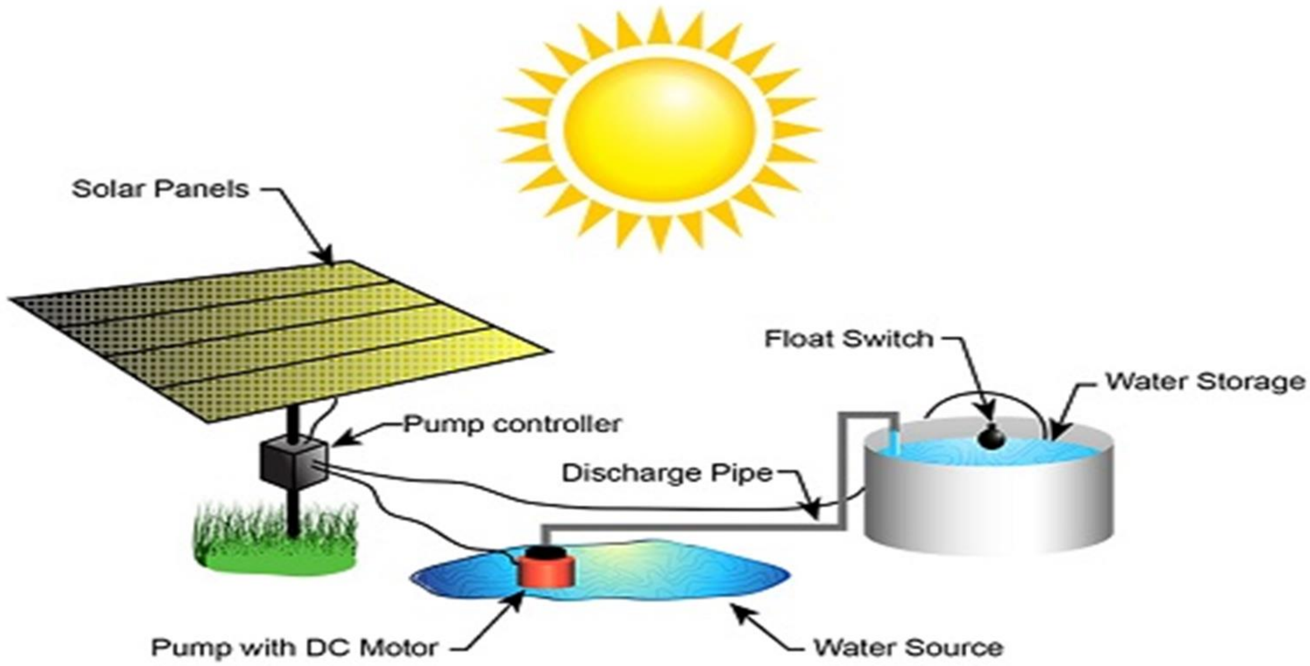


Control System & PLC

PLC Hardware & Programming

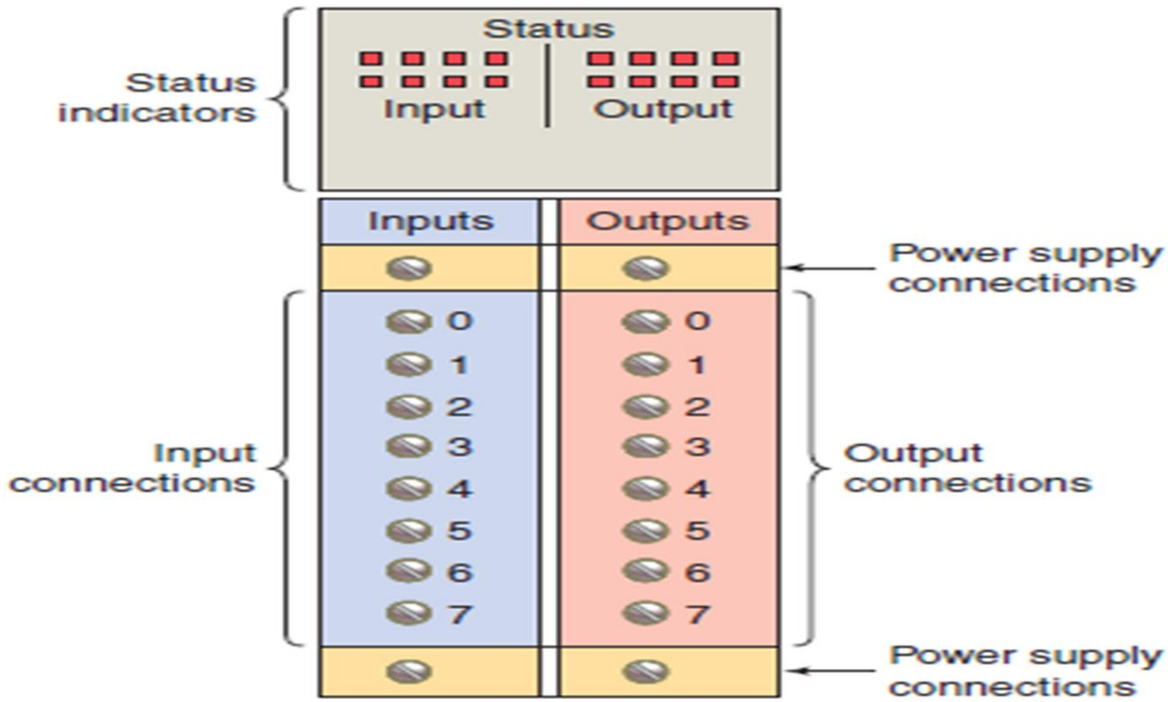
Content

- Review of Last Lecture.
- Discrete input & output modules.
- Analog input & output modules.
- I/O addressing of PLC.
- PLC instruction set.
- PLC programming examples.

Learning Objectives

- Function of discrete input & output module.
- I/O addressing of PLC.
- Analysis of PLC instruction set.
- PLC programming examples.

I/O Module

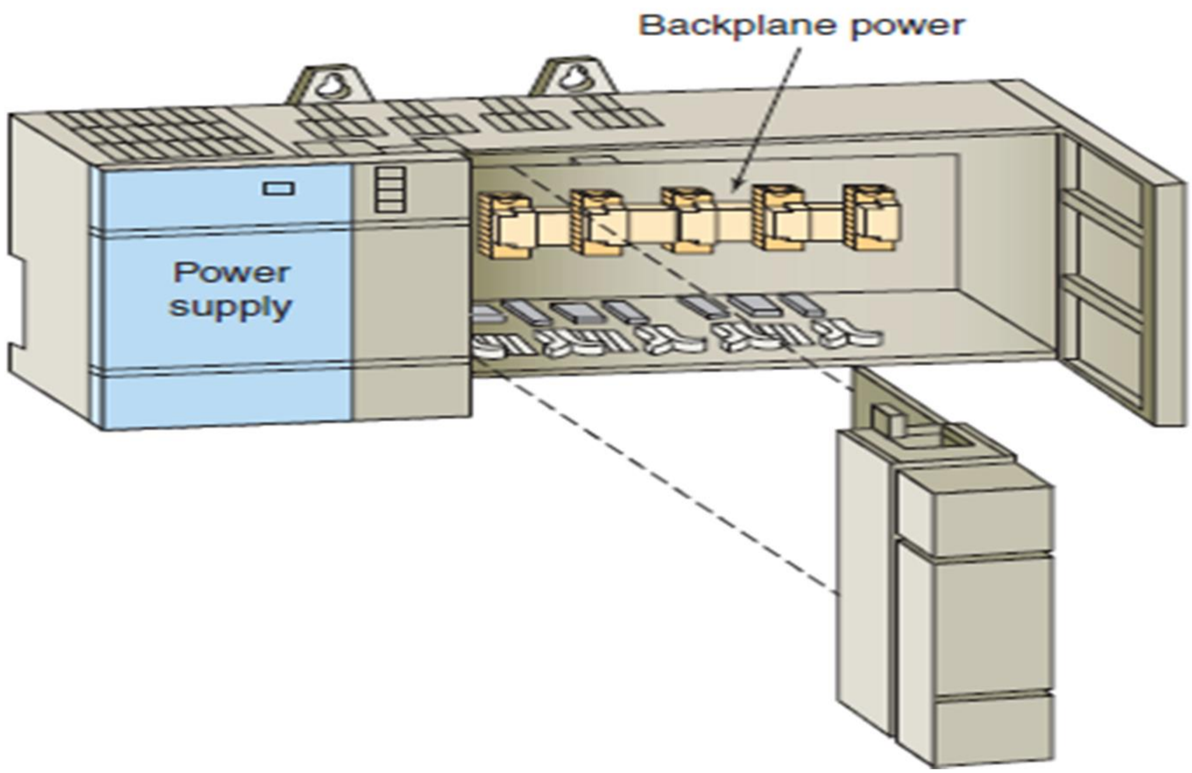


- Most common type of I/O interface module is the discrete type .
- Classification of discrete I/O covers bit oriented inputs and outputs

I/O Module

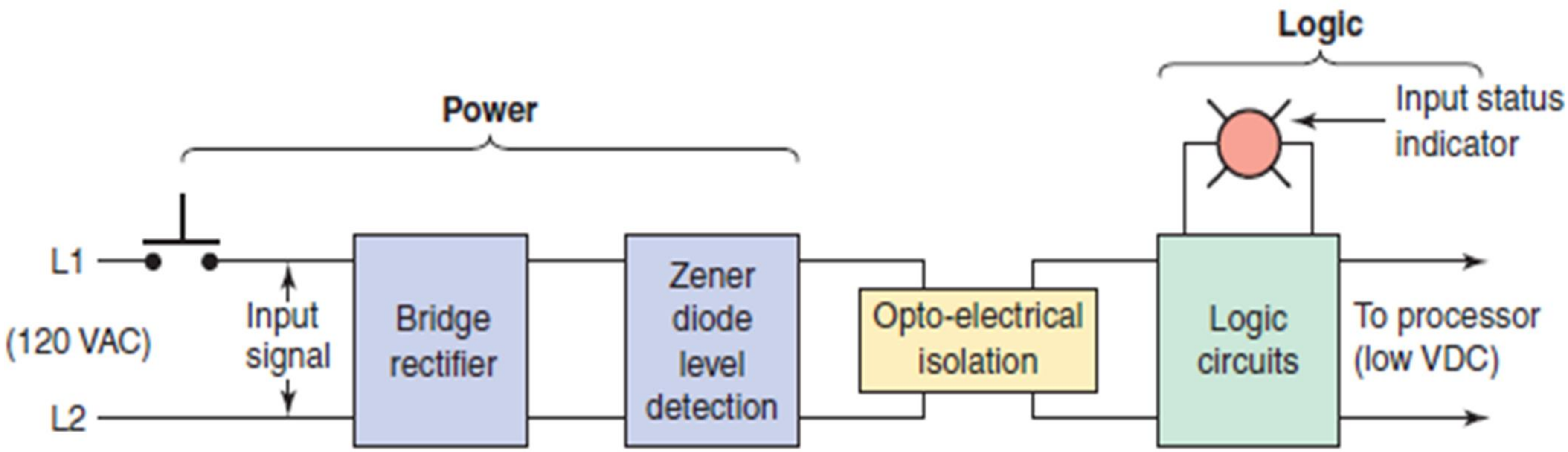
- In this type of input or output, each bit represents a complete information element in itself and provides the status of some external contact.
- Modules themselves receive their voltage and current for proper operation from the backplane of the rack enclosure into which they are inserted.
- Backplane power is provided by the PLC module power supply.

I/O Module



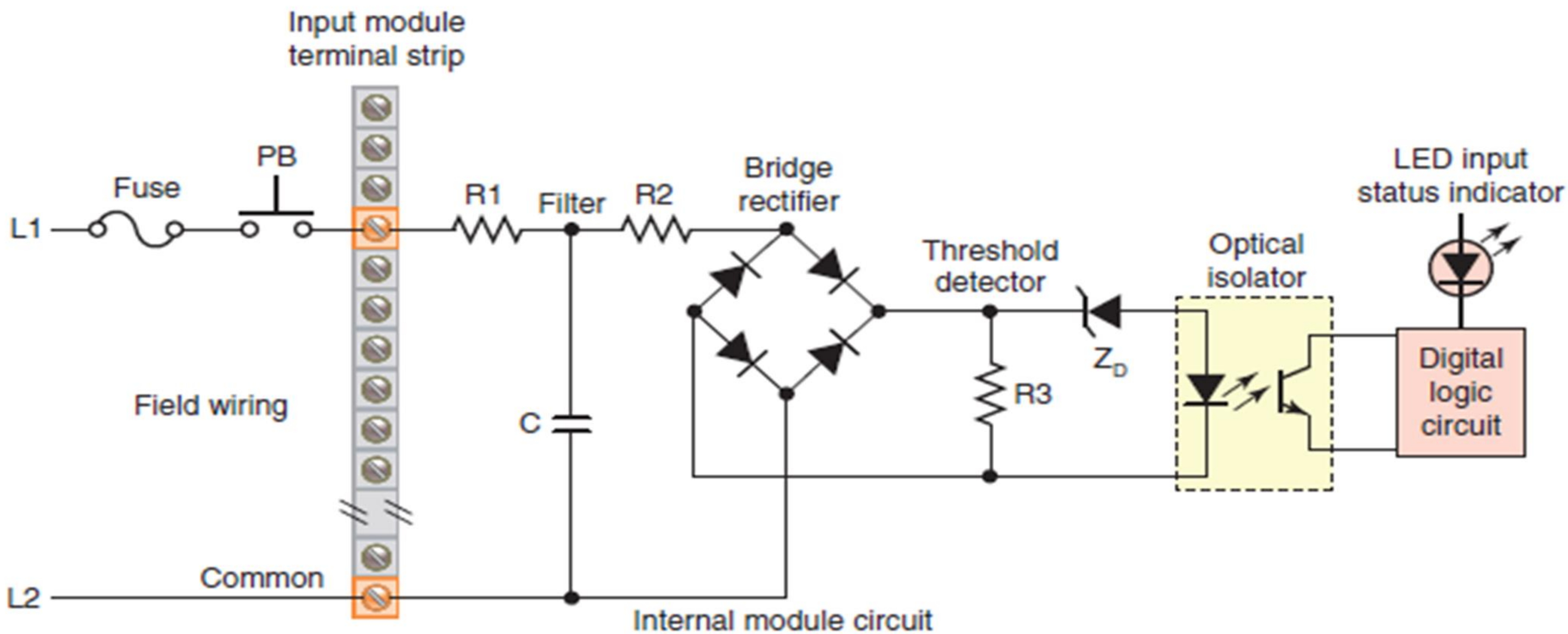
- PLC module power supply is used to power the electronics that reside on the I/O module circuit board.

Discrete Input Module



- Optical isolator is used to provide electrical isolation between the field wiring and the PLC backplane internal circuitry.
- Internal PLC control circuitry typically operates at 5 VDC or less volts.

Simplified Diagram (AC I/P)

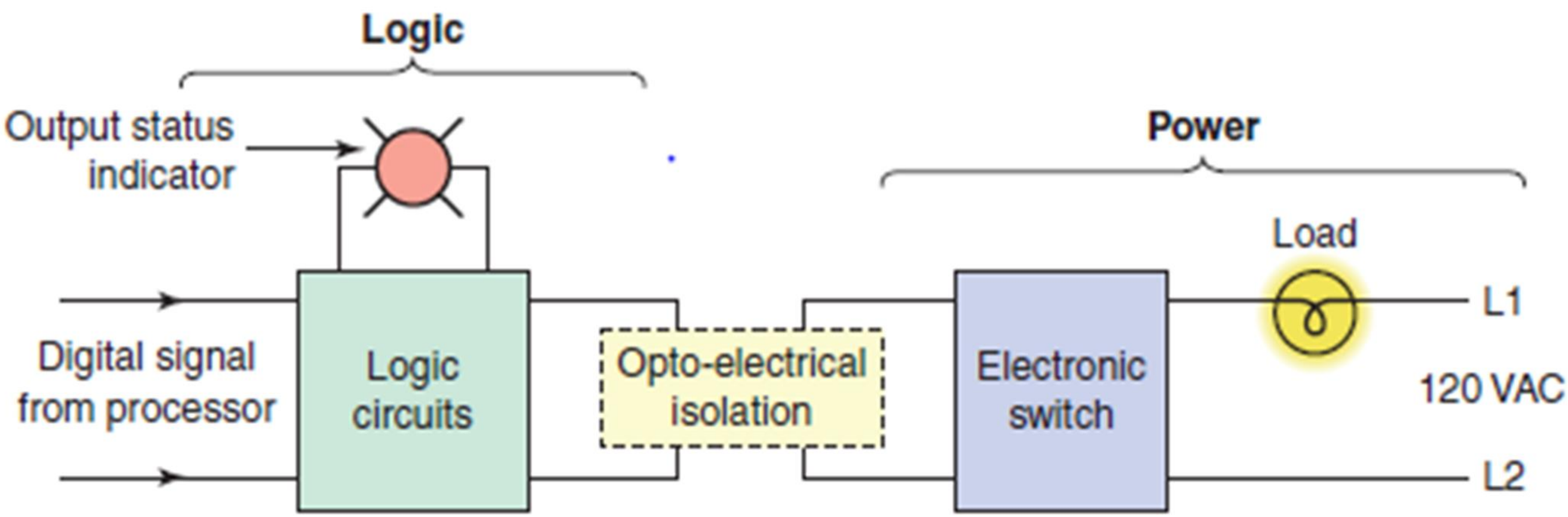


- Input noise filter removes false signals that are due to contact bounce or electrical interference.
- When the pushbutton is closed, 120 VAC is applied to the bridge rectifier input.

Simplified Diagram (AC I/P)

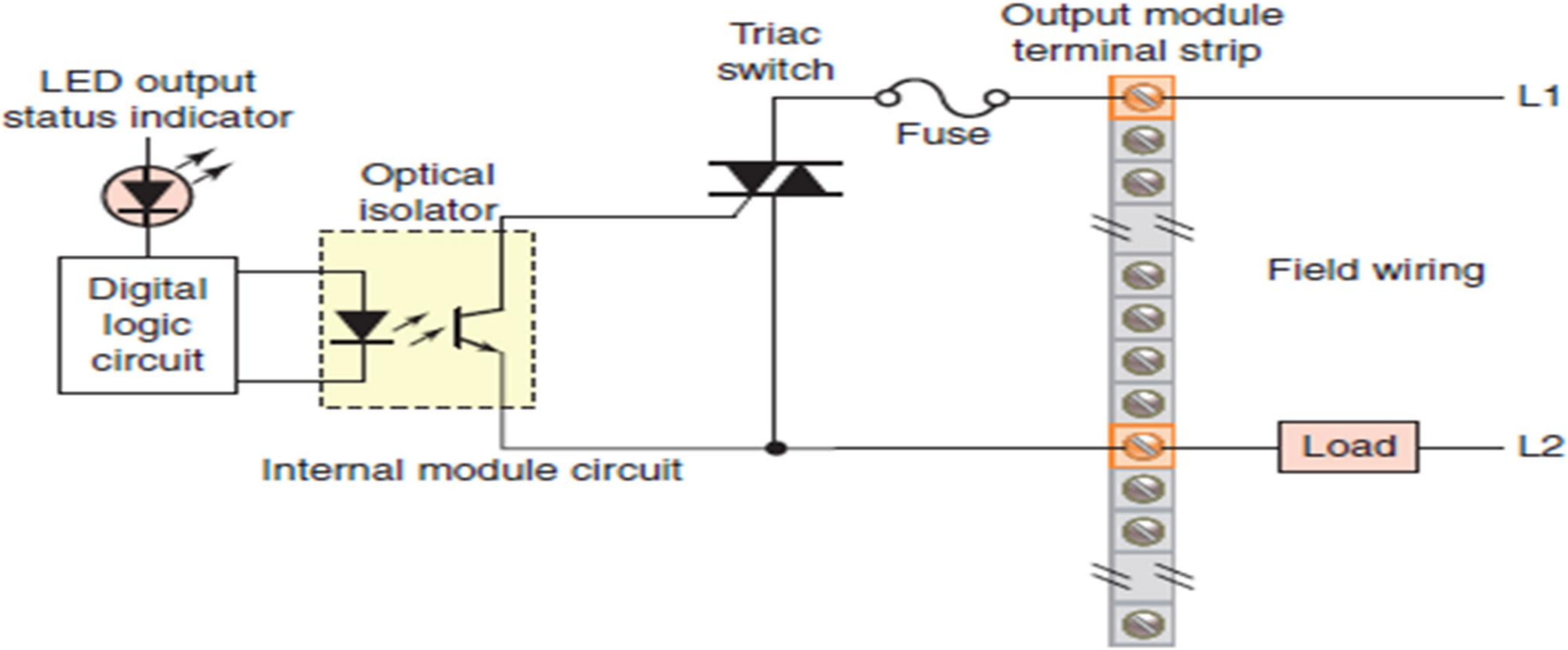
- Low-level DC output voltage is applied across the LED of the optical isolator.
- ZD voltage rating sets the minimum threshold level of voltage.
- Light from the LED strikes the phototransistor, it switches into conduction and the status of the pushbutton is communicated in logic to the processor.
- optical isolator not only separates the higher AC input voltage from the logic circuits but also prevents damage to the processor due to line voltage transients.
- An input state LED indicator is on when the input pushbutton is closed.

Discrete Output Module



- The output interface can be thought of as an electronic switch.
- An output LED indicates the status of the output signal.

Simplified Diagram (AC O/P)



- Digital logic circuits of the processor sets the output status according to the program.
- When processor calls for an output load to be energized, a voltage is applied across the LED of the opto-isolator

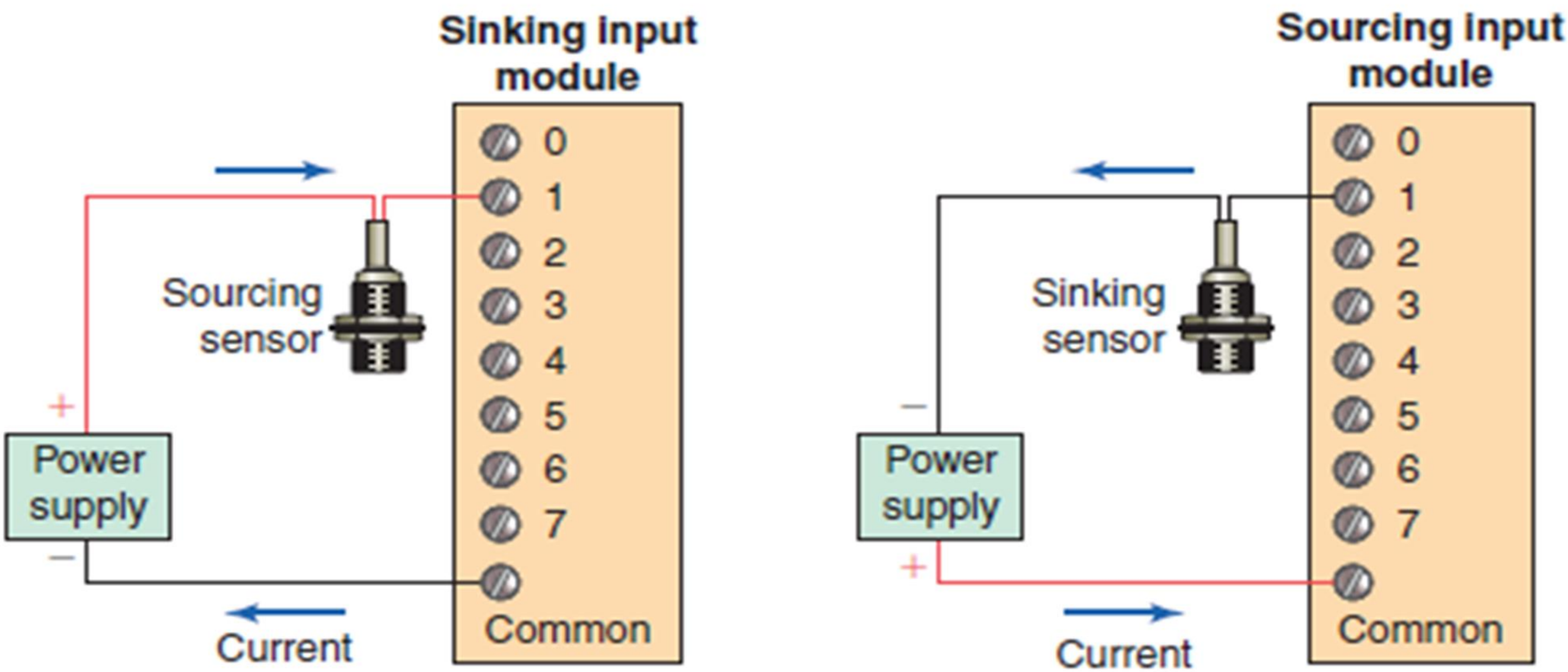
Simplified Diagram (AC O/P)

- LED then emits light, which switches the phototransistor into conduction.
- Triac AC semiconductor is triggered to switch into conduction allowing current to flow to the output load.
- Output to the load is alternating current due to which triac has LOW and HIGH resistance levels.
- Output interface is usually provided with LEDs that indicate the status of each output.
- Triac cannot be used to switch a DC load.
- Transistor outputs can be used only for control of DC devices

Sourcing and Sinking Devices

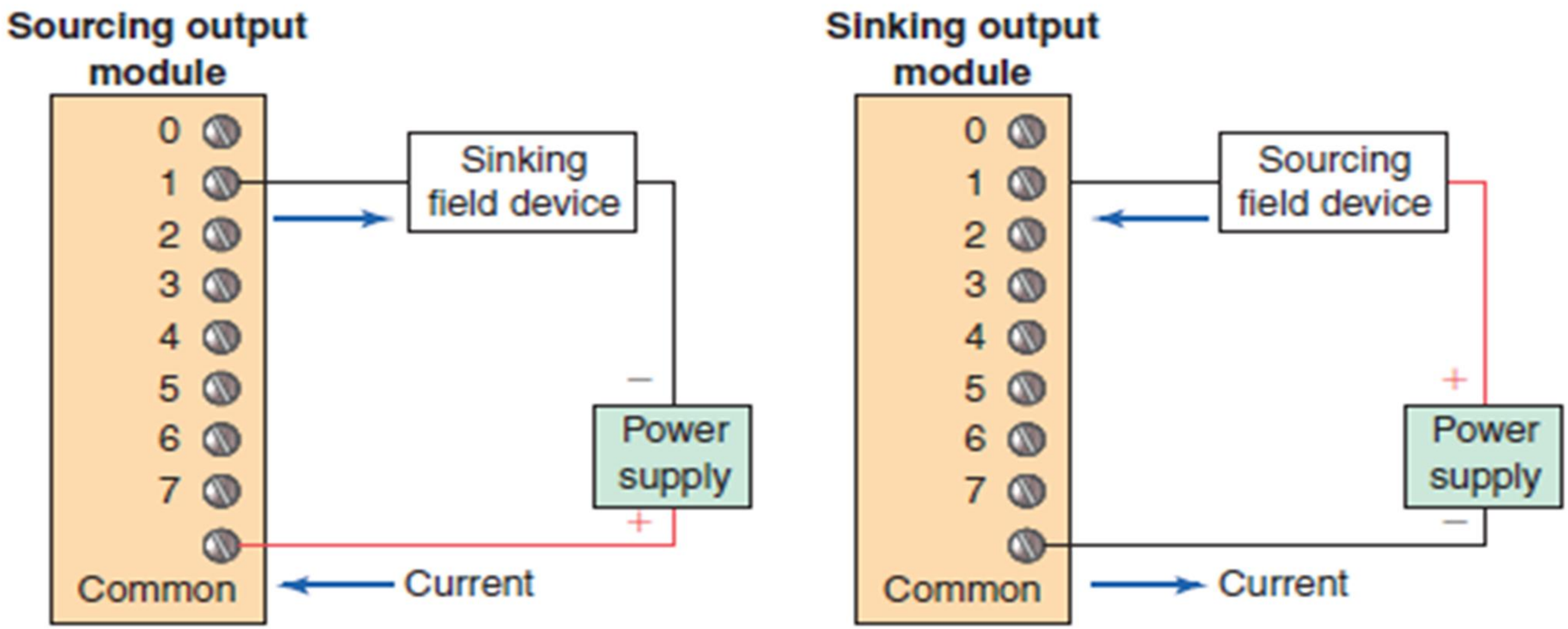
- If the module is a current-sourcing module, then the input or output device must be a current-sinking device.
- If the module is specified as current sinking, then the connected device must be current sourcing.
- In general, sinking (NPN) and sourcing (PNP) are terms used to describe a current signal flow relationship between field input and output devices in a control system and their power supply.

Sourcing and Sinking Input Devices



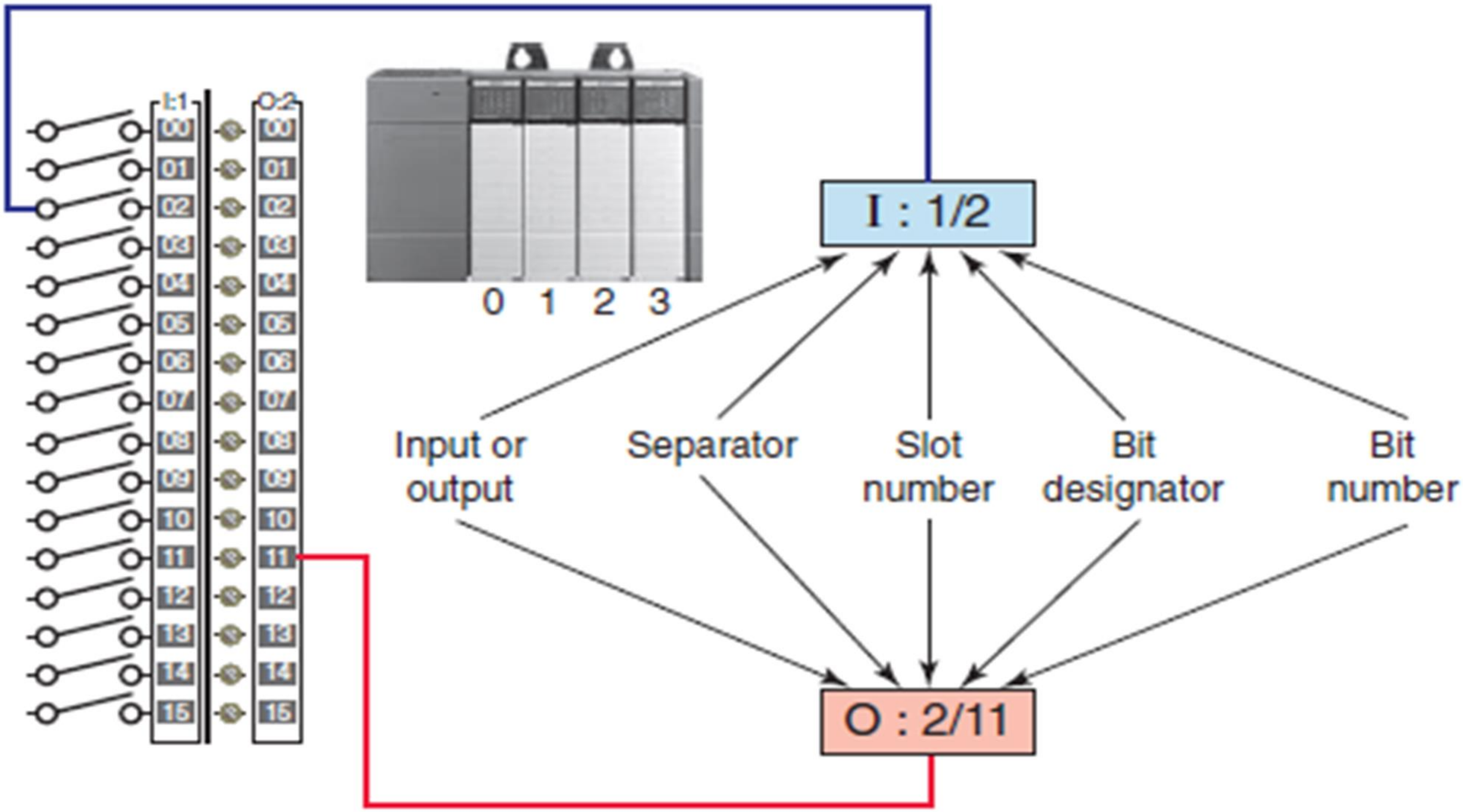
- Field devices connected to the positive (+) side of the field power supply are classified as sourcing field devices.

Sourcing and Sinking Output Devices



- Field devices connected to the negative (-) side or DC common of the field power supply are sinking field devices

I/O Addressing



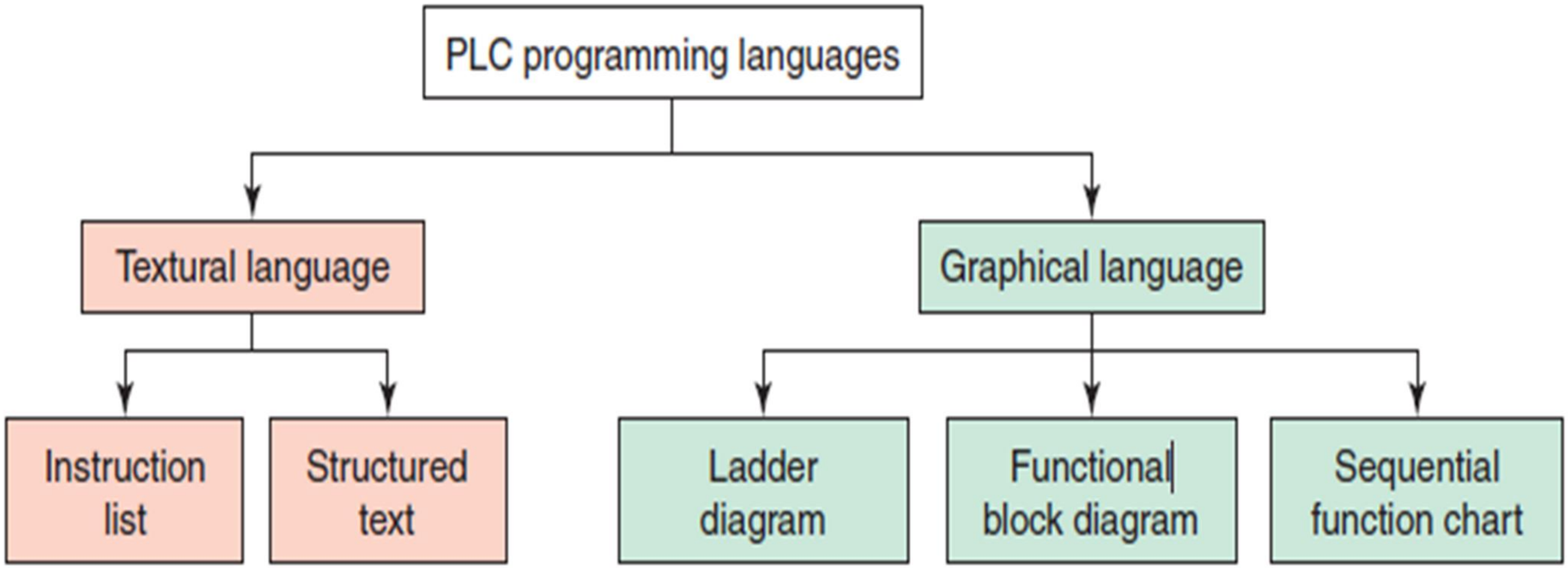
I/O Addressing

Format consists of the following three parts:

I:1/2 O:2/2

- **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.

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.

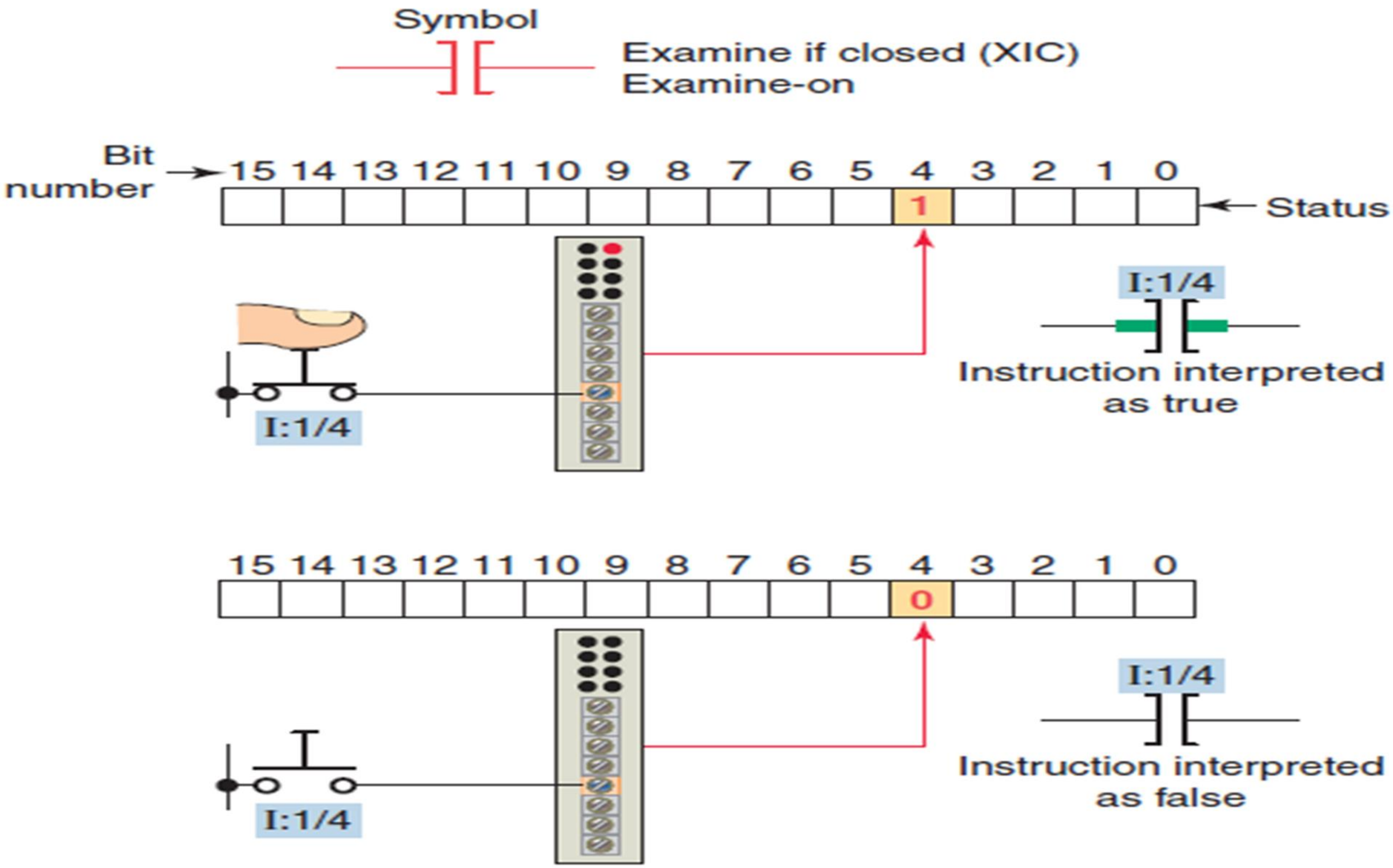
PLC Programming Languages

- **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.

Relay-Type Instructions

- Ladder diagram language is basically a symbolic set of instructions used to create the controller program.
- Ladder instruction symbols are arranged to obtain the desired control logic that is to be entered into the memory of the PLC.
- Three fundamental symbols that are used to translate relay control logic to contact symbolic logic are Examine If Closed (XIC), Examine If Open (XIO), and Output Energize (OTE), relates to a single bit of PLC memory.

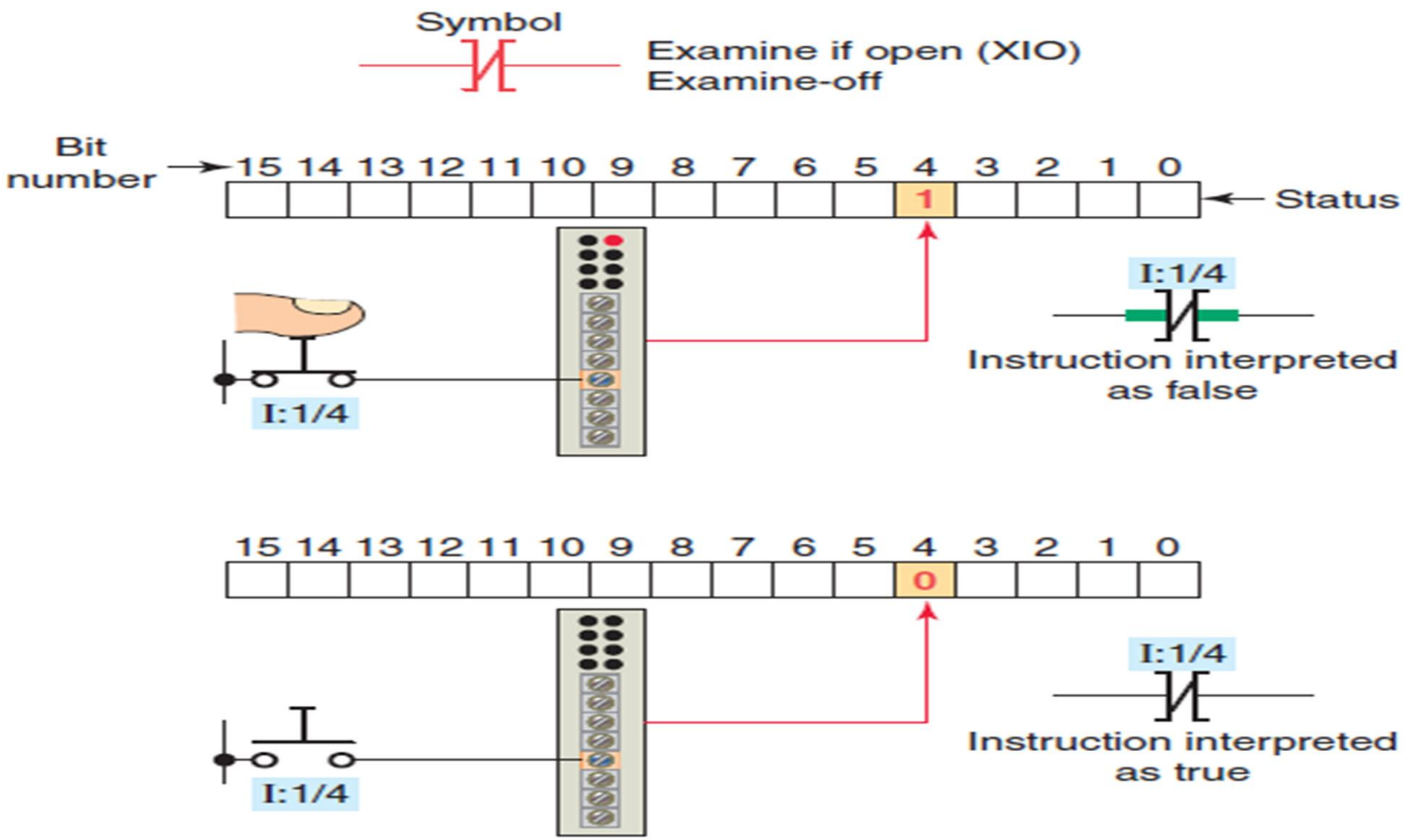
Examine If Closed (XIC)/Normally Open (NO)



Examine If Closed (XIC)/Normally Open (NO)

- It operates like a normally open relay contact.
- Associated with each XIC instruction is a memory bit linked to the status of an input device.
- A 1 corresponds to a true status or on condition.
- A 0 corresponds to a false status or off condition.
- Instruction will be set to 1 when a physical input is present (voltage is applied to the input terminal), and 0 when there is no physical input present (no voltage applied to the input terminal).

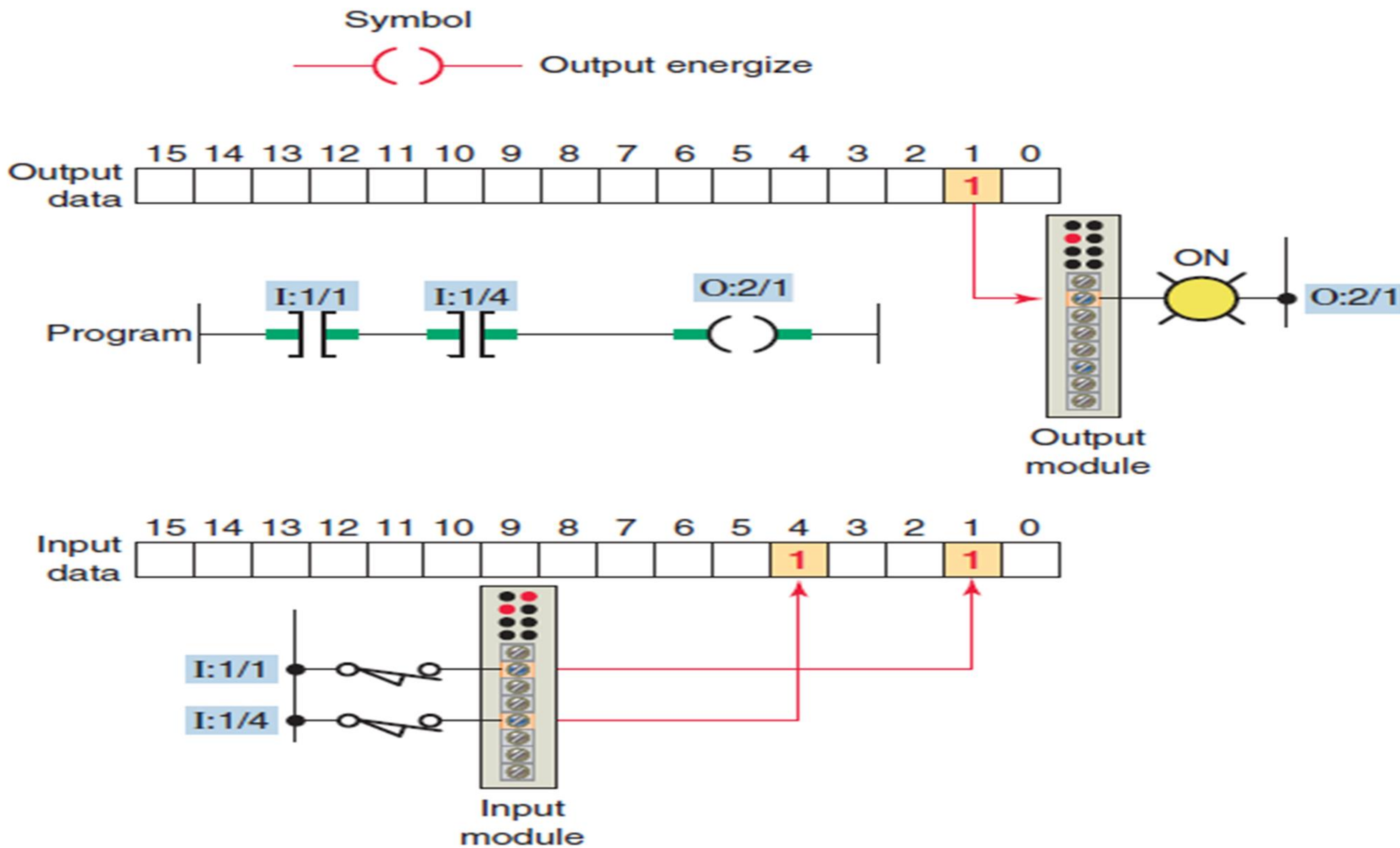
Examine If Open (XIO)/Normally Closed (NC)



Examine If Open (XIO)/Normally Closed (NC)

- It operates like a normally closed relay contact.
- Associated with each XIO instruction is a memory bit linked to the status of an input device.
- A 1 corresponds to a false status or off condition.
- A 0 corresponds to a true status or on condition.
- Instruction will be set to 0 when a physical input is present (voltage is applied to the input terminal), and 1 when there is no physical input present (no voltage applied to the input terminal).

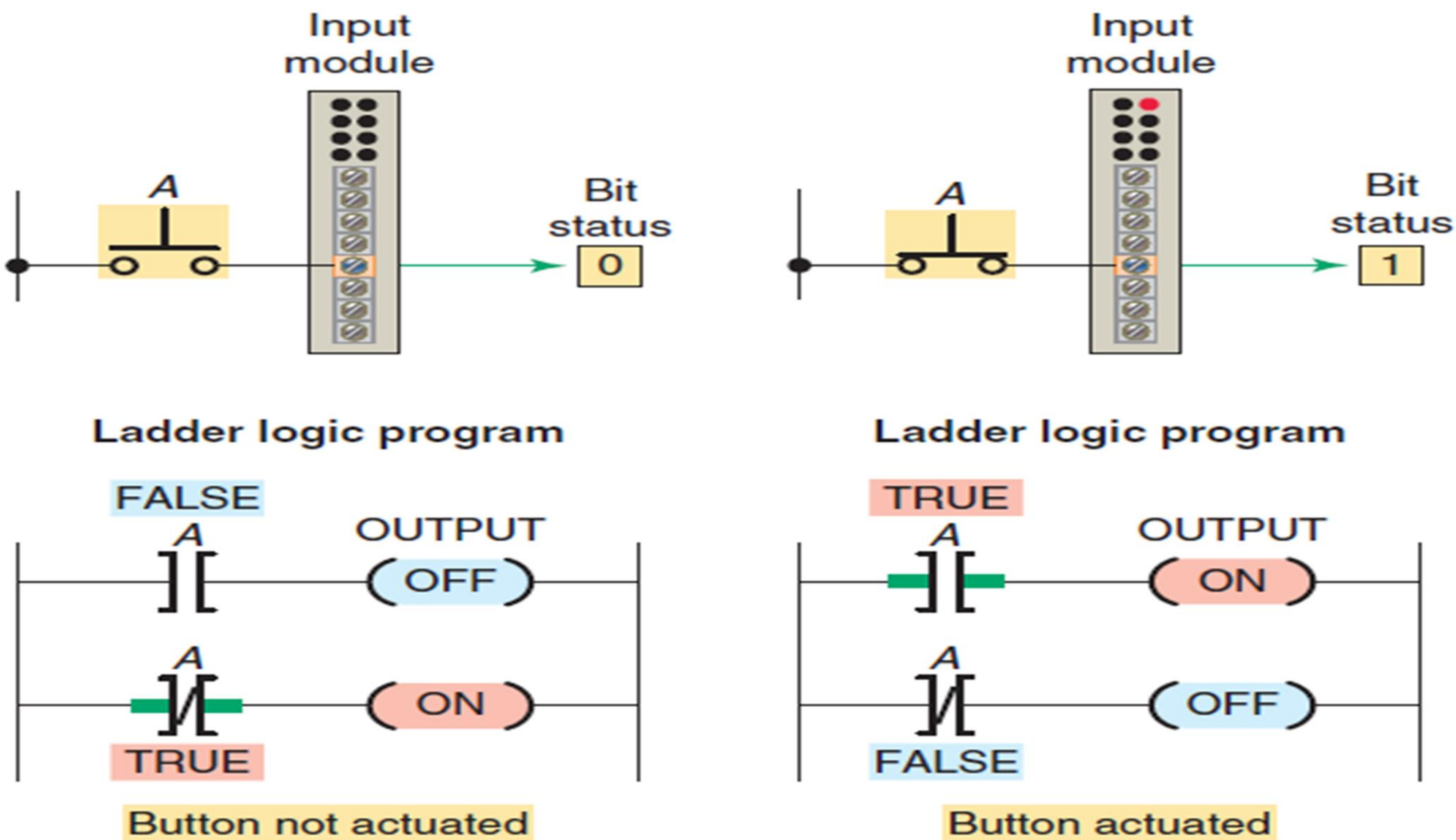
Output Energize (OTE)/Contact Coil



Output Energize (OTE)/Contact Coil

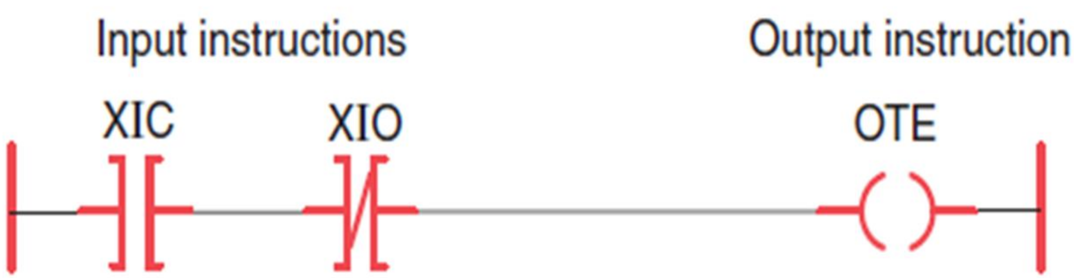
- It looks and operates like a relay coil and is associated with a memory bit.
- This instruction signals the PLC to energize (switch on) or de-energize (switch off) the output.
- The status bit of the addressed Output Energize instruction is set to 1 to energize the output and to 0 to de-energize the output.
- If a true logic path is established with the input instructions in the rung, the OTE instruction is energized and the output device wired to its terminal is energized and vice-versa.

Action of the field device and PLC bit



Program using both the XIC and XIO

If the data table bit is	The status of the instruction is		
	XIC EXAMINE IF CLOSED -] [-	XIO EXAMINE IF OPEN -]/[-	OTE OUTPUT ENERGIZE -( )-
Logic 0	False	True	False
Logic 1	True	False	True



Program using both the XIC and XIO

Time	Instruction outcome		
	XIC	XIO	OTE
t_1 (initial)	False	True	False
t_2	True	True	Goes true
t_3	True	False	Goes false
t_4	False	False	Remains false

Input bit status		
XIC	XIO	OTE
0	0	0
1	0	1
1	1	0
0	1	0

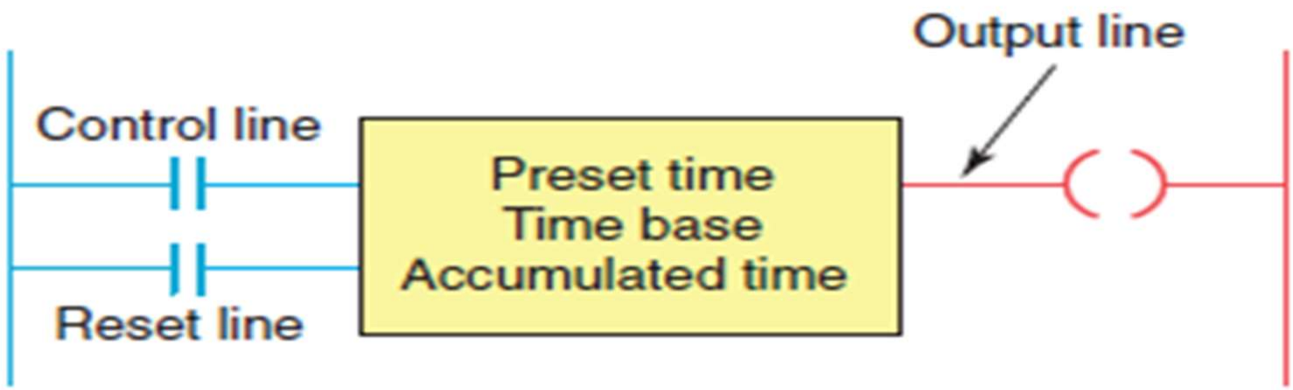
Timer Instructions

- Timer Instructions are used to delay the opening or closing of contacts for circuit control.
- It allow a multitude of operations in a control circuit to be automatically started and stopped at different time intervals
- TON (Timer On Delay) — Counts time-based intervals when the instruction is true.
- TOF (Timer Off Delay) — Counts time-based intervals when the instruction is false.

Timer Instructions

- RTO (Retentive Timer On) — Counts time-based intervals when the instruction is true and retains the accumulated value when the instruction goes false or when power cycle occurs.
- RES (Reset) — Resets a retentive timer's accumulated value to zero.
- Preset time — Represents the time duration for the timing circuit.
- Accumulated time — Represents the amount of time that has elapsed from the moment the timing coil became energized.
- Time base — Intervals that the timers time out. Timers can be programmed with several different time bases: 1 s, 0.1 s, and 0.01 s are typical time bases.

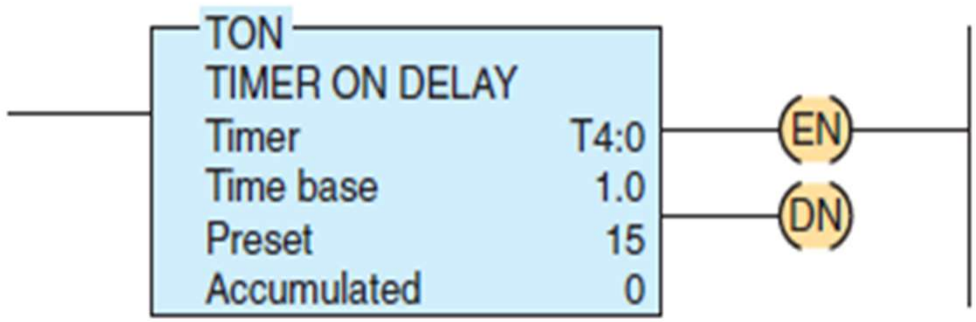
Block-formatted timer instruction.



- It provide at least one output signal from the timer.
- The timer continuously compares its current time with its preset time, and its output is false (logic 0) as long as the current time is less than the preset time.
- When the current time equals the preset time, the output changes to true (logic 1).

ON-Delay Timer Instruction

- ON-Delay timer is used when you want to program a time delay before an instruction becomes true.



- Timer number — Timer number is T4:0, which represents timer file 4, timer 0 in that file.
- Time base — The time base (which is always expressed in seconds) may be either 1.0 s or 0.1 s or 0.01 s.
- Preset value — Here preset value is 15.
- Accumulated value —Here, the accumulated value is 0.

ON-Delay Timer Instruction

- The preset value (PRE) word is the set point of the timer, that is, the value up to which the timer will time. It will not store a negative number.

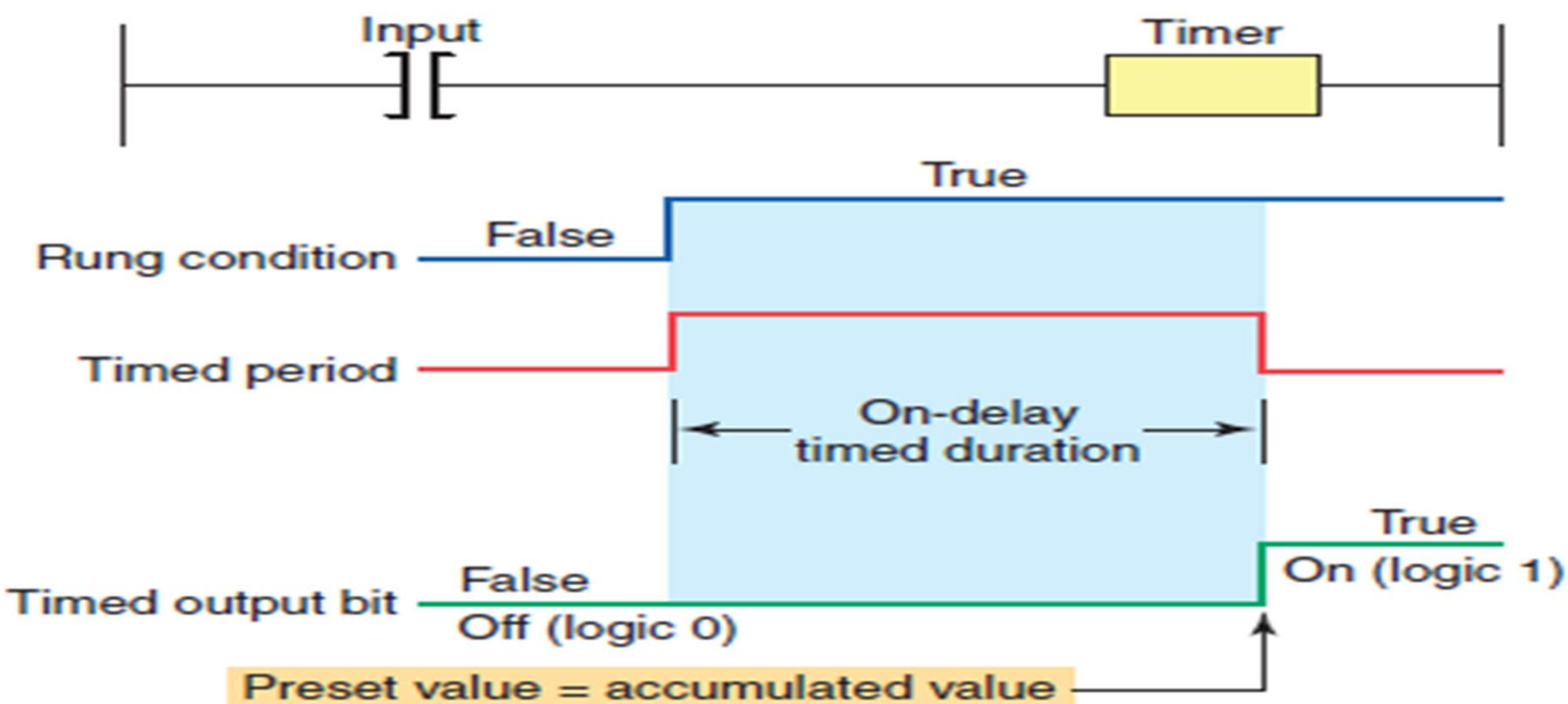
- The accumulated value (ACC) word is the value that increments as the timer are timing. The accumulated value will stop incrementing when its value reaches the preset value.

- Enable (EN) bit —The enable bit is true (has a status of 1) whenever the timer instruction is true. When the timer instruction is false, the enable bit is false (has a status of 0).

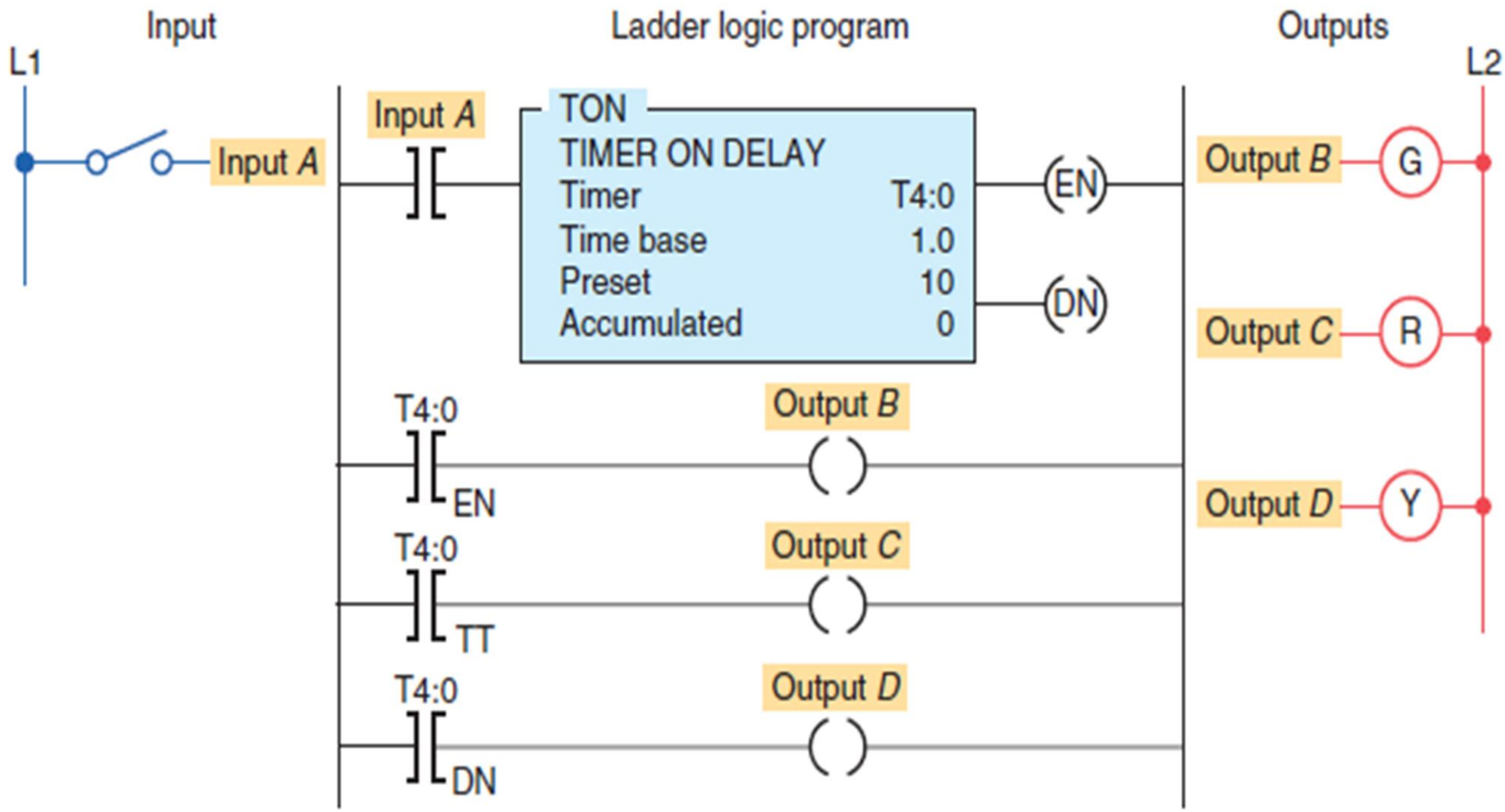
- Done (DN) bit —The done bit changes state whenever the accumulated value reaches the preset value. Its state depends on the type of timer being used.

ON-Delay Timer Instruction

- Timer-timing (TT) bit —The timer-timing bit is true whenever the accumulated value of the timer is changing, which means the timer is timing. When the timer is not timing, the accumulated value is not changing, so the timer-timing bit is false.



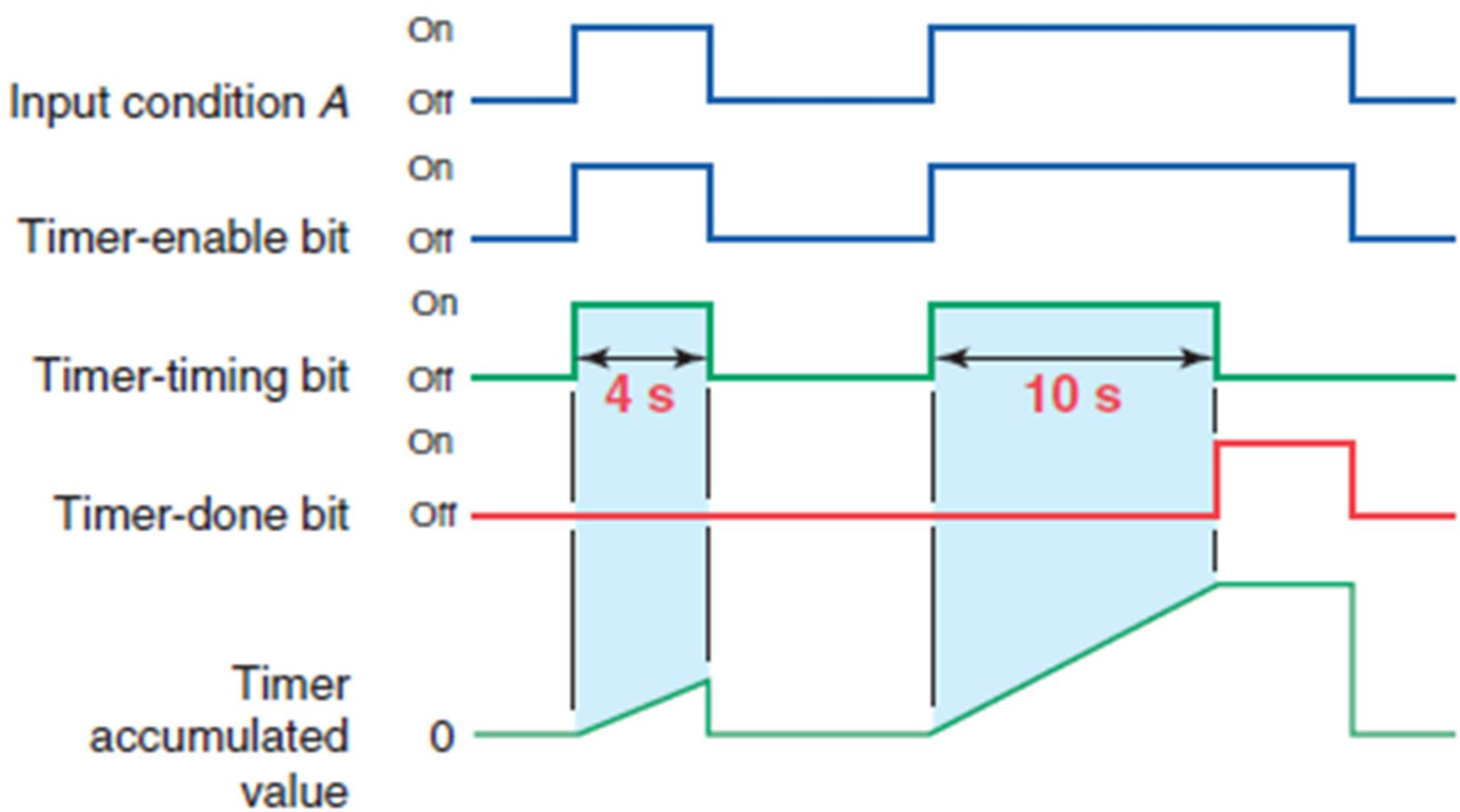
ON-Delay Timer Program



ON-Delay Timer Operation

- The preset time for this timer is 10 s, at which time output D will be energized.
- When input switch is A is closed, the timer becomes true and the timer begins counting and counts until the accumulated time equals the preset value; the output D is then energized.
- If the switch is opened before the timer is timed out, the accumulated time is automatically reset to 0.
- This timer configuration is termed non-retentive because any loss of continuity to the timer causes the timer instruction to reset.
- This timing operation is that of an on-delay timer because output D is switched on 10 s after the switch has been actuated from the off to the on position.

ON-Delay Timer Program

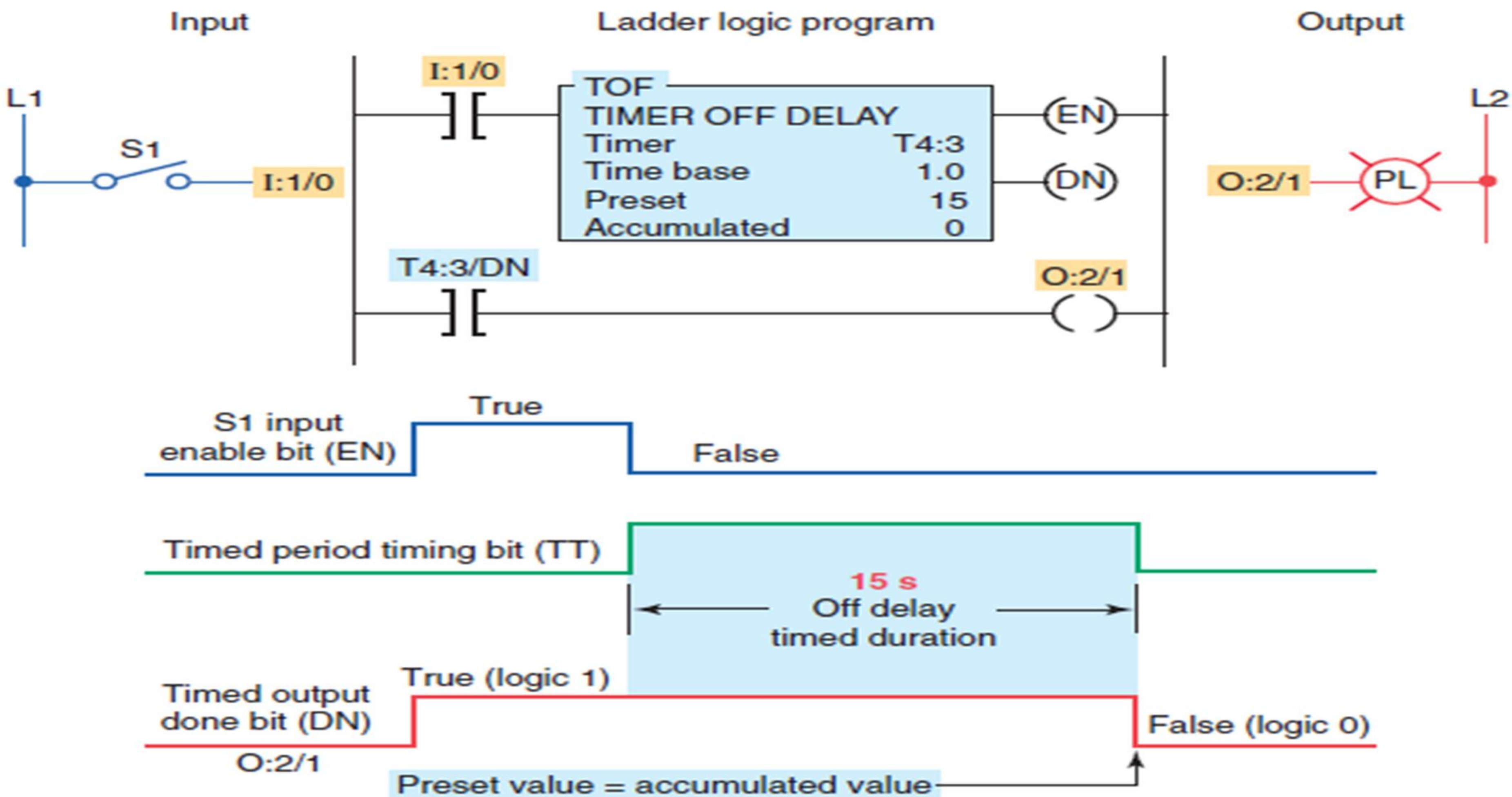


ON-Delay Timer Operation

- The first true period of the timer rung shows the timer timing to 4 s and then going false.
- The timer resets, and both the timer-timing bit and the enable bit go false. The accumulated value also resets to 0.
- For the second true period input A remains true in excess of 10 s.
- When the accumulated value reaches 10 s, the done bit (DN) goes from false to true and the timer timing bit (TT) goes from true to false.
- When input A goes false, the timer instruction goes false and also resets, at which time the control bits are all reset and the accumulated value resets to 0.

OFF-Delay Timer Operation

- OFF-Delay timer (TOF) operation will keep the output energized for a time period after the rung containing the timer has gone false.



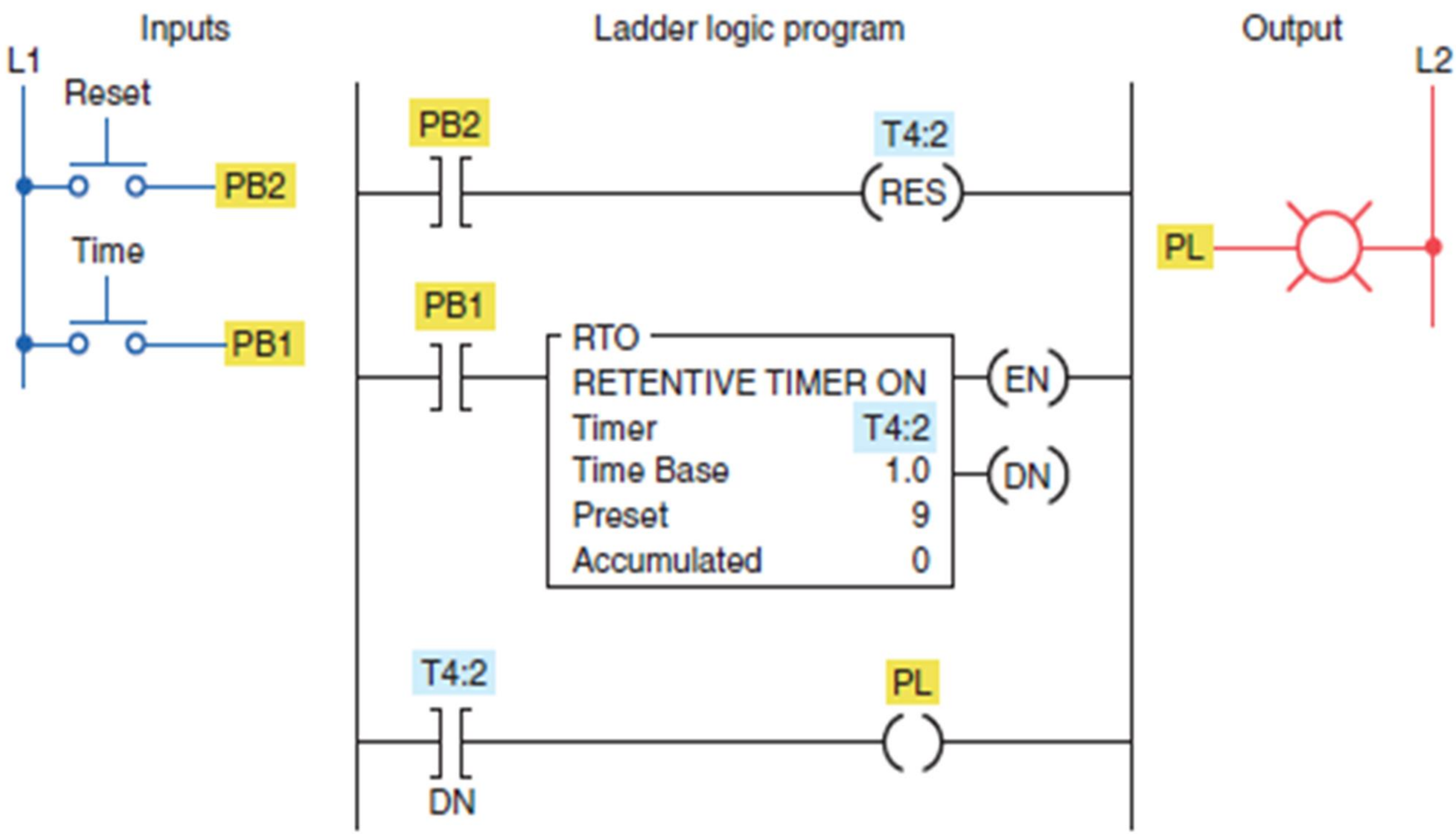
OFF-Delay Timer Operation

- When the switch connected to input I:1/0 is first closed, timed output O:2/1 is set to 1 immediately and the lamp is switched on.
- If this switch is now opened, logic continuity is lost and the timer begins counting.
- After 15 s, when the accumulated time equals the preset time, the output is reset to 0 and the lamp switches off.
- If logic continuity is gained before the timer is timed out, the accumulated time is reset to 0. For this reason, this timer is also classified as non-retentive.

Retentive Timer Operation

- Retentive timer accumulates time whenever the device receives power, and it maintains the current time when power is removed from the device.
- Loss of power to the timer after reaching its preset value does not affect the state of the contacts.
- Retentive timer must be intentionally reset with a separate signal for the accumulated time to be reset and for the contacts of the device to return to its non-energized state.
- Whenever the RES instruction is true, both the timer accumulated value and the timer done bit (DN) are reset to 0.

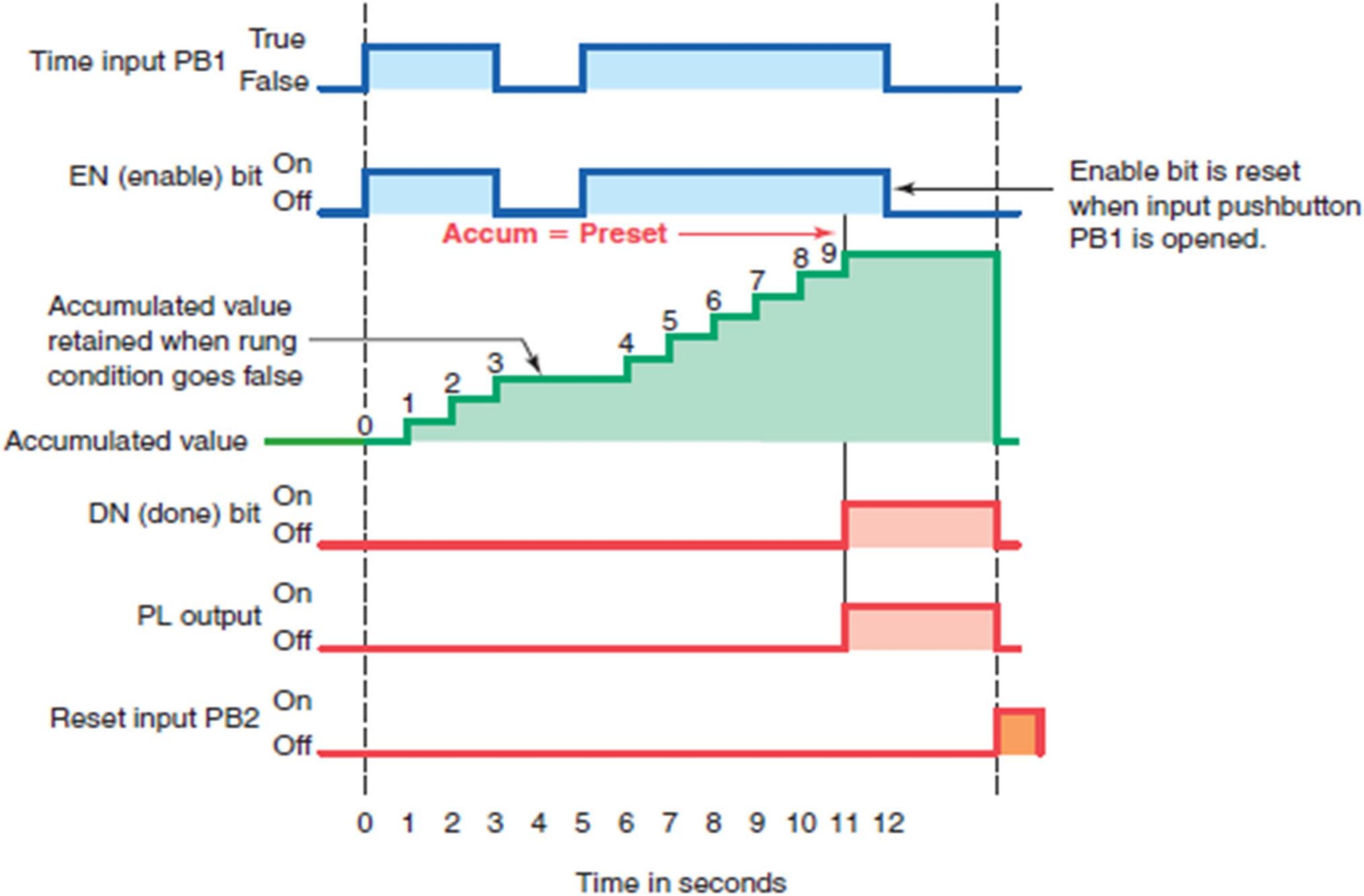
Retentive Timer Operation



Retentive Timer Operation

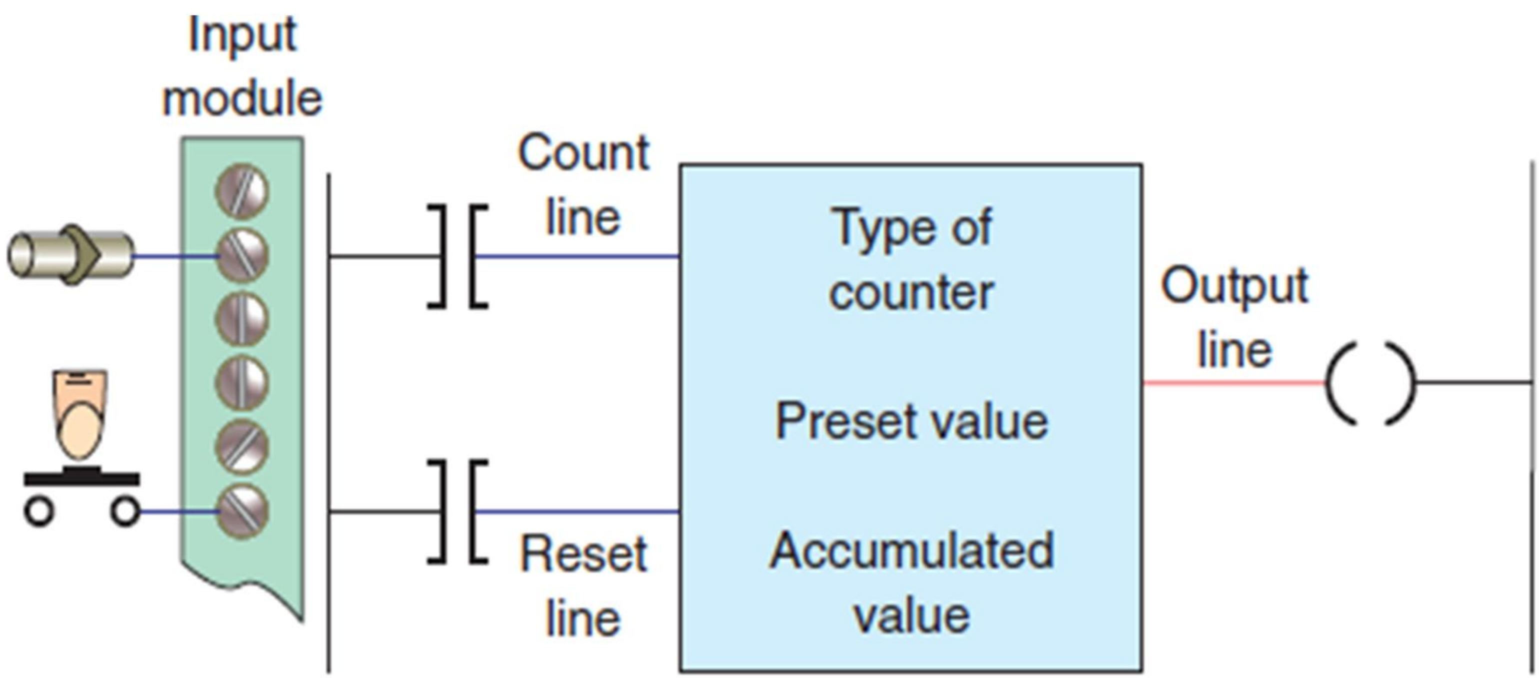
- When the timing rung is true (PB1 closed) the timer will commence timing.
- If the timing rung goes false the timer will stop timing but will recommence timing for the stored accumulated value each time the rung goes true.
- When the reset PB2 is closed, the T4:2/DN bit is reset to 0 and turns the pilot light output off. The accumulated value is also reset and held at zero until the reset pushbutton is opened.

Retentive Timer Operation



Counter Instruction

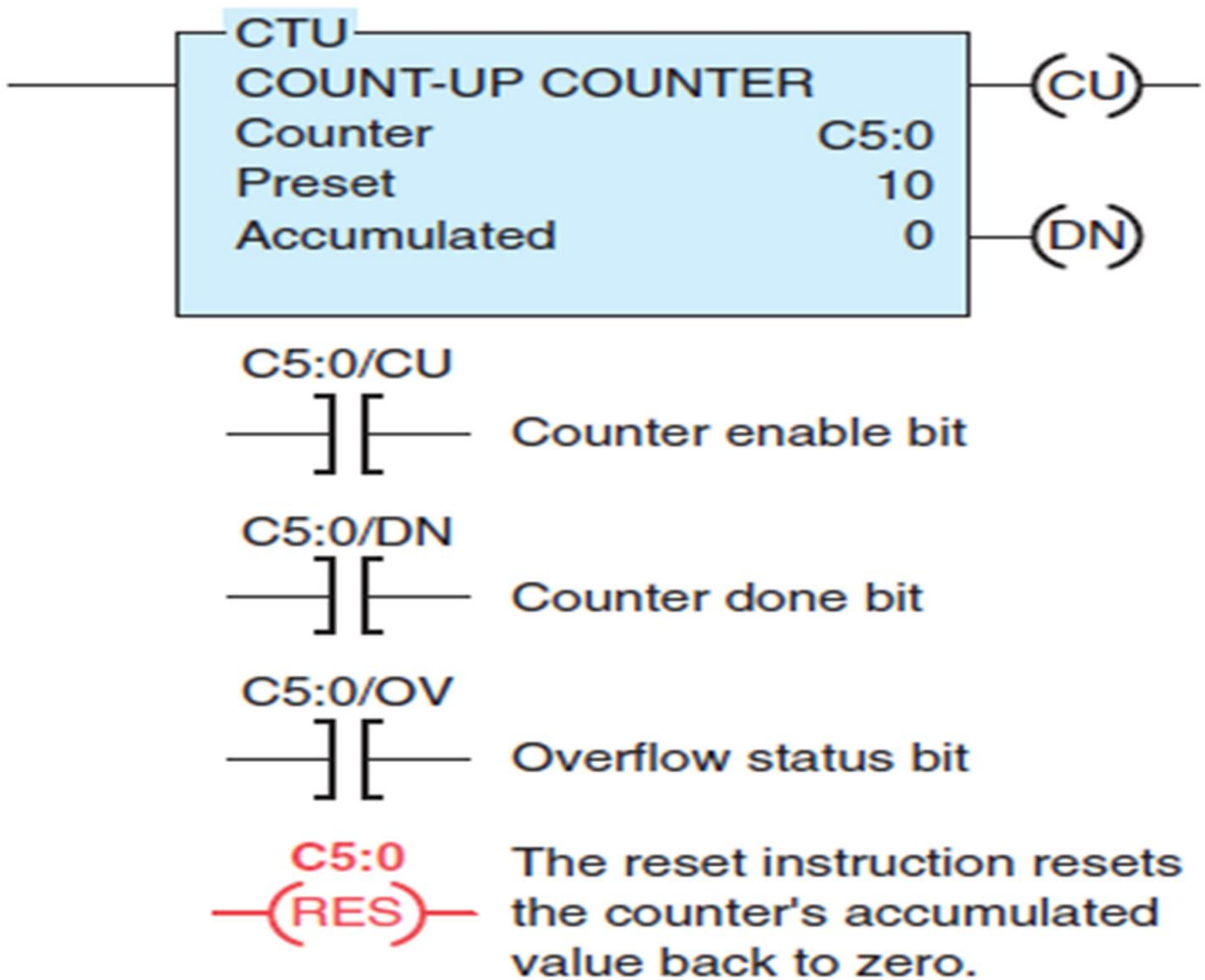
- Counters are similar to timers except that they do not operate on an internal clock but are dependent on external or program sources for counting.



Counter Instruction

- Counter has two input conditions associated with it, namely, the count and reset.
- The counter will either increment or decrement whenever the count input transfers from an off state to an on state.
- The counter will not operate on the trailing edge, or on-to-off transition, of the input condition.
- PLC counters are normally retentive.
- PLC counters can be designed to count up to a preset value or to count down to a preset value.

Up - Counter Instruction



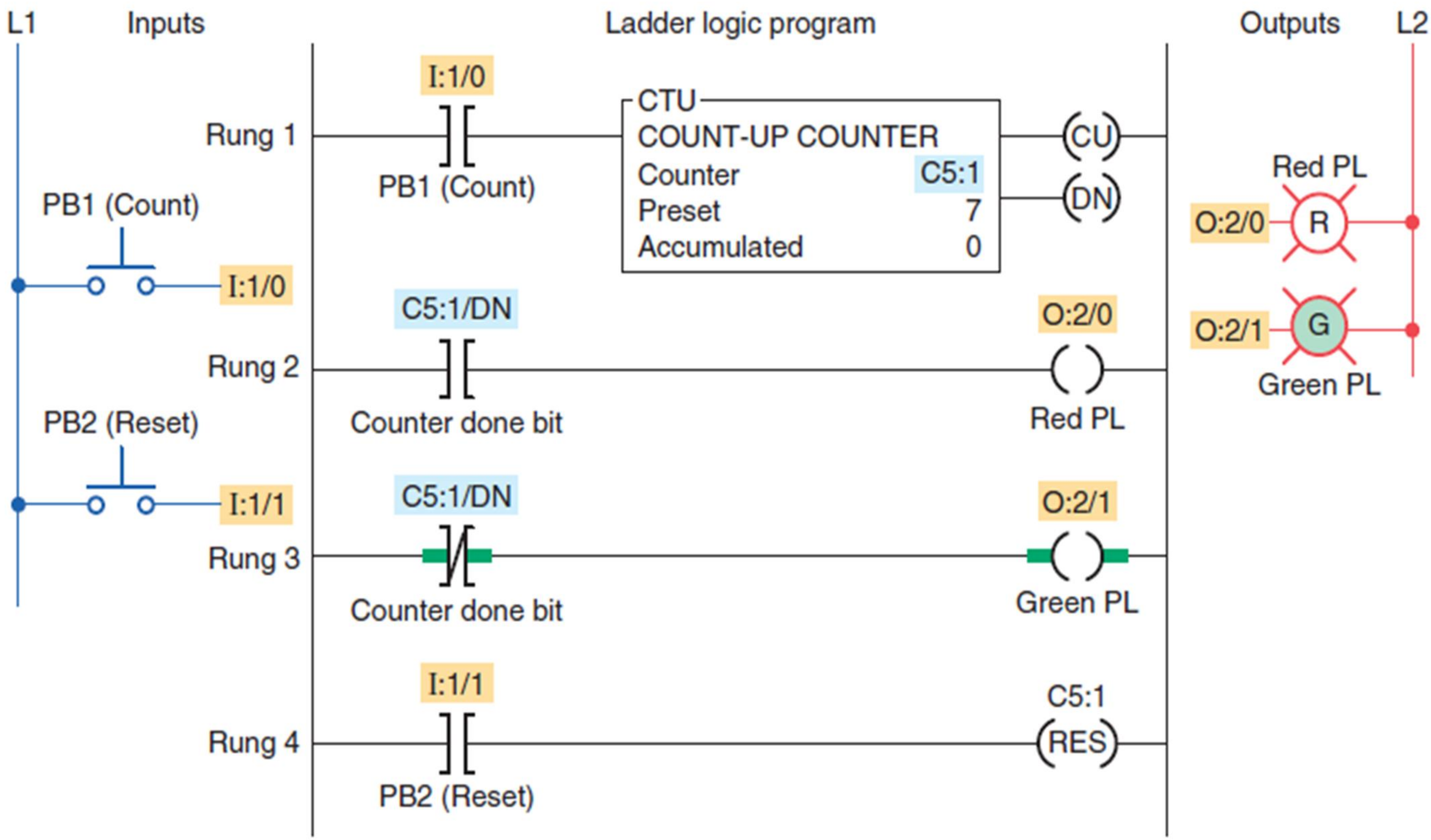
Up - Counter Instruction

- The up-counter is an output instruction whose function is to increment its accumulated value on false-to-true transitions of its instruction.

- It thus can be used to count false - to - true transitions of an input instruction and then trigger an event after a required number of counts or transitions.

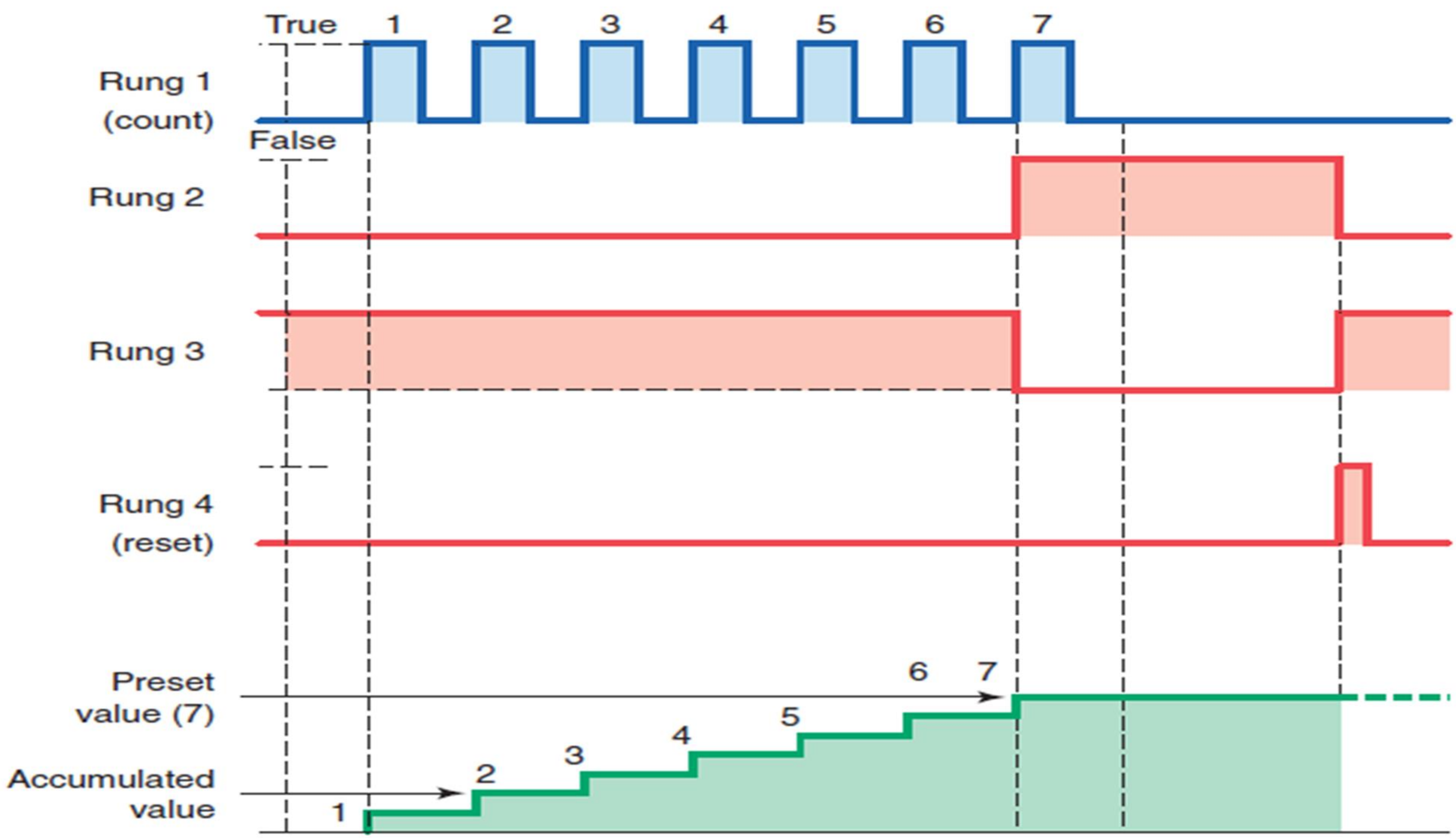
- The up-counter output instruction will increment by 1 each time the counted event occurs.

Up - Counter Instruction



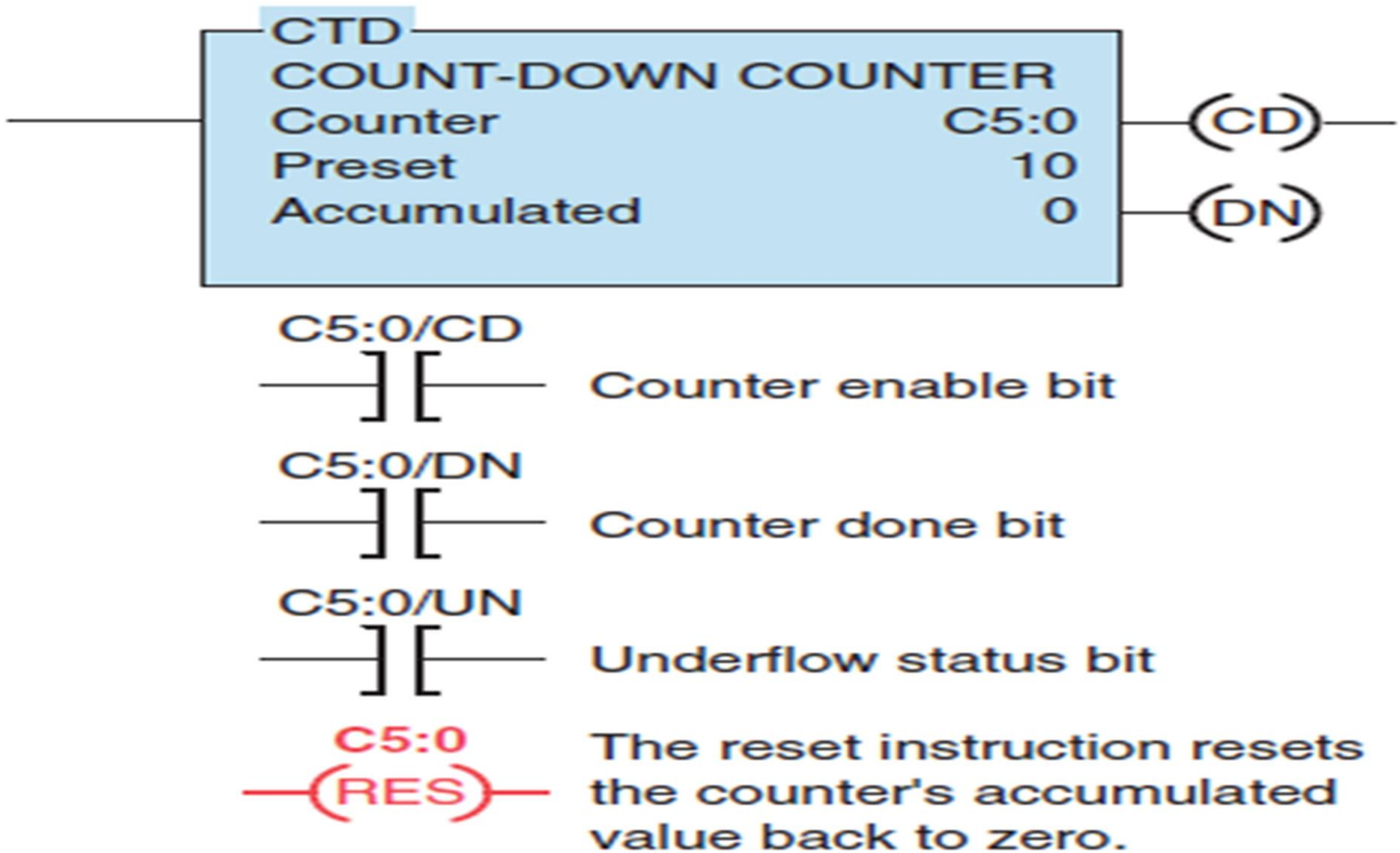
(a)

Up - Counter Instruction



(b)

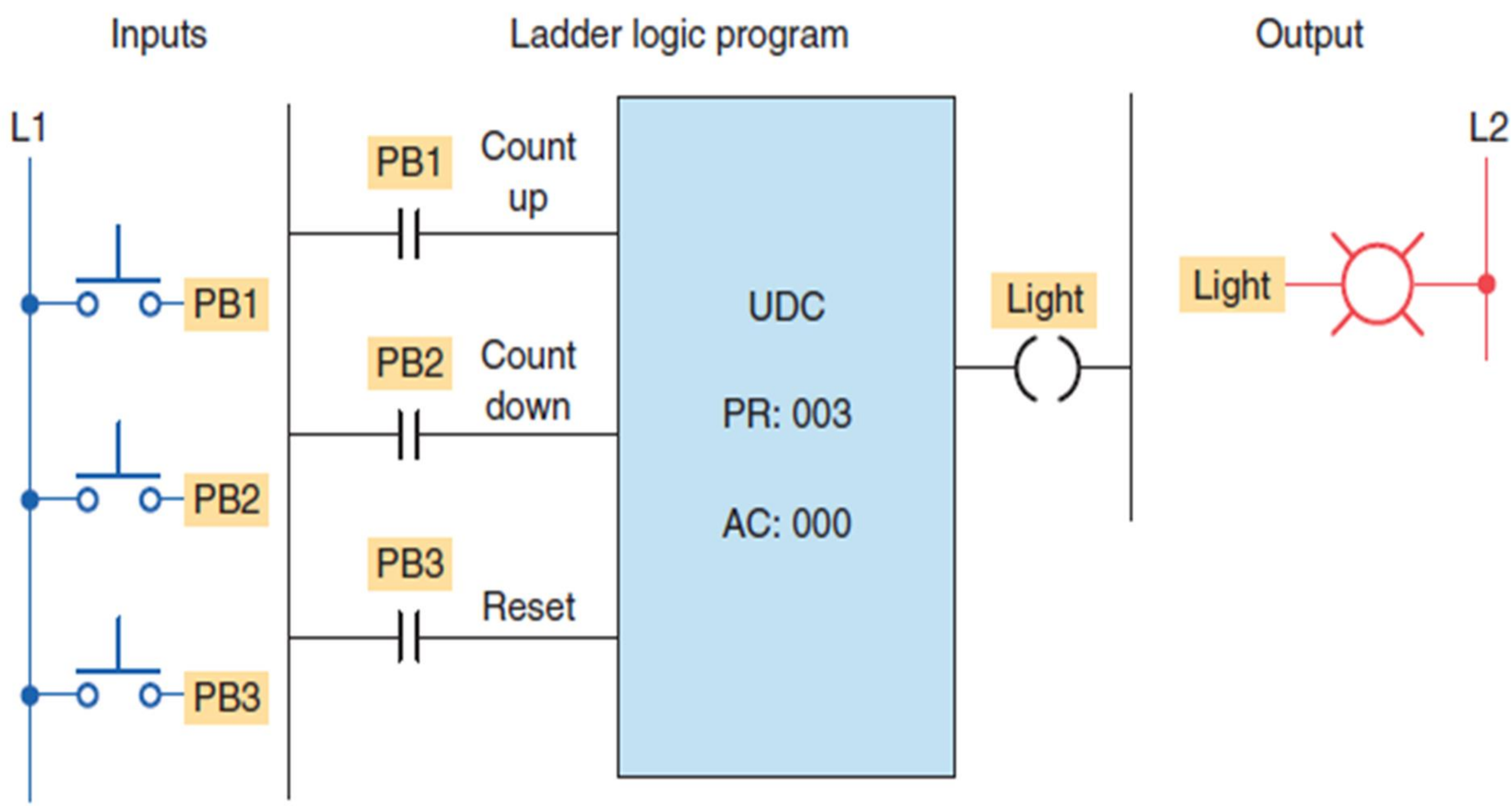
Down - Counter Instruction



Down - Counter Instruction

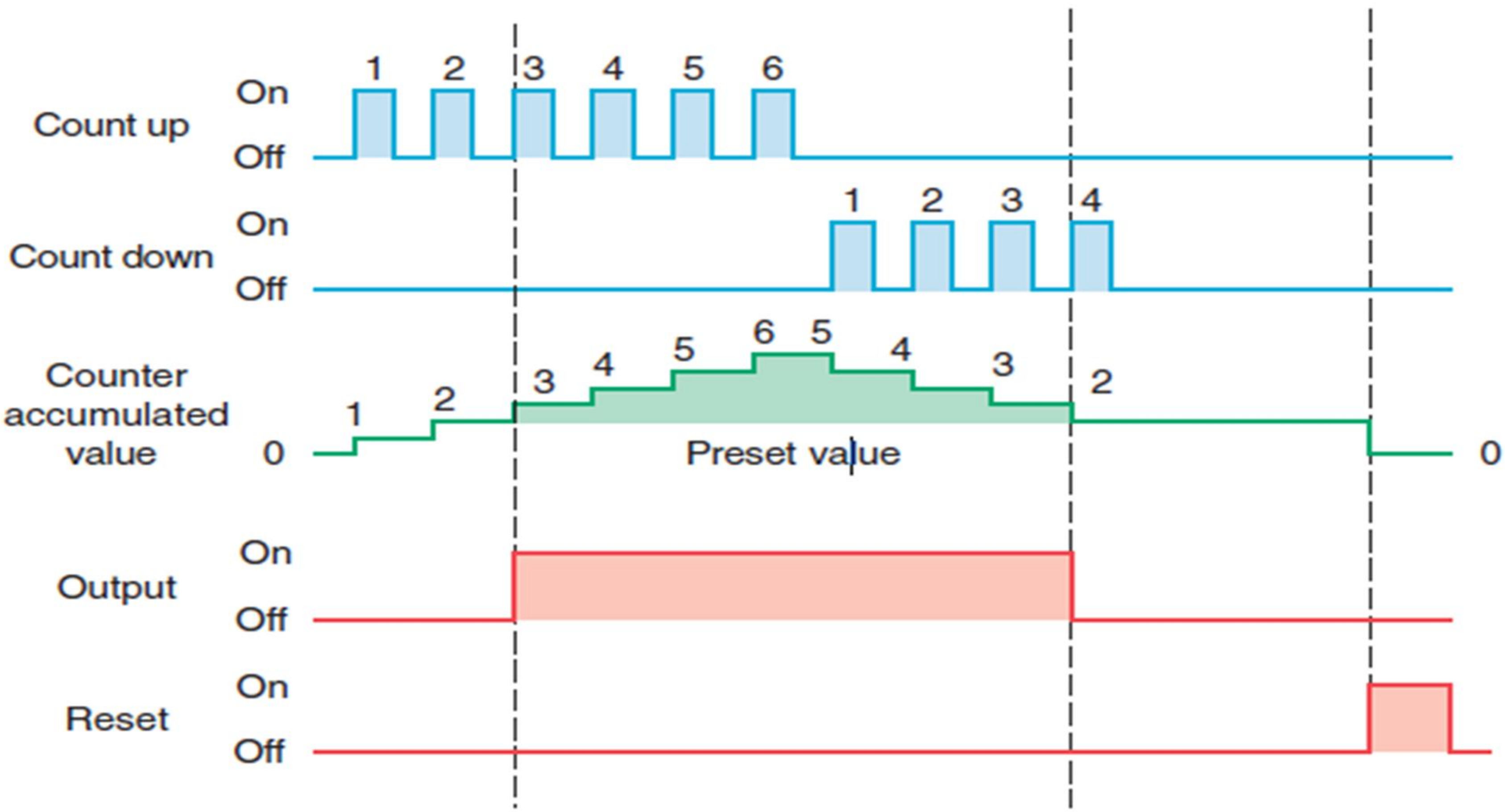
- The down-counter is an output instruction whose function is to decrement its accumulated value on false-to-true transitions of its instruction.
- It thus can be used to count false - to - true transitions of an input instruction and then trigger an event after a required number of counts or transitions.
- The down-counter output instruction will decrement by 1 each time the counted event occurs.

Down - Counter Instruction



(*a*)

Down - Counter Instruction



(b)

Data Manipulation:

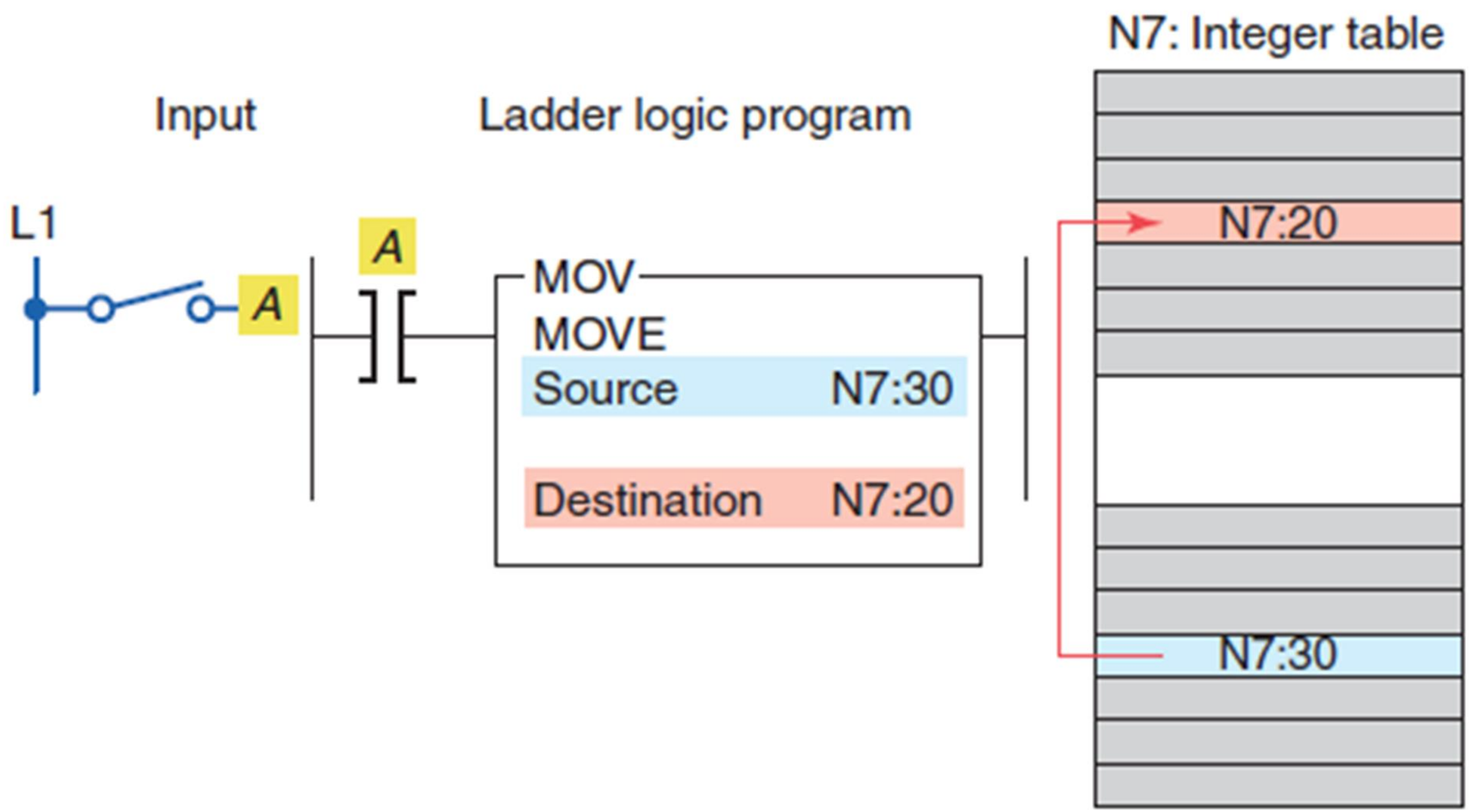
- Data manipulation instructions allow the movement, manipulation, or storage of data in either single- or multiple-word groups from one data memory area of the PLC to another.

- Data manipulation can be placed in two broad categories: data transfer and data comparison.

- Data transfer instructions simply involve the transfer of the contents from one word or register to another.

- MOV instruction is used to copy the value in one register or word to another.

Data Manipulation:



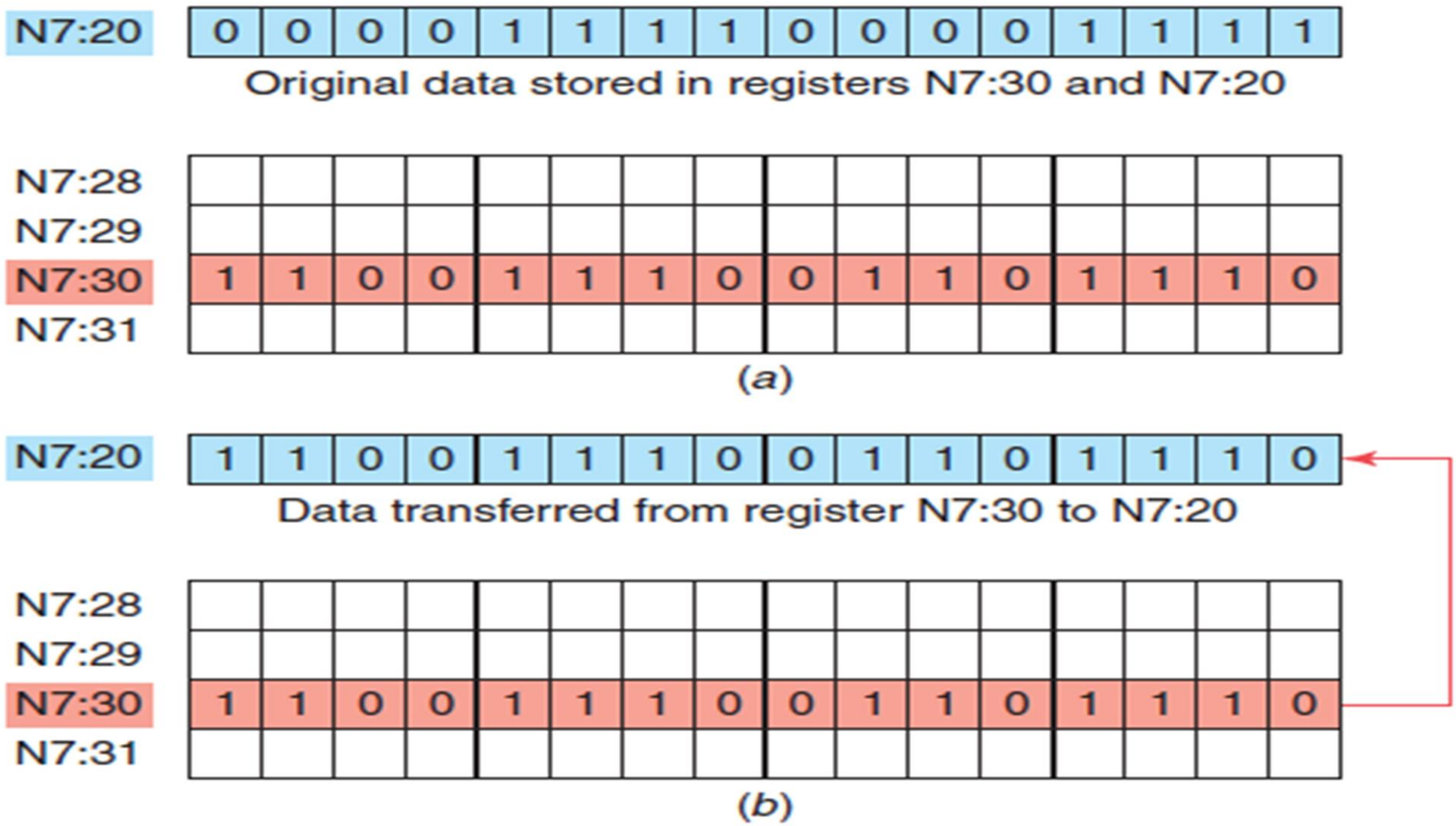
Data Manipulation:

- When the rung is true, input switch A closed, the value stored at the source address, N7:30, is copied into the destination address, N7:20.

- When the rung goes false, input switch A opened, the destination address will retain the value unless it is changed elsewhere in the program.

- The source value remains unchanged and no data conversion occurs.

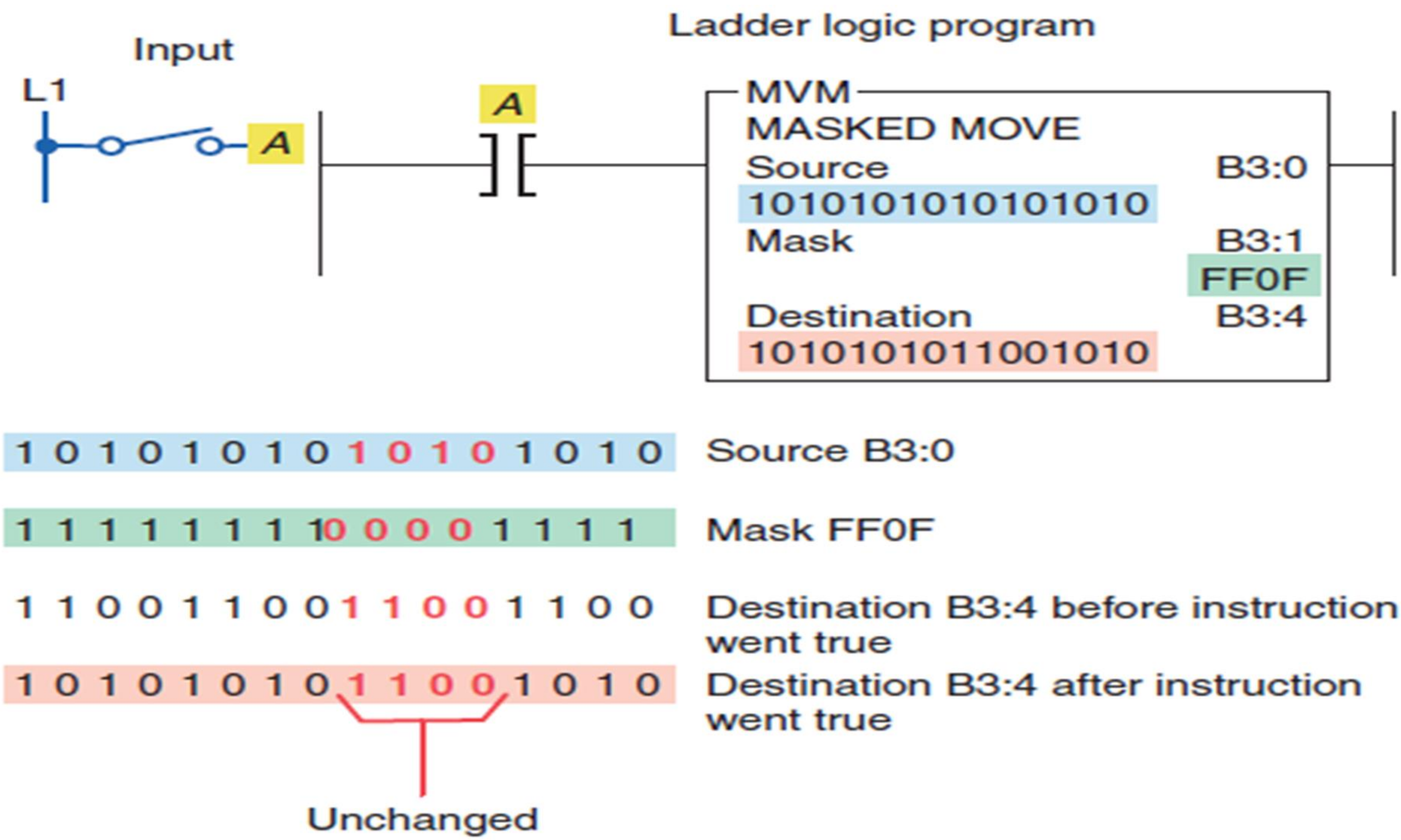
Data Manipulation:



Data Manipulation:

- Move with mask (MVM) instruction differs slightly from the MOV instruction because a mask word is involved in the move.
- Data being moved must pass through the mask to get to their destination address.
- Masking refers to the action of hiding a portion of a binary word before transferring it to the destination address.

Data Manipulation:



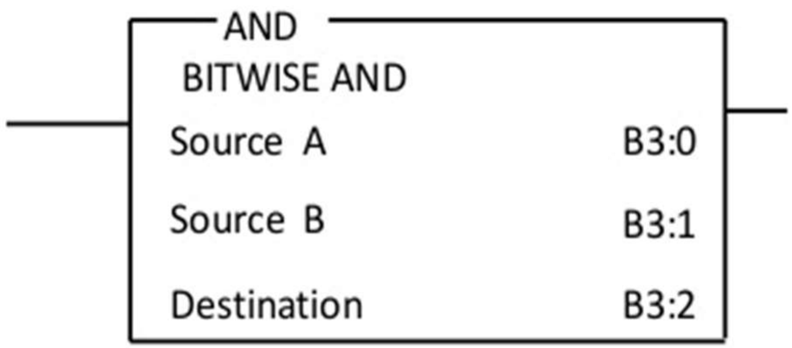
Data Manipulation:

- Mask may be entered as an address or in hexadecimal format, and its value will be displayed in hexadecimal.
- Where there is a 1 in the mask, data will pass from the source to the destination.
- Where there is a 0 in the mask, data in the destination will remain in their last state.
- Status in bits 4–7 are unchanged due to zeroes in the mask (remained in their last state).
- Status in bits 0–3 and 8–15 were copied from the source to destination when the MVM instruction went true.
- The mask must be the same word size as the source and destination.

Logical Instruction:

- **AND Instruction:**

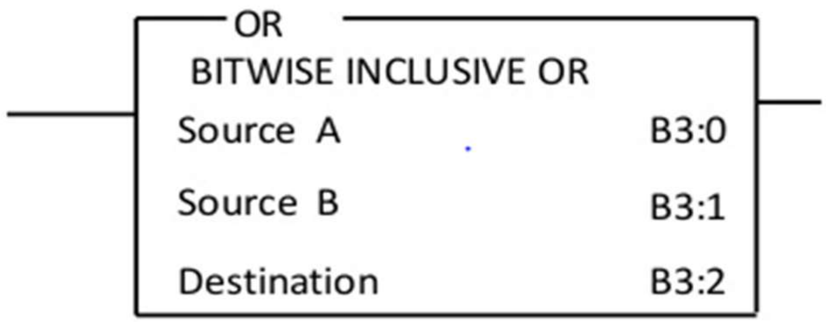
- AND command is used to perform logical AND instruction on each bit of value in source A with each bit of the value of source B, storing output logic in destination.



B3:0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
B3:1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
B3:2	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0

Logical Instruction:

- **OR Instruction:**
  - OR command is used to perform logical OR instruction on each bit of value in source A with each bit of the value of source B, storing output logic in destination.

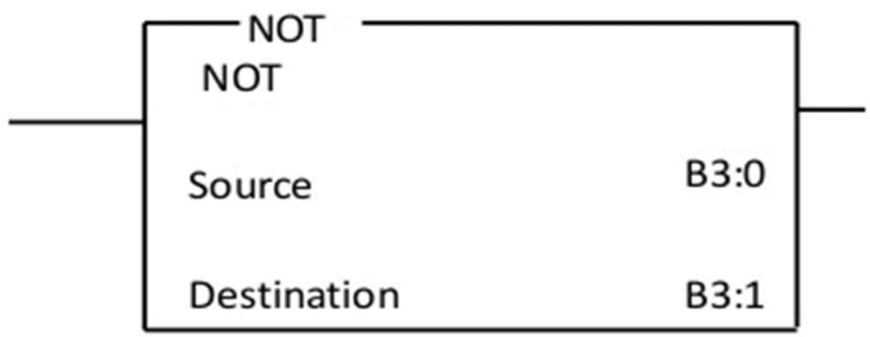


B3:0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
B3:1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
B3:2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0

Logical Instruction:

- **NOT Instruction:**

- NOT command is used to perform logical NOT instruction on each bit of value in source, storing output logic in destination.

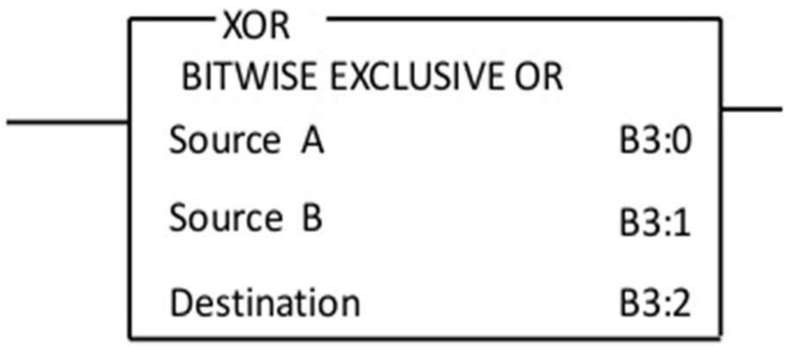


B3:0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
B3:1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1

Logical Instruction:

- **XOR Instruction:**

- XOR command is used to perform logical XOR instruction on each bit of value in source A with each bit of the value of source B, storing output logic in destination.

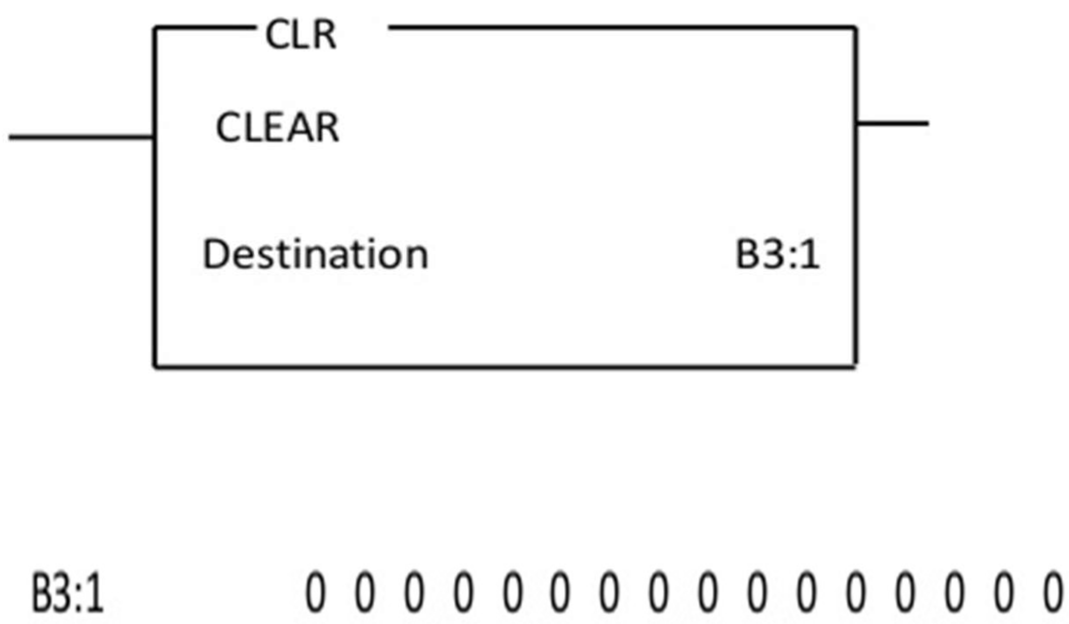


B3:0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
B3:1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
B3:2	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Logical Instruction:

- **CLR Instruction:**

- CLR instruction is to set destination value of word to zero.



Summary

- Discrete input & output modules.
- Analog input & output modules.
- I/O addressing of PLC.
- PLC instruction set.
- PLC programming examples