.1 External Inputs

External input addresses can range from 0 to 195, depending upon which slot the input module is plugged into. A maximum of 160 local external input points are available.

4.2.2 External Outputs

External output addresses can range from 200 to 395, depending upon which slot the output module is plugged into. A maximum of 160 local external outputs are available.

If an output module is not used in the assigned slot, the addresses can be used as internal outputs. For example, if slot 1 holds an input module instead of an output module, addresses 200 to 219 can be used as internal outputs.

4.2.3 Internal Outputs

There are three types of internal outputs: non-retentive, retentive, and special function.

Non-retentive internal outputs are reset to zero when incoming power is removed. A total of 256 non-retentive outputs are available with addresses ranging from 400 to 655.

Retentive internal outputs retain their current status when incoming power is removed. A total of 256 retentive outputs are available with addresses ranging from 700 to 955. Internal outputs 940 to 955 are reserved for use with the real time clock and are described below.

Internal outputs 960 to 991 are reserved for special functions and are described below.

4.2.3.1 Real-Time Clock (Addresses 940 - 953)

Addresses 940 through 953 allow access to the real-time clock. This clock is programmable for the year, month, day, day of the week, hour, minute, and second. See figure 4.8. Accuracy is +/– 30 seconds per month. Clock values are maintained regardless of the state of the processor or power supply.

Address contents for March 5, 1991 (09:07:36)

Address	Contents
940 (years)	0 0 9 1
942 (month & day)	0 3 0 5
944 (day of the week)	0 0 0 3
946 (hour & minute)	0 9 0 7
948 (seconds)	0 0 3 6

Figure 4.8 - Real-Time Clock Addresses

The lower two digits of address 940 contain the last two digits of the year. The upper two digits of address 940 always equal 00H.

The upper two digits of address 942 contain the month (01H - 12H) while the lower two digits contain the day (01H - 31H).

The lower two digits of address 944 contain the day of the week: Sunday (01H), Monday (02H), Tuesday (03H), Wednesday (04H), Thursday (05H), Friday (06H), Saturday (07H). The upper two digits of address 944 always equal 00H.

The upper two digits of address 946 contain the hour (00H - 24H) while the lower two digits contain the number of minutes (00H - 59H).

The lower two digits of address 948 contain the number of seconds (00H - 59H). The upper two digits of address 948 always equal 00H.

Addresses 940 through 948 either contain the current clock values or they function as normal internal outputs depending upon the state of address 950. When address 950 equals zero, the real-time clock's internal registers constantly update the status of registers 940 to 948. When address 950 equals one, the real-time clock's internal registers continue to operate but do not update registers 940 to 948.

When the state of address 951 changes from zero to one, the contents of addresses 940 to 948 are transferred to the real-time clock's internal registers. Address 950 is then reset to zero.

Address 952 provides a +/− 30 second timing adjustment. When the state of address 952 changes from zero to one, the number of seconds is set to 0, if the current number of seconds is from 0 to 29. The number of seconds is set to 59, if the current number of seconds is from 30 to 59. The time adjustment is first made in the clock's internal registers and then is reflected in the values displayed in addresses 940 through 948 if address 950 is equal to zero.

Address 953 is equal to one when a low voltage condition is detected.

Addresses 950, 951, and 952 are reset whenever power is cycled.

Refer to section 5.11 for instructions on setting the real-time clock.

4.2.3.2 Address 960

When address location 960 is equal to one, all outputs are automatically reset to zero. Use this address to generate logic that detects a malfunction and stops program execution. See figure 4.9. The programmer identifies the address with: 960 E. If a malfunction occurs, correct the problem and cycle power.

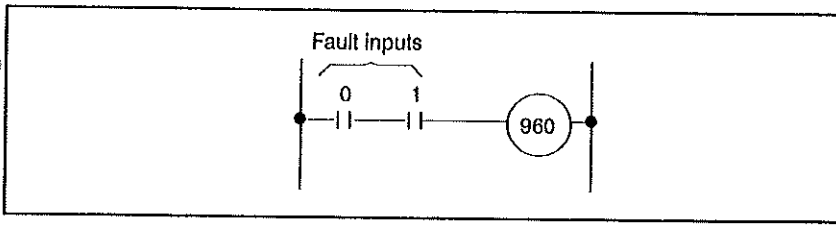


Figure 4.9 – Example of Address 960 Programming

4.2.3.3 Address 961

When address location 961 is equal to one, the retentive data registers (retentive outputs, counters, and shift registers) are reset to zero. Note that retentive data will not be reset if address 961 is set to one after the first program scan. Use this address in conjunction with address 967, which turns on for one scan when program execution begins. As shown in figure 4.10, counters 50 and 51 are reset to zero once the program begins executing.

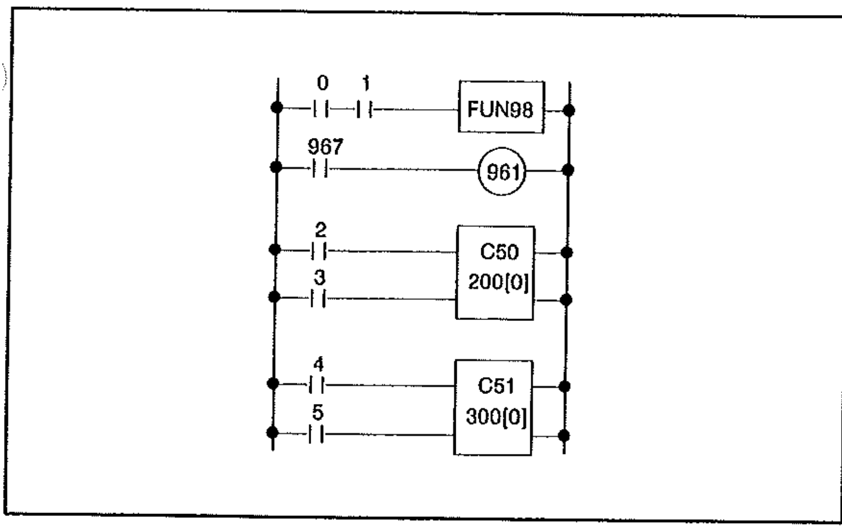


Figure 4.10 - Example of Address 961 Programming

4.2.3.4 Address 962

Address location 962 is turned off and on each time one program scan is completed (every other scan). See figure 4.11.

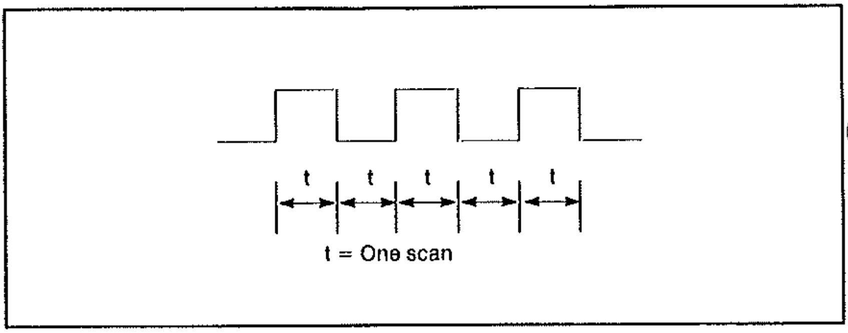


Figure 4.11 - Example of Address 962 Timing

4.2.3.5 Address 963

Address location 963 provides a 100 millisecond clock signal as shown in figure 4.12.

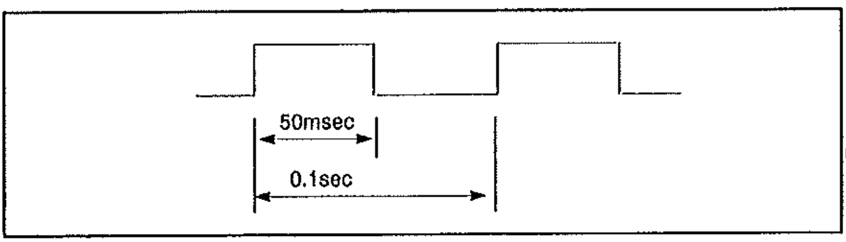


Figure 4.12 - Example of Address 963 Timing

4.2.3.6 Address 964

Address location 964 provides a 1 second clock signal as shown in figure 4.13.

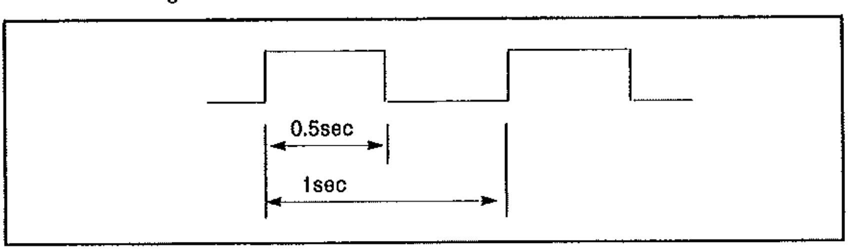


Figure 4.13 - Example of Address 964 Timing

4.2.3.7 Address 965

Address location 965 provides a 10 second clock signal as shown in figure 4.14.

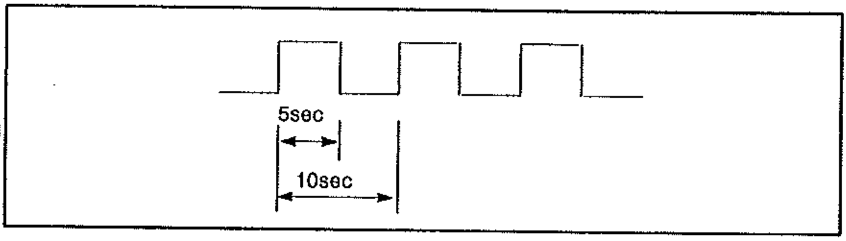


Figure 4.14 - Example of Address 965 Timing

4.2.3.8 Address 966

Address location 966 provides a 1 minute clock signal as shown in figure 4.15.

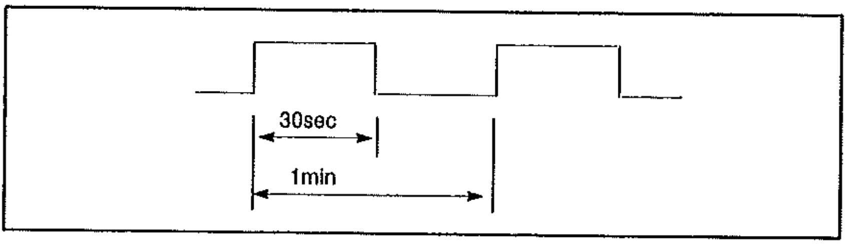


Figure 4.15 - Example of Address 966 Timing

4.2.3.9 Address 967

Address location 967 is turned on for one scan when the program begins executing. See figure 4.16. Use this address in conjuction with address 961 to reset retentive coils, timers, and counters.

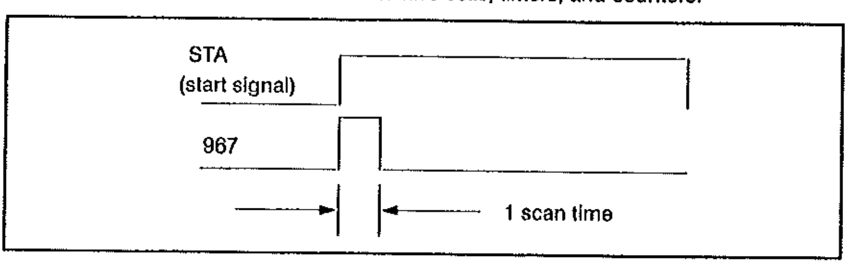


Figure 4.16 - Example of Address 967 Timing

4.2.3.10 Address 968

Address location 968 is turned on and off once every 1,000 program scans as shown in figure 4.17.

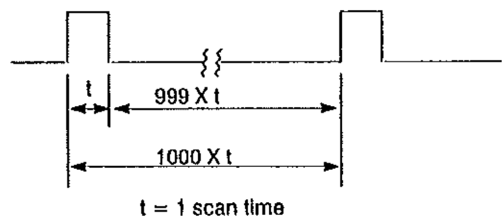


Figure 4.17 - Example of Address 968 Timing

4.2.3.11 Address 969

Address location 969 provides a 10 millisecond clock signal as shown in figure 4.18.

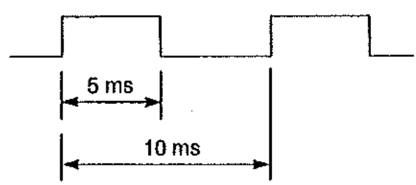


Figure 4.18 - Example of Address 969 Timing

4.2.3.12 Address 970

If the processor's ERR LED is on, a system error has occurred. Address location 970 contains a system error code from 0 to 65535. See Table 7.4 for a listing of the system error codes.

4.3.2.13 Address 971

This address is reserved for use by the processor.

4.2.3.14 Address 972

If the processor's ERR LED is on, a system error has occurred. Address location 972 contains the number of the instruction that was active when the error occured.

4.2.3.15 Addresses 973-979

These addresses are reserved for use by the processor.

4.2.3.16 Address 980

When a syntax error occurs, this address location contains an error code from 0 to 65535. See Table 7.1 for a listing of the error codes.

4.2.3.17 Address 981

This address is reserved for use by the processor.

4.2.3.18 Address 982

This address location contains the latest scan time in increments of 10 milliseconds.

4.2.3.19 Address 983

This address is reserved for use by the processor.

4.2.3.20 Address 984

This address location contains the maximum scan time in increments of 10 milliseconds.

4.2.3.21 Addresses 985-988

These addresses are not used.

4.2.3.22 Address 989

This address location provides system attribute information in bits 5, 6, and 7. See figure 4.19.

Bit 5 =	0 = Processor cannot go into run (programming error). 1 = Processor can go into run.
Bit 6 =	0 = Serial port baud rate is 9600 bps. 1 = Serial port baud rate is 4800 bps.
Bit 7 =	0 = Not defined. 1 = Shark XL II processor.

Figure 4.19 - Address 989 System Attribute Information

4.2.3.23 Address 990

This address location is always equal to one.

4.2.3.24 Address 991

This address location is equal to one whenever the processor is in the run mode.

4.2.4 Timers

On-delay timer addresses range from T/C 0 to T/C 95 for a maximum of 96 timers. The timers are self-resetting and are non-retentive. When a timer's current value counts up to the preset value, the timer's contacts are energized. Note that timers and counters share the same address range.

Adding 100 to the timer's address (T/C 100 to T/C 195) will give the address of the timer's current value. Adding 200 to the timer's address (T/C 200 to T/C 295) will give the address of the timer's preset value. Current and preset timer values are stored as 16-bit BCD words.

4.2.5 Counters

Two types of counters are available.

Standard counters are up-counters with a reset input. These counter addresses range from T/C 0 to T/C 95, for a maximum of 96 counters. When a counter's current value reaches the preset value, the counter's contacts are energized. Note that counters and timers share the same address range.

Adding 100 to the counter's address (T/C 100 to T/C 195) will give the address of the counter's current value. Adding 200 to the counter's address (T/C 200 to T/C 295) will give the address of the counter's preset value. Current and preset counter values are stored as 16-bit words.

The up/down counter (FUN40) uses an internal coil and counts either up or down in four-digit binary-coded-decimal (BCD).

4.3 Non-Digital I/O Addresses

The processor uses the following addresses when operating with non-digital devices. A summary of these addresses are shown in Table 4.2. The processor uses the hexadecimal (base sixteen) numbering system when addressing analog module, counter module, network module, or remote I/O module data.

Table 4.2 - Non-Digital I/O Addresses

Signal Type		Addresses: Analog Input Module / Counter Module Status / Network Module Input / Remote Module Input	Addresses: Analog Output Module / Counter Module Control / Network Module Output / Remote Module Output
Slot 1	Ch. 0	0	200
	Ch. 1	2	202
	Ch. 2	4	204
	Ch. 3	6	206
	Ch. 4	8	208
	Ch. 5	10	210
	Ch. 6	12	212
	Ch. 7	14	214
Slot 2	Ch. 0	20	220
	Ch. 1	22	222
	Ch. 2	24	224
	Ch. 3	26	226
	Ch. 4	28	228
	Ch. 5	30	230
	Ch. 6	32	232
	Ch. 7	34	234
Slot 3	Ch. 0	40	240
	Ch. 1	42	242
	Ch. 2	44	244
	Ch. 3	46	246
	Ch. 4	48	248
	Ch. 5	50	250
	Ch. 6	52	252
	Ch. 7	54	254

Table 4.2 - Non-Digital I/O Addresses (Continued)

Signal Type		Addresses	
		Analog Input Module Counter Module Status Network Module Input Remote Module Input	Analog Output Module Counter Module Control Network Module Output Remote Module Output
Slot 4	Ch. 0	60	260
	Ch. 1	62	262
	Ch. 2	64	264
	Ch. 3	66	266
	Ch. 4	68	268
	Ch. 5	70	270
	Ch. 6	72	272
	Ch. 7	74	274
Slot 5	Ch. 0	80	280
	Ch. 1	82	282
	Ch. 2	84	284
	Ch. 3	86	286
	Ch. 4	88	288
	Ch. 5	90	290
	Ch. 6	92	292
	Ch. 7	94	294
Slot 6	Ch. 0	100	300
	Ch. 1	102	302
	Ch. 2	104	304
	Ch. 3	106	306
	Ch. 4	108	308
	Ch. 5	110	310
	Ch. 6	112	312
	Ch. 7	114	314
Slot 7	Ch. 0	120	320
	Ch. 1	122	322
	Ch. 2	124	324
	Ch. 3	126	326
	Ch. 4	128	328
	Ch. 5	130	330
	Ch. 6	132	332
	Ch. 7	134	334

Table 4.2 - Non-Digital I/O Addresses (Continued)

Signal Type		Addresses: Analog Input Module, Counter Module Status, Network Module Input, Remote Module Input	Addresses: Analog Output Module, Counter Module Control, Network Module Output, Remote Module Output
Slot 8	Ch. 0	140	340
	Ch. 1	142	342
	Ch. 2	144	344
	Ch. 3	146	346
	Ch. 4	148	348
	Ch. 5	150	350
	Ch. 6	152	352
	Ch. 7	154	354
Slot 9	Ch. 0	160	360
	Ch. 1	162	362
	Ch. 2	164	364
	Ch. 3	166	366
	Ch. 4	168	368
	Ch. 5	170	370
	Ch. 6	172	372
	Ch. 7	174	374
Slot 10	Ch. 0	180	380
	Ch. 1	182	382
	Ch. 2	184	384
	Ch. 3	186	386
	Ch. 4	188	388
	Ch. 5	190	390
	Ch. 6	192	392
	Ch. 7	194	394

4.3.1 Non-Digital Inputs

Non-digital input addresses can range from 0 to 194, depending upon which slot the input module is plugged into.

4.3.2 Non-Digital Outputs

Non-digital output addresses can range from 200 to 394, depending upon which slot the output module is plugged into.

4.4 I/O Data Handling

I/O data is stored and accessed in different forms depending upon the command or function being used. See Table 4.3.

All addresses have an 8-bit (byte) memory location. External input and output data is stored in 8-bit bytes. Internal output, timer, counter, coil, and contact data is stored in 1-bit data areas. Timer and counter preset and current values are stored in 16-bit words (two 8-bit bytes). See figure 4.20.

Four different commands are used to access the I/O data stored in memory:

- bit processing commands
- word processing commands
- commands that process bit data as word data
- byte processing commands

Bit processing commands only access bit 7 (MSB) of external input and output data, internal output coil data, and timer/counter coil data. See figure 4.21.

Word processing commands access 16 bits (two consecutive 8-bit bytes) of analog module data, counter module data, network module data, and remote I/O module data. Word processing commands can also access constant value data, internal output data, timer/counter preset and current value data. See figure 4.22.

Commands that process bit data as word data use bit 7 (MSB) of 16 consecutive 8-bit bytes. See figure 4.23. Use this type of command when BCD data is being loaded in from an external input or when 16 bits of data in the arithmetic register are being written to an external output.

Byte processing commands access eight bits of data and can be used with eight bit analog signals. Byte processing commands can also access eight bits of constant value data, internal output data, and timer/counter preset and current value data. See figure 4.24.

Table 4.3 - I/O Data Processing Commands

I/O / Command	External I/O		Internal Outputs		Special function internal output	Timer/ Counter		Constant Value
	Bit data (b7)	Word data	Bit data (b7)	Word data		Coil/ Contact	Current and Preset Values	
Bit Processing Command	●		●		●	●		
Word Processing Command		●*		●			●	●
Command that Processes Bit Data as Word Data	●		●					
Byte Processing Command		●**		●+			●+	●+

* Word processing commands are only used with external I/O that are connected to analog modules, counter modules, remote I/O modules, and network modules.

** Byte processing commands are only used with external I/O that have 8 bits of data resolution.

\+ Byte processing commands only access 8 bits of internal output data, timer/counter values, and constant values.

External/Internal I/O Addresses

0	b7	b6	b5	b4	b3	b2	b1	b0
1	b7	b6	b5	b4	b3	b2	b1	b0
2	b7	b6	b5	b4	b3	b2	b1	b0
3	b7	b6	b5	b4	b3	b2	b1	b0
4	b7	b6	b5	b4	b3	b2	b1	b0
5	b7	b6	b5	b4	b3	b2	b1	b0
6	b7	b6	b5	b4	b3	b2	b1	b0
7	b7	b6	b5	b4	b3	b2	b1	b0
8	b7	b6	b5	b4	b3	b2	b1	b0
9	b7	b6	b5	b4	b3	b2	b1	b0
10	b7	b6	b5	b4	b3	b2	b1	b0
11	b7	b6	b5	b4	b3	b2	b1	b0
12	b7	b6	b5	b4	b3	b2	b1	b0
13	b7	b6	b5	b4	b3	b2	b1	b0
14	b7	b6	b5	b4	b3	b2	b1	b0
15	b7	b6	b5	b4	b3	b2	b1	b0

Bit processing commands will only access bit 7 from addresses 0 to 15. Word processing commands will access two (8-bit) bytes (16 bits total). Byte processing commands will access one (8-bit) byte.

Internal Special Function Output Addresses

960	b7	-	-	-	-	-	-	-
961	b7	-	-	-	-	-	-	-
962	b7	-	-	-	-	-	-	-
963	b7	-	-	-	-	-	-	-
	↓							
991	b7	-	-	-	-	-	-	-

Bit processing commands will only access bit 7(MSB) from addresses 960 to 991. Bits 0 to 6 should not be used.

Timer/Counter Preset Value Addresses

T/C200	b15	b14	b13	b12	b11	b10	b9	b8
	b7	b6	b5	b4	b3	b2	b1	b0
T/C201	b15	b14	b13	b12	b11	b10	b9	b8
	b7	b6	b5	b4	b3	b2	b1	b0
	↓							
T/C295	b15	b14	b13	b12	b11	b10	b9	b8
	b7	b6	b5	b4	b3	b2	b1	b0

Word processing commands will access two (8-bit) bytes (16 bits) for each timer/counter. Byte processing commands will only access the lower 8 bits (b0-b7) of each timer/counter value.

Timer/Counter Coil/Contact Addresses

T/C0	b7	-	-	-	-	-	-	-
T/C1	b7	-	-	-	-	-	-	-
	↓							
T/C95	b7	-	-	-	-	-	-	-

Bit processing commands will access bit 7(MSB) from timer/counter addresses 0 to 95. Bits 0 to 6 should not be used.

Timer/Counter Current Value Addresses

T/C100	b15	b14	b13	b12	b11	b10	b9	b8
	b7	b6	b5	b4	b3	b2	b1	b0
T/C101	b15	b14	b13	b12	b11	b10	b9	b8
	b7	b6	b5	b4	b3	b2	b1	b0
	↓							
T/C195	b15	b14	b13	b12	b11	b10	b9	b8
	b7	b6	b5	b4	b3	b2	b1	b0

Word processing commands will access two (8-bit) bytes (16 bits) for each timer/counter. Byte processing commands will only access the lower 8 bits (b0-b7) of each timer/counter value.

Figure 4.20 - I/O Data Storage in Memory

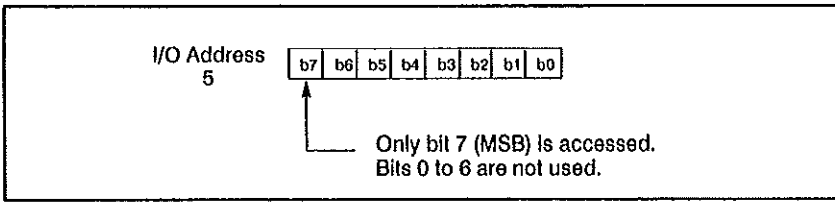


Figure 4.21 - Bit Processing

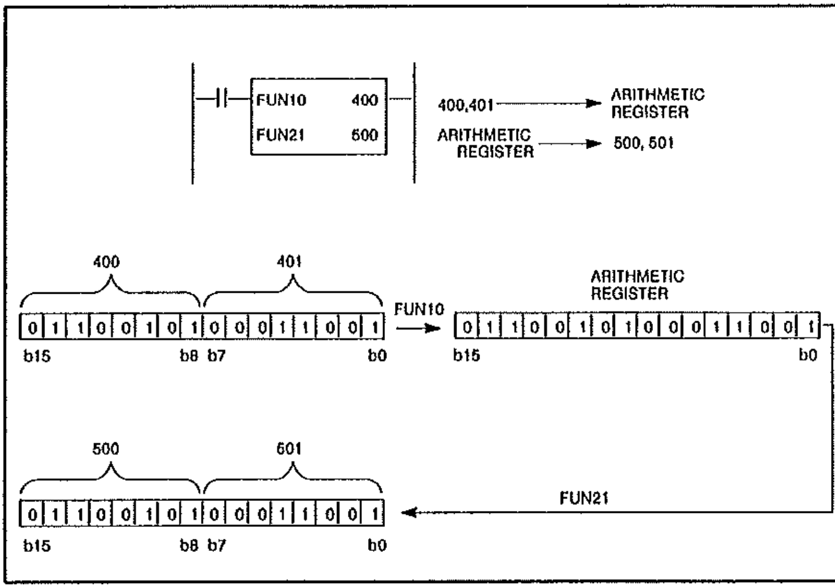


Figure 4.22 - Word Processing

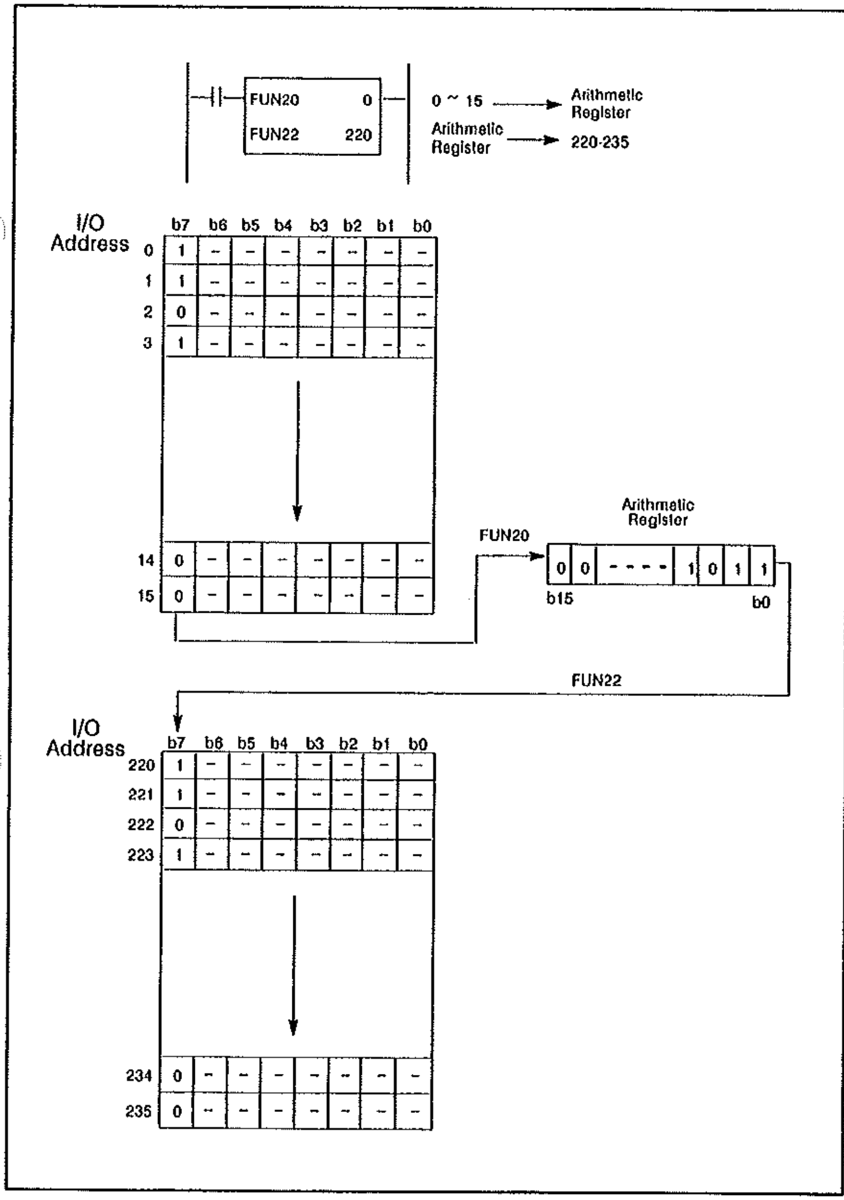


Figure 4.23 - Processing Bits as Words

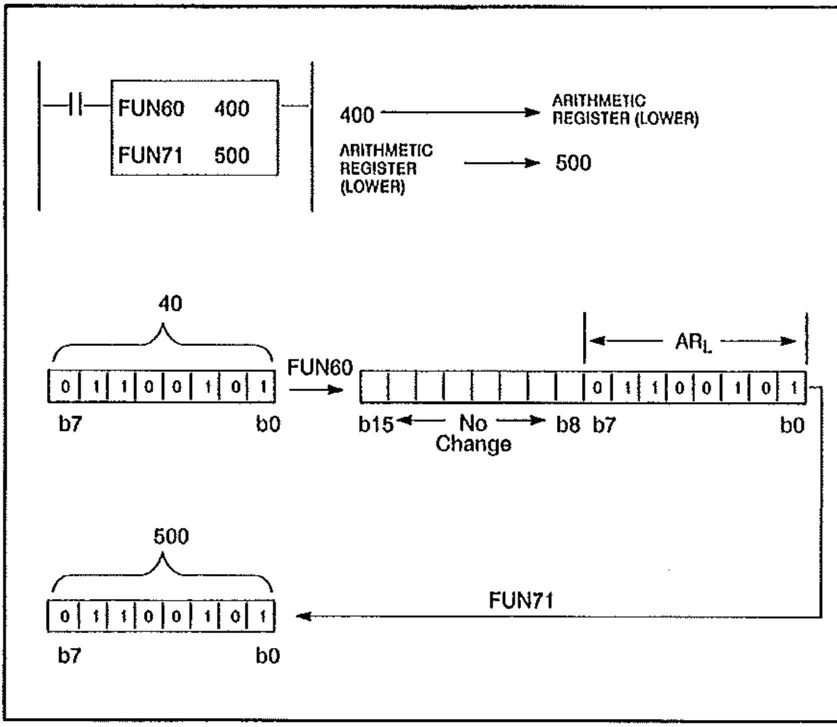


Figure 4.24 - Byte Processing

5.0 OPERATING PROCEDURES

This chapter tells you how to operate your programmable controller using the handheld programmers. The chapter covers how to create, enter, and edit a program. It explains how to monitor and force outputs. It also tells you how to save and load programs from cassette tape.

Refer to the Reliance Shark Programming Executive Manual (J-3804) if you are using the SPX software in place of the handheld programmers.

> **WARNING**
>
> **ONLY QUALIFIED PERSONNEL MAY INSTALL, ADJUST, OPERATE, AND MAINTAIN THIS EQUIPMENT. READ AND UNDERSTAND THIS INSTRUCTION MANUAL IN ITS ENTIRETY BEFORE ANY WORK IS PERFORMED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.**

5.1 Operating Guidelines

Use the following guidelines as you prepare your program:

1. Clear memory before you begin a new program.
2. Write and then key-in the program.
3. Check the program.
   - Be sure each rung has been entered correctly.
4. Search for the address if a programming error has been made.
5. Edit the rung as necessary.
6. Check the program's syntax.
   - Correct an error if present.
7. Execute the program in test mode using a handheld programmer.
   - Use the test mode and output forcing to verify operation.

> **WARNING**
>
> **EXECUTING A PROGRAM IN TEST MODE COULD RESULT IN UNEXPECTED MACHINE MOTION. STATION A PERSON AT THE MACHINE NEXT TO THE E-STOP BUTTON TO TERMINATE UNEXPECTED RESULTS. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.**

8. Run the program.
   - Run the program once all wiring and programming have been corrected.

Table 5.1 provides a summary of the operating guidelines for the handheld programmers only. A detailed listing of the operating guidelines listed above is found after the table.

Table 5.1 - Summary of Operating Procedures

No.	Operation		Key-in Procedure	Contents of Display: Step No.	Contents of Display: Data – Data	Contents of Display: Data – _	Contents of Display: Set Value	Contents of Display: Current Value	Contents of Display: Conductivity	Contents of Display: DATA	Contents of Display: STEP	Mode Switch: PROG	Mode Switch: TEST	Mode Switch: RUN	Processor Running	Processor Stopped
1	Clear memory		[CLR] [ENT] [DEL]			○				○		○	×	×	×	○
2	Enter a new sequence		[CLR] [ENT] [DEL] → [Generation of program for each step] [ENT] → Next Sequence (loop back)		○		○			○		○	×	×	×	○
	Insert a sequence		[CLR] [·/STEP] → [Generation of program for each step] [ENT] → Next Sequence (loop back)		○		○			○		○	×	×	×	○
3	Display Mode	Starting from step 000	[CLR] [STEP +], then [STEP +] OR [STEP −] (1 step forward or backward)		○		○		○	○		○	○	○	○	○
	Display Mode	Starting from a specified step #	[CLR] [Step No.] [·/STEP], then [STEP +] OR [STEP −] (1 step forward or backward)		○		○		○	○		○	○	○	×	○
	Display Mode	Searching from I/O commands	[CLR] [I/O number or command] [SRC], then [STEP +] OR [STEP −] (1 step forward or backward)		○		○		○	○		○	○	○	○	○
	Display Mode	First available memory location	[CLR] [·/STEP], then [STEP +] OR [STEP −] (1 step forward or backward)			○				○		○	○	○	○	○
	Toggle between the data display and the step display		[Read-out] [·/STEP] (Data step display can be switched over with the [·/STEP] key)	○–○ (toggle with Data)	○–○ (toggle with Step No.)		○			○–○ (toggle with STEP)	○–○ (toggle with DATA)	○	○	○	○	○

○ — [illegible] Operation   × — Not Valid   ○–○ — [illegible]ggle

Table 6.1 - Summary of Operating Procedures (Continued)

No.	Mode	Operation		Key-in Procedure	Step No.	Contents of Display: Data: Data	Contents of Display: Data: —	Contents of Display: Set Value	Contents of Display: Current Value	Contents of Display: Conductivity	Contents of Display: DATA	Contents of Display: STEP	Mode Switch: PROG	Mode Switch: TEST	Mode Switch: RUN	Processor Running	Processor Stopped
4	Search Mode	Contact	External input/ output, internal output timer, counter	[CLR] [Input/output no.] [SRC] / [CLR] [Timer/counter no.] [SRC]		○		○		○	○		○	○	○	○	○
		Coil	External output, internal output timer, counter	[CLR] [Input/output no.] [SRC] / [CLR] [Timer/counter no.] [SRC]		○		○		○	○		○	○	○	○	○
		Command	Basic and application commands	[CLR] [Basic command] [SRC] / [CLR] [Application command] [SRC]		○		○		○	○		○	○	○	○	○
				then: [STEP +] or [STEP −] (1 step forward or backward), or [SRC] (Search for step in which searched data is written.)													
5	Edit Mode	Insert		[Read program to be inserted] [DCLR] [Generation of step to be deleted] [INS]		○		○			○		○	×	×	×	○
		Delete		[Read step to be deleted] [DEL]		○		○			○		○	×	×	×	○
		Change		[Read step to be changed] [DCLR] [Generation of program to be changed] [ENT]		○		○			○		○	×	×	×	○

Table 5.1 - Summary of Operating Procedures (Continued)

No.	Operation			Key-in Procedure	Step No.	Contents of Display: Data: Data	Contents of Display: Data: —	Contents of Display: Set Value	Contents of Display: Current Value	Contents of Display: Conductivity	Contents of Display: DATA	Contents of Display: STEP	Mode Switch: PROG	Mode Switch: TEST	Mode Switch: RUN	Processor Running	Processor Stopped
6	Monitoring	Contact	External input/ output, internal input/output, timer, counter	[CLR] [Input/output no.] [MON] → [STEP +] or [STEP −]		○				○	○		○	○	○	○	○
				[CLR] [T/C] [Timer/counter no.] [MON] → [STEP +] or [STEP −]		○				○	○		○	○	○	○	○
		Coil	External output, internal output	[CLR] [OUT] [Input/output no.] [MON] → [STEP +] or [STEP −]		○				○	○		○	○	○	○	○
				[CLR] [OUT] [T/C] [Timer/counter no.] [MON] → [STEP +] or [STEP −]		○			○		○		○	○	○	○	○
7	Check Syntax			[CLR] [SRC]	○							○	○	×	×	×	○

Table 5.1 - Summary of Operating Procedures (Continued)

No.	Operation			Key-in Procedure	Step No.	Contents of Display: Data: Data	Contents of Display: Data: —	Contents of Display: Set Value	Contents of Display: Current Value	Contents of Display: Conductivity	Contents of Display: DATA	Contents of Display: STEP	Mode Switch: PROG	Mode Switch: TEST	Mode Switch: RUN	Processor Running	Processor Stopped
8	Forcing	Forced output	External output	CLR SET SET ENT FUN 3 OUT [External output no.] SET or RES		○				○	○		×	○	×	×	○
		Forced set/ reset	Internal output protected from power failure	CLR OUT [Input/output no.] MON SET or RES		○				○	○		×	○	○	○	×
			Timer/ counter	CLR OUT T/C [Timer/counter no.] MON SET or RES		○				○	○		×	○	○	○	×

WARNING

FORCING OUTPUTS CAN RESULT IN UNEXPECTED MACHINE MOTION. ALL PERSONNEL MUST BE CLEAR OF THE MACHINE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

5.2 Clearing Memory

Be sure to clear memory before you begin entering a new program. Use the procedure in figure 5.1 to clear memory. This procedure deletes the program currently stored there. It takes four seconds to clear the standard 926-word memory. It takes 19 seconds to clear the 3997-word memory.

A clear memory operation will also do the following:

- clears timers/counters
- resets shift registers
- resets retentive contacts

Operation	Mode Switch Setting	Processor Status
Clear Memory	PROG	Stop

Key Stroke	Type of Display			Remarks
	Command	Numerical Display	Mode display	
CLR			• PROG • DATA	
ENT		E		
DEL		_		Memory deleted

Figure 5.1 - Clearing Memory

When using a Standard Programmer, normally the programmer's display shows the contents of a memory location. Use the procedure in figure 5.2 to toggle between the display of data and the current memory location.

Key Stroke	Type of Display			Remarks
	Command	Numerical Display	Mode display	
CLR ENT DEL		_	• PROG • DATA	Data Display
·/ STEP		0	• PROG • STEP	Step No. Display
·/ STEP		_	• PROG • DATA	Data Display

Figure 5.2 - Displaying Data or a Memory Location

5.3 Entering a Program Using the Handheld Programmers

When entering a program with the handheld programmers, you must begin each program with the sequence shown in figure 5.3. You need to assign the two contacts to either external inputs or internal outputs or both. If you have only one external input, you can program the other as internal output 990, which is always energized. Obviously, you should not program both contacts as internal output 990 as the processor will stay in the run mode continuously.

Note that the processor cannot go into the run mode unless the start sequence (FUN 98) is programmed as the first sequence in the program.

When both inputs are energized, FUN98 becomes energized, and the processor goes into run if the handheld programmer is not attached. If the handheld programmer is attached, the mode selector switch must be in the RUN position for the processor to go into run.

To take the processor out of the run mode, de-energize the inputs. Note that if the hand-held programmer is attached, moving the mode selector switch out of the RUN position will not take the processor out of the run mode by itself. The inputs to FUN98 must be de-energized to stop the processor.

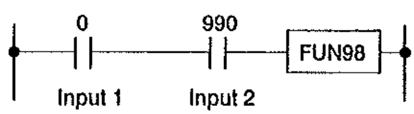


Figure 5.3 - Required Start Sequence

Figure 5.4 shows how to enter a new program step by step.

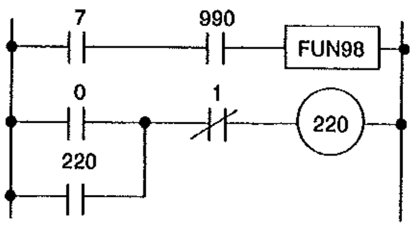
When you press the [ENT] key, the data displayed is stored in memory. The programmer's display then shows the contents of the next memory location.

The display's contents in figure 5.4 are the values present before you press the [ENT] key.

Refer to the Standard Programmer Instruction Manual (J-3803) or the Universal Programmer Instruction Manual (J-3800) for additional information.

Operation	Mode Switch Setting	Processor Status
Creating a new program	PROG	Stop

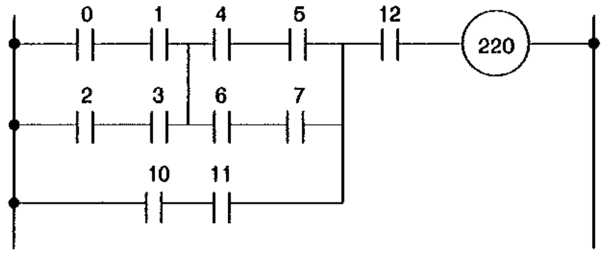
Example Sequence



Key-in Procedure	Type of Display: Command	Type of Display: Numerical Display	Type of Display: Mode display
CLR ENT DEL		–	• PROG • DATA
ORG 7 ENT	• ORG	*7*	
AND 9 9 0 ENT	• AND	*990*	
FUN 9 8 ENT	• FUN	*98*	
ORG 0 ENT	• ORG	*0*	
OR 2 2 0 ENT	• OR	*220*	
AND NOT 1 ENT	• AND • NOT	*1*	
OUT 2 2 0 ENT	• OUT	*220*	

Figure 5.4 - Entering a New Program

A more complex example is shown in figure 5.5.



Key-in Procedure		
ORG	0	ENT
AND	1	ENT
STR	2	ENT
AND	3	ENT
OR	STR	ENT
STR	4	ENT
AND	5	ENT
STR	6	ENT
AND	7	ENT
OR	STR	ENT
AND	STR	ENT
STR	10	ENT
AND	11	ENT
OR	STR	ENT
AND	12	ENT
OUT	220	ENT

Order of Programming (See Figure 5.6)

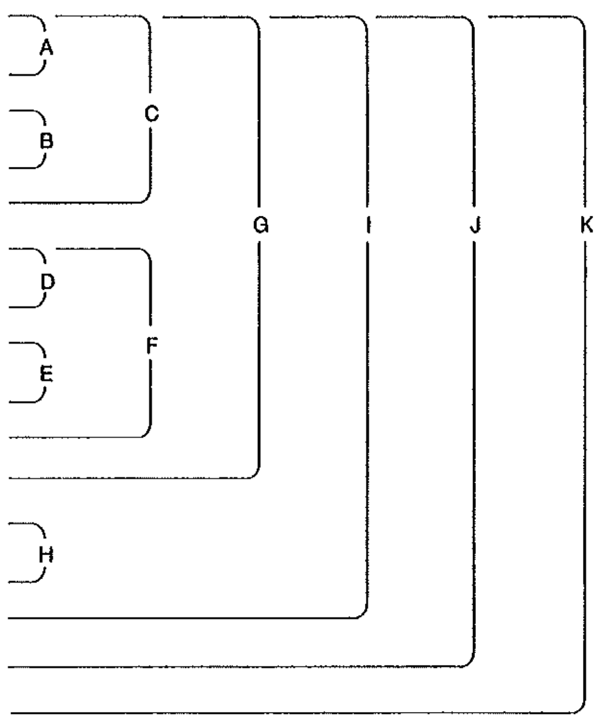


Figure 5.5 - Programming Example

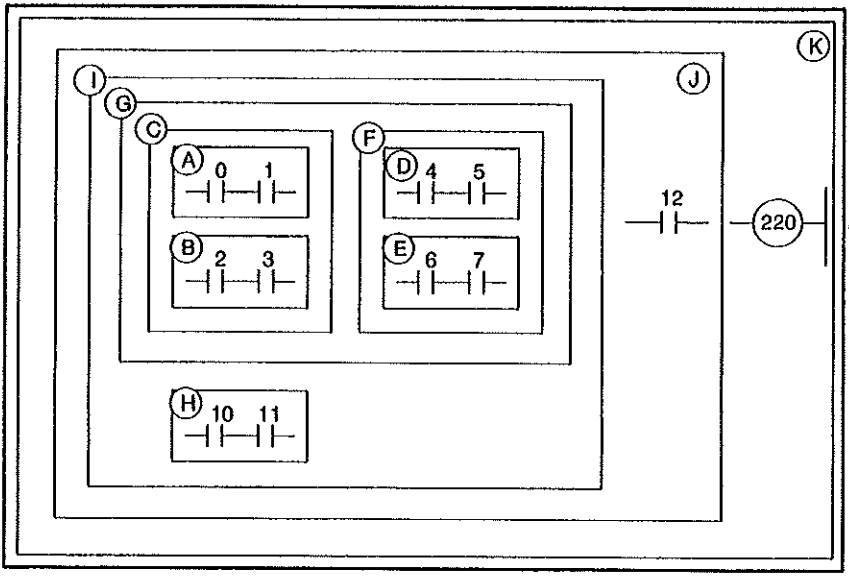


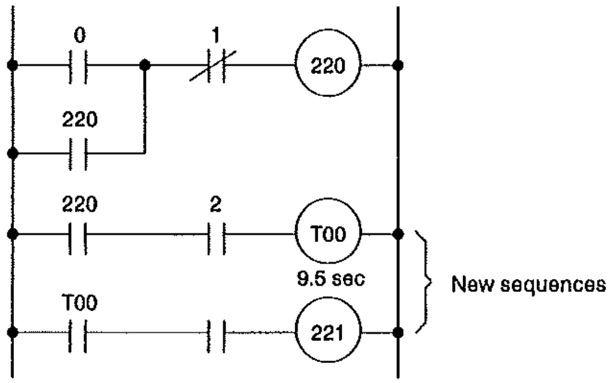
Figure 5.6 - Order of Programming for the Example in Figure 5.5

5.3.1 Adding Rungs

Use the procedure in figure 5.7 to add additional rungs to the end of your program. When you press the [CLR] [./STEP] keys, the programmer displays the first available unused memory location.

Operation	Mode Switch Setting	Processor Status
Add a new sequence	PROG	Stop

Sample Sequence



Key-in Procedure	Type of Display: Command	Type of Display: Numerical Display	Type of Display: Mode Display
[CLR] [./STEP]		–	
[ORG] [2] [2] [0] [ENT]	• ORG	*220*	
[AND] [2] [ENT]	• AND	*2*	• PROG
[OUT] [T/C] [0] [./STEP] [9] [./STEP] [5] [ENT]	• OUT • T/C	*0.9.5*	• DATA
[ORG] [T/C] [0] [ENT]	• ORG • T/C	*0*	
[OUT] [2] [2] [1] [ENT]	• OUT	*221*	

Figure 5.7 - Entering Additional Rungs

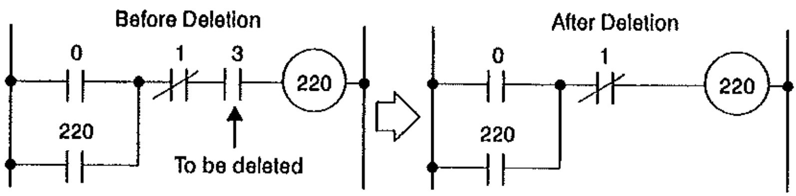
5.3.2 Deleting an Element or Rung

To delete a rung's elements use the procedure in figure 5.8.

When you press the [DEL] key, the current memory location is cleared and all following steps are moved to compress memory. The programmer then displays the next memory location. The programmer will sequentially clear memory locations each time you press the [DEL] key.

Operation	Mode Switch Setting	Processor Status
Deletion of an Element	PROG	Stop

Example Sequence



Key-in Procedure	Type of Display		
	Command	Numerical Display	Mode Display
CLR 3 SRC	• AND	*003*	• DATA
DEL	• OUT	*220*	• PROG

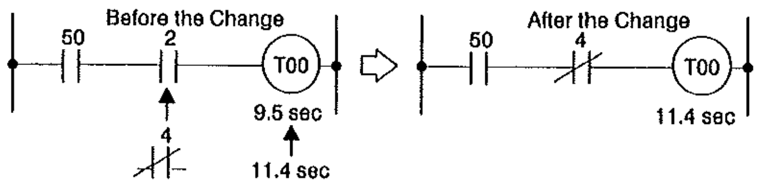
Figure 5.8 - Deleting an Element

5.3.3 Editing a Program

Use the procedure in figure 5.9 to change an element within a rung. With the element displayed, press the [DCLR] key to clear the element. Enter the new element and press the [ENT] key. This enters the change into memory.

Operation	Mode Switch Setting	Processor Status
Changing an Element	PROG	Stop

Example Sequence



Key-In Procedure	Type of Display		
	Command	Numerical Display	Mode Display
[CLR] [AND] [2] [SRC]	• AND	002	
[DCLR]			• DATA
[AND] [NOT] [4]	• AND • NOT	4	
[ENT]	• OUT • T/C	0009.5	• PROG
[DCLR]			
[OUT] [T/C] [0] [•/STEP] [1] [1] [•/STEP] [4] [ENT]	• OUT • T/C	0.11.4	

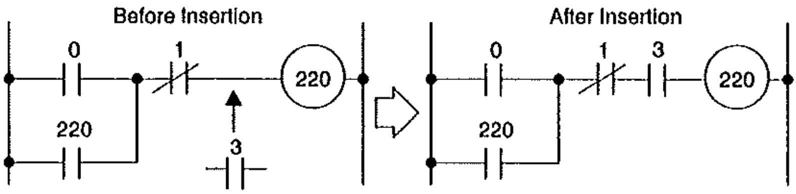
Figure 5.9 - Changing an Element

Use the procedure in figure 5.10 to add an element to an existing rung. Display the element after the address where you want to insert the new element. See figure 5.11. Press the [DCLR] key to clear the element. Type in the element to be inserted. Press the [INS] key. The addresses automatically increment by one to make space available for the new element.

Once you have inserted the element, press the [CLR] [SRC] keys to check the program.

Operation	Mode Switch Setting	Processor Status
Inserting an Element	PROG	Stop

Example Sequence



Key-in Procedure	Type of Display		
	Command	Numerical Display	Mode Display
CLR OUT 2 2 0 SRC	• OUT	*220*	• DATA • PROG
DCLR			
AND 3	• AND	*3*	
INS	• OUT	*220*	

Figure 5.10 - Inserting an Element

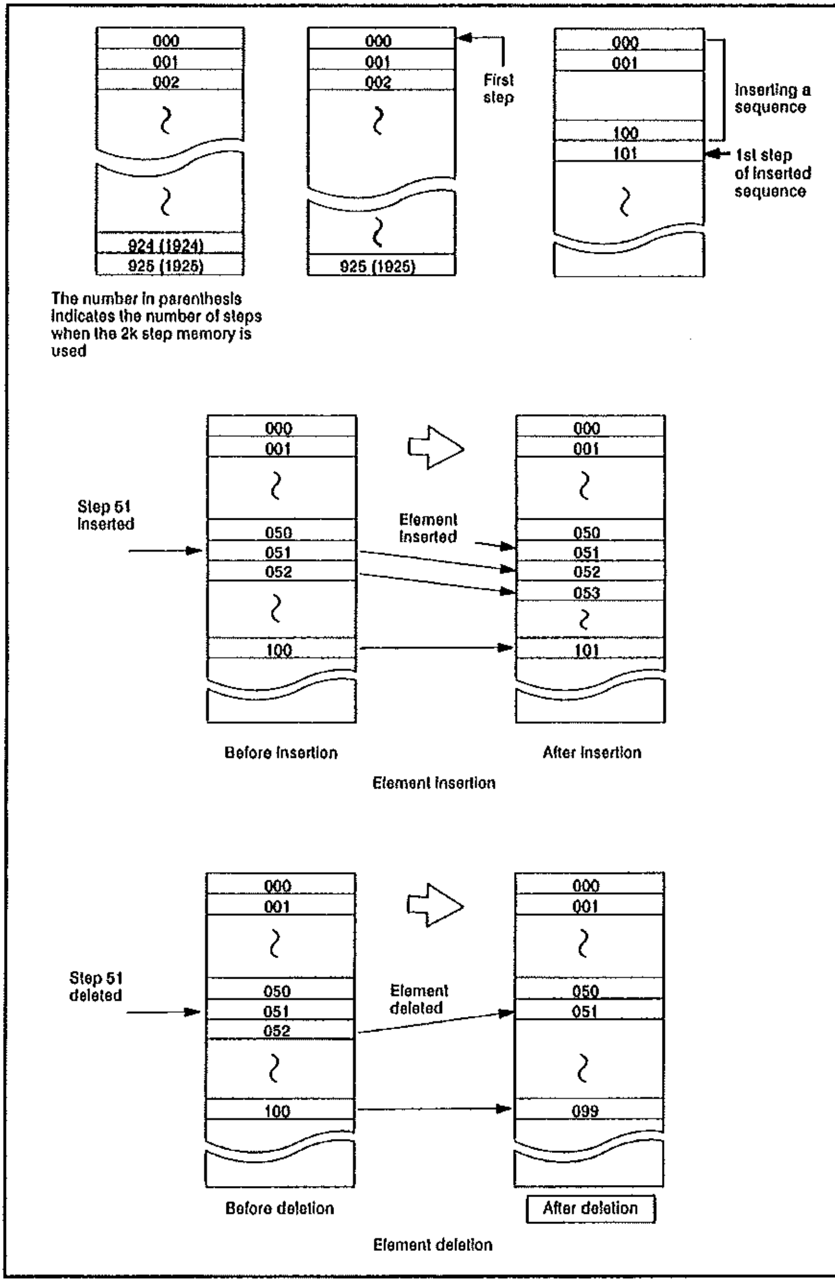


Figure 5.11 - Effect on Memory of Inserting and Deleting Elements

The processor can have a memory capacity of 926 (1K) words or 1926 (2K) words. The processor assigns each element a decimal number starting from 000.

Arithmetic instructions except for the BCD, BNR, SFR, and JMP are 2-word instructions. All other instructions use 1-word of memory.

Note that the 925th step (or 1925th step) is reserved for the END instruction. You cannot program a 2-word instruction in the 924th (or 1924th) step.

You will not be able to insert an element when memory is full. The display will show the overflow symbol in figure 5.12.

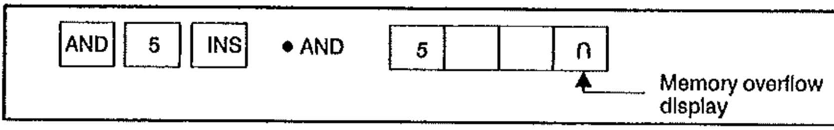
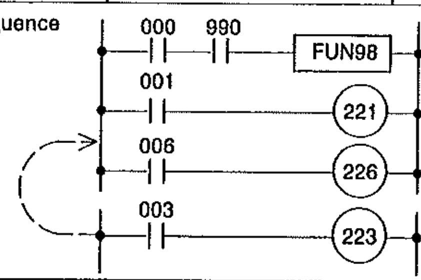


Figure 5.12 - Memory Overflow Display

Use the procedure in figure 5.13 to insert a rung into a program. Display the input element (input 006 in figure 5.13) of the rung that immediately follows the location you want to insert the rung at. Press the [DCLR] key to clear the input element from the handheld programmer's display. Type in the input element of the new rung (input 003 in the figure). Press the [INS] key. Press the [DCLR] key to clear the programmer's display. Type in the output of the new rung (output 223 in the figure). Press the [INS] key. The addresses in memory automatically increment to make space available for the new rung.

Operation	Mode Switch Setting	Processor Status
Inserting a Rung	PROG	Stop

Example Sequence



Key-in Procedure	Type of Display: Command	Numerical Display	Mode Display
[DCLR]			
[ORG] [3] [INS]	• ORG	3	• DATA
[DCLR]			• PROG
[OUT] [2] [2] [3] [INS]	• OUT	223	

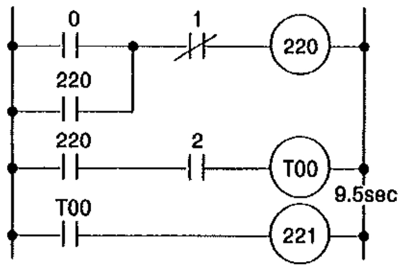
Figure 5.13 - Inserting a Rung

5.4 Displaying a Program

Use the procedure in figure 5.14 to display a program stored in memory.

Operation	Mode Switch Setting	Processor Status
Displaying a Program	PROG TEST RUN	Running or Stopped

Sample Sequence



Key-in Procedure	Type of Display: Command	Type of Display: Numerical Display	Type of Display: Mode Display
CLR STEP +	• ORG	7	
STEP +	• AND	990	
STEP +	• FUN	98	
STEP +	• ORG	000	
STEP +	• OR	220	• DATA
STEP +	• AND • NOT	001	
STEP +	• OUT	220	(• PROG
STEP +	• ORG	220	• TEST
STEP +	• AND	002	• RUN)
STEP +	• OUT • T/C	0.9.5	
STEP +	• ORG • T/C	0	
STEP +	• OUT	221	

Figure 5.14 - Displaying a Program

You can display a program beginning at a specific address or I/O number. Refer to figure 5.15.

Displaying a Program	Key-in Procedure
From step number 000.	[CLR] [STEP +] → [STEP +] or [STEP −]
From the first step of an unprogrammed area	[CLR] [·/ STEP] → [STEP +] or [STEP −]
From a specified step number	[CLR] [Step number] [·/ STEP] → [STEP +] or [STEP −]
From a specified contact number	[CLR] [Input/output number] [SRC] → [STEP +] or [STEP −]
From a specified coil	[CLR] [OUT] [Output number] [SRC] → [STEP +] or [STEP −]
From a specified command	[CLR] [Command] [SRC] → [STEP +] or [STEP −]

Figure 5.15 - Displaying a Program from a Specific Memory Location

5.5 Searching a Program

You can search a program for a specific I/O address, an internal coil address, a timer, a counter, or an instruction. Just enter the instruction number or address and press the [SRC] key. Refer to figure 5.16. The number is displayed on the programmer. Each time you press the [SRC] key, the processor searches the program for the next occurrence of the instruction. When the processor locates the instruction or address, the contents are displayed on the programmer's display.

If the processor does not find the instruction or address, the display shows the instruction underlined with the next available unused memory location.

Operation		Mode Switch Setting			Processor Status
Searching a program		PROG TEST RUN			Run Stop
Type of Search	Key-in Procedure	Type of Display: Command	Type of Display: Numerical Display	Type of Display: Mode display	Remarks
External and internal inputs/outputs (contact)	[CLR] [Input/Output Number] [SRC]	• AND	001	• DATA • PROG	External input 1
External/internal outputs (coil)	[CLR] [OUT] [Output Number] [SRC]	• OUT	200		Internal output 200
Timer/counter (contact)	[CLR] [T/C] [number] [SRC]	• AND • T/C	4		Timer 4
Timer/counter (coil)	[CLR] [OUT] [T/C] [number] [SRC]	• OUT • T/C	10.300		Timer 10 Preset value 300 sec
External input/output timer/counter	[CLR] [number] [SRC]	• ORG	002		External input 2
	[SRC]	• AND	002		External input 2
	[SRC]	• OR	002		External input 2
Search from step number	[CLR] [Step number] [•/STEP]	• AND	003		Command of specified step number displayed
Search from command	[CLR] [FUN] [0] [2] [SRC]	• FUN	02		
	[CLR] [FUN] [4] [7] [SRC]	• FUN	47400		

Figure 5.16 - Searching

Use the [STEP+] and [STEP-] keys to advance through and back up from the memory location currently displayed. Press the [SRC] key to continue searching.

Note: You cannot search for an address above step 926 unless your system has a 2K memory.

5.6 Executing a Program in Test Mode

Test mode allows you to run your program under controlled conditions. The test mode enables you to force outputs and change timer/counter presets. Refer to figure 5.17. The test mode can only be accessed through the handheld programmers. It cannot be accessed through the programming executive software.

Operation	Mode Switch Settings	Processor Status
Test Mode Start/Stop Procedure	TEST	Running

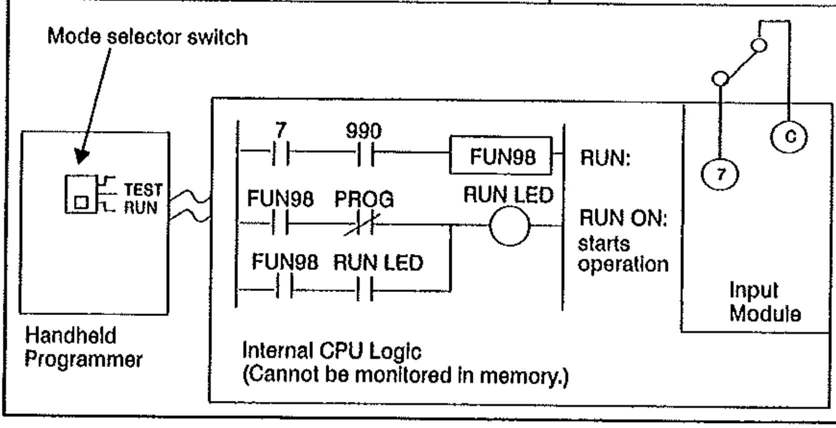


Figure 5.17 - Test Mode Operation

WARNING

EXECUTING A PROGRAM IN TEST MODE COULD RESULT IN UNEXPECTED MACHINE MOTION. STATION A PERSON AT THE MACHINE NEXT TO THE E-STOP BUTTON TO TERMINATE UNEXPECTED RESULTS. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

Program execution is controlled by FUN98. FUN98 is enabled when its inputs are active.

If the programmer is attached to the processor, set the mode selector switch to RUN or TEST. You cannot begin executing a program in PROG.

If the programmer is not attached to the processor, you cannot change the processor's mode of operation. If the processor is running and you remove the programmer, and then set the mode selector switch to PROG, the processor will continue to run.

The RUN LED is lit when the processor is running. To stop the processor, deactivate FUN98's inputs. You can attach or remove the programmer from the processor with power applied.

WARNING

DO NOT PROGRAM BOTH INPUTS TO FUN98 AS ADDRESS 990 AND THEN APPLY POWER AS THE PROCESSOR WILL GO INTO THE RUN MODE WHEN YOU REMOVE THE PROGRAMMER. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

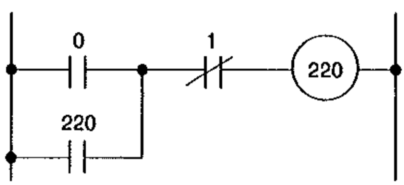
If you program both inputs to FUN98 as address 990 and then apply power, the processor will go into the run mode when the handheld programmer is removed. When you re-install the programmer, the processor will not respond to the mode selector switch setting. For the processor and programmer modes to coincide, you must cancel the run mode by de-activating FUN98 inputs or by cycling power.

5.6.1 Checking Continuity

Use the procedure in figure 5.18 to check contact continuity. If a contact is conducting, a decimal point is displayed.

Operation	Mode Switch Setting	Processor Status
Check Continuity	TEST	Running

Example Sequence



Contacts 1 and 220 are Conducting

Contact 0 is Not Conducting

Key-in Procedure	Type of Display: Command	Type of Display: Numerical Display	Type of Display: Mode display	Remarks
CLR 2 2 0 SRC	• OUT	220 .	• DATA • RUN	Indicates that output 220 is ON.
STEP –	• AND • NOT	001 .		Indicates that the contact is conducting.
STEP –	• OR	220 .		
STEP –	• ORG	000		Indicates that the contact is not conducting.

Figure 5.18 - Checking Continuity

5.6.2 Forcing Bit Data

You can test external inputs and outputs, internal outputs, timers, and counters through forcing in the test mode. Refer to figure 5.17. You need to monitor the device and press either the [SET] or [CLR] keys. Pressing the [SET] key energizes the internal output, timer, or counter. Pressing the [CLR] key clears the energized condition.

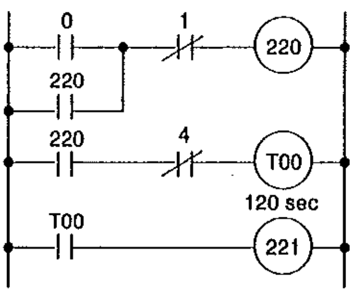
You can force a timer only when its enabling input is on. You can force a counter only when its reset input is off. A force command is executed after the processor runs one program scan. See figure 5.19.

In figure 5.19, if you try to force the output with contacts 220 and 4 de-energized, the timer will be turned off. The output will not be energized.

> **WARNING**
>
> **FORCING OUTPUTS CAN RESULT IN UNEXPECTED MACHINE MOTION. ALL PERSONNEL MUST BE CLEAR OF THE MACHINE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.**

Operation	Mode Switch Setting	Processor Status
Force External Inputs and Outputs, Internal Coils, Timers, or Counters Either On (Set) or Off (Reset)	TEST	Running

Example Sequence



Key-in Procedure	Type of Display: Command	Type of Display: Numerical Display	Type of Display: Mode display	Remarks
[CLR] [OUT] [T/C] [0] [MON]	• OUT • T/C	0.0108	• DATA • RUN	Monitor the current value of timer T00
[SET]	• OUT • T/C	0.120.0.		Forced set Timer T00 ON Output 221 ON
[CLR]				Clear forced output

Figure 5.19 - Forcing Outputs

5.6.3 Forcing Word Data

Word values can be forced in both decimal (5 digits) and hexadecimal (4 digits) notation. Refer to figures 5.20 and 5.21. Forcing hexadecimal word values into addresses 940 to 948 enable you to set the real time clock. Refer to section 5.11 for additional information.

You can test external inputs and outputs, internal outputs, timer values, and counter values through forcing in the test mode. You need to monitor the device and press either the [SET] or [CLR] keys. Pressing the [SET] key energizes the internal output, timer, or counter. Pressing the [CLR] key clears the energized condition.

You can force a timer's current or preset value only. You can force a counter's current or preset value only. A force command is executed after the processor runs one program scan.

> **WARNING**
> **FORCING OUTPUTS CAN RESULT IN UNEXPECTED MACHINE MOTION. ALL PERSONNEL MUST BE CLEAR OF THE MACHINE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.**

Operation	Mode Switch Setting	Processor Status
Forcing a Decimal Word Value	PROG TEST RUN	Running Stopped

Key-in Procedure	Type of Display: Command	Type of Display: Numerical Display	Type of Display: Mode Display	Remarks
[CLR] [OUT] [7] [0] [0] [MON]	• OUT	700	• DATA	Bit monitor address 700
[MON]	• OUT	00000	• PROG • TEST • RUN	Monitor address 700 as a decimal word
[1] [2] [3] [4] [5]	• FUN • OUT	12345		Enter 5-digit decimal value
[SET]	• OUT	12345		Force output whose address is in 700

Figure 5.20 - Forcing a Decimal Word Value

Operation	Mode Switch Setting			Processor Status
Forcing a Hexadecimal Word Value	PROG TEST RUN			Running Stopped
Key-in Procedure	Type of Display			Remarks
	Command	Numerical Display	Mode Display	
CLR OUT 8 0 0 MON	• OUT	*800*	• DATA	Bit monitor address 800
MON	• OUT	*00000*	• PROG • TEST • RUN	Monitor address 800 as a decimal word
MON	• OUT	*0000H*		Monitor address 800 as a hexadecimal word
1 2 3	• FUN • OUT	*0123*		Enter 4-digit hexadecimal value
SET	• OUT	*0123H*		Force output whose address is in 800

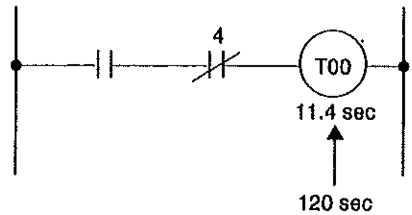
Figure 5.21 - Forcing a Hexadecimal Word Value

5.6.4 Changing Timer/Counter Preset Values

You can search for a specific timer or counter and change its preset value when the mode selector switch is in the TEST position. You replace the preset value by entering the new value and pressing the [ENT] key. Refer to figure 5.22.

Operation	Mode Switch Settings	Processor Status
Change a Timer/ Counter Preset Value	PROG	Stopped
	TEST	Running

Example Sequence



Key-In Procedure	Type of Display			Remarks
	Command	Numerical Display	Mode display	
[CLR] [OUT] [T/C] [0] [SRC]	• OUT • T/C	0.011.4	• DATA • TEST	Search for timer T00
[1] [2] [0] [ENT]	• OUT • T/C	0.120		Enter new preset value

Figure 5.22 - Changing Timer/Counter Values

When you change a timer/counter preset value, during timer/counter operation, the new value becomes active immediately.

NOTE: You cannot change a timer/counter preset value when the programmer is in the Run mode. Set the mode selector switch to TEST before you change a preset value. Changing modes requires the processor to first go to stop and may require cycling power to the processor.

5.7 Monitoring I/O

Use the procedures in figure 5.23 to monitor specific I/O points and word data.

Operation	Mode Switch Settings	Processor Status
Monitoring Input Output Addresses	TEST RUN	Running

Device or data to be Monitored	Key-in Procedure	Type of Display: Command	Type of Display: Numerical Display	Type of Display: Mode Display	Remarks
External or internal input/ output (contact)	[CLR] [Input/Output number] [MON]		001 .	• DATA	Internal input 1. ON
	[STEP +]		002		External input 2. OFF
External or internal output (coil)	[CLR] [OUT] [Input output no.] [MON]	• OUT	201 .		Internal output 201. ON
	[STEP +]	• OUT	202	(• PROG • TEST • RUN)	External output 202. OFF
Timer/ counter (contact)	[CLR] [T/C] [NO] [MON]	• T/C	10 .		Timer 10. ON
	[STEP −]	• T/C	9		Timer 9. OFF
Timer/ counter current value	[CLR] [OUT] [T/C] [NO] [MON]	• OUT • T/C	60.005		Counter 60, current value
	[STEP +]	• OUT • T/C	61.010		Counter 61, current value
Word Monitoring (external/ internal)	[CLR] [4] [0] [0] [MON]		400		Monitor internal output 400.
	[MON]		00255		Value of address 400 in decimal
	[MON]		00FFH		Value of address 400 in hexadecimal
	[MON]		400		Monitor internal output 400

Figure 5.23 - Monitoring I/O and Word Data

The programmer displays the external I/O address and indicates the status as a decimal point. A decimal point indicates that a contact is on. No decimal point indicates that a contact is off.

When you monitor a timer or counter, the two left-most digits represent the address of the device. The next three digits are the device's current value.

The I/O addresses and timer/counter numbers increment or decrement by one each time you press the [MON] key and either the [STEP+] key or the [STEP-] key.

External and internal word data can be monitored in either decimal or hexadecimal notation. As shown in figure 5.23, you monitor a word value by pressing the [CLR] key, entering the desired address, and then pressing the [MON] key. Pressing the [MON] key again, displays the contents of the address in decimal notation. Pressing [MON] a third time displays the contents in hexadecimal.

5.8 Saving and Loading Programs on Tape

Use the procedures in figure 5.24 to save or load a program on cassette tape.

Procedures are given for saving programs from the processor (EEPROM) to the cassette recorder, loading programs from the cassette recorder to the processor (EEPROM), and verifying the program transfer from either operation.

Operation		Mode Switch Setting					Processor Status
Saving and Loading Programs on Tape		PROG					Stop
Function	Key-in Procedure		Type of Display				Remarks
	Tape recorder	Programmer	Command	Numerical Display		Mode Display	
1 Enable cassette Operation		CLR SET SET ENT		R – – –			
		FUN 1		C – – –	Cassette function		
2 Record (SAVE)	Recording: MIC (Programmer) — MIC (Tape Recorder)	OUT ENT	● OUT	C – – P	Recording	● PROG	Processor (EEPROM) ↓ Cassette tape recorder
				C – – –	Complete		
Playback (LOAD)	Playback: EAR (Programmer) — Earphone (Tape Recorder)	STR ENT	● STR	C – – H	Waiting for start bit (30sec)	● DATA	Processor (EEPROM) ↑ Cassette tape recorder
				C – – P	Playing back		
				C – – –	Complete		
Verify	Playback: EAR (Programmer) — Earphone (Tape Recorder)	AND ENT	● AND	C – – H	Waiting for start bit (30sec)		Processor (EEPROM) ↕ Cassette tape recorder
				C – – P	Verifying		
				C – – –	Complete		

Figure 5.24 - Saving and Loading Programs

Function		Key-in Procedure		Type of Display				Remarks
		Tape recorder	Programmer	Command	Numerical Display		Mode Display	
2	Error display			● Command	C − − E	Key stroke error	● PROG	Press the CLR key to clear the error and retry.
					C 6 2 E	Playback error		
					C 7 − E	Verification error		
					C 8 − E	Format error		
3	Disable cassette operation		CLR RES RES ENT				● DATA	

*Be sure to verify the data after saving or loading a program on tape.

Cassette Tape Recorder	Description
Type of Cassette Recorder	Use a monaural cassette tape recorder.
Tone Setting	Set the tone to maximum.
Volume Setting	Set the volume to maximum.
Tape	Select a tape that is not scratched or wrinkled.

Figure 5.24 - Saving and Loading Programs (Continued)

Be sure to rewind the cassette before recording, loading, or verifying a program.

You should re-start the loading or saving procedure if you:

- lose power
- remove the tape
- press the [CLR] key.

When you are loading or verifying a program, the letter H is displayed for approximately 30 seconds. If the letter remains displayed for more than 30 seconds the tape is blank.

5.9 Transferring Programs in Memory

Use the procedures in figure 5.25 to transfer programs between the processor and external EEPROM/EPROM memories (M/N 45C975, 45C977, 45C978). The programmable memories attach to the PROM burning port on the M/N 45C951 handheld programmer.

No	Operation		Key Stroke Procedure	Type of Display	Remarks
1	Enable PROM Programming		[CLR] [SET] [SET] [ENT] [FUN] [2]	PROG R – – – 2 ROM MODE	
2	COPY	Mode 0	[OUT] [0] [0] [ENT] 1k EEPROM = 0 2K EEPROM : 1, EPROM : 2	PROG R – – P_OUT00 2 ROM MODE	Processor EEPROM → External Memory
		Mode 2	[OUT] [2] [0] [ENT] Same as mode above	PROG R – – P_OUT20 2 ROM MODE	45C951 RAM → External Memory
	LOAD	Mode 0	[STR] [ENT]	PROG R – – P_STR00 2 ROM MODE	Processor EEPROM ← External Memory
		Mode 2	[STR] [2] [ENT]	PROG R – – P_STR20 2 ROM MODE	45C951 RAM ← External Memory
	VERIFY	Mode 0	[AND] [ENT]	PROG R – – P_AND00 2 ROM MODE	Processor EEPROM → ← External Memory
		Mode 2	[AND] [2] [ENT]	PROG R – – P_AND20 2 ROM MODE	45C951 RAM → ← External Memory
	Blank check		[NOT] [ENT]	PROG R – – P_NOT 2 ROM MODE	Blank EEPROM Check
	Error display		Key Stroke error	R – – – E	
			Copy error	R62E OUT	Replace memory pack
			Verification error	R7 – E AND	
			Blank check error	R61E NOT	EPROM is not erased
3	Disable PROM Programming		[CLR] [RES] [RES] [ENT]		
					"P" disappears after completing the operation

Figure 5.25 - Processor/Programmer Memory Pack Program Transfer

5.10 Initializing the Second Half of 2K/4K EEPROM Memory

When transferring a programmed EEPROM (M/N 45C977) from an XL processor (M/N 45C900) to an XL II processor (M/N 45C901), you must follow the procedure in figure 5.26 to initialize the second half of memory (from 2K to 4K bytes).

Operation	Mode Switch Setting	Processor Status
Initializing the Second Half of 2K/4K EEPROM Memory	PROG	Stopped

Key Stroke	Type of Display		
	Command	Numerical Display	Mode Display
CLR			• PROG • DATA
ENT		E	
INS	• OUT	990	
ENT	• AND	990	

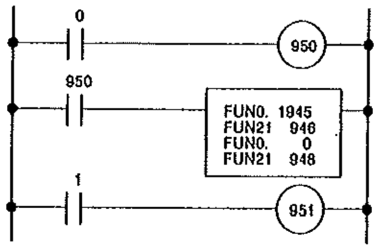
Figure 5.26 - Initializing the Second Half of EEPROM Memory

5.11 Setting the Real-Time Clock

To set the real-time clock, follow the procedure in figure 5.27. For additional programming information on the real-time clock, refer to section 4.2.3.1.

Operation	Mode Switch Settings	Processor Status
Setting the Real Time Clock	Test	Running

Sample Sequence



Key-In Procedure	Display	Description
CLR OUT 9 5 0 MON	950	Bit monitor address 950.
SET	950.	Set address 950 equal to one.
CLR OUT 9 4 0 MON MON MON	0089H	Monitor address 940 in hexadecimal.
0 0 9 0 SET	0090H	Set address 940 to the appropriate year (in hexadecimal).
STEP + STEP +	0927H	Monitor address 942 in hexadecimal.
0 7 2 8 SET	0728H	Set address 942 to the appropriate month and day (in hexadecimal).
STEP + STEP +	0004H	Monitor address 944 in hexadecimal.
0 0 0 7 SET	0007H	Set address 944 to the appropriate day of the week (in hexadecimal).
STEP + STEP +	1513H	Monitor address 946 in hexadecimal.
1 9 4 5 SET	1945H	Set address 946 to the appropriate hour and minute (in hexadecimal).

Figure 5.27 - Setting the Real-Time Clock

Key-In Procedure	Display	Description
STEP + STEP +	*0023H*	Monitor address 948 in hexadecimal.
0 0 3 6 SET	*0036H*	Set address 948 to the appropriate number of seconds (in hexadecimal).
CLR OUT 9 5 1 MON	*951*	Bit monitor address 951*.
SET	*951.*	Bit address 951 equal to 1.

* Be sure address 951 is equal to zero. If it is equal to one, press RES and go to the next step.

Figure 5.27 - Setting the Real-Time Clock (Continued)

6.0 INSTRUCTION SET/COMMAND SUMMARY

6.1 Introduction

This chapter explains the instructions and commands you use to program your Shark XL II programmable controller. Appendix B contains a summary of the instructions and commands. Appendix C lists the execution times of the instructions and commands. Appendix D contains a command and function (FUN) number cross-reference.

6.2 Instruction/Command Overview

The instructions and commands in this chapter are divided into three main categories according to their function:

- Conventional ladder instructions perform the relay replacement, timing, counting, sequencing, and logic control operations that are commonly provided by most programmable controllers.
- Program control commands provide a means of executing sections of the program when certain conditions are met. These commands include the program start/end functions, master control relays, one-shot inputs, and program jump operations.
- Advanced data manipulation commands provide the flexibility required in more complex control applications. They include four-function math commands, data compare commands, data transfer commands, data conversion commands, and shift register commands.

Three common terms are used throughout this chapter: arithmetic register (AR), extension register (ER), and carry bit (C).

The arithmetic register (AR) is a working register which serves as a temporary buffer for data while the CPU is processing instructions. The contents of the arithmetic register are reset to zero at the beginning of each program scan.

The extension register (ER) is 16-bit register which stores the upper word resulting from a multiplication operation or the remainder resulting from a division operation.

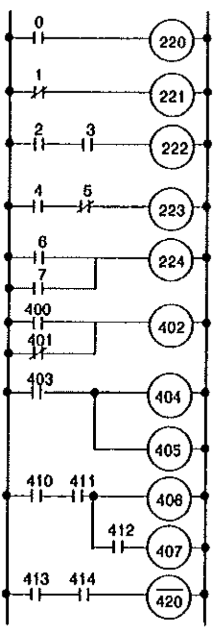
The carry bit (C) is a one-bit register which serves as a control bit for the compare instructions and an error bit for the mathematical instructions. The status of the carry bit is reset to zero at the beginning of each program scan.

6.3 Conventional Ladder Instructions

Use the following instructions for relay replacement and logic control operations. These instructions will also allow you to program timer, counter, and sequencer logic.

6.3.1 ORG, ORG NOT, AND, AND NOT, OR, OR NOT, OUT, and OUT NOT Instructions

- Use an ORG instruction to begin a rung with a normally open contact. Use an ORG NOT instruction to begin a rung with a normally closed contact. Each new rung must begin with either an ORG or an ORG NOT instruction. Refer to figure 6.1.
- Use an AND instruction for a normally open contact in series. Use an AND NOT instruction for a normally closed contact in series. Refer to figure 6.1.
- Use an OR instruction for a normally open contact in parallel. Use an OR NOT instruction for a normally closed contact in parallel. Refer to figure 6.1.
- Use the OUT instruction for external and internal outputs, timers, and counters. Use OUT NOT instruction to disable OUT instructions. You can use multiple OUT instructions in parallel. Refer to figure 6.1. You can enter a contact after an OUT instruction as shown in figure 6.2.



Coding		Remarks
ORG OUT	0 220	Use ORG for the start of a circuit.
ORG NOT OUT	1 221	
ORG AND OUT	2 3 222	Use AND for the contacts connected in series.
ORG AND NOT OUT	4 5 223	
ORG OR OUT	6 7 224	Use OR for the contacts connected in parallel.
ORG OR NOT OUT	400 401 402	
ORG OUT OUT	403 404 405	Multiple output
ORG AND OUT AND OUT	410 411 406 412 407	
ORG AND OUT NOT	413 414 420	When 413 and 414 are activated, output 420 is DE-ENERGIZED

Figure 6.1 - Sample ORG, ORG NOT, AND, AND NOT, OR, OR NOT, OUT, and OUT NOT Instructions

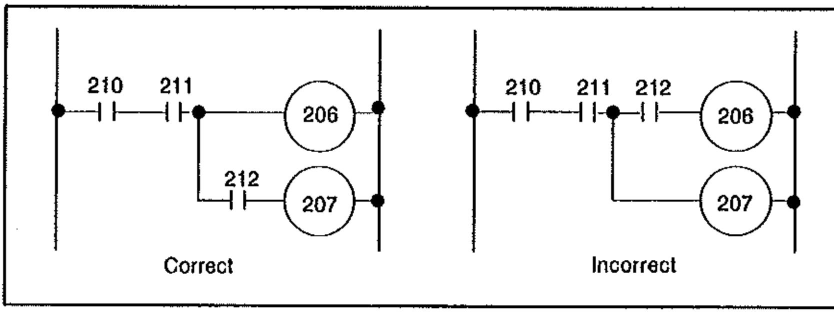
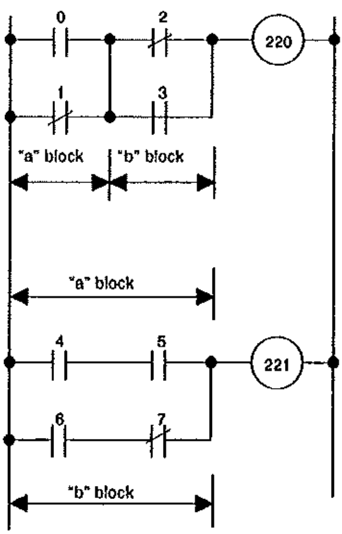


Figure 6.2 - Programming a Contact after an Output Instruction

6.3.2 STR, STR NOT, AND STR, and OR STR Instructions

- A circuit in which you connect two or more contacts in parallel is a parallel circuit block. For the first example, use the STR NOT instruction to begin a branch and the AND STR instruction to end a branch when you connect parallel circuit blocks in series. Refer to figure 6.3.
- A circuit in which you connect two or more contacts in series is a serial circuit block. Use the STR or STR NOT instructions to begin a branch and the OR STR instruction to end a branch when you connect serial circuit blocks in parallel. Refer to figure 6.3, second example.
- There is no limit to the number of circuit blocks you can connect together with AND STR or OR STR instructions.
- You are limited to seven STR or STR NOT instructions when you connect circuit blocks together.
- The number of STR/STR NOT instructions must equal the number of AND STR/OR STR instructions. If there is not an equal number, a syntax error will result. This requirement does not apply to counter, shift register, or up/down counter circuits.



Coding		Remarks	
ORG	0	a	Programming for "a" block.
OR NOT	1		
STR NOT	2	b	Programming for "b" block.
OR	3		
AND STR		a · b	"a" block and "b" block are formed by AND STR.
OUT	220		
ORG	4	a	Programming for "a" block.
AND	5		
STR	6	b	Programming for "b" block.
AND NOT	7		
OR STR		a+b	"a" block and "b" block are formed by OR STR.
OUT	221		

Figure 6.3 - Example of STR, STR NOT, AND STR, and OR STR Instructions

6.3.3 T (Timer) Instruction

- You enter a timer by pressing the out key [OUT], pressing the timer/counter key [T/C], keying in the timer number (1 or 2 digits), pressing the decimal point key [.], and then entering the preset value (3 or 4 digits). See figure 6.4.

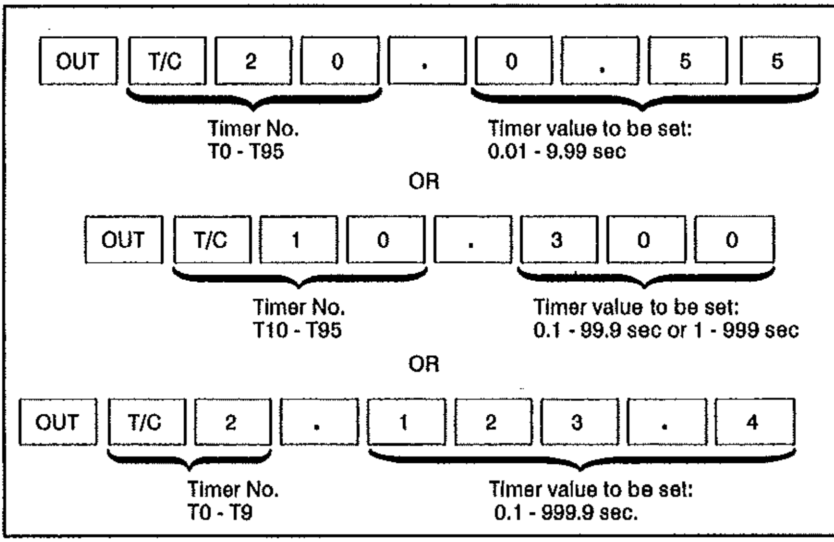
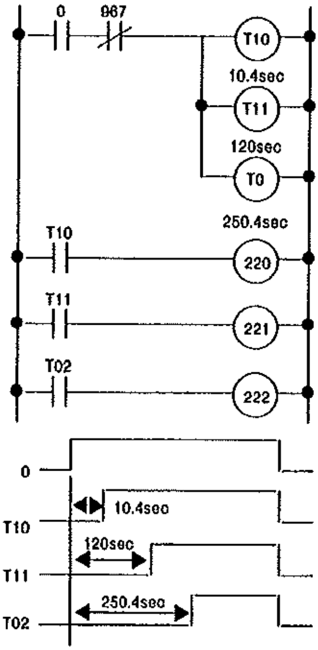


Figure 6.4 - Entering Timer Values

- Timers can be numbered from T0 to T95. Note that timers and counters share the same address range. You cannot use the same number for both a counter and a timer.
- Timers are of the on-delay type. When the timer input signal is energized, the timer begins counting from 0 up to the preset value. When the preset value is reached, the output is energized. When the input de-energizes, the timer stops counting and the elapsed value returns to 0. See figure 6.5.
- Timer accuracy:

  Timer error = +/– 0.005 x preset value + 0.1 seconds + 1 scan
- The timer's elapsed value is retentive. Reset the value by using internal output 967 when you first power-up the processor.

T (Timer) Instruction (Continued)



Coding			Remarks
ORG		0	
AND NOT		967	
OUT	T/C	10.10.4	
OUT	T/C	11.120	
OUT	T/C	2.250.4	
ORG	T/C	10	
OUT		220	
ORG	T/C	11	
OUT		221	
ORG	T/C	02	
OUT		222	

Figure 6.5 - Example of the T (Timer) Instruction

6.3.4 C (Counter) Instruction

- You enter a counter by pressing the out key [OUT], pressing the timer/counter key [T/C], keying in the counter number (1 or 2 digits), pressing the decimal point key [.], and then entering the preset value (3 or 4 digits). See figure 6.6.

  Counters C0 to C9 can count to 9999. Counters C10 to C95 can count to 999.

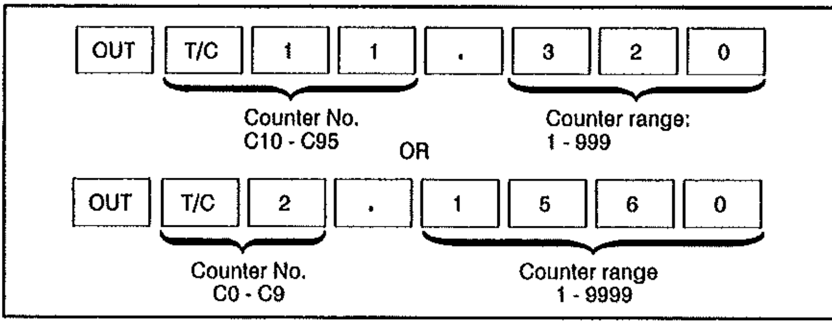
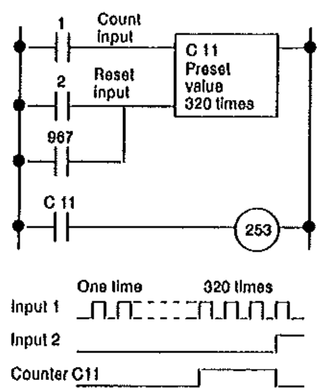


Figure 6.6 - Entering Counter Values

- Counters can be numbered from C0 to C95. Note that counters and timers share the same address range. You cannot use the same number for both a counter and a timer.
- Counters are of the up-counter type. The counter counts the rising edge of the input signal. When the preset value is reached, the output is energized. When the reset input signal is energized, the counter is reset to 0. See figure 6.7.
- Counter values are retentive. Reset the counter by using internal output 967.



Coding			Remarks
ORG		1	Count input
STR		2	} Reset input
OR		967	
OUT	T/C	11.320	Counter 11 with a preset of 320
ORG	T/C	11	
OUT		253	

Figure 6.7 - Example of the C (Counter) Instruction

C (Counter) Instruction (Continued)

- Figure 6.8 shows an interruptible timer using a counter instruction. An interruptible timer will retain its elapsed or current value when it is started and stopped by program execution.

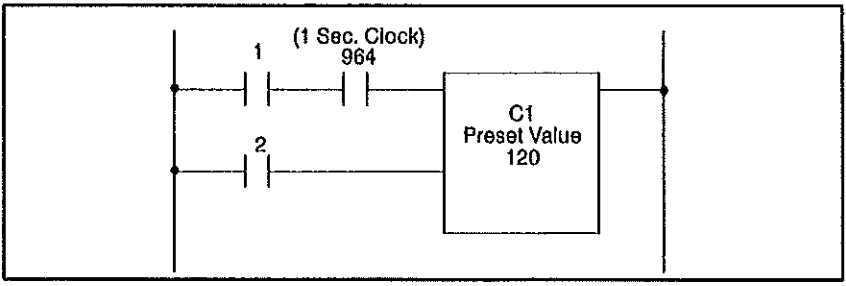


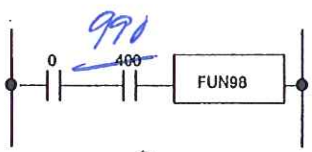
Figure 6.8 - Interruptible Timer Using a Counter Instruction

6.4 Program Control Commands

These commands allow you to execute sections of the program when certain conditions are met. These commands include the program start/end functions, master control relays, one-shot inputs, and program jump operations.

6.4.1 Start and End Commands

- FUN98 = Start Command = STA
- FUN99 = End Command = END
- Two contacts in series are required in the program's beginning rung to provide the input signal for FUN98. See figure 6.9. If you have only one contact, program the second one as internal output address 990. Do not program both contacts as address 990 because this will cause the processor to stay in the run mode continuously.
- FUN99 marks the end of the program. Program scanning begins at step 000 and continues until FUN99 is encountered. FUN99 is useful when testing a program. By inserting FUN99 within a program, only the sequences between FUN98 and the first FUN99 are executed. Refer to figure 6.10. Note that even though FUN99 is not displayed, it is placed in all unused program areas by the CPU.
- When FUN99 is used before the end of the program, the sequences following FUN99 will not be scanned.



Coding		Remarks
ORG	0	
AND	400	
FUN98		STA
FUN99		END

Figure 6.9 - Example of FUN98 and FUN99

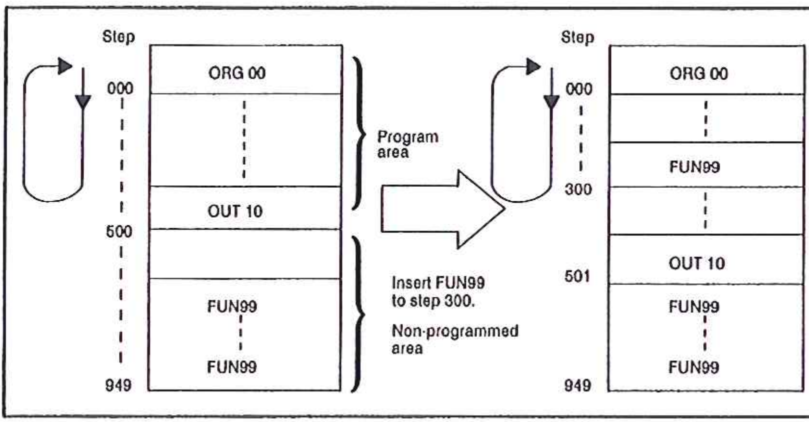
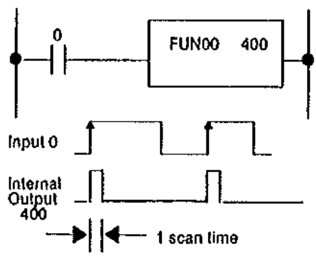


Figure 6.10 - Example of FUN99 Operation

6.4.2 Edge Detection Commands

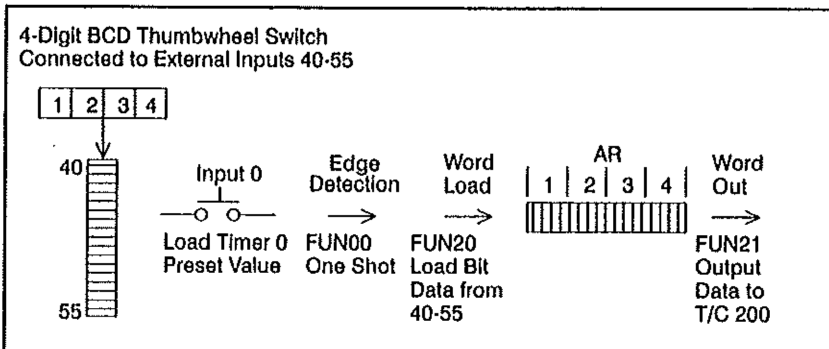
- FUN00 = Leading Edge Detection Command = DIF
- FUN01 = Trailing Edge Detection Command = DFN
- You can use FUN00 and an internal output to detect the leading edge of an input signal and output a pulse for one scan. Refer to figure 6.11.

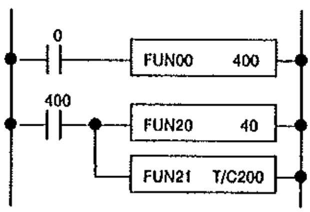
  You can also use FUN00 to change a timer's preset value through an external thumbwheel switch. See figure 6.12.
- FUN00 functions as a "one-shot" instruction and is useful for beginning the execution of Load Instructions (FUN0., FUN10, FUN20, FUN50, and FUN60), Compare Instructions (FUN7., FUN8., FUN9., FUN17, FUN18, FUN19, FUN57, FUN58, and FUN59), and similar instructions.
- You can use FUN01 and an internal output to detect the trailing edge of an input signal [a status change from high (on) to low (off)] and output a pulse for one scan. Refer to figure 6.13.
- Any number of FUN00 and FUN01 commands may be programmed subject to the availability of internal outputs.



Coding		Remarks
ORG	0	Use FUN00 to detect the leading edge of an input.
FUN00	400	

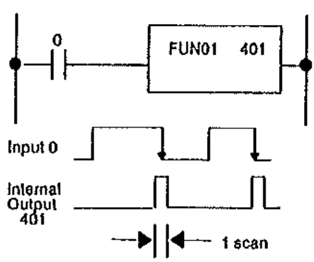
Figure 6.11 - Example of FUN00





Coding		Remarks
ORG	0	One shot to load timer 0 preset into AR
FUN00	400	
ORG	400	Load external inputs 40-55 into the AR
FUN20	40	
FUN21	T/C 200	Load new preset value for timer 0 from the AR

Figure 6.12 - Changing a Timer's Preset Value Using an External BCD Thumbwheel Switch

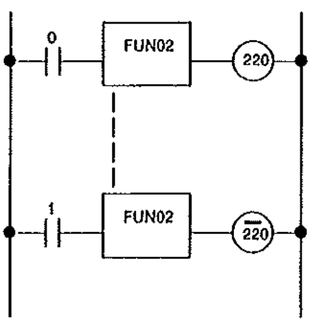


Coding		Remarks
ORG	0	Use FUN01 to detect the trailing edge of an input and to turn on an internal output for one scan.
FUN01	401	

Figure 6.13 - Example of FUN01

6.4.3 If, Set, Reset, and If with Reset Commands

- FUN02 = If Command = IF
- FUN88 = Set Command = SET
- FUN89 = Reset Command = RES
- FUN03 = If Command with a Reset Input = IFR
- Use FUN02 when the on/off status is being referenced to an output. If FUN02 is used in conjunction with an OUT instruction, and the rung's input signal is true, the output will stay on until another input to FUN02 turns it off. See figure 6.14. A latch circuit can be programmed using FUN02 and a retentive internal output.
- FUN88 functions the same as a FUN02 command with an OUT instruction. FUN89 functions the same as an FUN02 command with an OUT NOT instruction. See figure 6.15. FUN88 and FUN89 use less memory than the equivalent FUN02 command and OUT/OUT NOT instructions. If an output is programmed through a FUN88 command and a FUN89 command, a syntax error (double coil error) will result.
- Use FUN03 when the on/off status is being referenced to an input. See figures 6.16, 6.17, and 6.18.



Coding		Remarks
ORG	0	
FUN02		
OUT	220	
ORG	1	
FUN02		
OUT NOT	220	

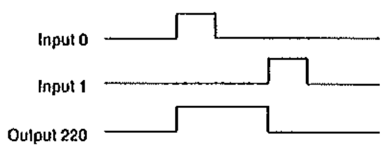
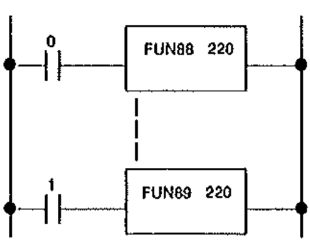
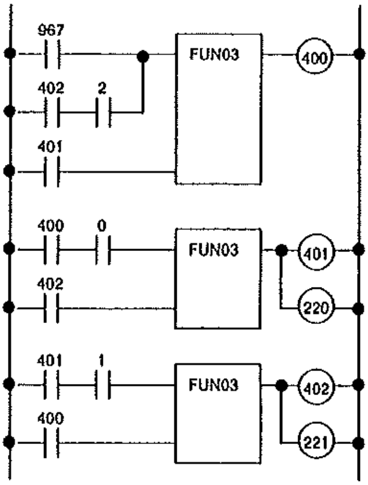


Figure 6.14 - Example of FUN02



Coding		Remarks
ORG	0	
FUN88	220	
ORG	1	
FUN89	220	

Figure 6.15 - Examples of FUN88 and FUN89



Coding		Remarks
ORG	967	
STR	402	
AND	2	
OR STR		
STR	401	
FUN03		
OUT	400	
ORG	400	
AND	0	
STR	402	
FUN03		
OUT	401	
OUT	220	
ORG	401	
AND	1	
STR	400	
FUN03		
OUT	402	
OUT	221	

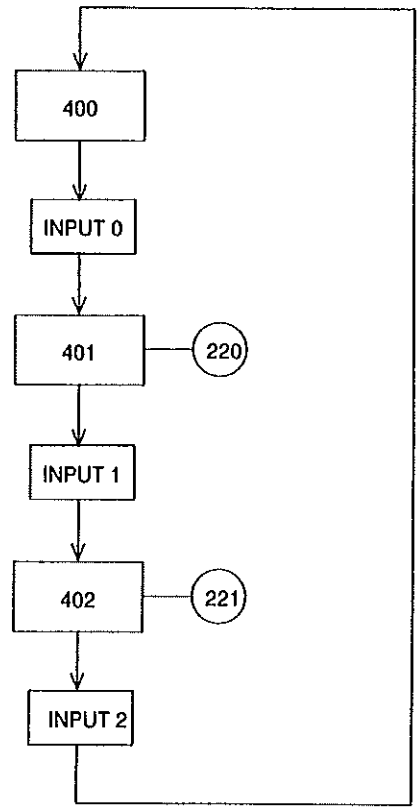
A. If input 0 is on, internal output 401 and external output 220 are energized.

B. External output 220 will remain on even if input 0 turns off.

C. If input 1 is on, external output 220 is turned off.

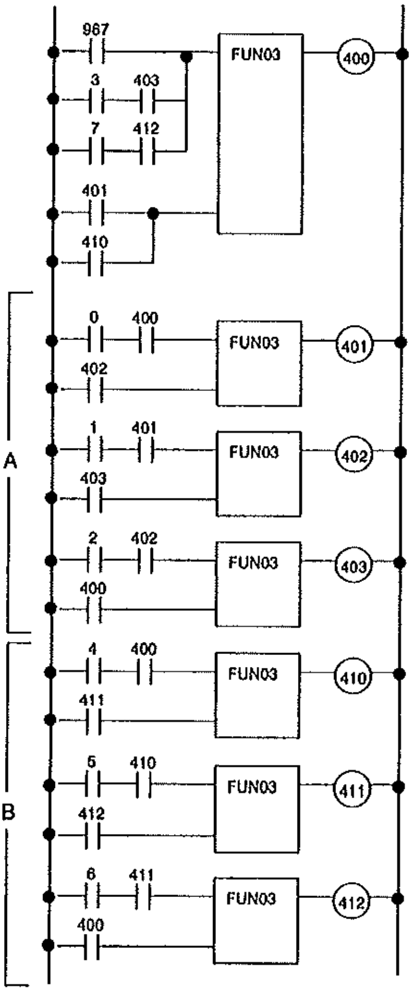
D. If input 1 is on when internal output 401 is on, internal output 402 will not turn on.

Figure 6.16 - Example of FUN03 Programmed for Series Operation



Block Diagram

Figure 6.16 - Example of FUN03 Programmed for Series Operation (Continued)



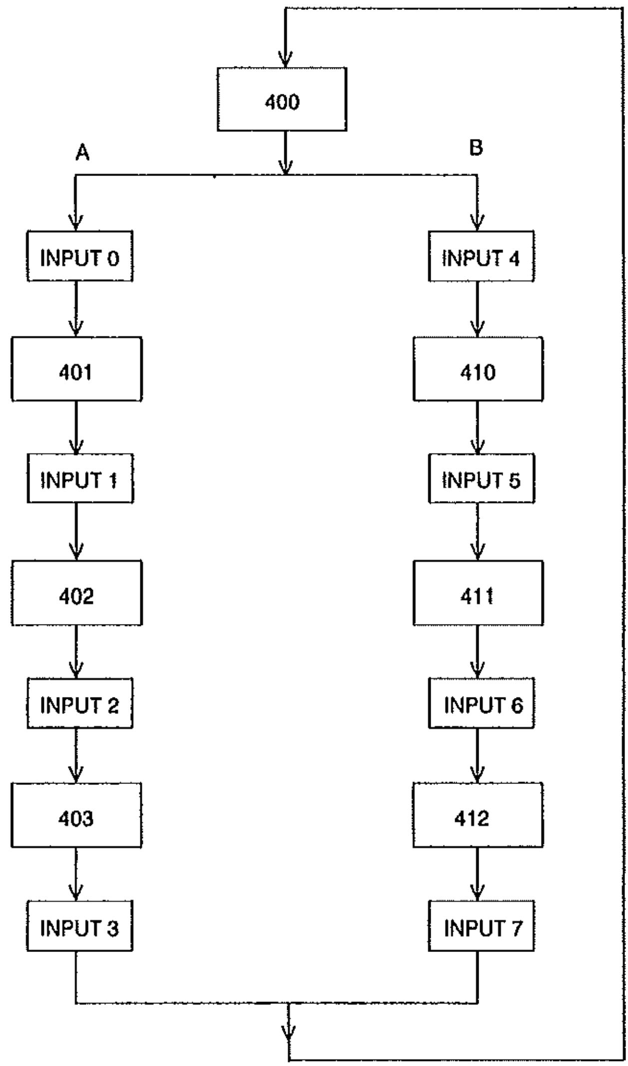
Coding		Remarks
ORG	967	
STR	3	
AND	403	
STR	7	
AND	412	
OR STR		
STR	401	
STR	410	
FUN03		
OUT	400	
ORG	0	
AND	400	
STR	402	
FUN03		
OUT	401	
ORG	1	
AND	401	
STR	403	
FUN03		
OUT	402	
ORG	2	
AND	402	
STR	400	
FUN03		
OUT	403	
ORG	4	
AND	400	
STR	411	
FUN03		
OUT	410	
ORG	5	
AND	410	
STR	412	
FUN03		
OUT	411	
ORG	6	
AND	411	
STR	400	
FUN03		
OUT	412	

A. If internal output 400 is on and then input 000 turns on, the "A" group of rungs is enabled for execution. If internal output 400 is on and then input 4 turns on, the "B" group of rungs is enabled for execution.

B. The "A" group of rungs, (rungs containing internal outputs 401, 402, and 403), is evaluated in sequence. When input 3 turns on, program control returns to the rung containing internal output 400.

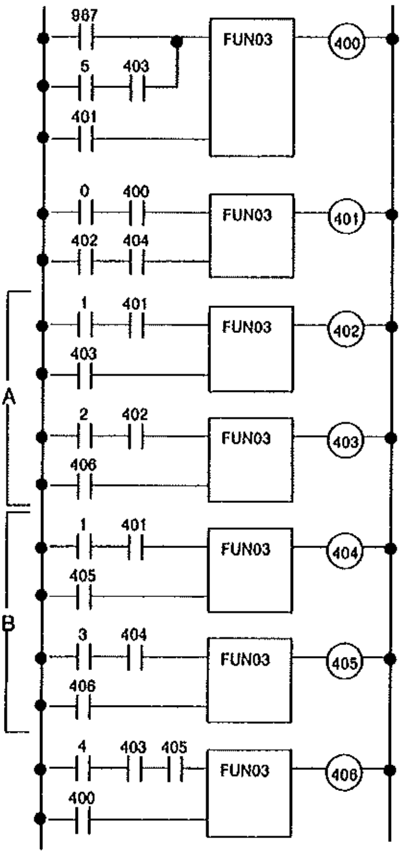
C. The "B" group of rungs, (rungs containing internal outputs 410, 411, and 412), is evaluated in sequence. When input 7 turns on, program control returns to the rung containing internal output 400.

Figure 6.17 - Example of FUN03 Programmed for Parallel Operation



Block Diagram

Figure 6.17 - Example of FUN03 Programmed for Parallel Operation (Continued)



Coding		Remarks
ORG	987	
STR	5	
AND	406	
OR STR		
STR	401	
FUN03		
OUT	400	
ORG	0	
AND	400	
STR	402	
AND	404	
FUN03		
OUT	401	
ORG	1	
AND	401	
STR	403	
FUN03		
OUT	402	
ORG	2	
AND	402	
STR	406	
FUN03		
OUT	403	
ORG	1	
AND	401	
STR	405	
FUN03		
OUT	404	
ORG	3	
AND	404	
STR	406	
FUN03		
OUT	405	
ORG	4	
AND	403	
AND	405	
STR	400	
FUN03		
OUT	406	

A. If internal output 400 is on and then input 000 is on, internal output 401 turns on.

B. When input 001 turns on, internal outputs 402 and 404 turn on simultaneously. The "A" and "B" groups of rungs begin executing together.

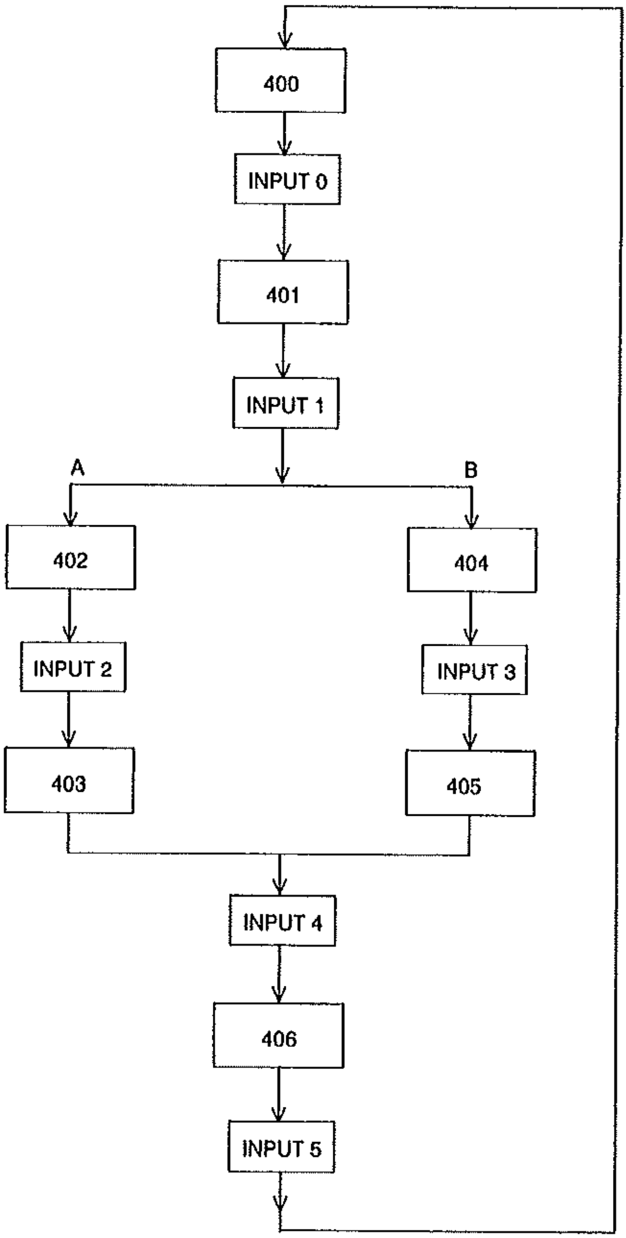
C. In the "A" group of rungs, internal output 403 turns on when input 002 is on.

D. In the "B" group of rungs, internal output 405 turns on when input 003 is on.

E. If internal outputs 403 and 405 are on and input 004 is on, internal output 406 turns on.

F. When input 005 is on, program control is taken from the "A" and "B" groups of rungs simultaneously and returned to the rung containing internal output 400.

Figure 6.18 - Example of FUN03 Programmed for Series and Parallel Operation (Continued)

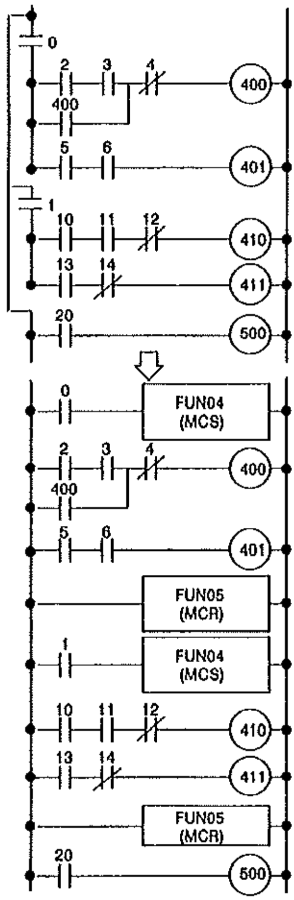


Block Diagram

Figure 6.18 - Example of FUN03 Programmed for Series and Parallel Operation (Continued)

6.4.4 Master Control Set and Master Control Reset Commands

- FUN04 = Master Control Set Command = MCS
- FUN05 = Master Control Reset Command = MCR
- These commands control the bus line. Use them together or a syntax error will result. Refer to figure 6.19.
- A control contact must preceed the FUN04 command.
- When the FUN04 command is enabled, the status of the outputs between FUN04 and FUN05 is determined by the program's logic.
- When the FUN04 command is disabled, the logic is scanned and the outputs between FUN04 and FUN05 are turned off.
- Enter an ORG or ORG NOT command after the FUN04 command or a syntax error will result.



Coding		Remarks
ORG	0	
FUN	04	MCS
ORG	2	
AND	3	
OR	400	
AND NOT	4	
OUT	400	
ORG	5	
AND	6	
OUT	401	
FUN	05	MCR
ORG	1	
FUN	04	MCS
ORG	10	
AND	11	
AND NOT	12	
OUT	410	
ORG	13	
AND NOT	14	
OUT	411	
FUN	05	MCR
ORG	20	
OUT	500	

Figure 6.19 - Example of FUN04 and FUN05 Commands

- You can program up to three nesting levels of the FUN04 and FUN05 commands. See figure 6.20.

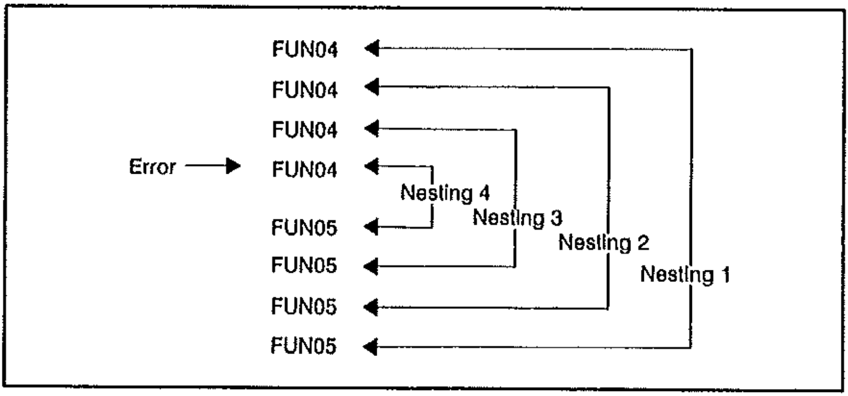


Figure 6.20 - Nesting Levels

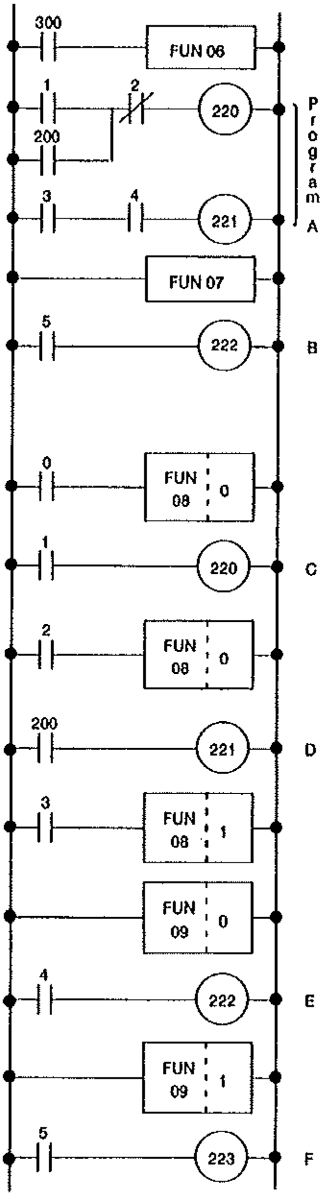
6.4.5 Jump, Jump End, Addressed Jump, and Addressed Jump End Commands

- FUN06 = Jump Command = JMP
- FUN07 = Jump End Command = JMP END
- FUN08 = Addressed Jump Command = AJMP
- FUN09 = Addressed Jump End Command = AJMP END
- These commands allow the processor to skip over specific program sequences. Refer to figure 6.21.
- FUN06 and FUN07 do not require addresses. You must use them together. If you do not, a syntax error will result. You cannot nest logic with these commands.
- FUN08 and FUN09 do require addresses. Addresses may range from 0 to 9. You can program multiple jumps from FUN08 to a single FUN09. Nesting is allowed.
- Jump commands are not allowed between FUN04 and FUN05.
- When a program sequence is skipped, output status does not change and timer/counter current values are retained. When the normal program resumes, the timers and counters begin operating again.
- The functional combination of FUN06 and FUN07 or FUN08 and FUN09 may be used to reduce processing scan time.

WARNING

INADVERTENT MACHINE OPERATION MAY RESULT WHEN A JUMP COMMAND IS ACTIVE. RECOGNIZE THAT THE PORTION OF THE PROGRAM BETWEEN FUN06 AND FUN07 OR BETWEEN FUN08 AND FUN09, IS NOT EVALUATED AND THE OUTPUTS IN THAT PORTION OF THE PROGRAM REMAIN IN THEIR PREVIOUS STATES. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

Jump, Jump End, Addressed Jump, and Addressed Jump End Commands (Continued)

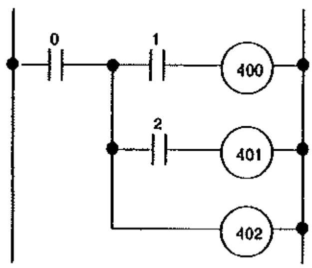


Coding		Remarks
ORG	300	
FUN06		JMP
ORG	1	
OR	200	
AND NOT	2	
OUT	220	Status before JMP retained.
ORG	3	
AND	4	
OUT	221	
FUN07		JMP END
ORG	5	
OUT	222	
ORG	0	
FUN08	0	JMP 0
ORG	1	
OUT	220	Status before JMP retained.
ORG	2	
FUN08	0	JMP 0
ORG	200	
OUT	221	Status before JMP retained.
ORG	3	
FUN08	1	JMP 1
FUN09	0	JMP 0 END
ORG	4	
OUT	222	Status before JMP retained.
FUN09	1	JMP 1 END
ORG	5	
OUT	223	

Figure 6.21 - Example of FUN06, FUN07, FUN08, and FUN09 Commands

6.4.6 Branch and Return Commands

- FUN28 = Branch Command
- FUN29 = Return Command
- The FUN28 and FUN29 commands allow the programming of multiple outputs. FUN28 enables multiple outputs to share contacts up to the branch point. FUN29 brings the program's logic back to the branch point if additional outputs are to follow. See figure 6.22.



Coding		Remarks
ORG	0	
FUN	28	BRANCH
AND	1	
OUT	400	
FUN	29	RETURN
AND	2	
OUT	401	
FUN	29	RETURN
OUT	402	

Figure 6.22 - Example of FUN28 and FUN29

- Note that you cannot use master control commands (FUN04 and FUN05) between the branch and return commands.
- Note that the SPX software does not recognize the FUN28 and FUN29 commands.
- FUN28 cannot be used more than once in a rung. If additional branching is required, use the master control commands (FUN04 and FUN05). See figure 6.23.

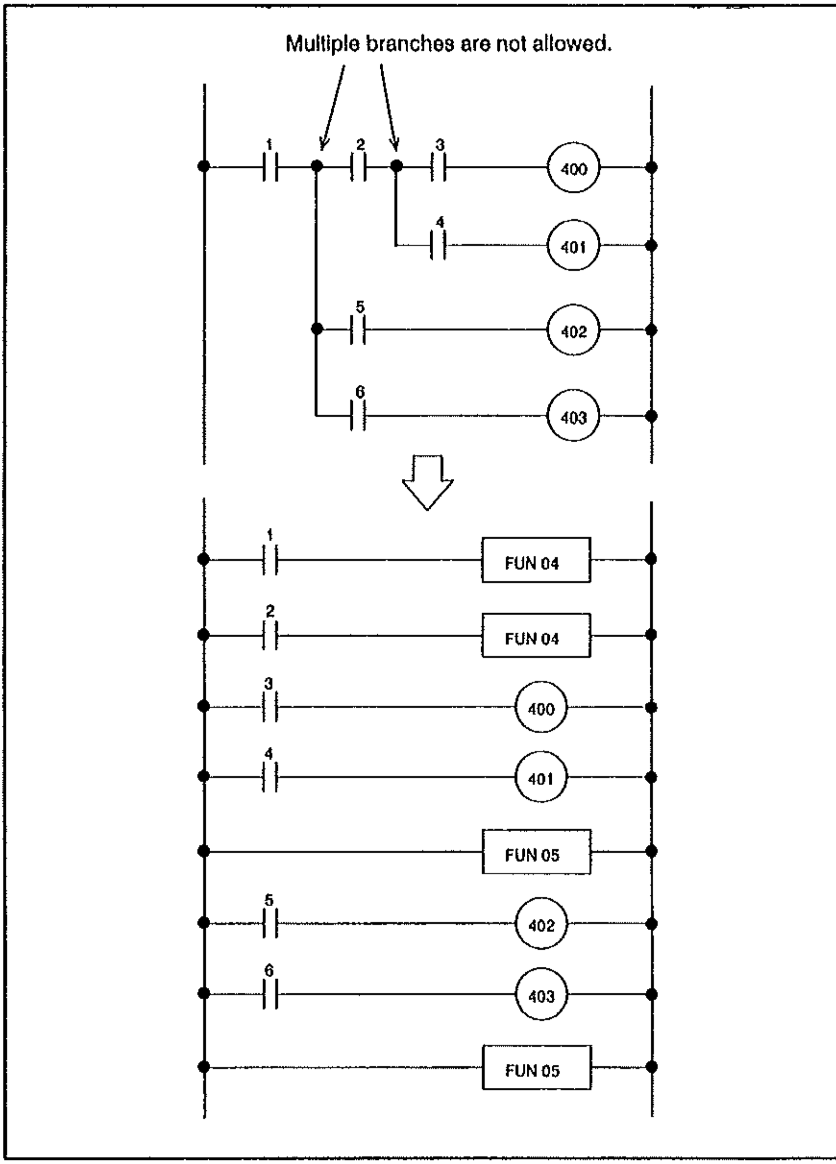
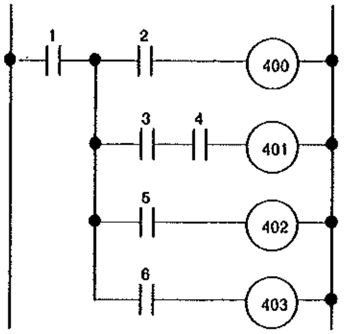


Figure 6.23 - Use FUN04 and FUN05 to Program Multiple Branches

Branch and Return Commands (Continued)

- FUN29 can be used any number of times with a single FUN28 command. See figure 6.24.

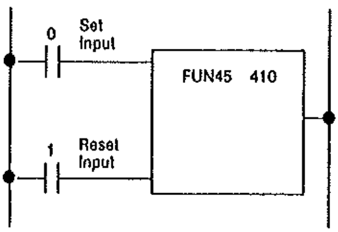


Coding		Remarks
ORG	1	
FUN 28		
AND	2	
OUT	400	
FUN 29		
AND	3	
AND	4	
OUT	401	
FUN 29		
AND	5	
OUT	402	
FUN 29		
AND	6	
OUT	403	

Figure 6.24 -Multiple Return Commands can be Programmed With a Single Branch Command

6.4.7 Latch Command

- FUN45 = Latch Command = LATCH
- FUN45 must be programmed with an internal output. Refer to figure 6.25.
- Internal outputs latch with the leading edge of the input when the reset input is off.
- This command will function as a retentive output if the chosen internal output is retentive (700-955).
- When the reset input's status changes from off to on, the latch resets.



Coding		Remarks
ORG	0	Use the Str Instruction when programming FUN 45.
STR	1	
FUN45	410	

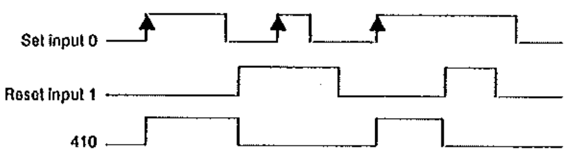
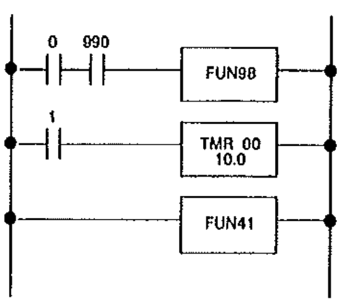


Figure 6.25 - Example of FUN45

6.4.8 No Operation Command

- FUN41 = No Operation Command = NOP
- This command prevents the rung from being executed.
- FUN41 must be programmed as an output. See figure 6.26.



Coding		Remarks
ORG	0	
AND	990	
FUN98		
ORG	1	
T/C00.10.0		
FUN41		No Operation

Figure 6.26 - Example of FUN41

6.4.9 I/O Update Commands

- FUN91 = Update Specified External Inputs Command = REF X
- FUN92 = Update Specified External Outputs Commands = REF Y
- Upon execution, FUN91 updates the state of the specified external input during the normal scan cycle. See figure 6.27. Input signals of shorter duration than the program scan time may be detected by the uniform placement of FUN91 throughout the program. FUN91 does not require a start condition. See figure 6.28.
- Upon execution, FUN92 updates the state of the specified external output during the normal scan cycle. See figure 6.27. FUN92 requires a start condition. See figure 6.28. This command is useful when an output update is required prior to the end of the program scan cycle.

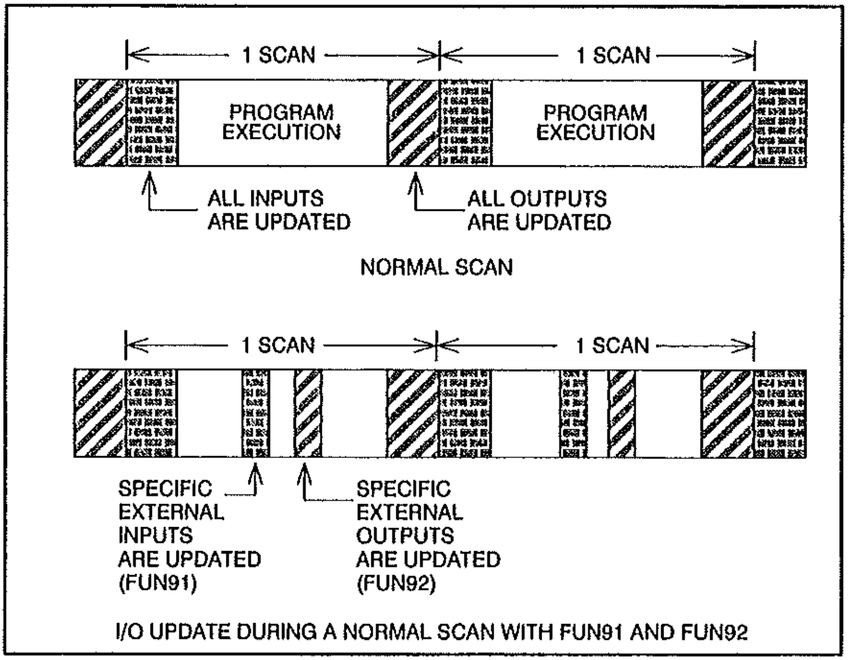
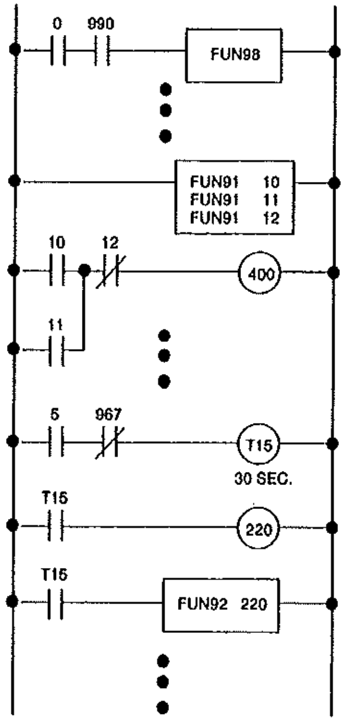


Figure 6.27 - Updating I/O with FUN91 and FUN92 Active



Coding		Remarks
ORG	0	
AND	990	
FUN98		
FUN91	10	Update the current state of external inputs 10, 11, 12.
FUN91	11	
FUN91	12	
ORG	10	
OR	11	
AND NOT	12	
OUT	400	
ORG	5	
AND NOT	967	
OUT	T/C15.030	
ORG	T/C15	
OUT	220	
ORG	T/C15	
FUN92	220	When T15 is on, update the status of external output 220.

Figure 6.28 - Example of FUN91 and FUN92

6.4.10 Interrupt Commands

- FUN93 = Interrupt Command = INT
- FUN94 = Return from Interrupt Command = RTI
- The interrupt commands allow the execution of an interrupt program at a fixed interval of 10 milliseconds.
- Only one interrupt program is permitted and must be placed at the end of the main program. FUN93 marks the beginning of the interrupt program. FUN94 marks the end of the interrupt program and returns processor execution to the main program.
- When FUN93 is used, processor execution begins with the main program. At 10 millisecond intervals, execution of the main program will be suspended and the interrupt program will be executed. When FUN94 is encountered, the main program will resume executing at the same point it was at prior to the interrupt. See figure 6.29.
- When FUN93 and FUN94 are active, FUN06 (jump) and FUN07 (jump end) may not be used.

WARNING

CARE MUST BE EXERCISED WHEN USING THE INTERRUPT COMMAND. MACHINE OPERATION MAY BE OUT OF SYNCHRONIZATION WITH THE REST OF THE PROGRAM WHEN AN INTERRUPT PROGRAM IS EXECUTED. UNCONTROLLED MACHINE OPERATION MAY RESULT IF THIS IS NOT DONE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

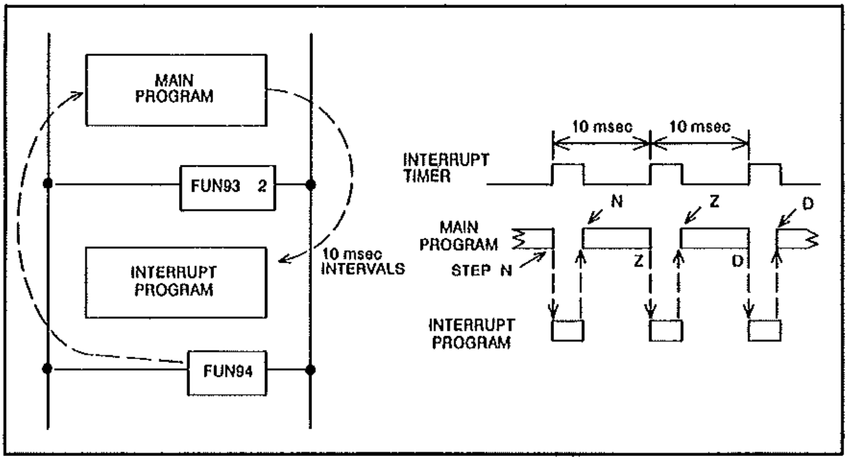


Figure 6.29 - Interrupt Command Operation

6.4.11 Subroutine Commands

- FUN42 = Call Subroutine Command = CALL
- FUN43 = Subroutine Command = SB
- FUN44 = Return from Subroutine Command = RTS
- These commands allow up to 64 subroutine programs in conjunction with the main program. Subroutine programs follow the main program in memory. See figure 6.30.
- FUN42 is used to initiate the execution of a specified subroutine program from the main program. FUN42 requires a start contact. When the start contact is true, execution control is transferred from that point in the main program to the specified subroutine program (0 to 63).
- FUN43 identifies the start point of the subroutine program. Each subroutine must be terminated with a FUN44 command. FUN44 returns processor execution to the step in the main program that follows the FUN42 command. See figure 6.31.
- FUN04, FUN05, FUN06, FUN07, FUN08, and FUN09 may not be used within the subroutine commands.

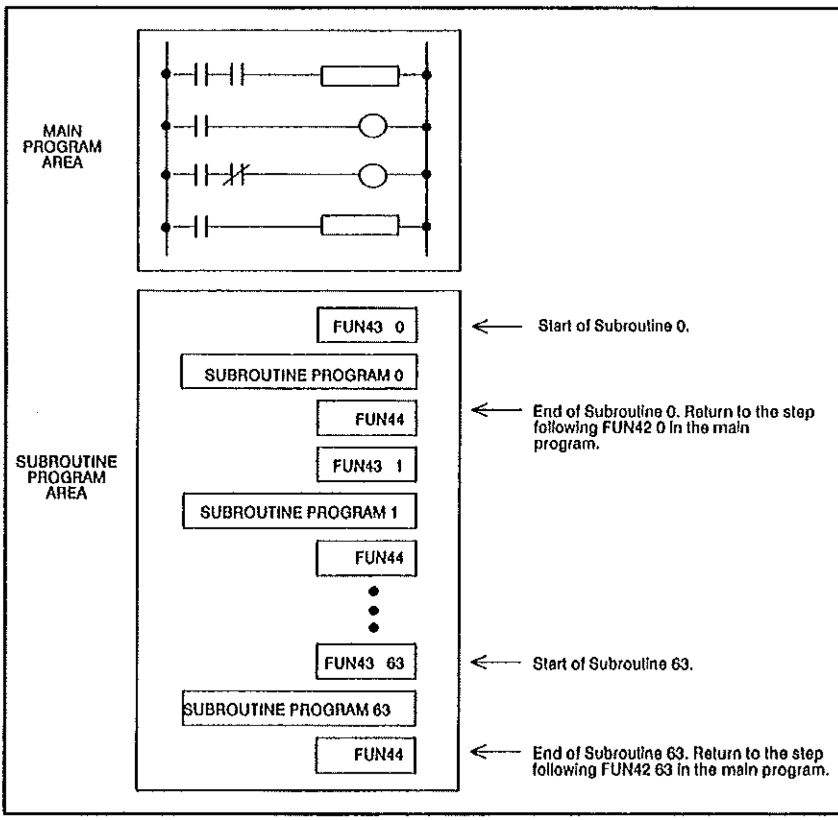
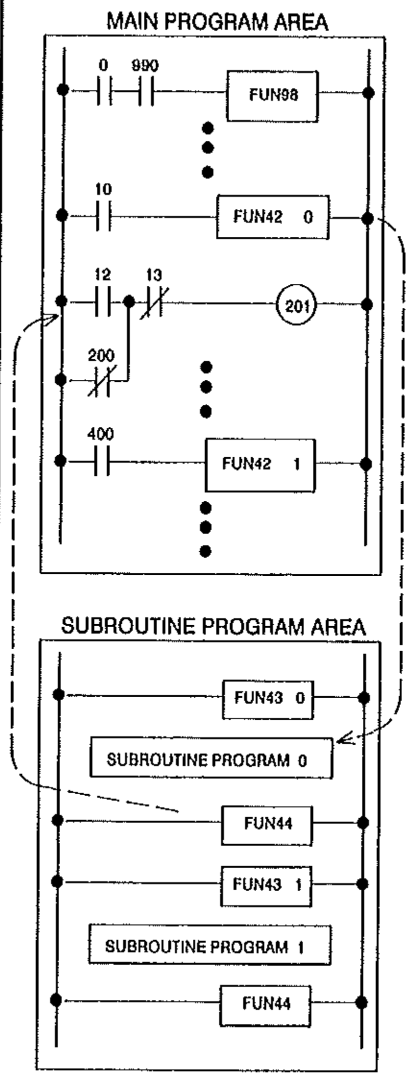


Figure 6.30 - Organization of Main and Subroutine Program Areas



Coding		Remarks
ORG	0	
AND	990	
FUN98		
• • •		
ORG	10	When contact 10 is on, call subroutine program 0. (Exit the main program and execute subroutine 0.)
FUN42	0	
ORG	12	
OR NOT	200	
AND NOT	13	
OUT	201	
• • •		
ORG	400	
FUN42	1	

Coding		Remarks
FUN43	0	When FUN42 0 is executed in the main program, subroutine 0 is called and executed.
SUBROUTINE 0		
FUN44		When execution of subroutine program 0 is completed, FUN44 returns processor execution control to the step following the call command for subroutine 0.
FUN43	1	
SUBROUTINE 1		
FUN44		

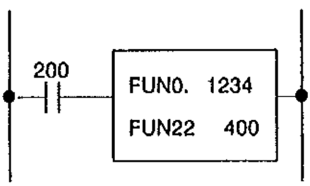
Figure 6.31 - Example of FUN42, FUN43, and FUN44

6.5 Advanced Data Manipulation Commands

Advanced data manipulation commands provide the required logic for more complex control applications. They include four-function math commands, data compare commands, data transfer commands, data conversion commands, and shift register commands.

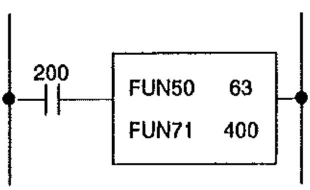
6.5.1 Constant Value Load Commands

- FUN0. = Constant Value Word Load Command = LOAD C
- FUN50 = Constant Value Byte Load Command = LOAD CB
- FUN0. loads a BCD constant value word into the arithmetic register. Refer to figure 6.32. The decimal point indicates a constant value.
- FUN50 loads a BCD constant value byte into the lower half of the arithmetic register (AR_L). The upper half of the arithmetic register (AR_H) retains the previous value. See figure 6.33.
- The arithmetic register is used by the processor as a temporary storage area for data manipulation.



Coding		Remarks
ORG	200	
FUN 0.	1234	"1234" → AR
FUN 22	400	AR → 400~415

Figure 6.32 - Example of FUN0.



Coding		Remarks
ORG	200	Load constant value byte 63 into the lower half of AR (AR_L)
FUN50	63	
FUN71	400	Output constant value byte 63 to internal output byte 400.

Figure 6.33 - Example of FUN50

- In figure 6.34 a 4-digit BCD value is loaded into the arithmetic register using the FUN0. command. When the FUN22 command is executed, the value in the arithmetic register is sent to internal output bits 400 to 415.

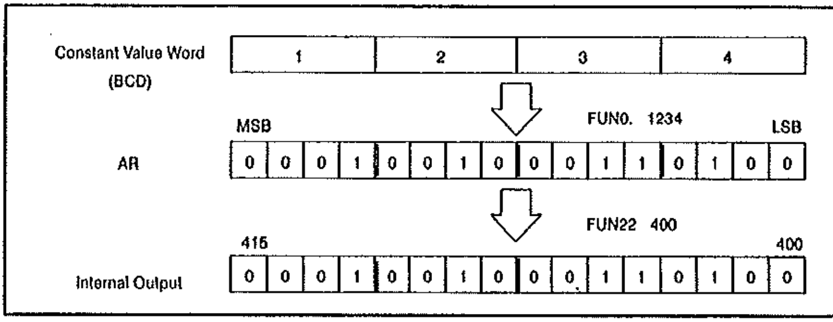


Figure 6.34 - Constant Value Word Load

- In figure 6.35, a 2-digit BCD value is loaded into the lower half of the arithmetic register using the FUN50 command. When the FUN71 command is executed, the value in the lower half of the arithmetic register is sent to internal output byte 400.

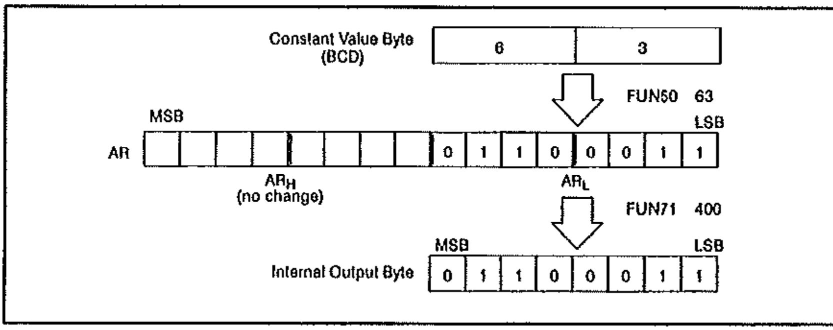
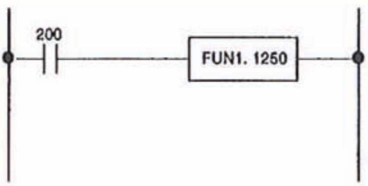


Figure 6.35 - Constant Value Byte Load

6.5.2 Constant Value Addition Commands

- FUN1. = Constant Value BCD Addition Command = ADD C
- FUN51 = Constant Value Binary Addition Command = ADD BNRC
- FUN1. adds two BCD numbers. The specified BCD constant is added to the contents of the arithmetic register. The result is then stored in the arithmetic register. See figure 6.36.
- If the addition result is greater than 9999, the carry bit is turned on and the previous value in the arithmetic register remains unchanged.
- If non-BCD data is added together by FUN1., the arithmetic register data and the carry bit status will be indeterminate.
- FUN51 adds two binary numbers. The specified binary constant (0 to 999) is added to the contents of the arithmetic register. The binary result is then stored in the arithmetic register. See figure 6.37.
- If the binary addition result is greater than 9999, the carry bit is turned on (C=1) and the binary value of the lower four digits of the result are loaded into the arithmetic register.



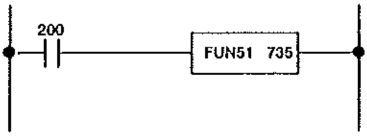
Coding		Remarks
ORG	200	Constant value 1250 is added to the contents of the AR.
FUN1.	1250	

A. If the result is ≤ 9999, the carry bit = 0 and the value of 1250 plus the contents of the AR is placed in the AR.

B. If the result is > 9999, the carry bit = 1 and the contents of the AR are not changed.

Figure 6.36 - Example of FUN1.

Constant Value Addition Commands (Continued)



Coding		Remarks
ORG	200	Binary constant 735 is added to the contents of the AR.
FUN51	735	

A. If the result is ≤ 9999, the carry bit = 0 and the result is placed in the AR.

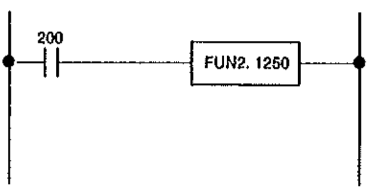
B. If the result is > 9999, the carry bit = 1 and the binary value of the lower four digits of the result is placed in the AR.

Figure 6.37 - Example of FUN51

- Note that the arithmetic register and the carry bit are cleared at the beginning of a scan and are updated during the scan.
- If the addition operation (FUN1. or FUN51) is to be performed only once when an input is enabled, instead of every scan, use FUN00 (Edge Detection command) before the addition rung.

6.5.3 Constant Value Subtraction Commands

- FUN2. = Constant Value BCD Subtraction Command = SUB C
- FUN52 = Constant Value Binary Subtraction Command = SUB BNRC
- FUN2. subtracts two BCD numbers. The specified BCD constant value is subtracted from the contents of the arithmetic register. The BCD result is then stored in the arithmetic register. See figure 6.38.
- If the subtraction result is negative, the carry bit is turned on and the previous value remains unchanged.
- If non-BCD data is subtracted by FUN2., the arithmetic register's data and the status of the carry bit will be indeterminate.
- FUN52 subtracts two binary numbers. The specified binary constant (0 to 999) is subtracted from the contents of the arithmetic register. The binary result is then stored in the arithmetic register. See figure 6.39.
- If the binary subtraction result is a negative number, the carry bit is turned on and the result is loaded into the arithmetic register in two's complement notation.

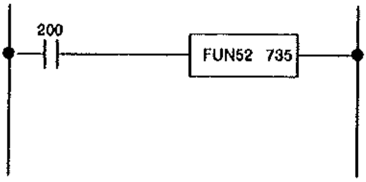


Coding		Remarks
ORG	200	Constant value 1250 is subtracted from the contents of the AR.
FUN2.	1250	

A. If the result is ≥ 0, the carry bit = 0 and the result of 1250 subtracted from the contents of the AR is placed in the AR.

B. If the result is negative, the carry bit = 1 and the contents of the AR are not changed.

Figure 6.38 - Example of FUN2.



Coding		Remarks
ORG	200	Binary constant 735 is subtracted from the contents of the AR.
FUN52	735	

A. If the result is $\geq$ 0, the carry bit = 0 and the result is placed in the AR.

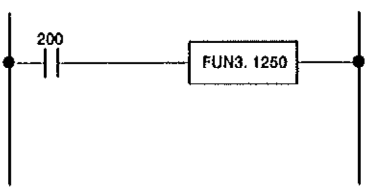
B. If the result is negative, the carry bit = 1 and the result is placed in the AR expressed in two's compliment notation.

Figure 6.39 - Example of FUN52

- Note that the arithmetic register and the carry bit are cleared at the beginning of a scan and are updated during the scan.
- If the subtraction operation (FUN2. or FUN52) is to be performed only once when an input is enabled, instead of every scan, use FUN00 (Edge Detection command) before the subtraction rung.

6.5.4 Constant Value Multiplication Commands

- FUN3. = Constant Value Multiplication Command = MUL C
- FUN53 = Constant Value Binary Multiplication Command = MUL BNRC
- FUN3. multiplies two BCD numbers. The specified BCD constant value is multiplied with the contents of the arithmetic register. The BCD result is then stored in the arithmetic register. See figure 6.40.
- If the result is greater than 9999, the carry bit is turned on and the previous value remains unchanged.
- If non-BCD data is multiplied by FUN3., the arithmetic register's data and the status of the carry bit will be indeterminate.
- FUN53 multiplies two binary numbers. The specified binary constant (0 to 999) is multiplied with the contents of the arithmetic register. The binary product is then stored in the arithmetic register. See figure 6.41.
- If the binary product is > 9999 (exceeds four digits), the carry bit is turned on, the lower four digits of the product are placed in the arithmetic register, and the remaining digits are placed in the extension register.



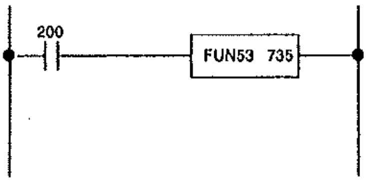
Coding		Remarks
ORG	200	Constant value 1250 is multiplied with the contents of the AR.
FUN3.	1250	

A. If the result is < 9999, the carry bit = 0 and the BCD product of 1250 multiplied with the contents of the AR is placed in the AR.

B. If the result is > 9999, the carry bit = 1 and the contents of the AR are not changed.

Figure 6.40 - Example of FUN3.

Constant Value Multiplication Commands (Continued)



Coding		Remarks
ORG	200	Binary constant 735 is multiplied with the contents of the AR.
FUN53	735	

A. If the result is ≤ 9999, the carry bit = 0 and the product is placed in the AR.

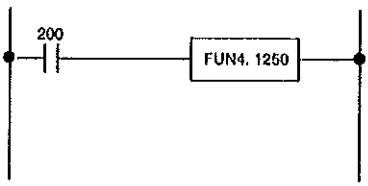
B. If the result is > 9999 (exceeds 4 digits), the carry bit = 1, the lower four digits of the product are placed in the AR, and the remaining digits are placed in the ER.

Figure 6.41 - Example of FUN53

- Note that the arithmetic register and the carry bit are cleared at the beginning of a scan and are updated during the scan.
- If the multiplication operation (FUN3. or FUN53) is to be performed only once when an input is enabled, instead of every scan, use FUN00 (Edge Detection command) before the multiplication rung.

6.5.5 Constant Value Division Commands

- FUN4. = Constant Value BCD Division Command = DIV C
- FUN54 = Constant Value Binary Division Command = DIV BNRC
- FUN4. divides two BCD numbers. The contents of the arithmetic register are divided by the specified BCD constant value. The BCD result is then stored in the arithmetic register. See figure 6.42.
- If the arithmetic register's contents are divided by 0, the carry bit is set to one and the previous value remains unchanged. The remainder is always ignored.
- If non-BCD data is divided by FUN4., the arithmetic register's data and the status of the carry bit will be indeterminate.
- FUN54 divides the contents of the arithmetic register by a specified binary constant value (1 to 999). The binary quotient is placed in the arithmetic register and the binary remainder is stored in the extension register. See figure 6.43.
- If the contents of the arithmetic register are divided by zero, the carry bit is turned on and the previous values in the arithmetic and extension registers remain unchanged.



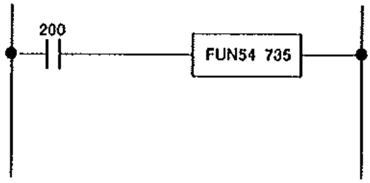
Coding		Remarks
ORG	200	Contents of the AR are divided by constant value 1250.
FUN4.	1250	

A. If the specified BCD constant value = 1 or < 9999, the carry bit = 0, the quotient is stored in the AR, and the remainder is disregarded.

B. If the specified BCD constant value is zero, the carry bit = 1 and the contents of the AR are not changed.

Figure 6.42 - Example of FUN4.

Constant Value Division Commands (Continued)



Coding		Remarks
ORG	200	Contents of the AR are divided by binary constant value 735.
FUN54	735	

A. If the specified binary constant is = 1 or ≤ 999, and the carry bit = 0, the binary quotient is placed in the AR and the binary remainder is stored in the ER.

B. If the specified binary constant value = 0, and the carry bit = 1, the previous values in the AR and the ER remain unchanged.

Figure 6.43 - Example of FUN54

- Note that the arithmetic register and the carry bit are cleared at the beginning of a scan and are updated during the scan.
- If the division operation (FUN4. or FUN54) is to be performed only once when an input is enabled, instead of every scan, use FUN00 (Edge Detection command) before the division rung.

6.5.6 Constant Value AND Commands

- FUN5. = Constant Value BCD Word AND Command = WAND C
- FUN55 = Constant Value Binary Byte AND Command = BANDI
- FUN5. logically ANDs together the specified BCD constant value and the contents of the arithmetic register according to the truth table in figure 6.44. The result is then stored in the arithmetic register. See figure 6.45.
- The carry bit is not affected by FUN5.
- FUN55 logically ANDs together the specified binary constant value (0 to 255) and the contents of the lower eight bits (AR_L) of the arithmetic register according to the truth table in figure 6.44. The result is stored in the lower eight bits of the arithmetic register. The contents of the upper eight bits (AR_H) remain unchanged. See figure 6.46.
- The carry bit is not affected by FUN55.

AR bit	Constant bit		AND'ed Result
0	0	=	0
0	1	=	0
1	0	=	0
1	1	=	1

Figure 6.44 - AND Truth Table

Example: FUN5.1250

Result: The BCD contents of the AR (5761) are ANDed with BCD constant value 1250. The result (1240 BCD) is stored in the AR.

5	7	6	1	
0 1 0 1	0 1 1 1	0 1 1 0	0 0 0 1	AR = 5761 (BCD)

1	2	5	0	
0 0 0 1	0 0 1 0	0 1 0 1	0 0 0 0	FUN5. = 1250 (BCD Constant)

1	2	4	0	
0 0 0 1	0 0 1 0	0 1 0 0	0 0 0 0	Result in AR = 1240 (BCD)

Figure 6.45 - Example of FUN5.

Example: FUN55 136

Result: The binary contents of the AR (95) are ANDed with binary constant value 136. The result in binary (8) is stored in the lower eight bits of the (AR_L).

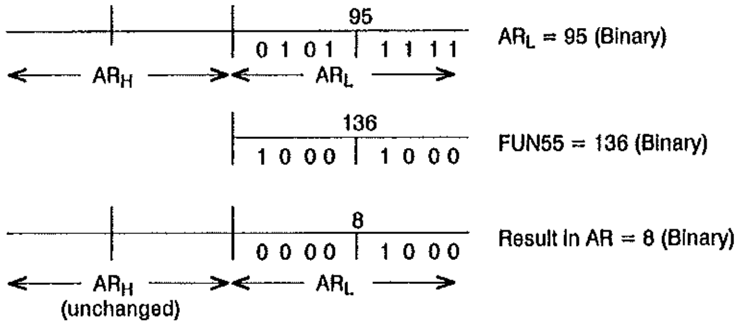


Figure 6.46 - Example of FUN55

6.5.7 Constant Value OR Commands

- FUN6. = Constant Value BCD Word OR Command = WOR C
- FUN56 = Constant Value Binary Byte OR Command = BOR I
- FUN6. logically ORs together the specified BCD constant value and the contents of the arithmetic register according to the truth table in figure 6.47. The result is then stored in the arithmetic register. See figure 6.48.
- The carry bit is not affected by FUN6.
- FUN56 logically ORs together the specified binary constant value (0 to 255) and the contents of the lower eight bits (AR_L) of the arithmetic register according to the truth table in figure 6.47. The result is stored in the lower eight bits of the arithmetic register. The contents of the upper eight bits (AR_U) of the arithmetic register remain unchanged. See figure 6.49.
- The carry bit is not affected by FUN56.

AR bit	Constant bit		OR'ed Result
0	0	=	0
0	1	=	1
1	0	=	1
1	1	=	1

Figure 6.47 - OR Truth Table

Example: FUN6.1250

Result: The BCD contents of the AR (5761) are ORed with BCD constant value 1250. The result (5771 BCD) is stored in the AR.

5	7	6	1	
0 1 0 1	0 1 1 1	0 1 1 0	0 0 0 1	AR = 5761 (BCD)
1	2	5	0	
0 0 0 1	0 0 1 0	0 1 0 1	0 0 0 0	FUN6. = 1250 (BCD Constant)
5	7	7	1	
0 1 0 1	0 1 1 1	0 1 1 1	0 0 0 1	Result in AR = 5771 (BCD)

Figure 6.48 - Example of FUN6.

Example: FUN56 136

Result: The binary contents of the AR_L (95) are ORed with binary constant 136. The result in binary (223) is stored in the lower eight bits of the (AR_L).

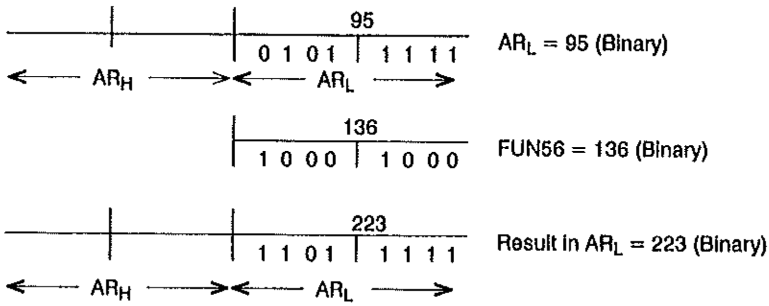
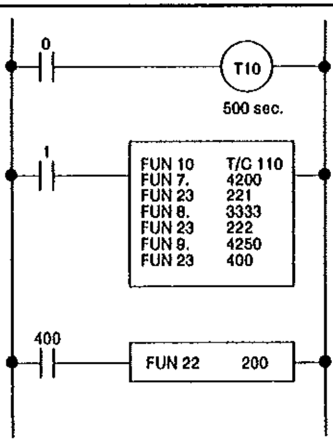


Figure 6.49 - Example of FUN56

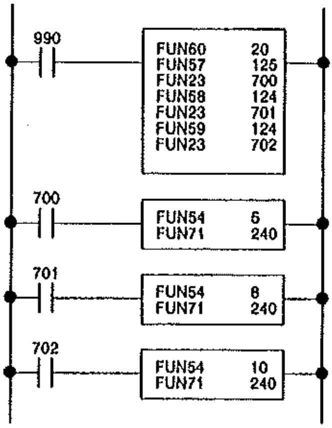
6.5.8 Constant Value Compare Commands

- FUN7. = Constant Value Word Compare ($\geq$) Command = CMP$\geq$C
- FUN8. = Constant Value Word Compare (=) Command = CMP=C
- FUN9. = Constant Value Word Compare (<) Command = CMP<C
- FUN57 = Constant Value Byte Compare ($\geq$) Command = BCPHI
- FUN58 = Constant Value Byte Compare (=) Command = BCPEI
- FUN59 = Constant Value Byte Compare (<) Command = BCPLI
- FUN7., FUN8., and FUN9. compare the contents of the arithmetic register with the specified BCD constant value word (0-9999). Refer to figure 6.50.
- FUN57, FUN58, and FUN59 compare the contents of the lower eight bits (AR_L) of the arithmetic register with the specified binary constant value byte (0 to 255). Refer to figure 6.51.
- When the result of the comparison is true, the carry bit is equal to 1.
- When the result of the comparison is false, the carry bit is equal to 0.
- Use FUN23 (Out Carry Bit command) to have the result of the comparison control an internal or external output.



Coding		Remarks
ORG	0	
OUT T/C10	500	
ORG	1	
FUN 10	T/C 110	Load current value of Timer 10 into the AR
FUN 7.	4200	If AR $\geq$ 4200, C = 1
FUN 23.	221	If C = 1, turn 221 on
FUN 8.	3333	If AR = 3333, C = 1
FUN 23.	222	If C = 1, turn 222 on
FUN 9.	4250	If AR $\leq$ 4250, C = 1
FUN 23	400	If C = 1, turn 400 on
ORG	400	When 400 is on, transfer contents of AR to external outputs 200 to 215
FUN 22	200	

Figure 6.50 - Example of FUN7., FUN8., and FUN9.



Coding		Remarks
ORG FUN60	990 20	Load 8-bit analog input value from slot 2., channel 0 into the AR_L.
FUN57 FUN23 FUN58 FUN23 FUN59 FUN23	125 700 124 701 124 702	If $AR_L > 125$, C = 1 When $\overline{C}$ = 1, turn 700 on. If $AR_L = 124$, C = 1 When C = 1, turn 701 on. If $AR_L < 124$, C = 1 When C = 1, turn 702 on.
ORG FUN54	700 5	When 700 is on, divide the contents of AR_L by 5 and store the quotient in AR_L.
FUN71	240	Output the value from AR_L to 8-bit analog output channel 0, slot 3.
ORG FUN54	701 8	When 701 is on, divide the contents of AR_L by 8 and store the quotient in AR_L.
FUN71	240	Output the value from AR_L to 8-bit analog output channel 0, slot 3.
ORG FUN54	702 10	When 702 is on, divide the contents of AR_L by 10 and store the quotient in AR_L.
FUN71	240	Output the value from AR_L to 8-bit analog output channel 0, slot 3.

Figure 6.51 - Example of FUN57, FUN58, and FUN59

- In figure 6.52, when the input is on, the current value of timer 10 is loaded into the arithmetic register and compared with a constant word value for a greater than or equal to condition.

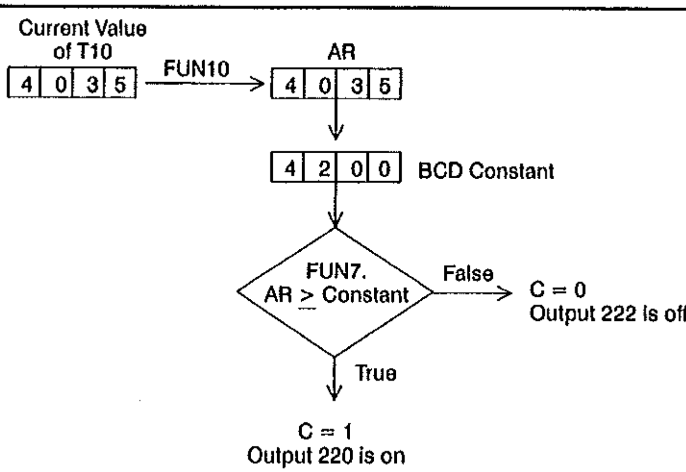


Figure 6.52 - Constant Value Word Comparison (≥)

Constant Value Compare Commands (Continued)

- You can output the results of the compare operation in bit format. Note that the status of the carry bit is also output. See figure 6.53.

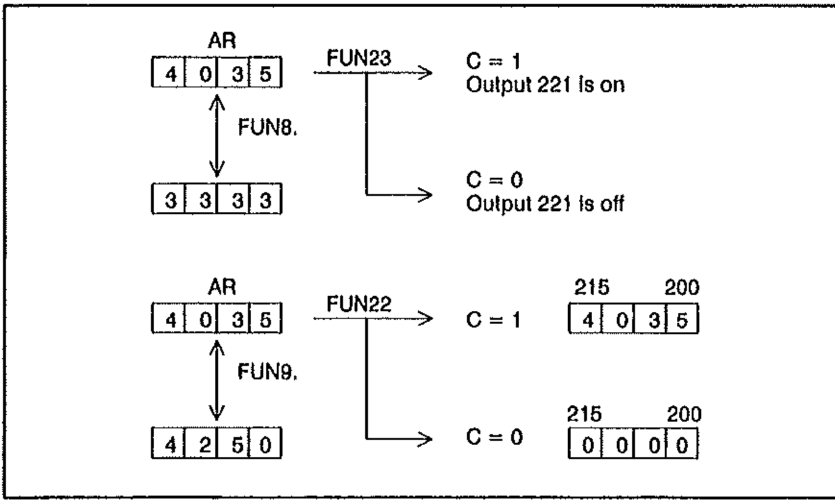


Figure 6.53 - Constant Value Word Comparisons (=,<)

- Figure 6.54 shows the status of the compare commands' carry bit.

	AR > constant	AR= constant	AR < constant
FUN7. =	C = 1	C = 1	C = 0
FUN8. =	C = 0	C = 1	C = 0
FUN9. =	C = 0	C = 0	C = 1

	AR_L > constant	AR_L= constant	AR_L < constant
FUN57 =	C = 1	C = 1	C = 0
FUN58 =	C = 0	C = 1	C = 0
FUN59 =	C = 0	C = 0	C = 1

Figure 6.54 - Carry Bit Status of the Compare Commands

- In figure 6.55, when the input is on, the analog input value (binary) is loaded into AR_L and compared with a constant binary value for a greater than or equal to condition.

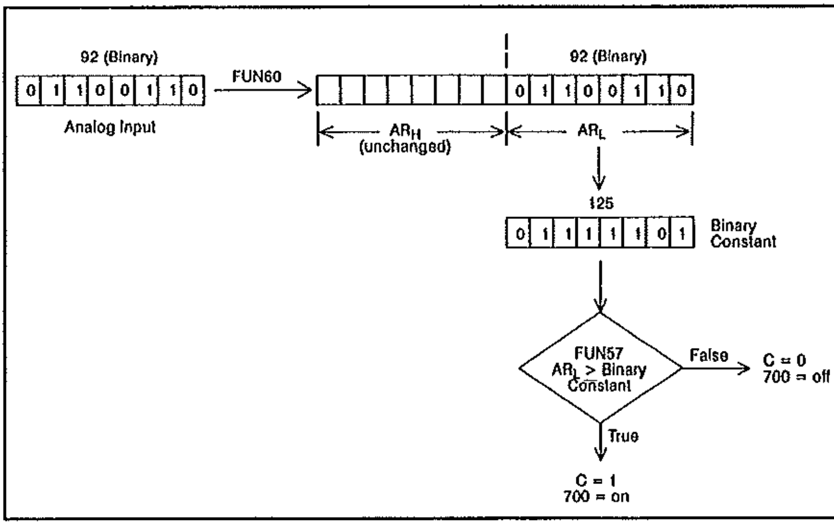


Figure 6.55 - Constant Value Byte Comparison (≥)

- In figure 6.56, the value in AR_L is compared with a binary constant (124), for an equal to condition (FUN58) and then for a less than condition (FUN59). The rest of these comparisons control the status of the carry bit and the respective output.

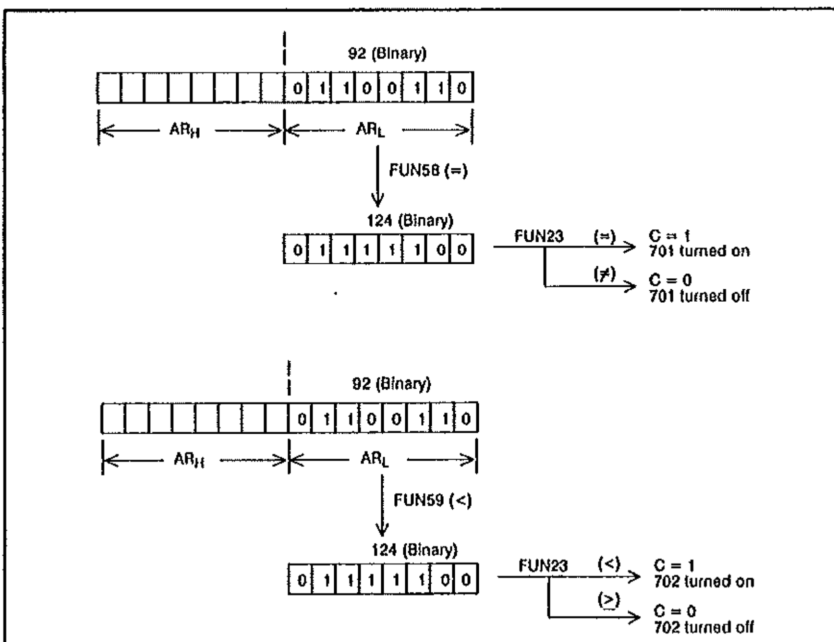


Figure 6.56 - Constant Value Byte Comparisons (=,<)

6.5.9 Word Load (Two Bytes) Command

- FUN10 = Word Load (2 bytes) Command = WLOAD
- This command loads two consecutive bytes of data into the arithmetic register. The carry bit does not change. See figure 6.57.
- This command can also be used to load the current or preset value of a timer or counter into the arithmetic register.

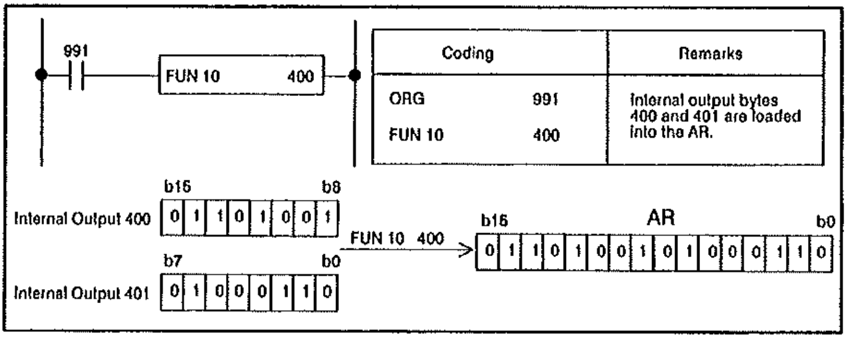


Figure 6.57 - FUN10 Word Load of Two Internal Output Bytes Into the AR

- This command can also load analog input values, counter module current values, remote I/O data, and network data into the arithmetic register. See figure 6.58.

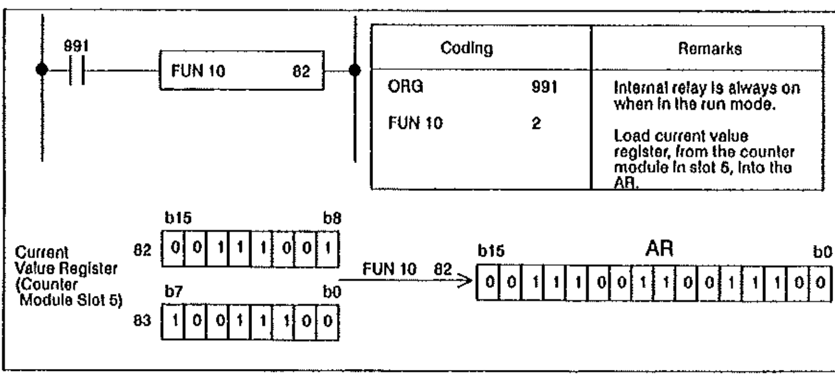


Figure 6.58 - FUN10 Word Load of the Counter Module's Current Value Register into the AR

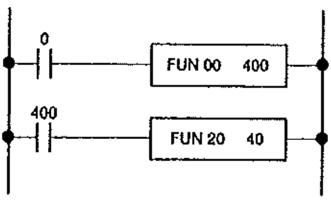
6.5.10 Word Load (Sixteen Bits) Command

- FUN20 = Word Load (Sixteen Bits) Command = WLOAD B
- This command simultaneously loads 16 consecutive bits into the arithmetic register as a single word. The carry bit does not change. See figure 6.59.

Data to be Loaded	Example	Result
External input or internal output or timer/counter contacts	FUN20 40 (bit to word operation)	External inputs 40 to 55 are loaded into the AR.

Figure 6.59 - Example of FUN20

Figure 6.60 shows an example of loading a value from external thumbwheel switches into the arithmetic register.



Coding		Remarks
ORG	0	One-shot to load new thumbwheel data.
FUN 00	400	
ORG	400	Load external inputs 40 to 55 (new thumbwheel data) into the AR.
FUN 20	40	

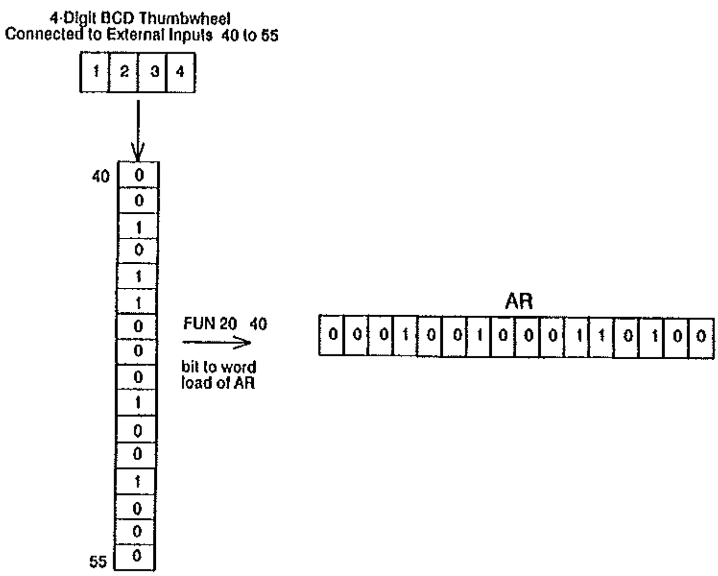
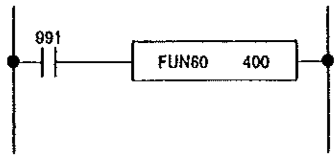


Figure 6.60 - Loading Data from Thumbwheel Switches

6.5.11 Byte Load Command

- FUN60 = Byte Load Command = BLOAD
- This command loads one byte (8 bits) of data into the lower byte (AR_L) of the arithmetic register. The carry bit and the upper byte (AR_H) of the arithmetic register do not change.
- This command may be used to load external byte data, such as the binary value of an 8-bit analog input signal. It can also be used to load internal output byte data, such as the lower 8-bits of timer/counter preset/current values or internal output bytes such as 400, 401, etc. See figures 6.61 and 6.62.



Coding		Remarks
ORG	991	Load internal output byte 400 into the AR_L
FUN60	400	

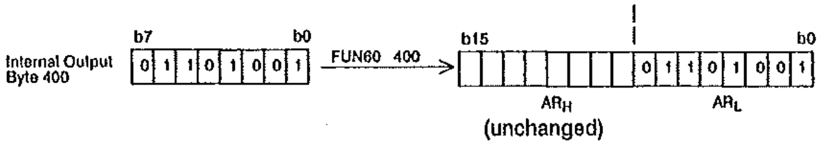
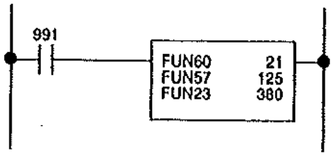


Figure 6.61 - FUN60 Load of an Internal Output Byte into AR_L

Byte Load Command (Continued)



Coding		Remarks
ORG	991	
FUN60	21	Load the 8-bit analog input binary value from slot 2, channel 0 into the AR_L.
FUN57	125	If $AR_L \geq 125$, C = 1 else C = 0.
FUN23	380	When C = 1, turn external output 380 on. When C = 0, turn external output 380 off.

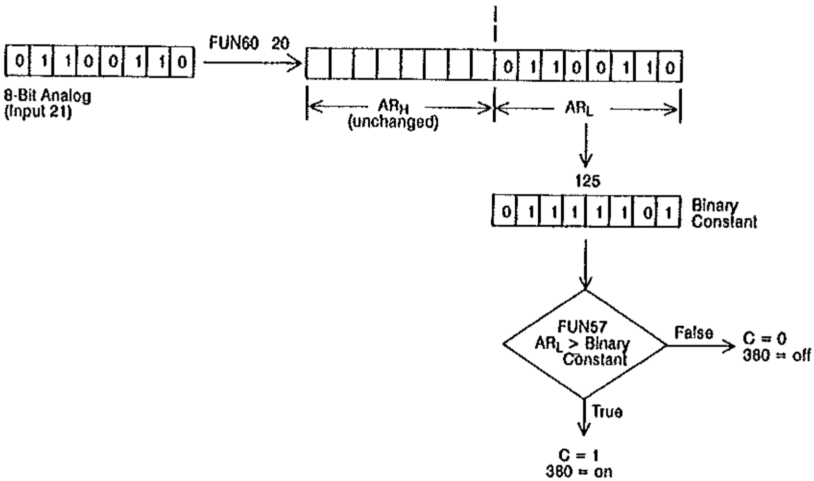


Figure 6.62 - FUN60 Load of an 8-Bit Analog Input Value into the AR_L

6.5.12 Addition Commands

- FUN11 = BCD Addition Command = ADD
- FUN61 = Binary Addition Command = ADD BNR
- FUN11 adds 2 BCD numbers. The BCD values in the specified addresses are added to the contents of the arithmetic register. The result is then stored in the arithmetic register as a BCD value. See figure 6.63.
- If the results are greater than 9999, the carry bit is turned on and the previous value remains unchanged.
- If non-BCD data is added together by FUN11, the arithmetic register's data and the status of the carry bit will be indeterminate.
- FUN61 adds 2 binary numbers. The specified binary word is added to the contents of the arithmetic register. The result is stored in the arithmetic register as a binary value. See figure 6.64.
- If the binary addition result is greater than 9999, the carry bit is turned on and the binary value of the lower four digits of the result are loaded into the arithmetic register.

Data to be Added	Example	Result
External input word or Internal output bytes	FUN11 400 (word operation)	The BCD value formed by internal output bytes 400 and 401 is added to the contents of the AR. The result is stored in the AR.
Timer or counter current or preset values	FUN11 T/C110	Current value of T/C10 is added to the contents of the AR. The result is stored in the AR.

Figure 6.63 - Example of FUN11

Data to be Added	Example	Result
External input word or Internal output bytes	FUN61 400	The binary value formed by internal output bytes 400 and 401 is added to the contents of the AR. The result is stored in the AR.
Timer or counter current or preset values	FUN61 T/C110	Current value of T/C10 is added to the contents of the AR. The binary result is stored in the AR.

Figure 6.64 - Example of FUN61

- Note that the arithmetic register and the carry bit are cleared at the beginning of a scan and are updated during the scan.
- If the addition operation (FUN11) is to be performed only once when an input is enabled, instead of every scan, use FUN00 (Edge Detection command) before the addition rung.

6.5.13 Subtraction Commands

- FUN12 = BCD Subtraction Command = SUB
- FUN62 = Binary Subtraction Command = SUB BNR
- FUN12 subtracts 2 BCD numbers. The BCD values in the specified addresses are subtracted from the contents of the arithmetic register. The result is then stored in the arithmetic register as a BCD value. See figure 6.65.
- If the results are negative, the carry bit is turned on and the previous value remains unchanged.
- If non-BCD data is subtracted by FUN12, the arithmetic register's data and the status of the carry bit will be indeterminate.
- FUN62 subtracts two binary numbers. The specified binary word is subtracted from the contents of the arithmetic register. The result is stored in the arithmetic register as a binary value. See figure 6.66.
- If the binary subtraction result is a negative number, the carry bit is turned on and the result is loaded into the arithmetic register in two's complement notation.

Data to be Subtracted	Example	Result
External input word or Internal output bytes	FUN12 400 (word operation)	The BCD value formed by Internal output bytes 400 and 401 is subtracted from the contents of the AR. The result is stored in the AR.
Timer or counter current or preset values	FUN12 T/C110	Current value of T/C10 is subtracted from the contents of the AR. The result is stored in the AR.

Figure 6.65 - Example of FUN12

Data to be Subtracted	Example	Result
External input word or Internal output bytes	FUN62 400	The binary value formed by Internal output bytes 400 and 401 is subtracted from the contents of the AR. The result is stored in the AR.
Timer or counter current or preset values	FUN62 T/C110	Current value of T/C10 is subtracted from the contents of the AR. The binary result is stored in the AR.

Figure 6.66 - Example of FUN62

- Note that the arithmetic register and the carry bit are cleared at the beginning of a scan and are updated during the scan.
- If the subtraction operation (FUN12) is to be performed only once when an input is enabled, instead of every scan, use FUN00 (Edge Detection command) before the subtraction rung.

6.5.14 Multiplication Commands

- FUN13 = BCD Multiplication Command = MUL
- FUN63 = Binary Multiplication Command = MUL BNR
- FUN13 multiplies 2 BCD numbers. The BCD values in the specified addresses are multiplied with the contents of the arithmetic register. The result is then stored in the arithmetic register as a BCD value. See figure 6.67.
- If the results are greater than 9999, the carry bit is turned on and the previous value remains unchanged.
- If non-BCD data is multiplied by FUN13, the arithmetic register's data and the status of the carry bit will be indeterminate.
- FUN63 multiplies two binary numbers. The specified binary word is multiplied with the contents of the arithmetic register. The binary product is then stored in the arithmetic register. See figure 6.68.
- If the binary product is greater than 9999 (exceeds 4 digits), the carry bit is turned on, the lower 4 digits of the product are in the arithmetic register, and the upper 4 digits of the product are placed in the extension register.

Data to be Multiplied	Example	Result
External input word or Internal output bytes	FUN13 400 (word operation)	The BCD value formed by internal output bytes 400 and 401 is multiplied by the contents of the AR. The result is stored in the AR.
Timer or counter current or preset values	FUN13 T/C110	Current value of T/C10 is multiplied by the contents of the AR. The result is stored in the AR.

Figure 6.67 - Example of FUN13

Data to be Multiplied	Example	Result
External input word or Internal output bytes	FUN63 400	The binary value formed by internal output bytes 400 and 401 is multiplied by the contents of the AR. The binary product is stored in the AR.
Timer or counter current or preset values	FUN63 T/C110	Current value of T/C10 is multiplied by the contents of the AR. The binary product is stored in the AR.

Figure 6.68 - Example of FUN63

- Note that the arithmetic register and the carry bit are cleared at the beginning of a scan and are updated during the scan.
- If the multiplication operation (FUN13) is to be performed only once when an input is enabled, instead of every scan, use FUN00 (Edge Detection command) before the multiplication rung.

6.5.15 Division Commands

- FUN14 = BCD Division Command = DIV
- FUN64 = Binary Division Commmand = DIV BNR
- FUN14 divides 2 BCD numbers. The contents of the arithmetic register are divided by the BCD value in the specified addresses. The results are then stored in the arithmetic register as a BCD value. See figure 6.69.
- If the arithmetic register's contents are divided by 0, the carry bit is turned on and the previous value remains unchanged. The remainder is always ignored.
- If non-BCD data is divided by FUN14, the arithmetic register's data and the status of the carry bit will be indeterminate.
- FUN64 divides the contents of the arithmetic register by a specified binary word value. The binary quotient is placed in the arithmetic register. The binary remainder is stored in the extension register. See figure 6.70.
- If the contents of the arithmetic register are divided by zero, the carry bit is turned on and the previous values in the arithmetic and extension registers remain unchanged.

Specified Data Type	Example	Result
External input word or Internal output bytes	FUN14 400 (word operation)	Contents of the AR are divided by the BCD value formed by internal output bytes 400 and 401. The result is stored in the AR.
Timer or counter current or preset values	FUN14 T/C110	Contents of the AR are divided by the current value of T/C10. The result is stored in the AR.

Figure 6.69 - Example of FUN14

Specified Data Type	Example	Result
External input word or Internal output bytes	FUN64 400	Contents of the AR are divided by the binary value formed by internal output bytes 400 and 401. The binary quotient is placed in the AR. The binary remainder is placed in the ER.
Timer or counter current or preset values	FUN64 T/C110	Contents of the AR are divided by the current value of the T/C10. The binary quotient is placed in the AR. The binary remainder is placed in the ER.

Figure 6.70 - Example of FUN64

- Note that the arithmetic register and the carry bit are cleared at the beginning of a scan and are updated during the scan.
- If the division operation (FUN14) is to be performed only once when an input is enabled, instead of every scan, use FUN00 (Edge Detection command) before the division rung.

6.5.16 Word AND Command

- FUN15 = Word AND Command = WAND
- This command logically ANDs together the value in the specified addresses and the contents of the arithmetic register. The result is stored in the arithmetic register. See figures 6.71 and 6.72.

Data to be ANDed	Example	Result
External input word or Internal output bytes	FUN15 400 (word operation)	Internal output bytes 400 and 401 are ANDed with the contents of the AR. The result is stored in the AR.
Timer or counter current or preset values	FUN15 T/C110	Current value of T/C10 is ANDed with the contents of the AR. The result is stored in the AR.

8				F				4				D				
1	0	0	0	1	1	1	1	0	1	0	0	1	1	0	1	= AR

2				5				1				6				
0	0	1	0	0	1	0	1	0	0	0	1	0	1	1	0	= FUN15 T/C110 (Current Value of T/C10)

0				5				0				4				
0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	= Results Stored in AR

Figure 6.71 - Example of FUN15

AR bit	Data bit		ANDed Result
0	0	=	0
0	1	=	0
1	0	=	0
1	1	=	1

Figure 6.72 - Word AND Truth Table

6.5.17 Word OR Command

- FUN16 = Word OR Command = WOR
- This command logically ORs together the value in the specified addresses and the contents of the arithmetic register. The result is stored in the arithmetic register. See figures 6.73 and 6.74.

Data to be ORed	Example	Result
External input word or Internal output bytes	FUN16 400 (word operation)	Internal output bytes 400 and 401 are ORed with the contents of the AR. The result is stored in the AR.
Timer or counter current or preset values	FUN16 T/C110	Current value of T/C10 is ORed with the contents of the AR. The result is stored in the AR.

8				F				4				D				
1	0	0	0	1	1	1	1	0	1	0	0	1	1	0	1	= AR

2				5				1				6				
0	0	1	0	0	1	0	1	0	0	0	1	0	1	1	0	= FUN16 T/C110 (Current Value of T/C10)

A				F				5				F				
1	0	1	0	1	1	1	1	0	1	0	1	1	1	1	1	= Results Stored in AR

Figure 6.73 - Example of FUN16

AR bit	Data bit		ORed Result
0	0	=	0
0	1	=	1
1	0	=	1
1	1	=	1

Figure 6.74 - Word OR Truth Table

6.5.18 Word Exclusive OR Command

- FUN66 = Word Exclusive OR Command = EXOR
- This command performs a logical exclusive OR operation (XOR) between the value in the specified addresses and the contents of the arithmetic register. The result is stored in the arithmetic register. See figures 6.75 and 6.76.

Data to be XORed	Example	Result
External input word or Internal output bytes	FUN66 400	Internal output bytes 400 and 401 are XORed with the contents of the AR. The results are stored in the AR.
Timer or counter current or preset values	FUN66 T/C110	Current value of T/C10 is XORed with the contents of the AR. The result is stored in the AR.

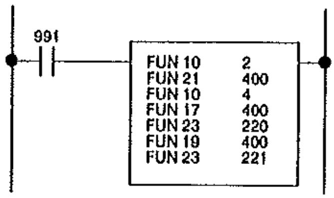
Figure 6.75 - Example of FUN66

AR bit	Data bit		XORed Result
0	0	=	0
0	1	=	1
1	0	=	1
1	1	=	0

Figure 6.76 - Word Exclusive OR Truth Table

6.5.19 Word Compare Commands

- FUN17 = Word Compare (≥) Command = CMP≥
- FUN18 = Word Compare (=) Command = CMP=
- FUN19 = Word Compare (<) Command = CMP<
- These commands compare the contents of the arithmetic register with an external input word, internal output bytes, or a timer/counter value. See figure 6.77.
- When the result of the comparison is true, the carry bit is equal to one.
- When the result of the comparison is false, the carry bit is equal to zero.
- Use FUN23 (Out Carry Bit command) to have the result of the comparison control an internal or external output.



Coding		Remarks
ORG	991	Internal relay always on when in run mode.
FUN 10	2	Load analog input value 1 (slot 1, channel 1) into the AR.
FUN 21	400	Move contents of the AR to bytes 400 and 401.
FUN 10	4	Load analog input value 2 (slot 1, channel 2) into the AR.
FUN 17	400	Compare the value of AR with the contents of bytes 400 and 401. Compare analog value 2 with analog value 1. If the AR (analog value 2) is ≥ bytes 400 and 401 (analog value 1), the carry bit = 1.
FUN 23	220	If the carry bit = 1, turn output 220 on.
FUN 19	400	Compare the value of AR (analog value 2) with the contents of bytes 400 and 401 (analog value 1). If AR < bytes 400 and 401, the carry bit = 1.
FUN 23	221	If the carry bit = 1, turn output 221 on.

Figure 6.77 - Example of the Word Compare Commands

Word Compare Commands (Continued)

- Figure 6.78 shows the status of the compare commands' carry bit.

	AR > Specified Data	AR = Specified Data	AR < Specified Data
FUN17 =	C = 1	C = 1	C = 0
FUN18 =	C = 0	C = 1	C = 0
FUN19 =	C = 0	C = 0	C = 1

Figure 6.78 - Carry Bit Status of the Word Compare Commands

6.5.20 Word Out (Two Bytes) Command

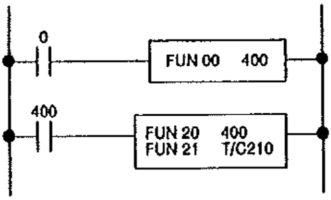
- FUN21 = Word Out (2 Bytes) Command = WOUT
- This command transfers data from the arithmetic register to two consecutive bytes of an external or internal output. This command can also transfer data from the arithmetic register to a timer/counter's current or preset value. See figure 6.79.
- The carry bit does not change.
- Note that when data is transferred to a timer/counter's preset value, the programmed preset value is restored with a system restart.
- This command can be used to:
  - write data to an analog output (> 8 bits of resolution)
  - set the control and preset registers of the High Speed Counter module (M/N 45C982).
  - transfer external output data through the Remote I/O modules.
  - transfer data among processors through the Network modules.
  - store values following a math operation.

Data to be Transferred	Example	Result
External output word or Internal output bytes	FUN21 400 (word operation)	AR's contents are transferred to Internal output bytes 400 and 401.
Timer or counter current or preset values	FUN21 T/C210	AR's contents are transferred to the preset of T10 or C10.

Figure 6.79 - Example of FUN21

Word Out (Two Bytes) Command (Continued)

Figure 6.80 shows an example of how to change a timer preset value from external thumbwheel switches.



Coding		Remarks
ORG	0	One-shot to load new preset value for T/C 10.
FUN 00	400	
ORG	400	Load external inputs 40 to 55 (T/C 10 preset value) into the AR.
FUN 20	40	
FUN 21	T/C210	Transfer the value in the AR (new T/C 10 preset) to the T/C 10 preset memory location.

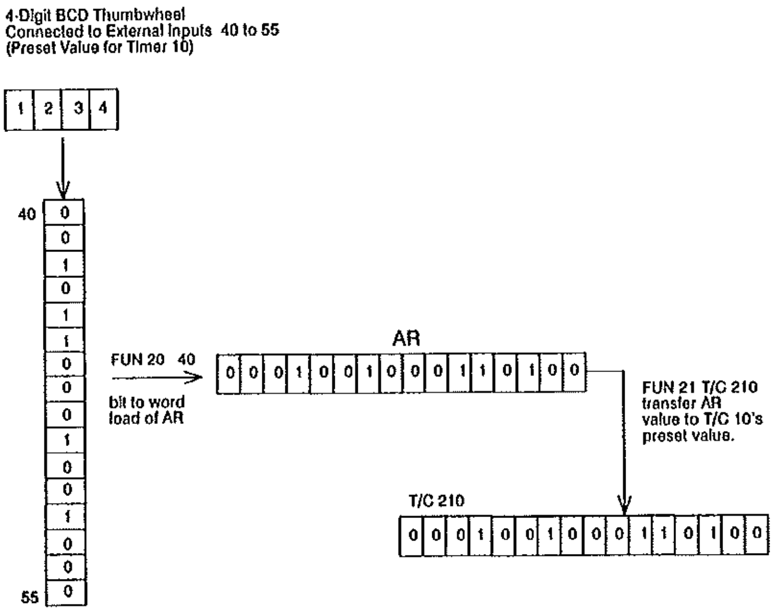


Figure 6.80 - Changing a Timer Preset Value from External Thumbwheel Switches

6.5.21 Word Out (Sixteen Bits) Command

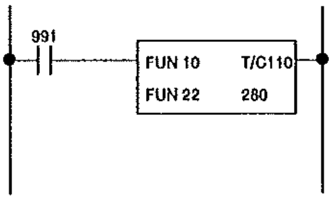
- FUN22 = Word Out (Sixteen Bits) Command = WOUT B
- This command transfers data from the arithmetic register to the sixteen consecutive bits of an external or internal output. See figure 6.81.
- The carry bit does not change.
- This command can be used to:
  - store values in a binary format.
  - transfer data to a 7-segment display via digital outputs.

Figure 6.82 shows how to display a timer's current value on a 7-segment display.

Data to be Transferred	Example	Result
External output word or internal output bytes	FUN22 400 (bit to word operation)	AR's contents are transferred to internal outputs 400 to 415.

Figure 6.81 - Example of FUN22

Word Out (Sixteen Bits) Command (Continued)



Coding		Remarks
ORG	991	Internal contact is always on when in the run mode.
FUN 10	T/C 110	Load current value of T/C 10 into the AR.
FUN 22	280	Transfer current value of T/C 10 from the AR to external output 280-295

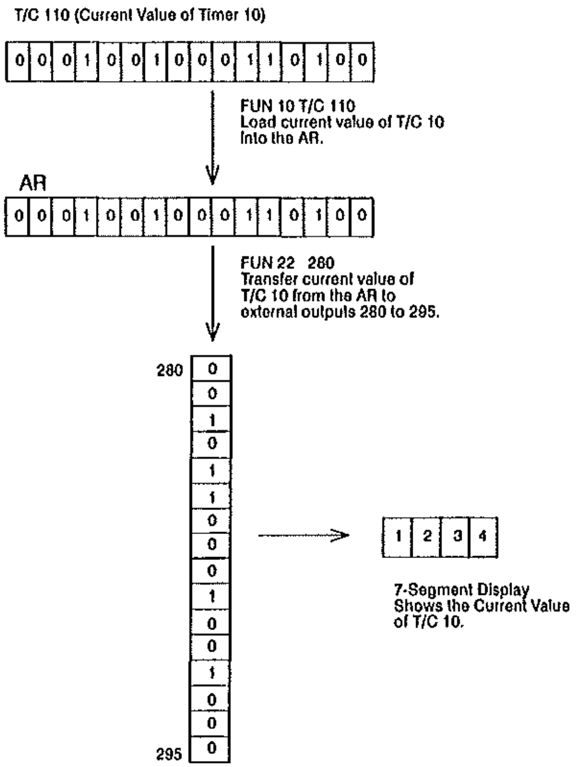


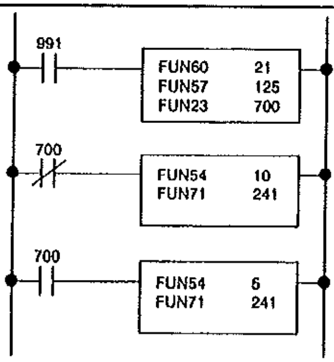
Figure 6.82 - Displaying a Timer's Current Value on an External 7-Segment Display

6.5.22 Byte Out Command

- FUN71 = Byte Out Command = BOUT
- This command transfers data from the lower byte of the arithmetic register (AR_L) to the specified external or internal byte. This command can also transfer data from the AR_L to the lower byte of a timer/counter's current or preset value. See figures 6.83 and 6.84.
- The carry bit does not change.
- Note that when data is transferred to a timer/counter's preset value, the programmed preset value will be restored if a system restart occurs.
- This command can be used to:
  - change the value of an analog output (resolution $\leq$ 8 bits)
  - set the lower byte of a preset register in the High Speed Counter module (M/N 45C982).
  - store values following a math operation.

Data to be Transferred	Example	Result
External output byte or internal output byte	FUN71 241	Contents of AR_L are transferred to 8-bit analog output channel 0, slot 3.
Lower byte of timer or counter current or preset values	FUN71 T/C 210	Contents of AR_L are transferred to the lower byte of the T/C10 preset.

Figure 6.83 - Example of FUN71



Coding		Remarks
ORG	991	8-bit analog input → AR_L.
FUN60	21	If $AR_L \geq$ 125, C = 1
FUN57	125	else C = 0.
FUN23	700	When C = 1, 700 is on. When C = 0, 700 is off.
ORG NOT	700	
FUN54	10	When AR_L < 125, divide AR_L by binary constant 10 and store the binary quotient in AR_L.
FUN71	241	Output binary quotient from AR_L to 8-bit analog output channel 0, slot 3.
ORG	700	When AR $\geq$ 125, divide
FUN54	5	AR_L by binary constant 5 and store the binary quotient in AR_L.
FUN71	241	Output binary quotient from AR_L to 8-bit analog output channel 0, slot 3.

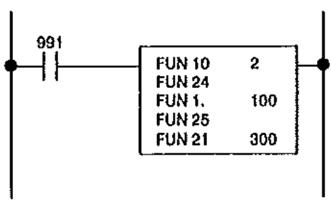
Figure 6.84 - Moving Data through a Byte Out Command to an 8-Bit Analog Output Module

6.5.23 Carry Bit Commands

- FUN23 = Output Carry Bit Command = OUC
- FUN83 = Clear Carry Bit Command = CLC
- FUN84 = Set Carry Bit Command = SEC
- FUN23 outputs the status (one or zero) of the carry bit to either an external (200 to 395) or internal (400 to 955) output.
- FUN83 sets the carry bit to zero.
- FUN84 sets the carry bit to one.
- The carry bit is automatically reset to zero at the beginning of each scan.

6.5.24 Binary Coded Decimal and Binary Commands

- FUN24 = Binary Coded Decimal = BIN-BCD
- FUN25 = Binary = BCD-BIN
- FUN24 converts the data in the arithmetic register from binary data to BCD data. Refer to figures 6.85 and 6.86.
- FUN25 converts the data in the arithmetic register from BCD data to binary data.



Coding		Remarks
ORG	991	Internal contact is always on when in the run mode.
FUN 10	2	Load analog input data (slot 1, channel 1) into AR.
FUN 24		Convert the contents of the AR (analog input data) from binary to BCD.
FUN 1.	100	Add BCD constant "100" to the value in the AR. Store the result in the AR.
FUN 25		Convert the BCD value in the AR to a binary value.
FUN 21	300	Transfer the binary value from the AR to the analog output (slot 6, channel 0).

Figure 6.85 - Example of FUN24 and FUN25 with Analog Inputs and Outputs

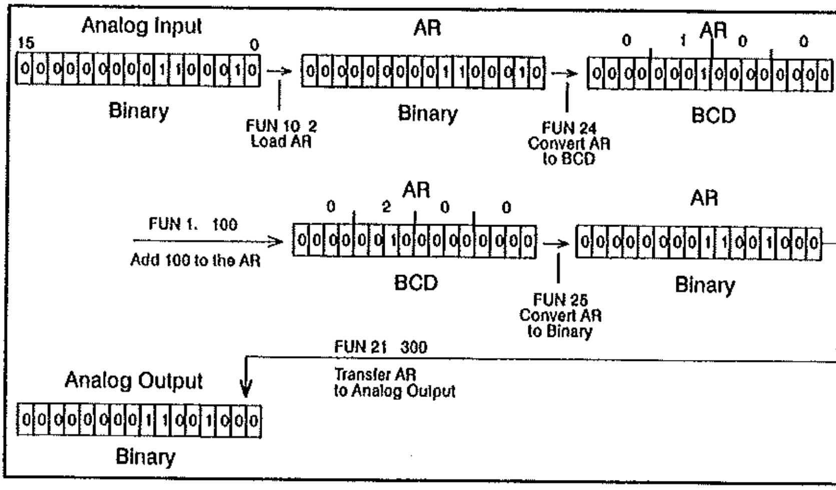


Figure 6.86 - Converting Analog Input and Output Data Using FUN24 and FUN25

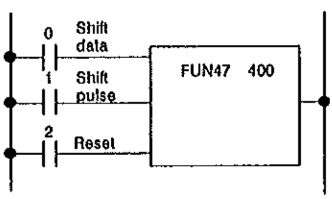
- If BCD data that is converted from binary data is larger than 9999, the carry bit is equal to one and the previous value remains unchanged. See figure 6.87.

	AR or Input Data		Results	
	BIN	BCD	0 - 9999	>9999
FUN24 =	C = 0	---	C = 0	C = 1
FUN25 =	C = 1	C = 0	---	---

Figure 6.87 - Carry Bit Status of FUN24 and FUN25

6.5.25 Shift Register Command

- FUN47 = Shift Register Command = SFR
- A shift register is a group of storage locations that are synchronized by a shift pulse signal.
- You use internal coils when programming the shift register command. You need to specify the command's shift data, shift pulse, and reset input in that order. Refer to figure 6.88.
- The sixteen bits starting from the internal output address form the shift register (bits 400 to 415 in figure 6.88).
- The status (on/off) of the shift data is stored in the first bit position of the shift register when the shift pulse changes from off to on (internal output 400 in figure 6.88). As data is shifted into the register, each bit is shifted to the next higher position.
- When the reset input is energized, the shift register is reset to all zeros.
- For a retentive shift register, you must use internal outputs 700 to 940.
- The shift register's most significant bit is the overflow bit. Using the overflow bit, you can program cascaded shift functions as shown in figure 6.89. When programming cascaded shift registers, you must program the upper sixteen bits first and the lower sixteen bits second.
- This command is useful when tracking the status of a component through a series of positions or stations.



Coding		Remarks
ORG	0	Use the Str instruction when programming FUN 47.
STR	1	
STR	2	
FUN47	400	

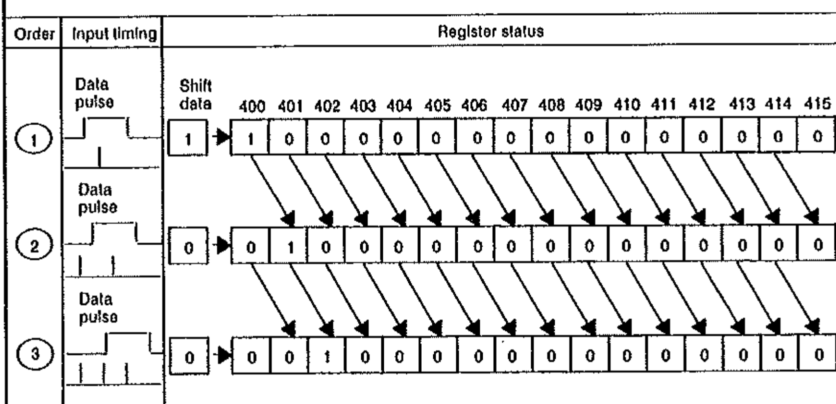
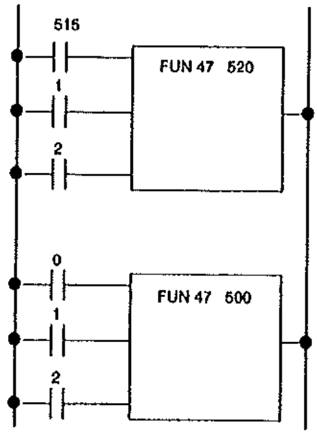


Figure 6.88 - Example of FUN47

Shift Register Command (Continued)



Coding		Remarks
ORG	515	Program the upper 16 bits (520-535) first.
STR	1	
STR	2	Bit 515* is the overflow bit from the lower 16-bit shift register.
FUN47	520	
ORG	0	Program the lower 16 bits (500-515) second.
STR	1	
STR	2	
FUN47	500	

* In order to capture the high bit (515), FUN 47 520 must be programmed before FUN 47 500 in memory.

Figure 6.89 - Cascaded Shift Register Example

6.5.26 Left Shift Register and Right Shift Register Commands

- FUN26 = Left Shift Register Command = SFR L
- FUN27 = Right Shift Register Command = SFR R
- FUN26 shifts the contents of the arithmetic register one bit to the left. The least significant bit (b0) is loaded with a zero and the overflow bit sets the carry bit to one. Refer to figure 6.90.
- FUN27 shifts the contents of the arithmetic register one bit to the right. The most significant bit (b15) is loaded with a zero and the overflow bit sets the carry bit to one. Refer to figure 6.91.

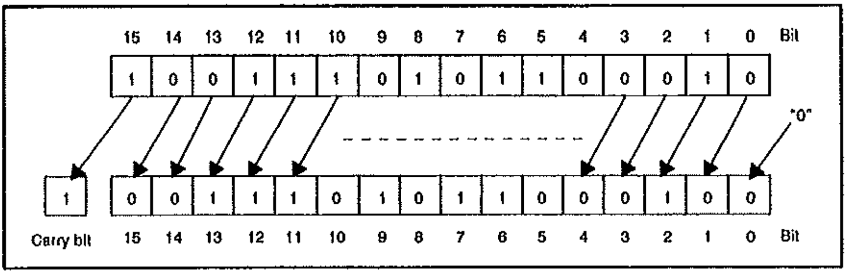


Figure 6.90 - Example of FUN26

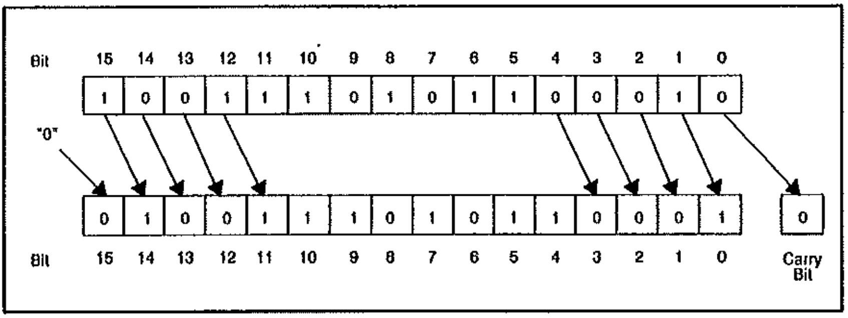


Figure 6.91 - Example of FUN27

6.5.27 Rotate Shift Register Commands

- FUN76 = Rotate Shift Register Left Command = ROL
- FUN77 = Rotate Shift Register Right Command = ROR
- FUN76 shifts the contents of the arithmetic register one bit to the left. As the shift occurs, the status of the carry bit is shifted into the least significant bit (LSB) of the arithmetic register and the status of the most significant bit (MSB) is shifted to the carry bit. See figure 6.92.
- FUN77 shifts the contents of the arithmetic register one bit to the right. As the shift occurs, the status of the carry bit is shifted into the most significant bit (MSB) of the arithmetic register and the status of the least significant bit (LSB) is shifted to the carry bit. See figure 6.93.

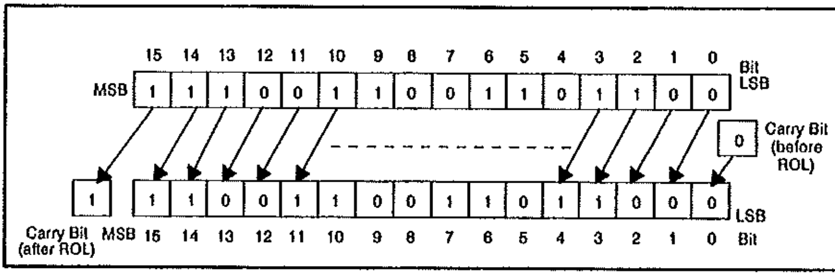


Figure 6.92 - Example of FUN76

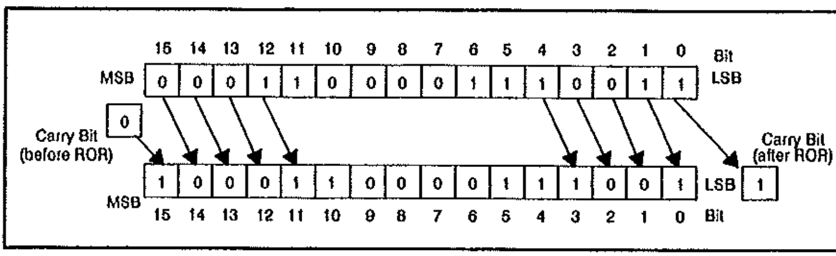
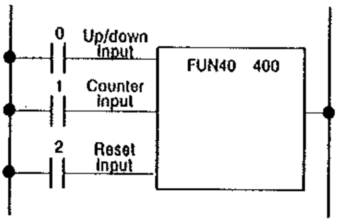


Figure 6.93 - Example of FUN77

6.5.28 Up/Down Counter Command

- FUN40 = Up/down counter = UDC
- To program a counter sequence, you must enter the up/down input (on = count up, off = count down), the counter input, and the reset input, in that order. See figure 6.94.
- This command is programmed in combination with an external output. The specified internal output is the first of sixteen bits that comprise the current value register of the up/down counter.
- To make the current value register retentive upon power loss, you must specify a retentive internal output (700 to 940) with the FUN40 command.
- In the count-up mode, the counter counts the leading edge of the counter input and increments the count register. The count register begins at 0000 and counts up to 9999. When the count register reaches 9999, it rolls over to 0000.
- In the count-down mode, the counter counts the leading edge of the counter input and decrements the count register. The count register begins at 0000 and counts down from 9999 to 0001. When the count register reaches its maximum or minimum value (0001 or 9999), it rolls over to 0000.
- When the reset input is energized, the count register resets to 0000.



Coding		Remarks
ORG	0	Use the Str instruction when programming FUN40.
STR	1	
STR	2	
FUN40	400	

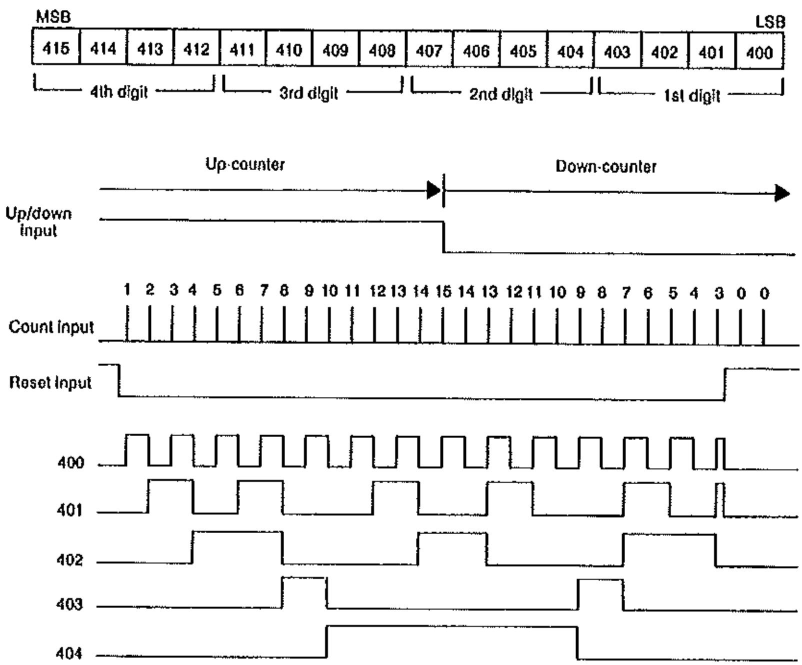


Figure 6.94 - Example of FUN40

6.5.29 Invert Arithmetic Register Command

- FUN85 = Invert arithmetic register command = WNOT
- This command performs a logical NOT operation on the contents of the arithmetic register and stores the result in the arithmetic register. See figure 6.95.
- The carry bit is not affected by this command.

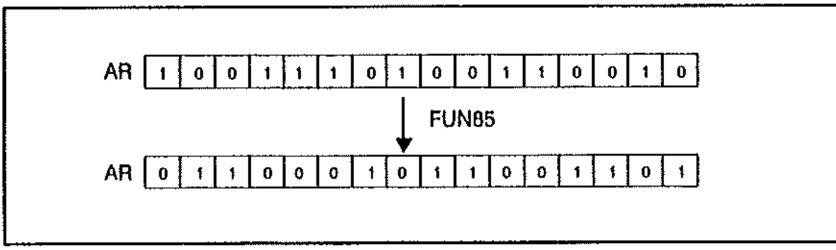
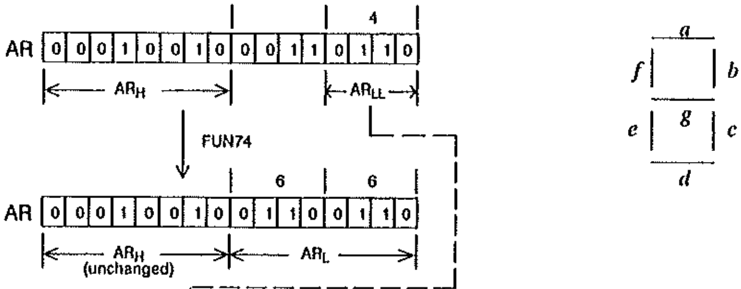


Figure 6.95 - Example of FUN85

6.5.30 7-Segment Conversion Command

- FUN74 = 7-Segment Conversion Command = SEG
- FUN74 converts the lower 4 bits (AR_{LL}) of the arithmetic register into a 7-segment display code. The 7-segment display code is then stored in the lower 8-bits (AR_L) of the arithmetic register. See figure 6.96.
- The upper 8 bits (AR_H) of the arithmetic register remain unchanged.

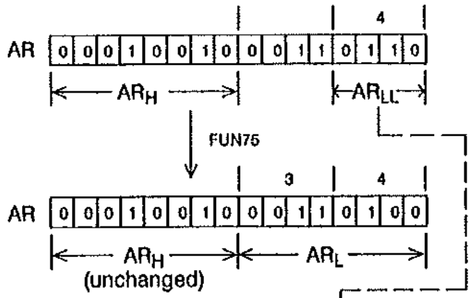


Input Data 4 Bits (AR_{LL})	FUN74								
	Output Data								Display
		g	f	e	d	c	b	a	
0	0	0	1	1	1	1	1	1	0
1	0	0	0	0	0	1	1	0	1
2	0	1	0	1	1	0	1	1	2
3	0	1	0	0	1	1	1	1	3
4	0	1	1	0	0	1	1	0	4
5	0	1	1	0	1	1	0	1	5
6	0	1	1	1	1	1	0	1	6
7	0	0	1	0	0	1	1	1	7
8	0	1	1	1	1	1	1	1	8
9	0	1	1	0	1	1	1	1	9
A	0	1	1	1	0	1	1	1	A
B	0	1	1	1	1	1	0	0	b
C	0	0	1	1	1	0	0	1	C
D	0	0	1	0	1	1	1	0	d
E	0	1	1	1	1	0	0	1	E
F	0	1	1	1	0	0	0	1	F

Figure 6.96 - Example of FUN74

6.5.31 ASCII Conversion Command

- FUN75 = ASCII Conversion Command = ASC
- FUN75 converts the lower 4 bits (AR_{LL}) of the arithmetic register into ASCII code. The ASCII code is then stored in the lower 8-bits (AR_L) of the arithmetic register. See figure 6.97.
- The upper 8 bits (AR_H) of the arithmetic register remain unchanged.



Input Data 4 Bits (AR_{LL})	FUN75 Output Data
0	30
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39
A	40
B	41
C	42
D	43
E	44
F	45

Figure 6.97 - Example of FUN75

6.5.32 Arithmetic Register Encode and Decode Commands

- FUN78 = Arithmetic register encode command = ENCOD
- FUN79 = Arithmetic register decode command = DECOD
- FUN78 encodes the upper-most bit position (0 to 15) that is set to one. The number of the bit position is stored in the lower 4 bits (AR_{LL}) of the arithmetic register. The remainder of the bits in the arithmetic register are set to zero. See figure 6.98.
- If all bits in the arithmetic register are equal to zero, the arithmetic register contents remain unchanged and the carry bit is set to one.
- FUN79 sets to one the bit position corresponding to the value in the arithmetic register (bits 0 to 15). All other bits are set to zero. See figure 6.99.
- If the value in the arithmetic register is greater than 15, all bits in the arithmetic register are set to zero and the carry bit is set to one.

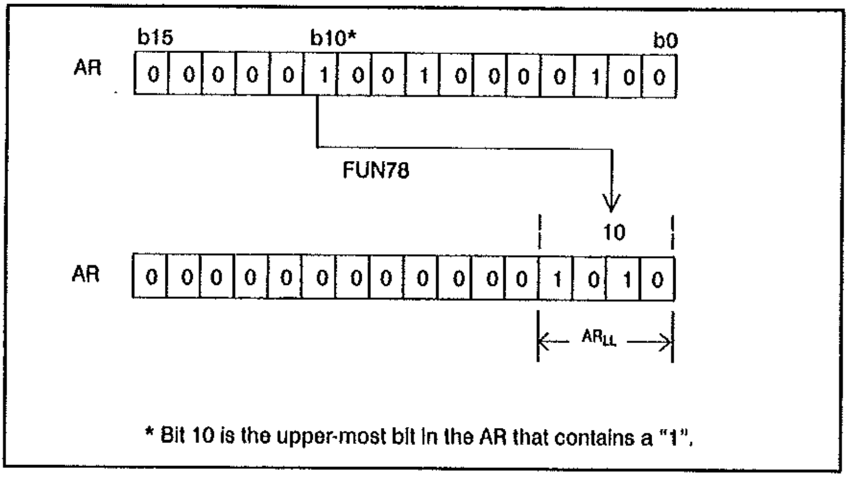


Figure 6.98 - Example of FUN78

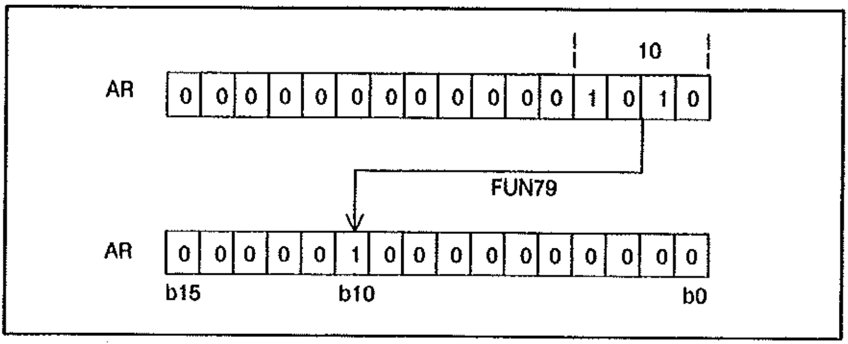
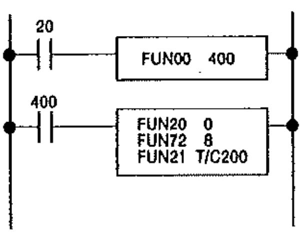


Figure 6.99 - Example of FUN79

6.5.33 Arithmetic Register Mask Commands

- FUN72 = Mask the arithmetic register from the most significant bit (MSB) = MASK L
- FUN73 = Mask the arithmetic register from the least significant bit (LSB) = MASK R
- FUN72 and FUN73 mask the arithmetic register's data by the specified number of bits (1 to 16).
- FUN72 begins the mask from the MSB of the arithmetic register.
- FUN73 begins the mask from the LSB of the arithmetic register.
- The example in figure 6.100 shows a 16-point input module with a 2-digit thumbwheel switch wired to inputs 0 through 7 and normally-open switches wired to inputs 8 through 15. When using FUN20 to input the thumbwheel switch setting, all 16 bits of information from the input module are loaded into the arithmetic register. FUN72 masks the status of the normally-open switches (inputs 8 to 15), which leaves the status of the thumbwheel switches (inputs 0 to 7) in the arithmetic register.



Coding		Remarks
ORG FUN00	20 400	"One-shot" operation to load the new Timer 0 preset value.
ORG FUN20	400 0	Load external inputs 0-15 into the AR.
FUN72	8	Mask the AR 8-bits starting from the MSB.
FUN21	T/C200	Load the new Timer 0 preset value from the AR.

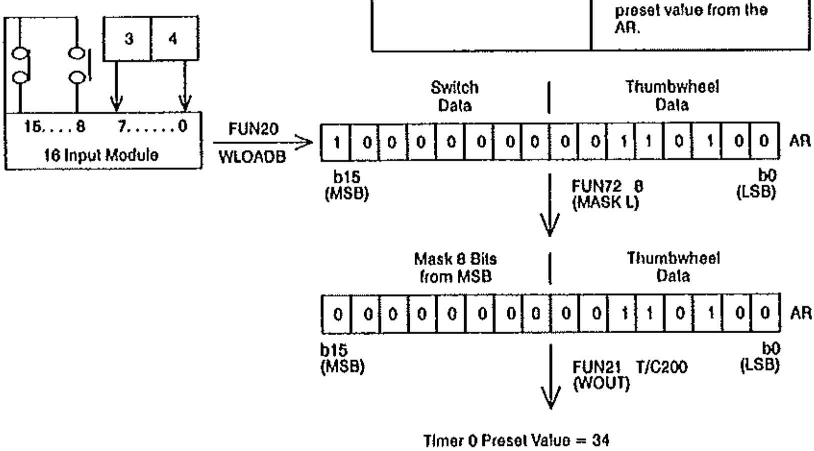


Figure 6.100 - Example of FUN72 and FUN73

6.5.34 Arithmetic Register Exchange Commands

- FUN80 = Exchange arithmetic register's upper byte (AR_H) and lower byte (AR_L) = SWAP
- FUN81 = Exchange the nibbles of the arithmetic register's lower byte (AR_L) = BSWAP
- FUN82 = Exchange arithmetic register data and extension register data = XCG
- FUN80 exchanges the arithmetic register's upper byte (bits 8 to 15) with the lower byte (bits 0 to 7). See figure 6.101.
- FUN81 exchanges the upper nibble (bits 4 to 7) with the lower nibble (bits 0 to 3) of the arithmetic register's lower byte. See figure 6.102.
- FUN82 exchanges arithmetic register data with extension register data. This command can be used to set the extension register or to access the value in the extension register through the arithmetic register. See figure 6.103.

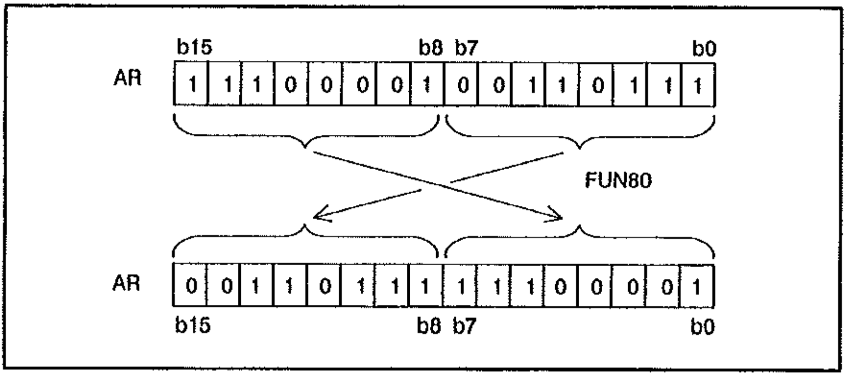


Figure 6.101 - Example of FUN80

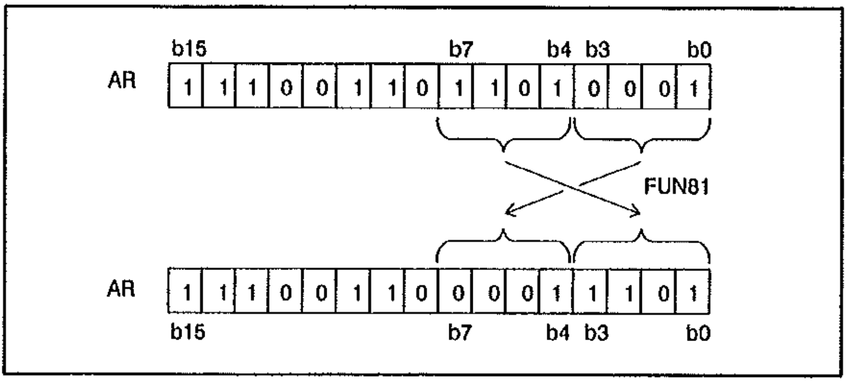


Figure 6.102 - Example of FUN81

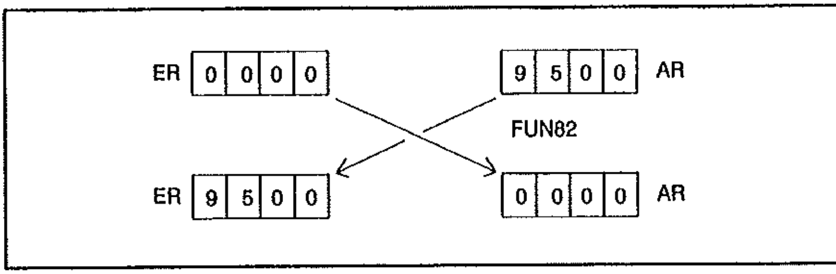
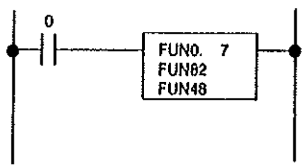


Figure 6.103 - Example of FUN82

6.5.35 Extract and Distribute Commands

- FUN48 = Load data into the arithmetic register from the I/O address specified by the extension register = EXTRACT
- FUN49 = Output data from the arithmetic register to the I/O address specified by the extension register = DISTRIB
- FUN48 loads word data into the arithmetic register from the I/O address specified by the extension register. See figure 6.104.
- FUN49 outputs word data from the arithmetic register to the I/O address specified by the extension register. See figure 6.105.
- The extension register contains BCD data. The most significant digit in the extension register determines the type of address specified when using FUN48 and FUN49. When the most significant digit is equal to zero, the specified address is an I/O address. When the most significant digit is equal to one, the specified address is a timer/counter address.
- If the FUN48 or FUN49 is executed with an undefined I/O specification in the extension register, the carry bit will be set to one.



Coding		Remarks
ORG	0	
FUN0.	700	Load constant 700 into the AR.
FUN82		Exchange AR and ER data (ER now = 700)
FUN48		Load word data from I/O addresses 700 and 701.

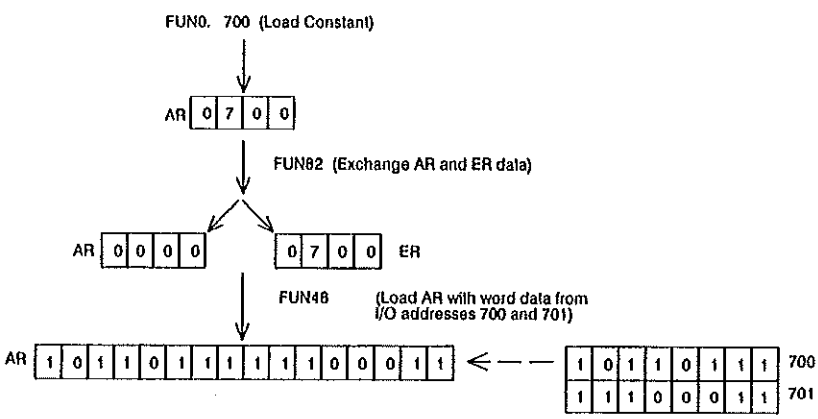
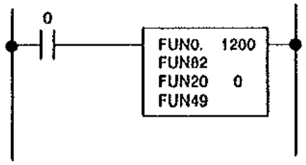


Figure 6.104 - Example of FUN48



Coding		Remarks
ORG	0	
FUN0.	1200	Load constant 1200 into AR.
FUN82		Exchange AR and ER data (ER now = 1200)
FUN20	0	Load external inputs 0-15 into AR
FUN49		Output AR word data to timer 0 preset (address specified in ER)

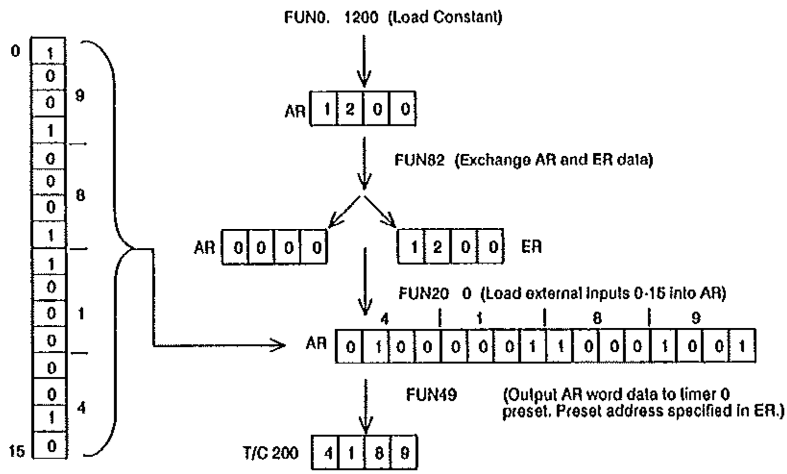


Figure 6.105 - Example of FUN49

6.6 Analog Data

- The Shark XL Analog Input modules provide the interface for 0 to 10VDC input signals (M/N 45C992) and 4 to 20mA input signals (M/N 45C990).
- These input modules convert the analog signals into digital binary data with eight bits of resolution. FUN60 (BLOAD) is used to load the converted binary data into the arithmetic register. To perform binary operations on this data, use the appropriate binary instructions. To perform BCD operations on this data, you must first use FUN24 to convert the data from binary into BCD format. See figure 6.106.
- The Shark XL Analog Output modules provide the interface for 0 to 10VDC output signals (M/N 45C994 or 45C997) or 4 to 20mA output signals (M/N 45C993 or 45C995).
- These output modules convert digital binary data with eight bits of resolution into analog signals. The data being sent to the analog output address must be in binary format. If the data is in BCD format, use FUN25 to convert the data to binary format prior to the transfer of data to the analog output module. Use FUN71 (BOUT) to transfer eight bit digital binary data from the arithmetic register to the analog output channel.
- These analog modules have eight bits of resolution. This provides for a digital range of 0 to 255 when using analog signals with these devices.

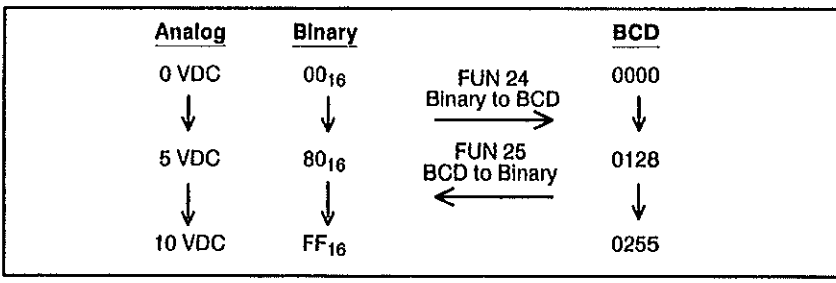


Figure 6.106 - Analog Signal Conversion for BCD Operations

Analog Data (Continued)

- Figures 6.107, 6.108, and 6.109 show typical analog examples. The analog input module is in slot one, while the analog output module is in slot two.

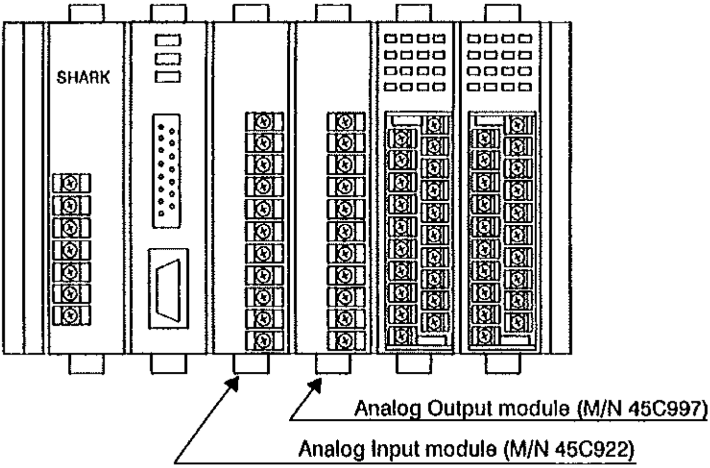
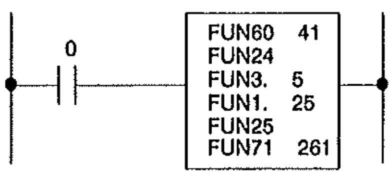


Figure 6.107 - Analog I/O Example

The equation to be programmed is: $Y = X * A + B$

where:

- X = 8-bit analog input from channel 0 of the module in slot 3
- Y = 8-bit analog output to channel 0 of the module in slot 4
- A = 5 BCD (constant value)
- B = 25 BCD (constant value)



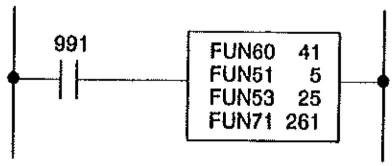
ORG 0		
FUN60 41:	Analog input from the lower byte of channel 0 (8-bit BNR)	→ AR (8-bit BNR)
FUN24:	BNR to BCD conversion of AR data	→ AR (16-bit BCD)
FUN3. 5:	AR x 5 (constant)	→ AR (16-bit BCD)
FUN1. 25:	AR + 25 (constant)	→ AR (16-bit BCD)
FUN25:	BCD to BNR conversion of AR data	→ AR (8-bit BNR)
FUN71 261:	AR (8-bit BNR)	→ Analog output to the lower byte of ch 0 (slot 4)

Figure 6.108 - Sample Analog I/O Programming with BCD Operations

The equation to be programmed is: Y = X * C + D

where:

X = 8-bit analog input from channel 0 of the module in slot 3
Y = 8-bit analog output to channel 0 of the module in slot 4
A = 5 (binary constant value)
B = 25 (binary constant value)



ORG 991:	Internal contact is always on in the run mode		
FUN60 41:	Analog input from the lower byte of channel 0 (8-bit binary)	→	AR (8-bit BNR)
FUN51 5:	AR x 5 (binary constant)	→	AR (8-bit BNR)
FUN53 25:	AR + 25 (binary constant)	→	AR (8-bit BNR)
FUN71 261:	AR (8-bit binary)	→	Analog output to the lower byte of ch 0 (slot 4)

Figure 6.109 - Sample Analog I/O Programming for Binary Operations

6.7 Counter Module I/O

The Counter module (M/N 45C982) is an input/output module for use with pulsetach encoders.

This module uses status registers and control registers. See figures 6.110 and 6.111. Counter module addressing is the same as for analog modules. Refer to Table 4.2 for Counter module slot starting addresses.

In figure 6.112, the status registers are assigned input addresses 20 to 35 because the Counter module is in slot 2. The control registers are assigned output addresses 220 and 235.

External Input Addresses	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Overflow Flag	–	–	–	–	–	–	Input A Status
1	Underflow Flag	–	–	–	–	–	–	Input B Status
2	Current Counter BCD Value (UDC) High Byte							
3	Current Counter BCD Value (UDC) High Byte							
4	MCE	–	–	–	–	–	–	–
5	CPE	–	–	–	–	–	–	–
6	Input m Status	–	–	–	–	–	–	–
7	1-Phase/ 2-Phase	–	–	–	–	–	–	–
8	Counter 0 = Flag	–	–	–	–	–	–	–
9	Counter 0 > Flag	–	–	–	–	–	–	–
10	Counter 1 = Flag	–	–	–	–	–	–	–
11	Counter 1 > Flag	–	–	–	–	–	–	–
12	Counter 2 = Flag	–	–	–	–	–	–	–
13	Counter 2 > Flag	–	–	–	–	–	–	–
14	Counter 3 = Flag	–	–	–	–	–	–	–
15	Counter 3 > Flag	–	–	–	–	–	–	–

Status Registers

Overflow Flag: When the current counter value goes from 9999 to 0, this flag turns on.

Underflow Flag: When the current counter value goes from 0 to 9999, this flag turns on.

Current Counter Value: BCD value ranging from 0-9999.

Input A, B, m Status: Indicates the current status of the inputs coming from the pulse encoder.

1-Phase/2-Phase: Indicates whether the pulse encoder is wired for one or two-phase operation.

Figure 6.110 - Counter Module Status Registers

Counters 0-4 = Flags: When the current counter value equals their respective present values, these flags will turn on.

Counters 0-4 > Flags: When the current controller value is greater than their respective present values, these flags turn on.

MCE: Marker Enable Complete = "1" when Marker Enable (ME) = "1"

CPE: Preset Data to Counter Current Value is complete when CPE = "1"

Figure 6.110 - Counter Module Status Registers (Continued)

External Output Addresses	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Overflow Clear	Clear "3" = Flag	Clear "2" = Flag	Clear "1" = Flag	Clear "0" = Flag	All Clear	Counter Enable	Marker Enable
1	Underflow Clear							Counter Preset
2	Counter Preset Value High Byte							
3	Counter Preset Value Low Byte							
4	Output 0 Control	–	–	–	Output 3 Enable	Output 2 Enable	Output 1 Enable	Output 0 Enable
5	Output 1 Control	–	–	–	Output 3 Data	Output 2 Data	Output 1 Data	Output 0 Data
6	Output 2 Control	–	–	–	–	–	–	–
7	Output 3 Control	–	–	–	–	–	–	–
8	Counter 0 Preset Value High Byte							
9	Counter 0 Preset Value Low Byte							
10	Counter 1 Preset Value High Byte							
11	Counter 1 Preset Value Low Byte							
12	Counter 2 Preset Value High Byte							
13	Counter 2 Preset Value Low Byte							
14	Counter 3 Preset Value High Byte							
15	Counter 3 Preset Value Low Byte							

Overflow Clear Flag: 0 = Do not clear the flag
1 = Clear the flag

Underflow Clear Flag: 0 = Do not clear the flag
1 = Clear the flag

Clear "0" to Clear "3" = Flag: 0 = Do not clear the "=" flag for the output (0-3)
1 = Clear the "=" flag for the output (0-3)

All Clear: 0 = no function
1 = resets counter current value to "0000" and all flags to "0"

Counter Enable: 0 = CPU read/write of counter disabled
1 = CPU read/write of counter enabled

Marker Enable: 0 = Marker terminal input invalid
1 = Marker terminal input valid

Counter Preset: 0 = no function
1 = The counter preset value is written to the counter current value register

Figure 6.111 - Counter Module Control Registers

Note 1: Set the following when initializing the counter module:
Counter Enable = 1 (continuous)
Marker Enable = 1 (continuous)
All Clear = 1 (pulse)

Note 2: Counter Preset control bit should be set to "1" for one scan when it is necessary to restore counter current value

Counter Preset Value: Use these registers to reset counter current value.

Output 0-3 Control: 0 = send the contents of the output ">" flag to the counter module external output (0-3)

1 = send the contents of the output "=" flag to the counter module external output (0-3)

Output 0-3 Enable: 0 = Output 0-3 Control Bits control data to counter module external outputs (0-3)
1 = enables forcing of counter module external outputs (0-3)

Output 0-3 Data: When output 0-3 Enable = 1, the data "0" or "1" is sent to the external output

When output 0-3 Enable = 0, the Output Control bits determine the data sent to the external outputs.

Counter 0-3 Preset Value: Registers to define up to 4 preset values. Counter Module will compare current value to each of these preset values and control the respective counter ">" and "=" flags.

Figure 6.111 - Counter Module Control Registers (Continued)

Counter Module I/O (Continued)

		1	2	3	4
P.S.	CPU	I/O	45C982	I/O	I/O

External Input Addresses	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
20	Overflow Flag	–	–	–	–	–	–	Input A Status
21	Underflow Flag	–	–	–	–	–	–	Input B Status
22	Current Counter BCD Value (UDC) High Byte							
23	Current Counter BCD Value (UDC) High Byte							
24	MCE	–	–	–	–	–	–	–
25	CPE	–	–	–	–	–	–	–
26	Input m Status	–	–	–	–	–	–	–
27	1-Phase/ 2-Phase	–	–	–	–	–	–	–
28	Counter 0 = Flag	–	–	–	–	–	–	–
29	Counter 0 > Flag	–	–	–	–	–	–	–
30	Counter 1 = Flag	–	–	–	–	–	–	–
31	Counter 1 > Flag	–	–	–	–	–	–	–
32	Counter 2 = Flag	–	–	–	–	–	–	–
33	Counter 2 > Flag	–	–	–	–	–	–	–
34	Counter 3 = Flag	–	–	–	–	–	–	–
35	Counter 3 > Flag	–	–	–	–	–	–	–

Status Registers

Figure 6.112 – Counter Module I/O Addressing

Counter Module I/O (Continued)

External Output Addresses	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
220	Overflow Clear	Clear "3" = Flag	Clear "2" = Flag	Clear "1" = Flag	Clear "0" = Flag	All Clear	Counter Enable	Marker Enable
221	Underflow Clear							Counter Preset
222	Counter Preset Value High Byte							
223	Counter Preset Value Low Byte							
224	Output 0 Control	–	–	–	Output 3 Enable	Output 2 Enable	Output 1 Enable	Output 0 Enable
225	Output 1 Control	–	–	–	Output 3 Data	Output 2 Data	Output 1 Data	Output 0 Data
226	Output 2 Control	–	–	–	–	–	–	–
227	Output 3 Control	–	–	–	–	–	–	–
228	Counter 0 Preset Value High Byte							
229	Counter 0 Preset Value Low Byte							
230	Counter 1 Preset Value High Byte							
231	Counter 1 Preset Value Low Byte							
232	Counter 2 Preset Value High Byte							
233	Counter 2 Preset Value Low Byte							
234	Counter 3 Preset Value High Byte							
235	Counter 3 Preset Value Low Byte							

Control Registers

Figure 6.112 - Counter Module I/O Addressing (Continued)

7.0 DIAGNOSTICS AND TROUBLESHOOTING

This chapter explains how to check the program's syntax. This chapter also explains how to locate a problem with the programmable controller by observing the modules' LEDs and the error messages on the handheld programmer. If the problem cannot be found by using the troubleshooting instructions below, the hardware is not user-serviceable.

> **DANGER**
> **ONLY QUALIFIED ELECTRICAL PERSONNEL FAMILIAR WITH THE CONSTRUCTION AND OPERATION OF THIS EQUIPMENT AND THE HAZARDS INVOLVED SHOULD INSTALL, ADJUST, OPERATE, OR SERVICE THIS EQUIPMENT. READ AND UNDERSTAND THIS MANUAL IN ITS ENTIRETY BEFORE PROCEEDING. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.**

7.1 Checking Program Syntax

Use the procedure in figure 7.1 to check the program's syntax. Check the program through FUN99 (end of program). If no errors are found, the first available (unused) memory location will be displayed.

Operation	Mode Switch Setting	Processor Status
Checking Program Syntax	PROG TEST RUN	Stop

Key-In Procedure		Type of Display			Remarks
		Command	Numerical Display	Mode Display	
CLR SRC	There is no error in the program		*300*	• TEST	Displays the 1st available (unprogrammed) memory location.
	There is an error		*115 E*	• PROG • TEST • RUN	Indicates that an error has been found in memory location 115.

Figure 7.1 - Checking Program Syntax

Table 7.1 shows the syntax error codes that appear on the handheld programmers. The step number displayed is the one where the error was detected. When a syntax error is detected, monitor address 980 in decimal for the proper error code. Note that these error codes will not be cleared by cycling system power. They are cleared when the syntax error has been corrected.

Table 7.1 - Syntax Error Codes

Syntax Error Code in Decimal	Standard Programmer Error Display	Universal Programmer Error Display	Cause of Error	Corrective Action
0	Blank	Blank	No error	Normal operation
1	E	E	General syntax programming error.	Locate and correct the error.
2	E	E	Structure of the main or interrupt processing routine is incorrect.	Program the interrupt commands correctly.
3	E	E	INT command is programmed incorrectly.	Program the command correctly.
4	E	E	JMP and JMP END commands are programmed incorrectly.	Program the commands correctly.
5	E	E	AJMP and AJMP END commands are programmed incorrectly.	Program the commands correctly.
6	└─┘	UE	STR command is programmed incorrectly.	Program the command correctly.
7	┌─┐	oE	STR command is programmed incorrectly.	Program the command correctly.
8	└─┘	oE	Master control command is programmed incorrectly.	Program the command correctly.
9	┌─┐	oE	Master control command is programmed incorrectly.	Program the command correctly.
10	E	E	IF or IFR command is duplicated.	Locate and correct the error.
11	E	E	I/O number or constant value is out of range.	Locate and correct the error.

Table 7.1 - Syntax Error Codes (Continued)

Syntax Error Code in Decimal	Standard Programmer Error Display	Universal Programmer Error Display	Cause of Error	Corrective Action
12	E	E	Two outputs are programmed in one rung.	Remove one output.
13	E	dE	Occurrence of dual coil operation is allowed.	Notification only. Corrective action is not required.
14	E	E	CALL and SB commands do not match.	Verify that the CALL# and SB# correspond.
15	E	E	JMP and INT commands are programmed in the same rung.	Correct the error.
20	F	fE	Undefined operation code.	Clear memory and reprogram the unit.
30	E	E	Checksum error.	Clear memory and reprogram the unit.

7.2 LED Status and Troubleshooting

Table 7.2 explains the LED indicators which can help you locate a malfunction.

WARNING
TURN OFF POWER TO THE CONTROLLER AND FIELD WIRING BEFORE SERVICING THE EQUIPMENT. INADVERTENT MACHINE OPERATION MAY RESULT IF THE POWER IS NOT TURNED OFF. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

Table 7.2 - LED Troubleshooting

LED Indicator	Item to be Tested	Cause of Problem	Corrective Action
Processor module's POWER LED is Off	Check incoming power	Low A-C voltage, a malfunctioning power supply or Processor module, or a defective rack	Correct the input power problem. Replace the defective rack or module.
Processor module's RUN LED is Off	Programmer's Mode Selector Switch	Switch is in the wrong position	Place switch in the RUN position
	Programming Error	FUN98 start circuit is not energized, is missing, or is programmed incorrectly	Check the FUN98 start circuit
	Cycle power and reload program. Test syntax with the start input disabled.	Watchdog error due to electrical noise	Add a noise filter or suppressor to the A-C input line
		Checksum error	Check the program for an error
		Processor module failure	Replace the Processor module
Input module's LED is On	Voltage is present on the input's terminals	Monitor the input. The input address in memory equals one.	**Normal operation**
		Monitor the input. The input address in memory equals zero.	Replace Input module
	Voltage is not present on the input's terminals	Check external input wires. Input module failure.	Correct the input wiring problem and replace the Input module

Table 7.2 - LED Troubleshooting (Continued)

LED Indicator	Item to be Tested	Cause of Problem	Corrective Action
Input module's LED is Off	Voltage is not present on the input's terminals	Monitor the input. The input address in memory equals zero.	**Normal operation**
		Monitor the input. The input address in memory equals one.	Replace Processor module or the Input module
	Voltage is present on the input's terminals	Input module failure	Replace the Input module
Output module's LED is Off	Processor instructs the output to be off	Check the voltage on the module's output terminals. Voltage is present.	Replace the Output module
		Check voltage on the module's output terminals. Voltage is not present.	**Normal operation**
	Processor instructs the output to be on	Check the voltage on the module's output terminals. Voltage is present.	Replace the Output module
		Check voltage on the module's output terminals. Voltage is not present.	Check and replace Output module's fuses. Replace the Output module.
Output module's LED is On	Processor instructs the output to be on	Check the voltage on the module's output terminals. Voltage is present.	**Normal operation**
		Check voltage on the module's output terminals. Voltage is not present.	Check and replace Output module's fuses. Replace the Output module.
	Processor instructs the output to be off	Check the voltage on the module's output terminals. Voltage is present.	Replace the Output module.
		Check voltage on the module's output terminals. Voltage is not present.	Replace the Output module

7.3 Error Codes

The handheld programmer will display an error code when a programming error is detected. Table 7.3 lists the error codes.

Table 7.3 - Error Codes

Error Code:	4 E
Type of Error:	Undefined Command Error
When Detected:	When the processor is running
Description:	An undefined operation command has been detected. Internal output 972 is on.
Corrective Action:	Cycle power. If this does not clear the error, replace the processor. If this error occurs frequently, electrical noise may be affecting the system. To correct this, connect a noise filter to the incoming power line and ground the GND terminal.
Error Code:	
Type of Error:	Watchdog Timer Error
When Detected:	When the processor is running.
Description:	The system's scan time has been exceeded and a watchdog timer error has been detected. The RUN LED on the Processor module is off.
Corrective Action:	Cycle power. If this does not clear the error, replace the processor. If this error occurs frequently, electrical noise may be affecting the system. To correct this, connect a noise filter to the incoming power line and ground the GND terminal.
Error Code:	51 E
Error Type:	Checksum Error
When Detected:	When the processor is first put into run.
Description:	A checksum error has been detected during the user memory test.
Corrective Action:	Check the program for syntax or other programming errors and correct them.
Error Code:	A 62 E
Error Type:	Memory Error
When Detected:	When loading a program into the processor from the cassette recorder.
Description:	Data cannot be written (loaded) into the processor's EPROM/EEPROM memory from the cassette recorder.
Corrective Action:	Cycle power and try to reload the data. If the processor's memory still will not accept the data, replace the EPROM/EEPROM memory.

Table 7.3 - Error Codes (Continued)

Error Code:	C 62 E
Error Type:	EPROM/EEPROM Write Error
When Detected:	During a cassette verification.
Description:	Data cannot be written (loaded) into the processor's EPROM/EEPROM memory from the cassette recorder.
Corrective Action:	Cycle power and try to re-load the data. If the processor's memory still will not accept the data, replace the EPROM/EEPROM memory.
Error Code:	C 7 - E
Error Type:	EPROM/EEPROM and Cassette Verification Error.
When Detected:	During a cassette verification.
Description:	The contents of processor memory and the cassette tape do not match.
Corrective Action:	Re-save the program onto tape or re-load the program from tape and try the verification again.
Error Code:	C 8 - E
Error Type:	Format Error
When Detected:	During a cassette verification.
Description:	The contents of the cassette tape are not in the proper data format.
Corrective Action:	Replace the cassette tape with a tape that is known to be good. Check the setting of the cassette recorder's playback control.
Error Code:	C - - H
Error Type:	Time-out Error
When Detected:	During a cassette verification.
Description:	The cassette tape cannot be read. The letter "H" is displayed for 30 seconds until the tape is rewound.
Corrective Action:	Replace the cassette tape with a tape that is known to be good. Check the cables connecting the cassette recorder and the handheld programmer. Be sure the cables are tightly connected. Check the setting of the cassette recorder's playback control.
Error Code:	Step number E
Error Type:	Syntax Error
When Detected:	When the processor is first put into run or when the program syntax is checked.
Description:	The ORG, MCS, MCR, JMP, or JMP END instructions have been programmed improperly or have been placed in the wrong part of the program.
Corrective Action:	Correct the program.

Table 7.3 - Error Codes (Continued)

Error Code:	Step number ^
Error Type:	Memory Stack Error
When Detected:	When the processor is first put into run or when the program syntax is checked.
Description:	The STR/STR NOT instructions have been used more than 8 times, the MCS instruction has been used more than 4 times, the JMP instruction has been used more than twice, the number of MCS instructions does not match the number of MCR instructions, the number of AND STR/OR STR instructions does not match the number of STR/STR NOT instructions, or the STR/STR NOT instructions have been used too many times with FUN03, FUN40, FUN45.
Corrective Action:	Delete the excess STR/STR NOT instructions, delete or add MCS/MCR instructions, delete excess AND STR/OR STR instructions or add STR/STR NOT instructions, delete excess STR/STR NOT instructions when used with FUN03, FUN40, FUN45.
Error Code:	Step number u
Error Type:	Memory Stack Error
When Detected:	When the processor is first put into run or when the program syntax is checked.
Description:	The number of MCR instructions does not match the number of MCS instructions, the number of AND STR/OR STR instructions does not match the number of STR/STR NOT instructions, a STR or OUT instruction is missing.
Corrective Action:	Either add a MCR instruction or delete a MCS instruction, add an AND STR/OR STR or STR/STR NOT instruction, or add an OUT instruction.
Error Code:	Step number F
Error Type:	Framing Error
When Detected:	When the processor is first put into run or when the program syntax is checked.
Description:	Syntax error.
Corrective Action:	Correct the programming error. If the problem remains, replace the processor module.
Error Code:	^
Error Type:	Keystroke Error
When Detected:	When keying-in the program.
Description:	Memory is full, a non-program area of memory was accessed, or the [STEP +] key was pressed after 926 steps or rungs.
Corrective Action:	Press the [CLR] key to clear the error and re-enter the data.

Table 7.3 - Error Codes (Continued)

Error Code:	u
Error Type:	Keystroke Error
When Detected:	When keying-in the program.
Description:	The [STEP -] key was pressed immediately after system power was turned on or program execution began.
Corrective Action:	Press the [CLR] key to clear the error and re-enter the data.
Error Code:	E
Error Type:	Keystroke Error
When Detected:	When keying-in the program.
Description:	Error when keying-in the program.
Corrective Action:	Press the [DCLR] or [CLR] keys to clear the error and re-enter the data.

7.4 System Error Codes

The ERR LED on the processor module will go on when a system error is detected. Cycle power and monitor address 970 in decimal to troubleshoot the problem. Address 970 will contain an error code in decimal. This code will identify the type of error. See Table 7.4.

If applicable, address 972 will contain the number of the element where the error occurred. To clear a system error, clear memory and reload the program.

Table 7.4 - System Error Codes

System Error Code in Decimal	Cause of Error
10	Trap interrupt
11	Stack pointer error
12	Error in logic programming
13	Interrupt error
14	Interrupt error
20	Memory error
21	ROM checksum error
30	Undefined instruction
31	PCS stack pointer error
32	User program checksum error
40	Buffer overflow

7.5 Output Module Fuse Replacement

Use the following procedure to replace a fuse on an output module:

Step 1. Be sure the system is in a safe state and stop the application program from executing.

Step 2. Remove main input power from the rack. Remove power from the field wiring.

> **DANGER**
> **EXTERNAL POWER WIRING MAY REMAIN ENERGIZED WHEN THE MAIN A-C POWER IS DISCONNECTED. IDENTIFY ALL SUCH EXTERNAL WIRING AND LOCK OUT OR TAG IT. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.**

Step 3. Disconnect the external output wires that are attached to the module. Be sure to mark the wires so that they can be easily re-installed later.

If the module has a removable terminal strip (M/N 45C959, 45C964, and 45C965), you do not have to remove the wires attached to the terminal strip. Just remove the two screws that hold the terminal strip to the module's faceplate. See figure 3.25.

Step 4. Remove the module from the rack. Press the tabs at the top and bottom of the module and carefully pull it out of the rack's backplane.

Step 5. Carefully remove the blown fuse and replace it with a new one. Fuse specifications are given in Appendix A.

Step 6. Insert the module into the correct slot in the rack. Press the module into the slot until the locking tabs on the top and bottom of the module latch into the rack.

Step 7. Re-connect the external output wires or the terminal strip.

Step 8. Restore main input power to the rack. Restore power to the field wiring.

7.6 Power Supply Module Fuse Replacement

Use the following procedure to replace a fuse on a Power Supply module:

Step 1. Be sure the system is in a safe state and stop the application program from executing.

Step 2. Remove main input power from the rack. Remove power from the field wiring.

> **DANGER**
>
> **EXTERNAL POWER WIRING MAY REMAIN ENERGIZED WHEN THE MAIN A-C POWER IS DISCONNECTED. IDENTIFY ALL SUCH EXTERNAL WIRING AND LOCK OUT OR TAG IT. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.**

Step 3. Disconnect the input power wires from the L1, L2, and GND terminals.

Step 4. Remove the module from the rack. Press the tabs at the top and bottom of the module and carefully pull it out of the rack's backplane.

Step 5. Carefully remove the blown fuse and replace it with a new one. Fuse specifications are given in Appendix A.

Step 6. Insert the module into the correct slot in the rack. Press the module into the slot until the locking tabs on the top and bottom of the module latch into the rack.

Step 7. Re-connect the input power wires to the L1, L2, and GND terminals.

Step 8. Restore main input power to the rack. Restore power to the field wiring.

7.7 Processor Memory Replacement

Use the following procedure to replace a processor's EPROM or EEPROM memory:

Step 1. Be sure the system is in a safe state and stop the application program from executing.

Step 2. Remove main input power from the rack. Remove power from the field wiring.

DANGER

EXTERNAL POWER WIRING MAY REMAIN ENERGIZED WHEN THE MAIN A-C POWER IS DISCONNECTED. IDENTIFY ALL SUCH EXTERNAL WIRING AND LOCK OUT OR TAG IT. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

Step 3. Remove the processor from the rack. Press the tabs at the top and bottom of the module and carefully pull it out of the rack's backplane.

CAUTION

THIS MODULE CONTAINS STATIC-SENSITIVE COMPONENTS. CARELESS HANDLING CAN CAUSE SEVERE DAMAGE.

DO NOT TOUCH THE CONNECTORS ON THE BACK OF THE MODULE. WHEN NOT IN USE, THE MODULE SHOULD BE STORED IN AN ANTI-STATIC BAG. THE PLASTIC COVER SHOULD NOT BE REMOVED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN DAMAGE TO OR DESTRUCTION OF THE EQUIPMENT.

Step 4. The EPROM/EEPROM memory connects to the processor through a slide-in connector. See figure 2.3. Remove the EPROM/EEPROM memory module from the processor by carefully sliding it out away from the connector. Lift the memory module away from processor once the nylon standoffs have cleared the notches cut into the processor's printed circuit board.

Step 5. Install the replacement EPROM/EEPROM memory module by placing it on the processor module and aligning it with the slide-in connector. Carefully push the memory module into the connector until the nylon standoffs have secured the module to the processor.

Step 6. Insert the processor module into the correct slot in the rack. Press the module into the slot until the locking tabs on the top and bottom of the module latch into the rack.

Step 7. Restore main input power to the rack. Restore power to the field wiring.

Appendix A

System Model Numbers

Shark XL II CPU Module

- M/N 45C901
- Memory is not included with the module

Memory

- M/N 45C975 926-word EEPROM
- M/N 45C977 3,997-word EEPROM
- M/N 45C978 3,997-word EPROM

Power Supply Module

- M/N 45C920 115/230 VAC
- M/N 45C921 24 VDC
- M/N 45C922 115/230 VAC

Racks

- M/N 45C910 3-Slot Rack
- M/N 45C911 4-Slot Rack
- M/N 45C912 5-Slot Rack
- M/N 45C913 6-Slot Rack
- M/N 45C914 7-Slot Rack

Digital Input Modules

- M/N 45C940 8-Channel 115/230 VAC Input
- M/N 45C941 16-Channel 115/230 VAC Input
- M/N 45C942 16-Channel 115/230 VAC Input with Removable Terminal Block
- M/N 45C944 8-Channel 24 VDC Input
- M/N 45C945 16-Channel 24 VDC Input
- M/N 45C946 16-Channel 24 VDC Input with Removable Terminal Block

Appendix A

System Model Numbers (Continued)

Digital Output Modules

- M/N 45C959 16-Channel Relay Output with Removable Terminal Block
- M/N 45C960 8-Channel 115/230 VAC Output
- M/N 45C961 16-Channel 115/230 VAC Output
- M/N 45C962 8-Channel 5-27 VDC Output
- M/N 45C963 16-Channel 5-27 VDC Output
- M/N 45C964 16-Channel 24 VDC Output with Removable Terminal Block
- M/N 45C965 16-Channel 115/230 VAC Output with Removable Terminal Block
- M/N 45C966 8-Channel Relay Output
- M/N 45C967 16-Channel Relay Output
- M/N 45C968 8-Channel Isolated High Power Relay Output
- M/N 45C969 8-Channel Isolated Low Power Relay Output

Input/Output Modules

- M/N 45C957 16-Channel TTL Input/16-Channel TTL Output
- M/N 45C958 8-Channel 24 VDC Input/8-Channel 24 VDC Output

Analog Input Modules

- M/N 45C990 8-Channel 4-20 mA Input
- M/N 45C992 8-Channel 0-10 VDC Input

Analog Output Modules

- M/N 45C993 2-Channel 4-20 mA Output
- M/N 45C994 2-Channel 0-10 VDC Output
- M/N 45C995 4-Channel 4-20 mA Output
- M/N 45C997 4-Channel 0-10 VDC Output

Counter Module

- M/N 45C982

Network Module

- M/N 45C987

Appendix A

System Model Numbers (Continued)

Remote Master Module

- M/N 45C988

Remote Target Module

- M/N 45C989

Expansion Rack Cable

- M/N 45C938

Blank Faceplate for Unused Module Slots

- M/N 45C999

Standard Programmer

- M/N 45C950

  Cassette recorder interface

Universal Programmer

- M/N 45C951

  Cassette recorder interface
  PROM programming capability
  RS-232 interface

ReSource Shark IBM-Programming and Documentation Executive (SPX)

- M/N 45C152 Software, key, and instruction manual
- M/N 45C153 Upgrade (Software and instruction manual)

Shark XL II SPX Programming Cables

- M/N 45C973 9-Pin Programming Cable
- M/N 45C974 25-Pin Programming Cable

Appendix A

Shark XL II CPU Module Specifications

Model Number

- M/N 45C901 Shark XL II CPU Module

Ambient Conditions

- Storage Temperature: 4° to 158°F
  −20° to 70°C
- Operating Temperature: 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% (non-condensing)

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Programming Languages

- Boolean and standard ladder logic with use of the IBM-compatible software

Program Execution Speed

- 1.5 micro-seconds (average) per standard instruction

Memory Capacity

- EEPROM (926 and 3997 words)
- EPROM (3997 words)

Standard Instructions

- ORG, STR, AND, OR, AND STR, OR STR, NOT, OUT

Arithmetic Instructions

- Word load, word out, addition, subtraction, multiplication, division, word AND, word OR, comparison (equal to or greater than, equal to, less than), out carry, BCD conversion, BNR conversion, set, reset, binary addition, binary subtraction, binary multiplication, binary division, binary comparison, exclusive OR, clear carry, set carry, ASCII conversion, seven-segment conversion, bit mask

Appendix A

Shark XL II CPU Module Specifications (Continued)

Application Instructions

- Shift register, master control, IF command, up/down counters, jump, addressed jump, latch, return, start, end, clockwise and counter-clockwise bit shifting, define subroutine, call subroutine, return from subroutine, immediate I/O update

Real-Time Clock

- Processor accessible

Number of External I/O

- 160

Number of Internal Outputs

- 256 Non-retentive
- 256 Retentive

Number of Special Function Internal Outputs

- 32

Number of Timer/Counters

- 96

Timer Preset Values

- 0.1 to 999.9 seconds (10 points maximum), 0.1 to 99.9 seconds, or 1 to 999 seconds

Counter Preset Values

- 1 to 9,999 times (10 points maximum), 1 to 999 times

Number of Remote Digital I/O

- 128

Maximum Remote I/O Distance

- 1000 Feet (300 Meters)

Number of Shark Network Nodes

- 1 Master Node
- 7 Slave Nodes

Appendix A

Shark XL II CPU Module Specifications (Continued)

Maximum Shark Network Distance

- 1000 Feet (300 Meters)

Agency Approvals

- UL Listed

RS-232 Port Pinouts

25-Pin Connection to an IBM-Compatible Computer	Pin	Pin	9-Pin Connection to the Shark XL II CPU Module
XMIT Data	2	2	RECV Data
RECV Data	3	3	XMIT Data
RTS	4	8	CTS
CTS	5	7	RTS
Sig Com	7	5	Sig Com
FG	1	1	FG
DR	6		
ER	20		

9-Pin Connection to an IBM-Compatible Computer	Pin	Pin	9-Pin Connection to the Shark XL II CPU Module
XMIT Data	3	2	RECV Data
RECV Data	2	3	XMIT Data
RTS	7	8	CTS
CTS	8	7	RTS
DSR	6	4	DTR
Sig Com	5	5	Sig Com

Appendix A

Shark XL CPU Module Specifications

Model Number

- M/N 45C900 Shark XL CPU Module

Ambient Conditions

- Storage Temperature: 4° to 158°F
  −20° to 70°C
- Operating Temperature: 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% (non-condensing)

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Programming Languages

- Boolean and standard ladder logic with use of the IBM-compatible software

Program Execution Speed

- 5 micro-seconds (average) per standard instruction

Memory Capacity

- EEPROM (926 and 1926 words)
- EPROM (1926 words)

Standard Instructions

- ORG, STR, AND, OR, AND STR, OR STR, NOT, OUT

Arithmetic Instructions

- Word load, word out, addition, subtraction, multiplication, division, word AND, word OR, comparison (equal to or greater than, equal to, less than), out carry, BCD conversion, binary conversion

Application Instructions

- Bit shift register, master control, IF command, up/down counters, jump, addressed jump, latch, return, start, end

Appendix A

Shark XL CPU Module Specifications (Continued)

Number of External I/O

- 160

Number of Internal Outputs

- 256 Non-retentive
- 256 Retentive

Number of Special Function Internal Outputs

- 32

Number of Timer/Counters

- 96

Timer Preset Values

- 0.1 to 999.9 seconds (10 points maximum), 0.1 to 99.9 seconds, or 1 to 999 seconds

Counter Preset Values

- 1 to 9,999 times (10 points maximum), 1 to 999 times

Number of Remote Digital I/O

- 128

Maximum Remote I/O Distance

- 1000 Feet (300 Meters)

Number of Shark Network Nodes

- 1 Master Node
- 7 Slave Nodes

Maximum Shark Network Distance

- 1000 Feet (300 Meters)

Agency Approvals

- UL Listed

Appendix A

A-C Power Supply Module Specifications

Model Number

- M/N 45C920 115/230 VAC Power Supply Module
- M/N 45C922 115/230 VAC Power Supply Module

Ambient Conditions

- Storage Temperature: −4° to 158°F
−20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Input Voltage

- 115 VAC or 230 VAC (jumper selectable)

Input Frequency

- 47-63 Hz

Input Current

- 600 mA or less

Fuse Type

M/N45C920

- Buss™ 2 Amp, 250 V, GMA, or Equivalent

M/N45C922

- Wickmannwerke GmbH, M/N K19374, 2 amp, 250V, or Equivalent

Appendix A

A-C Power Supply Module Specifications (Continued)

Output

- The total power consumption of the modules in the rack must not exceed the maximum load current of the power supply module. Check the power requirements of each I/O module to be sure that the total current consumption does not exceed the maximum load current of channels 1, 2, and 3 as shown below.

M/N 45C920

- Channel 1: 5 VDC at 1000 mA (Maximum load current)
- Channel 2: 24 VDC at 300 mA (Maximum load current)
- Channel 3: 24 VDC at 450 mA (Maximum load current)

M/N 45C922

- Channel 1: 5 VDC at 1700 mA (Maximum load current)
- Channel 2: 24 VDC at 500 mA (Maximum load current)
- Channel 3: 24 VDC at 250 mA (Maximum load current)

Operating Environment

- Operation in areas containing corrosive gases, saline air, or iron powder is prohibited.

Power Interruption Time Period

- 20 milli-seconds

A-C Power Noise Immunity

- Simulated noise 1500V peak to peak, 1 micro-second width

Insulation Resistance

- 20M ohms or more between an external terminal and frame ground (GND) terminal as measured with a 500 VDC Megger

Dielectric Strength

- 1500 VAC for one minute between the power/output terminals and the frame ground (GND) terminal

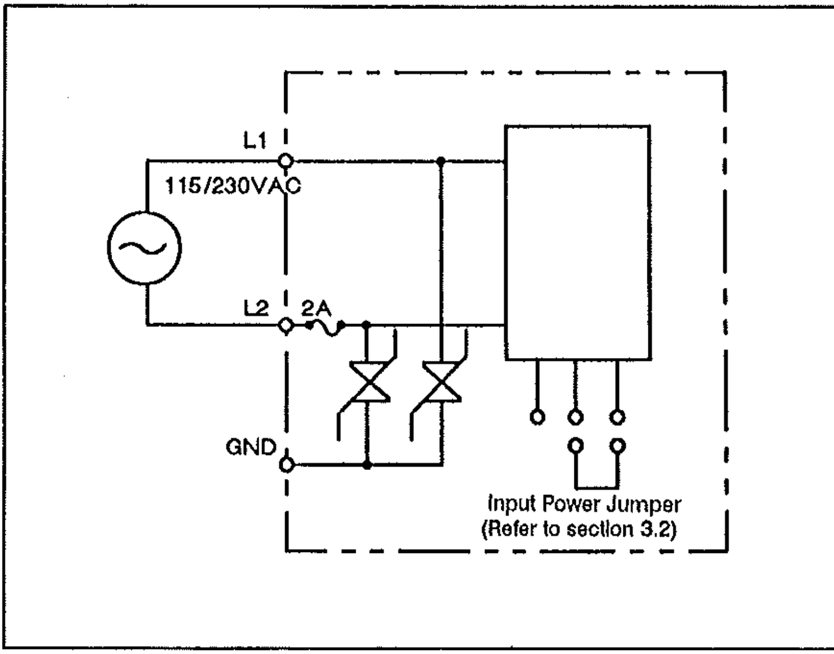
Vibration Resistance

- Vibration with a frequency of 16.7 Hz and an amplitude of three millimeters applied for two hours in X, Y, and Z directions

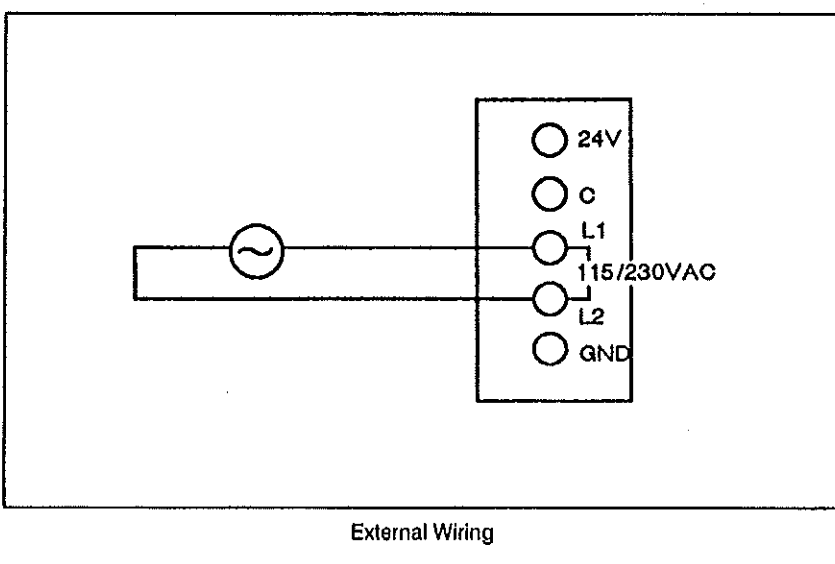
A-C Power Supply Module Specifications (Continued)

Agency Approvals

- UL Listed



Circuit Diagram



External Wiring

Appendix A

D-C Power Supply Module Specifications

Model Number

- M/N 45C921 24 VDC Power Supply Module

Ambient Conditions

- Storage Temperature: −4° to 158°F
  −20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.6 cm)
- Weight: 7 ounces (0.2 kilograms)

Input Voltage

- 19.2-30 VDC

Input Current

- 1.6 Amps or less

Fuse Type

- Buss™ 3 Amp, 250 V, GMA, or Equivalent

Output

- Channel 1: 5 VDC at 1000 mA (Maximum load current)
- Channel 2: 24 VDC at 300 mA (Maximum load current)
- Channel 3: 24 VDC at 1000 mA (Maximum load current)
- The total power consumption of the modules in the rack must not exceed the maximum load current of the power supply module. Check the power requirements of each I/O module to be sure that the total current consumption does not exceed the maximum load current of channels 1, 2, and 3 as shown above.

Operating Environment

- Operation in areas containing corrosive gases, saline air, or iron powder is prohibited.

Appendix A

D-C Power Supply Module Specifications (Continued)

Power Interruption Time Period

- 20 milli-seconds

D-C Power Noise Immunity

- 500V peak to peak, 1 micro-second width

Insulation Resistance

- 20M ohms or more between an external terminal and frame ground (GND) terminal as measured with a 500 VDC Megger

Dielectric Strength

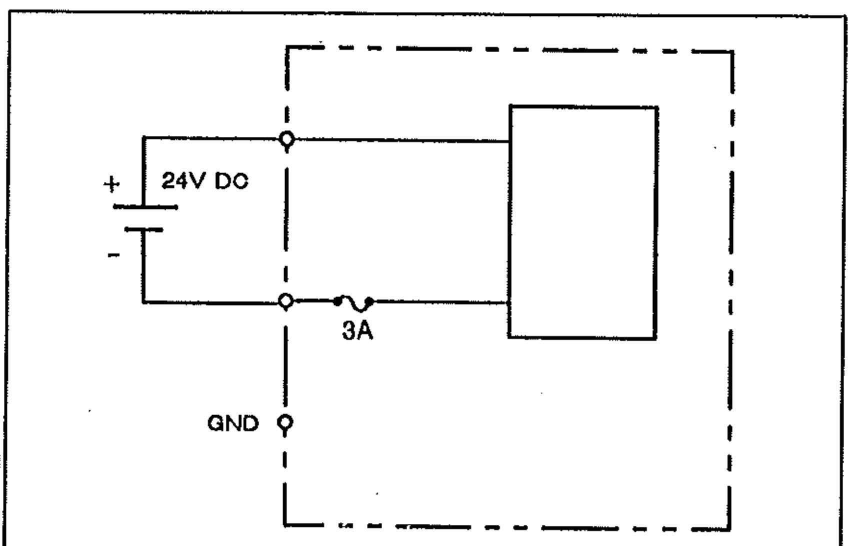
- 1500 VAC for one minute between the power/output terminals and the frame ground (GND) terminal

Vibration Resistance

- Vibration with a frequency of 16.7 Hz and an amplitude of three millimeters applied for two hours in X, Y, and Z directions

Agency Approvals

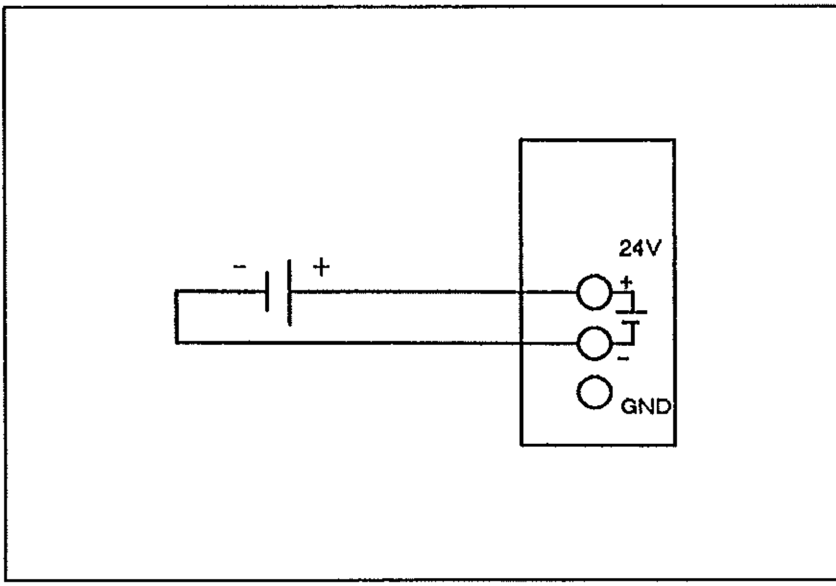
- UL Listed



Circuit Diagram

Appendix A

D-C Power Supply Module Specifications (Continued)



External Wiring

Appendix A

24 VDC Input Module Specifications

Model Numbers

- M/N 45C944 8-Channel 24 VDC Input Module
- M/N 45C945 16-Channel 24 VDC Input Module
- M/N 45C946 16-Channel 24 VDC Input Module with Removable Terminal Block

Ambient Conditions

- Storage Temperature: -4° to 158°F
  -20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Nominal Voltage

- 24 VDC

Input Voltage

- 21.6-26 VDC

Input Current

- 9 mA Sourcing

Operational States

- On Voltage: 19 VDC or more
- Off Voltage: 7 VDC or less

Input Delay Time

- On to Off: 4 milliseconds or less
- Off to On: 4 milliseconds or less

Appendix A

24 VDC Input Module Specifications (Continued)

Common Output Connection

- 8 input points per common terminal

Polarity

- Common terminal (-)

Isolation Method

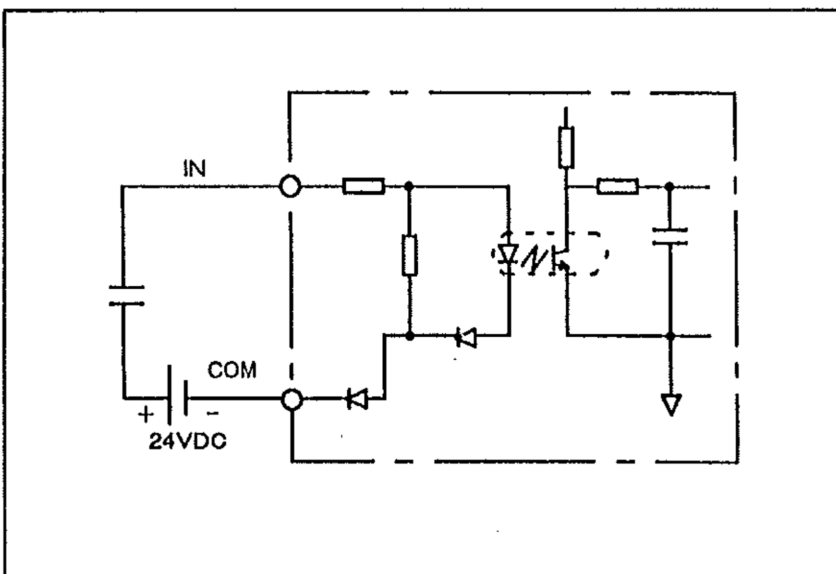
- Opto-isolator

Average Module Current Consumption

- Channel 1: 0.5 mA + (number of ON inputs x 0.5 mA)
- Channel 2: 0 mA
- Channel 3: 0 mA

Agency Approvals

- UL Listed



Circuit Diagram

Appendix A

115 VAC Input Module Specifications

Model Numbers

- M/N 45C940 8-Channel 115 VAC Input Module
- M/N 45C941 16-Channel 115 VAC Input Module
- M/N 45C942 16-Channel 115 VAC Input Module with Removable Terminal Block

Ambient Conditions

- Storage Temperature: −4° to 158°F
  −20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Nominal Voltage

- 115/230 VAC

Input Voltage

- 85-264 VAC

Input Current

- 7 mA (100 VAC)

Operational States

- On Voltage: 85 VAC or greater
- Off Voltage: 30 VAC or less

Input Delay Time

- On to Off: 16 milliseconds or less
- Off to On: 16 milliseconds or less

Appendix A

115 VAC Input Module Specifications (Continued)

Common Output Connection

- 8 input points per common terminal

Isolation Method

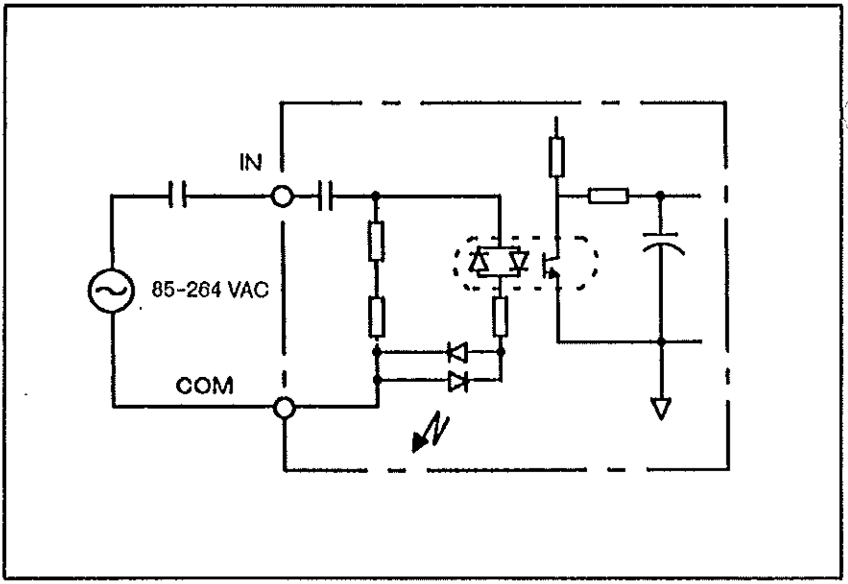
- Opto-isolator

Average Module Current Consumption

- Channel 1: 1 mA
- Channel 2: 0 mA
- Channel 3: 0 mA

Agency Approvals

- UL Listed



Circuit Diagram

Appendix A

Analog Current Input Module Specifications

Model Number

- M/N 45C990 8-Channel 4-20 mA Analog Input Module

Ambient Conditions

- Storage Temperature: -4° to 158°F
  -20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Current Range

- 4-20 mA

Input Impedance

- 220 ohms

Resolution

- 8 bits

Conversion Time

- 1 millisecond

Overall Accuracy

- +/- 1% + 1 bit

Isolation Method

- Not isolated from D-C inputs

Isolation Between Inputs

- Not provided

Appendix A

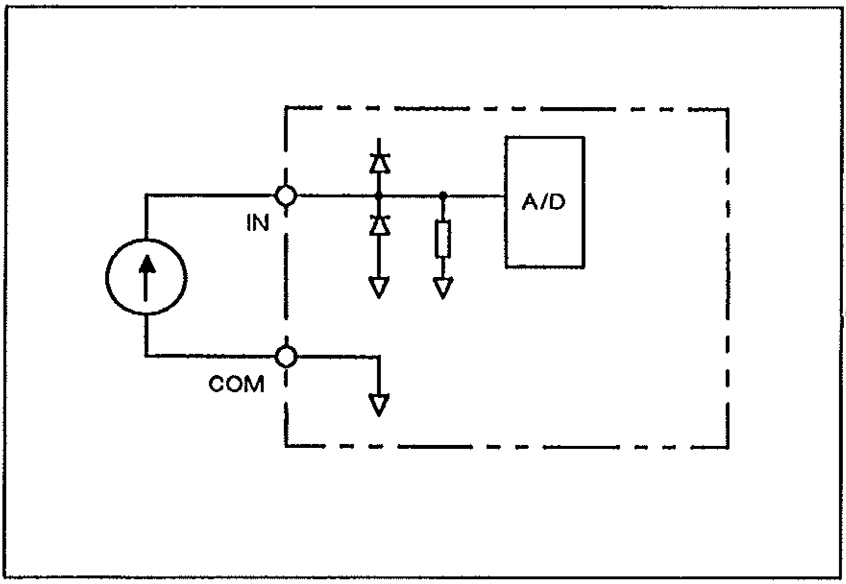
Analog Current Input Module Specifications (Continued)

Average Module Current Consumption

- Channel 1: 25 mA
- Channel 2: 0 mA
- Channel 3: 60 mA

Agency Approvals

- UL Listed



Circuit Diagram

Appendix A

Analog Voltage Input Module Specifications

Model Number

- M/N 45C992 8-Channel 0-10 VDC Analog Input Module

Ambient Conditions

- Storage Temperature: −4° to 158°F
  −20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Voltage Range

- 0-10 VDC

Input Impedance

- 100K ohms

Resolution

- 8 bits

Conversion Time

- 1 millisecond

Overall Accuracy

- +/- 1% + 1 bit

Isolation Method

- Not isolated from D-C inputs

Isolation Between Inputs

- Not provided

Appendix A

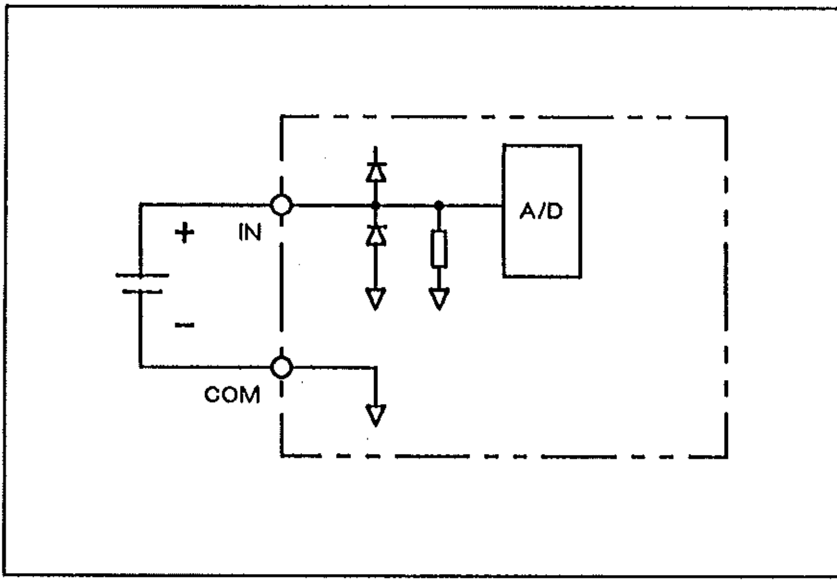
Analog Voltage Input Module Specifications (Continued)

Average Module Current Consumption

- Channel 1: 25 mA
- Channel 2: 0 mA
- Channel 3: 60 mA

Agency Approvals

- UL Listed



Circuit Diagram

Appendix A

Analog Current Output Module Specifications

Model Numbers

- M/N 45C993 2-Channel 4-20 mA Analog Output Module
- M/N 45C995 4-Channel 4-20 mA Analog Output Module

Ambient Conditions

- Storage Temperature: −4° to 158°F
  −20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Current Range

- 4-20 mA

Load Impedance

- 0-500 ohms

Resolution

- 8 bits

Conversion Time

- 1 millisecond

Overall Accuracy

- +/- 1%

Isolation Method

- Not isolated from D-C inputs

Appendix A

Analog Current Output Module Specifications (Continued)

Isolation Between Outputs

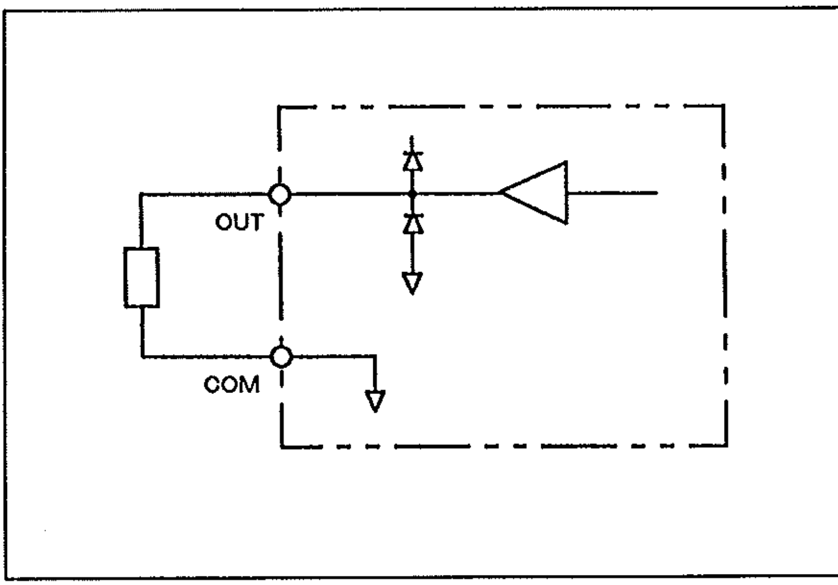
- Not provided

Average Module Current Consumption

- Channel 1: 50 mA
- Channel 2: 0 mA
- Channel 3: 140 mA (M/N 45C993)
  250 mA (M/N 45C995)

Agency Approvals

- UL Listed



Circuit Diagram

Appendix A

Analog Voltage Output Module Specifications

Model Numbers

- M/N 45C994 2-Channel 0-10 VDC Analog Output Module
- M/N 45C997 4-Channel 0-10 VDC Analog Output Module

Ambient Conditions

- Storage Temperature: −4° to 158°F
−20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Voltage Range

- 0-10 VDC

Load Impedance

- 10K ohms or more

Resolution

- 8 bits

Conversion Time

- 1 millisecond

Overall Accuracy

- +/- 1%

Isolation Method

- Not isolated from D-C inputs

Appendix A

Analog Voltage Output Module Specifications (Continued)

Isolation Between Outputs

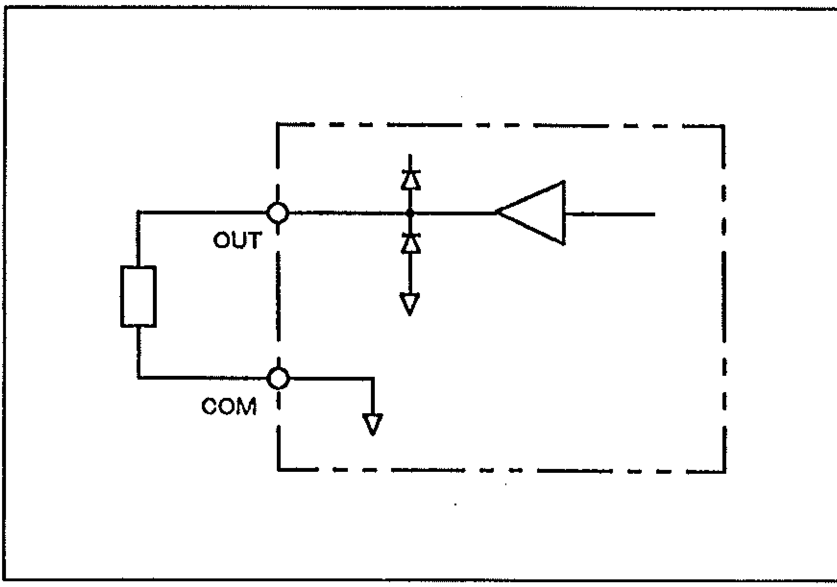
- Not provided

Average Module Current Consumption

- Channel 1: 30 mA (M/N 45C994)
  50 mA (M/N 45C997)
- Channel 2: 0 mA
- Channel 3: 70 mA (M/N 45C994)
  140 mA (M/N 45C997)

Agency Approvals

- UL Listed



Circuit Diagram

Appendix A

Relay Output Module Specifications

Note that a surge suppressor is required with all inductive loads.

Model Numbers

- M/N 45C966 8-Channel Relay Output Module
- M/N 45C967 16-Channel Relay Output Module
- M/N 45C959 16-Channel Relay Output Module with Removable Terminal Block

Ambient Conditions

- Storage Temperature: −4° to 158°F
−20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Nominal Voltage

- 115/230 VAC, 24 VDC

Output Voltage

- 21-27 VDC, 85-264 VAC

Maximum Load Current

- 1 circuit: 2 Amps
- 8 circuits: 4 Amps

Maximum Inrush Current

- 6 Amps for 100 milliseconds

Maximum Output Delay Time

- On to Off: 4 milliseconds
- Off to On: 5 milliseconds

Appendix A

Relay Output Module Specifications (Continued)

Common Output Connection

- 8 output points per common terminal
- 1 fuse per common

Isolation Method

- Relay

Fuse Type

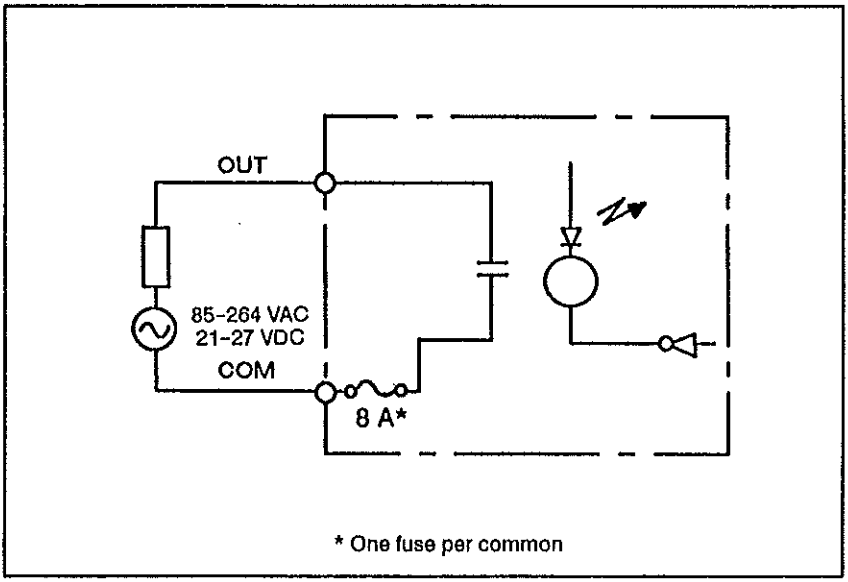
- SOC, 8 Amp, 250 V, MQ4

Average Module Current Consumption

- Channel 1: 0.2 mA + (number of ON outputs x 0.2 mA)
- Channel 2: number of ON outputs x 10 mA
- Channel 3: 0 mA

Agency Approvals

- UL Listed



Circuit Diagram

Appendix A

Isolated Relay Output Module Specifications

Note that a surge suppressor is required with all inductive loads.

Model Number

- M/N 45C968 High Power Isolated Relay Output Module
- M/N 45C969 Low Power Isolated Relay Output Module

Ambient Conditions

- Storage Temperature: −4° to 158°F
  −20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Nominal Voltage

- 115/230 VAC, 24 VDC

Output Voltage

- 85-264 VAC
- 5-27 VDC

Minimum Load Current

- M/N 45C968: 10 mA
- M/N 45C969: 0.1 mA

Maximum Load Current

- M/N 45C968: 2 Amps
- M/N 45C969: 100 mA

Maximum Inrush Current

- M/N 45C968: 6 Amps for 100 milliseconds
- M/N 45C969: 1 Amp for 100 milliseconds

Appendix A

Isolated Relay Output Module Specifications (Continued)

Maximum Output Delay Time

- On to Off: 4 milliseconds
- Off to On: 5 milliseconds

Number of Outputs

- 8

Common Output Connections

- 1 output per common terminal

Isolation Method

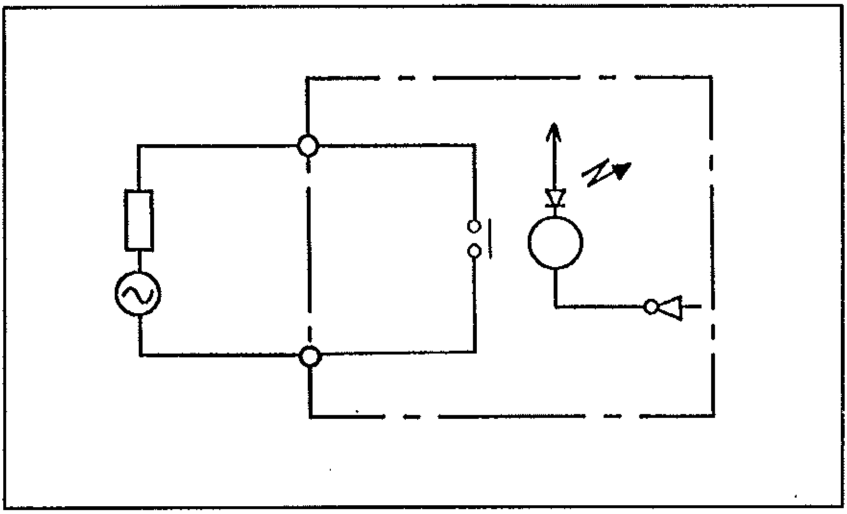
- Relay

Average Module Current Consumption

- Channel 1: 0.2 mA + (number of ON outputs x 0.2 mA)
- Channel 2: number of ON outputs x 10 mA
- Channel 3: 0 mA

Agency Approvals

- UL Listed



Circuit Diagram

Appendix A

115/230 VAC Output Module Specifications

Note that a surge suppressor is required with all inductive loads.

Model Numbers

- M/N 45C960 8-Channel 115/230 VAC Output Module
- M/N 45C961 16-Channel 115/230 VAC Output Module
- M/N 45C965 16-Channel 115/230 VAC Output Module with Removable Terminal Block

Ambient Conditions

- Storage Temperature: −4° to 158°F
  −20° to 70°C
- Operating Temperature: (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Nominal Voltage

- 115/230 VAC

Output Voltage

- 85-264 VAC

Maximum Load Current

- 1 circuit: 1 Amp
- 4 circuits: 2 Amps

Maximum Leakage Current

- 1 mA (100 VAC)

Maximum Inrush Current

- 20 Amps (1 Cycle)

Appendix A

115/230 VAC Output Module Specifications (Continued)

Maximum Output Delay Time

- On to Off: 11 milliseconds
- Off to On: 11 milliseconds

Common Output Connection

- 8 output points per common terminal
- 1 fuse per common

Fuse Type

- SOC, 8 Amp, 250 V, MQ4

Isolation Method

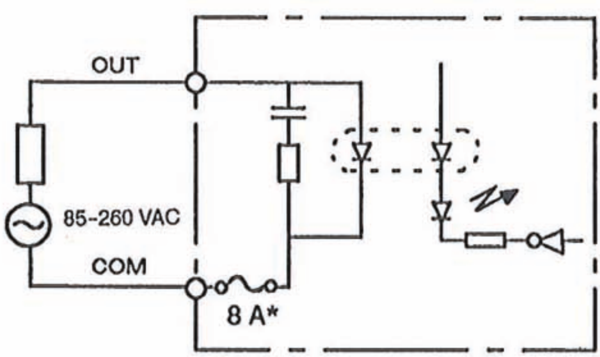
- Opto-isolator

Average Module Current Consumption

- Channel 1: 0.3 mA - (number of ON outputs x 0.2 mA)
- Channel 2: number of ON outputs x 6.5 mA
- Channel 3: 0 mA

Agency Approvals

- UL Listed



* One fuse per common

Circuit Diagram

Appendix A

5-27 VDC Output Module Specifications

Note that a surge suppressor is required with all inductive loads.

Model Numbers

- M/N 45C962 8-Channel 5-27 VDC Output Module
- M/N 45C963 16-Channel 5-27 VDC Output Module
- M/N 45C964 16-Channel 5-27 VDC Output Module with Removable Terminal Block

Ambient Conditions

- Storage Temperature: −4° to 158°F
  −20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Nominal Voltage

- 24 VDC

Output Voltage

- 5-27 VDC

Maximum Load Current from Sourcing Supply

- 1 circuit: 0.5 Amp
- 4 circuits: 1.25 Amps
- 8 circuits: 2.5 Amps

Maximum Leakage Current

- 0.1 mA

Maximum Inrush Current

- 3 Amps

Appendix A

5-27 VDC Output Module Specifications (Continued)

Maximum Output Delay Time

- On to Off: 1 millisecond
- Off to On: 1 millisecond

Common Output Connection

- 8 output points per common terminal
- 1 fuse per common

Fuse Type

- SOC, 8 Amp, 250 V, MQ4

Polarity

- Common terminal

Isolation Method

- Opto-isolator

Average Module Current Consumption

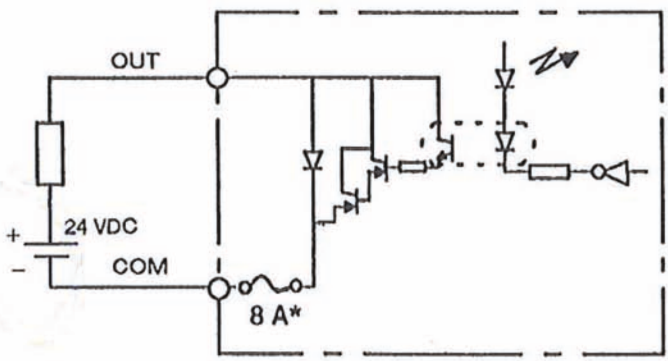
- Channel 1: 0.2 mA + (number of ON outputs x 0.2 mA)
- Channel 2: (number of ON outputs) x 6.5 mA
- Channel 3: 0 mA

Agency Approvals

- UL Listed

Appendix A

5-27 VDC Output Module Specifications (Continued)



* One fuse per common

Circuit Diagram

Appendix A

16-Channel TTL Input/Output Module Specifications

Note that a surge suppressor is required with all inductive loads.

Model Number

- M/N 45C957 16-Channel TTL Input/Output

Ambient Conditions

- Storage Temperature: −4° to 158°F
  −20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Number of Inputs

- 16

Number of Outputs

- 16

Nominal I/O Voltage

- 4-15 VDC

Input Current Sinking

- 6 ma at 5 VDC

Input Voltage

- On: 1.5 VDC or less
- Off: 3.5 VDC or more

Maximum Load Current Sinking

- 20 mA per output

Appendix A

16-Channel TTL Input/Output Module Specifications (Continued)

Maximum Leakage Current

- 50 uA

Input Common Connection

- 16 input points per common terminal

Maximum Input and Output Delay Time

- On to Off: 1 millisecond
- Off to On: 1 millisecond

Output Common Connection

- 8 output points per common terminal

Polarity

- Common terminal

Isolation Method

- Opto-Isolator

Average Module Current Consumption

- Channel 1: (number of ON outputs x 5 mA) + 30 mA
- Channel 2: 0 mA
- Channel 3: 0 mA

External Wiring Connector

- AWG 22 to 26: Hirose Denki Connector HIF3-2226SC
- AWG 24 to 28: Hirose Denki Connector HIF3-2428SC
- On-board socket: Hirose Denki HIF3C-40D-2.54C

Agency Approvals

- UL Listed

Appendix A

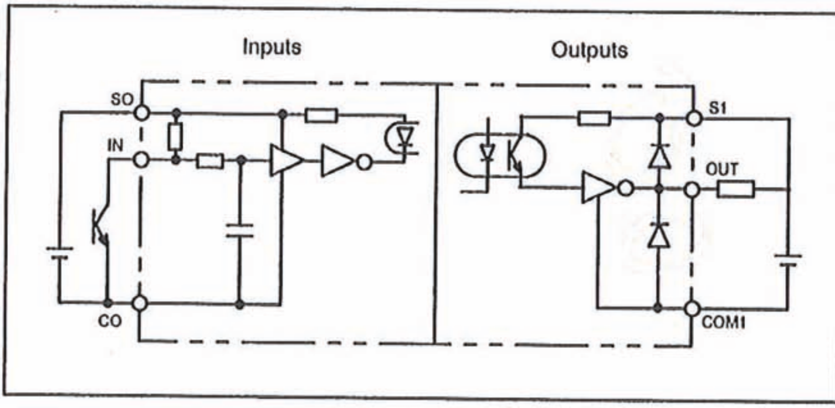
16-Channel TTL Input/Output Module Specifications (Continued)

Forty-Pin Connector Pinout

• Pin 1 = Input Common 0	Pin 2 = Output Common 1
• Pin 3 = Input Positive	Pin 4 = Output Positive for 0 to 7
• Pin 5 = Input 0	Pin 6 = Output 0
• Pin 7 = Input 1	Pin 8 = Output 1
• Pin 9 = Input 2	Pin 10 = Output 2
• Pin 11 = Input 3	Pin 12 = Output 3
• Pin 13 = Input 4	Pin 14 = Output 4
• Pin 15 = Input 5	Pin 16 = Output 5
• Pin 17 = Input 6	Pin 18 = Output 6
• Pin 19 = Input 7	Pin 20 = Output 7
• Pin 21 = No Connection	Pin 22 = Output Common 2
• Pin 23 = No Connection	Pin 24 = Output Positive for 8 to 15
• Pin 25 = Input 8	Pin 26 = Output 8
• Pin 27 = Input 9	Pin 28 = Output 9
• Pin 29 = Input 10	Pin 30 = Output 10
• Pin 31 = Input 11	Pin 32 = Output 11
• Pin 33 = Input 12	Pin 34 = Output 12
• Pin 35 = Input 13	Pin 36 = Output 13
• Pin 37 = Input 14	Pin 38 = Output 14
• Pin 39 = Input 15	Pin 40 = Output 15

Appendix A

16-Channel TTL Input/Output Module Specifications (Continued)



Circuit Diagram

Appendix A

8-Channel 24 VDC Input/Output Module Specifications

Note that a surge suppressor is required with all inductive loads.

Model Number

- M/N 45C958 8-Channel 24 VDC Input/Output

Ambient Conditions

- Storage Temperature: −4° to 158°F
−20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Number of Inputs

- 8

Number of Outputs

- 8

Nominal Voltage

- 24 VDC

Voltage Range

- Input: 21.6 to 26 VDC
- Output: 5 to 27 VDC

Input Current Sinking

- 9 mA

Input Resistance

- On: 300 ohms or less
- Off: 200k ohms or more

Appendix A

8-Channel 24 VDC Input/Output Module Specifications (Continued)

Maximum Input Delay Time

- On to Off: 4 milliseconds
- Off to On: 4 milliseconds

Maximum Load Current Sourcing

- 1 Circuit: 0.5 A
- 4 Circuits: 1.25 A
- 8 Circuits: 2.5 A

Maximum Leakage Current

- 0.1 mA

Maximum Inrush Current

- 3 A for 20 milliseconds

Maximum Output Delay Time

- On to Off: 1 millisecond
- Off to On: 1 millisecond

Common Connection

- 8 input points per common terminal
- 8 output points per common terminal

Polarity

- Common terminal

Isolation Method

- Opto-Isolator

Average Module Current Consumption (Input Side)

- Channel 1: 10 mA + (number of ON inputs x 9 mA) +
  (number of ON outputs x 8 mA)
- Channel 2: 0 mA
- Channel 3: (number of ON inputs x 9 mA)

Appendix A

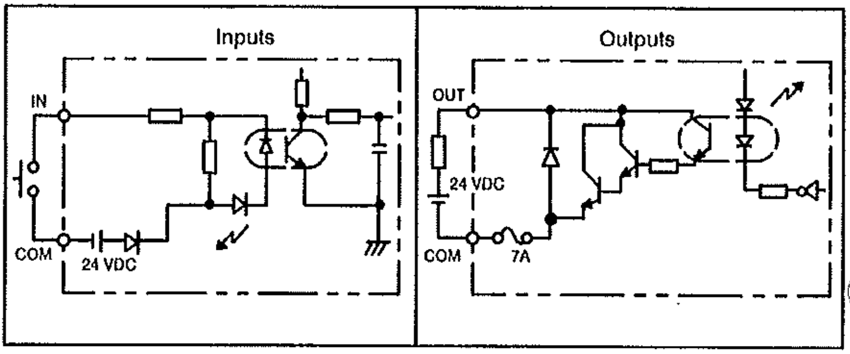
8-Channel 24 VDC Input/Output Module Specifications (Continued)

Average Module Current Consumption (Output Side)

- Channel 1: 10 mA + (number of ON inputs x 9 mA) +
  (number of ON outputs x 8 mA)
- Channel 2: 0 mA
- Channel 3: 0 mA

Agency Approvals

- UL Listed



Circuit Diagram

Appendix A

Counter Module Specifications

Model Number

- M/N 45C982 Counter Module

Ambient Conditions

- Storage Temperature: −4° to 158°F
  −20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Input Pulse Frequency

- 10K Hz maximum

Input Pulse Voltage

- On: 0-2 VDC
- Off: 5-12 VDC

Input Pulse and Marker Pulse Width

- 20 microseconds minimum

Input Impedance

- Approximately 10k ohms

Input and Output Isolation Method

- Opto-Isolator

Input Signals

- A, B, and Marker

Input and Output Polarity

- Common terminal

Appendix A

Counter Module Specifications (Continued)

Two-phase Input Pulses

- Count Up (add): B input lags the A input by 90°
- Count Down (subtract): B input leads the A input by 90°

Counter Range

- 0-9999

Power Available for External Inputs

- 12 VDC (+/-10%), 50 mA maximum current

Output Signal Voltage

- 10-30 VDC

Number of Outputs

- OUT1, OUT2, OUT3, OUT4

Type of Outputs

- Open-collector transistor

Output Load Current

- 0.5 Amp per output, non-fused
- 1.25 Amps total for 4 outputs
- 1 mA minimum

Output Leakage Current

- 0.1 mA maximum

Output Delay Time

- On to Off: 1 millisecond maximum
- Off to On: 1 millisecond maximum

Output On-time Voltage Drop

- 1 VDC, 0.5 Amp maximum

Power Source for Outputs

- 10-30 VDC, 50 mA of current is required

Appendix A

Counter Module Specifications (Continued)

Dielectric Strength

- 500 VDC for 1 minute between the external terminals and the ground terminal

Average Module Current Consumption

- Channel 1: 200 mA
- Channel 2: 0mA
- Channel 3: 160 mA wlten supplying 50 mA to an external input device
  110 mA when not supplying an external input device

Agency Approvals

- UL Listed

Appendix A

Remote I/O Module Specifications

Model Number

- M/N 45C988 Remote Master Module
- M/N 45C989 Remote Target Module

Ambient Conditions

- Storage Temperature: –4° to 158°F
  –20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
  0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Average Module Current Consumption

- Channel 1: +5V, 130 mA (M/N 45C988)
  +5V, 150 mA (M/N 45C989)
- Channel 2: +5V, 20 mA
- Channel 3: +5V, 5 mA

Recommended Remote I/O Cable

- 20 Gauge Shielded Twisted-Pair
- Belden 9463 or equivalent

Remote I/O Refresh Rate

- 5 milliseconds

Total Remote I/O Cable Length

- 1000 feet (300 meters)

Maximum Number of I/O per Remote Network

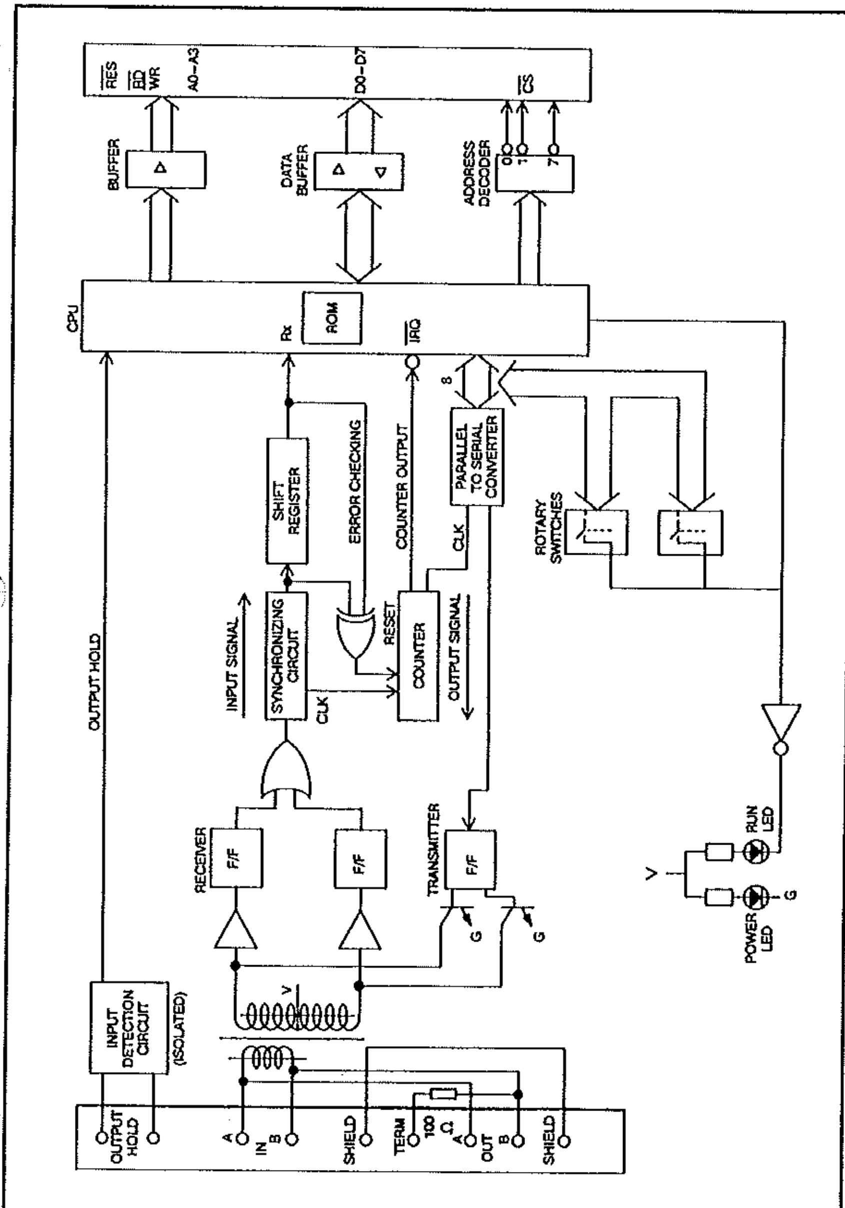
- 128

Agency Approvals

- UL Listed

Appendix A

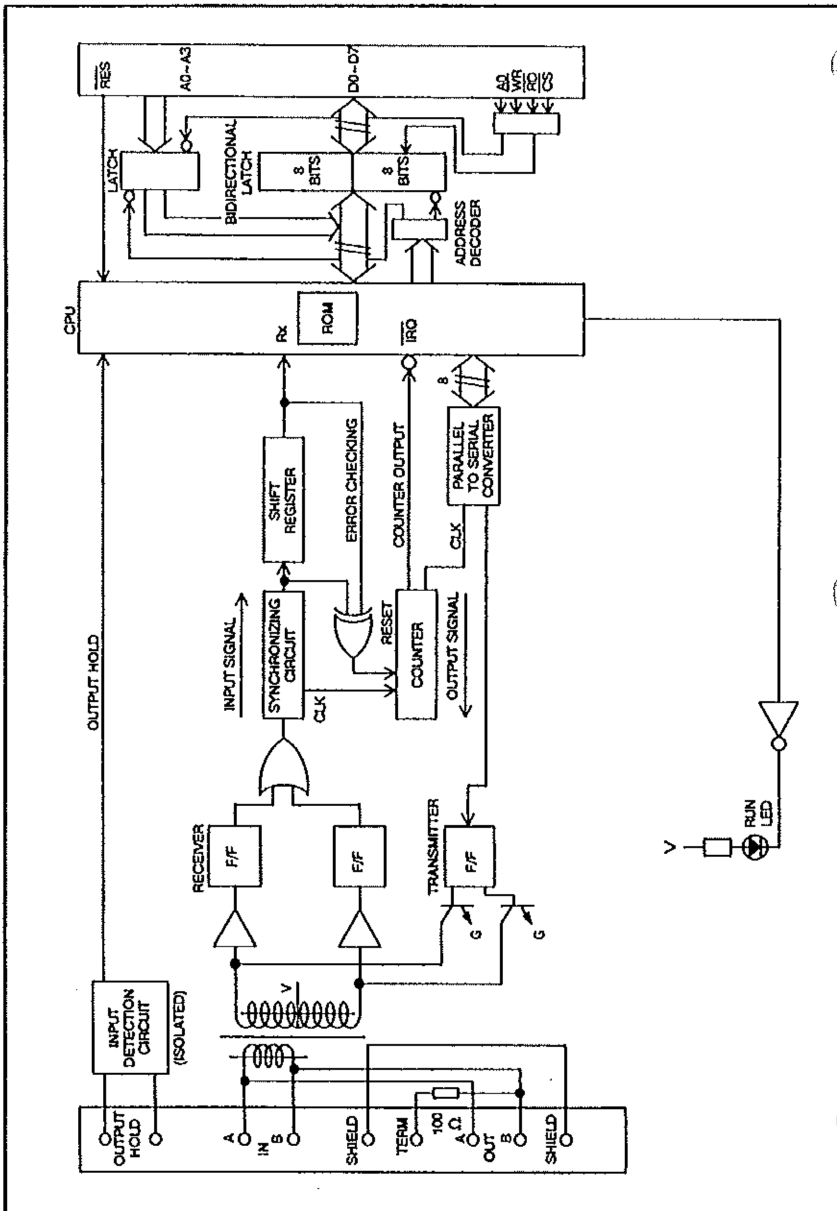
Remote I/O Module Specifications (Continued)



Circuit Diagram (45C989)

Appendix A

Remote I/O Module Specifications (Continued)



Circuit Diagram (45C988)

Appendix A

Network Module Specifications

Model Number

- M/N 45C987 Network Module

Ambient Conditions

- Storage Temperature: −4° to 158°F
−20° to 70°C
- Operating Temperature (at the module): 32° to 131°F
0° to 55°C
- Humidity: 30 to 90% non-condensing

Dimensions

- Height: 6 inches (15 cm)
- Depth: 4.6 inches (11.7 cm)
- Width: 1.4 inches (3.5 cm)
- Weight: 7 ounces (0.2 kilograms)

Average Module Current Consumption

- Channel 1: +5V, 150 mA
- Channel 2: +5V, 20 mA
- Channel 3: +5V, 5 mA

Recommended Network Cable

- 20 Gauge Shielded Twisted-Pair
- Belden 9463 or equivalent

Network Refresh Rate

- 10 milliseconds x # of nodes

Total Network Cable Length

- 1000 feet (300 meters)

Nodes

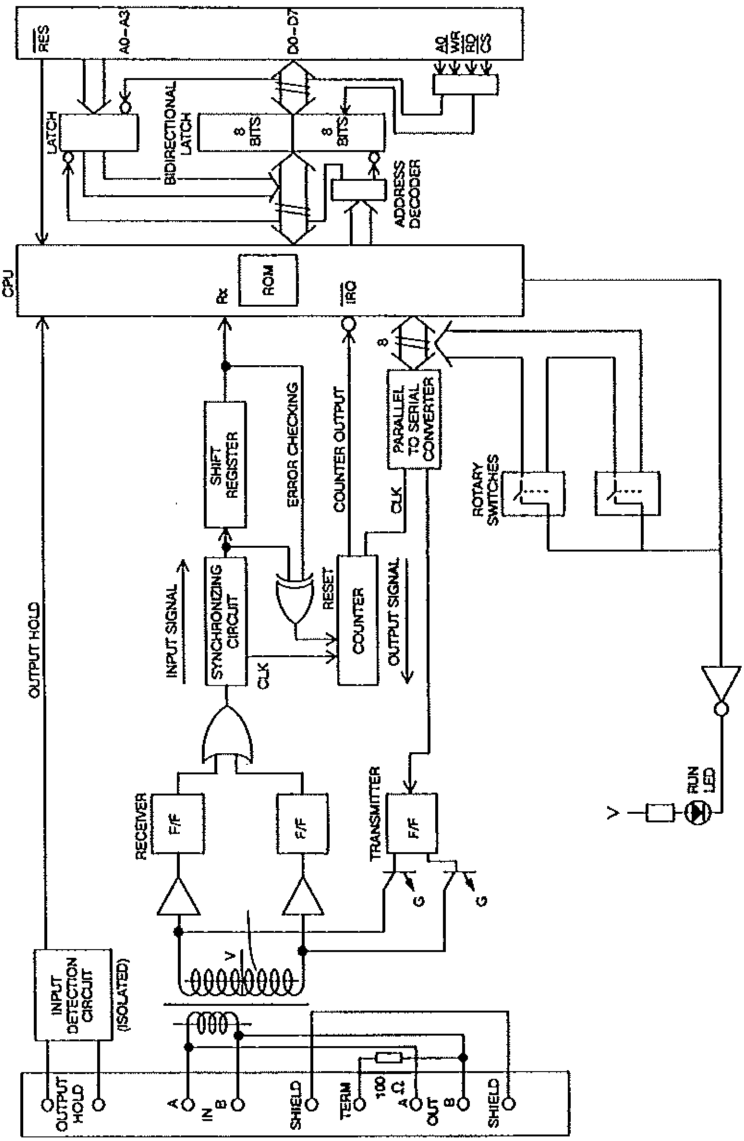
- 1 master
- 7 slaves maximum

Agency Approvals

- UL Listed

Appendix A

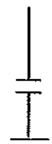
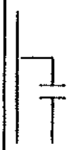
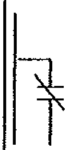
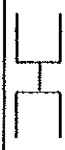
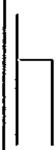
Network Module Specifications (Continued)



Circuit Diagram

Appendix B

Instruction Set/Command Summary

Command	Name	Bit/Byte/Word* Instruction	Words of Memory Used	Function	Symbol	
ORG	Origin	Bit	1	Begin a rung with a N.O. contact.	Start	
STR	Store	Bit	1	Store the AR result and begin a branch with a N.O. contact.	Branch	
AND	And	Bit	1	Connect a N.O. contact in series.	Series	
OR	Or	Bit	1	Connect a N.O. contact in parallel.	Parallel	
ORG NOT	Origin not	Bit	1	Begin a rung with a N.C. contact.	Start	
STR NOT	Store not	Bit	1	Store the AR result and begin a branch with a N.C. contact.	Branch	
AND NOT	And not	Bit	1	Connect a N.C. contact in series.	Series	
OR NOT	Or not	Bit	1	Connect a N.C. contact in parallel.	Parallel	
AND STR	And store	Bit	1	Series combination of the AR and the first storage location.	Parallel-parallel	
OR STR	Or store	Bit	1	Parallel combination of the AR and the first storage location.	Join	

* Word = 2 bytes = 2 consecutive addresses

Appendix B

Instruction Set/Command Summary (Continued)

Command	Name	Bit/Byte/Word* Instruction	Words of Memory Used	Function	Symbol
OUT	Out	Bit	1	Output coil.	( )
OUT NOT	Out Not	Bit	1	Not coil. Result is logically opposite of the OUT command.	(/)
FUN 00	DIF	Bit	1	Leading edge detection (_↑‾).	FUN 00
FUN 01	DFN	Bit	1	Trailing edge detection (‾↓_).	FUN 01
FUN 02	IF	Bit	1	If command. Latch or unlatch a coil from different areas of the program.	FUN 02
FUN 03	IFR	Bit	1	If command with a reset input. Latch or unlatch a coil within a block.	FUN 03
FUN 04	MCS	Bit	1	Master control set.	FUN 04
FUN 05	MCR	Bit	1	Master control reset.	FUN 05
FUN 06	JMP	Bit	1	Begin a jump.	FUN 06
FUN 07	JMP END	Bit	1	End a jump.	FUN 07
FUN 08	AJMP	Bit	2	Begin an addressed jump.	FUN 08
FUN 09	AJMP END	Bit	2	End an addressed jump.	FUN 09

Appendix B

Instruction Set/Command Summary (Continued)

Command	Name	Bit/Byte/Word* Instruction	Words of Memory Used	Function	Symbol
FUN 0.	LOAD C	Word	2	Load a constant word value into the arithmetic register.	FUN 0
FUN 1.	ADD C	Word	2	Add a BCD constant value to the contents of the arithmetic register.	FUN 1
FUN 2.	SUB C	Word	2	Subtract a BCD constant value from the contents of the arithmetic register.	FUN 2
FUN 3.	MUL C	Word	2	Multiply a BCD constant value with the contents of the arithmetic register.	FUN 3
FUN 4.	DIV C	Word	2	Divide the contents of the arithmetic register by a BCD constant word value.	FUN 4
FUN 5.	WAND C	Word	2	AND together a BCD constant word value and the contents of the arithmetic register.	FUN 5
FUN 6.	WOR C	Word	2	OR together a BCD constant word value and the contents of the arithmetic register.	FUN 6
FUN 7.	CMP ≥ C	Word	2	Compare the contents of the arithmetic register with a constant word value (≥) and set the carry bit.	FUN 7
FUN 8.	CMP = C	Word	2	Compare the contents of the arithmetic register with a BCD constant word value (=) and set the carry bit.	FUN 8
FUN 9.	CMP < C	Word	2	Compare the contents of the arithmetic register with a constant word value (<) and set the carry bit.	FUN 9
FUN 10	WLOAD	Word	2	Load a word of data into the arithmetic register.	FUN 10
FUN 11	ADD	Word	2	Add a BCD word of data to the contents of the arithmetic register.	FUN 11

Appendix B

Instruction Set/Command Summary (Continued)

Command	Name	Bit/Byte/Word* Instruction	Words of Memory Used	Function	Symbol
FUN 12	SUB	Word	2	Subtract a BCD word of data from the contents of the arithmetic register.	FUN 12
FUN 13	MUL	Word	2	Multiply a BCD word of data with the contents of the arithmetic register.	FUN 13
FUN 14	DIV	Word	2	Divide the contents of the arithmetic register by a BCD word of data.	FUN 14
FUN 15	WAND	Word	2	AND together a word of data and the contents of the arithmetic register.	FUN 15
FUN 16	WOR	Word	2	OR together a word of data and the contents of the arithmetic register.	FUN 16
FUN 17	CMP	Word	2	Compare the contents of the arithmetic register with a word of data (≥) and set the carry bit.	FUN 17
FUN 18	CMP	Word	2	Compare the contents of the arithmetic register with a word of data (=) and set the carry bit.	FUN 18
FUN 19	CMP	Word	2	Compare the contents of the arithmetic register with a word of data (<) and set the carry bit.	FUN 19
FUN 20	WLOAD B	Bit (16 Addresses)	2	Load bit data into the arithmetic register as word data. (Bit to Word)	FUN 20
FUN 21	WOUT	Word	2	Transfer the contents of the arithmetic register to an output as two bytes.	FUN 21
FUN 22	WOUT B	Bit (16 Addresses)	2	Transfer the word data of the arithmetic register to an output as bit data. (Word to Bit)	FUN 22
FUN 23	OUC	Bit	1	Transfer the status of the carry bit to an output.	FUN 23

Appendix B

Instruction Set/Command Summary (Continued)

Command	Name	Bit/Byte/Word* Instruction	Words of Memory Used	Function	Symbol
FUN 24	BIN – BCD	Word	1	Convert binary data to binary coded decimal data.	FUN 24
FUN 25	BCD – BIN	Word	1	Convert binary coded decimal data to binary data.	FUN 25
FUN 26	SFR L	Bit (16 Addresses)	1	Shift the contents of the arithmetic register one bit to the left.	FUN 26
FUN 27	SFR R	Bit (16 Addresses)	1	Shift the contents of the arithmetic register one bit to the right.	FUN 27
FUN 28	Branch	Bit	1	Mark the beginning of a branch to multiple outputs in the same rung.	FUN 28
FUN 29	Return	Bit	1	Start logic from a branch for additional outputs in a rung.	FUN 29
FUN 40	UDC	Bit (16 Addresses)	1	Up/down counter	FUN 40
FUN 41	NOP	Bit	1	No operation.	FUN 41
FUN 42	CALL	Word	2	Call subroutine.	FUN 42
FUN 43	SB	Word	2	Define subroutine.	FUN 43
FUN 44	RTS	Word	1	Return from subroutine.	FUN 44

Appendix B

Instruction Set/Command Summary (Continued)

Command	Name	Bit/Byte/Word* Instruction	Words of Memory Used	Function	Symbol
FUN 45	LATCH	Bit	1	Latch.	FUN 45
FUN 47	SFR	Bit (16 Addresses)	1	Shift register.	FUN 47
FUN 48	EXTRACT	Word	1	Load data into the AR from the I/O address specified in the ER.	FUN 48
FUN 49	DISTRIB	Word	1	Output data from the AR to the I/O address specified in the ER.	FUN 49
FUN 50	LOAD CB	Byte	2	Load a constant byte value into the arithmetic register.	FUN 50
FUN 51	ADDBNR C	Word	2	Add a binary constant value to the contents of the arithmetic register.	FUN 51
FUN 52	SUBBNR C	Word	2	Subtract a binary constant value from the contents of the arithmetic register.	FUN 52
FUN 53	MULBNR C	Word	2	Multiply a binary constant value with the contents of the arithmetic register.	FUN 53
FUN 54	DIVBNR C	Word	2	Divide the contents of the arithmetic register by a binary constant word value.	FUN 54
FUN 55	BANDI	Byte	2	AND together a binary byte value and the contents of the arithmetic register.	FUN 55
FUN 56	BORI	Byte	2	OR together a binary byte value and the contents of the arithmetic register.	FUN 56

Appendix B

Instruction Set/Command Summary (Continued)

Command	Name	Bit/Byte/Word* Instruction	Words of Memory Used	Function	Symbol
FUN 57	BCPHI	Byte	2	Compare the contents of the arithmetic register with a constant byte value (≥) and set the carry bit.	FUN 57
FUN 58	BCPEI	Byte	2	Compare the contents of the arithmetic register with a constant byte value (=) and set the carry bit.	FUN 58
FUN 59	BCPLI	Byte	2	Compare the contents of the arithmetic register with a constant byte value (≤) and set the carry bit.	FUN 59
FUN 60	BLOAD	Byte	2	Load a byte of data into the arithmetic register.	FUN 60
FUN 61	ADD BNR	Word	2	Add a binary word of data to the contents of the arithmetic register.	FUN 61
FUN 62	SUB BNR	Word	2	Subtract a binary word of data from the contents of the arithmetic register.	FUN 62
FUN 63	MUL BNR	Word	2	Multiply a binary word of data with the contents of the arithmetic register.	FUN 63
FUN 64	DIV BNR	Word	2	Divide the contents of the arithmetic register by a binary word of data.	FUN 64
FUN 66	EXOR	Word	2	Exclusive OR together a word of data and the contents of the arithmetic register.	FUN 66
FUN 71	BOUT	Byte	2	Transfer the contents of AR_L to an output as a byte.	FUN 71
FUN 72	MASK L	Bit	2	Mask the AR the specified number of bits starting at the MSB.	FUN 72

Appendix B

Instruction Set/Command Summary (Continued)

Command	Name	Bit/Byte/Word* Instruction	Words of Memory Used	Function	Symbol
FUN 73	MASK R	Bit	2	Mask the AR the specified number of bits starting at the LSB.	FUN 73
FUN 74	SEG	Byte	1	Convert AR_{LL} into 7-segment display code.	FUN 74
FUN 75	ASC	Byte	1	Convert AR_{LL} into an ASCII code.	FUN 75
FUN 76	ROL	Word	1	Rotate shift register left with carry bit.	FUN 76
FUN 77	ROR	Word	1	Rotate shift register right with carry bit.	FUN 77
FUN 78	ENCOD	Word	1	Encode the contents of the arithmetic register.	FUN 78
FUN 79	DECOD	Word	1	Decode the contents of the arithmetic register.	FUN 79
FUN 80	SWAP	Word	1	Exchange AR_H and AR_L.	FUN 80
FUN 81	BSWAP	Byte	1	Exchange bits 0–3 and 4–7 of AR_L.	FUN 81
FUN 82	XCG	Word	1	Exchange AR data and ER data.	FUN 82
FUN 83	CLC	Bit	1	Clear the carry bit.	FUN 83

Appendix B

Instruction Set/Command Summary (Continued)

Command	Name	Bit/Byte/Word* Instruction	Words of Memory Used	Function	Symbol
FUN 84	SEC	Bit	1	Set the carry bit.	FUN 84
FUN 85	WNOT	Word	1	Logical NOT condition of AR.	FUN 85
FUN 88	SET	Bit	1	Set command to enable an output.	FUN 88
FUN 89	RES	Bit	1	Reset command to disable an output.	FUN 89
FUN 91	REF X	Bit	1	Update an external input.	FUN 91
FUN 92	REF Y	Bit	1	Update an internal input.	FUN 92
FUN 93	INT	Word	2	Interrupt.	FUN 93
FUN 94	RTI	Word	1	Return from interrupt.	FUN 94
FUN 98	STA	N/A	1	Start program execution.	
FUN 99	END	N/A	1	End of program.	
T/C	Timer/ Counter	Bit	1	On-delay timer or up-counter.	T, C

Appendix C

Execution Times

Execution time is in micro-seconds. (µs)

ORG = 1.5
ORG NOT = 1.5
AND = 1.5
AND NOT = 1.5
OR = 1.5

OR NOT = 1.5
STR = 1.5
STR NOT = 1.5
AND STR = 1.5
OR STR = 1.5

OUT = 1.5
OUT NOT = 1.5
OUT T (0.1 s) = 85.5
OUT T (10 ms)= 94.5
OUT NOT T = 1.5

OUT C = 86
FUN0. = 8.5
FUN1. = 19.5
FUN2. = 457
FUN3. = 523.5

FUN4. = 1445
FUN5. = 11.5
FUN6. = 11.5
FUN7. = 13
FUN8. = 13

FUN9. = 13
FUN00 = 13
FUN01 = 13
FUN02 = 1.5
FUN03 = 1.5

FUN04 = 15
FUN05 = 13.5
FUN06 = 1.5
FUN07 = 1.5
FUN08 = 13

FUN09 = 3
FUN10 = 14.5
FUN11 = 26
FUN12 = 463
FUN13 = 529.5

FUN14 = 1451
FUN15 = 18
FUN16 = 18
FUN17 = 19
FUN18 = 19

FUN19 = 19
FUN20 = 187
FUN21 = 14.5
FUN22 = 247.5
FUN23 = 8.5

FUN24 = 279
FUN25 = 97.5
FUN26 = 18.5
FUN27 = 18.5
FUN28 = 1.5

FUN29 = 1.5
FUN40 = 427.5
FUN41 = 1.5
FUN42 = 51
FUN43 = – –

FUN44 = 41.5
FUN45 = 23
FUN47 = 162.5
FUN48 = 1153.5
FUN49 = 1150

FUN50 = 7.5
FUN51 = 15
FUN52 = 15
FUN53 = 106.5
FUN54 = 1028

FUN55 = 9
FUN56 = 9
FUN57 = 12
FUN58 = 12
FUN59 = 12

FUN60 = 13.5
FUN61 = 19.5
FUN62 = 19.5
FUN63 = 112.5
FUN64 = 1034

FUN66 = 18
FUN71 = 13.5
FUN72 = 27
FUN73 = 27
FUN74 = 11

FUN75 = 11
FUN76 = 17
FUN77 = 17
FUN78 = 107.5
FUN79 = 19

FUN80 = 8
FUN81 = 9
FUN82 = 11
FUN83 = 6
FUN84 = 6

FUN85 = 9
FUN88 = 1.5
FUN89 = 1.5
FUN91 = 21.5
FUN92 = 15

FUN93 = – –
FUN94 = 68
FUN98 = 1.5
FUN99 = 2300

Appendix D

Command Cross-reference

FUN0. = Constant Value Word Load Command = Page 6-32
FUN1. = Constant Value BCD Addition Command = Page 6-34
FUN2. = Constant Value BCD Subtraction Command = Page 6-36
FUN3. = Constant Value BCD Multiplication Command = Page 6-38
FUN4. = Constant Value BCD Division Command = Page 6-40
FUN5. = Constant Value BCD Word AND Command = Page 6-42
FUN6. = Constant Value BCD Word OR Command = Page 6-44
FUN7. = Constant Value Word Compare (≥) Command = Page 6-46
FUN8. = Constant Value Word Compare (=) Command = Page 6-46
FUN9. = Constant Value Word Compare (<) Command = Page 6-46
FUN00 = Leading Edge Detection Command = Page 6-10
FUN01 = Trailing Edge Detection Command = Page 6-10
FUN02 = If and If Reset Commands = Page 6-12
FUN03 = If and If Reset Commands = Page 6-12
FUN04 = Master Control Set Command = Page 6-19
FUN05 = Master Control Reset Command = Page 6-19
FUN06 = Jump Command = Page 6-21
FUN07 = Jump End Command = Page 6-21
FUN08 = Addressed Jump Command = Page 6-21
FUN09 = Addressed Jump End Command = Page 6-21
FUN10 = Word Load (Two Bytes) = Page 6-50
FUN11 = BCD Addition Command = Page 6-54
FUN12 = BCD Subtraction Command = Page 6-55
FUN13 = BCD Multiplication Command = Page 6-56
FUN14 = BCD Division Command = Page 6-57
FUN15 = Word AND Command = Page 6-58
FUN16 = Word OR Command = Page 6-59
FUN17 = Word Compare (≥) Command = Page 6-61
FUN18 = Word Compare (=) Command = Page 6-61
FUN19 = Word Compare (<) Command = Page 6-61
FUN20 = Word Load (Sixteen Bits) Command = Page 6-51
FUN21 = Word Out (Two Bytes) Command = Page 6-63
FUN22 = Word Out (Sixteen Bits) Command = Page 6-65
FUN23 = Output Carry Bit Command = Page 6-68
FUN24 = Binary Coded Decimal Command = Page 6-68
FUN25 = Binary Command = Page 6-68
FUN26 = Left Shift Register Command = Page 6-72

Appendix D

Command Cross-reference (Continued)

FUN27 = Right Shift Register Command = Page 6-72
FUN28 = Branch Command = Page 6-23
FUN29 = Return Command = Page 6-23
FUN40 = Up/Down Counter Command = Page 6-74
FUN41 = No Operation Command = Page 6-26
FUN42 = Call Subroutine Command = Page 6-30
FUN43 = Subroutine Command = Page 6-30
FUN44 = Return from Subroutine Command = Page 6-30
FUN45 = Latch Command = Page 6-26
FUN47 = Shift Register Command = Page 6-70
FUN48 = Load Data into the Arithmetic Register = Page 6-83
FUN49 = Output Data from the Arithmetic Register = Page 6-83
FUN50 = Constant Value Byte Load Command = Page 6-32
FUN51 = Constant Value Binary Addition Command = Page 6-34
FUN52 = Constant Value Binary Subtraction Command = Page 6-36
FUN53 = Constant Value Binary Multiplication Command = Page 6-38
FUN54 = Constant Value Binary Division Command = Page 6-40
FUN55 = Constant Value Binary Byte AND Command = Page 6-42
FUN56 = Constant Value Binary Byte OR Command = Page 6-44
FUN57 = Constant Value Byte (≥) Compare Command = Page 6-46
FUN58 = Constant Value Byte (=) Compare Command Page 6-46
FUN59 = Constant Value Byte (<) Compare Command Page 6-46
FUN60 = Byte Load Command = Page 6-52
FUN61 = Binary Addition Command = Page 6-54
FUN62 = Binary Subtraction Command = Page 6-55
FUN63 = Binary Multiplication Command = Page 6-56
FUN64 = Binary Division Command = Page 6-57
FUN66 = Word Exclusive OR Command= Page 6-60
FUN71 = Byte Out Command = Page 6-67
FUN72 = Mask the Arithmetic Register from the Most Significant Bit = Page 6-80
FUN73 = Mask the Arithmetic Register from the Least Significant Bit = Page 6-80
FUN74 = 7-Segment Conversion Command = Page 6-77
FUN75 = ASCII Conversion Command = Page 6-78
FUN76 = Rotate Shift Register Left Command = Page 6-73
FUN77 = Rotate Shift Register Right Command = Page 6-73

Appendix D

Command Cross-reference (Continued)

FUN78 = Arithmetic Register Encode Command = Page 6-79

FUN79 = Arithmetic Register Decode Command = Page 6-79

FUN80 = Exchange Arithmetic Register's Upper Byte and Lower Byte = Page 6-81

FUN81 = Exchange the Nibbles of the Arithmetic Register's Lower Byte = Page 6-81

FUN82 = Exchange Arithmetic Register Data and Extension Register Data = Page 6-81

FUN83 = Clear Carry Bit Command = Page 6-68

FUN84 = Set Carry Bit Command = Page 6-68

FUN85 = Invert Arithmetic Register Command = Page 6-76

FUN88 = Set Command = Page 6-12

FUN89 = Reset Command = Page 6-12

FUN91 = Update Specified External Inputs Command = Page 6-27

FUN92 = Update Specified External Outputs Command = Page 6-27

FUN93 = Interrupt Command = Page 6-29

FUN94 = Return from Interrupt Command = Page 6-29

FUN98 = Start Command = Page 6-9

FUN99 = End Command = Page 6-9

For additional information
1 Allen-Bradley Drive
Mayfield Heights, Ohio 44124 USA
Tel: +1 440 646 3434
www.rockwellautomation.com/support

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters
Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication J-3809-2 - August 1993