Kenyatta University

BSc Electrical and Electronic Engineering

EEE556 PROGRAMMABLE LOGIC CONTROLLERS

October 5, 2015

PROGRAMMABLE LOGIC CONTROLLERS

Prerequisites

EEE 305: Digital Electronics I
EEE 406: Microprocessor Systems and Applications

Purpose

To enable the student to understanding the operation and applications of PLC's.

Learning Outcomes

At the end of this course, the student should be able to;

1. Describe and explain the hardware architecture and mode of operation of Programmable Logic Controllers
2. Develop, write, monitor and debug basic PLC programmes
3. Program a PLC to perform simple control functions such as, timing, sequencing, counting, with the aid of ladder logic and hand held controllers.

Course Outline

Review of control systems. Overview of programmable logic controllers, input/output modules, signal conditioning and wiring.

PLC programming using ladder diagrams, functional block diagrams, statement lists, sequential function charts or structured programming, IEC 1131-3 programming.

Supervisory Control and Data Acquisition. PID control modules. Communications and automation. Application of PLCs in areas such as robotics, flexible manufacturing systems, batch and continuous processes and production monitoring.

Practicals with a PLCs such as the Mitsubishi, Siemens or Allen-Bradley series.

Teaching Methodology

Lectures: 2 hours per week; 3 hours per week for Laboratory/Tutorials.

Assessment

Ordinary Examination at end of Semester: 70

Instructional Materials/Equipment

1. Control Engineering laboratories;
2. Computer laboratory;
3. Overhead projector;

Prescribed Text Books

1. Jay F. Hooper (2006), Introduction to Plcs, 2nd edition, Carolina Academic Press. ISBN-13: 9781594603310.171
2. Frank Petruzella (2010), Programmable Logic Controllers, 4th edition, Career Education. ISBN-13: 978-0073510880.

References

1. Pradeep Kumar Srivastava (2004), Exploring Programmable Logic Controllers with Applications, B.P.B. Publications. ISBN-13: 9788176569330

1 Introduction to Programmable Logic Controllers

1.1 Review of Control Systems

Control system: an arrangement of physical components connected or related in such a manner as to command, direct, or regulate itself or another system.

A control system must have

- An input or inputs
- An output or outputs
- An arrangement to achieve this input-output combination

Control action is that quantity responsible for actuating the system to produce the output. Depending on such a control action, control systems are classified as

1.1.1 Open loop and closed systems

Open loop system is a control system in which the output has no effect on the control action i.e. the control action is totally independent of the output of the system. The output is neither measured nor fed back for comparison with the input. To each reference input, there corresponds a fixed operating condition and as a result the accuracy of the system depends on the calibration. Open loop control can be used, in practice, if the relationship between the input and output is known and if there are neither internal nor external disturbances.

A general open loop control system is shown in Figure 1 below.

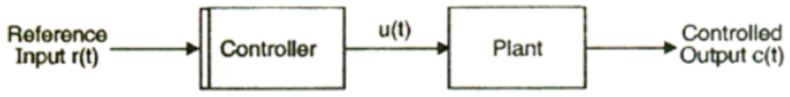

Figure 1: Open Loop Control System

The reference input r(t) is applied to the controller which generates the actuating signal u(t) to give the controlled output c(t).

Examples of open loop control systems

1. Automatic washing machine
2. Automatic hand drier
3. Traffic control by means of signal operated on time basis.
4. Bread toaster.

Advantages of open loop systems

1. They are simple in construction and design
2. They are easy to maintain
3. Convenient to use when the output is difficult to measure
4. Not much problems of stability

Disadvantages of open loop systems

1. Inaccurate and unreliable
2. Inaccurate results obtained with parameter variations, internal disturbances
3. Recalibration of the controller from time to time is necessary to maintain quality and accuracy

A closed loop system is a control system in which the control action is somehow dependent on the output of the system. There is a comparison of the state of the output and the reference state. This property is known as feedback and is the main difference between open loop and closed loop systems.

Feedback is the property of the system which permits the output to be compared with the reference input so that appropriate control action is formed

The generalized closed loop control system is shown in Figure 2 below

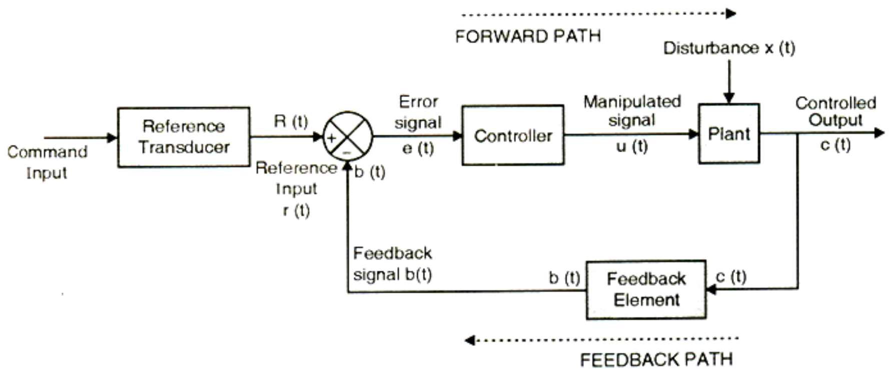

Figure 2: Closed Loop Control System

$r(t)$ - reference input

$u(t)$ - actuating signal

$e(t)$ - error signal

$c(t)$ - controlled output

$b(t)$ - feedback signal

Part of the output is fed back to the input. The comparison between the reference input $r(t)$ and the feedback $b(t)$ gives the error $e(t)$.

When feedback signal is positive, the system is called positive feedback system and $e(t) = r(t) + b(t)$.

When the feedback signal is negative, the system is called negative feedback system and $e(t) = r(t) + b(t)$.

The error is applied to the controller which gives the actuating signal $u(t)$. The action of the controller will be to drive the controlled output in such a manner so that the error is reduced to zero i.e. the feedback signal is equal to the input signal.

Examples of closed looped control systems include

1. Automatic electric iron
2. DC motor speed controlled by tacho feedback
3. Sun-seeker solar systems
4. Automatic water level controller.

Advantages of closed loop systems

1. Accuracy is very high since any error arising is corrected.
2. Reduced effects of non-linearities
3. It senses changes in input due to parameter changes, internal disturbances etc and corrects the same
4. High bandwidth
5. Facilitates automation

Disadvantages of closed loop systems

1. Complicated in design and maintenance costlier
2. System may become unstable

Comparison of open loop and closed loop systems

Table 1: Open loop and closed loop systems

Sr. No.	Open Loop System	Closed Loop System
1.	No feedback. Hence feedback elements absent	Feedback exists. Hence feedback elements exists
2.	No error detector	Error detector present
3.	It is inaccurate	It is accurate
4.	Highly sensitive to parameter changes	Less sensitive to parameter changes
5.	Small bandwidth	Large bandwidth
6.	Stable	May become unstable
7.	Economical	Costly
8.	Example: automatic toaster, hand drier	Examples: Temperature control oven, guided missile

1.1.2 Servomechanisms

Out of a number of closed loop systems in the industry, a particular group of applications are called servomechanisms. Here the controlled output is position, speed, velocity, acceleration etc. i.e. position or time derivative of position.

A servomechanism is a power amplifying feedback control system in which the controlled variable is mechanical position or its time derivative, such as velocity, acceleration.

Example: an automobile steering mechanism where the angular position of wheels on the road is controlled as shown in Figure 3 below

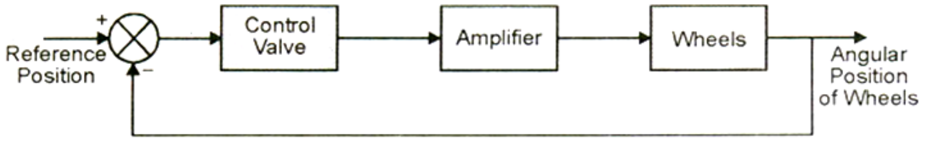

Figure 3: Servomechanism of automobile steering system

1.1.3 Linear and Non-linear Control Systems

Linear systems are systems which obey the principle of superposition and proportionality.

If an input x_1 produces output y_1 by proportionality theorem an input αx_1 produces an output αy_1. By superposition theorem, an input $\alpha x_1 + \beta x_2$ will produce an output $\alpha y_1 + \beta y_2$.

Linear systems are systems where mathematical tools like Laplace, Fourier etc. can be used. These systems can be analysed mathematically or graphically.

- Practical systems are not fully linear but exhibit certain non-linearities. The common examples of non-linearity are saturation, friction, deadzone, backlash etc.
- Non-linear systems exhibit peculiar behaviour
- They do not obey the law of superposition.
- Standard test signals approach cannot be used here
- The stability of linear systems depends on the root location and is independent of the initial state. In non-linear systems, stability depends on root location as well as the initial condition and type of input.
- Non-linear systems exhibit self sustained oscillations of fixed frequency and amplitude called Limit Cycles. Linear systems do not have this feature
- Non linear systems may exhibit hysteresis while linear systems do not show this behaviour

Table 2: Linear and Non Linear Systens

SR. NO	Linear Systems	Non Linear Systems
1	Obey superposition theorem	Do not obey superposition theorem
2	Can be analysed by standard test signals	Cannot be analysed by standard test signals
3	Stability depends only on the root location	Stability depends on the root location, initial condition and type of input
4	Do not exhibit Limit Cycles	Exhibit Limit Cycles
5	Do not exhibit Hysteresis / Jump Resonance	Exhibit Hysteresis /Jump Resonance
6	Can be analysed by Laplace, Fourier and Z-transforms	Cannot be analysed by these methods

1.1.4 Proportional, Integral and Derivative (PID) Controllers

Proportional (P) Control

The control action or signal $u(t)$ is proportional to the error signal $e(t)$ i.e

$$u(t) = k_p e(t)$$

where k_p is the proportional gain constant

The controller is an amplifier with a gain k_p(the gain maybe adjustable)

Taking the Laplace transform

$$U(s) = k_p E(s)$$

Integral Control
For integral control, the control signal $u(t)$ is given by

$$u(t) = k_i \int_0^t e(t)\,dt$$

where k_i is the integral gain constant
Converting to frequency domain

$$U(s) = \frac{k_i}{s}E(s)$$

Derivative Control
The control signal $u(t)$is given by

$$u(t) = k_d \frac{d}{dt} e(t)$$

where k_d is the derivative gain
In frequency domain

$$U(s) = sk_d E(s)$$

1.2 Introduction to PLC systems

A PLC is a microprocessor-based controller with multiple inputs and outputs. It uses a programmable memory to store instructions and carry out functions to control machines and processes. The PLC performs the logic functions of relays, timers, counters and sequencers. It has the following advantages:

Programming the PLC is easier than wiring the relay control panel.

1. The PLC can be reprogrammed. Conventional controls must be rewired and are often scrapped instead.
2. PLCs take less floor space then relay control panels.
3. Maintenance of the PLC is easier, and reliability is greater.
4. The PLC can be connected to the plant computer systems more easily than relays can.

1.2.1 Hardware Components of the PLC

A schematic diagram of a programmable logic controller is presented. The basic components of the PLC are the following

1. Input module

2. Output module
3. Processor
4. Memory
5. Power supply
6. Programming device

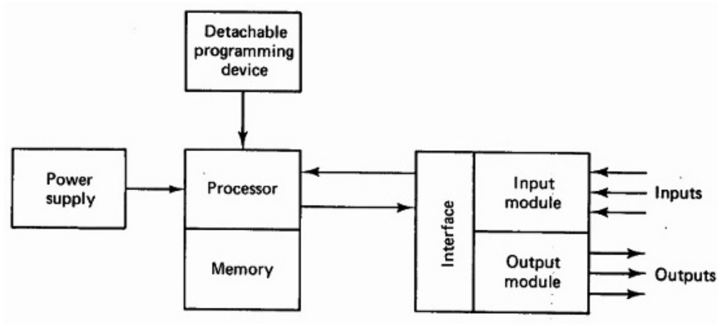

Figure 4: Diagram of the programmable logic controller

The components are housed in a suitable cabinet designed for the industrial environment.

The input module and output module are the connections to the industrial process that in to be controlled. The inputs to the controller are signals from limit switches, push-buttons, sensors, and other on-off devices. In addition, as we will describe later, larger PLCs are capable of accepting signals from analogue devices of the type modelled. The outputs from the controller are on-off signals to operate motors, valves, and other devices required to actuate the process.

The processor is the central processing unit (CPU) of the programmable controller. It executes the various logic and sequencing functions described in previous Sections by operating on the PLC INPUTS TO DETERMINE THE APPROPRIATE OUTPUT SIGNALS. The processor is a microprocessor, very similar in its construction to those used in personal computers and other data-processing equipment.

Tied to the CPU is the PLC memory, which contains the program of logic, sequencing, and other input/output operations. The memory for a programmable logic controller is specified in the same way as for a computer, and may range from 1 k to over 48 k of storage capacity.

A power supply is specially used to drive the PLC even though the components of the industrial process that are regulated may have a higher voltage and power rating than the controller itself.

The PLC is programmed by means of a programming device. The programming device (sometimes referred to as a programmer) is usually detachable from the PLC cabinet so that it can be shared between different controllers. Different PLC manufactures provide different devices, ranging from simple teach pendant-type devices, similar to those used in robotics, to special PLC programming keyboards and CRT displays.

1.2.2 Internal Architecture of the PLC

Figure 5 shows the basic internal architecture of a PLC.

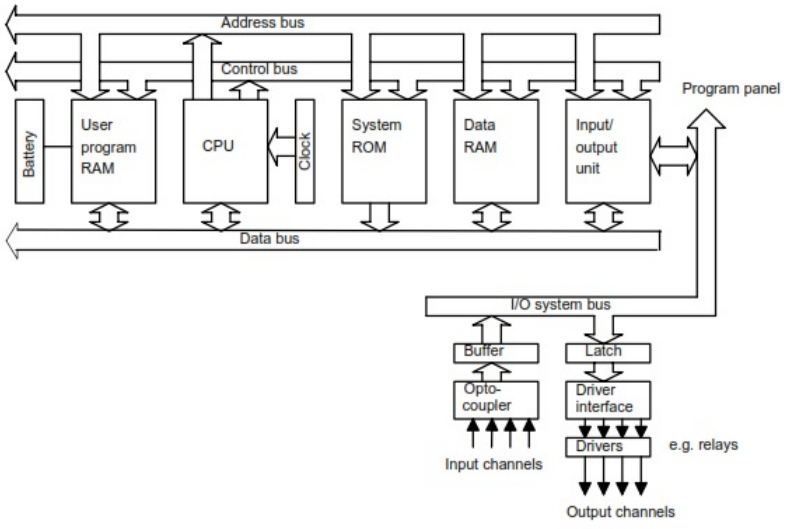

Figure 5: Internal Architecture of the PLC

It consists of a central processing unit (CPU) containing the system microprocessor, memory, and input/output circuitry. The CPU controls and processes all the operations within the PLC. It is supplied with a clock with a frequency of typically between 1 and 8 MHz. This frequency determines the operating speed of the PLC and provides the timing and synchronisation for all elements in the system. The information within the PLC is carried by means of digital signals. The internal paths along which digital signals flow are called buses. In the physical sense, a bus is just a number of conductors along which electrical signals can flow. It might be tracks on a printed circuit board or wires in a ribbon cable. The CPU uses the data bus for sending data between the constituent elements, the address bus to send the addresses of locations for accessing stored data and the control bus for signals relating to internal control actions. The system bus is used for communications between the input/output ports and the input/output unit.

The CPU

The internal structure of the CPU depends on the microprocessor concerned. In general they have:

1. An arithmetic and logic unit (ALU) which is responsible for data manipulation and carrying out arithmetic operations of addition and subtraction and logic operations of AND, OR, NOT and EXCLUSIVE-OR.
2. Memory, termed registers, located within the microprocessor and used to store information involved in program execution.
3. A control unit which is used to control the timing of operations.

The buses

The buses are the paths used for communication within the PLC. The information is transmitted in binary form, i.e. as a group of bits with a bit being a binary digit of 1 or 0, i.e. on/off states. Each of the bits is communicated simultaneously along its own parallel wire. The system has four buses:

1. The data bus carries the data used in the processing carried out by the CPU. A microprocessor termed as being 8-bit has an internal data bus which can handle 8-bit numbers. It can thus perform operations between 8-bit numbers and deliver results as 8-bit values.
2. The address bus is used to carry the addresses of memory locations. So that each word can be located in the memory, every memory location is given a unique address.
3. The control bus carries the signals used by the CPU for control, e.g. to inform memory devices whether they are to receive data from an input or output data and to carry timing signals used to synchronise actions.
4. The system bus is used for communications between the input/output ports and the input/output unit.

Memory

There are several memory elements in a PLC system:

1. System read-only-memory (ROM) to give permanent storage for the operating system and fixed data used by the CPU.
2. Random-access memory (RAM) for the user's program.
3. Random-access memory (RAM) for data. This is where information is stored on the status of input and output devices and the values of timers and counters and other internal devices. The data RAM is sometimes referred to as a data table or register table. Part of this memory, i.e. a block of addresses, will be set aside for input and output addresses and the states of those inputs and outputs. Part will be set aside for preset data and part for storing counter values, timer values, etc.

4. Possibly, as a bolt-on extra module, erasable and programmable read-only-memory (EPROM) for ROMs that can be programmed and then the program made permanent.

Input/output unit

The input/output unit provides the interface between the system and the outside world, allowing for connections to be made through input/output channels to input devices such as sensors and output devices such as motors and solenoids.

The input/output channels provide isolation and signal conditioning functions so that sensors and actuators can often be directly connected to them without the need for other circuitry. Electrical isolation from the external world is usually by means of opto-isolators (the term opto-coupler is also often used). Figure 6 shows the principle of an opto-isolator.

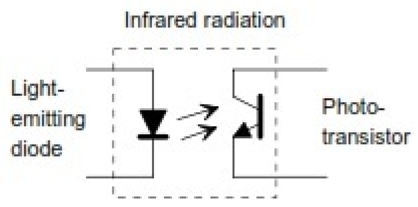

Figure 6: Opto-isolator

When a digital pulse passes through the light-emitting diode, a pulse of infra-red radiation is produced. This pulse is detected by the photo-transistor and gives rise to a voltage in that circuit. The gap between the light-emitting diode and the photo-transistor gives electrical isolation but the arrangement still allows for a digital pulse in one circuit to give rise to a digital pulse in another circuit.

The digital signal that is generally compatible with the microprocessor in the PLC is 5 V d.c. However, signal conditioning in the input channel, with isolation, enables a wide range of input signals to be supplied to it (see Chapter 3 for more details). A range of inputs might be available with a larger PLC, e.g. 5 V, 24 V, 110 V and 240 V digital/discrete, i.e. onâoff, signals (Figure 7). A small PLC is likely to have just one form of input, e.g. 24 V.

The output from the input/output unit will be digital with a level of 5 V. However, after signal conditioning with relays, transistors or triacs, the output from the output channel might be a 24 V, 100 mA switching signal, a d.c. voltage of 110 V, 1 A or perhaps 240 V, 1 A a.c., or 240 V, 2 A a.c., from a triac output channel (Figure 8). With a small PLC, all the outputs might be of one type, e.g. 240 V a.c., 1 A. With modular PLCs, however, a range of outputs can be accommodated by selection of the modules to be used.

Outputs are specified as being of relay type, transistor type or triac type

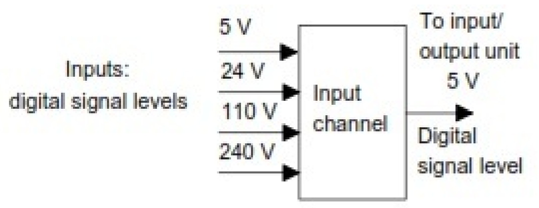

Figure 7: Input level

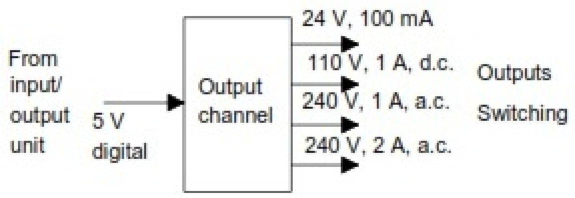

Figure 8: Output level

1. With the relay type, the signal from the PLC output is used to operate a relay and is able to switch currents of the order of a few amperes in an external circuit. The relay not only allows small currents to switch much larger currents but also isolates the PLC from the external circuit. Relays are, however, relatively slow to operate. Relay outputs are suitable for a.c. and d.c. switching. They can withstand high surge currents and voltage transients.

2. The transistor type of output uses a transistor to switch current through the external circuit. This gives a considerably faster switching action. It is, however, strictly for d.c. switching and is destroyed by over-current and high reverse voltage. As a protection, either a fuse or built-in electronic protection are used. Opto-isolators are used to provide isolation.

3. Triac outputs, with opto-isolators for isolation, can be used to control external loads which are connected to the a.c. power supply. It is strictly for a.c. operation and is very easily destroyed by over-current. Fuses are virtually always included to protect such outputs.

1.2.3 Operation of the PLC

The operation of a programmable controller is relatively simple. The input/ output (I/O) system is physically connected to the field devices that are encountered in the machine or that are used in the control of a process. These field devices may be discrete or analogue input/output devices, such as limit switches, pressure transducers, push buttons, motor starters, solenoids, etc. The I/O interfaces provide the connection between the CPU and the information providers (inputs) and controllable devices (outputs).

During its operation, the CPU completes three processes: (1) it reads, or accepts, the input data from the field devices via the input interfaces, (2) it executes, or performs, the control program stored in the memory system, and (3) it writes, or updates, the output devices via the output interfaces. This process of sequentially reading the inputs, executing the program in memory, and updating the outputs is known as scanning. Figure 9 illustrates a graphic representation of a scan.

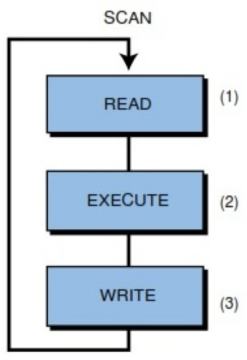

Figure 9: Illustration of scan

2 PLC Programming

There are different methods of programming PLCs, some of which include

1. Ladder Diagram Programming
   Ladder Diagram is very similar to a circuit diagram. Symbols such as contacts and coils are used. This programming language appeals to those who 'grew up' with contactors.
2. Functional Block Diagram (FBD)
   The Function Block Diagram uses 'boxes' for the individual functions. The character in the box indicates the function (e.g. –> AND Logic Operation). This programming language has the advantage that even a 'non-programmer' such as a process engineer can work with it.
3. Statement List (STL) Programming
   The Statement List consists of STEP 7 instructions. You can program fairly freely with STL (sometimes to the point of being unable to follow it any more). This programming language is preferred by programmers who are already familiar with other programming languages.

2.1 Ladder Diagram Programming

The main method for PLC programming is called ladder logic. It is a graphical programming language that uses graphical symbols to provide the PLC with the logical instructions needed to perform control operations.Ladder logic programs resemble relay logic schematics.

Ladder logic is so named because the diagram looks like a ladder. Each step in the program is called a rung. The vertical lines on the left and right are the power rails. Each rung defines one operation in the control process. The ladder diagram is read from left to right and from top to bottom. Each rung starts with one or more inputs and ends with at lease one output.

There are quite a few manufacturers of PLCs. Each has its own brand of ladder logic programming. Though they are all very similar and if you can program in one manufacturer's ladder logic language it is easy to use them all. Figure 10 shows a few standard symbols.

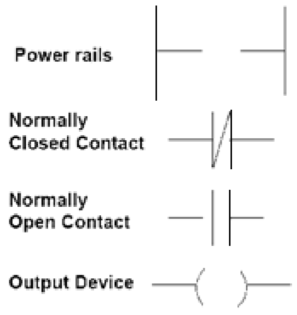

Figure 10: Ladder Logic Symbols

To illustrate the drawing of the rung of a ladder diagram, consider a situation where the energising of an output device, e.g. a motor, depends on a normally open start switch being activated by being closed. The input is thus the switch and the output the motor. Figure 11 shows the ladder diagram.

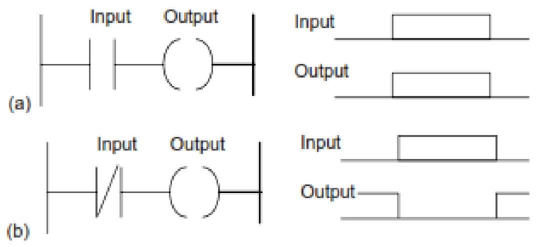

Figure 11: Ladder Rung

In drawing ladder diagrams the names of the associated variable or addresses of each element are appended to its symbol. Thus Figure 12 shows how the ladder diagram of Figure 11 would appear using (a) Mitsubishi, (b) Siemens, (c) Allen-Bradley, (d) Telemecanique notations for the addresses.

Figure 12: Notations (a) Mitsubishi, (b) Siemens, (c) Allen-Bradley, (d) Telemecanique

In drawing a ladder diagram, certain conventions are adopted:

1. 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.
2. Each rung on the ladder defines one operation in the control process.
3. A ladder diagram is read from left to right and from top to bottom, Figure 13 showing the scanning motion employed by the PLC. The top rung is read from left to right. Then the

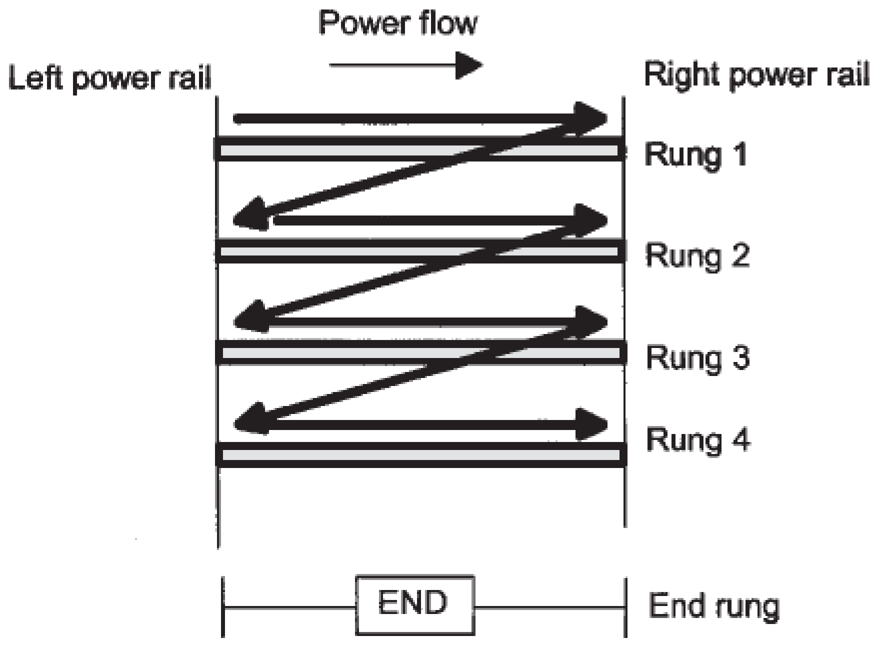

Figure 13: Scanning the ladder program

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. This procedure of going through all the rungs of the program is termed a cycle. 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.

4. 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, used as an input to the PLC. The term output is used for a device connected to the output of a PLC, e.g. a motor.

5. Electrical devices are shown in their normal condition. Thus a switch which 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.

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

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

2.1.1 Logic Functions

AND Logic

Figure 14(a) shows a situation where an output is not energised unless two, normally open, switches are both closed. Switch A and switch B have both to 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 14(b))

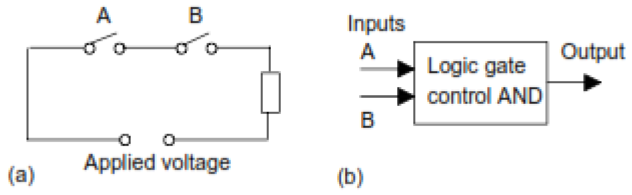

Figure 14: AND Logic

Such an operation is said to be controlled by a logic gate and the relationship between the inputs to a logic gate and the outputs is tabulated in a form known as a truth table. Thus for the AND gate we have:

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

Figure 15 (a) shows an AND gate system on a ladder diagram. The ladder diagram starts with a normally open set of contacts labelled input A, to represent switch A and in series with it, another normally open set of contacts labelled input B, to represent switch B. The line then terminates with O to represent the output. For there to be an output, both input A and input B have to occur, i.e. input A and input B contacts have to be closed (Figure 15(b)).

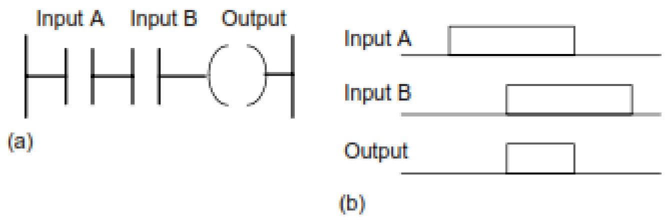

Figure 15: AND Logic Ladder Rung

OR Logic

Figure 16 (a) shows an electrical circuit where an output is energised when switch A or B, both normally open, are closed. This describes an OR logic gate (Figure 16 (b) in that input A or input B must be on for there to be an output. The truth table is:

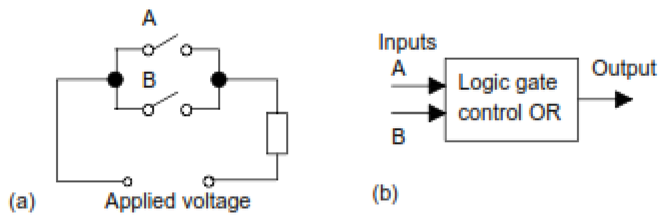

Figure 16: OR Logic

Such an operation is said to be controlled by a logic gate and the relationship between the inputs to a logic gate and the outputs is tabulated in a form known as a truth table. Thus for the OR gate we have:

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

Figure 17(a) shows an OR logic gate system on a ladder diagram, Figure 17(b) showing an equivalent alternative way of drawing the same diagram. The ladder diagram starts with, normally open contacts labelled input A, to represent switch A and in parallel with it, normally open contacts labelled input B, to represent switch B. Either input A or input B have to be closed for the output to be energised (Figure 17(c)). The line then terminates with O to represent the output.

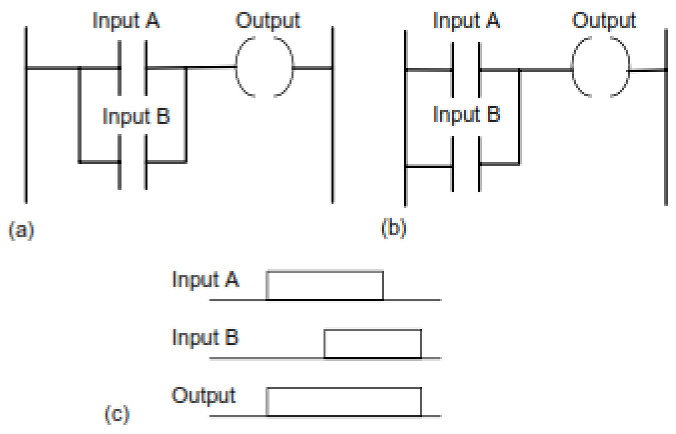

Figure 17: OR Logic Ladder Rung

NOT Logic

Figure 18(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. This illustrates a NOT gate in that there is an output when there is no input and no output when there is an input (Figure 18(c)). The gate is sometimes referred to as an inverter. The truth table is:

A	Output
0	0
1	1

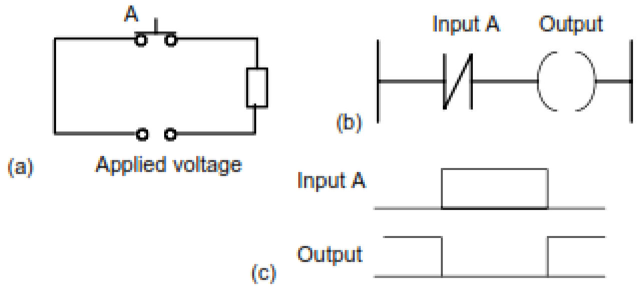

Figure 18: NOT Logic (a) Circuit, (b)Ladder rung, (c) input/ouput profile

Figure 18(b) shows a NOT gate system on a ladder diagram. The input A contacts are shown as being normally closed. This is in series with the output. With no input to input A, the contacts are closed and so there is an output. When there is an input to input A, it opens and there is then no output.

NAND Logic

Suppose we follow an AND gate with a NOT gate (Figure 19(a)). The consequence of having the NOT gate is to invert all the outputs from the AND gate. An alternative, which gives exactly the same results, is to put a NOT gate on each input and then follow that with OR (Figure 19(b)).

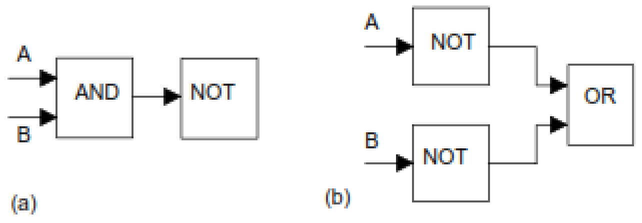

Figure 19: NAND Logic

Both the inputs A and B have to be 0 for there to be a 1 output. There is an output when input

A and input B are not 1. The combination of these gates is termed a NAND gate. The truth table is given as

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

Figure 20 shows a ladder diagram which gives a NAND gate. When the inputs to input A and input B are both 0 then the output is 1. When the inputs to input A and input B are both 1, or one is 0 and the other 1, then the output is 0.

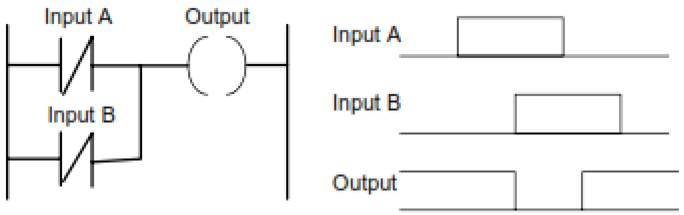

Figure 20: NAND Logic Ladder Rung

NOR Logic

Suppose we follow an OR gate by a NOT gate (Figure 21(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 21(b)). The following is the resulting truth table:

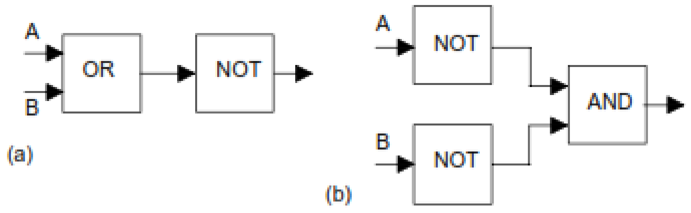

Figure 21: NOR Logic

The following is the resulting truth table:

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

Figure 22 shows a ladder diagram of a NOR system

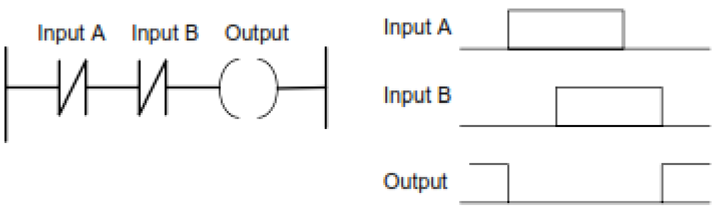

Figure 22: NOR Logic Ladder Rung

XOR Logic

The OR gate gives an output when either or both of the inputs are 1. Sometimes there is, however, a need for a gate that gives an output when either of the inputs is 1 but not when both are 1, i.e. has the truth table:

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

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 23.

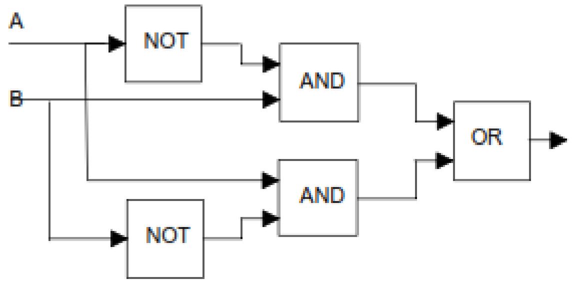

Figure 23: XOR Logic

Figure 30 shows a ladder diagram for an XOR gate system. When input A and input B are not activated then there is 0 output. When just input A is activated, then the upper branch results in the output being 1. When just input B is activated, then 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 have as many sets of contacts as necessary.

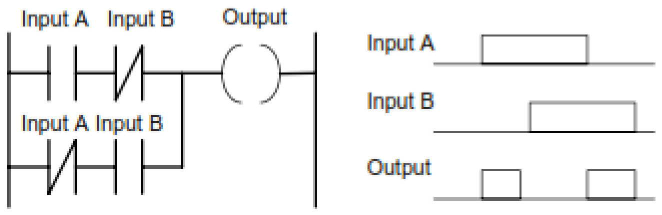

Figure 24: XOR Logic Ladder Rung

XNOR Logic

Suppose we follow an XOR gate by a NOT gate. The consequence of having the NOT gate is to invert the outputs of the XOR gate. The following is the resulting truth table:

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

2.1.2 Latching

There are often situations where it is necessary to hold an output energised, even when the input ceases. A simple example of such a situation is a motor which 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 used to carry out such an operation. It is a self-maintaining circuit in that, after being energised, it maintains that state until another input is received.

An example of a latch circuit is shown in Figure 25.

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 the input A opens, the circuit will still maintain the output energised. The only way to release the output is by operating the normally closed contact B.

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

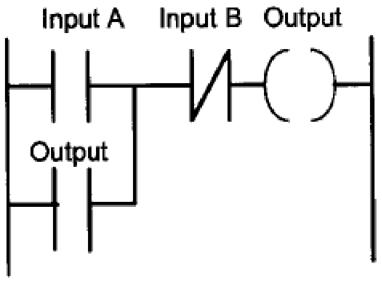

Figure 25: Latched Circuit

applied to the motor and another when it is not applied. Figure 26 shows the ladder diagram with Mitsubishi notation for the addresses.

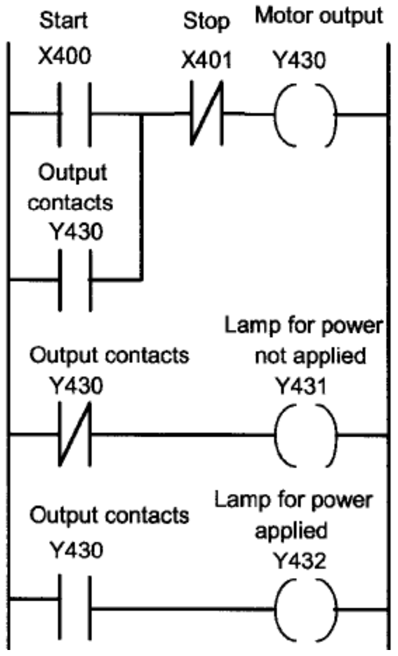

Figure 26: Motor on-off, with signal lamps, ladder diagram

X401 is closed when the program is started. When X400 is momentarily closed, Y430 is energised and its contacts close. This results in latching and also 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 off. Latching is widely used with start-ups so that the initial switch on of an application becomes latched on.

2.1.3 Multiple Outputs

With ladder diagrams, there can be more than one output connected to a contact. Figure 27 shows a ladder program with two output coils. When the input contacts close both the coils give outputs.

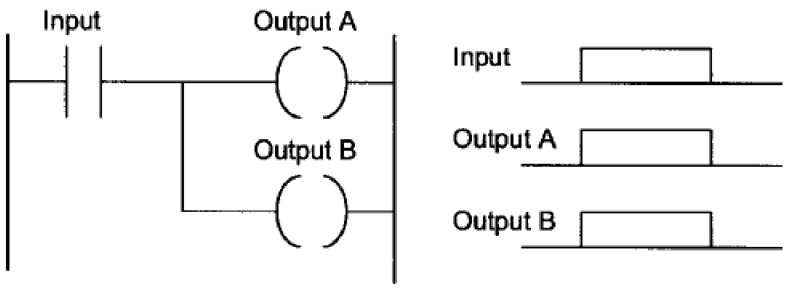

Figure 27: Ladder rung with two outputs

For the ladder rung shown in Figure 28, output A occurs when input A occurs. Output B only occurs when both input A and input B occur.

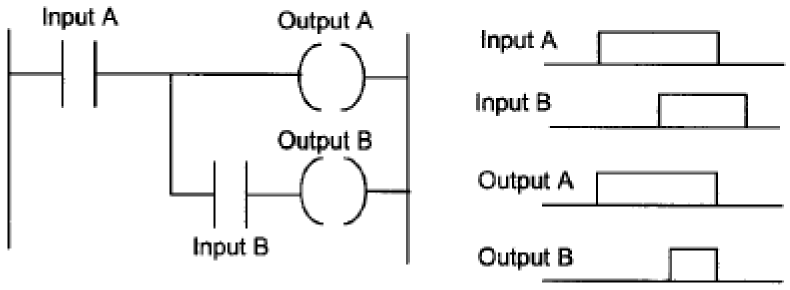

Figure 28: Ladder rung with two inputs and two outputs

Such an arrangement enables a sequence of outputs to be produced, the sequence being in the sequence with which contacts are closed. Figure 29 illustrates this with the same ladder program in Siemens notations.

Outputs A, B and C are switched on as the contacts in the sequence given by the contacts A, B and C are being closed. Until input A is closed, none of the other outputs can be switched on. When input A is closed, output A is switched on. Then, when input B is closed, output B is switched on. Finally, when input C is closed, output C is switched on.

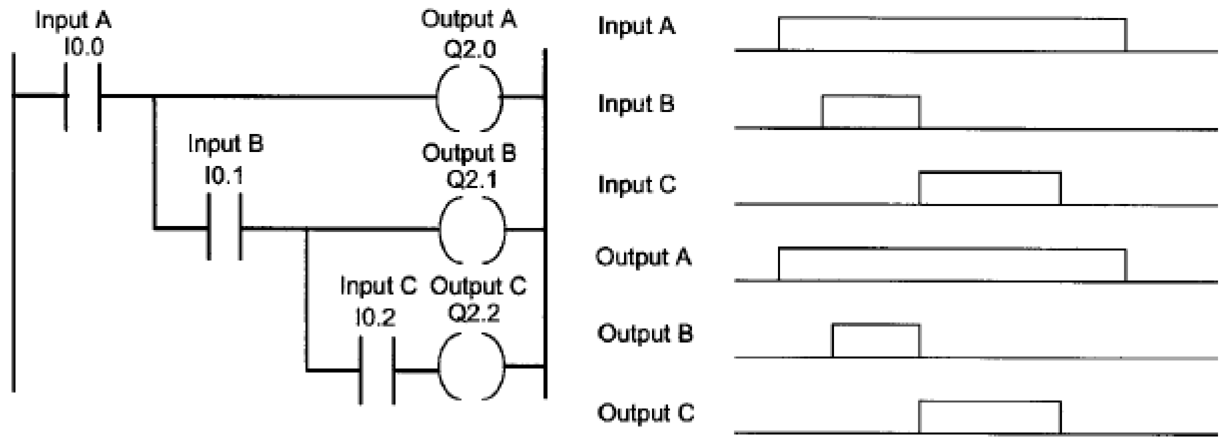

Figure 29: Sequenced Outputs

2.2 Functional Block Diagram (FBD)

The term function block diagram (FBD) is used for PLC programs described in terms of graphical blocks. It is described as being a graphical language for depicting signal and data flows through blocks, these being reusable software elements. A function block is a program instruction unit which, when executed, yields one or more output values. Thus, a block is represented in the manner shown in Figure below with the function name written in the box

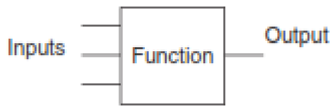

Figure 30: Function Block

A function block is depicted as a rectangular block with inputs entering from the left and outputs emerging from the right. Function blocks can have standard functions, such as those of the logic gates or counters or times, or have functions defined by the user, e.g., a block to obtain an average value of inputs.

2.2.1 Blocks for Logic Gates

A negated input is represented by a small circle on the input, a negative output by a small circle on the output (Figure 31)

Figure 32 shows the functional blocks for the various logic gates

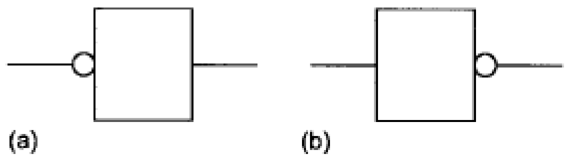

Figure 31: (a)Negated Input (b) Negated Output

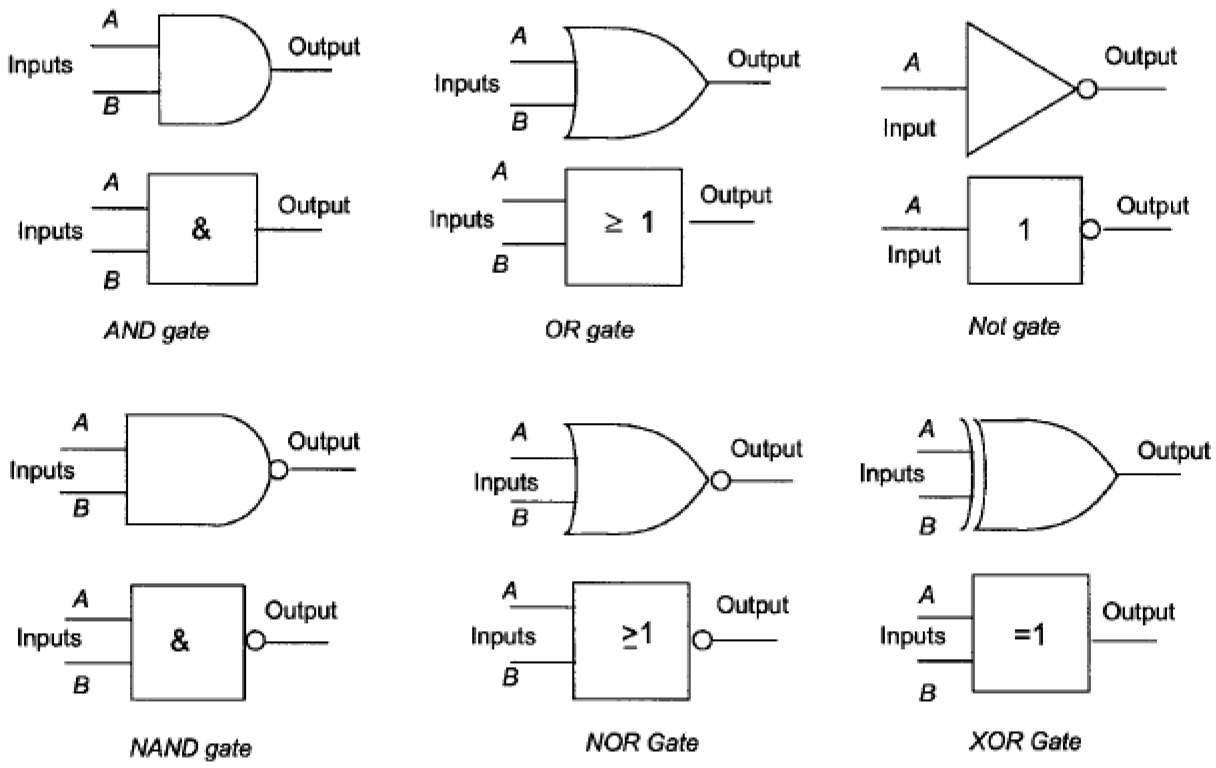

Figure 32: Function Blocks for Logic Gates

2.2.2 Boolean Logic

As an illustration of how we can relate Boolean expressions with ladder diagrams, consider the expressions:

$$A + BC = Q$$

This tells us that we have A or the term B and C giving the output Q. Figure 33 shows the ladder and functional block diagrams.

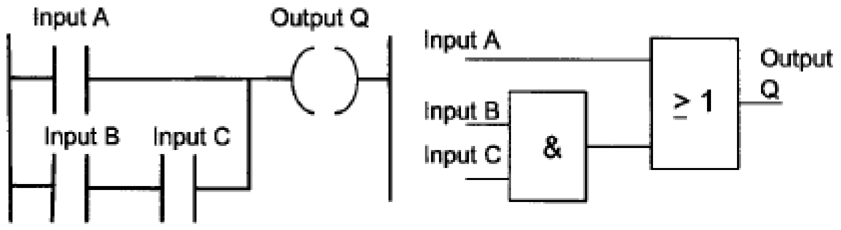

Figure 33: FBD implementation

$$A + \overline{B} = Q$$

The Function Block Diagram is

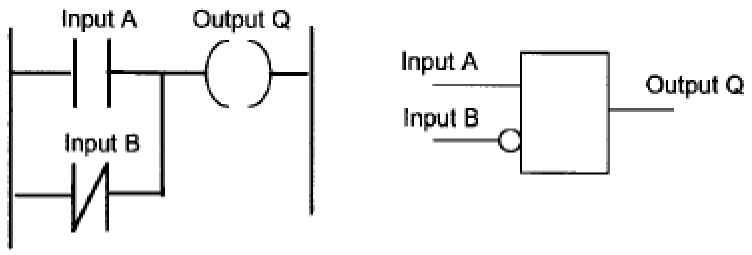

Figure 34: FBD implementation

Consider a logic diagram with many inputs, as shown in Figure 35, and its representation by a Boolean expression and a ladder rung.

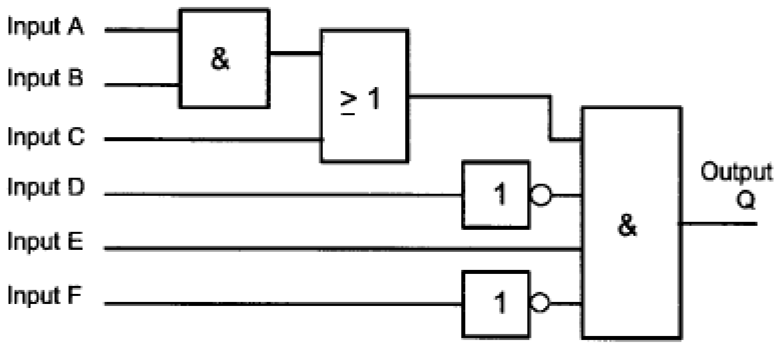

Figure 35: Example of FBD

For inputs A and B we obtain an output from the upper AND gate of $A.B$. From the OR gate we obtain an output of $A.B + C$. From the lower AND gate we obtain an output Q of:

$$(A.B + C) - \overline{D}.E.\overline{F} = Q$$

The ladder diagram to represent this is shown in Figure 36.

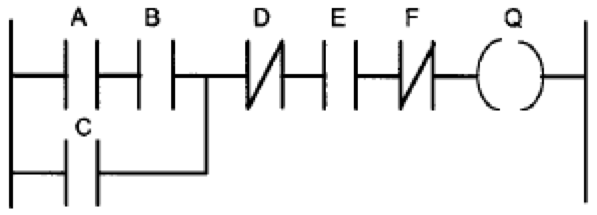

Figure 36: Implementable ladder logic