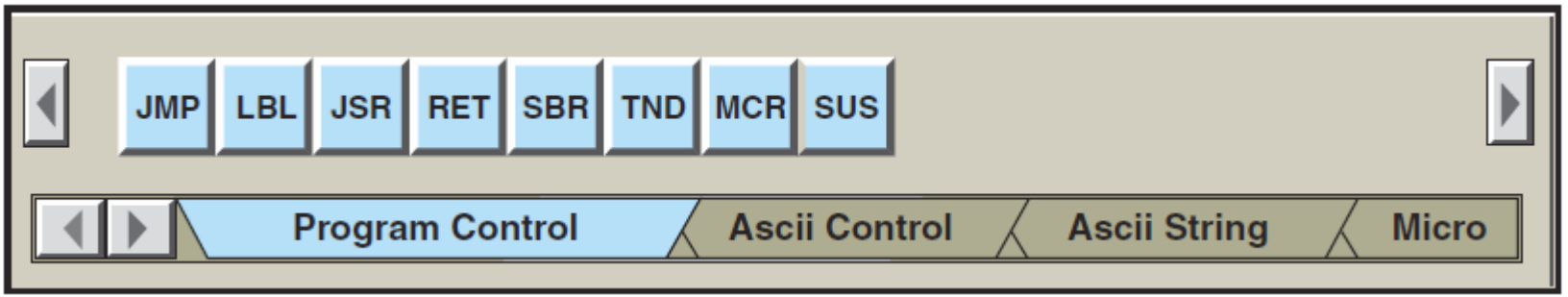
Chapter 9

Program Control Instructions

9.1

Master Control Reset Instruction

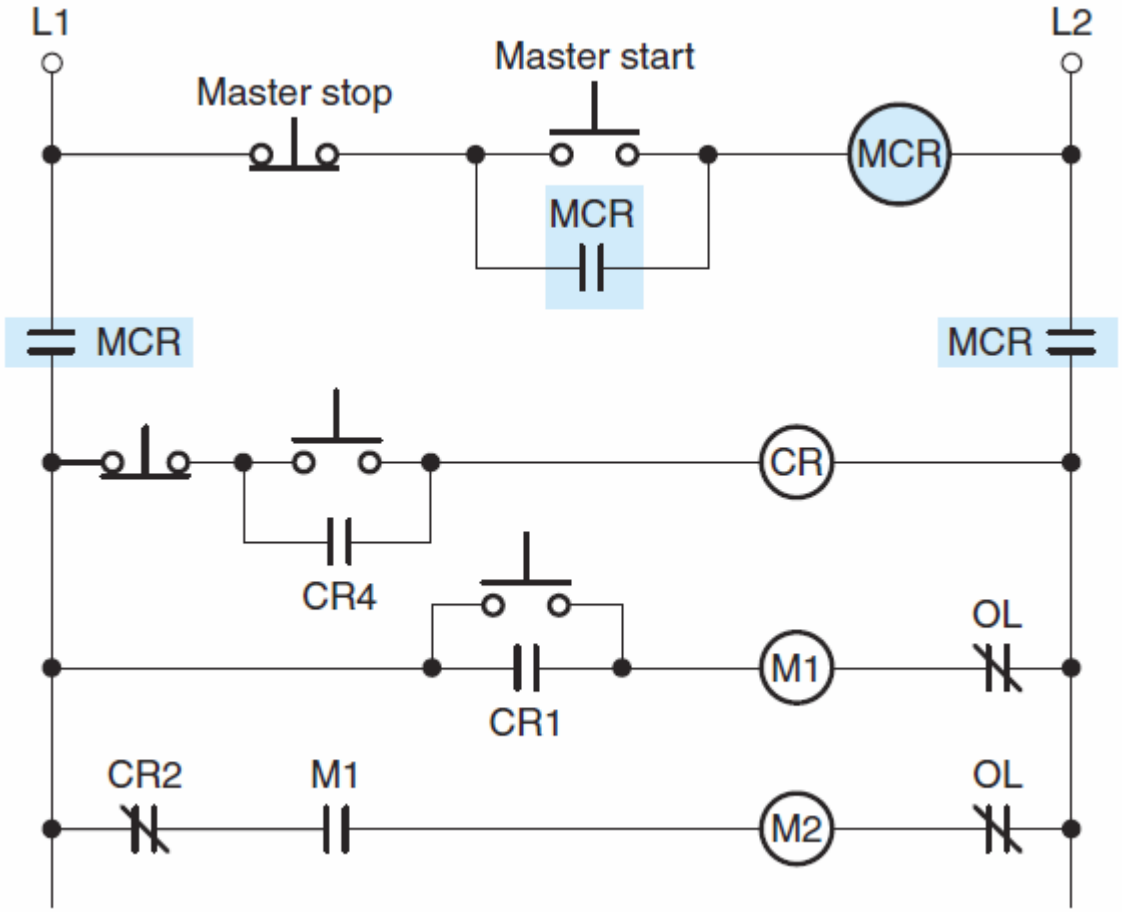
Program control instructions **are used to enable or disable a block of logic program or to move execution of a program from one place to another place.**



Program Control menu tab for the Allen-Bradley SLC 500 PLC

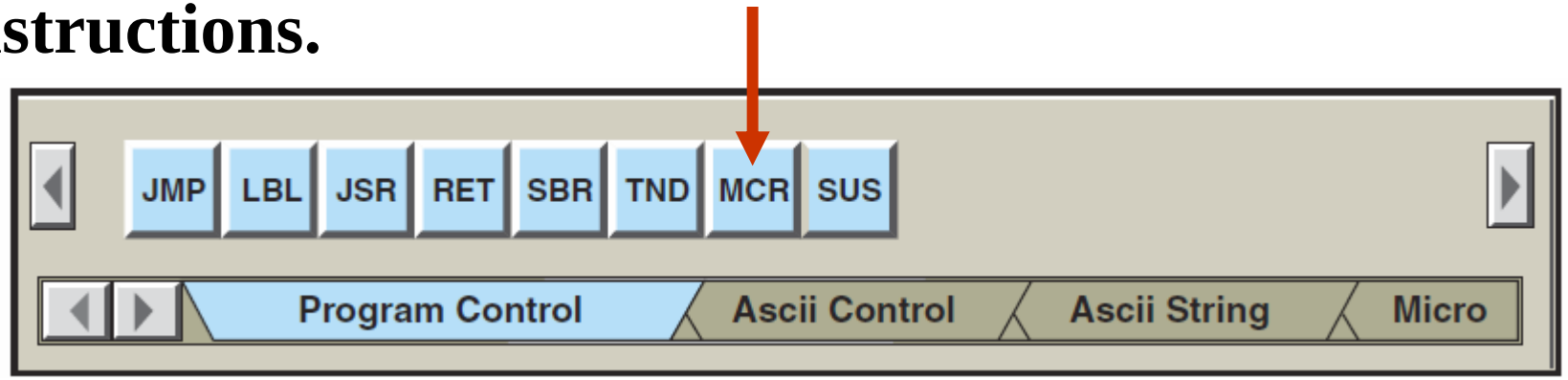
Hardwired master control relays **are used in relay control circuitry to provide input/output power shutdown of an entire circuit.**

Hardwired master control relay circuit.



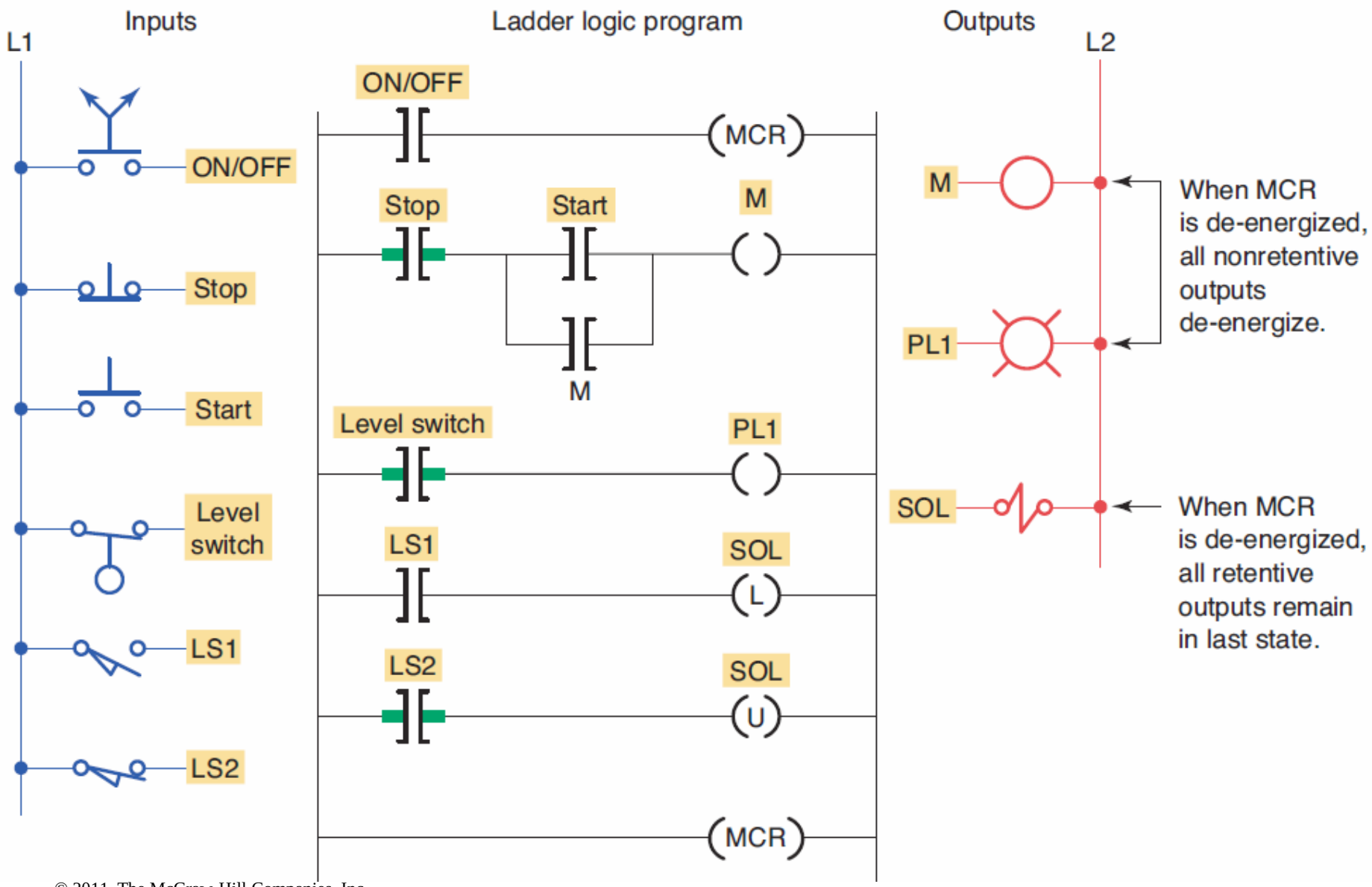
PLC manufacturers offer a form of a master control relay as part of their instruction set.

MCR (Master Control Reset) **-Clears all set nonretentive output rungs between the paired MCR instructions.**

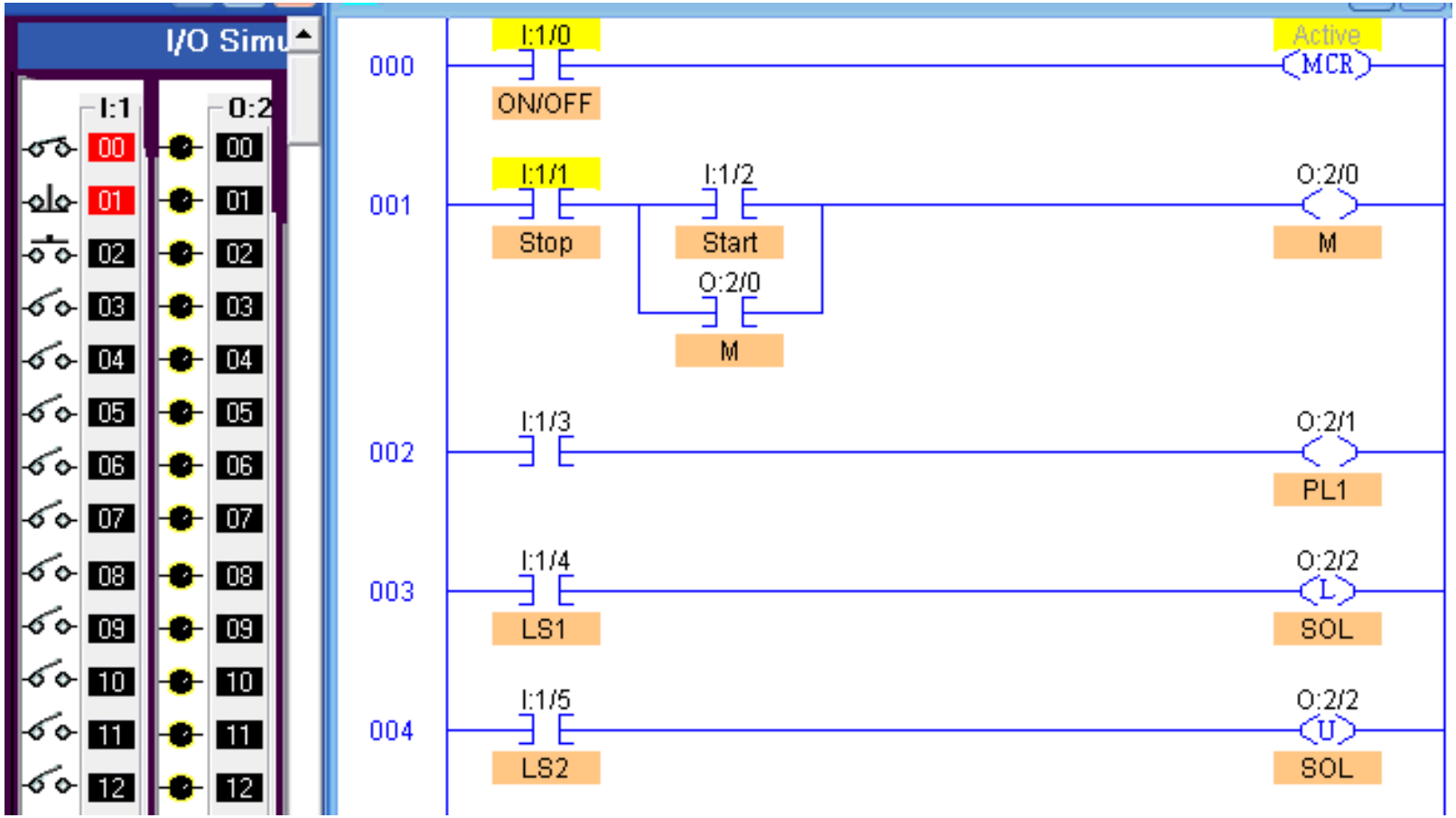


Because the MCR instruction is programmed, for safety reasons, it should not be used as a substitute for a hardwired master control relay, which provides *emergency* I/O power shutdown.

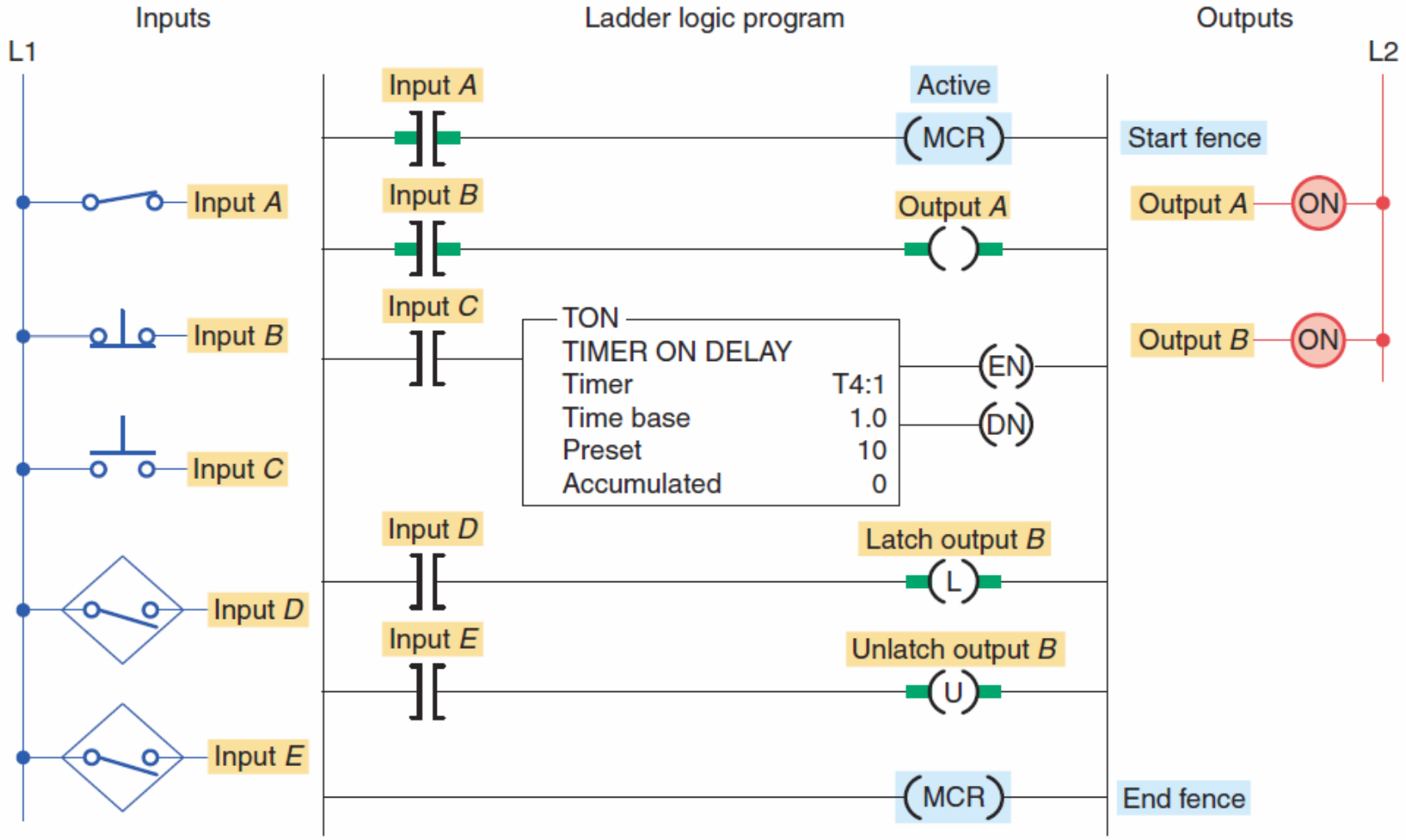
Programmed Master Control Reset instruction.



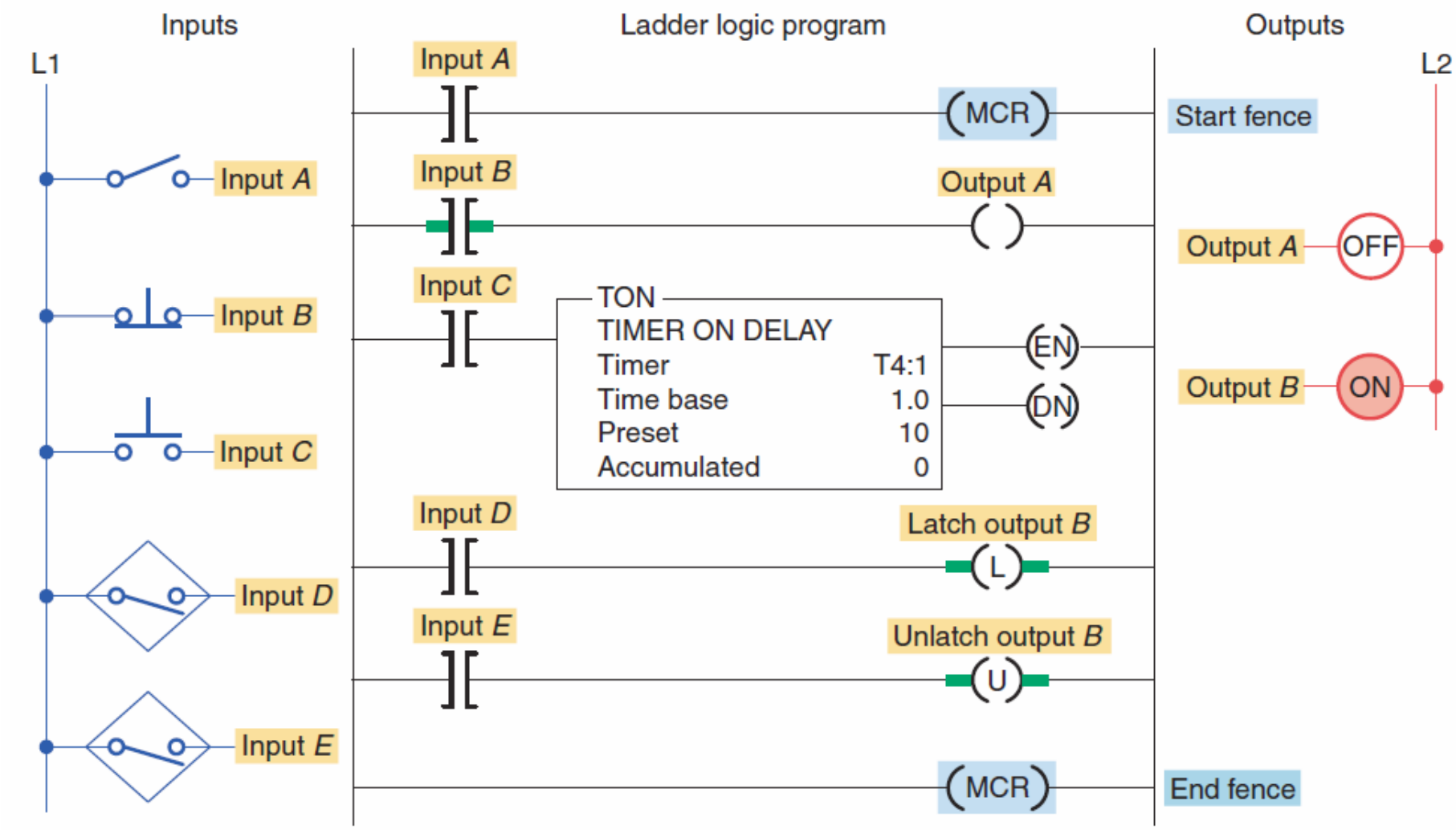
Simulated Master Control Reset instruction.



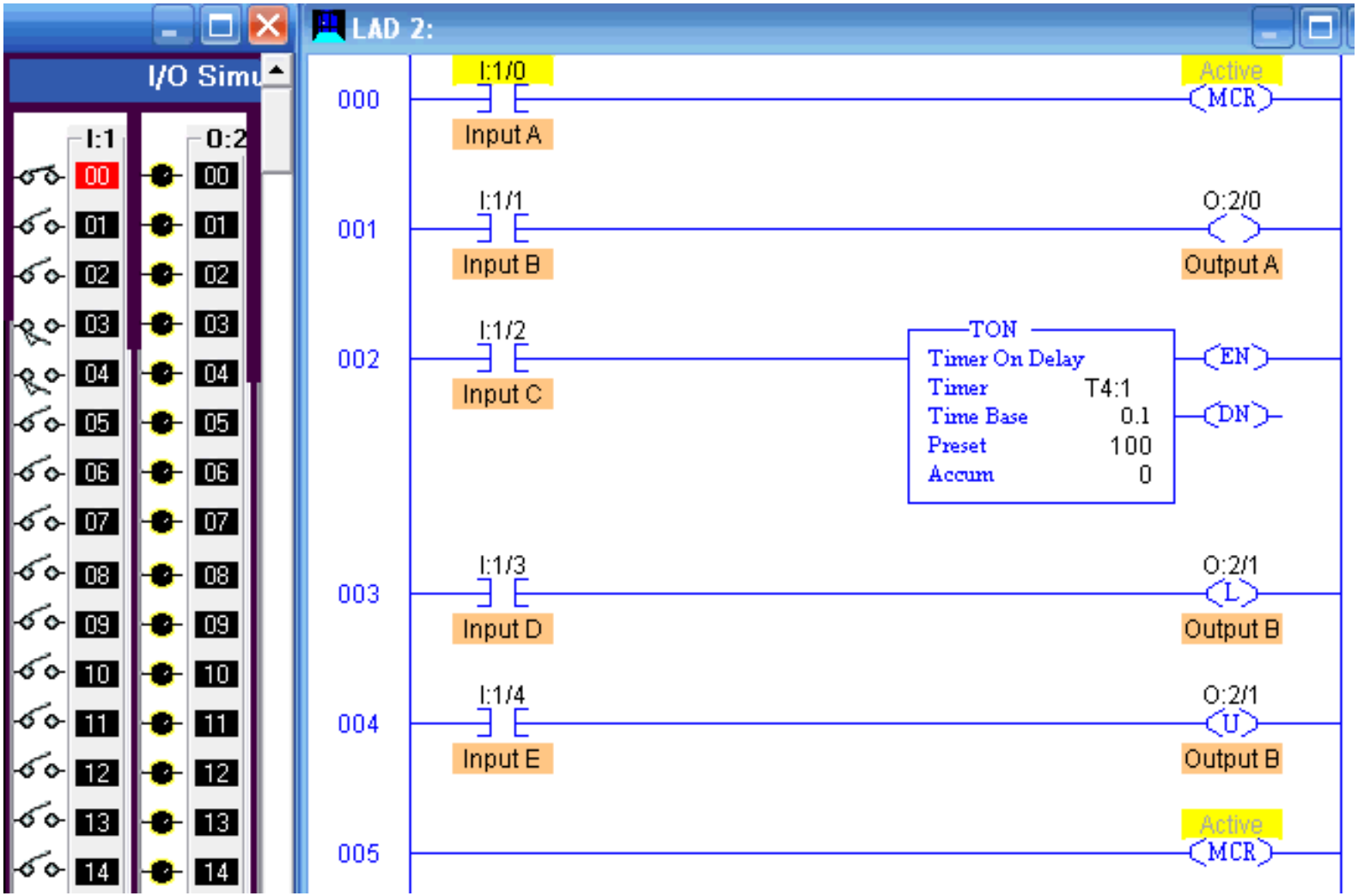
MCR *fenced zone* with the zone *true*.



MCR *fenced zone* with the zone *false*.



Simulated MCR fenced zone.

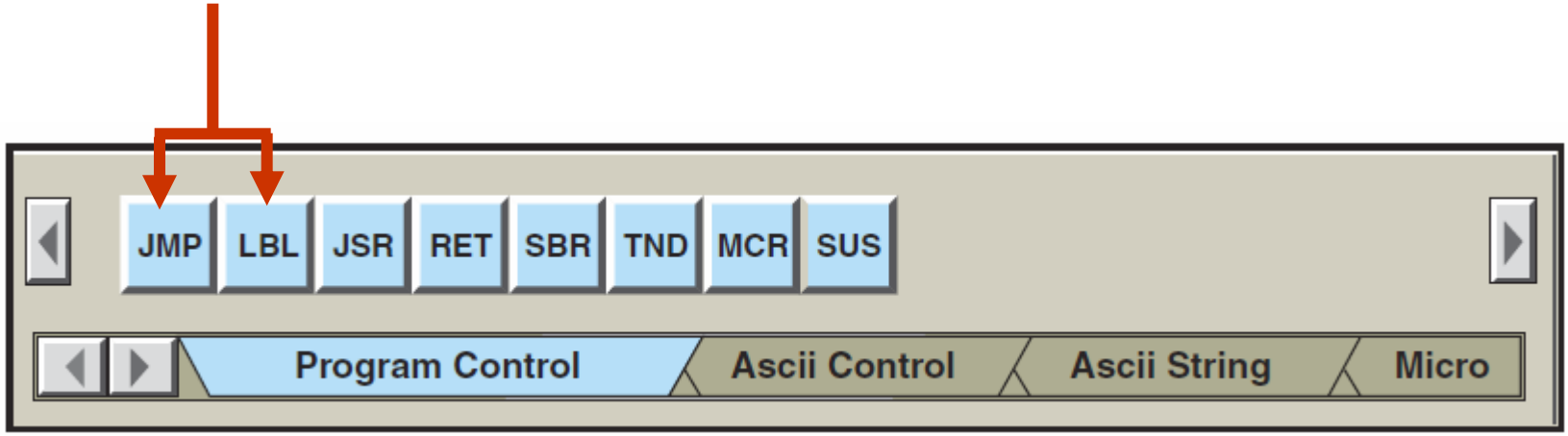


9.2

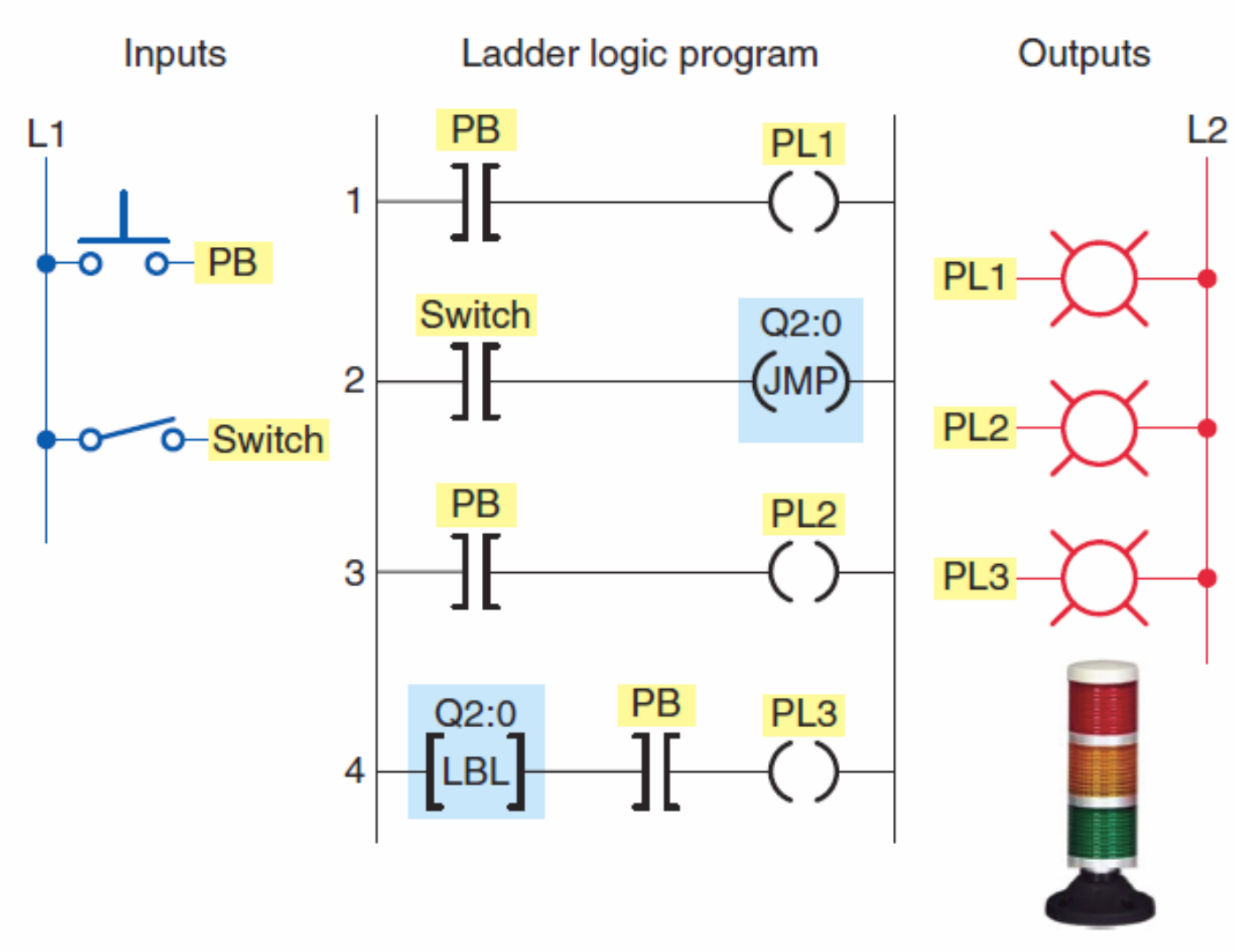
Jump Instruction

The *jump (JMP)* instruction is used to jump over certain program instructions when certain conditions exist.

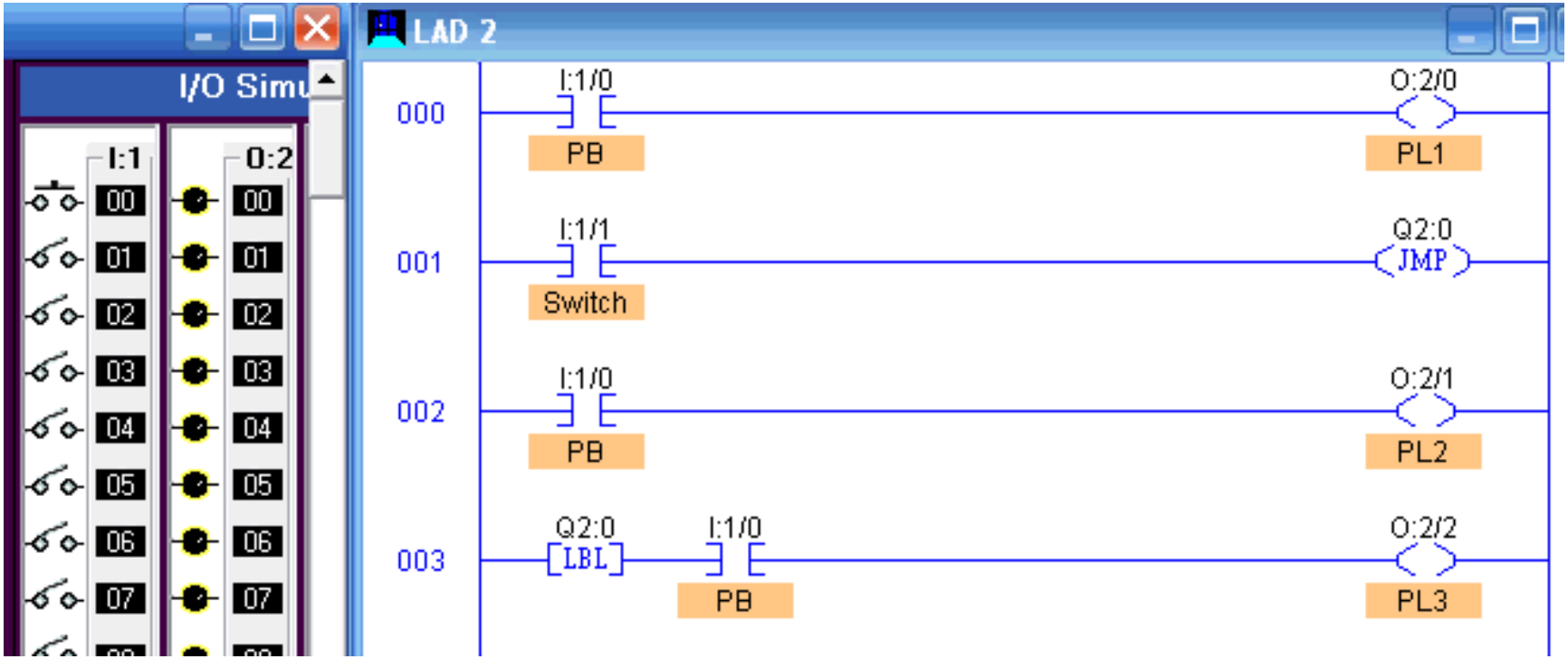
The label instruction is used to identify the ladder rung that is the target destination but does not contribute to logic continuity.



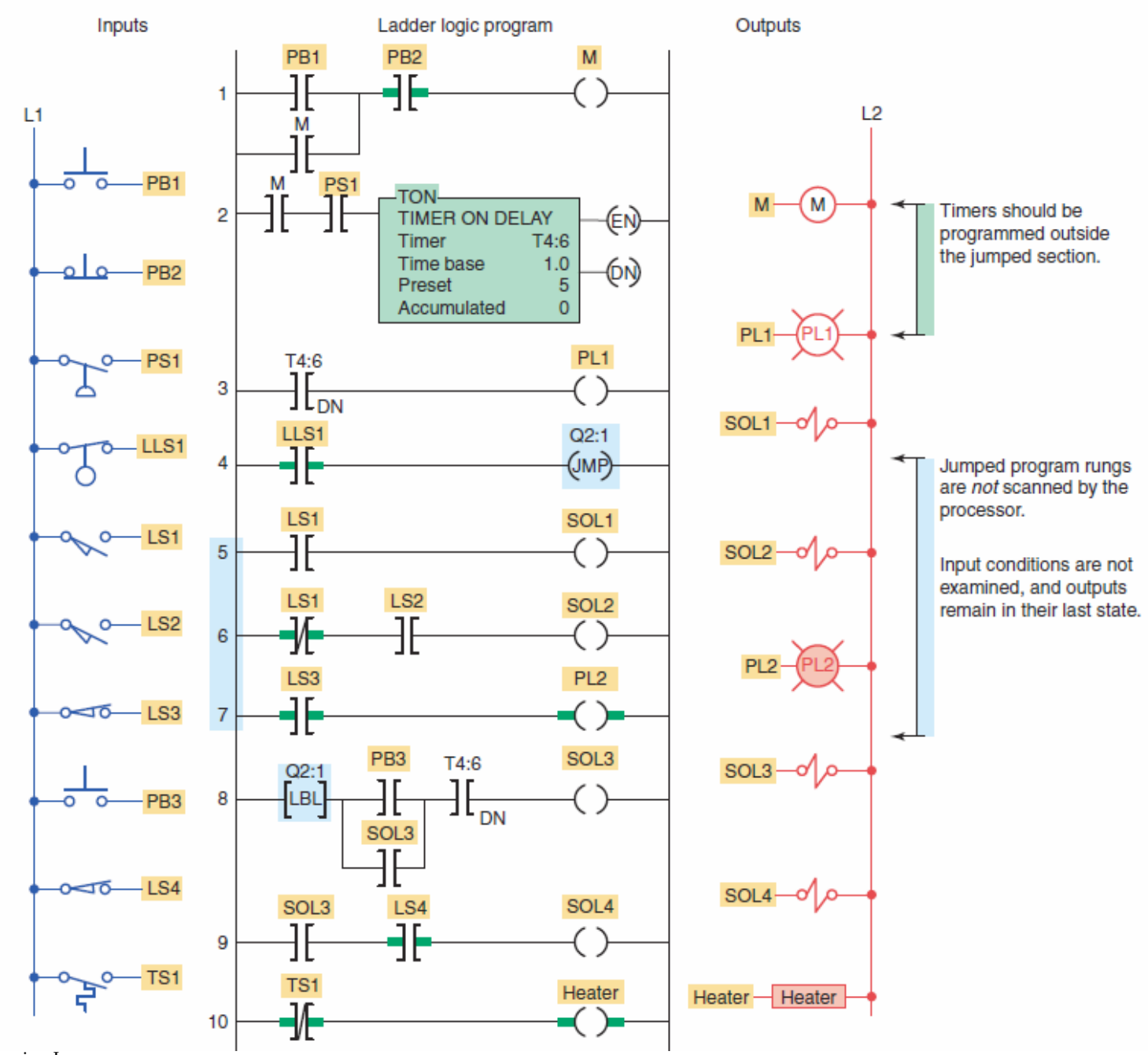
Jump (JMP) operation program.



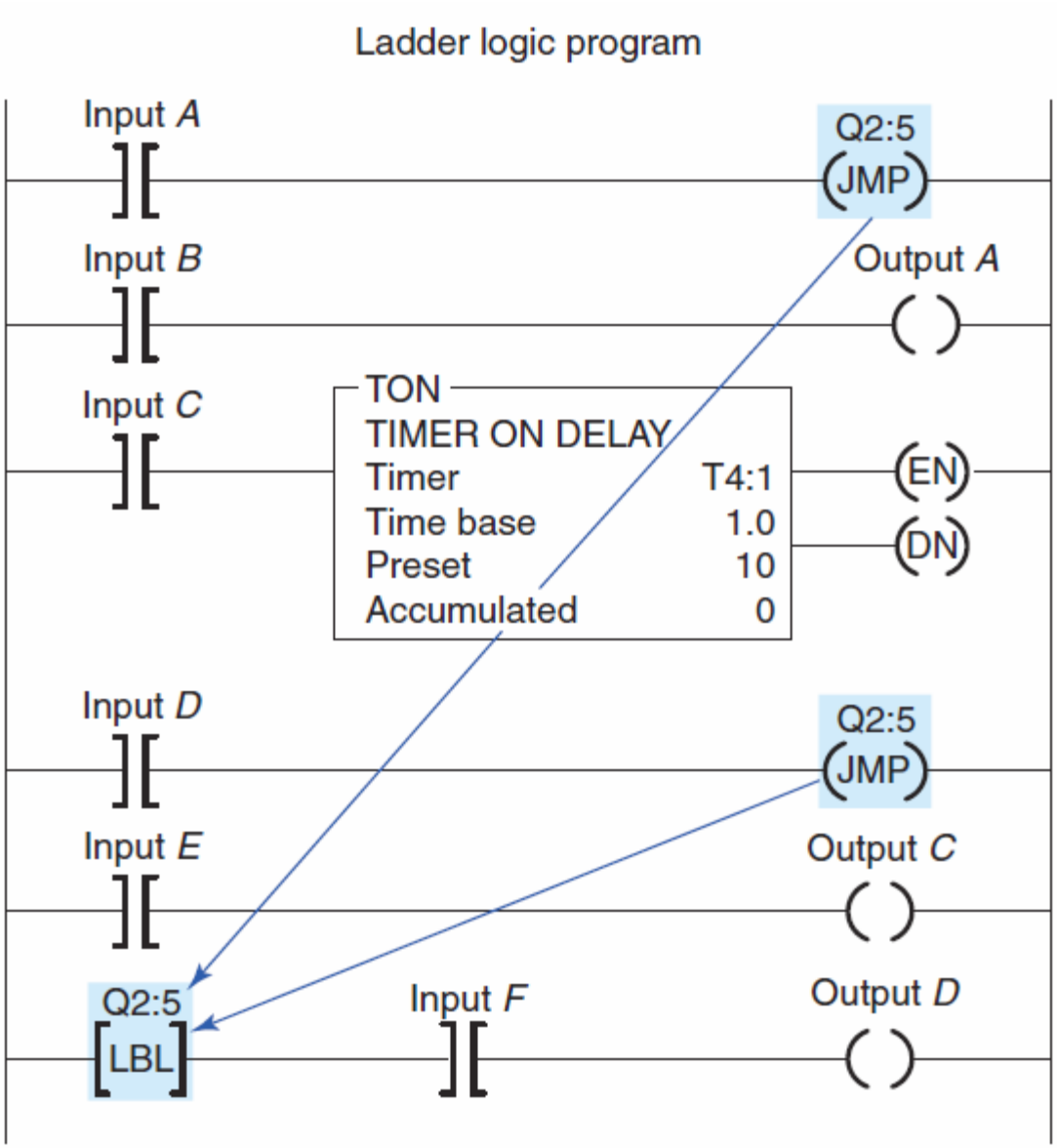
Simulated jump (JMP) operation program.



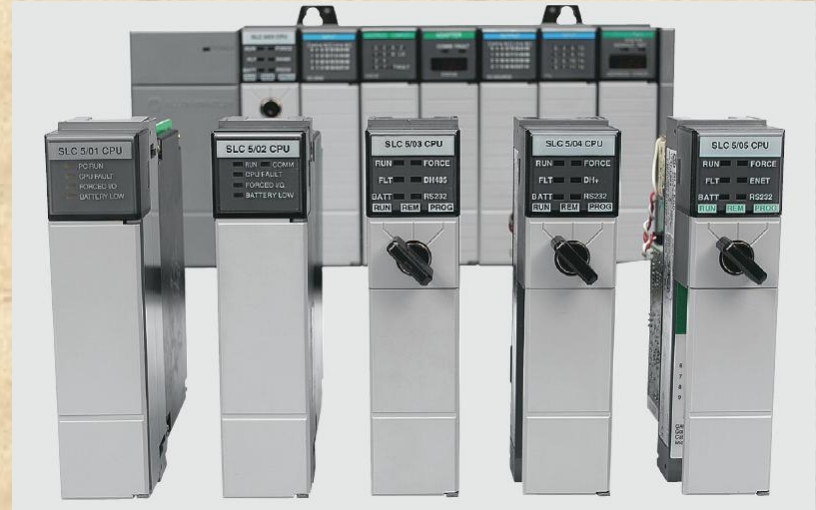
Effect on instructions of jumped rungs.



Jump-to-label from two locations.



9.3



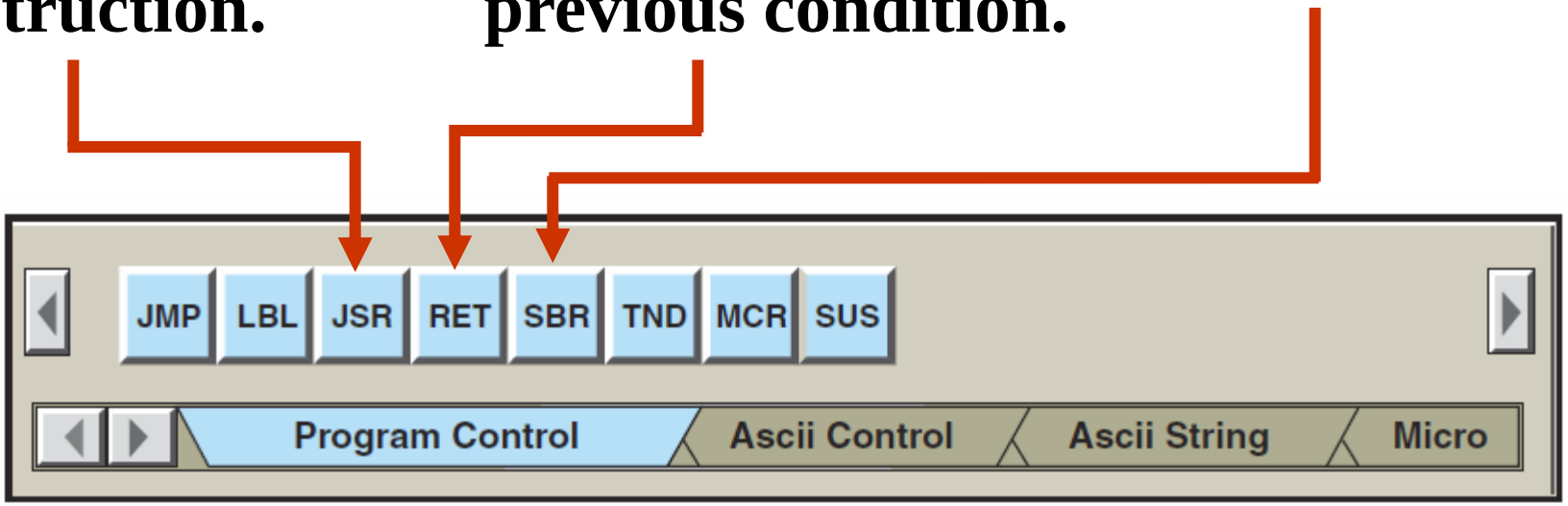
Subroutine Functions

A *subroutine* is a short program that is used by the main program to perform a specific function.

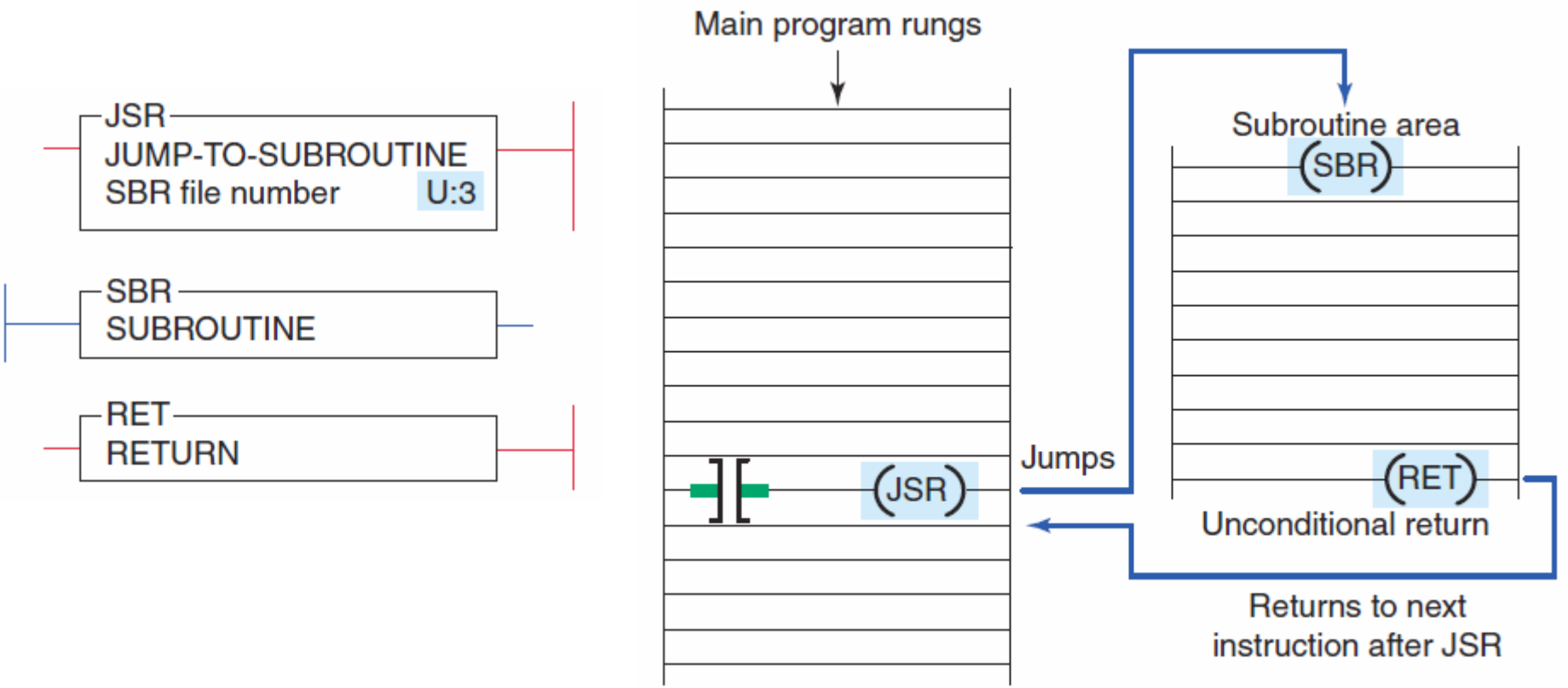
JSR (Jump to Subroutine)
Jump to a designated subroutine instruction.

RET (Return from Subroutine)
Exits current subroutine and returns to previous condition.

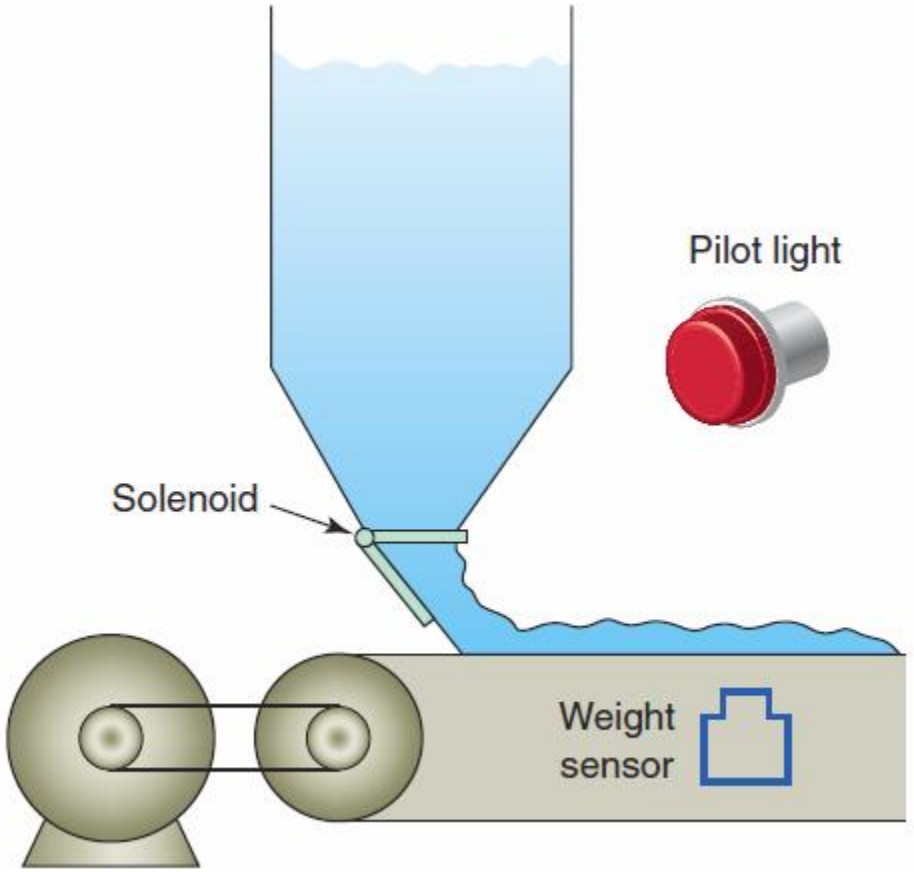
SBR (Subroutine)
Identifies the subroutine program.



When a subroutine is called from the main program, the program is able to escape from the main program and go to a program subroutine to perform certain functions and then return to the main program.

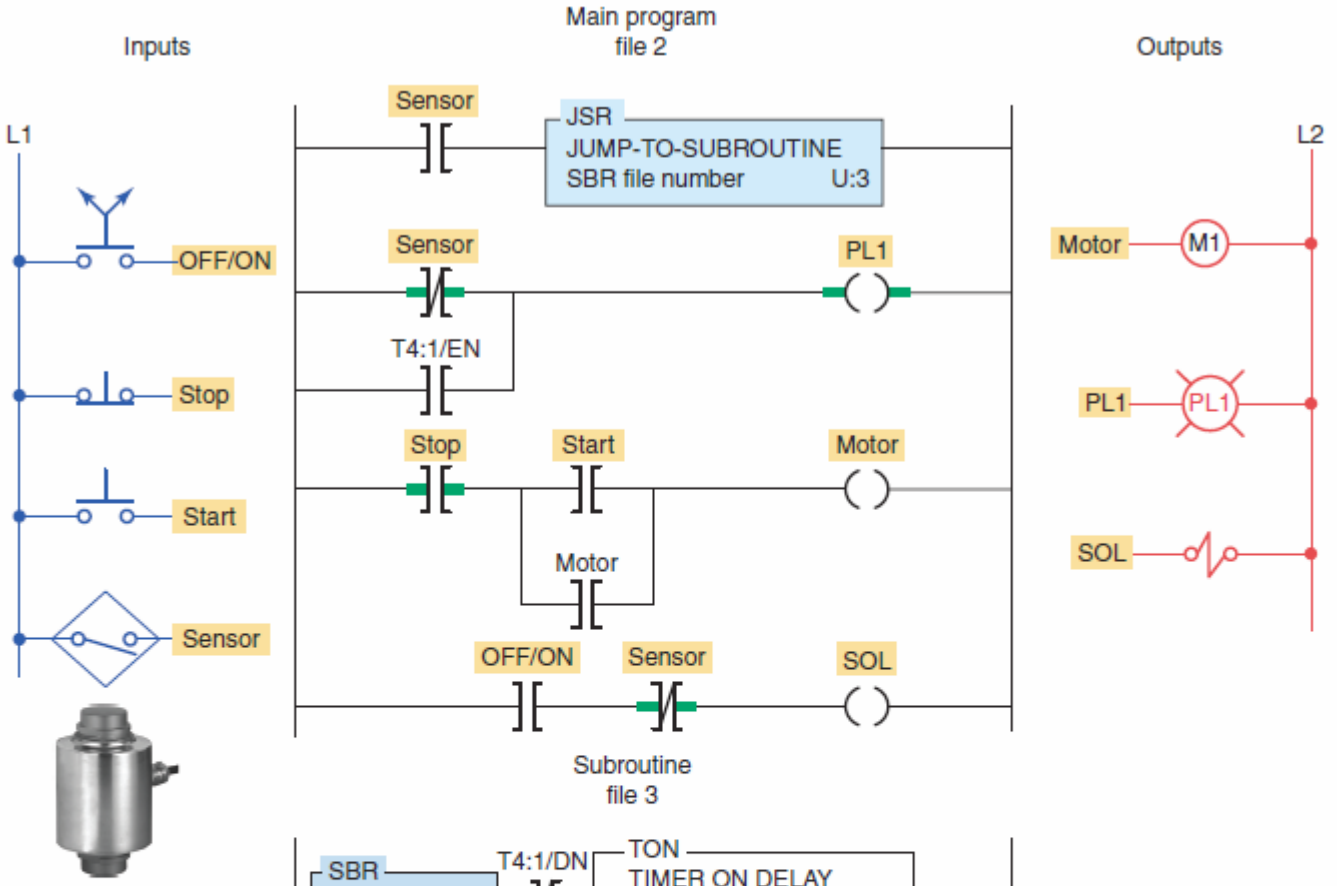


Flashing pilot light subroutine program.

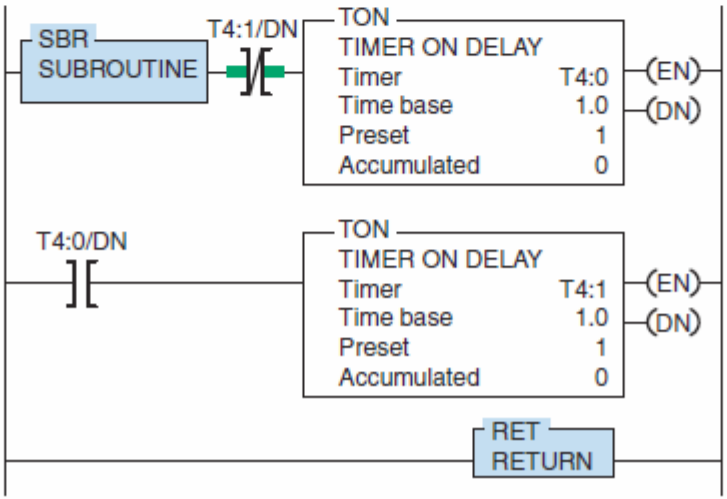


Anytime the weight on the conveyor exceeds a preset value, the subroutine program is scanned and pilot light changes from on to flashing.

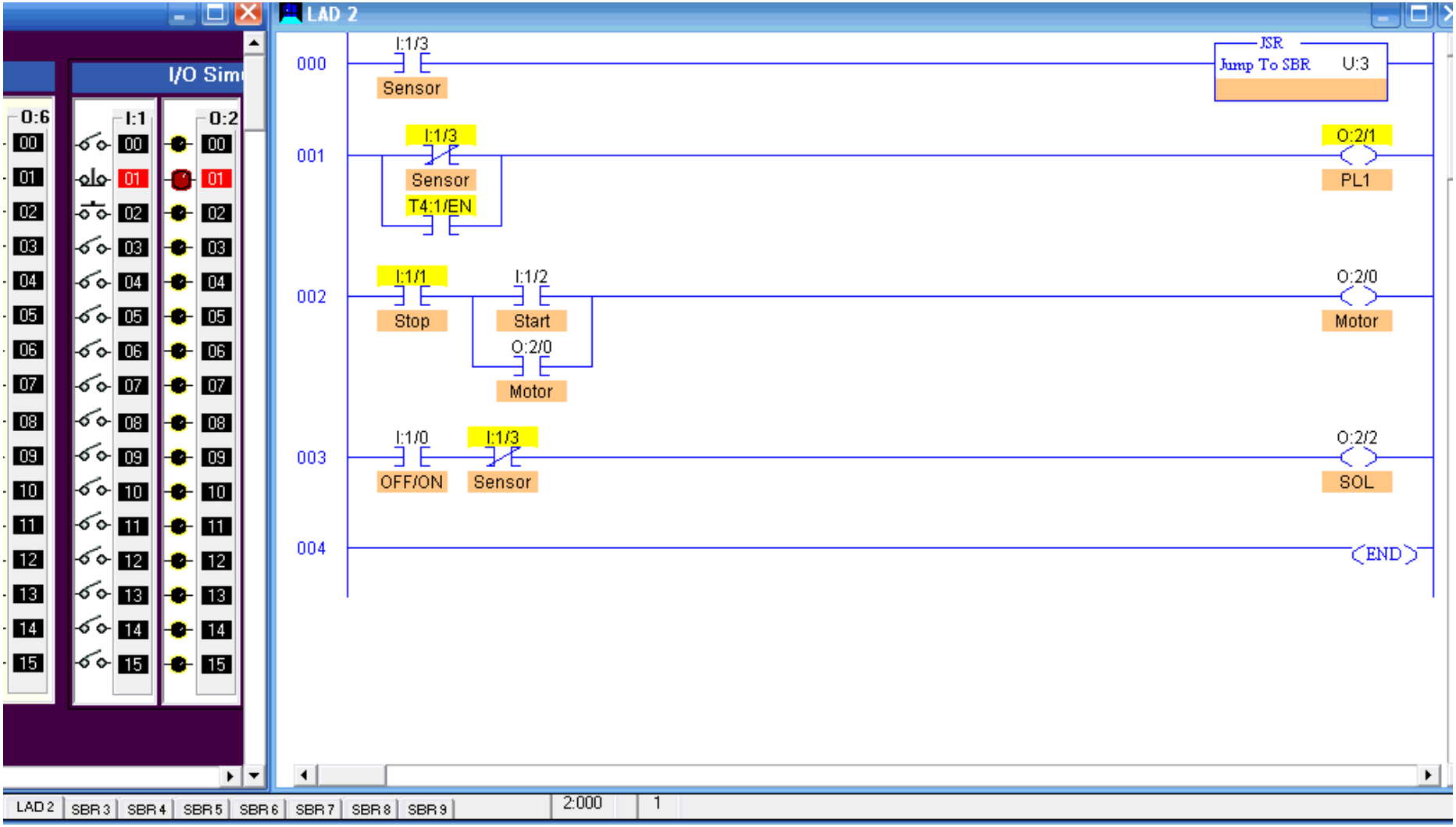
Main Program



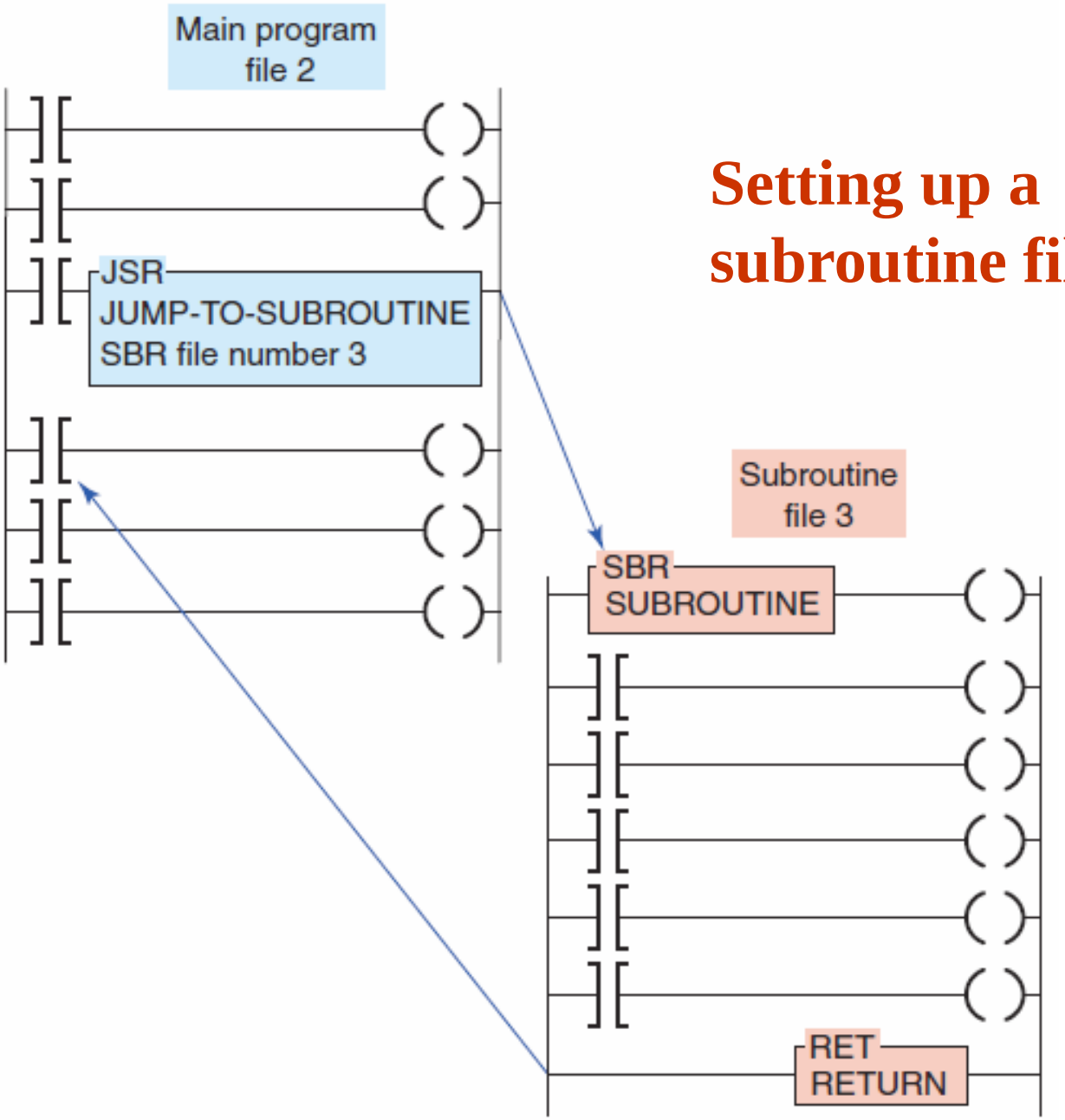
Subroutine Program



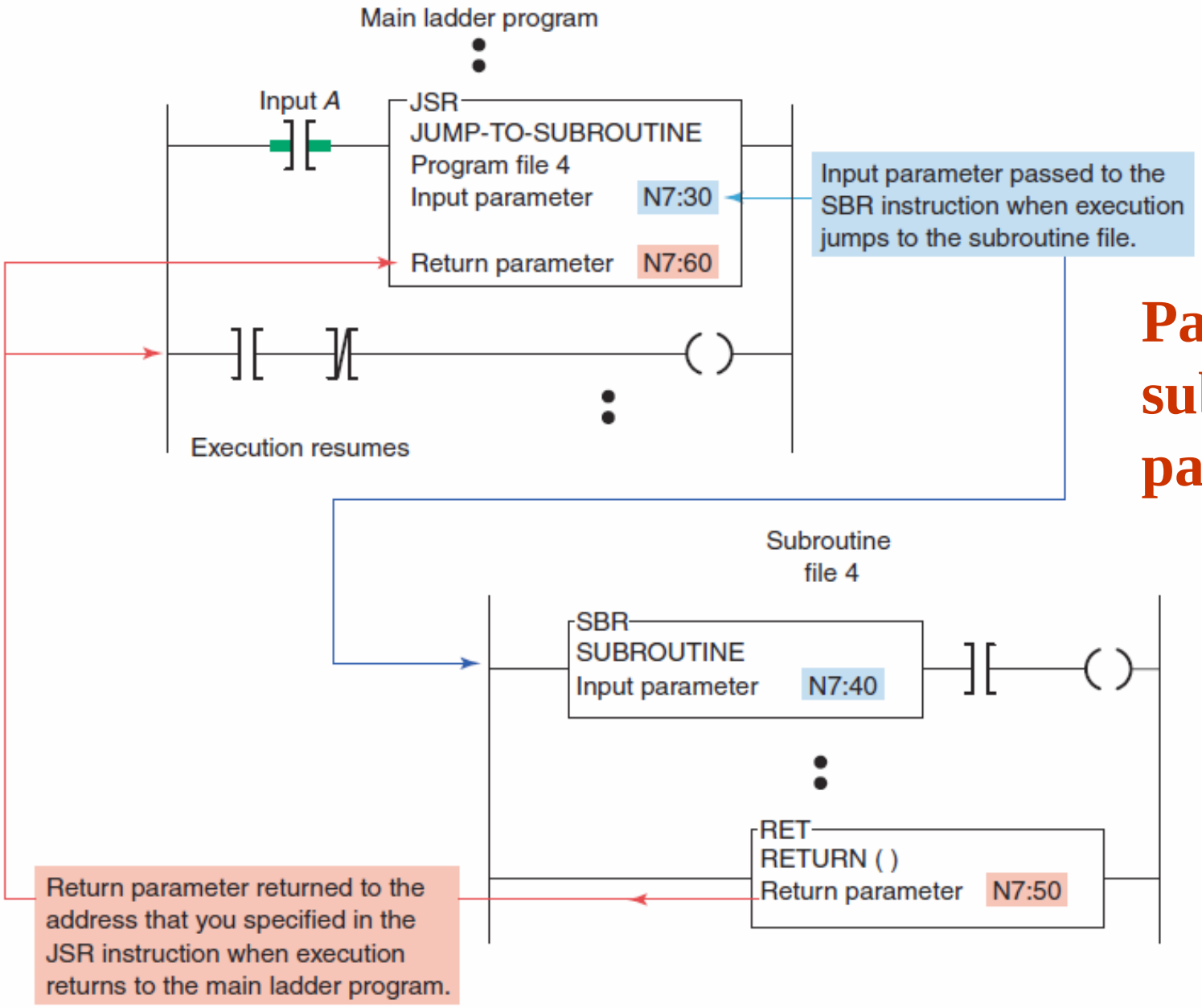
Simulated subroutine program.



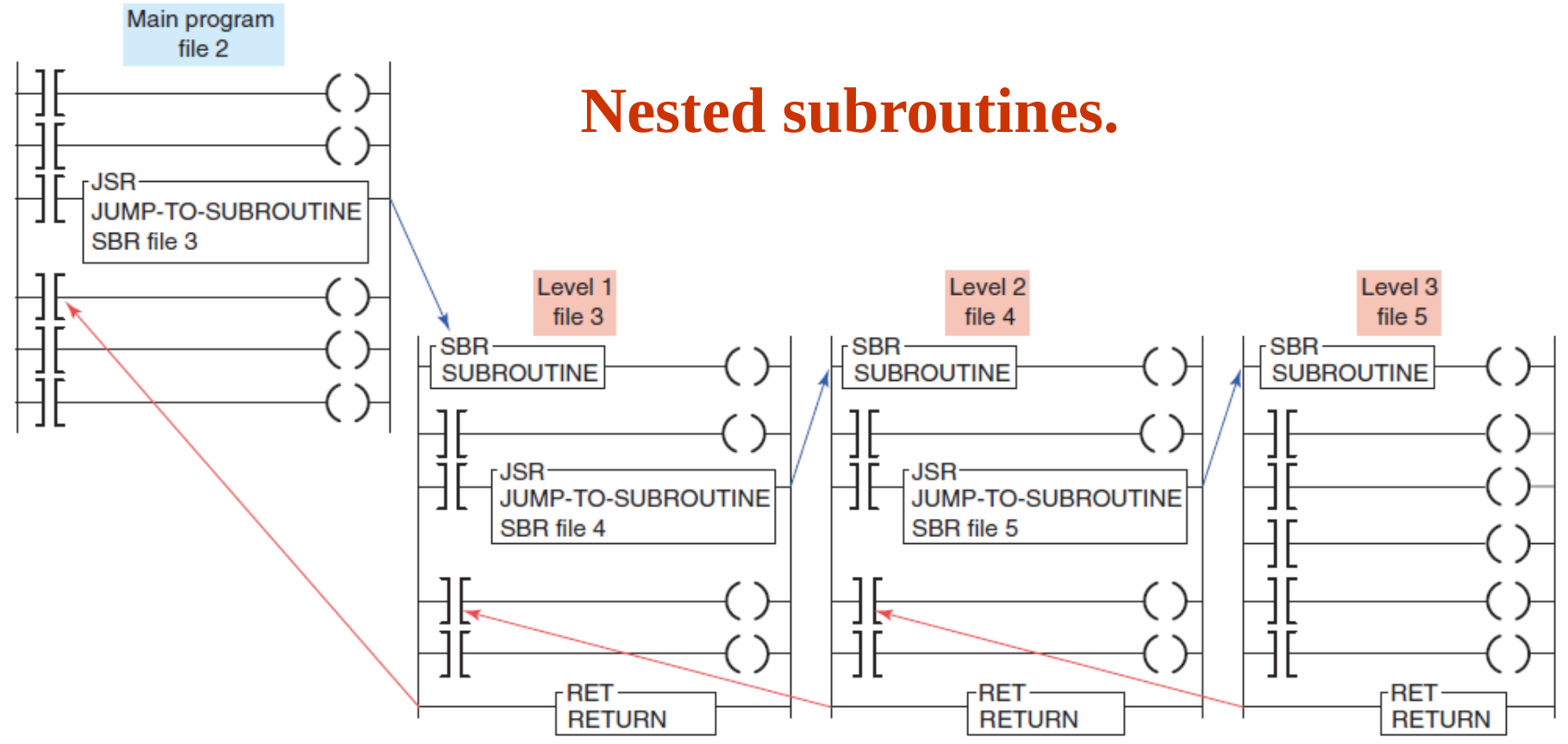
Setting up a subroutine file.



Passing subroutine parameters



Nested subroutines.

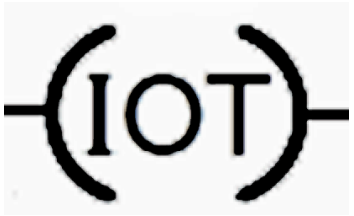


9.4

Immediate Input and Immediate Output Instructions

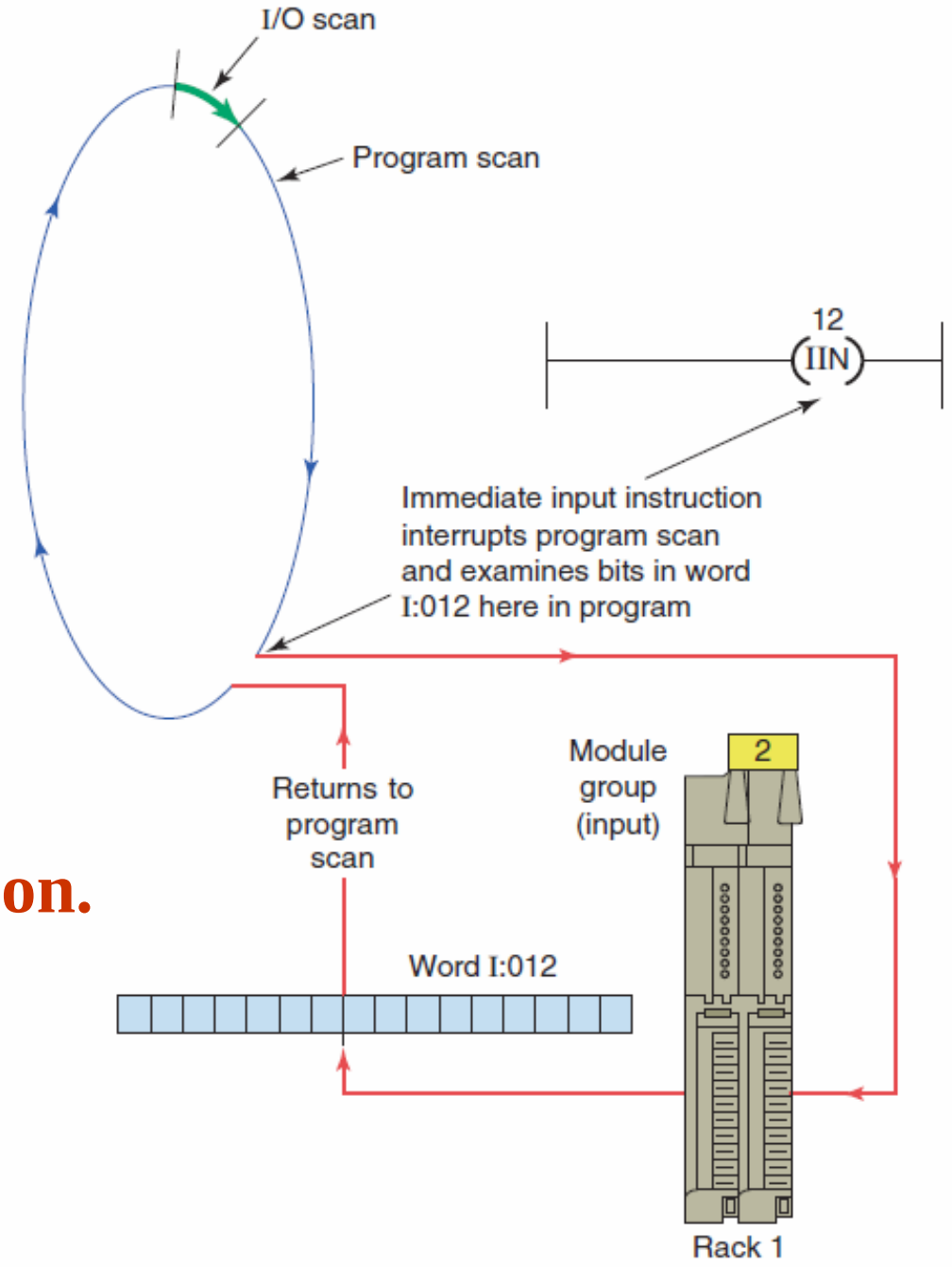
The *immediate input* and *immediate output* instructions interrupt the normal program scan to update the input and output image table files with current data.

Immediate input (IIN)
PLC-5 instruction is used to read an input condition before the I/O update is performed.

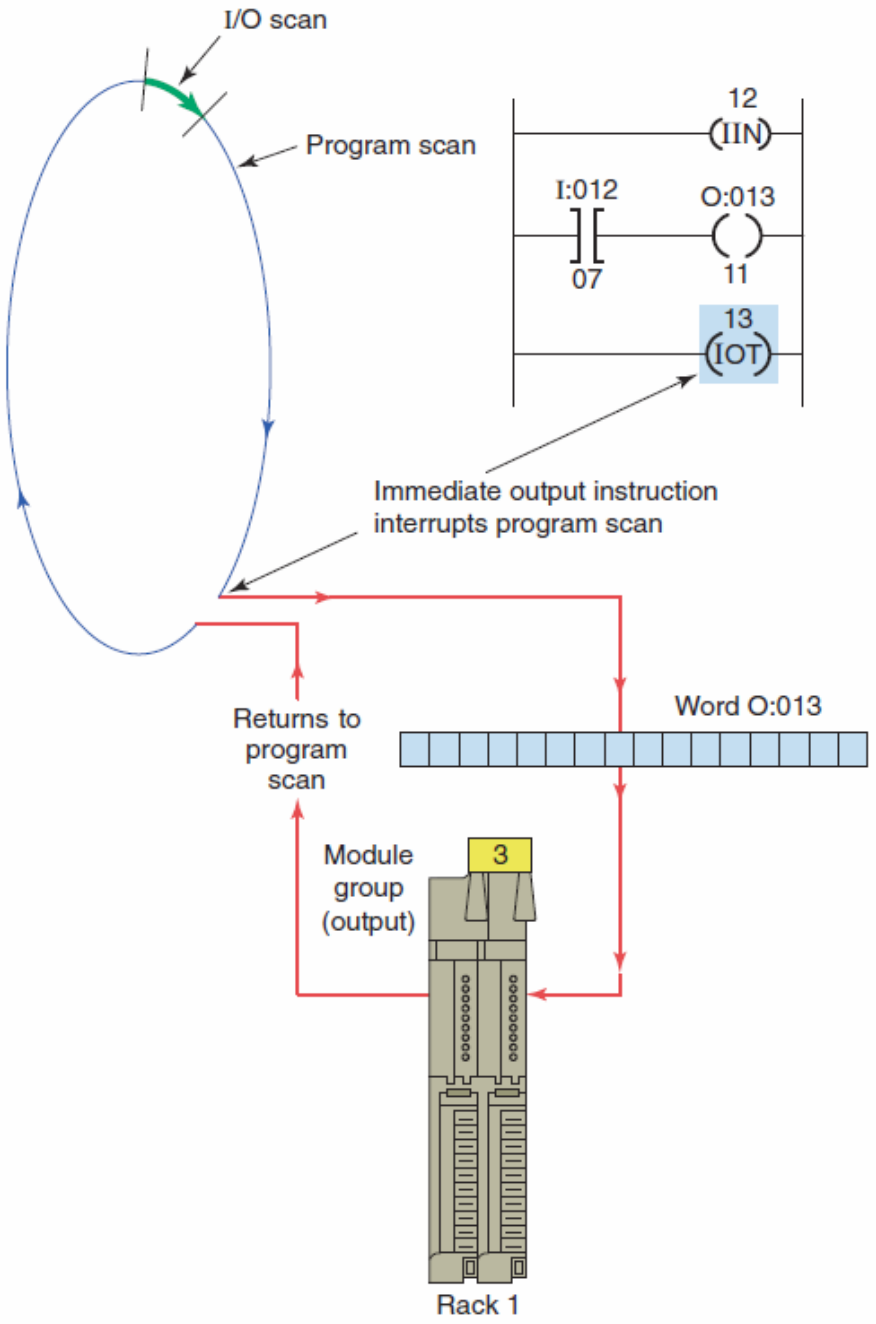


Immediate output (IOT)
PLC-5 instruction is used to update the status of an output device before the I/O update is performed.

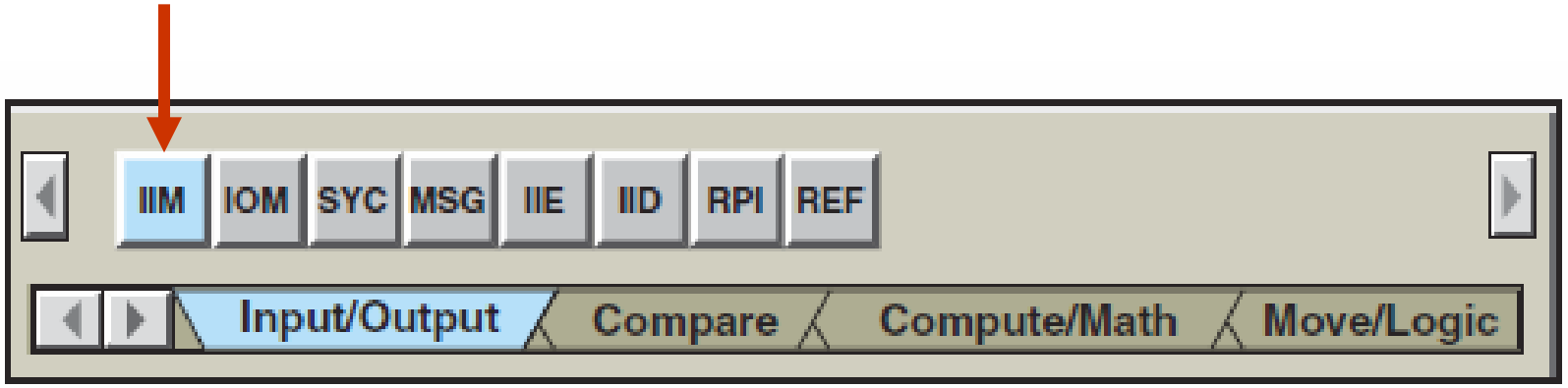
Immediate input instruction operation.

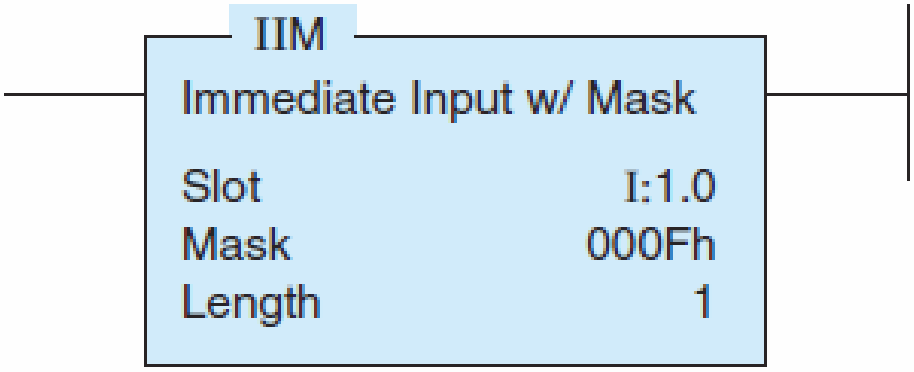


Immediate output instruction operation.



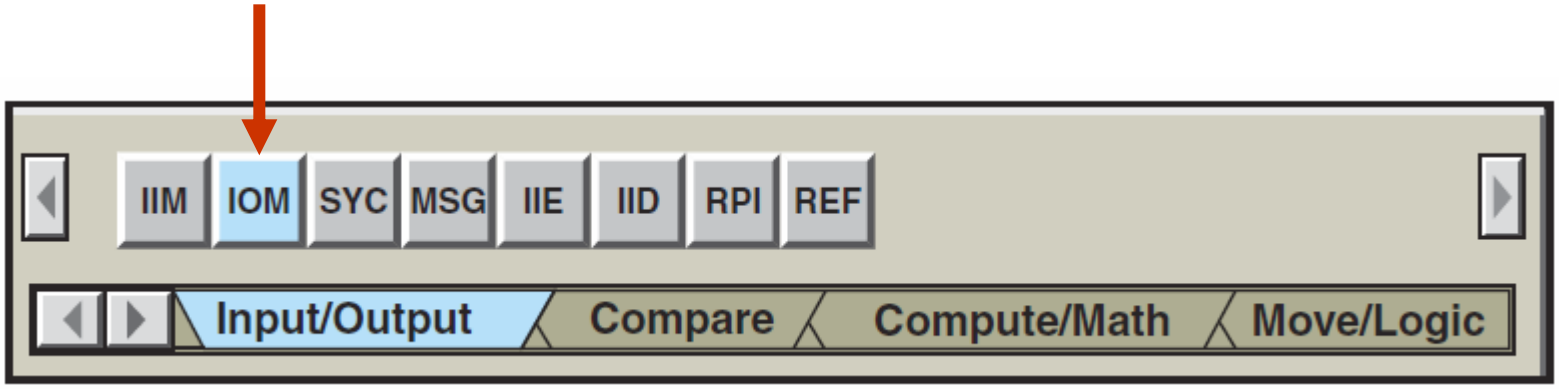
SLC 500 *Immediate Input with Mask (IIM)* instruction.

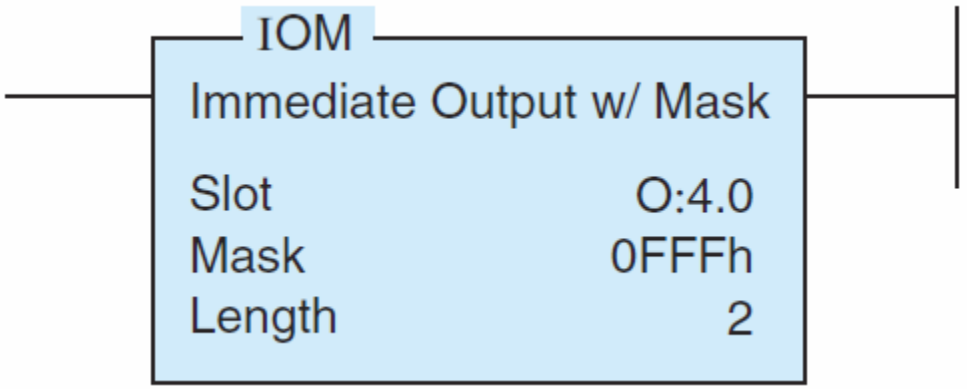




When the rung is true, the program scan is interrupted, and data from a specific input slot are transferred through the mask to the input data file.

SLC 500 *Immediate Output with Mask (IOM)* instruction.

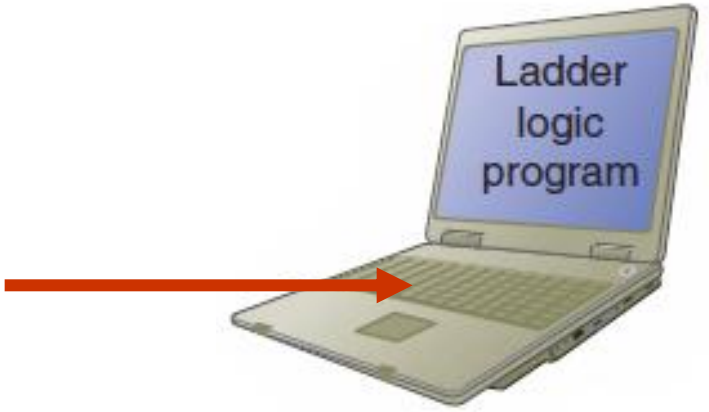


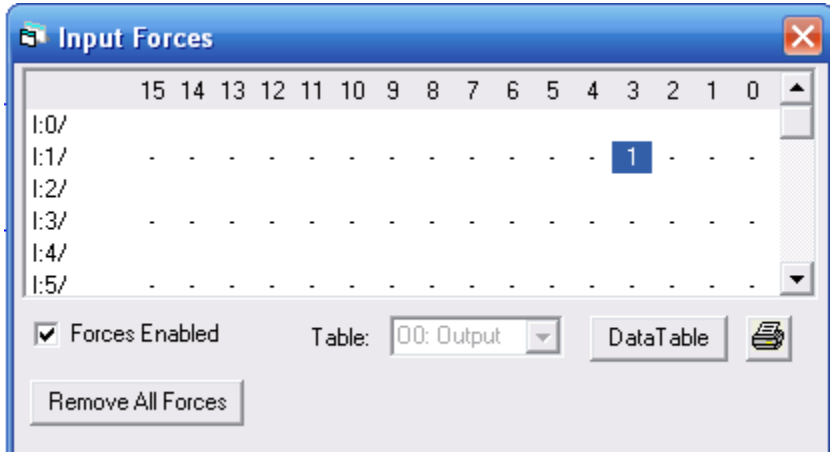


When the rung is true, the program scan is interrupted to update output data to the module located in the slot specified in the instruction.

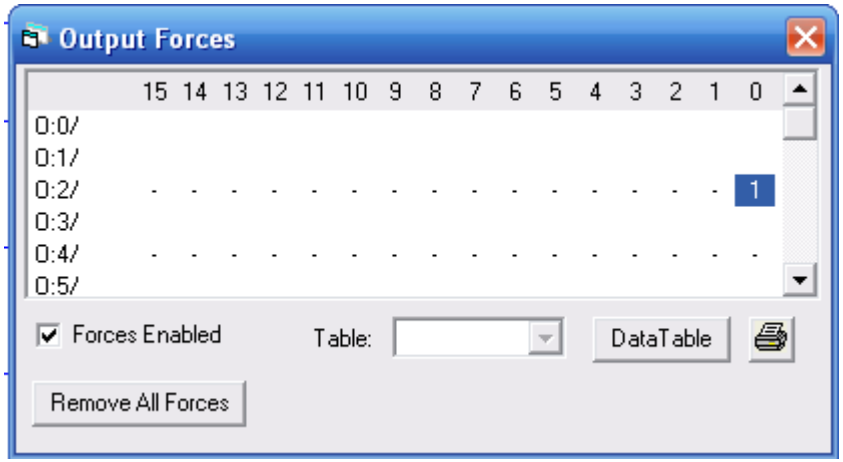
Forcing External I/O Addresses

Forcing **allows the PLC user to turn an external input or output *on or off* from the keyboard of the programming device.**

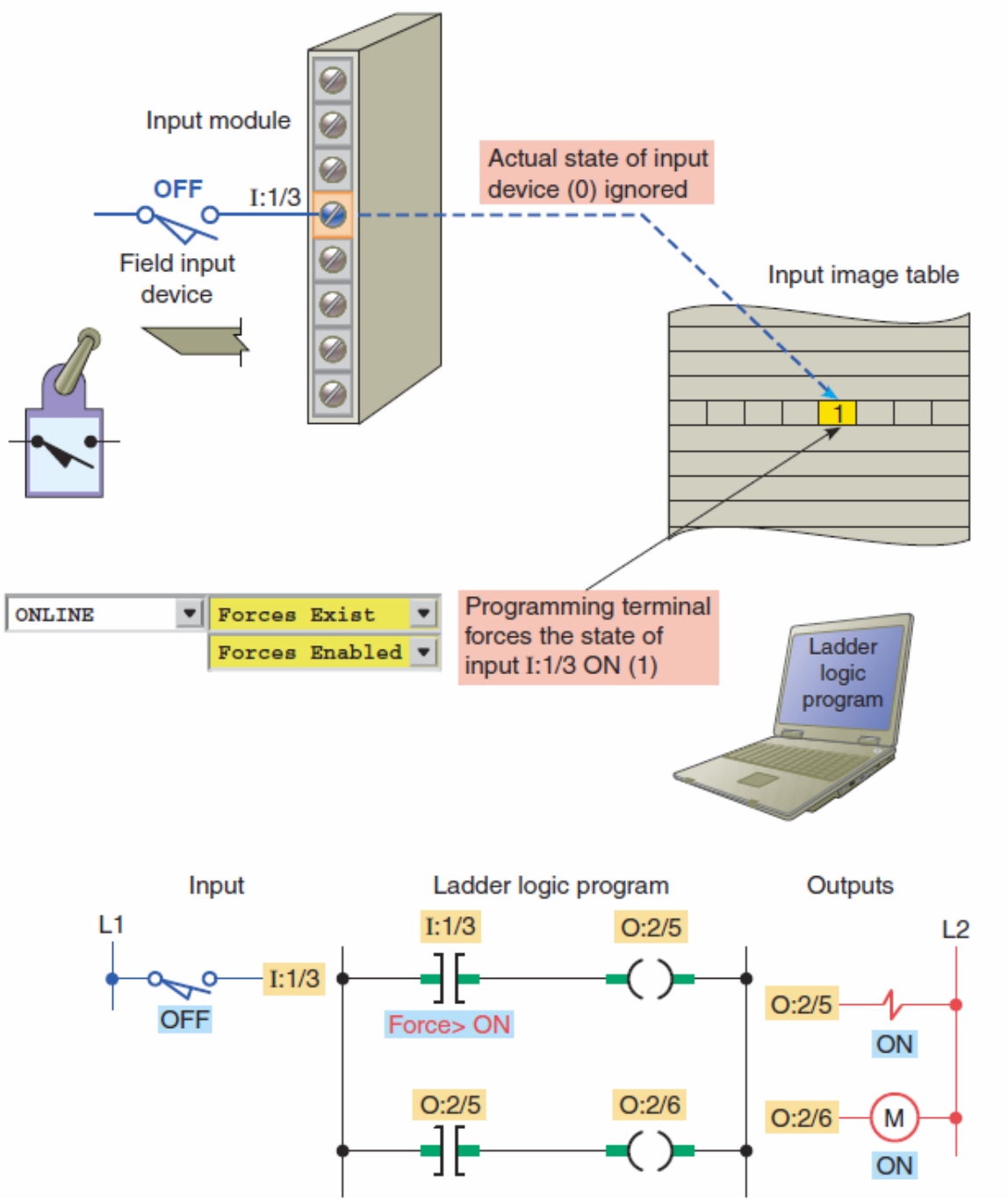




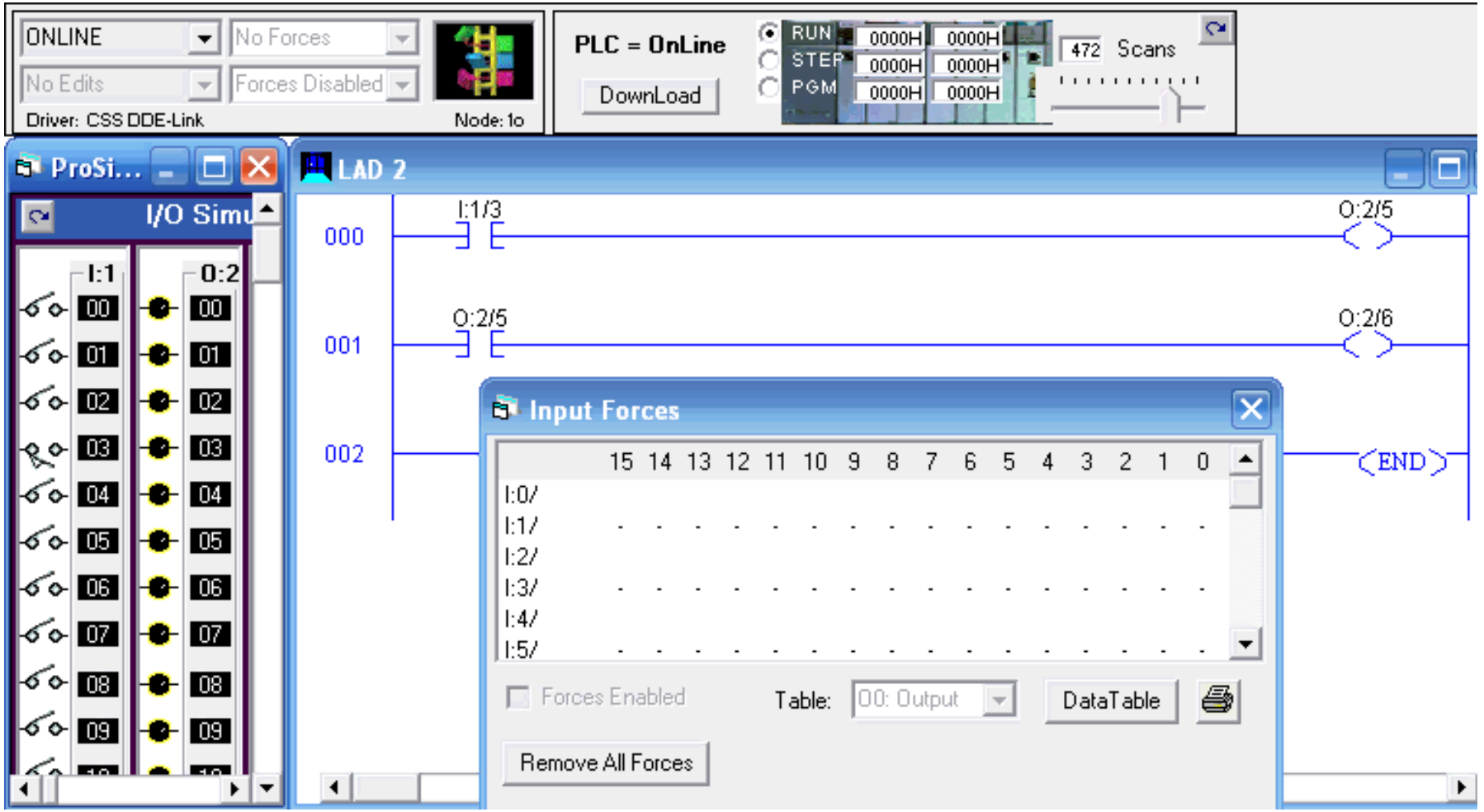
Forcing is accomplished regardless of the actual state of the field device.



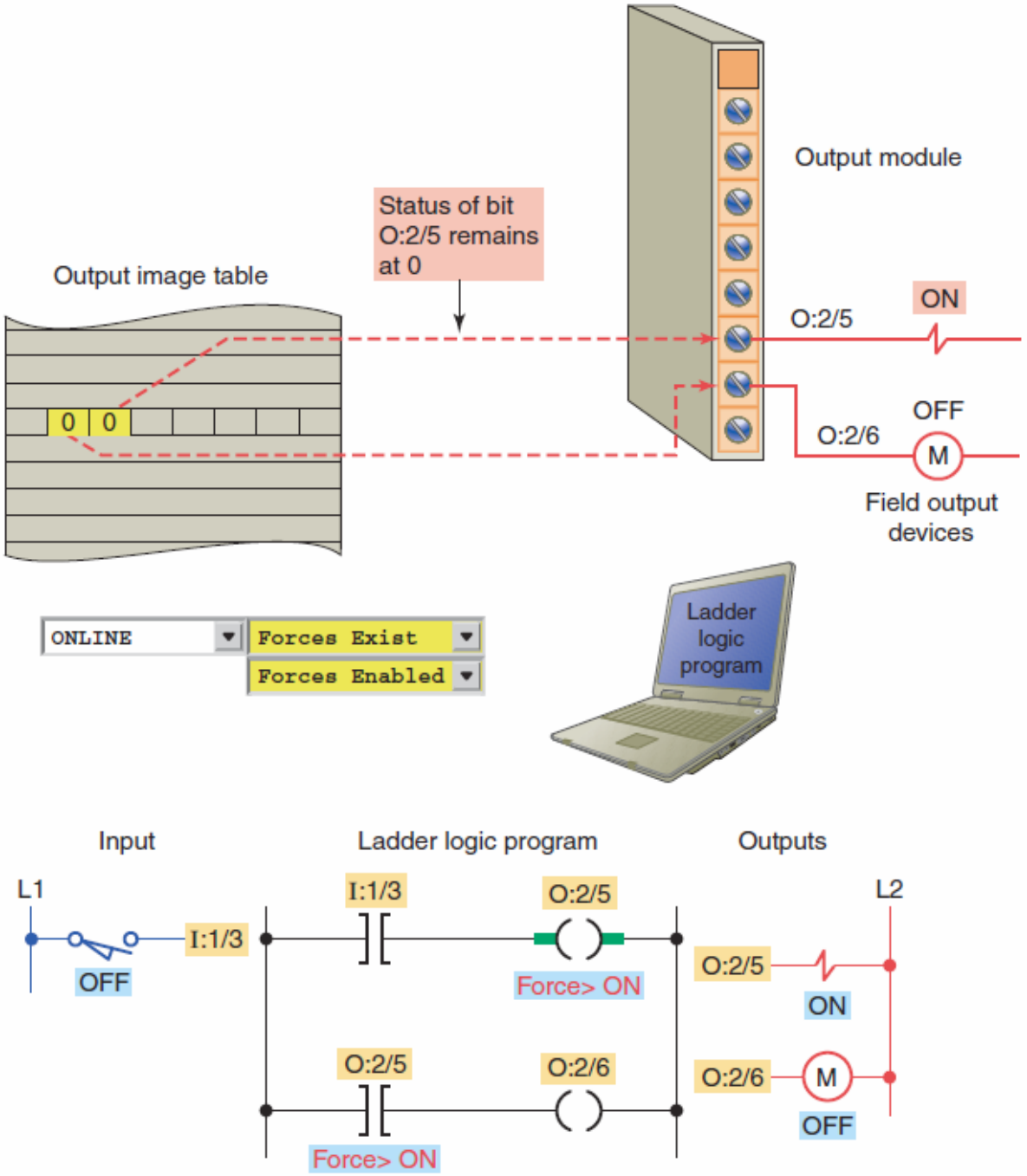
Forcing an input on.



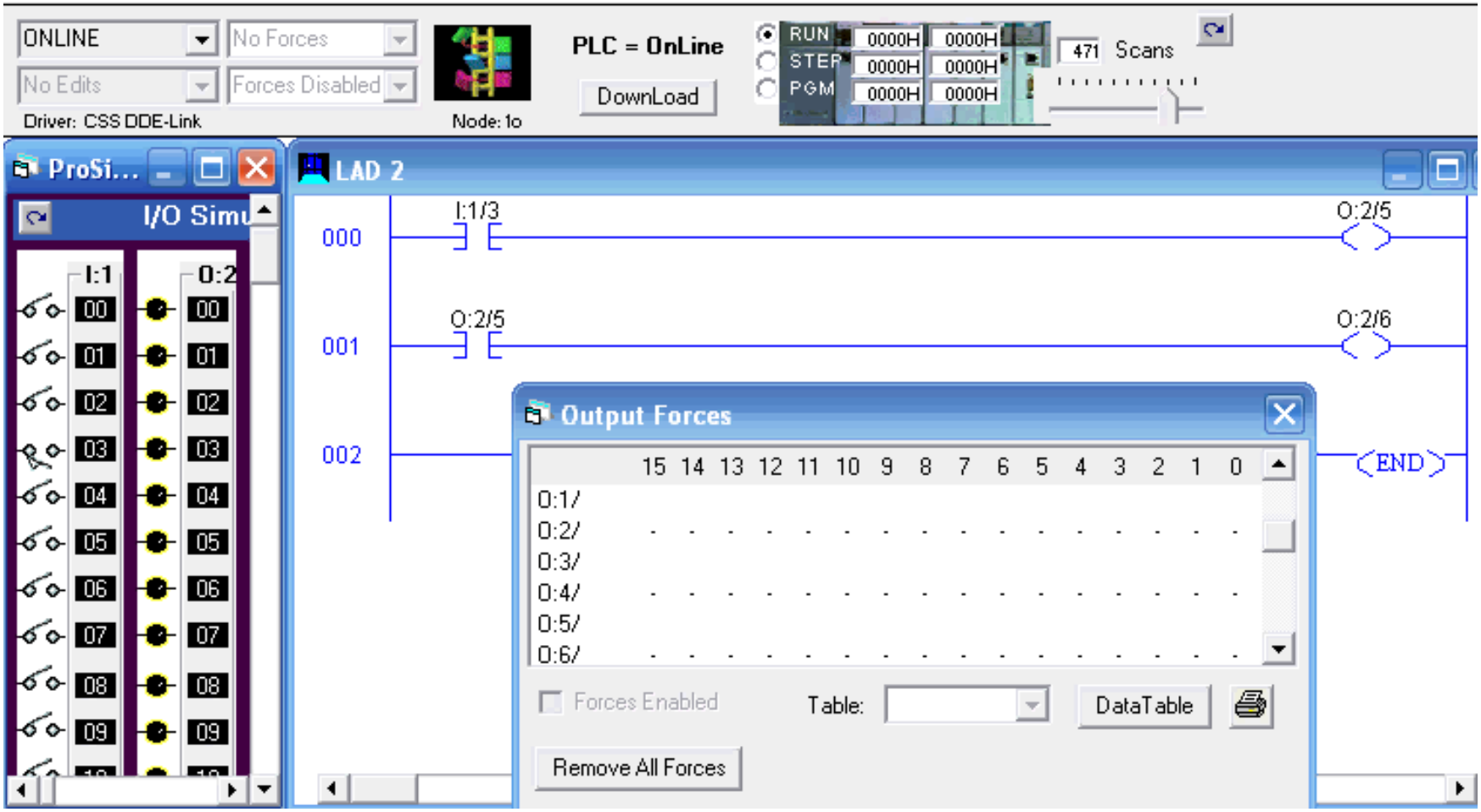
Simulating forcing an input on.



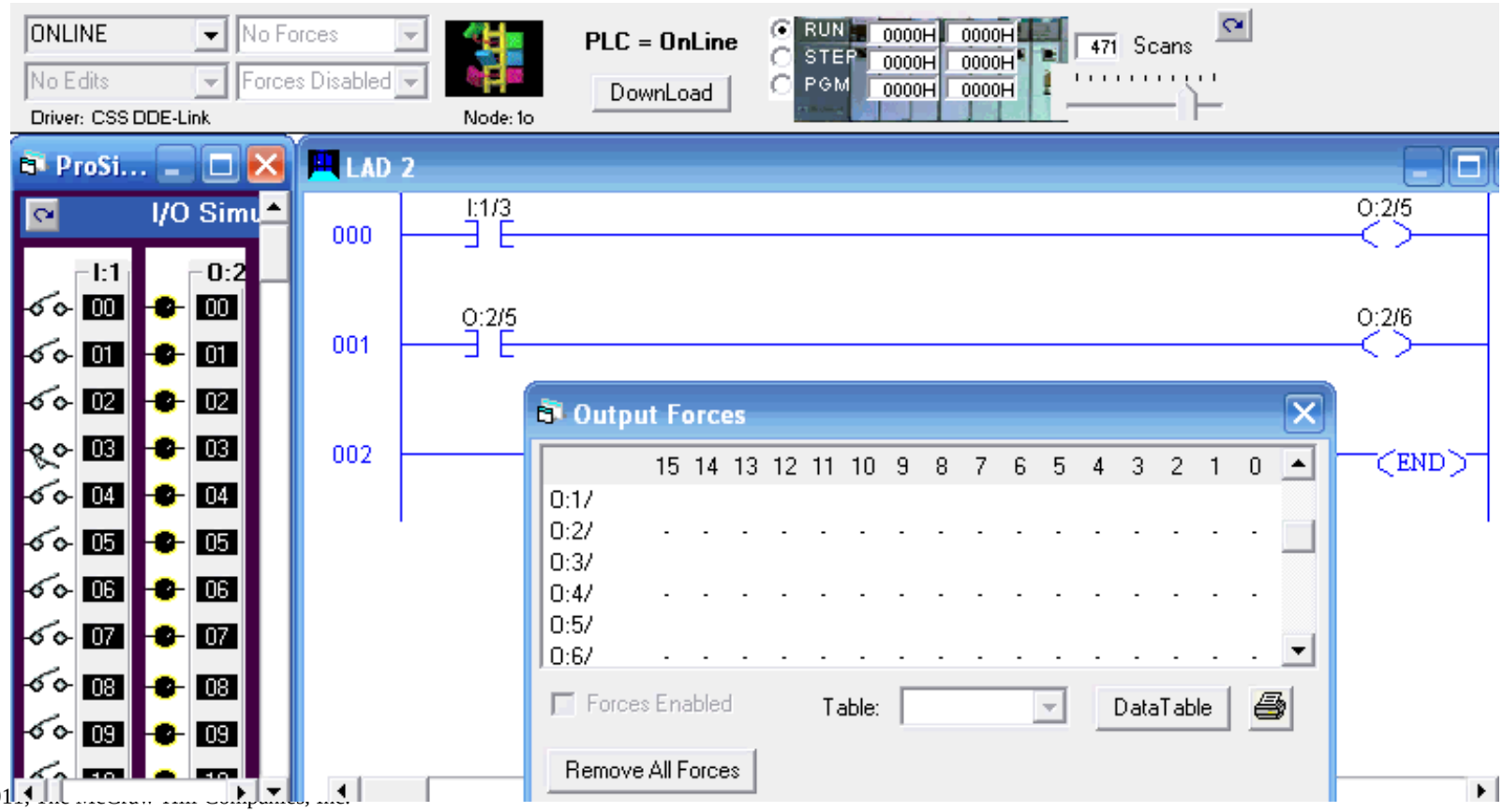
Forcing an output on.



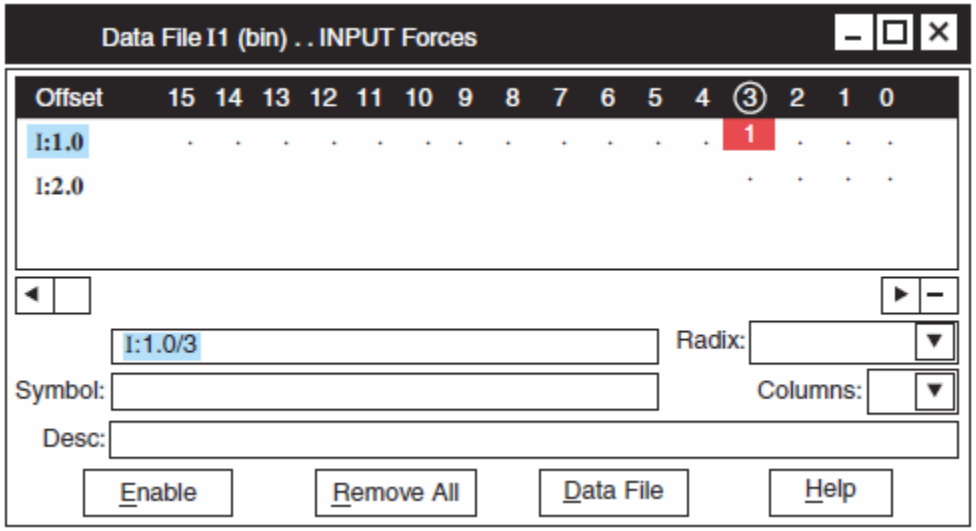
Simulating forcing an output on.



By forcing *outputs off*, you can prevent the controller from energizing those outputs even though the ladder logic, which normally controls them, may be true.



You can enter and enable or disable forces while you are monitoring your file offline, or in any processor mode while monitoring your file online.



Forces version of the data table with bit I:1/3 forced on.

If used *incorrectly*, force functions can cause injuries to persons working around a system, and/or equipment damage.

You must understand the potential effect that forcing given inputs or outputs will have on machine operation in order to avoid possible personal injury and equipment damage.

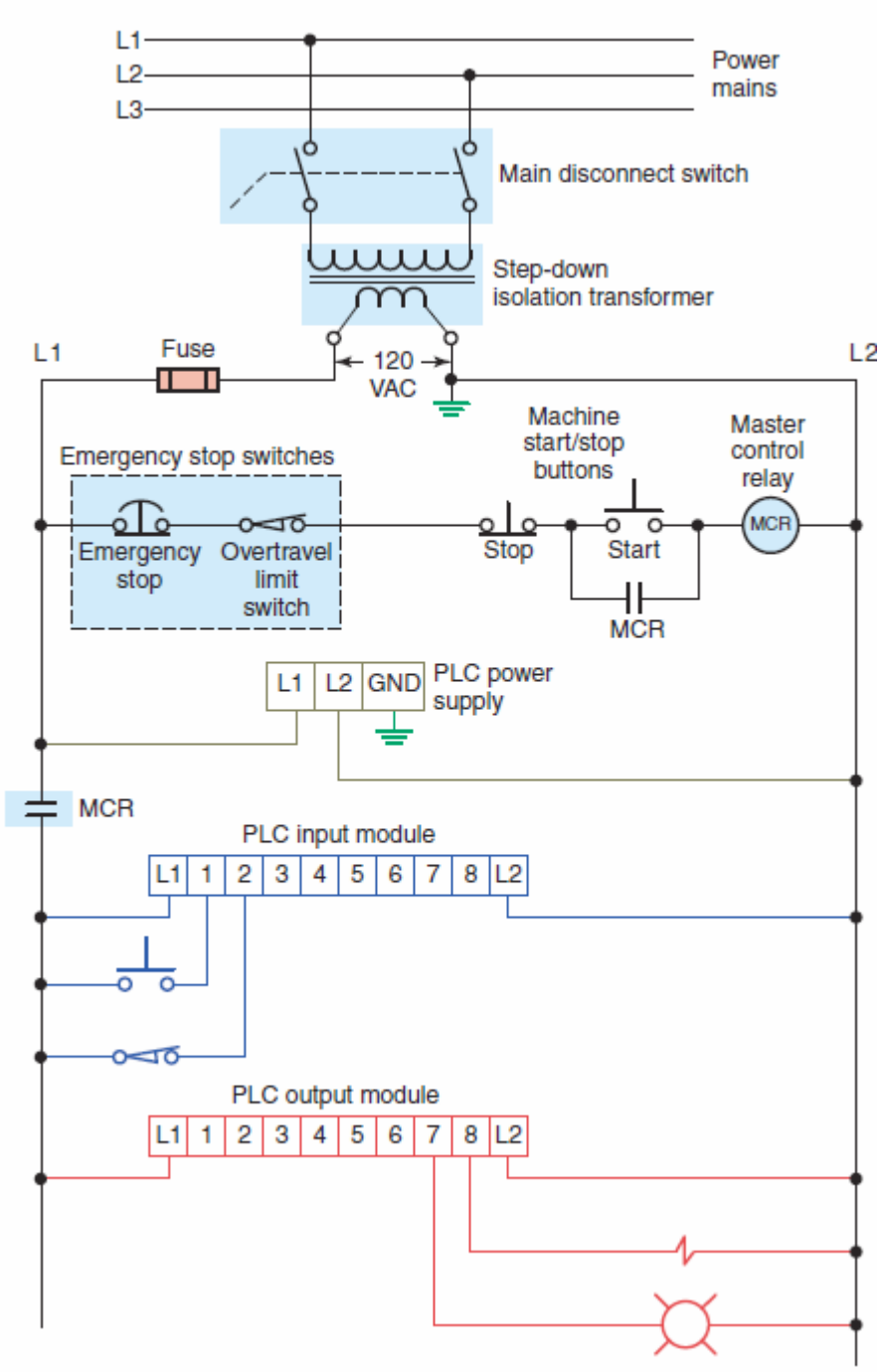
9.6

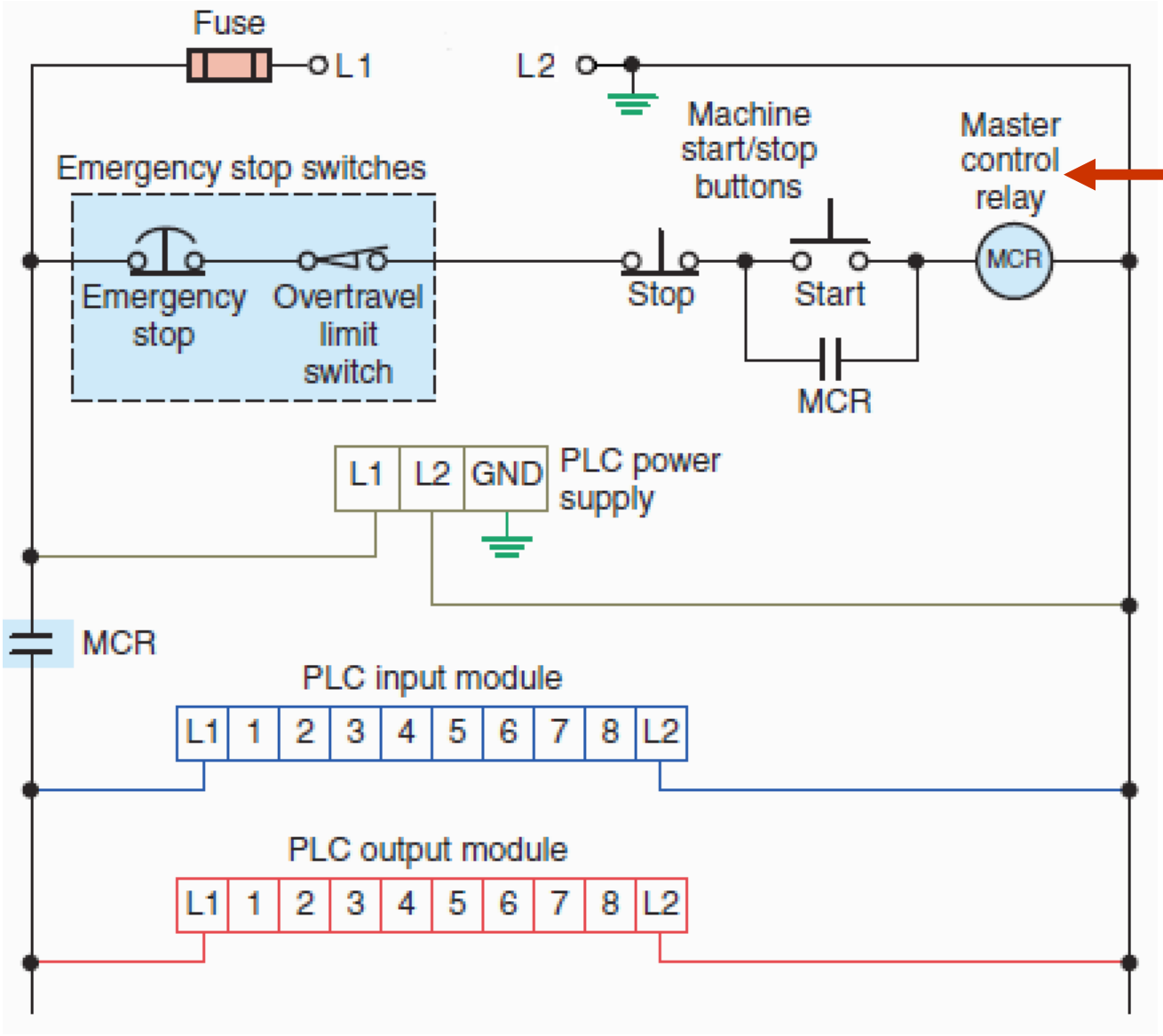
Safety Circuitry

Sufficient emergency circuits must be provided to stop either *partially or totally* the operation of the controller or the controlled machine or process.

These circuits should be hardwired outside the controller.

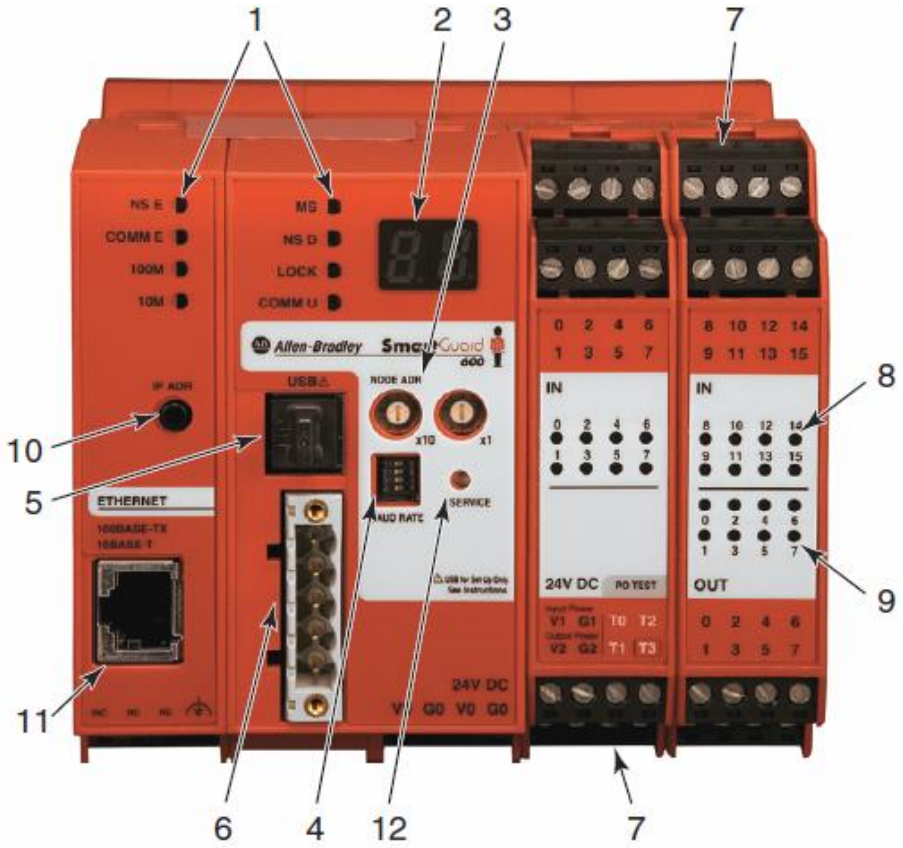
Safety wiring requirements for a PLC installation.





The master control relay must be able to inhibit all machine motion by removing power to the machine I/O devices when the relay is deenergized.

Safety PLCs are now available for applications that require more advanced safety functionality.

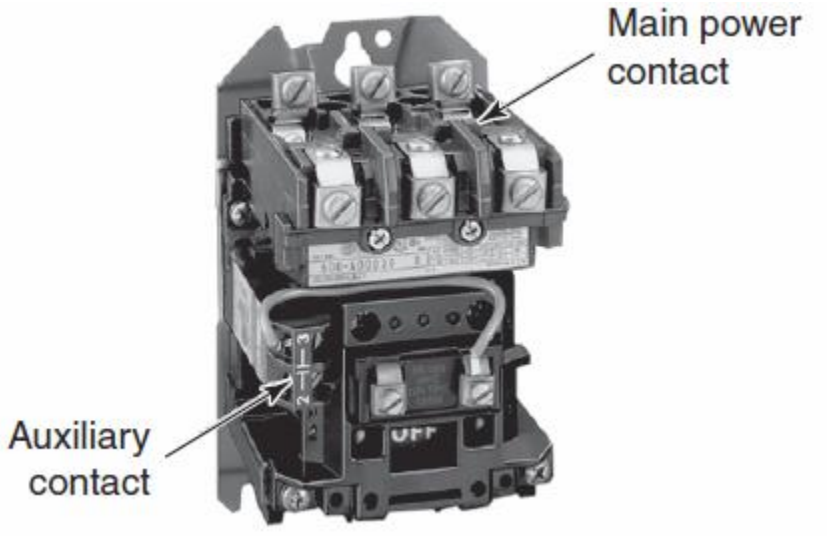


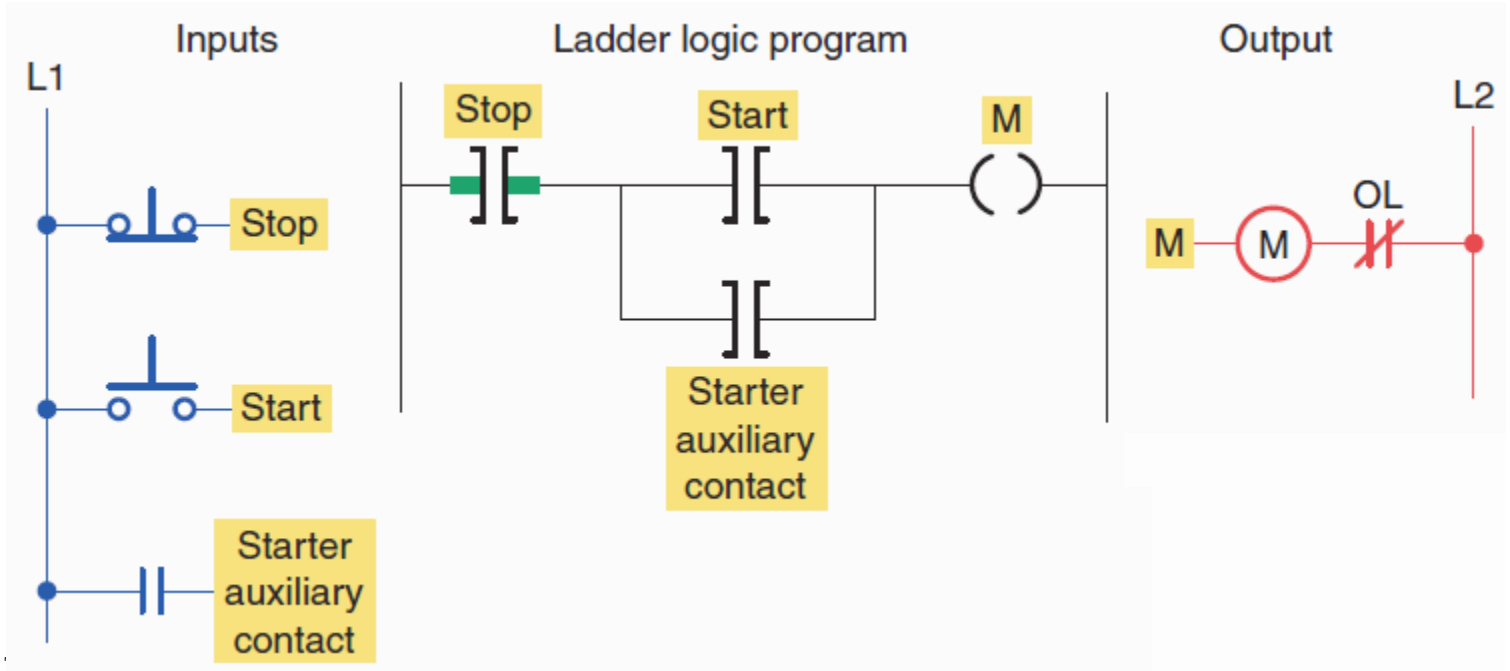
Number	Feature
1	Module status indicators
2	Alphanumeric display
3	Node address switches
4	Baud rate switches
5	USB port
6	DeviceNet communication connector
7	Terminal connectors
8	Input status indicators
9	Output status indicators
10	IP address desplay switch
11	Ethernet connector
12	Service switch

Both standard and safety PLCs have the ability to perform control functions but a standard PLC was not initially designed to be *fault tolerant and fail-safe.*

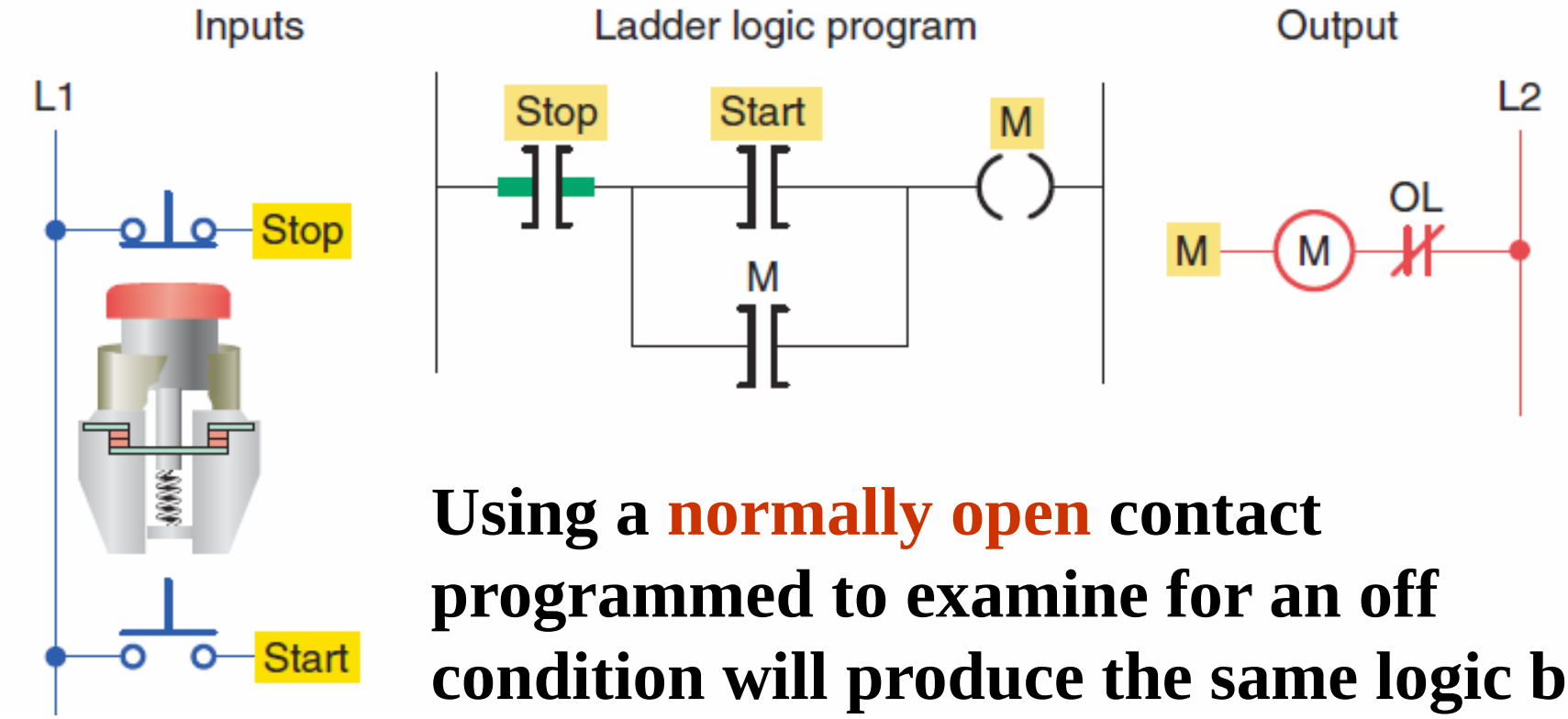
- **Safety PLCs have redundant (dual) microprocessors.**
- **Safety PLCs have an internal output circuit associated with each input for the purpose of testing the input circuitry.**
- **Safety PLCs use power supplies designed specifically for use in safety control systems and redundant backplane circuitry between the controller and I/O modules.**

The use of the field-generated starter auxiliary contact is *safer* because it provides positive feedback to the processor about the exact status of the motor.





All stop buttons should be wired using a *normally closed* contact programmed to examine for an *on condition*.



Using a normally open contact programmed to examine for an off condition will produce the same logic but is not considered to be as safe.

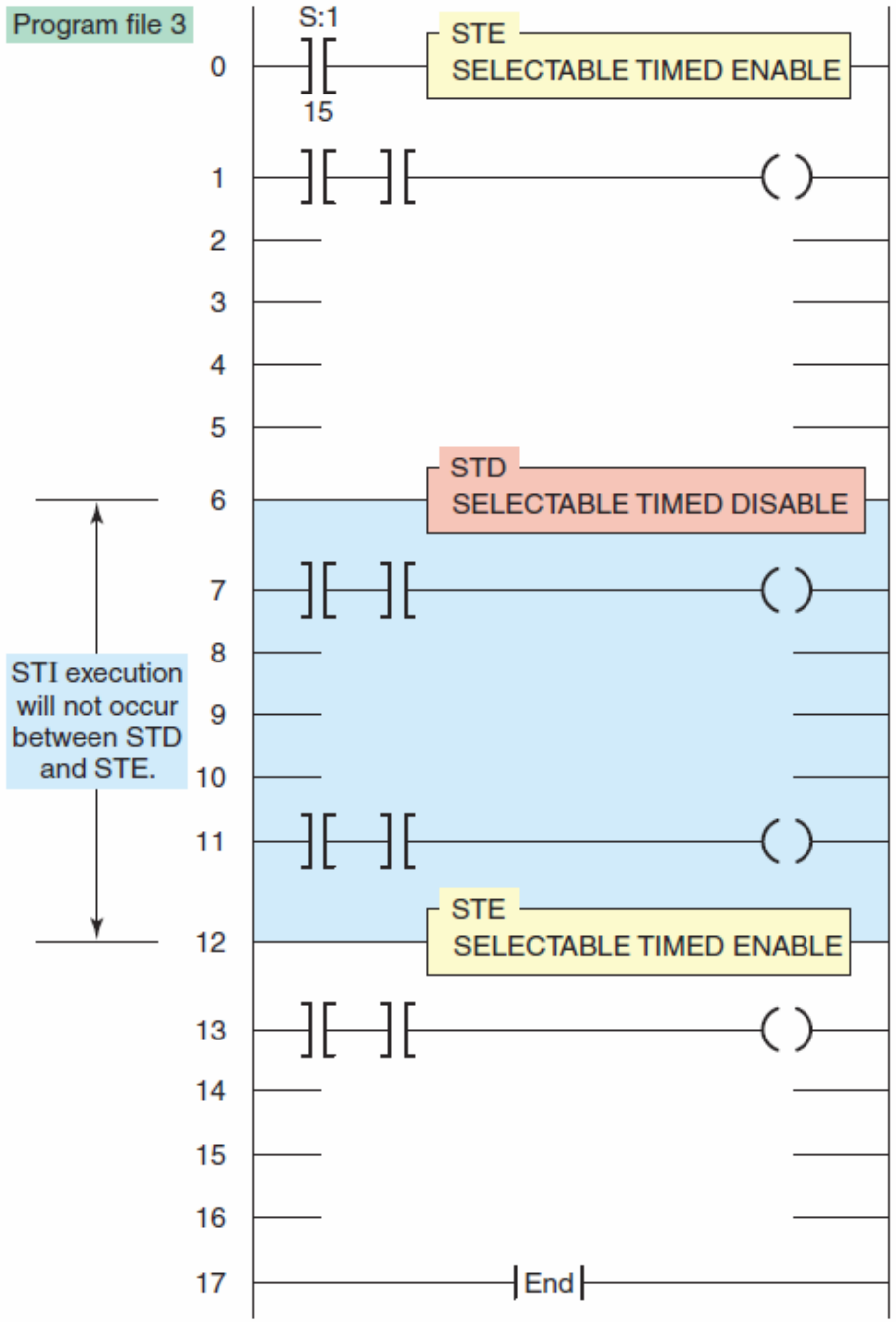
9.7

Selectable Timed Interrupt

The *selectable timed interrupt (STI)* instruction is used to *interrupt* the scan of the main program file automatically, on a *time basis*, to scan a *specified subroutine file*.

The *selectable timed disable (STD)* instruction is generally paired with the *selectable timed enable (STE)* instruction to create zones in which STI interrupts cannot occur.

The *selectable timed disable (STD)* instruction is generally paired with the *selectable timed enable (STE)* instruction to create zones in which STI interrupts cannot occur.



Fault Routine

A *fault subroutine file*, if used, determines how the processor responds to a programming error.

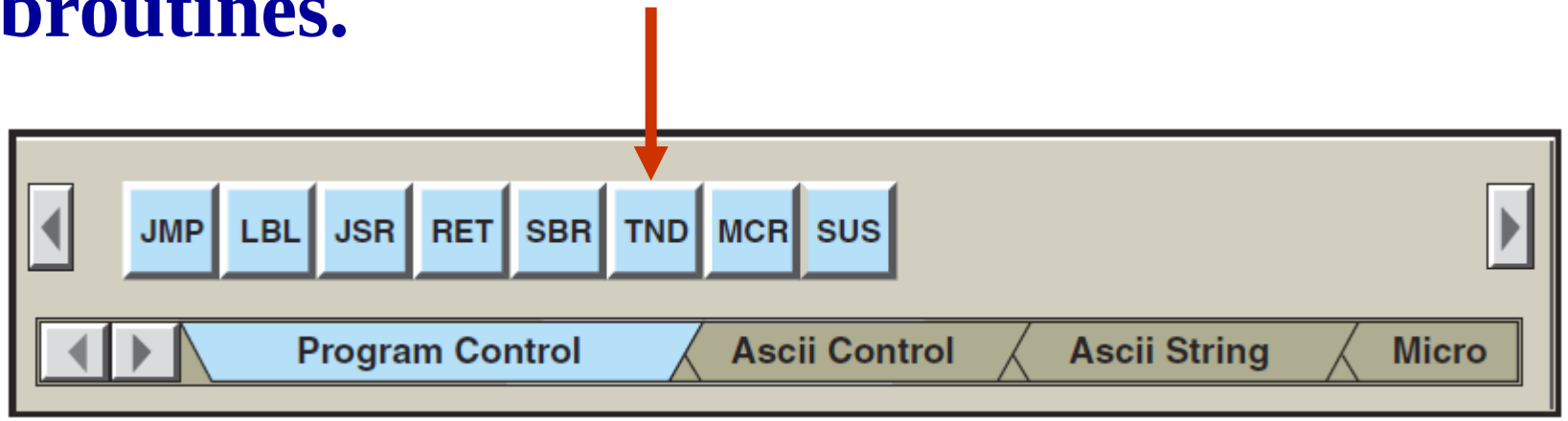
- When the processor detects a major fault, it looks for a fault routine. If a fault routine exists, it is executed; if one does not exist, the processor shuts down.

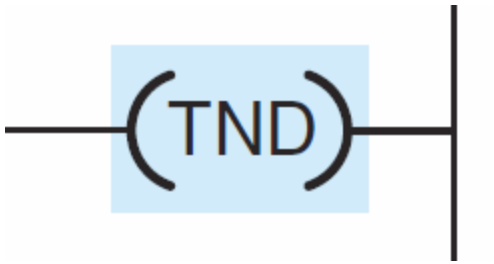
- When there is a fault routine, and the fault is recoverable, the fault routine is executed.

- If the fault is non-recoverable, the fault routine is scanned once and shuts down.

Temporary End Instruction

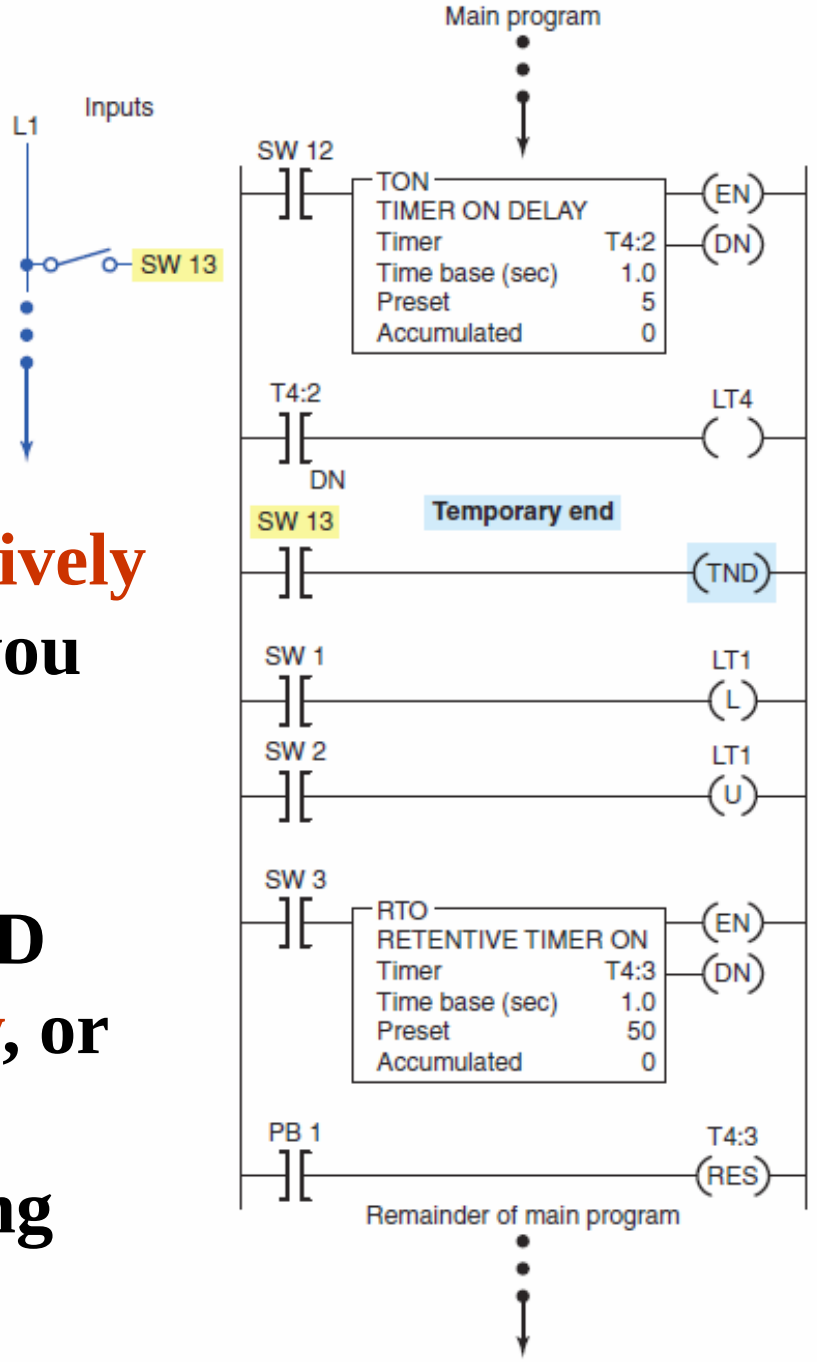
The *temporary end (TND)* instruction is used to progressively debug a program or conditionally omit the balance of your current program file or subroutines.





When rung conditions are true, this instruction stops the program scan, updates the I/O, and resumes scanning at rung 0 of the main program file.

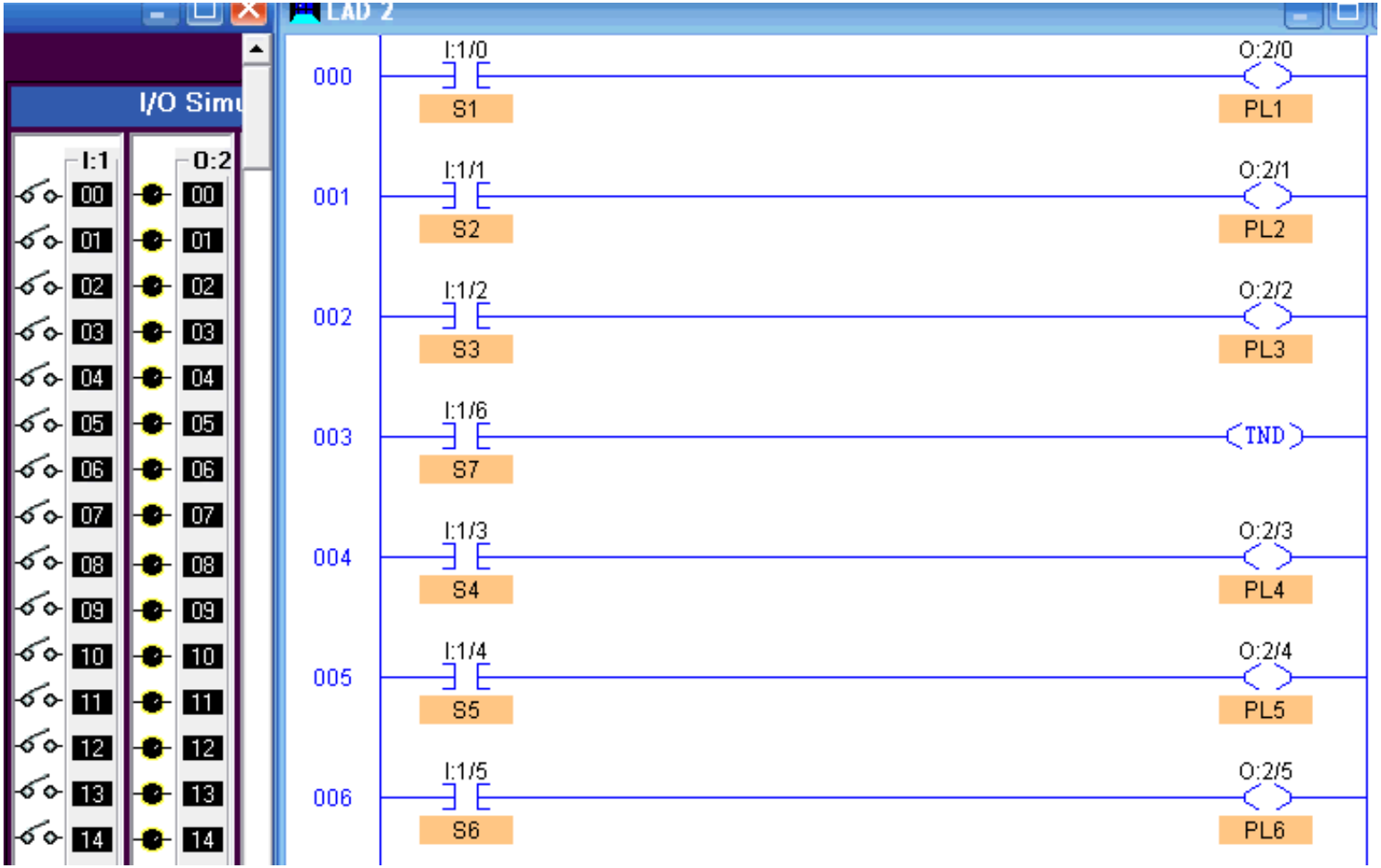
➢**The TND instruction lets your program run only up to this instruction.**



➢**You can move it progressively through your program as you debug each new section.**

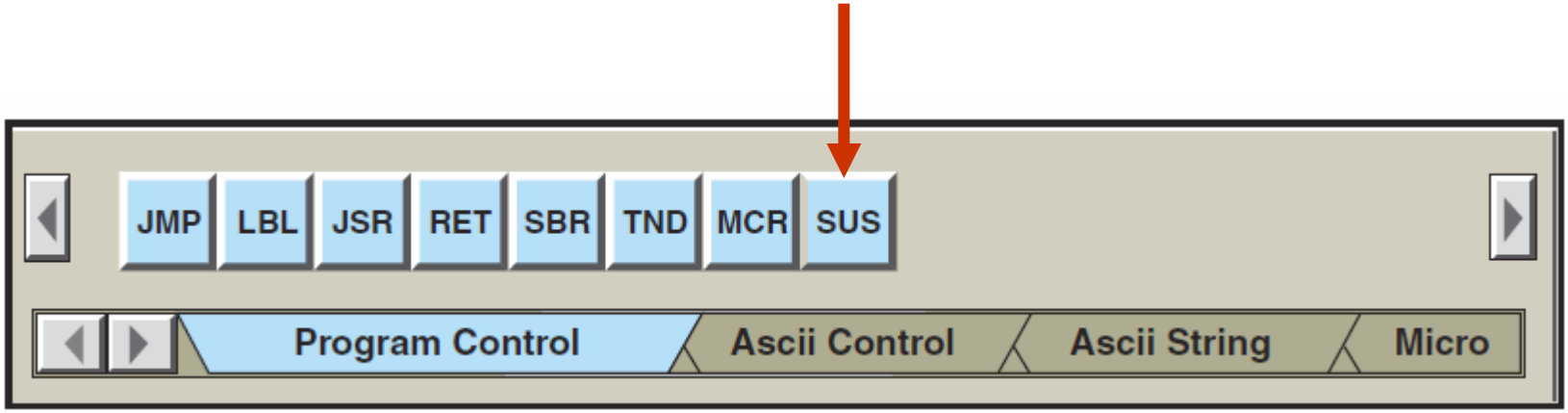
➢**You can program the TND instruction unconditionally, or you can condition its rung according to your debugging needs.**

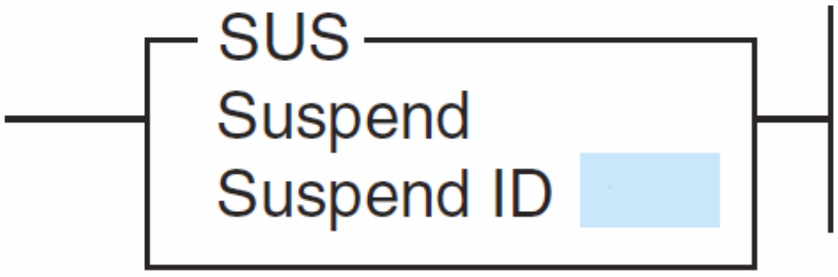
Simulated temporary end instruction program.

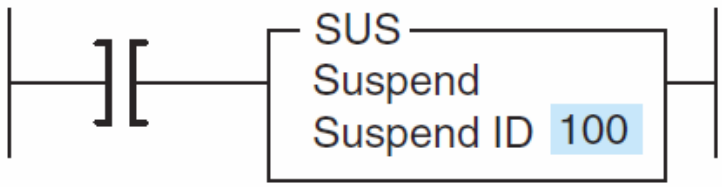


Suspend Instruction

The *suspend (SUS)* instruction is used to trap and identify specific conditions during system troubleshooting and program debugging.







- When the rung is **true**, the SUS output instruction places the controller in the suspend mode and the PLC immediately **terminates scan** cycling.
- All ladder logic **outputs** are **de-energized**, but other status files have the data present when the suspend instruction is executed.
- The SUS instruction writes the suspend ID number (**100**) to S:7 as it executes.
- You can include several SUS instructions in a program, each with a **different suspend ID**.