➢Chapter 5
➢Basics of PLC Programming

❑overview

- Each input and output PLC module terminal is identified by a unique address
- In PLCs, the internal symbol for any input is a contact
- Similarly, in most cases, the internal PLC symbol for all outputs is a coil
- This chapter shows how these contact/coil functions are used to program a PLC for circuit operation
- This chapter covers only the basic set of instructions that perform functions similar to relay functions.
- You will also learn more about the program scan cycle and the scan time of a PLC.

❑Processor Memory Organization

➢The memory map or structure for a PLC processor consists of several areas, some of these having specific roles.

➢Memory organization takes into account the way a PLC divides the available memory into different sections.

➢The memory space can be divided into two broad categories:

❖ *program files*

❖ *data files.*

➢*sections, their* order, and the sections' length will vary and may be fixed or variable, depending on the manufacturer and model.

❑Processor Memory Organization

- The data files store the information needed to carry out the user program.
- This includes information such as the status of input and output devices, timer and counter values, data storage, and so on.
- Contents of the data table can be divided into two categories: status data and numbers or codes.
- Status is ON/OFF type of information represented by 1s and 0s, stored in unique bit locations.
- Number or code information is represented by groups of bits that are stored in unique byte or word locations.

❑Processor Memory Organization

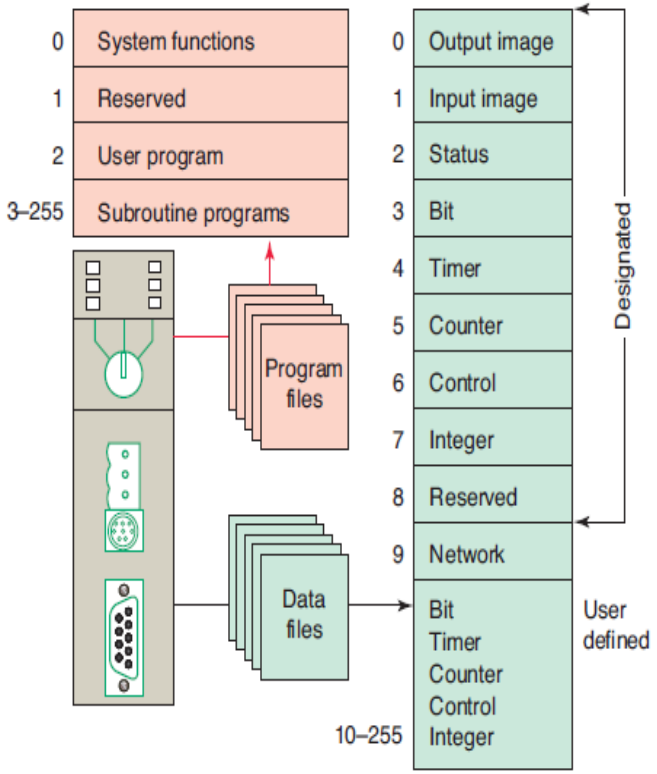
➢Figure 5-1 shows the program and data file organization for the SLC 500

❖Program Files

➢Program files are the areas of processor memory where ladder logic programming is stored.

- **System functions (file 0)**

This file is always included and contains various system-related information and user-programmed information such as processor type, I/O configuration, processor file name, and password.

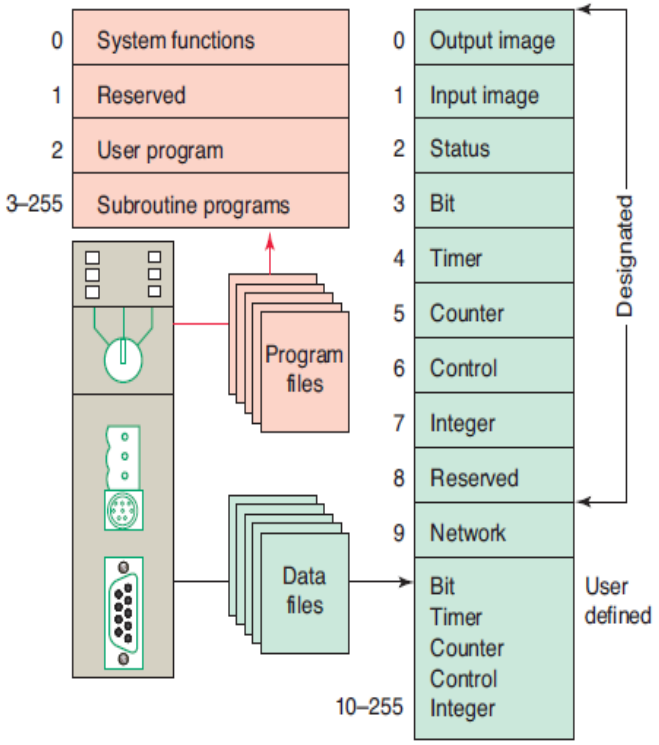


❑Processor Memory Organization

•**Reserved (file 1) —**This file is reserved by the processor and is not accessible to the user.

• **Main ladder program (file 2) —**This file is always included and contains user-programmed instructions that define how the controller is to operate.

• **Subroutine ladder program (files 3–255)** These files are user-created and are activated according to subroutine instructions residing in the main ladder program file.



❑Processor Memory Organization

❖Data Files

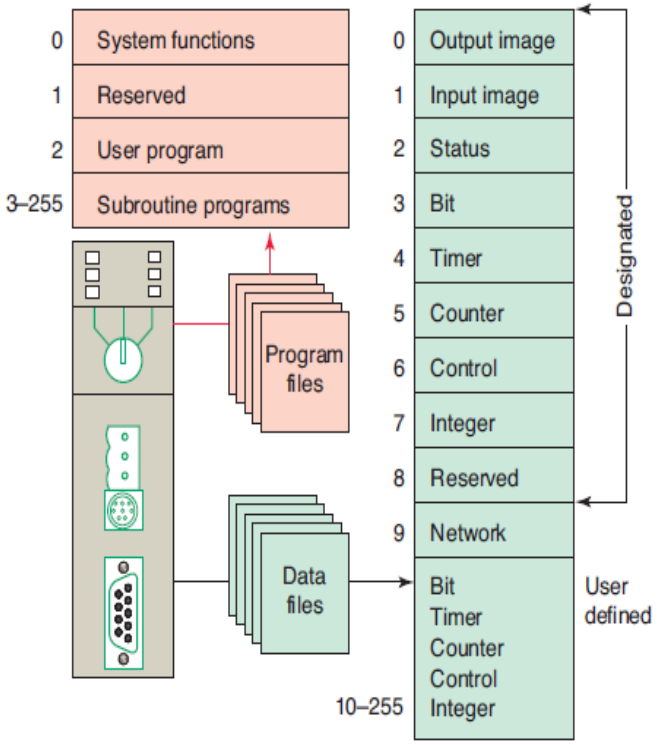
➢The data file portion of the processor's memory stores input and output status, processor status, the status of various bits, and numerical data.

➢All this information is accessed via the ladder logic program.

➢These files are organized by the type of data they contain and may include:

Output (file 0) —This file stores the state of the output terminals for the controller.

• **Input (file 1) —**This file stores the status of The input terminals for the controller.



❑Processor Memory Organization

❖Data Files

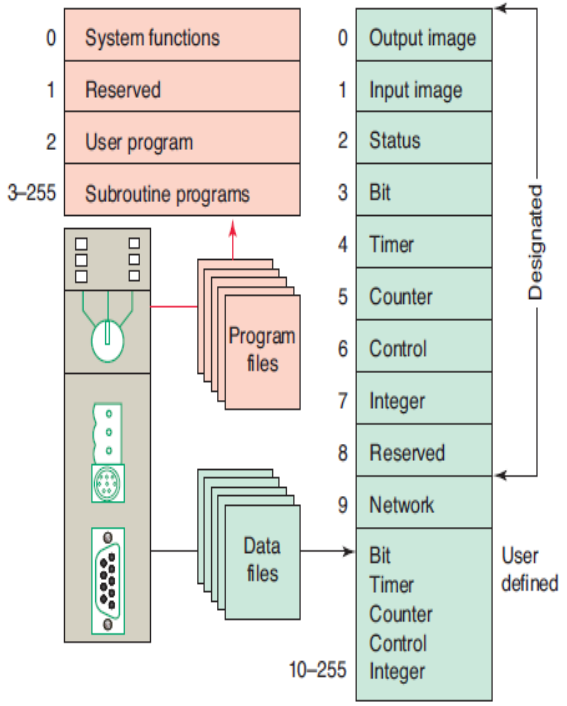
➢**Status (file 2) -**This file stores controller operation information and is useful for troubleshooting controller and program operation.

• **Bit (file 3) -**This file is used for internal relay logic storage.

• **Timer (file 4) -**This file stores the timer accumulated and preset values and status bits.

•**Counter (file 5) -**This file stores the counter Accumulated and preset values and status bits.

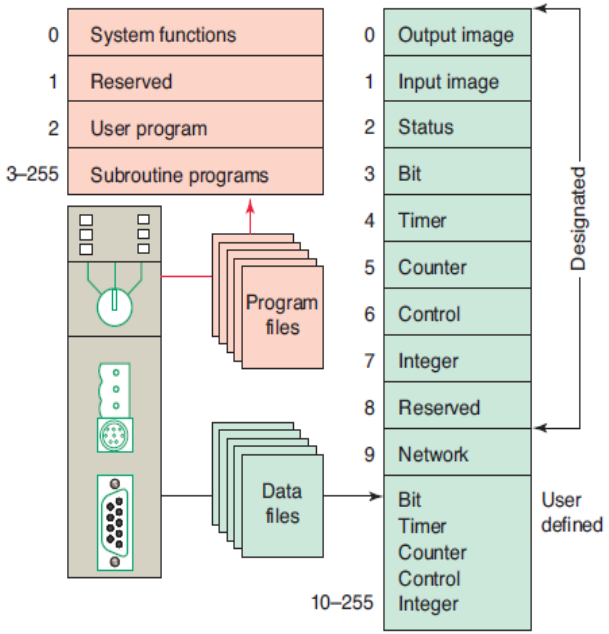
• **Control (file 6) -**This file stores the length, Pointer position, and status bit for specific instructions such as shift registers and sequencers.



❑Processor Memory Organization

❖Data Files

- **Integer (file 7) -**This file is used to store numerical values or bit information.
- **Reserved (file 8)** This file is not accessible to the user.
- **Network communications (file 9) -**This file is used for network communications if installed or used like files 10–255.
- **User-defined (files 10–255) -**These files are userdefined as bit, timer, counter, control, and/or integer data storage



❑Processor Memory Organization

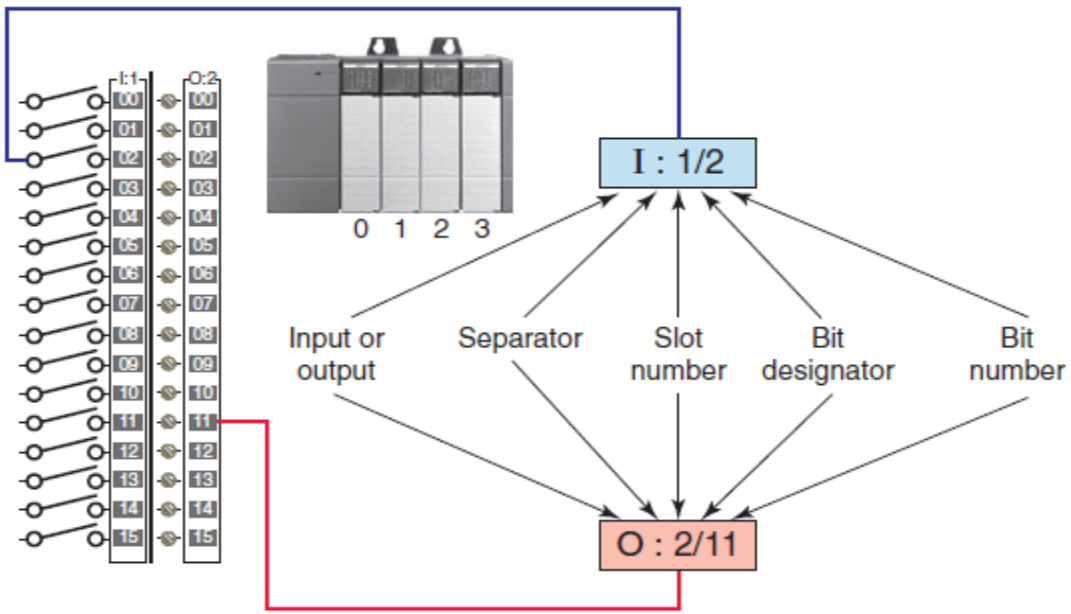
➢The I/O address format for the SLC family of PLCs is shown in Figure.

➢The format consists of the following three parts:

❖***Part 1:*** I for input, and a colon to separate the module type from the slot. O for output and a colon to separate the module type from the slot.

Part 2: The module slot number and a forward slash to separate the slot from the terminal screw.

Part 3: *The screw terminal number.*



❑Processor Memory Organization

- ➢Figure shows a typical data file memory organization for an Allen-Bradley PLC-5 controller.
- ➢Each data file is made up of numerous *elements.*
- ➢*Each element may be* one, two, or three words in length.
- ➢Timer, counter, and control elements are three words in length; floating-point elements are two words in length; and all other elements are a single word in length

Address range		Size, in elements
O:000 O:037	Output image file	32
I:000 I:037	Input image file	32
S:000 S:031	Processor status	32
B3:000 B3:999	Bit file	1–1000
T4:000 T4:999	Timer file	1–1000
C5:000 C5:999	Counter file	1–1000
R6:000 R6:999	Control file	1–1000
N7:000 N7:999	Integer file	1–1000
F8:000 F8:999	Floating-point file	1–1000
	Files to be assigned file nos. 9–999	1–1000 per file

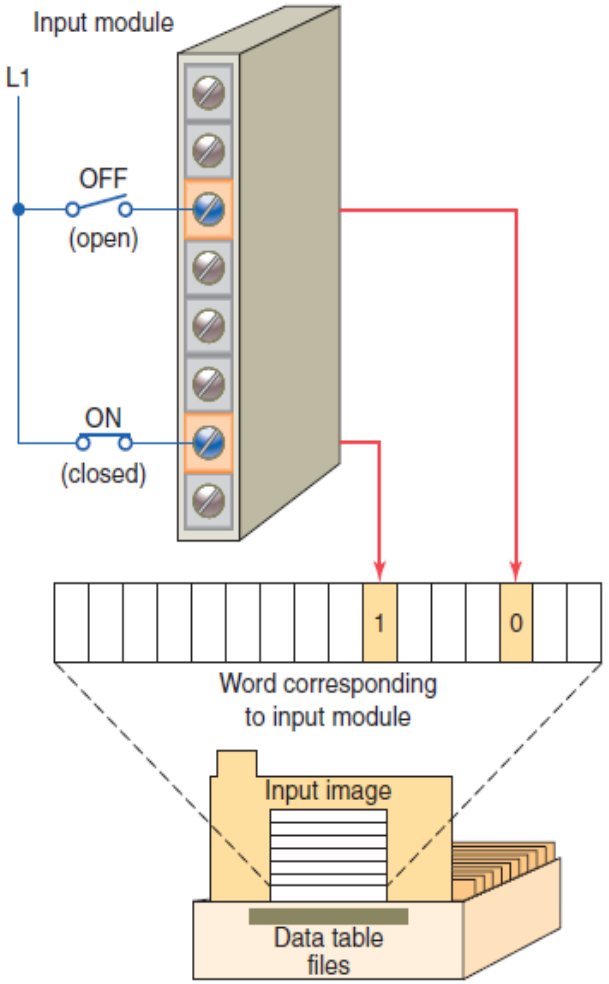
❑Processor Memory Organization

➢Figure shows the connection of an open and closed switch to the input image table file through the input module.

➢Its operation can be summarized as follows

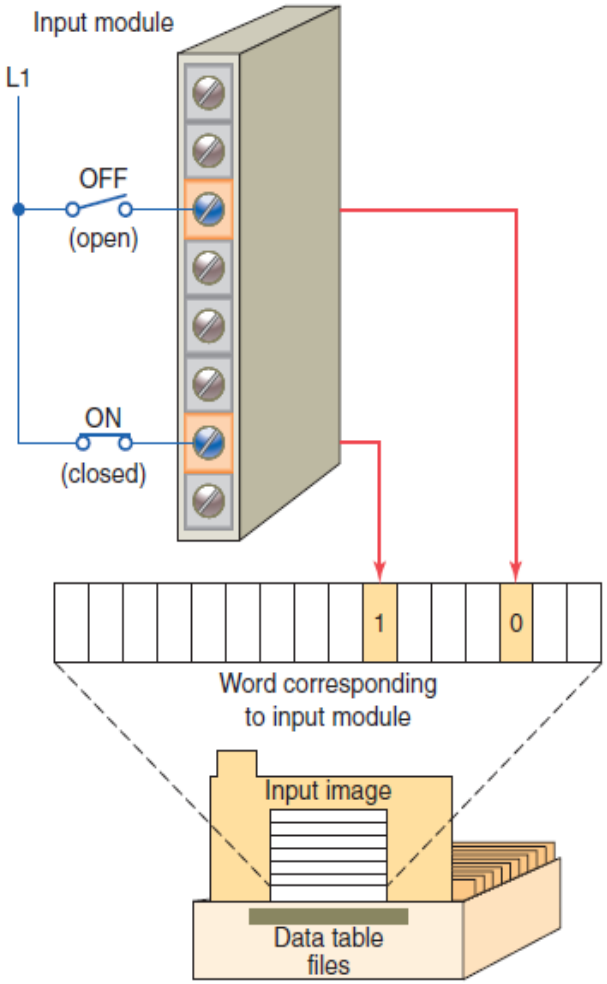
▪For the switch that is closed, the processor detects a voltage at the input terminal and records that information by storing a binary 1 in its bit location.

• For the switch that is open, the processor detects no voltage at the input terminal and records that information by storing a binary 0 in its bit location.



❑Processor Memory Organization

- Each connected input has a bit in the input image table file that corresponds exactly to the terminal to which the input is connected.
- The input image table file is changed to reflect the current status of the switch during the I/O scan phase of operation.
- If the input is on (switch closed), its corresponding bit in the table is set to 1.
- If the input is off (switch open), the corresponding bit is cleared, or reset to 0.
- The processor continually reads the current input status and updates the input image table file.



❑Processor Memory Organization

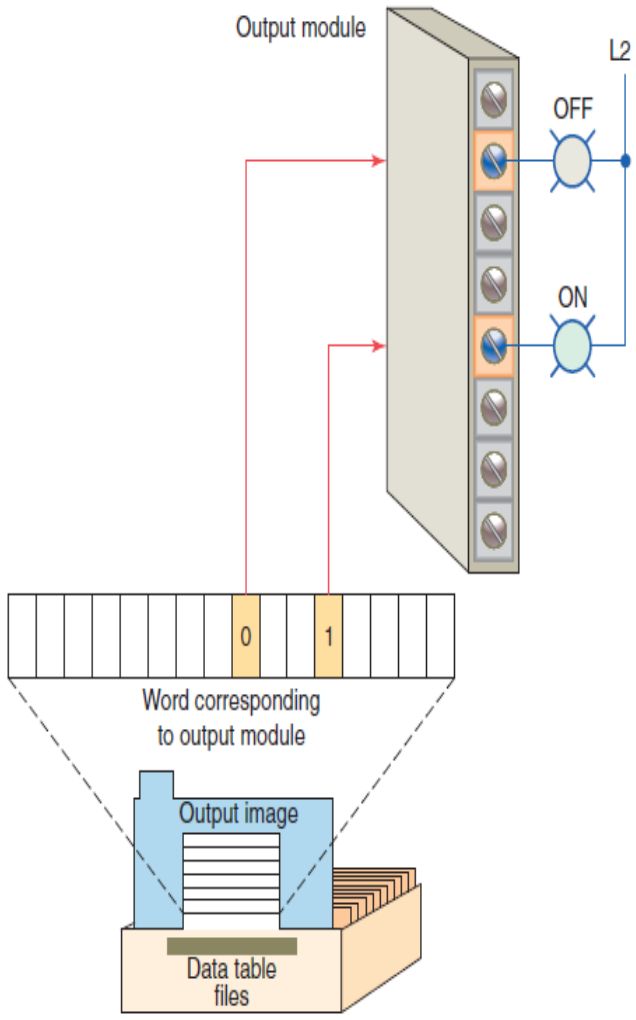
➢Figure shows a typical connection of two pilot lights to the output image table file through the output module.

➢Its operation can be summarized as follows:-

➢The status of each light (ON/OFF) is controlled by the user program and is indicated by the presence of 1 (ON) and 0 (OFF).

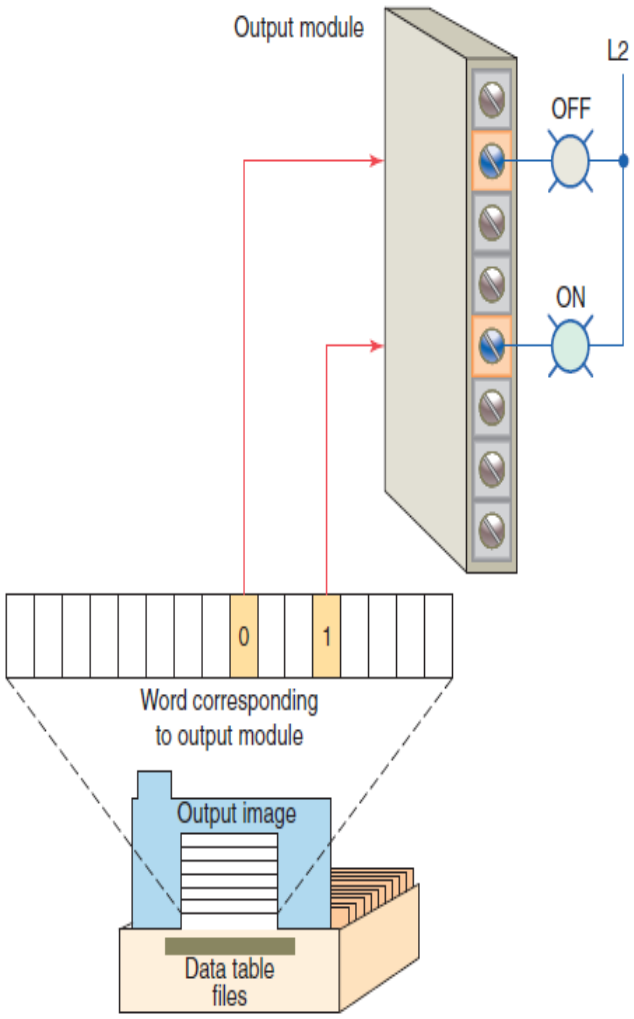
➢Each connected output has a bit in the output image table file that corresponds exactly to the terminal to which the output is connected.

• If the program calls for a specific output to be ON, its corresponding bit in the table is set to 1.



❑Processor Memory Organization

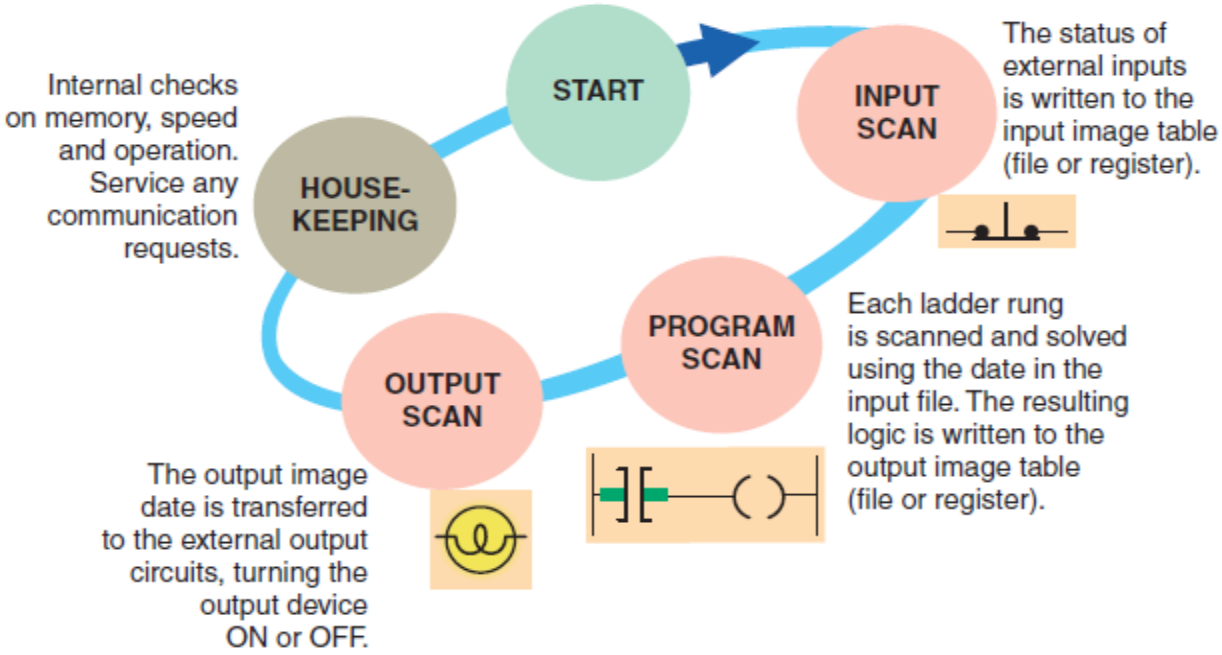
- If the program calls for the output to be OFF, its corresponding bit in the table is set to 0.
- The processor continually activates or deactivates the output status according to the output table file status



❑Program Scan

➢Figure illustrates a single PLC operating cycle consisting of the *input scan, program scan, output scan, and housekeeping* duties.

➢Because the inputs can change at any time, it constantly repeats this cycle as long as the PLC is in the RUN mode.



❑Program Scan

- The time it takes to complete a scan cycle is called the scan cycle time and indicates how fast the controller can react to changes in inputs.
- The time required to make a single scan can vary from about 1 millisecond to 20 milliseconds.
- If a controller has to react to an input signal that changes states twice during the scan time, it is possible that the PLC will never be able to detect this change.
- For example, if it takes 8 ms for the CPU to scan a program, and an input contact is opening and closing every 4 ms, the program may not respond to the contact changing state.
- The CPU will detect a change if it occurs during the update of the input image table file, but the CPU will not respond to every change.

❑Program Scan

➢ The scan time is a function of the following:

- The speed of the processor module
- The length of the ladder program
- The type of instructions executed
- The actual ladder true/false conditions

➢ Typical scan time data include the maximum scan time and the last scan time.

➢The scan is normally a continuous and sequential process of reading the status of inputs, evaluating the control logic, and updating the outputs.

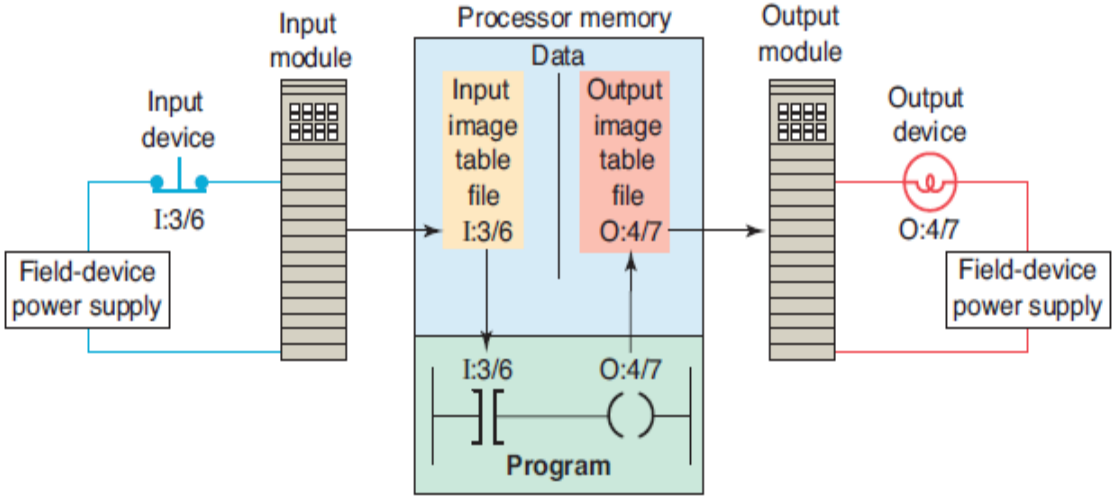
❑Program Scan

➢ If the input device connected to address I:3/6 is closed, the input module circuitry senses *electrical continuity and a 1 (ON) condition is entered into the* input image table bit I:3/6.

➢During the program scan, the processor examines bit I:3/6 for a 1 (ON) condition.

➢ In this case, because input I:3/6 is 1, the rung is said to be TRUE or have logic continuity.

• The processor then sets the output image table bit O:4/7 to 1.



❑Program Scan

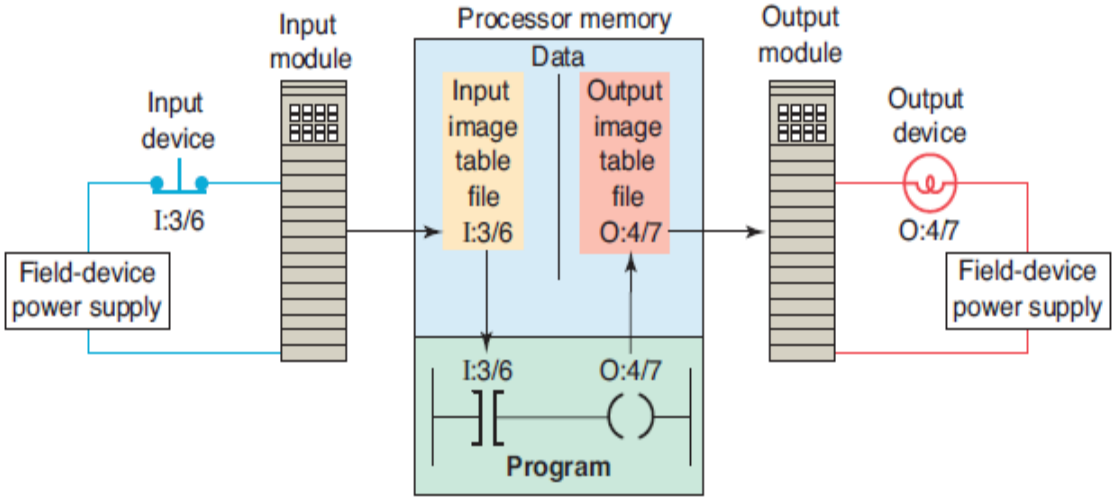
• The processor turns on output O:4/7 during the Next I/O scan, and the output device (light) wired to this terminal becomes energized.

• This process is repeated as long as the processor is in the RUN mode.

• If the input device opens, electrical continuity is lost, and a 0 would be placed in the input image table.

•As a result, the rung is said to be FALSE due to loss of logic continuity.

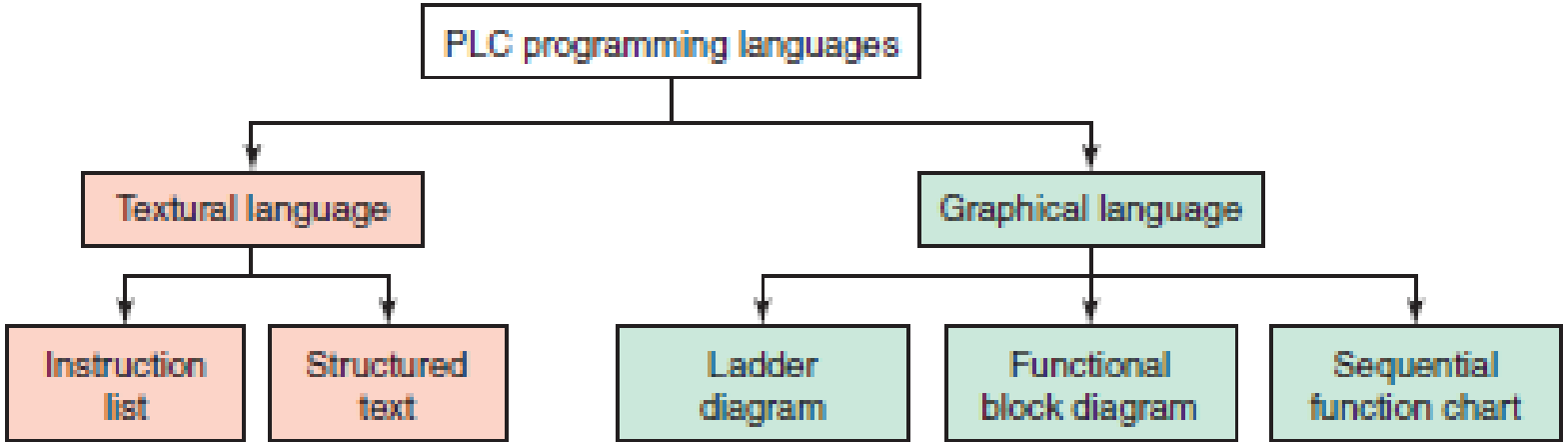
• The processor would then set the output image table bit O:4/7 to 0, causing the output device to turn off.



❑PLC Programming Languages

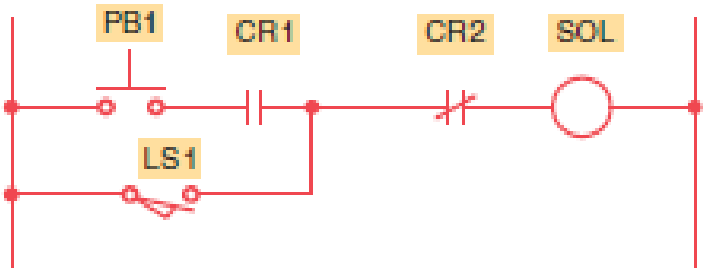
➢The term *PLC programming language refers to the* method by which the user communicates information to the PLC.

➢The standard IEC 61131 (Figure) was established to standardize the multiple languages associated with PLC programming by defining the following five standard languages:

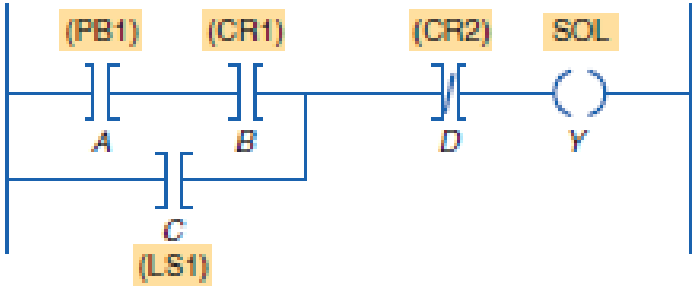


❑PLC Programming Languages

• **Ladder Diagram (LD) —**a graphical depiction of a process with rungs of logic, similar to the relay ladder logic schemes that were replaced by PLCs.



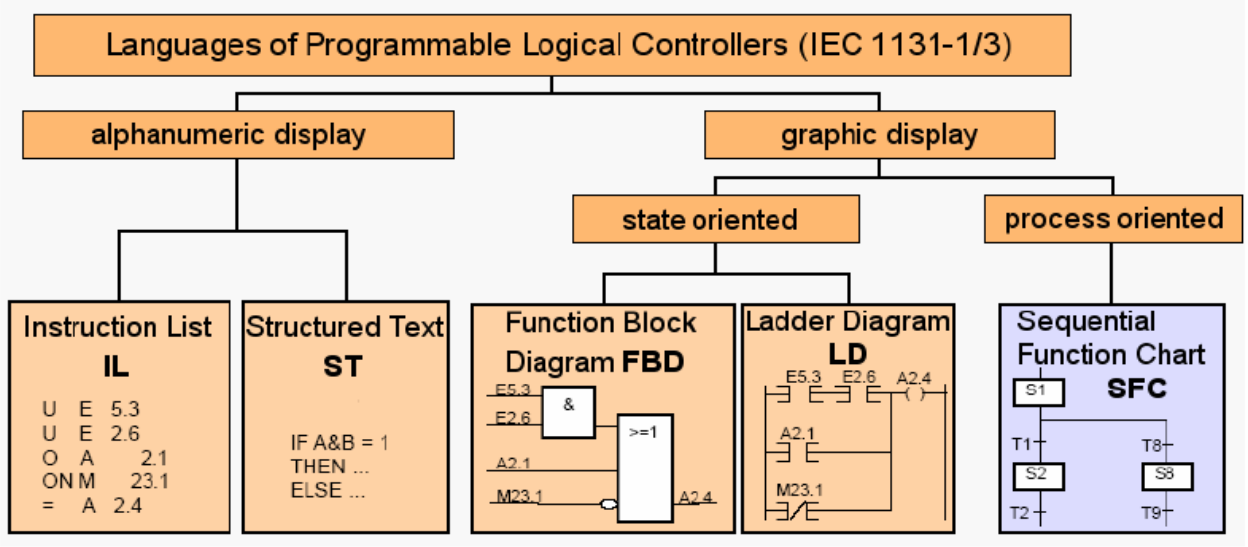
(*a*) Hardwired relay control circuit



(*b*) Equivalent ladder diagram (LD) program

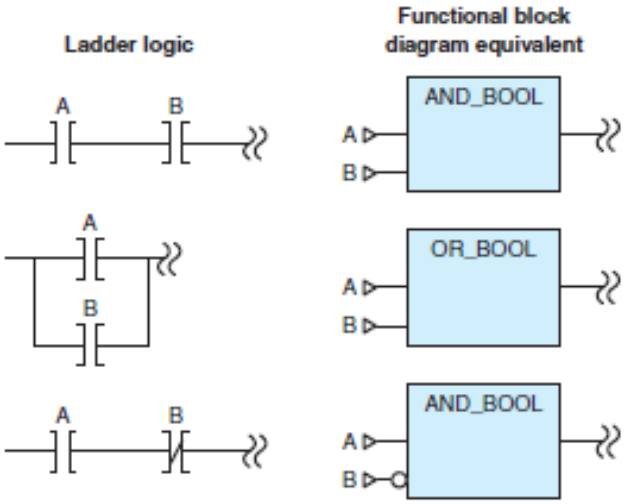
❑PLC Programming Languages

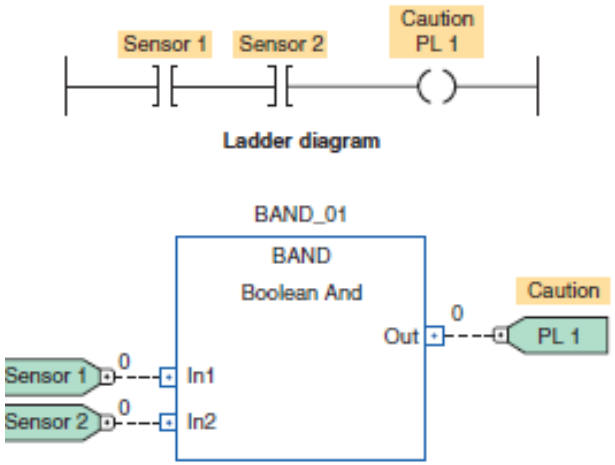
- **Function Block Diagram (FBD)** —a graphical depiction of process flow using simple and complex interconnecting blocks.
- **Sequential Function Chart (SFC)** —a graphical depiction of interconnecting steps, actions, and transitions.
- **Instruction List (IL)** —a low-level, text-based language that uses mnemonic instructions.
- **Structured Text (ST)** —a high-level, text-based language such as BASIC, C, or PASCAL specifically developed for industrial control applications.



❑PLC Programming Languages

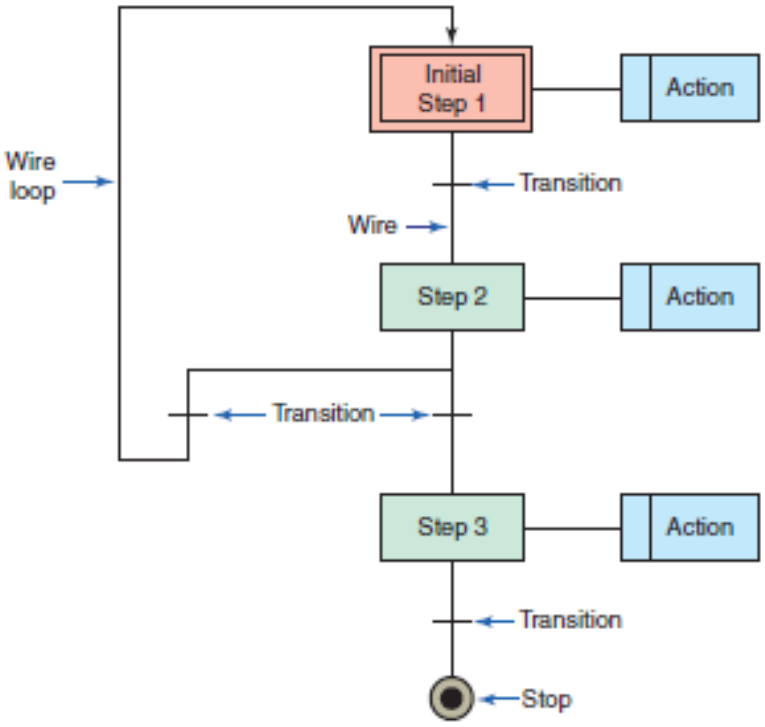
• **Function Block Diagram (FBD) —**a graphical depiction of process flow using simple and complex interconnecting blocks.





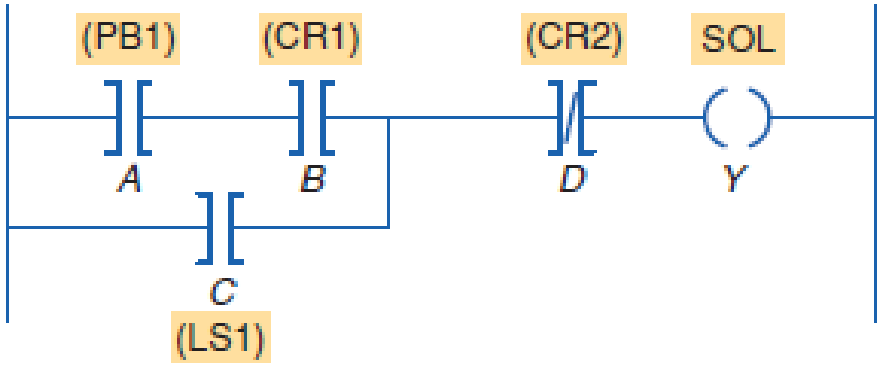
❑PLC Programming Languages

• **Sequential Function Chart (SFC) —a graphical** depiction of interconnecting steps, actions, and transitions.



❑PLC Programming Languages

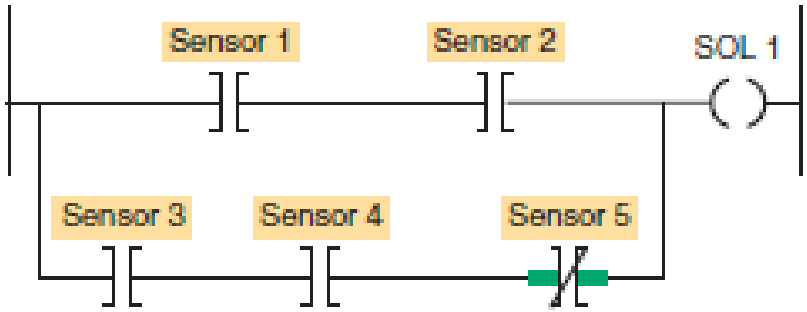
- **Instruction List (IL) —a low-level, text-based** language that uses mnemonic instructions.



START	PB1
AND	CR1
OR	LS1
AND NOT	CR2
OUT	SOL

❑PLC Programming Languages

• **Structured Text (ST) —**a high-level, text-based language such as BASIC, C, or PASCAL specifically developed for industrial control applications.



Ladder diagram (LD) program

```
IF Sensor_1 AND Sensor_2 THEN
        SOL_1 := 1;
ELSEIF Sensor_3 AND Sensor_4 AND NOT Sensor_5 THEN
        SOL_1 := 1;
END_IF;
```

❑Relay-Type Instructions

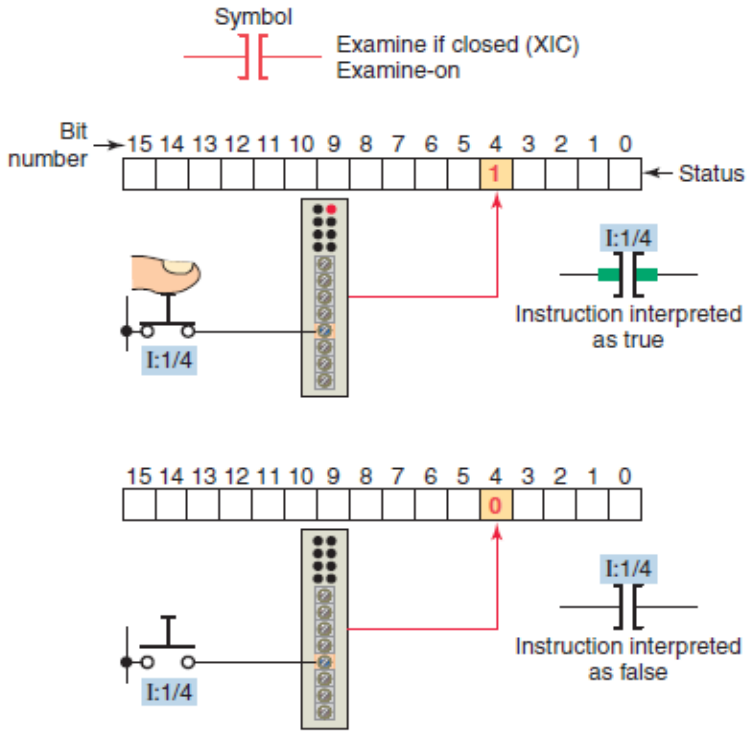
➢*Examine If Closed (XIC)*

✓The symbol for the *Examine If Closed (XIC) instruction* is shown in Figure .

✓Examine-on instruction, looks and operates like a normally open relay contact

✓This instruction asks the PLC's processor to examine if the contact is *closed.*

✓*It does this by examining the* bit at the memory location specified by the address



❑Relay-Type Instructions

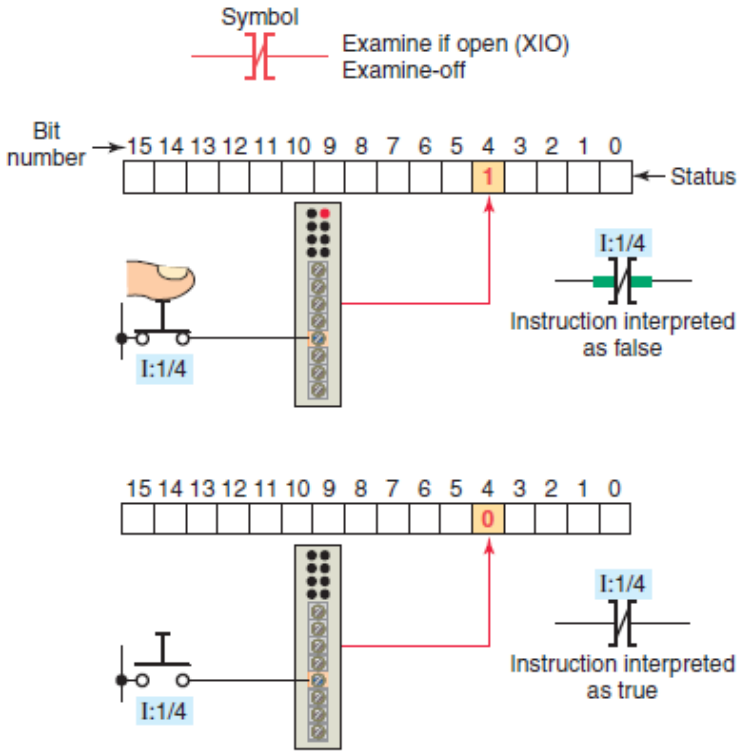
➢*Examine If Open (XIO)*

➢The symbol for the *Examine If Open (XIO) instruction* is shown in Figure .

✓Examine-off instruction, looks and operates like a normally closed relay contact

✓This instruction asks the PLC's processor to examine if the contact is *open.*

✓*It does this by examining the* bit at the memory location specified by the address



❑Relay-Type Instructions

➢*Output Energize (OTE)*

➢The symbol for the *Output Energize (OTE) instruction* is shown in Figure

✓ The OTE instruction looks and operates like a relay coil and is associated with a memory bit.

✓This instruction signals the PLC to energize (switch on) or de-energize (switch off) the output.

✓The processor makes this instruction true (analogous to energizing a coil) when there a logical path of true XIC and XIO instruction in the rung.