LAD Language

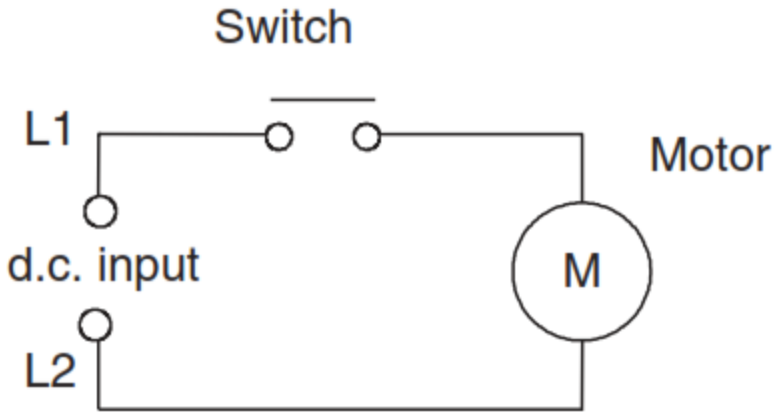
Lex 3

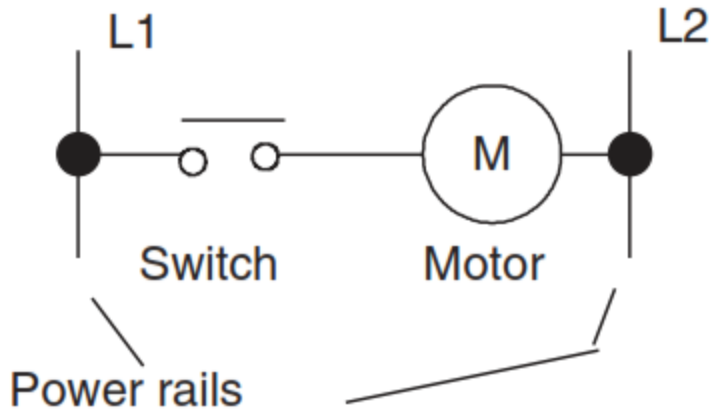
Why LAD

- Programming can be made even easier by the use of the so-called high-level languages, such as C, BASIC, Pascal, FORTRAN, and COBOL.
- These languages use prepackaged functions, represented by simple words or symbols descriptive of the function concerned. For example, with C language the symbol & is used for the logic AND operation.
- However, the use of these methods to write programs requires some skill in programming, and PLCs are intended to be used by engineers without any great knowledge of programming.
- As a consequence, ladder programming (LAD) was developed as a means of writing programs that can then be converted into machine code by software for use with the PLC microprocessor.
- This method of writing programs became adopted by most PLC manufacturers, but each tended to develop its own version, and so an international standard has been adopted for ladder programming and, indeed, all the methods used for programming PLCs. The standard, published in 1993, is IEC 1131-3.
- Functional block programming (FBD) is another method of programming.

Ladder Diagrams

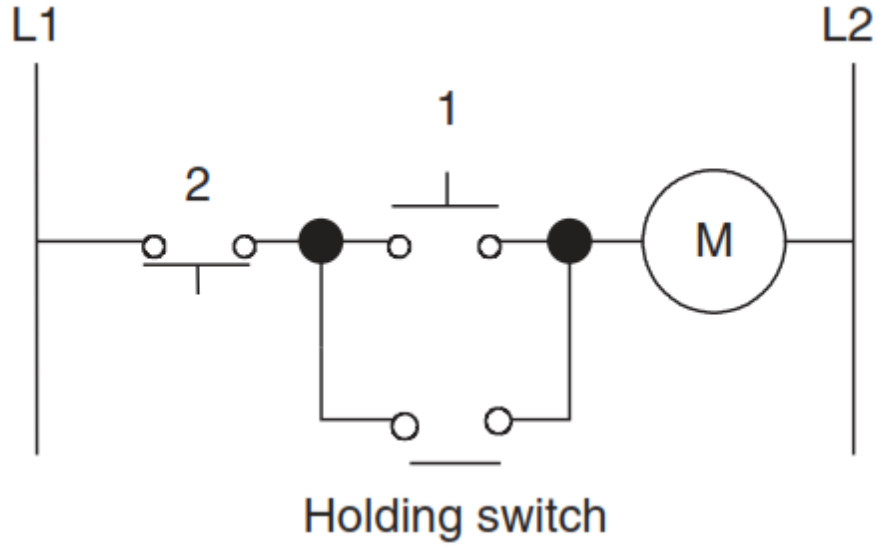
- As an introduction to ladder diagrams, consider the simple wiring diagram for an electrical circuit in Figure a. The diagram shows the circuit for switching on or off an electric motor.
- We can redraw this diagram in a different way, using two vertical lines to represent the input power rails and stringing the rest of the circuit between them. Figure b shows the result.
- Both circuits have the switch in series with the motor and supplied with electrical power when the switch is closed. The circuit shown in Figure b is termed a ladder diagram.





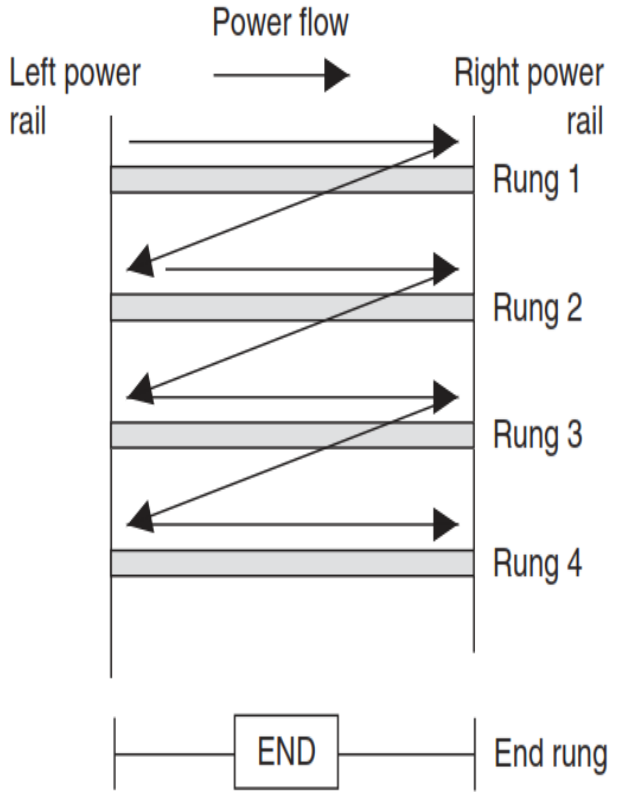
Ladder Diagrams

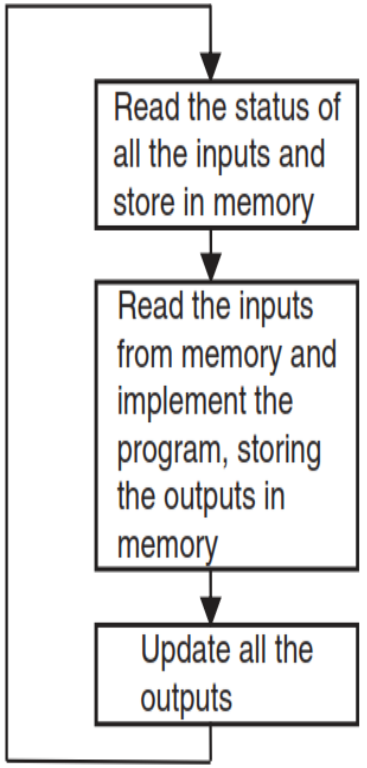
- Figure shows an example of a ladder diagram for a circuit that is used to start and stop a motor using push buttons.
- In the normal state, push button 1 is open and push button 2 closed. When button 1 is pressed, the motor circuit is completed and the motor starts.
- Also, the holding contacts wired in parallel with the motor close and remain closed as long as the motor is running. Thus when the push button 1 is released, the holding contacts maintain the circuit and hence the power to the motor.
- To stop the motor, push button 2 is pressed. This disconnects the power to the motor, and the holding contacts open. Thus when push button 2 is released, there is still no power to the motor.
- Thus we have a motor that is started by pressing button 1 and stopped by pressing button 2.



PLC Ladder Programming

- Writing a program is then equivalent to drawing a switching circuit.
- The ladder diagram consists of two vertical lines representing the power rails.
- Circuits are connected as horizontal lines, that is, the rungs of the ladder, between these two verticals.
- In drawing a ladder diagram, certain conventions are adopted:
- The vertical lines of the diagram represent the power rails between which circuits are connected. The power flow is taken to be from the left-hand vertical across a rung.
- Each rung on the ladder defines one operation in the control process.
- A ladder diagram is read from left to right and from top to bottom. Figure shows the scanning motion employed by the PLC. The top rung is read from left to right. Then the second rung down is read from left to right and so on. When the PLC is in its run mode, it goes through the entire ladder program to the end, the end rung of the program being clearly denoted, and then promptly resumes at the start.





PLC Ladder Programming

- The end rung might be indicated by a block with the word END or RET, for return, since the program promptly returns to its beginning. The scan time depends on the number of runs in the program, taking about 1 ms per 1000 bytes of program and so typically ranging from about 10 ms up to 50 ms.

- Each rung must start with an input or inputs and must end with at least one output. The term input is used for a control action, such as closing the contacts of a switch. The term output is used for a device connected to the output of a PLC, such as a motor. As the program is scanned, the outputs are not updated instantly, but the results stored in memory and all the outputs are updated simultaneously at the end of the program scan.

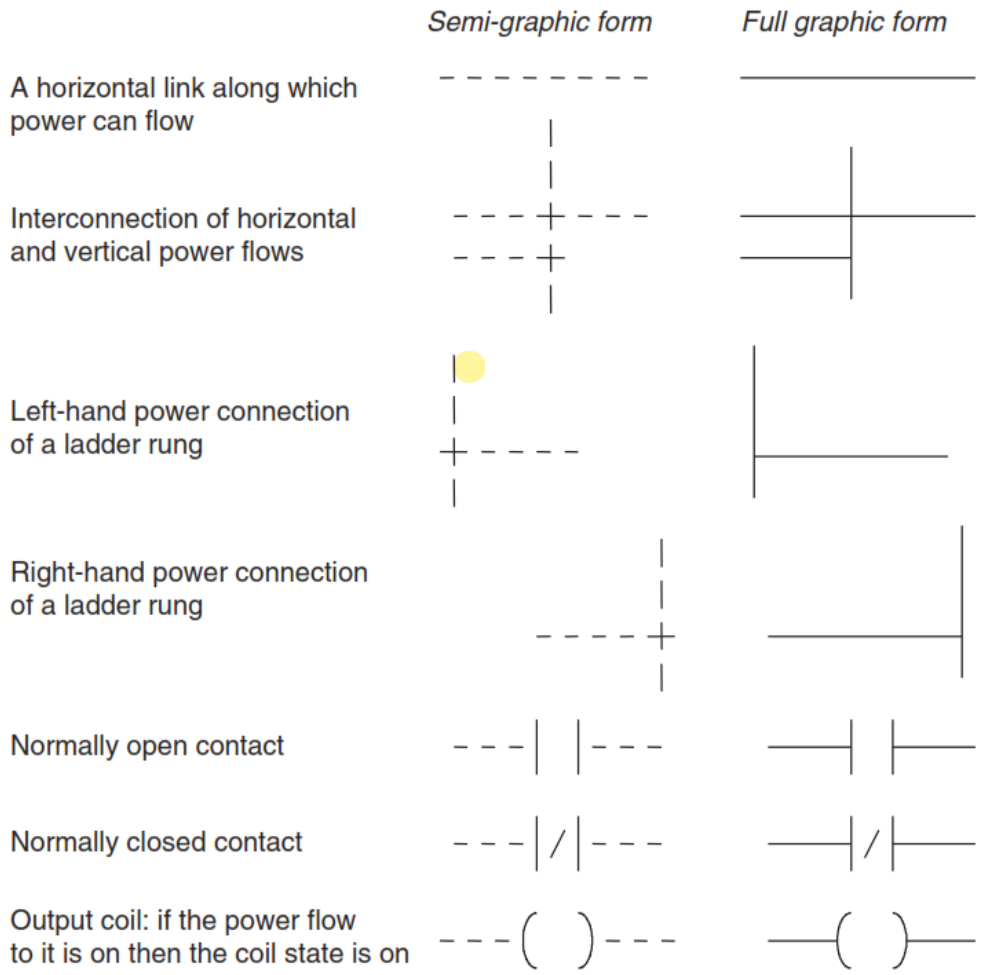
- Electrical devices are shown in their normal condition. Thus a switch that is normally open until some object closes it is shown as open on the ladder diagram. A switch that is normally closed is shown closed.

- A particular device can appear in more than one rung of a ladder. For example, we might have a relay that switches on one or more devices. The same letters and/or numbers are used to label the device in each situation.

- The inputs and outputs are all identified by their addresses; the notation used depends on the PLC manufacturer. This is the address of the input or output in the memory of the PLC.

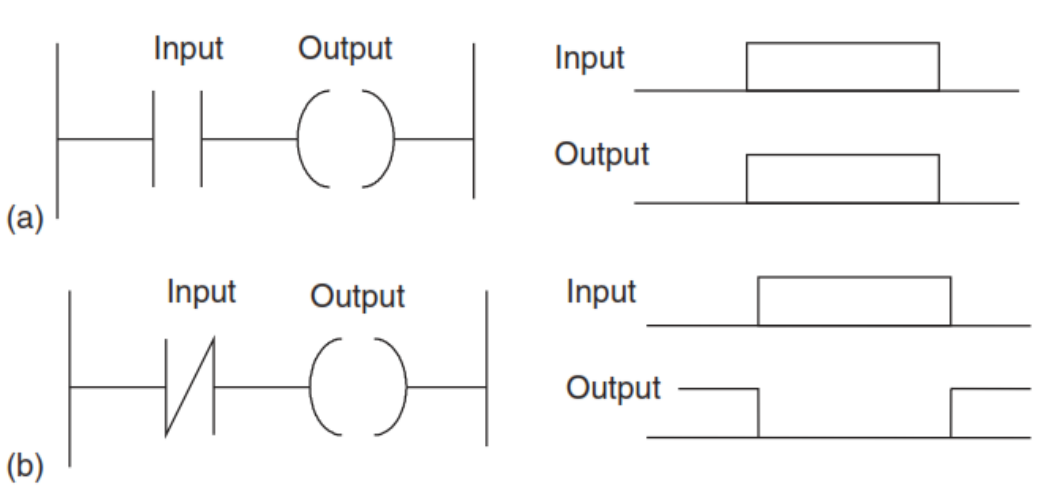
PLC Ladder Programming

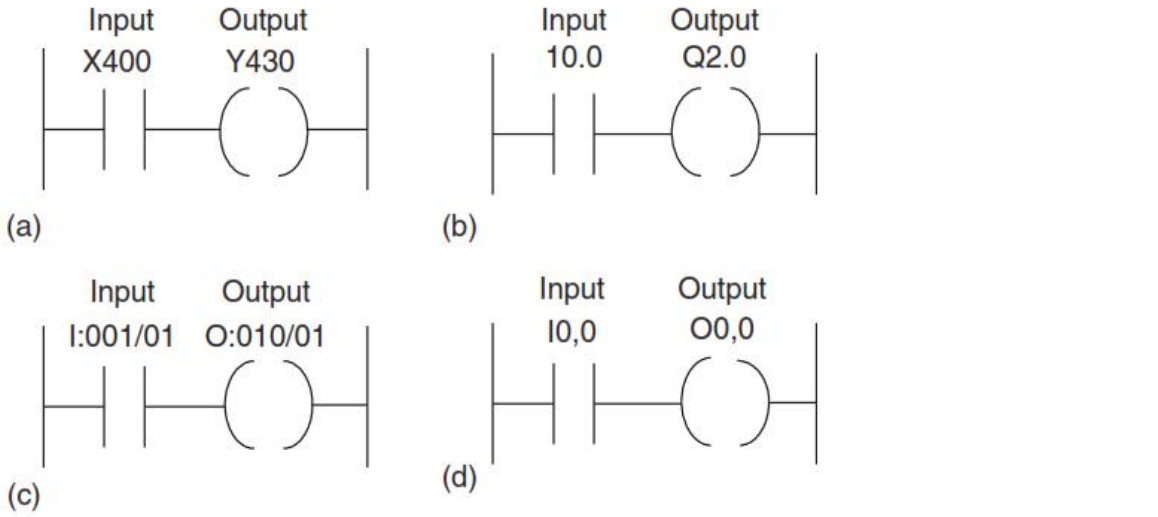
- Figure shows standard IEC 1131-3 symbols that are used for input and output devices. Some slight variations occur between the symbols when used in semigraphic form and when in full graphic, the semigraphic form being the one created by simply typing using the normal keyboard, whereas the graphic form is the result of using drawing tools.
- Inputs are represented by various symbols representing normally open or normally closed contacts. The action of the input is equivalent to opening or closing a switch.
- Output coils are represented by just one form of symbol. (More symbols are introduced.) The name of the associated variable with its address is displayed directly above the symbol (for example, for an input start switch, X400, and for an output Motor 1, Y430).



PLC Ladder Programming

- Figure show gate Bufer and NOT.
- Bufer: When the switch is closed, there is an input, the output of the motor is activated. NO contact.
- NOT: . Only while there was no input to the contacts would there have been an output. NC contact.





Notation: (a) Mitsubishi, (b) Siemens, (c) Allen-Bradley, and (d) Telemecanique.

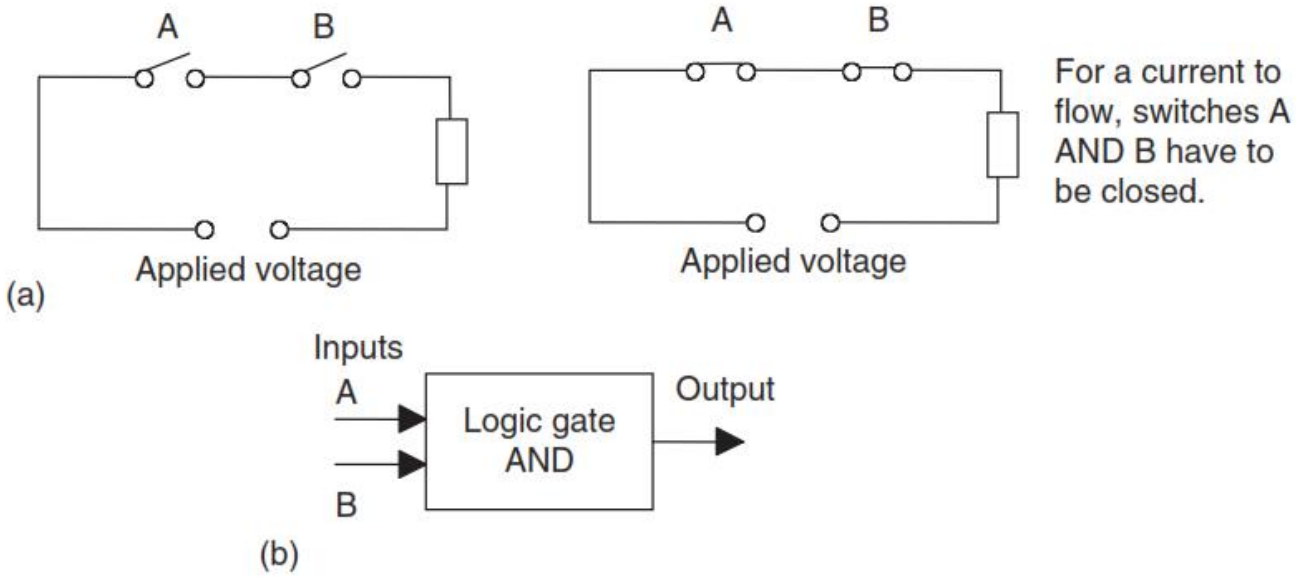
PLC Ladder Programming- Logic Function

- AND gate. Figure a shows a situation in which an output is not energized unless two normally open switches are both closed.

- Switch A and switch B must both be closed, which thus gives an AND logic situation. We can think of this as representing a control system with two inputs, A and B (Figure b).

- Only when A and B are both on is there an output. Thus if we use 1 to indicate an on signal and 0 to represent an off signal, for there to be a 1 output, we must have A and B both 1.

Inputs		Output
A	B	
0	0	0
0	1	0
1	0	0
1	1	1



PLC Ladder Programming- Logic Function

- AND gate. Figure a shows a LAD symbol of gate AND and the time diagram fig. b.
- "On a ladder diagram, contacts in a horizontal rung, that is, contacts in series, represent the logical AND operations."
- The ladder diagram starts with | |, a normally open set of contacts labeled input A, to represent switch A and in series with it | |, another normally open set of contacts labeled input B, to representswitch B. The line then terminates with ( ) to represent the output.

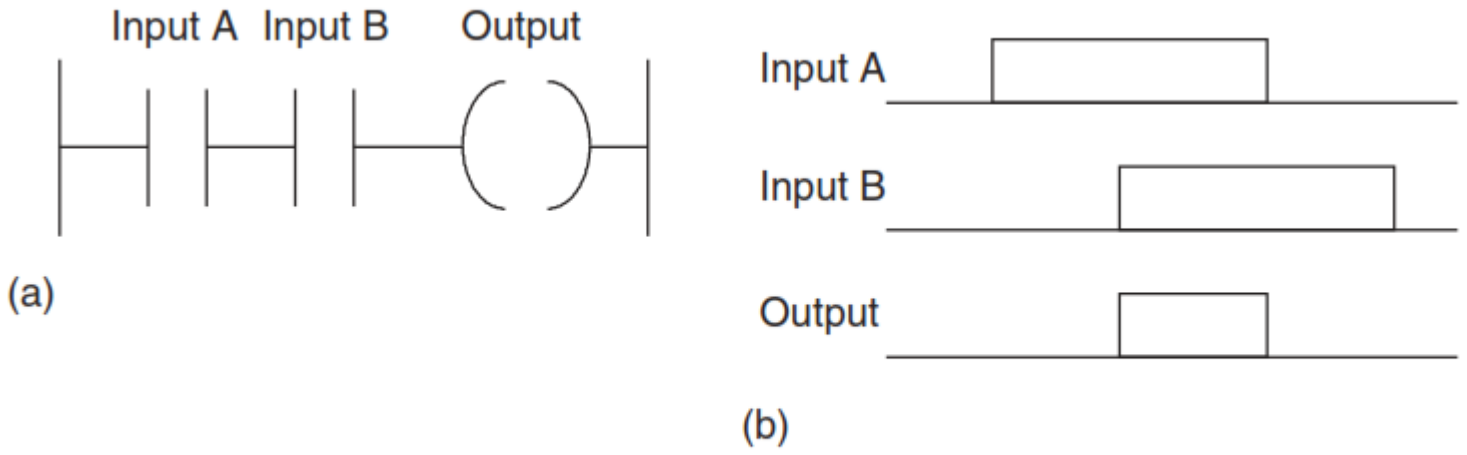


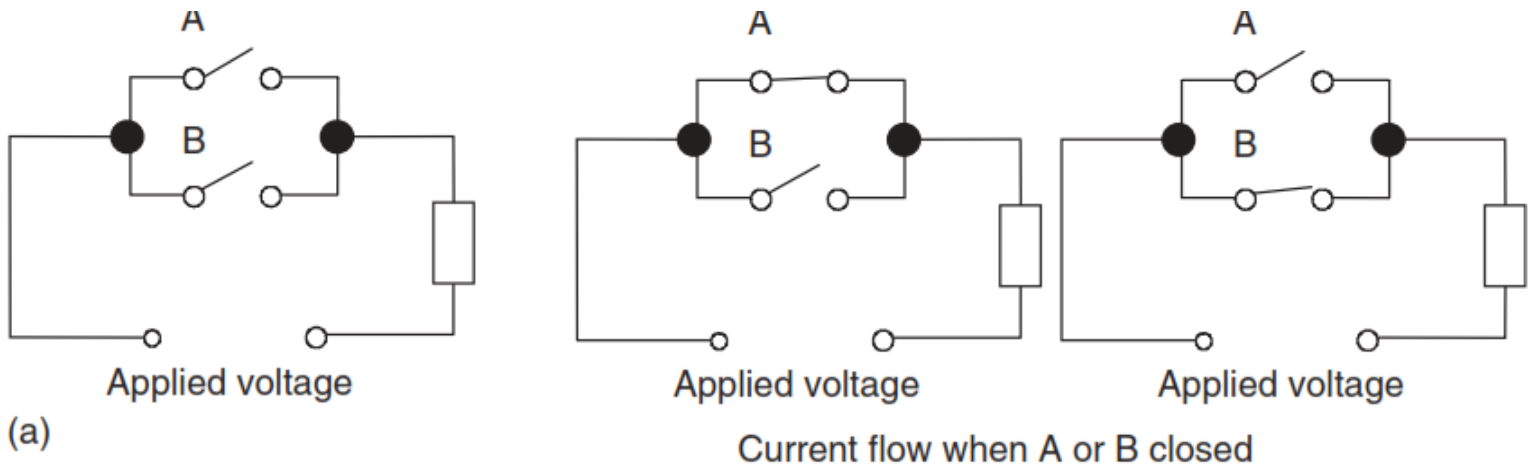
Figure: An AND gate with a ladder diagram rung.

PLC Ladder Programming- Logic Function

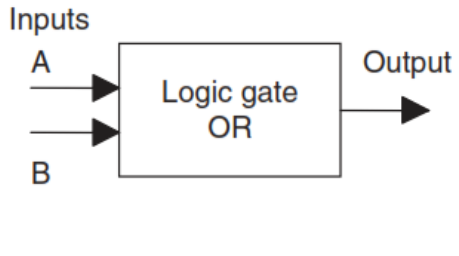
- OR gate. Figure a shows an electrical circuit in which an output is energized when switch A or B, both normally open, are closed.

- This describes an OR logic gate (Figure b) in that input A or input B must be on for there to be an output. The truth table is as follows:

Inputs		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	1

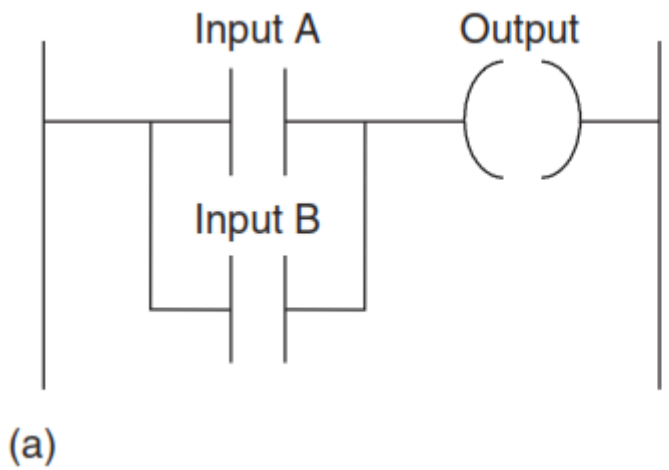


basta che uno sia giu per far arrivare la corrente

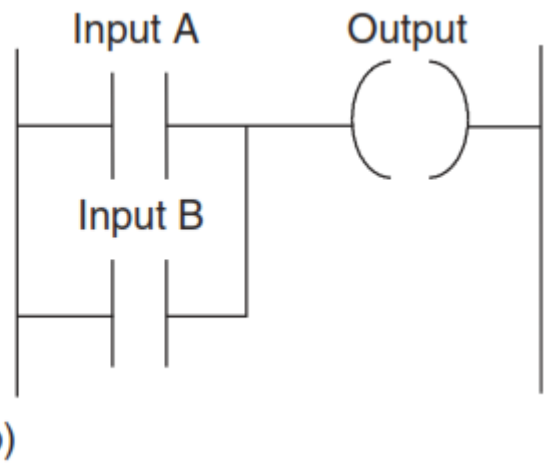


PLC Ladder Programming- Logic Function

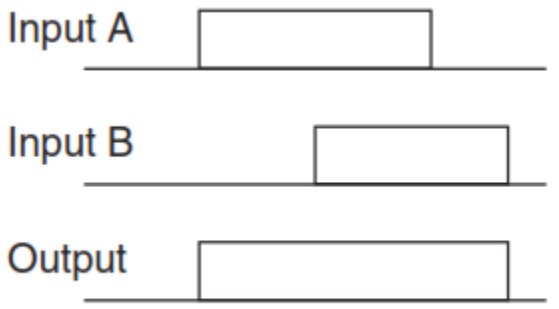
- OR gate. Figure shows an OR logic gate system on a ladder diagram. The ladder diagram starts with | | normally open contacts labeled input A, to represent switch A and in parallel with it | |, normally open contacts labeled input B, to represent switch B.
- Either input A or input B must be closed for the output to be energized (Figure c). The line then terminates with ( ) to represent the output. In general:
- "Alternative paths provided by vertical paths from the main rung of a ladder diagram, that is, paths in parallel, represent logical OR operations."



(a)



(b)

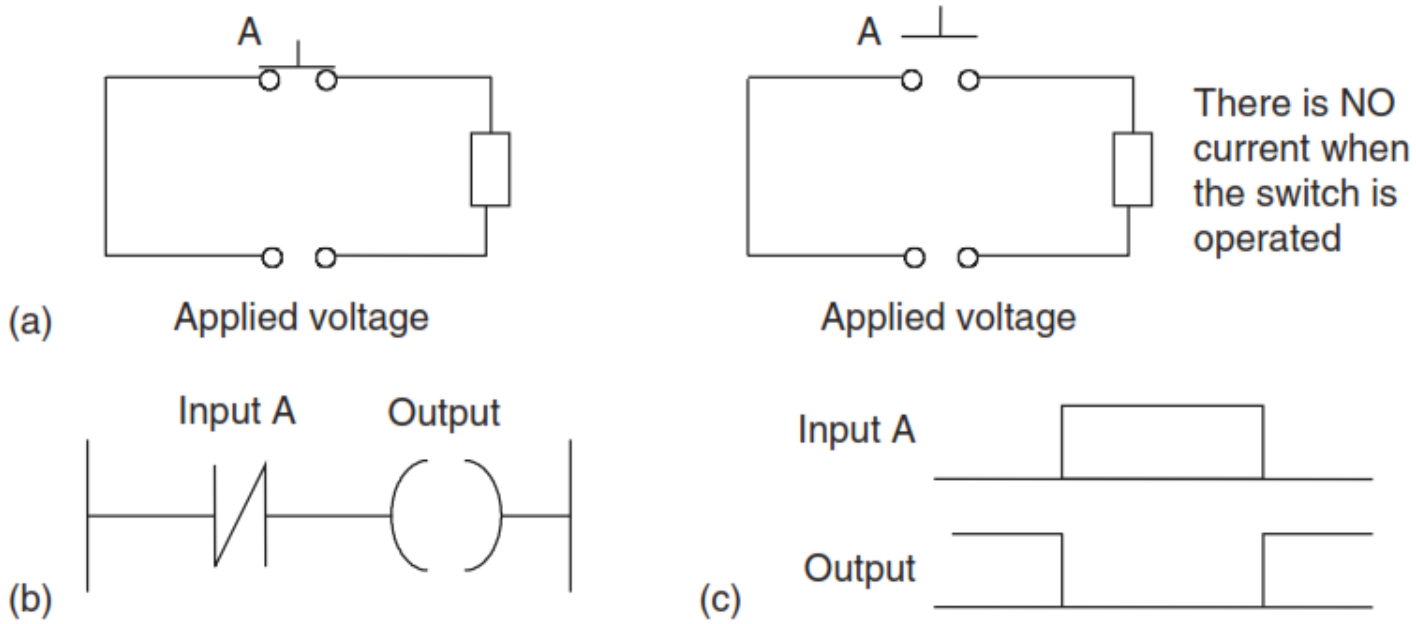


(c)

PLC Ladder Programming- Logic Function

- NOT gate: Figure a shows an electrical circuit controlled by a switch that is normally closed. When there is an input to the switch, it opens and there is then no current in the circuit.
- An example of a NOT gate control system is a light that comes on when it becomes dark, that is, when there is no light input to the light sensor there is an output.
- This example illustrates a NOT gate in that there is an output when there is no input and no output when there is an input (Figure c). The gate is sometimes referred to as an inverter. The truth table is as follows:

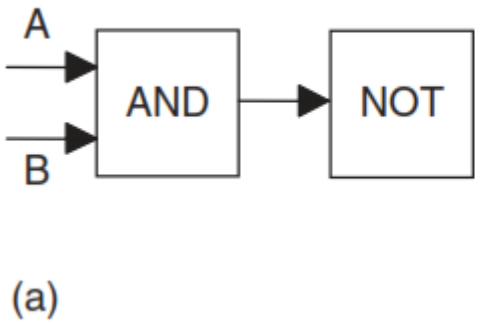
Input	Output
A	
0	1
1	0

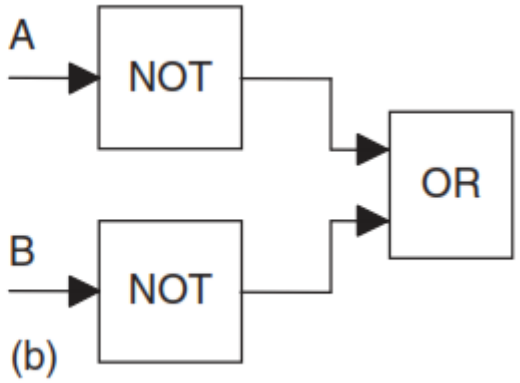


PLC Ladder Programming- Logic Function

- NAND gate: Suppose we follow an AND gate with a NOT gate (Figure a).
- The consequence of having the NOT gate is to invert all the outputs from the AND gate.
- An alternative that gives exactly the same result is to put a NOT gate on each input and then follow that with an OR gate (Figure b). The same truth table occurs, namely:

Inputs		Output
A	B	
0	0	1
0	1	1
1	0	1
1	1	0



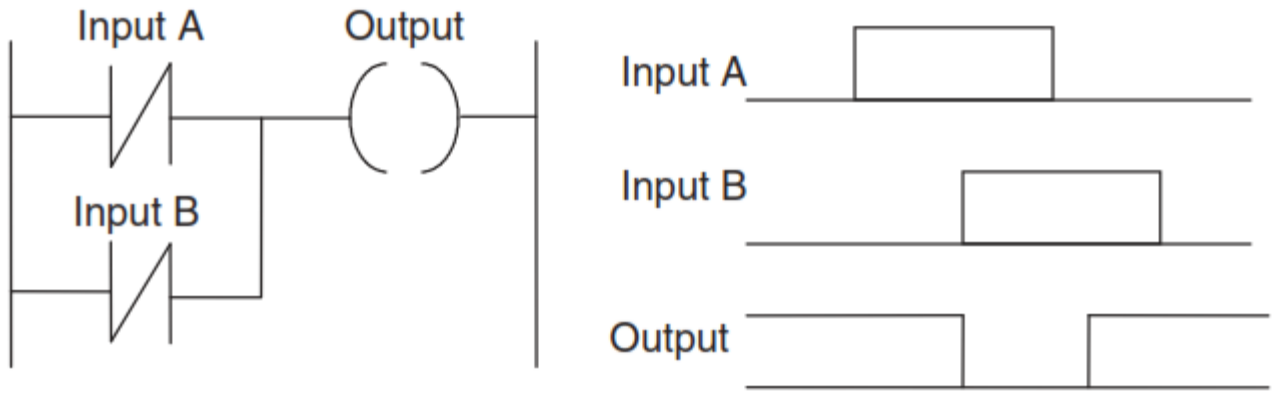


PLC Ladder Programming- Logic Function

- NAND gate: parche un and negato è = a un not A or not B NOT(A and B)= not A or not B

- Figure shows a ladder diagram that gives a NAND gate. When either input A is 0 or input B is 0 (or both are 0), the output is 1. When the inputs to both input A and input B are 1, the output is 0.

- An example of a NAND gate control system is a warning light that comes on if, with a machine tool, the safety guard switch and the limit switch signaling the presence of the workpiece have not been activated.

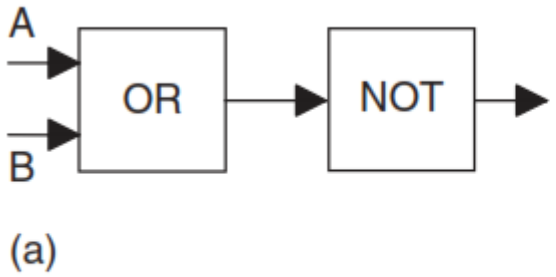


PLC Ladder Programming- Logic Function

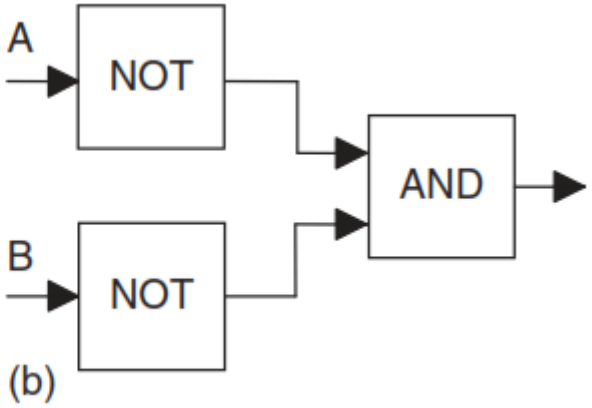
- NOR gate: Suppose we follow an OR gate by a NOT gate (Figure a). The consequence of having the NOT gate is to invert the outputs of the OR gate.

- An alternative, which gives exactly the same results, is to put a NOT gate on each input and then an AND gate for the resulting inverted inputs (Figure b). The following is the resulting truth table:

Inputs		Output
A	B	
0	0	1
0	1	0
1	0	0
1	1	0



(a)



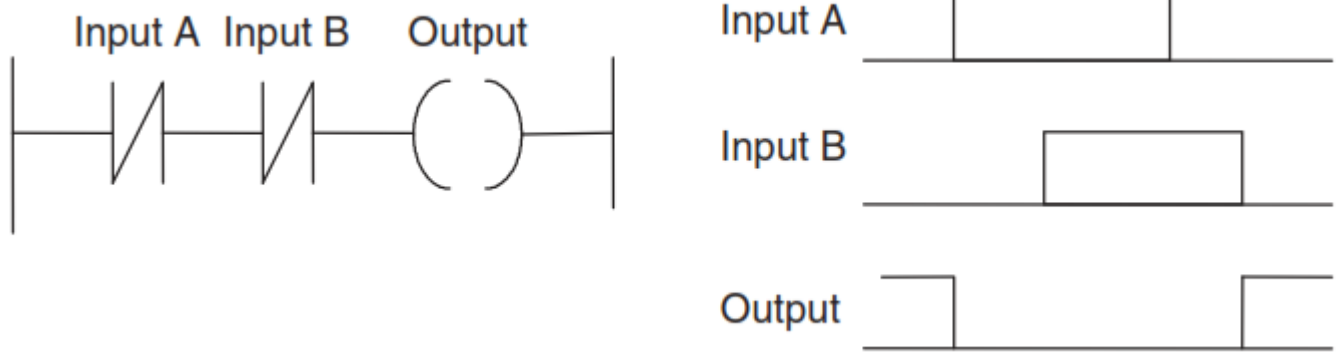
(b)

PLC Ladder Programming- Logic Function

- NOR gate:

NOT(A or B)=not A and not B

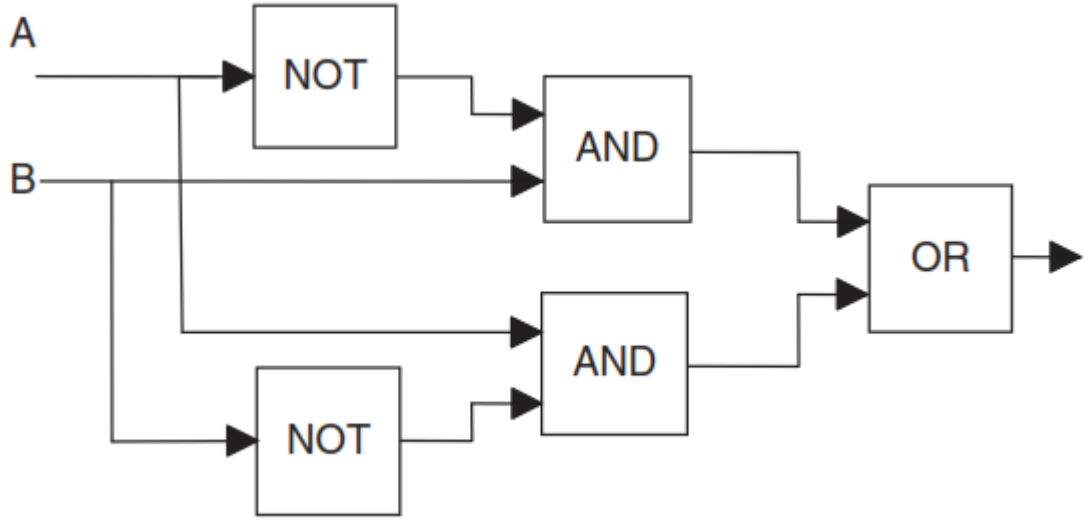
- The combination of OR and NOT gates is termed a NOR gate. There is an output when neither input A nor input B is 1.
- Figure shows a ladder diagram of a NOR system. When input A and input B are both not activated, there is a 1 output. When either input A or input B are 1, there isa 0 output.



PLC Ladder Programming- Logic Function

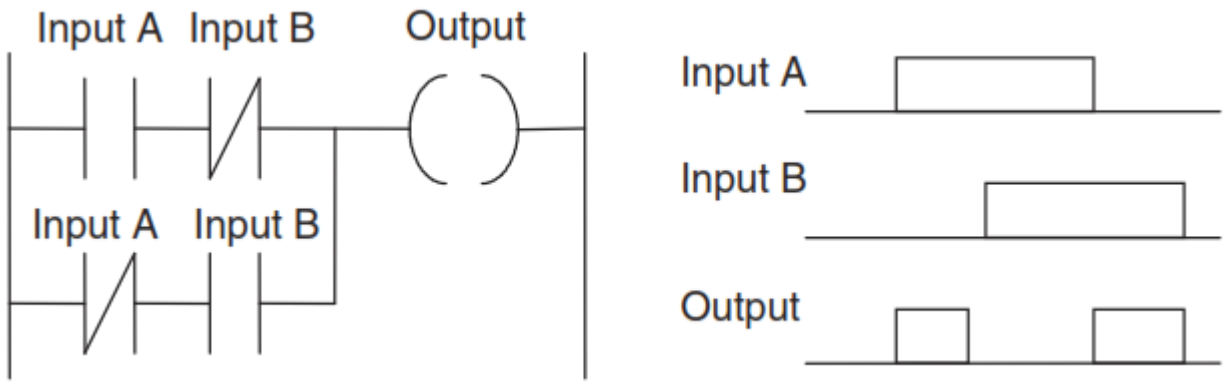
- Exclusive OR (XOR) gate: The OR gate gives an output when either or both of the inputs are 1. However, sometimes there is a need for a gate that gives an output when either of the inputs is 1 but not when both are 1, that is, has the truth table.
- Such a gate is called an EXCLUSIVE OR, or XOR, gate. One way of obtaining such a gate is by using NOT, AND, and OR gates as shown in Figure.

Inputs		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	0



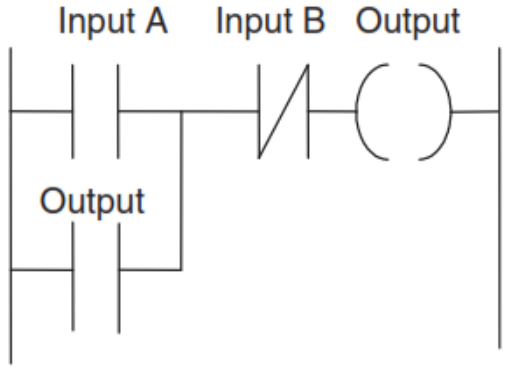
PLC Ladder Programming- Logic Function

- Figure shows a ladder diagram for an XOR gate system.
- When input A and input B are not activated, there is 0 output.
- When just input A is activated, the upper branch results in the output being 1.
- When just input B is activated, the lower branch results in the output being 1.
- When both input A and input B are activated, there is no output.
- In this example of a logic gate, input A and input B have two sets of contacts in the circuits, one set being normally open and the other normally closed.
- With PLC programming, each input may be considered to have as many sets of contacts as necessary.



PLC Ladder Programming- Logic Function

- Latching: There are often situations in which it is necessary to hold an output energized, even when the input ceases.
- A simple example of such a situation is a motor that is started by pressing a push-button switch.
- Though the switch contacts do not remain closed, the motor is required to continue running until a stop push-button switch is pressed.
- The term latch circuit is used for the circuit that carries out such an operation. It is a self-maintaining circuit in that, after being energized, it maintains that state until another input is received.
- *An example of a latch circuit is shown in. When the input A contacts close, there is an output. However, when there is an output, another set of contacts associated with the output closes. These contacts form an OR logic gate system with the input contacts. Thus, even if input A opens, the circuit will still maintain the output energized. The only way to release the output is by operating the normally closed contact B.*

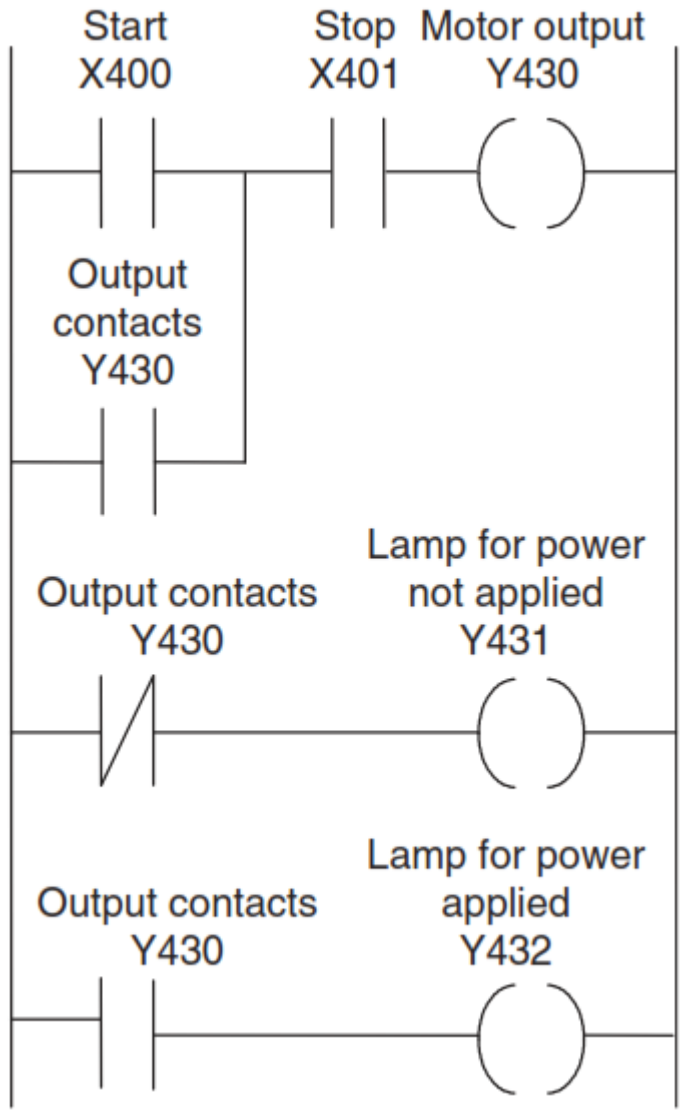


PLC Ladder Programming- Logic Function

- As an illustration of the application of a latching circuit, consider a motor controlled by stop and start push-button switches and for which one signal light must be illuminated when the power is applied to the motor and another when it is not applied.

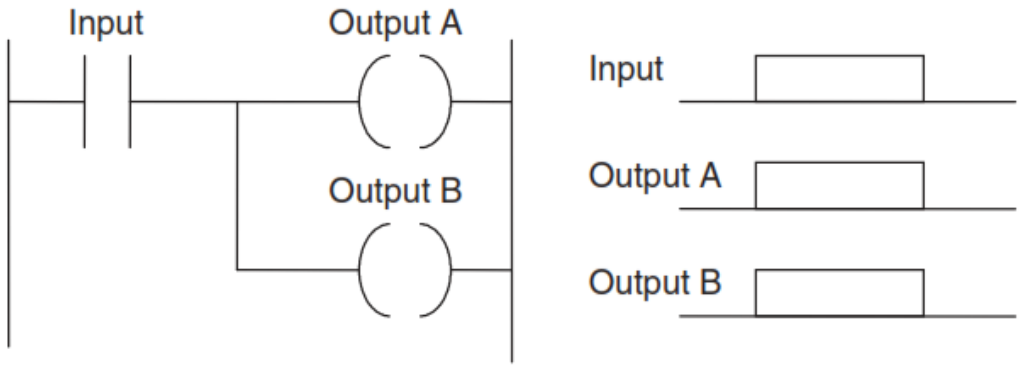
- Figure shows a ladder diagram with Mitsubishi notation for the addresses. X401 is closed when the program is started. When X400 is momentarily closed, Y430 is energized and its contacts close. This results in latching as well as the switching off of Y431 and the switching on of Y432. To switch the motor off, X401 is pressed and opens. Y430 contacts open in the top rung and third rung but close in the second rung. Thus Y431 comes on and Y432 goes off.

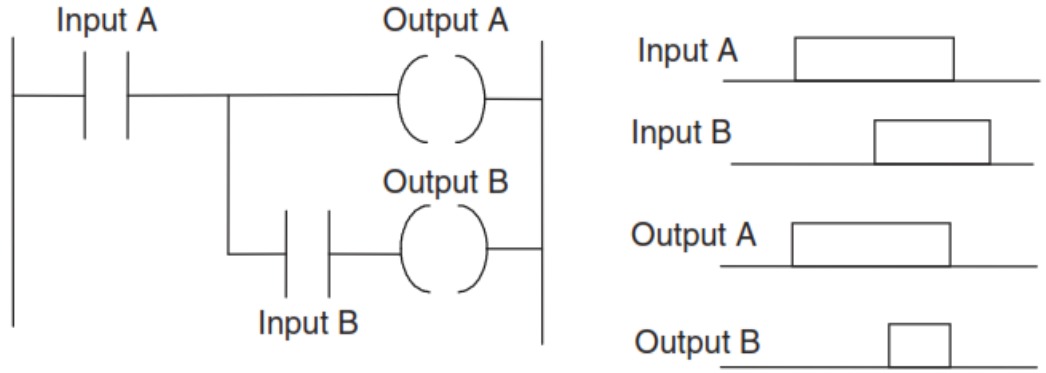
- Latching is widely used with startups so that the initial switching on of an application becomes latched on.



PLC Ladder Programming- Logic Function

- Multiple Outputs: With ladder diagrams, there can be more than one output connected to a contact. Figure a shows a ladder program with two output coils. When the input contacts close, both the coils give outputs.
- For the ladder rung shown in Figure b, output A occurs when input A occurs. Output B occurs only when both input A and input B occur.





PLC Ladder Programming- Logic Function

- Ladder Symbols
- One method of entering the program into the programming terminal involves using a keypad with keys with symbols depicting the various elements of the ladder diagram and keying them in so that the ladder diagram appears on the screen of the programming terminal. The terminal then translates the program drawn on the screen into machine language.
- Computers can be used to draw up ladder programs. This involves loading the computer with the relevant software, such as RSLogix from Rockwell Automation Inc. for Allen-Bradley PLCs, MELSOFT GX Developer for Mitsubishi PLCs, and STEP 7 Micro/WIN V4 for Siemens PLCs. The software operates on the Windows operating system and involves selecting items, in the usual Windows manner, from pull-down menus on the screen.

IL Instruction List Language

- A programming method that can be considered to be the entering of a ladder program using text is the instruction list (IL). An instruction list gives programs as a series of instructions, with each instruction on a new line. Each instruction consists of an operator followed by one or more operands, that is, the subjects of the operator. Thus we might have:

LD A

to indicate that the operand A is to be loaded, LD being the operator used to indicate loading. In terms of ladder diagrams, an operator may be regarded as a ladder element, and LD is equivalent to starting a rung with open contacts for input A. Another instruction might be:

OUT Q

to indicate that there is to be an output to Q.

- Mnemonic codes are used for operators, each code corresponding to an operator/ladder element. The codes used differ to some extent from manufacturer to manufacturer, though a standard under IEC 1131-3 has been proposed and is being widely adopted.

IL Instruction List Language

Table 6.1: Instruction Code Mnemonics

IEC 1131-3	Mitsubishi	OMRON	Siemens	Operation	Ladder Diagram
LD	LD	LD	A	Load operand into result register.	Start a rung with open contacts.
LDN	LDI	LD NOT	AN	Load negative operand into result register.	Start a rung with closed contacts.
AND	AND	AND	A	Boolean AND.	Series element with open contacts.
ANDN	ANI	AND NOT	AN	Boolean AND with negative operand.	Series element with closed contacts.
OR	OR	OR	O	Boolean OR.	Parallel element with open contacts.
ORN	ORI	OR NOT	ON	Boolean OR with negative operand.	Parallel element with closed contacts.
ST	OUT	OUT	=	Store result register into operand.	An output.

IL Instruction List Language

- Instruction List is a low-level textual language that is simple to implement and used by a number of PLC manufacturers, mainly for small and medium-sized PLCs. It is particularly suitable for small, straightforward programs.
- Some manufactures do not support ILs but use only higher-level language of structured text (ST).
- As an illustration of the use of IEC 1131-3 operators, consider the following:

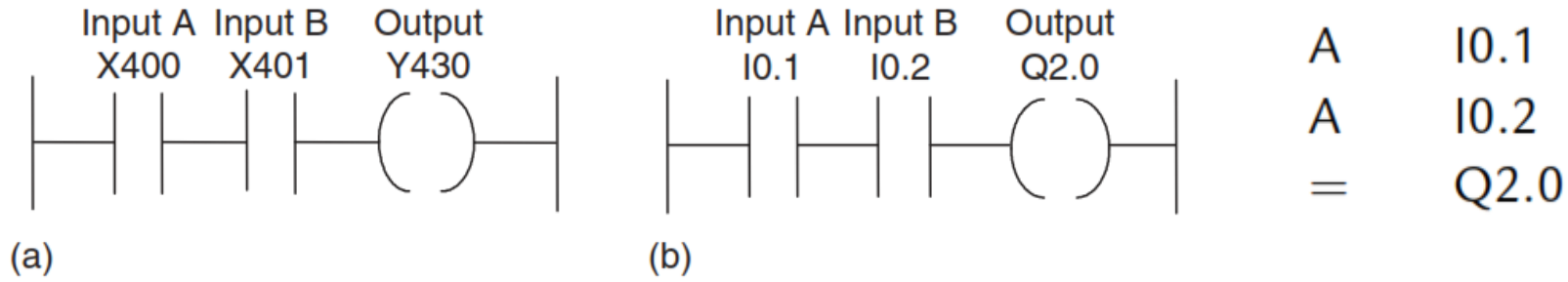
```
LD      A     (*Load A*)
AND     B     (*AND B*)
ST      Q     (*Store result in Q, i.e. output  to Q*)
```

- LD is the operator, A the operand, and the words at the ends of program lines and in parentheses shown and preceded and followed by * are comments added to explain what the operation is and are not part of the program operation instructions to the PLC.
- LD A is thus the instruction to load A into the memory register.
- The next line of the program has the Boolean operation AND performed with A and B.
- The last line has the result stored in Q, that is, output to Q.

PLC Logic Function – LAD, IL

- When looked at in terms of ladder diagrams, whenever a rung is started, it must use a “start a rung” code.
- This might be LD, or perhaps A or L, to indicate that the rung is starting with open contacts, or LDI, or perhaps LDN, LD NOT, AN, or LN, to indicate it is starting with closed contacts.
- All rungs must end with an output or store result code.
- **Gate AND**

```
LD      X400
AND     X401
OUT     Y430
```



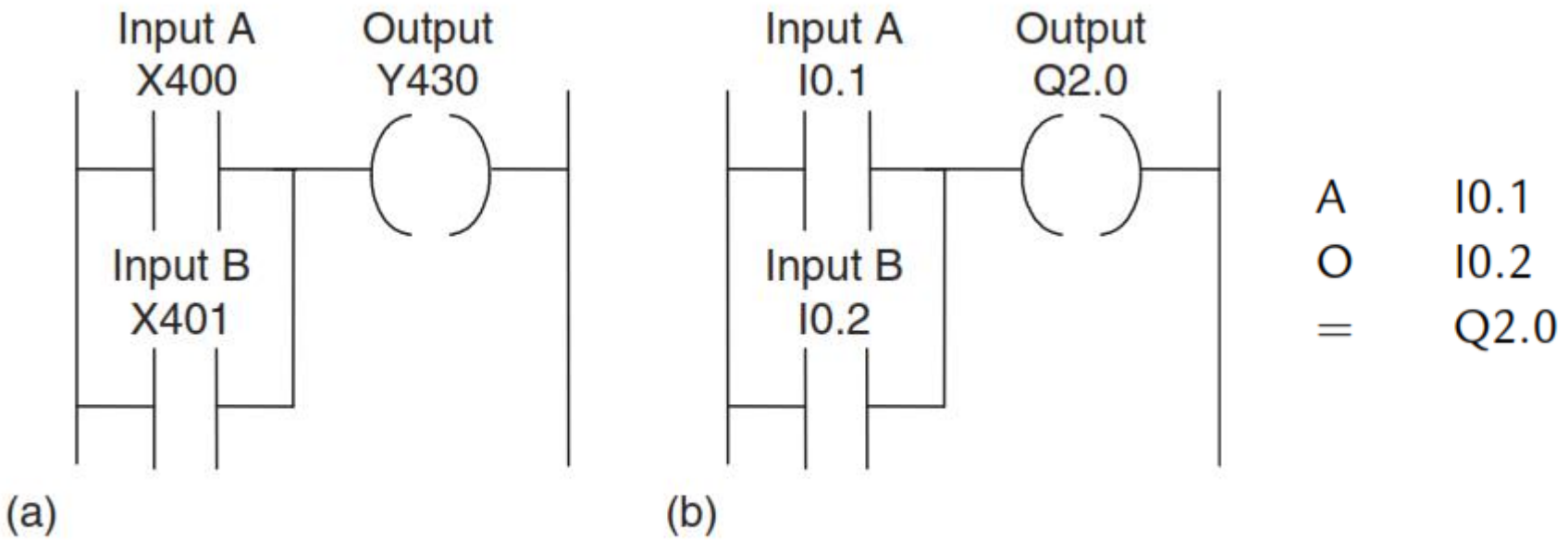

```
A     I0.1
A     I0.2
=     Q2.0
```

Figure 6.1: AND gate: (a) Mitsubishi, and (b) Siemens.

PLC Logic Function – LAD, IL

- **Gate OR**

```
LD      X400
OR      X401
OUT     Y430
```



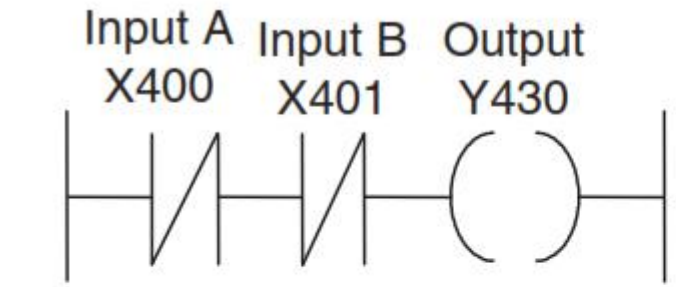

```
A       I0.1
O       I0.2
=       Q2.0
```

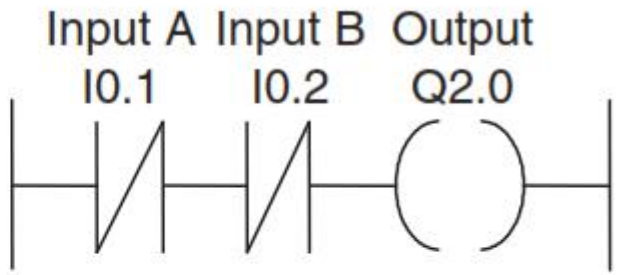
Figure 6.2: OR gate: (a) Mitsubishi, and (b) Siemens.

PLC Logic Function – LAD, IL

- **Gate NOR**

```
LDI     X400
ANI     X401
OUT     Y430
```






```
LN      I0.1
AN      I0.2
=       Q2.0
```

Figure 6.3: NOR gate: (a) Mitsubishi, and (b) Siemens.

PLC Logic Function – LAD, IL

- **Gate NAND**

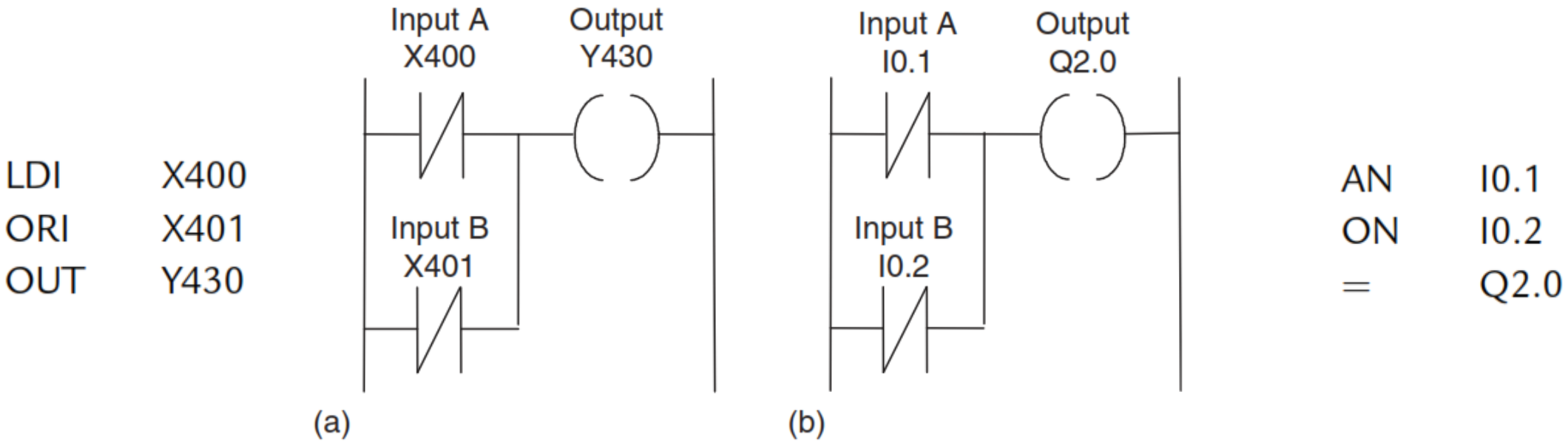


Figure 6.4: NAND gate: (a) Mitsubishi, and (b) Siemens.

PLC Logic Function – LAD, IL

- **Branch Code**

The EXCLUSIVE OR (XOR) gate shown in Figure 6.5 has two parallel arms with an AND situation in each arm.

Figure 6.5a shows Mitsubishi notation. With such a situation, Mitsubishi uses an ORB instruction to indicate "OR together parallel branches." The first instruction is for a normally open pair of contacts X400. The next instruction is for a series set of normally closed contacts X401, hence ANI X401. After reading the first two instructions, the third instruction starts a new line. It is recognized as a new line because it starts with LDI, all new lines starting with LD or LDI. But the first line has not been ended by an output.

The PLC thus recognizes that a parallel line is involved for the second line and reads together the listed elements until the ORB instruction is reached.

```
LD      X400
ANI     X401
LDI     X400
AND     X401
ORB
OUT     Y430
```

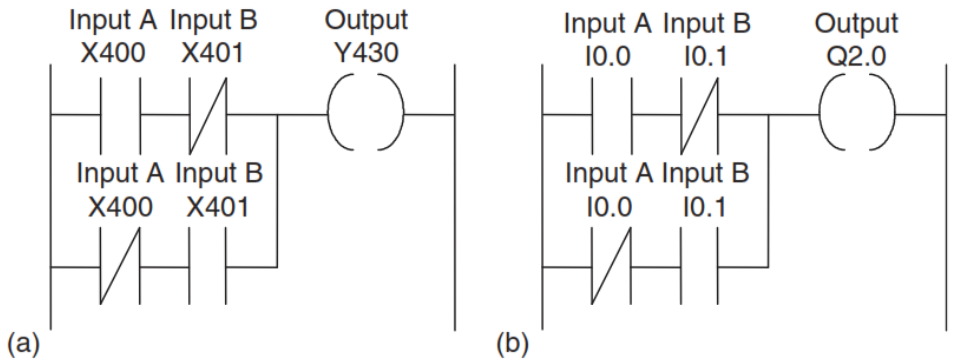


Figure 6.5: XOR gate: (a) Mitsubishi, and (b) Siemens.

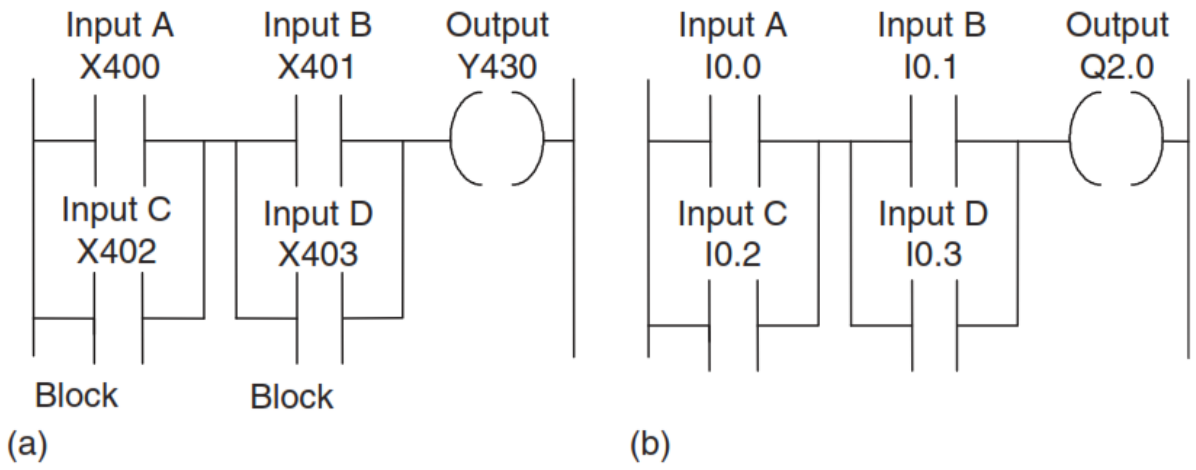
Step	Instruction	
0	A(	
1	A	I0.0
2	AN	I0.1
3	)	
4	O(	
5	AN	I0.0
6	A	I0.1
7	)	
8	=	Q2.0

PLC Logic Function – LAD, IL

- **Branch Code**

Figure shows a circuit that can be considered as two branched AND blocks. Figure 6.6a shows the circuit in Mitsubishi and Siemens notation. The instruction used here is ANB.

Step	Instruction	
0	LD	X400
1	OR	X402
2	LD	X401
3	OR	X403
4	ANB	
5	OUT	Y430



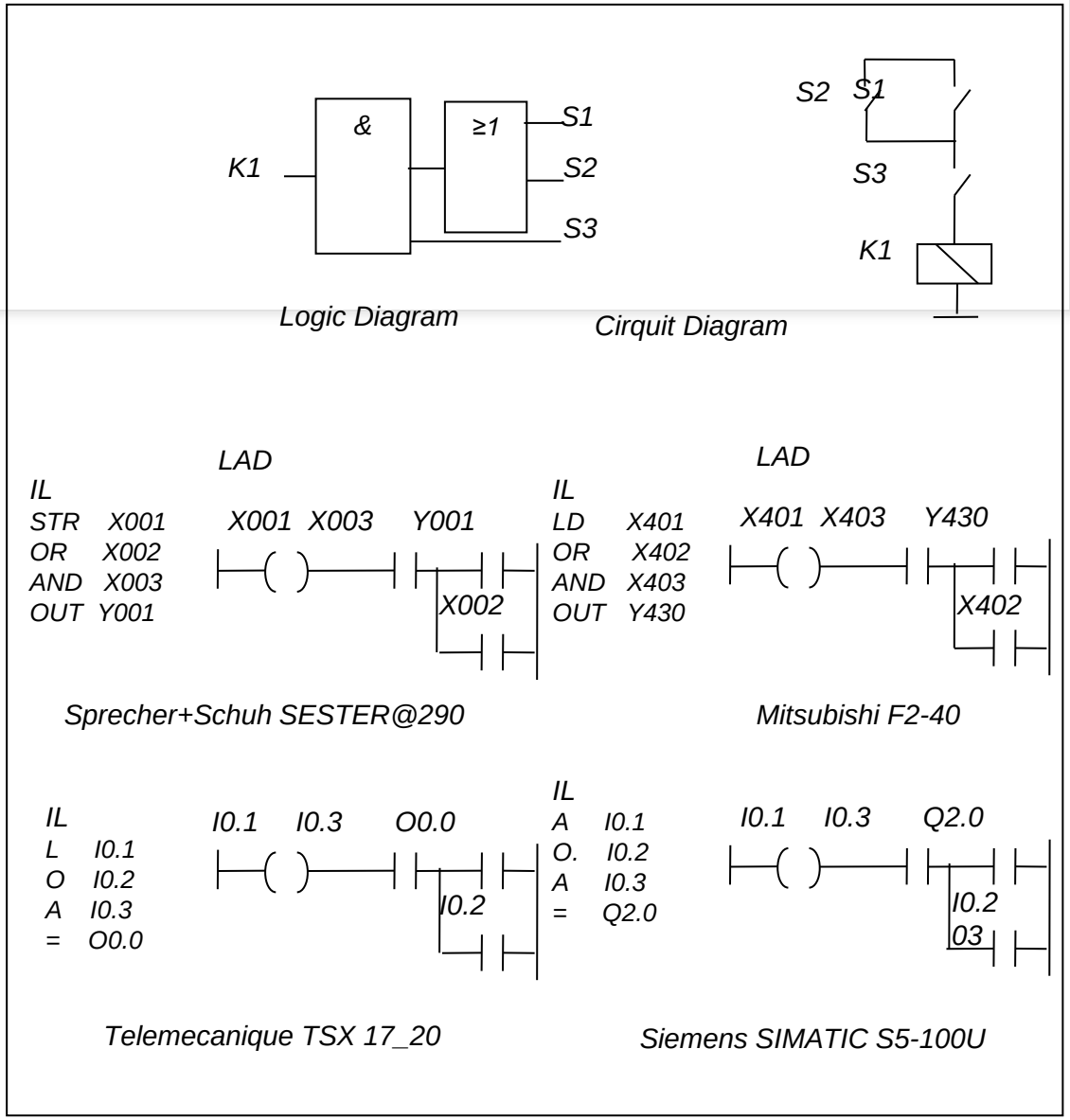
Step	Instruction	
0	A(	
1	A	I0.0
2	O	I0.2
3	)	
4	A(	
5	A	I0.1
6	O	I0.3
7	)	
8	=	Q2.0

PLC Logic Function – LAD, IL

- **OR-AND funnction**

Figure shows a circuit that can be considered with or priority

We read all dhe OR and after we read dhe AND port.

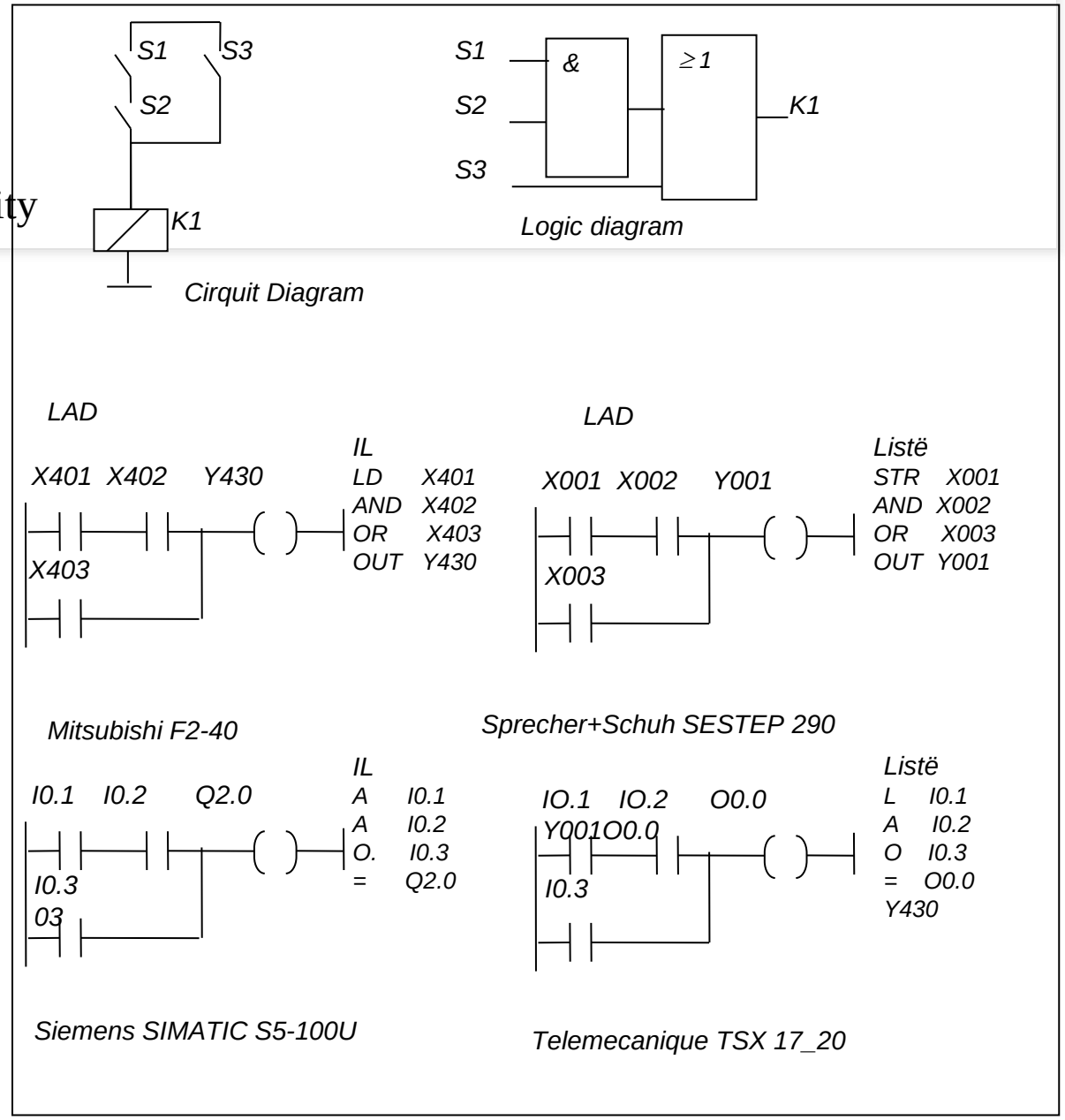


PLC Logic Function – LAD, IL

- **AND – OR funnction**

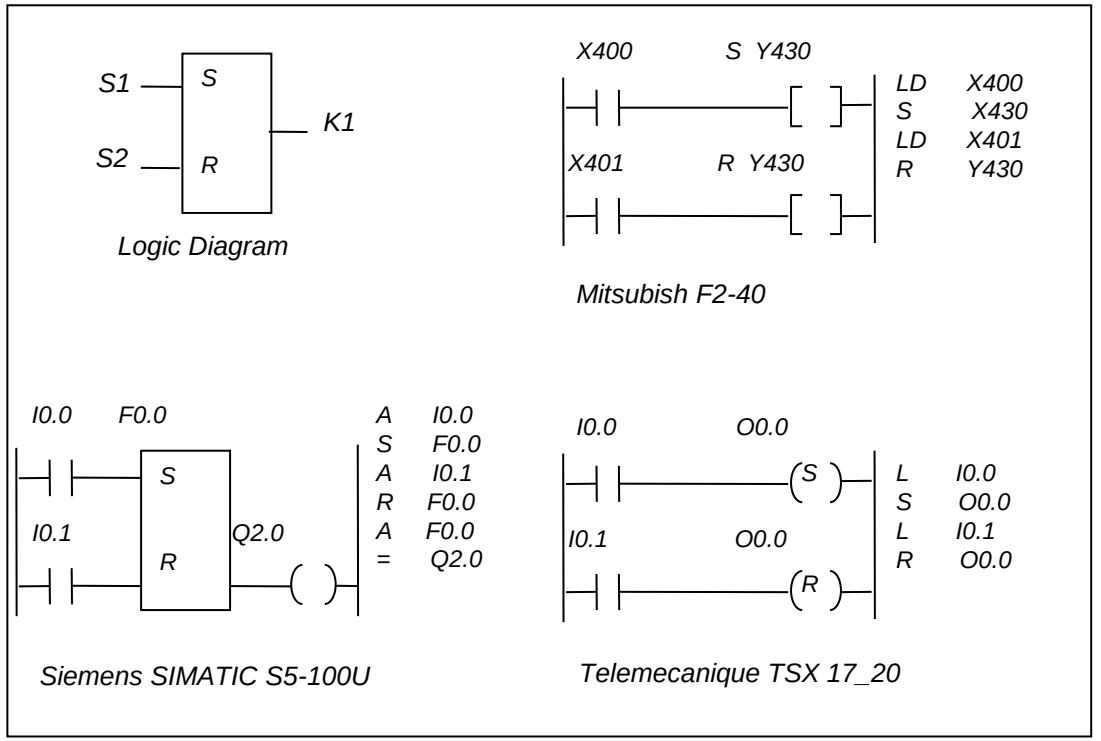
Figure shows a circuit that can be considered with or priority

We read all dhe AND and after we read dhe OR port.



PLC Logic Function – LAD, IL

- **FlipFlop**
- **A** flip-flop represents a two-step operation or function that is applied from an output or an internal relay. Actions (ON and OFF) set and reset work independently and both have a locking effect.
- The action of locking or enabling is known as the term set and unlocking as reset.
- We generally use the flip-flop when we need a short (short-duration) Input signal to produce a longer-duration output.



PLC Logic Function – LAD, IL

- **One-shot**
- A "one-shot" is a programming function that enables us to detect a change in the state of the signal.
- Figure. shows the logic symbol and its operation, which resembles that of an electromechanical relay.
- Whenever S1 goes from OFF to ON, when it turns on briefly (program review time, which is typically <0.1S). to reset K1, S1 must be turned OFF and ON again.
- A "one-shot" is a very useful function and is used, for example, to:
- create pulses for restarting counters and timers for starting cycles,
- artificially short signals of a long duration,
- to mark the moment of change of state of a signal.