eSkill AP
Learn Anytime Anywhere

Andhra Pradesh State Skill Development Corporation

Industrial Automation With PLC

Ladder Logic Explanation

Ladder Logic Basics

Ladder logic is a programming language that is used to program a PLC (Programmable Logic Controller). It is a graphical PLC programming language that expresses logic operations with symbolic notation using ladder diagrams, much like the rails and rungs of a traditional relay logic circuit. It is used by engineers and electricians to execute logical, sequential, counting, timing, and arithmetic tasks in order to carry industrial automation applications.

In the good old days, machine and process automation was accomplished using a hard-wired control system known as relay logic. With the advent of microprocessors and the invention of the PLC, relay logic quickly became superseded by programming languages such as ladder logic. Because ladder logic was originally designed to replace the use of hard-wired relay logic circuits for machine control the ladder logic programming code actually looks like an electrical schematic drawing.

You might be thinking that ladder logic sounds like an old programming language. So, with the advancements in software that the world has recently made, is ladder logic still used? Ladder logic is definitely still being used in PLC programming. It is the most common method for programming a PLC. Ladder logic programming is still used today because the core fundamental logic principles for machine and process control are still the same.

Having said that, the advancements in software have resulted in other PLC programming languages that compliment ladder logic, and vastly improved interface and programming software has also been developed.

In PLC programming, ladder logic is a programming language that is used for developing logic expressions in order to automate tasks. Recent advancements in software technology mean that PLC programming using ladder logic has been extended into counting, timing, arithmetic, sequencers, PID control, data manipulation functions, and more. Over the years ladder logic has developed into the powerful PLC programming language that it is today.

Ladder logic is used extensively for programming PLCs in a multitude of industrial automation applications. Some examples include….

- Material Handling Conveyor System.
- Pallet Packing and Strapping.
- Ball Mill Lubrication System.
- Logistics Package Conveying and Sorting.
- Cement Batching.
- Beverage Bottling and Labelling.
- Hopper and Tank Level Control.
- Air and Liquid Flow and Pressure Control.

Trying to learn ladder logic basics can be an overwhelming task. If you have some experience with electric circuits then the PLC programming concepts of ladder logic should be relatively easy to grasp. Otherwise, take comfort in knowing that ladder logic is the quickest and easiest PLC programming language to learn. In order to help you grasp ladder logic basics, we will….

- Examine the seven basic parts of a ladder diagram.
- Identify the binary and logic concepts used in ladder logic.
- Reveal the hidden ladder logic functions that are automatically built into the structure of the ladder diagram.
- Discover the five fundamental logic functions that are essential to know.

So let's begin….

What is a Ladder Diagram?

A **ladder diagram** is a type of schematic diagram used in industrial automation that represents logic control circuits. Ladder diagrams are composed of two vertical power rails and horizontal logic rungs to form what looks like a ladder. The control logic in a ladder diagram is contained within the rungs.

There are two differences between an electrical schematic and a ladder diagram. The first difference is the control logic in an electrical schematic is represented using components whereas in a ladder diagram symbols are used. The second difference is the control logic execution in an electrical schematic is as per the operation of an electrical circuit whereas in a ladder diagram it relies on the methodical nature of the PLC scan.

Why is a ladder diagram used for PLC programming?

The reason why ladder diagrams are used for PLC programming is that the early control system designers were accustomed to relaying logic control circuits and ladder diagrams closely mimic these. They preferred to use ladder diagrams instead of using text-based programming languages of the day like C, BASIC, Pascal, and FORTRON. The other reason ladder diagrams are used is because factory maintenance staff already understand how to read relay control circuits so using ladder diagrams for programming a PLC meant they were easily able to troubleshoot control system problems.

Ladder diagrams help you to formulate the logic expressions in a graphical form that are required to program a PLC. They represent conditional, input, and output expressions as symbols. So writing a PLC program using ladder diagrams is similar to drawing a relay control circuit.

Ladder diagram (LD) is the official name given in the international PLC programming standard IEC-61131. But, These days the terms ladder diagram, ladder logic diagram, ladder drawing, ladder control, ladder circuit, control logic diagram, and logic diagram (to name a few) are all used to describe relay logic circuits and ladder logic programming.

So don't get too caught up in the specific definition of each of these expressions, they kind of generally all mean the same thing. At the end of the day, most people will know what you are talking about anyway. Personally, I use the term **ladder logic** for PLC programming **and relay logic** for relay control circuits.

How to Draw Ladder Logic Diagrams

The simple way to describe a ladder diagram is a graphic programming language that uses a series of rails and rungs containing logic symbols that are combined to form decision making expressions. Ladder diagrams actually look like a ladder and are more commonly known as ladder logic programming.

The rails in a ladder diagram represent the supply wires of a relay logic control circuit. There is a positive voltage supply rail on the left-hand side and a zero voltage rail on the right-hand side. In a ladder diagram, the logic flow is from the left-hand rail to the right-hand rail.

The rungs in a ladder diagram represent the wires that connect the components of a relay control circuit. In a ladder diagram symbols are used to represent the relay components. The symbols are placed in the rung to form a network of logic expressions.

When implementing a ladder logic program in a PLC there are **seven basic parts of a ladder diagram** that critical to know. They are rails, rungs, inputs, outputs, logic expressions, address notation/tag names and comments. Some of these elements are essential and others are optional.

To help understand **how to draw ladder logic diagrams** the seven basic parts of a ladder diagram are detailed below.

1. **Rails** – There are two rails in a ladder diagram which are drawn as vertical lines running down the far most ends of the page. If they were in a relay logic circuit they would represent the active and zero volt connections of the power supply where the power flow goes from the left-hand side to the right-hand side.
2. **Rungs** – The rungs are drawn as horizontal lines and connect the rails to the logic expressions. If they were in a relay logic circuit they would represent the wires that connect the power supply to the switching and relay components.
3. **Inputs** – The inputs are external control actions such as a push-button being pressed or a limit switch is triggered. The inputs are actually hardwired to the PLC terminals and represented in the ladder diagram by a normally open (NO) or normally closed (NC) contact symbol.
4. **Outputs** – The outputs are external devices that being are turned on and off such as an electric motor or a solenoid valve. The outputs are also hardwired to the PLC terminals and are represented in the ladder diagram by a relay coil symbol.
5. **Logic Expressions** – The logic expressions are used in combination with the inputs and outputs to formulate the desired control operations.
6. Address Notation & Tag Names – The address notation describes the input, output, and logic expression memory addressing structure of the PLC. The tag names are the descriptions allocated to the addresses.
7. **Comments** – Last but not least, the comments are an extremely important part of a ladder diagram. Comments are displayed at the start of each rung and are used to describe the logical expressions and control operations that the rung, or groups of

rungs, are executing. Understanding ladder diagrams is made a lot easier by using comments.

Parts of a Ladder Logic Diagram
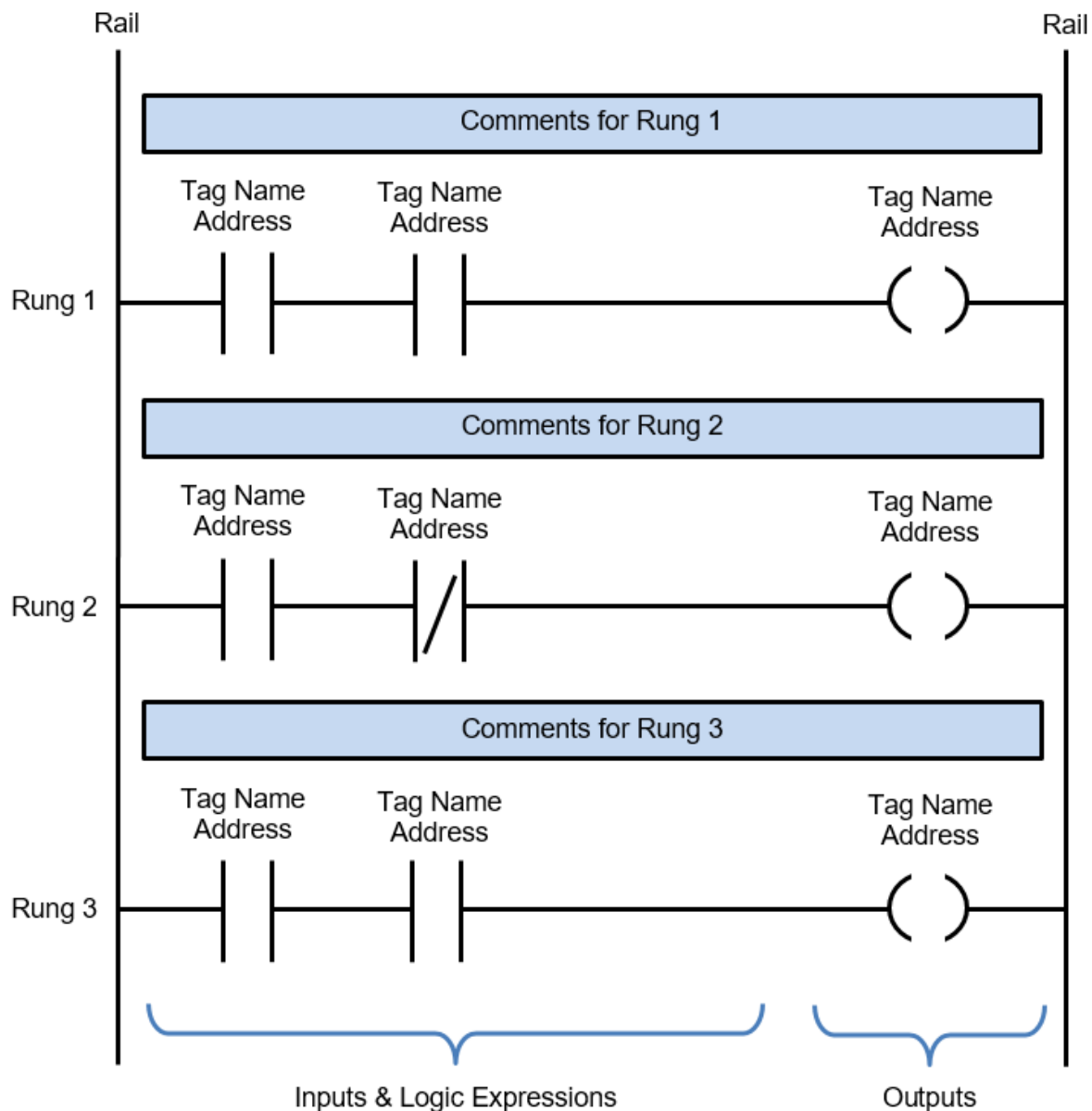
How to Read Ladder Logic

Microprocessors like the ones found in PLC's and personal computers operate on the binary concept.

You've probably heard of the term 'binary'. It refers to the principle that things can be thought of in one of two states. The states can be defined as :
1 or 0

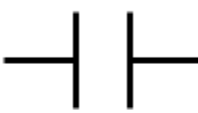True or False

On or Off

High or Low

Yes or No

Microprocessors love

Binary…..10101011101000111010001010100010100100101010010011.

I don't know about you, but my head hurts just looking at that.

Luckily ladder logic uses symbolic expressions and a graphical editor for writing and reading ladder diagrams making it easier for us mere humans to comprehend. If we translate a real-world event into ladder logic we can express it symbolically in the form of a normally open (NO) contact. This event could be something like a button being pushed or a limit switch being activated.

Let's call it event 'A'. It follows the binary concept and has one of two states, TRUE or FALSE.

The event associated with the normally open (NO) contact can be TRUE or FALSE. When that event is TRUE then it is highlighted green and the logic flow can move past it to the next logic expression. Just like the current flow in an electric circuit when a switch is turned on.
The ladder logic truth table for a normally open (NO) contact which denotes event 'A' is shown below….

Ladder Logic Truth Table – Normally Open (NO) Contact

A normally open (NO) contact alone cannot decide what action to take to automate something, it merely tells us in what state the event is.

We need binary's best friend 'logic' to help out.

Logic is the ability to decide what action needs to be taken depending on the state of one or more events.

We use binary and logic concepts every day in our own lives. For example, if I feel cold then I put my sweater on, but if I feel hot then I take my sweater off.

Binary concept – Cold or Hot, Sweater On or Sweater Off.
Logic concept – IF, THEN logic functions.
Binary logic in action!

The binary and logic concepts are what make ladder logic work. The hidden key to unlock your understanding of how ladder logic works are that the IF, THEN logic functions are automatically built into the structure of the ladder diagram.

Let me show you.

Ladder Logic Functions

Let's take a real-world event, allocate it to a normally closed (NC) contact and call it 'A'. In ladder logic, the real world events are defined as PLC inputs.

Now, let's call the result of the logic function 'Y'. In ladder logic, the result of a rung logic function is defined as a PLC output.

When we take these two fundamental elements and insert them into a rung in a ladder diagram we get your first line of code!

It's equivalent to "Hello World" in other programming languages.
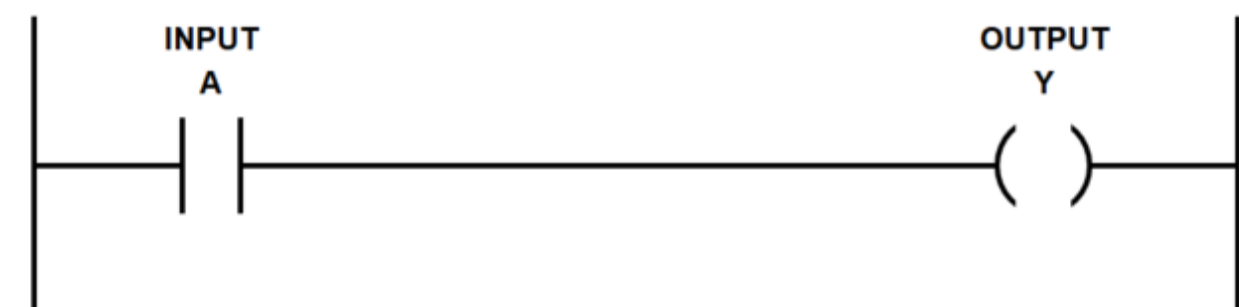
Ladder Logic Basics – Hello World

Now, let's expose the hidden inbuilt functions by highlighting them in blue in order to illustrate the relationship between the ladder diagram rung structure and its inbuilt IF, THEN functions.
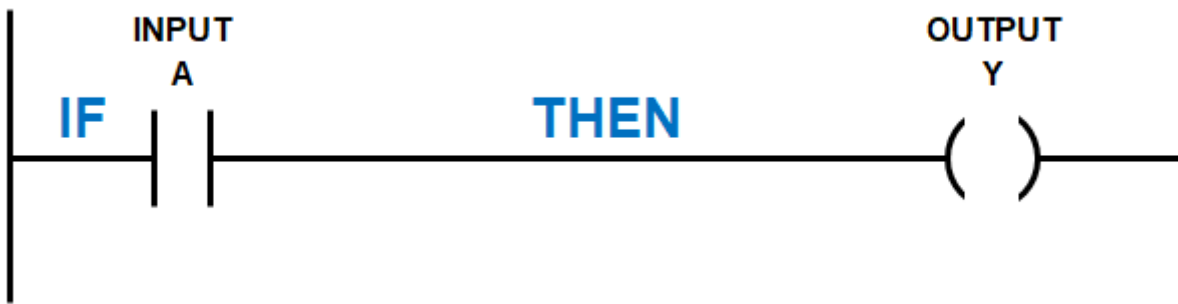
Reading Ladder Logic Diagrams – In-Built Functions

We can write out the logic expressed in the above as rung as **IF** A **THEN** Y.

Because PLC input A follows the binary concept it has two possible states, TRUE or FALSE.

Therefore it results in two possible logic iterations:
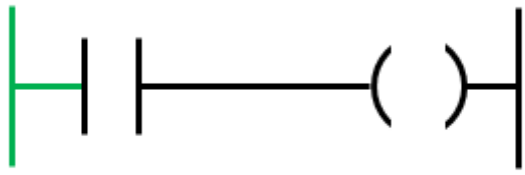**IF** A = FALSE **THEN** Y = FALSE
IF A = TRUE   **THEN** Y = TRUE

We also can express this in a truth table.

INPUT	OUTPUT
A	Y
FALSE	FALSE
TRUE	TRUE

Truth Table

If we translate this into a ladder logic diagram we express it symbolically in the form of a normally open (NO) contact for the input and a relay coil for the output. Remember the logic flow is from left to right and follows the same concept of current flow in an electric circuit. The ladder logic truth table is shown below.

INPUT	OUTPUT
A	Y

Ladder Logic Basics Truth Table – Hello World

In order to fast-track our understanding of ladder logic, there are three more fundamental logic functions that are essential to know.

You may be surprised, but when we combine these three functions with IF, THEN we will be able to program the majority of automation control requirements.

The three functions are:

1. NOT
2. AND
3. OR

Ladder Logic NOT Function

The result of the NOT function is basically the opposite state of an event that occurs. So if PLC input A is FALSE the result will be TRUE. And vise versa when PLC input A is TRUE the result will be FALSE.

The NOT Function is sometimes referred to as reverse logic. Check out the truth table below.

A	NOT A
FALSE	TRUE
TRUE	FALSE

Truth Table – NOT Function

If we translate a NOT function into a ladder logic diagram we express it symbolically in the form of a normally closed (NC) contact.

8

The ladder logic truth table is shown below.

Ladder Logic Basics Truth Table – NOT Function

Ladder Logic AND Function

The AND function examines multiple PLC inputs and has one resulting output.

If we translate an AND function into a ladder diagram we can express it symbolically in the form of two normally open (NO) contacts (PLC inputs A and B) and a relay coil (PLC output Y). They are all connected in line, just like a series connection in an electric circuit.
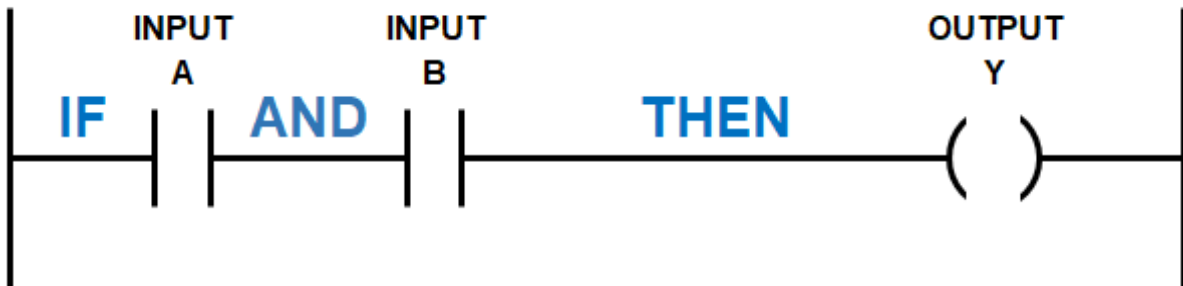
Ladder Logic Basics – AND Function

This time we have also highlighted the hidden AND function to illustrate the relationship between the ladder logic functions and the ladder diagram rung structure.

We can write out the logic expression above as **IF** A **AND** B **THEN** Y.

The AND function examines if all the PLC inputs are TRUE, then the corresponding result is also TRUE. However, if any one of the PLC inputs is FALSE then the corresponding result is also FALSE.

Because PLC input A and B follow the binary concept and are part of the AND function there are four possible logic iterations.
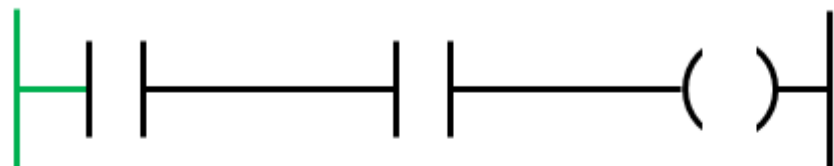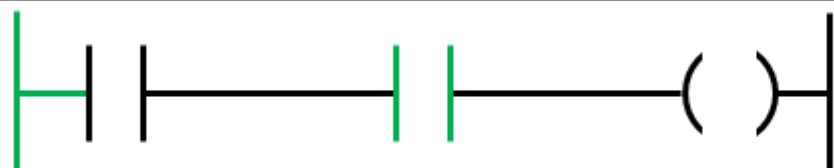
9

Check out the truth table below.

INPUTS		OUTPUT
A	B	Y
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

Truth Table – AND Function

The number of logic iterations increases with the number of PLC inputs (2PLC Inputs). But that doesn't matter too much with the AND function because the result can only be TRUE if all the PLC inputs are TRUE.

If we translate an AND function into a ladder logic truth table we get the table below.


Ladder Logic Basics Truth Table – AND Function

Ladder Logic OR Function

The OR function examines multiple PLC inputs and has one resulting output.

If we translate an OR function into a ladder diagram we can express it symbolically in the form of two normally open (NO) contacts (PLC inputs A and B) and a relay coil (PLC output Y). The inputs are placed in the rung in what is known as a branch. This is the equivalent of a parallel connection in an electric circuit.

The output is then connected in line with the rung.
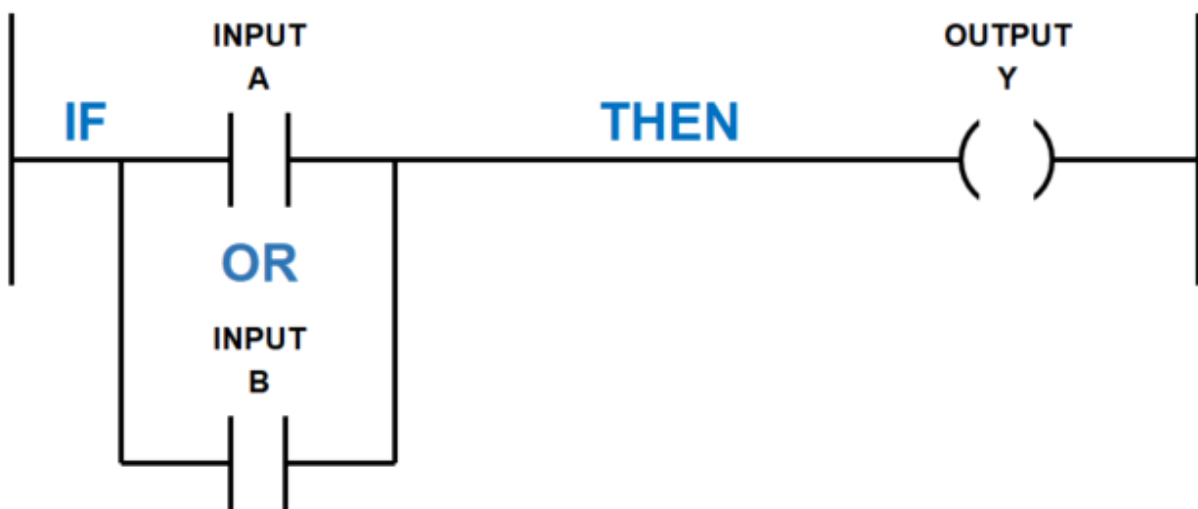
Ladder Logic Basics – OR Function

This time we have also highlighted the hidden OR function when we create a branch (parallel connection) with PLC input B across PLC input A.

We can write out the logic expression above as **IF** A **OR** B **THEN** Y.

The OR function examines if any of the PLC inputs are TRUE, then the corresponding result is also TRUE. However, all the PLC inputs must be FALSE in order for the corresponding result is also be FALSE.

Because PLC input A and B follows the binary concept and are part of the OR function there are four possible logic iterations.
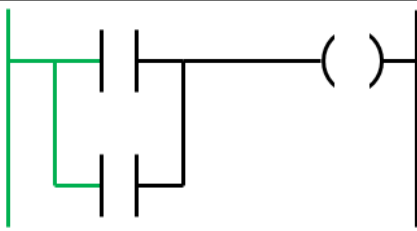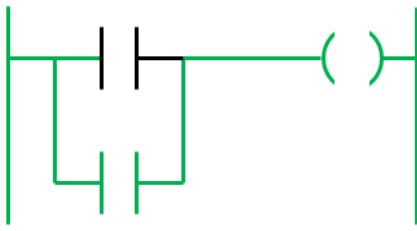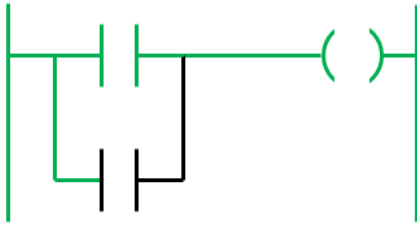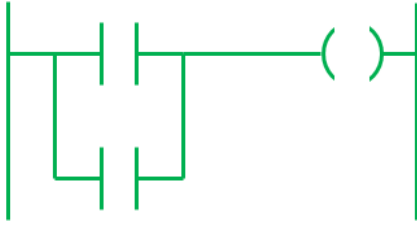
Check out the truth table below.

INPUTS		OUTPUT
A	B	Y
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

Truth Table – OR Function

Remember, the number of logic iterations increases with the number of PLC inputs (2PLC Inputs). But that doesn't matter too much with the OR function because the result can be TRUE if any of the PLC inputs are TRUE.

If we translate an OR function into a ladder logic truth table we get the table below….

Ladder Logic Basics Truth Table – OR Function

Well done! You now have a handle on the basics of ladder logic.

12