

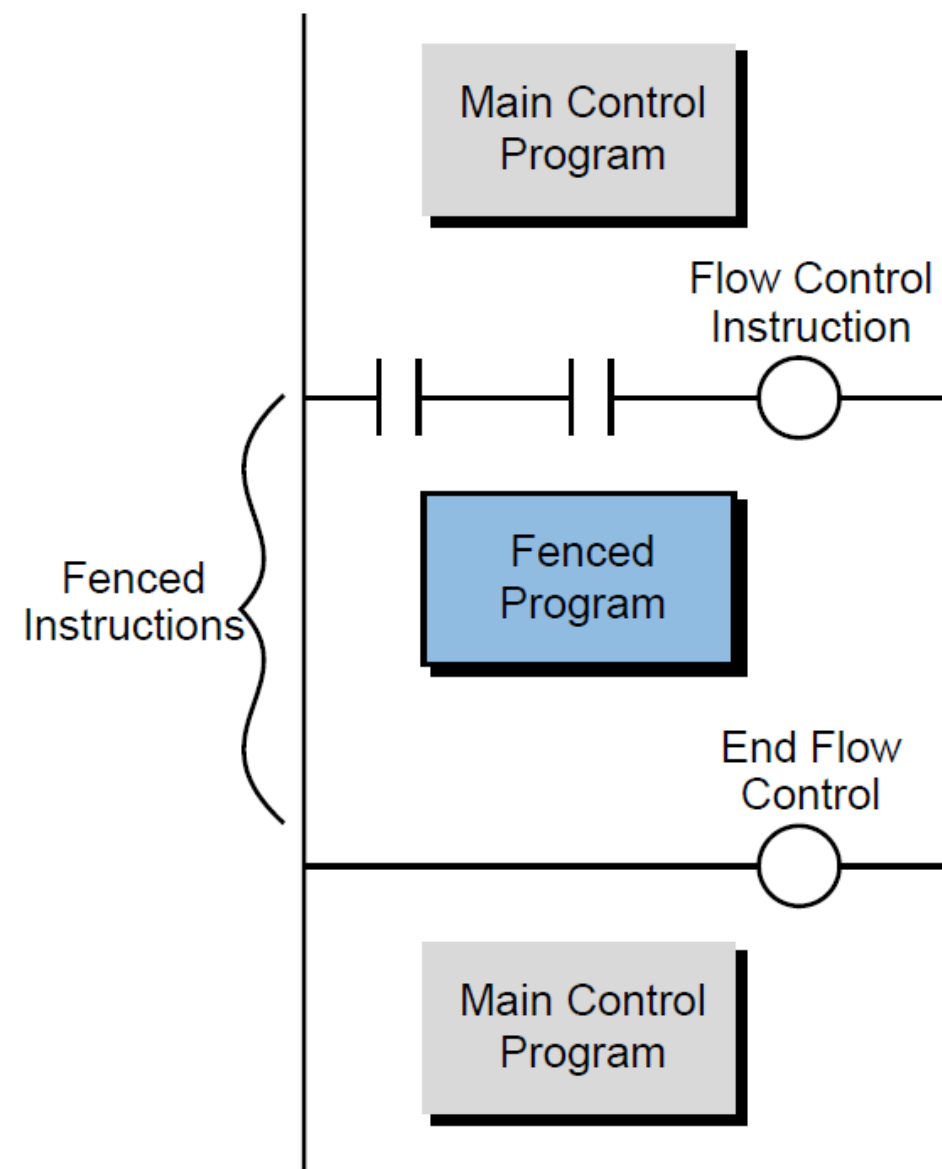
Chapter 8

Program/Flow Control and Sequential Step

8.1 Program/Flow Control

Master Control/Master Control Reset

- If the rung is TRUE, the fenced program (routine) is executed.
- If the rung is FALSE, the fenced program is bypassed.
- These instructions are generally used in pairs.
- The first instruction starts the flow control change, sending the PLC to a special routine of instructions in another section of the control program.
- The other instruction returns the PLC to the program it was running when the flow control change occurred.



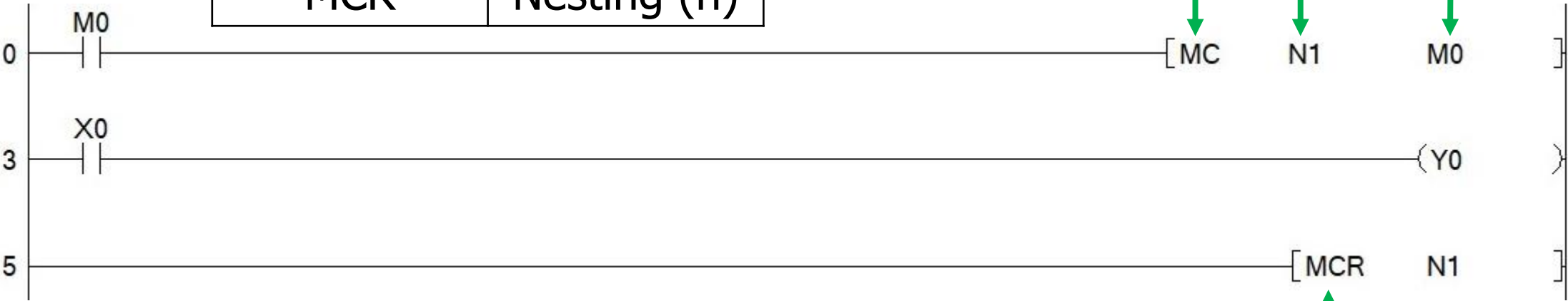
8.1 Program/Flow Control

Master Control/Master Control Reset

MC	Nesting (n)	Device (Bit)
----	-------------	--------------

Fenced Master Control Zone

MCR	Nesting (n)
-----	-------------



Master Control

Order number of Fenced Program

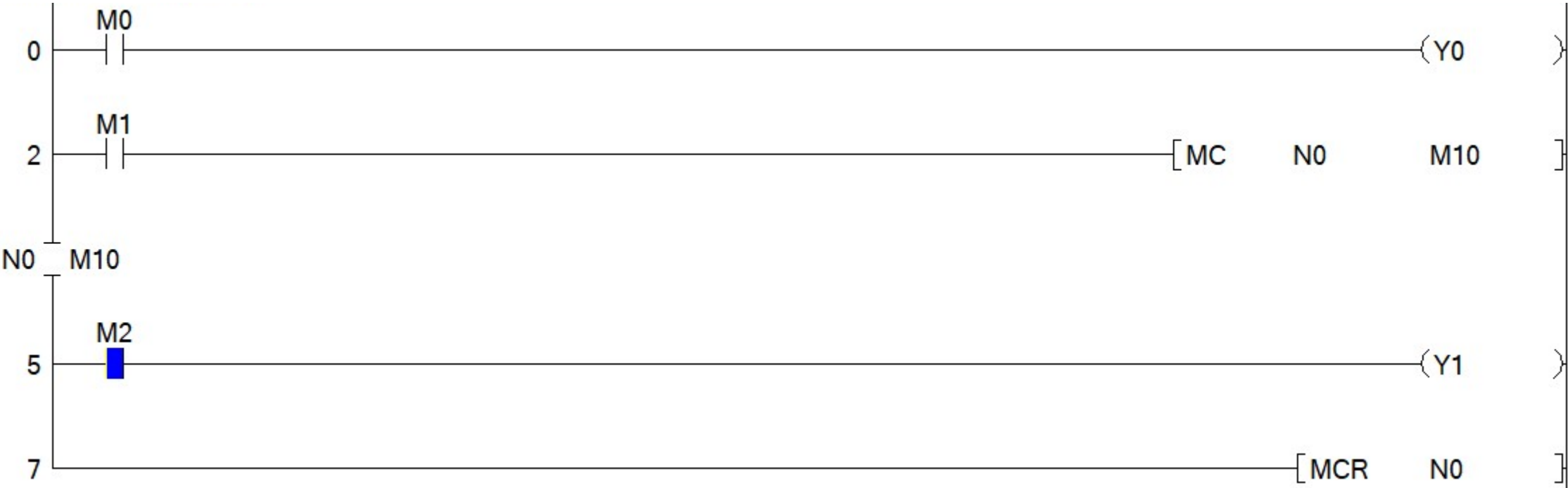
Status Bit

Master Control Reset

8.1 Program/Flow Control

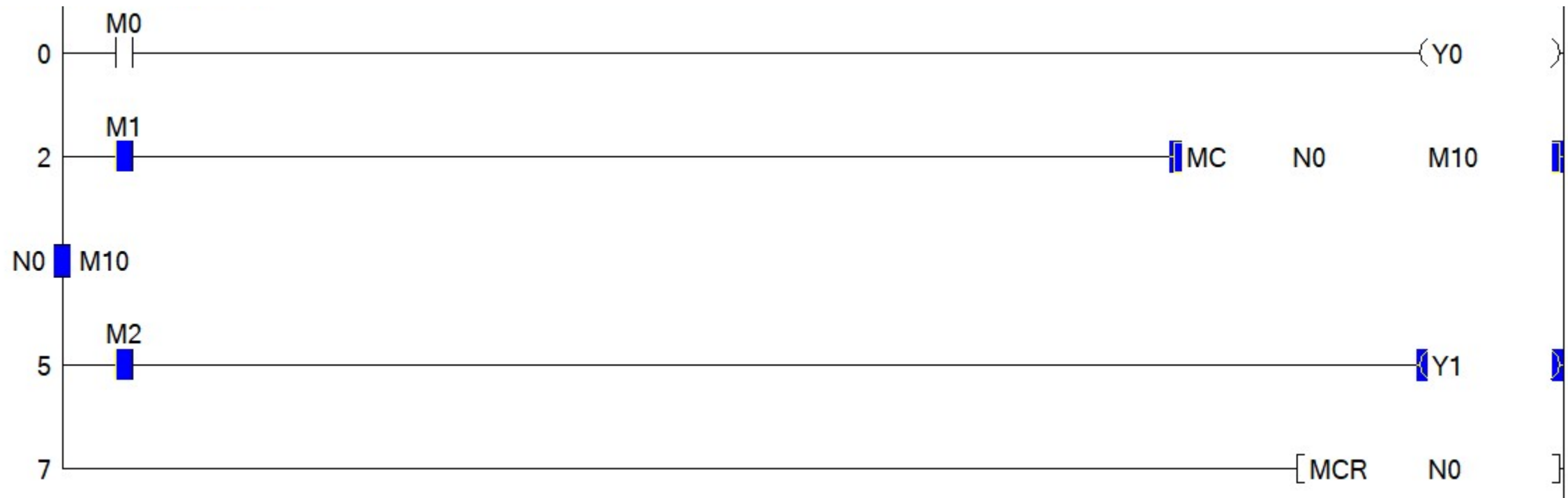
Master Control/Master Control Reset

The fifth rung is ignored while the MC is not activated.
M10 device indicates the status of MC.



Master Control/Master Control Reset

If the MC is activated, the fenced program is implemented.



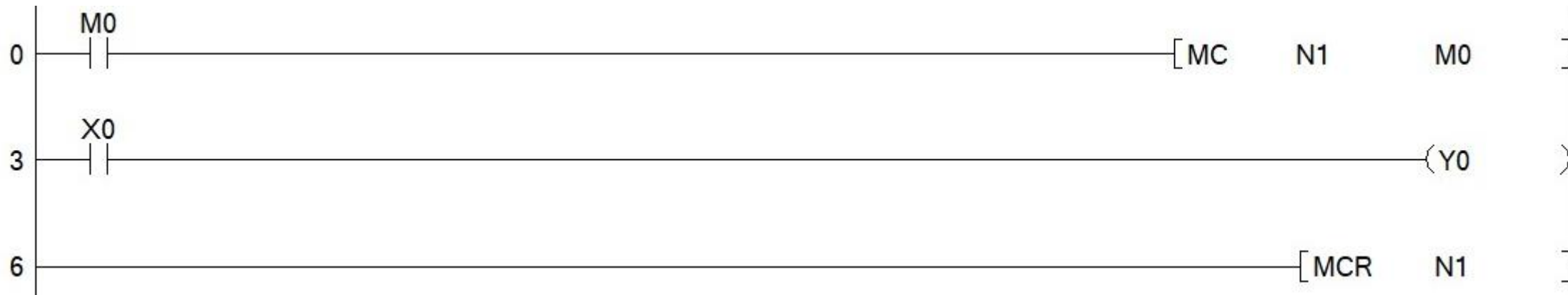
8.1 Program/Flow Control

Pointer

- A pointer instruction identifies the ladder rung where is the target destination of a condition jump (CJ).
- The pointer indicates the ladder rung which is implemented while the rung of CJ is ON.

Pointer

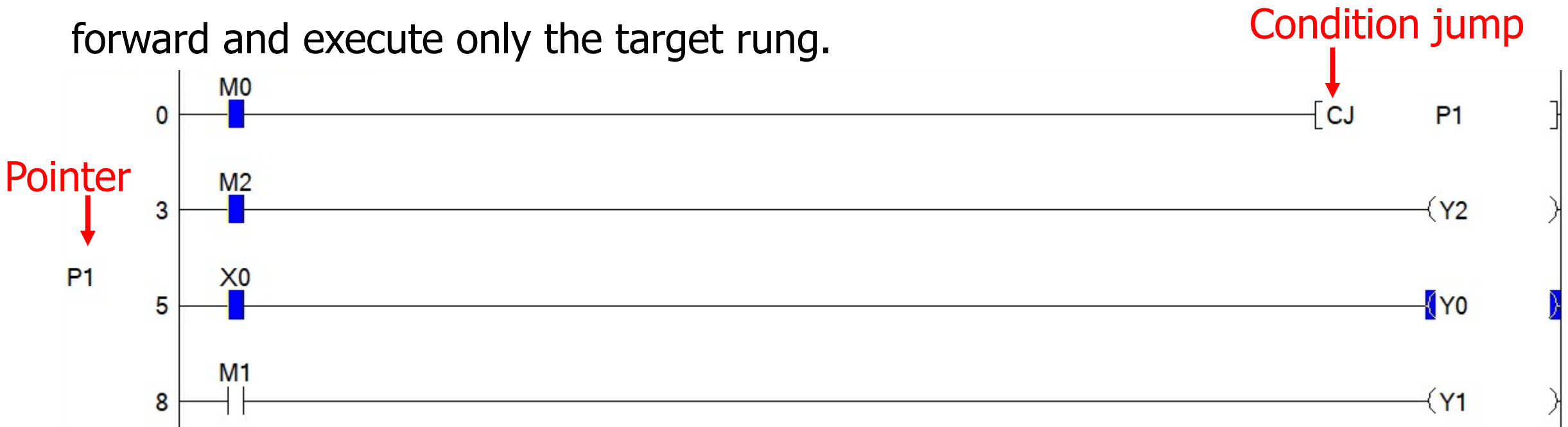
P1



8.1 Program/Flow Control

CJ Instruction

- Condition jumps instruction allows the control program sequence to be altered if certain conditions exist.
- If the rung condition is TRUE, the jump to the pointer tells the processor to jump forward and execute only the target rung.



8.1 Program/Flow Control

CJ Instruction



- The third rung is ignored while the condition jump instruction is ON.
- There is only the fifth rung implemented where the CJ instruction jumps to.

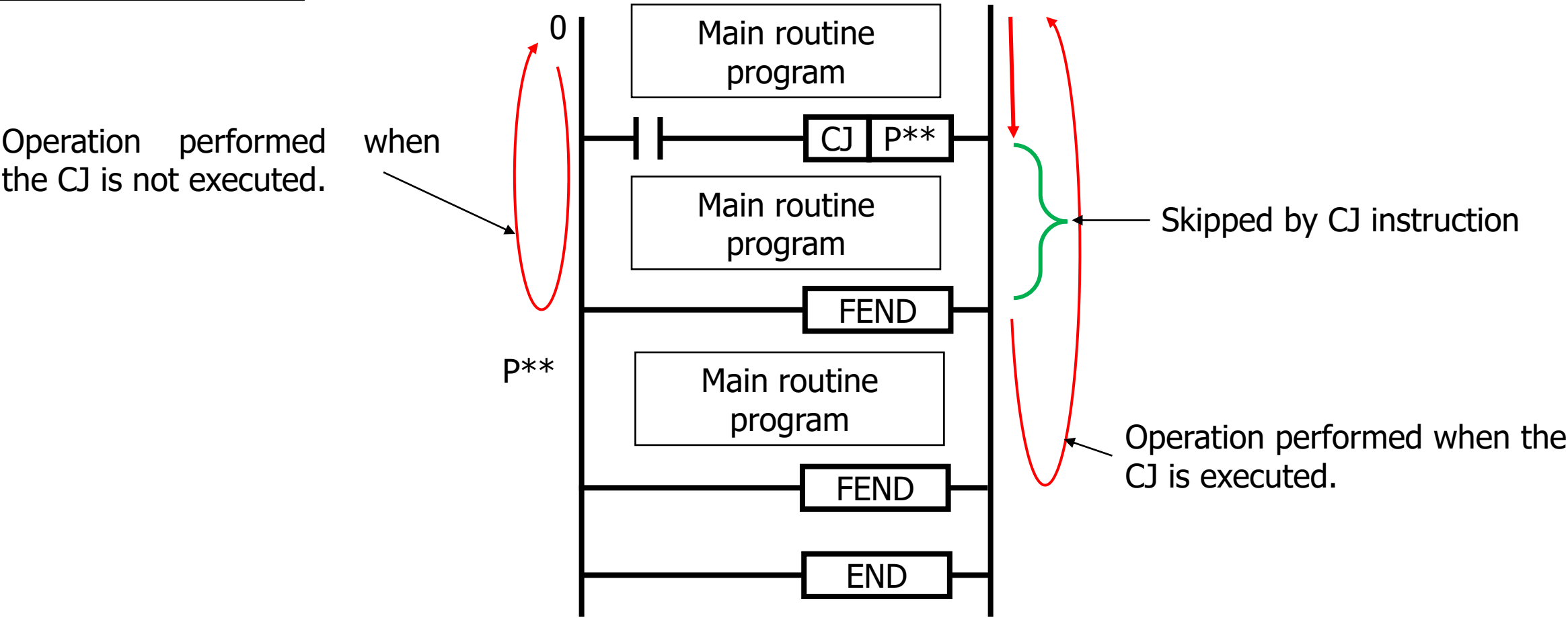
8.1 Program/Flow Control

FEND Instruction

- The FEND instruction is used in cases where the CJ instruction or other instructions are used to cause a branch in the sequence program operations
- Execution of the FEND instruction will cause the CPU module to terminate the program it was executing.
- Even sequence programs following the FEND instruction can be displayed in ladder display at a peripheral device.

8.1 Program/Flow Control

FEND Instruction



8.1 Program/Flow Control

FEND Instruction



Without FEND instruction

8.1 Program/Flow Control

FEND Instruction



With FEND instruction

8.2 Sequential Step

- Many applications require the working flow in sequence.
- The flow control can be divided into many steps.
- The steps are executed sequentially, one by one. Each step performs a task.

Logical equation in step sequence:

$$S_i = (S_{i-1} \cdot C_i) \cdot \overline{S_{i+1}}$$

S_i stands for i^{th} step.

S_{i-1} stands for $(i - 1)^{th}$ step.

S_{i+1} stands for $(i + 1)^{th}$ step.

C_i stands for the i^{th} step condition.

8.2 Sequential Step

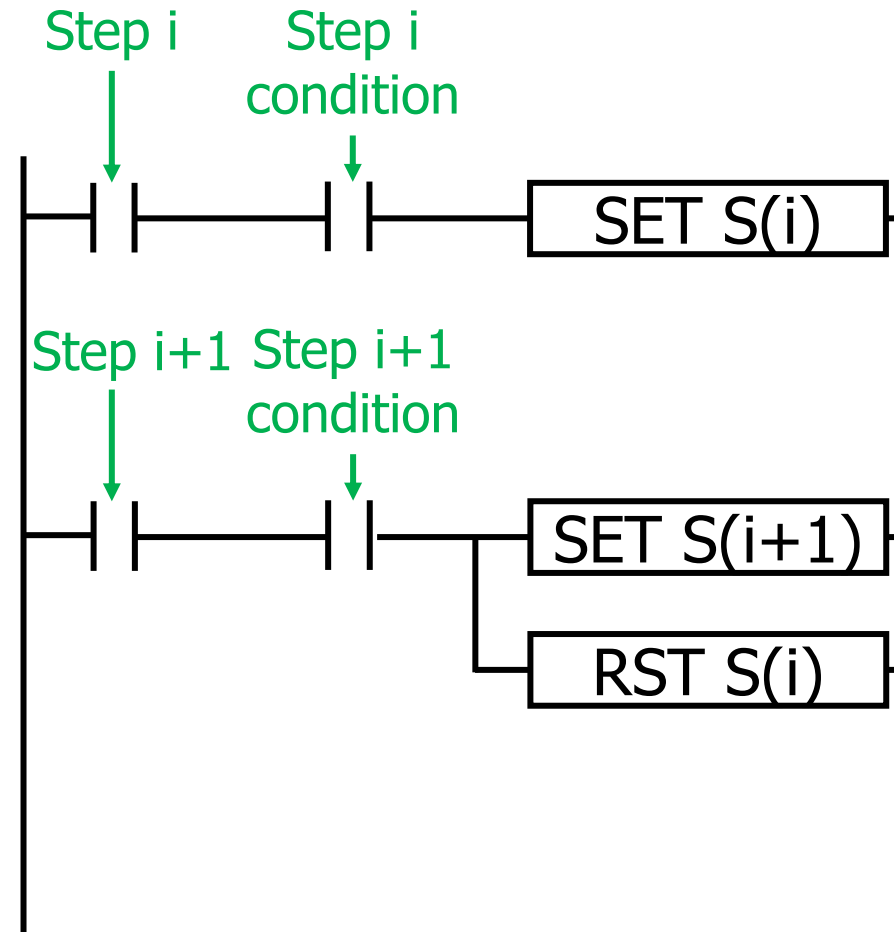
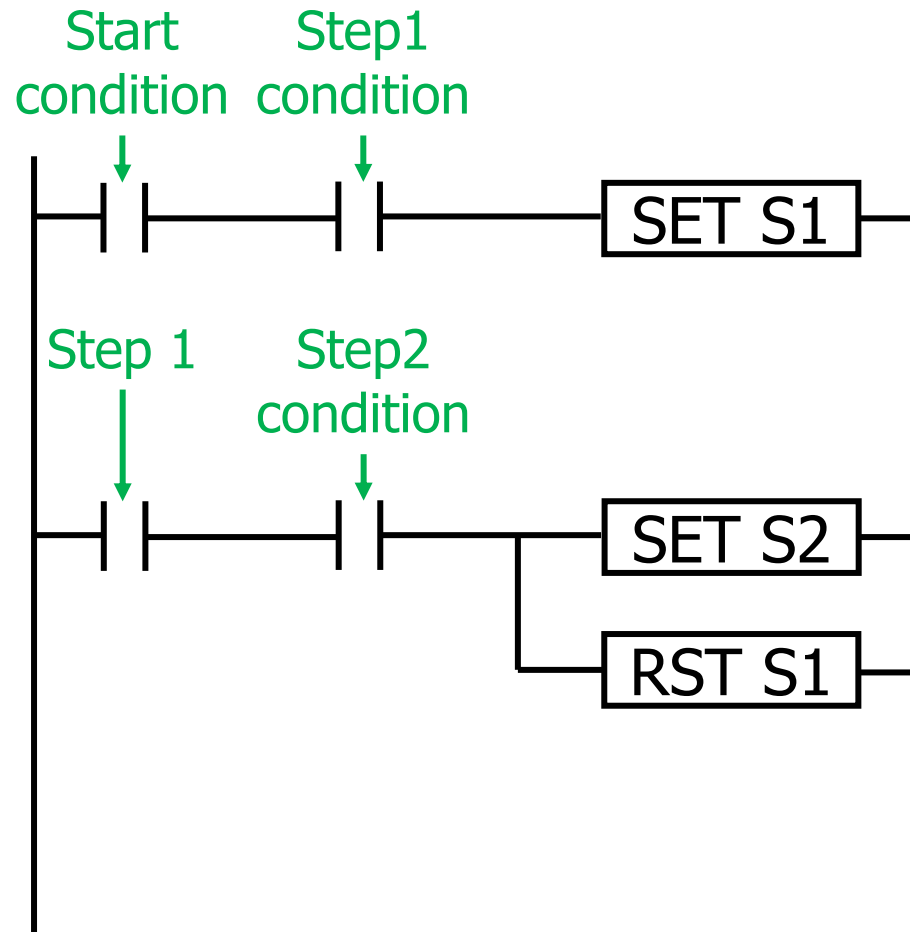
- i^{th} step is activated when the $(i - 1)^{th}$ step was activated and the i^{th} step condition is ON.
- i^{th} step is deactivated when the $(i + 1)^{th}$ step is activated.

$$S_i = (S_{i-1} \cdot C_i) \cdot \overline{S_{i+1}}$$

- The status of steps are indicated by internal devices.
- There are some techniques to construct the sequential steps.

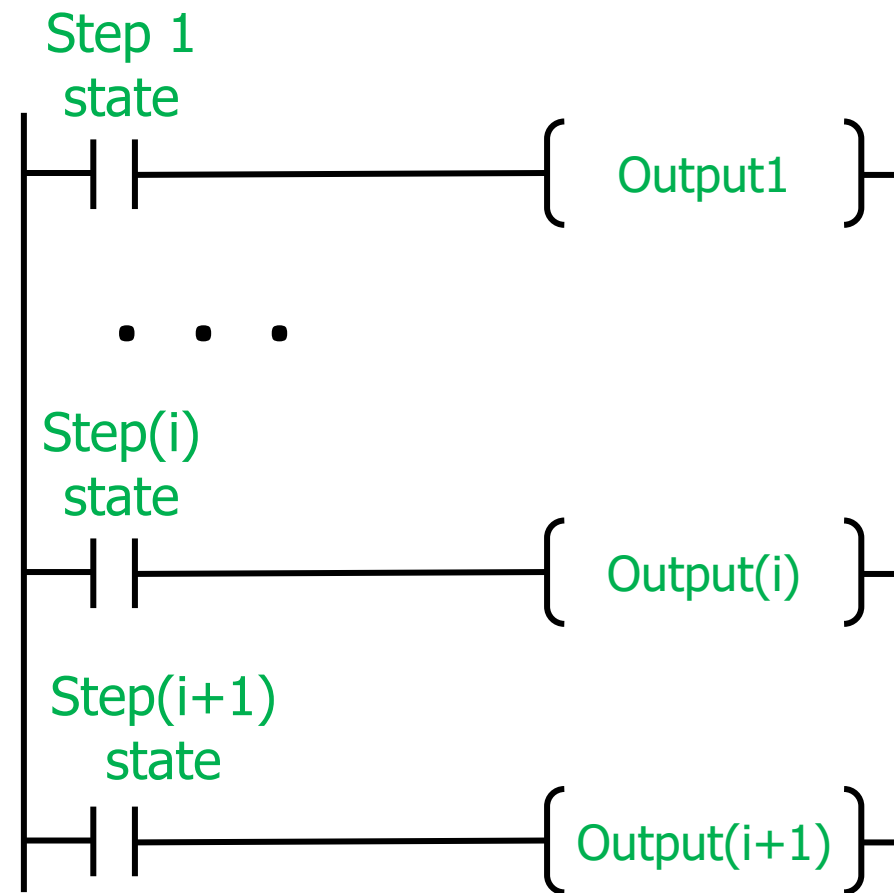
8.2 Sequential Step

○ Using SET/RESET instruction



8.2 Sequential Step

- Using SET/RESET instruction



8.2 Sequential Step

○ Using SET/RESET

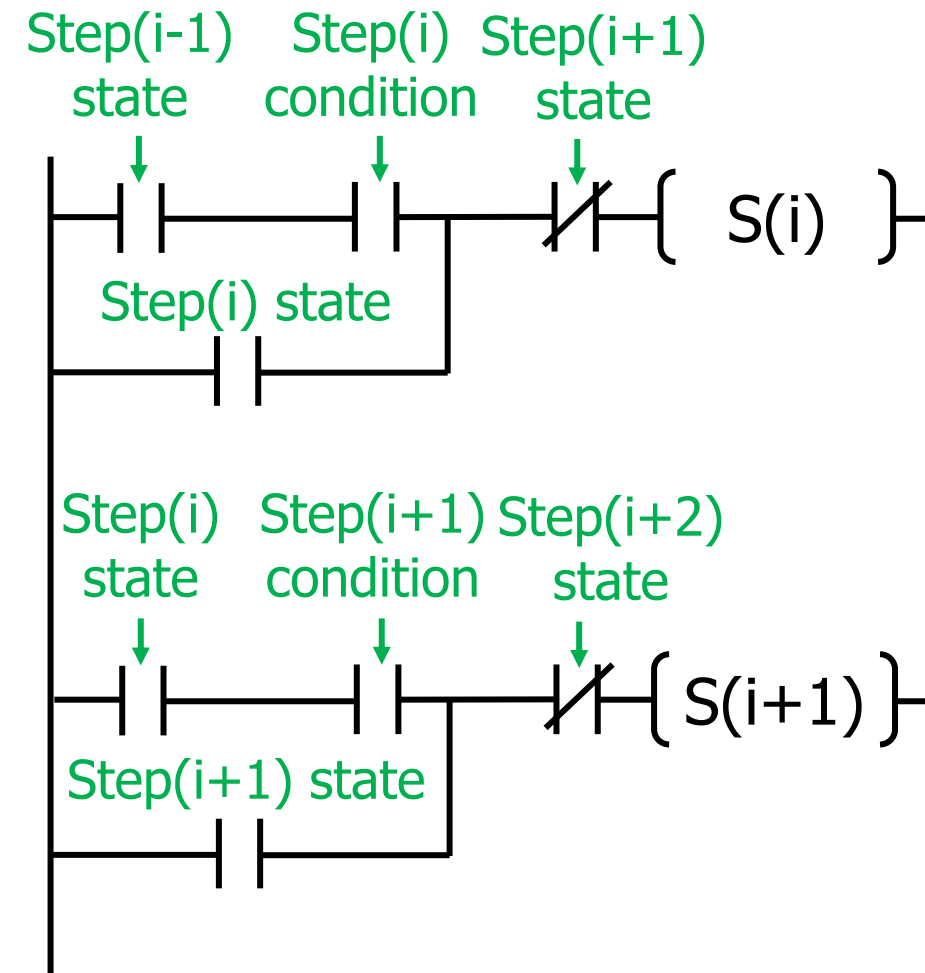


8.2 Sequential Step

○ Using SET/RESET

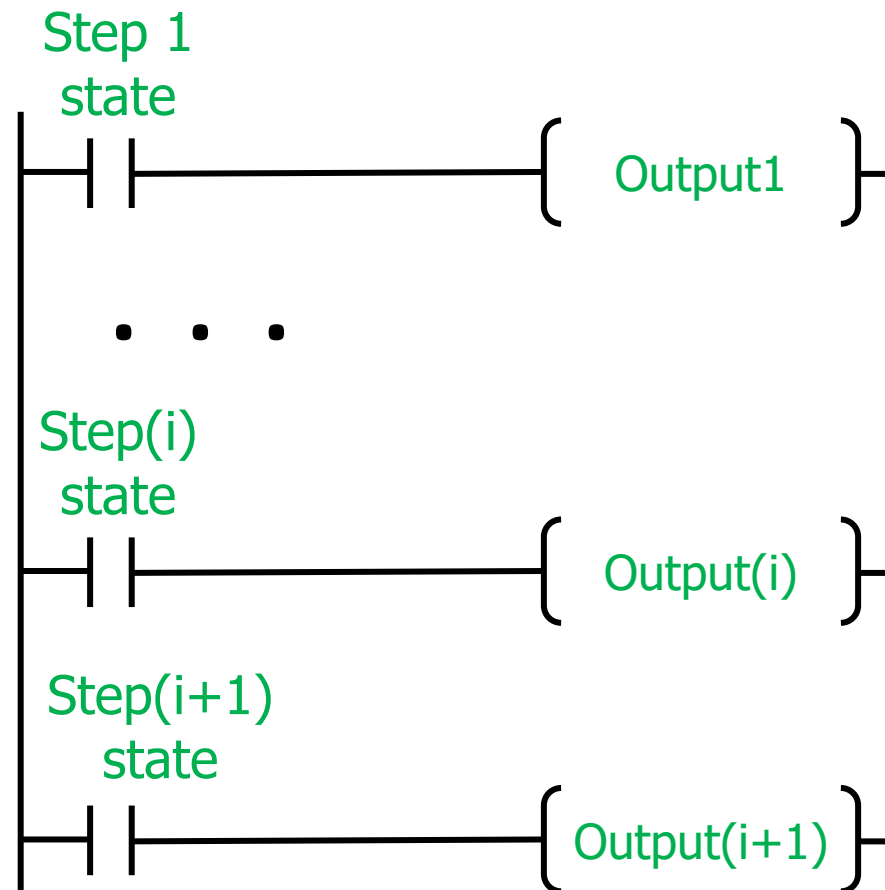


- Using Output coil



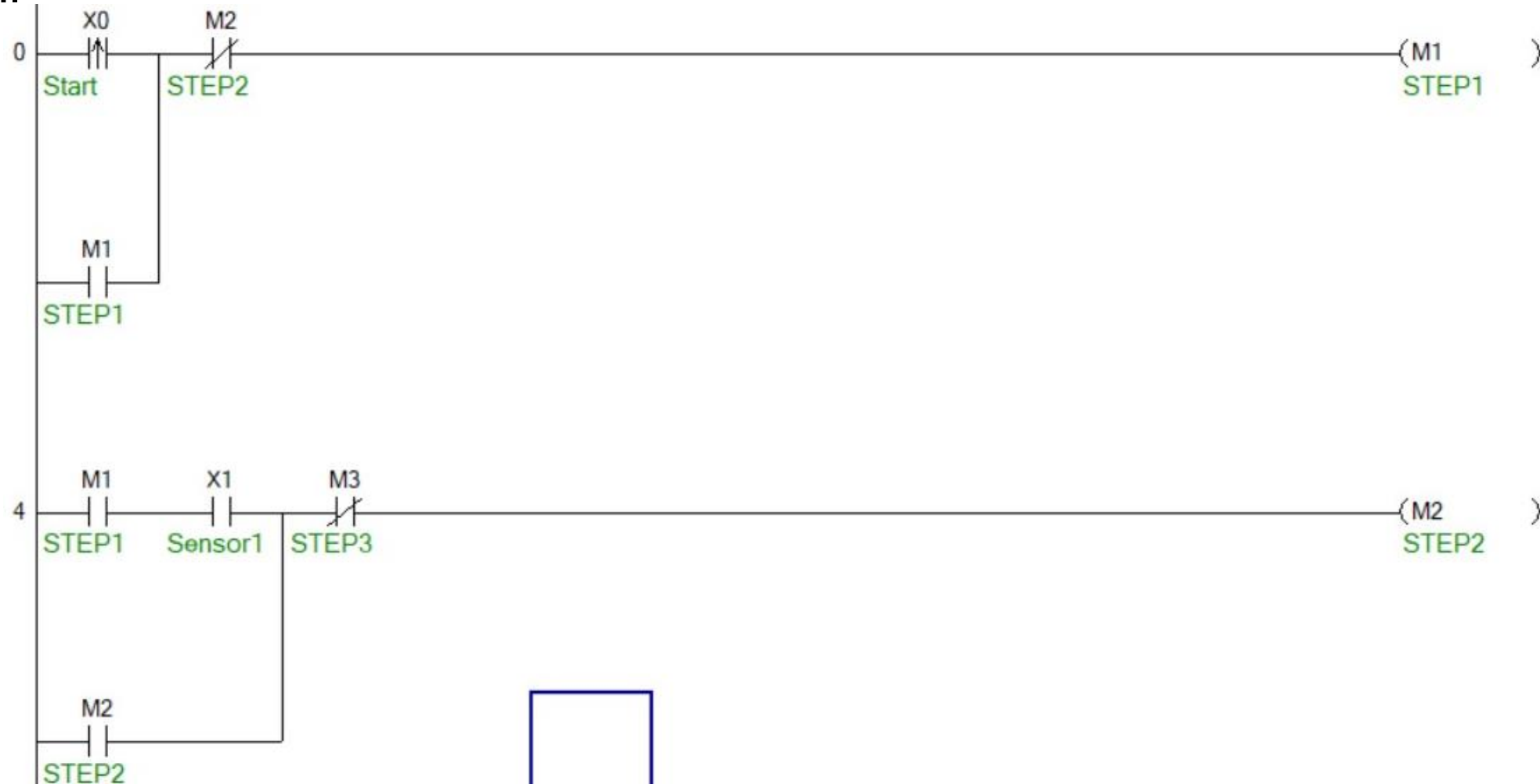
8.2 Sequential Step

- Using Output coil



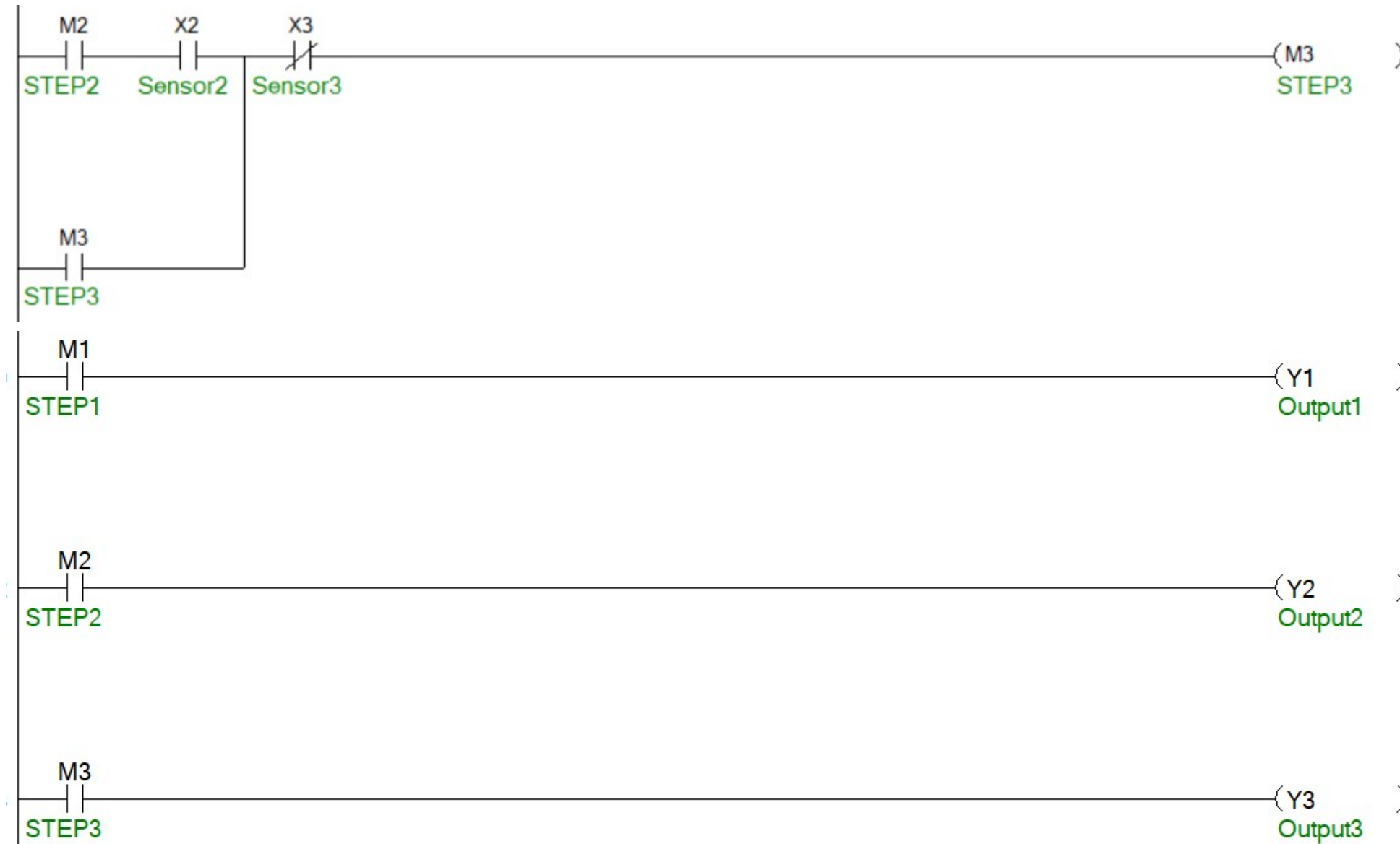
8.2 Sequential Step

○ Using Output coil



8.2 Sequential Step

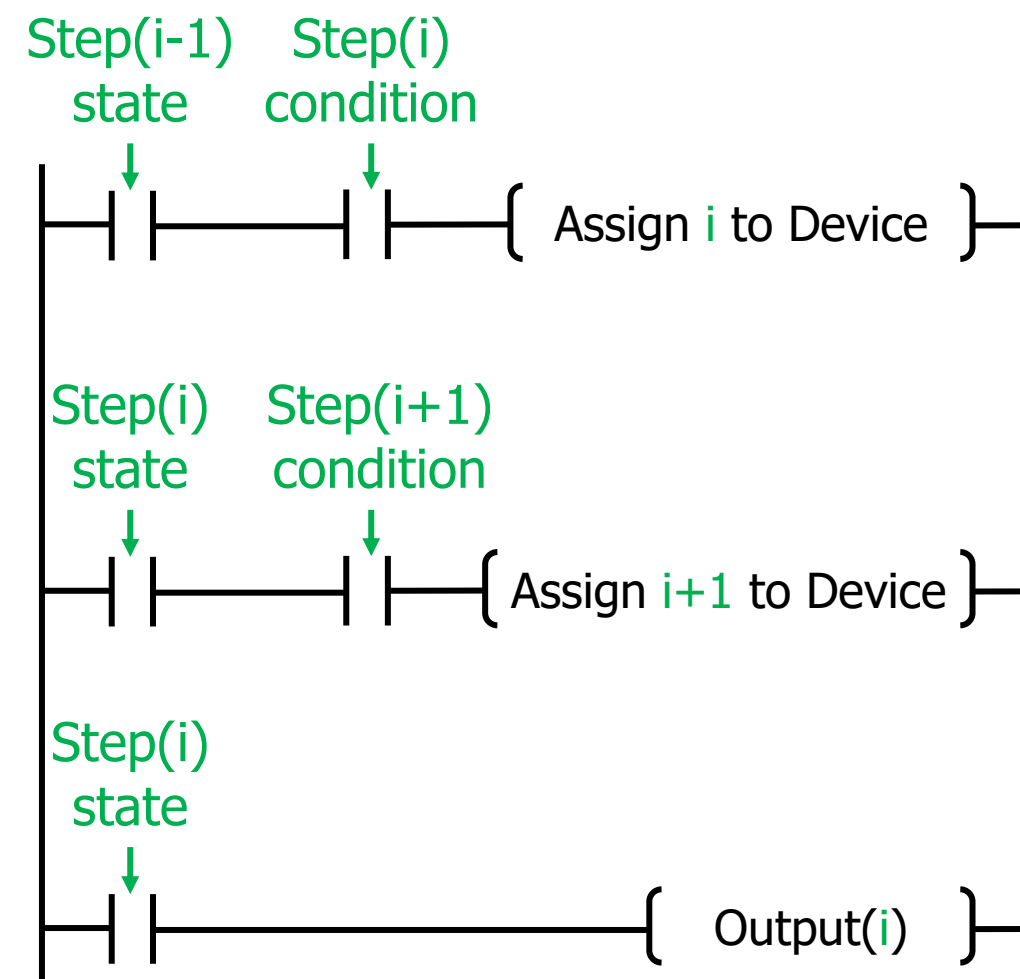
- Using Output coil



8.2 Sequential Step

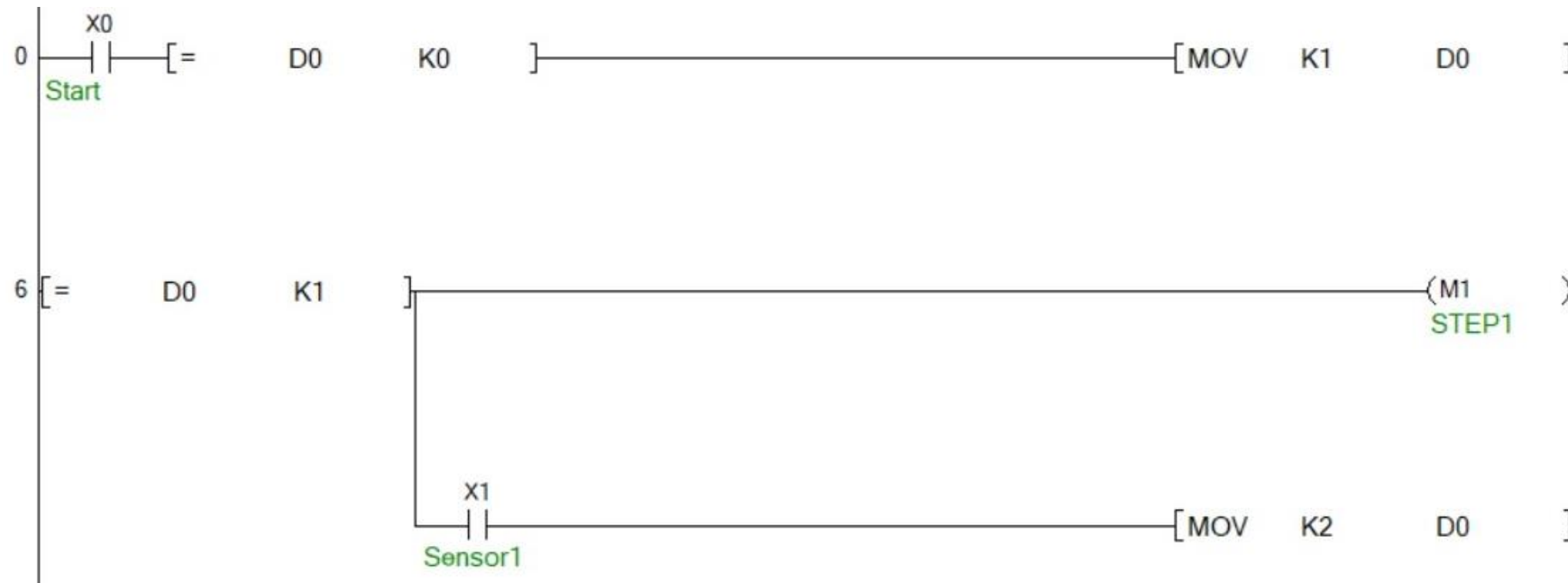
○ Using Register

- Use an internal device as a step index.
- The value of the step index indicates the current step.
- Since the value of the step index is unique, the steps do not overlap.



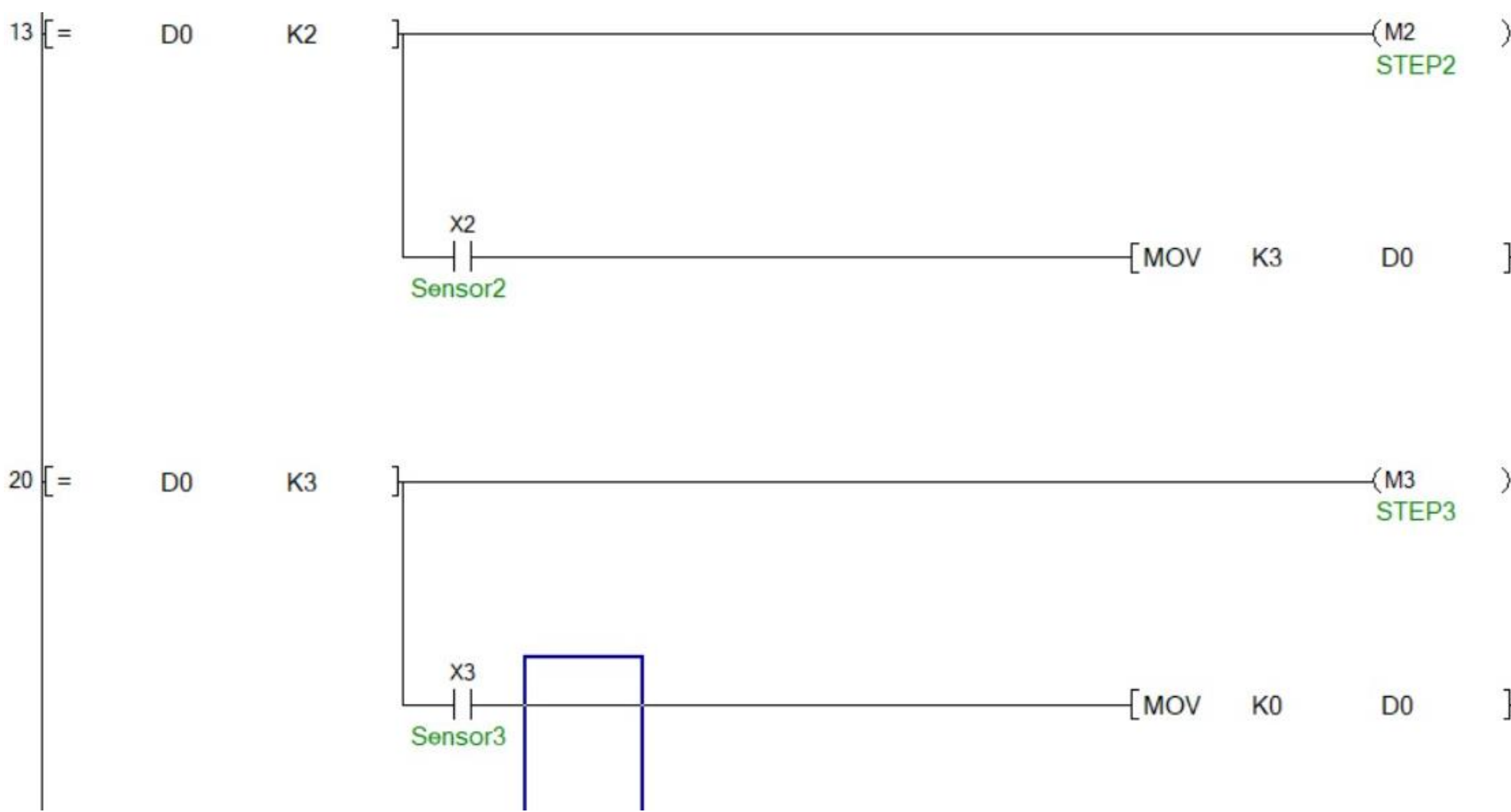
8.2 Sequential Step

○ Using Register



8.2 Sequential Step

○ Using Register



8.2 Sequential Step

Start: X0

Stop: X1

FS: X2

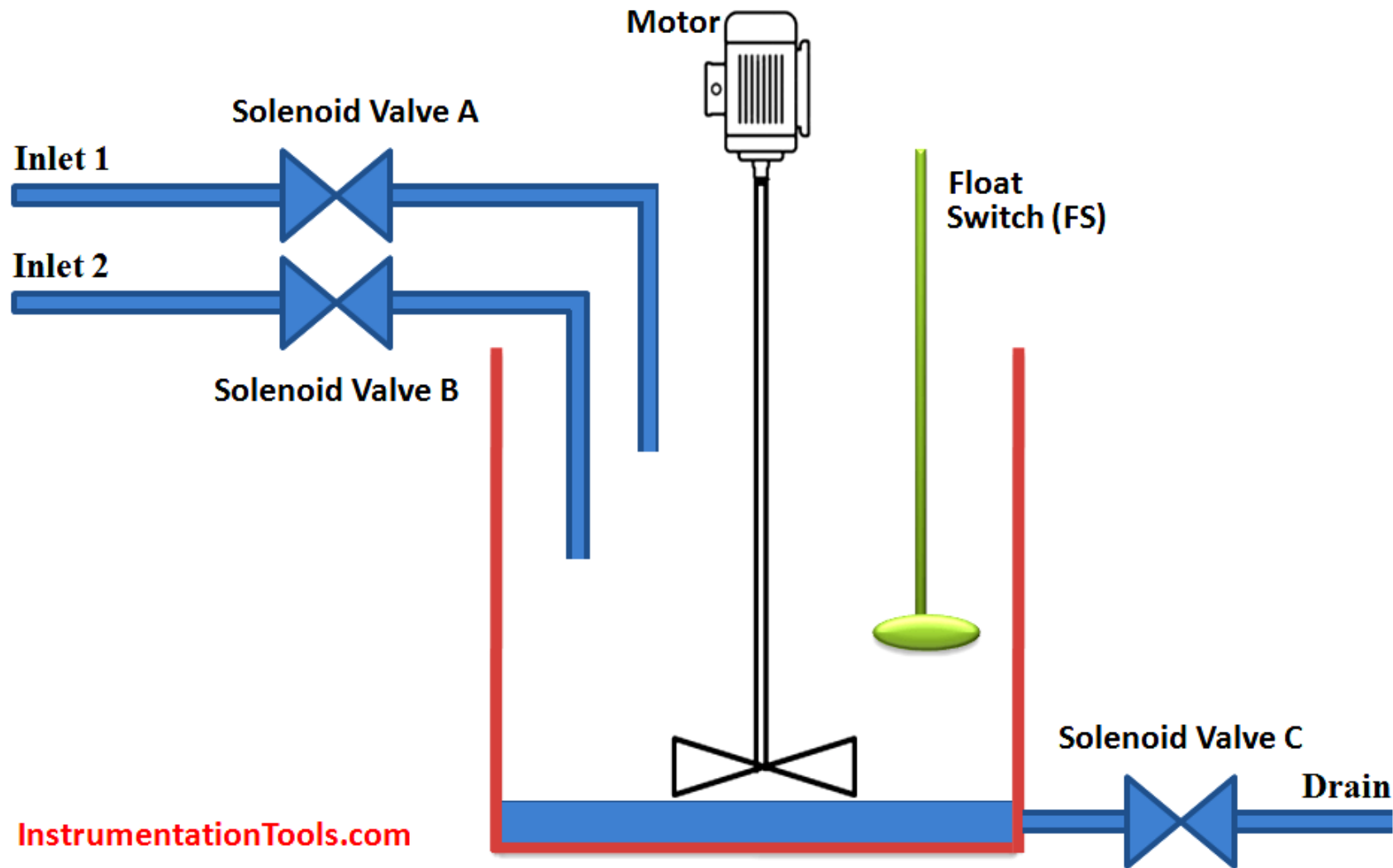
Valve A: Y20

Valve B: Y21

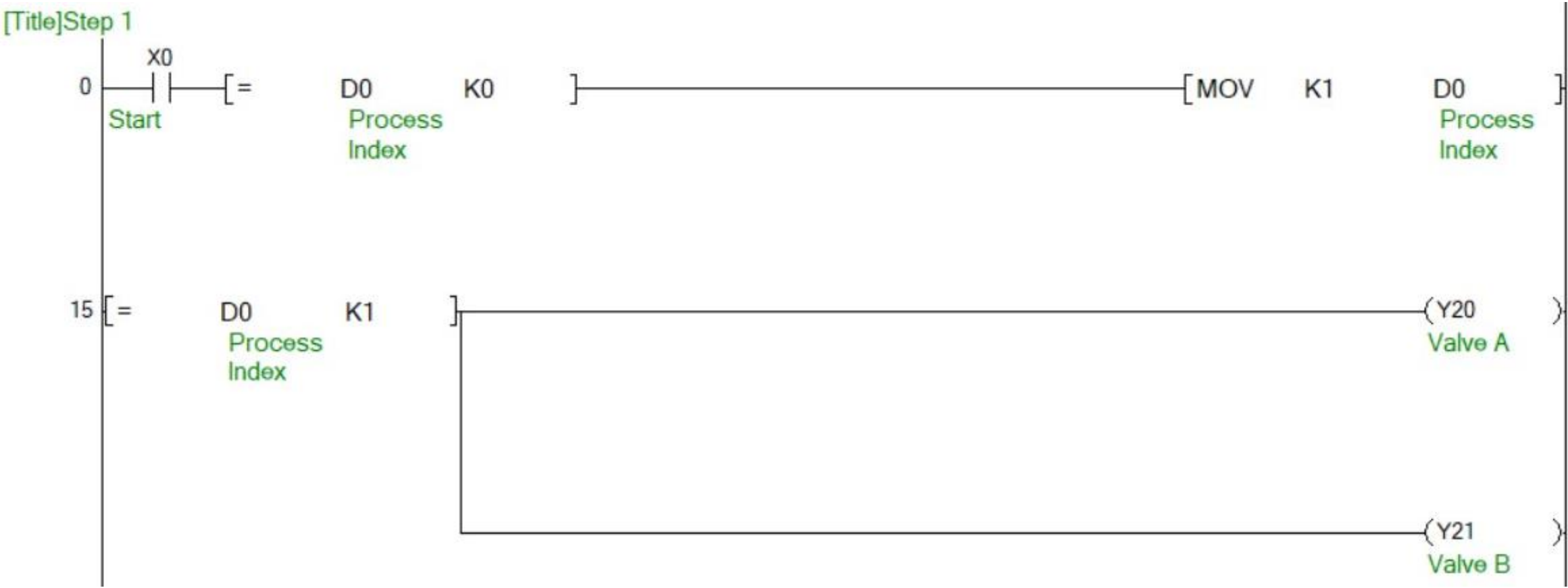
Motor: Y22

Valve C: Y23

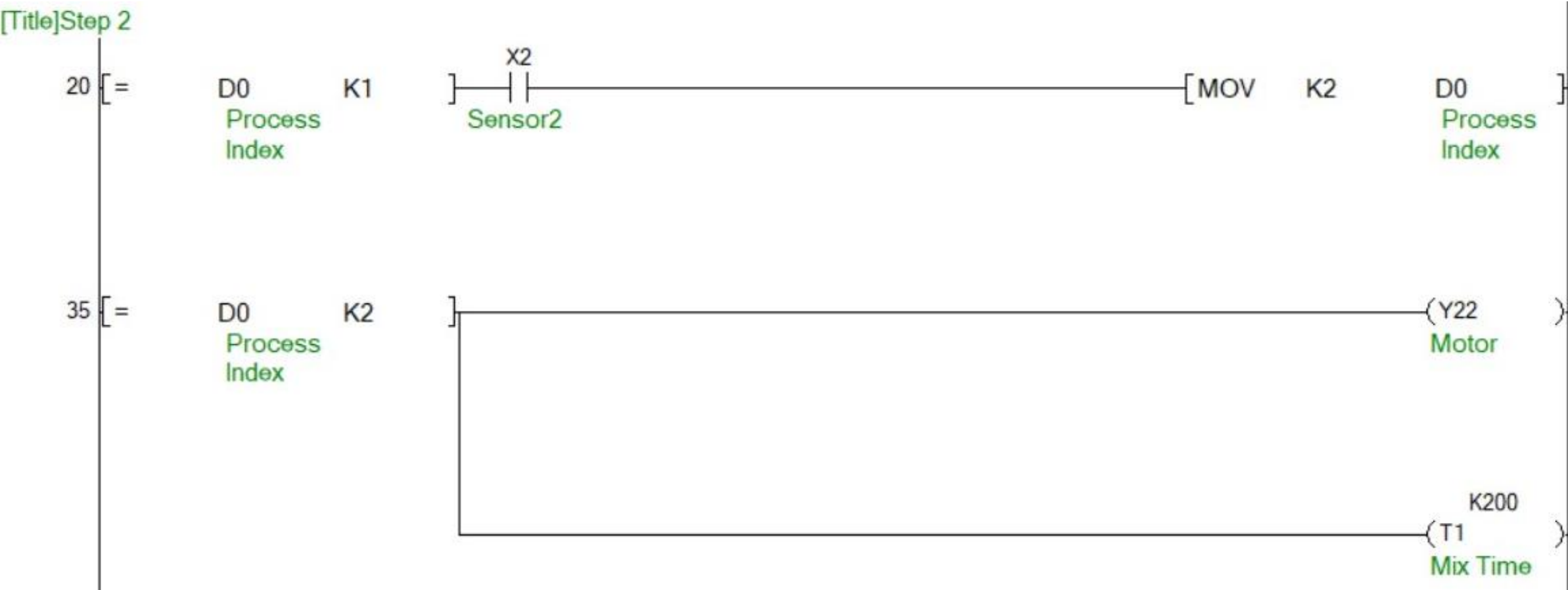
On: Open/Run
Off: Close/Stop



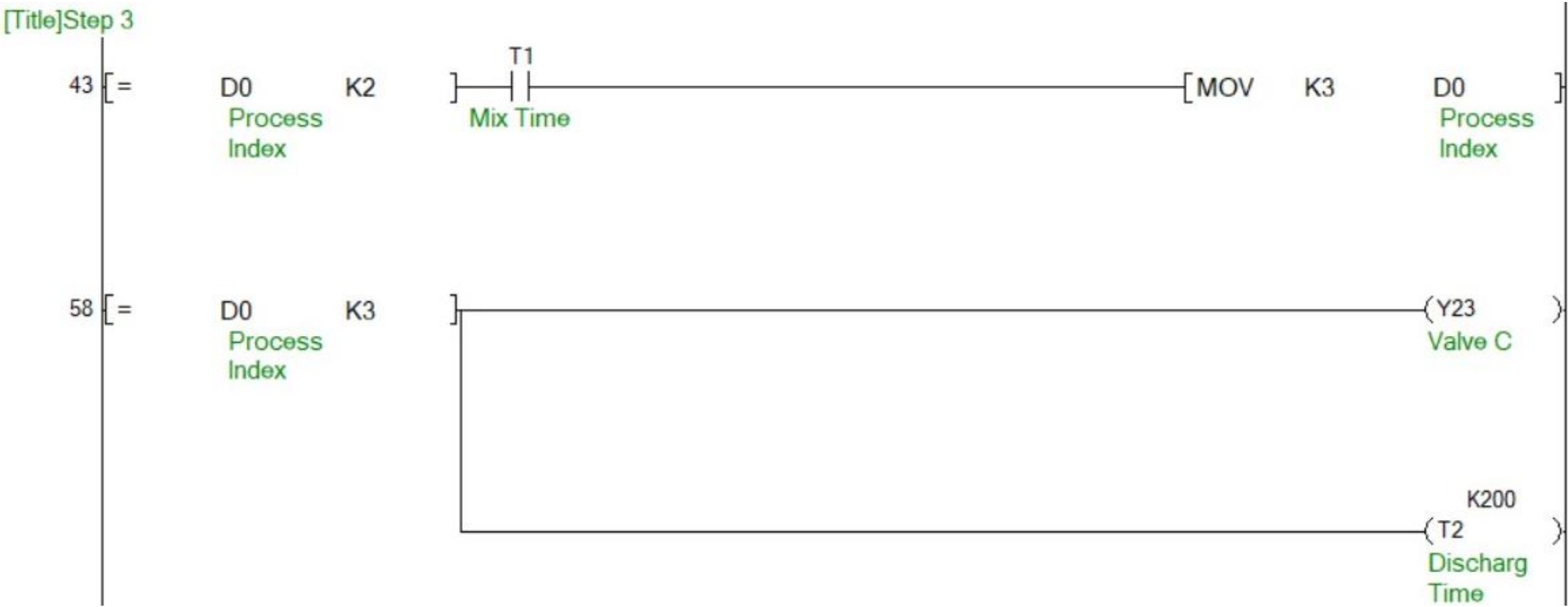
8.2 Sequential Step



8.2 Sequential Step



8.2 Sequential Step



8.2 Sequential Step

