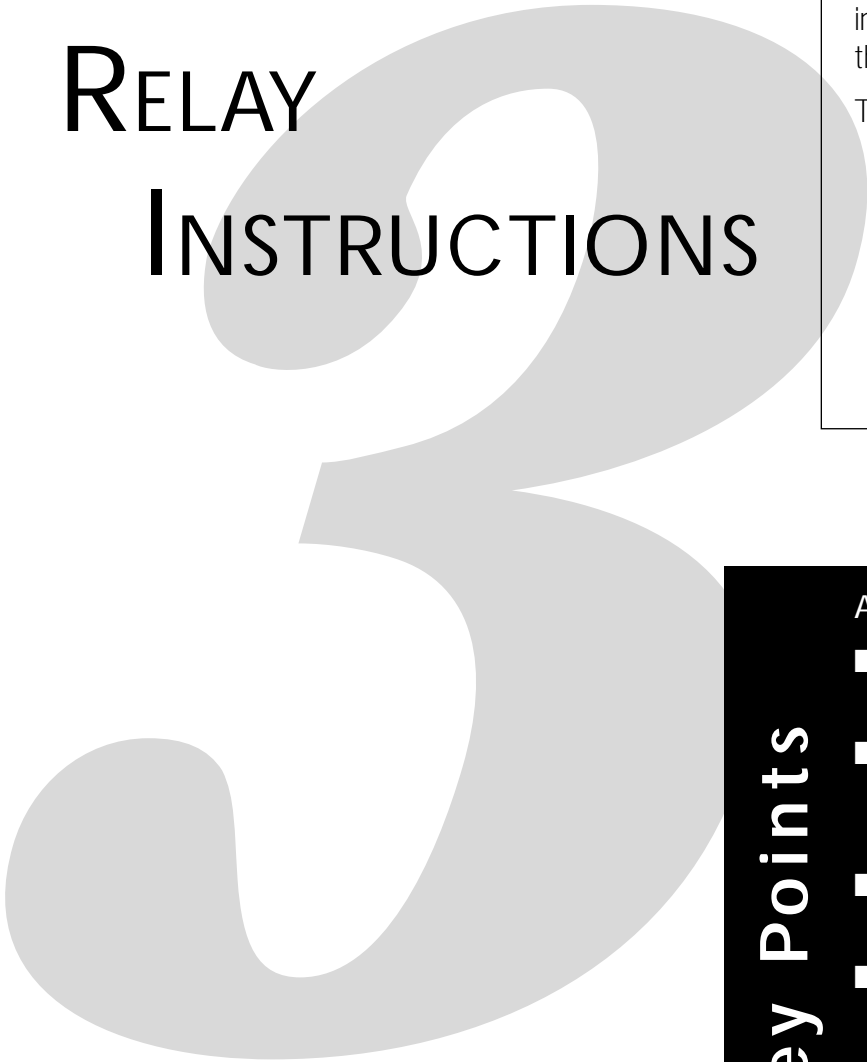


# BASIC RELAY INSTRUCTIONS



So far, you have learned about the components of the MicroLogix 1000 PLC, including the CPU, the memory system, the power supply, and the input/output system. This module teaches you about the control program that links all of these components together.

There are five sections:

1. Ladder programming
2. Input/contact instructions
3. Output/coil instructions
4. Special programming issues
5. Basic relay programming examples

## Key Points

After finishing this module, you will:

- understand the basics of ladder programming and the types of relay instructions available in the MicroLogix 1000
- know the two types of contact instructions used to program a MicroLogix 1000, how they operate, and their reference addresses
- know the four basic output instructions commonly used with the MicroLogix 1000
- understand key points concerning the ladder scan evaluation, the programming of normally closed input devices, and safety device programming
- be able to apply techniques learned from specific programming examples

## 3-1 Ladder Programming

This section covers the fundamentals of ladder programming. A MicroLogix 1000 uses the control program stored in its CPU to analyze input data and determine the corresponding output action. This control program is entered and stored in the CPU as ladder diagrams. At the end of this section, you will know:

- the basics of ladder programming
- the two types of relay instructions available in the MicroLogix 1000

### Ladder Diagram Basics

Ladder diagrams and electromechanical circuits function in much the same way. In an electromechanical circuit, one side of the circuit is connected to the incoming L1 power line, while the other side is connected to the L2 return power line. The input and output devices are wired between the two power lines. In the circuit shown in Figure 3-1a, when the limit switch input device closes, it completes the electrical path between the incoming and return power lines. This turns on the pilot light. When the limit switch opens, it breaks continuity in the circuit, turning off the pilot light.

The ladder diagram equivalent of a hardwired circuit also shows the devices connected between the L1 and L2 lines. However, in a ladder diagram, the L1 and L2 lines are represented by the left and right rails, respectively. Also, in a ladder diagram, the inputs and outputs are represented by contacts and coils. Thus, when the limit switch in the ladder circuit shown in Figure 3-1b closes, the LS contact energizes, providing power to the PL coil and turning on the pilot light.

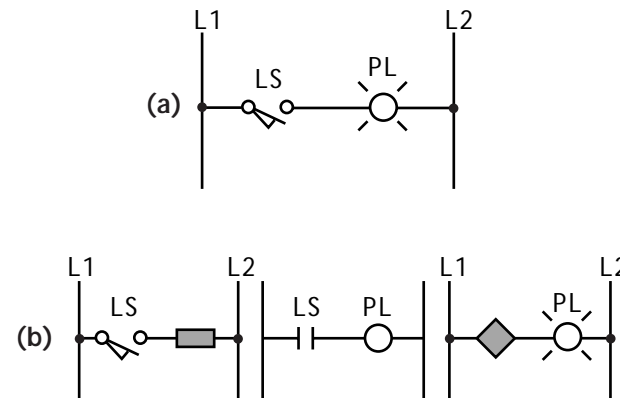
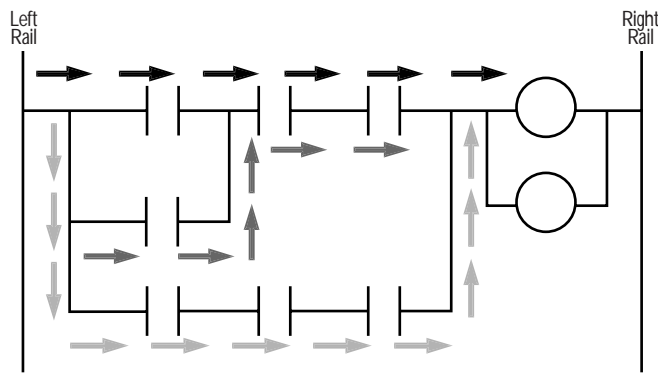
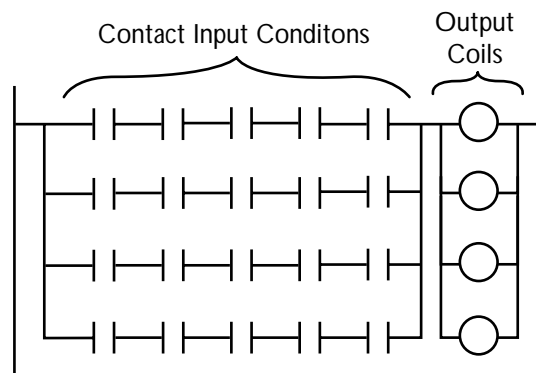


Figure 3-1. (a) An electromechanical circuit and (b) its equivalent ladder diagram.



**Figure 3-2.** A ladder rung with arrows indicating the flow of power through the rung.



**Figure 3-3.** A ladder rung matrix.

**Contacts and Coils.** In a ladder diagram, **contact instructions** are used to reference input devices and **coil instructions** are used to reference output devices. A contact examines the state of its corresponding input field device. It then either energizes or de-energizes, depending on whether the input device is ON or OFF, respectively.

Conversely, a coil energizes or de-energizes according to the status of the contacts in its ladder rung. When a coil energizes, it turns its associated output device ON. When a coil de-energizes, it turns its associated output device OFF.

**Ladder Rung Structure.** In addition to contacts and coils, there are two other terms used when describing a ladder program:

- rung
- ladder rung matrix

**Rung.** A **rung** is a set of programmed instructions that forms a complete electrical path between the left and right rails of a ladder diagram (see Figure 3-2). A MicroLogix's ladder rung can have many contacts and coils in many branches. Since a rung can have many branches, it may provide several different complete paths from the left to the right rail.

Note that a MicroLogix will not allow power to flow from right to left. Power must always flow from left to right through a ladder rung. Job Aid 3-1 explains how to program around reverse power flow in a MicroLogix 1000.

**Ladder Rung Matrix.** The **ladder rung matrix** (see Figure 3-3) determines the maximum number of branches that can be used to program a ladder rung. In a MicroLogix 1000, a rung can have a maximum of 75 branches. The ladder rung matrix also defines all of the possible locations for a contact or coil instruction in a rung.

## Relay Instructions

**Relay instructions** form the category of programming instructions that deals with the simple energizing and de-energizing of inputs and outputs. Contacts and coils fall into this category.

There are two kinds of relay contact input instructions:

- examine-if-closed
- examine-if-open

There are four kinds of relay coil output instructions:

- simple output
- internal output
- latch/unlatch output
- one-shot rising instructions

Other instructions, in addition to simple relay, are available in the MicroLogix 1000. These instructions, which include the following, will be discussed in Modules 4 and 5:

- *timing instructions*: timer ON-delay, timer OFF-delay, and retentive timer
- *counting instructions*: count up, count down, and reset
- *data-handling instructions*: convert-from-BCD, convert-to-BCD, move, and masked move
- *comparison instructions*: equal, masked comparison for equal, and limit test
- *flow control instructions*: jump, jump to subroutine, master control reset, temporary end, suspend, immediate input with mask, and immediate output with mask



## Examine-If-Open Contacts

The **examine-if-open contact instruction**, also called a *normally closed instruction*, is represented by the symbol shown in Figure 3-6. It is abbreviated as XIO. An XIO instruction works the opposite of an XIC instruction. An XIO contact checks the status of its reference address for an OFF, or open, condition. If the reference address is OFF, then the contact remains closed, providing continuity to the ladder rung.

Figure 3-7 illustrates how an XIO contact instruction works. The limit switch in this diagram is connected to terminal 1 of the MicroLogix's input interface, which means that it has an address of I:0.0/1. The XIO instruction in the ladder diagram references this address. When the limit switch is open (see Figure 3-7a), bit 1 of the input file will contain a 0. When the XIO contact detects the 0, the contact will stay in its normally closed condition, resulting in continuity to the rung. When the limit switch is closed, the opposite will occur (see Figure 3-7b). The XIO contact will detect a 1 in the address bit, which will cause the contact to open and break continuity to the rung.

The RSLogix software can be used to monitor the status of the contacts in a ladder program. If the software shows a contact as highlighted, then the contact is providing power continuity. If a contact is unhighlighted, then it is not providing continuity.

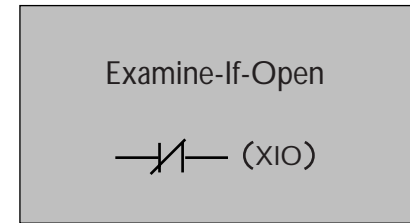


Figure 3-6. An examine-if open instruction.

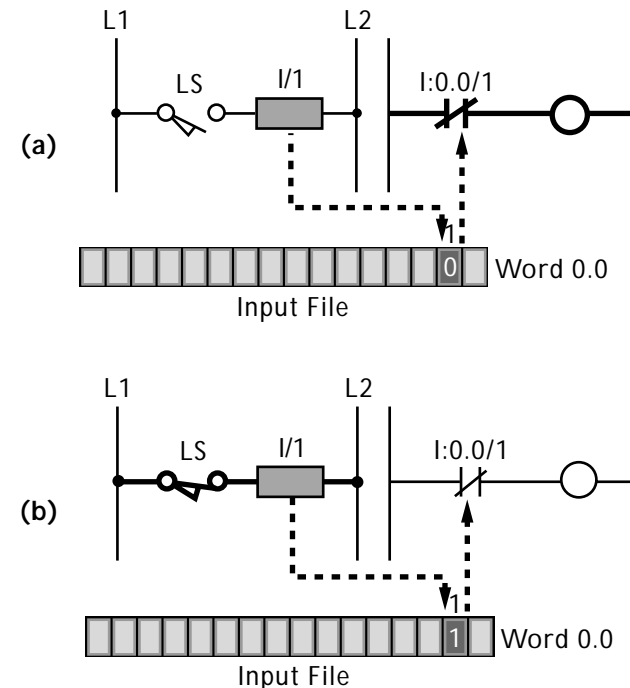


Figure 3-7. (a) When an XIO instruction detects a 0 in its address bit, it closes, resulting in continuity to the ladder rung. (b) When an XIO instruction detects a 1 in its address bit, it opens, breaking continuity in the rung.

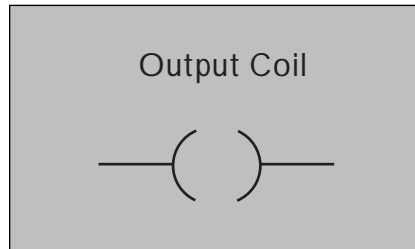


Figure 3-8. A standard output coil instruction.

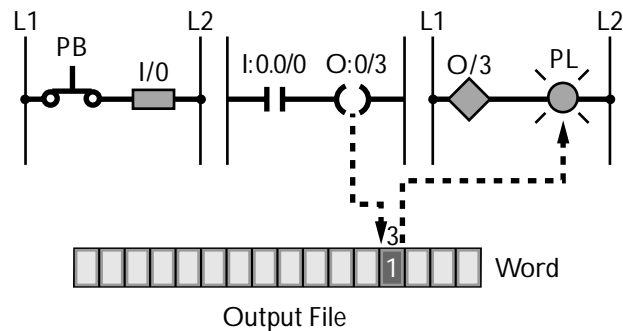


Figure 3-9. When coil O:0/3 turns on, its associated output device will turn on.

### 3-3 Output/Coil Instructions

A coil instruction is a ladder instruction that references an output device. Thus, it represents the output action of a ladder rung. At the end of this section, you will know the operation of the four basic coil instructions used with the MicroLogix 1000:

- standard output coils
- internal output coils
- latch/unlatch output coils
- one-shot rising instructions

#### Standard Output Coils

A **standard output coil instruction**, which is also called an *output energize instruction*, is represented by the symbol shown in Figure 3-8. If the coil's ladder rung has continuity, then the coil will energize, turning on the output device that shares its reference address. This reference address is located in the output file of the data file section.

Figure 3-9 shows a push button driving an output coil that references a pilot light output. When the push button closes, its XIC contact will energize, providing continuity to the ladder rung and causing the output coil to turn ON. When the output coil turns ON, the MicroLogix will store a 1 in the coil's reference address (O:0/3) to indicate that the coil is ON. The PLC will then turn on the output device that shares this address—in this case, the pilot light.

An output coil can be referenced by a contact instruction. Therefore, an output coil can drive a real field output device while simultaneously providing the input logic to another rung. The contacts used to reference an output coil can be either XIC or XIO instructions.

### Internal Output Coils

An **internal output coil instruction** looks like a regular output coil. It is even represented by the same symbol. The functional difference between the two is that a standard output coil references an actual field output device wired to the PLC, while an internal output coil does not. Thus, instead of having a reference address located in the output file, an internal output coil has a reference address located in the binary file (file 3). Internal coils are used with contacts to implement interlocking in a ladder program. They are also used to store bit information that does not directly impact the outside world.

Aside from being mapped to different files, an internal output coil and a regular coil work almost the same way (see Figure 3-10). If its rung has continuity, an internal output will turn on and store a 1 in its reference address bit. Since an internal output coil does not map a real output device, no field output device will turn on. However, any contact that shares the internal coil's address will turn on or off with the coil.

### Latch/Unlatch Output Coils

A **latch/unlatch output coil instruction**, which is actually composed of two separate coil instructions, is represented by the symbols shown in Figure 3-11. These coil instructions are always used together in a ladder program and always share the same address.

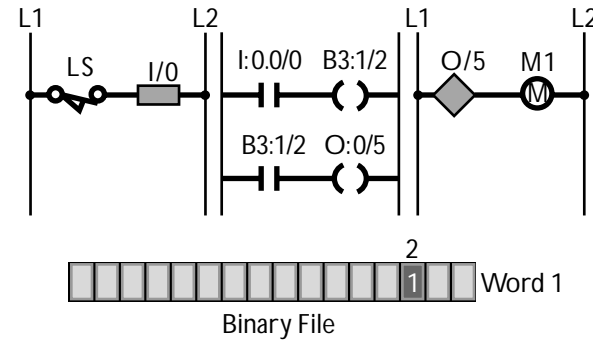


Figure 3-10. An internal output coil (B3:1/2) referenced by an XIC instruction.

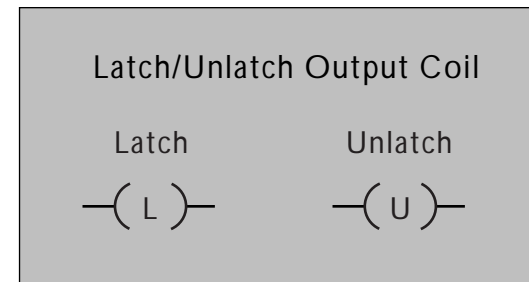


Figure 3-11. The latch and unlatch output coil symbols.

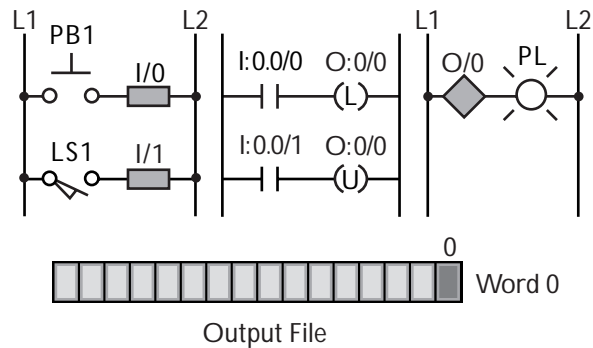


Figure 3-12. A latch/unlatch ladder circuit.

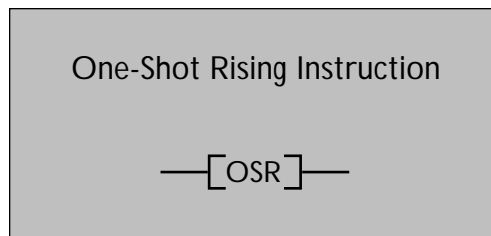


Figure 3-13. A one-shot rising instruction.

The function of a latch coil is to turn on and stay on, even if its input condition turns off. The function of an unlatch coil is to turn off a latched output. A latch coil instruction is abbreviated as OTL, and an unlatch coil is abbreviated as OTU.

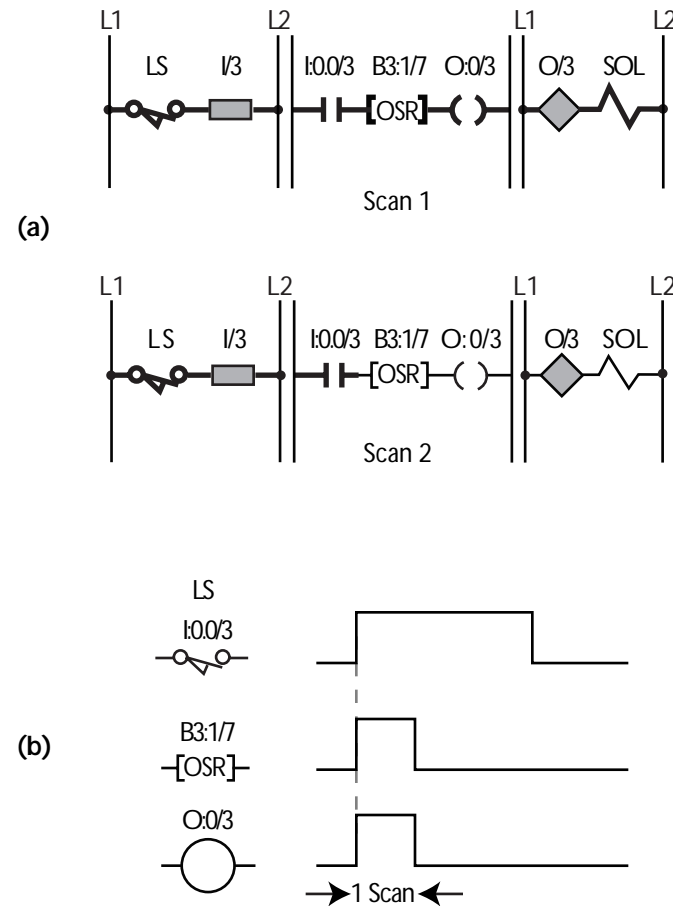
Figure 3-12 illustrates a latch/unlatch ladder circuit. When push button PB1 in the first rung is pressed, the latch output coil will turn on, causing pilot light PL to turn on. However, if the push button is released, the latch output coil and pilot light will remain on even though the first rung no longer has power continuity. When LS1 closes, the unlatch output coil in the second rung will be energized. This will change the status of address bit O:0/0 from a 1 to a 0. As a result, the latch output coil and pilot light will turn off. An unlatch coil instruction is the only way to turn off a latched output coil. Note that the output cannot be latched again until the unlatch coil instruction turns off.

### One-Shot Rising Instructions

A **one-shot rising instruction** is represented by the symbol shown in Figure 3-13. A one-shot rising instruction is not a coil instruction, but rather, a contact instruction. It is covered in this section, however, because it is used to control the behavior of an output coil.

A one-shot rising instruction is used to energize an output coil for only one scan. It is usually the last contact in a rung, located just before the output coil. A one-shot instruction's reference address bit can be located in either the binary file (file 3) or the integer file (file 7). Its address cannot be shared by another contact or coil, and it cannot correspond to a real input or output device. Also, the MicroLogix allows only one one-shot rising instruction per rung.

When the logic before a one-shot rising instruction becomes energized (see Figure 3-14), the one-shot rising instruction will be energized. As a result, the output coil and output device will turn on. A one-shot rising instruction, however, will only stay energized for one scan, even if its driving logic remains energized. Thus, after one scan, the one-shot rising instruction, the output coil, and the associated field output device will turn off, even if the rest of the rung still has continuity. A one-shot rising instruction is very useful for resetting output signals in one scan—for example, for unlatching an output in one scan so that it will be ready to latch again in the next scan.



**Figure 3-14.** (a) Ladder diagrams showing a one-shot rising instruction, which will be energized for only one scan even if its input logic remains energized. (b) The timing diagram for a one-shot rising instruction.

### 3-4 Special Programming Issues

This section covers special issues you may encounter while programming a MicroLogix 1000 controller. At the end of this section you will know the techniques associated with three special programming issues:

- ladder scan evaluation
- the programming of normally closed input devices
- safety device programming

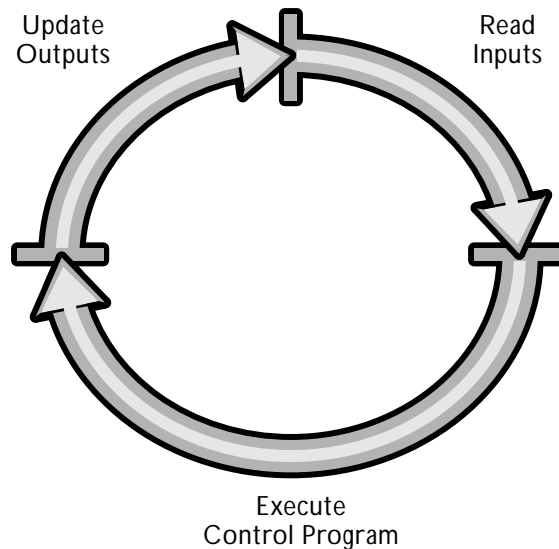
#### Ladder Scan Evaluation

The way the controller evaluates the ladder program determines the order in which the programmed instructions will be executed. Thus, ladder scan evaluation can have a profound effect on the way the PLC and its associated machine or process operate.

A PLC's scan consists of three activities (see Figure 3-15):

1. reading the inputs
2. executing the control program
3. updating the outputs

Once the PLC reads the status of the input devices, it begins to execute the ladder control program. It starts its execution at the top left corner of the ladder program and proceeds from left to right down the control program until it reaches the end of the last ladder rung. The PLC evaluates the ladder program once per scan. After the control program has been executed, the controller updates the status of the outputs according to the results of the control program.



**Figure 3-15.** The three parts of a PLC's scan: reading the inputs, executing the control program, and updating the outputs.

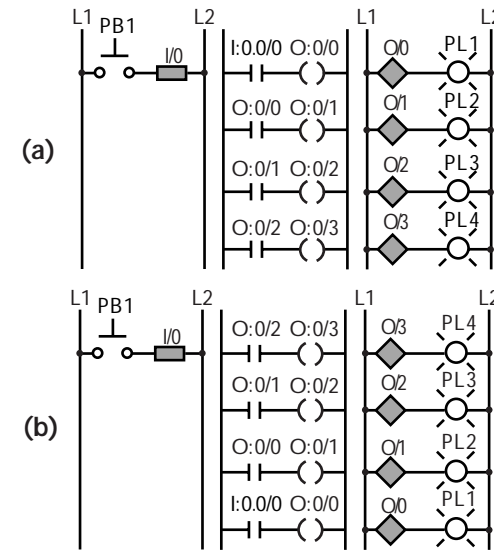
When developing a control program for a MicroLogix 1000, you must pay close attention to the order in which the contacts and coils are programmed. Arranging the same coils and contacts in a different order will produce different results. Figure 3-16 shows two ladder programs containing the same elements but programmed differently. The first program will require only one scan to turn on all of the outputs. The second program, however, will require four scans before all the pilot lights will be on.

### Programming Normally Closed Devices

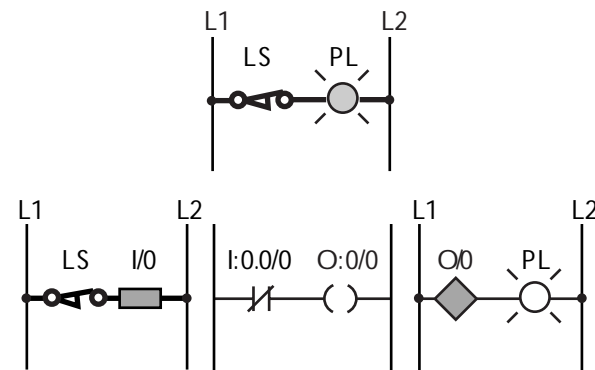
When a normally open input device is connected to a PLC, it is programmed with an XIC contact instruction for the device to act like a normally open one in the ladder program. Consequently, the contact will energize if the input device closes and de-energize if the device opens.

Normally closed devices are programmed differently. Instead of programming a normally closed device with an XIO contact, you must program it with an XIC contact instead. This is necessary for the device to function like a normally closed one in the PLC program. Following are examples of a normally closed device programmed with XIO and XIC contacts to explain why.

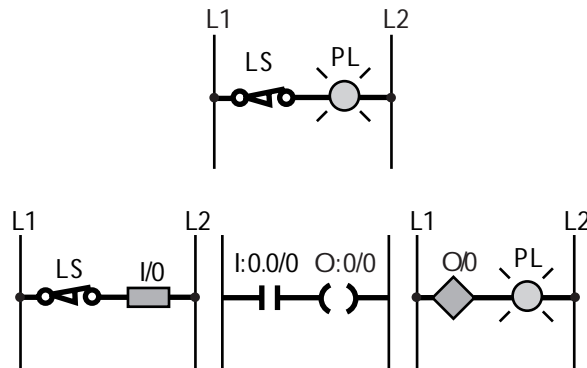
**XIO Contact.** If an XIO contact is used to program a normally closed input device, the ladder diagram will look like Figure 3-17. Logically, this is the way you would expect a normally closed device to be programmed to function properly in a ladder diagram. A closer inspection, however, reveals this to be incorrect. In the figure, the limit switch normally provides a voltage signal to the PLC in the same manner as it would in an electromechanical circuit, meaning that the pilot light should be on. However, in the PLC program, the XIO contact looks for the input device to be open, or OFF, in order to provide continuity to the output. Since the limit switch is closed, the XIO contact



**Figure 3-16.** Ladder programs in which (a) all of the outputs will turn on after one scan and (b) all of the outputs will be on after four scans.



**Figure 3-17.** A normally closed input device programmed with an XIO contact, along with its equivalent hardwired circuit.



**Figure 3-18.** A normally closed input device programmed with an XIC contact, along with its equivalent hardwired circuit.

will not be energized. Accordingly, the pilot light will be off. Under these conditions, the limit switch must open for the XIO contact to be energized and the pilot light to turn on. This is contrary to the way the circuit should work.

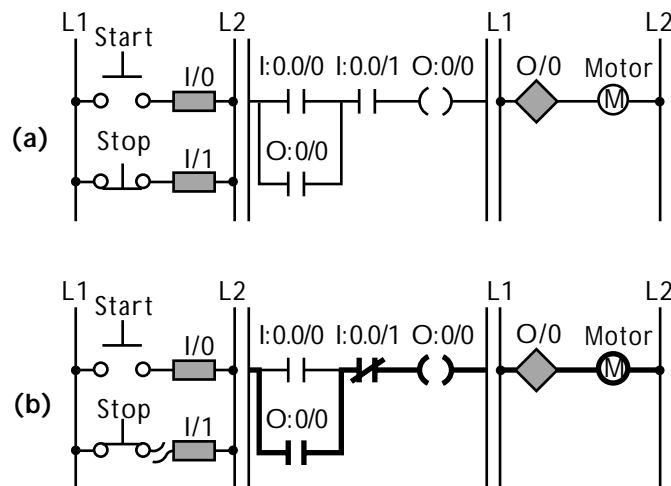
**XIC Contact.** Figure 3-18 shows the circuit from Figure 3-17 programmed with an XIC contact. As in the previous ladder circuit, the normally closed limit switch provides a voltage to the PLC. However, in this program, the XIC contact examines the limit switch for a closed condition in order to provide continuity to the rung. Since the limit switch is closed, the contact will be energized. In turn, the output coil and pilot light will be energized. If the limit switch opens, the pilot light will turn off. Thus, the circuit works correctly.

## Safety Device Programming

When connecting safety devices to PLCs, there is one cardinal rule to remember: never use a normally open device programmed with an XIO contact instruction. The following explains why.

To implement the safety stop circuit shown in Figure 3-19, you must use a normally closed input device programmed with an XIC instruction. In this configuration, when the start push button is pressed, the motor will turn on because both of the XIC contacts will detect that their inputs are closed. When the start button is released, the motor will stay on because contact O:0/0 seals the circuit. When the stop push button is pressed, the XIC instruction will detect that the stop button is open and turn off. This will turn off the motor circuit.

Moreover, if the motor in this circuit is sealed on and the connection from the stop push button breaks, the XIC instruction that references the stop push button will detect an OFF, or open, condition. As a result, it will turn off the motor automatically. This creates a fail-safe circuit that is ideal for safety situations.



**Figure 3-19.** (a) A fail-safe safety stop circuit using a normally closed device programmed with an XIC instruction. (b) A safety stop circuit using a normally open device programmed with an XIO instruction. In this circuit, once the motor is on, the motor cannot be turned off if the safety stop button malfunctions.

### 3-5 Basic Relay Programming Examples

This section includes practical examples to help you apply what you have learned to two specific types of control circuits that are often used in MicroLogix 1000 applications. At the end of this section, you will know about the I/O device wiring, addressing, power wiring, and ladder programming of:

- simple motor control circuits
- forward/reverse motor circuits

#### Simple Motor Control Circuit

Figure 3-20 shows the hardwired diagram for a simple start/stop motor circuit. In this circuit, the motor turns on when the start push button closes. The auxiliary contact in line 2 seals the circuit so that the motor remains on when the start push button is released. When the stop push button is pressed, the motor turns off. The overload contacts can also turn the motor off in the event of an overload situation.

**PLC Requirements.** To convert this circuit into a PLC circuit, you could use a PLC with the following specifications:

- 32 I/O
- 24-volt DC power supply
- 24-volt DC input interface
- relay/transistor output interface

The start and stop input devices will be connected to the 24-volt DC input interface. The motor starter, which has a 24-volt DC coil, will be connected to one of the transistor outputs.

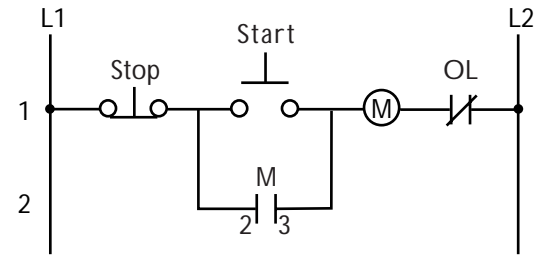
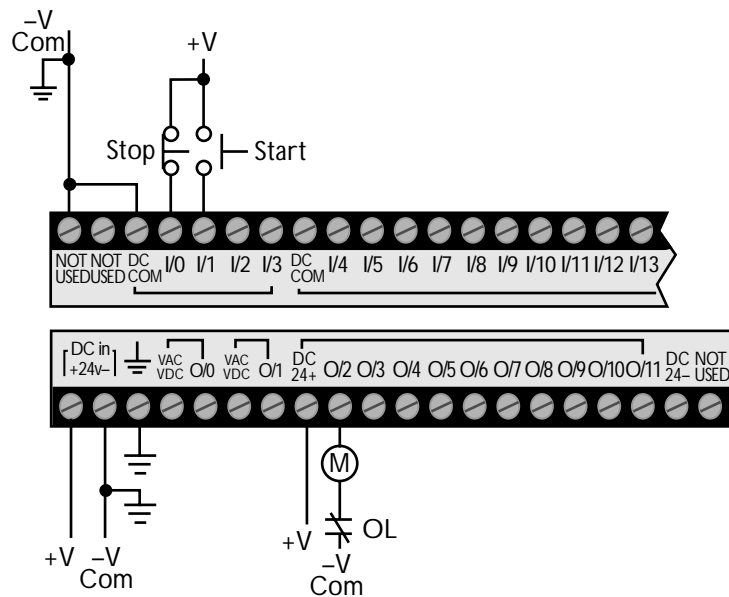
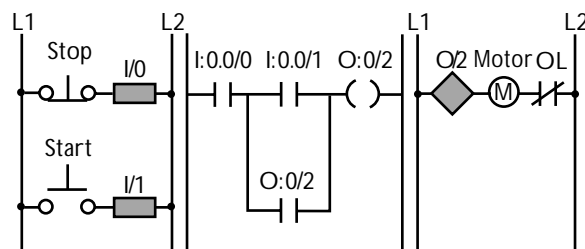


Figure 3-20. Hardwired diagram for a simple start/stop motor circuit.



**Figure 3-21.** The I/O device and power wiring for the start/stop motor circuit. The +V supply connection shown occurs after the MCR power contact, which controls the power to the I/O devices.



**Figure 3-22.** Ladder program for the start/stop motor circuit.

**I/O Device and Power Wiring.** The I/O devices to be wired to the PLC are the stop push button, the start push button, and the motor starter. An internal instruction will be used to implement the auxiliary contact in the PLC.

Figure 3-21 shows how the I/O devices and power will be wired to the PLC. The stop button will be wired to the first input terminal (I:0.0/0) and the start button to the next input terminal (I:0.0/1). These are both DC inputs. The motor starter's coil will be wired to the first transistor output terminal (O:0/2). The overload contacts will remain wired in series with the motor starter's coil, just as they were in the electromechanical circuit.

To apply power to the PLC, you should connect the MicroLogix to the DC power supply's positive and negative voltages, noting that the negative is connected to ground. The PLC itself should also be grounded. Next, connect both the input devices and the output device to their respective DC power supply positive and negative terminals.

**Ladder Programming.** Figure 3-22 shows the ladder control program for the start/stop motor circuit. The ladder program uses the same logic that was used in the electromechanical circuit; however, the PLC program uses XIC contacts to reference the two inputs and an output coil to reference the motor. Internal contact O:0/2, which references the output coil, seals the circuit when the start push button is released.

The PLC program will work just like the hardwired circuit. When the start button is pressed, the motor will turn on and stay on. When the stop button is pressed, the motor will turn off. If the motor is sealed on and the overload contact opens, the motor starter will turn off. The reason for this is that the open overload contact will break the power continuity in the motor starter's circuit. This situation is called **low-voltage release**.

Note that, since the motor's PLC output coil is still on, the motor will turn on again as soon as the overload contact recloses (i.e., when the thermal overload heater cools off). Consequently, this circuit does not provide **low-voltage protection**. This can be undesirable and, in fact, dangerous in some applications. For instance, someone may be troubleshooting the motor to find out why it stopped. During this time, the overload contact may cool off and close again, causing the motor to start up again without warning.

**Low-Voltage Protection.** To provide low-voltage protection in this circuit (see Figure 3-23), you must connect the auxiliary contact from the motor starter as a real input to the PLC at input terminal 2 (I:0.0/2). Then use an XIC contact to implement the auxiliary contact in the PLC. This replaces the internal contact used before. Now the motor is sealed after the start button is pushed because the auxiliary contact stays on as long as the motor starter's coil is on. Therefore, this circuit still provides low-voltage release, since the motor will turn off when the overload opens. However, it also provides low-voltage protection, since the auxiliary contact will turn off when the motor starter's coil turns off. This means that the motor will stay off even after the overload recloses. You must press the start button for the motor to turn on again.

### Forward/Reverse Motor Circuit

Figure 3-24 illustrates the hardwired electromechanical diagram for a forward/reverse motor circuit (reversing motor starter). This circuit has two normally open start push buttons—a forward button and a reverse button. It also has two motor starter coils—one to start the motor going forward and another to start it going in reverse. The forward part of the circuit contains a contact labeled F1, which seals the forward push button. It also contains an R2 interlocking contact that prevents the forward

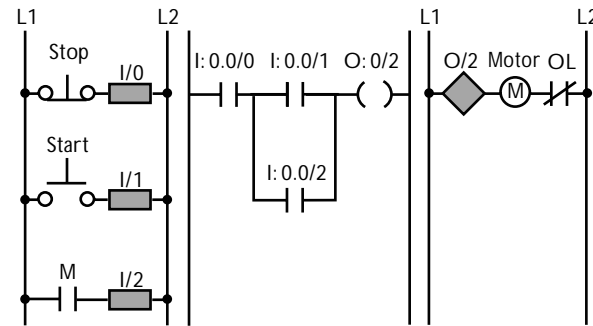


Figure 3-23. A star t/stop motor circuit with low-voltage release and low-voltage protection.

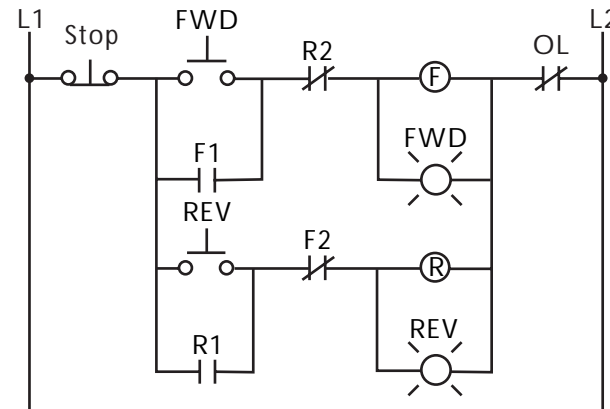


Figure 3-24. Hardwired diagram for a forward/reverse motor circuit.

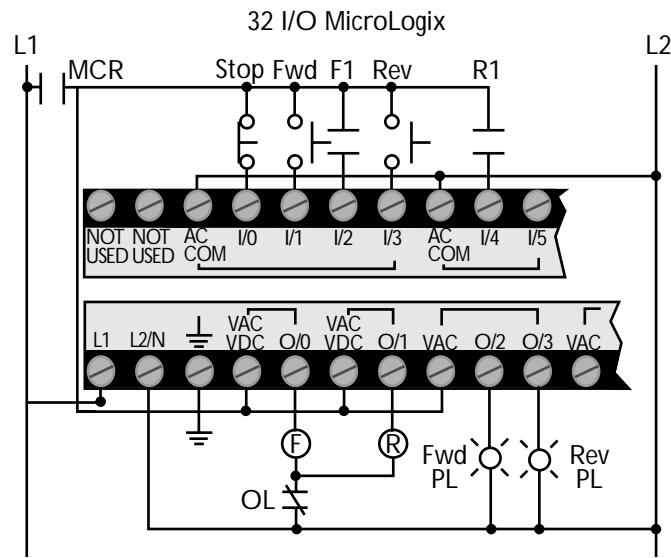


Figure 3-25. The I/O device and power wiring for the forward/reverse motor circuit.

motor starter from turning on if the reverse starter is already on. Moreover, it has a forward pilot light that turns on when the motor is moving in a forward direction.

The reverse part of the circuit contains similar elements and works identically to the forward part. Additionally, the circuit has a normally closed stop push button that can turn off either the forward or the reverse motor starter. It also contains an overload contact that will turn off either starter in the event of an overload condition.

**PLC Requirements.** To begin the implementation, you must choose the appropriate MicroLogix model for the application. In this example, the PLC will have the following specifications:

- 32 I/O
- 120-volt AC power supply
- 120-volt AC input interface
- relay/triac output interface

**I/O Device and Power Wiring.** In this forward/reverse circuit, all of the devices except the R2 and F2 contacts and the overload contact will be wired to the PLC. The motor starter contacts R1 and F1 will be wired to the PLC to provide low-voltage protection in the program. The R2 and F2 interlocking contacts, however, will be implemented in the PLC through internal contacts that reference the forward and reverse coils. The overload contact, which protects the motor, will be wired after the forward and reverse starters.

Figure 3-25 shows how the I/O devices will be wired to the MicroLogix's input and output interfaces. The input devices will be connected to the first five input terminals, and the output devices will be connected to the first four output terminals. The starter coils will be connected to output terminals 0 and 1, the two relay terminals, and the pilot lights will be connected to the

first two triac output terminals, which are terminals 2 and 3. The PLC will be connected to the L1 and L2 power lines. The input and output devices will also be connected to the L1 and L2 power lines.

**Ladder Programming.** Figure 3-26 shows the ladder program that will implement the forward/reverse motor circuit. The top two lines of the diagram control the forward motor starter and forward pilot light. The coils and contacts in the first line turn on the forward starter. This line starts out with an XIC contact that references the stop push button. The next contact is an XIC contact that maps the forward start push button. The final contact is an XIO interlocking contact that references the reverse starter output. This contact provides interlocking so that the forward starter will not turn on while the reverse starter is energized. When all of these inputs conditions are satisfied, the forward starter and the forward pilot light will turn on. When they turn on, the F1 auxiliary contact mapped to input I:0.0/2 will seal the circuit so that the forward starter remains on even if the forward push button is released. Pushing the stop push button will stop the forward motion of the motor.

The bottom two lines of the ladder diagram control the reverse starter and pilot light. They operate just like the first two lines do. Line three contains contacts referencing the reverse start push button and the forward motor starter. These are programmed in series with the stop push button contact located in line one. When all of these contacts are energized, the reverse motor starter and pilot light will turn on. The XIC contact in line four will seal the circuit. Again, pushing the stop button will remove power to the motor.

**Compelling Circuits.** In the circuit shown in Figure 3-26, the forward and reverse starters are interlocked so that one cannot turn on while the other is operating. However, the way the

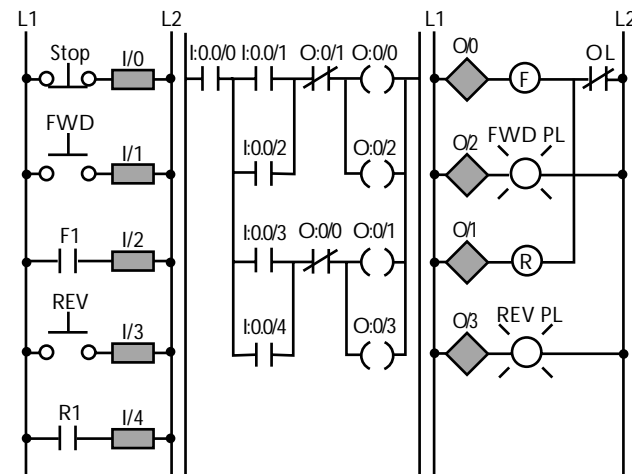


Figure 3-26. Ladder program that implements the forward/reverse circuit.



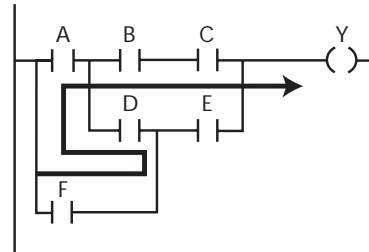
### 3-6 Review

- A ladder diagram consists of many rungs formed by contacts and coils.
- The ladder rung matrix determines how many branches, contacts, and coils can be used to program a ladder rung.
- Power must flow through a ladder rung in a left-to-right direction.
- There are two types of contact instructions used to program inputs in a MicroLogix 1000: examine-if-closed (XIC) and examine-if-open (XIO).
- An examine-if-closed contact provides power to the ladder rung when the contact's reference device is closed, or ON.
- An examine-if-open contact provides power to the ladder rung when the contact's reference device is open, or OFF.
- Four basic output instructions are used with the MicroLogix 1000: the standard output coil, the internal output coil, the latch/unlatch coil, and the one-shot rising instruction.
- A standard output coil is used to energize a real field output device connected to the PLC.
- An internal output coil is used to implement interlocking in a ladder program.
- Latch/unlatch coils are used to turn an output on and keep it on, even if its input logic turns off.
- A one-shot rising instruction is used to turn an output on for one scan.
- In ladder scan evaluation, a PLC only evaluates the ladder program once during each scan, and it does so from left to right and from top to bottom.
- When programming a normally closed input device, you must use an examine-if-closed instruction for the device to behave like a normally closed one in the PLC program.
- When programming a safety stop device in a PLC, never use a normally open device programmed with an examine-if-open contact.
- You should approach the programming of all PLC applications the same way: (1) identify the control requirements, (2) determine which I/O devices will be connected to the PLC, (3) determine which terminals the I/O devices will be connected to, (4) determine how the I/O devices and PLC will be powered, and (5) develop a ladder program that implements the required control.

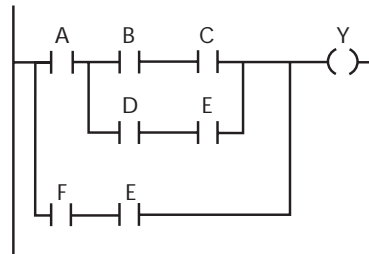
## 3-7 Job Aids

### Job Aid 3-1: Programming Around Reverse Power Flow

The following ladder diagram allows power to flow from right to left through contact D. This type of reverse power flow is not permitted in a MicroLogix ladder program.



If the reverse condition is not a required part of the output's logic, then the ladder rung can be reprogrammed as follows to avoid the reverse power flow condition:



If the reverse condition is a required part of the output's logic, then the ladder rung can be reprogrammed as follows to avoid the reverse power flow condition:

